UNIVERSIDAD DE LA REPÚBLICA DEL URUGUAY

MASTER THESIS

Constructing privacy aware blockchain solutions: Design guidelines and threat analysis techniques

Author:
Ing. Fernanda MOLINA

Supervisors:
Dr. Gustavo BETARTE
Dr. Carlos LUNA

A thesis submitted in fulfillment of the requirements for the degree of Magister en Informática

December 18, 2021

Declaration of Authorship

I, Ing. Fernanda MOLINA, declare that this thesis titled, "Constructing privacy aware blockchain solutions:
Design guidelines and threat analysis techniques" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

UNIVERSIDAD DE LA REPÚBLICA DEL URUGUAY

Abstract

Facultad de Ingeniería
Instituto de Computación

Magister en Informática

**Constructing privacy aware blockchain solutions:
Design guidelines and threat analysis techniques**

by Ing. Fernanda MOLINA

Blockchain is an incipient technology that offers many strengths compared to traditional systems, such as decentralization, transparency and traceability. However, if the technology is to be used for processing personal data, complementary mechanisms must be identified that provide support for building systems that meet security and data protection requirements. In this work we study the integration of off-chain capabilities in blockchain-based solutions, moving data or computational operations outside the core blockchain network. Additionally, we develop a thorough analysis of the European and Uruguayan data protection regulation and discuss the weaknesses and strengths, regarding the security and privacy requirements established by that regulation, of solutions built using blockchain technology. Based on this analysis, we present a system architecture for the design of privacy aware solutions that are built using blockchain technology. We also put forward a systematic approach for performing a security and privacy threat analysis of such kind of solutions. Finally, we illustrate the use of the proposed methodological tools, presenting and discussing both the design and the security and privacy assessment of a system that provides services to handle, store and validate digital academic certificates.

Keywords: Blockchain, Off-chain, GDPR, personal data protection laws, design principles, security and privacy, threat analysis.

UNIVERSIDAD DE LA REPÚBLICA DEL URUGUAY

Resumen

Facultad de Ingeniería
Instituto de Computación

Magister en Informática

**Constructing privacy aware blockchain solutions:
Design guidelines and threat analysis techniques**

por Ing. Fernanda MOLINA

Blockchain es una tecnología incipiente que ofrece muchas fortalezas en comparación con los sistemas tradicionales, como la descentralización, la transparencia y la trazabilidad. Sin embargo, si se va a utilizar esta tecnología para el procesamiento de datos personales, se deben identificar mecanismos complementarios que brinden soporte a los sistemas de construcción que cumplan con los requisitos de seguridad y protección de datos. En este trabajo estudiamos la integración de capacidades de soluciones off-chain en soluciones basadas en blockchain, moviendo datos u operaciones computacionales fuera de blockchain. Adicionalmente, desarrollamos un análisis exhaustivo del reglamento europeo y uruguayo de protección de datos personales y discutimos las debilidades y fortalezas, en cuanto a los requisitos de seguridad y privacidad que establece dicho reglamento, de las soluciones construidas con tecnología blockchain. En base a este análisis, presentamos un marco metodológico para el diseño de soluciones basadas en tecnología blockchain, pensando en la privacidad. También presentamos un enfoque sistemático para realizar un análisis de amenazas a la seguridad y la privacidad de este tipo de soluciones. Finalmente, ilustramos el uso de las herramientas metodológicas propuestas, presentando y discutiendo tanto el diseño como la evaluación de seguridad y privacidad de un sistema que brinda servicios para manejar, almacenar y validar certificados académicos digitales.

Acknowledgements

I would like to thank the following people who have helped me with this research project: My supervisors, Dr. Gustavo Betarte and Dr. Carlos Luna, for their consistent support and guidance during the running of this project. To the Thesis committe, who took the time to analyze my work and carry out the corresponding evaluation. In particular I want to thank Dr. David García Rosado for having done the revision work of my thesis and to Dr. Gerardo Scneider for the thoughtful comments and recommendations on this work. Finally, I would like to thank my family for supporting me during the development of this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Blockchain is a disruptive and innovative technology [Nak09]: it provides support to build decentralized systems where transactions are processed by the participating nodes of the network without need of a responsible intermediary or authority. Blockchain offers many strengths compared to traditional systems, such as decentralization, transparency and traceability. On the other side, this technology has some general weaknesses concerning scalability and performance issues, but most importantly in our view, with confidentiality, immutability and access control.

In this work we elaborate on the approach that favors the incorporation of *off-chain* capabilities in blockchain-based solutions, moving data or computational operations outside the core blockchain network [ET17]. Current proposals of off-chain processes aim to leverage a blockchain solution by addressing some of the intrinsic functional weaknesses we have previously pointed out. Typical scenarios that are identified as requiring the use of off-chain solutions are those that, for their operational behavior, need to improve performance or cost calculation processing, to perform intermediate operations on the off-chain leaving the final transaction on the blockchain (off-chain Signatures Pattern [ET17]) or to perform a final complex calculation on the off-chain (challenge Response Pattern [ET17]).

In this work we have also carried out a thorough analysis of the GDPR [Eur16] regulation as well as the Uruguayan personal data protection regulation in order to determine the weaknesses and strengths of solutions built using blockchain technology regarding the security and privacy requirements established by those regulation. In particularly, we study the different possible architectures of blockchain and off-chain technology and their advantages and disadvantages regarding the protection of personal data regulations.

Of special interest was to try to give a solution to controversial subjects such as pseudonymisation and data anonymization when using hash functions and public key cryptography. In [Art14; LCT18; IOS18; FFM20; TGN20] several challenges the GDPR poses to solutions built using blockchain technology are presented and discussed. In those works it is analyzed, for instance, the processing of pseudoanonymized and anonymized data and the potential privacy violations that might take place from the use of hash values and private/secret keys in blockchain solutions.

One of the main objectives of our work has been to develop a system architecture that helps system designers to select operational off-chain constructs that integrated with traditional blockchain functionalities allow to build more secure and privacy aware solutions. In [LPX19], [Tru+19] and [Mol+18], different software architecture are proposed and discussed that have been conceived to make use of blockchain mechanisms to perform access control to data and auditing and to use off-chain solutions to safely storage and process personal data. We propose, additionally, to use the blockchain as an integrity network in order to validate data stored off-chain, by means of links to data stored on the off-chain network and the use of an integrity control check

mechanism. In the first phase of our investigation we have focused in developing two main constructs of the system architecture: a high-level system architecture model and a use case model. The use case model consists of several use cases that cover the principal services we understand can be used to build a blockchain and off-chain based system compliant with well established security and privacy requirements, in particular those established by the GDPR.

We also put forward a methodology to perform a security and privacy threat analysis of such kind of solutions. We illustrate the use of the proposed methodological tools presenting and discussing both the design and the security and privacy assessment of a system that provides services to handle, store and validate digital academic certificates. The problem was raised by the IT services unit of the Universidad de la República, Uruguay, SeCIU (Servicio Central de Informática Udelar).

1.1 Related work

There exist several proposals for off-chain solutions, the most common ones being either an external datastore, an external server or an external peer-to-peer network. In [Gud+19] two differente off-chain models are proposed. The *distributed or channels* model consist of a group of equal nodes in a peer-to-peer network outside the blockchain, which are organized by pre-defined rules. The *commit-chains or centralized* model consists of a centralized system which receives and processes user requests and periodically responds to the chain. The result of the procesing is transmitted to the blockchain, which in turn verifies the result before persisting it. In order to carry out that verification without disclosing confidential information the use of zero-knowledge test and verification processes like zk-SNARKs [Rei16] and zk-STARK [ET18] has been proposed.

An off-chain solution may also be conceived as a storage or external processing system or as an hybrid one [EH18]. In [ET17] different off-chain processing patterns are proposed.

Some of the challenges concerning data protection requirements the GDPR regulation poses to solutions built using blockchain technology are discussed in [Art14] and [LCT18]. In those works it is analyzed the processing of pseudonymised and anonymized data and the potential privacy violations that might occur from the use of hash values and private/secret keys in blockchain solutions. In [DE+21] is analyzed the use of smart contracts to manage the agreements between a data owner and a service provider. In [LPX19], [Tru+19] and [Mol+18], different software architecture are proposed and discussed that have been conceived to make use of blockchain mechanisms to perform access control to data and auditing and to use off-chain solutions to safely storage and process personal data.

The methodology for security and privacy threat analysis we present in this work makes use of concepts and procedures present in STRIDE [OWA20] and the CNIL [CNI12] methodology por privacy risk analysis.

As to the design of systems that store and manage digital educational certificates, there exist several solutions that have been proposed to deal with this problem, with different degrees of development. In particular, we have analyzed Latin American solutions like the Brazilian RAP System [al.18] and the Argentine System BFA [bfa20]. In [Say19] the solutions *Blockchain for Education* and *EduCTX* are discussed. All these systems propose a hybrid solution, using the blockchain to perform the validation of the certificates by storing a hash of the certificate and, in some cases, some additional information. Some of those solutions use public blockchain networks and others private

ones, but none of them implements access control mechanisms to perform the certificate verification process. In that process the candidate gives the employer his certificate (an act which is considered an implicit consent of personal data access) and the employer validates the provided certificate with the blockchain, typically comparing hash values. However, performing this verification implies access to personal data, since the hash of a personal data constitutes personal information, and therefore the execution of that procedure should be authorized by the data owner.

1.2 Contributions

The results of our work contribute to the growing body of research on blockchain technology putting forward methodological tools that place emphasis on security and privacy issues.

In the first place we explore and analyze in detail the state of the art of blockchain and off-chain technologies. We provide a detailed explanation of different proposals for blockchain and off-chain solutions, as well as guidelines to select off-chain models and their corresponding architectures and technologies according to the problem to be addressed.

We discuss the challenges the European General Data Protection Regulation (GDPR) requirements pose to the design and implementation of information systems that manage personal data and are built using blockchain technology. Additionally, we analyze Uruguayan personal data protection laws and their relationship with the GDPR.

Based on this analisys, we categorize GDPR-complying design bases to use these technologies and present a system architecture for the design of privacy-aware information systems that make use of blockchain technology. To the best of our knowledge, the design of the system architecture we propose is a first attempt in providing formal support to the design of privacy-aware information systems that use blockchain technology.

We also put forward a methodology for performing security and privacy threat analysis of systems of that kind. We illustrate the use of the design of the system architecture and the security and privacy threat methodology on a realistic and not trivial digital certificates system. We believe that the proposed threat analysis methodology is in itself a contribution that can be used in the analysis of systems that manipulate personal data. Regarding the risk analysis itself, we do not know of analysis of that kind, except the one of fraudulent activities to make fake diploma described in [Say19], being performed over systems that manage digital certificates.

The specification of the digital certificates system we present addresses privacy and GDPR compliance aspects that have not been considered elsewhere. For example, in our design the verification of certificates is mediated by an access control mechanism, in contrast to other proposals, where this function is enabled without such type of control. This type of solution, which allows the validation of university degrees, is of special interest of the central IT services unit of our University. Furthermore, it becomes of special importance in the presence of the national and international frauds related to the subject that have recently been reported in[Cer].

As a result of our investigation we have presented and published two papers in two international conferences. We present in [MBL21a] a condensed formulation of the GDPR-complying analysis, the integration of off-chain capabilities in blockchain-based solutions and a summary of the system architecture described on this work. In the work [MBL21b] in turn we present in detail the system architecture for the design of

privacy aware solutions that are built using blockchain technology and the methodology developed for security and privacy threat analysis. We also illustrate the use of those methodological tools in the design and analysis of a specification of the digital academics certificate system. We have also made available a preprint paper that provides a self-contained description of the analysis and design of the certificates system [MBL20].

1.3 Thesis organization

The rest of this thesis is structured as follows. Chapter 2 provides a primer on blockchain an off-chain concepts adn technology. In Chapter 3 we analyze the European and Uruguayan data protection regulation and discuss how well blockchain and off-chain mechanisms adapt to provide support for building GDPR-compliant digital systems. In Chapter 4 we put forward a system architecture for hybrid blockchain and off-chain solutions, as well as a methodology for security and privacy threat analysis we have developed. Then, in Chapter 5 we provide the specification of the digital certificates system and discuss the results of the threat analysis carried out on that specification. We conclude and discuss further work in Chapter 6. Finally, in Appendix A, we present in detailed the result of the threat analysis.

Chapter 2

Background on blockchain and off-chain technologies

This Chapter presents and analyzes blockchain and off-chain technologies, as well as a compendium of the main security problems related to this technologies.

2.1 Blockchain concepts

Blockchain is a peer-to-peer system which builds a chain of blocks with no centralized authority. A blockchain network is composed of a set of transactions grouped in blocks. Each transaction is a unique cryptographically signed instruction that represents the valid passage from one state to another. A transaction can be a message or a code (called *smart contract*), and can include a payment for its execution.

Transactions are grouped and processed in blocks in order to make the system more efficient. Through a consensus protocol it is defined which node publishes the new block. Each time a new block is created, it is downloaded, processed and validated by all the nodes in the network. Once a block is validated, it is added to the chain and the result are distributed to all members of the network. Thus, during that process each node executes all transactions contained in the block. Being a decentralized system, the nodes of the network have a copy of the entire chain and are responsible for validating and processing the blocks.

The immutability of the chain is based on the fact that each block of the chain contains the hash of the head of the previous block. In this way a change in a block implies a change in the whole chain. Figure 2.1 shows how blockchain works for bitcoin [Nak09].

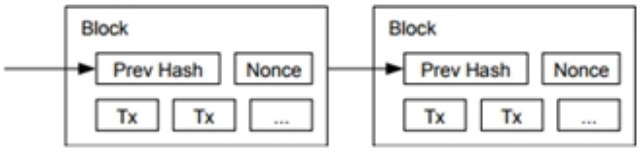

FIGURE 2.1: Blockchain node structure in bitcoin ([Nak09])

There are three types of blockchain networks: i) *networks that are used for specific cryptocurrency transactions*, such as Bitcoin, ii) *networks that handle cryptocurrencies and business logic*, such as Ethereum and finally iii) *networks that only handle business logic*, such as Hyperledger.

Regarding the access control of users, blockchain networks can be categorized into two types: i) *Permissionless,* the network is open to anyone willing to participate, so the level of distrust among the participants is high. Examples of these networks are Bitcoin [Nak09] and Ethereum; and ii) *Permissioned*, the system is a private network where the entry of new participants is controlled, such as, for instance, Hyperledger [Hyp20].

2.1.1 Consensus protocols

In blockchain all the nodes have the same information, the complete chain of nodes. To achieve this, the nodes must agree on the order and the way of updating the information. The way to do this on permissionless networks is by applying consensus protocols, which allow the network to define which node will add the next block to the chain.

Consensus protocols are designed to prevent what is called double spend, that is, spending the same amount of money more than once. This can happen when some consensus protocols are used that allow for multiple simultaneous chains to be generated over a period of time, until one is dropped [Cho17]. The integrity of the system is based on the honesty of most miners: a miner may try not to run a program or run it incorrectly, but honest miners will reject that block and fork the chain. Thus, by design, system security is based on the fact that no participant is able to control more than 50 percent of the network. This problem is discussed in more detail in the Section 2.6.3.

The protocols can be based on lottery systems (such as the Proof of Work system used by Bitcoin) or based on voting systems (such as the Redundant Byzantine Fault Tolerance system used by Hyperledger). In i) *lottery systems* the winner of the lottery proposes the block and transmits it to the rest of the nodes to be validated. These systems are based on a challenge that consists in generating a hash for the new block with certain criteria. The hash is created by using the header of the previous block and a nonce, and this can only be achieved by varying the nonces used. The first node that gets it is the one that will earn the right to add its block to the chain, and therefore earn a reward. The lottery systems are more scalable than the voting systems, but it has the disadvantage that is an inefficient system in terms of computing power consumption, and that there may be a fork if two winners propose a block at the same time. Each fork must be solved, which causes the completion of the process to take longer. In the case of a fork, the largest chain is the one considered valid, as shown in Figure 2.2. On the other hand, ii) *voting systems* have less latency (the time required to confirm a transaction that has been already included in the blockchain) than lottery systems, because when a majority of nodes validate a transaction or a block, there is consensus that the process ends. However, they are less scalable, since the voting system involves sending messages to the rest of the nodes, so the larger the network, the longer the process takes.

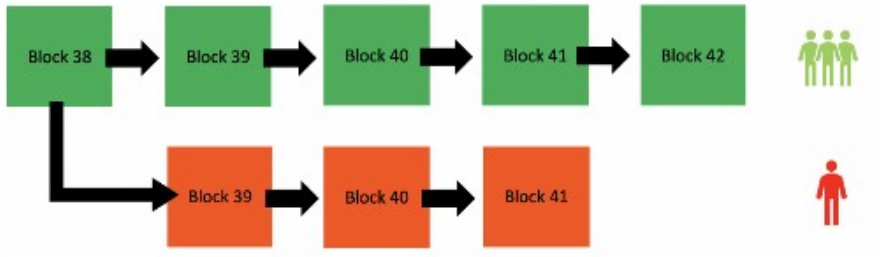

FIGURE 2.2: Chain creation - consensus protocol

2.1.2 Ethereum and smart contracts

Unlike Bitcoin, Ethereum [Woo17] supports contracts with state, where a value can persist in the blockchain to be used in multiple invocations. The handling of states allows

the use of Ethereum to process programs (*smart contracts*) and not only transfer cryptocurrencies. A smart contract is therefore a program that is stored and executed in a blockchain without someone from the outside being able to interfere with its operation. For this reason, by design a smart contract is executed in a distributed way through consensus protocols and each node executes all the programs that are executed on the platform. Using smarts contracts it is possible to build a system where agreements are forced to be reached autonomously, since it is an algorithm that forces compliance.

The support of this network is the Ethereum Virtual Machine (EVM), a network of independent machines, in constant communication. The EVM executes a code that is completely deterministic, that is to say, the same algorithms with the same inputs will always yield the same results.

Ethereum pays the miners in *Ether* for the computational resource used. In order to do this, each Ethereum instruction has a predefined amount of *Gas*, a unit of work used to measure how computationally expensive an Ethereum operation will be. Because a smart contract can be arbitrarily complex, a brief fragment of instructions could generate a lot of computational work, while a long fragment could generate less. This is why the rates in the EVM are based on the amount of work that is done, not on the size of the transaction. The price of Gas in Ether is proposed by the one who creates and sends the transaction, taking that into account, the higher the price, the more likely it is that a miner wants to introduce it in the next block. So, when a user sends a transaction, he must specify how much Gas he can provide for the execution (gasLimit) and the price for each unit of gas (gasPrice). A miner who includes the transaction in his block receives the payment corresponding to the Gas required to execute the code, multiplied by gasPrice. If any execution requires more Gas than gasLimit, the execution ends with an exception, returning to the state before executing the code. In this case, the one who sent the transaction must pay all the gasLimit to the miner. If little value is offered, the nodes will not execute the transaction, but a high value will not make the transaction run faster.

A transaction to be executed needs to goes through some validations, like review if the transaction is well formed, if transaction signature is valid, the gas limit is not less than the gas to be used to execute the transaction and if the account from which the transaction is sent (sender) has at least the cost required to pay for the execution of the transaction

2.2 Off-chain concepts

Off-chain transactions refer to those transactions occurring on a cryptocurrency network which move the value outside of the blockchain. The premises of blockchain are: transaction validation, consensus protocols and decentralization. These premises, however, generate all transactions to be validated, processed and stored in all the nodes of the network, which generates an overhead of work. Additionally, there is a cost to carrying out transactions, may be due to the computational cost of processing or the cost of network storage. Finally, the peer-to-peer system, where information is processed and stored on all nodes of the network, generates a confidentiality problem.

In summary, we find the following problems that can be solved with off-chain: i) scalability problems through throughput and transaction processing latency in blockchain; ii) cost problems derived from the payment made for transactions; iii) confidentiality problems, when it is required to process a calculation, therefore the information

must be exposed and iv) requirement to store information outside the blockchain, either because of confidentiality or because it is a large amount of information where a cost problem arises.

2.2.1 Off-chain processing

There exist several proposals for off-chain solutions, as explained in [EH18]. Storing large amounts of data could be expensive in blockchain, so a possible solution is to store the data outside the network, in a *off-chain storage*, leaving a pointer on the network to the location of the data. In this model, when processing a state change, the stored information is received from an external node, which communicates the change of state to the blockchain. Upon entering the new state, the information is stored again in the external node.

To ensure the integrity of the data stored externally, a verification step should be performed. An option to validate the data stored in the chain is to keep in the blockchain a hash value of the data that is externally stored. Additionally, in order to preserve confidentiality it is necessary to implement an external storage access control system. On the other hand, if the availability of the information stored is important, it is pertinent to think on schemes where the information is stored redundantly. The Interplanetary File System (IPFS) [IPF20], and SWARM [SWA20] systems help with access control and availability issues.

An alternative off-chain model is *off-chain computation*, where a part of the processing is performed on the off-chain. The result of the execution is transmitted to the blockchain, which in turn verifies the result before persisting it. This verification is very important to ensure that the result returned is correct, since external processing should in itself be considered unreliable.

Finally there are cases in which it may be necessary to use a *hybrid model*, using off-chain to store data and to perform computational process.

