A Blockchain-Empowered Document Notarization Service

Vincenzo Aiello
University of Salerno
Salerno, Italy
v.aiello14@studenti.unisa.it

Franco Cirillo
University of Salerno
Salerno, Italy
f.cirillo30@studenti.unisa.it

Christian Esposito
University of Salerno
Salerno, Italy
esposito@unisa.it

Ciro Fusco
University of Salerno
Salerno, Italy
c.fusco19@studenti.unisa.it

Chang Choi
Gachon University
Seongnam, Rep. of Korea
pcatalano@thereload.it

ABSTRACT

The recent digitisation process carried our in various application domains have caused the proliferation of digital documents, which are easier to be tampered and altered than the paper ones. This is calling out for a proper notarisation service to certify the authenticity and integrity of digital documents. The possible security and privacy issues related to the traditional centralised realisation of such a service require the usage of decentralised design, which has the added value to also allow handling the increasing number of requests, offering high availability and resiliency degrees, and removing third trusted parties able to profile users' activities. This paper presents the design and implementation of such a service based on the blockchain service and related technologies.

KEYWORDS

Knowledge Management, Document Notarisation, Blockchain, Information Retrieval

ACM Reference Format:
Vincenzo Aiello, Franco Cirillo, Christian Esposito, Ciro Fusco, and Chang Choi. 2022. A Blockchain-Empowered Document Notarization Service. In *Proceedings of October 19-22, 2022 (SMA 2022).* ACM, New York, NY, USA, 6 pages. https://doi.org/xx.xxx/xxx_x

1 INTRODUCTION

Historically, the notarisation of a document is a procedure carried out by a specific person in charge, commonly the notary, who guarantees the authenticity, integrity and proof-of-existence of an agreement between two or more parties, represented by a proper paper-based document. The notary by means of a notarial deed ensures the validity of an agreement by ensuring that the arrangements being made derive from an effective consent between the involved parties, and the agreement terms agreed upon the involved parties are the ones in the presented document. Notarisation, in practice, certifies the integrity and immutability of the information over time, but not necessarily if all the information entered is true. In fact, technology only allows information to be recorded in the form of a string of numbers and letters that identify the document. For these reasons, recording information on one or more distributed registers is not the same as certifying the veracity of the same. We therefore speak of notarisation and not certification in this paper. This is an act that has the function of delegating to technology the only things it can guarantee the presence of the information and its immutability over time. With the advent of the digital technology, there is the need to automate, digitise, but above all decentralise this process [14]. This is extremely demanding if we consider that nowadays the majority of documents in the various application domains are digital, which are extremely simple to be tampered/altered by malicious entities. The practice of notarisation is represented in Figure 1, which is made of two phases: the registration, when the original document expressing an agreement is given to the notary, and verification, when a document is given to the notary so as to be compared with the registered one and checked if they match or not. The notary has a proper archive of all the registered documents, and signs all the registered documents.

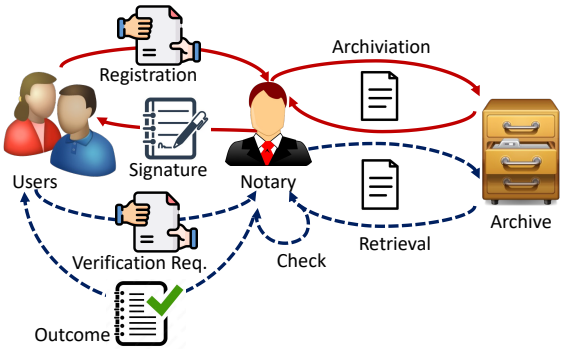

Figure 1: Classic scheme for document notarisation

A naive solution to attempt the digitisation of this process would be to store a copy of the document within a database, so that a notary would be able to check a document authenticity by checking it against the stored copy. This is the digital version of the above mentioned figure, where documents are not paper-based but in their digital version and the archive is a database running a server or in the cloud. Despite being simple, such a solution has the issue of requiring a centralised notary service, and the document database should be properly secured as it may be involved in various attack attempts to violate and compromise it. If the notary disposes only of

