Como Operar e Usar Hyperledger Besu em Redes Públicas e Privadas

Hyperledger Foundation Workshop

14 de Dezembro de 2023

Conteúdo

- Introdução ao Besu - o que ele faz, como funciona
- Configuração do Besu - variáveis de ambiente, arquivo toml, flags do CLI, flags ocultas. Instalação por Docker versus instalação binária.
- Redes públicas
  - Proof of Stake e The Merge
  - Como executar um nó Besu em redes públicas
- Execução de uma rede Besu
  - Besu em modo de desenvolvimento, curl, primeiros passos fáceis.
  - Geração do bloco Genesis
  - Criação de uma rede privada, com monitoramento e relatório de saúde, usando o Docker compose.
- Como contribuir

Samuel Venzi

- Experiência com blockchain desde 2018
  - 2018: Início dos estudos em Hyperledger Fabric
  - 2021: Início dos estudos em Hyperledger Besu
- CTO na GoLedger desde de 2020
  - Membros da Hyperledger Foundation
- Contribuidor do projeto Hyperledger Fabric
- Contribuidor da certificação oficial de Fabric: HFCP
- Mantenedor da biblioteca de contratos-inteligentes para Fabric: cc-tools

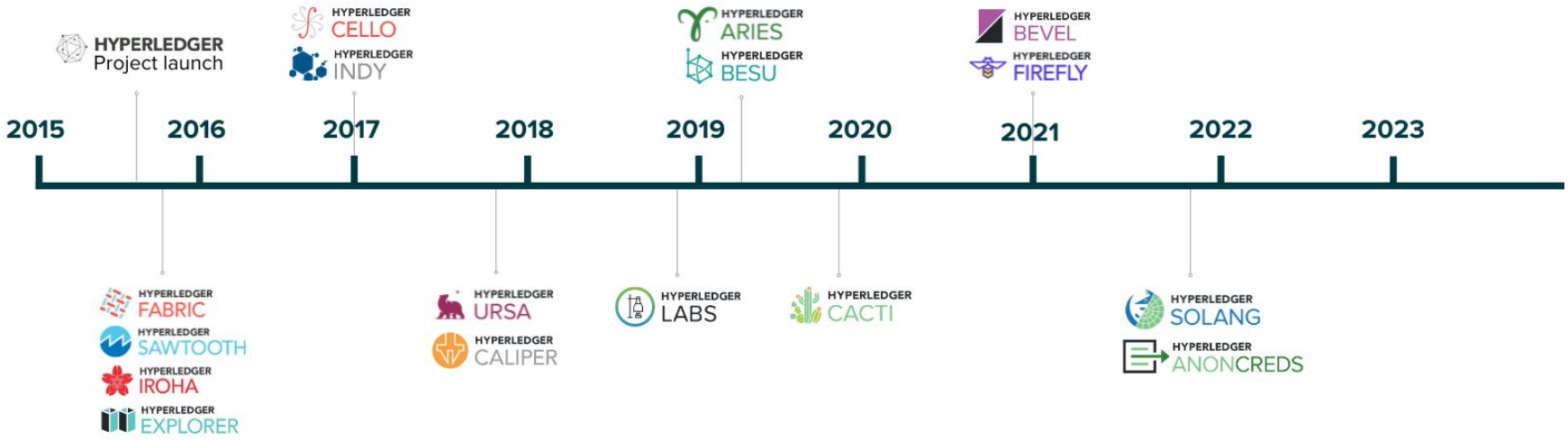
Hyperledger Foundation

- Open Source
- Open Development
- Open Governance

- 7 anos de idade
- 6 projetos graduados
- 7 projetos incubados
- 50 projetos Hyperledger Labs

- Capítulos regionais: Hyperledger Chapter Brazil

Hyperledger Foundation



Ethereum

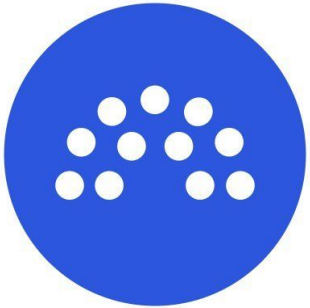
- Segunda maior criptomoeda em *market cap*
- Início em 2014
- Suporta vários clientes diferentes
  - Contribui para segurança da rede
- Execução de contratos inteligentes em uma camada específica, programável
  - EVM

Enterprise

- O mundo enterprise tem requisitos diferentes
- Suporte do cliente é vital
- Implantação
  - Consenso diferente
  - Permissionado
- Segurança!
  - Auditoria
  - Gestão dos dados

Quorum

- Primeira tentativa do Ethereum no mundo enterprise
- Desenvolvido pelo JP Morgan, e após isso pela Consensys
- Fork do Geth
- Novos algoritmos de consenso
- Camadas de privacidade de transações
- Licença GPL



