

Article

NFT-Vehicle: A Blockchain-Based Tokenization Architecture to Register Transactions over a Vehicle's Life Cycle

Juan Carlos López-Pimentel ^{1,*}, Luis Alberto Morales-Rosales ², Ignacio Algreto-Badillo ³
and Carolina Del-Valle-Soto ¹

¹ Facultad de Ingeniería, Universidad Panamericana, Álvaro del Portillo 49, Zapopan 45010, Jalisco, Mexico; cvalle@up.edu.mx

² Facultad de Ingeniería Civil, CONAHCYT-Universidad Michoacana de San Nicolás de Hidalgo, Morelia 58000, Michoacán, Mexico; lamorales@conacyt.mx

³ Departamento de Ciencias Computacionales, CONAHCYT-Instituto Nacional de Astrofísica, Óptica y Electrónica, Tonantzintla 72840, Puebla, Mexico; algredobadillo@inaoep.mx

* Correspondence: clopezp@up.edu.mx; Tel.: +52-3313682200

Abstract: The sale of second-hand vehicles is a popular trade worldwide, and vehicle fraud is currently a common issue, mainly because buyers can lack a complete view of the historical transactions related to their new acquisition. This work presents a distributed architecture for stakeholders to register transactions over a vehicle's life cycle in a blockchain network. The architecture involves a non-fungible token (NFT) linked to a physical motorized vehicle after a tokenization process, which denote as the NFT-Vehicle. The NFT-Vehicle is a hierarchical smart contract designed using an object-oriented paradigm and a modified version of the ERC721 standard. Every stakeholder engages with the NFT-Vehicle through distinct methods embedded within a smart contract. These methods represent internal protocols meticulously formulated and validated based on a finite-state machine (FSM) model. We implemented our design as a proof of concept using a platform based on Ethereum and a smart contract in the Solidity programming language. We carried out two types of proof: (a) validations, following the FSM model to ensure that the smart contract remained in a consistent state, and (b) proofs, to achieve certainty regarding the amount of ETH that could be spent in the life cycle of a vehicle. The results of the tests showed that the total transaction cost for each car throughout its life cycle did not represent an excessive cost considering the advantages that the system could offer to prevent fraud.

Keywords: blockchain; vehicles; smart contract; smart property

MSC: 68-04



Citation: López-Pimentel, J.C.; Morales-Rosales, L.A.; Algreto-Badillo, I.; Del-Valle-Soto, C. NFT-Vehicle: A Blockchain-Based Tokenization Architecture to Register Transactions over a Vehicle's Life Cycle. *Mathematics* **2023**, *11*, 2801. <https://doi.org/10.3390/math11132801>

Academic Editor: Jan Lansky

Received: 15 April 2023

Revised: 16 June 2023

Accepted: 17 June 2023

Published: 21 June 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Vehicles are evidently important in our daily lives. They provide the opportunity to travel long distances; in some cases, having a car might be a source of inspiration or confer status. Sometimes, a vehicle is the most valuable possession a person has.

The sale of second-hand vehicles is a common trade around the world. Customers' lack of financial resources to buy new cars is one of the reasons for the growing volume of second-hand car sales, complemented by the investments made by industry participants to establish their dealership networks in the market [1]. Unfortunately, the vehicle market is highly fraud-prone.

If fraud is committed in the ownership exchange of a vehicle, or the vehicle is legally compromised, the new owner could be affected and lose a large fraction of their investment. Second-hand vehicle fraud is a common issue [2]. For example, buyers could lack a complete view of the life cycle of their new acquisition. Although a buyer could consult different stakeholders' logbooks, one of them could easily be omitted, or some registers

could not be readily available; hence, all of these registers being isolated causes serious problems for buyers when trying to review them or establish certainty about the real value of the asset. Additionally, some tricky sellers are able to change a car's mileage and fraudulently generate another ledger. On the other hand, fraud is also committed by falsified invoices, since the commercial invoice is still, in most countries, the traditional mechanism for linking the ownership of goods.

Based on the above, a question arises: how can we generate more trust in the historical transactions related to a used car and avoid fraud?

These processes can be secured and improved with the help of smart property technologies. Smart property implements smart contracts by tokenizing the physical good using blockchain technology. A smart contract is a computer program intended to digitally facilitate, verify, or enforce the negotiation and enactment of a contract using transactions. These transactions are located within a blockchain network. Blockchain is a technology that provides greater trust in the digital world [3]. Blockchain uses a distributed ledger, wherein the transactions and data are recorded identically in multiple locations, providing complete transparency. All participants with authorized access see the same information. All transactions are recorded with immutability and are date- and time-stamped, allowing one to view the entire history of a transaction (traceability) and virtually eliminate any opportunity for fraud.

We discovered previous studies incorporating blockchain that primarily focused on addressing the problem of odometer fraud, but they did not incorporate blockchain-based transactions involving different parties [4–6]. Conversely, certain studies have considered multiple stakeholders in transactions throughout a vehicle's life cycle [7–13]; however, none of these integrated the concept of a legal owner, which we introduce herein. Furthermore, when it comes to tokenizing physical cars into NFTs, there remains a scarcity of studies [14,15]. We noted that none of the existing works offered comprehensive information regarding the underlying smart contract based on the ERC721 standard for tokenizing physical vehicles.

In particular, we were interested in using NFTs linked to physical vehicles. Vehicles have considerable importance in the research community, with the Internet of Vehicles (IoV) area aimed at making them smarter and more digitized. This paper presents a secure distributed architecture based on the smart property idea to register transactions over a vehicle's life cycle in a blockchain. Our architecture consists of the tokenization of a physical vehicle through a smart contract, which we call the NFT-Vehicle. Different actors, such as manufacturer, owner, government, legal owner, helper, and seeker, can interact with the NFT-Vehicle to execute transactions and generate value across the life cycle of the physical vehicle. These interactions are represented using a finite-state machine (FSM), which is applied in the development phase to validate the smart contract created in the Solidity programming language. The NFT-Vehicle assumes the standard ERC721 interface, which we modified to be more suitable for smart property. Furthermore, we implemented a proof of concept, wherein a vehicle-wallet and stakeholder interfaces established communication with the blockchain network to execute transactions. We also generated a set of proofs and transaction costs that showed the amount of ETH that could be spent over the life cycle of a vehicle.

The contributions of this research can be summarized as follows:

- A secure distributed architecture, called the NFT-Vehicle, was designed based on the smart property idea to register transactions over a vehicle's life cycle in a blockchain, involving stakeholders such as the manufacturer, owner, government, legal owner, buyer, insurance provider, and maintenance provider, who interact directly using a user interface and indirectly through a smart contract representing the physical car.
- A smart contract hierarchy was developed based on the ERC721 standard, detailing the role of each stakeholder and how they interact with the NFT-Vehicle through different internal protocols.

- We provided transparency regarding the estimated ETH cost of the NFT-Vehicle throughout the vehicle's life cycle using the Ethereum platform.

The rest of the paper is organized as follows: first, Section 2 explains the technologies employed in this paper; Section 3 describes the problem and presents a discussion, addressing some related works; Section 4 explains the problem and the general architecture proposal; Sections 5–7 describe the architecture in detail; Section 8 presents the proof of concept; Section 9 analyzes both the architecture and the proof of concept and addresses some of the limitations of our architecture; and, finally, our conclusions are presented in Section 10.

2. Background

Nowadays, people stay more connected to internet services in different digital technology sectors. This was caused mainly by the restrictions imposed by the COVID-19 pandemic, forcing organizations to adopt more digital technology in their operational models [16].

Digitization allows organizations to speed up their operations for economy, practicability, and even environmental reasons. However, digitization is not simple, especially when issues regarding trust in computer security are at play. Computer security has been increased with the introduction of blockchain technology into cryptocurrency applications [17], allowing the evolution of money systems in some countries towards a cashless economy [18]. We will describe this technology in more detail in the following subsection.

2.1. Blockchain

Blockchain is a distributed platform technology introduced by Nakamoto [17]. It can be summarized as a chain of blocks linked together, wherein each new block holds the information of new transactions added to the platform and also recognizes the previous block. By following the references of the blocks, starting with the newest, it is possible to track down the whole recorded transactional history up to the first block, called the *genesis block*. Theoretically, information can never be deleted or modified, only added [19].

Blockchain combines hash algorithms, asymmetric cryptography, time stamps, consensus algorithms, and other technologies. Blockchain does not depend on a trusted central server to ensure the security of the system ledger; instead, the network nodes (called miners) validate each other by consensus. Using cryptographic protocols, the network can ensure that a block can only be modified with the previous agreement of the network members. Hence, multiple copies of the blockchain exist among its members. Blockchain features the following characteristics: decentralization, trustfulness, collective maintenance, a reliable database, openness, security and tamper-proofing, anonymity, programmability, verifiability, and traceability [20]. Every one of these features is reason enough by itself to choose blockchain over traditional database solutions.

Blockchain is a technology that has grown considerably and encouraged the development of many platforms, which we will explain in the following subsection.

2.2. Blockchain Platforms

There are two predominant blockchain types: public and private [21]. A public blockchain, also known as non-permissioned, allows any participant to create and validate blocks and modify the block state by storing and updating data through transactions among participating entities; the stored information is transparent and accessible to everyone. Private blockchain, also known as permissioned, is restricted because only authorized and trusted entities can participate in the activities within the blocks.

Blockchain was initially developed to support Bitcoin [17]; this cryptocurrency is based on a public blockchain. Bitcoin enables anyone to transfer digital currencies from one entity to another via transactions. Inspired by this innovation, other cryptocurrencies have been developed; for example, Ethereum, Solana, Cardano, Polkadot, Bitcoin Cash, Litecoin, Dogecoin, and EOS. Despite this, Bitcoin remains the market leader in this domain.

Ethereum [22], together with Bitcoin, is one of the more successful cryptocurrencies; however, Ethereum's embracing of smart contracts caused a large number of blockchain platforms to grow steadily, including Multichain, Hyperledger, Iota, Corda, and Waltonchain [21]. With the growth of these platforms, blockchain has become prominent and gained popularity across a wide range of industry applications, for example, supply chains, healthcare, education, finance, the Internet of Things, digital rights management, insurance, transport, and governance [23].

2.3. Smart Contracts

With the advent of blockchain, smart contracts have become one of the most sought-after technologies [24,25]. Blockchain technology can potentially enlarge a transaction space through smart contracts. A transaction is an exchange of goods, services, or funds involving two parties that reciprocally affect or influence each other [26]. Transactions involve one or more atomic operations that can be executed by any user as long as they have adequate permissions.

The smart contract concept was introduced by Nick Szabo [27]; it refers to a set of computer instructions stored on a blockchain network that resides at a specific address and runs when predetermined conditions are met. These code instructions allow for a series of conditions to be met by the sender and receiver of the transaction before it succeeds. The whole process is completely automated without external help, requiring only the participation of the interested parties and the blockchain network. It is used to automate the execution of an agreement so that all participants can be immediately sure of the outcome without any intermediary's involvement or time loss.

Currently, smart contracts are the key component of the blockchain and have expanded the scope of blockchain technology beyond cryptocurrencies, making them applicable for a variety of applications, such as healthcare, IoT, supply chains, digital identity, business process management, insurance, financial systems, and real estate systems [23,28]. In the digital world, NFTs and smart property are very prominent applications, which we will explore further below.

