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 [1]. First responders and aid workers can share data between them to better coordinate disaster relief efforts [2].

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 [3] 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 2, 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 3, 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 A, 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 1 presents the architecture of the Latitude blockchain in terms of the core technological innovations that will be built into it.

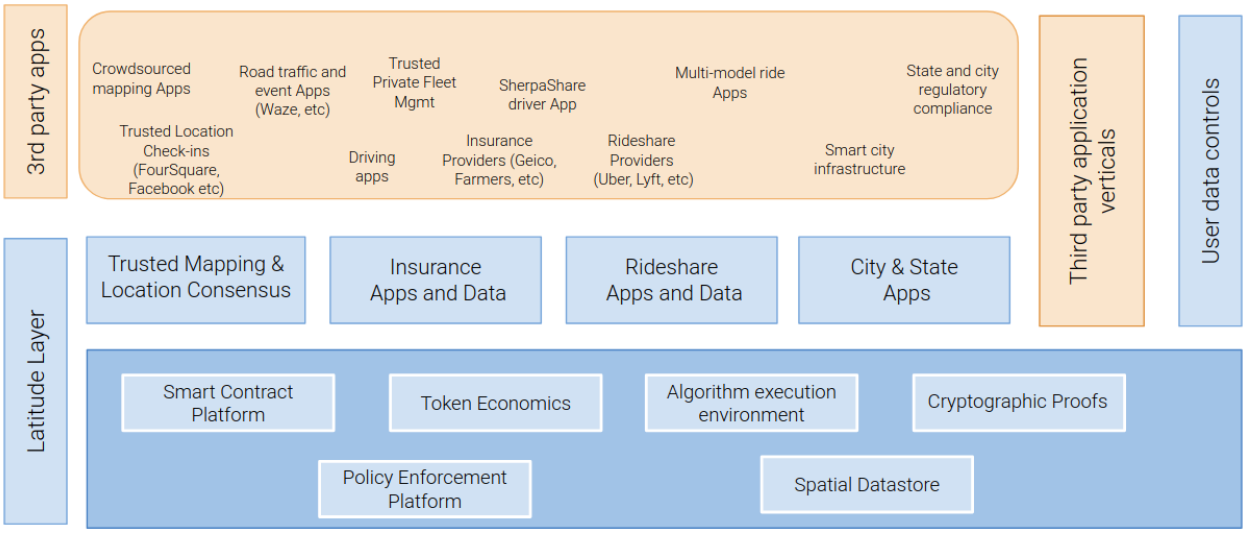

Figure 1: Architecture of the Latitude Blockchain and associated platform ecosystem.

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 2 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 [4]. 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 [5] especially for geo-spatial data such as location and maps. Latitude provides anonymity guarantees using cryptographic set-preserving computations as derived from research in [6]. These can be suitably modified to allow for location-based anonymity which require stronger guarantees when compared to set-based anonymity methods [7, 8].

Latitude also uses Merkle trees for cryptographic proof of audit, existence of data and verifiable computations [9]. 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 [10].

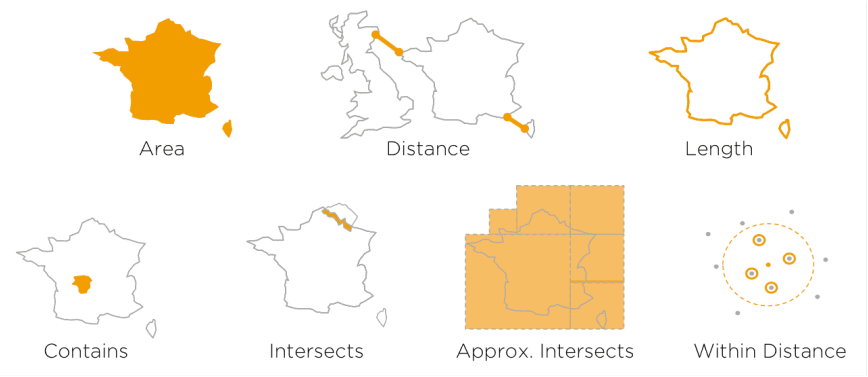

Figure 2: Examples of Geo-spatial queries that a spatial data-structure can support on the Latitude blockchain.

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 [9] 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) [11, 12, 13]. 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 [14].

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.

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 2.4.2.

6.

Trust computation for a proof of observation X:

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

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.

