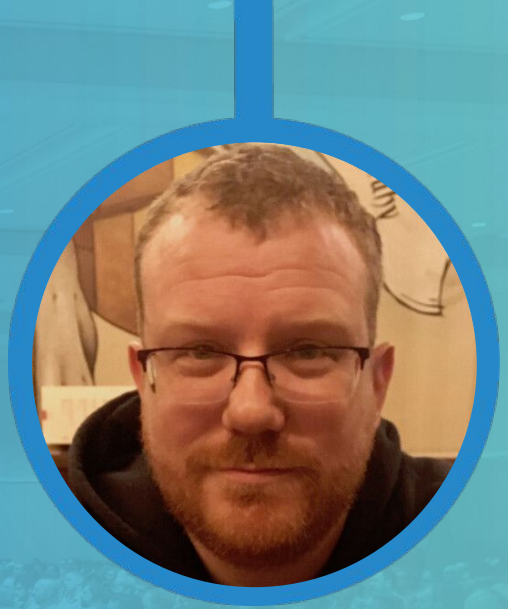
HYPERLEDGER FOUNDATION IN DEPTH:

Splunk

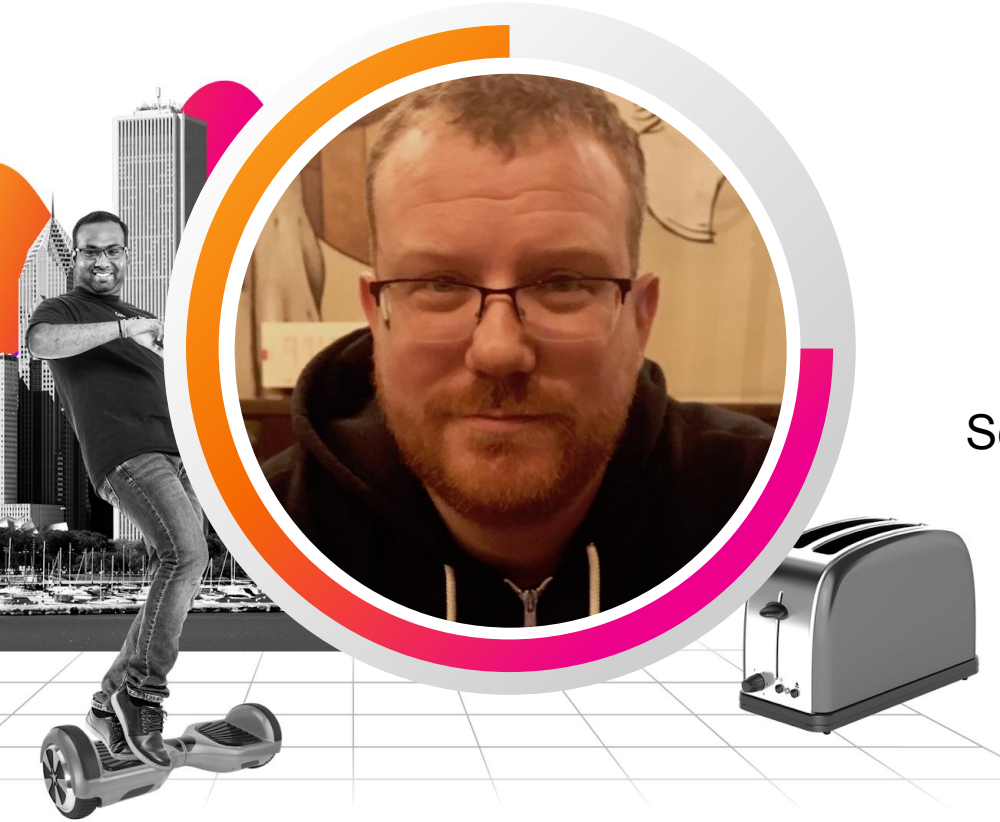
Observability and blockchain - a deep dive into Fabric and Besu



Antoine Toulme
Senior Engineering Manager
Splunk

MEMBER

Wednesday, August 3rd, 2022
7:00am PT | 2:00pm GMT | 10:00pm HKT, SGT



Antoine Toulme

Senior Engineering Manager | Splunk

splunk> .conf22

We will cover

Observability

Signals

OpenTelemetry

Blockchain

Multi-party computation

Distributed systems

Observability primer

Applications for blockchains

A peek into the future

Observability

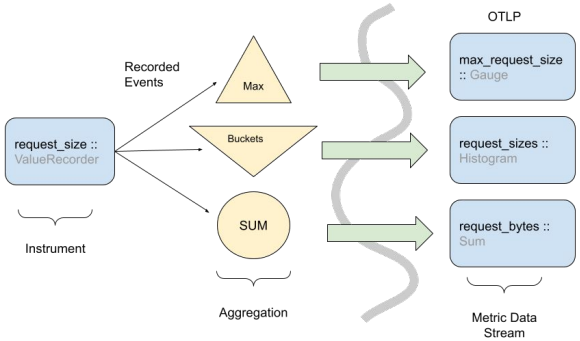
- Referred to as o11y

  Observability lets us understand a system from the outside, by letting us ask questions about that system without knowing its inner workings.

