# The Tokenization of Vehicles:

## Implications for an open web and the 4th industrial revolution


**Author**
Sebastian Gabriel Savu

**ID**
2173620

**Supervisor**
Mohammed Bahja

**Degree Course**
BSc. in Computer Science

**Date**
25th March 2022

Word Count
9983

School of Computer Science
University of Birmingham

# Contents

# Abstract

This paper expands on how a public ledger can be used to revolutionize massively adopted vulnerable systems and how the impact of such a solution can drive towards the 4th industrial revolution. Topics such as ethics, security and scalability are also discussed around this subject.



## Keywords

Blockchain, Crypto, Cryptocurrency, Open Web, Bitcoin, Ethereum, BTC, ETH, Openness, Web3, Public ledger, consensus, ethics, security, cryptography, civilization, society, decentralization, centralization, immutability, gas, scalability, massive adoption, governmental bodies, vehicles, vehicle manufacturers, corruption, ownership, solidity, smart contracts, IoT, API, industrial revolution, common effort, bioengineering, AI, AGI, quantum.

# 1 Introduction

Over the course of time, humanity has observed several key movements which enabled it to move past industrial and technological barriers, laying the foundation for what we recognize today as the usual. Those specific periods are referred to as industrial revolutions and they allowed for significant motion and progress in all human activity on a global scale. Be it mechanical (First Industrial Revolution), electrical (Second Industrial Revolution), electronic (Third Industrial Revolution) they have all raised the life standard of an ordinary individual and contributed to a massive change in society. Those breakthroughs have been the result of a common effort led by humans, to reform and remodel the frameworks of the generations before them.

Our building blocks are the most advanced that have ever been. But what it means to revolutionize might be different than what we've encountered before (K. Schwab 2016), as for the first time, different scientific fields can be seen to coalesce into new domains, bringing answers to problems that no individual field was able to within its own secluded boundaries.

Human genome modification and artificial augmentation will have to answer the question of "What it means to be human?", and how much will such technology impact the course of our species.

Morality in relation to and from artificially intelligent systems will have to be discussed as we draw closer to engineering consciousness in machines.

Consumer-level breakthroughs in Quantum technologies will have to be evaluated, as such technology threats the very fabric of secure cryptographic barriers in place today.

It is of upmost importance that we implement such advancements carefully, in a way that does not obtrusively obfuscate the valuable civilizational blocks that have been built before us.

As we dive deeper into the 21$^{st}$ century, we are as close as ever to another industrial revolution.

# 2 Research Field

This paper is however focused on another emerging technology, one that is closer to achieving its peak in the near future, and that is the blockchain.

Although several key concepts which define blockchain technologies have seen motion dating as far back as 1991 through the research of various scientists describing a chain of blocks (S. Haber et al. 1991), digital gold (N. Szabo 1998) or cryptographically secured chains (S. Konst 2000), the official conceptualization of a blockchain is Bitcoin (S. Nakamoto 2008).

The simple idea of a peer-to-peer electronic cash system validated by a network of computers running a consensus algorithm, was further expanded in more recent years by the general-scope blockchain (V. Buterin 2014) which has found its way to prominence through projects such as Ethereum, as it is today, the blueprint of a modern blockchain.

In Ethereum we can observe how blockchain networks can go further than just a mere cryptocurrency and how abstraction and generalization can be used to build a multitude of decentralized applications.

Those applications, "smart contracts", benefit from the immutability principle of the blockchain, showcasing how such infrastructure/code cannot be altered and is bound to deliver on the terms previously established by the developer.

The slow progress in blockchain technology was mainly determined by the blockchain trilemma which seeks an answer even to this day: decentralization, security, and scalability. It is impossible implementing all three aspects at the same time as can be seen in modern blockchains which either detrimentally rely on a slightly more centralized and risk-prone architecture to deliver performance (E. Kereiakes et al. 2019) or suffer from scalability issues for mass adoption.

Although consistent work is being put into blockchain engineering even to this day to solve those issues, the current milestones and developments in the space have allowed for real life proof of concepts to appear and showcase what such technology is truly capable of.

The implementation of decentralized finance (DeFi) enabling peer to peer investments and bridging financial primitives, such as lending, exchanging, etc. over to the cryptographic space removes the need for third-party intermediaries like banks and brokerages. New notions in this spectrum such as over-leveraged yield farming through liquidity pools, automated market making and liquid staking, showcase how new ideas can appear to revolutionize the traditional world of finance through new technology.

DeFi, however, pales in comparison with the notoriety non fungible tokens (NFT's) have managed to amass in mainstream media. Making the New York Times front page (K. Roose 2021), the hype and mania generated by digital art ownership through NFT's have allowed it to become in 2021, alongside crypto, as prominent as the internet had been back in 1990. Similar in principle to cryptocurrencies, which are fungible tokens, non-fungible tokens have been designed and focused on representing proof for an object or asset. Although the usefulness of digital art ownership can be questioned, there are other solutions for NFT's such as in DeFi, where they are used as proof to showcase the percentage share of liquidity a user has in a pool.

The two main use cases for blockchain mentioned above and many others have sparked the imagination and creative process of developers and enterprises worldwide to start a massive shift into this space. What we are seeing currently is something similar to the dot com bubble observed at the dawn of this millennium.

This time around the idea is around an open web, Web3. Originated from a libertarian mindset, it envisions an internet where majority of services are decentralized, upholding the peer-to-peer concept to its maximum potential (A. A. Monrat et al. 2019). A landscape of decentralized applications engulfs this space, as old concepts such as social media can be reimagined through ideas such as the metaverse and other exotic terms representing the new vision. Even some of the core fundamentals of the internet such as DNS have also been reimagined on to this new scheme. The focus on openness is key at a direct relationship with blockchains and cryptocurrency as can be seen from the massive amounts of investments shifted into this field. More than 33 billion dollars have been poured by venture capital (VC) in 2021 into crypto/blockchain/web3 oriented startup companies (J. Melinek 2022).

Also in the new decade, countries and governments have started to acknowledge blockchain and cryptocurrencies as many regulations have been introduced by various parliaments across the globe to ensure proper legislation over such technology:

The discussion regarding algorithmic stable coins in the US between parliament officials and blockchain experts expanded upon the legitimacy of collateralizing a crypto asset behind various forms of the United States dollar

Countries like China have decided to completely cut off from cryptocurrency and outright ban it.

Others like El Salvador have decided to fully adopt it and completely overhaul the nation's reserves through it.

One thing is for certain: numerous opportunities are appearing in this new space as the world moves forward and transitions to a new era.

# 3 Surfaced Concern

In this paper I would like to discuss how such technology, blockchain, can be used to redesign and overhaul one of the core, fundamental systems society has set up for its own internal operation and that is the vehicle registry and licensing system adopted worldwide.

Although the current system works well as it has seen success in each country globally, in a similar shape or form, several pitfalls have been identified in relation to its internal operation.

First, the system fails in the face of corruption. Without the necessary extra measures put in place in developed countries such as the United Kingdom, it is extremely easy to subvert its mechanics to benefit a political agenda. Not necessarily specific to vehicle ownership but any type of ownership such as property owned by an individual is susceptible to being forged or framed against them by governmental bodies in developing regions such as Africa or the Middle East, depriving them of a rightful good that they are entitled to. Such actions have massive consequences and impact on the individual, constituting a crime against the human rights.

Second, due to this nonuniformity of implementations worldwide, even the most secure approaches in first world countries are being externally defeated. Organized crime profits from the reforging of documents of old vehicles through sale and portrayal of a false, newer identity of the vehicle on third party markets. The exporting of the vehicle into a less secure implementation of the system allows for corruption and corner case exploitation to successfully reimport the vehicle under reset odometer, new paint and new fabricated documents.

Example: Exporting a car from Germany to Romania, resetting the vehicle's odometer and forging new documents through bribing and then sold on Romanian markets as brand new for a higher price.

