Edit: Dec-2023, the litepaper is no longer up to date and should not be relied upon for latest info about how the protocol works. We have made many improvements and will release a new whitepaper for Mainnet in Q1 2024.

Lava: A decentralized RPC network for blockchains

Yair Cleper, Gil Binder

22 September 2022

Litepaper v0.1

Abstract

We introduce the Lava Network, a trustless market for blockchain Remote Procedure Calls (RPC). Lava facilitates relay requests to any blockchain. The market runs on an appchain built using the Cosmos Software Development Kit (SDK) and Tendermint Byzantine Fault Tolerance (BFT) consensus.

Lava exists because there are no cryptoeconomic incentives for providing access to blockchains, like the mining / validator rewards that exist for block building. It is no longer reasonable to expect each network participant to maintain their own node. Today, the most commonly used RPC endpoints are provided through trusted Node-as-a-Service or rate-limited public nodes because running a node requires expertise, is time-intensive, costly and requires an increasingly significant amount of compute resources. Growing demand for robust RPC service has led to concentrated reliance on a small group of node operators. This reintroduces a single point of failure for Web3 and requires trust in node runners to ensure data integrity, scalability, privacy, and censorship-resistance.

This litepaper covers Lava's key value proposition, explores the core design mechanisms of the Network, and outlines the future of the Protocol.

1 Introduction

Node infrastructure connects users to the blockchain. Every time a user opens their wallet, makes a swap on a decentralized exchange or interacts with the blockchain via an application, they make an RPC call to a node. The node either reads the blockchain state or sends the user's transaction to the blockchain. This service is crucial to all blockchain-powered software

applications (“Web3”); without it, validators and miners cannot build blocks, nor can applications display current data.

The current state of node infrastructure represents a significant potential attack vector on Web3. If any individual party controls this layer, that party becomes vulnerable to hacks, can manipulate query responses, capture and sell valuable user data, and even censor transactions to specific dapps. Despite this, the market for node services has consolidated into a small group of providers. These providers have serviced the vast majority of Web3 applications up to now, providing an important service for the ecosystem and its developers. However, they represent a single point of failure.

Lava uses a cryptoeconomic incentive framework and appchain to coordinate node runners and applications in the trustless exchange of blockchain RPC service. Web3 must diversify its node operators to ensure that the ecosystem can achieve censorship-resistance; Lava creates this diversity. The Network also includes several novel innovations across its technology stack and blockchain, including mechanisms for ensuring data integrity, scalability and privacy.

Blockchain infrastructure & the tragedy of the commons

Users of Web3 often describe blockchain infrastructure as a common good. Common goods are resources that share two attributes:

- They are publicly accessible by default or are administered by some neutral allocator
- They are rivalrous and require coordination/regulation to maintain fair use

Due to the fact that these public resources can be depleted if left unmanaged and poorly allocated, they are often administered by the government using taxes e.g. fire departments, parks, and roads. An economic problem called the tragedy of the commons arises if there is no entity that carries out this maintenance. Today, public RPC endpoints suffer from such a problem, rendering them poor solutions for scalable blockchain data access.

The tragedy of the commons describes a scenario where a common resource e.g. fish in an unregulated lake, is depleted because individuals act in their own self-interest to exploit that resource. Individuals have little incentive to maintain the lake - the commons - due to the free rider problem, which describes when individuals who benefit from a public resource do not contribute towards the cost of its production. The free rider problem represents a type of market failure, and it is often resolved by centralized intervention. For example, the local government may pass regulations to restrict overfishing and the fisherman may be required to apply for a license.

However, regulation administered by centralized authorities is subject to corruption and a misallocation of focus and funds. Blockchain maintenance requires an alternative decentralized coordination and allocation mechanism such that a tragedy of the commons is prevented.

That alternative is the free market.

Markets & the centralizing need for trust

Free markets are systems where the price of a good reaches equilibrium through decentralized self-regulation by market participants. Rather than a regulatory body deciding which common goods to fund, the “invisible hand”" of the market is the economic concept that describes how rational economic actors are incentivized to create products and services valuable to others.

In Web2, marketplace platforms are useful for suppliers because they provide easy distribution through aggregation. On paper, buyers can also more easily appraise different sellers and pick the best choice. In reality, the best choice is unclear because Web2 markets rely on trust.

In a peer-to-peer marketplace such as eBay, the market is largely decentralized, but the tradeoff is a lack of quality control. How do you know if the item you ordered will come? How do you ensure it is of good quality?

Some marketplaces have emerged that charge a premium in exchange for more active management of the supply side. For example, Opendoor is a real estate marketplace that offers real-time appraisals and bids on homes and holds the inventory for as long as it takes for a sale. Buyers pay a higher price, but the burden of trust on the seller is reduced.

This model requires active management which is often costly; prices rise as a result and take-rates increase. Moreover, the marketplace moves from a peer-to-peer exchange to a centralized vertically integrated supplier. As all markets grow, they converge towards some form of centralization due to this need to maintain a level of trust.

Altruism, light clients & Node-as-a-Service

One essential element of blockchain infrastructure is the RPC layer. Nodes allow access to the blockchain via RPC endpoints. Access to any blockchain data requires calling the relevant RPC nodes.

