ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO
POLITÉCNICO
DO PORTO

M

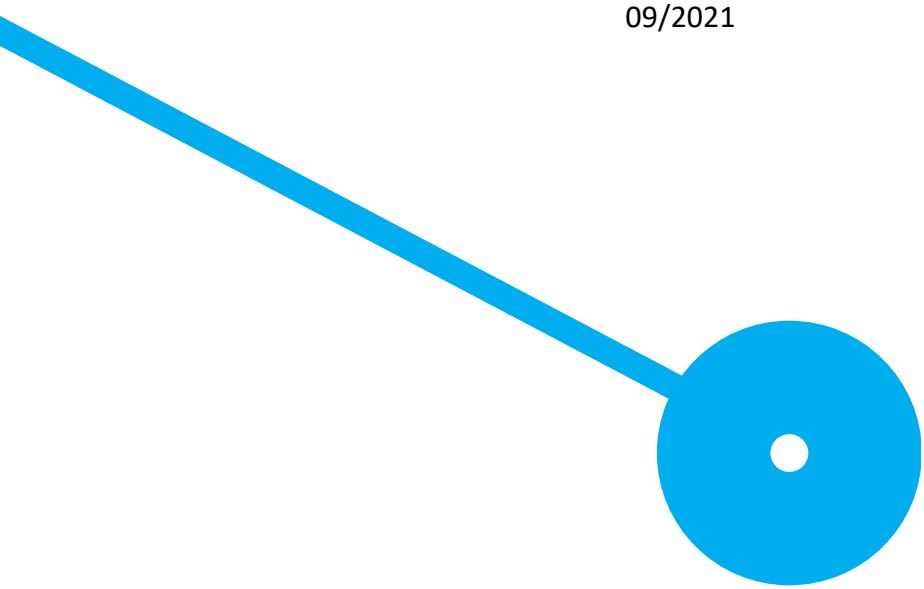
MESTRADO
ENGENHARIA INFORMÁTICA

blockchain-assisted Academic Council

Electronic Vote System (bACEVS)

João Pedro Gonçalves Alves

09/2021



ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO
POLITÉCNICO
DO PORTO

M

MESTRADO
ENGENHARIA INFORMÁTICA

blockchain-assisted Academic Council Electronic Vote System - (bACEVS)

João Pedro Gonçalves Alves

Professor Doutor António Alberto dos Santos Pinto

Acknowledgments

In first place i must say a word of thanks to Professor Dr. António Pinto to accept to be my mentor in this work, meaning a lot in this academic journey of my life, as his dedication, support and passion shown by the subject, it was decisive for my personal motivation. I also leave my sincere thanks to the people who, directly or indirectly, contributed to the progress and conclusion of this study:

To institution for providing the possibility of attending this course and supporting this work, as well as its professors in general and administrative technicians who support it.

To my family and friends who support me in this journey.

Resumo

A digitalização de tarefas e processos administrativos é uma realidade nos dias de hoje, traduzindo-se em mais valias como a agilidade na gestão de processos, ou no acesso simplificado a dados armazenados. A digitalização de processos relacionados com tomada de decisão em órgão colegiais, como são os Conselhos Académicos, ainda não é algo totalmente alcançado. Atos de votação ainda são realizados de forma presencial, ou quando muito em reuniões à distância, sem que se tenha uma cabal confirmação do sentido de voto de cada elemento. Tal é particularmente complexo de conseguir em cenários de reuniões à distância, onde podem existir quebras de ligação ou interrupções de fluxos de áudio ou vídeo.

Neste trabalho, propõe-se uma nova plataforma digital que permita que tomadas de decisão, por votação em Conselhos Académicos, sejam suportados por um sistema que garanta a integridade das decisões tomadas, mesmo quando reunidos à distância. Mais, além dos sentidos de voto e respetivos resultados, poder-se-á ainda garantir a integridade de todos os documentos partilhados com todos os elementos que constituem um Conselho Académico. Tal integridade de votos e documentos será conseguida com recurso a uma Blockchain.

Com esta solução pretende-se demonstrar a viabilidade da digitalização completa da tomada de decisão em Conselhos Académicos de Instituições de Ensino Superior. O processo de digitalização compreenderá o acesso à informação, por meios electrónicos, de forma segurança, com garantias de integridade da informação, bem assim como a digitalização do processo de votação.

Keywords: Blockchain, Conselhos Académicos, Votação, Digitalização

Abstract

The digitisation of administrative tasks and processes is a reality nowadays, translating into added value such as agility in process management, or simplified access to stored data. The digitisation of processes related to decision-making in collegiate bodies, such as Academic Councils, is not yet something fully achieved. Voting acts are still carried out in person, or at most in online meetings, without having a real confirmation of the vote of each element. This is particularly complex to achieve in remote meeting scenarios, where connection breaks or interruptions of audio or video streams may exist.

In this work, a new digital platform is proposed that allows decision-making, by vote in Academic Councils, to be supported by a system that guarantees the integrity of the decisions taken, even when meeting online. Furthermore, in addition to the votes and their results, it will also be possible to guarantee the integrity of all documents shared with all the elements that constitute an Academic Council. Such integrity of votes and documents will be achieved using a Blockchain.

With this solution, it is intended to demonstrate the feasibility of the complete digitisation of decision making in Academic Councils of Higher Education Institutions. The digitisation process will include access to information, in a secure manner, with guarantees of integrity, as well as the digitisation of the voting process.

Keywords: Blockchain, Academic council, Voting, Digitisation

Contents

List of Figures

List of Tables

Listings

List of Acronyms

EVS	Electronic Vote System
DEX	Decentralized EXchanges
DApps	Decentralized applications
ERC-20	Ethereum Request for Comments 20
LAMP	Linux, Apache, MySQL, PHP
CENTOS	Community ENTerprise OS
PHP	Hypertext Preprocessor
MYSQL	My Structured Query Language
SQL	Structured Query Language
JSON	JavaScript Object Notation
AD	Active directory
LDAP	Lightweight Directory Access Protocol
VM	Virtual Machine
CRUD	Create, Read, Update and Delete
VYV	Verify-Your-Vote
CA	Certification Authority
E2E	End-to-End
HMAC	keyed-Hash Message Authentication Code
ICO	Initial Coin Offering
PoW	Proof-of-Work

bACEVS	Blockchain Academic Council Electronic Vote system
RAM	Random Access Memory
GB	Gigabytes
SSD	Solid State Drive
CN	Common Name
SAM	Security Account Manager
HTTPS	Hypertext Transfer Protocol Secure
vscode	Visual Studio Code

Chapter 1

Introduction

In Portugal, a common organisation structure present in higher education institutions is the council, examples being the scientific council or the pedagogical council. A significant part of the activities of Portuguese Universities and Polytechnic Institutes is governed by these academic councils. Their decision making process is based on majority decisions by nominal member voting without anonymity. They operate in a way much similar to the nominal group technique firstly developed in 1971 where topics are described and presented to voters group to be discussed and voted [2]. The councils are usually composed of members, a president and a secretary. The president will present one or more topics to be discussed and voted, each member will manifest himself either for or against the decision, and the votes are counted. Decisions are then written in a minutes book that accounts the votes of all members on all issues but without a formal assurance that the vote of each member was considered or rightly considered. Moreover, if no additional measures of integrity are considered, a section of a minutes book may, in principle, be altered without its members being aware of such change. Such alteration, if performed by someone with bad intentions, may even occur long after the end of the mandates of the related members.

Electronic Vote System (EVS) is considered as a way to perform a more effective act of voting in democratic societies [3]. It reduces the cost of electoral acts by requiring less human resources per act. While EVS has yet to become the *de facto* way of voting of modern days, the forced push to remote working due to the COVID pandemic [4] may increase the pace of its adoption.

Blockchain was news every day specially in financial market and bitcoin currency [5] is a good example of it. Today, Blockchain is explored by many industries as a secure and cost-effective way to create and manage distributed databases and maintain records for digital transactions of all types

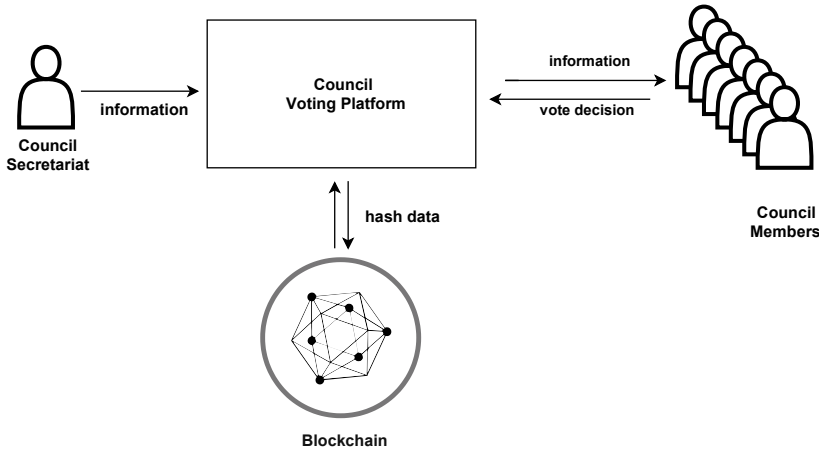


Figure 1.1: Reference Scenario

and increasingly viewed as a solution for securely tracking and sharing data between multiple business entities. Trust, security, storage and reliability of information achieve a new level with this process and business processes and transactions all over the world that depend on these values are now beginning to try to incorporate Blockchain in their core system[6]. Due to being decentralised and neutral from government laws or rules, it begins to extend to other areas such as social, human and political[7] being applied to solve real problems.

That lead the author to consider a Blockchain-assisted EVS to enable the integrity of the decision making process of academic councils that are gathering remotely or online. In [8], the same author already analysed the general adequacy of Blockchain technologies when applied to EVS. The conclusion was that EVS can benefit from the integration of Blockchain, but requires a case-by-case analysis. No solution that fulfils the common requirements of EVS, plus not need anonymity of vote, while being scalable and without requiring the spending of digital currency, was identified. Key shortcomings where related to the anonymity requirement that, while not present in academic council voting, is a common requirement of general EVS.

The adopted reference scenario, depicted in previous figure 1.1, comprises an EVS for academic councils based on a web platform where it is possible to access information regarding a meeting or topic under discussion and record their vote. The President, or Secretary, of the academic council starts a meeting and puts one, or more, topics up to discussion. Each topic, after being discussed, is then subject to a nominal vote by each member. Members cast their vote on their device when the President closes each topic. The votes are then collected and counted, and a decision is made. After which, the decision is computed and stored in a ledger or Blockchain. The decision will consist of the result of a secure hash function over the collected votes.

The author, in the work herein, propose a novel platform that will automate vote counting and decision recording in situations that have integrity requirements but do not require voter anonymity, has is the case of the adopted reference scenario.

This work is organised in chapters. Chapter 2 defines what is Blockchain its characteristics and what smart contracts are. I describe too EVS and their main principles. Chapter 3 is a study of existing solutions that appears to implement features close to proposed solution and their comparation. Chapter 4 presents requirements and specification of the the proposed solution, including database entities overview, architecture and some security considerations. Chapter 5 is dedicated to the prototype description and validation of requirements. In chapter 6 a review of this work is made including considerations for future work.

Chapter 2

Background

This chapter is divided in two sections, a section is dedicated to Blockchain and is definitions, the other is a review of different types and characteristics of EVS where its features are analysed.

2.1 Blockchain

2.1.1 Definitions

The main definitions and terms used in Blockchain technology are described below in general terms[9][10].

Decentralized ledger - Its a ledger stored on any number of computers connected to same network such as the Internet. Each computer is called a node too and contains a complete history of every completed transaction in a particular blockchain, beginning with the first transaction processed on the first block of the blockchain.

Consensus - A consensus mechanism is methodologies used to achieve agreement, trust, and security across a decentralized computer network. The Proof-of-Work (PoW) is a common consensus algorithm used by the most popular cryptocurrency networks like bitcoin and litecoin. in wich a participant node is required to prove that the work done and submitted by them qualifies them to receive the right to add new transactions to the blockchain.

Peer-to-Peer Network - A network of computers or nodes, connected to each other. The nodes are all connected, but on a decentralized basis, there is no single server that all nodes connect to. The network of nodes operates under the same set of rules or protocol. The protocol is expressed in computer code that resides on each node on the network. The agreed

protocol guarantees that only information about which the network reaches consensus will be included in the blockchain. All computers on the network running the protocol code must agree on whether a change to the blockchain should be made and, if so, what the change should be and no one can unilaterally impose a blockchain change.

