Summary

- Nomic Labs, the team behind Hardhat, has become the Nomic Foundation, a non-profit organization dedicated to Ethereum. Our mission is to empower developers to decentralize the world.

- The Nomic Foundation's work will be focused on Ethereum's developer platform with the objective of achieving a world-class developer experience, and generally improving Ethereum's public goods support structures.

- [Hardhat](#) is the de facto standard developer tool used to build Ethereum software, with more than 23000 Github repositories using it and tens of thousands of active users. Prominent teams relying on it include ENS, Uniswap, Optimism, OpenZeppelin, Aave, Balancer, Chainlink, Synthetix, and many more leading teams.

- The new foundation will expand the Hardhat suite of tools and, most importantly, build long-term infrastructure to catalyze organic growth in the Ethereum tooling ecosystem, decreasing Ethereum's dependence on any one organization to build and maintain core development platform components.

- Seeking $30m in total funding from the ecosystem. Donations of $15M already secured by the Ethereum Foundation, Vitalik Buterin, Coinbase, a16z, The Graph, Polygon, Chainlink, a16z, and Kaszek Ventures.

- We're proposing to Uniswap Governance to make a contribution of $5m to the Nomic Foundation to support its mission.

What is developer experience?

If you are already familiar with this concept feel free to skip this section, if you'd like to learn more, keep on reading.

Understanding developer experience starts with realizing that software development platforms are multi-layered technical products and that software engineers are their users. Developer experience is analogous to UX for software developers, particularly UX in professional tools that are essential for abstract tasks and have direct impacts on productivity, effectiveness, happiness, frustration, and therefore cost.

The practice of software development is a complex discipline, and developer experience is significantly different from the UX of a note-taking app. When a professional is working on an intricate problem that requires deep thinking and accessing information via a tool, or when one's creative output is directly limited by the capabilities of their tool, then the professional will be directly restricted—or empowered—by the tools being used.

An additional aspect that may not be obvious and emphasizes the impact tooling has, is that developers frequently don't understand what they've created. Writing code that doesn't behave as expected is routine, and the process of making it work as expected (debugging) is one of the core activities of software development, which is heavily aided by development tools.

Developer experience is about the effectiveness of interactions with platforms and their tools, as well as the feelings that arise when trying to meet a specific software development objective with them.

Empowering users to create something without knowing what they're going to be creating is a difficult challenge. The author of a tool or platform cannot optimize for a specific use case, only for the abstract needs of software development in general. In addition, the development platform that software is built on ultimately becomes part of the software itself, and very often, the software directly interacts in some way with the tools.

Since the created software is dynamic and has a limitless life of its own, it's easy for tools and platforms to end up constraining what is created or how. Software development, in general, is a moving target that keeps evolving. Tools and platforms need to accommodate a future that is partially unknown, making achieving a solid developer experience a challenge.

Some aspects that contribute to a good developer experience include:

- Ability to get up and running quickly as a beginner

- Clear, thorough, and detailed documentation that's easy to explore and facilitates learning

- Accessible developer community as a support resource

- Freedom and flexibility to take the preferred path towards a solution without rigid obstacles

- Ability to easily gather relevant technical information related to code failure

- Clear and explicit error messages that explain issues and suggest potential solutions

- Automatic resolution of tedious automatable tasks

- Reusable infrastructure to easily build ad-hoc solutions

- Availability of specialized tools for domain-specific problems (local environments, programming editors, debuggers, etc)

- Fast performance

Developers are empowered to meet their objectives with a high degree of comfort and efficiency when there are high-quality resources available, well-designed architectures with APIs that get out of the way, and insightful tooling functionalities to address domain-specific problems.

When the basic tenets of developer experience aren't met, developers feel frustrated and suffer significantly lower productivity.

Ethereum developer experience

When it comes to Ethereum, which is primarily a software development platform to build decentralized systems, developer experience is a key strategic aspect for success. Ecosystem growth requires more developers to build more software on top of Ethereum. Developer adoption and learning speed, core contributors to this growth, are critically affected by developer experience.

