ISCTE IUL
Instituto Universitário de Lisboa

Departamento de Ciências e Tecnologias da Informação

The Coexistence between Blockchain and GDPR

Nuno Miguel Simões Teixeira

Dissertation submitted as partial fulfillment of the requirements for the degree of

Master in Information Systems Management

Supervisor:
Dr. Carlos Eduardo Dias Coutinho, Assistant Professor
ISCTE-IUL

October, 2019

Acknowledgements

I would first like to thank my thesis' supervisor Professor Carlos for his guidance through each stage of this dissertation and helping me conclude it. He was always available when I ran into issues and was always dedicated in searching for solutions.

To my parents, for always being there for me as well as guiding and accompanying me, always making sure I progress as a human and as an individual.

To my brother, Daniel, for always backing me up and lending a hand when I needed and being a great friend.

To my girlfriend, Bárbara for being supportive, for the encouragement and all the motivation you have given me to finish this dissertation.

To all my friends who believed in me.

Thank you.

Abstract

The constant evolution of technology sometimes cannot avoid conflict with the parallel evolution of surrounding regulations and legislation. This dissertation highlights the Blockchain architectural design and its inherent and apparent incompatibility with the standing European directives concerning General Data Protection Regulation (GDPR) thanks to one of its most prominent features - immutability. As Blockchain-based solutions emerge and their adoption increases, the concerns about current regulation regarding storage of personal data and the conciliation with the Blockchain's model arises. As a consequence, this research aims to find out a practical way of making Blockchains compatible with GDPR and providing a solution, with the elaboration of a Proof of Concept, along with interviews to experts of Blockchain and GDPR's fields with the purpose of obtaining results and drawing conclusions.

Keywords: Blockchain, GDPR, Proof-of-Concept

Resumo

A constante evolução que categoriza a tecnologia não pode, por vezes, evitar conflitos com a evolução paralela de regulamentos e de legislações envolventes. Esta dissertação destaca a discrepância entre a arquitetura inerente dos sistemas de Blockchain e a sua incompatibilidade aparente e inerente às diretrizes europeias assentes sobre o Regulamento Geral de Proteção de Dados, graças a uma das suas características mais importantes – imutabilidade. À medida que as soluções baseadas em Blockchain surgem e a sua adopção aumenta, surgem preocupações sobre a regulamentação atual em relação ao armazenamento de dados pessoais e a conciliação com o modelo da Blockchain. Consequentemente, esta pesquisa tem como objectivo descobrir uma maneira prática de tornar a tecnologia Blockchain compatível com o Regulamento Geral de Proteção de Dados e fornecer uma solução através da elaboração de uma Prova de Conceito, além de entrevistas com especialistas das áreas de Blockchain e Regulamento Geral de Proteção de Dados com o objetivo de obter resultados e tirar conclusões.

Palavras-chave: Blockchain, GDPR, Proof-of-Concept

Index

Figure Index

List of Abbreviations and Acronyms

ABI - Application Binary Interface

API – Application Programming Interface

ASIC – Application-Specific Integrated Circuits

CPU - Central Processing Unit

DAO - Decentralized Autonomous Organizations

DPD – Data Protection Directive

DTO - Data Transfer Object

EU – European Union

GDPR – General Data Protection Regulation

GPU – Graphics Processing Unit

HTML – Hyper Text Markup Language

IDE – Integrated Development Environment

IOT – Internet of Things

JPA – Java Persistence API

JSF - Java Server Faces

JSP - Java Server Pages

MVC - Model View Controller

P2P – Peer-to-Peer

POS – Proof of Stake

POW – Proof of Work

REST - Representational State Transfer

SOAP - Simple Object Access Protocol (SOAP)

URI - Uniform Resource Identifier

1 Introduction

Living in a fully technological era, it’s often hard to keep track of the progress that is made everyday. Technological advances are often associated with needs, requiring investment, leading to new discoveries.

In 2009, a pseudonymous person or group of people named “Satoshi Nakamoto” launched Bitcoin – a digital currency that served as a solution to address the complexities and inefficiencies of current transaction systems. This was motivated by inefficiencies such as time taken for third-parties’ validations, the cost for onboarding of merchants, the time between transactions and settlements and the simple fact that half of the people in the world does not have access to a bank account, needing other solutions to being able to make transactions (Gupta, 2017).

While the system works well enough for the majority of transactions, Satoshi still pointed out flaws such as complete non-reversible transactions and the cost associated with mediation, which increases transaction costs (Nakamoto, 2008). The main objective with Blockchain was creating a financial system that supported disintermediation, where it was possible to conduct transactions with no third-parties involved.

In May 25, 2018, the European General Data Protection Regulation (GDPR) was applied to every individual within the European Union and European Economic Area. The purpose of this regulation is to prevent the increasing, constant concern of cybersecurity and forcing a standard response to these threats and future resilience, as well as protecting the rights and privacy of European Union citizens. The export of personal data outside these areas is also addressed in this regulation.

It just happens that this regulation, as it is stipulated, is not fully compatible with every single technology out in the market. As so, Jan Philip Albrecht said:

“Certain technologies will not be compatible with the GDPR if they don't provide for [the exercising of data subjects' rights] based on their architectural design. This does not mean that blockchain technology in general has to adapt to the GDPR, it just means that it probably cannot be used for the processing of personal data.” (David Meyer, 2018)

While there are Blockchains that can be compatible with GDPR by not storing personal data, those that intent to are not compliant with this new regulation, which could put strain in further development of this technology, abandon of adoption and ultimately result in discardment of this technology as it is simply not viable under this new regulation.

1.1. Motivation and Scope

Blockchain allowed the first reliable, trustless, peer-reviewed, decentralized and pseudonymous digital cash transactions. In 2016, 300 million US dollars were invested in this technology. It is considered by adopters as a disruptive technology that introduced a change of paradigm on everyday activities and business processes (Gatteschi, Lamberti, Demartini, Pranteda, & Santamaría, 2018).

Analyzing the prospects of Blockchain, with Blockchain we can have contracts embedded in code stored in public databases completely transparent to everyone while also being safe from deletion and manipulation. Aside from contracts, Blockchain enabled agreements, processes, tasks and payments would have a digital record with a signature that allowed them to be identified, validated, stored and distributed all around the world. Third-parties like financial institutions or lawyers would not be as necessary as nowadays, as we could have interactions done by digital systems, integrated with Blockchain and that could review transactions done by humans (Lakhani, 2017).

In the future, we could see adaptations of Blockchain's technology that could be used additionally for a variety of cases, more specifically in the fields of smart contracts, public services, Internet of Things (IoT), reputation systems and security services (Zheng, Xie, Dai, Chen, & Wang, 2017).

This said, it's still unknown if blockchain and GDPR can both coexist, since the GDPR gives individuals certain rights that are not verified in the actual Blockchain model. These rights include (Miglicco, 2018):

1. The right of requesting that certain information about the respective individual is deleted;
2. The right to have certain information corrected, if the data about the individual does not correspond to the truth;
3. The right of an individual to know what data a company stores about the respective individual.

1.2. Research Questions and Objectives

One of the objectives of this master thesis is to discover a solution to the actual state of non-compliance between the Blockchain technology and the European General Data Protection Regulation (GDPR), which can not be used to store personal data under this law. Another objective of this dissertation is to publish a research paper to a peer-reviewed journal.

The research question of this dissertation is “How can a Blockchain system that processes personal data be implemented while being compliant with GDPR?”.

Furthermore, the subject of study of this master thesis aims to find out if there is a possibility of employing a Blockchain system that is compliant with GDPR. The resolution of this problem would be of great value for companies that want to deploy a ledger (Blockchain) with personal data about its clients. Furthermore, it is able to draw conclusions for the Blockchain community about the immutability factor of this technology and in which ways the challenge to conciliate both the technology and the regulation could be an obstacle to the storage and sharing of personal data. The uncertainty around this matter could even affect the adoption of the Blockchain technology throughout the years, greatly reducing its potential.

This said, the plan is to develop a Proof of Concept using the Blockchain technology which will allow a user to introduce data - specifically personal data that can be used to trace this same user back if intended. Using this personal data as reference, from there, the objective is to find out what can be done for it to be compliant with GDPR, analysing every potential solution and drawing conclusions about the possibilities. Since the existence of the principle of immutability is applied in Blockchain systems, it is not possible to alter or tamper with records already verified and stored in the public ledger. Therefore, the goal here is to get a solution that could be compliant with GDPR and compliant with the philosophy behind Blockchain, be it the architectural design, the philosophy behind it or even the great potential it has in the long run.

Another objective of this thesis is to do a critical analysis about the solution discovered, evaluating the positive and negative aspects. The solutions could be a workaround on Blockchain forks, which is splitting the Blockchain into two, the new one and the old one, generating a rollback in one of the cases, or a consensus mechanism, where the users could vote whether to remove a record in the ledger or not. In private blockchains it’s easier, since the number of users in the network is diminished, so this could be a potential solution in this case.

This said, the hypothesis of this dissertation is that the implementation of Chameleon hash functions in a Blockchain system can solve the GDPR-Blockchain incompatibility.

1.3.Research Methodology

This dissertation has followed the classic approach to research. It has described an actual real problem, followed by the definition of one or more research questions.

The defining of one or more hypotheses that answer the respective research questions is also contemplated.

To aid on this task, the main objective of this thesis is to develop Proof of Concept for a Blockchain application from scratch with the goal of implementing solutions about these subjects, having personal data stored but in compliance with GDPR, gathering results from this application. The results of this experiment will substantiate or oppose the stated hypotheses. If the results support the stated hypotheses, these will serve as provided arguments to confirm the elaborated hypotheses.

Once the respective hypotheses are validated, our research questions will be, at first, confirmed.

Parallelly to this process, papers will be published to show the results to the scientific community.

2 Theoretical Background

This section intends to introduce the main concepts covered in this research. The topics that will be further described are: Blockchain and GDPR.

2.1 Blockchain

Blockchain has seen a rapid increase in adoption since it has been conceived, with applications based in Blockchain revolutionizing the financial sector. Applications go from the notorious Bitcoin cryptocurrency to proprietary networks used to process financial transactions or insurance claims and platforms that can issue and trade equity shares and corporate bonds. (Reuters, 2018)

Blockchain's relevance and scope goes further than the technology behind Bitcoin's currency. Blockchain serves as a public ledger of all the transactions, be it data or digital currency that has ever occurred in a certain Blockchain. It has a consistent growing rate as miners add new blocks to it, to permanently record the most recent transactions, with the blocks being added to the blockchain in a chronological order. (Swan, 2015)

As mentioned before, miners play an important role in the Blockchain ecosystem: miners provide a solution to the issues associated with the removal of the intermediary agent as well as ambiguity (transaction manipulation) and conflicts (double-spending) occurring in financial transactions in such systems. To achieve consensus in a P2P (Peer-to-Peer) environment, Satoshi came up with a proof-of-work mechanism, where miners allocate processing power from their machines to solve mathematical puzzles that are hard to solve but easy to verify for the rest of the network. This was created as a solution for the lack of identity of every single participant of the network. The objective with this process is to verify and confirm every transaction in this Blockchain, creating a permanent record of such transactions, known as a block. The first miner to solve the mathematical puzzle finds the new block, receiving bitcoins as an incentive (Tappscot & Tappscot, 2016). In other Blockchains, the mining (block) reward is most commonly the respective token of that Blockchain.

When Blockchain was first conceived, it was originally associated with the proof-of-work hash-based mechanism. Nowadays, more than a hundred blockchains exist: some are inspired in the original Bitcoin, others opted to vary and alternate their design for functional and security purposes (Karame & Capkun, 2018).

Blockchain has specific key characteristics that defines it as a breakthrough technology, such as real-time, immutable records, pseudonymity and security, since Blockchain technology has met hacking and tampering attempts with great resistance (Reuters, 2018). This sets Blockchain as a technology well-suited for various sectors conciliated with Internet of Things (IoT), such as supply chain solutions, autonomous vehicle solutions and manufacturing plant asset management (Miller & Laplante, 2018).

2.2 GDPR

Data is an important and valuable asset for the majority of the world's companies. While there are some benefits associated with the technological progress and the data-driven economy we live with, it has reached a concerning point for user-privacy, since there's really a blurred line between storing user-related data and guaranteeing user rights to act on their data as they wish.

On May 25th, the General Data Protection Regulation (GDPR) took effect, a regulation negotiated by the European Union Parliament and the Council of the European Union, comprising the citizens of the European Union (EU) and the European Economic Area (EEA). The objective of this regulation is to strengthen and enforce data protection legislation for every individual in the EU, while attenuating the disregard for consumer privacy that emerged from intrusive computing acts and the increasing change of data present in the big data era (Politou, Michota, Alepis, Pocs, & Patsakis, 2018).

Such regulation was deemed necessary a long time ago. For instance, the EU already has an existing Diretive called Data Protection Directive (DPD). The problem with this initiative is that it is outdated, since it was drafted back in 1995, before the internet was so widely available, only reaching out 1% of the world's population. This directive does not take into account the current situation of the data availability provided by social medias, the existence of cloud computing nor smartphones or tablets (Tankard & Pathways, 2016).

GDPR came to define five fundamental changes on how organizations collect and guarantee their customers personal data, as well as attributing certain rights to customers (Miglicco, 2018).

These are:

1- Obtaining consent. Organizations need their customer's permission to collect personal data;

2- Right to be forgotten. EU citizens have the right to demand that their personal data gets deleted from an organization's database or system.

3- Data transfer. EU citizens also have the right of demanding that their personal data gets transferred to another vendor.

4- Data Protection Officer. When organizations meet certain criteria related to the size of data traffic or meet a certain size, they are required to appoint a data protection officer to manage customers' data and assure compliance.

5- Security breach notifications. In an event of a security breach, organizations have to notify the affected EU customers within 72 hours of the breach discovery and correct the situation immediately.

While GDPR is revolutionizing user rights in this information-driven era, it is certainly a nuisance for organizations. High costs associated with the employment of frameworks, acquiring external services and purchase of expensive software prove that GDPR compliance was not an easy task. Even so, GDPR compliance can be itself a business opportunity. It will force, on a certain level, that companies have a more comprehensive view of all their data, which will facilitate analytic studying and identifying patterns. By managing data in a proper, more formal way, it is actually easier to benchmark success and identify best practices, bringing more clarity to certain processes, enabling organizations to pinpoint trends, the ability to predict future events, getting an upper-hand if properly planned and it can even identify new market opportunities (Garber & Focus, 2018).

3 Literature Review

This chapter intends to point out the research done by the scientific community with the topics being studied: Blockchain and GDPR.

Since this research aims to study the innate state of lack of compliance between Blockchain and GDPR, it’s necessary to search for literature done by other researchers in order to find out what other solutions and proposals for this problem. This chapter also serves as a survey to discover what has been done literature-wise about both of these subjects. Even though Blockchain has been around since 2008, it has changed a lot throughout the years, as progress has been made, new technological designs arised as well as new projects that serve a totally different purpose than the acclaimed Bitcoin. As for GDPR, it really is a recent topic, even though it has been planned and discussed since 2015, it only came into action on May 25 of 2018, which increases the importance of chapters like this one.

The libraries used to access these documents were: ScienceDirect, IEEE and Google Scholar.

3.1 Blockchain

Blockchain Technologies: The Foreseeable Impact on Society and Industry

The author starts by detailing the history of Blockchain and the business opportunities when Blockchain is employed, followed by the main characteristics and features of the decentralized ledger technology, categorizing it as the drivers of the Blockchain revolution, such as decentralized and transparent consensus, security, immutability, automation and metadata (Aste, Tasca, & Centre, 2017).

It goes on to evidentiate the kind of effects Blockchain can have on services, businesses and regulations specifically. The author is able to identify potential for an increase in operational efficiency, through automation and the trustless foundation Blockchain is based on. It also refers to a rebalance in information symmetry by justifying it on real-time, transparent records that are easily auditable and monitored, followed by a mention of the decentralization of corporations and governance regarding the use of smart-contracts and the shift of paradigm from centralized hierarchies to Decentralized Autonomous Organizations (DAO), where decision making is not focused solely on the center, but instead is spread across the networks' nodes, giving way to autonomous, incorruptible business management through the use of smart-contracts.

The author explains briefly why Blockchain is a disruptive technology and attributing it mainly to the P2P operations without intermediation from financial entities, or any other entities for that matter, noting that it withstood widespread attraction, adoption and capitalization in an autonomous way without falling victim to any attack, followed by the technology's limits and efficiency, where the author concludes that Blockchain is not as scalable nor easier to manage and not faster to operate as centralized systems. It also attributes some limitations to the fact that miners usually group their computational power and share the profits among the participants of those groups, concentrating a large percentage of the market share in Bitcoin's Blockchain, mentioning that from 2013 to 2015, the 10 largest groups (known as mining pools) owned 70% to 80% of the total computational power. The trend kept going and, as of May 2017, mining pools produced 45% of the Bitcoin Hashrate. The author identifies other constraints based on the power cost associated to mining, as well as slower operation compared to other centralized financial transaction processors (such as Paypal or Nasdaq), limited governance and sector concentration.

Concluding, this research clearly identifies strong and weak points in Blockchain. For one, the author clearly introduces us to the main features of Blockchain, using it as evidence as to why Blockchain can be disruptive and a game-changer for business processes and even business models, while also mentioning and enumerating certain flaws associated not only with Blockchain's design, but also flaws created by the community, from the possible lust-lacking future of centralized Blockchain, thanks to mining pools, to the high cost of mining, which is a burden for our planet.

Blockchain and Smart Contracts for the Internet of Things

Blockchain and Internet of things are two concepts that are coupled together, and that's what this specific research has achieved to exemplify.

The author starts by explaining the brief history of Blockchain and it's catapulting to mainstream adoption by several industries, attributing this to the fact that this technology enabled the creation of trustless networks through cryptography and the absence of a intermediary. This, allied to smart contracts – self-executing scripts on the Blockchain –

provides great conciliation capability to the Internet of Things (IoT) domain (Christidis & Member, 2016).

Next, there is a detailed explanation on how transactions are done on Blockchain involving two peers and how smart contracts can be conciliated with Blockchain, set up and automated, followed by how both IoT and Blockchain can be used, highlighting existing applications that employ both of them, along with the authors' end conclusions.

Concluding, this paper evidentiates the interesting synergy between both Blockchain and IoT. Blockchain could provide automation functionalities, while IoT could easily make it accessible and reachable to every device, preventing users and customers from extensive and lengthy business processes, or even other simple tasks, like a monetary transaction.

Blockchain's roles in strengthening cybersecurity and protecting privacy

This paper relates Blockchain with the concepts of computer security as well as privacy, two actual terms that we hear frequently nowadays.