Third, due to the complications of making it secure, simple interactions with the system are particularly lengthy and complex for the citizen. Just for the transfer of ownership over a vehicle, both parties have to complete a series of complex paperwork and the validation of this paperwork is also lengthy in nature, a usual interaction being measured in the span of weeks.

Furthermore, the current system is closed loop. It does not allow for third party integration such as applications which could benefit from its data, and even if it does to some degree in certain countries, it will not be present worldwide. A naïve example of an open system would be an API call which returns the total number of registered vehicles in that country or the total number of blue cars. This would allow for a range of opportunities in terms of applications which could benefit both the citizen and the manufacturing companies.

Overall, the current system involves complex paperwork referenced to a centralized corruptible entity and requires a considerate amount of time to establish authenticity and validate simple tasks such as transferring ownership, which in the long term, might prove to slow down the human vision to revolutionize.

At the time of this paper, deep research into the field of blockchain has showcased numerous implementations of the non-fungible token for use cases such as, food traceability, where a private distributed ledger is used by the stakeholders to identify the parties responsible for any issue raised with a product (M. Kim et al. 2018), however, none regard the use of such tokens for the registry and licensing of vehicles. The closest the blockchain industry/research has been to automobiles is regarded through velink (D. Pirker et al. 2021), a vehicle sharing platform that uses the non-fungible token to expand on to the ride sharing concept, which unfortunately does not approach the vehicle ownership concern explained in this paper.

As such in the next couple of chapters I will be presenting how a distributed public ledger allows for an authority to only be trusted through its enforcement (K Wüst et al. 2018) and how such implementations can be realized into an adopted global solution.

# 4 Viewpoint & Methodology

As we embrace new technologies, it is but rational that we apply the knowledge achieved to remodel faulty, older systems that we as society have set up, especially when the difference in capabilities and performance is significant.

I believe a revitalization of the current vehicle registry and licensing system is possible through a decentralized solution using blockchain technology. An ecosystem in which multiple parties can effortlessly interact in a mutually beneficial way:

- The individual to have direct and ultimate ownership over his vehicles.
- The regulatory bodies to enforce such ownership.
- The manufacturing companies to participate in the sale of the vehicles.

(APPENDIX J)

Each vehicle will be represented by a non-fungible token stored in a smart contract on a blockchain.

Proof of ownership for such tokens will be verified through public/private key cryptography. KYC verifiable crypto wallets would then be used to map such addresses to in real life identities.

The relevant history of a vehicle would be stored on the blockchain to ensure accuracy of data and impossibility of falsifying such important information. Specifically, the local odometer will be transformed into a modern solution to allow direct communication with the blockchain and stamp its value. Such technology could either be deployed natively by the vehicle manufactures themselves or as an extension by the regulatory bodies. In this paper I will showcase the latter option on how a microcontroller device can be linked to a vehicle through its licensing plate and autonomously stamp the vehicle's total distance traveled on the blockchain.

In the case of vehicle servicing which leads to a significant change in the vehicle's appearance, the vehicle metadata will be updated by the same licensed entity who completed the handiwork. This will allow for a controlled, valid input of information into the ecosystem by a registered body.

The uniqueness of such a system comes from its openness and further integration. An example of a decentralized marketplace would be later showcased in this paper, highlighting how various tangent entitites can benefit from the simplicity of such a system.

As privacy is a key focus when looking at how transparent such a system can be, no user identifiable data will be processed through the blockchain, leaving KYC information from the wallets such as legal name, region and contact information to the discretion of the various dApps built on top of it.

This leads into the global adoption topic and localization of such a system:

- Either have one massive registry for the entire world and localization of information could be realized from the third-party services.

or

- Have a registry for each country and the information would be already localized.

Nonetheless, this paper showcases one such registry.

The proposed system attempts to mimic the structure of the current system as close as possible as to reduce the effort of switching over and maximize adoption, however some methodologies are unique to the blockchain and must be respected.

I would like to iterate again how this is an initiative that I the author start, an effort to showcase the potential of such a technology, but to truly realize its maximum potential and vision, it is a necessity that regulatory bodies collaborate by regulating its use and implementing it to the nation. The system is not designed as a product but as a societal solution.

I believe the transparency, security and resolve that come with such a solution make it a great alternative to the current system and over the next few chapters I will be expanding in detail how such an entity interacts.

# 5 Architecture & Components

The ecosystem is comprised of four key elements which ensure its functionality: the blockchain smart contracts, the odometer device, a front end dApp and the API.

## 5.1 The Odometer Device

### 5.1.1 Motivation

Storing the local odometer of vehicles on the blockchain to ensure its integrity and immortalize its value, is one of the key reasons for choosing this kind of approach. This is but one of the key elements which uniquely requires to be tracked across the lifespan of a vehicle.

### 5.1.2 Abstract

My project showcases a script running on an ARM microchip board, for example Raspberry PI 4, which connects to the open board diagnostics (OBD) socket of any car through the ELM327 interface via Bluetooth or serial line, computing and transacting the data of the vehicle to the blockchain.

### 5.1.3 Prerequisites & Choices

Starting in the 1990's, all vehicles in NA and Europe have been required to adopt the on-board diagnostics (OBD) system which enables direct access to a vehicle's computer to retrieve information such as the vehicle's speed, engine pressure, etc. (The European Parliament and of the Council 1998)

My project takes advantage of the fact that most vehicles implement such technology nowadays, to showcase how the bare data pulled from a vehicle through such technology can be transformed and stamped on to a blockchain.

Python will be used for its simple runtime and systemwide infinite loop task scheduling capabilities. It is also a great source of libraries and tools for interacting with the real world.

To facilitate productivity and enable ease of diagnosis in a controlled environment, an ELM327 emulator acting as a real-life vehicle with an OBD socket has been used for the development of this device.

The communication with the vehicle is realized through an exchange of hexadecimal bytecode by either Bluetooth or serial line. For ease of use, the official OBD Python library using the bytecode tables for such dialog has been used in the exchange of data.

### 5.1.4 Design

The device is assigned a cryptographic public/private key address/account which is later permissioned into the ecosystem to allow for the transactions to go through. Upon its initialization, the device is configured through an environment file specifying the vehicle it is supposed to monitor, the blockchain and the private key for the assigned account. Using this private key, the device is able to authenticate itself into the ecosystem by signing transactions and updating the value of the vehicle it represents. To reduce waste and promote reusability, the devices can be reconfigured to track any other vehicle on any other blockchain in the ecosystem through any other address.

Initially on live deployment of the ecosystem, the devices will come with a preinstalled cache of the smart contracts that they would be using. However, to ensure fortification of such installations and remove the need to recall the devices in the case of an emergency, they have been equipped with remote API calling capabilities which are securely dictated by the blockchain. The devices are actively viewing the state of a number of variables on the blockchain such as Booleans or strings. Using such immutable values, they are able to act accordingly and either refresh, reset, or even change the blockchain smart contract monitored for such values. As such authenticated and secure remote control is natively built for those devices.

When communicating with the API, several random cooldowns have been put in place to ensure that even the weakest server serving the API could handle the load of millions of vehicles trying to call at the same time.

### 5.1.5 Implementation

(APPENDIX A)

The script running on an endless loop of 1 second schedules, queries the speed of the vehicle, determining the distance traveled by applying the speed formula, and adds this distance to a variable stored locally. Once this variable reaches a certain set threshold, for example 10km, the sending window activates and the script attempts to submit the accumulated distance to the blockchain, immortalizing the value to never be distorted again. Once the sending window activates the device is handicapped at a maximum set percentage of 5% to successfully send the transaction. This is to introduce randomization and unpredictability of the time the transaction reaches the blockchain. More will be discussed in the Risk and Security section on this topic. Once a transaction is successfully sent, the accumulated value is reset to 0 and the sending window is disabled. The initial configuration file is used throughout this process to point the script on tracking the right vehicle on the right blockchain.

