Blockchain Certification and Granular Editing Permissions in Document Management System

Filippo Eros Pani[1], Giacomo Ibba[2], Michele Marchesi[2], Andrea Pinna[3], Simone Porru[1],
Roberto Tonelli[2] and Bartolomeo Valcalda[1]
[1]*T Bridge S.p.A., Genova, Italy*
[2]*Department of Mathematics and Computer Science, University of Cagliari, Italy*
[3]*Department of Electrical and Electronic Engineering (DIEE), University of Cagliari, Italy*

Keywords: Document Management, Blockchain, Decentralized Applications, Smart Contract, Collaborative Editing.

Abstract: Ever-growing digitalization and increasingly competitive markets are driving industry and the public sector into fast-paced transformation. Competitive advantage is being acquired through technology investments made possible by previously unavailable resources, freed by process automation, simplification, and rationalization. Under these contingencies, we propose an innovative document management platform, featuring a collaborative document editing technique and a blockchain certification procedure. The two proposing parties - a private company and an academic organization - mutually agreed on employing open-source technologies as a strategic means to promote software reuse and developer communities' support, and consequently reduce implementation costs and ensure interoperability.

1 INTRODUCTION

Ever-growing digitalization and increasingly competitive markets require companies to adapt at a fast pace. Technology investments ensure a competitive edge, and such investments are made possible thanks to the automation, simplification, and rationalization of processes which, in turn, unleashes previously unavailable resources. The potential market segment addressed by document management software offers a plethora of products and services that support communication and interoperability among different subjects. These products and services usually target business users, such as institutions and public administration. Document management solutions can especially benefit local public authorities. Public organizations, usually relatively complex, can leverage such solutions to save on storage space, application servers, back-office maintenance, and front-office extensions (Pani et al., 2015) On the other hand, managers are increasingly interested in analysing how information is managed within companies, since the associated issues are often at the root of inefficiencies. In fact, information management issues' resolution sets the base for a successful strategy. Documents-related processes (Mahajan and Banerjee, 2018), in particular, need to undergo significant transformations in order to increase business productivity and process efficiency and, at the same time, reduce operating costs. The solution proposed in this paper aims to be a cost-effective solution for business users, in particular by exposing features dedicated to information flows and communications support. In the context of a R&D project, our aim is to offer an innovative software platform, to be provided as "software-as-a-service" (SaaS). The creation of electronic documents, communication management, digital signing and stamping, document recording protocol, and secure transmission, are all features that will be provided by the document management platform. Multi-channel communication will be ensured by leveraging mail, e-mail (also certified email), fax, voice, and text messages. Storage and archiving processes will fulfill local regulations (McHenry and Burt, 2018). We propose an innovative document management platform featuring a collaborative document editing technique, The platform is conceived to be remotely available as a Rich Internet Application (RIA) (Fraternali et al., 2010). The platform features include an innovative blockchain based certification procedure. We decided to use the Aergo blockchain which was released in 2019 to be "the blockchain for business" and allows

Pani, F., Ibba, G., Marchesi, M., Pinna, A., Porru, S., Tonelli, R. and Valcalda, B.
Blockchain Certification and Granular Editing Permissions in Document Management System.
DOI: 10.5220/0008349902990306
In *Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2019)*, pages 299-306
ISBN: 978-989-758-382-7

the development of smart contracts.

In the second section of this paper, we present a background of related works and our development approach. The third section describes the proposed system. In particular, it first presents the key elements of our solution (the collaborative editing system and the blockchain document certification) and then discuss the main features of this two elements focusing on research questions and technological opportunity. Last section presents the conclusions about the proposed project.

2 BACKGROUND

A collaborative editing system is essentially system in which users can contribute concurrently to the same document in remote. The design of collaborative editing platforms must take into account and address the issues and risks related to this specific typology of system. The research is very much in line with the development of new technologies, such as the blockchain. The new proposals of collaborative editing systems can refer on studies and results that have been proposed for some decades. The collaborative writing and related open questions were the subject of a review of the 2004 (Noël and Robert, 2004). In this work the collaborative writing activity was framed in the Computer Supported Cooperative Work (CSCW) research field (Bannon and Schmidt, 1989).

