Use Case Examples of Ethereum Non-Fungible Tokens Tied to Assets Using ERC-4519

Javier Arcenegui, Rosario Arjona and Iluminada Baturone
Instituto de Microelectrónica de Sevilla (IMSE-CNM)
Universidad de Sevilla, CSIC
Seville, Spain
{arcenegui, arjona,lumi}@imse-cnm.csic.es

***Abstract*— Ethereum is a dynamic blockchain that grows every day thanks to a community that creates and decides on various of its issues. The community participates through Ethereum Improvement Proposals (EIPs) at many levels, from proposals that describe new standards for the creation of new tokens to proposals that define new ways for the Ethereum main network to generate new blocks. This paper describes how an Ethereum Request for Comments (ERC)-type EIP, the ERC-4519, was proposed last year to standardize the way to define non-fungible tokens (NFTs) representing assets that can generate their own Ethereum addresses and obey users and owners. Advantages provided by ERC-4519 in several use cases are illustrated. The examples show the facilities and the security improvements introduced in the management of both physical and digital assets by their owners and users. Particularly, use cases with physical assets such as IoT devices are illustrated.**

Keywords—Blockchain, Ethereum, EIP, smart contracts, non-fungible tokens (NFTs), IoT devices

I. Introduction

A blockchain is a chain of information blocks that are linked through cryptographic hash functions, so if even a single bit of information is changed in one of the blocks, this generates a different hash that corrupts the chain. Corruption is prevented because several nodes of the blockchain have a complete or partial copy of the entire chain and carry out a consensus to agree on the blocks to be added to the chain. An attacker has to corrupt the chain simultaneously in more than half of the nodes to be successful. Since this is difficult, the information written on the blockchain is considered immutable.

There are different types of blockchains depending on who can read or create transactions in the blocks of a blockchain. If everyone can create transactions or read data from the blockchain, it is considered public. If only a small group can do so, it is considered private. There are intermediate cases such as permissioned blockchains, where some of the blockchain can be accessed, and semi-public blockchains, where only some nodes can create transactions, but everyone can read the data of the blocks. This paper focuses on Ethereum, which is a public blockchain.

Ethereum technology provides a secure and decentralized platform for executing smart contracts [1]. Smart contracts are deterministic programs that run on the blockchain, so that all users of the blockchain can read their results, making their executions transparent. Among the blockchains capable of executing smart contracts, Ethereum is the most widely used blockchain in the world. Smart contracts allow the secure creation and management of tokens. Tokens are digital assets that are owned by blockchain participants. There are two types of tokens: fungible and non-fungible tokens (NFTs). Fungible tokens are usually employed as cryptocurrency to pay for a specific task or function. For example, a kind of computation or the consumption of a fixed amount of a product can be paid by fungible tokens. The fungible tokens used in a given application have all the same value. In the other side, non-fungible tokens (NFTs) are quite different among them and are generally associated with a unique digital asset. An NFT determines the properties of the asset as well as who owns it at any given time.

Ethereum, as a public blockchain, allows the community to improve its functionality through Ethereum Improvement Proposals (EIPs) [2]. The process of an EIP involves a series of steps. Initially, the EIP is in a "Draft" state. Once the EIP is robust enough, it evolves towards the "Review" stage. At this point, community members show their interest in the EIP and can suggest modifications. Once the EIP has been refined and discussed during a considerable amount of time, it moves to the "Last Call" stage. If no further changes are suggested during the timeframe specified by the editors, the EIP enters "Final" state and becomes an official EIP. If the community rejects the EIP, its status changes to "Withdrawn". If the EIP authors do not solve the flaws or do not incorporate the suggestions, it moves to a "Stagnant" state. From the later, the EIP can be revived and brought back to the "Living" state if someone proposes relevant modifications and goes on with the EIP development. This process facilitates interactions, encourages EIP authors to work in the development of their proposals, and ensures that the community could scrutinize and approve changes in a systematic and collaborative manner. If an EIP is abandoned, it goes to “Withdrawn” state.

