Hyperledger Besu Proposal

Project Identifier

HIP: Hyperledger Besu

Sponsor(s)

Joseph Lubin, ConsenSys, joseph.lubin@consensys.net

Daniel Heyman, ConsenSys / PegaSys, daniel.heyman@consensys.net

Rob Dawson, ConsenSys / PegaSys, rob.dawson@consensys.net

Danno Ferrin, ConsenSys / PegaSys, danno.ferrin@consensys.net

Grace Hartley, ConsenSys / PegaSys, grace.hartley@consensys.net

Jonathan Levi, Unbounded Network & HACERA, email: jonathan@hacera.com

Samer Falah, JPM, samer.faleh@jpm.com

Mark Wagner, Red Hat, mwagner@redhat.com

Abstract

Besu is an open-source Ethereum client developed under the Apache 2.0 license and written in Java. It runs on the Ethereum public network, private networks, and test networks such as Rinkeby, Ropsten, and Görli.

You can also use Besu to develop enterprise applications requiring secure, high-performance transaction processing in a private network.

Besu supports enterprise features including privacy and permissioning.

It includes:

- A command line interface and JSON-RPC API for running, maintaining, debugging, and monitoring nodes in an Ethereum network
- *Consensus Protocols:* Besu implements a number of consensus protocols: Ethash (Proof of Work), IBFT 2.0 and Clique (Proof of Authority).
- *Privacy*: Privacy in 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.
- *Permissions*: Besu enables onchain and local permissioning.
- *Monitoring*: You can monitor various aspects of node and network performance in Besu using Prometheus with Grafana, the JSON-RPC APIs or the Alethio Ethereum Lite Explorer.

Context

Ethereum is the most popular smart-contract based public blockchain, used by over 300,000 developers around the globe. Current projects and companies vary widely — from interbank settlement, to consumer lending platforms, to games and supply chain applications. While the original Ethereum software did not necessarily take the needs of enterprises into account, Besu originated as a way to implement the Ethereum protocol while using the best practices in enterprise architecture and distributed systems, under a permissive open source license. However, unique among enterprise blockchain architectures, Besu is versatile in that it can be configured for syncing to the public blockchain or to create private, permissioned networks in accordance with the Enterprise Ethereum Alliance specification. The project was made open source in October 2018, with a 1.0 release in April 2019.

The article Why We Rebuilt Ethereum from Scratch provides some of the context and justification behind the project. Simply put, we wrote a new client to build a more enterprise-grade design and architecture, and to lower the barrier to entry for enterprises with a more friendly license (Apache 2.0) and language (Java). Design and architecture decisions have been aimed at clean interfaces and modularity, with the goal of making Besu a platform for others to build on. We have heard stories of enterprises that completed a successful pilot on Ethereum, only to be stopped from going to production because of company policies around OSS licenses. We solved that pain point by releasing Besu under an Apache 2.0 license and smoothed the path for adoption.

We have talked about our motivation for using Java in the past, but it bears repeating that our goals — to create a vibrant ecosystem of knowledgeable enterprise users while building high quality, production-grade software — are helped greatly by using Java. From IT approvals to monitoring and infrastructure tools to the massive community of Java and JVM champions, we want to make this path easier and bring in existing tools whenever possible.

Dependent Projects

There are no projects dependent on Besu in the Hyperledger community. Besu is also not dependent on any other project.

Motivation

PegaSys believes that contributing the Besu codebase to Hyperledger could be mutually beneficial. We believe the introduction of a fully public-chain-compatible project to Hyperledger will help bridge the gap between public and private blockchains, and help create partnerships and integrations between the two. The Besu submission will increase the diversity of blockchain platforms offered. Additionally, PegaSys will be able to assist with ongoing standards work across business blockchain technology within Hyperledger and provide a unique bridge to the Ethereum community, the largest developer community in the blockchain space. Likewise, PegaSys believes collaboration with a top tier open source body like the Linux Foundation and Hyperledger will expose PegaSys to seasoned enterprise developers, along with best practices of running open-source community governance.