The development of the system we proposed in this paper can be framed in the current research on decentralized collaborative editing platforms. This research focuses on the development of web based applications in which RIAs are included. In particular, technologies like cloud computing and decentralized system allowed the design of new architecture of editing platform, studied to work also in mobility (Ahmad and Imine, 2015) (Resmi and Taiani, 2017). New systems take into account the need for more efficient editing algorithms, such as the Commutative Replicated Data Type (Lv et al., 2018).

The blockchain technology for the decentralized document management is currently the object of investigation. This technology is suitable for providing a verifiable and unchangeable time marking to a document. First studies proposed to implement time stamping by using the Bitcoin protocol (Gipp et al., 2015) (Gao and Nobuhara, 2017). Differently, in our approach we want to exploit the newest generation of blockchain system, which provides a smart contract environment. The smart contract allows to implement several features, such as the control of accessing. In the past, we saw the Ethereum network as the absolute protagonist in this application field (Nizamuddin et al., 2019). Ethereum was the first blockchain studied to develop decentralized applications. Instead of Ethereum we proposed the use of the Aergo blockchain network, developed in 2019 and studied explicitly for business applications.

2.1 Development Approach

For the software development activities related to the project, the lean-based software development approach proposed in (Leffingwell, 2010) is being employed. Such approach was selected not only for the initial implementation, but also for the maintenance and evolution phase. More specifically, Kanban, a tool used in lean methodologies, has been used in maintenance processes (Anderson, 2012). This flexible approach has several advantages when compared to older approaches, especially considering its capability to meet the actual needs of the IT industry. Being it considerably different from more common methodologies used within the IT industry, its impact at the marketing, organizational, and company level is still to be fully understood. Nevertheless, leveraging the afore-mentioned approach helps a single developer team in dealing with the management of concurrent projects - a scenario that is neither ideal nor uncommon.

3 THE SOLUTION PROPOSED

The proposed document management platform is being devised to be a set of safe and reliable web-based RIA tools for both individuals and companies, with the aim of allowing communication with public institutions, banks, private companies, freelancers, and other entities. The solution features two main innovative aspects:

- a dedicated collaborative document editing technique;
- a document certification procedure based upon the blockchain technology.

In fig. 1 the main elements of the proposed solution are represented. The key aspects of the research relating to the elements of the proposed system will be described below.

3.1 Collaborative Document Editing

Research has been conducted on the collaborative creation of complex documents such as:

- invitations to tender;

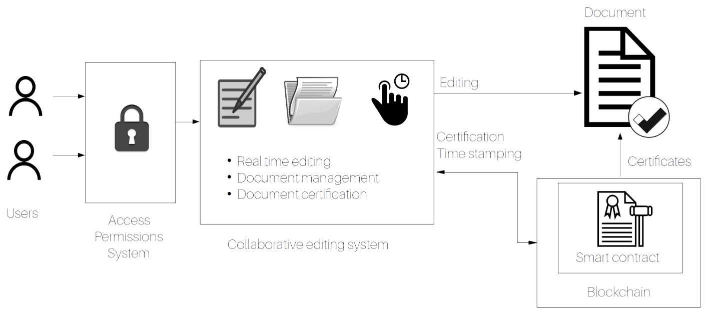


Figure 1: Conceptual diagram of the main elements of the proposed document management platform.

- technical projects;
- contracts;
- commercial offers;
- administrative acts.

The platform allows to support users' real-time interactions during document drafting. In particular, the platform structuring the document into separate sections with a specific type of access and editing. Users' collaboration and interaction is also eased by dedicated auxiliary features, such as the ability to highlight the text of each author with a different colour, or the real-time chat window. The general objectives are as follows:

- simplify and make the preparation of a complex document more efficient, even in case of multi-organization editing;
- ease management and control of the drafting of a document while maintaining granular responsibility levels;
- create links and interactions between the document and the organization's processes;
- simplify and make document signing processes more user-friendly;
- uncouple, as far as possible, the semantic value of the information from its representation, and the medium where it is presented.

