

Block Builder Architecture

Overview

?

This system's architecture is comprised of two main components: mempool and builder. They are decoupled and can operate as independent services, allowing the mempool to serve multiple builders, and likewise, builders to connect to multiple mempools.

These components all run within the protected environment of SGX and utilize remote attestation technology to verify if they are operating within an enclave, thereby ensuring the security of communication. Upon initial communication, they will exchange internally generated random keys, and all subsequent communication data will be transmitted through symmetric encryption.

Within this architectural framework, the mempool is responsible for receiving users' private transactions or bundles, and pushing them to the builder for processing. The builder, on the other hand, collects transactions or bundles from various channels such as mempool and geth public mempool, achieving centralized processing and collaborative work of transactions.

Automata Mempool

The automata mempool is a decentralized private mempool that directly receives user transactions and stores them inside an enclave. If a builder wants to access a private transaction, they must first go through attestation. The attestation checks whether the builder is also within the enclave and whether it is the expected code.

Automata Block Builder

The automata block builder operates within an enclave. Due to limited space, we designed the [stateless executor](#) within the builder to be stateless, fetching the corresponding state on demand. Meanwhile, we decoupled the mempool to become an independent service. The builder can simultaneously fetch data from multiple mempools. Before initiating encrypted communication between the builder and the mempool, they undergo an attestation process.

[Previous TEE Builder](#) [Next Stateless Executor](#) Last updated 1 month ago On this page Was this helpful?