Soarchain

A decentralized mobility network for cars, pedestrians, and infrastructure

Soar Robotics Team
Release 0.7.6(2023-08-08)

Abstract

This paper proposes a new blockchain architecture based on Proof of Availability, a proof to cryptographically verify whether a device made its connectivity and computational resources available for use within a specified timeframe and record it on a ledger. Used in combination with open-source, secure, and tamper-proof hardware for communication and computation; Delegated Proof-of-stake (DPoS) for staking and governance; Byzantine Fault Tolerant (BFT) replicated state machine for block generation and finalization; and smart contracts for regulating the exchange of data and resources between the network users, it offers a novel and scalable way to create decentralized mobility networks that depend on highly mobile and heterogeneous sets of participants securely and honestly exchanging application-specific, protocol-agnostic data.

The emergence of blockchain technology has been a game-changer for many industries, and mobility is no exception. A blockchain-based vehicle-to-everything (V2X) network can revolutionize how vehicles collect, process, and utilize data and communicate with each other, infrastructure, and people. Our new architecture provides a cryptographically secure and resource-optimized way to verify that an individual message-generating node has correctly broadcast a certain number of messages and that an individual message-processing node has independently and correctly received and processed a certain number of messages, using only a negligibly small subset of the complete message set.

1. Introduction

As of 2022, there are approximately 1.446 billion vehicles worldwide excluding two-wheelers [1]. This number is only expected to grow, with newer generation cars equipped with a range of sensors and systems to support safer and smarter driving. However, even with the most well-equipped of these vehicles, ubiquity of autonomous driving is still far from reality. One of the greatest obstacles to the

widespread adoption of autonomous vehicles is that they are not yet sufficiently connected to their surroundings. While modern automobiles can perceive millions of data points per second, they do so in isolation and without context. Bringing context and higher degrees of situational awareness to vehicles is only possible with the introduction of communication and computation technologies that seamlessly integrate with the vehicles we drive and the infrastructure that we depend on. Vehicle-to-everything(V2X) communication is the key technology that will solve the problem of disconnected and contextless cars.

Vehicle-to-everything (V2X) is a set of wireless communication technologies that enable direct communication between vehicles and other road users and infrastructure components such as traffic lights and road signs. V2X technology can revolutionize road safety by providing real-time information to drivers and driverless systems about potential hazards on the road ahead, as well as enhancing the efficiency of traffic flow. Moreover, V2X will enable new applications such as automated parking, electric vehicle charging, pay-how-you-drive insurance, car resource sharing, cooperative adaptive cruise control, and many more. Used in combination with edge computing capabilities, it will ultimately pave the road to fully-autonomous fleets of vehicles. However, V2X technology has not yet taken off. One reason is that it requires a critical mass of vehicles to be equipped with the necessary hardware and software. Otherwise, the benefits of V2X would be limited. Another reason is that there are still some technical challenges to overcome, such as ensuring that data is accurately and consistently transmitted between vehicles and that it is authentic, integral, and plausible.

Note: This white paper is in constant development. We will strive to keep this document updated with the most recent development progress. Due to the continuing and iterative nature of our development process, it is likely that the final code and implementation will differ from what is described in this paper. As we continue to open source various system components over time, we invite interested readers to explore our GitHub repository at https://github.com/soar-robotics

2. Overview

Soarchain addresses the first of the challenges mentioned above through its novel crypto-economic protocol that incentivizes its users to form a decentralized wireless network of cars, infrastructure, and people. One of the most powerful aspects of crypto-economic protocols is their capacity to design incentive systems that enable anybody in the world to freely(permissionless) participate in achieving a set of common objectives. Furthermore, these incentive systems can be precisely adjusted to enhance the collaboration between their participants to accomplish these objectives[2]. Bitcoin's "Peer-to-Peer Electronic Cash System"[3], Ethereum's "allowing anyone to write smart contracts and decentralized applications where they can create their own arbitrary rules for ownership, transaction formats and state transition functions"[4], Zerocash's "Decentralized Anonymous Payments from Bitcoin"[5], Cosmos's "network connecting many independent blockchains"[6] and many other successful implementations of well-known crypto-economic protocols have proven that it is possible to coordinate large-scale global digital communities to create digital economies. The success of these networks has inevitably led the way for the creation of opportunities for innovation in the coordination of people and capital formation that extend beyond the digital realm and into the physical world. This is referred to as "Proof of Physical Work"[2]. It is an exceptionally practical way of incentivizing people to crowd-source the rapid deployment of large-scale infrastructure networks, which would otherwise be extremely time-consuming and create unimaginable amounts of capital and operational expenses for a

single operating company. Soarchain's goal is to become the Layer-1 blockchain for token-incentivized physical infrastructure networks[7], starting from transportation and telecommunications.

Soarchain addresses the second of these challenges through its novel "Proof of Availability(PoAv)" mechanism, which ensures -through a set of well-established cryptographic primitives and stochastic models- that the network contributors make their connectivity and computational resources available in a predetermined way which is beneficial for the network. By making these resources available through PoAv, it is possible to create a network of interconnected vehicles that directly(without any intermediaries, wirelessly, over the air) communicate with each other where authenticity, integrity, confidentiality, continuity, and consistency of data, privacy, and nonrepudiation of users are guaranteed.

3. Key Components

Blockchain

Soarchain is a distributed ledger that is a decentralized, immutable, and append-only database that keeps track of the account balances, unique node addresses, and various scores of the network participants. It is composed of blocks containing a header and a Merkle tree comprising a list of hashes generated from the transactions in that block.

Proof-of-availability is one of the main building blocks of the whole network architecture, thus the Motus devices which are secure hardware that enables data collection and processing abilities within the car, are required to participate in the creation of proofs by submitting the necessary data when they are asked by the challengers. Each Motus has a score which should be kept over a certain threshold to be able to receive rewards from the finalized proof processes. The signatures and data collected from the challenged Motus are checked by the challengers and submitted as reward requests to the validators. At each consensus round, validators achieve consensus on the validity and order of the transactions and generate and finalize the blocks according to their consensus mechanism. At the end of each epoch, equivalent to 30 blocks, coins are minted to the PoAv module.

The rewards are distributed to validators only when a proposed block is verified and added to the blockchain. A verified block consists of the following main fields:

- Header
- Transactions
- Evidence
- Last Commit

Soarchain makes use of Tendermint Core as its Byzantine Fault Tolerant (BFT) middleware. As some of the aforementioned fields are specific to Tendermint, we invite interested readers to go through the exhaustive rundown of the field and subfield definitions, which can be found in Tendermint Data Structures documentation[8].

Nodes

Soarchain Mobility Network comprises different types of nodes, either categorized according to the degree that they store blockchain data and allow access to the data or according to their duty and expected collaboration within the network.

Full Nodes

A full node is a blockchain participant that possesses a complete copy of the blockchain. Full nodes play an important role in maintaining the security and stability of the blockchain network by validating and relaying transactions. They also help to keep the blockchain network decentralized by providing a platform for participating nodes to reach a consensus on the blockchain's consensus rules.

Light Nodes

The notion of light nodes was implied under the name of “Simplified Payment Verification” in the Bitcoin whitepaper[3]. It discusses a distributed consensus process tracker that only verifies the consensus method and not the state machine transitions included within the blockchain. A light node is a simplified version of a complete node. Full nodes consume a lot of resources since they execute transactions and verify the results. On the other hand, light clients have much lower resource requirements because they just validate results (without actually executing the transactions). As opposed to full nodes that store a large amount of data, including blocks, transactions, results, etc., light clients only keep a few of the most recent headers.

Challenger Nodes

As their name suggests, challenger nodes create challenges to be accepted by the participants of the network. Different types of challenges are defined within the network, each of which serves a different purpose and plays a key role in ensuring the honest behavior of the network participants.

Validator Nodes

A validator node is responsible for block proposal, block validation, block information propagation(to other nodes) and block finalization. They participate in the consensus protocol by “voting” on the validity of the next block through their cryptographic signatures signed by their corresponding private keys. It is essential that the validator nodes are also full nodes since the whole network depends on validators for the generation and finalization of blocks through consensus.

Messaging Nodes

A messaging node is responsible for honestly and correctly generating, broadcasting, receiving, processing, and verifying purpose-specific messages through an air interface which enables direct communication between geographically proximate nodes. These nodes are rewarded through Soarchain’s incentive mechanism on a regular basis according to their reputation scores.

Computational Nodes

A computational node is responsible for honestly and correctly executing a specific computational task demanded by an authorized and authenticated node. The computational tasks they are expected to perform could range from receiving, preprocessing, and compressing simple ego-vehicle data to more advanced tasks such as running plausibility checks for the received proximate vehicle data and even running distributed computing tasks which are delegated to them by projects that institutions run. Ultimately, computational nodes will run the latest autonomous driving software version while periodically generating proofs to verify that they have been running that version of the software.