The author starts the paper introducing the Blockchain's security features, proposing that hacking attacks are not very effective against Blockchain, since the data is distributed around the many computers of the network that are interlocked and that, for the hacking attempt to be successful, it would have to hack more than 50% of the computers of the network (Kshetri, 2017), also known as a "51% attack", where the blockchain is classified as compromised. The paper goes on to state that, on Bitcoin's Blockchain, a transaction between peers has never been compromised.

The next section compares security and privacy between both cloud technology and Blockchain, where the author concludes that most concerns about cloud adoption include privacy, security and availability issues, which could be minimized by the employment of Blockchain allied to Cloud computing.

The author proceeds by illustrating how Blockchain-based systems can be used in order to guarantee security and privacy in the specific case of healthcare industry, followed by the detailing of Blockchain and IoT coexistence effectiveness, pointing out that IoT security has been a worrying problem, from lack of encryption and the standard accessibility and ease-of-connection, which coupled together can be a disastrous recipe, if a hacking attempt is successful. Next, there is a group sections describing measures on how Blockchain can complement IoT lack of security, just before a descriptive part where the author indicates how Blockchain can help the application of Fair Information Practices, by proposing that Blockchain could be a substitute for internet cookies and also aid on digital marketing and advertisement, by allowing payments through bitcoins generated from ad-revenue, or connecting concerned advertising parties with smart contracts, and even tracking the accounting process.

The author concludes the paper by stating that Blockchain's security features are tamper-proof and not easily manipulated, making it a hard task for hacking attempts to be successful, leading to a promising future.

The paper is very detailed and well structured. However, the author, while presenting innovative use cases for Blockchain's security key points, also forgets to point out important flaws in the Blockchain. Some Blockchains environments allow specific languages for smart contracts development, which itself is an interesting feature as well as a flaw, since some languages allow for loops in the code, exploiting vulnerabilities. While the paper mentions that, for a Blockchain to be hacked, more than 50% of the mining hashrate should be from hackers' computing power, the author does not mention the threat of quantum computing, which have exceeding computing power that could easily gather most of the Blockchain's mining hashrate, rendering Blockchain compromised and useless if applied maliciously.

An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends

This paper decided to focus on the essence of Blockchain, presenting a comprehensive overview, as well as achievements and future trends for the technology.

The author starts by targeting Blockchain's architecture, focusing on the parts that consist a transaction, like blocks, digital signatures and parallelly, the features of this technology, such as decentralization, persistency, anonymity and auditability. It finishes this section with the taxonomy of Blockchain systems, citing that current blockchain systems can be classified into 3 categories: public, private and consortium Blockchains (Zheng et al., 2017).

Following last section, we have the consensus algorithms, the key to reach consensus among untrustworthy nodes in a network, where the author elaborates on the main ones, like Proof of Work (PoW), Proof of Stake (PoS) and some others.

The paper's next section focuses on challenges and recent advances, targeting arising concerns such as scalability, privacy leakage and selfish mining.

Possible future directions comes just before the paper's conclusions, where it states the possibility of Blockchain testing as a standard for certification and quality of new Blockchains that are constantly being developed. Other future directions are putting a stop to the tendency of centralization, as mining pools are racking up a big percentage of the mining hashrate of most Blockchains, followed by big data analytics and finally, Blockchain applications.

The author's conclusions evidentiate Blockchain's potential and pinpoints once again Blockchains features as a recipe for success.

Concluding, this paper was nicely elaborated. The author focuses pretty much on the Blockchain's main flaws which could contribute to the impoverishing of the technology and adoption, tendencies that could not be as easy to prevent or stop as it was once thought, such as stopping centralization, which is against Blockchain's nature.

3.2 GDPR

What the GDPR means to businesses

The author chose to adapt the reality of GDPR and apply it in a business environment with this paper.

The paper starts by stating what is to change when GDPR is called into action, talking about the necessity of notifying data breaches for companies to the respective authorities and users affected by the data breach, followed by stating facts about individual rights, such as the need of consent from the users to process data and the right of the users for the data to be forgotten. Next, the author states that, according to Ovum, 52% of the organizations believe GDPR will result in fines against their companies and that 68% believe in a higher cost of doing business with European organizations, leading to an increase up to 10% of their annual budgets over the next two years (Tankard & Pathways, 2016). These statistics demonstrate the general worrisome perception of organizations on the elevated impact of this regulation

The author focuses right after on the requirements for GDPR compliance in organizations, focusing one part on the pseudonymous data, stating that, according to GDPR, data from users should never be traced back to an individual, hence the need for pseudonymization and pinpointing 5 important steps according to Baker & McKenzie law firm, focusing the last sections of the paper on the necessity of data encryption, minimising data collection, the entitlement of data for a specific user, which cannot be attributed to someone else and finally, industry standards, best practice frameworks and the author’s conclusions, warning that organizations should not be caught off-guard by compliance deadlines.

Concluding, the paper is quite solid. The author keeps a calming tone the whole paper, showing that GDPR compliance is very possible for every organization where the only variable should be the organization’s size and the lack of effort to meet established deadlines by European authorities. The major flaw with this paper is that it was created in 2016, where GDPR’s final publication was not yet available, being subject to tweaks and changes, giving a slightly inaccurate acknowledgment of the final version of GDPR and its reality.

GDPR is here and it’s time to get serious

A paper that is more up-to-date than the previous one, dating back to 2018, where the author follows the same approach of the existing constraint of GDPR for businesses.

The author starts by citing that half of the affected companies will not be fully compliant with GDPR by the end of 2018 and that, in United Stated of America alone, 52% of their national companies currently possess data that could be compromised by GDPR (Miglicco, 2018).

The next sections elaborate on what GDPR consists, the major changes for organisations and what is implied for consumers.

The paper also focuses on unclear applications of GDPR, citing cloud-based systems, where it’s very common for organizations to exchange data with cloud systems. The author states that the law differentiates between organizations that collect and manage data and organizations that process and control data, such as cloud-based system providers.

The author concludes the paper with a future perspective, mentioning that the United States of America will be elaborating similar regulations, as concerns for privacy increase.

Concluding, the paper is simple but approaches the main chapters of the GDPR implications and how organizations should proceed for compliance. The paper does not focus solely on EU countries, but also states the state of context regarding privacy regulations, such as in United States of America and China, where the author states that the later is already pretty much compliant, asking users for consent to process personal data, as well as remediation for data breaches.

Data in the Post-GDPR World

This paper starts by using a metaphor, associating data with oil, as it is increasingly valuable in the modern world. The author associates the value of data harvesting and monetisation with the escalated disregard for privacy, pointing out that organizations can no longer discard customer’s rights since the implementation of GDPR (Datoo & Technology, 2018).

The rest of the paper keeps the focus on the importance of privacy, highlighting the Facebook data scandal, that cost the company $60 billion and motivated regulations like this one, to the fines associated with non-compliance with GDPR. The author focuses the final part on the post-deadline set by May 25, raising concern about the companies that were not fully compliant, to examples where organizations tried to circumvent GDPR with stickers or warnings on their data-gathering websites, simply because they were not ready by the time they met the deadline.

Concluding, the author did not elaborate as much as the previous papers that approached this subject, focusing mostly on organization's philosophy and governance for data management, before and after GDPR came into action. The paper focuses a bit too much on what justified the creation of this regulation and does not aid much organizations or the readers on how the paradigm has changed from focusing solely on data monetisation to shifting some power into consumers.

3.3 Blockchain and GDPR

Blockchains and Data Protection in the European Union

The paper focuses on the Blockchain technology and other forms of distributed ledger technologies and the relation it has with GDPR, introducing the existent incompatibility between them, as that, in a first glance, the standard concept of Blockchain and how it is designed is in a state of non-compliance with GDPR (Finck, 2018).

The next chapter focuses on data on Blockchains and its implications. The author points out incompatible features of Blockchain, such as the data in permissionless Blockchains (like Bitcoin's Blockchain) being transparent and accessible to everyone, which is not lawful according to the new regulation.

The Section III mentions future work to be done and how Blockchain can be more GDPR-compliant from a developer's perspective, like giving data subjects some control of their data.

Section IV focuses in the essence of GDPR and Blockchain original incompatibilities, stating that even though users of Blockchain are pseudonymous, data can still be traced back to find the original identity. There's the possibility of data being totally anonymous, though, but that's not the case in every Blockchain. This section focuses on public key trails and storage of personal data as well. The next section focuses on making Blockchain GDPR-compliant, researching data controllers' obligations, the GDPR territorial scope and enforcing data protection rights on Blockchains, such as the right to amendment, the right to be forgotten and the right to access.

The last section talks about the "reconciling of the protection of fundamental rights and the promotion of innovation", where the author states that laws and regulations always lagged behind technological progress, becoming more evident as the progress speeds up in the current era, and associating this problematic with distributed ledger technologies and the constraints it originates.

Concluding, this is a very well-written paper that approaches pretty much the scope of GDPR and Blockchain's as well, clearly evidentiating the concerns of GDPR's arise, focusing on the most important and essential points and potential solutions.

On Blockchains and the General Data Protection Regulation

The author initiates the paper by introducing Blockchain's technology relevance investment-wise and parallelly introducing GDPR, focusing the next 2 sections elaborating on these subjects, respectively.

The author approaches the state of play, surveying GDPR roles in Blockchain networks, where he states that permissionless Blockchains can be problematic, because all participants are joint-controllers of data in the ledger and that the fact that participants are pseudonymous complicates matters even more, since it's impossible to address anyone directly and raises concerns about the transfer of data to third countries, since there is a need of a validator in the foreign jurisdiction (Ibáñez, Hara, & Simperl, 2018). The state of play section also contemplates hashes and public keys as personal data, the principles of data processing and proposed solutions from the technical and legal side, as well as data erasure for efficiency reasons.

The next section pinpoints some strategies that could be employed by Blockchain adopters that could be compliant with the regulation, differentiating from different scenarios, like storing personal data and non-personal data and Blockchains that can simply store any kind of data.

The last section talks about Blockchain as data protection by design enablers, talking about possible implementations for Blockchain developers to possibilitate and/or facilitate compliance, like shifting data access rights everytime a transaction occurs, from data controllers to data processors, followed by the paper's conclusions, where the author concludes that both Blockchain and GDPR serve to empower individuals and reduce the inequality of rights between data subjects and organizations, raising some legal and technical questions which are in the grey area so far.

Concluding, this paper was interesting and captivating. The author provides many solutions for the Blockchain – GDPR dilemma with various scenarios for the different kinds of distributed ledger technologies. The scope of this paper was pretty centered on these two subjects, which is honestly not a flaw, since the author achieved a somewhat specialized and specific problem-solving literature.

Blockchain and the General Data Protection Regulation: Can distributed ledgers be squared with European data protection law?

This paper, written at the request of the Panel for the Future of Science and Technology, which belongs to the authority of the European Parliament, pinpoints the discrepancies between Blockchain and GDPR and aims to find solutions for a coexisting environment.

The authors start by describing both subjects by detailing their specific characteristics and how they both work. The introduction is concluded by assimilating both in the same context and drawing attention to the existing conflict, developing the issue during the following sections.

The paper also focuses on points like hash functions, mentioning that the fact of hashing data will not transform personal data into anonymous data (Finck, 2019). It also cites off-chain data storages as a possible solution for compliance. The following section cites chameleon hashes and editable blockchains, mentioning that this solution defeats the purpose of using blockchain and that it will still depend on surrounding governance arrangements for this to be a fitting solution.

The paper also refers to permissioned (private) blockchains as a better implementation for a solution that aims for compliance, since the network in these kind of implementations tend to have a smaller number of participants and the data persisted in the blockchain, as well as its nature, can be more easily identified and managed.

Overall, this paper sums up the existing problematic between both subjects and goes as far as to expose some solutions to try to circumvent the lack of inherent compliance between Blockchain and GDPR.

4 Features and Characteristics

In this chapter, we will explore blockchain's available mechanisms and characteristics that exist to make it operable. Some of these mechanisms can also aid in the conciliation of this technology with GDPR. Even though Blockchain has been considered irreconcilable with the actual model of GDPR, some solutions can probably be achieved, although at a cost.

4.1 Forking

Forking is one of the most popular concepts in the Blockchain technology.

A fork in Blockchain occurs when different parties, which have to use the same common rules, are not in concurrence. This lack of consensus originates alternative chains.

Even though the originated blockchains are unique and distinct, the way they are connected to each other depends on the type of fork that occurred. There are 2 main types of forks:

Soft Forks

Soft forks are software or protocol updates that includes backward-compatibility, or compatibility with the older blocks that are already present in the blockchain. It usually requires the miners to upgrade to the new version by downloading the latest software, however, the nodes can refuse to upgrade to the new functionalities and the old rules will be kept (Reyna, Martín, Chen, Soler, & Díaz, 2018).

Hard Forks

Hard forks are a rougher version of soft forks. They contain similarities to soft forks, where there are software or protocol updates where the old versions of the software become obsolete and incompatible, forcing the nodes to upgrade to the latest version. Hard forks occur when fundamental changes occur to the blockchain, such as consensus rules, which can be changes to the mining algorithm, block sizes or consensus protocol.

These forks can place the community in discordance, where some nodes do not agree with the radical change to the rules of the system and simply refuse to update to the latest version of the blockchain, resulting in two different versions of the blockchain.

Some of the most notorious forks did not reach or achieved agreement between the miners. For example, the Bitcoin's SegWit2x hard fork was cancelled due to lack of agreement among the community at the time. (Leising, M, 2017).

Other hard forks have achieved mild success, such as the Ethereum hard fork. This fork was planned out to reverse the effects of the Ethereum DAO hack, where tokens were stolen by hackers (Güçlütürk, 2018). The fork was carried out successfully and it originated two versions of the blockchain.

While soft forks are often classified as light upgrades, hard forks are not. Hard forks can indeed be a workaround to blockchain's immutability that prevents the tampering of the registered entries in the ledger. While this can be seen as a solution to correct unsolicited occurrences, this also disregards one of the technology's most prominent features. The value of immutability is that transactions are recorded permanently and can't be altered or reversed once they are validated by miners. If hard forks were to be carried out frequently to manipulate immutability, then the value of blockchain would be greatly reduced and the need for this technology would be diminished, since we are reducing many of its' use cases. Furthermore, most of the community has been concerned with this controversial subject for some time. Some agree that code should be regarded as "law" and that immutability is mandatory, as it is the foundation for the trustless operations that blockchain can provide, while others agree that decision-making should be more pondered, which can also be a valid point of view, since blockchain is a concept developed by humans based on code and sometimes can be flawed.

Since immutability is one of the main protagonists of the blockchain-GDPR lack of compliance, hard forks can be seen as a key solution. They can be classified as a classical data rollback, hard forks would be relying on going back to a certain block validated in the past and every block after that chosen one would be deleted.

Permissioned blockchains provide the most practical applications for companies since there's a much broader sense of control over the respective use of this technology, allowing them to manage risk, control costs, infrastructures as well as the much more accessibility to comply with the demanding industry regulatory requirements, especially in difficult business areas, such as finances. There is not an easy approach when it comes to circumventing immutability on blockchain, whereas forks can provide a simple solution, while offering drawbacks at the same time.

There are five main reasons to consider forking to avoid immutability:

Scalabilty of data storage capability – Since every single transaction is kept as a record permanently and is never deleted, it raises concerns about the scalability of the blockchain system. While blocks themselves are not large in storage size, many use cases in the chain can create massive amounts of data volume and put a strain in the hardware supporting it. Soft and hard forks could reduce the risk of potential unsustainability presented by blockchain and reduce workload stress in the hardware.

Criminality – While it is not expected to have criminal activity in a permissioned blockchain, the same is not true for permissionless blockchains such as Bitcoin and Ethereum. These systems are prone to malicious actions by their respective users, which are then consequently recorded in the ledgers permanently. The Ethereum DAO hack previously was avoided thanks to a hard fork voted successfully by its community. Forking can effectively solve these problems of illegal nature by providing the option to rollback to another block and undo the prejudice done.

Operational errors – Humans are flawed by nature and it is reflected in our actions. We cannot avoid human error completely even with the aid of technology. This is especially troublesome on a technology that stores transactions chronically with no way to correct it. Regulation and jurisdiction can also change the paradigm and dictate that some specific set of data is mistakenly stored and needs to be deleted. Forking can provide a solution to this problem.

Permanent misconduct – Another aspect that forking can solve is the misconduct of the users in the transactions executed. The fact that permanent records are kept in the ledger presents a large problem thanks to the potentially sensitive nature of the data that can end up stored. There are blocks in the Bitcoin blockchain - the most popular cryptocurrency, that contain pornography and which will not be deleted anytime soon. Other data stored in different blocks contain sensitive Wikileaks data (Matzutt, 2018). The possibility for information leakage and consequential damage is enormous. This concerns arises in permissionless blockchains where there is no overseeing or governance by entities, while permissioned blockchains have an increased management of data and, if used by companies, data of such nature would not be tolerated. Nevertheless, forking would be the perfect solution for a rollback on the damage done.

Regulatory concerns – From a regulatory perspective, forking should be able to aid such problematic. As said before, it could solve the existing dilemma between blockchain and GDPR, even though it may not be an optimal solution. Other industries that employ blockchain and have demanding requirements, such as confidentiality and/or the need to alter data already registered in the blockchain may prove forking to be deemed as a necessary action.

The use of the forking capability of blockchain could possibilitate the circumvention of some of blockchains imposed native limitations, increasing the ease of access to adopt this technology outside of the typical cryptocurrency use case and therefore opening doors for different industries. But forking is not an inoffensive mechanism: it mixes up the hashes of the remaining blocks. Since these remaining blocks have hash pointers that indicate the previous and the following block, removing even a block can label a blockchain as illegitimate and inoperable because the block hash will be altered.

In permissioned blockchains, this is no problem, since there is some level of intermediation, but in a permissionless blockchain such as Bitcoin or Ethereum, this is not feasible with an enormous level of hassle.

4.2 Hashing

Hash functions are an important application of cryptography. They compute a fixed-length bit-string which have a standard set length. This string can be seen as an unique representation of a specific message just like a fingerprint. Hash functions are required and fundamental in modern computing as part of digital signatures schemes and message authentication mechanisms, as well as for storing passwords (as hashes) and key derivations (Konheim, 2010).

Some of the main characteristics of hashing consist on:

- A certain input always generates the same output. Independently of how many times you run a specific set of data through a hashing function, it always must produce the same output.
- Any changes that occur to the original set of data that was defined as input must produce a different output. Even the slightest modification should generate a significantly different change.
- The data set as input can never be calculated from the output, meaning that there should be no way to reverse-engineer the hashing process and/or the output to find out the input.
- The output has a fixed set of characters always, regardless of the size of the input.