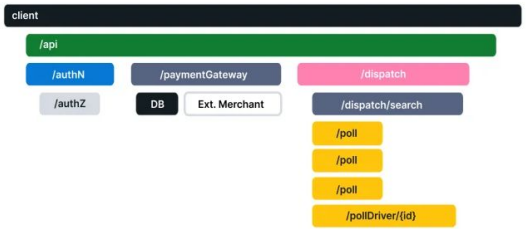
- A common interoperable standard to monitor software
  - Common way to instrument the software
  - Common data model for all data emitted by the software
  - Common naming of conventions and semantics

Signals

Metrics



Traces



Logs

Metrics

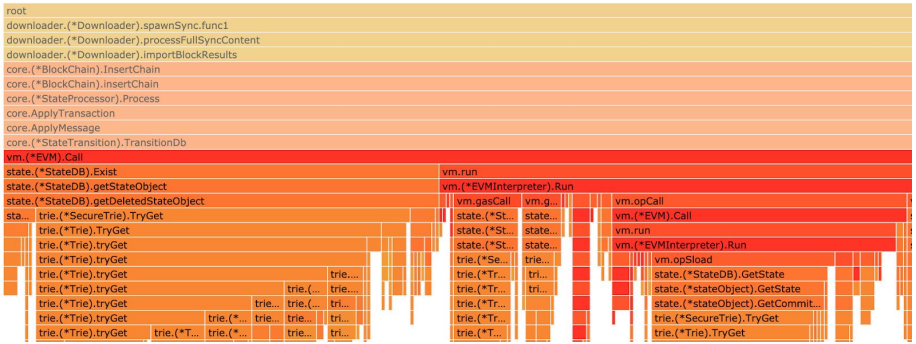
Stable model https://opentelemetry.io/docs/reference/specification/metrics/datamodel/

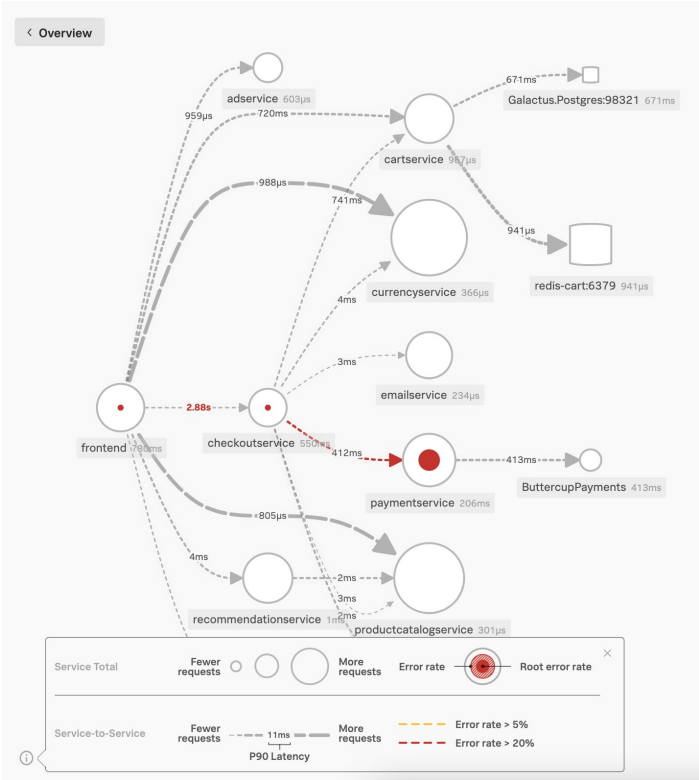
1. Sum
2. Gauge
3. Histogram
4. Exponential Histogram

Traces

- Notion of a span - an individual measurement of performance
- Attributes as metadata
  - Span kind
  - Service information (name, version)
- Span parent and links

Traces


root
downloader.(*Downloader).spawnSync.func1
downloader.(*Downloader).processFullSyncContent
downloader.(*Downloader).importBlockResults
core.(*BlockChain).InsertChain
core.(*BlockChain).insertChain
core.(*StateProcessor).Process
core.ApplyTransaction
core.ApplyMessage
core.(*StateTransition).TransitionDb
vm.(*EVM).Call
state.(*StateDB).Exist
state.(*StateDB).getStateObject
state.(*StateDB).getDeletedStateObject
trie.(*SecureTrie).TryGet
trie.(*Trie).TryGet
vm.run
vm.(*EVMInterpreter).Run
vm.gasCall
vm.opCall
vm.(*EVM).Call
vm.run
vm.(*EVMInterpreter).Run
vm.opSload
state.(*StateDB).GetState
state.(*stateObject).GetState
state.(*stateObject).GetCommit...
trie.(*SecureTrie).TryGet
trie.(*Trie).TryGet


Overview
adservice 603µs
959µs
720ms
671ms
Galactus.Postgres:98321 671ms
cartservice 967µs
988µs
741ms
941µs
redis-cart:6379 941µs
currencyservice 366µs
4ms
3ms
emailservice 234µs
2.88s
frontend 790ms
checkoutservice 550ms
412ms
413ms
ButtercupPayments 413ms
paymentservice 206ms
805µs
4ms
2ms
3ms
2ms
recommendationservice 1ms
productcatalogservice 301µs
Service Total
Fewer requests
More requests
Error rate
Root error rate
Service-to-Service
Fewer requests
11ms
P90 Latency
More requests
Error rate > 5%
Error rate > 20%

Logs

[this should never happen]

```
if(temp2.length === 0 || temp2.length > 1) {
            console.error('Duplicate lists found. This should never happen.');
            res.status(400).send('Duplicate lists found. This should never
happen.');
            return;
      }
```

New directions for use:

- Metricizing logs
- Associating logs to traces

```
double downTimeSoFar = 0.0;
downTimeSoFar = downTimeList.get(indexforDownTime);
System.out.println("So far migration downTime for clet "+cletID+ " is "
+downTimeSoFar);
downTimeSoFar = downTimeSoFar + downTime;
System.out.println("New so far migration downTime for clet "+cletID+ " is "
+downTimeSoFar);
```

The web if everyone standardizes on o11y

OpenTelemetry and Our Needed Data

	Tracing	Metrics	Logs, etc
Instrumentation APIs			
Canonical implementations			
Data infrastructure			
Interop formats			

How it works

Specification:

https://opentelemetry.io/docs/reference/specification/

SDK configuration:
https://github.com/open-telemetry/opentelemetry-specification/blob/main/specification/sdk-environment-variables.md

How it works

OTEL COLLECTOR

Observability with blockchain

Nicola Cocchiaro, Infura, December 8th 2020:

> This is where the broader sense of observability comes in. It is the ability to look at characteristics of a system that individually may offer only partial information, but when taken together in context may help explain why something is happening. Even when considering only monitoring, ultimately that “why” is still the question we want to answer. It's easy to see how this definition now starts to move away from active monitoring and toward the realm of debugging, of catching failures before they occur by seeing that something is trending in a bad direction, and generally of answering all those questions that we don't know yet we should be asking, but that we will be *urgently* asking when something goes wrong.
>
> https://blog.infura.io/post/observability-for-developers-infura

Operate and Extend Hyperledger Besu

Hyperledger Foundation workshop

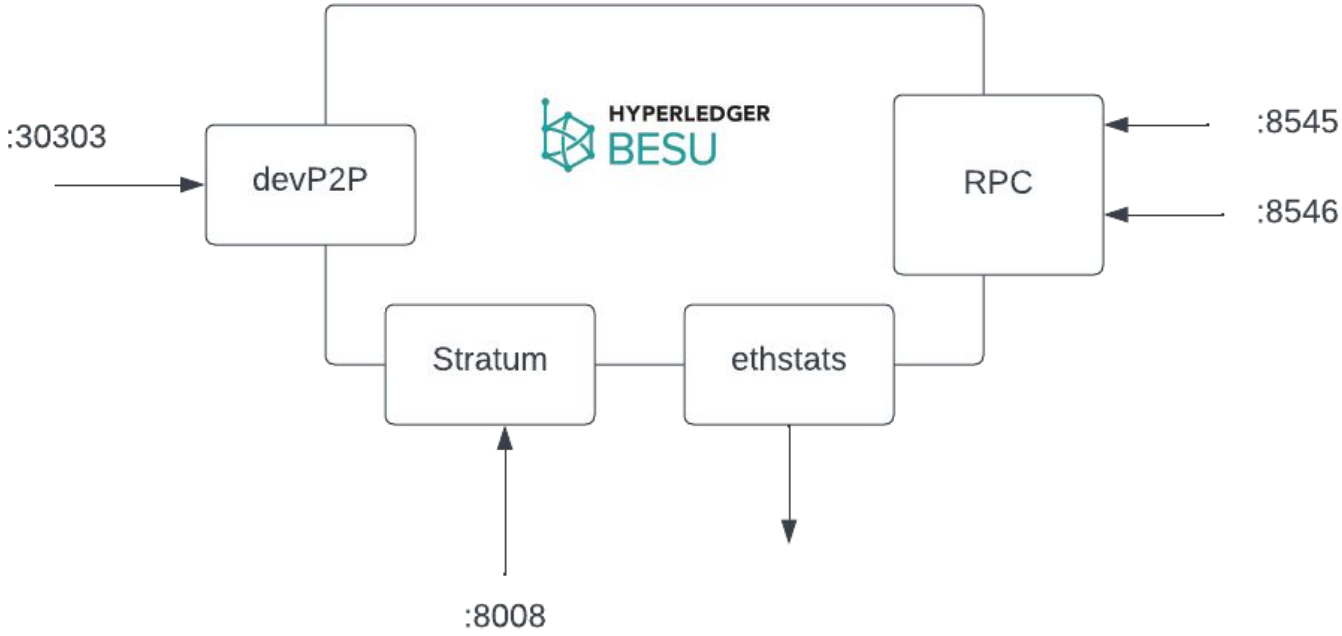
July 14th, 2022

HYPERLEDGER
BLOCKCHAIN TECHNOLOGIES FOR BUSINESS

Ethereum client - high level

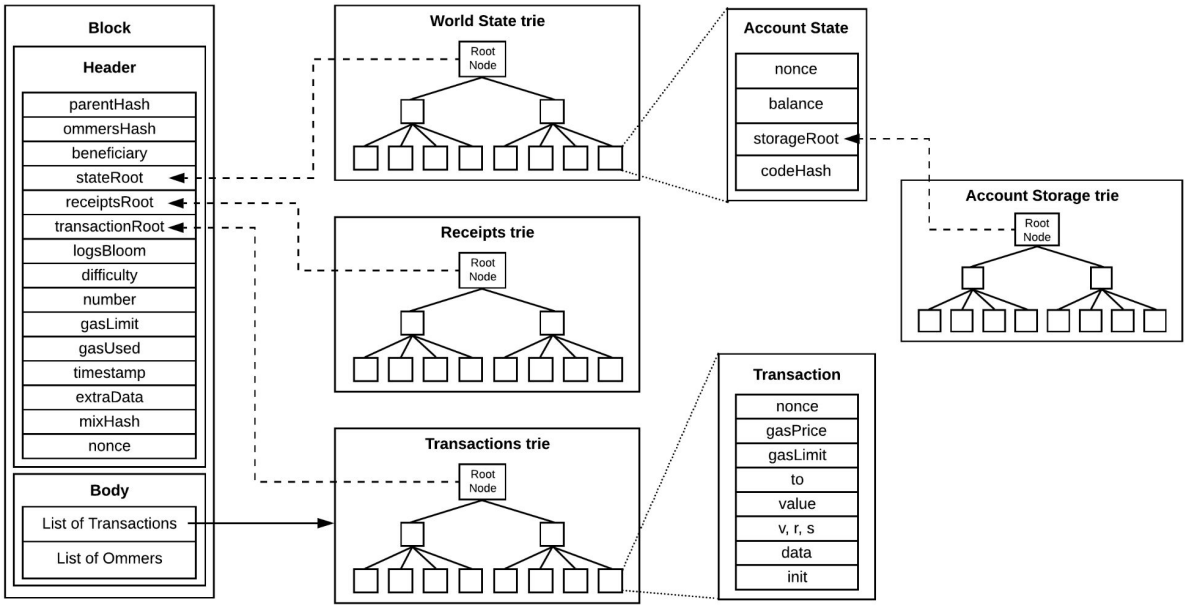
- Misnomer for a server, a peer-to-peer agent
- Runs as a single process
- Independent
  - Can perform all exchanges
  - Can submit transactions
  - Can interrogate the chain

