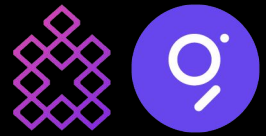


PINAX

Advocating for Firehose & Substreams

March 30th 2023 Graph Advocates

Who is Pinax?



- Web3 Service Provider
  - specialized in blockchain data indexing operations
  - building data analytics tools and products
- Indexer for The Graph and passionate community member
- Manage our own bare metal hardware
  - multiple hosted data centres
- Operating firehose-related tech stack for more than 3 years
- Self-funded
  - 7 co-founders
  - value resilience, independence, and decentralization
- Headquartered in Canada
  - growing international team approaching 50 members

Have we met?

The year of the SPORK

ETHDENVER 2023

#Buidlathon + Workshops + Community Gathering + Innovation Festival





Table of Contents

- Why is Firehose needed?
- Firehose Architecture
- What are Substreams?
- Firehose/Substreams for ...
  - ... decentralized applications (dApps)
  - ... developers of all kinds
  - ... blockchains
  - ... indexers
- Questions?



First of all: Who built Firehose?



StreamingFast Co-Founders

STREAMING FAST

Why does Web3 need Firehose?

This is ethereum ➡

an ocean of blockchain data

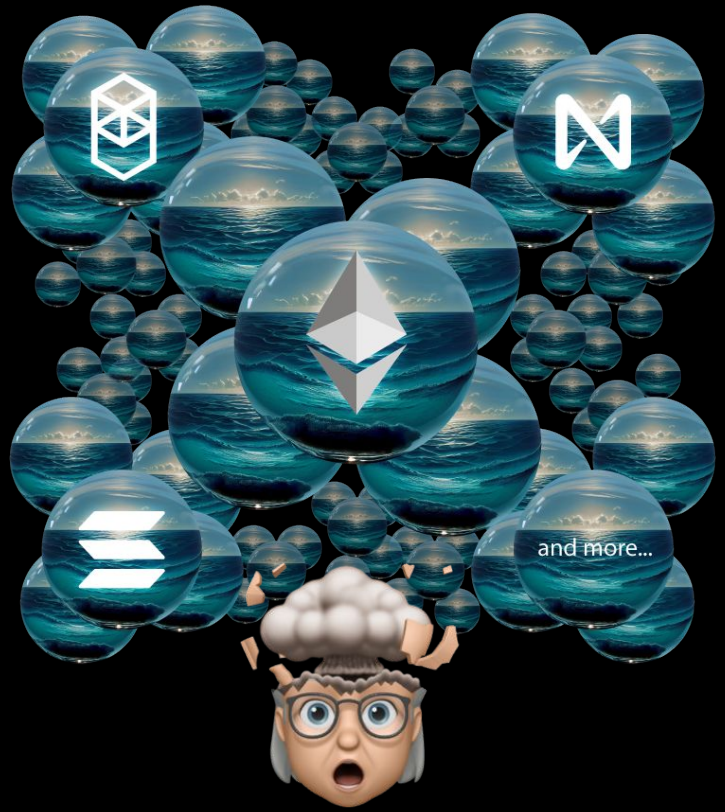


The Graph makes it possible to organize this data: What you need; How you need it!

But wait there's more…

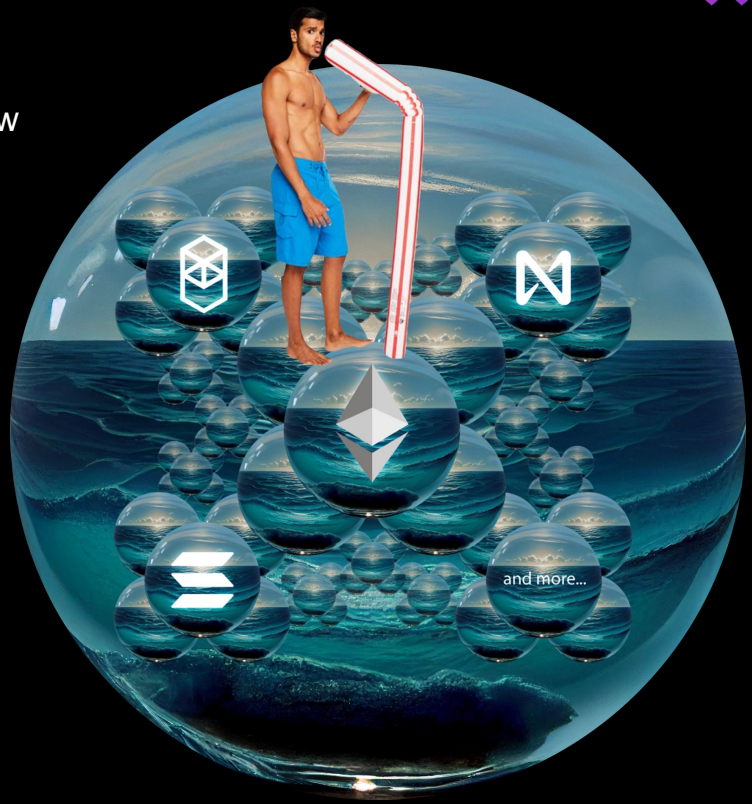
The Future is Multichain

... and the data volume keeps expanding



Get what you need when you need it!