Onboard Units (Motus)

Motus are the open-source hardware units that enable the vehicles to communicate with each other, the infrastructure, people, and the cloud through different communication protocols:

- **Cellular-V2X PC5 Mode 4** protocol for disseminating signed and encrypted messages that help vehicles and other proximate nodes become aware of each other and directly exchange information
- **Cellular communication (5G-NR/4G-LTE)** for constant connectivity to the main network. This enables message-generating vehicles and message-processing vehicles to send verified proofs to be validated by the validator network and in order to successfully get included in the next block
- **Wi-Fi and Bluetooth** for user/driver connectivity, visualization of applications and data on the smartphone
- **Storage and processing power** for executing proof-related processes and storing logs from different sources within the internal network of the vehicle
- **Trusted Platform Module and Hardware Security Module** for securely storing master and private keys, accelerating cryptographic functions and preventing tampering of the hardware
- **CAN** and **OBD-II** access to the vehicle for reading real-time device and diagnostics information
- **High-precision GNSS and Inertial Measurement Unit (IMU)** to provide data for localization, safety and autonomy applications

There are different types of Motus devices which support different configurations and proof mechanisms. Motus is the full-fledged hardware that employs all the aforementioned features and is able to participate in all the proof mechanisms in Soarchain, as well as issue all types of challenges. Motus Mini is the low-end hardware which employs only a few of these features and links directly to the user's mobile device to use it as a relay device. They can only participate in network-based proof mechanisms. Motus can be thought of as a light, messaging and computational node.

Messages

Cooperative Awareness Message (CAM)

Cooperative awareness means that road users and roadside infrastructure are aware of one other's position, dynamics, and characteristics. Road users include all road vehicles, including automobiles, trucks, motorcycles, bicycles, and even pedestrians, as well as roadside infrastructure equipment, such as

road signs, traffic lights, and barriers and gates. Cooperative Awareness is achieved through the regular exchange of information among vehicles, which are facilitated by CAMs. CAM is a message transmitted by a vehicle to other vehicles and roadside units in order to share information about the vehicle's self and its surroundings. The message contains information such as the vehicle's position, speed, heading, braking status etc. This information is then used by other vehicles to improve situational awareness and make decisions such as changing routes or speeds to avoid collisions [9].

Decentralized Environmental Notifications Message (DENM)

DENMs are designed to be sent and received quickly, allowing drivers to be swiftly alerted to any potential hazards or changes in their environment. These changes include but are not limited to car accidents, road construction, adverse weather conditions, and other traffic-related events. This helps to improve safety by allowing drivers to be aware of their surroundings and make informed decisions about their route. DENMs are emitted based on events in comparison to CAMs emitted all the time in regular intervals. DENMs are also used to help manage congestion by informing drivers of traffic patterns and other conditions that may impact their journey[10].

Collective Perception Message (CPM)

The purpose of Collective Perception is to tell other vehicles and road users about the current driving environment. It contains information regarding detected items for this purpose (i.e. other road participants, obstacles and alike). By contributing context information, other vehicles help a vehicle reduce ambient uncertainty about its current environment. CPM is basically the extended version of CAM, that contains more data fields for disseminating raw and processed data from data-intensive sensors that are used in assisted and autonomous driving such as radars, lidars, cameras and alike. In order to reduce the resulting channel load and concentrate on reporting changes in the dynamic road environment, the CPM is broadcast cyclically with adaptive message generation rates[11].

Vulnerable Road User Awareness Message (VAM)

Vulnerable Road User Awareness Message is an important tool in helping to reduce the number of collisions between motor vehicles and people who are considered to be more at risk of serious injury or death. This includes pedestrians, cyclists, motorcyclists, and those who use mobility assistance devices such as wheelchairs or walkers. VAMs are messages delivered by the wireless communication device that the VRU carries in order to create and maintain awareness of vulnerable road users utilizing the VRU system.

A VAM holds information regarding the status and attributes of the originating VRU. The content may vary depending on the VRU profile. Status information often comprises time, location, motion state, cluster status, etc. Typical attribute information provides data about the VRU profile, type, and dimensions, among other characteristics. The receiving vehicle becomes aware of the presence, type, and status of the originating VRU upon reception of a VAM. The information received can be utilized by the receiving vehicle to enable multiple VRU-related ITS applications. Comparing the status of the originating VRU to its own status, for instance, a receiving vehicle can evaluate the collision risk with the originating VRU and warn the driver via the HMI or take assisted action to brake.[15]

Custom-built Messages

These are ad-hoc messages that serve the purpose of different applications that run on vehicles using Soarchain. The aforementioned messages are intended to be disseminated over the air through direct vehicle-to-vehicle communication. They are verified and incentivized through the vehicle-to-vehicle proof-of-availability(V2V PoAv). On the other hand, each message also has a vehicle-to-network counterpart within Soarchain where decentralized servers help to move data between vehicles, and they are verified and incentivized through the vehicle-to-network proof-of-availability.

Challenge Mechanisms

As an efficient data availability verification method, Soarchain utilizes various challenge mechanisms. A challenge is simply a request to receive data to be verified from a node which claims to have generated that specific data. It provides the participating nodes a cryptographically secure way to prove that their connectivity and computational resources have been available throughout a specific timeframe. Due to the high degrees of mobility and heterogeneous distribution of the vehicle types of the participating network nodes, multiple types of challenges exist within the network to ensure the correct and honest operation of nodes which results in fair and timely distribution of rewards. Challenge mechanisms in Soarchain are designed to maximize the useful work done by each node to serve the main applications/functions of the network while also verifying the data and securing the nodes. The presented data in response to a challenge is utilized for doing useful work such as verifying that the node which provided the response in fact is acting honestly.

V2V Receiver Challenge

V2V receiver nodes get challenged as described in Challenger Procedure A, and respond to it as described in Receiver Procedure A. The purpose of this challenge is to prove that the receiver has been successfully receiving the generated messages from at least one broadcaster node.

V2V Broadcaster Challenge

V2V broadcaster nodes get challenged as described in Challenger Procedure B , and respond to it as described in Broadcaster Procedure B. The purpose of this challenge is to prove that the broadcaster node has been generating the required set of messages which were claimed to be received by one of the receivers.

V2N Runner Challenges

When a V2N broadcaster successfully transmits a message to a runner, it gets recorded to the runner's temporary database. This runner can request a challenge when it is eligible(either due to time limit or amount of messages in its database). A V2N broadcaster's rewards are calculated according to the number of successful messages that they transmit. The eligibility for the reward is verified by the runner and is submitted to the validators as a reward distribution transaction for the V2N broadcaster of interest.

Scoring

When messaging nodes join the Soarchain network, they are assigned a score ***S***. Any messaging node that scores higher than the initially assigned score is considered to be a well-functioning node. The reward distribution process is executed according to the score of a messaging node, which is an indicator of how well it has been performing in its correct and honest state.

When a challenge result is received, the score is updated according to the outcome of the challenge. The score is always kept within the range of 0 to 100, and all nodes start with an initial score of 50. If the challenge was successful, the score is increased by a certain amount based on the current value of the score and maximum increase factor. If the challenge was unsuccessful, the score is decreased by a certain amount also based on the current value of the score and maximum decrease factor.

The calculation of the score and the corresponding increase or decrease is performed using the following formulas:

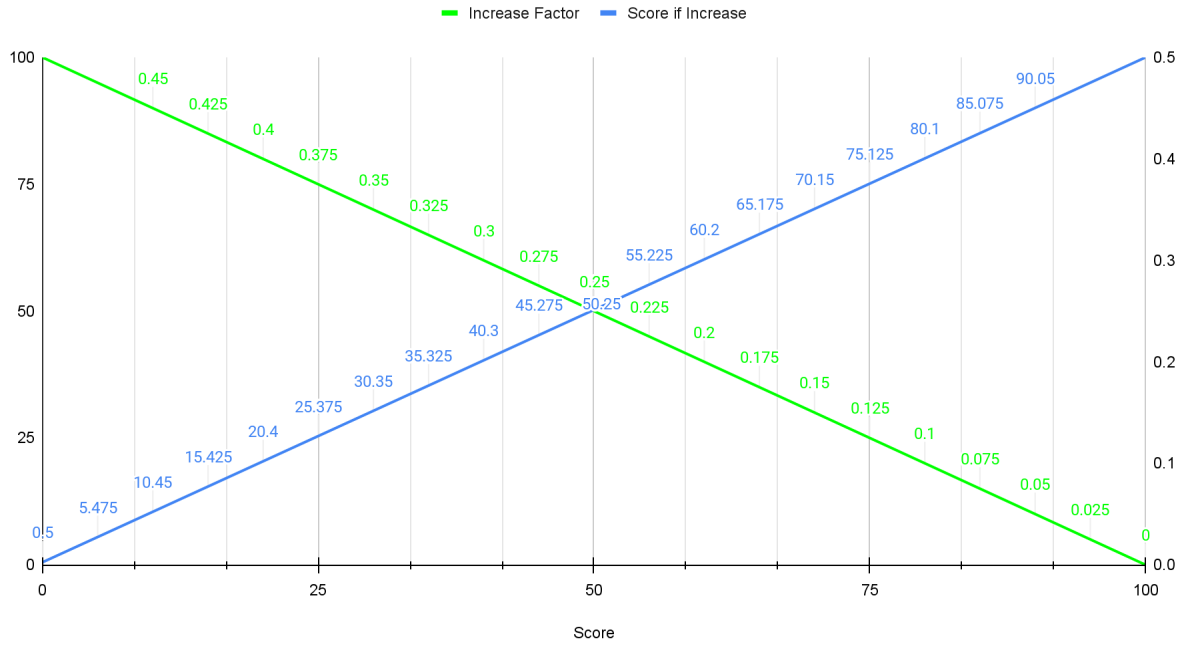
Let ***S*** be the current score, and ***C*** be a challenge result indicating whether the score should be increased (*C=true*) or decreased (*C=false*).

