Abstract

We introduce a data availability layer design based on Ethereum Layer2, offering a more cost-effective data layer for Ethereum-based Layer2 social, gaming, and other applications, while ensuring a close integration of data commitments with Ethereum. Users can pay fees on this Layer2 to send special types of transactions, which include data commitment for data D that needs to be published. The Topia Operator processes these transactions, calculates the validity proofs, and publishes them along with the transaction data to Ethereum. Using the Restaking technology, a subset of Ethereum validators operates the Topia Validator client to form a p2p data network, verifying blocks using techniques of randomly sampled committees and data availability sampling. This project is still in its early stages, and this article aims to provide a comprehensive analysis of its architecture, motivations, and design, welcoming feedback from the community.

Motivation

The current landscape of Ethereum or other blockchain systems presents several challenges:

1. Trust through Economic Security:
2. Validators must inject funds to safeguard the network, resulting in marginal costs.
3. The cryptoeconomic security of the set of validators depends on the total value of staked tokens. If these tokens are highly volatile, a significant drop can reduce the cost of attacks and potentially trigger security incidents.
4. Validators must inject funds to safeguard the network, resulting in marginal costs.
5. The cryptoeconomic security of the set of validators depends on the total value of staked tokens. If these tokens are highly volatile, a significant drop can reduce the cost of attacks and potentially trigger security incidents.
6. Diverse Security Needs:
7. As pointed out by Polynya, not all applications, users, or use cases need billions in cryptoeconomic security. WhileEIP-4844 and the anticipated Danksharding are expected to significantly reduce per-byte costs for data publication, the expenses might still be too high for data-intensive applications that require “millions of TPS” but don’t need high security.
8. As pointed out by Polynya, not all applications, users, or use cases need billions in cryptoeconomic security. WhileEIP-4844 and the anticipated Danksharding are expected to significantly reduce per-byte costs for data publication, the expenses might still be too high for data-intensive applications that require “millions of TPS” but don’t need high security.

Solution

We’ve designed an architecture for the data availability layer based on Ethereum Layer2. We introduce the Restaking technology, allowing Ethereum validators to voluntarily take on additional slashing risks to provide services, enhancing rewards for honest validators, and offering token security and validator quality at the Ethereum level. We also propose a dedicated fee model where per-byte gas correlates with the size of the Topia validator set, so different validator set sizes result in various costs and cryptoeconomic securities. Our design capitalizes on the latest achievements of Danksharding, optimizing node workload and maximizing security.

TopiaDA’s Architecture

The Topia system is primarily composed of three parts:

- Topia Operator
- DA network composed of Topia DA nodes
- Topia Smart Contract on Ethereum

DA Transaction Lifecycle:

1. DA Buyer submits a DA transaction carrying a blob to Topia Operator and pays the corresponding fees.
2. Topia Operator processes the transaction, verifies the blob’s KZG commitment, and submits it to the DA contract.
3. Topia Operator combines multiple blobs to form a DA matrix B, then uses erasure coding to extend it to matrix E, and subsequently publishes the expanded matrix E, KZG commitments, and related proofs to the Topia DA network.
4. DA validators in the Topia DA network store shard blobs, check KZG commitments, perform data availability checks, and sign data availability attestations, then broadcast these attestations to the network.

5. Topia Operator collects availability attestations from the DA network, performs verification and aggregation, then submits the aggregated attestation to the DA contract.
6. Topia Operator sends a transaction receipt to the DA Buyer.

1. Topia Operator:

The Topia Operator maintains a state machine that uses Ethereum as its data layer and relies on Rollup technology to ensure the correctness of state transitions. It acts as both the builder of DA blocks and the central link between DA buyers, Ethereum, and the DA network. Its responsibilities include:

- Processing Topia transactions: Topia transactions are limited to several types, which include submitting DA transactions carrying blobs, deposit/withdrawal transactions between Topia and Ethereum, transfer transactions, cross-rollup information synchronization transactions, and so on. The Topia Operator sequences and executes these transactions. At the same time, the Topia Operator handles reward distributions, DA fee allocations, and validator slashing operations.
- Rollup: Publishes Topia blocks and commitments to Ethereum, submits validity proofs or accepts fraud challenges, and handles deposit/withdrawal transactions between Topia and Ethereum. It's worth noting that blobs in DA transactions are not published to Ethereum since they are already stored in the DA network.
- Handling DA data: Constructs the DA expansion matrix, KZG commitments, proofs, and broadcasts them to the DA network; receives availability attestations, aggregates them, and submits them to Ethereum.

2. DA Network:

The DA network is a p2p network comprised of many Topia DA nodes, mainly dealing with data synchronization and data availability sampling, similar to the current Danksharding specification.