Protocol - The protocol consists in common code or software that each computer in the network agrees to use.

Mining - is the act of make Blockchain transactions valid and requires computing power and electricity consumption to solve "puzzles" or challenges. Mining rewards coins based on your computing power spent to do it.

Merkle Tree - Its a tree in which every leaf or node is labelled with the hash of a data block and each single non-leaf node is labelled with the cryptographic hash of the labels of its child nodes.

Nonce - A number to be used only once in a cryptographic communication and generally includes a timestamp.

The term Blockchain originally appeared in [5] and consisted if a data structure that comprises information regarding transactions and guaranties data integrity by storing the secure hash of each block, on the following block, thus creating a chain of blocks. Since then, Blockchain has gained public attention and is more commonly used to denote what might be more "trust-less systems" where everyone can be part of the ledger. The basic idea is that everyone who mines a block of transactions in bitcoin, maintains a record of other transactions that have happened or are happening in the system. This decentralised ledger gives multiple authenticity points for every transaction that has happened; plus, the rules are predefined and equal for each wallet user.

The first [11] version of Blockchain uses the SHA-256[12] secure hash function but introduced a concept of signed block. A block would be considered as signed if, and only if, the resulting hash value with start with a specific number of zeros. The task of finding such hash values was named as mining, or PoW. These functions made the system secure but very inefficient. Specialised hardware, known as mining platforms, were specifically created to handle the work of processing higher numbers of transactions. Each mined block is rewarded with bitcoins. This way, it is possible to detect attempts to fraudulently adding records to a chain. A simple definition [13] of Blockchain is a type of ledger which is decentralised, transparent and secure. Its secure not because it hides information or restricts access to it, but by making it tamper proof. Figure 2.1 shows a simple representation of the bitcoin Blockchain where a public ledger is provided ordered and timestamped for recorded transactions. In this simplified version, one or more transactions are stored into data part of each block and copies of each transaction as hashed then paired, hashed and paired again and hashed again until one single hash remains the merkle root of a merkle tree. The

merkle root is stored in the block header and each block also stores the hash of previous previous block's header making a chain of blocks.[14]

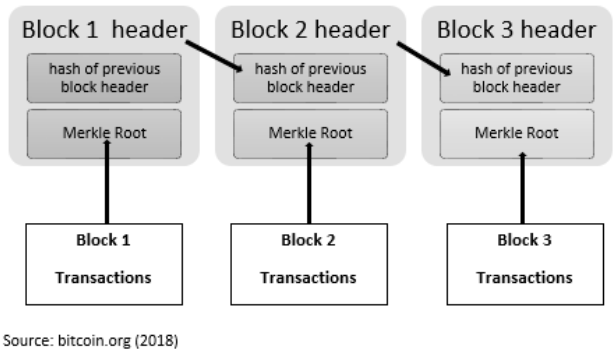


Figure 2.1: Representation of the bitcoin's Blockchain (adapted)[1]

Its architecture is based on a network composed by several nodes, which store transaction related data, and processors, which validate the introduction of new information onto the Blockchain. Other common feature of Blockchains is its use of a trust-less proof mechanism [15] that certifies all transactions on the network. This feature substitutes any third-party trust authority or any intermediary, such as banks or brokers, and only involving two entities in each transaction. Delloite in a published work "KEY CHARACTERISTICS OF THE BLOCKCHAIN" [16] identified eight key characteristics of Blockchain technologies:

1. Distributed - All peers, with no distinction, may have a copy of all information within the Blockchain.
2. Digital - Blockchain is fully digitised, without any manual interaction.
3. Updated in Real time - Information is stored within the Blockchain in (near) real time.
4. Chronological and time stamped - The Blockchain repository stores block information of each transaction with links previous blocks in a chronological manner, making a trail of the underlying transaction sequence.
5. Cryptographically sealed - All blocks are stored cryptographically sealed, making it impossible to delete or alter blocks within a network, ensuring a high level of trust and robustness, creating true digital assets.
6. Irreversible an auditable - Because it is stored in a decentralised way, it becomes failure-resistant, even when a large number of peers fails, for as long as there are even more peers still operating.

7. Operates trustless - Be being consensus-based, normally all transactions are only executed if all participants on the network unanimously approve it.

8. Fewer third parties - Participants of Blockchain networks independently validate information without the need for centralised authorities

Blockchains can also be grouped in two types: permissionless and permissioned [17]. A permissionless Blockchain consists of a distributed network of equal peers that offers a trust mechanism without requiring intermediaries assuring trust to the involved parties. Hashes are used to link transaction blocks. Multiple anonymous miners compete among themselves to solve a problem through trial and error. When a miner solves the algorithm, the others validate it in a consensual process. Access is granted through public keys and personal transaction information is assured using private keys. Permissioned Blockchains normally involve a consortium of multiple organisations, where block transitions are created and verified by authorised gatekeepers instead of anonymous miners. In essence, in a permissioned Blockchains, peers that have privileges to change the information on the Blockchain are different from the ones the can only read the Blockchain.

Blockchains can also implement smart contracts [7]. Smart contracts are computer programs that can be consistently executed by a network of mutually distrusting nodes, without the arbitration of a trusted authority. Because of their resilience to tampering, smart contracts are appealing in many scenarios, especially in those which require that transfers of money must respect certain predetermined rules, examples being the financial sector or games.

2.1.2 Smart Contracts

Smart contracts are a piece of computer software created to automate a self-enforcing contract. It means that it will trigger a certain action when predetermined conditions are met. Although they become popular in the context of Blockchains and cryptocurrencies, the concept was first described by Nick Szabo in 1994[18], where he described smart contracts many years before the creation of Bitcoin. Another definition can be[18]:

> "A computerised transaction protocol that executes the terms of a contract. The general objectives of smart-contract design are to satisfy common contractual conditions (such as payment terms, liens, confidentiality, and even enforcement), minimise exceptions both malicious and accidental, and minimise the need

> for trusted intermediaries. Related economic goals include lowering fraud loss, arbitration and enforcement costs, and other transaction costs." (in [19])

Smart contracts assumed an important role in the Blockchain space and cryptocurrency markets. Particularly with regard to Ethereum tokens that follow the Ethereum Request for Comments 20 (ERC-20) standard, these tokens are often distributed through Initial Coin Offering (ICO) events and the use of smart contracts enables its trustless operation. Its uses can also facilitate payment processing for Decentralized applications (DApps), decentralised markets or Decentralized EXchanges (DEX). Another area where the use of smart contracts becomes interesting is for supporting financial services. Technology can be used to automate the clearing and settlement of transactions, the payment of bond coupons, or even the calculation and payment of insurance claims. Smart contracts are versatile enough to be applicable to virtually any industry where it is necessary to transfer funds, digital assets or any type of digital information between two parties. The equipment leasing industry, for example[20], has made extensive use of these contracts in real-world scenarios in order to make leasing arrangements more efficient[21]. In the healthcare sector, the technology is being explored as a measure against data manipulation in clinical trials[22]. Smart contracts can even be used to enforce intellectual property agreements[23][24] by establishing a definitive record of shared property rights and allocating all royalties and proceeds from pieces of intellectual property conformity.

2.1.3 Smart Contracts on Hyperledger Fabric

According to the producers of Hyperlegder [25], from an application developer's perspective, a smart contract paired with the ledger, forms the heart of a Hyperledger Fabric Blockchain system. A ledger contains facts that comprise the state of a set of objects, a smart contract defines executable logic that generates new facts that are added to the ledger. A chaincode is a group that contains related smart contracts that will be deployed together, but can also be used for low-level system programming.

Before organisations can interact with each other, they must define a common set of contracts (terms, data, rules, concepts, and processes). This group of contracts establish the business model that will governs all future interactions. A smart contract can implement governance rules for any type of business object so they can be automatically applied when the smart contract runs. For example, a smart contract can ensure that a new car is delivered within a specified time frame or that funds revert back to the buyer, improving the flow of goods or capital. Hyperledger Fabric users often refer to smart contract and chaincode interchangeably, but these are not truly the same. A smart contract defines the transaction logic of an

object. It is then packaged into a chaincode which is then deployed on to a Blockchain network. Multiple smart contracts can be defined in the same chaincode and when a chaincode is deployed, all smart contracts within it are made available to applications. The ledger is a Blockchain that immutably records transactions that, in turn, update states of objects. A smart contract accesses two distinct parts of the ledger, the Blockchain and a world state that maintains a cache of the current value of those states. Smart contracts basically "put", "get", and "delete" states in the world state and can also query the Blockchain for record of transactions. A "get" typically represents a query to retrieve information about the current state of a business object. A put option usually creates a new business object or modifies an existing one. A deletion generally represents the removal of a business object from the ledger's current state.

2.2 Electronic Voting Systems

Public administrations encourage the development[26] of EVS to bring efficiency, reduction of bureaucracy, reduction of costs and providing voters with an online system capable of providing an electronic ballot that gathers all the characteristics that it must have, essentially secrecy in most cases. The adoption of EVS is a step that must be taken on the path of dematerialisation of all processes. These must maintain some of the key characteristics of the traditional ballot systems. Vote secrecy appearing as a key characteristic in most cases.

Accordingly to [3], the main constitutional requirements of an secure EVS are: Generality, Freedom, Equality, Secrecy, Directness, Democracy.

- **Generality** - every voter has the right to participate in an electoral process. Eligibility must be founded and regulated by law. Voting technology should be accessible to all voters and should be seen as alternative way for an elector to cast his vote. The democratic principle, in which all eligible voters should be included in the electoral process, must ensure that public electronic infrastructures are available (public Internet kiosks or state infrastructure offices or others) so that they can exercise their right to vote. Eligibility can be assured by the registration of eligible voters and by their identification at the time of their registration. Registration and authentication are essential procedures so that the principle of universality is respected and that elections can not be manipulated. The purpose of maintaining voter registration is to ensure that only eligible voters can vote and that they can not vote more than once.

- **Freedom** - the election process must ensure that it occurs without

any violence, coercion, pressure or any other manipulative intervention that may be inflicted on the voter by a third party. EVS assumes an important role here, guaranteeing the uncoercibility[27] and the prevention of vote buying and also the guarantee that no voter can be audited in order to perceive their tendency to vote.

- **Equality** - in a general election context represents the generic principle of equality and represents a pillar in the democracies. Two main requirements must be ensured: 1) equality for each party / candidate and equal to the voting rights of each voter. The electronic registration of each voter card must have the same paper registration requirements. Electronic ballot papers must also be similar to those printed and on the same form accessible. Transparency must be supported to so all parties should have same opportunity for equal access to elements of voting procedure. An EVS must assure that each voter can only vote once and eligible voters only can vote. Duration of electoral act must be defined and assure voting period is not exceeded.

- **Secrecy** - depending on each type of elections, secrecy may be necessary as is the case of free political elections.

- **Directness** - in traditional voting systems, direct election requires that there be no intermediaries in the voting process. This principle should also be adapted to an electronic voting procedure. The relevant requirement is that all online ballots are recorded and counted directly. A problem may arise if the voting period differs from the voting procedure (online or offline) used to vote. The results of online voting can influence the outcome of the entire electoral process and limit the integrity and legitimacy of the whole process. To avoid this, a system can be developed allowing the recording and maintenance of ballots, while prohibiting any counting before the end of the voting period (off-line).

- **Democracy** - a democratic EVS must respect the requirements of a traditional electoral system. However, additional requirements must also be met, in particular because of the nature of electronic voting. These requirements relate to the preservation of attributes and properties, such as transparency, accountability, security, accuracy and legitimacy of the system. Voters should be able to understand how elections are conducted. Traditional voting procedures work transparently for both voters and other constituencies. On the contrary, electronic voting procedures are not transparent because the average voter does not have the knowledge to understand how the system works. Therefore, in electronic voting, voters must demand much more confidence in the

technology used and the people involved (election officials, technology providers, etc.)

However, additional requirements must also be met due to the nature of electronic voting. Mainly because other features are being added and new requirements are being imposed [3]. These requirements relate to the preservation of attributes and properties, such as transparency, accountability, security, accuracy and legitimacy of the system. Voters should be able to understand how elections are conducted. The overall security of the voting process must be assured and fraud proof. From the voter's point of view, the EVS must confirm his participation in the electoral act. It should enable indiscriminate access to the voting platform, as well as maintain a record of votes. There must be a reliable certification procedures for hardware and software. The entire infrastructure, as well as, any system functionality, must be registered (for example, all software excluding interfaces must be open source). All operations (authentication, recording of votes, etc.) must be monitored, while the confidentiality is preserved. The infrastructure must be open for inspection by authorised bodies. Voters, parties and candidates must be assured that there has been no malpractice. Adequate system security must be ensured, be simple and easy to use.