Firehose with substreams is like a super powerful straw that can suck up all the information you want from blockchains and organize it the way you want

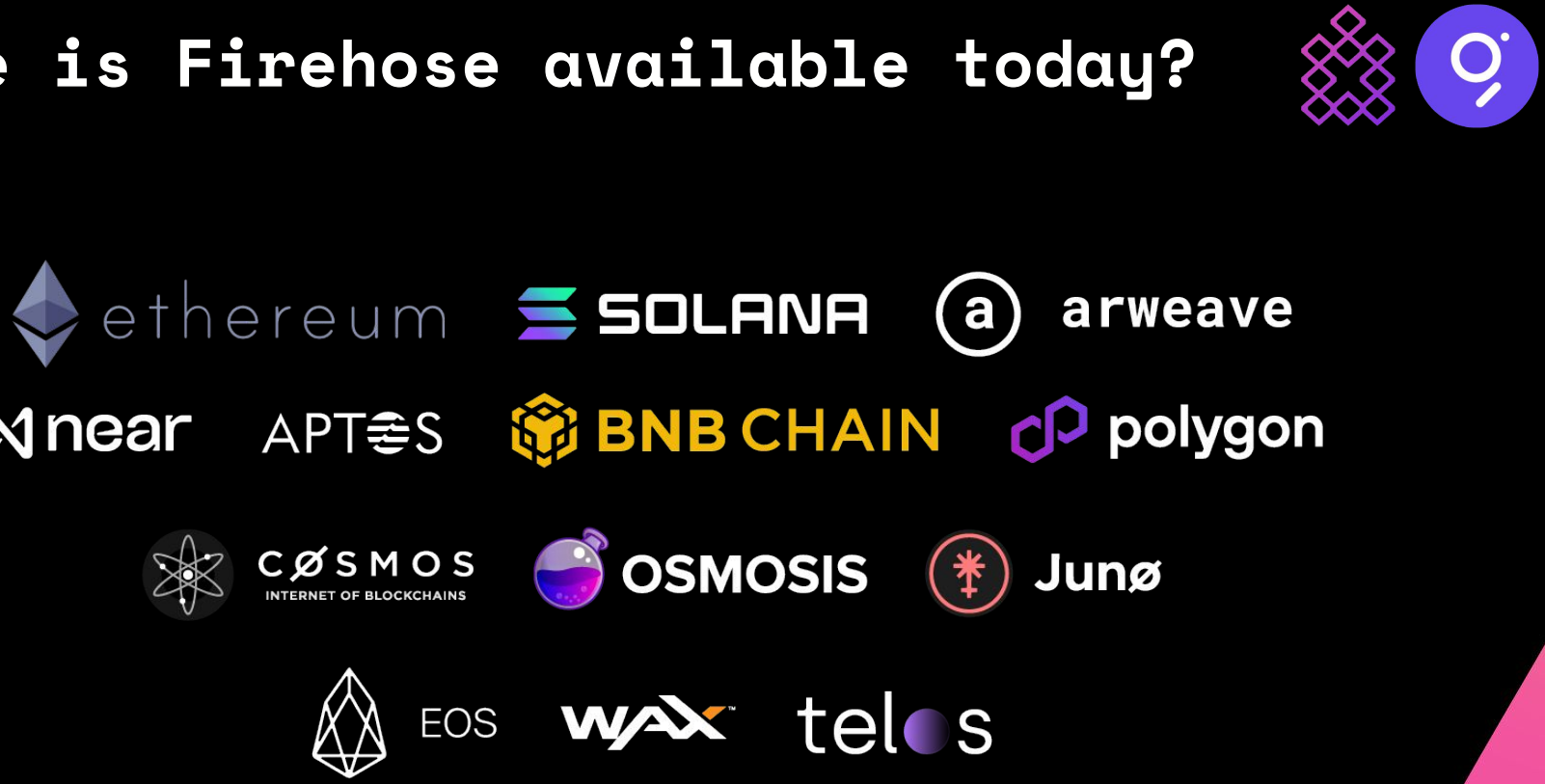


RUNNING BLOCKCHAIN INFRA IS HARD

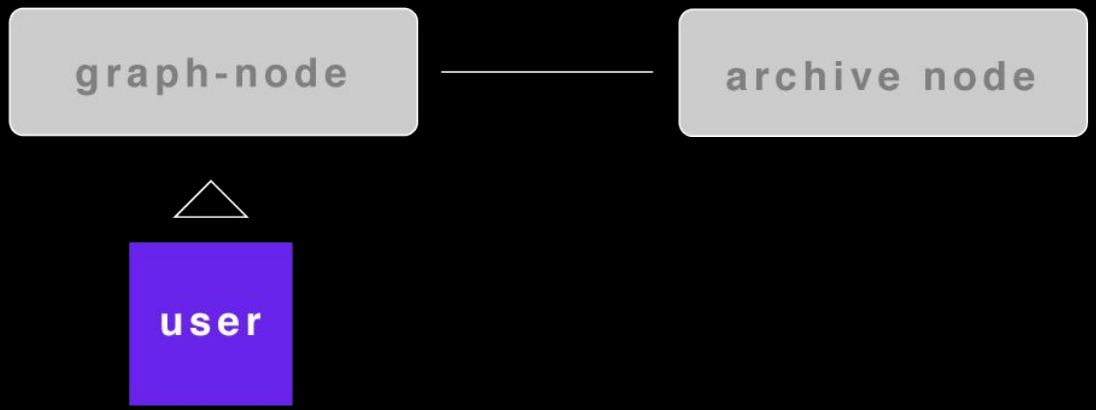
- Hard Forks / Upgrades
- Chains work in different ways
- Chains don’t have common APIs
- Scaling requires multiple clients

