Service Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Recommending Products and Services Belonging to Online Businesses Using Intelligent Agents

Adrian Alexandrescu, Cristian Nicolae Butincu, Mitică Craus

To cite this article:
Adrian Alexandrescu, Cristian Nicolae Butincu, Mitică Craus (2017) Recommending Products and Services Belonging to Online Businesses Using Intelligent Agents. Service Science 9(4):338-348. https://doi.org/10.1287/serv.2017.0188

http://pubsonline.informs.org/journal/serv/

SERVICE SCIENCE
Vol. 9, No. 4, December 2017, pp. 338–348
ISSN 2164-3962 (print), ISSN 2164-3970 (online)

Recommending Products and Services Belonging to Online Businesses Using Intelligent Agents

Adrian Alexandrescu,[a] Cristian Nicolae Butincu,[a] Mitică Craus[a]

[a] Faculty of Automatic Control and Computer Engineering, Gheorghe Asachi Technical University of Iasi, Iasi, 700050 Romania

Contact: aalexandrescu@tuiasi.ro, http://orcid.org/0000-0003-0663-4795 (AA); cbutincu@tuiasi.ro (CNB); craus@tuiasi.ro, http://orcid.org/0000-0002-4136-9387 (MC)

Received: November 30, 2016
Revised: May 19, 2017; July 29, 2017
Accepted: September 14, 2017
Published Online: December 12, 2017

https://doi.org/10.1287/serv.2017.0188

Abstract. A sure method for a business organization to sell more products is to expand its customer base and to have its products recommended by other organizations and individuals. This paper takes a look at the techniques used by shopping websites in order to entice the user in purchasing their products, and proposes a system for recommending products and services provided by different online businesses to potential customers. The solution is built upon a service-oriented architecture that allows businesses to share information regarding customers' purchases while taking into account the user privacy issue. Intelligent agents, which rely on a product type association dynamically weighted graph, are employed in order to obtain and to process the information needed to make the suggestions. The use of intelligent agents significantly improves the quality of the recommendations made by the system. This improvement is achieved by suggesting products and services depending on other users' purchasing patterns while also considering the different product types and quantities sold by the business organizations that are part of the system.

Keywords: service design • service marketing • intelligent agents • social computing • recommendation system

1. Introduction

Online businesses thrive when they are offering products and services that are tailored to the customer's needs and expectations and, every so often, the customer receives recommendations regarding other somewhat related products and also can give his input in terms of the quality of the purchased products and services.

Service systems are an important component of a business model that wants to be successful, especially in an online environment. The concept of service system is defined as "dynamic value cocreation configurations of resources" (Maglio and Spohrer 2008, p. 1), and it refers to the creation of value by both the business organization and the customer; this is achieved by allowing the customer to tailor the services in a way that is meaningful to them. Prahalad and Ramaswamy (2004, 2013) identify four building blocks of cocreation: dialogue, access, risk, and transparency, all these translate into a constant interaction between the customer and the business organization. The service science referring to these types of systems and the innovations that can improve the service systems are discussed in Spohrer and Maglio (2008) and there is also new research in this field shown in Lin et al. (2014).

Besides underlining the importance of the relationship between the customer and the business organization, another key aspect refers to the infrastructure required to promote and deliver the products and services. There are many companies that offer easy access to platform-as-a-service (PaaS) architectures in order for a business organization to quickly make its services available to the general public. Using a service-oriented architecture poses more issues, some of which are discussed in Demirkan et al. (2009), especially from a managerial point of view. In terms of service science and service management, there are several solutions that expand the PaaS concept; for example, Boniface et al. (2010) present a PaaS architecture that focuses on a real-time quality of service management. In this instance, web service architectures are also used for a more streamlined communication between the actors and components of service delivery; for example, Sundaram et al. (2010) use such an architecture for radio-frequency identification applications in supply chain management.

The next step for a business organization, after having a product and service delivery architecture in place, is to make the general public aware of the products that are available for purchase. Nowadays, people are assaulted with offers for different products and services by means of advertising posters on the streets and in the stores, by receiving regular mail catalogs from various companies and, when using the Internet, by means of banners and ads. One's activity on the web is monitored to some degree and, based on what the user visited and was interested in, the user is served targeted ads when accessing certain websites. For example, if a user is

looking at a specific product on a shopping site, then he will see offers for products in the same category when going to another nonshopping website that employs the same ad network.

The different ad networks use different technologies and techniques like monitoring user behavior or displaying ads based on search queries and the textual relevance of the web page the ads appear in (Mei et al. 2011). Notable solutions for showing ads include Google AdSense and AdWords, Chitika, Yahoo! Bing Network, Facebook Paid Ads, and Kontera. The way the user sees ads varies from banners showing the recommended products to in-text advertising. In the second situation, certain keywords are highlighted and the ad is displayed when the user moves the mouse over the keyword, or the ad is seamlessly blended in order to make it appear as belonging to that web page.

Having targeted ads served to the user raises security and user privacy concerns. Castelluccia et al. (2012) show that the profile of a user can be reconstructed by a third party with a fairly high precision by having access to a small number of websites that employ Google Ads. This is because of the fact that Google Ads is widely spread and anyone can adhere to the advertising network without any screening.

This paper is a continuation of the research done by Alexandrescu et al. (2016), which proposed a service-oriented architecture used for improving after-sales services that used mobile agents in order to handle the communication between the businesses and the central node of the system.

The new research presented in the current paper proposes an application of the aforementioned architecture by adding a module that handles the ads displayed by the businesses which adhered to the service-oriented system, that is, a recommendation module that adapts different techniques and concepts like collaborative filtering and user profiling in order to offer relevant products and services. The module uses a product type association dynamically weighted graph in order to have complementary products and services for each purchased product. The recommendations are made using a combination employing the association graph, the product types, and quantities sold by each business and the user profile created by the system.

2. Problem Statement

Online shopping is the easiest method for purchasing products and services with minimum effort from the customer. Business organizations have websites that sell a wide range of products from different manufacturers, where customers log in, add the desired products to the shopping cart, and usually pay online using an e-payment solution. Everything is done so that the customer has a good experience in using the website up until the moment they make their purchase.

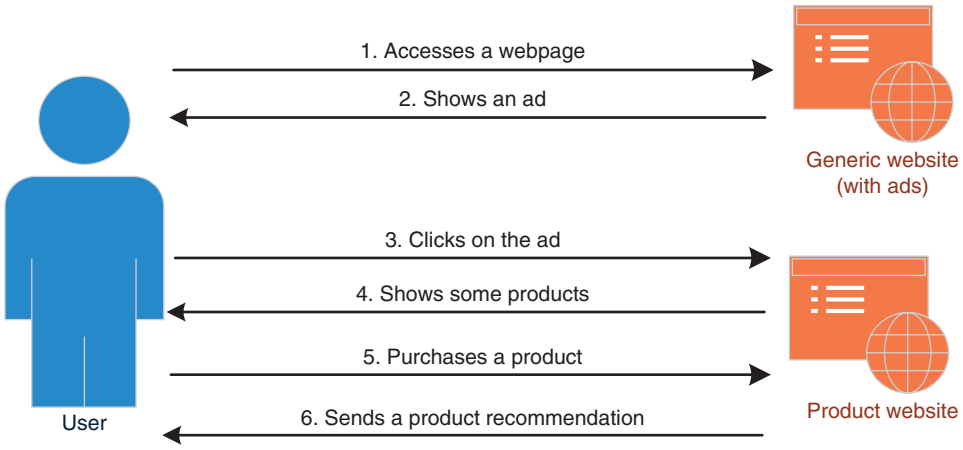
To increase their revenue, online businesses use different techniques to convince the potential customer to buy their products. For example, if a logged-in user adds something in the shopping cart but does not finish his purchase at that particular time, several hours or days later, the user can receive an email from the online business reminding him to complete his purchase. Another more common example of making a potential customer more likely to purchase something is by showing the user various ads with product recommendations. The online business can join an ad network for a fee so that their products get recommended to potential customers when they visit different websites that are part of the same network. Those websites mostly have nothing to do with the products that advertised on their pages; the ad system handles what products are recommended on which websites. An example of an interaction flow between the customer, the website with the ads, and the online business is shown in Figure 1.

Ideally, the recommendation system must have only one goal: to recommend products and services that the user needs. This means that sometimes a product, which the user does not necessarily expect, might be recommended. In reality, based on the user profile, the system can draw the conclusion that the user's life would be somewhat improved by purchasing that product. In other words, suggest a product that the user needs but does not know it yet. This does not have to be taken to the extreme by recommending a product that the user cannot afford, thus causing undue frustration and a dismissal of the website that made that recommendation.

Taking all these into account, the purchasing patterns must determine three things: *what the user expects*, *what the user needs*, and *what the user affords*.

From the perspective of a business organization, having its products recommended, seemingly more or less randomly on different websites is, in part, a good thing because it raises awareness regarding the business and what it offers. However, it can leave the potential customer reluctant to purchase from that online business because of the aggressiveness and the way the recommendation is made. In this particular case there is such a thing as bad publicity. Businesses increase their revenue by selling more products and services. This can be achieved by having repeat customers, that is, customers that come back to purchase more products, and by word-of-mouth, that is, customers tell other potential customers about the purchased products and their quality.

Figure 1. Example of an Interaction Flow Between a Customer, a Website with Ads, and a Website with Products



3. Related Work

The novel research presented in this paper expands on the system proposed by Alexandrescu et al. (2016), where mobile agents are used in a service-oriented architecture in order to handle the communication between the business organization and the system. One of the key points there is exemplifying a possible communication protocol between a recommendation module and the rest of the actors from the system (i.e., the central system node, the mobile agents, and the business organizations). The internal structure and logic of the recommendation module is not discussed in depth because the focus is only on the interaction between components. In the current paper, the main research refers to a possible and efficient architecture for that recommendation module.

Another base point for the presented research is the generic solution for performing actions on business organization items described by Puiu and Alexandrescu (2016), where the authors suggest a possible improvement of the proposed system by having the system make recommendations for other related/complementary business items. For example, if a customer uses the system to rent a room at a hotel, the system can recommend making a reservation at a nearby restaurant, buying a ticket at a nearby theatre, or renting a car form a business located in that area. The current paper expands on that logic to recommending products and services belonging to a wider range of seemingly unrelated businesses in the system.

The solution proposed in this paper takes the positive aspects of two concepts: ad networks and recommendation systems with collaborative filtering.

Regarding ad networks, Davis (2006) takes a look at web advertising and focuses on Google AdSense, which delivers ads that are targeted to the website they appear in, and on Google AdWords, which delivers ads related to specific keywords. In the world of online advertising there are several methods by which the people who display ads on their websites make money, for example, pay-per-click, cost-per-thousand-impressions, or cost-per-acquisition. Mookerjee et al. (2012) present a decision model working with a click-through rate for showing ads delivered by the Chitika online advertising firm in order to maximize the firm's revenue. A very important issue when using ad networks is click spam or click fraud, that is, simulating user clicks in order to make the company whose products are advertised lose money. Zhang and Guan (2008) and Dave et al. (2012) discuss techniques of measuring and detecting click fraud, and they both conclude the severity of the problem (especially in mobile ads) and the fact that the efficiency of click-fraud prevention methods implemented by the ad networks is not guaranteed. For this reason, the solution proposed in this paper offers an alternative to current methods of having ads placed on different websites by eliminating the money incentive of allowing those ads to be placed on one's website.

In terms of recommendation methods, the most widely used is collaborative filtering, but other methods like content based and demographic based prove to be efficient solutions. Onofrei and Archip (2016) provide an introduction into collaborative filtering and content-based recommendation methods, present the specific concepts, and discuss possible improvements. In collaborative filtering, the main principle is that if user A has the same opinion as user B regarding item X, then user A is more likely to have user B's opinion regarding item Y. The solution presented in the current research has its inspiration from the collaborative filtering method, but it uses a simplified technique in order to generate a relationship between products based on the users' purchasing behavior.

Intelligent agents are used by Hostler et al. (2012) in a simulated movie shopping environment in order to make recommendations for other movies that the user can buy. This is similar to the recommendations made by the Internet Movie Database (IMDb) and the authors focus only on movies, as opposed to the research presented in this paper where multiple product types are taken into consideration.

An increase in the revenue of a business organization can be obtained by offering the customer after-sales services. In this manner, the customer becomes more likely to purchase again from that online business. For example, when a customer purchases a product, he can be offered an extended guarantee, another free product, or another product at a discount. Such a framework for offering after-sales services based on a different configuration model is presented by Legnani et al. (2009). The research presented there has only tangential relevance with the work in the current paper in the sense that the method of delivering the recommendation in our proposed system can be seen as offering the customer an after-sales service that offers the chance to purchase another needed product.

4. Proposed Solution Architecture

The goal is to have a system to which online businesses can adhere in order to increase revenue and that allows potential customers to purchase complementary products from the other businesses in the system. Recommendations are made based on the user's needs and expectations, and the product types and quantities sold by each business in the system. No business organization is at a disadvantage because the number of times the products and services of an organization are recommended is determined mostly on its implication in the system, that is, relevance of the product types and the quantities of items that are being sold.

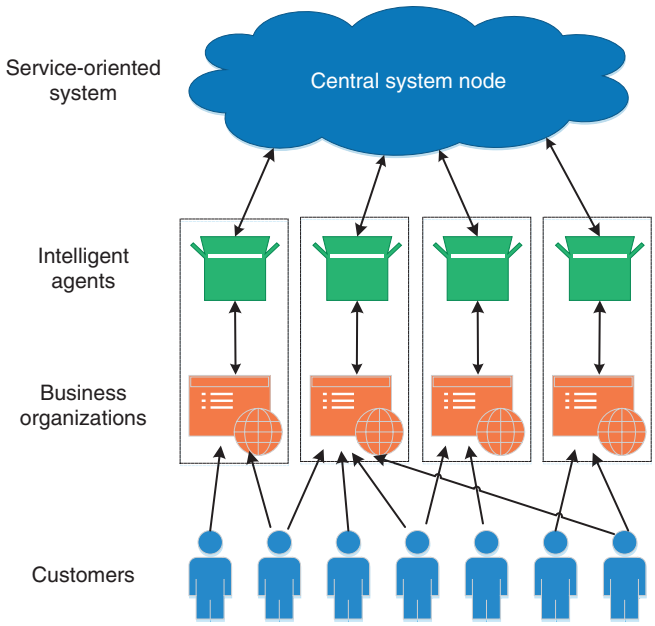
The proposed solution takes the service-oriented architecture described by Alexandrescu et al. (2016) and the recommendation idea mentioned in the single-access system proposed by Puiu and Alexandrescu (2016), and provides a novel and more efficient recommendation method that is tailored to the user and also benefits the business organizations.

Figure 2 shows the architecture of the system and the four actors and components that form the proposed solution. The main roles of each of them are as follows:

The central system node

—coordinates the entire information flow in the system;
—allows for online business to adhere to the system;
—creates intelligent agents and sends them to the online business;
—facilitates communication between the agents.

Figure 2. Architecture of the Used System



Note. Each business organization has associated an intelligent agent that handles the communication between the organization and the business node and that makes local recommendations.

The intelligent agent
—interacts with the business organization where it resides;
—monitors product and service purchases;
—makes local recommendations;
—forwards recommendations received from the central system node.

The business organization
—represents an existing online business;
—provides the agent with limited access to the purchases made by their customers;
—receives from the agent recommendations that are forwarded to the customer.

The customer
—purchases products and services from the business organization;
—receives recommendations for products sold by the businesses in the system.

The novelty of the proposed solution consists in two parts: first, the *system description* with the central node and the intelligent agents at each of the business organizations, and, second, the *recommendation module* logic used by the intelligent agents.

5. System Description

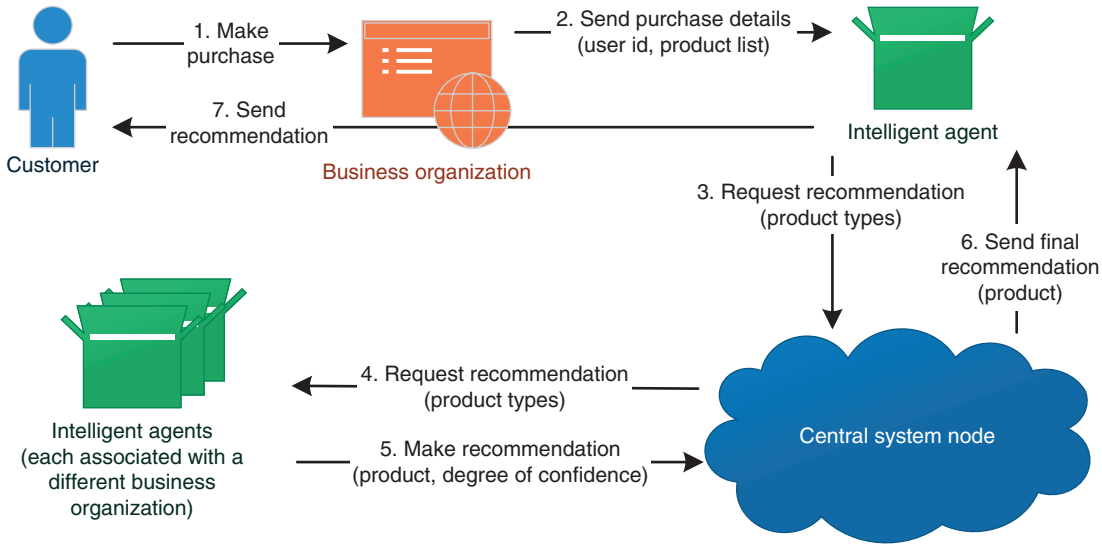
5.1. General Information Flow

The coordinator of the system information flow is the central system node. This node allows businesses to be part of the system by providing an intelligent agent when a business registers, which acts as a middleman between the business organization and the system. The first role of the central system node is to create an agent for each business in the system and, afterward, to facilitate communication between the agents; the whole system can be seen as lying on top of a distributed mobile agent platform.

Figure 3 presents the communication logic between the system actors and components required in order to recommend a product.

Each time a purchase is made, the business organization sends the details to the agent who processes the purchase information and makes a request for a recommendation to the central system node. The central node forwards the request to the other agents and each of them make a product recommendation with a certain degree of trust. The central node selects one of the product and sends the details to the agent that requested the recommendation. Lastly, the agent forwards the product info to the business organization, who sends the recommendation to the customer. The details referring to the product recommendation logic and the degree of trust are discussed in Section 6.

Figure 3. Proposed Recommendation System



Note. A scenario in which a customer purchases some products and the system makes a recommendation for another relevant product or service.

5.2. Design Decisions and Discussion

5.2.1. Business Organization—Intelligent Agent Communication. For the communication between the business organization and the agent to take place, the organization must call an agent application programming interface method each time a purchase is made and it also must allow asynchronous receiving of product recommendations from the agent. The presence of the agent on the business organization's node allows for a clear separation of logic from the central system node and it also permits dynamical changes to the system's structure. The information being sent from the business to the agent does not contain any user information.

5.2.2. Customer Interaction with the System. Each time a *transaction* (i.e., the customer purchases one or more products at once) is made, each sold product details and the product type are sent to the agent's recommendation module. The agent requests a recommendation pertaining to the purchase from the central system node. Sending the recommendation to the customer is the responsibility of the business organization because it alone has knowledge of the pertinent user details such as the email address.

After the user sees the recommendation, his feedback or lack thereof is sent to the agent's recommendation module. If the user privacy policy allows the business organization to share customer details such as name and email address with third parties, then the intelligent agent can communicate directly with the customer, thus reducing the communication overhead.

In a simple use-case scenario, the business organization or the agent sends an email with the recommendation and also asks the user to give a score from 1 to 10 signifying the *recommendation relevance* (ρ) (1 being irrelevant and 10 being most relevant). If the customer does not offer any feedback, then the relevance is deduced based on his two possible actions. If the user clicks on the link to the recommended product then he is probably interested in it, therefore, the relevance is considered to be 8. On the other hand, if the user ignores or does not even notice the email, then the relevance has a value of 3. The main issue here is that there is no sure method of distinguishing between the case were the customer has seen the email and ignored it, and the case where the user deleted the email without even reading it; this is why the relevance chosen in this case has a fairly safe value of 3. The recommendation relevance value is used by the intelligent agent in order to offer more relevant recommendations by updating a product type association weighted graph.

5.2.3. Central System Node Role. Besides allowing business organizations to be part of the system, the central system node (CSN) has two other roles: to handle the communication between the intelligent agents and to make the final recommendation.

When a purchase is made, the intelligent agent, which communicates with that business organization, performs a request to the CSN for a recommendation. It sends to the CSN the purchased product types and, in turn, the CSN sends that information to the other agents. Each intelligent agent responds with its recommendation and its degree of confidence (which are described in Section 6) and the CSN uses a selection method to pick the final product or service, which is then sent to the agent that requested the recommendation. The simplest selection method is to randomly pick a product or service so that all the businesses have an equal chance of having their products chosen. Other selection methods and their impact on the system are discussed in Section 7.

6. Recommendation Module

The recommendation module receives from the central system node a request for a recommendation regarding a specific list of product types. To obtain a simple but effective solution, the product details are not processed by the system, but instead associations are formed between the different product types. Therefore, the proposed solution determines what other product types are the customers more likely to buy and then recommends a product from one of those categories.

Each intelligent agent makes that recommendation by means of a selection algorithm applied on a *product type association weighted graph*, which is dynamically updated based on the users' purchases. Using the product type association weighted graph allows the agent to keep track of which product types go together in order to make more efficient recommendations. The vertices of the graph represent the product types and the edges represent a relation between those types. Each edge has an associated weight that signifies the probability of recommendation for a product of that type. Also, each vertex has a *payload* that consists of a list of the previously purchased products and the respective purchased quantities.

With each transaction, the agent receives a list of purchased products and, afterward, the relevance value obtained after recommending a product to that customer; both are used to update the product type association weighted graph.

Once the transaction information is received, the association graph is updated as follows:

—The payload of each vertex (i.e., product type) from the transaction is updated by adding a new product or, if the product already exists in the list, by incrementing the purchased quantity.

—Each weight associated with an edge between two product types belonging to a transaction gets incremented (if the product types are not in the graph then they are added, and if there was no edge between them, then one is created with a weight value of 1).

The module makes a recommendation by first selecting a product type from the ones associated with the transaction. The chosen selection method is roulette wheel selection (Pandey et al. 2016), which is typically used by genetic algorithms because it allows product types with a stronger associations to have a higher probability of being selected. Roulette wheel selection is applied on the weights of the graph edges connected to the vertices affected by the transaction, which results in selecting a product type. Afterward, the same selection method is applied again on the product list (from the vertex payload) based on the quantities sold for each product. Basically, if another customer previously purchased a product belonging to the same product type acquired by the current customer, then there is a high probability that additional products purchased by that other customer in the same transaction to be recommended (especially if that other product was previously purchased by more customers).

After the selection is determined, the intelligent agent computes the *degree of confidence* (δ) in that selection, which is defined as the average weight for the valid edges between the product types belonging to the transaction and the selected product type. Then, the product selection along with the degree of confidence are sent to the central system node.

Let $T = \{x_k: 1 < k < n\}$, where x_k a purchased product type belonging to the latest transaction and n is the number of purchased products, and let w_{ij} the weight between vertices i and j, then the probability of selecting another product type i using the roulette wheel selection method is

$$p_i = \frac{\sum_{k \in T} w_{ik}}{\sum_{j \notin T} \sum_{k \in T} w_{jk}}.$$

The degree of confidence in the selected recommendation is

$$\delta_i = \frac{1}{n} \cdot \sum_{j \in T} w_{ij}.$$

Figure 4 shows an example of a product type association weighted graph, where the vertices 1, 2, and 5 represent the product types belonging to the transaction for which a recommendation has to be made. The probability of selecting the three vertices (3, 4, and 6) that do not belong to the transaction is

$$p_3 = \tfrac{1}{9}, \quad p_4 = \tfrac{6}{9}, \quad p_6 = \tfrac{2}{9}.$$

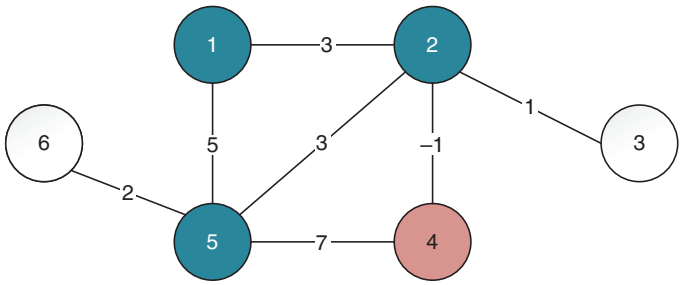
Note that if there is no edge between two vertices the default weight is 0.

The most likely product type to be selected by the roulette wheel selection is 4 because it has the highest probability. In this case, the degree of confidence in the recommendation is $\delta = 6/2$ because there are only two edges between vertex 4 and the vertices belonging to the transaction.

When an intelligent agent receives the relevance value for one of its recommendations, the agent updates the product type association weighted graph, by using a technique inspired from the backpropagation learning method used in neural networks, as follows:

—Update the edges between the selected product type and the product types belonging to the transaction by adding five minus the recommendation relevance value. This way a positive recommendation (more than five)

Figure 4. Example of a Product Type Association Weighted Graph



Note. The vertices 1, 2, and 5 represent the products belonging to a transaction and vertex 4 represents the chosen recommendation.

will strengthen the edge weight, while a negative recommendation (less than five) will make that association between product types less relevant.

—The same update, for similar reasons, is made to the quantity value associated with the product that was recommended.

Basically, the relevance value artificially adds or removes purchases so that the user's feedback, not just the purchased products, has relevance in making the recommendation.

Given the fact that the business organizations sell different types of products, there will most likely be situations in which an intelligent agent will have to recommend a product based on product types that are not in the association graph. In this case, the agent will consider all the product types when making the selection and the selected product will have a default degree of confidence of one. The association graph can have other product types, which are not sold by that business organization, by means of the recommendation relevance value obtained from the user. When a relevance value is obtained for a product that has no association with the transaction product types, then a vertex is added for each of those types and the edges to the recommended product type are set to a value of five minus the relevance value.

The restriction that, each time a transaction is made, the customer receives a single recommendation was imposed in order to simplify and to better exemplify the communication flow and the recommendation logic. The system can easily be extended to allow multiple recommendations to be made for each purchase, or to periodically send the customer different recommendations.

7. Performance Evaluation

7.1. Measuring the Efficiency

The efficiency of the proposed method depends on the customers' different tastes and preferences, and on the quantities and prices of the products that are being sold by the businesses in the system.

Regarding the customer behavior, for a specific purchased product, the same recommendation can be useful for a customer and ignored by another. Ideally, the recommendations must be tailored to each customer as is the case with the traditional collaborative filtering approach, but this method is only applicable on each individual business organization because of user privacy. There needs to be a way of linking the user accounts of the same customer on the different business nodes in the system without actually sharing sensitive information between the node. The recommendation module presented in this paper considers the preferences of all the users. As future research, the authors will consider expanding the recommendation module by adapting the collaborative filtering technique, which takes into account each individual user's preferences.

Considering the fact that the recommendation modules establish relationships between product types, a method to determine the relevance of a recommendation is to have a directed graph in which the vertices are the products and the edges represent the probability of the customer to be interested in a specific product after a specific purchase. This graph has to be created beforehand, and, each time a product X is purchased and a product Y is recommended, the *efficiency* of the solution is measured as the probability associated with the edge from vertex X to vertex Y. The efficiency (i.e., quality of the recommendation) is higher when the associated probability has a value closer to 1.

The simplest form of evaluating the performance of the proposed system is to compare the case in which, each time a purchase is made, a random product is recommended from a random business organization in the system with the case in which the proposed recommendation method is used.

An important issue with performing this comparison is the fact that the system has to be populated with business organizations and customers, and those customers must make purchases. Initially, the quick solution is to create a simulator in which mockup customers purchase mockup products and the system recommends other products. Intuitively, the efficiency should increase as more recommendations are made and the numbers of customer feedback increases.

However, determining the exact efficiency of the proposed recommendation module is beyond the scope of this paper and this subject will be further researched and discussed.

7.2. Final Recommendation Selection Method

The central system node has to select a product or service among the recommendations received from the intelligent agents. The used selection method is pure random selection in order to have a uniform distribution of recommendations, so that no business in the system has its products recommended more often than others. This can also be in the detriment of businesses that consistently sell many products because their products will get recommended as much as the items of businesses that sell fewer products. There is also an advantage of this approach, smaller businesses will have their products recommended more often thus increasing their revenue

and becoming more competitive. In a way, the playing field becomes somewhat leveled because large businesses help small businesses to thrive; however, this might not be acceptable for the large businesses. For now, the focus is on associating product types based on the user purchasing patterns than on making it fair for the businesses in the system. There are methods of making the system more equitable by making recommendations based on how much each business pulls its own weight (e.g., quantities sold, commercialized product types, positive feedback regarding the recommendations) but this is not the scope of this research.

The efficiency of the final selection depends significantly on the types of businesses in the system. Nonetheless, the proposed solution becomes highly effective by using a module that monitors the system for a period of time and makes constant adjustments while taking into consideration the aforementioned recommendation aspects regarding fairness toward the businesses.

8. Use-Case Scenario and Improvements

To better exemplify the efficiency of the proposed solution, a use-case scenario is considered in which there are the following online businesses:

—a bookstore, which deals with travel guides, cookbooks, and romance novels;
—a tourism agency, which sells vacations and rents cars;
—an appliance store, which offers food processors, blenders, and cookbooks;
—a retailer, which sells external hard drives, USB cables, cookbooks, travel guides, and vacations.

Considering the state of the system at a point in time (Figure 5), the goal is to determine what kind of product is going to be recommended when a user makes a purchase and how new vertices are added and how the edges are updated in the product type association weighted graph.

Step 1. A customer buys a vacation from the tourism agency. The association graph suffers no initial change because there is only a single item in the transaction.

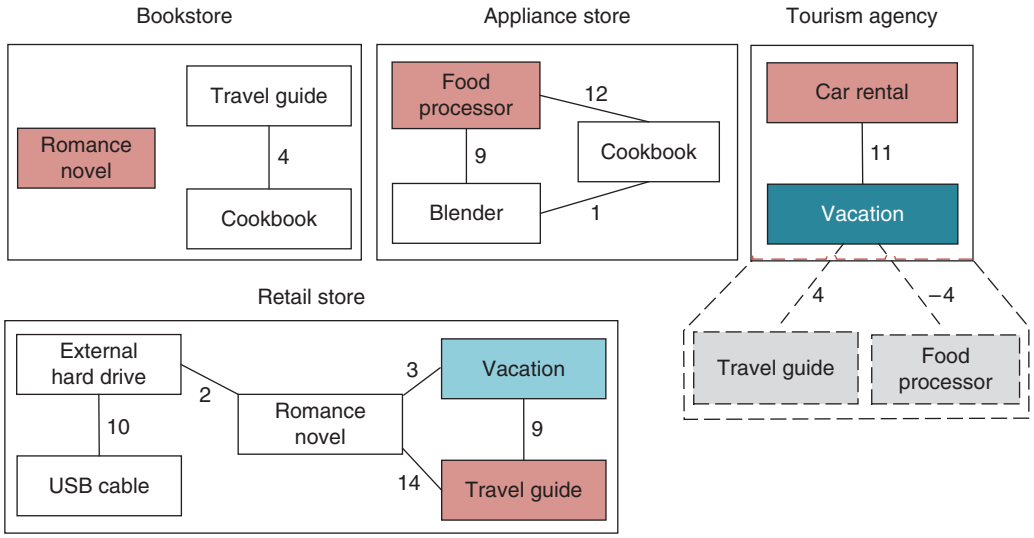
Step 2. Each of the intelligent agents makes a recommendation and sends the degree of confidence in their selection. The transaction consists of purchasing a single product/service, therefore, the degree of confidence (δ) is the weight of the edge between the purchased and the recommended product types. The recommendations made by each business are as follows:

—the tourism agency recommends a car rental service, with $\delta = 11$;

—the bookstore and the appliance store have no vacation product type and they each recommend a random product from the available product types (e.g., a romance novel and a food processor, respectively), with $\delta = 1$ (the default value);

—the retail store can recommend either a romance novel (weight 3) or a travel guide (weight 5); in this example, the roulette wheel selection method is more likely to recommend a travel guide, with $\delta = 5$.

Figure 5. Use-Case Scenario for the Proposed Recommendation System



Notes. A vacation is purchased from a tourism agency and each business recommends the product types with red background (i.e., romance novel, food processor, car rental, and travel guide). The dashed lines represent the updates made in the case of the travel guide being recommended by the bookstore and a food processor by the appliance store. The edge values are obtained from recommendation relevance value given by the customer.

Step 3. The central system node then randomly chooses one of the four selections and recommends to the customer a travel guide from the retail store. It would help in this situation if the recommendation process takes into account a localization element, that is, a customer would be more interested in a travel guide pertaining to his vacation destination. However, this would require an extra interpretation of the product details and in many cases this information is not available.

Step 4. In the user feedback phase, the customer gives a recommendation relevance value of 9 ($\rho_1 = 9$), which means that the travel guide was a product suggestion relevant to his needs. This translates into adding a vertex with the travel guide product type in the association graph from the tourism agency. The edge between the travel guide and the vacation vertices will have a value of $\rho_1 - 5 = 4$. This means that if a travel guide is bought from the bookstore then there is a high probability of a vacation being recommended by the tourism agency.

If the central system node recommended a food processor, then the customer would likely give a low relevance value (e.g., $\rho_2 = 1$), which would result in assigning the tourism agency an edge weight of $\rho_2 - 5 = -4$, between the vacation and the food processor vertices. In this case, when a customer buys a food processor, there is no chance for the tourism agency to recommend a vacation.

Because the association graphs update continuously there can be a situation in which two businesses become isolated from one another, that is, the products of one of the businesses can never be recommended to the other business. This can become the case with the appliance store and the tourism agency. The recommendation selection method can be modified to allow with a very small probability for a product to be recommended even if the association edge has a negative weight.

In the aforementioned use-case scenario, the retail store has a positive edge weight of 2 between the external hard drive and the romance novel product types, because two customers purchased an item from each of the two categories in the same transaction. This is a temporary edge state, and even if the same kind of transaction happens again it will quickly reach a negative value or at least a low positive value based on user feedback.

9. Conclusions and Future Work

The proposed system for recommending products and services uses techniques inspired from ad networks and collaborative filtering when suggesting different products based on different criteria; the roulette wheel selection method used in genetic algorithms for selecting a recommendation; and the backpropagation technique used in neural networks for allowing the user feedback to have an important role in deciding which product type to recommend. It uses a product type association weighted graph, which is dynamically updated based on user purchasing patterns and user feedback in order to offer the customer a recommendation for products and services he is more likely to need.

There are two main advantages of the proposed solution:

—The purchasing patterns are determined by the agent located at the online business and no information that would violate the user's privacy is shared with the system.

—Each intelligent agent can make recommendations based on a product type association weighted graph that is dynamically updated each time a purchase is made and each time the user offers feedback.

In the current state, the system acknowledges the customer's privacy by not sharing user information among the businesses in the system (not even the intelligent agent that communicates with the business organization knows the user details) and by looking at the purchased products one transaction at a time.

As future research, in order to increase efficiency, it is interesting to process the purchasing patterns of a customer over all his transactions, but this would imply an extra complexity layer in terms of the association graph. Another direction is to make a global profile for each customer, but this would require the users' permission to share its data among the businesses in the system; in this case, classical collaborative filtering techniques can be applied and improved upon. The next step of the research presented in this paper is to make an environment with several businesses each with their own product taxonomy, to simulate a high number of purchases, and to research a mechanism for evaluating various recommendation methods in different scenarios.

References

Alexandrescu A, Butincu CN, Craus M (2016) Improving after-sales services using mobile agents in a service-oriented architecture. Borangiu T, Dragoicea M, Nóvoa H, eds. *Exploring Services Science*. Lecture Notes in Business Information Processing, Vol. 247 (Springer International Publishing, Cham, Switzerland), 444–456.

Boniface M, Nasser B, Papay J, Phillips SC, Servin A, Yang X, Kousiouris G (2010) Platform-as-a-service architecture for real-time quality of service management in clouds. *Internet Web Appl. Services* (*ICIW*), *2010 Fifth Internat. Conf.* (IEEE, Barcelona, Spain), 155–160.

Castelluccia C, Kaafar MA, Tran MD (2012) Betrayed by your ads! Reconstructing user profiles from targeted ads. Fischer-Hübner S, Wright M, eds. *Proc. 12th Internat. Conf. Privacy Enhancing Tech.* (Springer-Verlag, Berlin), 1–17.

Dave V, Guha S, Zhang Y (2012) Measuring and fingerprinting click-spam in ad networks. *ACM SIGCOMM Comput. Comm. Rev.* 42(4): 175–186.

Davis H (2006) *Google Advertising Tools: Cashing in with AdSense, AdWords, and the Google APIs* (O'Reilly Media, Sebastopol, CA).
Demirkan H, Kauffman RJ, Vayghan JA, Fill HG, Karagiannis D, Maglio PP (2009) Service-oriented technology and management: Perspectives on research and practice for the coming decade. *Electronic Commerce Res. Appl.* 7(4):356–376.
Hostler RE, Yoon VY, Guimaraes T (2012) Recommendation agent impact on consumer online shopping: The movie magic case study. *Expert Systems Appl.* 39(3):2989–2999.
Legnani E, Cavalieri S, Ierace S (2009) A framework for the configuration of after-sales service processes. *Production Planning Control* 20(2): 113–124.
Lin FR, Maglio PP, Shaw MJ (2014) Introduction to service science, management, and engineering (SSME) minitrack. *System Sci.* (*HICSS*), *2014 47th Hawaii Internat. Conf.* (IEEE, Waikoloa, HI), 1240–1240.
Maglio PP, Spohrer J (2008) Fundamentals of service science. *J. Acad. Marketing Sci.* 36(1):18–20.
Mei T, Zhang R, Hua XS (2011) Internet multimedia advertising: Techniques and technologies. *Proc. 19th ACM Internat. Conf. Multimedia* (ACM, New York), 627–628.
Mookerjee R, Kumar S, Mookerjee VS (2012) To show or not show: Using user profiling to manage Internet advertisement campaigns at Chitika. *Interfaces* 42(5):449–464.
Onofrei G, Archip A (2016) Achieving better recommendations with overclassification: Practical considerations. *20th Internat. Conf. System Theory, Control Comput.* (IEEE, Sinaia, Romania), 410–416.
Pandey HM, Shukla A, Chaudhary A, Mehrotra D (2016) Evaluation of genetic algorithm's selection methods. Satapathy S, Mandal J, Udgata S, Bheteja V, eds. *Information Systems Design and Intelligent Applications*. Advances in Intelligent Systems and Computing, Vol. 434 (Springer, New Delhi, India), 731–738.
Prahalad CK, Ramaswamy V (2004) *The Future of Competition* (Harvard Business School Press, Boston).
Prahalad CK, Ramaswamy V (2013) *The Future of Competition: Co-Creating Unique Value with Customers* (Harvard Business Press, Boston).
Puiu VG, Alexandrescu A (2016) An event-driven service-oriented architecture for performing actions on business organization items. Borangiu T, Dragoicea M, Nóvoa H, eds. *Exploring Services Science*. Lecture Notes in Business Information Processing, Vol. 247 (Springer International Publishing, Cham, Switzerland), 432–443.
Spohrer J, Maglio PP (2008) The emergence of service science: Toward systematic service innovations to accelerate co-creation of value. *Production Oper. Management* 17(3):238–246.
Sundaram D, Zhou W, Piramuthu S, Pienaar S (2010) Knowledge-based RFID enabled web service architecture for supply chain management. *Expert Systems Appl.* 37(12):7937–7946.
Zhang L, Guan Y (2008) Detecting click fraud in pay-per-click streams of online advertising networks. *Distributed Comput. Systems, 2008* (*ICDCS'08*), *28th Internat. Conf.* (IEEE, Beijing), 77–84.

information

MDPI

Article

Optimization and Security in Information Retrieval, Extraction, Processing, and Presentation on a Cloud Platform

Adrian Alexandrescu

Department of Computer Science and Engineering, Faculty of Automatic Control and Computer Engineering, "Gheorghe Asachi" Technical University of Iași, Iași 700050, Romania; aalexandrescu@tuiasi.ro

Received: 16 April 2019; Accepted: 4 June 2019; Published: 5 June 2019

Abstract: This paper presents the processing steps needed in order to have a fully functional vertical search engine. Four actions are identified (i.e., retrieval, extraction, presentation, and delivery) and are required to crawl websites, get the product information from the retrieved webpages, process that data, and offer the end-user the possibility of looking for various products. The whole application flow is focused on low resource usage, and especially on the delivery action, which consists of a web application that uses cloud resources and is optimized for cost efficiency. Novel methods for representing the crawl and extraction template, for product index optimizations, and for deploying and storing data in the cloud database are identified and explained. In addition, key aspects are discussed regarding ethics and security in the proposed solution. A practical use-case scenario is also presented, where products are extracted from seven online board and card game retailers. Finally, the potential of the proposed solution is discussed in terms of researching new methods for improving various aspects of the proposed solution in order to increase cost efficiency and scalability.

Keywords: optimization; security; cost efficiency; information retrieval; information extraction; crawler; cloud services; vertical search engine; e-shopping; distributed systems

1. Introduction

The Internet contains a huge amount of ever-growing data that end-users must sift through in order to obtain the information that is relevant to them. When a user wants to find a specific product for online purchase, that person usually uses a general search engine or a specialized shopping search engine like Google Shopping, PriceGrabber, or Shopzilla [1]. Two of the first problems that occur when developing a shopping search engine is where you find the product information and how you retrieve that data. If those problems are solved, the next step is to use information retrieval techniques, extract the data, then perform a reverse indexing to obtain, for each word in the product names, a list of products that contain that word. Finally, the end result is to develop a search engine user interface so that one can search for a product by name. More complex shopping search engines allow searching depending on the features of the product (e.g., manufacturer name, dimensions, color), not just the product name.

Existing relevant research in this domain refer to vertical search engines [2,3], which are focused on a specific topic and generally use a targeted crawler, as opposed to crawling the "entire" web. One of the most common verticals is shopping, but we can go even further and consider the vertical to be shopping for a specific type of product (e.g., smartphones, power tools, clothes). When developing a shopping search engine, it helps to start with a specific type of product and then add more verticals to see if the provided solution is as efficient.

In terms of information retrieval, there are web crawlers and web scrapers [4]. A web crawler is an application that accesses webpages, extracts the URLs, and then repeats this process for each page

denoted by the extracted URLs. The process can be viewed as a breadth first traversal in a directed graph where each vertex represents a webpage and each edge represents a link from one page to another. On the other hand, a web scraper extracts data from webpages by identifying the useful information and then storing it in a database. The method of identifying the required information can be as trivial as specifying the HTML path to the data beforehand, or it can employ data mining techniques to ensure a fully automated retrieval process [5]. Crawling implies some scraping in the sense that the links have to be extracted, and scraping implies some crawling because, usually, the data is gathered from multiple webpages and the scraper needs to access those pages' content.

Finding the URLs that contain product data and the location of the product features on a webpage is very difficult due to fact that websites are very different in presenting their information. Usually, the research regarding this topic is proprietary and existing shopping search engines keep secret their techniques of extracting data. There is some public research in terms of structured data extraction [6,7], but the solutions are very general and their efficiency is questionable when it comes to product information extraction.

Due to the fact that the research presented in this paper follows a complex flow of information that passes through clearly distinct components, more related work is referred to in the sections presenting the proposed solution.

The main aim of the research is to propose a solution for having a vertical shopping search engine, from retrieving the product information from online retailers to offering an interface where the user can search for the desired products and can also view the price history of each product. This research is a continuation of the work from [8], where a framework for information retrieval, processing, and the presentation of data is presented. The main differences are streamlining the information flow between components, a better modularization, and especially, the multiple optimizations that were made to lower resource consumption and therefore lower the running costs. On top of this, a security layer was added to make the crawler run smoothly and to prevent third parties from exploiting the proposed system.

In the current paper, the proposed crawling and extraction components are a combination between traditional web crawling and web scraping, i.e., the proposed crawler is bound to specific websites, but it also finds new pages on those websites and retrieves them for extraction. Moreover, at this point, the crawling and extraction template is manually set for each website, and it is used to obtain the product features. In terms of the communication between the main components, this is achieved by means of web services, which are a very good method for sending data in heterogeneous environments because of the little volume of overhead information that is being sent, while having a flexible information transfer protocol structure [9].

Throughout the paper, the challenges that the considered environment entails, the chosen solutions, and possible improvements are identified. The main optimizations are: a novel method of representing the crawl and extraction configuration/template, several steps to lower cloud resource usage by keeping a separate product index, grouping multiple products in the same database entity, using a memory cache layer, and other methods for keeping the database reads and writes to a minimum while taking into account the limitations imposed by the cloud platform.

2. Proposed Solution

2.1. General Overview

The goal is to gather specific information and to make it easily accessible to the user in a cost-effective and secure manner. Therefore, this paper proposes a solution for information retrieval, extraction, processing, and the presentation of data that is optimized for efficiency and cost reduction. Throughout this paper, the specific information pertains to product details gathered from online retailers, and the whole process flow is essentially everything needed for a fully functional shopping search engine. On the other hand, the proposed solution is designed to be generic, which means

that it can be used to obtain different types of structured data from webpages, and not just product information. For example, it can be used for finding medication based on symptoms, recipes based on ingredients, or news articles referring to specific topics. A significant improvement compared to a traditional search engine is the ability to search for targeted information based on clearly defined information features. The main challenges of a targeted search engine are how to automatically find specific data in a highly unstructured environment, content-wise (i.e., a webpage), and how to efficiently process and store that data in order to make it accessible to the user.

Tackling all the possible scenarios and solutions in detail is an impossible task; therefore, several limitations are imposed on the considered environment. As previously mentioned, the focus is on extracting product information, but the location of the product features within the webpage is given for each website in the form of a configuration file. The method of obtaining that configuration file is beyond the scope of this paper, and it will be researched and properly tested by means of the proposed solution. Moreover, the novel method of increasing cost effectiveness when storing data on a cloud platform is better tailored to finding products rather than searching for other types of information.

To summarize, the considered scenario details and limitations are as follows:

1. The user can find product information pertaining to a single specific vertical.
2. The overall number of crawled websites is rather small and known beforehand.
3. The location in the webpage of the items that have to be extracted is determined manually for each website and is provided to the system in a configuration file.
4. The search engine must be publicly available to the end-user.

The current state of the proposed solution has the aforementioned limitations, but throughout this paper, various solutions to improve the system and to make it scalable are presented.

2.2. System Architecture

As mentioned previously, the research presented in this paper is based on the distributed framework from [8], where a highly modularized system architecture is presented without going into detail in regards to each component and without any optimizations or security considerations. Compared with the framework that was presented there, in this paper, the proposed solution streamlines the previous version and describes in detail the implementation specifics that are designed for cost effectiveness when deployed on a cloud platform.

An overview of the proposed system architecture is presented in Figure 1. The information flow is as follows: at a specified period in time (e.g., daily), a dispatcher starts a crawler for each website, and an extractor obtains the product information, which is then processed and deployed on a cloud web application in order to be accessed by the user.

The proposed solution can be summarized by four actions: retrieve, extract, process, and present. Each action represents a stage in the information flow, and each one can be easily replaced with a different implementation. For many of the identified challenges, the chosen solutions are basic in the sense that they represent the path of least resistance to obtain the goal of having a cost-effective targeted shopping search engine hosted on a cloud platform. In addition, the retrieval, extraction, and processing actions are performed on a local server, while the presentation action is performed on a cloud platform. This decision was taken due to the fact that running computing-intensive tasks on the cloud incurs a higher cost, rather than running them locally and just uploading the results to the cloud platform. Of course, this is feasible only in certain conditions, which are discussed later in this paper.

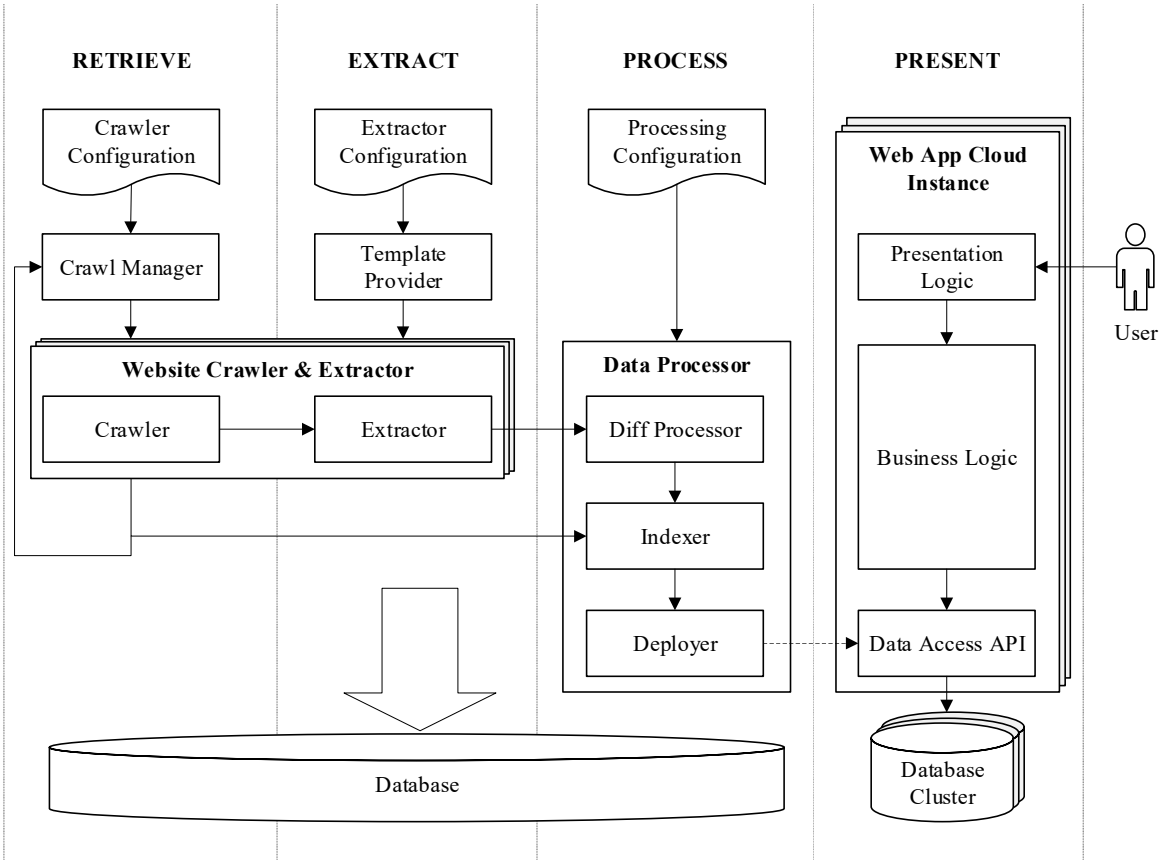


Figure 1. Overview of the proposed solution architecture and the information flow between the main components. For the first three actions (i.e., retrieve, extract, process) the components are on a local server and share the same database. For the last action (i.e., present), the components are deployed on a cloud platform.

2.3. Retrieve Action

The crawler needs a seed list of websites to crawl first and from which to extract all the links, which are added to the database and subsequently crawled. In theory, with the proper seed list, the crawler can go through all of the surface web; the deep web, for the purposes of this paper and product extraction, does not contain relevant product information [10]. For a vertical search engine, crawling most of the web is not an efficient solution because only an incredibly small percentage of the websites contain information relevant to that vertical. Instead of hectically scanning the web, the chosen trivial solution is to have a seed list with all the considered websites from that vertical and to process only those websites. In this situation, we have a crawler in the sense that it finds new webpages, but these are only from the initial website pool.

The retrieval process is coordinated by the crawl manager component, which has three subcomponents: the initializer, the scheduler, and the dispatcher. A configuration file is used to obtain the seed list and the crawling parameters; the former is used by the initializer to populate the database, while the latter are used by the scheduler. Keeping in mind that the goal is to have a cost-effective solution, the crawler will only access, if possible, pages that contain multiple product details. The idea is to crawl as few pages as necessary to extract the required information.

2.4. Proposed Method for Representing the Crawler Configuration

There are two parts to the crawler configuration. Firstly, the global crawl parameters that specify how the webpages are retrieved, and secondly, the site-specific parameters that describe what sites and what pages are going to be retrieved.

In terms of the global crawl parameters, these are:

- *database connection string*—host, port, username, password, database name, and type;
- *minimum and maximum waiting times* between crawls from the same site—a random value between that interval is chosen when retrieving each page;
- *recrawl interval/time*—the recrawl takes place after a specified time from the last recrawl or at a specified time each day;
- *web service URL and credentials*—used to upload the site and product information;
- *a list of site-specific parameters.*

For each site, the proposed system receives a list of parameters that are used by both the crawler, which needs to know what pages must be retrieved, and the extractor, which requires the location on the webpage of the product information that must be extracted. Therefore, the list of site-specific parameters is as follows:

- *name, url, logoUrl*—the site name, URL, and logo URL, which are needed by the presentation system component; the *url* property also serves as unique key for the purpose of determining if the site already exists in the database;
- *fetchUrlRegEx*—regular expression used to determine if a URL is added to the database for retrieval;
- *seedList*—list of webpages used to start the crawling process on that website;
- *extractor*—an object describing the extraction parameters:
    - *extractUrlRegEx*—a regular expression denoting which pages contain extractable data;
    - *baseSelector*—a custom selector describing the in-page location of the information that needs to be extracted;
    - *properties*—an object with key-value pairs, where the key is the product feature name, and the value is a custom selector relative to the base selector and contains the feature value that is to be extracted;
    - *uniqueKeyProperty*—property name from the properties object that serves as a unique key for the purpose of uniquely identifying a product belonging to a website.

The advantage of this representation is the flexibility that allows new websites to be easily added and allows different types of products to have completely different properties. Moreover, the uniqueKeyProperty offers the possibility of specifying the criteria for determining if a product is the same as one from the previous crawl (it can be the URL for some websites, or a product number for others).

2.5. Extract Action

After each a page is retrieved, for the sake of cost-effectiveness, it is sent to the extractor component, which obtains the product information. Another approach would have been to completely separate the crawler and the extractor, i.e., the crawler stores the page in the database, and a separate extractor process retrieves the page from the database and obtains the product data. The former solution was chosen to reduce the number of calls to the database and the volume size of the stored information at the expense of perhaps having a less scalable system. More details about this choice are presented in the Discussion section of this paper.

Extracting the information implies the knowledge of where the product features are on the page, i.e., a page template. At this point, the template is set manually for each website and is given as input to the extractor component in the form of a configuration file. A continuation of the research presented in this paper is to study the possibility of determining the template automatically for each website. There are multiple approaches to this problem, as shown in the review in [6].

Depending on what information is to be extracted, each website is analyzed so that all the products are obtained with the minimum number of webpages being retrieved. In accordance, multi-product

pages are favored as long as they contain all the required product features. In a majority of situations, only the product name, price, and availability are shown on those webpages, and for the extra features, the crawler needs to visit the single-product pages. The same product on different retail websites should have the same characteristics, which can be taken from the manufacturer website or from just one of the retailers. This means that, from the rest of retailers, we need only a means of getting the price and availability, after identifying that it is the same product. Determining that two products from different websites are the same is a difficult problem that will be researched as a continuation of the work from this paper.

2.6. Proposed Method for Representing the Extractor Template

The location of each product/item feature on the webpage varies on each website, and once it is determined, the extractor must be able to obtain the required data from the webpages that are identified as containing product information. An important challenge is how to represent the location of each product feature so that it can be quickly extracted from the webpage. This translates into reverse engineering the webpage template, or at least the part of the webpage that contains product information. Many existing crawler solutions extract data by means of CSS-like selectors or by using XPath, and then use library-specific methods to extract text or attributes from the selected HTML elements. Representing the product extraction template using only those methods is a problem that can be solved by adding extra functionality to the selector.

The new method for extracting data presented in this paper extends upon the JSOUP library [11], which allows the selection of HTML elements, and improves it by allowing the selection of attributes, text, and a Boolean value if a certain condition is met. A CSS selector is designed to identify only HTML elements; so in order to extract information, if the selector identifies an element, then all the text from that element (including all the text from its children) is returned by the proposed extractor. Besides all the CSS selector features, new symbols are added to allow more information to be extracted, as shown in Table 1.

Table 1. The custom selector symbols that were added and are used by the product information extractor.

Symbol	Usage	Description
/ (slash)	selector/selector	Separates multiple selectors. The value that is returned is the one found by the first matching selector in the list. If the custom selector ends with a slash and no element was selected, then an empty string is returned instead of an exception being thrown.
% (percent)	selector%attr selector%	Returns the value for the attribute name following the symbol from the element selected by the string preceding the symbol. If there is no expression after the symbol, then it returns the element text not including the text from that node's children elements.
= (equals)	selector%attr=expr selector%=expr	Returns a Boolean value depending on whether the left-hand value obtained from the selector matches the right-hand expression.
~ (tilde)	selector%attr=~expr	Returns a Boolean by negating a right-hand expression when used with the equals sign.
abs: (abs colon)	selector%abs:attr	Returns the absolute URL for an attribute value; can only be used after the percent symbol (%).

Examples of custom selectors:

- `td.productlisting_price>.productSpecialPrice/td.productlisting_price>.price` returns the text from the element that has the *productSpecial* class attribute, which has a *<td>* parent element with class *productlisting_price*; if there is no such element, then it returns the text from the element with class *price,* which has the same parent.
- `div.product-details>p.price%` returns the text from element *<p>* with class *price* (not including the text from the child elements), which has a * * parent with class *product-details.*

- `div.image>img%class=~outofstock` returns true if it is not *outofstock*, the value of the <*img*> element's *class* attribute, which is a child of a <*div*> element with the class *image*.
- `div.name>a%abs:href` returns the absolute URL from the value of the *href* attribute of an <*a*> element having a parent of type <*div*> with class *name*.

The proposed approach to representing the extraction template is a flexible way of specifying the location of each product feature in a webpage. A direction for future research regarding this topic can be to determine a method of automatically obtaining the template, maybe using unsupervised learning techniques.

2.7. Process Action

The data processor determines what information will be deployed to the presentation layer (i.e., the remote web server). There are two important types of information that are sent: the product data and the product index. Again, the goal is to minimize resource consumption at the presentation layer. This is why each extracted product data is sent to the diff processor, which determines if there are any changes compared to the previous crawl. Only the products that have different information are sent to the remote server. In order to minimize calls to the remote server's data access API, the products are sent in batches of 50 and each product in a batch is from the same website, as this can be used to optimize the data storing process at the server.

An important issue is how the presentation layer distinguishes between products that have no feature changes and products that are no longer on the website. Running the indexer at the process layer partly solves this problem by indexing only the products that are extracted at the last crawl. This means that when the user performs a search, only the products that are currently present on the websites are returned as results. Unfortunately, this poses a dilemma regarding the products that are no longer on the websites: should they appear in the search results or not? Usually, when a product is removed from a retailer website, it is removed from that website's search and the product page is not accessible by navigating the website; however, it might be accessible by using an external search engine. If we also take into account that a shopping search engine needs to have a price history component, the chosen option is to leave the products in the database and to just mark them as permanently out of stock.

The indexer's role is to split each product name into words and to obtain for each word a list of products whose names contains that word; this is a typical example of the MapReduce algorithm [12]. In its current state, the indexer allows only words that have a length greater than three and contain no prepositions. This simple approach is enough for the proof-of-concept.

2.8. Present Action

This action is represented by a web application deployed on a cloud instance, which allows the end-user to search products by name. Other main features of the presentation solution are: the ability to view the list of products that were added in the last week, to view the list of retailer websites that were crawled, and to save the desired products in the user's favorites list. Another useful feature is the product price history chart, which allows the user to see the evolution of the product price. There are many features that can be implemented, but the main focus is to minimize resource usage, especially at the presentation layer.

Regarding the communication between the processing and the presentation layers of the proposed solution, this is achieved by means of RESTful micro web services, as shown in Table 2. The communication situations are designed to have a low impact on the number of requests and on the number of reads and writes on the database.

Table 2. Exposed RESTful micro web services at the presentation layer.

URL	HTTP Method	Payload / Query	Description
/sites	GET	-	Returns the websites
	POST [1]	*website*	Adds a new website
	PUT [1]	*website list*	Replaces the website list
	DELETE [1]	-	Removes all the websites
/sites/*siteUrl*	PUT [1]	*website*	Replaces the website
/products	GET	search=*terms*	Performs a product search by name
		newest=true	Returns the products that were added in the last week (limited to 100 items)
		ids=*listOfIds*	Returns the products with the specified IDs (limited to 100 items)
	PATCH [1]	*product list*	Batch-updates the products
	DELETE [1]	-	Removes all the products
/productindex	GET [1]		Returns the product index
	PUT [1]	*product index*	Replaces the product index

[1] Method available to be called only from the processing layer.

Optimizations are made on the presentation server by using a memory cache, which is shared among the cloud-created instances of the web application, thus reducing the communication with the database. In order to lower costs, the product index is kept in the database and in the memory, rather than relying on the database's indexing.

2.9. Proposed Solution for Product Indexing

When the user enters a search string, for each word entered, a lookup is performed in the product index. Afterwards, if there is more than one word in the search string, an intersection of the results for each word is performed. The product index is a Patricia Trie data structure, which can be distributed as shown in [13], where the key is the word and the associated value is a list of product IDs. For each ID, the product information is retrieved from the memory cache with the database as fallback. In terms of time complexity in the average case, the product indexing is performed in $O(n)$, where n is the number of products that are indexed, while the product search is performed in $O(1)$. These low time complexities are possible because a Patricia trie is used to store the information.

The separate indexing approach has the advantage of allowing optimizations to be made regarding the database access, i.e., reducing the number of reads and writes. Currently, the product index is compressed and is kept as an entry in the cloud database, where the key is an empty string. If the entry gets too large, the storage method takes advantage of the trie structure and splits the index by the string prefix. Further research regarding the indexing will pe performed in order to achieve better scalability.

An advantage of not relying on the database indexing solution is that database access costs are reduced, at the expense of some computational costs. Moreover, the computation of the index can be performed either on the crawler, extractor, and processor side, or on the deployment side. Taking a low-level approach and moving the indexing to the cloud can benefit from the cloud's push/pull queues feature. On the other hand, the indexing logic can be easily extended to use third-party solutions like Apache Lucerne. Currently, the indexing is performed locally as the product information is processed, but there is also a re-indexing service implemented on the cloud application.

2.10. Security and Ethics

2.10.1. Crawler Security and Ethics

The proposed crawling solution simulates the behavior of a user by sending requests at random times and by sending the appropriate HTTP headers to the crawled server as if the request was made from a browser.

In terms of ethics, a crawler must adhere to the robots exclusion standard by processing a robots.txt file, which regulates what pages can and cannot be visited by a crawler on a specific website [14]. If those rules are not obeyed by the crawler, there is a slight chance of getting itself trapped in a honeypot [15]. For example, in a webpage there can be links to fake webpages that are not visible in the browser, which, when crawled, result in fake product data or in other fake webpages, maybe from different websites.

Usually, retail websites want to have their products indexed by a search engine, but the crawler must be designed to prevent situations in which it retrieves and processes useless webpages. From this point of view, the proposed solution ensures the crawler's security by specifying, in the website template and with the help of regular expressions, what URLs are crawled and what URLs are extracted.

2.10.2. Presentation Layer Security

The presentation layer hosted in the cloud exposes RESTful web services in order for the product information to be added to the database and allows the users to search for products by name. Regarding the security issues identified in [16,17], the cloud solution where the presentation layer is deployed ensures server security, and in terms of the web application, the appropriate measures have been taken to ensure protection from attackers taking advantage.

At this stage, there is no possibility of end-users logging in on the website, so there is no user access and control to secure. However, there is the communication between the processing and the presentation layer, which is protected from man-in-the-middle attacks through HTTPS. In addition, some of the micro web services can only be accessed by the deployer component of the processing layer via IP filtering. Protection against improper input validation (e.g., SQL injections, cross-site scripting) is achieved by allowing only letters and numbers in the website's search input and by limiting the number of characters in the search string. The end-user is able to call only two API methods: to get all the site information and to get product information, as can be seen in Table 2. Regarding the favorite products logic, this is performed in the browser using the local storage and has no influence on the website's security. As the presentation component evolves, penetration testing tools will be employed to prevent attackers from exploiting the application.

Another aspect is third-party applications using the data presented on the proposed website. In a sense, this is similar to having a search engine and preventing other applications from using the search results. The first employed step is the cross-origin resource sharing (CORS) method, which controls access when requests are coming from other origins. Unfortunately, this does not prevent another web application to simply do a GET request server-side and to obtain the data. The best way to protect from third-party applications is to detect that the request came from an application rather than an end-user. This situation is similar to the one discussed in the Crawler Security subsection of this paper, but this time the presentation layer is the victim, whereas there, the crawled websites were the victims and the crawler was the attacker.

3. Use—Case Scenario and Results

The goal is to have a fully-functional vertical search engine, and in order to test the proposed solution, the chosen vertical is board and card games. Seven websites are crawled: barlogulcujocuri.ro, lelegames.ro, lexshop.ro, ludicus.ro, pionul.ro, redgoblin.ro, and regatuljocurilor.ro. The deployed website for the end-user is shown in Figure 2 and is accessible at: http://boardgamesearch.h23.ro/.

All of the crawling, extraction, and processing is performed on the same computer for an accurate measurement of the system performance. There are two main applications hosting the solution components. The first is a Java application that uses a MongoDB database and hosts the crawling, extraction, and processing components. For the presentation component, a Java-based web application is used and deployed on the Google Cloud platform with Google Datastore as the database [18]. Other employed technologies are Jersey RESTful web services, memory cache services, and push queues.

The reasoning behind choosing the Google Cloud platform is that it is free with daily limitations; every 24 h the quotas are reset.

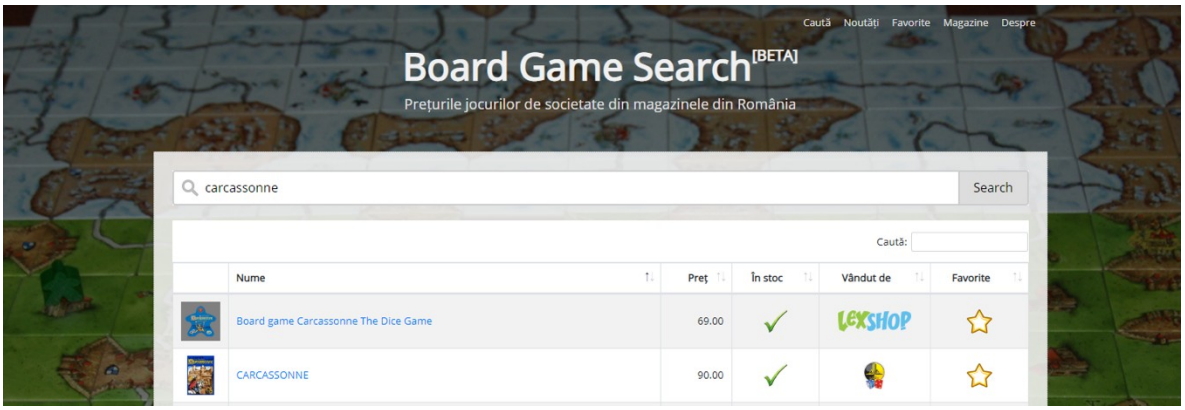


Figure 2. Results page for the board game search website: http://boardgamesearch.h23.ro.

For each website, a crawler and extractor configuration file are used, as can be seen in Listing 1 for the lexshop.ro website. For each product, the properties/features that are extracted are: URL, name, image, price, and availability, which represent the bare minimum for offering relevant results to the user. Ideally, the configuration is to be obtained manually, but for now, as previously mentioned, it is entered manually after analyzing the websites in question. More aspects regarding the website template analysis are presented in the Discussion section of this paper.

```
{
   "name": "lexshop.ro",
   "url": "https://www.lexshop.ro",
   "logoUrl": "https://www.lexshop.ro/app/images/logo.png",
   "fetchUrlRegEx": "\\Qhttps://www.lexshop.ro/?page=produse&categorie=8&n=\\E[0-9]*",
   "seedList": [
"c8-board-games"
   ],
   "extractor": {
          "extractUrlRegEx": "\\Qhttps://www.lexshop.ro/?page=produse&categorie=8&n=
          \\E[0-9]*",
          "baseSelector": "div.list-products>div>div>div[data-href]",
          "uniqueKeyProperty": "url",
          "properties": {
                 "url": "div.product_img_container>a%abs:href",
                 "name": "div.prod_title_container>a",
                 "image": "div.product_img_container>a>img%abs:data-original",
                 "price": "div.prod_prices>span.actual_price",
                 "availability": "div.product_img_container>div.eticheta-stoc=
                 ~STOC EPUIZAT"
          }
   }
}
```

Listing 1. Sample crawler configuration and extractor template file representing a single website.

The crawl, extraction and processing times per page for each of seven online board game stores are shown in Table A1 (in Appendix A), where the processing time is the sum of the page retrieval, link extraction, product extraction, and server upload times. The sum values from the table depend on the number of links found and the number of extracted products from each page; these values are presented in Table A2 (in Appendix A), The volume of data that is extracted from each website varies significantly. For example, on one website there are around 852 links per page, while on another there are 190 links. In addition, some sites permit the product display with a maximum pagination of 60, while others only allow 12 products per page. All of this leads to different page processing times. Overall, from the 851 retrieved pages, there were 14860 unique products extracted in 1937 s. There were approximately 17 products per page, and the time to process a product in these conditions is around 0.13 s. The compressed product index is kept in the datastore in a blob of 275 KiB; the size is significantly affected by the fact that, for now, the product IDs are strings instead of long-type values.

An interesting aspect is that for some sites, the number of unique products found in the last crawl is significantly less than the number of unique products stored in the database. For example, the Red Goblin website currently has 4575 products in the database, but in the last run, only 2037 products got extracted. This is due to the fact that 2538 products were previously available in the last year on the site but have been removed since then. Four of the six websites were crawled for approximately one year, and two among them have removed most of the products that are out of stock, while the other two just change their availability.

4. Discussion

The system architecture decisions are presented in Section 2 together with a description of the components. Here, there is a focus on the implications that the results might have on expanding and improving the proposed solution.

In order to minimize the number of pages that are processed, the information is extracted from multi-product pages. From a single webpage, on average, there are 17 unique products extracted. The advantage is obvious: less requests to the servers hosting the websites, less chance to ban the crawler for initiating too many requests, and most importantly, less time to extract the information. This works because we need only the bare minimum of information. On the other hand, for example, if one wants to also extract the product description, then the regular expression that matches the page URLs used for extraction will have to be changed to match single-product pages that have that description. Moreover, all of the custom selectors will have to be updated to the single-product page template. Figure 3 shows the number of extracted products per second from each considered retailer. These values depend mostly on the webpage content size and secondly, on the number of extracted pages and the number of products per page. For example, on the Pionul website, an average of 14.57 products/page were extracted, compared to 1.96 products/page on the Barlogul cu jocuri website, because the latter had a smaller average page content size. Considering all of the page processing times, it takes the longest time to retrieve the webpage, as can be seen in Figure A1 (in Appendix B), which shows the total sum of the processing times in relation to the total number of uniquely extracted products for each of the seven considered websites.

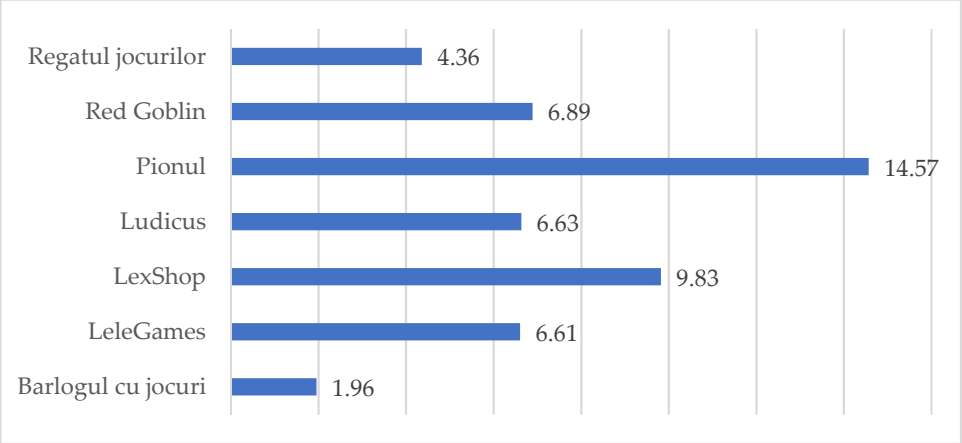


Figure 3. Number of extracted products per second for each of the seven considered retail websites.

A site that was in contention to be crawled was transylvaniagames.com. The problem with the site is that the product availability is not present in the multi-product pages and, more importantly, the product URLs are in the form transylvaniagames.com/*product-name*. This poses the problem of distinguishing between product pages and the other pages. The best solution is to go from the multi-product pages to the product pages and extract from there. Therefore, the next step is to extend the template to allow, in some cases, the retrieval and processing of the extracted product URLs, which are always present on multi-product webpages.

