Latitude: A Blockchain for the Transportation Industry[☆]

Latitude Labs

Silicon Valley, California

Summary

Blockchains are rapidly becoming the new vehicles for decentralized applications for the data sharing economy. By their design, they are able to create privacy aware, secure, trusted, verifiable and user incentivized ways of sharing data. This ability has created a new wave of applications in various industries which are disrupting the status quo of the incumbents. Also, the Transportation industry is seeing a shift in terms of the amount of data that has become available due to mobile phones, dedicated sensors and crowd-sourced availability of data. The industry has a large number of applications, users and regulations which can benefit from data sharing if it can be done correctly.

We present the Latitude blockchain which is designed to be the best blockchain for transportation applications. Latitude is designed using principles derived from the core decentralized blockchain technologies available today combined with fundamental considerations for the unique aspects of Transportation data and applications. Latitude includes the world's first smart contract system specifically tailored for geo-spatial, mapping, location and sensor data and related applications. The system is powered by a geo-spatial decentralized datastore which can scale to planet level storage capabilities incentivized by a token economics built over the LAT (Latitude Token).

The Latitude network is also provisioned with a library of algorithms, both open source and closed form, which allow Government, insurance companies and regulators to compute, certify and create cryptographic proofs which can help enforce policies or verify contracts. Latitude has a decentralized Governance model using a Council system which is resistant to collusion and Byzantine behavior and is incentivized to maintain honest and proper operation of the network.

Latitude can support four classes of applications. First, Ridesharing applications consist of the sharing economy applications for multi-modal ridesharing and transport. Using Latitude, it becomes possible to create better incentives for ride providers and users including incentives for data sharing and route optimizations. This also allows regulatory bodies to participate in the execution of these applications using smart contracts. Second, Telematics applications such as Usage-based Insurance (UBIs) and fleet management can benefit from real-time or historical data and corresponding intelligent algorithms, thus providing shared mutual benefits. The third class of application include mapping and location. Using tangible user incentives while providing strong trust, privacy and security for user data, it is possible to create new or better data sharing applications (similar to Waze among others) on the Latitude platform. Finally, all the geo-spatial, location, mapping and real-time city level data can be used by the City and State governments for smart city applications to provide a better quality of life for their citizens.

Keywords: location, cryptography, decentralization, blockchain, ethereum, database, sql, access control, p2p, tokens, incentives, mechanism design, byzantine faults, data sharing, latitude, longitude, driverscore, mapping, transportation, insurance

[☆]Full whitepaper Version 2.0

[*]Latitude Labs at https://latitude0x.com

06.2018

Contents

1. Introduction

There is a fundamental shift happening in the transportation industry due to the availability of data and resulting applications in ways that was not possible before. For instance, due to the proliferation of mobile phones and real-time location data, it became possible to disrupt the traditional taxi-based transportation market using Ride-sharing applications. Similar trend in real-time traffic and incident data collected in a crowd-sourced manner has led to apps like Waze which have improved our daily commute life. Using data-sharing it also became possible to build apps that can do multi-modal ride computations (bike, scooters, etc). Smart cities are able to use transportation data for urban planning, zoning and development use cases. Insurance companies are able to provide per-mile insurance using driver behavior determined by data from sensors.

Benefits of Data sharing:

Users of these applications are providing a staggering amount of data. However, due to centralization this data is subject to policies, security and privacy practices that are not in the user's control. Moreover, its not possible for regulators or city governments to use this data for benefit of their citizens. Centralization creates data silos which dramatically reduce the utility of the technology. The recent breaches in data usage, security and privacy have shown that users want control over who is using their data and how. This includes how advertisers are using their data, for example. Also, regulators want to make sure policies are implemented correctly but have no programmatic way of doing so.

Now, imagine if these hurdles can be taken care of using the right technology, would data sharing enable new and disruptive applications? Lately, Blockchain technology has emerged as a way for creating decentralized data-based applications in a secure, privacy-aware, verifiable and trusted manner. The possibility of enabling data sharing using blockchain technology with cryptographically protected sharing methods has the potential to change the sharing economy.

We present Latitude, which is designed to be the best blockchain for the Transportation industry. Latitude allows the development of decentralized apps using smart contracts powered by a geo-spatial datastore. This is built off strong cryptographic proofs and primitives that provide security, privacy and anonymity guarantees. For the Transportation Industry, such data sharing can help solve some critical problems. For example, cities can share the data they collected about transport patterns with different stake holders such as transportation companies to reduce the traffic jams during rush hour [**?**]. First responders and aid workers can share data between them to better coordinate disaster relief efforts [**?**].