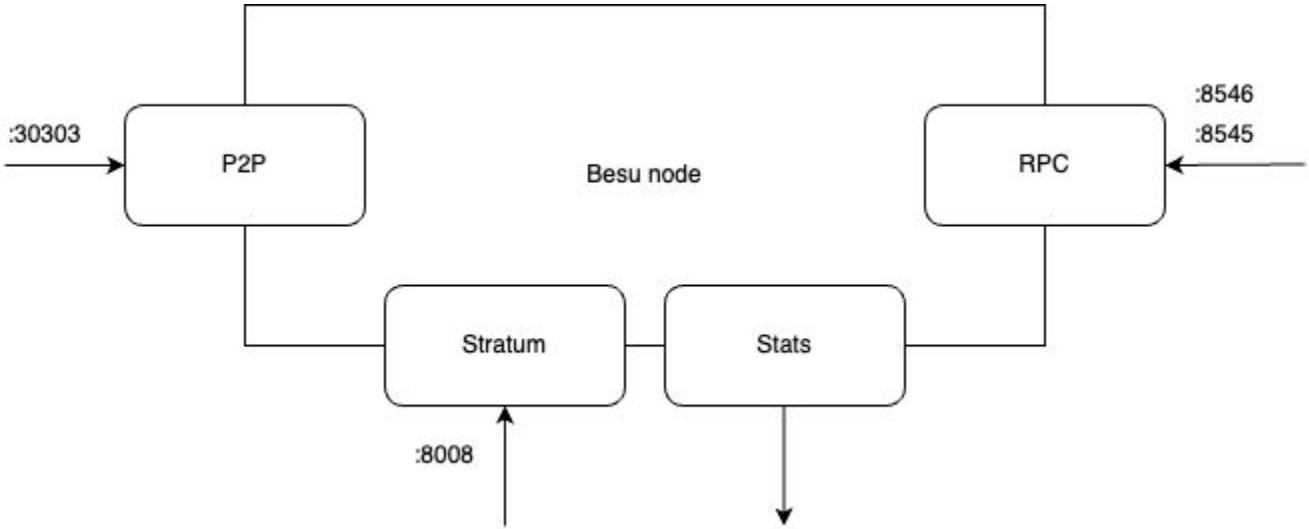
Hyperledger Besu

- Cliente open-source para Ethereum escrito em Java
  - Consequentemente, EVM-compatible
- Projeto contribuído pela Consensys em 2019
  - Conhecido originalmente como Pantheon
  - Agora faz parte dos projetos da Hyperledger Foundation
- Pode ser usado em redes públicas ou privadas

Clients de Ethereum

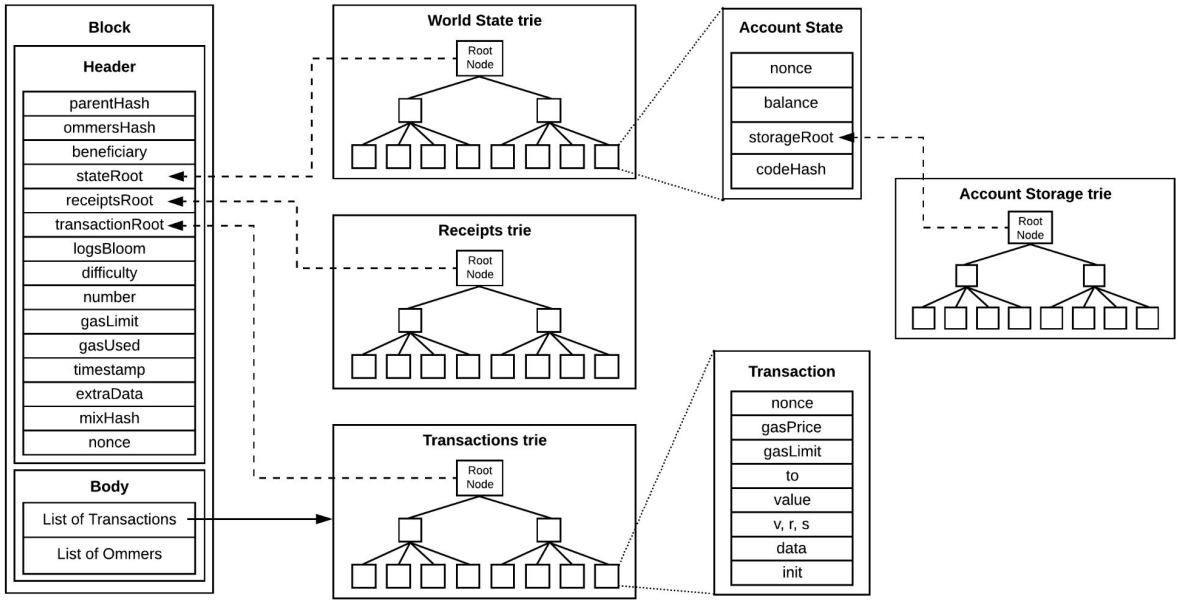
- Mais precisamente, um servidor peer-to-peer
- Roda um único processo
- Independente
  - Pode fazer todas as "trocas" que a rede requer
  - Pode submeter transações
  - Pode recuperar dados da cadeia
- Exemplos
  - Geth – Go
  - Hyperledger Besu – Java
  - Nethermind – .NET
  - Reth – Rust