Web3 has temporarily overcome these issues by self-hosting RPC nodes or relying on rate-limited public RPC endpoints. However, such solutions are either impractical or unscalable. For example, today, running a full node on the Cosmos Hub requires 1.4TB of always-online data storage. An archive node requires 3.1TB. Neither requirement is trivial; 1.4TB is equivalent to storing approximately 1,500 hours of HD video, and this minimum requirement is only increasing as blockchain usage grows.

Light clients are frequently touted as the panacea for node centralization. While useful, they are not a total solution for scalable decentralization. These clients can be used to communicate with other RPC nodes and verify state with lower trust assumptions around data integrity. Despite recent research breakthroughs, light node functionality is still blockchain specific at best, and not available for all blockchains.

Moreover, the mass usage of light clients as a replacement for RPC nodes would quickly outgrow the insufficient supply of nodes required to service them in the peer-to-peer network.

With constrained resources and a lack of RPC node incentivization, a tragedy of the commons problem remains. Lava will use light clients to improve the efficiency of our Conflict Detection process, working with unfinalized data and reducing the trust assumptions with our Conflict Resolution consensus protocol.

Professional “Nodes-as-a-service” businesses have emerged as an essential service to solve these issues. Despite their benefits, these are still centralized, trusted third parties.

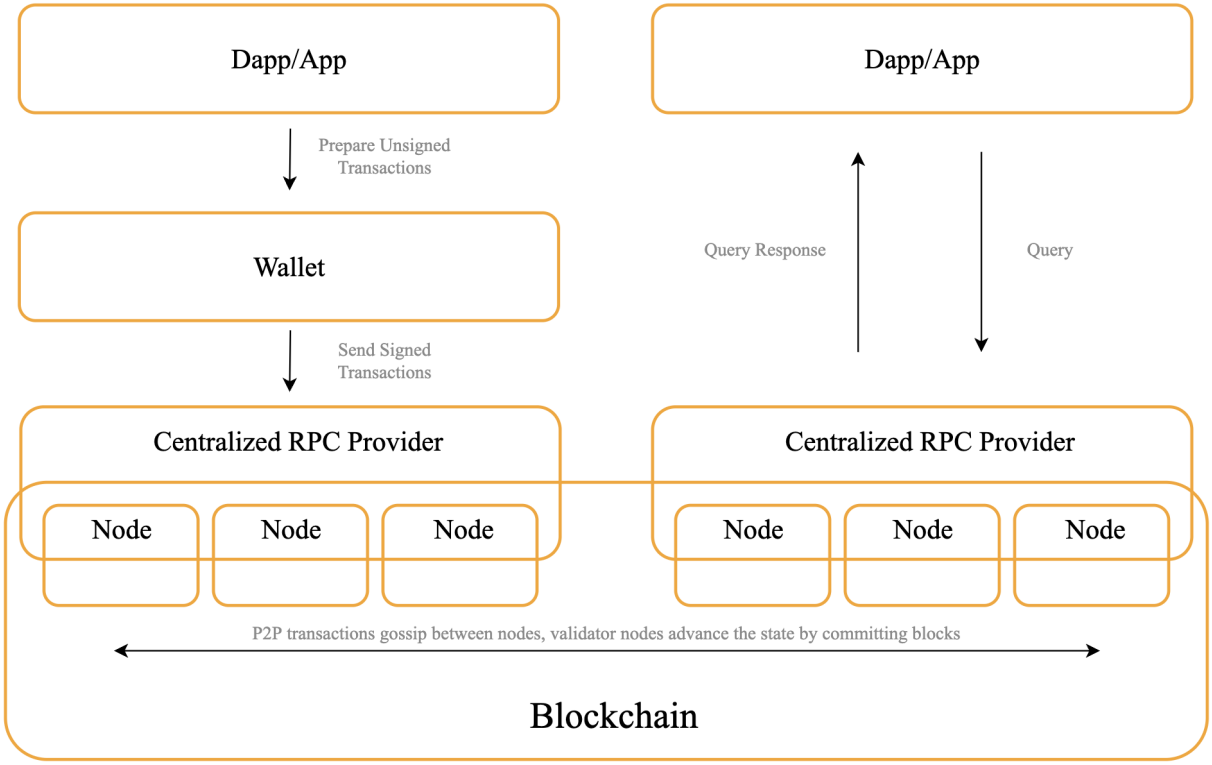
Drawbacks of centralized infrastructure

Dependence on a small group of centralized node providers reintroduces trusted intermediaries back into Web3. The ecosystem experiences consequential damages when these central dependencies are offline, block access, or deliver stale data.

For example, in early 2022, access to major chains such as Polygon and Fantom was severely disrupted because of a DNS hijack. The primary solution was to move to other, equally centralized alternatives. Venezuelan users were similarly impacted when a major RPC provider accidentally censored their IP addresses, severing their ability to use their wallets and therefore blocking their access to Web3. There are many more examples where this single point of failure has led to significant disruption. As long as Web3 continues to rely on trusted third parties, the current RPC layer is an exploitable flaw at the heart of the system.

Figure 1: Blockchain access today

Reliance on trusted third parties



With centralized infrastructure, data integrity is maintained by the impractical process of cross-referencing wallet balances (as an example) against multiple RPC data providers. Additionally, 100% availability can only be achieved by implementing fallback mechanisms which redirect to other nodes if one becomes unavailable. Poor availability can be the result of downtime, mandated censorship or, frequently with public endpoints, high network congestion. Today, the best practice for applications processing high-value transactions, is to use and pay multiple data providers in parallel. The additional costs are justified when data integrity and uptime matters most.