Complex software stack



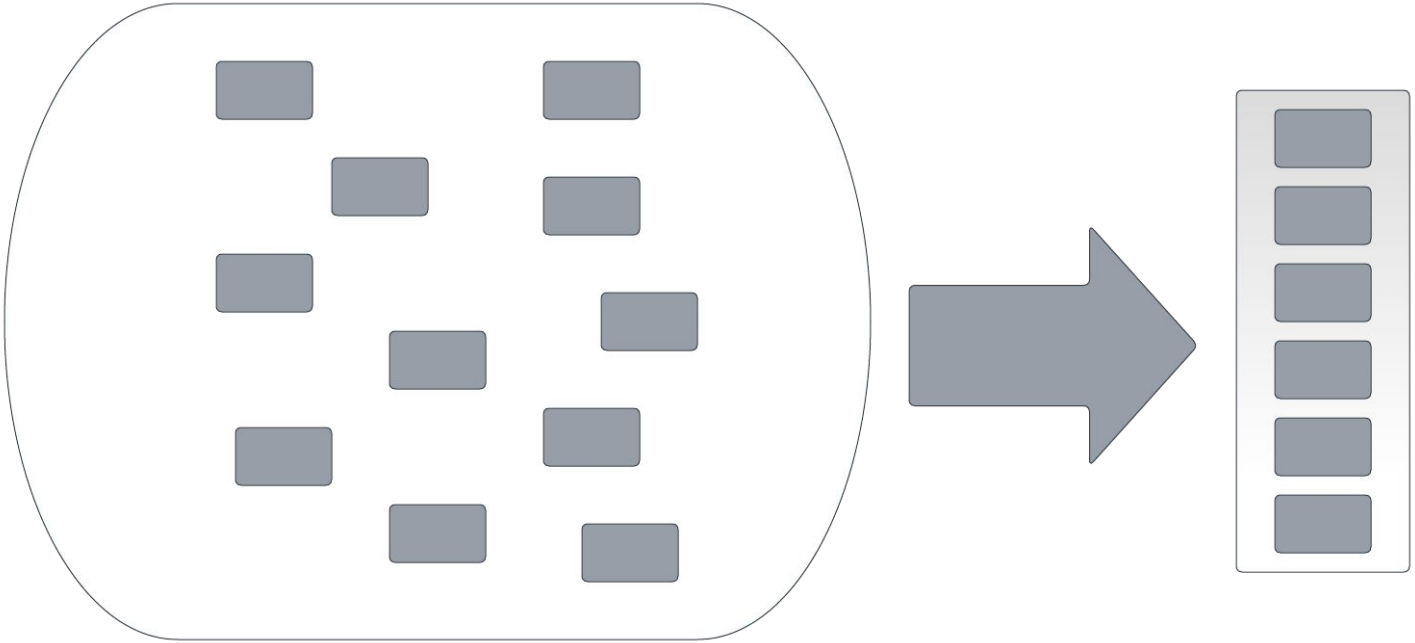
Besu as a database

- Using RocksDB storage
- Multiple stores
- Here is why:



Source: https://www.lucassaldanha.com/ethereum-yellow-paper-walkthrough-2/

Besu as a transaction pool



Besu network for Ethereum

- Each client is completely independent, so it requires configuration.
  - A genesis block
  - A consensus engine
  - Bootnodes to discover other peers