SMA 2022, Saipan, USA,

ACM ISBN 978-1-4503-6843-8/20/10.
https://doi.org/xx.xxx/xxx_x

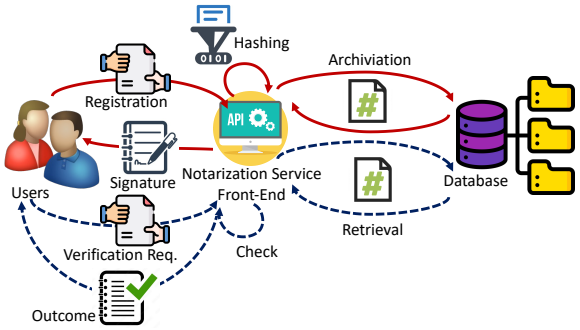

Figure 2: Hash-based scheme for document notarisation

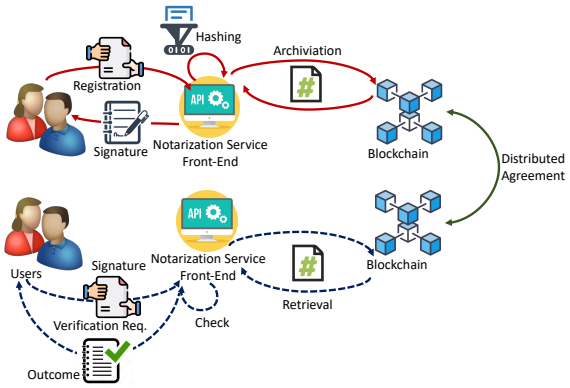

Figure 3: Decentralised scheme for document notarisation

the hash of the document, instead of the overall document, it would be more suitable to preserve privacy; however, it will miss some valuable details useful for checking various versions of the same document. Such a solution is depicted in Figure 2, where the system exposes a web front-end so as to be directly used by the users, with or without the intervention of a notary (which is a public official to whom the State entrusts the value of legal proof to the acts it stipulates). Such a solution is still centralised and vulnerable to attacks, both at the web front-end and its back-end.

An alternative solution would be to device a more decentralised approach, where various agents, running at different premises of a given organisation, may be involved in the checking of documents provided by users, which are able to register a document at an agent and verify it at another one. This is a suitable approach for large-scale organisations such as public administrations and companies, where a birth certificate issues at a municipality can be verified by another one, or a deed of sale done by an office in a country can be notarised by an office in a different country, just to give some examples. An enabling technology for such a solution is represented by the blockchain [10], which consists of blocks holding pieces of information being linked among each others by having a block containing the hash of the next block (consisting in an append-only distributed ledger) and being approved by running distributed consent among the involved participants to the blockchain. The usage of distributed consent and hashing offers a high degree of tamper-resistance, robustness and effectiveness required by the envisioned solution for notarisation. In fact, such a technology offers the possibility of recording any type of information on a distributed ledger. The information entered cannot be changed and, depending on the case, can be consulted by anyone. The applied consistency model by blockchain allows the validity of the data hold by different agents spread across the network. Registering any digital document in a distributed register allows you to have a non-editable and non-counterfeit digital artefact, registered on a platform, which guarantees the authenticity of the document. In the case of private documents, it is not a good idea to have the plain-text document in the blockchain, so an encrypted or hashed version may be preferable. Thanks to the approval of a specific law in Italy on 11th February 2019, the information deposited using technologies based on distributed registers have legal value, and the EU state members have similar laws in place. This is boosting the application of blockchain within document management in various domains and making our proposed notarisation service potentially impacting notarisation within the public administrations and among industries. The limited size of the blocks and data that can be stored within them makes unsuitable the blockchain to have copies of digital documents, so we have coupled it with a decentralised peer-to-peer data storage solution represented by InterPlanetary File System (IPFS) [3], which is a communication protocol and a peer-to-peer network for storing and sharing data in a content-addressable distributed file system. This allows our solution to be able also to return the complete document and not only to verify its authenticity. Such retrieval operation is a crucial point to allow a good usability of the proposed solution, and presents a novelty with respect to the available solutions: not only by an identifier a document can be queried and retrieved, but also with more complex predicates. We have integrated an automatic classification approach for documents [1], so that at the registration phase a document has a set of terms being assigned. So, the user can express one or more categories so as to have all related documents presented and he/she can choose which one wants to obtain. A proper authentication and authorisation solution has been implemented by defining roles and privileges for each of them on the operations they can do within the system.