Other proof of location chains: The framework provided by Latitude is general enough to capture different implementations. For instance, [15] 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

2.4.5. Proof of Landmark

2.4.6. Proof of DriverScore

2.4.7. Proof of Traffic

2.4.8. Proof of Route

- Proof of location: This is perhaps the most easily motivated functionality that the Latitude blockchain can provide. Proof of location is a cryptographic credential 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. The proof of location has been explored by other blockchains such as Platin [3]. 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.
- 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.
- Proof of landmark (mapping): This proves the existence of a particular landmark such as a monument, a building, a sign post at a given lat/lng. This can also be used to prove the existence of a particular road or the lack of. These proofs can be constructed using the proof

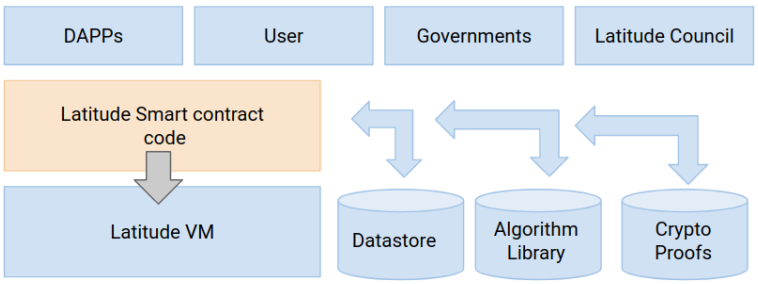

Figure 3: Architecture of the Latitude Smart Contract framework.

of location combined with consensus among users. This proof can become the backbone for verified mapping and landmark data-based applications.

- Proof of driver score: This proof can be computed on the blockchain over the data aggregated on the datastore. The algorithm itself shall be made available to the nodes either as an executable docker image or an open-source version. This allows different nodes to run the computation and create a certificate of driver score which can then be attributed to the driver. This open framework can also allow different driver score algorithms to co-exist in the system creating a community where better driver score algorithms can be agreed upon and used as the industry standard.

- Proof of traffic: Similar to the driver score, traffic is also an algorithm that looks at the statistics of location and speed data on roads. The proof is similarly computed through consensus and recorded as a certificate on the blockchain. The proof could be about real-time traffic or historic traffic patterns which can be shared with smart city applications, the Census Bureau or other regulatory bodies.

- Proof of routes: Similar to the functionality in the popular Waze app, this is about whether there exists a certain route (or a better route) from point A to point B. More the consensus, higher the trust. For example, if a user actually takes the route from A to B and provides a proof of ride, that helps create the proof of its existence. This is useful for trusted and verifiable mapping / routing applications.

2.5. Latitude Smart Contract system

Smart contracts are self-executing nuggets of code which specify the terms of the agreement between various participants on the blockchain. For the Latitude blockchain, the participants can be a user, a regulatory body, an application written by an insurance company or the city. The contract is directly written into lines of code in a certain smart contract language. Popular smart contract frameworks today include ones used by Ethereum which runs on the Ethereum Virtual Machine (EVM), the Neo which runs on the NeoVM and the EOS blockchain which runs on the WASM (WebAssembly VM).

The smart contract code and the agreements contained therein exist across the distributed, decentralized Latitude blockchain network. Because of this, Smart contracts permit trusted transactions and agreements to be carried out among disparate, anonymous parties without the need for a central authority, legal system, or external enforcement mechanism. They render transactions traceable, transparent, and irreversible as the state is always available on the Latitude blockchain.

Figure 3 shows the high-level architecture of the Latitude Smart contract system. Latitude uses a WebAssembly (wasm or eWasm) based compiler for a smart contract written in Go or C++.

– Overview of the webassembly architecture.

WebAssembly (or wasm) is a binary instruction format for a stack-based virtual machine. It includes a complier that is portable and has a compilation target for high-level languages such as C++ and Go. The Latitude smart contract system shall support both C++ and Go languages. The choice of wasm was to make the execution of the smart contract efficient and thus have high throughput. Also, popular networks such as Ethereum are planning to move to eWasm as part of the Ethereum Virtuam Machine (EVM) 2.0. This which will help create a common developer pool (and other aspects such as tooling, educational material, etc) in the community. Wasm's execution happens in a safe and sandboxed environment which can be beneficial in an adversarial and open network. It might also be possible to secure the Wasm execution using Trusted Execution Environments (TEEs) in the near future.

The reason for chosing WebAssembly over the popular Solidity (or Vyper) used by the Ethereum Virtual Machine (EVM 1.0) is twofold: WebAssembly compiles into native code and thus executes faster than Solidity or Javascript. Also, EVM 2.0 will feature a modified version of WebAssembly, namely *eWasm* thus reducing the cognitive burden on developers across both ecosystems.

Latitude uses a modified version of Solidity (or the new Vyper programming language) as used by the Ethereum Virtual Machine. The choice of this language is based on the production quality of the EVM, the language tools available in the community for creating contract code, developer support and talent pool. Latitude shall add enhancements to the language to support new spatial data types, indexes, cryptographic proofs and other mechanisms that are fundamental to the platform. Some of the goals of the smart contract system include the ability to convert data policies such as GDPR [5] into Latitude smart contract code which can then get automatically verified and enforced on the blockchain. Also its possible to share location data in an ephemeral manner for a specific purpose – the data item gets automatically destroyed using consensus and smart contract constructs on the blockchain. An example includes a user sharing their location with an app for a very small duration of time.