2.4. NFTs and Smart Property

During the digitization era, some companies took advantage of the moment and used digital transformation to market goods of the future on the Internet thanks to the blockchain technology, such as museum images and digitized collections, using non-fungible tokens (NFTs) and the implementation of smart contracts. In 2021, these goods reached the mainstream trade [29].

Non-fungible tokens (NFTs) have become one of the more notable successful applications of blockchain technology [30]. Furthermore, NFTs are becoming the first application of blockchain technology to achieve clear public prominence [31]. NFTs are digital cryptographic assets recorded in a blockchain network, e.g., Ethereum. An NFT represents something unique and is, therefore, not mutually interchangeable. This differentiates them from cryptocurrencies such as Bitcoin and from many network or utility tokens that are fungible. NFT characteristics include: uniqueness, indivisibility, and transferability. The ownership of these assets is recorded in smart contracts with a unique identification code and metadata that distinguish them from each other.

NFT products can be organized into three main categories: art and collectibles, games and metaverses, and utilities and DeFi [32]. Examples are artwork collectibles, event tickets, music and media, games, virtual items, real-world assets, identities, memes, domain names, and properties.

Smart property is property whose ownership is controlled using smart contracts. Examples include physical property, such as vehicles, which we considered in this work. One of the reasons for proposing smart property as a solution in this paper is that users can remove the trust component of a transaction, making it a secure way to exchange the ownership of property between strangers using smart contracts.

3. Related Work

This section begins by highlighting the problem of the second-hand car market, describing how malicious people might take advantage of this problem and commit fraud. Then, we outline some related works, identifying the issues that led to the introduction of our proposal.

3.1. *The Lack of Trust*

One important issue that impacts the value of second-hand cars is the lack of trust. If a consumer wants to buy a used vehicle, he/she may have serious doubts regarding the authenticity of the displayed mileage and maintenance history and whether the vehicle has been wrecked or salvaged.

Mileage is an important data category that usually impacts the value of second-hand cars. Despite the introduction of sophisticated digital odometers within modern vehicles, mileage fraud continues to be effected by dedicated criminals. Countries around the world have already expressed concern with this phenomenon, including in Europe [33–36]; the United States [37]; and China [38,39].

The lack of trust in vehicles' maintenance histories is becoming a growing problem internationally. Although buyers could consult different logbooks, they could easily omit one, or some registers could not be readily available; hence, buyers could lack a complete view of the life cycle of their new acquisition. Vehicle stakeholders might keep a logbook of details relevant to them, registering activities throughout the vehicle's life cycle; owners can keep records either physically (there usually exists a vehicle logbook wherein the maintenance services are registered) or digitally; cars contain an internal computer register; the government usually has a database containing legal logs of cars; and maintenance agencies and insurers also keep relevant records. With all these registers being isolated, it is challenging for buyers to review them or increase their certainty regarding an asset's actual value.

Another example of vehicle fraud pertains to vehicles declared as wrecks by an insurance expert and thus no longer allowed to use the roads.

Nevertheless, with deceptive support, some damaged vehicles are returned to the second-hand car market. These unqualified vehicles could cause severe accidents, injuring people and incurring considerable costs for insurance companies and consumers.

Ingrid Bauer et al. [40] carried out an exploratory study, including both quantitative and qualitative measures, and showed the possibility of a market wherein the participants (buyers and sellers) appreciate trusted car data, and that this market must increase transparency; they concluded that when mature, blockchain-based applications would provide the core values of this market.

3.2. *Dealing with the Odometer Fraud Problem*

Chanson et al. [4] focused on the application of blockchain to vehicular odometers. They presented a system wherein a dongle retrieves the odometer values and the VIN of the car via the onboard diagnostics II (OBD-II) interface and sends the data via Bluetooth to a laptop in the car; then, the car sends the data into Ethereum network using Web3 with the Node.js application.

Another study addressing odometer fraud was presented by Lucas et al. [5]. They proposed an API system with two parts: data insertion through HTTP requests from a simulated vehicle arriving at the blockchain, and data visualization using a web application. They based their design on the proof-of-work algorithm.

Cyril et al. [6] also focused on solving the odometer fraud problem; they proposed an architecture whereby vehicles send odometer data to a consortium blockchain; this architecture was designed with two approaches, wallet- and non-wallet-based.

These studies addressed the odometer problem but did not model different stakeholder roles, since they simulated the vehicles interacting directly with the blockchain.

3.3. Vehicle Transactions Ledger including Stakeholders

Brousriche et al. [7] proposed using hybrid storage, classic databases, and blockchain technology to manage transactions over a vehicle's life cycle and provide more transparency and collaboration between the involved stakeholders. They presented a digitization of a vehicle's life cycle and proposed the creation of a digital and certified maintenance book using a Quorum-based blockchain. Their proposal included automatic mileage registration, a car sale process, and the presence of an insurance provider. The published research lacked technical details about the protocol of registering the mileage in the cloud service and how it was matched in the blockchain. These authors extended their work in [8], proposing a hybrid cryptographic protocol to enable the sharing of private data between stakeholders. However, the smart contract details were not provided.

Masoud et al. [41] proposed a system framework for used motor vehicles that implemented the blockchain concept. The framework included several stakeholders, such as owners, repairing companies, and insurance agencies, who could register and add transactions for cars. They used a combination of off-chain and blockchain transactions. A traditional database was used to cache intermediate data, and the blockchain used four smart contracts to manage the transactions: registration, organization registration, history reports, and contract updates. Carchain was not based on a standard known blockchain, such as Ethereum or Hyperledger.

Mehmet et al. [9] presented a tamper-free ledger of events as an insurance record for motor vehicles. They proposed the use of blockchain to capture the history of vehicle details in order to generate value for the insurance participant; different actors (such as individual drivers, insurance companies, and government agencies) collaborated to build the records in the ledger. Their solution was designed to be implemented in Hyperledger technology; the details and proofs were not shared.

Sharma et al. [10] proposed a blockchain-based framework for the automotive industry focused on smart cities. They proposed an automotive life cycle categorized into seven phases: government regulator, manufacturer, dealer, leasing company, user, maintenance, and recycling. Unlike our framework, these authors propose that the government regulator should create the new vehicle and be responsible for loading it into the ledger; in our proposal, we argue that the manufacturer is responsible for loading the vehicle into the ledger and must specify the government. Sharma et al.'s model included various additional roles, such as dealer and leasing company; in our case, these roles can be added during the purchase process by adding new owners to the ledger. Their experimental analysis used the Ethereum platform, Node.js, PHPStrom, and Truffle.

Wang et al. [11] presented a blockchain-based product service system framework. The framework included five components: data, stakeholders, blockchain, connections, and applications. The stakeholders interacted with the blockchain using applications that established connections to store data. The published research did not provide technical details about the blockchain, smart contracts, and transaction types.

Syed et al. [12] offered a complete overview of the vehicle life cycle and a blockchain-based solution. Their proposal included four modules: (1) For new and used vehicles, changing ownership details after buying and selling transactions; (2) For regular maintenance, including the checking and renewal of road registration, violation management, and accident management; (3) For the prediction of used vehicle prices; (4) For scrapping a vehicle. They also included various roles: drivers, owners, insurance companies, and mechanics. They used Hyperledger Fabric, which is a permissioned blockchain platform. A significant difference compared to our work was the prediction of used vehicle prices, which they calculated based on two important factors: mileage and the vehicle's original color. The prediction was carried out off-chain using machine learning technology and some manual configuration to determine the value by considering the original color. The accuracy decreased slightly by altering these configurations manually. However, this framework was only a proposal.

Jiang and Sun [13] proposed a model for the second-hand vehicle market in Taiwan wherein vehicle transactions were stored in an Ethereum blockchain platform. Their model included different roles that could execute transactions at different moments in the life cycle of a vehicle: manufacturer, maintenance plant, government branch, and customer. They used Go-Ethereum for the blockchain network, the Solidity programming language, Node.js, and Web3.js in the back-end to connect with the blockchain and a web page for the user–client interface. The smart contract was not published.

While some works have involved multiple stakeholders in the transactions over a vehicle’s life cycle, none have incorporated the crucial notion of a legal owner. The legal owner represents the role vested with ownership rights following a formal procedure involving the government.

3.4. Tokenizing Physical Cars

NFTs, as a recent technology, have begun to be applied in diverse fields, including energy [42]; medicine [43]; and administration (e.g., ticket generation [44] and ticket sales [45]). Regarding the transport sector, the authors of [45] analyzed different existing applications and aimed to discover potential usage areas for NFTs in this sector, showing that they can be used for train and bus ticket sales or ride-sharing platforms when passengers are traveling in the same direction at the same time. However, when it comes to tokenizing physical cars using NFTs, there remains a scarcity of research. Nevertheless, we describe some relevant studies below.

Dominic Pirker et al. [14] presented a shared mobility platform wherein cars were tokenized based on the ERC721 standard. They added a hardware module to store the credential keys and used it as a wallet, in which the token was also stored; this module was accessed via Bluetooth using a smartphone application. This work applied the same standard as our proposal; however, they did not implement the concept of various stakeholder roles.

It comes as no surprise that major players in the automotive industry have been captivated by the potential of NFTs. Global brands such as Alfa Romeo, Porsche, Lamborghini, Ferrari, Mercedes-Benz, Rolls-Royce, Audi, and Nissan actively engage in experimentation, exploring various avenues to incorporate NFT technology into their product lines. These experimentations were summarized in Vitelaru et al.’s study [15]. This work also proposed a car ownership framework based on the ERC-1155 token standard. Furthermore, this study concentrated on dividing vehicle ownership and examining the viability of distributing revenue among the owners based on the percentage invested in acquiring the vehicle. Although the objective of this work differed from what we propose, the idea of dividing vehicle ownership sounds interesting; however, this work did not implement the concept of various stakeholder roles either.

In summary, our analysis revealed that existing studies have primarily addressed the issue of odometer fraud without incorporating blockchain-based transactions involving various stakeholders. Furthermore, while some previous works have involved multiple stakeholders in the transactions over a vehicle’s life cycle, none have incorporated the crucial notion of a legal owner. We also noted that none of the existing works have provided comprehensive details about the underlying smart contract using the ERC721 standard.

4. The Architecture of the NFT-Vehicle

In the following, we describe our general proposal architecture using the smart property concept taking into account the research gaps described in the Related Work section. We assess the virtual odometer variable and consider different stakeholders involved in the transactions related to a car from its insertion into the market until its revocation. We start by providing a general overview of the life cycle of a vehicle. Then, we describe the stakeholder roles and their main tasks, ending the section by outlining the main characteristics of our proposal.

4.1. The Transactions throughout a Vehicle's Life Cycle

Figure 1 shows an overview of our architecture. A vehicle is a physical asset represented by an NFT with a user owner. Each role has a specific interaction (i_n) with another role through the vehicle's NFT, functioning as a smart contract. The architecture involves the following roles: manufacturer, government, owner, legal owner, buyer, helper, and seeker. In the figure, the smart contract is illustrated as a rectangular document that changes for each interaction with a stakeholder, as shown by the documents in the background, denoting the transaction chain. The figure illustrates the start of the process instigated by the manufacturer, who creates the smart contract; this involves linking the physical car with the NFT-Vehicle. Then, the different stakeholders modify the NFT throughout the vehicle's life cycle, which will be explained in the following subsection. The figure includes numbered blue arrows pinpointing the attribute or method that is modified due to a previous interaction.