2.2.2 Off-chain architectures

In turn, we can find two types of architectures in off-chain. The *distributed or channels model* consist of a group of equal nodes in a peer-to-peer network outside the blockchain, which are organized by pre-defined rules, for example, through a smart contract. This system requires using a peer-to-peer protocol between participating nodes, such as the Whisper Messaging Protocol [Whi20].

One way of operating this model is described in the paper [Gud+19], where is described a system that works in three phases. In an initial phase of foundation, i) *establishment*, the nodes are joined paying a fund, which is agreed or predefined. Then there is a ii) *transition* state, where a node proposes a new state by sending it along with its signature. The other nodes perform the calculation to verify that the proposed state is valid by executing the command sent. They send the signed result to the other nodes as approval. During the process it can be iii) *disputes*, for example if a node does not receive a certain number of signatures of a proposed state after a pre-determined time out, it can assume that there was a dispute regarding the new state. This node can then send the controversy to the blockchain, to force a new state over the off-chain.

In general these systems requires an unanimous consensus of the participants, since only transactions that are approved and signed by all nodes are considered valid. A node can at any time dump the approved off-chain calculations into the blockchain. In the same way, if there is a dispute about the outcome of the off-chain, such as a

dishonest participant trying to lie about it, honest participants can resolve the dispute on the blockchain.

Two possible cases of use of channels, described in [ET17] are i) *Payment Channel*, where two participants want to keep intermediate transactions in private, and only publish to the blockchain the final state of these movements. It is possible to specify a smart contract that uses an external state as an argument and that verifies the signature of the participants as acceptance of the new status. It usually starts with an initial amount and a new state is reached only if both participants accept the new state by signing it. The transactions take place off chain and peer-to-peer, until it is decided to send a transaction status to the blockchain. Another cases of use is ii) *Challenge-response system*, that is used when only is required to improve the processing of a final calculation. For example in chess, calculate that a move is check mate is very expensive, so it is more efficient for the game that the player who considers that made check mate declares it and wait for another player to prove him wrong by submitting a valid move. On the other side, the signatures pattern seeks to reduce the cost of intermediate transactions. In this pattern the nodes carry out transactions with each other outside the chain, leaving the final state in the blockchain.

A potential risk in distributed or channels model is the retention of funds, when an initial fund is required to enter to the system and a malicious actor blocks them, simply by not accepting the signature of the other participant. In addition, if the objective of the use of off-chain is to protect the confidentiality of the information, this is compromised in a dispute, given that it is resolved in the blockhain, where the information is seen by all the nodes.

On the other sidem the *centralized or commit-chains model* consists of a centralized and not necessary realiable system which receives and processes user requests and periodically responds to the chain. This system performs the processing and returns the result along with a proof that it is correct, which is verified by the blockchain, using mechanisms of zero-knowledge test and verification process, as is explained in Section 2.3. The Figure 2.3 explains this architecture.

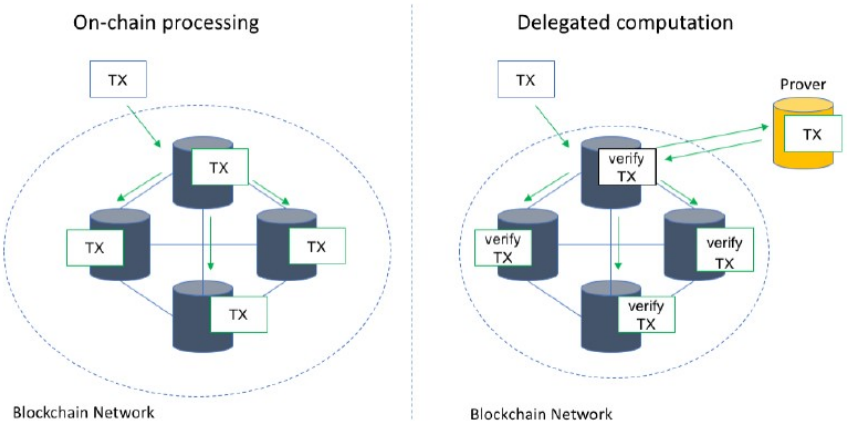

FIGURE 2.3: Off-chain: Delegated computation [EH18]

As it is a centralized system, an availability problem is added, which does not occur in the case of the channels model. Another difference of the channels model is that the establishment and state transition is free, since it is a communication between a node and the external entity.

2.3 Test and verification processes

As previously seen, testing and verification processes are necessary in a centralized off-chain architecture, where the third part that performs the processing and provides the result to the chain is not trusted. This type of verification is used by a third party to demonstrate that they have certain information or meet a certain requirement. In short, it consist of a party (the prover) proving to another (the verifier) that a statement is true, without revealing any information beyond the validity of the statement itself. For example, given the hash of a random number, the prover could convince the verifier that there is indeed a number with this hash value, without revealing what it is. This property is important for confidentiality and privacy.

The prover, who performs the calculation, must send a proof that will be verified by the verifier. For the test and verification process to be correct and efficient, it must meet the requirements described bellow. First i) the test sent (by the prover) must be short and non-interactive, that is, the prover must be able to send the test in a single message. Adittionally, ii) the cost of verification must be independent of the computational cost of the problem to be solved. Ideally, the cost of verification should be less than the cost of solving the problem. Finally iii) the off-chain provider can use private information during the generation of the test, but the verifier will not learn or infer anything from the test obtained. This property is known as *zero-knowledge*.

Currently, the most efficient way to produce zero-knowledge, non-interactive tests that are short enough to be published on a blockchain, is to have an initial configuration phase that generates a common reference chain, shared by the prover and the verifier. For the system to work, it must be assumed that this common reference chain was honestly generated and that it can be trusted since, if a malicious user performs the initial configuration step, the verification process will be compromised.

In summary, these systems are based on the idea that the test and verification system is more efficient than the resolution of a complex calculation, since it should be much easier to verify than to perform the computation. Researchers have made great progress in this method in these recent years, however, the initial configuration costs and computational overhead for the cloud in the current state of technology make these methods not yet suitable for most of the real world applications.

2.3.1 Models of test and verification solutions

There are several models of test and verification solutions, which we proceed to describe in what follows.

zk-SNARKs (Zero-Knowledge Succinct Non-Interactive Argument of Knowledge) allows a person to generate zero-knowledge tests using an initial configuration stage. This model uses the concept of arithmetic circuits to generate a zero-knowledge test, through a series of stages described in [Rei16]. The zk-SNARK are considered computationally solid, which means that a dishonest participant has a very low probability of successfully cheating the system. This property is fulfilled assuming that the prover has a limited calculation capacity. Theoretically, a prover with sufficient computing power could create false evidence, and this is one of the reasons why quantum computers are considered a threat to zk-SNARK and blockchain systems.

There are other models that do not requires a zero stage configuration, such as *zk-STARKs* (Zero-Knowledge Scalable Transparent Arguments of Knowledge) [ET18]. The zk-STARKs was created as a faster and cheaper alternative version of the zk-SNARK tests and, more importantly, zk-STARKs do not require an initial trust configuration,

because it is based on simpler symmetric cryptography through hash-resistant collision functions. This approach also eliminates the theoretical numerical assumptions of zk-SNARK that are computationally expensive and theoretically prone to attack by quantum computers. ZK-STARKs is also more scalable in terms of speed and computational size compared to ZK-SNARKs.

2.3.2 Zero-knowledge implementations

Some implementations of these solutions are i) *ZoKrates* [ET18] is a set of free software developments that can solve the verification process, from the generation of tests to their verification, using zk-SNARKs as a test system. It allows, through a high level language, to generate off-chain code and allows to export smart contract verification, which verify the tests generated outside the chain in a blockchain, such as Ethereum. ii) *Bulletproofs* [SU18] is a zero-knowledge system that, like zk-STARKs, does not require an initial configuration phase. It is intended for small tests, since the verification process is more expensive than SNARKs. It is designed to allow efficient confidential operations in Bitcoin and other cryptocurrencies where confidential transactions hide the amount of money transferred and contains cryptographic proof that the transaction is valid. iii) *TrueBit* [JT18] proposes a different alternative to the test and verification method. Instead of providing proof of the results, as ZoKrates does, the result is optimistically accepted and the nodes are financially encouraged to provide proof of fraud in case of invalid results. In the absence of a dispute, the party performing a computational task on behalf of a TrueBit contract receives a reward. Dispute resolution occurs as a "game verification" subroutine in TrueBit, using principles of game theory.

2.4 Smart contract design using blockchain and off-chain

Beyond the chosen off-chain architecture, when executing a smart contract, it is necessary to choose whether it will run: on the off-chain, on the blockchain, or on a hybrid model. Related to this, the paper [LPX19] suggests to divide a smart contracts into two contracts: a contract that contains the processes of major computations and/or the logic that allows identifying private information of the participants, and another contract that contains the rest of the code, less expensive and/or less sensitive. Using this architecture, in a channel off-chain model, this paper suggests to divide the process into 4 stages:

1. *Separation and generation:* The first thing to do is classify the functions of a smart contract into two categories: heavy-private and light or public. By generating two smart contracts from the original smart contract, extra functions are created to process disputes and match both smart contracts.

2. *Implementation and signature:* Before starting the execution of the smart contract, a copy of the off-chain contract, signed by all the participants, is stored. This will put on the blockchain in case of a dispute.

3. *Execution:* During execution, if all participants are honest, they can perform the off-chain calculation themselves and manually send the results to the blockchain to force a change of status. A "challenge" period start, where a participant who disagrees with the outcome can process a dispute.

4. *Dispute resolution:* During the off-chain execution, in the case of a dispute generated by a non-honest participant, any honest participant can deploy the heavy or

private functions to the blockchain to be executed there. To prove that the functions to be deployed are the same ones that were approved unanimously at the beginning, the participant must put on the blockchain the copy of the smart contract off-chain signed by everyone in step 2, in which case this execution will be taken as the correct one. When this occurs, an extra function is invoked in order to verify the signed copy that is returned to the blockchain, an extra function is invoked to impose the result obtained and another to penalize the participant who gave an incorrect result. After verifying the integrity of the deployed heavy or private functions, it is necessary to reconstruct the connection between these functions and the already deployed light or public functions. Therefore, the smart contracts that come from the same smart contract must be connected.

It is also possible to implement a similar model by using a centralized third part (commit-chains architecture) as off-chain. The paper [Mol+18] provides an implementation model for a payment system between nodes, as shown in Figure 2.4. In this model, a Gateway node is used to communicate with the off-chain (Trusted Third Parties TTP) and with a private repository, where confidential information is stored (D1, D2 and D3 represents personal information). The smart contract SCc, executed in the Trusted Third Parties (TTP), determines whether the operation is valid (cc) or not (ncc), and returns the result to the Gateway so that it decides, according to this result, whether access to the storage is provided or not. The payment operation is executed on the blockchain, with the smart contract SCd.

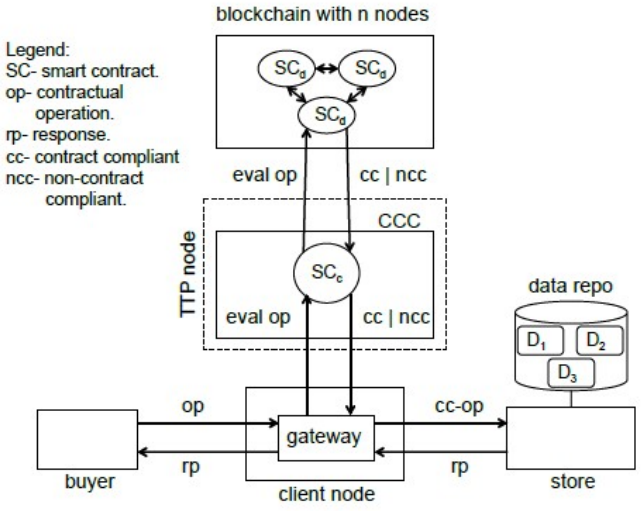

FIGURE 2.4: Off-chain: A hybrid architecture of smarts contracts [Mol+18]

2.5 Criteria for selecting off-chain models and architectures

As has been explained, there exist several proposals for off-chain solutions. In Table 2.1 we summarize those variants together with some guidelines to select off-chain models and their corresponding architectures and technologies according to the problem to be addressed.

Off-chain models and implementation					
Type of problem	**Off-chain model**	**Off-chain architecture**	**Method**	**Additional technology**	**Tools**
Improve calculation processing	off-chain computation	channels / commit-chain	Delegate the computation work to the off-chain	Non-interactive and short verification process	Whisper Msj. Protocol / zk-SNARKs / zk-STARKs
Improve processing of a final calculation	off-chain computation	channels	Challenge-response system	Not required	does not apply
Reduce calculation processing costs	off-chain computation	channels / commit-chain	Delegate the computation work to the off-chain	Cost verification process less than the problem to solve	zk-SNARKs / zk-STARKs
Ensure the cost of intermediate transactions	off-chain computation	channels	Payment channel	Peer-to-peer system	Whisper Msj. Protocol
Reduce storage costs	off-chain storage	channels / commit-chain	External storage systems	Cost verification process less than the problem to solve	zk-SNARKs / zk-STARKs / BD: DHT, DDB
Ensure the confidentiality of information required to perform a calculation	off-chain computation	commit-chain	Delegate the computation work to the off-chain	Verification process zero-knowledge	zk-SNARKs / zk-STARKs
Ensure the confidentiality of intermediate transactions	off-chain computation	channels	Payment channel	Peer-to-peer system outside the blockchain	Whisper Msj protocol
Ensure the confidentiality of the information to be stored	off-chain storage	commit-chain	External storage systems	hash verification integrity / External storage access control system	Interplanetary File System (IPFS), SWARM DHT, DDB
Compliance with GDPR regulation	hybrid model	Commit-chain	External storage systems / Delegate computing work to off-chain	Zero-knowledge verification process / Hash verification integrity / External storage access control system	Zk-SNARKs / zk-STARKs
Ensure the confidentiality and availability of information to be stored	off-chain storage	channels	External storage systems	Zero-knowledge verification process / Hash verification integrity / External storage access control Peer-to-peer system	Interplanetary File System / SWARM /DHT/, DDB

TABLE 2.1: Criteria for selecting off-chain models and architectures

2.6 Blockchain and off-chain issues

Finally, this section analyzes technological and security problems presented by these technologies.

2.6.1 Scalability and performance issues

Since all transactions must be validated and processed by all network nodes, blockchain systems have *scalability problems*. The scalability of blockchain depends on two factors: the throughput of transactions (maximum ratio at which the system can process transactions) and latency (the time required to confirm a transaction that has been already included in the blockchain). The latency in Bitcoin is at least 10 minutes and

throughput, a function between the block size (currently 1 MB) and latency, is 7 transactions per second. In Ethereum the throughput is 15 transactions per second. As an example, the VISA throughput is about 24,000 transactions per second and the latency is a few seconds. Bitcoin is discussing about adjusting its operation without changing its essence, to improve the scalability of the solution, for example, increasing the size of the block or improving the time between blocks. This is estimated to lead to a throughput of 27 transactions per second, which does not eliminate the escalation problem, it only mitigates it temporarily.

Another problem of scalability of this solution is related to the *cost of transactions*, which increases depending on the complexity of computation to be processed. Programmers must therefore consider the cost when designing and programming a Smart Contract.

2.6.2 Blockchain security issues

Blockchain security relies on public key cryptography for identifying transactions and hash technology to provide guarantees of the immutability of the chain: each block of the chain contains the hash value of the head of the previous block, therefore, a change in a block implies a change in the whole chain. This architecture ensures integrity, immutability and traceability by design. However it presents some security problems that are described below.

Confidentiality

Since all transactions must be validated and processed by all the nodes of the network, all the information necessary to perform this processing must be public, what undermines the confidentiality of the information. For example, if a customer wants to prove that he has the token of a provider that publishes the hash of the tokens he provides, the customer can generate the hash of his token and thereby prove that has one of the hashes published by the provider. However, to do this on the blockchain, he must perform the execution of the hash in the blockchain, so that all the nodes perform the execution, thus revealing their token.

Everything used in a smart contract is publicly visible, including local variables and status variables marked private. Defining something as private only prevents other contracts from accessing and modifying information, but this information will always be visible to everyone in the chain. Additionally, although the information can be stored in encrypted form, certain metadata is always available, through which information can be inferred (through pattern recognition, for example), such as the type of activity and the volume associated with the activity of any public address of the network.

Immutability

The impossibility of modifying smarts contracts implies a problem of immutable bugs, that is once a contract is published, it cannot be altered. This would be a problem if bugs are found in the code. Nevertheless, there are ways to use a new contract instead of using the original one, which is similar to "updating" a contract. One way to update the code is to create an intermediary smart contract that will keep the smart contract address active. Therefore, all calls and transactions will be redirected to the active version with the delegatecall function. That way, the same contract address will always

be used, but that contract can execute a different smart contract code at the end. In addition, as will be seen in Chapter 3, immutability threatens compliance with personal data protection regulations.

Key management risk

Blockchain is susceptible to theft of private keys and the control of the assets associated with external addresses being taken away. Digital assets could become unrecoverable in the case of theft of private keys, especially due to the lack of an administrator or system controller [SB17]. In the world of blockchain, the possession of keys and ownership of content are synonyms and the best way to obtain keys is to attack the weakest point in the chain, that is, personal or cellular computers. To mitigate this issue, it is essential to follow good practices in the use of wallet and key management.

Unpredictable state

When a user sends a transaction to the network to invoke a contract, he cannot be sure of the state in which the contract will be when the transaction is executed. This is because in the interim, other transactions could alter the status of the contract, since no order is guaranteed in the execution of the transactions. This happens specially in lottery systems (see Section 2.1.1).

Transaction security issues

A risk related to transactions is the risk of double spending, that is, spending the same money more than once. This can happen when some consensus protocols are used that allow for multiple simultaneous chains to be generated over a period of time, until one is dropped [Cho17]. Blockchain users protect themselves from this fraud by waiting for several confirmations when receiving payments, as transactions become more irreversible as the number of confirmations increases. Each time a new block is added to the chain, the verification is confirmed again. As a consensus, many users expect to have six confirmations before accepting a transaction as payment, to avoid the problem of double spending.

2.6.3 Consensus protocols security issues

Lottery systems (see Section 2.1.1) may imply that several nodes propose a block at the same time. Block duplication is solved because miners use the criterion of attaching new blocks to the longer chain, so shorter chains are discarded. The system works fine, as long as the network is not controlled by a coordinated group.

A known security problem of the permissionless blockchain refers to consensus protocols and what would happen if the network is dominated by a majority (51 percent attack). These consensus protocols require most miners to be honest, because it there is a group of miners working together who dominate more than 50 percent of the network, the network stops being decentralized and becomes controlled by a group. In such a situation, it is even possible to alter transactional records or generate double spend fraud (see Section 2.6.2). This has not happened so far but it is not impossible for this risk to materialize, taking into account that the miners merge into groups to join their effort. Figure 2.5 shows a graph referenced by Kaspersky in a report published in 2017 [Kas17] which shows the market share of Bitcoin mining pools. It shows that 4 mining groups control more than 50 percent of all computing power.

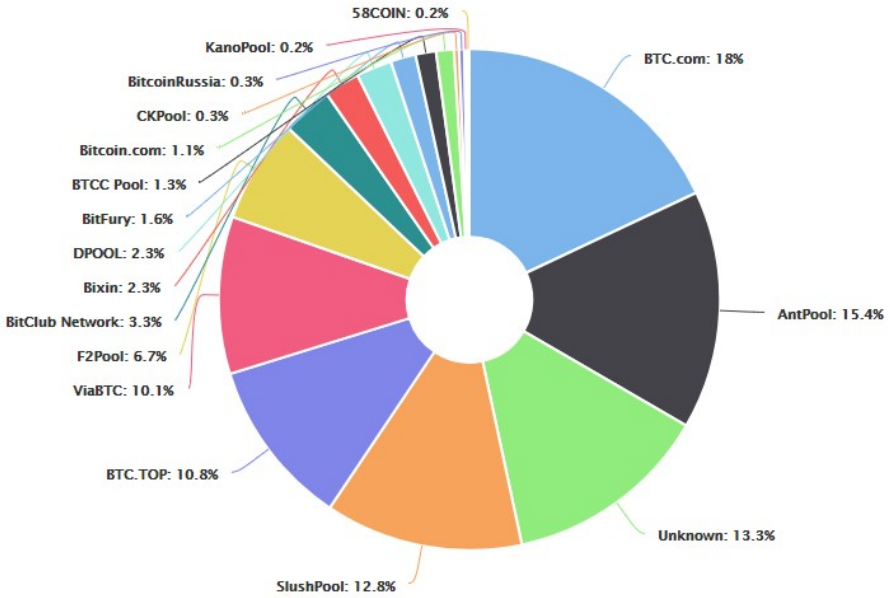

FIGURE 2.5: Market share of Bitcoin mining pools ([Kas17])

Additionally, the report "Majority Is Not Enough: Bitcoin Mining Is Vulnerable" [IE18] explains a method in which a group of non-honest miners work in coordination, in order to obtain proportionally higher incomes than honest miners: subsequently, the income of the non-honest group (called selfish-mines) increases linearly with the size of the group. The system used for this is based on forcing honest miners to spend their mining on blocks that are destined not to be part of the blockchain, since they are in chains that will be discarded. Because of this, the selfish miners work on a private branch of the chain, while honest miners work on the public branch. The selfish-mines make their chain public when it is longer than the public one, so honest miners leave the chain they were working on, throwing away their work. This large-scale operation can be combined with a Sybil attack ([Wik20]), where selfish-miners can use their power to invalidate transactions on the network, isolating honest nodes that will only receive information from non-honest nodes.