When addressing blockchain, we can pinpoint six operations where the use of hash functions are involved in blockchains (Wang, Shen, Li, Shao, & Yang, 2019):

- Consensus algorithm hashing (for example, Proof-of-Work);
- Address generation;
- Block generation;
- Message digest in signatures;
- Pseudorandom number generation;
- Bridge components.

Blockchain uses hashing on each transaction and groups them in blocks. Each block contains hash pointers, which indicate the previous block. This creates a foundation for immutability in blockchain, since any slight change to the transaction will alter the hash, assured by the hashing function. Every single transaction hash will be altered if a transaction is tampered with.

Bitcoin set the standard with its use of the SHA256, which was created by the United States National Security Agency (NSA). But other blockchains use different cryptographic hash functions, which is the case of litecoin and its use of the Scrypt hash function, or Dash blockchain with their X11 hashing algorithm. X11 appeared at the end of 2014 and it was designed for cryptocurrencies and it achieved a security standard which is considerably high, even more so than the SHA256. One of its characteristics is that it cannot be operated by Application Specific Integrated Circuits (ASIC), which are typical computational machines that are created to serve a specific use case, performing a particular computing task. In this case, they have an increased processing capability with the objective of mining. The X11, besides being more secure, is also ASIC-resistant. The job of an ASIC machine is to make as many hashing functions per second as possible. The fact that X11 makes a blockchain that employs this hashing function ASIC-resistant makes it so that using these machines makes it not as profitable or sometimes not even a viable option for mining. However, since it was conceived, some ASICs capable of mining blockchains that employ the X11 hashing function have been developed, which puts miners with dedicated graphics processing unit (GPU) and central processing units (CPU) at a disadvantage and creates centralization, since it enables the existence of mining farms, a place where ASICs are stacked and focused on the goal of mining a blockchain, creating a massive hashing power. Blockchain suffers from this since it creates centralization, which is the complete opposite of the main objective of its creator Satoshi Nakamoto.

Blockchains can rely on other methods that are ASIC-resistant by design – such as Proof-of-Stake (PoS) and Proof of Authority (PoA), which are different types of consensus algorithms, which will be referred to ahead.

Being ASIC-resistant in a blockchain system is a powerful feature to have: it creates independence and decentralization. Currently, the vast portion of network hash rates on the most popular blockchains is generated in mining farms provided by some specialized companies, stacking a large number of ASICs and making them available through cloud services and rental schemes. This is a bad scenario for this technology, particularly because some blockchains rely on this massive hash power originated in these farms to maintain the hash rate needed to function effectively and keep the network secure.

Meanwhile, the fact that these farms are retaining a big part of the hash rate of blockchains and that they keep growing constantly, means that decentralization in blockchain is becoming more and more of an illusion. This is evident in the below.

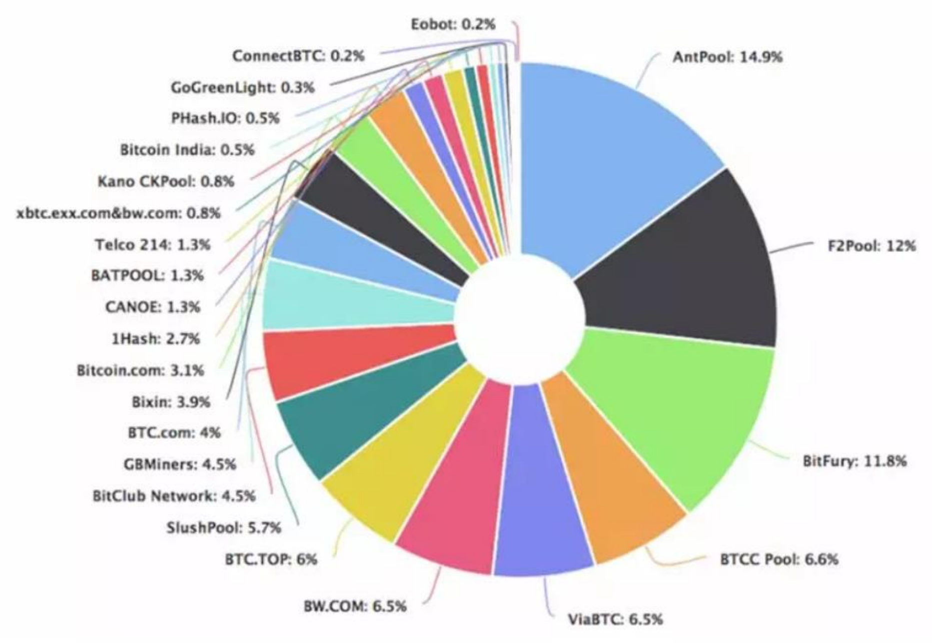


Figure 1 – Mining Pools and Respective Hash Rates. Source: https://bestpoolmining.com

In fact, some mining farms have achieved such a considerable amount of collective hash rate in bitcoin's blockchain that, if they collaborated, they could elaborate a 51% attack, concentrating more than 50% of the total hash power on Bitcoin's blockchain and take control of the whole blockchain, manipulating transactions, forcing forks, possibilitating double-spending by exchanging bitcoin and then deleted the transaction, recovering the currency. They can also prevent remaining miners from finding new blocks, allowing the controllers to monopolize the mining of new blocks and earn all of the rewards. (Ye, Li, Cai, Gu, & Fukuda, 2018)

It is safe to assume that decentralization benefits the correct functioning of blockchain and that new solutions must be developed to prevent the increasing centralization caused by unfair mining strategies.

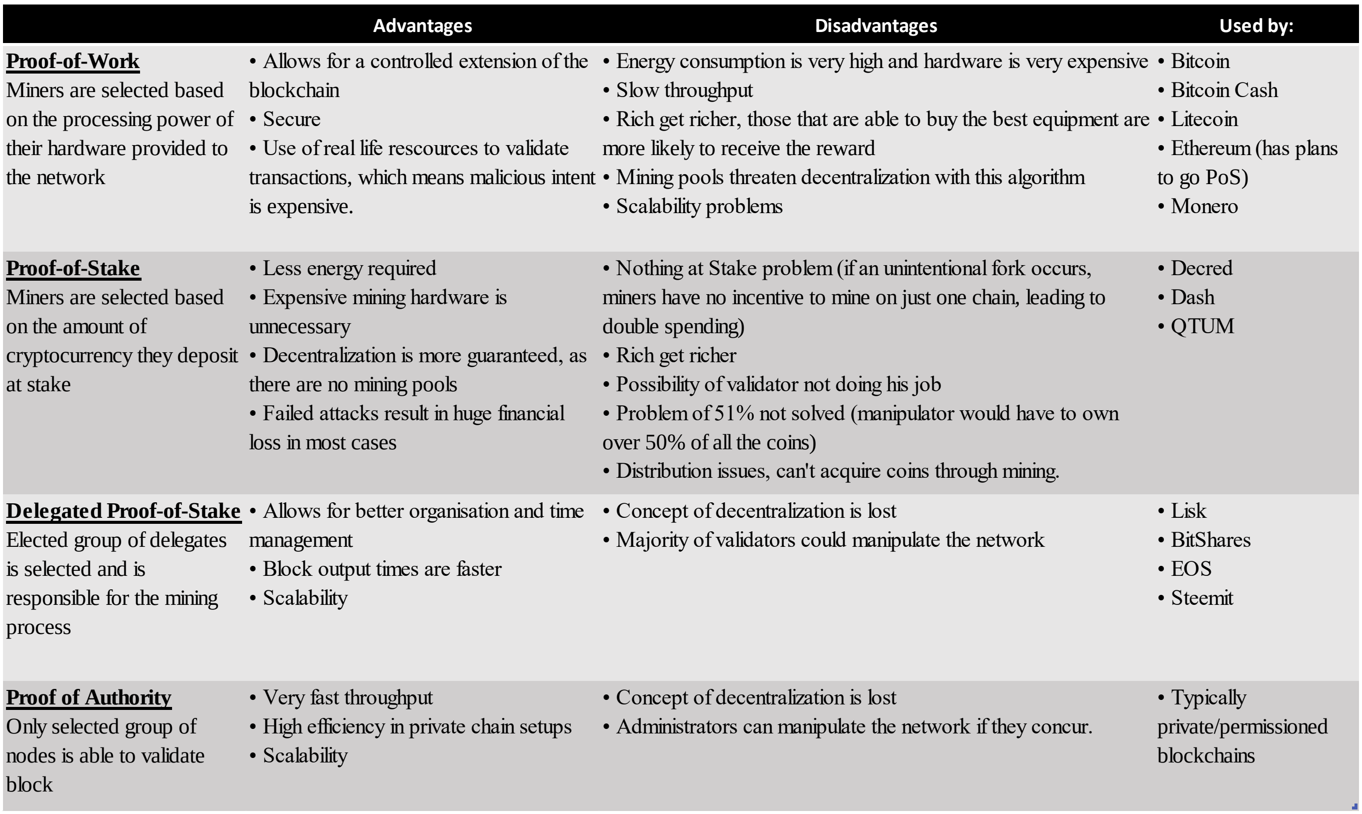
4.3 Consensus Algorithm

Consensus algorithms are a fundamental part of every blockchain system, as they are the mechanism responsible for the miners and participants of the blockchain to attain a trustless system, as well as maintaining the integrity and security. The most popular consensus algorithm is the PoW, which was first implemented in the original Bitcoin.

Since blockchains are disintermediated and we are referring to a distributed system, this mechanism is used for the existing nodes in the system to agree among them on the validity of transactions, assuring that the existing protocol rules are being used and to make sure that all the transactions occur in a trustless way, avoiding the double-spending of a token or the tampering of the blockchain (Bach, Mihaljevic, & Zagar, 2018).

There are many consensus algorithms, but the most popular are described in below:

Table 1 – Main Consensus Algorithms



	Advantages	Disadvantages	Used by:
Proof-of-Work Miners are selected based on the processing power of their hardware provided to the network	• Allows for a controlled extension of the blockchain • Secure • Use of real life rescources to validate transactions, which means malicious intent is expensive.	• Energy consumption is very high and hardware is very expensive • Slow throughput • Rich get richer, those that are able to buy the best equipment are more likely to receive the reward • Mining pools threaten decentralization with this algorithm • Scalability problems	• Bitcoin • Bitcoin Cash • Litecoin • Ethereum (has plans to go PoS) • Monero
Proof-of-Stake Miners are selected based on the amount of cryptocurrency they deposit at stake	• Less energy required • Expensive mining hardware is unnecessary • Decentralization is more guaranteed, as there are no mining pools • Failed attacks result in huge financial loss in most cases	• Nothing at Stake problem (if an unintentional fork occurs, miners have no incentive to mine on just one chain, leading to double spending) • Rich get richer • Possibility of validator not doing his job • Problem of 51% not solved (manipulator would have to own over 50% of all the coins) • Distribution issues, can't acquire coins through mining.	• Decred • Dash • QTUM
Delegated Proof-of-Stake Elected group of delegates is selected and is responsible for the mining process	• Allows for better organisation and time management • Block output times are faster • Scalability	• Concept of decentralization is lost • Majority of validators could manipulate the network	• Lisk • BitShares • EOS • Steemit
Proof of Authority Only selected group of nodes is able to validate block	• Very fast throughput • High efficiency in private chain setups • Scalability	• Concept of decentralization is lost • Administrators can manipulate the network if they concur.	• Typically private/permissioned blockchains

Proof of Work (PoW):

The first step in a PoW consensus algorithm is to choose which node gets to create a block. Then the nodes in the blockchain use their computer's processing power to solve complicated mathematical equations, which in itself is a process that requires a lot of computing power. The first one to solve the problem gets to create the block, which is then broadcasted to the entire network and this node is consequently rewarded with the respective blockchain's tokens. It functions consistently well, but it has disadvantages, such as the huge costs in cutting-edge equipment and electricity to mine efficiently, which results in the creation of mining pools in specific places (such as Asia) where the practice of mining is more convenient, such as cheap electricity. This creates centralization by putting a major portion of the hashing power in the hands of an exclusive group of the community, which is against the well-functioning laws of blockchain.

Proof of Stake (PoS)

Proof of Stake does not include the concept of miners like PoW, but includes validators, who mint or forge instead of mining.

In this process, a participator of the network that want to create the next block is randomly chosen in a process that takes into account by the amount of the cryptocurrency that is chosen to be deposited as stake. The validator, randomly chosen, is then assigned to validate all the block's transactions are indeed valid and, in the process, gets to earn the transaction fees in due time, after the rest of the network has time to guarantee that there was no tampering and that the transactions are indeed valid.

The drawback in this algorithm is that there is no incentivation for malicious activity on behalf of the validator, since there is cryptocurrency at stake and if malicious activity is uncovered, the stake would be lost. This algorithm also solves the problem with high energy consumption, since there is no actual need for high-end hardware equipment nor large amounts of electricity consumption. The negative aspect of this algorithm is that, if there is a fork caused by a block being found at the same time, there is no incentive for the participants to not mine in only one chain, which can originate double spending problems. In this model, the participants who have the most cryptocurrency tokens will tend to be assigned validators the most, since they are able to deposit a larger quantity at stake than other participators, which contributes to the rich getting even richer.

Delegated Proof of Stake (DPoS)

The Delegated Proof of Stake consists in picking a group of validators (from 21 to 100 network participants) and establishes this group as the party responsible for approving and validating the transaction blocks. This group is chosen and elected by all of the participants of the network. If one of the validators misses on the validation process or validates a transaction that was manipulated, the same validator is kicked out and replaced by another validator. The rewards attributed to validators are chosen by the voters.

There are clear advantages of picking this algorithm: it is much more organized and structured, which possibilitates the creation of new blocks in a rapid succession. But unfortunately, it also impossibilitates the remaining participants of the network, apart from the validators, to effectively mine and participate in the voluntary validation of transactions, like in PoW, making this algorithm somewhat unfair for most and of very exclusive nature, since the concept of decentralization is lost. It also reduces the need of high-grade mining hardware and high energy consumption.

Proof of Authority (PoA)

In this algorithm, the operation consists in rounds during which an elected group of accounts acts as mining leaders with the purpose of proposing new blocks by validating transactions. In theory, PoA is an optimized version of PoS that takes into account the identity as a factor for picking the validator(s) instead of the amount of cryptocurrency staked

4.4 Chameleon Hashes

Chameleon (or trapdoor) hash functions are a type of hash functions introduced by Krawczyk and Rabin in 1997.

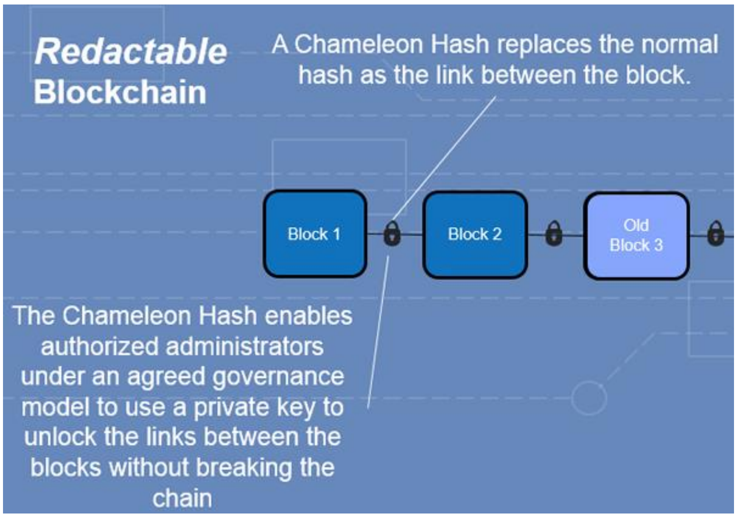

Redactable Blockchain
A Chameleon Hash replaces the normal hash as the link between the block.
Block 1
Block 2
Old Block 3
The Chameleon Hash enables authorized administrators under an agreed governance model to use a private key to unlock the links between the blocks without breaking the chain

Figure 2 - Chameleon Hashing. Source: https://www.bankingexchange.com/blogs/reporter-s-notebook/item/6492-a-blockchain-you-can-edit

A chameleon hash function is basically a non-standard type of hash function that is collision resistant (a situation where two distinct pieces of data have the same hash value). It is associated with a public and a private key, where the private key is known as *trapdoor* (Krawczyk & Rabin, 1997). It has the following properties:

- Anyone with the knowledge of the public key used is able to compute the associated hash function;
- For those that are not aware of the trapdoor, the hash function is collision-resistant, which means that it's infeasible to find two inputs which are mapped to the very same output;
- The owner(s) of the private key/trapdoor can easily acquire collisions for every given input.

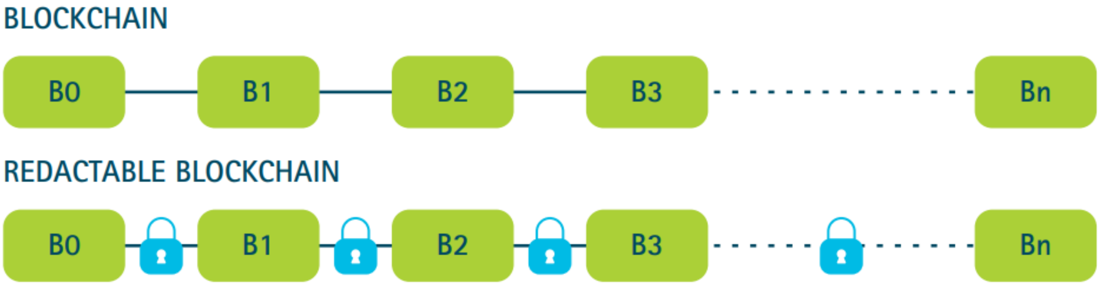


Figure 3 - Blochain with Chameleon Hashing Implemented. Source: Accenture.com

This means that this hash function potential lies in the possession of its private key, which possibilitates the recreation of matching algorithms. Typically, if there is any modification to a block in the ledger, the remaining blocks' integrity is compromised. But with chameleon hashing, the original blockchain remains fully intact, whereas this modification in another scenario would render the blockchain inoperable and would need to update the hash pointers of the affected blocks (Lumb, Treat, & Jelf, 2016). As designed, the peers of blockchains established under this technology would not be able to change anything. For them, the blockchain would remain immutable. This chameleon hash modification has been patented by Accenture.

5 Proof of Concept

The proposal in this thesis is to create a Proof-of-Concept (PoC) of a private, permissioned Blockchain using the Ethereum blockchain technology and to provide a concept of compliance between the Blockchain system and the GDPR. The solution to be applied will be adapted to a realistic example of a set of health institutions that want to store their appointments data on a blockchain-based application, which will be called Healthchain.

