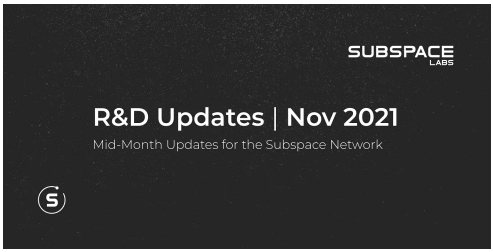
R&D Updates | Nov 2021 - Subspace Network

By Subspace Network

Source: https://blog.subspace.network/r-d-updates-nov-2021-7acc86f2a100

Welcome to the third edition of our monthly R&D updates for the Subspace Network!



Our engineering team has been working like crazy to bring several new features to our test network.

🔗 Subspace Blockchain Client → https://github.com/subspace/subspace

A highly-customized substrate node using our novel Proof-of-Archival-Storage (PoAS) consensus algorithm.

Merged 38 PRs → https://github.com/subspace/subspace/pulls?q=is%3Apr+is%3Aclosed

- Deployed our novel *Proof-of-Archival-Storage (PoAS)* consensus algorithm to the Aries Testnet → https://polkadot.js.org/apps/?rpc=ws

s%3A%2F%2Ftest-rpc.subspace.network#/explorer, allowing us to identify and resolve a host of bugs and performance issues.
- Added a notion of Subspace Credits (our coin) and basic issuance, with the smallest unit being a Shannon → https://en.wikipedia.org/wiki/Claude_Shannon. A vesting pallet for the team has also been implemented.
- Refactored our consensus code to be more readable and in-line with best practices in the Substrate ecosystem; established the first version of our Rust documentation. → https://subspace.github.io/subspace/

Next Steps — Implement a proof-of-concept of decoupled execution framework.

🖇 Subspace Relayer → https://github.com/subspace/sloth256-189

A standalone Node.js app that archives every block of the Kusama relay chain and all parachains on the Subspace Network.

Merged 28 PRs → https://github.com/subspace/subspace/pulls?q=is%3Apr+is%3Aclosed *(backend) + 3 PRs →* https://github.com/subspace/subspace-relayer *(front-end)*

- The Kusama Relayer is running in archival mode on our devnet → https://aries-dev-relayer.subspace.network/?rpc=wss%3A%2F%2Faries-test-rpc-b.subspace.network and will soon be deployed to our testnet → https://testnet-relayer.subspace.network/. Be on the lookout for several announcements on this topic!
- Resolved a variety of performance issues across the full stack of our substrate-client, relayer, and Polkadot-js, leading to over 10x better upload times from genesis.
- Significantly improved our backend infrastructure, we can now support multiple dev and test environments, with more auto-

mated deployments and smoother transitions.

Next Steps — Deploy to Aries Testnet, extend for new parachains and the Polkadot Network.

👩‍🌾 Subspace Farmer Protocol → https://github.com/subspace/sloth256-189

A heavily optimized disk-based plotter and distributed storage node.

Merged 5 PRs → https://github.com/subspace/sloth256-189/pulls?q=is%3Apr+is%3Aclosed

- Completed the refactoring of the farmer codebase from a binary to a library structure, abstracting plotting and farming so they are more composable for external applications.
- Extended the farmer with a basic RPC server, allowing for pieces of the blockchain history and archived objects to be retrieved over the network through a simple API.

Next Steps — Create an initial farmer K-DHT network and begin the transition from syncing a new node from genesis, which currently uses the substrate client block database, to instead use the encoded history stored by farmers.

🧠 Research Topics → https://subspace.network/technology

Our ongoing efforts to explore and incorporate the latest and greatest ideas from industry and academia into our architecture, including collaboration with third-party partners.

- Security Research Labs → https://www.srlabs.de/, the same team that audits Polkadot, is scheduled to conduct a full audit of our

substrate-based implementation of Subspace before Mainnet launch.

- Initiated a new contract with Supranational → https://www.supranational.net/ to prepare an AMD compatible GPU plotter and explore the ASIC resistance of our proof-of-time and proof-of-replication.
- Completed an initial review of our decoupled execution framework with our research advisors and validated the feasibility of an initial implementation in substrate.

Next Steps — Formally describe our vertical and horizontal scaling architecture.

🖖 Thanks for reading and don't hesitate to reach out if you have any questions or ideas!