UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

FORMAL VERIFICATION OF BLOCKCHAIN BASED TENDER SYSTEMS

TESIS
QUE PARA OPTAR POR EL GRADO DE
MAESTRO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

PRESENTA:
RENÉ ADRIÁN DÁVILA PÉREZ

TUTOR:
DR. ISMAEL EVERARDO BÁRCENAS PATIÑO
FACULTAD DE INGENIERIA

CIUDAD DE MÉXICO, JUNIO, 2022

Quiero dedicar este trabajo a mis padres, hermanos, mis tutores, mis amigos, colegas, y especialmente a gatita, gracias por seguir siendo bellas referencias para mi en esta senda. . .

Declaration

I hereby declare that, except where specific reference is made to the work of others, the content of this dissertation is original and has not been submitted in its entirety for consideration for any other degree or qualification at this or any other university. Only some of the results presented in this document have already been published in “R. Dávila, R. Aldeco-Pérez and E. Bárcenas, "Tender System Verification with Satisfiability Modulo Theories", 2021 9th International Conference in Software Engineering Research and Innovation (CONISOFT), 2021, pp. 69-78, doi: 10.1109/CONISOFT52520.2021.00021” and published in “R. Dávila, R. Aldeco-Pérez and E. Bárcenas, "Formal Verification of Blockchain Based Tender Systems", Programming and Computer Software, Springer ISSN 0361-7688, 2022”.

René Dávila
June 2022

Acknowledgements

This research was supported by the Mexican Council CONACyT (1006953) in collaboration with Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas: Posgrado de Ciencia e Ingeniería de la Computación of the Universidad Nacional Autónoma de México. And was supported by UNAM-PAPIIT(TA101021) and UNAM-PAPIIT(IA105420).

Abstract

A tender process consists in competing offers from different candidate suppliers or contractors. The tender winner is supposed to supply or provide a service in better conditions than competitors. Tenders are developed using centralized unverified systems, which reduce transparency, fairness and trust on the process, it also reduces the ability to detect malicious attempts to manipulate the process. Systems that provide formal verification, decentralization, authentication, trust and transparency can mitigate these risks. Satisfiability Modulo Theories provides a formal analysis to prove correctness of tender offers properties, verified properties ensures system reliability. In addition one technology that claims to provide decentralization is Blockchain, a chain of distributed and decentralized records linked in a way such that integrity is ensured. This thesis document presents a formal verified and decentralized proposal system, based on Satisfiability Modulo Theories and Blockchain technology, to make electronic procurement tenders more reliable, transparent and fair.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Most governments do not directly supply goods and services to their citizens. Instead, they buy these goods and services from the private sector, which applies to become supplier through a tender. This process is called *government procurement* or *public procurement* which is the procurement of goods and services on behalf of a public authority[10].

Public tenders are sensitive to fraud and corruption, therefore, the laws of most countries regulate government procurement. One example is the European scheme for public tenders, which is one of the most organized and documented [2]. In this scheme, contracts typically go through competitive processes, following common and local legal guidelines of each member country of the European Union. The purpose of this scheme is to offer a fair process for the participants, with a fair price for the taxpayers of the country issuing the tender. Currently, this scheme handles various types of procedures for tendering, such as open or restricted. These procedures have in common a negotiation about what the participants will supply, but with different rules between each type of procedure.

1.1 Problem Statement

Although governments have robust legal rules for bidding procedures, these procedures are carried out centrally, where a collective or an individual entity reviews each bid based on the rules established by the corresponding tender. So later, the supplier with the proposal offering the best cost/quality ratio proposal is selected.

This centralization creates different risks for the tendering procedures. Centralized entities might give preferential treatment to some of the participants, thereby, undermining the fairness

of the process. There is also the possibility that bids are manipulated to favor a specific participant. In addition, the transparency of the procedures can also be compromised, as the results of the tendering process presented to the public are not reliable [13] as malicious manipulations are not published.

1.2 Research questions

Based on the above, it is necessary to ask the following questions:

- Is it possible to formalize and automate tendering procedures?
- What is considered relevant to verify and validate in a tender process?
- Is Blockchain a proper tool to guarantee transparency on tendering processes?

1.3 Hypothesis

It is possible to validate, automate, offer immutable transparency and ensure fairness in tendering procedures through the model and application of a system based on Satisfiability Modulo Theories and Permissioned Blockchain.

1.4 Objectives

The research raises the following objectives that range from the general to the particular.

1.4.1 General objectives

Model and implement a system based on Satisfiability Modulo Theories and Permissioned Blockchain to automate, validate, and strengthen government tendering procedures.

1.4.2 Specific objectives

- Verify the tendering rules established by public entities, and validate the bids of the tendering participants.

- Implement permissioned blockchain to record the events that occurred in the tender through transactions.
- Design smart contracts for consensus rules and for tender winner selection rules.
- Determine and implement a method with selection criteria for the tender winner.
- Design sequence diagrams for the implementation of the model.
- Implement the system model in programming tools.

1.5 Justification

Some governments have proposed initiatives for electronic tendering schemes. Some of these schemes are implemented using information systems that carry out bidding procedures through the Internet. One example of these kind of government is presented in [3], where a large-scale implementation was developed.

Despite the advantages offered by these systems, they are still centralized, therefore, managed by selected entities who have to comply with the applicable rules. Centralization might hide malicious manipulation.

In addition, it is also not possible to automatically verify if the tender rules are meet by the participants. By doing this, human errors and data manipulation can be reduced. Therefore, systems that provide automated verification, decentralization and transparency can mitigate these risks.

F. S. Hardwick and J. Deshpande present examples of systems that improve bidding procedures using Blockchain. Blockchain is a chain of distributed and decentralized records linked in a way that integrity is ensured. Once these records are saved they cannot be changed, making Blockchain an immutable and append-only record [9] [12]. Even though, these systems provide decentralization and transparency, they do not offer automatic verification of tender rules.

Satisfiability Modulo Theory (SMT) [5] is a verification technique to prove correctness of system's properties. Properties are expressed in a formal language and when all given properties are satisfied, it is said that the system is valid. This technique can be used to implement automatic verification of rules on a system.

Permissioned Blockchain [13] it is a type of restricted Blockchain, where access to participants is controlled by having full identification of them. These participants are impartial entities that attest to the records that are generated in the Blockchain.

In light of this, here is a proposal for a tendering system based on Satisfiability Modulo Theories and Permissioned Blockchain that supports biding processes.

By using this system, participants' bids will be automatically validated to later be registered in a blockchain, that through consensus of a number of peers supports decentralization. Therefore, reducing the reliance on a single entity. As consequence, reliability, fairness, integrity and transparency of a tendering process can be guaranteed.

1.6 Contributions

This research offers the following contributions:

- Present a system design for the public tendering procedure, as a reference for investigations of a similar nature.
- Show the operation of the system model with facilities for its optimization and improvement.
- Validate inputs to the system through the use of a formal verifier.
- Securely and robustly register the operations carried out in the tender process, in a permissioned Blockchain system.
- Offer a proof of concept to set a precedent that implementation is possible.

1.7 Outline

The presentation of this research work is organized as follows. Chapter 2 presents the related work to the research, as well as the theoretical concepts of e-Procurement or electronic provisioning, types of Blockchain, Smart Contracts as an important element to make records in the Blockchain, the RAFT consensus algorithm for establish the agreements in the Blockchain system, and some optimization method to determine the candidates with the best price/benefit criteria, which are the theoretical background of this research.
In Chapter 3 the formalization of the tender rules is presented, of which a logical model is presented and proved, which serves as the basis for automatic validation with a verification tool.
In Chapter 4 the design of the system is presented, highlighting the interaction between the verification blocks and the Blockchain. Also, the actors of the Blockchain are shown, along

with the actions they carry out to make the records of the system's operations, as well as a series of sequence diagrams that illustrate the steps carried out by the system, from the beginning of its execution until the presentation of the results.
Chapter 5 presents the most important parts of the system implementation, emphasizing the operation of the verification tool and the Blockchain system, closing with the presentation of the results produced by the execution of the system.
Finally, in Chapter 6, a discussion and conclusions of the research carried out are presented, including the future work for this system.

Chapter 2

Related Work & Background

This chapter describes some proposals for tender systems, mentioning their strengths, weaknesses and a comparison with the system proposed by this research work. In addition, the theoretical foundations that are the basis of this research are addressed.

2.1 Related Work

X-Road

Of all the countries implementing electronic procurement, we focus on Estonia as a pioneer in transforming public services into e-solutions. Its model X-Road (existing since 2001) is an e-Procurement system allowing government institutions to offer electronic tenders.

The Estonian government has been digitizing many of its services since the mid-90s. In 2001, an initiative to unify and protect the exchange of information between different government entities was proposed. The open source implementation of this initiative is known as X-Road [3] .

One of the goals of the X-Road initiative is that information systems do not directly exchange information between government entities. The information exchange is done through Standardized Security Servers, which are the entry points to X-Road. They are necessary to produce and to consume services on the X-Road, as well as, they mediate service calls and service responses between information systems. They encapsulate security aspects of the X-Road infrastructure such as: key management for signing and authentication, sending messages through a secure channel, creation of proof value for messages with digital signatures,

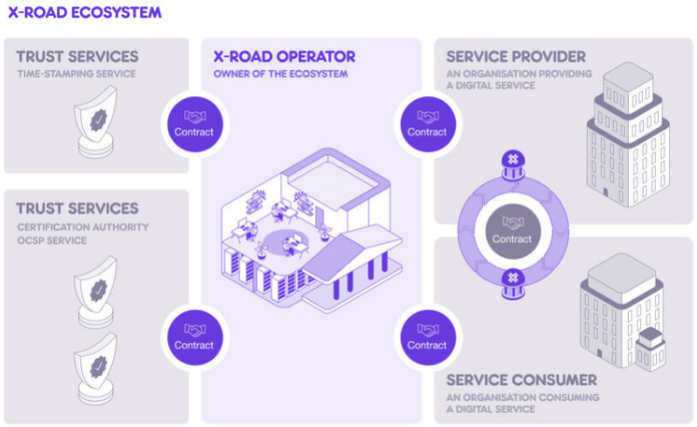

Fig. 2.1 X-Road Architecture. [3]

time-stamping and logging, so the information can be carried out in a more secure way [3]. X-Road implements public tendering procedures following the scheme presented in Fig. 2.1.