5.1 Specifications

Healthchain is a Blockchain and Java-based application that will store information of its clients' appointments, the participants of the appointments and some personal data, notably their name, email address and tax identification number. It will also store information of the health institutions and the medicine prescribed to the patient in the respective appointment, if there is any. The transactions will be triggered by smart contract functions called by the middleware layer through Application Programming Interface (API) calls and each appointment between a doctor and a patient was designed to constitute a transaction. Designing the system this way creates a perfect environment to test our hypothesis of whether or not it is indeed possible to comply with GDPR using blockchain technology. There is going to be a lot of personal data that will allow us to trace back to a subject (person) based on the data alone, from patients to doctors, which violates GDPR's article 17 (Wolford, 2019). Therefore, there will be a need for edition and/or removal of data regarding data subjects if they request so. This can be easily achieved on the application's operational database, since it will be based in a typical SQL database, but not regarding blockchain. This will be made possible in the blockchain if chameleon hashing is implemented, replacing the Ethereum's Ethash PoW function, which is the main mechanism behind the hashing process in Ethereum-based blockchains. This way, with access to the private key, our PoC blockchain can be exceptionally altered and have transactions edited and/or removed by finding the same hash for different inputs.

The Healthchain PoC is then based in the premise that it will be adapted to make use of chameleon hashing.

It will include three main layers:

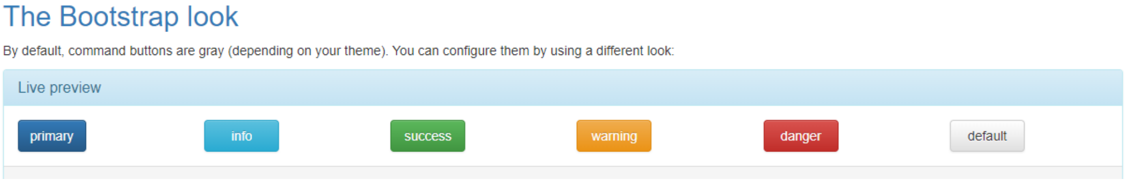
- User Interface Layer
- Middleware Layer
- Blockchain Layer

5.1.1 Full Project Vision and Requirements

- The blockchain will act as a digital certificate that will record an appointment. The main objective is to create an environment that is able to record data that can be used to track back people based on the personal data persisted on the blockchain.
- The hash function algorithm used in this Ethereum-based blockchain (Ethash) is to be replaced with Chameleon Hashing and adapted to perform in a similar way as the default version;
- The information will be held in a Ethereum private network;
- It will grant an easy access for auditing appointments, patients and doctors. It will enable a number of players to set and to get data into the network based on permissions;
- The transactions will be triggered by calls from the Java application to smart contracts deployed in the blockchain using Web3J;
- The data regarding Appointments, Doctors and Patients tables will be held inside the transactions own field “data”. This information is considered personal data.

5.2 User Interface Layer

For the user interface sub-layer there was used the Java Server faces (**JSF**) framework with BootsFaces extension, which implements JQuery technology along with Bootstrap. Both of these frameworks are part of BootsFaces, which is open-sourced.. This extension enables the development of user interfaces that are fully responsive and can be easily be used for a fully functioning graphical user interface for the end-user, besides being free of charge. below is an example of Bootsfaces buttons. But besides buttons, there are many other components such as data tables, dropdown lists, icons, sliders and more which are ready-to-use and will make the user interface more appealing and can be used with little developing effort.


The Bootstrap look

By default, command buttons are gray (depending on your theme). You can configure them by using a different look:

Figure 4 - Example of pre-made buttons from Bootsfaces

The Java Server Faces itself is a Java framework for web applications, popular for its flexibility and its component logic, making it an ideal choice for Model View Controller (MVC) architectures (Schalk, 2005). JSF development uses .xhtml extension files, which is the combination of Hyper Text Markup Language (HTML) pages with some Extensible Markup Language (known as XML) features.

For the authentication and security filtering was used the **Spring Security Framework** which was referred previously, which is an easy and safe configuration for this project.

5.3 Middleware Layer

The middleware layer will be developed using the Eclipse Integrated Development Environment (IDE), which is a popular choice for handling Java technologies, where one can develop Java code (or other languages) and compile it.

To complement the project and to ease the creation of new functionalities and to aid on the functioning of the application, Spring Framework will be implemented in the project. Spring Framework is an open-source lightweight framework for building Java Applications, possibilitating the integration of various frameworks with the source code in a project without conflicts or inconsistent (Cosmina, Schaefer, Ho, & Harrop, 2017).

It also possibilitates the use of Spring MVC implementation, which is a segregation between the services layer (controllers), the business layer (model) and the web layer (views). Spring Framework also includes

It also enables the use of Spring Data JPA. It’s an adaptation of JPA, which itself is an API definition for object-relational mappings and for managing database objects. Spring

Data JPA creates another layer which makes it useful for entity and association mappings, entity lifecycle management and to use JPA's query capabilities, as well as the creation of Spring Data repositories. These repositories are Java classes that allow persisting, updating or deleting entities with little coding necessary. (Janssen, 2019)

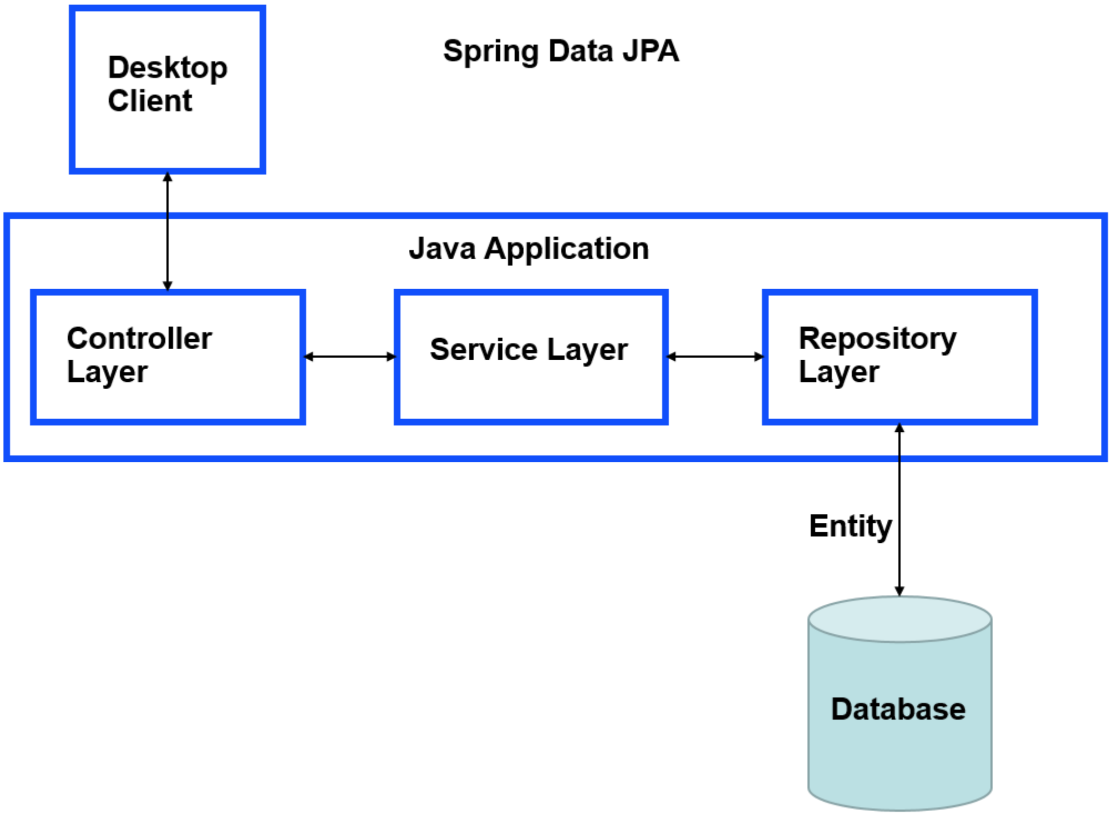


Figure 5 - Example of Spring Data object management

As a standalone sub-project that does not come with the Spring Framework, this PoC will also include Spring Security. Spring Security is a framework that provides authorization and authentication mechanisms for Java applications, enabling an access-control feature for the project. This way, certain pages can only be accessed after the user has logged in.

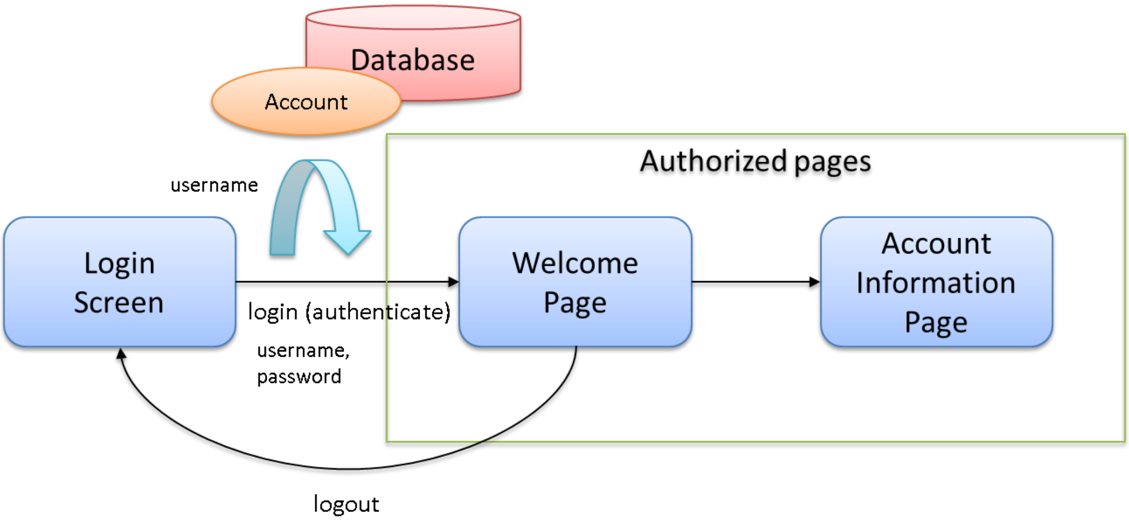
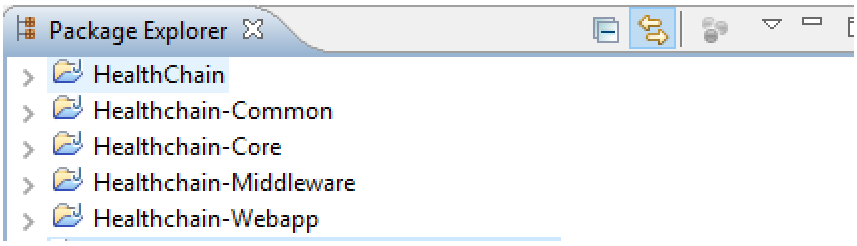


Figure 6 - Spring Security Use case example Source: https://terasolunaorg.github.io/guideline/5.2.0.RELEASE/en/Tutorial/TutorialSecurity.html

5.3.1 Java Project Structure

The project was created in Eclipse Integrated Development Environment, used to compile Java code and is also able to implement Apache Tomcat, which is used to deploy the middleware layer



HealthChain
Healthchain-Common
Healthchain-Core
Healthchain-Middleware
Healthchain-Webapp

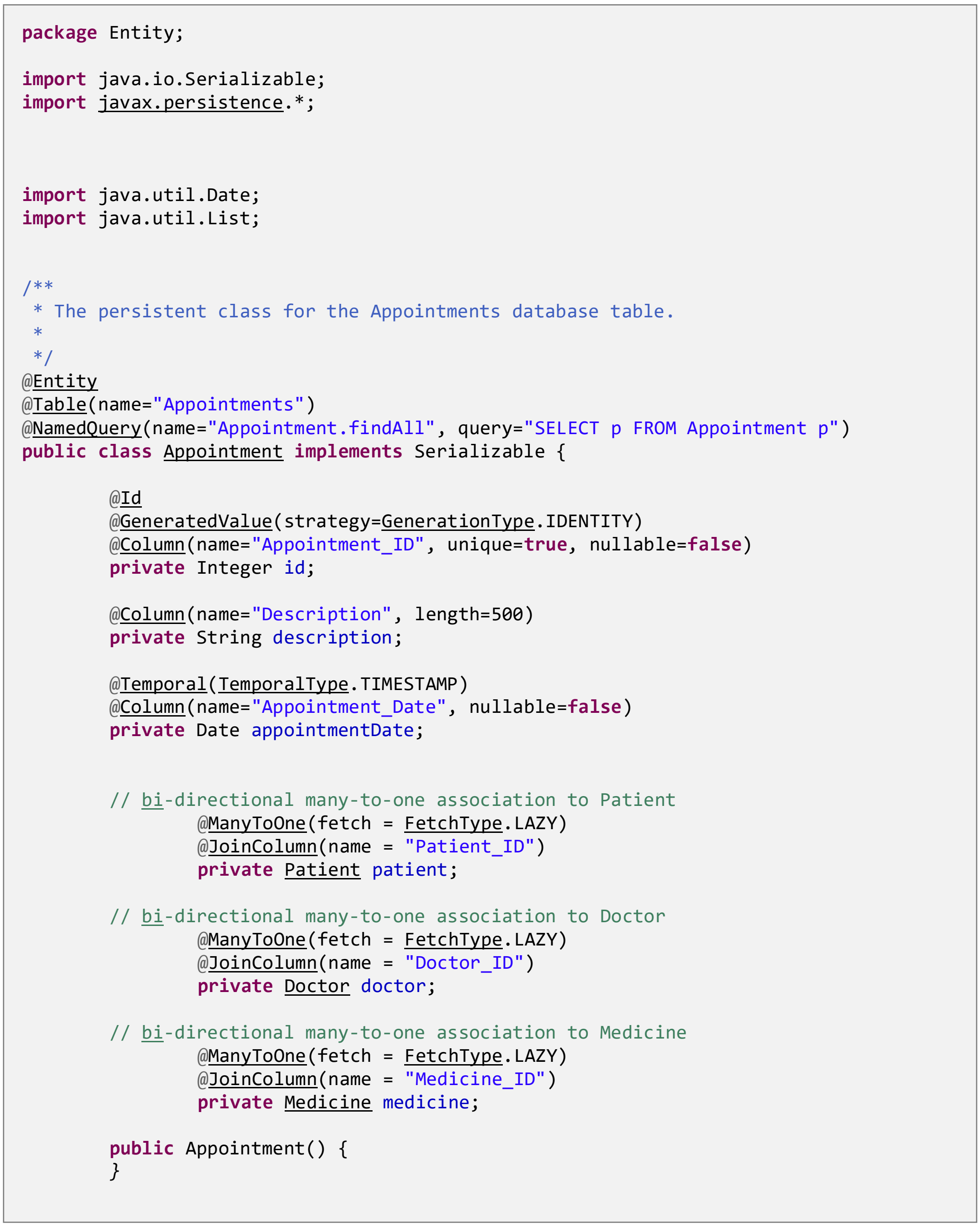
Figure 7 - Structure of the Java Project

Healthchain-Common

This project contains Data Transfer Objects (DTOs) that mirrors the existing entities in the database and in addition contains the interfaces of the services that act on the database, available in the middleware layer. This way, the application can be exposed to end-users and third-parties because it is only a cover of the business logic present in the Healthchain-Core project.

Healthchain-Core

Previously referred, this project is responsible for containing database entities, the repositories that allow direct interaction with the database and the entities. For the database entities, database management and JPA repository creation existent in this specific project, the Spring Data JPA was used. The below is an example of our Appointment class as an entity:



```java
package Entity;

import java.io.Serializable;
import javax.persistence.*;



import java.util.Date;
import java.util.List;


/**
 * The persistent class for the Appointments database table.
 *
 */
@Entity
@Table(name="Appointments")
@NamedQuery(name="Appointment.findAll", query="SELECT p FROM Appointment p")
public class Appointment implements Serializable {

        @Id
        @GeneratedValue(strategy=GenerationType.IDENTITY)
        @Column(name="Appointment_ID", unique=true, nullable=false)
        private Integer id;

        @Column(name="Description", length=500)
        private String description;

        @Temporal(TemporalType.TIMESTAMP)
        @Column(name="Appointment_Date", nullable=false)
        private Date appointmentDate;


        // bi-directional many-to-one association to Patient
                @ManyToOne(fetch = FetchType.LAZY)
                @JoinColumn(name = "Patient_ID")
                private Patient patient;

        // bi-directional many-to-one association to Doctor
                @ManyToOne(fetch = FetchType.LAZY)
                @JoinColumn(name = "Doctor_ID")
                private Doctor doctor;

        // bi-directional many-to-one association to Medicine
                @ManyToOne(fetch = FetchType.LAZY)
                @JoinColumn(name = "Medicine_ID")
                private Medicine medicine;

        public Appointment() {
        }
```

Figure 8 - Appointment as an Entity

Healthchain-Webapp

This project is responsible for the presentation/front-end and everything related, like graphical user interfaces and the Java classes present in this project will behave like controllers. This project folder will contain HTML and CSS classes used to create a graphical user interface to enable interaction between the user and the application.

5.3.2 Controllers and Webservices

The controllers in Healthchain-Webapp are implemented based on **Java Spring Framework**, and make all the integrations between the User interface layer, Data base and with the other middleware modules, for example, webservices (if needed). To accomplish this, the Spring Framework Java Beans will be used and it will be processed in Session scope, which means sessions for users are created and deleted based on the opening and closing of the user's browser window.

The controllers mentioned previously are responsible for the persistence connection, based in Spring Data. To produce the database queries, Spring Data using the JPA Repository was used, which are CRUD (Create, Read, Update and Delete):

- **Spring Data using the JPA Repository (Crud Repositories)** – This are used for queries, persisting objects, database management fro, JAva

The Maven project is capable of external connections and the integration with the Ethereum (Blockchain) Layer needs this component. Maven projects have features that are responsible for the Simple Object Access Protocol (SOAP) and Representational State Transfer (REST) WebServices communication.