As shown in Figure 3, the smart contract code has access to the datastore, cryptographic proofs (including algorithms to create proofs), an algorithm library (which hosts algorithms such as traffic, driver score etc). These are directly accessible through language constructs making it easy to write high quality smart contract code. The smart contract execution framework and related functionalities are accessible to dapps, entities such as users, governments and the Latitude Council (discussed later in the Governance Section) on the system.

The Latitude smart contract system can becomes the world's first such smart contract framework specifically tailored for transportation data and applications.

- Explain action methods.
- Data structures.
- Callback methods.
- Events.

2.5.1. Secret contracts

Typical smart contracts are public, including the data they operate on. For privacy or other reasons, it might be desirable to achieve consensus on data without making it open on the chain. Latitude supports what are called secret contracts using multi-party key distribution protocols. Multi-party key distribution creates a set of keys, such that a function (or a smart contract) can be computed in

Listing 1: Structure of a Latitude Smart Contract

```
#include <latitude/contract.hpp>

// Shorthand for a location update that gets deleted in one hour.
typedef latitude::Location<TimeUnit<OneHour>> OneHourLocation;

class sample_contract : public latitude::Contract {

    private:
      // private data structures to the contract. The data does not go onchain.

    public:
      // data structures that can go onchain.
      // The contract methods will write to these.


      // There are two types of methods: Actions and callbacks.

      void onLocationUpdate(latitude::OneHourLocation curr) {
          // business logic
      }

      void computeProximity(OneHourLocation given, Location<NoExpiry> target) {
          // business logic
      }
}
```

a distributed manner over a random piece of data without any single entity having full access to the data. The result of the computation has consensus and can be put on-chain using a cyrptographic proof or can be directly communicated to a decentralized app.

The availability of Trusted Execution Environments or hardware enclave can dramatically assist in the implementation of secret contracts. In the first version of Latitude, we shall use existing MPC communication protocols in this area to create a secret contract system similar to the one used in Enigma [11]. In a later version, it might become possible to use TEEs using a design similar to Ekiden [16].

2.5.2. Contract sharding

Latitude smart contracts will be sharded to improve performance. Existing blockchains such as Ethereum, Neo and EOS suffer from slow throughput on smart contract execution. One technique that has recently emerged as a way to scale performance is to shard the contract using annonatations on methods to expose specific semantics. For example, by understanding portions of a smart contract that store data, that verify computations and that are callbacks from user-facing apps, it becomes possible to separate the execution among parallel nodes for much higher throughput [17].

For example, in a given smart contract, certain data structures are annotated to store data. The dependencies between the data structures is also specified as a group. There are three kinds of methods: Actions, callbacks and Verifiers. The Callbacks are used by the system to update the contract when a data or event happens, such as a user updates their location. The Action methods are executed by the smart contract to take an action and the verifiers are methods that verify transaction state. By isolating these methods and their data dependencies, one can shard a smart contract to execute in pieces on different nodes on a blockchain thereby increasing efficiency and reducing the probability of a coordinated attack.

Much like the other components, Latitude's smart contract sharding system shall leverage available algorithm libraries and tools for rapid and iterative development.

2.6. Token economics: The Latitude Token (LAT)

Latitude has its own token for use on the Latitude Blockchain, called LAT. There would be a fixed token supply for a certain period of time (4 years). A certain percentage of the tokens are reserved for funding and other operations, the detials of which are not discussed in this document. The rest of the tokens will be available for the network for use.

INSERT TOKEN PIE CHART

Cryptoeconomics: Cryptoeconomics refers to the mechanics of the protocol underlying the blockchain operations which creates incentives for the various stake-holders. For example, in the Latitude Blockchain it is possible to reward users for sharing their data for certain purposes. For example, users can be rewarded if they share traffic data or accident information or better routes. This is similar to the Waze model but creates legitimate rewards for users that have meaningful value outside the Waze application. For an overview of Cryptoeconomics in the blockchain space, please see [18].

The core building block of the Cryptoeconomics in Latitude is the Latitude Token, or LAT. This will be transacted in each and every micro-transaction that takes place on the blockchain to create the right incentives, enforce smart contracts and penalize Byzantine behavior. One of the fundamental design principles behind the protocol for the LAT token is to create incentives assuming nodes are greedy and are interested in maximizing their gain. We will also build safeguards against reasonable amount of collusion among nodes to subvert the system. The design is crafted such that the best way for a node to maximize its revenue would be to participate with full honesty.

Latitude shall employ a Proof of Stake model (delegated or non-delegated) for participation and core node-level mining. This mechanism has recently gained popularity among a notable number of blockchains [19]. This also allows for deposit slashing as a technique to tackle Byzantine behavior. Latitude will employ techniques such as Minimal Slashing [20] for Byzantine fault tolerance and safety under distributed asynchronous operation.

