Journal of Internet Services and Applications, 2025, 16:1, doi: 10.5753/jisa.2025.5039

Blockchain-Based location validation in an environment of mutual distrust

Eduardo Busch Loivos [**Universidade Federal Fluminense** | ***eduardo_loivos@id.uff.br***]
Arthur A. Vianna [**Universidade Federal Fluminense** | ***arthurvianna@id.uff.br***]
Antonio A. de A. Rocha [**Universidade Federal Fluminense** | ***arocha@ic.uff.br***]

Institute of Computing, Universidade Federal Fluminense, Av. Gal. Milton Tavares de Souza, s/n, São Domingos, Niterói, RJ, 24210-590, Brazil.

Received: 04 October 2024 • **Accepted:** 13 May 2025 • **Published:** 09 July 2025

Abstract As technology advances, more and more of our day-to-day objects are made smart and communicate with each other. Vehicles are no exception. They can exchange data to increase safety on the road or find more efficient routes, among other conveniences. Just like every other moving object, many of these applications are location-dependent. However, the network is vulnerable to safety and privacy threats. Certain dishonest and misbehaving peers may try to exploit the network. Therefore, establishing trust between vehicles is one of the network's most important challenges. We propose a decentralized and verifiable model for verifying GPS data using Cartesi Rollups. In this way, we count on the scalability of the multilayer blockchain solution and avoid the communication overhead of a voting system. We evaluated this model by simulating it with SUMO, varying the density of fraudulent nodes and the signal range. Under these conditions, it was able to identify fraud attempts with more than 85% accuracy in the worst cases.

Keywords: Blockchain, Decentralized, Location proof, Trust management, Vehicular networks

1 Introduction

Each phase of the World Wide Web, invented by English computer scientist Tim Berners-Lee in 1989 [Getting, 2007], moves a step closer to decentralization. Web stages were defined as Web of Documents (Web 1.0), Web People (Web 2.0), and Web of Data (Web 3.0) [Choudhury, 2014]. The increasing popularity of Blockchain technology, combined with the introduction of the General Personal Data Protection Law, marks the beginning of the third era of the Web. These developments have granted users greater freedom and participation in the virtual environment.

From the first implementation of the WEB to the most recent, each phase represents accomplishing a challenge. The transition to Web 2.0 brought benefits to several areas, attracting countless users and allowing for better, faster, and lower-cost applications [Murugesan, 2007]. Similarly, the transition to Web 3 comes with an expectation to return the control of personal data to the original contributor [Ragnedda and Destefanis, 2019]. The structure created by Nakamoto [Nakamoto, 2008] allows transactions between users to be carried out without a trusted third party. This property of blockchain, applied to the network, enables decentralized operation without a centralized coordinator [Ragnedda and Destefanis, 2019], fulfilling the challenge to start a Web phase.

Blockchain networks achieve consistency between nodes through consensus algorithms. This procedure ensures that every trusted participant reaches the same states in the same order [Muratov *et al*., 2018]. This way, transactions are only carried out with the network majority's approval. This model assumes honest behavior from most nodes and uses their numbers to establish the truth. It is impossible to guarantee the quality of personal information, but it is possible to confront conflicting data.

Location data is essential for several systems present in everyday life, such as navigation routes, autonomous vehicles, and regional advertisements or monitoring. Computational systems that rely on such data to make decisions face a challenge when in a mutual distrust scenario where nodes can purposely fail or collude. Due to this barrier, system developers usually try simplistic solutions like the one presented by this report [Prefeitura Rio, 2021]. In the report, the municipal government presents a system that monitors bus lines and allows them to impose a fine on companies that are not operating with the agreed fleet. This approach creates a scenario of mutual distrust between the city hall and bus companies, where the government makes decisions based on data sent by bus companies.

Knowing this challenge and the consequences of fault data for those systems [Boeira *et al*., 2017]. This work proposes a decentralized and verifiable method for validating GPS data through the historical records on the Blockchain. The notion of proximity between vehicles makes it possible to detect inconsistencies between inputs so that attempts to cheat the system can be identified. An in-depth explanation of this model can be found in the section 4. Session 5 shows another use of logs to identify irregularities; this method seeks to find changes in a vehicle's behavior through discrepancies in the distance traveled.

The remainder of the paper is organized as follows. Section 2 provides a quick overview of existing studies. In section 3, we describe the proposed scheme. Session 6 covers the steps to simulate traffic behavior, and the simulation re-

sults are presented in Section 7. In Section 8, a DApp (Decentralized Application) that implements the proposed scheme introduced in Section 3 is presented. Finally, we draw conclusions and future work in section 10.

2 Related Work

Smart transportation has become an important subject in recent studies. Coordinates are a key component of these works since they can be used to automate vehicles or provide region-based information to drivers in traffic applications. D-PARK, as described in the paper by Park [2021], offers a high-resolution estimation to aid in the search for available parking spots. Another research approach is to extract information such as direction and linear speed to detect accidents [Kumar *et al*., 2020].

In a decentralized environment, it is crucial to prioritize security and trust. Some researchers also exploit some features of the blockchain to achieve the necessary robustness, such as Kudva *et al*. [2021], Singh *et al*. [2020], and Yang *et al*. [2018]. However, the unique dispute system of Cartesi change how consensus is reached to an optimistic approach. Through this tool, an honest user can demonstrate his execution of a deterministic application. Dismissing the need for votes or major computing power.

The closest searches to the present work are the Vouch+ from Boeira *et al*. [2017] and the Trust-based Exclusion Acess-control Mechanism from Ferraz *et al*. [2014].

The Vouch+ uses travel distances and angles to predict the movement in Ad Hoc networks of automated vehicles. The decentralized network generates a security requirement that each vehicle must prove its position to the platoon. When the environment becomes unsafe, cars operating without human interaction must have strategies to keep passengers secure. Vouch+ uses inter-vehicle communication to strengthen the evidence, but doesn't consider beacons as a source of truth.

The Trust-based Exclusion Access-control Mechanism presents a node exclusion mechanism for Ad-Hoc Networks. TEAM also uses the neighborhood to inspect the behavior of nodes. What differs is that each node has the power to evaluate the data of its neighbors and report only cases of misbehavior to the jury. The jury is a set of randomly chosen nodes that decides about the report by vote to avoid collusion and slander attacks.

The main difference between this study and the others presented is the usage of GPS information. Most projects that use this type of data aim to increase the position accuracy or improve the organization of sensors. In addition, those who provide an analysis of the exchange of messages focus on the protocol and not the content, so they lack GPS-specific techniques.

3 System Model

In a decentralized environment, every piece of data requires either proof or recognition from the network majority. We propose a model that leverages Cartesi Rollups to overcome this limitation and enable scalable, verifiable computation. Cartesi provides two key advantages that align with the needs of our protocol: (1) it abstracts the underlying blockchain infrastructure, allowing the use of conventional programming languages and standard development tools; (2) it implements optimistic rollups with dispute mechanisms, enabling any user to enforce the correct result of a deterministic algorithm, even in adversarial contexts; This combination of flexibility, enforceability, and scalability makes Cartesi a fitting choice for the proposed location validation system. The subsection 3.1 further explains the Cartesi technology and its components.