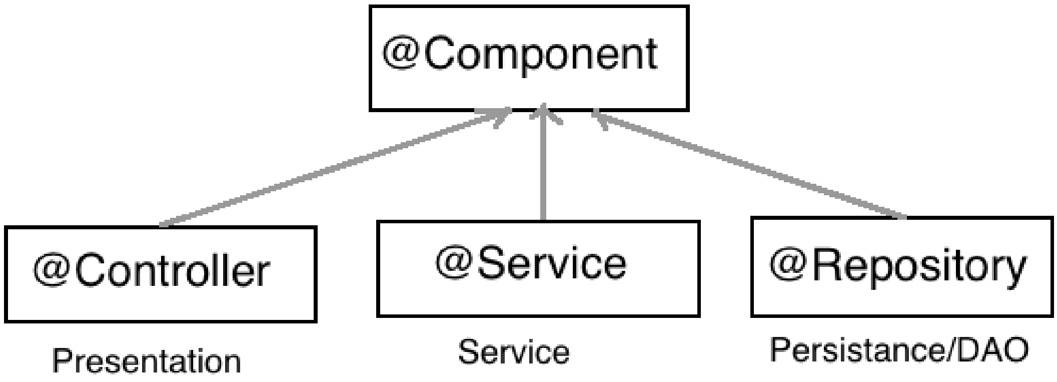


Figure 9 - Controllers Representation in Spring projects. Source: https://www.techgalery.com/2019/06/controllers-in-spring-framework.html

5.3.3 Apache Maven

This project will feature Apache Maven. Maven is a useful tool for building projects. It also serves as a project management tool, since besides having build capabilities, Maven can also run and generate reports, generate a website and ease communication among team members. (O'Brien et al., 2010).
The building capabilities of Maven consist of:

- Generating source code;
- Generating documentation from the source code;
- Compilation of the source code;
- Packaging and installation of the same packaged code in various repositories, like local or server repositories.

Besides being an automation tool, Maven also enables external and internal dependencies needed for the project. It is necessary for this project since Spring Framework and Web3J are to be implemented and that their respective classes, services and model objects will be used by our project.

5.3.4 Apache Tomcat

Apache Tomcat is an open-source, Java-based server and web application container that was developed with the purpose of running servlet and Java Server Pages (JSP). Since its inception, Tomcat evolved into a very popular project that is maintained as a standalone project apart from the Apache project (Vukotic & Goodwill, 2011).
In this context, Tomcat will be used to deploy the project's developments into a local site, which will function just like a normal webpage but only being accessible locally on the respective machine that deployed it. This way, it enables the representation of the code developed and also the ability to test functionalities and services developed in this PoC.

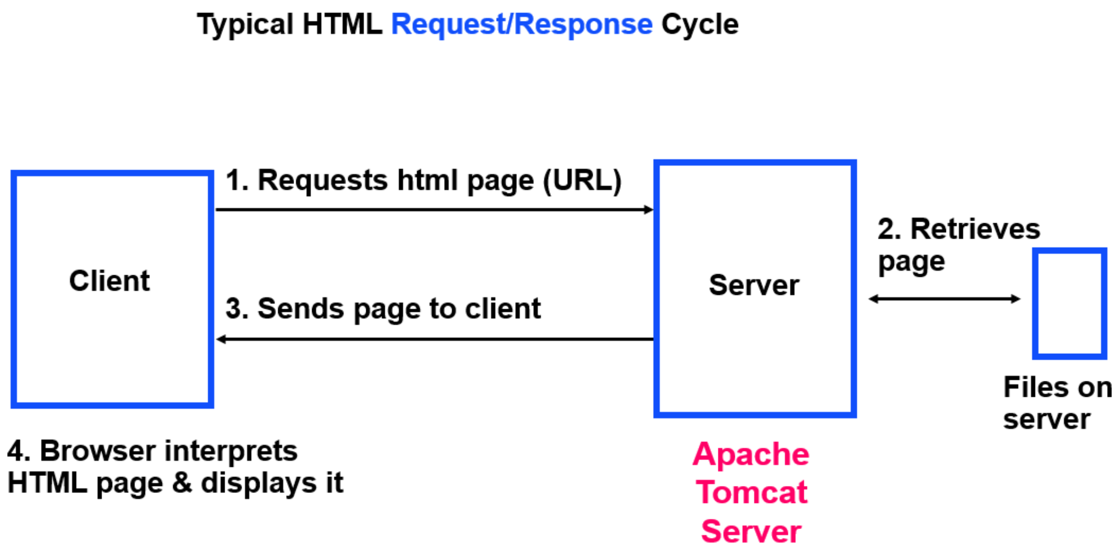

Typical HTML Request/Response Cycle
Client
1. Requests html page (URL)
3. Sends page to client
Server
2. Retrieves page
Files on server
4. Browser interprets HTML page & displays it
Apache Tomcat Server

Figure 10 - Representation of Tomcat functioning

5.3.5 Web3J

Web3J is a Java and Android library used with the purpose of working with SCs and to provide an easy way to interact and integrate with Ethereum clients (nodes). This possibilitates working with the Ethereum blockchain without having to develop integration code for Java-Geth communication.It is the official java port of Web3 abstraction library.

Web3J has some prominent features which will be fundamental for this PoC, which are:

- Interaction with Ethereum JSON-RPC client API over HTTP;
- Ethereum wallet support;
- Generation of Java smart contract wrappers, which are Java representations of the smart contract and its functions. These are used to create, deploy, transact with and call SCs from Java code.

5.3.6 Database

The database used on this PoC is the MySQL**,** which is one of the most popular Open Source Databases. This database is used to store everything that needs to be configured or maintained in the application and is additionally used to store the information for the Healthchain application that is not stored on the Blockchain Layer.

This way we can have a solid and functioning application providing the service of storing appointments and other data related to the individuals associated with the process without being overly dependent on using blockchain for storage while also creating a fully functioning middleware layer.

Figure 11 - MySQL Logo. Source: Wikipedia.com

5.3.6.1 Data Model

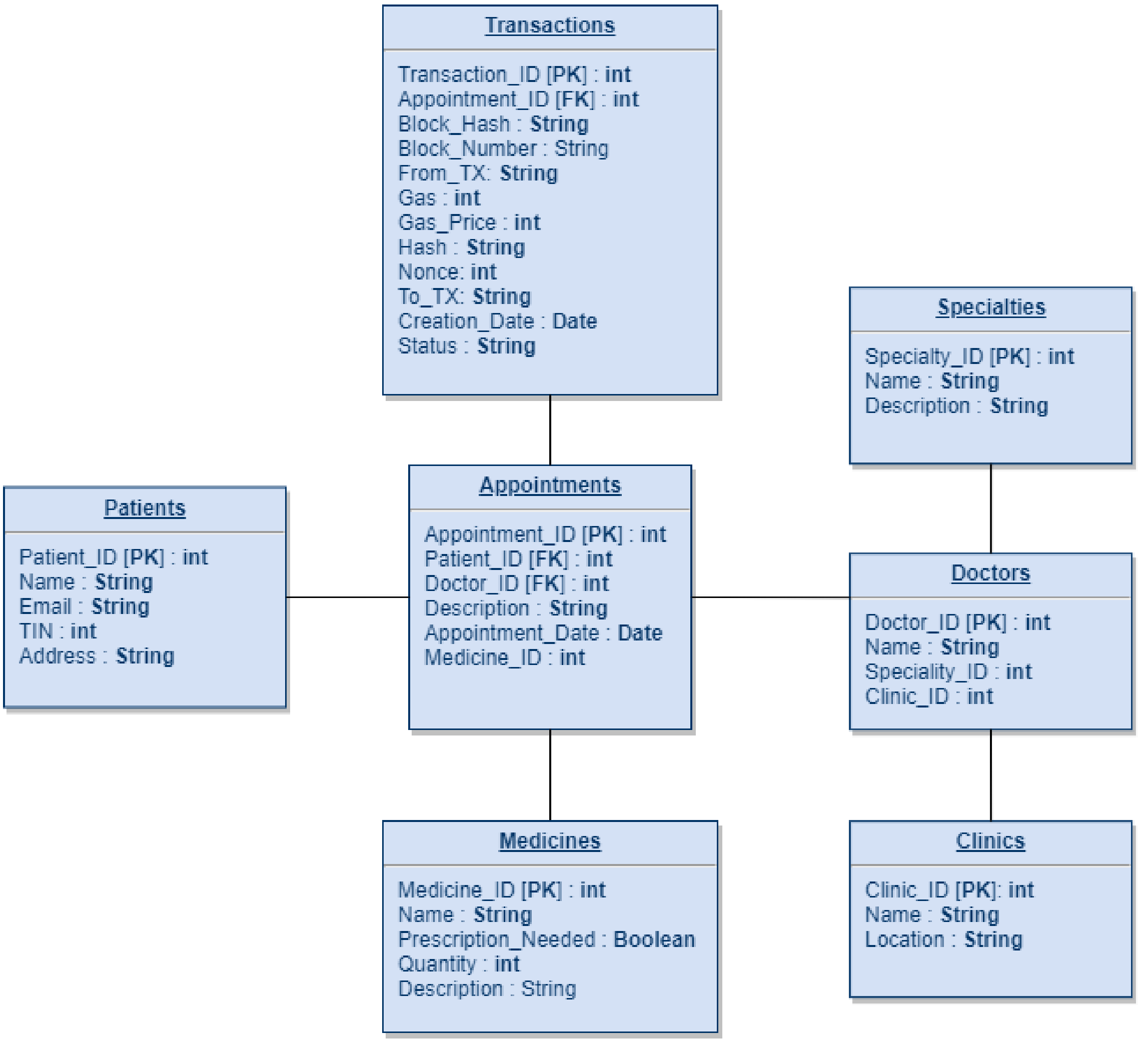


Figure 12 - Healthchain's Data Model

5.3.6.2 Tables

Table 2 - List of tables in Healthchain

Table	Description	Purpose
Appointments	List of appointments	Application
Patients	List of application users	Application
Doctors	List of application doctors	Application
Specialty	List of specialties	Application
Clinics	List of clinics	Application
Medicine	List of existing medicines	Application
Transactions	List of information regarding the blockchain and existing transactions.	Blockchain

Appointments

Table 3 - Appointments Table

Columns	Descriptions	Type	Required
Appointment_ID	Table primary key	INT(11)	Yes
Patient_ID	Reference to the respective patient	INT(11)	Yes
Doctor_ID	Reference to the respective doctor	INT(11)	Yes
Description	Field for appointment description or other notes	VARCHAR(3000)	No
Appointment_Date	Date when appointment occurred	DATETIME	Yes
Medicine_ID	Reference to the prescribed medicine, if there is any.	INT(11)	No

Patients

Table 4 - Patients Table

Columns	Descriptions	Type	Required
Patient_ID	Table primary key	INT(11)	Yes
Name	Name of the patient	VARCHAR(80)	Yes
Email	Email address of the patient	VARCHAR(40)	Yes
TIN	Tax identification number of the patient	INT(15)	Yes
Address	Physical address of the patient	VARCHAR(120)	Yes

Doctors

Table 5 - Doctors Table

Columns	Descriptions	Type	Required
Doctor_ID	Table primary key	INT(11)	Yes
Name	Name of the doctor	VARCHAR(80)	Yes
Specialty_ID	Specialty's ID of the doctor	INT(11)	Yes
Clinic_ID	Clinic's ID of the doctor	INT(11)	Yes

Specialties

Table 6 - Specialties Table

Columns	Descriptions	Type	Required

Specialty_ID	Table primary key	INT(11)	Yes
Name	Designation of the specialty	VARCHAR(80)	Yes
Description	Description of the specialty	VARCHAR(300)	Yes

Clinics

Table 7 - Clinics Table

Columns	Descriptions	Type	Required
Clinic_ID	Table primary key	INT(11)	Yes
Name	Name of the clinic	VARCHAR(80)	Yes
Location	Physical address of the clinic	VARCHAR(300)	Yes

Medicines

Table 8 - Medicines Table

Columns	Descriptions	Type	Required
Medicine_ID	Table primary key	INT(11)	Yes
Name	Name of the medicine	VARCHAR(80)	Yes
Prescription_Needed	Flag to verify if prescription is needed. 1-Yes; 2- No	INT(1)	Yes
Quantity	Quantity prescribed	INT(11)	Yes
Description	Description of the medicine	VARCHAR(300)	Yes

Transactions

Table 9 - Transactions Table

Columns	Descriptions	Type	Required
Transactions_ID	Table primary key	INT(11)	Yes
Appointment_ID	Reference to the respective appointment	INT(11)	Yes
Block_Hash	Hash from the Block related with Transaction	VARCHAR(300)	Yes
Block_Number	Number from the Block related with Transaction	INT(11)	Yes
From_TX	Hash of the previous transaction	VARCHAR(300)	Yes
Gas	Gas used in the transaction	INT(11)	Yes
Gas_Price	Gas price of the transaction	INT(11)	Yes
Hash	Hash of the current transaction	VARCHAR(300)	Yes
Nonce	Nonce value of the transaction	INT(80)	Yes
To_TX	Hash to the next transaction	VARCHAR(300)	Yes
Creation_Date	Creation date of the transaction	DATETIME	Yes
Status	Status of the transaction (Pending/Failed/Successful)	VARCHAR(30)	Yes

5.4 Blockchain Layer

5.4.1 Ethereum Client

The Ethereum blockchain is a complete decentralized platform that allows the possibility to implement smart contracts inside the network (Wood, 2017). It is based in the Ethereum Virtual Machine maintained by a multiple set of nodes. The traditional systems for networks are usually splitted in servers and clients. The server will enable the infrastructures by providing databases and access to the resources. In the Ethereum ecosystem the Ethereum nodes can execute different tasks as they act as server and client simultaneously while they can be also miners.

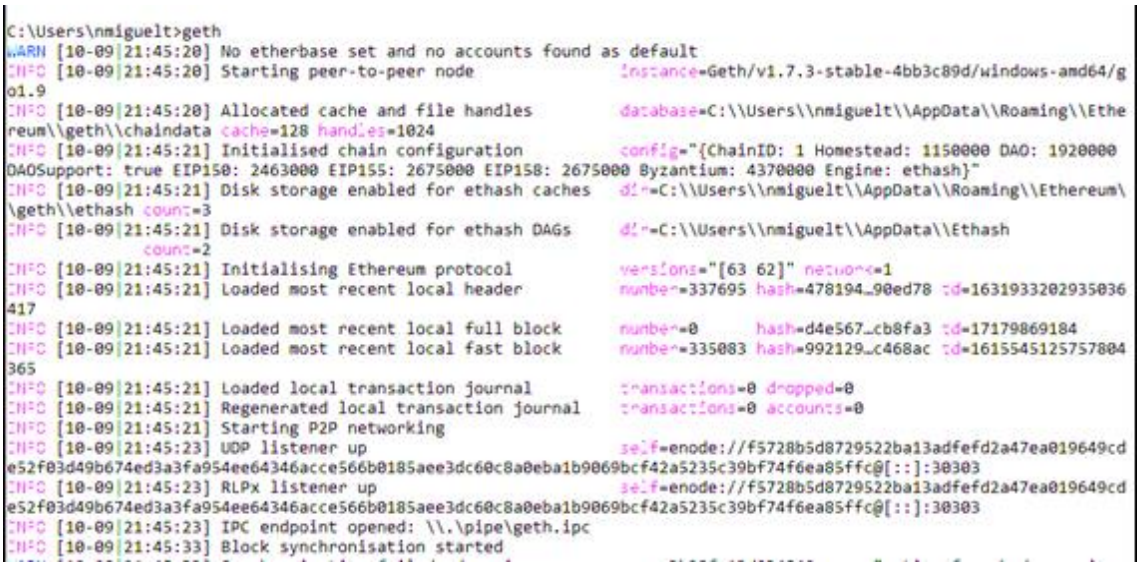
- **Client / Server**: As the Ethereum network is based in a decentralized system the information as is storage in the different nodes having all the nodes the same valid ledger of information;
- **Miners**: The nodes can perform as a miner, where they will process new transactions into the network blocks by performing the consensus mechanism.

5.4.1.1 Geth

The chosen option to operate the Ethereum client was the download of the Geth binary package with the version v1.7.3 (at the time of writing, the last stable version was v1.8.2).

The access to the Geth client can be done by HTTP JSON-RPC, API's (Web3, eth and SSH) or using the built-in management API methods developed specifically for the Command Prompt. For this PoC, the last one was used.

Deploying a Local Network with Geth



```
C:\Users\nmiguelt>geth
WARN [10-09|21:45:20] No etherbase set and no accounts found as default
INFO [10-09|21:45:20] Starting peer-to-peer node               instance=Geth/v1.7.3-stable-4bb3c89d/windows-amd64/g
o1.9
INFO [10-09|21:45:20] Allocated cache and file handles         database=C:\\Users\\nmiguelt\\AppData\\Roaming\\Ethe
reum\\geth\\chaindata cache=128 handles=1024
INFO [10-09|21:45:21] Initialised chain configuration          config="{ChainID: 1 Homestead: 1150000 DAO: 1920000
DAOSupport: true EIP150: 2463000 EIP155: 2675000 EIP158: 2675000 Byzantium: 4370000 Engine: ethash}"
INFO [10-09|21:45:21] Disk storage enabled for ethash caches   dir=C:\\Users\\nmiguelt\\AppData\\Roaming\\Ethereum\
\geth\\ethash count=3
INFO [10-09|21:45:21] Disk storage enabled for ethash DAGs     dir=C:\\Users\\nmiguelt\\AppData\\Ethash
            count=2
INFO [10-09|21:45:21] Initialising Ethereum protocol           versions="[63 62]" network=1
INFO [10-09|21:45:21] Loaded most recent local header          number=337695 hash=478194…90ed78 td=1631933202935036
417
INFO [10-09|21:45:21] Loaded most recent local full block      number=0      hash=d4e567…cb8fa3 td=17179869184
INFO [10-09|21:45:21] Loaded most recent local fast block      number=335083 hash=992129…c468ac td=1615545125757804
365
INFO [10-09|21:45:21] Loaded local transaction journal         transactions=0 dropped=0
INFO [10-09|21:45:21] Regenerated local transaction journal    transactions=0 accounts=0
INFO [10-09|21:45:21] Starting P2P networking
INFO [10-09|21:45:23] UDP listener up                          self=enode://f5728b5d8729522ba13adfefd2a47ea019649cd
e52f03d49b674ed3a3fa954ee64346acce566b0185aee3dc60c8a0eba1b9069bcf42a5235c39bf74f6ea85ffc@[::]:30303
INFO [10-09|21:45:23] RLPx listener up                         self=enode://f5728b5d8729522ba13adfefd2a47ea019649cd
e52f03d49b674ed3a3fa954ee64346acce566b0185aee3dc60c8a0eba1b9069bcf42a5235c39bf74f6ea85ffc@[::]:30303
INFO [10-09|21:45:23] IPC endpoint opened: \\.\pipe\geth.ipc
INFO [10-09|21:45:33] Block synchronisation started
```

Figure 13 - Geth Deployment

The API used in this has some main commands that are defined in the table below. All of them have different options and can be parameterized by necessity and are described in detail.