The increase and decrease factors are calculated as follows:
*increaseFactor = maxIncreaseFactor * (100 - S) / 100*
*decreaseFactor = maxDecreaseFactor * S / 100*

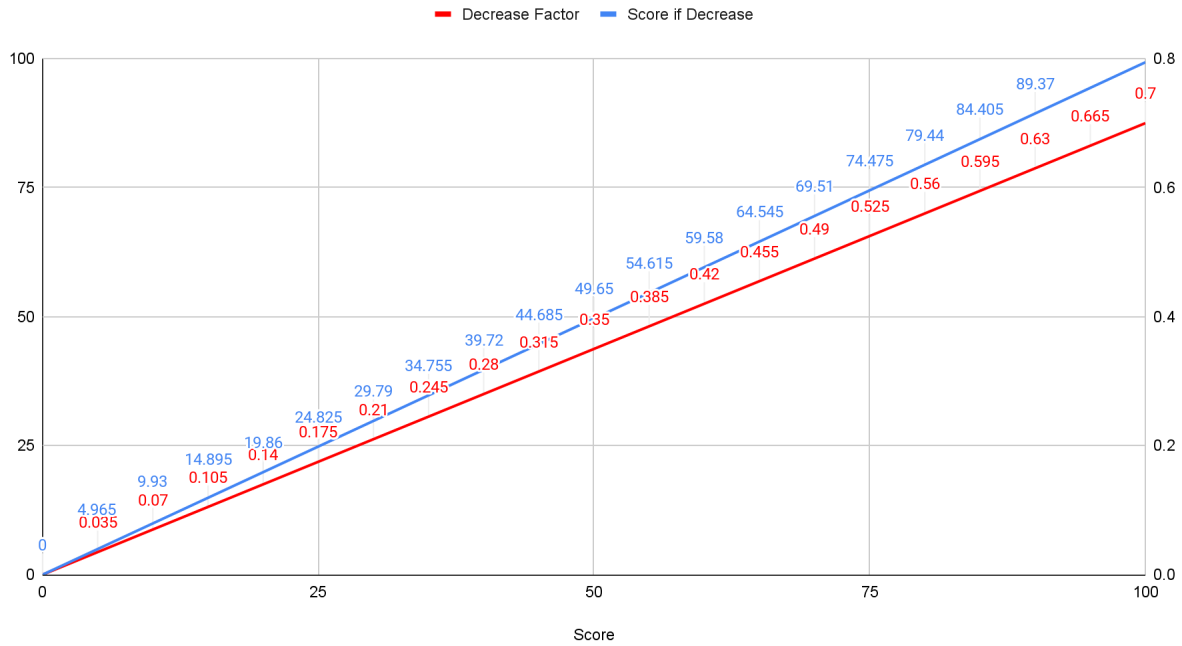
where *maxIncreaseFactor = 0.5* and *maxDecreaseFactor = 0.7* according to the current development network parameters. Note that main network parameters may differ, thus affecting these factors.

The new score is determined based on the outcome of the challenge. We check the value of the ***C*** variable, which represents the outcome of a challenge. If ***C*** is true, it means that the challenge was successful and the score should be increased. In that case, we check if the current score (***S***) is already close to the maximum score (**100**). If it is, the score is increased by a small amount. If the current score is not close to the maximum, the score is increased by a larger increase factor that was calculated earlier.

If ***C*** is false, it means that the challenge was unsuccessful and the score should be decreased. In that case, we check if the current score is already close to the minimum score (**0**). If it is, the score is decreased by a small amount. If the current score is not close to the minimum, the score is decreased by a larger decrease factor that was calculated earlier.



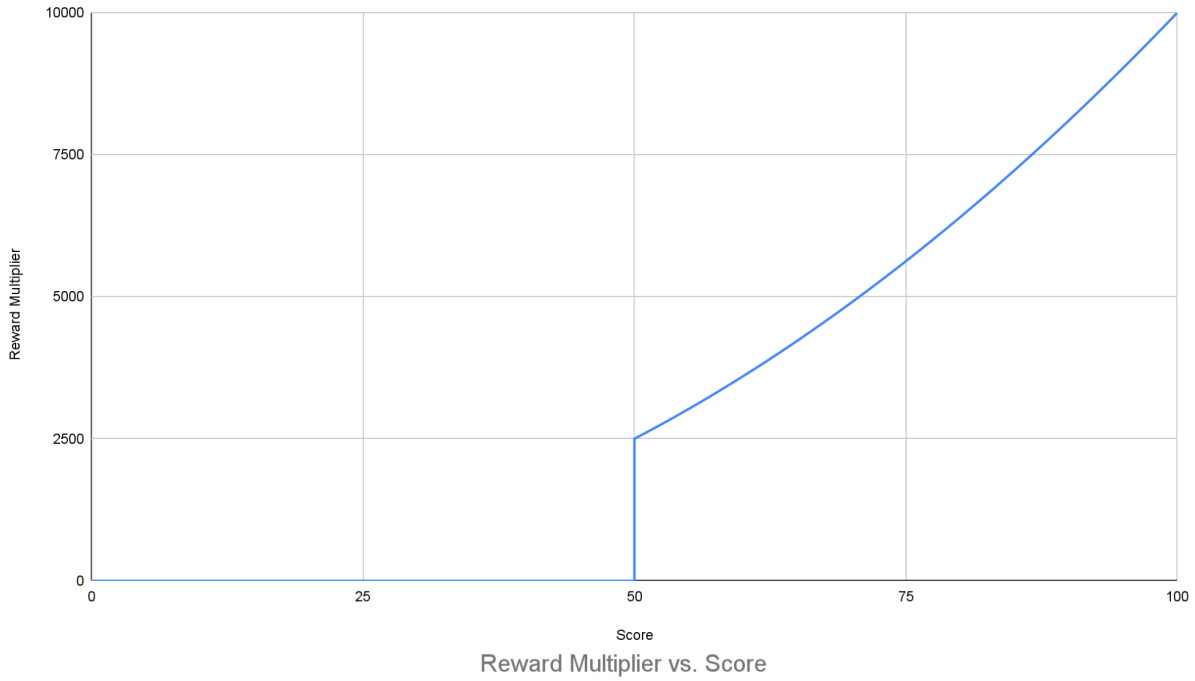
Increase Factor and Score if Increase



Decrease Factor and Score if Decrease

Vehicles don’t get any rewards if their score drops below 50. The relationship between the Reward Multiplier and Score is given as following:

rewardMultiplier = (S^{ω}) * (1 - (τ*ψ)) where ω, τ and ψ are network parameters. τ is calculated based on the timestamp of the last completed challenge and starts increasing after a threshold is passed, penalizing non-contributing behavior. ψ on the other hand, tracks the regularity of a vehicle's contributions and becomes infinitesimally small if a vehicle's contributions are more uniformly distributed. Currently τ and ψ are chosen to be 0 and $\omega = 2$



Reward Multiplier vs. Score

At each score epoch, the scores of the nodes are updated. At each reward epoch, the rewards are distributed according to a calculation which takes ***rewardMultiplier*** into account alongside some variables that depend on all the vehicles in the network. At any given reward epoch, the nominal reward that a vehicle will get is calculated according to the following formula:

(rewardMultiplier / Σ_{all}) * (epochRewards) where Σ_{all} is the sum of each vehicle's ***rewardMultiplier*** within the network and ***epochRewards*** is the number of tokens that will be emitted as rewards (initially configured as a network parameter). We refer the interested readers to go over the details at Soarchain documentation page[13].

Transactions

Soarchain transactions enable the execution of the proof-of-availability mechanism within the network. These transactions ensure that network participants are registered, locally-aligned and online, contributing to the overall health and stability of the network.
In this section, we will provide an overview of the transaction capabilities of Soarchain, including details on the types of transactions and how they are used to support the proof-of-availability mechanism[12].

i. MsgGenClient

The MsgGenClient transaction plays a significant role during the initialization process of the Motus device in the Soarchain network. It is responsible for the device's registration within the network, an essential step that enables the Motus device's functionality.

Once the device has been successfully registered through this transaction, it remains as a recognized entity in the network unless it is explicitly unregistered by the owner. Therefore, the transaction does not need to be invoked again unless the device's registration status changes, underscoring the enduring significance of this initial setup step.

Every Motus device registered in the network through the MsgGenClient transaction, is characterized by a unique set of properties that contribute to its operation within the Soarchain network. Each device is distinctly identified by its public key (pubKey) and address, which are required to be unique. Additional properties such as score, reward multiplier, and net earnings offer insights into the device's performance and financial attributes. The last time that the device was challenged and the cooldown tolerance property, offer details about the device's interactions within the network, while the type property signifies whether it's a Motus Mini or a Motus Pro device.

ii. MsgUnregisterClient

Allows users to remove their Motus devices from the network if they no longer wish to participate in the Proof-of-Availability mechanism. After unregistering, the Motus device will no longer be able to receive PoAv challenge requests or receive PoAv rewards.

iii. MsgGenRunner

Registers a guard object on the Soarchain network. This object is associated with a specific guard public key, and includes information about the v2xChallenger, v2nChallenger, and Runner, including their respective Soarchain addresses, stake amounts, and IP addresses.

By broadcasting this information through this transaction, network participants can easily verify the eligibility of the V2X and V2N challengers as well as runners to participate in the Proof-of-Availability mechanism, ensuring the integrity and security of the network.

For each guard public key, only one guard object can be registered.

