POLITECNICO DI TORINO
Repository ISTITUZIONALE

Improving driving behavior: a blockchain-based gamification system

Original
Improving driving behavior: a blockchain-based gamification system / Butera, A., Romani, N., Gatteschi, V.. - ELETTRONICO. - 3791:(2024). (DLT 2024: 6th Distributed Ledger Technologies Workshop Turin (ITA) May, 14-15 2024).

Availability:
This version is available at: 11583/2990920 since: 2024-07-16T19:58:40Z

Publisher:
CEUR

Published
DOI:

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

27 June 2026

Improving Driving Behavior: A Blockchain-Based Gamification System

Alberto Butera[1,†], Noemi Romani[1,†] and Valentina Gatteschi[1,*,†]

[1]*Department of Computer and Control Engineering, Politecnico di Torino, Turin, Italy, Corso Duca degli Abruzzi 24, 10129, Turin, Italy*

Abstract

In an era of increased vehicles usage, traffic accidents caused by improper driving behavior have also risen. Therefore, there is a growing need to improve road safety by promoting a correct driving style and counteracting accidents with innovative strategies. This paper presents the implementation of a decentralized system, based on blockchain, that utilizes gamification principles to incentivize safe driving. The proposed solution aims to reduce the number of traffic accidents, through gamification, and could serve as a tool for insurance companies to create customized insurance policies based on users' driving behavior. The system leverages on a reward mechanism that incentivizes the most careful drivers with tokens, creating a virtuous circle that benefits all people, drivers and non-drivers alike. The article proposes a working prototype platform and describes its implementation details, demonstrating its potential to incentivize a prudent driving style and contributing to a future where good driving is the norm, rather than the exception.

Keywords

Blockchain, Smart Contract, Gamification, Personalized Insurance, Safe Driving, Vehicle, Driving behaviors

1. Introduction

Each year, more than 1.35 million people die in road traffic accidents worldwide [1], with 37.806 of deaths in the United States, in 2016 [2]. The WHO's *Global status report on road safety 2018* [1] reported that road traffic injuries are the leading killer of people aged between 5 and 29, especially for pedestrians, cyclists and motorcyclists living in developing countries. The majority of traffic accidents can be blamed to human factors [3, 4], especially to aggressive driving [5].

Letting drivers become aware of their driving style, through the monitoring and logging of driving events, could reduce aggressive driving behaviors [6] and, consequently, avoid 20% of road traffic accidents [7]; furthermore, exploiting incentives or penalties could improve this virtuous cycle [8]. Apart from the aspects related to drivers' security, a reduction of aggressive driving behavior could also result in a lowering of vehicle consumption and gas emissions [9, 10] as aggressive driving has been estimated to increase fuel consumption by around 40% [11].

Technological advancements made it possible to detect driving styles or aggressive driving behavior by relying on acceleration data, among others: in fact, non-aggressive driving events generally generate low(er) accelerations, whereas aggressive ones are characterized by high(er) readings acquired from accelerometers.

Historically, accelerometer data started to be obtained through black-boxes, ad-hoc hardware installed on the vehicle able to collect its acceleration (by exploiting Inertial Measurement Units, IMUs), location and speed (through Global Positioning System, GPS), and possibly gather additional information from the on-board diagnostics (OBD).

As time passed, technological advancements made it possible to acquire this type of information by relying on smartphones' sensors. An advantage of smartphones is that they do not require additional

DLT2024: 6th Distributed Ledger Technologies Workshop, May, 14-15 2024 – Turin, Italy

*Corresponding author.

†These authors contributed equally.

alberto.butera@polito.it (A. Butera); noemi.romani@polito.it (N. Romani); valentina.gatteschi@polito.it (V. Gatteschi)

https://staff.polito.it/valentina.gatteschi (V. Gatteschi)

0009-0002-2175-1471 (A. Butera); 0000-0001-6075-6430 (V. Gatteschi)

© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

CEUR-WS.org/Vol-3791/paper1.pdf

costs to be paid by the driver (since he/she does not have to buy dedicated hardware), they have access to communication networks needed for data transfer, and can even process data onboard, as they generally have more powerful processors than those contained in black-boxes.

Applications able to detect driving behaviors could indeed generate new revenue streams [12], especially in fleet management and insurance telematics fields.

Regarding fleet management, the objective is to collect and monitor data on the location and current conditions of corporate vehicles to inspect drivers' behavior and motivate them, e.g, to reduce travel time, lower fuel consumption, etc.

For what it concerns insurance telematics, the above devices are increasingly exploited in Usage-Based Insurance, a new form of insurance, in which the cost of a car insurance policy is computed not only on driver demographics data such as age, vehicle characteristics and location, but also includes information on the annual kilometers driven (the so-called Pay-As-You-Drive insurance), or on the behavior of the driver (the so-called Pay-How-You-Drive insurance) [13]. In particular, in Pay-How-You-Drive insurance, a score is assigned to the driver, based on the amount (and, in some case, of the type) of harsh events generated in a selected period. In some cases, the driver can also receive a feedback after each driving session or even during the drive, on how safe was/is her driving style, and on possible corrective measures.

In this work, we take a step forward with respect to insurance telematics, and propose a decentralized system, based on blockchain technology, to support gamification with the objective of promoting good driving behaviors. The proposed system is composed of a mobile application, used to record and process driving-related data, and a smart contract, devoted to managing the gamification logic. Our solution uses blockchain to ensure complete transparency in the competition's standings, final scores, entry fees, and prize money. Smart contracts automate functions when conditions are met, guaranteeing that winners receive their prizes when the game ends. This level of certainty and transparency is unattainable with centralized systems, where data could be modified. Moreover, to the best of our knowledge, no other works exist, that address driving behaviors using gamification. Our system could be either exploited by end users in a decentralized way, or could be easily integrated in more complex Pay-How-You-Drive insurance applications.

The rest of the paper is organized as follows: Section 2 presents relevant research works in the context of this paper. Section 3 provides an overview of the proposed system, whereas Section 4 discusses its costs and limitations. Finally, conclusions and future works are reported in Section 5.