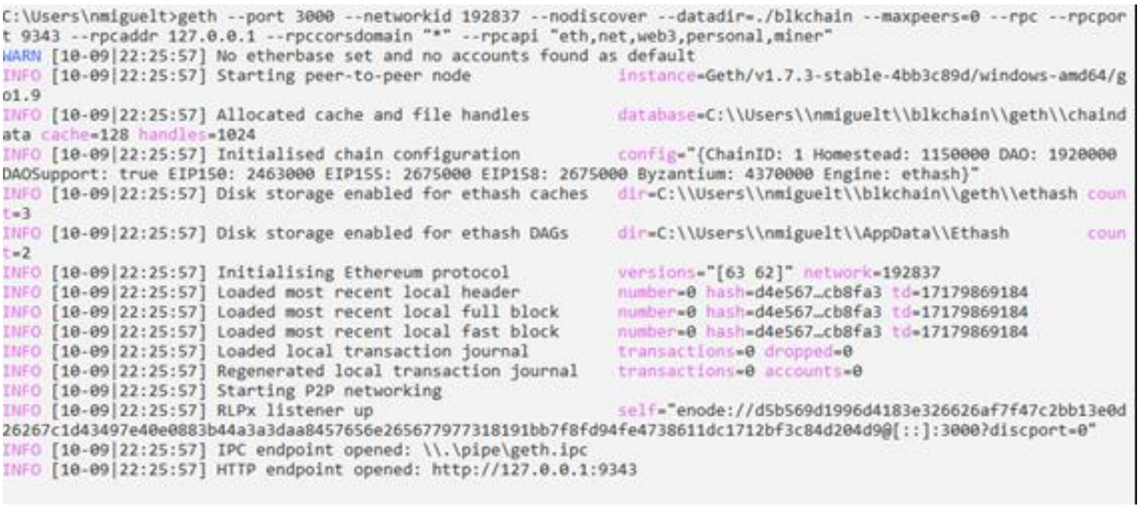
Table 10 - Geth’s Management API

API set	Description
admin	Geth node management
debug	Geth node debugging
miner	Miner management;
personal	Account management
txpool	Transaction pool inspection

Geth possibilitates the creation of a private network. By setting up and running specific commands, it enables us to set up a private network. In this case, the command used is detailed below:

```
geth --port 3000 --networkid 192837 --nodiscover --datadir=./blkchain --
maxpeers=0  --rpc --rpcport 9343 --rpcaddr 127.0.0.1 --rpccorsdomain "*" --
rpcapi "eth,net,web3,personal,miner"
```

Figure 14 - Geth's Node Launch Command



```
C:\Users\nmiguelt>geth --port 3000 --networkid 192837 --nodiscover --datadir=./blkchain --maxpeers=0 --rpc --rpcpor
t 9343 --rpcaddr 127.0.0.1 --rpccorsdomain "*" --rpcapi "eth,net,web3,personal,miner"
WARN [10-09|22:25:57] No etherbase set and no accounts found as default
INFO [10-09|22:25:57] Starting peer-to-peer node            instance=Geth/v1.7.3-stable-4bb3c89d/windows-amd64/g
o1.9
INFO [10-09|22:25:57] Allocated cache and file handles      database=C:\\Users\\nmiguelt\\blkchain\\geth\\chaind
ata cache=128 handles=1024
INFO [10-09|22:25:57] Initialised chain configuration       config="{ChainID: 1 Homestead: 1150000 DAO: 1920000
DAOSupport: true EIP150: 2463000 EIP155: 2675000 EIP158: 2675000 Byzantium: 4370000 Engine: ethash}"
INFO [10-09|22:25:57] Disk storage enabled for ethash caches   dir=C:\\Users\\nmiguelt\\blkchain\\geth\\ethash coun
t=3
INFO [10-09|22:25:57] Disk storage enabled for ethash DAGs     dir=C:\\Users\\nmiguelt\\AppData\\Ethash         coun
t=2
INFO [10-09|22:25:57] Initialising Ethereum protocol        versions="[63 62]" network=192837
INFO [10-09|22:25:57] Loaded most recent local header       number=0 hash=d4e567…cb8fa3 td=17179869184
INFO [10-09|22:25:57] Loaded most recent local full block   number=0 hash=d4e567…cb8fa3 td=17179869184
INFO [10-09|22:25:57] Loaded most recent local fast block   number=0 hash=d4e567…cb8fa3 td=17179869184
INFO [10-09|22:25:57] Loaded local transaction journal      transactions=0 dropped=0
INFO [10-09|22:25:57] Regenerated local transaction journal    transactions=0 accounts=0
INFO [10-09|22:25:57] Starting P2P networking
INFO [10-09|22:25:57] RLPx listener up                      self="enode://d5b569d1996d4183e326626af7f47c2bb13e0d
26267c1d43497e40e0883b44a3a3daa8457656e265677977318191bb7f8fd94fe4738611dc1712bf3c84d204d9@[::]:3000?discport=0"
INFO [10-09|22:25:57] IPC endpoint opened: \\.\pipe\geth.ipc
INFO [10-09|22:25:57] HTTP endpoint opened: http://127.0.0.1:9343
```

Table 11 - Setting Geth’s Private Network

Networks

The Ethereum is supported by the decentralized peer-to-peer network. There is a main network where most production Ethereum deployments are situated but there are several other testnets. The table below lists the most important, but there are others, which cannot be equal to ours.

Table 12 - Existing Ethereum Networks

Network Id	Description
0	Olympic, Ethereum public pre-release testnet
1	Frontier, Homestead, Metropolis, the Ethereum public main network
2	Classic, the (un)forked public Ethereum Classic main network, chain ID 61
3	Expanse, an alternative Ethereum implementation, chain ID 2
4	Morden, the public Ethereum testnet, now Ethereum Classic testnet
5	Ropsten, the public cross-client Ethereum testnet
6	Rinkeby, the public Geth PoA testnet
7	Kovan, the public Parity PoA testnet

— **nodiscover:** Disables the mechanism for peer discovery. Peers need to be added manually

— **datadir:** The data directory where the blockchain data will be stored.

— **maxpeers:** Maximum number of peers allower. When value is set to 0, network is disabled.

— **rpc:** Enable the HTTP-RPC server, which allows the use of specific commands.

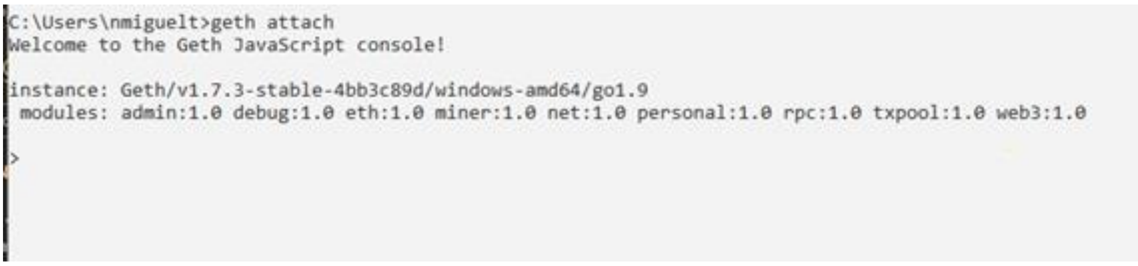
— **rpcapi:** Allows communication with the Ethereum network using the web3js RPC methods in the Geth javascript console.

JavaScript Console Interaction

There is available a built-in javascript runtime environment Console (JSRE). This console enables dynamic interaction and is a main advantage for working with our local network. This console can be launched at any point, even after the network is deployed. It is also based on the Web3js library, which simplifies the syntax for requesting/setting information in this respective network. The command to launch it is described below:

```
geth attach
```

Figure 15 - Geth Javascript Console Launch Command



```
C:\Users\nmiguelt>geth attach
Welcome to the Geth JavaScript console!

instance: Geth/v1.7.3-stable-4bb3c89d/windows-amd64/go1.9
 modules: admin:1.0 debug:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

>
```

Figure 16 - Geth's Javascript Console

With this console seen above, it enables the creation of accounts and addresses, management of the node/network settings and information, enabling connections and others. If some request is not possible by the JS console, there is also a possibility of additional support of management Application Programming Interfaces (API) by Geth.

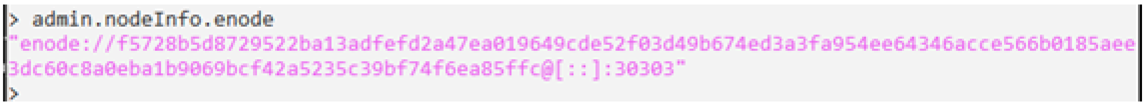
If necessary, specific peers can be added as miners, so a solid, ruled network can be created.

- The flag --maxpeers can be used in the CMD to determine the number of peers connected to our client. If zero is selected only the client is open;
- In the Geth JavaScript console, one can add peers with the flag “admin.addpeer()”;
    - Inside the brackets the target URL of the Ethereum Node (enode) should be placed.
    - The format will be the node URL followed by a @ the IP of the target node and the port.
    - The enode of our node, which is the node’s Uniform Resource Identifier (URI) can be found in the client by typing the command below:

```
admin.nodeInfo.enode
```

Figure 17 - Node URI Info Command

The result is shown below:



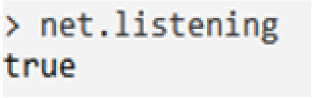
```
> admin.nodeInfo.enode
"enode://f5728b5d8729522ba13adfefd2a47ea019649cde52f03d49b674ed3a3fa954ee64346acce566b0185aee
3dc60c8a0eba1b9069bcf42a5235c39bf74f6ea85ffc@[::]:30303"
>
```

Figure 18 - Node URI

- It's possible to test the connectivity of the network in the JS console by applying the following command. It will provide the status of the network (if connected, it returns true):

```
net.listening
```

Figure 19 - Network Status Command



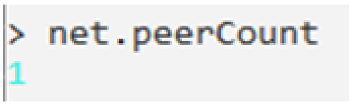
```
> net.listening
true
```

Figure 20 - Network Status Command Result

- The amount of peers in the network can also be found with this flag:

```
net.peerCount
```

Figure 21 - Network Participants Number Command



```
> net.peerCount
1
```

Figure 22 - Number of Network Participants

In Figure 23 we can see the result of the process of Geth's node deployment. There is no interaction with Ethereum's main network and we successfully created an isolated, permissioned network that needs special permissions to be accessed, like intended.

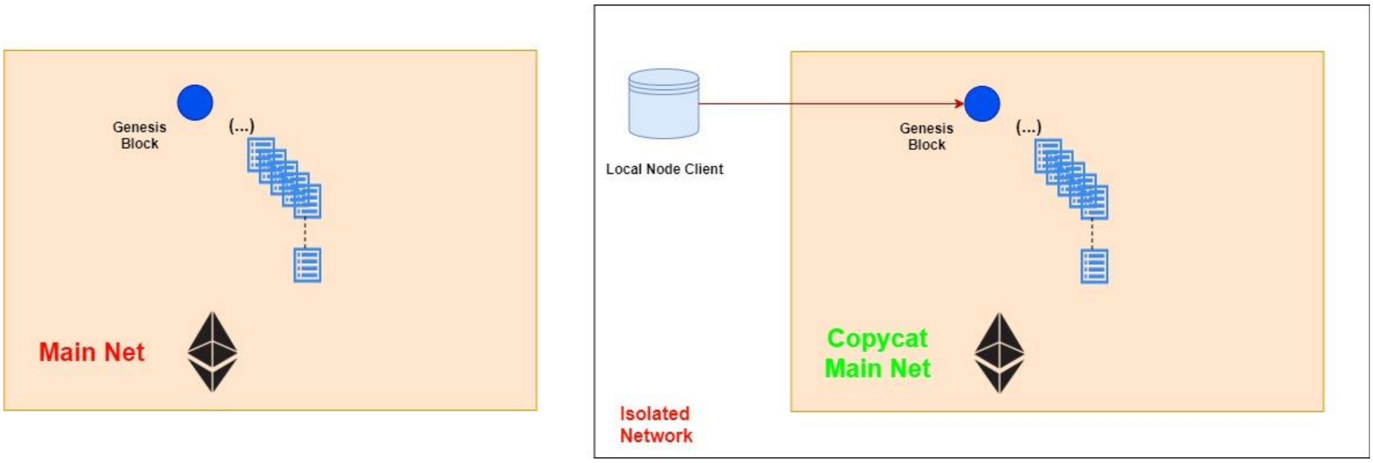


Figure 23 - Representation of Geth deployment

5.4.2 Smart Contracts

The transactions that are submitted in the network can interact with the other key features of the blockchain, like the smart-contracts. The Ethereum network itself is a static platform that only changes its state upon transfers of the Ethereum token (ether) or due to creation and requests of smart contracts.

Smart contracts (SC) are a key factor in the added value of the Ethereum network. They allow for crafting snippets of code that are completely transparent and with the objective of operating without any possibility of downtime, censorship or fraud. (Christidis & Devetsikiotis, 2016)
The smart contracts exist, are present in the network and can be triggered by external requests or when a set of conditions is achieved. They can go from a line of code to a complex set of instructions and can be written in different languages. For this dissertation, the Solidity language was the language picked.

It is important to clarify some concepts to understand better how the smart contracts are deployed and interact with the network (Ethereum Community, 2014):

- In order to be able to send data to the network one has to possess an account;
- There are two types of accounts:

- Private key ruled: Each account have a private a public key. The concept of proof of identity is based in hashing crypto algorithms associated to this specific keys;
- Smart contract ruled: Controlled by the inherent code of the smart contract;

- Each account is associated with a singular and exclusive address; The address is described by a hashing key of 20 bytes linked with with the private key of the account or with the SC creator address (Luu, Chu, Olickel, Saxena, & Hobor, 2016);
- The Ethereum transactions have a specific normalized data model with a set of 6 principal fields that are described in the table below;

A simple ether transaction will cross the network using all the fields except the data field that can be used to attach extra information to the transaction. The fields are described in - Ethereum Transaction Data StructureTable 13:

Table 13 - Ethereum Transaction Data Structure

Data Field	Description
nonce	Is a transaction counting sequence number associated with the sending account;
gasprice	Quantity of gas that one is willing to pay for executing the transaction; Is correlated with the speed of deployment of the transaction. The more data is included in the transaction, the more gas it costs.
startgas	Maximum amount of gas that one is willing to pay for the transaction to be executed.
to	Destination address
value	Amount of ether that should be transacted
data	Attached input information data;

With the above concepts one could conclude that all the network transactions would follow the same scheme of a regular ether transaciton. But the crucial difference that enables the SC comes from the “data” field that can receive any kind of data, which is a fundamental aspect.

From an abstract point of view, the SC code is written and is then transformed to a hexadecimal bytecode input that is attached as the “data” in the regular transaction. This information will be then recognized by the Ethereum nodes and is built in the network on a specific address, allowing from that moment for the SC enhanced features.

The figure below details in a simplified way the process to submit a SC into the network. The code of the smart code is compiled and we should have focus on 2 principal products that are generated:

- The SC is translated into binary data that is attached into a regular transaction;
- A JSON-RPC file (ABI) is generated to interact in the future with the SC after the network deployment.

Then the SC binary information will be added to a regular transaction with a hashing due to a cryptographic process. The resultant transaction candidate is then submitted by a client node into a pool of transactions. The miners will then process the transaction into a block that will be submitted into the process of consensus.
If the transaction gets accepted into the network the SC will become available for interaction.

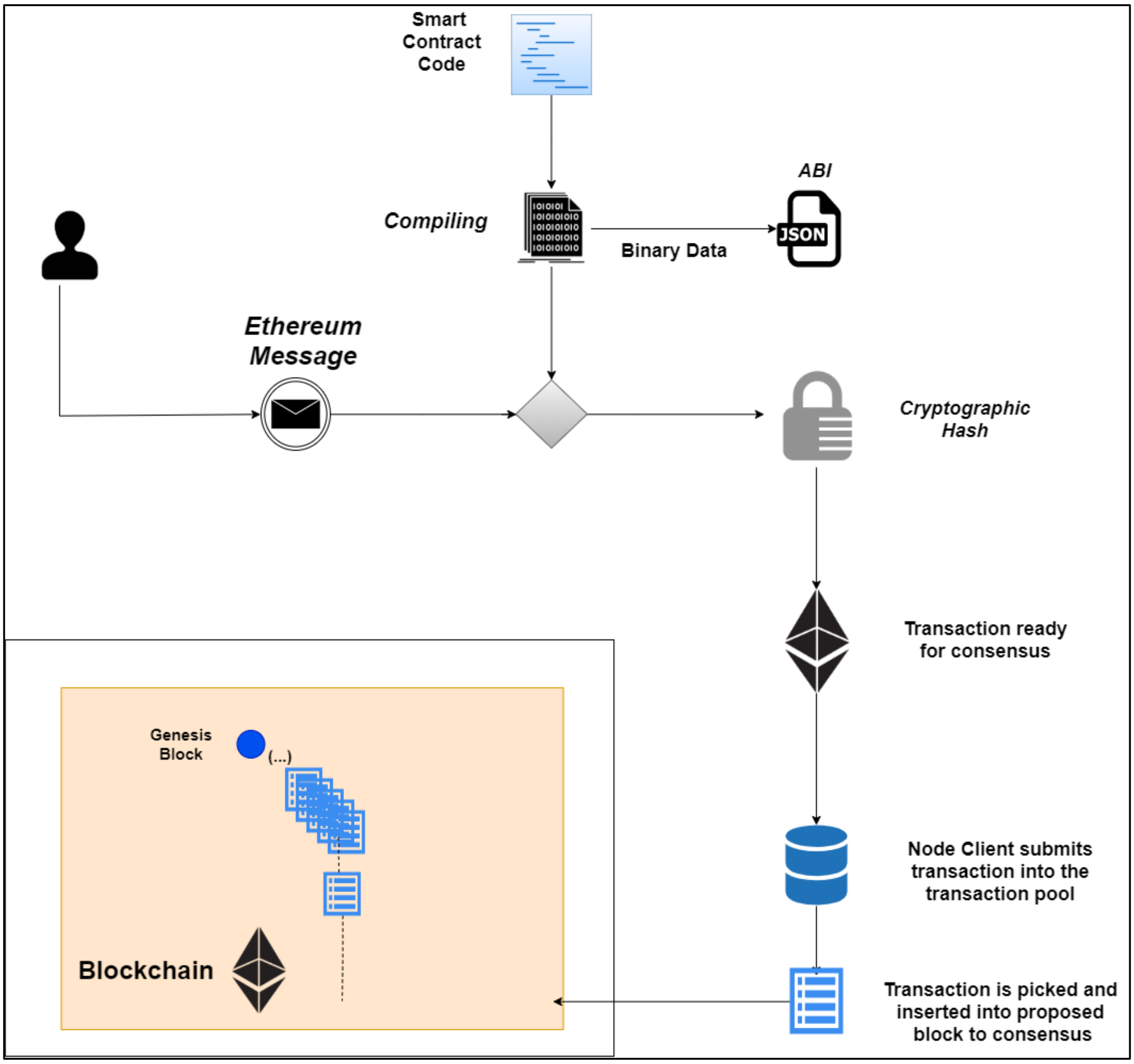

Smart Contract Code
Compiling
ABI
JSON
Binary Data
Ethereum Message
Cryptographic Hash
Transaction ready for consensus
Node Client submits transaction into the transaction pool
Transaction is picked and inserted into proposed block to consensus
Genesis Block
(...)
Blockchain

Figure 24 - Smart Contract Deployment into Ethereum Blockchain network

Doing an analogy, a deployed SC can be perceived as a web service that can be accessed from any node in our network and provide a set of enhanced features depending on the code written in the SC.