There are six types of EIPs depending on their purpose: (1) The "Core" type of EIPs propose modifications to the intrinsic functionality of Ethereum. Hence, these EIPs require a consensus fork, since they aim to be implemented directly on the main network. (2) The “Meta” EIPs propose implementations for Ethereum prior to its core and, hence, often require community consensus also. (3) The “Networking” type of EIPs aim to enhance Ethereum's subprotocols (or peer-to-peer communication) with network protocols in Ethereum. (4) An “Interface” type is proposed to enhance client specifications or standards, as well as coding languages for smart contracts. (5) “Informational” EIPs propose designs or informational guidelines for Ethereum that do not include new functionalities. (6) Finally, the “Ethereum Request for Comments” or ERC type of EIPs propose token standards, libraries, or abstractions of Ethereum addresses.

This paper focuses on the ERC types, which are very relevant for defining tokens. Smart contract implementers or designers may ignore them, but it is simpler and recommended to follow them. ERC-20, which was created in the end of

This research was conducted thanks to Grant PDC2021-121589-I00 funded by MCIN/AEI/10.13039/501100011033 and the “European Union NextGenerationEU/PRTR”, and Grant PID2020-119397RB-I00 funded by MCIN/AEI/ 10.13039/501100011033.

2015, defines the most used standard for a fungible token [3]. The first standard proposed for a non-fungible token was the ERC-721, which was created in the beginning of 2018 [4]. As an improvement of ERC-721, the ERC-4519 became official in the end of 2022 [5].

This paper is organized as follows. In Section II, the concept and evolution of non-fungible tokens in Ethereum as well as use cases found in the literature are summarized. The ERC-4519 is detailed in Section III, introducing the three possibilities of using ERC-4519 NFTs. In Section IV, several examples of use cases with ERC-4519 NFTs are given together with a more detailed description of some of them. Finally, in Section V, conclusions are exposed.

II. Non-Fungible Tokens in Ethereum

Non-Fungible Tokens (NFTs) are designed to identify unique assets, which usually are possessions like houses, cars, artworks, and so on. The first definition of an NFT in Ethereum, and the most widely used, is the ERC-721 [4]. It describes the basic functionalities that an NFT should have, such as its identifier, owner, or who can manage the NFTs of an owner.

The ERC-1155, created in the middle of 2018, is the first EIP that unifies fungible and non-fungible tokens [6]. The idea is to recreate a gamification of tokens, for example, in a "shooter" game, the player owns a specific weapon that serves as an NFT, but uses bullets as fungible tokens. None of these proposals establishes the entity of a user, but the owner is the only entity with the capacity to use it. However, in real life, owners can lend or rent their possessions, allowing others to use their assets without losing possession of them. Moreover, these proposals do not include any attribute to actively identify an asset. The ERC-5006 is an extension of the ERC-1155. It establishes an NFT adapted to NFT rentals, assigning the value of the user [7]. However, no authentication protocol between the user and the asset is established. EIP-4973 and EIP-5114, currently at Review state, discuss a standard that links an NFT with an Ethereum address, defining a property that cannot be transferred, but they do not specify if other users can make use of these tokens [8]-[9].

The ERC-4519 is an official proposal that includes user (besides owner) management, Ethereum addresses to actively identify assets that can perform blockchain transactions, and secure mutual authentication protocols between owners and assets, and between users and assets [5]. Preliminary works can be seen in [10]-[12]. The work in [10] does not define NFTs but proposes using a public key for a physical device considered as an asset (obtained from a private key stored secretly inside the device) without defining NFTs. Authors of ERC-4519 introduce NFTs tied to IoT devices as assets in [11] and [12].