The rate at which the ecosystem innovates, coming up with new creations and solving difficult problems, both at the dapp layer and EVM/Solidity/Vyper layer, is also directly affected by developer productivity.

Software development platforms aren't new, and playbooks established by the great developer experience success stories (Rust, .Net, TypeScript, etc) prove that achieving a quality developer experience requires a specialized approach paired with a long-term, big-picture strategy. The potential impact in executing a dedicated effort for Ethereum would increase the ecosystem's pace of innovation and growth, building a powerful compounding effect over the long term for the entire industry.

The inspiration for our vision came from our experience building Hardhat, which allowed us to see how deeply challenging it is to build sophisticated Ethereum tooling. These challenges must be alleviated to bring about organic ecosystem-led improvement of developer experience that achieves world-class quality.

Nomic Foundation

Nomic Labs has been fully dedicated to Ethereum developer experience since 2019, and we're now pivoting to a non-profit foundation formally dedicated to Ethereum. We're aiming to build a long-lasting organization that makes Ethereum's public goods support structures stronger by contributing to the Ethereum Foundation's existing efforts, and reducing the ecosystem's reliance on any one organization for development platform components.

Roadmap

Given the size and innovation pace of the ecosystem, there's no way to foresee exactly what needs developers are going to have as things scale. However, we do know what engineering foundations the ecosystem will need in order to build its own solutions.

Our overarching engineering strategy is to empower the ecosystem to build its own specialized tools. This plan is based on four strategic pillars of the stack, each of which offers an opportunity to leverage a platform to empower the ecosystem to keep building open-source infrastructure.

For each of these pillars, we will build a platform. The four ecosystem pillars and platform opportunities we've identified are:

1. Solidity

2. EVM tooling

3. Local development environment

4. Ethereum connector library

The projects

Slang & Rethnet

Over the long term, these are our most important projects. Targeting the Solidity and EVM tooling pillars, Slang and Rethnet will serve as core infrastructure for the ecosystem to build new tools faster, cheaper, and better. We're essentially building the tools that would have let us build Hardhat a lot faster. We previously published a Medium post with high-level descriptions of how both projects will complement each other.

Slang

A Solidity compiler designed as a platform for tooling development, an approach also known as compiler as a service. Its top priority will be servicing tools through domain-specific APIs. Much like .Net's Roslyn, it will feature a compilation pipeline made of distinct reusable components with standalone APIs. A completely modular design guarantees that others can build on top of it by replacing the part of functionality they need to, and reusing everything else:

1. Parser that is only concerned with producing trees from code. Usable on its own, for example, to create third-party formatters like Prettier plugins.

2. Semantic analysis (binding) is concerned only with building a type system and validating the produced trees. Usable on its own, to implement third-party type checking, security/threat models, and more advanced third-party linting.

3. Code generation. By replacing just this isolated part, the compiler can compile for different targets (e.g. non-EVM L2s).

4. Language services. These will receive an immutable representation of the above (syntax trees, bound trees, codegen settings), and will only be tasked with answering questions. Usable on its own to expose in different IDEs (same service for VSCode, IntelliJ, Vim, etc). Reusable to extend the functionality of other editor features (task runners, testing, deployment, CI, debugging).

5. Runtime observation APIs to support Rethnet.

All of this will be reusable to create entirely new EVM programming languages, since by replacing the parser and type system, one can get an entire high-quality toolchain working from the get-go.

Rethnet

To provide a simulated environment where developers can build and test their Ethereum software, tools need to replicate many of the components that make up a full Ethereum node implementation. This is a significant engineering effort, which given the complexity of Ethereum, represents a barrier to entry to tooling development given the depth of knowledge that is required.