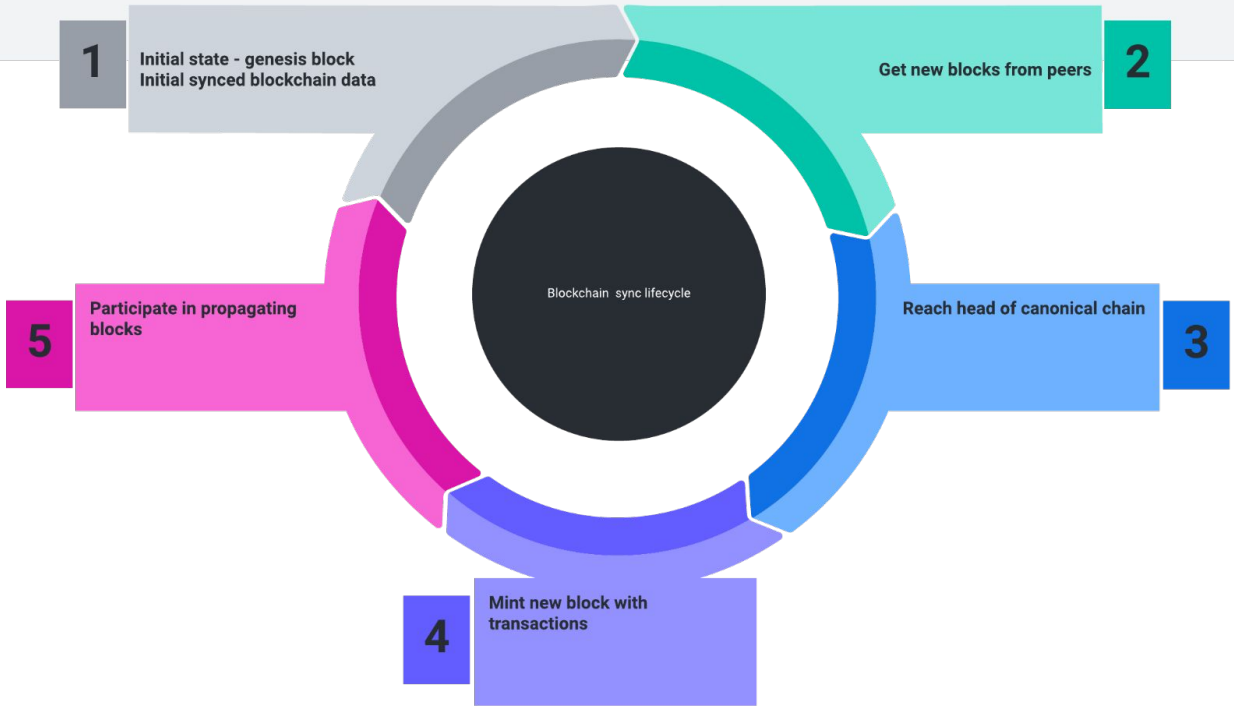
Besu discovery

- Connect to other nodes using UDP-based messages
  - First to bootnodes, then all peers exposed by them
- Store peers into buckets to avoid eclipse attacks
  - Use a Kademlia hashtable
- New discovery mechanism using DNS
  - Indexing from a bootnode on a regular basis
  - Easy to download and check integrity
- Static peering
  - Set enodes as part of configuration
  - `enode://6f8a80d14311c39f35f516fa664deaaaa13e85b2f7493f37f6144d86991ec012937307647bd3b9a82abe2974e1407241d54947bbb39763a4cac9f77166ad92a0@10.3.58.6:30303?discport=30301`

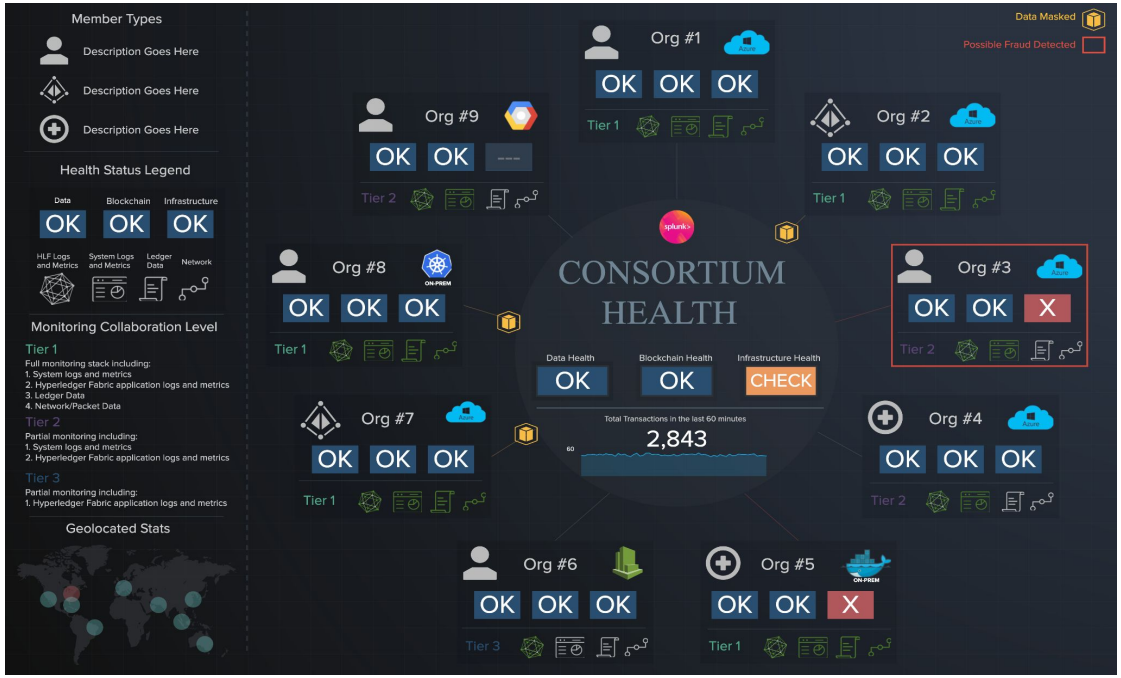
Besu network client

- Using devp2p, embedding node identity
  - Send HELLO message to other nodes
  - Negotiates subprotocols such as eth (others exist such as Whisper, or IBFT)

Hyperledger Besu lifecycle



Besu as part of consensus



- Clique
- IBFT
- Ethash (PoW)
- PoS (the merge)

JSON-RPC server

```
{
    "jsonrpc":"2.0",
    "method":"web3_clientVersion",
    "params":[],
    "id":1
}
```

HTTP

- Supports batching
- Used by wallets such as Metamask

WS

- Web socket
- Great for subscriptions
- New events and logs

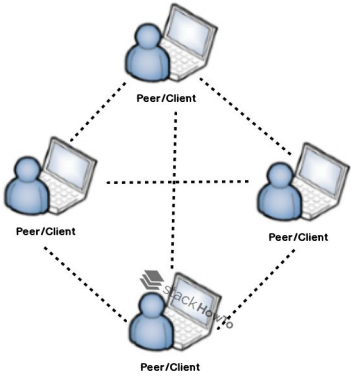
IPC

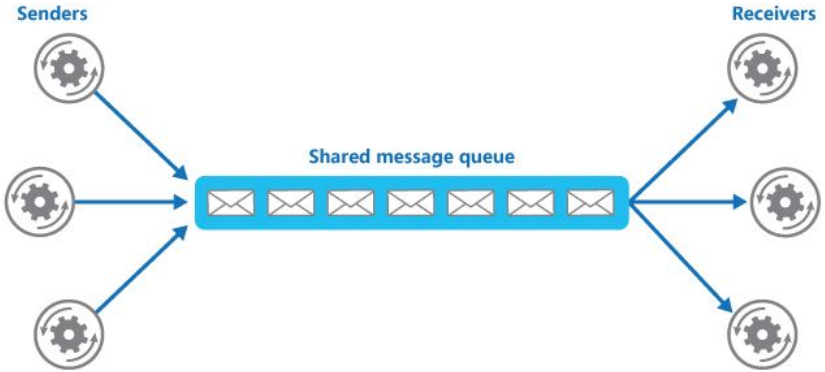
- Using a file socket
- Can be used to attach to the client with geth
- Most secure option
- Just added to Besu in April!