2.6.4 Smarts contract security issues

The paper [NA16] presents a study of the best known vulnerabilities of smart contracts in Ethereum, classifying them according to whether it is a vulnerability of the Solidity programming language, the EVM or the blockchain technology. In [Eth] known attacks on Ethereum smart contracts can be found; within these, one of the best known attacks is known as the DAO attack.

An important security risk related to smart contrats are the *oracles* (a participant external to the blockchain that provides data) that many smart contracts use to function. If these oracles were attacked by malicious agents, they could potentially corrupt the network by generating a chain effect across the entire network.

Another security problem is the difficulty of *smart contracts verification*, that is how to verify that a smart contract effectively does what it says it does and does not perform other operations or that its execution has no unwanted side effects. One way to mitigate this problem is to provide information or rely on standards that can give users confidence in the characteristics of the published smart contract. An example of standards to which a smart contract can adhere is Ethereum improvment proposal (EIP) [Eip],

that is a design document of a smart contract, that gives information to the Ethereum community or describes a new function for Ethereum, its processes or environment. The author of the EIP is responsible for generating consensus within the community and for documenting dissenting opinions. Another method to mitigate this problem is to look for mechanisms to verify the published code, so that it can be compared to the source code, either to make our contract more reliable or to rely on contracts created by third parties. There are initiatives that seek to verify that the published code matches the source code, such as Etherscan [Eth21]. The objective is to trust the contracts that have been verified, because it is possible to see the source code and verify that the published bycode matches the source code. Finally, there are initiatives that work to formally verify the Solidity source code, or the EVM compilation, to demonstrate that it has no programming vulnerabilities or to demonstrate that a contract complies with his high-level specification. A general problem with these initiatives is that most smart contracts do not publish their source code.

Solidity programming language also has security problems, like i) *call to the unknown*, which refers to the fact that some Solidity primitives may have the effect of invoking the fallback function that can be used for attacks. ii) *Gasless send*, that is problems arising from an "out-of-gas" exception when using the send function. There may be vulnerabilities arising from the mishandling of exceptions in Solidity, called iii) *exception disorders*. iv) *Type casts* are vulnerabilities derived from mishandling of some exceptions of type of data in Solidity. v) *Reentrancy*, also known as the "recursive call" vulnerability. The fallback mechanism can generate the recursive call of non-recursive functions. Another problem is vi) *keeping secrets*, since the blockchain is public, everyone can inspect the content of a transaction and infer the value of the fields, even if it is declared private. To ensure that a field remains secret, it is necessary to use cryptographic techniques.

2.6.5 Off-chain security issues

The use of off-chain to store information introduces problems already solved in blockchain.

Integrity issues

In blockchain solutions integrity is guaranteed by default, because by design it is impossible to modify data without being noticed by the rest of the network. However, when storing data outside the blockchain this feature is lost since a third party becomes responsible of handling the storage. Therefore, a verification process must be required to ensure the integrity of the stored information. One way to do this is to store in the blockchain a reference and a hash value of the information stored externally, so that the information can be corroborated in case of alteration. Delegating the computational process also causes problems of integrity of the result, since the calculation is not performed by all the nodes of the blockchain, but it is instead performed by an external one. In a commit-chains model one way to ensure the correctness of the result obtained outside the blockchain is through the application of zero-knowledge test and verification processes, as was described in Section 2.3. In channels models the dispute are resolved on the blockchain.

Availability Issues

In blockchain, the availability of data is guaranteed by storing the information in all the nodes of the network. When data is stored outside the blockchain this property is difficult to guarantee since there is a single point of failure. Thus, solutions such as IPFS [IPF20] and SWARM [SWA20] have been proposed, where information is stored in a decentralized and redundant manner.

Traceability issues

Off-chain solutions do not define by design audit processes, so auxiliary mechanisms must be considered in order to register an audit trail of the accesses and changes made.

Chapter 3

Blockchain constructs, data protection principles and requirements

The GDPR [Eur16] is the European regulation on the protection of natural persons with regard to the processing of personal data and the free circulation of these data. The GDPR entered into force on May 25, 2016 and came into effect on May 25, 2018. European countries had their own personal data protection laws, but with GDPR they became governed by a common legislation, which not only reaches companies or organizations resident in countries belonging to the European Union, but also foreign companies or organizations that deal with data of EU residents.

3.1 GDPR scope, roles and responsibilities

The GDPR defines in its *Article 4* that personal data includes all the data that is or can be assigned to a natural person, such as, for instance, the phone number, credit cards, account information, registration numbers, appearance, customer number or address. As discussed in [Eur19], the reference to an identifiable person indicates that it is not required the data to be identified as belonging to someone to qualify as personal data, but that the mere possibility of identification is sufficient. This concept is important in the context of blockchain solutions, where individuals can be identified through the use of public keys.

In its *Article 4* the GDPR also defines the roles that are responsible for the handling of personal data:

- *Data Controller*: it is responsible for processing of information and the appointment of the processor role (*Article 28*). A data controller can process the data collected using its own processes, or it can also work with a third party or an external service to process the data that has been collected. Even in this situation, the data controller will not transfer control of the data to the third-party service, as it will be responsible for specifying how the external services will use and process the data.
- *Data Processor*: a data processor processes the data that is provided to him by the data controller, but he does not own the data he processes or controls. This means that the data processor cannot change the purpose and the means in which the data is used, since this is defined by the data controller.
- *Representative*: is a natural or legal person designated by the controller or processor, that represents the controller or processor with regard to their respective obligations under this Regulation.

- *Owner*: is the owner of personal information.
- *Recipient*: a natural or legal person, public authority, agency or another body, to which the personal data are disclosed, whether a third party or not.

Who can assume these roles in a blockchain network is an issue that is under discussion and it depends largely on the type of blockchain network (permissionless or permissioned, as discussed in Chapter 2).

Beyond the assignment of roles, part of the responsibilities of these actors can be automated through an smart contract, which achieves: i) treat personal data only following documented instructions; ii) ensure that the people authorized to process personal data have agreed to respect confidentiality or are subject to a confidentiality obligation and iii) delete or return all personal data once the provision of the processing services is finished and delete existing copies unless the conservation of personal data is required under Union law or the member states.

3.2 Shared responsibility

Regarding the responsibility in the processing of personal data in a blockchain network, the cases of permissionless (open networks) and permissioned (closed networks) networks should be considered separately. In the case of closed networks, it is easier to define who is the responsible, so whenever possible it is better to use this type of networks. In an open blockchain network this role cannot be applied to a person or institution, but it can be analyzed if it is admissible in the legal field, consider the shared responsibility of all or some of the members of the blockchain. This is considered in the GDPR in its *Article 26*, as joint controllers for the treatment, if the following is fulfilled *"Where two or more controllers jointly determine the purposes and means of processing, they shall be joint controllers"*.

The report [Eur19] analyzes the possible responsibilities of the different roles involved in a blockchain network comparing different reports. These roles are *network developers*, part of the volunteer who adjusts the protocol using the blockchain. *Nodes*, who sign and send transactions to the blockchain network through a node. *Miners*, who execute the transactions sent by the nodes. Finally, *smart contract developers*, that are companies or developers that create smart contracts used on the network.

The report refers to the document *"Blockchain and the GDPR of the European Union Blockchain Observatory and Forum"* [LCT18], that dismisses the responsibility of developers, with the understanding that *"they volunteer to work on an open source project and, in many cases, they do not receive direct compensation for their efforts and, in essence, they simply create a useful tool, they do not prescribe how this tool should be used"*. As for network users, who sign and send transactions to the blockchain network through a node, it is stated in this document, that if they send personal data to the blockchain as part of a commercial activity, they are more likely to be considered data controllers. However, if they send their own personal data for their own personal use, for example, to buy or sell cryptocurrencies, they are likely to be subject to the family exemption of the GDPR and cannot be considered data controllers. An argument in favor of using permissionless networks, is that by choosing to use the network, the user is giving his consent. However, the GDPR stipulates that the consent must be specific and unambiguous, which implies an active granting of permission, not a passive one. Similarly, it could be argued that when initiating a transaction, a user is assuming a contractual obligation with the platform, but here it is also a passive act without explicit terms. However, *"Commission Nationale de l'Informatique et des Libertés"* (CNIL) [Com18] notes that the

nodes, which have the right to write in the chain, can be considered data controllers, arguing that blockchain nodes define the purposes (objectives pursued by the processing) and the means (data format, use of blockchain technology, etc.) of processing. On the contrary, the CNIL does not consider the miners as a data controller since they only validate transactions and do not decide on the purpose and means of processing.

The CNIL has indicated, in relation to smart contracts, that software developers can be external providers but, if they actively participate in data processing, they can also be joint processors or controllers, depending on their role in determining the purposes of processing. The CNIL also considers the responsibility of smart contracts developers as a data processor. An example is that of a software developer offering a solution to an insurance company, in the form of a smart contract that allows passengers to be automatically reimbursed when their flight has been delayed. This developer would be qualified as a data processor if it is involved in the processing of personal data, the insurance company being the data controller.

As background of responsibility, in Google Spain, the Court of Justice [LCT18] emphasized the need to *"guarantee, through a broad definition of the concept of controller, effective and complete protection of stakeholders."* Consequently, the Google search engine operator was qualified as a data controller even though it did not exercise control over personal data published on third-party websites.

What is clear is that we are in a grey area where, in some cases, it will not be possible to identify a controller. The issue of legality is more direct in the context of an authorized private network, since it is possible to require each participant of the network to accept certain terms and conditions before being granted access to the network.

3.3 Pseudonymisation and data anonymization

The GDPR differentiates the processing of pseudonymised and anonymized data. With regard to pseudonymised data, the *Recital 26 of the GDPR* states that this type of information is under the scope of the GDPR, while the principles of data protection should therefore not apply to anonymous information. Pseudonymisation is important for risk minimization (*Recital 28 of the GDPR*), but it should not be seen as an anonymization technique.

Related to this, the document [Eur19] analyzes the *Article 29 Working Group* (an independent European working group that has dealt with issues related to the protection of privacy and personal data) [Art14], who defines some pseudonymization techniques used by blockchain, such as encryption with secret keys and hash functions. They are considered pseudonymization techniques because it is still possible to obtain the original data which is supposed to be protected.

This article defines three different criteria to determine whether anonymization is "irreversible" or "as permanent as erasure". i) *Singularization*, defined as the possibility of extracting from a data set some records that identify a person. ii) *Linkability*, the ability to link at least two records of a single interested party or a group of stakeholders, either in the same database or in two different databases. If the attacker can determine (through a correlation analysis) that two records are assigned to the same group of people but cannot single out people in this group, then the technique is resistant to singularization, but not the linkability. Finally, iii) *inference*, the possibility of deducing the value of an attribute from the values of other attributes.

3.4 Personal data management using blockchain

The GDPR guarantees rights to the owner of personal data that is managed by a third party, including the right to access the data and the right to erase and rectify it. The law also requires data protection and transparency in the processing of that data. In what follows we analyze the fulfillment of these rights in a blockchain-based system.

3.4.1 The right of access to personal data

In GDPR, in its *Articles 12 to 15 and Article 20 of the right to data portability*, indicates the right of the owner to request information about personal data and the obligation to provide it within a month. That information might concern the identification of the stored data, to whom it has been transmitted, the period of retention and the existence of automated decisions on the data, among others. This type of access to data is facilitated with blockchain, since the data is, in fact, available to all members of the network.

3.4.2 Confidentiality of personal data

Confidentiality can be a weak point for blockchain solutions, since the data reside by design in all the participating nodes of the network. In principle, all personal data can become accessible to all people with access to the chain. As we saw previously, an alternative is to store this information in an off-chain. As an alternative, proposals for addressing confidentiality issues using Trusted Execution Environments (TEEs) are described in [Eni20; Zha+20].

3.4.3 Deletion or modification of personal data

The GDPR stipulates the obligation to erase or modify the data at the request of the owner of the personal data (*Article 16 - Right of rectification*; *Article 17 - Right to be forgotten*; and *Article 18 - Right to limit the treatment of the GDPR*). This implies the need to delete data if either the data is not correct or the consent of the owner of the data is withdrawn, or when the stipulated period of use or the purpose for which the data was used ends. In turn, the data must be stored for a specific purpose and deleted when the service is finished (*Article 23 - Limitations of the GDPR*). In blockchain the stored data cannot be modified or deleted once it is in the chain, so alternatives must be sought to meet this requirement, such as storing personal data outside the blockchain, for instance in a off-chain storage. As to the right of rectification, it is impossible to modify the data in a block once it resides in the chain, so one way of achieving this is by entering the updated data in a new block and allowing a subsequent transaction to cancel the initial transaction, even if the first transaction is still in the chain.

3.4.4 Traceability of personal data

The GDPR law states in its *Article 30 - Registration of treatment activities*, the need to keep a trace of the activity carried out with personal information, such as to whom has the data been shared. Additionally, *Article 15 - Right of access of the interested party*, indicates the right of the data owner to receive information regarding the processing of his data, including with whom has the information been shared. The activity log is natural in blockchain, where everything is stored in the chain and all participants have the information of the system transactions, so these requirements are met by design.

3.5 Public/private keys as identifying data

Blockchain uses a public/private key system to identify the owners and recipients of the transactions, so it is quite crucial to analyze how this information is managed and whether it is considered or not as personal data. Related to this, *Recital 30 of the GDPR* states that persons may be associated with online identifiers and, as discussed in [Eur19], in the context of blockchain public/private keys serve as the type of identifiers mentioned in this recital, since they are often used to identify the origin and destination of each transaction and to sign the transactions. Therefore, they should be treated as personal data, that is, as an identifier of a person.

Public keys can also reveal a pattern of transactions that could be used to identify an individual user. Related to this, report [Eur19] mentions a judgment on April 2014 of Digital Rights Ireland [Tjc], where it was considered that metadata (such as location data or IP addresses) can also be personal data, since *"those data, taken as a whole, may allow very precise conclusions to be drawn concerning the private lives of the persons whose data has been retained, such as the habits of everyday life, permanent or temporary places of residence, daily or other movements, the activities carried out, the social relationships of those persons and the social environments frequented by them"*.

The problem is that in blockchain public keys cannot be obfuscated or deleted since they are used to identify transactions. Related to this, the document on blockchain and the GDPR of the *European Union Blockchain Observatory and Forum* [LCT18] indicates that one way of trying to avoid traceability is to use a new pair of keys for each transaction to prevent them from being linked to a common owner. The objective is to mitigate the risk, if the owner of a key is revealed, that the binding could reveal all other transactions that belonged to the same owner.

3.6 Hashes as pseudonymized information

It can be inferred that applying a hash algorithm on personal data is a way of pseudonymizing that data, since it can indirectly identify a personal data. Thus, hash values used in blockchain must be considered personal data. Related to this, the work reported in [LCT18] warns that when using a hash function it is necessary to be aware that patterns that could allow traceability are not being created. This is due to the risk of bonding, what refers to situations in which pattern analysis allows an analyzer to discover information about a particular individual. The risk of bonding increases if simple information is saved, because the outputs of a hash function can be guessed from known inputs. The paper gives as an example an application that performs purchase or sale transactions that publishes a hash with the buyer's address on a blockchain to keep a record of each transaction. In this case, if the registered hash is the same each time a user orders a transaction, can be easily inferred the times and frequency of each user's transactions. That is because if simple information is saved, the outputs of a hash can be guessed from known inputs. An example mentioned in the report [Eur19], is given by Edward Felten of the *American Consumer Protection of the Federal Trade Commission* [Fel12], showed that it is quite easy to establish someone's identity based on the hash functions that are derived from social security numbers, doing brute force work on the possible social security numbers for a country (about a billion in the United States).

Therefore, it is advisable to use a salt in the hash function as a means to reduce the probability of obtaining the input value. Nevertheless, the *Article 29 Working Group* [Art14] makes it clear that the use of this technique does not produce anonymous data, since it is still possible to calculate the original attribute hidden behind the hash value.

The *data protection Agency of Spain* analyzes in [Age19] the reversibility of a hash function and concludes that, when the data used in a processing operation has an implicit order, the set of possible messages is reduced, which makes message reversal easier. Given a dataset to which a hash is to be applied, the degree of entropy of the dataset influences the reversibility of the hash. The smaller the message space the lower the entropy are, the lower the risk of collision in hash processing is, but re-identification will be more likely. On the contrary, the higher the entropy is the higher the possibility of a collision, but the risk of re-identification will be lower. In [Age19] it is also stipulated that it is also necessary to take into account the identifiers linked to a hash value, because the more personal information is linked, the higher the risk of identifying the contents of this hash value. Finally, is is also suggested using a single-use salt model that generates a separate random element for each message. This random element must be completely independent of any message and any other salt generated for any other message. The format of the extended message suggested for processing a hash would be as seen in the Figure 3.1.

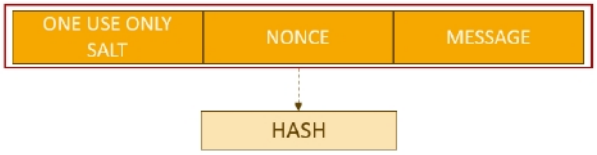

FIGURE 3.1: Format of the extended message for processing a hash [Age19]

3.7 Deletion of hashes and public key as deletion method

Given the immutability of the data stored in blockchain, it is important to determine what can be considered acceptable for personal data protection regarding data erasing. An alternative to erase the information is to process the data, instead of deleting it, so that it becomes anonymized and it is no longer within the scope of the GDPR. In the case of hashes, this is equivalent to eliminate the original data from which the hash was formed. Regarding anonymization techniques, the work [Eur19] concludes that the GDPR takes a risk-based approach, as it takes into account not only current technology, but also future one. *Recital 26* of the GDPR states that for anonymization techniques it should be considered which is the available technology at the time of the processing. In that respect, the Article 29 Working Group [Art14] postulates that the possible advancement of technology should be considered in the period of time in which personal data will be stored. In the case of blockchain this period of time is undefined, so in [Eur19] it is argued that any data should be considered as personal data, since it cannot reasonably be assumed that identification will remain anonymous in the future.

Despite these arguments, there are antecedents where anonymization has been accepted as a form of erasure. In fact CNIL, in an article on blockchain [Com18], accepts the deletion of an original data as a method of deletion, even if the hash remains in the blockchain.

3.8 Concerning the Uruguayan personal data protection laws

In this section we will discuss Uruguayan personal data protection laws and their relationship with GDPR. In 2008 the *Law 18.331* [Rep08] about protection of personal data

was promulgated in Uruguay. This law considers the protection of personal data of natural persons and (unlike the GDPR), of legal persons. In 2018 the *Law 19.670* [Rep18] was published, which made changes regarding the protection of personal data, in order to comply with European regulations. This law was regulated on February 2020, through the decree 64/020 ([Rep20]). The recitals analyzed on the compliance of the GDPR by blockchain technology also apply to the Uruguayan laws on personal data since, as will be seen below, both regulations contemplate the main analyzed points.

3.8.1 Scope, roles and responsabilities of the personal data law

Uruguayan laws apply to the Uruguayan territory and to companies whose products or services are offered to inhabitants of Uruguay. According to the *Law 18.331*, personal data is defined as information of any kind referred to determined or determinable natural or legal persons. This latter makes a difference with the GDPR, since legal persons are also considered included in the law. The personal data protection standards for the Ibero-American States [Dat17] also differ in this, since they consider personal data as any information concerning an identified or identifiable natural person.

The fact that determinable personal data is included in the scope of the Uruguayan laws, makes them apply the concepts we have discussed concerning pseudonymized information. In turn, the Uruguayan laws handle the concept of disassociated information, that is, information that is not related to an identified or identifiable person, which is outside the scope of the law, as well as the anonymous information of the GDPR. Thus, we can apply the same previously discussed criteria regarding the use of public/private keys and hashes, which were analyzed for the GDPR law. Also, like in the GDPR case, it is recommended to apply proactive security principles such as security by design or privacy by default.

Regarding the roles, similar to the GDPR the Uruguayan *Law 18.331* defines the role of the person in charge of a database or treatment, to which is added the role of *data protection delegate*, defined in the new *Law 19.670, Article 40*. This causes the same liability problems to arise for blockchain technology. Although, the possibility of shared responsibility is not explicitly mentioned as it is in the GDPR, which could make less direct the application of the shared responsibility criteria analyzed before.

3.8.2 Personal data management using blockchain

The same principles of the GDPR that were previously analyzed are applicable concerning the treatment of information. For example, in the *Law 18.331* the right of access to personal data is contemplated in *Article 14 - Right of access*. On the other side, the Uruguayan laws also contemplate the obligation to erase or modify the data upon request of the owner of the information (*Article 7 of Law 18.331*). In turn, the information must be stored for a specific purpose and deleted when the service is finished (*Article 8 - Principle of purpose of Law 18.331*). Therefore, the analysis made regarding the deletion or modification of personal information in GDPR is equally valid.

Traceability requirements are less explicit in the Uruguayan laws than in the GDPR. In turn, unlike the GDPR, in *Law 18.331*, although the right of access to information by the owner of personal data is mentioned (*Article 14*), it is not explicitly named the obligation to inform the owner of personal data about to whom has the information been transmitted.