Regarding use cases of Ethereum NFTs, the work in [13] utilizes augmented reality (AR) in conjunction with ERC-721 and ERC-1155 to create NFTs that allow for property rights preservation. In [14], an ERC-1155 NFT was utilized to encapsulate code of Open-Source and Commercial Software Licensing in a smart contract. This creates an NFT for the code in the smart contract, which must be paid by the client who wishes to use the software. In [15], a model was developed for managing real estate assets as ERC-721 NFTs, providing proof of ownership and specifying a system for exchanging or selling assets. The solution presented in [16] proposes the use of ERC-721 NFTs for improving or mitigating communication and storage issues of healthcare information such as drug tracing. The work presented in [17] allowed users to upload encrypted content and mint it into ERC-721 NFTs so that other users can access the NFTs' content by requesting a purchase or a license.

None of the proposals in the literature employs ERC-4519 NFTs due to the novelty of this EIP. In the following sections, the ERC-4519 will be described and use cases will be presented to show the advantages of NFTs based on it.

III. ERC-4519

The interface for NFTs introduced by ERC-4519 allows representing physical or digital assets that can generate their own Ethereum addresses and obey their users and owners if they authenticate conveniently. Table 1 illustrates the new main attributes (named *user* and *asset*) of ERC-4519 in comparison with ERC-721 (with only *owner*).

ERC-721 defines only the token identifier and the address of its owner. There is no mention to NFT users and there is no bound between the asset and the token apart from the token identifier. In this sense, it is said in Table 1 that ERC-721 allows describing basic NFTs. In ERC-4519, the attributes *user* and *asset* are introduced as addresses, besides *owner*. If *user* and *asset* attributes are not needed in the application, the NFT can be represented by an ERC-721, which allows the management of the NFT owners. However, if the NFT requires also user management and/or requires a bound between the asset and the NFT, the NFT should be represented by an ERC-4519. The latter applications are more detailed in the following.

A. NFT with User Management

Adding the attribute *user* allows ERC-4519 representing assets that can be lent, shared, or rented to users, without the owner (or owners) losing ownership of the token at any time.

New functions are included in ERC-4519 to enable user addition to the NFT. The function "setUser" can only be executed by the NFT owner. If the address provided to the function "setUser" is empty or is equal to 0, it is understood that no user is assigned. Additionally, there are other functions such as "userOf" to determine who is the user of a token, "userBalanceOf" to know how many tokens a user has, and "userBalanceOfAnOwner" which specifies how many tokens a user has with a specific owner.

In these applications, the NFT created (by the function "createToken") can be at one of two states: "notAssigned" and "userAssigned". Figure 1 shows the state diagram of these NFTs together with the relevant functions.

B. Asset Tied to NFT without Users

Adding the attribute *asset* allows ERC-4519 representing assets that are tied to their tokens, since the assets can

TABLE I. Attributes Employed by ERC-721 and ERC-4519

owner	user	asset	Applications
Yes	No	No	Basic NFT (ERC-721)
Yes	Yes	No	NFT with user management (ERC-4519)
Yes	No	Yes	Asset tied to NFT without users (ERC-4519)
Yes	Yes	Yes	Asset tied to NFT with user management (ERC-4519)

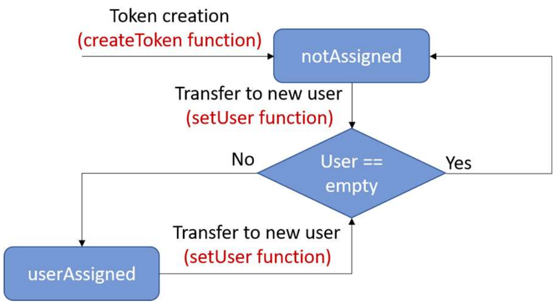

Fig. 1. Flow chart of the token states with user address defined and asset address undefined.

participate in Ethereum with their own addresses. Therefore, the asset can query the smart contract associated with its token and obtain the associated data. Moreover, the asset can identify and authenticate the owner of its token and establish a secure communication channel after carrying out a key exchange protocol with the owner. If the asset is a physical asset such as an IoT device, it should have enough computing and power resources to generate its own Ethereum address and to authenticate its owner.