As regard the non-functional requirements of the document platform, each application component, in addition to ensuring an optimal user experience, must be available to organizations both on-premise and in the cloud; among other advantages, this would allow to maintain direct control of the data that are most critical. The platform design requires the implementation of the ECM Enterprise through the CMIS standard (OASIS Standard, 2015). This will make it possible to replace the ECM component at any time without any modification to the general architecture. The interfaces of the various application components ensure a modern user experience in order to facilitate users in completing their tasks. On the other hand, as for the functional requirements, an innovative feature is the possibility to structure a document into separate sections with specific user access and editing permissions, as mentioned before. A user with specific permissions must be able to create a new document by defining its descriptive metadata. Then, the system automatically assign the role of document responsible. The user responsible for the document must be able to define the structure of the document by creating appropriate independent sections. When first configuring a section, it is possible to associate to the section the responsible person, by selecting them from the users' system database. The section responsible can also define the team of users that are involved in the creation of that specific section. When multiple users access the section's editing tools, their presence and actions are visible to all the other authors, ensuring real-time collaboration editing. The editorial process of a section comprises the following steps:

- opening of the editing phase;
- collaborative editing by users with permission on the specific section;
- request for approval of the product content at any time;
- approval of the changes by the section manager.

Regarding the versioning, each single version created must be stored in the system's database and made accessible to the users. Metadata such as creation time and date, contributors, as well as the content of each single version, must be recorded. Each version must be automatically and sequentially numbered, and must indicate if it was approved by the section's responsible during the approval process. Moreover, the platform must provide the section's editors a dedicated communication environment for that specific section. This communication channel must allow to view all messages sent by the users, and possibly foresee the exchange of private messages. The same communication channel must be used by the system to provide automatic notifications about the main events related to the section itself.

3.2 Blockchain Document Certification

Another research activity concerned making the document's main responsible capable of ensuring document immutability through blockchain document certification. To this purpose, the system allows to publish the document on a blockchain by simply initiating the blockchain publication process on the page dedicated to the relevant document's version. Once the document is successfully published on the blockchain, and the mining activities are completed, the system will allow to visualize the publication time and date, the transaction ID, and the number of blockchain confirmations.

3.2.1 Blockchain for Document Management: An Overview

The use of **blockchain** technology in document management presents various advantages coming from the intrinsic properties of the technology itself, like documents unalterably, simplification of internal processes, managing information confidentiality, cryptography and private key management for accessing the system. A blockchain is a growing list of records, called blocks, which are sequentially linked using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data. By design, a blockchain is proof tamper to data modification, indeed is defined as "*an open, distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way*"(Nakamoto et al., 2008).

Currently, several business enterprises propose blockchain based solutions in field of document management. Among these we mention *Blocko* (https://www.blocko.io/platform.html), *Storj Labs* (https://storj.io/) and *Cryptyk* (https://www.cryptyk.io/), three companies which are trying with their blockchain and smart contract based solutions to revolutionize the reference market by proposing more efficient and sustainable solutions(Salmerón-Manzano and Manzano-Agugliaro, 2019). Fields of application includes:

- **Distributed Document Management:** A document is stored on multiple machines perfectly synchronized to work at a specific document at the same time.
- **Security:** A document can be encrypted so that the owner of the encryption itself, is the only one to be able to access to that document.
- **Multisignature:** A document can be owned by a set of predefined owners who can act upon only if all the owners or a predefined fraction of them agree.
- **Traceability:** Could be useful on food industry to keeping track of the origin and the freshness of a product and keeping information about transport.
- **Certification:** Blockchain could be useful to demonstrate authenticity and ownership of a document.
- **Recognition:** Blockchain is becoming popular on the management of documents which allow identification such as identity cards and passports, so, because of this, the blockchain could be used to simplify task as boarding and landing of passengers.
- **Timestamping:** Associating a timestamp to a document establishes a legally valid date and time to that document.