Besu como uma caixa preta



Besu como base de dados

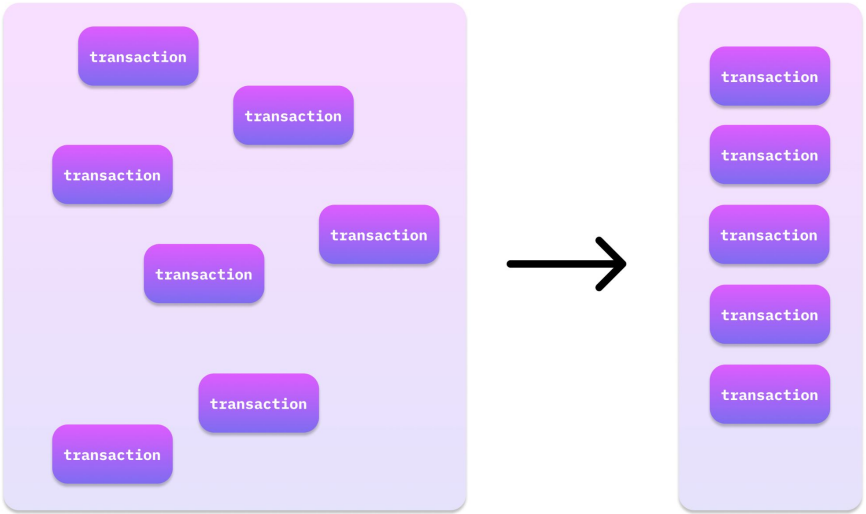
- RocksDB – key-value
- Merkle Patricia Trie



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

Besu como tx pool

- Besu recebe transações na sua tx pool
- Ordenar em um possível bloco
    - Aplicação de um algoritmo de ordenação (baseado em fees)



Redes Besu

- Cada cliente é independente, então é necessário configurá-lo

Bloco genesis **Consenso** **Bootnodes**

Besu discovery

- Conectar a outros nós via UDP
  - Os primeiros são os bootnodes
  - O restante são os nós expostos pelo bootnode
- Guardar nós para evitar *eclipse attacks*
  - Kademlia hashtable
- Discovery com DNS
  - Indexa a partir do bootnodes
  - Guardado em estrutura criptográfica
- Estático
  - Configurar enodes
  - enode://...

Besu client

- Usando P2P
  - Enviar uma mensagem HELLO para outros nós
  - Confirmação de subprotocolos suportados (incluindo consenso: IBFT, QBFT)
  - enode com chave pública é importante para verificação das assinaturas

Ciclo de vida do Hyperledger Besu

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

Besu e consenso

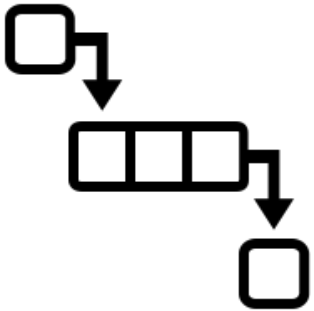
- Ethash – Proof of Work disponível para Ethereum Classic
- Proof of Stake – The Merge
  - Execution layer – Hyperledger Besu
  - Consensus layer
- Clique – Proof of Authority
- IBFT e QBFT – Proof of Authority

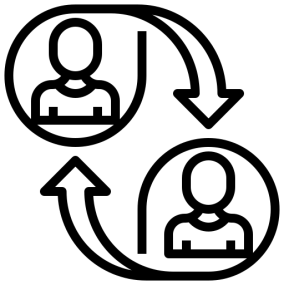
Servidor JSON-RPC

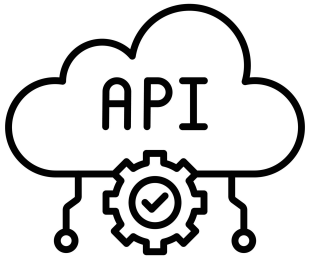
- HTTP
  - Suporte a batching
  - Usado por wallets como MetaMask
- WebSocket
  - Subscriptions
  - Eventos e logs
- IPC (*early access*)
  - File socket
  - Opção mais segura
  - Pode ser usado para anexar o cliente ao Geth
- GraphQL
  - API versátil
  - Fazer queries específicas

Em resumo

- Besu é uma base de dados
- Besu é um fila
- Besu é um agente peer-to-peer
- Besu é uma API