Besides the attribute *asset*, the attribute *hashK_OA* can be used to store the hash of the shared key agreed between the asset and the owner, so that they can check if they have calculated the shared key correctly. The attribute *dataEngagement* can also be used to store the public information that the asset and owner need to generate the same key.

Other optional attributes are *timestamp* and *timeout*. The first can be used to know the last time that the asset connected to its NFT. The second, to define the maximum delay time that the asset can be without connecting again with its NFT. They serve as proofs of liveness of the asset, that is, allow checking if the asset is functioning properly and has Internet connection. When the ERC-4519 was launched, there was no other EIP with this kind of attributes.

New functions are included in ERC-4519 to enable the mutual authentication between assets and owners. With the function "startOwnerEngagement", the owner sends the hash of the key to share and the public information needed by the asset to obtain the same key. With the function "ownerEngagement", the asset can confirm that it has generated the same key.

The function "updateTimestamp" allows the asset to update the value of the timestamp, whenever it refreshes its link with its NFT or it has been inactive for a long time. The

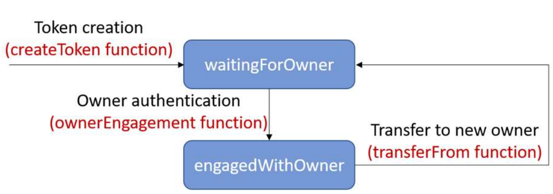

Fig. 2: Flow chart of the token states with asset address defined and user address undefined.

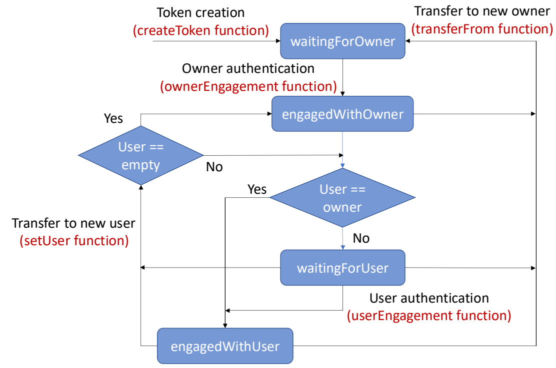

Fig. 3: Flow chart of the token states with asset and user addresses defined.

function "setTimeout" specifies how long the asset can be inactive without being considered faulty, and the function "checkTimeout" verifies if the asset is still tied to its NFT. Additionally, there are other functions like "tokenFromBCA", which determines the identifier of an asset from its Ethereum address, and "ownerOfFromBCA", which determines the owner of the asset from its Ethereum address.

In these applications, the NFT created can be at one of two states: "waitingForOwner ", whenever no owner has been authenticated, and "engagedWithOwner", once owner and asset have authenticated each other. Figure 2 shows the state diagram of these NFTs together with the relevant functions.

C. *Asset Tied to NFT with User Management*

Both attributes *asset* and *user* should be employed by ERC-4519 to represent assets tied to their tokens and with users. In these applications, besides the functions specified above, the new functions “startUserEngagement” and “userEngagement” are included. They operate similarly to “startOwnerEngagement” and “ownerEngagement”, but with the user. Additionally, the function “userOfFromBCA” can be used to retrieve the user of the token from the address of the asset tied to the token.

The attribute *hashK_UA* can be used to store the hash of the shared key agreed between the asset and the user, so that they can check if they have calculated the shared key correctly. The attribute *dataEngagement* can be used to store the public information that the asset and user need to generate the same key. It is not necessary to add another variable for temporal data in this case, since the same variable *dataEngagement*, being temporal, will never be used to authenticate both the owner and the user simultaneously.

In these applications, the NFT can be at two new states, as shown in Figure 3, “waitingForUser”, whenever the asset is engaged with the owner but has no authenticated user, and “engagedWithUser”, once the owner and user are authenticated with the asset.

IV. EXAMPLES OF USE CASES

Several use cases are illustrated in this Section to point out the advantages of using ERC-4519. The same combination of attributes described in the previous section and shown in Table 1 are considered. One example within each category is explained more in detail.