The role of the blockchain technology and especially that of smart contracts is continuously emerging to support new features of software systems. Many standard features of software systems can be implemented within the blockchain which plays the role of decentralized services (also called blockchain as a service - BaaS)(Daniel and Guida, 2019) or microservices (Tonelli et al., 2019).

3.3 Document management with Aergo

Aergo (AERGO, 2019) is a blockchain project for business applications recently launched by the company Blocko. The first version of the platform was released on 9 April 2019. Aergo (AERGO, 2019) is a hybrid blockchain which tries to gather the advantages of private and public blockchains. It is **Proof of Stake** (Bitcoinwiki, 2019) based, so every user must prove that he owns a certain amount of tokens.
Another interesting feature of Aergo is that it uses a particular protocol of interoperability called **Merkle Bridge** to allow more chains to communicate with each other. Furthermore it implements the possibility of Merkle proofs usage to create a communication between Ethereum and Aergo to demonstrate and certificate the existence of something in another blockchain. Merkle bridge could be seen as a multi-signature contract which signs the state of the root of a sidechain instead of signing individual transactions, like in other blockchain implementations. Aergo allows to interact with the blockchain with different programming languages, in particular Java, Javascript and Python. For Java there is *Heraj*, for Javascript *Herajs*, and the

Python library is called *Herapy* (AERGO contributors, 2019) .

We first analyze Heraj which provides features and functionality to interact with the Aergo blockchain and with the development and the execution of smart contracts. Next we will have a look at how Heraj could be used to deploy a Lua smart contract.

Aergo Smart Contracts. Aergo allows to develop and run smart contracts using LUA language and Athena-Ide, which is an extension of the Atom text editor. In addition Aergo provides a testnet blockchain in which it is possible to run and test decentralized applications without spending real Aergo. By following the instructions given by the company available at (AERGO team, 2019) everyone can obtain an Aergo wallet and an Aergo account. Aergo provides *Aergo Connect*, which is a blockchain bridge that allows to connect to the Aergo blockchain without installing a blockchain node. It is very similar to the more famous Ethereum's metamask. Thanks to Aergo Connect it is possible to chose to work in the Aergo main net or in the testnet. We worked using the Aergo test net because it is free. Aergo provides a tool to require tokens spendable only on the testnet. This tool provides an amount of five AERGO, which is more than necessary to build up our smart contract. Aergo use a particular set up for Lua language, indeed the use of some default libraries like the "os" one is not allowed for security purposes, and it provides a type Map which is not of the original Lua. This type, is like a Java Hash-Map but keys are allowed to be only strings, while values are allowed to be of any type.Inside the **state.var** scope we define our global variables, and inside the **constructor()** we set up variables.

As represented in Fig 2, on the left panel we can notice the Athena-Ide interface, which is very similar to Remix environment for Solidity smart contracts development. Indeed, we can see that is possible to import an Aergo account, compile contracts and make the contract **deploy** to append it on Aergo blockchain; also, is possible to select if we want to work locally, on test net or on main net.

3.4 The Document Timestamping

We now describe the implementation of a blockchain based system for document timestamping we developed during the Document management project.

Given a document, we want to use the blockchain to associate a timestamp to this document exploiting the Aergo blockchain. The timestamp must be immutable, so once the document is timestamped, it can not be timestamped again.

In the following we describe functions and variables defined in the implementation. In the Appendix of this paper we we provide the complete source code of our smart contract.