For the common user, Latitude can help build decentralized apps that can provide incentives to both drivers and users for sharing their data. This creates new incentive structures for apps such as ride sharing or per-mile insurance that were not possible before. While Latitude allows safe and incentivized sharing of data, it is also a platform for building decentralized applications for the Transportation Industry. For example, a city can build applications on Latitude to understand traffic patterns or regulate driver licenses. Centralized Telematics exchanges such a Verisk or Octo can be fully decentralized and open to everyone. Their operations can be replaced by intelligent and cryptographically strong smart contracts.

Latitude will provide a state-of-the-art environment for developers including the availability of a production geo-spatial datastore, a library of smart contracts and a community to foster new application development. Latitude shall include mobile SDKs to help build mobile apps that can directly tie into the platform. We believe that Latitude has the potential to disrupt existing centralized applications in the Transportation Industry and create new ones that were simply not possible before.

The Lattiude system consists of a base blockchain which uses Delegated Proof of Stake (dPOS)

as its consensus mechanism. This base blockchain supports various application vertical where each vertical is a side-chain. Only Merkle-proofs of aggrgated transactions from the side-chains are propagated onto the base blockchain layer. This architecture allows each side-chain to define its own rules on how the transactions are committed including their own consensus mechanisms which migth work better for the application vertical.

Related projects:

There are a host of projects that are leveraging the blockchain technology to disrupt the sharing economy such as renting homes or cars (AirBnb on the blockchain), etc. Projects such as the Origin Protocol and Uchain among others are building platforms that empower developers and businesses to create their own decentralized marketplaces for sharing data in accordance with strict controls on the blockchain. Such Blockchain based approaches make it quick and easy for organizations to develop and manage listings for assets and services. Similar data sharing can yield benefits in the Internet of Things (IoT) landscape. Projects such as the IoTex blockchain are building solutions to collect, store and share data from IOT sensors, protected with a smart contract system. Platin [**?**] is a project that focuses on proof of location on the blockchain and can be used for applications involving location sharing in a trusted and secure manner. Thus in many industry verticals it is now becoming apparent that a well-designed blockchain-based platform can be disruptive to centralized incumbent players. Such a platform can also foster the creation new applications that were simply not possible before. This is a win-win scenario for users, governments and regulators alike.

Till date, there has been a void for a decentralized planet-wide system for all transportation applications. Latitude is designed to fill that void by providing core blockchain constructs tailored for transportation applications including specific SDKs and software components for each specific industry vertical (such as Ride-sharing, Usage-based insurance, etc).

The rest of this whitepaper is organized as follows. In Section **??**, we present the design of the Latitude blockchain, including the core building blocks that make it work well for applications in the transportation industry. In Section **??**, we discuss details of four different classes of applications that can be built on Latitude and how third-party companies can build decentralized applications on the platform. Each of these classes correspond to different markets or industries that Latitude can support within the Transportation umbrella. We shall also discuss the specific modules that Latitude will have to fully support each of these classes of applications. In the Appendix **??**, we discuss why Blockchains are the right technology to disrupt the Transportation industry.

2. The Latitude Blockchain

In this Section, we present a high-level design of the Latitude blockchain. The full-version of this whitepaper shall contain a very detailed design of each component. We start with an overview of the key design principles that will guide the rest of the design for the Latitude Blockchain. Its important to note here that for implementation purposes, it might be possible to use an existing core blockchain for the underlying functionalities and build Latitude as a layer on top. We shall make these determinations in the full-version of the whitepaper.

The purpose of the Latitude blockchain is to become the best platform in the world for decentralized applications for the Transportation Industry. Specifically, this boils down to constructs in the blockchain that can handle spatial, mapping, traffic, driving data including data relating to other modalities of transport (bike sharing, walking, even air routes later on). Figure **??** presents the architecture of the Latitude blockchain in terms of the core technological innovations that will be built into it.

