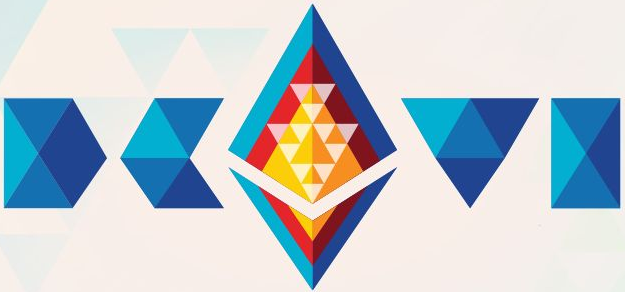


Decentralized Blockchain Data Supply Chain

Supporting the Ethereum Vision & Roadmap

Brandon Ramirez

Co-Founder, Edge & Node + The Graph

@RezBrandon

Section 1

Blockchain Data Supply Chain

Blockchain Data Supply Chain

Sends Blockchain Transaction

Reads Blockchain State

Blockchain Data Supply Chain

Writing and Reading

Sends Blockchain Transaction

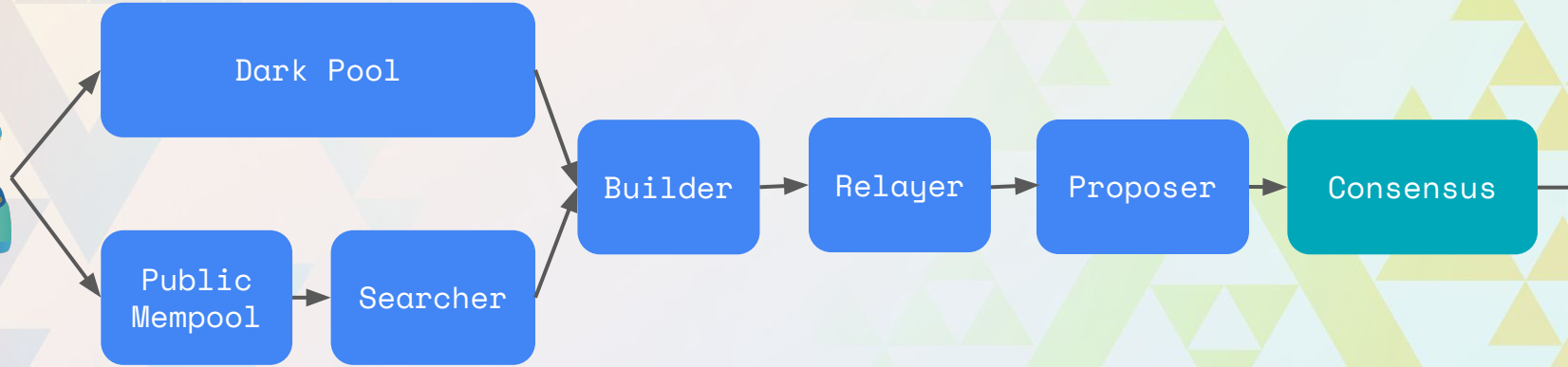
Writing State → Consenus → Reading State →