Users can choose to register only specific types of guard objects among v2xChallenger, v2nChallenger, or Runner, by leaving the relevant fields empty in the transaction body. This allows users to register the specific types of guard objects that they are interested in, providing flexibility and customization in the registration process.

The MsgGenRunner transaction serves a crucial role in the Soarchain network's operations, specifically designed for the registration of the V2N Runner object. The V2N Runner object is an integral component

within the network, enabling the distribution of messages generated by Motus devices in the V2N network.

Each V2N Runner object is identified by a unique set of characteristics that contribute to its functionality and performance. The identifying public key (pubKey) and address must be unique to each Runner, ensuring the object's distinctiveness and integrity within the network. Moreover, each Runner features a score and a reward multiplier that reflect its performance metrics, while the staked amount and net earnings provide financial data. The IP address, the last time the Runner was challenged, and the cooldown tolerance are additional properties that inform about the Runner's network behavior and status. Together, these properties make up the V2N Runner's profile, supporting its operational stability and overall contribution to the Soarchain network.

The runner may be unregistered from the network only via a MsgUnregisterRunner transaction sent by the address which registered the runner in the first place.

iv. MsgRegisterChallenger

The MsgGenChallenger transaction forms an essential part of the validation process in the Soarchain network, serving to register a V2N or V2V challenger. This is a critical system function, acting to ensure the correct behavior of the network by challenging its participants, effectively functioning as a verification mechanism within the network's operation.

A V2N challenger's role is to pose challenges to the V2N runner, thereby verifying its correct behavior within the V2N network. Similarly, the V2V challenger is responsible for challenging the Motus devices operating within the V2V network, ensuring their correct operation and behavior within the network framework. Each challenger, whether V2N or V2V, is identified by a unique set of properties, which include a distinctive public key (pubKey) and address. Additional properties such as score, staked amount, and net earnings provide further information about the challenger's standing in the network. The type property indicates the kind of challenger, V2N or V2V, and its network interaction. The inclusion of an IP address serves to identify the challenger's network location. Together, these unique properties support the challenger's role in maintaining the integrity and accuracy of the Soarchain network's operation.

The challenger may be unregistered from the network only via a MsgUnregisterChallenger transaction sent by the address which registered the Challenger in the first place.

iv. MsgUnregisterChallenger

Allows users to remove their V2X or V2N challenger nodes from the network if they no longer wish to participate in the proof-of-availability mechanism. After unregistering, the challenger node will no longer be able to execute the PoAv challenge or receive challenger rewards.

This transaction can only be sent by the address that has registered the challenger, or else it would be rejected by the blockchain.

v. MsgUnregisterRunner

Allows users to remove their runner nodes from the network if they no longer wish to participate in the V2N process. After unregistering, the runner node will no longer be able to receive V2N rewards.

This transaction can only be sent by the address that has registered the runner, or else it would be rejected by the blockchain.

vi. MsgChallengeService

The MsgChallengeService transaction marks the subsequent stage after a challenger has engaged with a Motus device in the Soarchain V2X network. Its purpose is to report the outcome of the challenge to the blockchain, verifying the accurate functioning of the device. Only challengers who have registered in the network can initiate this transaction, following a successfully completed V2X challenge against a Motus device.

This transaction fundamentally includes the address of the Motus client and the result of the challenge, which can take one of two possible outcomes: "reward" or "punish". The consequences of these results are reflected in the client's score. A "reward" result, meaning the client was able to successfully fulfill the PoAv challenge, leads to an increment in the client's score, accompanied by a V2X challenge reward. In contrast, a "punish" result, meaning the client was not able to provide a valid response to the challenge, leads to a reduction in the client's score, with no financial penalty applied.

Once the challenge operation concludes, the V2X challenger's score also increases. This mechanism permits registered V2X challengers to assess the performance of Motus nodes within the network and appropriately reward or penalize them. As a result, this transaction contributes to maintaining the integrity of the PoAv mechanism.

The transaction also brings about modifications to the epoch rewards, the Client object (representing the Motus device), the associated wallet, and the Challenger involved in the challenge. These entities see changes in their score, net earnings, cool down tolerance (derived from a verifiable random number), and last challenged time based on the object type and the result of the challenge.

vii. MsgRunnerChallenge

The MsgRunnerChallenge transaction is a crucial function within the Soarchain network, operating as the verification mechanism for the behavior of V2N runners. It is the subsequent step after a challenger has engaged with a runner, and seeks to communicate the outcome of this engagement to the blockchain. It serves as an evaluative tool, verifying and reporting the integrity of the V2N runner's operations.

This transaction can only be initiated by registered V2N challengers as part of the proof-of-availability (PoAv) mechanism within the Soarchain network. It comprises the runner node's address and the challenge's outcome. The result can either be "reward" or "punish," indicating the runner's performance during the challenge. A "reward" outcome bolsters the runner's score and results in a V2N challenge

reward. Conversely, a "punish" result decreases the runner's score, yet it does not impose a financial penalty.

Once the challenge operation concludes, the V2N challenger also receives an increased score, recognizing its role in maintaining the network's operational integrity. The MsgRunnerChallenge transaction subsequently adjusts the scores, net earnings, cooldown tolerance (derived from a verifiable random number), and last time challenged of the challenger, runner, and participating Motus devices. These adjustments depend on the object type and challenge outcome. By enabling this evaluative and rewarding mechanism, this transaction fosters a healthy, competitive environment within the Soarchain network, driving high performance and network integrity.

Vehicle-to-vehicle Proof-of-availability(V2V PoAv)

One of the most important aspects of Soarchain is the vehicle-to-vehicle connectivity that enables geographically proximate vehicles(and any node that contains V2V functionality on the hardware level) to communicate application-specific data to each other without depending on any type of fixed infrastructure or direct line-of-sight between the vehicles. Through this mechanism, our goal is to create a global fleet of locally-aligned vehicular ad hoc networks that communicate mission-critical data to each other and unlock countless road safety, carbon-emission reduction and collective perception applications.

We propose a method to verify that $\boldsymbol{n_{bx}}$ number of messages have been broadcast by a Motus device by only validating $\boldsymbol{p_{bx}}$ number of those messages where $log_{\boldsymbol{pbx}}(n_{bx}) > f_{bx}$ and $\boldsymbol{f_{bx}}$ is a network security parameter, as well as verify $\boldsymbol{n_{rx}}$ number of messages have been independently and correctly received and processed by a Motus device by only validating $\boldsymbol{p_{rx}}$ number of those messages where $log_{\boldsymbol{prx}}(n_{rx}) > f_{rx}$ and $\boldsymbol{f_{rx}}$ is a network security parameter.

Since all devices capable of V2V communication are required to both broadcast(bx) and receive(rx) messages, Soarchain incentivizes honest behavior for both of these modes for each device, and each device is expected to participate and provide proofs for both V2V Broadcaster Challenge and V2V Receiver Challenge. Note that the names are given according to their modes of V2V communication, therefore a “receiver” can also be found “sending” data to the network, which is realized through the cellular network rather than the V2V interface.

Broadcaster Procedure A

1. After the device boot sequence is complete, the first CAM is generated. CAMs are generated as long as a Motus device is running. Each CAM from an individual broadcaster is expected to be broadcast at 1 message/second. CAM is created by the CA basic service module within the dITS (decentralized Intelligent Transportation System) software stack where one of its core functionalities is writing the data received from various sources such as CAN bus, GNSS, IMU etc. into CAM fields and encapsulating them according to the message definition. Soarchain uses a slightly different version of the CAM than the version that’s specified in [9], where

blockchain-related fields are appended. Message and CAM are used interchangeably in this section.

2. Until the number of generated messages reaches n_{bx} , a Merkle root is computed by utilizing the hash of each message as the leaves of the Merkle tree. The set of leaves can be represented as $\{m_j, m_{j+1}, \ldots\ldots m_{nbx}\}$ where $j=1$. This marks the first instance that a broadcaster becomes challengeable. At each subsequent message a new Merkle root is computed from the new set of messages where the size of the set S_{bx} remains the same where $S_{bx} = n_{bx}$, i.e. the Merkle root is computed from the set $\{m_{j+1}, m_{j+2}, \ldots\ldots m_{nbx+1}\}$ when the first message after n_{bx} is generated.
3. Hash of the current message, hash of the previous message, computed Merkle root, and raw CAM data is written inside ***bxObject***.
4. ***bxObject*** is recorded on the device's onboard storage in a database.
5. ***bxObject*** gets signed by the Hardware Security Module of the device and becomes ***bxPacket***.
6. dITS then sends the ***bxPacket*** to lower layers to finally reach the physical layer to be broadcast through the air interface.

The whole procedure is repeated for each newly generated message.

Receiver Procedure A

1. After the ***bxPacket*** is received by the receiver physical layer it is sent up to the dITS.
2. Signature verification is done for the signature inside the ***bxPacket***
    a. If the signature is not verified then the message is dropped and recorded to the device's onboard storage in a database. Optionally, an Issue Report transaction can be submitted to the validators.
    b. If the signature is verified then the ***bxPacket and its hash*** is recorded to the device's onboard storage in a database
3. The number of ***bxPacket*** in the receiver's database is checked whether it has reached n_{rx} or a time t_{rx} has passed. Let the size of the ***hashed bxPacket*** set be S_{rx} where $S_{rx} \leq n_{rx}$**.** The next step is executed if either one of the conditions becomes true.
4. A Merkle root is calculated by using the set of S_{rx} hashed bxPacket as leaves.
5. ***bxPacket*****,** computed Merkle root, and size of the claim are written into the ***rxObject*****.**
6. ***rxObject*** gets signed by the Hardware Security Module of the device and becomes ***rxClaim*****.**
7. dITS sends the ***rxClaim*** to a randomly* chosen challenger.
8. S_{rx} counter is reset to 0.