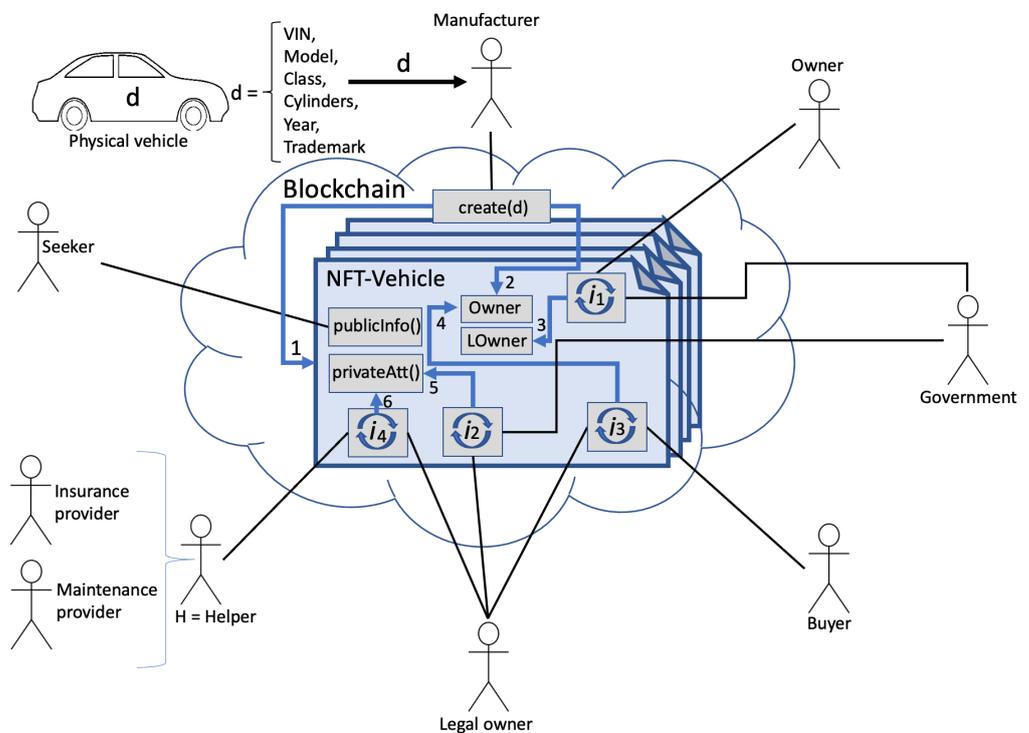


Figure 1. General architecture: recording the life cycle of a vehicle.

4.2. Roles

Following Figure 1, the roles and their interaction with the smart contract are explained as follows:

- Manufacturer.** This role is capable of creating the NFT ($create(d)$, see blue arrow 1). This is the initial process of introducing a new vehicle into the blockchain network. The manufacturer adds the vehicle's genesis information d , becoming the first owner (see blue arrow 2). Let d be the genesis information, which denotes the essentials of the car, for example, the NIV, model, class, cylinders, year, and trademark; these data never change throughout later transactions.
- Owner.** Every vehicle has one and only one owner at a given time. When the NFT-Vehicle is created, the manufacturer becomes the first owner (see blue arrow 2); when a vehicle is sold, the buyer is the new owner after making the purchase (see blue arrow 4). However, an owner must interact with the government (i_1) and request the owner's rights to become a legal owner (see blue arrow 3).
- Government.** This role can change the legal status of a vehicle when certain legal situations are encountered (see blue arrow 5). Each vehicle has a different legal status:

stolen, arrested, penalty, owner rights, or plate change. Usually, the government establishes a transaction cost to execute any of these processes.

- **Legal owner.** This role is acquired when an owner requests such rights from the government, which executes the respective transaction protocol (see blue arrow 3). A legal owner (LOwner in the figure) is the only stakeholder with permission to sell the car (i_3) and request that the government change its legal status (i_2). In addition, he/she can request and authorize a helper to add information about the car (i_4).
- **Helper.** This stakeholder receives temporary permission from the legal owner to add one transaction to the smart contract (i_4). The helper role is used for transactions that change one or more detail of the NFT-Vehicle's properties (see blue arrow 6). This includes official services, mechanical adjustments, significant aesthetic modifications, and insurance payments.
- **Seeker.** A seeker can obtain free or paid public information about the vehicle at any time. Free public information includes, for instance, genesis attributes, debt with the government, arrest, service attributes, legal status, owner, legal owner, manufacturer, and stolen status. Payed public information includes historical information about *arrests, penalties, owner rights, and plate changes*, among other things.

We considered a complete environment wherein the transaction operations are carried out using the cryptocurrency of the Ethereum platform.

4.3. Characteristics of the Architecture

The proposed architecture was focused on providing the following characteristics:

- a Provenance: any user can know the genesis information of the vehicle, the manufacturer, and all initial token information.
- b Transparency regarding the purchase procedure and all transactions concerning the vehicle.
- c The traceability of historical transactions (including ownership, legal status, and updates related to the car's value).
- d Inheritance: as it is implemented on a blockchain network, the architecture inherits blockchain characteristics such as decentralization, trustfulness, reliability, security, tamper-proofing, and verifiability.

The next sections will explain in detail the mechanism of each of the roles.

5. Smart Property Hierarchy

Many stakeholders execute transactions related to vehicles on a daily basis. The government performs vehicle transactions as part of legal management. Vehicular transactions also include manufacturers requesting ownership rights and owners or users who sell, buy, change ownership, pay insurance, perform maintenance services, assign insurance observations, and report vehicles as stolen.

An NFT, abbreviated as a token, is a representation of something in the blockchain context. This 'something', in our case, was a virtual vehicle. By representing things as tokens, smart contracts can interact with them using attributes and methods. Sending tokens between users at a high level involves implementing a smart contract method, and, internally (using logic code), the attribute is transferred to another user, who becomes the new owner of the token.

Our design focused on an NFT-Vehicle smart property linked to a smart contract hierarchy, which contains different methods for stakeholders to interact and perform various transactions. Figure 2 illustrates the complete class diagram hierarchy of the NFT-vehicle. We illustrated the smart contract following the class diagram of the object-oriented paradigm. The following subsection describes the general notation; then, we explain the smart contract of the NFT-Vehicle.

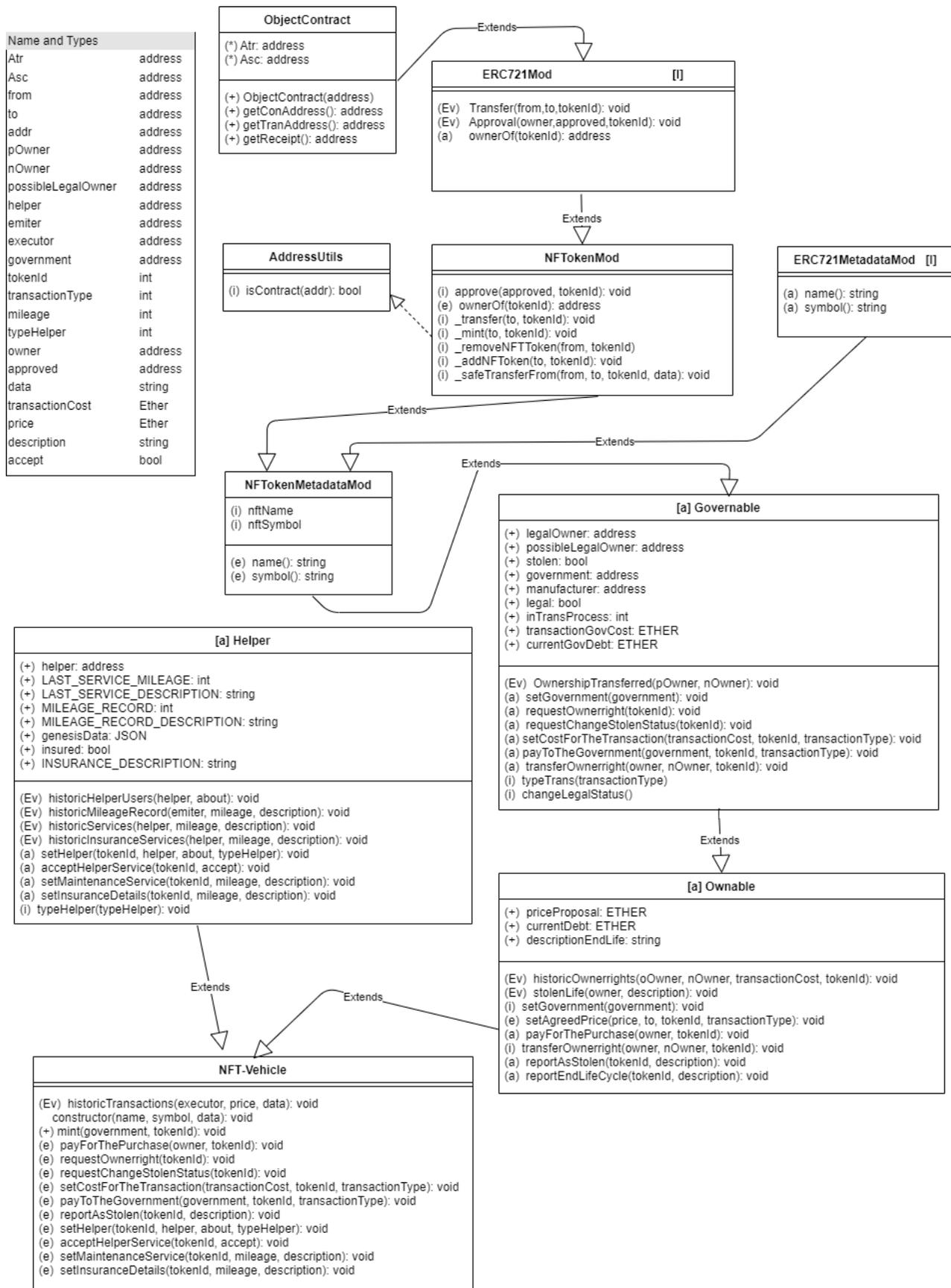


Figure 2. Class diagram of the NFT-Vehicle model.

5.1. Notation

Figure 2 illustrates the NFT-Vehicle model as a class diagram. To represent the smart contracts hierarchy, we used notation similar to that employed for classes in the object-oriented paradigm [46], with some differences, as explained below. Each smart contract has a constructor, a function with the same name as the smart contract that is used to create the contract and might include parameters. The smart contract is illustrated as a rectangle divided into three parts: the name at the top, the attributes, and the methods. The name might be marked as an abstract [a] or interface [i]. Attributes and methods include: abstract (a), public (+), private (−), external (e), and internal (i). Attributes marked with (*) are values auto-generated within the smart contract. Methods can also include events (Ev), which are inheritable contract members; they store the arguments passed down in the transaction logs when emitted. Abstract methods are template methods that are not implemented in the contract. If a smart contract includes at least one abstract method, it is also considered abstract and can only be deployed when abstract methods are implemented. When all methods are abstract, the contract is called an interface. Private methods can only be implemented within the contract if public methods are still accessible from other contracts. External methods can be implemented from other contracts and via transactions but cannot be implemented internally. Internal methods can only be accessed internally from within the current contract or contracts deriving from it (inheritance).

The figure also illustrates some relationships between contracts; for example, inheritance ('extends' arrow) and dependency (dotted arrow).

