Hi everyone! This is Butian from the Blockless team. I would like to share with you an open-source middleware that we've been exploring with the EigenLayer team, to be secured by the Restaking Collective. I'm eager to hear your thoughts and feedback, so please don't hesitate to comment below!

TL;DR

- We have developed an edge-native WebAssembly-based Turing-complete compute engine that operates on lightweight nodes, suitable for everyday devices like personal computers.
- Blockless offers zkServerless executions and pioneers the industry's first x86 emulator within a WebAssembly (WASM) runtime. Blockless as an Actively Validated Service (AVS) enables verifiable computing on a trustless network, collectively secured and powered by EigenLayer's restakers and operators.
- The implementation allows L1 blockchains like Ethereum to achieve uncapped throughput, leveraging high-performance verifiable off-chain executions and automatic horizontal scaling with an advanced load distribution algorithm.

Introducing Blockless

For developers, Blockless is a WebAssembly-based verifiable execution platform that enables the creation and execution of serverless functions and x86 programs with near-native machine speed. It guarantees off-chain execution security through pluggable zk and dynamic consensus verification methods. As a community-powred network, the decentralized stack is non-custodial and censorship-resilient.

Blockless provides fully-managed and hands-off deployment, allowing builders to focus solely on writing the business logic. It provides uncapped throughput and ready-to-use function templates to cater to the common needs of decentralized applications (dApps), such as programmable oracle, indexer, automation trigger, and hosting solutions.

In simple terms, Blockless is the solution that handles heavy computing or external-resource-required tasks that blockchains/smart contracts are not capable of executing, without compromising security & decentralization.

How does restaking with Blockless work?

As a permissionless network, Blockless invites EigenLayer restakers to secure and power the network via two options:

1. Native staking: ETH can be restaked on EL to operate or delegate a node.
2. ETH LP staking: ETH-BLS LP tokens can be restaked on EL to operate or delegate a node.

There is no marginal cost for restakers, and the minimum hardware requirement for being an operator is considerably low. In principle, Blockless allows participation from IoT, personal, and server-grade devices. However, for initial network bootstrapping, the minimum hardware requirement spec is set to that of a Raspberry Pi (4 Core 1.5 GHz 64-bit 8GB RAM).

As a node operator, you will be involved in running serverless functions, x86 emulators, generating zk proofs (may require higher-grade hardware

), or participating in application-specific local consensus. However, once the node is set up, tasks are automatically assigned and executed seamlessly in the background without requiring your attention. Blockless incorporates an automated orchestration mechanism that distributes work to suitable nodes across the network. Nodes are selected for execution based on hardware capacity and the resource consumption requirements of the specific task.

Furthermore, Blockless features built-in resource consumption metering. This ensures that restakers and operators can anticipate receiving compensation proportional to their stake and hardware capacity.

What are the benefits of the collaboration?

1. Security guarantee

for restakers, operators, and computing tasks

- There are built-in failover and fault tolerance mechanisms in Blockless' P2P protocol which protect restakers' asset safety - even if their node operators mess up for unintended reasons, their staked ETH or LP will not be slashed - the job will simply get picked up by the next available and most qualified node.
- Any execution failure exceeding the timeout limit will be caught without triggering slashing, even if the bug was omitted from the security audit so that restakers and operators can rest assured with the extra layer of protection.
- A strictly isolated sandbox environment on the host machine prevents operator from tampering with or accessing running process. Furthermore, with end-to-end encryption and MPC extensions, even private keys can be securely handled and parsed as variables, safeguarding sensitive information from unauthorized access or exposure.
- Operators have the ability to specify the maximum hardware resource consumption, eliminating the risk of slashing

due to service overload.

- Portability and high-performance efficiency

for dApps

- 100MB docker image can be 1MB or less in a comparable WASM format. The optimization enables faster deployment and execution of applications in resource-constrained environments.
- WASM runs at near-native speed across devices such as IoTs, smartphones, laptops, and as well as professional bare metal CPUs, GPUs, and cloud servers. Theoretically, any device that can open up a web browser can participate as a network operator.

A bonus

for Web2 devs

- Developers have the flexibility to code in popular programming languages such as Go, Rust, AssemblyScript, and Python, etc.,and achieve immediate interoperability with smart contracts, and other JavaScript programs in the open web ecosystm.

By combining the restaker network and the trustless verifiable compute engine, the collaboration between EigenLayer and Blockless creates a community-driven execution network that is more secure, efficient, and scalable. This collaboration brings benefits to developers, validators, and users, as it enhances the overall reliability, resilience, and performance of the system.

What’s in it for me?

- For restakers: you can expect to earn more profit at zero marginal cost, with an additional guarantee of the security of your restaked assets. You also have full transparency to operators for delegation, giving you more control and confidence in your restaking activities.
- For developers: ship feature-rich and high-performance advanced decentralized applications (dApps) without worrying about the underlying infrastructure. This significantly reduces your go-to-market time, engineering headcounts, and execution risks, allowing you to focus on creating innovative dApps.
- For those who believe in an open web, together we can create a community-driven, community-controlled, and community-powered decentralized stack. One that shapes a collective future and open ecosystem.

What’s next?

- The EigenLayer and Blockless teams will collaborate on the staking requirements and guidelines. We will integrate the restaking and execution modules, and conduct security audits and end-to-end testing. In the spirit of transparency, which both teams uphold, this integration will be developed in public as the Eigenlayer client and contract interfaces emerge.
- The technical documentation can be found here. Need more? Additionalresources about Blockless can be found here.
- Interested in staking your claim, running a node, or developing with this technology? Register hereEigenLayer x Blockless.
- Let us know what you think by replying below! Comments and questions are welcomed and appreciated. As we keep building, community feedback becomes a crucial part of what we’re trying to achieve.