The Lava Network solves these issues by coordinating a globally distributed, decentralized network of nodes, while ensuring data accuracy and network redundancy using cryptoeconomic incentives and consensus mechanisms.

Introducing cryptomarkets

Blockchains can be used to create markets powering decentralized public infrastructure. These markets are cryptomarkets, a specific subset of cryptonetworks.

Cryptomarkets use token-aligned incentives, cryptography and decentralized consensus to ensure that public infrastructure maintains a minimum standard of service, even at scale. Provider

quality is enforced through the Protocol and cryptoeconomic incentives. Because service can be cryptographically signed and conducted off-chain, while payments are settled and verified on-chain, cryptomarkets coordinate completely trustless, peer-to-peer value exchange. *This is one of the primary use cases of public blockchains - the ability to create networks and markets which resist the centralizing need for trust.*

Other parts of the middleware stack are moving towards decentralization by building cryptomarkets. For example, Filecoin and Arweave offer decentralized file storage and The Graph connects users with blockchain data indexers. Cryptomarkets are especially well-suited to public infrastructure, given they incentivize supply to satisfy demand while permanently removing any ability or incentive for suppliers to abuse that relationship e.g. manipulating data, exploiting user privacy, or maintaining poor uptime. Cryptomarkets are consistently reliable, since all suppliers are held to a set of rules and incentivized to provide high quality service.

The Lava Network's novel protocol, cryptography, and aligned incentive framework, is key to unlocking the many benefits of decentralization.

Lava - the cryptomarket for blockchain APIs

Lava Protocol implements a cryptomarket which coordinates RPC nodes and applications.

Until now, developers could only choose from a set of centralized RPC node operators due to a lack of practical alternatives. Applications relying on these Providers either trust the data is not stale, or proactively cross-reference multiple nodes to mitigate security, data integrity and latency issues. This approach is not a sufficient, long-term solution.

Our Protocol governs a peer-to-peer market for RPC data while probabilistically detecting and arbitrating data conflicts. The result is an efficient, trustless market for accurate blockchain data. Access to that data is scalable, private and uncensored.

2 System Overview

Lava Blockchain is a Proof-of-Stake appchain built using the Cosmos SDK and Tendermint Core, a BFT consensus algorithm. This development kit is the open-source standard for creating sovereign, decentralized and high throughput blockchains. Examples include Binance Smart Chain, Osmosis and Cosmos Hub.

Lava operates as a market and settlement layer for accessing blockchain data, underpinned by dynamic market driven pricing. Consumer-Provider pairing, Conflict Resolution and payment settlement are all done on-chain. Data and requests are peer-to-peer and off-chain to improve network scalability but are cryptographically signed and verifiable by both ends. This cryptography ensures that network actors can be held accountable for inaccurate responses and requests; trust requirements are minimized.

The Protocol scales support to new RPCs by adding Specifications ("specs") via governance. Each spec describes the schema needed for the RPC and aligns the different actors on the provided interface.

3 Protocol Design

Lava Protocol can be divided into 4 key stages which govern the interactions and exchange of value between Consumers, Providers and Validators:

- Pairing and Service
- Quality of Service
- Conflict Detection and Resolution
- Payment Requests and Settlement

Lava also deploys scalability optimizations across two dimensions:

- Protocol scalability
    - A *Lazy Blockchain* design to allow for more transactions and *Backfilling* of payments to prevent congestion
    - *Separation of Validator and Provider* roles optimizes the amount of Validators participating in the Network
    - *Vertical Scaling* means Providers receive more work by increasing stake on the same node, rather than adding more nodes. This allows service scalability without increasing the votes needed for Providers to reach an *Honest Majority* consensus during *Conflict Detection*
    - Transactions are conducted off-chain and uploaded on-chain in aggregate, increasing throughput and reducing gas fees
- Scalability of participation
    - All geographies
    - All blockchains
    - All API specs, e.g. REST, GraphQL, GRPC

3.1 Network Actors

There are three core economic agents in the Lava Network: the Validators who secure the Network, the Providers who run services for Consumers, and the Consumers who pay Providers for API access. All actors require LAVA tokens to perform their roles, and staking secures the Network by aligning incentives. Lava's model rewards Providers who provide a high quality of service, subjectively rated by each client [see QoS for more details], and punishes malicious or faulty behavior [see Conflict Detection and Resolution] e.g. providing unreliable data or

persistent downtime. Additionally, there are End Users who use blockchain powered applications but do not pay for RPC usage.

Validators

Validator nodes are responsible for the security of the Lava Blockchain, proposing blocks, voting on blocks and validating state.

Providers

Providers service relay requests by staking on the Network and running RPC nodes on the *Relay Chains* queried by Consumers (e.g. Cosmos, Ethereum, Osmosis, Polygon, etc). They earn fees in the form of LAVA from the Consumers for servicing requests.

Consumers

Developers who build applications on blockchains are Consumers. Consumers initiate relay requests. They are interacting with Relay Chains and participate in the Lava Network to make trustless RPC calls. Lava facilitates decentralized interactions with blockchains enabling Consumers to maximize privacy, transaction speed, data reliability, censorship-resistance, and node availability. Consumers on the Lava Network are categorized into Developers and the Applications they build.