The idea of the proposed scenario is not to estimate the exact position of each device. The need arises from an environment of mutual distrust, which is prone to attack. All nodes can determine their position. However, confirming whether the stated position aligns with the other available information is essential. The vehicles transmit their location and the list of devices they manage to detect in their vicinity. To correlate data across time and vehicles, each node includes a public identifier in its broadcast, such as the vehicle's license plate, which is already a publicly visible and commonly used means of identification in real-world traffic environments. By mapping the declared position of every device in the snapshot and checking the vicinities, we can confirm the plausibility of the declared position through Lateration. Along the way, there are times when it is not possible to find devices in the vicinity. To increase the coverage, we use speed to determine whether the distance traveled between two snapshots is consistent.

Both models have their requirements. The Lateration proof needs a minimum of nodes to form a vicinity and determine the location. In contrast, the Distance proof requires a previously established position. Figure 1 shows the sequence of checks made on an incoming input to decide the validation method. The sections 4 and 5 present the validity of the Lateration and the Distance proofs, respectively.

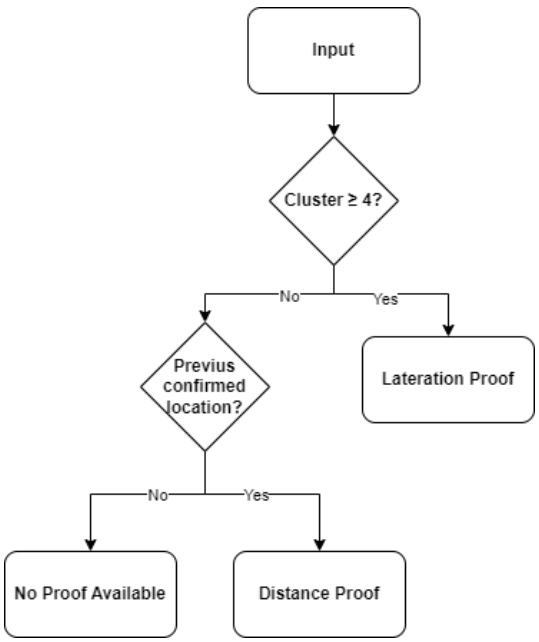

Figure 1. Proof decision workflow

3.1 Cartesi

Cartesi [Cartesi, 2022] is a layer-2 Blockchain solution for developing decentralized applications. It allows DApps to be developed using conventional programming languages by offering a Linux operating system coupled with a Blockchain infrastructure. In addition, various software, tools, libraries, and services used to create traditional applications are available to the developer, ensuring a familiar development environment.

One of the core ideas of this work is to utilize a permissionless Blockchain, since we are in a mutual distrust scenario, to run our DApp that validates GPS data. However, permissionless Blockchains have a scalability problem that became evident once they gained popularity [Thibault *et al.*, 2022]. To attain the scalability required for the DApp, Cartesi uses a combination of two technologies, the Cartesi Machine and the Optimistic Rollups framework, in what is called Cartesi Rollups. A Rollup is a category of Layer-2 scalability solutions for Blockchain where the transactions are processed off-chain in a cheaper environment, and later, the result is posted on-chain. Therefore, using Cartesi Rollups, we can run the computation off-chain in the Cartesi Machine, which provides a Linux environment, and the result is later posted on-chain (Layer-1).

3.1.1 Cartesi Rollups

The Cartesi Rollups are Cartesi's version of Optimistic Rollups for EVM-compatible Blockchains. On Optimistic Rollups, it is assumed that computations executed off-chain by a layer-2 node are valid unless contested. So, after the result of said computation is posted on-chain, other layer-2 nodes have a dispute period during which they can challenge the results by computing a fraud-proof; during this period, transactions cannot be reversed.

This solution has layer-1 and layer-2 components. Layer 1 contains a series of smart contracts that together form the Cartesi Rollups framework, which is responsible for data availability, consensus, and settlement of the DApp. The Cartesi Node is the layer-2 component responsible for performing the application calculations using a Cartesi Machine. Since each Cartesi DApp has a whole machine for itself, a Cartesi Node validates a specific application. The node is then responsible for ensuring that the computation result obtained off-chain is reliable, fighting against dishonest validators to ensure that honest claims are met.

The Cartesi Node is also responsible for maintaining an SQL database that stores the application's outputs and serves them through a GraphQL [Hartig and Pérez, 2018] API. Cartesi DApps have three types of outputs: Notices, Vouchers, and Reports. A notice is a verifiable data declaration that attests to off-chain events or conditions and is accompanied by proof. It is a mechanism to communicate essential off-chain events in the execution layer to the base layer in a verifiable manner. Vouchers serve as a mechanism for facilitating on-chain actions initiated in the execution layer. They are like digital authorization tickets that grant dApps the authority to execute specific actions directly on the base layer. Reports are stateless logs, offering a means to record read-only information. They are similar to notices without associated proof.

Instead of posting on-chain each new update of the Cartesi Machine's state, the Cartesi Nodes do this at the end of each epoch. An epoch is a sequence of grouped entries that follows the same cycle. By verifying in this way, we avoid excessive interaction with the Blockchain. The application's back-end reads the input load and processes it. Figure 2 details the work of a validator node to verify and register a batch of transactions on the Blockchain. This way moves most of the computing outside the Blockchain, using it as a data provider for the off-chain execution of applications. Also, it's important to highlight that only the machine's Merkle Root hash is posted on-chain.

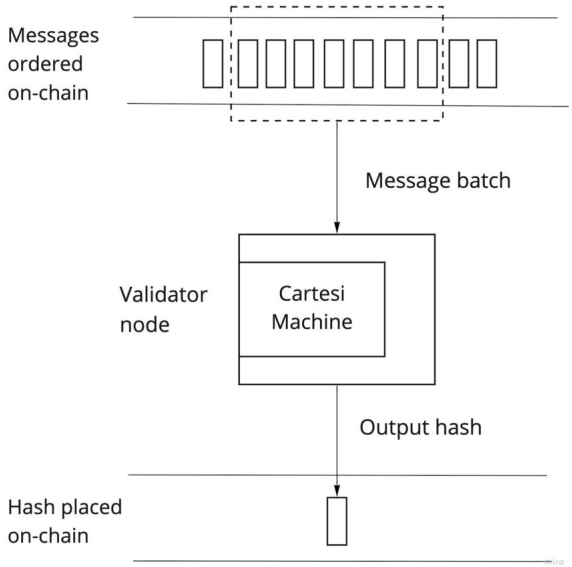

Figure 2. Transaction grouping and output hash storage [Moura, 2021]

3.1.2 Cartesi Machine

Cartesi Machine is the component that allows developers to use traditional tools to create scalable decentralized applications. So, when developing a Cartesi DApp, the developer generates a Cartesi Machine that contains its application back-end, and the Cartesi Node validating this application will run this very same machine to process the inputs. Unlike traditional Virtual Machines, the Cartesi Machine is not intended to be interactive. Instead, it should boot up, open the operating system, run some predefined software, and then stop [Cartesi, 2022].