Reads Blockchain State

Blockchain Data Supply Chain

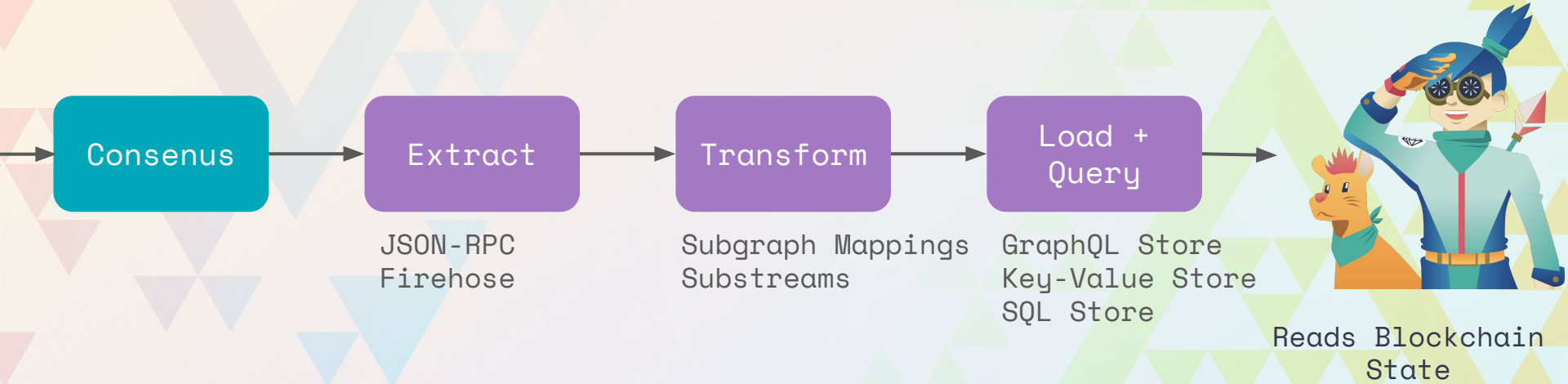
Writing Blockchain State

Sends Blockchain Transaction



Blockchain Data Supply Chain

Reading Blockchain State



Decentralization in the Data Supply Chain

Two high-level categories describe the degree of decentralization in the blockchain supply-chain.

- Web2.5
  - Decentralized writes.
    - **i.e., Proposer/Builder Separation, MEV Boost**
  - Centralized reads
    - **Proprietary APIs, SaaS Platforms, Centralized Servers, etc.**

Blockchain Data Supply Chain

Write Failure Modes

Sends Blockchain Transaction

Writing State

Consenus

Reading State

Reads Blockchain State

Failure modes while writing to the blockchains are highly visible.

i.e., Consensus failure, obvious censorship

Decentralization in the Data Supply Chain

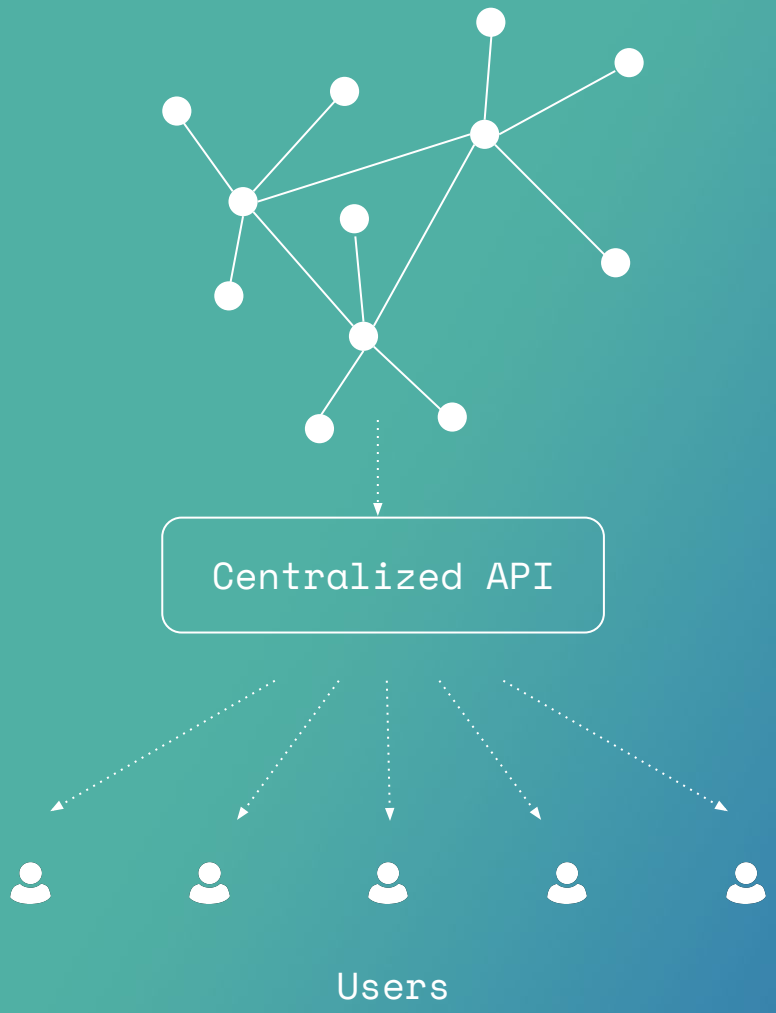
Two high-level categories describe the degree of decentralization in the blockchain supply-chain.