3.2 LAVA Token

To support the economics of the Lava market, the Protocol introduces a native utility token: LAVA. LAVA has the following use cases:

- Transactions fees & transfers
- Paying for Compute Units
- Cryptoeconomics (e.g. for aligning incentives by staking, and for slashing and burning to punish malicious or faulty actors)

4 Consumer-Provider Pairing and Service

Becoming a Provider

Providers permissionlessly join and participate in the Lava Network. After ensuring that their Relay Chain RPC Nodes are operational and meet required specifications, they will install and configure Lava's lightweight Provider process. Providers must stake LAVA via the ProviderStaking(stake,geolocation,chainID,Endpoints[], apiInterfaces) transaction, which defines the spec they support under this stake. There are four parameters used in the transaction:

- Stake: Amount of LAVA to stake for the service
- Geolocation: Location of the Providers Nodes
- ChainID: Identifier of the target blockchain network such as Cosmos, Ethereum, etc
- Endpoints: List of endpoints each defining an address geolocation and an API interface such as REST, JSON-RPC, etc

Provider stakes are per spec. If Cosmos and Ethereum are supported then two separate stakes are needed. After the request is verified and included in the chain state, they will be included in the *Pairing List* from the next *Epoch*. An Epoch is the number of blocks *n* for which a Consumer will use a Pairing List. At the start of the next Epoch, Providers will begin servicing Consumer requests through their nodes.

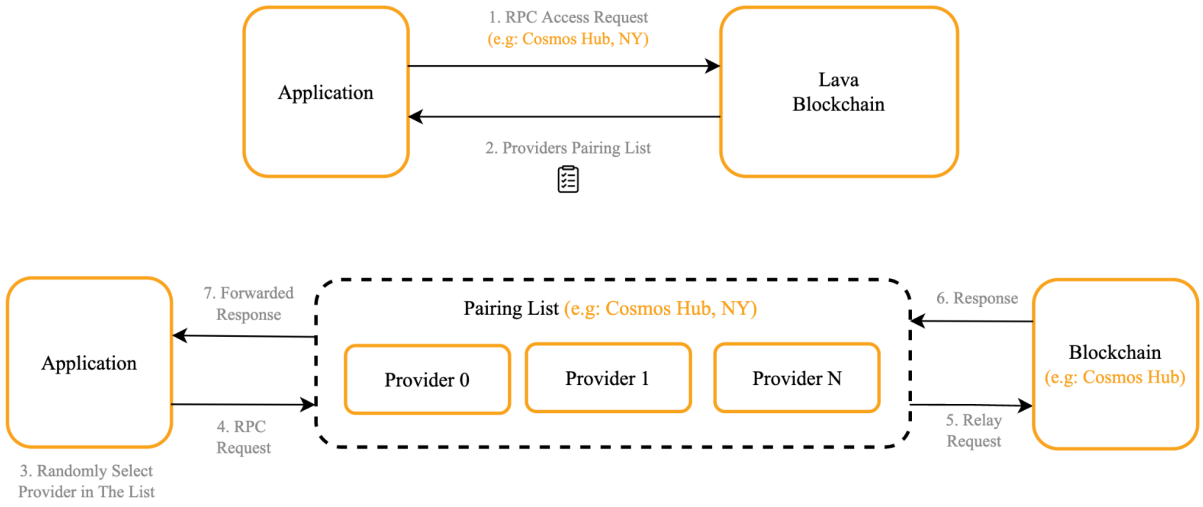
Pairing List Requests

When a Consumer wants to make an RPC call or send a transaction , they request a Pairing List of eligible Providers. Consumers receive a list of Providers for the requested Relay Chain and are providing service in the relevant geography.

Consumers request the most recent Pairing List calculable. To receive this Pairing List a Consumer will submit an API call Pairing() to the Lava Blockchain, returning a list of N service Providers which will service the API requests for that Epoch.

Figure 2: Lava's Peer to Peer Market

Walking Through the Pairing Process



Pairing List Generation Algorithm

A Pairing List is requested by either the Consumer or Provider by calling the function Pairing(Available Services, Blockhash, Consumer pk, API id). A different Pairing List is generated per Epoch for each Consumer. The following parameters are taken into account for Pairing List generation:

- Provider Stake (PS) - Greater stake amounts yield higher chances of pairing with that Consumer
- Epoch Blockhash (BH) - Pseudorandom element
- Geolocation (GL) - The requested geolocation
- Consumer Address (CA) - The public key address of the Consumer

:

Sessions

A Session is an ongoing interaction between a Consumer and a Provider that will provide the relay service. Sessions are represented by a unique session id. A Session is only permitted if the Provider is part of that Consumer's generated Pairing List for that Epoch. Providers can call the Pairing() function and interrogate the result for the inclusion of that Consumer to determine if the session is valid.

Consumers can have multiple concurrent Sessions as they interact with a list of Providers. To increase privacy, throughput and scalability, Consumers should maximize Session usage during an Epoch. When a new Epoch begins, there is an overlap period where relays can still be serviced from their current Session, but Consumers will need to request the latest Pairing List and initiate new Sessions.

Relay Requests

Consumers submit their relay requests to the RPC endpoints of randomly chosen Providers in the Pairing List. The requests are only honored if the Consumer and Provider are paired for that Epoch.