An important aspect is that the Cartesi Machine is deterministic to guarantee that every Cartesi Node running the same machine reaches the same state. The machine is a custom RISC-V architecture with all entropy removed to be deterministic. The Cartesi Node executes the machine using an emulator written in C/C++ with POSIX dependencies restricted to the terminal, process, and memory mapping facilities. RISC-V guarantees stability by simulating real hardware and can work at a lower level with more development tools [Teixeira and Nehab, 2018].

The dispute resolution process between two Cartesi Nodes relies on the Cartesi Machine determinism to settle on the ma-

chine state (DApp state) through a Verification Game, which is a protocol that allows an arbiter with limited computational resources to referee a game between two computationally unlimited players [Teixeira and Nehab, 2018]. In this game, the layer-1 blockchain is the "referee", and the "game" is between a "player" Bob, who defends a result for an off-chain computation, and a "player" Alice, who disputes it.

The dispute resolution has two steps, **the disagreement step** and **settling the dispute** [Teixeira and Nehab, 2018]. The first step is to find the exact instruction where the validators, Bob and Alice, disagree. It works by first identifying the machine instructions interval between an initial state that they agree and a final one that they don't. Then, using n-ary search, the interval is reduced to a single instruction. The settlement step occurs on the base layer. Using a reference implementation of the entire Cartesi Machine architecture, the blockchain is capable of running the instructions found in the previous step and reaching the state in which the validators disagree. Whoever claims the same state reached by the blockchain wins the dispute. This dispute mechanism allows an honest Cartesi Node always to win the challenge and, therefore, enforce the correct result.

4 Lateration Proof

Lateration uses the distance between an object and reference points to calculate its position. These distances form circles (or spheres in three-dimensional space) around the reference points, and their intersections contain the object's position. The most common use of Lateration is called Trilateration. The "tri" comes from the three points it uses the distance to define the location. The communication between transmitters and receivers determines the distance between network nodes by measuring the signal travel time [Asaad and Maghdid, 2022].

For a quick reference, the definitions of the values used in this session can be found in table 1

Table 1. Acronyms table of Lateraion Proof formulas

Acronym	Definition
C_i	Device of known position at (x_i, y_i)
d_i	Distance between R_i and the target object
D_i	Maximum distance between R_i and the target object

Theorem 1. *$N+1$ distances are required to determine the coordinates of points in an N-dimensional space through Lateration.*

Proof. In a 2D environment, with the fixed points $C_1 = (x_1, y_1)$ e $C_2 = (x_2, y_2)$ and the respective distances to the object: d_1 e d_2. The object's coordinate is determined by the point of intersection between the circles:

$$(x-x_1)^2 + (y-y_1)^2 = d_1^2 \quad (1)$$
$$(x-x_2)^2 + (y-y_2)^2 = d_2^2 \quad (2)$$

Expanding each equation:

$$x^2 - 2x.x_1 + x_1^2 + y^2 - 2y.y_1 + y_1^2 = d_1^2 \quad (3)$$
$$x^2 - 2x.x_2 + x_2^2 + y^2 - 2y.y_2 + y_2^2 = d_2^2 \quad (4)$$

And then subtract equation 4 from 3

$$2x(x_2 - x_1) + 2y(y_2 - y_1) + (x_1^2 + y_1^2) - (x_2^2 + y_2^2) = d_1^2 - d_2^2 \quad (5)$$

Once y is isolated in 5, it is evident that the equation represents a straight line

$$y(y_2 - y_1) = -x(x_2 - x_1) + \frac{d_1^2 - d_2^2 + ||C_1||^2 - ||C_2||^2}{2} \quad (6)$$

Isolating x and replacing it in the starting equations 1.

$$y^2(\frac{(y_2-y_1)^2}{(x_2-x_1)^2} + 1) - y(\frac{(y_2-y_1)}{(x_2-x_1)}(2A - 2x_1) + 2y_1) + A^2 - 2A + x_1^2 + y_1^2 = d_1^2 \quad (4)$$

$$\text{Where } A = \frac{\frac{r_1^2 - r_2^2 + ||C_1 - C_2||^2}{2}}{(x_2 - x_1)}$$

Since the system can have more than one solution, disproved by the principle of reduction to absurdity. ■

A single solution is possible if the object and both fixed points are on the same line, as shown in Figure 3. A new distance is required to undo the ambiguity and define the exact position of the device.

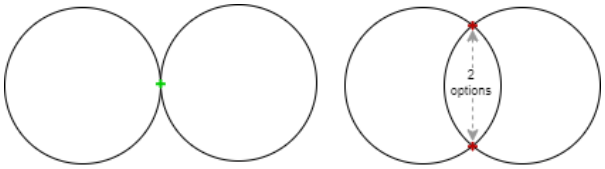

Figure 3. 2-dimensional intersection

Theorem 2. *It is possible to define a device position through its distances to 3 reference points in known positions.*

Proof. For $e_i = (x_i, y_i)$ the coordinate of the reference point i and d_i is the distance between the point i and the device. In the same way as theorem 1, the device position is on the circles drawn around reference points with the distances as radius. The coordinates can be obtained by solving the system of equations:

$$(x-x_1)^2 + (y-y_1)^2 = d_1^2$$
$$(x-x_2)^2 + (y-y_2)^2 = d_2^2$$
$$(x-x_3)^2 + (y-y_3)^2 = d_3^2$$

Expanding the equations:

$$x^2 - 2x.x_1 + x_1^2 + y^2 - 2y.y_1 + y_1^2 = d_1^2 \quad (1)$$
$$x^2 - 2x.x_2 + x_2^2 + y^2 - 2y.y_2 + y_2^2 = d_2^2 \quad (2)$$
$$x^2 - 2x.x_3 + x_3^2 + y^2 - 2y.y_3 + y_3^2 = d_3^2 \quad (3)$$

The next step is to subtract equations 2 and 3 from equation 1 and simplify to obtain a system with two linear equations:

$$x(x_2 - x_1) + y(y_2 - y_1) = \frac{d_1^2 - d_2^2 + a^2}{2} \quad (4)$$
$$x(x_3 - x_1) + y(y_3 - y_1) = \frac{d_1^2 - d_3^2 + b^2}{2} \quad (5)$$

Where $a = ||e_1 - e_2||$ and $b = ||e_1 - e_3||$.

The device location is obtained by isolating x and y in Equations 4 and 5. ■

4.1 Inaccurate Lateration

Without precise information, the circles will not intersect at a single point. Interference delays the communication, generating a perceived distance greater than the real one. From that, the intersections of the circles will result in a large region. In this case, the estimated location is somewhere in the intersection.