- Web2.5
  - Decentralized writes.
    - **i.e., Proposer/Builder Separation, MEV Boost**
  - Centralized reads
    - **Proprietary APIs, SaaS Platforms, Centralized Servers, etc.**
- Web3
  - Decentralized writes
  - Decentralized reads
    - **i.e., The Graph, Portal Network, Pokt Network, TrueBlocks, etc.**

Section 2

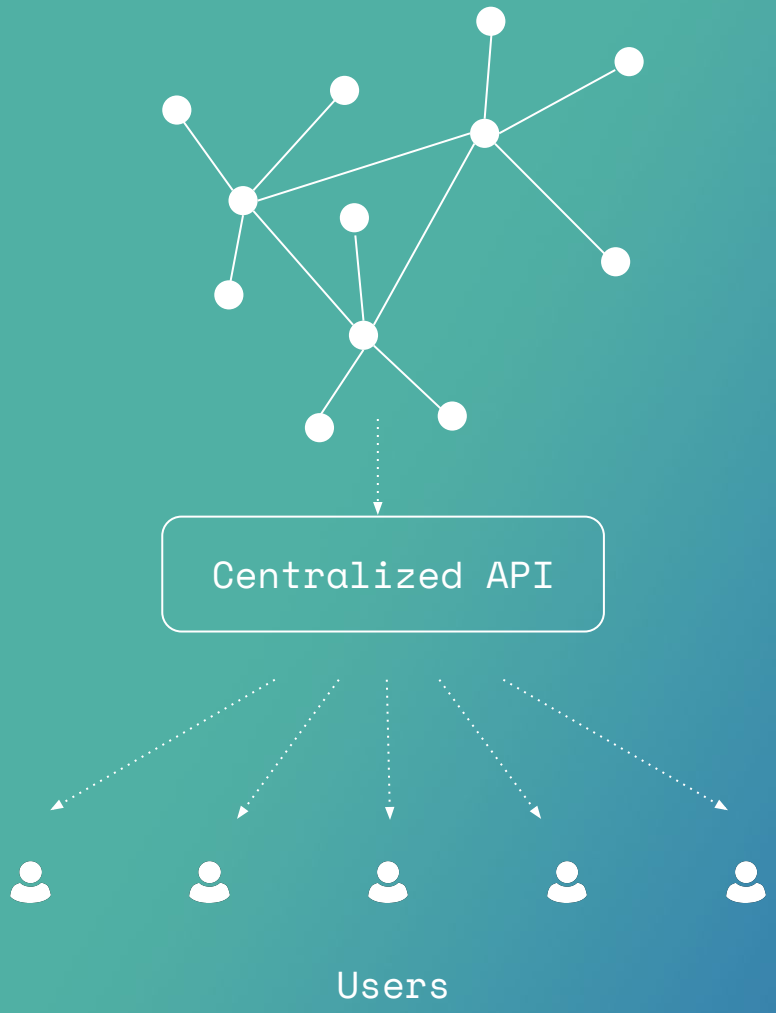
Why Decentralized Reads Matter

Reading a Decentralized System through Centralized Intermediary



“If we don’t stop relying on infura, the vision of Ethereum failed. Or build a strong network of thin and light clients. There is no point in having d-apps connecting through metamask to a blockchain hosted by someone else.”

-Afri Schoedon, Oct 2018



Web2

Web3

Web2

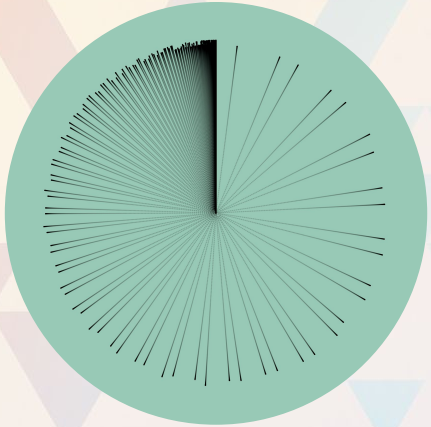
Web2.5

Web3

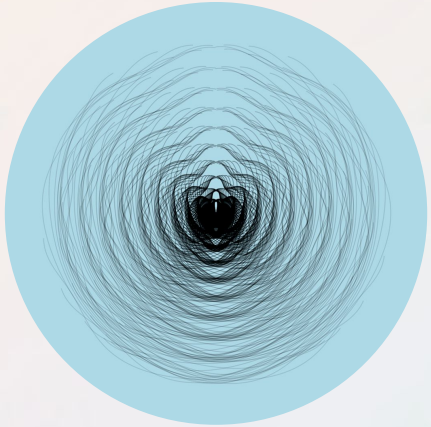
Why Decentralized Reads Matter

Centralizing read access to the blockchain undermines some of the core value propositions of Ethereum + Web3.