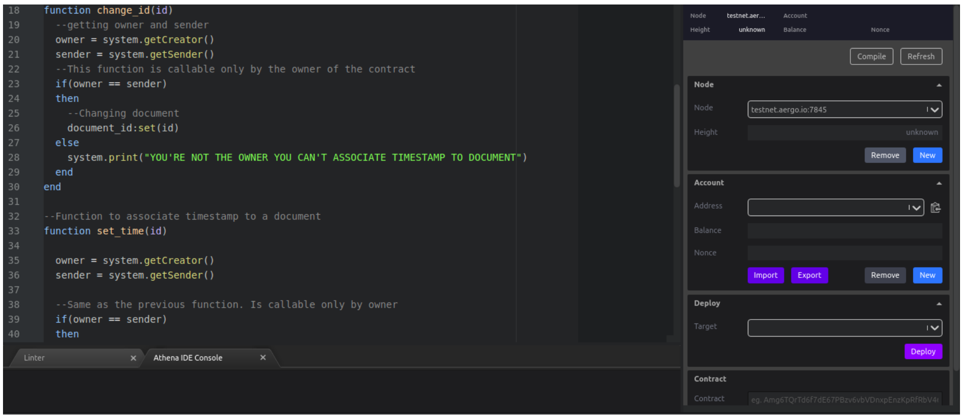
Global Variables. We used two global variables, **document_prove** and **document_id**. *document_prove* is a Map type, and it stores the document id and the associate timestamp. *document_id* is an integer which represents the document id.

Function set_time. This functions should work only if the one to call them is the owner of the contract. Thanks to the **"system"** library provided by Aergo, we can call functions like **getCreator()** and **getSender()** which allows us to make those types of controls; so, if the contract is called by the owner we can set a new document id.
As before, we check that **owner** and **sender** coincide, and so if this is the case, we check that for a particular document id a timestamp has not been associated yet; if this is the case, then we associate a timestamp to that document else we return an alert message.

Function get_timestamp_of_a_document. As ever we check that the one which is calling the function is the owner of the contract. If the owner is calling the function then the associated timestamp to the document id given in input will be returned. Else, if the function is not called by the owner, an alert message will be printed.

3.5 System Integration

As we said previously, Aergo provides SDK to interact with the blockchain. *Heraj* is the library provided by Aergo for Java language; this library allows to interface Java with Aergo blockchain. This type of interaction with Java and Aergo allows the development of decentralized applications (dAPPs). Heraj makes the development and the execution of smart contracts very simple and efficient. To use the library, we can take the advantages provided by tools such as *Gradle* (https://gradle.org/) or *Maven* (https://maven.apache.org/). In the collaborative document management system, the platform ALFRESCO is modified to include the Heraj library. This allows the interfacing of the software with the Aergo blockchain in order to use the smart contracts deployed in it such as the timestamping one. To given an example, when we want to modify a document, ALFRESCO could execute a



```
function change_id(id)
  --getting owner and sender
  owner = system.getCreator()
  sender = system.getSender()
  --This function is callable only by the owner of the contract
  if(owner == sender)
  then
    --Changing document
    document_id:set(id)
  else
    system.print("YOU'RE NOT THE OWNER YOU CAN'T ASSOCIATE TIMESTAMP TO DOCUMENT")
  end
end

--Function to associate timestamp to a document
function set_time(id)

  owner = system.getCreator()
  sender = system.getSender()

  --Same as the previous function. Is callable only by owner
  if(owner == sender)
  then
```

Node testnet.aer... Account
Height unknown Balance Nonce

Node: testnet.aergo.io:7845
Height: unknown

Contract: eg. Amg6TQrTd6f7dE67PBzv6vbVDnxpEnzKpRfRbV4(

Figure 2: The AERGO development platform for smart contracts.

smart contract using Heraj library and then, the contract associate the block timestamp to the document id to provide an incontrovertible date of the last document changes.

By using Heraj it is possible to interact and even to deploy a smart contract in the Aergo blockchain directly from the Java application. To connect the Heraj library to our Aergo account we need all of our credentials (in particular we need our private key and our decryption account) and we need an Aergo client installed.