5 Conflict Detection and Resolution

The Lava Network ensures security for its participants by probabilistically sampling random responses from Provider endpoints and implements a consensus protocol for maintaining the integrity of the relay data. These mechanisms can detect, verify, resolve, and prevent fraudulent behavior or conflict.

The Fisherman Test & Conflict Detection

The *Fisherman Test* is a random check initiated by a Consumer to detect malicious or faulty Providers. Using a Verifiable Random Function (VRF) to determine frequency, the Consumer sends a query to at least two different Providers from that Epoch's Pairing List. For selection the Consumer uses the VRF result as a modulo into the deterministic Provider list, this ensures probability of who the query is to be sent to. This check does not incur additional costs.

If responses match, the likelihood of conflict is low. In the case where responses do not match, the Consumer will send a *Conflict Detection* transaction to the Lava chain, and the *Conflict Resolution* process is initiated.

Where a conflict has been detected and the Conflict Resolution has concluded, the stake of the offending Provider and potentially those who voted in support of the Provider will be slashed. The party which detected the anomaly will be rewarded with part of the slashed amount. This model rewards those who act in the interests of the Network and enforces penalties against bad actors.

The Conflict Resolution & Honest Majority

Lava implements an *Honest Majority* protocol to determine which Provider is acting maliciously or is faulty. The Protocol requires all Providers for the spec in the Network to re-execute the relay which is related to the conflict. Providers submit their answers secretly to the Lava Network using a commit and reveal scheme:

1. The Provider commits their answer by hashing it with a salt and public key, before submitting the resulting hash
2. The Provider reveals their answer by sending the result and the salt so anyone can verify their original commit is authentic

After the expiration of the VOTING_WINDOW, the vote is closed and the Network will resolve the conflict by accepting the answer with majority consensus among Providers. Votes are weighted by stake, to increase Sybil-resilience and reduce the possibility of colluding voters.

If no consensus is achieved in the Honest Majority protocol due to low Provider participation, then there is no resolution and missing Providers are penalized. If consensus is reached in the Honest Majority and the total stake of Providers which voted for the majority consensus decision meets the PROVIDER_OFFENDING_SLASHING_THRESHOLD, the offending Provider is slashed. If the total stake of Providers which voted for the majority consensus decision also meets the PROVIDER_ACCURATE_VOTING_THRESHOLD, Providers which voted in the minority are slashed along with the offending Provider.

Penalty

Following Conflict Resolution, dishonest or faulty actors will be slashed as a penalty and honest actors will be rewarded.

The following actors' stake in the Network will be slashed as a penalty:

- The Provider who caused the conflict and submitted fraudulent or incorrect data
- The Providers who voted and reported wrong answers

A portion SLASHED_TOKEN_BURN_AMOUNT of the slashed tokens will be burned and the other part will be used to reward honest actors. The following actors will be rewarded:

- The Consumer who detected the conflict
- The Provider who responded with the consensus answer in the initial verifiably random check (if consensus was reached)
- All the other Providers voted for the consensus answer in the Honest Majority

Consumer & Validator Honesty

Validators stake LAVA in the Network, and the Network disincentivizes any malicious behavior by slashing bad actors. Lava uses the default Tendermint and Cosmos consensus protocol to enforce Validator honesty.

Consumers must prepay to participate in the Network. After a session is complete, Providers submit an aggregated payment request, the value of which is deducted from the initial Consumer payment and burned.

6 Quality of Service

The Network is designed to account for subjective differences in service requirements across different applications. Consumers assign a *Performance Score* from 0.0 to 1.0 to each Provider per session. This Score is measured across three key areas:

Latency: Latency measures the total time the Consumer waits from sending the request to receiving the Provider's reply.

Sync: Sync is a measure of the freshness of the response returned by a Provider. This is dependent on the Provider having synced to the latest block of the Relay Chain state and returning accurate responses to requests from Consumers. Consumers will discover the Provider's latest block height from their responses and can compare these values to determine the freshest or stalest Providers.

Availability: Availability refers to the accessibility of the server, its uptime on the Network and its reliability in responding to requests. This is measured by a Provider's response coverage

against all requests; a Provider with intermittent responses will receive poor Performance Scores. Within Lava, all Providers must also expose the full API to Consumers; if a Provider does not support any API functions, they will earn a lowered Performance Score.

Provider Performance Score

Each Consumer is able to submit a Performance Score for their Service Providers, by subjectively rating based on the quality of service received. An individual Score contributes to a Provider's overall score. If the Score crosses a low enough threshold PERFORMANCE_SCORE_THRESHOLD, the payment made to the Provider for that session is lowered and burned. Consumers do not receive this refund and have no incentive to give disingenuous Scores.