The rectangle in the upper left part of the figure presents a list of attribute names and types. Here, *address* denotes the cryptographic public address; *int*, *byte*, *bool*, and *string* are commonly known primitive data types; and *Ether*, the money used in Ethereum, can be treated as integer data.

5.2. The NFT-Vehicle Model

5.2.1. Father Contract

Following the idea introduced in [47] and developed in [48] regarding the base model of all contracts, *ObjectContract* is similar to the object class in the Java programming language. At the top of Figure 2, the object contract denotes the parent of all contracts. It contains two attributes: the contract address *Atr* and transaction address *Asc*. When a contract is created, such attributes are inherited and can be accessed publicly by methods *getConAddress()* and *getTranAddress()*, respectively. Method *getReceipt()* is used to obtain the receipt of a transaction. The contract address identifies the contract in the blockchain, and the transaction address determines the transaction.

5.2.2. NFT Support

The community has developed a variety of standards [49], including the ERC 721 standard, which is a standard interface for non-fungible tokens [50]. This standard provides basic functionality to track and transfer NFTs compatible with physical property. In Figure 2, we present a modified version of this standard called ERC721Mod; it includes two events and an abstract method to pinpoint the contract owner. The ERC 721 standard includes more public abstract methods, but in our model, these were modified to be internal and implemented in *NFTTokenMod*, as can be seen in the figure.

NFTTokenMod is an implementation of ERC721Mod; it implements a utility called *AddressUtils*, which contains a function for indicating whether an address is a contract and is available at <https://github.com/nibbstack/erc721/blob/master/src/contracts/utills/address-utills.sol> (accessed on 7 April 2023)). With this contract, it is possible to execute several operations involving the token, such as minting, transferral, and removal. The method *ownerOf(tokenId)* is the only one that can be accessed externally; the rest are internal, which means that they are accessed from inherited contracts.

NFTTokenMetaDataMod is an extension of *NFTTokenMod* and an implementation of the interface *ERC721MetaDataMod*, which is used to include attributes' names and symbols.

5.3. NFT-Vehicle Contract Implementation

In Figure 2, **Governable** is an abstract contract, and most of its methods are abstract. This contract includes the government’s role; the abstract methods are aimed at determining the government address and the transactions pertaining to legal procedures such as ownership transferral, requesting owner rights, changing a stolen status, paying the government, and setting the cost for transactions with the government. This contract also includes a list of public attributes that can be consulted publicly. These attributes are related to the changes that the above methods can implement.

Ownable is an abstract contract containing several abstract methods and several methods implemented from the **Governable** contract. This contract implements methods related to the purchase and ownership rights transferral processes. Reporting a vehicle as stolen is an abstract method implemented in the **NFT-Vehicle**.

Helper is an abstract contract whose methods are mostly abstract and are implemented in the **NFT-Vehicle**. These methods are related to assigning maintenance service roles and the insurance provider role. Some attributes are related to the mileage, the description of mileage changes, the genesis data, and the description of insurance.

Finally, the **NFT-Vehicle** is the smart contract that implements all the abstract methods of its father contracts; it is a deployed contract. This contract inherits all public, external, and internal attributes, but only the public and external ones are visible. The NFT-Vehicle’s methods are explained in detail in the following sections.

6. Finite-State Machine of the NFT-Vehicle

Internally, the NFT-Vehicle comprises a collection of attributes and methods that can be accessed and modified by various user types throughout a vehicle’s life cycle, from its creation until its eventual end. During the design phase, we developed a finite-state machine (FSM) model to prevent any inconsistencies in the smart contract. Figure 3 illustrates the FSM of the NFT-Vehicle.

An FSM is a computation model widely used in modeling application behavior, the design of digital hardware systems, software engineering, compilers, network protocols, and computational linguistics. An FSM is an abstract machine that can be in exactly one of a finite number of states at any given time. It is defined by a list of states, starting with its initial state, and it can change from one state to another in response to certain inputs; the change from one state to another is called a transition [51].

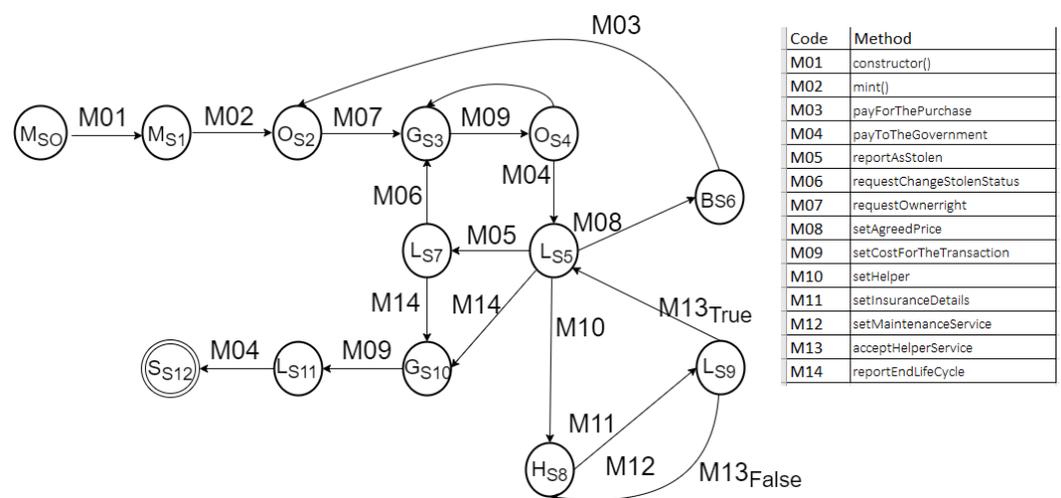


Figure 3. Finite-state machine of the NFT-Vehicle.

In Figure 3, within the circle R_{sn} , R denotes the stakeholder in the state sn that is permitted to execute a method M . Transitions are denoted with an arrow that connects one state to another after executing method MN , where N represents the method’s number.

The inputs are the execution of a method with its corresponding parameter (these methods were shown in Figure 2). The end state is depicted with a double circle. The right-hand side of the figure shows all the methods of the NFT-Vehicle contract. The FSM of the figure illustrates all the correct states, not considering malicious or mistaken executions. In practice, the methods can be implemented by any stakeholder at any moment from state s_0 to s_{12} . However, a malicious execution (or a method without permission) leads to an error state; although this is not displayed in the figure, it must be assumed.

6.1. NFT Generation

The manufacturer constructs vehicles with specific characteristics, for example, vehicle identification number (*id*), *trademark*, *model*, *class*, *version*, and number of *cylinders*. These attributes form the initial description of a vehicle and cannot be changed over time. The following shows an example in the JSON format.

```
d = {
  id:      "1FMYU02Z97KA580G2",
  tradeMark: "abcd",
  model:   "2012",
  class:   "auto",
  version: "TA XLS 4X2 I4 TELA 4 CIL",
  color:   "White",
  cylinders: "L4"
}  :: others
```

Let a manufacturer generate a token with genesis information d , as exemplified in the JSON format; the manufacturer in state M_{s_0} (Figure 3) creates the token t through the constructor ($M01$); with this, the vehicle is tokenized. The builder of t is the manufacturer, and nobody can execute transactions in the smart contract (state M_{s_1}) except the seeker, who can consult the public values.

6.2. Minting

Once the token has been created, the manufacturer must mint it ($M02$), which means establishing some conditions for its commercialization, involving the following:

- Identifying the government;
- Including an identifier for the token;
- Establishing the manufacturer as the first likely legal owner.

Once minted, arriving at state O_{s_2} , the owner of t is the manufacturer. However, to establish the first legal owner, someone must request the ownership rights.

6.3. Requesting the Ownership Rights

To market the token, the owner must carry out two main steps: (a) Requesting the ownership rights from the government ($M07$ in Figure 3), as will be detailed in Section 7.2; and (b) executing the transaction protocol with the government ($M09$ and $M04$, Figure 3), as will be detailed in Section 7.1. After executing these steps, the owner becomes the legal owner L_{s_5} .

6.4. Legal Owner

According to Figure 3, a legal owner in state L_{s_5} can sell a vehicle ($M08$), report it as stolen ($M05$), assign a helper ($M10$), or report the end a vehicle's life ($M14$).

6.4.1. Purchase

($M08$): The legal owner must establish an initial price;

($M03$): The buyer B_{s_6} must transfer the funds established in the previous step; once the balance is liquidated, the buyer becomes the owner O_{s_2} .

The new owner must repeat the steps described in Section 6.3.

6.4.2. Reporting as Stolen and Requesting Recovered Status

The legal owner in state L_{S5} :

- (M05): The legal owner reports the car as stolen; in this state, the legal owner could also report (M14) the end of the token's life.
- (M06): If the car is found; the legal owner L_{S7} can request that the government modify this status again.

The transaction protocol is executed with the government as described in Section 6.3.

6.4.3. Helper Maintenance Service

In state L_{S5} :

- (M10): The legal owner can assign an external user to modify some properties of the vehicle, for example, when a mechanical provider services the car or an insurance provider becomes involved;
- (M11 or M12): In this case, the helper role H_{S8} includes maintenance services or insurance details for an insurance provider;
- (M13): The legal owner L_{S9} verifies the service (the change) via the smart contract and can accept ($M13_{True}$) or reject it ($M13_{False}$).

If the service is accepted, then the vehicle returns to the original state L_{S5} ; otherwise, it returns to H_{S8} .

6.4.4. End of Life

Ending the life of a vehicle starts from state L_{S5} or L_{S7} :

- (M14): The legal owner reports the car at the end of its life. After this operation, the legal owner cannot execute any operation involving the smart contract. The only stakeholder that can execute an action is the government.
- (M09): Having received the request, the government G_{S10} establishes a transaction cost to officially remove the vehicle;
- (M04): The owner L_{S11} pays the complete balance to the government.

Automatically, no user can execute any operation on the smart contract S_{S12} ; only the seeker can consult public information related to the life cycle of the vehicle.

7. Protocols

We established a set of protocols whereby the stakeholders interact with each other through the NFT-Vehicle. One interaction is executing a payment transaction with the government; for example, an owner, to become a legal owner, must interact with the government and pay for this transaction. Other interactions with the government occur when a legal owner wants to change a stolen status after having reported and recovered a stolen vehicle, when a legal owner executes a purchase procedure with a buyer, and when a legal owner interacts with a maintenance or insurance provider.

Interactions with the NFT-Vehicle result in state changes, which we implemented through various protocols. These protocols aligned with the finite-state machine (FSM) described in Section 6.

7.1. Transaction Payment to the Government

The government can establish the legal status of a vehicle through the NFT-Vehicle. The legal status might change when the vehicle is involved in legal situations such as changing *the owner rights, changing the licence plate, theft, arrest, and penalties*.

In Figure 4, we show a general transaction payment to the government involving three parts: (a) The government must verify certain conditions (depending on the transaction type) and establish the transaction cost; (b) the established transaction cost must be transferred to the government; and (c) once the government has received the payment, it must close the legal procedure. Note that parts (b) and (c) must be executed instantly, one after the other, and automatically.