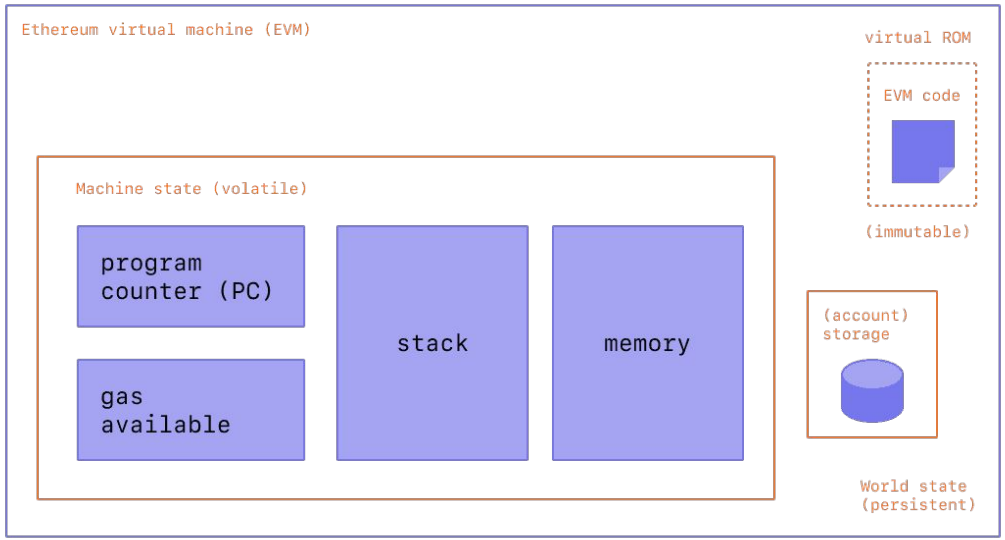






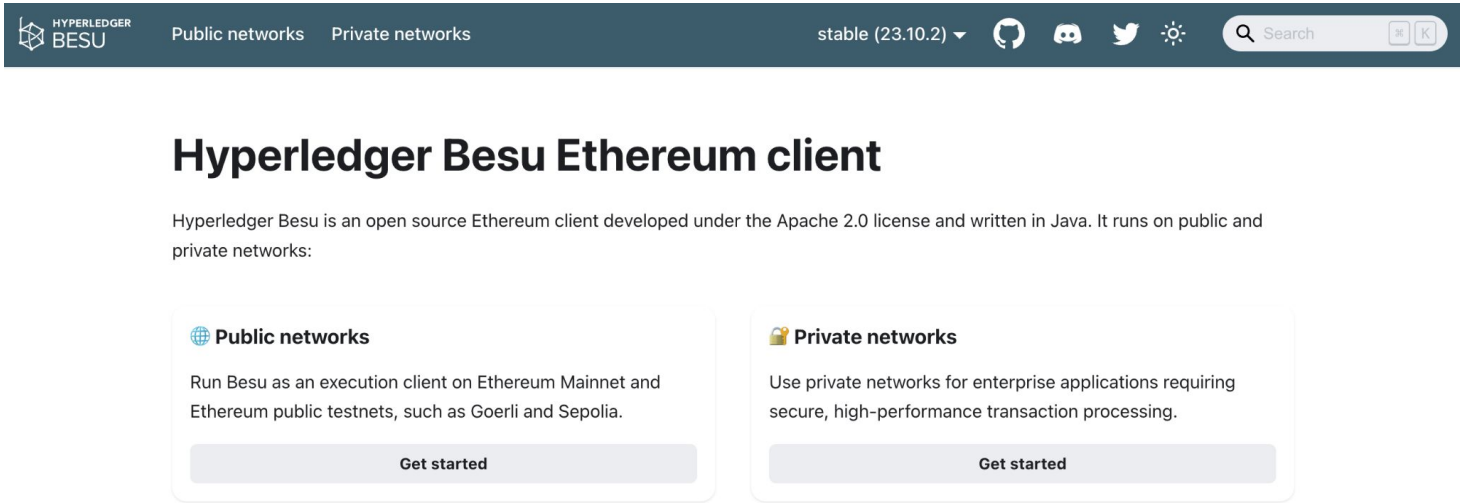
Besu é EVM

- Como isso influencia?
  - Validação de blocos
  - Atualizações do estado do mundo
  - Criação de blocos



Configuração do Hyperledger Besu

- Documentação!
  - https://besu.hyperledger.org/public-networks/reference/cli/options



Configuração do Hyperledger Besu

- Command Line Interface
  - Configuração com suporte a argumentos, variáveis de ambiente e arquivos de configuração – nessa ordem de prioridade

Specify options

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.

