Glossary

Smart Contracts

Programs that run on the Aztec network are called smart contracts, similar to programs that run on Ethereum.

However, these will be written in the Noir programming language, and may optionally include private state and private functions .

Barretenberg

Aztec's cryptography back-end. Refer to the graphic at the top of this page to see how it fits in the Aztec architecture.

Barretenberg's source code can be found here .

Sequencer

Aztec will be launched with a fully permissionless sequencer network that anyone can participate in.

How this works is being discussed actively in the Discourse forum . Once this discussion process is completed, we will update the glossary and documentation with specifications and instructions for how to run.

Sequencers are generally responsible for:

- Selecting pending transactions from the mempool
- Ordering transactions into a block
- Verifying all private transaction proofs and execute all public transactions to check their validity
- Computing the ROLLUP_BLOCK_REQUEST_DATA
- Computing state updates for messages between L2 & L1
- Broadcasting the ROLLUP_BLOCK_REQUEST_DATA to the prover network via the proof pool for parallelizable computation.
- Building a rollup proof from completed proofs in the proof pool
- Tagging the pending block with an upgrade signal to facilitate forks
- Publishing completed block with proofs to Ethereum as an ETH transaction

Previously in Aztec Connect there was a single sequencer, and you can find the Typescript reference implementation called Falafel here .

Provers

Aztec will be launched with a fully permissionless proving network that anyone can participate in.

How this works will be discussed via a future RFP process on Discourse, similarly to the Sequencer RFP Edit this page

Previous Migration notes Next Roadmap