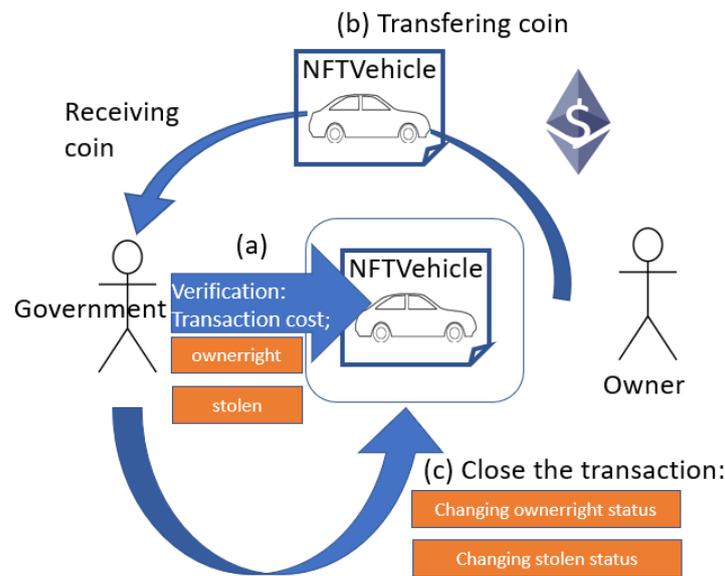


Figure 4. Execution of a transaction payment to the government.

These three parts are illustrated by transitions (1) and (2):

$$G_{S3} \xrightarrow{\text{setCostForTheTransaction}(ether,tokenId,tranType)} O_{S4} \tag{1}$$

The government establishes a transaction cost in *ethers*, a token id (*tokenId*), and a transaction type (*tranType*). The government account must implement this transition, which is executed successfully if the NFT-Vehicle is not occupied by other legal procedures; *currentGovDebt* is a public attribute whereby a seeker can access information regarding debts to the government.

$$O_{S4} \xrightarrow{\text{payToTheGovernment}(government,tokenId,tranType)} L_{S5} \tag{2}$$

In this transition, the owner must specify the government address, the token id, and the transaction type. The owner must execute the transaction, and if the established amount is paid completely, the requested status is applied automatically. If a debt is pending, the owner must execute this method again to complete it; *currentTransDebt* is a public variable that allows one to access information regarding debts to the government. This method involves steps (b) and (c) of Figure 4.

7.2. Acquiring the Owner Rights

Every vehicle must have one and only one owner at a given time. When the manufacturer creates the vehicle, they are the owner. When a vehicle is sold, the buyer who has made the purchase is the new owner. However, an owner might become a legal owner by requesting the owner’s rights from the government and paying the established cost, executing the protocol described in Section 7.1. The transitions are the following:

$$O_{S2} \xrightarrow{\text{requestOwnerright}(tokenId)} G_{S3} \tag{3}$$

This transition is implemented by the owner, who will likely be the new legal owner. The token must be free of any illegal status to execute this procedure successfully. Then, the protocol explained in Section 7.1 is applied. In this case, the government must verify that the owner has requested a change of ownership rights; the token must be free of any illegal status to establish a transaction cost. If the likely legal owner pays the amount entirely,

the ownership rights are obtained automatically, and the vehicle acquires the legal status of *active*.

7.3. Legal Owner Protocols

The legal owner is the only stakeholder authorized to sell a vehicle and ask the government to change the vehicle’s legal status. The legal owner can request and allow a helper to add certain information about a car.

7.3.1. Transferring Ownership

Selling the physical vehicle requires transferring the token, which involves two steps: (i) Transferring the ownership to a new owner, as we will explain in this section; and (ii) acquiring the ownership rights, as explained in Section 7.2.

Figure 5 shows how to transfer ownership. This process involves three parts:

- (a) The legal owner must establish an initially agreed price with the buyer:

$$L_{S5} \xrightarrow{\text{setAgreedPrice}(\text{price}, \text{to}, \text{tokenId}, \text{transType})} B_{S6} \tag{4}$$

The owner establishes a price to transfer the token to the likely new owner *to*.

- (b) The buyer must transfer the amount established in the previous step:

$$B_{S6} \xrightarrow{\text{payForThePurchase}(\text{owner}, \text{tokenId})} O_{S2} \tag{5}$$

This method internally implements the private method:

$$\text{safeTransferFrom}(\text{owner}, \text{to}, \text{tokenId})$$

Once the agreed amount has been received, the owner must transfer the token to the buyer, since the buyer has become the new owner. Note that these operations of sending and receiving money and transferring tokens are executed instantly, consecutively, and automatically within the smart contract.

Additionally, some variables might be consulted publicly by a seeker. Variable *priceProposal* returns the price agreed on by the owner. Variable *currentDebt* returns the debt of the buyer, since the buyer could have previously provided an advance, so this variable can be consulted to ascertain the amount required to acquire the token.

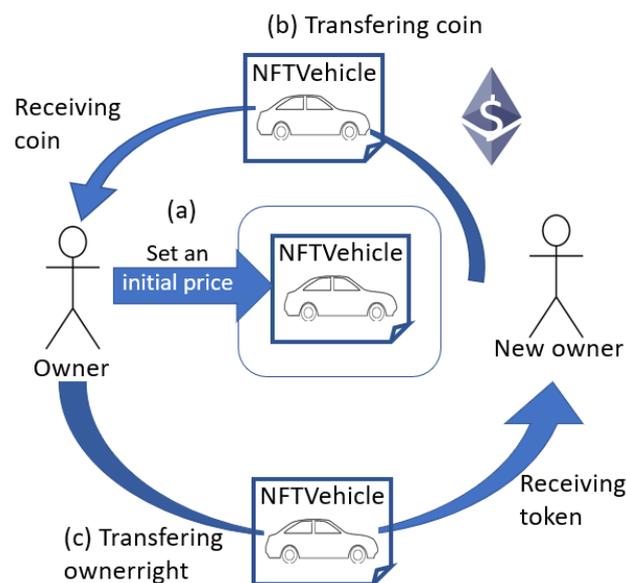


Figure 5. Transferral of ownership.

7.3.2. Reporting as Stolen

To avoid fraud, we implemented an efficient mechanism whereby the legal owner can report a vehicle as stolen. The following transition is executed by the legal owner to perform this task:

$$L_{S5} \xrightarrow{\text{reportAsStolen}(tokenId,description)} L_{S7} \quad (6)$$

Once reported, the following conditions apply:

- The token cannot be reported as stolen again.
- The legal owner cannot claim the ownership rights again.
- The token cannot be sold; hence, the owner cannot execute (4).
- The government can modify this status again, but this would involve a transaction cost for the legal owner.

7.3.3. Changing Stolen Variable Status

The stolen status can only be changed by the government; the process is as follows:

- (a) Firstly, the token must be reported as stolen by the legal owner executing (6);
- (b) The legal owner, with the vehicle in state L_{S7} , requests that the government changes the stolen status by executing:

$$L_{S7} \xrightarrow{\text{requestChangeStolenStatus}(tokenId)} G_{S3} \quad (7)$$

- (c) Finally, the legal owner must interact again with the government using the protocol explained in Section 7.1.

Once the previous steps have been carried out, the stolen status is changed.

7.3.4. Reporting the End of the Vehicle's Useful Life

In the last stage of the useful life of a vehicle, it is taken out of circulation. This process involves the legal owner reporting the vehicle to the government:

$$L_{S7} \xrightarrow{\text{reportEndLifeCycle}(tokenId,description)} G_{S10} \quad (8)$$

Once reported, the following conditions apply:

- The legal owner cannot execute more transactions involving the token (NFT-Vehicle);
- The legal owner can see public attributes, similarly to a *seeker*.
- The government modifies the status to inactive, which involves a transaction cost for the legal owner, following (1) and (2).

7.4. Helper Interaction Protocol

The helper receives temporary permission from the legal owner to add one transaction to the blockchain. Furthermore, this role is used for transactions that change one or more detail of the vehicle's properties. These include official services, mechanical adjustments, significant aesthetic modifications, and insurance payments. The subsections below will explain the helper protocol and provide an example of a maintenance service.

The Helper Protocol

Figure 6 shows the helper protocol, which involves three parts:

- (a) The legal owner must assign a helper and designate the type of helper:

$$L_{S5} \xrightarrow{\text{setHelper}(tokenId,helper,about,typeHelper)} H_{S8} \quad (9)$$

(b) The helper must execute their task in the smart contract:

$$H_{S8} \xrightarrow{\text{setInsuranceDetails}(tokenId,mileage,description)} L_{S9} \tag{10}$$

or

$$H_{S8} \xrightarrow{\text{setMaintenanceService}(tokenId,mileage,description)} L_{S9} \tag{11}$$

Independently of having executed (10) or (11), the helper must set the current mileage of the vehicle and provide a description of the service carried out for the vehicle. This method ensures that the mileage entered by the helper is greater than the previous value, but if the owner disagrees with the introduced data, they can be restored.

(c) Finally, the legal owner must accept or reject the changes. Once the helper has carried out the transaction, executing (10) or (11), the legal owner can verify the modifications and accept them:

$$L_{S9} \xrightarrow{\text{acceptHelperService}(tokenId,True,accept)} L_{S5} \tag{12}$$

If the legal owner accepts the service, then the changes are accepted in the smart contract, and the helper cannot make any more updates. However, if the owner rejects the changes, the following is executed:

$$L_{S9} \xrightarrow{\text{acceptHelperService}(tokenId,False,accept)} H_{S8} \tag{13}$$

Then, the helper must execute the service again and repeat the process.

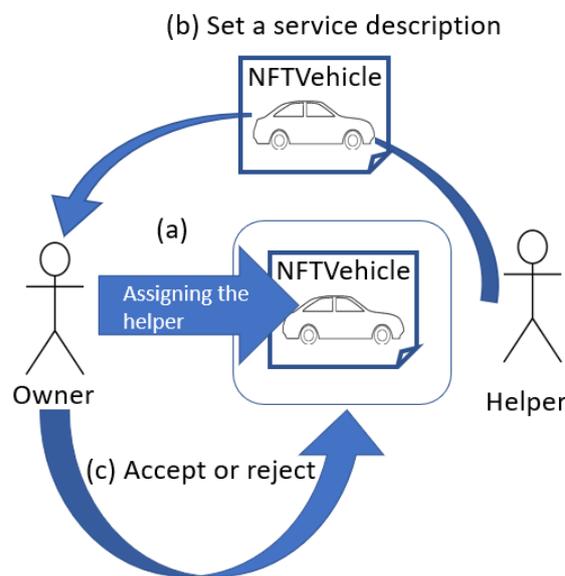


Figure 6. Helper protocol.

8. Proof of Concept

This section presents a proof of concept of the operations that stakeholders can perform with using the NFT-Vehicle. Our proof of concept aimed to provide a clearer picture of the feasibility of our architecture and how it could be replicated or adapted. First, we explain the technologies used to build the NFT-Vehicle wallet and the communication with the blockchain network. Then, as an example, we describe a general system involving various transactions across a vehicle’s life cycle executed by different stakeholders and show how they interacted with the token and the costs generated by the transactions. Finally, we explain how a seeker can ascertain the status of a vehicle at each moment of its life cycle.