Opções essenciais de configuração

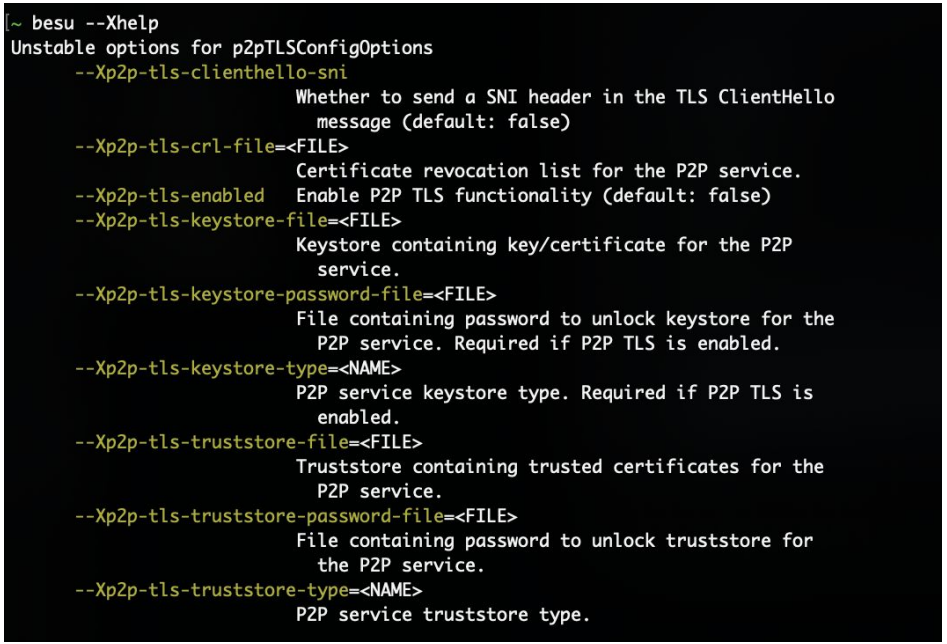
Network	`--network=dev` `--network=ropsten`
Data	`--data-path=folder`
P2P	`--p2p-host=localhost` `--p2p-port=30303`
Discovery	`--enabled=true` `--bootnodes=...`

- Por padrão a API RPC é desabilitada, habilite com
  - `--rpc-http-enabled` e `--rpc-http-api=...`

Flags ocultas

- Opções instáveis, são representadas com o prefixo --X

`besu –Xhelp`



Arquivo .toml

```
besu --config-file=/path/config.toml
```

Sample TOML configuration file

```toml
# Valid TOML config file
data-path="~/besudata" # Path

# Network
bootnodes=["enode://001@123:4567", "enode://002@123:4567", "enode://003@123:4567"]

p2p-host="1.2.3.4"
p2p-port=1234
max-peers=42

rpc-http-host="5.6.7.8"
rpc-http-port=5678

rpc-ws-host="9.10.11.12"
rpc-ws-port=9101

# Chain
genesis-file="~/genesis.json" # Path to the custom genesis file

# Mining
miner-enabled=true
miner-coinbase="0xfe3b557e8fb62b89f4916b721be55ceb828dbd73"
```

Como rodar o Besu

- Baixar a distribuição
  - https://github.com/hyperledger/besu/releases
- Homebrew
  - `brew install besu`
- Docker
  - `docker pull hyperledger/besu`

- From source: `./gradlew assemble`

OS Support: x86, ARM (Apple Silicon) em progresso

Opções avançadas de configuração

Genesis file	`--genesis-file=<genesis.json>`

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`

Rodando o Besu com rede dev

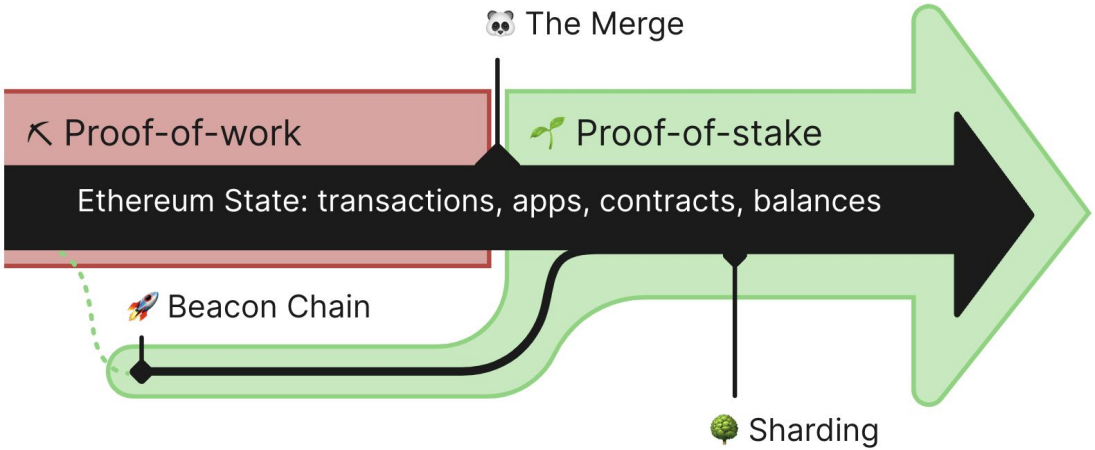
- `besu --network=dev --rpc-http-enabled`