Another issue is what is considered to be the unique key, which is used to determine if a product that was extracted is the same as a product that is already in the database. In practice, and considering that there is no product that can be accessible by two different URLs, the unique key is composed of the site ID and the product page URL path: it needs to be composed because we can have the same URL path on different websites.

For all seven websites, the page retrieval time is approximately ten times the sum of the link extraction, product extraction, and server upload times. This means that if we want to achieve scalability, it would make sense to have more instances that just retrieve the webpage and store it in a database, compared to the number of instances that perform the extraction and upload. The problem with storing the webpage content in the database is in terms of storage and the transfer times to store and to retrieve the page content. An optimization for storage is to compress the page before saving it in the database. Another advantage is that the crawler can use the cache-related HTTP headers to determine if the webpage has been modified since the last crawl. Unfortunately, in most situations the cache-control header is set to no-cache, or it is not accurate. From a crawler's point of view, the best way is to use a pool of crawlers, with each crawler obtaining from the database the URL that was accessed earliest and comes from a website that wasn't crawled in the last seconds. In this way, we have no crawler downtime and the website servers do not ban the crawler for making too many requests.

In terms of resource usage for a shopping search engine, there are a few key aspects that need to be optimized: the incoming bandwidth, the instance hours, and the datastore read and write operations. By having the presentation component on the cloud, we are left to optimize only the datastore read and write operations. In the free version of the Google Cloud platform, these are limited to 50,000 reads and 20,000 writes. The writes are consumed when deploying products on the remote web server, and the reads are consumed when the user performs a search, or visits the latest products section or the favorites section. While the cache service helps to significantly reduce the number of database reads, if no one accesses the website for a certain amount of time, the cloud instance of the website is removed from memory together with the cache. Initially, each product was stored as one entry in the database, but a more efficient way is to store multiple products in the same entry. This is feasible because the proposed solution keeps its own product index. We can obtain another increase in efficiency if we can

minimize the number of database entries that are retrieved when a search is performed. Thusly, it will help to keep track of the most searched terms and keep most of the results in a single entry in the database. The drawback is that there is a limitation on the entry size of approximately 1 MiB. This means that we need a function to determine if the entry is less than 1 MiB by computing the size in bytes of each product.

In terms of related work, existing shopping search engines rely on a product feed from the online retailers, which is usually in the form of an XML containing the elements: manufacturer, product name, category, product URL, price, identifier, image URLs, description, and other similar fields. Examples of shopping search engines that obtain product information from retailer feeds are: pricegrabber.com, shopzilla.com, shopping.com, compare.ro, and price.ro.

- *Similarities between shopping search engines that use retailer-provided feeds and the proposed solution*

The end-result is the same: provide the user the means to search for products and find the best available price. In both situations, the search engine needs to periodically check if new products were added, if products were removed, and if the product information somehow changed. The traditional shopping search engines offer each retailer the possibility of specifying the feed update frequency, while the proposed solution crawls each website daily. On the other hand, the crawl frequency can differ for each website based on the product update history, e.g., some retailers add new products only at the beginning of the week, and other retailers rarely change the product prices.

- *Advantages of shopping search engines that use retailer-provided feeds vs. the proposed solution*

For existing shopping search engines, the product information is accurate, regardless of any change in the webpage's template; however, websites rarely change their page structure and the proposed solution sends a notification for a template update if a structure change is detected. There is little resource usage on the search engine server because the only processing is to interpret the XML feed for each website. The feeds use pull technology, i.e., the initial request in made by the client (search engine), and therefore any change in the product information is obtained only when the search engine performs a request to that retailer's feed.

- *Advantages of the proposed solution vs. shopping search engines that use retailer-provided feeds*

A critical advantage of the proposed solution is that it does not require any involvement on the retailer's part. In order to obtain a product feed, there needs to be an interaction between the administrators of the search engine and the retail website, and more often than not, the retailer has to pay a fee to have its products indexed. The proposed solution eliminates this interaction by simply crawling the retailer webpages and extracting the product information, even from websites that are not willing to provide a product feed. Another important advantage is that the proposed solution is designed to be generic and works on any type of extractable information (e.g., food recipes), compared to shopping search engines, which allow only product searches.

In regards to comparing the performance of the proposed solution with existing shopping search solutions, it is significantly difficult to compare resource usage because the third-party solutions do not provide any information of this type. On the other hand, there is one aspect that can be evaluated: the volume and accuracy of the product data. In this regard, the proposed solution has the advantage of having product information from more websites, not just the ones that want to provide XML feeds, while maintaining the same data accuracy. For board games, an example is the website boardgameshub.ro, which is a search engine that uses XML feeds but does not index products from big retailers that also sell other types of products (e.g., elefant.ro, carturesti.ro), because those retailers feel that they do not get any added benefit from providing such a feed to the board game search engine.

Another somewhat related solution in terms of product extraction is the web scraper from webscraper.io. It is a browser extension that allows the user to specify, on a webpage, the location

of items to be extracted, which are then saved in CSV format. The main disadvantage compared to the proposed solution is that the web scraper allows only simple extraction from HTML elements using CSS selectors, i.e., it does not permit conditional extraction. For example, there are websites that have some products on sale, and the real price has a different CSS selector compared to the normal products. The web scraper has the advantage of its simplicity and ease-of-use, but it only provides simple scraping, which is only a part of the solution proposed in this paper.

In terms of security, the main aspects are discussed in the Security and Ethics section of this paper. Regarding the interaction between the user and the board game search website, that interaction is minimal, i.e., the user can input a search string, which is sanitized in order to prevent cross-site scripting. The retrieval solution adheres to the crawler ethics, and therefore no website banned (even temporarily) the crawler component. In addition, the communication between the processing and the presentation component is not visible to the public, and the data transfer is secured by means of HTTPS.

5. Conclusions

The proposed solution is a fully-functional vertical search engine that works not only on products but also on other types of extractable information (e.g., food recipes). A novel contribution that this paper presents is a product retrieval solution that does not depend on XML feeds provided by the retailers, but rather provides a concise, flexible, and efficient method of retrieving and extracting information by employing a novel template used to represent the location of the extractable data from webpages. Another novel aspect is the cost-effective method of storing data on a cloud platform and indexing it to minimize resource usage, while also providing efficient solutions for reducing resource consumption and, therefore, the costs in the other system components. Finally, a discussion is presented on the ethics and security that the proposed system poses.

In terms of extending the research in the considered field, the proposed solution has the great potential of being a framework for developing and testing new methods for product extraction (e.g., determining the extraction template using neural networks or determining if a product is the same as one from another website even though it has a slightly different name). The next research step is to move the crawler onto the cloud platform and make it as efficient as possible. The indexing component is already moved to the cloud and uses Google's push queues to execute the tasks that perform the indexing. There are two types of indexing: a live indexing as the web server receives batches of products, and a re-indexing service, which splits the process into multiple tasks in order to follow the cloud's task duration restrictions. Another considered extension of the proposed solution is to deploy it on an OpenStack cloud platform.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. Crawl, extraction, and processing times per page for each of seven online board game stores (expressed in seconds). The processing time (written in bold) is the sum of the page retrieval, link extraction, product extraction, and server upload times.

	Barlogul cu jocuri	LeleGames	LexShop	Ludicus	Pionul	Red Goblin	Regatul jocurilor	Total
Sum of page retrieval times	43.05	43.90	185.60	106.81	27.38	211.78	947.70	1566.21
Sum of link extraction times	0.30	2.13	8.94	1.00	4.22	5.25	19.39	41.22
Sum of product extraction times	2.75	9.04	32.42	23.09	10.44	54.30	51.08	183.11
Sum of server upload times	0.44	4.84	47.50	10.63	7.31	24.19	52.24	147.14
Sum of processing times	**46.53**	**59.91**	**274.46**	**141.53**	**49.34**	**295.51**	**1070.41**	**1937.68**
Average of page retrieval times	4.78	1.83	0.78	5.09	0.76	2.75	2.13	1.84
Average of link extraction times	0.03	0.09	0.04	0.05	0.12	0.07	0.04	0.05
Average of product extraction times	0.31	0.38	0.14	1.10	0.29	0.71	0.11	0.22
Average of server upload times	0.05	0.20	0.20	0.51	0.20	0.31	0.12	0.17
Average of processing times	**5.17**	**2.50**	**1.15**	**6.74**	**1.37**	**3.84**	**2.41**	**2.28**

Table A2. Number of links and products found per page, the number of retrieved pages, and the total number of uniquely extracted products in the last crawl.

	Barlogul cu jocuri	LeleGames	LexShop	Ludicus	Pionul	Red Goblin	Regatul jocurilor	Total
Sum of number of found links	2065	4552	70682	5317	4858	65596	131527	284597
Sum of number of found products	103	432	2858	1246	719	4620	9151	19130
Average of number of found links	229	190	296	253	135	852	296	334
Average of number of found products	11	18	12	59	20	60	21	22
Number of retrieved pages	9	24	239	21	36	77	445	851
Number of uniquely extracted products	91	396	2697	939	719	2037	4672	11551

Appendix B

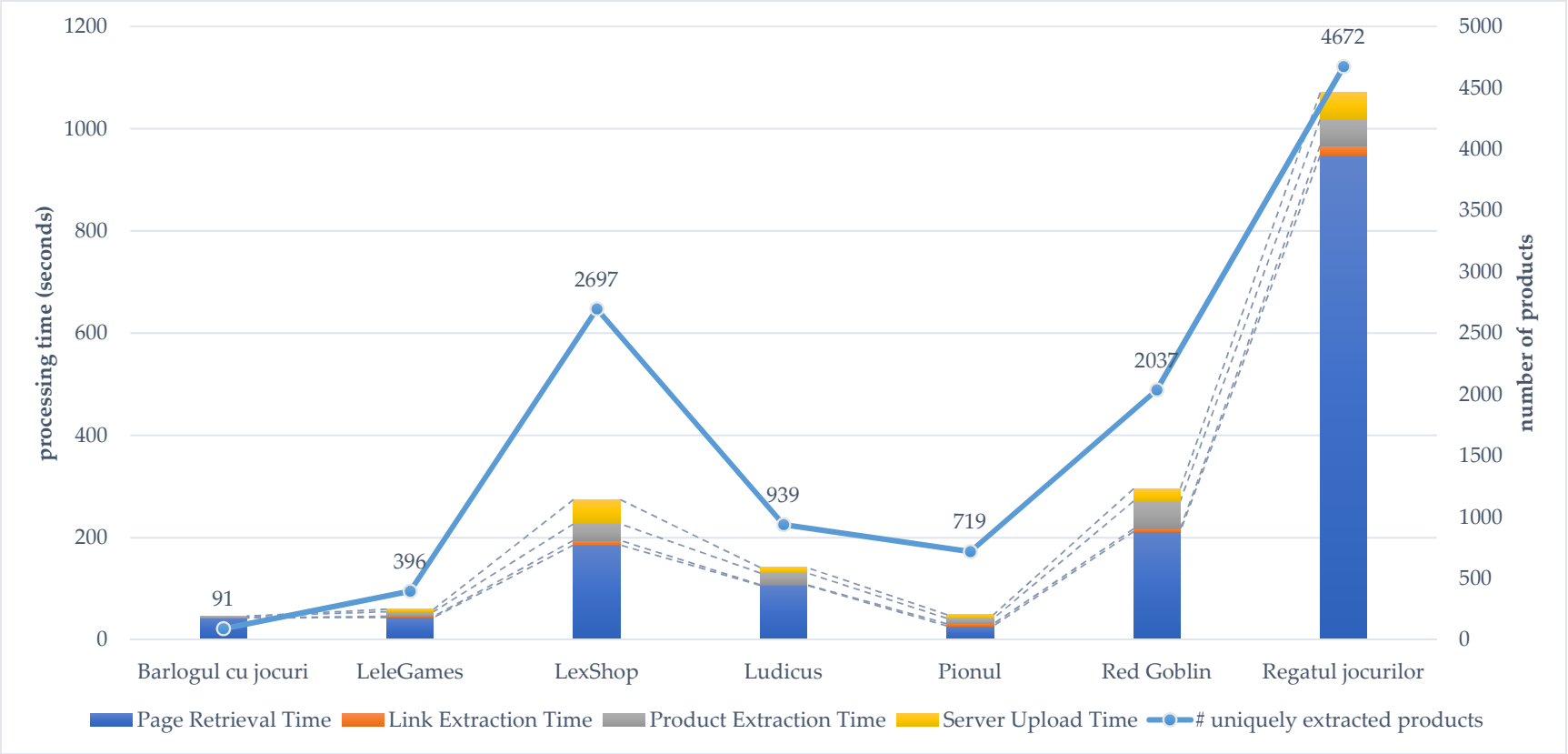


Figure A1. Total sum of processing times (i.e., page retrieval, link extraction, product extraction, and server upload times) in relation to the total number of extracted products from each of the seven considered websites.

References

1. The 10 Best Shopping Engines—Search Engine Watch Search Engine Watch. Available online: https://searchenginewatch.com/sew/study/2097413/shopping-engines (accessed on 4 October 2018).
2. Lin, L.; He, Y.; Guo, H.; Fan, J.; Zhou, L.; Guo, Q.; Li, G. SESQ: A model-driven method for building object level vertical search engines. In Proceedings of the International Conference on Conceptual Modeling, Barcelona, Spain, 20–24 October 2008; Springer: Berlin, Germany, 2008; pp. 516–517.
3. Ahmedi, L.; Abdurrahmani, V.; Rrmoku, K. E-Shop: A vertical search engine for domain of online shopping. In Proceedings of the 13th International Conference on Web Information Systems and Technologies, Porto, Portugal, 25–27 April 2017; pp. 376–381.
4. Mahto, D.; Singh, L. A dive into Web Scraper world. In Proceedings of the 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 16–18 March 2016; pp. 689–693.
5. Gogar, T.; Hubacek, O.; Sedivy, J. Deep Neural Networks for Web Page Information Extraction. In Proceedings of the IFIP International Conference on Artificial Intelligence Applications and Innovations, Thessaloniki, Greece, 16–18 September 2016; Springer: Cham, Germany, 2016; pp. 154–163.
6. Kamanwar, N.V.; Kale, S.G. Web data extraction techniques: A review. In Proceedings of the 2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave), Coimbatore, India, 29 February–1 March 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1–5.
7. Ferrara, E.; De Meo, P.; Fiumara, G.; Baumgartner, R. Web data extraction, applications and techniques: A survey. *Knowl.-Based Syst.* **2014**, *70*, 301–323. [CrossRef]
8. Alexandrescu, A. A distributed framework for information retrieval, processing and presentation of data. In Proceedings of the 2018 22nd International Conference on System Theory, Control and Computing, ICSTCC 2018, Sinaia, Romania, 10–12 October 2018; IEEE: Piscataway, NJ, USA, 2018.
9. Lampesberger, H. Technologies for web and cloud service interaction: A survey. *Serv. Oriented Comput. Appl.* **2016**, *10*, 71–110. [CrossRef]
10. He, B.; Patel, M.; Zhang, Z.; Chang, K.C.-C. Accessing the deep web: A survey. *Commun. ACM* **2007**, *50*, 94–101. [CrossRef]
11. Hedley, J. jsoup: Java HTML Parser. Available online: https://jsoup.org/ (accessed on 23 February 2019).
12. Hiemstra, D.; Hauff, C. MIREX: MapReduce Information Retrieval Experiments. *Comput. Res. Repos. arxiv* **2010**, arXiv:1004.4489.
13. Knollmann, T.; Scheideler, C. A Self-stabilizing Hashed Patricia Trie. In Proceedings of the 20th International Symposium on Stabilization, Safety, and Security of Distributed Systems, Tokyo, Japan, 4–7 November 2018; Springer: Cham, Switzerland, 2018; pp. 1–15.
14. Sun, Y.; Councill, I.G.; Giles, C.L. The Ethicality of Web Crawlers. In Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Washington, DC, USA, 31 August–3 September 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 668–675.
15. Nawrocki, M.; Wählisch, M.; Schmidt, T.C.; Keil, C.; Schönfelder, J. A Survey on Honeypot Software and Data Analysis. *arxiv* **2016**, arXiv:1608.06249.
16. Subashini, S.; Kavitha, V. A survey on security issues in service delivery models of cloud computing. *J. Netw. Comput. Appl.* **2011**, *34*, 1–11. [CrossRef]
17. Deepa, G.; Thilagam, P.S. Securing web applications from injection and logic vulnerabilities: Approaches and challenges. *Inf. Softw. Technol.* **2016**, *74*, 160–180. [CrossRef]
18. Cloud Computing Services | Google Cloud. Available online: https://cloud.google.com/ (accessed on 15 January 2019).

Monitoring and processing of physiological and domotics parameters in an Internet of Things (IoT) assistive living environment

Adrian Alexandrescu*, Nicolae Botezatu†, Robert Lupu‡
Faculty of Automatic Control and Computer Engineering, *Gheorghe Asachi Technical University of Iaşi*
Iași, Romania
* adrian.alexandrescu@academic.tuiasi.ro, aalexandrescu@tuiasi.ro
† nicolae-alexandru.botezatu@academic.tuiasi.ro
‡ robert-gabriel.lupu@academic.tuiasi.ro

2022 26th International Conference on System Theory, Control and Computing (ICSTCC) | 978-1-6654-6746-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICSTCC55426.2022.9931865

***Abstract*—In relation to an ever changing epidemiological world context, a category of people that is more subject to be impacted consists of the elderly. Certain steps can be taken in order to improve their quality of life especially in case of illness. One way of achieving this is to have a smart assistive living environment, which includes home automation and medical monitoring. The proposed system expands on an IoT solution for assisted living and introduces a highly flexible rules engine for processing physiological and domotics data obtained from the home environment, and for interacting with the system actuators. As proof-of-concept, there are several use-cases that are discussed depending on the type of patient: diabetic, cardiac, hypertensive, obese, COVID or Alzheimer. These scenarios emphasize the efficiency of the proposed solution and offer an insight on the high degree of abstraction and extensibility of the system.**

***Index Terms*—ambient assisted living, home monitoring, IoT, patient monitoring, rules engine, palliative care**

I. Introduction

In our society there is a sense of duty to take care of the ill and the elderly, and there are multiple dedicated facilities that house such people. However, living in such an establishment lacks the feeling of being at home. The best place where one feels at home is home, but there is the issue of how can ill elderly persons handle functioning by themselves.

With the appearance of concepts such as Internet of Things, smart homes and smart cities [1] having an automated home goes a long way to achieve an ambient assisted living (AAL) environment, which can significantly help the lives of elderly people. There is extensive research being done when it comes to assisted living [2] and this research goes across different fields of study. Even before the COVID outbreak and even more in the last years there is an increasing need to reimagine assisted living [3] and this was accentuated by the COVID-19 pandemic and the need to include this concept in response to the pandemic [4]. Apart from home automation, an ill person requires more care and monitoring. This is why there is a need also for a medical monitoring system that can allow a caretaker to remotely see various physiological parameters of the patient without actually being in the patients home.

One such system is the SMARTCARE solution for 'assisted living' [5] allows intelligent home automation through the monitoring and control of physiological parameters, as well as the home environment. Assistance for autonomy at home allows remote control of the home, along with ensuring palliative care and an independent living of the elderly and/or persons suffering from chronic or mental illness.

This paper expands on this solution by automating even more the whole process with a rules engine and by considering specific scenarios depending on the type of illness.

II. Related Work

The current paper is a continuation of the research from [5], where the authors present an overview of an IoT solution for assisted living, namely the SMARTCARE system, with a focus on the user roles and on how the communication with various devices is handled in terms of the real-time interaction with the devices and the abstraction of the different means of interfacing with devices (e.g., Z-Wave wireless network, Bluetooth, more or less standardized APIs, self-organizing networks like ZigBee). Another related research is the one from [6], in which the authors consider an ambient assistive environment that allows the acquisition of data pertaining to physiological parameters of patients living in their homes and accessing that data remotely. The focus here is on what vital parameters are a obtained and why is this data needed.

The main novelty of the current research compared the two aforementioned papers consist of the presentation of the rules engine, which allows setting triggers and actions depending on the acquired data from devices, and the considered use-case scenarios, which show how the information regarding each domotics and vital parameter is processed and tailored depending on the type of patient in home assisted living.

There are other papers that tackle data acquisition from domotics and other devices, which are loosely related to the present research. For example, in [7], the authors present an infection tracing system, in which data is acquired from a low-power device for detecting dispenser interaction, with a focus on determining a solution that allows the reducing

the power consumption. In [8] it is presented a system for monitoring air quality in an indoor environment by obtaining data regarding temperature, moisture, carbon monoxide and dioxide, and glow. The acquired data can be viewed in a web and a mobile application, but, compared to the current paper, the communication with the sensors is done only by means of ZigBee, and there are no means to set rules and triggers.

In terms of handling rules, in [9] the authors describe an ontology for activities in an ambient assisted living environment. Each sensors is seen as an entity with associated properties, and specific and more complex sensors are defined as a composition of other sensors. The main difference compared to the current paper is that, in [9], the focus is on specific activities that the person can do (e.g., the person is dressing up) and it is more difficult to specify complex rules.

III. General Overview of the Proposed Solution

The proposed solution is a continuation of the initial SMARTCARE prototype proposed in [5] and it focuses on considering specific devices employed for different types of patients and on the added value given by the rules engine, which is need in order to have a complete and efficient palliative care environment.

Looking at the architecture of the proposed solution from Fig. 1, there two main parts: firstly, the domotics and the physiological/vital devices, and, secondly, the Gateway system.

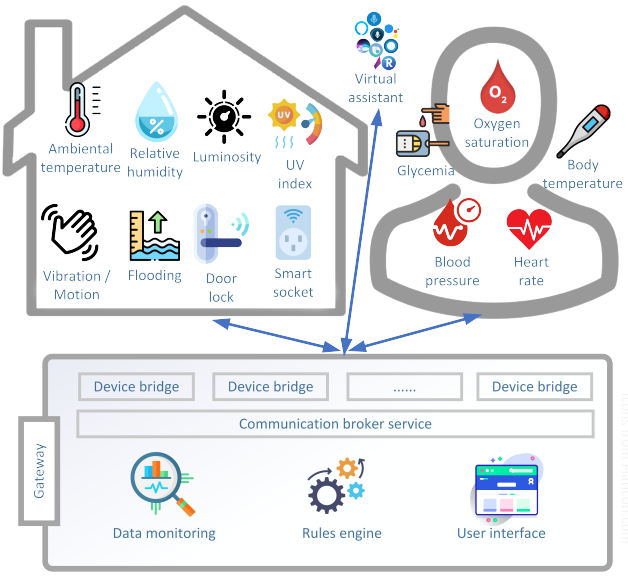


Fig. 1. SMARTCARE gateway system architecture

In terms of the devices, there are sensors and actuators for home automation (i.e., domotics), sensors and devices that acquire physiological data, and a virtual assistant (e.g., Alexa) for sending audio messages to the patient.

The domotics measurements that the system processes are ambient temperature, relative humidity, luminosity, UV index, vibrations / motion detection and flooding detection. There are also an actuator for door locking and a linear actuator with consumption monitoring for controlling the water tap. For monitoring the vital parameters there are sensors and devices that provide the values for blood pressure, glycemia, heart rate, body temperature, oxygen saturation (SpO2).

The communication between the Gateway system and the connected devices is achieved by means of Device Bridge processes, which posses the business logic to interact with specific devices. The main business logic of the system in the Gateway Core, which encompasses the data monitoring logic, the rules engine and the graphical user interface. In order to have a high level of abstraction the communication between the Gateway Core and the Device Bridges is done via the Communication Broker Service, which is a publish-subscribed messaging protocol well-suited for IoT solutions.

Each device parameter is uniquely identified in the system in the form `bridgeName/deviceName/resourceName`. The first part of the identifier is the `bridgeName` because a Device Bridge process can handle one or more devices depending on the employed communication protocol, and we need to know with which bridge we need to communicate. The second part is the `deviceName` and it represents the identifier of the connected device. Lastly, there is the `resourceName`, which is needed because there are devices that measure multiple parameters (e.g., heart rate and SpO2). In the Communication Broker Service there are multiple message queues for each uniquely identified device parameter. For example, the Core Gateway can listen to the queue `vital/1/glycemia/get` for values of the `glycemia` resource from device `1` of the `vital` bridge, and it can send a message to an actuator by publishing to a queue like `va/1/virtual-assistant/set`. The Communication Broker Service is more complex than this, but the intricacies of this component are beyond the scope of this paper.

IV. Connected Devices

Most AAL (Ambient Assisted Living) and HMS (Home Monitoring System) are dependent on specific hardware either because they lack flexibility at the interface level (i.e. fixed number of hardware interfaces, limited number of communication protocols) or due to the use of custom hardware. Also, with each technological iteration, manufacturers might integrate new features (i.e. updated hardware, new communication standards) in their devices thus having limited or no backward compatibility to an existing system.

Our solution mitigates the aforementioned problems by implementing a device layer that supports heterogeneous deployments with a wide range of sensing devices and actuators. The layer consists of modular software interfaces called *Device bridges* that implement standardized Service Access Points for interaction with the upper layers of the system. These bridges can be run on any hardware (i.e. with respect to the CPU architecture) allowing the integrator to choose the most suitable platform (e.g. number and type of USB and network interfaces, RAM memory capacity) for the implementation.

The SAPs describe a custom interaction protocol implemented on top of MQTT, the *Communication broker service* component of the Gateway having a central role for configuration and data acquisition.

V. Rules Engine

One of the core Gateway elements is the *Rules engine* component, whose role is to react to changes in the acquired values and to trigger alerts.

There are three classes of messages:

- *notification* - the corresponding message is shown in the user interface and the patient is notified by means of the virtual assistant,
- *alert* - same action as for the notification class, plus the legal caretaker is notified,
- *urgency* - same action as for the alert class, plus the help procedure is initialized.

Each time data is obtained from the sensors, that data is checked against the facts (i.e., the rules defined for that particular patient) and, if certain conditions are met, then the appropriate actions are taken. For example, if the recorded glycemia value is above 120 mg/dL, then the virtual assistant is used to send an audio notification to the patient: "Warning! The glycemia level is too high. The value is at ...ml/dL".

When a rule is defined there are five properties that have to be set: name, description, priority, condition and actions. The first two properties are there for readability, the priority is taken into consideration if multiple rules trigger at the same time (higher priority triggers first), the condition is an expression that uses recorded parameter values, comparison operators, conditional operators and static values, and, the actions property is a list of actions that represent anything from messages saved in the database to setting specific values to the device actuators in the system; basically, an action can be any custom line of code with a limited context, which includes access to communication with the devices.

The rules are salved in JSON format as it is shown in Listing 1 for the aforementioned glycemia example.

```
{
  "name": "glycemia-notification",
  "description": "The patient is notified if the glycemia is
      ↪ over 120 mg/dL",
  "priority": 3,
  "condition": "${vital/1/glycemia}{value} > 120",
  "actions": [
    "${va/1/virtual-assistant}{value}{Warning! The glycemia
        ↪ level is too high. The value is at ${vital/1/
        ↪ glycemia{value} mg/dL}"
  ]
}
```

Listing 1. Rules engine - glycemia notification JSON string

Both the condition and the actions property can employ a custom notation that is used to interact with the sensors and actuators in the system. There are two variants of the custom notation. The first one is ${*resourceName*}{*propertyName*} and it will be replaced by the latest value of the property specified by *propertyName* of the resource with *resourceName* from the database. While the first variant is, more or less, equivalent to a *getter* for the specified resource value, the second variant is somewhat equivalent to a *setter*. The notation ${*resourceName*}{*propertyName*}{*propertyValue*} set the specified property's value of the *resourceName* to *propertyValue*. This triggers an immediate communication with the corresponding Device Bridge via the Communication Broker Service.

The Rules Engine component is comprised of the *Facts Service*, which encompasses the defined rules and which triggers the appropriate one each time a specific parameter (vital or domotics) is changed, the *Bridge Interaction Service*, which updates the facts used by the facts service and which communicates the appropriate actuator when an action is performed, and the *Database Access Service*, which obtains from the database the latest parameter values.

The role of the Bridge Interaction Service is twofold: to acquire data from the sensors and to signal actuators in the system to perform actions (e.g., to send to the virtual assistant the text that has to be "read" to the patient). The Bridge Interaction Service communicates directly with the Communication Broker Service, who is the mediator between it and the Device Bridges. Each time a piece of data is acquired, that information is stored in the Gateway database by means of the Database Access Service, and the Facts Service is employed to check if there are facts that trigger one or more actions.

VI. Testing Template

A. Hardware setup

The testing setup included several devices for home automation and physiological parameters monitoring. A Z-Wave network based on an *Z-Wave.Me UZB* Primary Controller running the Z-Wave stack interfaced with a *Device bridge* based on openZWave library was set up to expose the parameters presented in Table I. It included an *Aeotec Multisensor 6*, an *ABUS SHWM10000* water sensor, a *Danalock V3* door lock, a *Fibaro FGS-223* power switch and an *POPP Flow Stop 2* actuator. For the patient monitoring we employed a *Libelium MySignals* system that integrates sensors for all the parameters included in the testing scenarios (Table II).

B. Use-Case Context

In order to test the proposed system and especially the rules engine, multiple use-case scenarios were created by using sensor data referring to the domotics parameters from Table I and the vital parameters from Table II. The readings are generated based on data from actual devices. There are six types of patients that are considered and, depending of the patient type, several parameters are monitored as follows:

- diabetic patient: blood pressure, glycemia, oxygen saturation, ambient temperature, relative humidity
- cardiac / hypertensive patient: blood pressure, heart rate, oxygen saturation, ambient temperature,relative humidity
- obese patient: blood pressure, heart rate, glycemia, ambient temperature
- COVID patient: body temperature, heart rate, oxygen saturation
- Alzheimer patient: door lock status, running water status, ambient temperature

All the following monitoring scenarios are performed over a six hour time window and, even though all the aforementioned

TABLE I
TESTING TEMPLATE FOR **DOMOTICS PARAMETERS**, WHICH REPRODUCES A CADENCE TO GENERATE NOTIFICATIONS SIMILAR TO A REAL SITUATION

Parameter	Testing period	Testing template
Ambient temperature	24h	Day-night variation of monitored values in a room. Humidity increases, the temperature decreases in the first part of the day (0:00 - 7:00), the humidity decreases, the temperature increases during the day, then they resume their initial trend for the last part of the day. The brightness increases / decreases at sunrise / sunset, the UV index reaches its maximum value at noon.
Relative humidity		
Luminosity		
UV index		
Vibrations/PIR		Crossing an entrance hall into the house 4 times (once in the morning, 3 times in the afternoon)
Water sensor	4h	Water persistence in the monitored area for 3 hours
Door lock	12h	Unlocking/Locking the door twice in quick succession and with a 5 minute delay
Load switch	12h	Modeling the consumption of a lamp connected to an outlet

TABLE II
TESTING TEMPLATE FOR **VITAL PARAMETERS**, WHICH REPRODUCES A CADENCE TO GENERATE NOTIFICATIONS SIMILAR TO A REAL SITUATION

Parameter	Testing period	Testing template
Blood pressure	12h	Low values in the morning, an increase by 20% in the evening, exceeding the alert threshold
Glycemia	12h	The ”hump” type variation is specific to the three daily meals, with the notification threshold being exceeded
Heart rate	2h	Short episodes exceeding alert values
Temperature	4h	Hypothermia episode with slow decrease and rapid increase in value
Oxygen saturation	2h	Episodes of up to 90% associated with apnea

parameter values are acquired and processes, only the data and graphs that are relevant to emphasize the efficiency of the rules engine are presented for each patient type.

Each time the value of a parameter steps over imposed limits for that parameter, a message class gets triggered depending on the limit, and this can be seen in the considered scenario graphs. The message classes that are represented are N for notification, A for alert and U for urgency. For the sake of brevity, the imposed limits for each level and each parameter are not shown in this paper.

VII. USE-CASE SCENARIOS AND RESULTS

A. Diabetic Patient

For the diabetic type of patient, an episode with a slight increase in blood glucose (i.e., glycemia) above the notification threshold was modeled. There were also increased (and relatively constant) blood pressure values. The other monitored parameters do not correlate with the changed values.

Since the systolic blood pressure of a diabetic is usually higher, in this scenario the patient receives several notifications (marked with N) as can be seen in Fig. 2. Regarding diastolic blood pressure, the patient receives a notification due to too high a value, and 540 minutes after the start of the monitoring, the patient receives an alert because the diastolic blood pressure reaches 90 mmHg (marked with A on the graph), followed by a notification because the pressure drops a little more.

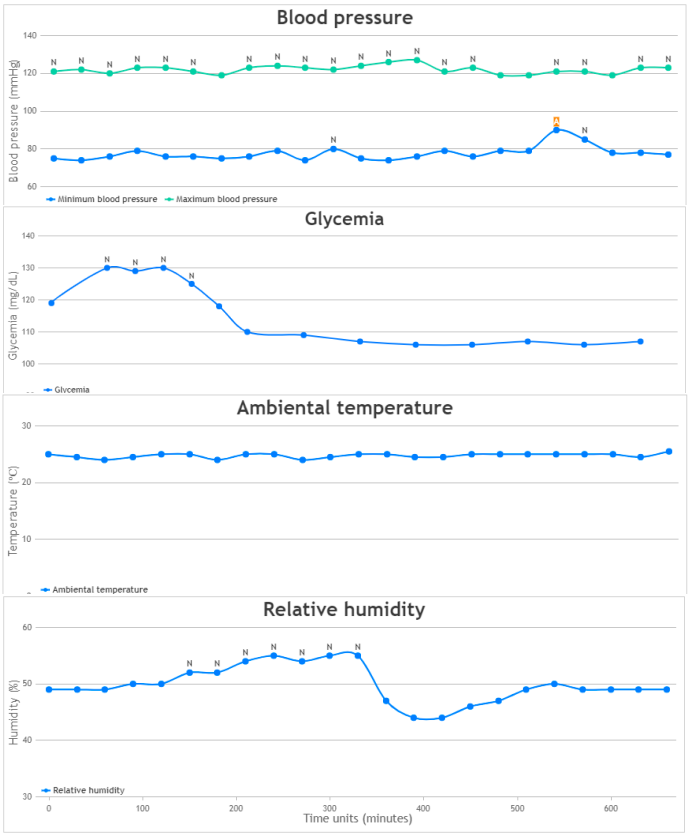


Fig. 2. Diabetic patient - blood pressure, glycemia, ambient temperature, relative humidity

Another monitored parameter is blood glucose. In the first part of the patient's monitoring interval, he receives four notifications because his blood sugar level is above the normal limit. From the point of view of oxygen saturation, it is within normal limits despite the fact that the saturation drops to almost 94% in the first minutes monitored. The ambient temperature in the patient's room is relatively constant around 25 degrees Celsius, so no notifications or alerts are generated. Regarding the relative humidity, this is a slight increase in the first part of the time interval, generating seven notifications until it drops to a normal level.

B. Cardiac/Hypertensive Patient

Testing for cardiac and hypertensive patients was merged due to the similarity of monitoring profiles. Thus, a burst of

hypertension was modeled correlated with an increase in heart rate and a slight decrease in the degree of oxygenation of the blood. After installing the discomfort due to the changed parameters, the patient presses the panic button. The other monitored parameters do not correlate with the changed values. In Fig. 3, a significant increase in systolic and diastolic blood pressure can be seen approximately 9 hours after the start of monitoring. As the recorded value is very high, alert states are preceded and followed by notification states.

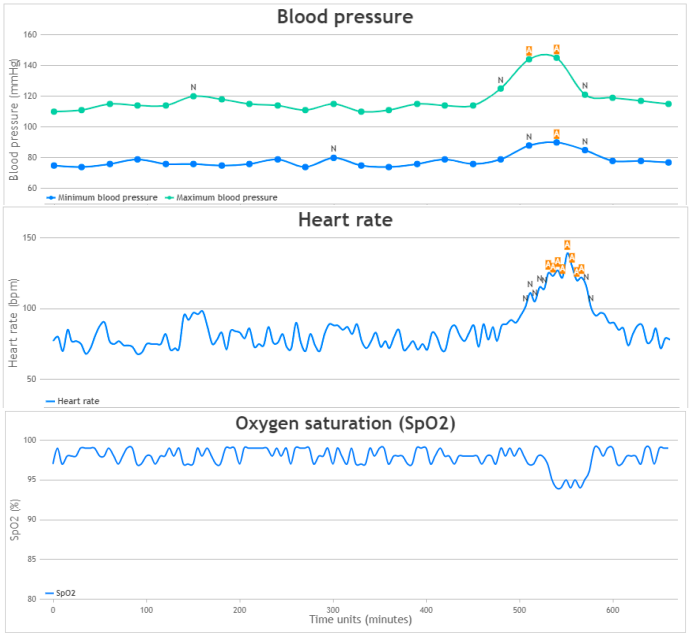


Fig. 3. Cardiac/Hypertensive patient - blood pressure, heart rate, oxygen saturation (SpO2)

High blood pressure is associated with a high heart rate over the same period of time. Thus, several alerts are generated when the heart rate exceeds the notification interval. Oxygen saturation is normal in the first part of the time interval, but falls below 95% at the time of the onset of hypertension, although the saturation does not decrease enough to generate a notification. The ambient temperature is normal throughout the monitoring interval and does not cause discomfort to the patient. From the point of view of relative humidity the patient experiences an increase in this value even before the negative change of vital parameters (blood pressure, heart rate and oxygen saturation). Thus, the patient receives five notifications regarding the increase in humidity above the normal range.

C. Obese Patient

The patient has a heart rate at the upper limit of the range of normal values with frequent exceedances. An episode of high blood sugar is also being reported. The other monitored parameters do not correlate with the changed values. Despite the fact that both systolic and diastolic blood pressure are close to the upper limit of the range with normal values, no notifications are generated because the range has not been exceeded (Fig. 4).

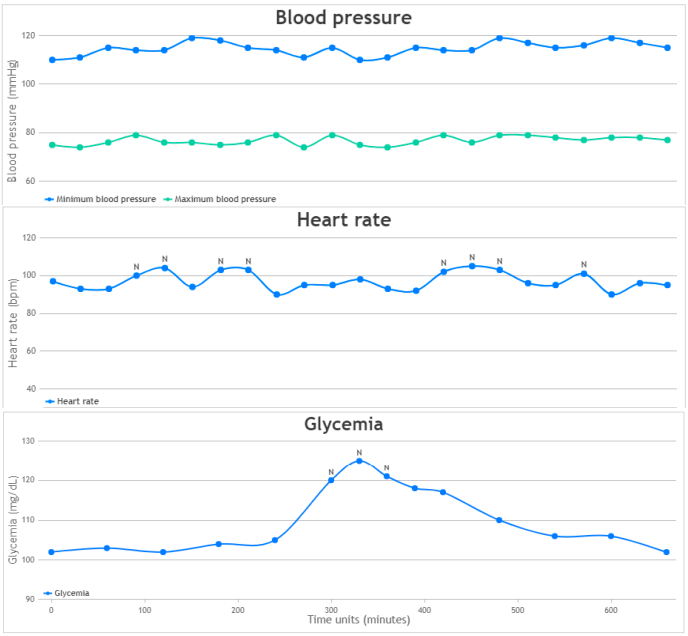


Fig. 4. Obese patient - blood pressure, heart rate, glycemia

In this scenario, the heart rate exceeds the normal limit in some places, and the patient receives a series of notifications of this fact. These increases in heart rate are not large enough to trigger an alert. The patient's blood glucose level is within normal limits in the first part of the time interval, but increases above 120 mg/dL which leads to the patient's notification of this fact. After three notifications, your blood sugar level drops slightly to normal. As in the previous scenarios, the ambient temperature is not a factor that negatively influences the patient and no notifications or alerts are generated.

D. COVID Patient

The patient has a febrile episode exceeding the emergency threshold. An increase in heart rate above the notification threshold is associated with an increase in body temperature. The patient presses the button during the event. Distinctly, a slight decrease in blood oxygen saturation is modeled, without exceeding the notification threshold.

From the first hour after the start of monitoring, the patient's body temperature begins to rise, and the patient is notified of this (Fig. 5). In a short time, two alerts are generated, followed by three states of emergency because the body temperature reaches a critical threshold. After this time, the temperature starts to drop slightly and a series of alerts are generated, followed by notifications until the temperature returns to normal.

During the period when the body temperature is critical, the patient's heart rate is high and notifications are generated from this point of view. Oxygen saturation is in normal parameters in the first part of the monitored interval, given it becomes lower after the episode with high temperature, although it does not decrease enough for the patient to be notified.

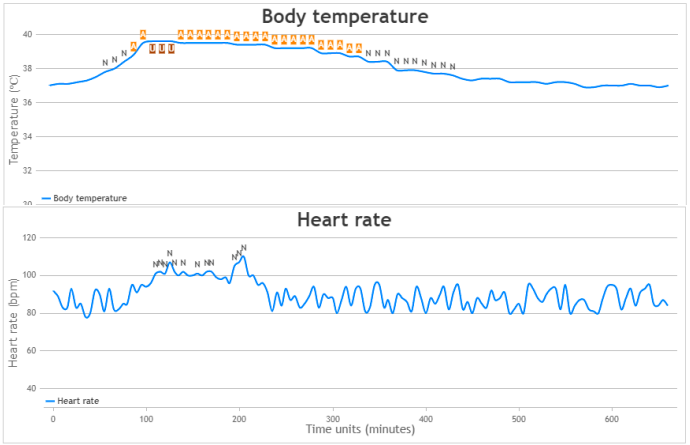


Fig. 5. COVID patient - body temperature, heart rate

E. Alzheimer Patient

In the case of this type of patient, home automation parameters were monitored, with direct actions based on the previously established operating rules. The home automation parameters in this scenario are a lock with automatic shut-off system, the flood monitoring system and the ambient temperature. Three scenarios were generated for the lock: the first in which the patient forgets to lock the door and is notified, the second in which the door locks normally and the third in which it does not react to the notification and the lock is activated automatically (Fig 6).

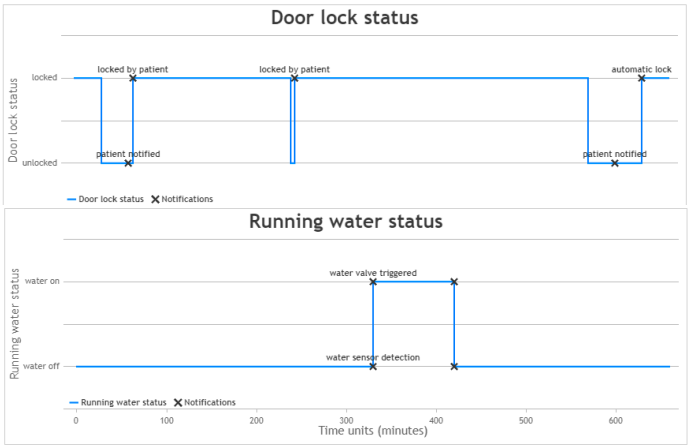


Fig. 6. Alzheimer patient - door lock and running water statuses

An initial notification was generated for flood monitoring, which triggered the shutdown of the tap, and later notifications were generated during the persistence of water in the monitored area. During all this time, the ambient temperature is in normal parameters, and the temperature variations are minimal.

VIII. Conclusion

This research presents a critical part of the SMARTCARE system for palliative care, namely the rules engine, which offers the possibility of setting triggers and corresponding actions depending on the type of patient in assisted living. Another important aspect discussed in this paper is the testing template with the use-case scenarios in which different domotics and vital parameters are monitored depending on the type of considered patient: diabetic, cardiac, hypertensive, obese, COVID and Alzheimer.

The Gateway component of the presented system allows the users to view real time information regarding each monitored parameter, and it allows the setting of rules and, depending of different message classes, signal notifications to the patient, alert the caretaker or urgently initialize the help procedure. As it stands, the latter two message classes are just for proof-of-concept and they are implemented similarly to the notification class. Even though the triggers presented in the use-case scenarios section of this paper refer only to a single parameter for each rule, the way that the rules engine is designed, it allows complex triggers that involve multiple parameters and it also permits a list of actions as consequences of triggering those rules.

Acknowledgment

This work was supported by the PN-III-P2-2.1-PTE-2019-0756 project (SMARTCARE) funded by UEFISCDI under the PTE program (PN-III-P2-2.1-PTE-2019), grant number 42PTE/01.06.2020.

References

[1] S. Balakrishnan, H. Vasudavan, and R. K. Murugesan, "Smart home technologies: A preliminary review," in *Proceedings of the 6th International Conference on Information Technology: IoT and Smart City*, pp. 120–127, 2018.

[2] H. Sun, V. De Florio, N. Gui, and C. Blondia, "Promises and challenges of ambient assisted living systems," in *2009 Sixth International Conference on Information Technology: New Generations*, pp. 1201–1207, Ieee, 2009.

[3] S. Zimmerman, P. Carder, L. Schwartz, J. Silbersack, H. Temkin-Greener, K. S. Thomas, K. Ward, R. Jenkens, L. Jensen, A. C. Johnson, *et al.*, "The imperative to reimagine assisted living," *Journal of the American Medical Directors Association*, vol. 23, no. 2, pp. 225–234, 2022.

[4] S. Zimmerman, P. D. Sloane, P. R. Katz, M. Kunze, K. O'Neil, and B. Resnick, "The need to include assisted living in responding to the covid-19 pandemic," *Journal of the American Medical Directors Association*, vol. 21, no. 5, pp. 572–575, 2020.

[5] S. D. Achirei, O. Zvoristeanu, A. Alexandrescu, N. A. Botezatu, A. Stan, C. Rotariu, R. G. Lupu, and S. Caraiman, "Smartcare: On the design of an iot based solution for assisted living," in *2020 International Conference on e-Health and Bioengineering (EHB)*, pp. 1–4, IEEE, 2020.

[6] S. D. Achirei, O. Zvoristeanu, A. Alexandrescu, N. A. Botezatu, A. Stan, C. Rotariu, R. G. Lupu, and S. Caraiman, "Remote monitoring of physiological parameters used in ambient assistive technologies," in *2020 International Conference on e-Health and Bioengineering (EHB)*, pp. 1–4, IEEE, 2020.

[7] N. Botezatu, A. Alexandrescu, S. Caraiman, F. Ungureanu, and R. Lupu, "Sensing architecture for a nosocomial infection tracing system," in *2021 25th International Conference on System Theory, Control and Computing (ICSTCC)*, pp. 378–383, IEEE, 2021.

[8] G. Marques and R. Pitarma, "An indoor monitoring system for ambient assisted living based on internet of things architecture," *International journal of environmental research and public health*, vol. 13, no. 11, p. 1152, 2016.

[9] R. Zgheib, A. De Nicola, M. L. Villani, E. Conchon, and R. Bastide, "A flexible architecture for cognitive sensing of activities in ambient assisted living," in *2017 IEEE 26th international conference on enabling technologies: infrastructure for collaborative enterprises (WETICE)*, pp. 284–289, IEEE, 2017.

sensors

MDPI

Article

Parallel Processing of Sensor Data in a Distributed Rules Engine Environment through Clustering and Data Flow Reconfiguration

Adrian Alexandrescu

Department of Computer Science and Engineering, Faculty of Automatic Control and Computer Engineering, Gheorghe Asachi Technical University of Iaşi, Str. Prof. dr. doc. Dimitrie Mangeron, nr. 27, 700050 Iași, Romania; adrian.alexandrescu@academic.tuiasi.ro or aalexandrescu@tuiasi.ro

Abstract: An emerging reality is the development of smart buildings and cities, which improve residents' comfort. These environments employ multiple sensor networks, whose data must be acquired and processed in real time by multiple rule engines, which trigger events that enable specific actuators. The problem is how to handle those data in a scalable manner by using multiple processing instances to maximize the system throughput. This paper considers the types of sensors that are used in these scenarios and proposes a model for abstracting the information flow as a weighted dependency graph. Two parallel computing methods are then proposed for obtaining an efficient data flow: a variation of the parallel k-means clustering algorithm and a custom genetic algorithm. Simulation results show that the two proposed flow reconfiguration algorithms reduce the rule processing times and provide an efficient solution for increasing the scalability of the considered environment. Another aspect being discussed is using an open-source cloud solution to manage the system and how to use the two algorithms to increase efficiency. These methods allow for a seamless increase in the number of sensors in the environment by making smart use of the available resources.

Keywords: parallel processing; smart city; sensor; rules engine; k-means clustering; genetic algorithm; sensor network; clustering; cloud computing

Citation: Alexandrescu, A. Parallel Processing of Sensor Data in a Distributed Rules Engine Environment through Clustering and Data Flow Reconfiguration. *Sensors* **2023**, *23*, 1543. https://doi.org/10.3390/s23031543

Academic Editor: Naveen Chilamkurti

Received: 30 December 2022
Revised: 27 January 2023
Accepted: 28 January 2023
Published: 31 January 2023

1. Introduction

As technological development spreads across the world, there is a tendency to automatize and remove the human element from many aspects of everyday life, from automated production lines to solutions based on artificial intelligence such as self-driving cars or generating human language.

When it comes to improving the day-to-day lives of people and their respective communities, the concept of smart-entities emerges, which make use of the Internet of Things (IoT). Examples of entities are homes, buildings or cities. The term IoT started in 1999 and involved interconnected entities by means of radio frequency identification technology. Nowadays, the IoT is characterized by concepts such as wireless sensor networks (WSNs), identifiable devices and actuators, cloud computing or low energy [1]. A wireless sensor network (WSN) represents a network of spatially dispersed sensors that monitor various aspects of the environment in which it is deployed.

The Internet of Things concept is the cornerstone of smart entities. A simple form of smart entity is a smart home, in which basic heating, ventilation and air conditioning (HVAC) elements are automatically managed. HVAC implies the use of different technologies to control heating, ventilation and air conditioning in the home or other buildings. The next step is the smart building [2,3], in which the building's functions are automatically controlled to improve the lives of the residents while being cost and energy efficient. Another step forward is smart cities [4,5], which take the concepts of comfort and efficiency even further by focusing on sustainability, connecting citizens to various public services, managing traffic and providing utilities. There are also spin-offs to these concepts such as

smart villages for developing and introducing smart technologies to rural areas [6] and smart regions [7], which focus on connecting smart cities while having in mind a regional strategy for "promoting smartness". Another aspect of smart entities is the use of artificial intelligence to identify patterns and make decisions.

Regardless of the goal of smart entities, a very high amount of sensors is required. Those sensors produce large volumes of information that need to be handled, in most cases, nearly in real time.

There are several surveys regarding the different types of sensors employed in an IoT environment [8–11]. The main sensor category types that can be deployed in an IoT environment and smart cities include ambient, motion, vital, identification, positioning, entity detection or presence, interaction, acoustic, hydraulic, force or load, vibration or chemical sensors.

A key element of processing sensor data is the use of a rule engine. A rule engine allows the definition of multiple rules that activate when a condition is met. The condition is usually a comparison between values obtained from sensors and other fixed data. Thus, the activation can be triggered depending on if a sensor reading changes. If there are multiple sensors in the system, then there are also multiple rules that handle the data. All this information must be processed as fast as possible.

The current paper focuses on handling large volumes of data from sensors by efficiently using a limited amount of computing resources. The considered environment comprises multiple sensors, sensor networks, gateways, data storage solutions and computing instances. The main goal is to determine a strategy for distributing the sensor data processing among the available computing resources.

The approach proposed in the current paper focuses on processing raw sensor data in a central point with distributed computing capabilities. Multiple gateways can continue to exist in the system, but their role is just to filter noise and forward the data. Two methods are proposed for determining the data flow strategy: one based on the k-means clustering algorithm and a variation of a genetic algorithm.

The main contributions and novelty that the current paper proposes are summarized as follows:

- A sensor data processing architecture in which raw sensor data from a large number of different sensors is efficiently processed in a distributed computing environment;
- An abstraction of sensor data processing by using rule engines, which allows data flow configuration by means of algorithms that analyze communication patterns;
- Two methods for streamlining data flow and improving data processing by creating clusters. where these algorithms process the information by taking into account the number of sensors for each rule and the data volume from each sensor. The two methods are the following:
  - An adaptation of the k-means clustering algorithm;
  - A genetic algorithm with a complex fitness function based on two desired criteria;
- A solution for cloud deployment while ensuring scalability and adaptation to the sensor rule particularities of the system.

The remainder of this paper is organized as follows. Section 2 discusses how the proposed architecture and the two proposed methods compare to existing solutions for the considered topic. Section 3 presents the considered context as well as the proposed goal, the rules engine, the sensor rule abstraction, the performance metrics that were used, the two proposed clustering methods and the solution to deploying the system for a cloud solution. Section 4 describes the simulation set-up and the experimental results. Finally, Section 5 showcases the importance to the proposed methods and how they perform when considering various other aspects of the environment. Another discussion in this section is related to scalability and managing a dynamic environment.

2. Related Work

The related work presented in this section is concentrated on the main aspects of the proposed research: rule engines, sensor networks, data flow reconfiguration and clustering.

Existing rule engines focus on the representation of the facts that are used in handling sensor information. This is performed as a way to increase the efficiency of processing a large amount of sensor data. In [12], the authors extracted atomic events from sensor data and proposed a scheme that minimized the rule-matching overhead. Another related piece of research is [13], where the authors proposed a rule engine which allowed flexible rule strategies. In terms of smart rules engines, in [14,15], the authors presented a solution based on fuzzy logic for processing information. The research presented in the current paper uses the rule engine from [16], mainly due to its simplicity. The way the proposed system is designed, it allows any of the aforementioned rule engines to be used as long as the time it takes to process the sensor data is directly dependent on the volume of received sensor data.

There is existing research regarding various aspects of the considered environment, but none of it focuses on optimizing the data flow in the system in both a centralized manner (by pooling data from all of the sensors) and a decentralized manner (by distributing the processing to multiple instances). The advantage is that multiple gateways can be used to communicate with specific sensor networks, and they each can provide local processing while providing the results up the hierarchical chain, perhaps to a central cloud. This method has the disadvantage of requiring computational capabilities for each gateway, and some relevant data may also be overlooked if they are processed locally. Those data may be relevant in the global scope.

In terms of environments with multiple connecting gateway systems and wireless sensor networks, the current research topics cover a wide area. An example of handling communication in hierarchical wireless sensor networks with multiple gateways is presented in [17]. However, in that research, there is a method proposed for solving the authentication problem that is not focused on improving information flow but rather on securing access to resources.

Another aspect which is discussed in the current paper is related to data flow reconfiguration. Although the main focus is obtaining a very good configuration from the start, Section 5 of this paper tackles the dynamic reconfiguration scenario. In terms of network configuration, the authors of [18] proposed a platform for fast sensor network prototyping. Regarding dynamic network reconfiguration, there is existing research in this field [19] and also comparisons between existing methods [20,21]. None of that research can be directly applied to the considered environment of multiple rule engines that have to process the sensor data efficiently.

Much research regarding the considered environment focuses on improving energy efficiency. In [22], two methods were proposed for optimizing energy consumption in a WSN for monitoring a smart city. Even though the k-means clustering algorithm was employed, the features used for the algorithm and the overall scope differ from the research proposed in the current paper. Another paper involving clustering, gateways and sensors used in a different scope is [23]. The authors proposed that there is a solution for improving energy consumption in a wireless body area network used for patient monitoring. The used clustering method selects cluster heads based on the residual energy of the cluster nodes. Other papers used hierarchical clustering [24,25] and other techniques [26] to determine the optimal number of clusters for minimizing energy consumption. While those papers used clustering to achieve their proposed goals, none of them focused on the particular problem considered in the current paper.

The research in [27] used k-means clustering in order to enhance privacy for data clustering of information from an IoT environment. The idea was centered on improving privacy and not on increasing the efficiency, as is the case in this proposed research. Another paper involving clustering in WSNs is [28], but there the authors considered multi-hop routing trees and reduced the total energy consumption. In [29], the authors focused

again on energy consumption and the distribution of cluster heads. These papers had more features available for consideration through the k-means algorithm, which led to an increase in the efficiency of using that algorithm in solving a specific problem. Other related research includes parallel clustering on low-power devices in edge computing environments [30] or methods for delivering sensor data and storing them in an IoT environment [31,32].

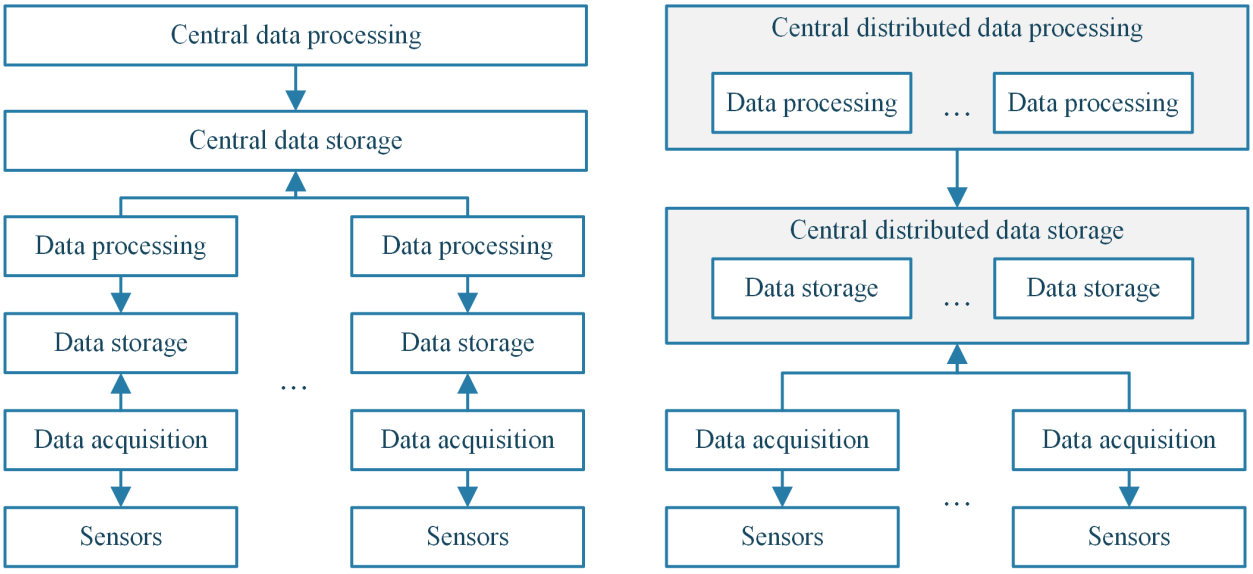
3. Proposed Architecture and Methods

3.1. Considered Context and Proposed Goal

Smart buildings and smart cities employ thousands and even tens of thousands of sensors to monitor various parameters and to interact with the environment based on specific rules. The data from those sensors need to be processed nearly in real time.

Three issues arise: how to obtain the data from the sensors, where to store them and how to properly process said data. In terms of obtaining data from sensors, a good approach is the one from [33], in which there are bridge and driver processes that gather data from one or more sensors belonging to a specific communication protocol (e.g., Z-Wave, ZigBee or other standardized APIs) and filter data noise. Storing the data can be accomplished in two main ways: message-queuing services (e.g., MQTT broker, RabbitMQ or a cloud queue solution) or databases (e.g., a data store or data lake, usually stored in a public or private cloud). Regardless of the storage solution, the sensor information must be efficiently and swiftly processed in order to take specific actions, or large amounts of data can be processed to find certain patterns. Either way, there is the need for a gateway component to use the acquired data and perform some computations and processing.

The aforementioned approaches work well for smart home and smart building systems, but for smart cities and smart regions, the incoming data need some sort of distributed computing in order to be properly processed. Figure 1 shows two approaches for handling the data flow and processing.



(a) Local processing and central aggregation of results (b) Central distributed storage and processing

Figure 1. Two architectures for processing sensor data. (**a**) The sensor acquisition, storage and processing are performed locally, and the processed results are sent to the central data storage component, which is accessed by the central data-processing component for further data analysis. (**b**) The components that perform the data acquisition store the information in central distributed data storage, which is accessed by distributed data-processing components.

The first architecture (Figure 1a) is the traditional approach, in which local gateways process the information from sensors. That information is used to make decisions and, from

time to time, to send relevant data to a central node or gateway for storage and further processing. This a good approach, but it has two disadvantages. First, the central node does not have access to all the local data, and it can be less efficient at finding patterns. Secondly, each gateway is an extra computing component and overhead to the entire system. On the other hand, it makes quick local decisions, but without the overall picture, those decisions might not be the most efficient ones.

The second architecture (Figure 1b) starts from the premise that all the sensor data are directly sent to the central distributed storage (e.g., a distributed message queue service or massive distributed data store). Having a central distributed data-processing component allows having an overview of the entire environment and triggering local decisions based on the general context. Another advantage is having fewer overall data processing instances, because one instance can process data from multiple and various sources. The research presented in this paper focuses on the second architecture model, and the two proposed solutions aim to improve the data processing performed by each instance.

3.2. Data Processing and Rule Engines

Processing sensor data consists of having data from specific different sensors as input, computing those data by some means and triggering one or more actions if needed.

Basic processing can be achieved by employing rule engines [16]. Each piece of data passes though the rule engine, which looks at all the encompassing rules, determines which ones trigger (if any) and takes the associated action(s). The rules can be described in JavaScript Object Notation (JSON) format as shown in Listing 1. JSON is a heavily used text-based method of representing structured data in a both human- and machine-readable format. In this example for a home room, a rule is set to trigger turning on the air conditioning unit in dehumidify mode if the humidity in the room is over 55% and the temperature is over 26 degrees Celsius. The condition for this rule is checked each time the humidity or temperature in the room changes. Slight variations will not trigger the rule because one of the roles of the sensor data acquisition component is to send the new data only if the change in value is above a certain increment.

Listing 1. Example of a rule for HVAC in JSON string format based on the rule engine described in [16].

```
{
  "name": "humidity-control-on",
  "description": "If the humidity in the room is over 55% and the temperature is over 26 degrees Celsius,
      ↪ then start the AC and set it to dehumidify",
  "priority": 3,
  "condition": "${home-room-1/hvac/humidity}{value} > 55 and ${1/hvac/temperature}{value} > 26",
  "actions": [
    "${home-room-1/hvac/ac}{value}{on}",
    "${home-room-1/hvac/ac}{mode}{dehumidify}"
  ]
}
```

The above example is a trivial one, but the rule engine can be used for complex rules that depend on different types of sensors and even on other rules. Anything that has a unique identifier in the system (e.g., home-room-1/hvac/humidity) can be referenced in both the condition and the action properties of the rule. The rules are not exclusive to HVAC or home monitoring and can be employed for monitoring traffic, for automatic street lightning or even for triggering facial recognition algorithms when using CCTV. The latter requires a component that processes the images, filters them and, from the rule engine point of view, acts as a data-producing sensor. Nonetheless, this is beyond the scope of the current paper.

To summarize, the considered architecture employs a rule engine for each computing instance, and each rule engine manages a set of rules whose conditions need to be checked, depending on the incoming sensor data.

3.3. Sensor: Rule Abstraction

A high amount of computational power is required when having a large amount of sensors which produce data and many rules that need to be checked, depending on the received data. As previously established, the proposed solution is to distribute the load among computing instances. Therefore, each instance will handle a distinct set of rules, and each rule needs the data from specific sensors. It does not make sense to check the rules from all the instances once a new piece of sensor data enters the system; rather, only the rules that involve that sensor should be checked. Only the instances with the involved rules will ever receive data from that sensor.

A rule can be seen as a generic check if certain conditions are met, and if that conditional expression is true, then one or more actions are triggered. This concept can be extended to include the possibility of triggering more complex processing if specific sensor values satisfy certain conditions. Each sensor produces, by means of the data acquisition component, a variable amount of data depending on the sensor type and the environment conditions. The system can be monitored in order to determine the approximate amount of produced data for each sensor in a time period (e.g., hour, day, month and year).

Regardless of the sensor type and the acquired data format, the information that gets transmitted by means of WiFi, Bluetooth or other methods is basically an array of bytes. In order to abstract the volume of the acquired and transmitted sensor data, the data quantities are in generic data units (d.u.). These units represent the amount of information from one sensor during a specific period of time.

An example of connections between sensors and rules is presented in Figure 2. There are six sensors (S_1–S_6) and four rules (R_1–R_4), and each edge represents the volume of information that gets sent from a sensor to a rule. The volume of information is a numerical value expressed in generic data units (d.u.). In a real scenario, this value represents the number of bytes sent from a sensor to a rule. Each rule needs data from specific sensors (e.g., rule R_1 requires 12 d.u. from S_1 and 10 d.u. from S_2). Therefore, rules need data from multiple sensors, and sensors send data to multiple rules.

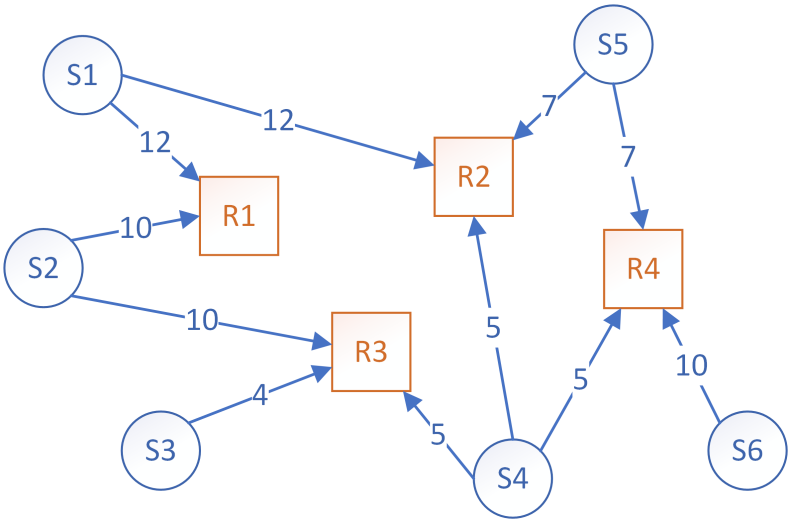


Figure 2. Example of the connections between six sensors (S) and four rules (R). Each sensor, represented by the round vertices, produces the amount of data specified on the graph edge, and those data have to be processed by the rules, represented by square vertices.

For simplicity's sake, the scenario considered in the proposed paper implies that rules cannot send data to other rules, but this can easily be made possible if certain rules are set to also act as sensors (i.e., produce data). For each rule that produces data, a virtual sensor would be created in the system to simulate this. If a cycle is created in the resulting graph by considering this scenario, then the clusterization approaches proposed in this paper are still valid as long as the volume of information being sent from one rule to another is

quantifiable. By knowing how many data are being sent in this case prior to running the algorithms, the proposed solutions work perfectly.

As mentioned earlier, the problem statement is that there is a limited amount of computing instances available. Each computing instance must process a group of rules so that the information that is being sent to that instance is minimal. Therefore, the goal is to somehow group the rules into clusters so that the information is being sent efficiently to the clusters. Each cluster of rules is then the responsibility of a single computing instance.

Figures 3 and 4 show two of the possible different clusters that can be formed. Both figures show two clusters, but each cluster has a different composition and, consequently, will have a different amount of data volume that needs to be processed. In the first case (Figure 3), cluster C_1 encompasses rules R_1 and R_3 and receives a total of 31 d.u. from sensors S_1, S_2, S_3 and S_4, while cluster C_2 encompasses rules R_2 and R_4 and receives a total of 27 d.u. from sensors S_1, S_4, S_5 and S_6. In the second case (Figure 4), cluster C_1 encompasses rules R_1, R_2 and R_3 and receives a total of 38 d.u. from sensors S_1, S_2, S_3, S_4 and S_5, while cluster C_2 encompasses rules R_4, S_5 and R_6 and receives a total of 22 d.u. from sensors S_4, S_5 and S_6.

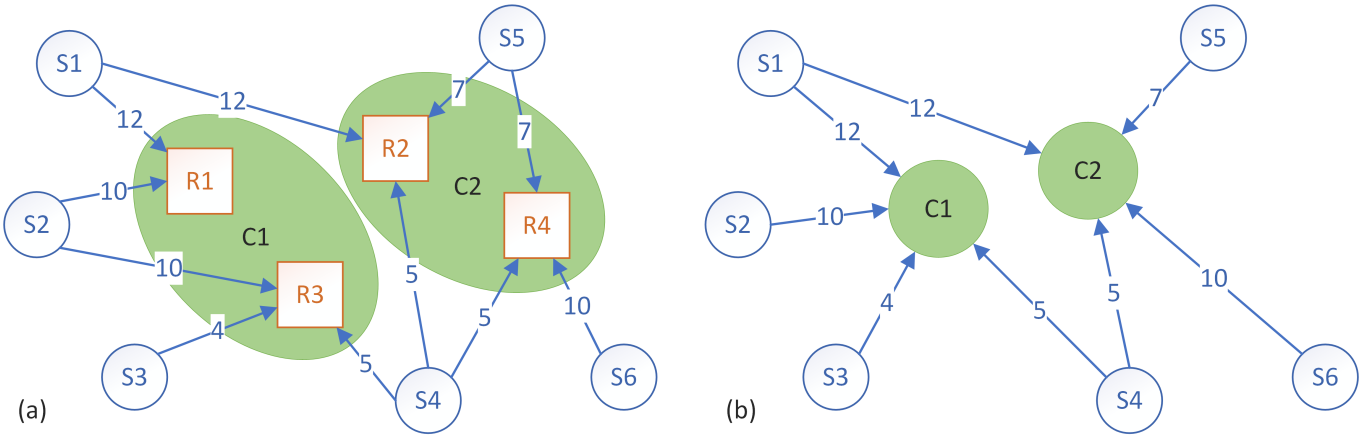


Figure 3. Example of clustering rules for the graph from Figure 2. (**a**) There are two clusters: C_1 with rules R_1 and R_3 and cluster C_2 with rules R_2 and R_4. (**b**) The formed clusters receive the data only once from the connected sensors (S).

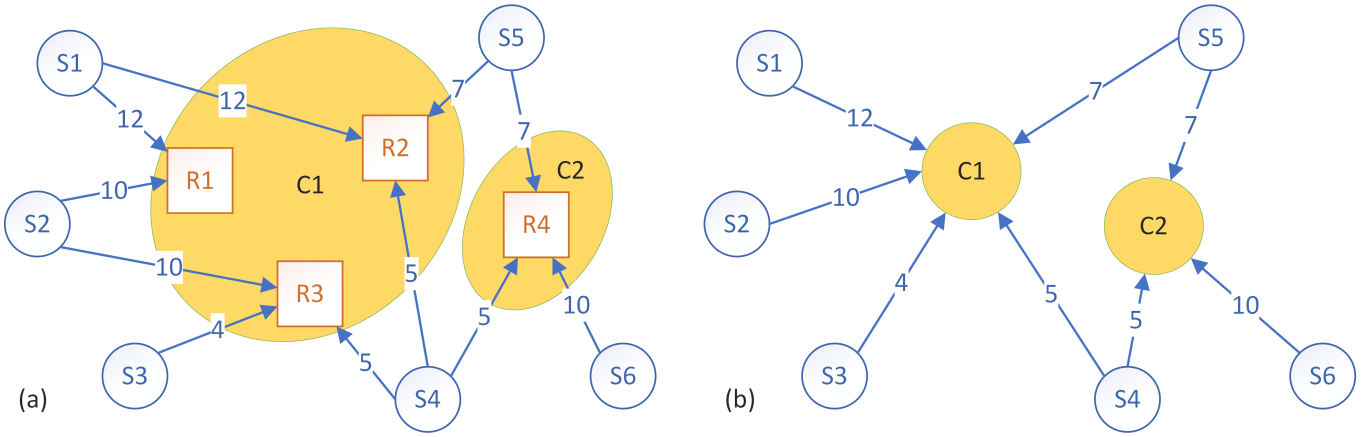


Figure 4. Example of clustering rules for the graph from Figure 2. (**a**) There are two clusters: C_1 with rules R_1, R_2 and R_3 and cluster C_2 with rule R_4. (**b**) The formed clusters receive the data from the connected sensors (S) only once.

When a sensor is removed from the system, the corresponding rules need to be updated to account for this. Removing just a single sensor in the context of tens of thousands of sensors will have an insignificant impact on the system's performance. However, if

multiple sensor networks are removed, then one of the two proposed methods has to run to reconfigure the data flow. When a single sensor and the associated rules are added to the system, those rules can be assigned to the cluster with the smallest load. When adding multiple sensor networks to the system, the clustering process needs to trigger again. As it stands, the clustering method might perform a complete reconfiguration, which might trigger multiple rule migrations from one instance to another. Depending on the complexity of the migration, this might not be desired. On the other hand, this massive migration might provide a much better processing time in the long run. Therefore, when considering the dynamism of the proposed system, the scalability is highly dependent on the moments when the reconfiguration is triggered and on the impact of the rule migrations. The best way to achieve high scalability is achieved in two steps. First, when sensors are added to the system, they are assigned to the least loaded instance. Secondly, a job scheduler needs to be set up so that the clustering algorithms periodically reconfigure the data flow. The reconfiguration implies transferring rules in JSON format from one instance to another, which can be performed in a timely manner. The sensors need to know to which instance they should send the data. This problem can be easily solved by using a gateway or load balancer. The sensors send the data to the gateway instance, which then redirects it to the corresponding instance.

The way that the clusters are formed can have a significant impact on the performance of the system. Therefore, performance metrics have to be set in order to quantify the quality of the clustering.

