Operate and Extend Hyperledger Besu

Hyperledger Foundation workshop

March 21st, 2023

HYPERLEDGER
BLOCKCHAIN TECHNOLOGIES FOR BUSINESS

Welcome!

https://wiki.hyperledger.org/display/events/Operate+and+Extend+Hyperledger+Besu+Workshop

Operation session (8AM to 10AM Pacific)

- 30 minutes intro to Besu - what it does, how it works
- 30 minutes on Besu configuration - environment variables, toml file, cli flags, hidden flags. Docker vs binary installation.
- 60 minutes on running a Besu network
    - 15 minutes on Besu in dev mode, curl, easy first steps.
    - 15 minutes on Genesis block generation
    - 30 minutes on consortium creation, with monitoring and health report, with a Docker compose.

Developer session (10AM to Noon Pacific)

- 20 minutes on Besu and Github - source (3 repos), issues, PRs, code checkout, CI.
- 20 minutes on setting up with an IDE. Run and compile with Gradle. Checks and validations in place with spotless and errorprone.
- 20 minutes on the main repository content, describing types of tests in place, showing the tree of dependencies between Gradle modules.
- 30 minutes on adding an opcode to the EVM: show how to add a new opcode, add to the next hard fork revision, how to test it.
- 30 minutes on how to add a new JSON-RPC method, add tests and docs.

Prerequisites

https://wiki.hyperledger.org/display/events/Operate+and+extend+Hyperledger+Besu+Workshop+Prerequisites

Install:

- Java
- Git
- Docker
- Docker-Compose
- Your favorite IDE

Download:

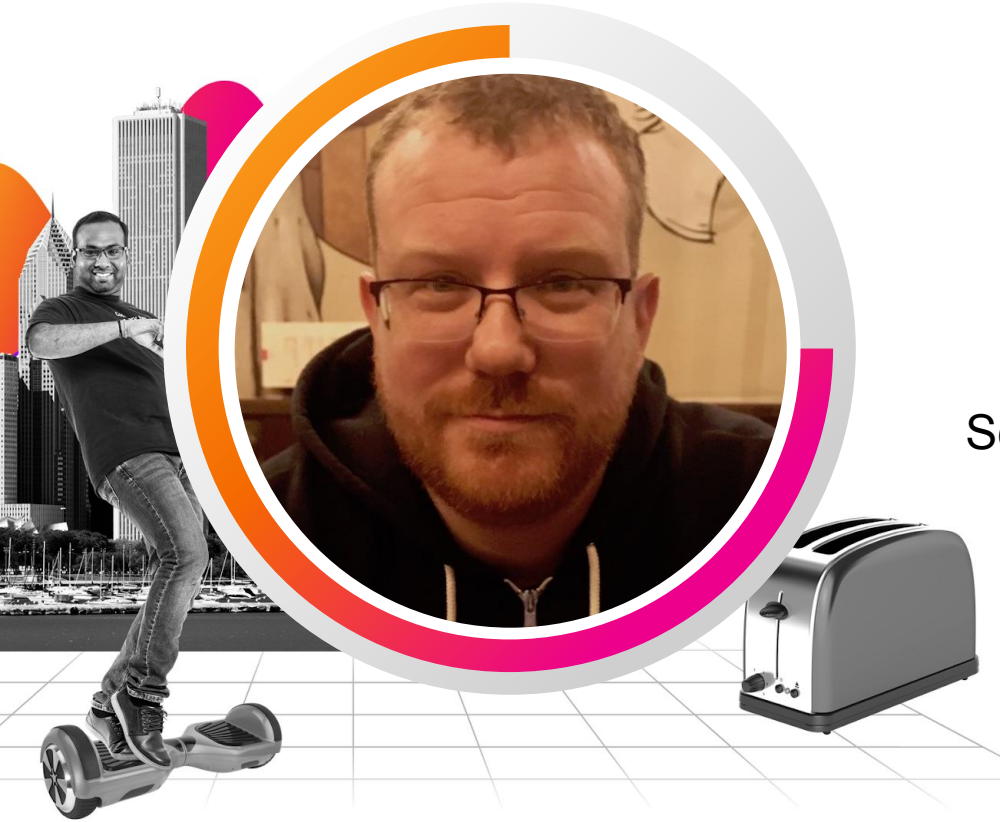
- Besu sources
- Quorum Dev Quickstart

Besu, Public Networks, & The Merge - What's Next

What do you want to learn about the future of Besu, staking, & Ethereum public networks?