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

https://github.com/samuelvenzi/besu-workshop

The Merge e Proof of Stake

- O Merge foi a união da camada de execução original do Ethereum (a Mainnet que existe desde o início) com sua nova camada de consenso proof-of-stake, a Beacon Chain.



https://ethereum.org/en/roadmap/merge/

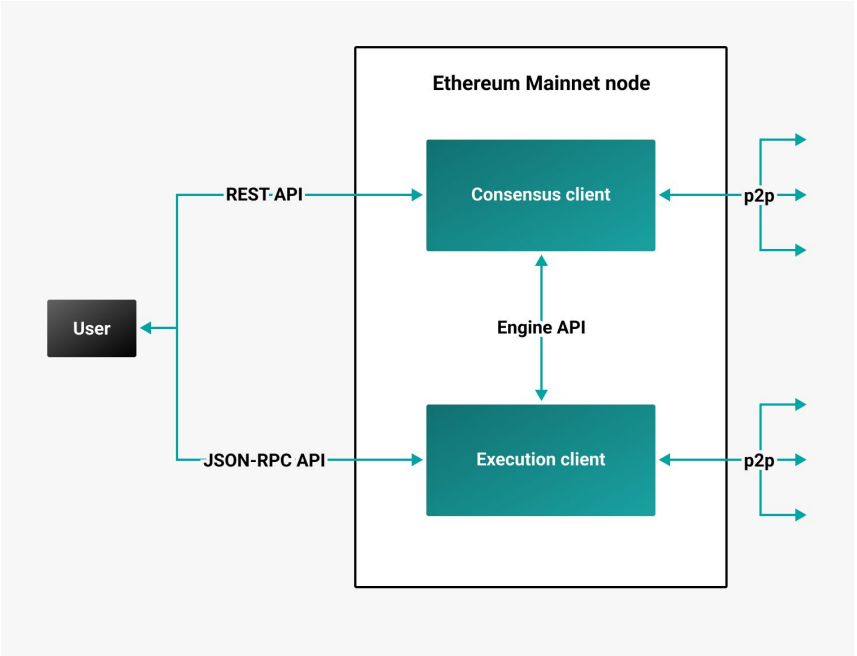
Proof of Work vs. Proof of Stake

- PoW é um mecanismo de consenso que requer a **resolução de problemas computacionais** complexos para validar e registrar transações na blockchain

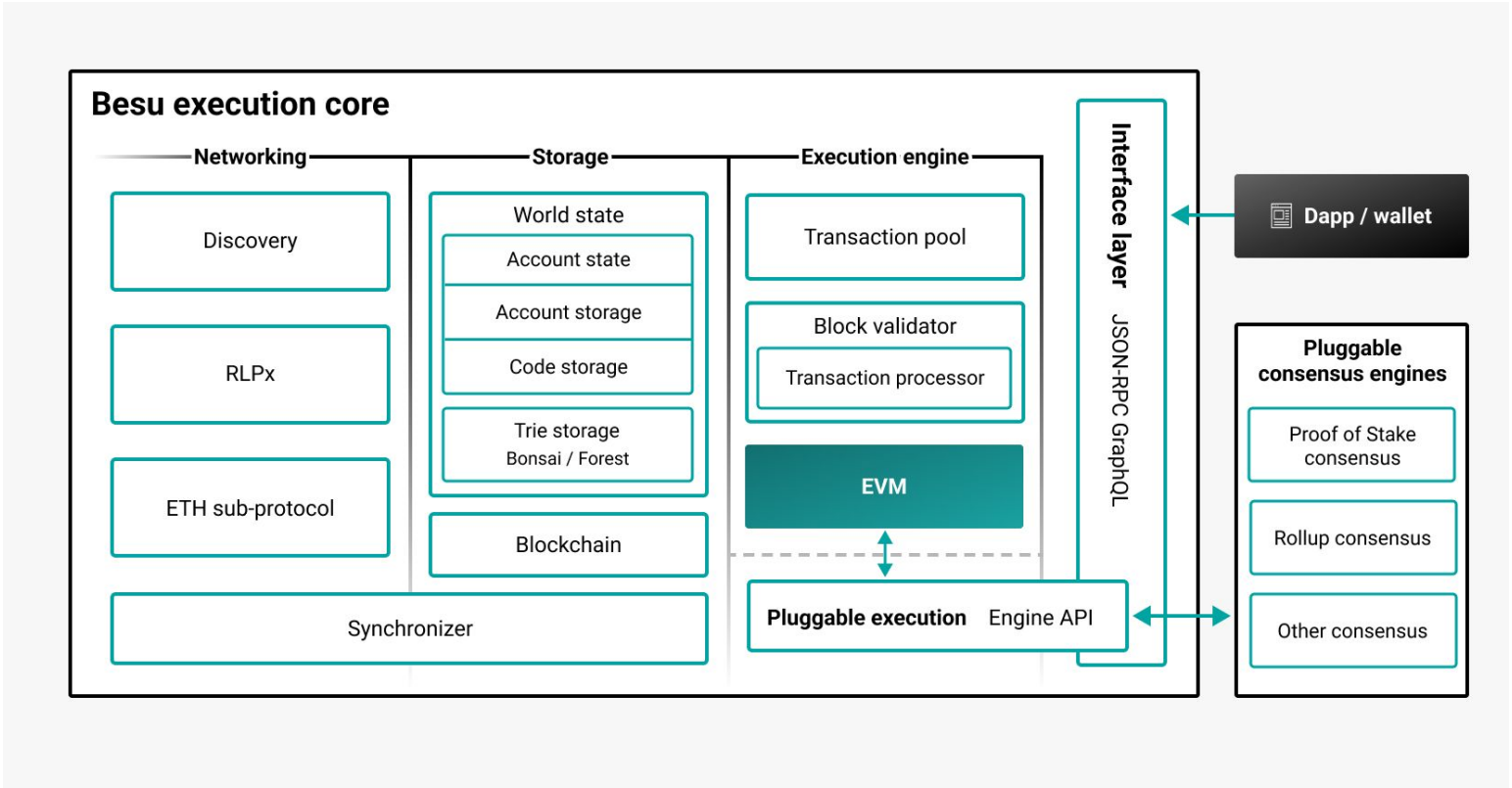
- PoS é um mecanismo de consenso que permite a validação de transações e criação de novos blocos **baseado na quantidade de criptomoeda** que um usuário detém e está disposto a **"apostar" ou bloquear como garantia**.