3.4. Performance Metrics

The main performance indicator is the speed with which the computing instances process the data. To keep the issue as simple and clear as possible, this research assumes that the time it takes to process data units from each sensor is directly proportional to the amount of data units that the instance receives. If, in reality, this is not the case, then the data unit value from each sensor can be multiplied by a factor depending on each specific sensor. Therefore, the problem becomes reduced to the equivalent of having normalized the volume of information and the time it takes to process it to generic units (i.e., data units), as is discussed further in this paper. The problem could be further reduced to a task mapping problem, but the situation is different from the standard approach because of the extra layer introduced by having rules. Another reason for this is because sensors should not send data to instances regardless of what rules are processing the information.

In terms of performance metrics, the goal is to process information as fast as possible. More data to be processed equals more time required to handle those data. When looking at the considered problem as a static task scheduling issue, where a task is considered to be the processing of a piece of data, the goal is to minimize the makespan (i.e., the time it takes for all instances to finish processing the assigned tasks) [34]. Because the environment is a dynamic one, the quality of the solution to the considered problem cannot be accurately evaluated. Using the makespan is a good performance metric in a static task allocation that uses no heuristics, but it has the disadvantage that certain instances are not used to their full potential. The makespan value is given by the instance that finishes the allocated tasks last, and there can be instances that finish much earlier.

Given the nature of the dynamic environment, static task allocation methods cannot be used, and other heuristics must be employed. The makespan is still useful in this case, but a better approach may be to consider balancing the load so both metrics must be taken into account. On the one hand, the goal is to finish processing as fast as possible, and on the other hand, the goal is to keep the load balanced among the instances.

Considering the example from Figure 3, the makespan was 31 d.u., and the load imbalance (i.e., the difference between the instance that finished latest and the instance that finished earliest) was 4 d.u. For the example in Figure 4, the makespan was 38 d.u., and the load imbalance was 16 d.u. Low values of both the makespan and the load imbalance are desired so the first clustering example is better from both points of view. Intuitively, the

makespan is a good enough metric, but tests have shown that using the load balance as a criterion for certain algorithms can provide good results as well.

3.5. Problem and Solution Representation

The problem that needs to be solved is how to cluster the rules together so that each cluster requires the least amount of data to pass through the associated rules. The two proposed algorithms that solve this problem receive as input a graph similar to the one from Figure 2, represented by a two-dimensional array (A). The rows represent n sensors (S), the columns represent m rules R, and A_{ij} represents the volume of data in data units (d.u.) which the sensor S_i sends to rule R_j for processing. Therefore, the goal is to find p clusters C, where $C_k \subset R_j, j \in \{1..m\}$ and $\forall k \in \{1..p\}, \forall l \in \{1..p\}, k \neq l$, $\nexists C_k \subset R_j$ and $C_l \subset R_j, \forall j \in \{1..m\}$.

The solution consists of a list of rules that are associated to each cluster. A rule can be assigned to only one cluster. A simple representation is using an array in which each position is associated with a rule. The value at that position represents the cluster number to which that rule belongs.

The available solution space for the considered problem is very large. The number of possible solutions is p^m, because each of the m rules can be associated to one of the p clusters. For example, if there are 10,000 rules and 36 clusters, then it would take more than a lifetime to generate all the solutions. Each time a solution is generated, it has to be evaluated, and this implies many computations. All those computations also depend on the number of sensors in the system. It is impossible to develop an algorithm that offers the best solution in a timely manner. Genetic algorithms are a good method for solving problems in which good enough solutions need to be obtained in a decent amount of time while looking at a large solution space. A drawback of genetic algorithms is the execution time. Methods that obtain faster results are traditional clustering algorithms. Clustering algorithms usually require that the metric is some sort of distance that has to be computed. The characteristics of the considered environment do not make that distance straightforward to compute. Therefore, the method that was chosen in this paper was k-means clustering because it is widely used in existing research, and its simpler form allows an easier adaptation for solving the problem.

The criteria used for grouping the rules into clusters depend on the approach. For the k-means algorithm, the parameters that can be considered are limited due to the nature of the problem. Because the way that the rule engine processes the data is considered a black box, all that remains is focusing on the data that are being transmitted. The two parameters that are taken into consideration are the number of pieces of data that are being sent by each sensor and the total volume of information that is being transmitted by a sensor to each rule. When using the genetic algorithm approach, the focus is on the end goals of minimizing the makespan and the load imbalance. This method generates multiple solutions and tweaks the resulting chromosome. Therefore, the fitness function considers the data volumes and determines the quality of a solution.

3.6. K-Means Clustering Approach

The k-means clustering algorithm is a good approach for grouping together items that have similar features. The basic steps of the algorithm adapted to the considered problem are as follows:

1. Choose the number of desired clusters (p).
2. Randomly choose p centroids (each one represents the center of the cluster).
3. For each rule, perform the following steps:
    - Compute the distance to each centroid.
    - Find the closest centroid.
    - Assign the rule to the cluster corresponding to that centroid.
4. For each cluster, perform the following step:

- Compute the new centroids as the mean of all the rule features assigned to that cluster.

5. Go to step 3 until there are no changes compared to the previous iteration or a specific number of iterations has passed.

Usually, this method uses the Euclidean distance for determining the closest centroid, but there are similar approaches available [35]. Regardless of the method for computing the distance, certain numerically quantifiable features need to be considered. Due to the complexity of the considered problem, modeling the input data to be suitable for the k-means method proves quite difficult. After considering a few variations, the best results were obtained by considering for each rule two features: the number of sensors that send data to that rule and the total volume of data that is being sent to that rule by all the sensors. These two features are used as metrics when computing the distance between each rule and each centroid.

This method of clustering does not help in providing a low makespan or even a balanced load; instead, it helps group together rules with similar features. To obtain a fairly well-balanced load, a second part was added to the proposed solution which split the clusters based on their size. The cluster size is represented by the total data units that are required by the encompassing rules. The novel proposed k-means-based algorithm is as follows:

1. Choose a much lower number (e.g., $q = 4$) than the number of desired clusters (p).
2. Create q clusters with the previously described k-means algorithm.
3. Compute each cluster's size.
4. Split evenly the larger clusters so that each resulting cluster is around the same size.
5. Recompute each cluster's size.
6. Split the clusters again or combine clusters depending on the desired number of clusters while recomputing each cluster's size if the cluster composition changed.

The advantages of this approach are that it is fast, it can be easily parallelized [36] and it provides a better alternative to assigning rules to clusters randomly. The downside of this method is that it may prove to be difficult to obtain a specific number of clusters. This approach is better suited if there is not a fixed number of clusters to be obtained.

3.7. Genetic Algorithm Approach

Genetic algorithms provide good solutions in a very large solution search space at the expense of possibly not finding the best solution but rather a good enough one. There are two key elements when it comes to the genetic algorithm (GA): the candidate solution representation and the fitness evaluation function. The GA starts with a population made out of individuals (chromosomes), and each chromosome represents a candidate solution. The proposed solution representation is straightforward. Each gene represents a specific cluster while its locus represents the rule, and the alleles are the possible clusters that can encompass the rules. There are two fitness functions being considered: the makespan and the load balance, as discussed in Section 3.4. The proposed genetic algorithm solution with the chosen parameter values and functions is as follows:

1. Use the input data and initialize the GA parameters, such as in the following example:
   - Population size: 200.
   - Crossover probability: 0.7.
   - Mutation probability: 0.5.
   - Elitism percent: 0.05.
   - Maximum iteration count: 1000.
   - Maximum "no change" iteration count: 300.
2. Randomly generate the initial population.
3. Evaluate the population using the custom fitness evaluator.
4. While the stop condition is not met, perform the following steps:

(a) Generate a new population of the same size by going through the following steps multiple times:
- Select two chromosomes using roulette wheel selection.
- Possibly a apply one-point crossover to each selected chromosome.
- Possibly apply a random allele switch mutation to each selected chromosome.
- Add the two resulting chromosomes to the new population.

(b) Apply elitism, under which the best chromosomes from the previous population are kept in the new population.

(c) Compute the fitness of each chromosome from the new population.

5. Output the solution represented by the chromosome with the best fitness.

All the chosen parameter values and all the chosen functions (i.e., selection, crossover and mutation) are based on the existing literature regarding this topic [37,38] and on previous experiments with genetic algorithms performed by the author [39–41]. There is one exception regarding the mutation probability parameter, which is usually lower. Tests have shown that for the considered problem, a higher mutation probability yields better results.

The complexity of the proposed genetic algorithm is given by the fitness evaluation function, which has to take into account the sensor data needed for each created cluster. This is somewhat computationally intensive, and a caching mechanism is used so as to not recompute the fitness of previously evaluated chromosomes. Overall, the proposed method has the advantage of providing a better solution than the k-means-based method at the expense of a higher execution time and increased difficulty of achieving parallelization of the execution.

3.8. Deploying the System in a Cloud Solution

Another proposed concept is a generic framework for allowing the deployment of the described system architecture in a cloud solution (i.e., the open-source OpenStack solution). There is similar research in this area, namely the Stack4Things solution presented in [42], but it follows the standard architecture of an OpenStack service, and it uses the cloud IaaS services to provide the basic required functionality for an IoT solution.

An interesting approach is to deploy the proposed algorithms on cloud instances and use that instance to coordinate sensor data and instances as presented in Figure 5. The instances can be managed using the Nova instance controller. Depending on the output of the algorithms, the controller can automatically provision new instances depending on the formed clusters, which handle the rules. The sensor information can be saved with block and object storage using the Cinder and Swift services, while advanced orchestration and easy redeployment can be obtained using the Heat service. The cluster manager instance, which employs the clustering algorithms, can configure the custom sensor data load balancer to store more efficiently the sensor data based on the clusters (i.e., Nova instances) that require those data.

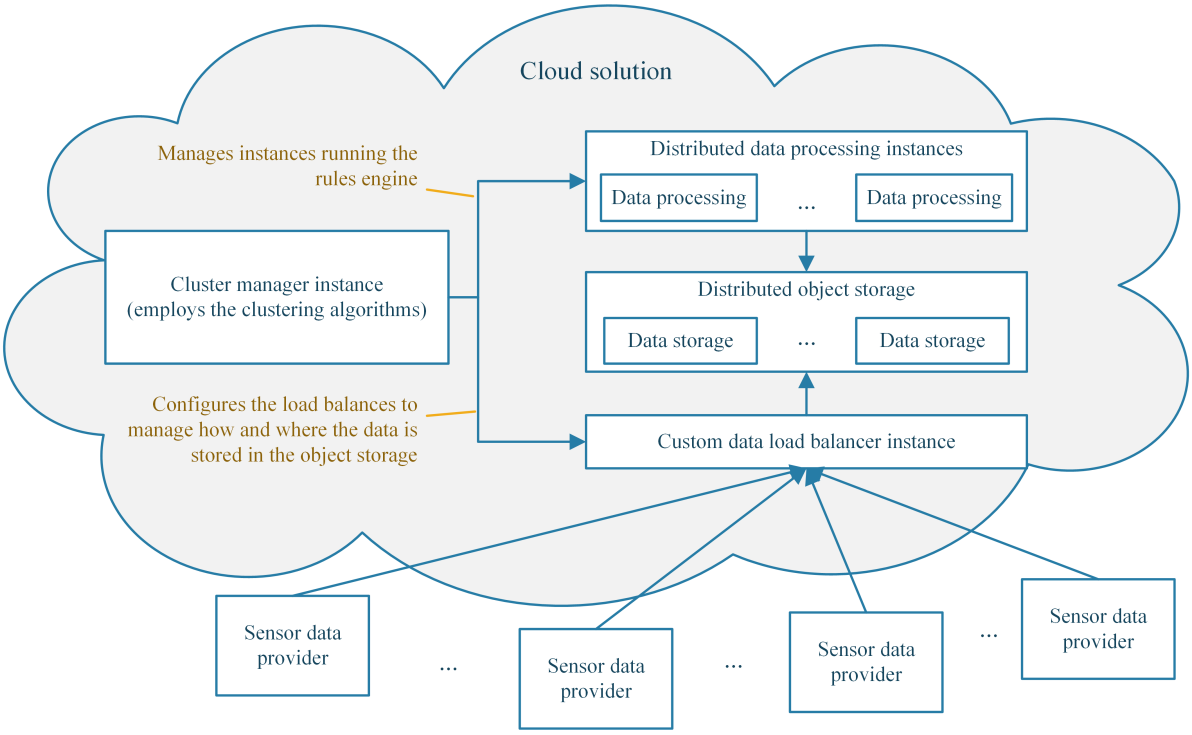


Figure 5. Proposed cloud solution architecture.

4. Results

4.1. Simulation Set-Up

In order to test the efficacy of the proposed solution, a simulation framework was developed, and the two proposed methods were implemented and compared. The framework makes use of three parameters: the number of sensors, number of rules and number of desired clusters. Multiple tests were performed with various combinations of the three parameters and with different seed values for the random value generator function. The selected parameters and data presented in this paper represent a test case in which the obtained results are in the middle range of performance metric values. For most of the performed tests, the result variation for each metric was $\pm 4\%$ compared with the presented results.

The considered test simulated 12,000 sensors with 6000 rules, and the goal was to generate 36 clusters with those rules. Each sensor produced a generated number of data units, which was a random value with a Gaussian distribution. The formula for generating the value was $500 + 300 * g$, where g is a pseudo-random Gaussian distributed number with a mean of 0.0 and a standard deviation of 1.0. The range for the volume of data produced by a sensor was between 1 and around 800 d.u. Each rule received data from between 1 and 11 sensors, as shown in Figure 6.

Because this is not a straightforward approach for applying an algorithm, Figures 7 and 8 provide more context and offer a perspective on how the information is generated and associated to sensors and rules. The distribution of data units among sensors and the sensor-rule association were performed in order to simulate a real sensor environment as closely as possible.

Figure 7 presents the computed data volume required by a rule, depending on the number of sensors associated with that rule. For example, there was a rule that had 11 sensors associated with it, and it required a total of 5630 d.u. to process it. In Figure 8, the number of rules that need to process different ranges of data volumes is shown (e.g., 356 rules need to process a data volume in the range (1146, 1336] d.u.).

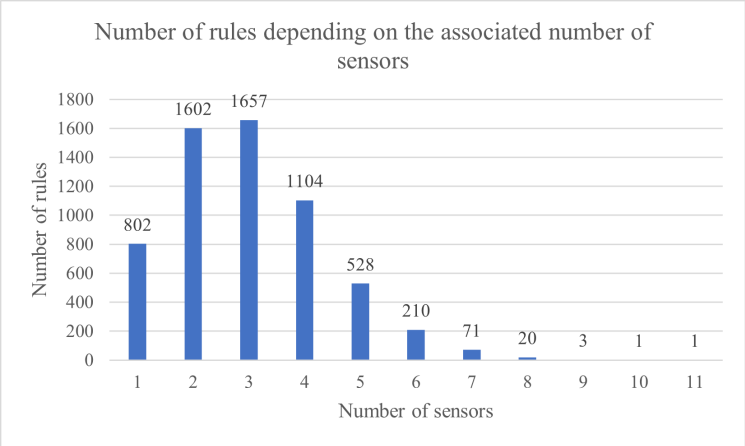


Figure 6. Number of rules depending on the associated number of sensors.

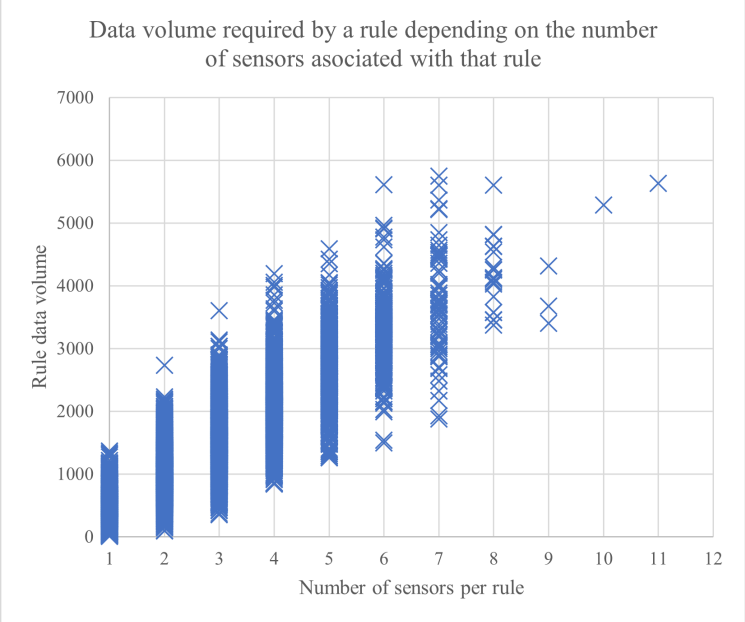


Figure 7. The data volume required by a rule, depending on the number of sensors associated with that rule.

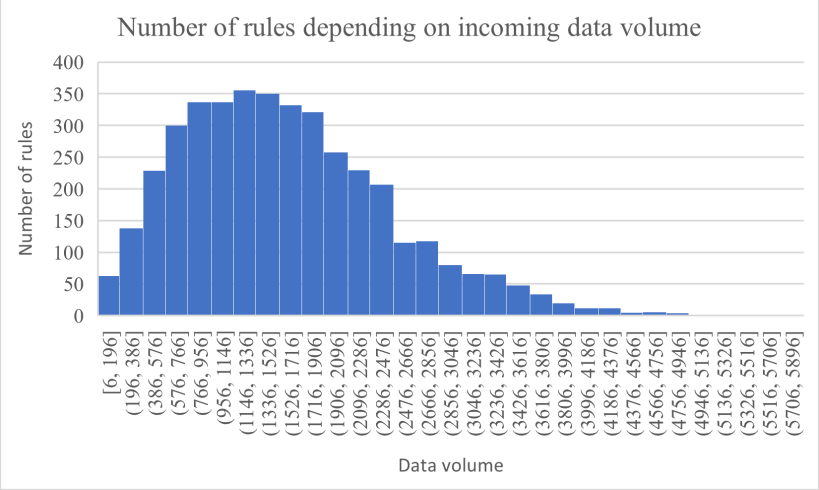


Figure 8. The number of rules whose incoming data from the sensors are in a specific range.

4.2. Experimental Results

Each of the proposed algorithms was executed on the same generated test case, and the results were compared depending on the data volumes that needed to be processed by each cluster of rules. In the evaluation process, we used the two performance metrics described in Section 3.4. The goal was to have a low makespan and a low load imbalance.

The k-means-based algorithm was configured to initially generate only four clusters. Because the features used by the k-means method were the number of sensors and the data volume processed by each rule, each of the four clusters had grouped together rules with similar features. The total data volume required to be processed by each initial cluster was 1,645,684 d.u., 2,887,941, d.u., 2,738,362 d.u. and 1,350,257 d.u. As can be observed, the two middle clusters had to process around twice as many data compared with the other two. The first operation performed on the clusters by the k-means-based algorithm was to split the middle clusters, resulting in a total of six clusters with data volumes of 1,645,684 d.u., 1,531,453 d.u., 1,520,189 d.u., 1,350,257 d.u., 1,470,455 d.u. and 1,460,521 d.u. Each of these six clusters was split into six parts to reach the desired 36 clusters. The minimum data volume that needed to be processed by a cluster was 240,164 d.u., and the maximum (i.e., the makespan) was 289,776 d.u., resulting in a load imbalance of 49,612 d.u., which represented 17% of the makespan. This might not seem like a good result, but considering that traditional task scheduling could not be applied given the sensor-rule constraints, the proposed method was very fast and provided an adequate solution.

For an initial comparison, there were 200 random solutions generated, and the best one among them was selected by evaluating it with the fitness function of the genetic algorithm. When using the makespan as a fitness criterion, the best solution had a minimum data volume of 219,256 d.u. and a maximum of 290,519 d.u., resulting in a load imbalance of 71,263 d.u., which represented 24% of the makespan. When using the load balance as a fitness criterion, the best solution had a minimum data volume of 235,601 d.u. and a maximum of 300,391 d.u., resulting in a load imbalance of 64,790 d.u., which represented 21% of the makespan. The proposed k-means-based algorithm performed better compared with the two solutions selected among 200 random solutions using the two performance metrics. It also ran in a fraction of the time compared with the time it took to generate and evaluate 200 candidate solutions.

The tested genetic algorithm used the parameters and functions described in Section 3.7. The formula for computing the fitness depends on the number of sensors associated with each rule, the range of d.u. generated by the sensors and the considered criterion (i.e., the makespan or the load balance). Let min be the minimum data volume that needs to be processed by a cluster and max be the maximum data volume that needs to be processed by a cluster. For the presented test case, the chromosome fitness when trying to minimize the makespan was computed as $10^8/(max+1)$, while the chromosome fitness when trying to balance the load was computed as $10^6/((max-min)+1)$. This was carried out to obtain more human-readable values for the fitness. The evolution of the results obtained by the genetic algorithm using the makespan for the fitness is presented in Figure 9, and the ones using the load balance are shown in Figure 10. For both graphs, higher values mean a better solution because the fitness is inversely proportionate with the makespan and the load imbalance. In order to better compare the results, the fitness of the k-means-based algorithm and the average fitness obtained from the initial population of the genetic algorithm were added to the charts. The average population fitness was determined by generating the initial population for the genetic algorithm and then calculating the fitness of each candidate solution. Finally, the average of the fitness values was calculated, and that resulting value represented the average population fitness.

The obtained fitness values are shown in Table 1. When trying to improve the makespan, the proposed k-means-based algorithm was 8% more efficient than the average population fitness, while the proposed genetic algorithm provided an efficiency increase of 17%. Regarding improving the load balance, the differences were more signifi-

cant. The k-means-based approach was almost twice as good compared with the average population fitness, while the genetic algorithm was more than 17 times better.

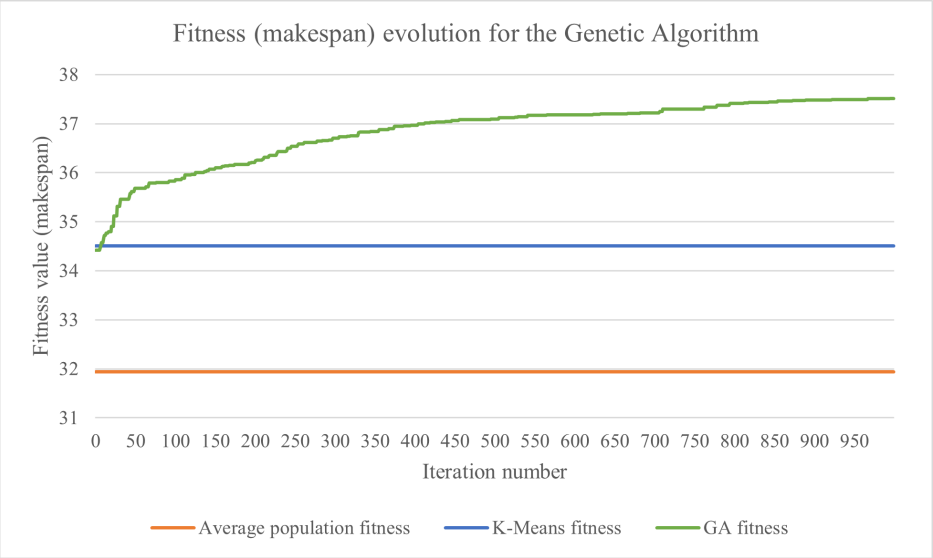


Figure 9. The evolution of the best chromosome's fitness for a genetic algorithm that uses the makespan as the fitness function, with the average population fitness and k-means fitness for reference.

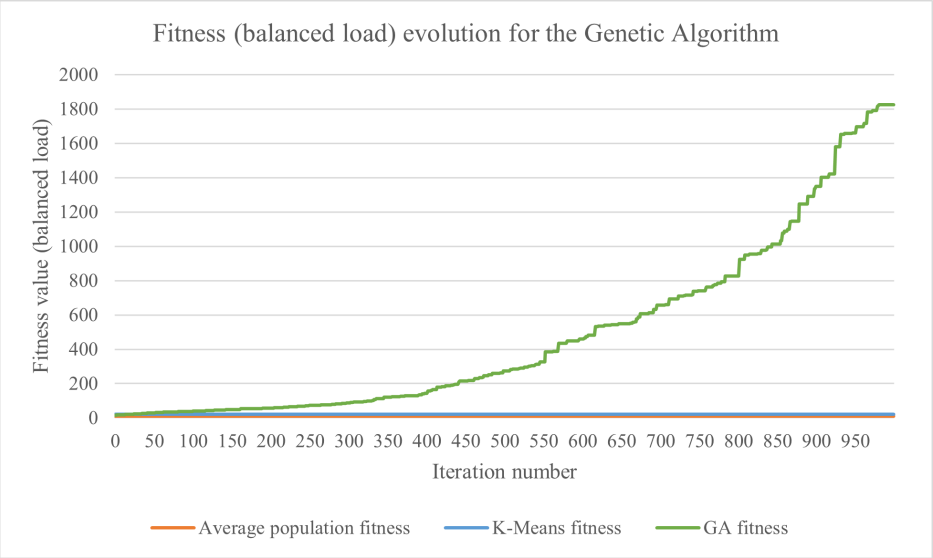


Figure 10. The evolution of the best chromosome's fitness for a genetic algorithm that uses the load balance as the fitness function, with the average population fitness and k-means fitness for reference.

Table 1. Fitness values depending on the aspect that was considered when computing the fitness (i.e., makespan or load balance).

Fitness Method	Avg. Population Fitness	K-Means-Based Fitness	Genetic Algorithm Fitness
makespan	31.94	34.51	37.52
load balance	10.51	20.16	1824.82

How the best fit candidate solution evolved with each iteration of the genetic algorithm, in terms of the data volume that was assigned to each cluster, can be observed in Figures 11 and 12. The largest data volume that needed to be processed by a cluster started from 290,519 d.u. for the GA for the makespan and from 300,391 d.u. for the GA with the

load balance. The final obtained results were 266,556 d.u. and 263,889 d.u., respectively. The lower the data volume was, the better the result. An interesting aspect is the fact that the GA with the load balance obtained a lower value for the largest data volume to be processed by the cluster while also obtaining a load imbalance (in terms of the associated data volumes) of only 547 d.u. compared with 25,832 d.u. for the GA with the makespan.

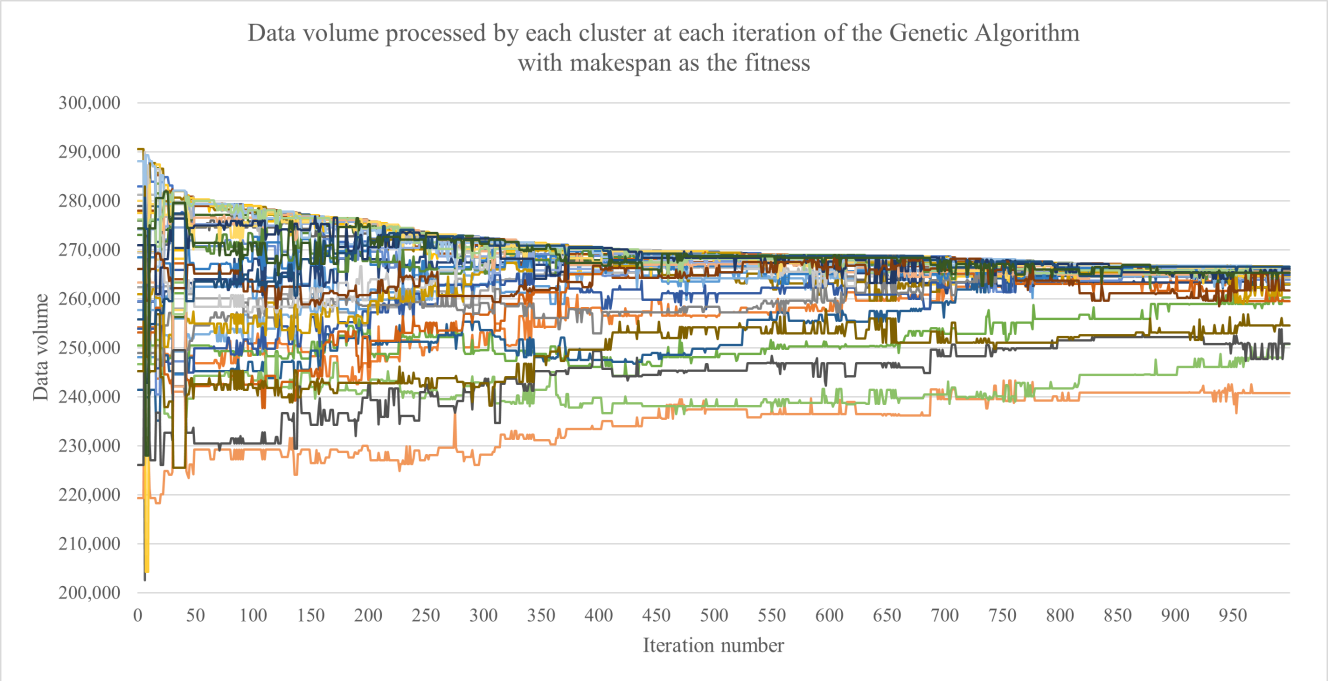


Figure 11. The evolution of the data volume that was assigned to each cluster, represented by the best fit candidate solution at each iteration of the genetic algorithm when using the makespan as the fitness function.

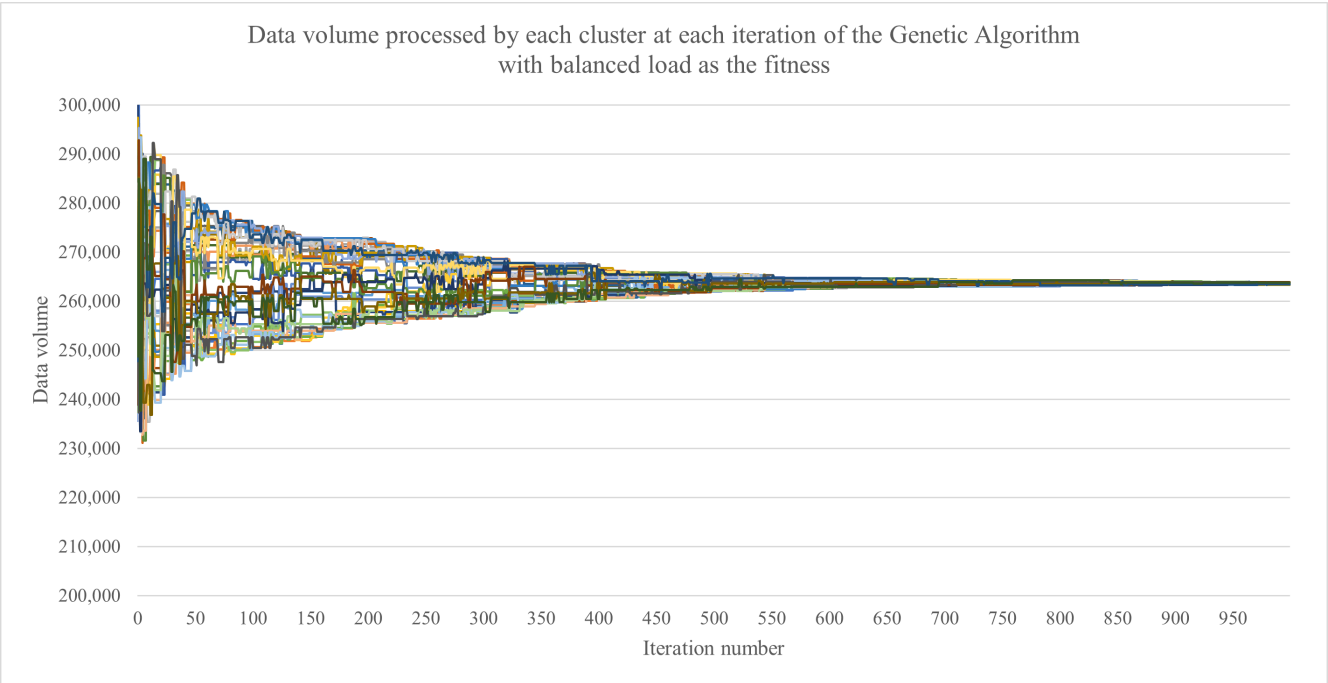


Figure 12. The evolution of the data volume that was assigned to each cluster, represented by the best fit candidate solution at each iteration of the genetic algorithm the load balance as the fitness function.

An overall view of the impact of the proposed methods can be observed in Figure 13, which presents the data volumes that were assigned to be processed by each of the 36 clusters depending of the employed algorithm: a GA with the makespan for fitness (best chromosome from the final iteration), a GA with the load balance for fitness (best chromosome from the final iteration), the k-means adaptation algorithm, a GA with the makespan for fitness (best chromosome from the initial iteration) and a GA with the load balance for fitness (best chromosome from the initial iteration). The final result obtained by the GA with load balance for fitness clearly provided the best results, and it was best in most of the tested scenarios with different numbers of sensors, rules and clusters.

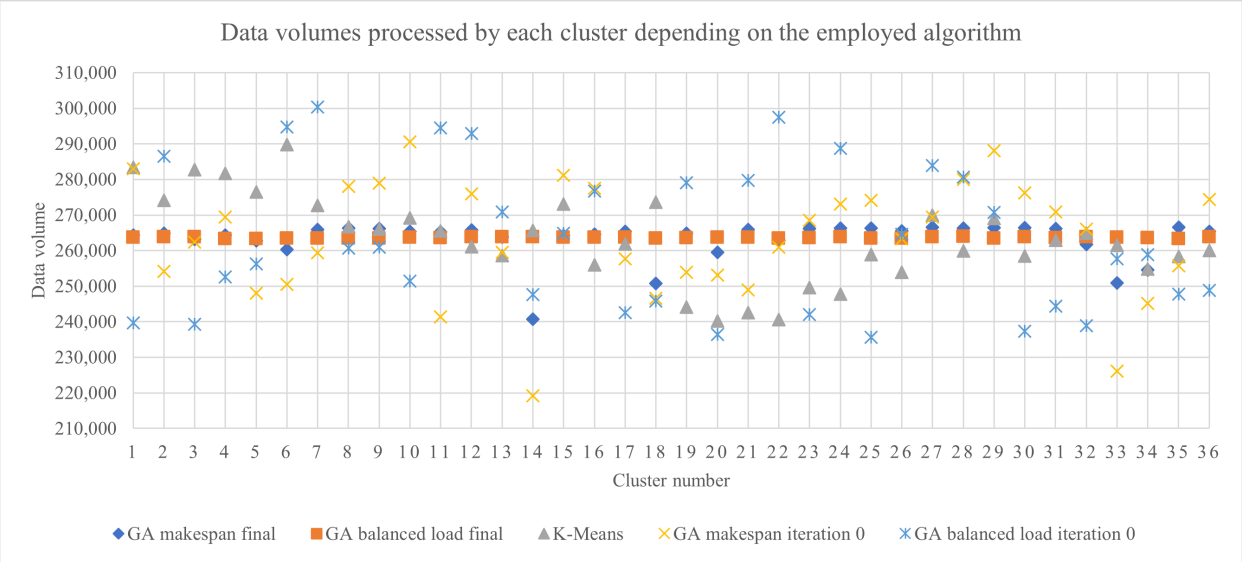


Figure 13. The data volumes that were assigned to be processed by each of the 36 clusters, depending on the employed algorithm.

5. Discussion

The research regarding the two proposed methods presented in this paper expands on the task scheduling problem by considering an extra layer regarding the input data. In the traditional approach, the problem is summarized as having a number of tasks that need to be processed as efficiently as possible in a heterogeneous environment. If only considering the sensor data, a task can be seen as processing that data. The issue is that the sensor data must be evaluated by rule engines, which check each new piece of data against a series of rules. Therefore, traditional methods are not directly applicable. The two new methods proposed in this paper (i.e., the k-means-based clustering and the customization of the genetic algorithm) take into account this extra layer of complexity. They also provide the means to distribute the computation so that the data are transferred for processing only where needed, thus ensuring that less information goes through the network and the rules are evaluated in a shorter time. The two methods can be used in a broader context, such as having task result aggregators, which are equivalent to having rules that aggregate or process information from different specific sensors.

The main scenario in which the proposed methods can be used is if there is a known number of instances available to process sensor data. In this case, the clustering methods use as input the number of instances, and they produce a good rule-to-cluster association for handling the data. This is one of the reasons why the k-means algorithm is appropriate for solving this method. The proposed k-means solution is adapted to first generate a fixed small number of clusters and afterward split the clusters to obtain the desired number. An interesting variation is to consider a variable number of clusters and to evaluate the quality of the solution at each split. This way, using a slightly smaller number of clusters, a solution with a better performance metric might be obtained. This approach is inspired

by hierarchical k-means methods, which allow the determination of the optimal number of clusters.

The proposed architecture stores all the sensor data in a global distributed storage system, which allows the correlation of data from multiple sources. Instead of having multiple local processing instances, cost efficiency is increased by using a global pool of computing instances. On the other hand, using this approach has the disadvantage of ensuring data privacy and security, but this is somewhat similar to the way that cloud providers ensure these aspects. In addition, some entities may not want their sensor data to be shared and used by artificial intelligence methods to find data or even behavioral patterns. If the solution is used for a smart city or region, and it is governed by the same entity, then the data privacy issue becomes mute.

In terms of where the data processing is performed, there are three main directions: edge computing, fog computing and cloud computing. Edge computing means that the processing is performed directly on the devices attached to the sensors. Fog computing is processing in a network close to the edge but somewhere between the edge and the cloud. Finally, cloud computing means handling the data in a cloud solution, which has access to information from a large amount of sensors. Each of them has advantages and disadvantages, and some solutions employ all three [43,44]. The current paper focuses on a cloud solution in order to have a broader view of the underlying sensor networks. Because all the data are considered, the best decision can be made but at the expense of an increased computing time. Considering a scenario with a high number of sensor networks, the costs to perform these computations are similar to the costs in the case of fog and edge computing. Instead of having centralized processing, there is processing on each fog computing node and on each edge computing node. In order to decrease costs and processing times, the rule engine system can be designed to perform processing at all three levels (edge, fog and cloud). The sensor data go to the edge level, where some local rules are triggered. Then, the relevant data go to the fog level, where other rules are used. Finally, the cloud rules perform the processing and trigger events. This method has the advantage of making quick real-time decisions at the edge level while considering the big picture at cloud level. The two proposed methods can be applied best at the fog and especially cloud levels because they thrive when a limited amount computing instances are available to process a large amount of sensor data.

Looking at the proposed cloud solution architecture from Figure 5, a mandatory discussion regards how exactly the data flow is configured and reconfigured depending on changes in the environment (e.g., new sensors are added or removed). The decision to reconfigure the data flow must be made by the cluster manager instance. The proposed algorithms can periodically run and re-evaluate the load of each cluster (i.e., processing instance) and, if needed, trigger a rule migration from one cluster to another or the creation of new clusters or instances or the destruction of existing ones. By using this approach, a high degree of scalability was achieved. The execution of the k-means-based algorithm can be easily parallelized and therefore ensure swift reaction to changes in the system, while the custom genetic algorithm can be used when major changes occur in the environment.

The issues with the proposed system and the two proposed methods were discussed throughout the paper. The efficacy was demonstrated through simulation, which tried to mimic a real multi-sensor network environment. It is highly unlikely that the system will be tested in a real-world environment due to the high amount of sensors needed. However, it can be tested for the sensor data from a real smart home, and those data can be extrapolated to simulate a smart neighborhood. Before this can be tested, the next research step is to deploy the proposed cloud solution in an in-house OpenStack cloud. The system will be generalized for handling any type of task with processing constraints. This generic solution will be offered as a PaaS service for developers, and it will also include an SaaS extension on the Horizon service for monitoring the proposed service.

Funding: This research was supported by the project "Collaborative environment for developing OpenStack-based cloud architectures with applications in RTI" SMIS 124998 from The European Regional Development Fund through the Competitiveness Operational Program 2014–2020, priority axis 1: Research, technological development and innovation (RTI)—the POC/398/1/1 program.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this paper and the code for the implemented solutions are available by requesting them from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

GA	Genetic algorithm
IoT	Internet of Things
WSN	Wireless sensor network
AC	Air conditioning
HVAC	Heating, ventilation and air conditioning
JSON	JavaScript Object Notation
CCTV	Closed-circuit television
IaaS	Infrastructure as a service
PaaS	Platform as a service
SaaS	Software as a service

References

1. Li, S.; Xu, L.D.; Zhao, S. The internet of things: A survey. *Inf. Syst. Front.* **2015**, *17*, 243–259. [CrossRef]
2. Jia, M.; Komeily, A.; Wang, Y.; Srinivasan, R.S. Adopting Internet of Things for the development of smart buildings: A review of enabling technologies and applications. *Autom. Constr.* **2019**, *101*, 111–126. [CrossRef]
3. Dong, B.; Prakash, V.; Feng, F.; O'Neill, Z. A review of smart building sensing system for better indoor environment control. *Energy Build.* **2019**, *199*, 29–46. [CrossRef]
4. Pau, G.; Arena, F. Smart City: The Different Uses of IoT Sensors. *J. Sens. Actuator Netw.* **2022**, *11*, 58. [CrossRef]
5. Silva, B.N.; Khan, M.; Han, K. Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities. *Sustain. Cities Soc.* **2018**, *38*, 697–713. [CrossRef]
6. Zavratnik, V.; Kos, A.; Stojmenova Duh, E. Smart villages: Comprehensive review of initiatives and practices. *Sustainability* **2018**, *10*, 2559. [CrossRef]
7. Salvia, M.; Cornacchia, C.; Di Renzo, G.; Braccio, G.; Annunziato, M.; Colangelo, A.; Orifici, L.; Lapenna, V. Promoting smartness among local areas in a Southern Italian region: The Smart Basilicata Project. *Indoor Built Environ.* **2016**, *25*, 1024–1038. [CrossRef]
8. Sehrawat, D.; Gill, N.S. Smart sensors: Analysis of different types of IoT sensors. In Proceedings of the 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 23–25 April 2019; pp. 523–528.
9. Morais, C.M.d.; Sadok, D.; Kelner, J. An IoT sensor and scenario survey for data researchers. *J. Braz. Comput. Soc.* **2019**, *25*, 1–17. [CrossRef]
10. Wilson, J.S. *Sensor Technology Handbook*; Elsevier: Amsterdam, The Netherlands, 2004.
11. Alías, F.; Alsina-Pagès, R.M. Review of wireless acoustic sensor networks for environmental noise monitoring in smart cities. *J. Sens.* **2019**, *2019*, 7634860. [CrossRef]
12. Sun, Y.; Wu, T.Y.; Zhao, G.; Guizani, M. Efficient rule engine for smart building systems. *IEEE Trans. Comput.* **2014**, *64*, 1658–1669. [CrossRef]
13. El Kaed, C.; Khan, I.; Van Den Berg, A.; Hossayni, H.; Saint-Marcel, C. SRE: Semantic rules engine for the industrial Internet-of-Things gateways. *IEEE Trans. Ind. Inform.* **2017**, *14*, 715–724. [CrossRef]
14. Kargin, A.; Petrenko, T. Internet of Things smart rules engine. In Proceedings of the 2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, Ukraine, 9–12 October 2018; pp. 639–644.
15. Kargin, A.; Petrenko, T. Knowledge Representation in Smart Rules Engine. In Proceedings of the 2019 3rd International Conference on Advanced Information and Communications Technologies (AICT), Lviv, Ukraine, 2–6 July 2019; pp. 231–236.
16. Alexandrescu, A.; Botezatu, N.; Lupu, R. Monitoring and processing of physiological and domotics parameters in an Internet of Things (IoT) assistive living environment. In Proceedings of the 2022 26th International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, Romania, 19–21 October 2022; pp. 362–367.

17. Das, A.K.; Sutrala, A.K.; Kumari, S.; Odelu, V.; Wazid, M.; Li, X. An efficient multi-gateway-based three-factor user authentication and key agreement scheme in hierarchical wireless sensor networks. *Secur. Commun. Netw.* **2016**, *9*, 2070–2092. [CrossRef]
18. Piadyk, Y.; Steers, B.; Mydlarz, C.; Salman, M.; Fuentes, M.; Khan, J.; Jiang, H.; Ozbay, K.; Bello, J.P.; Silva, C. REIP: A Reconfigurable Environmental Intelligence Platform and Software Framework for Fast Sensor Network Prototyping. *Sensors* **2022**, *22*, 3809. [CrossRef] [PubMed]
19. Auroux, S.; Dräxler, M.; Morelli, A.; Mancuso, V. Dynamic network reconfiguration in wireless DenseNets with the CROWD SDN architecture. In Proceedings of the 2015 European Conference on Networks and Communications (EuCNC), Paris, France, 29 June–2 July 2015; pp. 144–148.
20. Helkey, J.; Holder, L.; Shirazi, B. Comparison of simulators for assessing the ability to sustain wireless sensor networks using dynamic network reconfiguration. *Sustain. Comput. Inform. Syst.* **2016**, *9*, 1–7. [CrossRef]
21. Derr, K.; Manic, M. Wireless sensor network configuration—Part II: Adaptive coverage for decentralized algorithms. *IEEE Trans. Ind. Inform.* **2013**, *9*, 1728–1738. [CrossRef]
22. Sundhari, R.M.; Jaikumar, K. IoT assisted Hierarchical Computation Strategic Making (HCSM) and Dynamic Stochastic Optimization Technique (DSOT) for energy optimization in wireless sensor networks for smart city monitoring. *Comput. Commun.* **2020**, *150*, 226–234. [CrossRef]
23. Akash, A.R.; Hossen, M.; Hassan, M.R.; Hossain, M.I. Gateway node-based clustering hierarchy for improving energy efficiency of wireless body area networks. In Proceedings of the 2019 5th International Conference on Advances in Electrical Engineering (ICAEE), Dhaka, Bangladesh, 26–28 September 2019; pp. 668–672.
24. Solaiman, B. Energy optimization in wireless sensor networks using a hybrid k-means pso clustering algorithm. *Turk. J. Electr. Eng. Comput. Sci.* **2016**, *24*, 2679–2695. [CrossRef]
25. Kumar, G.; Mehra, H.; Seth, A.R.; Radhakrishnan, P.; Hemavathi, N.; Sudha, S. An hybrid clustering algorithm for optimal clusters in wireless sensor networks. In Proceedings of the 2014 IEEE Students' Conference on Electrical, Electronics and Computer Science, Bhopal, India, 1–2 March 2014; pp. 1–6.
26. Bharany, S.; Sharma, S.; Frnda, J.; Shuaib, M.; Khalid, M.I.; Hussain, S.; Iqbal, J.; Ullah, S.S. Wildfire Monitoring Based on Energy Efficient Clustering Approach for FANETS. *Drones* **2022**, *6*, 193. [CrossRef]
27. Xiong, J.; Ren, J.; Chen, L.; Yao, Z.; Lin, M.; Wu, D.; Niu, B. Enhancing privacy and availability for data clustering in intelligent electrical service of IoT. *IEEE Internet Things J.* **2018**, *6*, 1530–1540. [CrossRef]
28. Zhang, D.G.; Zhang, T.; Zhang, J.; Dong, Y.; Zhang, X.D. A kind of effective data aggregating method based on compressive sensing for wireless sensor network. *EURASIP J. Wirel. Commun. Netw.* **2018**, *2018*, 1–15. [CrossRef]
29. Wang, J.; Gao, Y.; Wang, K.; Sangaiah, A.K.; Lim, S.J. An affinity propagation-based self-adaptive clustering method for wireless sensor networks. *Sensors* **2019**, *19*, 2579. [CrossRef]
30. Lapegna, M.; Balzano, W.; Meyer, N.; Romano, D. Clustering Algorithms on Low-Power and High-Performance Devices for Edge Computing Environments. *Sensors* **2021**, *21*, 5395. [CrossRef] [PubMed]
31. Barron, A.; Sanchez-Gallegos, D.D.; Carrizales-Espinoza, D.; Gonzalez-Compean, J.L.; Morales-Sandoval, M. On the Efficient Delivery and Storage of IoT Data in Edge-Fog-Cloud Environments. *Sensors* **2022**, *22*, 7016. [CrossRef] [PubMed]
32. Lopez-Arevalo, I.; Gonzalez-Compean, J.L.; Hinojosa-Tijerina, M.; Martinez-Rendon, C.; Montella, R.; Martinez-Rodriguez, J.L. A WoT-Based Method for Creating Digital Sentinel Twins of IoT Devices. *Sensors* **2021**, *21*, 5531. [CrossRef] [PubMed]
33. Achirei, S.D.; Zvoristeanu, O.; Alexandrescu, A.; Botezatu, N.A.; Stan, A.; Rotariu, C.; Lupu, R.G.; Caraiman, S. Smartcare: On the design of an iot based solution for assisted living. In Proceedings of the 2020 International Conference on e-Health and Bioengineering (EHB), Iasi, Romania, 29–30 October 2020; pp. 1–4.
34. Ahmadian, M.M.; Khatami, M.; Salehipour, A.; Cheng, T. Four decades of research on the open-shop scheduling problem to minimize the makespan. *Eur. J. Oper. Res.* **2021**, *295*, 399–426. [CrossRef]
35. Faisal, M.; Zamzami, E. Comparative analysis of inter-centroid K-Means performance using euclidean distance, canberra distance and manhattan distance. *Proc. J. Phys. Conf. Ser.* **2020**, *1566*, 012112. [CrossRef]
36. Zhang, Y.; Xiong, Z.; Mao, J.; Ou, L. The study of parallel k-means algorithm. In Proceedings of the 2006 6th World Congress on Intelligent Control and Automation, Dalian, China, 21–23 June 2006; Volume 2, pp. 5868–5871.
37. Vasconcelos, J.; Ramirez, J.A.; Takahashi, R.; Saldanha, R. Improvements in genetic algorithms. *IEEE Trans. Magn.* **2001**, *37*, 3414–3417. [CrossRef]
38. Kumar, M.; Husain, D.; Upreti, N.; Gupta, D. Genetic algorithm: Review and application. *SSRN* **2010**. ssrn.3529843. [CrossRef]
39. Alexandrescu, A.; Agavriloaei, I.; Craus, M. A task mapping simulation framework for comparing the performance of mapping heuristics in various scenarios. In Proceedings of the 2012 16th International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, Romania, 12–14 October 2012; pp. 1–6.
40. Alexandrescu, A.; Agavriloaei, I.; Craus, M. A genetic algorithm for mapping tasks in heterogeneous computing systems. In Proceedings of the 15th International Conference on System Theory, Control and Computing, Sinaia, Romania, 14–16 October 2011; pp. 1–6.
41. Alexandrescu, A. Mapping interdependent tasks in a computational environment using genetie algorithms. In Proceedings of the 2015 14th RoEduNet International Conference-Networking in Education and Research (RoEduNet NER), Craiova, Romania, 24–26 September 2015; pp. 173–177.

42. Longo, F.; Bruneo, D.; Distefano, S.; Merlino, G.; Puliafito, A. Stack4things: An openstack-based framework for iot. In Proceedings of the 2015 3rd International Conference on Future Internet of Things and Cloud, Rome, Italy, 24–26 August 2015; pp. 204–211.
43. Sánchez-Gallegos, D.D.; Galaviz-Mosqueda, A.; Gonzalez-Compean, J.; Villarreal-Reyes, S.; Perez-Ramos, A.E.; Carrizales-Espinoza, D.; Carretero, J. On the continuous processing of health data in edge-fog-cloud computing by using micro/nanoservice composition. *IEEE Access* **2020**, *8*, 120255–120281. [CrossRef]
44. Cao, H.; Wachowicz, M. An Edge-Fog-Cloud Architecture of Streaming Analytics for Internet of Things Applications. *Sensors* **2019**, *19*, 3594. [CrossRef]

applied sciences

MDPI

Article

Blockchain-Based Platform to Fight Disinformation Using Crowd Wisdom and Artificial Intelligence

Cristian Nicolae Buțincu * and Adrian Alexandrescu

Department of Computer Science and Engineering, Faculty of Automatic Control and Computer Engineering, Gheorghe Asachi Technical University of Iasi, 700050 Iasi, Romania; adrian.alexandrescu@academic.tuiasi.ro
* Correspondence: cristian-nicolae.butincu@academic.tuiasi.ro

Abstract: Disinformation and fake news are used by multiple actors to manipulate and influence the public with the purpose of gaining a series of advantages. This paper describes a promising solution to the increased spread of disinformation on the Internet. Our approach leverages blockchain technology combined with both crowd intelligence and federated artificial intelligence to develop efficient capabilities that address the disinformation phenomenon. The blockchain-based architecture of the platform creates a decentralized ecosystem that ensures transparency and trust, enabling the users to make correctly informed decisions in the face of disinformation. The key differentiating factor of the platform is the incorporation of both crowd and artificial intelligence in a system that can identify and respond to disinformation quickly and efficiently. The presented architecture can be used to build reactive and proactive platforms to effectively challenge disinformation.

Keywords: blockchain; fighting disinformation; crowd wisdom; artificial intelligence; governance protocol; cryptocurrency tokens

Citation: Buțincu, C.N.; Alexandrescu, A. Blockchain-Based Platform to Fight Disinformation Using Crowd Wisdom and Artificial Intelligence. *Appl. Sci.* **2023**, *13*, 6088. https://doi.org/10.3390/app13106088

Academic Editor: Gianluca Lax

Received: 14 April 2023
Revised: 2 May 2023
Accepted: 11 May 2023
Published: 16 May 2023

1. Introduction

Producing and publishing digital content on the Internet has become increasingly convenient and easy. Digital content (e.g., news, articles, images, videos) is being created and published at large scales today. Not only do humans are part of the information revolution, but also machines, AI (Artificial Intelligence) playing an important role in the generation of digital content. In this context, current and future generations will be overwhelmed with large amounts of digital content, impossible for the general user to sift through, analyze and consume. Therefore, information and information control are of paramount importance for a society and, as the future of almost everything is becoming more digital, information manipulation is becoming one of the most challenging aspects of the future.

Disinformation is becoming day by day more present at all levels of society, from the individual to the state. Now, the offensive in the information space is highly effective, cheap and difficult to counter. At the same time, defending against it is almost impossible, due to the huge flow of information that is generated on the Internet every day and the fact that social media algorithms are owned and controlled by companies rather than states. The reasoning for and the effects of disinformation vary, and mainstream media plays an important role [1]. Detecting disinformation is not trivial, but certain steps have been taken in order to fight this phenomenon [2].

Most of the population has no time nor the skills to check the authenticity of content published on news sites or social media platforms. Therefore, it has become essential to reach out the authenticity and truth behind the published information, i.e., where it has come from, who created it and its factuality.

Current fact-checking initiatives are small-scaled and human-led, which are prone to errors and interpretation. Having so many different fact-checking initiatives creates

confusion and huge duplication of work, which is not sustainable in the long run. Thus, even though the intention of fact-checkers is admirable it does not make them a viable solution to solve the issue of disinformation.

No technology can fully solve the challenges of establishing trust between people, nor eliminate the underlying motivations for profit and/or political gain that drive disinformation in the first place. This is why, to solve the problem of disinformation and fake news, society must fight back using various tools, ranging from technological tools (like blockchain [3] and AI [4]) to educational tools.

This paper describes a decentralized anti-disinformation platform for fact checking and trust assessment based on Blockchain, Crowd Wisdom and AI technologies.

At present, trust in mainstream media is lower than ever due to ever increasing disinformation levels and the amount of information that cannot be reliably checked. In the current media landscape, publications are driven by click-based ad revenue. To increase the number of views (and therefore to increase their income), publications tend to take journalistic shortcuts (like not thoroughly verifying the facts or presenting incomplete information just to be the first to announce some news). In this landscape, even reputable publications tend to favor engagement over clarity. This in turn impacts the readers' ability to distinguish the truth from disinformation and fake news.

A survey [5] regarding trust in the media (television, radio, written press, internet and social networks) was conducted in 2019 in all 28 EU countries. It was found that 41% of Europeans had medium trust in the media, 40% of the EU citizens had low or no trust in the media at all and only 19% of adults had high trust in the news media. So, the population in general is aware about the media quality and the spread of disinformation and fake news. That was in 2019 and since then, the sensitive geopolitical context further increased the spread of disinformation and fake news.

Because, in recent years, disinformation and misinformation spread increased to levels never seen before, the lack of trust in the media also increased in the general population. This affects all publishers, not only the ones responsible for spreading fake news. The lack of trust that consumes our society can be fixed by revealing, in a trusted way, what is true and what is not true. The platform described in this paper comes to address these issues, to expose disinformation and misinformation spread online by dishonest publishers and to finally restore trust to reputable publishers.

Currently, there are not enough platforms that can provide valuable and scalable fact-checking. This is in part because their operation is not sustainable for the long term, due to the dependency on paid human interaction.

Given the context, there is a real need for scalable and reliable tools that can be used at the level of society to fight against disinformation and fake news. In addition, when developing such tools, there are important questions to consider about who sets the standards, who is responsible for validation, and who manages the entire system. This is where blockchain technology comes into play since its decentralized nature can help address many of these concerns. Most importantly, it provides transparency since it eliminates the need for a single, trusted institution to make these decisions.

This paper describes the main drivers behind building such a platform, explores different solutions that try to solve the same problem and their drawbacks, and presents the overall architecture of the system.

The next section of the paper presents the current approaches to fighting disinformation and focuses on blockchain and smart contracts. The Related Work section of the paper describes the main approaches and the state of the art for blockchain-based solutions to fight disinformation, as well as their drawbacks and trade-offs. A detailed description of the proposed platform architecture is given in Section 4, which is divided into several subsections that discuss the general overview, blockchain and smart contracts, AI and human validators, system components, article extraction architecture, the platform protocol, the trust and security framework, and the provided APIs. In Section 5 we present some of the results obtained running the platform on a subset of publisher websites along with configu-

ration and runtime details. The paper ends with the conclusions section, where the main features of the platform are highlighted along with its current and future developments.

2. Background

2.1. Fighting Disinformation

In terms of fact-checking [6,7], there are a couple of current approaches to fighting disinformation, each with its own drawbacks:

1. Verification is performed by ***Professional Fact Checkers*** with accurate results following a standardized methodology for fact checking. This approach is time consuming, has little impact (since it is conducted most of the time in the "post-mortem" phase of the misinformation spreading) and it is very difficult to scale.
2. Verification is performed by non-professional communities—***Crowdsourced Fact Checking***, leveraging "cross wisdom" with no supporting tools. This approach has good potential to scale, but it is prone to becoming partisan or to be detoured. The level of impact is still low due to the "post-mortem" nature of the analysis.
3. Verification is conducted by software tools that utilize statistical algorithms mimicking the critical thinking process—***Non-assisted Automated Verification***. This approach has huge scaling and impact potential, but it cannot achieve the same level of accuracy as 1 and 2.

In the approaches mentioned above, the system must choose between scalability and accuracy. In any completely distributed system, consensus is considered one of the key properties when talking about a system that is considered reliable. The first blockchain has run on a public network since 2009 and opened the door to a new type of architecture, but at the same time, the technology offers just an infrastructure that must be combined with other techniques to offer solutions for today's problems.

The approaches for blockchain-based solutions to address the challenges of online disinformation are:

- ***Verifying provenance.*** This approach tracks and verifies the sources of online digital content. Examples of projects employing this approach are:
  - New York Times' News Provenance Project (https://newsprovenanceproject.com, accessed on 8 May 2023; https://www.nytimes.com/2020/07/06/insider/could-we-fight-misinformation-with-blockchain-technology.html, accessed on 8 May 2023)
  - Truepic (https://truepic.com/, accessed on 8 May 2023) notarizes content on the Bitcoin and Ethereum blockchains to establish a chain of custody from capture to storage
- ***Assessing identity and reputation.*** Blockchain-based solutions can track and verify the reputation of content creators in a transparent and decentralized way, thus eliminating the need for a trusted and centralized institution. Smart contracts built on blockchain offer the necessary mechanisms to achieve these goals.
- ***Incentivizing high quality content and content assessment.*** The blockchain generated cryptocurrency can be used to reward/penalize the actors that are responsible with trust assessment of both content and content creators. The trust score will be reflected upon content creators which will be incentivized to create high quality content. Pressland (https://pressland.com/, accessed on 5 August 2022) is another example of a system that stores online data on blockchain after it was analyzed using machine learning algorithms.

2.2. Blockchain and Smart Contracts

The blockchain technology was made popular by Bitcoin [8] and since then it was extensively adopted by the developer community into a broad range of different implementations such as Ethereum, Litecoin, Cardano, Tezos, and Solana, to name only a few. Instead of executing transactions through a centralized entity, transactions are executed by a net-

work of participants, mediated by smart contract programs. These smart contracts usually run using open-source software built and maintained by a community of developers.

The blockchain and smart contracts [9] technology enabled a wide range of applications in areas such as money transfer, cryptocurrencies [10], decentralized finance (DeFi) [11,12], Internet of Things (IoT) [13], Self-Sovereign Identity (SSI) [14], healthcare [15], logistics [16], Non-Fungible Tokens (NFTs) [17], government, media and so on.

The Blockchain and smart contracts technology enables the platform to inherently expose the following features:

- ***Zero downtime***—Since the smart contracts are deployed on the blockchain, the network will serve the clients. This also protects the platform from DDoS attacks.
- ***Privacy***—The access to the platform is guaranteed to be anonymous with no links to real world profiles.
- ***Resistance to censorship***—No single entity on the network can block users from interacting with the platform.
- ***Complete data integrity***—Data stored on the blockchain is immutable and indisputable; this is guaranteed by cryptographic primitives. Malicious actors cannot forge transactions or other data that has already been made public.
- ***Trustless computation and verifiable behavior***—Smart contracts can be analyzed and are guaranteed to be executed in predictable ways, without the need to trust a central authority.

Smart contracts are designed to be ***trustless***; this implies that the users don't have to trust third parties (developers, companies, or any other entities) to interact with a contract. By design, smart contracts are also ***immutable*** and cannot be altered; a contract will only execute the business logic defined in the code at the time of deployment.

Once the core of the platform is set on the blockchain, additional building blocks can come in place to provide a reliable and trusted anti-disinformation platform. These building blocks take the form specialized web crawlers and scrapers (distributed pieces of software fetching data from the Internet), Crowd Wisdom (human actors performing data analysis), and Artificial Intelligence Modules assembled into a federation.

3. Related Work

There is an increased scientific effort to develop new solutions based on blockchain that try to detect and fight against fake news. Many scientific papers that deal with different aspects of the fight against misinformation and disinformation were written.

In [18], the authors propose a blockchain platform focused on the "fact", where the news posted on the platform by news publishers will be analyzed by an AI component. This platform is similar to a social media platform where the content is generated and consumed within it. Having the content posted and reposted inside the platform, allows to easily track the source and the propagation path of the news.

Another solution is presented in [19], where the authors present an incentive-aware blockchain-based solution for internet of fake media things. The system architecture runs on a customized PoA (Proof-of-Authority) protocol. Here, news organizations must register to be able to publish news into the blockchain network, and the nodes, represented by news publishers, are also in charge of news validation.

The authors of [20] introduce a blockchain solution that uses a deep learning hybrid model to detect fake news. Their approach uses a pretrained GloVe (Global Vectors for Word Representation) embedding matrix for word embeddings. These embeddings are used as input for deep learning models to classify the news. The news is directly published into their system by reporters, analyzed by analyzers and validated by validators. There are three types of analyzers: deep learning, verified journalists and individuals.

In [21], the authors propose a combination of blockchain and machine learning techniques to detect fake news. They use NLP (Natural Language Processing) and reinforcement learning techniques and a customized blockchain consensus algorithm based on PoA protocol.

Another blockchain-based solution for fake news detection in social media is given in [22]. In this approach, the news is published into their system and random users are selected to act as validators. To select the users that are closer to the source of the news event, the system uses the BFS (Breadth First Search) algorithm to calculate the proximity of a user.

The authors of [23] propose a system where multiple news sources and social media platforms use the same blockchain to track how users share the news on their timeline. A modified version of a posted news can be easily tracked, flagged and removed from the system. However, this system assumes that the original published news is true and only detects if a user modifies the content on his timeline.

In another paper [24], the authors present a blockchain system featuring machine learning for mitigating the effects of fake news called Reliable News Sharing Platform. The news is published by publishers into the system. There are four entities involved in the system: reporters, machine learning analyzer, miners and readers. The system uses NLP and CNN (Convolutional Neural Networks) to analyze the news. The news articles are stored in the InterPlanetary File System (IPFS) and only a reference to IPFS id is stored on the blockchain.

A news verification blockchain system based on Ethereum and IPFS is presented in [25]. In this system, the journalists use a smart contract to publish news on the blockchain, after which they are validated by validators. The news content is stored on IPFS and the IPFS id is stored on blockchain, similar to the approach presented in [24].

In [26], a private blockchain and watermarking based social media framework is proposed to control the fake news propagation. In this system, users must register and provide their identity. The model focuses on identifying the source of news and verifies the fake news based on user reports.

Most solutions, which are trying to solve the problem of fake news, try to establish a news platform where publishers publish their news articles. The main problem with this approach is that it is extremely hard if not impossible to get all news agencies and publishers to use a particular platform. Another problem is that using only one approach for news analysis, AI or humans, is close to impossible to get things right in all cases.

4. Solution Description

This paper presents a way to fight disinformation using a coordinated approach between technologies and human intelligence in a unique way that will strengthen the defensive mechanisms against manipulated information and offers the reader an easy way to identify and verify the authenticity and factuality of the information. This solution enables building a system that counters disinformation using decentralized actors featuring AI, crowd wisdom, smart contracts and blockchain technologies.

The blockchain technology creates accountability and traceability. Artificial intelligence and human crowd wisdom help in detecting fake news. With the traceable, transparent and decentralized nature of the blockchain, it is possible to verify the authenticity of the information or its sources and build trust on the news available on the Internet.

The solution we are currently proposing consists of designing and deploying a scalable, decentralized anti-disinformation platform with the core of the platform built upon Blockchain technology and where verification is performed by Crowd Wisdom with support from Non-assisted Automated Verification through federated Artificial Intelligence modules.

Our proposal follows a similar approach to search engines and does not require news agencies and publishers to publish their news on a particular platform. Instead, a series of specialized web crawlers crawl their sites and index their content on blockchain. It is this blockchain stored content that is further verified for authenticity and trust using AI based algorithms and human crowd wisdom. This ensures that a large amount of content is verified and authenticated in a decentralized manner by the AI and human curators.

The platform will incentivize publishers to promote trustworthy content to reach a higher reputation score and increase the trust of their readers. The power of blockchain combined with crowd wisdom and AI will apply constant pressure on media outlets to scrutinize their content and promote trusted, high-quality content.

The platform architecture described in this paper stands at the core of the FiDisD project (https://www.trublo.eu/fidisd/, accessed on 8 May 2023). The FiDisD project provides an efficient and trusted solution to the problem of misinformation and disinformation by taking both a reactive and a proactive approach to effectively challenge the disinformation phenomenon.

4.1. General Overview

All the solutions proposed by various authors, which have been previously discussed, have some limitations. A major drawback is the fact that news publishers must use a particular platform to publish their news. This is the main reason why these solutions were not adopted by the mainstream. To address this problem, our solution does not impose a unique news platform. We let news agencies and publishers complete freedom as to where to publish their news and crawl their sites to fetch the news content into our blockchain-based system. Here, federated AI modules and crowd wisdom take over and analyze this content in a transparent and decentralized manner.

The system architecture described in this paper integrates the blockchain technology, artificial intelligence, human actors, off-chain distributed data storage and web crawlers. Online news from different agencies and news publisher sites are automatically fetched, aggregated and analyzed in parallel by federated AI modules and human validators. The end results are made available through a web portal where everyone can come and check the truth behind online news articles together with detailed analysis proofs.

Figure 1 presents the main features of overall architecture of the system.

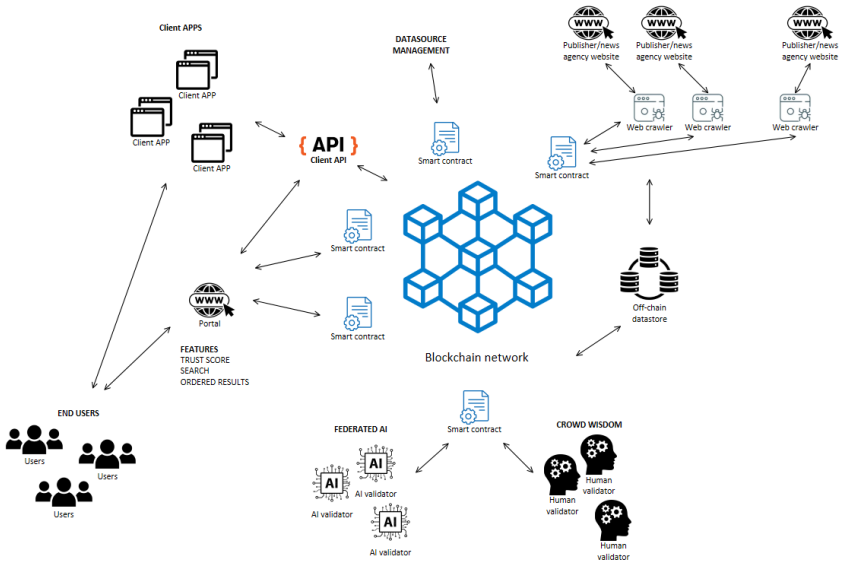


Figure 1. System Architecture Overview.

The main components are:

- the FiDisD blockchain network, which stores information regarding the news items from the off-chain datastore,
- the web crawlers, which include the web scrapers, which are used to extract the news articles from the publisher agencies,
- the datasource management, which is used to determine what article URLs are used by which web crawlers and what web scrapers extract the actual article information,
- the off-chain datastore, which stored the extracted article information,

- the federated AI and the crowd wisdom, which are used to validate the articles,
- the FiDisD portal application, which allows the users to view the extracted and validated articles,
- the client APIs, which can be used by third-party application to interrogate and obtain the information provided by the FiDisD solution.

Regarding the implementation of the conceptual architecture from Figure 1, for simplicity's sake, the datasource management, the communication with the off-chain datasource, and other business logic required for exposing the required APIs are all encapsulated in the Offchain Core component.

4.2. AI and Human Validators

The platform's architecture integrates a federated AI component that is formed by separated autonomous AI modules. These AI modules perform, in parallel, along with human actors, the analysis of the news and assign a trust score to each of them. Moreover, each of these modules can provide additional information as to why a certain trust score was given.

The platform is designed to support a plugin-like architecture while also exposing well defined interfaces, so that anyone can develop an AI module and register it in the platform. The AI modules are subject to reputation and incentivization/penalization schemes, such as human validator components.

An AI module usually employs a mix of NLP, Deep Learning and Neural Networks to extract the relevant entities from the news, to perform sentiment and bias analysis, to detect clickbait, to check for semantic similarities and to automatically classify the news items. A Cross-Lingual Semantic Textual Similarity can also be considered, which will allow the use pretrained models on multiple languages.

The platform combines the AI modules and the human validators effort through a recommender system that selects which news will be validated by humans. This system must consider at least the following three factors:

- Validator affinity: a human validator receives news on topics and areas that he is familiar with, based on previously validated news. This approach is similar to the behavior of classic recommender systems.
- Confidence scores of the AI validators: the fake news detection can be seen as a semi-supervised learning problem. Most news articles are unlabeled data, with few data being labeled by human validators. Uncertainty sampling will allow the recommender system to select the news for which the AI validators are least certain to be validated by humans.
- A random factor to reduce the assignments predictability and to prevent malicious actors from exploiting the system and to modify the verdict for certain news. An entity that spreads fake news might try to add bot-like human validators to our system to manipulate the news assessment.

4.3. System Components

4.3.1. Blockchain Network

This component provides a decentralized ledger used for the cryptographic proofs of system data: Crowd Wisdom and Federated AI modules score assessments, news timestamps and content proof hashes, user activity tracking (used to distribute token awards). The blockchain technology through its inherent decentralized nature provides transparency and eliminates the need for a single, trusted institution to make these decisions.

The blockchain network component stores transactions that indicate what news items are stored in the off-chain datastore and the identity of the entities that fetched the data. The blockchain stores cryptographic proofs of off-chain data to guarantee its validity and the fact that it was not tampered with.

4.3.2. Smart Contracts/Platform Protocol

Smart contracts record the validators activity (AI modules and crowd wisdom). These smart contracts encapsulate mathematical models for calculating trust scores for news articles and reputation scores for validators and publishers. The penalization and incentivization schemes are also implemented at this level.

This component comprises all smart contracts developed to provide a decentralized and verifiable business logic that resides at the core of the platform. End users through Portal, Crowd Wisdom, AI Modules, private and public client APIs, all interact with these smart contracts to create, read and update data into blockchain. Therefore, through blockchain cryptographic proofs, anyone can check the data validity and be assured that no one can manipulate this data except through clear and transparent access protocols.

These contracts form a decentralized protocol that gives its users access to a DAO-like (Decentralized Autonomous Organization) structure. A governance module together with governance and activity tokens, timelocks and access control primitives are used at the core of the protocol.

4.3.3. Web Crawler

The Web Crawler component has two subcomponents:

- Web Crawler;
- Web Scraper.

Both these components are completely distributed. The Web Crawler extracts the URLs from news publishers' web pages and compiles a list of URLs that identify news articles. This list is fed into the Web Scraper component.

Each news page is processed by multiple scrapers in such a way that they will not become blacklisted or banned from crawling on a particular website. This also guarantees that a particular news site serves the same content to its readers.

