João Vicente Meyer

Document validation using blockchain

A validation scheme for natural person's documents

Florianópolis - SC, Brazil

2019

João Vicente Meyer

Document validation using blockchain

A validation scheme for natural person's documents

Monografia submetida ao Programa de Graduação em Ciências da Computação para a obtenção do Grau de Bacharel em Ciências da Computação.

Universidade Federal de Santa Catarina

Departamento de Informática e Estatística

Ciências da Computação

Orientador: Jean Everson Martina

Coorientador: Lucas Machado da Palma

Florianópolis - SC, Brazil

2019

Document validation using blockchain
A validation scheme for natural person's documents / João Vicente Meyer. – Florianópolis - SC, Brazil, 2019- 65 p. : il.(alguma color.); 30 cm.
Orientador:Jean Everson Martina
Coorientador:Lucas Machado da Palma
Trabalho de Conclusão de Curso de Graduação – Universidade Federal de Santa Catarina
Departamento de Informática e Estatística
Ciências da Computação , 2019.
1. blockchain. 2. cartório. I. Jean Everson Martina. II. Universidade Federal de Santa Catarina. III. Faculdade de Ciências da Computação. IV. Document validation using blockchain
A validation scheme for natural person's documents
CDU 02:141:005.7

João Vicente Meyer

Document validation using blockchain
A validation scheme for natural person's documents

Monografia submetida ao Programa de Graduação em Ciências da Computação para a obtenção do Grau de Bacharel em Ciências da Computação.

Projeto aprovado. Florianópolis - SC, Brazil, 28 de novembro de 2019:

Jean Everson Martina
Orientador

Lucas Machado da Palma
Coorientador

Pablo Rinco Montezano
Convidado 1

Igor Heidrich
Convidado 2

Florianópolis - SC, Brazil
2019

Dedicated to, in no particular order, my mother, my father, my older sister, my sister who is older than me but not older than the other sister, my friends.

Agradecimentos

To my family and friends.

Resumo

O sistema notarial brasileiro é grande. Isto se deve, principalmente, ao tamanho do país. Tanto em território quanto em população. Cada cartório é como uma entidade privada com sua própria organização e e métodos de registro de documentos. O resultado disto é um sistema muito complexo e lento. Toda vez que uma destas entidades precisa de informação de outra, é necessário, na maioria das vezes, que isto seja realizado manualmente. Seja indo ao local físico do outro cartório, ligação telefônica ou correio. Como é fácil perceber, isto não é aceitável nos dias de hoje com tecnologias atuais. É um processo muito caro e lento, com diversos pontos fracos. Pessoas cometem erros, registros físicos são perdidos, até cartórios inteiros são, às vezes, perdidos.

Tecnologias de blockchain podem ser de grande ajuda neste cenário. Ela nos dá uma maneira distribuída de guardar e validar data entre diversos participantes. Neste caso, os cartórios. Além disso, pode providenciar uma maneira extremamente transparente de garantir a autenticidade e validade de todos os documentos colocados no sistema em qualquer momento.

Este projeto desenvolve um protótipo capaz de armazenar e validar registros públicos de pessoas naturais em uma blockchain. O protótipo engloba os documentos de nascimento, casamento, divórcio e óbito. Ao final, este projeto demonstra os custos operacionais do protótipo e realiza uma comparação com o sistema cartorário utilizado atualmente.

Palavras-chave: blockchain. cartório. documento. validação

Abstract

The Brazilian public notary system is quite big. Mainly, because of the sheer scale of the country. In territory and in population. Each notary is like a private entity, with its own way of handling the records registered into it. The result is a very complex and slow system. Every time one of these entities needs info from another peer, it needs to, most of the time, make a request manually. Either by going to the physical place of the other notary, or calling it. As it is easy to note, not acceptable in current times with current resources and technologies. It is a very slow and costly process with many loose ends. People make mistakes, physical records are lost, even full notaries are sometimes lost.

Blockchain technology can be of huge help in this scenario. It provides a distributed way of storing and validating data between many players. In this case, the notaries. Besides, it is capable of providing an incredibly transparent way to guarantee the authenticity and validity of every record every put in the system at any point in time.

This project creates a prototype that capable of storing and validating public records of natural persons in a blockchain. The prototype encompass the birth, marriage, divorce and death records. In the end, this project shows the operational costs of such prototype and makes comparisons with the current notary system.

Keywords: blockchain. notary. document. validation

Contents

Contents **13**

1 INTRODUCTION **15**

2 BASIC CONCEPTS **17**

2.1 Public Notaries **17**

2.1.1 History 17

2.1.2 Function 18

2.2 Cryptography **19**

2.2.1 Hash Functions 19

2.2.1.1 Hash Pointers and Hash Lists 20

2.2.1.2 Merkle Trees 20

2.2.2 Asymmetric cryptography 21

2.2.2.1 Digital signatures 21

2.2.3 Blockchain 22

2.2.3.1 Consensus 23

2.2.4 Smart contracts 23

2.3 Tools **24**

2.3.1 Ethereum 24

2.3.2 IPFS 24

2.3.3 Quorum 24

2.3.4 Hyperledger 25

3 METHODOLOGY **27**

3.1 Review protocol **27**

3.1.1 Questions 27

3.1.2 Research 27

3.1.3 Filtering 28

3.1.4 Results 28

4 PROPOSAL **31**

4.1 Problem **31**

4.2 Related works **32**

4.3 Proposed solution **33**

4.3.1 Considerations 33

4.3.1.1 Limitations 33

4.3.1.2 Requisites 34

5 IMPLEMENTATION 37
5.1 Model 37
5.2 Architecture 37
5.2.1 Notary 37
5.2.2 Record 38
5.2.3 Person 39
5.3 Validation 39
5.3.1 Costs 40
5.3.1.1 Notary 40
5.3.1.2 Birth 41
5.3.1.3 Marriage 41
5.3.1.4 Death and divorce 41
5.3.2 Comparison 43

6 CONCLUSION 45

BIBLIOGRAPHY 47

APPENDIX A – IMPLEMENTATION 51

APPENDIX B – PAPER 57

1 Introduction

The Brazilian notary system is composed of thousands of institutions. They are privately controlled and have the legal power provided by the state (RODRIGUES, 2013, p. 232). Unfortunately, this huge network still uses archaic types of communication between its peers and paper to store most of its records. Furthermore, Brazilian law states that every natural person's registered document can be accessed by any person interested in it (BRASIL, 1973, Art. 17). It could be said that the notary system is like a set of databases scattered throughout the Brazilian territory. Each one using its own schema and formalities.

The problem arises when there is the need of communication between all of the institutions. Usually, this is a manual process. A human needs to make a request to another notary for a piece of document or information. This makes the process very error prone, slow and expensive. In addition, there is the possibility of loosing documents or a notary building, and all its documents, being lost in some catastrophe. Another problem is the scattering of data. Only the notaries at the place of birth of some person are obligated to have the info about said birth (BRASIL, 1973, Art. 50).

We aim to create a decentralised document validation system prototype. Using blockchain technologies. This system should be able to incorporate already existing records, usually physical ones, into itself. Therefore, the adaptation process from the current system to this new one would be hugely facilitated.

As a result of creating such a system, and putting it to use, the data would be replicated in every single node of the network. The problem of data loss would be entirely out of question. Besides, there is the history of every single record. They would all be available to whoever has access to this blockchain network. Also, it would lower the costs with people. Workers would not need to spend time requesting and waiting for a response from another notary. They would simply access some API or user interface that has access to the network and "fetch" the desired record, if it exists. Thus, reducing bureaucracy and inefficiency.

On the other hand, there are some problems that are a concern. There are special cases in Brazilian law that some documents should not be available to the public in its fullest. Adoptions are a good example of it (BRASIL, 1973, Art. 19). These cases are not part of the scope of this project. Only the cases of already publicly available records will be considered.

Some works have already been done in this area. IDStack (LAKMAL et al., 2017) uses the idea of automatically extracting data from existing documents to make them

easily validated and authenticated. The use of blockchain for Brazilian high education system has already been discussed (COSTA et al., 2018). There is also an implementation of a decentralised issuer of diplomas based on historic data (PALMA et al., 2019).

This document structure is as follows: first, a brief history of the Brazilian notary system is given. Basics on cryptographic primitives and blockchain technologies will be described afterwards. State of the art research and correlated works will follow. Finally, the proposal and its software implementation will be discussed and evaluated.

2 Basic Concepts

This chapter aims to give the reader a context on the theory behind the design of this project. Here we will describe a little bit of the history of the public notary system of Brazil. In the same way, cryptography primitives that are the base in which blockchain systems are designed upon will also be explained in this chapter.

2.1 Public Notaries

Throughout Brazil's history, there were many attempts by the government to create a functioning notary system (RODRIGUES, 2013, p. 24). However, the current system has a very important role in the functioning Brazilian society (NETO, 2008, p. 79).

2.1.1 History

The "Ordenações Filipinas", 1595, were an early formal instrument regarding public records. Yet, they were simply very broad words with a punitive approach (NETO, 2008, p. 15). More than 200 years later, "Lei das Terras", 1850, was another attempt from the government to formalise land owning documents. According to Sanches (2008, p. 25), this law did not have great effect.

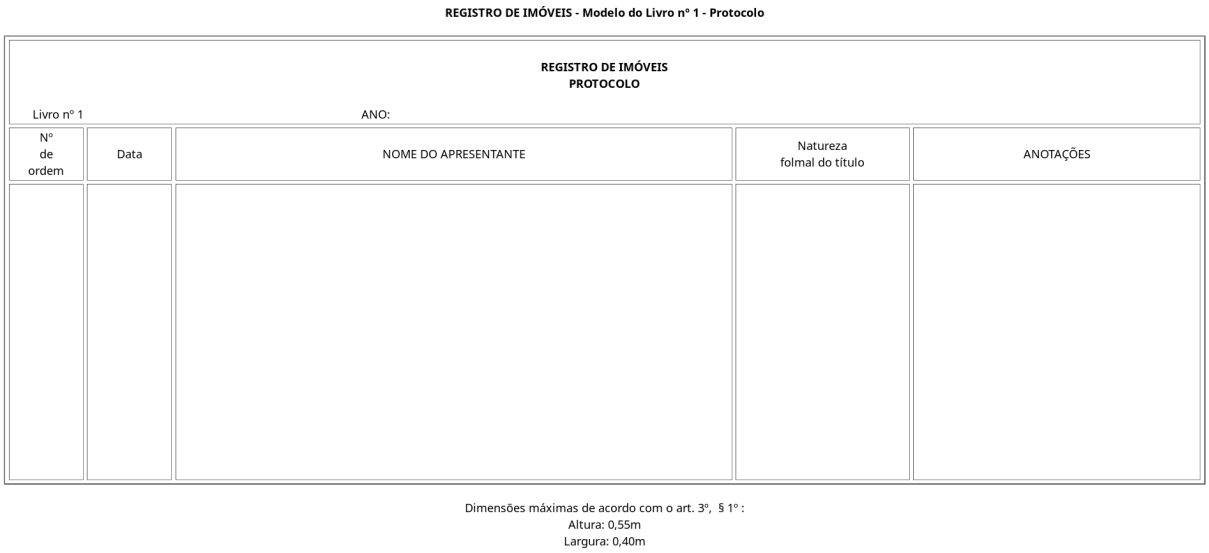
One of the first attempts in trying to formalize natural person's documents occurred in 1851, decree 798. It said:

> Art. 1º Haverá em cada Districto de Juiz de Paz hum livro destinado para o registro dos nascimentos, e outro para o dos obitos que tiverem lugar no Districto annualmente.

