CHAPTER 1

A Blockchain Use Case for Car Registration

Tiago Rosado

Instituto Superior Técnico, Universidade de Lisboa, Portugal

André Vasconcelos

INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Portugal

Miguel Correia

INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Portugal

CONTENTS

Blockchain enables the development of decentralized business models with enhanced security for critical data. In this chapter we present a car registration system based on the Hyperledger Fabric blockchain technology. This system considers several government entities in a single country, but might be extended to a cross-border scenario, possibly supporting car data sharing at the level of the European Union (EU).

The system – BCar – handles the processes related to car registration management, e.g., registering a vehicle, changing ownership status, and registering a leasing contract between a lessor and a lessee. This system can simplify the information exchange among multiple states as the car registration information is distributed to each government entity in a single decentralized system. We analyse the benefits and implications of the blockchain technology application and present an evaluation of the system's performance.

1.1 INTRODUCTION

Blockchain technology has reached the mainstream due to the fact that it is the basis for many cryptocurrencies, which have been shaking the notion of currency in recent years. Bitcoin, as the first and best known application of blockchain technology, is a digital currency that does not rely on a central authority to be managed [7]. The decentralization provided by blockchain technology can be considered a direct competitor to organizations relying on a centralized business model, such as banks and governments. In an utopian scenario, blockchain technology could present a decentralized collection of services competing with government's services such as property registration, citizen registration and even a financial system replacement that could render most of the government's work useless.

Nevertheless, organizations can also look at blockchain technology as an innovation that is an opportunity for them to improve their efficiency. Currently many organizations are investigating and implementing blockchain technology to the benefit of business efficiency and government transparency. One example comes from the Estonian government that has been implementing services using this technology since 2012 [16].

Considering the potential of blockchain technology, the main objective of this chapter is to present a car registration infrastructure based on blockchain. A car registration system based on blockchain can decentralize this kind of registries and, in consequence, improve data availability and resilience to faults. As a set of entities, ranging from leasing companies to government offices, rely on the car registration system, a decentralized system based on blockchain can improve performance and security when compared with a centralized solution.

Given the decentralization inherent to blockchain, it is crucial to understand the role of an entity such as the national car registry. As it will be discussed, a blockchain application for car registration can still take in consideration the authority of the national registry entity. With this requirement, a blockchain based car registration system can benefit from decentralization but still maintain a centralized authority to partially manage and control the system.

One may consider blockchain technology has a limited set of use cases, however the case for a car registration system based on blockchain may provide a starting point to research and implement further government services and information systems based on this technology. Regarding car related services the proposed solution could provide a starting point for a single system to manage registration, tax and vehicle's characteristics and requirements for a car to be legal to drive. Regarding government registration services, blockchain may also be considered for civil, commercial and criminal registration systems, driving innovation of government services.

This technology can also represent an effort to improve government efficiency and transparency [16]. As blockchain technology tends to decentralize data storage, it is also necessary to weight the effort needed to implement the system in a full scale and identify possible struggles over this approach, ranging from the initial setup of the system, to the maintenance effort towards the needed hardware, and possible software updates. Therefore, we propose an implementation of blockchain technology for car registration using smart contracts and analyze its impact on car registration business processes.

1.2 BACKGROUND

In this section we go over background that is relevant for our work. First, we present an overview of how car registration works today, including its main components. Then, we introduce the notion of smart contract, which is at the core of most applications based on blockchain. Next, the notion of consensus is presented and the two major alternative approaches are discussed, as this is a major design decision that we had to make. These two approaches are tightly related to the two main families of blockchains, permissioned and permissionless, which are discussed next. Then Hyperledger Fabric, the blockchain technology that we selected is presented. Finally, we mention some applications currently using blockchain technology, their configuration and how businesses benefit from them.

1.2.1 Current car registration system

A car registration system, as implemented by most government entities around the world, is usually a centralized information system. This is the case of Portugal, which is the country's car registration system that we studied in more detail. This information system handles every information related to car registration and is managed by a national registry entity, although other governmental and non-governmental entities have access to services handling car registration information.

Citizens are able to request information related to vehicles, as most of the information stored in the car registration system is available for the public. Government entities responsible for controlling motorized vehicles are able to interact with the system to issue and cancel registration plates and to change the official characteristics of a vehicle.

Regarding seizure orders, lawyers, courts and solicitors are allowed to interact with the car registration system to issue or consult those orders. External entities, such as leasing companies, are also allowed to consult car records on the car registration system.

A car registration system may be composed of several servers responsible for different operations, such as having servers responsible for providing web services to external entities relying on car registration information, as tax authorities, vehicle regulation authorities or even some companies directly related to vehicle registrations, such as leasing companies. Another component of such a system is a client interface for employees handling car registry data, which can be presented in a client side application or a website providing a mean to execute operations over a car registry.

All of the system components described interact with the core of the information system which in most of the cases is a relational database hosted on a different server, responsible for managing and storing the information required to handle car registries. Each of the components described might vary on its complexity by having backup servers to tolerate system fault or attacks from malicious parties, preventing data losses and system failures on each of the components constituting the car registration system.

Within the EU there is cooperation among member states to exchange car registration information, to issue transit violations or simply consult registry information regarding a vehicle in a different EU country. Thus, some of the described system components are exposed to cross-border access to provide vehicle information to different member states. However, as opposed to using blockchain technology member states rely on the availability of each other's car registration systems to read cross-border vehicle's information.

1.2.2 Smart contracts

Smart Contracts are programs written to form agreements between users in the blockchain [2]. Using smart contracts it is possible to ensure that the clauses of a contract are accomplished automatically, and that breaching the contract is expensive or even prohibitive [10].

Blockchain establishes a consensus based on minimal trust between network nodes to execute smart contracts. When a node receives a transaction, contract functions are ran to ensure the validity of the transaction and the conditions stated in the contract are met. In case of failure the transaction is discarded by the network nodes.

By extending the capabilities of smart contracts, it is possible to run decentralized applications based on blockchain technology. Therefore, we can create applications such as car registry platform completely based on smart contracts.

1.2.3 Consensus

A blockchain is a distributed replicated sequence of blocks of data. To guarantee that all nodes have the same sequence of blocks, the nodes have to reach consensus on which are the blocks and their order. There are basically two approaches to do that.

1.2.3.1 Proof-of-work

The first approach is proof-of-work (PoW), first implemented in Bitcoin [7]. The idea is that in order for nodes to agree on the next block to add to the blockchain, there is a need to decide who should be the author of the block.

In Bitcoin, Ethereum and other blockchains, PoW is used to decide that, i.e., who is the author of the block. A PoW must be hard to produce and easy to verify. This implies the need for a computational intensive problem to be solved. The node responsible for solving the problem is the potential author of the next block to be added to the blockchain [17]. In the case of Bitcoin, a node needs to scan for a nonce value for a block so that the hash of the block starts with a defined number of zero bits.

This form of consensus has the side effect of slowing down the transaction processing within the network, as major computational work is wasted to create a block instead of processing transactions. The approach used by proof-of-work increases the difficulty of successful attacks, given the need for a great computational power to create a proof-of-work and use it to tamper blockchain data.

Given the concurrency between nodes, it is possible for two nodes to broadcast different versions of the next block with a valid PoW at approximately the same time. In this situation, other nodes start working over the first block they receive but still save the other branch, this represents a fork. A fork is removed once the next proof-of-work is found. As the new block is added to one of the branches, the branch becomes longer and the shorter branch is removed [7].

There are a set of other approaches related to PoW, e.g., Proof-of-Stake, but they are still not much adopted.

1.2.3.2 BFT approach

We designate the second consensus approach Byzantine fault-tolerant (BFT), as it follows a long line of distributed algorithms (e.g., [3, 4, 13]) that started with the work of Lamport et al. on agreement among a set of "Byzantine generals" [6]. These algorithms are known not to scale well, as they involve several steps of communication involving all nodes, on the contrary of the PoW approach that involves one node flooding the network with its block and PoW. However, for a small number of nodes, the BFT approach allows fast transaction processing (e.g., tens of thousands per second).

Regarding consensus conflicts, such as the creation of temporary forks in the blockchain, BFT algorithms do not suffer from this problem. BFT algorithms grant the property of consensus finality, as in Definition 1 [14]. Considering this property, block addition to the blockchain is immediately confirmed.

Definition 1 *(Consensus Finality) If a correct node p appends block b to its copy of the blockchain before appending block b', then no correct node q appends block b' before b to its copy of the blockchain.*

Resilience to attacks is also a matter of analysis as one of the key advantages of blockchain is assurance of immutable data, once the data is stored in the blockchain. The PoW approach in theory supports up to 50% of faulty nodes in the network, however regarding Bitcoin's approach to fault mitigation, tolerance drops to 25% [5], contrasting with BFT algorithms that support up to $\lfloor \frac{n-1}{3} \rfloor$ faulty nodes, where n is the number of nodes.