Help us shape Besu and determine the topics for the next rounds of workshops by filling out this quick survey (only 7 questions)!

Matt Nelson, Besu Product Manager, ConsenSys Protocols



Antoine Toulme

Senior Engineering Manager | Splunk

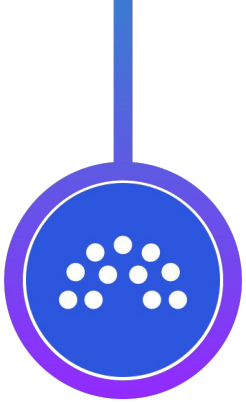
splunk> .conf22

Ethereum

- Second largest crypto by market capitalization
- Started in 2014
- Many different clients
- Instead of a single application, a programmable layer that executes smart contracts
  - EVM

The Enterprise

- As in - existing businesses
- Well supported client
- Different approach to deployments
  - Consensus is different
  - Everything permissioned
- Security!
  - Data management
  - Audits



Quorum

First take on an enterprise Ethereum client

Built by JPMC, eventually owned by ConsenSys

Using a private enclave to host data, private transactions and state

Using new consortium consensus algorithms: RAFT, IBFT

Using a fork of Geth

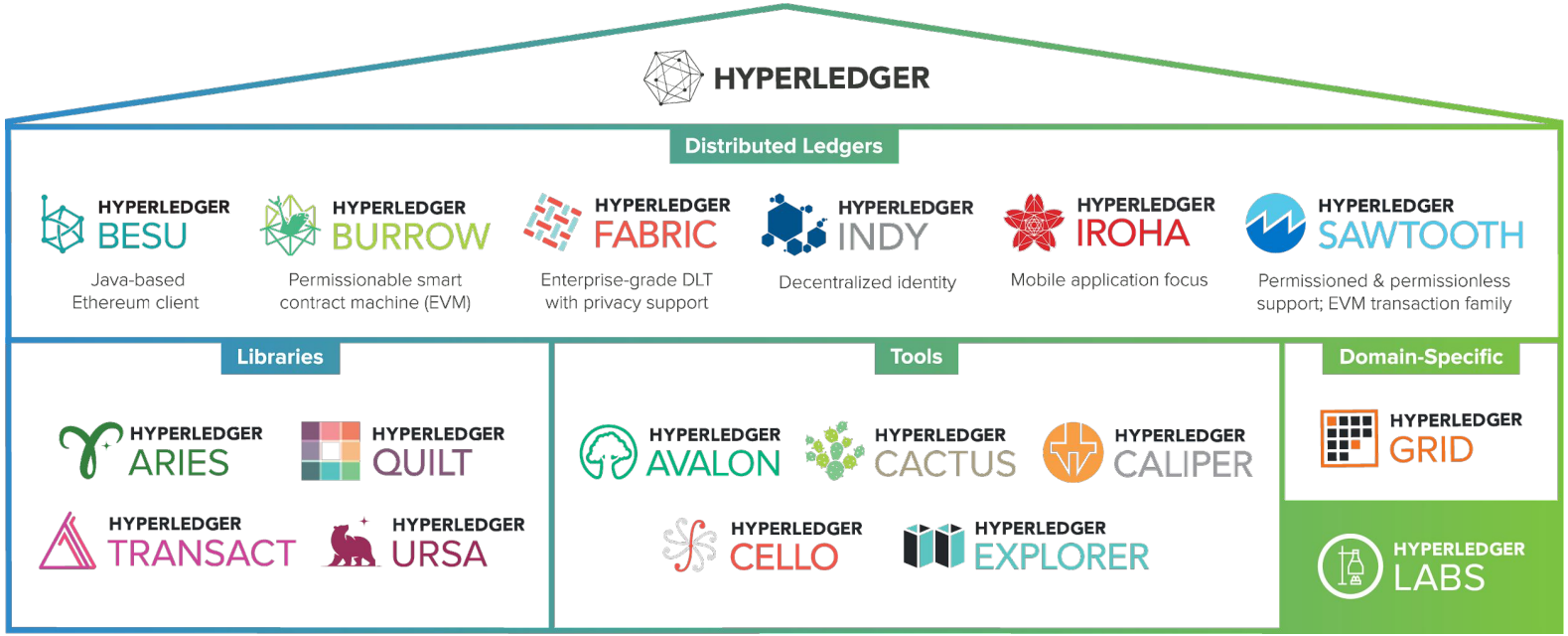
A contribution of ConsenSys in 2019

Formerly named Pantheon

Java-based mainnet client for Ethereum

Supports enterprise requirements

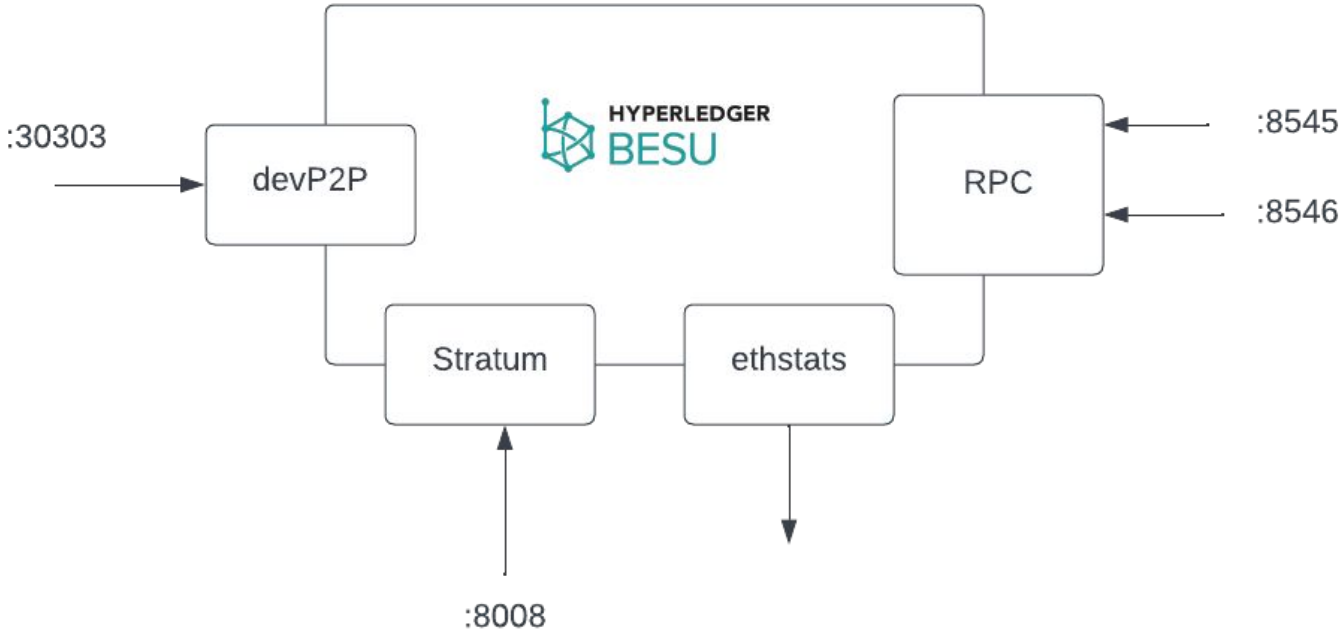
The Hyperledger Greenhouse



Ethereum client - high level

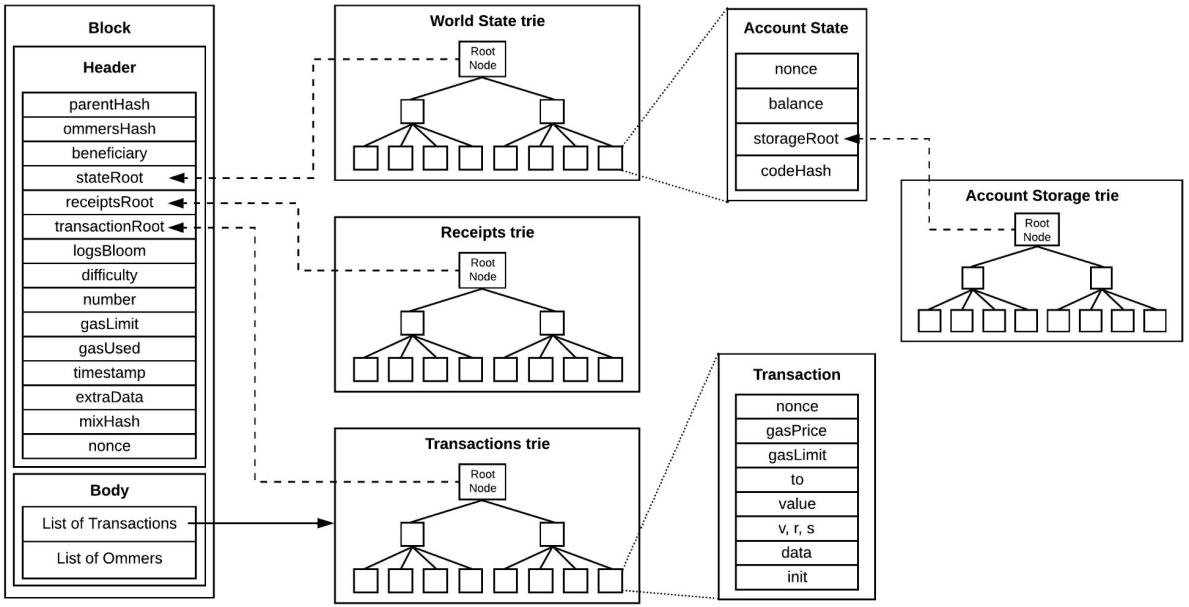
- Misnomer for a server, a peer-to-peer agent
- Runs as a single process
- Independent
  - Can perform all exchanges
  - Can submit transactions
  - Can interrogate the chain