The whole procedure is repeated until $S_{rx} = n_{rx}$ or time t_{rx} has passed. Note that we assume that a challenger can be randomly chosen and the details are omitted in this section.

Challenger Procedure A

1. A challenger receives the ***rxClaim***.

2. Signature of the Receiver which has sent the ***rxClaim*** is verified and signature of the Broadcasters inside the ***rxClaim*** is verified.
    a. If the signatures are not verified then this is recorded to the challenger's database. Optionally, an Issue Report transaction can be submitted to the validators.
    b. If the signature is verified then move on to the next step.
3. The blockchain is queried for the challengeability of the receiver and input it to the challengeability check function.
    a. If the node is not challengeable, then the process is skipped to step 5 in Challenger Procedure B
    b. If the node is challengeable, then the next step is executed.
4. $\boldsymbol{p_{rx}}$ random integers chosen by the challenger from a set of $\{1, 2, \ldots n_{rx}\}$ where $\boldsymbol{p_{rx}}$ is chosen to be a negligibly-small number in comparison to n_{rx}
5. The chosen random integers correspond to the indexes of the requested Merkle leaves among the set that comprises the latest $\boldsymbol{S_{rx}}$ number of leaves(generated messages). The ***rxChallenge*** is created which contains the ***rxClaim*** and ***integers that represent the indexes of the leaves of the Merkle tree***.
6. ***rxChallenge*** is recorded to the challenger's database and a unique challenge ID is added to the ***rxChallenge*** structure.
7. ***rxChallenge*** is signed and becomes the ***rxChallengePacket***
8. ***rxChallengePacket*** is sent to the device that created the initial ***rxClaim***

Receiver Procedure B

1. ***rxChallengePacket*** is received from the challenger
2. Merkle root is extracted from the ***rxClaim*** inside the ***rxChallenger***
3. ***rxClaimGroup*** document is found from the local database, which corresponds to the exact value of the Merkle root
4. A Merkle tree is created from the message hashes which were previously stored in ***rxClaimGroup***.
5. Merkle proofs for the leaves that correspond to the indexes specified inside the ***rxChallengePacket*** are retrieved.
6. ***rxChallengeResponse*** object is created that comprises challenge ID, proofs of the indexed leaves, and the array of the proofs.
7. ***rxChallengeResponse*** is signed and turned into ***rxChallengeResponsePacket***
8. ***rxChallengeResponsePacket*** is broadcast to the challenger network.

Challenger Procedure B

1. A challenger received ***rxChallengeResponsePacket***
2. Receiver and Broadcaster signatures are verified
    a. If the signatures are not verified then this is recorded to the challenger's database. Optionally, an Issue Report transaction can be submitted to the validators.
    b. If the signature is verified then move on to the next step.

3. ***proofLeaves*** and ***proofArray*** are extracted from the data structure and checked to see if values match the ones requested in ***rxChallenge***
    a. If values do not match then this is recorded to the challenger's database. Optionally, an Issue Report transaction can be submitted to the validators.
    b. If the values match then move on to the next step.
4. From the Merkle proofs, the root is recreated and compared to the initially received root.
    a. If values do not match then this is recorded to the challenger's database. Optionally, an Issue Report transaction can be submitted to the validators.
    b. If the values match then move on to the next step.
5. n_{ch} number of challengers are requested from the challenger network where n_{ch} is the number of Broadcasters which are found to be eligible for the challenge operation.
6. Identifiers of Broadcasters are shared with the respective challengers where each challenger gets a single Broadcaster address to send a challenge to.

Challenger Procedure C

Note that this procedure is executed by each and every challenger that received the Broadcaster addresses from Challenger Procedure B

1. Index value from the ***bxObject*** inside ***bxPacket*** is extracted
2. p_{bx} number of random index are specified where p_{bx} chosen to be a negligibly-small number in comparison to n_{bx}
3. ***bxChallenge*** object is created with ***target*** and ***leaves*** inside of it, where they respectively represent the ***bxObject*** inside ***bxPacket*** and ***the array of indexes for randomly specified Merkle leaves requested from Broadcasters***
4. ***bxChallenge*** is recorded into the database and a corresponding ***challengeId*** is put inside it from the database
5. ***bxChallenge*** is signed and turned into ***bxChallengePacket***
6. ***bxChallengePacket*** is sent to the corresponding Broadcasters

Broadcaster Procedure B

Note that this procedure is executed by each and every broadcaster that received the bxChallenge from Challenger Procedure C

1. The target is extracted from the ***bxChallengePacket*** and the corresponding leaves and previous n_{bx} number of leaves are retrieved from the local database
2. A Merkle tree is created from these leaves (of which the root should be identical to the Merkle root inside the target but that's not checked here)
3. ***leaves*** array is extracted from the ***bxChallengePacket*** and Merkle proof of each of the items inside it is created
4. ***bxChallengeResponse*** is created with challengeID, proofLeaves and proof inside of it.

5. ***bxChallengeResponse*** is signed to become ***bxChallengeResponsePacket***
6. ***bxChallengeResponsePacket*** is sent to the challenger that has initially sent the ***bxChallengePacket***

Challenger Procedure C

Note that this procedure is executed by each and every challenger that received the ***bxChallengeResponsePacket*** from its corresponding Broadcaster in Broadcaster Procedure B

1. ***bxChallengeResponsePacket*** is received from the Broadcaster
2. Broadcaster signature is verified
    a. If the signature is not verified then this is recorded to the challenger's database. Optionally, an Issue Report transaction can be submitted to the validators.
    b. If the signature is verified then move on to the next step.
3. ***bxChallenge*** is retrieved from the database using the ***challengeID*** inside the ***bxChallengeResponsePacket***
4. extract ***proofLeaves*** and ***proof*** to check to see if received values are the ones requested in the ***bxChallenge***
    a. If false then return. Optionally, an Issue Report transaction can be submitted to the validators.
    b. If true, move on to the next step.
5. Merkle proofs inside the ***bxChallengeResponsePacket*** is verified
    a. If not verified, return. Optionally, an Issue Report transaction can be submitted to the validators.
    b. If verified, issue a successful transaction.
6. Mark the ***bxChallenge*** inside the local database as complete.
7. Finally, a reward transaction is submitted to the network which includes the participants of the whole challenge process and their respective reward percentages.

Vehicle-to-network Proof-of-Availability (V2N PoAv)

Another very significant aspect of Soarchain is the vehicle-to-network connectivity that enables any vehicle or road user to communicate application-specific data to the consumers of that data through any protocol that can act as a relay to the internet. Through this mechanism, our goal is to create a global fleet that streams large amounts of data to the cloud and unlock countless data collection and device management applications as well as third-party services.

V2N PoAv works similar to the V2V PoAv, except that the transmission of data is done to a decentralized entity called "runner", which is a cloud server that is powered by the people of the network. They are responsible for receiving messages from the V2N broadcasters and keeping the messages in their

database temporarily. Once they become ready to be challenged, they send a claim that contains the last message in its buffer, the Merkle root that is generated from the messages, and some other metadata to the "V2N Challengers", which sends back a challenge to the runner, asking for a random set of messages that are used to create the Merkle tree that was shared with it in the first place. This part is very similar to the V2V PoAv procedures that take place between the Receiver and the Challenger(Runner can be thought of as the Receiver in V2V PoAv).

Consensus

Most of the heavy duty operations such as continuously creating a new Merkle tree from each new message generated and received, as well as recreate parts of the Merkle trees whenever demanded by the challengers are all handled by Motus devices participating in the network. Once all challenge response data have been input back to a group of randomly chosen challengers by the message generating and receiving parties, default verification steps are executed on the data by the challengers such as signature verification, plausibility of received signal's attributes, location verification etc. While doing these operations, the challengers create a verifiable computation fingerprint to be later on verified by the validators, as part of the built-in logic. After the input data passes the default verification steps and the computation fingerprint is created, validators start the consensus process and try to achieve consensus on the proposed block data received from the challengers. Once the supermajority of the validator nodes reach consensus on the input data, the reward distribution process starts on the validator network and the block is finalized. For more information block proposal, block generation, block finalization and consensus, please refer to Tendermint research paper[14].

4. Ecosystem

Road Users

Four-wheelers

In recent years, vehicles have become increasingly equipped with sensors, cameras, and computers, almost like moving servers with perception and communication abilities. As a result, the vehicle paradigm is changing as vehicles are becoming more software-defined. This shift is opening up new business opportunities and revenues, as well as enabling game-changing features such as zero-accidents and autonomous driving. These advanced technologies are transforming the way we think about transportation and have the potential to revolutionize the way we travel. They offer the potential for increased safety, efficiency, and convenience, as well as the opportunity for new business models and revenue streams. As technology continues to advance, cars will be at the forefront of this revolution and it is likely that we will see even more significant changes in the way we use and interact with vehicles.

Vulnerable Road Users

A vulnerable road user is a person who is at a higher risk of being involved in a traffic accident due to their mode of transportation or their physical or cognitive abilities. Examples of vulnerable road users include pedestrians, bicyclists, motorcyclists, and people with disabilities. Especially with the increasing modes of urban transport, the growing number of vulnerable road users and their safety becomes increasingly important yet difficult to maintain. They are incentivized to participate in the network through the reward mechanism for making their trip information available to the relevant road users that they have a high probability of interacting with and might pose risk to them.