### 5.1.6 Risk, Failure & Security

The intrusiveness to such a device has also been considered during development. As such in the case of a criminal assuming control over the cryptographic account through hardware access, there will be no integrity damage to the ecosystem. The script does not set the value on the blockchain but only increment it, so that any attempt at distorting the value in the favor of the perpetrator is turned

against them. There would be monetary loss as the funds, used by the device to send transactions, can be fully retrieved by the criminal, however, the ability of the criminal to extract such funds into the real world would prove very difficult as such transactions are visible on the blockchain and the off-ramps to fiat are government regulated and monitored.

Although the idea of this project was microcontroller connection to the blockchain, a production example could feature a secure microcontroller chip similar to the ones used in the Ledger hardware wallets. Such chips only allow read access and easily prove ownership over an address, allowing the odometer devices to resist against hardware attacks.

Delaying and randomizing the time the transaction is committed to the blockchain is very important and addresses the issue of privacy through tracking live movement of a vehicle/individual. An even greater factor of randomization can be introduced by also randomizing the sending window threshold.

The autonomy and independence of the devices has also been considered during development. Downtime must be minimal and automatic failure correction and recovery must always be present.

The device was not designed with the idea of always being on, and for this reason the current accumulated distance variable is cached locally so that on the event of downtime to the microcontroller, it will be resumed by the device on boot. However, further research into the OBD interface points to an always-on 6V pin which could power the microcontrollers. Live experimentation must be conducted in order to determine this kind of future update.

The connection the devices have with the API and the blockchain allow them to automatically reconfigure and be up to date with any new changes the system might have to offer. A benefit here is that such devices can be deployed even before the entire ecosystem is live and later begin their official activity. This kind of flexibility is great in emergency situations where new smart contracts might have to be redeployed.

## 5.2 The Blockchain Smart Contracts

A smart contract is a stateful piece of code which is executed by the nodes of a blockchain running the Ethereum virtual machine (EVM).

There are two types of functions in the smart contract domain: view-only functions which do not require either authentication or gas to be paid by the user, and state modifying functions, which do require the user to prove ownership over the calling address and pay gas.

### 5.2.1 Prerequisites & Choices

The Ethereum virtual machine (EVM) group of blockchains has been chosen for its high availability of deployment and cutting-edge advancements. Developing an application in such an environment of blockchains enables the application to be deployed on any of them through the execution runtime provided by the EVM, allowing for reusability and a possible integration of our system into a multi-chain future. Not only is this beneficial in the long term, but the community support and toolkit available is far more developed than any other independent blockchain.

Any blockchain from this group can be chosen for the sake of this project however there are a couple extra factors which were involved in the selection process:

- The gas price for transactions has to be reasonable (< 1 dollar)
- The public gateways store the logs of the events emitted

As such we would be using the Fantom blockchain for the main deployment and to showcase the multi chain ability of the later explained dApp, we will be also deploying on the Rinkeby testnet.

The language used for the development of the smart contracts is Solidity, specifically designed for EVM development. The framework used for compilation and deployment is Truffle.

### 5.2.2 Designing on the blockchain

#### *5.2.2.1 Gas Adaptations*

The gas price of a transaction is calculated by multiplying two variables: the execution cost, representing the computational power required to run the transaction, and the base rate, an algorithmic value calculated by the transactions queued for processing.

A huge effort was invested into minimizing the gas footprint of each smart contract. The amount of data stored in a smart contract directly influences the price of the execution cost, as the EVM is required to iterate over many sets of bits. Thus, we are required to only store critical information which necessitates the trust element.

##### 5.2.2.1.1 IPFS

One of the methods used was to store as much bulk information as possible off the blockchain. The inter planetary file system (IPFS) has been designed specifically for this purpose. IPFS is an open web peer-to-peer decentralized protocol designed to store data and persistently enable its access through the use of hashes. The key point here is that the data that we store becomes immutable and permanent similar to how it would be in a smart contract. We will be storing the metadata of a vehicle such as the image, its attributes (make, color, body, etc.) aka non-sensitive data on IPFS saving a lot of space in the smart contract and ensuring that the data is reliable and accurate when accessed through its hash. Each vehicle in the smart contract will then have an IPFS hash linked to it referencing all of the metadata.

##### 5.2.2.1.2 Boolean Bit Storage

Another way of effectively utilizing storage was to change the way Boolean values were stored. The Solidity language defines a Boolean as an 8-bit data type. However, this is 8 times the space required for a simple 0 or 1 value which could be determined by 1 single bit. For this reason, the project would benefit from storing the values for such variables in unsigned integer 256-bit data types where each bit could be interpreted as a Boolean. This way the same space allocated before for 32 values is now allocated to 256 values which greatly increases total available storage and saves a lot of unnecessary gas fees.

##### 5.2.2.1.3 Error Codes

When a function fails, or a check is not passed it reverts the state of the blockchain and returns whatever error message was set in place for that function. Since the number of bits in the contract is also influenced by the length of the error messages, another approach was to return simple 2–3-digit error codes which can be mapped to a more in-depth explanation of what went wrong that can later on be made public, for example, in an API:

E1 - Token doesn't exist
E2 - Not owner of token
E3 - Duplicate URI
E4 - Vehicle is not for sale.
E5 - Vehicle is for sale.
E6 - Vehicle is not an auction

E7 - Vehicle is an auction
E8 - Price must be higher than 2.8237 USD
E9 - Money sent either not enough or too much.
E10 - Bid must be higher than the current price.
E11 - You cannot bid on your own auction.
E12 - Failed to send money to beneficiary
E13 - Failed to transfer vehicle to buyer

#### *5.2.2.2 Modularity & Linking*

Deploying a smart contract returns the address at which this contract can be interacted with on the blockchain. Deployments are unique; two identical contracts would have different addresses. Once a contract is deployed on the blockchain its code cannot be modified. Updating the code of a smart contract would mean to have a new deployment. There is the possibility of "migrating" the state of the old contract to the new contract, but it would involve a massive amount of work, so it is preferred that a system is finished or that it is modular enough to accommodate for plugins.

(APPENDIX K)

For ease of development and management, the functionality for the proposed system shall be split into many contracts as to allow for a modular framework of interlinked parts. A key concept here is the idea of linking, which allows for deployed smart contracts to interact with each other. For example, in our system, the vehicle contract holding all of the vehicles will have to be referenced in the market contract as to point to the underlying assets the interaction is performed on. Another example is the roles contract which will have to be referenced across the entire system to authenticate for permissions in the function modifiers.

This concept is often implemented during deployment passing the address of already deployed smart contracts in the constructor of new contracts, however it is possible to change the links between smart contracts at a later date through functions modifying the state of such references.

Modularity and linking allow for the redeployment of a contract in the case of an emergency. It would impact the system at a minimum as only the information in the redeployed contract needs to be migrated/replaced. However, the degree of impact also depends on the number of dependencies and links the contract is responsible for.

Splitting also allows for ease of detection of events specific to a single interaction. For example, to check when an address incremented its odometer in the last 24 hours it is only necessary to look and filter the Odometer contract where only those type of events are fired.

Splitting also helps with the maximum number of bytes, which is a hard limit at how much information can be contained in a contract.

#### *5.2.2.3 Visibility, Authentication & Gateways*

Function visibility is very important as to control who is allowed to execute a function in a smart contract. Some of the functions do not need to be called outside of the smart contract so they can be set as internal. This automatically restricts any attempt at calling that function from the outside of the smart contract. However, for public functions, aka functions that should be called outside of the blockchain, there has to be a mechanism put in place which dictates who is allowed to do so. Modifiers can be placed in the declaration of a function which filters requests attempting to execute

it based on conditions. We can authorize a single address to be able to execute a function or we could store many addresses in a list and only allow the members of that list to execute a function.

It is very important to make sure that functions with modifiers do not call between each other as the authentication mechanism fails due to the address of the initial caller not being propagated across calls and the smart contract address is used instead.