As presented in this figure, the X-Road Operators stand out from this architecture, since in addition to being the owners of it, they have the responsibility of all operational aspects, such as the definition of the rules and acceptance of members or participants, providing support to them. This is achieved through the use of contracts with all parties that are part of the procedures. Particularly in the tendering processes, the operators determine the tender rules and through the Standardized Security Servers they offer a secure scheme for the exchange of information between suppliers and the government entities.

Thus, this solution even though is a recent implementation, has the disadvantage of being centralized, i.e. all responsibilities fall on the owners of the system. Despite showing correct operation in the architecture and operating rules, there is still the possibility that the administrator or the owners of the system, might be corrupted and, as consequence, the fairness, integrity, and transparency of a tendering processes are compromised.

Blockchain-based solutions

As an improvement to centralized tendering systems, some Blockchain-based solutions has been proposed. These are discussed below.

F. S. Hardwick present a “Fair and Transparent Blockchain based Tendering Framework” [12].They propose an architecture based on public Blockchain and a framework for open tenders with transparency. The proposal makes use of Ethereum [8] technology together with Smart

Contracts. The smart contracts are used to validate bids and a series of algorithms perform the evaluation of the bids to determine the winner of the tender.

J. Deshpande et. al present a "Permissioned blockchain based public procurement system" [9]. They propose a permissioned Blockchain based system and a multiorganizational architecture. Using smart contracts permits are generated for recording and reading transactions. When the bidding period ends, the bids that satisfy the established requirements are sent to the government entity that issued the tender. This system presents elements in common with our proposal such as the use of a permissioned Blockchain and the definition of tender rules to be validated in a smart contract.

Table 2.1 below shows a series of drawbacks of these proposals.

Differences

As it is presented in table 2.1 both proposals define the necessary technical elements, such as algorithms or implementations in frameworks. However, neither of both present a formalization of their elements. Moreover, none of them propose a formal verification of rules on bids. As presented on Chapter 4, it is considered the use of Satisfiability Modulo Theories to perform such a formal verification on the bids of a tender offering a reliable validity to government entities, tender participants and citizens. Additionally, the difference between Blockchain schemes is shown, highlighting why one of the permissioned type is more suitable to carry out this process efficiently. Finally, it is shown that automation allows avoiding human interaction in important parts of the tender process.

2.2 e-Procurement

Government procurement is the procurement of goods or services on behalf of a public authority. To prevent fraud and corruption the law of many countries usually require the procuring authority to issue public tenders. These tenders describe the legal rules that participants must comply with, including the way in which bids are presented.

Under that context, this process is similar to a reverse auction [15], in which each bid is kept confidential and one clear winner is defined after the auction finishes. That is presented in Fig. 2.2.

Reverse auction is a type of auction in which the roles of the buyer and the seller are reversed, with the main goal to decrease purchase prices. Then, suppliers submit multiple offers, usually as a response to competing suppliers' offers, bidding down the price of a good or

Current Models or Systems	**SMT-Blockchain Proposal**
Validate bids using Smart Contracts.	Verify and validate bids using SMT.
They do not present formalization in none of their elements or procedures.	Formally defines Tender Rules and Bids.
They do not present automatic formal verification to tests their elements or procedures.	Uses an automated SMT verifier to prove bids' rules
F. S. Hardwick use a public Blockchain scheme. This type of Blockchain has longer processing time and higher energy consumption, factors that in the long term create higher expenses.	We propose a permissioned scheme, where only the participants are given access. There is less processing time and less energy consumption.
J. Deshpande system depends on the criteria of the issuing entity that launches the tender to validate the winning bid, then it is not an automated process, and human errors represent a risk for its correct operation.	Bids are automatically validated with a formal verifier, smart contracts automate the conditions of the winner and a consensus algorithm determines the bid with the best price / benefit criteria, in autonomous and decentralized processes.

Table 2.1 Related work comparison

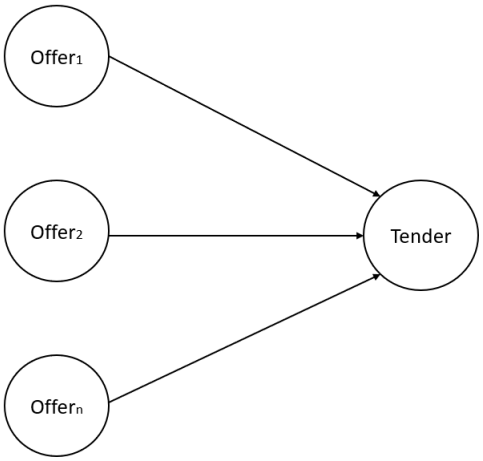

Fig. 2.2 Reverse Auction [15].

service to the lowest price they are willing to receive. By revealing the competing bids to every participating supplier, these type of auctions support "information transparency".

There are different procedure types of public tendering [2], of which the most common among the different governments of the countries are:

Open procedure. In this type of tender any potential supplier may submit a complete offer to the requesting government or government entity.

Restricted procedure. In this type of tender any supplier can request participation, to later go through a pre-selection process. The elected suppliers are the ones who launch offers.

Competitive procedure. In this type of tender any supplier can request participation, to later go through a pre-selection process. The elected suppliers are the ones invited to issue offers to begin a negotiation. This type of procedure is used by federal service entities.

Public e-Procurement (electronic procurement) refers to the use of electronic means to implement public procurement procedures.

2.3 Satisfiability Modulo Theories

The *Satisfiability Modulo Theories problem* (SMT) is a decision problem for logical formulas with respect to combinations of background theories expressed in classical First Order Logic with equality [5].

A *decision problem* is a problem which can be abstracted as a yes or no question of the input values while a *formal theory* is a set of sentences that can be used to restrict the models we wish to consider.

SMT are expressed in *First Order Logic* (Formally and in detail defined in Section 3.1, Definitions 5, 6) which is defined over an alphabet $(X, \mathscr{P}, F)$, which is a set of variables, predicate symbols such as $(0, 1, ..., +, -, ..., <, >)$ and functions.

The equality symbol $=$ is assumed to be included in every alphabet. Variables and function symbols in the alphabet can be used to build theory-terms (t_i-terms). A t-term is either a variable or, recursively, an application of a function symbol in alphabet to terms.

The logical symbols $(\rightarrow, \wedge, \vee)$ in an alphabet can be used to build theory-atoms (T-atom). A T-atom is the application of a predicate symbol in the alphabet to T-terms.

A theory-literal (T-literal) is either a T-atom or its negation. A formula is any boolean combination of T-atoms and boolean variables.

An approach to solve SMT formulae is based on the observation that an SMT can be reduced to a Propositional Satisfiability Problem (SAT) formula. Reductions can be solved atomically, to finally combine the results, in order to prove if the input formula is valid. This approach will be useful to validate the inputs during the operation of the proposed system model.

2.4 Blockchain

At the end of 2008 [16], with the invention of cryptocurrencies, the idea of a decentralized database emerged. This idea is known as the Blockchain scheme.

Quoting a definition from the NIST IR 8202 standard, "Blockchains are distributed ledgers of cryptographically signed transactions that are grouped into blocks. Each block is cryptographically linked to the previous one (making it tamper evident) after validation and undergoing a consensus decision. As new blocks are added, older blocks become more difficult to modify (creating tamper resistance and strength integrity). New blocks are replicated across copies of the ledger within the network, and any conflicts are solved automatically using established rules [18]" The Figure 2.3 shows the Blockchain structure.

As we can see in that figure, a block is made up by the following elements [13]:

Data. It is the information stored in the Blockchain, this data depends on the main service or application using Blockchain.

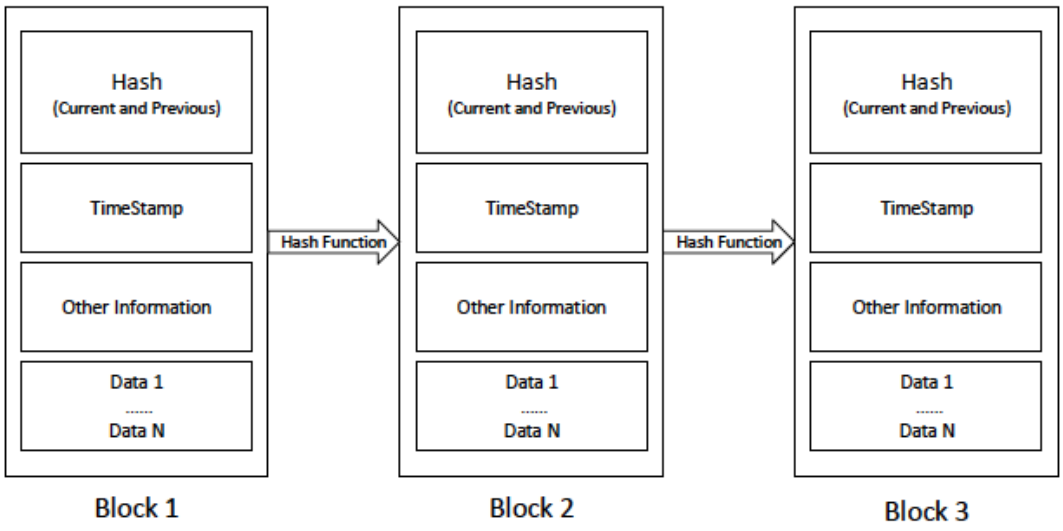

Fig. 2.3 Blockchain Structure [13].

Hash. A cryptographic hash function that takes an input of any length and generates an output with unique fixed length. If a value in the input is modified, then the output will be different. This element supports the integrity of the data and the relationship with other pieces of data.

Timestamp. This is a record of the creation time of a block. This element can be used to track the creation or modification time of a piece of data in a secure way, i.e. with integrity and non-repudiation.

Other Information. This part contains digital signatures, nonce values, public and private keys, and any other necessary cryptographic elements that are used to support different security properties.

In terms of operation, Blockchain works as follows [13]:

1. A node (or user) who wants to initiate a transaction will record and broadcast the data from such a transaction to the network. The network is formed by more nodes.
2. A node, who is part of the network, receives data from another node to verify its authenticity. After such piece of data is verified, then it is stored in a block.
3. All nodes in the network participate on the verification of transactions by executing a consensus algorithm on the block that needs to be verified.

4. A consensus algorithm is used by the nodes on the network to decide which blocks are added to the Blockchain and extend the chain base on the block.

The continuously generated blocks are linked and secured by cryptographic hash values, created by a cryptographic hash algorithm such as the SHA-256, SHA-2 or SHA-2. Each block contains a hash value of the previous block, if there are changes in any of the blocks then the hash value will change, this ensures the property of integrity and inmutability, and that change will be visible to all pending Blockchain participants ensuring transparency.