Given the possibility of forks, in Bitcoin it is a good practice to take a block as final only when 6 or more blocks have been appended to the blockchain after that block. On the other hand, this approach can be circumvented by timestamp manipulation [14]. BFT algorithms, as complying with Definition 1, support long asynchronous periods and global outages. Latency in BFT algorithms usually matches network latency, contrary to proof-of-work approach where rising block size translates in higher throughput with the downside of higher latency. As latency of the system increases the number of possible blockchain forks increases as well, resulting in more opportunities to perform double spending attacks and successfully executing them.

1.2.4 Permissionless versus permissioned blockchains

Public blockchains rely on public nodes, thus any node can: contribute to the network by running a client designed to keep a local blockchain copy, contribute to network consensus, contribute to process transactions and create blocks. Public blockchains rely on public nodes without restricting their access and actions in the blockchain. Considering public blockchains, the requirement of making public any data stored in the blockchain has the advantage of increasing data transparency but it sacrifices privacy [16].

Permissioned blockchains emerged as a necessity to restrict blockchain network participants to identifiable and explicitly authorized nodes with specific permissions [15]. Therefore, private blockchains provide mechanisms, to restrict the nodes accessing and contributing to the blockchain. As a result, private blockchains increase the confidentiality of the data they store. However, a restrict number of nodes with access to the blockchain may provide less resilience and sturdiness.

1.2.5 Hyperledger Fabric

Hyperledger Fabric is a blockchain software sponsored by the Linux Foundation and IBM. Hyperledger Fabric, as a private blockchain, restricts the nodes participating in the system to trusted and identifiable nodes. This enables for performance improvements over consensus mechanisms and can reduce the power consumption of such system. Permissioned blockchains may be used by a group of entities who may not completely trust each other [1].

Hyperledger Fabric uses an execute-order-validate architecture [15, 1], based on a modular consensus mechanism that can be adjusted to the specific need of smart contracts running on the blockchain. Through execute-order-validate, a transaction entails three different phases. During the execution phase, each transaction is executed and its correctness verified by a restricted set of peers called endorsement peers. During this phase it is possible for transactions to be executed in parallel. The second phase is assigned to an ordering service, responsible for establishing signed transactions' total order, using the consensus mechanism defined and fabricating blocks accordingly. Ordering service nodes are also responsible for updating the blockchain's state to all peers using atomic broadcast. For any of the operations, ordering service nodes do not need to execute smart contracts, know about the current application state nor validate the smart contracts. On validate phase, transactions are validated by the remaining peers of the network, checking against the trust assumptions considered for each specific application, and endorsement policies are verified. If there is no issue in this last step, the block is appended to the blockchain on each peer's local copy.

Regarding ordering services' operations related to consensus mechanisms, Hyperledger Fabric currently has three consensus mechanisms implemented [1]: a one node consensus, requiring a single node to establish total order of transactions, a method normally used to speed up the development environment of smart contracts, a crash fault-tolerant (CFT) based ordering service ran on cluster, and a BFT-SMaRt [9] consensus mechanism tolerating at most 1/3 of faults. The Ordering Service creates a block as one of the following conditions occurs: (1) the maximum number of transactions per block is reached, (2) the maximum block size is reached or (3) a certain time has passed since the first transaction was added to the block.

1.2.6 Applications

Although cryptocurrencies, as Bitcoin, are the most popular application of blockchain technology, there is a large set of applications based on blockchain backed by governments, banks and private companies. The use of blockchain and smart contracts may improve the government relationship with citizens and help government initiatives to take part of the digital world. As stated [16], this technology improves the privacy of citizens and transparency of government work.

Applying blockchain technology to business may lower operations costs and coordination efforts, as the system automaticaly handles possible conflicts [16]. Distributed ledgers can be used to better secure data itself contained in the blockchain, easily share citizen's data between government entities and secure critical infrastructures [16], such as a country's electrical grid.

The Estonian government implemented a blockchain solution in 2012 applied to registries of the national health system (e-Health Records [1]), judicial and citizen registry system. A concrete application of the blockchain technology by the Estonian government is e-Residence [2]. Through this application digital identities are issued (together with a digital ID smart card) for people and organizations around the world who want to develop a location indepen-

[1] e-Health - https://e-estonia.com/solutions/healthcare/e-health-record

[2] e-Residency - https://e-resident.gov.ee/

dent business. Using this application, e-Residents can start a company, access business banking, sign and securely send documents and declare taxes online. The Estonian government takes advantage of blockchain technology, in order to keep track of all changes performed to the system, ensuring data integrity [12].

1.3 BUSINESS PROCESSES OF CAR REGISTRATION

In this section we present an overview of the business processes involved in the operation of a centralized car registration system. A new car registration system, based on blockchain technology, may bring innovation over a centralized car registration system and change some of the business processes currently in place. An analysis of the current business processes, of the Portuguese national registry entity, will be conducted and a set of changes over the current business processes will be suggested as a way to improve those processes. The information to model the presented business processes was obtained through public domain information. Finally, we will go over a scenario using a blockchain based car registration system and model some of the business processes associated with this new system.

1.3.1 Current business processes

Analyzing the current car registration system, a user can interact with the car registration system by going to a national registry office and requiring information regarding a vehicle or requiring to change a vehicle's registry. Then a national registry employee consults the car registration system and provides the information to the user or updates the vehicle's registry. Information changes, for registered vehicles, mostly require the owner or an authorized party to fill in a form[3].

On the other hand, a vehicle owner can have a narrower but more direct access to the car registration system by using an online platform, although most of the information updates still require approval from a registry employee. In both methods described, most of the operations available require some sort of payment to be executed. A simplified version of the required actions to interact with the current car registration system, is shown as a business process model diagram in Figure 1.1 and Figure 1.2.

1.3.2 National registry office operations

A back-office process, as modelled in Figure 1.2, is needed to fully process the citizen's request. The back-office process starts when a registry employee looks for the list of pending requests in the car registration system. As a

[3] Documento Único Automóvel - http://www.irn.mj.pt/sections/irn/a_registral/servicos-externos-docs/impressos/automovel/requerimento-de-registo/downloadFile/file/DUA_modelo_unico.pdf

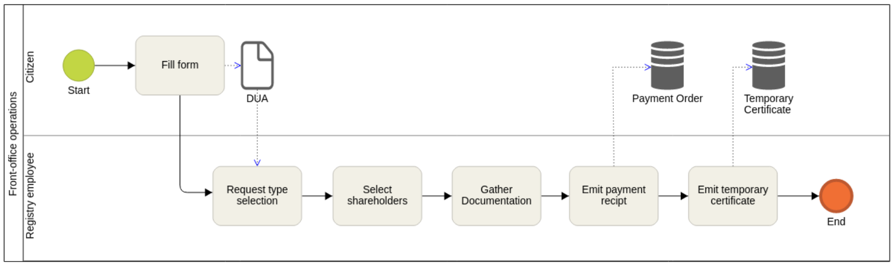

FIGURE 1.1 Over the counter (or front-office) operations

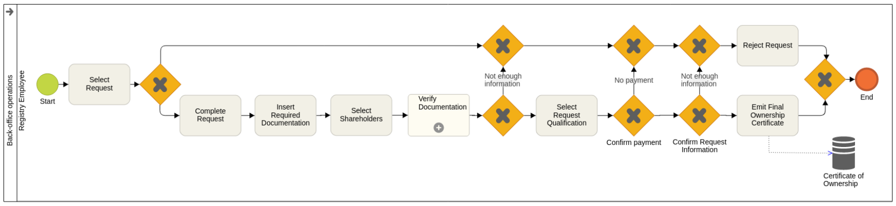

FIGURE 1.2 Back-office operations

registry employee selects a pending request, he or she may reject the request or complete a pending request. When completing a request, the necessary documentation may be introduced. After this step, the list of shareholders to the processing request is inserted and a document verification is executed by the registry employee.

Finally, the request result is issued and the process is finished. A request result may vary with the type of request specified in the form by the citizen. In case of a vehicle ownership change, this process results in a final ownership certificate being issued to the new owner.

1.3.3 Business processes on a blockchain based system

Given the properties of a blockchain based car registration system, the business processes presented in the previous sections can be modified to benefit from blockchain properties. A blockchain based car registration system should work following the principle that information updates over vehicle's data are correct unless this information is latter proven to be wrong. Thus we propose a blockchain based car registration system on which change requests over vehicle information can be firstly registered to the system, as soon as the request is issued by the respective participant and the request's payment is confirmed. This approach takes advantage of blockchain technology and its immutability to simplify business processes.