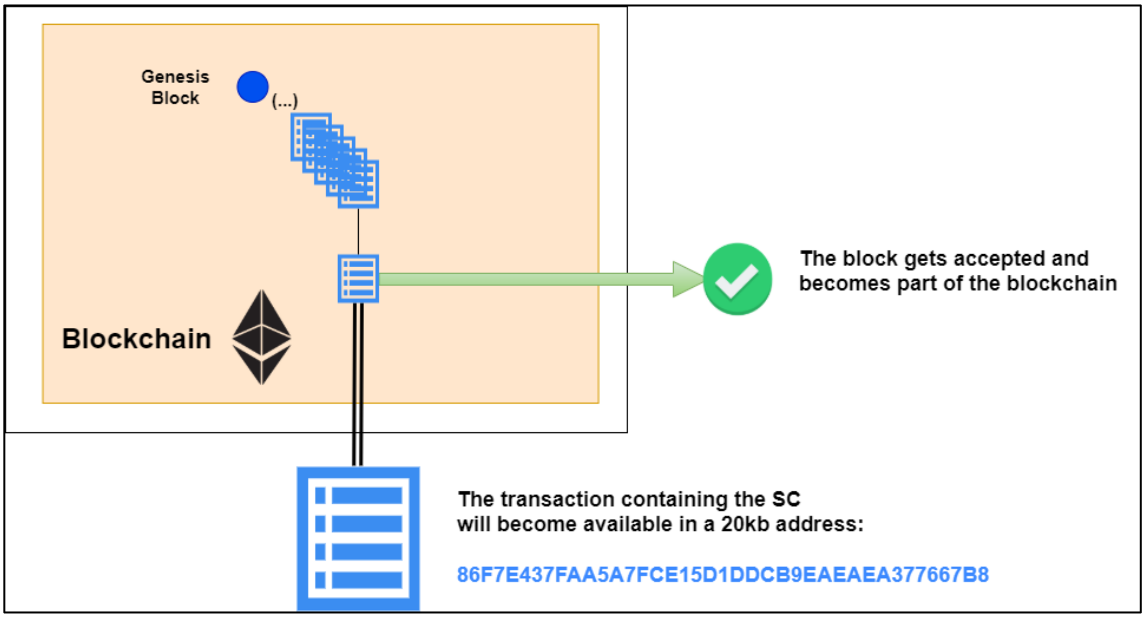

Genesis Block
(...)
Blockchain
The block gets accepted and becomes part of the blockchain
The transaction containing the SC will become available in a 20kb address:
86F7E437FAA5A7FCE15D1DDCB9EAEAEA377667B8

Figure 25 - Deployed Smart Contract

As web services are traditionally accessed using an Application Programming Interface (API), the SC will be accessed by an ABI (Application binary interface). An ABI is a low-level format that will define how data structures can be accessed defining the parameters to invocate the intrinsic functions of the SC. This ABI can be perceived as the bridge between the SC and the person that is trying to interact with it. The ABI is a product of the compilation of the contract as explained and is written in JSON. However, Web3J also includes an API to interact with the SC, which can trigger transactions from one address to another.

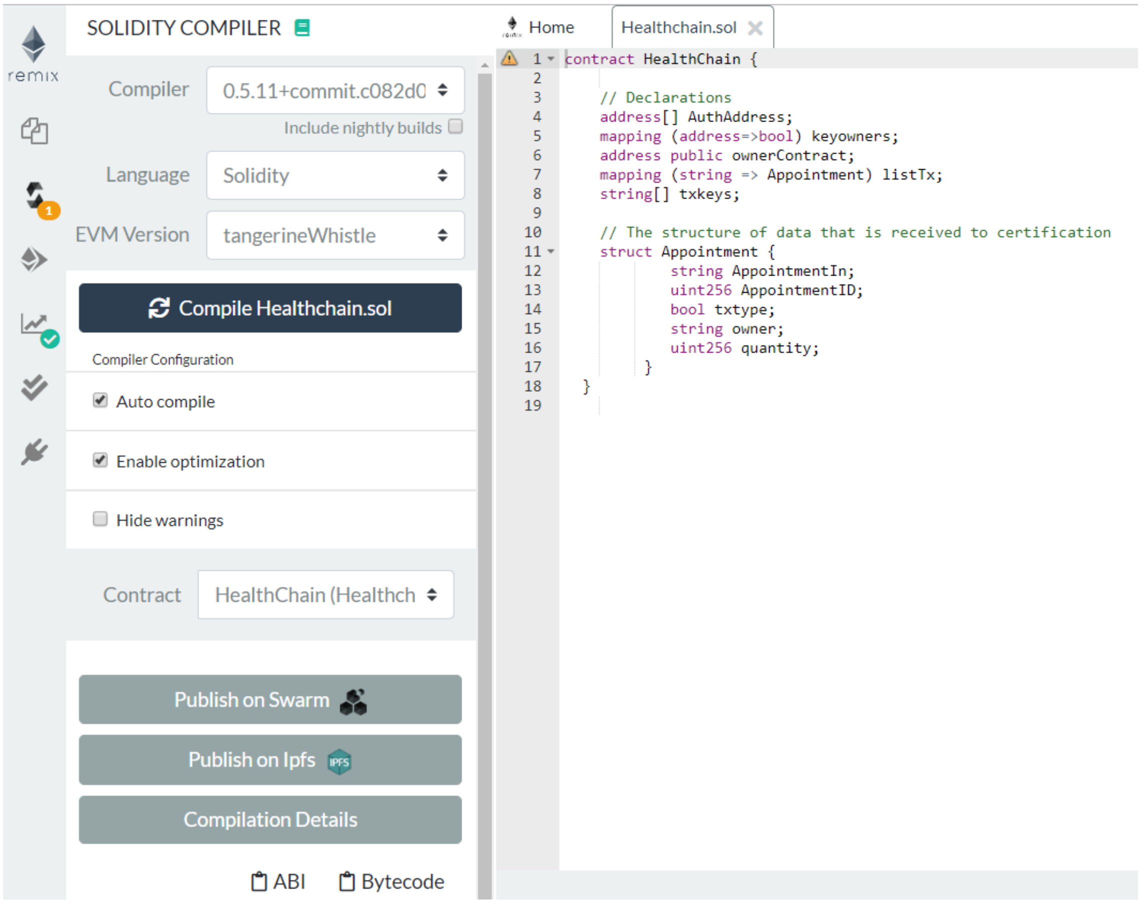
5.4.2.1 Smart Contract Deployment

The Smart contract had to be deployed into the Ethereum networks. The Ethereum foundation software Remix was used.

It is the advised development tool for small solidity scripts. It’s supported by Ethereum and was the one chosen to design SCs because:

- It enables the Solidity language;
- Solidity is a language that requires compiling – the Remix tool is able to compile it;
- Is web-based and therefore web-browser compatible.

- Allows for deploying smart contracts mockups into a simulated JavaScript network, which is fundamental for this PoC.



Compiler: 0.5.11+commit.c082d0

□ Include nightly builds

Language: Solidity

EVM Version: tangerineWhistle

☑ Auto compile

☑ Enable optimization

□ Hide warnings

Contract: HealthChain (Healthch

```
contract HealthChain {

    // Declarations
    address[] AuthAddress;
    mapping (address=>bool) keyowners;
    address public ownerContract;
    mapping (string => Appointment) listTx;
    string[] txkeys;

    // The structure of data that is received to certification
    struct Appointment {
            string AppointmentIn;
            uint256 AppointmentID;
            bool txtype;
            string owner;
            uint256 quantity;
        }
  }

```

Figure 26 - Remix IDE Graphic User Interface

The deployment of SCs into the network can be compared to the deployment of an application/webservice for API consuming. It will behave as a snippet of code that is available in the network in available at all times as long as the network lives. As an API the SC will have internal functions for is inside logic and external functions that can be accessed by request of users (or even other smart contracts).

5.4.2.2 Development Environment

The development environment is built totally inside the REMIX browser that empowers a full development framework with a built in JavaScript Virtual machine that will “host” a block as part of a mock blockchain.

This framework will enable tools to interact with the functions inside the SC.

The functional diagram are expressed in the below:

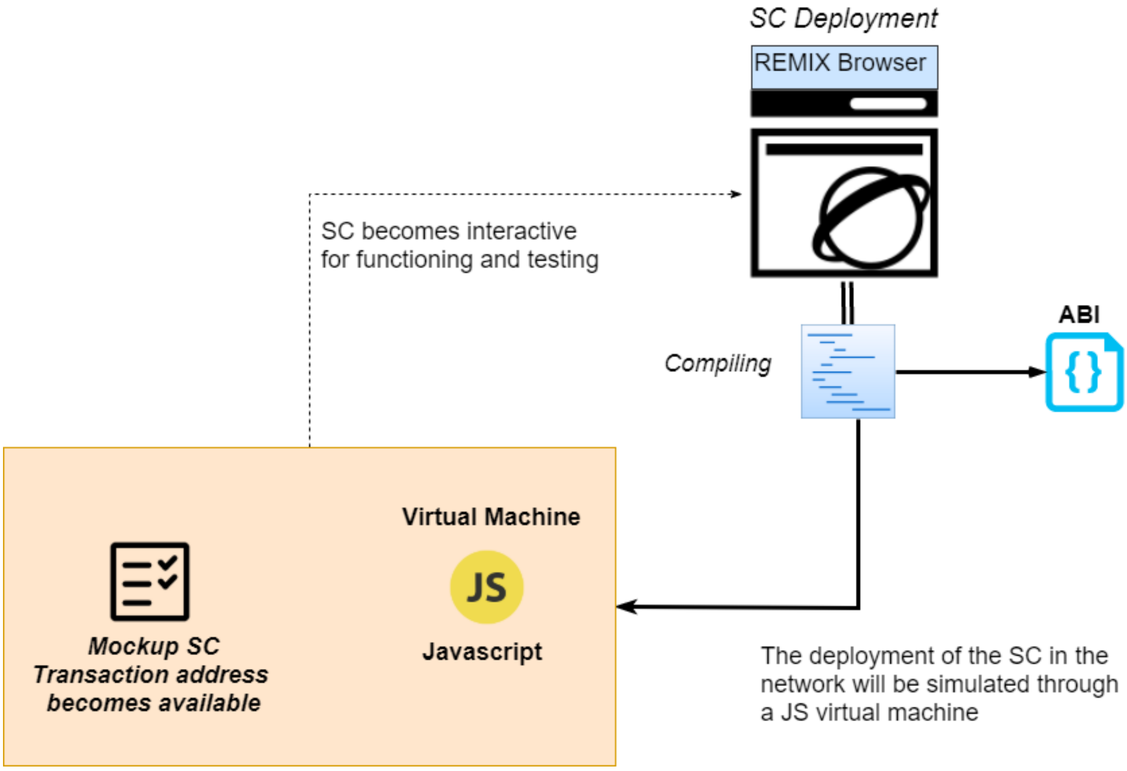
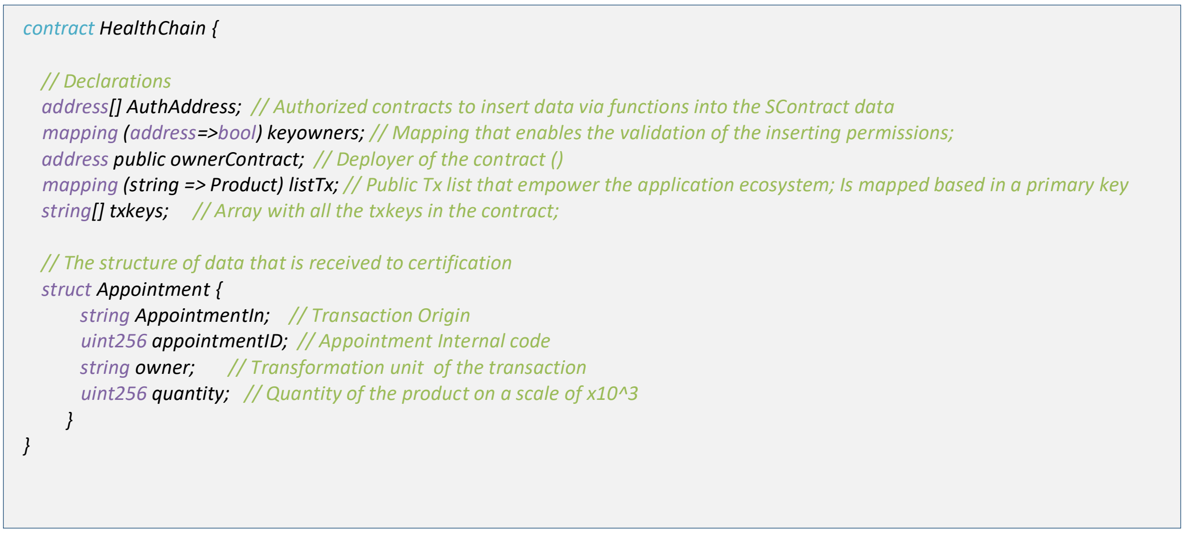


Figure 27 - Functional Flow of Dploying a Contract into a JS Virtual Machine using Remix IDE

1. The Solidity code is written and is compiled by the Remix browser IDE;
2. The Remix IDE will deploy the SC into a mock block (registered in cache memory);
3. The JS Virtual Machine will provide an address hash of the transaction into the Remix console and will abilitate an interaction menu.

below is a snippet example of code in Remix, with a contract named HealthChain and an object named "Appointment":



```
contract HealthChain {

  // Declarations
  address[] AuthAddress;  // Authorized contracts to insert data via functions into the SContract data
  mapping (address=>bool) keyowners; // Mapping that enables the validation of the inserting permissions;
  address public ownerContract;  // Deployer of the contract ()
  mapping (string => Product) listTx; // Public Tx list that empower the application ecosystem; Is mapped based in a primary key
  string[] txkeys;     // Array with all the txkeys in the contract;

  // The structure of data that is received to certification
  struct Appointment {
        string AppointmentIn;    // Transaction Origin
        uint256 appointmentID;  // Appointment Internal code
        string owner;      // Transformation unit  of the transaction
        uint256 quantity;   // Quantity of the product on a scale of x10^3
      }
}
```

Figure 28 - Smart Contract Snippet Example

5.4.2.3 Integration Environment

Once the contract is deployed, the application can benefit of transactions and transactions management if the code is pointing to the address of the smart contract in the blockchain. There is a difference between setting and creating transactions through functions and consulting data from the blockchain without state change.

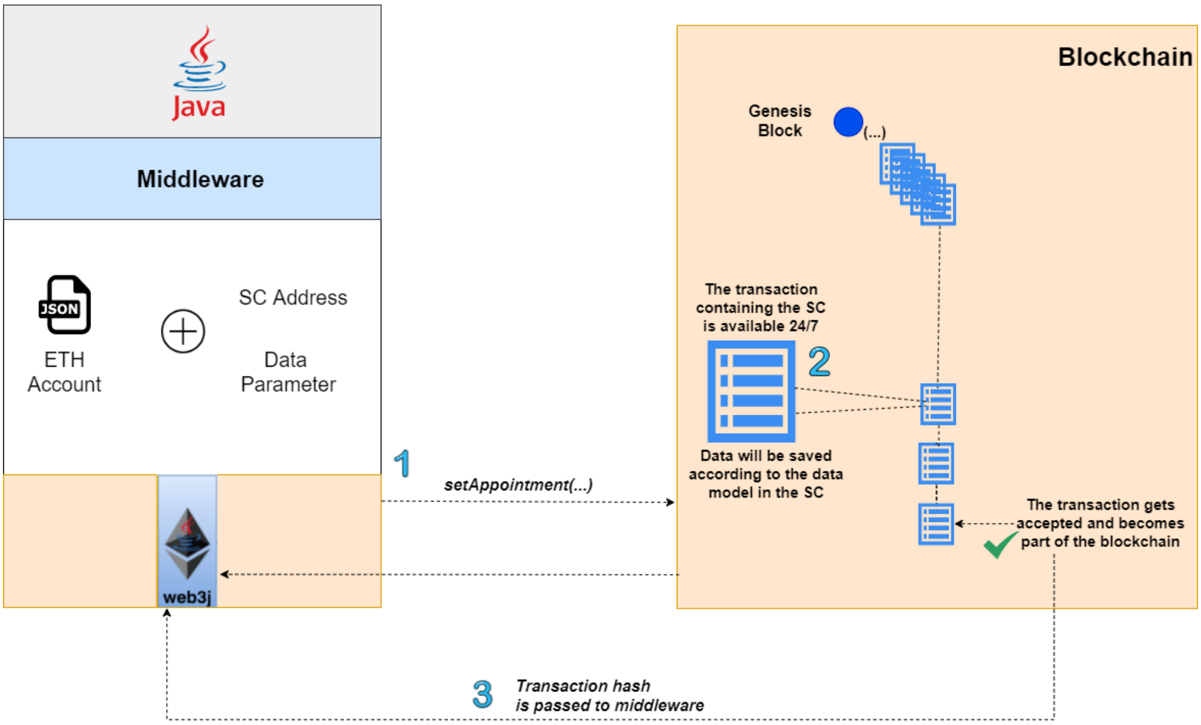
Set data/Change state

If a function is invoked and data is recorded or the state of the contract is changed, the transaction that will trigger the state change will be set to point to the SC but the information regarding the transaction will be held in a new transaction inside a new block in the network. In this case, a new transaction will be created with the input data in the "Data" field existent in Ethereum transactions.

1. The function is called from the Middleware using Web3J and sent to Geth. The middleware is able to sign the message using an Ethereum account that is authenticated in the network and this can be managed by means of middleware code
2. If the transaction is fit for purpose to interact with the SC then the transaction record will be saved in a block in the chain and submitted into

the consensus mechanism. The transaction can be denied due to parameters (for example). If so, changes in the SC will be reversed and the transaction will fail;

3. The transaction hash originated from the registering in the blockchain is passed back to the middleware.



Java
Middleware
JSON
ETH Account
SC Address
Data Parameter
web3j
1
setAppointment(...)
Blockchain
Genesis Block
(...)
The transaction containing the SC is available 24/7
2
Data will be saved according to the data model in the SC
The transaction gets accepted and becomes part of the blockchain
3
Transaction hash is passed to middleware

Figure 29 - Representation of Persisting Data using a SC

View functions/Consult data

Consulting data is enabled in the blockchain for the user. These view functions can be seen as calls and can be used by the user with no associated costs regarding blockchain's transaction model concept.

