

Design Philosophy & Design Principles

This guide covers design philosophy and design principles for contributing to the OP Stack.

Design Philosophy

OP Stack is built according to a strong design philosophy that stands on four main pillars: simplicity, pragmatism, sustainability, and, of course, optimism. It's important to understand these pillars as they heavily influence the design of Optimism as a whole.

Simplicity

Optimism is designed to be as simple as possible for the featureset it provides. Ideally, Optimism should be composed of the minimum number of moving parts required for a secure, scalable, and flexible L2 system. This simplicity gives Optimism's design a number of significant advantages over other more complex L2 constructions.

Simplicity reduces engineering overhead, which in turn means we can spend our time working on new features instead of re-creating existing ones. Optimism prefers to use existing battle-tested Ethereum code and infrastructure where possible. The most visible example of this philosophy in practice is the choice to use Geth as Optimism's client software.

When dealing with critical infrastructure, simplicity is also security. Every line of code we write is an opportunity to introduce unintentional bugs. A simple protocol means there's less code to write and, as a result, less surface area for potential mistakes. A clean and minimal codebase is also more accessible to external contributors and auditors. All of this serves to maximize the security and correctness of the Optimism protocol.

Simplicity is also important for the long-term vision of Optimism. By limiting the amount of code that we write on top of Ethereum tooling, we're able to spend most of our time working directly with existing codebases. Engineering effort that goes into Optimism can also directly benefit Ethereum, and vice versa. This will only become more pronounced as the Optimism protocol solidifies and existing resources can be redirected towards core Ethereum infrastructure.

Pragmatism

For all its idealism, the design process behind Optimism is ultimately driven by pragmatism. The core Optimism team has real-world constraints, the projects that build on Optimism have real-world needs, and the users that engage with Optimism have real-world problems. Optimism's design philosophy prioritizes user and developer needs over theoretical perfection. Sometimes the best solution isn't the prettiest one.

Optimism is also developed with the understanding that any core team will have limited areas of expertise. Optimism is developed iteratively and strives to continuously pull feedback from users. Many core Optimism features today (like [EVM Equivalence](#) [\(opens in a new tab\)](#)) were only made possible by this iterative approach to protocol development.

Sustainability

Optimism is in it for the long haul. Application developers need assurance that the platform they're building on will remain not only operational but competitive over long periods of time. Optimism's design process is built around the idea of long-term sustainability and not taking shortcuts to scalability. At the end of the day, a scalable system means nothing without the ecosystem that sustains it.

Sustainability actively influences Optimism's protocol design in ways that go hand-in-hand with our philosophy of simplicity. The more complex a codebase, the more difficult it is for people outside of the core development team to actively contribute. By keeping our codebase simple we're able to build a bigger community of contributors who can help maintain the protocol long-term.

Optimism

Of course, none of this would be possible without a sense of optimism. Our optimism about the Ethereum vision keeps this project moving forward. We believe in an optimistic future for Ethereum, a future where we get to redesign our relationships with the institutions that coordinate our lives.

Although Optimism looks like a standalone blockchain, it's ultimately designed as an extension to Ethereum. We keep this in mind whenever we're creating new features or trying to simplify existing ones. Optimism is as close to Ethereum as possible not only for pragmatic reasons, but because Optimism exists so that Ethereum can succeed. We hope that you can see the influence of this philosophy when looking at Optimism's design.

Design Principles for USEful Software

The OP Stack is USEful software. The OP Stack is a set of software components for building L2 blockchain ecosystems,

built by the Optimism Collective to power Optimism. Components to be added to the OP Stack should be built according to three key design principles: Utility, Simplicity, Extensibility. Software that follows these principles is useful software for the Optimism Collective!

Utility

For something to be part of the OP Stack, it should help power the Optimism Collective. This condition helps guide the type of software that can be included in the stack. For instance, a powerful open-source block explorer that makes it easier for users to inspect [the Superchain \(opens in a new tab\)](#) would be a great addition to the OP Stack.

Although utility is important for inclusion in the OP Stack, you shouldn't be afraid to experiment. Do something crazy. Build something that's never been built before, even if it doesn't have any clear utility. Make a blockchain for Emojis, or whatever. Have fun!

Simplicity

Complex code does not scale. Code that makes it into the OP Stack should be simple.

Simplicity reduces engineering overhead, which in turn means the Collective can spend its time working on new features instead of re-creating existing ones. The OP Stack prefers to use existing battle-tested code and infrastructure where possible. The most visible example of this philosophy in practice is the choice to use Geth as the OP Stack's default execution engine.

When dealing with critical infrastructure, simplicity is also security and maintainability. Every line of code written is an opportunity to introduce bugs and vulnerabilities. A simple protocol means there's less code to write and, as a result, less surface area for potential mistakes. A clean and minimal codebase is also more accessible to external contributors and auditors. All of this serves to maximize the security and correctness of the OP Stack.

Extensibility

Good OP Stack code is inherently open, collaborative, and extensible. Collaboration allows us to break out of siloed development. Collaboration allows us to spend more time building on top of one another's work and less time rebuilding the same components over and over again. Collaboration is how we win, together.

Extensible code should be designed with the mindset that others will want to build with and on top of that code. In practice, this means that the code should be open source (under a permissive license), expose clean APIs, and generally be modular such that another developer can relatively easily extend the functionality of the code. Extensibility is a key design principle that unlocks the superpower of collaboration within the Optimism Collective ecosystem.

Contributing to the OP Stack

The OP Stack is a decentralized software stack that anyone can contribute to. If you're interested in contributing to the OP Stack, check out [the Contributing section on our Community pages \(opens in a new tab\)](#). Of course, software that has impact for the Optimism Collective is eligible to receive [Retroactive Public Goods Funding \(opens in a new tab\)](#). Build for the OP Stack — get rewarded for writing great open source software. What's not to love?

[Rollup Protocol Overview](#) [Smart Contract Overview](#)