Firehose provides a scalable & stable interface!

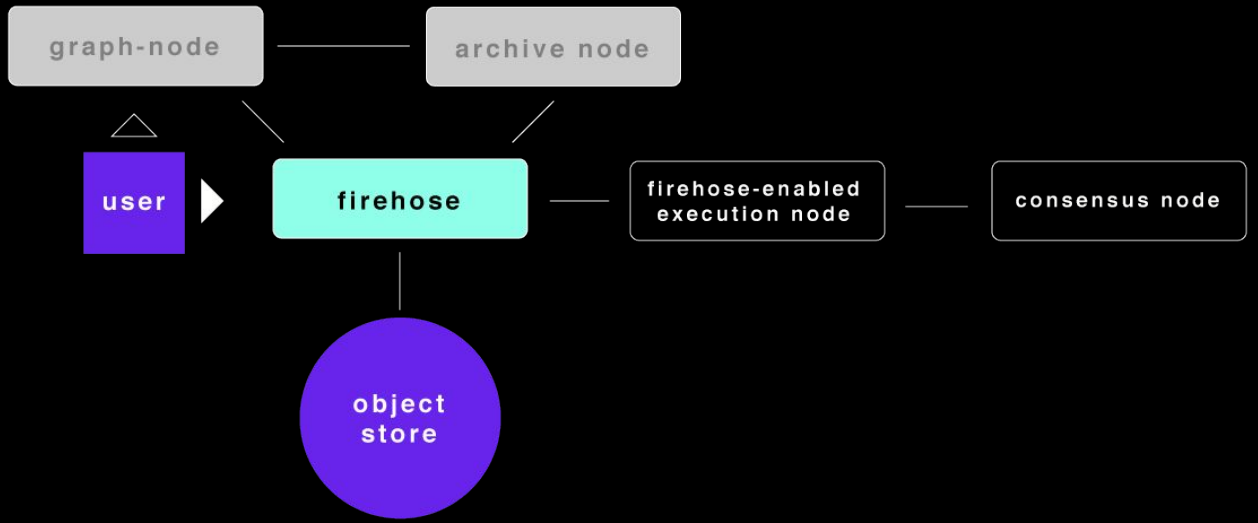
Where is Firehose available today?