This led to designing the principle of a gateway on top of a contract, where the underlying function of a contract is set as internal and only the gateway which is a superset of that contract presents public authenticated functions to dictate a fail fast approach.

#### *5.2.2.4 Events*

The events emitted by the smart contracts of our ecosystem have been designed in mind with the role of notifying an update of a certain kind in regard to a vehicle. This is mainly taught for the various third-party applications which could benefit from them off-chain. Events are stored as logs on the different nodes securing the blockchain, so it is up to the node to maintain the validity of their logs. For this reason, events are only to be used to compute past events such as history of owners of a vehicle which would otherwise be storage unfeasible for a smart contract. They are not to be used to provide accurate real time state of the contract.

### 5.2.3 Contract Overview/Specification

(APPENDIX B)

#### *5.2.3.1 Roles*

We define a hierarchy of roles through which we will be able to authenticate interaction with our ecosystem. Each main stakeholder in our system will be assigned a specific role. Some roles have also been assigned an admin role which is able to propagate that respective role without the authorization of the admin role overseeing them. The idea is to mimic real-life organizations and to allow institutions to create their own party in the system. We can also encapsulate each role with its admin role into a role CLASS such that we can refer to both of them for tasks that they have in common. The hierarchical compartmentalization is done in such a way that it represents the direct relationships between parties/roles in real life (APPENDIX D). As such since the governmental bodies are issuing the odometer devices they should manage that role. The same goes for the manufacturing companies and their licensed third-party mechanics.

Spec:

- The contract shall establish the parent-child relationships between the different roles
- The contract shall grant all available roles to the deployer
- The contract shall allow the granting and revoking of permissions for a role of an address
- The contract shall allow checking if an address has a specific role
- The contract shall allow checking if an address is part of a class

#### *5.2.3.2 Vehicle*

This is the ownership registry of the vehicles. It abides by the ERC-721 non fungible token standard which defines an interface that all non-fungible tokens on the Ethereum blockchain should follow.

Spec:

- It shall reference the roles and permissions defined in the Roles contract.
- It shall allow the MINTER_CLASS to create tokens/vehicles
- It shall allow the AUTHORITY_CLASS to burn tokens/vehicles

- It shall allow the retrieval of an owner for a token/vehicle.
- It shall allow the retrieval of the total nr of tokens/vehicles registered.
- It shall allow the retrieval of the metadata of the token/vehicle.
- It shall allow the querying of the existence of a token/vehicle.
- It shall allow the owner of a vehicle to freely transfer his token/vehicle to another person.
- It shall allow the owner of a vehicle to approve other users to interact/use his token/vehicle on behalf of them
- It shall allow the owner of a vehicle to approve a specialized body (garage) to modify the metadata of their token/vehicle
- It shall allow the retrieval of the approved specialized body (garage)
- It shall allow the notification of any state change to a vehicle through events

#### *5.2.3.3 Market + Gateway*

The market shall allow the users to effortlessly exchange their vehicles using listings. The market defines the core of a decentralized market. Users do not directly interact with the market but with the Gateway which is a superset of the market.

The transactions will happen in cryptocurrency, specifically in ether as this is the native currency on the Ethereum blockchain. There is a 10 cents tax withheld by the contract on any successful exchange. A later implementation of a treasury is discussed in the Prospects section.

Spec:

- It shall reference the vehicle registry defined in the Vehicle contract.
- It shall allow the owner of a vehicle to list it for sale as either instant purchase or auction
- It shall allow the owner of a vehicle to delist it from sale
- It shall allow the owner of a vehicle to conclude an auction if there are any bidders for that listing
- It shall allow the owner of a vehicle to amend the price for an instant listing
- It shall allow the retrieval of the price for a vehicle
- It shall allow the retrieval of the top bidder for a vehicle
- It shall allow the retrieval of sale status for a vehicle
- It shall allow a user to purchase a vehicle that he is not owner of
- It shall allow a user to bid on a vehicle that he is not owner of
- It shall allow a user to donate to the contract out of good will
- It shall emit events when there is any state update on a listing
- It shall retrieve all the information regarding vehicles and listings
- It shall withhold a 10 cent tax for any successful exchange between users

#### *5.2.3.4 Odometer*

The odometer contract natively stores the value of each car’s odometer.

Spec:

- It shall reference (syntactically) the vehicle registry defined in the Vehicle contract.
- It shall reference the roles and permissions defined in the Roles contract.
- It shall allow an odometer to increase the value for the vehicle it has been assigned
- It shall allow the AUTHORITY_CLASS to assign an address as odometer for a vehicle
- It shall retrieve the value of the odometer for a vehicle

#### 5.2.3.5 Management

The management contract is directly addressing the odometers and uniquely shows how an IoT device can be controlled through the immutable data of a blockchain. Several variables are used in this contract to instruct the listening odometer devices on how to act.

Spec:

- It shall reference the roles and permissions defined in the Roles contract.
- It shall allow the DEFAULT_ADMIN to set the domain name for the API address
- It shall allow the DEFAULT_ADMIN to set the Boolean value for refreshing the odometers cache
- It shall allow the DEFAULT_ADMIN to set the Boolean value for restarting the odometers
- It shall allow the retrieval of the Boolean value for refreshing
- It shall allow the retrieval of the Boolean value for restarting
- It shall allow the retrieval of the value for the API domain name

#### 5.2.3.6 Bool Bit Storage

Spec:

- It shall set the value of a bit in a 256-bit integer to either 0 or 1
- It shall get the value of a bit in a 256-bit integer
- It shall get the position of a token/vehicle in the 256-bit integer variable

### 5.2.4 Security

There are two types of clever attacks which can profit from the inexperience of smart contract developers.

One of them is the replay attack (T. Nakamura 2020) which is related to the storage of user signatures. A benefit of storing signatures is to reduce the number of transactions the user has to post to the blockchain. Although it will involve a larger number of transactions, our contracts benefit from protection against such attacks by not storing and manually handling signatures of users.

The second clever attack against smart contracts is the reentrancy attack (T. Nakamura 2020). This mainly occurs in smart contracts where the regular user can withdraw value that belongs to them back to their address. The problem is when the user is a specially crafted smart contract which on money receipt initiates an infinite loop fallback exploiting poor defense mechanisms of such contracts. Our contracts do not allow users to directly retrieve any value. The contract acts as an atomic intermediary during money transfers.

### 5.2.5 Testing

The smart contracts have been tested for correct functioning since the early stages of development. Unit tests have been created for each contract to ensure its functionality adheres to the desired standard and on success, the contract would be tested for integration with case exploration to evaluate if the functionality implemented breaks the rest of the system. System testing was conducted periodically after majority of contracts have been deployed since changes would be smaller in nature and an overall functionality was desired.