The Web Scraper component performs a targeted extraction from each news page depending on the extraction template for each news website. The extraction template contains the information of interest from a news webpage: article name, author, published date, referenced articles, and the news content in rich-text format (i.e., html with the associated multimedia content). For each considered website there is a predetermined extraction template with update triggers in place for the case in which the website's structure changes and the template is no longer applicable.

To prevent malicious actors from inserting invalid or altered data into our system, each site is crawled and scraped in parallel by multiple entities, and the majority gives the final data that will be recorded into our systems. All data transfers are cryptographically signed using public-key cryptography. Moreover, the platform incorporates an algorithm that deterministically allocates the resources to be fetched based on entities public keys and resource hashes.

4.3.4. Off-Chain Core

This component contains all business logic executed off-chain. It is responsible for providing public and private (internal) APIs to the other components. The storage off-chain core is datastore agnostic, meaning that any storage technology can be used without triggering modifications into the upper layers. The off-chain storage access is performed through the Data Persistence component (see Figure 2), that provides the data storage abstraction.

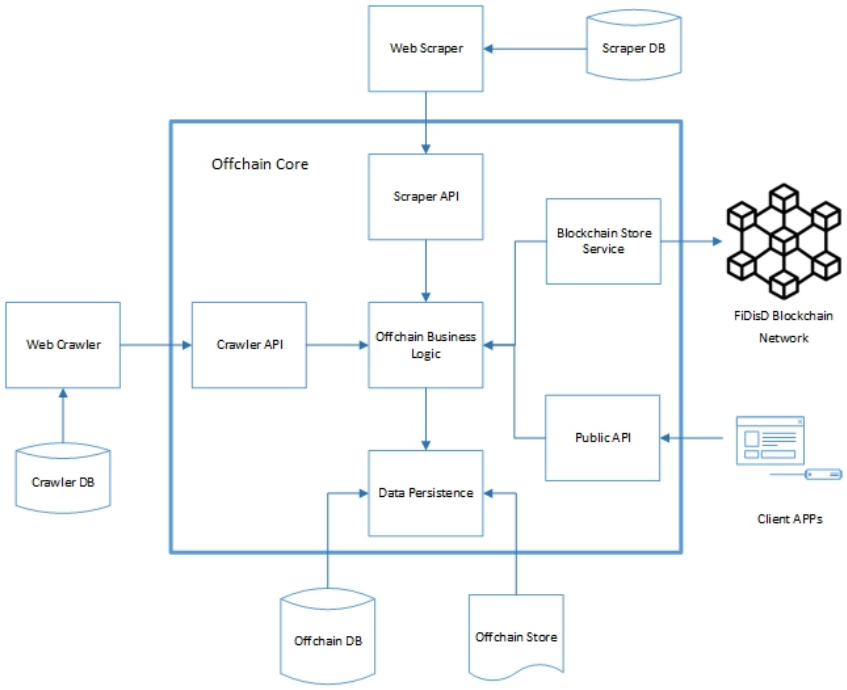


Figure 2. Article Extraction Architecture.

4.3.5. Off-Chain Datastore

This component provides off-chain storage. It is accessed by Off-chain Core component and, together with blockchain storage, stores all the data in the system. Curated data extracted from the distributed crawlers/scrapers is stored off-chain, in a separate distributed storage system; hashes of the data will be stored on-chain.

The off-chain datastore component persists a bundle (text and media resources) of the processed piece of news for each news URL; it also includes the list of crawler ids that processed that piece of news along with the corresponding hash. This component is distributed to ensure redundancy, scalability and a high degree of fault tolerance.

The component acts as a database for keeping track of stored and processed news items and integrates a distributed file system for storing the actual information and the multimedia content.

The interaction with the off-chain datastore is agnostic of any underlying storage implementation and it is performed via a facade API, which ensures the abstraction of the communication from the Web crawler component, the smart contracts from the Blockchain network, and the Client API. The implementation provides an interface to easily integrate different storage solutions in the off-chain datastore.

4.3.6. Federated AI

The Federated AI component employs validator modules based on artificial intelligence and machine learning techniques for determining the trust scores of news articles. The platform also provides an AI implementation to be used as a proof-of-concept/reference implementation.

The main job of an AI module is to analyze the news provided by the system. Similar to the Crowd Wisdom component, each AI module will assign a trust score to news articles and will participate in activity tokens rewards. This way, AI module development entities will be encouraged to join our ecosystem and will further expand the platform AI capabilities.

Any AI module that performs information analysis can be integrated with the platform, using the provided API. We expect that a 3rd party AI module to contain some form of Natural Language Processing based on Deep Learning and Neural Networks to extract

the relevant entities from the news, check pieces of text for semantic similarities, and automatically classify the news items.

The service provided by this component may also be used by the Crowd Wisdom (i.e., human validators) to help them assess the trust scores.

4.3.7. Crowd Wisdom

The Crowd Wisdom component is the other validator component on the blockchain system. Together with the Federated AI component, it forms the overall validator logic. This component is composed of human entities and any person is free to join this system to provide insights regarding the truthfulness behind news articles. For analysis, it may leverage the insights from AI components.

Like AI modules, each human actor analyzes the news and provides a trust score, participating in activity token rewards. This way, more human participants are incentivized to join our system and will further expand the platform analysis capabilities.

To prevent malicious human validators to defraud voting on different news articles, the system uses an algorithm that deterministically allocates human validators to news articles based on human validator public keys and news articles hashes.

The system also tracks the reputation scores of Crowd Wisdom entities and AI modules. This reputation impacts the voting on the news. In addition, with the integrated reward and penalization schemes, the validator components are incentivized to provide objective analysis on news articles.

4.3.8. Client API

The Client API offers integrated access to the system based on access permissions. Software developers can develop 3rd party applications that integrate with the system using this API.

Parts of this component are also used by the Web Crawler component to trigger storing data in the off-chain datastore and storing the news URLs, crawler ids and the hashes in the blockchain network.

The Portal uses the exposed services from the Client API to show end-users the news with their trust scores.

4.3.9. Portal

This component represents the front-end of the platform in the form of a publicly accessible web portal. Here, any user can access the news and the original source, check the trust scores of news articles from both Crowd Wisdom and Federated AI modules and inspect the distribution of scores.

Searches can be performed based on the identified news categories and keywords, and the search result order is tailored to favor high trust articles. The users are also presented with the reasoning of why a particular trust score was given.

4.4. Article Extraction Architecture

One of the main elements of the FiDisD solution is the article extraction architecture, which includes three main components: Web Crawler, Web Scraper and Offchain Core (Figure 2). Multiple web crawlers and web scrapers are employed to identify and extract the article information from various news websites, while the Offchain Core gathers and manages that data.

The role of the Web Crawler is to go through each page of specific websites, identify if the page represents an article and extract all the URLs. The website seed list is obtained from the Offchain Core component. Each Web Crawler has an associated account (i.e., username and password), which is used to authenticate to the Offchain Core. As the crawling is performed, the URLs that are identified as containing article information are sent to the Offchain Core.

Actual article extraction is performed by the Web Scrapers, which extract article information from the pages that were identified by the Web Crawlers. That data is then sent to the Offchain Core together with a hash on the extracted contents; the latter, gets put on the blockchain to ensure trust.

The considered news websites from which to extract information are managed by the Offchain Core. The bootstrap process of the components involves reading the website seed list and the corresponding extraction templates. These are stored on the Offchain Database by the Data Persistence module. Another aspect of bootstrapping is populating the database with the users that are allowed to crawl and scrap the websites.

In order to ensure trust in the system, multiple crawlers and scrapers process the same information. That data together with hash used to check the data integrity are sent to the Offchain Core, which, in turn, stores the information in the database. As soon as a large enough number of crawlers/scrapers obtain the same information then consensus is achieved. At this point, the hash on the article content is stored in the blockchain (by means of the Blockchain Store Service) and a status field is marked accordingly in the database to signify that the article data is available to be retrieved by external Public API clients.

4.5. The Platform Decentralized Protocol

The platform protocol is a decentralized news trust assessment protocol running on blockchain. At its core, it is composed of a series of smart contracts for decentralized governance (voting, proposals, executions), access control, time locks, token contracts for governance and activity tokens, news article score assessment.

4.5.1. Tokens

The protocol uses two types of tokens:

- a governance token, used by community to control the future direction of the project;
- an anti-disinformation activity token, minted and rewarded to users for their work in news analysis.

In general, governance tokens are minted and distributed to the community either via an airdrop to all users that interacted with the platform (according to a predefined release schedule) or via token exchanges (a user might trade his accumulated activity tokens to governance tokens), or a combination of both. The governance token serves the purpose of enabling shared community ownership in the growth and future development of the protocol. This will allow governance token holders to participate in the governance of the protocol in a neutral and trustless manner. As the platform adoption increases, it will positively impact the governance token value and will further incentivize token holders to contribute to the self-sustaining development of the platform.

4.5.2. On-Chain Governance

The protocol is governed and upgraded by governance token holders, using three distinct components implemented as smart contracts: the governance token, a governance module, and a timelock. Taken together, these contracts will allow the community to propose, vote, and implement changes to the protocol.

In general, a good governance design enables the core development team to eventually step out of the decision-making process entirely, achieving a truly self-sustaining and completely decentralized protocol.

The governance token acts to secure the future development of the platform, creating a decentralized voting system that ensures bad actors cannot propose and force development upgrades that may damage the reputation and security of the platform.

In addition, the governance protocol allows governance token holders to delegate their voting power. Users can also choose to delegate themselves.

4.5.3. Access Control

Access control is extremely important in the world of smart contracts. The access control of a contract may govern who can mint or burn tokens, who can vote, cancel or execute proposals or freeze transfers, and many other things.

The most common and basic form of access control is the concept of ownership: an account is the owner of a contract and can perform administrative tasks on it. This approach is best suited for contracts that have a single administrative user.

4.5.4. Role-Based Access Control

While the simplicity of ownership can be useful for simple systems or quick prototyping, different levels of authorization are often needed. Role-Based Access Control offers this kind of flexibility. Under this paradigm, one defines multiple roles, each allowed to perform a different set of actions.

4.5.5. Delayed Operation

Access control is essential to prevent unauthorized access. However, it does not address the issue of a malicious user with administrator rights to attack the system to the prejudice of the other users. This issue is addressed by timelocks.

A timelock acts as a proxy that is governed by proposers and executors. When set as the owner of a smart contract, it ensures that any operation ordered by the proposers will be delayed a certain amount of time. Thus, the users of the smart contract are protected by giving them enough time to review the proposed changes and exit the system before these changes take effect if they consider so.

4.6. Trust and Security Framework

In the process of designing a solid Trust and Security Framework we have integrated into the system the following aspects.

4.6.1. Decentralized Autonomous Organization (DAO)

The underlying principle of the platform is decentralization and transparency thus the implementation of DAO is at the core of our architectural design. DAO is implemented through a series of smart contracts that form the backbone of the whole system. To participate in the platform protocol, members need to hold governance tokens that will enable them to be part of the decision-making process and the future evolution of the project.

On-chain governance refers to a kind of governance where the rules for making changes are encoded into the blockchain protocol. In this system, smart contracts play a fundamental role in executing collective decisions taken through a voting mechanism.

On-chain governance operates in an autonomous and transparent way, and all changes are recorded on the blockchain and are accessible to anyone.

Community members of a DAO can collectively make decisions about the future directions of the project, such as technical updates and token allocation directives, to name only a few. The governance allows members to make proposals and to vote on them. If a proposal passes the voting phase, then it will be executed. Therefore, each member of the DAO can influence the future of the project regardless of his identity.

Power in such systems is fluid as actors can form alliances or delegate their votes, depending on the issues that are at stake. Participants in on-chain governance are free to make decisions according to their best interests that also coincide with the best interest of the protocol itself, since they need to be token holders to be able to participate.

In a DAO, members are also incentivized to be active participants, because the final decisions will affect them and their resources directly. Moreover, on-chain governance is performed by voting through governance tokens, and incentives for engaging in the voting process are usually offered to encourage user participation. As opposed to traditional voting, on-chain voting successfully addresses some of its challenges such as lack of

accountability, low transparency and external influence. Since everything is recorded on a blockchain and is openly available, external influence is very difficult.

The key advantages of on-chain governance:

- security provided by blockchain technology: the smart contracts run on a distributed network with thousands of computers that reach consensus, therefore altering the voting results, a huge amount of processing power is required (at least 50% of the entire network), not to mention that the community can fight back and with the help of honest miners can revert the attack. Moreover, the costs to conduct such an attack usually greatly exceeds the benefits;
- the decision-making approach is effective and decentralized because it is achieved through the community and it is not influenced by a single entity;
- maximum transparency because everyone can look at the code and see how the majority is established and how the decisions are made and executed.

4.6.2. Dynamic Validator Sets

Once a news article is fetched into the system by web scrapers, a series of blockchain accounts will be selected to be eligible for its trust assessment. This selection is performed by a specially devised algorithm that selects accounts based on their public keys and the news article hash. This approach will generate dynamic validator sets for each news article to be assessed. This way, we protect the system against a malicious attack that can use a large number of accounts to bias the final trust scores assigned to articles, since only a subset of accounts that match the selection scheme will be eligible for assessment.

The dynamic validator sets will change the validator sets constantly and randomly reducing the risk of centralization. Slashing is a set of techniques which incentivize actors to act honestly through the obligation to put some of their stake as collateral and enforce the threat of slashing their stake if they are proven to act maliciously and not follow the protocol. The threat of losing a significant part of their stake will incentivize users to act correctly.

4.6.3. Dishonest Behavior

Malicious actors that gain in spreading disinformation might try to attack and defraud the news article assessment system of the platform protocol by using multiple blockchain accounts that assign high trust scores on articles that contain disinformation. To address this aspect, the system implements dynamic validator sets so that the probability of forming a majority around a particular news article is extremely low. Moreover, each eligible selected account that enters the new article assessment procedure stakes tokens as collateral. All tokens from participating accounts are locked into an article assessment pool, until the assessment period ends. Once the assessment period ends, the tokens from the pool are redistributed among participants based on their assessments. Therefore, malicious voters, that will likely be on the minority side, will lose tokens in favor of the accounts whose assessment is clustered near a particular trust score. This in turn acts as a big deterrent to dishonest behavior.

4.6.4. Reputation System

By putting in place a reputation system, the platform protocol makes sure that its members are held responsible for their actions and that malicious users are reduced to insignificance. The voting power of users that consistently vote erroneously will decrease, while the voting power of the users that make correct assessments will increase over time. Activity tokens are rewarded for correct assessments. These tokens can be later transformed into governance tokens of the platform protocol. A smart contract incorporates the logic of tracking user reputation over time and distributes the activity tokens rewards or applies penalties.

The platform protocol, as it was designed, is expected to withstand the attacks from malicious individual or collective actors while also inflicting damage upon them by making them lose their tokens.

4.6.5. Governance Tokens and Quadratic Voting

The platform protocol governance features a token based voting system to ensure that major platform decisions are taken by the community. A scheme of governance tokens or activity tokens offering will favor the users that reach higher reputation levels. The governance token represents the users' ownership and stake in the platform protocol. A voting system based on Quadratic Voting [27,28] ensures that the cost to cast additional votes grows quadratically, while the number of the cast votes grows linearly. This in turn prevents a Sybil attack or 51% attack, where malicious actors try to gain control over the governance tokens to make changes in the platform's underlying architecture.

4.6.6. Randomized Voting System

After reaching a significant user base, a randomized voting system is planned to be employed. This system will assign voting rights on a specific article to several random accounts, taking into consideration aspects such as geographical location for specific news articles. Together with the aforementioned dynamic validator sets, this approach will significantly reduce the risk of vote manipulation, ensuring that only a percentage of the user base has voting rights on a specific news article, being randomly enforced each time a new news article arrives into the system.

4.6.7. DDoS (Distributed Denial of Service) Protection

Blockchain technology is inherently decentralized. Thousands of nodes participate in the network and a successful DDoS attack would mean a successful attack on all of them. Therefore, the blockchain component of the system is protected against DDoS attacks.

The only components exposed to the exterior are the front-end servers that provide the web portal application and client APIs accessible to anyone who wishes to integrate with the system. In the discussed system architecture, these front-end components are designed to be horizontally scalable, meaning that a successful DDoS attack would have to bring down all front-end servers.

4.7. APIs

4.7.1. Authentication API

The Offchain Core exposes an authentication API, which is used to obtain access to the other APIs, depending of the actor's role: Crawler API, Scraper API and Public API.

The role of this module is to provide authentication and authorization of the identified actors (i.e., crawler, scraper and public). The exposed API routes are shown in Table 1. The response status for invalid username-password pairs or for an invalid JWT is 401 Unauthorized. Unless otherwise specified, the response status for a proper request is 200 OK.

4.7.2. Crawler API

The role of this module is to provide the web crawler instances the seed data required to know which sites to crawl and to provide the means for the web crawlers to send the acquired and identified URLs to the Offchain Core. The web services that are exposed to the crawler instances are presented in Table 2.

4.7.3. Scraper API

The role of the Scraper module is to provide the web scraper instances the article URLs required to know from which to extract information, the extraction templates, and to provide the means for the web scrapers to send the extracted article information to the

Offchain Core. The web services that are exposed to the scraper instances are shown in Table 3.

Table 1. Authentication API routes.

URL Path	Method	Description
/api/auth/signin	POST	Receives two parameters (username and password) and returns a JWT with the user roles Request payload: username: string, password: string Response payload: JWT object
/api/auth/signup	POST	Receives three parameters (username, password and email) and creates a new user with the PUBLIC role Request payload: username: string, password: string, email: string Response status-payload: 201 Created or 409 Conflict—Username already exists

Table 2. Crawler API routes.

URL	Method	Description
/api/crawler/sites	GET	Returns the list of sites that have to be crawled by the user identified in the JWT used for authorization. Request payload: - Response payload: [{ siteName: string, urlBase: string, pageTypeClassifier: jsonString }, . . .]
/api/crawler/pages	POST	Receives a list of page URLs representing articles, which were identified by the crawler. The article list is associated to the user identified in the JWT used for authorization. The user can only send page URLs from the sites it was previously assigned to. Request payload: { siteName: string, pages: [urlString1, . . .] } Response payload: -

Table 3. Scraper API routes.

URL Path	Method	Description
/api/scraper/ extractor-templates	GET	Returns the list of extractor templates used to extract information from article URLs. Request payload: - Response payload: [siteName: string, templateVersion: versionNumber, removeElements: stringArray, title: stringArray, contents: stringArray, featuredImage: stringArray, publishDate: stringArray, author: stringArray , . . .]
/api/scraper/ article-urls? page=pageNumber &size=pageSize	GET	Returns the list of URLs representing articles that have to be scraped by the user identified in the JWT used for authorization. The list of URLs is organized by site and templateVersion, and are returned paginated; each page is identified by pageNumber and contains pageSize article URLs. Request payload: - Response payload: [siteName: string, templateVersion: versionNumber, urls: [urlString1, . . .] , . . .]
/api/scraper/articles	POST	Receives a list of objects representing article extracted information. The article information is associated to the user identified in the JWT used for authorization. Request payload: [siteName: string, articles: [url: string, title: string, contents: string, featuredImage: base64String, publishDate: string, author: string, extractedDate: string , . . .] , . . .] Response payload: -

4.7.4. Public API

Access to the stored articles is provided by the Public API module. It can be used by any entity to obtain article information, i.e., title, author, publish date, extracted date, publisher, featured image, contents. It is mainly used by the application frontend and the web services that are exposed are shown in Table 4.

Table 4. Public API routes.

URL Path	Method	Description
/api/public/articles	GET	Returns a paginated and filtered list of articles. Request payload: it allows the following query params: pageNumber, pageSize, publisher, start date, end date, title, author, and order by date. Response payload: [id: number, title: string, lastUpdated: string, extractedDate: string, publishDate: string, publisher: string, author: string, url: string, contentsHash: string , . . .]
/api/public/articles/{article-id}	GET	Returns information regarding the requested article-id. Request payload: - Response payload: single article information similar to the previous web service.
/api/public/articles/{article-id}/featured-image	GET	Returns the featured image. Request payload: - Response payload: byte array containing the featured image.
/api/public/articles/{article-id}/contents	GET	Returns the article contents with the HTML tags stripped. Request payload: - Response payload: Text representing the article contents.
/api/public/articles/{article-id}/contents-full	GET	Returns the article contents, including the HTML tags. Request payload: - Response payload: Text representing the article contents.

5. Use-Case Validation

5.1. Configuration of the Article Extraction Components

The Offchain Core is configured to handle seven news websites. For each website specific information is required in order to perform the URL and article extraction. This information is stored in JSON format with the following properties:

- name—news website name, which must be unique,
- urlBase—website base URL address,
- logoUrl—URL of the news outlet's logo,
- pageTypeClassifier—JSON object which contains information that differentiate article pages from other, irrelevant, pages from the website; it has two object arrays:
  - containsList—array with strings that are present in article web pages,
  - containsNotList—array with strings that must not be present in order to identify a page to be an article,
- extractorTemplate—CSS selectors and other metadata are used to identify the various properties that need to be extracted from an article page; all the JSON property values are in array format—the first string represents the selector and other optional strings represent from where the identified element(s) the information must be extracted:
  - removeElements—contents that must be removed from the extracted data,
  - title—article title,
  - contents—article contents,
  - featuredImage—URL location of the main image associated with the article,
  - publishDate—publish date as it appears on the article web page (if any),
  - author—author name or URL as it appears on the article web page (if any).

Both the crawler and the scraper are configured with the credentials to access the Crawler API and Scraper API, respectively, from the Offchain Core. Other configurations include access to the local database for each crawler/scraper and logging configurations.

In terms of actual gathering information from the news websites, the following parameters and corresponding values were used for each crawler/scraper instance:

- minimum waiting time between crawls from the same site (in milliseconds): 1800,
- maximum waiting time between crawls from the same site (in milliseconds): 2700,
- number of crawler instances running at the same time: 1
- recrawl type—can be interval (recrawlInterval is used) or time (recrawlTime is used): interval,
- recrawl time interval (in ISO 8601 format for durations): PT0H20M,
- recrawl time of day (in HH:mm:ss format for time): 08:00:00,
- recrawl time of day delta (in ISO 8601 format for durations); recrawl will occur at recrawlTime/recrawlInterval +/− rand(recrawlTimeDelta), depending on recrawlType: PT1H.

Those parameters were used so not to stress the news web servers with concurrent or many consecutive requests, and so not to get the crawler's/scraper's IP address blacklisted. Using the aforementioned configuration, the time between crawls is a random value between 1.8 and 2.7 s. In addition, the recrawl is triggered at a somewhat random time each day.

5.2. Extracted Articles

In order to demonstrate the efficiency of the proposed solution, seven news websites were crawled and scraped. Table 5 shows each news website, the total number of pages processed from each website, the number of extracted articles, and the percentage of pages representing articles compared to the total number of processed pages.

Table 5. Number of crawled pages and extracted articles for each considered news website.

Site Name	Site URL	Number of Pages	Number of Articles	Article Percentage
Adevarul	https://adevarul.ro	2516	482	19%
AgerPres	https://www.agerpres.ro	2636	490	19%
DCNews	https://www.dcnews.ro	1649	1302	79%
Digi24	https://www.digi24.ro	2695	2245	83%
G4Media	https://www.g4media.ro	2745	2391	87%
Hotnews	https://www.hotnews.ro	3521	1626	46%
Stiripesurse	https://www.stiripesurse.ro	1468	860	59%

The difference between the percentages among the considered websites is due to the fact that some of the websites had malformed URLs (especially representing email addresses or telephone numbers) or has more pages containing snippets of multiple articles on the same page. The information from these latter pages was ignored. Information regarding the 9.396 extracted articles can be obtained using the Public API.

5.3. Results Validation

News articles are extracted, stored in the Offchain Core, and validated by the federated AI and crowd wisdom. For each article, the Offchain Core calls an exposed service from the FiDisD Blockchain network, and sends the hash string obtained by applying the SHA3-256 algorithm on the article contents. The contents hash is stored on the blockchain network for later verification and in order to ensure trust in the system.

Validation of the results is obtained by the Frontend application, which is basically an aggregator portal that presents articles from those seven news websites. The Frontend application uses the Public API from the Offchain Core, which allows filtering and paginating the article list. Each piece of news shown contains the extracted article information and the

article score, which is stored on the blockchain. Anyone can validate the accuracy of the stored data by accessing the transactions that are stored on the blockchain network.

6. Discussion

There has been much innovation on the offensive side of threat actors, yet the defensive side is still lacking the innovation force to properly respond to these kinds of threats. This represents a key reason of why innovation is required in this space, and the platform proposed in this paper represents a powerful alternative to what is available on the fact-checking space, a rather reactive approach that does not have sufficient backbone to scale and effectively counteract the disinformation phenomenon.

The complex information environment that is aggressively developing in the digital ecosystem is threatening the way information is perceived. With high amounts of disinformation being spread on the Internet, AI-generated content that has the capacity to sky-rocket disinformation, and a diverse range of threat actors, humanity's access to information is threatened like never before. Not long from now, every piece of information on the Internet will be questioned because people will soon realize that it is no longer possible to separate truth from fiction. In this context, the platform described in this paper is set to fight disinformation using a coordinated approach between technologies and human intelligence in a unique way that will strengthen the defensive mechanisms against manipulated information and will offer the reader an easy way to identify and verify the authenticity of information.

In current stages, the anti-disinformation market is immature. The majority of the fact-checking initiatives are entirely dependent on funding, not being sustainable in the long-term, and also lacking scaling capability. There has not been any initiative that can be pinpointed as an important effect in effectively combating disinformation at scale, so the proposed system architecture can open the way for true scalable, decentralized and trusted platforms that can be used by all members of a society. Currently, there are no scalable and trusted solutions that can assess online information, and centralized solutions have the drawback of being controlled by a small number of entities that might influence the overall analysis. Therefore, our solution that uses decentralized blockchain technologies and a combination of human and artificial intelligence would naturally appeal to all the members of society that want to check the trust of online information.

Our system does not require publishers and news agencies to publish their news on a dedicated platform. Instead, our approach is to let the sites retain their identity and fetch their content using web crawlers that analyze and index the content on blockchain (proof of data) and off-chain (data). This content is further verified for authenticity and trust using AI based algorithms and human crowd wisdom.

The platform intends to help regular users to check the information available on the Internet, providing them with the necessary tools to verify the main triggers of disinformation and give a vote of confidence to the information they consider trustworthy. The platform aims to rally a fact-checking community where all the information can be aggregated and checked by human actors and AI modules. This will ensure an efficient alignment of fact-checking efforts and will allow people from all over the world to engage in the voting process, thus validating content that can be trusted in a truly decentralized fashion.

7. Conclusions

This paper describes a scalable system that can counter online disinformation using decentralized actors featuring artificial intelligence, crowd wisdom, smart contracts and blockchain technologies. The proposed solution uses distributed crawlers and scrapers to extract articles from news websites, federated AI and crowd wisdom to validate the information, and the blockchain network to ensure trust in the system.

The platform design enables both a reactive and proactive approach to fighting disinformation, firstly by significantly strengthening the fact-checking process through a

machine-human model, secondly by incentivizing publishers to adjust, verify and improve their content through an objective yet impactful way of voting that in turn reflects over their overall reputation.

Author Contributions: Conceptualization, C.N.B.; methodology, C.N.B. and A.A.; software, C.N.B. and A.A.; validation, C.N.B. and A.A.; formal analysis, C.N.B. and A.A.; investigation, C.N.B. and A.A.; writing—original draft preparation, C.N.B. and A.A.; writing—review and editing, C.N.B. and A.A.; supervision, C.N.B. All authors have read and agreed to the published version of the manuscript.

Funding: The research presented in this paper is part of the FiDisD project. FiDisD is the acronym for "Fighting disinformation using decentralized actors featuring AI and blockchain technologies". The FiDisD project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No 957228. FiDisD is developed in the context of TruBlo ("Trusted and reliable content on future blockchains") which is part of the European Commission's Next Generation Internet (NGI) initiative.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
BFS	Breadth First Search
CNN	Convolutional Neural Networks
DeFi	Decentralized Finance
DAO	Decentralized Autonomous Organization
DDOS	Distributed Denial of Service
FiDisD	Fighting disinformation using decentralized actors featuring AI and blockchain technologies
GloVe	Global Vectors for Word Representation
IoT	Internet of Things
IPFS	InterPlanetary File System
NFT	Non-Fungible Token
NLP	Natural Language Processing
PoA	Proof-of-Authority
SSI	Self-Sovereign Identity

References

1. Tsfati, Y.; Boomgaarden, H.G.; Strömbäck, J.; Vliegenthart, R.; Damstra, A.; Lindgren, E. Causes and consequences of mainstream media dissemination of fake news: Literature review and synthesis *Ann. Int. Commun. Assoc.* **2020**, *44*, 157–173. [CrossRef]
2. Shu, K.; Wang, S.; Lee, D.; Liu, H. Mining disinformation and fake news: Concepts, methods, and recent advancements. In *Disinformation, Misinformation, and Fake News in Social Media: Emerging Research Challenges and Opportunities*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 1–19. [CrossRef]
3. Fraga-Lamas, P.; Fernandez-Carames, T. Fake news, disinformation, and deepfakes: Leveraging distributed ledger technologies and blockchain to combat digital deception and counterfeit reality. *IT Prof.* **2020**, *22*, 53–59. [CrossRef]
4. Choraś, M.; Demestichas, K.; Giełczyk, A.; Herrero, Á; Ksieniewicz, P.; Remoundou, K.; Urda, D.; Woźniak, M. Advanced Machine Learning techniques for fake news (online disinformation) detection: A systematic mapping study. *Appl. Soft Comput.* **2021**, *101*, 107050. [CrossRef]
5. Guttmann, A. Survey: Index of Respondents' Trust towards Media in European Union (EU 28) Countries in 2019. 2023. Available online: https://www.statista.com/statistics/454409/europe-media-trust-index/ (accessed on 8 May 2023).
6. Walter, N.; Cohen, J.; Holbert, R.; Morag, Y. Fact-checking: A meta-analysis of what works and for whom. *Political Commun.* **2020**, *37*, 350–375. [CrossRef]
7. Nakov, P.; Corney, D.; Hasanain, M.; Alam, F.; Elsayed, T.; Barrón-Cedeño, A.; Papotti, P.; Shaar, S.; Martino, G. Automated fact-checking for assisting human fact-checkers. *arXiv* **2021**, arXiv:2103.07769.

8. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 8 May 2023).
9. Zheng, Z.B.; Xie, S.; Dai, H.N.; Chen, W.L.; Chen, X.P.; Weng, J.; Imran, M. An overview on smart contracts: Challenges, advances and platforms. *Future Gener. Comput.-Syst.-Int. J. eSci.* **2020**, *105*, 475–491. [CrossRef]
10. Hardle, W.K.; Harvey, C.R.; Reule, R.C.G. Understanding Cryptocurrencies. *J. Financ. Econom.* **2020**, *18*, 181–208. [CrossRef]
11. Chohan, U.W. *Decentralized Finance (DeFi): An Emergent Alternative Financial Architecture*; Critical Blockchain Research Initiative (CBRI) Working Papers; Critical Blockchain Research Initiative: Islamabad, Pakistan, 2021. [CrossRef]
12. Schar, F. Decentralized Finance: On Blockchain and Smart Contract-Based Financial Markets. *Fed. Reserve Bank St. Louis Rev.* **2021**, *103*, 153–174. [CrossRef]
13. Atlam, H.F.; Wills, G.B. Technical aspects of blockchain and IoT. *Adv. Comput.* **2019**, *115*, 1–39. [CrossRef]
14. Ferdous, M.S.; Chowdhury, F.; Alassafi, M.O. In Search of Self-Sovereign Identity Leveraging Blockchain Technology. *IEEE Access* **2019**, *7*, 103059–103079. [CrossRef]
15. Attaran, M. Blockchain technology in healthcare: Challenges and opportunities. *Int. J. Healthc. Manag.* **2020**, *15*, 70–83. [CrossRef]
16. Vijay, C.; Suriyalakshmi, S.M.; Elayaraja, M. Blockchain Technology in Logistics: Opportunities and Challenges. *Pac. Bus. Rev. Int.* **2021**, *13*, 147–151.
17. Ante, L. Non-fungible token (NFT) markets on the Ethereum blockchain: Temporal development, cointegration and interrelations. *Econ. Innov. New Technol.* **2022**. [CrossRef]
18. Shae, Z.; Tsai, J. AI Blockchain Platform for Trusting News. In Proceedings of the 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), Dallas, TX, USA, 7–9 July 2019; pp. 1610–1619. [CrossRef]
19. Chen, Q.; Srivastava, G.; Parizi, R. M.; Aloqaily, M.; Ridhawi, I.A. An incentive-aware blockchain-based solution for internet of fake media things. *Inf. Process. Manag.* **2020**, *57*, 102370. [CrossRef]
20. Agrawal, P.; Anjana, P.S.; Peri, S. DeHiDe: Deep Learning-based Hybrid Model to Detect Fake News using Blockchain. In Proceedings of the International Conference on Distributed Computing and Networking 2021, Nara, Japan, 5–8 January 2021; pp. 245–246. [CrossRef]
21. Shahbazi, Z.; Byun, Y.-C. Fake Media Detection Based on Natural Language Processing and Blockchain Approaches. *IEEE Access* **2021**, *9*, 128442–128453. [CrossRef]
22. Paul, S.; Joy, J.I.; Sarker, S.; Shakib, A.-A.-H.; Ahmed, S.; Das, A.K. Fake News Detection in Social Media using Blockchain. In Proceedings of the 2019 7th International Conference on Smart Computing & Communications (ICSCC), Miri, Malaysia, 28–30 June 2019; pp. 1–5. [CrossRef]
23. Saad, M.; Ahmad, A.; Mohaisen, A. Fighting Fake News Propagation with Blockchains. In Proceedings of the 2019 IEEE Conference on Communications and Network Security (CNS), Washington, DC, USA, 10–12 June 2019; pp. 1–4. [CrossRef]
24. Katal, A.; Singh, J.; Kundnani, Y. Mitigating the Effects of Fake News using Blockchain and Machine Learning. In Proceedings of the 2021 2nd International Conference for Emerging Technology (INCET), Belagavi, India, 21–23 May 2021; pp. 1–7. [CrossRef]
25. Ramadhan, H.F.; Putra, F.A.; Sari, R.F. News Verification using Ethereum Smart Contract and Inter Planetary File System (IPFS). In Proceedings of the 2021 13th International Conference on Information & Communication Technology and System (ICTS), Surabaya, Indonesia, 20–21 October 2021; pp. 96–100. [CrossRef]
26. Dwivedi, A.D.; Singh, R.; Dhall, S.; Srivastava, G.; Pal, S.K. Tracing the Source of Fake News using a Scalable Blockchain Distributed Network. In Proceedings of the 2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), Delhi, India, 10–13 December 2020; pp. 38–43. [CrossRef]
27. Lalley, S.; Weyl, E.G. Quadratic Voting: How Mechanism Design Can Radicalize Democracy. *Am. Econ. Assoc. Pap. Proc.* **2018**, *108*, 33–37. [CrossRef]
28. Lalley, S.; Weyl, E.G. Nash Equilibria for Quadratic Voting. *arXiv* **2014**, arXiv:1409.0264.

mathematics

MDPI

Article

Decentralized News-Retrieval Architecture Using Blockchain Technology

Adrian Alexandrescu * and Cristian Nicolae Butincu *

Department of Computer Science and Engineering, Faculty of Automatic Control and Computer Engineering, Gheorghe Asachi Technical University of Iasi, 700050 Iasi, Romania

* Correspondence: adrian.alexandrescu@academic.tuiasi.ro (A.A.); cristian-nicolae.butincu@academic.tuiasi.ro (C.N.B.)

Abstract: Trust is a critical element when it comes to news articles, and an important problem is how to ensure trust in the published information on news websites. First, this paper describes the inner workings of a proposed news-retrieval and aggregation architecture employed by a blockchain-based solution for fighting disinformation; this includes a comparison between existing information retrieval solutions. The decentralized nature of the solution is achieved by separating the crawling (i.e., extracting the web page links) from the scraping (i.e., extracting the article information) and having third-party actors extract the data. A majority-rule mechanism is used to determine the correctness of the information, and the blockchain network is used for traceability. Second, the steps needed to deploy the distributed components in a cloud environment seamlessly are discussed in detail, with a special focus on the open-source OpenStack cloud solution. Lastly, novel methods for achieving a truly decentralized architecture based on community input and blockchain technology are presented, thus ensuring maximum trust and transparency in the system. The results obtained by testing the proposed news-retrieval system are presented, and the optimizations that can be made are discussed based on the crawling and scraping test results.

Keywords: blockchain; cloud computing; decentralized architecture; information retrieval; news aggregation; OpenStack; web crawler; web scraper

MSC: 68U35

Citation: Alexandrescu, A.; Butincu, C.N. Decentralized News-Retrieval Architecture Using Blockchain Technology. *Mathematics* **2023**, *11*, 4542. https://doi.org/10.3390/math11214542

Academic Editors: Zaki Malik and M. Mustafa Rafique

Received: 28 September 2023
Revised: 31 October 2023
Accepted: 2 November 2023
Published: 3 November 2023

1. Introduction

Disinformation in the online environment is spread mostly by actors with malicious intent or with specific not-so-honest agendas. The fake news distributed all over the Internet can have significant consequences regardless of fields, e.g., political, social, business, and media [1]. Detecting fake news is an important subject in any global context. Specific dishonest pieces of information can be shared by a news publication, and from there, other news outlets can unknowingly spread misleading information, which can lead to serious effects. There is ample research when it comes to detecting fake news and rumors, as well as identifying various disinformation strategies [2] and countermeasures [3,4]. Techniques based on machine learning, deep learning, and natural language processing (NLP) are employed to solve this problem [5]. Some methods approach this as a classification problem, while others use data mining to make predictions or to assess the credibility of a piece of news. None of them provide maximum accuracy and, in many situations, not even pass an acceptable threshold.

In recent times, cognitive and informational warfare has become the preferred tool of action of rogue entities that are trying to destabilize societies. Trust in mainstream media is lower than ever, and the click-based ad revenue model that online media relies upon favors user engagement at the expense of thoroughly checking and vetting published information. This, in turn, generates a blend between real and fake news that affects the

ability to distinguish the truth from disinformation and misinformation. A survey [6] conducted in 2019 revealed that only 19% of EU adults have high trust in the news media, while 40% had low or no trust at all.

A concept that, by its inherent nature, ensures trust, transparency, and traceability is blockchain technology. A blockchain is a decentralized structure based on distributed ledger technology (DLT) [7,8] that records and replicates data across many computing nodes. In general, it employs a peer-to-peer (P2P) network and consensus algorithms to guarantee that the data are the same on all computing nodes. In a blockchain, the recorded data are organized in a set of blocks linked together in a chain-like structure. These blocks can only be appended at the end of the chain and cannot be updated or deleted afterward. This implies that once recorded, the data cannot be altered, making the write operations append-only, thus providing traceability and transparency. The security is enforced using cryptographic hashes, keys, and digital signatures and guarantees the data's validity and integrity. Each node in the blockchain network replicates an identical copy of the data independently of other nodes. The consensus algorithms employed by the network ensure that the data preserved at each node is identical across the network. This implies that read operations of blockchain data exhibit high scalability, as any node in the network can be queried independently. However, the write operations need to go through the consensus algorithms that limit the maximum number of transactions per second. This metric varies between different DLT/blockchain implementations [9] and mainly depends on the consensus mechanisms [8,10]. The primary advantage of blockchain technology is the lack of central authority, and therefore, it does not expose a single point of failure. There are many applications of blockchain technology in areas like government, healthcare, the Internet of Things, transport systems, or cloud computing [11]. One of the areas with great potential for innovation is ensuring trust in the information published on the Internet.

At the moment, there are not enough scalable fact-checking platforms nor consistent standards for tracking, labeling, identifying, or responding to disinformation. In paper [12], the authors present a scalable fact-checking approach; however, the scalability comes from semantic aggregation, while the fact-checking is performed by humans. A computational fact-checking approach based on knowledge networks is given in [13]; however, in real-world fact-checking scenarios, the best performance for the AUC (area under the curve) of the ROC (Receiver Operating Characteristic) [14] curve was between 0.55 and 0.65, where 0.5 indicates random performance. Assuming a scalable system, important questions need to be answered, like: Who sets the standards? Who is responsible for validation? Who controls the system? Who manages potential disputes? The decentralized nature of the blockchain can address many of these concerns transparently, as it eliminates the need for a single, trusted institution to make critical decisions.

To have a fake news detection system, the first step is obtaining the said news articles. This implies using data extraction methods (i.e., crawlers and scrapers) to go through the web pages of a news website and extract the article information. In this sense, Figure 1 is a generic news-retrieval system diagram for extracting URLs and article information from a specific news website. The crawling process implies going through the website and extracting the URLs, while the scraping process consists of extracting the actual useful data (i.e., the article information). Usually, there is a list of news websites, which is used as a seed for the crawler, and the crawler stays in the confinements of those websites.

Regarding trust in the system, the main issue is that, now, there is only one actor (the one controlling the crawling/scraping processes), and that sole actor is not enough to ensure trust in the system.

In this direction, the FiDisD project [15] has been developed as an initiative to address the problem of misinformation and disinformation in online media. At its core, the project uses blockchain technology combined with both crowd intelligence and federated artificial intelligence. Its blockchain-based architecture creates a decentralized ecosystem that ensures scalability, transparency, and trust.

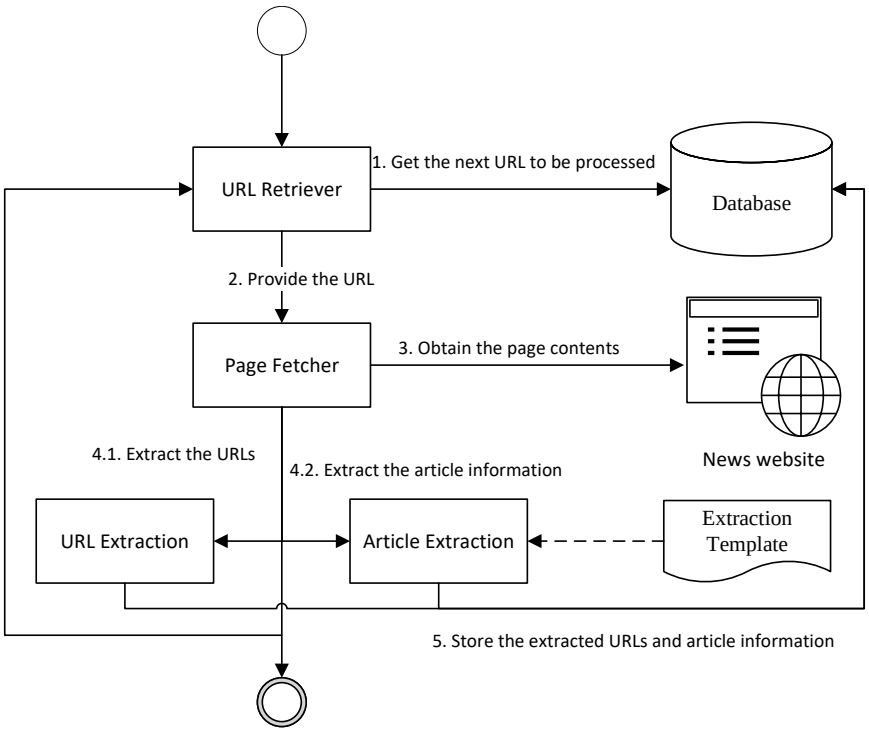


Figure 1. Existing news-retrieval system diagram for extracting URLs and article information from a specific news website.

The research presented in this paper has, as its primary focus, the in-depth presentation of the news-retrieval system used by the FiDisD solution for fighting disinformation while also discussing potential architectural improvements. In [16], the authors presented only the overall architecture of the system and the communication between the news-retrieval component, blockchain, federated AI, crowd wisdom, web portal, and end users.

The most important and novel contribution that our paper brings is the method of achieving decentralization in news article retrieval. Other major contributions and novelty that the current paper showcases are:

- the proposed news-retrieval solution with its features: community involvement for decentralized URL and article extraction, distributed architecture, multiple web crawling and web scraping instances, component communication with RESTful APIs, interaction with the blockchain;
- the extraction template used for exact article details extraction, which overcomes the inconsistencies in the HTML web pages' structure;
- the provided solution for cloud deployment, especially on the open-source OpenStack cloud, with a focus on the services that have to be employed for increasing efficiency;
- the novel proposed solution for a community-based truly decentralized architecture.

First, this paper looks at existing content extraction and news-retrieval solutions and discusses the advantages and disadvantages compared to the proposed solution. The theoretical approach, the proposed system architecture, the components, and the information flow between the different services are then presented. Several solutions for deploying the proposed system are discussed, with the focus being mainly on the deployment of the open-source OpenStack cloud solution [17]. A discussion regarding having a community-based truly decentralized architecture follows, and possible solutions are presented. The proposed system is then evaluated by processing the articles from several news websites, and various optimizations are proposed and discussed. Finally, the conclusions of our research are highlighted.

2. Related Work

Part of the research presented in this paper is an in-depth presentation of the decentralized news-retrieval solution, which is one of the main components of the FiDisD project. The overall architecture of that project is described in [16] and consists of distributed crawlers/scrapers, an Offchain system, crowd wisdom, and artificial intelligence modules for detecting fake news, and the blockchain platform for ensuring trust and transparency. The crawling and scraping approaches are presented in that paper summarily without showing the underlying system and without any discussion regarding performance and efficiency.

In terms of existing news-retrieval research, the authors in [18] describe a crawler and extractor from news websites that require only the root URL. They used generic extractors that use heuristics to determine from where, on the web page, the information is to be extracted. Based on the authors' study, the Newspaper library (now evolved to Newspaper3k [19]) proved to obtain the best results. The authors from [20] use machine learning in extracting article-like information from multiple websites. The disadvantage of that proposed approach is the lack of perfect accuracy of the obtained results.

Our previous research regarding extracting specific information from websites containing products [21,22] showed that, for some websites, custom extraction methods must be employed. For example, on a few websites, the product information was available when the page loaded or when a specific HTML element finished loading. This required the use of a tool for automating web browsers and the execution of JavaScript code. Given the heterogeneity of websites in general and news websites in particular, generic extractors cannot provide 100% accuracy, which is needed when it comes to having a solution that is focused on providing trust.

Another example is in [23], where the authors propose a generic method for extracting news information, which is based on heuristics. The obtained results did not yield full success, and for 10% of the considered websites, the results were not accurate. This is unacceptable if the goal is to have a maximum success rate.

There are many generic web content extraction tools and libraries for extracting text from web pages, as is shown in [24]. In that paper, the authors present a web scraping library and compare various tools for text extraction. None of them provided optimum results and, therefore, cannot be employed by the fighting disinformation system considered in our paper.

The best information retrieval results are obtained using targeted extraction based on the structure of each website with a valuable payload. Even this approach proves challenging. The authors in [25] identify 13 issues regarding web scraping, e.g., data cleaning, web page structure, memory, and time consumption. They also propose an algorithm for extracting data, which auto-updates the parameters that are used to locate the target data on the web page. Basically, the extraction template defines the "start" and "end" unique identifiers for each attribute. Given the heterogeneity of the websites' structure, specifying the surrounding entities does not offer enough flexibility. For example, for some pages from the same website, the author's name is not present, there are multiple authors for the same article, or the author's name is surrounded by the same HTML entities as another piece of information. Another issue with their solution is the use of a headless browser for obtaining the data, which adds a significant overhead compared to simply obtaining the response from accessing the URL (Uniform Resource Name).

Other research focuses on dynamically determining the underlying web templates used by the websites [26]. The traditional approach involves clustering the web pages represented by the DOM (Document Object Model) tree. Even if the generic web template for a cluster of pages is determined, it would still require pinpointing the exact location of the article features (e.g., title, author, contents).

In terms of decentralized information retrieval and crawling, there is little to no published research. The authors from [27] present a peer-to-peer model for crawling, which uses geographical proximity and protocols based on distributed hash tables to

exchange information between peers. This approach does not tackle the trust issue, but it is an interesting method for achieving full decentralization; more regarding this issue is discussed in Section 5. A paper related to decentralization, blockchain, and community involvement is [28], where the authors propose a method for constructing a knowledge graph based on crowdsourcing and smart contracts, which are used for recommendation systems. Involving the community is a significant step towards full decentralization in detecting fake news. Another issue is regarding traceability and a mechanism to determine how fake news spreads can increase the public's trust in online news media outlets. One such solution is described in [29], where a Python-based web crawler is employed to help with the traceability problem.

Existing distributed crawling solutions focus on performance and how each crawl process knows which URLs to handle. In [30], a hybrid peer-to-peer web crawler is deployed on the AWS (Amazon Web Services) cloud solution. Another example is a distributed news crawler in which the navigation between URLs is performed based on the URL's domain priority URL queues and by deploying it in the fog and cloud layers [31]. A summary of existing research related to distributed crawling is shown in [32]. Other distributed crawling approaches include crawling the hidden web [33], a web crawling solution deployed by a cloud service [34], and a crawler that extracts information only regarding certain topic by classifying the crawled articles [35].

The main novelty of our proposed solution compared to existing methods consists of the decentralized nature of the news extraction process, which involves multiple actors, a data allocation, and a majority-rule algorithm for ensuring trust in the extracted data and the separation of the actual crawling (extracting the URLs) and scraping (extracting the article information). In terms of the individual crawling and scraping processes, the algorithms are based on the traditional approach presented in Figure 1.

3. Proposed News-Retrieval System

The proposed news-retrieval architecture is critical for obtaining the article information needed to detect fake news. As previously mentioned, this proposed solution is an in-depth look at the data acquisition methods and components from the FiDisD project described in [16]. The main aspects discussed here are the theoretical approach to the proposed retrieval system, the overall retrieval system architecture with the reasoning behind the design decisions that were taken, the three distinct system components (i.e., OffchainCore, WebCrawler, and WebScraper), and the communication between them, the exposed API with authorization based on the user's role, the extraction template structure, and the method for data allocation for processing by the crawlers and scrapers.

The novelty of the news-retrieval system proposed in this paper encompasses two aspects: separating the URL extraction (performed by the crawler) from the article extraction (performed by the scraper) to increase scalability and, more importantly, allowing third-party actors to perform the crawling and scraping. The latter provides dynamic system distribution and makes the solution decentralized because multiple independent actors process the same URLs, and the majority decides on the correctly extracted information. Therefore, no bad actors can negatively influence the extraction process.

3.1. Theoretical Approach to the Proposed Decentralized Retrieval

Let $U = \{u_1, u_2, \ldots, u_n\}$, the set of URLs belonging to a particular website, and $A = \{a_1, a_2, \ldots, a_m\}$ a subset of U containing the URLs of pages identified as containing articles: $|U| = n, |A| = m, A \subseteq U, m \leq n$.

Let $C = \{c_1, c_2, \ldots, c_k\}$ and $S = \{s_1, s_2, \ldots, s_l\}$, the set of crawlers and scrapers, respectively, enrolled into the system: $|C| = k, |S| = l$. Although not a requirement, usually $k < l$, i.e., the number of scrapers exceeds the number of crawlers. This is because the number of potential article link URLs that need to be processed by scrapers is, in general, greater than the number of URLs processed by crawlers.

Let O be a random oracle used to generate uniformly distributed random values based on input URLs domain: $O : D \rightarrow b_{256}$, where D is the domain of URLs represented as ASCII strings and b_{256} is the domain of 256-bit values.

To determine the allocation of URLs to crawlers and scrapers, a simple single-valued modulo function applied on the oracle's output can be used: $svf : b_{256} \rightarrow b_{nb}$, where b_{nb} is the domain for crawler and scraper identifiers represented as a nb bit values, e.g., b_{32} represents the domain of 32-bit values. This modulo function uses k and l as modulo parameters for its second operand, for crawlers and scrapers, respectively. An efficient implementation is possible when the modulo's second operand is a power of two; in this case, the modulo can be obtained by simply applying a bit mask to the first operand: $v \bmod 2^t = v \,\&\, (2^t - 1) = v \,\&\, 0b\underbrace{11\cdots1}_{t \text{ times}}$, where & is the bitwise AND operator. However, this naive implementation is not fit for an in-production system as it does not account for the possibility that the selected crawler/scraper might be unavailable, in which case the overall system function would be impaired as the URLs remain unprocessed. To overcome this issue, our system uses a multi-valued modulo function $mvf : b_{256} \rightarrow b_{nb}$ that selects multiple crawler/scraper identifiers. This function outputs, for each input value, a set IDS of crawler/scraper ids with $|IDS| = sp$, where sp is the selective power of the function and is a configurable system parameter.

There are several ways to define such a multi-valued function. A simple approach would be to use a sliding bit window of size nb that sweeps over the b_{256} input to select the crawler/scraper identifiers. However, this approach increases the probability for clusters of crawlers/scrapers to be selected together and limits the selective power of the function to a maximum value of $256 - nb + 1$. Another approach would be to use a set of sp random oracles, $OS = \{O_1, O_2, \ldots, O_{sp}\}$, instead of just one, combined with the single-valued modulo function svf defined above. The end results are assembled to provide a multi-valued output for an input URL, u: $\{svf(O_1(u)), svf(O_2(u)), \ldots, svf(O_{sp}(u))\}$. This approach does not limit the selective power in any way and does not suffer from selection clustering. To increase the selective power, one would add more random oracles into the system. If managing a large number of random oracles is not desired, a third approach would be to use the random oracle O defined above, along with another random oracle $R : b_{256} \rightarrow b_{256}$ that can be applied recursively over its own outputs, and the single-valued module function svf, also defined above. As in the previous case, the end results are assembled to provide a multi-valued output for an input URL, u: $\{svf(R(O(u))), svf(R(R(O(u)))), \ldots, svf(\underbrace{R(R(\cdots R(}_{sp \text{ times}} O(u) \underbrace{))\cdots))}_{sp \text{ times}}\}$. As with the previous approach, there are no limits to selective power and no selection clustering. Increasing the selective power implies making more recursive calls to the R oracle. However, the downside of this approach, as opposed to the first two approaches, is that due to the recursive nature of the R oracle, it cannot be parallelized.

Independent of the selection scheme, the probability of a URL remaining unprocessed is inversely proportional to the selective power sp and represents the probability for all selected sp crawlers/scrapers to be unavailable. If we model the probability for a crawler/scraper to be available at a certain time as $P(A) = pa$, then the probability for it not to be available is $P(\overline{A}) = 1 - pa$. It follows that the probability for all sp crawlers/scrapers not to be available at a certain time is $P(\overline{A})^{sp} = (1 - pa)^{sp}$. Therefore, the probability for a URL to remain unprocessed drops exponentially as the selective power sp increases, and the probability to be processed by at least one crawler/scraper is $1 - P(\overline{A})^{sp} = 1 - (1 - pa)^{sp}$.

This approach to crawler/scraper selection solves two problems: first, it ensures that multiple crawlers/scrapers process the same URLs. This is necessary to detect and ban rogue actors that might try to inject invalid data into the system. The correct results are considered the ones that have the majority, and any crawler/scraper that deviates from this is penalized. By employing the aforementioned selection mechanism, a Sybil 51% majority attack is extremely improbable, and the probability for an attacker to control most selected

crawlers/scrapers drops exponentially as the selective power sp increases. Second, it builds trust in the system, as the stored data are vetted by multiple randomly chosen actors.

A URL hacking attack is a technique employed by an attacker that alters the value of a URL (e.g., by adding invalid anchors and/or unused query parameters) while keeping its semantics in the hope that the selection function of the system selects crawlers/scrapers that are under his control. To prevent this type of attack, each URL is stored in the system using a canonical form.

3.2. General News-Retrieval Architecture

The news-retrieval components have the role of obtaining the news article information in a decentralized manner, storing the acquired data, and sending proof of what has been stored to the blockchain network. The overall architecture of the proposed news-retrieval system is presented in Figure 2. It encompasses the following components:

- OffchainCore—the main component, which contains the business logic for managing the crawlers and scrapers,
- WebCrawler—multiple instances that extract URLs from web pages and identify the relevant web pages that contain article information,
- WebScraper—multiple instances that extract the article information from article-containing web pages,
- ClientAPI—used by multiple actors to access the stored article information,
- Blockchain network—used to store the article hashes.

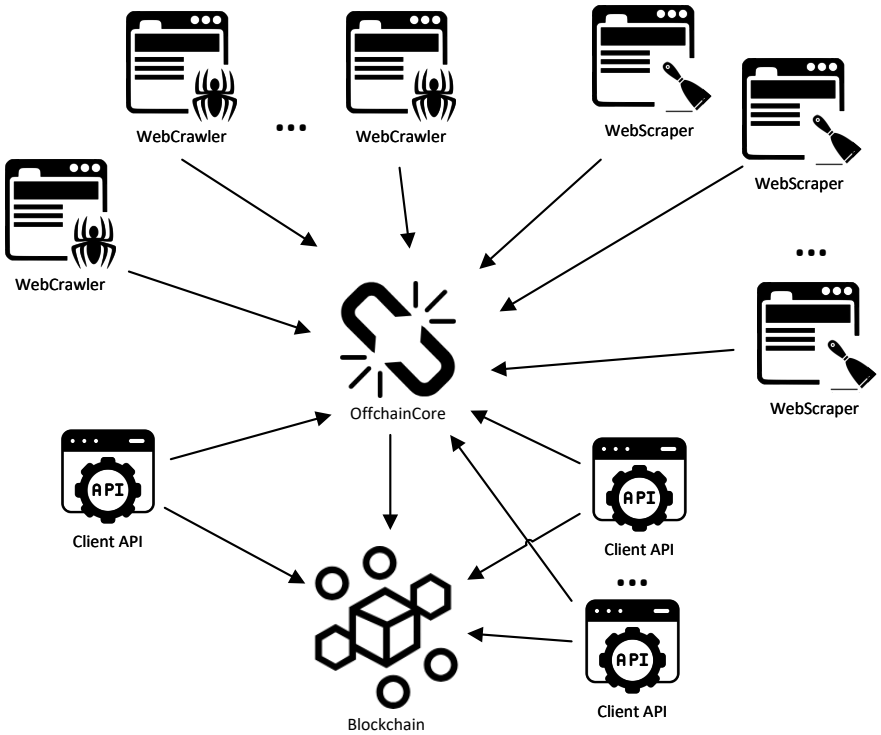


Figure 2. Proposed news-retrieval system architecture. The main components/applications are pictured, and the arrows show the direction of communication—the base of the arrow represents the entity initializing the communication.

Other components of the FiDisD project include Federated AI modules, which analyze the news arriving into the system and assign trust scores, Crowd Wisdom, which consists of human actors that also participate in news analysis and activity token rewards, and a public Frontend Web Portal, which serves as a news aggregator and a view to the OffchainCore database and the article trust scores. These components are beyond the scope of the current paper but are relevant to the overall view of the considered environment.

To have a distributed information retrieval system, multiple actors extract the data (i.e., URLs and article information from web pages) and send it to the OffchainCore component. Each actor processes specific websites, and the crawling is restricted to the URLs belonging to those websites. A first important decision was made to separate the URL extraction, performed by the WebCrawler, from the article information extraction, performed by the WebScraper. This way, a certain number of actors can obtain the URLs while a different number of actors can extract the article data. This is useful because each news website has its own particularities depending on the number of relevant web pages (the ones that contain article information) compared to the total number of pages and the different HTML page structures, which can influence the time it takes to extract information from a page, or even the article's featured image, which can take different amounts of time to download and process depending on the website.

However, having a distributed and separated crawler and scraper is not enough to achieve decentralization. The most important step in this direction is the decision to have independent persons/organizations perform the crawling and scraping. This raises the question of trust in those entities because entities can maliciously report that they extracted a specific article, which is altered with fake information introduced by that entity. Trust must be ensured at three levels: the news publisher, the crawlers, and the scrapers. For the former, the Federated AIs and the Crowd Wisdom ensure trust by verifying the published information. For the latter two, trust in the crawled and scraped data is obtained by majority rule. Therefore, each URL is processed by multiple crawlers, and each article URL is processed by multiple scrapers. Once a specific number of responses are received from processing a specific URL, the majority of the received responses are considered to be the trusted response. Because OffchainCore is the one that pseudo-randomly assigns the URLs to the crawlers and scrapers, no outside entity can intervene and negatively influence the majority decision. If one or more mal-intent scrapers send to the OffchainCore fake article information, then that information is ignored because it is different from the formed majority. Each crawler and scraper actor has an associated user in the OffchainCore. Therefore, if attempts to undermine the system by a specific actor are detected, then that actor can be banned from the system.

3.3. Data Allocation to Crawlers/Scrapers

Depending on the number of WebCrawler and WebScraper actors in the system, the same data will be processed by multiple actors. This is configurable at the OffchainCore component.

The crawling process depends on the number of considered news websites and the number of registered crawler actors. Ideally, each actor should process a limited number of sites, e.g., five sites, so as not to use too many resources from the computing instance the crawler runs on. Each WebCrawler handles a specific number of sites and has one thread/process running for each site. Every crawl request is succeeded by a small waiting period of between one and three seconds because, otherwise, the crawled website might ban the crawler's IP address. To obtain a majority regarding a URL that was extracted and identified as representing an article page, there must be enough WebCrawlers that obtain that URL so no mal-intent actors can take advantage of the system. Let us consider the scenario in which one site is assigned to be crawled by six random actors. In this case, if a URL is identified as containing an article and it is received by the OffchainCore from four of the actors, then we have a majority, and it can be marked as processable, subsequently, by the scrapers.

There is also the off chance that most actors processing a site are composed of bad actors. In this case, a flag would be raised because the minority of good actors did not extract that URL. At this point, the system would consider, based on probabilities, that the minority group is formed of bad actors and flag them accordingly. This can be partially overcome by constantly switching the sites that need to be crawled among the crawlers so no crawler handles a specific site for a long period of time. On the other hand, the paradigm must be updated so that when a crawler receives a new site, it should not process

the already extracted URLs before the site allocation update. This can be dealt with by having OffchainCore provide a list of hashes on the URLs that have already been processed.

In terms of the scraping process, scrapers are not bound by extracting article information from a specific list of websites. When OffchainCore receives a list of article-containing URLs, it assigns each URL to a random subset of scraper actors in a similar manner to assigning sites to crawler actors. The same principle applies to ensuring trust in the system and to determining the bad actors. For a URL, each WebScraper extracts the article information and sends it to the OffchainCore together with the hash on the article contents. The OffchainCore then checks the integrity of the received information and stores it in the database and the article store (i.e., the file system). If there are different hashes for the article contents associated with the same URL, then the hash reported by most scrapers identifies the correct article content, and it is stored on the blockchain network.

3.4. Extracted Article Information and Extraction Template

The information extracted from each article page includes the article title, article contents, featured image, publishing date, and author. Only the first two are mandatory because some sites do not have a featured image, publishing data, or an author specified for every article. This information is extracted only from the pages identified by the crawler as having article data. The identification is made using the information provided in the `pageClassifier` property of the object describing the site that needs to be processed. The value of this property is another object containing two keys `containsList` and `containsNotList`, which describe the HTML code and text that have to be present or have to be missing in order for a page to be classified as containing an article. Basically, if certain strings are present in the web page contents and/or other strings are not present, that page is classified as having an extractable article.

To extract the article information, the scraper needs to know where the data are located inside the web page. This is why, for every website, there is an extraction template, which pinpoints the exact location in the HTML structure of the extractable features and filters unwanted data, like ads, scripts, or irrelevant elements. The location is given in the form of CSS selectors, which identify the targeted HTML elements. The extractor module used by the WebScraper is designed to be fast and accurate.

The extraction template is represented by a JSON object in which the keys are the features that need to be extracted, and the corresponding values are arrays describing the extraction process. A special key, `removeElements`, provides the means to target the HTML elements that must be removed before extraction. An example of a value for the `removeElements` key is `["script, style, div.ad-wrapper"]`. This removes all the `script` and `style` elements, and all the `div` elements that have the attribute `class="ad-wrapper"`. The array describing the extractable feature is composed of (1) a CSS selector, (2) a custom property, (3) a Boolean value stating if the extractable value can be missing or not, and (4) a regular expression. The custom property is used to further filter the HTML element identified by the CSS selector, and it can have one of the following values with the specified processing outcome:

- `text`—the contents of the HTML element as text (without any HTML tags),
- `html`—the contents of the HTML element as is (including the HTML tags),
- `attr:`*attributeName*—the value of attribute identified by the specified *attributeName* of the HTML element,
- `attr:src`—the value of the `src` attribute of the HTML element, which is resolved as an URL (normalized to an absolute URL),
- `attr:href`—same value as the previous one only for the `href` attribute.

The last possible element of the array is represented by a regular expression that further processes the value extracted using the CSS selector and the custom property. For example, the URL value corresponding to the featured image of an article is the value of the `srcset` attribute of multiple `source` elements belonging to a `picture` element. The goal is to obtain only the first URL and only that URL without any extra information. In

the considered example, after the URL, there is a space character followed by the image dimension. We need only the URL string. Using the `([\^\\s]*)\\s?.*` regular expression as the fourth element of the extraction array achieves that.

3.5. Inter-Component Communication

If we look at the WebCrawler as a black box, its role is to extract URLs from the pages of websites provided by OffchainCore. The main communication between the OffchainCore and the WebCrawler actor is presented in Figure 3. Basically, a WebCrawler actor authenticates itself to the OffchainCore. Then, it has two communication workflow loops, which are handled in parallel. The first one is to obtain the list of sites to be crawled associated with that actor from the OffchainCore. This request is performed with a lower frequency, given the fact that the site list seldom changes. The second one is to send batches of URLs representing pages identified as articles to the OffchainCore. This request is sent only if a sufficient number of URLs are to be sent or if a specific time period has passed since the last sent information and there are URLs available.

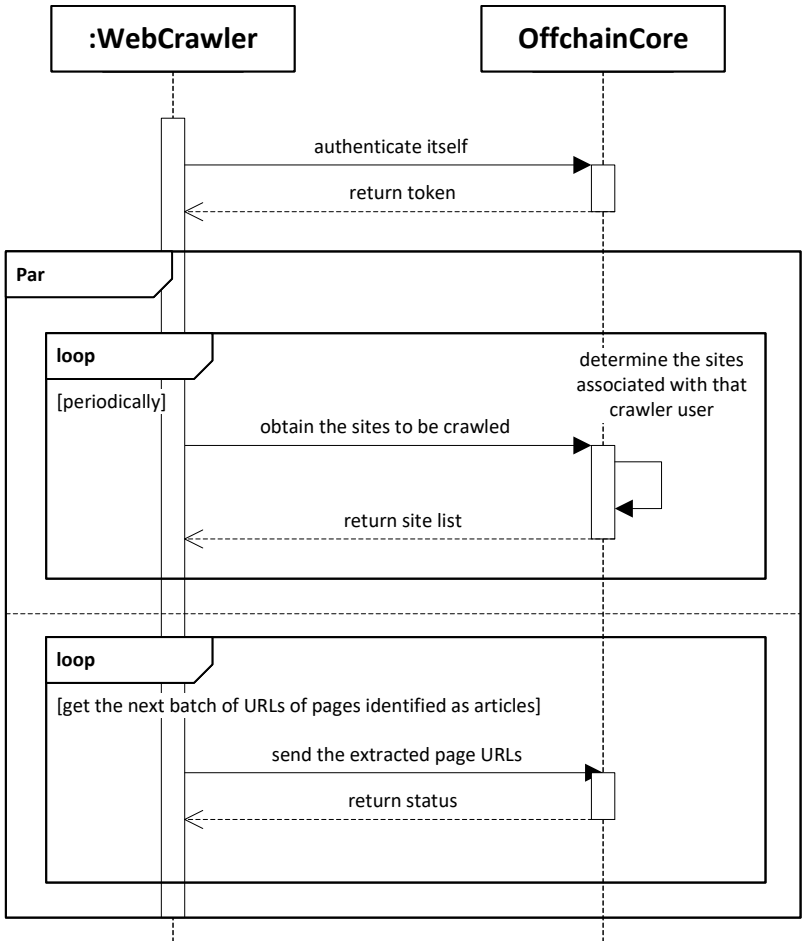


Figure 3. UML Sequence diagram showing the main data flow between the WebCrawler instance and the OffchainCore component.

The WebScraper's role is to extract article information from the URLs provided by the OffchainCore. Three parallel loops handle the WebScraper–OffchainCore communication (Figure 4), as well as the authentication logic of the WebScraper actor to the OffchainCore. The first communication loop is to periodically obtain the extraction templates used to determine the exact location of the pieces of article information on the web page. This request can be performed occasionally because the page structure corresponding to a site rarely changes. The second communication situation is to obtain the URLs representing

article pages from OffchainCore. Each request obtains the next batch of URLs, and the request is made, usually, when all the URLs received so far have been processed. Each WebScraper actor receives only the article URLs associated with that user. Lastly, as the article information is extracted, the third loop obtains the next batch of article data and sends it to OffchainCore. Similarly, to the WebCrawler, the request is made if there is enough data to create a batch or if enough time has passed since the last request. This is so the OffchainCore is not overwhelmed with too many requests from all the crawlers and scrapers.

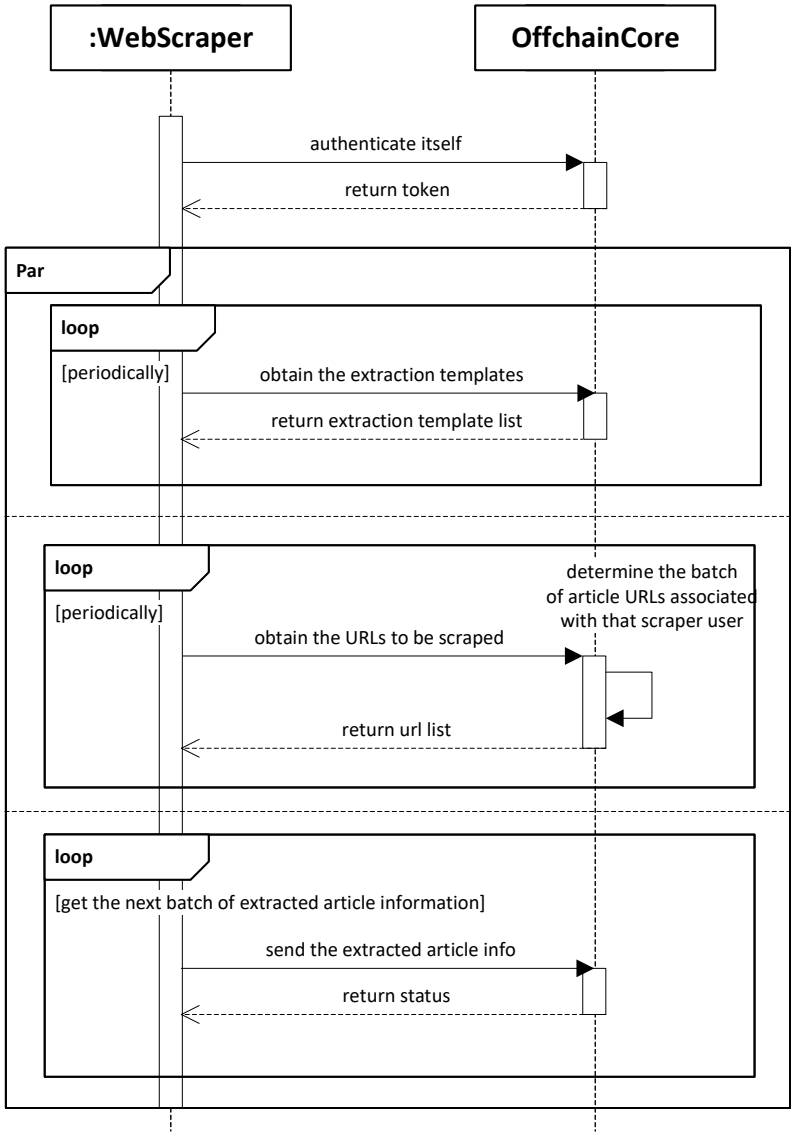


Figure 4. UML Sequence diagram showing the main data flow between the WebScraper instance and the OffchainCore component.

Even though both the crawler and the scraper require input from the OffchainCore, the communication between the OffchainCore and the WebCrawlers/WebScrapers is initiated by the latter two, so they work properly even if they are in a sub-network or are behind a firewall and are not accessible from outside their network. The OffchainCore is the only component that needs to be reachable from anywhere.

Another component that needs access to the OffchainCore is the ClientAPI. This component is used by third-party party applications to consume the public services provided by OffchainCore, i.e., the article information and the associated trust scores assigned by the Crowd Wisdom and the Federated AI modules.

The communication between the components is made using the RESTful (REpresentational State Transfer) APIs, which are an efficient means of using HTTP requests for communicating with server applications. These APIs are exposed by the OffchainCore component and consist of the following main URL paths, with the corresponding HTTP method in parenthesis:

- `/api/auth/signin` (POST)—authentication using username and password; returns a JWT (JSON Web Token) with the user roles,
- `/api/auth/signup` (POST)—creates a new user with the specified role,
- `/api/crawler/sites` (GET)—returns a list of sites (with the page type classifier) assigned to a specific crawler user,
- `/api/crawler/pages` (POST)—receives a list of page URLs representing articles,
- `/api/scraper/extractor-templates` (GET)—returns the site list with the extractor templates, which are used to extract article information,
- `/api/scraper/article-urls` (GET)—returns a paginated list of the article URLs assigned to the authenticated scraper user,
- `/api/scraper/articles` (POST)—receives a batch of article information with the corresponding URLs,
- `/api/public/articles` (GET)—returns a paginated and filtered list of articles based on the query params: pageNumber, pageSize, publication start date and end date, publisher, title, author, and order by,
- `/api/public/articles/article-id` (GET)—returns the article information for the specified article id—it includes the option to return the article metadata, the featured image, the contents with the stripped HTML tags and the full contents of the article.