Status

We recognize all projects submitting to Hyperledger enter with Incubation status. Besu is a production-ready project with a 1.0 release, production deployments, and a very active community. Our team reviewed the Project Incubation Exit Criteria in detail and believe we fulfill all of the requirements to meet Active status with the exception of being onboarded onto Hyperledger's tools (e.g., GitHub, Rocketchat) and obtaining the CII badge. We will complete these steps shortly after an approval vote by the TSC. Once these few steps are completed, we believe the project would be ready for Active status and we would apply immediately for that designation. We would expect such designation to process swiftly if Hyperledger agrees all criteria are met.

Scope

The Besu Ethereum Client provides a fully compatible mainnet client. It runs on the Ethereum public network, private networks, and public test networks.

Besu's Features include:

1. The Ethereum Virtual Machine (EVM) and block processing tools
2. Transaction and block validation and mining functionality
    a. Proof of Authority: Besu implements the Clique and IBFT 2.0 Proof of Authority consensus protocols. Proof of Authority consensus protocols are used when participants are known to each other and there is a level of trust between them. For example, in a permissioned consortium network.
    b. Proof of Work: Ethhash
3. World state storage -- implemented by using the RocksDB Storage engine.
4. Devp2p networking integration
    a. Eth sub-protocols
    b. Discovery
    c. Eth wire protocol
    d. IBF sub-protocol
5. APIs to interact with the client
    a. HTTP based JSON RPC
    b. Websockets
    c. GraphQL

Also the Besu client is implementing the EEA Client specification. It aims to be compliant once compliance testing is introduced. Currently this involves implementation of:

- Privacy
- Permissioning
- Efficient consensus mechanisms (iBFT2).

What can you do with Besu?

Besu includes a command line interface and JSON-RPC API for running, maintaining, debugging, and monitoring nodes in an Ethereum network. You can use the API via RPC over HTTP or via WebSockets, and Pub/Sub is supported. The API supports typical Ethereum functionalities such as:

- Ether mining
- Smart contract development
- Decentralized application (Dapp) development

This function is also exposed via a Graphql interface.

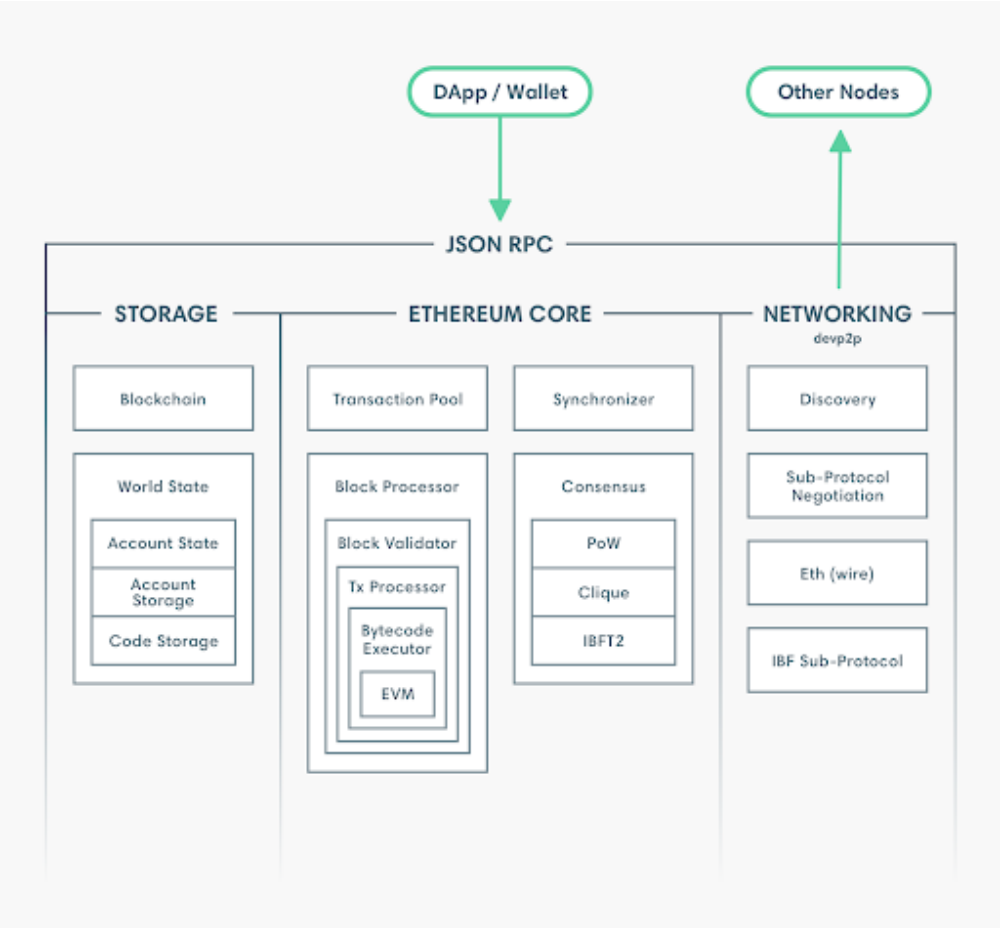
What Does Besu Support?