2. Related works

This Section reports the results of the analysis of the state of the art and solutions identified in the literature for blockchain-based car insurance solutions tailored to drivers' driving styles. Additionally, we compare the identified solutions with our implementation, highlighting the differences and novelties introduced by our work. We excluded works based on centralized architectures for two reasons:

- blockchain offers advantages such as trust, transparency, and accessibility that are difficult to achieve with centralized solutions. This is why we chose a decentralized architecture.
- Since we have chosen a blockchain-based solution, it is more useful to compare our work with other works that use blockchain, rather than those based on centralized architectures.

As just mentioned, all identified works are blockchain-based. However, the proposed architectures utilize different blockchain frameworks based on implementation choices. For instance, [14] offers a consortium/private blockchain-based solution developed with Hyperledger Fabric. This solution has advantages in terms of cost, throughput, and customization. However, its limited number of nodes and node owners restrict the traditional advantages in terms of decentralization, making it more similar to a centralized solution. The solution presented by [15] describes an architecture based on the concept of cross-chain, that is, a set of blockchains (even of different types and with different characteristics) connected by a relay blockchain that allows interoperability to store and exchange data. The authors

mention WeCross as a cross-chain platform to be used to develop their solution. Although a cross-chain strategy provides network actors with greater flexibility in choosing which blockchain to use, it also makes the entire architecture technically more complex and more exposed to security risks due to operations such as synchronization between different blockchains. In [16], a comparable approach was taken, utilizing Tendermint as the basis for the proposed architecture, a framework that allows for the construction of customizable, modular, and Cosmos Network-compatible blockchains, which differ from classical public blockchains such as Ethereum in that they support the concept of asynchrony. Furthermore, the consensus mechanism, which differs from Ethereum's, allows for reduced transaction costs and latency while still maintaining decentralization. However, similar to the previous solution, this approach also requires higher development complexity (developing the blockchain in addition to the smart contracts), which inevitably increases exposure to bugs and security risks. Finally, works such as [17, 18, 19], including ours, leverage the public Ethereum blockchain to implement their solution. This design choice simplifies development by allowing programmers focus solely on developing smart contracts and implementation strategies. Additionally, Ethereum is currently the most widely used blockchain for coding decentralized applications, providing ample developer support.

When managing data privacy, it is crucial to compare various solutions for protecting sensitive driver data. Blockchain technology is a solution that does not allow data to be deleted or modified once it is stored in it, and one of the most important features of public blockchains is that they allow everyone to read the data. These properties ensure reliability, transparency, and accessibility but compromise the privacy of the data. Therefore, if sensitive data is present, appropriate protection techniques must be implemented. For instance, in works such as [14, 18, 16, 19], data is encrypted with suitable algorithms before being stored on the blockchain, making it readable only to those with the private keys for decryption. Furthermore, [14, 19] add an additional layer of security by leveraging Zero Knowledge Proof (ZKP) to allow users to prove they have the right informations for the requested tasks, such as their identity in the case of authentication, without sharing any sensitive data. In [17], the authors propose a privacy protection strategy based on the use of multiple temporary addresses. In general, an address is considered pseudonymous because the identity of its owner is unknown. However, since it is fixed, it is possible to make assumptions and extrapolate insights that allow tracing back to the identity of users. To partially solve the problem associated with pseudonymizing, different and temporary addresses can be used. The solution proposed by [15] involves aggregating data and working with the resulting aggregations, making it difficult or impossible to apply the inverse function to trace back to the original data. Finally, in our solution we distinguish between sensitive and non-sensitive data, storing only the latter on the blockchain for ranking drivers. Sensitive data is protected and kept private within a centralized database. It is read exclusively to populate the mobile app user page for each individual user.

An important aspect to consider when comparing our solution with the existing ones is how to evaluate driving. The cited works mainly use two methods: statistical analysis of parameters obtained during driving and calculation of a single score. The first method, used in [18, 19, 15], involves extrapolating statistical data from values received from drivers/vehicles, such as acceleration, braking, steering, and timing, to evaluate users' driving. The second method, used both in our work and in [17, 16], aggregates the parameters obtained during driving through mathematical functions to calculate and assign a score to users. Additionally, a logistic regression model is utilized in [14] to calculate the driving evaluation score.

Our proposal introduces a new concept of gamification, which we have not identified in any other state-of-the-art work. The idea is to create a competition among drivers to promote safe driving by ranking them based on their scores during drives and rewarding the best ones with prizes (that, in the current implementation, are represented by tokens, but, should the proposed solution be adopted by insurance companies, could result in better/cheaper insurance policies). By incentivizing drivers to adopt a safer driving style, we can reduce the risk of traffic accidents. In addition, we have developed a mobile application that simplifies users' interaction with our solution and allows them to automatically capture and send driving data to the gamification unit.

Table 1 summarizes the comparison between the various solutions and ours described above.

Table 1
Comparison among blockchain-based solutions for personalized car insurance

Solution	Architecture	Data Privacy	Driver Evaluation	Gamification
[14]	Hyperledger/Consortium BC	Encryption/ZKP	Logistic Regression	No
[17]	Ethereum/Private BC	Multiple addresses	Score-based	No
[18]	Ethereum	Encryption	Statistics-based	No
[16]	Tendermint	Encryption	Score-based	No
[15]	Cross-chain/WeCross	Data aggregation	Statistics-based	No
[19]	Ethereum	Encryption/ZKP	Statistics-based	No
Our	Ethereum	Limited Data Access	Score-based	Yes

3. System's description and functioning

In order to better understand the functioning of the developed system, it could be worth considering the activity diagram shown in Figure 1.

Since the gamification mechanism is guaranteed by the blockchain, the user interested in joining a game needs an Ethereum wallet. Hence, after the login, in case he/she is not in possession of a wallet, a new one is created and added to the user's profile. Then, in case the user is not already involved in a game, he/she can decide whether to join an existing one, or create a new one. After this step, he/she could use the mobile app to insert details on the trip's destination (in this case, the mobile app displays the route), and he/she could start the trip. At the end of the trip, the number (and type) of harsh events triggered during the trip is computed, and this information is used to update the users' profile/score.