The aforementioned API routes are accessible based on the role of the authenticated user.

The last communication scenario in the news-retrieval architecture is the communication with the blockchain network. This is made using a smart contract, which adds the computed hash for each article content stripped of HTML tags to the blockchain. The hashes are only computed for articles that have been validated by enough scrapers. The data are added to the blockchain network in batches to increase efficiency, i.e., send the hashes in fixed batches or send all the hashes that are available at that point in time if enough time has passed since the previous update. Storing only the content hashes on the blockchain network acts as proof of data and ensures that the extracted article information has not been tampered with. Therefore, storing the entire article's contents on the blockchain is not necessary and not even feasible due to the involved costs (i.e., the gas fee).

3.6. OffchainCore Component

The OffchainCore component represents the brains of the news-retrieval system. Its main components are showcased in Figure 5. It is comprised of various modules:

- `common`—contains useful methods for handling JSON strings, errors, and other functions,
- `config`—handles the application initialization,
- `security`—contains the model, configuration, and logic to ensure secure access to the OffchainCore,
- `auth`—contains the authentication and authorization logic,
- `persistence`—contains the data models, the services, and the repositories for ensuring data persistence,
- `api`—exposes three REST APIs, i.e., CrawlerAPI, ScraperAPI, and ClientAPI,
- `service`—contains various services for storing the article data in the file system, storing the article hash in the blockchain network, and other helper services for the API module.

There are three entry points in the OffchainCore component, which are handled by the three controllers that process incoming requests: one controller for communication with the WebCrawler, one for the WebScraper, and one for the ClientAPI. Each time a WebCrawler sends a batch of URLs identified as containing articles, the CrawlerController receives that

information and calls the Site/Page Persistence module, which stores the information in the Offchain database. In this situation, a service that randomly allocates URLs to scrapers is triggered. The allocation information is stored in the database and is received upon request from the corresponding WebScraper. That request is handled by the ScrapperController module. When the WebScraper sends a batch of article information, that information is processed by the ScraperController, which calls on the ArticlePersistence module. As well as storing the article information in the Offchain DB and the Article Store, this module also updates the content hash count, which determines if an article information is considered to be valid. The Blockchain Article Store service periodically checks for newly determined valid articles by interrogating the Article Persistence module and storing the article hash on the blockchain network.

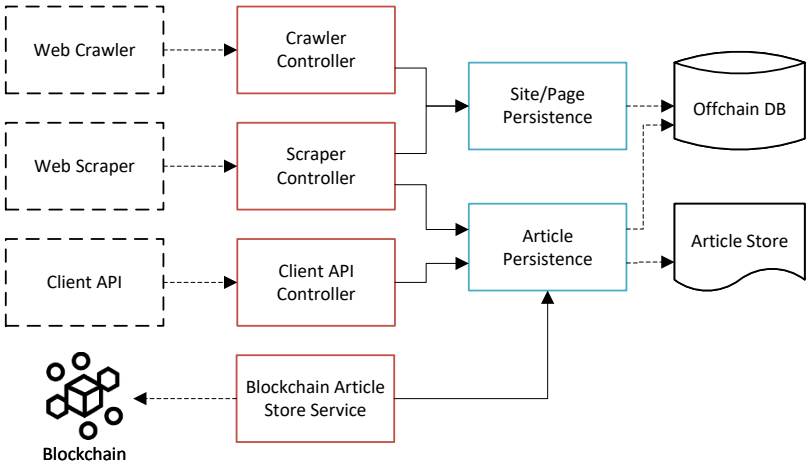


Figure 5. OffchainCore component main architecture, represented by the colored blocks. The red blocks represent services that run when a specific web service is called (for the three controllers) or run periodically (for the Blockchain Article Store service). The blue blocks represent the Data Persistence modules, and the arrows represent the module/component dependency direction.

3.7. WebCrawler Component

The WebCrawler component, whose main architectural blocks are presented in Figure 6, is executed by each crawler actor in the news-retrieval system. Three main services run periodically: the Site Retrieval Service and the Page Sender Service both use the OffchainCore API Handler module to communicate with the OffchainCore, while the Crawler Service manages the extraction of URLs from web pages and identifies which pages contain article information. All three services use the Data Persistence module to store and retrieve data from the crawler's database.

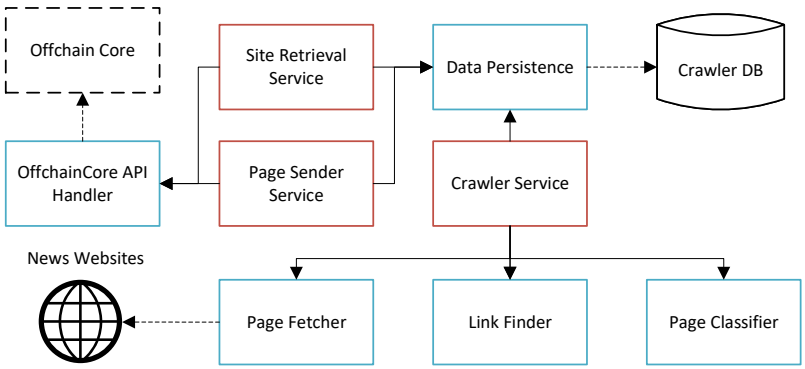


Figure 6. WebCrawler component main architecture, represented by the colored blocks. The red blocks represent services that run periodically. The blue blocks represent modules employed by the services, and the arrows represent the module/component dependency direction.

Periodically, the WebCrawler obtains the latest version of the site list to be crawled, including the page classifier for each site. A Crawler Service instance is created for each site that needs to be processed, and it is assigned its own thread; all the created instance threads run in parallel. Processing each site page is performed as follows:

1. Obtain the next unprocessed URL for that site from the database.
2. Obtain the page contents corresponding to that URL from that website.
3. Extract the URLs belonging to the crawled site by parsing the page contents.
4. Store each extracted URL in the crawler's database.
5. Use the page classifier to determine if the page contains an article and update the crawl status of the web page accordingly in the database.
6. Check if there still are unprocessed URLs. If there are, go to step 1. If not, then go to step 7.
7. Wait for a specified period of time (e.g., half an hour).
8. Reset the crawl status of the main page URL to unprocessed. Optionally, reset the crawl status of all page URLs if we want to support extracting news articles that update periodically.
9. Go to step 1.

The Page Sender Service periodically checks the database for processed URLs that have been identified as representing articles and that have not been sent to the OffchainCore. The URL list is sent in batches so as not to overload the OffchainCore with many requests.

3.8. WebScraper Component

The actual article information is extracted by the WebScraper component (Figure 7). There are a total of four services that run periodically, and all of them use the Data Persistence layer to access the database and/or the article store. Three of them involve communication with the OffchainCore, which is performed similarly to the WebCrawler component, using another OffchainCore API Handler module. The Extractor Templates Retrieval Service runs rarely, and it obtains updated versions of the extraction templates required to extract the article information. The Article URLs Retrieval Service obtains a batch of article URLs to be processed and is called after each batch is processed to obtain the next one. The Article Info Sender Service monitors the database and sends the article information from the Scraper DB and Scraper Store, also in batches. Finally, the core of the WebScraper component is represented by the Scraper Manager Service. This service constantly monitors the database for new article URLs and manages multiple Site Scrapers. There must be only one thread per site that handles the page contents retrieval in order not to have the instance's IP banned by the news website. A batch can contain URLs from different sites and multiple URLs from the same site. Therefore, a variable number of Site Scraper threads are used.

The information stored in the Scraper Store consists of a directory for each website in which there are two files saved for each article. The first is a JSON file containing an object with all the extracted information, i.e., URL, title, contents, contentsTextOnly, contentsTextOnlyHash, featuredImageUrl, featuredImageHash, publishDate, author, extractedDate. The other file is the featured image of the article in the format (e.g., png, jpg, gif, webp) downloaded using the Web Resource Retriever Service.

The steps taken by each Site Scraper are as follows:

1. Obtain the next unprocessed article URL.
2. Obtain the page contents corresponding to that URL from that website.
3. Remove the HTML elements specified in the extractor template and remove the comments.
4. Extract the article information specified in the extractor template.
5. Retrieve the featured image.
6. Compute the hashes for the extracted text-only content and the featured image.
7. Save the article information in the filesystem and update the scrap status of the corresponding article URL in the database.

8. Check if there still are unprocessed URLs. If there are, go to step 1. If not, then terminate (the scraper manager will update the list from the next batch).

Some other modules and functionalities make the OffchainCore, WebCrawler, and WebScraper function properly, but for the sake of brevity, these were omitted from this paper. Only the main components and main workflows are described.

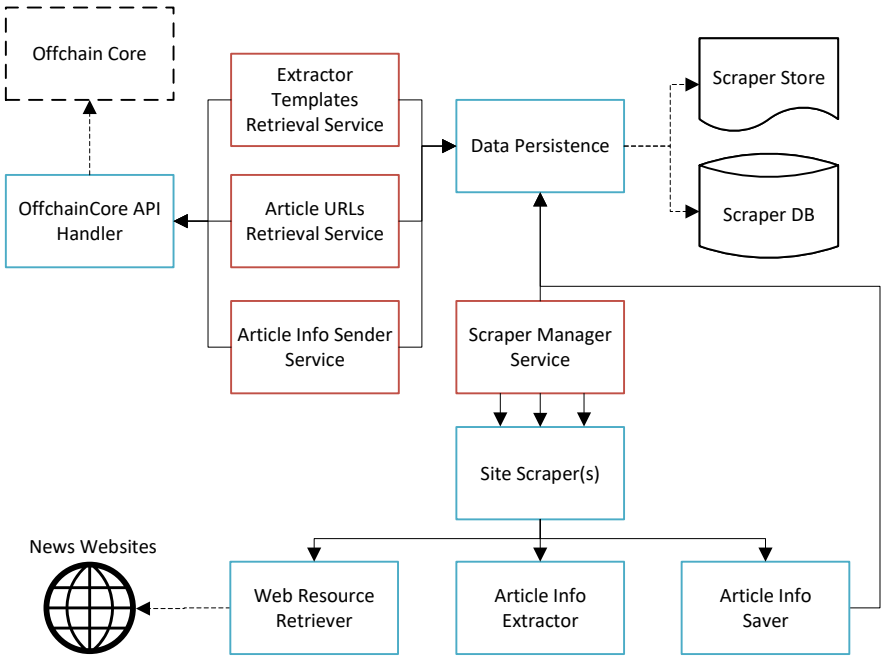


Figure 7. WebScraper component main architecture, represented by the colored blocks. The red blocks represent services that run periodically. The blue blocks represent modules employed by the services, and the arrows represent the module/component dependency direction.

4. Cloud Deployment

In this section, we identify the requirements of each system component and propose solutions for cloud deployment to four major cloud solutions: OpenStack, Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure. OpenStack is an open-source cloud platform that is deployed on premises, while the other three are popular commercial cloud solutions. Deploying the components of the proposed system on a cloud solution provides the inherent advantages that the cloud brings, especially scalability and availability.

The proposed news-retrieval system must run in a decentralized way in which different actors run crawler and scraper instances. There is still a centralized component, i.e., the OffchainCore, in charge of coordinating the crawlers and scrapers. In terms of cloud deployment, we propose a solution for the OffchainCore and for the individual crawler and scraper instances for easy deployment and execution.

The minimum requirements for the OffchainCore component are a compute instance for the web server with the business logic, a database for handling all the data, and a storage solution for storing the featured image and the article JSON files. In order to make it more scalable, a load balancer can be used to manage multiple OffchainCore web server instances. The web server can be deployed on a virtual machine by installing first the necessary dependencies. On OpenStack, the Nova service can be used for managing virtual servers and Neutron for networking. On AWS, Elastic Compute Cloud (EC2) can be used. On GCP, there is Google Compute Engine, and on Microsoft Azure, we have Virtual Machine. A disadvantage of using OpenStack is that it does not provide an out-of-the-box solution for Platform-as-a-Service (PaaS), and the existing third-party solutions lack the proper support. A PaaS represents a service in which the cloud provider provisions everything that the developers need to manage applications. It is basically a compute

instance (or more) with all the necessary application dependencies installed and with a seamless means to deploy the application in that environment. The web server can easily be deployed on AWS Lambda, Google App Engine, or Microsoft App Service, which also handles scaling automatically.

For database storage, each cloud provider takes advantage of a database cluster, and there are several solutions: OpenStack's Trove, AWS RDS, Google SQL, or Azure Database for MySQL. The OffchainCore also requires file storage, for which we can use OpenStack's Swift service, AWS Simple Storage Service (S3), Google Cloud Storage, or Microsoft Storage. The idea is to use a storage service to have the same storage for all the instances of OffchainCore so the stored data are consistent.

Regarding the crawler and scraper actors, the installation of the crawler/scraper application must be done without much interaction from the user. Both applications require a Java runtime environment. The crawler also needs a MySQL database server, whereas the scraper requires access to the file system for storing the SQLite database as a file and for storing the article information JSON files and the article featured images. The simplest method for cloud deployment is to have an instance for the crawler/scraper with everything installed. This is neither efficient nor scalable. Another issue is regarding configuration because each crawler/scraper must be configured with the proper credentials to communicate with the OffchainCore.

One solution is to use containerization and create a container with the application and one with the database server. There are two approaches here: run everything on a single machine or run it in a cluster of container daemons. There are cloud services that provide the means to work with containers: OpenStack's Zun and Magnum, AWS Elastic Container Service (ECS), Google Kubernetes Engine, and Azure Kubernetes Service. Using orchestration, the deployment of the crawler/scraper can be done quickly and with ease as long as the container solution is properly configured. The downside of this approach is the extra layer of containerization, which adds overheads.

Another solution is to use dedicated cloud services to deploy the crawler/scraper. Figure 8 shows an application architecture for the WebCrawler and WebScraper components for deployment on the OpenStack cloud solution. A similar approach can be achieved for the other aforementioned cloud solutions by employing the appropriate services.

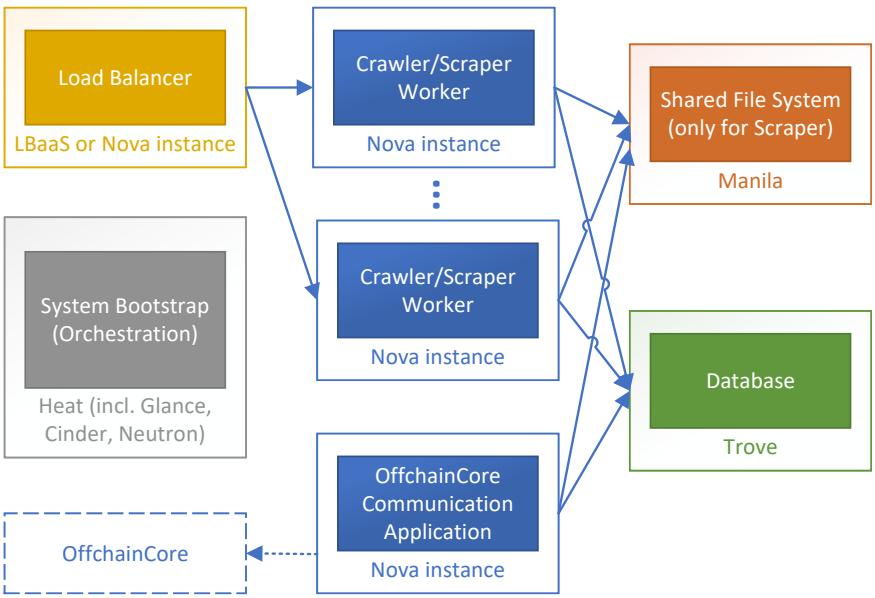


Figure 8. Application architecture for the WebCrawler and WebScraper components for deployment on the OpenStack Cloud.

The proposed architecture is highly scalable and allows multiple crawler/scraper worker instances to run in parallel and process the same sites. The trick is to allocate each instance a different floating IP (i.e., which allows the instance to have a real IP) so that a news website does not ban the application. Otherwise, many requests would have been sent from the same IP, and there would be a greater chance for that IP to be banned from crawling a specific website. There is a change that has to be made to the proposed news-retrieval architecture from Section 3. The logic that allows the communication between the crawler and the OffchainCore must be extracted from the system into another application so the two parts (the extraction logic and the OffchainCore communication logic) can scale independently. This is easily achievable thanks to the way the crawler was designed. The communication between the two parts is performed indirectly through the database server (i.e., the Trove service for the OpenStack deployment). The scraper instances require that the article information be saved on the file system. A good approach is to use the Manila service for a shared file system. This way, all the scrapers write in the same place, and the OffchainCore Communication Application can provide the OffchainCore with necessary article information and featured image resources. There are no concurrency issues when writing the data in the shared file system because each scraper creates, under its own unique ID path, two different files for each extracted article.

In terms of actual deployment, the Heat orchestration service from OpenStack can be employed. This service communicates with the Glance service to specify the details of the image used to create the instances, Cinder for the virtual disk drives, Neutron for networking, and the other aforementioned services required to run the crawler/scraper solutions. Using Heat Orchestration Template (HOT), we can configure the compute instances by specifying the image used to create the instance, the flavor (i.e., number of vCPUs, disk size, and RAM size of the instance), the instance IP/host, and the script that runs after the instance is created. The script includes the installation of all the application dependencies, obtaining the latest version from the application repository, and launching the application. The HOT YAML file also allows the configuration of the load balancing service (LBaaS) by defining the worker pools and the floating IPs that are used.

Having different crawler/scraper actors as part of the system is crucial to ensuring trust in the proposed solution. This is why it is important to have a solution for easy deployment and execution of the applications. Deploying on the cloud is not an easy feat, but it allows a significant increase in performance and efficiency.

5. Community-Based Truly Decentralized Architecture

A decentralized system means that it is controlled by multiple authorities rather than a single one. In the current state of the proposed solution, decentralization is achieved both at crawler and scraper levels, but the information is aggregated, and the majority rule is obtained at the OffchainCore component level.

One easy-to-implement improvement is to have the crawler instances put the hashes of the URLs identified as articles on the blockchain network through a smart contract. Similarly, the scraper instances can put themselves the hashes of the extracted article information and the featured image. This way, the hash is put on the blockchain network by each crawler/scraper rather than only by OffchainCore so that anyone can verify the veracity of the stored information. Now, when OffchainCore determines the majority rule regarding an article, it puts the hash on the blockchain network. Anyone can verify the fact that the uploaded hash adheres to the majority rule by looking at the hashes uploaded by each actor.

However, there are two main downsides. First, each written transaction on the blockchain costs gas, which represents the cost required to perform a transaction on the blockchain network. This can lead to unwanted costs for the crawler and scraper actors. An improvement can be made so that a transaction consists of multiple hashes. If the blockchain in question is a community blockchain solution, e.g., a private Ethereum network, then a custom gas price standard can be defined. Another downside is that

the actors must have a valid crypto wallet for signing the transactions. As it stands, the OffchainCore is responsible for signing the transactions and paying the gas fee. The actors can be incentivized to be part of the network by obtaining certain benefits. For example, to have access to analytics regarding the information extracted by all actors or even to receive part of the revenue obtained by monetizing the proposed solution. In this way, the whole community of actors contributes to having a trusted news aggregator and the associated benefits.

Now, the OffchainCore component is a single point of failure and requires trust from the actors in the system. An alternative to the aforementioned improvement involving the actors uploading hash values on the blockchain network is to somehow decentralize the OffchainCore and its associated database. Custom majority-rule algorithms can be employed to ensure that multiple nodes handle the data and processing in a trustful manner, but this is a huge task, and it is not a feasible solution. A better approach is the use of a decentralized storage system, such as the InterPlanetary File System (IPFS), in order to have the OffchainCore as a decentralized web platform. The authors from [36] discuss the performance of having a decentralized web on IPFS.

A decentralized storage system consists of a peer-to-peer network of nodes holding parts of the overall data, creating a resilient file storage system. Such a system can be implemented on top of other decentralized architectures, such as a blockchain-based application or a dedicated peer-to-peer-based network. To analyze a decentralized storage solution, we have to look at the following aspects: persistence mechanism and incentivization schemes, data retention enforcement policy, the level of decentralization, and the majority-rule mechanisms. IPFS does not have a built-in incentive scheme; however, contract-based incentive solutions can be used to ensure longer-term persistence.

Another possible solution is using decentralized oracles [37]. Blockchain oracles are services that allow smart contracts to have access to information from the outside world, i.e., from external data sources. An example of a decentralized oracle-based mechanism for verification that can be used by our proposed solution is presented in [38]. These hybrid smart contracts can use the information provided by the crawler and scraper actors to determine the majority decision regarding specific URLs identified as containing articles and specific article information, respectively. Using oracles, each crawler and scraper can call upon a smart contract to send a hash string associated with processed information. The smart contract logic, based on the information already stored on the blockchain, can determine if a majority exists regarding that information and store the final decision on the blockchain network. The identity of each involved actor is also stored on the blockchain because each one must authenticate and sign the transaction. This approach eliminates OffchainCore's intermediary role of adding the hashes to the blockchain network and, consequently, improves the decentralization of the proposed system.

There are ways of having a truly trusted decentralized architecture using solutions like IPFS, which is an integral part of the Web 3.0 concept [39], and blockchain oracles, which can interact with Offchain resources. The aforementioned possible methods are the basis for extending the research presented in this paper and exploring new possibilities for ensuring trust, not just regarding news publications. As long as the Internet community is willing to contribute and participate as nodes in this decentralized environment, further developments, and enhancements can be made more easily.

6. Use-Case Scenario and Results

The proposed news-retrieval system was tested on seven news websites from Romania. The information published on those websites was written in Romanian, a fact that has no negative influence on running the presented system. Even more so, the system supports any language as the data are saved in UTF-8 character encoding. The details regarding the chosen use-case scenario, the testing environment, and the results are as follows.

6.1. Scenario Description

The seven news websites chosen are Adevarul, AgerPres, DCNews, Digi24, G4Media, Hotnews, and Stiripesurse. These are among the top news publishers in Romania and are considered to be reliable sources of information.

For each website, an extraction template was created manually by analyzing the HTML contents of the web pages containing articles. This can be done easily using the Inspect Element feature of the web browser and copying the selector associated with the selected element containing the extractable value. Listing 1 shows an example of the website input data, including the extraction template, for the Digi24 news website.

Listing 1. Example of an extraction template JSON string.

```
{
  "name": "Digi24",
  "urlBase": "https://www.digi24.ro",
  "logoUrl":"https://www.digi24.ro/static/theme-repo/bin/images/digi24-logo
      .png",
  "pageTypeClassifier": {
    "containsList": [
      "<main id=\"article-content\""
    ],
    "containsNotList": [
                "<a href=\"/video\" title=\"Video\" class=\"breadcrumbs-
                    item-link\"> Video </a> "
    ]
  },
  "extractorTemplate": {
    "removeElements": ["script, style, div.ad-wrapper"],
    "title": ["#article-content > article.article > div.container > div.
        flex.flex-center > div > h1", "text"],
    "contents": ["#article-content > article.article > div.container > div.
        flex.flex-end.flex-center-md.flex-stretch > div.col-8.col-md-9.col-
        sm-12 > div > div > div.entry.data-app-meta.data-app-meta-article",
         "html"],
    "featuredImage": ["#article-content > article.article > div.container >
         div.flex > div  > figure.article-thumb > img", "attr:src"],
    "publishDate": ["#article-content > article.article > div.container >
        div.flex.flex-center > div  > div.flex.flex-middle > div > div.
        author > div.author-meta > span:last-child > time", "attr:datetime"
        ],
    "author": ["#article-content > article.article > div.container > div.
        flex.flex-end.flex-center-md.flex-stretch > div.col-8.col-md-9.col-
        sm-12 > div > div > div.entry.data-app-meta.data-app-meta-article a
        [href*=/autor/]", "attr:href", true]
  }
}
```

The OffchainCore process is initialized with a JSON file similar to the one from Listing 1 for each website. These files are used to initially populate the database with the site list and to allocate sites to web crawlers.

6.2. Testing Environment

For testing purposes, three OpenStack Nova instances were created, one for each of the OffchainCore, WebCrawler, and WebScraper, with the `m1.medium` flavor, which means that each instance has 2 vCPUs (virtual CPUs), 40 GB disk space and 4096 MB of RAM. Also, the three instances had allocated a floating IP so they could be publicly accessible for logging, debugging, and external access. Each Nova instance has installed the Rocky Linux operating system and the Java Virtual Machine for running the applications. The instances that run the OffchainCore and WebCrawler also had a MariaDB database server installed, while the WebScraper used SQLite. The decision to use different database solutions was made for performance reasons. The WebScraper does not need to store much information,

and the database updates are few. Therefore, there was no need for a separate database server; instead, a simple in-process database was used.

In terms of the configuration of the crawler and scraper, the following parameters and corresponding values were used:

- minimum waiting time between requests made to the same site (in milliseconds): 1000,
- maximum waiting time between requests made to the same site (in milliseconds): 2500,
- re-crawl time interval (in HH:mm:ss format for time): 00:30:00,
- re-crawl time interval delta (in ISO 8601 format for the duration [40]); re-crawl will occur at the re-crawl time interval +/− a random value between 0 and re-crawl time interval delta).

The first two parameters are designed to generate requests to the same site with a random delay of between 1 and 2.5 s. This way, the target server is not overloaded with requests, and it is less likely that the server will have a problem with the crawler/scraper. The last two parameters refer to the situation in which a site is fully crawled. In this case, the crawler runs after a "re-crawl time interval" modified with a "delta" has passed and starts by re-crawling the main page of the site, which contains the latest news.

6.3. Results

All three processes, i.e., OffchainCore, WebCrawler, and WebScraper, were executed for a fixed period of time on the seven news websites. The statistics regarding the processing are presented in Table 1.

Table 1. Statistics regarding crawling and scraping from seven news websites.

		First Experiment [a]					Second Experiment [b]		
Site Name	**Site URL**	**Total Number of Extracted URLs**	**Number of Waiting to Be Processed Pages**	**Number of Article Pages**	**Number of Irrelevant Pages**	**Number of Invalid Pages**	**Avg. Time to Retrieve Contents (ms)**	**Avg. Time to Process a URL (ms)**	**Number of Processed URLs**
Adevarul	https://adevarul.ro (accessed on 4 August 2023)	35,041	30,526	3769	617	129	873	76,334	454
AgerPres	https://www.agerpres.ro (accessed on 4 August 2023)	14,145	840	2993	3696	6616	217	19,771	1673
DCNews	https://www.dcnews.ro (accessed on 4 August 2023)	26,156	22,730	2644	633	149	254	82,284	425
Digi24	https://www.digi24.ro (accessed on 4 August 2023)	60,977	56,043	3130	1803	1	334	193,100	181
G4Media	https://www.g4media.ro (accessed on 4 August 2023)	18,733	16,566	2076	81	10	364	88,129	392
Hotnews	https://www.hotnews.ro (accessed on 4 August 2023)	11,537	4749	6216	220	352	393	47,873	722
Stiripesurse	https://www.stiripesurse.ro (accessed on 4 August 2023)	45,661	40,625	2254	1720	1062	887	89,788	389
Total		212,250	172,079	23,082	8770	8319	475 [c]	85,325 [c]	4236

[a] over a longer period of time in which the crawler processed multiple times a smaller number of sites at the same time; [b] over a 4 h time period in which seven site crawler threads ran on the same instance at the same time (one crawler thread/site); [c] average values.

Two experiments were performed. The first one was over a longer period of time, and the crawler ran on subsets of the seven websites at the same time for performance reasons. Before the first test started, the main page URL of each of the seven sites was marked for a re-crawl. The second one was a shorter test over a four-hour period in which seven site crawler threads ran on the same instance at the same time (one crawler thread per site).

For the first experiment, the goal is to observe the percentage of useful web pages, i.e., the web pages that contain an article, compared to the total number of the website's pages. Table 1 is a good indicator of how the crawling progresses over a period of time. If we consider only the processed pages (the sum of the number of article pages, irrelevant pages, and invalid pages), the proportion that represents articles varies between 44% and 96%, with an exception. That exception is represented by AgerPres with a percentage of only 22% due to a large number of considered invalid pages of almost 50%. Looking at those invalid pages, these fall into three categories: URLs that represent, in fact, a phone number and contain the text `+tel`, URLs with fragments (e.g., `#mm-2`), and pages that were not available when the crawling ran (either because the web server was down or because the web crawler was temporarily blocked). Excluding those invalid pages, the article page/processed page percentage among the seven websites is between 45% and 97%. AgerPres also has a large number of irrelevant pages because they provide separate URLs to HTML pages that allow the user to view the photographs that appear on the page. Examples of such URLs are `/foto/watermark/11689895` and `/fotografia\_zilei/146/page/9`. The irrelevant pages of the Digi24 website consisted mostly of paginated views of multiple article snippets per page and URLs for each tag associated with the news articles. The same situation is for Stiripesurse, which also had many pages marked as irrelevant.

Regarding the second experiment, the main goal is to determine how many URLs can be processed by a single crawler, which processes all seven sites in parallel over four hours. The results from Table 1 have to be correlated with the data presented in Figure 9 to have a broader picture of the obtained time measurements.

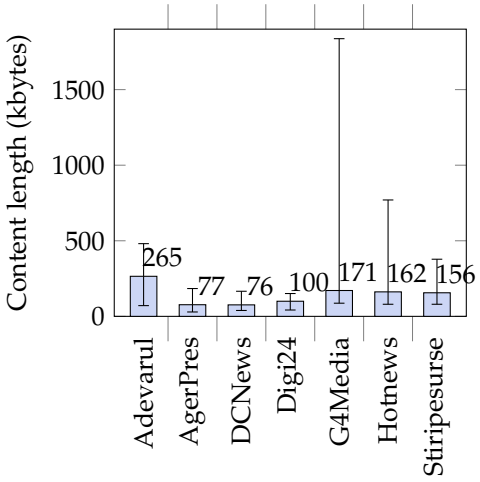


Figure 9. Average content length per page in kbytes. The minimum and maximum lengths are also highlighted.

Regarding the average time it takes to retrieve the web page contents, that time was between 217 ms for AgerPress and 887 ms for Stiripesurse. The time is directly related to the web page content length (the number of kilobytes received from the server) and the latency of the connection to the news web server. Looking at Figure 9, we see that G4Media's maximum page length is around 1840 KB, but this is the case for only three pages from that website out of the 392 that were processed. This is because one of those pages is the main page, which has multiple article snippets. Another consists of a list of article snippets that belong to a specific author, and the last one is a page with many article snippets.

During the four hours, the number of processed URLs varied from 181 for Digi24 to 1673 for AgerPres. The explanation for this discrepancy is directly related to the average time of processing a page. The fact that a page is marked as irrelevant still implies that the URLs are extracted and stored in the database. The average time of processing a URL varies between 19,771 ms and 193,100 ms. This is largely because of the extracted number

of links that must be checked and written in the database. Figure 10 shows the average number of extracted links per page, while Figure 11 shows an average number of new links per page, i.e., links that have not appeared on the previously crawled pages. For G4Media, the maximum number of extracted links and new links was significantly higher due to the three web pages that contain many article snippets. Looking at the average values from all seven sites, a maximum of around 10% of links from a web page are new URLs that are not stored in the database.

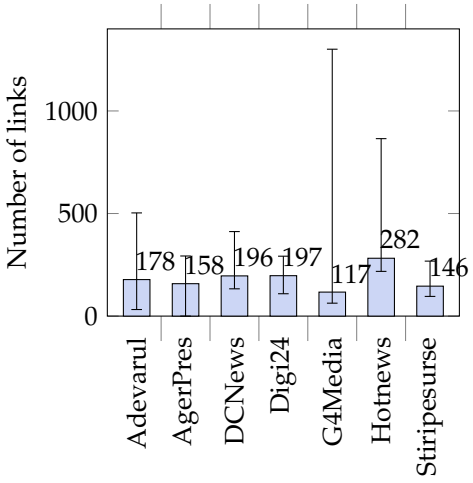


Figure 10. Average number of extracted links per page. The minimum and maximum number of links are also highlighted.

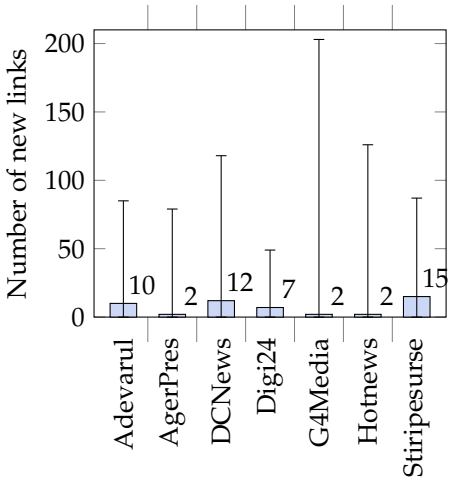


Figure 11. Average number of new links per page (links that have not appeared on the previously crawled pages). The minimum and maximum number of links are also highlighted.

In terms of the web scraper, which extracts the article information, accessing the page, extracting, and storing the data takes less, mainly because the interaction with the database comes down to obtaining the URL and updating twice the processing status (i.e., processing and processed). On the other hand, there is the overhead of writing the article information JSON file and the featured image in the local file system. The average time to entirely process a URL on the tested system was 722 ms. Each URL was processed one at a time, whereas for the second crawling experiment, there were seven threads in parallel. Even if we consider the scraping time seven times slower, the average URL processing time when crawling was around 16 times slower than scraping on the tested instances. The reasoning behind the approach and the results are discussed in the next section of this paper.

7. Discussion

Analyzing the results of the crawling and scraping processes, i.e., the obtained execution times for the various processing, the web page content length, the format of the URLs, and the execution environment, optimizations can be made to speed up the extraction.

The fact that, for some websites, there are many irrelevant and/or invalid page URLs adds more processing time to the crawling process. One improvement that can be made is to add an extra filter based on the URL string so that certain URLs are ignored. The filter must be customized for each site and can be established by analyzing the URLs of an article and non-article pages. The filter efficiency depends on the website's approach to establishing the URLs. Elements like a taxonomy or some other URL patterns can help in determining the filter.

Tests have shown that checking if a URL is in the database and writing the URL if it is not there takes a significant amount of time compared to the time it takes to retrieve the page from the web server and to parse it. Using certain data structures for fast look-up, e.g., PatriciaTrie [41], and adding a caching layer, e.g., using Redis [42], can significantly improve the crawler's performance.

Looking at the individual crawling and scraping methods, compared to existing solutions, those methods produce similar results. The proposed crawler and scraper were designed to be lightweight, and they can be substituted by an existing solution, provided that adaptations are made in order to adhere to the communication protocol with the OffchainCore.

The main focus of this paper was not the individual crawler and scraper but rather the method of achieving decentralization using the URL allocation algorithm and the majority-rule method. Section 3.1 presents a theoretical analysis of the efficiency of the proposed solution for ensuring trust in the decentralized news-retrieval solution. Even though the same pages are processed by a group of crawlers/scrapers, this allows the system to account for mal-intent actors that want to inject fake information and, using majority rule, to detect malicious attempts and to prevent this from happening.

Storing the news article hash on the blockchain network allows actors to verify the accuracy of the majority-rule decision and, implicitly, the veracity of the extracted information. One can say for certain that the article contents stored in the system are the same as the article on the news website from which it has been extracted. As for the veracity of the actual contents of the article, i.e., determining if the presented information is real and true, the method of determining this is based on crowd wisdom and AI methods, which are employed in the FiDisD system, although the exact details are beyond the scope of this paper.

The integration of blockchain technology in the proposed solution does not impact the end-user or the crawler and scraper actors. The OffchainCore component is the one interacting directly with the blockchain network. In terms of performance, the costliest operation is writing on the blockchain. Fortunately, only the hash string on the article contents decided by the majority rule is stored on the blockchain network. This significantly reduces the cost of using blockchain. Also, any blockchain solution can be employed by OffchainCore because the information stored on the blockchain network is public, so anyone can verify that the article contents saved locally by our system are accurate and it is the same as the one extracted by the scrapers.

In terms of the actual news content, usually, an article is picked up by multiple news agencies, which rephrase the piece of news and publish it on their websites. The current proposed system considers each news article from different websites as a distinct piece of news, even if certain passages of text are common to multiple article contents. An extension of our proposed system is to develop an artificial intelligence-based method for determining similar news articles from different websites. Based on the crawl timestamp, the news origin can be determined. This allows for tracing and determining the websites that spread misinformation and disinformation.

8. Conclusions

This paper presents the details in terms of architecture, implementation, and the obtained results of a proposed news-retrieval system, which is used for assessing trust and fighting disinformation when it comes to online news articles. Based on the tests conducted on seven news websites, various optimizations are proposed for improving the system by minimizing the processing times. Another important contribution that this paper brings is an extensive discussion referring to the system deployment in a cloud solution, specifically on the open-source OpenStack platform. Also, various improvements are proposed for having a community-based truly decentralized architecture. The advantages and disadvantages of each solution are discussed, and the potential for further research is emphasized.

Author Contributions: Conceptualization, A.A.; methodology, A.A. and C.N.B.; software, A.A. and C.N.B.; validation, A.A. and C.N.B.; formal analysis, A.A. and C.N.B.; investigation, A.A. and C.N.B.; data curation, A.A.; writing—original draft preparation, A.A.; writing—review and editing, A.A. and C.N.B.; supervision, A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the project "Collaborative environment for developing OpenStack-based Cloud architectures with applications in RTI" SMIS 124998 from The European Regional Development Fund through the Competitiveness Operational Program 2014–2020, priority axis 1: Research, technological development and innovation (RTI)—the POC/398/1/1 program.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wu, Y.; Ngai, E.W.; Wu, P.; Wu, C. Fake news on the internet: A literature review, synthesis and directions for future research. *Internet Res.* **2022**, *32*, 1662–1699. [CrossRef]
2. Chen, C.-H.; Ma, Y.; Lai, Y.H.; Chang, W.-T.; Yang, S.-C. Analyzing Disinformation with the Active Propagation Strategy. In Proceedings of the 24th International Conference on Advanced Communication Technology (ICACT)—Artificial Intelligence Technologies toward Cybersecurity, Pyeongchang, Republic of Korea, 13–16 February 2022; pp. 262–266.
3. Dowse, A.; Bachmann, S.D. Information warfare: Methods to counter disinformation. *Def. Secur. Anal.* **2022**, *38*, 453–469. [CrossRef]
4. Schneider, E.J.; Boman, C.D. Using Message Strategies to Attenuate the Effects of Disinformation on Credibility. *Commun. Stud.* **2023**, *74*, 393–411. [CrossRef]
5. Bondielli, A.; Marcelloni, F. A survey on fake news and rumour detection techniques. *Inf. Sci.* **2019**, *497*, 38–55. [CrossRef]
6. Guttmann, A. Survey: Index of Respondents' Trust towards Media in European Union (EU 28) Countries in 2019. Available online: https://www.statista.com/statistics/454409/europe-media-trust-index/ (accessed on 7 August 2023).
7. Gorbunova, M.; Masek, P.; Komarov, M.; Ometov, A. Distributed Ledger Technology: State-of-the-Art and Current Challenges. *Comput. Sci. Inf. Syst.* **2022**, *19*, 65–85. [CrossRef]
8. Soltani, R.; Zaman, M.; Joshi, R.; Sampalli, S. Distributed Ledger Technologies and Their Applications: A Review. *Appl. Sci.* **2022**, *12*, 7898. [CrossRef]
9. Antal, C.; Cioara, T.; Anghel, I.; Antal, M.; Salomie, I. Distributed Ledger Technology Review and Decentralized Applications Development Guidelines. *Future Internet* **2021**, *13*, 62. [CrossRef]
10. Chauhan, A.; Malviya, O.P.; Verma, M.; Mor, T.S. Blockchain and Scalability. In Proceedings of the 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), Lisbon, Portugal, 16–20 July 2018; pp. 122–128. [CrossRef]
11. Mohanta, B.K.; Jena, D.; Panda, S.S.; Sobhanayak, S. Blockchain technology: A survey on applications and security privacy challenges. *Internet Things* **2019**, *8*, 100107. [CrossRef]
12. Yang, J.; Vega-Oliveros, D.; Seibt, T.; Rocha, A. Scalable Fact-checking with Human-in-the-Loop. In Proceedings of the 2021 IEEE International Workshop on Information Forensics and Security (WIFS), Montpellier, France, 7–10 December 2021; pp. 86–91. [CrossRef]
13. Ciampaglia, G.L.; Shiralkar, P.; Rocha, L.M.; Bollen, J.; Menczer, F.; Flammini, A. Computational Fact Checking from Knowledge Networks. *PLoS ONE* **2015**, *10*, e0141938. [CrossRef]
14. Classification: ROC Curve and AUC | Machine Learning. Available online: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc (accessed on 7 August 2023).

15. FiDisD—Fighting Disinformation Using Decentralized Actors Featuring AI and Blockchain Technologies. Available online: https://www.trublo.eu/fidisd/ (accessed on 4 July 2023).
16. Buțincu, C.N.; Alexandrescu, A. Blockchain-Based Platform to Fight Disinformation Using Crowd Wisdom and Artificial Intelligence. *Appl. Sci.* **2023**, *13*, 6088. [CrossRef]
17. OpenStack. Available online: https://www.openstack.org/ (accessed on 15 July 2023).
18. Hamborg, F.; Meuschke, N.; Breitinger, C.; Gipp, B. News-please: A Generic News Crawler and Extractor. In *Everything Changes, Everything Stays the Same: Understanding Information Spaces, Proceedings of the 15th International Symposium of Information Science (ISI 2017), Berlin, Germany, 13–15 March 2017*; Gäde, M., Ed.; Hülsbusch: Glückstadt, Germany, 2017; pp. 218–223.
19. Ou-Yang, L. Newspaper3k: Article Scraping & Curation. Available online: https://newspaper.readthedocs.io/en/latest/ (accessed on 4 July 2023).
20. Le Huy Hien, N.; Tien, T.Q.; Van Hieu, N. Web Crawler: Design and Implementation for Extracting Article-like Contents. *Cybern. Phys.* **2020**, *9*, 144–151. [CrossRef]
21. Alexandrescu, A. A distributed framework for information retrieval, processing and presentation of data. In Proceedings of the 2018 22nd International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, Romania, 10–12 October 2018; pp. 267–272.
22. Alexandrescu, A. Optimization and security in information retrieval, extraction, processing, and presentation on a cloud platform. *Information* **2019**, *10*, 200. [CrossRef]
23. Dong, Y.; Li, Q.; Yan, Z.; Ding, Y. A generic Web news extraction approach. In Proceedings of the 2008 International Conference on Information and Automation, Changsha, China, 20–23 June 2008; pp. 179–183. [CrossRef]
24. Barbaresi, A. Trafilatura: A Web Scraping Library and Command-Line Tool for Text Discovery and Extraction. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations, Online, 1–6 August 2021; pp. 122–131. [CrossRef]
25. Qudus Khan, F.; Tsaramirsis, G.; Ullah, N.; Nazmudeen, M.; Jan, S.; Ahmad, A. Smart algorithmic based web crawling and scraping with template autoupdate capabilities. *Concurr. Comput. Pract. Exp.* **2021**, *33*, e6042. [CrossRef]
26. Gupta, G.; Chhabra, I. Optimized template detection and extraction algorithm for web scraping of dynamic web pages. *Glob. J. Pure Appl. Math.* **2017**, *13*, 719–732.
27. Singh, A.; Srivatsa, M.; Liu, L.; Miller, T. Apoidea: A Decentralized Peer-to-Peer Architecture for Crawling the World Wide Web. In *Distributed Multimedia Information Retrieval*; Callan, J., Crestani, F., Sanderson, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2004; pp. 126–142.
28. Wang, S.; Huang, C.; Li, J.; Yuan, Y.; Wang, F.Y. Decentralized construction of knowledge graphs for deep recommender systems based on blockchain-powered smart contracts. *IEEE Access* **2019**, *7*, 136951–136961. [CrossRef]
29. Ye, H.; Lu, Y.; Qiu, G. Tracing Method of False News Based on Python Web Crawler Technology. In Proceedings of the International Conference on Advanced Hybrid Information Processing, Changsha, China, 29–30 September 2022; pp. 489–502.
30. Kim, Y.Y.; Kim, Y.K.; Kim, D.S.; Kim, M.H. Implementation of hybrid P2P networking distributed web crawler using AWS for smart work news big data. *Peer-Netw. Appl.* **2020**, *13*, 659–670. [CrossRef]
31. Prismana, I.G.L.P.E. Distributed News Crawler Using Fog Cloud Approach. In Proceedings of the International Joint Conference on Science and Engineering 2022 (IJCSE 2022), Surabaya, Indonesia, 10–11 September 2022; pp. 251–260.
32. Ren, X.; Wang, H.; Dai, D. A summary of research on web data acquisition methods based on distributed crawler. In Proceedings of the 2020 IEEE 6th International Conference on Computer and Communications (ICCC), Chengdu, China, 11–14 December 2020; pp. 1682–1688.
33. Kaur, S.; Geetha, G. SIMHAR-smart distributed web crawler for the hidden web using SIM+ hash and redis server. *IEEE Access* **2020**, *8*, 117582–117592. [CrossRef]
34. ElAraby, M.; Moftah, H.M.; Abuelenin, S.M.; Rashad, M. Elastic web crawler service-oriented architecture over cloud computing. *Arab. J. Sci. Eng.* **2018**, *43*, 8111–8126. [CrossRef]
35. Gunawan, D.; Amalia, A.; Najwan, A. Improving data collection on article clustering by using distributed focused crawler. *Data Sci. J. Comput. Appl. Inform.* **2017**, *1*, 1–12. [CrossRef]
36. Trautwein, D.; Raman, A.; Tyson, G.; Castro, I.; Scott, W.; Schubotz, M.; Gipp, B.; Psaras, Y. Design and evaluation of IPFS: A storage layer for the decentralized web. In Proceedings of the ACM SIGCOMM 2022 Conference, Amsterdam, The Netherlands, 22–26 August 2022; pp. 739–752.
37. Breidenbach, L.; Cachin, C.; Chan, B.; Coventry, A.; Ellis, S.; Juels, A.; Koushanfar, F.; Miller, A.; Magauran, B.; Moroz, D.; et al. Chainlink 2.0: Next steps in the evolution of decentralized oracle networks. *Chain. Labs* **2021**, *1*, 1–136.
38. Ma, L.; Kaneko, K.; Sharma, S.; Sakurai, K. Reliable decentralized oracle with mechanisms for verification and disputation. In Proceedings of the 2019 Seventh International Symposium on Computing and Networking Workshops (CANDARW), Nagasaki, Japan, 26–29 November 2019; pp. 346–352.
39. Alabdulwahhab, F.A. Web 3.0: The decentralized web blockchain networks and protocol innovation. In Proceedings of the 2018 1st International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia, 4–6 April 2018; pp. 1–4.

40. W3C Date and Time Formats. Available online: https://www.w3.org/TR/NOTE-datetime (accessed on 15 July 2023).
41. Chatterjee, T.; Ruj, S.; Das Bit, S. Efficient Data Storage and Name Look-Up in Named Data Networking Using Connected Dominating Set and Patricia Trie. *Autom. Control Comput. Sci.* **2021**, *55*, 319–333. [CrossRef]
42. Su, Q.; Gao, X.; Zhang, X.; Wang, Z. A novel cache strategy leveraging Redis with filters to speed up queries. In Proceedings of the International Conference on High Performance Computing and Communication (HPCCE 2021), Haikou, China, 17–19 December 2021; Volume 12162, pp. 150–154.

SoftwareX 25 (2024) 101624

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

DARS: Decentralized Article Retrieval System

Adrian Alexandrescu *, Cristian Nicolae Butincu

Department of Computer Science and Engineering, Faculty of Automatic Control and Computer Engineering, Gheorghe Asachi Technical University of Iasi, 700050, Romania

ARTICLE INFO

Keywords:
Decentralized system
Web crawler
Web scraper
Information retrieval
Distributed computing
Parallel processing

ABSTRACT

DARS is a decentralized article retrieval system designed to bring the community together for parallel and distributed extraction of article content from the web. The system is comprised of three types of components: web crawlers to extract the links from website pages, web scrapers to extract article information from the web pages identified as articles, and a retrieval core to manage the extraction process. To attain decentralization, multiple such systems can be deployed in different locations. When a client queries one of the nodes, the returned information can be an aggregate of data from multiple nodes. The system is flexible and can be adapted to extract different types of information in a decentralized manner.

Code metadata

Current code version	v1.0
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-23-00680
Permanent link to Reproducible Capsule	–
Legal Code License	Apache License 2.0.
Code versioning system used	git
Software code languages, tools, and services used	Java, MySQL, SQLite
Compilation requirements, operating environments & dependencies	JDK 11+, Maven, 64-bit operating system
If available Link to developer documentation/manual	https://github.com/h23ro/decentralized-article-retrieval-system#readme
Support email for questions	adrian.alexandrescu@academic.tuiasi.ro

1. Motivation and significance

The Internet hosts a vast number of online articles like news from different domains, various tutorials, and blog posts, covering a wide range of topics. There are situations in which the published information needs to be obtained and processed automatically to provide relevant insights. Web crawlers are used to extract URLs from web pages and to extract meaningful information.

To have an efficient system, some issues must be resolved. Firstly, handling the data requires multiple resources (computing power, network bandwidth, storage) due to the sheer volume of information that needs to be obtained. When getting the web page contents from a URL, the crawler needs to throttle the requests so as not to get banned by the remote server. Another issue is obtaining the desired information from the web page, which is usually achieved by using an extraction template that pinpoints the exact location of the required data inside the HTML page contents.

The initial motivation for developing the proposed software is fighting disinformation in the online environment and, more precisely, detecting the fake news that is published by online news media. In this paper we present a system that is a variation of a software component from the FiDisD project (Fighting disinformation using decentralized actors featuring AI and blockchain technologies), which is presented in [1] and [2]. There are notable differences between the components used in the FiDisD project and the Decentralized Article Retrieval System (DARS) proposed in this paper. In the FiDisD project, the data acquisition components consist of web crawler instances, which perform the URL extraction, web scraper instances, which handle the article extraction, and an off-chain core component that manages the extraction and stores data. Each crawler/scraper is managed by independent actors and the information is aggregated at the core component. Blockchain technology and majority rules are employed to ensure

* Corresponding author.
E-mail addresses: adrian.alexandrescu@academic.tuiasi.ro (Adrian Alexandrescu), cristian-nicolae.butincu@academic.tuiasi.ro (Cristian Nicolae Butincu).

https://doi.org/10.1016/j.softx.2023.101624
Received 7 October 2023; Received in revised form 14 December 2023; Accepted 18 December 2023
Available online 22 December 2023

Table 1
Comparison of the FiDisD and DARS systems.

Feature	FiDisD	DARS
Core-crawlers-scrapers system deployment	Single deployment	Multiple deployments
Decentralization	Centralized retrieval core with blockchain-based decentralization	Decentralization through peer-to-peer between multiple retrieval cores
URL extraction	Groups of crawlers extract the same list of websites	Each crawler extracts a specific list of unique websites
Article extraction	Groups of scrapers extract the same articles	Each scraper extracts a random list of unique articles
Storage	The core stores all the articles	Each deployment has its own article store
Access to extracted data	Public API	Public API with access to the data stored in other deployments
Interaction with external systems	Article content hash stored on the blockchain network	Self-contained

trust in the system; this is needed because groups of crawlers/scrapers handle the same data. While the majority rule determines the canonical representation of an article, its content is stored offchain. Only the proof of data, in the form of the article's content hash, is stored on blockchain, therefore reducing to a minimum the volume of data that is written on the blockchain.

In the FiDisD project, the decentralization aspect is obtained by employing blockchain technology. For the software proposed in the current paper, decentralization is achieved by deploying multiple retrieval system instances, geographically distributed, that communicate with each other in a peer-to-peer manner.

The main motivation behind this approach is to have a community-based solution that enables an efficient processing of a very large number of websites by extracting URLs and article information from specific lists of target websites. The idea is to have multiple deployments of the extraction system, which constitute the extraction environment, and to have each of them manage the processing of a specific list of websites. Each deployment consists of a retrieval core node (a central server application) and one or more crawlers and scrapers. A public API is made available by each core node that provides access to the extracted article information from that node and all other nodes in the entire extraction environment. This is the embodiment of the community idea, because any entity can become part of the environment and contribute to the extraction process, while also benefiting from the data extracted by the community and, in turn, giving back to the community by contributing to the article database. The solution provides end-users access to a decentralized database with article information. The system is based on trust that no entity enters invalid information in the environment. If one deployment is found to be corrupt, then the access to and from that deployment can be cut off, without interfering with the rest of the environment's function.

Compared to the extraction system developed for the FiDisD project, the decentralized article retrieval system proposed in this paper displays notable differences as presented in Table 1. Each system has its advantages and goals. In FiDisD, all the crawler and scraper actors are not considered trustworthy and the same information is extracted by multiple actors. Even though it is a centralized retrieval core and storage, decentralization is achieved using blockchain technology. The main idea here is to have other actors obtain the information and to store it centrally, while also ensuring trust for the stakeholders. In DARS, there is the idea of the community, which pulls together its resources to obtain the article information, and the assumption that all the actors are trustworthy. Decentralization is achieved by having multiple deployments that communicate peer-to-peer, each one with their extraction logic and storage.

There are many approaches to web crawling and to determine where exactly is the desired information located in the page contents; some of the existing research is summarized in [3]. A tool that analyses Romanian news websites is NewsCompare [4], which is an open-source Java application for detecting news influence by employing a crawler, similarity finder, content indexer and news compare. Although the subject has some similarities to the FiDisD project, the implementation differs significantly compared to both the FiDisD project and the DARS system. The focus of the latter two is on the decentralized and distributed nature of the article extraction and analysis process, while the NewsCompare application focuses more on the content analysis. Other solutions for extracting news articles, but which lack the distributed aspect, are Newspaper3k [5], which is a Python library, news-please [6] and the web news extractor from [7], which use heuristics to determine the location in the web page of the relevant information, and the web crawler from [8], which employs machine learning for extraction, at the expense of accuracy.

There are also distributed generic crawlers. Crawlab [9] is a Golang-based web crawler management platform, which supports various programming languages and different crawler frameworks. It uses docker compose for deploying the application and it also has a graphical interface that provides various statistics. The solution is a generic one and it offers flexibility at the expense of communication and processing overhead. Scrappyd [10] and Gerapy [11] are similar solutions that manage a cluster of crawlers; both of them use the Scrapy [12] web crawling and a scraping Python framework. The main differences between these solutions and DARS are that they do not specifically focus on nor facilitate article extraction, they do not have the decentralized aspect and are based on Python as opposed to the Java language used by DARS.

Another crawler application is RCrawler [13], which is an application written in the R programming language that features parallel URL extraction and content scraping by using multiple threads. Our DARS solution provides multi-threaded extraction for crawlers and scrapers, but also has a decentralized and distributed architecture. The are multiple reviews regarding existing crawling solutions [14–18], which look at the means of navigating through the websites, methods of obtaining the relevant information and increasing the extraction efficiency, but none of them focus on the decentralized and community-based approach to this problem.

2. Software description

The DARS software is designed for extracting article information from websites. Each deployment of the system consists of a retrieval core application with an associated database server and multiple crawlers and scrapers that can be run by anyone who wants to contribute. The retrieval core server acts as the extraction coordinator for the deployment.

The configuration of the retrieval core consists of specifying the extraction template for each site from which articles are to be extracted, and the credentials for the crawlers and scrapers. The information extracted from each article consists of title, contents, featured image,

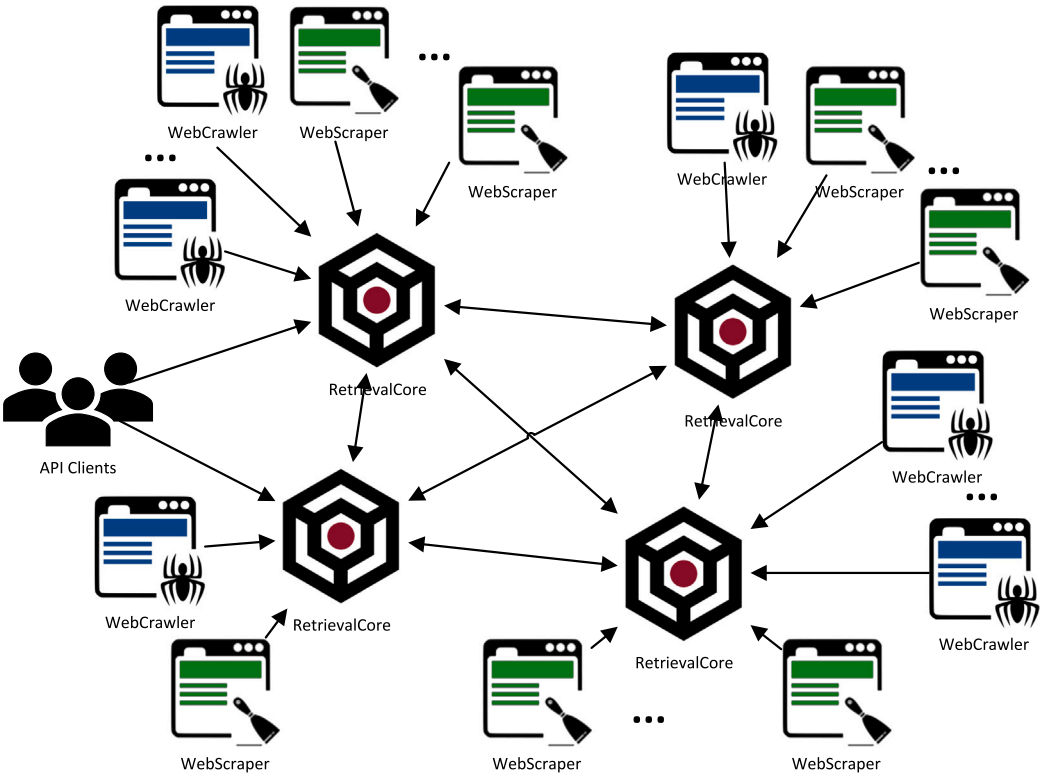


Fig. 1. Decentralized Article Retrieval System (DARS) architecture. The arrows show the direction of communication — the base of the arrow represents the entity initializing the communication.

publish date and author. Each extraction template also provides the means to identify relevant pages, i.e., pages that contain article information. Those means imply the existence or absence of a list of strings in the HTML contents of the web page being analyzed. The creation template must be created manually or through other techniques, which are beyond the scope of this paper. Usually, there is a single template for a specific website, but, if there are multiple templates, a separate file can be created for each template, with different regular expressions that identify the relevant pages belonging to that template.

The actors that are involved in the system are:

- *System administrator* - Configures and runs the core-crawlers-scrapers system, i.e., the database and the retrieval core application;
- *Crawlers/Scrapers* - Applications that crawl/scrap based of the information provided by the retrieval core, after authentication with proper credentials;
- *End-users* - Public API consumers that can obtain extracted article information from any deployment.

After the system administrator deploys a retrieval system, he must register the crawlers and scrapers by generating, for each of them, the credentials needed for authentication and authorization. The retrieval core distributes the tasks to each registered crawler and scraper in the system. When adding a site to a retrieval core, one must manually allocate a crawler actor to handle the URL extraction. The decision was made to have a single crawler extract from a specific site because each crawler has its own local database and this reduces the communication overhead of having multiple crawlers using the same database. On the other hand, each UR identified as containing an article is randomly associated to a specific scraper. Each scraper receives, from the retrieval core, batches of URLs from which to extract article information. One potential issue is having many scrapers extract at the same time from the same site. This can cause a problem for that site, but it can be solved by updating the URL allocation to favor only a small list of scrapers to handle a specific site.

2.1. Software architecture

The DARS environment comprises of multiple retrieval systems, each of them being composed of a retrieval core component and multiple web crawlers and scrapers, as can be seen from Fig. 1. The communication between the retrieval systems is done by means of the API exposed at each RetrievalCore. Each core has knowledge of the other cores in the environment. The actual implementation for knowing the other cores is basic, but functional, and implies that each core keeps a list of the other cores in the environment. Its simplicity allows for the retrieval system to choose the cores with which it interacts. Whenever an API client queries a core node for article information, it can obtain aggregated data from all the other associated nodes. Each retrieval system has its own database and its own crawlers and scrapers. In this way, both the data and the processing are decentralized. All the communication between applications is based on RESTful APIs.

The retrieval core instance uses a MySQL database having the structure from Fig. 2. The stored data is regarding the sites and pages from which the information is extracted, the articles and the users with their roles. Each web crawler instance has its own MySQL database for managing the extracted URLs and their state. Its structure is simple as it stores only the site and page information regarding the crawled data. Each scraper instance has a SQLite database with only two tables for keeping the extraction templates and the article extraction status, and it uses the local file system to store the extracted article information. If anyone wants to adapt the proposed system to extract other information, one can use the existing article structure and expand the article contents to include a JSON with the extra information. This is possible because the article contents not kept in the database, but rather in the file system. On the other hand, better results are obtained by adding extra fields in the article table from the RetrievalCore and updating the extractor template and the extraction logic to include the desired extra information.

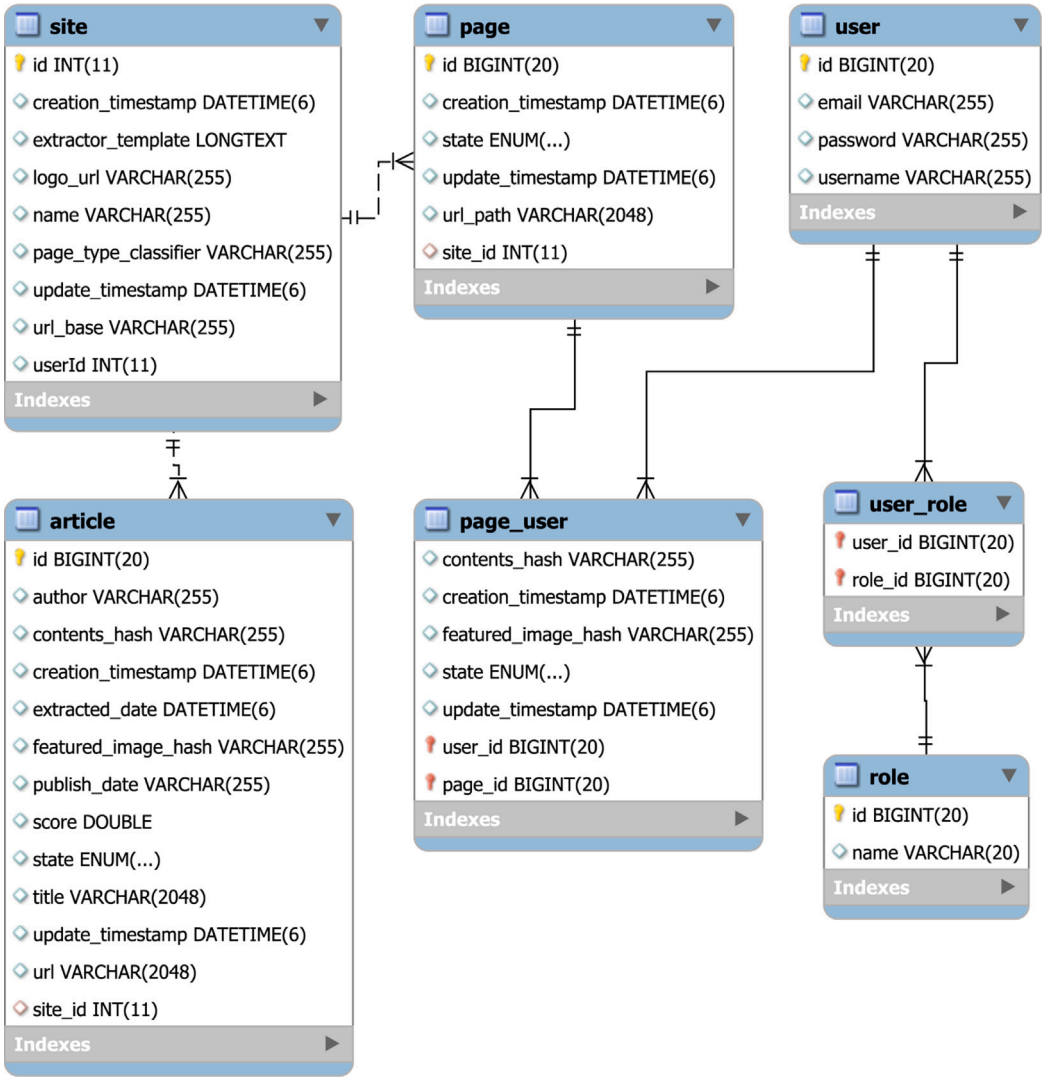


Fig. 2. Entity relationship database diagram for the RetrievalCore component.

2.2. Software functionalities

The major software functionalities are summarized as follows:

- The system provides distributed article retrieval,
- The community can contribute by running crawlers and scrapers, and by deploying their own retrieval system, thus, resulting in a decentralized environment,
- From any retrieval core node, one can access the article information stored on other nodes in the system,
- Only crawlers/scrapers that registered to a retrieval core node can extract URL/article information,
- The article extraction is performed based on an extraction template for each targeted website, therefore eliminating the risk of inaccurate information,
- The environment can be easily adapted to extract any information from the web in a decentralized manner.

3. Illustrative examples

The most relevant illustrative example is the successful use of the lighter version of the DARS software in the FiDisD project. The result was a news aggregator website that ensured the truthfulness of the displayed article information to its users. The extraction template is a critical part of the extraction process and Listing 1 shows an example of an extraction template JSON string, which provides the necessary information for extracting data from the Hotnews website.

Listing 1: Example of an extraction template JSON string

```
{
  "name": "Hotnews",
  "urlBase": "https://www.hotnews.ro",
  "logoUrl":"https://upload.wikimedia.org/
      wikipedia/ro/c/cb/Logo_HotNews.gif",
  "pageTypeClassifier": {
    "containsList": [
      "<article class=\"article-page\""
    ],
    "containsNotList": [
    ]
  },
  "extractorTemplate": {
    "removeElements": ["script, style, div.ad-
        wrapper"],
    "title": ["#main > div > div.left >
        article.article-page > h1.title", "
        text"],
    "contents": ["#main > div > div.left >
        article.article-page > div.article-
        body", "html"],
```

```
    "featuredImage": ["#main > div > div.left
        > article.article-page > div.article-
        body > div.lead-mm > img", "attr:src"]
        ,
    "media": null,
    "publishDate": ["#main > div > div.left >
        article.article-page > div.header.info
         > span.ora", "text"],
    "author": ["#main > div > div.left >
        article.article-page > div.author >
        span > span.author", "text", "true"]
}
```

Let us consider another use-case scenario in which multiple geographically spread organizations want to create a database with news articles from around the world. Each organization handles the news websites from specific countries based on access speed, for efficiency's sake. Each organization deploys an instance of the proposed retrieval system and configures it to extract articles from a specific list of news websites. Other entities in that geographical area can also contribute to the crawling and scraping. An actor that accesses any node can obtain the extracted information from the other nodes. The information can be presented in an aggregated news portal. If, for example, an organization only handles websites in Spanish, then the extracted information can be passed through a Spanish–English translator library. This way, a website containing the aggregated news can show all the news in the same language.

4. Impact

Using the proposed solution allows for decentralized online news extraction by involving different actors and entities for a common goal, i.e., retrieving news articles with geographically distributed crawlers and scrapers. A notable practical example of using the DARS environment is extracting article information from news websites from different countries and in different languages. There can be one or multiple retrieval system instances for each language/country and an extra translation module can be easily integrated. This way, a news aggregator can provide news from all over the world translated in the same language.

An improvement to the existing state-of-the-art in terms of information retrieval is, for example, to use the DARS system to efficiently build a database with existing scientific articles. The extraction template can be extended to extract other types of information and even PDF files. Therefore, multiple entities, e.g., universities, can deploy the retrieval system on their computing resources, and each one can focus on a particular journal publisher or indexing service for journals and conferences.

The fundamental difference of the proposed system, compared to existing parallel web crawlers and scrapers, is the decentralization aspect. Each RetrievalCore component coordinates an independent group of web crawlers and scrapers, but the extracted information can be obtained from all the RetrievalCores by querying any of them. Therefore, we have a decentralized extraction and a decentralized storage system. Another important aspect is the community-based approach to information retrieval, in which any interested party can contribute to extract the desired information, while also providing the extracted data to the other stakeholders.

The next iteration of the system will deal with trust management across participating retrieval cores. In its current state, there is a premise of trust in all the involved actors. Further research needs to be performed to design the trust flow into the system in a completely decentralized manner, including different approaches for reward and penalization mechanisms.

Regarding the performance of the presented system, it highly depends on the resources that are available to each component. Tests using the DARS system and the FiDisD extraction showed that the most time consuming processes were the communication with the database and with the news websites. Performance-related information regarding the FiDisD extraction process can be found in [2].

The retrieval system components can be containerized and deployed on a cloud solution [19], e.g., using Apache Mezos, Contained or Kubernetes. Another alternative is to deploy the system on the OpenStack cloud solution by employing virtual machines for the compute components using the Nova service, Swift for object storage and the Trove service for the database. The idea is to use Neutron to configure the Nova instances for crawling and scraping so that different IPs are used for outgoing requests to the same news web server. The proposed system can be easily adapted to extract more generic information from websites. Using cloud orchestration on OpenStack, i.e., using the Heat service, the retrieval system can be deployed on any OpenStack instance.

5. Conclusions

The novelty of the DARS environment is that it brings together the community for contributing to the information extraction process. The decentralized and distributed nature of the system components offers the possibility of extracting large amounts of information in little time. Another novel aspect is the description and integration of the extraction template, which provides the flexibility of obtaining exact information from web pages that have certain particularities when it comes to the presence or absence of specific elements in the page contents, or when the data is not that straightforward to obtain.

The proposed solution can be easily adapted to extract other types of data by expanding the extraction template and updating the data model. For example, it can be used to extract product information from shopping websites, online leaflets for medicines, cooking recipes, and even information about software applications from public repositories. Basically, it can be used to extract any structured information available on the web in a decentralized manner, and to provide a database with information that can be further analyzed and used for various purposes.

CRediT authorship contribution statement

Adrian Alexandrescu: Conceptualization, Formal analysis, Investigation, Methodology, Software, Writing – original draft, Writing – review & editing. **Cristian Nicolae Butincu:** Conceptualization, Formal analysis, Project administration, Supervision, Validation, Writing – original draft, Methodology.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

I have shared the code repository and the link is provided in the paper.

Acknowledgments

Funding: This research was supported by the project "Collaborative environment for developing OpenStack-based Cloud architectures with applications in RTI" SMIS 124998 from The European Regional Development Fund through the Competitiveness Operational Program 2014–2020, priority axis 1: Research, technological development and innovation (RTI), Romania—the POC/398/1/1 program.

References

[1] Buţincu CN, Alexandrescu A. Blockchain-based platform to fight disinformation using crowd wisdom and artificial intelligence. Appl Sci 2023;13(10):6088. http://dx.doi.org/10.3390/app13106088.

[2] Alexandrescu A, Butincu CN. Decentralized news-retrieval architecture using blockchain technology. Mathematics 2023;11(21):4542. http://dx.doi.org/10.3390/math11214542.

[3] Ren X, Wang H, Dai D. A summary of research on web data acquisition methods based on distributed crawler. In: 2020 IEEE 6th international conference on computer and communications. IEEE; 2020, p. 1682–8. http://dx.doi.org/10.1109/ICCC51575.2020.93451575.

[4] Pop C, Popa A. NewsCompare-A novel application for detecting news influence in a country. SoftwareX 2019;10:100305. http://dx.doi.org/10.1016/j.softx.2019.100305.

[5] Ou-Yang L. Newspaper3k: Article scraping & curation. 2023, URL https://github.com/codelucas/newspaper. [Accessed 04 July 2023].

[6] Hamborg F, Meuschke N, Breitinger C, Gipp B. News-please : a generic news crawler and extractor. In: Gäde M, editor. Everything changes, everything stays the same : understanding information spaces; proceedings of the 15th international symposium of information science. Schriften zur informationswissenschaft, (no. 70):Glückstadt: Verlag Werner Hülsbusch; 2017, p. 218–23.

[7] Dong Y, Li Q, Yan Z, Ding Y. A generic web news extraction approach. In: 2008 international conference on information and automation. 2008, p. 179–83. http://dx.doi.org/10.1109/ICINFA.2008.4607992.

[8] Le Huy Hien N, Tien T, V.H. N. Web crawler: Design and implementation for extracting article-like contents. Cybern Phys 2020;9:144–51. http://dx.doi.org/10.35470/2226-4116-2020-9-3-144-151.

[9] Crawlab. 2023, URL https://github.com/crawlab-team/crawlab. [Accessed 05 July 2023].

[10] Scrapy. 2023, URL https://github.com/my8100/files. [Accessed 06 July 2023].