There are two types of Blockchain: *Permissionless* and *Permissioned*. The former, called Public or Permissionless, it is open to all participants in the Blockchain, where participants preserve anonymity and have full ledger transparency. Everyone in the network can validate transactions and can partake in the process of consensus. However, they have a high energy consumption and use a consensus algorithm for which the outcome is not always the best option in the consensus process [13].

The latter, called Private or Permissioned, contrary to the permissionless type, it is not open to all participants. The participation of nodes is manage by third parties, usually impartial entities, i.e. they do not belong to the same organization and do not share interests. In this type of Blockchain, none of the nodes in the network can participate in the verification and validation of the transactions. Instead a selected group of nodes perform the processes of initiating, verifying and validating transactions, improving efficiency of such processes. At difference of public blockchains, private blockchains do not provide decentralized security due to restricted access [13]. However, since in a private blockchain a third party assigns the access rights to each participant, the privacy levels is increased making this type of blockchain suitable for government sectors. Another benefit of this type is its low energy consumption consequence of the used consensus algorithms [13].

Both previously described types carry out a consensus process. A consensus process is one in which, the participants of the network agree on a block be valid. Specifically, if the majority of nodes approve a block, the corresponding transaction is registered. This process is what provides decentralization. In the case of public blockchains, the consensus is defined by the validations of the participants in the network, and in the case of private blockchains, the consensus is defined by the selected entities accepted in the network [13].

2.5 Smart contracts

Blockchain networks also allow the creation of *Smart Contracts*. Referring to the definition provided by the NIST: "A Smart Contract is a set of code and data (sometimes referred to as functions and state) that is deployed using cryptographically signed transactions on the Blockchain Network. The Smart Contract is executed by nodes within the blockchain network; all nodes that execute the smart contract must derive the same results from the execution, and the results of execution are recorded on the blockchain [18]".

2.6 RAFT Consensus Algorithm

Voting-based consensus protocols use voting processes to designate the node that will be in charge of handling communication between the rest of the nodes or accepting a block as valid. These are the preferred type of consensus protocols on private blockchains. Voting-based protocols are Crash Fault Tolerance (CFT), i.e. they provide protection against the failing of the network as consequence of certain nodes failing.

RAFT [14] is a CFT type algorithm, as it remains effective if more than 50% of the nodes in the network are working normally.

In this algorithm nodes play the role of leaders (orderers), followers or candidates to perform consensus on the transaction record. Each role performs the following actions.

- **Leader (Orderer).** There is only one leader in the network that is responsible for generating entries in the log of transactions received by peers, any change in the network go through it first.
- **Follower.** Nodes with passive behavior which function is to answer the requests of the leader and candidate nodes.
- **Candidate.** Nodes that do not find a leader node run as candidates and request be elected as leaders.

RAFT protocol works as follow[14]:

1. At the beginning, the nodes are in the follower state, waiting for some type of communication from the unique leader node.

2. All nodes have a random waiting time. When this time ends, if there was no communication from a leader, they change their state to candidate and ask the other nodes for a vote to be elected as leader.

3. RAFT divides time into periods. A node in candidate state becomes a leader if the majority votes for it within a given period of time.

4. The minimum number of nodes available for consensus (or Quorum) is defined as $\frac{N}{2} + 1$ where N is the number of nodes.

The proposed system model makes use of a private or permissioned Blockchain, where RAFT is used to make consensus efficiently, as RAFT is a protocol based on voting. Then it is not necessary to present some type of proof and consequently there is no high resources consumption.

2.7 Simplex Method

It is a linear programming technique for solving optimization problems, where a function and numerical values are expressed as inequalities. The inequalities define a polygonal region, where the solution is usually one of the vertices. Specifically, the simplex method is a systematic procedure to test the vertices of possible solutions [17].

When considering problems that have several decision variables, the mathematical calculations become more complex to obtain an optimal feasible solution. The optimal solution of a certain problem is found at one of the extreme points of the polygon of feasible solutions, this means that each vertex of the polygon is a basic feasible solution [17]. Using the simplex method one can go through all the vertices of the polygon of feasible solutions, until you find the vertex that contains the optimal solution of the problem, for reference see figure 2.4.

The path that is made between the vertices is called iteration, from that movement the variables and the function are transformed and new values are obtained for the next vertex, these new values become a basic feasible solution.

In order for the traversal to be possible, the mathematical model of the problem needs to have variations, since the model is made up of inequalities instead of equations, which are difficult to deal with in algebra. The Simplex Method uses some changes that are made in the inequalities, such as changing inequalities into equalities when they are less than or equal to, for

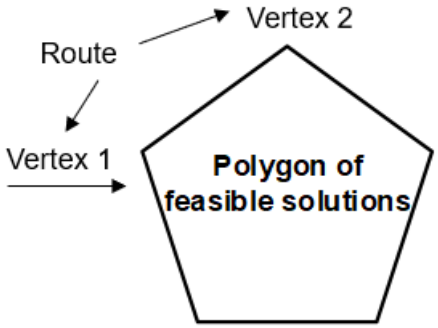

Fig. 2.4 Simplex Method. [17]

this a variable called slack or filler is added. If the inequality is greater than equal, it is changed by adding a slack variable in negative form and an artificial variable in positive form.

This optimization method is considered viable to determine the best offer received in the tender process for the proposed system.

2.8 Hyperledger Fabric

Since a permissioned blockchain is proposed to be used, then it is considered to make use of the open project Hyperledger Fabric, based on its documentation the following is highlighted:

Hyperledger Fabric is a modular and configurable architecture, it enables innovation, optimization and is highly versatile for a range of permissioned industry cases, such as banking, finance, insurance, supply chain, or in this case, tendering.

Fabric supports general-purpose programming languages such as JavaScript, Go, Java, then is not necessary to learn a new platform or technology.

In Fabric the access is permissioned, the participants are known to each other, it is important for a process like tendering, because governments has to know the identity of the participants.

One of the most important features is the use of the Raft consensus algorithm (explained in point 2.6 of this chapter), it works as a CFT algorithm to provide protection against network failures by certain nodes.

Fabric does not require high resource consumption for mining, and also does not suffer from the risk of attack vectors from cryptocurrencies [1].

Chapter 3

Formal Model Analysis

This chapter presents the formalization of the offers of the participants, in said formalization a series of definitions is presented, the definitions result in the approach of logical formulas. The logical formulas are proved by induction to show that the formalization of the offers was done correctly. First, the formal part is built through logic and SMT of a tender process, then the result of the formalization is demonstrated by induction, finally the flow of operation of the participants in the tender and the verification process is presented, within a model of system proposed in chapter 4.

3.1 Tender rules & offers formalization

This section presents the results of the formalization of the tender rules and the offers of the participants, that occur in the verifier process (Fig.4.1 in Subchapter 4.1).

Following tender rules specified in [2], we have identified four types of general rules in a tender process:

- specifications associated to particular tender entities (several tender entities may form part of the tender), such as antitrust regulations or import or export taxes;
- specifications associated to bidders, such as your legal identification or certifications;
- general specifications, such as a tender registration; and
- numerical constraints, such as the price limit of the tender or budget of some offer proposal.

To explain how these rules are used in the tender, the following set of examples and definitions are presented.

Example 1. The communications department calls a tender to provide 5G cell phone service to citizens, under the following defined rules:

- The communications department determines the type of contract (where the costs are established), the date of the call, and the sector to which the participants must belong. In addition, the tax department defines the fiscal guidelines to be met as tax percentages, while the economy department establishes antitrust regulations.
- Bidders must present the following requirements: Legal identification, financial identification and commercial agreements.
- As general specifications, it is established that the communications department must have a legal registration of the tender, further, bidders are requested that they must have technical certifications and quality control certifications.
- Regarding numerical constraints, the communications department establishes a limit price of the tender, the tax department establishes the percentages of the most relevant taxes in terms of services, finally the economy department establishes that bidders must not exceed a limit percentage of control of the market so that their participation is allowed. On the other hand, the bidders must present their budget with all the established costs.

In order to formalize the tender rules (like the rules of Example 1), a hybrid specification based on a rule-based expert system which are non-numerical specifications [11] and a numerical constraint system [7] are proposed. The rule-based expert system formalizes the knowledge required to express the type of rules not involving numerical constraints, that is, specifications associated to particular tender entities and bidders, and general specifications. Numerical constraints are formalized by the corresponding system.

Definition 1 (Non-numerical specifications)**.** Non-numerical specifications are expressed by a set of rules of the following form:

$$\text{IF } antecedent \text{ THEN } consequent$$

where *antecedent* and *consequent* may represent the boolean combination of statements.

Example 2. To explain Definition 1 we express some rules from Example 1 in non-numerical specifications, where the following cases may happen:
- Case 1 (Strict Tender)
Bidder must satisfy all the rules established:
Sub-case 1.1:
IF $(Bidder$
AND NOT $Legal\_ID)$
THEN NOT $Valid\_offer$
Sub-case 1.2:
IF $(Bidder$
AND $Legal\_ID)$
THEN $(Financial\_ID$
AND $Commercial\_agreements)$
Sub-case 1.3:
IF $(Bidder$
AND $Legal\_ID$
AND $Financial\_ID$
AND $Commercial\_agreements)$
THEN $(Quality\_certifications$
AND $Technical\_certifications)$
Sub-case 1.4
IF $(Other\_Bidder$
AND $Partner(Bidder, Other\_Bidder))$
THEN NOT $Valid\_offer$
Sub-case 1.5
IF $(Government\_Entity$
AND $Partner(Bidder, Government\_Entity))$
THEN NOT $Valid\_offer$
Sub-case 1.6
IF $(Bidder$
AND $Legal\_ID$
AND $Quality\_certifications$
AND NOT EXISTS $(Other\_Bidder$
AND $Partner(Bidder, Other\_Bidder))$

THEN $Valid\_offer$

- Case 2 (Flexible Tender) Bidder must satisfy mandatory rules and some other rules have to be met with at least one:
Sub-case 2.1:
IF ($Bidder$
AND $Legal\_ID$
AND $Financial\_ID$
OR $Commercial\_agreements$)
THEN ($Technical\_certifications$
OR $Quality\_certifications$)
Sub-case 2.2:
IF ($Bidder$
AND NOT $Commercial\_agreements$
AND NOT $Quality\_certifications$)
THEN ($Legal\_ID$
AND $Technical\_certifications$)
Sub-case 2.3:
IF ($Bidder$
AND NOT $Commercial\_agreements$
AND NOT $Financial\_ID$)
THEN ($Legal\_ID$
AND $Technical\_certifications$
OR $Quality\_certifications$)
Sub-case 2.4
IF ($Other\_Bidder$
AND $Partner(Bidder, Other\_Bidder)$)
THEN NOT $Valid\_offer$
Sub-case 2.5
IF ($Government\_Entity$
AND $Partner(Bidder, Government\_Entity)$)
THEN NOT $Valid\_offer$
Sub-case 2.6
IF ($Bidder$

AND $Legal\_ID$)
AND ($Quality\_certifications$
OR $Technical\_certifications$)
AND NOT EXISTS ($Government\_Entity$
AND $Partner(Bidder, Government\_Entity)$)
THEN $Valid\_offer$

Definition 2 (Numerical specifications)**.** Numerical constraints are expressed by an equation system:

$$\begin{aligned} a_{1,1}x_1 + a_{1,2}x_2 + \ldots a_{1,n}x_n &= t_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \ldots a_{2,n}x_n &= t_2 \\ &\vdots \\ a_{m,1}x_1 + a_{m,2}x_2 + \ldots a_{m,n}x_n &= t_m \end{aligned}$$

for any positive integers n and m. Notice other relations, such as $>$, $<$, $\leq$, $\geq$, may also be expressed, for instance $x \leq k$ holds if and only if $x + y = k$ for some positive integer y.