Provider Fee

Lava calculates the fee (Pf) the Consumer pays LAVA (L) which is a product of the Compute Units (Cu) used for the request. The Provider's *Payment Request* (*PR*) fee may be adjusted if they currently have a lower Performance Score; they will still receive rewards, but at a reduced amount (Fm), although this payment discount only affects the specific Session in which the Provider received the low Score. The fee is calculated using the following formula:

$$P_f = (L * C_u) * F_m$$

Fee Modifier

A Provider's Performance Score is reflected in the rate of fees earned. When the standard of service is degrading their score (Range 0 to 1) the earned fees will be reduced linearly with the Performance Score and capped at a maximum reduction of PERFORMANCE_SCORE_CAP. The DAO governance can raise a proposal to upgrade and reconfigure this threshold.

$$PR = \begin{matrix} PPS \geqslant 0.5 \ \ then \ \ PPS, \\ PPS < 0.5 \ \ then \ \ 0.5 \end{matrix} \Bigg\}$$

7 Payment Requests and Settlement

Provider Payment Requests

A Provider will verify a Consumer's staked LAVA (Cs) is sufficient to cover the Compute Units (Cu) used in a submitted request. Consumers prepay for the service. If there is no LAVA (L) or

an insufficient amount is paid, the Provider will not process the Payment Request (PR). Providers check a Consumer's prepaid amount to determine whether they can pay for the needed computational units for its request, ensuring the Provider's eventual compensation for servicing relays. This calculation is represented by the following equation:

$$PR = Cs - (Cu \ * \ L) > 0$$

Settlement

Consumers have their deposited LAVA allocated to session slots, each slot is one Epoch which is an individual session. This guarantees the Consumer has funds available for any services used within that epoch. This locking mechanism is key in allowing Backfilling of payments requests by Providers who may have become disconnected from the Network. After the maximum duration of Backfilling has expired the LAVA from that slot can become re-locked for future slots. The Backfill period BACKFILL_PERIOD can be adjusted by Lava governance.

8 Scalability

Given that Lava aims to power the RPC layer for all blockchains, the Protocol implements several mechanisms which increase throughput, minimize gas fees during high activity periods and reduce the amount of data stored on-chain. Scalability is also encouraged across participation; Providers are incentivised to offer services across all blockchains, geographies, applications and API specifications.

8.1 Protocol Scalability

Lazy Blockchain & Backfilling

Compared to other blockchains in the Cosmos ecosystem, the Lava Blockchain favors larger block sizes. This ensures that more transactions can be included in a block. Additionally, to mitigate the impact of high-used periods where gas fees are high, Providers do not need to submit payment requests instantly. Instead, the Network uses a lazy settlement process which allows Providers to retroactively make payment requests when gas is low.

This process of backdating reward claims is referred to as Backfilling. The ability to submit delayed payment requests ensures that Providers who have suffered from a brief data outage or unexpected downtime will typically have sufficient time to claim rewards.

By using this lazy design, Providers are encouraged to focus on servicing as many Consumers as possible, while minimizing periods of high-demand and congestion.

Separation of Roles

Lava separates all roles in the Network, to ensure that there is no additional burden on network actors to serve more than one type of role. For example, anyone who wants to become a Provider in the Network runs a Provider process, without needing Validator node functionality. This approach makes it easy for actors with blockchain access to offer their services through the Lava Network.

In case the same actor wishes to secure the Network and validate blocks, they can install a full Lava node and serve as a Lava Validator, simultaneously validating blocks while also providing Lava chain RPC access.

This approach and separation of roles provides flexible participation to anyone who wishes to participate in the Lava Network, whether they are a Provider, Validator, or both. The design prevents unnecessary processing from the machine of the Provider or Validator.

Vertical Scaling

Lava simplifies the process for Providers to participate in the Network, serving relays, and earning rewards by letting them use one on-chain identity for all their services. If Providers wish to serve more clients and earn more rewards, they do not need to open a new account to create a new address for additional staking. Instead, they can simply increase their stake in the Network within their existing account, benefiting them in the stake-weighted pseudorandom pairing function Pairing(Available Services, Blockhash, Consumer pk, API id), subsequently yielding more pairings and higher rewards. No new nodes or addresses are required to increase the number of Consumer pairings a Provider receives.

This approach discourages horizontal bloating and duplication of machines and addresses which create a false sense of decentralization.

Instead, Lava allows and encourages vertical scaling to reduce unnecessary messages and on-chain data. A Provider is incentivized to gain a Quality of Service reputation and own only a limited number of identities on the Lava Blockchain by linking a single address and signing key to the multiple endpoints and services that they provide.

On-chain Data Minimization

To improve privacy and further minimize the number of on-chain transactions, Lava does not record RPC related communication between the Providers and the Consumers. The blockchain only tracks macro-data.

Examples of data stored on-chain:

- Introducing new participants to the Network (Providers, Consumers, and Validators)
- Payment settlements

- Queries and messages used in Conflict Detection and Resolution

Examples of data not stored on-chain:

- RPC relays exchanged between Consumers and Providers
- Timestamps of Consumer-Provider communication

By storing only a minimal amount of data on-chain, the Network is able to significantly optimize performance. In total, there will be only one transaction per Consumer per Provider per Epoch. Further on-chain data storage optimizations are planned.

Reports & Compute Units (CUs) Aggregation

To handle high transaction frequency and keep transaction costs low, the Network uses Payment Channels to track microtransactions between Consumers and Providers after each Compute Unite is transferred. Transactions are aggregated, signed and settled on-chain within a configurable claims period, initially set by parameter BACKFILL_PERIOD in alignment with the Provider unstake period.

Consumer-side Aggregation: Consumer-Provider communication occurs off-chain. The Consumers aggregate all relays and CU served in the session into one cryptographically signed message. Once the session is finalized, the Consumer signs the message and sends it over to the Provider.