User incentives: Token economics allow for the creation of what we call user incentives. These are protocol constructs in the blockchain that allow users to benefit from the value they create for the ecosystem. Refer to [21] for an overview on incentive mechanisms to reward users for various methods of participation in the network. In general, the Latitude token ecosystem will be based on market economics, that is, supply and demand from various participants will be the primary driver for prices and incentives in the network. This philosophy falls in line with decentralized control and operation while also allowing for creating most reward for honest behavior in the network.

The ability of user incentives to exist in a decentralized manner can be disruptive to existing incumbents in the sharing economy space, such as Uber, Lyft, Airbnb, etc since users can get rewarded in a tangible manner for their contributions [22]. Consumers shifted to apps in the sharing economy as they provided cheaper and better alternatives to traditional services like Uber and Airbnb. However, since all transactions go through these centralized providers, the platform owners determine the fees, percentages and are in complete control of any data practices and policies which cannot be verified. They often become accused of predatory behavior. Using blockchain based incentives, sharing and open-source software these problems can be addressed in a singular fashion.

As an example, consider the Ridesharing application. As users contribute data on what rides they are taking, it becomes possible for the network to reward them with tokens. They could, upon sufficient contribution, redeem them for free rides or share with them others on the network. A similar model can be adopted for data concerning driver behavior where drivers using different apps and sensor algorithms can elect to share their data towards building a better driver score in return for suitable incentives.

2.7. Governance on Latitude

Governance refers to a decentralized manner in which decisions are made using a consensus mechanism on the blockchain. Decisions include basic constructs whether a node can join or leave the network. Or it can include key decisions on whether an upgrade should be mandated on every node, a given participant such as a data provider should be penalized. It could also include issues where humans get involved, such as when a user complains of a loss of privacy or a breach in contract.

Recently blockchains have been moving towards governance using a small set of participants, such as trusted miners in the case of Stellar and Ripple [23]. The EOS blockchain uses a similar concept of a core set of block producers who are elected based on a nomination and voting process [24]. For discussion around Governance in Ethereum, refer to [25]. Latitude uses a similar concept of a *council* of participants. These are entities (nodes or organizations) that have demonstrated participating using earned trust through honest operation, accumulating stake, demonstrated good intent and establishing trust. Some members of this council might include the core Latitude developers which allows them to implement operations such as updates, bug fixes and so on. The council members shall be elected using the an election protocol on the blockchain. It might be possible to directly nominate certain council members such as regulatory bodies who have general interest in user rights, privacy and enforcement.

	Bitcoin's blockchain	Ethereum	Stellar	Ripple
Average Transaction Confirmation Time	1 hour	15 minutes	3 to 5 seconds	3 to 5 seconds
Average Transaction Fees	$0.61 per transaction	$0.02 per transaction	$0.01 will pay for 300,000 transactions	$0.01 will pay for 3 transactions
Transactions Per Second	3 transactions per second	7 transactions per second	1000 transactions per second	1000 transactions per second
Consensus Mechanism	Proof of Work	Proof of Work	Stellar Consensus Protocol (SCP)	Ripple Consensus Algorithm
Validator control	Decentalized	Decentalized	Decentalized	Centralized
Governance	Non-profit	Non-profit	Non-profit	For profit

Figure 4: Comparison of transactions per second of the major blockchains today.

2.8. Performance considerations

When thinking of performance, one of the key metrics that is hotly debated in the community is transactions per second. Bitcoin is know for its long time to produce a block, on the order of minutes which limits the number of transactions the network can process. Figure 4 shows a comparison of the major blockchains today with respect to this crucial metric. Note that these blockchains shown in the Figure are primarily evaluated against the concept of a transaction which represents a transfer of assets, goods or monetary value digitally on the blockchain. Stellar and Ripple are two blockchains that have gained popularity for financial transactions as they tout a higher transaction velocity.

In the context of Latitude, the performance of the blockchain is important. The blockchain will handle different types of data and transactions which will require different levels of consensus and trust. Figure 5 shows the different types of transactions that the Latitude blockchain can process and the performance we expect to achieve. Shown are four types of transactions:

- Datastore transactions: These refer to basic transactions to store data values into the geo-spatial data store. For instance, if a user shares their driving data, this can include the sensor information, lat/lng of the trip taken and any mapping data collected. For ride-sharing applications, this can include any multi-model ride details that the user has booked. We expect the blockchain to be able to process close to 1-10 million transactions per second, since most such transactions require very low level of consensus and can tolerate eventual consistency [26].

- Algorithmic computations: These refer to transactions which include executing known algorithms on the blockchain. For instance, this could include the computation of various driver behavior algorithms with the computation being shared among certain participants on the network. Another computation could include statistics such as real-time traffic or aggregate traffic statistics shared with a city for better zoning and planning purposes. The amount of consensus required is relatively small but higher than datastore operations in order to ensure correct execution of algorithms and lack of malicious intent. Also such operations would require strong consistency for their CRUD functionalities. We expect the Latitude design to support 10-100K transactions per second at its peak usage.