Example 3. To explain Definition 2 we express some rules from Example 1 in numerical specifications, where an offer contains the following numerical constraints:
Budget specifications.
The offer budget considers the following variables:

- **Top price:** It cannot exceed the value of the variable t_1.
- x_1 represents the total price of all materials to be used for the infrastructure.
- x_2 represents the total price of the physical spaces (land or sites) where the infrastructure will be installed.
- x_3 represents the total price of professional services for the installation of devices and software.
- x_4 represents the total price of operating expenses to perform the installation.
- Then the numerical expression of the budget is represented by the equation $x_1 + x_2 + x_3 + x_4 \leq t_1$.

The prices of each variable are different between the offers.

Taxes specifications.

The tax department has the following taxes established:

- **16%** of taxes in the acquisition of materials.
- **30%** of taxes in the acquisition or rent of real estate.
- **25%** of taxes for the provision of professional services.
- **Total taxes:** represented by the variable t_2, this amount will be defined by the sum of taxes on materials, real estate and professional services.
- Then the numerical expression of the taxes is represented by the equation $0.16x_1 + 0.30x_2 + 0.25x_3 + 0.16x_4 = t_2$.

Market share specifications.

The economy department establishes that the bidders in the tender must not exceed **40%** of the market. Then the numerical expression of the market share is represented by the equation:

- $0.2y \leq 0.4x$

where x is the variable that represents the total market share and y represents the market share of the present offer.

Therefore, the numerical constraints of Example 1 are expressed by the following equation system:

$$\begin{aligned} x_1 + x_2 + x_3 + x_4 + k_1 &= t_1 \\ 0.16x_1 + 0.30x_2 + 0.25x_3 + 0.16x_4 &= t_2 \\ y - 0.4x + k_2 &= 0 \end{aligned}$$

where k_1 and k_2 represents a complementary value to fulfill the reason $x \leq t$ if and only if $x + k = t$.

A bidder is defined below.

Definition 3 (Bidder Offer)**.** A bidder offer is defined by the tuple

$$(Statements, NumericalEqualities)$$

where Statements is a set of fulfilled properties, defined by the tender rules, and NumericalEqualities is a set of equalities between variables and positive real numbers, associated to costs.

Example 4. To explain Definition 3, consider the following bidder b with rules based on Examples 2 and 3:

- The $Statements$, according to sub-case 1.6 of the Example 2, are

$$LegalID(b), QualityCertifications(b)$$

- The $NumericalEqualities$, according to Example 3 are $x_1^b = 10,000$, $x_2^b = 50,000$, $x_3^b = 30,000$, $x_4^b = 10,000$, $y^b = 0.2$, where x_i^b are budget costs and y^b is the market share percentage of b.

We are now ready to define when a bidder satisfies the tender rules.

Definition 4 (Bidder offer fulfillment)**.** Given a set of tender rules, expressed in terms of a rule-based expert system (Definition 1) and a numerical constraints system (Definition 2), we say a bidder offer fulfills the rules, if and only if, the statements and numerical equalities (Definition 3) fulfill all numerical and non-numerical specifications.

Example 5. To explain Definition 4, consider the tender rules of Sub-case 1.6 of Example 2 and the numerical rules defined by the equation system of Example 3 with the following defined values by the communications department

$$\begin{aligned} x_1 + x_2 + x_3 + x_4 &\leq 120,000 \\ 0.16x_1 + 0.30x_2 + 0.25x_3 + 0.16x_4 &= t_2 \\ y &\leq 0.4x \end{aligned}$$

the following cases may happen:

- Case 1

A bidder b_1 bidder presents its offer with the following statements

$$LegalID(b_1), QualityCertifications(b_1)$$

And with the following numerical equalities $x_1^{b_1} = 10,000$, $x_2^{b_1} = 50,000$, $x_3^{b_1} = 30,000$, $x_4^{b_1} = 10,000$, $y^{b_1} = 0.2$. Then, we conclude that b_1 complies with the tender rules because its statements satisfy the non-numerical rules and the sum of the budget elements together with its percentage of marketshare does not exceed the established limits. Therefore, b_1 complies with the tender rules.

- Case 2

A bidder b_2 bidder presents its offer with the following statements $LegalID(b_2)$. And with the following numerical equalities $x_1^{b_2} = 10,000$, $x_2^{b_2} = 60,000$, $x_3^{b_2} = 50,000$, $x_4^{b_2} = 10,000$, $y^{b_2} = 0.45$. In this case, it can be concluded that b_2 do not fulfill the tender rules because its statements do not satisfy the non-numerical rules and the sum of the budget elements together with its percentage of marketshare exceed the established limits. Therefore, b_2 do not complies with the tender rules.

We now describe a brief reminder of syntax and semantics of First Order Logic (FOL). Then it is necessary to use FOL to model and verify bidder satisfiability. We assume a fixed alphabet of variables X, function and predicate symbols, F and $\mathscr{P}$.

Definition 5 (FOL syntax)**.** FOL formulas are defined by the following grammar:

$$\phi := P(t_1, t_2, \ldots, t_n) \mid \neg\phi \mid \phi \vee \phi \mid \exists x\phi$$

where terms t_i $(i = 1, \ldots, n)$ are defined as

$$t := x \mid c \mid f(t_1, t_2, \ldots, t_m)$$

such that $x \in X$, $c, f \in F$, $P \in \mathscr{P}$ and t_j $(j = 1, \ldots, m)$ are also terms.

We now describe the notion of first order structure as a tuple $S = (D, \mathscr{P}, F)$, where: D is a non-empty set called domain, and $\mathscr{P}$ is a set of relations and F a set of functions among the domain. We write P^S and f^S when $P^S \in \mathscr{P}$ and $f^S \in F$. We also need the notion of valuation $V : X \mapsto D$.

Definition 6 (FOL semantics)**.** Given a first order structure S and a valuation V, FOL formulas are interpreted as follows:

$$\begin{aligned}
[\![P(t_1,t_2,\ldots,t_n)]\!]^S_V &= 1, \text{ iff}, P^S\left([\![t_1]\!]^S_V, [\![t_2]\!]^S_V, \ldots, [\![t_n]\!]^S_V\right)\\
[\![\neg\phi]\!]^S_V &= 1, \text{ iff}, [\![\neg\phi]\!]^S_V \neq 1\\
[\![\phi \vee \psi]\!]^S_V &= 1, \text{ iff}, [\![\neg\phi]\!]^S_V = 1 \text{ or } [\![\psi]\!]^S_V = 1\\
[\![\exists x\phi]\!]^S_V &= 1, \text{ iff, there is } d \in D \text{ s.t.}\\
&\quad [\![\phi]\!]^S_{V[^x/_d]} = 1
\end{aligned}$$

where

$$\begin{aligned}
[\![x]\!]^S_V &= V(x),\\
[\![f(t_1,t_2,\ldots,t_m)]\!]^S_V &= f^S\left([\![t_1]\!]^S_V, [\![t_2]\!]^S_V, \ldots, [\![t_m]\!]^S_V\right),
\end{aligned}$$

and $V[^x/_d]$ stands for a valuation V', such that $V'(y) = V(y)$ for all $y \neq x$ and $V'(x) = d$.

We also consider the following notation: $\phi \wedge \psi := \neg(\neg\phi \vee \neg\psi)$, $\phi \rightarrow \psi := \neg\phi \vee \psi$, $\phi \leftrightarrow \psi := (\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$, $\forall x\phi := \neg\exists x\neg\phi$, $\top := \phi \vee \neg\phi$, $\bot := \neg\top$.

Definition 7 (Satisfiability)**.** We say a formula ϕ is satisfiable, if and only if, there is a first order structure S, such that $[\![\phi]\!]^S_V = 1$ for any valuation V. A formula ϕ is satisfiable under a theory $\{\phi_1, \phi_2, \ldots, \phi_n\}$, if and only if, $(\bigwedge_{i=1}^n \phi_i) \rightarrow \phi$ is satisfiable. When clear from context, we say ϕ is satisfiable only. We consider a particular the theory of arithmetic [6]. Under this theory and when obvious from context we use the symbol $=$ for the equality relation, and functions $+, -, *$ in infix notation.

In order to verify the notion of bidder satisfiability, we model the bid rules using FOL.

Definition 8 (Tender Rules Formalization)**.** Given a set of tender rules, expressed in terms of a rule-based expert system (Definition 1) and a numerical constraints system (Definition 2), and a bidder b, we define the FOL formula $TR(b)$ (b occurs in TR) as follows:

$$TR(b) := ES(b) \wedge NS$$

where n rules of the expert system are defined by the formula

$$ES(b) := \bigwedge_{i=1}^{n} (Antecedent(b)_i \rightarrow Consequent(b)_i)$$

and m numerical constraints are defined by the formula

$$NS := \bigwedge_{j=1}^{m} \sum_{k=1}^{l} a_{j,k} x_l = c_j$$

where c is a value given by the bidder b. Other relations, such as $>$, $<$, $\leq$, $\geq$, may also be expressed.