The rest of this paper is structured as follows. In Section 2, the main related works are presented and compared against the proposed solution, which is described in details in the Section 3. We present the implemented solution in Section 4, and conclude with some lesson learnt in Section 5.

2 RELATED WORKS

In this section, a brief background will be provided on the approaches existing in the literature related to the notarisation of documents and the verification of their authenticity, using blockchain technology.

The solution proposed in [13] consists of a decentralised application offering both the functionalities for notarisation and anti-plagiarism. The proposed decentralised app prototype includes a front-end, which uses a service for the management of notarial operations, a service for plagiarism operations and a service for the generation of Non Fungible Token (NFT) [8] to facilitate the transfer of ownership process. In a usage scenario, the system acquires

one or more files to be authenticated by the user. Before the actual authentication, a plagiarism check is performed on the files sent. At this point, the decentralised app starts two separate processes. One process is used to create the fingerprint (hash) and a separate process is used to perform file fragmentation and store them in decentralised storage. The creation of the hash from the files sent is performed through the use of the SHA256 encryption algorithm, which, in addition to guaranteeing very high cryptographic security standards, allows a quick identification of any transaction on the blockchain network. The file hash is recorded on three different blockchains to ensure high reliability in case of saturation and/or collapse of a blockchain network. Once having stored the hash on at least one of the blockchain networks, the process of file fragmentation and storage begins. This process consists of encrypting the files and dividing them into N different segments. Each segment is then encrypted with N different randomised encryption keys. Finally, each encrypted segment is distributed on the storage of the decentralised nodes. The proposed solution then generates the certificate of authenticity of the files sent, containing the previously generated hash and the transaction on the blockchain used. A total of three certificates of authenticity will be generated, one for each of the three blockchains used. At this point, the proposed solution will allow the user, starting from the hash in his possession, to retrieve the notarial files, reconstructing the structure present within the decentralised nodes. In case it is necessary to transfer the ownership of a document to another user, the document will be transformed into an NFT according to the standard and will be transferred to the new user.

In [11], a solution is proposed to guarantee the originality and authenticity of posted digital content, such as books and documents. This solution is based on the use of IPFS and the smart contracts of the Ethereum blockchain. IPFS is used to store digital content in a decentralised and distributed manner, publicly and globally accessible by all through the use of IPFS hashes. This IPFS hash is used by the Ethereum smart contract to ensure integrity, originality and authenticity. The hash value remains the same if the content of the document or book remains intact. If content alteration occurs during the publishing stages, the IPFS hash for the book changes and therefore would not match the hash stored in the smart contract. Therefore, each participating entity can go back in the process and verify the authenticity of the document and make sure it is a legitimate copy. The information published on the blockchain is visible, on the basis of the registry use policy, at least to all participants in the consent. This feature affects the privacy of the user who enters their data.

The paper [2] proposes an integration between Blockchain and Personal Data Store (PDS) [6]. PDS allows a user to share private information in a completely private way, and also to manage the various authorisations on who can or cannot access the data and which can be accessed or not. In particular, the user uses the Blockchain to certify the authenticity of the information released while PDS guarantees total control by the owner over the visibility of their data.

In [9], a solution based on the NFB protocol is proposed that allows the notarisation of files on the blockchain. It is based on the use of three separate modules. The first defines all the functions that allow reading and writing within the blockchain through the use of the smart contract and the sending of transactions. The second module called OKORO allows the user to manage and archive documents through a centralised solution. The third module acts as a gateway to orchestrate the operations between the blockchain and the OKORO module. This solution ensures

- Proof of Archivability. When the document is uploaded, its SHA512 is calculated, it is stored on OKORO and a transaction is performed on the blockchain to save other information about the document.
- Proof of Retrievability. The user provides the key to search for the document. A link is then established with the OKORO layer to retrieve the archived document. At the same time a link is established with the blockchain to retrieve the document information.
- Proof of Existence. The user uploads the document. The document hash is calculated. The hash is searched within the blockchain. Information about the requested file is retrieved and then it is searched in OKORO.