Similar to the GDPR, the Uruguayan laws regulate the actions to be carried out in the case of an incident, through *Article 38 of the Law 19.670*, which indicates the obligation to inform the owners of the personal data and to the Regulatory and Personal

Data Control Unit (URCDP) of Uruguay, after a security event. *Article 12 - Principle of responsibility*, indicates the evaluation of the impact on data protection as one of the techniques indicated for the treatment of information. Additionally, the new law indicates that one of the obligations of the delegate of data protection is to propose all the pertinent measures to comply with local and international regulations and standards related to personal data protection. As we have pointed out, blockchain can help fulfill some of these concepts.

3.9 Weaknesses and strengths of blockchain and off-chain solutions regarding personal data protection regulations

Considering the previous discussion, we understand that the data protection requirements defined by GDPR can be hardly addressed alone by the blockchain technology and the concept of a ledger stored by all participants of the network. In permissionles blockchain networks in particular, as there is no identified and assigned managers of the network, it is not possible to ensure that access to the data is controlled and regulated. In the general case, we find it difficult for systems built using blockchain technology to guarantee the confidentiality, deletion or modification of the data managed in those systems.

As a summary, we can indicate as breaches of blockchain permissionless against the personal data protection laws, issues like i) lack of assigned managers, ii) lack of control over access to information, iii) lack of control over regulation in the use of information and the fact that iv) there is no territorial control. And in general for blockchain technology there is i) difficulty in guaranteeing the confidentiality of the information, ii) difficulty to delete or correct information and to iii) comply with the right to a human intervention goes against the automation proposed by blockchain.

On the other side, blockchain technology meets by design some of the requirements of these data protection laws, such as the right of access to data by the owners, traceability, encryption and hash techniques as security by design mechanisms and finally transparency in data processing.

The use of public keys is necessary by design in blockchain, so if keys are belong to natural persons (and legal entities in the context of Uruguayan law), alternatives must be evaluated to avoid traceability of these keys, such as using unique keys. Regarding other personal data, as they clearly cannot be stored in the blockchain, the recommendation is that personal data should be stored in an off-chain, leaving in the blockchain a link to access personal data. To ensure the integrity of the information, techniques defined in off-chain can be used, such as storing a hash with the information in the blockchain. When doing this, it must be carefully evaluated what type of information is hashed and how this hash is done (using salt, for example), to avoid the problems described in Section 3.6. In case of storing hashes in the blockchain, the possibility of anonymization as a deletion method should be argued, as described in Section 3.7. As an option, alternative methods may be sought that do not involve storing personal hashed information, such as using zero-knowledge generation mechanisms. However, these mechanisms are rather intended to confirm that a third party has certain information or meets a certain requirement, so it is not always aligned with the needs of the business (such as verifying the integrity of stored data).

3.10 Privacy-oriented blockchain design guidelines

We put forward the following guidelines for the design of a system that integrates blockchain technology with off-chain processing of data and that is intended to comply with personal data protection requirements like those stated in the GDPR:

- In the design stage evaluate the impact related to data protection. This will contribute to meet the principle of *data protection by design and by default* described in the *Article 25* of the GDPR. It will also help defining what information should be stored or processed in the off-chain and how the blockchain and off-chain mechanisms shall interact. Additionally, consider use encryption or hash functions as a means of data protection.
- In order to comply with *Article 5 - Principles relating to processing of personal data* and *Article 6 - Lawfulness of processing* of the GDPR, the treatment of smart contracts data should be regulated in accordance with what was agreed with the owner of the data. Related to this, paper [DE+21] proposes a system architecture that implements two smart contracts, a Consent contract to register the authorization of the data owner to the data controller and a Purpose contract, which contains the permission given to a data processor.
- As explained in Chapter 2, if the personal information is stored off-chain, an access control system must be implemented.
- GDPR, in its *Article 15*, considers the right of the owner of the personal data to request information about the processing of their data. Therefore it is important to consider implementing this access upon request if the information is stored outside the blockchain.
- Transparency in the processing of information, one of the principles defined in *Article 5* of the GDPR, could be fulfilled by publishing the smart contract used.
- Operations must be registered according to *Article 30* of the GDPR, such as to whom data was given and all the activity carried out.
- Mechanisms should be implemented to allow automatic deletion of data, for example when the consent of the owner of the data is revoked or the service has ended.
- As was explained on Chapter 2, consider implementing evidence generation mechanisms such as hash values or zero-knowledge to ensure integrity in the processing of informationm when using off-chain.
- Evaluate alternatives of which off-chain model use, in order to guarantee the availability of the data.

Chapter 4

Privacy and security oriented blockchain design guidelines

According to the analysis presented, we can conclude that the design and implementation of systems that use blockchain technology must incorporate complementary mechanisms in order to comply with the data protection requirements stipulated by the GDPR.

In particular, our proposal is to consider an hybrid model in which off-chain is used to store personal data and blockchain to ensure integrity, traceability and access control. In short, personal data should not be stored on the blockchain, but instead register a link to be able to access the data residing in an off-chain storage. Additionally, it must be carefully evaluated what type of information is hashed and how this hash is done (using some salt, for example). We also postulate that if a system is intended to use off-chain processing of personal data, it is preferable not to use channels model (Section 2.2.2), since the controversies in it are resolved by executing the processing in the blockchain, exposing personal information. A centralized model may be more appropriate.

4.1 A system architecture and use case model for hybrid blockchain and off-chain solutions

We first proceed to schematically present in Table 4.1 what are the blockchain and off-chain functional components that combined make it possible to design a solution satisfying data protection requirements. In particular, given a data security or privacy property we point out the behavior that can be (or can not be) guaranteed by a blockchain and off-chain mechanism. Associated to the property we also make explicit the bases that shall constitute the methodological reference to build the architecture of hybrid models.

Blockchain and off chain models to complain GDPR			
Point to comply	**Blockchain**	**Off chain**	**System architecture considerations**
Responsibility	If possible, use permissioned networks, where it is easier to define the responsibilities	Use storage off-chain with commit-chain model, to ensure that data is never stored on the blockchain	Responsibility is based on private blockchain models and centralized off-chain models
Confidentiality	Difficult to ensure the confidentiality of the data stored	Use commit-chain model, to ensure that data is never stored on the blockchain	The data stored in off-chain must be encrypted, as well as the communications
Access Control	The access control policy should be stored in the blockchain, to ensure integrity and transparency	Validate access control policy to authorize access to data	Offchain authentication and authorization validation using the access policy stored in the blockchain

The right of access to personal data	Solved by design in blockchain	Allow access of the owner to his own data, upon request	Off-chain authentication and authorization valildation using the access policy stored in the blockchain
Delete / Modification of personal data	The immutability of blockchain prevents to perform these operations	Storage personal data in off-chain to guarantee this right	The process of deletion or modification must be authorized, as well as the process of access to data
Integrity	Solved by design in blockchain	Additional integrity control mechanism must be implemented	Store in blockchain a hash of the data stored in off-chain as an integrity control check
Traceability	Solved by design in blockchain	Off-chain does not offer traceability by design	Use blockchain as audit log
Transparency	Smart contracts used in blockchain networks can be published	Minimize process in off-chain	All access to data on the off-chain must be recorded in the audit log, which is kept in the blockchain
Availability	Solved by design in blockchain	Issue to resolve when choosing commit-chain architecture	This point should be evaluated when instantiating an off-chain solution

TABLE 4.1: GDPR-complying design bases

Given the functional characteristics of the blockchain technology, it has been suggested, for instance in [Tru+19], to use those functionalities to build embedded in the system i) an *access control network*, that stores the access control policy and, upon request, performs the corresponding authentication and authorization tasks, and ii) an *audit network* that records all actions that have been performed, such as authorizations, access requests, modification and deletion of data. It must be registered, for instance, who accessed the data and both approved and denied accesses. The immutability of blockchain provides an appropriate setting for recording audit logs.

In addition to those two networks, we propose to use additionally the blockchain as an *integrity network* in order to validate data stored off-chain. When a data is stored in the off-chain network, a link can be stored in the blockchain network so that it is possible to locate the data. Together with this information, it can be stored an integrity control check mechanism, for instance using a hash function.

The use of blockchain and off-chain is described in the Figure 4.1.

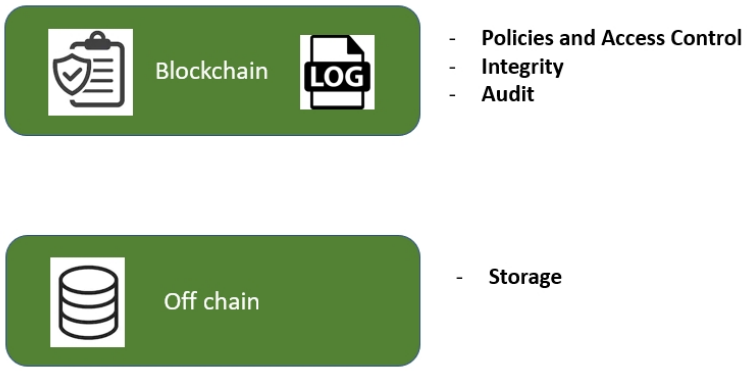

FIGURE 4.1: Networks used for handling personal data

In this high-level system architecture, we can use a Gateway for external communication towards the blockchain and off-chain, similar to how is done in the architecture proposed in [Mol+18]. As authorization mechanism we propose to use a Security Token Service, like the one proposed in [Tru+19]. In order for a user to access personal

data, which is stored in the off-chain, first he has to obtain an authorization token from the access control network. When attempting to access the data, the user presents that token to the off-chain, which in turn validates it with the access control network.

4.1.1 The use case model

This model consists of eight use cases that cover the principal services we understand can be used to build a blockchain and off-chain based system compliant with well established security and privacy requirements.

We consider the following actors as requestors of services provided by and to the system: *Data owner* (DO), the user, owner of personal data; *Data controller* (DC), responsible for data processing; *Data processor* (DP), a data controller can delegate processing tasks to a data processor and the *Receiver (recipient)*, a third party who wants to access personal data. These actors must be registered in the system to be able to perform authentication and authorization operations. For this, different solutions can be used, for example using a system of certificates with private-public key to authenticate data owner and a user and password system for recipients.

In what follows we introduce a brief characterization of each use case, to later analyze them in detail. This use cases are illustrated in Figure 4.2 and are formally specified by the sequence diagrams illustrated in Figure 4.3.

1. *Register personal data*, data is stored in the off-chain and registered in the access control and integrity networks;
2. *Grant access*, authorization of access to personal data is granted to a third party, the data policy is updated in the blockchain;
3. *Revoke access*, revocation of third party access permissions;
4. *Data access*, authenticated and authorized access to data stored in the off-chain;
5. *Verify data*, integrity verification of data stored in the off-chain;
6. *Delete or modify data from owner*, request by the owner of modification or deletion of his data;
7. *Delete or modify data from data controller*, request of modification or deletion of data controller; and
8. *Request for access log*, requested by the data owner or an external authority. All these operations are in turn registered in the audit network.

Register personal data use case

The first use case corresponds to entering the personal data into the system. In order to register personal data, either the DC or the DP must first obtain authorization from the DO to record the information and establish a policy regulating the access to that information. This can be done by submitting a request signed by the user and either the DC or the DP. Then, the data is sent to the Gateway, that validates the access intent with the access control network blockchain. If the access policy authorizes the registration, the Gateway sends the data to the off-chain. Finally, the off-chain network communicates with the integrity network blockchain to stores an integrity control (a hash of the personal data). All these changes are stored in turn in the audit network blockchain.

Grant Access use case

The second flow considers the case where access to a third party (recipient) is processed. In order for a recipient to access the information, the DO and the DC or DP must authorize the request and send this authorization to the blockchain, so that the access policy is updated. This implies, therefore, registering the recipient in the system. It should be noted that the off-chain does not intervene in this flow since the data is neither accessed nor modified.

Revoke Access use case

The process of removing permissions involves updating the access policy stored on the blockchain. This action can be done by the DO, DC or DP. The blockchain first validates the identification and access authorization, before processing the requested change. Finally the change in the audit is recorded.

Data access use case

When a data owner or recipient wants to access to personal data, it has to send a request to the Gateway. The Gateway then sends the request to the access control blockchain network, which validates the access by reviewing the defined policy and, if the person requesting access is authorized, the blockchain delivers to the Gateway a token and the link to the data stored in the off-chain network. With this information, the Gateway can request access to the off-chain network, which in turn validates the token against the access control blockchain. If the Token is validated, the off-chain network returns the data and the Gateway sends the data to the requestor. All these accesses are stored in turn in the blockchain audit network.

Verify data use case

Once personal data is available in the system, the data owner or a third party might be interested in verify its integrity. We take the view that the processes of validating personal data and accessing the hash used to perform the validation are constitute sensible operations. Therefore, it is necessary to implement access control for this use case. To do that, the Gateway first validates access against the access control network blockchain. Once authorized, the integrity blockchain performs the integrity control of the data passed by the user. This is done by comparing the stored hash with the one presented by the user. It should be noted that in this use case it is not necessary to consult the off-chain.

Delete/Modify Data from Data Owner

If the DO wishes to modify or delete his information, he can communicates it to the DC or DP who send the request to the Gateway. The Gateway verifies the policy against the access control network blockchain who sends a token to validate the access (similar to the register personal data use case). The Gateway sends the request to the off-chain together with the token, so that the off-chain can verify it against the blockchain. If the token is validated, the off-chain network makes the change and adjusts the integrity check hash.

Delete/Modify Data from DC or DP

In turn, the DC or DP can modify or delete the information they have stored, giving notice to the data owner of the setting. Depending on the action to be taken and the context, it may be necessary to previously request the authorization of the DO. For example, we understand that in the case of deletion of information, it is only required to report the fact to the data owner, while the modification of a data may require prior authorization.

Request for Access Log

Finally, the data owner or an authority can request audit information from the DC or DP, who consult it with the blockchain. As in the previous cases, before providing the information, the access policy must be reviewed.

4.1.2 Some implementation considerations

In addition to the design constructs just described, it is also important to evaluate which network models will be adopted, as well as other implementation considerations.

As previously discussed in Chapter 3, the creation of a permissioned blockchain network is recommended to handle personal data. In turn, the use of an off-chain with commit-chain architecture network is recommended, but, being a centralized and non-distributed system, this implies having to design solutions to ensure system availability.

As already mentioned, a hash value of data could be stored in the blockchain as an integrity control mechanism. Thus, the proposed validation is based on the comparison of the hash of the data stored in the blockchain with the data stored in the off-chain. To do this safely, namely avoiding the hash reversibility problem, it is necessary to consider the recommendations given in Section 3.6, designed to increase the entropy level of possible hashes. That implies, for instance, introducing random elements and taking into account the identifiers linked to a hash. Related to this, it is important to carefully choose the certificates and public keys to be used to identify the actors of the system, for example, analyzing if it is applicable to use a new pair of keys for each transaction in order to a void traceability.

4.2 A methodology for security and privacy threat modelling

The analysis of security and privacy risks of a computational system includes, in a joint and complementary way, the study of aspects of hardware, software, operational practices and policies of the organization and application of the basic principles of protection of personal data. The basic methodological component of work that is typically applied to conduct one such risk analysis consists of the following steps: i) characterization of the context of the processing of personal data that takes place in the system; ii) creation of a catalog of the critical assets in this particular context and the security and privacy properties to be guaranteed; iii) identification of the threats to which these assets may be exposed; iv) assessment of the risks involved; and if required v) proposal of adequate measures to mitigate/treat critical level risks.

The execution of those procedures is carried out by making systematic use of the processes and catalogs of good or known practices provided by recognized methodologies for the analysis of privacy risks. In what follows we shall briefly describe a methodology that provides support for developing a security and privacy threat model

for solutions that make use of blockchain technology to build privacy aware information systems.

4.2.1 Basic steps of the threat model

There exist several methodological proposals for the development of threat models. We shall follow the one proposed by OWASP [OWA20], which consists of a process with six steps. The first step is to identify the *security and privacy properties* that should be guaranteed the system satisfies. In order to identify security objectives we use the CI4AM security mechanism [OWA20], which proposes grouping security objectives into categories of confidentiality, integrity, availability, authentication, authorization, auditing and management. Additionally, we shall also require for *compliance* with the requirements of the GDPR, which relate to the processing of personal data.

The second and third steps consist of *profiling* and *decomposing* the application. The first one involves asking questions like: where the system will be displayed?, who are the actors?, what data does it handle?, which permissions the actors will have?, what technology will be used?, what security mechanisms are going to be applied? By decomposing the system it is meant to break down the application into the following components: trust boundaries (indicates where trust level changes), entry points (principal attack targets), exit points and data flows. We use the Threat Modelling tool of Microsoft [STR20] that provides mechanical support to perform the decomposition of the application.

Once the components of the application have been identified we shall proceed to perform the *threat analysis*. This consists of determining, characterizing and classifying threats, possibly based on the attacker, the impacted assets and the attack method. Once the threats are detected it is required to identify the *vulnerabilities* the system has and that can be exploited by the threats analyzed. Finally, we *rank the threats* and define if each risk can be either accepted, mitigated, transferred or ignored.

4.2.2 Threat risk analysis

In order to achieve both a security and a privacy analysis we shall make combined use of the well-known Microsoft's STRIDE methodology [STR20] and the risk methodology defined by CNIL for privacy risk management [CNI12]. STRIDE considers threats strictly from a security point of view, categorizing them into spoofing, tampering, repudiation, information disclosure, denial of service, and elevation of privilege. These threats are related to authentication, integrity, non-repudiation, confidentiality, availability and authorization security properties.

CNIL considers threat groups related to processes, like unavailability of legal processes and change in processing. It also includes threat groups related to personal data, like illegitimate access to personal data (confidentiality), unwanted changes in personal data (integrity) and disappearance of personal data (availability).

Some of the threats related to *unavailability of legal processes* that we consider affect GDPR compliance are legal problems derived from: i) *lack of responsibility for the treatment of personal data*; ii) *lack of processes to ensure the veracity or the consent of the treatment*; iii) *lack of processes for the correct treatment of data*; iv) *lack of processes to delete data*; and v) *failures in the information provided to the user*. Related to change in processing threats, we consider legal problems derived from vi) *failures in the processes to ensure the veracity or the purpose of the treatment*, vii) *failures in the processes to delete data*, and viii) *deficiency of controls over the treatment*.

Blockchain and off-chain related threats are those described in Section 2.6.