- Unstoppable applications
- Composability
- Reusability
- Ownership
- Censorship resistance
- Fairness
- etc.



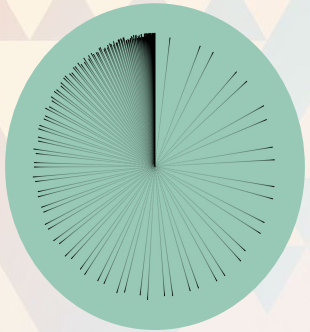
Same NFT on OpenSea



Same NFT on Rarible

Same NFT in a wallet

Source: https://moxie.org/2022/01/07/web3-first-impressions.html



“All this means that if your NFT is removed from OpenSea, it also disappears from your wallet. It doesn’t functionally matter that my NFT is indelibly on the blockchain somewhere, because the wallet (and increasingly everything else in the ecosystem) is just using the OpenSea API to display NFTs, which began returning 304 No Content for the query of NFTs owned by my address!”

-Moxie Marlinspike, Jan 2022

Source: https://moxie.org/2022/01/07/web3-first-impressions.html

Indexer Extractable Value (IEV)

Partially analogous to MEV, Payment for Order Flow (PFOP)

Thread

Tom
@ThomasConlan2

Indexer Extractable Value, IEV, is it a thing? @graphprotocol @0xsamgreen @RezBrandon @yanivgraph @_cryptovestor 💁‍♀️✨💁‍♀️✨💁‍♀️ Questions below.

5:19 AM · Apr 28, 2022 · Twitter Web App

Tom @ThomasConlan2 · Apr 28
Replying to @ThomasConlan2
Does Indexer Extractable Value, IEV, exist? Can the arrival of queries from a DAPP be analysed and built into a strategy for making money? If indexers are the first to see what data the users want could ML be used to make statistically backed bets based on user clicking?
1 1 2

Tom @ThomasConlan2 · Apr 28
e.g. a particular NFT collection on a decentralised, graph-utilising exchange receives 5x the normal query load one day, the floor price of NFT collections on average rises by 6.7% the following day.
1 1 2

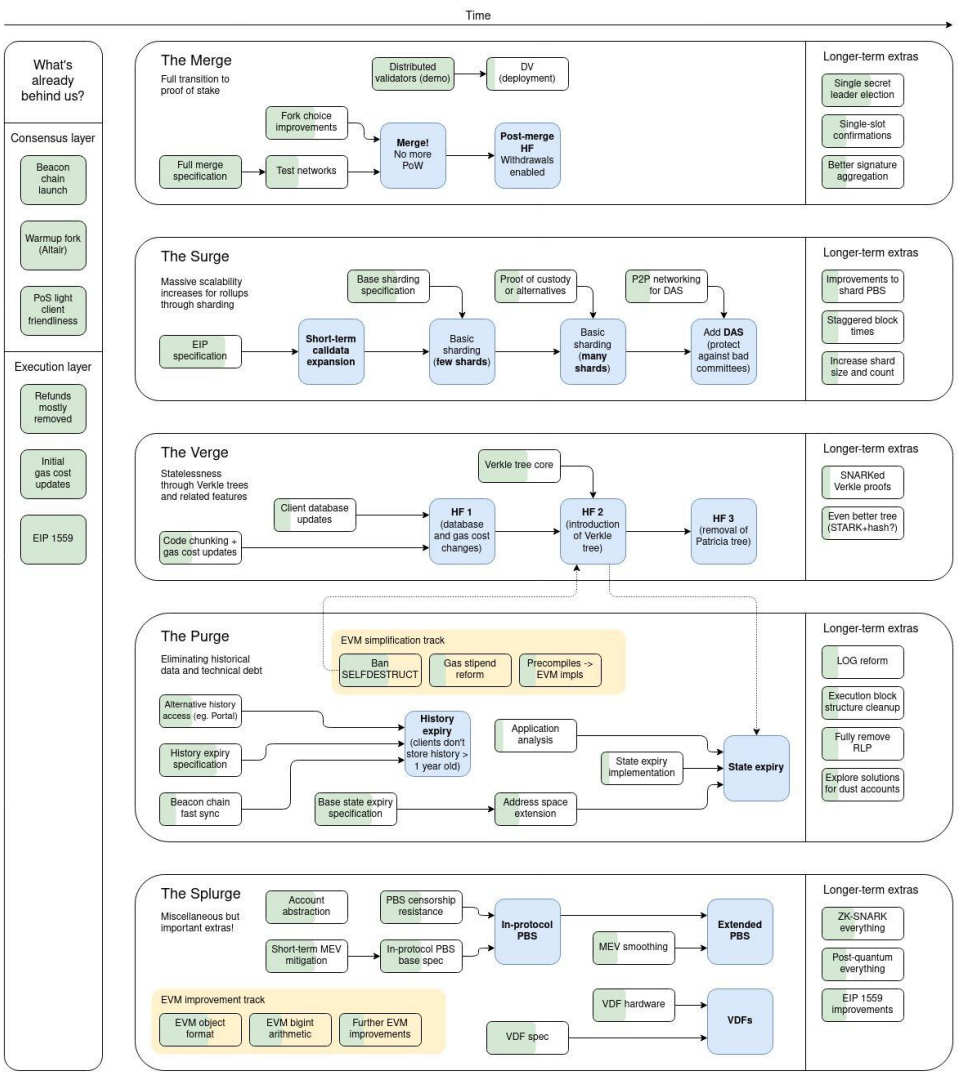
Tom @ThomasConlan2 · Apr 28
Front-running the fi 5:19 AM · Apr 28, 2022 -running miners, front-running based on sentiment. Analogous to a weaponised Google Trends except centred in the hands of Indexers.
1 1