It declares that, in every district, should have a book with the function of registering births and another one for registering deaths. However, this decree wasn't well received by the population. It resulted in a popular revolt called "Ronco da Abelha" (TIZIANI, 2016).

The establishment of non catholic marriages in Brazilian law happened in law 1.144 of 1861. Besides, it also declared the obligation, for record keepers, to have a new record book. A marriage records book. Law 5.604 of 1874 united all the public registries into a single civil registry nomination. Also, it listed the main facts of to be recorded throughout a person's life. Birth, marriage and death:

> Art. 1º O registro civil comprehende nos seus assentos as declarações especificadas neste Regulamento, para certificar a existencia de tres factos: o nascimento, o casamento e a morte.



REGISTRO DE IMÓVEIS - Modelo do Livro nº 1 - Protocolo

REGISTRO DE IMÓVEIS
PROTOCOLO

Livro nº 1 ANO:

Nº de ordem	Data	NOME DO APRESENTANTE	Natureza folmal do título	ANOTAÇÕES

Dimensões máximas de acordo com o art. 3º, § 1º :
Altura: 0,55m
Largura: 0,40m

Figure 1 – Model of the book to be used in real state records (BRASIL, 1973).

The list of laws created by the government in attempts to create a notary system continue for some time. Finally, in 1973, Law 6.105, regulated, with success, the public notaries. It also provided a model for the registry books. See Figure 1. With some further modifications by law 8.935 of 1994, "Lei dos cartórios", we arrive at the current state of the public notary system in Brazil.

2.1.2 Function

Notary function is defined by:

> ...a função de autenticação que consistem em investir o ato praticado da presunção de veracidade, que estão apots a import, por si mesmos, nas relações jurídicas, em virtude da fé pública, garantindo a certeza do fato e a identidade do seu autor (JÚNIOR, 1961).

Which means, public notaries have the function of providing authenticity to contracts, business and all the parts involved in such actions. In this view, "notary law is the law of authenticity and format" (MUSTAPICH, 1974, our translation).

It is easy to note that, according to many authors in the area, the main function and responsibility of the notary system is to guarantee the authenticity of documents. The power of doing so is delegated by the state (RODRIGUES, 2013, p. 232). When a notary authenticates a document, it is officially recognising it as a valid and truthful document (PUGLIESE, 1989, p. 67). This will be the main view used throughout this project.

2.2 Cryptography

This section describes, briefly, what is considered essential knowledge for understanding how a blockchain system works. There are a set of cryptographic tools, usually called primitives, that are used in every single blockchain, and even non-blockchain, information systems that makes its use possible and practical.

2.2.1 Hash Functions

In a colloquial manner, and according to Stallings (2011, p. 328), a hash function[1] digests a message of indiscriminate size into a fixed size one. A "message", in this context, is any sequence of bits. Any input to a hash function, no matter the size or its contents, will generate an output of a predefined length.

For the following of this work, we will consider a hash function as simply as a function $H(M) = m$, where M is the original message and m is the digested, fixed size, message. Called hash.

For a hash function to be considered useful in cryptography, it must have some properties. The main ones will be listed (NARAYANAN, 2016):

1. First preimage resistance: for any given digested message h, it is impractical to find a message M such that $H(M) = h$;

2. Second preimage resistance: given a message M, it is computationally impractical to find another message Q where $H(M) = H(Q)$;

3. Collision resistance. It is impractical to find any pair (X, Y) that obey $H(X) = H(Y)$;

4. The resulting digested message must pass in pseudo random tests: like the ones described in Marsaglia, Tsang et al. (2002) to be considered cryptographically safe to use;

5. The outputs of the hash function must be equally distributed throughout the possible range.

The most famous hash functions in use today are the MD family, already considered broken (DOBBERTIN, 1996), and the SHA family, used by Bitcoin's proof of work algorithm (NAKAMOTO, 2008), for example.

[1] Or a trapdoor function.

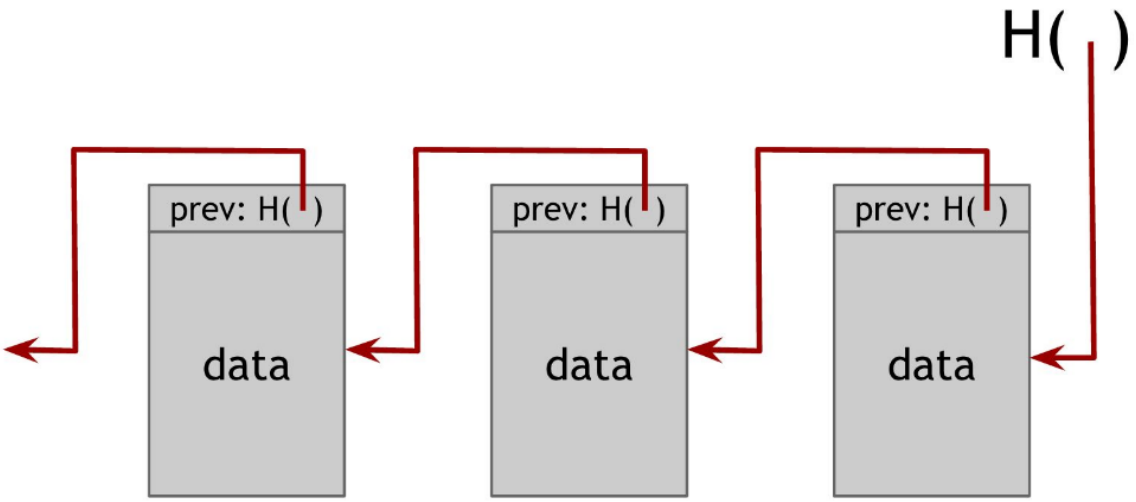


Figure 2 – Hash pointer representation (NARAYANAN, 2016, p. 12).

2.2.1.1 Hash Pointers and Hash Lists

Hash pointer is a data structure widely used in the blockchain environment (NARAYANAN, 2016, p. 11). It is basically a pointer. The same concept that is seen in many programming languages, like C/C++. But, besides pointing to some information, it also contains some a hash of the data is pointing to. Figure 2, better describes this concept.

This is a very good way to prevent tampering with the data. Because, if someone wanted to change some random block of data, it would need to update all the subsequent pointers of the other blocks. The hash of the current block was created by hashing the contents of the previous block together with the hash pointer of that block. This the main motive that hashes are used to assure authenticity of data. The structure shown in Figure 2 should keep going to the left, until it reaches the leftmost block, which is the first block of these series. Usually called “genesis block”.

2.2.1.2 Merkle Trees

Merkle trees are for hash pointers what trees are for linked lists. They allow very efficient verification of big data structures. Like in a tree, each node stores two or more pointers to the next nodes and a hash of the data of that node. In order to verify the integrity of data stored in the tree, you only need a logarithmic amount of verifications in comparison with the size of the tree (BECKER, 2008).

This structure is usually used to store large quantities of data inside the blockchain. This ends up saving lots of space, because the chain only needs to store the hash of the root node of the stored tree instead of the whole tree. Figure 3 represents it nicely.

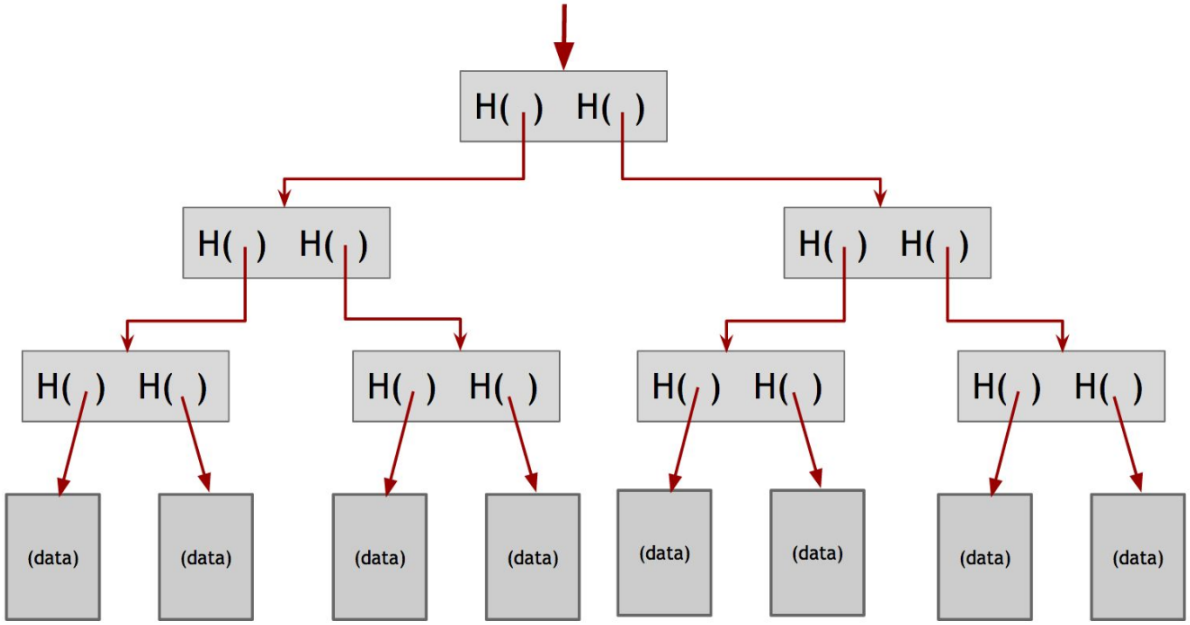


Figure 3 – Merkle tree representation (NARAYANAN, 2016, p. 14).

2.2.2 Asymmetric cryptography

Asymmetric cryptography arose from the necessity of sharing messages without a secure channel (STALLINGS, 2011, p. 269). Symmetric algorithms, like the Advanced Encryption Standard (AES), require a key to be exchanged previously to the communication of the data. This is a very limiting property of symmetric cryptography. To tackle this limitation, asymmetric cryptography was envisioned.

2.2.2.1 Digital signatures

The concept of a digital signature is better described through an analogy:

> A digital signature is supposed to be the digital analog to a handwritten signature on paper. (NARAYANAN, 2016, p. 16)

A digital signature must have some properties that are very similar to its non-digital counterpart (STALLINGS, 2011, p. 398): (a) It must verify the author, date and time of signature; (b) It must authenticate[2] the contents of the signature; (c) It must be verifiable by third parties to resolve disputes.

The properties above are solved by what is known as public key cryptography and protocol. Known as the "Diffie-Hellman" protocol (DIFFIE; HELLMAN, 1976). Public key cryptography is based on each one of the exchanging parties to have a pair of keys. One public, one private. The public key is, as the name says, public. Anyone may have access to it. The private one, in the other hand, must be kept secret.

[2] Here we see the use of hash functions already.

