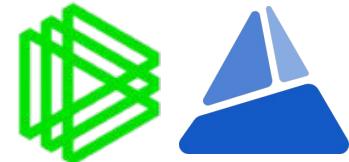


Reth 1.0

The past, present, future of building nodes.

*Georgios Konstantopoulos
gakonst*



Nodes are the soul of every network.

Nodes are perceived as:

- **Slow.**
- **Rigid.**
- **Scary.**
- **Expensive.**

Nodes must be:

- **Fast.**
- **Extensible.**
- **Welcoming.**
- **Cheap.**

We built Reth to fix this.

Reth is a contributor-friendly implementation of Ethereum

The screenshot shows the GitHub repository page for `paradigmxyz/reth`. The main interface displays a list of recent commits from the `main` branch. Key commit details include:

- `mattse perf: spawn eth orchestrator as regular task (#10113)` - 443f7d5 · 1 hour ago · 6,358 Commits
- `.cargo fix: bump stack size on Windows (#5301)` - 9 months ago
- `.config Breaking changes (#5191)` - 6 months ago
- `.github chore(deps): bump EmbarkStudios/cargo-deny-action fr...` - 2 hours ago
- `assets Add the Sigma Prime audit (#8728)` - 2 months ago
- `audit chore: update audit doc to v2 (#9177)` - 2 months ago
- `bin chore(net): Move zeth_network::NetworkEvents into zet...` - 4 hours ago
- `book implement part of reth p2p rlpx ping (#9762)` - 3 days ago
- `crates perf: spawn eth orchestrator as regular task (#10113)` - 1 hour ago
- `docs doc: replace expired example (#9973)` - 5 days ago
- `etc chore(grafana): re-order prune stages (#9511)` - 3 weeks ago
- `examples chore(net): Move zeth_network::NetworkEvents into zet...` - 4 hours ago
- `testing feat(primitives): k256 crate fallback for secp256k1 modul...` - 3 days ago
- `.codespellirc chore(github): use codespell to inspect and correct spel...` - 4 months ago
- `.dockerrignore fix: dockerrignore comment (#4971)` - 10 months ago
- `.editorconfig chore(editor): set indent size=2 for the yaml files (#8366)` - 3 months ago
- `.gitattributes docs: improve CLI reference generator (#5740)` - 8 months ago
- `.gitignore book: remove non exist files links (#9360)` - last month
- `CONTRIBUTING.md Add commands from CONTRIBUTING.md to Makefile an...` - 7 months ago
- `Cargo.lock feat: enable asm-keccak on more platforms (#10101)` - 3 hours ago
- `Cargo.toml chore: sort workspace Cargo.toml entries (#10022)` - 3 days ago
- `Cross.toml fix: tell jemalloc to use 64-KiB pages when cross buildin...` - 5 months ago
- `Dockerfile chore: install apt-get deps before cargo deps (#8262)` - 3 months ago
- `Dockerfile.cross chore: change metrics default port from 9000 to 9001 (#...` - last year
- `DockerfileOp.cross fest(c): add op-reth to release workflow (#8132)` - 3 months ago

The repository has 172 branches and 38 tags. The `About` section describes it as a "Modular, contributor-friendly and blazing-fast implementation of the Ethereum protocol, in Rust". It includes a link to `reth.rs/` and tags for `rust`, `distributed-systems`, `modular`, `ethereum`, `blockchain`, `p2p`, `contributor-friendly`, and `execution-layer`. The repository has 3.6k stars, 963 forks, and 54 watching. It uses Apache-2.0, MIT, and ISC licenses. The `Releases` section shows `Reth v1.0.3` as the latest release, updated 3 weeks ago. There are also 36 other releases. The `Packages` section lists `reth` and `op-reth`. The `Contributors` section shows 336 contributors, with 322 more listed. The `Languages` section indicates the code is written in Rust (99.6%), Makefile (0.2%), Python (0.1%), and Mermaid (0.1%).

Reth is:

- **Fast**
- **Extensible.**
- **Contributor-friendly.**
- **Cheap to operate.**

**Over the last 2 years, we've been
shipping.**

Reth started in 2022, released in 2023...

The screenshot shows a web browser window with the Paradigm logo at the top. The main content is titled "Introducing Reth". Below the title is a date: "Dec 07, 2022 | Georgios Konstantopoulos". A "Contents" section lists 6 numbered items. At the bottom is a dark green image of a circuit board with a central component labeled "Reth Made in Rust".