Corollary 1. *The region is generated from circles with larger radii, so the resulting intersection also consists of a larger area. Figure 4 shows the difference between precise and inaccurate Lateration. The system of equations that defines the shaded area of inaccurate Lateration is:*

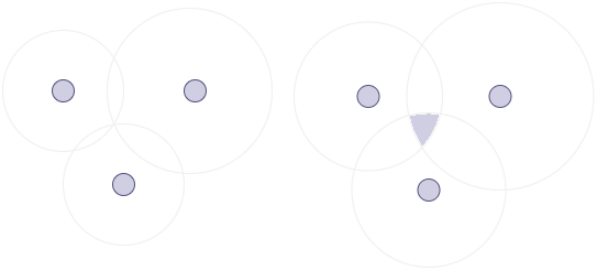

Figure 4. Accurate Trilateration and inaccurate Trilateration

$$(x-x_1)^2+(y-y_1)^2 \leq D_1^2 \quad (1)$$
$$(x-x_2)^2+(y-y_2)^2 \leq D_2^2 \quad (2)$$
$$(x-x_3)^2+(y-y_3)^2 \leq D_3^2 \quad (3)$$

As $D_i > d_i$ for each station, the device should be inside each circle. Therefore, the device is inside their intersection.

4.2 Fault tolerance

Every system is prone to failure, whether due to losses in communication or deliberate attempts to bypass the protocol. The data flow has a decentralized origin. Therefore, the validator is not able to assess the quality of the information provided. The system must be tolerant to Byzantine faults, the position must be carried out by consensus, and it must be capable of identifying flaws of any nature.

Lemma 3. *In a cluster with $2c$ nodes. If the $c+1$ nodes are trustworthy and $c \geq 2$, it is possible to verify the position of the generator node of this cluster.*

Proof. Considering the system is partially synchronous, every message arrives at the validator with maximum delay t. Any message that exceeds the limit is regarded as a refusal or loss. Furthermore, the devices in each vehicle are similar and have the same detection range.

If g is the node where the position is being validated. Each device that detects g must be within its detection reach, even if g doesn't declare. This allows the identification of groups that support or oppose the declared position.

- The devices that confirm the position of g are:
  - Nodes in range of g that declare g.
- Devices that reject the position of g are:
  - Nodes outside the range of g that declare g.
  - Nodes in range of g that do not declare g.

There must be no disagreement between honest nodes. So, at least $c+1$ nodes will either confirm or deny the position of g, making one of the groups the absolute majority. Selecting the largest group of nodes is equivalent to following the decision of the most reliable and trustworthy nodes.

By definition, $c+1 \geq 3$, it is possible to define the estimated position of g through Trilateration using this minimum majority. ■

5 Plausible Displacement Proof

This method checks the viability of a reported position based on the last established location of a vehicle. This check allows the vehicle's route to be proven when forming groups is impossible. A mobility model determines if the distance traveled since the last proof is consistent with the rest of the data.

Table 2. Acronyms table of Lateraion Proof formulas

Acronym	Definition
t	
x_t	The vehicle's position at time t
v_t	The vehicle's velocity at time t
A_{max}	Highest recorded acceleration on a vehicle of kind
A_{min}	Minimum recorded acceleration on a vehicle of kind
$v_0(i)$	Speed at beginning of interval i

The mobility model predicts vehicle movement between two-time instants k and $k+1$. Limits are determined using the maximum value for acceleration, while the remaining information is taken from the previous proof. The equations use the variables defined in table 2 to establish the limits of a vehicle's displacement.

$$\Delta x = v_k * \Delta t + \tfrac{1}{2} A_{max} * \Delta t^2 \quad (1)$$

The position declared by a vehicle is classified as valid if its total movement is lower than the thresholds defined by the equation. This way, it is possible to validate any route taken, even in areas where Trilateration is impossible.

Lemma 4. *Given the two instants of time $t_2 > t_1$, the distance traveled in the interval $[t_1, t_2]$ must be less than the distance traveled using the maximum acceleration.*

Proof. Dividing the interval into several sub-intervals ΔT_i whose acceleration is constant. The initial velocity in the interval $v_0(i)$ is given by:

$$\forall \Delta T_i, v_0(i) = \Sigma_{n=0}^{i-1} v_0(n) + a_n * \Delta T_n \quad (2)$$

By definition, $a_n \leq A_{max}$ and $v_0(n)$ is less than the velocity at the same instant of uniformly varied motion with the maximum acceleration.

For each interval, the distance traveled is given by:

$$\Delta S_i = v_0(i) * \Delta T_i + \tfrac{a_i * \Delta T_i^2}{2} \quad (3)$$

Since $v_0(i) * \Delta T_i$ and $\frac{a_i * \Delta T_i^2}{2}$, are positive and the velocity and acceleration are smaller than their respective ones in uniformly varied motion. The distance traveled is less than or equal to the distance generated with maximum acceleration. ■

6 Evaluation model

The data to validate the models was generated through simulation using the Simulation of Urban Mobility (SUMO) [Behrisch *et al*., 2011] application toolkit, which is used to prepare the map and configure the traffic demand. This tool is used to ease routes' manipulation and create a dataset with a clear definition of node behavior. Figure 5 shows the sequence of steps required to prepare the simulation and to generate and evaluate the simulated database. Each of those steps is further explained in the following subsections.

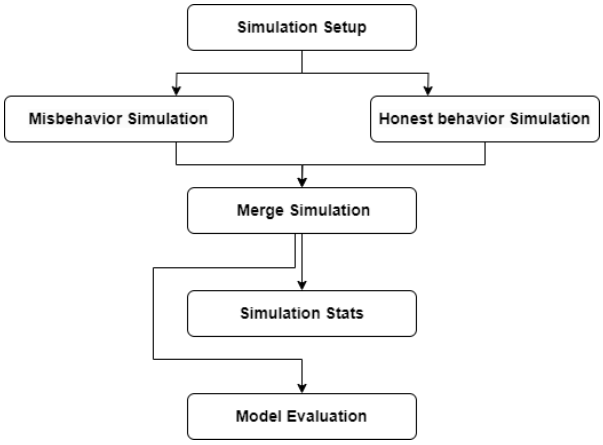

Figure 5. Simulation workflow

6.1 Simulation Setup

To avoid errors when creating the simulation network and to ensure behavior that allows the proposed models to occur, the following requirements were defined:

- **Street layout:** The region must contain several intersections. Crossroads leads to the formation of clusters.
- **No overpass or underpasses:** Overlapping tracks can cause errors in the simulator conversion.
- **No roundabouts:**"Reduce the number of random decisions made in creating routes to minimize the chance of errors."

Based on the requirements established and familiarity with the region, it was determined that the simulation would be mapped based on Icaraí. The simulation graph was formed from a file exported by OpenStreetMap. This file contains road information, such as direction and number of lanes. Structures and pedestrian-only lanes were removed. The map conversion also includes commands to simplify traffic lights and predict overlapping lanes and roundabouts behavior. Figure 6 shows the simulation map after the setup step.

Figure 6. Simulation roadmap based on Icaraí

6.2 Misbehavior and honest behavior Simulations

Using the SUMO toolkit, it is possible to generate random routes. A route consists of an origin, a destination, and a departure time. Once the set of routes has been defined, misbehavior is simulated by changing the file. The origin and destination to generate a detour or a factor of 0.6 is applied to the speed to create a delay.

The next step is to simulate both configuration files, with and without misbehavior, to extract the FCD (floating car data) output. This output contains location and speed, along with other information, for every vehicle in the network at every time step. To ease handling, these files are converted to JSON containing only the relevant information: ID, speed, latitude, and longitude.

