

Axonum Sepolia testnet is undergoing beta testing. Check out our [documentation](#).

# Introducing Axonum: The Brain of Ethereum

Axonum enshrines AI into blockchain to build a decentralized supercomputer powered by global collective intelligence.

## The Age of AI EVM

We are building Axonum, an AI optimistic rollup, with the world's first AI EVM.

We aim to democratize access to AI-powered DApps, making AI model inferences both accessible and user-friendly.

Axonum is an optimistic rollup with enshrined AI powered by opML and AI EVM. It enables users to seamlessly employ AI models natively within smart contracts without being encumbered by the intricacies of underlying technologies.

## Overview

### AI EVM: Enshrined AI

To enable native ML inference in the smart contract, we need to modify the execution layer of the layer 2 chain. Specifically, we add a precompiled contract inference in EVM to build AI EVM.

AI EVM will conduct the ML inference in native execution and then return deterministic execution results. When a user wants to use the AI model to process data, all the user needs to do is to call the precompiled contract inference with the model address and model input, and then the user can obtain the model output and use it natively in the smart contract.

```
import "./AIBLib.sol";
```

```
contract AIBContract { ... function inference(bytes32 model_address, bytes memory input_data, uint256 output_size) public {
bytes memory output = AIBLib.inference(model_address, input_data, output_size); emit Inference(model_address,
input_data, output_size, output); } }
```

The models are stored in the model data available (DA) layer. All the models can be retrieved from DA using the model address. We assume the data availability of all the models.

The core design principle of the precompiled contract inference follows the design principles of opML, that is, we separate execution from proving. We provide two kinds of implementation of the precompiled contract inference. One is compiled for native execution, which is optimized for high speed. Another is compiled for the fraud proof VM, which helps prove the correctness of the opML results.

For the implementation for execution, we re-use the ML engine in opML. We will first fetch the model using the model address from the model hub and then load the model into the ML engine. ML engine will take the user's input in the precompiled contract as the model input and then execute the ML inference task. The ML engine guarantees the consistency and determinism of the ML inference results using quantization and soft float.

Besides the current AI EVM design, an alternative approach to enable AI in EVM is adding more machine learning-specific opcodes into EVM, with corresponding changes to the virtual machine's resource and pricing model as well as the implementation.

### Optimistic Rollup

opML (Optimistic Machine Learning) and optimistic rollup (opRollup) are both based on a similar fraud-proof system, making it feasible to integrate opML into the Layer 2 (L2) chain alongside the opRollup system. This integration enables the seamless utilization of machine learning within smart contracts on the L2 chain.

Just like the existing rollup systems, Axonum is responsible for "rolling up" transactions by batching them before publishing them to the L1 chain, usually through a network of sequencers. This mechanism could include thousands of transactions in a single rollup, increasing the throughput of the whole system of L1 and L2.

Axonum, as one of the optimistic rollups, is an interactive scaling method for L1 blockchains. We optimistically assume that every proposed transaction is valid by default. Different from the traditional L2 optimistic rollup system, the transaction in Axonum can include AI model inferences, which can make the smart contracts on Axonum "smarter" with AI.

In the case of mitigating potentially invalid transactions, like optimistic rollups, Axonum introduces a challenge period during which participants may challenge a suspect rollup. A fraud-proving scheme is in place to allow for several fraud proofs to be submitted. Those proofs could make the rollup valid or invalid. During the challenge period, state changes may be disputed, resolved, or included if no challenge is presented (and the required proofs are in place).

# Workflow

[

workflow

2443×1437 183 KB

](https://ethresear.ch/uploads/default/original/3X/4/1/41403b199568fcf421dabd3b05c089511968469a.png)

Here's the essential workflow of Axonum, without considering mechanisms such as pre-confirmation or force exit:

1. The basic workflow begins with users sending L2 transactions (we allow native AI inference in the smart contract) to a batcher node, usually the sequencer.
2. Once the sequencer receives a certain number of transactions, it will post them into an L1 smart contract as a batch.
3. A validator node will read these transactions from the L1 smart contract and execute them on their local copy of the L2 state. As for the AI inference execution, the validator needs to download the model from model DA and conduct the AI inference within the opML engine.
4. Once processed, a new L2 state is generated locally and the validator will post this new state root into an L1 smart contract. (Note that this validator can also be the sequencer.)
5. Then, all other validators will process the same transactions on their local copies of the L2 state.
6. They will compare their resultant L2 state root with the original one posted to the L1 smart contract.
7. If one of the validators gets a different state root than the one posted to L1, they can begin a challenge on L1.
8. The challenge will require the challenger and the validator who posted the original state root to take turns proving what the correct state root should be. This challenge process is also known as fraud proof. The fraud proof of Axonum includes the fraud proof of L2 state transition and the fraud proof of opML.
9. Whichever user loses the challenge, gets their initial deposit (stake) slashed. If the original L2 state root posted was invalid, it will be destroyed by future validators and will not be included in the L2 chain.