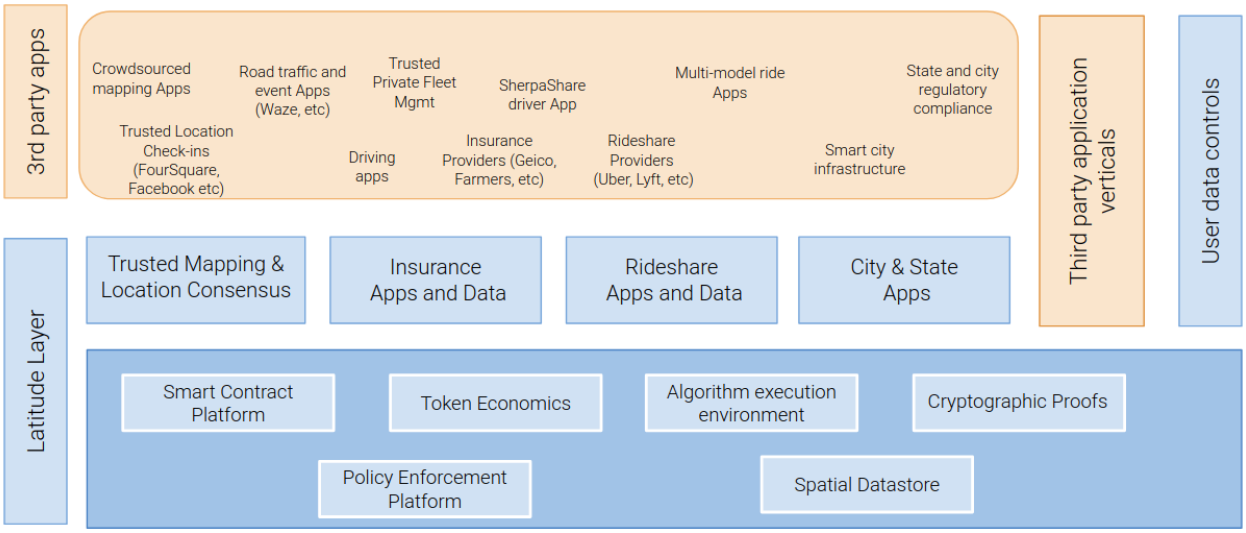

Figure 1: Architecture of the Latitude Blockchain and associated platform ecosystem.

2.1. Architecture overview

The overall architecture of the Latitude Blockchain consists of a base blockchain and a side-chain for each application vertical. This choice of an architecture gives the maximum flexibility for each side-chain to use its own mechanisms while allowing for summaries or Merklized proofs of transactions to be commited to the base chain (or the main chain).

- Overall architecture.
- Side chains. Main chain.
- chains interact mechanism.
- Cross-chain transactions are commitment protocol.

Each application vertical utilizes atleast one side-chain. Depending on the application, it might be necessary for multiple side-chains to exist within a vertical, in which case we shall use composable side-chains similar to Plasma. This allows for a hierarchical side-chain structure if needed with eventual commitment to the mainchain.

Implementation Note: For the alpha version of Latitude, it might be possible to use an existing chain with high transaction velocity and low cost, such as EOS, Neo, Zilliqa or Harmony as a drop-in replacement for the base chain functionality. In the long term, it might be beneficial to have our own base-chain with built in smart contract functions for transportation applications. This will also work better with Latitude's own token economics.

2.2. Base Blockchain layer

Blocks, proof of stake, transaction velocity.

Delegated proof of stake system where a set of "delegates" are used. The delegates are "promoted" based on time-based trust or stake (or a combination of these). This design is similar to the Cosmos blockchain and differs from other dPOS system such as EOS.

The Delegated Proof of Stake (dPOS) takes the best of both cooperative and competitive consensus algorithms. DPoS uses votes from stakeholders to achieve consensus. The competitive part

is larger stakeholders having an influence on their delegate of choice. In Latitude, it is also possible to gain a larger stake by accruing what we call "trust" through honest operation over a period of time. The delegates that have the most votes will take their turn to produce a block cooperatively in a sequence. DPoS is also scalable because anyone can participate in the consensus. Additionally, DPoS is environmentally friendly because electricity isnâĂŹt wasted like in Proof of Work.

Implementation using Tendermint or plasma chains. Only Merkle proof is committed. There can be any number of side-chains. But side-chains would correspond to one of the application platforms described later.

PBFT for consensus on high volume side-chain commitments.

2.3. Spatial datastore

One of the central aspects of the Latitude blockchain is a geo-spatial datastore that fundamentally understands various datatypes that are specific to transportation data. This datastore can use existing GIS databases that allow de-centralized storage and access. The types of data include (i) geographic data such as location (latitude, longitude), (ii) mapping data such as roads, terrain, addresses, etc, (iii) sensor data such as driving data, driver score, miles driven, route information, etc, (iv) multi-modal transport data such as biking, walking and other means of transport. Each of these data types have very special characteristics which the underlying datastore can be optimized for and allow for programming using what we call the *Latitude Smart Contract* framework.