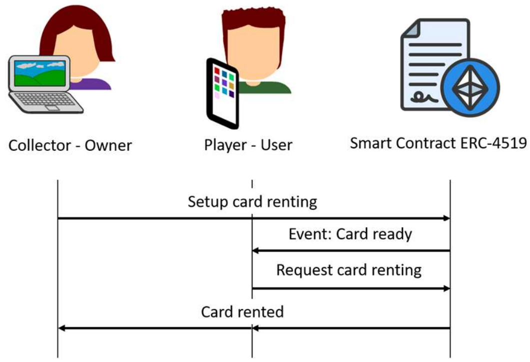

Fig. 4: The process of renting a card in a trading card game.

A. NFT with User Management

Many examples of use cases that arise from the inclusion of the user field in an NFT are in the digital art market. ERC-4519 NFTs representing digital artworks enable their owners to digitally lend or rent them at expositions and festivals.

Another example is a music platform that uses ERC-4519 to represent songs. The songs can be sold and their owners can share their licenses with family and friends temporarily.

Let us explain more in detail these use cases with the use of ERC-4519 in a trading card game. In these games, there are cards that are common, uncommon, rare, epic, legendary, and unique. Each type of card is an NFT described by an ERC-4519. Each NFT also has a series of fields that determine the card features besides the card type. This is done through a 256-byte variable, where the bits are parsed to provide a multitude of unique powers and abilities described at a DApp server.

The game creator limits the number of cards that can be created for each type, except for the common cards, which can be unlimited. Therefore, when a digital card pack is opened, there is a certain probability of receiving specific cards. In this type of games, there are two types of consumers: (1) players, who use the cards exclusively to play, seeking to have the most efficient virtual set of cards in order to gain, and (2) collectors, who want to have all the possible cards, making harder to obtain the most exclusive cards.

Traditionally, this type of games only allows players to use the cards they own. Hence, if players want to play with a legendary or unique card, they have to buy it at a very high price. The advantage of adding users to the cards is that collectors, who do not play with these cards, can rent them without losing their ownership.

A card owner (collector) can put cards for renting by creating an offer that includes the price and rental period. The offer can be given to the first user (player) who accepts the terms of the offer and pays the specified amount during the transaction. Then, the smart contract assigns the user field to the payer for the specified duration in the offer. Owners can also specify who they want to rent the card to, which would result in an automatic rejection of the offer by any other user. Interested users can subscribe to an event that notifies them of the availability of cards for renting, or they can review available offers in a card marketplace.

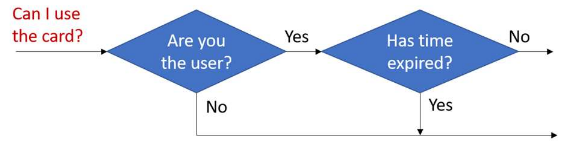

Fig. 5: Checking if a card can be used or not.

Once the rental period expired, the owner can create a new offer for renting the card, or withdraw the money and set the user field to zero.

Figure 4 summarizes the process of renting a card and Figure 5 shows how a card can be used or not.

Additionally, the owner also has the option of directly lending the card without specifying a rental offer but only setting a loan period. The same variable in the NFT defines the loan or the rental period. During this period, the owner cannot make use of the card, and the card can only be returned from the user by setting the user field to zero.

B. Asset Tied to NFT without Users

Many examples of use cases that arise from the inclusion of the asset field in an NFT are in the field of luxury smart assets, like smart watches, smart rings, and other smart wearables. ERC-4519 representing these assets make them only respond to their original owners and may send location data if they are attempted to be manipulated without authentication.

This type of NFTs can also be used for creating a wallet of a decentralized identity linked to a physical device, so that the owner of the physical device can prove his/her identity with the data stored in the device. If the device is stolen, the thief will not be able to use it because he/she will not be able to authenticate with the device.

