qChain: A Blockchain Solution Applied to Citizen-Centric Document Automation

Alexandre Gaspar Manso
alexandre.manso@tecnico.ulisboa.pt
Instituto Superior Técnico, Lisboa, Portugal
October 2021

ABSTRACT
Documents are an important method of registering human's data and actions of public and private interest that involve one or more authorities. Document automation technology is useful in facilitating processes related to every stage and aspect of document digitalization and management in a rising digital world. Document automation platforms have been progressing to public use, focusing on critical certificates and agreements, very present in legal departments and government authorities, but slowly, they are migrating to wider audiences, focusing more on easing citizen daily lives. Alike any paper document that needs to be authorized, signed, sealed according to importance, and stored securely, these operations performed in network-based systems, need that same treatment digitally. qDocs is a citizen-centric and multi-organization document automation platform designed for citizens and friendly use, focusing on reducing unneeded and exhaustive processes while assuring digital safety and validation. This feature is the concept and implementation of this thesis, qChain, which intends to integrate the qDocs platform with a blockchain-based system, as an immutable database and integrity validator of the system's reliability and authenticity.

Author Keywords
Document Automation; Blockchain; Multichain; Smart Contracts; qDocs; qChain; Security.

1. INTRODUCTION

This chapter explains the motivation behind this project, presents a short description of the requirements for the solution and its initial prototype. We also refer the objectives of the project, their state of development and its research methodology.

We discuss how document automation has become more relevant nowadays. This refers to a particular sector for automation of critical computer processes such as managing and storing relevant documents in digital format [1]. Therefore, the main purpose for this technology is to help reduce time wasted on these processes while being transparent to the everyday citizen that benefits with this service. Public document automation platforms have been then created, centered on the everyday citizen's needs, but still, the biggest implementations for automation of documents are privative, primarily within legal firms [2].

Parallel to the growth of the previously mentioned market, the blockchain technology eagers to be a sustainable solution for safe transactions of digital goods in the internet, which explains its versatile integration with various IT sectors [3][4]. New platforms supporting blockchain technology are increasingly appearing constantly, and rising new communities of developers, majorly in virtual currency or digital asset transactions.

qDocs [5] is a citizen-centric document automation platform, created to promote this environment of transparently providing an easier alternative for document management to the citizen as a remote service, all in a user-friendly web platform. Entities, called curators, representing official notaries, are responsible for the creation of form-like templates for fast document creation which are then stored locally, close to the official owner, the curator itself, while ubiquitously and unfailingly accessible to the client without any constraint [6]. Nevertheless, if the platform is also responsible for upholding the document's management, storage, and traceability, it's important to implement good security measures to assure the effectiveness of the service, without compromising performance and platform usability.

qDocs is a document automation platform that intends to ease document management in the public sector, accessible to every citizen. In the context of this system, the information flux is in the form of digital documents which are in constant traffic from the user to the curator's databased where they are stored. The user interacts directly with the qDocs platform from which he/she creates and manages documents and stores them alongside the curator, in their database, using an intermediary, qBox. qBox resides between the qDocs main server and the curator and it is used for document retrieval, translating requests for a standardized format holding logical and essential information for document fetching.

The context this thesis focuses the integration and interoperability of qDocs with a blockchain relying on this technology's security properties as an immutable database offering a wide array of advantages over a normal database such as smart contracts.

Therefore, qChain is a blockchain-based system that intends to secure operations performed by qDocs users that are considered critical and important, offering asset integrity, non-repudiation of user operations, secure transactions and

storage processes, in a decentralized network with non-dependance on trusted participants [7].

1.1. Motivation

The motivation behind the integration of document automation and blockchain is to facilitate citizen centered processes such as managing important documents, reducing time wasted on these operations when performed physically, at a notary, for example, which can sometimes be time consuming, while keeping these processes secured.

Consider, the use case of creating a critical digital document using qDocs, in which two public companies wish to emit a partnership contract. This contract also needs government approval and acknowledgement. Two managers from each company have an account in the platform and often use it to create digital documents not relevant enough to be issued in the blockchain. Nevertheless, this document is deemed important and therefore its information should be secured using proper measures. A template may be created, with indications to use blockchain security, be attached to both companies and only for that type of document, using this template, additional security properties are conferred.

We can also consider the use case of a simple citizen card issuance, which the content should not be meddled with after its creation, and consequently should be stored in a blockchain-based environment. The card is signed by the person wishing to create the document and further stamped by the government's official authority as a seal of approval. The two operations This will not only confer stronger integrity enforcement and authenticity but reduce identity fraud attacks, time consumed in a citizen's shops' queues and in mailing them home and increase the throughput for these operations.

These use cases, although different in complexity, can surely benefit from an architecture able to reliably automate the stages related to the time-consuming processes of document management. As explained, qDocs, is a document automation citizen-centered solution, aims to mitigate the long operations that normally are performed by the human hand and, therefore, automating every process document related. However, qDocs does not offer the security measure already discussed in the previous paragraphs that a blockchain-based system surely does, hence the need to integrate both technologies, document automation and blockchain, and promote a more secure and reliable system, with automating capabilities and integrity checking properties while not impacting the user's experience.