Figure 2 shows the system's architecture. As it is possible to see, smartphones' sensors are used to retrieve data related to the users' driving behavior. The computation is performed both on the smartphone itself (orange boxes), both on the blockchain (blue box). The developed modules/components are the following:

1. Sensors raw data acquisition/transformation module: this module acquires raw data from the smartphone sensors, and transforms them;
2. Events classification module: this module, starting from the transformed data, identifies and classifies driving events;
3. Score computation: this module computes a score for the just-ended trip, based on the identified harsh driving events;
4. Smart contract: the smart contract contains the gamification logic, and stores the scores obtained by the users involved in the game.

In the following, each module is described in detail.

3.1. Sensors raw data acquisition/transformation module

This module relies on smartphone's sensors and retrieves acceleration and gps readings. In particular, accelerometer readings are acquired and then pre-processed through the gravity readings detected by the gravity sensor (in the data transformation phase), whereas the rotation sensor's readings are exploited for determining the device's orientation (more details will be provided in the following).

The accelerometer is a motion sensor that measures the acceleration force (in m/s^2) that is applied to a device on all the three physical axes (x, y, and z), including the force of gravity. In particular, the three axes, when the device is held in the default orientation, are the following:

- The x axis, that is horizontal. This axis is used to read lateral acceleration;
- The y axis, that is vertical and points up. This axis is used to read the force of gravity, as a stationary device has an acceleration value detected on the y axis of +9.81 m/s^2;

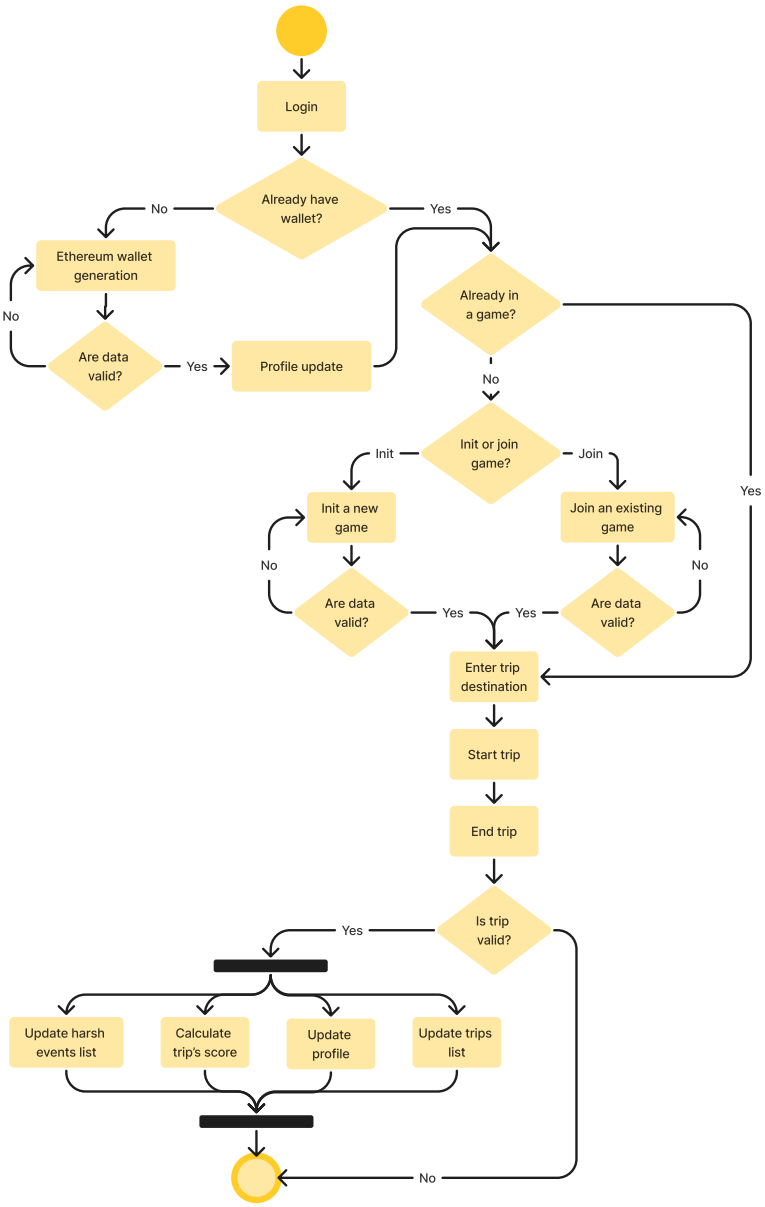


Figure 1: Activity diagram describing the behavior of the system.

- The z axis, that points toward the outside of the screen face: in this system, coordinates behind the screen have negative z values. This axis is used to read frontal acceleration.

Based on the three readings acquired through the accelerometer, it is possibile to identify whether an event is harsh or not. In particular, in the case of harsh events, accelerometer readings along one or more axes are higher than those acquired during non-harsh events. An example is displayed in Figure 3.

Acquired raw data are then processed by using low-pass and high-pass filters. The objective of this transformation phase is to eliminate gravitational forces and reduce noise. In particular, for the application of low- and high-pass filters, the gravity sensor – a three-dimensional vector indicating the