8.1. The NFT-Vehicle Wallet and Blockchain Network Communication

The NFT-Vehicle wallet and the communication with the blockchain network at a high level are shown in Figure 7. Although the generation of the token is explained in the following subsections, here, we assumed that the token had already been submitted to the wallet; the explanation is as follows:

- The wallet application communicated with the blockchain network. The wallet interface used was Metamask, which stores different accounts A ; one of these accounts, e.g., a , is used to execute any transaction.
- Wallet network sending: let s be the service method requested to be executed in the smart contract, which was sent to the blockchain network. We used Remix IDE to interact with the methods of the NFT-Vehicle contract.
- Blockchain network receiving: s was received in the blockchain network to execute the service method requested for the smart contract. The software component installed to execute the blockchain was Ganache CLI v6.12.2. The smart contract hierarchy shown in Figure 2 was implemented in the Solidity programming language version 8.0 (available via <https://docs.soliditylang.org/en/v0.8.0/> (accessed on 7 April 2023)).
- Wallet viewing: $f(s)$ was received by the Remix application, and the transaction could be checked with Metamask.

As one can see in the right-hand side of Figure 7, the experiment was executed using the following hardware and operating system infrastructure: (i) processor—11th Gen Intel(R) Core(TM) i7-1165G7 @2.80 GHz 1.69 GHz; (ii) RAM—32 GB; (iii) system type—64 bits, x64-based processor; (iv) operating system—Ubuntu 20.04.

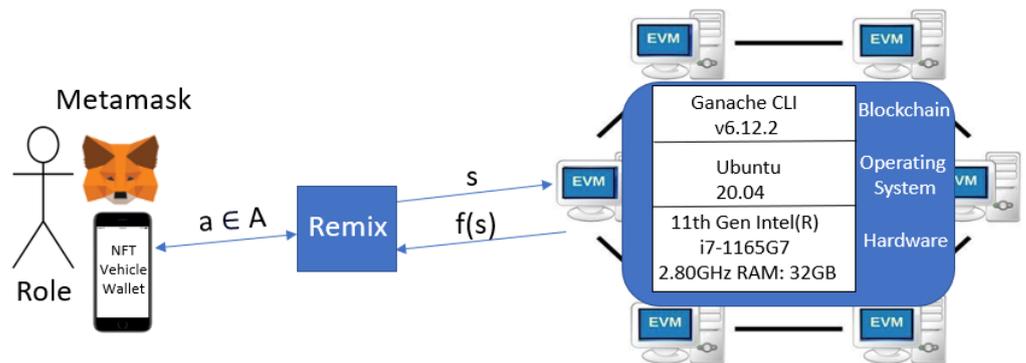


Figure 7. NFT-Vehicle wallet and blockchain network.

8.2. A General Execution

The following data d in the JSON format denote the genesis information of a vehicle that was used in the runs explained below. This particular information had to endure throughout the transactions:

```

d = {
  "name": "GiJo CXL5 Sport Edition",
  "symbol": "GMCX5SE22",
  "idToken": "1FMYU02Z97KA580G2",
  "data": {"ownername": "UPGDL",
    "color": "white",
    "tradeMark": "UP-Vehicle",
    "year": "2022",
    "class": "automatic",
    "version": "TA XLS 4X2 I4 TELA 4 CIL",
    "cylinders": "L4"}
}

```

The following is a list of public address accounts for the stakeholders, denoted as *A*. These accounts are abbreviated as *M*, *G*, *HM*, *HI*, and *B*.

```
A = {
  "Manufacturer" : "0xdbfAc94D966b97eD614c68cb6222a6347214A8EC",
  "Government" : "0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2",
  "HMaintenance" : "0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db",
  "HInsurance" : "0x617F2E2fD72FD9D5503197092aC168c91465E7f2",
  "Buyer" : "0x78731D3Ca6b7E34aC0F824c42a7c18A495cabaB"
}
```

Next, we followed the execution path described in the finite-state machine of the NFT-Vehicle of Section 6 for only one vehicle. If more buyers, manufacturers, or helpers are required, then each role must have its own account.

8.2.1. Tokenization

Figure 8 illustrates all steps of the manufacturer’s role and the interaction between the physical vehicle, the wallet, and the blockchain network to create the NFT-Vehicle. The explanation is as follows:

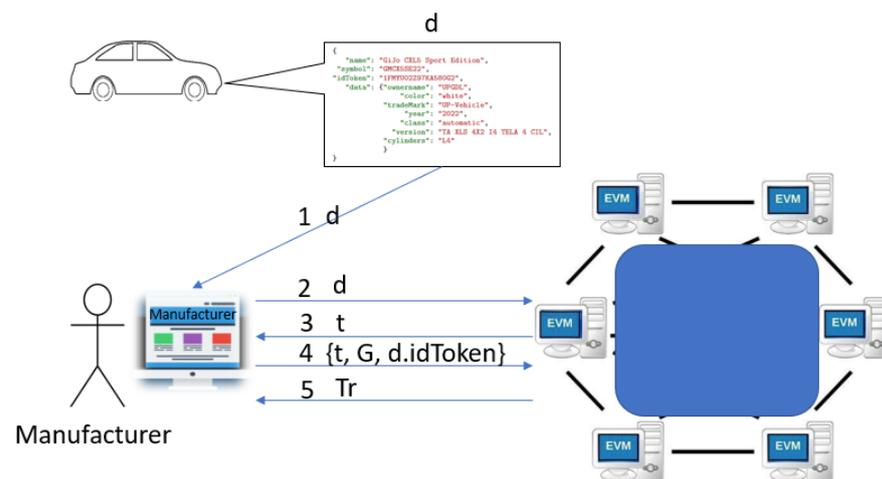


Figure 8. Manufacturer’s role: interaction between the vehicle and the blockchain network.

- Step 1: *d* is obtained from the physical vehicle. This critical initial information is obtained manually.
- Step 2 and 3: The NFT-Vehicle wallet used by the manufacturer applies the communication explained in Section 8.1. In this case, *d* is received in the blockchain network to create the NFT-Vehicle through the constructor of the smart contract; this process generates the token *t*. Finally, *t* is received by the manufacturer application.
- Step 4 and 5: Applying the communication described in Section 8.1 again, the manufacturer mints the token by sending *t*, government (*G*), and *idToken*. As a result, a transaction *Tr* is returned, which includes details about the transaction address, contract address, transaction cost (gas used), block number, the hash of the block, and who executed the transaction. An example is provided below.

```
Tr = {
  "Result": "Success",
  "transactionHash": "0x9386062f12c8dd86536817327569351fcc871d42e25bfba2f19bd5d8eb22a91d",
  "contractAddress": "0x36921a0088Fa4752FC0b441454A086E15a0F6cd1",
  "gasUsed": 3027641,
  "blockNumber": 1,
  "blockHash": "0x62e8fd92fedbba805b89992c6f6f6f2136cc97e329cb7133ed019003a23c019c",
  "from": "0xdbfAc94D966b97eD614c68cb6222a6347214A8EC",
  "fromorigin": "0X2FCBB9CF2910FBA7E7E7A8092AA1A40BC5BA341"
}
```

8.2.2. Transactions

Tables 1–6 show an example of the transactions carried out in the transitions and states illustrated in Figure 3. For all tables, the columns State From and State To indicate the origin and destination state, respectively, whereas the columns Code and Method are the transitions applied by the Executor. The last column represents the transaction cost (gas used), denoted in wei; this refers to the smallest denomination of ether (ETH), the currency used on the Ethereum network. For instance, one ether (ETH) equals 1×10^{18} wei.

Table 1 shows the transactions required by the manufacturer to create the NFT-Vehicle using the JSON data described previously as *d* and to become the legal owner.

Table 2 shows the scenario when a car is sold. In this case, the legal owner establishes a cost for the vehicle. The buyer pays for the car and also pays the government for the rights to become the new legal owner.

Table 1. Manufacturer tokenizing the physical vehicle and becoming the legal owner.

State From	Executor	Transition Code	Transition Method	State To	Transaction Cost
M_{S0}	M	M01	constructor	M_{S1}	3,123,207
M_{S1}	M	M02	mint	O_{S2}	121,141
O_{S2}	O	M07	requestOwnerright	G_{S3}	53,550
G_{S3}	G	M09	setCostForTheTransaction	O_{S4}	75,546
O_{S4}	O	M04	payToTheGovernment	L_{S5}	81,123

Table 2. Purchasing process.

State From	Executor	Transition Code	Transition Method	State To	Transaction Cost
L_{S5}	L	M08	setAgreedPrice	B_{S6}	123,545
B_{S6}	B	M03	payForThePurchase	O_{S2}	101,014
O_{S2}	O	M07	requestOwnerright	G_{S3}	55,550
G_{S3}	G	M09	setCostForTheTransaction	O_{S4}	55,550
O_{S4}	O	M04	payToTheGovernment	L_{S5}	64,013

Table 3 shows the scenario when a car is reported stolen and then recovered. This latter transaction involves payment to the government to change the status of the stolen vehicle.

Table 3. Stolen and recovered transaction cost.

State		Transition		State	Transaction
From	Executor	Code	Method	To	Cost
L_{S5}	L	M05	reportAsStolen	L_{S7}	37,495
L_{S7}	O	M06	requestChangeStolenStatus	G_{S3}	48,267
G_{S3}	G	M09	setCostForTheTransaction	O_{S4}	56,068
O_{S4}	O	M04	payToTheGovernment	L_{S5}	44,797

Table 4 indicates the scenario when a car is delivered for maintenance. First, the legal owner assigns a helper; then, the helper sets up the maintenance service, which the owner agrees to.

Table 4. Helper maintenance transaction cost.

State		Transition		State	Transaction
From	Executor	Code	Method	To	Cost
L_{S5}	L	M10	setHelper	H_{S8}	97,301
H_{S8}	HM	M12	setMaintenanceService	L_{S9}	220,236
L_{S9}	L	M13	acceptHelperService	L_{S5}	57,239

Table 5 shows the scenario when an insurance provider insures a car. The procedure is very similar to the description in Table 4: the legal owner assigns a helper (in this case, the insurance provider); then, the helper sets the details in the contract (in the NFT-Vehicle), which the owner accepts.

Table 5. Insurance helper transaction cost.

State		Transition		State	Transaction
From	Executor	Code	Method	To	Cost
L_{S5}	L	M10	setHelper	H_{S8}	80,131
H_{S8}	HI	M11	setInsuranceDetails	L_{S9}	89,973
L_{S9}	L	M13	acceptHelperService	L_{S5}	70,675

Finally, Table 6 shows how the legal owner reports to the government the end of the vehicle’s useful life so that the government officially retires it. The transaction involves a cost that must be paid to the government.

Table 6. Ending the useful life of a vehicle.

State		Transition		State	Transaction
From	Executor	Code	Method	To	Cost
L_{S5}	L	M14	reportEndLifeCycle	G_{S10}	76,976
G_{S10}	G	M09	setCostForTheTransaction	L_{S11}	33,366
L_{S11}	L	M04	payToTheGovernment	S_{S12}	53,436

8.3. Seeker

A seeker can obtain public information about the vehicle, such as genesis attributes, debts to the government, arrests, service attributes, legal status, owner, legal owner, manufacturer, and stolen status.

Figure 9 presents a data flow diagram (DFD) of the seeker role application. The DFD starts with the reading of a QR code from the physical vehicle; the code obtained is t ; this code may be obtained manually or in plain text. Then, the application establishes communication with the blockchain network using the same client server mechanism as specified in Section 8.1. Here, the seeker accesses two types of services: public and private. Access to public services is direct and without restrictions. However, private services require permission, so the DFD illustrates that the seeker application must first obtain permission to use the services.