Tom @ThomasConlan2 · Apr 28
Big money may start paying indexers for vision of their query stream.
1 1

Tom @ThomasConlan2 · Apr 28
Basically, can anything be gleaned from incoming queries? I ask all this with the hope that we don't head in that direction.
Thanks 💁‍♀️
1 1 2

Section 3

Supporting the Ethereum Roadmap

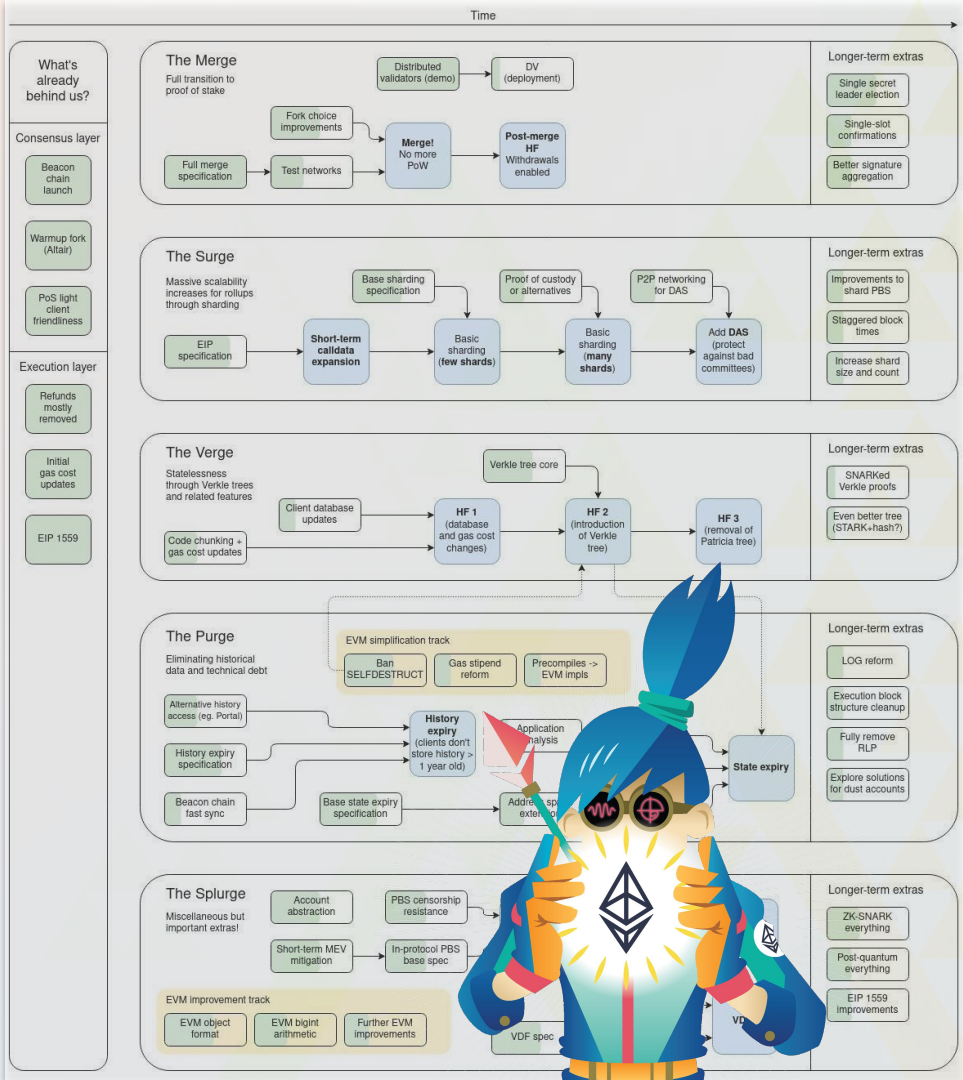


The Merge,
The Surge,
The Verge,
The Purge,
The Splurge…

…Oh my!

Ethereum Roadmap In-Focus

- The Verge
  - Stateless Clients
    - **Weak Statelessness**
    - **Strong Statelessness**
- The Purge
  - History Expiry (EIP-4444)
  - State Expiry
- Light Clients



“[Light Ethereum Subprotocol] is vast desert of starved clients desperate for data.”

-Piper Merriam, Jan 2021

Light Client Problems