The Besu client supports common smart contract and Dapp development, deployment, and operational use cases, using tools such as Truffle, Remix, and web3j. The client supports common JSON-RPC API methods such as eth, net, web3, debug, and miner.

Besu doesn't support key management inside the client. You can use EthSigner with Besu to provide access to your key store and sign transactions.

Solution

Below please find the current view of Besu's architecture and what would be included in a submission to the Foundation.



The JSON RPC layers and below in the above diagram are a part of the existing Besu implementation. It has already reached the maturity level of being compatible and working with the Ethereum mainnet, and has been deployed in live environments.

The codebase is Java-based, utilising the gradle build system, with a multi-module based build. Components in the above diagram match the modules and sub-modules in the gradle system.

The enterprise features are included into the above architecture as follows.

- The IBFT2 consensus is configured at runtime and acts as the consensus mechanism. Additionally there are some RPC endpoints added to support the configuration. https://Besu.readthedocs.io/en/stable/Consensus-Protocols/IBFT/
- Privacy cuts across a number of components. There is a customer RPC endpoint, a pre-compiled smart contract, and the world state has separate states for each privacy group. https://Besu.readthedocs.io/en/stable/Privacy/Privacy-Overview/
- The Permissioning function integrates in the networking stack, the API, the transaction pool, and the block processor. It uses a defence-in-depth approach to permission the chain. https://Besu.readthedocs.io/en/stable/Permissions/Permissioning-Overview/

Efforts and Resources

The Pantheon project, to become Hyperledger Besu, has built an extensive and active community around its project. While the PegaSys team has been, and will continue to be, leaders of the project, the external community has played a large role in the project's development. Some data points around the project's community include:

a. The PegaSys team has an engineering team of 26 full-time employees dedicated to Besu's development.
b. There are 28 external contributors outside of the PegaSys team with 111 contributions (as of July 31,2019).
c. There have been over 100 community-raised issues on the project since November 2018, when the project was launched. This demonstrates the engagement and excitement around the project.
d. Dozens of companies building blockchain applications have tested out Besu, provided feedback, and demonstrated its useability for their application. There are several organizations that are deeply integrated in our stack, including web3 labs and iobuilders. These teams are major contributors using Pantheon and providing feedback on the codebase.

Because the project is already in production use and it is anticipated that those production systems move to the Besu code base as soon as releases start happening.