Finally, the neighborhoods are formed. A list of nearby devices is created for each distance (10, 20, 50, 100, and 200), and evaluated vehicle. Attempts to bypass the model are produced by combining the files, as explained in the following subsection.

6.3 Merge Simulation

One file contains the expected behavior while the other contains the executed behavior, but vehicles in both contribute to the truth. The attempts to cheat the protocol appear with different views when combining those files. The Byzantine faults are emulated as follows:

- The data from the trusted nodes comes from the file with improper behavior. They must see and report the faults;
- The data from the falsifying nodes comes from the file with no faults. They report as if there were no problems on the route.

6.4 Simulation Stats and Evaluation

Simulation Stats consists of a script to capture information that does not depend on the proposed protocol, such as the average period spent in the system, the largest and average size of the groups, and the average time between Lateration proofs.

Evaluation classifies each message as true, false, or unclassified following the model described in section 3. It then compares this with the expected result and exports the number of errors and hits for the correct and incorrect entries. It also exports the total of unclassified inputs.

7 Results discussion

The experiments were performed using uniform arrival with a single vehicle per second in a 10-hour simulation. Under these conditions, the average time spent in the system is approximately 3.46 minutes. The following parameters and variations were used to evaluate the model described in section 3:

- **Signal range:** Variation between 10, 20, 50, 100, and 200 meters
- **Misbehavior percentage:** Variation between 1, 5, 10, 20, and 30 percent
- **Maximum acceleration:** according to the study Szumska *et al.* [2022], the maximum acceleration on a dry surface is $2.81 m/s^2$.
- **Update or not:** Starting points for the distance test should only be from Lateration or previous distance tests.

To evaluate the effect of the possible parameters, each possible configuration was repeated 10 times with the same set of seeds.

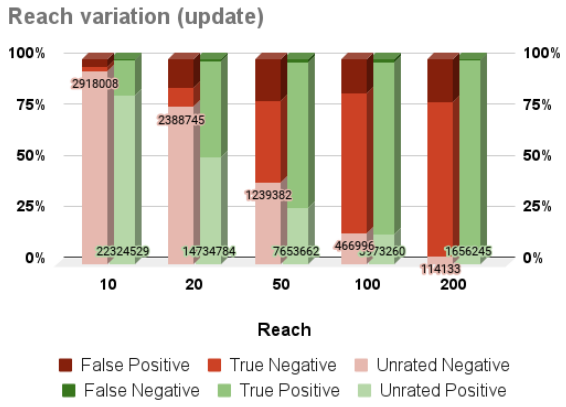

Figure 7. Total messages by reach graph with update

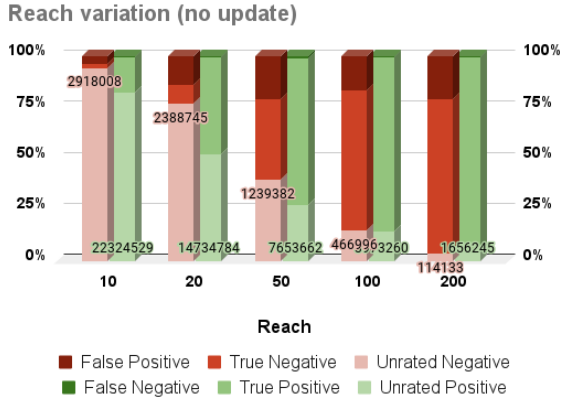

Figure 8. Total messages by reach graph without update

Various types of signals are used to measure the distance between objects. These vary in range among many other factors. To choose which beacon to use in a real-world scenario, understanding the impact of reach on protocols is key. The figures 7, 8 and 9 show the consequences of varying this value. The primary impact of this metric is on the coverage, with a significant reduction in the percentage of non-measurable values. The coverage percentage of values in the negative classification (misbehavior) increased from 5.5% to 96.3%, while the increase for the expected behavior was 17.6% to 94.1%. The accuracy of the Lateration model was slightly lower with the shortest range, and it performed best with a range of 100m.

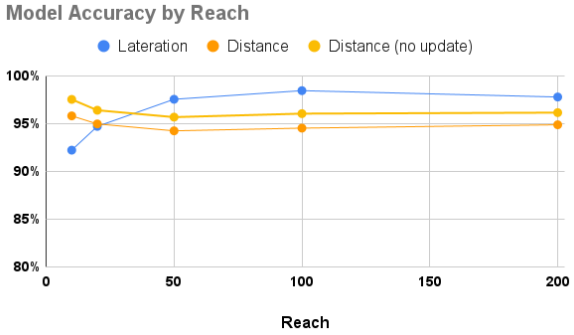

Figure 9. Model accuracy by reach graph

Regarding reach, there was no significant difference between accepting previous distance proofs or just lateration proofs. The no-update model was slightly more accurate due to the more relaxed rules and the greater volume of normal behavior.

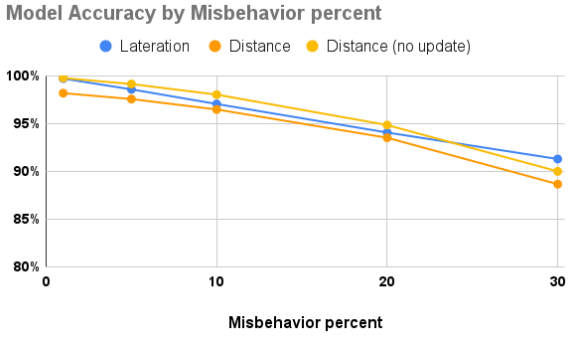

Figure 10. Model accuracy by fault percentage graph

As the proportion of bad behavior increases, the likelihood of conspiracy grows. Therefore, it is important to evaluate the variation in accuracy. The precision decline can be seen in Figure 10. In addition, lateration proved to be slightly more resistant to the increase in faults, as shown in the less severe downward tendency line. To deepen the understanding of this decline, Figures 11 and 12 reveal the aggregation of every possible outcome. The graphs show a gradual increase in the number of untrustworthy values classified as trustworthy by the model, which may be a reason for this decline.