Moreover, from the user standpoint, additional requirements can also be identified[3]. Examples being:

1. Participation in the voting process should be confirmed.
2. Uncoercibility should be ensured.
3. Ability for consciously non-valid vote should be provided for.
4. Only eligible voters should be able to vote (eligibility)
5. Each eligible voter should be able to vote only once (un-reusability)
6. No voter should be able to duplicate/change his vote or the vote of someone else (integrity).
7. The voter should be able to verify that her vote is calculated in the final tally (verifiability).
8. Voters should be bale to have indiscriminating access to the voting infrastructure (accessibility).
9. Registration, authentication and voting procedures should be evidently separated.
10. Votes should be validated separately and independently from voter authentication.

11. Uniqueness - No intermediaries should be involved in the voting process (i.e. no person can be authorised to vote for another person).

12. Each and every ballot should be recorded and counted correctly.

Multiple EVS proposals can be found in literature. Next, the ones that satisfy voter anonymity will be described.

Zerocoin is an extension to the bitcoin protocol that enables fully anonymous currency transactions. These services are also known as bitcoin laundry services [28]. Takabatake, et al., in [29] proposed the use of Zerocoin as a basis for an EVS. Doing so, the transparency, integrity and anonymity requirements are resolved. The transparency and integrity due to the distributed and public nature of the Blockchain. The anonymity with the use of Zerocoin. This solution, however, is not completely anonymous because the IP address of public nodes can be known. Moreover, it is not clear if registration, authentication and voting procedures are separated.

Ahmed Ben Ayed proposed a Blockchain based EVS [30]. He identified four main requirements for an EVS, these being: authentication, anonymity, accuracy and verifiability. Accuracy and verifiability can easily be extracted from the Blockchain. Authentication, to assure that only registered voters can cast votes, is a parallel process not supported by Blockchain. The solution proposed by Ayed, will hash the voter's name, number, vote and the hash of the previous block. Of this information, only the vote is *a priori* unknown and, by being a limited set of options, it should be feasible to break as one would only need to run has many hash operations as the number of voting possibilities. Moreover, anonymity of the voter's identity is said to be achieved by not allowing any links between voters and ballots. Bitcoin transactions are considered pseudo-anonymous as only wallet addresses are used in transactions and not the identifiers of the people behind each transaction. However, upon discovery of who owns the wallet, anyone can view all transactions with that address. To introduce anonymity, solutions like Zerocoin [31] started to appear.

BlockVotes, proposed in [32], is yet another Blockchain based EVS. BlockVotes uses a ring signature algorithm to generate values that are stored in the Blockchain. While satisfying requirements as ballot privacy, anonymity individual verifiability, eligibility completeness, uniqueness, robustness, coercion-resistant, it does not satisfy fairness nor receipt-freeness requirements.

Jason & Yuichi proposed another EVS based on Bitcoin [33]. It consists in using the Bitcoin Protocol in conjunction with a blind signature scheme [34]. Blind signatures enable that a third party can attest for the contents of data (a vote) without having access to the data. This solutions enables anonymity in voting but requires the usage of *cyprocurrency* to cast a vote. The authors suggest the use of prepaid bitcoin cards and proposed three dis-

tinct entities: the voter, an administrator and a counter. The administrator is the entity that makes use of blind signatures.

Tasarov, et al., proposed another Blockchain EVS [35] that, instead of bitcoin, uses Zcash [36] as the underlying payment system. The proposed EVS achieves transaction anonymity using Zcash, unmodified. This anonymity relied on zero-knowledge proofs as a substitution of the PoW scheme present in bitcoin. It supports both anonymous and transparent transactions and has two types of addresses, which differs from the bitcoins single address and uses four distinct steps: registration, notification, voting ad counting. For voters identification and verification it uses X.509 certificates and a certificate authorities (CA).

Multiple others EVS proposals and reviews of EVS[37] can be found in literature, namely those described in [36], [35], [32], [30], and [28]; all supporting voter anonymity. Moreover, in [38], the authors review various voting protocols and methodologies based on Blockchain. Most of them provide features as non re-usability, verifiability, self-tallying, eligibility and principally anonymity, which is present in most of them. Vote anonymity may not be present in all types of elections. The adopted reference scenario does not require vote anonymity and, therefore, none of these solutions can by directly used in the reference scenario. Other more suited solutions exist and were classified as related work, they will be analysed in the next section.

Chapter 3

Related Work

In this chapter, we review four EVS that are similar to the work presented herein. These either use the same technology or share common features. The case of not having anonymity as a requirement assumes special importance in our analysis.

3.1 Verify-Your-Vote (VYV)

In [39], the authors present the design of an EVS that uses Blockchain technology. It consists of an online voting solution where verifiability has more relevance by offering a public bulletin board. It shows public information giving a persistent view to all voters. The main system entities are: the registration server, the election administrator, the eligible voters, and the tallying authority. The Registration Server is a trusted server that assumes the registration of eligible voters registration and their authentication. The Election Administrator consists of an external account that is responsible for managing the election process and for setting timers to guarantee that the election occurs on time, authenticating voters and designating ballots. This entity cooperates with counting authority. Each Eligible Voters has an external account and has the right to vote and can his change vote intention up to the end of the voting phase. The Tallying Authority consists of an externally owned account that participate in ballots creation, decrypting votes and calculating the final election result.

Verify-Your-Vote (VYV) has a unique ballot comprising the ballot number to identify the ballot, candidate names, pseudo ID as candidate position in ballot, choice and counter-values used for verification. The election stage is composed by the setup phase, the registration phase, the authentication phase, the voting, the tallying and the verification phase. In setup, adminis-

trator starts the process by generating election parameters, and publishing it in a bulletin board. He then creates timers to schedule the distinct phases. The registration phase is an offline process. Voter registers at a polling server (Registration Server) with a valid password and his identification, done by an in person meeting. In the authentication phase, after being successfully authenticated, the voter has access to an interface that allows him to create his own externally owned account on the election Blockchain. Afterwhich, he can then publish his public key, as well as his address, which is derived from the public key, by taking the last 20 bytes of the hash of the public key. In the voting phase, the tallying authority randomly chooses a ballot for each voter and encrypts it with the voter's public key, previously published on the bulletin board during the authentication phase, and sends it to the voter in a transaction on the Blockchain. The voter then decrypts his ballot using his secret key, chooses a candidate with pseudo ID and encrypts his bulletin number, casting a vote to the tallying authority via the Blockchain. Each voter should memorise the counter-value corresponding to the chosen candidate. In tallying, each tallying authority calculates the number of votes of a specific pseudo ID. Each authority consults the exchange history stored in the Blockchain to derive the value of the offset corresponding to each bulletin. Tallying authorities reconstruct ballots, identify chosen candidates and increment their counters. The verification phase is divided in two sub-phases. First is the reconstruction of counter-values associated with the ballot and the name of the candidate where they must be identical to the ballots of voters. Thus, each voter should access to the Blockchain and check the existence of his counter-value in the list of reconstructed counter-values. The second sub-phase uses the homomorphism property of pairings to check the accuracy of the count. In fact, at the end of tallying phase, each tallying authority publishes the vote count of each candidate on the bulletin board.

VYV satisfies multiple proprieties of EVS. In particular, it supports voter eligibility, fairness, integrity, individual verifiability, universal verifiability, vote privacy, receipt freeness, and coercion resistance. This solution uses a Blockchain as a bulletin board, supporting election acts that are verifiable. Voters can only vote after being authenticated. VYV author's formally verified its security using the ProVerif tool.

3.2 E-Voting with Blockchain

Hardwick, et al, in [40], proposed an EVS based on Blockchain technology that addresses some requirements that are less frequent. In particular, they addressed the capability of a voter changing its vote after being cast. Additionally, the authors strive for a maximum degree of decentralisation where the voters control the network, being its peers. They proposed an

e-voting protocol with a Blockchain based scheme that meets the following goals identified by them: Fairness, Eligibility, Privacy, Verifiability, and Coercion resistance.

The authors, with respect to the resistance to coercion, state that it is not actively achieved. According to them, this property is not possible to be achieved by technological means in remote electronic voting solutions. However, it has the property of forgiveness, which can be perceived as a weaker notion of the property of resistance to coercion. For them, forgiveness is the ability that a voter has to change his vote after it has been cast, if a coercion period has past and the voter doesn't agree with the initial vote intention. Additionally, if the identity of the voters is to remain secret, and only eligible voters can participate in the elections, a Central Authority is introduced that acts as a trusted third party. This solution stores voted ballots in the Blockchain, acting here as a transparent ballot box. The main reason for using the Blockchain is to take advantage of the fact that it enables a group of people to maintain a public database as a ledger, that is owned, updated, and maintained by every participant, but controlled by no one. It will be realised as a network of peers. Each voter will be a peer and will be responsible for making sure that fraudulent votes are rejected, hence the consensus is maintained according to the election rules.

Their solution comprises five components: the Voter, a Central Authority, the Vote, the Ballot, and the Alteration Ballot. Each Voter is identified by his public key and considered as an entity that is allowed to vote for one of the candidates and, as a way of protest, can be cast as "invalid". Voters access the e-voting platform through a voting client installed in their device of preference, the security of which is assumed. The Central Authority provides assurance that only eligible voters can vote, authenticating Voters and issuing token that prove eligibility to vote. Eligibility token takes the form of a digital signature. The Vote consists on a message with a predefined structure, similar to one of a bitcoin transaction, and is required to include a ballot. Once recorded in the Blockchain, each vote is uniquely identified by an ID. The Ballot is a digital representation of the paper where the choice of a voter is written on. It is considered sealed while the opening value of the digital commitment has not been revealed yet and, thus no party, other than the voter, can determine the way a voter voted. Once the opening value has been revealed and the choice of the voter is publicly known, the ballot is considered open. Finally, the Alteration Ballot enables to voters to change their vote after it has been cast, and stored in the Blockchain, minimising coercion acts.

This solution, similarly to the previous, is divided in phases, comprising: the Initialisation, the Preparation, and the Counting phase. The Initialisation phase serves the purpose of preparing the system. The duration of each phase is defined. The rule for the acceptance or not of vote cancellation is also set in this phase. A Blockchain infrastructure will be created. The

Certification Authority (CA) will provide a list of the eligible voters and will provide voters with a public key pair. Blockchain will be initialised with an initialisation block, acting as the genesis block, of the chain. This block does not contain any votes, instead stores all information of the election act, including the key used by the CA for signature validation and the set of valid vote options. In the Preparation phase, voters use their client application of the e-voting platform to make authentication process against the CA that holds the valid list of voters. Upon successful authentication, the client application will generate a public key pair to identify the voter and to be later used as a verifying key. Also within this phase, the voter is invited to cast his vote. In the Counting phase, the system broadcasts a ballot opening message onto the network so that all voters reveal final choice. All nodes of network participate in ballot collecting and in the signatures verification.

This solutions satisfies multiple properties of EVS. In particular, it supports eligibility, privacy, fairness, individual verifiability, and universal verifiability.

3.3 End-to-end voting-system based on bitcoin

The solution proposed in [41] aims to be a re-adaption of the Bitcoin electronic payment system as a decentralized End-to-End (E2E) voting platform. Authors have been inspired by e-payment systems and Bitcoin. Reported properties comprise: eligibility and authentication, verifiability and auditability, uniqueness, accuracy, integrity, anonymity, counting, and recounting. The recounting is foreseen so that the system should allow for the recounting votes as a way of confirming the final results.

This solution, similarly to the previous ones, is divided in phases, comprising: the Pre-voting, the Voting, and the Post-voting phases. In the Pre-voting phase is where the approval process for eligible candidates takes place. A candidate in this context is either a named individual or a different entity (eg. just a declaration). The objective is to obtain a list of eligible candidates with some asymmetric keys: the public one is associated with the candidate's identity and has to be freely available to all voters, while the related private key must be kept secret by each candidate. The solution allows a voter to receive the list of verified public keys at the same time it gets a voting token. In this proposed scenario, the authors assume that candidates register in person, showing an identity and by communicating your public key directly to the organiser of the election. To avoid this need, a digital record can also be implemented. The process of approval of voters is carried out. Due to its nature, this step should be fully digital. The public key of a registered voter will be charged with an amount of bitcoins, which represents the election token to be spent as a vote. A voter's private key will be used by their bitcoin wallet to give his preference at the voting stage

by signing the transaction that transfers your vote. Each voter generates its public/private keys associated with their wallet. In the Voting phase, each voter will own a token that an be used to cast a vote. The voting process takes place when the voters transfer their tokens to a Bitcoin address of the candidate the voter has chosen. Authentication is performed by signing the transaction with the private key of the voter. Transmission is done when the effective execution of the transaction and confirmation is performed by checking the transactions in the Blockchain. Finally, in the Post-voting phase, is where token counting and results reporting takes place. Here the possibility of recounting can also be done as it is also one of the characteristics of this solution. Votes are counted by counting the tokens obtained by each candidate in the Blockchain. Authors also implemented a web-interface to allow the possibility of voters participating in the elections even without having wallets.