1.2. Objectives

This thesis aims to research how to extend and adapt the qDocs platform with blockchain technology, taking advantage of its security properties. Therefore, multiple blockchain technologies were studied, some document automation platforms were analyzed and further integrations of both were analyzed, all to assure the best efficiency in interoperability for this integration.

The main objective for this project is to secure document related operations in the qDocs platform, registering every event from creation and modification to access in a distributed and trustless blockchain-based system, without compromising the normal functionality of qDocs and user experience and conferring integrity, authenticity, non-repudiation, and reliability to qDocs

The solution considers the transparency of the user experience, focusing on maintaining the platform's friendliness and applicability without impacting the user's experience, as the citizen does not even have to be aware of qChain. He/she must only know that some extra security properties were offered to his/her service and feel safe about it.

The integration also desires be light and not overload the system with heavy usage of the platform weighing the system's performance and reliability, being highly maintainable, customizable, and expandable.

Lastly, license free libraries are used with huge online support by developers and communities, helping future maintenance and error solving.

Overall, the objectives line up to create a reliable security mechanism for document management as a citizen-centered service.

1.3. Proposed Solution

The Blockchain as an intermediary between the platform and the curator's database, in which the digital documents were upheld. Therefore, every operation in the domain of document management will be registered in an immutable database if the documents involved are deemed to have extra security measures by the curator.

The blockchain technology excels at conferring security measures necessary for the system's wellbeing without compromising its utilization, such as integrity of information with repetitive cycles of checking the information conformity and reliability and immutability of storage, timestamping every operation, forever. Therefore, blockchain-based networks possess integrity persistence mechanisms which confer proof of temperance to the digital assets that are part of the network [4][8]. It is highly unwanted that the documents are altered by external attackers with malicious intentions or that data conflicts happen within the database compromising the system's reliability and corrupting the content which is precious to the user.

We decided to call this solution qChain. qChain holds text tokens representative of specific events that happen within the qDocs server, as form of timestamping and registering every operation performed. This feature is essential for information integrity and system reliability, and it is only

turned on for types of documents in which the respective curator decided that it needed additional security measures.

2. BACKGROUND

Both technologies have been around for a while now, and some are still striving with the constant appearance of new solutions, platforms, and integrations as sometimes they reveal to be a lot more useful than we think, considering that both sectors mainly focused on a privative sector of legal firms.

We studied some of the most famous document automation and blockchain solutions in order to not only be familiarized with the main aspects of a public and successful document automation platform but also to pick the best blockchain implementation to be integrated with qDocs. We already had proposed Multichain as the best candidate, but a few more had to be studied to better understand the state-of-the-art.

We considered how easy it is to learn these technologies, to integrate them with existing platforms, how effective and vast the solutions are, and the integrity's transparency to the everyday user's operations. It is equally important how the amount of documentation and community support is present, and the legal license for each technology for maintenance and system sustainability purposes. Therefore, it is expectable that in each technology I gave my opinion in these matters.

2.1. qDocs as a Citizen-Centric Document Automation Platform

qDocs is a solution for public citizen-centric document automation developed by the MDSS company. It has 3 components: a curator server with a database responsible for storing the documents, as close as possible to the official owner entity a main server which processes the entire platform's operations and a user interface, accessed via browser, in which curators, citizen or users, and administrators, are offered a wide array of operations and permissions to the platform.

The platform resumes to a peer-based service, citizen-centered, but managed by admins and seeded by curators. In these platform, the documents can be something simple such as a citizen card to an academic record to a certification of open financial activity [6].

Designers assigned from the curator's affiliation, define the architecture of the document using templates. These are fillable forms, with multiple input types, which produce a digital asset equivalent to the physical one, in shape, appearance and legitimacy. The input types vary from specific fields for decimals, text, signatures, or drop-down lists with multiple choices, all declared by the curator as important. After the form is properly filled and the document created, the latter is stored onto the curator's database. Simultaneously, the qDocs server saves metadata relevant to document fetching performed by qBox, an intermediary to the document retrieval [9].

When the user wants to access an existing document he owns, or create a new one, both the existing document and respective template are stored within the curator that legally issues it. qBox serves as an intermediary for document fetching from the platform to the document's storage, upon request, using metadata present in qDocs, such as its reference in memory or the document's file name.

The curator's database helps provide security and legal ownership of the documents as they are kept close to the respective legal issuer. Curators are responsible for officially issuing these documents, legally binding them to their owner while being aware of the importance that is to not only store the document securely but take careful measures about its retention [10]. The qDocs architecture, in more detail, can be observed in **Figure 1.**

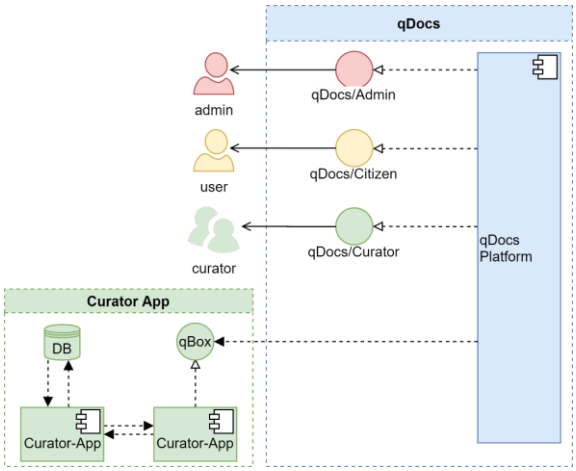

Figure 1 - qDocs Network and System Topology

We quickly refer that this project's intention is to integrate an integrity checking mechanism, such as blockchain, between the platform and the curator, securing the documents' related operations and processes from their creation, management and sharing.