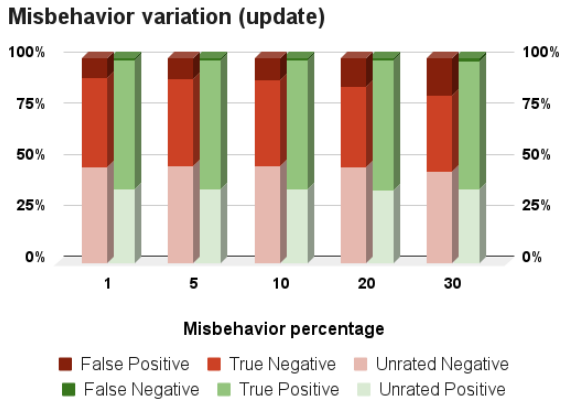

Figure 11. Total messages by fault percentage graph with update

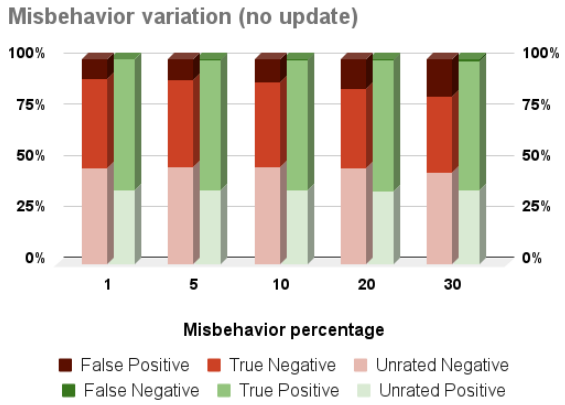

Figure 12. Total messages by fault percentage graph without update

8 Use case: IoT Rollups DApp

The purpose of the use case is to enable fines issued through GPS records [Prefeitura Rio, 2021] to be made through the blockchain. The ledger structure is capable of resolving conflicts and bringing transparency and trust to those involved in the process. For public transport to work properly, each vehicle must follow a schedule. This schedule refers to the route the vehicle must take and the time it must be at each stop to pick up or drop off passengers. Therefore, this can be seen as a contract between the municipality and the company providing the service.

To do this, each vehicle must provide the data that identifies it, a unique ID, and a trip identifier, along with the information needed to check for fines and validate the position. In practice, the unique ID can correspond to an existing public identifier, such as the vehicle's license plate, which is already commonly used for identification in traffic systems. This avoids introducing new privacy risks while ensuring consistency in the verification process. The information provided for fining consists of the vehicle's coordinates and the timestamp at the measurement time. The DApp receives this information and consults a database that has the schedule of that vehicle, and then the verification is done using these two pieces of information. The fines are generated in the form of a notice[1], which is then queried using GraphQL to generate the public dashboard. If there is no problem with the schedule, you need to check that the information provided is reliable. To test the compliance, each vehicle must share its speed and the identifier of every other vehicle in the range of detection. This allows for the generation of proofs as described in the sections 4 and 5.

To provide the described data, they must be equipped with IoT (Internet of Things) devices. These devices, usually small, are pieces of equipment that connect to the internet and together form an ecosystem in which a particular application runs using them somehow. They generally function as sensors or controllers. As sensors, these devices are intended to generate measurement data, such as temperature, humidity, and proximity. As controllers, these devices are designed to perform state changes, for example, opening a door or window when requested. In this context, they would be sensors that provide the vehicle's coordinates, speed, and proximity to other devices.

If it is proven that a particular vehicle is not complying with the schedule, either by deviating from it or providing falsified information, a fine should be generated. The fines generated should be available on an online dashboard so that the public can consult them. The following subsections provide details about the application's front-end and back-end.

8.1 Front-End

The application must provide a dashboard for the public to access, but this is not the only role of the front-end in this DApp. The front-end is provided by a web server developed in NodeJS and is also responsible for receiving inputs and passing them on to the Blockchain using Web3, which is a new standard for interaction on the internet that relies on decentralized technologies such as the Blockchain.

The Web3 library used to build the front-end was Web3.js since javascript is the native language of NodeJS. The web server developed follows the MVC architecture, which is made up of 3 components (**M**odel, **V**iew, and **C**ontroller) that give the pattern its name.

The front-end uses the following functions to communicate with the blockchain and the application's backend:

- getNoticePage: Performs queries to the Cartesi Node to retrieve the generated events (notices).
- getAccounts: Using the Hardhat Blockchain connection module defined in the config, it performs a query to retrieve the existing accounts.
- addInput: It executes the addInput method of the input contract to send the input to the DApp back-end.

The notices retrieved are used to assemble a table summarizing the events, showing the ID of the vehicle involved, the time, the value of the fine, and a symbol representing the type of occurrence. Selecting an entry shows the vehicle's position on the map and specific elements relating to the kind of occurrence. Figure 13 presents two real-world examples of suspicious behavior detected by the protocol. On the left, a vehicle deviates from its expected route, clearly visible on the map. On the right, a vehicle arrives late to a scheduled stop, with the stop's position and the associated delay displayed through a pop-up for contextual clarity.

Figure 14 illustrates two types of misbehavior detection. On the left, a reported position is rejected by neighboring

[1] A verifiable record that certifies off-chain computations.

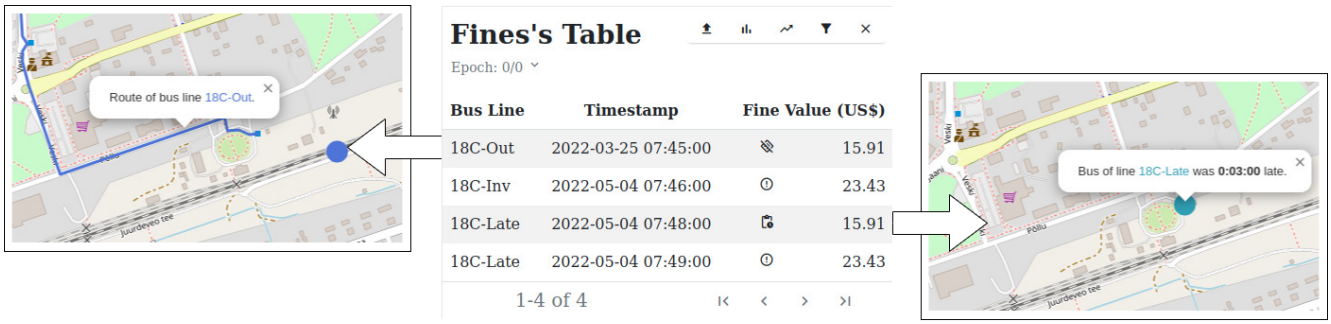

Figure 13. Application Dashboard Displaying Route Deviations and Stop Delays

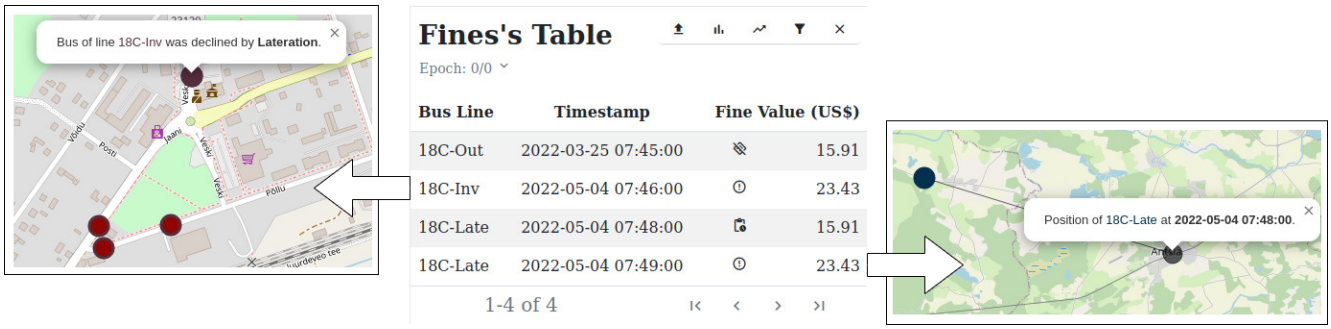

Figure 14. Application Dashboard Displaying Misbehavior Detection

vehicles through vicinity analysis; the rejecting vehicles are marked with red-filled icons. On the right, the rejection is based on distance: the traveled segment exceeds the maximum expected range, which is depicted by a line connecting the start and end points of the report.