This solutions satisfies multiple properties of EVS. In particular, it supports: uncoercibility, receipt-freeness, data confidentiality and neutrality. According to the authors, any property that involves any kind of privacy is deemed problematic, since the Blockchain is public.

3.4 Secure Electronic Voting System (SecEVS)

The solution proposed in [42] reinforces the issue of security. It implements a more secure and robust electronic voting system designed for university elections. They claim to make the voting procedure happen without human intervention, while also providing security against all major attacks. They perform a security analysis review of their solution. The proposed network model was into four zones: east, west, north and south. Each zone contains a number of faculties. The purpose is to elect a student leader from among the competitors. Each faculty starts the voting process where each vote in a faculty creates a block and each block joins together to make a sub-Blockchain. Upon completion, sub-Blockchains of each faculty in each zone are joined together to make a zone level Blockchain. Each zone level Blockchain is then joined to make a single final master Blockchain that will the only one considered by the committee. They considered a pre-voting step that includes voters registration. When a voter registers in the system, he receives an ID. Voters can cast votes by encrypting their vote with the public key of University Election Commission and with his own private key. Vote information is stored into a Blockchain. As a post-voting step, when elections are over, at college level, sub-Blockchains are joined together to make master Blockchain. The election committee will check all votes cast to the master chain and will declare the final result of the election.

In terms of security analysis and data privacy authors identified a point of attack when a block data is generated and uploaded to internet server,

however they do not consider an high risk due all information is encrypted and presented in hashed form. To prevent confidentiality, they used a SHA-256 hash algorithm and encryption algorithm on vote information and if tamper occurs attacker will no be able to known the vote. In the case of duplication and falsification of votes, they created a Blockchain to override it. So they used the unique voter ID for unique identification. The Blockchain contains the hash of the previous block, the signature, and the hash of the merkle root. The signature is used to prove the authenticity and integrity of transaction data. The previous block hash is used to maintain data integrity in the Blockchain. The hash of the merkle root informs the root (source) of the voter data, therefore, they claim that their system withstands cases of duplication and forgery.

3.5 Comparison

Table 3.1 compares the related work with respect to multiple characteristics. Namely confirmation, eligibility, verifiability, unreusability, uncoercibility, integrity, uniqueness, validation, and ballot counting. Confirmation requires a process to confirm the participation in the voting process. Eligibility requires a process to verify if a voter is eligible to vote. Verifiability requires processes to allow each voter verify is own vote. Unreusability requires that each voter can only vote once, preventing future changes in votes already made. Uncoercibility means that the system must prevent any attempt to coerce the vote of others. Integrity refers to maintaining each vote as it was cast, making it difficult or impossible to tamper with. Uniqueness requires the guarantee that a voter can only cast one vote, per voting act. Validation requires a process that enables the verification of the overall voting process. Ballot counting requires a vote counting process that can be validated. At system level threats and attacks, due voting system is based on encryption and hashing if an attacker performs any type of attacks into the system, the system will identify and block them. In example if any attacker performs the data modification attack on one block the hash of the modified block wil l change and it will reflect into the whole Blockchain.

Requirements such as the confirmation of the participation in the voting process, or only allowing eligible voters to vote, or permitting voters to verify that their vote is considered in the result, or that votes are not reused are almost satisfied by all solutions. For instance, the solution proposed by Bistarelli [41] supports vote confirmation, voter eligibility, unreusability, integrity, separation, validation and requires the use of currency to cast votes. Uniqueness, ballot counting, uniqueness, and uncoercibility are not satisfied by this solution. The SecEVS solution is one that considers a reference scenario that is closer to the one adopted in our work, in the sense that both address voting in academic institutions and do not require the

Related EVS	VYV	Hardwick	Bistarelli	SecEVS
Confirmation	Y	Y	Y	N
Eligibility	Y	Y	Y	Y
Verifiability	Y	N	Y	Y
Unreusability	–	N	Y	Y
Uncoercibility	N	Y	N	N
Integrity	Y	Y	N	Y
Uniqueness	Y	Y	N	Y
Validation	Y	Y	Y	Y
Ballot counting	Y	–	N	Y

Table 3.1: Comparison of related work

expenditure of digital currency for its operation. Nonetheless, they differ because SecEVS is geared towards an large electoral vote, while, in our case, multiple voting processes are to be expected per plenary meeting.

The analysis of the existing literature and related work, complementary to previous work [8] on this subject, lead the authors to conclude that the solution proposed herein is novel and not previously addressed by our peers.

Chapter 4

Proposed Solution

The proposed solution, named Blockchain-assisted Academic Council EVS (bACEVS), aims to specifically address the recording of decisions made in academic councils. With the proposed solution, a faithful record of voting acts is maintained as well as a significant increase in usability and verifiability, of the entire decision making process, is expected.

The Polytechnic Institute of Porto (IPP) was selected as an example organisation were one could implement the now proposed solution. IPP integrates a large community of students, teachers and others, belonging to several schools spread across different geographical areas. Each of these schools has several Academic Councils (Pedagogical Council, Scientific Council, and possibly others). All schools fall under the same administrative umbrella of the IPP, for this reason a global, centralised and immutable record of all electoral acts that occur in the various Academic Councils of the various schools is important to have. At the same time, each school has some degree of independence with well-defined levels of access and decision making. Nonetheless, each school can only carry out, maintain and access its own electoral acts, without being able to interfere in the electoral acts of other schools. Based on these considerations, the Hyperledger Fabric was the select Blockchain solution, as it allows using it as a modular framework capable of maintaining a global Blockchain composed of sub-Blockchains (or channels), each with its consensus, persistence model, and identity services with independent access control and smart contracts.

4.1 Identified requirements

Please recall that, while voter anonymity (voters are not known) and voting privacy (hidden or private vote) are not required within the envisioned

reference scenario, characteristics such as voter eligibility, ballot counting, integrity, verifiability and uniqueness of votes still need to be addressed. Such considerations as lead to the identification of the following list of key requirements:

1. Support multiple Academic Councils within one organisation - With our solution, it is possible to register several academic councils, such as scientific, pedagogical councils, etc... which will then have terms of office with a well-defined duration as the members who will be part of that term.

2. Support an operation based on the fact that members are elected for a mandate (period of time) - The system should allow choosing which members belonged to a mandate, what type of member it will be, President, Secretary or normal member. Each mandate will have a well defined period of activity.

3. Support the management of Academic Councils and their members - The system must implement Create, Read, Update and Delete (CRUD) operations for all tables allowing entire management process of relevant information.

4. Schedule meetings and record the list of members participating in each one - One of the important requirements is to be able to create Meetings, schedule them, invite the members who will participate in each meeting and register your participation in each one of them.

5. Insert, modify and remove topics subject to be decided by a voting act - Possibility to have one or several topics to be discussed in each meeting and each one of them having the possibility to be voted on. Here the vote can be a yes or a no according to the position taken by each member or the issuance of a written opinion regarding that topic.

6. Support the association of external digital documents that are relevant to the topics to be voted on - One important feature of our system is the total digitization of all documents related to electoral acts. Thus, we will have to provide the system with the possibility of associating digitalized physical documents either to each topic that will be voted on or to each meeting as a whole, as in this area it is very common to have physical documents from various sources.

7. Register the participation of all members in a voting act, including their vote and the overall decision - The participation of each member is critical to the success of each voting process, thus, the record of the participation of each member shall exist as well as the position taken by each one in each voting process.

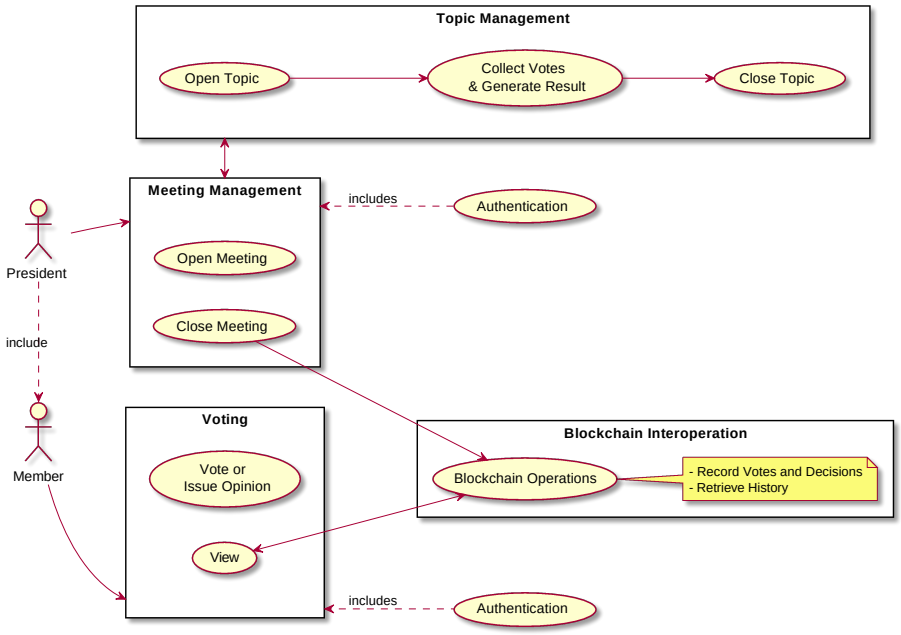


Figure 4.1: Main Use Cases

8. Guaranty the immutability of records pertaining voting acts - As a main objective, the guarantee that the final result of a voting process that took place in a given meeting cannot be changed needs to be ensured. Thus, the use of Blockchain with its characteristic of immutability will be fundamental for the success of this project.

4.2 Use cases

The crossed analysis of the reference scenario (see Fig. 1.1) with these key requirements resulted in the definition of a set of use cases. The main use cases that were identified are depicted in Fig. 4.1. These comprise two actors: President and Member. The Member will be able to cast a vote or issue an opinion when a topic is open for voting. Members will also be able to view the information recorded on Blockchain. The President, besides meeting management operations that were not represented for the sake of simplicity, can start a meeting, manage topics that will be voted on, and close meetings. The President, at the topic level, can open a topic for discussion within a meeting, collect its votes and generate the decision result that will imply the topic being closed and for its decision to sent to the Blockchain, as soon as the President closes the meeting.

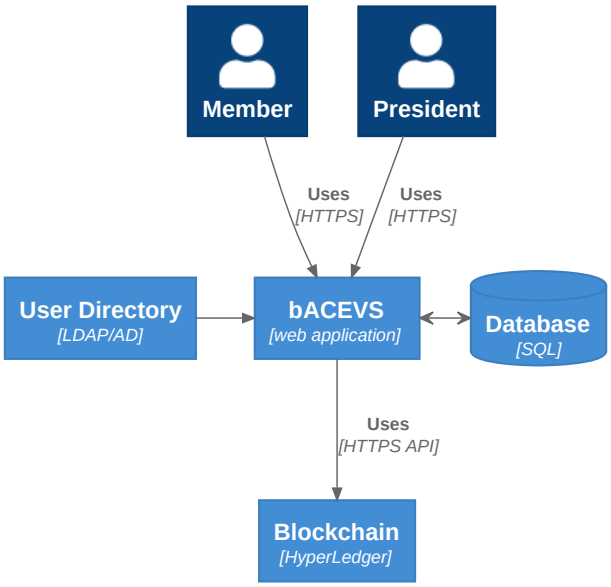


Figure 4.2: Architecture of the proposed solution

4.3 Proposed Architecture

Figure 4.2 shows the architecture of the proposed solution. It comprises six components, these being: the Members, the President, a User Directory, the bACEVS, a Database, and the Blockchain.

A Member represents a real person of the Academic Council that is eligible to participate in meetings by voting or issuing opinions on a presented topic.