We're excited to announce **Reth**, a free, open-source Ethereum execution layer client built by **Paradigm**. In this post, we'll discuss why we are making Reth and what to expect from us in the future.

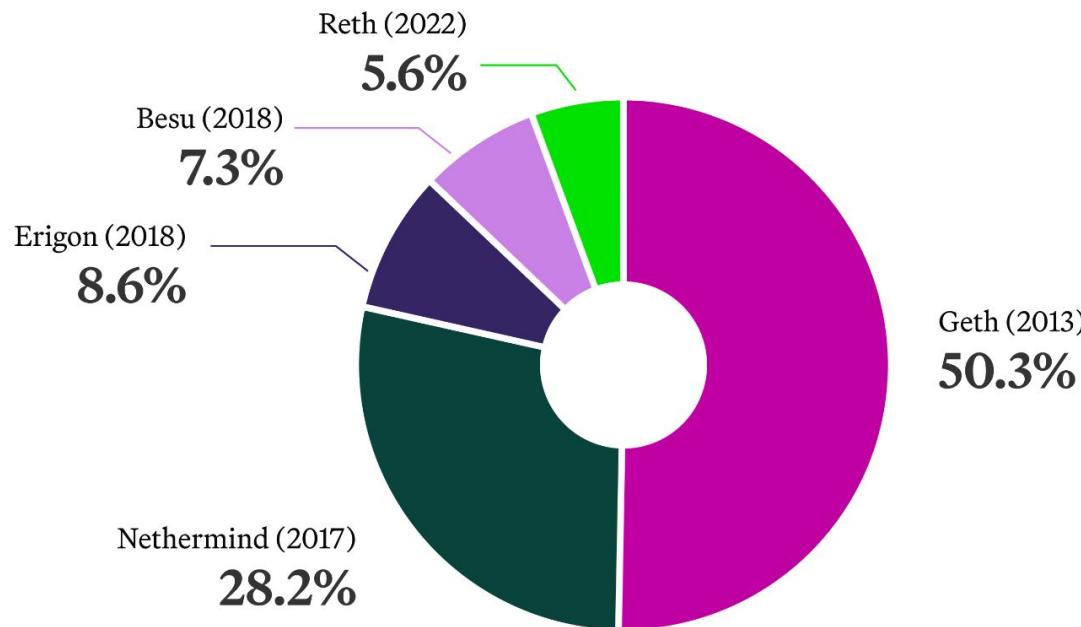
The screenshot shows a web browser window with the Paradigm logo at the top. The main content is titled "Releasing Reth!". Below the title is a date: "Jun 20, 2023 | Georgios Konstantopoulos". A "Contents" section lists 11 numbered items. At the bottom is a dark green image of a circuit board with a central component labeled "Reth Made in Rust".

In 2024, Reth was audited & had its first stable release...

The screenshot shows the Paradigm website at paradigm.xyz. The header includes the Paradigm logo, navigation links for About, Portfolio, Opportunities, Open Source, Team, Writing, and Contact. The main title is "Releasing Reth 1.0!". Below it, a timestamp says "Jun 26, 2024 | The Reth Team, Georgios Konstantopoulos". A "Contents" section lists numbered items: 1. Reth is ready for production, 2. How stable is Reth 1.0?, 3. How secure is Reth 1.0?, 4. OK, I'm sold! How do I install Reth?, 5. Collaborate with the Reth team!, and 6. Where do we go from here?. A large image of the Reth logo is displayed, followed by a text box containing the Reth 1.0 logo and the text: "Reth is now feature-complete, stable, and recommended for production usage & staking." At the bottom, a note states: "We are excited to announce that after almost 2 years of development and a successful audit by Sigma Prime, we are finally releasing Reth 1.0, the first 'prod-ready' release of our blazing-fast Ethereum execution client."