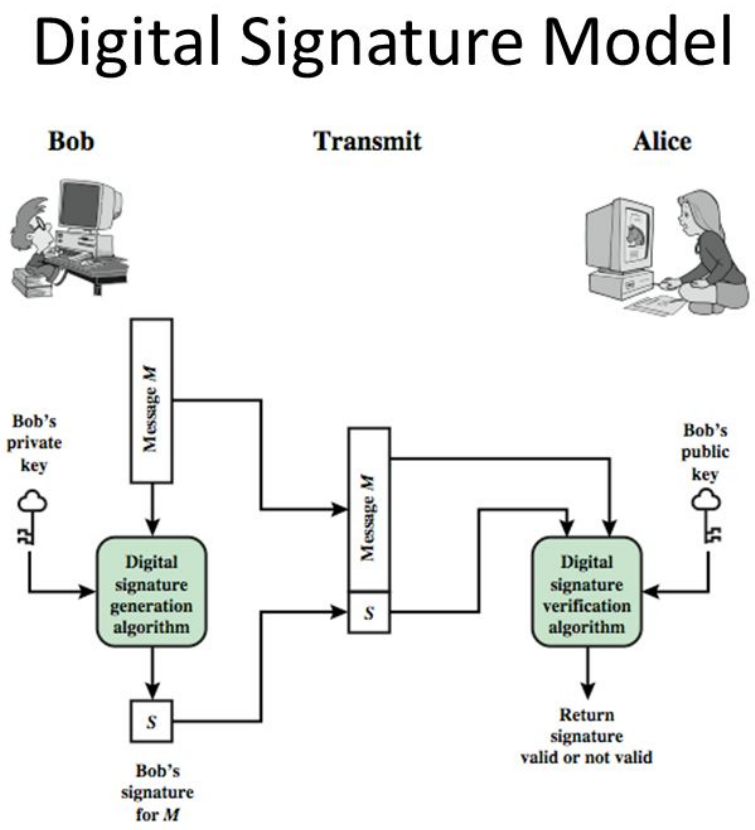


Figure 4 – Generic digital signature process (STALLINGS, 2011, p. 397).

If a user, Bob, wants to send a private message to another user, Alice, Bob would use his private key to sign a hash of the original message and send it to Alice. Alice, using Bob's public key to decrypt the hash, would compare the resulting hash with the hash of the received message. If they are the same, it means that Bob really did send the message. Figure 4 shows the general idea of the process.

The Diffie-Hellman protocol is extensively used in any network based system. If it needs to exchange messages and assure the authenticity of said messages, it is a must. Distributed blockchain systems are not different and use digital signatures broadly throughout its networking.

2.2.3 Blockchain

A blockchain is a chain of blocks. The blocks can be made of any data. The first published work that made use of blockchain[3] was in 1990 (HABER; STORNETTA, 1990). It devised a way to create tamper-proof time stamps for computer files. The big difference between the modern blockchains and the one devised in 1990 is the network distribution

[3] Although it was not called "blockchain" when it was published.

of it.

Every node in the network[4] has an identical copy of the blockchain. Every change performed in one of them is sent to every other node. Unfortunately, a big problem arises from it. How to guarantee that the changes made are valid/trustworthy? This is famously known as the Byzantine Fault (WENSLEY et al., 1978). To solve this problem, there is the use of consensus algorithms.

2.2.3.1 Consensus

Consensus algorithms are used to achieve, as the name indicates, consensus on some data, transaction or communication shared between nodes. Nakamoto (2008) published the incentive based consensus algorithm together with the Bitcoin proposal. The base idea is that there must be incentives for the entities in the network to play by the rules. It does not sound like a perfectly safe idea, in theory. However:

> ...we observe consensus working, but have not developed the theory to fully explain why it works (NARAYANAN, 2016, p. 35).

There are many consensus algorithms already devised: proof of work; proof of stake; proof of luck; etc. The most famous one is Bitcoin's proof of work (NAKAMOTO, 2008). It boils down to finding a partial hash based on the hash of the current block of the blockchain. When found, the value used to generate the hash is sent to every node on the network to be tested as a "proof of work". Which means that you have spent computational power. If valid, the nodes that verified it will add your block to the chain and the process repeats itself.

2.2.4 Smart contracts

It was first conceptualised by Szabo (1997). It works on the idea that a blockchain network can agree in some random pieces of data to be added in the list of blocks, it can also validate computer programs executions. As the execution is limited to only use data inside the chain itself, the execution of these computer programs should be the same in every node that has the same chain. This logic is used to validate smart contracts in the network. As a consequence, every single state that the contract will ever have will be permanently stored in the blockchain.

Smart contracts provide a new paradigm where participants do not need to rely on any third parties to "force" the execution the contract's clauses. The whole network of nodes will act as the "enforcer". The Ethereum network is the most famous example of a smart contract based blockchain. It provides a virtual machine that executes, and charges for it, code on every single node of the network.

[4] Usually a peer-to-peer (P2P) network.

2.3 Tools

Here we discuss the available tools used to create blockchain-based software. Note that this is not an exhaustive list. They are described with the intent of showing the market available options that could be used to further develop the proof-of-concept system designed.

2.3.1 Ethereum

Ethereum (WOOD et al., 2014) is a blockchain platform and distributed computing system. It was released in 2015 and it is fully open-source. Released under the GPL v3 license. The main feature of Ethereum is its smart-contracts. The Solidity language is used to write said contracts using a high level language. They are compiled to bytecode that runs in the Ethereum Virtual Machine (EVM). The language is based in Javascript.

2.3.2 IPFS

The Interplanetary File System (IPFS) (BENET, 2014) is a peer-to-peer (P2P) distributed file system. IPFS creates a hash of every single file stored in it. The files are, subsequently, accessed using these same hashes. Besides, it also features file versioning, similar to *Git*, and duplicate file removal. Its uses are mostly for creating distributed file sharing services.

It is also widely used coupled with Ethereum. When users need to store data in Ethereum, instead of storing it whole (expensive), only the IPFS hash is stored. This method saves considerable amounts of gas, Ethereum's costs for code execution.

2.3.3 Quorum

Ethereum-based, Quorum (MORGAN, 2016) is the "enterprised" version of Ethereum. As described in its website[5]. It supports all of the features that Ethereum offers, but it adds extra ones. It has contract and transaction privacy, permission management system, peer permissions etc. It is maintained by JPMorgan Chase & Co. the big financial firm. It was developed with financial applications first, but it claims to be useful for every enterprise need.

It could be of big use in the Brazilian notary area. The permissioned system allows confidential documents to be stored in its blockchain and to be accessed only by the legally allowed individuals. This is a must for undisclosed documents.

[5] Quorum, <https://www.goquorum.com/>, accessed on 14/05/2019

2.3.4 Hyperledger

Developed by the Linux Foundation, Hyperledger(ANDROULAKI et al., 2018) is a platform for blockchain tools and software. It was developed to be the go-to for enterprise blockchain. Besides that, Hyperledger is also an institution-like project. It helps develop business ideas and software development in the area. It has the support of over 200 members, mostly companies.

3 Methodology

This chapter aims to show how the research for the project was performed. Besides, it targets in demonstrating related works in the area. With that said, the research method used was the systematic literature review (KITCHENHAM, 2004) and is described below.

3.1 Review protocol

3.1.1 Questions

The following questions were created as a way of helping the review process. These are considered of utter importance for being able to define a good system of document validation.

- What are the steps necessary to validate a digital document?
- What are the already available techniques to validate documents using blockchain?
- What are the tools used when developing software of this niche?

The first question is to situate the research on the best techniques of document validation. Digital signatures, certificates, authentication etc. The second and third questions serve to find already available techniques for use in this project.

3.1.2 Research

This subsection is concerned with how the research of the literature was performed. To demonstrate precisely what was executed, the API[1] calls of each one of the online tools consulted.

It is important to comment that only two terms were used in the research. "Blockchain" and "document". The motivation for this was that adding other terms would filter out too many papers, possibly leaving out important ones[2]. This is an issue, in part, because the subject is rather new in the academic world.

Four online databases/tools were consulted. They were: (a) IEEE Xplore Digital Library; (b) Google Scholar; (c) Science Direct; (d) Springer Link. Besides, a date range was set, using the available "advanced search" options, to only show publications made

[1] Firefox developer tools were used to extract the HTTP requests of each tool.
[2] In some cases, there were no results at all.

in 2009 and afterwards. The following table shows the query strings that each one of the APIs generated.

Search tool	*Query*
IEEE Xplore Digital Library	searchField=Search_All_Text queryText=((blockchain) AND document) ranges=2009_2019_Year
Google Scholar	q=blockchain+document hl=pt-BR as_ylo=2009 as_yhi=2018
Science Direct	qs=blockchain document date=2009-2018
Springer Link	all-words=blockchain+document date-facet-mode=between facet-start-year=2009 facet-end-year=2018

Table 1 – Queries performed in each of the databases

3.1.3 Filtering

The research conducted in the last section resulted in a very big number of results. It would be virtually impossible to read all of the hundreds of results in time. Thus, in order to filter out (considered) unimportant papers, some inclusion and exclusion parameters were set and are described below.

For including a work: (a) Only papers written in Portuguese and English; (b) Only papers published after 2009[3]; (c) The work should be about possible software implementations and protocols of document validating systems; (d) Describe tools used in the area; (e) Talks about possible government applications.

The following excluding parameters were set because of the big influx papers that solely discuss the future of blockchain technology and its possible uses, but it does not talk about document validation. To filter those out: (a) The text talks about only generic uses of blockchain technology; (b) Has no focus in document validation and/or government uses.

3.1.4 Results

The next tables shows how many papers were found and the filtering process. The research was conducted between december 18th and 24th of 2018. It is important to note that some results may overlap. For example, Google Scholar may return results that are also returned by IEEE or Springer. It happens so because Google Scholar is a web aggregator. It is not a database of papers in itself.

[3] In case the searches returned wrong results.

Tool	N° of results
IEEE Xplore	423
Google Scholar	10.900
Science Direct	302
Springer Link	605
TOTAL	12.230

Table 2 – Results of queries performed in 18/12/2018

The first step was to decide how many results from each set to use. The first 25 items[4] returned were selected. 100 results in total. Of these, only the abstract was read and, according to the predefined filtering parameters, 25 were selected. After that, the resulting 25 works were read fully. This step resulted in 14 papers that were considered important for this project. Besides that, there was one extra paper that was later suggested by a professor and was fully read.

Article	Implementation	Protocol	Tools used	Government applications
(PALMA et al., 2019)	x	x	x	x
(COSTA et al., 2018)	x			x
(ØLNES; JANSEN, 2017)				x
(YUAN et al., 2017)		x		
(MAGRAHI et al., 2018)		x	x	
(BUCHMANN et al., 2017)				x
(KSHETRI; VOAS, 2018)				x
(LAKMAL et al., 2017)		x		
(CHEN; ZHU, 2017)	x			
(LEMIEUX, 2017)				x
(SULLIVAN; BURGER, 2017)				x
(SOLARTE-RIVERA et al., 2018)	x		x	x
(HELMER et al., 2018)	x		x	
(ZHANG; LIN; XU, 2018)		x		
(NIZAMUDDIN; HASAN; SALAH, 2018)	x		x	

Table 3 – Articles and topics

[4] The results were sorted by “relevance”.