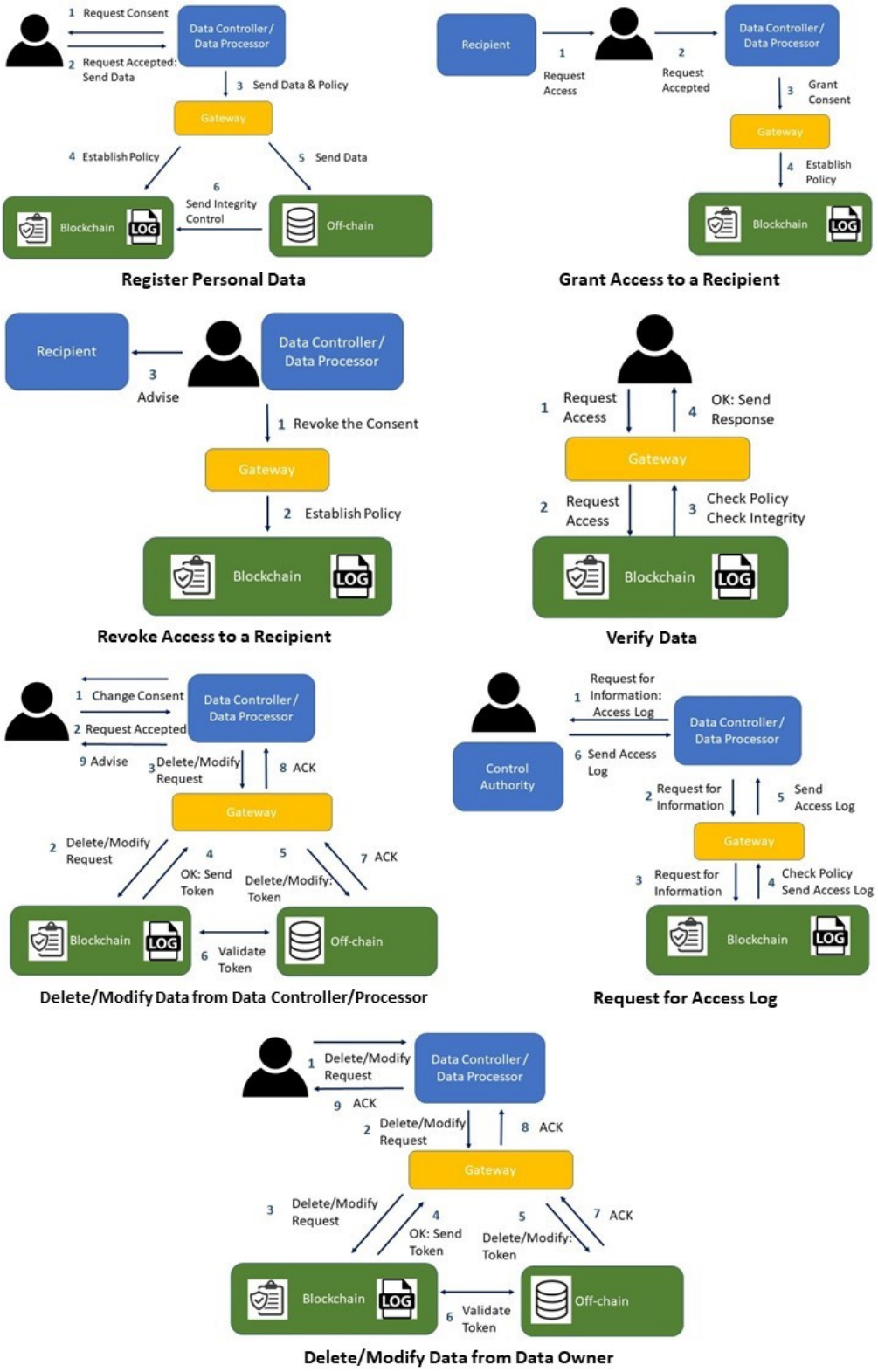

FIGURE 4.2: Use case diagrams

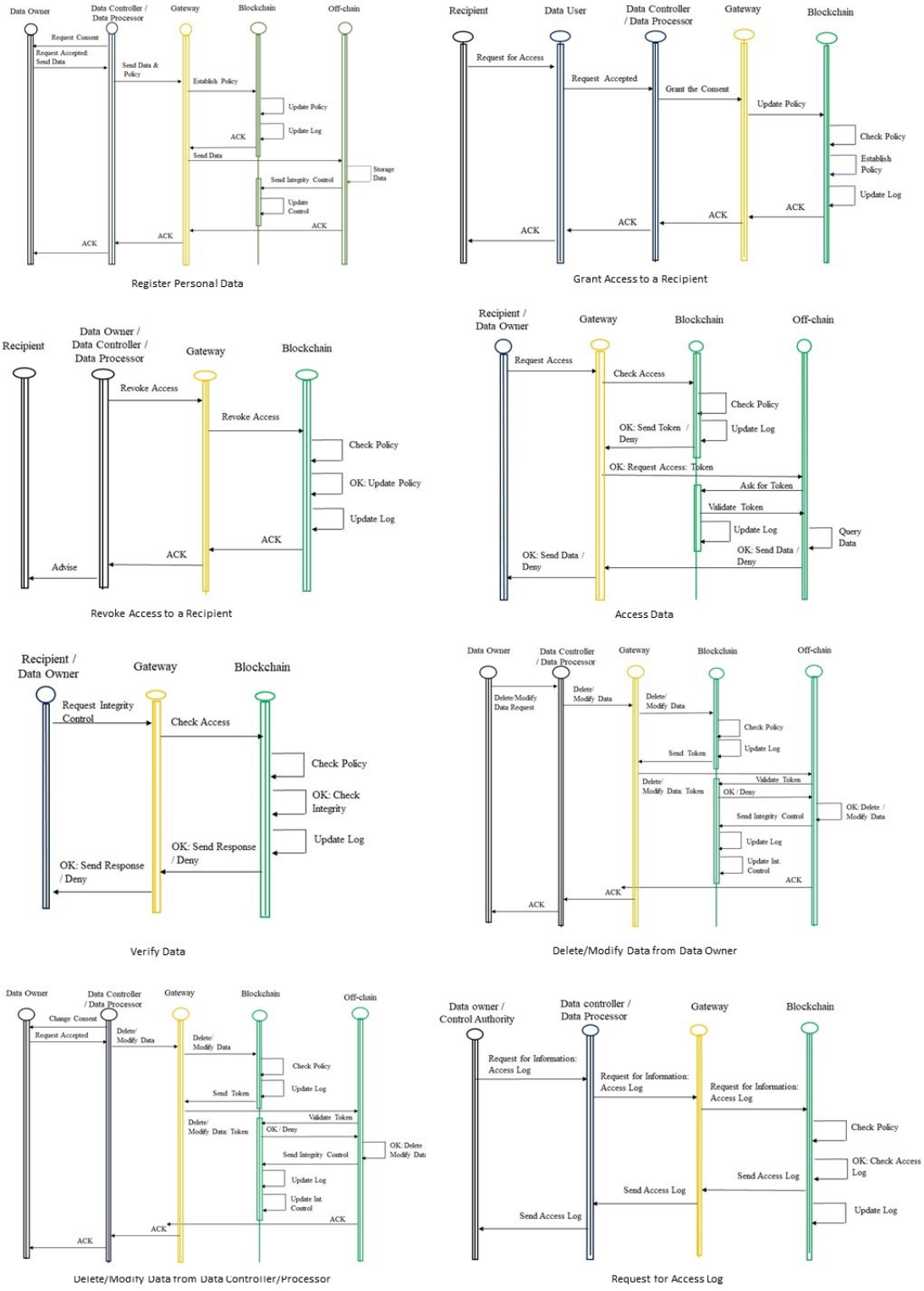

FIGURE 4.3: Sequence diagrams

Chapter 5

Applying the tools: a case study on digital academic certificates

The IT services unit of the Universidad de la República, Uruguay, which is called SeCIU (Servicio Central de Informática), is in the process of studying alternatives for implementing and deploying a system that provides functionalities to handle, store and validate university certificates, like degree diplomas, using blockchain technology. The main objective of the solution is that (graduated) students are able to obtain their digital titles or certificates and deliver them to third parties (for example potential employers), which in turn could be able to validate the authenticity and legitimacy of those certificates using that same system.

Institutions all around the word have been concerned for a long time with finding good solutions to solve the problems related to educational certificates. There exists worldwide a severe problem of fake degrees, which is aggravated by the fact that, for instance, employers do not have the ability to validate the certificates a candidate present when applying to a job post. In the context of certification of studies, the blockchain technology has been visualized as the solution to counter the manipulation of fake certificates by providing an easy validation mechanism that provides integrity assurance.

5.1 Existing solutions

There exist several solutions that have been proposed to deal with this problem, we now proceed to describe in more details the solutions we have analyzed. One of the systems analizyed by SeCIU is i) *Brazilian RAP System*. In Brazil, there exists a regulation that sets a 2-year period from March 2019 for universities to implement a diploma verification system. Universities are working to generate digital diplomas, using signed XML documents and blockchain. RAP system [al.18] proposes to use the public networks Ethereum and Bitcoin to provide existence and integrity controls, and off-chain network to preserve digital documents. Additionally there is an authentication module, that performs data validation with the blockchain network and retrieves the information from the off-chain network. Another system analyzed was ii) *Argentine BFA System*. This system was developed by the University of Córdoba and based on the BFA (Blockchain Federal Argentina) [bfa20]. BFA is based on Ethereum technology and works under the model of a permissioned blockchain. Once the University records are validated, a digital document is saved and digitally signed by the teacher, and then is stored on the blockchain. In BFA no documents or files are stored within the blockchain, only the hashes of those documents are saved. Beyond Latin America, there are similar systems, such as iii) *Blockchain for Education*. BFE[PKR20] consists of a network of universities that have implemented a hierarchy system to add Universities to the system and to

validate the signing certificates. There is an accreditation authority, responsible for authorizing the entry of other universities to the system and certification authorities, that are the universities that belong to the system, responsibly for signing certificates and storage it in the blockchain. BFE uses a public network Ethereum so, like adding any transaction on this network, adding certificates comes into a cost, that is a disadvantage of the system. It also uses a smart contract for identity management and another smart contract to manage and store certificates in the blockchain. Finally iv) *EduCTX* [EDU20] is a system that proposes a global higher education credit platform, based on the concept of the European Credit Transfer and Accumulation System (ECTS). It constitutes a globally trusted, decentralized higher education credit and grading system, based on a permissioned blockchain network of higher education institutions (HEI). In this system, every time a student completes a course or save an exam, his HEI will transfer the appropriate number of ECTX tokens to his blockchain address. When an organization wants to verify the student's course obligation completion, the student has to send his blockchain address and redeem script to the verifier organization. Then the organization checks the amount of ECTX tokens againt the blockchain, which represents the student's academic credit achievements.

5.2 The design of our solution

We have specified the expected behavior of the system using the GDPR-compliant constructs of the system architecture introduced in Chapter 4. In what follows we shall present and discuss the most relevant design decision we have taken.

In the first place, we have defined the following mapping from the data related roles presented in Section 4.1.1 to the institutions and individuals pertaining to the problem's domain: i) *SeCIU* is the *Data controller* (DC), the responsible for the personal data handled by the system; ii) the *School Registry Offices* are the *Data processor* (DP), working as delegates assigned by the SeCIU; iii) the *Candidates* are the *Data owner* (DO), the owners of personal data; and finally iv) the *Receiver (recipient)* are the employers or institutions that want to validate a certificate.

Then, degrees and schooling (that corresponds to personal data) are stored in an off-chain network under the responsibility of the DC, and the operations of audit, access control and verification are carried out in a blockchain network that is accessed by School Registry Offices, students and authorized third parties, through a Gateway.

To validate the certificates, a hash of each certificate is stored in the blockchain, so that the validation system is based on the comparison of this hash with the hash of the certificate presented by the student. As was discussed in Section 3.6, hash values should be considered personal data as they are pseudo-anonymized data, then access to that information should be authorized by the Data owner. Therefore, certificate verification should not be a public function and access to that operation should be controlled. In this respect, the authorities of the Universidad de la República decided to formally ask the Uruguayan Data Protection Agency, the *URCDP* (*Unidad Reguladora y de Control de Datos Personales*), whether it is valid, from the legal point of view, to implement a system where the verification of a degree diploma is public. The response of the URCDP was that such verification requires the consent of the individual that earned the diploma and therefore cannot be provided as an open and public function (Dictum 9-2018, Expediente 2017-2-10-000394) [URC18].

Figure 5.1 depicts the data flow diagram (DFD) of the system, detailing the data and control flow that take place.

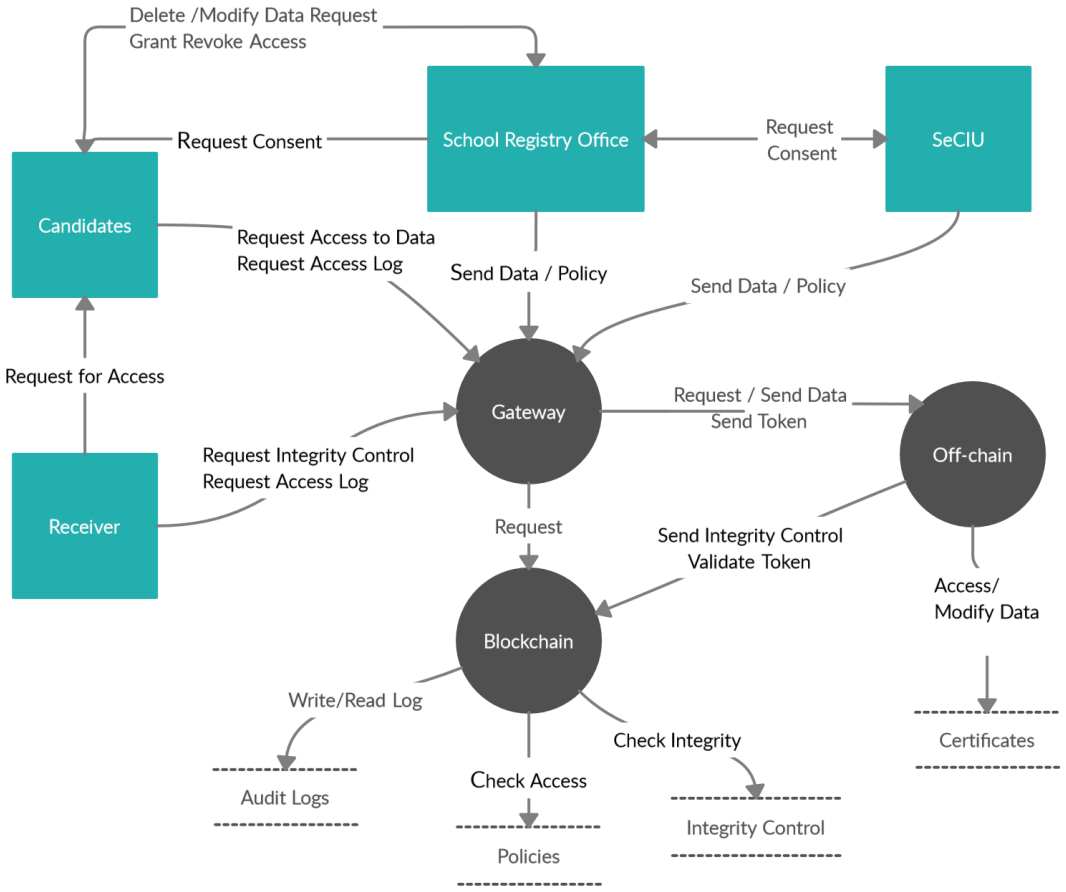

FIGURE 5.1: Data Flow Diagram of the system

Additionally, we have specified the behavior of the digital certificates system using refined versions of the generic use cases introduced in the use case model discussed in Section 4.1.1. We present and discuss the specification of some of the processes of the system.

5.2.1 Registration

Registration is carried out by the School Registry Offices. Thus, they are in charge of obtaining the authorization to manage data from its corresponding owner. The School Registry Offices send the data to the Gateway, which in turn sends the access control policy to the blockchain. The sequence diagram is described in Figure 5.2.

5.2.2 Grant and Revoke access

In order for a third party (an employer or another educational institution) to either access or verify a certificate, the student must give his consent and the system notified. This authorization will be stored in the blockchain network. The access authorization given by the student, though, can be revoked. This operation can be requested either by the student, the School Registry Office or the SeCIU. The corresponding sequence diagrams are described in Figure 5.3.

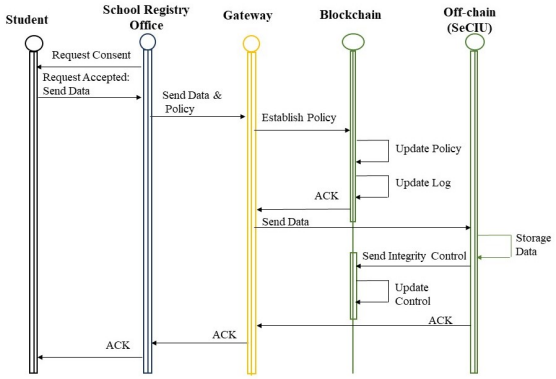

FIGURE 5.2: Sequence diagram POC: Register Personal Data

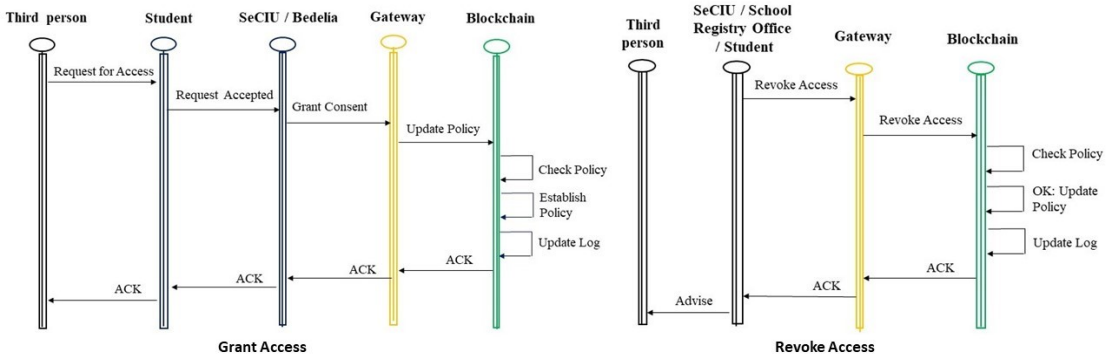

FIGURE 5.3: Sequence diagram POC: Grant and revoke access

5.2.3 Data access and verification

In this case it is considered that the access to the certificates will be carried out only by the students. The student sends a request to the Gateway, which in turn requests access to the access control blockchain network in order to validate the access policy. Authorization is done through a token, as described in Section 4.1.

Employers or institutions that want to verify a certificate (that could have been delivered by the student), can do so by consulting the Gateway if the student previously made the corresponding authorization. For this reason the Gateway first verifies this access, before carrying out the integrity check. The sequence diagram is described in Figure 5.4.

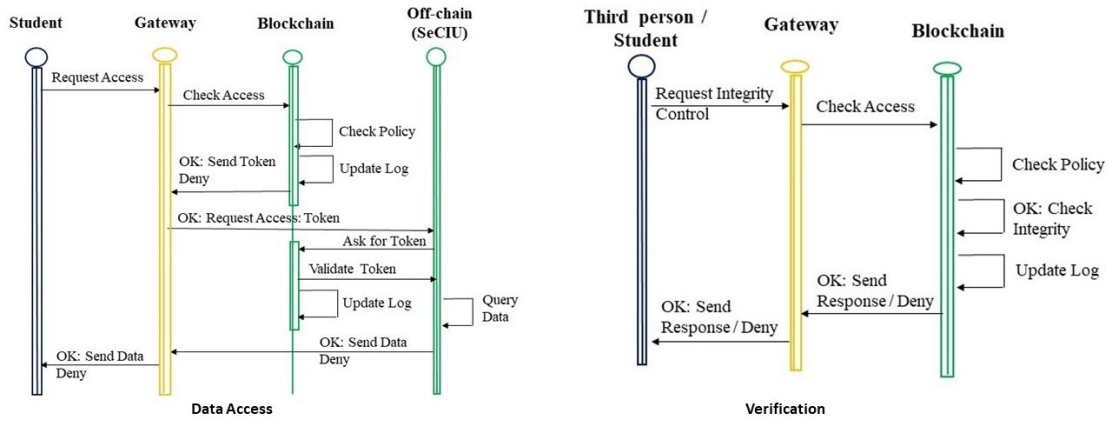

FIGURE 5.4: Sequence diagram POC: Data access and verification

5.2.4 Delete/Modify data from Owner, DC or DO

It could be the case that a student requests to modify stored information owned by him. That request is evaluated and processed by the corresponding School Registry Office, passing the request to the Gateway. The change involves checking the access policy, updating the information in the off-chain and updating the integrity control in the blockchain.

Similarly, it may be SeCIU or the School Registry Office who must modify a student's data (for example, the correction of a grade). The cycle is similar to the previous one, but notice of the modification is also provided to the student. The corresponding sequence diagrams are described in Figure 5.5.

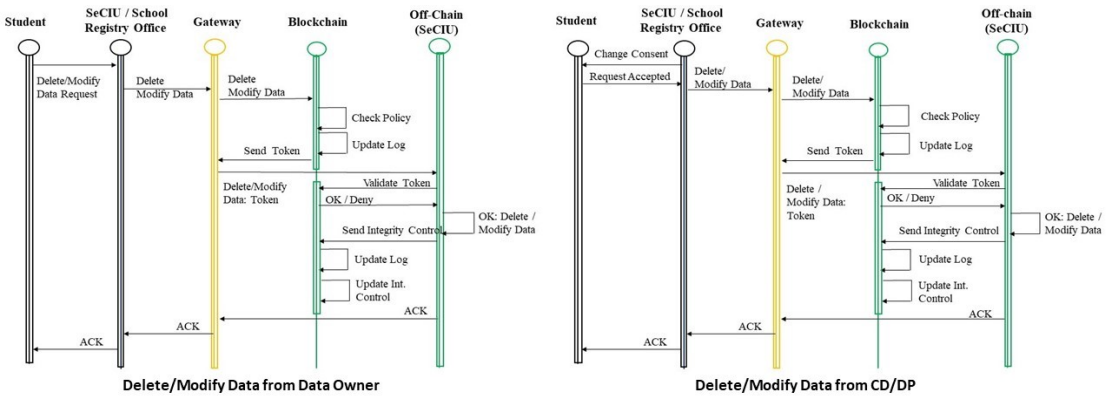

FIGURE 5.5: Sequence diagram POC: Delete/Modify data from Owner, DC or DO

5.3 Security and privacy threat modelling of the system

We shall illustrate the use of the proposed methodology discussing the definition of the security objectives and reviewing the threat analysis we have performed.

5.3.1 Security objectives

We have identified the following security objectives:

- *Confidentiality:* protect certificates against unauthorized access and allow verification process only to authorized persons;
- *Integrity:* certificates must not be modified without authorization and must be verifiable using a validation process;
- *Availability:* availability of certificates, as well as the certificate verification system;
- *Authentication:* all participants, SecIU, School Registry Offices, students and employers must be identified and authenticated;
- *Authorization:* students must authorize companies or third parties who want their certificates to be validated and only students can see their certificates. Additionally, the SeCIU or the School Registry Offices must authorize changes in the access policy requested by the students. Finally, no person can access the information if it was not previously authorized by the system;

- *Auditing:* all accesses made, requested permissions and unauthorized access requests changes must be recorded;
- *Management:* the certificates are managed and stored by the SeCIU and the Gateway concentrates the interaction with the blockchain and off-chain.
- Finally as the *compliance* objective, the system processes must comply with GDPR requirements, like those related to responsibility for data processing, deletion and traceability of personal data and veracity and the consent of the treatment.

5.3.2 Threat risk analysis

We have used the Microsoft Threat Modeling Tool [OWA20] and STRIDE to identify threats and vulnerabilities of the designed system. We managed to derive 110 threats related to the STRIDE categories. In addition 18 threats related to either the use case described in [Say19] or to the use of blockchain and off-chain technology, as described in Section 2.6 were detected. From the application of the CNIL methodology [CNI12] we identified 10 traditional threats related to the processing of personal data. Out of the threats detected, six are considered not applicable, because either concern interactions with other universities, the manipulation of certificate translations or incorrect grading by the teaching body or School Registry Office, which are all cases not considered in the proposed solution. Finally, three, related to Denial of Service threats, were not mitigated. Figure 5.6 illustrates the diagram generated by the Microsoft Threat Modeling Tool.