“However, a light Geth node relies upon full nodes serving light node data. Few full nodes opt to serve light node data, meaning light nodes often fail to find peers. There are currently no production-ready light clients on the consensus layer; however, several are in development.”

Source:
https://ethereum.org/en/developers/docs/nodes-and-clients/

Ethereum Roadmap Dependencies

These parts of the Ethereum Roadmap require a reliable way for **efficiently and verifiably** accessing:

- Historical/ Pruned Blocks
- Expired or Uncached State

The solution space break down:

- Financially Incentivized
  - i.e., The Graph, Pokt Network
- Tit-for-Tat Incentives
  - i.e., Portal Network
    - **Gossip Network, DHT**

Ethereum Roadmap Unlocks: Decentralized Scaling!

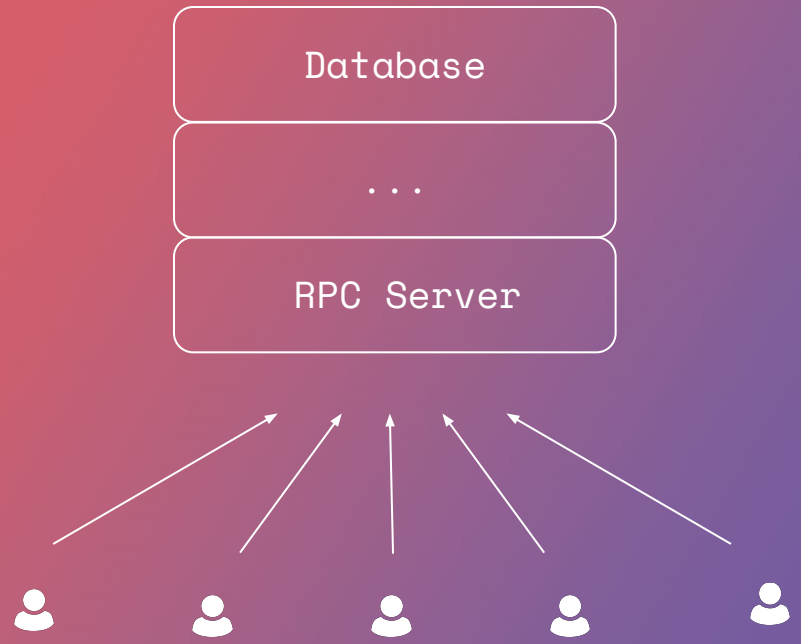
Keeping the footprint of validator and light nodes small while allowing for larger blocks.