Complex software stack



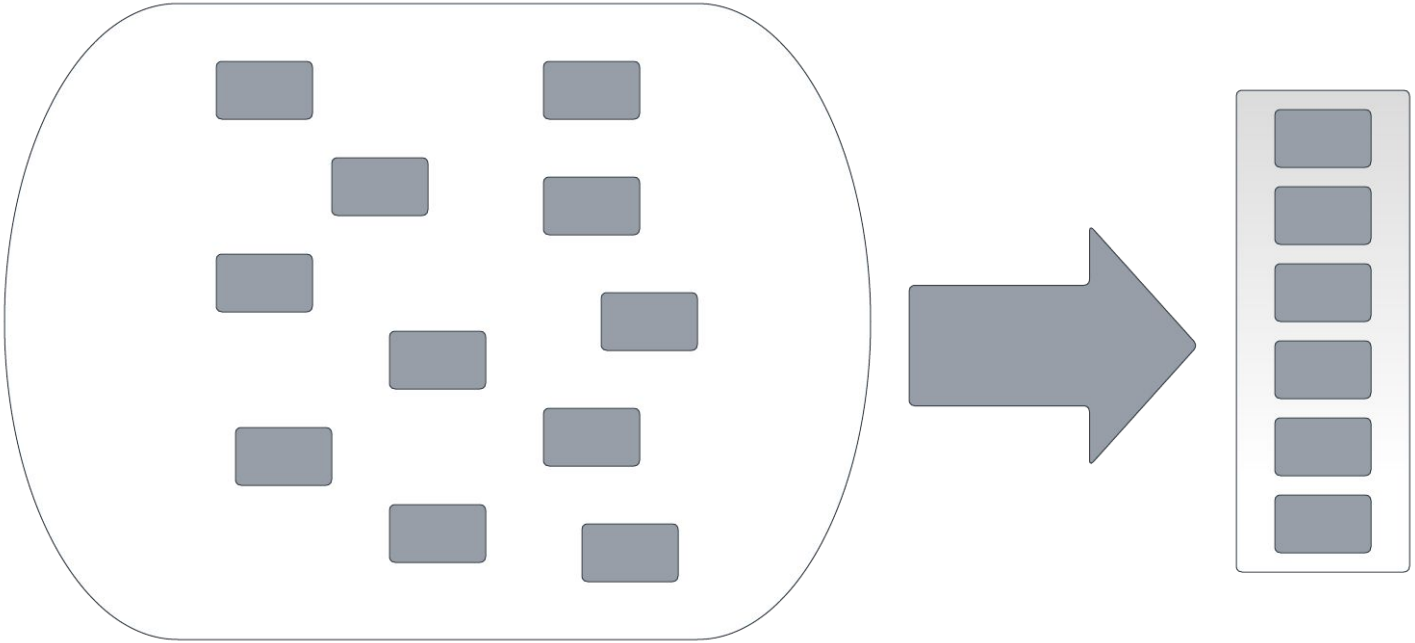
Besu as a database

- 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

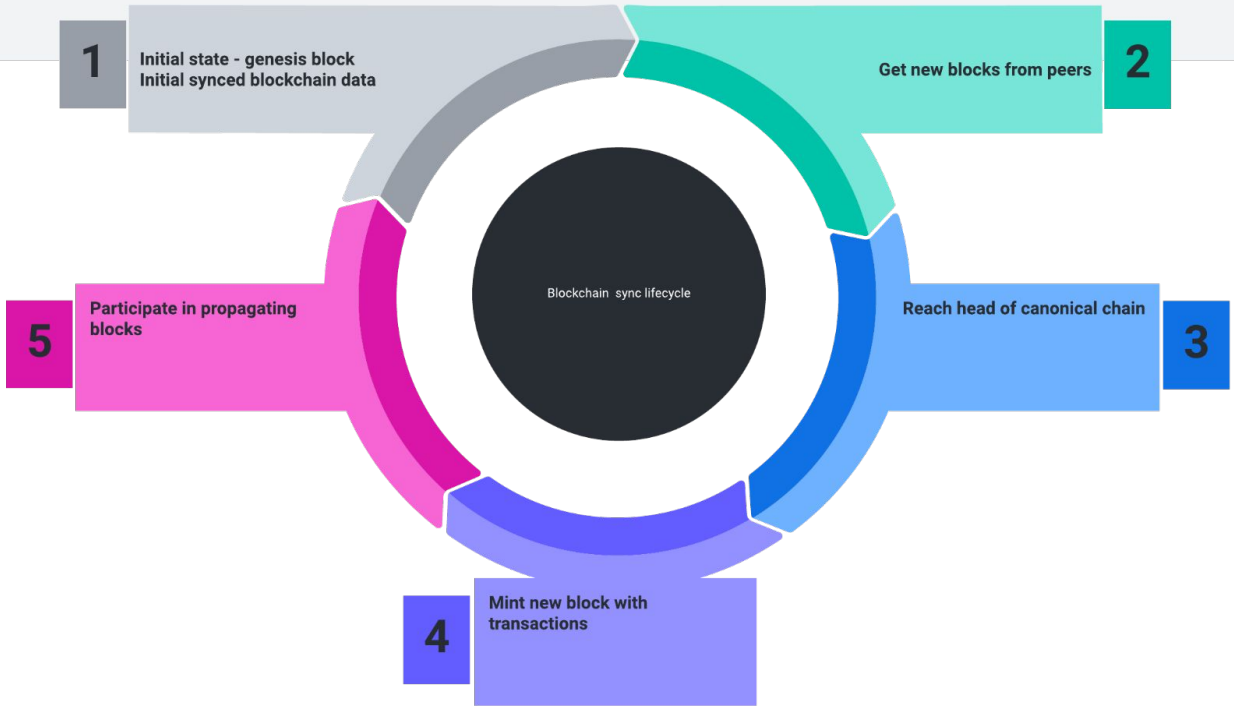
Besu discovery

- 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`

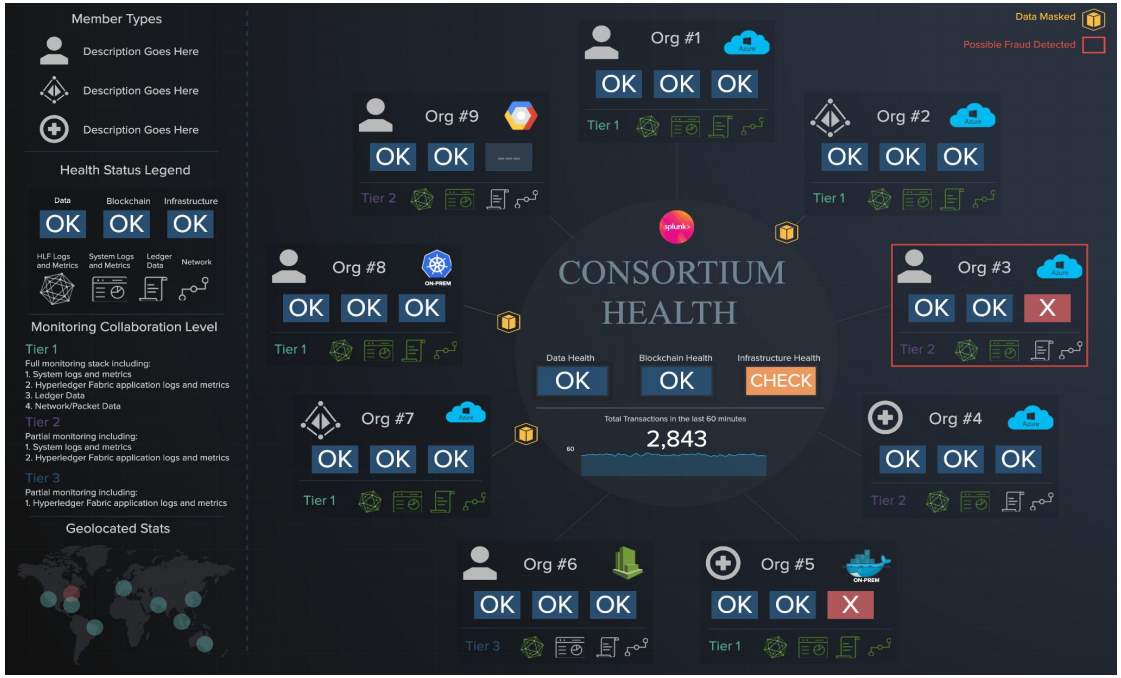