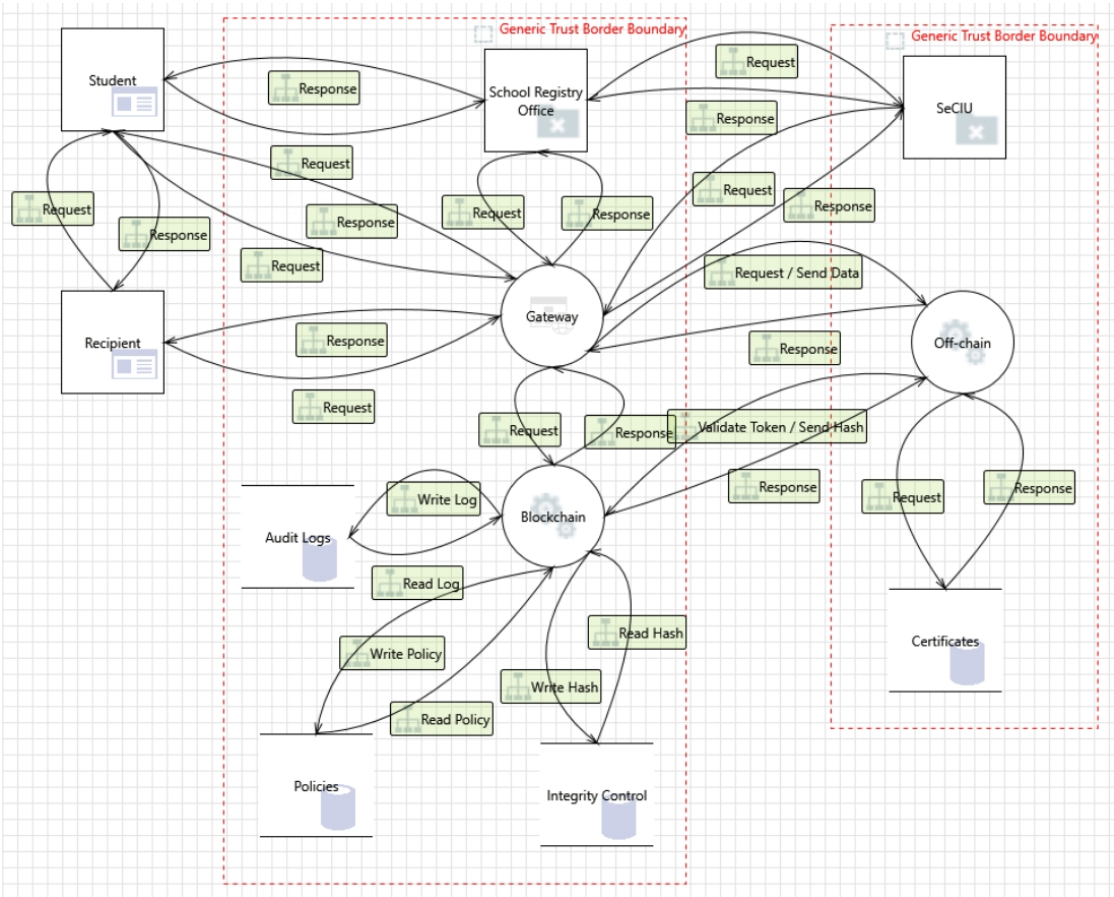

FIGURE 5.6: Threat risk analysis

In Table A.1 we classify and discuss the threats that are mitigated by design, which amount to the 56% of the total identified. The rest of the threats can be mitigated by implementing mechanisms that guarantee, among others, traffic encryption, protection of user credentials, control measures for personnel of School Registry Offices and the use of cryptographic hash functions following the recommendations to avoid hash reversibility we have provided in Section 3.6. All the analyzed threats are presented and discussed in detail in detail in Chapter A.

Threats mitigated by design		
Type	**Threat examples**	**Possible mitigations**
Spoofing	An adversary may spoof a user and gain access to the Web Application [Mic20].	Certificates are used to identify users.
Tampering	An adversary can modify the database and deny the action [Mic20]. Student adds certificate or altere certificates [Say19]. Blockchain security issues that affect integrity.	Authentication and authorization are validated by the access policy blockchain. Aditionally, the integrity control protects against unauthorized changes of certificates.
Repudiation	Attackers can deny malicious acts and remove the attacks footprints leading to repudiation issues [Mic20].	Authentication and authorization are validated by the access policy blockchain. Audit log and access policy stored in blockchain are immutable by design.
Information Disclosure	An adversary can gain access to sensitive data by performing SQL injection to certain pages or to the site as a whole [Mic20]. Blockchain confidentiality issues.	Certificates are stored off-chain with not direct access provided to users. Authentication and authorization are validated by the access policy blockchain. Design the generation of hashes in a way that makes reversibility a hard problem [LCT18]. Use permissioned blockchain and a commit-chain off-chain architecture [MBL20].
Elevation of Privileges	An adversary may bypass critical steps or perform actions on behalf of other users. An adversary can gain unauthorized access to the database [Mic20].	Authentication and authorization are validated by the access policy blockchain. Use permissioned blockchain and commit-chain off-chain architecture [MBL20].
Unavailability of legal processes	Lack of responsibility for the treatment of personal data. Lack of processes to ensure the veracity or correctness of the treatment. Failures in the information provided to the user [CNI12].	All operations are registered in the audit blockchain and this technology ensures by design the integrity of the audit and access policy. Use permissioned blockchain [MBL20].
Change in processing	Failures in the processes to ensure the veracity or purpose of the treatment. Deficiency of controls over the treatment or the processes to delete data [CNI12].	Information deletion is resolved by design, since the information is stored in the off-chain and there are specific use cases. All operations are registered in the audit blockchain and blockchain ensures by design integrity on audit logs and access policy.

TABLE 5.1: Threats mitigated by design

Chapter 6

Conclusion and further work

6.1 Conclusion

We have identified and discussed the challenges that the European and Uruguayan regulation on data protection pose to the design and implementation of software systems that manage personal data. In particular we are interested in those systems that are built using blockchain technology. To analyze this, we first conducted a study of blockchain and off-chain technologies, to understand the weaknesses and strengths of these solutions regarding personal data protection regulations.

The first conclusion we can make is that intrinsic characteristics of blockchain make in principle this technology incompatible with personal data protection regulations, so it is necessary to incorporate other components, such as the off-chain constructs we have presented and discussed in this thesis. On the other hand, blockchain technology meets by design some of the requirements of these personal data protection regulations, which helps to meet the requirements of security by design or privacy by default. In this sense, we propose to use blockchain to performs access control, audit and integrity control operations, while personal data is stored on an off-chain network.

As a result of the analysis we have carried out, we put forward a high-level system architecture to specify the behavior and services of a system that integrates blockchain and off-chain functionalities, complying with the requirements of the personal data protection regulations. The two main components of the system are a software architecture model and a use case model, which have been conceived to provide support for the construction of systems that are personal data protection regulations compliant by design. Based on these models we propose an architecture that incorporates some of the concepts presented, adding the integrity check functionality, even as a standalone functionality and we also propose some implementation considerations that should be taking into account, since there are requirements in personal data protection regulations that are not satisfied by design. We additionally propose a model that embodies 8 use cases, included the verification operation that does not require access to the data itself. This operation is of special importance for the completeness of the proof of concept proposed later. In this use cases we analyze the relationship that should exist between the owner, the data controller and the data processor, according to the personal data protection regulations requirements.

We have also put forward a methodology for performing security and privacy threat analysis of systems of that kind, combining STRIDE methodology and the risk methodology defined by CNIL for privacy risk management.

We illustrate the use of the proposed high-level system and the threat analysis methodology on a realistic and not trivial digital certificates system, intended to be used by SeCIU, who is interested in implementing a system that solves this problem, with blockchain technology. The design constructs are shown to be expressive enough

to specify the required functionalities of that system. In particular, as result of the analysis we have carried out we observe that about half of the threats detected are mitigated by design.

6.2 Future work

As future work it remains to complete the implementation of the prototype in order to assess the adequacy of the proposed high-level system architecture. An identified challenge is the conception of the mechanisms to register and authenticate the third parties that shall interact with the digital certificates system. In particular, as SECIU would prefer not to provide and manage those mechanisms, a decentralized solution must be considered. How well one such solution would integrate with the rest of system preserving the privacy requirements requires further study.

Related to the threat analysis methodology, it could be interesting to work on defining a set of generic threats related to blockchain and off-chain technology and personal data protection problems, as the threats analyzed are specific to the proof of concept use case.

Bibliography

[Age19] Agencia Española de Protección de Datos. "Introduction to the hash function as personal data pseudonymisation tecnhnique". In: *Agencia Española de Protección de Datos* (2019). URL: https://edps.europa.eu/data-protection/our-work/publications/papers/introduction-hash-function-personal-data_en.

[al.18] Rostand Costa et al. "Uso Não Financeiro de Blockchain: Um Estudo de Casao Sobre o Registro, Autenticação e Preservação de Documentos Digitais Acadêmicos". In: *Anais do I Workshop em Blockchain: Teoria, Tecnologias e Aplicações*. Campos do Jordão: SBC, 2018. URL: https://sol.sbc.org.br/index.php/wblockchain/article/view/2356.

[Art14] Article 29 Working Party. *Working Group on data protection - Opinion 05/2014 on anonymization techniques*. URL: https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/index_en.htm. 2014.

[bfa20] bfa.ar. *BLOCKCHAIN FEDERAL ARGENTINA*. URL: https://bfa.ar/. 2020.

[Cer] *Certificate Fraud News*. URL: https://www.montevideo.com.uy/Noticias/Siete-condenados-por-estafa-en-caso-de-venta-de-titulos-universitarios-falsos-en-Colonia-uc792929. 2021.

[Cho17] Usman W. Chohan. "The Double Spending Problem and Cryptocurrencies". In: *Information Systems & Economics eJournal* (2017). URL: https://ssrn.com/abstract=3090174.

[CNI12] CNIIL. *Methodology for Privacy Risk Management*. Edition 2012. 2012.

[Com18] Commission Nationale de l'Informatique et des Libertés. *Blockchain and the GDPR: Solutions for a responsible use of the blockchain in the context of personal data*. URL: https://www.cnil.fr/en/blockchain-and-gdpr-solutions-responsible-use-blockchain-context-personal-data. 2018.

[Dat17] Red Iberoamericana de Protección de Datos. *Estándares de protección de datos personales para los Estados Iberoamericanos*. URL: https://www.redipd.org/es/documentos/estandares-iberoamericanos. 2017.

[DE+21] Cristòfol Daudén-Esmel et al. "Lightweight Blockchain-based Platform for GDPR-Compliant Personal Data Management". In: *2021 IEEE 5th International Conference on Cryptography, Security and Privacy (CSP)*. 2021, pp. 68–73. DOI: 10.1109/CSP51677.2021.9357602.

[EDU20] EDUCTX.org. *EDUCTX – Blockchain Lab:UM*. URL: http://www.eductx.org/. 2020.

[EH18] Jacob Eberhardt and Jonathan Heiss. "Off-Chaining Models and Approaches to Off-Chain Computations". In: *Proceedings of the 2nd Workshop on Scalable and Resilient Infrastructures for Distributed Ledgers*. SERIAL'18. Rennes, France: Association for Computing Machinery, 2018, 7–12. ISBN: 9781450361101. DOI: 10.1145/3284764.3284766.

[Eip] *Ethereum Improvement Proposals (EIPs)*. URL: https://eips.ethereum.org/. 2021.

[Eni20] Enigma. *Enigma Project*. URL: https://blog.enigma.co/@EnigmaMPC. 2020.

[ET17] Jacob Eberhardt and Stefan Tai. "On or Off the Blockchain? Insights on Off-Chaining Computation and Data". In: *Service-Oriented and Cloud Computing*. Ed. by Flavio De Paoli, Stefan Schulte, and Einar Broch Johnsen. Cham: Springer International Publishing, 2017, pp. 3–15. ISBN: 978-3-319-67262-5.

[ET18] J. Eberhardt and S. Tai. "ZoKrates - Scalable Privacy-Preserving Off-Chain Computations". In: *2018 IEEE iThings and IEEE GreenCom and IEEE CPSCom and IEEE SmartData*. 2018, pp. 1084–1091.

[Eth] *Ethereum Wiki*. URL: https://eth.wiki/en/howto/smart-contract-safety. 2021.

[Eth21] Etherscan. *The Ethereum Blockchain Explorer*. URL: https://etherscan.io/. 2021.

[Eur16] European Parliament and of the council. "Regulation (EU) 2016/679". In: *Official Journal of the European Union* (2016). URL: https://eur-lex.europa.eu/eli/reg/2016/679/oj.

[Eur19] European Parliamentary Research Service. *Blockchain and the General Data Protection Regulation. Can distributed ledgers be squared with European data protection law?* URL: https://www.europarl.europa.eu/RegData/etudes/STUD/2019/634445/EPRS_STU(2019)634445_EN.pdf. 2019.

[Fel12] Edward Felten. *Does Hashing Make Data "Anonymous"?* URL: https://www.ftc.gov/news-events/blogs/techftc/2012/04/does-hashing-make-data-anonymous. 2012.

[FFM20] Gislaine Parra Freund, Priscila Basto Fagundes, and D. Macedo. "An Analysis of Blockchain and GDPR under the Data Lifecycle Perspective". In: *Mobile Networks and Applications* (2020), pp. 1–11.

[Gud+19] Lewis Gudgeon et al. "SoK: Off The Chain Transactions". In: *IACR Cryptol. ePrint Arch.* 2019 (2019), p. 360.

[Hyp20] Hyperledger.org. *Hyperledger*. URL: https://www.hyperledger.org/. 2020.

[IE18] Emin Gün Sirer Ittay Eyal. *Majority Is Not Enough: Bitcoin Mining Is Vulnerable*. URL: https://doi.org/10.1145/3212998. 2018.

[IOS18] Luis-Daniel Ibanez, Kieron O'Hara, and Elena Simperl. *On Blockchains and the General Data Protection Regulation*. Project Report. 2018. URL: https://eprints.soton.ac.uk/422879/.

[IPF20] IPFS. *IPFS*. URL: https://ipfs.io/. 2020.

[JT18] Christian Reitwießne Jason Teutsch. *A scalable verification solution for blockchain*. 2018.

[Kas17] Kaspersky. *Bitcoin Pools*. URL: https://www.kaspersky.com/blog/bitcoin-Blockchain-issues/18019. 2017.

[LCT18] Tom Lyons, Ludovic Courcelas, and Ken Timsit. "Blockchain and the GDPR. A thematic report prepared by the European Union Blockchain Observatory and Forumtitle". In: *European Union blockchain observatory and forum* (2018).

[LPX19] Chao Li, Balaji Palanisamy, and Runhua Xu. "Scalable and Privacy-Preserving Design of On/Off-Chain Smart Contracts". In: *35th IEEE International Conference on Data Engineering Workshops, ICDE Workshops 2019, Macao, China, April 8-12, 2019*. IEEE, 2019, pp. 7–12. DOI: 10.1109/ICDEW.2019.00-43. URL: https://doi.org/10.1109/ICDEW.2019.00-43.

[MBL20] Fernanda Molina, Gustavo Betarte, and Carlos Luna. *A Blockchain based and GDPR-compliant design of a system for digital education certificates*. URL: https://arxiv.org/abs/2010.12980. 2020.

[MBL21a] Fernanda Molina, Gustavo Betarte, and Carlos Luna. "Design principles for constructing GDPR-compliant blockchain solutions". In: *2021 IEEE/ACM 4th International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB), ICSE 2021*. 2021. DOI: 10.1109/WETSEB52558.2021.00008.

[MBL21b] Fernanda Molina, Gustavo Betarte, and Carlos Luna. "Privacy Aware Blockchain Solutions: Design and Threat Analysis". In: *XXIV Ibero-American Conference on Software Engineering, CibSE 2021*. Costa Rica, 2021. URL: https://cibse2021.citic.ucr.ac.cr/en/home/.

[Mic20] Microsoft. *Microsoft Threat Modeling Tool*. URL: https://www.microsoft.com/en-us/securityengineering/sdl/threatmodeling. 2020.

[Mol+18] Carlos Molina-Jiménez et al. "Implementation of Smart Contracts Using Hybrid Architectures with On and Off-Blockchain Components". In: *8th IEEE International Symposium on Cloud and Service Computing, SC2 2018, Paris, France, November 18-21, 2018*. IEEE, 2018, pp. 83–90. DOI: 10.1109/SC2.2018.00018. URL: https://doi.org/10.1109/SC2.2018.00018.

[NA16] Tiziana Cimoli Nicola Atzei Massimo Bartoletti. "A survey of attacks on Ethereum smart contracts". In: *Universita degli Studi di Cagliari, Cagliari, Italy* (2016).

[Nak09] Satoshi Nakamoto. *Bitcoin: A peer-to-peer electronic cash system*. 2009. URL: http://www.bitcoin.org/bitcoin.pdf.

[OWA20] OWASP. *OWASP Threat Modeling*. URL: https://owasp.org/www-pdf-archive/AdvancedThreatModeling.pdf. 2020.

[PKR20] Wolfgang Prinz, Sabine Kolvenbach, and Rudolf Ruland. "Blockchain for Education: Lifelong Learning Passport". In: *ERCIM News* 2020.120 (2020). URL: https://ercim-news.ercim.eu/en120/special/blockchain-for-education-lifelong-learning-passport.

[Rei16] Christian Reitwiener. *zkSNARKs in a Nutshell*. URL: https://blog.ethereum.org/2016/12/05/zksnarks-in-a-nutshell/. 2016.

[Rep08] República Oriental del Uruguay. *Ley 18.331: Protección de datos personales y "Habeas Data"*. URL: https://www.impo.com.uy/bases/Leyes/18331-2008. 2008.

[Rep18] República Oriental del Uruguay. *Ley 19.670: Aprobación de rendición de cuentas y balance de ejecución presupuestal. Ejercicio 2017*. URL: https://www.impo.com.uy/bases/Leyes/19670-2018. 2018.

[Rep20] República Oriental del Uruguay. *Decreto 64.2020*. URL: https://www.impo.com.uy/bases/decretos/64-2020. 2020.

[Say19] Rakibul Hasan Saye. *Potential of Blockchain technology to solve fake diploma problem*. URL: https://jyx.jyu.fi/bitstream/handle/123456789/64817/1/URN:NBN:fi:jyu-201906253406.pdf. 2019.

[SB17] Prakash Santhana and Abhishek Biswa. *Blockchain risk management – Risk functions need to play an active role in shaping blockchain strategy*. URL: https://www2.deloitte.com/content/dam/Deloitte/us/Documents/financial-services/us-fsi-blockchain-risk-management.pdf. 2017.

[STR20] STRIDE. *STRIDE Methodology*. URL: https://docs.microsoft.com/en-us/archive/msdn-magazine/2006/november/uncover-security-design-flaws-using-the-stride-approach. 2020.

[SU18] Blockstream Stanford University University College London. *Bulletproofs: Short Proofs for Confidential Transactions and More*. 2018.

[SWA20] SWARM. *SWARM*. URL: https://www.swarm.fund/. 2020.

[TGN20] Unal Tatar, Yasir Gokce, and Brian Nussbaum. "Law versus technology: Blockchain, GDPR, and tough tradeoffs". In: *Computer Law & Security Review* 38 (2020), p. 105454. ISSN: 0267-3649. DOI: https://doi.org/10.1016/j.clsr.2020.105454. URL: http://www.sciencedirect.com/science/article/pii/S0267364920300595.

[Tjc] *Digital Rights Ireland Ltd, cases C-293/12 y C-594/1*. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A62012CJ0293. 2014.

[Tru+19] Nguyen Binh Truong et al. "GDPR-Compliant Personal Data Management: A Blockchain-based Solution". In: *CoRR* abs/1904.03038 (2019). URL: http://arxiv.org/abs/1904.03038.

[URC18] URCDP. *Dictum 9-2018, Exp. 2017-2-10-000394*. URL: https://www.gub.uy/unidad-reguladora-control-datos-personales/sites/unidad-reguladora-control-datos-personales/files/documentos/normativa/Dictamen%2B9%2B2018_0.pdf. 2018.

[Whi20] Whisper. *Whisper*. URL: https://geth.ethereum.org/docs/whisper/whisper-overview. 2020.

[Wik20] Bitcoin Wiki. *Bitcoin Weaknesses*. URL: https://en.bitcoin.it/wiki/Weaknesses. 2020.

[Woo17] Gavin Wood. *Ethereum: A secure decentralised generalised transaction ledger EIP-150 REVISION (759dccd - 2017-08-07)*. Publication Title: Ethereum Project Yellow Paper. 2017. URL: https://ethereum.github.io/yellowpaper/paper.pdf.

[Zha+20] Fan Zhang et al. "The Ekiden Platform for Confidentiality-Preserving, Trustworthy, and Performant Smart Contracts". In: *IEEE Secur. Priv.* 18.3 (2020), pp. 17–27. DOI: 10.1109/MSEC.2020.2976984. URL: https://doi.org/10.1109/MSEC.2020.2976984.