4 Proposal

4.1 Problem

Currently, in Brazil, document validation and authentication is performed by notaries. These notaries are privately managed but state delegated institutions (RODRIGUES, 2013, p. 232). Which means, they work as a private company, but it's legal power is *given* to it by the government. There are lots of notaries scattered around Brazil's territory. A list, downloaded from the Ministry of Justice's website has more than thirteen thousand entries[1]. These include notaries of civil documents, contracts, real state records and others.

The problem arises when notaries need to communicate between them to validate data. For example: someone, born and registered in Manaus, capital city of Amazonas, wants to get married in Florianópolis, capital city of Santa Catarina, approximately 2.900 Km away. To do so, the person goes to a notary in Florianópolis with his or her birth certificate in hand. After receiving the documents, the notary worker assumes that the data in the birth certificate is correct. Yet, if he or she wanted to check the validity of the data, he or she would have to call someone in the respective Manaus notary and ask for confirmation. The worker, in Manaus, would have to check the physical books, required by law[2], and manually check the information before returning the results to the caller.

The process described above is extremely slow, costly and inefficient. Besides, it adds the human variable, which may cause further problems and errors. However there are bigger problems. What if the data is missing from the books? Or worst. What if the books, or any databases, are lost in some sort of natural catastrophe[3]? What if one of the people getting married are already married to someone else? Bigamy is a crime in Brazilian law[4].

The problems and difficulties described above are just some of the possible ones. They are just the result of the not-so-creative imagination of the author. The real world is much weirder and much more complex. This proposal aims to fix problems like the ones described.

---

1 <http://dados.mj.gov.br/dataset/lista-de-cartorios-do-brasil>, accessed on 19/06/2019.

2 Cápitulo 2, Da Escrituração, law 6.015 of 1973.

3 Cartórios de Alagoas perdem todos os livros de registro por causa das chuvas, <https://administradores.com.br/noticias/cartorios-de-alagoas-perdem-todos-os-livros-de-registro-por-causa-das-chuvas>, accessed on 22/06/2019.

4 Art. 235, Código Penal Brasileiro, Decreto-Lei 2.848 de 1940.

4.2 Related works

This section will reference some related works in the area. There will be a brief discussion about the implementations and the differences between these works and this one.

The first work is from Palma et al. (2019). In a similar manner, the authors of this article developed a validation scheme for higher education diplomas. The main point of the article, besides creating an implementation that works in Ethereum, is its automated validation process. When a student gets enrolled in a course, this course has a predefined number of classes to be taken. As time goes by, finished classes are added to a contract representing the student in said course. When the has finished the amount defined, a diploma is automatically emitted by the contracts. Unfortunately, our solution cannot make use of a something that is predefined. Differently from university courses, peoples' lives are not predefined. A person may, or may not, marry. We have no idea when it will marry, die or be born, for that matter. Because of that, we cannot make an automated record emission system. Our proposal will work as set of records that is updated through time.

The second work, by Costa et al. (2018), in a similar way to the previous work, discusses the use of blockchain technologies to register diplomas in a blockchain. Differently from our proposal and Palma et al. (2019), this article does not perform any data validation on the registered documents. It uses the blockchain as a simple ledger. It proposes the creation of publicly available APIs to be used by any entity that wants to authenticate a digital diploma. Our proposal can be adapted to work in a similar way. A service can be created that would serve as a layer between clients[5] and the underlying blockchain service. Our proposal will focus on the blockchain logic, mainly smart contracts implementations.

The third, and last, related work is Magrahi et al. (2018). It defines a protocol for notarizing documents inside a blockchain. The core functionalities are described as document archivability, retrievability and proof of existence. It uses the blockchain as registry of actions in a trusted archiving solution. The data is stored mainly in the archiving solution, but only its metadata is stored in the blockchain itself. This article is very similar to Provchain's (LIANG et al., 2017) solution. It discusses a database solution where every interaction to it is registered in a blockchain network. Both works are very important to guarantee data provenience. But, unfortunately, they do not use the distributed data solutions available. IPFS, for example. Our implementation will make the use of the distributed file system that IPFS provides and store non-essential data in it. Besides, our implementation will be more focused on the data distribution and its validation.

[5] An way for the user to interact with the data and contracts of our proposal.

4.3 Proposed solution

The proposal of this project is to create a distributed, data validation and storage, network using blockchain technologies. It provides an append only distributed data structure. When a document is inserted in the blockchain, it is available to every other node in the network, making it safely accessible for every participant. The data validation logic is implemented using smart contracts and the data validation is performed based on the preexisting data already in the blockchain.

The idea is as follows. Every notary is a node in the distributed blockchain network. This network may be public or private. Each node would have its respective smart contract instance. This contract would be the interaction entry point of the notary. It represents the institution, and the records it contains. Figure 5 shows a basic, high-level, view of the proposed system. The documents are registered into the notaries, represented by the green icons. Which, in part, register them into the blockchain (centre). Every document registered in any notary would be available in the blockchain network[6].

Every notary is connected to the blockchain network. And every document that gets registered in it will be added to this network. The benefits are quite big. First, every node will have a copy of the data. If, by some reason, one of the notaries gets destroyed, the data will be safe in every single one of the nodes.

Another point is, after any record is placed in the network, every single node will be able to audit it. At a later time. If it is a public network, any third party could audit the document in question. For free [7].

The above explanation is quite informal and broad at the moment. But it will be further explained in the next sections. There are some limitations that will be discussed as well.

4.3.1 Considerations

4.3.1.1 Limitations

To keep the scope of this project at hand, some limitations are going to be set beforehand: (a) Secrecy of documents is not part of the scope of this proposal; (b) Every single document and data will be considered public knowledge; (c) Security concerns regarding non blockchain related processes, like user permissions, database accesses, etc; will not be part of this proposal.

For item (a), there are legal situations in Brazil's law that some documents are not public knowledge and are kept secret from public. Item (b) is related to item (a). All civil

[6] The blockchain is presented as an unique entity. But, as described in previous chapters, it is distributed. This was just a presentation choice.

[7] Only possible because of Solidity's view functions.

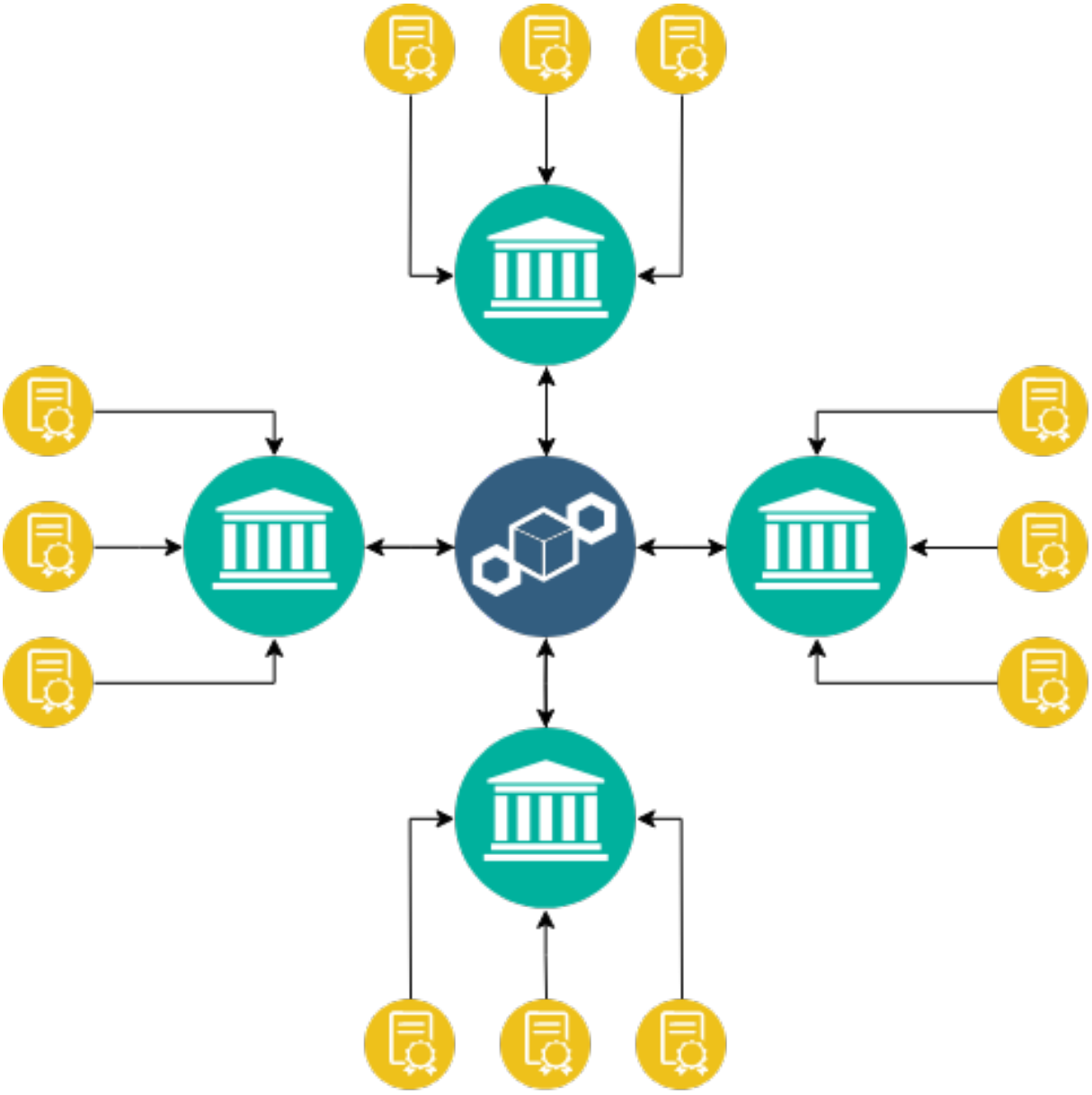


Figure 5 – High level overview of the system.

documents, birth and marriage certificates, for example, are considered public and can requested by any other person. Finally, item (c) is to not consider problems that arise in other systems. Possible hacks, steal/lost of credentials, etc.

4.3.1.2 Requisites

Some requisites are going to be set as well: (a) The designed system should be able to incorporate already existing documents into itself; (b) The data in the blockchain will have a link to some another physical document, if it exists;

For (a), it is important that the current notary system is able, in a considerably easy way, to migrate to the proposed system. The easier the migration process is, the faster

it can be accepted by everyone and quickly implemented. Item (b) is concerned about the migration process of (a) itself. It is not cheap to store all document's data into Ethereum. Instead, it is way cheaper to store only some way of locating the document outside of the blockchain. A database identification number or an IPFS hash, for example.

5 Implementation

Here the implementation of the project will be discussed more in depth.

5.1 Model

The model of the project consists in three basic classes. Notary, record and an authority. For the sake of explaining the implementation using more common terms, we will refer to the contracts as classes in this chapter. So, when saying “the record class” we really mean “the record smart contract”.

Figure 6 shows an UML representation of the model. The notary is used as a simple set of information. It contains the set of registered records and a set of its officials. If a document is registered in it, it is considered valid. The second set is one of authorized entities. These entities represent people who have the legal power to add records to the notary. Last, the record class represents a generic record that can be stored in the notary.