Before we get acquainted with the blockchain's background, basic functionalities, and famous implementations, let's first discuss two present document automation platforms which may be comparable to qDocs, as a helper to further understand how this technology impacts us today.

2.2. Blockchain

Blockchain strives, nowadays, as a solution for a world governed by economic, legal, and political systems, in which multiple entities are bonded together through contracts and various organizations perform numerous transactions of currency and digital assets through the world wide web [11]. Net worth as been one of the closest concepts of empowerment, while operations performed that benefit the digital flux have been the source of that power. Therefore, the blockchain was successfully designed as a mechanism to handle the economy's digitalization when other solutions couldn't, and when governs realized the

criticality of these processes, recording all of them properly was a priority.

This technology was first openly proposed by Satoshi Nakatomo in 2008 through cryptocurrency by the creation of Bitcoin as a way to securely and in a decentralized form, store transactional data of a virtual currency network [12]. Nevertheless, it was in 1991 that the concept first appeared by the names of Haberr and Stornetta [13], who thought of it as a solution to solve timestamping issues, by registering each data asset's time in a way that was fully proof of tampering and non-repudiation.

Involving multiple ledgers in a distributed network, it is considered a distributed ledger technology in which trustless operations are the key, i.e.: every peer participation in a transaction or action in which timestamping is important enough to be permanently registered, doesn't have to trust each other for the transaction to stop being secure. Therefore, the parties involved in a transaction, do not have to be trusted or trust each other, as the blockchain uses mechanisms to abstract that component from its users, in one way, using smart contracts as a set of codified rules to validate transactions securely. These will be discussed further ahead.

Blockchain stores typically metadata for an action in the network associated with an asset of value and places it in blocks, in which each block references the previous block by containing a cryptographic hash of it, creating a growing chain [8]. Hashing is a one-way function producing a unique code for each input, therefore, the probability of collision between hashes of different assets is virtually zero. Each blockchain implementation has its configurations declaring multiple properties of the network such as block size, consensus algorithm, and more, as each platform has either base standard configurations or some more specific and unique to the that implementation.

As explained, the chain of blocks manages to achieve immutability using hashes. Since being a one-way function, they confer some toughness and integrity to information, as it is easy to check if a file was tampered constantly comparing its previously stored hash with a newly calculated one. Every-time a block's content was altered, and a validation confirms it, the entire chain needs to be updated, but when a new block is added before this can be done, the entirety of the chain's block's content will be protected again. Therefore, the blockchain is resilient to malicious intent as long as the time it takes to generate a new block is smaller than the estimated time it takes to tamper with the block and revert the blockchain.

When an asset is issued onto the chain, it is also broadcasted to every node to keep all peers actualized, while these are constantly mining new blocks for storing metadata or for generating noise blocks, increasing the chain, and increasing the time it takes to revert it. Whenever the data is check, a consensus must take place for the information to be validated and still considered secured. This is done through an algorithm used for nodes to argue the information's validity in various ways, using a majority of votes, like the Proof-of-Work (PoW) or Proof-of-Stake (PoS) do, which are the most used consensus mechanisms in the blockchain implementations.

Proof-of-work's concept was invented in 1993 by Cynthia Dwork and Moni Naor to avoid DoS and other malicious intents in email headers by requiring work from a service requester that would compute mathematical equations to prove itself towards the platform [14]. Later, in 1999, Markus Jakobsson and Ari Juels formalized the concept which would be used in Bitcoin [15]. It consists of one node that attempts to validate itself to the other nodes through computational efforts such as calculation problems or puzzles that can be easily verifiable.

PoW increases the blockchain's resiliency to attacks because an attacker must control most of the network's hashing power to actually have an impact on the network and that is why blockchain integrated systems benefit greatly from the distributivity of the network. However, as explained, this method is very demanding for miners and very power consuming therefore multiple nodes tend to group together to join computational power and increase their chances for winning the race, forming what is called, mining pool which may control most of the network, which disrupts the whole concept of decentralization.

Therefore, Proof-of-Stake or PoS was created by Peercoin in 2012 as a solution to all the drawbacks present with PoW, with a very similar concept but different methods. Whereas in PoW nodes would need to prove their worth in the form of computational work, with PoS nodes have in their pose a stake of the network and use that to gamble between them using tokens in the hopes of being selected as righteous miners for the new block.

Nevertheless, both consensuses eventually decentralize the network, as with PoS, the richest node increases greatly its chances when gambling for the block, whereas using PoW, a more expensive system or server, with more computational power, also reduces greatly the time of answer for the challenge. Therefore, these two consensus algorithms, although still often used today, do have major issues compared with more recent proof methods used today [16].