Besu e Proof of Stake

- Hyperledger Besu é um cliente de execução (Ethereum 1.0)
- Em conjunto com um cliente de consenso (Ethereum 2.0) compõem "Ethereum node"
- Comunicação entre os cliente via Engine API



Arquitetura do Besu para Redes Públicas



Rodando o Besu com Teku na Goerli

```
besu \
  --network=goerli             \
  --p2p-host=<external-ip>     \
  --p2p-port=30303             \
  --rpc-http-enabled=true      \
  --rpc-http-host=0.0.0.0      \
  --rpc-http-cors-origins="*" \
  --rpc-ws-enabled=true        \
  --rpc-ws-host=0.0.0.0        \
  --host-allowlist="*"         \
  --engine-host-allowlist="*" \
  --engine-rpc-enabled         \
  --engine-jwt-secret=jwtsecret.hex
```

```
teku \
  --network=goerli                                  \
  --ee-endpoint=http://localhost:8551               \
  --ee-jwt-secret-file=jwtsecret.hex                \
  --metrics-enabled=true                            \
  --rest-api-enabled=true                           \

--initial-state=https://checkpoint-sync.goerli.ethpandaops.io/eth/v2/debug/beacon/states/finalized
```

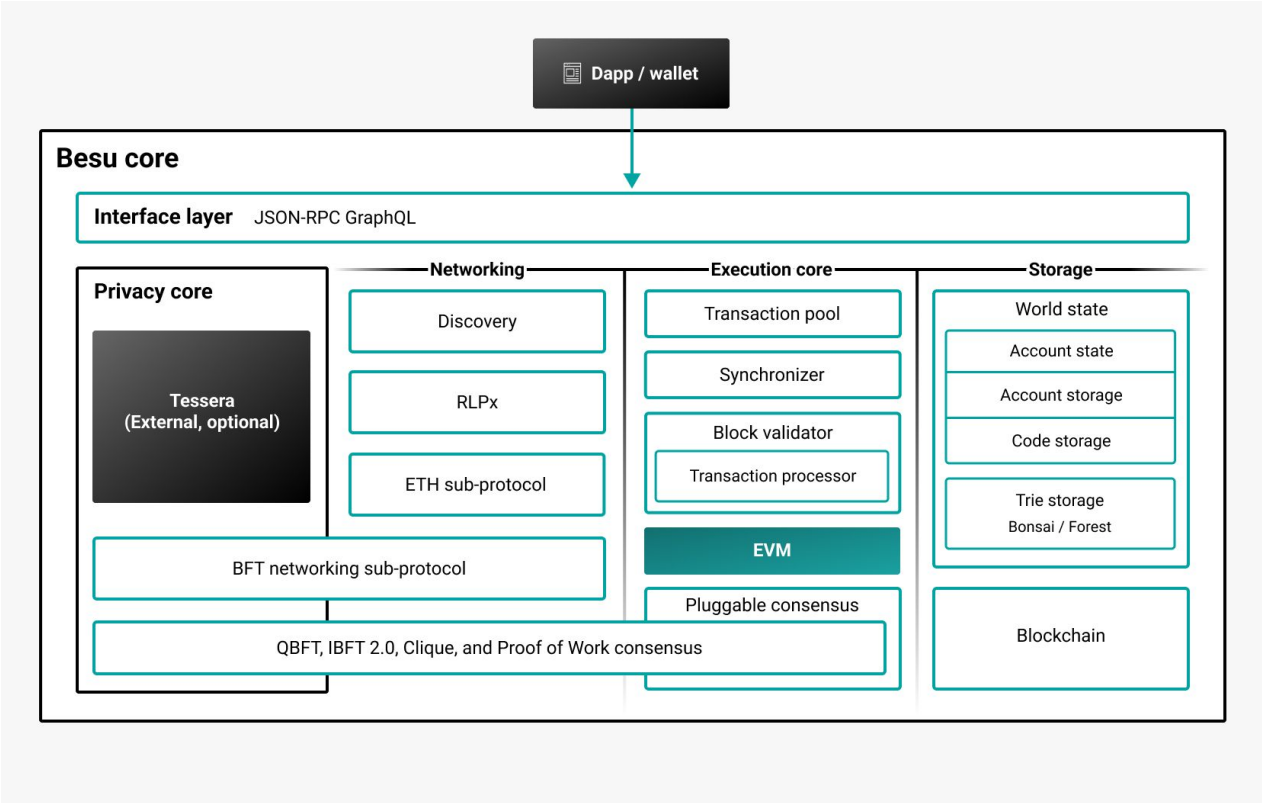
Besu e redes privadas

- Permissionamento
  - Acesso restrito a membros selecionados.
- Maior controle sobre governança e privacidade.
  - Adição e remoção de validadores
- Adequada para transações confidenciais empresariais.
  - Blockchain independente
- Free gas
  - Validadores não necessitam de incentivo para validação

Consenso para redes privadas

- Proof of Authority: Protocolos de consenso PoA funcionam quando os participantes se conhecem e há um nível de confiança entre eles. Por exemplo, em uma rede de consórcio permissionada.
  - Clique
    - Não possui finalidade imediata