8.2 Back-End

The application's back-end was developed in Python and runs inside Cartesi Machine. It receives two types of input, one of which is the schedule of a bus line, and the other is data sent by the IoT device in a vehicle. Depending on the input received, the back-end must process it differently, as shown in the flowchart in the figure 15.

If the input received is a timetable for a bus line, the processing consists of saving the timetable in Dapp's SQLite database. This input contains the identifier of the bus line, the route it takes, the coordinates of the stops, and the schedule for each trip made by that line.

If the input received is data sent by a vehicle, it is necessary to group the messages to apply the Lateration test. The inputs are grouped according to the timestamp of the message in 1-minute epochs. When checking the timestamp of an input, there are three possible outcomes:

- **It belongs to the current epoch:** The input is stored and will be processed at the end of the epoch;
- **It belongs to a previous epoch:** In this case, the input is rejected due to the delay;
- **It belongs to the next epoch:** It triggers the epoch change, the data from the current epoch is retrieved for processing, and the Dapp starts storing inputs for the "new" current epoch.

Once the epoch changes, the inputs are evaluated one by one. From the trip identifier, which is also provided by the vehicle, query the schedule database to retrieve the route and schedule for that trip. With the data received by the input and the data retrieved from the database, it is possible to check whether the vehicle is on the route and whether the vehicle is late or not.

Dapp assumes that no vehicle would lie to prejudice itself and only assesses the reliability of vehicles that have not been fined. This reduces processing time since the Lateration test consists of searching for all vehicles at a certain distance from the author of the message and comparing them with the list of vehicles that have "seen" it. Vehicles on both lists simultaneously represent a favorable vote, while vehicles on only one list represent a negative vote. If the negative votes are the majority and the union of the groups is large enough for the test, i.e., greater than 4, the input position is rejected.

Once the input completes the Lateration test without rejection, the last test is to verify if the distance traveled from the last known position is acceptable. The data from the last non-rejected position is retrieved, and the maximum expected distance is calculated from this recorded speed. This is based on the highest acceleration recorded for a vehicle of this kind. According to Szumska *et al.* [2022], this acceleration is 2.81 m/s^2

9 Limitations and Future Work

While the proposed protocol offers a novel approach for validating vehicle positions using historical data, certain limitations remain and point toward valuable future work:

9.1 Simplified Movement Modeling:

The current method considers speed and acceleration but does not explicitly incorporate the direction of movement. While sufficient for consistency checks in many scenarios,

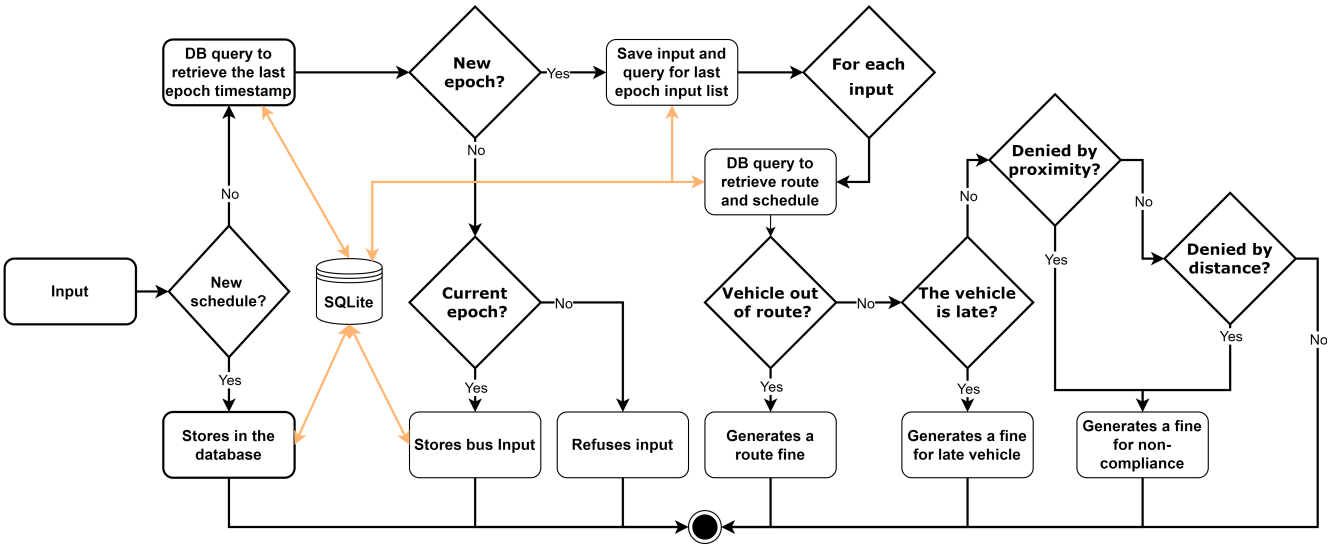

Figure 15. Back-End Flowchart

incorporating direction data could enhance accuracy, especially in complex urban environments. Additionally, with directional data and tighter sampling intervals, it would be possible to estimate not only a maximum displacement but also a minimum expected traveled distance, allowing for finer-grained validation and better detection of implausible position reports.

9.2 Evaluation Based on Simulation:

The evaluation presented in this work is based entirely on simulation, which allows controlled testing but does not fully capture the challenges of real-world environments, such as GPS signal loss or atmospheric interference. However, the proposed protocol is designed to tolerate short-term inaccuracies in GPS readings by validating the consistency of reported positions over time, rather than relying on precise instantaneous data. While minor GPS fluctuations have limited impact on the verification process, real-world testing is essential to assess the system's robustness under these degraded conditions and unpredictable scenarios.

9.3 Comparison with Alternative Approaches:

This work does not provide a detailed comparison with other approaches, such as smart contract-based systems for position verification or fine enforcement. A meaningful comparison would require not only evaluating performance but also conducting a viability study, taking into account aspects such as cost, complexity, and latency of different implementations. Additionally, the integration of an application with blockchain technologies introduces scalability challenges. Solutions like Cartesi, which enable off-chain computations with verifiable results, are promising candidates for supporting the execution of this protocol in a decentralized context. While this paper focuses primarily on the security constraints of the model, future work must also investigate how the solution interacts with the operational characteristics and limitations of various blockchain's scalability solutions. A complete comparison must consider both the technical viability and security compatibility of each approach to assess which is best suited for practical deployment.

10 Conclusion

The configuration with the best results was the one with a range of 100m and without updating the previous proofs. A prototype was implemented using these parameters using Cartesi Rollups and the data available on the data.rio API [Prefeitura da Cidade do Rio de Janeiro, 2022], demonstrating how the solution operates in the real environment. The application's front end displays a list of occurrences and a map to visualize the vehicle and its neighbors of the selected occurrence.