These works some important features, such as user authentication, which is a fundamental and highly relevant topic when it comes to certification of notary documents. The method of accessing the system cannot be a secondary topic, but must be well studied because it is a pillar for the notarisation activity. Only the authorised user must be able to access the services offered because their operations will be signed by him, so you must try to limit intrusions as much as possible. Another fundamental point not present in the analysed literature is the possibility of being able to carry out a semantic search among the loaded documents. This could be of great help to users who wish to search through their uploaded documents, without the need to retrieve all of them entirely. The system must then be fast and guarantee relatively low waiting times, so it needs to be as less complex as possible. This was not found, for example, in the solution of the first article as it shows a certain redundancy of operations, which inevitably lead to slowdowns. It should then persist the documents on a non-centralised solution to guarantee the decentralisation property that underlies the design of the system itself.

3 PROPOSED SOLUTION

The proposed solution consists in the use of the permissioned blockchain offered by the Hyperledger Fabric, in order to make the unique identifier of the document to be notarised and its hash persistent and immutable, together with other information related to additional features requested by the client. So, the document will not be uploaded on the blockchain, but only the identifier (CID) obtained from IPFS, which will then allow the query in the system parameterised with the CID. The functions that have been developed are

- insertion of the document in the system by saving the document on IPFS and all related information (CID, hash, timestamp, signatories) on the blockchain;
- document search, which will be done by calculating the hash of the document provided and searching for the associated CID within the blockchain. Once found, we will search for content on IPFS;

- semantic search of documents, which will be done by specifying a category to which the searched documents belong;
- document certification, by calculating the hash, verifying that it is present in the blockchain and checking all related information.

The choice to use a permissioned blockchain was made because they are closer to the needs of companies. When a new data or record is added, the approval system is not tied to the majority of participants in the blockchain, but to a limited number of actors, defined as Trusted. This type of blockchain lends itself very well to our use. Moreover, it also responds to the need for a widespread update on several actors who can operate independently, but with limited control to those who are authorised and then allow to define special rules for access and visibility of all data, introducing the concept of Governance. Among these, Hyperledger Fabric was chosen because it has a highly modular and configurable architecture and supports smart contracts, which are called chaincodes, written in common programming languages, such as Go. The participants are not anonymous, this allows to obtain a model governance built on trust. As already mentioned, the documents will not be uploaded directly to the blockchain, this is because it is not a viable solution due to the complexity due to the size of the data. IPFS was chosen for the persistence of documents because it is a distributed system and therefore more robust, it uses the search by content based on the hash, thus ensuring the immutability of documents. The choice to use the hash also guarantees the uniqueness of the document, every time the system publishes a new file on IPFS, the network verifies through its hash if this is already present, automatically avoiding that multiple copies are kept. However, IPFS is a public content-addressable file system, so anyone with the CID or hash is able to retrieve the related stored document. In certain use cases where documents may contain sensitive infos, this is not a viable solution, but the problem can be easily resolved by using encryption. Basically, the document is encrypted before being stored on IPFS, so that only those with the relative decryption key is able to access the document. Such a solution is not an obstacle to the document certification, as the hash is hosted in the blockchain and this is the only element needed for the document verification.

The overall system implementation is divided into 3 logical levels, as shown in Figure 4: presentation, business and persistence layers, which respectively deal with the front-end to handle requests and provisioning of information to the user, definition of the application logic and management of persistence for the system data. The presentation layer consists of a single subsystem that defines the user interface for the various notarisation features. The business tier consists of two subsystems. One is for the Notarization, which manages the submission and verification of documents. The other one deals with the parameterised search of documents. They interact with the Blockchain, thus with the relative smart contracts, and the timestamper for the process of generating and verifying the Timestamp certificate. On the blockchain, these are the details being stored per each document: CID, Signatories, Document Hash, Timestamp, and a list of Categories. The persistence level is composed of the Data Access subsystem, which takes care of retrieving and saving files and related information on IPFS.

