Abstract:

Centralized Sequencers in Layer 2 (L2) networks present a potential single-point-of-failure. Metis, operating as an optimistic rollup L2 network, is proposing a transition to a decentralized sequencer pool. This discussion delves into the technical nuances of this proposition, its implications for L2 networks, and the broader shift towards community governance.

Motivation:

The move to a decentralized sequencer pool is another step towards complete decentralization of the network.

This approach ensures:

- Expulsion of malfunctioning or malicious entities.
- Seamless and safe sequencer rotation.
- Enhanced network stability.

Moreover, the overarching goal is to transition the ownership of the network to the public. This ensures that even if the original developers or operators cease operations, the Metis network remains operational, preserving the integrity and continuity of services for its users.

Abstracted structure:

[

](https://app.diagrams.net/?page-id=RdPEqVS_jT8QG1Zhghx8&scale=auto#G1PvJfDEM7JXE2hmbH-tRXvKY1aZI9lNwk)

Abstracted entities:

1. User: sends transactions;
2. Admin: manages the locking nodes;
3. Whitelists potential sequencers.
4. Sets an upper bound on the stake that a single node can hold.
5. Controls reward emissions based on block production.
6. Metis Node (Sequencer) consists of:

- L2 Geth (including the OP-Node) - Responsible for transaction sequencing and the assembly of the blocks on Metis layer;
- Adapter module - Responsible for interacting with the other external modules on the consensus layer (POS Node);
- Batch submitter (Proposer) - Responsible for building the batches and submitting them to L1 after it gets signed by multiple sequencers;
- POS Node — works on 3 layers:
- Ethereum layer:
- A set of smart contracts on the Ethereum network responsible for locking and rewards for validators;
- A set of smart contracts on the Ethereum network responsible for locking and rewards for validators;
- Consensus (PoS) layer:
- A set of PoS Nodes based on Tendermint, these nodes run in parallel on the Ethereum mainnet
- When started, it detects the MPC addresses and calls the MPC module (see below) to trigger the keys generation if they do not exist;
- When the sequencer submits L2BatchTxs to L1, the signature needs to be generated by multiple existing sequencers (more than 2/3 of the Sequencer nodes participate in MPC signing);
- When the new sequencer node joins or exits, it performs the MPC resharing of private key shards without updating the

verification address in the locking contract (the verification address can also be generated if needed);

- Provides a variety of data query interfaces for Metis layer;
- A set of PoS Nodes based on Tendermint, these nodes run in parallel on the Ethereum mainnet
- When started, it detects the MPC addresses and calls the MPC module (see below) to trigger the keys generation if they do not exist;
- When the sequencer submits L2BatchTxs to L1, the signature needs to be generated by multiple existing sequencers (more than 2/3 of the Sequencer nodes participate in MPC signing);
- When the new sequencer node joins or exits, it performs the MPC resharing of private key shards without updating the verification address in the locking contract (the verification address can also be generated if needed);

- Provides a variety of data query interfaces for Metis layer;

- Metis layer:
- On this layer, for every new epoch another entity called Block Producer is getting selected and/or rotated according to the information generated by the consensus layer;

- On this layer, for every new epoch another entity called Block Producer is getting selected and/or rotated according to the information generated by the consensus layer;

4.1 MPC module:

[

](https://app.diagrams.net/?page-id=IMer1-ZH6BBqVrTDvv15&scale=auto#G1ZMZIi4oP4UWWh3RUZMPO9drWjFCNW1u3)

- It is responsible for the management of the entire life-cycle of the multisignature keys.
- Conducts external operations such as
- Multisig generation;
- Key resharing;
- Applying the signature;
- Deletion of signature.

- Multisig generation;
- Key resharing;
- Applying the signature;
- Deletion of signature.

- Provides support for the asynchronous usage of many multisignatures;

4.2 TSS Library - Threshold Signature Scheme Library - open-source multisig tool library and the main source of MPC logic:

- Responsible for the multisig key algorithm layer;

4.3 Key Local Storage:

- Conducts the saving and encrypting the key’s info in the local kv storage (levelDB) provided by the corresponding node;

4.4 Tendermint channel - open source p2p communication and consensus library provided by cosmos-sdk:

- PoS Node creates a separate Tendermint channel for communication messages between multiple p2p nodes during MPC operations;

4.5 libp2p

- libp2p Is an open-source p2p network communication library
- MPC Will use libp2p inCommunication messages between multiple different p2p nodes, supporting information

transmission during MPC operation

- Bridge module: The connection between Consensus layer (POS) and Sequencer (Metis node). It has 2 functions:
- Listener: It monitors the L1 Staking contract events and obtains the info about Sequencer node list (join/exit/update), also it listens to Metis block's events to determine whether to send tasks;
- Processor: It sends the transactions to TSS/Themis for consensus. Scans epoch and Metis block information. Asks Themis to resend the proposal for the epoch if the block is wrong. Scans the MPC service interface and sends the transaction batches to the consensus layer for MPC signing.

Notes:

- Staking Mechanism: Nodes will be allowed to stake an amount of Metis. The upper bound of this stake will be initially controlled by the Admin, but will be subject to change later based on governance decisions.
- Reward Emissions: While initially will be controlled by the Admin, the plan is to transition this to a more decentralized mechanism, potentially influenced by community governance.
- Transition to Community Governance: As the network matures and more sequencers join, the aim is to transfer the Admin's responsibilities to a community governance model. This would ensure that key decisions, like whitelisting sequencers or adjusting staking bounds, are made collectively by the network's participants.

Discussion:

Discussion Points:

- Impact on the performance and security metrics of traditional centralized sequencers
- potential challenges or flaws introduced by transitioning to a decentralized sequencer pool
- The need for decentralized sequencers, and impact on the broader landscape of L2 Rollups