Provider-side Aggregation: After the Provider has received multiple cryptographically signed messages from multiple Consumers, they will combine them into a bulk payment request message and submit it to the Lava chain to claim their payment. Here is an example of payment request aggregation from the Lava source code:

```
func (s *Sentry) UpdateCUServiced(CU uint64) {

        …

        currentCU := atomic.LoadUint64(&s.totalCUServiced)

        atomic.StoreUint64(&s.totalCUServiced, currentCU+CU)

}
```

To promote efficiency, the Protocol encourages aggregation which is optional:

- Aggregating messages reduces gas fees for the Provider
- Aggregated report messages cost less gas than individual transactions because different message types on the Lava Blockchain will vary in gas cost. Variances in gas cost are explained the computing resources required by opcode (e.g. EVM opcodes gas prices)

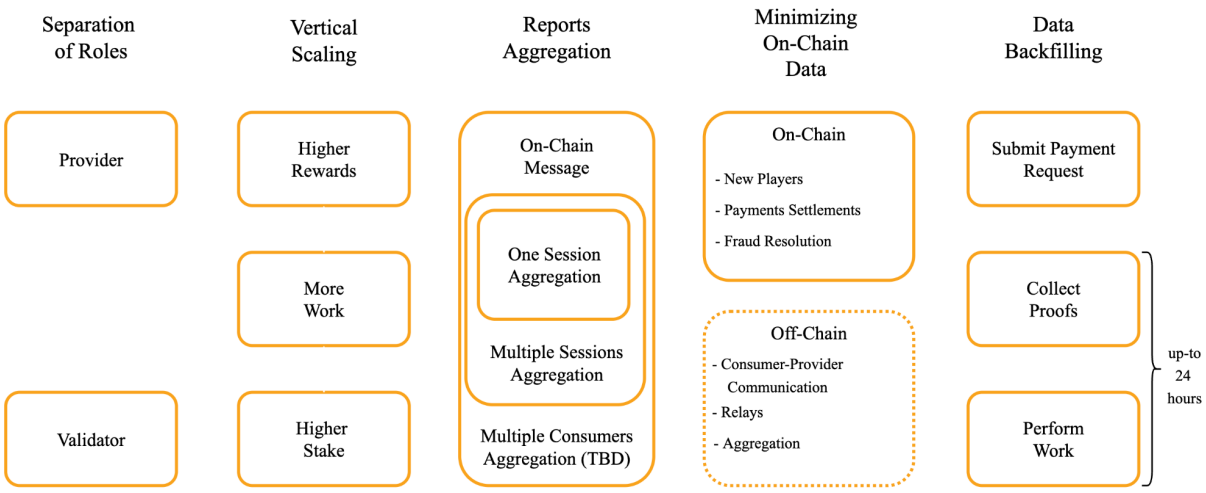
8.2 Participation Scalability

The architecture of the Protocol allows scalable use for any blockchain. The design encourages modular onboarding of new blockchains, to support each ecosystem of applications. We consider growth and scalability in four primary dimensions:

- Across chains: As new blockchains are launched, Providers are incentivized to join the Lava Network and bootstrap the new chain.
- Across geographies: Locations where interest and adoption are increasing present opportunities for Providers to expand services.
- Across applications: the Protocol supports all applications requiring access to blockchain data.
- Across API specifications: In the future, the Protocol will enable Providers to service relays for any API.

Figure 3: Scalability Mechanisms

Understanding the Lava Protocol



9 Token Economics

LAVA is the native token of the Lava Network, used to pay for transactions, align incentives between participants and to secure and govern the Network. Mechanics have already been described in the relevant sections above, but the following is a summary of each token use case.

Staking

Staking is the process by which an entity will bind a number of LAVA to the Protocol. Lava is a Proof of Stake blockchain and the staked amount aligns incentives between network participants; the higher the stake, the more invested they are in the Network and more probable they will act in the Network's best interests.

- Validator Staking: Lava is a PoS blockchain in which Validators are required to stake LAVA to participate in state consensus and secure the chain
- Provider Staking: Providers are incentivized to stake and increase their stake amount for improved reputation and greater visibility in Pairing Lists generated by a stake-weighted pseudorandom function. Increasing their stake will grow their request throughput and their earnings, and serve as an incentive for providing honest data
- Unstaking: Staked tokens can be unstaked at any time, there is an unbonding period which is sufficiently long for finality of transaction payments

Transactions & Transfers

- Lava Token Transfer: The transfer of the LAVA token from one account to another
- Consumer-Provider Service: Consumers deposit LAVA tokens to pay for their service / Compute Units

Consumers make relay requests to Providers and the Provider account claims LAVA rewards for the service they provide. Consumers' fees are proportional to the amount of effort or Compute Units used for their requests. The Providers receive payment after a Session has completed and they submit a Provider Payment Request for that Session.

Compute Units

Compute Units (CUs) are set for every single service API in every spec. They reflect how much work a Provider has to do to service a relay.

They are used when calculating relay payments for Providers. The more CUs a Provider serviced - the higher the reward.

Incentives & Rewards

- Validator Rewards: Validators receive LAVA for proposing blocks
- Provider Rewards: - Providers receive LAVA for successfully completing work

In addition to the uses outlined above, LAVA incentivizes network actors to bootstrap new geographies and specifications. This is beneficial to both network participants, who earn as the Network scales, as well as new chains which require initial node support.

Slashing / Burning / Jailing

Slashing is an economic disincentive for malicious and also faulty actors. If a Provider is caught for wrongdoing (see Conflict Detection and Resolution section below) the stake will be slashed. A portion of it will be rewarded to the Consumer that detected the conflict, the Providers that voted on the correct resolution and a portion will be burned.