Block builders combine m

fixed-length data-blobs into a block. Each data-blob is divided into m

chunks, forming an m \times m

matrix B

as the raw block data. The builder further extends this m \times m

matrix in both horizontal and vertical directions by RS coding, producing a 2m \times 2m

expanded matrix E

. \overline{C}=(C_0, ..., C_{m-1})

can be viewed as the commitment to the entire matrix E

, with the KZG commitment of the i^{th}

row being C_i

. After this process, block builders publish the expanded matrix E

, commitment \overline{C}

, and proof \pi

to the network.

After a block builder publishes a 2m \times 2m

sized expanded block to the network, Topia validators don't download the entire expanded block. Instead, they download randomly assigned a

rows and a

columns of data. Validators are responsible for storing this data for up to 30 days and responding to chunk requests on these rows and columns. Additionally, validators will try to recover any missing chunks through reconstruction and broadcast them to the network.

Besides the above tasks, validators will randomly select k

chunks from the expanded matrix, chunks located in different positions outside the assigned a

rows and a

columns. Validators will then request these chunks and their corresponding proofs from the entire validator network. If all k

chunks, as well as the assigned a

rows and a

columns, are available and match the commitment \overline{C}

, the validator can consider the block to be available.

3. Smart Contract:

1. Restaking Contract:
2. Allows Ethereum validators to stake their ETH a second time. After restaking, validators can earn rewards on the Topia network. Misbehavior here could result in their original ETH stake being slashed.
3. Allows Ethereum validators to stake their ETH a second time. After restaking, validators can earn rewards on the Topia network. Misbehavior here could result in their original ETH stake being slashed.
4. Data Availability Contract:
5. The data availability contract receives data commitments submitted by Layer2 and verifies signatures from the Topia validator set. Other applications can directly read the latest data commitments from the contract.
6. The data availability contract receives data commitments submitted by Layer2 and verifies signatures from the Topia validator set. Other applications can directly read the latest data commitments from the contract.
7. Rollup Contract:
8. Handles interactions with the Topia Operator in terms of Rollup.
9. Handles interactions with the Topia Operator in terms of Rollup.

Cost Model: DA Fees

Data availability sampling techniques crucially rely on client-side randomness that differs from each client. Assuming each validator is unique, a larger validator set size results in more redundancy and greater decentralization, thereby enhancing data security. For DA users, it's reasonable to pay more per-byte gas for increased data security. Considering the summary of the reward formula by Vitalik in the discouragement attacks paper:

r = N^{-p}

where r

represents the validator's interest rate, and N

is the size of the validator set.

We take p=0

, implying that the validator's interest rate is a constant. This is based on the following three reasons:

1. Per-byte gas is proportional to the size of the validator set.
2. It favors scaling the network size, as the interest rate for Restaking remains the same regardless of the DA network's size.
3. It prevents discouragement attacks.

The detailed cost model is as follows:

1. Validator DA earnings = Total DA fees paid by all DA Buyers - Protocol cut (e.g., 10%)
2. When the block is at the target size:
3. Validator's DA interest rate (APY_{fee}

) is a constant:

APY _ {fee} = C

- Per-byte user cost is proportional to the total staked amount:

gas_per_byte = k _ 1 \times \sum active_balance

1. Validator's DA interest rate (APY_{fee}

) is a constant:

1. Per-byte user cost is proportional to the total staked amount:
2. For the stability of average node loads and to prevent DoS, a mechanism similar toEIP-1559 has been added to adjust fees when the block size deviates from the target.

Pros and Cons:

Advantages:

- Provides an economical and sufficiently secure solution for low-to-mid value use cases.
- Leverages the unique randomness of Ethereum validators due to Restaking technology.
- Tightly integrated with Ethereum compared to other base chains serving as data layers, making it developer-friendly for Ethereum-based Layer2 developers.
- Data availability sampling technology offers malicious-majority security features. Light clients can independently verify block availability without relying on an honest majority assumption.

Disadvantages:

- Data availability sampling technology is relatively new and has not been extensively tested in practice, and designing a supportive p2p layer for it is challenging.
- Restaking technology might introduce leveraged andcentralization risks.
- Data throughput is limited compared to L2 data layers without a p2p network.

Further Challenges:

1. Token Complexity:

How can we efficiently handle the complexity of supporting multiple settlement tokens?

1. Variable Security Levels:

Allowing users to more flexibly choose the size of the validator set. * For instance, if a user is only willing to pay 50% of the fee, could we potentially support this by randomly selecting 50% of the validator set?

1. For instance, if a user is only willing to pay 50% of the fee, could we potentially support this by randomly selecting 50% of the validator set?
2. Topia Operator Architecture:

The security and censorship resistance of the Topia Operator architecture require more rigorous analysis.