Challengers

V2V Challengers

This is a node that expects data to be verified from a V2V receiver that claims to have generated an array of specific messages. V2V challengers can be run on Motus devices, smartphones, and any type of resource-constrained device, and anyone can stake a small amount of tokens to become a V2V challenger. Challengers receive the challenge requests from the participating nodes according to their score, which is calculated through a formula that takes their uptime and integrity into account.

V2N Challengers

This is a node that expects data to be verified from a Runner that claims to have generated an array of specific messages. V2N challengers can be run on Motus devices, smartphones, and any type of resource-constrained device, and anyone can stake a small amount of tokens to become a V2N challenger. Challengers receive the challenge requests from the participating nodes according to their score, which is calculated through a formula that takes their uptime and integrity into account.

Validators

A validator is a special type of node that is responsible for participating in the consensus process to reach agreement on the state of the blockchain. Validators can be thought of as the “miners” of the blocks, as they are responsible for ensuring the integrity and security of the network. Validators are chosen by the community through a staking process, and they are rewarded for their participation in the consensus process. Soarchain is a public Proof-Of-Stake (PoS) blockchain, which means that the weight of validators is based on the quantity of staking tokens ($MOTUS) bonded as collateral. These MOTUS tokens may be self-delegated by the validator or delegated by other MOTUS holders to the validator.

By sending a create-validator transaction, every user in the system can declare their intention to become validator candidates. A validator's weight (voting power) decides whether they are an active validator. The active validator set is restricted to an amount which may slightly change over time.

Runners

A runner is a special type of virtualized generic compute node, which receives structured data from the network participants to run computations over the data. The validity of the data can be verified by the V2N Runner Challengers through the challenge mechanism with a special procedure if the data is received from a V2N Broadcaster to participate in V2N rewards. Otherwise, it can be directly verified through a verifiable computational fingerprint that the runner generates. Runners can be spun up on Soarchain by sending a create-runner transaction and staking the minimum amount of tokens required for a single runner.

Runners have the ability to execute generic computations and generate computational proofs from these computations that prove to the validator network that exact computations demanded by the network. Runners can be utilized as trusted parties and can be deployed by any node that needs to outsource their computation. Alternatively, runners support a trustless set up where they can compute over encrypted data and return the output to the owner without ever decrypting it.

Developers

Developers are an extremely important part of the Soarchain ecosystem. There are multiple functions assigned to the developers, ranging from testing, prototyping, developing and deploying different decentralized applications on Soarchain for various mobility use cases. Additionally, Soarchain core development will also benefit from individual developers' effort to maintain the networking, blockchain, consensus, application logic and various other layers of the technology stack. Furthermore, they can also help create and distribute SDKs, libraries and APIs for developers to interact with the Soarchain platform. Developers can also contribute to the development of the DApps and Smart Contracts that run on Soarchain, helping make them more secure, reliable, and efficient. Finally, developers can help to build a community of developers and users around the Soarchain platform, creating a vibrant ecosystem of innovation and collaboration.

Industry

Many industries will leverage Soarchain to get the most out of their use cases by utilizing the mobility data that is generated by the contributing nodes of Soarchain, and deploy their decentralized applications through smart contracts to interact with these nodes. These industries include but are not limited to; financial services such as insurance, bank and loan provision, vehicle leasing, and other lending; OEMs and motor industry such as vehicle manufacturers, automotive supply chain, dealerships and other OEM partners and vehicle servicing providers; others such as energy suppliers, businesses with vehicle fleets, retail, entertainment, and media, and telecommunications; public services such as municipal authorities, emergency services, central government, law enforcement and urban planners.

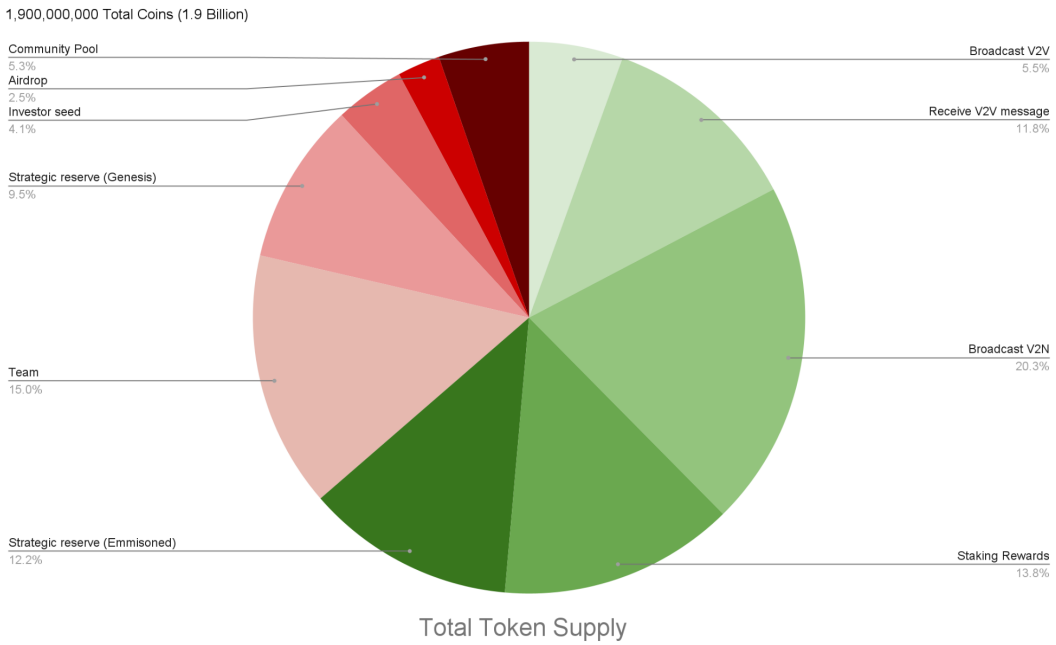
5. Economic Model

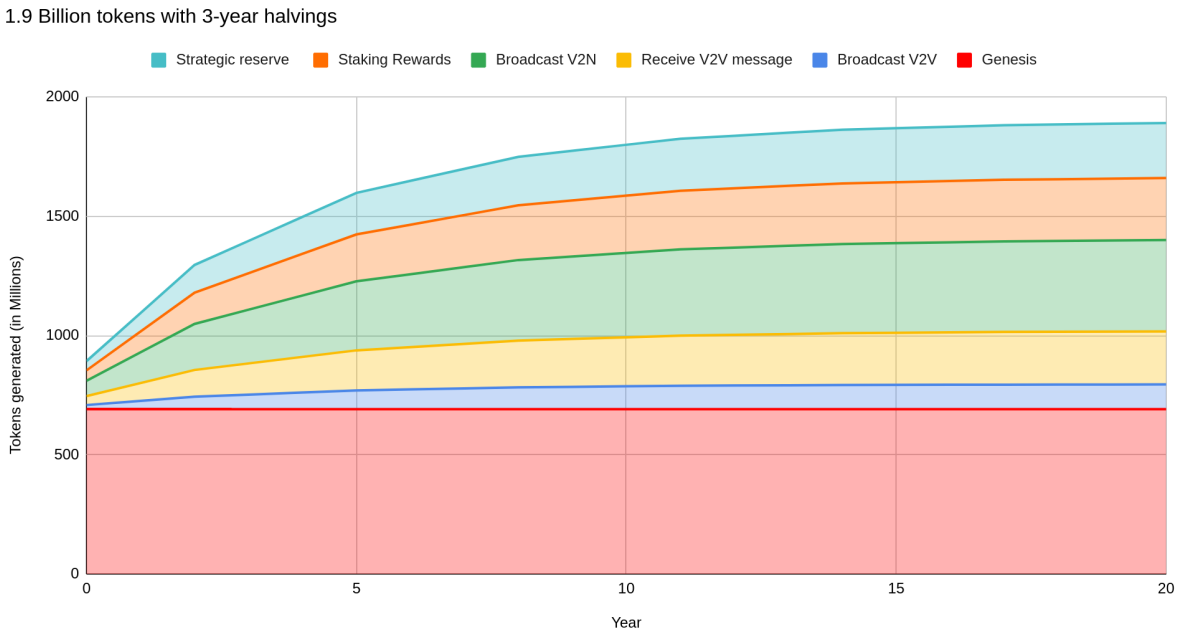
Soarchain's token economy is based on using $MOTUS coins as a means of exchange and as a reward for participating in the network. This innovative approach enables the platform to power the secure execution of smart contracts and dApps while providing data and transaction fees to its nodes.

The primary purpose of $MOTUS coins is as an exchange currency throughout the Soarchain network. They can be used to pay for services and dApps or to pay data and transaction fees to nodes. In addition to being used as an exchange currency, $MOTUS coins are rewards for participating in the Soarchain network by making vehicles' connectivity and computational resources available to the network.

Every year, a fixed number of tokens are minted by the system and then distributed amongst participants who prove their availability within the network through Proof of Availability (PoAv). This ensures that users who actively contribute to the security and stability of the platform are rewarded accordingly.

Our token economy incentivizes participation with a 54-year distribution of 1.9 billion tokens, halved every three years to ensure an equitable balance in rewards allocation. Transaction fees are burned, and the genesis supply is 36.4%. In comparison, 63.6% will be distributed as a reward for network participants - providing long-term sustainability into our economics model development that encourages continued engagement within these parameters over time. These 63.6% is mainly attributed to three group: i) V2N rewards %20.3, ii) V2V rewards %17.3 and iii) staking rewards %13.8.