Example 6. To explain Definitions 5, 6, 7 & 8, we present an example of formal abstraction, based on Examples 2 and 3 where the communications department establishes the tender rules specified in Sub-case 1.6 of Example 2 for the bidders, the rules are represented by the logical formula: $TR(b) := ES \wedge NS$, where ES has the following non-numerical specifications:

- Bidder must be registered in the tender $Bidder(b)$
- Bidder must have a Legal identification $LegalID(b)$.
- Bidder must have Quality certifications $QualityCertifications(b)$
- Other bidders can participate $Bidder(x)$.
- Other government entities can participate $GE(g)$.
- The partner relationship is represented as $Partner(b,x)$
- Partner relationship between bidders is not accepted $\neg\exists x(Bidder(x) \wedge Partner(b,x))$.
- Partner relationship between bidder and government entity is not accepted $\neg\exists g(GE(g) \wedge Partner(b,g))$.

The abstraction of Sub-case 1.6 for non-numerical specifications ES are established as follows:

$$\begin{aligned}
&ES := \text{IF } \top \\
&\text{THEN } (Bidder(b) \\
&\wedge LegalID(b) \wedge QualityCertifications(b) \\
&\wedge \neg\exists x(Bidder(x) \wedge Partner(b,x)) \\
&\wedge \neg\exists g(GE(g) \wedge Partner(b,g)))
\end{aligned}$$

The numerical specifications specified in Example 3 NS are established as follows:

$$\begin{aligned}
&NS := (x_1^b + x_2^b + x_3^b + x_4^b \leq 120,000 \\
&\wedge 0.16x_1^b + 0.30x_2^b + 0.25x_3^b + 0.16x_4^b = t_2^b \\
&\wedge y^b \leq 0.4x)
\end{aligned}$$

where:

- Top price of budget is $120,000$ USD.
- t_2^b is the total amount of taxes.
- Top percentage of market share is 40%.

Definition 9 (Bidder Offer Formalization)**.** Given a bidder b, expressed in terms of statements and numerical equalities (Definition 3), and his offer, we define the FOL formula $BO(b)$ (b occurs in BO) as follows:

$$BO(b) := ST \wedge NE$$

where n statements of the offer are defined by the formula

$$ST(b) := \bigwedge_{i=1}^{n} (Statements(b)_i)$$

and m numerical equalities of the offer are defined by the formula

$$NE := \bigwedge_{j=1}^{m} \left(a_j x_j = c_j\right)$$

where c is a value given by the bidder b.

Example 7. To explain Definition 9 we make use of the abstraction of the cases of Example 5 expressed as follows: - A registered bidder b_1 participates with the following proposal entries for the ST

$$Bidder(b_1) \wedge LegalID(b_1) \wedge QualityCertifications(b_1)$$

and the b_1 entries for the NE are

$$x_1^{b_1} = 10,000 \wedge x_2^{b_1} = 50,000 \wedge x_3^{b_1} = 30,000$$
$$\wedge x_4^{b_1} = 10,000 \wedge y^{b_1} = 0.2$$

- A registered bidder b_2 participates with the following proposal entries for the ST

$$Bidder(b_2) \wedge LegalID(b_2)$$

and the b_2 entries for the NE are

$$x_1^{b_2} = 10,000 \wedge x_2^{b_2} = 60,000 \wedge x_3^{b_2} = 50,000$$
$$\wedge x_4^{b_2} = 10,000 \wedge y^{b_2} = 0.45$$

Once we have built logical formulas from the definitions, let us now see how the formal verification process is carried out by proving the formulas.

3.2 Bidder offer verification proof

Based on the definitions and examples from the section 3.1 of section 3.1, below we propose a theorem to validate the bids of the tender participants.

Theorem 1 (Bidder offer verification)**.** *Given a set of tender rules and a bidder offer b, the FOL formula $TR(b) \wedge BO(b)$ is satisfiable if and only if the bidder offer fulfills the tender rules.*

Proof. $[\![TR(b) \wedge BO(b)]\!]_V^S = 1 \implies b$ fulfills tender rules
Induction over the size of $TR(b) \wedge BO(b)$

Base case:
There is only one rule for $BO(b)$ then there is only one $TR(b)$ rule to be satisfied,
$ES(b) \wedge NS \wedge ST(b) \wedge NE$
where
$ES := (Antecedent(b)_1 \rightarrow Consequent(b)_1)$
$NS := a_{1,1}x_1 = c_1$
$ST := Statement(b)_1$
$NE := a_1x_1 = c_1$
Assume $BO(b)$ rule satisfies $TR(b)$ rule
Therefore $(TR(b) \wedge BO(b)) = 1$ and by Definition 4 in Section 3.1 then b fulfills the tender rules.

Induction hypothesis: if there are n rules for $BO(b)$ then there are n $TR(b)$ rules to be satisfied.
Inductive step: proof for $n+1$ rules for $BO(b)$ over $n+1$ $TR(b)$ rules.
Case 1:
There is one $ES(b)$ rule and $n+1$ NS rules
where
$ES := (Antecedent(b)_1 \rightarrow Consequent(b)_1)$
$NS := \bigwedge_{i=1}^{n+1} \sum_{j=1}^{m+1} a_{i,j}x_m$
$ST := Statement(b)_1$
$NE := \bigwedge_{i=1}^{n+1} a_ix_i = c_i$
Assume $BO(b)$ rules satisfies $TR(b)$ rules
Therefore $(TR(b) \wedge BO(b)) = 1$ and by Definition 4 in Section 3.1 then b fulfills the tender rules.
Case 2:
There is $n+1$ $ES(b)$ rules and one NS rule
where
$ES := \bigwedge_{i=1}^{n+1} (Antecedent(b)_i \rightarrow Consequent(b)_i)$
$NS := a_{1,1}x_1 = c_1$
$ST := \bigwedge_{i=1}^{n+1} Statements(b)_i$
$NE := a_1x_1 = c_1$
Assume $BO(b)$ rules satisfies $TR(b)$ rules

Therefore $(TR(b) \wedge BO(b)) = 1$ and by Definition 4 in Section 3.1 then b fulfills the tender rules.
Case 3:
There is $n+1$ $ES(b)$ rules and $n+1$ NS rules
where

$$ES := \bigwedge_{i=1}^{n+1} (Antecedents(b)_i \rightarrow Consequents(b)_i)$$

$$NS := \bigwedge_{i=1}^{n+1} \sum_{j=1}^{m+1} a_{i,j} x_m$$

$$ST := \bigwedge_{i=1}^{n+1} Statements(b)_i$$

$$NE := \bigwedge_{i=1}^{n+1} a_i x_i = c_i$$

Assume $BO(b)$ rules satisfies $TR(b)$ rules
Therefore $(TR(b) \wedge BO(b)) = 1$ and by Definition 4 in Section 3.1 then b fulfills the tender rules.

The other implication direction is proved in an analogous manner. □

The demonstration presented gives us the certainty that the rules of a tender process could be formalized correctly, this increases confidence for the participants in the tender, for the governments and for the citizens.

3.3 Verification functionality

In this section we present the operation of the formality presented in the previous sections of this chapter within the system presented in chapter 4, the functionality is described in detail below:

System operation presented in chapter 4 begins with Tender rules & offers verification, Fig.3.1 shows the sequence of the entries of the tender rules and the offers of the participants. In the case of $TR(b)$, the logical abstraction of the rules is sent to the verifier, at the same time, the information on the rules registered in the verifier is sent to the blockchain system for the block record of that transaction. Also, in the case of $BO(b)$, the logical abstraction of the formula is sent to the verifier. The verifier automatically performs the proof of the FOL formula

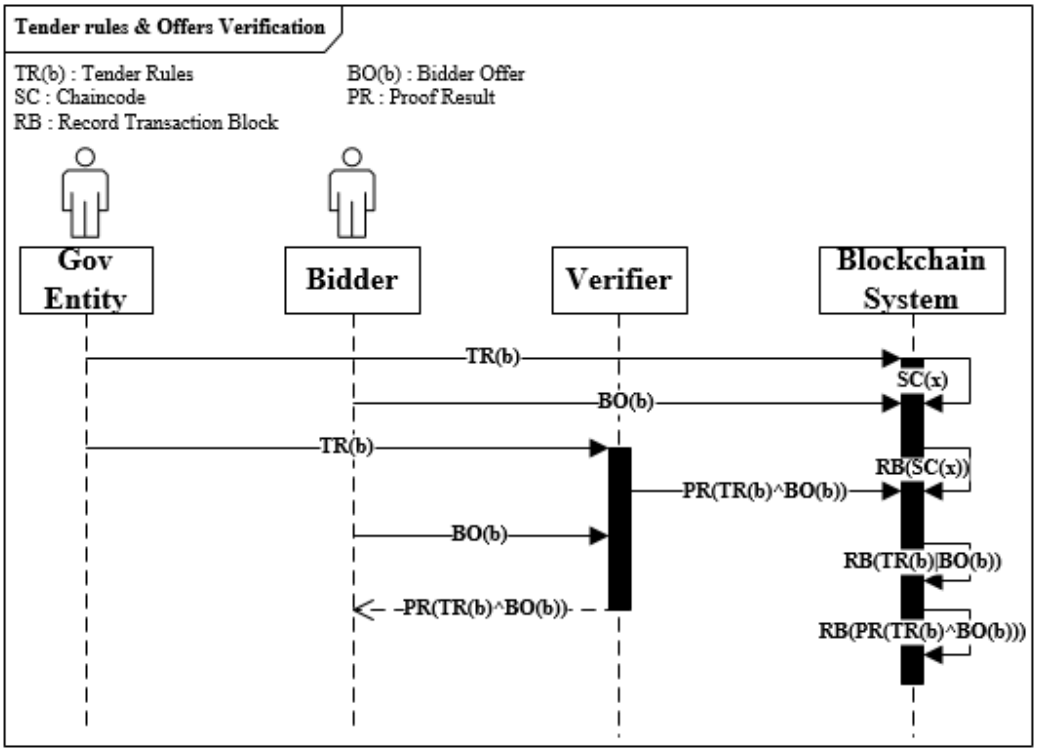

Fig. 3.1 Tender rules & Offers Verification

$TR(b) \wedge BO(b)$ of each b, and sends the result of the proof to the Formal blockchain-based system (see chapter 4 Fig.4.1) to record the transaction in the Blockchain.

It can be seen that the formalization process, in addition to strengthening the tender procedure, it is also possible to model it towards functionality within a system, in chapter 4 we will see the operation of the entire proposed system, and in chapter 5 we will see how the implementation of the verification process will be carried out.