The President also represents a real person but one with a specific role within an Academic Council. The President is eligible to participate in meetings, voting and issuing opinions on topics. Additionally, this is the entity responsible for the coordination of meetings, performing actions such as opening meetings, presenting topics, closing topics, gathering votes or issued opinions, and closing meetings.

The bACEVS system consists of a web-based application used to interact with all members of Academic Councils, supporting all relevant functionalities. It can be seen as the front-end application from the members perspective, and as the gateway to thee database of Institution, to the users directory database and also includes an API to the Blockchain. This API is responsible for the interface between the Hyperledger Blockchain and the

bACEVS.

The User Directory, consists of a database of users. It will be used by the proposed solution as a location for user related information, including authentication related information, such as usernames and password. This component is expected to already exist in the organisations either as an Lightweight Directory Access Protocol (LDAP) or Active directory (AD) database where bACEVS will authenticate members.

The Database consists of a common relational database with Structured Query Language (SQL) support. It will used to store information about meetings, topics, the composition of the Academic Councils, member mandates and decisions taken in meetings. The decisions will be structured in JavaScript Object Notation (JSON) format.

Ultimately, the Blockchain will be used to store all decisions made within the meetings of the Academic Councils. The decisions will be stored in an immutable , verifiable and secure way, in the Blockchain. An example Blockchain can be the Hyperledger Fabric.

4.4 Database Entities

At database level, we propose to implement it as a MySql database comprising the tables: User, School, Vote, Meeting, MeetingDetails, Mandate, MandateDetails Vote, Topic, Decision, AcademicCouncil and ActivityLog.

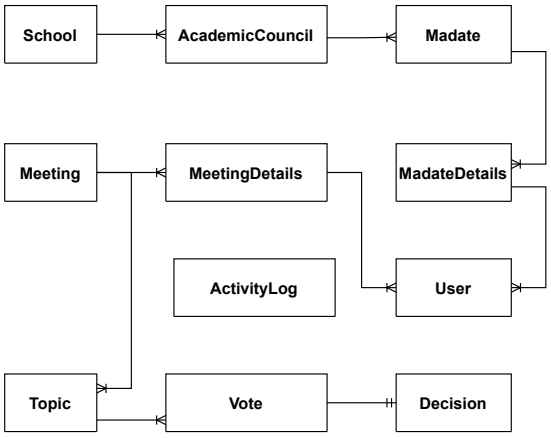


Figure 4.3: Database entities

The User table will store user or member information like personal details and is Blockchain private key, which will be used to access the Blockchain. Users will be authenticated using a preexisting Active Directory infrastructure.

The School table will store information about the institution, likewise

the AcademicCouncil table will store information about existing Academic Councils.

The Mandate and MandateDetails tables maintain information about mandates of the member users, such as type and duration.

The Meeting table stores information about scheduled meetings, including references to which topics, from the Topic table, are to be included in each meeting and its status (closed or not). Each topic will be created in table Topic and associated with a specific meeting.

The Vote table stores vote related information. Not all topics require a vote from the members, some topics only require the issuing of an opinion on behalf of the council. An hash value of each vote and voted topic is calculated and then recorded in an hash array of decisions.

The Decision table stores all information about decisions and is saved in a file in JSON. There, the yea, nay and abstention decisions, of the members, will be stored. A global hash of this information will be sent to Blockchain, comprising a reference to all votes, topics and decisions.

4.5 Diagrams and Specification

Table 4.1 shows the adopted notation. The letter P represents the president, letter M represents a member, letter S represents the proposed system, and letter B refers to the Blockchain. ID_X represents the unique identifier of member X. N_X represents the nonce generated by member X. T_s represents a timestamp. SEK represents a session encryption key. $PrivK_X$ represents the private key of member X. D represents the decision message. $HMAC(D, SEK)$ represents the result of a secure keyed-Hash Message Authentication Code (HMAC) function with input D and the key SEK.

ID_X	ID of member X
T_s	Timestamp
N_X	Nonce generated by X
$PubK_X$	Public key of member X
$PrivK_X$	Private key of member X
SEK	Session Encryption Key
D	Decision message
$HMAC(D,SEK)$	Authentication code of data D with key SEK

Table 4.1: Adopted notation

```
1  "topicID": "055919b9-44eb-4771-853c-34cd95c697a3",
```

```
2   "positiveDecisions": {
3     "members": [
4       ["088f0fd4-e8c5-4a0e-b042-2517b8e95971"],
5       ["bdcb984e-7fe8-4be5-bc19-79ee97368331"],
6       ["36989b61-6906-438d-8d32-15942fee92b0"],
7       ["477698a6-2f31-11ec-8d3d-0242ac130003"]
8     ]}
9 }
```

Listing 4.1: Sample decision

The overall sequence diagram, referencing key elements of the proposed solution and their interactions, is depicted in Figure 4.4.

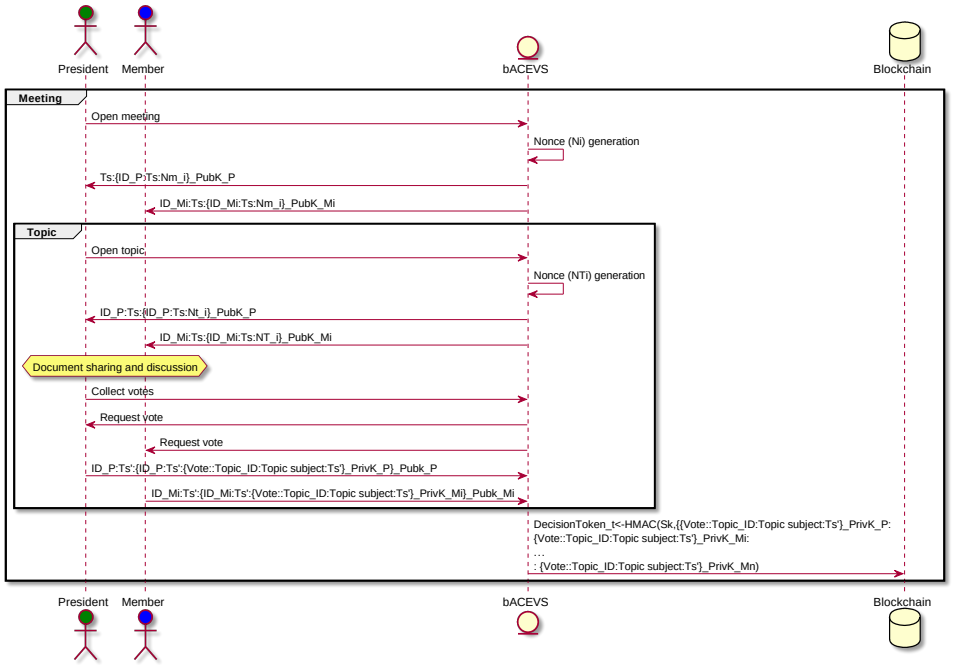


Figure 4.4: Meeting sequence diagram

A meeting starts when it is opened by the President or Secretary. In this step, a timestamped nonce is generated to identify this meeting action and then they present the topics that will be discussed or voted on, making them available to all members. Since a session can be interrupted or suspended after it has been opened, the recording of the decisions is taken topic by topic. This way, meeting can be resumed on a later date, without having members to vote on all topics again, and just continuing the session at the

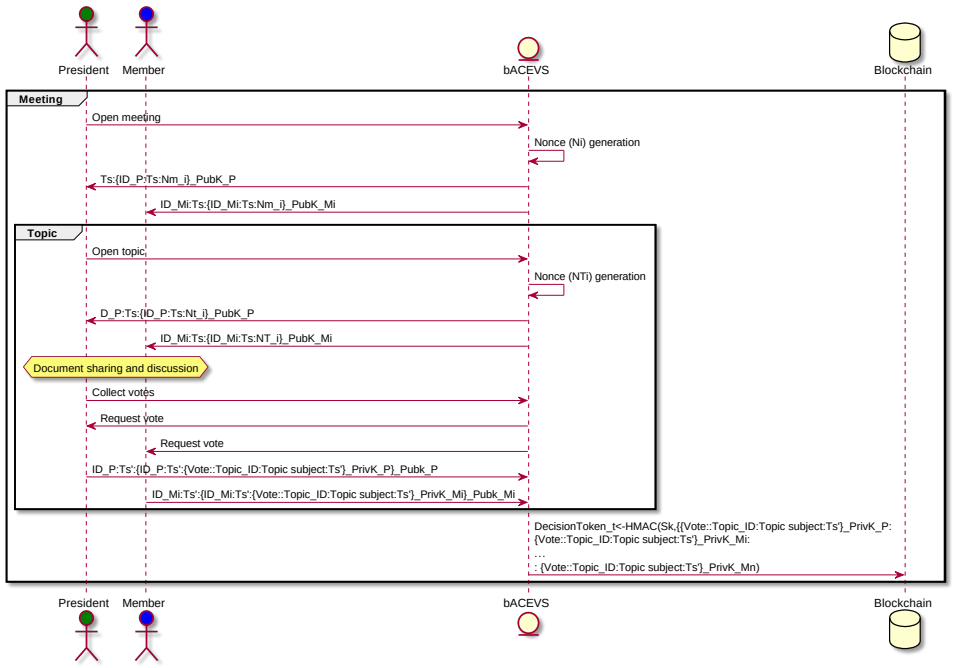


Figure 4.5: Decision token generation

topic that was suspended. A notify message is sent to all members who participate in meeting alerting meeting is open following the presentation and opening of each topic to be voted on, by the President. At this point a new timestamped nonce is generated again to identify topic action and topic discussion start by presenting and sharing documents by Blockchain Academic Council Electronic Vote system (bACEVS) to all voting members. When topic discussion ends, President starts a vote collection action in bACEVS and this sends topic votation request message to all participants. bACEVS collect votes made by each voter in this mobile device and generates a JSON file with result. Generated final decisions JSON file is sent to Blockcahin to be stored in. After an end of successful topic votation process, President closes voted topic a starts all over again until no more topic are available to be voted. Finally President closes the meeting on bACEVS and a notify message informing that meeting is over is sent to all participants.

Figure 4.5, whenever a topic is closed, a decision record is created, stored in the database and its hash is sent to the Blockchain for storage. This decision is a JSON formatted data structure. Listing 4.1 shows a sample decision data structure with positive decisions. It includes the topic ID and the ID's of members who voted positively.

4.6 Blockchain interoperation

As described before, a smart contract was implemented to interact with ledger, defining the executable logic which generates new facts to be added to the ledger. The Fabric ledger is maintained by each peer including blockchain and world state. Transaction reads and writes sets and channel configurations are written in blockchain. The world state can be wither a LevelDB by default or CouchDB. LevelDB is a simple key/value store and CouchDB is a document store where complex queries are allowed. Smart contract is responsible to **put**, **get** and **delete** states in world state and make queries to the immutable blockchain record of transactions. A **put** creates new business object or modifies it, a **get** query the ledger to retrieve information about current business object and **delete** removes a business object from current state of the ledger, maintaining its history. Generally smart contract decides what information is stored in world state. In listing 4.2 is the example of our smart contract to interact with Blockchain. A JSON array is sent to blockchain, containing **TopicID** which is the identification on bACEVS of voted topic and **HashDecisionFile** referring original hash of decision file stored in bACEVS database.

```
1  type Decision struct {
2      topicid       string 'json:"TopicID"'
3      hashdecisonfile  string 'json:"HashDecisionFile"'
4  }
```