2.2.1. Multichain

MultiChain is an open-source platform for easy creation and deployment of private blockchain applications supporting multiple operating systems. It has Bitcoin as its base but with the purpose to reduce the issues about participants' permissions over the network. MultiChain tries to mitigate how decentralization problems, such as the ones presented a few paragraphs above, affect the entire network by letting peers overpower it. The particular aspect that stands out with this technology is easy configurable

fine-grained control of permissions for the participants with easy maintenance and monitorization of such measures [17].

MultiChain is very easy to use and develop with. Uses a simple API and command-line interface and communicates with the backend server using JSON-RPC commands via HTTP from an executable client. The technology possesses a vast variety of commands like permission granting, mining configuration, requests for network, asset and block information, consensus re-run, and more. All of these are properly documented in the MultiChain's web page.

This blockchain implementation doesn't have a native cryptocurrency and doesn't force the user to. Supports multi-currency and creation of virtual fictitious tokens for private simulations or enterprise encapsulation of digital currency. The most important of all, the use of data streams to store data with greater sizes without weighting too much on the blockchain's performance and not strictly focusing on currency transactions. Moreover, it is highly configurable for easy personalization of the network on permissions, block size and more, being adaptive of the environment it is deployed to.

The most salient key feature of MultiChain is data streams. Data streams behave like private channels in which the blockchain writes information to and can be configured along with the peer's permissions for a finer accessibility. In a way, this feature partitions the blockchain into segments that each can be associated with an ID, but overall, the information will still be stored in the same chain, as this serves more as a tag for better organization of the information and faster retrieval. Nodes of the network may subscribe to streams to participate in operations related to that stream such as block mining, read, or write to that stream, as referred, taking advantage of the permission features of this technology.

Multichain uses a distributed consensus similar to Practical Byzantine Fault Tolerance. However, in MultiChain's approach, there is only one validator per block, working in a round-robin strategy. These validators are picked according to the initial configurations. If one fails in a round-robin type scheme, the system performs a re-round picking a new validator that may even step in from outside of the initial round-robin rotation and become part of the new round set. If not, then there is a fork in the system resolved by the validator picked in the next rotation.

3. QCHAIN ARCHITECTURE

We came across a few challenges when implementing the project's infrastructure.

The first challenge was choosing where the blockchain network should reside inside of the existing structure. Considering that qChain serves as an intermediary to document integrity and conformity approval, it does not make sense to implement it between the curators and the qBox. Since it is managed by the same machine or set of machines running the qDocs server, then the qChain should be deployed and operating between qBox and the main server, registering and timestamping a new entry before the document is sent to the curator for storage and checking its integrity before the document gets to the user that requested it.

The second challenge was how the MultiChain would be deployed and how would the server communicate with it. We chose to deploy the blockchain encapsulated in two Docker environments, one for the documents with multiple stages of creation and another for the final documents. This setup was preferred because MultiChain does not implement smart contracts, and although it has a similar technology, we thought would be more secure if we could register when, per example, a document must be sign by multiple peers, asynchronously, and every time it does, we wish to register the event. This will be discussed in more detail in the next chapter about the backend operations.

3.1. qDocs Requirements

qDocs had some initial requirements that were considered when choosing the most appropriate blockchain technology and how to approach its deployment.

Most importantly, the implementation should be sustainable and easily maintainable with ongoing updates and developer and community support. Fortunately, MultiChain is heavily documented and has already a strong community filled with integration related common issues.

Secondly, we must assure the implementation is transparent and, in any way, reduces the user satisfaction and experience, since this is a citizen-centered platform that should be ubiquitous, fluid, and efficient. Therefore, the normal use of the platform must not be interrupted, and the actual user does not have to know much about the backend functionalities of the integration. The user will surely be notified when the document in question has extra security measures but will not be worried about anything related to whether qChain is running or what is it doing in detail.

Lastly, this solution also considers the learning rate for the technology's usage. In some case studies researched, MultiChain excelled at these characteristics as one of the easiest blockchain technologies to understand, learn, and develop compared with technologies like Ethereum and Fabric.

3.2. System Infrastructure

The overall system's architecture is observed in **Figure 2** below**,** representing two blockchains, within docker containers, one provisional for unfinished documents with multiple stages of creation and a final blockchain for documents that are instantaniously stored after approval. This choice was based on the lack of smart contract technology from MultiChain.

Although the platform possesses a similar, but simpler, technology, we decided that having a blockchain for

registering intermmidieary stages of the document's full generation, was a priority, since we can't control its flux with the finesse smart contracts do. Nevertheless, the platform already has flux mechanisms that help conferring the system some security and reliability. Therefore, since the documents are signed asynchronously in the case of existance of multiple participants, these sign the document sequentially and resulting in multiple versions of the document that all need to be registered, not only because of the conformity with previous versiosn but because a new signature, modification or access is worth the security measures.