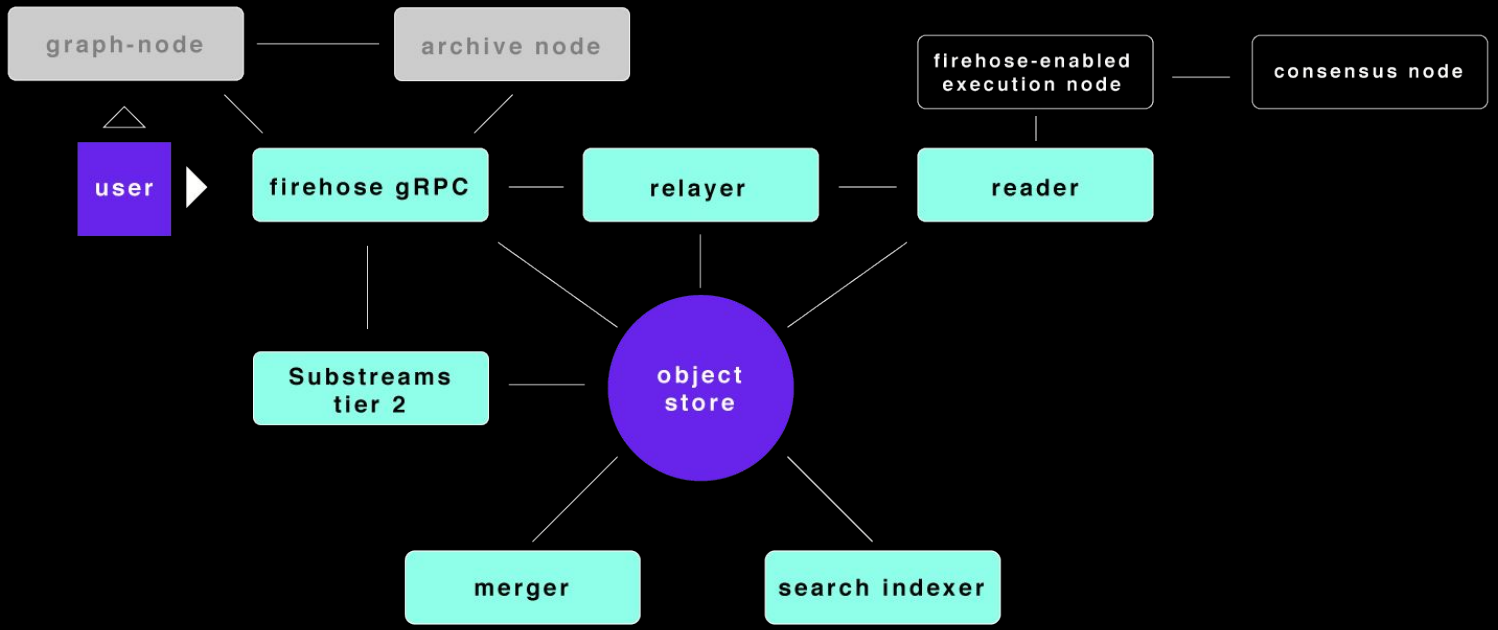
The Graph Architecture



Firehose Architecture



Firehose Architecture Details

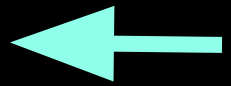


Firehose Data Format

Firehose stores data in “Flat Files”

```
Directory of:

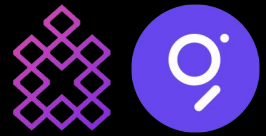
0000000000.dbin.zst
0000000100.dbin.zst
0000000200.dbin.zst
0000000300.dbin.zst
0000000400.dbin.zst
        ...
0016876900.dbin.zst
0016877000.dbin.zst
0016877100.dbin.zst
0016877200.dbin.zst
0016877300.dbin.zst
        ...
```



Merged Blocks

Up to 100 blockchain blocks in each file

What are Substreams?

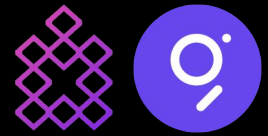


A way to get **specific** information from Firehose (rather than the whole block)

- **Composable**: Developers can built on top of each other's work
- **Parameterizable**: Get specific details from a more generic Substream
- **Transformable**: Define output format to fill the "**sink**" of your choice
- **Parallelizable**: Block data is processed in chunks and then results merged

Substreams are generally available since March 16, 2023
https://streamingfastio.medium.com/substreams-reach-general-availability-48272f6e942e