[11] Gerapy. 2023, URL https://github.com/Gerapy/Gerapy. [Accessed 06 July 2023].

[12] Scrapy. 2023, URL https://github.com/scrapy/scrapy. [Accessed 05 July 2023].

[13] Khalil S, Fakir M. RCrawler: An R package for parallel web crawling and scraping. SoftwareX 2017;6:98–106. http://dx.doi.org/10.1016/j.softx.2017.04.004.

[14] Khder MA. Web scraping or web crawling: State of art, techniques, approaches and application. Int J Adv Soft Comput Appl 2021;13(3). http://dx.doi.org/10.15849/ijasca.211128.11.

[15] Yu L, Li Y, Zeng Q, Sun Y, Bian Y, He W. Summary of web crawler technology research. J Phys: Conf Ser 2020;1449(1):012036. http://dx.doi.org/10.1088/1742-6596/1449/1/012036.

[16] Udapure TV, Kale RD, Dharmik RC. Study of web crawler and its different types. IOSR J Comput Eng 2014;16(1):01–5. http://dx.doi.org/10.9790/0661-16160105.

[17] Kausar MA, Dhaka V, Singh SK. Web crawler: A review. Int J Comput Appl 2013;63(2):31–6. http://dx.doi.org/10.5120/10440-5125.

[18] Dhenakaran S, Sambanthan KT. Web crawler-an overview. Int J Comput Sci Commun 2011;2(1):265–7.

[19] Singh PK, Kumari M. Containers in openstack: leverage openstack services to make the most of docker, kubernetes and mesos. Packt Publishing Ltd; 2017.

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 12 February 2024, accepted 19 April 2024, date of publication 29 April 2024, date of current version 6 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3394537

RESEARCH ARTICLE

Design Aspects of Decentralized Identifiers and Self-Sovereign Identity Systems

CRISTIAN NICOLAE BUTINCU AND ADRIAN ALEXANDRESCU
Department of Computer Science and Engineering, Faculty of Automatic Control and Computer Engineering, Gheorghe Asachi Technical University of Iaşi, 700050 Iaşi, Romania

Corresponding author: Cristian Nicolae Butincu (cristian-nicolae.butincu@academic.tuiasi.ro)

This work was supported in part by the Project "Collaborative Environment for Developing OpenStack-Based Cloud Architectures with Applications in Research, Technological Development and Innovation (RTI)" from European Regional Development Fund through the Competitiveness Operational Program 2014–2020, priority axis 1: RTI—the POC/398/1/1 Program under Grant SMIS 124998.

ABSTRACT The increased digitalization of society raises concerns regarding data protection and user privacy, and criticism on how the companies handle user data without being transparent and without providing adequate mechanisms for users to control how their own data is being processed or shared. To address this problem and open the way for a secure and efficient society, where the privacy of citizens is paramount, the identity concept and proof of identity mechanisms need to be redesigned from the ground up. In this paper we discuss how the emerging Web3 technologies like distributed ledger technology (DLT), blockchain, smart contracts, decentralized storage systems, and crypto wallets can be leveraged to design and implement a decentralized digital identity system based on decentralized identifiers (DID) and self-sovereign identities (SSI). Such a system puts the users in full control over their own data while also providing a solid backbone for building interoperable systems that are secure, scalable, and efficient. We propose different architectures for the decentralized identity infrastructure and storage layer, and also discuss the mapping of these architectures on cloud platforms. The main goal is to provide an architectural blueprint for a scalable, secure, privacy-preserving and trusted system.

INDEX TERMS Blockchain, crypto wallets, decentralized identifiers, distributed ledger, distributed storage, self-sovereign identity, smart contracts.

I. INTRODUCTION

In recent years, we are witnessing an accelerated rate of digitalization in our societies. This transformation, although it can lead to efficient societies, must be performed with great care. As more and more users get involved with online digital processes, data protection becomes extremely important. Personal data-related activities like collection, processing, storage and legal conformance pose big challenges to companies. Advancing digitalization in all fields, ranging from mobility, administration, health and financial services to entire smart cities is not possible without a well-defined digital identity for the entities that access the system (e.g., citizens, authorities, universities, public and private companies).

The associate editor coordinating the review of this manuscript and approving it for publication was Engang Tian.

The digital identity represents its users when interacting with digital services. Following this interaction, the service providers can acquire great amounts of personal data from users. In some countries, the management of personal data is governed by legislation (e.g., in Europe, General Data Protection Regulation (GDPR) [1]). Moreover, the amount of collected data increases with the level of service customization. This data is stored in data silos (on company premises, data centers, cloud etc.). However, the details regarding data storage and security measures taken to secure the data remain opaque to the end users.

Most online systems rely on centralized or federated identity management systems. Basically, the federated authentication through a Federated Identity Management (FIM) system is an agreement between an organization and an identity provider, where the latter authenticates the user so that the organization no longer manages the authentication

process or user credentials. Common examples are services that allow users to login with their existing social network credentials (e.g., Facebook, LinkedIn). In these cases, social networks are the identity providers. This approach to identity management and user authentication has proved to be unreliable and unable to provide the security needed to guarantee the user privacy and to protect user personal data.

Another example is an organization with an identity management system that provides integration with third party solutions like Microsoft Office 365 or Google G Suite. In this situation, the users have no control over how much of their personal details are shared by the organization with third party solutions. A practical scenario is an identity management solution for university students and teachers integrated with third-party solutions that facilitate online teaching and learning [2]. The students and teachers have no control over what the university is sharing with the external systems.

Since the inception of these systems, massive data breaches that exposed Personal Identifying Information (PII) occurred regularly. Some high profile examples of data breaches are: Yahoo! (2013-2016) [3] – over 3 billion user accounts exposed over a period of 3 years, First American Financial Corp. (2019) [4] – 885 million file records leaked (bank account numbers, bank statements, mortgage payments documents, wire transfer receipts with social security numbers, drivers' licenses), Facebook (2021) [5] – 533 million users exposed (names, phone numbers, account names, and passwords), Facebook/Cambridge Analytica (2018) [6] – 87 million users exposed, LinkedIn (2021) [7] – over 700 million user records scraped (>93% of the total user base), JPMorgan Chase (2014) [8] – affected 76 million households and 7 million small businesses, Marriott International (2018) [9] – 500 million affected guests, Adobe (2013) [10] – 38 million credit card numbers, eBay (2014) [11] – 145 million affected users.

Given the context, in recent years, there is a raising concern from security experts regarding data protection and user privacy [12], [13] and several studies reveal that companies fail to protect user data [14], [15], [16]. There is a lot of criticism on how the companies handle user data without being transparent [17] and without providing adequate mechanisms for users to control how their own data is being processed or shared [18].

The increased reliance on technology combined with the increased speed of technological developments and the rate at which new services, devices and platforms that require user data are being released, is leading people to surrender much of their private data without considering the long-term effects. This is a major societal-level privacy concern that is currently and primarily addressed with efforts of regulation, in a manner that can be classified reactive at best [19].

A paradigm shift is required by companies, individuals and authorities when it comes to data protection and user privacy. It becomes clear that relying on centralized or federated identity management systems cannot provide the required guarantees to the user privacy that a truly digital society needs. To address this problem, a new approach to identity management has to be conceived.

This paper discusses how emerging Web3 technologies, like distributed ledger technology (DLT) [20], blockchain [21], smart contracts, decentralized storage systems and crypto wallets can be leveraged to design and implement a decentralized digital identity system based on decentralized identifiers (DID) and self-sovereign identities (SSI). Such a system has the potential to provide a smart and innovative solution to the digital identity problem.

The scope of this paper is not to provide a step-by-step recipe or to describe a particular prototype that uses one of the many possible combinations of Web3 technologies and distributed architectures, but to provide a general overview of how these technologies can be combined and what are the possible distributed architectures that can be employed. Depending on a particular setup that is constrained by computing costs, available resources, network infrastructure, consensus algorithms and blockchain implementations, a different architecture from the proposed ones can be used.

II. CONCEPTS

In this section, we introduce a series of concepts that are used throughout the paper. We discuss the notions behind these concepts, why they are needed and give examples of usage.

A. GLOBALLY UNIQUE IDENTIFIERS

A globally unique identifier is a type of identifier for an entity, (e.g., individual, organization, system etc.) used in a wide range of use cases like communication, identification, tracking, URLs (Uniform Resource Locator) etc. Examples of globally unique identifiers are phone numbers, email addresses, ID numbers (e.g., ID card, passport, tax ID, health insurance), product identifiers (barcodes, serial numbers), URIs (Uniform Resource Identifier) and so on.

The vast majority of information systems do not allow entities identified by these globally unique identifiers to exercise control over them. Usually, these identifiers are issued by external authorities that have full control over who or what they refer to, including revocation aspects. Moreover, they are usually useful only in certain contexts and recognized only by a limited number of entities and might also reveal unnecessary personal information. They might also cease to be valid with the failure of the organization that controls them. Without proper security mechanisms, they may be replicated and used by malicious third parties in a type of action known as ''identity theft''.

B. SIMPLE DIGITAL IDENTITY

A simple digital identity is created whenever a user goes through the registration process on a website to create an account. As part of this process, personal data such as the name, email address, phone number, address and other account details are usually required. This kind of digital identities can be created either on simple websites or on social

networks or other big companies' websites that can also act as identity providers. For example, many websites allow users to login using credentials from social media accounts.

C. SELF-SOVEREIGN IDENTITY (SSI)

In the physical world, a user can provide proof of his identity by using an identity card, which is an official document issued by authorities and considered to be forgery-proof. The digital twin of the identity card in the digital world would be a digitally signed identity document. However, this digitally signed identity document has limited use. A full-featured digital identity contains more than the simple identity of the owner; aside basic personal data, this digital identity can control an unlimited number of credentials that can be used to authenticate the user on different systems and on different service providers. Moreover, the user that controls this digital identity can share only the pieces of information required for a particular use case. For example, when a user is required to prove that he is over 18 years old when buying alcohol drinks, he can prove just this information alone, without disclosing his full identity that contains other personal data. Therefore, a digital identity is more than just a simple identity card counterpart from the physical world.

A digital identity that is fully controlled by the user and allows him to selectively share and prove information to third parties is called a self-sovereign identity. It can contain, besides basic personal identifying information, other type of credentials such as: driver's license, diploma accreditations, work certificates, mobility tickets, insurances, credit card information, biometric data, health information and so on.

D. DECENTRALIZED IDENTIFIER (DID)

A decentralized identifier [22] is a new type of identifier that enables verifiable, decentralized digital identity. A DID can represent any entity (e.g., person, organization, thing etc.). By design, DIDs can be entirely decoupled from centralized registries, identity providers and certificate authorities. This contrasts with typical federated identifiers. While an entity can prove control over a DID without involving other entities, other parties might still be involved to enable the discovery of related DID information. A DID is an URI that associates a DID entity with a DID document that describes and enables trustable interactions with the DID entity. For example, a DID document can describe cryptographic-based data (e.g., public keys), verification methods and services, that can be used by the DID entity to assert its control over the associated DID.

A DID is a new type of globally unique identifier, designed to enable an entity to have full control over its creation by using systems it trusts. It enables the entity to prove control over it by using cryptographic proofs such as digital signatures, while also providing control over how much private data should be revealed, without depending on a central authority.

In terms of interoperability, DIDs can also be created based on identifiers registered in centralized or federated identity management systems, thus bridging the gap between all these systems. Depending on the overall context, specific types of DIDs can be designed for different computing infrastructures, such as distributed ledgers, distributed databases, decentralized file systems, peer-to-peer networks and so on.

E. CREDENTIALS

Credentials are used to prove some qualifications, qualities or abilities for particular entities. For example: a driver's license asserts the ability to drive a vehicle, a university degree asserts the level of education etc. While providing great benefits in the physical world, they are difficult to implement in the digital world as they require a properly designed infrastructure and a wide range of security mechanisms.

F. VERIFIABLE CREDENTIALS

A verifiable credential [23] provides a standard way to implement credentials in the digital world in a manner that is cryptographically secure and machine-verifiable, while also respecting user privacy. It can encapsulate the same information that a physical credential represents, while also providing increased security with the addition of advanced cryptographic methods (e.g., digital signatures, zero-knowledge proofs), making it more tamper-proof and more trustworthy than its physical counterpart.

G. VERIFIABLE PRESENTATIONS

Verifiable presentations are assembled from verifiable credentials by the credentials' holder to provide some proofs regarding their identity or their abilities or any other characteristics, disclosing the minimum amount of private data and without involving any other third party in their generation. Based on verifiable presentations, verifier entities can quickly and easily check that a particular entity holds credentials with certain characteristics. Therefore, verifiable presentations are more convenient to be used to establish trust at distance than their physical counterparts.

When implementing verifiable credentials and verifiable presentations, a wide range of security and privacy aspects must be considered to guarantee user privacy, user control, data protection and identity preservation. This is because the persistence of digital information and the ease with which separate sources of data can be collected and correlated triggers serious privacy concerns for security experts.

III. RELATED WORK

In the past years there has been an increased interest in developing decentralized identity systems, both as public and private initiatives.

Starting 2018, the European Commission and the European Blockchain Partnership (EBP) have been developing the European Blockchain Services Infrastructure (EBSI) [24], a network of distributed blockchain nodes across Europe. It is the first EU-wide blockchain infrastructure, driven by the public sector, offering cross border public services. One of its Core Services is the Self-Sovereign Identity service, that aims to provide a decentralized identity management solution for

citizens and businesses. EBSI is a public read, permissioned write blockchain, controlled by EU member states.

eID [25] is a set of services being developed by the European Commission to enable the usage and recognition of national electronic identification schemes (eID) across borders. By using their national eIDs, European citizens will be able to access online services from other European countries. While there are many eID-like initiatives across Europe (e.g., BankID [26] in Sweeden, BankID in Norway [27]), they do not provide the same user privacy benefits as SSI.

The European Self-Sovereign Identity Framework (ESSIF) [28] is based on the principles of SSI and is part of EBSI. ESSIF aims to implement a generic and interoperable SSI framework, defining the necessary specifications and building support services that will allow European citizens to manage their own digital identity without having to rely on a single centralized authority.

eSSIF-Lab [29] was an EU-funded project aimed at advancing Self-Sovereign Identities as a next generation, open and trusted digital identity solution.

The ''Self-Sovereign Identity for Germany'' (SSI4DE) [30] project is a blockchain-based SSI system that aims to allow users to store and manage their own identity information and to set up an identity network distributed across Germany.

Another project is the IDunion [31] research project, funded by the German Federal Ministry for Economic Affairs and Climate Protection as part of the innovation competition ''Showcase Secure Digital Identities''. The aim of the project is to build an open ecosystem for decentralized SSI for natural persons, companies and things, based on DLT. It builds on the results of previous projects, such as LISSI (Let's Initiate Self-Sovereign Identity) [32] and SSI4DE.

The Ego Company GmbH [33] provides solutions to build and implement SSI systems, while enabling interoperability across different ecosystems.

It has long been recognized, especially in Europe, that the data sovereignty of users is threatened by the big tech companies. Germany assumed the pioneering role in researching and implementing SSI technologies. Moreover, it issued legislation, such as The German Online Access Act (Onlinezugangsgesetz, OZG) [34] that legally obligates the public administration in Germany to provide most of its services digitally by the end of 2022. The law came into effect in 2017.

The research regarding DID and SSI is also covered by the scientific community, and one of the main focuses is authentication. In [35], the authors propose a generic decentralized OpenID Connect Provider, which allows the user to choose from a large number of identity providers. Applications of DID for authentication can extend to communication between vehicles (i.e., vehicular ad-hoc networks) for secure registration and authentication [36], to health care for maintaining trust and ensuring privacy [37], [38], to smart homes for access control to smart devices [39], [40], or, to the Internet of Things (IoT) in general for identification and authentication of devices [41].

There is little research in terms of DID and SSI involving cloud solutions. In [42], the authors propose a service model for authentication of researchers based on DID in a cloud solution used to store medical records. The authors focus only on the cloud Common Data Model published by Microsoft and provide no information regarding the decentralized identity infrastructure and how it relates to the employed cloud services. A decentralized identity and access management of cloud instances is presented in [43]. The authors used the AWS cloud with eight virtual machines to evaluate the user registration and identity verification duration, but with no decentralized identity infrastructure deployed on the cloud solution.

IV. DESIGN OF THE SYSTEM

When designing a system that is meant to be scalable, secure, privacy-preserving and trusted by its participants, several aspects must be considered. When dealing with private and personally identifiable information, the accent must be placed on the security measures that guarantee data protection and user privacy. Such a system must be designed so that data breaches are impossible, while also hindering data correlation. The latest advances in cryptography field and Web3 technologies make it possible to design and implement systems of this kind.

Registering a digital identity and depositing personal data with online companies is something normal for today's users. However, due to the many possibilities, the number of digital identities possessed by each user increases significantly. Companies store large amounts of personal information about their users, while failing to provide control mechanisms over how this data is being processed or stored. This is in stark contrast to decentralized identity systems based on DID and SSI, where users bring their own identities with them. In this case, their sensitive data is completely managed and stored in their own digital wallet, and, using cryptography, the users' identity can be established beyond doubt.

In the previous sections we have introduced the current approaches to digital identity and proof of identity used by centralized and federated identity management systems and discussed their shortcomings, like security problems, massive data breaches, lack of privacy and lack of user control. A paradigm shift for identity and proof of identity is needed so that the future systems can provide the needed support for our digital societies. This paradigm shift places the user at the center of the system, making him self-sovereign and in full control over its identity.

In the next sections, we will present different approaches that can be employed when designing secure identity management systems and we will also discuss the pros and cons of each approach. The building blocks of these systems revolve around the concepts of DID and SSI.

The goal is to design an identity management system that is scalable, secure, privacy-preserving and trusted by its

participants, while also designing mechanisms for: trust and trust registries; credential issuance, management and storage; credential verification; credential revocation.

A. SECURITY AND PRIVACY OF DIGITAL IDENTITIES

The identification process must be secure for all interacting entities (e.g., users, companies, software programs etc.). The entities can use a self-managed SSI digital wallet to store all their sensitive information (e.g., digital identities, credentials and other private data). The digital identities are stored in the wallet as cryptographic documents, signed by trusted issuers. These documents can be used in further interactions as proof of identity. By using the SSI digital wallet, users can share only the minimum amount of information needed to prove their identity or any other characteristics. This approach to digital identity also minimizes bureaucratic processes, while keeping users in full control over their private data. Moreover, this also simplifies the registration processes and reduces the data protection effort in dealing with sensitive data that no longer needs to be stored on local servers, thus eliminating the risk of data breaches.

B. SSI TRUST TRIANGLE

Part of the Web3, SSI represents a new way of managing and verifying digital identities, that revolves around the trust concept and how the trust between different participants can be established. Specifically, in the SSI model, trust is established by following a triangle-like structure with three participants: Issuer, Holder and Verifier. Figure 1 illustrates the basic concept of the SSI Trust Triangle.

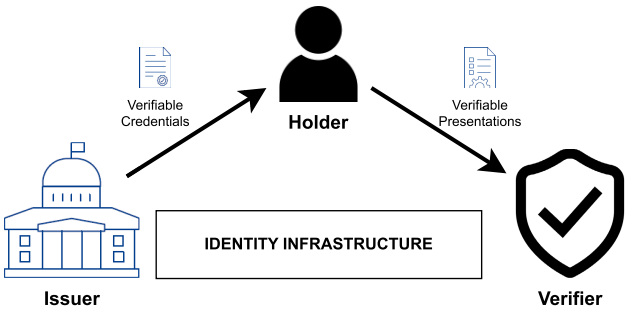


FIGURE 1. SSI Trust Triangle.

The Issuer is an entity that creates and issues digital credentials in the form of verifiable credentials. These credentials contain information about an individual (e.g., name, address, phone number, nationality, date of birth, qualifications etc.), and are cryptographically signed by the Issuer to attest to their authenticity. Examples of Issuers are authorities, universities, employers and so on. It is the Issuer's responsibility to verify the identity or qualifications of an individual before issuing a verifiable credential. These credentials are then stored in the individual's digital wallet.

The Holder is an individual who receives and stores verifiable credentials issued by an Issuer. These credentials are stored in the individual's digital wallet and can be presented to Verifiers, assembled as verifiable presentations, when required to prove the identity or the qualifications. Holders have full control over their digital wallets, and therefore over their stored credentials, and can choose which credentials, or part of credentials, to share with Verifiers.

The Verifier is an entity that verifies the identity or qualifications of an individual. Examples of Verifiers are potential employers, authorities, service providers and so on. When the Verifier asks for specific credentials from the Holder, this one presents them directly from his digital wallet, in the form of verifiable presentations. At this point, the Verifier can verify the authenticity of the credentials by validating the cryptographic signatures of the Issuer.

An important aspect of the trust triangle can be summarized as follows: the trust that the Verifier has in the Issuer is transferred to the Holder. This is possible due to three things:

- the Verifier trusts the Issuer;
- the Verifier trusts that the Issuer performed all the necessary operations to thoroughly check the Holder's identity and qualifications before issuing the credentials;
- the credentials are cryptographically signed by the Issuer, making them tamper-proof and impossible to be forged.

C. GENERAL ARCHITECTURE

Figure 2 illustrates the general architecture of a decentralized identity management system based on DID and SSI.

The flow for creating, establishing and providing proof of identity is illustrated in Figure 2 as a sequence of numbered steps, which are:

1) The user configures his Digital Wallet and creates his DID.
2) Anchor DID to Identity Infrastructure* (depending on system security levels, the anchor might be required to be signed by a trusted Issuer; this implies a preflight call to the Issuer; in this case, it is also possible for the Issuer to anchor the user's DID to Identity Infrastructure).
3) The user requests Verifiable Credentials from the Issuer.
4) The Issuer anchors the Verifiable Credentials to Identity Infrastructure.
5) The Issuer sends the Verifiable Credentials to the user; these are stored in the user's Digital Wallet.
6) The user assembles Verifiable Presentations from Verifiable Credentials and sends them to Verifier/Service Provider.
7) The Verifier/Service Provider checks the Verifiable Presentations and also checks the Identity Infrastructure.
8) The user receives the response from Verifier/Service Provider.
9) Once verified, the user can proceed interacting with the Service Provider.

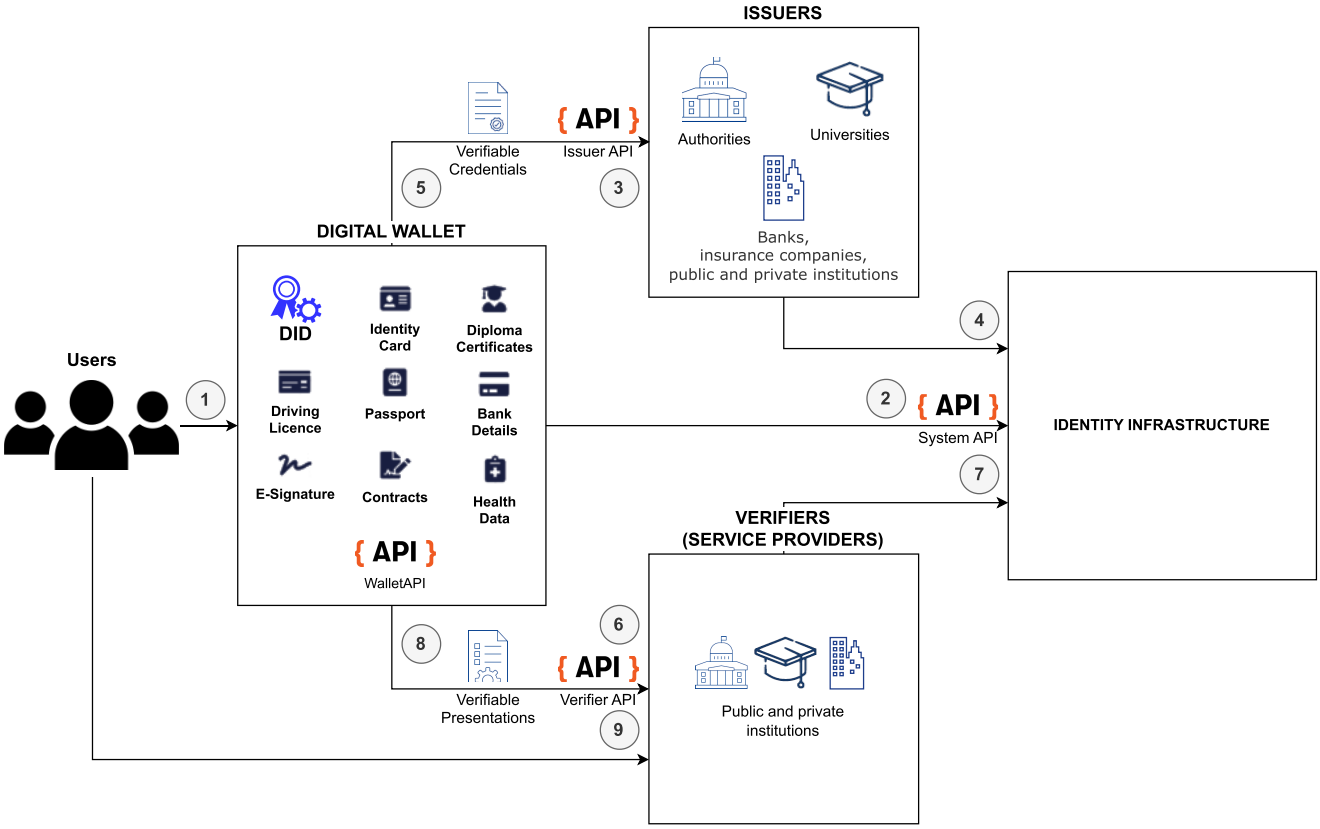


FIGURE 2. The general architecture of a decentralized identity management system based on DID and SSI.

D. SYSTEM COMPONENTS

Considering the drawbacks and the scalability, security and privacy issues of centralized and federated identity management systems, the main goal is to provide the blueprints for an identity management system that is scalable, secure, privacy-preserving and trusted by its participants. The proposed architecture, which is built around the concepts of trust transfer, as introduced by the SSI Trust Triangle, DID and SSI, achieves that goal.

Besides being secure, the scalability and resilience aspects are very important. Thus, well-designed architecture should be decentralized and eliminate all potential single points of failure. In the illustrated general architecture, we can identify the following components:

- Users
- Digital Wallets
- Issuers
- Verifiers/Service Providers
- Identity Infrastructure

Users and Digital Wallets are usually linked together, as every user controls his own digital wallet, and are inherently distributed. Issuers and Verifiers/Service Providers run their own systems, thus are also distributed. The only component that remains to be analyzed is Identity Infrastructure.

E. IDENTITY INFRASTRUCTURE-PROPOSED ARCHITECTURES

Identity Infrastructure is a very important component as it provides the necessary support for the entire system. It is responsible for mediating trust, keeping trust registries, anchoring cryptographic proofs, assisting DID and SSI operations, offering support for Issuers and Verifiers/System Providers, providing credential revocation mechanisms, providing storage space for credential backups (optional feature), providing support for notification services etc.

There are multiple ways to design this component and to choose the technologies that can be used for its implementation. Technology agnostic, at its most basic form, the Identity Infrastructure is a storage system with services needed to manage the information stored at its level.

To be a viable component in a well-designed architecture of the general identity management framework, the Identity Infrastructure component should also exhibit a series of characteristics such as scalability, availability, security, resilience, traceability and privacy. These requirements eliminate all potential architectures that introduce single points of failure at any level. Simply put, the Identity Infrastructure component must be completely decentralized, and, to emphasize this, we will further use the term Decentralized Identity Infrastructure (DII) component.

The reason we have eliminated all potential architectures that introduce single points of failure at any level is that, in case of an event that brings down a subcomponent that is classified as such, the entire system ceases to function. Moreover, a single point of failure component tends to become a bottleneck of the entire system, which leads to performance degradation and scalability issues. This contradicts our goal to provide a well-designed architecture of a system that, among others, is scalable, secure and resilient.

With all these in mind, a potential technology agnostic architecture of the DII component is given in Figure 3.

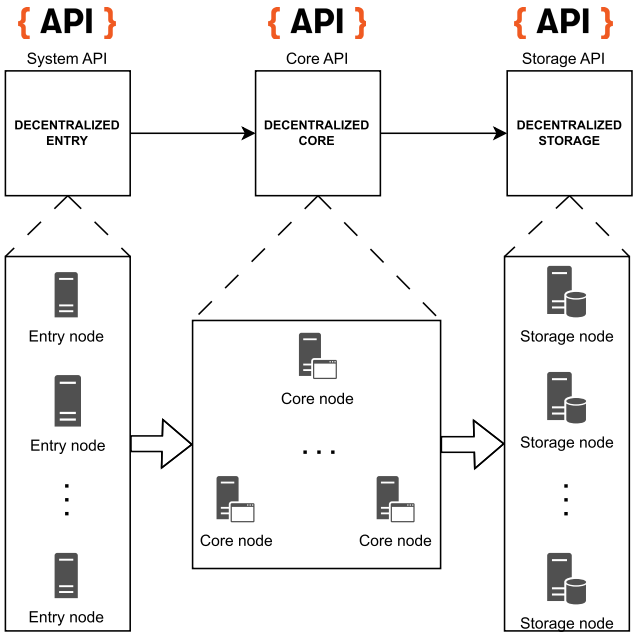


FIGURE 3. The general architecture of the DII component.

As it can be seen in Figure 3, the DII component is designed on three separate layers. This is the recommended design for a scalable, secure and high-availability system. It features a modular design, loosely coupled layers and a clear separation of roles. This also leads to a highly efficient and maintainable system, paving the way to integrate design patterns, such as Separation of Concerns and Single-Responsibility Principle, more easily. Moreover, all access into the system is tunneled through the entry layer. This is the point where orthogonal functionalities can be deployed, like Identity and Access Management (IAM), logging, behavior analysis etc. The layers of the DII component are:

- Entry Layer – this layer acts as a front-end for system users, controls security aspects and supports orthogonal functionalities; can be accessed from exterior through the System API.
- Core Layer – comprises computing nodes that carry out the business logic of the system; the Entry layer accesses it through the Core API.
- Storage layer – comprises storage nodes and is responsible for recording the entire system state, specifically, the data that the decentralized identity management system relies upon to mediate trust and assist the Issuers, Verifiers and Holders in their interactions; the Core layer accesses it through the Storage API.

As a side note, the Storage layer can be implemented as a standalone project, and many different implementations can be provided. Possible architectures for the Storage layer are illustrated in Figure 4.

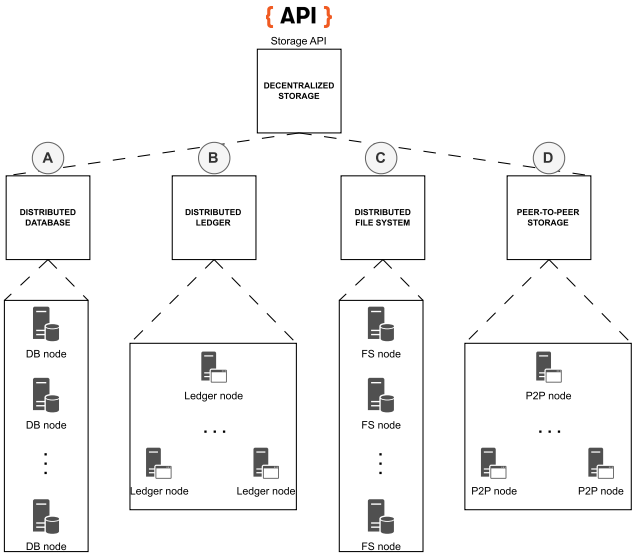


FIGURE 4. Decentralized Storage architecture.

If implemented as a standalone project or developed as a separate component to be integrated into the DII component, it can follow the same architecture as the one of DII. Specifically, when the Storage Layer represents a distributed database or a distributed file system (cases A and C), it can have either two dedicated layers in front (similar to Entry Layer and Core Layer), or only one layer (a Service Layer that provides access to the underlying storage); when the Storage Layer represents a distributed ledger, a storage system described by the DLT technology, or a peer-to-peer (P2P) storage network [44] (cases B and D), an optional layer can be placed in front (a Gateway Layer that offers high performance access nodes to the underlying storage network); a Core Layer is not needed in these cases, because the required logic is present at DLT and P2P level.

A possible architecture of the DII component, based on front-end/back-end servers, is illustrated in Figure 5.

This architecture contains three layers: Front-end Layer, Back-end Layer and Storage Layer, that map one-to-one on the general architecture of the DII component and is mostly familiar to the distributed application developers.

Not every architecture includes an Entry Layer. Figure 6 represents such an architecture.

This architecture contains only two layers: Service Layer and Storage Layer. It is similar to the Front-end/Back-end architecture of the DII component, only that the Service Layer maps to both Front-end Layer and Back-end Layer. This is not a recommended approach since it decreases the

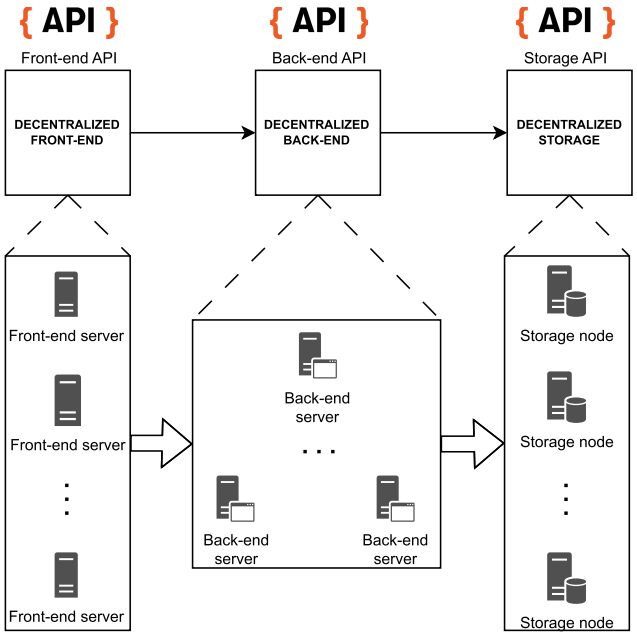


FIGURE 5. The Front-end/Back-end architecture of the DII component.

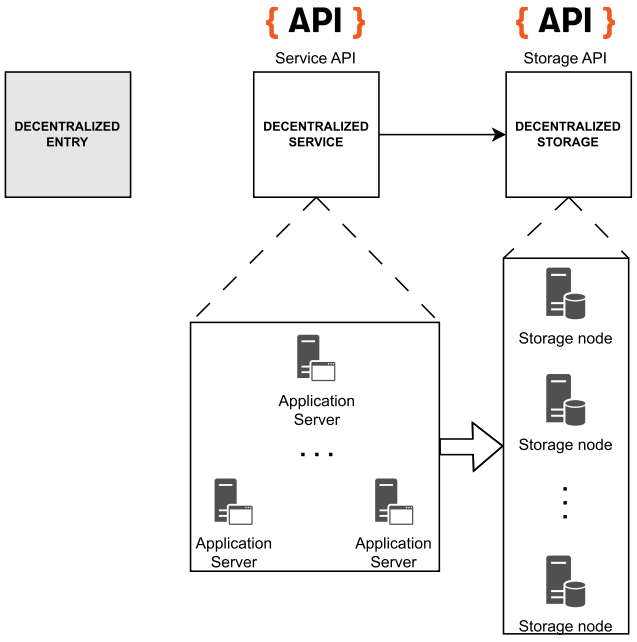


FIGURE 6. The Service architecture of the DII component.

maintainability of the system and poses security implications since all IAM checks must be performed on all nodes in the Service Layer.

The next architecture, illustrated in Figure 7, is based on blockchain, a Distributed Ledger Technology, that records and secures the data in a series of blocks linked together via cryptographic operations.

This architecture has only one layer, the Blockchain layer, that maps on all layers of the general architecture of DII component. It encapsulates the logic of the Entry layer as all security checks are already performed by the blockchain

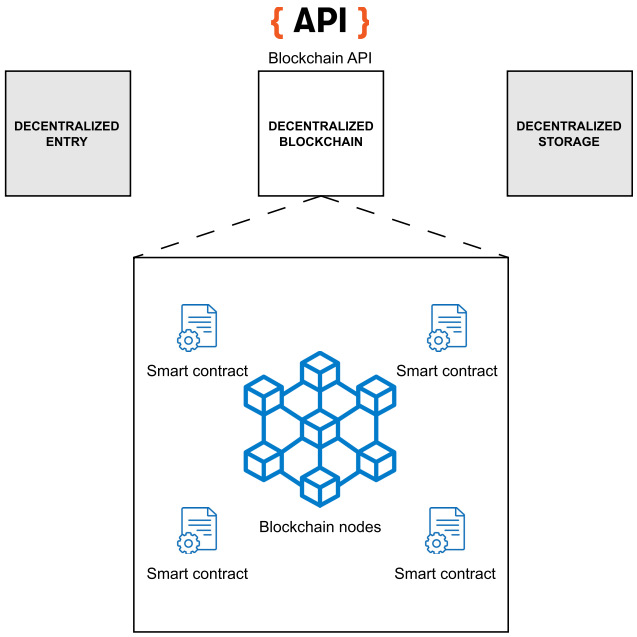


FIGURE 7. The Blockchain architecture of the DII component.

nodes by using public-key cryptography; moreover, every node in the blockchain network acts as a front-end node to the entire system. It encapsulates the business logic of the Core Layer into a series of smart contracts that also reside and execute on blockchain. Blockchain nodes also act as storage nodes, therefore this layer also encapsulates the Storage Layer of the DII component.

The Blockchain architecture of the DII component can be further improved, as illustrated in Figure 8.

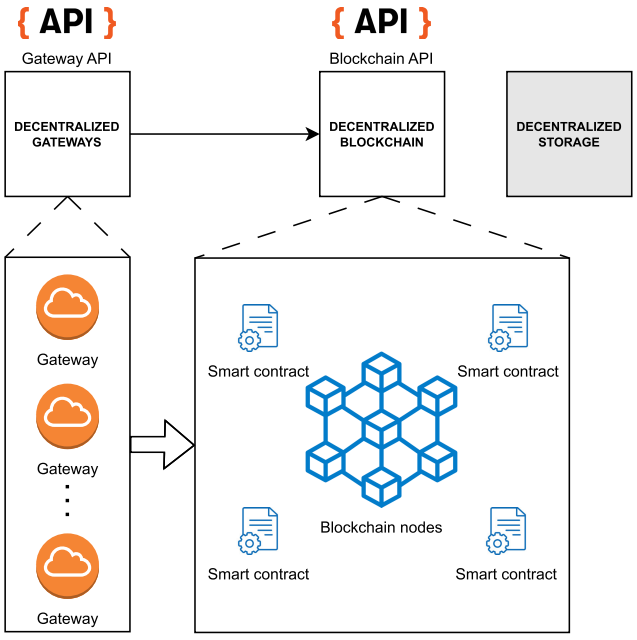


FIGURE 8. The Gateway/Blockchain architecture of the DII component.

This architecture adds the Gateway Layer in front of the Blockchain architecture of DII component. This layer maps to the Entry layer of the general architecture of DII component. As the blockchain can be accessed using any node in the network, not all nodes offer the same performance. The role of the Gateway Layer is to provide high performance, high availability access nodes into the blockchain network. Example gateway node providers to blockchain networks: Cloudflare [45] for Ethereum (layer 1 blockchain network) and Polygon (layer 2 sidechain blockchain network); Infura [46] for Ethereum, Polygon, and for Optimism and Arbitrum (layer 2 blockchains).

F. CLOUD MAPPING

In this section, we show how different architectures of the DII component can be mapped on cloud services of four major cloud solutions: OpenStack [47], Amazon Web Services (AWS) [48], Google Cloud Platform (GCP) [49] and Microsoft Azure [50]. Special consideration is given to OpenStack, which is an open-source cloud platform that can be deployed on local data centers, while the other three are popular commercial cloud solutions that also include free tiers of usage.

Almost any distributed system can be deployed on a cloud solution. In [51], the authors present a framework for taking a distributed system and deploying it to a cloud platform. There, the main services offered by each considered cloud provider are identified, and various advantages and disadvantages are emphasized. An instance regarding a distributed system and blockchain technology, for which the framework was used, is presented in [52], which discusses the aspects of deploying a decentralized news retrieval system on the OpenStack cloud.

Regarding the system proposed in this paper, we will look first at the Front-end/Back-end architecture of DII component illustrated in Figure 5, as it contains mappings for all three layers of the general architecture of DII component. Considering the aforementioned framework, the cloud mapping is as follows:

- Front-end Layer: mapped to a cloud layer of compute instances. Each instance will run front-end logic. A load balancer can be used in front of this layer to route incoming requests to the compute instances.
- Back-end Layer: mapped to a cloud layer of compute instances. Each instance will run back-end business logic.
- Storage Layer: mapped to database instances (case A – distributed database), compute instances (case B – distributed ledger), file-system instances (case C – distributed file system), compute instances (case D – peer-to-peer storage).

Compute instances represent virtual machines. Although the logic of provisioning virtual machines on different cloud platforms follows the same basic steps, it implies using different services: OpenStack Nova service for managing virtual servers and Neutron for networking; AWS Elastic Compute Cloud (EC2), GCP Google Compute Engine, Microsoft Azure Virtual Machine.

One of the disadvantages of using OpenStack is that it does not provide an out-of-the-box solution for Platform-as-a-Service (PaaS), with existing third-party solutions lacking proper support. Under a PaaS solution, which is supported by AWS, GCP and Azure, the management of compute instances is done automatically by the cloud provider, that scales the number of instances up or down depending on predefined metrics (e.g., number of concurrent users, number of requests/second, average CPU utilization etc.). The compute instances from the Front-end Layer and Back-end layer can easily be deployed on AWS Elastic Beanstalk, Google App Engine or Microsoft App Service.

These services cover the cloud mappings of the Front-end Layer and Back-end Layer. For the Storage layer, as it can have different implementations (cases A, B, C and D of the Decentralized Storage architecture - Figure 4), different mappings are required:

- Case A – distributed database: each cloud provider offers a wide range of storage solutions and database types, ranging from simple in-memory databases to relational and document databases to NoSQL databases to entire database clusters. A few examples of data storage solutions are OpenStack Trove, AWS RDS, Google SQL, Azure Database for MySQL.
- Case C – distributed filesystem: each cloud provider offers filesystem-like storage services. Examples: OpenStack Swift service, AWS Simple Storage Service (S3), Google Cloud Storage, Microsoft Storage.
- Case B – distributed ledger, Case D – peer-to-peer storage: these cases are mapped to compute instances that use their own storage. Instead of having a compute instance with application and storage logic, an optimization for these layers is to use containerization and create one container for the application logic (DLT / P2P) and another container for the storage solution. The containers can be run in a cluster of container daemons. Cloud services that provide the means to work with containers: OpenStack Zun and Magnum, AWS Elastic Container Service (ECS), Google Kubernetes Engine, Azure Kubernetes Service. The downside of this approach might be a small overhead due to the extra layer of containerization; however, the benefits are more efficient usage of CPU time.

The system deployment can use the OpenStack Heat orchestration service. This service communicates with the Glance service to specify the image used to create the instances, Cinder for the virtual disk drives, Neutron for networking and the other services required to run the solution. Using Heat Orchestration Template (HOT), we can configure the compute instances by specifying the image, the characteristics (i.e., number of vCPUs, disk size, RAM size), the networking details and the script that should run after the instance is created. For each layer, the script performs the

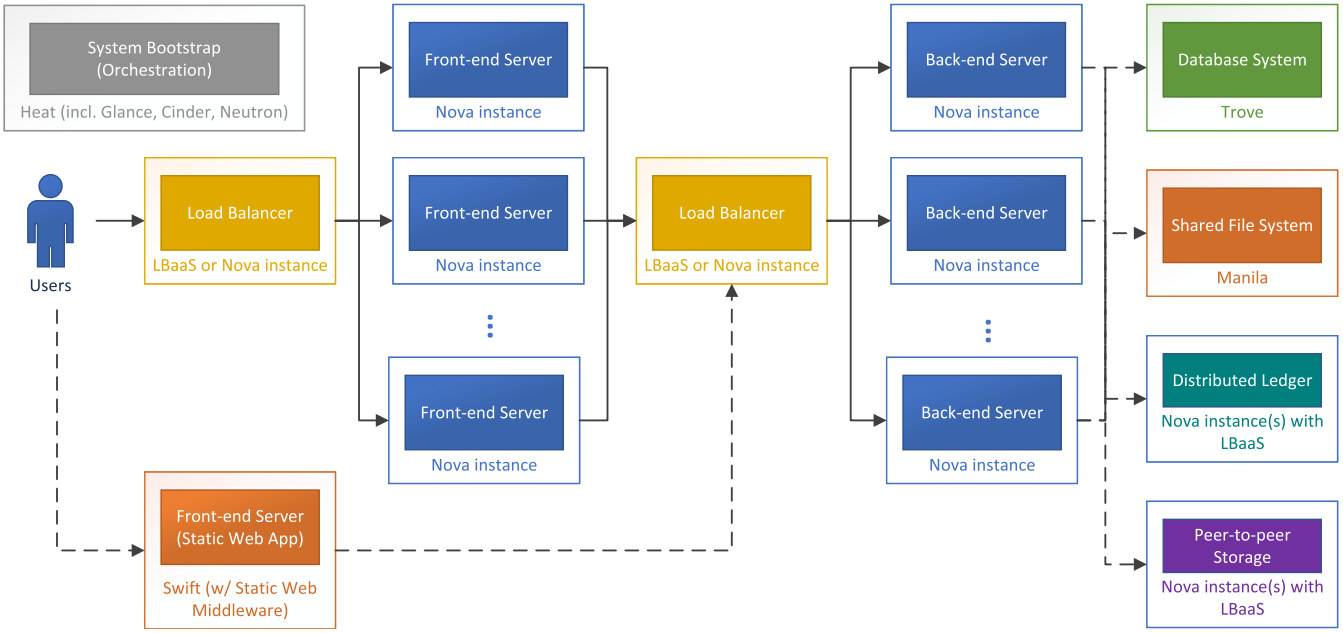


FIGURE 9. The architecture of the Front-End/Back-End architecture for the DII component mapped on OpenStack cloud solution.

installation of the dependencies, gets the latest version from the repository and launches the application. The HOT YAML file also enables the configuration of the load balancing service (LBaaS).

The cloud mapping of the DII component's Front-End/Back-End architecture for the OpenStack cloud is presented in Figure 9.

For quick prototyping, as can be seen from the figure, an alternative is to use the Swift service with the Static Web Middleware, if the API communication between the client and the back-end is done directly, without passing through the front-end servers.

In regard to the service architecture of the DII component from Figure 6, the main differences, compared to the front-end/back-end architecture, consist of the absence of the extra layer pertaining to the front-end, and the conversion of the back-end to services using application servers. In terms of cloud deployment, there are no more front-end server instances, and the back-end server instances are converted to service instances. The latter can provide basically the same functionality as the back-end server instances, or, more interestingly, they ca provide multiple micro-services that can scale independently. Given the lack of proper support for PaaS on OpenStack, the solution could be using the Zun and Magnum services and create containers for each micro-service, similarly to the solution proposed for the decentralized storage architecture in cases B and D. For the other three cloud providers, the services that can be employed for serverless compute are AWS Lambda, Google Cloud Functions and Azure Functions.

Lastly, for the blockchain architecture of the DII component from Figure 7, a cloud solution wraps around the smart contracts and the blockchain network. In this case, the blockchain nodes are represented by the cloud instances. Another way is to deploy a private blockchain on instance nodes belonging to different entities from potentially different cloud providers.

Using a cloud platform is straightforward for the gateway/blockchain architecture of the DII component from Figure 8. The decentralized gateways layer can easily be deployed to a cloud solution by employing multiple compute instances, or by using an API Gateway service like the one provided by AWS. We can even go one step further by having API gateways for multiple cloud installations that act as a single cloud (e.g. by using OpenStack's Trio2o service).

To conclude the cloud mapping discussion, all components of the proposed DII architectures can be deployed to cloud solutions. Their efficiency and ease of integration depends significantly on the chosen cloud provider, selected resources, correct mappings and on particularities of the problem being solved. Existing commercial cloud solutions have the advantage of providing more flexibility in terms of the services that are available at the PaaS tier, but the incurring costs can increase significantly as the system scales up. On the other hand, if there is a hardware infrastructure available, a deployment of the open-source OpenStack platform can prove to be more cost-effective albeit having less support for PaaS. In the worst-case scenario, any component can be deployed on an instance with sufficient compute, memory and storage resources. Also, a hybrid approach can be taken by adding an extra layer of abstraction and by using resources and services from different cloud providers.

V. DISCUSSION

Table 1 presents a comparison of different characteristics between all architectures of the DII component presented

in Section IV, accounting for different types of the Storage Layer.

All presented architectures and storage types are decentralized. This is a critical requirement for an Identity Infrastructure system that must exhibit among others scalability, availability, high levels of security and resilience. However, as can be seen from Table 1, not all architectures feature the same characteristics.

A. TRUST AND SECURITY

An Identity Infrastructure system must be built with the concept of trust at its core and must be able to guarantee trustless interactions among participants. This means that the participants do not need to trust each other or a third party and still be assured that their interactions are performed correctly. To achieve this goal, the Identity System must provide tamper-proof, or at least tamper-evident, guarantees. These characteristics protect the participants against fraudulent interactions, as they can always successfully dispute them. Out of the ten possible decentralized architectures, only four meet this criterion: Front-end/Back-end @Distributed ledger, Service @Distributed ledger, Blockchain, Gateway/Blockchain. All four of them are based on distributed ledger technologies, being either general distributed ledger implementations or blockchain implementations. However, of these four architectures, two of them (Front-end/Back-end @Distributed ledger and Service @Distributed ledger) have characteristics that might not be available in certain implementations and are marked in the table with ''DLT dependent''. These characteristics are immutable storage, tamper-proof storage and traceability. Immutable storage guarantees that once a transaction is recorded it can never be deleted or changed (i.e., persists forever); tamper-proof storage guarantees the impossibility for someone to alter an already recorded transaction or to inject fraudulent transactions; traceability means that all valid interactions are recorded, and it can be proved that they unfolded exactly as stored – this serves as base for audits and non-repudiation. For the highest degree of security and accountability, the system must also feature these characteristics. Therefore, if a @Distributed ledger architecture is chosen, it should be checked that the DLT implementation supports these characteristics. The other 2 architectures (Blockchain and Gateway/Blockchain) feature all these characteristics out-of-the-box. Therefore, they can reliably be used as the basis for a highly trusted Identity Infrastructure.

Asymmetric cryptography [53], mechanisms similar to Certificate Revocation Lists (CRLs) [54], multi-signature credentials, data encryption at rest, Zero-Knowledge Proofs (ZKP) [55] are some of the building blocks that stand at the base of a decentralized self-sovereign digital identity system. Together will user-controlled crypto wallets they provide security guarantees for both data protection and privacy. In case a user wallet is stolen (e.g. the smartphone that holds the wallet), it cannot be deciphered without the user key or without unlocking the wallet. In the unlikely event that the wallet is unlocked, only that user data is leaked with no repercussions on the rest of the system (the other users, issuers and validators). If an issuer or verifier key is compromised, it can be immediately enrolled in a revocation list that invalidates that key for future uses. Another way to address this would be to issue multi-signature credentials, so if one key is stolen, it cannot be used without other signatory keys. As for tampering, asymmetric cryptography guarantees that tampering is not possible.

B. TRANSPARENCY

Traditional approaches like centralized and federated identity systems store user data in private data silos (on company premises, data centers, cloud etc.). The end user has no information about how this data is stored, if it is encrypted or not, what is the security level of the encryption algorithms and whether they are resistant to cryptanalysis, what processes access the data and with what purpose, what other security measures are deployed to protect the data, and so on. Because of this opacity, as stated in Section I, massive data breaches that expose PII occur regularly. Therefore, these systems fail to provide the required guarantees to the user privacy that a truly digital society needs. On the other hand, an open-for-review system, where any security researcher can analyze and expose its vulnerabilities, will be much more secure and the impact and frequency of data breaches would be greatly minimized.

The new approach to identity management described in this paper puts the user in full control over his own data. The data is stored in cryptographically secure wallets controlled by users alone.

Many crypto wallets are released as open source projects (e.g. Metamask [56], [57], [58], Taho [59], [60]) and anyone can fork or contribute to these wallets to further expand their capabilities. Moreover, many of these wallets rely on advanced cryptographic algorithms that were proven to provide high levels of security (e.g. Elliptic Curve Digital Signature Algorithm (ECDSA) [61], also used in Bitcoin and Ethereum blockchains).

Besides open source crypto wallet implementations, a wide range of open source libraries (e.g. Web3.js [62], [63], Ethers [64], [65]) are available that can be used to develop crypto wallets on a variety of operating systems and hardware devices.

Although crypto wallets are usually designed to operate on a blockchain network, this is not the only context in which they can be used. For instance, in the context of DID and SSI, they can operate independent of a blockchain network and in fact, the blockchain network can be replaced with other decentralized architectures as discussed in Subsection IV-C and Subsection IV-E.

One of the benefits of this new approach to identity management is that the companies no longer have to allocate resources to implement different rules and regulations enforced by authorities, regarding user data, and can reallocate these resources to provide better services.

TABLE 1. Characteristics of DII component architectures and storage types.

DII Architecture @Storage type	Dedicated Entry layer	Dedicated Storage layer	Append-only storage	Immutable storage	Tamper-evident storage	Tamper-proof storage	Data replicated on all nodes	Traceability
Front-end/Back-end @Distributed database	✓	✓						
Front-end/Back-end @Distributed ledger	✓	✓	✓	DLT dependent	✓	DLT dependent	✓	DLT dependent
Front-end/Back-end @Distributed filesystem	✓	✓						
Front-end/Back-end @Peer-to-peer storage	✓	✓						
Service @Distributed database		✓						
Service @Distributed ledger		✓	✓	DLT dependent	✓	DLT dependent	✓	DLT dependent
Service @Distributed filesystem		✓						
Service @Peer-to-peer storage		✓						
Blockchain			✓	✓	✓	✓	✓	✓
Gateway/Blockchain	✓		✓	✓	✓	✓	✓	✓

C. DECENTRALIZATION AND SCALABILITY

As mentioned in Subsection IV-D, the scalability and resilience aspects of the platform are very important. The general architecture of the decentralized identity system, discussed in Subsection IV-C and presented in Figure 2, combined with the decentralized identity infrastructures presented and discussed in Subsection IV-E, provide the overall blueprint for the entire system that is scalable, secure, privacy-preserving and trusted by its participants.

One of the key aspects that promotes scalability is a complete decentralization across the system. Although decentralization does not imply scalability, in the proposed system, having different decentralized identity management use-cases with different entities involved in separate transactions and different issuer and validator hierarchies, helps achieving scalability. All of the system components identified in the general architecture together with the DII component exhibit decentralization: Users and Digital Wallets are inherently decentralized; Issuers and Verifiers/Service Providers are decentralized since they are operated by different entities in different networks with different resource capabilities; Identity Infrastructure, implemented by following the proposed architectures presented in Subsection IV-E, that are different implementations of the technology agnostic blueprint presented in Figure 3, also exhibits multiple levels of decentralization.

Decentralization also eliminates single points of failure, enabling the system to provide yet another important property to its users: operational resilience.

D. ARCHITECTURE

Building a scalable, secure and trusted decentralized identity system that brings together and mediates the interactions between a large number of different entities is difficult to implement. This paper aims to provide insights about various architectures of such systems, together with their different characteristics.

There is no "best" architecture that fits all situations for implementing a decentralized identity system, thus one should weigh carefully when deciding what architecture to choose from based on particular needs, available resources, existent systems and foreseen developments. This is why this paper proposes several architectures and lists and compares their features (see Table 1) so that one can choose the right combination of characteristics supported by a particular architecture. As an example, suppose that a private consortium already runs a series of datacenters around the world and they already have storage support in the form of DLT. Also, suppose that they want to implement a decentralized identity solution across their systems, while taking advantage of existent infrastructure and storage solution. In this particular case, the Service @Distributed ledger architecture would be the best combination for the Decentralized Identity Infrastructure and storage solution.

Once a particular DII architecture/storage solution is chosen, the entire system can be built by following the general blueprint presented in Figure 2.

E. INTEROPERABILITY

Independent of the chosen architecture, an orthogonal aspect of the system is interoperability. This is a very important aspect that may have a big impact on the system adoption and on the success of the project. Therefore, the implementation should follow the guidelines set by international standards, like W3C for Decentralized Identifiers [22] and Verifiable

Credentials [23]. Interoperability with other existing systems should also be considered, as it is more likely to quickly grow the user base and to develop and expose new services to different entities.

F. UTILITY, USABILITY AND INTEGRATION

One of the biggest drawbacks in the development and success of any system is its potential for user adoption. No matter how good a system is, from the technical point of view, if it does not appeal to its target audience, it will eventually fail.

There are two main aspects that control user adoption, that must be addressed together: utility and usability. In terms of utility, any services that provide added value to the users should be considered to be implemented. In terms of usability, simple and ergonomic ways to interact with the system should be developed (e.g. mobile and desktop applications, plugins/extensions for already established applications, integration and compatibility with generic applications and hardware devices etc.).

A third aspect, that does not directly relate to the target audience, is ease of integration. This opens the possibility for third party applications and systems to integrate with the current system so that the entire ecosystem can grow. This in turn generates a win-win situation for both the cooperating systems and their users.

G. MAINTAINABILITY

Designing loosely-coupled components and abstracting their roles is industry best practice. This is why, any particular implementation of the components presented in Figure 2 and Figure 3 can be replaced and/or modified without triggering any modification across the other components, as long as the interface API contract is maintained.

H. LIMITATIONS

To be able to interact with a digital identity management system, one would need access to a smart device. Fortunately, a large segment of today's society has access to devices such as smartphones, tablets, laptops and ultrabooks. According to Statista [66], more than 85% of world population have access to a smartphone, with even higher rates in North America and Europe, and this is just for smartphones alone, not including the other devices. Therefore, the means and the technical capabilities for society to access digital identity management systems are already available. However, this is just part of the problem and society still has to embrace the paradigm shift addressed by the new digital identity managements systems. This is a coordinated effort between multiple entities (local and central authorities, public and private companies, NGOs) that have the role to implement and present the benefits that these new systems bring to the entire society. Ways to promote these benefits with a long-lasting impact are education, video showcases promoted by authorities on various media channels (e.g. in public transportation vehicles and stations, outdoor advertising panels), influencers, tech talks hosted by universities etc. Incentives also play an important role in the adoption of these systems. For instance, a smart city/smart mobility application can grant discounts to early adopters. Once a critical mass is reached and people realize the full potential of these new technologies, an entire ecosystem can be developed to shape the future digital society.

VI. CONCLUSION

The current approaches to digital identity and proof of identity used by centralized and federated identity management systems exhibit security problems, massive data breaches, lack of privacy and lack of user control. To address these problems, this paper discusses how emerging Web3 technologies can be used to reshape the identity concept and proof of identity mechanisms and to give users full control over their digital identity. To this end, a general architecture for a decentralized digital identity management system based on Decentralized Identifiers and Self-Sovereign Identities was presented. In the framework of this architecture, where the user is placed at the center of the system, the Decentralized Identity Infrastructure component plays a key role in providing and mediating trust across the entire system by enabling trustless interactions among participants.

A system that deals with private and personally identifiable information must be based on an architecture that allows it to guarantee data protection and user privacy and to make data breaches impossible. The main novelty consists of the proposed different architectures for the Decentralized Identity Infrastructure component along with different approaches for the Storage Layer implementations. All these architectures were compared, and a series of characteristics were identified as being mandatory. For the decentralized storage architecture, four solutions were identified: distributed database, distributed ledger, distributed file system and peer-to-peer storage. In terms of the Decentralized Identity Infrastructure component, the proposed architectures were: Front-end/Back-end with storage, Service with storage, Blockchain, and Gateway/Blockchain.

As mentioned on the previous sections, there is no ''best'' distributed architecture that fits all situations and this is why several architectures were introduced and analyzed in the context of the overall blueprint of a decentralized identity management system (see Figure 2). Based on particular needs and constraints, one can choose the right combination of characteristics and select a particular architecture, guided by the information presented in Table 1.

This paper also discusses how the proposed architectures can be mapped on a cloud solution, with focus on the Front-end/Back-end architecture of DII component. The services that can be employed from four major cloud solutions (i.e., OpenStack, Amazon Web Services, Google Cloud Platform and Microsoft Azure) were identified, and a potential mapping solution for OpenStack, in terms of deploying the Decentralized Identity Infrastructure component using the appropriate services, was provided.

The proposed architectures can be used to build identity systems that are scalable, ensure privacy, trust, transparency and traceability, while allowing users to maintain full control over their own identity and whom they share it with, paving the way for our future digital societies.

AUTHOR CONTRIBUTIONS

Conceptualization, Cristian Nicolae Butincu; methodology, Cristian Nicolae Butincu and Adrian Alexandrescu; validation, Cristian Nicolae Butincu and Adrian Alexandrescu; formal analysis, Cristian Nicolae Butincu and Adrian Alexandrescu; investigation, Cristian Nicolae Butincu and Adrian Alexandrescu; writing—original draft preparation, Cristian Nicolae Butincu; writing—review and editing, Cristian Nicolae Butincu and Adrian Alexandrescu; and supervision, Cristian Nicolae Butincu. All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] *General Data Protection Regulation (GDPR) Compliance Guidelines. GDPR.eu.* Accessed: Dec. 7, 2023. [Online]. Available: https://gdpr.eu/

[2] A. Alexandrescu and G. Butnaru, "An architecture of identity management and thirdparty integration for online teaching in a university," in *Proc. 24th Int. Conf. Syst. Theory, Control Comput. (ICSTCC)*, Sinaia, Romania, Oct. 2020, pp. 850–855, doi: 10.1109/ICSTCC50638.2020.9259652.

[3] N. Perlroth. (Oct. 2017). *All 3 Billion Yahoo Accounts Were Affected By 2013 Attack. The New York Times*. Accessed: Dec. 7, 2023. [Online]. Available: https://www.nytimes.com/2017/10/03/technology/yahoo-hack-3-billion-users.html

[4] *885 Million Records Exposed Online: Bank Transactions, Social Security Numbers, and More. Gizmodo*. Accessed: Dec. 7, 2023. [Online]. Available: https://gizmodo.com/885-million-sensitive-records-leaked-online-bank-trans-1835016235

[5] (2022). *Roth, E. Meta Fined $276 Million Over Facebook Data Leak Involving More Than 533 Million Users. The Verge*. Accessed: Dec. 7, 2023. [Online]. Available: https://www.theverge.com/2022/11/28/23481786/meta-fine-facebook-data-leak-ireland-dpc-gdpr

[6] (2022). *Meta Settles Cambridge Analytica Scandal Case for $725m. BBC News*. Accessed: Dec. 7, 2023. [Online]. Available: https://www.bbc.com/news/technology-64075067

[7] *Hackers Leak LinkedIn 700 Million Data Scrape*. Accessed: Dec. 7, 2023. [Online]. Available: https://therecord.media/hackers-leak-linkedin-700-million-data-scrape

[8] J. S.-G. Perlroth. (2014). *Matthew Goldstein and Nicole. JPMorgan Chase Hacking Affects 76 Million Households. DealBook*. Accessed: Dec. 7, 2023. [Online]. Available: https://dealbook.nytimes.com/2014/10/02/jpmorgan-discovers-further-cyber-security-issues/

[9] N. Lomas. (2020). *U.K. Watchdog Reduces Marriott Data Breach Fine To $23.8M, Down From $123M. TechCrunch*. Accessed: Dec. 7, 2023. [Online]. Available: https://techcrunch.com/2020/10/30/uk-watchdog-reduces-marriott-data-breach-fine-to-23-8m-down-from-123m/

[10] (Oct. 30, 2013). *Adobe Hack: At Least 38 Million Accounts Breached. BBC News*. Accessed: Dec. 7, 2023. [Online]. Available: https://www.bbc.com/news/technology-24740873

[11] (2014). *Hackers Raid EBay in Historic Breach, Access 145M Records. CNBC*. Accessed: Dec. 7, 2023. [Online]. Available: https://www.cnbc.com/2014/05/22/hackers-raid-ebay-in-historic-breach-access-145-mln-records.html

[12] *Why Are Companies Failing At Data Protection? | 2021-08-18 | Security Magazine*. Accessed: Feb. 9, 2024. [Online]. Available: https://www.securitymagazine.com/articles/95893-why-are-companies-failing-at-data-protection

[13] *Rising To the Challenge of Modern Data Security and Growing Privacy Regulations | Security Magazine*. Accessed: Feb. 9, 2024. [Online]. Available: https://www.securitymagazine.com/articles/98176-rising-to-the-challenge-of-modern-data-security-and-growing-privacy-regulations

[14] *Gaivenyte, E. Study: Companies Neglect Client Data Security. NordPass.* Accessed: Feb. 9, 2024. [Online]. Available: https://nordpass.com/blog/companies-data-breach-study/

[15] (2022). *Data Transparency's Essential Role in Building Customer Trust. Cisco 2022 Consumer Privacy Survey*. [Online]. Available: https://www.cisco.com/c/dam/en_us/about/doing_business/trust-center/docs/cisco-consumer-privacy-survey-2022.pdf

[16] M. Komnenic. (2024). *64 Alarming Data Privacy Statistics Businesses Must See in 2024. Termly*. Accessed: Feb. 9, 2024. [Online]. Available: https://termly.io/resources/articles/data-privacy-statistics/

[17] *The Dark Side of Data Collection: How Companies Can Build Transparency and Accountability | Namasys Analytics | LinkedIn*. Accessed: Feb. 9, 2024. [Online]. Available: https://www.linkedin.com/pulse/dark-side-data-collection-how-companies-can-build-transparency/

[18] B. A. Turner. (2019). *Lee Rainie, Monica Anderson, Andrew Perrin, Madhu Kumar and Erica. Americans and Privacy: Concerned, Confused and Feeling Lack of Control Over Their Personal Information. Pew Research Center: Internet, Science & Tech*. Accessed: Feb. 9, 2024. [Online]. Available: https://www.pewresearch.org/internet/2019/11/15/americans-and-privacy-concerned-confused-and-feeling-lack-of-control-over-their-personal-information/

[19] K. L. Walker, "Surrendering information through the looking glass: Transparency, trust, and protection," *J. Public Policy Marketing*, vol. 35, no. 1, pp. 144–158, Apr. 2016, doi: 10.1509/jppm.15.020.

[20] K. Zhang and H.-A. Jacobsen, "Towards dependable, scalable, and pervasive distributed ledgers with blockchains," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2018, pp. 1337–1346, doi: 10.1109/ICDCS.2018.00134.

[21] A. Gorkhali, L. Li, and A. Shrestha, "Blockchain: A literature review," *J. Manage. Analytics*, vol. 7, no. 3, pp. 321–343, Jul. 2020, doi: 10.1080/23270012.2020.1801529.

[22] *Decentralized Identifiers (DIDs) V1.0.* Accessed: Dec. 7, 2023. [Online]. Available: https://www.w3.org/TR/did-core/

[23] *Verifiable Credentials Data Model V1.1.* Accessed: Dec. 7, 2023. [Online]. Available: https://www.w3.org/TR/vc-data-model/

[24] *Home-EBSI*. Accessed: Dec. 7, 2023. [Online]. Available: https://ec.europa.eu/digital-building-blocks/wikis/display/EBSI/Home

[25] *EID*. Accessed: Dec. 7, 2023. [Online]. Available: https://ec.europa.eu/digital-building-blocks/sites/display/DIGITAL/eID

[26] *BankID—Fast and Secure Digital Identification and Signing*. Accessed: Dec. 7, 2023. [Online]. Available: https://www.bankid.com/en

[27] *Faster BankID*. Accessed: Dec. 7, 2023. [Online]. Available: https://www.bankid.no/en

[28] *ESSIF Playground—ESSIF Playground*. Accessed: Dec. 7, 2023. [Online]. Available: https://docs.essif.sk/

[29] *Home ESSIF-Lab*. Accessed: Dec. 7, 2023. [Online]. Available: https://essif-lab.eu/

[30] *Digitale Technologien Projekte Wettbewerbsphase. Projekte Wettbewerbsphase*. Accessed: Dec. 7, 2023. [Online]. Available: https://www.digitale-technologien.de/DT/Redaktion/DE/Standardartikel/SchaufensterSichereDigIdentProjekte/sdi-projekt_ssi.html

[31] *IDunion Ermglicht Selbstbestimmte Identten*. Accessed: Dec. 7, 2023. [Online]. Available: https://idunion.org/

[32] *Lissi Building Trusted Relationships*. Accessed: Dec. 7, 2023. [Online]. Available: https://www.lissi.id/

[33] *Myego EN Decentralized Identity Management.* Accessed: Dec. 7, 2023. [Online]. Available: https://myego.io/

[34] *BMI Homepage of the Online Access Act*. Accessed: Dec. 7, 2023. [Online]. Available: https://www.onlinezugangsgesetz.de/Webs/OZG/EN/home/home-node.html

[35] Z. A. Lux, D. Thatmann, S. Zickau, and F. Beierle, "Distributed-ledger-based authentication with decentralized identifiers and verifiable credentials," in *Proc. 2nd Conf. Blockchain Res. Appl. Innov. Netw. Services (BRAINS)*, Sep. 2020, pp. 71–78.

[36] X. Li, T. Jing, R. Li, H. Li, X. Wang, and D. Shen, "BDRA: Blockchain and decentralized identifiers assisted secure registration and authentication for VANETs," *IEEE Internet Things J.*, vol. 10, no. 14, pp. 12140–12155, Jul. 2023.

[37] A. M. Alnour and K. H. Kim, "Decentralized identifiers (DIDs)-based authentication scheme for smart health care system," in *Proc. 13th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, Jul. 2022, p. 443.

[38] M. T, K. Makkithaya, and N. V G, "A blockchain based decentralized identifiers for entity authentication in electronic health records," *Cogent Eng.*, vol. 9, no. 1, Dec. 2022, Art. no. 2035134.
[39] X. Zhao, B. Zhong, and Z. Cui, "Design of a decentralized identifier-based authentication and access control model for smart homes," *Electronics*, vol. 12, no. 15, p. 3334, Aug. 2023.
[40] P. N. Mahalle and G. R. Shinde, "OAuth-based authorization and delegation in smart home for the elderly using decentralized identifiers and verifiable credentials," in *Security Issues and Privacy Threats in Smart Ubiquitous Computing*. Singapore: Springer, 2021, pp. 95–109.
[41] M. T. Hammi, B. Hammi, P. Bellot, and A. Serhrouchni, "Bubbles of trust: A decentralized blockchain-based authentication system for IoT," *Comput. Secur.*, vol. 78, pp. 126–142, Sep. 2018.
[42] Y. Kang, J. Cho, and Y. B. Park, "An empirical study of a trustworthy cloud common data model using decentralized identifiers," *Appl. Sci.*, vol. 11, no. 19, p. 8984, Sep. 2021.
[43] S. P. Otta and S. Panda, "Decentralized identity and access management of cloud for security as a service," in *Proc. 14th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2022, pp. 299–303.
[44] N. Anjum, D. Karamshuk, M. Shikh-Bahaei, and N. Sastry, "Survey on peer-assisted content delivery networks," *Comput. Netw.*, vol. 116, pp. 79–95, Apr. 2017, doi: 10.1016/j.comnet.2017.02.008.
[45] *Cloudflare—The Web Performance & Security Company | Cloudflare*. Accessed: Dec. 7, 2023. [Online]. Available: https://www.cloudflare.com/
[46] *Web3 Development Platform | IPFS API & Gateway | Blockchain Node Service*. Accessed: Dec. 7, 2023. [Online]. Available: https://www.infura.io
[47] *Open Source Cloud Computing Infrastructure OpenStack*. Accessed: Dec. 7, 2023. [Online]. Available: https://www.openstack.org/
[48] *Cloud Computing Services Amazon Web Services (AWS). Amazon Web Services, Inc*. Accessed: Dec. 7, 2023. [Online]. Available: https://aws.amazon.com/
[49] *Cloud Computing Services | Google Cloud*. Accessed: Dec. 7, 2023. [Online]. Available: https://cloud.google.com/
[50] *Cloud Computing Services | Microsoft Azure*. Accessed: Dec. 7, 2023. [Online]. Available: https://azure.microsoft.com/en-us/
[51] A. Alexandrescu and C. Mironeanu, "A framework for anything-as-a-service on a cloud platform," in *Proc. 27th Int. Conf. Syst. Theory, Control Comput. (ICSTCC)*, Timisoara, Romania, Oct. 2023, pp. 333–338, doi: 10.1109/icstcc59206.2023.10308434.
[52] A. Alexandrescu and C. N. Butincu, "Decentralized news-retrieval architecture using blockchain technology," *Mathematics*, vol. 11, no. 21, p. 4542, Nov. 2023, doi: 10.3390/math11214542.
[53] *What is Asymmetric Cryptography? Definition From SearchSecurity. Security*. Accessed: Feb. 10, 2024. [Online]. Available: https://www.techtarget.com/searchsecurity/definition/asymmetric-cryptography
[54] *What Is a Certificate Revocation List (CRL) and How Is It Used? Security*. Accessed: Feb. 10, 2024. [Online]. Available: https://www.techtarget.com/searchsecurity/definition/Certificate-Revocation-List
[55] *Zero-Knowledge Proof (ZKP)—Explained | Chainlink*. Accessed: Feb. 10, 2024. [Online]. Available: https://chain.link/education/zero-knowledge-proof-zkp
[56] *The Ultimate Crypto Wallet for DeFi, Web3 Apps, and NFTs | MetaMask*. Accessed: Feb. 9, 2024. [Online]. Available: https://metamask.io/
[57] (2024). *MetaMask/Metamask-Extension*. Accessed: Feb. 9, 2024. [Online]. Available: https://github.com/MetaMask/metamask-extension
[58] (2024). *MetaMask/Metamask-Mobile*. Accessed: Feb. 9, 2024. [Online]. Available: https://github.com/MetaMask/metamask-mobile
[59] *Taho—The Community Owned & Operated Wallet*. Accessed: Feb. 9, 2024. [Online]. Available: https://taho.xyz/
[60] (2024). *Tahowallet/Extension*. Accessed: Feb. 9, 2024. [Online]. Available: https://github.com/tahowallet/extension
[61] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *Int. J. Inf. Secur.*, vol. 1, no. 1, pp. 36–63, Aug. 2001, doi: 10.1007/s102070100002.
[62] *Web3.JS—Javascript Ethereum API*. Accessed: Feb. 9, 2024. [Online]. Available: https://web3js.org/
[63] (2024). *Web3/Web3.Js*. Accessed: Feb. 9, 2024. [Online]. Available: https://github.com/web3/web3.js
[64] *Ethers*. Accessed: Feb. 9, 2024. [Online]. Available: https://ethers.org
[65] (2024). *Ethers-Io/Ethers.Js*. Accessed: Feb. 9, 2024. [Online]. Available: https://github.com/ethers-io/ethers.js
[66] (2023). *Mobile Network Subscriptions Worldwide 2028. Statista*. Accessed: Feb. 10, 2024. [Online]. Available: https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/ .