Listing 4.2: bACEVS main data structure

Possible operations will be described in developed smart contract "DecisionContract" where example information is instantiated when smart contract is deployed. It was developed in a test environment recurring to IBM BLOCKCHAIN PLATFORM for Visual Studio Code (vscode). In listing 4.3 developed smart contract in Javascript language is presented and possible operational functions are defined there. Function **initLedger** initialises the ledger with some information about decisions of voted topics as example.

```
1
2  const { Contract } = require('fabric-contract-api');
3  class DecisionContract extends Contract {
4  async initLedger(ctx) {
5      console.info('============= START : Initialize Ledger
       ===========');
6      const decisions = [
7          {
```

```
8             topicid: 'T151120211435a9ab7fc521fd',
9             hashdecisionfile: '477698a62f3111ec8d3d0242ac130003',
10        },
11
12        {
13            topicid: 'T1511202114357f505aad471f',
14            hashdecisionfile: '36989b616906438d8d3215942 fee92b0',
15        },
16
17        {
18            topicid: 'T1511202116584903d950588f',
19            hashdecisionfile: 'Tobdcb984e7fe84be5bc1979ee97368331'
    ,
20        },
21
22        {
23            topicid: 'T1711302114367d505bbd417a',
24            hashdecisionfile: '088 f0fd4e8c54a0eb0422517b8e95971',
25        },];
26
27    for (let i = 0; i < decisions.length; i++) {
28        decisions[i].docType = 'decision';
29        await ctx.stub.putState('DECISION' + i, Buffer.from(JSON.
    stringify(decisions[i])));
30        console.info('Added <--> ', decisions[i]);
31        }
32        console.info('============= END : Initialize Ledger
    ===========');
33    }
34
35    async decisionExists(ctx, decisionId) {
36        const buffer = await ctx.stub.getState(decisionId);
37        return (!!buffer && buffer.length > 0);
38    }
39
40    async createDecision(ctx, decisionId, value) {
```

```
41          const exists = await this.decisionExists(ctx, decisionId);
42          if (exists) {
43              throw new Error(`The decision ${decisionId} already
    exists`);
44          }
45          const asset = { value };
46          const buffer = Buffer.from(JSON.stringify(asset));
47          await ctx.stub.putState(decisionId, buffer);
48      }
49      async readDecision(ctx, decisionId) {
50          const exists = await this.decisionExists(ctx, decisionId);
51          if (!exists) {
52              throw new Error(`The decision ${decisionId} does not
    exist`);
53          }
54          const buffer = await ctx.stub.getState(decisionId);
55          const asset = JSON.parse(buffer.toString());
56          return asset;
57      }
58      async updateDecision(ctx, decisionId, newValue) {
59          const exists = await this.decisionExists(ctx, decisionId);
60          if (!exists) {
61              throw new Error(`The decision ${decisionId} does not
    exist`);
62          }
63          const asset = { value: newValue };
64          const buffer = Buffer.from(JSON.stringify(asset));
65          await ctx.stub.putState(decisionId, buffer);
66      }
67      async deleteDecision(ctx, decisionId) {
68          const exists = await this.decisionExists(ctx, decisionId);
69          if (!exists) {
70              throw new Error(`The decision ${decisionId} does not
    exist`);
71          }
72          await ctx.stub.deleteState(decisionId);
```

```
73      }
74  }
75  module.exports = DecisionContract;
```

Listing 4.3: bACEVS smart contract for Decisions

Next step is to deploy it into local Fabric test environment and if it succeeds test transactions can be made invoking described functions in smart contract. Figure 4.6 shows the output log where information about a successful deployment occurred for our smart contract.

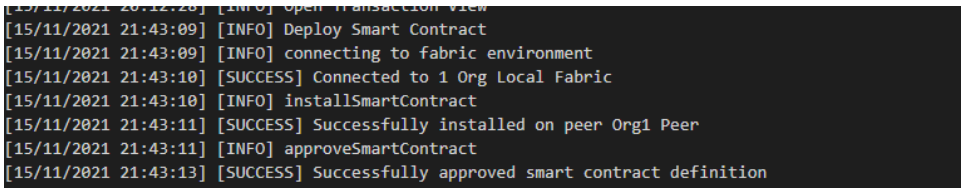


Figure 4.6: Local Fabric output log

4.7 Security considerations

As security, for login process our solution will integrate existing User Directory services, such as an AD infrastructure. The application will also assure that votes can only occurs whenever a session an been opened, checking all meeting related information. Whenever a meeting is closed, the system will not allow any more votes to be cast.

The session integrity resides on a secret nonce, generated by the system with the session opening, and securely distributed to all Members. The nonce will be used to sign each vote or issued opinion, satisfying this requirement. Anonymity in decisions made by member is not required internally, but externally, institution decisions are confidential.

Chapter 5

Validation and discussion

This chapter presents and discuss the results achieved from the implementation of a prototype of the proposed solution. The bACEVS web application being the main element of the implemented prototype.

5.1 Implemented prototype

The prototype that was implemented to validate the proposed solution is depicted in Figure 5.1. It comprises a bACEVS server, an authentication server and the President's personal computer. The prototype was setup in a PC with 16 Gigabytes (GB) of memory Random Access Memory (RAM) running MS Windows 10 operating system with a Intel I5 processor with 240 GB Solid State Drive (SSD).

Components where run in virtual environments supported in Oracle Virtual Box, version 6.0.14, and it comprised the authentication server Virtual Machine (VM), running Windows Server 2019 with AD services, the bACEVS server VM, running Community ENTerprise OS (CENTOS), and a normal laptop running MS Windows 2010 to simulate a client device operated by an Member with a president role.

The bACEVS Server was built using a Linux, Apache, MySQL, PHP (LAMP) stack (Linux, APACHE, MySQL, and PHP). The used versions were LAMP version 7.9.2009, CENTOS version 7, and PHP version 7.3.31. The choice for this bundle of software was made because its an Open Source software bundle and APACHE, a web server software that delivers web content through the internet and runs actually over 54%[43] of all web-servers in the world.

Following prerequisites need to be installed to run prototype in a test network environment available from Hyperledger Fabric sources for creating

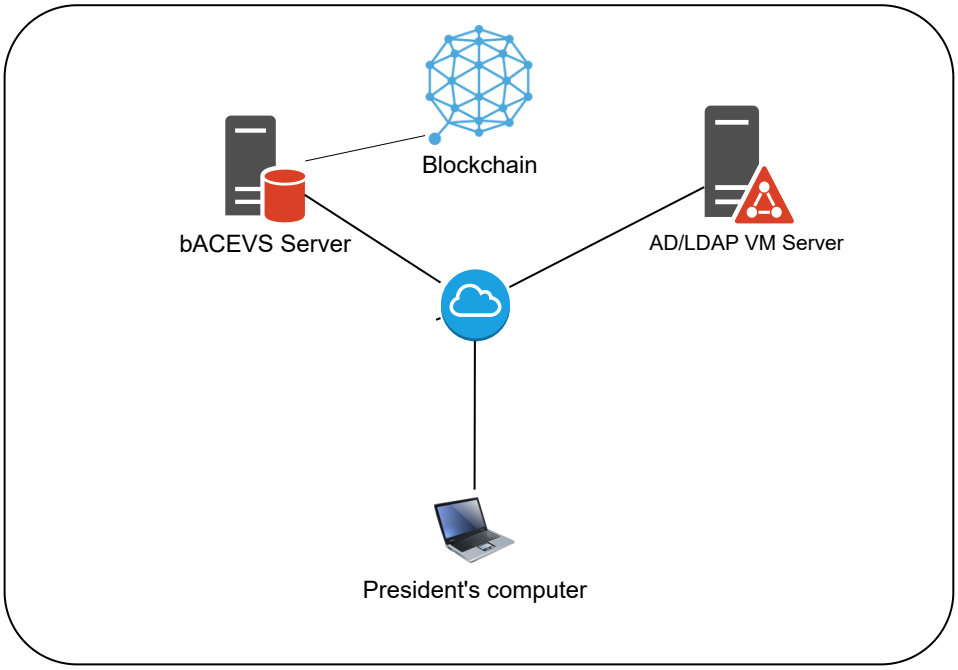


Figure 5.1: Test environment

reference network in Ubuntu Based installation:

```
1  $sudo apt-get install curl
2  $sudo apt-get install golang-go
3  $export GOPATH=$HOME/go
4  $export PATH=$PATH:$GOPATH/bin
5  $sudo apt-get install nodejs
6  $sudo apt-get install npm
7  $sudo apt-get install python
8  $sudo apt-get install docker
9  $curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo
       apt-key add -
10 $sudo add-apt-repository "deb [arch=amd64] https://download.docker
       .com/linux/ubuntu $(lsb_release -cs) stable"
11 $sudo apt-get update
12 $apt-cache policy docker-ce
13 $sudo apt-get install -y docker-ce
14 $sudo apt-get install docker-compose
```

```
15 $sudo apt-get upgrade
```

Listing 5.1: Environment network test preparation

Next commands downloads latest release of Fabric samples, docker images, and binaries. First command will provide a clone of fabric-samples into a local folder, for convenience we are positioned in folder "fabric-samples". Then with instruction indicated in line 2 command to create a test network is invoked with flag **up** and **createChannel -c** giving indication to create a channel with personalised name, in this case "mychannel". Then we bring up network and make it available with command in line 3.

```
1 $curl -sSL https://bit.ly/2ysbOFE | bash -s
2 $ ./network.sh up createChannel -c estgchannel -ca
3 $ ./network.sh up estgchannel
4 $./network.sh deployCC -v -c mychannel -ccn basic -ccp ../asset-
    transfer-basic/chaincode-go -ccl go
5 $ export PATH=${PWD}/../bin:$PATH
6 $ export FABRIC_CFG_PATH=$PWD/../config/
```

Listing 5.2: Hyperledger Fabric setup

Then test network is ready to interact with bACEVS. To check network test environment see next commands. In figure 5.2, we can see command result with a list of Docker containers that are running in test network environment.
Command to check which Docker containers are running on test machine:

```
$ docker ps -a
```

Figure 5.2: Command results: loaded containers

After running Docker with default containers loaded, command to create a channel is invoked and to give a customised name to it a flag -c must be used. In this phase, channel names must be written in lowercase and without any special characters or spaces. Command illustrated in 5.3 makes **mychannel** creation.

Command to create a customised channel and joining peers:

```
$ $ ./network.sh up createChannel -c mychannel -ca
```

Figure 5.3: Command results: channel creation

5.2 Validation

The list of requirements that were identified for the proposed solution can be found in Section 4.1. The first requirement (R1) states that the proposed solution must support multiple Academic Councils within one organisation. To achieve this requirement, firstly the proposed solution was implemented in way that integrates with external authentication serves, such as an LDAP or AD. This way, the bACEVS dos note need to maintain user profiles or passwords, increasing its overall security. Each time a user logs in, a validation is made in AD server. Figure 5.4 shows the implemented login form.

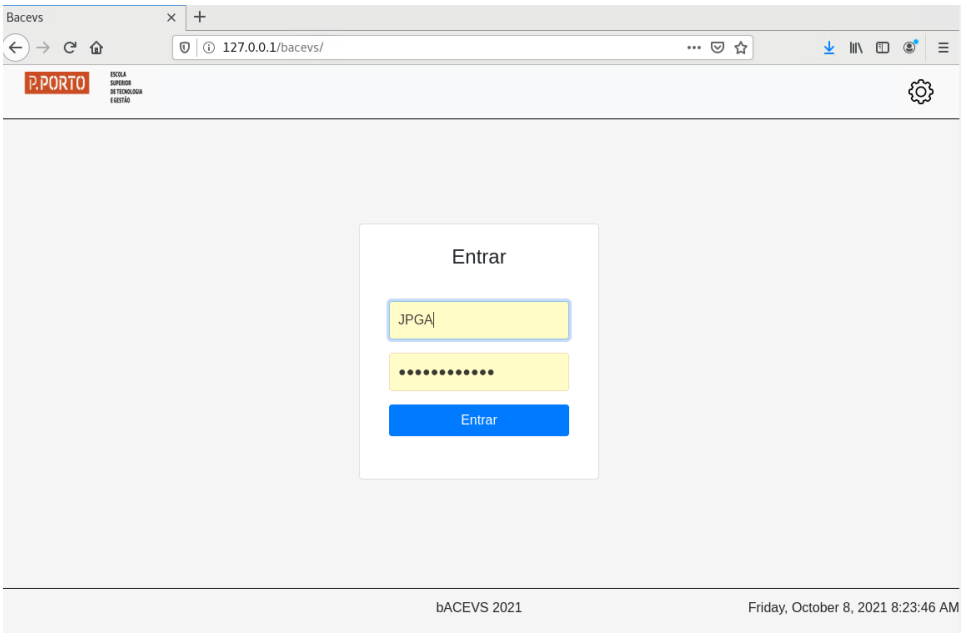


Figure 5.4: Login form of the bACEVS

This support is added through the "login.php" file, shown in Listing 5.3. The process starts by validating if we are dealing with POST message that contains a username and password collected by login form and in then a connection to indicated LDAP server (192.168.51.5) is made. In case of successful bind a query to users database is made filtered by "SAMAccountName" and an array with specific fields is filled (Common Name (CN), fullName, mail, and displayName) to be used by bACEVS. It happens only if a valid comparation occurs between username and password sent in POST method and LDAP stored info, otherwise an "Invalid username or password!" message is displayed.

```
1 if(isset($_POST['submit'])){
2     $username=trim($_POST['userN']);
3     $password=trim($_POST['pass']);
4     $valores= array();
5     $ldap = ldap_connect("192.168.91.5") //JPGADOMAIN.LOCAL
      192.168.91.5
6     or die("Could not connect to LDAP server.");
7     ldap_set_option($ldap, LDAP_OPT_PROTOCOL_VERSION, 3);
8     $bind = ldap_bind($ldap, 'JPGADOMAIN\\'.$username, $password);
9     if ($bind) {
```

```
10          $dn = "DC=JPGADOMAIN,DC=LOCAL";
11          $filter="SAMAccountName=$username";
12          $justthese = array("CN","fullName", "mail","displayName");
13          $sr=ldap_search($ldap,$dn, $filter,$justthese);
14          $info = ldap_get_entries($ldap, $sr);
15         //ECHO  $info[0]['dn'];
16          for ($i=0; $i<1; $i++){
17              echo $info[$i]['cn'][0];
18              echo $info[$i]['displayname'][0];
19              $_SESSION['username'] = $info[$i]['displayname'][0];
20          }
21         header('location:'.$path.'/home.php');
22      } else {
23          $message = "Invalid username or password!\\nPlease try
        Again.";
24          header('location:'.$path.'/index.php?msg='.$message);
25      }
26      }
27  }
```

Listing 5.3: Login source code

Moreover, by opting for the Hyperledger Fabric as the adopted Blockchain, Requirement 1 can also be partially addressed. Hyperledger Fabric supports segmented use by being a permissioned Blockchain and be allowing for the creation of channels, each channel having its own participants. Thus, one can setup a multi-node Hyperledger Fabric infrastructure. One node per school, all storing the information of all councils, but still separating information access privileges using segregation by channels.