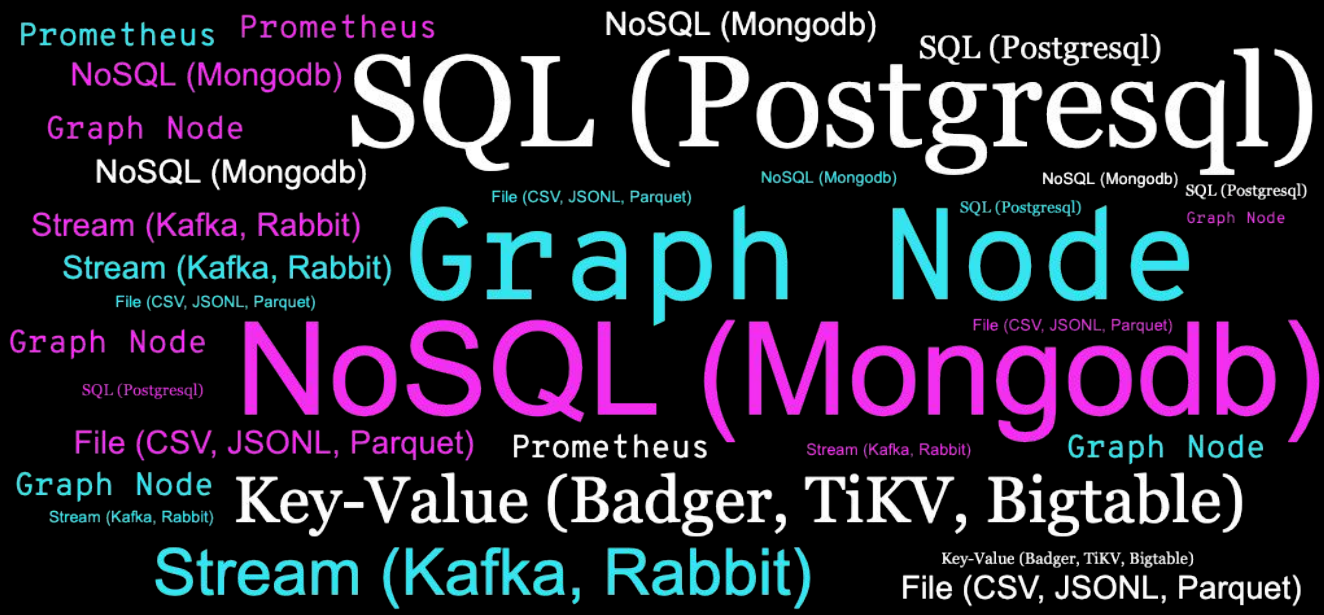
Firehose/Substreams Data Pipeline



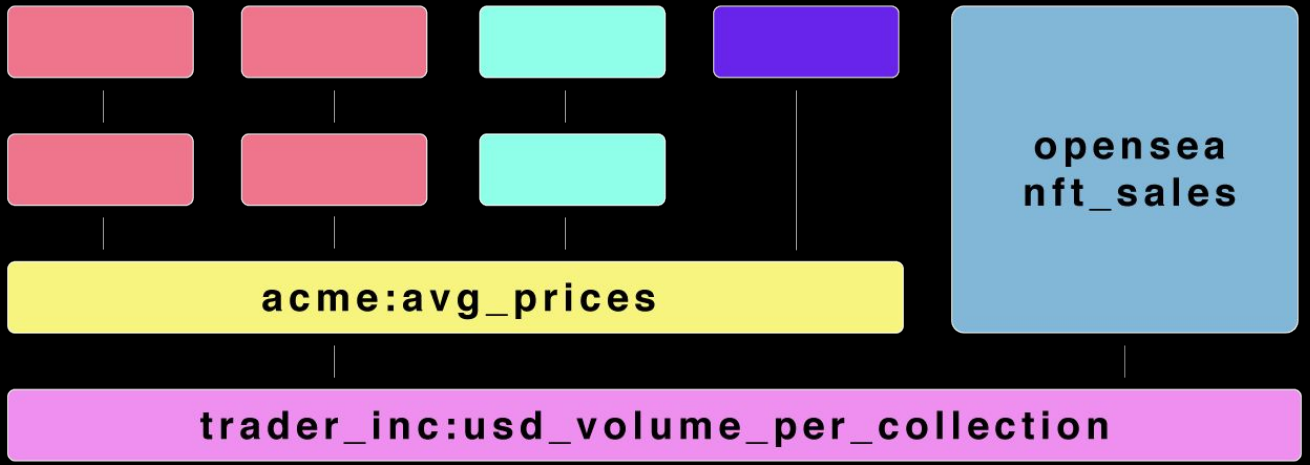
Extract → **Transform** → **Load** → **Query**

Firehose	Substreams	Substream Sink	Query Layer
Stream block data & create Flat Files	Stream transformed data & output cache files	Store the data however it makes sense for the application eg Graph Node	Query according to how it is stored eg Graph Node

Where to store all that Substream data?

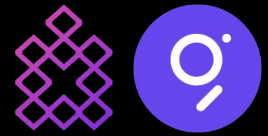


Substreams are composable!



*Different ***color*** == different ***author***

Substreams developer workflow



Write	Compile	Connect	Read	Action!
Write Code in Rust	Compile to WASM	Connect to Firehose grpc	Read results	Do something!
Leverage other people's work!	*Bundle into .spkg*	*Send the .spkg you want to run*	as they become available	*Eg store the results into a sink*

Substreams are performant!

8x faster

2 months → 1 week



Vincent @v_for_vincent · Oct 28, 2022

Just managed to process the entire history of Ethereum in 20 min using #substreams ⏫. LFG 🚀🚀🚀 @graphprotocol @streamingfastio

How did we do it?



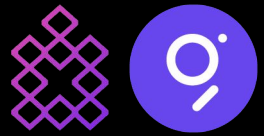
ISSUE, without Firehose

RPC / data access inconsistency between chains

SOLUTION

Protobuf gRPC stream

How did we do it?



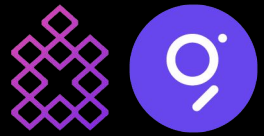
ISSUE, without Firehose

RPC not fast enough (eg Uniswap)

SOLUTION

Flat Files data streaming

How did we do it?



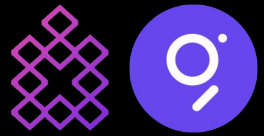
ISSUE, without Firehose

Developer time wasted waiting for results

SOLUTION

Server side parallel processing like Google BigQuery

How did we do it?



ISSUE, without Firehose

Reorg/Forks are hard

SOLUTION

Cursors allow reconnect with “undo”

How did we do it?



ISSUE, without Firehose

Fixed GraphQL API

SOLUTION

However you want it!

What it means:



FOR DAPPS

Substreams powered dApps



Simple:

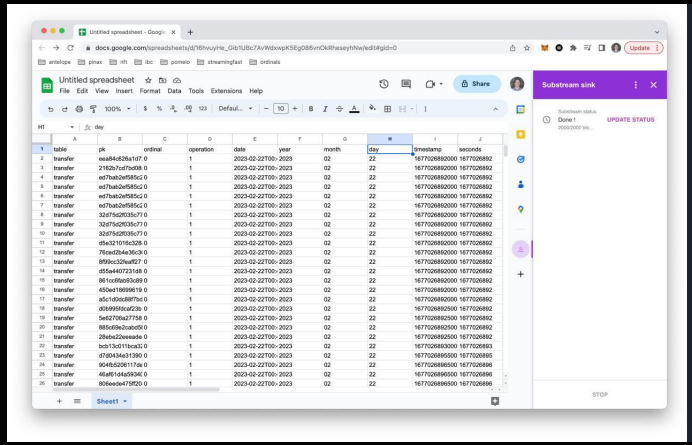
- Use Substreams in Subgraphs
- Stream data directly to a website in realtime
- Generate Telegram alerts
- Create operational dashboards in prometheus/grafana