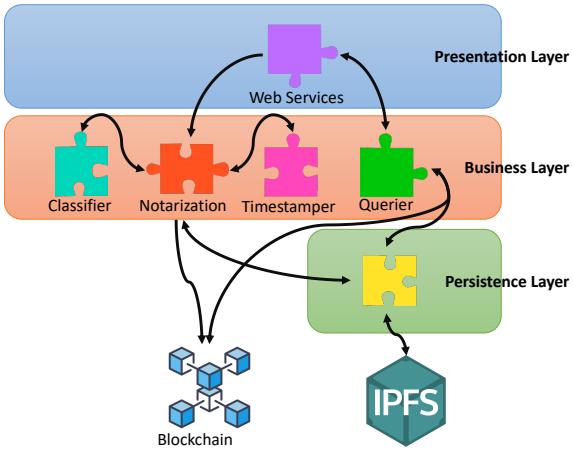

Figure 4: Architectural scheme of the proposed solution

The timestamp service is offered by an trusted party, called Time Stamp Authority (TSA), which allows associating a certain and legally valid date and time to a digital document. The affixing of the time stamp therefore allows you to establish the existence of an IT document starting from a certain instant and guarantees its validity over time. In addition, it certifies that the signer's ID is valid at the time of signing. Its use does not affect the content of the document and does not change its signature. Each document loaded into the system, therefore, needs to have assigned a timestamp at the time of submission, which can be checked whenever there is a need. This ensures that the date and time cannot be changed, so they are certainly authentic, an essential condition for the concept of notarisation. In this system, the timestamper interacts with the Notarisation subsystem through calls to the REST API. This implementation choice guarantees us a complete decoupling between our system and the one that generates the timestamps, thus leaving the possibility of changing the service provider at any time, thus not being bound to a single provider. For integration purposes in this version of the system, a separate project in Java has been created which generates timestamps through the BouncyCastle library. To respect the implementation paradigm stated above, this project exposes a REST endpoint on a predetermined port to which the Notarisation subsystem sends requests for the generation and control of certificates.

Classification, in a computational sense, tries to assign labels to data [5], given a set of characteristics. The classifier does this by drawing on knowledge derived from examples of how other objects have been tagged. These examples, called training data, serve as a source of preliminary knowledge that the classifier uses to make decisions about previously not considered objects. Categorisation is a specialisation of classification, and deals with assigning a category to an object. There are many different types of classification algorithms [7], and a distinctive feature is the output they produce. There are binary algorithms that produce two discrete results, such as a yes/no answer, about the belonging of an object to a class. Other algorithms support multiple results, producing a result from a discrete set of categories or a continuous value, such

as a floating point score or probability. Clustering boils down to grouping similar unlabelled documents based on some measure of similarity [5]. The goal is to split all documents in the collection that are similar in the same cluster to each other, while ensuring that dissimilar documents are in different clusters. Grouping can be applied to many different aspects of text, including words in a file or document, the documents themselves, or search results. Document clustering is typically performed as an offline batch process, and the output is typically a list of documents and a centre of gravity vector. In this work, we have used the OpenNLP library for natural language processing and categorisation of notarised documents. The Carrot2 library was used to create a document clustering engine. The solution was designed to be able to manage documents in PDF format using the Apache PDFBox library. A text summarization solution was implemented from scratch [4]. The classification of documents in Italian is carried out using the Naive Bayes algorithm offered by Apache OpenNLP, while the Lingo algorithm of Carrot2 was used for clustering. Specifically, a Naive Bayes document classification realises a super-visioned learning [15] and exploits the Bayes theorem to compute how probable it is that a document can be assigned to a given label, or set of labels. The "hypothesis" upon which the classifier has to reason is "the document fits into category C", where the "evidence" is the words W occurring in the document. If the probability of a document being classifiable with the category C_1 a given the presence of the words extracted from the document is greater than the probability of its being associated with category C_2, Naive Bayes returns category C_1. This is repeated for all the categories with which the classifier has been trained, and the one with highest probability is returned as a result of the classification. Last, a summary of a document was implemented as follows: after extracting the text from the PDF file using Apache PDFBox, the word frequency was considered, and the summary is made considering a limited number of sentences containing the words with higher frequency in the text. The Lingo algorithm is described at [12], and works as follows: for each document, text filtering is done by identifying the document's language, applying stemming, and marking stop words; then, frequent terms and phrases are discovered and Latent Semantic Indexing is exploited to discover abstract concepts by determining statistical co-occurrences of words that appear together in various documents. For each abstract concept, the best-matching phrase is found, and similar cluster labels are pruned. For each cluster label, Vector Space Model is used to determine the cluster contents by representing cluster descriptions as vectors of identifiers and applying proper operators such as the ones in the term frequency-inverse document frequency model; then, cluster merging is applied.