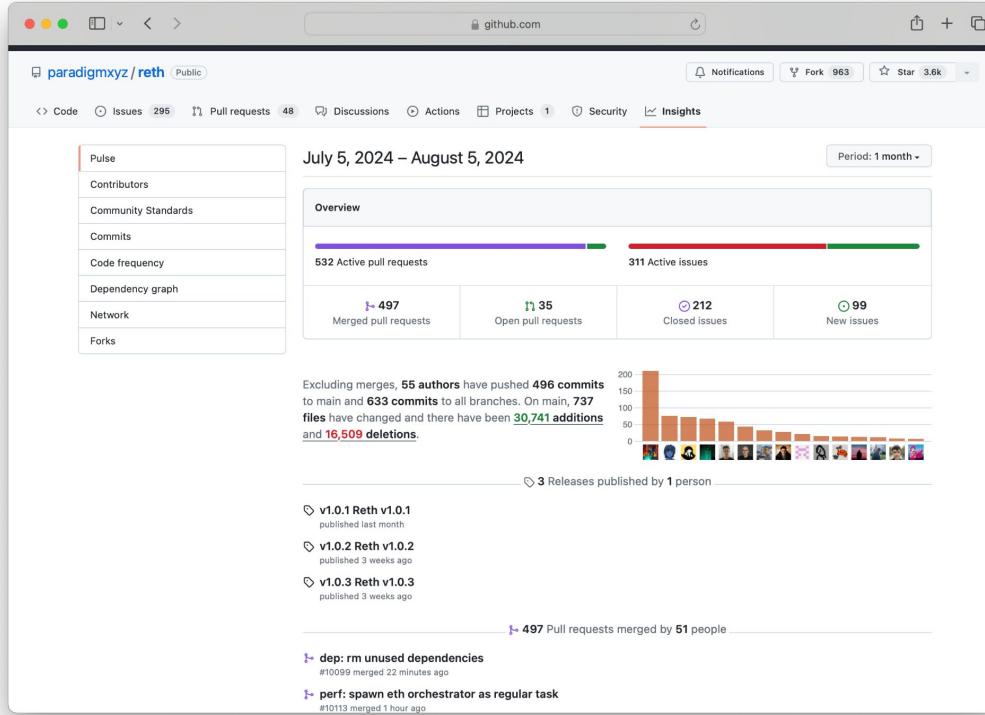
The screenshot shows a browser window titled "sigma_prime_audit_v2...". The page features the Sigma Prime logo (a stylized sigma symbol) and the word "sigma prime". In the top right corner, there is a "PARADIGM" watermark. The main content is the "Reth – Ethereum Execution Client Security Assessment Report" version 2.0. The date "June, 2024" is visible at the bottom right.

... is now being adopted in Ethereum mainnet



Source: Ethernets Node Crawler
Aug 2024

...and it's evolving fast! ~500 PRs, ~50 contributors, ~30 days



We care for Open Source developers.

Contributors 402



+ 388 contributors

Languages



● Rust 99.7%

● Makefile 0.2%

Open Source means community:

- In the eyes of thousands of people, every bug is shallow.
- Enable 3rd party builders.
- Give the project life.
- Move fast.
- Hire the best.

How to build community?

- **Polished issue tracker.**
- **Great docs for newbies.**
- **Great tests to move fast w/o breaking.**
- **Responsive & mentoring maintainers.**
- **Good vibes.**
- **Lively chatroom, zero tolerance for non-tech spam, trolling or ads.**

We learned a lot:

- Ethereum is hard. s/o other client teams.
 - Testing is hard. s/o EthPandaOps.
 - Benchmarking is hard. s/o Samplify.
 - Contributor friendliness is hard. s/o Rust.
- ..but we made it through!*

So what's the future?

The Future of Reth is:

- **Stable**...via *Ethereum Core Development*.
- **Performant**...via *OP Stack & Gigagas*.
- **Extensible**...via *Reth SDK & ExEx*.

Stable = Cross-client Tests & Benchmarks

The image displays two side-by-side GitHub repository pages. The left repository is `ethpandaops/ethereum-package` and the right is `ethpandaops/optimism-package`. Both repositories are public and have been updated recently.

ethpandaops/ethereum-package (Left):

- Code:** 13 Watchers, 146 Forks, 253 Stars.
- Issues:** 28 open.
- Pull requests:** 9 open.
- Actions:** 1 open.
- Projects:** 1 open.
- Code tab:** Shows a list of commits from `EthPandaOpsBot` and other contributors. Notable commits include "feat: add rbuilder remove ..." and "feat: migrate from kurtos...".
- Tags:** `testing`, `docker`, `kubernetes`, `simulation`, `ethereum`, `blockchain`, `geth`, `lighthouse`, `kurtosis`, `starlark`, `mev`, `flashbots`, `reth`, `mev-boost`, `mev-bot`, `kurtosis-package`.
- Readme:** Available.
- MIT license:** Available.
- Activity:** Last month.
- Custom properties:** 253 stars, 13 watching, 146 forks.
- Releases:** 22 releases, including `v4.4.0 (Latest)` and `+ 21 releases`.
- Packages:** No packages published.