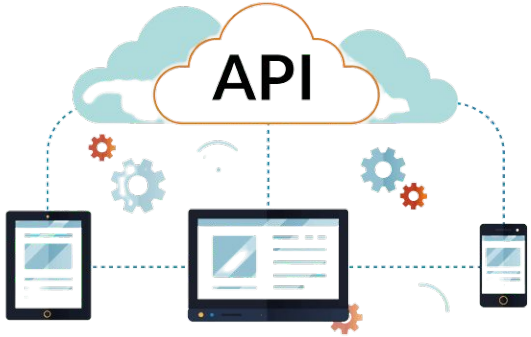
GraphQL

- Versatile API
- Allows to query specific data from the chain

In recap


Peer/Client
Peer/Client
Peer/Client
Peer/Client


Senders
Receivers
Shared message queue


API

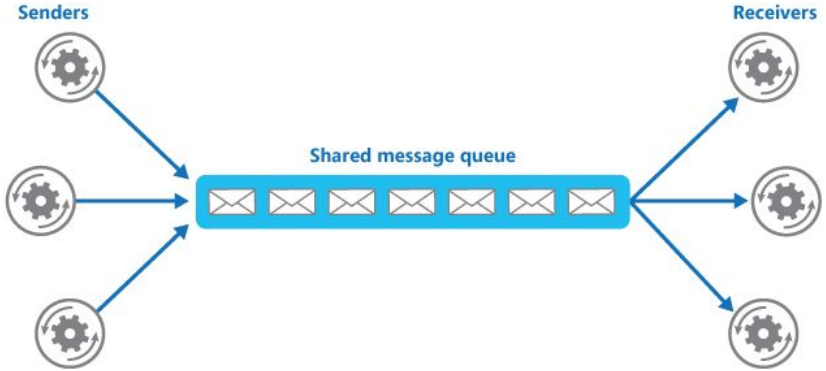
Database

- RocksDB metrics
- Compaction and garbage collection metrics
- Writer and indexer tracing
- Merkle tree and Bonsai tree complex debugging

API

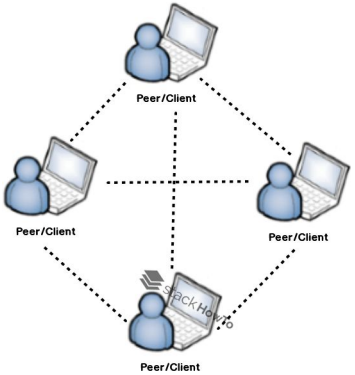
- HTTP headers
  - W3C, Jaeger, B3
- Capture state and errors
  - What is the average latency of a JSON-RPC roundtrip to get a block by number?

Message queue



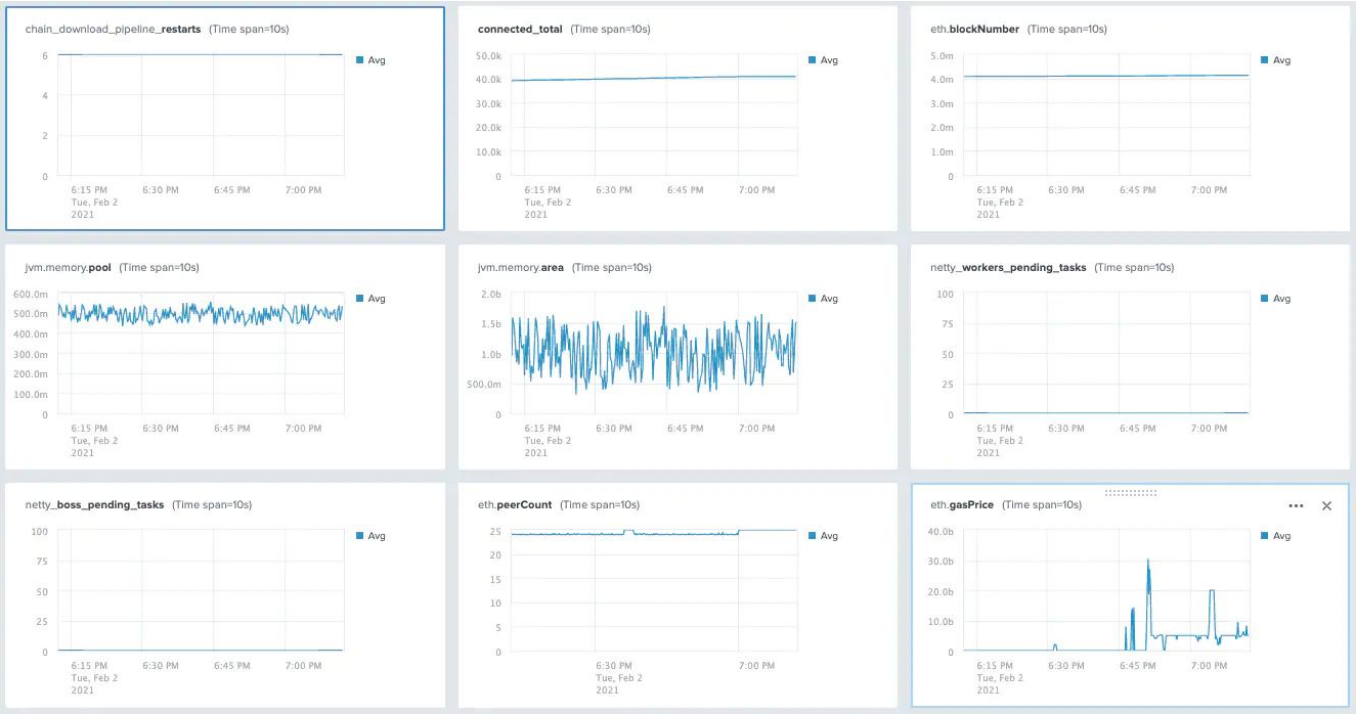
- Transaction pool debugging
  - Metrics
  - Tracing a transaction by its id through the pool to a block
- Mining
  - Solution metrics
  - Active miners
- Syncing