4 IMPLEMENTATION

We have implemented our solution my means of RESTful services exposed to the users as web pages. The users are required to authenticate themselves by leveraging on a two-phase authentication, as in Figure 5. At the first, the user perform a naive login request as described above. At first, a filter extracts the IP address of the requester that performed the login, and contact a proper service to obtain the user location. In our implementation we have used the GeoIP2 database provided by MaxMind [**?**], which allows to obtain city and country information from the IP Address. If the location is the one associated to the user when it has been created by the administrator, than the login can be executed, where the password obtained from the decryption of the second input to the request matches the one within the SQLite database [**?**] queried by using the first input as username. If there is a match, a random string is generated, and encrypted by using the server public key. Such a encrypted string is provided to the user by email by using the Java package javax.mail. When the user receives the email, he/she can make a login confirmation by providing its username and the received token, which is matched by the server with the one sent by email, and in the positive case the user receives in return the security token to request the other operation to the server.

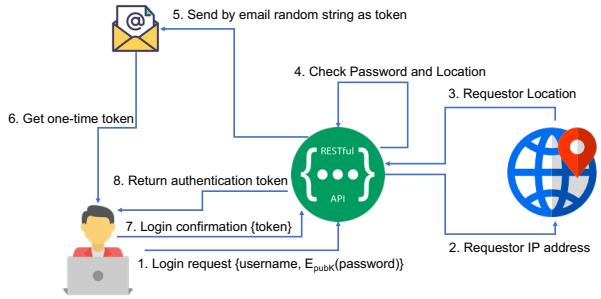

Figure 5: Two-step authentication execution.

We have implemented user management, where user details are kept in a database, but a state is associated to a user. At its creation by the administrator, the user is in an inactive state, and a one-time token is sent at the email address associated to the new user. The user has to confirm the creation by invoking a proper operation at the service and providing the received token. This allows the user to assume the active state, which allows to fully use those operations to which he/she is authorised. When there are three failed attempts to make a login by providing false information as input, the user is assumed to be compromised and passes to the suspected state. Only the administrator can unlock such a user and move it to the active state. While in the suspected state, the user is not able to request any operation, even if providing legitimate and valid inputs as login or a valid security token. The core of this solution is the security tokens issued during the login and generated/verified by using the JSON Web Tokens (JWT) [**?**], with JWT.IO library in Java [**?**]. Such a library is used for the token generation at the end of the login, where a temporal validity is associated to the token, and for the verification of provided tokens within a HTTP security filter for the authorisation. In fact, the filter obtains the username as output of the token verification (if it has not been expired), access to the user database and obtain the role associated to the user. If the requested operation falls within the allowed ones for the specific role, than the request is granted and the operation executed. This makes our solution as a series of two filters preceding the execution of the operation, in a way that the authentication and authorisation logic can change independently without affecting the other code

After being successfully registered, a user can upload a document as shown in Figure 6, by loading a PDF file from its computer's file system. Our solution will upload it on IPFS, and load in the

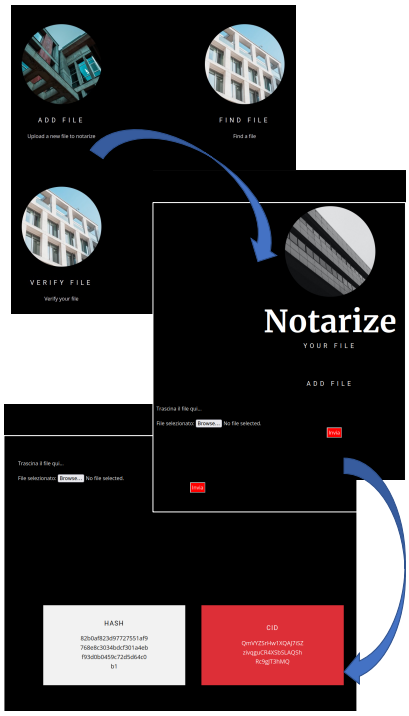

Figure 6: Document Registration

Figure 7: Document Verification

blockchain the relevant details, and presents to the user the hash and the CID released by IPFS. Similarly, as in Figure 7, a document can be verified by uploading it so that the system can check his presence within the blockchain by computing the hash and inferring a smart contract to test if such a hash has been stored. As a result, the hash, signers and the timestamp is showed.