Let us explain more in detail these use cases with the use of ERC-4519 to link firearms with tokens. Firearms are increasingly equipped with advanced technology, so it's not far-fetched to think that a firearm could be connected to the Internet to share its location, check its status, implement security locks and more. Due to the dangers associated with firearms, a relevant advantage of this solution is that only the owner of the firearm can use it. This prevents that family members or malicious individuals could use firearms stored at home, thereby avoiding serious accidents and criminal actions. It is important to notice that this use case should not be confused with a firearm license or registration, which is an administrative procedure that depends on the legislation of each state or country.

When an individual (owner) acquires a firearm (asset), he/she also acquires the token that is linked to the firearm. In order to use the firearm, the owner and asset must authenticate themselves and establish a secure communication channel by sharing a cryptographic key. Subsequently, when the owner wishes to use the firearm, he/she must send an encrypted message to the ERC-4519 specifying the period of time during which the firearm will be used. To do this, the owner sends the hash of the message that establishes the time of use along with a nonce provided by the firearm off-chain. The smart contract verifies that only the owner can write in this field, and if so, it will update it. In this way, the firearm can check

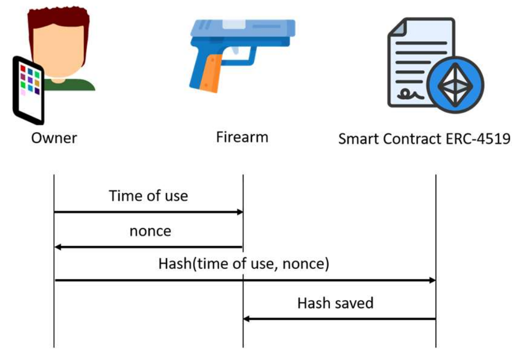

Fig. 6: The process of using a firearm.

whether it can be used or not based on the result of the operation performed by the firearm, which should obtain the same hash. This process is illustrated in Figure 6.

Likewise, if the owner stops using the firearm before the time of use expires, the owner can write to the smart contract that he/she has finished of using the firearm. This way, the firearm stops working. Figure 7 shows how a firearm can be securely used or not.

C. *Asset Tied to NFT with User Management*

Many examples of use cases arise from the inclusion of the asset and user fields in an NFT. In the context of regulation compliance and risk prevention, a measuring device (of levels of radioactivity, harmful emissions, etc.) calibrated by a company is assigned to an inspector as a user. If the device is representing by an ERC-4519, only authorized inspectors can access the device and all the measurements are conveniently registered.

Another example is a software license that can only be used by the owner or lent to other users through the assignment of specific user fields. This approach allows for accurate tracking of who is using the software or license.

In a crowdfunding platform, investors, instead of lending money, can lend their smart assets associated with NFTs in order to generate profits through an ERC-4519 defining a smart contract. Examples of smart assets are vehicles and homes.

Let us explain more in detail the use of ERC-4519 in the case of vehicle renting. A vehicle is a valuable asset that is commonly rented for either short or long periods of time. As vehicles increasingly have more computing power, it's not surprising that they can be connected to Internet and to Ethereum. Therefore, an ERC-4519 token can be linked to the Ethereum address generated by the gateway in the vehicle. Obviously, there is the need of a server that associates tokens with specific vehicles and users with specific drivers, in order to preserve the privacy and ensure that both the vehicles and the drivers comply with the state traffic regulations and, hence, insurance companies will cover any accidental damage that may occur.

Other variables that the ERC-4519 token should include are the *rentalPrice* (to establish the rental price), *rentalTime* (to establish the rental time), *rentalDeposit* (to set the money to repair damages caused to the vehicle that are not covered by the insurance), and *fuelDeposit* (to adjust the fuel deposit difference to pay when returning the vehicle).