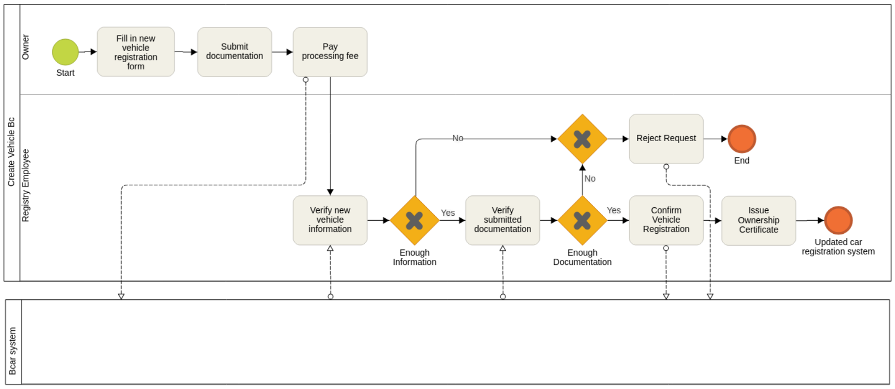
Create Vehicle Bc
Owner
Start
Fill in new vehicle registration form
Submit documentation
Pay processing fee
Registry Employee
No
Reject Request
End
No
Verify new vehicle information
Enough Information
Yes
Verify submitted documentation
Enough Documentation
Yes
Confirm Vehicle Registration
Issue Ownership Certificate
Updated car registration system
Bcar system

FIGURE 1.3 Proposed process for creating a vehicle in a blockchain based system

1.3.3.1 Register a new vehicle

Taking into account the registration of a newly fabricated vehicle into a blockchain based car registration system, we assume the intervention of a registry employee is recommended. Given this operation might require several authorizations from authorities, such as the Department of Motor Vehicles. On the other hand, registering a vehicle implies creating a new registry, thus allowing user's requests to directly create entities in the system can lead to a system bloated with malformed asset registries.

We propose a registration process for a new vehicle in the car registration system as presented in Figure 1.3. A request is made through a front-office or an online portal and it is latter processed by a national registry employee. The employee will verify the information available in the request and complete some of the information. On a second step, the employee will verify that all request's information is according to the documents provided and might add new documentation before completing the process. As every information and documentation is correct and verified by a national registry employee, the request is fulfilled, a new vehicle is registered in the blockchain registration system.

1.3.3.2 Request a change of ownership

Regarding ownership change process we propose a two phase process. The current owner of a vehicle wishing to pass his ownership position to another entity is required to fill an online form with all the necessary information and documentation. Once this form is submitted, a pending ownership change is submitted to the blockchain based car registration system, as modeled in Figure 1.4, which is latter confirmed by the prospect owner.

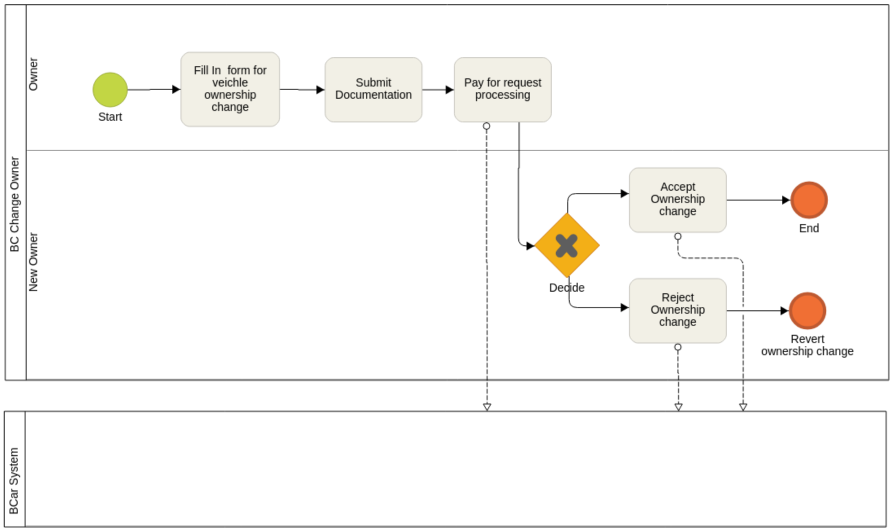

FIGURE 1.4 Proposed process for changing ownership information of a vehicle in a blockchain based system

As the vehicle's owner issues a change of ownership request, with information about the new owner and its share of ownership, the vehicle ownership is registered as a pending ownership and the prospect new owner is required to take action. The prospect owner is then able to accept or reject the ownership change (Figure 1.4). Only after the prospect owner confirms the ownership change, the new owner is effectively registered as owner of the vehicle in the blockchain and the old owner is removed as owner. Considering the prospect owner of the vehicle rejects the ownership change, the vehicle ownership information in the blockchain is updated so that the old ownership settings remain valid and the prospect owner is no longer tied to the vehicle.

The proposed process aims to simplify the ownership change using blockchain technology to improve overall efficiency of the process, considering a national registry employee is only required to intervene when the process is incorrectly executed.

1.3.3.3 Register as guarantee

When registering a vehicle as loan guarantee, the process can be modified as modeled in Figure 1.5. Thus, when a vehicle owner wants to register a vehicle as guarantee for a loan, he issues a request with the intended documentation. In the request, the vehicle owner is required to specify who is the creditor entitled for this guarantee, the total value to which the vehicle is given as guarantee and the penalty for missing payments. Once the payment

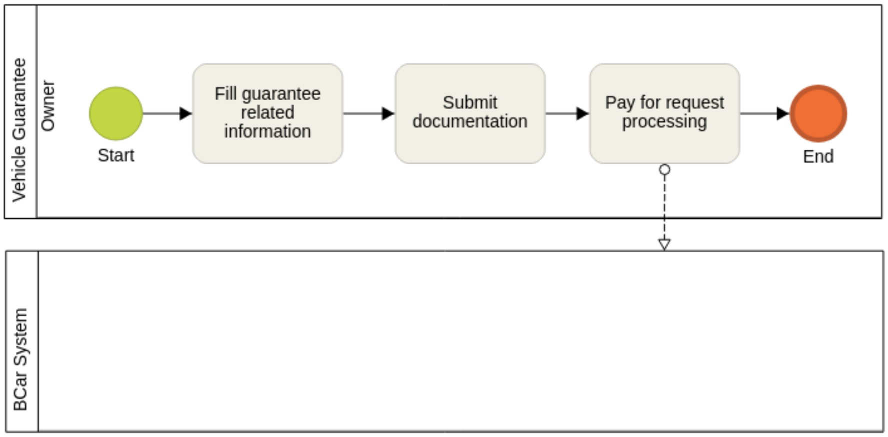

FIGURE 1.5 Proposed process for registering a vehicle as guarantee for a loan in a blockchain based system

for the request is confirmed, the blockchain is updated with new information regarding the guarantee tied to the vehicle.

1.3.3.4 Associate lease contract

When owning a vehicle, the owner is able to issue a lease contract signed by the owner, as a lessor and a third-party, as a lessee, with a specific duration. A lease contract expects the lessee to make regular payments for a specified number of months. In exchange, the lessor gives permissions for the lessee to use the vehicle for the duration specified on the contract.

A lease contract association in a blockchain based car registration system is divided in two phases. The lease registration process is started by a vehicle owner, submitting lease information, as well as, a copy of the lease contract in digital form, as shown in Figure 1.6. Once the request's payment is completed, the lease details are associated with the vehicle's information in the blockchain based system. As this operations hands over responsibility to the lessee, the lease information remains in a waiting state. Then the lessee is required to confirm or deny his involvement in the contract. If the lessee accepts the lease contract, the contract is validated in the system. Otherwise, the lease operation is reverted once the prospect lessee rejects the pending proposal.