For requirement (R2) bACEVS supports an operation based on the fact that members are elected for a mandate (period of time). It allows to create a record for mandate with a start and end date. After create a mandate it will be available for update with the association of users to be part of this mandate. In figure 5.5 an example of implemented form is shown and in point 1 and 2 we can see the start and end date associated with its mandate. At point 3 we have an action button that opens a mini modal form (a window that forces the user to interact with it before he can go back to using the parent application) to choose users from table "User" to be added and what will be their roles in selected mandate (point 4) .

At requirement (R3), in all tables referred in figure 5.8 CRUD operations

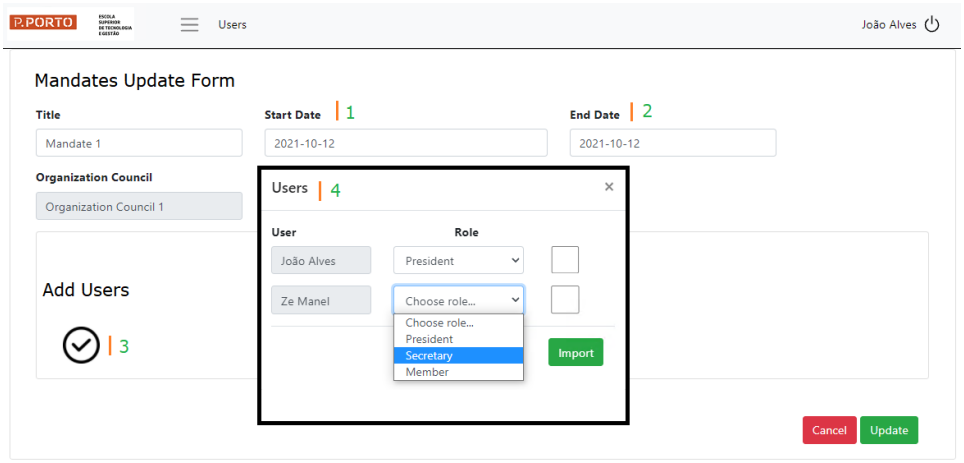


Figure 5.5: Mandates update form

where implemented. We can add records, view, update and delete them. Menu entry "Tables" is where principal CRUD table operations are located. Its possible to access table users to make its management process including users importation from institution AD server. We can create Organizations an Organization Councils too. Requirements for (R4) where implemented and to fulfill them bACEVS has feature to insert a record for a meeting in specific date an display it in a calendar table. For convenience a better management we decide to show it in background of maim menu. Next figure demonstrates main menu of bACEVS and a scheduled meetings example being visible in calendar table format.

Validating (R5), topics can be inserted individually in meeting, one or more, ordered by order to be discussed. In "Meeting" form we have general information regarding selected meeting and an plus blue button to add topics, permitting its association with it. Figure 5.6 shows multiple topics added in a meeting.

To achieve the goal in requirement (R6), the possibility to eliminate physical documents is made by the association of any type of digitized documents to a topic to be considered in its discussion. In figure 5.7 is illustrated the addition of new topic with a file attached that is stored in database.

Validating requirement (R7), in table **MeetingDetails** information about members of a meeting are stored and a control of presences is made by checking column "presence" when a meeting starts. With this mechanism, we are able to control who is actually present at the meeting. Votes are registered in database on vote table, allowing to know the position taken by each member in each decided or discussed topic.

As a summary of previous validated requirements, bACEVS supports an operation based on the fact that members are elected for a mandate in

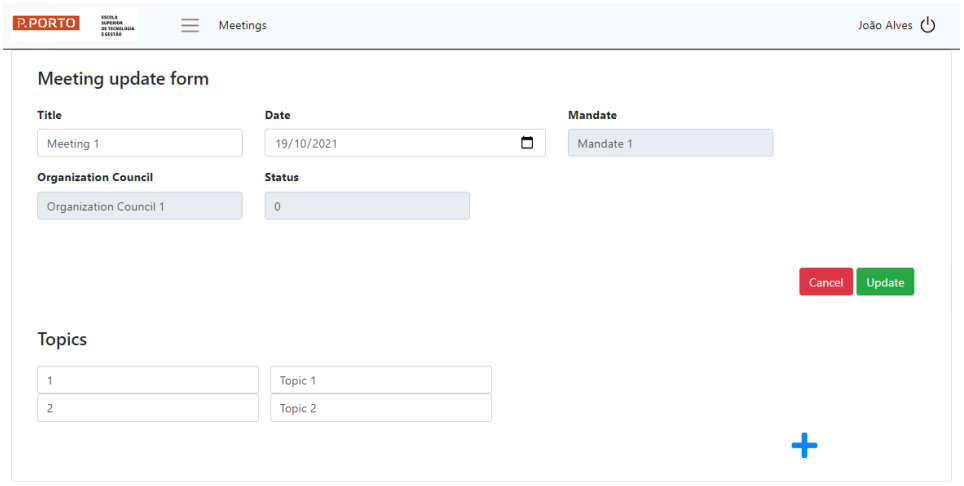


Figure 5.6: Meeting update form

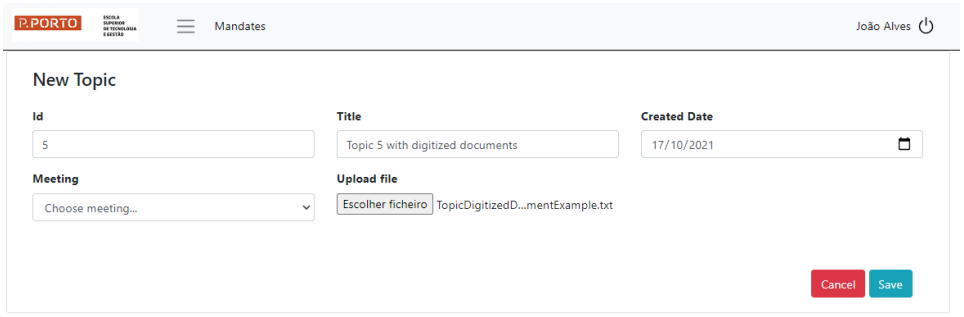


Figure 5.7: New Topic form

a period of time. It allows to create a record for a mandate with a start and end date well defined. Supports multiple Academic Councils within one organisation with multiple records permitted to be added. In management of Academic Councils we can add users from "User" table and assign a role in Academic Council such as president, secretary or member. In meeting management is possible to "Schedule meetings" with feature of insert a record for a meeting in specific date an display it in a calendar table. For convenience a better management we decide to show it in background of maim menu. Figure 5.8 demonstrates main menu of bACEVS and principal components are presented:

1. Menu entry "Tables" is where principal CRUD table operations are located. Here is possible to access table users to make its management process including users importation from institution AD server. We can create Organizations an Organization Councils too.

2. On "Management" menu entry is possible to register new mandates and select members to be part of it, assigning a role at same time. Meetings are scheduled here too, being visible in calendar table format.

3. In background of main application we have a calendar table displaying next meetings.

4. On top right side logged in information is displayed with possibility to a user make logoff.

5. Menu icon with functionality of toggle between wide display with left menu hidden or showing opening down menu possibilities

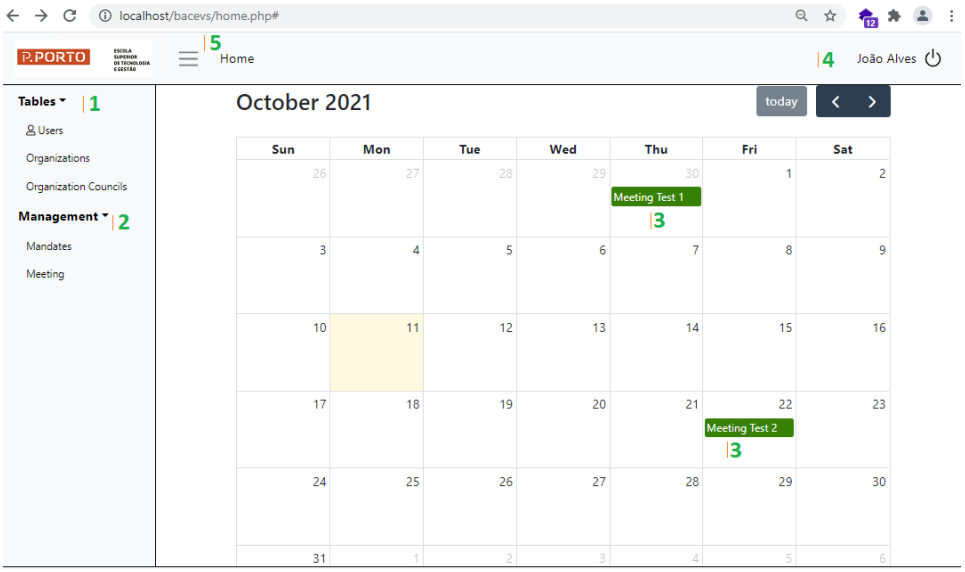


Figure 5.8: Main menu

Meeting operations are simplified by interacting with calendar object by selecting the day with scheduled meetings. When a day is selected in right side of screen is possible to control the meeting by opening session (in this point a email notification is sent to all registered and allowed members who will participate on it). In figure 5.9 is illustrated a meeting opening operation. In this case, President or Secretary opened meeting and a mail notifications was generated and sent to all participants as well a topics list related to this meeting is presented bellow.

At this point, logged members in mobile browser application can interact with meeting session by voting or issue opinion when a topic is opened to be voted. In figure 5.10 is possible to vote or issue opinion on a topic.