Peer-to-peer

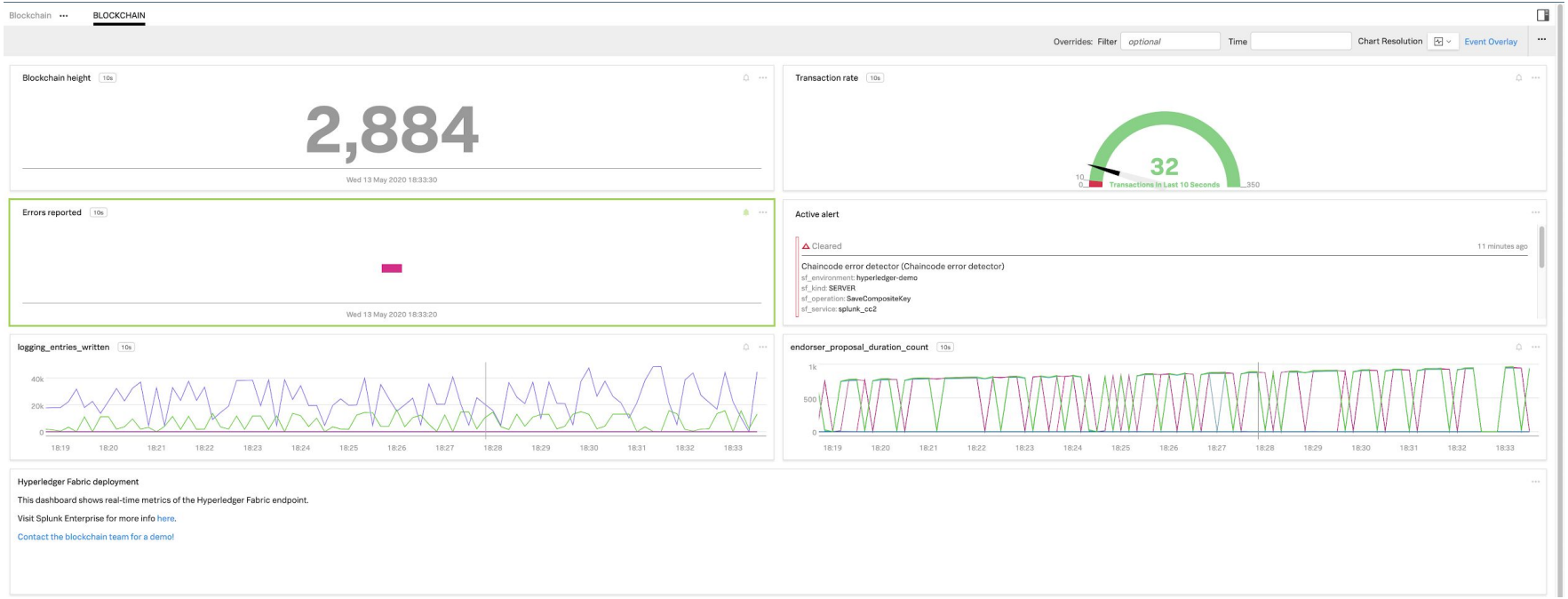


- Discovery
  - Peers
  - Connections and disconnections
- Tracing devp2p messages
- Future of tracing across clients

Besu metrics



Fabric metrics



Community

Fabric RFC:
https://github.com/hyperledger/fabric-rfcs/blob/main/text/0000-opentelemetry-tracing.md

PR for Fabric Java chaincode:
https://github.com/hyperledger/fabric-chaincode-java/pull/200/files

PR to add grpc interceptors for Fabric:
https://github.com/hyperledger/fabric/pull/3502

