



Hyperledger Besu

Project	
Status	ACTIVE
CI/CD Badge	
Description	Besu is an Ethereum client that runs on the Ethereum public network, private networks, and test networks such as Rinkeby, Ropsten, and Görli.

Key Characteristics

What is Hyperledger Besu?

Hyperledger Besu is an open source Ethereum client developed under the Apache 2.0 license and written in Java. It can be run on the Ethereum public network or on private permissioned networks, as well as test networks such as Rinkeby, Ropsten, and Görli. Hyperledger Besu includes several consensus algorithms including PoW, PoA, and IBFT, and has comprehensive permissioning schemes designed specifically for uses in a consortium environment.

What are Hyperledger Besu's Features?

Hyperledger Besu implements the [Enterprise Ethereum Alliance](#) (EEA) specification. The EEA specification was established to create common interfaces amongst the various open and closed source projects within Ethereum, to ensure users do not have vendor lock-in, and to create standard interfaces for teams building applications. Besu implements enterprise features in alignment with the EEA client specification.

Hyperledger Besu's features include:

- **The Ethereum Virtual Machine (EVM)** : The EVM is the Turing complete virtual machine that allows the deployment and execution of smart contracts via transactions within an Ethereum blockchain.
- **Consensus Algorithms**: Hyperledger Besu implements various consensus algorithms which are involved in transaction validation, block validation, and block production (i.e., mining in Proof of Work). They include:
 - **Proof of Authority**: Hyperledger Besu implements several Proof of Authority protocols. Proof of Authority consensus protocols are used when participants are known to each other and there is a level of trust between them—in a permissioned consortium network, for example.
 - **IBFT 2.0**: In IBFT 2.0 networks, transactions and blocks are validated by approved accounts, known as validators. Validators take turns creating the next block. Existing validators propose and vote to add or remove validators. IBFT 2.0 has immediate finality. When using IBFT 2.0, there are no forks and all valid blocks are included in the main chain.
 - **Clique**: Clique is more fault-tolerant than IBFT 2.0. Clique tolerates up to half of the validators failing. IBFT 2.0 networks require greater than or equal to of validators to be operating to create blocks. Clique does not have immediate finality. Implementations using Clique must be aware of forks and chain reorganizations occurring.
 - **Proof of Work (Ethash)**: Proof of Work is used for mining activities on mainnet Ethereum.
- **Storage**: Hyperledger Besu uses a RocksDB key-value database to persist chain data locally. This data is divided into a few sub-categories:
 - **Blockchain**: Blockchain data is composed of block headers that form the “chain” of data that is used to cryptographically verify blockchain state; block bodies that contain the list of ordered transactions included in each block; and transaction receipts that contain metadata related to transaction execution including transaction logs.
 - **World State**: Every block header references a world state via a stateRoot hash. The world state is a mapping from addresses to accounts. Externally owned accounts contain an ether balance, while smart contract accounts additionally contain executable code and storage.
- **P2P networking**: Hyperledger Besu implements Ethereum's devp2p network protocols for inter-client communication and an additional sub-protocol for IBFT2:
 - **Discovery**: A UDP-based protocol for finding peers on the network
 - **RLPx**: A TCP-based protocol for communication between peers via various “sub-protocols”:
 - **ETH Sub-protocol** (Ethereum Wire Protocol): Used to synchronize blockchain state across the network and propagate new transactions.
 - **IBF Sub-protocol**: Used by IBFT2 consensus protocol to facilitate consensus decisions.
- **User-facing APIs**: Hyperledger Besu provides mainnet Ethereum and EEA JSON-RPC APIs over HTTP and WebSocket protocols as well as a GraphQL API.
 - **JSON-RPC**
 - HTTP JSON-RPC Service
 - WebSocket JSON-RPC Service
 - GraphQL
- **Monitoring**: Hyperledger Besu allows you to monitor node and network performance.
 - Node performance is monitored using Prometheus or the debug_metrics JSON-RPC API method.
 - Network Performance is monitored with [Alethio](#) tools such as Block Explorer and EthStats Network Monitor.

- **Privacy:** Privacy in Hyperledger Besu refers to the ability to keep transactions private between the involved parties. Other parties cannot access the transaction content, sending party, or list of participating parties. Besu uses a Private Transaction Manager to implement privacy.
- **Permissioning:** A permissioned network allows only specified nodes and accounts to participate by enabling node permissioning and/or account permissioning on the network.

Documentation

Documentation on Hyperledger Besu can be found here: <https://besu.hyperledger.org/>

Repositories

<https://github.com/hyperledger/besu/>

<https://github.com/hyperledger/besu-docs>

Communication

Mailing List

- besu@lists.hyperledger.org
 - [subscribe](#)
 - [messages](#)

Chat (for questions and ephemeral discussions)

The Hyperledger Project use Rocket Chat.

First you must get a Linux Foundation Identity at <https://identity.linuxfoundation.org/> and then log in at <https://chat.hyperledger.org/>, then you can log in directly to one of the channels below.

- [#besu](#)
- [#besu-contributors](#)

Meeting

[Create meeting note](#)

Incomplete tasks from meetings

Task report

Looking good, no incomplete tasks.

All meeting notes

Title	Creator	Modified
2020-04-28 Contributor Call Notes	Danno Ferrin	Apr 28, 2020
2020-04-14 Contributor Call	Danno Ferrin	Apr 16, 2020
2020-03-17 Contributor Call Notes	Felipe Faraggi	Mar 17, 2020
2020-02-18 Besu Contributor Call Notes	Edward Evans	Feb 18, 2020
2020-02-04 Besu Contributor Call Notes	Danno Ferrin	Feb 05, 2020
2020-01-21 Contributor Call	Brett Henderson	Jan 21, 2020
2020-01-07 Contributor Call	Felipe Faraggi	Jan 09, 2020
2019-12-10 Contributor Call	Danno Ferrin	Dec 10, 2019
2019-11-26 Contributor Call	Danno Ferrin	Nov 26, 2019
2019-11-12 Contributor Call	Danno Ferrin	Nov 12, 2019
2019-10-29 Contributor Call	Danno Ferrin	Nov 01, 2019
2019-10-15 Contributor Call	Danno Ferrin	Oct 18, 2019
2019-10-01 Contributor call	Felipe Faraggi	Oct 02, 2019
2019-09-17 Contributor call	Danno Ferrin	Sep 30, 2019
2019-09-04 Onboarding call	Ry Jones	Sep 11, 2019

History

Recent space activity



[Antoine Toulme](#)

[2020-05-26 Contributor Call](#) updated May 26, 2020 • [view change](#)



[Danno Ferrin](#)

[2020-05-26 Contributor Call Notes](#) updated May 26, 2020 • [view change](#)

[2020-05-26 Contributor Call](#) updated May 26, 2020 • [view change](#)

[Lucas Saldanha](#)

[2020-05-26 Contributor Call](#) updated May 25, 2020 • [view change](#)



[Sally MacFarlane](#)

[Defect Prioritisation Policy](#) commented May 14, 2020

Space contributors

- [Antoine Toulme](#) (11 days ago)
- [Danno Ferrin](#) (11 days ago)
- [Lucas Saldanha](#) (11 days ago)
- [Sally MacFarlane](#) (23 days ago)
- [Meredith Baxter](#) (24 days ago)
- ...