#### 5.2.5.1 Vehicle

| Test Nr. | Test Description | Expectation | Actual Outcome |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 1 | Check the token URI for a vehicle that does not exist | The returned valued should be an empty string | PASS – On completion the value returned was “” |
| 2 | Check if setting the URI for a vehicle is correctly authenticated as a basic user | The smart contract should revert. | PASS – The smart contract successfully throws “Insufficient permissions” |
| 3 | Check if minting is correctly authenticated as a basic user | The smart contract should revert. | PASS – The smart contract successfully throws “Insufficient permissions” |
| 4 | Check if burning is correctly authenticated as a basic user | The smart contract should revert. | PASS – The smart contract successfully throws “Insufficient permissions” |
| 5 | Check if identical URIs is allowed when minting | The smart contract should revert. | PASS – The smart contract successfully throws “E3” |
| 6 | Check if setting the URI for a vehicle as a garage is allowed for non-authorized vehicles | The smart contract should revert. | PASS – The smart contract successfully throws “Reverted.” |
| 7 | Check if approving a garage works as the owner of a vehicle | The approved address should be returned by getApprovedGarage() | PASS – The smart contract successfully returns the approved address. |
| 8 | Check for the existence of an inexistent token | The smart contract should return false | PASS – The smart contract successfully returns false |
| 9 | Check for correct approvals | The smart contract should return the address of the natively approved address | PASS – Calling getApproved() correctly returns the approved address |
| 10 | Check if approval for new vehicle is empty | The smart contract should return the 0x0 address | PASS – Calling getApprovedGarage() returns the 0x0 address |
| 11 | Check total Supply functionality | The smart contract should return the number of vehicles in the contract | PASS—The contract successfully returns 6 vehicles |
| 12 | Check vehicle of owner at index functionality | The smart contract should return the token Id of the owner on the first position | PASS – The contract successfully returns the indexed vehicle |

#### *5.2.5.2 Market + Gateway*

| Test Nr. | Test Description | Expectation | Actual Outcome |
|---|---|---|---|
| 1 | Token listing as instant | On querying of isForSale the smart contract should return true | PASS – The smart contract successfully returns true |
| 2 | Token listing as auction | On querying of isAuction the smart contract should return true | PASS – The smart contract successfully returns true |
| 3 | Token delisting | On querying of isForSale the smart contract should return false | PASS – The smart contract successfully returns false |

| | | | |
|---|---|---|---|
| 4 | Check top bidder for vehicles with not top bidder | On querying of getTopBidder the smart contract should return 0x0 | PASS – The smart contract successfully returns the 0x0 address |
| 5 | Set vehicle price of a non for sale vehicle | The contract should revert | PASS – The contract successfully throws “E4” |
| 6 | Set vehicle price of for sale vehicle | The contract should display the correct price | PASS – The price is the one set |
| 7 | Purchasing a vehicle | The contract should correctly display the new owner | PASS – The new owner is the purchaser |
| 8 | Bidding on a vehicle | The contract should correctly display the new top bidder | PASS – The new top bidder is the caller of the function |
| 9 | Conclude auction | The contract should correctly delist the vehicle and change owners | PASS – the vehicle is delisted and the new owner is the top bidder |

#### *5.2.5.3 Roles*

| Test Nr. | Test Description | Expectation | Actual Outcome |
|---|---|---|---|
| 1 | Granting a role without being the admin for it | The contract should reject any unauthorized attempt at granting | PASS – The smart contract successfully reverts |
| 2 | Revoking a role without being the admin for it | The contract should reject any unauthorized attempt at revoking | PASS – The smart contract successfully reverts |
| 3 | Granting a role with enough privileges | The contract should change the state of the role for the address | PASS – The address is successfully part of the new role list. |
| 4 | Check class functionality | The contract should permit 2 addresses of the same class to the same action | PASS – The functions are successfully called with the class functionality |

#### *5.2.5.4 Management*

| Test Nr. | Test Description | Expectation | Actual Outcome |
|---|---|---|---|
| 1 | Setting the refresh without permission | The contract should revert | PASS – The smart contract successfully reverts |
| 2 | Setting the restart without permission | The contract should revert | PASS – The smart contract successfully reverts |
| 3 | Setting the apiAddress without permission | The contract should revert | PASS – The smart contract successfully reverts |
| 4 | Retrieving the apiAddress | The contract should return the api Address | PASS – The smart contract returns “fyp.sgsavu.com” |

#### *5.2.5.5 Odometer*

| Test Nr. | Test Description | Expectation | Actual Outcome |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 1 | Check if incrementing as other than odometer | The contract should revert | PASS – The smart contract successfully reverts |
| 2 | Check if incrementing as odometer but unauthorized for vehicle | The contract should revert | PASS – The smart contract successfully reverts |
| 3 | Check if authorizing odometer for vehicle as unauthorized user | The contract should revert | PASS – The smart contract successfully reverts |
| 4 | Check if getting odometer value is correctly returned | The contract should return the correct odometer value | PASS – The smart contract returns the correct value |

More automatic testing available in the /test directory of the project

## 5.3 The Decentralized App (dApp)

### 5.3.1 Motivation

The role of a decentralized application is to visually represent the data on a blockchain and allow the users to seamlessly interact with it. It is different to that of a regular front end application as no data from the user is stored at any point. The decentralized app presented in the next sections represents one clear example of a third-party integration to the open system created previously.

### 5.3.2 Tools & Prerequisites

For the scope of our app the React framework using JavaScript is suitable and versatile enough to deliver reliability, performance, and scalability to a multi-page website.

We are also using the Redux state store which is a great addon to React for organizing global state across the app.

The redux store will be compartmentalized into four different state managers, segregating the stored data into: blockchain connection, data from the blockchain, global app alerts, form data. Such app-wide state is persistent and can be referenced at any point. Updating the stores is done through the use of concurrent dispatches.

For realizing connection to the blockchain and instantiating smart contracts we will be using the popular Web3js library. To send authenticated transactions the users are required to use crypto wallets. We will be implementing the functionality of the popular MetaMask wallet for our users to use, however other wallets could be added at a later point.

A network tables JSON file is to be constructed with templates referencing the networks/blockchain the app is deployed on. This is to be referenced across the app for network switching and adding networks to users' wallets.

### 5.3.3 Design Concepts

#### *5.3.3.1 Main Flow*

The main goal of the app is to grab data from the blockchain and display it to the user in an organized fashion allowing several actions to be performed on such data.

When the app's network is bound to a blockchain it triggers the flow of loading our smart contracts from that network and then fetching the data from them. On initialization the app sets the default network to the first value in the network tables which triggers the initial flow.

#### 5.3.3.2 Smart Contract Procedure Calls

Several procedure calls have been developed to allow for any type of function to be called in the smart contract ecosystem. Those procedures automatically check if the methods that are being called are present in any of the smart contracts and if so, pass the arguments to the transaction. This has also been extended to allow for searching and querying any attribute in relation to the contracts such as the contract's events, networks, function parameters etc.

#### 5.3.3.3 Account Dynamics

Users do not need to connect their wallet in order to view information on the blockchain. This is a great feature as users can browse the market freely.

To accommodate for this, when the wallet is not connected, the app uses the public gateway RPC URLs provided by the blockchain. When authenticated functionality is needed, such as purchasing a vehicle or viewing their vehicles, the user will have to login and the app will be using the wallet to send transactions through to the blockchain.

When a user first connects their wallet to a dApp, the wallet will remember that domain and on the next time the domain is loaded the wallet will automatically provide the user's address to the dApp without the request of the user's permission. We will be using this feature to automatically login discovered users for a smoother experience.

#### 5.3.3.4 Multichain Support

The availability of a dApp on multiple chains enables the formation of a larger community. Users prefer using their favorite blockchain and restricting such activity to one might lose the desire to interact with the app.

The challenge here was to synchronize the network selector the app is using to the one the user is currently on in their wallet.

The user might have never used one of the networks the app has been deployed on so for this reason the template for that blockchain must be added to their wallet. When a user logs in, the app detects their wallet's network and if it is one of the networks the app is deployed on it automatically switches to it, otherwise it will ask the user to add that network to their wallet and switch to it.

When a user is connected, they can change the network the app is on from both their wallet and from the app. In the case of the user using their wallet's selector to change to a foreign network, the app will automatically ask them to switch to its default network.

#### 5.3.3.5 Data Manipulation & Data Marking

To fetch data from the blockchain we query the smart contacts for a list of all the information for all the vehicles. Reason explained in the Testing section.

A JS object will then be assembled for each vehicle which represents the aggregated data pulled. Finally, an object of objects of all the vehicles will be stored in the redux store for data to be queried app wide. The selection of an object of objects instead of a list drastically improves updating times and allows for modular interchange of data.

The assembling process into objects is what defines which data is being displayed to the user. During the assembling process the app checks if the address of the user matches or is included in any of the data pulled from the blockchain. If it does, such as for example if the user is the owner for that vehicle, it injects a Boolean value into that JS object to mark it and later identify it as being owned by

the user. Such injected marks span a multitude of attributes and allow the different pages to easily display tailored content to the user.