Complex:

- Store whatever you need in a database and build custom applications on top

Substreams powered dApps: Today

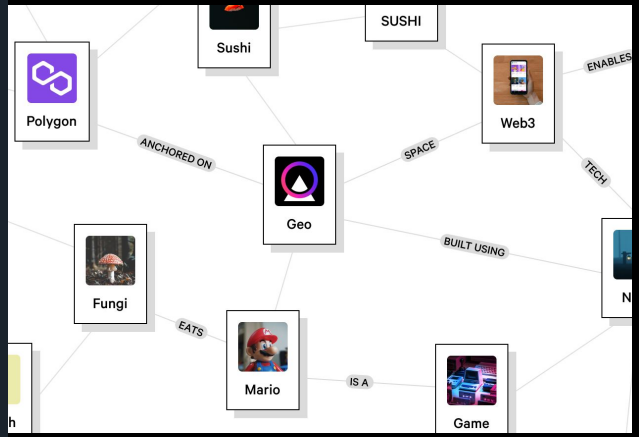
Basic applications for daily use



Google Sheets



ChatBots



GEO Browser

Substreams powered dApps: Future

Reliable Data Infrastructure is the backbone of Web3

- GameFi Economies
- Digital Identity
- DAO Reputation Systems
- AR and VR Fashion
- Web3 Social Media
- Tokenized Real Estate
- And ...



If everyone wants to own and access their data, that's a LOT of data.

What it means:



FOR DEVELOPERS OF ALL KINDS

For Developers



Smart Contracts:

Stop doing weird things to manage on-chain data!
And instead have proper tools to query data.

Data Analysis:

Pull all the data you need into a data warehouse

What do you want to do?

For Developers



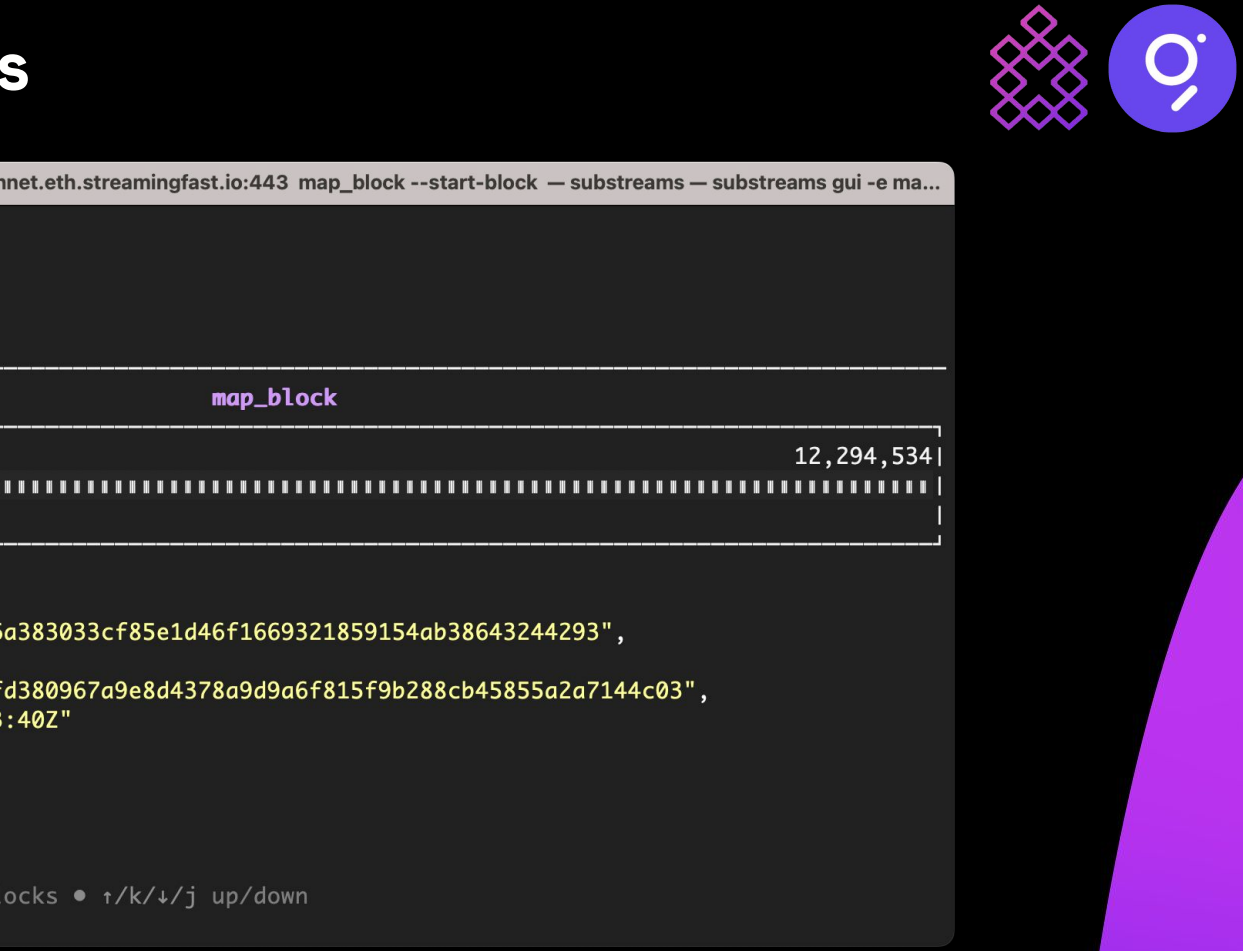
You need to learn a bit of Rust to create Substreams

But once you have your data feed, you can use whatever language you want to process it

- Javascript/Typescript
- Golang
- Python
- ...