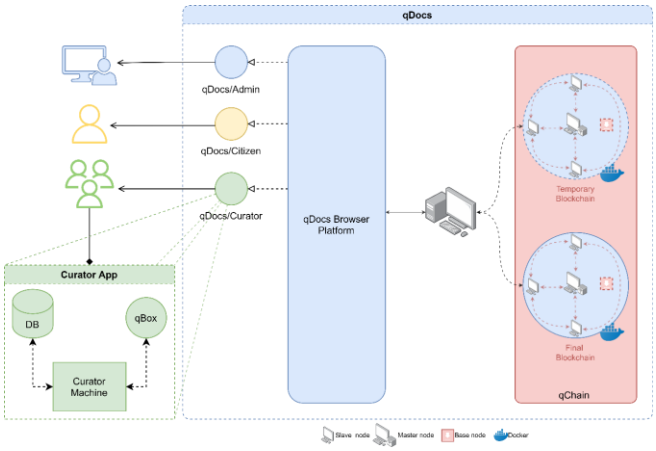

Figure 2 - qDocs/qChain Infrastructure Architecture

qDocs's main functionalities are preserved, the platform will be only responsible for forwarding to the qChain all requests that rely in the document management domain, using the wrapper's methods.

The network deployment and distribution had to heavily consider the communication flow between each component to correctly position each module in its righteous place, therefore, it was the right decision to let the blockchain-based system, or qChain, reside as an optional feature to be used, when requested by the curator, as a security extension of qDocs.

4. QCHAIN IMPLEMENTATION

In this chapter, we discuss the in-depth configurations and specifications for the blockchain network as well as the backend communication with the qDocs server. We also refer the modifications made with the existing code of the platform, what behaviors were change and what was observable in the old structure that might have been optimized in the new one.

4.1 Development Environment

The solution was encapsulated within docker images, in which there are two containers each with 5 nodes: a master node which was the first to be set up and to which the others connect to, three slave nodes responsible for block mining and management and a base node that did the initial setup, downloading and installing the libraries. In total, four operating nodes, including the master, manage the network, as the latter also performs mining operations similarly to its slaves.

The development environment is implemented in a simple way due to resource accountability and simulation, since the qDocs server is not openly in production. The development and further testing of the platform's implementations were conducted in a home computer after proper tool and libraries installation.

Visual Studio was used as a text editor to adjust code to better fulfil the qDocs requirements and the communication with the platform, therefore, we did not perform drastic changes in the code, at most, we deployed method calls for the functions present in the wrapper's library. We only touched a few files, adding fields to further complement and persist the database implementation. These changes will be further detailed in the next subchapter.

Every process of qDocs infrastructure was initiated and tailed through MobaXterm, a toolbox which lets us open multiple command lines in parallel and monitor their status while the data was persisted in a home POSTGRESQL database from pgAdmin's interface. As referred, since qDocs is not actually in full production, the development environment is simulation-based.

4.2. Modules

The overall qDocs solution has multiple components, that successfully communicate between them to offer the most efficient and reliable system they can. Therefore, it is essential that each module has the capability to commit to the system's purpose and interoperate without any internal individual mishaps or network connectivity issues.

Amid the end of the integration, we added two more modules which rely in the middle of the infrastructure: two docker containers each of 5 nodes, containing a provisional blockchain and a final blockchain. Each docker container, as already explained, has only four workable nodes that communicate with the rest of the network, the fifth node only serves to download and install the MultiChain's libraries.

Every module communicates using TCP/HTTP, even the MultiChain does so with the wrapper that performs the JSON RPC calls. Considering the development environment is close to a simulation, it facilitated the encapsulation of each module and further, easy access to each logging and debugging that had to be done. The communication between the qDocs server and the nodes is done through a wrapper is a community library developed by the GitHub user JonathanCrossland as an open-source implementation.

When a there are no specifications for the use of qChain's validation, qDocs will only use qBox for document retrieval. It is important to clarify that qBox will always be used to assist document fetching directly from the curator's

database and that qChain will only store tokens representing important events of each document, which is also stored in the database alongside the document's specifications. Consequentially, **Figure 3** is presented as a full solution considering this condition and further increment the complexity of qDocs/qChain's integration.

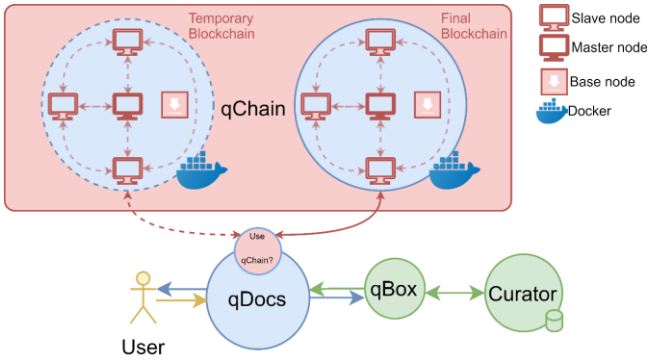

Figure 3 – Final Flow Diagram of Operations for qDocs/qChain Integration

4.3. System Functionality and Specifications

This subchapter details the system's operations on board with MultiChain's architecture along with images, charts and examples of each process and stage of document creation and further access.

Firstly, the document will only ever flow to the qChain if the curator, when designing the respective template for its creation, deems that this should have additional security measures, in this case, secured in the qChain solution.

This is indicated to the curator at first glance when choosing the template's name. The curator will be presented a checkbox along with the words qChain Protection. This is all then persisted in the database as a new column named *MultiChain* in the DocumentTemplate's table in the database. The new interface was only slightly modified to include a simple checkbox under the name qChain Protection which is not mandatory to be filled and does not harmfully impact the curator's experience with the platform.