Once the owner of the vehicle and the driver agree on the rental conditions, they execute the function “setupRental” with the rental time, rental deposit, and rental price established. This function sends an event that will be read by the user. The user can start the engagement with the vehicle by executing the function “startUserEngagement” with enough Ether value to pay both the rental and the deposit. The vehicle, instead of executing “userEngagement”, will execute “rentalAccepted”, which, in addition to executing “userEngagement”, will store the vehicle's deposits, thus avoiding a message from the vehicle to Ethereum. Figure 8 shows this process.

If, before the rental period ends, the owner of the vehicle and the driver agree, they could renew the rental. To do this, the owner executes “setupRenewalRental” with the new rental time and the new price that the user must pay with “renewalRental”. The latter function sends an event to the vehicle for it to update the new renewal time. This process is shown in Figure 9.

Once the rental time has elapsed, the vehicle will execute “contractTermination”, which stores the fuel deposit difference and the vehicle's location before the vehicle is turned off, avoiding possible risks. The owner must check if there is any damage and if the vehicle has been left at the agreed-upon location If everything has gone well, the owner withdraws his/her money and the contract returns the deposit to the user. Otherwise, the owner establishes the cost of damage repairs or of taking the vehicle to the agreed upon location, can recover this money from the deposit and return the rest to the user.

V. CONCLUSIONS

The ERC-4519 offers a number of advantages over the traditional ERC-721. Its flexibility in selecting which attributes to integrate into each project allows it to adapt to the specific requirements of each project. This paper has presented several application examples that demonstrate these benefits, and has elaborated on some of them. However, the possibilities offered by the ERC-4519 are numerous and include NFTs associated with physical or digital assets,

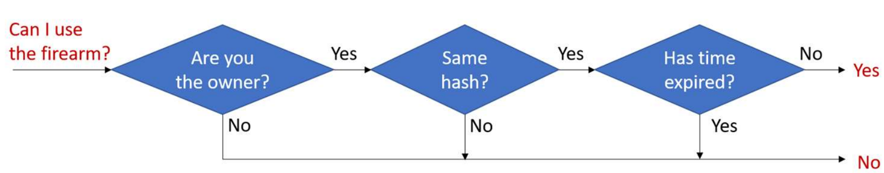

Fig. 7: Checking if a firearm can be used or not.

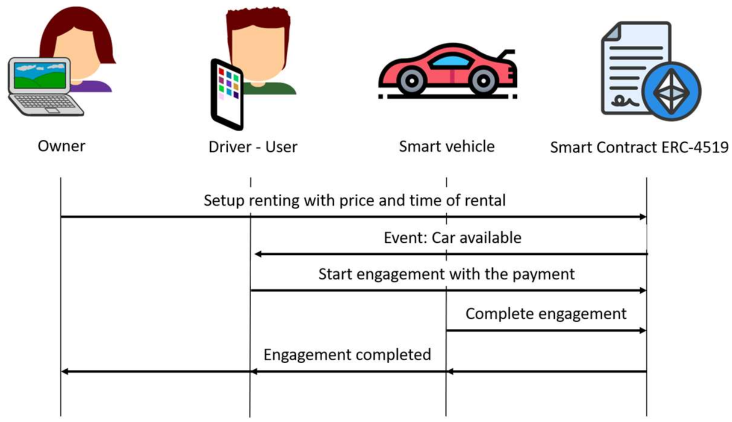

Fig. 8: The process of renting a vehicle.

whether real or virtual. In addition, the ERC-4519 serves as a contribution to the circular economy since it facilitates the ability to share goods.

References

[1] V. Buterin (2023, March 11). “Ethereum whitepaper” [Online Available] https://ethereum.org/en/whitepaper/

[2] M. Becze et al. (2015, October 27). “EIP-1: EIP Purpose and guidelines” [Online Available] https://eips.ethereum.org/EIPS/eip-1

[3] F. Vogelsteller and V. Buterin (2015, November 19). “ERC-20: Token standard” [Online Available] https://eips.ethereum.org/EIPS/eip-20

[4] W. Entrinken, D. Shirley, J. Evans and N. Sachs (2018, January 24). “ERC-721: Non-Fungible Token standard” [Online Available] https://eips.ethereum.org/EIPS/eip-721