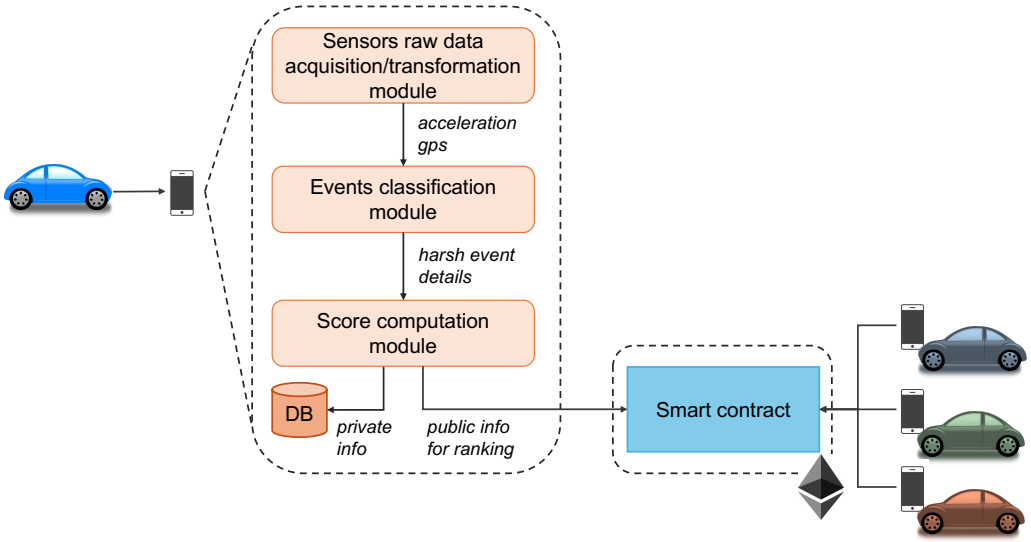
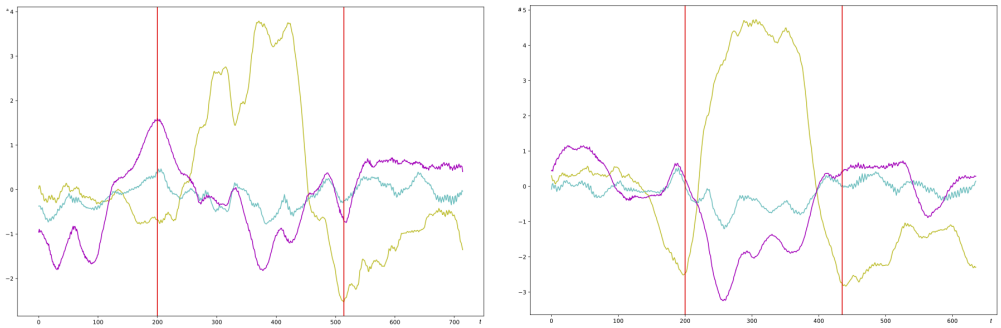


Figure 2: Architecture of the realized system



(a) (b)

Figure 3: Acceleration data related to a non-aggressive (a) and an aggressive (b) cornering event acquired in a roundabout using the mobile app. The vertical lines represent the start and end time of the event, respectively. x-axis shows time (around 6-7 seconds per chart), y-axis shows acceleration (from around $-0.3g$ to around $0.5g$). Yellow line: lateral acceleration; Magenta line: frontal acceleration; Cyan line: vertical acceleration.

direction and magnitude of gravity – is used as in equations 1 and 2:

$$filtered\_gravity_a = \alpha \cdot gravity_a + (1 - \alpha) \cdot acceleration_a \tag{1}$$

$$linear\_acceleration_a = acceleration_a - filtered\_gravity_a \tag{2}$$

where a is a given axis (x, y or z), $gravity_a$ is the gravity sensor's reading, $acceleration_a$ is the acceleration data acquired for the axis a and α is a constant.

Concerning GPS readings, raw data – acquired every second – are used as is, without further transformation, to record where a harsh event occurred.

3.2. Events classification module

This module takes as input the transformed sensors' data, and identifies harsh events, together with their type. In the literature, three main approaches have been explored to classify driving events: anomaly detection-, threshold-, and machine learning classifier-based methods [20].

For sake of simplicity, and to keep computation time low, we relied on threshold-based methods, and, in particular, on the "simple threshold" method reported in [20].

According to this approach, which refers to the individual sensors' readings:

- a sensor reading on the z axis (related to frontal acceleration) is labeled as "aggressive" when the value is higher than a certain acceleration threshold, or lower than a certain brake threshold;
- a sensor reading on the x axis (related to lateral acceleration) is labeled as "aggressive" when the absolute value of the acceleration is higher than a certain "harsh turn" threshold.

For the detection of harsh events, we considered the best configurations proposed by [20] and selected a moving windows of 4 seconds length. In particular, a moving window contains consecutive transformed sensors' data acquired in the last 4 seconds. If at least 50% of the sensors' readings in the window exceed a given threshold t, the event is labeled as "harsh". The threshold t varies based on the event type, and, for the proposed system, we decided to rely on the following values: $0.25g$ for harsh accelerations and harsh braking – detected by analyzing accelerometer data acquired along the z axis – and $0.3g$ for harsh turn events – detected by analyzing accelerometer data acquired along the x axis – (an overview of thresholds proposed so far in the literature is reported in [20]). Once a harsh event has been detected, its transformed sensors' data, together with other information, are stored on the device's database, to be subsequently used for the computation of the driver's score (more details are provided in the following).

3.3. Score computation module

This module is in charge of computing the driver's score for both the single trip, and for the whole game.

Concerning a single trip, this module takes as input the number of harsh events identified by the *event classification module*, respectively, the *harsh_acceleration_events_number*, *harsh_braking_events_number* and *harsh_cornering_events_number*, and computes a score for the trip according to the following approach.

First of all, the score obtained by the user for harsh events is computed. In particular, the score depends on the number of different harsh events detected, and on the number of kilometers actually traveled during the trip (which is computed based on GPS data). In case of trips with the same number of harsh events, the shorter the trip (the lower the trip's number of kilometers) the more severe the impact of the detected harsh events is; i.e. two events of harsh acceleration detected on a trip of 100 kilometers have a different impact on the trip's score than two events of harsh acceleration detected on a trip of one kilometer.

In the former case, the score for the acceleration's driving behavior is higher than the one computed in the second case. The score is computed according to equation 3.

$$harsh\_event\_score = Max[max\_trip\_score \smile \frac{harsh\_events\_number \cdot \beta}{trip\_km\_number}, 0] \tag{3}$$

where max_trip_score is a parameter set to 100, by hypothesizing a maximum number of user's trips equal to 10, and a maximum score reachable in the whole game by each user equal to 1000 (the max_trip_score is equal to the maximum score obtainable in the game divided by the maximum number of user's trips); $harsh\_events\_number$ contains the number of harsh events detected (per each harsh event type); β is introduced to distribute the number of detected harsh events on the trip's total kilometers, with the aim of differentiating the number of detected harsh events based on to the trip's length. As the trip's length is measured in kilometers, the value for β is set to 1000; the $Max()$ function is used to avoid negative scores, in case of extremely bad driving behavior (the lowest assignable score is 0).