Burning is where tokens are removed from the overall supply of LAVA. Burning occurs in the following instances

- When slashing
- When Consumers pay for service

When a Provider is jailed they are still a staked Provider but are no longer eligible for rewards or inclusion in Pairing Lists. They remain in this state until their jailing time expires or they post bail by increasing stake to remove themselves from the jailed state. Providers are jailed if their service is offline for a short amount of time.

10 Roadmap

Decentralized access (Daccess)

Access to the Network will initially be facilitated via a gateway for testnets and Consumers with low RPC usage. Lava supports a fully decentralized alternative which will be announced closer to the mainnet.

Governance

The Network will launch with a fully DAO-governed model, once all the key components of the Protocol have been built and thoroughly tested.

Conflict Detection & Resolution

The Network currently relies on Honest Majority consensus in the case of conflict detection. Future improvements to Conflict Resolution will be implemented to reduce resource consumption and later remove the requirements for majority consensus, namely by (1) introducing a jury protocol, (2) deploying an Honest Minority protocol and (3) using cryptographic proofs.

Privacy

The Protocol minimizes profiling abilities by randomly distributing Consumer relays to Providers on the Pairing List. However, despite eliminating advance knowledge of Consumer-Provider pairings, Providers can still create a fragmented image of a Consumer’s

query history. Lava will later implement a mixer mechanism which completely eliminates this possibility, ensuring private pairing and communication process.

Quality of Service

In the future the Network will cluster Providers and Consumers by score across the three parameters: sync, latency, availability. E.g. if a Consumer gives a Provider a good score, they are more likely to pair with them in the future. In this way, Consumers can optimize their service across sync, latency and availability, and pair with Providers rated highly across the most relevant parameters.

Services beyond RPC

Specification support for enhanced APIs will be introduced later.

11 Conclusion

We believe blockchain data should be accurate and access to that data should be scalable, private and uncensored. Web3 represents a paradigm shift in how we operate as a society, enabling distributed ownership and novel mechanisms for solving complex coordination problems. We believe the Web3 ecosystem should be accessible to all without prejudice or discrimination.

Web3 continues to grow as a global cultural, economic and technological movement. As it expands, the proposition that each participant runs their own node becomes more difficult to realize. As Moxie Marlinspike wrote in his first impression of Web3, “People don’t want to run their own servers, and never will.” The only alternative which adheres to Web3’s foundational premise of decentralized access is Lava. Lava Protocol powers an unstoppable market for blockchain API access which will run permissionlessly and censorship-resistant, forever.

Lava Protocol is a public good, creating a Web3 accessible to all.

Appendix

Lava Blockchain Parameters

The following are some of the initial blockchain parameters which are configurable by DAO governance:

BLOCK_TIME: The amount of time it takes to build a block on the Lava Blockchain, currently set at ~1-2 minutes

EPOCH_TIME: The amount of time a Consumer for which is able to use a Pairing List, currently set at ~30 minutes

BLOCK_SIZE: The maximum block size on the Lava Blockchain, currently set at 100MB (Tendermint default)

UNBOND_PERIOD_CONSUMER: The period of time in which Providers can claim unpaid payments after a Consumer has initiated the LAVA unstaking process

PERFORMANCE_SCORE_THRESHOLD: The lowest Performance Score a Provider can receive before their payment fee is reduced

PERFORMANCE_SCORE_CAP: The maximum % reduction in fees a Provider can receive due to a poor Performance Score

BACKFILL_PERIOD: The period of time in which Providers can backfill a Payment Request, currently set at 24 hours

SLASHED_TOKEN_BURN_AMOUNT : The portion of fees which will be slashed from dishonest or faulty actors during a Conflict Resolution process

PROVIDER_OFFENDING_SLASHING_THRESHOLD: If an Honest Majority consensus is reached during the Conflict Resolution process and the total stake of Providers which voted for the majority consensus decision exceeds this threshold, the offending Provider is slashed

PROVIDER_ACCURATE_VOTING_THRESHOLD: If an Honest Majority consensus is reached during the Conflict Resolution process and the total stake of Providers which voted for the majority consensus decision , Providers who voted in the minority are slashed

VOTING_WINDOW: The period in which Providers are allowed to voting in the Honest Majority protocol, as measured by number of blocks added in that period

Glossary

RPC: RPC is a form of client-server request-response interaction - the client sends a request and the server responds to that request. RPC is the most common software communication protocol used to interact with blockchains

RPC Node: A node which has opened its RPC endpoints, allowing clients to make RPC requests

Relay: A blockchain RPC request and response transmitted using the Lava Protocol

Compute Unit (CU): Compute Unit is the unit of measurement for the resources consumed by Providers carrying out relays

LAVA: The native token of the Lava Blockchain used to secure the Network, pay for service, and incentivize Validators and Providers

Lazy Blockchain: A blockchain characterized by large block sizes and the ability to backfill transactions. Lava runs on a Lazy Blockchain

Backfilling: The process by which Providers can submit payment requests and proof of service retrospectively within a configurable time period (BACKFILL_PERIOD)

Epoch: The number of blocks *n* for which a Consumer will use a Pairing List

Session: A parallel connection between a Consumer and a Provider. There can be multiple sessions at once i.e. Consumers can send relays requests to multiple Provider endpoints