1.3.3.5 Execute a vehicle seizure

A vehicle seizure occurs when, by judicial order, the ownership of an asset, in this case a vehicle, is forcibly changed in order to liquidate owner's debt. A

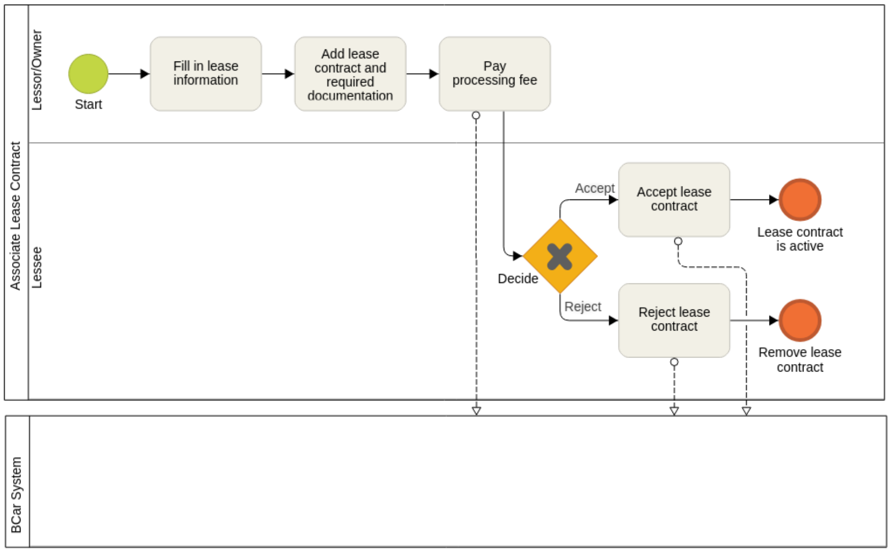

FIGURE 1.6 Proposed process for registering a lease in a blockchain based system

vehicle seizure request is currently made by a judicial officer and latter executed by national registry employees. On a blockchain based car registration system, we propose the process to be executed by a judicial order, providing the supervision role to a registry employee.

Once a vehicle's seizure order is emitted, a judicial officer must fill in a seizure request with the required information and documentation, as the judicial order number issuing the seizure. As soon as a judicial officer submits the required information for seizure of the vehicle, the action is submitted to the blockchain and the ownership of the vehicle is changed. On the other hand, the entity specified in the seizure order becomes the new owner of the vehicle.

1.4 BCAR SYSTEM

In this section we describe the proposed car registration system. At first we consider the requirements of a car registration system, starting from the currently centralized system and its properties.We then propose to implement a blockchain based information system with granular access control. A set of use cases is defined, as part of the requirements for the proposed system and a light consideration on the permissions of each participant is presented. Considering vehicle information required by the European law, a data model is presented.

1.4.1 Requirements

Car registration systems in Europe are controlled and maintained by each member state. As blockchain technology relies on distributed nodes, the control detained by government entities should be adapted. Therefore, a car registration system based on blockchain technology can distribute most of the maintenance effort and some of the system's control by network nodes. However the government entities of each European member state could still detain control over the registered cars in that state. The European member states or even the European Commission should also be able to have the role of a supervising authority.

As part of a car registration system's requirements, at least the following use case scenarios were identified:

A. Main use cases:

   1. A car seller wishes to transfer car ownership to a car buyer.
   2. A leasing company provides a contract to a client with the clauses of the client using the car for a defined period of time on which the leasing company is responsible for paying maintenance and insurance expenses in exchange for a defined monthly payment by the client.
   3. A car owner submits a request to give the car ownership as a guarantee to a creditor in case the car owner does not fulfil the contract.

B. Secondary use cases:

   1. A car manufacturer submits a request to create a car registry entry for a newly produced car.
   2. Given a judicial order and in order to liquidate a car owner debt, the car ownership is transferred to the entity responsible for collecting the debt payments.

1.4.2 Solution

Considering the granular access control required for a government service, such as the car registration system, it was necessary to select a permissioned blockchain. As we intend to provide most of the blockchain control to a set of entities such as the national registry institution of each European member state.

This configuration ensures the safety and confidentiality of data. As a private blockchain, only authorized nodes detained by the government entities or by trusted external entities to join the network.

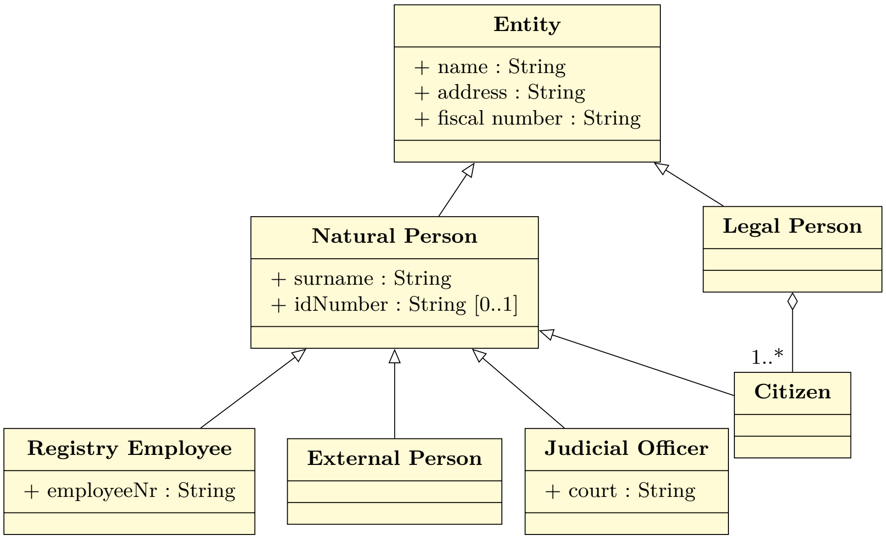

FIGURE 1.7 Proposed car registry system's participant types

1.4.2.1 Participant definition

As the car registration system requires access control to registry information, a set of participants were defined along with their specific permissions. Considering main and secondary use cases specified in Section 1.4.1, five concrete participants were defined. A class diagram as an overview of the informations stored for each participant of the car registration system can be consulted in Figure 1.7.

1.4.2.2 Citizen

A Citizen type participant encompasses the simplest entity able to take ownership of a vehicle. As other entities are represented in the system, a citizen inherits from an abstract Natural Person participant. A Natural Person participant type is defined by a name, an address (which represents the official residence of the entity) and its fiscal number. Optionally the personal identification document number can be stored with the remaining information, although only a fiscal number is required to identify a citizen in the car registration system. A citizen is able to execute the following operations, when owning a vehicle: Start a change of ownership of a vehicle; Start a lease contract; Accept or reject a lease contract; Request a lease contract cancellation; Accept or reject a lease contract cancellation; Add the vehicle as guarantee for a loan; Request for a vehicle guarantee to be cancelled.

1.4.2.3 Legal person

A legal entity, is represented as a Legal Person type participant on the proposed car registration system. Similarly to a Citizen type participant, it is identified by a name, an official fiscal address and a fiscal number which are all encoded as strings on the data model. A fiscal number is the primary identifier of a legal person in this car registration system. A Legal Person type participant, also requires a list of entities which are Citizen type participants. This entities are the Legal Person's owners or entities responsible for this Legal Person's actions on the car registration system.

As a Legal Person may not represent a singular entity, car registry operations can not be performed by a Citizen type participant. Car registry operations on behalf of a Legal Person participant are required to be performed by the Citizen identified as owners or Citizen participants in charge of the Legal Person's actions on the car registration system.

1.4.2.4 Judicial officer

Operations executed as fulfilment of a judicial order can only be executed by judicial entities or by a national registry employee. Judicial entities are represented in the proposed car registration system by Judicial Officer type participants.

A Judicial Officer participant type is extended from a Natural Person participant type, thus includes the same data required to identify a Citizen type participant already described. On the other hand a Judicial Officer is required to have an additional field called court, which is encoded as a string, and registers the judicial entity for whom the Judicial Officer works and identifies the judicial entity issuing the car registry operations.

1.4.2.5 Registry employee

A registry employee is defined as an extension to a Natural Person type participant. A registry employee is required to be associated with a employee number, in order to uniquely identify each employee and the registry entities he works for. A Registry Employee type represents users of the car registration system, working for national registry entities. When compared to the other participant types, registry employees have unrestricted permissions to car registries in the information system. Registry Employee participants are able to solve conflicts on the system and are still accountable for their actions, given the nature and architecture of the blockchain technology.