Chapter 4

Model Design

This chapter presents the high-level design of the system, the actors and functionality. This functionality is later described through a set of sequence diagrams, as well as, the description of the operation of the system's Blockchain network.

4.1 System overview

The Fig.4.1 illustrates the architecture of the model. In this, there are 5 blocks that groups the main actors of the system. These blocks are defined as follows.

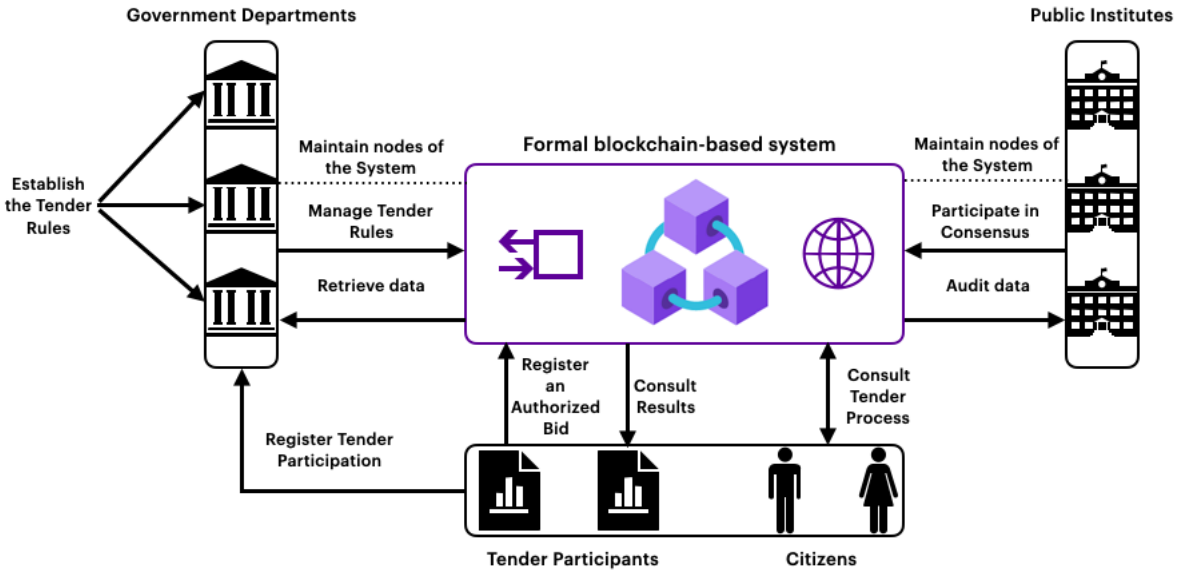

Fig. 4.1 System Model.

Formal blockchain-based system. This is the main block of the model that is depicted in purple on Fig.4.1 . Here the tender rules are established, registration of operations in the Blockchain, determination of the winning offer are carried out in an automated way, and it offers access to the information registered in the Blockchain to participants and citizens interested in reviewing the procedures carried out within the tender system.
Government Departments. This block represents the public government institutions that issue the calls for bids, establish the tender rules, control access to participants and are constantly managing the operations that occur in the Blockchain.
Public Institutes. This block represents the public institutions that participate in the bidding process as members of the consensus for registering transactions in the Blockchain. They audit the information that is recorded in the system, and also handle the operations that occur within the Blockchain. The participation of these institutions is considered impartial, to strengthen the fairness of the tender process.
Tender participants. It represents the companies or organizations interested in participating in the tender process. They are obliged to register their participation so that they have control over their access. Once registered, they can send their offers to the system, they can consult the results of the valid rules that they comply with, or consult the transactions with information on the procedures that were carried out in the tender process in a transparent manner.
Citizens. This part of the block represents citizens interested in reviewing a tender process, in order to check the procedure was fair and that the use of their taxes will be made according to the legislation.

Once the blocks that represent the actors in the model have been described, the proposed functionality of the system is presented below.

4.2 System functionality

In this section, sequence diagrams are presented (Figs.4.2 and 4.3) that picture the interaction between the parts of the system model (Fig.4.1), including the interaction with the verification process presented in Chapter 3, Fig.3.1. Below, we give a detailed description of the sequence of steps of the model for its implementation:

1. After the verification process shown in chapter 3, section 3.3, Fig.3.1, the system continues with the transaction registration, a procedure that is illustrated with Fig.4.2. The transactions such as the tender rules ($TR(b)$), bidder offer ($BO(b)$), the FOL formula proof

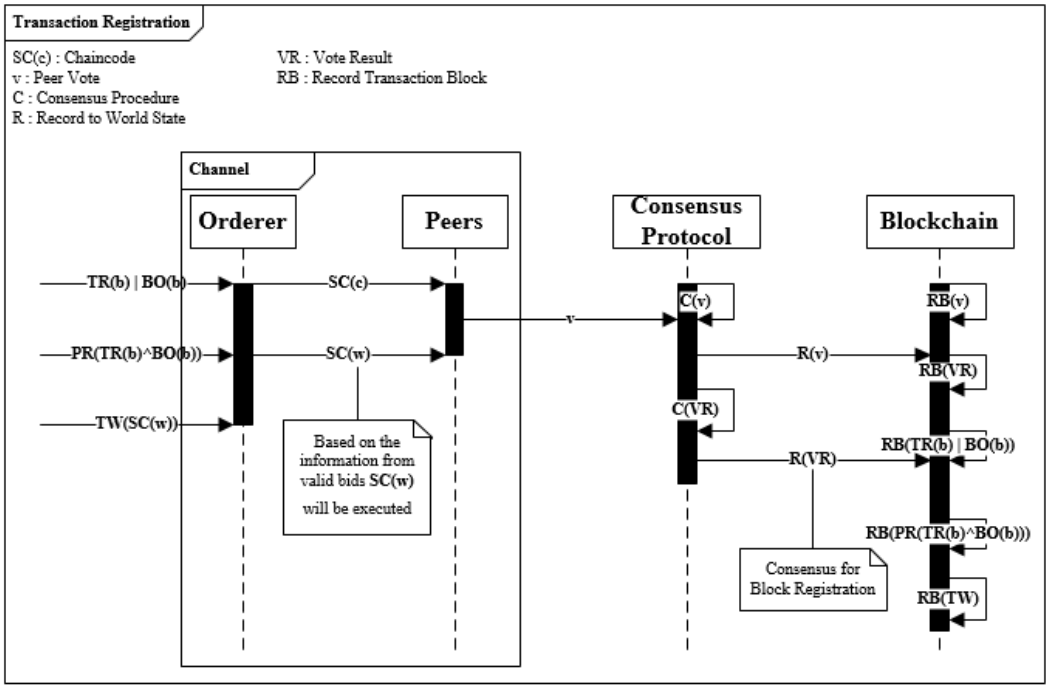

Fig. 4.2 Transaction registration.

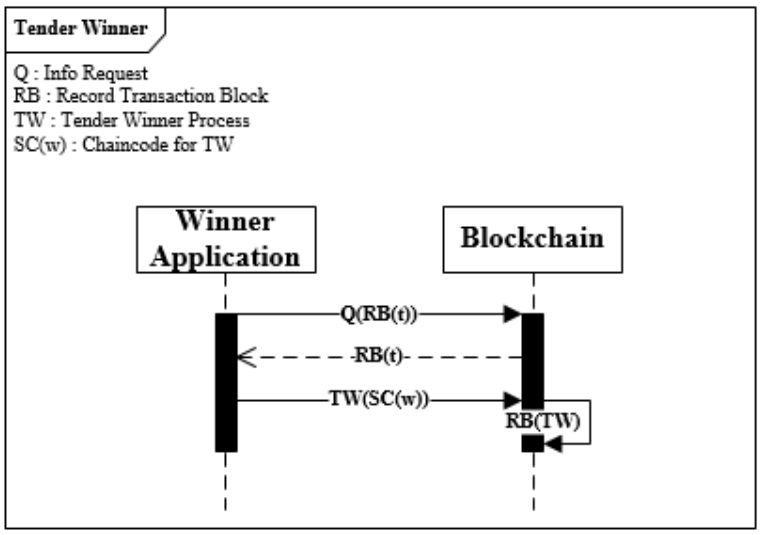

Fig. 4.3 Tender Winner

result ($TR(b) \wedge BO(b)$), or the tender winner record (TW) can be recorded. In any of the cases, the orderer node sends the Smart contract ($SC(x)$) to the peer nodes, the peers by consensus vote ($C(v)$) to validate ($R(VR)$) the registration of a block to the Blockchain record(RB).

2. After the bids are received the next step is to find a winner, that is done by the Winner Application, procedure represented in Fig.4.3 where the Winner application requests information ($Q(PR)$) about the valid offers (PR) to the Verifier, and under the criteria of higher profit with lower cost the tender winner is established . This criteria is implemented through an algorithm based on the Simplex Method (an implementation example will be shown in chapter 5), which together with the result of the test carried out by the verifier, determines which proposal is the best. When the result of the best proposal is obtained, the Blockchain is sent to execute a chaincode ($TW(SC(w))$) to record the transaction of the operation that defined the winner of the tender.

3. Finally, as we saw in Fig.4.1 in Section 4.1, there is a view of the system (user interface) where any citizen, bidder or government entity can query the information stored in the blockchain. This information provides transparency and traceability to the system, strengthening justice for bidders and giving confidence on the information that is consulted.

The sequence diagrams exposed in this section, result in the series of steps of the Formal blockchain-based system, to facilitate the code implementation of the model.

4.3 Blockchain network

Finally, the permissioned Blockchain network model is presented, highlighting the description of the operation of the network for the proposed system model. The figure 4.4 shows the actors and their interaction within the Blockchain network.

There is a set of organizations that are part of the network, among the organizations are government departments and public institutions.

The government departments establish the set of configurations to determine the number of communication channels between nodes, the certification authorities, the chaincode (Smart Contracts) that must be fulfilled to carry out the consensus process, as well as the applications external to the network to run additional procedures.