#### *5.3.3.6 Asynchronous Data Refreshing*

The events emitted by the smart contracts on successful execution of a transaction are at the core of the application. Several local listeners are set up for the smart contracts the app is currently connected to. In the case of an event triggering, the state of the app changes as to accurately represent the real time update which happened on the blockchain. For example, if a vehicle becomes listed for sale on the blockchain, the app will detect this change and asynchronously add it to the list of vehicles. The global redux store is modified here. We use the information from the event to rapidly identify the vehicle in the store and manipulate the data.

#### *5.3.3.7 Modals and Status*

A large amount of functionality occurs asynchronously. For this reason, the status of the application in terms of processing and loading must always be displayed to the user. Several last in first out alert stacks in the redux store have been created to better display such activity. When each stack is empty that means the app is doing no processing.

1. Loading - The loading backdrop is a persistent stack which continuously displays a loading screen until it has been emptied of loading activities such as fetching the data or initialization.
2. Pending - The pending modal is an incrementing reference value message which actively displays the number of transactions currently sent for processing. This will allow the user to send a transaction and be able to complete other activities in the app while this transaction is validated by the blockchain to later on be notified of its outcome.
3. Custom (Success/Error) - The custom modal is a one-time message which displays either success or failure of actions across the app.

#### *5.3.3.8 Currency*

The dApp will integrate all official currencies worldwide using the Coinbase API. This is used to localize the blockchain ether value of each vehicle. A huge benefit here is for the user input which can now be referenced in their preferred currency. All inputs will be converted into ether before being sent to the blockchain as to natively use the transactions as payments.

#### *5.3.3.9 Roles and Menus*

The user's role is determined from the smart contracts when signing in and this allows for the app to display the correct menus for the user. Although access to certain pages is restricted for unauthorized users, URL navigation security is a best effort since transaction authentication is truly realized on the blockchain.

Menu Tabs Availability:

- Not connected user: The Market
- Connected User: The Market, My Vehicles
- Minter User: The Market, My Vehicles, Mint
- Minter Admin User: The Market, My Vehicles, Mint, Admin*
- Authority User: The Market, My Vehicles, Verify
- Authority User: The Market, My Vehicles, Verify, Admin*
- Default Admin User: The Market, My Vehicles, Mint, Verify, Admin*

*The functionality within the admin page is displayed based on the level of privileges.

#### *5.3.3.10 Market Discoverability*

The market is designed as an easy to use discover-it-all tool. Using an aggregated keyword filtering search bar, the user can set combinations of different terms such as:

{show: my listings, type: auctions, price: descending, make: Tesla}

The result will be a subset of the initial list displaying vehicles that match the criterion. When selecting keywords, the app creates a filtering object with the required criteria. The current vehicle list is dynamically filtered and sorted on meeting the conditions mentioned in the filtering object.

The list of vehicles shown at all times is split into multiple pages to not overwhelm the user.

The vehicle attributes in the search bar such as color, make, body, etc. are automatically generated from the list all vehicles on the market to remove uncertainty and favor user exploration. As this might create a long list of options for the user, the search bar can also be searched for the desired keywords.

#### *5.3.3.11 Bite-sizing displayed information*

The vehicles are represented by compressed information cards which allow for information to be neatly relayed to the user in a non-overwhelming fashion.

Each vehicle is represented by a tile image card briefly summarizing the key information about the listing: Make, Model, Year, and the price. A set of icons providing extra visual cues about the type of the listing are also present in this view. A legend for those icons can be found by hovering over a tooltip above the vehicle list.

Hovering over each card, reveals a set of extra information showcasing the transmission, fuel, and color of the vehicle. A more in-depth look at the vehicle can be found through its profile page by clicking on the card.

### 5.3.4 Specification

(APPENDIX C)

#### *5.3.4.1 Market*

- It shall display a list of cards of all vehicles up for sale
- It shall redirect a user to the vehicle's profile page upon clicking on a card
- It shall allow the user to filter the list by using a combination of predefined keywords
- It shall allow the user to filter the list by using a price range slider.
- It shall allow the user to navigate pages when using the bottom navigator

#### *5.3.4.2 Vehicle Card*

- Clicking on this component shall redirect to the vehicle profile page
- It shall display the make, model and year of a vehicle as a focus for the card
- It shall display the price of the vehicle as secondary focus
- It shall display upon hovering an additional set of information: transmission, fuel, color
- It shall display various helper icons to indicate the type of the listing

#### *5.3.4.3 My Vehicles*

- It shall display a list of all vehicles held by the user
- It shall present the information as is in the market
- It shall redirect to the vehicle's profile page when clicking on a vehicle

#### *5.3.4.4 Vehicle*

- It shall display the picture of the vehicle
- It shall display the metadata information of the vehicle
- It shall display the history of the vehicle
- It shall display the odometer value of the vehicle
- It shall allow the owner to list the vehicle for sale as either instant or auction through the contract
- It shall allow the owner to amend the price of the vehicle through the contract
- It shall allow the owner to delist the vehicle from sale through the contract
- It shall allow a user to buy a vehicle if the vehicle is for sale through the contract
- It shall allow the user to navigate between history/listing when clicking on the tab menu
- It shall allow the user to navigate to the approve garage screen if the user is the owner

#### *5.3.4.5 Mint*

- It shall allow the user to upload a picture of the vehicle
- It shall allow the user to input information about the vehicle in four steps
- It shall display to the user all the information about to be submitted before submitting
- It shall return to empty after successful transmission

#### *5.3.4.6 Verify*

- It shall allow the user to search for any vehicle listed or unlisted from the system
- It shall return a list of all vehicles matching the current searched terms
- It shall redirect to the vehicle's profile page on clicking on any returned search results

#### *5.3.4.7 Admin*

- It shall the user to grant/revoke roles through the contract
- It shall allow the user to burn vehicles through the contract
- It shall allow the user to set odometers for vehicles through the contract

#### *5.3.4.8 Garage*

- It shall display a list of all approved vehicles for that user
- Clicking on any of those vehicles shall redirect to the minting form with prefilled information from that vehicle

#### *5.3.4.9 Main Menu*

- It shall change the content of the page when clicking on any tab
- It shall only display the relevant tabs for the user's role

#### *5.3.4.10 Network Selector*

- It shall refresh the system as to display the data on the selected chain

#### *5.3.4.11 Account selector*

- It shall open a popup dialog with information regarding the user's account if the user is connected
- It shall open a dialog with the available list of wallets to connect if the user is not connected
- It shall allow the user to disconnect at the click of a button

### 5.3.5 UX

The accessibility and ease of use were key when designing the dApp as this is an interface which would be adopted globally by all kinds of individuals. The following represent the adopted key principles and heuristics:

- Error prevention can be noticed in how certain buttons and confirmations pop up and can be interacted with only when the input from the user is correct/viable.
- Persistence of state through the usage of modals and loading, allowing the user to understand the current status of the system at any time.
- A consistent and simple view throughout the entire app with only small element differences on certain pages.
- Freedom to browse the app during loading periods.
- Resemblance of real world through the use of intuitive icons and interactions throughout
- Recognition rather than recall through the use of predefined keywords in the market search rather than user's own input
- Guidance provided when certain features might stir confusion in the user (tooltip in market).
- A great effort was put into making each visible component as reactive as possible in the face of different aspect ratios and screen resolutions. As such the components are dynamically resizing themselves to better fit the entire screen as the resolution becomes smaller.

(J. Nielsen 1994)

A demo of the dApp can be found at:

- https://fyp.sgsavu.com/ (AWS HOSTING)
  or
- https://fyp2.sgsavu.com/ (OWN HOSTING)

### 5.3.6 Risk Analysis

#### *5.3.6.1 Importance of the wallet*

The wallet mnemonic or the private key of an address are the sole agents which prove that a user has ownership over that resource/address. As such upmost care and precautions must be taken by the user to ensure their assets are not lost or stolen. The loss of access to the wallet means inability to prove ownership over the vehicle. Although fraudulent activity is public on the blockchain, the damage can still be felt by the user.