Community - Besu

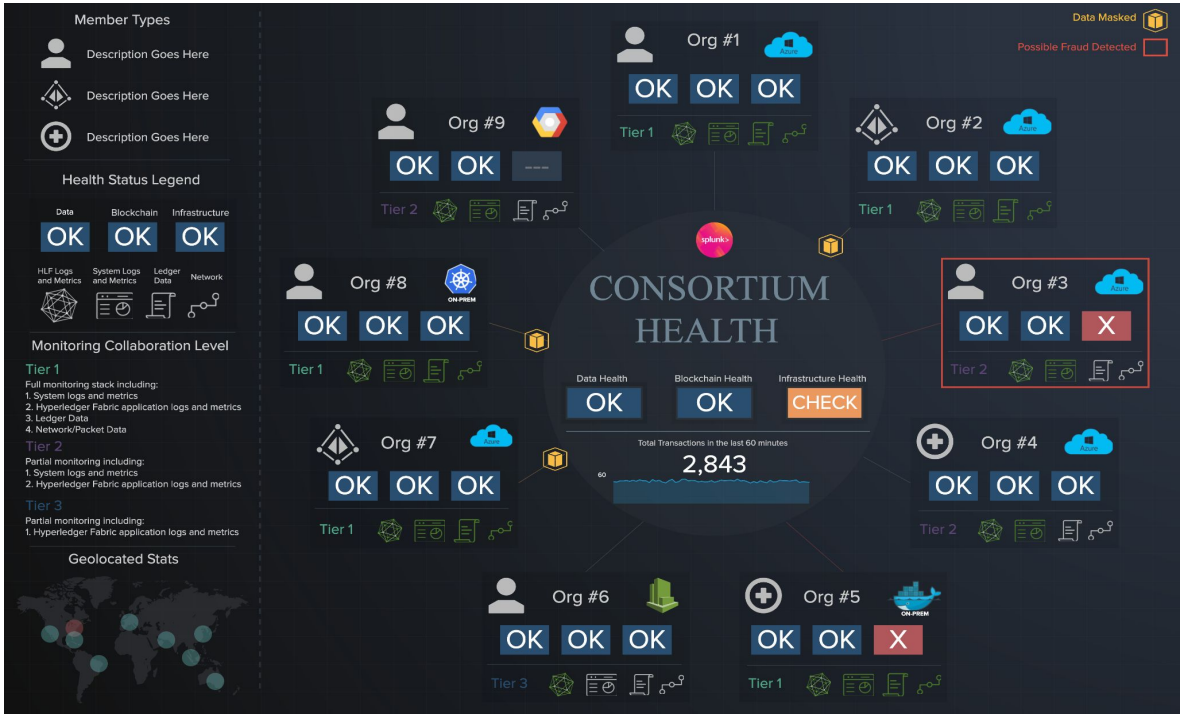
Use OpenTelemetry - Hyperledger Besu

https://github.com/hyperledger/besu/blob/main/metrics/core/src/main/java/org/hyperledger/besu/metrics/opentelemetry/OpenTelemetrySystem.java

https://github.com/hyperledger/besu/blob/main/ethereum/api/src/main/java/org/hyperledger/besu/ethereum/api/jsonrpc/JsonRpcService.java#L473

https://github.com/hyperledger/besu/blob/main/services/pipeline/src/main/java/org/hyperledger/besu/services/pipeline/Pipeline.java#L141

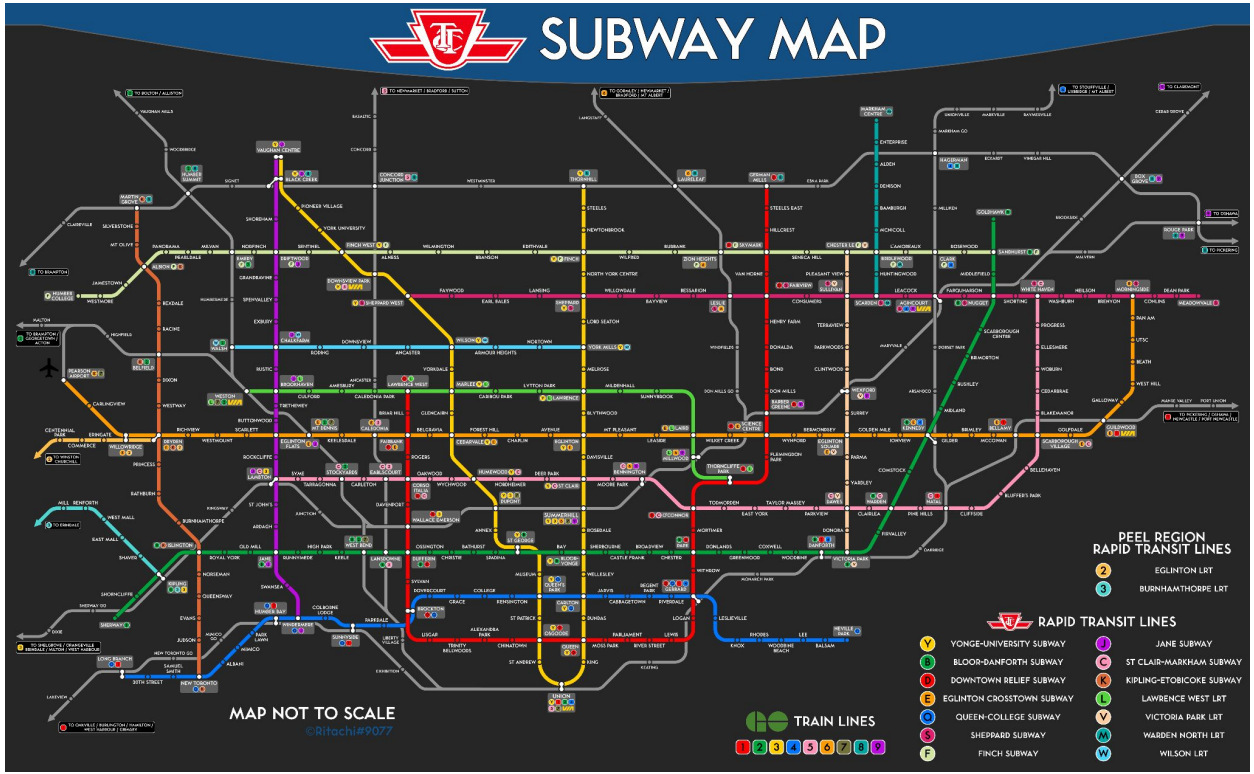
Trust and no trust scenarios



The future

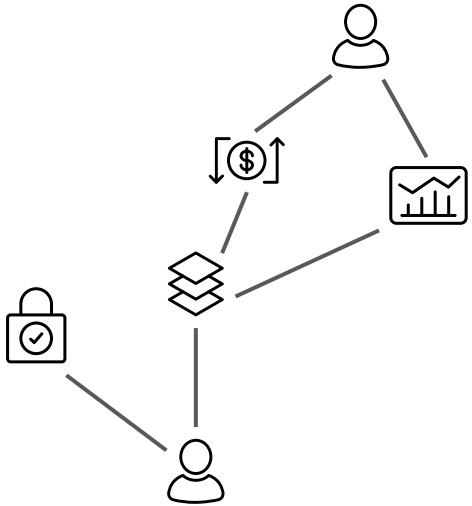
https://www.deviantart.com/exitmothership/art/Commission-Future-Metropolis-801953824

Trace Central



https://www.deviantart.com/ritachi9077/art/Updated-TTC-Subway-LRT-Map-Fantasy-857576854

Free with a data subscription on the side



Trust providers

- Sign your trace
- Check its execution

https://twitter.com/gakonst/status/1553881575745413120

Further reading

- Infura on observability
  How to Gain Visibility, Track Costs, and Build Transparency with Observability
- Besu is an observability pioneer
  Hyperledger Besu is an Observability Pioneer | Splunk
- Besu workshop
  Operate and Extend Hyperledger Besu Workshop


Q&A


Thank you!