The platform is responsible for only letting the document be created if all fields are filled properly. If the curator deems that document should be stored on qChain via the indication on the template the process, on this point on time, will start being parallelly followed by qChain. Furthermore, if the document needs more than one stage of creation, such as awaiting confirmation or signature from one or more participants, then it will not be stored in a final blockchain, but instead, in a provisional instance where documents with intermediate processes will be secured. Not needing one, the document's metadata is stored in the final blockchain, which, as explained, is meant to store documents that do not need further operations.

Secondly, we implemented one of the MultiChain's features already referred in the Background chapter called Streams. These reside in the curator, having each curator a stream identified by their full name and referenced by their abbreviature. Streams serve as channels in which documents will be attached too, like tags, and facilitate the retrieval process for documents.

Furthermore, it is here that another feature, smart filters, is deployed, as gates at the entrance of the blockchain, validating the entry's fields and specifications. For now, no smart filters will be developed, because these should be adapted to the curators' needs and therefore, similarly to them being responsible for designing the template, it's also the curator's responsibility to negotiate with the qDocs development team, the desired verifications for documents using that template.

When creating a stream channel, the aggregated process will produce a token identifying the stream, which is then used to connect that stream and retrieve its items and their metadata. This token was also added to the Organization class as a string and further added to the respective table in the database as a new column. Every time a new curator representing an authority or organization is added to the platform, a new stream is also created in the qChain's MultiChain with the already discussed details and identifications.

Lastly, we altered the Document and Persistency class. The Persistency class holds all methods for operations of the platform that will eventually persist information in the database. In this class we implemented the calls for creating streams amid the creation of new organizations, new entries after new documents are filled and updates prior to every document being created, along with integrity checking for each retrieval.

Figure 4 represents the logical process of creating a new document in qDocs, in which this document does not need more layers of creation and its hash is immediately stored in qChain and secured. The operation waits for approval from the qChain's task of creation and produces a result in JSON format with a token. This token is the registration of the event and is added to the document's class and persisted in the database.

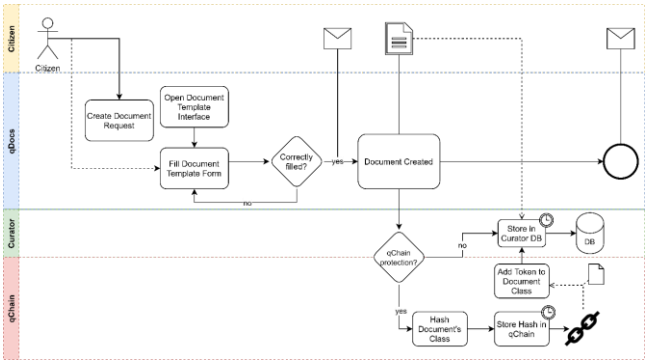

Figure 4 – Business Process of Creating a Document in qDocs

It is also in this class that the document's integrity verification is performed, by fetching the document, also through a wrapper call, and parsing the JSON from the block, retrieving the hash stored in that asset. After the hash is obtained, we hash the document's class we obtained from qBox, before updating the last accessed date, so it does not produce a different output. We then compare both hashes. Assuming they match, the document can be considered secured and the system reliable.

The document is considered open until its closure is requested and the document considered finalized, therefore, it is stored in the temporary MultiChain instance. The last accessible date of the document is updated after this verification and persisted in the database.

In **Figure 5** below, we can see represented the logical operation of a simple document retrieval and integrity checking described in the previous paragraph. This operation uses the token representing the stream which is stored in the organisation's class and the token referencing the document, which was also stored in the creation process, both essential for qChain's asset retrieval.

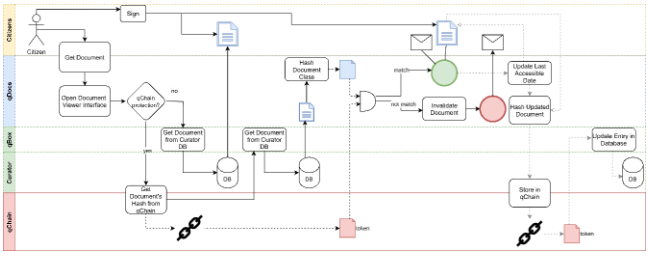

Figure 5 - Integrity check from the qDocs server

Lastly, the platform displays a message confirming the successfulness of the operation, letting the user now that he does not have to worry about the solution's resilience and the document's safety. Next time the user attempts to access this document, the whole operation of the **Figure 5** above will take process to not only help argue the consistency and reliability of the system but to serve its main purpose of securing the document's integrity and operations, from creation to storage to access.

As explained, MultiChain's implementation is a little different due to lack of complexity offered by their smart contract solution, or as they call it, smart filters. Still, we want to assure that a similar mechanism can be present and offer the same level of security, therefore we developed the solution of separating closed documents, in which the only constant update is the last accessible date, from documents that need multiple layers of alterations before officially being issued, such as multiple signatures from more than one participant, performed asynchronously.

Furthermore, Smart contracts would only be used, in the qDocs domain, to hold assets, in this case, documents, to be signed properly, waiting for the signatures of how many people had to sign them or to wait until a ransom was paid for the agreement. Since we can't implement this feature, we register every stage of the document.