The datastore would include spatial, quad-tree or an R-tree based indexes for efficient querying and other operations that most Geographic Information Systems (GIS) would support in a centralized manner today. It would also include functions to compute heatmaps, driving maps and statistics such as Traffic predictions including real-time analytics. Depending on how Latitude evolves, the datastore can include additional functionalities to support the data sharing among autonomous vehicles since they use most of the similar datatypes mentioned above. The datastore would support circular, rectangular and other range queries, K-nearest neighbor searches, route optimization algorithms, etc. Figure **??** shows some of the queries that such a datastore can support.

2.4. Cryptography layer

Latitude makes use of state-of-the-art cryptographic protocols to provide various proofs, access control, confidentiality and other properties that are important in a decentralized system.

such as AES encryption, secure hash functions, PKI certificates, multi-party key distribution protocols, proxy key re-encryption schemes, Elliptic-curve based Digital Signatures [**?**]. They help provide strong security, privacy, access control, confidentiality and anonymity guarantees. Anonymity guarantees are an important part of data-sharing smart contracts and privacy policies such as GDPR [**?**] especially for geo-spatial data such as location and maps. Latitude provides anonymity guarantees using cryptographic set-preserving computations as derived from research in [**?**]. These can be suitably modified to allow for location-based anonymity which require stronger guarantees when compared to set-based anonymity methods [**?** **?**].

Latitude also uses Merkle trees for cryptographic proof of audit, existence of data and verifiable computations [**?**]. These proofs can be shared as certificates among participants or be used in the Latitude smart contract system discussed later. They allow for verification of data existence or data-sharing contracts. They also allow for the maintenance of a cryptographic log of all operations that happen on the network. These techniques are similar to the ones used by some of the other blockchains today [**?**].

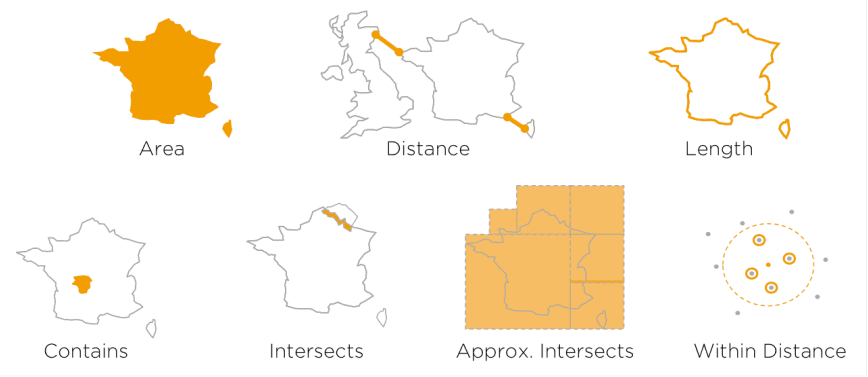

Figure 2: Examples of Geo-spatial queries that a spatial data-structure can support on the Latitude blockchain.

2.4.1. Integrity and Access Control

Latitude uses standard and well understood cryptographic protocols to provide robust access control and maintain basic integrity of transactions. Data integrity is maintained using digital signatures. Every transaction is signed by one or more of the participants certificates. The blockchain ledgers are protected using a Merkle hash [**?**] as discussed in the previous section. A Merkle proof is used as a proof of transaction for cross-chain communication.

Access control for data that is not public or open, can be managed using multi-party key communication protocol (MPC) [**?** **?** **?**]. MPC protocols work over a set of N trusted participants or a consortium set. They can be designed to allow a minimum of $m < N$ participants to reach a consensus (using a off-chain protocol) in order to compute the key that would grant access. Using these primitives, it becomes possible to have granular access control, such as different amount of consensus for read, for writes and other semantic actions. Latitude shall make these mechanisms available to the app developer through platform APIs. As always, since this is an open system, it is possible for developers to build their own access control methods if they so wish.