    - Tolera até metade dos validadores falhando
  - IBFT
    - Possui finalidade imediata
    - Tolera até ⅓ dos validadores falhando
  - QBFT
    - Possui finalidade imediata
    - Tolera até ⅓ dos validadores falhando
    - Recomendada para produção em Enterprise

Arquitetura do Besu para Redes Privadas



Geração de um genesis

```
besu operator generate-blockchain-config
--config-file=../config/qbftConfigFile.json --to=networkFiles
--private-key-file-name=key
```

Criação da rede e deploy de contrato

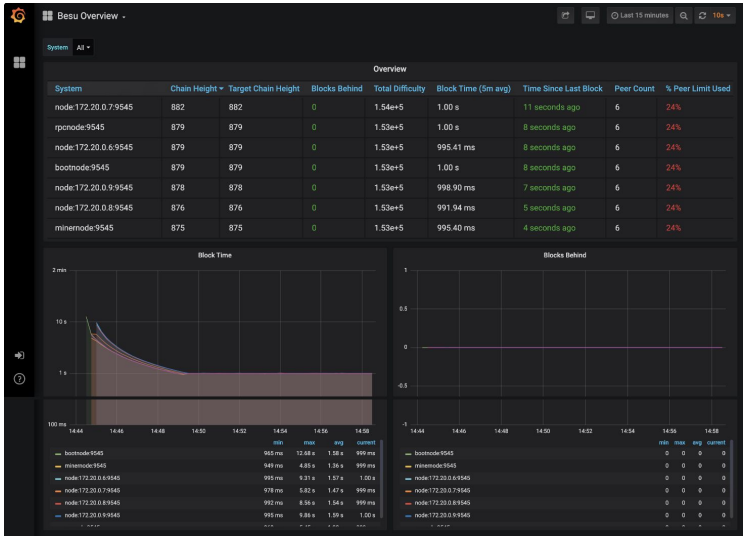
- Deploy de uma rede com 4 nós com consenso QBFT
  - Utilizando containers e Docker Compose

```
besu --data-path=data --genesis-file=genesis/genesis.json
--min-gas-price=0 --rpc-http-enabled --rpc-http-api=ETH,NET,QBFT
--host-allowlist="*" --rpc-http-cors-origins="all"
```

Monitoramento

- Prometheus para scraping de métricas
- Grafana para montagem de dashboards

https://grafana.com/grafana/dashboards/16455-besu-full/




```
global:
  scrape_interval: 15s

scrape_configs:
  - job_name: "prometheus"
    static_configs:
      - targets:
["prometheus:9090"]
  - job_name: besu
    scrape_interval: 15s
    scrape_timeout: 10s
    metrics_path: /metrics
    scheme: http
    static_configs:
      - targets:
          -
besu-node-0:9545
```

Contribuições para o projeto

- Hyperledger Besu Core
  - https://github.com/hyperledger/besu/blob/main/CONTRIBUTING.md
- Hyperledger Besu Documentação
  - https://wiki.hyperledger.org/display/BESU/Documentation
- Discord
  - https://discord.gg/hyperledger
  - Canal **#besu**
- Discussões em Português
  - Hyperledger Chapter Brasil
  - **#brasil-chapter** no Discord
  - WhatsApp (convite via Discord)