The reason behind the inclusion of the trip's length in the score computation is because some drivers could think that the less they travel, the lower the probability to commit harsh events will be, resulting in a higher score. Hence, we decided to introduce the trip's length, and to set a minimum amount of kilometers that need to be traveled for each trip, to avoid cheating (e.g., 10-meters travels with no harsh events).

Each $harsh\_event\_score$ is then combined with the ones computed for other harsh events types, according to equation 4.

$$final\_trip\_score = \sum_{n=1}^{N} harsh\_event\_score_n \cdot harsh\_event\_weight_n \tag{4}$$

with $1 < n < 3$, as the current system considers three types of harsh events (acceleration, braking, and cornering events). The $harsh\_event\_weight$ is used to increase or reduce the contribution of a specific harsh events type (since we decided to assign the same weight to each harsh event type, we set the weight equal to 33.3% for each n).

It is worth remarking that the current equations currently consider all the harsh driving events as equals, without penalizing the score gained by the user based on "how harsh" the generated events were. In the future, the equations could be modified in order to take into account also how much the acceleration values recorded surpassed the threshold defined in Section 3.2.

The scores assigned to each trip are then added together to compute the overall score obtained by a driver during the whole game.

3.4. Smart Contract

The Smart Contract[1] is in charge of managing the competition, rankings, and prize distribution, whereas data related to the trip's scores and events is processed by the mobile app. In fact, once the user finishes a trip, all the sensors' data collected in the background are analyzed, in order to identify the number of harsh events and compute the score. Once the score has been calculated, this information is sent to the Smart Contract, that is in charge to manage the gamification part.

Given the public nature of the Ethereum blockchain, we had to pay particular attention to where to store data, in order to provide both transparency and driver privacy-preservation.

From the perspective of the drivers, the concept of transparency can be intended as related to the algorithms implemented to compute the score, to define the players' ranking, as well as to the way data acquired through the sensors are managed.

Table 2 provides an overview of the variables stored both on the mobile application and on the blockchain. In particular, concerning harsh events, we decided to store them only in the application's database. The information stored for each driving event is the location where the event occurred (longitude and latitude), the magnitude of the event (how "harsh" the event was), and the timestamp of the event. Also, the user can decide whether to store in the application's database the path he/she traveled, in order to have a diary of travels made. It is worth remarking that we decided to store information related to harsh events to increase transparency for the end user, as well as to make him/her better aware of his/her driving behavior. The user could also inspect previous trips by plotting the traveled path (if available) and harsh events generated on a map, as displayed in Figure 4

About the transparency of the gamification process, each player is allowed to look at the scores achieved by the other players, stored on the blockchain.

Concerning the transparency of the implemented algorithms, at the mobile app level, the user can inspect the mechanism behind the assigned scores, whereas at the blockchain level, the gamification logic is defined by the methods included in the Smart Contract, which are accessible and visible to everyone.

With regard to privacy-preservation, it is achieved, at the level of the mobile app, by allowing the users to access only those data related to their own trips and making other users' trips' data not visible. At the blockchain level, the privacy is preserved by making public only the scores achieved, without publishing the driving events, positions, and timestamps leading to those results.

Focusing on the gamification mechanism, the competition involves obtaining the highest score among participants within a limited time period. The driver should aim to drive accurately and avoid sudden

[1] https://github.com/noemiromani/SmartContract_Gamification

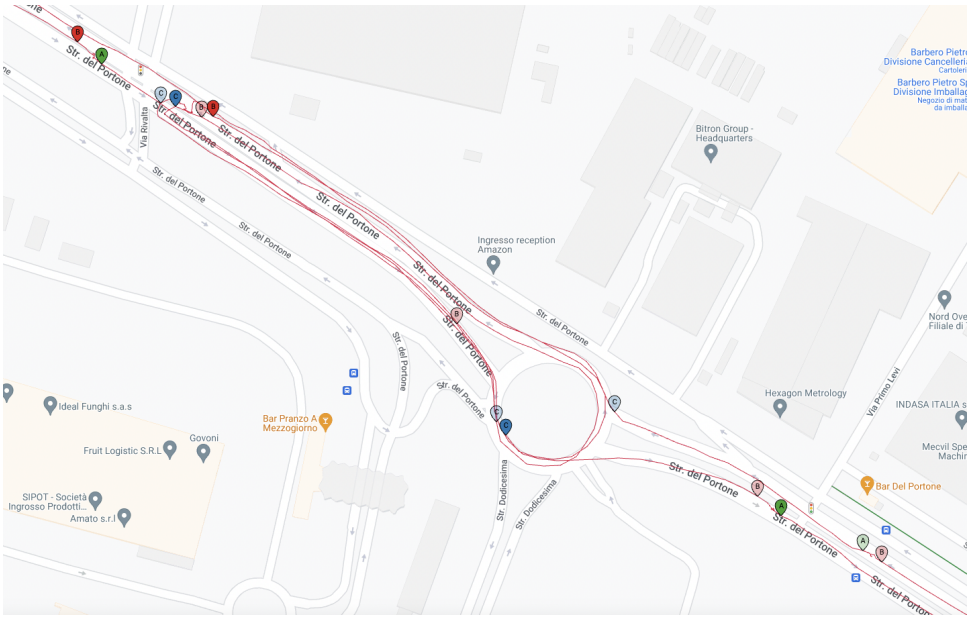


Figure 4: Example of recorded driving events (green "A" marker: acceleration, red "B" marker: brake, blue "C" marker: cornering, transparent marker: non-aggressive event, opaque marker: aggressive event, red line: path traveled by the vehicle).

Table 2
Overview of variables' visibility: Mobile App and Smart Contract (Public: everyone can access data, Saved in DB: only the user to whom data are related can access data).

Variables	Mobile App	Smart Contract
Driver's Name	Saved in DB	-
Driver's Surname	Saved in DB	-
Driver's Nickname	Public	Public
Driver's Wallet Address	Saved in DB	Public
Driver's Wallet Password	Saved in DB	-
Trip's Score	Public	Public
Trip's Destination	Saved in DB	-
Trip's Timestamp	Saved in DB	-
Number of harsh acceleration events detected during a trip	Saved in DB	-
Number of harsh braking events detected during a trip	Saved in DB	-
Number of harsh cornering events detected during a trip	Saved in DB	-
Location of harsh acceleration events	Saved in DB	-
Location of harsh braking events	Saved in DB	-
Location of harsh cornering events	Saved in DB	-
Timestamp of harsh acceleration events	Saved in DB	-
Timestamp of harsh braking events	Saved in DB	-
Timestamp of harsh cornering events	Saved in DB	-
Path travelled during trips	Saved in DB	-
Weight of driving features	Public	-

maneuvers that could negatively impact the final score. Below is a comprehensive explanation of the developed process.