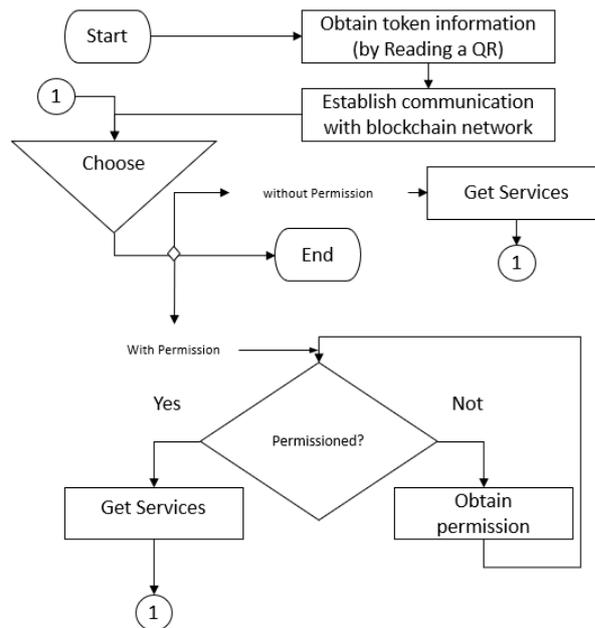


Figure 9. General DFD of the seeker role application for using services.

In Table 7, we show a list of services, indicating whether or not the seeker has permission to access each of them. We designed some services to be available without permission in order to achieve transparency for the whole system (see Table 7). Examples include:

- The verification of the owner and the existence of current stolen or debt reports.
- Inquiries pertaining to basic characteristics such as trademark, model, class, version, and the number of cylinders.

Additionally, some services are available with permission, and these incur transaction costs; for example, ascertaining:

- A list of previous owners;
- How many times the vehicle has been involved in an arrest;
- The number of penalties;
- The history of taxes.

In Table 7, the column ‘Who’ denotes the role authorized to execute changes in the smart contract of the respective service (owner, legal owner, government, manufacturer, and helper). Some variables that are publicly available are shown in Table 8.

Table 7. Consulting services provided in the NFT-Vehicle and their access permissions.

Num.	Service	Permissioned		Who
		Yes	No	
1	Owner		X	O
2	Stolen status		X	G and L
3	Debt status		X	G
4	Basic characteristics		X	M
Historical registers				
5	Owners	X		O
6	Arrest history	X		G
7	Penalty history	X		G
8	Tax history	X		G
9	Maintenance history	X		H

Table 8. Public variables of the token.

Attribute	Description
Color	The color of the vehicle
currentDebt	The amount of current debt related to the vehicle during the purchase process
currentGovDebt	The size of debt to the government during legal processes
government	The government address
helper	The assigned helper's address
inTransProcess	Whether the vehicle is currently involved in a legal or transaction process
LastServDescrip	A description of the last service
legal	True if the vehicle currently possesses a certain legal status; false if any problems such as <i>theft</i> , <i>arrests</i> , or <i>penalties</i> have been resolved.
legalOwner	The legal owner's address
manufacturer	The manufacturer's address
Mileage	The mileage assigned by the last service
name	Name assigned to the vehicle by the manufacturer
No_Cylinders	The number of cylinders
ownerOf(idToken)	The owner's address linked to the idToken
possiblelegalOwner	The address of the likely new owner
priceProposal	The sale price proposed by the owner
stolen	Whether or not the vehicle is currently classed as stolen
symbol	Symbol assigned to the vehicle by the manufacturer
transactionGovCost	Price proposed by the government for a legal procedure transaction
Version	The version of the vehicle

9. Analysis

This section analyzes our proposal from three perspectives: (a) First, the viability of the transactional costs is assessed via two scenarios related to gas consumption. (b) Next, we outline some of the limitations that our architecture still presents. (c) Finally, we discuss our architecture and how it adds more trust and value to physical vehicles.

9.1. Gas Consumption Scenarios

To provide an idea of the amount of gas consumed over the life cycle of a vehicle, we designed two scenarios.

The first scenario represents minimal gas consumption. This was estimated with the operations presented in Tables 1 and 6:

$$\begin{aligned}
 Gas_{mc} &= M01 + M02 + M07 + M09 + M04 + \\
 &\quad M14 + M09 + M04 \\
 &= 3123207 + 121141 + 53550 + 75546 + 81123 \\
 &\quad 76976 + 33366 + 53436 \\
 &= 3618345
 \end{aligned} \tag{14}$$

These operations describe a manufacturer becoming a legal owner and ending the vehicle's life cycle. The minimum gas consumption Gas_{mc} was 3618345 weis.

The second scenario considered the gas consumption of a car that had traveled under 100,000 km. This scenario involved the operations shown in Tables 1, 2 and 4–6. In this scenario, we assumed three purchase processes ($p = 3$), ten maintenance services ($m = 10$), and ten insurance services $i = 10$:

$$\begin{aligned}
 Gas_c &= M01 + M02 + M07 + M09 + M04 + \\
 &\quad p * (M08 + M03 + M07 + M09 + M04) + \\
 &\quad m * (M10 + M11 + M13) + \\
 &\quad i * (M10 + M12 + M13) + \\
 &\quad M14 + M09 + M04 \\
 &= 3123207 + 121141 + 53550 + 75546 + 81123 \\
 &\quad 3 * (123545 + 101014 + 55550 + 55550 + 64013) + \\
 &\quad 10 * (80131 + 89973 + 70675) + \\
 &\quad 10 * (97301 + 220236 + 57239) + \\
 &\quad 76976 + 33366 + 53436 \\
 &= 10972911
 \end{aligned} \tag{15}$$

The total gas consumption Gas_c was 10,972,911 weis. This cost was estimated for each vehicle taking into account the Ethereum platform. If a car is stolen and recovered, its cost is calculated as shown in Table 3 and Equation (15).

As a proof of concept, we developed an experiment showing the transaction costs in ETH. In addition, it could be adapted to more particular solutions, such as adding other roles and methods. To our knowledge, we are the first research group to present the transaction costs in a blockchain solution focused on tokenizing physical vehicle ownership. We found a study analyzing gas consumption [52], but it implemented very different smart contracts. We noted a similarity concerning gas consumption; however, it would be difficult to establish an accurate comparison. In any case, this would be beyond the scope of this article.

9.2. Limitations

In this proposal, the NFT-Vehicle smart contract only supports a single owner, which implies that multiple users cannot buy the vehicle together. Another limitation is that it is impossible to authorize other users to play the owner role, for example, interacting with the government, or to authorize another user to assign helper users or insurance providers. Additionally, we will continue to investigate the routes to obtaining permission to become a manufacturer.

A significant improvement would be the possibility of connecting the onboard diagnostics generation data automatically with the NFT-Vehicle. The onboard diagnostics

system provides access to certain internal status information of the vehicle [53]. Various standards exist, such as OBD-I, OBD-II, EOBD, EOBD2, and ADR, so each maintenance helper in our system could include the scanner result as part of the service.

9.3. Discussion

The NFT-Vehicle architecture we presented herein involves several stakeholders, such as the manufacturer, owner, government, legal owner, buyer, insurance provider, and maintenance provider, who interact throughout the life cycle of a physical car via a token by executing transactions in a smart contract. This contract itemizes the transactions and embeds the life cycle registers of a car in an individual and digital token. Thus, the token can be consulted with public and restricted private (more advanced) methods. Considering the current literature, we added the legal owner role, i.e., an owner following a formal procedure with the government.

Our proposal could be attractive to stakeholders. Additionally, owners will always be grateful for a complete historical view of their physical assets, resulting in better value. The government could provide more and better services to improve accuracy, the transparency of transactions, and the speed of procedures (without the need to process documents manually). Insurance companies could obtain a better estimation of the price of vehicles. Furthermore, vehicle purchase companies will benefit when they act as owners of multiple cars. Finally, buyers could have a greater level of trust, having more certainty regarding the historical transactions related to a vehicle by consulting a unique logbook, and this could increase the value of such cars.

10. Conclusions

We proposed a distributed architecture to store historical transaction records for physical vehicle ownership through blockchain technology smart contracts. This could help solve issues in vehicle purchase transactions, such as fraud.

We modified the ERC721 standard interface to be more suitable to smart properties, thus extending the NFT-Vehicle. The architecture involved the creation of an NFT by a vehicle manufacturer in a blockchain platform such as Ethereum. Then, stakeholders can execute typical transactions in the NFT-Vehicle such as selling, buying, changing the legal status, mechanical services, and adding necessary information. These transactions were protocol interactions between different roles, which we represented and validated as a finite-state machine.

A proof of concept was implemented. We generated a set of proofs and estimated the transaction costs to calculate the certainty regarding the amount of ETH that could be spent in the life cycle of a vehicle. Our test showed that the total transaction cost for each car throughout its life cycle did not represent an excessive cost considering the advantages that the system could offer to stakeholders.

In our architecture, blockchain demonstrated advantages over a traditional database; in the latter, registers can be changed, whereas the former requires a new block for modifications; hence, the historical log persists. In addition, with blockchain, the NFT-Vehicle encapsulates the physical vehicle digitally.

Author Contributions: Conceptualization, data curation, investigation, methodology, software, and writing—J.C.L.-P.; formal analysis, funding acquisition, project administration, resources, supervision, validation, and writing (review and editing)—J.C.L.-P., L.A.M.-R., I.A.-B. and C.D.-V.-S. All authors have read and agreed to the published version of the manuscript.

Funding: The APC was partially funded by Universidad Panamericana Campus Guadalajara. Besides, it was founded by the Catedras-Conahcyt projects 882 and 613.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data presented in this research were generated executing the smart contracts available at <https://github.com/jclopezpimentel/nftVehicles/tree/main/currentVersion>, accessed on 7 April 2023.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

NFT	Non-fungible token
IoV	Internet of Vehicles
API	Application programming interface
HTTP	Hypertext transfer protocol
NIV	Number identifier for vehicle
FSM	Finite-state machine