And then do whatever you want!

For Developers



Results:



Substreams allow you to test and get feedback rapidly

Shorter Development Cycles → **Faster to Market** → **More Cool Stuff**

What it means:



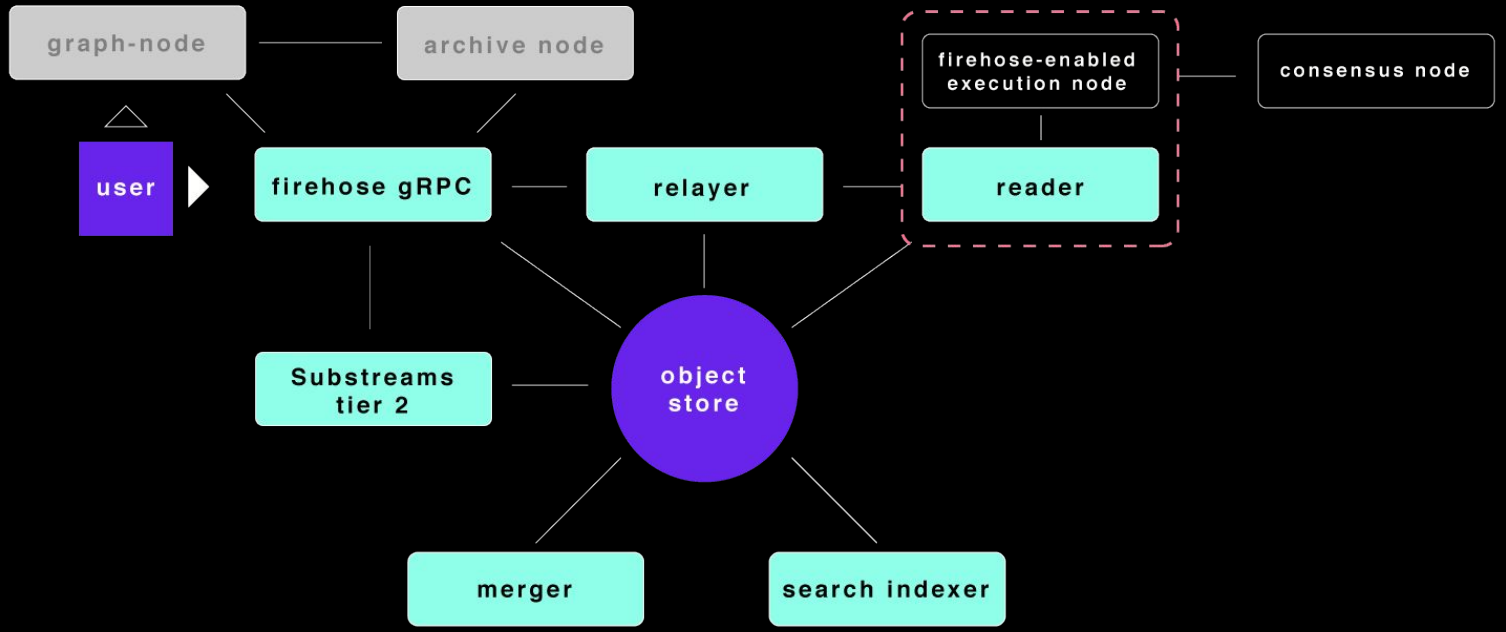
FOR BLOCKCHAINS

How to get Firehose support?



- Effort level really depends on the blockchain
- Integrations can use several different techniques
  - Poll the API (not ideal)
  - Use chain-provided SDK (if they have one)
  - Modify the software to generate LOG messages directly
- Need to create a reader that understands the specific blockchain stack
  - All Ethereum-like chains use the same reader
- Needs updating whenever there is a new release

Firehose Architecture Details



Firehose Everywhere...