After registering in the app, drivers can choose to start a new game or join an existing one.

- Start a new game: the driver must call the *init_game* function of the smart contract through the app's GUI. This function requires the driver's nickname and the desired name of the game, which will serve as a unique identifier. Additionally, the function automatically initializes certain parameters, such as the *started* flag, which indicates whether the game is active, and the start and

end timestamps of the competition.

- Join an existing game: the *add_player* function of the smart contract is called, which requires the driver's nickname and the game identifier as parameters. The function will generate an error message if it cannot find a game with the requested identifier. Additionally, when a new player is added, the function verifies that the number of participants is equal to or greater than the minimum number of required participants defined by the variable *N_MIN_PLAYERS*. If this condition is met, the *started* flag is set to *True* and the game commences. A further check is performed to ensure that the number of participants does not exceed the maximum allowed, as defined by the variable *N_MAX_PLAYERS*.

To participate in the competition, drivers are required to pay an entry fee. The fees paid by all players are added together and stored in the variable *total_amount* for each game. The *total_amount* refers to the prize of the competition, which will be distributed in varying percentages to the players who finish in the top three positions of the final ranking.

At the start of the game, players can begin recording their trips through the mobile app. After each completed trip, the app calls the *stop_trip* function to end the current trip and interacts with the smart contract by calling the *load_Trip* function. The *load_Trip* function updates the score by storing it on the blockchain, receiving as parameters the score obtained during the trip and the name of the game in which one is participating.

Once the game is finished, the leaderboard becomes final and the competition winners are determined. They are rewarded with the amount collected during registration phase. The top three drivers are considered the winners and receive a portion of the total sum, with the first place winner receiving the largest percentage. To transfer funds to the winners, we have defined the *sendPrize* function, which transfers the established amounts to the players' addresses.

Competitions remain active for up to one year, after which they automatically end. The competition end date is updated when the minimum number of players is reached, and the one-year countdown starts from that time. If a player wishes to withdraw from the competition before its conclusion, they may do so by calling the *Leave_game* function. This function requires a string parameter containing the player's motivation and the name of the game from which they wish to withdraw. Once a player has withdrawn from the game, they will receive a refund of the registration fee, even though a penalty could be applied in the future.

The sequence diagram depicted in Figure 5 represents the main steps described above for the game process. Further information about the smart contract can be accessed via the Github repository.

4. Discussions

This Section proposes a cost analysis for deploying the smart contract and executing its main functions. Additionally, our proposal serves as a prototype to demonstrate how adding gamification to the driving context, by leveraging blockchain technology, can incentivize users/drivers to adopt a safer driving style. However, there are several limitations that require further study and investigation to overcome.

Starting with the cost analysis, blockchains leverage consensus algorithms to ensure decentralization, reliability, and data security by allowing nodes in the network to agree and validate transactions. There are different types of consensus algorithms based on various strategies, but they all require computational effort, which must be paid for. Storing information or running code on a blockchain incurs a cost that varies depending on the blockchain, network congestion, and the value of the cryptocurrency used for payment. Currently, these parameters are highly variable, resulting in significant volatility in the cost of executing transactions over time.

Among the various public blockchains available, we chose Ethereum because, as stated in Section 2, it is the most commonly used blockchain for developing decentralized applications and programming smart contracts, despite not being the most cost-effective. Smart contracts on Ethereum are executed through the Ethereum Virtual Machine (EVM), which is a virtual machine shared among all nodes in the network. The cost of executing functions is measured in Gas Units, with the price per Gas Unit