Further, we have the example of creating a document that is meant to be signed by n participants, which can be observed in **Figure 6**

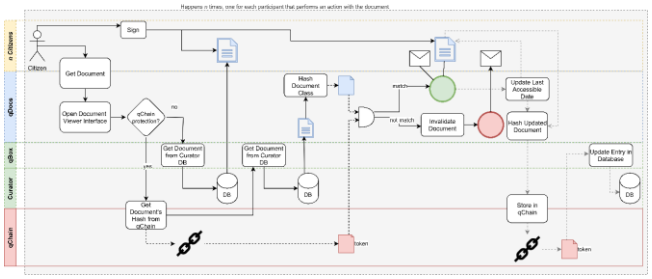

Figure 6 - Creation of a document with signing from multiple participants

This operation will happen as many times as participants need to sign the document in which the curator himself may be one of these participants, to issue its officiality as an authority. Every time a participant signs the document, the platform checks if everyone has already signed. If not, a new updated version of the document is created, stored on the curator's database and its metadata stored again on the provisional blockchain and so on. Finally, after qDocs confirms all signatures and that the document is completed and ready to be stored, this will be secured in the final blockchain as a closed document.

Therefore, creating a document that the curator overrules the need for a signature or that has not yet been closed, will have its metadata stored on the provisional blockchain. The qDocs platform will be responsible for knowing when all the signature fields are filled correctly for the final document to be stored in the final blockchain. Because in this approach we do not control the upholding of this asset the same way a smart contract would, every time the document is modified, a new asset will be written onto the MultiChain's provisional chain. Therefore, every time a participant signs the document, it will be stored as provisional, until the platform confirms that all three signature fields are indeed signed.

This solution confers increased security to documents with multiple stages of creation because when the document is fetched for signing, its integrity will be checked and assured by the provisional blockchain. This happens as many times as the document needs to be modified, assuring security not only on the final stage of creation but also in every process throughout the full generation of the document. The final stage will produce an entry in qChain of different name, instead of aggregating the LastAccessibleDate at the end, it will add CloseDate and store it in the final chain.

5. QCHAIN EVALUATION

There are two main use cases to properly test the system's functionalities and operations, which simulate as close as possible to real ones. No extensive testing is made or unit

testing due to time shortage, but JonathanCrossland, the creator of the C# MultiChain's wrapper, developed some unit tests in his repository regarding the use of the wrapper and proves its functionality's success. The two use cases developed are the ones described as follows:

Case Study (A): Consider a simple document, an opinion poll for the semester satisfaction with the curricular unit's measures. This document only needs to be filled up and issued to the college administration services. This document will surely be stored straight to the final blockchain, as it will be deemed as closed. In this case, the curator is the administration services, and the participant is the student.

Case Study (B): Also consider a more complex example, involving multiple participants to sign one single document minute of a thesis defense. Let us say that after the student has had his dissertation's final discussion, the jury creates a document of the presentation's minute using qDocs, and each one of them needs to sign that same document asynchronously and remotely from each one's home or office. Only after multiple stages, as many as participants signing the minute, is the document closed and stored in the final blockchain. The college administration also represents the role of the curator and every member of the jury and professor present in the thesis defense and the student serve the role of participants

After these two use cases are repeated with various documents fitting to the use cases' criteria to properly analyses the system's operational consistency, we collect metrics of the blockchain's usage.

5.1. Test Environment

The test environment is particularly simple and mostly hardcoded. Mainly we want to test not only the successfulness of creating documents and storing them, but the connectivity and uptime of the qChain as a service. Therefore, we use qDocs platform to create templates that satisfy the purpose of the use cases described above and mark them as deemed for "qChain Protection".

We proceed to create such use case documents using the aforementioned templates and access them multiple times to check the output for the validation of the document performed by qChain. For the second use case we must take extra steps, adding participants and having them sign the document asynchronously to produce various versions of the document, as many as participants signing it, and further produce a final version with the closure of the document.

5.2. Test Results

Unfortunately, the tests couldn't be fully conducted due to connectivity issues with the docker and misconfiguration of the network which led to a heavy load of requests with timed out exceptions. Some features could be tested directly in the docker terminal which proved the system to work, but the communication between the qDocs server instance and the docker program were being cut off. The docker software presents metrics as graphics for memory and CPU usage for each container, which would have been useful to observe when used extensively with the tests.

6. CONCLUSION

We started this project with the objective of integrating citizen critical processes in the document sector, automated digitally, with a secure way of registering events related to these operations. This solution was designed for public, everyday human use, defocusing on enterprise levels of infrastructure and more on the target population that also benefits from these features but does not have them yet.

We stumbled across many challenges such as learning this new technology, despite of it being easy to use and connectivity problems with the docker containers and the MultiChain's inner components. Nevertheless, we agree that MultiChain was the most appropriate platform for the ambit of this project, not focusing only on financial statuses and transactional data, and being optimized with easy deployment and configuration for storage and validation.

Therefore, integrating a blockchain-based network with operations related to documents is good idea if the purpose is to secure them infinitely with no compromise. Mostly this technology benefits the application it integrates with using distributivity properties and non-dependance on trust over the network peers. Considering we are dealing with operations performed in a network-based environment, and of public use, this is ideal for this market.