ethpandaops/optimism-package (Right):

- Code:** 3 Unwatchers, 24 Forks, 39 Stars.
- Issues:** 8 open.
- Pull requests:** 3 open.
- Actions:** 1 open.
- Projects:** 1 open.
- Code tab:** Shows a list of commits from `kikvr` and other contributors. Notable commits include "fix: handling of config optio..." and "feat: add additional confi...".
- Tags:** `.github`, `src`, `static_files`, `.editorconfig`, `.gitignore`, `.gitpod.Dockerfile`, `.gitpod.yml`, `CHANGELOG.md`, `LICENSE`, `README.md`, `kurtosis-package-icon...`, `kurtosis.yml`, `main.star`, `network_params.yaml`, `rename.sh`.
- Readme:** Available.
- MIT license:** Available.
- Activity:** Last month.
- Custom properties:** 39 stars, 3 watching, 24 forks.
- Report repository:** Available.
- Releases:** 1 release, `v1.0.0 (Latest)` on Jun 17.
- Packages:** No packages published. Publish your first package.
- Contributors:** 12 contributors.
- Languages:** Starlark 94.2%, jq 4.5%, Shell 1.3%.

Performance = L2 + Gigagas & beyond

reth ships with the [optimism](#) feature flag in several crates, including the binary, enabling support for OP Stack chains out of the box. Optimism has a small diff from the L1 EELS, comprising of the following key changes:

1. A new transaction type, `0x7E (Deposit)`, which is used to deposit funds from L1 to L2.
2. Modifications to the `PayloadAttributes` that allow the `sequencer` to submit transactions to the EL through the Engine API. Payloads will be built with deposit transactions at the top of the block, with the first deposit transaction always being the "L1 Info Transaction."
3. EIP-1559 denominator and elasticity parameters have been adjusted to account for the lower block time (2s) on L2. Otherwise, the 1559 formula remains the same.
4. Network fees are distributed to the various `fee vaults`.
5. ... and some other minor changes.

For a more in-depth list of changes and their rationale, as well as specifics about the OP Stack specification such as transaction ordering and more, see the documented [op-reth diff](#), the [L2 EL specification](#), and the [OP Stack specification](#).

Running on Optimism

You will need three things to run `op-reth`:

1. An archival L1 node, synced to the settlement layer of the OP Stack chain you want to sync (e.g. `reth`, `geth`, `besu`, `nethermind`, etc.)
2. A rollup node (e.g. `op-node`, `magi`, `hldr`, etc.)
3. An instance of `op-reth`.

For this example, we'll start a `Base Mainnet` node.

Installing op-reth

To run Reth on Optimism, first install `op-reth` via the `Makefile` in the workspace root:

Reth's path to 1 gigagas per second, and beyond

Apr 24, 2024 | Georgios Konstantopoulos

Contents

- 1 Are we scaled yet?
 - 1 How do you measure performance? What does gas per second mean?
 - 2 Where are we today?
- 2 How will Reth get to 1 gigagas per second? Beyond that?
 - 1 Reth's vertical scaling roadmap
 - 1 Just-In-Time & Ahead-of-Time EVM
 - 2 Parallel EVM
 - 3 Improving the State Commitment
 - 2 Reth's horizontal scaling roadmap
 - 1 Multi-Rollup Reth
 - 2 Cloud-native Reth
 - 3 Outlook

We started building Reth in 2022 to provide resilience to Ethereum L1, and solve

Extensibility = Reth SDK + ExEx

```
reth git:(main) x tree examples/ -d -L 1
examples/
├── beacon-api-sidecar-fetcher
├── beacon-api-sse
├── bsc-p2p
├── custom-dev-node
├── custom-engine-types
├── custom-evm
├── custom-inspector
├── custom-node-components
├── custom-payload-builder
├── custom-rlpx-subprotocol
├── db-access
├── manual-p2p
├── network
├── network-txpool
├── node-custom-rpc
├── node-event-hooks
├── polygon-p2p
├── rpc-db
├── stateful-precompile
└── txpool-tracing

21 directories
> reth git:(main) x
```



```
Alacrity
> reth-exex-examples git:(main) tree -d -L 1
.
├── backfill
├── discv5
├── in-memory-state
├── minimal
├── op-bridge
├── remote
└── rollup

8 directories
> reth-exex-examples git:(main)
```

2024-08-05 20:49 gak-9

Paradigm

About Team Portfolio Opportunities Contact

Reth Execution Extensions

May 03, 2024 | Georgios Konstantopoulos

Contents

- 1 How do we build off-chain infrastructure today?
- 2 Introducing Reth Execution Extensions (ExEx)
 - 1 How do ExExes work?
 - 2 Hello ExEx!
 - 3 Building an indexer for the OP Stack using ExEx
 - 4 Building a Rollup using ExEx
- 3 What can I build with Execution Extensions?
- 4 What's next for Execution Extensions?