4 CONCLUSION

In this work, we propose an innovative document management solution built by using development tools backed by large developer communities, with the aim of ensuring the availability of adequate support during development. The architectural design is being developed as a modular solution, due to the need for integrating additional modular, interoperable components, and built with open standards to make the system flexible. The prototype will try to effectively tackle the complexities stemming from the interaction of separate, interdependent processes. Specific services will have to work together with the proposed solution, thus requiring to devise a proper infrastructure to meet users' requirements as seamlessly as possible. The system will ensure effective integration, and employ a cloud computing system (Lewis, 2012) (Rehman, 2018). The involved parties will acquire new valuable knowledge on document management, thanks to the cooperation network which will stem from the development activities. The very nature of the project itself is bound to ensure knowledge sharing: being open-source, the project promotes the intervention of external contributors, especially from individuals and organizations directly interested in tackling their own business challenges with the proposed solution. Among the innovative features of the proposed document management solution, are i) the possibility to assign specific access permissions to each document section, and ii) document certification based on blockchain technology.

ACKNOWLEDGEMENTS

The work presented in this paper has been partially funded by Regione Autonoma della Sardegna, under projects “Gestione Documentale” - Progetto PIA, Annualità 2013”, “Crypto-Trading” - Programmazione unitaria 2014-2020 POR FESR Sardegna 2014 - 2020, “Easy Wallet” - Programmazione unitaria 2014-2020 POR FESR Sardegna 2014 - 2020, “SardCoin – SardCoin: tecnologie blockchain a supporto del turismo in Sardegna” - TOP-DOWN POR FESR sardegna 2014 – 2020 Asse prioritario 1, and “CAFCha - Certification of AgriFood Chain”- POR FESR Sardegna 2014 – 2020 Asse 1.

The authors would like to thank Prof. Giovanni Michele Pinna and Prof. Fabio Zuddas from Department of Mathematics and Computer Science, University of Cagliari, Italy, and Prof. Alessandro Spano from Dipartimento di Scienze Economiche ed Aziendali, University of Cagliari, Italy, for their helpful comments and suggestions.

REFERENCES

AERGO (2019). Aergo the blockchain for business. https://www.aergo.io/.

AERGO contributors (2019). Aergo project repository. https://github.com/aergoio.

AERGO team (2019). Aergo project documentation. https://docs.aergo.io.

Ahmad, M. and Imine, A. (2015). Decentralized collaborative editing platform. In *2015 16th IEEE International Conference on Mobile Data Management*, volume 1, pages 323–326.

Anderson, D. J. (2012). *Lessons in agile management: on the road to Kanban*. Blue Hole Press.

Bannon, L. J. and Schmidt, K. (1989). Cscw: Four characters in search of a context. In *ECSCW 1989: Proceedings of the First European Conference on Computer Supported Cooperative Work*. Computer Sciences Company, London.

Bitcoinwiki (2019). Proof-of-stake. https://en.bitcoinwiki.org/wiki/Proof-of-stake.

Daniel, F. and Guida, L. (2019). A service-oriented perspective on blockchain smart contracts. *IEEE Internet Computing*, 23(1):46–53.

Fraternali, P., Rossi, G., and Sánchez-Figueroa, F. (2010). Rich internet applications. *IEEE Internet Computing*, 14(3):9–12.

Gao, Y. and Nobuhara, H. (2017). A decentralized trusted timestamping based on blockchains. *IEEJ Journal of Industry Applications*, 6(4):252–257.

Gipp, B., Meuschke, N., and Gernandt, A. (2015). Decentralized trusted timestamping using the crypto currency bitcoin. *arXiv preprint arXiv:1502.04015*.

Leffingwell, D. (2010). *Agile software requirements: lean requirements practices for teams, programs, and the enterprise*. Addison-Wesley Professional.

Lewis, G. (2012). The role of standards in cloud-computing interoperability. software engineering institute.

Lv, X., He, F., Cai, W., and Cheng, Y. (2018). Supporting selective undo of string-wise operations for collaborative editing systems. *Future Generation Computer Systems*, 82:41 – 62.

Mahajan, K. and Banerjee, P. (2018). A review study on document management system. *Computer Software and Media Applications*.