## Fraud Proof Design

The core design principle of the fraud proof system of Axonum is that we separate the fraud proof process of Geth (the Golang implementation of the Ethereum client on layer 2) and the opML. This design ensures a robust and efficient fraud proof mechanism. Here's a breakdown of the fraud proof system and our separation design:

1. Fraud Proof System Overview:
2. The fraud proof system is a critical component that guarantees the security and integrity of transactions on the Axonum optimistic rollup Layer 2.
3. It involves the verification of transactions and computations to ensure that any malicious behavior or inaccuracies are detected and addressed.
4. The fraud proof system is a critical component that guarantees the security and integrity of transactions on the Axonum optimistic rollup Layer 2.
5. It involves the verification of transactions and computations to ensure that any malicious behavior or inaccuracies are detected and addressed.
6. Separation of Fraud Proof Processes:
7. Geth Fraud Proof Process:
8. Geth, responsible for the Ethereum client on layer 2, handles the initial stages of fraud proof related to transaction validation and basic protocol adherence.
9. It verifies the correctness of transactions and ensures that they comply with the rules and protocol of the layer 2 system.
10. Geth, responsible for the Ethereum client on layer 2, handles the initial stages of fraud proof related to transaction validation and basic protocol adherence.
11. It verifies the correctness of transactions and ensures that they comply with the rules and protocol of the layer 2

system.

12. opML Fraud Proof Process:

13. opML, the Optimistic Machine Learning system integrated with Axonum, takes charge of the more intricate aspects of fraud proof related to machine learning model execution.

14. It verifies the correctness of machine learning computations and ensures the integrity of AI-related processes within the layer 2 framework.

15. opML, the Optimistic Machine Learning system integrated with Axonum, takes charge of the more intricate aspects of fraud proof related to machine learning model execution.

16. It verifies the correctness of machine learning computations and ensures the integrity of AI-related processes within the layer 2 framework.

17. Geth Fraud Proof Process:

18. Geth, responsible for the Ethereum client on layer 2, handles the initial stages of fraud proof related to transaction validation and basic protocol adherence.

19. It verifies the correctness of transactions and ensures that they comply with the rules and protocol of the layer 2 system.

20. Geth, responsible for the Ethereum client on layer 2, handles the initial stages of fraud proof related to transaction validation and basic protocol adherence.

21. It verifies the correctness of transactions and ensures that they comply with the rules and protocol of the layer 2 system.

22. opML Fraud Proof Process:

23. opML, the Optimistic Machine Learning system integrated with Axonum, takes charge of the more intricate aspects of fraud proof related to machine learning model execution.

24. It verifies the correctness of machine learning computations and ensures the integrity of AI-related processes within the layer 2 framework.

25. opML, the Optimistic Machine Learning system integrated with Axonum, takes charge of the more intricate aspects of fraud proof related to machine learning model execution.

26. It verifies the correctness of machine learning computations and ensures the integrity of AI-related processes within the layer 2 framework.

27. Benefits of Separation Design:

28. Enhanced Efficiency:

29. By distributing the fraud proof responsibilities, we optimize the efficiency of the overall system. Geth focuses on transactional aspects, while opML handles ML-specific fraud proofs.

30. By distributing the fraud proof responsibilities, we optimize the efficiency of the overall system. Geth focuses on transactional aspects, while opML handles ML-specific fraud proofs.

31. Scalability:

32. The separation design allows for scalability, enabling each component to independently scale based on its specific processing requirements.

33. The separation design allows for scalability, enabling each component to independently scale based on its specific processing requirements.

34. Flexibility:

35. This separation provides flexibility for upgrades and improvements in either the Geth or opML components without compromising the entire fraud proof system.

36. This separation provides flexibility for upgrades and improvements in either the Geth or opML components without compromising the entire fraud proof system.

37. Enhanced Efficiency:

38. By distributing the fraud proof responsibilities, we optimize the efficiency of the overall system. Geth focuses on

transactional aspects, while opML handles ML-specific fraud proofs.

39. By distributing the fraud proof responsibilities, we optimize the efficiency of the overall system. Geth focuses on transactional aspects, while opML handles ML-specific fraud proofs.
40. Scalability:
  41. The separation design allows for scalability, enabling each component to independently scale based on its specific processing requirements.
  42. The separation design allows for scalability, enabling each component to independently scale based on its specific processing requirements.
43. Flexibility:
  44. This separation provides flexibility for upgrades and improvements in either the Geth or opML components without compromising the entire fraud proof system.
  45. This separation provides flexibility for upgrades and improvements in either the Geth or opML components without compromising the entire fraud proof system.

## Axonum: The Brain of Ethereum

Axonum is the first AI optimistic rollup that enables AI on Ethereum natively, trustlessly, and verifiably.

Axonum leverages optimistic ML and optimistic rollup and introduces innovations of AI EVM to add intelligence to Ethereum as a Layer 2.

We enshrine AI into blockchain to build a decentralized supercomputer powered by global collective intelligence.

Learn more about Axonum: <https://docs.axonum.io>

Bridge to Axonum Sepolia: <https://app.axonum.io/bridge/deposit>

Add Axonum Sepolia to MetaMask: [Add Axonum](#)