5.2 Architecture

The system was created by trying to mimic the functionality of a physical notary. There will always be a subject, a person, which the records are related to. There are, also, the authorities of the notary. Theses people are the only ones that have legal power to register or edit records of a notary. And the notary itself. The institution. Each one of these contracts will be further explained below.

5.2.1 Notary

The notary contract is one of the simplest contracts implemented. It consists of only two sets of information. One of the registered records and the other of the notary officials. The first one is used to store all the addresses of registered documents. If a document is registered, it is considered valid. Just like their physical counterpart. The second set, in

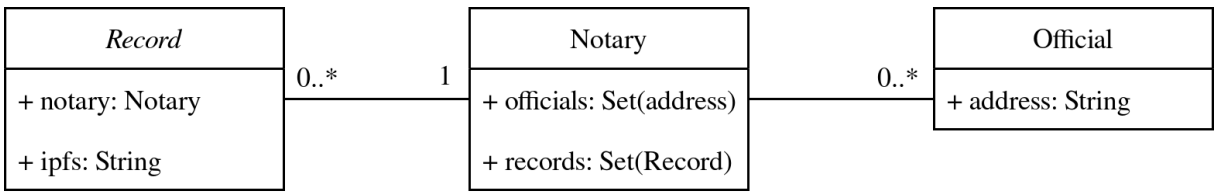


Figure 6 – UML diagram of classes implemented.

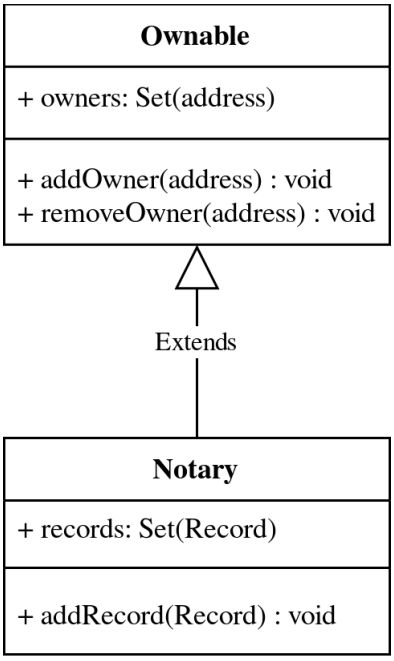


Figure 7 – Class diagram of Notary contract.

the other hand, is used to store the addresses of people who have authority in the notary. This is done so to prevent unauthorised third parties to register invalid documents.

However, the current implementation has a flaw. Every single official registered in the notary has the same permissions. That is, they can add and remove other officials and documents. Though, this could be easily fixed by adding a new layer of permissions. Another contract, controlled by some government institution, for example, that contains the sole power of adding or removing officials from notaries. For simplicity, this feature was left out and we assume all officials are trustworthy. Figure 7 shows the related methods and properties.

The Ownable contract is a common pattern in Solidity contracts(WORLEY; SKJELLUM, 2018). It is a base class that uses a modifier to control access to parts of the contract. The Owners set contains the addresses of all users which have permissions. Our notary inherits from this contract. In this manner, only the appointed authorities can modify the state of the system. This is, add or edit records.

5.2.2 Record

The record is where data is stored. This is an abstract contract and should be used with inheritance to create more specialised contracts. Figure 8 shows the derived classes implemented for this project.

The validate method is used to validate the data of a contract and it is implemented in every other subclass of Record. It should not be called from within the blockchain. It is

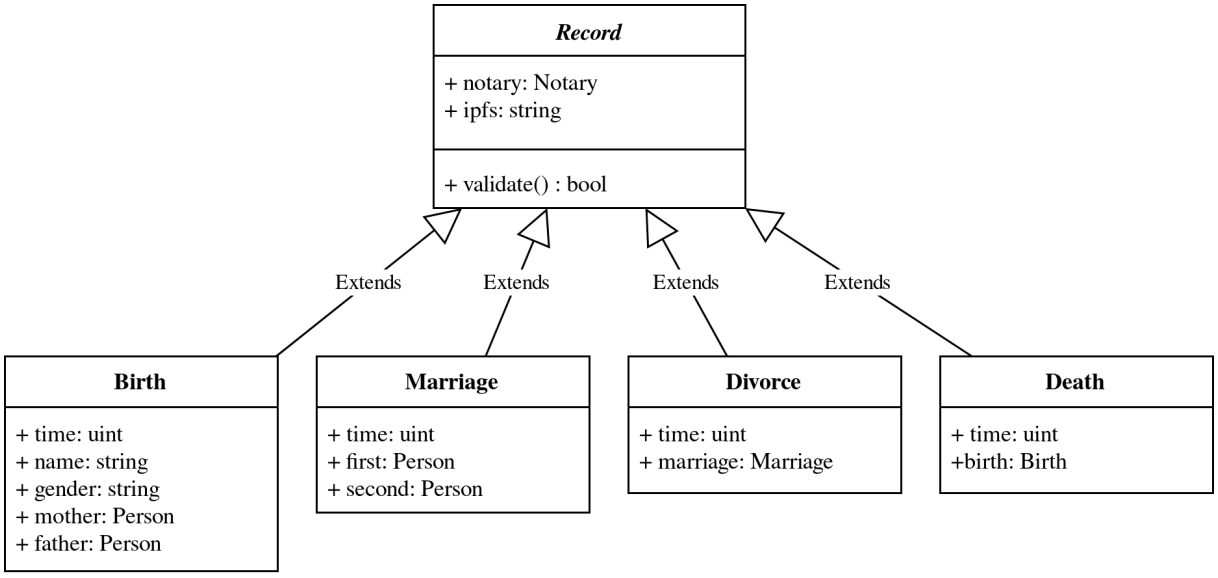


Figure 8 – Class diagram of Record, and derived, contracts.

an expensive method. For example, the birth contract. The data in it already makes it possible to do some data validations. For example, if the mother wasn't alive at the time of birth or if the father wasn't born at the time of birth, the document is invalid.

As pointed out in section 4.3.1.2, both items, it should be easy to migrate from current system to this one. In order to do that, there should be a way to link each record to some sort of document. Physical or digital. Each record has an IPFS hash attribute that store the location of some document. IPFS was chosen as the data storing for the sake of simplicity. Instead, some database information, like an unique id, could be stored instead.

5.2.3 Person

The person contracts represents an individual. A person to which records will be related. This contract is used as an anchor point to validate data from records. It contains references to possible contracts that a person may generate throughout his or her life. A birth; one or more marriages; an equal number, or one less, of divorces compared to marriages; a death.

The methods implemented are just utility methods. They are used by the sub classes of the record class to make data validation simpler. Figure 9 shows the UML diagram of the person record.

5.3 Validation

In this section we will discuss the operational costs of the prototype and draw conclusions on the applicability of such system. We will first discuss its operational

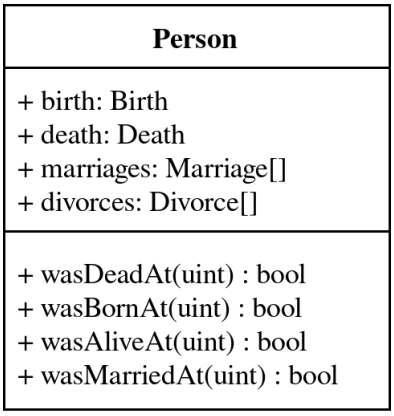


Figure 9 – Class diagram of Person contract.

Notary	Transaction cost	Execution cost
new Notary	300237	189285
addOwner	43657	20977
Total	343894	210262
Cost	R$ 0.2421632769	R$ 0.1480622952

Table 4 – Transaction costs for contracts.

costs. After, we will compare these costs with the current model used in Brazil and draw conclusions about the applicability of the prototype.

5.3.1 Costs

The first thing we need to see is the deployment costs of such prototype. All of the costs listed here were obtained by deploying the contracts in a custom network using Remix IDE[1]. All of the costs described here are considering the price of Ethereum on the day 19/10/2019, which was U$D 171.2 at 17:48. Which is R$ 704.18. All the transactions and executions costs are represented by gas, and the gas price was considered the base gas price of 1 gwei. For every record registered, we also included the linking of an IPFS hash. According to section 4.3.1.2, this is one of the requisites imposed to this project.

5.3.1.1 Notary

First, to deploy the contracts into Ethereum, there is a cost. it is possible to make an analogy to the real world as "finding the correct location" to place our notary. This cost will be incurred only once, at the beginning. Table 4 shows the price for deploying a single Notary contract and adding a responsible account for it. As we can see, the deployment incur in a very low cost. Around 39 cents.

[1] <https://remix.ethereum.org/>, accessed in 19/10/2019.

Birth	Transaction cost	Execution cost
new Person	1218369	876437
notary.addRecord	45117	22501
new Birth	931399	661223
setName	45452	23476
setGender	45334	23486
setTime	44130	22602
setMother	45479	22863
setFather	45503	22887
setIpfs	88488	63632
person.setBirth	48450	25770
notary.addRecord	45181	22501
Total	2602902	1787378
Cost	R$ 1.83291153	R$ 1.25863584

Table 5 – Birth registering costs.

5.3.1.2 Birth

The creation and registering of a birth is the most expensive one in this system. That is because we need to create 2 records for it. One Person contract and a Birth contract. This step is exclusive to the birth registering process. We do not need to create a new person when registering a marriage, for example. The costs shown in table 5. The total cost to register a newborn is around R$ 3.10. This considerably high cost is mainly because of the creation of a new Person contract, which is one of the most complex contracts in the system.

5.3.1.3 Marriage

After the birth record, marriage registering is the most complex and expensive process. It follows the same steps as the birth registering but is is simpler. We do not need to create a Person contract, it already exists. Besides, there is less data to be stored in it as well. We only have to store a time and the address of the people involved. The costs shown in table 6. The total cost to register a new marriage is around R$ 1.25.

5.3.1.4 Death and divorce

Both contracts, Death and Divorce, are included in the same subsection because they are very similar. Even the data related to each one of them is very similar. The process to register either one is the same. Creation, adding the record, passing a time and saving to the notary. The following tables, 7 and 8, show the costs of registering a death and a divorce, respectively. The total cost to register them is R$ 0.95 and R$ 0.98 respectively.

Marriage	Transaction cost	Execution cost
new Marriage	734247	512671
setFirst	45457	22841
setSecond	45524	22908
setTime	44108	22580
setIpfs	88509	63653
person.setMarriage	68492	45812
notary.addRecord	45181	22501
Total	1071518	712966
Cost	R$ 0.7545415452	R$ 0.5020563979

Table 6 – Marriage registering costs

Death	Transaction cost	Execution cost
new Death	550068	374140
setBirth	45587	22907
setTime	44108	22580
setIpfs	88443	63587
person.setDeath	48429	25749
notary.addRecord	45181	22501
Total	821816	531464
Cost	R$ 0.5787063909	R$ 0.3742463195

Table 7 – Death registering costs

Divorce	Transaction cost	Execution cost
new Divorce	550324	374140
setMarriage	45544	22864
setTime	44130	22602
setIpfs	88487	63631
person.setDivorce	68557	45877
notary.addRecord	45181	22501
Total	842223	551615
Cost	0.5930765921	0.3884362507

Table 8 – Divorce registering costs

Fact	Occurrences	Total cost
Birth	98978	R$ 305,995.18
Death	39406	R$ 37,552.05
Marriage	34098	R$ 42,847.48
Divorce	8556	R$ 8,397.82
Total		R$ 394,792.53

Table 9 – Yearly number of registries, by category, and costs.