	Amount of consensus	Transactions per sec (TPS)
Datastore transactions *(real-time data read/write operations)*	low	1 - 10 Million
Algorithmic computations *(driver score computations, traffic statistics, ridesharing computations)*	medium	10-100K
Data sharing contracts *(long term contract executions, enforcement operations)*	high	100 - 1000
Governance operations *(node dynamics, Byzantine behavior penalties, council voting)*	very high	1 - 10

Figure 5: Expected Transaction per second velocity of various types of functionalities on the Latitude Blockchain.

- Data sharing contracts: These transactions refer to the creation, deletion or arbitration of long-term sharing smart contracts between participating entities. For instance, it could include a new contract between an insurance company and a data provider for sharing certain types of driver score data for certain geographic locations. Since these transactions have higher value they require larger amount of trust and consensus in the system. Latitude shall support a transaction speed of around 100-1000 transactions per second for this category.
- Governance operations: The Governance operations require the highest amount of trust and full consensus of the network. These include voting to add/remove council members, critical council decisions such as forks or updates/upgrades, decisions on high-value smart contract disputes, etc. The transaction velocity is low for reasons of trust, accuracy and correctness and thus we expect the Latitude blockchain to support 1-10 transactions per second under this category.

Another metric of importance for Latitude is the storage capacity in the network. This can be important for datastore operations. We expect the storage capacity, network bandwidth and any other computing resource to become available on an incentivized model as determined by usage contracts. For instance, if a user is willing to share data with a data consumer, the consumer should be able to allocate token resources to provision the network with sufficient storage. The token can be used to purchase storage using other blockchain storage providers such as Filecoin, Siacoin or Golem can be used to incentivized nodes to directly supplant on-chain storage.

3. Application Sidechains

In this section, we discuss the four different classes of applications that can be built on the Latitude platform. Each of these applications is supported by a suite of software modules, smart contract add-ons and SDKs (mobile and desktop) that utilize the core functionality offered by Latitude and provision it in unique ways to suit the specific class of applications. Each application is a side-chain that posts its aggregated transactions to the main-chain if needed. The Latitude platform is extensible in the sense that its not limited to these four classes of application side-chains. For example, its possible to add additional applications such as for the shipping or the airline industry as

verticals on the platform. Next we discuss each of these applications and how they can be supported on the platform.

3.1. Ride-share application side-chain

Ride-sharing applications are the pinnacle of the applications in the data sharing economy. They empower the user to choose from a multi-modal ways of finding or providing rides. Latitude can provide the decentralized infrastructure to store, share and build decentralized applications for their corresponding counterparts.

The ride-sharing market alone is large enough to power the Latitude blockchain as its primary use-case. The ride-sharing market is around 17 Billion dollars with around 60 Million users. This does not include the multi-modal ride sharing segment which is growing rapidly. This segment includes bike, scooter and such modes of transport. The ARPU is around 293 dollars which is high enough to create user incentivized models for data sharing. The data generated by these users currently sits in silos in the respective ride sharing apps, which can be unlocked and put to good use through such mechanisms.

Latitude shall provide a mobile SDK and a blockchain API specifically to suit ride-sharing apps for sharing data. This would allow the creation of decentralized ride sharing apps that can provide users with incentives to share data, subsidize rides and provide a better deal for ride providers. They can also allow regulators to enforce the use of roads, lanes, parking and other structures in accordance with city or neighborhood ordinances. Latitude's SDK shall also provide a multi-modal ride API that such apps can use to request rides on the platform.

An app that integrates with the ridesharing SDK becomes a trusted data provider. Further verification of the letigmacy of the data source can be performed by integrating with Trusted Execution Environments (TEEs) available on most iOS and Android phones today.

Side-chain design: The design of the ride-share side-chain shall include token rewards for contributing data. The reward mechanism shall be provided using a vetted smart contract system. The spatial datastore shall be used for hosting the data. The side-chain shall also host applications that use the data in accordance with the data contract between the data providers and the app developers.

As an preferred third-party integration, Latitude shall integrate with SherpaShare's platform for drivers. This would allow greater sharing of the geo-spatial and driving data collected through SherpaShare's driver app. This will be one of the early launches for the beta version of the Latitude platform.

3.2. Telematics side-chain

Data is the most important asset for Telematics applications. Thus, sharing of data, computations and algorithms can be beneficial to companies, users and the ecosystem in general. In fact, by using the Latitude blockchain it becomes possible for regulators to enforce laws better while maintaining data privacy. This can result in lower crime rates and incidents.