[5] J. Arcenegui, R. Arjona, R. Román, and I. Baturone (2021, December 3). “ERC-4519: Non-Fungible Token tied to physical assets” [Online Available] https://eips.ethereum.org/EIPS/eip-4519

[6] F. W. Radomski et al. (2018, June 17). “ERC-1155: Multi token standard” [Online Available] https://eips.ethereum.org/EIPS/eip-1155

[7] Lance, Anders and Shrug (2022, August 1). “ERC-5006: Rental NFT, NFT user extension” [Online Available] https://eips.ethereum.org/EIPS/eip-5006

[8] T. Daubenschütz (2022, April 1). “ERC-4973: Account-bound tokens” [Online Available] https://eips.ethereum.org/EIPS/eip-4973

[9] M. Zoltu (2022, May 30). “ERC-5114: Soulbound badge” [Online Available] https://eips.ethereum.org/EIPS/eip-5114

[10] M. N. Islam and S. Kundu, “Poster Abstract: Preserving IoT privacy in sharing economy via smart contract,” IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI), Orlando, FL, USA, 2018, pp. 296-297.

[11] J. Arcenegui, R. Arjona, and I. Baturone, ‘‘Secure management of IoT devices based on blockchain non-fungible tokens and physical unclonable functions,’’ in Proc. Int. Conf. Appl. Cryptogr. Netw. Secur. Cham, Switzerland: Springer, 2020, pp. 24–40.

[12] J. Arcenegui, R. Arjona, R. Román, and I. Baturone, “Secure combination of IoT and blockchain by physically binding IoT devices to smart nonfungible tokens using PUFs,” Sensors, vol. 21, no. 9, pp. 1-23, Apr. 2021.

[13] M. Duguleană and F. Gîrbacia, “Augmented reality meets Non-Fungible Tokens: Insights towards preserving property rights,” IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct), Bari, Italy, 2021, pp. 359-361.

[14] M. Madine, K. Salah, R. Jayaraman and J. Zemerly, “NFTs for open-source and commercial software licensing and royalties,” in IEEE Access, vol. 11, pp. 8734-8746, 2023.

[15] M. K. Hari, A. Agrawal, R. Bhatia, A. Bhatia and K. Tiwari, “T-PASS: A blockchain-based NFT enabled property management and exchange system,” International Conference on Information Networking (ICOIN), Bangkok, Thailand, 2023, pp. 140-145.

[16] S. S. Bala, B. Prasad, A. M. Jadhav and K. Wasnik, “Online storage, retrieval & authentication of healthcare documents using Ethereum blockchain,” IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS), Pune, India, 2022, pp. 1-6.

[17] M. Madine, K. Salah, R. Jayaraman, A. Battah, H. Hasan and I. Yaqoob, “Blockchain and NFTs for time-bound access and monetization of private data,” in IEEE Access, vol. 10, pp. 94186-94202, 2022.

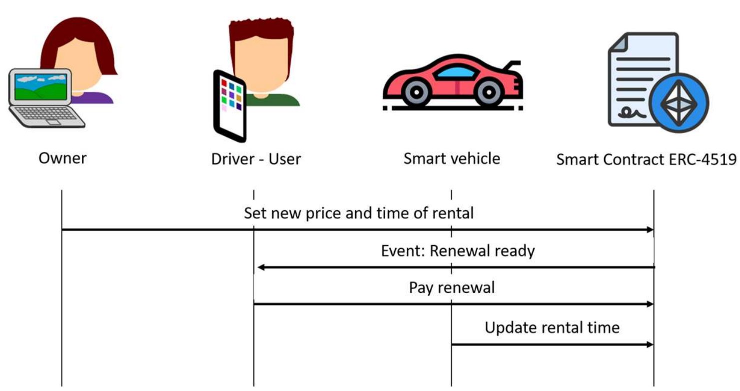

Fig. 9: The process of renewing the rental.