In addition to these standard primitives, Latitude provides a host of other proofs that are tailor made for geo-spatial, mapping, location and sensor data. These proofs can be used by applications, users and other participants in the network. The network can be extended to create new forms of proofs as the application needs grow. Below, we present the core set of cryptographic proofs that are unique to the Latitude blockchain:

2.4.2. Proof of Real-world Observation

A real-world observation could be a location computation, a ride from point A to B, a traffic observation or a landmark at a given location. The core concept behind these observations is the flow of trust. The concepts presented in this Section and the subsequent Sections replace and solve the offline Oracle problem that blockchains face today. These mechanisms provide a practical method of creating a trust source of observation without requiring a perfect off-chain Oracle [**?**].

Any observation in the real world, be it a location or a landmark sighting is only as good as the trust placed in the method and apparatus used to compute it. For example, if a GPS receiver returns a location, the amount of trust in that location result is proportional to the amount of trust in the receiver construction and the satellite system being used (such as Navstar, GNSS or others). Latitude uses this concept of trust as an internal metric to compute a proof of a real-world observation. Here we present the common algorithm used to compute these proofs which can be shared across the Latitude platform.

Definitions:

- Levels of trust $t_i, i \in \{0..M\}$, where M is the max level of trust in the system. Trust level $t_{i+1} > t_i$.
- t_0 is the base level of trust assigned to any third party untrusted source of data.
- Each data source (such an an app installation, or a third-party source) is indentified using a certificate's public key. If this is a fully untrusted source that belongs to a third party, they start at the lowest level of trust t_0.
- t_1 is the trust assigned to a second party integration with Latitude's SDKs where the SDKs directly compute the observation and report it to the system using second-party APIs. Examples include Apps in the trusted App stores that integrate with Latitude.
- t_2 is the trust assigned to a first-pary integration, such as a Latitude mobile app, or a first-party app from trusted partners such as SherpaShare.
- $tmin_{pf}$ is the minimum trust needed for a specific proof or obervation. Each proof type might have a different requirment for this parameter. The computed proof also carries this parameters as an indication of consensus or trust.
- Trust map, $Tm(e)$ gives the amount of trust recorded in the ledger for the entity e. The entity could be an organization, an individual or a user (as identified by an app installation, for example).
- N_{min} is the minimum number of entities that need to participate in the creation of this proof. This can be a function of the proof being created.
- The entity requesting the proof, submits a request $R(e, t_b, V, Tm(e))$, where R represents the proof request. The request includes credentials for entity e, the base trust level in the request t_b, the evidence of real-world observation (such as radio signal strength) V and the existing entry in the trust legder $Tm(e)$.
- Concurrence Weight: Each witness that concurs with an observation provides a *concurrence weight* which is the probability that they think the event happened. This is a value between 0 and 1. Denoted as $W_c(w, e, V)$, where the parameters are witness or observer w, entity e and evidence V.
- Location specific trust normalization factor γ_L. γ for a location L is a location specific normalization factor which represents the average trust consensus in a specific location. For example, certain areas might have a higher concentration or honest or dishonest nodes. Or certain methods of location determination might have a bias that needs to be factored in.

Each proof is implemented as a special smart contract supported by the Latitude platform. The smart contract that computed these proofs would provide a signed blob of data that certifies a certain observation as determined by the respective proof.

Algorithm for Proof of an observation X:

1. Suppose e is the entity that initiates a request for proof of an obervation X made by e.
2. The proof system finds a subset (possibly randomly sampled) of participants S who are able to *concur* with the obervation X. Each participant, $p_i \in S$, assigns a *concurrence* weight $W(p_i, X) \in [0, 1]$ depending on how well they concur with the observation.

3. Normalization factor $Nf(p_i, X)$: This is a multiplier, usually greater than 1, that signifies the amplification in trust as a funciton of how the observation was concurred upon.
4. Each observer $p_i \in S$, provides a normalization factor $Nf(p_i, X)$ and a concurrence weight $W(p_i, X)$ to the proof.
5. The proof computes the total trust in the observation X as $T_{pf}(X, e, S)$, given by Equation **??**.
6. Trust normalization: The total trust in $T_{pf}(X, e, S)$ gets normalized by γ_L where L is a coarse grained (city-level) location being considered. At bootstrap, a normalization factor of 1.0 can be used.