The requirement for prior proof and the high incidence of misbehavior recognized through distance highlights the need to explore new proof mechanisms relying entirely on vehicle information. In addition, assigning each vehicle a reputation value could improve the performance of these or future models. Another way to improve performance is to allow passengers to participate. They are the ones most interested in the proper operation of the system and could send opposing data to help identify conspiracies. With more inputs in the system, the probability of forming clusters will undoubtedly be higher.

The DApp developed in session 8 shows just one way of using the proofs. The same logic can be used to solve any dispute related to location. Tracking mail packages, finding lost or stolen cell phones, or locating drivers of carpool apps are other examples of using the proposed model.

Acknowledgements

The authors want to thank the Cartesi Foundation for all the support.

Funding

IARA partially funded this work - Center of Artificial Intelligence

in the Remaking of Urban Environments (FAPESP project grant 2020/09835-1).

Authors' Contributions

EBL contributed to the conception, development, deployment, and writing of this study. AAV contributed to the conception, analysis of the results, and writing. AAAR contributed to the paper's conception, analysis of the results, and manuscript writing. All authors read and approved the final manuscript.

Competing interests

The authors declare that they do not have competing interests.

Availability of data and materials

The simulation environment, including the base files and a guide for generating the data, is available at `https://github.com/EdLoivos/Fault_simulation_SUMO`. The application showcased in the use case section can be accessed at `https://github.com/EdLoivos/IoT-Smart_Line`, which includes a short demonstration as well as the interface and backend components.

References

Asaad, S. M. and Maghdid, H. S. (2022). A comprehensive review of indoor/outdoor localization solutions in iot era: Research challenges and future perspectives. *Computer Networks*, 212:109041. DOI: 10.36227/techrxiv.15138609.v1.

Behrisch, M., Bieker, L., Erdmann, J., and Krajzewicz, D. (2011). Sumo–simulation of urban mobility: an overview. In *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. ThinkMind. Available at: `https://sumo.dlr.de/pdf/simul_2011_3_40_50150.pdf`.

Boeira, F., Barcellos, M. P., de Freitas, E. P., Vinel, A., and Asplund, M. (2017). Effects of colluding sybil nodes in message falsification attacks for vehicular platooning. In *2017 IEEE Vehicular Networking Conference (VNC)*, pages 53–60. IEEE. DOI: 10.1109/vnc.2017.8275641.

Cartesi (2022). Cartesi Documentation. Available at: `https://www.cartesi.io/en/docs/`.

Choudhury, N. (2014). World wide web and its journey from web 1.0 to web 4.0. *International Journal of Computer Science and Information Technologies*, 5(6):8096–8100. Available at: `https://ijcsit.com/docs/Volume%205/vol5issue06/ijcsit20140506265.pdf`.

Ferraz, L. H. G., Velloso, P. B., and Duarte, O. C. M. (2014). An accurate and precise malicious node exclusion mechanism for ad hoc networks. *Ad hoc networks*, 19:142–155. DOI: 10.1016/j.adhoc.2014.03.001.

Getting, B. (2007). Basic definitions: Web 1.0, web. 2.0, web 3.0. Available at: `https://www.practicalecommerce.com/Basic-Definitions-Web-1-0-Web-2-0-Web-3-0/` Acesso em 13.07.2023.

Hartig, O. and Pérez, J. (2018). Semantics and complexity of graphql. In *Proceedings of the 2018 World Wide Web Conference*, pages 1155–1164. DOI: 10.1145/3178876.3186014.

Kudva, S., Badsha, S., Sengupta, S., La, H., Khalil, I., and Atiquzzaman, M. (2021). A scalable blockchain based trust management in vanet routing protocol. *Journal of Parallel and Distributed Computing*, 152:144–156. DOI: 10.1016/j.jpdc.2021.02.024.

Kumar, N., Acharya, D., and Lohani, D. (2020). An iot-based vehicle accident detection and classification system using sensor fusion. *IEEE Internet of Things Journal*, 8(2):869–880. DOI: 10.1109/jiot.2020.3008896.

Moura, E. (2021). Cartesi rollups. Available at: `https://medium.com/cartesi/scalable-smart-contracts-on-ethereum-built-with-mainstream-software-stacks-8ad6f8f17997` Accessed on 2022-02.

Muratov, F., Lebedev, A., Iushkevich, N., Nasrulin, B., and Takemiya, M. (2018). Yac: Bft consensus algorithm for blockchain. *arXiv preprint arXiv:1809.00554*. DOI: 10.48550/arxiv.1809.00554.

Murugesan, S. (2007). Understanding web 2.0. *IT professional*, 9(4):34–41. DOI: 10.1109/mitp.2007.78.

Nakamoto, S. (2008). Bitcoin whitepaper. *URL: https://bitcoin. org/bitcoin. pdf-(Дата обращения: 17.07. 2019)*. DOI: 10.21428/9610ddb2.a6a2490c.

Park, S. (2021). D-park: User-centric smart parking system over ble-beacon based internet of things. *Electronics*, 10(5):541. DOI: 10.3390/electronics10050541.

Prefeitura da Cidade do Rio de Janeiro (2022). Transporte rodoviário: Api de gps dos ônibus (sppo). Available at: `https://www.data.rio/documents/PCRJ::transporte-rodoviá1rio-api-de-gps-dos-ônibus-sppo-beta/explore`.

Prefeitura Rio (2021). Prefeitura poderá multar consórcios por meio dos dados de gps dos ônibus. Available at: `https://prefeitura.rio/transportes/prefeitura-podera-multar-consorcios-por-meio-dos-dados-de-gps-dos-onibus/.Acessoem13.07.2023`.

Ragnedda, M. and Destefanis, G. (2019). *Blockchain and Web 3.0*. London: Routledge, Taylor and Francis Group. DOI: 10.4324/9780429029530.

Singh, P. K., Singh, R., Nandi, S. K., Ghafoor, K. Z., Rawat, D. B., and Nandi, S. (2020). Blockchain-based adaptive trust management in internet of vehicles using smart contract. *IEEE Transactions on Intelligent Transportation Systems*, 22(6):3616–3630. DOI: 10.1109/tits.2020.3004041.

Szumska, E., Stańczyk, T., Zuska, A., Grabski, P., Jaśkiewicz, M., Jurecki, R., Kurczyński, D., and Łagowski, P. (2022). Experimental testing of longitudinal acceleration in urban buses. In *IOP Conference Series: Materials Science and Engineering*, volume 1247, page 012017. IOP Publishing. DOI: 10.1088/1757-899x/1247/1/012017.

Teixeira, A. and Nehab, D. (2018). The core of cartesi. *Whitepaper, Cartesi*. Available at: `https://w3.impa.br/~diego/publications/TexNeh18.pdf`.

Thibault, L. T., Sarry, T., and Hafid, A. S. (2022). Blockchain scaling using rollups: A comprehensive survey. *IEEE Access*. DOI: 10.1109/access.2022.3200051.

Yang, Z., Yang, K., Lei, L., Zheng, K., and Leung, V. C. (2018). Blockchain-based decentralized trust management in vehicular networks. *IEEE internet of things journal*, 6(2):1495–1505. DOI: 10.1109/jiot.2018.2836144.