Besu network client

- 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

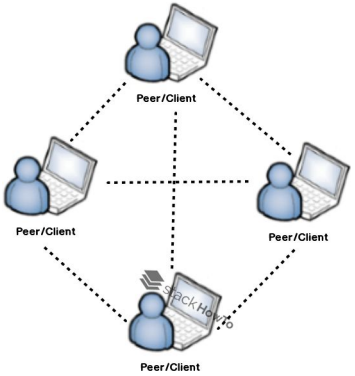
IPC

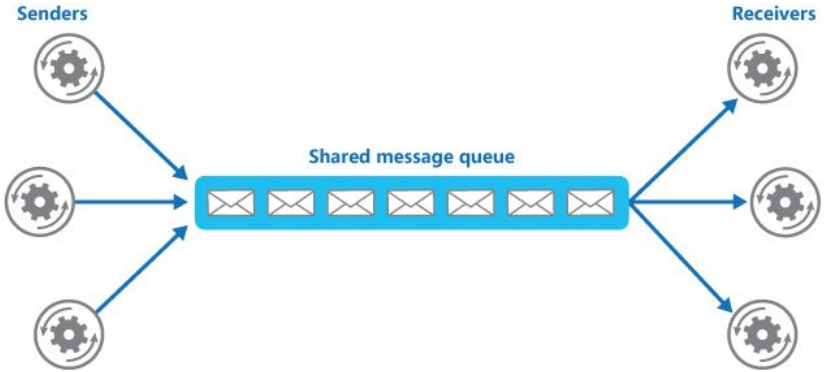
- Using a file socket
- Can be used to attach to the client with geth
- Most secure option
- Just added to Besu in April!

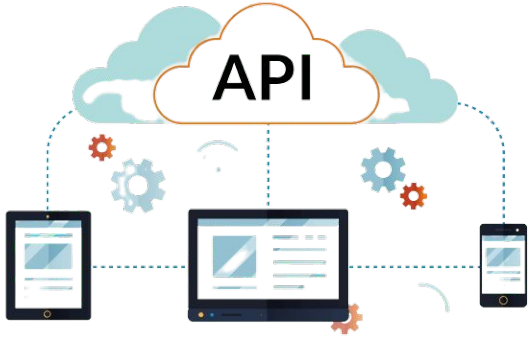
GraphQL

- 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

One more thing…

Ethereum is special because of the EVM.
But where does that play out?

- Validate blocks
- Update the world state
- Create our own blocks by executing transactions


Questions?