Trust computation for a proof of observation X:

$$T_{pf}(X, e, S) = \sum_{i \in S} T(p_i) * Nf(p_i, X) * W(p_i, X) T_{pf}(X, e, S) = \gamma_L * T_{pf}(X, e, S)$$

where $T_{pf}(X, e, S)$ is the trust for the proof of observation X proposed by entity e and observed by participants S. A proof is considered valid if $T_{pf}(X, e, S) > tmin_{pf}(X)$, that is, the accumulated trust is above the minimum required for the type of observation X.
Trust updates: Once a proof gets computed, the entity that initiated the observation gets its trust updated using an EWMA formula. Assuming entity e gets its trust updated for a proof p_e:

$$T(e)_{p_e} = (1 - \alpha) * T(e) + \alpha * T(p_e)/|S|$$

The goal of the above formula is multi-fold:

1. To increase the average trust in an entity e as a function of successful proofs. The larger the number of successful proofs, higher is the average trust in the entity.
2. Similarly, the goal is also to reduce trust in case, with adequate participants, the system was unable to verify the claim. In fact, a larger draining of trust shall be instrumented if the system finds the claim to be demonstrably false.
3. Malicious behavior: If an observer or the entity consistently disagrees with others in the proof system, over time their trust level will get degraded using a gradient method used in other reputation systems [**?**].
4. Rewarding honest behavior: Over time as observers and entities produce results with consistency, their historicla reputation gets better and recorded in the trust ledger.

2.4.3. Proof of Location:

This is perhaps the most easily motivated functionality that the Latitude blockchain can provide. Proof of location is a proof of real-world observation that proves that a given user, entity or participant is/was physically present at a given location at a specific time. The location could also be relative to another participant or landmark.

Latitude shall provide the mobile, browser and sensor SDKs that can directly tie into the datastore to provide consensus based proofs. These proofs can unlock applications such as access to facilities or help increase trust in crowd-sourced mapping, traffic and incident reports.

The proof of location uses the above framework for a real-world observations with the following specification:

- Entity e computes a location on a mobile device. This could be an Android/iOS phone or a tablet/laptop. Depending on how the entity uses Latitude's SDKs, the location computation starts with a base trust level of t_0, t_1 or t_2.

- As a part of the location proof request $R(e, t_b, Tm(e), V)$, the entity can submit any evidence V that would help prove the location. For example, on Android, this can include wifi-radio signals, cell tower signals, GPS satellite location and timing signals, Bluetooth-LE scans, and other signals that can help prove the location.
- Either the entity or the platform can find other participants for concurrence.
- Concurrence: A participant can concur by confirming the observations included in the evidence V. This can include, for example, a degree of concurrence or other factors that quantify how well they concur. The degree of concurrence can be a function of the radio signal properties [**?** **?**]. This is used to compute the concurrence metric $W_c(w, e, v)$.
- Depending on the trust level of the observer, or for other considerations, the normalization factor can be used to boost or marginalize the contribution from a witness or observer w, denoted by $Nf(w, e, V)$.
- The witnesses or observers independently submit their recordings to the smart contract system, which either generates a valid proof or declines the request. At the end of each such operation, the new trust levels for various entities are computed.
- Historical Location: A past location proof from the same entity can be used as a "virtual observation" if the location computation is in the recent past. This combined with reasonable laws of Physics or motion can provide a small amount of trust for the current proof. For instance, if the entity was spotted at a grocery store about 5 minutes in the past, then any new location update that is within a few miles of the grocery store could be trusted.
- Using the proof equation **??**, a trust level and proof is computed.

The location proof and the corresponding data including trust parameters are recorded in a web-friendly format (such as JSON/XML) on a ledger specifically used for such proofs. The ledger entry can be used as a URL to refer to this proof for other applications that can be built on top of it.

Other proof of location chains: The framework provided by Latitude is general enough to capture different implementations. For instance, [**?**] uses a trusted set of radio beacons or "anchors" to provide location signals. These essentially become highly trusted observers or participants in our framework as they are essentially first-party observers (with a large amount of base-level trust). As another example, Platin is a blockchain designed specifically for a proof of location. Their use of sensors and increase in trust over time falls in line with the general concept of proof of obervation (possibly with different trust constants).

2.4.4. Proof of Ride

Provides a proof that a particular user has taken a ride from point A to point B using a certain type of transport. This proof can be used across multi-modal ride platforms such as bike or car rides. This can be extended to include bus trips, flights, train rides and so on. The proof of ride credential will be available via the Latitude mobile SDK on various platforms. Using the mobile SDK to construct these proofs also adds to the amount of trust on the nature and parameters of the ride.

The proof of ride is computed using a time series of location updates. A Hidden Markov Model or other methods [**?**] can be used as "algorithmic observers or verifiers" of a location trace. The location trace can be supplanted with sensor data collected using a direct integration of the Latitude