1.4.2.6 External person

An additional participant was required to be defined in order to include operation's permissions within the car registration system which were executed by external entities to the national registry and were not included in the above

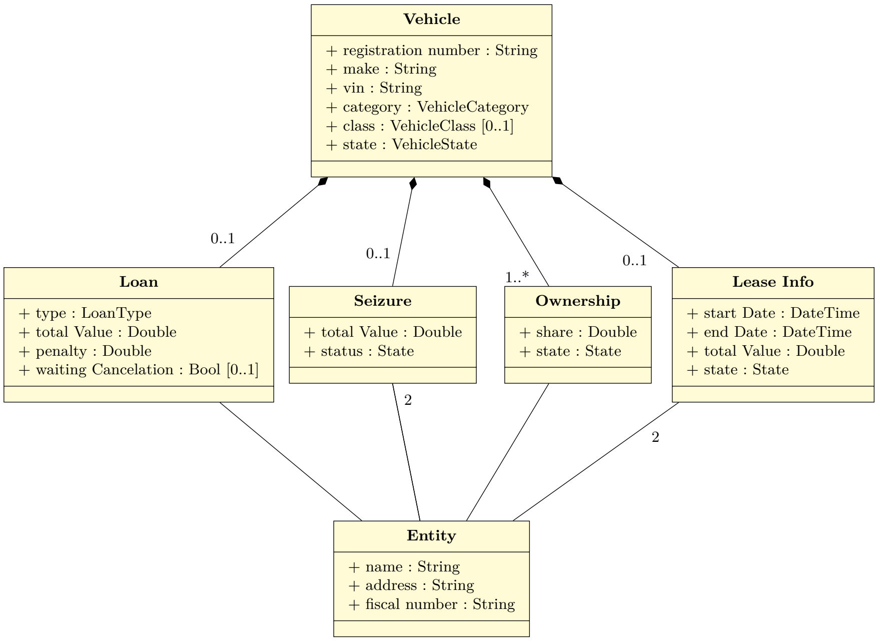

FIGURE 1.8 Proposed vehicle data model

mentioned participant types. For example, consider the use case described in Section 1.4.1, an authority needs to verify the ownership of a car. Thus, employees working for an external entity allowed to read car registry information are registered as a External Person type participant.

1.4.3 Data model

Based on the European regulations [8] and the vehicle registration modification form[4], a simplified data model was built. The resulting data model was also based on the use cases specified in Section 1.4.1.

A vehicle is registered in the car registration system with the following information: a registration number, a make, a vehicle identification number and category. Vehicle owners are identified and their shares are registered in a Ownership object. A Ownership object for each owner is associated with the vehicle, as presented on Figure 1.8.

Considering vehicle category's classification [11], a vehicle can be categorized as able to carry passengers (M), able to carry goods (N), 2 or 3 wheel vehicle or a quadricycle (L) and agricultural and forestry tractors and their

[4] Documento Único Automóvel - http://www.irn.mj.pt/sections/irn/a_registral/servicos-externos-docs/impressos/automovel/requerimento-de-registo/downloadFile/file/DUA_modelo_unico.pdf

State	Description
Active	Vehicle in circulation
Inactive	Vehicle no longer in circulation (e.g. exported vehicle)
Destructed	Vehicle with a destruction order issued
Suspended	Vehicle reported as not proper for circulation
Stolen	Vehicle reported as stolen by authorities

TABLE 1.1 VehicleState object possible states.

trailers (T). In case of vehicles carrying passengers (M) or goods (N), they are also categorized in classes by their weight as light duty vehicles when under 3500 Kg and heavy duty vehicles when equal or above 3500 Kg. A vehicle can be registered in the car registration system in several conditions; this information is stored as a VehicleState type. Different available states a vehicle can belong to are presented in Table 1.1.

Analysing special conditions of a vehicle, information regarding leases, loans or seizures are also stored in the car registration system. Lease information related to a certain vehicle is stored in the LeaseInfo object associated with the vehicle, as in Figure 1.8.

A vehicle can be submitted as collateral for loans, therefore the entity responsible for the loan and possible penalties are registered, as well as the total value of the loan and its type. A loan guarantee is intended to provide a guarantee for loan payment, in this case the vehicle. Thus, we consider each vehicle must be tied to a maximum of one loan at a time.

Lastly, we consider the case of vehicle seizures resulted from court orders or other judicial entity's decisions. Information regarding seizures is stored in a SeizureInfo object.

1.4.4 Car registry operations

Next we present the operations available in the proposed car registration system.

1.4.4.1 Create a vehicle

In order to simplify interactions over the blockchain-based car registration system we assume this information is registered manually by national registry personal. As most of the vehicle information is given by external entities responsible for managing motorized vehicle regulation. Therefore when a vehicle is registered, the following information is required: registration number (number plate), Vehicle Identification Number (VIN), make, model, vehicle category and the owners information along with their respective shares.

1.4.4.2 Change ownership

Changing ownership of vehicles is separated in two steps and is specific for each owner, as vehicles can have multiple owners with different shares each. As expected, this operation is only allowed to be executed by the owners of the vehicle subject to ownership change.

Thus, a ownership change needs to be initialized by the current owner wishing to give up his position on a certain vehicle, as proposed in Section 1.3.3.2. This step requires the owner to issue a *Change Owner* transaction, specifying the VIN, the registration number and the make of the vehicle, as well as the list of the new owners to which the current owner will give his share of the vehicle and the share percentage that he wants to transfer to the new owners.

As the first step is concluded, the vehicle ownership changes to the new owner. However the new ownership is still required to be approved by the new owners. This state is represented in Figure 1.9 as *Waiting Ownership Change* state. In order to complete the ownership change, only the new owner registered in the initial *Change Owner* transaction is able to confirm this operation issuing a *Confirm Ownership* transaction specifying the vehicle's VIN, the registration number, the make, and the ownership share. Only after this step the ownership information is considered valid, regarding judicial obligations and the new owner is considered legitimate owner.

1.4.4.3 Lease

As proposed in the business process presented in Section 1.3.3.4, the process for registering a lease contract is separate into a two steps operation. The owner is proposed to issue a *Create Lease* transaction which can be canceled through a *Cancel Lease* transaction by the lessor, the lessee or by a registry employee. On the other hand, the lease registration can be accepted either by a registry employee or by the prospect lessee issuing a *Confirm Lease* transaction.

As part of the *Create Lease* transaction, the VIN, the registration number, the make of the vehicle, the start and end dates of the contract, the expected total value of the lease, and the third-party tied to the lease contract are required.

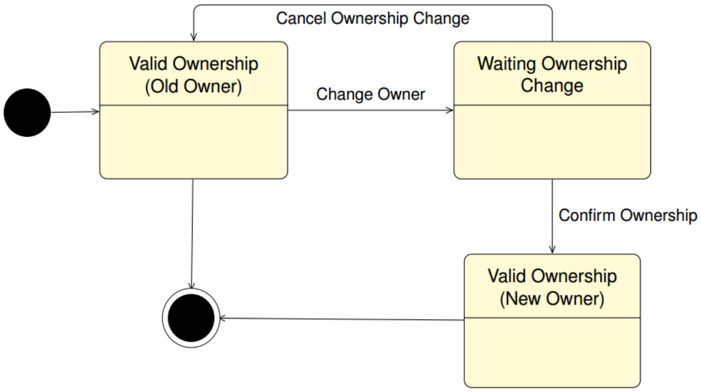

FIGURE 1.9 Change Ownership transaction flow

Once the *Create Lease* transaction is correctly executed, the vehicle enters a *Waiting lease confirmation* state. In order to successfully register a lease contract, the lessee registered in the initial *Create Lease* transaction is required to issue a *Confirm Lease* transaction containing the VIN, the vehicle's make and the registration number as well as the total value of the lease contract. In case the transaction issued by the lessee matches with the information passed by the lessor, in the *Create Lease* transaction, the system registers the lease as associated with the vehicle and the vehicle enters the *Active lease* state.

For a lease contract termination to happen, it is necessary to issue a *Cancel Lease* transaction. Considering the *Cancel Lease* operation is issued by a judicial officer, this action takes effect immediately, thus the lease contract is revoked.

In case the *Cancel Lease* transaction is issued by a lessee or a lessor and the vehicle is in *Active lease* state, the process takes two steps, as both parties need to agree on cancelling the contract. To this extent, the first transaction to be issued will be the *Cancel Lease* transaction, thus the lease contract state is changed to *Waiting lease cancellation*. At this stage, any of the parties can revert the cancellation process by issuing a *Cancel Lease Termination*, specifying the Vehicle Identification Number (VIN), the vehicle's make and the registration number. For the lease contract to be actually terminated, after a *Cancel Lease Termination* transaction, it is necessary for the other party to issue a *Confirm Lease Termination* transaction with the same arguments as *Cancel Lease* transaction. It is possible to reject or cancel the lease termination process by issuing *Cancel Lease Termination* transaction.

1.4.4.4 Seize vehicle

Given a process on which a vehicle owner is subject to a court to liquidate his debts, judicial officers can seize a vehicle or issue a pending seizure. Thus, if a pending seizure is in place, the asset can not be sold or change ownership until the process is solved. Latter the same process can also result in a seizure and the ownership of the asset can only be changed according to court order.