McHenry, C. A. and Burt, S. W. (2018). Electronic document classification. US Patent 9,928,244.

Nakamoto, S. et al. (2008). Bitcoin: A peer-to-peer electronic cash system.

Nizamuddin, N., Salah, K., Azad, M. A., Arshad, J., and Rehman, M. (2019). Decentralized document version control using ethereum blockchain and ipfs. *Computers & Electrical Engineering*, 76:183 – 197.

Noël, S. and Robert, J.-M. (2004). Empirical study on collaborative writing: What do co-authors do, use, and like? *Computer Supported Cooperative Work (CSCW)*, 13(1):63–89.

OASIS Standard (2015). Content management interoperability services (cmis). http://docs.oasis-open.org/cmis/CMIS/v1.1/CMIS-v1.1.html.

Pani, F. E., Porru, S., and Ibba, S. (2015). A model for digital content management. In *Proceedings of 4th International Conference on Data Management Technologies and Applications*, pages 240–247. SCITEPRESS-Science and Technology Publications, Lda.

Rehman, T. (2018). *Cloud Computing Basics*. Stylus Publishing, LLC.

Resmi, A. C. and Taiani, F. (2017). Filament: A cohort construction service for decentralized collaborative editing platforms. In *IFIP International Conference on Distributed Applications and Interoperable Systems*, pages 146–160. Springer.

Salmerón-Manzano, E. and Manzano-Agugliaro, F. (2019). The role of smart contracts in sustainability: Worldwide research trends. *Sustainability*, 11(11).

Tonelli, R., Lunesu, M. I., Pinna, A., Taibi, D., and Marchesi, M. (2019). Implementing a microservices system with blockchain smart contracts. In *2019 IEEE International Workshop on Blockchain Oriented Software Engineering (IWBOSE)*, pages 22–31. IEEE.

APPENDIX

In this appendix we present the AERGO smart contract source code written in LUA language for document time stamping.

```

--Defining global variables
state.var
{
--Id of the document and map to
     associat timestamp to document

  document_id = state.value(),
  document_prove = state.map()

}

function constructor()
  --Setting up the constructor
  --Currently the id 0 is reserved
  document_id:set(0)
  document_prove[tostring(document_id)] =
        "RESERVED"
end

function change_id(id)
  --getting owner and sender
  owner = system.getCreator()
  sender = system.getSender()
  --This function is callable only by
        the owner of the contract
  if(owner == sender)
  then
    --Changing document
    document_id:set(id)
  else
    system.print("YOU'RE NOT THE OWNER YOU
          CAN'T ASSOCIATE TIMESTAMP TO DOCUMENT")
  end
end

--Function to associate timestamp to a
     document
function set_time()

  owner = system.getCreator()
  sender = system.getSender()

  --Same as the previous function. Is
        callable only by owner
  if(owner == sender)
  then
    --If document has been already
          timestamped is not modifiable
          anymore
    if(document_prove[tostring(document_id)]
          == nil)
      then
        --Associating timestamp to
              document
        document_prove[tostring(document_id)]
            = system.getTimestamp()
    else
        system.print("ALERT!!! DOCUMENT ALREADY
              TIMESTAMPED. YOU CAN'T MODIFY IT")
    end
  else
    system.print("YOU'RE NOT THE OWNER YOU
          CAN'T ASSOCIATE TIMESTAMP TO DOCUMENT")
  end
end

--Function to get timestamp of document
function get_timestamp_of_document(id)
  --Callable only by owner
  owner = system.getCreator()
  sender = system.getSender()
  if(owner == sender)
  then
    --Return the timestamp associated to
          that document
    return document_prove[tostring(id)]
  else
      system.print("YOU'RE NOT THE OWNER YOU
            CAN'T ASSOCIATE TIMESTAMP TO
            DOCUMENT")
  end
end


--Setting on the Abi viewable functions
abi.register(set_time,change_id,
     get_timestamp_of_document)
```