5.3.2 Comparison

According to Conselho Nacional de Justiça (CNJ), the declared revenue of Santa Catarina's notaries, in the last 6 months, was around 370 million Brazilian reais[2]. Of these, about 95 million were spent in natural persons registries. We will use the 95 million as a base to our calculations here because our prototype was built to mimic natural persons documents. We will also double its value to consider a full year, totalling 190 million reais. It is valid to note that the system implemented here could be further developed to encompass all of the notary system, but we are only concerned with this part for now.

Of the 190 million reais, about 35% is considered operational costs (LUIZARI, 2019). Which means, around 66 million reais are spent, each year, only in the state of Santa Catarina, only with natural persons documents, just to operate the notaries.

If we consider the number of births, deaths, marriages and divorces from Instituto Brasileiro de Geografia e Estatística (IBGE), we can draw very interesting conclusions about the cost of running this prototype in large scale. Using data gathered from public government datasets[3], we made a very good estimation of the system cost in a whole year scale. Table 9 uses data from year 2017. It shows that, if 100% of the state used our system, the cost would be very low when comparing to the current system.

Of course, our calculations does not take into account the costs of physical space, labour etc. We are only considering if the system was already in place, fully operational, its cost related to the blockchain network. The value of 394 thousand reais could be further brought down. This value considers the price of Ethereum. Which is expensive. Using a private network, like Quorum, this cost could be brought down to basically the costs of electricity and server maintenance.

[2] <https://www.cnj.jus.br/corregedoria/justica_aberta>, accessed on 19/10/2019.
[3] <https://www.ibge.gov.br/estatisticas/sociais/populacao/>, accessed in 19/10/2019.

6 Conclusion

As we can see, blockchain technologies are revolutionising the way business, governments and many other categories of industries work. It created a way of doing business logic without the need of a centralised point. However, one of the best uses is the governmental one. Public institutions which operate using a public accessible blockchain can be audited by anyone, anywhere. The way it should be.

This project showed that it is possible to bring costs down. A fully functioning system, when in place, would work at a fraction of the cost, reducing bureaucracy and making data easily accessible by everyone with internet connection. Of course, this vision is a very naive one. The process of implementing such system has many, many more details that do not fit the scope of this project. There are the problems of document secrecy, in cases of judicial decision. Problems of adoptions, where the documents must not be fully disclosed to anyone.

These are only the practical problems. The biggest challenge would be the legal challenge. Bitcoin has popularised the term "blockchain". Unfortunately, lawmakers and the population in general do not understand it fully. The media usually draws attention to the bad uses of bitcoin, such as ransomware. This makes the acceptance process a very slow and difficult one.

However, economics usually prevails in these cases. As this study showed, the use of blockchain for notary systems could improve the quality of service and bring new, unseen, features to it. A highly distributed, public and freely accessible dataset of public records. This implementation is only for notaries, but it could be applied to many more sectors. Government contracts, transparency, elections. These are just some of the possibilities.

Bibliography

ANDROULAKI, E. et al. Hyperledger fabric: a distributed operating system for permissioned blockchains. In: ACM. *Proceedings of the Thirteenth EuroSys Conference*. [S.l.], 2018. p. 30.

BECKER, G. Merkle signature schemes, merkle trees and their cryptanalysis. *Ruhr-University Bochum, Tech. Rep*, 2008.

BENET, J. Ipfs-content addressed, versioned, p2p file system. *arXiv preprint arXiv:1407.3561*, 2014.

BRASIL. *Lei nº 6.015 de 1973*. 1973. Disponível em: <http://www.planalto.gov.br/ccivil_03/Leis/L6015compilada.htm>. Acesso em: 26 fev. 2019.

BUCHMANN, N. et al. Enhancing breeder document long-term security using blockchain technology. In: IEEE. *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*. [S.l.], 2017. v. 2, p. 744–748.

CHEN, Z.; ZHU, Y. Personal archive service system using blockchain technology: case study, promising and challenging. In: IEEE. *2017 IEEE International Conference on AI & Mobile Services (AIMS)*. [S.l.], 2017. p. 93–99.

COSTA, R. et al. Uso não financeiro de blockchain: Um estudo de caso sobre o registro, autenticação e preservação de documentos digitais acadêmicos. In: SBC. *Anais do I Workshop em Blockchain: Teoria, Tecnologias e Aplicações (WBlockchain-SBRC 2018)*. [S.l.], 2018. v. 1, n. 1/2018.

DIFFIE, W.; HELLMAN, M. New directions in cryptography. *IEEE transactions on Information Theory*, IEEE, v. 22, n. 6, p. 644–654, 1976.

DOBBERTIN, H. The status of md5 after a recent attack. *Crypto-Bytes The technical newsletter of RSA Laboratories, a division of RSA Data Security, Inc.*, v. 2, n. 2, 1996.

HABER, S.; STORNETTA, W. S. How to time-stamp a digital document. In: SPRINGER. *Conference on the Theory and Application of Cryptography*. [S.l.], 1990. p. 437–455.

HELMER, S. et al. Ethernitydb–integrating database functionality into a blockchain. In: SPRINGER. *European Conference on Advances in Databases and Information Systems*. [S.l.], 2018. p. 37–44.

JÚNIOR, P. d. C. *Profissão de Escreventes Habilitadi - Funções Notariais*. [S.l.: s.n.], 1961.

KITCHENHAM, B. Procedures for performing systematic reviews. *Keele, UK, Keele University*, v. 33, n. 2004, p. 1–26, 2004.

KSHETRI, N.; VOAS, J. Blockchain-enabled e-voting. *IEEE Software*, IEEE, v. 35, n. 4, p. 95–99, 2018.

LAKMAL, C. et al. Idstack—the common protocol for document verification built on digital signatures. In: IEEE. *2017 National Information Technology Conference (NITC)*. [S.l.], 2017. p. 96–99.

LEMIEUX, V. L. A typology of blockchain recordkeeping solutions and some reflections on their implications for the future of archival preservation. In: IEEE. *2017 IEEE International Conference on Big Data (Big Data)*. [S.l.], 2017. p. 2271–2278.

LIANG, X. et al. Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability. In: IEEE PRESS. *Proceedings of the 17th IEEE/ACM international symposium on cluster, cloud and grid computing*. [S.l.], 2017. p. 468–477.

LUIZARI, L. Repasses e despesas: Para onde vai o dinheiro pago aos cartórios brasileiros? *Cartórios com Você*, n. 8, p. 25, Jul 2019.

MAGRAHI, H. et al. Nfb: A protocol for notarizing files over the blockchain. In: IEEE. *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*. [S.l.], 2018. p. 1–4.

MARSAGLIA, G.; TSANG, W. W. et al. Some difficult-to-pass tests of randomness. *Journal of Statistical Software*, v. 7, n. 3, p. 1–9, 2002.

MORGAN, J. Quorum whitepaper. *New York: JP Morgan Chase*, 2016.

MUSTAPICH, J. M. *Revista Notarial Brasileira - nº. 1*. [S.l.: s.n.], 1974.

NAKAMOTO, S. Bitcoin: A peer-to-peer electronic cash system. Working Paper, 2008.

NARAYANAN, A. *Bitcoin and cryptocurrency technologies : a comprehensive introduction*. Princeton, New Jersey: Princeton University Press, 2016. ISBN 0691171696.

NETO, A. C. R. O alcance social da função notarial no brasil. *Florianópolis: Conceito Editorial*, 2008.

NIZAMUDDIN, N.; HASAN, H. R.; SALAH, K. Ipfs-blockchain-based authenticity of online publications. In: SPRINGER. *International Conference on Blockchain*. [S.l.], 2018. p. 199–212.

ØLNES, S.; JANSEN, A. Blockchain technology as s support infrastructure in e-government. In: SPRINGER. *International Conference on Electronic Government*. [S.l.], 2017. p. 215–227.

PALMA, L. M. et al. Blockchain and smart contracts for higher education registry in brazil. *International Journal of Network Management*, Wiley Online Library, p. e2061, 2019.

PUGLIESE, R. J. *Direito Notarial Brasileiro*. [S.l.: s.n.], 1989.

RODRIGUES, M. G. *Tratado de registros públicos e direito notarial*. [S.l.]: Editora Atlas SA, 2013.

SANCHES, A. T. *A questão de terras no início da República: o Registro Torrens e sua (in) aplicação*. Tese (Doutorado) — Universidade de São Paulo, 2008.

SOLARTE-RIVERA, J. et al. Document management system based on a private blockchain for the support of the judicial embargoes process in colombia. In: SPRINGER. *International Conference on Advanced Information Systems Engineering*. [S.l.], 2018. p. 126–137.

STALLINGS, W. *Cryptography and network security : principles and practice*. Boston: Prentice Hall, 2011. ISBN 0136097049.

SULLIVAN, C.; BURGER, E. E-residency and blockchain. *Computer Law & Security Review*, Elsevier, v. 33, n. 4, p. 470–481, 2017.

SZABO, N. Formalizing and securing relationships on public networks. *First Monday*, v. 2, n. 9, 1997.

TIZIANI, M. G. Uma breve história do registro civil contemporâneo. 2016.

WENSLEY, J. H. et al. Sift: Design and analysis of a fault-tolerant computer for aircraft control. *Proceedings of the IEEE*, IEEE, v. 66, n. 10, p. 1240–1255, 1978.

WOOD, G. et al. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, v. 151, p. 1–32, 2014.

WORLEY, C. R.; SKJELLUM, A. Opportunities, challenges, and future extensions for smart-contract design patterns. In: SPRINGER. *International Conference on Business Information Systems*. [S.l.], 2018. p. 264–276.

YUAN, C. et al. Blockchain with accountable cp-abe: how to effectively protect the electronic documents. In: IEEE. *2017 IEEE 23rd international conference on parallel and distributed systems (ICPADS)*. [S.l.], 2017. p. 800–803.

ZHANG, Y.; LIN, X.; XU, C. Blockchain-based secure data provenance for cloud storage. In: SPRINGER. *International Conference on Information and Communications Security*. [S.l.], 2018. p. 3–19.