Emission Schedule

Soarchain's unique token model provides users with a secure way to exchange value within the platform and receive rewards for participating in its network. Having both uses for its tokens creates an incentive structure that encourages active engagement from those involved for them to reap maximum benefits from their interaction with the Soarchain network. This balance between usage and reward makes Soarchain stand out.

6. Use Cases and Applications

Soarchain enables the development of smarter, more coordinated, and efficient transportation networks with the potential to increase safety for travelers and the public, minimize environmental impact, and improve traffic management. Soarchain enables the set of technologies where vehicles exchange information with other road users and the surrounding infrastructure. Applications include road safety, traffic control, fleet management, location-based services, driver assistance, hazard warnings, and support for emergency services, ultimately leading to the realization of fully autonomous driving and fleet autonomy.

Soarchain also enables the development of decentralized applications, known as Dapps, which can run on top of the Soarchain platform through native smart contracts. These Dapps can provide a wide range of services, from on-demand ridesharing to real-time traffic updates, fleet management to autonomous driving. Dapps offer secure and transparent services that can be accessed by anyone with an internet connection and a Soarchain wallet. This will further enhance the coordination and efficiency of transportation networks, as well as provide new opportunities for businesses and individuals to develop

innovative solutions to transportation challenges. Some of the use cases that are unlocked are listed and elaborated in the section below.

Road Safety

Some of the road safety applications that Soarchain enables include collision avoidance, emergency vehicle notification, and vulnerable road user safety.

Collision Avoidance

Soarchain technology can be used to help vehicles detect potential collisions and take appropriate action to avoid them. For example, a vehicle equipped with Soarchain technology is able to receive a warning from another vehicle about a stationary vehicle on the road ahead, and the driver could take evasive action to brake immediately.

Emergency Vehicle Notification

Soarchain technology enables emergency vehicles, such as ambulances and fire trucks, to alert other vehicles on the road when they are approaching. This can help other vehicles to quickly move out of the way, allowing the emergency vehicle to reach its destination more quickly.

Emergency vehicle notification is typically implemented using V2V (vehicle-to-vehicle) communication, which allows vehicles to exchange information with each other in real time. When an emergency vehicle is approaching, it can use V2V communication to send a warning to other vehicles in the area. This warning can be displayed on the other vehicles' dashboard or navigation system, alerting the driver to the presence of the emergency vehicle and prompting them to move out of the way.

Vulnerable Road User Safety

Soarchain technology can be used to improve safety for pedestrians by alerting drivers when a pedestrian is nearby. For example, a vehicle could receive a warning from a pedestrian's smartphone about their presence and take appropriate action to avoid them. Vulnerable road users are particularly at risk in traffic because they are often less visible to other road users and are less protected in the event of a collision. For example, pedestrians and bicyclists are not enclosed in a vehicle and do not have the same level of protection as car occupants, making them more vulnerable to injury in the event of a collision and Soarchain technology can help to reduce this risk by providing drivers with additional information about the presence and movements of vulnerable road users.

Vulnerable road user safety is implemented using Soarchain’s V2V (vehicle-to-vehicle) and V2I (vehicle-to-infrastructure) communication. For example, a vehicle equipped with V2V technology could receive a warning from another vehicle or a pedestrian's smartphone about a vulnerable road user in the area. The vehicle could then use this information to adjust its speed and route to avoid the vulnerable road user, improving their safety.

Data Collection

Soarchain is developed to adhere to the highest industry standards for mobility. It employs a distributed ledger technology and relies on a decentralized network in which no entity stores or controls the data. Collected data remains in encrypted form off-chain, with just its cryptographic signatures retained on-chain. Only those entities with permission from the data's owners can access the data. The integrity and validity of the data is ensured by the cryptographic signatures stored on-chain. Moreover, Soarchain employs a permissioned ledger and enables private transactions. This enables the owner-approved disclosure of specific data to specific third parties. Any type of data that vehicles and road users generate can be utilized by the interested parties while preserving the privacy of the users.

Diagnostics and Predictive Maintenance

In the case of diagnostics, Soarchain technology could be used to collect and transmit real-time and retrospective diagnostic data both in pre-processed and raw form from vehicles, allowing for more efficient and accurate analysis of vehicle performance. This could be particularly useful for detecting and diagnosing problems before they become more serious and expensive to fix, thus saving thousands of dollars in total and increasing the vehicle's service life.

Connectivity Performance

Participant nodes of Soarchain are connected to each other and to the cloud through various physical layer protocols such as cellular, Wi-Fi, V2V and Bluetooth. It is very important for infrastructure providers to assess the health of their wireless networks, especially when it is serving highly-mobile nodes such as cars and other road users. Network participants can choose to monetize their connectivity data such as signal metrics, channel quality index, and many other variables that telecom operators are willing to pay for.

Perception/ML Data

Soarchain facilitates direct and secure access to sensor and camera data that are generated by the vehicles in its network and allow the vehicle owners to anonymously share this data to get rewarded in return. Soarchain’s novel data scaling solution can support the pre-processing, storage and sharing of large amounts of vehicle data that could include information about the environment, vehicle performance, driver behavior and more without compromising user privacy while ensuring data quality, integrity and authenticity. A reference to this data can be securely recorded and stored, ensuring that the data is immutable, traceable, and easily verifiable. Use cases for this data can range from developing new deep learning algorithms for self-driving capabilities to creating applications that monitor the conditions of the road and even creating an HD map of the environment that the vehicles have been driving in.

Road Conditions

An HD map of the environment might also include an accurate and recent snapshot of the road conditions, including the severity of potholes, the amount of traffic, road works, and the presence of any obstacles that might interfere with the secure driving of the vehicles or transportation of the vulnerable road users. To ensure the accuracy of

Passenger/Driver

Device Management

As Soarchain creates a secure, decentralized, private and trustless way of connecting a vehicle to other endpoints, management of all the connected devices must be done in a secure and robust way. Soarchain can potentially become a platform to deploy decentralized and upgraded versions of today's remote configuration, monitoring, and over-the-air (OTA) software updates solutions on the market.

Remote Configuration

Remote configuration can be a time consuming and error-prone process. However, with Soarchain's vehicle-to-network technology and smart-contracts supporting blockchain, this process can be automated, secure, and reliable. By leveraging the distributed ledger, smart contracts can be used to automatically configure settings, devices, and networks. Furthermore, since each transaction is stored on the distributed ledger, it can be verified and audited to ensure that the configuration is secure and compliant with the appropriate regulations.

Remote Monitoring

Soarchain can also be used to monitor remote systems in a secure and reliable manner. By leveraging the distributed ledger and decentralized identifiers(DID), organizations can track and monitor remote systems in real-time, ensuring that all data is secure, accurate, authentic and integral. Furthermore, the distributed ledger can also be used to store data related to the systems, such as system configurations, user access, and other relevant information.

Secure Native OTA

Similar to Remote Configuration, Soarchain can also enable over-the-air (OTA) software updates through its vehicle-to-network technology and smart-contracts supporting blockchain, and even facilitate vehicle-to-vehicle OTA without needing to backhaul to an internet-connected server. By leveraging the distributed ledger, organizations can ensure that all updates are cryptographically signed and verified before being deployed. Furthermore, it can also be used to store metadata related to the updates, such as version numbers, security patches, and other relevant information.

Collective Perception

Collective perception applications refer to the use of Soarchain's technology to enable vehicles to share and combine sensory data to improve their understanding of their surroundings. Some examples of collective perception applications include platooning, collective sensor fusion, and flow optimization.

Platooning

Soarchain enables platooning, which involves groups of vehicles traveling and closely together in a coordinated manner, with each vehicle receiving information from the other vehicles in the platoon to improve their understanding of the surrounding environment. This can help to improve the efficiency and safety of the platoon by enabling the vehicles to react to changes in their surroundings more quickly and effectively.

Soarchain's direct vehicle-to-vehicle (V2V) communication is a key technology that enables platooning. By using V2V communication, vehicles in the platoon can exchange information about their speed, position, and intended actions with each other in real time. This allows the platoon to maintain a closely spaced formation and to coordinate their movements, allowing them to react to changes in their surroundings more quickly and effectively.

For example, suppose one vehicle in the platoon needs to brake suddenly to avoid an obstacle. In that case, it can use V2V communication to send a warning to the other vehicles in the platoon. This allows the other vehicles to brake more quickly, reducing the risk of a collision and improving the safety of the platoon.

Soarchain's V2V communication is essential for enabling platooning, as it allows vehicles in the platoon to exchange information and coordinate their movements in real time, improving the efficiency and safety of the platoon.

Collective Sensor Fusion

Soarchain enables collective sensor fusion which involves vehicles sharing sensory data with each other, such as information from their cameras, radar, and lidar sensors. This data can be combined and used to improve the accuracy and reliability of the vehicle's understanding of its surroundings, allowing it to make more informed decisions about how to navigate the environment.
Soarchain's direct vehicle-to-vehicle (V2V) communication is a key technology that enables collective sensor fusion. By using V2V communication, vehicles can share their sensory data with each other in real time. This allows the vehicles to combine their sensory data and create a more complete picture of their surroundings, improving their understanding of the environment and allowing them to make more informed decisions about how to navigate it.