References

1. Research, G.V. Used Car Market Size & Share Report, 2022–2030. 2022. Available online: <https://www.grandviewresearch.com/industry-analysis/used-car-market> (accessed on 29 November 2022).
2. Duboka, Č.; Filipović, Ž.; Gordić, M.; Došlić, M. Second hand vehicle maintenance frauds. In Proceedings of the Paper NMV0912 presented at the XXII JUMV International Automotive Conference, Belgrade, Serbia, 14–16 April 2009; pp. 31–42. [CrossRef]
3. Shin, D.D. Blockchain: The emerging technology of digital trust. *Telemat. Inform.* **2019**, *45*, 101278. [CrossRef]
4. Chanson, M.; Bogner, A.; Wortmann, F.; Fleisch, E. Blockchain as a privacy enabler: An odometer fraud prevention system. In Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers, Maui, HI, USA, 11–15 September 2017; pp. 13–16. [CrossRef]
5. Abbade, L.R.; Ribeiro, F.M.; Silva, M.H.D.; Morais, A.F.P.; Morais, E.S.D.; Lopes, E.M.; Alberti, A.M.; Rodrigues, J.J.P.C. Blockchain Applied to Vehicular Odometers. *IEEE Netw.* **2020**, *34*, 62–68. [CrossRef]
6. Samue, C.N.; Severine, G.; David, B.; Verdier, F.; Patricia, G.O. Automotive Data Certification Problem: A View on Effective Blockchain Architectural Solutions. In Proceedings of the 2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Virtual, 4–7 November 2020; pp. 0167–0173. [CrossRef]
7. Brousmiche, K.L.; Heno, T.; Poulain, C.; Dalmieres, A.; Ben Hamida, E. Digitizing, Securing and Sharing Vehicles Life-cycle over a Consortium Blockchain: Lessons Learned. In Proceedings of the 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Paris, France, 26–28 February 2018; pp. 1–5. [CrossRef]
8. Leo Brousmiche, K.; Durand, A.; Heno, T.; Poulain, C.; Dalmieres, A.; Ben Hamida, E. Hybrid Cryptographic Protocol for Secure Vehicle Data Sharing Over a Consortium Blockchain. In Proceedings of the 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Halifax, NS, Canada, 30 July–3 August 2018; pp. 1281–1286. [CrossRef]
9. Demir, M.; Turetken, O.; Ferworn, A. Blockchain Based Transparent Vehicle Insurance Management. In Proceedings of the 2019 Sixth International Conference on Software Defined Systems (SDS), Rome, Italy, 10–13 June 2019; pp. 213–220. [CrossRef]
10. Sharma, P.K.; Kumar, N.; Park, J.H. Blockchain-Based Distributed Framework for Automotive Industry in a Smart City. *IEEE Trans. Ind. Inform.* **2019**, *15*, 4197–4205. [CrossRef]
11. Wang, X.; Wang, Y.; Liu, A. Trust-Driven Vehicle Product-Service System: A Blockchain Approach. *Procedia CIRP* **2020**, *93*, 593–598.
12. Syed, T.A.; Siddique, M.S.; Nadeem, A.; Alzahrani, A.; Jan, S.; Khattak, M.A.K. A Novel Blockchain-Based Framework for Vehicle Life Cycle Tracking: An End-to-End Solution. *IEEE Access* **2020**, *8*, 111042–111063. [CrossRef]
13. You-Ting Jiang, H.M.S. A Blockchain-Based Vehicle Condition Recording System for Second-Hand Vehicle Market. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 10. [CrossRef]
14. Pirker, D.; Fischer, T.; Witschnig, H.; Steger, C. velink-a blockchain-based shared mobility platform for private and commercial vehicles utilizing erc-721 tokens. In Proceedings of the 2021 IEEE 5th International Conference on Cryptography, Security and Privacy (CSP), Zhuhai, China, 8–10 January 2021; pp. 62–67.
15. Vitelaru, E.; Persia, L. Fractional Vehicle Ownership and Revenue Generation Through Blockchain Asset Tokenization. *Transp. Telecommun. J.* **2023**, *24*, 120–127. [CrossRef]
16. Kudyba, S. COVID-19 and the Acceleration of Digital Transformation and the Future of Work. *Inf. Syst. Manag.* **2020**, *37*, 284–287. [CrossRef]
17. Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. *Manubot. Econom.* **2019**, 1–9. Available online: <https://bitcoin.org/bitcoin.pdf> (accessed on 7 April 2023).
18. Mitrofanova, I.V.; Larina, O.I.; Dubovik, M.V.; Moryzhenkova, N.V. Evolution of Money Systems or Cashless Economy? In *Institute of Scientific Communications Conference*; Springer: Cham, Switzerland, 2020; pp. 1021–1032. [CrossRef]
19. Di Pierro, M. What Is the Blockchain? *Comput. Sci. Eng.* **2017**, *19*, 92–95. [CrossRef]

20. Xinyi, Y.; Yi, Z.; He, Y. Technical Characteristics and Model of Blockchain. In Proceedings of the 2018 10th International Conference on Communication Software and Networks (ICCSN), Chengdu, China, 6–9 July 2018; pp. 562–566. [CrossRef]
21. Chowdhury, M.J.M.; Ferdous, M.S.; Biswas, K.; Chowdhury, N.; Kayes, A.; Alazab, M.; Watters, P. A comparative analysis of distributed ledger technology platforms. *IEEE Access* **2019**, *7*, 167930–167943. [CrossRef]
22. Wood, G. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Proj. Yellow Pap.* **2014**, *151*, 1–32.
23. Mohanta, B.K.; Panda, S.S.; Jena, D. An overview of smart contract and use cases in blockchain technology. In Proceedings of the 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Bengaluru, India, 10–12 July 2018; pp. 1–4. [CrossRef]
24. Macrinici, D.; Cartofeanu, C.; Gao, S. Smart contract applications within blockchain technology: A systematic mapping study. *Telemat. Inform.* **2018**, *35*, 2337–2354. [CrossRef]
25. Negara, E.S.; Hidayanto, A.N.; Andryani, R.; Syaputra, R. Survey of Smart Contract Framework and Its Application. *Information* **2021**, *12*, 257. [CrossRef]
26. Merriam-Webster. Transaction Definition & Meaning. Available online: <https://www.merriam-webster.com/dictionary/transaction> (accessed on 7 April 2023)
27. Szabo, N. Formalizing and securing relationships on public networks. *First Monday* **1997**, *2*, 9. [CrossRef]
28. Rouhani, S.; Deters, R. Security, Performance, and Applications of Smart Contracts: A Systematic Survey. *IEEE Access* **2019**, *7*, 50759–50779. [CrossRef]
29. Valeonti, F.; Bikakis, A.; Terras, M.; Speed, C.; Hudson-Smith, A.; Chalkias, K. Crypto Collectibles, Museum Funding and OpenGLAM: Challenges, Opportunities and the Potential of Non-Fungible Tokens (NFTs). *Appl. Sci.* **2021**, *11*, 9931. [CrossRef]
30. Dowling, M. Fertile LAND: Pricing non-fungible tokens. *Financ. Res. Lett.* **2022**, *44*, 102096. [CrossRef]
31. Dowling, M. Is non-fungible token pricing driven by cryptocurrencies? *Financ. Res. Lett.* **2022**, *44*, 102097. [CrossRef]
32. Park, A.; Kietzmann, J.; Pitt, L.; Dabirian, A. The Evolution of Nonfungible Tokens: Complexity and Novelty of NFT Use-Cases. *IT Prof.* **2022**, *24*, 9–14. [CrossRef]
33. Willemarck, T.; Graafland, H.; Gonderinger, C.; Cobbaut, J.; Lycke, B.; Hakkenberg, J.; Peelman, M. Protect European Consumers Against Odometer Manipulation. Brussels, 6 October 2014. Available online: https://www.fiaregion1.com/wp-content/uploads/2017/05/joint_appeal_against_odometer_manipulation_final.pdf (accessed on 7 April 2023).
34. Montag, J. Identifying odometer fraud in used car market data. *Transp. Policy* **2017**, *60*, 10–23. [CrossRef]
35. PAPE, Marketa. *Odometer Manipulation in Motor Vehicles*; CID: 20.500.12592/54p8g1; EPRS: European Parliamentary Research Service: Brussels, Belgium, 2018. Available online: <https://policycommons.net/artifacts/1332648/odometer-manipulation-in-motor-vehicles/1936331/> (accessed on 7 April 2023).
36. Borkowski, P. Reducing Odometer Fraud in the EU Second-Hand Passenger Car Market Through Technical Solution. In *Integration as Solution for Advanced Smart Urban Transport Systems*; Sierpiński, G., Ed.; Springer: Cham, Switzerland, 2019; pp. 184–194. [CrossRef]
37. Miller, T. Prevent Odometer Fraud. 20007. Available online: <http://publications.iowa.gov/5850/1/AG3auto.pdf> (accessed on 7 April 2023).
38. Yu, Y.; Yao, C.; Zhang, Y.; Jiang, R. Second-Hand Car Trading Framework Based on Blockchain in Cloud Service Environment. In Proceedings of the 2021 2nd Asia Conference on Computers and Communications (ACCC), Singapore, 24–27 September 2021; pp. 115–121. [CrossRef]
39. Ren, H. The Current Situation of China’s Used Car Market and Development Suggestions. In Proceedings of the 2022 7th International Conference on Financial Innovation and Economic Development (ICFIED 2022), Harbin, China, 21–23 January 2022; Atlantis Press: Amsterdam, The Netherlands, 2022; pp. 1424–1429. [CrossRef]
40. Bauer, I.; Zavolokina, L.; Schwabe, G. Is there a market for trusted car data? *Electron. Mark.* **2020**, *30*, 211–225. [CrossRef]
41. Masoud, M.Z.; Jaradat, Y.; Jannoud, I.; Zaidan, D. CarChain: A Novel Public Blockchain-based Used Motor Vehicle History Reporting System. In Proceedings of the 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Amman, Jordan, 9–11 April 2019; pp. 683–688. [CrossRef]
42. Karandikar, N.; Chakravorty, A.; Rong, C. Blockchain Based Transaction System with Fungible and Non-Fungible Tokens for a Community-Based Energy Infrastructure. *Sensors* **2021**, *21*, 3822. [CrossRef]
43. Sitaru, S.; Kaczmarczyk, R.; Zink, A. Nonfungible tokens (NFTs) in dermatology and health care: A guide for health care professionals. *JEADV Clin. Pract.* **2023**, *2*, 32–37. [CrossRef]
44. Regner, F.; Urbach, N.; Schweizer, A. NFTs in Practice—Non-Fungible Tokens as Core Component of a Blockchain-based Event Ticketing Application. In Proceedings of the 40th International Conference on Information Systems (ICIS), Munich, Germany, 15–18 December 2019.
45. Balci, E.; Yelce, T.U.; Sikar, R.B.; Özgür Bezgin, N. Non-fungible tokens (NFTs) as a potential business tool in the transportation sector. In Proceedings of the 2022 International Technological Sciences and Design Symposium (ITESDES 2022), Giresun, Turkey, 2–5 June 2022; pp. 1–7.
46. Linda, H.P.; Colpoys, L.; McGill, M.; Carrington, D.; Britton, C.; England, H. UML Class Diagram Syntax: An Empirical Study of Comprehension. In Proceedings of the 2001 Asia-Pacific Symposium on Information Visualisation, Sydney, Australia, 1 December 2001. [CrossRef]

47. López-Pimentel, J.C.; Morales-Rosales, L.A.; Monroy, R. RootLogChain: Registering Log-Events in a Blockchain for Audit Issues from the Creation of the Root. *Sensors* **2021**, *21*, 7669. [[CrossRef](#)] [[PubMed](#)]
48. López-Pimentel, J.C.; Morales-Rosales, L.A.; Algreto-Badillo, I. A Cloud Microservices Architecture for Data Integrity Verifiability Based on Blockchain. *Appl. Sci.* **2022**, *12*, 2754. [[CrossRef](#)]
49. Docs, O. Tokens. 2022. Available online: <https://docs.openzeppelin.com/contracts/2.x/tokens#ERC721> (accessed on 13 July 2022).
50. Proposals, E.I. EIP-721: Non-Fungible Token Standard, 2022. 24 January 2018. Available online: <https://eips.ethereum.org/EIPS/eip-721> (accessed on 13 July 2022).
51. Wang, J.; Tepfenhart, W. *Formal Methods in Computer Science*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2019.
52. Khan, M.M.A.; Sarwar, H.M.A.; Awais, M. Gas consumption analysis of Ethereum blockchain transactions. *Concurr. Comput. Pract. Exp.* **2022**, *34*, e6679. [[CrossRef](#)]
53. ISO. ISO 15031-4:2014. 2022. Available online: <https://www.iso.org/standard/55047.html> (accessed on 7 April 2023).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.