APPENDIX A – Implementation

This is the source code of the smart contracts developed in the project. The notary contract represents the institution. The factory is a design patter used to create records. The record contract represents the documents of the notary and its relations to other contracts.

```
1  pragma solidity ^0.5.0;
2
3  contract Birth is Record {
4      uint public time;
5      string public name;
6      string public gender;
7      Person public mother;
8      Person public father;
9
10     constructor (Notary n) Record(n) public {}
11
12     function validate() public view returns (bool) {
13         // Mother alive on birth time
14         if (!mother.wasAliveAt(time))
15             return false;
16
17         // Father already born
18         if (!father.wasBornAt(time))
19             return false;
20
21         return true;
22     }
23
24     function setTime(uint _time) onlyOfficials public { time = _time; }
25     function setName(string memory _name) onlyOfficials public { name =
           _name; }
26     function setGender(string memory _gender) onlyOfficials public {
           gender = _gender; }
27     function setMother(Person _mother) onlyOfficials public { mother =
           _mother; }
28     function setFather(Person _father) onlyOfficials public { father =
           _father; }
29 }
30
31 contract Death is Record {
32     uint public time;
33     Birth public birth;
34
```

```
35      constructor (Notary n) Record(n) public {}
36
37      function validate() public view returns (bool) {
38          return birth.time() < time;
39      }
40
41      function setTime(uint _time) onlyOfficials public { time = _time; }
42      function setBirth(Birth _birth) onlyOfficials public { birth =
            _birth; }
43  }
44
45  contract Divorce is Record {
46      uint public time;
47      Marriage public marriage;
48
49      constructor (Notary n) Record(n) public {}
50
51      function validate() public view returns (bool) {
52          return marriage.time() < time;
53      }
54
55      function setTime(uint _time) onlyOfficials public { time = _time; }
56      function setMarriage(Marriage _marriage) onlyOfficials public {
            marriage = _marriage; }
57  }
58
59  contract Marriage is Record {
60      uint public time;
61      Person public first;
62      Person public second;
63
64      constructor (Notary n) Record(n) public {}
65
66      function validate() public view returns (bool) {
67          // Checks if they were alive
68          if (!first.wasAliveAt(time) || !second.wasAliveAt(time))
69              return false;
70
71          // Checks if they were married
72          if (first.wasMarriedAt(time) || second.wasMarriedAt(time))
73              return false;
74
75          return true;
76      }
77
78      function setTime(uint _time) onlyOfficials public { time = _time; }
```

```
79      function setFirst(Person _first) onlyOfficials public { first =
            _first; }
80      function setSecond(Person _second) onlyOfficials public { second =
            _second; }
81  }
82
83  contract Migrations {
84    address public owner;
85    uint public last_completed_migration;
86
87    constructor() public {
88      owner = msg.sender;
89    }
90
91    modifier restricted() {
92      if (msg.sender == owner) _;
93    }
94
95    function setCompleted(uint completed) public restricted {
96      last_completed_migration = completed;
97    }
98
99    function upgrade(address new_address) public restricted {
100     Migrations upgraded = Migrations(new_address);
101     upgraded.setCompleted(last_completed_migration);
102   }
103 }
104
105 contract Notary is Ownable {
106     mapping(address => bool) public records;
107
108     event Created(address creator, address record, uint time);
109
110     function addRecord(Record r) onlyOwners public {
111         records[address(r)] = true;
112         emit Created(msg.sender, address(r), now);
113     }
114 }
115
116 contract Ownable {
117     mapping(address => bool) public owners;
118
119     constructor() public {
120         owners[msg.sender] = true;
121     }
122
123     function addOwner(address newOwner) onlyOwners public {
```

```
124         owners[newOwner] = true;
125     }
126
127     function removeOwner(address oldOwner) onlyOwners public {
128         owners[oldOwner] = false;
129     }
130
131     modifier onlyOwners {
132         require(owners[msg.sender], "Only owners can call");
133         _;
134     }
135 }
136
137 contract Person is Record {
138     Birth public birth;
139     Death public death;
140     Marriage[] public marriages;
141     Divorce[] public divorces;
142
143     event Fact(string category, address creator, address record, uint
        time);
144
145     constructor(Notary notary) Record(notary) public {}
146
147     function validate() public view returns (bool) {
148         // Person should have birth, at least
149         require(address(birth) != address(0), "Person should have birth"
            );
150
151         // Birth
152         if (!birth.validate())
153             return false;
154
155         // Marriages
156         for (uint i = 0; i < marriages.length; i++)
157             if (!marriages[i].validate())
158                 return false;
159
160         // Divorces
161         for (uint i = 0; i < divorces.length; i++)
162             if (!divorces[i].validate())
163                 return false;
164
165         // Death
166         if (address(death) != address(0) && !death.validate())
167             return false;
168
```

```
169         return true;
170     }
171
172     function setBirth(Birth _birth) onlyOfficials public {
173         birth = _birth;
174         emit Fact("Birth", msg.sender, address(birth), now);
175     }
176
177     function setDeath(Death _death) onlyOfficials public {
178         death = _death;
179         emit Fact("Death", msg.sender, address(death), now);
180     }
181
182     function setMarriage(Marriage _marriage) onlyOfficials public {
183         marriages.push(_marriage);
184         emit Fact("Marriage", msg.sender, address(_marriage), now);
185     }
186
187     function setDivorce(Divorce _divorce) onlyOfficials public {
188         divorces.push(_divorce);
189         emit Fact("Divorce", msg.sender, address(_divorce), now);
190     }
191
192     function wasDeadAt(uint time) public view returns (bool) {
193         if (address(death) == address(0))
194             return false;
195         return death.time() < time;
196     }
197
198     function wasBornAt(uint time) public view returns (bool) {
199         if (address(birth) == address(0))
200             return false;
201         return birth.time() < time;
202     }
203
204     function wasAliveAt(uint time) public view returns (bool) {
205         return wasBornAt(time) && !wasDeadAt(time);
206     }
207
208     function wasMarriedAt(uint time) public view returns (bool) {
209         for (uint i = 0; i < marriages.length; i++) {
210             // Married before
211             if (marriages[i].time() < time) {
212                 // No divorce for this marriage
213                 if (divorces.length <= i)
214                     return true;
215
```

```
216                 // Divorce happened after
217                 if (divorces[i].time() > time)
218                     return true;
219             }
220         }
221 
222         return false;
223     }
224 }
225 
226 contract Record {
227     Notary public notary;
228     string public ipfs;
229 
230     constructor(Notary _notary) public {
231         notary = _notary;
232     }
233 
234     function setIpfs(string memory _ipfs) onlyOfficials public {
235         ipfs = _ipfs;
236     }
237 
238     function validate() public view returns (bool);
239 
240     modifier onlyOfficials {
241         require(notary.owners(msg.sender), "Only notary officials can
                call");
242         _;
243     }
244 
245 }
```

APPENDIX B – Paper

Document validation using blockchain

A validation scheme for natural person's documents

João Vicente Meyer, Lucas Palma, Jean Everson Martina

Departamento de Informática e Estatística
Universidade Federal de Santa Catarina (UFSC) – Florianópolis, SC – Brazil

{1994meyer,lucaspalma.m}@gmail.com, jean.martina@ufsc.br

Resumo. *O sistema notarial brasileiro é grande. Isto se deve, principalmente, ao tamanho do país. Tanto em território quanto em população. O resultado disto é um sistema muito complexo e lento. Tecnologias de blockchain podem ser de grande ajuda neste cenário. Ela nos dá uma maneira distribuída de guardar e validar data entre diversos participantes. Neste caso, os cartórios. Este projeto desenvolve um protótipo capaz de armazenar e validar registros públicos de pessoas naturais em uma blockchain. O protótipo engloba os documentos de nascimento, casamento, divórcio e óbito. Ao final, este projeto demonstra os custos operacionais do protótipo e realiza uma comparação com o sistema cartorário utilizado atualmente.*

Abstract. *The Brazilin public notary system is very big. Mainly because of the sheer scale of the country. In territory and in population. The result is a very complex and slow system. Blockchain technology can be of huge help in this scenario. It provides a distributed way of storing and validating data between many players. In this case, the notaries.This project creates a prototype that is capable of storing and validating public records of natural persons in a blockchain. The prototype encompasses the birth, marriage, divorce and death records. In the end, this project shows the operational costs of such prototype and makes comparisons with the current notary system.*

1. Introduction

The Brazilian notary system is composed of over 13 thousand institutions [1]. They are privately controlled and have the legal power provided by the state [2]. Unfortunately, this huge network still uses archaic types of communication between its peers and paper to store most of its records. Furthermore, Brazilian law states that every natural person's registered document can be accessed by any person interested in it [3]. It could be said that the notary system is like a set of databases scattered throughout the Brazilian territory. Each one using its own schema and formalities.

The problem arises when there is the need of communication between all of the institutions. Usually, this is a manual process. A human needs to make a request to another notary for a piece of document or information. This makes the process very

error prone, slow and expensive. In addition, there is the possibility of losing documents or a notary building, and all its documents, being lost in some catastrophe. Another problem is the scattering of data. Only notaries at the place of birth of some person are obligated to have the info about said birth [3].

2. Basic concepts

2.1. Public notaries

Public notaries have the function of providing authenticity to contracts, business and all the parts involved in such actions. In this view, “notary law is the law of authenticity and format” (our translation) [4]. When a notary authenticates a document, it is officially recognising it as a valid and truthful document [5]. This will be the main view used throughout this project.

2.2. Blockchain

A blockchain is a chain of blocks. The blocks can be made of any data. The first published work that made use of blockchain was in 1990 [6]. It devised a way to create tamper-proof time stamps for computer files. The big difference between the modern blockchains and the one devised in 1990 is the network distribution of it.

Every node in the network (a P2P network) has an identical copy of the blockchain. Every change performed in one of them is sent to every other node. Unfortunately, a big problem arises from it. How to guarantee that the changes made are valid/trustworthy? This is famously known as the Byzantine Fault [7]. To solve this problem, there is the use of consensus algorithms.

2.2.1. Consensus

Consensus algorithms are used to achieve, as the name indicates, consensus on some data, transaction or communication shared between nodes. Nakamoto (2008), published the incentive based consensus algorithm together with the Bitcoin proposal [8]. The base idea is that there must be incentives for the entities in the network to play by the rules.

2.3. Smart contracts

It was first conceptualised by Szabo (1997). It works on the idea that a blockchain network can agree in some random pieces of data to be added in the list of blocks, it can also validate computer programs executions [9]. As the execution is limited to only use data inside the chain itself, the execution of these computer programs should be the same in every node that has the same chain. This logic is used to validate smart contracts in the network. As a consequence, every single state that the contract will ever have will be permanently stored in the blockchain.

Smart contracts provide a new paradigm where participants do not need to rely on any third parties to “force” the execution of the contract's clauses. The whole network of nodes will act as the “enforcer”. The Ethereum network is the most famous

example of a smart contract based blockchain. It provides a virtual machine that executes, and charges for it, code on every single node of the network.

2.4. Tools

2.4.1. Ethereum

Ethereum [10] is a blockchain platform and distributed computing system. It was released in 2015 and it is fully open-source. Released under the GPL v3 license. The main feature of Ethereum is its smart-contracts. The Solidity language is used to write said contracts using a high level language. They are compiled to bytecode that runs in the Ethereum Virtual Machine (EVM). The language is based in Javascript.

2.4.2. IPFS

The Interplanetary File System (IPFS) [11] is a peer-to-peer (P2P) distributed file system. IPFS creates a hash of every single file stored in it. The files are, subsequently, accessed using these same hashes. Besides, it also features file versioning, similar to *Git*, and duplicate file removal. Its uses are mostly for creating distributed file sharing services.

It is also widely used coupled with Ethereum. When users need to store data in Ethereum, instead of storing it whole (expensive), only the IPFS hash is stored. This method saves considerable amounts of gas, Ethereum's costs for code execution.