In order to issue a pending seizure for a vehicle, a judicial officer is required to submit a *Issue Pending Seizure* transaction. A *Issue Pending Seizure* transaction should contain information about the owner of the vehicle, the total value in debt, the creditor and information about the vehicle as the Vehicle Identification Number (VIN), the registration number and the make.

Considering a case on which the *Pending seizure* state needs to be reverted, a *Cancel Seizure* transaction should be emitted, containing the same information required for the *Issue Pending Seizure* transaction. The *Cancel Seizure* transaction can only be issued by a judicial officer.

In order to effectively seize the asset, a *Issue Seizure* transaction is required to be issued. A *Issue Seizure* transaction requires the same information used by *Issue Pending Seizure* transaction, but also requires to refer the date on which the seizure order was emitted and the court order number supporting this decision. When a vehicle is in its initial state, no seizure nor pending seizure associated, it is possible for the judicial officer to directly emit a *Issue Seizure* transaction.

During *Issue Seizure* transaction execution, the ownership of the vehicle is changed so the owner implied in the seizure is removed as owner. The creditor specified in the *Issue Seizure* transaction is then assigned as an owner of the vehicle.

1.4.4.5 Register as guarantee

In order to register a vehicle as guarantee, the vehicle's owner is required to order a *Register Guarantee* transaction mentioning the creditor, the type of loan (Collateral or Mortgage), the total value of the loan given to the vehicle owner and the penalty which the debtor needs to pay in case the loan is paid ahead of time. Vehicle information is also registered on the *Register Guarantee* transaction, such as the registration number, the make and the Vehicle Identification Number (VIN). During the *Register Guarantee* transaction execution, a set of rules is verified in order for the transaction to be successful. The transaction can only be issued by the owner and it is not possible to register a vehicle as a guarantee to a loan if the vehicle is already tied as a guarantee to a previously issued loan. The *Register Guarantee* transaction will not succeed if the vehicle is subject to a seizure or a pending seizure by a judicial order.

It is possible to cancel the guarantee by issuing a *Cancel Guarantee* transaction. A creditor can issue a *Cancel Guarantee* transaction with immediate effects, requiring to identify the vehicle through the vehicle identification number, registration number and make.

On the other hand, if *Cancel Guarantee* transaction is issued by the vehicle owner it requires the creditor to confirm or deny the operation using accordingly the transactions *Confirm Guarantee Cancellation* or *Reject Guarantee*

Cancellation. Both transactions are required to include the Vehicle Identification Number (VIN), the registration number and the make.

A *Confirm Cancellation* transaction is required to be issued by the creditor associated with the loan guarantee or by a national registry employee, for the vehicle to enter *No loan guarantee* state. On the other hand, as a *Reject Guarantee Cancellation* transaction is issued by the creditor or by a registry employee, the guarantee remains valid.

1.4.4.6 Change vehicle state

As presented in vehicle data model of Figure 1.8 and explained in Section 1.4.3, a vehicle can be registered in the proposed car registration system according to different states. A change in a registered vehicle's state is usually presented by an external entity such as the Department of Motor Vehicles; however as simplification, the national registry employee is in charge of this update on the car registration system. Thus, when a national registry employee receives a request for updating the state of a vehicle in the system, he issues a *Change State* transaction.

A *Change State* transaction is required to specify the vehicle's make and the registration number to which the state update is necessary. This transaction is also required to include the new state to which the vehicle will transit to and the vehicle identification number. The vehicle states which a vehicle can have were already described in Section 1.4.3.

1.5 IMPLEMENTATION

Based on the data model described on Section 1.4.3 a similar data model was created using a domain specific language of Hyperledger Composer[5] which latter was deployed to a Hyperledger Fabric version 1.1 based network. Hyperledger Composer Modeling Language is an object-oriented modeling language designed to define the domain model for a business network defined for the Hyperledger Fabric.

Each transaction described through out Section 1.4.4 was also modeled in Hyperledger Composer Modeling Language. Every information regarding the created data model was stored on the Hyperledger Fabric's blockchain with no off-chain database handling car registry records. Furthermore, all the developed smart contract functions where developed using Javascript and Hyperledger Composer API version 0.19.7. Hyperledger Composer provides an interface adaptable to any language to interact with Hyperledger Fabric blockchain.

Specific access control rules where defined using Hyperledger Composer access control language. Hyperledger Composer access control is define through a set of rules which can detail CRUD access control considering the participant

[5] Hyperledger Composer - https://hyperledger.github.io/composer/latest/

executing the transaction or reading asset information. Access control is fine grained with the ability to restrict access to certain participants only through specific pieces of data of an asset and through specific transactions. Therefore a set of rules were created to enforce permissions for each participant and each transaction.

1.5.1 Implemented access control rules

Regarding the vehicle creation, as presented in Sections 1.3.3.1 and 1.4.4, this operation can only be executed by a Registry Employee participant type. Thus, a rule allowing Registry Employee type participants to issue *Create Vehicle* transactions was created.

Considering *Change Ownership* transactions, all participants of the system are entitled to issue this transaction. However, the verification of this rules is done through the smart contract's code, as verifying this rules through the access control mechanism leads to performance issues. Specific rules verifying that only the owner, owner of the Legal Person owning the vehicle or a registry employee have permissions to issue *Confirm Ownership* and *Cancel Ownership* transactions, were defined.

In the process of creating a lease, as issuing a *Create Lease* transaction, only the Registry Employee participants and vehicle owners are entitled to execute this transaction. As presented earlier in this Section, given performance issues, the access control verification when issuing a *Create Lease* transaction is provided through the smart contract. Regarding lease transaction flow the same rules, as described for *Create Lease* transaction, are defined for *Confirm Lease*, *Cancel Lease*, *Confirm Lease Termination* and *Cancel Lease Termination* transactions.

All transactions regarding seizure flow, thus *Issue Pending Seizure* , *Issue Seizure* and *Cancel Seizure* transactions have the same rules as following. Hyperledger Composer access control mechanism have a rule allowing Judicial Officer and Registry Employee participant types to create each of the vehicle seizure flow transactions.

Registering a vehicle as guarantee (*Register as Guarante* and (*Cancel Guarantee* transactions) requires that only vehicle owners or registry employees are allowed to execute the transaction. Thus, the verification of ownership and verifying that the transaction issuer is a registry employee is made through the smart contract's code. Regarding *Cancel Guarnatee*, *Confirm Cancelation* and *Reject Cancelation* transactions. Finally, Hyperledger Composer access control mechanism only allows Registry Employee participant types to create *Change Vehicle State* transaction.

1.5.2 Hyperledger Fabric configuration

In this section we go over each system's components, explaining their contribution in the Hyperledger Fabric infrastructure. Then we present how a client

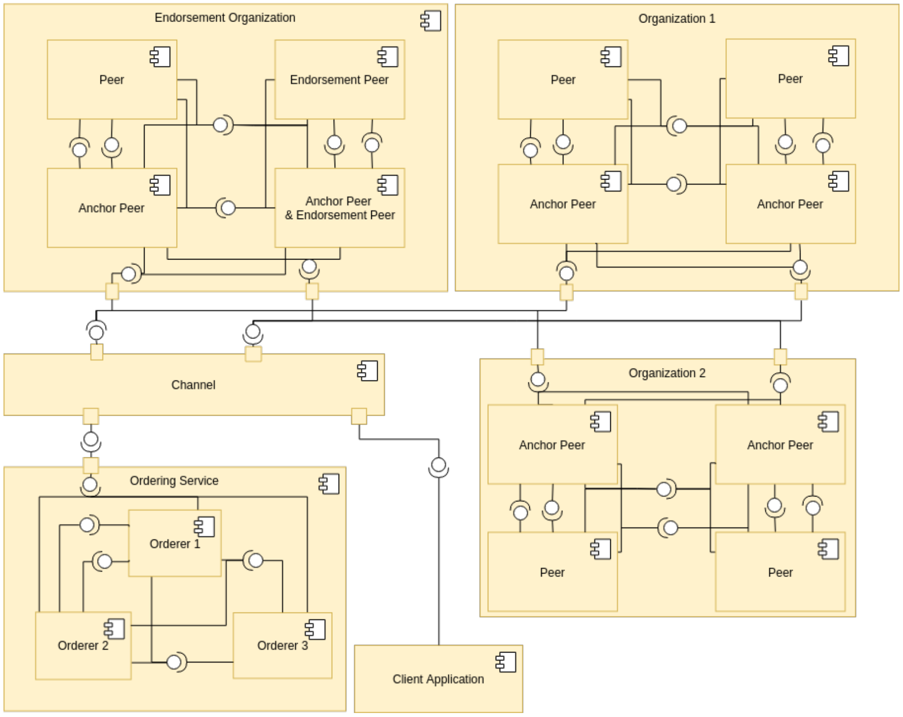

FIGURE 1.10 Component diagram of Hyperledger Fabric infrastructure

is able to interact with the Hyperledger Fabric's blockchain and describe each step required to execute a transaction.