In Figure 30, the call for a function *getTransaction* – which retrieves a specific transaction, with the parameters TransactionID, which is the transaction we want to read, is demonstrated.

1. The function is called from the middleware using Java and Web3J to interact with Geth;
2. If the function called is perceivable and readable by the SC, it will read the data and run Solidity code to retrieve it from the blockchain;

3. Data is passed from the blockchain to the middleware using Web3J services to retrieve it.

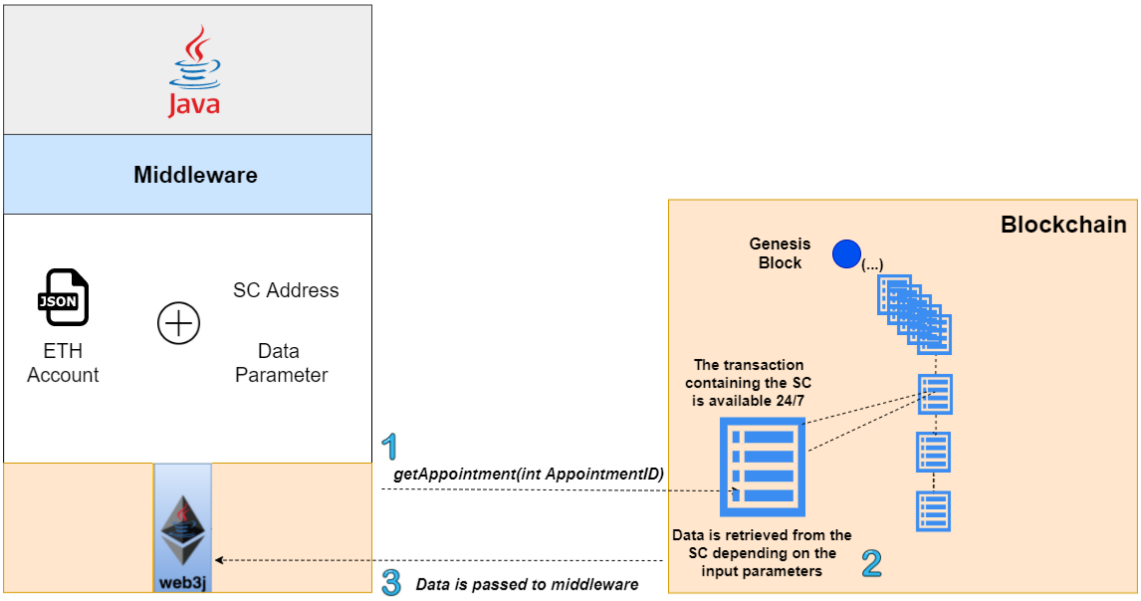

Java
Middleware
JSON
ETH Account
SC Address
Data Parameter
web3j
1
getAppointment(int AppointmentID)
3
Data is passed to middleware
Blockchain
Genesis Block
(...)
The transaction containing the SC is available 24/7
Data is retrieved from the SC depending on the input parameters
2

Figure 30 - Representation of Retrieving Data using a SC

6 Interviews

To perform the interviews, 10 Blockchain and GDPR professionals were interviewed. The LinkedIn application was chosen to reach out these individuals. Overall, 17 invites were made to the potential interviewees and out of those 17, 13 accepted. However, only 10 effectively responded to the interview.

For this research, only candidates with one year of experience or more were considered and technical backgrounds were favoured. The result can be seen in Table 14

Table 14 - Information Regarding Interviews

ID	Role	Age	Experience (Years) Blockchain/GDPR	Industry	PoC Approval
1	GDPR Project Manager	41	2 (GDPR)	Software development	✓
2	Blockchain Developer	31	4 (Blockchain)	Blockchain Solutions	✓
3	Information Security and GDPR Project Manager	29	3 (GDPR)	Software development	✓
4	Blockchain Lead Engineer	37	3 (Blockchain)	Blockchain Solutions	✓
5	Blockchain Developer	28	2 (Blockchain)	Blockchain Solutions	✓
6	Blockchain and Solutions	40	4 (Blockchain)	Blockchain Solutions	✓
7	Information Security Specialist	35	3 (GDPR/Data	Software Development	✓
8	Blockchain Developer	21	1 (Blockchain)	Software development	✓

9	Blockchain Developer	26	2 (Blockchain)	Software Development	✓
10	GDPR Data Protection Officer	36	3 (GDPR)	Data Protection	✓

Although Blockchain was originated in 2008 with Bitcoin, mainstream adoption is considered recent and efforts in developing blockchain solutions are somewhat recent in the IT Industry. For GDPR, it took effect in May 2018 and it is still a sensitive subject for companies, where some have taken measures to be compliant with while others are in violation of the regulation, sometimes by disregard.

The average age of the 10 professionals interviewed was of 32 years old. Related to the blockchain industry and solutions, the average age was of 30.5. Regarding GDPR/Data Security, the average age was of 35.25. Out of the 10 interviewees, all of them worked in the IT Sector.

The same interviewer carried out the 10 interviews and assuring the same interview script and methods were applied and used throughout the interviews. The first, ninth and tenth interview were carried out in the interviewee’s workplace, while the remaining were carried out by Skype. The interview was conducted using a script with questions written out beforehand with semi-structured open-ended questions and the objective was showing them the PoC and questioning them about the feasibility and if it was indeed possible to execute it and obtain results of compliance between blockchain and GDPR.

The interviews were all carried out between August and September 2019.

6.1 Evaluation of Results

Out of all the interviewed professionals, and due to the nature of the two different sectors that are blockchain and data protection and the various degrees of technicality that separate them, this dissertation focused on a larger number of interviewees with technical background (7) than data protection officers or other experts of data management, due to the technical nature of the PoC.

The general consensus of the interviewees is that the PoC is feasible and that it could solve the immutability issue that prevents blockchain technology as-is from being compliant with GDPR.

From the interviewees that carried out roles associated with GDPR, none of them had worked with blockchain technologies yet, so their answers were unfortunately dismissive when it came to answer questions regarding the PoC's feasibility from a purely technical perspective. From the three interviewees with roles associated with GDPR, all three agreed that the PoC would create an environment of compliance with GDPR. Interviewee #1 referred that she had heard of the problem of blockchain's features and the infraction of GDPR and said that "regarding GDPR, many companies are still struggling with the 'right to be forgotten' because the majority of the systems are not ready to erase personal data with such ease, which means that any attempt to facilitate that process is always welcome".

From the more technical side, out of the seven interviewees, all seven approved the PoC and its feasibility. Once again, this group of professionals were somewhat more focused on the technical aspect, which is accordingly to their roles and was what they focused more. The general consensus of the interviewees was that the PoC suggested an innovative solution for the compliance problem that was also contrasted by most of them, since the chameleon hashing algorithm would remove the immutability feature of the blockchain, considered by most of these professionals as fundamental. It also originates an aspect of centralization that was not otherwise there due to the fact that there has to be someone responsible for the editing and/or removal of transactional data of the blockchain. These interviewees were then assured by the interviewer that the PoC would have a private, enterprise-driven nature and that, in this case, absolute immutability is not always the way to go. The decreased centralization was also a necessary evil, since it is a must for this solution to work, although there are ways to reduce the hassle of this measure (for example, splitting the private key for the collision finding feature in chameleon hashing which is responsible for enabling the editing the data in various parts and splitting it through more than one person. The editing action of the transactions would then need cooperation and agreement among all parties.)

While all of the technical field professionals agreed that the PoC would effectively be feasible from a technical standpoint and that it would turn out to be compliant with GDPR, some interviewees also made some remarks regarding the performance and stability of the solution. Interviewee #2 said that "Chameleon hashing is a very recent solution. Although I have never been involved, altering the original Ethereum's algorithm hashing function and implementing chameleon hashing would be difficult but doable. There are also better technologies to use other than the Geth client, which is not as supported as other technologies". Interviewee #9 also referred to the Geth client as "not the most adaptable setup of the Ethereum technology" and that it would most likely suffer from performance problems. Interviewee #8 also referred to this problem, saying "…using Geth would need a vast number of software libraries and plugins to create it correspondingly to the PoC. There are better options for this purpose, like Hyperledger or maybe Corda"

These two suggestions are implementations of the Ethereum blockchain with the intent of creating private business-to-business adaptations of this technology.

The consensus among these seven technical individuals was that the PoC was achievable and that it would be attainable to make it work and function, but not all of them agreed on its praticable, realistic real-world value on the workaround solution that the PoC provided as a whole. This was mostly due to the poor choice of technologies (#8 and #9) and due to the loss of immutability perceived by the interviewees after studying the PoC, which was the main drawback of this whole implementation from the perspective of these seven interviewed individuals.

7 Conclusions

After the elaboration of this dissertation, some conclusions can be drawn. In this research, a total of 10 interviews were done to professionals regarding the industry fields concerning the dissertation's subject. All of them have approved the PoC and some results could be obtained:

- The PoC and its features designed in this dissertation can be developed as a practical, real-world application and are a valid solution to the constraining issue of Blockchain's immutability and the GDPR, more specifically the conflicting Article 17 (right to be forgotten), which requires personal data to be removed at order by the owner of the personal data.

- The technology used to elaborate the PoC, namely the Ethereum client Geth, was not the most fitting to develop an application like this. This is due to the fact that it is somewhat limiting and is mostly apt for testing networks and other non-complex purposes. Even though it would achieve the desired results, it is still a command line tool, which has its certain limitations and constraints implied. Other solutions like Hyperledger or Corda are more apt and have been released with the purpose of developing enterprise solutions and are more customizable and supported by blockchain technologies.
- It is indeed possible to reach a stage of compliance between Blockchain and GDPR. This is mostly made possible due to the implementation of chameleon hashing functions instead of the standard Ethash used in the Ethereum technology, which possibilitate the private-key holder of a specific chameleon hash function implementation to find collisions (same hash results) from different inputs, which results in the capability of altering/tampering with Blockchain transactions and preserving the transaction's hash. The drawbacks learned from research for this solution are the loss of immutability and the reduced decentralization.
- The previous points, the dissertation and the research made were able to answer the research question and confirm the hypothesis that it is indeed possible to conciliate Blockchain and GDPR, which was the target of this whole research. Furthermore, this dissertation is a contribution to the existing collective knowledge and lays down new bases for further research, as it was able to test certain scenarios and refining the systematized approach to blockchain solutions. This has not only implications for existing scientific knowledge, but also being useful to the whole Blockchain community and Blockchain solutions developers as well.
- The author, based on the dissertation's research, has produced a paper, submitted and accepted that was subject to a presentation in the 15th China-Europe International Symposium on Software Engineering Education, which focused on "Innovation on Research, Technology and Applications" topic.

7.1 Limitations

Regarding limitations, it was not possible to gather enough desirable information and conclusive data on certain subjects and topics, such as Chameleon Hashing or Ethereum’s hash function Ethash, since Blockchain is a recent subject and these topics have not been approached enough by researchers. The current research could not fully avoid bias since it has excluded literature sources that have been elaborated and written in different languages or that were inaccessible digitally.

7.2 Future Work

Regarding future work, research should be carried to other blockchain technologies other than Ethereum. Different technologies could still have the same solution but their specific characteristics could need a different approach.

It would be interesting to see the same approach of chameleon hashing applied to different contexts, as well as deeply exploring this hashing algorithm’s characteristics.

Bibliography

Aste, T., Tasca, P., & Centre, U. C. L. (2017). Blockchain Technologies: The Foreseeable Impact on Society and Industry.

Bach, L. M., Mihaljevic, B., & Zagar, M. (2018). Comparative Analysis of Blockchain Consensus Algorithms. 1545–1550.

Christidis, K., & Devetsikiotis, M. (2016). Blockchains and Smart Contracts for the Internet of Things. IEEE Access, 4, 2292–2303. https://doi.org/10.1109/ACCESS.2016.2566339

Christidis, K., & Member, G. S. (2016). Blockchains and Smart Contracts for the Internet of Things. 4.

Cosmina, I., Schaefer, C., Ho, C., & Harrop, R. (2017). Pro Spring 5 - An In-Depth Guide to the Spring Framework and Its Tools (Vol. 53). https://doi.org/10.1017/CBO9781107415324.004

Datoo, A., & Technology, D. L. (2018). Data in the post-GDPR world. Computer Fraud & Security Bulletin, 2018(9), 17–18. https://doi.org/10.1016/S1361-3723(18)30088-5

Ethereum Community. (2014). Ethereum Frontier Guide. https://doi.org/10.1002/ejoc.201200111

Finck, M. (2018). Blockchains and Data Protection in the European Union. 17–35. https://doi.org/10.21552/edpl/2018/1/6

Finck, M. (2019). Blockchain and the General Data Protection Regulation : can distributed ledgers be squared with European data protection law? : study.

Garber, J., & Focus, M. (2018). GDPR – compliance nightmare or business opportunity ? (June). https://doi.org/10.1016/S1361-3723(18)30055-1

Gatteschi, V., Lamberti, F., Demartini, C., Pranteda, C., & Santamaría, V. (2018). To Blockchain or Not to Blockchain : That Is the Question. (April), 62–74.

Güçlütürk, O. G. (2018). The DAO Hack Explained: Unfortunate Take-off of Smart Contracts. Retrieved May 17, 2019, from https://medium.com/@ogucluturk/the-dao-hack-explained-unfortunate-take-off-of-smart-contracts-2bd8c8db3562

Gupta, M. (2017). Blockchain for Dummies. John Wiley & Sons, Inc.

Ibáñez, L., Hara, K. O., & Simperl, E. (2018). On Blockchains and the General Data Protection Regulation. 1–13.

Janssen, T. (2019). What is Spring Data JPA? And why should you use it? Retrieved October 1, 2019, from https://thoughts-on-java.org/what-is-spring-data-jpa-and-why-should-you-use-it/

Karame, G., & Capkun, S. (2018). BLOCKCHAIN SECURITY AND PRIVACY Blockchain Security and Privacy. IEEE Security & Privacy, 16(August), 11–12. https://doi.org/10.1109/MSP.2018.3111241

Konheim, A. G. (2010). Hashing in Computer Science: Fifty Years of Slicing and Dicing. In Hashing in Computer Science: Fifty Years of Slicing and Dicing. https://doi.org/10.1002/9780470630617

Krawczyk, H., & Rabin, T. (1997). Chameleon Hashing and Signatures 1 Introduction. (October), 1–22.

Kshetri, N. (2017). Blockchain's roles in strengthening cybersecurity and protecting privacy. Telecommunications Policy, 41(10), 1027–1038. https://doi.org/10.1016/j.telpol.2017.09.003

Lakhani, K. R. (2017). The Truth About Blockchain. (February).

Lumb, R., Treat, D., & Jelf, O. (2016). Editing the Uneditable Blockchain: Why distributed ledger technology must adapt to an imperfect world.

Luu, L., Chu, D. H., Olickel, H., Saxena, P., & Hobor, A. (2016). Making smart contracts smarter. Proceedings of the ACM Conference on Computer and Communications Security, 24-28-Octo, 254–269. https://doi.org/10.1145/2976749.2978309

Matzutt, R. (2018). A Quantitative Analysis of the Impact of Arbitrary Blockchain Content on Bitcoin Roman. Lancet, 354(9172), 56. https://doi.org/10.1016/S0140-6736(05)75329-0

Miglicco, G. (2018). GDPR is here and it is time to get serious. Computer Fraud and Security, 2018(9), 9–12. https://doi.org/10.1016/S1361-3723(18)30085-X

Miller, D., & Laplante, P. A. (2018). Blockchain and the Internet of Things in the Industrial Sector. (June), 15–18.

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Www.Bitcoin.Org, 9. https://doi.org/10.1007/s10838-008-9062-0

O'Brien, T., Zyl, J. Van, Fox, B., Casey, J., Xu, J., Locher, T., & Moser, M. (2010). Maven : The Complete Reference The Complete Reference. (February), 318. https://doi.org/10.1036/0072133139

Politou, E., Michota, A., Alepis, E., Pocs, M., & Patsakis, C. (2018). Backups and the right to be forgotten in the GDPR: An uneasy relationship. Computer Law and Security Review, 34(6), 1247–1257. https://doi.org/10.1016/j.clsr.2018.08.006

Reuters, T. (2018). Technology Regulatory. The Journal Litigation, (March), 34–44.

Reyna, A., Martín, C., Chen, J., Soler, E., & Díaz, M. (2018). On blockchain and its integration with IoT. Challenges and opportunities. Future Generation Computer Systems, 88(2018), 173–190. https://doi.org/10.1016/j.future.2018.05.046

Schalk, C. (2005). Introduction to Javaserver Faces - What is JSF? Retrieved June 19, 2019, from https://www.oracle.com/technetwork/topics/index-090910.html

Swan, M. (2015). Blueprint for a new economy. In O'Reilly Media, Inc. https://doi.org/10.1017/CBO9781107415324.004

Tankard, C., & Pathways, D. (2016). What the GDPR means for businesses. Network Security, 2016(6), 5–8. https://doi.org/10.1016/S1353-4858(16)30056-3

Tappscot, D., & Tappscot, A. (2016). Blockchain Revolution : How the Technology Behind Bitcoin Is Changing Money , Business , and the World. 1–7.

Vukotic, A., & Goodwill, J. (2011). Apache Tomcat 7. In Apache Tomcat 7. https://doi.org/10.1007/978-1-4302-3724-2

Wang, L., Shen, X., Li, J., Shao, J., & Yang, Y. (2019). Cryptographic primitives in blockchains. Journal of Network and Computer Applications, 127, 43–58. https://doi.org/10.1016/j.jnca.2018.11.003

Wolford, B. (2019). Everything you need to know about the "Right to be forgotten." Retrieved September 21, 2019, from https://gdpr.eu/right-to-be-forgotten/

Wood, G. (2017). Ethereum: A Secure Decentralised Generalised Transaction Ledger.

2017, (August 1, 2017), 33. Retrieved from https://ethereum.github.io/yellowpaper/paper.pdf

Ye, C., Li, G., Cai, H., Gu, Y., & Fukuda, A. (2018). Analysis of security in blockchain: Case study in 51%-attack detecting. Proceedings - 2018 5th International Conference on Dependable Systems and Their Applications, DSA 2018, 15–24. https://doi.org/10.1109/DSA.2018.00015

Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2017). An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. Proceedings - 2017 IEEE 6th International Congress on Big Data, BigData Congress 2017, 557–564. https://doi.org/10.1109/BigDataCongress.2017.85

Finck, M. (2019). Blockchain and the General Data Protection Regulation : can distributed ledgers be squared with European data protection law? :