Appendix A

Threats analysis of the academic certificates system

In Table A.1 we classify and discuss the threats that are mitigated by design, which amount to the 56% of the total identified. Most of the rest of the threats can be mitigated by implementing mechanisms that guarantee, among others, traffic encryption, protection of user credentials, control measures for personnel of School Registry Offices and the use of cryptographic hash functions. Those threats and the corresponding controls are presented in Table A.2.

Out of the threats detected, six are considered not applicable and are described in Table A.3. Three unmitigated threats that concern Denial of Service are described in Table A.4.

A.1 Threats mitigated by design

Threats mitigated by design				
Id	**Category**	**Title**	**Description**	**Possible mitigations**
1	CNIL - Change in processing	Failures in the processes to ensure the veracity of the treatment	Changes in the process affect the veracity of the information processed	The processing of the information, as well as the accesses made and permits granted, are reflected in the blockchain
2	CNIL - Change in processing	Failures in the processes to ensure the purpose of the treatment	The information is not processed according to the defined purpose	The processing of the information, as well as the accesses made and permits granted, are reflected in the blockchain.
3	CNIL - Change in processing	Deficiency of controls over the treatment	The treatment of the information is not as expected, due to lack of controls over it.	The processing of the information, as well as the accesses made and permits granted, are reflected in the blockchain.
4	CNIL - Change in processing	Failures in the processes to delete data	Claims because the information is not deleted correctly, in accordance with the agreed treatment of the information	It is considered as a use case that the student can request the deletion of information
5	CNIL - Unavailability of Legal Processes	Lack of responsibility for the treatment of personal data	No one assumes responsibility for a claim by a student for the treatment of their information	In permissioned blockchain the participants of the chain can be considered responsible for the treatment of the information
6	CNIL - Unavailability of Legal Processes	Lack of processes to ensure the veracity of the treatment	In a claim, the veracity ot the treatment cannot be demonstrated	All operations, successful or not, are registered in the audit blockchain. Blockchain ensures by design the integrity of the audit and access policy

7	CNIL - Unavailability of Legal Processes	Lack of processes for the correct treatment of data	In a claim, the correct treatment of the information cannot be demonstrated	All operations, successful or not, are registered in the audit blockchain. Blockchain ensures by design the integrity of the audit and access policy
8	CNIL - Unavailability of Legal Processes	Failures in the information provided to the user	Incorrect information is provided to the student, such as access information to their data	All operations, successful or not, are registered in the audit blockchain. Blockchain ensures by design the integrity of the audit log
9	CNIL - Unavailability of Legal Processes	Lack of processes to delete data	Claims because the information is not deleted, in accordance with the agreed treatment of the information	It is considered as a use case that the student can request the deletion of information
10	Elevation of Privileges	An adversary may bypass critical steps or perform actions on behalf of other users (victims) due to improper validation logic	Failure to restrict the privileges and access rights to the application to individuals who require the privileges or access rights may result into unauthorized use of data due to inappropriate rights settings and validation.	Before allowing any access, the access policies associated with the user are validated on the Blockchain.
11	Elevation of Privileges	An adversary can gain unauthorized access to database due to loose authorization rules	Database access should be configured with roles and privilege based on least privilege and need to know principle.	Access is done through the Gateway, after authentication and privilege verification.
12	Elevation of Privileges	An adversary can gain unauthorized access to database due to lack of network access protection	If there is no restriction at network or host firewall level, to access the database then anyone can attempt to connect to the database from an unauthorized location	It is a private blockchain network, where the participating nodes are the School Registry Offices and the SeCIU.
13	Elevation of Privileges	An adversary can gain unauthorized access to database due to loose authorization rules	Database access should be configured with roles and privilege based on least privilege and need to know principle.	Access is done through the Gateway, after authentication and privilege verification.
14	Elevation of Privileges	An adversary can gain unauthorized access to database due to lack of network access protection	If there is no restriction at network or host firewall level, to access the database then anyone can attempt to connect to the database from an unauthorized location	It is a private blockchain network, where the participating nodes are the School Registry Offices and the SeCIU.
15	Elevation of Privileges	An adversary can gain unauthorized access to database due to lack of network access protection	If there is no restriction at network or host firewall level, to access the database then anyone can attempt to connect to the database from an unauthorized location	It is a private blockchain network, where the participating nodes are the School Registry Offices and the SeCIU.
16	Elevation of Privileges	An adversary can gain unauthorized access to database due to loose authorization rules	Database access should be configured with roles and privilege based on least privilege and need to know principle.	Access is done through the Gateway, after authentication and privilege verification.
17	Elevation of Privileges	An adversary may bypass critical steps or perform actions on behalf of other users (victims) due to improper validation logic	Failure to restrict the privileges and access rights to the application to individuals who require the privileges or access rights may result into unauthorized use of data due to inappropriate rights settings and validation.	Before allowing any access, the access policies associated with the user are validated on the Blockchain.
18	Elevation of Privileges	An adversary can gain unauthorized access to database due to lack of network access protection	If there is no restriction at network or host firewall level, to access the database then anyone can attempt to connect to the database from an unauthorized location	It is a commit Chain Off chain, located in SeCIU facilities, without direct public access.

19	Elevation of Privileges	An adversary can gain unauthorized access to database due to loose authorization rules	Database access should be configured with roles and privilege based on least privilege and need to know principle.	Access to data is made after validation of a token sent by the Blockchain.
20	Information Disclosure	An adversary can gain access to certain pages or the site as a whole.	Robots.txt is often found in site's root directory and exists to regulate the bots that crawl your site. This is where you can grant or deny permission to all or some specific search engine robots to access certain pages or your site as a whole.	Sensitive information is located in the blockchain and off chain networks, not in the Gateway.
21	Information Disclosure	An adversary can gain access to sensitive data by performing SQL injection	SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. The primary form of SQL injection consists of direct insertion of code into user-input variables that are concatenated with SQL commands and executed. A less direct attack injects malicious code into strings that are destined for storage in a table or as metadata. When the stored strings are subsequently concatenated into a dynamic SQL command, the malicious code is executed.	Access to data is made after validation of a token sent by the blockchain. Users do not have direct access to off-chain.
22	Information Disclosure	An adversary can gain access to sensitive PII or HBI data in database	Additional controls like Transparent Data Encryption, Column Level Encryption, EKM etc. provide additional protection mechanism to high value PII or HBI data.	No personal information is stored on the blockchain.
23	Information Disclosure	An adversary can gain access to sensitive data by performing SQL injection	SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. The primary form of SQL injection consists of direct insertion of code into user-input variables that are concatenated with SQL commands and executed. A less direct attack injects malicious code into strings that are destined for storage in a table or as metadata. When the stored strings are subsequently concatenated into a dynamic SQL command, the malicious code is executed.	Access to data is made after validation of a token sent by the blockchain. Users do not have direct access to off-chain.
24	Information Disclosure	An adversary can gain access to sensitive PII or HBI data in database	Additional controls like Transparent Data Encryption, Column Level Encryption, EKM etc. provide additional protection mechanism to high value PII or HBI data.	No personal information is stored on the blockchain.
25	Information Disclosure	An adversary can gain access to sensitive PII or HBI data in database	Additional controls like Transparent Data Encryption, Column Level Encryption, EKM etc. provide additional protection mechanism to high value PII or HBI data.	Personal information is stored hashed

26	Information Disclosure	An adversary can gain access to sensitive data by performing SQL injection	SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. The primary form of SQL injection consists of direct insertion of code into user-input variables that are concatenated with SQL commands and executed. A less direct attack injects malicious code into strings that are destined for storage in a table or as metadata. When the stored strings are subsequently concatenated into a dynamic SQL command, the malicious code is executed.	Access to data is made after validation of a token sent by the blockchain. Users do not have direct access to off-chain.
27	Information Disclosure	An adversary can gain access to certain pages or the site as a whole.	Robots.txt is often found in your site's root directory and exists to regulate the bots that crawl your site. This is where you can grant or deny permission to all or some specific search engine robots to access certain pages or your site as a whole. The standard for this file was developed in 1994 and is known as the Robots Exclusion Standard or Robots Exclusion Protocol. Detailed info about the robots.txt protocol can be found at robotstxt.org.	Sensitive information is located in the blockchain and off chain networks, not in the Gateway.
28	Information Disclosure	An adversary can gain access to sensitive PII or HBI data in database	Additional controls like Transparent Data Encryption, Column Level Encryption, EKM etc. provide additional protection mechanism to high value PII or HBI data.	The off chain is private without public access, only through the Gateway. Access to information is done only after verifying the token on the blockchain.
29	Information Disclosure	Blockchain Security Issues - Confidentiality	The information stored in the Blockchain is seen by all the nodes of the network.	Certificates are not stored on the Blockchain network. Hashes are generated in a way that makes reversibility difficult.
30	Repudiation	Attacker can deny the malicious act and remove the attack foot prints leading to repudiation issues	Proper logging of all security events and user actions builds traceability in a system and denies any possible repudiation issues. In the absence of proper auditing and logging controls, it would become impossible to implement any accountability in a system	All operations, successful or not, are registered in the audit blockchain
31	Repudiation	Attacker can deny the malicious act and remove the attack foot prints leading to repudiation issues	Proper logging of all security events and user actions builds traceability in a system and denies any possible repudiation issues. In the absence of proper auditing and logging controls, it would become impossible to implement any accountability in a system	All operations, successful or not, are registered in the audit blockchain
32	Repudiation	Attacker can deny the malicious act and remove the attack foot prints leading to repudiation issues	Proper logging of all security events and user actions builds traceability in a system and denies any possible repudiation issues. In the absence of proper auditing and logging controls, it would become impossible to implement any accountability in a system	All operations, successful or not, are registered in the audit blockchain

33	Repudiation	Attacker can deny the malicious act and remove the attack foot prints leading to repudiation issues	Proper logging of all security events and user actions builds traceability in a system and denies any possible repudiation issues. In the absence of proper auditing and logging controls, it would become impossible to implement any accountability in a system	All operations, successful or not, are registered in the audit blockchain
34	Repudiation	An adversary can deny actions on database due to lack of auditing	Proper logging of all security events and user actions builds traceability in a system and denies any possible repudiation issues. In the absence of proper auditing and logging controls, it would become impossible to implement any accountability in a system.	Solved by design on blockchain. In the solution, data accesses to policies blockchain are recorded in the audit blockchain.
35	Repudiation	An adversary can deny actions on database due to lack of auditing	Proper logging of all security events and user actions builds traceability in a system and denies any possible repudiation issues. In the absence of proper auditing and logging controls, it would become impossible to implement any accountability in a system.	Solved by design on blockchain.
36	Repudiation	An adversary can deny actions on database due to lack of auditing	Proper logging of all security events and user actions builds traceability in a system and denies any possible repudiation issues. In the absence of proper auditing and logging controls, it would become impossible to implement any accountability in a system.	Solved by design on blockchain. In the solution, data accesses to integrity control are recorded in the audit blockchain.
37	Repudiation	Attacker can deny the malicious act and remove the attack foot prints leading to repudiation issues	Proper logging of all security events and user actions builds traceability in a system and denies any possible repudiation issues. In the absence of proper auditing and logging controls, it would become impossible to implement any accountability in a system	All operations, successful or not, are registered in the audit blockchain
38	Repudiation	An adversary can deny actions on database due to lack of auditing	Proper logging of all security events and user actions builds traceability in a system and denies any possible repudiation issues. In the absence of proper auditing and logging controls, it would become impossible to implement any accountability in a system.	Ensure that login auditing is enabled. In the solution, data accesses in the off chain are recorded in the audit blockchain.
39	Repudiation	Attacker can deny the malicious act and remove the attack foot prints leading to repudiation issues	Proper logging of all security events and user actions builds traceability in a system and denies any possible repudiation issues. In the absence of proper auditing and logging controls, it would become impossible to implement any accountability in a system	All operations, successful or not, are registered in the audit blockchain
40	Spoofing	An adversary may spoof SeCIU and gain access to Web Application	If proper authentication is not in place, an adversary can spoof a source process or external entity and gain unauthorized access to the Web Application	Certificates are used to identify SeCIU

41	Spoofing	An adversary may spoof School Registry Offices and gain access to Web Application	If proper authentication is not in place, an adversary can spoof a source process or external entity and gain unauthorized access to the Web Application	Certificates are used to identify School Registry Offices
42	Spoofing	An adversary may spoof Blockchain and gain access to Web Application	If proper authentication is not in place, an adversary can spoof a source process or external entity and gain unauthorized access to the Web Application	Certificates are used to access to Blockchain
43	Spoofing	An adversary can create a fake website and launch phishing attacks	Phishing is attempted to obtain sensitive information such as usernames, passwords, and credit card details (and sometimes, indirectly, money), often for malicious reasons, by masquerading as a Web Server which is a trustworthy entity in electronic communication	It is a private Blockchain network
44	Spoofing	An adversary may spoof Browser and gain access to Web Application	If proper authentication is not in place, an adversary can spoof a source process or external entity and gain unauthorized access to the Web Application	Certificates are used to identify students
45	Spoofing	An adversary may spoof Browser and gain access to Web Application	If proper authentication is not in place, an adversary can spoof a source process or external entity and gain unauthorized access to the Web Application	To authenticate Recipients are used standard authentication mechanism or certificates
46	Spoofing	An adversary can create a fake website and launch phishing attacks	Phishing is attempted to obtain sensitive information such as usernames, passwords, and credit card details (and sometimes, indirectly, money), often for malicious reasons, by masquerading as a Web Server which is a trustworthy entity in electronic communication	It is a private network
47	Spoofing	An adversary may spoof Offchain and gain access to Web Application	If proper authentication is not in place, an adversary can spoof a source process or external entity and gain unauthorized access to the Web Application	Certificates are used to access the offchain
48	Tampering	An adversary can gain access to sensitive data by performing SQL injection through Web App	SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. The primary form of SQL injection consists of direct insertion of code into user-input variables that are concatenated with SQL commands and executed. A less direct attack injects malicious code into strings that are destined for storage in a table or as metadata. When the stored strings are subsequently concatenated into a dynamic SQL command, the malicious code is executed.	Access to data is made after validation of a token sent by the blockchain. Users do not have direct access to offchain.
49	Tampering	An adversary can gain access to sensitive data stored in Web App's config files	An adversary can gain access to the config files. and if sensitive data is stored in it, it would be compromised.	It is a private network

50	Tampering	An adversary can gain access to sensitive data by performing SQL injection through Web App	SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. The primary form of SQL injection consists of direct insertion of code into user-input variables that are concatenated with SQL commands and executed. A less direct attack injects malicious code into strings that are destined for storage in a table or as metadata. When the stored strings are subsequently concatenated into a dynamic SQL command, the malicious code is executed.	Access to data is made after validation of a token sent by the blockchain. Users do not have direct access to offchain.
51	Tampering	An adversary can gain access to sensitive data by performing SQL injection through Web App	SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. The primary form of SQL injection consists of direct insertion of code into user-input variables that are concatenated with SQL commands and executed. A less direct attack injects malicious code into strings that are destined for storage in a table or as metadata. When the stored strings are subsequently concatenated into a dynamic SQL command, the malicious code is executed.	Access to data is made after validation of a token sent by the blockchain. Users do not have direct access to offchain.
52	Tampering	An adversary can gain access to sensitive data by performing SQL injection through Web App	SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. The primary form of SQL injection consists of direct insertion of code into user-input variables that are concatenated with SQL commands and executed. A less direct attack injects malicious code into strings that are destined for storage in a table or as metadata. When the stored strings are subsequently concatenated into a dynamic SQL command, the malicious code is executed.	Access to data is made after validation of a token sent by the blockchain. Users do not have direct access to offchain.
53	Tampering	An adversary can gain access to sensitive data by performing SQL injection through Web App	SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. The primary form of SQL injection consists of direct insertion of code into user-input variables that are concatenated with SQL commands and executed. A less direct attack injects malicious code into strings that are destined for storage in a table or as metadata. When the stored strings are subsequently concatenated into a dynamic SQL command, the malicious code is executed.	Access to data is made after validation of a token sent by the blockchain. Users do not have direct access to offchain.

54	Tampering	An adversary may leverage the lack of monitoring systems and trigger anomalous traffic to database	An adversary may leverage the lack of intrusion detection and prevention of anomalous database activities and trigger anomalous traffic to database	Access to data is made after validation of a token sent by the blockchain. Users do not have direct access to off-chain. All operations, successful or not, are registered in the audit blockchain, that is inmutable by desing.
55	Tampering	An adversary can tamper critical database securables and deny the action	An adversary can tamper critical database securables and deny the action	This problem is solved by Blockchain design
56	Tampering	An adversary may leverage the lack of monitoring systems and trigger anomalous traffic to database	An adversary may leverage the lack of intrusion detection and prevention of anomalous database activities and trigger anomalous traffic to database	This problem is solved by Blockchain design
57	Tampering	An adversary can tamper critical database securables and deny the action	An adversary can tamper critical database securables and deny the action	This problem is solved by Blockchain design
58	Tampering	An adversary can tamper critical database securables and deny the action	An adversary can tamper critical database securables and deny the action	This problem is solved by Blockchain design
59	Tampering	An adversary may leverage the lack of monitoring systems and trigger anomalous traffic to database	An adversary may leverage the lack of intrusion detection and prevention of anomalous database activities and trigger anomalous traffic to database	This problem is solved by Blockchain design
60	Tampering	An adversary can tamper critical database securables and deny the action	An adversary can tamper critical database securables and deny the action	If a data is modified, the integrity control stored in the Blockchain will indicate a difference. Modification of off-chain data is only done after validating the access token provided by the blockchain netwok.
61	Tampering	An adversary may leverage the lack of monitoring systems and trigger anomalous traffic to database	An adversary may leverage the lack of intrusion detection and prevention of anomalous database activities and trigger anomalous traffic to database	If a data is modified, the integrity control stored in the Blockchain will indicate a difference. Modification of off-chain data is only done after validating the access token provided by the blockchain network.
62	Tampering	An adversary can gain access to sensitive data by performing SQL injection through Web App	SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. The primary form of SQL injection consists of direct insertion of code into user-input variables that are concatenated with SQL commands and executed. A less direct attack injects malicious code into strings that are destined for storage in a table or as metadata. When the stored strings are subsequently concatenated into a dynamic SQL command, the malicious code is executed.	Access to data is made after validation of a token sent by the blockchain. Users do not have direct access to off-chain.
63	Tampering	An adversary can gain access to sensitive data stored in Web App's config files	An adversary can gain access to the config files. and if sensitive data is stored in it, it would be compromised.	It is a private network

64	Tampering	Fraudulent activity 1 [Say19]	Student adds certificate to his/her CV although (s)he did not attend or complete the degree there (Grolleau et al., 2008).	Certificate can be verified using blockchain.
65	Tampering	Fraudulent Activity 2 [Say19]	Student makes a counterfeit paper copy of the original paper copy (Grolleau et al., 2008) by using similar paper and special security features used in the original paper certificate ("Sharadeshe jal sonoder," 2015; Lancaster, 2017).	Certificate can be verified using blockchain.
66	Tampering	Fraudulent Activity 5 [Say19]	Student hacks the university grading system and change grades	When changing the certificate, the hash stored in the blockchain would no longer be valid, so it could not be validated.
67	Tampering	Fraudulent Activity 10 [Say19]	Corrupt officials change the students' academic information like enrolment date, course completion date and grade in study data management system to prove the validity of the counterfeit certificate (Ekushey Television – ETV, 2014; Mir-Jabbar, 2017).	This problem is solved by Blockchain design
68	Tampering	Blockchain Security Issues - Immutability	The data stored in the Blockchain cannot be modified, in case it needs to be rectified.	Certificate modifications are made in the off chain and the hash is stored back on the blockchain. In the Blockchain, policy records or hashes that change are discarded, leaving the last record in effect.
69	Tampering	Blockchain Security Issues - Unpredictable state	When a user sends a transaction to the network to invoke a contract, he cannot be sure of the state in which the contract will be when the transaction is executed.	This does not happen in permissioned Blockchain.
70	Tampering	Blockchain Security Issues - Transaction security issues	Risk of double spending - Process something more than one time.	This does not happen in permissioned Blockchain.
71	Tampering	Off chain Blockchain Security Issues - Integrity	The information stored in the off chain could be altered without being detected.	If a data is modified, the integrity control stored in the Blockchain will indicate a difference. he off chain network must be protected against internal access without going through the Gateway.
72	Tampering	Off chain Blockchain Security Issues - Traceability	In off chain audit is not resolved by design, therefore, accesses or modifications could be made without leaving a trace.	All accesses are reported to the Blockchain, who keeps the audit record

TABLE A.1: Threats mitigated by design

A.2 Threats mitigated by implementation

Threats mitigated by implementation				
Id	**Category**	**Title**	**Description**	**Possible mitigations**
73	CNIL - Unavailability of Legal Processes	Lack of processes to ensure the consent of the treatment	In a claim, it cannot be demonstrated that consent is given to the processing of the information	The process must include the student's request for consent when registering their data.
74	Information Disclosure	An adversary can reverse weakly encrypted or hashed content	An adversary can reverse weakly encrypted or hashed content	For certificate validation use only approved cryptographic hash functions, following hash structure recommendations to avoid reversibility. Communications are encrypted and authenticated. Use only approved cryptographic hash functions. Verify X.509 certificates used to authenticate SSL, TLS, and DTLS connections.
75	Information Disclosure	An adversary can gain access to sensitive information through error messages	An adversary can gain access to sensitive data such as the following, through verbose error messages - Server names - Connection strings - Usernames - Passwords - SQL procedures - Details of dynamic SQL failures - Stack trace and lines of code - Variables stored in memory - Drive and folder locations - Application install points - Host configuration settings - Other internal application details	Do not expose security details in error messages. Implement Default error handling page. Exceptions should fail safely.
76	Information Disclosure	An adversary may gain access to sensitive data from log files	An adversary may gain access to sensitive data from log files	Ensure that the application does not log sensitive user data. Ensure that Audit and Log Files have Restricted Access.
77	Information Disclosure	An adversary can reverse weakly encrypted or hashed content	An adversary can reverse weakly encrypted or hashed content	For certificate validation use only approved cryptographic hash functions, following hash structure recommendations to avoid reversibility. Communications are encrypted and authenticated. Use only approved cryptographic hash functions. Verify X.509 certificates used to authenticate SSL, TLS, and DTLS connections.
78	Information Disclosure	An adversary can gain access to sensitive information through error messages	An adversary can gain access to sensitive data such as the following, through verbose error messages - Server names - Connection strings - Usernames - Passwords - SQL procedures - Details of dynamic SQL failures - Stack trace and lines of code - Variables stored in memory - Drive and folder locations - Application install points - Host configuration settings - Other internal application details	Do not expose security details in error messages. Implement Default error handling page. Exceptions should fail safely.
79	Information Disclosure	An adversary may gain access to sensitive data from log files	An adversary may gain access to sensitive data from log files	Ensure that the application does not log sensitive user data. Ensure that Audit and Log Files have Restricted Access.