1.5.2.1 System components

As part of the Hyperledger Fabric infrastructure, a network peer can be a endorsement peer, an anchor peer or a normal peer. As each peer type is not mutually exclusive, it is possible for a peer to be both a endorsement peer and an anchor peer, as in Figure 1.10.

Smart contracts' data is stored as key-value pairs in this database. Thus, when executing a transaction, the state database is used to make chaincode execution more efficient. As alternative, CouchDB[6] can be used as an external state database, providing additional query support and richer queries when compared with the default state database.

An endorsement peer is responsible for validating transactions by executing transactions' chaincode (functions of the smart contract required by the transaction). An anchor peer is responsible for communicating with the Hy-

[6] CouchDB - http://couchdb.apache.org

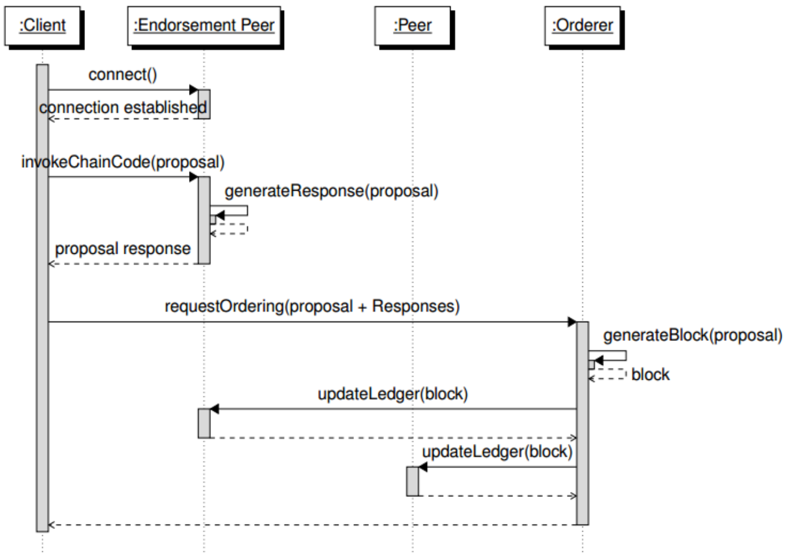

FIGURE 1.11 Block proposal and addition mechanism for Hyperledger Fabric.

perledger Fabric network exterior to the organization on which it belongs. Finally, a normal peer is only responsible for receiving a block of transactions issued by the ordering service and updating its local blockchain with the corresponding block.

The ordering service is a core component of any Hyperledger Fabric infrastructure. The service is composed by ordering nodes (see Figure 1.10), responsible for reaching a consensus on building a block of, signed and verified, transactions to update the blockchain. The endorsement policies in place may vary according to the configuration setup defined by network's administrators.

As in Figure 1.10, three organizations are presented and each of them is composed by four peers. An endorsement organization (which can be a National Registry Entity) is responsible for endorsing transactions, thus is composed by two endorsement peers and two normal peers. In addition, an endorsement peer and a normal peer also act as anchor peers. The remaining organizations are composed by four peers each and two of those peers act as anchor peers for the organizations.

An ordering service is presented, composed by three orderers in order for the infrastructure to be fault tolerant.Finally we consider a client application which is responsible for providing a platform for the end-users to use the BCar system proposed.

In order for the different components to interact and to maintain access control to information stored in the blockchain, Hyperledger Fabric provides channels. In case of the BCar system, a unique channel is used in the Hyperledger Fabric infrastructure.

1.5.2.2 System behavior

In order to use the proposed system, a client connects to an Hyperledger Fabric network running the described smart contract functions. Transactions with intent to update blockchain's state require to be submitted through Fabric's consensus mechanism, as shown in Figure 1.11. Then he submits the blockchain's update proposal to endorsement peers responsible for verifying and signing transaction proposals. According to a customizable policy, a specific number of valid proposal signatures are required in order for the transaction proposal to be accepted as valid by the network peers.

As the client collects the required proposal signatures, the list of signatures along with the proposal are sent to a ordering service, responsible for grouping signed transactions into a block, ordering them. Once transactions are ordered into a block, the block is sent to every peer on the network and the blockchain is updated with the newly proposed block.

1.6 RESULTS

To assess the performance of the blockchain based car registration system proposed we had at our disposal a single Virtual Machine (VM) with 8vCPU, 32 GB of RAM and 40 GB of HDD space, running Ubuntu Xenial (16.04.5 LTS). Regarding the software setup, tests were performed on top of Docker containers running over Docker v18.06.0-ce. Each peer, certified authority node and orderer was running on the base image of Hyperledger Fabric x86 64 v1.1.0. As state database, instances of Hyperledger Fabric CouchDB image version x86 64 v0.4.6 were used.

In order to measure system's performance, Hyperledger Caliper [7] software was used. Given Hyperledger Caliper is still in development phase, a version based on commit *c37860b042*[8] was adapted for the BCar registration system to be tested.

Hyperledger Caliper is a blockchain benchmark tool developed within the Hyperledger projects. Caliper allows to measure the performance of multiple blockchain implementations, one of them is Hyperledger Fabric, given a set of use cases. This tool can produce reports containing various performance indicators, such as transactions per second,transaction latency and resource utilization.

Considering Hyperledger Fabric nodes setup, we configured the system with a total of 9 containers with a total of 2 Hyperledger Fabric peers. The system was configured using solo consensus mechanism, thus a single orderer

[7] Hyperledger Caliper - https://www.hyperledger.org/projects/caliper

[8] Hyperledger Caliper Repository - https://github.com/hyperledger/caliper/tree/c37860b042

container was setup for the experiments. A total of 2 organizations where configured, requiring a certified authority running on a separate container for each organization.

Organization's peers used in the experimental setup were configured as endorsement peers. Thus an additional container (Chaincode Peer) is used to execute the transaction's required chaincode. This ensures process isolation from endorsement peer process. Hyperledger Fabric infrastructure was setup to ensure a single transaction signature from a endorsement peer is enough for the ordering service to accept the operation.

1.6.1 Methodology

A set of functions was selected to sample the system's overall performance, based on the principal transaction flows described in Chapter 1.4. Thus, we evaluated the performance of *Create Vehicle*, *Change Vehicle State*, *Change Ownership*, *Issue Seizure* and *Register as Guarantee* functions.

Each function was tested three times with a varying number of fixed throughput issued by the testing software, for each block size configuration. Firstly, a set of 100 vehicles was created and a set of 100 transactions where issued against the BCar registration system with a fixed send rate of 50 Transactions Per Second (TPS). Then, a set of 200 vehicles was created and a set of 200 transactions was issued with a fixed send rate of 100 Transactions Per Second (TPS). Finally, a set of 400 vehicles was created and 400 transactions were issued with a fixed send rate of 200 Transactions Per Second (TPS) against the BCar registration system. As each set of transactions was sent, the time taken for the system to fulfill the requests was measured, then the throughput was calculated dividing the number of transactions successfully executed by the time taken to execute such transactions.

As each component of the Hyperledger Fabric system is running on a single Virtual Machine, we took advantage of Hyperledger Caliper functionalities by tracking the RAM and CPU usage of each Docker container.

Regarding block size and maximum time for block formation, three experiments were conducted. Each function's throughput and latency information was measured against a block size of 1MB, with 250ms timeout configuration, 2MB block size with 500ms timeout and a 4MB block size with a 1 second timeout.

1.6.2 Evaluation

Considering the four functions selected to evaluate the system we averaged the throughput of each function given a certain block size, based on the information of Table 1.2. As we analyzed on Section 1.2, a higher block size is expected to provide a higher throughput to a block chain based system, as in Figures 1.12 and 1.13.

The maximum throughput achieved was around 7.67 Transactions Per

	1MB			2MB			4MB		
Send Rate (TPS)	**50**	**100**	**200**	**50**	**100**	**200**	**50**	**100**	**200**
Create Vehicle	5	6	timeout	6	6	5	6	6	5
Change Ownership	6	7	7	7	7	7	7	8	7
Issue Seizure	6	8	8	7	8	7	8	8	7
Register as Guarantee	6	7	6	7	8	6	7	8	6
Change State	7	7	8	6	8	8	7	8	8

TABLE 1.2 Throughput in TPS considering the variation of block size.