Ideally we want the main software repo to integrate Firehose instrumentation so it is automatically available for every new release

Example request:
https://github.com/OffchainLabs/go-ethereum/issues/209

General process: https://firehose.streamingfast.io/integrate-new-chains/new-blockchains

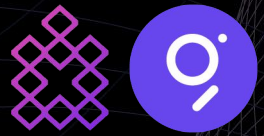
Can ask for free consult with **StreamingFast** team (ask Matt in **SF** discord)

What it means:



FOR INDEXERS

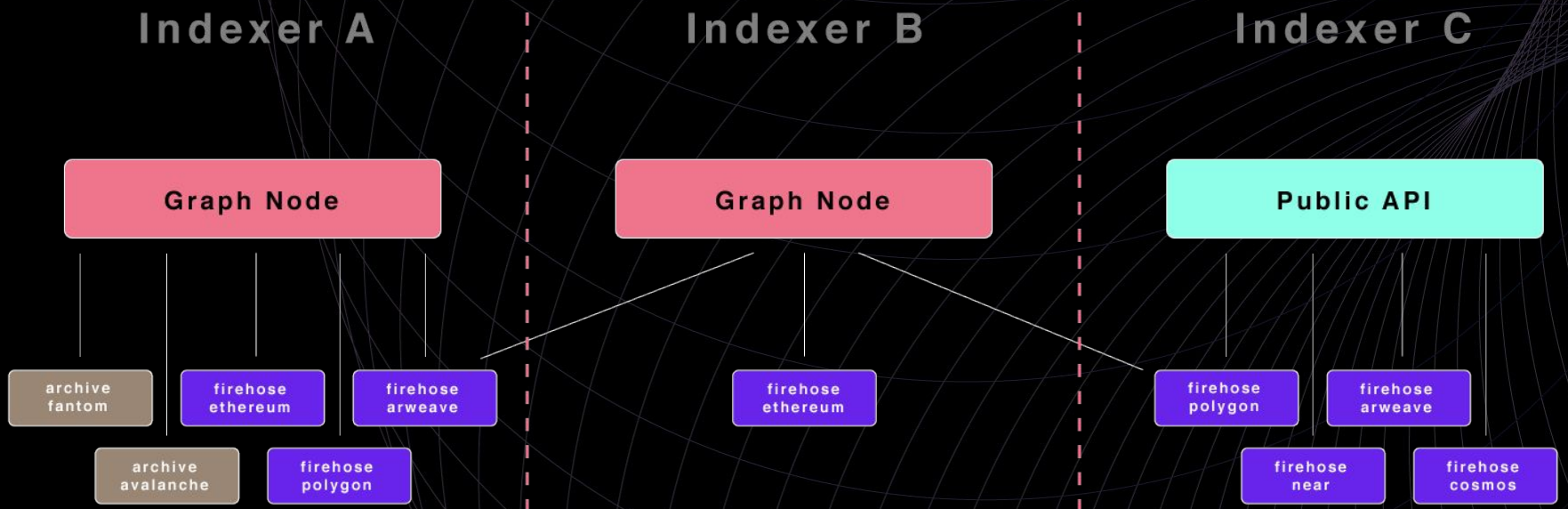
Shaping the Indexer Role



- Running blockchain infrastructure is hard
- Every Indexer running all the blockchains is not reasonable/profitable
- Firehose powered Indexers can leverage each other's infrastructure
  - Be an expert in some blockchains and run the full stack
  - Get Firehose data from another Indexer when you're not
- More Indexers = More Decentralization

Watch for more discussion on "**GIP-0042: A world of data services**"

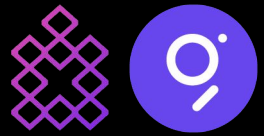
Shaping the Indexer Role



Future Indexers



Summary:

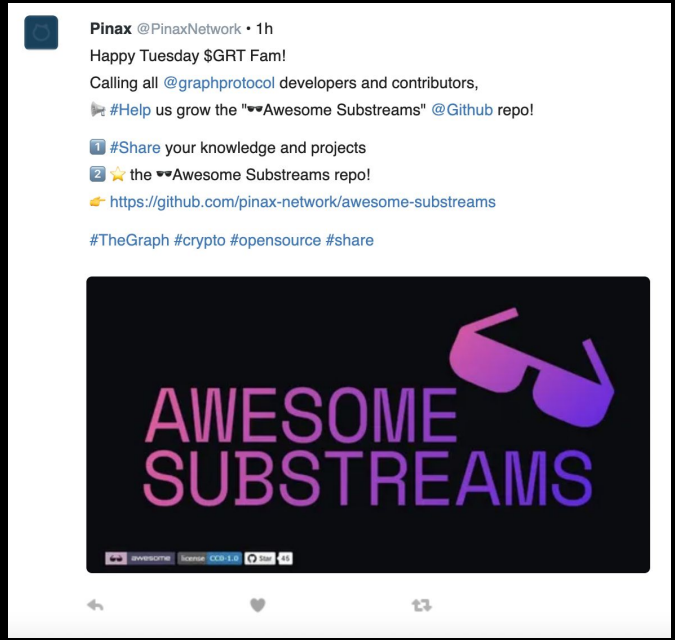


- ★ Decouple reads from writes
- ★ One Data Model for all blockchains, avoiding inconsistencies
- ★ Data Completeness - Extracting **ALL** data into flat files
  - ○ More granularity of data including state changes
  - ○ Single source of truth of all data needed for any API
  - ○ No more JSON-RPC calls
- ★ Parallel Processing on the server side to scale
- ★ Powering next generation Subgraphs = Substreams
- ★ Open Source

Learn more with Pinax

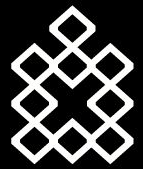
Dig deeper here:





Do you have

QUESTIONS?



PINAX

Thank You