For example, if one vehicle has mapped the road ahead, it can use V2V communication to share this information with the other vehicles in the group. This allows the other vehicles to take appropriate action to avoid the obstacle, improving the safety of the group.

Flow Optimization

Soarchain enables flow optimization which involves using vehicle-to-vehicle(V2V), vehicle-to-infrastructure(V2I) and vehicle-to-network(V2N)technology to coordinate the movement of vehicles in a way that optimizes the flow of traffic. For example, vehicles communicate with each other and with traffic signals to coordinate their speed and lane changes, reducing congestion and improving overall traffic flow. This

Soarchain's both direct vehicle-to-vehicle (V2V) communication and vehicle-to-network (V2N) communication are essential for enabling flow optimization. V2V communication allows vehicles to exchange information with each other in real time, allowing them to coordinate their movements and avoid congestion and drive cooperatively. V2N communication, on the other hand, allows vehicles to communicate with traffic signals and other infrastructure, allowing them to adjust their speed and route to optimize the flow of traffic.

For example, if a group of vehicles is approaching a traffic light that is about to turn red, they can use V2V communication to coordinate their speed and position to avoid having to stop at the light. This helps to reduce congestion and improve the flow of traffic, thus reducing overall energy consumed and carbon emissions.

Services

Soarchain's blockchain-enabled V2X technology, combined with its decentralized and privacy-preserving data streaming solution will enable countless third-party services which will maximize the benefit both the user and service provider.

Insurance

One of the main ways that Soarchain will improve automotive insurance is by providing real-time and retrospective data about road conditions and traffic. With this information, insurers can more accurately assess the risks associated with a given vehicle and its surroundings, and adjust premiums accordingly. This can lead to more fair and accurate pricing for insurance policies.

Additionally, Soarchain enables drivers to privately share their driving data. For example, if a vehicle is equipped with Motus hardware, which has sensors and other technologies that can monitor things like speed and acceleration, insurers can use this data to assess the riskiness of a driver and adjust their premiums accordingly. This will incentivize safer driving behavior, which can lead to fewer accidents and lower insurance costs overall.

Finally, Soarchain will enhance the claims process for automotive insurance. For example, if a vehicle is involved in an accident, blockchain records can automatically provide data about the incident to the insurer, which can speed up the claims process and make it more efficient. This can make it easier for

policyholders to receive the compensation they need after an accident, and can help insurers resolve claims more quickly and efficiently.

EV Charging

Soarchain could be used to create a smart contract that specifies the terms of the payment for EV charging services. This contract could be automatically executed when a vehicle connects to an EV charging station, and the payment for the charging services could be automatically transferred from the vehicle owner's account to the charging station operator's account. This would enable a decentralized and automated payment system for EV charging services without the need for a third party to facilitate the transaction.

Teleoperation

Teleoperation services, which allow users to remotely control their vehicles, can be securely and efficiently enabled through Soarchain’s V2X technology. To ensure that the vehicle teleoperation services are compliant with the relevant regulations and laws,. This will ensure that the services are compliant with the relevant regulations and laws, which will help to protect users and ensure that the services are reliable and secure.

Leasing and Loans

Soarchain’s built-in digital identity features will enable vehicle leasing companies in multiple ways. The privacy-preserving open ledger will allow the providers to securely and transparently record the terms and conditions of a lease or loan agreement, as well as any changes that are made to the agreement over time. This will help to reduce the cost, complexity, and risk of fraud or error associated with the leasing or lending process.

Soarchain can also facilitate the tracking and management of vehicle maintenance and repairs during a lease or loan through recording the relevant information on the ledger, which can then be accessed by all relevant parties. This will ensure that the vehicles are properly maintained and in good condition throughout the leasing or lending period.

Shared Mobility

A shared mobility application that is built on Soarchain can operate without a central authority, allowing for more equitable distribution of resources and decision-making power, thus benefiting all parties involved in the system. The default privacy-enabled

Data Provision Request

This is a special type of Dapp, which is unique to Soarchain and is natively supported through a special type of smart contract logic. In essence, it is a secure, decentralized and efficient way for data generators to share their data with data consumers. The specific flow is as follows:

1. Vehicle drivers, passengers, pedestrians and other road users are the generators of data, which is very valuable to the consumers of this data. Data consumers include but is not limited to: insurers, loan providers, vehicle leasing providers, banks and lenders, OEMs(vehicle manufacturers), Tier-1 and other suppliers, automotive supply chain optimization, vehicle servicing providers, dealerships, energy suppliers, businesses with vehicle fleets, retail, entertainment and media, telecoms.
2. Data generators don't know which of their data is valuable to whom, and when. For this, the data consumers should have a specific demand for this data. Soarchain enables "Data Provision Requests (DPR)," in which the data consumers can specify what type of data they want, in which time period, with what frequency and with which level of reliability(alongside a long list of these parameters). This feature, native to Soarchain, enables data consumers to open customizable DPRs on Soarchain.
3. DPRs are issued by staking a certain amount of Soarchain's native tokens proportional to the total value of that DPR, which is dynamically calculated and depends on the number of vehicles, size, quality, availability quotient and recency of the specific data.
4. Once the DPR request transaction is issued on-chain, data generators get notified of this DPR, and their Soarchain-connected devices automatically check their eligibility. Once they pass the eligibility check, they opt-in to sharing their data with the data consumer that has submitted the DPR. They can choose whatever they want to share, as long as they fulfill a minimum threshold (e.g., if the submitter of the DPR requests a minimum data resolution of 100 data points per day). They get rewarded proportionally to the contribution they make to the DPR. (Note that a person can choose to have their opt-in settings set to "Join all when eligible" by default on their wallets, which would automatically join any DPRs that they are eligible for.
5. Eligible data consumers issue a "join" transaction to the DPR and become responsible for providing the minimum data that is requested from each participant. Staking rewards are given out in proportion to the number of tokens staked to the DPR, and these rewards are sent to the data generators based on how much they contributed.
6. A DPR can be terminated if the predetermined time or data limit has been reached.

Note that the system described above assumes that the potential submitter of the DPR demands future data and not retrospective data. As it is extremely costly to store all data that is generated by all vehicles at all times, it makes more economical sense to design a system where requests are aligned with the future needs. It is possible that a certain Dapp on Soarchain could enable each vehicle owner to store all their data continuously and permanently on some decentralized storage solution, and sell their data if and when needed.

7. Conclusion

We have presented the Soarchain Mobility Network, which enables decentralized connectivity and data ecosystem for cars, pedestrians, and infrastructure. It facilitates the cryptographically-secure and resource-optimized verification of nodes within the network, ensuring that data is accurately and consistently transmitted and that the network remains honest and secure. By utilizing secure hardware, Delegated Proof-of-stake, Byzantine Fault Tolerance, and smart contracts, Soarchain offers a scalable solution for creating decentralized mobility networks that depend on mobile and heterogeneous sets of participants. Soarchain will revolutionize the way vehicles collect, process, and utilize data, and will pave the way for the widespread adoption of autonomous vehicles.

References

[1] Hedges Company, "How Many Cars Are There in the World?," https://hedgescompany.com/blog/2021/06/how-many-cars-are-there-in-the-world/, June 2021.

[2] Multicoin Capital, "Proof of Physical Work," https://multicoin.capital/2022/04/05/proof-of-physical-work/, April 2022.

[3] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," https://bitcoin.org/bitcoin.pdf, 2008.

[4] G. Wood, "Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform," https://ethereum.org/en/whitepaper/, 2014.

[5] E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized Anonymous Payments from Bitcoin," http://zerocash-project.org/paper, 2014.

[6] Cosmos Network, "Cosmos Whitepaper," https://v1.cosmos.network/resources/whitepaper, 2016.

[7] M. Zajko, "Token Incentivized Physical Infrastructure Networks," https://medium.com/@mikezajko_16091/token-incentivized-physical-infrastructure-networks-3548b3182d82, September 2021.

[8] Tendermint, "Tendermint Spec: Core Data Structures," https://github.com/tendermint/tendermint/blob/master/spec/core/data_structures.md, 2018.

[9] ETSI, "EN 302 637-2 - V1.4.1 - Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service," https://www.etsi.org/deliver/etsi_en/302600_302699/30263702/01.04.01_60/en_30263702v010401p.pdf, December 2020.

[10] ETSI, "EN 302 637-3 - V1.3.1 - Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service," https://www.etsi.org/deliver/etsi_en/302600_302699/30263703/01.03.01_60/en_30263703v010301p.pdf, December 2020.

[11] ETSI, "TR 103 562 - V2.1.1 - Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Analysis of the Collective Perception Service (CPS); Release 2," https://www.etsi.org/deliver/etsi_tr/103500_103599/103562/02.01.01_60/tr_103562v020101p.pdf, January 2021.

[12] Soar Robotics, “Soarchain Core: PoAv Transaction Protobuf,”

https://github.com/soar-robotics/soarchain-core/blob/dev/proto/poa/tx.proto, 2022.

[13] Soar Robotics, “Soarchain Docs,”

https://docs.soarchain.com, 2022.

[14] E. Buchman, J. Kwon, and Z. Milosevic, "The latest gossip on BFT consensus," Tendermint, https://arxiv.org/pdf/1807.04938.pdf, September 24, 2018.

[15] ETSI, "TR 103 562 - V2.1.1 - Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Analysis of the Collective Perception Service (CPS); Release 2," https://www.etsi.org/deliver/etsi_tr/103500_103599/103562/02.01.01_60/tr_103562v020101p.pdf, December 2019.