The important of data sharing here can be seen from the existence of *Telematics data exchanges* that are centralized cloud-based exchanges which act as data brokers among users and insurance companies. As an example, consider the Verisk Data Exchange (verisk.com) which allows the exchange of driver data to insurance companies. They share data for all kinds of connected vehicles for underwriting, rating, and claims handling through Usage-based Insurance (UBI) programs. The exchange stores and processes Telematics data of all types, volumes, and velocity from connected cars, after-market hardware, or mobile solutions. A second example is the Octo Telematics company which also operates an exchange that is used by over 100 insurance companies, has about 186 billion

miles of driving data and gets 11 billion new data points from 5.4 million connected cars and sensors every day.

The problem with such centralized solutions is that there is a central entity that extracts the fees and has direct control over the data. Such centralization can result in security and privacy loss and lack of trust from all parties involved. Latitude is directly poised to disrupt this market by providing a blockchain based platform for exchange of data, computation, algorithms and resulted information. Using Latitude smart contracts it becomes possible to enforce proper security, privacy and sharing of data in accordance with agreed upon incentives. This implies cleaner and better sharing methods which can be a win-win for everyone in the industry including the regulators.

Latitude shall provide a Telematics SDK and blockchain library module to enable such applications. This would include a suite of pre-existing smart contracts, datastore provisions and other software modules necessary to build decentralized data exchanges which can eliminate the expensive data brokers. The SDK shall also include APIs for Usage-based Insurance companies to act as data providers given appropriate involvement, permission and incentives for the users. Users can also be incentivized to provide data directly to the platform by getting token rewards. Such a model also allows for open experimentation with driver behavior algorithms such as DriverScore. It also becomes possible for drivers and users to "carry" their score or reputation from one platform to another, thus preventing lock-in scenarios.

3.3. Mapping and Location side-chain

Mapping and Location-based services and analytics is another big segment that Latitude can address through decentralization. Mapping and real-time location/speed data is a huge market that apps like Waze are able to access. However, the user incentives in Waze are limited to the Waze platform and cannot be carried over. Finally, the data that users contribute also stays isolated to the Waze platform and cannot be used by the City or the National Transportation Safety Board (NTSB) for altruistic purposes. To get a sense of the market size, today Waze has about 70 million users with 500K volunteers who provide real-time data. A Waze user on average spends about 480 minutes per month on the platform.

The Latitude blockchain can change all of data isolation and platform lock-in by providing strong incentives to the user by rewarding them with crypto-currencies. The smart contract system can also be used to enforce proper sharing of data with the right participants. The built-in primitives in Latitude for Byzantine behavior can be used to ensure honest operation at all times.

Location-based services can be similarly decentralized on Latitude. Using the mobile Latitude SDK, it becomes possible for users to directly contribute mapping and location data to the platform. This allows Latitude to construct the various cryptographic proofs as discussed in Section 2.4. Proof of location technology alone has the potential to disrupt the "Check-ins" industry. The Facebook platform alone gets about 50 million check-ins per year. The other major players here include Foursquare, Yelp and Google maps. The sharing of check-ins data on the Latitude platform can enable new applications not previously possible due to the platform lock-ins.

3.4. Smart city and Govt Application side-chain

Smart cities refer to a large set of applications that can improve the life of a citizen by providing better amenities. Examples include real-time data for parking, data-driven urban planning, city-wide zoning research, environmental location sensors, disaster management, transport/transit data (real-time and historical), smart transport (shuttle/bikes), traffic light management to name a few.

Latitude blockchain can host such smart city applications and collect data from their citizens.

Users are incentivized to provide data using token economics and the regulators or city officials can "purchase" data or results using smart contracts. It might be possible to share aggregated data with careful focus on privacy/anonymity to State level or Federal government for various regulatory reasons.

4. Conclusion

Blockchains have the potential to disrupt incumbent applications on the data sharing economy. In addition to this, the Transportation industry is going through a radical shift due to the new types of data and applications that have become available. For instance the proliferation of mobile phones has brought a surge in crowd-sourced data. Also widespread deployment of cheap hardware sensor networks combined with smart analysis and planning algorithms are able to utilize this data to create valuable new insights and applications that were not previously possible.

By bringing the blockchain technology and designing it from the grounds up for Transportation applications, we have created the world's first blockchain specifically tailored for Transportation applications. Our mission is for Latitude to become the de-facto platform for all transportation applications by building the right constructs for trusted, privacy-aware, secure and verifiable data and computation sharing/enforcement.

Our mission is to build Latitude to support applications across the planet and across different modes of transport. We are excited by how Latitude stands to disrupt existing applications such as Ride-sharing, Mapping, Location sharing and analytics and the driver-behavior industry (UBIs).

The full-whitepaper shall contain a deeper dive into the mechanics of the Latitude blockchain including details on how the smart contract system, the cryptographic proofs and the datastore would function.

5. References

[1] N. Shibata, T. Terauchi, T. Kitani, K. Yasumoto, M. Ito, and T. Higashino. A method for sharing traffic jam information using inter-vehicle communication. In *2006 3rd Annual International Conference on Mobile and Ubiquitous Systems - Workshops*, July 2006.