We propose the creation of the following repositories on Github:

- https://github.com/hyperledger/besu (a squashed commit of the current repo https://github.com/PegaSysEng/pantheon)

- https://github.com/hyperledger/besu-docs (a squashed commit of the current repo https://github.com/PegaSysEng/doc.pantheon)

We can confirm we have the CLAs on file for all non-employees of PegaSys. Since we have them, we are able to provide the DCO sign-off and the repo will pass the automated checks. We are planning on creating a copy of the current GitHub repo with the added DCO to avoid squashing commits.

We will also need a space on Jira. There is currently a Jira instance for tracking all of PegaSys's products, mixing public and private issues in a single instance. This makes simply moving the entire current Jira database over very difficult. We propose taking the existing issues and, over a period of time not more than six months but hopefully much less, ensure that all open issues on the Pantheon project within the PegaSys Jira are either closed or manually copied over to the Hyperledger Jira as new issues. All new issues will be on the Hyperledger Jira instance. Happy to discuss further with the TSC.

Currently there is a very active community of over 190 members using Gitter to engage the dev community via chat. We understand the standard at Hyperledger is RocketChat, and so we would like to see a #besu channel started, for answering basic Besu-related questions. We will assess other channels, such as #besu-committers for the committers to use for core team communications. #besu-docs for work related to documentation creation, and add them as needed. We will also use RocketChat for all direct core conversations. We are assessing whether we will continue to keep the Gitter channel or not for continuity purposes. We would welcome the TSC's advice on this matter.

Additionally, we would like an email mailing list too, standard naming convention, for a mix of questions and dev discussions.

FAQ

How Does Besu Submitting Align with Other Initiatives: As mentioned above, PegaSys and ConsenSys are already highly involved in the Ethereum community, including being active collaborators with the Ethereum Foundation and members of the Enterprise Ethereum Alliance. PegaSys knows how to work through standards across organizations because of this experience. This aligns with Hyperledger's mission of driving coherence across the various blockchain platforms. ConsenSys also has contributors working on the Trusted Component project proposal which is also being submitted to the Hyperledger. The team is keen to continue this work within Hyperledger community.

Cross-Platform Integration: One of the main priorities of submitting Besu to Hyperledger is finding synergies and opportunities to collaborate with other projects at Hyperledger, both for greater efficiency (reusing libraries and components), to add functionality we otherwise wouldn't be able to get to, and to help customers trying to use different Hyperledger frameworks together. To further that, we plan to follow closely the activities of Transact and Ursa, by having developers that follow what's happening on each and looking for every opportunity to integrate and collaborate. We also will look at how Aries and Indy can enhance our ability to meet the identity management needs of our users. We will talk with Silas and other Burrow developers about how we might help each other be increasingly valuable Ethereum ecosystem partners. And finally we'll engage with the developers across Hyperledger through participation on TSC calls and lists, the different Working Groups, and at Hyperledger events. We will not silo our activities in any way. We view active engagement as critical to our own success within Hyperledger, and hopefully it will bring value to others in the Foundation.

What does it mean for Besu to be an Ethereum client running on mainnet? To run on mainnet, Ethereum clients must implement specific features and comply to certain standards. Ethereum clients have standards to ensure clients can run on mainnet successfully. The Besu project currently participates in these discussions to ensure compatibility with mainnet requirements. The Enterprise Ethereum Alliance also has a set of specifications which Besu can comply to. Unlike the mainnet standards, these specifications are not all strict requirements. This project will continue to remain compatible with mainnet Ethereum and continue to comply with the Enterprise Ethereum Alliance specification at its discretion.

Closure

PegaSys and ConsenSys believe submitting Besu to Hyperledger is an exciting proposition to both organizations and we hope you consider our proposed submission thoughtfully.

Reviewed By

- [x] Arnaud Le Hors
- [x] Baohua Yang
- [x] Binh Nguyen
- [x] Christopher Ferris
- [x] Dan Middleton
- [x] Hart Montgomery
- [x] Kelly Olson
- [x] Mark Wagner
- [x] Mic Bowman
- [x] Nathan George
- [x] Silas Davis