Rethnet aims to make this easier by offering a native, flexible, extensible, fast, and language-agnostic EVM local development network, distributed as a Rust library, that is designed to be the underlying core in tools that provide debugging information to developers (like Hardhat, Foundry, Remix, Truffle, DappTools, etc). It will be a Rust library made to be consumed from other languages like TypeScript, Go, Python, etc as a native dependency. It will implement the baseline of essential functionality every tool should have like Solidity console.log, stack traces, and descriptive error messages, as well as implement code coverage, gas profiling, and a step debugger. At its core, it's an implementation of an Ethereum node with a layer of EVM runtime observation to provide development features.

Building a new Hardhat, Truffle, Remix, or DappTools using Rethnet will be a much more manageable project, and Rethnet will be completely reusable for any EVM language through adapters.

Hardhat

Our flagship project targets the local development environment pillar, and it's currently at an advanced level of progress and adoption. While Slang and Rethnet mature and catalyze organic growth in the tooling space, developers still have needs to be met, positioning Hardhat as our immediate-term solution to empower developers to keep decentralizing the world.

Hardhat is an Ethereum development environment that developers use to compile, deploy, test, and debug Ethereum software. Most importantly, it's highly flexible, extensible, and designed to empower the community to build their own solutions. This strategy has been successful, and there's already [a valuable ecosystem of reusable plugins](#).

Hardhat's roadmap is focused on becoming an extensible development environment with deep integrations across components in key areas of the tooling stack:

- [Hardhat VSCode](#) — programming editor

- [Hardhat Ignition](#) — contract deployment

- [Hardhat Network](#) — local development network

- [Hardhat Runner](#) — build/testing workflow

This roadmap leads to developers being well equipped to build powerful extensions to their workflow that increase their productivity according to their exact needs, and to then share them with the ecosystem in the form of plugins.

Hardhat will also eventually migrate to using Rethnet and Slang, increasing its feature richness, speed, and stability while enabling dogfooding at scale for our brand new building blocks.

Web3.js as a frontend platform

The OG Ethereum connector library, Web3.js, is being revitalized into a high-value project. By focusing on community and

ecosystem growth, supported by an extensible architecture, it can become a great source of value, much like React represents in the front-end world, but for dapps. A website hub connecting community spaces, support spaces, educational resources, extensions, and related projects, combined with an active ecodev effort (workshops, talks, contests, and incentives), will create a source of leverage for the ecosystem. This will provide better troubleshooting, faster developer training, more reusable code, and, most importantly, the possibility of extending the library. This effort is currently spearheaded by the ChainSafe team.

Funding

The Nomic Foundation aims to benefit the entire Ethereum ecosystem, which is why we're fundraising across multiple organizations and individuals within it.

The Ethereum Foundation is leading this round of contributions with $8M, alongside contributions from Vitalik Buterin, Coinbase, Consensys, The Graph, Polygon, Chainlink, Gnosis, a16z, a_capital, and Kaszek Ventures. These donors make up $15M, and we're aiming to raise $15M more.

Why Uniswap?

Generally, we think that allocating capital to the Nomic Foundation makes strategic sense for any protocol treasury that is aligned long term with the growth of Ethereum, and we've approached and will continue approaching several protocols.

Currently, Uniswap is built using Hardhat. While this is a signal of Hardhat's value, the projects that the Nomic Foundation will deliver will create more value not just for Uniswap, but for the entire ecosystem. We'll provide services to the Ethereum community that will:

1. Continue the maintenance of critical infrastructure used to build most protocols (Hardhat).

2. Increase developer productivity for every team in the ecosystem.

3. Accelerate developer onboarding to Ethereum, increasing the size of the experienced engineering hiring pool and making time-to-productivity shorter for new hires.

4. Accelerate the pace of innovation and the number of products being built.

5. Increase market volume driven by new users and new products.

We believe this grows the market for everyone, including Uniswap, and we'd love to have the Uniswap DAO contribute $5m in funding to this community effort

.

An ecosystem-wide effort

We're currently seeking funding from multiple DAOs. We'll update with the corresponding links below as we create each forum thread.

- Compound forum thread

- ENS forum thread

- Yearn forum thread

- SushiSwap forum thread

Links

On-Chain Proposal

Temperature Check

Consensus Check

Nomic Foundation Announcement