Overview

As described in the RFC post, Aztec's sequencer selection design leaves block building entirely up to the sequencing nodes. This allows for the outsourcing of block production, potentially creating a market structure similar to the status quo between MEV-Boost and Ethereum validators.

We propose a method of integrating Astria's decentralized shared sequencer into Aztec's sequencer nodes to provide this off-chain block building infrastructure.

Leveraging Astria's decentralized MEV marketplace and much quicker finality, Aztec sequencer nodes can provide MEV searching infrastructure and fast preconfirmations for bundle inclusion without sacrificing security or introducing centralized points of failure.

Background: Astria

[

image

620×920 25.8 KB

](https://europe1.discourse-cdn.com/business20/uploads/aztec/original/2X/f/f0c40c587b7eee166b5c9bb71cc20bd11de411d7.png)

Write Path: Transaction Submission

1. Users sign a transaction via a wallet, submitting it to the rollup node (e.g. Astria's modified Geth), which is then gossiped to the rollup's mempool.

2. The Composer

is a developer tool we have created to act as a gas station. It retrieves pending transactions from the rollup node's mempool, then wrapping them in an Astria transaction and submitting them to Astria's mempool.

1. Astria's shared sequencer constructs a cross-rollup "meta-block", finalizing an ordering over the data posted to multiple rollups' namespaces at once.

2. The finalized shared sequencer blocks are posted to Celestia, allowing rollups to leverage strong guarantees over data availability for cheaper.

Read Path: Block Retrieval and Execution

1. Each rollup runs its own instance of the Conductor

. The Conductor

reads and validates rollup data from finalized Astria blocks.

1. Filtered blocks are passed to the rollup node (e.g. Geth) using Astria's Execution API, and are then executed by it, resulting in an updated rollup state and state root. This is how a rollup derives its state from the sequenced data posted to and made available by the data availability layer.

Conductor

[

image

620×840 19.6 KB

](https://europe1.discourse-cdn.com/business20/uploads/aztec/original/2X/6/61bec239756d0544a60fcca130a2abac8ee6af0b.png)

The [Astria Conductor

](https://hackmd.io/@astriaorg/HJ6cCpp9T#22-The-Astria-Conductor) is a sidecar process that a rollup node operator runs alongside the rollup node. It acts as a light client of the shared sequencer and data availability networks, pulling in Astria and Celestia blocks, validating them and feeding them into the rollup node using the Execution API.

The Astria Execution API

Astria integrates block production infrastructure into rollup architectures by running a Conductor

instance as a sidecar to the rollup node. The sidecar reads and validates finalized Astria blocks for the rollup's data and communicates with the rollup-specific software using Astria's Execution API.

While an in-depth explanation of the API and the specific request semantics can be found in thespecification document and the protobuf definitions, the main request of interest in this situation is ExecuteBlock

.

Astria-Driven Aztec Sequencer Node

[

image

1118×1240 67.2 KB

](https://europe1.discourse-cdn.com/business20/uploads/aztec/original/2X/2/2190b132e13eb10a91192f19dad390c5ad2ff36d.png)

As described in the RFC, Aztec sequencer nodes are chosen apriori according to a ranking generated by RANDAO. This allows traders and arbitrageurs to know who the sequencer will be for a given slot ahead of time. If the sequencer for a given slot is known to support using Astria for orderflow, users can choose to route their transactions via Astria's MEV markets during that slot.

This proposal will have an Aztec sequencer node run both a p2p mempool and take in transactions from Astria's Conductor

via the Execution API. As described in the diagram above, the top of the proposal will be deterministically derived from the ExecuteBlockRequest

s.

Block Timing

[

image

1004×560 24.7 KB

](https://europe1.discourse-cdn.com/business20/uploads/aztec/original/2X/e/ec877ed8f39cf5bca4e1ebb84de8efb4aebd31d1.png)

As Aztec requires validity proofs to be generated for blocks to be finalized, it will operate on much longer block times than Astria's lazy sequencing model. This allows the Aztec sequencer to use Astria for fast preconfirmations on high-value transactions, providing an interim solution that balances the need for security with the demand for speed.

Fast preconfirmations are especially valuable in a long block time environment. Slow confirmation time typically causes financial applications to suffer, as DEX prices become stale and slow oracle updates lead to larger liquidation cycles. This results in large amounts of value leakage from users available for extraction in each block, and provides the Aztec sequencer a monopoly over extraction.

Providing fast preconfirmations using a decentralized shared sequencer network allows the Aztec sequencer nodes to reduce uncertainty for traders and arbitrageurs by reducing the delay for confirmation they receive on ordering. Less uncertainty directly translates to reduced operational risk and thus lowering costs inherent to activities such as inventory management.

At the end of the slot, the Aztec sequencer node will construct a proposal from both avenues of orderflow: the Astria shared sequencer and the Aztec p2p mempool. However, transactions sequenced via Astria will have their preconfirmations as soon as an Astria block is constructed, allowing for financial applications to operate more smoothly.