Reth is an all-in-one toolkit for building high performance and customizable nodes. We recently published our [performance roadmap](#) for improving Reth's performance >100x, and [Reth AlphaNet](#), our testnet rollup for pushing Reth's modularity and extensibility to the limits.

Today, we are excited to announce Reth Execution Extensions (ExEx). ExEx is a framework for building performant and complex off-chain infrastructure as post-execution hooks. Reth ExExes can be used to implement rollups, indexers, MEV bots and more with >10x less code than existing methods. With this release, we demonstrate from scratch a prod-ready reorg tracker in <20 LoC, an indexer in <250 LoC and a rollup in <1000 LoC.

ExEx was co-architected with [init4](#), a research collective building next-generation Ethereum infrastructure. We look forward to continuing to collaborate with the init4 team as we make Reth the #1 platform for building crypto infrastructure!

How do we build off-chain infrastructure

2022: Reth L1 node

2023: Reth L1 + L2 node

**2024+: Reth SDK for building high
performance & extensible nodes.**

We will accelerate crypto,

ithaca

Crypto must go faster. We have started Ithaca, a company designed to accelerate the crypto frontier, and have raised \$20M from Paradigm.

Over the past four years, our global group of less than 20 engineers has shipped some of the industry's most used open source tools like Reth & Foundry. We believe in sustainable open source development for the good of crypto, and in creating the right incentive system for commercial consumer experiences.

The stack is ready. We can achieve the step-level changes needed to enable consumers to fully leverage crypto with fast, cheap, global and private experiences.

Our first step in that mission is [Odyssey](#), an open-source Layer 2 testnet from the future, built with Reth, the OP Stack and Conduit. You can [try it out today on Sepolia](#) and start building with its [unique features](#).

We have been collaborating with the developer community and the industry for the last years, and we are excited to continue accelerating crypto together.

Get in touch [here](#). If you're looking to join our team, see our [hiring page](#).

As you set out for Ithaca
hope your road is a long one,
full of adventure, full of discovery.

Laistragonians, Cyclops,
angry Poseidon--don't be afraid of them:
you'll never find things like that on your way
as long as you keep your thoughts raised high,
as long as a rare excitement
stirs your spirit and your body.

— [Cavefy, 1-9](#)

Ithaca on X: "Hello EXP-0001."

Post

Ithaca @ithacaxyz Hello EXP-0001.

Today, we're releasing a deep dive on how Ethereum account delegation with EIP-7702 works, a spec for third parties looking to integrate, and a forkable codebase that you can extend to build next-generation wallets.

And, of course, a demo!

EXPERIMENT

EXP-0001

Account Delegation with EIP-7702

ithaca 0:32

ITHACA.XYZ

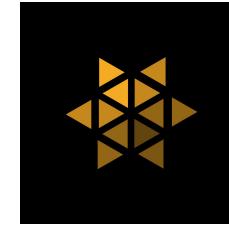
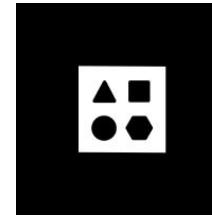
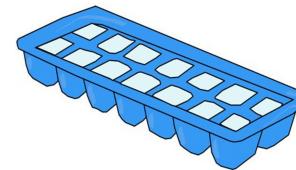
11:30 PM · Oct 29, 2024 · 37.4K Views

14 41 184 73

Post your reply Reply

Ithaca @ithacaxyz · Oct 29 EXP-0001 explores bringing enhanced functionality to Ethereum EOAs with:
- Contract Delegation via EIP-7702
- Passkey Authorization via RIP-7212
- Sequenceless on-chain Transaction via "Indirectly Signed Transactions"

How? Full-stack Ethereum for L1 & L2s.



Our north star:
Solve Ethereum's hardest problems.

thank you. let's build.

Georgios Konstantopoulos
gakonst