CRISTIAN NICOLAE BUTINCU received the B.S. degree in computer science and engineering, the M.S. degree in distributed systems, and the Ph.D. degree in computer science from Gheorghe Asachi Technical University of Iaşi, Romania, in 2001, 2002, and 2006, respectively.

Since 2013, he has been an Assistant Professor with the Department of Computer Science and Engineering, Faculty of Automatic Control and Computer Engineering, Gheorghe Asachi Technical University of Iaşi. He has published more than 20 scientific articles, coauthored two books, and was involved in nine national and international research projects. He was the Principal Investigator of the Fighting disinformation using decentralized actors featuring AI and blockchain technologies (FiDisD). The project was developed in the context of Next Generation Internet (NGI) Trusted and Reliable Content on Future Blockchains (TruBlo), part of European Commission's NGI Initiative. His research interests include parallel and distributed computing, cluster and cloud computing, high-performance computing, cybersecurity, self-healing systems, dynamic systems, mobile agent platforms, evolutionary algorithms, artificial intelligence, big data, databases and indexing technologies, and blockchain technologies.

ADRIAN ALEXANDRESCU received the B.S. degree in systems and computer engineering, the M.S. degree in distributed systems, and the Ph.D. degree in computers and information technology from Gheorghe Asachi Technical University of Iaşi, Romania, in 2008, 2009, and 2012, respectively.

Since 2014, he has been an Assistant Professor with the Department of Computer Science and Engineering, Faculty of Automatic Control and Computer Engineering, Gheorghe Asachi Technical University of Iaşi. Since 2019, he has been a Digital Academic Services Coordinator in informatization and digital communications directorate with Gheorghe Asachi Technical University of Iaşi. He was a member of 18 academic projects, out of which six were research projects. He is currently the author of more than 35 articles. His research interests include distributed and decentralized architectures, cloud computing, high-performance computing, the Internet of Things, e-learning, cybersecurity, information retrieval, recommendation systems, task mapping and scheduling, and genetic algorithms.

Dr. Alexandrescu was a winner of the first (2018) and sixth (2023) editions of the ANIS Scholarships, an initiative of the tech industry to stimulate the teaching skills of young university teachers, in the security field. He was also a winner of the Fulbright Project "Cybersecurity in Universities-Study Visits to U.S.," in 2021.

A Secure Real Estate Transaction Framework Based on Blockchain Technology and Dynamic Non-Fungible Tokens

Delia-Elena Bărbuță*, Adrian Alexandrescu†
Faculty of Automatic Control and Computer Engineering,
Gheorghe Asachi Technical University of Iaşi
Iași, Romania
* delia-elena.barbuta@academic.tuiasi.ro
† adrian.alexandrescu@academic.tuiasi.ro, aalexandrescu@tuiasi.ro

2024 28th International Conference on System Theory, Control and Computing (ICSTCC) | 979-8-3503-6429-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICSTCC62912.2024.10744685

***Abstract*—The real estate market, as an engine of economic growth, directly affects both financial stability and the development of communities at local and global levels. Nevertheless, it faces significant challenges, ranging from complex legislative processes and bureaucratic inefficiencies to fraud risks and high costs. Addressing these issues requires a paradigm shift to improve efficiency and transparency in this sector. The proposed solution aims to integrate emerging technologies such as blockchain and smart contracts to secure transactions and create a tamper-proof digital history. The implementation of dynamic non-fungible tokens (dNFTs) facilitates property transfers and digitally models real estate assets. A novel communication protocol and store policy are proposed for storing real estate information on the blockchain network and on decentralized storage so that the digital token representing the property can change based on external factors, e.g., change in ownership, actual updates being done on the property, or an update in valuation. The proposed framework ensures trustworthiness, transparency, traceability and security, while providing a flexible architecture with the main system components, including ways to model, update, list and buy a real estate property.**

***Index Terms*—blockchain, dynamic non-fungible tokens, dNFT, smart contracts, tokenized real estate**

I. Introduction

The real estate market is a crucial pillar of the economy, wielding considerable influence over financial activities on a global scale. With its significant economic value, the real estate market not only mirrors financial trends but also exerts a strong impact on economic growth prospects and development.

According to the Real Estate Association, the real estate market's contribution to Gross Domestic Product (GDP) in 2019 was 7.62%. In a broader context, considering that real estate also serves as a major source of investment, its total contribution to GDP can reach a remarkable 13.6% [1]. In emerging countries, the market's contribution to GDP in 2021 ranged from 6.9% to 18.5%, with an average of 13.1% in countries that had available data [2].

The high volume of real estate transactions also serves as a significant indicator of this market's vitality. It's not only a market that reflects the essential need for housing but also a source of sustained economic growth. Regarding investments, real estate often forms a substantial part of investor portfolios [3], thus contributing to national economic stability. Additionally, beyond the direct value it brings to the economy, the real estate market significantly impacts other related sectors, like construction.

However, this very dynamism presents challenges. The real estate market is increasingly complex, and a series of obstacles often diminish the efficiency and transparency of transactions. One of the main concerns is the increased risk of fraud and the opacity of information in the property acquisition process. This problem becomes even more pronounced and complex when exploring property acquisitions internationally.

The process of buying a property in a foreign country unravels new levels of difficulty and uncertainty [4]. The diverse laws, procedures, and cultural norms specific to each country become significant obstacles to an efficient acquisition process. Information on property titles, transaction history, and the real value of properties can become inaccessible or difficult to verify, creating a climate conducive to unethical practices and, implicitly, to the risk of fraud.

In addition to that, the large number of intermediaries that are involved in a transaction represents another major problem, adding complexity and additional costs to the process [5]. These intermediaries can include real estate agents, lawyers, notaries, banks, and other appraisal or lending entities, each with their own fees and commissions. This leads to a significant increase in the total transaction cost for buyers and sellers, affecting the financial accessibility and efficiency of the property acquisition process.

For example, an escrow agent in traditional real estate transactions acts as a guarantor of the fulfillment of mutual obligations. In the context of digitization and technological advancement, it is possible that the role of such agents could be taken over by programmatic solutions, such as digital protocols or emerging technology, thus contributing to the streamlining of the process and cost reduction [6].

Taking all these aspects into consideration, the need for a paradigm change becomes apparent, in order to simplify the process for all stakeholders. The aim of this paper is to

introduce a framework designed to be transparent enough to prevent fraud, while also properly securing real estate property information and documents, to avoid unauthorized access. This is achieved by storing encrypted documents on a decentralized platform, recording all transactions on the blockchain, and modeling real estate asset transfers through dynamic Non-Fungible Tokens (dNFTs).

The novelty of the research presented in this paper consists of leveraging dNFTs for real estate transaction, proposing a novel architecture to handle the asset management, and providing a framework for a trustworthy property listing, transfer and information updates.

The remainder of the paper is structured as follows. Section II presents the related work regarding real estate transaction using NFTs. The architecture and design of the proposed solution are described in Section III. A practical use case scenario is discussed in Section IV, and the conclusions regarding the efficacy of the proposed system are presented in Section V.

II. Related Work

Applications of blockchain technology and NFTs are varied [7] and they extend to domains that require traceability, trust, transparency and security like digital art, virtual world, tracking goods provenance, or even fighting disinformation [8]. The blockchain network is an immutable, secure and transparent distributed ledger. Smart contracts run on the blockchain network, they can interact with blockchain assets (e.g., NFTs) and contain the logic that allows for these assets to be created through minting, and to be transferred or sold. An NFT represents the ownership of a unique item on the blockchain, and it has an associated metadata that describes the asset, e.g., title, description, creator. The problem with NFTs is that their content can never change. On the other hand, dynamic NTFs allow for the metadata to be modified by smart contracts in response to external triggers [9]. A dNFT can be updated manually or automatically based on pre-defined conditions. They can also be created and updated through oracles, which enable access to external data sources. These oracles monitor the data sources and feed the relevant information to smart contracts. The trust in the data provided by an oracle is crucial [10].

An article from 2022 ([11]) mentions the advantages of using NFTs to sell your home and provides real-world examples of homes being sold through NFTs. The main problems, when buying a home the traditional way, are related to a lengthy and costly closing process, because it takes around four to six weeks to finish the documentation, inspection and the incurring costs are between 5%-10% of the home's value.

Using NFTs allows the seller to bypass intermediaries and the blockchain network is used to verify ownership, it ensures trust in the process and it allows the financial transaction. The process of selling a home through NTFs consists of having an LLC (Limited Liability Company) that owns the physical property and an NFT that is linked to the LLC and serves as proof of ownership. The first real estate sold as an NFT was an apartment in Ukraine in 2021, which was paid for using Ethereum. The sale was done through the Propy platform [12]. The issue is that, from a legal point of view, selling the NFT does not mean that the property is actually sold. A transaction must also be entered in the local physical registry. Nonetheless, blockchain technology is finding its way into real-world applications, with a global trend towards practical and legal adoption.

The use dynamic NFTs is still in its early stages and there is little existing research in this area. One of the first articles describing dNFTs and mentioning some applications for them from a conceptual level is [9]. The paper contains just one paragraph about real estate, highlighting the potential of dNFTs in this sector. In an article presenting the evolution of the NFT concept ([13]) the authors emphasize the applicability of dynamic NFTs, while in [14] the authors showcase how assets (characters) in trading games can evolve using Inter-Planetary File System (IPFS) and NFTs. In terms of how the dNFTs actually work, in [15] is proposed an architecture for a dNFT generation system that uses decentralized oracles and smart contracts, which can be employed as a reference protocol by our proposed architecture. Currently, there are no papers that present a framework or an architecture that allows the use of dNFTs in real estate transactions. There are only some references regarding the potential use in this manner.

III. Proposed System

A. Overview

This paper presents a framework for conducting real estate transactions, which is based on blockchain technology. This allows for the creation of a tamper-proof digital history for each property and provides a secure digital format for property ownership documents. Both public (e.g. price of the property) and private (e.g. the social security number of the owner) metadata are defined, that can adapt and evolve to accurately reflect the realities in the physical world.

To bridge the gap between the data from the physical word and the data that is on the blockchain, oracles play a pivotal role, by supplying dynamic NFTs with real-world updates like improvements, renovations, and past sales, ensuring that each token faithfully represents the current state of the property.

One innovation in this protocol is that property owners can choose to change the visibility of specific metadata, giving them control over which details are public or kept private.

Purchase and sale transactions are managed automatically through smart contracts, ensuring that if both the buyer and seller meet the set conditions, such as payment and document upload, the transfer occurs automatically. If either party fails to meet the requirements or behaves fraudulently, the transaction is programmatically canceled.

B. Stakeholders

1) Owners: Property owners are looking to sell their real estate with minimal hassle, quick transactions, and secure payment. Their concerns are getting a fair price for their property, ensuring that the buyer has the financial means to

complete the transaction and protecting themselves against fraud or disputes.

2) Buyers: Buyers need a dependable and transparent method to purchase property that suits their requirements, with minimal intermediaries. Key safeguards include ensuring the authenticity of property documentation and ownership, while also avoiding potential legal complications. In addition to this, providing detailed property information, such as comprehensive property history, also enhances buyer confidence.

3) Fiscal authority: Government tax departments are interested in ensuring property transactions are transparent and that taxes are properly collected. They require accurate tax assessments, based on authentic transaction data and their main focus is in property tax compliance and reducing underreporting.

4) Local authorities: Local governments ensure that property transactions align with the local regulations and land use policies. Properties need to be used in ways that comply with local zoning laws (residential, commercial etc.) and any specific land use restrictions. Accurate property transaction records assist in assessing future demands for public infrastructure like water supply, sewage and transportation networks.

5) Legal professionals: Legal professionals, including real estate lawyers and notaries, play a role in guiding real estate transactions and ensuring compliance with applicable laws. Their primary concerns are focused on verifying document accuracy and compliance, preventing fraud, and resolving disputes. They ensure the documentation is accurate, complete, and aligns with regulations. Legal professionals also help identify potential fraudulent behavior and advise clients on safeguarding against unauthorized transactions and in the event of disputes during transactions they mediate the conversation.

6) Financial institutions: Banks and mortgage companies offer financing to buyers who need loans to complete their property transactions. They require accurate and transparent property data to assess risks. Financial institutions must ensure that all transactions adhere to relevant regulations. They rely on the system to provide accurate property history and transaction data to manage compliance and risk assessment.

C. System Architecture

The system architecture in comprised of multiple components, as shown in Fig.1. Firstly, there is the blockchain network with the associated smart contracts for managing the dNFTs (minting, transferring, updating), property listing, escrow, arbiter and auction, and the oracles that interact with external trustworthy APIs, which provide the required information and data streams for updating the dNFTs. Another major component is the Decentralized File System (i.e., IPFS) where the majority of the real estate information is stored. The core of proposed solution is the platform for managing the real estate transactions, which has a front-end server that represents the user interface, a back-end server that contains the business logic, a database for keeping user and asset information, and a cache solution to increase response speed to various requests. The users interact with the platform using a web portal and perform transactions using their crypto wallet (e.g., with the MetaMask browser plugin).

The back-end platform contains modules that interact with the front-end server by exposing APIs, an Identity Access Management module, a Decentralized File System Handler for managing the information stored on the decentralized storage, a Blockchain Handler for providing a view on the relevant data stored on the blockchain, a Data Store module for handling the database and cache data, an Asset Model module, a Real Estate module for managing the assets, a Security module that provides encryption tools, and, finally, a Data Analysis module for analytics regarding the real estate transactions.

D. Modeling an Asset

To model an asset, we define a set of properties that the dNFT will contain:

1) *Property Description*: Address, type (residential, commercial, etc.), construction year, size, number of rooms.
2) *Maintenance History*: Records of past maintenance activities performed on the property, including repairs, renovations, upgrades, and any other maintenance work.
3) *Market Value*: The current market value of the property, based on factors such as location, size, condition, and recent comparable sales in the area.
4) *Property Features*: Details about specific features or amenities of the property, such as swimming pools, gardens, parking spaces, security systems, and any other notable attributes.
5) *Property History*: A chronological record of significant events related to the property, such as previous owners, sales transactions or zoning changes.
6) *Legal Documentation*: References to critical legal documents, such as the ownership deed, purchase contracts, pre-contracts, and other paperwork that establishes ownership or compliance with local regulations. These documents are securely stored and linked to the dNFT.
7) *Virtual Tour*: A multimedia presentation, such as a video, that provides a visual walk-through of the property, allowing potential buyers to explore its layout, design, and features remotely. Under this section, there can also be a collection of images.
8) *Environmental Factors*: Information about environmental factors that may affect the property, such as flood zones, seismic activity, soil quality, or proximity to natural hazards.
9) *Neighborhood Information*: Information about the surrounding neighborhood or community, including demographics, schools, parks, shopping centers, transportation options, crime rates, and other factors that influence the desirability and quality of life in the area.

E. Asset Evolution

With the exception of the details outlined in the Property Description, all other aspects listed above are expected to undergo periodic updates, with some requiring more frequent

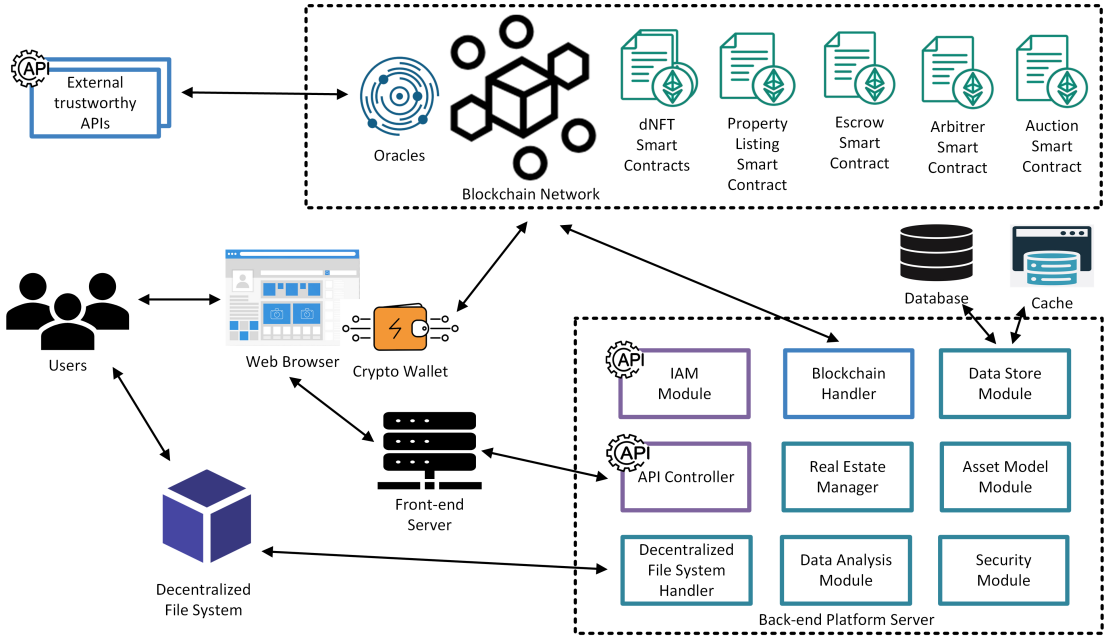


Fig. 1. System architecture for proposed the secure real estate transaction platform

revisions than others. Even the particulars within the property description hold the potential for alteration. Hence, all attributes are conceptualized as dynamic, allowing for the possibility of modifications over time. The owner maintains the authority to determine metadata visibility, whether private or public. This is achieved through smart contracts and oracles as it is explained later in this section.

The idea is to update the asset based on real world changes to the real estate. For example, an annex to a house is added, the current market valuation changes, natural hazard impacted the property, or, simply, the owner wants to add more pictures of the property or want to add a virtual tour presentation.

F. Listing a Property

The property owner is tasked with modeling the asset and ensuring that the relevant property information is accurate. When intending to list the property for sale, the owner must also upload a legal document proving ownership. Once submitted, both the fiscal and local authorities are notified of the owner's intent to sell. In the first iteration, these entities are also responsible for verifying the document's authenticity and the correctness of the metadata. In further implementation stages, this process can be automated.

Provided that the legal verification succeeds, the dNFT is created and can be listed for sale. Sellers can choose between fixed pricing or a bidding system. In a fixed price model, the seller sets a predefined value, while a bidding system allows prospective buyers to make offers, potentially leading to a higher sale price.

G. Buying a Property

- The potential buyer places an offer for the property via the platform, either by submitting a fixed-price bid or by placing a bid in an auction format.
- The seller reviews the offer and accepts it if it meets their requirements.
- The buyer and seller, after having consulted with legal professionals, digitally sign a pre-contract, which outlines the key terms and conditions, including the timeline for the final payment and any conditions for refund.
- Based on the information from the pre-contract, an escrow smart contract and an arbiter smart contract are generated. Another possibility is to skip the creation of the pre-contract altogether and generate the two smart contracts directly through the platform.
- The buyer deposits a predefined sum into the escrow smart contract, which acts as a neutral intermediary
- The seller provides the ownership documents
- During the escrow period, the buyer can conduct due diligence, such as inspections and assessments, to confirm the property's condition and compliance with local regulations. Legal professionals verify the property ownership documents and ensure there are no unresolved issues.
- The final contract is crafted by the legal professionals and signed by both parties.
- The buyer completes the final payment via the escrow smart contract.
- The smart contract releases the payment to the seller and transfers the NFT representing the property to the buyer.
- The ownership transfer is recorded on the blockchain, and the dNFT metadata is updated accordingly. The signed pre-contract and contract are linked to the dNFT and the data of the new owner is added to the history of the property. The fiscal authority is notified.

H. Escrow Smart Contract

The escrow smart contract functions as a secure intermediary between the buyer and the seller. When constructed, the

contract is provided with the addresses of the buyer, seller, and arbiter. The arbiter is a neutral party responsible for resolving disputes and giving final approval for the release of the funds.

In addition to the three addresses, the contract requires the amounts involved in the transaction: the initial deposit, which serves as a show of good faith from the buyer, and the final payment amount that constitutes the total cost of the property. The smart contract also maintains a data structure to track the status of the various contingencies and conditions that must be satisfied before the payment is released.

The escrow contract includes several critical operations. First, the buyer deposits their funds into the smart contract, which emits an event signaling the deposit. Next, the buyer and seller can check the status of the contingencies (via an associated arbiter smart contract) to ensure compliance before the funds are released. Finally, once the arbiter confirms that all the pre-contract conditions are met and the contingencies satisfied, the contract releases the funds to the seller. The smart contract emits events at each important step, such as the completion of the deposit, satisfaction of conditions, and successful transfer of funds to the seller. These events provide a transparent log of the transaction's progress.

I. Arbiter Smart Contract

The arbiter smart contract operates as the decision-making authority that oversees the conditions of the transaction. This contract is often controlled by a trusted legal professional or authority and holds the power to approve or deny specific contingencies outlined in the pre-contract. When a contingency is evaluated, an event is emitted to notify relevant parties of its approval or rejection. This provides accountability and transparency while minimizing fraud risk.

In technical terms, the arbiter contract has methods to mark contingencies as approved or rejected and to query the current status of any contingency. The arbiter can only approve a contingency after performing due diligence, ensuring that all necessary inspections and requirements have been fulfilled. The escrow smart contract communicates with the arbiter to verify that all contingencies are satisfied before releasing funds to the seller.

J. dNFT Smart Contracts

Dynamic Non-Fungible Tokens (dNFTs) are managed by a series of smart contracts. The primary smart contract handles the minting of NFTs, creating unique digital assets on the blockchain. These assets represent real estate properties. For the Ethereum blockchain, the appropriate standard is typically ERC-721. Each NFT is identified by a unique token ID, and the smart contract manages its metadata and behaviors. The same smart contract also contains logic for transferring NFTs and updating ownership records. This ensures that the transfer process is secure and transparent.

Additionally, a separate smart contract interacts with oracles to update the dNFT. When certain conditions are met, this smart contract requests data from the oracle. The oracle provides external data, such as property valuations or updates, which the smart contract validates and uses to update the dNFT properties. These updates are then recorded on the blockchain, ensuring that the dNFT accurately reflects the current state of the property.

K. Property Listing Smart Contract

When a property is listed, the Property Listing smart contract records information about the property and its associated NFT, such as the NFT's token ID. The Property Listing smart contract verifies the authenticity of the dNFT associated with the property listing to ensure that only valid dNFTs are being transacted. This helps prevent fraud and ensures the integrity of the transaction process. Another functionality consists of verifying the validity of a listing. In order to manage the sale, the Property Listing smart contract interacts with the Escrow smart contract.

L. Auction Smart Contract

An optional feature is to provide the possibility of auctioning a property. The basic flow is to start the auction process by specifying a starting price based on the valuation and an auction end-date. The interested parties can register their bids on the blockchain network. To ensure security in the process and the fact that interested buyers do not change their minds after placing a bid, this smart contract interacts with the aforementioned escrow smart contract. The actual process and the method of deciding the winning bid is beyond the scope of this paper, but it can be open-bid with the highest bidder as the winner, or a closed-bid, or any other method.

M. Data Storage and Encryption

Storing information directly on the blockchain network is costly [16]. The preferred method is to store the bulk of the associated data on a decentralized storage solution like the InterPlanetary File System (IPFS) and store the hash associated with that information on the blockchain.

Our solution stores the information associated with the dNFT (property description, maintenance history, legal documentation), on a decentralized storage file system. The dNFTs reference the saved data through secure links or hashes to maintain data integrity while minimizing the risk of exposing sensitive information.

There is sensitive information that also needs to be stored. For example, the legal documentation and property history must be accessible only to the owner, certain authorities and, eventually, to the buyer. The idea is to generate a symmetric key for each set of protected information, encrypt that data with the symmetric key and store on decentralized storage the encrypted data and also the generated symmetric key encrypted with the public key of the owner. This way, the owner does not need to keep track of the symmetric key. If the owner wants to access the encrypted stored data, that person can use their private key to decrypt the generated symmetric key and use the latter to decrypt the required information. The owner can choose with whom they share the protected information by providing the interested parties the symmetric key.

IV. Use-Case Scenario

Consider a typical scenario in which Alice holds a property and she wants to sell it. Firstly, Alice must use our proposed platform to mint the NFT representing her real estate property. The method of associating a real estate property to a digital asset can be done through an LLC as it was presented in the Related Work section of this paper. For simplicity's sake, Alice provides the necessary information in order for the digital asset to be created, e.g., she provides the property deed and the property description. After the dNFT is minted, Alice can now add details such as a virtual tour of the property, images or property features. She decides what information she wants to be protected and not be accessible to anybody.

One interesting aspect is concerning the market value, which must be ascertained by an authorized appraisal entity. In our example, a reputable real estate valuation provider can trigger, through a corresponding oracle, an update in the dNFT property regarding the real estate valuation. An example of such a valuation provider is Quantarium AVM, which is a company that uses artificial intelligence to determine property appraisal. Oracles can be connected to reputable entities from off-chain that provide any information that is relevant to a specific real estate property. This approach can be extended to update environmental factors, neighborhood information, and even maintenance history, provided there are reliable APIs managed by trusted legal authorities.

Alice can list the property by putting the dNFT for sale on the platform with the help of the Property Listing smart contract. All the interaction with the blockchain network, the smart contracts and the decentralized storage is done behind the scenes. All Alice needs to use is our platform's user interface and her crypto wallet.

If Bob wants to buy a property, he uses the platform to find the desired listing and manifest his buying intent. This means providing an initial deposit, which is handled by the Escrow smart contract. The details regarding the whole flow are shown in the Buying a Property section of this paper. Once the transfer of ownership is completed a new entry is added to the property history attribute of the dNFT.

This use-case scenario is just a proof-of-concept. There are legal aspects that need to be taken into account. At this point, there needs to be a notarized signed deed transfer contract between the parties involved. On the other hand, there is an increasing effort at the European Union level to adopt blockchain technology in public administration, so this scenario is not that far-fetched.

V. Conclusion

The proposed framework offers an efficient and secure way of performing real estate transactions by keeping track of the previous owners and providing the means of updating the listing through the use of dNFTs, all while transparency, traceability and security is ensured by the blockchain network and the partly encrypted data stored in decentralized storage.

Our solution streamlines the process by reducing the need for intermediaries, which cuts down costs, by automating contract execution, and by ensuring secure and transparent records. This means fewer manual checks, quicker verifications, and a reduction in bureaucratic delays, all of which simplify the process despite legislative constraints. The time saved and the reduction in administrative overhead also contribute to overall cost savings. Moreover, it can lay the groundwork for smoother integration once legal frameworks evolve. If the gas fee costs were to become a burden, a permissioned blockchain can be used to mitigate the issue.

A future direction for the presented research is regarding fractional ownership so that an NFT can be broken down into smaller fractions to be owned and sold individually. In the real estate context, it makes sense that a property is owned by multiple individuals, but selling only a part of it raises other concerns. Another interesting avenue is to integrate dNFTs in the metaverse real estate market, so interested buyers can use virtual reality to view the property.

References

[1] N. M. Ngoc, "The relevance of factors affecting real estate investment decisions for post pandemic time," *International journal of business and globalisation*, 2023.

[2] A. Acolin, M. Hoek-Smit, and R. K. Green, "Measuring the housing sector's contribution to gdp in emerging market countries," *International Journal of Housing Markets and Analysis*, vol. 15, no. 5, pp. 977–994, 2022.

[3] G. Georgiev, B. Gupta, and T. Kunkel, "Benefits of real estate investment," *Journal of Portfolio Management*, vol. 29, pp. 28–34, 2003.

[4] R. P. Malloy, J. C. Smith, A. J. Boyack, and J. J. Kelly, *Real Estate Transactions: Problems, Cases, and Materials [Connected EBook]*. Aspen Publishing, 2023.

[5] D. Aiello, M. J. Garmaise, and T. Nadauld, "What problem do intermediaries solve? evidence from real estate markets," in *What Problem Do Intermediaries Solve? Evidence From Real Estate Markets: Aiello, Darren— uGarmaise, Mark J.— uNadauld, Taylor*, [Sl]: SSRN, 2022.

[6] N. Kirit and P. Sarkar, "Escrowchain: Leveraging ethereum blockchain as escrow in real estate," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 5, no. 10, 2017.

[7] W. Rehman, H. e Zainab, J. Imran, and N. Z. Bawany, "Nfts: Applications and challenges," in *2021 22nd International Arab Conference on Information Technology (ACIT)*, pp. 1–7, IEEE, 2021.

[8] C. N. Buțincu and A. Alexandrescu, "Blockchain-based platform to fight disinformation using crowd wisdom and artificial intelligence," *Applied Sciences*, vol. 13, no. 10, p. 6088, 2023.

[9] M. Solouki and S. M. H. Bamakan, "An in-depth insight at digital ownership through dynamic nfts," *Procedia Computer Science*, vol. 214, pp. 875–882, 2022.

[10] P. Kochovski, S. Gec, V. Stankovski, M. Bajec, and P. Drobintsev, "Trust management in a blockchain based fog computing platform with trustless smart oracles," *Future Gener. Comput. Syst.*, vol. 101, pp. 747–759, 2019.

[11] U. Ogwu, "How to sell your home as an nft," 2022. Available at: https://blog.cryptostars.is/nfts-will-lead-a-revolution-in-the-real-estate-market-67e31a8b27c8, Last accessed February 23, 2024.

[12] "Propy — real estate transaction automated," 2024. Available at: https://propy.com/browse/, Last accessed February 23, 2024.

[13] B. Guidi and A. Michienzi, "From nft 1.0 to nft 2.0: A review of the evolution of non-fungible tokens," *Future Internet*, vol. 15, no. 6, p. 189, 2023.

[14] C. Karapapas, I. Pittaras, and G. C. Polyzos, "Fully decentralized trading games with evolvable characters using nfts and ipfs," in *2021 IFIP Networking Conference (IFIP Networking)*, pp. 1–2, IEEE, 2021.

[15] K. Shah, U. Khokhariya, and S. Patel, "Smart contract-based dynamic non-fungible tokens generation system," 2023.

[16] A. Jabbar and S. Dani, "Investigating the link between transaction and computational costs in a blockchain environment," *International Journal of Production Research*, vol. 58, pp. 3423 – 3436, 2020.

future internet

MDPI

Article

Design Analysis for a Distributed Business Innovation System Employing Generated Expert Profiles, Matchmaking, and Blockchain Technology

Adrian Alexandrescu †, Delia-Elena Bărbuță †, Cristian Nicolae Buțincu *,†, Alexandru Archip †, Silviu-Dumitru Pavăl †, Cătălin Mironeanu † and Gabriel-Alexandru Scînteie †

Department of Computer Science and Engineering, Faculty of Automatic Control and Computer Engineering, Gheorghe Asachi Technical University of Iași, 700050 Iași, Romania; adrian.alexandrescu@academic.tuiasi.ro (A.A.); delia-elena.barbuta@academic.tuiasi.ro (D.-E.B.); alexandru.archip@academic.tuiasi.ro (A.A.); silviu-dumitru.paval@academic.tuiasi.ro (S.-D.P.); catalin.mironeanu@academic.tuiasi.ro (C.M.); gabriel-alexandru.scinteie@academic.tuiasi.ro (G.-A.S.)

* Correspondence: cristian-nicolae.butincu@academic.tuiasi.ro

† These authors contributed equally to this work.

Abstract: Innovation ecosystems often face challenges such as inadequate coordination, insufficient protection of intellectual property, limited access to quality expertise, and inefficient matchmaking between innovators and experts. This paper provides an in-depth design analysis of SPARK-IT, a novel business innovation platform specifically addressing these challenges. The platform leverages advanced AI to precisely match innovators with suitable mentors, supported by a distributed web scraper that constructs expert profiles from reliable sources (e.g., LinkedIn and BrainMap). Data privacy and security are prioritized through robust encryption that restricts sensitive content exclusively to innovators and mentors, preventing unauthorized access even by platform administrators. Additionally, documents are stored encrypted on decentralized storage, with their cryptographic hashes anchored on blockchain to ensure transparency, traceability, non-repudiation, and immutability. To incentivize active participation, SPARK-IT utilizes a dual-token approach comprising reward and reputation tokens. The reward tokens, SparkCoins, are wrapped stablecoins with tangible monetary value, enabling seamless internal transactions and external exchanges. Finally, the paper discusses key design challenges and critical architectural trade-offs and evaluates the socio-economic impacts of implementing this innovative solution.

Keywords: business innovation; blockchain; decentralized storage; artificial intelligence; matchmaking; expert profile; security; trust; intellectual property; design analysis

Academic Editor: Gianluigi Ferrari

Received: 6 February 2025
Revised: 8 April 2025
Accepted: 11 April 2025
Published: 14 April 2025

Citation: Alexandrescu, A.; Bărbuță, D.-E.; Buțincu, C.N.; Archip, A.; Pavăl, S.-D.; Mironeanu, C.; Scînteie, G.-A. Design Analysis for a Distributed Business Innovation System Employing Generated Expert Profiles, Matchmaking, and Blockchain Technology. *Future Internet* **2025**, *17*, 171. https://doi.org/10.3390/fi17040171

1. Introduction

Innovation ecosystems are fundamental to fostering economic growth, entrepreneurship, and societal progress. Initiatives such as business incubators, hackathons, and in-person meetups have been widely deployed. However, the current environment remains fragmented and insufficiently integrated. Many innovators, particularly those in remote or under-represented regions, lack consistent pathways to engage with expert networks, secure financial resources, or access high-quality mentorship. As a result, valuable ideas often fail to mature into robust, market-ready solutions.

A common challenge faced by university students with innovative ideas and creative young people in general is the lack of access to experienced mentors. These mentors can

help transform technical concepts into viable startups. For example, a group of students from a university's AI research lab developed an advanced machine-learning model for fraud detection. They struggled to commercialize it due to their limited knowledge of market strategies, customer acquisition, and funding opportunities. They attended hackathons and online entrepreneurship courses but still lacked structured guidance. They needed more support in refining their business model, pitching to investors, and securing industry partnerships. Similarly, another team developing a blockchain-based credential verification system found it difficult to connect with legal and compliance experts. They faced challenges navigating regulatory requirements, which slowed their progress. These cases highlight the gap between technical innovation and the business ecosystem. They clearly demonstrate the need for our proposed solution to bridge this divide.

There are existing approaches, most notably incubators, that offer structured environments that combine operational infrastructure, investor connections, and curated mentorship. While these models are effective for certain use cases, they are frequently constrained by geographic accessibility, limited participant capacity, and rigid program schedules. Similarly, hackathons and physical meetups encourage rapid prototyping and peer interaction but typically occur as isolated, time-bounded events, leaving little room for sustained iterative development or long-term support mechanisms. These shortcomings are exacerbated by the logistical and financial barriers that prevent broader participation.

Existing platforms for fostering innovation and dedicated startup accelerators, like RubikHub (https://rubikhub.ro/ accessed on 23 January 2025) , ROTSA (https://rotsa.ro/en/homepage-en/ accessed on 23 January 2025), Techcelerator (https://techcelerator.co/ accessed on 23 January 2025), 100%Open (https://www.100open.com/ accessed on 23 January 2025), EdisonOpen (https://www.edison.it/en/open-innovation accessed on 23 January 2025), NineSigma (https://www.ninesigma.com/ accessed on 23 January 2025), or Y Combinator (https://www.ycombinator.com/ accessed on 23 January 2025), offer valuable opportunities for connecting innovators with potential mentors, investors, and collaborators. These platforms often provide tools for networking, funding, and showcasing business ideas.

These platforms can provide initial networking opportunities and exposure to potential investors. However, the depth and sustainability of these relationships can be inconsistent. Follow-up mechanisms for long-term mentorship or support are frequently insufficient. Additionally, current approaches often lack robust systems to protect intellectual property, maintain trustworthy data exchange, and ensure secure collaboration. As a result, participants may hesitate to share sensitive ideas or fully engage with potential partners. Many platforms also depend on manual or ad hoc methods of pairing innovators with mentors and investors, leading to suboptimal matches that fail to leverage the full potential of their expert networks.

While global initiatives and platforms aim to support entrepreneurs and researchers, they often fall short of addressing the critical challenges faced by innovators, particularly those in smaller startups, under-represented regions, or academic institutions.

Therefore, the main challenges of systems that support innovation are as follows:

Access to expertise: Innovators often struggle to connect with domain-specific experts who can provide the mentorship and guidance necessary for refining their ideas into viable business models. Current innovation hubs or consultancy platforms typically cater to larger organizations or well-connected individuals, creating an uneven playing field. This results in a gap where smaller startups or individual innovators lack the support required to thrive.

Lack of secure collaboration frameworks: The exchange of ideas between innovators and mentors poses risks related to intellectual property (IP) protection. Without robust

mechanisms to ensure the confidentiality of sensitive information, innovators may hesitate to share their ideas, fearing unauthorized usage or theft. Existing platforms rarely provide tamper-proof, transparent, and traceable systems to safeguard such collaborations.

Inefficiency in matchmaking: Traditional matchmaking between innovators and mentors is often manual, relying on limited databases or personal networks. This approach fails to consider the specific needs of the innovator, the expertise of the mentor, or the potential synergies between the two. As a result, many promising collaborations are either delayed or fail to materialize altogether.

Little or no incentivization: The lack of structured and transparent reward mechanisms for mentors discourages high-quality participation from domain experts. Existing platforms typically do not have adequate systems to recognize or incentivize meaningful contributions, resulting in low engagement levels and diminished trust in the ecosystem.

Under-representation of marginalized groups: Innovators from under-represented regions or demographics frequently lack access to innovation resources, mentors, and funding. These systemic barriers perpetuate inequality and exclude valuable contributions from diverse perspectives that could enrich the innovation landscape.

Absence of scalable and transparent systems: The innovation process involves multiple stakeholders, such as mentors, innovators, investors, and regulators. Each stakeholder requires secure access to data and a transparent workflow. Current centralized platforms struggle to scale while maintaining transparency, trust, and accountability across these diverse groups.

Without accessible and secure platforms, many brilliant ideas fail to see the light of day, ultimately resulting in lost opportunities for economic growth and societal benefit. Furthermore, the absence of transparent systems contributes to mistrust among stakeholders, hindering collaboration and investment.

The proposed solution is **SPARK-IT: Igniting Innovation with Trust, Collaboration, and Expert Mentorship**, which is a decentralized innovation ecosystem designed to address critical challenges in connecting innovators with experts, fostering collaboration, and enabling secure idea development. By leveraging cutting-edge technologies such as blockchain, artificial intelligence, and tokenomics, SPARK-IT creates a transparent, scalable, and inclusive environment that empowers innovators from diverse backgrounds to access mentorship, refine their ideas, and build viable business models.

The platform integrates the business model canvas (BMC) as a structured tool for innovators to articulate their proposals for collaboration and specific guidance needs. This strategic management template allows innovators to map out and analyze critical aspects of their business ideas, such as key activities, value propositions, customer segments, and revenue streams. Through the platform, innovators can submit their BMC in an interactive or document-based format, highlighting areas where mentorship is required. By encouraging innovators to approach their ideas systematically and enabling experts to focus their guidance, the integration of the BMC fosters clarity, precision, and productive collaborations within the SPARK-IT ecosystem.

At the core of SPARK-IT is its decentralized architecture, which ensures the security, traceability, and transparency of all interactions. Blockchain technology and decentralized storage underline the platform's key functionalities, including secure storage of proposals, immutable records of non-disclosure agreements (NDAs), and tamper-proof tracking of interactions between innovators and mentors. This creates a trusted environment where intellectual property is protected, fostering confidence among participants.

The platform's AI-driven matchmaking engine enhances efficiency and personalization by analyzing innovator proposals and expert profiles to recommend the most suitable mentors. This dynamic and feedback-driven system ensures that each collaboration is

tailored to the specific needs and expertise of its participants, thereby maximizing the potential for success.

SPARK-IT also incorporates a dual-token system to incentivize meaningful participation and high-quality contributions. Reputation tokens reflect user performance and influence future matchmaking, while reward tokens have monetary value, compensating mentors for their guidance and expertise. This mechanism ensures sustained engagement and accountability while fostering a vibrant and trustworthy community.

The modular design of the proposed solution allows for flexibility and scalability, making it adaptable to various use cases, such as startup acceleration, academic collaboration, and investment opportunities. By bridging gaps between innovators, experts, academia, and the private sector, SPARK-IT represents a transformative step in democratizing access to innovation resources and driving economic and social progress.

The SPARK-IT platform builds upon and integrates existing methodologies and tools to create a robust ecosystem for innovation. The following section describes the state-of-the-art research and methodologies that serve at the base of SPARK-IT's design, focusing on the use of the business model canvas, AI-powered matchmaking, and secure information storage to establish a foundation for its novel contributions.

The remainder of this paper is structured as follows. Section 2 presents a comprehensive review of the state-of-the-art approaches in business innovation platforms, expert profiling, AI-driven matchmaking, decentralized storage, and reward and reputation systems. Section 3 introduces the SPARK-IT system architecture, detailing its modular components, decentralized data management, intellectual property protection, and token-based incentives. Section 4 describes the design challenges and trade-offs, including balancing decentralization with usability, managing tokenomics in a permissioned blockchain, scalability, and the financial model design. Moreover, the socio-economic impacts are discussed when it comes to democratizing access to mentorship, enhancing trust in global innovation and potential use cases in academia and industry. Finally, Section 5 provides conclusions, summarizing key contributions, identifying potential limitations, and outlining future research directions.

2. State-of-the-Art

This section provides a structured overview of the key technologies and methodologies that inform and inspire the design and development of the SPARK-IT platform. We examine state-of-the-art solutions in four core dimensions: business model innovation, expert profile validation, AI-driven matchmaking, and decentralized data management with incentive mechanisms. This clearly illustrates how SPARK-IT addresses existing gaps and introduces novel enhancements.

Firstly, we discuss the business model canvas (BMC) and its contemporary variations. We highlight their strengths in organizing and visualizing business ideas. We also address limitations, such as their complexity and lack of standardization. By analyzing these models, SPARK-IT can strategically adopt or adapt specific canvas structures. The goal is to make them user-friendly and tailored specifically to innovators and mentors on our platform.

Secondly, we examine techniques for creating accurate and verified expert profiles from external sources, emphasizing methods such as data scraping and AI-driven information extraction from platforms like LinkedIn. These methods allow SPARK-IT to ensure expert credibility and reliability, directly addressing common concerns innovators have about finding trusted mentors.

Next, the analysis focuses on AI-driven matchmaking solutions, comparing various methods including ontology-based approaches, NLP algorithms, and advanced deep learning models. Understanding the advantages and challenges associated with these techniques

guides SPARK-IT towards a balanced, hybrid matchmaking strategy, combining semantic understanding with keyword analysis to achieve precise and meaningful connections between innovators and experts.

Lastly, we review current blockchain-enabled decentralized storage, reward, and reputation management systems, illustrating their importance in fostering trust, transparency, and effective collaboration. We highlight how incentive mechanisms, such as reputation scores and reward tokens, motivate high-quality interactions. This underscores SPARK-IT's commitment to transparency, accountability, and user-centric incentives within its ecosystem.

By clearly outlining these technological foundations, this section sets the stage for SPARK-IT's unique contributions, showcasing how our platform builds upon and advances current state-of-the-art approaches to better support secure, transparent, and effective innovation-driven collaborations.

2.1. Business Model Canvas for Innovation

The business model canvas (BMC) was first introduced in [1]. Its purpose was to facilitate business model innovation. It helps businesses and entrepreneurs visualize, design, and improve their business models clearly and systematically.

To apply the business model canvas, an entrepreneur must complete nine building blocks that represent the key elements of any business model. The process should begin by defining the customer segments, which are the different groups of people or organizations the business aims to reach and serve. Following this, the value propositions must be described, outlining the bundle of products and services that create value for these customer segments. Next, the channels should be outlined, detailing how the company communicates with and reaches its customer segments to deliver the value proposition. It is essential to specify the types of customer relationships the company establishes with its customer segments. Additionally, revenue streams must be identified, which represent the cash generated from each customer segment. The key resources, or the most important assets required to make the business model work, should be listed. Furthermore, the key activities, which are the essential actions a company must take to operate successfully, need to be detailed. Identifying the key partnerships is crucial, as these are the networks of suppliers and partners that help the business model function. Finally, the cost structure must be described, encompassing all costs incurred to operate the business model.

The BMC offers numerous benefits that contribute to its widespread adoption and effectiveness. It provides clarity and focus by breaking down the business model into key components, offering a clear and structured view of the business, and its visual nature enhances communication between team members and stakeholders. The BMC is also highly flexible, allowing for quick modifications and iterations and enabling businesses to adapt rapidly to changes. By covering all critical aspects of a business, it provides a holistic view, ensuring that no important area is overlooked. Additionally, the canvas encourages collaboration through brainstorming sessions, fostering teamwork and collective problem-solving.

The triple layered business model canvas [2] extends the traditional business model canvas by adding two additional layers: an environmental layer, based on lifecycle assessment, and a social layer, based on stakeholder management. These additional layers offer a more comprehensive understanding of the company's value creation and of the impact that the product can have on the market. However, a drawback of this model is its increased complexity and the time required to fill out all layers. This complexity may overwhelm some users, especially if it is their first experience with the model. The tool may also necessitate a higher level of expertise to effectively apply life cycle assessments

and stakeholder management principles, potentially limiting its accessibility to smaller businesses or those with limited resources.

Recent proposals have suggested various improvements and adaptations to the BMC, particularly for startups and innovation-focused businesses. One such framework is the blitz canvas [3], which incorporates ten different stages specifically designed for software startups. Initially, the focus is on building the foundation by establishing the mission, vision, goals, core culture, and competencies of the business. This is followed by studying the user, which involves utilizing personas and user stories to understand the users' goals, frustrations, and motivations. The next stage, defining the solution, outlines product goals and features while creating prototypes for user feedback. Identifying key differentiators and customer touchpoints forms the unique selling proposition. The framework then emphasizes collecting qualitative feedback in the user's feedback stage to validate the solution concept. It also involves studying the competition to analyze competing market offerings and inform product development. Business model elements are incorporated from the BMC, including IP and "As a Service" offerings. The synergies stage identifies interdependencies and synergies within the business model. Managing growth involves planning for growth management and complementary product offerings. Finally, metrics are established to identify key performance indicators (KPIs) and track progress, ensuring the business remains on course. In [4], the authors proposed enhancements to the BMC for startups focused on innovation. These enhancements include a section entitled disruptive strategy, which comprises elements such as product democratization, new business models for new technologies, fulfilling unmet customer needs, defending against low-end disruptors, and adapting to shifts in competition.

The advantage of tailoring the business model canvas to specific domains consists in its ability to better address the challenges and opportunities of that domain, which increases the relevance of the analysis. This, however, happens at the cost of the level of standardization. Benchmarking solutions that use different frameworks is a more difficult task, as different metrics and models are used. In addition to that, tailored models can have a steeper learning curve, which may prevent new users from properly applying them.

Another suggested improvement is the integration of the BMC with customer experience aspects. For example, [5] suggests defining the cognitive, emotional, physical, sensorial, and social elements marking a customer's interactions with the company. This framework better aligns business model innovation with customer values and needs, ensuring that strategic decisions are directly informed by customer experiences. This focus can lead to more relevant and appealing offerings, improving customer satisfaction and loyalty. Nevertheless, there is a risk of misalignment if the insights derived do not accurately reflect customer priorities or if the strategic orientation of the company changes rapidly due to market conditions.

To mitigate the previously described problem of requiring very specialized knowledge from users for them to be able to implement the BMC, in [6], the authors propose implementing the BMC, taking into consideration the three maturity levels defined by a user: novice, expert, and master. The BMC helps novices by providing a structured framework to elicit and build coherent business models. Experts use the BMC to evaluate interactions between business model elements and outline key threads in the business model's story. Masters create multiple versions of business models to evaluate alternatives and retain the history of the model's evolution.

Alternatives to the BMC, although not as widely adopted, have also been discussed. In [7], the authors suggest using the value proposition canvas (VPC) instead of the BMC to focus on why customers should buy the products/services of the company. The VPC promotes a more customer-oriented strategy, but there is a risk that businesses might

oversimplify their approach to value creation, ignoring the interconnectedness of various business model components that the BMC emphasizes.

2.2. Creating an Expert Profile Using External Sources

Creating a comprehensive and accurate expert profile involves various technologies and methodologies. These methods gather, verify, and synthesize data from multiple external platforms. The following papers mainly focused on the LinkedIn platform, as it is one of the most popular platforms for professional endeavors and career related networking.

In [8], the authors present how LinkedIn data can be mined in order to create expert profiles using LinkedIn API. The data available on LinkedIn contain redundancies because users enter their information in various ways. For example, in the "Education" section, the name of a university might appear in different formats across profiles, even though all refer to the same institution. Because of this, the paper presents how the collected data are normalized before being used so that redundancies are removed.

A novel approach to extracting expert information from personal homepages using deep learning techniques is described in [9]. A key contribution is the segmentation of web pages into smaller text units, enabling the model to focus on relevant information more effectively. Integrating prior knowledge, such as named entity recognition and regular expressions, significantly enhances the model's ability to accurately identify and extract crucial data.

The authors propose using a long short-term memory (LSTM) network to capture the contextual dependencies within text units. This allows a more precise classification of words and segments, leading to improved overall extraction performance. The experimental results presented in the paper demonstrate the model's superior accuracy in generating comprehensive expert profiles compared to traditional methods. While the model presented in the paper handles diverse web page formats, there are still challenges related to varying page structures. The paper highlights the potential for future work to address these issues and further enhance the model's capabilities.

In [10], the authors present a data-driven approach for profile extraction based on Resumes. In the article, the authors leverage natural language processing (NLP) techniques like keyword extraction and document representation to create profiles. While the paper focuses on resumes, the underlying techniques used can be similarly applied for creating expert profiles from LinkedIn profiles.

Another paper related to profile extraction is [11], in which the authors aim to create a comprehensive database of college alumni by scraping data from LinkedIn profiles. To do this, web scraping techniques are leveraged to collect and organize details from the user's profiles, such as employment history, educational background, skills, and professional connections. The methodology presented involves using Python programming and web scraping libraries to extract information from public LinkedIn profiles. This is carried out while ensuring compliance with LinkedIn's terms of service and privacy policies. After collecting the data, data cleaning and structuring techniques are necessary to ensure the accuracy and consistency in the collected data.

The paper demonstrates that web scraping techniques can collect data from LinkedIn profiles. However, a limitation is that it does not address interpreting unstructured data such as descriptions or comments. In order to overcome this issue NLP, techniques can be used to extract the essential information from these unstructured data sources.

In [12], the authors aim to extract content-based user profiles from the data available on LinkedIn to have an image of the users' interests that can be used to recommend interesting

academic research papers. In order to do so, an extractor system is developed, which processes the information extracted from LinkedIn for building the researcher profile.

A novel idea presented in the paper was to utilize both the professional data of the researcher and that of their social graph connections. This provides a more accurate picture of the researcher's interests. The paper shows that the data extracted from the connections have been revealed to be a valuable source of information, increasing the performance of the system since social networks grow around common interests.

2.3. Matchmaking Solutions Using AI

The application of artificial intelligence techniques to matchmaking systems has gathered significant attention in recent years. This section reviews key contributions in this domain, highlighting their methodologies, innovations, and limitations.

Wu et al. [13] introduce a method for matching experts to projects by employing domain ontologies to model expertise and project requirements. Using Protégé [14], the authors formalize concepts into structured trees, facilitating the computation of semantic similarities. While this approach effectively organizes knowledge, it is susceptible to semantic heterogeneity, where different terms refer to the same concept, resulting in potential mismatches. Although partially mitigated, this limitation remains unresolved, affecting the method's overall reliability.

In ref. [15], the authors propose a system for automating the pairing of researcher biosketches with funders' requests for proposals (RFPs) using advanced natural language processing techniques. The authors develop four deep neural network architectures based on a fine-tuned BERT model, comparing their performance with support vector machines and logistic regression as baselines. The DNNs utilize cross-encoding and Siamese encoding strategies, with CNNs or Bi-LSTMs as post-BERT layers. Among these, the cross-encoder BERT model with a Bi-LSTM layer and BC2BT-based data augmentation achieves the highest accuracy. This work demonstrates the potential of sophisticated NLP methods for automating complex matchmaking tasks, emphasizing the benefits of effective data augmentation techniques.

A matchmaking system for linking resumes with job descriptions to identify suitable candidates is presented in [16]. The approach integrates two models: a content-based recommender system and an NLP-based method using gensim for text summarization. The gensim model employs the TextRank algorithm and transformers to summarize resumes and job descriptions into comparable lengths, followed by k-nearest neighbors for similarity measurement. The TextRank algorithm focuses on keyword extraction, while transformer-based models paraphrase and abstract text. These complementary approaches enhance flexibility but underscore the challenge of balancing keyword-based and semantic representations in matchmaking.

In [17], the authors introduce an AI-powered platform to connect industry experts with companies seeking specialized skills. The system relies on the Word2Vec model with a Skip-Gram architecture to calculate semantic similarity between keywords provided by mentors and companies. By analyzing these keywords, the platform identifies and recommends experts whose skills align closely with organizational needs. However, the dependency on keyword-based inputs poses a limitation, requiring precise articulation of expertise and requirements to achieve effective matches.

The reviewed studies collectively underscore the diverse methodologies applied to matchmaking, ranging from ontology-based approaches to deep learning and NLP techniques. While ontology-based methods excel in knowledge organization, they are prone to semantic mismatches. In contrast, deep learning and transformer-based models offer advanced semantic understanding but may require extensive computational resources

and well-curated training data. These works pave the way for further research to address limitations such as semantic heterogeneity and reliance on structured input formats.

2.4. Storing Information/Documents with Managed Access and Methods Employing Decentralized Storage and Blockchain Technology

Controlling access to data in decentralized computing systems represents a manifold challenge in itself. Today, managing different user roles and access rights is complicated. This complexity arises from the variety of available sources for user authentication and validation. The second difficult task is handling the increasing mixture of data formats, representations and storage options that are available. Kayes et al. [18] present a possible solution for representation heterogeneity and access control. The authors focus on data that is available from multiple sources and introduce a unified data ontology to normalize different forms of data description. They then extend context-aware role-based access control (CAAC) models with a unified set of context-sensitive access control policies to manage data access. While the paper does not focus on decentralized storage per se or on blockchain, it does have the merit of standardizing access control policies for different data sources.

In [19] the authors use argumentation-based agents to model data access interfaces. The focus of the paper is on solving data sharing, access control, and privacy protection in Internet-of-Things (IoT) environments and smart applications. IoT data are categorized into private and public, and agents are divided into internal and external. The authors then expand category-based access control meta-models and emergency policies and introduce two argumentation schemes:

- An argumentation scheme for data access control, which allows access control management for internal data requests;
- An argumentation scheme for access-category assignment, which allows access control for external data requests.

Agents process these schemes in argumentation-based reasoning patterns and decide whether or not to allow access to the requested data. Aside from handling multiple data sources, the authors also address the issue of multiple authentication solutions.

An ontology-based access control (OBAC) model is presented in [20]. The authors address secure access to FAIR (findability, accessibility, interoperability, and reusability) data and consider three categories of information: the data itself, associated metadata expressing FAIR information, and additional metadata about the users. Target metadata are represented as a knowledge graph used to describe semantic relationships between the concepts (categories) expressed by the actual data. OBAC allows implementation of role and context dependent access policies based on these knowledge graphs. If authorized, users received access to data described by the current category stratum (i.e., graph neighbors of the same "parent" node) and to all the "parents" included on the path generated by the same "parent" category (i.e., the origin major concept for the current graph path). The proof-of-concept presented by the authors is agnostic to the actual location of the data (e.g., a Web API or a classic database) through these metadata categories describing the data.

Kiran et al. [21] focus on cloud computing and data access control. The authors introduce SA-ODAC (security-aware mechanism and ontology-based data access control) as a potential solution to control access rights over data stored in such open, decentralized, and distributed systems. The proposed model is composed of two distinct operational components: secure awareness techniques (SAT) and an ontology-based data access control (ODAC) module. ODAC is employed to handle data access control based on role and permission policies. SAT operates on the cloud level and encompasses the components

required to encrypt and decrypt data and handle encryption key management. Through SAT, the authors ensure that sensitive data are encrypted before being transmitted to the cloud for storage and decrypted by authorized clients. The scenario is similar to the one required by SPARK-IT to store project proposals in IPFS.

The challenges of decentralized data access control (DDAC) over consortium blockchains are addressed in [22]. Blockchains do not employ centralized data administration, unlike traditional, centralized database systems. Participants in such public networks have limited or no control over access control policies since the ledger data are replicated to all the nodes in the network. Consequently, participants must keep confidential data outside the ledger or encrypt data before storing it. The authors formally define DDAC with atomicity, consensus, and confidentiality (ACC) constraints:

- *Atomicity* implies that a transaction is either discarded or applied to the nodes of all allowed participants and that write operations included in such a transaction must be atomic.
- *Consensus* implies that data owners may veto a transaction involving their data with a single vote.
- *Confidentiality* implies that participants have read access to data if and only if they are included in the corresponding read-allowed group.

Following this definition, the authors implement a DDAC framework using Hyperledger Fabric. ACC modules are embedded within this framework as access control managers. The framework also employs encryption modules and ledger partitioning techniques to further control data access based on attribute-based access control (ABAC) rules.

In [23], the authors present a blockchain-based role-based access control (B-RBAC) mechanism for data sharing within federated systems. The authors focus on medical data and IoT environments. The access control mechanism is based on smart contracts, and the required access control policies are stored as key/value pairs through these smart contracts. The key represents data's security attribute level, and the value describes the corresponding set of conditions that must be matched to grant access. The authors further rely on the concept of "colored coins" to define security attribute tokens, which are then used to assess different user permissions and data access layers. Distributed access control decisions are performed through automatic executions of smart contracts. The solution also has the much-required advantage of supplying traceability information alongside solving RBAC in distributed federated environments. The prototype system is developed using HyperLedger Fabric.

Data privacy and security are the focus of [24]. The authors address the issues raised by multidimensional data aggregation and access control in cloud-based IoT scenarios. Data collected from different smart things are encrypted using EBGN homomorphic encryption before being aggregated and pushed into the blockchain. Furthermore, different data access attributes are handled through ciphertext-policy attribute-based encryption (CP-ABE): the EBGN private keys for each data dimension are encrypted using the CP-ABE algorithm. This ensures that authorized parties are able to access only the corresponding required data. The authors propose the use of a trusted authority to handle key management and renewal. Access renewal is ensured through the regeneration of EBGN's public and private keys for each affected data dimension.

In [25], the authors are primarily focusing on the development of a new blockchain-based protocol for secure data sharing and access. The article emphasizes the growing interest in decentralized storage solutions, particularly those based on blockchain technology. Blockchain's inherent features, such as immutability, transparency, and decentralization, offer potential advantages in addressing the limitations of centralized systems. The paper utilizes the InterPlanetary file system (IPFS) to store large files. IPFS is a peer-to-peer

distributed file system that provides content addressing and versioning. The authors acknowledge that blockchain is not suitable for storing large amounts of data and leverages IPFS to complement the blockchain's capabilities. The reference text emphasizes the importance of encrypting files before they are added to the IPFS to ensure data confidentiality. The authors state that "the files should be encrypted before added to IPFS to ensure data confidentiality." However, the paper does not delve into the specifics of the encryption approach, such as the encryption algorithms or key management schemes employed. The primary focus of the paper is on the blockchain-based protocol for secure data access and collaboration rather than the intricacies of the encryption process itself.

The framework in [26] proposes the use of a permissioned blockchain and distributed hash tables (DHTs) to decentralize the storage of PingER data, thereby eliminating the reliance on a central repository. The metadata of files are stored on the blockchain, while the actual files are stored off-chain using DHTs at multiple locations within a peer-to-peer network of PingER monitoring agents. The use of DHTs enables efficient data lookup and retrieval in a decentralized manner. The framework addresses the issue of blockchain bloat by storing the actual PingER data files off-chain. It employs erasure coding (K-of-M) to ensure data redundancy and availability, even if some nodes in the network go offline. The Merkle root, stored on the blockchain, provides a mechanism for auditing the integrity and immutability of the stored data. The permissioned blockchain serves as the foundation for managing identity and access control within the network. The use of digital signatures and cryptographic hash functions ensures the security and integrity of transactions and data stored on the blockchain. The simplified Byzantine fault tolerance (SBFT) consensus algorithm is proposed to achieve agreement on the state of the data across the distributed network, enhancing the system's fault tolerance and security.

The framework presented in [27] leverages the strengths of both IPFS (InterPlanetary file system) for decentralized storage and proxy re-encryption (PRE) for secure access control to healthcare data. The combination of these technologies allows the system to securely store large volumes of healthcare data while ensuring that only authorized users can access and decrypt specific files. In the proposed framework, sensitive data (such as medical records) are first encrypted using symmetric encryption, and then the encrypted data are stored on the IPFS network. The resulting IPFS hash (which serves as a unique identifier for the data) is stored on the blockchain. Using the PRE cryptographic technique, the system allows a third party (proxy) to convert ciphertext from one public key to another. The proxy does not learn anything about the underlying plaintext during this conversion. In this framework, PRE is used to manage access control for the encrypted data stored on IPFS.

In [28,29], the authors propose a complex architecture for decentralized news article extraction. Articles are stored off-chain, and proofs in the form of content hashes are saved on the blockchain. This way, anyone can verify the integrity of the stored article and be certain that it was not tampered with in any way. A variation of the decentralized retrieval system is presented in [30], where the emphasis is on modularity, which is also the case for our system. Another work that employs storing important information on decentralized storage and the proof on the blockchain network is the one from [31], which is related to real estate transactions. Blockchain is also used to ensure data integrity in the context of monitoring the driver's sobriety level [32].

2.5. Reward Systems with Monetary Value

Incentive mechanisms, as the driving force for maintaining long-term system operation, are indispensable elements of blockchain systems. The advanced properties of blockchain can also contribute to designing effective and efficient incentive mechanisms.

Han et al. [33] broadly review academic papers related to the incentive mechanisms in blockchain and blockchain-based incentive mechanisms. To systematically evaluate these papers, the authors proposed a set of requirements based on incentive properties and costs.

In [34], the authors propose a general decentralized rating framework based on blockchain, supporting recommender systems and rewarding its users for their reviews. The ratings of items, the reputations, the tokens of users, and the algorithms exploited to compute the score of the items are stored on the blockchain. The system directly supports cryptocurrency payments. It also allows item owners to accept off-chain payments by manually invoking a smart contract to confirm payment execution.

Merrill et al. [35] present a token-locking reward model that can be used to incentivize miners to accept transactions without forcing clients to sacrifice their tokens. The model can also be used to incentivize service providers. The authors' analysis showed that the model reduces volatility commonly associated with cryptocurrencies. Paying interest to token holders lowers the opportunity cost than holding fiat currencies. The authors claim that their model of rewarding service providers with interest generated from locked tokens is more profitable for clients compared to paying for services directly. An interesting aspect of the model is that the locked tokens are temporarily out of circulation.

In [36], the authors developed a dynamic model of the platform economy, where tokens are used as a means of payment among users and issued by the platform to finance investment. Tokens facilitate user transactions and compensate distributed ledger-keepers, open-source developers, and crowdfunders for their contributions to platform development. Platform owners maximize their seigniorage by carefully managing the token supply. This management considers that users optimally determine token demand and form rational expectations about token price changes.

By focusing on token price volatility, in [37], the authors investigate how reward uncertainty affects user contribution to a tokenized digital platform. The results reveal that, for systems that involve and reward user creativity, token price volatility facilitates the platform's short-term effect but impairs long-term user creativity.

In [38], the authors examined whether blockchain can serve as the technology underpinning decentralized marketplaces to promote trust. By utilizing tokens as an incentive mechanism, the authors demonstrated that rewarding peers for reporting malicious behaviors can mitigate misconduct. Despite its simplicity, the token rewarding mechanism can be used to incentivize users to behave consistently and tackle trust issues.

In [39], the authors suggest three adjustments to make a currency more stable. Firstly, they propose minimizing the difficulty of mining new blocks to prevent rapid changes in the supply of coins. Secondly, they recommend adjusting mining rewards if block intervals significantly change, thereby stabilizing the coin supply. Lastly, they introduce negative interest concepts to remove old coins from circulation. Mechanisms such as fees gradually reduce the coin's value over time. This approach encourages spending instead of hoarding.

The first suggestion only applies to proof-of-work (PoW) blockchains, but the other two have a broader applicability zone and can be used in other solutions as well. For example, in [40], the authors explain that cryptocurrencies are not widely adopted online due to their high volatility and propose the use of stable coins to solve the volatility issue by securing cryptocurrency with a stable asset.

Four types of collaterals are discussed for stablecoins, each with its own strong points and weak points. Fiat-collateralized stablecoins are pegged to fiat currency and are centralized, which contradicts decentralization principles. Crypto-collateralized stablecoins are pegged to other cryptocurrencies and are often over-collateralized due to volatility. Commodity-collateralized stablecoins are pegged to commodities like gold or oil and are centralized in a similar way to fiat-collateralized stablecoins. Non-collateralized stablecoins

have no backing assets and use algorithms to adjust supply based on demand to stabilize the price.

The solution proposed in [41] is a dual-deposit escrow smart contract that ensures cheat-proof delivery and payment for digital goods without third-party intervention. Necessary assumptions include the following: (1) the product can be secured with a digital key, (2) the buyer can verify the product using a pre-known hash value, (3) both parties have asymmetric key pairs known to each other, and (4) transaction fees for deploying and interacting with the smart contract are negligible. The purpose of this solution is to secure transactions by guaranteeing that both the buyer and the seller fulfill their obligations.

Building on the previous idea, the system presented in [42] uses a smart contract to act as an escrow, holding the buyer's payment until the trade is completed. Disputes are handled by having parties wager on their honesty, with an arbiter deciding the outcome. The contract requires both parties to place a deposit (wager) that is refunded if they behave honestly. An interesting concept is that there is both a penalty and a reward system. They are computed based on the wager size and a repayment constant.

The contract's security is analyzed using game theory, proving that it ensures honest behavior if the arbiter is biased towards honesty. The contract achieves strong game-theoretic security if the penalties for dishonesty are sufficiently high, making deviation from honesty unprofitable.

In [43], the author explains how a cryptocurrency should be designed to be valuable. Some guidelines are provided, which must be followed before and after the creation of the currency. Pre-creation guidelines include the following: defining the purpose of issuance, determining desirable behaviors and appropriate rewards, establishing measures to prevent undesirable behaviors such as hacking and spam attacks, and deciding on the block generation cycle along with the amount of cryptocurrency to be issued.

Post-creation guidelines include maintaining scarcity by controlling total issuance. They also involve creating continuous market demand and balancing these methods to stabilize or increase cryptocurrency value.

An example of how value can be created for a cryptocurrency is found in [44]. Cyclists earn cycle tokens by cycling, with the tokens calculated based on the plausibility and distance of the ride. The real-world value of these tokens is derived from the economic, health, and ecological benefits associated with cycling. A marketplace is established where cycle tokens can be exchanged for various incentives, such as discounts or spare parts. The number of tokens generated corresponds to the square root of the kilometers cycled, with a cap on the maximum number of tokens that can be earned per track and per day.

This example highlights the importance of aligning cryptocurrency incentives with real-world benefits. It also emphasizes controlling token supply and providing practical uses. These measures ensure cryptocurrency sustainability and value.

2.6. Reputation Systems

Reputation systems are of the utmost importance in decentralized networks, as they enforce trust and accountability among participants. These systems evaluate and quantify entities' trustworthiness based on their behaviors and interactions. This evaluation ensures reliability and promotes honest participation. Unlike traditional centralized reputation systems, decentralized approaches leverage blockchain technology to offer enhanced transparency, immutability, and security.

The authors in [45] have created a separate blockchain for storing reputation, where reputation is quantified objectively by storing binary ratings, 1 for a positive transaction and 0 for a negative one. A positive transaction is one where the user receives the requested file. Identity creation is linked to IP addresses, which increases the cost of creating multiple

identities. A Proof-of-Stake mechanism is used for low-reputation users; they must stake a small amount of currency into a triple-signed wallet (involving the sender, receiver, and a third party) to discourage dishonest behavior.

This binary approach to reputation has the advantage of simplicity. However, it lacks scalability because it cannot be applied to systems requiring more nuanced representations of reputation. In many scenarios, reputation is more nuanced and might need to be represented on a continuum, such as a rating scale from 1 to 10 or through qualitative feedback regarding user performance and behavior. Some notable proposals are made in [46], such as the weighting of feedback based on the evaluator's reputation, ensuring that feedback from low-reputation users matters less. To motivate users to provide ratings and comments, the system incorporates a monetary reward mechanism. The proposed system leverages the Ethereum blockchain, using Solidity for smart contracts, Truffle for development, Web3.js for blockchain interaction, and IPFS.js for decentralized storage.

The reputation system proposed in [47] uses a beta reputation model to calculate a vehicle's reputation score, which ranges between 0 and 1, based on its observed behavior. This computational model uses the beta distribution function, where the ratio of positive to negative behaviors determines the score. Positive behaviors, such as correctly reporting traffic conditions, increase the reputation score, while negative behaviors, such as false reports, decrease it.

The system incentivizes honest behavior by adjusting the number of coins a vehicle can use based on its reputation score, promoting active and honest participation in the network. To protect privacy, vehicles generate one-time public keys for local blockchain interactions, concealing their actual identities while maintaining the trustworthiness of their activities. This approach ensures that while a vehicle's temporary identity and reputation level are known to others in the network, its overall activity history remains hidden.

Bellini et al. [48] examine various decentralized reputation systems and divide them into three categories: deterministic, probabilistic, and flow models.

Deterministic methods use straightforward algorithms to compute reputation scores. These methods typically involve summing up ratings or feedback received from other participants. Probabilistic methods use statistical models to estimate the reliability of a participant based on past behavior. These models take into account the uncertainty and variability in user actions and make use of methods such as Bayesian networks, maximum likelihood estimation, and hidden Markov models. Flow models assess reputation by examining the flow of trust across the network. They consider both direct interactions (where one participant directly rates another) and indirect interactions (where reputation is influenced by ratings from trusted intermediaries). Examples of these models include the PageRank algorithm, EigenTrust algorithm, and trust propagation models (if user A trusts user B, and user B trusts user C, then user A might have a derived trust score for user C).

In the WorkerRep system [49], task completion involves evaluators scoring submissions based on two criteria: completeness and quality. These scores are then weighted according to the credibility of the evaluators, which is determined by their own reputation on the platform. Outlier scores are excluded, and a consensus score is calculated from the remaining evaluations. Workers who participate in evaluating peers also have their reputations updated based on the accuracy of their evaluations relative to the consensus. There is also a rewards system that takes reputation into account.

RepChain [50] uses a consortium blockchain (CBC) to enable e-commerce platforms to collaboratively maintain a decentralized ledger. It incorporates one-show anonymous credentials, created with two-move blind signatures, to protect customer identities and prevent multiple rating abuses. Zero-knowledge range proofs are used to verify the accuracy of ratings, protecting against abnormal rating attacks. Reputations are updated

using secure multiparty computation. Additionally, consensus hashing is used to verify ratings through batch processing and consensus hashes.

The authors in [51] propose a reputation system that incorporates a trust value, a distrust value, and an uncertainty value that adds up to 1. The trust value represents the level of trust that node i has towards node j. It is a measure of positive interactions and indicates the probability that node j will perform as expected. The higher the trust value, the more confidence node i has in node j's reliability. The distrust value shows the level of distrust node i has toward node j. A higher distrust value indicates greater suspicion that node j won't meet expectations. The uncertainty value captures the level of uncertainty in the opinion of node i regarding node j. It accounts for the lack of sufficient interaction history or contradictory information, representing the extent to which node i is unsure about node j's trustworthiness. A higher uncertainty value means that node i has less information or is less certain about the performance of node j.

