

Dear Ethereum Community,

I am excited to share with you an ongoing EIP proposal that aims to enable statelessness at Ethereum account level (yet) allowing for a stateless account abstraction that scales Ethereum. Our proposal is still under R&D in stealth, but we are fortunate to have some top-tier individuals collaborating on it. If you are a top-tier developer, or cryptographer and interested in collaborating and co-authoring this EIP, please feel free to comment and join us in this effort.

Account Abstraction (AA) has been a frequently discussed concept in the blockchain ecosystem, and it is widely believed to be the mainstream solution for mass adoption. Account abstraction allows for the programmability of blockchain accounts and the ability to set the validity conditions of a transaction programmatically. The idea is to decouple the account and the signer, enabling each account to be a smart contract wallet that can initiate transactions and pay for fees. This approach can open up new opportunities for stateless transactions without changing the underlying protocol.

However, existing proposals, such as ERC-4337, which introduce certain features of Account Abstraction, are not efficient enough. These proposals fall short in terms of efficiency around failed transactions, security, easier rollup transferability, state and execution bloat, which are some of the longest and still unresolved challenges in the Ethereum protocol. These challenges pose a threat to managing the problem of the growing state, basic operation costs, and Ethereum scalability. Stateful ERC.4337 results in extra gas overhead, making it less efficient than desired.

We present Xstreamly, the first account abstraction layer for Ethereum with stateless transaction validation. With Xstreamly's AA model, miners and validating nodes in Ethereum process transactions and blocks simply by accessing a short commitment of the current state found in the most recent block. Therefore, there is no need to store a large validation state off-chain and on-disk, which can save up to gigabytes of storage. Our instantiation of Xstreamly uses a novel distributed vector commitment, a type of vector commitment that has state-independent updates. This means it can be synchronized by accessing only updated data, such as sending 5 ETH from Alice to Bob. To achieve this, we have built a new succinct distributed vector commitment based on multiplexer polynomials and zk-SNARKs, which can scale up to one billion accounts. Through an experimental evaluation, we show that Bloom Filters and our new distributed vector commitment offer excellent tradeoffs in this application domain compared to other recently proposed approaches for stateless transaction validation.

Our Stateless account abstraction model, leverages Bloom Filters to enable fast and accurate account existence checks, client-side validation to reduce the amount of redundant computation performed by the network, and client-side caching to further improve transaction processing times. Additionally, the proposed model incorporates a mechanism for distributing account state updates across the network, which can help minimize the risk of state bloat and improve the overall scalability of the network.

Once the validity of the transaction is proven, we designed a cryptographic model to execute the stateless transaction in which, contract storage is only updated when necessary. This model can be implemented on top of the Ethereum Virtual Machine (EVM) without any modifications to the Ethereum protocol. This stateless account abstraction model has the potential to significantly enhance the performance and efficiency of the Ethereum network without requiring changes to the underlying consensus mechanism. It also enhances Ethereum's functionality, significantly scales Ethereum, and reduces state and execute bloating, data storage cost, and execution costs. This model allows for more sophisticated smart contracts, enabling users to define complex conditions for executing transactions.

We look forward to receiving your thoughts, comments, and suggestions on this topic. Together, we can work towards a more scalable and efficient Ethereum network.