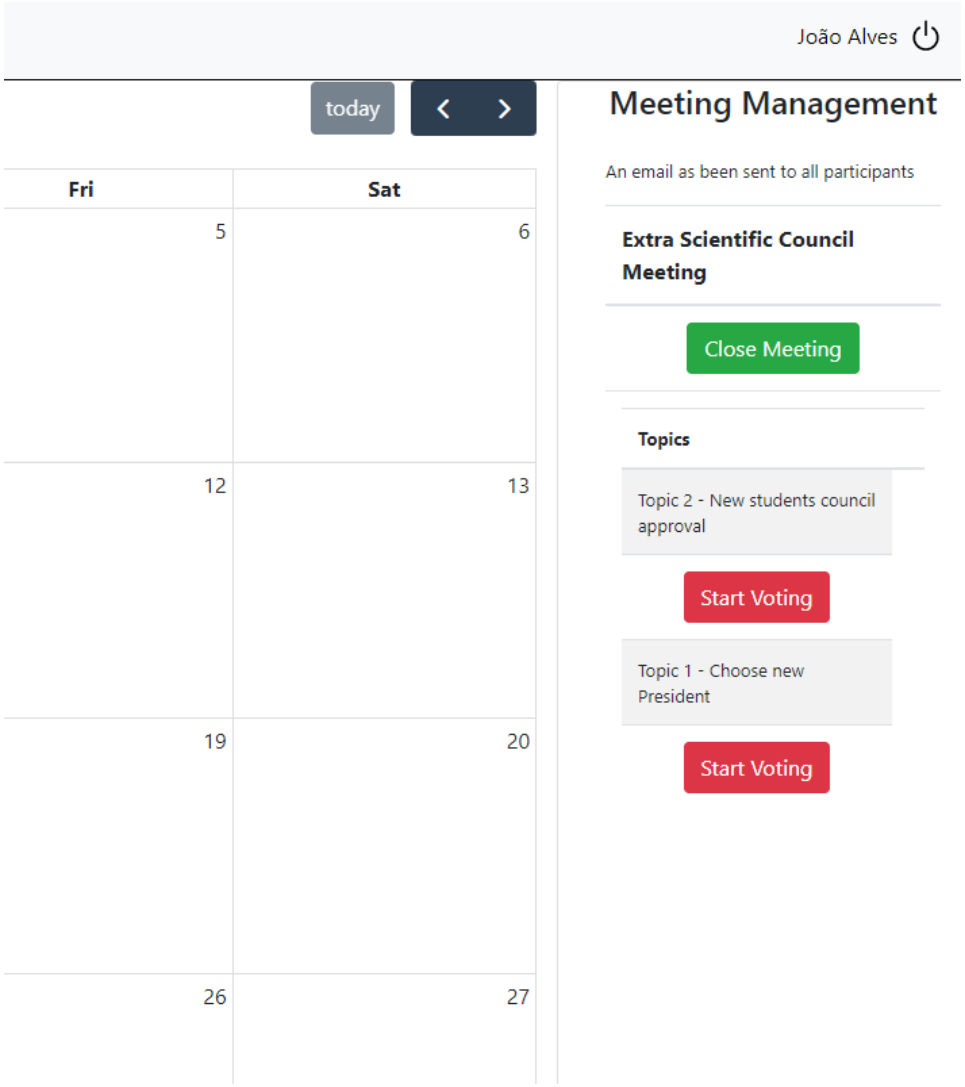


Figure 5.9: Managing meetings

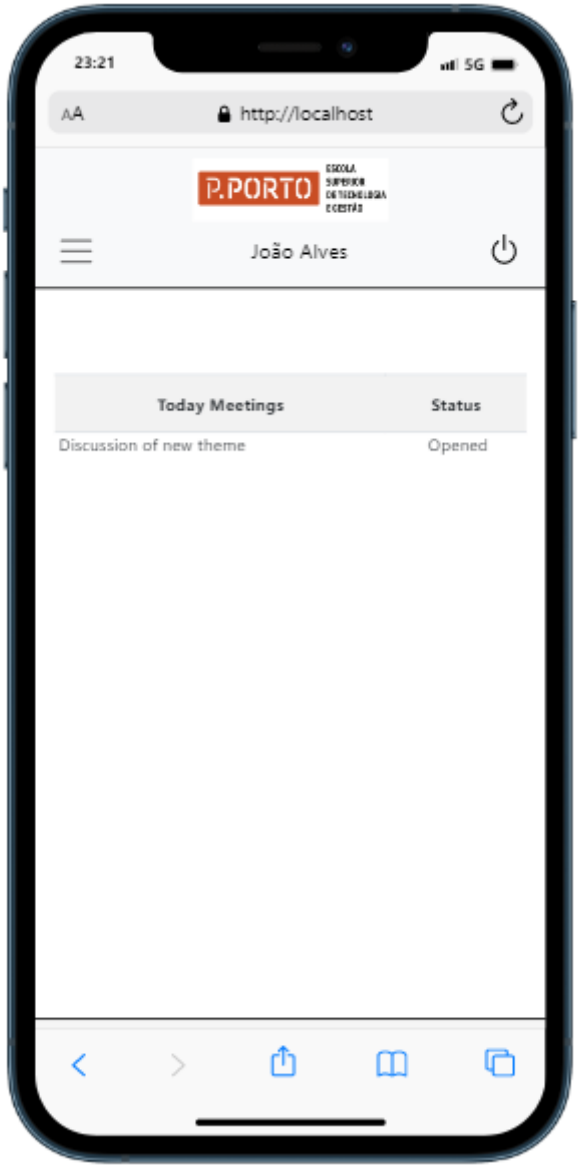


Figure 5.10: Mobile application

Chapter 6

Conclusion

Today's governments support and encourage the implementation of clean digital technologies creating commitments that will help accelerate the ecological and digital transition, specially in digitalization. With this work we described Blockchain technology and a review of EVS was made. Proposed solution pretends to be a valid option to be used by Academic Councils in Academic Institutions in digitization processes and giving the possibility to record decisions taken by them in their Academic Councils meetings on a system with tamper proof. For future work, we pretend to develop API interface to interact with Hyperledger Fabric blockchain automatically and test the use of this solution in a real multiple schools scenario considering it a interesting point, to test its functionality.

Bibliography

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system." [Online]. Available: www.bitcoin.org

[2] A. L. Delbecq and A. H. Van de Ven, "A group process model for problem identification and program planning," *The Journal of applied behavioral science*, vol. 7, no. 4, pp. 466–492, 1971.

[3] D. A. Gritzalis, "Principles and requirements for a secure e-voting system," *Computers and Security*, vol. 21, no. 6, pp. 539–556, 2002.

[4] E. Brynjolfsson, J. J. Horton, A. Ozimek, D. Rock, G. Sharma, and H.-Y. TuYe, "Covid-19 and remote work: An early look at us data," National Bureau of Economic Research, Tech. Rep., 2020.

[5] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[6] F. Casino, T. K. Dasaklis, and C. Patsakis, "A systematic literature review of blockchain-based applications: Current status, classification and open issues," *Telematics and Informatics*, vol. 36, pp. 55–81, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0736585318306324

[7] M. Bartoletti and L. Pompianu, "An empirical analysis of smart contracts: platforms, applications, and design patterns," in *International Conference on Financial Cryptography and Data Security*. Springer, 2017, pp. 494–509.

[8] J. Alves and A. Pinto, "On the use of the blockchain technology in electronic voting systems," in *International Symposium on Ambient Intelligence*. Springer, 2018, pp. 323–330.

[9] "Global glossary of blockchain terms 2.0 in 5 languages – blockchain training alliance." [Online]. Available: https://blockchaintrainingalliance.com/pages/glossary-of-blockchain-terms

[10] "Consensus mechanism (cryptocurrency) definition." [Online]. Available: https://www.investopedia.com/terms/c/consensus-mechanism-cryptocurrency.asp

[11] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[12] H. Handschuh, *SHA Family (Secure Hash Algorithm)*. Boston, MA: Springer US, 2005, pp. 565–567. [Online]. Available: https://doi.org/10.1007/0-387-23483-7$_3$88

[13] J. Norton, "Blockchain Easiest Ultimate Guide To Understand Blockchain," 2016.

[14] "Block chain — bitcoin." [Online]. Available: https://developer.bitcoin.org/devguide/block$_c$*hain.html*

[15] M. Swan, *Blockchain: Blueprint for a new economy.* " O'Reilly Media, Inc.", 2015.

[16] Deloitte, "Key charactertistcs of the blockchain." [Online]. Available: https://www2.deloitte.com/content/dam/Deloitte/in/Documents/industries/in-convergence-blockchain-key-characteristics-noexp.pdf

[17] C. V. Helliar, L. Crawford, L. Rocca, C. Teodori, and M. Veneziani, "Permissionless and permissioned blockchain diffusion," *International Journal of Information Management*, vol. 54, p. 102136, 10 2020.

[18] N. Szabo, "Nick szabo – smart contracts: Building blocks for digital markets." [Online]. Available: https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart$_c$*ontracts*$_2$*.html*

[19] M. G. Vigliotti, "What do we mean by smart contracts? open challenges in smart contracts," *Frontiers in Blockchain*, vol. 3, 2021.

[20] "New technologies - equipment leasing finance foundation." [Online]. Available: https://www.leasefoundation.org/industry-resources/new-technologies/

[21] "Blockchain beyond the hype: Perspective for the equipment leasing and finance industry." [Online]. Available: www.weforum.org.

[22] A. Kormiltsyn, C. Udokwu, K. Karu, K. Thangalimodzi, and A. Norta, "Improving healthcare processes with smart contracts," vol. 353. Springer Verlag, 2019, pp. 500–513.

[23] "Is intellectual property ready for blockchain?" [Online]. Available: https://dai-global-digital.com/is-intellectual-property-ready-for-blockchain.html

[24] "maj — 2021 — dbi – dynamic business institute." [Online]. Available: http://www.dbikursus.dk/2021/05/

[25] "Smart contracts and chaincode — https://hyperledger-fabric.readthedocs.io/." [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.2/smartcontract/smartcontract.htmlsmart-contract

[26] B. Binder, . Nadja, R. . Krimmer, G. . Wenda, and D.-H. . Fischer, "International standards and ict projects in public administration: Introducing electronic voting in norway, estonia and switzerland compared," *Halduskultuur: The Estonian Journal of Administrative Culture and Digital Governance*, vol. 19, pp. 8–22, 2019.

[27] K.-H. Wang, S. K. Mondal, K. Chan, and X. Xie, "A review of contemporary e-voting: Requirements, technology, systems and usability," *Data Science and Pattern Recognition*, vol. 1, no. 1, pp. 31–47, 2017.

[28] T. de Balthasar and J. Hernandez-Castro, "An analysis of bitcoin laundry services," in *Nordic Conference on Secure IT Systems*. Springer, 2017, pp. 297–312.

[29] Y. Takabatake, D. Kotani, and Y. Okabe, "An anonymous distributed electronic voting system using zerocoin," *IEICE technical report*, 2016.

[30] A. Ben Ayed, "A Conceptual Secure Blockchain Based Electronic Voting System," *International Journal of Network Security & Its Applications*, vol. 9, no. 3, pp. 01–09, 2017. [Online]. Available: http://aircconline.com/ijnsa/V9N3/9317ijnsa01.pdf

[31] I. Miers, C. Garman, M. Green, and A. D. Rubin, "Zerocoin: Anonymous distributed e-cash from bitcoin," in *Security and Privacy (SP), 2013 IEEE Symposium on*. IEEE, 2013, pp. 397–411.

[32] Y. Wu, "An e-voting system based on blockchain and ring signature," *Master. University of Birmingham*, 2017.

[33] P. C. Jason and K. Yuichi, "E-voting system based on the bitcoin protocol and blind signatures," *TOM*, vol. 10, no. 1, pp. 14–22, 2017.

[34] D. Chaum, "Blind signature system," in *Advances in cryptology*. Springer, 1984, pp. 153–153.

[35] P. Tarasov and H. Tewari, "Internet voting using zcash," Cryptology ePrint Archive, Report 2017/585, 2017, accessed: 2017-06-29. [Online]. Available: http://eprint.iacr.org/2017/585.pdf

[36] D. Hopwood, S. Bowe, T. Hornby, and N. Wilcox, "Zcash protocol specification," 2016-1.10. Zerocoin Electric Coin Company, Tech. Rep., 2016.

[37]

[38] Y. Abuidris, R. Kumar, and W. Wenyong, "A survey of blockchain based on e-voting systems," *ACM International Conference Proceeding Series*, pp. 99–104, 2019.

[39] M. Chaieb, S. Yousfi, P. Lafourcade, and R. Robbana, "Verify-your-vote: A verifiable blockchain-based online voting protocol," *Lecture Notes in Business Information Processing*, vol. 341, pp. 16–30, 2019.

[40] F. S. Hardwick, A. Gioulis, R. N. Akram, and K. Markantonakis, "E-Voting with Blockchain: An E-Voting Protocol with Decentralisation and Voter Privacy," *Proceedings - IEEE 2018 International Congress on Cybermatics: 2018 IEEE Conferences on Internet of Things, Green Computing and Communications, Cyber, Physical and Social Computing, Smart Data, Blockchain, Computer and Information Technology, iThings/Gree*, pp. 1561–1567, 2018.

[41] S. Bistarelli, M. Mantilacci, P. Santancini, and F. Santini, "An end-to-end voting-system based on bitcoin," *Proceedings of the Symposium on Applied Computing - SAC '17*, pp. 1836–1841, 2017. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3019612.3019841

[42] A. Singh and K. Chatterjee, "Secevs : Secure electronic voting system using blockchain technology," *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*, pp. 863–867, 2018.

[43] "Tutorial em vídeo what is apache http server, and what is it used for? - apache — linkedin learning, anteriormente lynda.com." [Online]. Available: https://www.linkedin.com/learning/apache-web-server-administration/what-is-apache-http-server-and-what-is-it-used-for