The reputation calculation includes the timeliness of interactions and the property of interactions. For example, the timeliness function applies a decay factor to older interactions to give more weight to recent activities. Positive and negative interactions are treated differently, with negative interactions leading to a more rapid decline in reputation.

3. System Design Overview

3.1. Platform Architecture

The SPARK-IT platform is designed to foster collaboration between innovators and experts through a secure, decentralized, and modular system. The design heavily leverages blockchain technology, AI-driven expert matching, and a decentralized storage architecture to ensure data security, transparency, and efficient communication between users.

Figure 1 presents the SPARK-IT platform architecture. The main components illustrated in this figure are as follows:

- Blockchain: permissioned blockchain and smart contracts for token management, proof of data records, contract execution records, and innovator and expert interaction records.
- Off-chain storage: templates for NDA contracts and the business model canvas, web application, and user data; high performance in-memory caches.
- IPFS (InterPlanetary File System): distributed storage for public and encrypted data.
- API Gateway: API front-end entry point that provides a unified API interface to the system; integrates/orchestrates back-end calls to all system components and provides authentication/authorization mechanisms, as well as support for audit services.
- Web App @innovator: front-end web application for innovators, used to interact with the system for profile and offer management, activity history, tokens, and various statistics; provides the means to interact with the matchmaking module to assist in expert selection.
- Web App @expert: front-end web application for experts, used to interact with the system for profile management, to accept/reject offers, activity history, score, accrued tokens, and various statistics.
- Web scraper: provides data to verify and prefill expert profiles; gathers data from professional social media platforms; can be extended to include other data sources.
- Data extractor: analyzes and extracts expert profile information from web scraper data; can be extended to extract data from different sources; written in a data-source agnostic way by means of pluggable extraction templates.
- Matchmaking module: employs AI and other algorithms used to perform the matchmaking between the innovators and experts; matchmaking data are recorded into blockchain along with reasoning data traces to ensure transparency into the selection process.

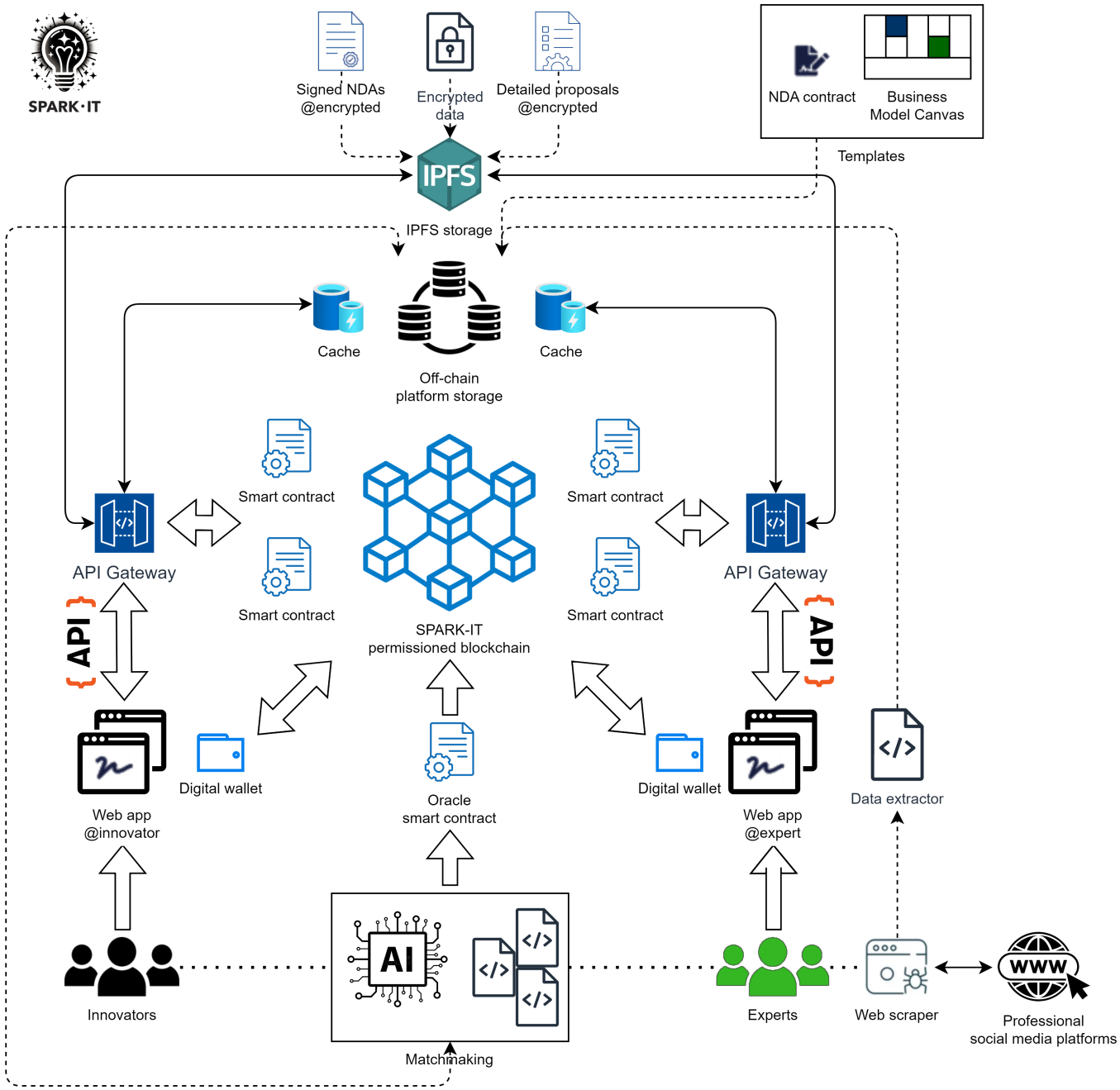


Figure 1. SPARK-IT platform architecture.

3.2. Functional Components

3.2.1. Expert Profile Generation

The expert profile component supports the creation and maintenance of expert profiles by merging user-provided information with automatically gathered data. Experts begin by entering their details through a user interface, supplying elements such as expertise, experience and education. This core data forms the initial profile and serves as a baseline for further enrichment.

The component employs two submodules for data acquisition and processing. The web scraper retrieves supplemental data from specified external platforms, such as LinkedIn, ResearchGate, ORCID, etc. By focusing on well-defined interfaces, it can be updated to incorporate additional sources as they become relevant. Once the web scraper acquires the raw data, the data extractor processes it, using configurable extraction tem-

plates designed to handle different source formats. Because these templates are pluggable, the data extractor can be adapted without altering the overall system whenever new formats or platforms arise.

The expert profile component integrates these steps into a single workflow. First, it verifies user input and identifies missing or outdated information. Next, it triggers the web scraper to gather relevant details that could fill the gaps or refresh the profile. Then, the data extractor refines and structures the retrieved data to align with the expert's existing information. After processing, the system presents the updated profile for expert verification or approval. If needed, the expert can correct or refine data within the interface, ensuring that the final profile remains accurate and aligned with the professional record.

This approach reduces the time experts spend manually updating their profiles and helps maintain current, consistent records. By supporting multiple data sources and flexible extraction methods, the component remains resilient to changes in the data landscape. Through its integrated workflow, profile creation, and maintenance process, the component helps experts present up-to-date information in a single, organized location.

3.2.2. Innovator–Expert Matchmaking

The matchmaking engine is a foundational element of the platform, designed to establish connections between innovators and mentors using natural language processing and machine learning algorithms. The engine evaluates several parameters, such as project requirements, expertise areas, reputation scores, and semantic details from proposals. This ensures the matches it generates are relevant.

A key strength of the matchmaking engine lies in its modular design. Each module represents a specialized approach to matchmaking, enabling a diverse range of techniques to be applied depending on the specific requirements of the task. These modules can be employed interchangeably or combined into hybrid solutions, providing flexibility and adaptability. The following paragraphs delve into the core modules of the matchmaking component, each contributing a unique capability to enhance the precision and effectiveness of the platform.

One foundational technique utilized in the matchmaking system is term frequency-inverse document frequency (TF-IDF), a classic approach to text analysis. This method transforms textual data into numerical vectors based on the frequency of keywords, enabling the system to compute the similarity between proposals and expert profiles using cosine similarity. By quantifying the angular relationship between the vectors, cosine similarity ensures that matches are not influenced by the magnitude of the documents.

Despite its simplicity, TF-IDF remains an indispensable component, particularly in contexts where computational efficiency is critical, or when keyword alignment is important. However, it is inherently limited in its ability to discern semantic relationships or account for variations in language. As such, while TF-IDF is well-suited for straightforward matching tasks, it lacks the depth required for more nuanced scenarios.

Advancing beyond keyword-based approaches, bidirectional encoder representations from transformers (BERT) bring significant innovation to the matchmaking process. Unlike traditional models, BERT captures the contextual relationships of words within a sentence by analyzing them in their bidirectional context. This results in dense embeddings that encapsulate the deeper semantic meaning of the text. By applying cosine similarity to these embeddings, the system is able to identify matches that align not only in terminology but also in context and intent.

The use of BERT is necessary for the matchmaking engine, as it ensures that the platform can recognize and connect related concepts even when they are expressed in

different terms. This context-aware matching capability positions BERT as a critical tool in scenarios demanding precision and semantic depth.

Building on the foundation of BERT, Sentence-BERT (S-BERT) introduces optimizations tailored for sentence-level comparisons. S-BERT generates embeddings that are specifically designed for the rapid calculation of similarity between short text segments, such as proposal descriptions and expert profiles. As with BERT, cosine similarity is employed to measure alignment, but S-BERT achieves this with significantly improved efficiency.

The enhanced speed and computational efficiency of S-BERT make it particularly well-suited for real-time applications where responsiveness is key. Its fine-tuning for semantic textual similarity tasks further ensures that even subtle differences in language are captured with precision, enhancing the overall performance of the matchmaking engine.

An alternative approach within the matchmaking system is provided by latent Dirichlet allocation (LDA), which focuses on topic modeling rather than direct text comparison. LDA decomposes text into a mixture of topics, identifying overarching themes that can serve as the basis for matching. By aligning proposals and expert profiles based on thematic relevance, LDA offers a broader perspective that complements the more granular techniques used elsewhere in the system. This method adds an additional dimension to the matchmaking engine, enabling it to uncover connections that might not be apparent through keyword or semantic analysis alone.

Another critical component of the system is the clustering of proposals and profiles using embeddings generated by BERT. The system applies clustering algorithms, such as K-means or DBSCAN, to group similar entities. This helps manage large datasets and identify possible matches efficiently. This clustering approach leverages the rich contextual information encoded in BERT embeddings, ensuring that the groupings are meaningful and relevant.

The scalability of the clustering process is essential for accommodating the platform's growth. By pre-organizing proposals and profiles into clusters, the system is able to streamline matchmaking operations and maintain high levels of efficiency, even as the dataset expands.

The integration of Sentence-BERT embeddings into the clustering process further enhances the system's efficiency. By utilizing the computationally optimized embeddings produced by S-BERT, the system achieves faster clustering without compromising accuracy. This capability is particularly important in dynamic environments where real-time clustering is required, making S-BERT clustering a very important component of the matchmaking module.

Finally, the hybrid approach within the matchmaking system demonstrates the power of modularity and adaptability. The hybrid module combines techniques such as TF-IDF for keyword alignment, LDA for thematic matching, and BERT for semantic precision. This creates a comprehensive, balanced matchmaking solution. Cosine similarity is applied across the various embeddings to compute similarity scores, which are then combined using a weighted scoring system.

By leveraging the unique strengths of its individual components, the hybrid module achieves a level of precision and adaptability that would not be attainable with any singular method. This approach exemplifies the modular nature of the matchmaking system, positioning it as a robust and scalable solution for connecting innovators with mentors.

To evaluate the effectiveness of the different matchmaking algorithms, the mean reciprocal rank (MRR) metric will be utilized as the primary measure of performance. This metric is particularly well-suited to the platform's design, where an innovator selects a single expert from a ranked list of recommendations. MRR measures ranking quality by placing more importance on recommendations where the selected expert is higher on the

list. It reflects the algorithm's effectiveness in prioritizing relevance. By averaging the reciprocal ranks of the selected experts across all sessions, MRR provides a robust and interpretable measure of ranking efficacy. This approach ensures a precise evaluation of the algorithms in their ability to deliver meaningful and contextually aligned matches, which is a fundamental objective of the platform.

3.2.3. Proposal Submission and Collaboration Workflow

SPARK-IT introduces a novel functionality that enables innovators to leverage expert guidance and obtain mentor feedback for submitted proposals, representing a key innovation in proposal evaluation systems. The overview of the basic innovator-expert flow is shown in Figure 2.

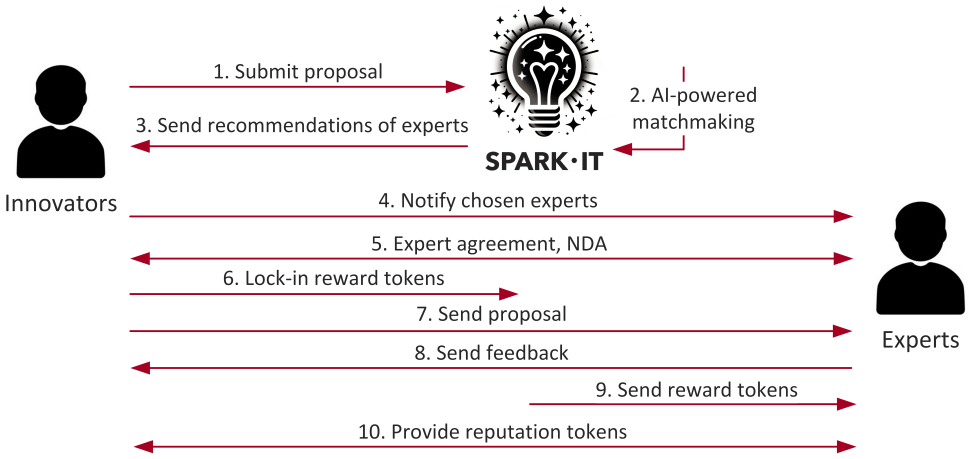


Figure 2. SPARK-IT collaboration workflow. Each arrow originates from the entity initiating the interaction—be it an innovator, expert, or the SPARK-IT platform—and points toward the intended recipient.

The SPARK-IT system features an AI-powered recommendation mechanism that identifies and suggests the best expert matches based on the innovator's specific proposal requirements. Once the chosen experts agree to collaborate, they must sign a non-disclosure agreement (NDA), which is stored on IPFS. By using blockchain to store proof of signing, SPARK-IT ensures that the NDA records are secure, immutable, and verifiable, providing both traceability of the signing process and non-repudiation, meaning the participants cannot deny their actions after the fact.

The detailed steps are as follows:

1. Proposal Submission by an Innovator
   - (a) The *innovator* submits a short-version proposal in a preferred format, e.g., a plain document or business model canvas in which are underlined the aspects in which he/she needs guidance.
   - (b) The proposal is encrypted and stored in the decentralized storage system (DSS, i.e., IPFS module in Figure 1) and a proof of submission is recorded on the blockchain for traceability.
2. AI-Powered Matching
   - (a) The AI algorithm analyzes the proposal based on keywords, business needs, and innovator's requirements.
   - (b) The system matches the proposal with suitable experts based on their areas of expertise, reputation, and profile data.
   - (c) The *innovator* reviews matched experts and selects preferred mentors.

3. Expert Notification
   (a) *Experts* are notified through the platform if the innovator selected them regarding his/her proposal.
   (b) If they agree, they become *mentors* for that proposal.
4. Expert Agreement and NDA Signing
   (a) Both parties sign an NDA through the platform.
   (b) The NDA is encrypted and stored in the DSS, with proof on the blockchain network.
5. Collaboration and Communication
   (a) Secure communication channels are established using RESTful APIs through the SPARK-IT platform.
   (b) The *innovator* shares detailed proposal information with the mentor.
   (c) The collaboration is tracked, and feedback is provided through the platform.
6. Reputation and Reward Tokens
   (a) *Mentors* earn reputation tokens based on the quality of feedback and contributions.
   (b) *Innovators* use reward tokens to compensate *mentors* for their services.
   (c) All transactions and token exchanges are recorded on the blockchain for transparency.
7. Project Development and Feedback
   (a) *Innovators* and mentors work together to refine the project.
   (b) *Mentor* provides continuous feedback, which is stored securely.
   (c) Reputation tokens are awarded by both *innovators* and *mentors* based on the effectiveness of the collaboration.
8. Completion and Public Disclosure
   (a) Upon project completion, select details of the collaboration may be made public with consent.
   (b) Additional reputation tokens can be awarded based on public feedback and project success.

The objective is to establish a structured and legally robust framework for collaborations, ensuring the protection of intellectual property (IP). Innovators often hesitate to share their ideas and business plans with potential mentors due to concerns about the unauthorized use or theft of their IP. In the absence of proper legal agreements, enforcing IP rights and safeguarding sensitive information becomes difficult. The mentor choosing protocol requires the signing of a non-disclosure agreement (NDA) before any detailed project information is shared. Unstructured collaborations may lead to misunderstandings, misaligned expectations, and ineffective mentoring relationships. To mitigate these risks, the protocol ensures that both innovators and mentors have clear expectations and objectives, fostering more productive and successful outcomes.

All agreements and collaborations are securely recorded and stored using blockchain technology, providing an immutable record that can be referenced in case of disputes. This protocol is designed to protect intellectual property, build trust, ensure accountability, and enhance the platform's overall credibility and appeal. By addressing these critical challenges, SPARK-IT offers a secure and efficient environment for innovators and mentors to collaborate and drive innovation.

3.3. System Components

From a technical point of view, the main system components that are deployed are the frontend server, the backend core with the storage module, the matchmaker, the expert profile, the off-chain storage, the IPFS nodes and the blockchain nodes with the smart

contracts, as can be observed in Figure 3. All these component deployments contain the conceptual functionalities shown in Figure 1.

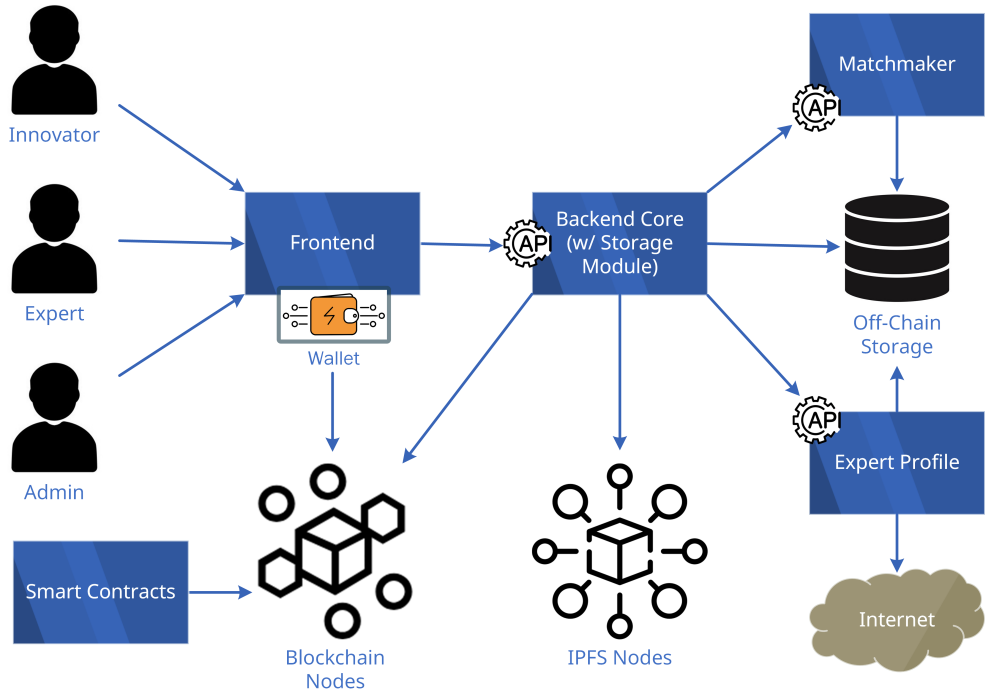


Figure 3. SPARK-IT system components.

One of the core flows from our proposed system is creating and sending a proposal for collaboration. This flow is showcased in the sequence diagram from Figure 4. There are several main components that interact with each other in this case. Those components are the user's browser, the frontend server, the backend server, the database, the decentralized storage (i.e., the IPFS) and the blockchain network. The other aforementioned communication scenarios follow a somewhat similar flow.

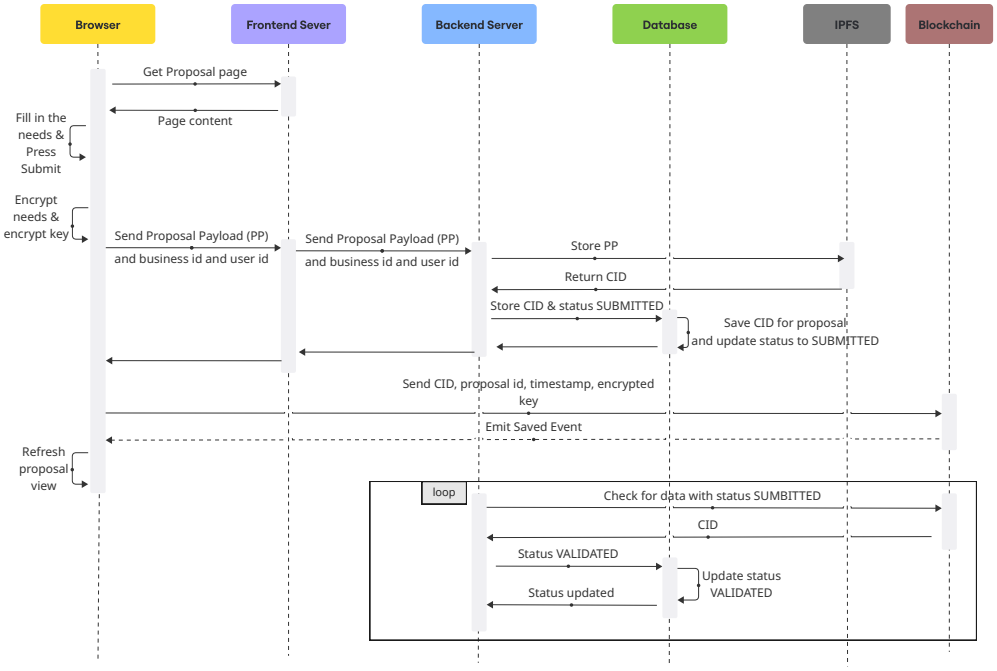


Figure 4. SPARK-IT sequence diagram for creating and sending a proposal.

The proposal information is stored encrypted on IPFS, and the proof is on the blockchain network. Periodically, the backend server checks that the proof was properly stored on the blockchain and updates the database status accordingly.

3.4. Intellectual Property Protection

SPARK-IT offers both protection of intellectual property and efficient matchmaking between innovators and experts, ensuring the protection of intellectual property and building trust among participants. Innovators often fear that sharing their ideas and business plans with potential mentors could lead to unauthorized use or theft of their intellectual property. In addition to struggling to find suitable mentors, innovators must also ensure their intellectual property is safeguarded and secure access to funding opportunities. The innovator–expert collaboration protocol described in Section 3.1 is a structured process. It enables innovators to get feedback from experts while protecting intellectual property through NDAs and secure communication channels. The protocol aims to build trust, ensure accountability, and enhance the platform's credibility and attractiveness.

User information, including the NDA and expert feedback, is stored in a custom encrypted format that leverages blockchain and decentralized storage, as outlined in Section 3.2. Proposals and associated data are encrypted and stored on IPFS, with proofs stored on the blockchain. This ensures that only authorized parties can access the data, providing trust, traceability, and verifiable transparency. The main motivation for this functionality is to ensure the security and privacy of intellectual property and sensitive data, fostering a trusted environment for collaboration. Innovators need assurance that their ideas and business plans are protected from unauthorized access and potential theft. By providing robust security measures, SPARK-IT fosters a trusted environment where innovators feel safe to share their intellectual property. Ensuring data privacy and transparency through blockchain technology builds trust among users, encouraging more participation from innovators and experts. Experts and mentors are more likely to participate in a platform that guarantees the protection of their contributions and maintains a high standard of data privacy.

The primary advantage of utilizing blockchain for intellectual property lies in eliminating reliance on credit from third-party intermediaries. This enables more individuals to store data on distributed blockchains, ensuring that the information remains immutable. Blockchain-based intellectual property solutions offer key benefits, including traceability, tamper resistance, and resource efficiency.

3.5. Token-Based Incentives

The platform includes a dual-token system consisting of reward tokens (SparkCoins) and reputation tokens to incentivize meaningful participation and encourage trustworthy behavior. The system is designed to balance financial rewards and reputational impact, encouraging both innovators and experts to contribute actively and responsibly.

3.5.1. Reward Tokens

SparkCoins serve as the main monetary incentive for experts. These tokens hold tangible value and can be used for transactions within the platform or exchanged externally (e.g., converted to stablecoins using an exchange). The reward token system employs blockchain-based escrow accounts, smart contracts, and tokenomics principles to ensure secure and dispute-resilient transactions.

Innovators submit their proposals through the platform, specifying the mentorship requirements. As part of this submission process, they lock a predetermined amount of SparkCoins into a blockchain-based escrow account. This locking mechanism pro-

vides a guarantee of payment for the expert's services, creating trust and accountability between participants.

Upon mutual agreement, the expert commences the mentoring process. During this period, the following is true:

- The locked SparkCoins remain in escrow, accessible only under the conditions explicitly defined within the associated smart contract.
- The release of tokens to the expert is contingent upon one of the following:
    1. **Approval by the innovator (manual release):** The innovator manually triggers the release of funds upon satisfactory completion of the mentorship process.
    2. **Automatic release:** Tokens are automatically disbursed to the expert upon the conclusion of the collaboration period, provided no disputes have been raised.

In the event of a dispute, the platform retains the locked SparkCoins in escrow until the conflict is resolved. Although dispute resolution mechanisms are beyond the scope of the current implementation, future versions could integrate transparent processes to address such issues effectively.

SparkCoins are implemented as wrapped USDT tokens on the platform's permissioned blockchain. The process of obtaining these coins involves the following steps:

1. A smart contract deployed on the Ethereum blockchain locks the specified amount of USDT when a user initiates a swap. The user sends the amount of USDT to this contract.
2. Upon successfully locking the tokens, the smart contract emits an event that records the amount of USDT locked and specifies the recipient address on the permissioned blockchain.
3. An off-chain service, referred to as the bridge component, monitors the Ethereum network for these lock events. The service validates the event to ensure the correct amount of USDT is locked and that the transaction is legitimate.
4. Once the lock event is validated, the bridge component transmits a message to the permissioned blockchain, relaying the details of the locked USDT.
5. A corresponding smart contract on the permissioned blockchain mints an equivalent amount of SparkCoins upon receiving the validated message from the bridge component. This contract implements the ERC20 standard and manages the SparkCoins, facilitating their minting or transfer based on user actions, such as compensating experts for completed services.

This mechanism ensures the secure and transparent exchange of value between the Ethereum network and the platform's permissioned blockchain, providing liquidity and maintaining a 1:1 backing ratio between USDT and SparkCoins.

3.5.2. Reputation Scores and Reputation Tokens

A clear distinction that has to be made within the system is between reputation scores and reputation tokens:

- **Reputation score:** This is an average of all ratings received by a user (on a scale from 1 to 10). For example, if an expert consistently receives high ratings, their reputation score reflects their overall performance and reliability.
- **Reputation tokens:** These are derived from the reputation score received after each interaction with an innovator and are directly tied to monetary incentives. For instance, for each collaboration with an innovator, up to five reputation tokens can be earned. A score of 10 yields five reputation tokens, while lower scores yield proportionally fewer tokens. Accumulated tokens contribute to a pool that can later be converted into reward tokens. These tokens are specifically designed for experts.

The reputation scheme within the SPARK-IT platform is designed to incentivize high-quality contributions from both experts and innovators while maintaining a transparent and reliable system for evaluating interactions and expertise. The purpose of enforcing this mechanism is to create a culture of accountability, reward useful contributions, and improve trust.

Reputation scores are used in the matchmaking phase to ensure that experts with higher reputation scores are prioritized in the process. Therefore, these tokens directly influence the visibility of users on the platform, with higher scores increasing their prominence in search results and recommendations.

The platform incentivizes users through a structured process:

- **Innovators' Perspective:**
  - Innovators receive ratings from mentors, which reflect the quality of their proposals and collaborative interactions. These ratings contribute to their innovation reputation score, which can attract future mentors and potential investors.
  - The platform can serve as a gateway for innovators to showcase projects to potential investors, using their reputation scores as an indicator of trustworthiness and project viability.
- **Experts' Perspective:**
  - Experts contribute to innovative projects, contributing to their professional growth and helping others.
  - Experts are rated by innovators on criteria such as relevance, depth of feedback, and overall helpfulness. These ratings translate into reputation tokens, which build a mentoring reputation score visible on their profile.
  - Experts can earn additional SparkCoins by maintaining high standards of quality when working with innovators, achieving strong reputation scores and then exchanging reputation tokens for reward tokens

At predefined intervals, reputation tokens can be exchanged for reward tokens at a fixed exchange rate. This ensures stability and predictability in the platform's ecosystem, increasing user trust and encouraging sustained engagement. To cover exchange expenses, a fixed fee policy is implemented, applying a small percentage to all transactions between innovators and experts.

The initial implementation allows both innovators and experts to award scores ranging from 1 to 10 based on the quality of their interactions. To ensure unbiased evaluations, these scores are not visible to the counterpart until both parties have completed their assessments.

3.6. Technology and Method Selection

SPARK-IT integrates blockchain technology, AI-driven matchmaking, and decentralized collaboration to create a secure and scalable innovation ecosystem. The selection of blockchain frameworks, AI models, and consensus mechanisms is guided by the need for high efficiency, data integrity, scalability, and low transaction costs while ensuring trust and transparency in mentor-innovator interactions.

3.6.1. Blockchain Framework Selection

The platform employs Hyperledger Besu, an Ethereum-compatible permissioned blockchain, to balance decentralization with governance control. Hyperledger Besu is chosen over fully public blockchains like Ethereum due to its lower transaction costs, enterprise-grade security, and flexibility in defining access control policies. The permissioned nature of the blockchain ensures that only verified participants (innovators, mentors, and stakeholders) can validate transactions, reducing the risk of spam and fraudulent activity.

Additionally, InterPlanetary file system (IPFS) is used for decentralized storage of intellectual property, business proposals, and NDAs, ensuring data availability while maintaining privacy. IPFS is selected over traditional cloud storage due to its tamper-proof nature and cryptographic content address, which prevent unauthorized data manipulation. Blockchain stores cryptographic hashes of IPFS files, ensuring data integrity while keeping large files off-chain to maintain efficiency.

3.6.2. Consensus Mechanism Selection

The system uses the QBFT (quorum Byzantine fault tolerance) consensus mechanism, which is optimized for permissioned blockchains. Unlike proof-of-work (PoW), which is computationally expensive and inefficient for enterprise applications, QBFT provides fast, deterministic finality, ensuring that transactions (such as mentorship agreements, token-based incentives, and proposal submissions) are processed quickly and reliably.

QBFT is selected over proof-of-stake (PoS) and delegated proof-of-stake (DPoS) because it provides enhanced security against Sybil attacks while maintaining low energy consumption. It ensures that only approved validators participate in transaction validation, reducing the risk of malicious actors influencing the network.

3.6.3. AI Model Selection for Expert Matchmaking

SPARK-IT leverages a hybrid AI-driven matchmaking system that combines natural language processing (NLP), machine learning (ML), and graph-based recommendation algorithms. The bidirectional encoder representations from transformers (BERT) model is selected for semantic analysis of business proposals and mentor profiles, ensuring that AI recommendations consider contextual nuances rather than relying solely on keyword matching. Sentence-BERT (S-BERT) is used to generate dense vector embeddings, which help measure the similarity between mentor expertise and innovator needs more accurately.

Additionally, graph-based recommendation models such as Node2Vec and graph convolutional networks (GCNs) analyze expert networks to identify the most well-connected, highly-rated mentors for a given business challenge. A reinforcement learning (RL) feedback loop continuously refines matchmaking accuracy by incorporating user satisfaction ratings and engagement metrics to improve recommendations over time.

4. Discussion

4.1. Design Challenges and Trade-Offs

Designing a decentralized platform that facilitates interactions between innovators and experts involves navigating several challenges and trade-offs across various technical and operational layers. Key design challenges and trade-offs we considered for the SPARK-IT platform are described below.

4.1.1. Balancing Decentralization with Usability

In decentralized platforms, the user experience can often be complex due to the need for users to manage their own private keys, interact with blockchain networks, and handle decentralized storage systems. Balancing the inherent complexity of decentralized technologies with the need for a seamless, intuitive user experience is a significant challenge.

- *Security vs. usability*: To achieve true decentralization, users need to control their private keys, which can be difficult for non-technical users. Managing keys and cryptographic signatures can be cumbersome but is essential for maintaining security. Offering simplified solutions, such as dedicated wallets or social recovery systems, can improve usability but may introduce centralization risks. In SPARK-IT, we opted for a decentralized blockchain-based solution, where the centralized components assist in

interaction flows. The essential data are stored encrypted in IPFS, and proof-of-data is anchored in blockchain. Moreover, the user can choose any digital wallet solution as long as it is compatible with an Ethereum blockchain.

- *Speed vs. decentralization*: Decentralized systems can introduce latency, particularly when dealing with blockchain-based interactions or IPFS file retrieval. Optimizing for speed (e.g., using centralized servers or content delivery networks) might decrease decentralization, as centralized components may become single points of failure. In SPARK-IT, we use dedicated in-memory caches to make the overall system as responsive as possible. However, if these fail, the front-end client applications, for both innovator and experts, have a fallback alternative that goes directly to the decentralized components (blockchain and IPFS). Moreover, we opted for a POS (proof-of-stake) permissioned blockchain that can process a high number of transactions per second.
- *Complexity of interactions*: Decentralized platforms often require users to engage with more steps, such as verifying transactions on-chain, interacting with smart contracts, or managing encrypted data. While a decentralized system gives more control to the user, simplifying these processes without losing the core decentralized nature is not an easy task. Designing intuitive front-end systems that abstract away the complexities of blockchain and IPFS interactions while still ensuring underlying security and decentralization is crucial. In SPARK-IT, we streamline the workflows for the users so that the interactions with the employed decentralized technologies, blockchain and IPFS, as well as the entire process of encrypting/decrypting data, are seamless. Our system provides a straightforward wallet integration and guided workflows for on-chain actions. The front-end applications contain JavaScript libraries that perform the heavy-lifting tasks of connecting and interacting with the blockchain and IPFS.

4.1.2. Managing Token Economics in a Permissioned Blockchain

In a permissioned blockchain, tokens can be used to incentivize behaviors such as expert participation, mentoring session completion, or contribution to the platform's success. However, managing a token economy in a permissioned environment introduces complexities regarding distribution, valuation, and governance.

- *Centralized control vs. tokenomics flexibility*: In a permissioned blockchain, some degree of centralization is often necessary for governance, which can limit the flexibility of the token economy. For example, the entity managing the platform might need to control token issuance, governance, or rewards, potentially undermining the decentralization aspect of token management. In SPARK-IT, we designed governance smart contracts that are open to everyone. Even if we use a permissioned blockchain, any potential innovator or expert can join the platform. Therefore, the community formed around our governance protocol is not restricted and behaves in a similar way as the ones deployed in public blockchains.
- *Token valuation and liquidity*: In a permissioned blockchain, the token's value may be more difficult to establish without the market dynamics that come with a public blockchain. Users might be less willing to use or accept a token that lacks liquidity or a clear external value. In SPARK-IT, we decided to use wrapped tokens from public blockchains (such as USDT). Moreover, even if we can support any token from public blockchains, we decided to go for stablecoin tokens to assure predictability and to protect our users from excessive volatility inherent to crypto markets. We have developed dedicated blockchain bridges that assist in moving assets between public blockchains and our permissioned blockchain.
- *Governance mechanisms*: Without strong decentralization, governance around tokenomics could become a bottleneck. Decision-making about token issuance, rewards,

and penalties might rely on a small group of entities, leading to potential trust issues and reducing transparency. In SPARTK-IT, we employ a decentralized governance mechanism (through a dedicated DAO and governance tokens) that ensures that decisions regarding token distribution and rewards are made in a transparent and inclusive manner. Our protocol is open to everyone, as everyone can register in the SPARK-IT platform as either an innovator or an expert.

- *Fair reward distribution*: Allocating tokens equitably while ensuring they incentivize high-quality contributions is very important of the platform's tokenomics. This process requires balancing several critical aspects, including fairness, transparency, and alignment with the platform's overall objectives. SPARK-IT achieves this through a dual-token system, where reputation tokens serve as a measure of quality and reward tokens (SparkCoins) hold monetary value. This mechanism not only incentivizes meaningful participation but also ensures that the rewards are tied to measurable and impactful contributions.

4.1.3. System Scalability

Our solution is designed to scale efficiently as the number of users, data storage needs, and computational demands increase. The system achieves this through a modular service architecture, decentralized storage solutions, and optimized resource management. The platform relies on a service-based design where components such as storage, frontend, backend business logic, expert profile generation, and matchmaking operate independently. This allows for horizontal scaling by dynamically deploying additional backend instances as demand grows, ensuring that the system can handle increasing loads without performance degradation. A load-balancing mechanism can distribute requests across multiple backend servers, preventing bottlenecks and maintaining high availability.

For data storage, SPARK-IT employs a hybrid approach that combines blockchain, decentralized storage, and traditional databases. Business proposals, mentorship agreements, and intellectual property-related documents are stored using the InterPlanetary file system (IPFS), reducing the reliance on centralized storage while ensuring tamper-proof, efficient file access. Only critical data, such as cryptographic proofs of intellectual property and mentorship agreements, are stored on Hyperledger Besu's permissioned blockchain. User profiles, general metadata, and transactional records are maintained in a distributed MariaDB cluster, which is optimized for scalability through sharding and replication techniques. This ensures fast retrieval of essential data while minimizing the computational overhead of storing all records on-chain.

4.1.4. Scalability Considerations in AI-Driven Matchmaking

The matchmaking system must efficiently process and evaluate large data sets to match innovators with suitable experts. AI algorithms used for matchmaking could require significant computational resources, which may hinder the scalability of the platform as the number of users grows. The need for real-time performance and personalization also adds complexity to scaling the system.

- *AI accuracy vs. computational resources*: AI-driven matchmaking systems often rely on complex models that require considerable computational power. As the platform scales, maintaining the same level of personalization and accuracy in recommendations becomes challenging due to the increased resource demands. In SPARK-IT, we consider a federated learning approach where multiple systems make predictions and can be rewarded based on their input. Some of these systems can be hosted by third parties, thus ensuring the overall system scalability.

- *Real-time performance vs. scalability*: The need for real-time matchmaking (matching experts to innovators on-demand) adds another layer of complexity. Real-time AI processing can lead to bottlenecks in systems with growing data inputs, especially when scaling to thousands of innovators and experts. In SPARK-IT, we devised an asynchronous workflow, where the innovator initiates the matchmaking process from its front-end application and, at a later time, receives a notification that the results are ready.
- *Data privacy vs. matching precision*: For AI models to provide effective matchmaking, they need access to a large amount of data about both innovators and experts (e.g., skills, preferences, historical records, and reputation scores). However, the need for privacy and confidentiality may limit the amount of data accessible to AI systems, affecting the accuracy of the matches. In SPARK-IT, we defined a clear set of attributes that are available to the AI systems. These attributes are needed to achieve the objective of the matchmaking process without disclosing more information than needed. Moreover, an additional data anonymization layer can and will be deployed in the next versions of the platform.

4.1.5. Financial Model Design

To ensure financial sustainability and accessibility for diverse user groups, SPARK-IT has to use a multi-faceted financial model. Users can choose between the following:

- *Flat Transaction Fee*: A small, fixed percentage fee on transactions between innovators and experts ensures predictable costs, ideal for occasional users.
- *Subscription Model*:
  - *Basic Plans*: Provides low-cost access to essential features, allowing users to explore the platform without significant commitment.
  - *Premium Plans*: Offer advanced features such as enhanced matchmaking algorithms, analytics, and priority support.
- *Organization Sponsorship*: Larger entities (e.g., universities or corporations) can sponsor platform licenses for their members. They can purchase SparkCoins and then redistribute them to their members as they see fit.
- *Pay-Per-Use Extra Features*: Specific value-added features, like a badge that proves the profile information has been verified, are available for a fee.

Each of these design considerations requires a careful balance of trade-offs. Striking the right balance is essential for creating a platform that is both functional and scalable while maintaining the decentralized aspects and providing a user-friendly experience. The employed solutions and trade-offs must be iterated over time to optimize the system further as the platform grows.

4.2. Socio-Economic Impacts

4.2.1. Democratizing Access to Mentorship and Innovation

SPARK-IT has the potential to transform innovation ecosystems by democratizing access to mentorship and resources. The platform uses decentralized architecture and AI-driven matchmaking. This bridges the gap between innovators and experts, no matter their geographic location or socio-economic background. Innovators from under-represented regions, startups, and academic institutions can gain access to high-quality mentorship and collaboration opportunities traditionally reserved for well-connected individuals or larger organizations. This democratization fosters inclusivity, empowering a broader range of voices and ideas to contribute to global innovation.

4.2.2. Enhancing Trust in Global Innovation Ecosystems

Trust is a critical factor in successful collaborations within innovation ecosystems. SPARK-IT enhances trust by integrating blockchain technology to ensure data transparency, security, and immutability. The use of smart contracts and decentralized storage safeguards intellectual property and ensures that all transactions and interactions are traceable and verifiable. These measures reduce hesitation among participants to share sensitive information, promoting a culture of openness and collaboration. Furthermore, the dual-token incentive system ensures accountability, encouraging meaningful participation and reinforcing trust among all stakeholders.

4.2.3. Potential Use Cases in Academia and Industry

SPARK-IT's versatile design makes it highly applicable to both academic and industrial contexts. In academia, the platform can facilitate research collaborations by connecting researchers with industry experts, mentors, or funding opportunities. It provides a secure environment for sharing ideas, receiving feedback, and co-developing innovative solutions. In industry, SPARK-IT is effective in accelerating startup development. It connects entrepreneurs with experienced mentors and investors and helps foster partnerships across sectors. Its ability to adapt to various use cases ensures relevance across diverse domains, making it a valuable resource for driving economic growth and societal progress.

Through these impacts, SPARK-IT contributes to creating a more inclusive, trusted, and efficient global innovation ecosystem.

4.3. Performance Metrics for Evaluating the Platform's Effectiveness

To ensure the effectiveness of SPARK-IT in bridging the gap between technical innovators and business mentors, the performance can be evaluated using key metrics that assess recommendation accuracy, system efficiency, and user engagement. The accuracy of expert recommendations is measured through precision and recall based on user feedback that compares AI-generated mentor matches with successful mentorship engagements. By analyzing the percentage of mentor-innovator pairings that lead to productive collaborations, the platform continuously refines its AI-driven matchmaking engine. Additionally, SPARK-IT monitors user satisfaction by gathering structured feedback on mentorship quality, business advice relevance, and overall collaboration outcomes. Mentor retention rates serve as an indicator of expert engagement, showing how effectively the platform incentivizes and sustains long-term participation.

4.3.1. Qualitative and Quantitative Insights from User and Stakeholder Research

When designing the SPARK-IT platform, we first conducted a user research stage through questionnaires and stakeholder interviews. More specifically, we conducted an online survey through Typeform to gather insights. The survey was designed to adapt to the different personas of our target audience: startup founders or innovators, experts, and startup accelerators. Up until now, we have received 28 responses from individuals from Central and Eastern Europe and Brazil. The distribution among the three respondent categories (experts, innovators, and organizational representatives) is illustrated in Figure 5.

The questions addressed in the survey can be categorized into the following themes:

1. **Mentor Selection and Matchmaking**
    - Mentor information;
    - Importance of the mentor's professional network;
    - Validation criteria for mentors (experience, academic background, certifications, peer reviews, and mentee feedback);

- Method of mentor–startup matching (self-selection, automated algorithms, and manual review).

2. **Feedback and Validation**
   - Methods of incorporating mentee feedback;
   - Role of peer reviews;
   - Comfort level with publicly sharing mentorship outcomes;
   - Demonstrating mentor effectiveness (professional achievements, endorsements, and peer reviews).
3. **Mentorship Process and Success Metrics**
   - Elements of successful mentorship (continuous feedback, defined goals, communication, and respect);
   - Tracking and measuring progress (milestone achievements, mentee feedback, mentor evaluations, project outcomes, and KPIs).
4. **Mentorship Management**
   - Handling unsuccessful mentorships (switching mentors, feedback loops, and follow-ups);
   - Scheduling and appointment methods (calendar integration and availability slots).
5. **Budget and Financial Considerations**
   - Budget preferences and flexibility;
   - Factors affecting budget decisions (mentor expertise, project nature, and engagement duration).
6. **Intellectual Property Protection**
   - Methods for IP protection (NDAs, confidentiality agreements, secure document sharing, and legal support).
7. **Resources and Support**
   - Elements of a successful mentorship (networking opportunities, communication tools, administrative support, and training materials).

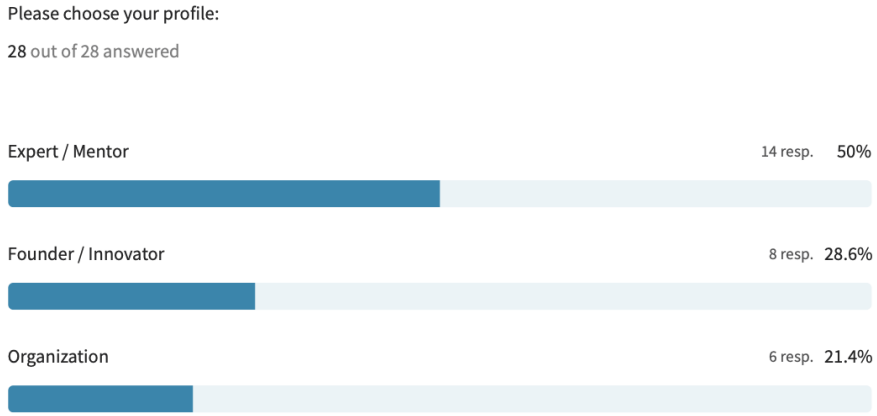


Figure 5. Distribution of respondents by profile type: experts/mentors, founders/innovators, and organization representatives (n = 28).

Regarding **mentor selection and matchmaking**, users emphasized industry experience (87.5%) and personality fit (75%) as the most important factors. A majority (71.4%) preferred a hybrid approach combining automated algorithms and human review for effective matchmaking. That is the reason behind our approach, where we designed an AI algorithm

to perform the matchmaking and we present the results to the user so that the final decision is theirs.

In the area of **feedback and validation**, respondents significantly valued public mentee ratings (83.3%) and showed high comfort levels with publicly sharing mentorship experiences (71.4%). Participants considered professional achievements (85.7%) and endorsements (78.6%) to best showcase mentor effectiveness. Therefore, we designed the public reputation system from SPARK-IT, and we integrated logic to scrape for achievements and endorsements on platforms that allow it, such as LinkedIn.

Concerning the **mentorship process and success metrics**, most respondents identified continuous feedback and defined goals (78.6%) as critical elements for mentorship success. Progress measurement was highly recommended through milestone achievements (82.1%) and mentee feedback (67.9%). This led to the decision to impose financial penalties on the mentor in case the deadline for collaboration is not respected.

In terms of **intellectual property protection**, respondents had to choose between six different protection measures. Non-disclosure agreements (NDAs) were identified as the most important, endorsed by 64.3% of respondents, followed by confidentiality agreements (53.6%) and secure document sharing (35.7%). These findings strongly support our platform's integration of NDAs as a primary mechanism for safeguarding intellectual property, reflecting community preferences for structured legal protection.

During our research, we also conducted interviews with five ambassadors from organizations such as Rubik Hub, Launch Community, Innovation Labs, and Orange Fab, known for their exceptional work in supporting startups in Romania. Based on their input, we designed a SOTA analysis of our system, the baseline being the systems currently in use in their ecosystems. The results are presented in Table 1.

Table 1. SWOT analysis of the SPARK-IT project.

Strengths	Weaknesses
• **Matchmaking:** The AI based recommendation system, which takes into account both professional information and reputation scores, is considered a strong point. • **IP protection:** The platform provides the opportunity to sign an NDA and stores that NDA in a secure environment. The system cannot access sensitive information, which builds trust. • **User Centric Design:** Feedback loops and iterative enhancements increase the chances for the system to succeed. • **Data-Driven Approach:** SPARK-IT combines quantitative metrics (KPIs) with qualitative feedback.	• **Documentation Overhead:** Increased workload from detailed process documentation was a concern among the stakeholders. The final documents that the mentor has to upload can be perceived as unnecessary overhead. • **Resource Intensiveness:** Requires additional staffing and technological investments. • **Unfamiliarity with Blockchain:** Users can find it difficult to use web3 wallets and blockchain. • **Long-Term Monitoring:** The platform does not provide the opportunity for long-term monitoring of a project after a collaboration with a mentor.
Opportunities	**Threats**
• **Innovative Business Model:** The system innovates current approaches and relatively few competitors have been identified. • **Partnerships:** The project has a great potential to form alliances with academic institutions, industry leaders, and venture capital firms. • **Multiple Revenue Streams:** There are multiple monetization models that could be implemented, such as subscriptions, premium services, and data analytics offerings. • **Open-Source:** Open-source initiatives are generally regarded positively by the community.	• **Adoption Resistance:** Stakeholders might not be prepared to use the technological stack. • **Budgetary Constraints:** Limited resources could restrict technology implementation. • **Risk of Over-Documentation:** Excessive focus on documentation may hinder creative engagement.

4.3.2. Scalability and Cost Efficiency

SPARK-IT also prioritizes system efficiency and scalability. Transaction throughput is tracked by measuring the number of mentorship requests, proposal submissions, and confirmed collaborations processed per second, ensuring the platform remains responsive under high demand. System latency is continuously analyzed to maintain sub-second re-

sponse times for matchmaking queries, proposal evaluations, and encrypted data retrieval from decentralized storage. Additionally, blockchain-related operations, such as executing smart contracts for mentor compensation and verifying non-disclosure agreements (NDAs), are optimized to minimize processing delays. Together, these metrics provide a comprehensive assessment of SPARK-IT's reliability, scalability, and impact. This ensures it effectively meets the needs of innovators and experts in a competitive innovation environment.

Using our own permissioned blockchain has the significant advantage of eliminating gas fees entirely. On a public blockchain network like Ethereum, however, a complete interaction between an innovator and an expert, comprising the creation of a business project, proposal submission, NDA signing, expert association, reward token locking and disbursement, reputation token transfer, and wrapped token bridge operations, would incur substantial transaction costs. Under the current network condition, the average gas price is 40 Gwei and ETH is at approximately USD 1600 at the time of writing. One ETH is equivalent to 1 billion Gwei (gigawei) and Wei represents the smallest unit of Ether (ETH). For one complete innovator–expert interaction there is a gas estimate of 455,000, which results, on Ethereum Mainnet, in an estimated cost of USD 29 per collaboration. Even when using a Layer 2 solution like Optimism, which reduces fees, the cost would still amount to approximately USD 0.23 per interaction. In contrast, the SPARK-IT platform's permissioned architecture enables cost-free transactions for users, removing a key barrier to adoption and making the platform particularly suitable for large-scale, inclusive innovation ecosystems.

Although operating our own blockchain nodes does incur infrastructure and maintenance costs, it offers greater control, eliminates unpredictable gas expenses, and ensures compliance with privacy and governance requirements.

4.3.3. API Performance Metrics

Table 2 presents the average response times for key backend operations within the SPARK-IT platform. These include user authentication, profile management, business project and proposal handling, and interactions with decentralized storage. The recorded times demonstrate that most API endpoints perform efficiently, with the majority of requests completed in under 200 ms, ensuring a responsive user experience even under typical operational loads.

Table 2. Response times for SPARK-IT platform operations (in ms).

Description	Method	Endpoint URL [a]	Duration (ms)
Access the account information	GET	/api/account	13
Authenticate	POST	/api/authenticate	164
Obtain the user profile	GET	/api/user-profiles/me	46
Update the user profile	PUT	/api/user-profiles/me	17
Create a business project	POST	/api/bp	148
Obtain the list of business projects	GET	/api/bp/me	19
Obtain the details of a business project	GET	/api/bp/{id}	23
Create a proposal for collaboration	POST	/api/bp/{id}/pfc	113
Obtain the list of proposals for collaboration	GET	/api/bp/{id}/pfc	17
Obtain the details referring to proposal	GET	/api/bp/{id}/pfc/{pid}	38
Various operations performed on a proposal	POST	/api/bp/{id}/pfc/{pid}/*	27
Obtain the list of found experts	GET	/api/bp/{id}/pfc/{pid}/experts-found	101
Obtain the list of collaborations	GET	/api/collaborations/me	33
Obtain a resource from IPFS	GET	/api/ipfs/cid/{cid}	11

[a] `bp` is shorthand for `business project`, and `pfc` is for `proposal-for-collaboration`, * the operations that are performed on a proposal are: `select-expert`, `mentorship-decision`, `sent-to-mentor`, `nda-skip`, `final-report`, `rating-submitted`, `mentor-paid`.

5. Conclusions

The SPARK-IT platform introduces a framework that addresses challenges within innovation ecosystems by combining blockchain infrastructure, AI-driven expertise matching,

and decentralized governance models. The system builds a complete business innovation ecosystem on a single digital platform. It improves transparency and participation for innovators, entrepreneurs, mentors, experts, and investors. Furthermore, the system removes the barriers related to intellectual property handling, uneven access to mentorship, and difficult collaboration processes.

The research shows several key advantages. Firstly, AI-based algorithms guide innovators toward experts whose skills align with project needs, improving the quality of mentorship and the value of outcomes. Secondly, blockchain technology, together with the legally compliant NDA template provided in the platform, preserves a reliable record of intellectual property exchanges. By using a permissioned network, with nodes hosted at trusted universities and with zero gas fees, we eliminate the typical barriers of traditional public chains, encouraging broad participation. Thirdly, token-based incentives keep users engaged. Innovators and experts earn reputation tokens based on performance and contributions. SparkCoins, which are linked to fiat currency, further encourage knowledge sharing and long-term participation. Additionally, the integrated web scraper provides a simple method to create user profiles from trusted sources, and the encryption module ensures that only authorized parties can access sensitive data. To empower the community further, the platform allows mentors who wish to give back to waive their fees, facilitating access for innovators who may not have the financial means; this allows for building a truly inclusive ecosystem for collaboration and growth.

By using a modular, scalable architecture and focusing on usability, SPARK-IT can be adapted to multiple contexts. These may include startup accelerators who want to improve the success rate of mentorship, academic groups seeking to improve research collaborations, and investors looking for emerging opportunities. The design also makes room for integration with data analytics, decision-support systems, and predictive modeling, enabling the platform to evolve alongside changes in markets and technologies.

As SPARK-IT advances toward practical deployment, several areas warrant sustained effort. Enhancements in AI algorithms will improve the accuracy and applicability of matches, ensuring that participants gain practical guidance. Revisions to incentive structures and reward mechanisms will help maintain an active community and encourage value exchange among all users. Ongoing work on scaling the platform, both technically and organizationally, will accommodate a growing user base, a wider range of industries, and more extensive use cases.

Future research and development may broaden the platform's scope to include additional sectors such as manufacturing, healthcare, finance, and sustainability-focused fields. Enabling the possibility for a user to join a decentralized autonomous organization and to be able to influence its development through a transparent voting system will trigger an active and engaged community. Incorporating advanced analytics and structured decision-making processes will help participants identify trends, forecast opportunities, and make more informed choices. SPARK-IT expands access to expert networks and protects intellectual assets. It promotes balanced participation across regions and communities. This supports innovation-led growth and benefits society.

In closing, SPARK-IT's approach to resolving ecosystem challenges not only improves and sparks connections between innovators and a larger entrepreneurial arena but also creates a basis for new activities, spin-offs, and cross-sector partnerships. The work presented here provides a solid foundation for refining, scaling, and enriching the platform, paving the way for its ongoing role in encouraging global innovation and societal advancements.

Author Contributions: Conceptualization, A.A. (Adrian Alexandrescu), D.-E.B. and C.N.B.; methodology, A.A. (Adrian Alexandrescu), D.-E.B., C.N.B. and G.-A.S.; software, A.A. (Adrian Alexandrescu), D.-E.B., C.N.B., G.-A.S. and S.-D.P.; validation, A.A. (Adrian Alexandrescu), D.-E.B., C.N.B., G.-A.S., S.-D.P., A.A. (Alexandru Archip) and C.M.; formal analysis, D.-E.B. and C.N.B.; investigation, A.A. (Adrian Alexandrescu), D.-E.B., C.N.B., G.-A.S., S.-D.P., A.A. (Alexandru Archip) and C.M.; resources, A.A. (Adrian Alexandrescu), D.-E.B. and C.N.B. ; data curation, A.A. (Adrian Alexandrescu), D.-E.B., C.N.B. and G.-A.S.; writing—original draft preparation, A.A. (Adrian Alexandrescu), D.-E.B., C.N.B., G.-A.S., S.-D.P., A.A. (Alexandru Archip) and C.M.; writing—review and editing, A.A. (Adrian Alexandrescu), D.-E.B. and C.N.B.; visualization, A.A. (Adrian Alexandrescu), D.-E.B. and C.N.B.; supervision, A.A. (Adrian Alexandrescu); project administration, A.A. (Adrian Alexandrescu); funding acquisition, A.A. (Adrian Alexandrescu). All authors have read and agreed to the published version of the manuscript.

Funding: The SPARK-IT project described in this paper has received funding from the Next Generation Internet Initiative (NGI) within the framework of the "NGI TrustChain project", which was funded by the European Union's Horizon Europe research and innovation programme under grant agreement No. 101093274.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ABAC	Attribute-Based Access Control
ACC	Atomicity, Consensus, and Confidentiality
AI	Artificial Intelligence
B-RBAC	Blockchain-based Role-Based Access Control
BERT	Bidirectional Encoder Representations from Transformers
BMC	Business Model Canvas
CAAC	Context-Aware role-based Access Control
CBC	Consortium Blockchain
CNN	Convolutional Neural Network
CP-ABE	Ciphertext-Policy Attribute-Based Encryption
DAO	Decentralized Autonomous Organization
DDAC	Decentralized Data Access Control
DHT	Distributed Hash Table
DNN	Deep Neural Network
DSS	Decentralized Storage System
FAIR	Findability, Accessibility, Interoperability and Reusability
IP	Intellectual Property
IPFS	InterPlanetary File System
IoT	Internet of Things
KPI	Key Performance Indicator
LDA	Latent Dirichlet Allocation
LSTM	Long Short-Term Memory
MRR	Mean Reciprocal Rank
NDA	Non-Disclosure Agreement
NLP	Natural Language Processing
OBAC	Ontology-Based Access Control
PRE	Proxy re-Encryption
PoS	Proof-of-Stake
PoW	Proof-of-Work
RBAC	Role-Based Access Control

RFP	Requests for Proposal
S-BERT	Sentence-BERT
SA-ODAC	Security-Aware mechanism and Ontology-based Data Access Control
SAT	Secure Awareness Technique
SBFT	Simplified Byzantine Fault Tolerance
TF-IDF	Term Frequency-Inverse Document Frequency
VPC	Value Proposition Canvas

References

1. Osterwalder, A.; Pigneur, Y. *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*, 1st ed.; John Wiley and Sons: Hoboken, NJ, USA, 2010.
2. Joyce, A.; Paquin, R.L. The triple layered business model canvas: A tool to design more sustainable business models. *J. Clean. Prod.* **2016**, *135*, 1474–1486. [CrossRef]
3. Shanbhag, N.; Pardede, E. The Blitz Canvas: A Business Model Innovation Framework for Software Startups. *Systems* **2022**, *10*, 58. [CrossRef]
4. Dorantes-Gonzalez, D.J. A novel business model frame for innovative startups. *Pressacademia* **2017**, *4*, 126–137. [CrossRef]
5. Keiningham, T.; Aksoy, L.; Bruce, H.L.; Cadet, F.; Clennell, N.; Hodgkinson, I.R.; Kearney, T. Customer experience driven business model innovation. *J. Bus. Res.* **2020**, *116*, 431–440. [CrossRef]
6. Fritscher, B.; Pigneur, Y. Extending the Business Model Canvas—A Dynamic Perspective. In Proceedings of the Fifth International Symposium on Business Modeling and Software Design—BMSD, Milan, Italy, 6–8 July 2015; INSTICC, SciTePress: Setúbal, Portugal, 2015; pp. 86–95. [CrossRef]
7. Azmy, A.; Wiadi, I.; Risza, H. Product Value Creation Training Through Value Proposition Canvas (VPC) with the South Jakarta Small Medium Enterprise (SME) Community. *J. Pengabdi. Pada Masy.* **2023**, *8*, 834–845. [CrossRef]
8. Garg, P.; Rani, R.; Miglani, S. Mining Professional's Data from LinkedIn. In Proceedings of the 2015 Fifth International Conference on Advances in Computing and Communications (ICACC), Kochi, India, 2–4 September 2015; pp. 98–101. [CrossRef]
9. Yi, L.; Yuan, R.; Long, S.; Xue, L. Expert Information Automatic Extraction for IOT Knowledge Base. *Procedia Comput. Sci.* **2019**, *147*, 288–294. [CrossRef]
10. Bondielli, A.; Marcelloni, F. A Data-Driven Approach to Automatic Extraction of Professional Figure Profiles from Résumés. In *Intelligent Data Engineering and Automated Learning—IDEAL 2019*; Series Title: Lecture Notes in Computer Science; Yin, H., Camacho, D., Tino, P., Tallón-Ballesteros, A.J., Menezes, R., Allmendinger, R., Eds.; Springer International Publishing: Cham, Switzerland, 2019; Volume 11871, pp. 155–165. [CrossRef]
11. Lade, S.; Billade, A.; Chandrapatle, A.; Chenna, S.; Chinchalpalle, G. LinkedIn Alumni Profile Data Extraction. In Proceedings of the 2024 4th International Conference on Pervasive Computing and Social Networking (ICPCSN), Salem, India, 3–4 May 2024; pp. 174–178. [CrossRef]
12. Lops, P.; De Gemmis, M.; Semeraro, G.; Narducci, F.; Musto, C. Leveraging the linkedin social network data for extracting content-based user profiles. In Proceedings of the Fifth ACM Conference on Recommender Systems, Chicago, IL, USA, 23–27 October 2011; pp. 293–296. [CrossRef]
13. Wu, J.; Yang, G. An Ontology-Based Method for Project and Domain Expert Matching. In Proceedings of the Fuzzy Systems and Knowledge Discovery, Yantai, China, 10–12 August 2010; Wang, L., Jin, Y., Eds.; Springer: Berlin, Heidelberg, 2005; pp. 176–185. [CrossRef]
14. Musen, M.A. The protégé project: A look back and a look forward. *AI Matters* **2015**, *1*, 4–12. [CrossRef]
15. Han, S.; Richie, R.; Shi, L.; Tsui, F. Automated Matchmaking of Researcher Biosketches and Funder Requests for Proposals Using Deep Neural Networks. *IEEE Access* **2024**, *12*, 98096–98106. [CrossRef]
16. Roy, P.K.; Chowdhary, S.S.; Bhatia, R. A Machine Learning approach for automation of Resume Recommendation system. *Procedia Comput. Sci.* **2020**, *167*, 2318–2327. [CrossRef]
17. Rus, G.; Vaida, C.; Gherman, B.; Pisla, A.; Nae, L.; Ciupe, M.; Pisla, D. On the Development and Validation of a Matchmaking Mentoring Platform. *Acta Tech. Napoc. Ser. Appl. Math. Mech. Eng.* **2023**, *66*, 193–198. Available online: https://atna-mam.utcluj.ro/index.php/Acta/article/view/2230 (accessed on 23 January 2025).
18. Kayes, A.S.M.; Rahayu, W.; Dillon, T.; Chang, E. Accessing Data from Multiple Sources Through Context-Aware Access Control. In Proceedings of the 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing and Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), New York, NY, USA, 1–3 August 2018; pp. 551–559. [CrossRef]

19. Panisson, A.R.; Ali, A.; McBurney, P.; Bordini, R.H. Argumentation Schemes for Data Access Control. In Proceedings of the 7th International Conference on Computational Models of Argument (COMMA) in Frontiers in Artificial Intelligence and Applications, Warsaw, Poland, 12–14 September 2018; Volume 305, pp. 361–368. [CrossRef]
20. Brewster, C.; Nouwt, B.; Raaijmakers, S.; Verhoosel, J. Ontology-based Access Control for FAIR Data. *Data Intell.* **2020**, *2*, 66–77. [CrossRef]
21. Kiran, G.M.; Nalini, N. Enhanced security-aware technique and ontology data access control in cloud computing. *Int. J. Commun. Syst.* **2020**, *33*, e4554. . [CrossRef]
22. Chen, Y.; Chen, S.; Liang, J.; Feagan, L.W.; Han, W.; Huang, S.; Wang, X.S. Decentralized data access control over consortium blockchains. *Inf. Syst.* **2020**, *94*, 101590. [CrossRef]
23. Ding, Y.; Feng, L.; Qin, Y.; Huang, C.; Dong, P.; Gao, L.; Tan, Y. Blockchain-based Access Control Mechanism of Federated Data Sharing System. In Proceedings of the 2020 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), Exeter, UK, 17–19 December 2020; pp. 277–284. [CrossRef]
24. Sun, B.; Dang, Q.; Qiu, Y.; Yan, L.; Du, C.; Liu, X. Blockchain Privacy Data Access Control Method Based on Cloud Platform Data. *Int. J. Adv. Comput. Sci. Appl.* **2022**, *13*, 10–17. [CrossRef]
25. Gazsi, J.S.; Zafreen, S.; Dagher, G.G.; Long, M. VAULT: A Scalable Blockchain-Based Protocol for Secure Data Access and Collaboration. In Proceedings of the 2021 IEEE International Conference on Blockchain (Blockchain), Melbourne, Australia, 6–8 December 2021; pp. 376–381. [CrossRef]
26. Ali, S.; Wang, G.; White, B.; Cottrell, R.L. A Blockchain-Based Decentralized Data Storage and Access Framework for PingER. In Proceedings of the 2018 17th IEEE International Conference On Trust, Security And Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE), New York, NY, USA, 1–3 August 2018; pp. 1303–1308. [CrossRef]
27. Mittal, S.; Ghosh, M. A three-phase framework for secure storage and sharing of healthcare data based on blockchain, IPFS, proxy re-encryption and group communication. *J. Supercomput.* **2024**, *80*, 7955–7992. [CrossRef]
28. Butincu, C.N.; Alexandrescu, A. Blockchain-Based Platform to Fight Disinformation Using Crowd Wisdom and Artificial Intelligence. *Appl. Sci. Basel* **2023**, *13*, 6088. [CrossRef]
29. Alexandrescu, A.; Butincu, C.N. Decentralized News-Retrieval Architecture Using Blockchain Technology. *Mathematics* **2023**, *11*, 4542. [CrossRef]
30. Alexandrescu, A.; Butincu, C.N. DARS: Decentralized Article Retrieval System. *SoftwareX* **2024**, *25*, 101624. [CrossRef]
31. Bărbuţă, D.E.; Alexandrescu, A. A Secure Real Estate Transaction Framework Based on Blockchain Technology and Dynamic Non-Fungible Tokens. In Proceedings of the 2024 28th International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, Romania, 10–12 October 2024; pp. 558–563. [CrossRef]
32. Bărbuţă, D.E.; Alexandrescu, A.; Tărniceriu, D.; Gavrilă, M. Leveraging Blockchain to Enhance the Efficiency and Data Integrity of Systems Monitoring Drivers' Sobriety. In Proceedings of the 2024 23rd RoEduNet Conference: Networking in Education and Research (RoEduNet), Bucharest, Romania, 19–20 September 2024; pp. 1–6. [CrossRef]
33. Han, R.; Yan, Z.; Liang, X.; Yang, L.T. How Can Incentive Mechanisms and Blockchain Benefit with Each Other? A Survey. *ACM Comput. Surv.* **2022**, *55*, 1–38. [CrossRef]
34. Lisi, A.; De Salve, A.; Mori, P.; Ricci, L.; Fabrizi, S. Rewarding reviews with tokens: An Ethereum-based approach. *Future Gener. Comput. Syst.* **2021**, *120*, 36–54. [CrossRef]
35. Merrill, P.; Austin, T.; Thakker, J.; Park, Y.; Rietz, J. Lock and Load: A Model for Free Blockchain Transactions through Token Locking. In Proceedings of the 2019 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON), Newark, CA, USA, 4–9 April 2019; pp. 19–28. [CrossRef]
36. Cong, L.W.; Li, Y.; Wang, N. Token-based platform finance. *J. Financ. Econ.* **2022**, *144*, 972–991. [CrossRef]
37. Chen, K.; Fan, Y.; Liao, S.S. Token Incentives in a Volatile Crypto Market: The Effects of Token Price Volatility on User Contribution. *J. Manag. Inf. Syst.* **2023**, *40*, 683–711. [CrossRef]
38. Kuo Chuen, D.L.; Li, Y.; Xu, W. Rewarding Honesty: An Incentive Mechanism to Promote Trust in Blockchain-Based E-commerce. *J. Br. Blockchain Assoc.* **2023**, *6*, 1–5. [CrossRef]
39. Saito, K.; Iwamura, M. How to make a digital currency on a blockchain stable. *Future Gener. Comput. Syst.* **2019**, *100*, 58–69. [CrossRef]
40. Mita, M.; Ito, K.; Ohsawa, S.; Tanaka, H. What is Stablecoin?: A Survey on Price Stabilization Mechanisms for Decentralized Payment Systems. In Proceedings of the 2019 8th International Congress on Advanced Applied Informatics (IIAI-AAI), Toyama, Japan, 7–11 July 2019; pp. 60–66. [CrossRef]
41. Asgaonkar, A.; Krishnamachari, B. Solving the Buyer and Seller's Dilemma: A Dual-Deposit Escrow Smart Contract for Provably Cheat-Proof Delivery and Payment for a Digital Good without a Trusted Mediator. In Proceedings of the 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), 14–17 May 2019, Seoul, Republic of Korea; pp. 262–267. [CrossRef]

42. Schwartzbach, N.I. An Incentive-Compatible Smart Contract for Decentralized Commerce. In Proceedings of the 2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Sydney, Australia, 3–6 May 2021; pp. 1–3. [CrossRef]
43. Yoo, S. How to Design Cryptocurrency Value and How to Secure Its Sustainability in the Market. *J. Risk Financ. Manag.* **2021**, *14*, 210. [CrossRef]
44. Seewald, A.K.; Ghete, M.; Wernbacher, T.; Platzer, M.; Schneider, J.; Hofer, D.; Pfeiffer, A.K. Cycle4Value: A Blockchain-based Reward System to Promote Cycling and Reduce CO_2 Footprint. In Proceedings of the 13th International Conference on Agents and Artificial Intelligence—Volume 2: ICAART, Online Streaming, 4–6 February 2021; INSTICC, SciTePress: Setúbal, Portugal, 2021; pp. 1082–1089. [CrossRef]
45. Dennis, R.; Owen, G. Rep on the block: A next generation reputation system based on the blockchain. In Proceedings of the 2015 10th International Conference for Internet Technology and Secured Transactions (ICITST), London, UK, 14–16 December 2015; pp. 131–138. [CrossRef]
46. Zhou, Z.; Wang, M.; Yang, C.N.; Fu, Z.; Sun, X.; Wu, Q.J. Blockchain-based decentralized reputation system in E-commerce environment. *Future Gener. Comput. Syst.* **2021**, *124*, 155–167. [CrossRef]
47. Lee, S.; Seo, S.H. Design of a Two Layered Blockchain-Based Reputation System in Vehicular Networks. *IEEE Trans. Veh. Technol.* **2022**, *71*, 1209–1223. [CrossRef]
48. Bellini, E.; Iraqi, Y.; Damiani, E. Blockchain-Based Distributed Trust and Reputation Management Systems: A Survey. *IEEE Access* **2020**, *8*, 21127–21151. [CrossRef]
49. Bhatia, G.K.; Gupta, S.; Dubey, A.; Kumaraguru, P. WorkerRep: Immutable Reputation System For Crowdsourcing Platform Based on Blockchain. *arXiv* **2020**, arXiv:2006.14782.
50. Li, M.; Zhu, L.; Zhang, Z.; Lal, C.; Conti, M.; Alazab, M. A nonymous and Verifiable Reputation System for E-Commerce Platforms Based on Blockchain. *IEEE Trans. Netw. Serv. Manag.* **2021**, *18*, 4434–4449. [CrossRef]
51. Fu, S.; Huang, X.; Liu, L.; Luo, Y. BFCRI: A Blockchain-Based Framework for Crowdsourcing with Reputation and Incentive. *IEEE Trans. Cloud Comput.* **2023**, *11*, 2158–2174. [CrossRef]