3. Related works

The first work is from [12]. In a similar manner, the authors of this article developed a validation scheme for higher education diplomas. The main point of the article, besides creating an implementation that works in Ethereum, is its automated validation process. When a student gets enrolled in a course, this course has a predefined number of classes to be taken. As time goes by, finished classes are added to a contract representing the student in said course. When the has finished the amount defined, a diploma is automatically emitted by the contracts. Unfortunately, our solution cannot make use of something that is predefined. Differently from university courses, peoples' lives are not predefined. A person may, or may not, marry. We have no idea when it will marry, die or be born, for that matter. Because of that, we cannot make an automated record emission system. Our proposal will work as a set of records that is updated through time.

The second work, by [13], in a similar way to the previous work, discusses the use of blockchain technologies to register diplomas in a blockchain. Differently from our proposal and [12], this article does not perform any data validation on the registered documents. It uses the blockchain as a simple ledger. It proposes the creation of publicly available APIs to be used by any entity that wants to authenticate a digital diploma. Our proposal can be adapted to work in a similar way. A service can be created that would serve as a layer between clients (a way for the user to interact with the data and contracts of our proposal) and the underlying blockchain service. Our proposal will focus on the blockchain logic, mainly smart contracts implementations.

The third, and last, related work defines a protocol for notarizing documents inside a blockchain [14]. The core functionalities are described as document archivability, retrievability and proof of existence. It uses the blockchain as registry of actions in a trusted archiving solution. The data is stored mainly in the archiving solution, but only its metadata is stored in the blockchain itself. This article is very similar to Provchain's [15] solution. It discusses a database solution where every interaction to it is registered in a blockchain network. Both works are very important to guarantee data provenience. But, unfortunately, they do not use the distributed data solutions available. IPFS, for example. Our implementation will make use of the distributed file system that IPFS provides and store non-essential data in it. Besides, our implementation will be more focused on the data distribution and its validation.

4. Proposal

The proposal of this project is to create a distributed, data validation and storage, network using blockchain technologies. It provides an append only distributed data structure. When a document is inserted in the blockchain, it is available to every other node in the network, making it safely accessible for every participant. The data validation logic is implemented using smart contracts and the data validation is performed based on the preexisting data already in the blockchain.

4.1. Limitations

To keep the scope of this project at hand, some limitations are going to be set beforehand: (a) Secrecy of documents is not part of the scope of this proposal; (b) Every single document and data will be considered public knowledge; (c) Security concerns regarding non blockchain related processes, like user permissions, database accesses, etc; will not be part of this proposal.

For item (a), there are legal situations in Brazil's law that some documents are not public knowledge and are kept secret from the public. Item (b) is related to item (a). All civil documents, birth and marriage certificates, for example, are considered public and can be requested by any other person. Finally, item (c) is to not consider problems that arise in other systems. Possible hacks, steal/lost of credentials, etc.

4.2. Requisites

Some requisites are going to be set as well: (a) The designed system should be able to incorporate already existing documents into itself; (b) The data in the blockchain will have a link to some other physical document, if it exists;

For (a), it is important that the current notary system is able, in a considerably easier way, to migrate to the proposed system. The easier the migration process is, the faster it can be accepted by everyone and quickly implemented. Item (b) is concerned about the migration process of (a) itself. It is not cheap to store all document's data into Ethereum. Instead, it is way cheaper to store only some way of locating the document outside of the blockchain. A database identification number or an IPFS hash, for example.

5. Implementation

The system was created by trying to mimic the functionality of a physical notary. There will always be a subject, a person, which the records are related to. There are, also, the authorities of the notary. Theses people are the only ones that have legal power to register or edit records of a notary. And the notary itself. The institution. Each one of these contracts will be further explained below. All the code is available in the following url (https://pastebin.com/zEcZ8sDH).

5.1. Notary

The notary contract is one of the simplest contracts implemented. It consists of only two sets of information. One of the registered records and the other of the notary officials. The first one is used to store all the addresses of registered documents. If a document is registered, it is considered valid. Just like their physical counterpart. The second set, on the other hand, is used to store the addresses of people who have authority in the notary. This is done so to prevent unauthorised third parties to register invalid documents.

5.2. Record

The record is where data is stored. This is an abstract contract and should be used with inheritance to create more specialised contracts. It contains a validate method that is used to validate data of a contract and it is implemented in every other subclass of Record. It should not be called from within the blockchain. It is an expensive method. For example, the birth contract. The data in it already makes it possible to do some data validations. For example, if the mother wasn't alive at the time of birth or if the father wasn't born at the time of birth, the document is invalid.

5.3. Person

The person contracts represents an individual. A person to which records will be related. This contract is used as an anchor point to validate data from records. It contains references to possible contracts that a person may generate throughout his or her life. A birth; one or more marriages; an equal number, or one less, of divorces compared to marriages; a death.

6. Validation

The first thing we need to see is the deployment costs of such prototype. All of the costs listed here were obtained by deploying the contracts in a custom network using Remix IDE. All of the costs described here are considering the price of Ethereum on the day 19/10/2019, which was U$D 171.2 at 17:48. Which is R$ 704.18. All the transactions and executions costs are represented by gas, and the gas price was considered the base gas price of 1 gwei. For every record registered, we also included the linking of an IPFS hash, according to the requisites section. Table 1 shows the result for all deployments.

Table 1. Cost of deployment of each smart contract developed

Fact	Cost (BRL)
Birth	3.09154737
Death	0.9529527104
Marriage	1.256597943
Divorce	0.9815128428

According to Conselho Nacional de Justiça (CNJ), the declared revenue of Santa Catarina's notaries, in the last 6 months, was around 370 million Brazilian reais [16]. Of these, about 95 million were spent in natural persons registries. We will use the 95 million as a base to our calculations here because our prototype was built to mimic natural persons documents. We will also double its value to consider a full year, totalling 190 million reais. It is valid to note that the system implemented here could be further developed to encompass all of the notary system, but we are only concerned with this part for now. Of the 190 million reais, about 35% is considered operational costs [17]. Which means, around 66 million reais are spent, each year, only in the state of Santa Catarina, only with natural persons documents, just to operate the notaries.

If we consider the number of births, deaths, marriages and divorces from Instituto Brasileiro de Geografia e Estatística (IBGE), we can draw very interesting conclusions about the cost of running this prototype in large scale. Using data gathered from public government datasets [18], we made a very good estimation of the system cost in a whole year scale. Table 2 uses data from year 2017. It shows that, if 100% of the state used our system, the cost would be very low when compared to the current system.

Table 2. Number of times each fact occurred in a year and the price for all occurrences

Fact	Occurrences	Cost/Year (BRL)
Birth	98978	R$305,995.18
Death	39406	R$37,552.05
Marriage	34098	R$42,847.48
Divorce	8556	R$8,397.82
	TOTAL	R$394,792.53

Of course, our calculations do not take into account the costs of physical space, labour etc. We are only considering if the system was already in place, fully operational, its cost related to the blockchain network. The value of 394 thousand reais could be further brought down. This value considers the price of Ethereum. Which is expensive.

Using a private network, like Quorum, this cost could be brought down to basically the costs of electricity and server maintenance.

7. Conclusion

As we can see, blockchain technologies are revolutionising the way businesses, governments and many other categories of industries work. It created a way of doing business logic without the need of a centralised point. However, one of the best uses is the governmental one. Public institutions which operate using a public accessible blockchain can be audited by anyone, anywhere. The way it should be.

This project showed that it is possible to bring costs down. A fully functioning system, when in place, would work at a fraction of the cost, reducing bureaucracy and making data easily accessible by everyone with an internet connection. Of course, this vision is a very naive one. The process of implementing such a system has many, many more details that do not fit the scope of this project. There are the problems of document secrecy, in cases of judicial decision. Problems of adoptions, where the documents must not be fully disclosed to anyone.

These are only the practical problems. The biggest challenge would be the legal challenge. Bitcoin has popularised the term ``blockchain''. Unfortunately, lawmakers and the population in general do not understand it fully. The media usually draws attention to the bad uses of bitcoin, such as ransomware. This makes the acceptance process a very slow and difficult one.

However, economics usually prevails in these cases. As this study showed, the use of blockchain for notary systems could improve the quality of service and bring new, unseen, features to it. A highly distributed, public and freely accessible dataset of public records. This implementation is only for notaries, but it could be applied to many more sectors. Government contracts, transparency, elections. These are just some of the possibilities.

References

[1] Lista de cartórios do Brasil - <http://dados.mj.gov.br/dataset/lista-de-cartorios-do-brasil>, acessed in 19/06/2019.

[2] RODRIGUES, M. G. Tratado de registros públicos e direito notarial. [S.l.]: Editora Atlas SA, 2013.

[3] BRASIL. Lei nº 6.015 de 1973. 1973. Available in: <http://www.planalto.gov.br/ccivil_03/Leis/L6015compilada.htm>. Accessed in: 26 fev. 2019.

[4] MUSTAPICH, J. M. Revista Notarial Brasileira - nº. 1. [S.l.: s.n.], 1974.

[5] PUGLIESE, R. J. Direito Notarial Brasileiro . [S.l.: s.n.], 1989.

[6] HABER, S.; STORNETTA, W. S. How to time-stamp a digital document. In: SPRINGER. Conference on the Theory and Application of Cryptography . [S.l.], 1990. p. 437–455

[7] WENSLEY, J. H. et al. Sift: Design and analysis of a fault-tolerant computer for aircraft control. Proceedings of the IEEE, IEEE, v. 66, n. 10, p. 1240–1255, 1978.

[8] NAKAMOTO, S. Bitcoin: A peer-to-peer electronic cash system. Working Paper, 2008.

[9] SZABO, N. Formalizing and securing relationships on public networks. First Monday, v. 2, n. 9, 1997.

[10] WOOD, G. et al. Ethereum: A secure decentralised generalised transaction ledger. Ethereum project yellow paper, v. 151, p. 1–32, 2014.

[11] BENET, J. Ipfs-content addressed, versioned, p2p file system. arXiv preprint arXiv:1407.3561, 2014.

[12] PALMA, L. M. et al. Blockchain and smart contracts for higher education registry in Brazil. International Journal of Network Management, Wiley Online Library, p. E2061, 2019.

[13] COSTA, R. et al. Uso não financeiro de blockchain: Um estudo de caso sobre o registro, autenticação e preservação de documentos digitais acadêmicos. In: SBC. Anais do I Workshop em Blockchain: Teoria, Tecnologias e Aplicações (WBlockchain-SBRC 2018). [S.l.], 2018. v. 1, n. 1/2018

[14] MAGRAHI, H. et al. Nfb: A protocol for notarizing files over the blockchain. In: IEEE. 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS). [S.l.], 2018. p. 1–4.

[15] LIANG, X. et al. Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability. In: IEEE PRESS. Proceedings of the 17th IEEE/ACM international symposium on cluster, cloud and grid computing . [S.l.], 2017. p. 468–477.

[16] Justiça Aberta. Available in: <https://www.cnj.jus.br/corregedoria/justica_aberta>, accessed in 19/10/2019.

[17] LUIZARI, L. Repasses e despesas: Para onde vai o dinheiro pago aos cartórios brasileiros? Cartórios com Você, n. 8, p. 25, Jul 2019.

[18] IBGE. Available in: <https://www.ibge.gov.br/estatisticas/sociais/populacao/>, accessed in 19/10/2019.