Configuring Hyperledger Besu

Hyperledger Foundation workshop

July 14th, 2022

HYPERLEDGER
BLOCKCHAIN TECHNOLOGIES FOR BUSINESS

Hyperledger Besu configuration

Supports command line arguments, env variables, config file - with that order of priority.

You can specify Besu options:

- On the command line.

```
besu [OPTIONS] [SUBCOMMAND]
```

- As an environment variable. For each command line option, the equivalent environment variable is:
  - Uppercase.
  - `_` replaces `-`.
  - Has a `BESU_` prefix.

  For example, set `--miner-coinbase` using the `BESU_MINER_COINBASE` environment variable.
- In a configuration file.

Great docs here! https://besu.hyperledger.org/en/stable/Reference/CLI/CLI-Syntax/

First options

Network	`--network=dev` `--network=ropsten`
Data	`--data-path=folder`
P2P	`--p2p-host=localhost` `--p2p-port=30303`
Discovery	`--enabled=true` `--bootnodes=...`

JSON-RPC

Enablement and APIs

By default JSON-RPC is not enabled. Open it up with --rpc-http-enabled.

--rpc-http-api allows to select which APIs to open:

The available API options are: `ADMIN`, `CLIQUE`, `DEBUG`, `EEA`, `ETH`, `IBFT`, `MINER`, `NET`, `PERM`, `PLUGINS`, `PRIV`, `QBFT`, `TRACE`, `TXPOOL`, and `WEB3`. The default is: `ETH`, `NET`, `WEB3`.

JSON-RPC spec https://ethereum.github.io/execution-apis/api-documentation/

Hidden flags

Unstable options, hidden flags are represented with the --X prefix.

```
$> besu --Xhelp
```

```
Unstable options for Ethereum Wire Protocol
      --Xewp-max-get-bodies=<INTEGER>
         Maximum request limit for Ethereum Wire Protocol GET_BLOCK_BODIES.
           (default: +128)
      --Xewp-max-get-headers=<INTEGER>
         Maximum request limit for Ethereum Wire Protocol GET_BLOCK_HEADERS.
           (default: +192)
      --Xewp-max-get-node-data=<INTEGER>
         Maximum request limit for Ethereum Wire Protocol GET_NODE_DATA.
           (default: +384)
      --Xewp-max-get-pooled-transactions=<INTEGER>
         Maximum request limit for Ethereum Wire Protocol
           GET_POOLED_TRANSACTIONS. (default: +256)
      --Xewp-max-get-receipts=<INTEGER>
         Maximum request limit for Ethereum Wire Protocol GET_RECEIPTS.
           (default: +256)
```

Ways to run Besu

Download the distribution	From Github https://github.com/hyperledger/besu/releases
Homebrew	`brew install besu`
Docker	`docker pull hyperledger/besu`

From source: `./gradlew assemble`

OS support: x86 with native libraries
ARM support - M1 support in progress

Advanced options

Genesis file	`--genesis-file=<genesis.json>` Use a custom Genesis file

RPC security	`--rpc-http-host`
	`--rpc-http-cors-origins`
	`--rpc-http-tls-client-auth-enabled`
	`--rpc-http-authentication-jwt-public-key-file`
	`--rpc-http-authentication-credentials-file`

Metrics	`---metrics-enabled`
	`---metrics-port` and `---metrics-host`
	`--metrics-protocol`

Miner	`--miner-enabled`
	`--miner-stratum-enabled`
	`--miner-coinbase`

Exercises - running with --network=dev

Check out Besu and run with --network=dev --rpc-http-enabled

Check out the genesis file here:
https://github.com/hyperledger/besu/blob/main/config/src/main/resources/dev.json

From the command line, check the balance of an address with:

```
curl http://localhost:8545/ \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{
           "method":"eth_getBalance",
           "params":["0x627306090abaB3A6e1400e9345bC60c78a8BEf57", "latest"],
           "id":1,
           "json-rpc":"2.0"
         }'
```

Exercises - Genesis file specification

Following the tutorial:

https://besu.hyperledger.org/en/stable/Tutorials/Private-Network/Create-IBFT-Network/

Exercises - Quorum Dev Quickstart

ConsenSys has created a tool to generate complex Besu networks, with the option to use private enclaves, and monitoring tools of your choice.

Use `npx quorum-dev-quickstart` to get started.


Questions?