5 CONCLUSIONS AND FUTURE WORK

A notarisation service offers the functionalities to check the authenticity, integrity and existence of a digital contents, and is particularly demanding in our society where the digitisation has shifted in the virtual space processes and documents. In this paper, we have proposed a solution implemented with blockchain so as to have a decentralised architecture, so as to respond to the demands of security, resiliency and efficiency. As a future work, we plan to improve the semantic query of documents within our solution, to implement a more decentralised and efficient authentication/authorisation scheme, and integrating stenography and NFT within the process of registration/verification of the notarised documents.

ACKNOWLEDGEMENT

This research was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2021R1A2B5B02087169).

REFERENCES

[1] Ida Bifulco, Stefano Cirillo, Christian Esposito, Roberta Guadagni, and Giuseppe Polese. 2021. An intelligent system for focused crawling from Big Data sources. *Expert Systems with Applications* 184 (2021), 115560.

[2] Mohammad Jabed Morshed Chowdhury, Alan Colman, Muhammad Ashad Kabir, Jun Han, and Paul Sarda. 2018. Blockchain as a notarization service for data sharing with personal data store. In *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)*. IEEE, 1330–1335.

[3] *Erik Daniel and Florian Tschorsch. 2022. IPFS and friends: A qualitative comparison of next generation peer-to-peer data networks. IEEE CommunicationsSurveys & Tutorials* 24, 1 (2022), 31–52.

[4] Christian Esposito and Oscar Tamburis. 2019. An Effective Retrieval Approach for Documents Related to Past Civil Engineering Projects. In *28th IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE 2019, Naples, Italy, June 12-14, 2019*, Sumitra Reddy (Ed.). 295–300.

[5] Grant Ingersoll, Thomas S Morton, and Drew Farris. 2012. *Taming text: how to find, organize, and manipulate it.* Simon and Schuster.

[6] Tom Kirkham, Sandra Winfield, Serge Ravet, and Sampo Kellomäki. 2012. The personal data store approach to personal data security. *IEEE security & privacy* 11, 5 (2012), 12–19.

[7] Vandana Korde and C Namrata Mahender. 2012. Text classification and classifiers: A survey. *International Journal of Artificial Intelligence & Applications* 3, 2 (2012), 85.

[8] Roman Kräussl and Alessandro Tugnetti. 2022. Non-Fungible Tokens (NFTs): A Review of Pricing Determinants, Applications and Opportunities. *Applications and Opportunities (May 17, 2022)* (2022).

[9] Haikel Magrahi, Nouha Omrane, Olivier Senot, and Rakia Jaziri. 2018. NFB: a protocol for notarizing files over the blockchain. In *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*. IEEE, 1–4.

[10] Ahmed Afif Monrat, Olov Schelén, and Karl Andersson. 2019. A survey of blockchain from the perspectives of applications, challenges, and opportunities. *IEEE Access* 7 (2019), 117134–117151.

[11] Nishara Nizamuddin, Haya R Hasan, and Khaled Salah. 2018. IPFS-blockchain-based authenticity of online publications. In *International conference on blockchain*. Springer, 199–212.

[12] Stanislaw Osinski and Dawid Weiss. 2005. A concept-driven algorithm for clustering search results. *IEEE Intelligent Systems* 20, 3 (2005), 48–54.

[13] Tonino Palmisano, Vito Nicola Convertini, Lucia Sarcinella, Luigia Gabriele, and Mariangela Bonifazi. 2021. Notarization and Anti-Plagiarism: A New Blockchain Approach. *Applied Sciences* 12, 1 (2021), 243.

[14] Martín Vigil, Johannes Buchmann, Daniel Cabarcas, Christian Weinert, and Alexander Wiesmaier. 2015. Integrity, authenticity, non-repudiation, and proof of existence for long-term archiving: a survey. *Computers & Security* 50 (2015), 16–32.

[15] Shuo Xu. 2018. Bayesian Naïve Bayes classifiers to text classification. *Journal of Information Science* 44, 1 (2018), 48–59.