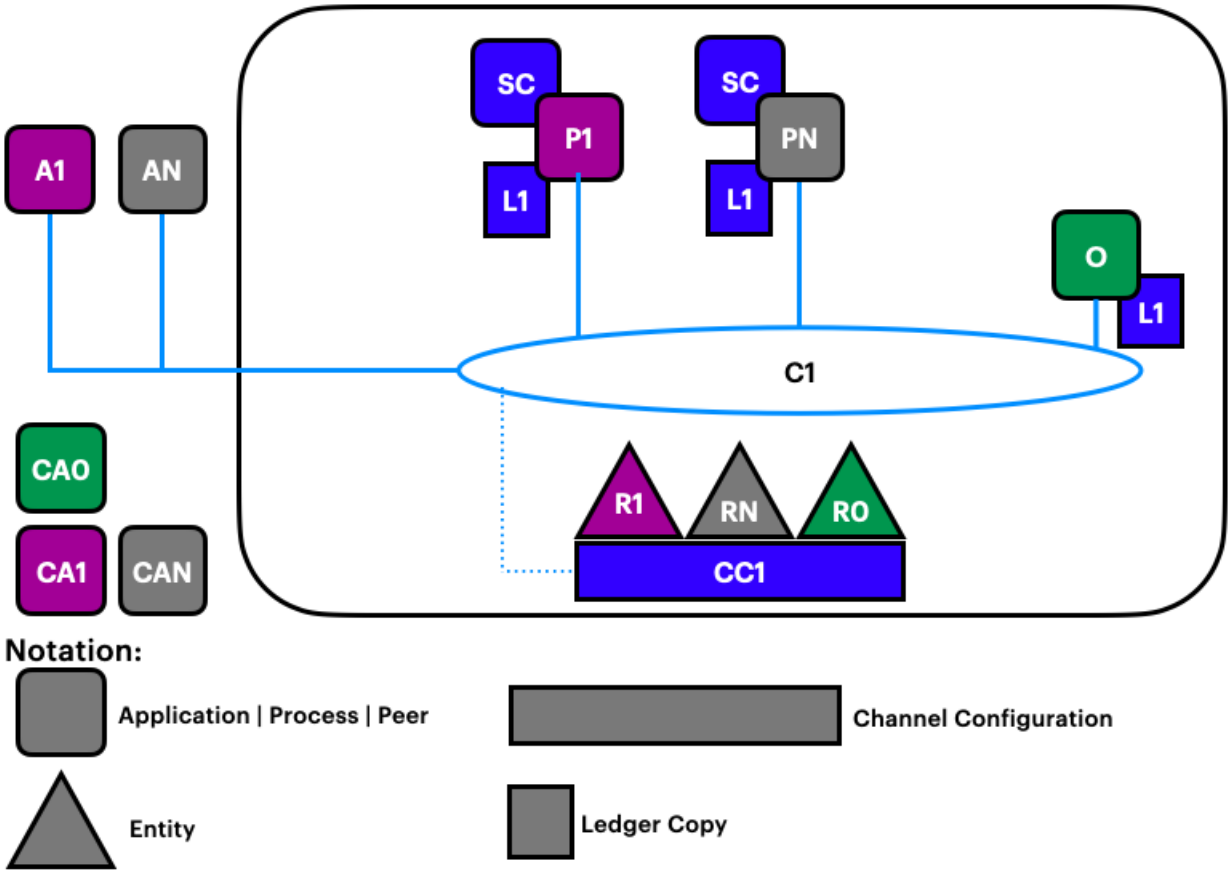

Fig. 4.4 Blockchain network.

The figure 4.4 shows us a basic Blockchain network configuration based on the operation of Hyperledger Fabric (Chapter 2 Section 2.8), to record transactions in a ledger through consensus based on RAFT (Chapter 2 Section 2.6).

The operation of the Blockchain network occurs as follows:

1. First, an ordering service (**O** in Fig. 4.4) is created as the initial management point of the network. This node is configured by an administrator on the network (**R0** in the figure 4.4). The configuration contains the policies that describe the set of actions for the network. At this point R0 has the rights to the network, this happens because R0 is the only member of the network.

2. Certifying authorities (**CA** in the figure 4.4) are established to manage organizations' access to the Blockchain network. Organizations must be authenticated and accepted to later participate in consensus and management of transactions that occur in the network.

Each organization participating in the network owns a valid digital certificate, therefore, there is the same number of certificates for each authorized organization within the network.

3. As more network management nodes are added, a channel is built as the primary communication mechanism. In this channel, network nodes participating establish communication between themselves. There can be multiple channels, however, as shown in figure 4.4, communication starts with one channel denoted as **C1**. To offer confidentiality between nodes one can create a private communication channel for such nodes that have this requirement. This allow us to create channels for different criteria and even to identify instances of collusion. In this case, we use only one channel as transparency is required.

4. With administrators registered and a communication channel created, nodes that have a copy of the ledger (**L1** in figure 4.4) can start registering transactions.

In order to record the transactions of the actions carried out on the network, smart contracts (**SC** in figure 4.4) are necessary. These contain the established policies under which actions on the system will be executed along with their corresponding transactions.

Finally, external applications such as the results given by the Z3 automated verifier or the optimization application to choose the winner of the tender (**AN** in figure 4.4) can be connected through the communication channel, in case it is required as an additional element for the network.

In this chapter the entire design of the system model has been presented, in the following chapters we will analyze the results obtained from a prototype built with all the elements of this chapter.

Chapter 5

System Implementation & Experimentation

In this chapter, an implemented prototype of the system model is presented. The an automated verifier (It is the icon of a square with arrows in the figure 4.1) is implemented used the Z3 tool. With this, the offers received in the tender process will be tested automatically as described in Chapter 3. To shown how this tool is used, we present an example. Afterwards, relevant aspects of the Blockchain network implemented in Hyperledger Fabric are presented, as well as how to query the transactions recorded on the ledger.

5.1 Z3 Python SMT-Solver

An SMT type problem is classified as a decision problem for logical formulas regarding the combination of theories such as arithmetic, arrays, and uninterpreted functions. The Z3 tool [4] is an efficient SMT Solver consisting of an API (Application Programming Interface), with different algorithms to solve problems with a varied combination of theories.

For the proposed model, the Python-based API is used to test the logical formulas presented in the Chapter 3 as it contains useful implementations of the necessary algorithms. The output is saved in a text file so that it can be registered in the Blockchain network as a transaction. This file is also used in the definition of the selection criteria for the winner of the tender.

5.1.1 Bidders' offers automated verification

Consider the example presented in Chapter 3, this can be represented in the Z3 API as follows:

```
from z3 import *
Bidder, LegalID, QualityC = Bools('Non numerical')
s = Solver()
s.add(Bidder, LegalID, QualityC)
print(s.check())
s.model()
f = Function('f')
x, y, z = Ints('x y z')
A = Array('A')
ns = Implies(x + 2 == y,
f(Store(A, x, 3)[y - 2]) == f(y - x + 1))
solve(Not(ns))
```

This implementation code corresponds to example 7 of section 3.1 in Chapter 3, an example that contains all the formalization carried out in a tender process, in the code the variable **s** establishes the non-numeric specifications to be evaluated, while in the **ns** variable sets the numeric specifications, both of which are evaluated by check and solve methods, which are part of the Z3 Solver API for Python.

As it can be seen, the output of this process will depend on the compliance with the tender rules. For this particular case, the result is satisfactory so the created plain text file will contain the word "sat" meaning that the result of the verification was satisfactory, that is, the offer logically complies with the requirements of the tender.

In our case, this result indicates that the test performed on the offer is valid, then that offer is considered as a candidate to win the tender.

This verification result will create a transaction on blockchain as evidence of an offer being approved or not. Additionally, this result can be use as a criterion for selecting the best offer.

For this to happen, a blockchain network and smart contract as the one described in Chapter 4 need to be put in place. This is explained in the next section.

5.2 Fabric Network

This section presents relevant pieces of code of the blockchain system described in Chapter 4 that is implemented on the Hyperledger Fabric framework for permissioned blockchain (2.8). We also shown the transactions recorded during the tender as described in Fig.3.1. For that purpose, the Hyperledger Explorer tool will be used.

This process starts with the list of requirements defined by a government entity in a tender process. This is represented by the JSON structured shown on Fig. 5.1 These must be the same as in the proofs with the automatic solver.

```
type Bid struct {
  ID       string `json:"id"`
  Budget   string `json:"budget"`
  Taxes    string `json:"taxes"`
  Cert     string `json:"certification"`
  Market   string `json:"market"`
  Lambda   string `json:"lambda"`
  AddedAt  uint64 `json:"addedAt"`
}
```

Fig. 5.1 Requirements structure

This structure is later registered as parameters for different chain codes, of which the transactions of procedures such as offer registration, offer verification result, and winning offer registration will be made. After that, the chain code that describes the transactions to be recorded is included. In Fig. 5.2 the fragment indicating the transaction creation is presented. There we can see how a chaincode is created based on the previous structure.

In the case of choosing the winner of the tender, information on the blocks registered in the Blockchain is consulted, specifically in the attribute that indicates whether they passed the solver proof, then the information is requested first, Fig. 5.3 shows the code for this particular chaincode.

The implementation of smart contracts was mainly shown because they are the most important element for achieving consensus between the nodes, and as we saw, a smart contract can be designed with different requirements depending on what you want to register in the Blockchain network.

Figure 5.4 shows the summary of the blocks registered in the Blockchain, their hash information, size, the channel to which it belongs, all in read-only form. As we can see, transacction X indicated the registration of an offer.

If we click on that transaction, we see in detail the information of the transaction stored in a particular block. In this case, the offer and biddes... All this is presented in Figure 5.5 shows

```
func (s *SmartContract)
CreateDocumentUsingTenderContract
(ctx contractapi.
TransactionContextInterface,
functionName string, documentData string)
(string, error) {
    if len(documentData) == 0 {
        return "",
        fmt.
        Errorf("Please
␣␣␣␣␣␣␣␣provide␣correct␣document␣data")
        }

    params :=
    []string{functionName, documentData}
    queryArgs :=
    make([][]byte, len(params))
    for i, arg := range params {
        queryArgs[i] = []byte(arg)
    }

    response := ctx.GetStub().
    InvokeChaincode("document_cc",
    queryArgs, "mychannel")

    return string(response.Payload), nil

    }
```

Fig. 5.2 Chaincode.

```
func (s *SmartContract)
getQueryResultForQueryString(ctx
contractapi.TransactionContextInterface,
queryString string)
([]Offers, error) {

    resultsIterator, err :=
    ctx.GetStub().
    GetQueryResult(queryString)
    if err != nil {
        return nil, err
    }
    defer resultsIterator.Close()

    results := []Car{}

    for resultsIterator.HasNext() {
        response, err :=
        resultsIterator.Next()
        if err != nil {
                return nil, err
        }

        newOffer := new(Offer)

        err =
        json.Unmarshal(response.Value,
        newOffer)
        if err != nil {
                return nil, err
        }

        results =
        append(results, *newOffer)
    }
    return results, nil
}
```

Fig. 5.3 Winner chaincode.