#### *5.3.6.2 Fake domains*

As with any website, there is the possibility of fake domains that trick the user into thinking they are using the original dApp communicating with the smart contract. Attacks could lead to the victim losing ownership over their vehicle or even entire wallet. Protections for this are automatically embedded into the browser wallets user use. When attempting to send a transaction to a smart contract, the wallet checks for a database of whitelisted domains for certain popular smart contract ecosystems and warns the user if the current website is not listed among them. This practice is similar to the certificate warning most browsers show today on invalid or expired certificates. Although sometimes those measures are efficient, some users might still fall victims to such attacks.

#### *5.3.6.3 DDoS*

Although unlikely, there is the possibility of a denial-of-service attack on the front-end app or the API which could lead to service outages and a poor experience. Although this does not impact the ecosystem's functionality, it will negatively impact users trying to use it. The users could use the block explorer to interact with the contract directly, but it is not a great experience. Solutions to this are presented in the Prospects section.

### 5.3.7 Testing

Performance testing was conducted after major milestones to determine design decisions and to allow for evaluation of next steps.

The performance test results showcased a massive opportunity for improvement where the time for pulling the data from the blockchain was 15 seconds for 7 vehicles. This would be unusable in production, so a more effective method had been developed to combat this, reducing the 15 second time to between 200-600ms constant pulling time for any amount of vehicles (up to 100 million).

The issue discovered was to do with how the data was being queried from the blockchain. Although the correct, developer friendly method would be to query the blockchain multiple times with different types of requests for each vehicle, in reality such a practice could introduce a vast amount of response delay due to the nodes processing a large number of requests. As such we would instead rely on the blockchain's distributed computational power to assemble all of this information and return it to the app in one request which vastly improves the delay time and completely removes the scalability of such delay.

**App-wide testing**

| Test Nr. | Test Description | Expectation | Actual Outcome |
|---|---|---|---|
| 1 | Search feature dropdown | Clicking on the search bar displays all of the available options | PASS – The list successfully displays the predefined keywords |
| 2 | Range filter | Moving the thumbs left and right dynamically changes the list of vehicles | PASS – The list is filtered accordingly |
| 3 | Setting keywords | Clicking on a keyword filters the vehicle list to that keyword | PASS – The list is successfully filtered |
| 4 | Clicking on next page | The page should display the next page | PASS – The new page with vehicles is show |
| 5 | Clicking on a vehicle card | It should redirect the user to the vehicle's profile page | Pass – The app redirects the user to the vehicle page |
| 6 | Tab navigability | Clicking on one of the three mini tabs redirect the user to either history/listing | Pass – the app switches content based on the selection |
| 7 | Sending transactions | The MetaMask wallet pops up asking the user to send the transaction | Pass – The transaction is successfully sent |
| 8 | Minting Form Reset | It should refresh on successful completion and transaction | Pass—The form is reset and the user set back to page1 |
| 9 | Minting Form Validation | The form should not allow the user to move to the next step unless all fields are completed | Pass – The form displays the errors on the incomplete fields |
| 10 | Verify Search | The search should return a list of vehicles if any are matched | Pass – The list of vehicles returned matches the search |

| | | | |
|---|---|---|---|
| 11 | Loading | It should stop loading after information has been loaded | Pass – the loading stops |

More tests available in /test in the project's directory.

## 5.4 The API

The scope of the API is for programmatic access to the smart contracts rather than the experience of the user.

### 5.4.1 Motivation

The idea of an API came when designing the minting form for the dApp. As some companies manufacture vehicles in the range of hundreds to thousands daily, the idea of going through the minting form for such a large number seemed very discouraging. Using the API, it is possible to send concurrent atomic transactions to the smart contract making sure the time it takes to register those vehicles onto the blockchain is reduced to a minimum.

### 5.4.2 Design

The API is based on the REST model and only uses JSON objects for requests and responses. All the functionality in the smart contract ecosystem is available through the API using similar procedure calls for smart contract functionality discovery as the dApp.

Apart from serving user's requests in relation to the smart contracts, the API is key in refreshing the cache in the odometers.

### 5.4.3 Specification & Structure

An API doc explaining the structure of each request and the responses that will be given has been created to help and promote exposure to the API. The documentation is available at: https://armenz-savu.gitbook.io/api-docs/

### 5.4.4 Risk & Security Analysis

Since the private key of users must be sent for some requests, all traffic will be run under HTTPS.

Since some files are directly being read from the API, the file paths have been hardcoded to protect against directory traversal.

A DDoS attack on the API during an event such as the odometers refreshing their cache, could mean potential disruption and a delay to completing the process, although the odometers would keep on functioning with the old cache. A solution is discussed in Prospects.

# 6 Evaluation

User evaluation has been conducted through 1-on-1 lab sessions to understand which features seem to resonate with the community and which do not. A 2-part conditional questionnaire (APPENDIX E) had to be completed by the selected participants. Minimal guidance and understanding have been given to the participants before interacting with our ecosystem. The audience has been chosen as to represent a likely scenario of diverse individuals who would be using the app:

- Manufacturers affiliates
- Government affiliates
- Civilians
- Mechanic shops affiliates

The results (APPENDIX F or APPENDIX G) show that the overall sentiment towards the proposed system is generally positive and that the problems identified in this paper with the current system are resonating with others as well.

An issue identified by the questionnaire was in relation with the browser wallet which stirred a bit of confusion overall in its functionality and understanding. Another low-ranking category was identified by the trust the users have given the system mostly this could be interpreted by the skepticism of digital ownership.

# 7 Prospects & Reflection

There are a couple of features and improvements that did not make the priority agenda for the publishing of this paper, that would benefit the ecosystem greatly if added in a later release. Alongside those features I would like to mention the next steps for this project for it to reach its maximum potential.

## 7.1 Protection against DDoS

A solution could be represented by hosting the front-end website and the caches on IPFS. Performance on IPFS in terms of loading and availability could be reduced in comparison to normal server hosting but it does present protection against DDoS.

Another option would be to invest and set up several anti DDoS defenses such as the Cloudflare check for automated requests and create a very strong barrier against such attacks. Distinctions need to be made from an odometer trying to repeatedly pull the cache from the API and a malicious user imitating such frequency of requests to ensure the defenses are not used against ourselves.

## 7.2 AI powered user editing

The idea of allowing the user to freely modify the image of the vehicle could become a reality. The current state of machine learning and AI is strong enough to allow for accurate detection of certain traits and objects present in a picture based on a list of keywords. As such, when the user would upload a picture to change their vehicle's avatar, it would then be submitted to a machine learning algorithm which would use the traits of the vehicle to identify that the picture is relevant and represents the automobile accurately.

## 7.3 TypeScript

The problems with the JavaScript language are well-known and most of the times they create an unfavorable environment for the developer trying to deliver a solid solution. Upgrading the codebase to the TypeScript language, a super set of JavaScript which uses data types and static typing, might prove a better solution for scaling the project further to a larger development team.

## 7.4 Treasury

Since a small tax is withheld by the Gateway contract upon a successful exchange, the money could be used to further fund the project or to collect governmental tax for such exchanges. An additional Treasury contract could be created to allow the various parties involved to syphon money into the real world based on their implications and roles.

## 7.5 Subnets

Although the deployment of a blockchain for the sake of this project is relatively undesirable, creating a custom subnet within another blockchain could look promising and pose enough officiality for it to become a global solution K. Sekniqi (2020).

## 7.6 The cost

The total amount of ether required by the odometers to send transactions is high and a collaboration with regulating bodies is absolutely necessary to ensure a steady influx of such currency. To prevent inflation, the injected value can be burned by the blockchain instead of being distributed to the nodes.