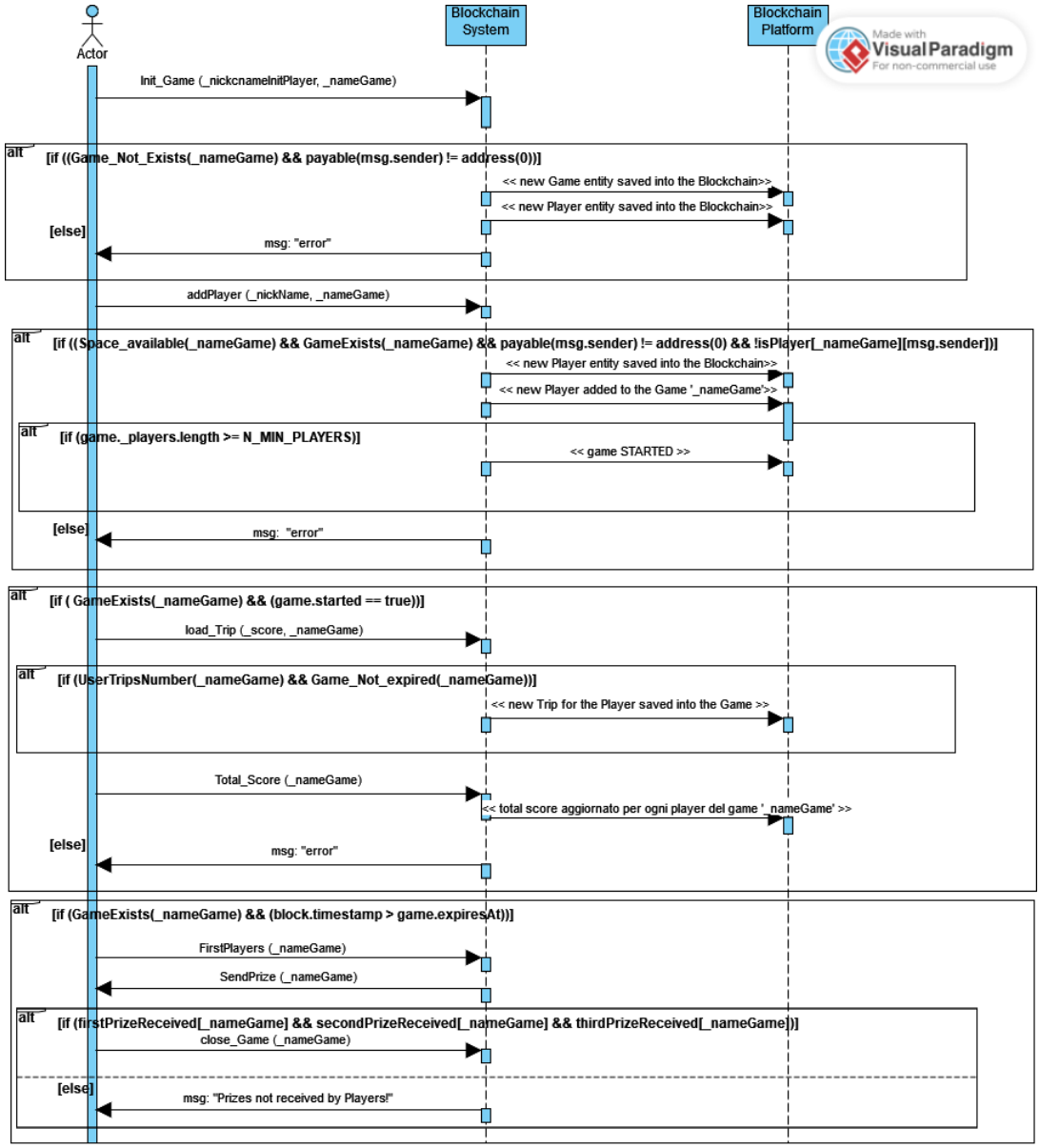


Figure 5: Sequence diagram for the game process

being affected by network congestion and cryptocurrency value. Additionally, users can add a higher or lower tip to prioritize function execution. At the time of analysis and article writing, a single unit of gas on Ethereum had an average base cost of 50 gwei, equivalent to approximately $1,70.

Table 3 displays the gas units required to execute the primary functions of our smart contract and the corresponding dollar value calculated by multiplying the gas units by the base cost of each unit. The calculation reveals that the most expensive function is *Init_game*, with a cost of $61.74, which is relatively high. Our focus was on the feasibility and operation of the smart contract. Therefore, we have not placed significant emphasis on optimizing the code, which could potentially reduce the number of gas units required for execution.

Despite the optimizations, Ethereum remains an expensive solution. To reduce costs, we decided to check the cost of our smart contract on a cheaper public blockchain. We chose Polygon, which is also based on EVM, allowing us to use the same smart contract without any changes from Ethereum. The units of gas required to execute the functions remain unchanged since they are always executed on EVM. The cost per unit of gas has changed, and as of the time of writing this paper, it is equivalent to

Table 3
Cost analysis of the main functions on Ethereum and Polygon

Function	Gas units	Ethereum (ETH)	USD ($)	Polygon (MATIC)	USD ($)
Init_game	305062	0,01525	61,74	0,02684	0,03
Add_player	224230	0,01121	45,38	0,01973	0,02
Load_trip	189983	0,00949	38,45	0,01672	0,02
Total_score	148030	0,00740	29,72	0,01303	0.02

88 GWei, which is less than $0.01. Table 3 presents the same calculation on both Ethereum and Polygon, and it shows that the most expensive function remains *Init_game*, with a total cost of $0.03. Thus, regardless of optimizations, our solution would not only be functional but also cheap and exploitable if the smart contract were deployed on a Polygon-like blockchain.

In addition to the cost of functions and possible optimization of smart contract code, there are still limitations that require further attention. A first limitation concerns score computation transparency. Currently, computation occurs off-chain, so users lack access to the algorithm and intermediate values. At the same time, moving computation on-chain would raise costs and hinder real-time data reception from sensors. Therefore, a potential solution is introducing a network of oracles, which would provide greater trust through decentralized computation without an excessive increase in cost. However, this requires further investigation before integration. Moreover, currently, a user can initiate, complete, and store a trip even if they are not the driver or if they are not moving by car (e.g., walking or using public transportation). To address the issue of driver and vehicle checks, our solution proposes performing vehicle recognition by ID and allowing users to register and associate their vehicles within the app. Additionally, our solution aims to promote safe driving practices, which can lead to reduced pollution levels. However, users may be incentivized to use their vehicles more in order to climb the ranking and achieve the highest score, which could cancel out the benefits of safe driving from an environmental perspective. To address this issue, user registration could be limited to those with electric vehicles, thus incentivizing a transition to electric vehicles, a key aspect of reducing pollutant emissions. With regard to the security of our solution, although measures to protect users' privacy have already been implemented, additional analysis and integration of security tools would provide users with an even more secure and reliable solution that does not compromise their privacy.

Finally, we set one year as the duration of the game, as this is the average duration of a car insurance policy in the event that our solution is integrated into a "Pay-How-You-Drive" system. Consequently, it was difficult to concretely demonstrate our solution's efficiency in promoting safer driving. However, we drew upon studies, such as [21, 22], that have shown how introducing a gamification-based system engages users, motivating them to adopt new habits to achieve game goals.

5. Conclusion and Future Works

This article presented a decentralized solution that incentivizes drivers to follow a safer driving style. Our proposal introduced gamification, which is an innovation in this context. The analysis of the state of the art and comparison with existing decentralized solutions revealed that no other solution has implemented this technique, to the best of our knowledge. In contrast, we used gamification to encourage safe driving by creating a competition based on rankings and scores that reflect users' driving style. Additionally, we planned to incentivize good driving behavior by rewarding the best drivers. To comprehend the development and implementation of our solution, we provided a detailed description of the architecture and its main components. This included the data receiving, event ranking, and scoring modules. Additionally, we explained the smart contract for managing the game stages and its interactions with the mobile application, demonstrating the entire process of participating in the competition. Finally, a cost analysis was conducted. The results showed that although the Ethereum blockchain is widely used, it may be too expensive for users during the production phase. The objective

of this study was to develop and test a prototype platform (mobile app, smart contract, and DB) to demonstrate the feasibility and its benefits to users. Therefore, additional analysis of cybersecurity aspects is necessary before implementing a platform that can be usable. Additionally, the limitations outlined in Section 4 must be addressed. In future works, we plan to prioritize the security of both the smart contract and the mobile application to create a platform that is resilient to cyber-attacks and provides reliability to users. In terms of cost reduction, we plan to test and compare the implementation on various public blockchains to identify the ones that offer the necessary technical features and cost-effectiveness.