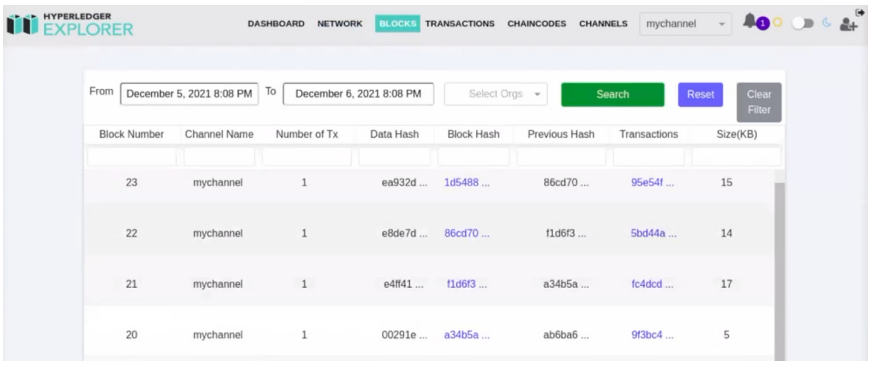

Fig. 5.4 Blocks

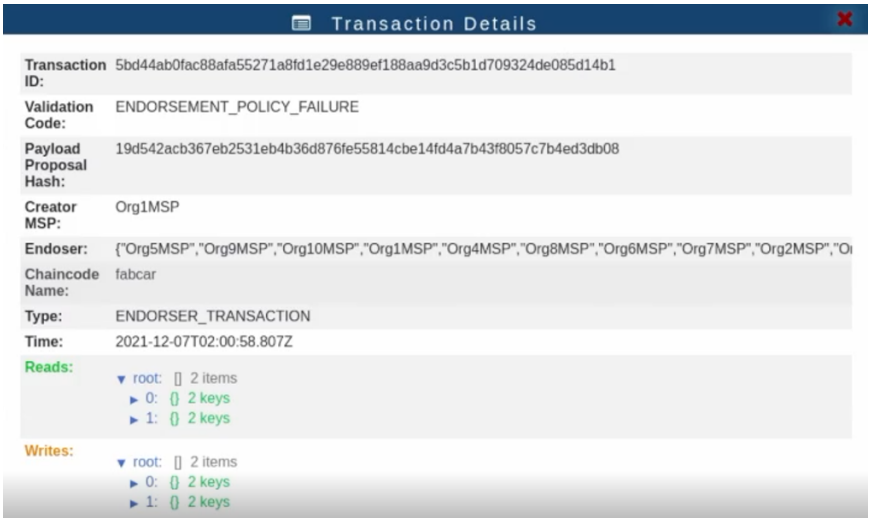

Fig. 5.5 Transaction

The information related to every interaction between the actors of our system is registered and available for consultation, this greatly strengthens transparency, since it reduces the possibilities of collusion between the participants and entities that are part of the tender process.

5.3 Criteria for the tender winner

Once the system has automatically selected a set of valid offers, now a winner need to be selected. For the selection of the winner of the bidding process, the following criteria are considered.

- Have the validation of the automated test by the Z3 tool.
- Have the most optimal result of the Simplex method.
- If in the previous criteria a tie is reached, the tiebreaker criterion is that one of the bids has non-numerical specifications in addition to the minimum requested.

The above elements are the input parameter for the simplex optimization method that will determine the offer that meets such criteria. Fig. 5.6 shows the implementation of the simplex optimization method.

5.3.1 Tender winner selection

Taking as reference the example presented in the Chapter 3 , here is an example of implementation of the simplex method to determine the winner of the model presented in section 4.2 of Chapter 4. The code receives the numerical specifications to check the proposals with a satisfactory result by the Z3 solver. (Fig. 5.6).

Throughout this chapter it is concluded that it is possible to take the design of the model to an implemented prototype.

```
import numpy as np

def pivot():
    l = list(d[0][:-1])
    jnum = l.index(max(l))
    m = []
    for i in range(bn):
        if d[i][jnum] == 0:
            m.append(0.)
        else:
            m.append(d[i][-1]/d[i][jnum])
    inum =
    m.index(min([x for x in m[1:] if x!=0]))
    s[inum-1] = jnum
    r = d[inum][jnum]
    d[inum] /= r
    for i in [x for x in range(bn)
    if x !=inum]:
        r = d[i][jnum]
        d[i] -= r * d[inum]

def solve():
    flag = True
    while flag:
        if max(list(d[0][:-1])) <= 0:
            flag = False
        else:
            pivot()

def printSol():
    for i in range(cn - 1):
        if i in s:
            print("x"+str(i)+
            "=%.2f" % d[s.index(i)+1][-1])
        else:
            print("x"+str(i)+"=0.00")
    print("objective is %.2f"%(-d[0][-1]))
```

Fig. 5.6 Simplex code.

Chapter 6

Conclusions

To finish, we will make an analysis of what was achieved during this project, we will review what else can be done to this work and to close a conclusion of all this work is offered.

6.1 Discussion

In chapter 2 the work related to our proposal was presented, F. S. Hardwick[12] proposes a public Blockchain solution implemented in Ethereum to carry out open tenders with transparency. An important disadvantage in this proposal is the consumption of resources (energy, hardware), this makes the global implementation of the proposal difficult, in addition to the fact that some countries have restrictive policies in large-scale public proposals. In contrast, in Chapter 4 we show that our proposed model considers using a private Blockchain to facilitate the adoption of restrictive policies, and considerably reduce the consumption of resources.

On the other hand, J. Deshpande et.al.[9] propose a solution similar to our proposed model, it is also based on private Blockchain with multiple organizations to perform decentralized consensus. However, in its implementation there is no formal verification procedure, consequently, properties such as integrity, fairness and reliability are not guaranteed. Given this, Chapter 4 showed that the proposal offers a verification process in the model system, in which through Satisfiability Modulo Theories it validates the offers of the participants. Also chapter 4 showed some sequence diagrams to carry out the implementation of the system.

In Chapter 3, the results of the proof of the theorem, the automation of the verification of the bids of the tender participants and in Chapter 5 the implementation of the system based on

Hyperledger Fabric as proof of concept were presented, with the purpose of illustrating that the proposal can be developed.

6.2 Future work

As future work, the proof of concept is available, from which an attractive and user-friendly view can be developed for users, or even make that system more efficient.

There is also the possibility of automating the procedure of giving logic-based formatting to the automatic solver inputs, so that all processes are fully automated.

6.3 Work conclusion

In conclusion, the following was achieved:

- Model a formal verification scheme to provide a more robust solution to a complex problem such as a tender process.
- Model a system that uses the verification scheme from the previous point, and that together with a Blockchain network can offer greater confidence in a tender process.
- Define logical formulas that are the basis for the formalization of offers in a bidding process.
- Demonstrate the correct operation of the logical formulas, and thus have the confidence that the verification works correctly.
- Show the possibility of designing a prototype that materializes the previous points.

This proposal contains a model based on private Blockchain, providing automation, integrity, traceability and transparency to the model designed for e-Procurement, i.e. is more reliable and adjusted to the resources and needs of government departments or public institutions in tender procedures. The proposal manages to reduce manual intervention to most of its procedures since these are done autonomously within the blockchain system, consequently reducing the risk of malicious acts or collusion.

Finally, some of the results presented in this work have already been published in “R. Dávila, R. Aldeco-Pérez and E. Bárcenas, "Tender System Verification with Satisfiability Modulo

Theories", 2021 9th International Conference in Software Engineering Research and Innovation (CONISOFT), 2021, pp. 69-78, doi: 10.1109/CONISOFT52520.2021.00021" and in "R. Dávila, R. Aldeco-Pérez and E. Bárcenas, "Formal Verification of Blockchain Based Tender Systems", Programming and Computer Software, Springer ISSN 0361-7688, 2022".

References

[1] (2020). Hyperledger fabric. [online] https://hyperledger-fabric.readthedocs.io/en/latest/index.html.

[2] (2020). Public tendering rules in the eu. [online] https://europa.eu/youreurope/business/selling-in-eu/public-contracts/public-tendering-rules/index_en.htm.

[3] (2020). Road data exchange layer. [online] https://x-road.global/.

[4] (2022). Programming z3.

[5] Barrett, C., Sebastiani, R., and Tinelli, S. C. (2009). *Handbook of Satisfiability*, volume 185.

[6] Biere, A., Heule, M., van Maaren, H., and Walsh, T. (2009). *Handbook of Satisfiability: Frontiers in Artificial Intelligence and Applications*.

[7] Bockmayr, A., Weispfenning, V., and Maher, M. (2001). Chapter 12 - solving numerical constraints. In Robinson, A. and Voronkov, A., editors, *Handbook of Automated Reasoning*, Handbook of Automated Reasoning, pages 751–842. North-Holland, Amsterdam.

[8] Buterin, V. (2014). A next-generation smart contract and decentralized application platform—white paper. *Ethereum Project*.

[9] Deshpande, J. J., Gowda, M., Dixit, M., Khubbar, M. S., Jayasri, B. S., and Lokesh, S. (2020). Permissioned blockchain based public procurement system. volume 1706.

[10] Dávila, R., Aldeco-Pérez, R., and Bárcenas, E. (2021). Tender system verification with satisfiability modulo theories. In *2021 9th International Conference in Software Engineering Research and Innovation (CONISOFT)*, pages 69–78.

[11] Grosan, C. and Abraham, A. (2011). *Rule-Based Expert Systems*, pages 149–185. Springer Berlin Heidelberg, Berlin, Heidelberg.

[12] Hardwick, F. S., Akram, R. N., and Markantonakis, K. (2018). Fair and transparent blockchain based tendering framework - a step towards open governance.

[13] Huynh, T. T., Nguyen, T. D., and Tan, H. (2019). A survey on security and privacy issues of blockchain technology.

[14] Ismail, L. and Materwala, H. (2019). A review of blockchain architecture and consensus protocols: Use cases, challenges, and solutions. *Symmetry*, 11.

[15] Klemperer, P. (1999). Auction theory: A guide to the literature. *Journal of Economic Surveys*, 13.

[16] Nakamoto, S. (2018). A. the bitcoin whitepaper by satoshi nakamoto - mastering bitcoin, 2nd edition [book].

[17] Peña, R. (2020). *Introducción a los modelos de optimización*.

[18] Yaga, D., Mell, P., Roby, N., and Scarfone, K. (2019). Blockchain technology overview. *arXiv*.