[2] Nitesh Bharosa, JinKyu Lee, and Marijn Janssen. Challenges and obstacles in sharing and coordinating information during multi-agency disaster response: Propositions from field exercises. *Information Systems Frontiers*, Mar 2010.

[3] Platin Proof of Location (PoL) . `https://platin.io/`.

[4] The Fundamentals of an ECDSA Authentication System. `https://www.maximintegrated.com/en/app-notes/index.mvp/id/5767`.

[5] General Data Protection Regulation (GDPR). `https://gdpr-info.eu/`.

[6] Lea Kissner and Dawn Song. Privacy-preserving set operations. In *Proceedings of the 25th Annual International Conference on Advances in Cryptology*, CRYPTO'05. Springer-Verlag, 2005.

[7] Aris Gkoulalas-Divanis, Panos Kalnis, and Vassilios S. Verykios. Providing k-anonymity in location based services. *SIGKDD Explor. Newsl.*, 2010.

[8] Toby Xu and Ying Cai. Location anonymity in continuous location-based services. In *Proceedings of the 15th Annual ACM International Symposium on Advances in Geographic Information Systems*, GIS '07. ACM.

[9] Georg Becker. Merkle signature schemes, merkle trees and their cryptanalysis. *PhD Thesis*, 2008.

[10] Merkling in Ethereum. `https://blog.ethereum.org/2015/11/15/merkling-in-ethereum/`.

[11] Enigma. `http://enigma.media.mit.edu/`.

[12] NuCypher KMS: Decentralized key management system. `https://github.com/nucypher/whitepaper`.

[13] X.-F Zhao, Q.-L Xu, and P He. Survey on group key agreement protocols. 36:70–75+103, 2010.

[14] Concurrence. `https://concurrence.io/`.

[15] Jaydip Sen. A survey on reputation and trust-based systems for wireless communication networks. *CoRR*, abs/1012.2529, 2010.

[16] Arunesh Mishra, Shravan Rayanchu, Ashutosh Shukla, and Suman Banerjee. Towards secure localization using wireless âĂŸcongruityâĂŹ, 2008.

[17] Suhas Mathur, Robert Miller, Alexander Varshavsky, Wade Trappe, and Narayan Mandayam. Proximate: Proximity-based secure pairing using ambient wireless signals. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, MobiSys '11. ACM, 2011.

[18] FOAM: The Future of Proof of Location. `https://foam.space/`.

[19] Raymond Cheng, Fan Zhang, Jernej Kos, Warren He, Nicholas Hynes, Noah M. Johnson, Ari Juels, Andrew Miller, and Dawn Song. Ekiden: A platform for confidentiality-preserving, trustworthy, and performant smart contract execution. 2018.

[20] Mustafa Al-Bassam, Alberto Sonnino, Shehar Bano, Dave Hrycyszyn, and George Danezis. Chainspace: A sharded smart contracts platform. 2017.

[21] Sinclair Davidson, Primavera De Filippi, and Jason Potts. Economics of blockchain. In *Proceedings of Public Choice Conference*, 2016.

[22] DPOS Consensus Algorithm - The Missing White Paper. `https://steemit.com/dpos/@dantheman/dpos-consensus-algorithm-this-missing-white-paper`.

[23] Minimal Slashing Conditions. `https://medium.com/@VitalikButerin/minimal-slashing-conditions-20f0b500fc6c`.

[24] Token Economy 101, or why Blockchain-powered decentralized networks are important. https://blog.otiumcapital.com/token-economy-101-or-why-blockchain-powered-decentralized-networks-are-important-310de1cc8bac.

[25] Decentralizing the Sharing Economy With Blockchain Technology. `https://bitcoinmagazine.com/articles/decentralizing-sharing-economy-blockchain-technology/`.

[26] Introducing Stellar. https://www.stellar.org/blog/introducing-stellar/#gateways.

[27] EOS Block Producer Voting Guide. https://medium.com/coinmonks/eos-block-producer-voting-guide-fba3a5a6efe0.

[28] Blockchain and Smart Contract Mechanism Design Challenges. https://fc17.ifca.ai/wtsc/Vitalik%20Malta.pdf.

[29] De-mystifying eventual consistency in distributed systems. http://www.oracle.com/technetwork/products/nosqldb/documentation/consistency-explained-1659908.pdf.

[30] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf, 2009.

[31] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 1982.

[32] Kevin Liu, Harsh Desai, Lalana Kagal, and Murat Kantarcioglu. Enforceable data sharing agreements using smart contracts. *CoRR*, 2018.

A. Why use a Blockchain ?

In this Section, we go over the key reasons why a Blockchain-based platform is the right solution for Latitude. We start with an overview of the Blockchain technologies and present the pivotal features that would work well for the business use case of Latitude.