- i.e., EIP-4488
  - Transaction calldata gas cost reduction with total calldata limit
    - **~5x cheaper call data**
  - Intended to be paired with EIP-4444 (History Expiry)

Section 4

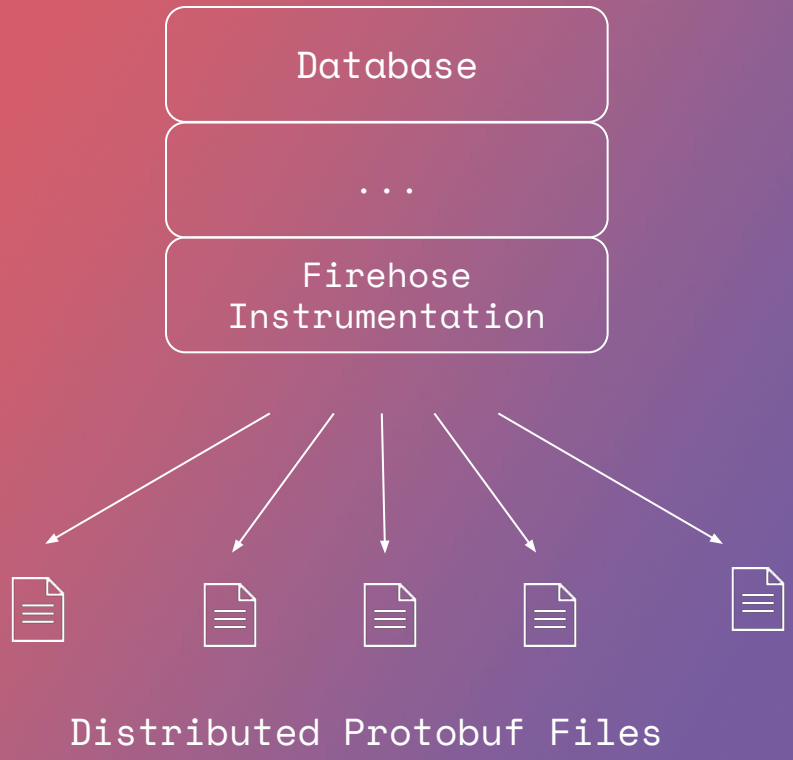
Unleashing the Firehose

STREAMING FAST



Challenges with JSON-RPC

- Depends on a running program for reads
  - Performance bottleneck
- Resource-intensive archive nodes
  - CPU, Memory, SSD
- Difficult to query intermediate states
- Difficult to debug
  - i.e., Dropped log messages
- Incomplete Verifiability Story



The Firehose Approach

- Streaming-first
  - Data pushed out as fast as it's available.
- Files-based
  - Can be stored on commodity hardware
  - Distributed for parallel processing
- **One to two order of magnitude improved read performance.**
  - (When paired with something like Substreams)

Firehose Integration

Firehose must be paired with a verifiability solution to support secure outsourced queries.

Integration strategies:

1. Basic Extraction
    - (Similar to how JSON-RPC is used today)
2. Verifiable extraction
    - Solution Space
        - **Read-oriented Optimistic or ZK Rollups**
        - **Integrate into consensus**

Firehose Integration

Firehose must be paired with a verifiability solution to support secure outsourced queries.

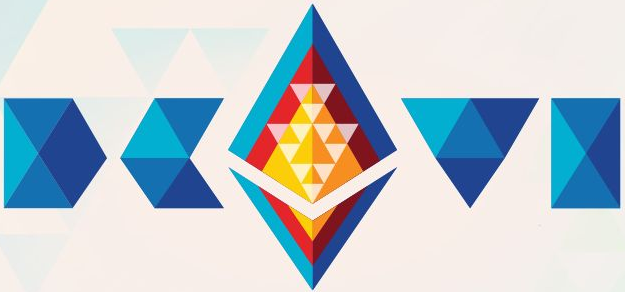
Integration strategies:

1. Basic Extraction
    - (Similar to how JSON-RPC is used today)
2. Verifiable extraction
    - Solution Space
        - **Integrate into consensus**
        - **Read-oriented Optimistic or ZK Rollups**

Also, Shim Client Integration to support The Purge (EIP-4444)!

First Self-Service Case Study!

APTOS



Thank you!

Brandon Ramirez

Co-Founder, Edge & Node + The Graph

brandon@edgeandnode.com

@RezBrandon