[

image

900×528 25.8 KB

](https://europe1.discourse-cdn.com/business20/uploads/aztec/original/2X/b/b014b2a89875c617ce0f3f9ee807a06af75abe43.png)

Private and Public Transaction Flow

[

image

1240×620 85.2 KB

](https://europe1.discourse-cdn.com/business20/uploads/aztec/original/2X/1/1711b582c2c1dfdc51944ede4b575f8dd0c954cc.png)

The above diagram is based on Aztec's private transation lifecycle, and visualizes the data flows for both public and private transactions.

With regards to preconfirmations/MEV infrastructure and private versus public transactions, there a few "user experiences" that are enabled by the construction described in this document:

1. Private transaction that is not impacted by ordering

2. A user that wishes to make a private transfer, for example, where ordering has no impact on its execution, is able to send the transaction directly to the Aztec node via the p2p network. This is no different than the default behavior. Specifically, this proposal is purely additive, allowing users who don't wish to pay for ordering and preconfirmations to simply not do so.

3. A user that wishes to make a private transfer, for example, where ordering has no impact on its execution, is able to send the transaction directly to the Aztec node via the p2p network. This is no different than the default behavior. Specifically, this proposal is purely additive, allowing users who don't wish to pay for ordering and preconfirmations to simply not do so.

4. Private transaction that is impacted by ordering

5. As Astria operates a lazy sequencer model, it simply sequences sets of bytes. This allows traders and arbitrageurs who wish to conceal their trade (e.g. in order to avoid bundle stealing attacks) to still leverage Astria's transaction ordering market. Private transactions can follow the same flow as public transactions, submitting the private data to Astria as bytes to be sequenced and receive a preconfirmation. They will then arive at the Aztec node just the same via Astria's Execution API.

6. As Astria operates a lazy sequencer model, it simply sequences sets of bytes. This allows traders and arbitrageurs who wish to conceal their trade (e.g. in order to avoid bundle stealing attacks) to still leverage Astria's transaction ordering market. Private transactions can follow the same flow as public transactions, submitting the private data to Astria as bytes to be sequenced and receive a preconfirmation. They will then arive at the Aztec node just the same via Astria's Execution API.

7. Public transaction that is not impacted by ordering

8. Similarly to private transactions, public transactions can be submitted through Aztec's p2p network directly to the sequencer node. These will only depend on Aztec's block finality, determined by validity proof production, and will not have any preconfirmation guarantees.

9. Similarly to private transactions, public transactions can be submitted through Aztec's p2p network directly to the sequencer node. These will only depend on Aztec's block finality, determined by validity proof production, and will not have any preconfirmation guarantees.

10. Public transaction that is impacted by ordering

11. Similarly to private transactions, public transactions are also able to leverage Astria's transaction ordering markets and the resulting preconfirmations. While this enables traders and arbitrageurs to act against faster confirmations, it also allows regular users to subidize their inclusion and ordering fees by allowing bundles to be constructed around their transactions.

For example, if a retail user wants a fast preconfirmation on their swap, they can submit it to a searcher source of orderflow. The searcher can then bundle their transaction with their own, for example with a backrun, extracting some value in exchange for the retail user's transaction being confirmed much faster.

1. Similarly to private transactions, public transactions are also able to leverage Astria's transaction ordering markets and the resulting preconfirmations. While this enables traders and arbitrageurs to act against faster confirmations, it also allows regular users to subidize their inclusion and ordering fees by allowing bundles to be constructed around their transactions.

For example, if a retail user wants a fast preconfirmation on their swap, they can submit it to a searcher source of orderflow. The searcher can then bundle their transaction with their own, for example with a backrun, extracting some value in exchange for the retail user's transaction being confirmed much faster.

Long-Term Considerations

Decentralizing Trust in the Aztec Sequencer Node

This proposal assumes an implicit agreement between users and certain Aztec sequencers that are known to use Astria for orderflow. While this is the status quo in Ethereum L1 and MEV-Boost, in the longer term it would be preferable to provide a trustless mechanism for Aztec sequencer nodes to commit to using an external order flow source like Astria.

One possible solution is to create a system of Aztec smart contracts that enable trustless participation by validating that an Aztec sequencer who committed to using Astria as an order flow has indeed used it to construct their proposal.

Specifically, the smart contracts would need to provide a mechanism for proving that an Aztec proposer has equivocated on preconfirmations provided by Astria to users. Aztec sequencer node operators could then bond some collateral that would be slashed when provided with an equivocation proof.

Such a system would provide decentralized mechanism for an Aztec node to commit to sequencing via Astria and not require users to trust specific Aztec node operators, up to the security provided by the slashing condition.

This would likely add additional steps for registration and exiting as an Aztec sequencer, but could be done in a smart contract, likely not requiring protocol-level changes.

It should be noted that designing such a system is out of scope for this document, and the above design should be thought of simply as a rough sketch.

Relevant references

Astria

- Data flow and verification spec
- Sequencer inclusion proofs spec
- SUAVE-driven top-of-block for Astria-based rollups

Aztec

- Public-private execution
- L1-L2 architecture
- Aztec transaction internals
- Private transation lifecycle
- Proposal confirmation rules
- Sequencer registration
- Exiting as a sequencer
- Private Batch Swaps using Aztec and SUAVE