Blockchain has the potential to become the new decentralized application platform for the Internet. Blockchain is a public register in which transactions between two users belonging to the same network are stored in a secure, verifiable and permanent way. The data relating to the exchanges are saved inside cryptographic blocks, connected in a hierarchical manner to each other. This creates an endless chain of data blocks – hence the name blockchain – that allows one to trace and verify all the transactions they have ever made.

The introduction of Bitcoin [27] triggered a new wave of decentralization in computing. Bitcoin illustrated a novel set of benefits: decentralized control, where "no one" owns or controls the network; immutability, where written data is tamper-resistant ("forever"); and the ability to create & transfer assets on the network, without reliance on a central entity.

The initial excitement surrounding Bitcoin stemmed from its use as a token of value, for example as an alternative to government-issued currencies. As people learned more about the underlying blockchain technology, they extended the scope of the technology itself (e.g. smart contracts), as well as applications (e.g. intellectual property).

Bitcoin was the first such blockchain to introduce the concept of full decentralization, but Ethereum has made this a general platform for executing arbitrary applications called dapps using smart contracts and a wide range of other tools. Since Ethereum, there has been an explosion in blockchain technology to allow a wide range of distributed decentralized applications to run over a network of untrusted arbitrary nodes over the planet, which almost mimic the structure and spread of the Internet.

The primary function of a blockchain is, therefore, to certify transactions between people. In the case of Bitcoin, the blockchain serves to verify the exchange of cryptocurrency between two users, but it is only one of the many possible uses of this technological structure. In other sectors, the blockchain can certify the exchange of shares and stocks, operate as if it were a notary and "validate" a contract or make the votes cast in online voting secure and impossible to alter.

Decentralization is one of the core concepts or features of a blockchain. It can mean different things in different contexts but for our purposes it allows for two important things:

- Decentralization of control or power: That is, no single entity such as a company or an institution has unrestricted control over all aspects of the data. In a centralized world, for example, Uber has direct control over all user data and how that data is being shared with third-parties, etc. Of course, there are privacy policies that are published, but as a user one has no choice but to place full trust in the policy or their implementations. There is no method of open verification or recourse in case of a breach. With a blockchain-based solution, all participants on the network have equal say. Using primitives such as consensus and smart contracts, it becomes possible to verify and enforce such policies.

- Decentralization of software: There is no single centralized place on the Internet which hosts the functionality or the data. The blockchain itself is stored in a decentralized manner on the nodes that form the network and thus, there is no single point of failure or trust associated with the system. This is a big advantage that blockchain systems have over centralized solutions.

As a result of this decentralization, the blockchains get some nice benefits as given below:

- Fault tolerance - decentralized systems are less likely to fail accidentally because they rely on many separate components that have uncorrelated failure models.
- Attack resistance - decentralized systems are more expensive to attack and destroy or manipulate because they lack sensitive central points that can be attacked at much lower cost than the economic size of the surrounding system. This can be important for transportation data as it does not remain under a single point of failure.
- Collusion resistance - it is much harder for participants in decentralized systems to collude to act in ways that benefit them at the expense of other participants, whereas the leaderships of corporations and governments collude in ways that benefit themselves but harm less well-coordinated citizens, customers, employees and the general public all the time.

One of the key concepts of blockchains becoming an application platform is smart contracts. A standard contract, as a legal document, binds two or more parties into an obligation to achieve certain deliverables or outcomes. A smart contract is a piece of code that similarly binds multiple parties into outcomes that are verifiable, computable or provable using code and strong cryptographic constructs. Ethereum was the first such platform to introduce the concept of smart contracts which has since been adopted by most other blockchains that wish to host decentralized applications (dapps).

Because smart contracts can be executed by arbitrary nodes on the blockchain, its possible for anyone to "verify" the smart contract. This creates the concept of trust using consensus. Consensus is defined as the agreement among a certain number (or fraction) of nodes on a particular result or outcome. With consensus, it becomes much harder for a Byzantine or adversarial node [28] to manipulate the smart contract in ways that was not intended or provisioned for. One of our goals, as discussed later in Section 2 is to build a smart contract framework that is tailored for Transportation applications, so that the contracts are readily available, trustable and enforceable.

Smart contracts can allow new ways of data sharing that were not possible before. By creating strong programmatic constructs combined with consensus protocols and cryptographic primitives it is possible to share data in a way that privacy, security and restrictions on use can be enforced. There has been a lot of recent progress on how to use Smart Contracts for data sharing [29]. Thus, the technology today is ready for creating disruptive new ways of sharing transportation data to create new applications such as smart cities, driver behavior, insurance, mapping etc. This technology can be disruptive to incumbents who might be late for adoption.

Blockchain software is generally open-source. This increases the trust level to what is not possible in centralized data silos. Open source software removes the need for policies to exist in text, but can be verified in code. Also it allows anyone in the community or the industry to contribute to functionality thereby moving the industry towards standardization which is good for the ecosystem. This makes it possible for the Industry or Government to create regulations or an industry standard and also implement a method of enforcing them on the network.