We also realize that this project could have traveled further in development and be more complete, but unfortunately, due to time shortage and programming challenges, this could not be done as we wanted, still, not only was a new useful implementation developed, but we also researched deep into the world of this integration, which, despite rising, it is still poorly used in the public sector. The tests should be enough to realize the extensibility of the solution, as it does not depend on the document, only on the demand from the curator, since the templates are all in the same format.

ACKNOWLEDGMENTS

I want to express my gratitude to both my thesis supervisors Professor Alberto Manuel Rodrigues da Silva and Dr. João Paulo Pedro Mendes de Sousa Saraiva for motivating me onto finishing this work and giving my best efforts, for always believing in my intentions and that I would, at least, try my best to overcome my challenges.

I also want to thank my parents João and Zara for supporting me through it all, as well as my brother Diogo for all the encouragement and relief they provided, helping me with exhaustive processes in my daily life outside of work and college, and making it possible to not only better manage my time but to keep my mental health.

Most importantly I want to give my deepest gratitude to my wife, Marta, for all that she has done for me, for every word

of encouragement and every bit of love that would give me hope to keep pushing through. Also, would like to thank her for believing in me, for making me feel better about my accomplishments and for always standing by my side and help me overcome every situation alongside me. Without her, I could have not been any successful.

REFERENCES

[1] J. A. Menezes, A. R. Da Silva, and J. De Sousa Saraiva, "Citizen-centric and multi-curator document automation platform: The curator perspective," in *Proceedings of the 28th International Conference on Information Systems Development: Information Systems Beyond 2020, ISD 2019*, 2019, pp. 1–12.

[2] R. Lankester, "Implementing Document Automation: Benefits and Considerations for the Knowledge Professional," *Leg. Inf. Manag.*, vol. 18, no. 2, pp. 93–97, 2018.

[3] V. Gatteschi, F. Lamberti, C. Demartini, C. Pranteda, and V. Santamaría, "Blockchain and smart contracts for insurance: Is the technology mature enough?," *Futur. Internet*, pp. 1–16, 2018, doi: 10.3390/fi10020020.

[4] J. Yli-Huumo, D. Ko, S. Choi, S. Park, and K. Smolander, "Where is current research on Blockchain technology? - A systematic review," *PLoS One*, vol. 11, no. 10, pp. 1–27, 2016, doi: 10.1371/journal.pone.0163477.

[5] MDSS, "qDocs , Citizen centric document technology," *White Paper*, 2018. .

[6] J. António Trocado de Saldanha Sousa e Menezes, "qDocs : Citizen-Centered and Multi-Curator Document Automation Platform : The Curator Perspective," Universidade de Lisboa - Instituto Superior Técnico, 2019.

[7] S. Priyanka and A. Nagaratnam, "Blockchain Evolution - A Survey Paper," *Int. J. Sci. Res. Sci. Eng. Technol.*, vol. 4, no. 8, pp. 1–4, 2018, [Online]. Available: www.ijsrset.com.

[8] M. J. W. Rennock, A. Cohn, and J. R. Butcher, "Blockchain Technology Regulatory and Investigations," *J. Litig.*, pp. 1–11, 2018, [Online]. Available: https://www.steptoe.com/images/content/1/7/v3/171269/LIT-FebMar18-Feature-Blockchain.pdf.

[9] D. José Matias Caramujo, "A Citizen-Centric and Multi-Curator Document Automation Platform : The qBox and Further Interoperability Aspects," Universidade de Lisboa - Instituto Superior Técnico, 2019.

[10] LexisNexis, "Document Retention & Destruction Policies for Digital Data: What You Don't Know Can Hurt You," *Hum. Rights*, pp. 1–6, 2004.

[11] T. Aste, P. Tasca, and T. Di Matteo, "Blockchain Technologies: The Foreseeable Impact on Society and Industry," *Computer (Long. Beach. Calif).*, vol. 50, no. 9, pp. 18–28, 2017, doi: 10.1109/MC.2017.3571064.

[12] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," pp. 1–9, 2008, doi: 10.1162/ARTL_a_00247.

[13] D. Bayer, S. Haber, and W. S. Stornetta, "Improving the Efficiency and Reliability of Digital Time-Stamping BT - Sequences II," 1993, pp. 329–334.

[14] C. Dwork and M. Naor, "Pricing via Processing or Combatting Junk Mail," in *Advances in Cryptology --- CRYPTO' 92*, 1993, pp. 139–147.

[15] M. Jakobsson and A. Juels, "Proofs of Work and Bread Pudding Protocols(Extended Abstract)," *Secur. Inf. Networks. Springer, Boston, MA, 1999*, pp. 258–272, 1999, doi: 10.1007/978-0-387-35568-9_18.

[16] L. M. Bach, B. Mihaljevic, and M. Zagar, "Comparative analysis of blockchain consensus algorithms," in *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2018, pp. 1545–1550, doi: 10.23919/MIPRO.2018.8400278.

[17] G. Greenspan, "MultiChain Private Blockchain - White Paper," 2015. http://www.multichain.com/download/MultiChain-White-Paper.pdf (accessed Dec. 04, 2020).