80	Information Disclosure	An adversary can reverse weakly encrypted or hashed content	An adversary can reverse weakly encrypted or hashed content	For certificate validation use only approved cryptographic hash functions, following hash structure recommendations to avoid reversibility. Communications are encrypted and authenticated. Use only approved cryptographic hash functions. Verify X.509 certificates used to authenticate SSL, TLS, and DTLS connections.
81	Information Disclosure	An adversary can gain access to sensitive information through error messages	An adversary can gain access to sensitive data such as the following, through verbose error messages - Server names - Connection strings - Usernames - Passwords - SQL procedures - Details of dynamic SQL failures - Stack trace and lines of code - Variables stored in memory - Drive and folder locations - Application install points - Host configuration settings - Other internal application details	Do not expose security details in error messages. Implement Default error handling page. Exceptions should fail safely.
82	Information Disclosure	An adversary may gain access to sensitive data from log files	An adversary may gain access to sensitive data from log files	Ensure that the application does not log sensitive user data. Ensure that Audit and Log Files have Restricted Access.
83	Information Disclosure	An adversary may gain access to sensitive data from uncleared browser cache	An adversary may gain access to sensitive data from uncleared browser cache	Ensure that sensitive content is not cached on the browser.
84	Information Disclosure	An adversary can gain access to sensitive information through error messages	An adversary can gain access to sensitive data such as the following, through verbose error messages - Server names - Connection strings - Usernames - Passwords - SQL procedures - Details of dynamic SQL failures - Stack trace and lines of code - Variables stored in memory - Drive and folder locations - Application install points - Host configuration settings - Other internal application details	Do not expose security details in error messages. Implement Default error handling page. Exceptions should fail safely.
85	Information Disclosure	An adversary can gain access to sensitive data by sniffing traffic to Web Application	An adversary may conduct man in the middle attack and downgrade TLS connection to clear text protocol, or forcing browser communication to pass through a proxy server that he controls. This may happen because the application may use mixed content or HTTP Strict Transport Security policy is not ensured.	Applications available over HTTPS must use secure cookies.
86	Information Disclosure	An adversary may gain access to unmasked sensitive data such as certificates	An adversary may gain access to unmasked sensitive data such as certificates.	Ensure that sensitive data displayed on the user screen is masked. The information provided must be encrypted with the student's public key.
87	Information Disclosure	An adversary may gain access to sensitive data from log files	An adversary may gain access to sensitive data from log files	Ensure that the application does not log sensitive user data. Ensure that Audit and Log Files have Restricted Access.

88	Information Disclosure	An adversary can reverse weakly encrypted or hashed content	An adversary can reverse weakly encrypted or hashed content	For certificate validation use only approved cryptographic hash functions, following hash structure recommendations to avoid reversibility. Communications are encrypted and authenticated. Use only approved cryptographic hash functions. Verify X.509 certificates used to authenticate SSL, TLS, and DTLS connections.
89	Information Disclosure	An adversary may gain access to sensitive data from uncleared browser cache	An adversary may gain access to sensitive data from uncleared browser cache	Ensure that sensitive content is not cached on the browser.
90	Information Disclosure	An adversary can gain access to sensitive information through error messages	An adversary can gain access to sensitive data such as the following, through verbose error messages - Server names - Connection strings - Usernames - Passwords - SQL procedures - Details of dynamic SQL failures - Stack trace and lines of code - Variables stored in memory - Drive and folder locations - Application install points - Host configuration settings - Other internal application details	Do not expose security details in error messages. Implement Default error handling page. Exceptions should fail safely.
91	Information Disclosure	An adversary can gain access to sensitive data by sniffing traffic to Web Application	An adversary may conduct man in the middle attack and downgrade TLS connection to clear text protocol, or forcing browser communication to pass through a proxy server that he controls. This may happen because the application may use mixed content or HTTP Strict Transport Security policy is not ensured.	Applications available over HTTPS must use secure cookies.
92	Information Disclosure	An adversary may gain access to unmasked sensitive data such as credit card numbers	An adversary may gain access to unmasked sensitive data such as credit card numbers	Ensure that sensitive data displayed on the user screen is masked. The information provided must be encrypted with the student's public key. Due to access policies, Recipients access is restricted to the validation of certificates.
93	Information Disclosure	An adversary may gain access to sensitive data from log files	An adversary may gain access to sensitive data from log files	Ensure that the application does not log sensitive user data. Ensure that Audit and Log Files have Restricted Access.
94	Information Disclosure	An adversary can reverse weakly encrypted or hashed content	An adversary can reverse weakly encrypted or hashed content	For certificate validation use only approved cryptographic hash functions, following hash structure recommendations to avoid reversibility. Communications are encrypted and authenticated. Use only approved cryptographic hash functions. Verify X.509 certificates used to authenticate SSL, TLS, and DTLS connections.

95	Information Disclosure	An adversary can reverse weakly encrypted or hashed content	An adversary can reverse weakly encrypted or hashed content	For certificate validation use only approved cryptographic hash functions, following hash structure recommendations to avoid reversibility. Communications are encrypted and authenticated. Use only approved cryptographic hash functions. Verify X.509 certificates used to authenticate SSL, TLS, and DTLS connections.
96	Information Disclosure	An adversary may gain access to sensitive data from log files	An adversary may gain access to sensitive data from log files	Ensure that the application does not log sensitive user data. Ensure that Audit and Log Files have Restricted Access.
97	Information Disclosure	An adversary can gain access to sensitive information through error messages	An adversary can gain access to sensitive data such as the following, through verbose error messages - Server names - Connection strings - Usernames - Passwords - SQL procedures - Details of dynamic SQL failures - Stack trace and lines of code - Variables stored in memory - Drive and folder locations - Application install points - Host configuration settings - Other internal application details	Do not expose security details in error messages. Implement Default error handling page. Exceptions should fail safely.
98	Information Disclosure	An adversary can gain access to sensitive data by performing SQL injection	SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. The primary form of SQL injection consists of direct insertion of code into user-input variables that are concatenated with SQL commands and executed. A less direct attack injects malicious code into strings that are destined for storage in a table or as metadata. When the stored strings are subsequently concatenated into a dynamic SQL command, the malicious code is executed.	Ensure that login auditing is enabled on SQL Server. Ensure that least-privileged accounts are used to connect to Database server.
99	Spoofing	An adversary can create a fake website and launch phishing attacks	Phishing is attempted to obtain sensitive information such as usernames, passwords, and credit card details (and sometimes, indirectly, money), often for malicious reasons, by masquerading as a Web Server which is a trustworthy entity in electronic communication	Verify X.509 certificates used to authenticate SSL, TLS, and DTLS connections. Validate all redirects within the application are closed or done safely.

100	Spoofing	An adversary can steal sensitive data like user credentials	Attackers can exploit weaknesses in system to steal user credentials. Downstream and upstream components are often accessed by using credentials stored in configuration stores. Attackers may steal the upstream or downstream component credentials. Attackers may steal credentials if, Credentials are stored and sent in clear text, Weak input validation coupled with dynamic sql queries, Password retrieval mechanism are poor,	Explicitly disable the autocomplete HTML attribute in sensitive forms and inputs. Perform input validation and filtering on all string type Model properties. Validate all redirects within the application are closed or done safely. Enable step up or adaptive authentication. Implement forgot password functionalities securely. Ensure that password and account policy are implemented. Implement input validation on all string type parameters accepted by Controller methods. Use private public key certificates
101	Spoofing	An adversary can spoof the target web application due to insecure TLS certificate configuration	Ensure that TLS certificate parameters are configured with correct values	Verify X.509 certificates used to authenticate SSL, TLS, and DTLS connections.
102	Spoofing	An adversary can steal sensitive data like user credentials	Attackers can exploit weaknesses in system to steal user credentials. Downstream and upstream components are often accessed by using credentials stored in configuration stores. Attackers may steal the upstream or downstream component credentials. Attackers may steal credentials if, Credentials are stored and sent in clear text, Weak input validation coupled with dynamic sql queries, Password retrieval mechanism are poor,	Use private public key certificates.
103	Spoofing	An adversary can spoof the target web application due to insecure TLS certificate configuration	Ensure that TLS certificate parameters are configured with correct values	Verify X.509 certificates used to authenticate SSL, TLS, and DTLS connections.
104	Spoofing	An adversary can create a fake website and launch phishing attacks	Phishing is attempted to obtain sensitive information such as usernames, passwords, and credit card details (and sometimes, indirectly, money), often for malicious reasons, by masquerading as a Web Server which is a trustworthy entity in electronic communication	Verify X.509 certificates used to authenticate SSL, TLS, and DTLS connections. Validate all redirects within the application are closed or done safely.

105	Spoofing	An adversary can steal sensitive data like user credentials	Attackers can exploit weaknesses in system to steal user credentials. Downstream and upstream components are often accessed by using credentials stored in configuration stores. Attackers may steal the upstream or downstream component credentials. Attackers may steal credentials if, Credentials are stored and sent in clear text, Weak input validation coupled with dynamic sql queries, Password retrieval mechanism are poor,	Explicitly disable the autocomplete HTML attribute in sensitive forms and inputs. Perform input validation and filtering on all string type Model properties. Validate all redirects within the application are closed or done safely. Enable step up or adaptive authentication. Implement forgot password functionalities securely. Ensure that password and account policy are implemented. Implement input validation on all string type parameters accepted by Controller methods. Use private public key certificates
106	Spoofing	An adversary can spoof the target web application due to insecure TLS certificate configuration	Ensure that TLS certificate parameters are configured with correct values	Verify X.509 certificates used to authenticate SSL, TLS, and DTLS connections.
107	Spoofing	An adversary can create a fake website and launch phishing attacks	Phishing is attempted to obtain sensitive information such as usernames, passwords, and credit card details (and sometimes, indirectly, money), often for malicious reasons, by masquerading as a Web Server which is a trustworthy entity in electronic communication	Verify X.509 certificates used to authenticate SSL, TLS, and DTLS connections. Validate all redirects within the application are closed or done safely.
108	Spoofing	Attackers can steal user session cookies due to insecure cookie attributes	The session cookies is the identifier by which the server knows the identity of current user for each incoming request. If the attacker is able to steal the user token he would be able to access all user data and perform all actions on behalf of user.	Applications available over HTTPS must use secure cookies. All http based application should specify http only for cookie definition.
109	Spoofing	An adversary can steal sensitive data like user credentials	Attackers can exploit weaknesses in system to steal user credentials. Downstream and upstream components are often accessed by using credentials stored in configuration stores. Attackers may steal the upstream or downstream component credentials. Attackers may steal credentials if, Credentials are stored and sent in clear text, Weak input validation coupled with dynamic sql queries, Password retrieval mechanism are poor.	Explicitly disable the autocomplete HTML attribute in sensitive forms and inputs. Perform input validation and filtering on all string type Model properties. Validate all redirects within the application are closed or done safely. Enable step up or adaptive authentication. Implement forgot password functionalities securely. Ensure that password and account policy are implemented. Implement input validation on all string type parameters accepted by Controller methods. Use private public key certificates
110	Spoofing	An adversary can spoof the target web application due to insecure TLS certificate configuration	Ensure that TLS certificate parameters are configured with correct values	Verify X.509 certificates used to authenticate SSL, TLS, and DTLS connections.

111	Spoofing	An adversary can get access to a user's session due to insecure coding practices.	The session cookies is the identifier by which the server knows the identity of current user for each incoming request. If the attacker is able to steal the user token he would be able to access all user data and perform all actions on behalf of user.	Encode untrusted web output prior to rendering. Sanitization should be applied on form fields that accept all characters e.g, rich text editor.
112	Spoofing	An adversary can get access to a user's session due to improper logout and timeout.	The session cookies is the identifier by which the server knows the identity of current user for each incoming request. If the attacker is able to steal the user token he would be able to access all user data and perform all actions on behalf of user.	Set up session for inactivity lifetime. Implement proper logout from the application.
113	Spoofing	An adversary can create a fake website and launch phishing attacks.	Phishing is attempted to obtain sensitive information such as usernames, passwords, and credit card details (and sometimes, indirectly, money), often for malicious reasons, by masquerading as a Web Server which is a trustworthy entity in electronic communication	Verify X.509 certificates used to authenticate SSL, TLS, and DTLS connections. Validate all redirects within the application are closed or done safely.
114	Spoofing	Attackers can steal user session cookies due to insecure cookie attributes.	The session cookies is the identifier by which the server knows the identity of current user for each incoming request. If the attacker is able to steal the user token he would be able to access all user data and perform all actions on behalf of user.	Applications available over HTTPS must use secure cookies. All http based application should specify http only for cookie definition.
115	Spoofing	An adversary can steal sensitive data like user credentials.	Attackers can exploit weaknesses in system to steal user credentials. Downstream and upstream components are often accessed by using credentials stored in configuration stores. Attackers may steal the upstream or downstream component credentials. Attackers may steal credentials if, Credentials are stored and sent in clear text, Weak input validation coupled with dynamic sql queries, Password retrieval mechanism are poor,	Explicitly disable the autocomplete HTML attribute in sensitive forms and inputs. Perform input validation and filtering on all string type Model properties. Validate all redirects within the application are closed or done safely. Enable step up or adaptive authentication. Implement forgot password functionalities securely. Ensure that password and account policy are implemented.Implement input validation on all string type parameters accepted by Controller methods.
116	Spoofing	An adversary can spoof the target web application due to insecure TLS certificate configuration.	Ensure that TLS certificate parameters are configured with correct values	Verify X.509 certificates used to authenticate SSL, TLS, and DTLS connections.
117	Spoofing	An adversary can get access to a user's session due to improper logout and timeout.	The session cookies is the identifier by which the server knows the identity of current user for each incoming request. If the attacker is able to steal the user token he would be able to access all user data and perform all actions on behalf of user.	Set up session for inactivity lifetime. Implement proper logout from the application.
118	Spoofing	An adversary can spoof the target web application due to insecure TLS certificate configuration.	Ensure that TLS certificate parameters are configured with correct values	Verify X.509 certificates used to authenticate SSL, TLS, and DTLS connections.

119	Spoofing	An adversary can steal sensitive data like user credentials.	Attackers can exploit weaknesses in system to steal user credentials. Downstream and upstream components are often accessed by using credentials stored in configuration stores. Attackers may steal the upstream or downstream component credentials. Attackers may steal credentials if, Credentials are stored and sent in clear text, Weak input validation coupled with dynamic sql queries, Password retrieval mechanism are poor.	Use private public key certificates.
120	Spoofing	Blockchain Security Issues - Key management risk.	Blockchain is susceptible to theft of private keys and the control of the assets associated with external addresses being taken away.	The certificates and keys of the SeCIU, School Registry Offices and students must be protected.
121	Spoofing	An adversary can get access to a user's session due to insecure coding practices.	The session cookies is the identifier by which the server knows the identity of current user for each incoming request. If the attacker is able to steal the user token he would be able to access all user data and perform all actions on behalf of user.	Encode untrusted web output prior to rendering. Sanitization should be applied on form fields that accept all characters e.g, rich text editor.
122	Tampering	An adversary can gain access to sensitive data stored in Web App's config files.	An adversary can gain access to the config files. and if sensitive data is stored in it, it would be compromised.	Encrypt sections of Web App's configuration files that contain sensitive data.
123	Tampering	An adversary can gain access to sensitive data stored in Web App's config files.	An adversary can gain access to the config files. and if sensitive data is stored in it, it would be compromised.	Encrypt sections of Web App's configuration files that contain sensitive data.
124	Tampering	An adversary can gain access to sensitive data stored in Web App's config files.	An adversary can gain access to the config files. and if sensitive data is stored in it, it would be compromised.	Encrypt sections of Web App's configuration files that contain sensitive data.
125	Tampering	An attacker steals messages off the network and replays them in order to steal a user's session	An attacker steals messages off the network and replays them in order to steal a user's session	Use encrypted network

126	Tampering	An adversary can deface the target web application by injecting malicious code or uploading dangerous files	Website defacement is an attack on a website where the attacker changes the visual appearance of the site or a webpage.	Access third party javascripts from trusted sources only. Ensure that each page that could contain user controllable content opts out of automatic MIME sniffing . Use locally-hosted latest versions of JavaScript libraries . Ensure appropriate controls are in place when accepting files from users. Perform input validation and filtering on all string type Model properties. Ensure that the system has inbuilt defences against misuse. Implement input validation on all string type parameters accepted by Controller methods. Sanitization should be applied on form fields that accept all characters e.g, rich text editor.
127	Tampering	An adversary can gain access to sensitive data stored in Web App's config files	An adversary can gain access to the config files. and if sensitive data is stored in it, it would be compromised.	Encrypt sections of Web App's configuration files that contain sensitive data.
128	Tampering	An attacker steals messages off the network and replays them in order to steal a user's session	An attacker steals messages off the network and replays them in order to steal a user's session	Use encrypted network
129	Tampering	An adversary can deface the target web application by injecting malicious code or uploading dangerous files	Website defacement is an attack on a website where the attacker changes the visual appearance of the site or a webpage.	Access third party javascripts from trusted sources only. Ensure that each page that could contain user controllable content opts out of automatic MIME sniffing . Use locally-hosted latest versions of JavaScript libraries . Ensure appropriate controls are in place when accepting files from users. Perform input validation and filtering on all string type Model properties. Ensure that the system has inbuilt defences against misuse. Implement input validation on all string type parameters accepted by Controller methods. Sanitization should be applied on form fields that accept all characters e.g, rich text editor .

TABLE A.2: Threats mitigated by implementation

A.3 Not applicable threats

Threats that does not apply				
Id	**Category**	**Title**	**Description**	**Possible mitigations**
130	Tampering	Fraudulent Activity 3 [Say19]	Student buys certificate from non-accredited university or diploma mill.	The case study does not include diplomas that are not from UDELAR.
131	Tampering	Fraudulent Activity 4 [Say19]	Student uses misleading translated copy of the real document.	It is not in the scope of the solution to translate certificates, although this functionality could be added and this threat could be mitigated by storing a hash of the translated certificates.
132	Tampering	Fraudulent Activity 6 [Say19]	Student uses work certificate or life experience and then convert that into academic credit with the support of corrupt officials in accredited university degree.	It is not within the scope of this solution to ensure the correct assignment of grades or certificates for each School Registry Office. However, these changes will be recorded in the system.
133	Tampering	Fraudulent Activity 7 [Say19]	Corrupt teacher takes unofficial fee to assure the passing grade without submitting the assignments or required studies done.	It is not within the scope of this solution to ensure the correct assignment of grades or certificates for each School Registry Office.
134	Tampering	Fraudulent Activity 8 [Say19]	Teachers are sometimes biased and grade students higher than their performance such as on exam paper.	It is not within the scope of this solution to ensure the correct assignment of grades or certificates for each School Registry Office.
135	Tampering	Fraudulent Activity 9 [Say19]	Fraud syndicate have links with corrupt officials to store fake certificate data in the university's database.	It is not within the scope of this solution to ensure the correct assignment of grades or certificates for each School Registry Office. However, these changes will be recorded in the system.

TABLE A.3: Non applicable threats

A.4 Unmitigated Threats

Threats not mitigated				
Id	**Category**	**Title**	**Description**	**Possible mitigations**
136	Denial of Service	An adversary can perform action on behalf of other user due to lack of controls against cross domain requests	Failure to restrict requests originating from third party domains may result in unauthorized actions or access of data	This threat is not mitigated in the proposed design. DoS protection measures must be implemented.
137	Denial of Service	An adversary can perform action on behalf of other user due to lack of controls against cross domain requests	Failure to restrict requests originating from third party domains may result in unauthorized actions or access of data	This threat is not mitigated in the proposed design. DoS protection measures must be implemented.
138	Denial of service	Off chain Blockchain Security Issues - Availability	In an off chain commit model, availability is not guaranteed.	Consider High availability infraestructure.

TABLE A.4: Unmitigated Threats