	1MB			2MB			4MB		
Send Rate (TPS)	**50**	**100**	**200**	**50**	**100**	**200**	**50**	**100**	**200**
Create Vehicle	13.74	25.68	timeout	12.68	25.62	57.87	11.65	24.15	64.25
Change Ownership	12.28	21.04	40.37	11.43	21.29	42.24	10.05	19.26	47.03
Issue Seizure	12.05	11.09	19.61	11.35	19.82	36.82	9.15	18.9	42.54
Register as Guarantee	39.73	11.75	20.74	9.75	18.66	46.85	10.05	20.63	45.36
Change State	10.79	20.29	35.38	11.56	18.32	36.85	11.18	19.73	36.39

TABLE 1.3 Latency (seconds) considering the variation of block size

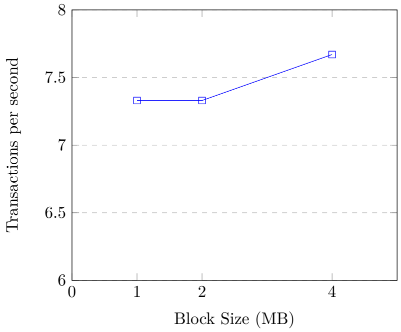

FIGURE 1.12 Throughput of Issue Seizure function

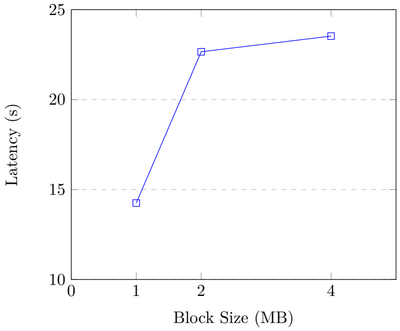

FIGURE 1.13 Latency of Issue Seizure function

Name	Memory(max)	Memory(avg)	CPU(max)	CPU(avg)
dev-peer0.org2	124.7MB	116.8MB	90.68%	25.30%
dev-peer0.org1	128.9MB	120.9MB	82.95%	25.42%
peer0.org2	77.5MB	65.9MB	96.47%	38.99%
peer0.org1	74.1MB	62.9MB	91.21%	40.05%
orderer	23.2MB	19.1MB	22.63%	3.91%
couchdb.org2	134.4MB	124.7MB	195.93%	82.82%
ca.org1	7.2MB	7.2MB	0.00%	0.00%
ca.org2	7.0MB	7.0MB	0.01%	0.00%
couchdb.org1	144.9MB	127.0MB	211.25%	83.66%

TABLE 1.4 Register Guarantee function (100.0 TPS send rate / 1MB block size)

Second (TPS), with a 4 MB block size and a 1 second timeout, when issuing a *Issue Seizure* transaction. However this same transactions presented a latency of 23.53 seconds. Analizing the graphics (Figures 1.12 and 1.13), we can conclude the best compromise to optimize both throughput performance and lower latency is achieved using a 2 MB block size.

1.6.3 System bottlenecks

In this section we go over the system resources statistics, collected throughout the system's evaluation, in order to understand the obtained results. Considering Table 1.3, we notice that across all tested block sizes, the latency suffers a major increase when comparing a 100 TPS send rate with the 200 TPS send rate. The overall latency is roughly doubled when the send rate is around 200 TPS, with no major increase in throughput. Raising the thesis that such performance is due to the single orderer setup used for the system tests, we expect an overload of CPU usage or RAM consumption. However, analysing the system statistics for 1 MB block size during a 100 TPS send rate test, as in Table 1.4, it is clear the ordering peer (orderer) is not requiring above average CPU or memory usage. Furthermore, the ordering peer presents low memory and CPU usage during the test. Regarding all components used in the test, CouchDB instances reveal to be the most resource hungry containers.

On the other hand, the use of Hyperledger Composer framework might have a significant impact on Hyperledger Fabric overall performance. It is possible that the use of native Hyperledger Fabric chaincode to develop the car registration system's smart contracts may lead to performance improvements. Considering the complexity of a car registration system it is plausible that the quantity of information stored in the blockchain and the complexity of available operations might provide a reason for such latency and throughput results. Even though access control rules embedded in the smart contract improved this results, it is still noticeable that the access control mechanisms used in BCar system contribute to lower performance regarding throughput and latency.

1.7 CONCLUSION

Through this chapter, a use case for blockchain technology in public registries was presented. Blockchain technology as of its decentralized nature can provide a different approach to registry storage. As of this fact, a blockchain based car registration system was proposed (BCar), taking advantage of decentralization capabilities of the technology.

Based on the car registration business processes and public information about the systems currently in place, a set of requirements for a car registration system was defined. A data model was build considering the operations identified for a car registration system as well as the participants of the system. The set of available operations for the BCar registration system was then described.

Considering BCar system roles, 5 participant types were defined with different permissions over the vehicle registration operations. A Citizen type participant defined as a singular entity able to own a vehicle. A Legal Person type participant as a collective of citizens owning a vehicle. A judicial officer participant entitled to execute judicial orders in the system. Then, a registry employee participant presented as a supervisor of the system and as able to perform operations to maintain a correct car registration system. Finally, an External Person type was presented to allow employees of external entities, as tax authorities, to read vehicle registry information.

Regarding vehicle registry's operations we presented a set of operations, such as the initial vehicle's registration transaction and two step transaction flows as *Change Ownership* transaction flow, taking advantage of a blockchain based car registration system. Those operations enabled for registry employees to lower their intervention in those transaction flow.

Finally we conducted a set of performance tests over a simple configuration of the Hyperledger Fabric system and conducted an analysis over the data collected from those tests. As of the test results we analyzed the throughput and latency results over a set of system's functions varying the block size of the system.

As the BCar registration system was designed, it encompasses most of the operations over car registries. However, the system focuses only on car registry data. Considering such a system, blockchain technology could be used to join the car registry systems of the EU member states into a distributed system. Furthermore, blockchain technology could be applied to every other government registry domains, as the civil registry system, the land registry or the business registry.

Bibliography

[1] Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, et al. Hyperledger Fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the 13th ACM EuroSys Conference*, 2018.

[2] Vitalik Buterin. Ethereum: A next-generation cryptocurrency and decentralized application platform, 2014.

[3] Miguel Castro and Barbara Liskov. Practical Byzantine fault tolerance. In *Proceedings of the 3rd USENIX Symposium on Operating Systems Design and Implementation*, pages 173–186, February 1999.

[4] Miguel Correia, Giuliana Santos Veronese, Nuno Ferreira Neves, and Paulo Veríssimo. Byzantine consensus in asynchronous message-passing systems: a survey. *International Journal of Critical Computer-Based Systems*, 2(2):141–161, 2011.

[5] Ittay Eyal and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. *Lecture Notes in Computer Science*, 8437:436–454, 2014.

[6] Leslie Lamport, Robert Shostak, and Marshall Pease. The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, July 1982.

[7] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. *www.bitcoin.org*, page 9, 2008.

[8] Council of European Union. Council directive 1999/37/ec of 29 april 1999 on the registration documents for vehicles. *OJ*, L 138:57–65, 1999-06-01.

[9] João Sousa, Alysson Bessani, and Marko Vukolic. A Byzantine fault-tolerant ordering service for Hyperledger Fabric. In *Proceedings of the 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, 2018.

[10] Nick Szabo. The idea of smart contracts. *Nick Szabo's Papers and Concise Tutorials*, 1997.

[11] UN-ECE. Consolidated resolution on the construction of vehicles (r.e.3), ece/trans/wp.29/78/rev.6, July 2017.

[12] Sarah Underwood. Blockchain beyond Bitcoin. *Communications of the ACM*, 59(11):15–17, 2016.

[13] Giuliana Santos Veronese, Miguel Correia, Alysson Neves Bessani, Lau Cheuk Lung, and Paulo Verissimo. Efficient Byzantine fault tolerance. *IEEE Transactions on Computers*, 62(1):16–30, 2013.

[14] Marko Vukolić. The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication. *Lecture Notes in Computer Science*, 9591:112–125, 2016.

[15] Marko Vukolić. Rethinking permissioned blockchains. In *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, pages 3–7, 2017.

[16] Mark Walport. Distributed ledger technology: Beyond block chain. *Government Office for Science*, pages 1–88, 2015.

[17] Xiwei Xu, Cesare Pautasso, Liming Zhu, Vincent Gramoli, Alexander Ponomarev, An Binh Tran, and Shiping Chen. The blockchain as a software connector. *Proceedings of the 13th Working IEEE/IFIP Conference on Software Architecture*, pages 182–191, 2016.