## 7.7 Closing thoughts

It is clear from the evaluation that there is doubt about the security of digital ownership amongst the public and maybe time and education will allow such concerns to dissipate.

Either way, the proposed solution has massive potential as detailed previously.

Such an ecosystem does not necessarily have to replace the current one, but it can always be used as an alternative option, dealing with the inconsistencies of the other.

I kindly ask the regulating bodies, manufacturing companies and other parties to join this effort, at making it a common struggle; to allow the human civilization advance in greatness and prosperity.

# 8 Project Management & Development

The AGILE development framework has been used for the development of this project due to its ability to reiterate on the process of design and implementation. Each sprint was determined by one of the major milestones of the project: the smart contracts, the dApp + API, the odometer device. The supervisory meetings have been used as a weekly standup to discuss about the work that has been completed and work which is to be conducted, providing valuable feedback on the decisions made throughout.

The following schedule of personal deadlines had been set since the start of the project and was in majority accurately respected:

| Sem | Week | Task |
|---|---|---|
| I | 1 | Setup the development environment and realize connection to the blockchain, choose blockchain. |
| | 2 | Understand the basics of the Solidity language and develop first contract - Vehicle |
| | 3 | Become intermediary in the Solidity language and continue smart contract development: Bool Bit Storage, Market |
| | 4 | Become proficient in Solidity language and implement the modifier – Gateway contract |
| | 5 | Split smart contracts and link together. |
| | 6 | Setup the basics of the React framework. |
| | 7* | Setup a basic dApp functionality to showcase the vehicle registry for inspection. |
| | 8 | Resume work on the smart contracts – Odometer, Management |
| | 9 | Refactor smart contract platform for maximum efficiency and gas |
| | 10 | Begin implementation of whole smart contract functionality into dApp |
| II | 1 | Setup API simple requests |
| | 2 | Entire smart contract functionality available in dApp. |
| | 3 | Finish API development and integrate entire smart contract functionality |
| | 4 | Research around IoT devices and blockchain, begin odometer solution |
| | 5 | Research and design styling for the dApp |
| | 6 | Finish implementing and adding odometer device to system |
| | 7 | Finish styling of dApp |
| | 8** | System-wide testing for demo, refactoring |
| | 9 | Conduct evaluation |
| | 10 | Determine future plans and draw a conclusion |

*Inspection Week

**Demonstration Week

# 9 References

**Min Xu et al. (2018)** The Fourth Industrial Revolution: Opportunities and Challenges [online].
[Accessed 20th September 2021]. Available at: <https://doi.org/10.5430/ijfr.v9n2p90>

**K. Schwab (2016)** The Fourth Industrial Revolution [online].
[Accessed 20th September 2021]. Available at:
https://law.unimelb.edu.au/__data/assets/pdf_file/0005/3385454/Schwab-The_Fourth_Industrial_Revolution_Klaus_S.pdf

**Yli-Huumo et al. (2016)** Where Is Current Research on Blockchain Technology? —A Systematic Review [online].
[Accessed 21st September 2021]. Available at:
https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0163477

**T. Aste et al. (2017)** Blockchain Technologies: The Foreseeable Impact on Society and Industry [online].
[Accessed 22nd September 2021]. Available at: https://ieeexplore.ieee.org/document/8048633

**Z. Zheng et al. (2018)** Blockchain challenges and opportunities: a survey
[Accessed 22nd September 2021]. Available at:
https://www.researchgate.net/publication/328271018_Blockchain_challenges_and_opportunities_a_survey

**S. Nakamoto (2008)** Bitcoin: A Peer-to-Peer Electronic Cash System [online].
[Accessed 22 September 2021]. Available at:
<https://www.ussc.gov/sites/default/files/pdf/training/annual-national-training-seminar/2018/Emerging_Tech_Bitcoin_Crypto.pdf>

**U. Bodkhe et al. (2017)** Blockchain for Industry 4.0: A Comprehensive Review [online].
[Accessed 23rd September 2021]. Available at:
https://ieeexplore.ieee.org/abstract/document/9069885

**Fakhar ul Hassan et al. (2020)** Blockchain and The Future of the Internet: A Comprehensive Review [online].
[Accessed 25th September 2021]. Available at: https://arxiv.org/pdf/1904.00733.pdf

**K Wüst et al. (2018)** Do you Need a Blockchain? [online].
[Accessed 27th September 2021]. Available at:
https://ieeexplore.ieee.org/abstract/document/8525392

**A. A. Monrat et al. (2019)** A Survey of Blockchain from the Perspectives of Applications, Challenges, and Opportunities [online].
[Accessed 27th September 2021]. Available at:
https://ieeexplore.ieee.org/abstract/document/8805074

**M. Kim et al. (2018)** Integrating Blockchain, Smart Contract-Tokens, and IoT to Design a Food Traceability Solution [online].

[Accessed 1st October 2021]. Available at: https://ieeexplore.ieee.org/document/8615007

**D. Pirker et al. (2021)** velink - A Blockchain-based Shared Mobility Platform for Private and Commercial Vehicles utilizing ERC-721 Tokens [online].
[Accessed 1st October 2021]. Available at: https://ieeexplore.ieee.org/document/9357605

**K. Yeow et al. (2017)** Decentralized Consensus for Edge-Centric Internet of Things: A Review, Taxonomy, and Research Issues [online].
[Accessed 3rd October 2021]. Available at: https://ieeexplore.ieee.org/abstract/document/8168250

**N. Szabo (2005)** Bit Gold [online].
[Accessed 3rd October 2021]. Available at: <https://nakamotoinstitute.org/bit-gold/>

**S. Haber & W. S. Stornetta (1991)** Cryptographically secured chain of blocks [online].
[Accessed 3rd October 2021]. Available at: <https://medium.com/@nehasoni1812/evolution-of-blockchain-f243f7509fe6>
https://coingeek.com/stuart-haber-and-scott-stornetta-how-our-timestamping-mechanism-was-used-in-bitcoin-video/

**V. Buterin (2014)** Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform [online].
[Accessed 3rd October 2021]. Available at:
<https://ethereum.org/669c9e2e2027310b6b3cdce6e1c52962/Ethereum_White_Paper_-_Buterin_2014.pdf>

**J. Melinek (2022)** Report: VCs Invested $33B in Crypto and Blockchain Startups in 2021 [online].
[Accessed 4th October 2021]. Available at: <https://blockworks.co/report-vcs-invested-33b-in-crypto-and-blockchain-startups-in-2021/>

**K. Roose (2021)** Why Did Someone Pay $560,000 for a Picture of My Column? [online].
[Accessed 4th October 2021]. Available at: <https://www.nytimes.com/2021/03/26/technology/nft-sale.html>

**J. Nielsen (1994)** 10 Usability Heuristics for User Interface Design [online].
[Accessed 4th October 2021]. Available at: < https://www.nngroup.com/articles/ten-usability-heuristics/>

**E. Kereiakes (2019)** Terra Money: Stability and Adoption [online].
[Accessed 12th October 2021]. Available at: < https://assets.website-files.com/611153e7af981472d8da199c/618b02d13e938ae1f8ad1e45_Terra_White_paper.pdf
>

**The European Parliament and of the Council (1998)** DIRECTIVE 98/69/EC [online]
[Accessed 22nd October 2021]. Available at: < https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CONSLEG:1998L0069:19981228:EN:PDF
>

**K. Sekniqi (2020)** Avalanche Platform [online].
[Accessed 12th October 2021]. Available at: < https://whitepaper.io/document/603/avalanche-whitepaper>

# 10 Glossary

**AI** - Artificial Intelligence

**AGI** – Artificial General Intelligence

**EVM** – Ethereum Virtual Machine

**JVM** – Java Virtual Machine

**DeFi** – decentralized finance

**NFT** – non fungible token

**IRL** – in real life

**VC** – venture capital

**dApp** – decentralized application

**OBD** – open board diagnostics