Acknowledgments

This study was carried out within the MICS (Made in Italy – Circular and Sustainable) Extended Partnership and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3 – D.D. 1551.11-10-2022, PE00000004). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

References

[1] World Health Organization, Global status report on road safety 2018: Summary, Technical Report, 2018.

[2] N. H. T. S. Administration, Preview of Motor Vehicle Traffic Fatalities in 2019, Technical Report, U.S. Department of Transportation, 2020.

[3] H. S. Kim, Y. Hwang, D. Yoon, W. Choi, C. H. Park, Driver workload characteristics analysis using eeg data from an urban road, IEEE Trans. Intell. Transp. Syst. 15 (2014) 1844–1849.

[4] E. Petridou, M. Moustaki, Human factors in the causation of road traffic crashes, European Journal of Epidemiology 16 (2000) 819–826.

[5] L. Evans, Traffic safety, 2004.

[6] J. S. Hickman, E. S. Geller, Self-management to increase safe driving among short-haul truck drivers, Journal of Organizational Behavior Management 23 (2005) 1–20.

[7] P. I. Wouters, J. M. Bos, Traffic accident reduction by monitoring driver behaviour with in-car data recorders, Accident Analysis & Prevention 32 (2000) 643–650.

[8] K. Bahadoor, P. Hosein, Application for the detection of dangerous driving and an associated gamification framework, in: Proc. 4th IEEE Int. Conf. on Future Internet of Things and Cloud Workshops, 2016, pp. 276–281.

[9] N. Haworth, M. Symmons, Driving to reduce fuel consumption and improve road safety, in: Proc. Australasian Road Safety Research, Policing and Education Conference, volume 5, Monash University, 2001.

[10] J. Van Mierlo, G. Maggetto, E. Van de Burgwal, R. Gense, Driving style and traffic measures-influence on vehicle emissions and fuel consumption, Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Eng. 218 (2004) 43–50.

[11] A. Alessandrini, A. Cattivera, F. Filippi, F. Ortenzi, Driving style influence on car CO2 emissions, in: Proc. International Emission Inventory Conference, 2012.

[12] G. Castignani, T. Derrmann, R. Frank, T. Engel, Smartphone-based adaptive driving maneuver detection: A large-scale evaluation study, IEEE Trans. Intell. Transp. Syst. 18 (2017) 2330–2339.

[13] M. Winlaw, S. H. Steiner, R. J. MacKay, A. R. Hilal, Using telematics data to find risky driver behaviour, Accident Analysis & Prevention 131 (2019) 131–136.

[14] C. Huang, W. Wang, D. Liu, R. Lu, X. Shen, Blockchain-Assisted Personalized Car Insurance With Privacy Preservation and Fraud Resistance, IEEE Transactions on Vehicular Technology 72 (2023) 3777–3792. URL: https://ieeexplore.ieee.org/abstract/

document/9924540?casa_token=RzvtaI1_8SYAAAAA:tkM34l3OLpO4GHIQR8LwjfWQRJT_FoqZcr--G107YmcXgui9aGzuc41GVfWuGBtEwy3RsyP. doi:10.1109/TVT.2022.3215811.

[15] L. Yi, Y. Sun, B. Wang, L. Duan, H. Ma, B. Wang, Z. Han, W. Wang, CCUBI: A cross-chain based premium competition scheme with privacy preservation for usage-based insurance, International Journal of Intelligent Systems 37 (2022) 11522–11546. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/int.23053. doi:10.1002/int.23053.

[16] W.-Y. Lin, K.-Y. Tai, F. Y.-S. Lin, A Trustable and Secure Usage-Based Insurance Policy Auction Mechanism and Platform Using Blockchain and Smart Contract Technologies, Sensors 23 (2023) 6482. URL: https://www.mdpi.com/1424-8220/23/14/6482. doi:10.3390/s23146482.

[17] P. K. Singh, R. Singh, G. Muchahary, M. Lahon, S. Nandi, A Blockchain-Based Approach for Usage Based Insurance and Incentive in ITS, in: TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON), 2019, pp. 1202–1207. URL: https://ieeexplore.ieee.org/abstract/document/8929322. doi:10.1109/TENCON.2019.8929322.

[18] Z. Wan, Z. Guan, X. Cheng, PRIDE: A Private and Decentralized Usage-Based Insurance Using Blockchain, in: 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2018, pp. 1349–1354. URL: https://ieeexplore.ieee.org/abstract/document/8726619. doi:10.1109/Cybermatics_2018.2018.00232.

[19] H. Qi, Z. Wan, Z. Guan, X. Cheng, Scalable Decentralized Privacy-Preserving Usage-Based Insurance for Vehicles, IEEE Internet of Things Journal 8 (2021) 4472–4484. URL: https://ieeexplore.ieee.org/document/9210591. doi:10.1109/JIOT.2020.3028014.

[20] V. Gatteschi, A. Cannavò, F. Lamberti, L. Morra, P. Montuschi, Comparing algorithms for aggressive driving event detection based on vehicle motion data, IEEE Transactions on Vehicular Technology 71 (2021) 53–68.

[21] J. Hamari, J. Koivisto, H. Sarsa, Does gamification work? - a literature review of empirical studies on gamification, 2014, p. 3025 – 3034. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84902291971&doi=10.1109%2fHICSS.2014.377&partnerID=40&md5=bd7772d917e17e6d228ef8ef9501880d. doi:10.1109/HICSS.2014.377, cited by: 2826; All Open Access, Bronze Open Access.

[22] P. Bitrián, I. Buil, S. Catalán, Enhancing user engagement: The role of gamification in mobile apps, Journal of Business Research 132 (2021) 170–185. URL: https://www.sciencedirect.com/science/article/pii/S0148296321002666. doi:https://doi.org/10.1016/j.jbusres.2021.04.028.