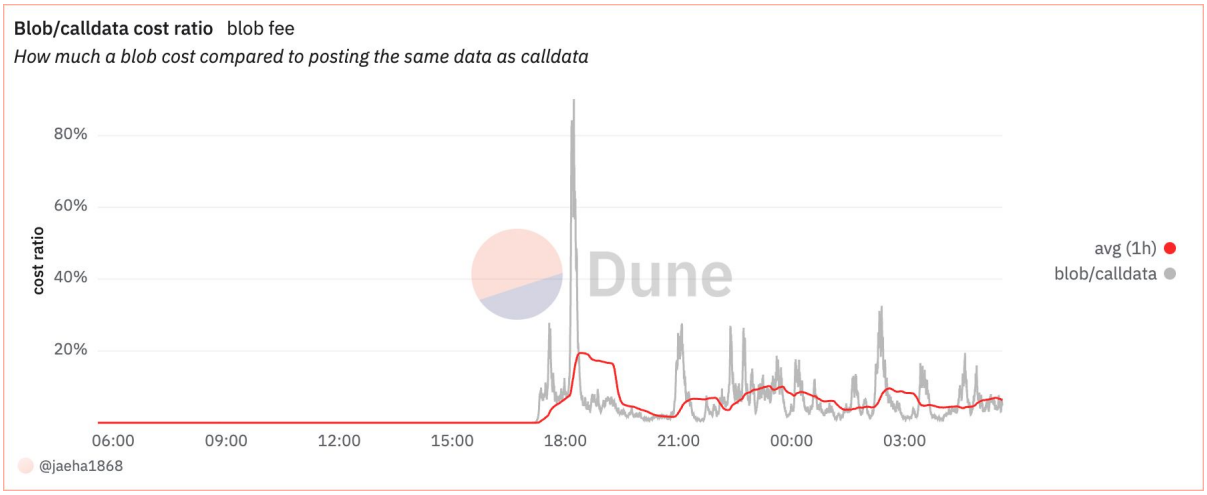
April 12th, 2024

Jaehyun Ha I Research Analyst
jaehyunha@prestolabs.io

Summary

- Through the Dencun mainnet upgrade, the Ethereum Foundation had recently implemented EIP-4844 (Proto-Danksharding) protocol update for enhancing its data availability (DA) space utilization. EIP-4844 had significantly reduced the L2 transaction fees and strains of L1 by introducing a new temporal storage format called “Blob”.
- Yet, EIP-4844 also presents potential limitations regarding fees and scalability. This arises from restrictions on the data amount loadable into blob and the exponential increase in blob fee prompted by the rising demand for roll-ups. The notable surge in blob fees due to Blobscription on March 27, 2024, exemplifies this.
- DA layer projects, such as Celestia, Avail and EigenDA can be an alternative solution for overcoming such constraints. These DA layers can scale without increasing hardware requirements or sacrificing security. However, DA layers relying on cross-chain bridges are potentially vulnerable to 51% attacks; thus the trade-off has to be considered by individuals.

Figure 1: Wreaking havoc on blob fees (March 27, 2024) Source: Optimism, Presto Research



What is DA, and what are the challenges it faced?

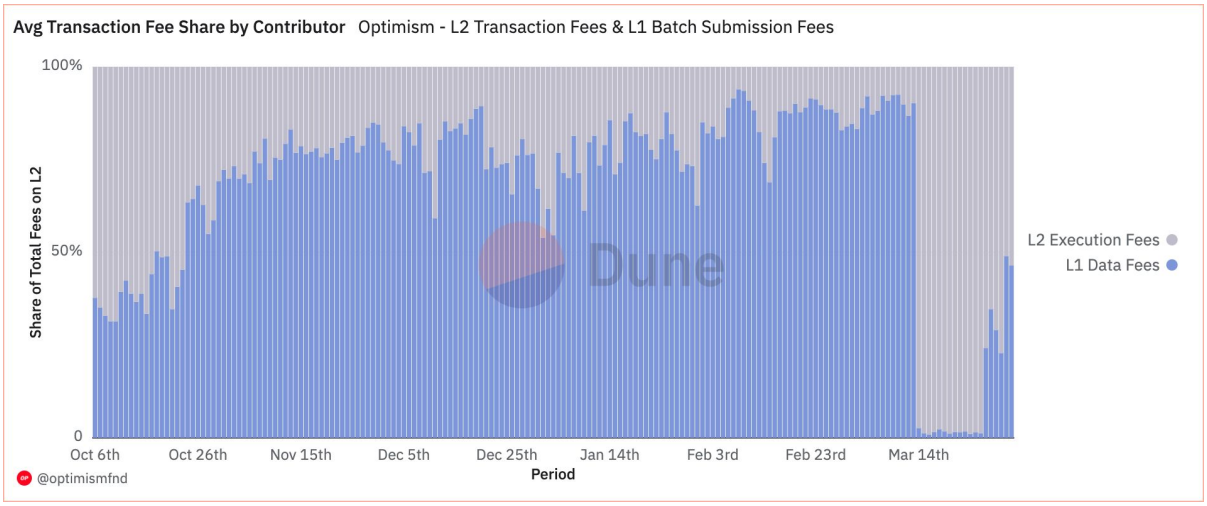
In the context of blockchain, data availability (DA) refers to the assurance that the data required for verifying a block (or new state) is accessible to all participants in the network. DA is typically not a significant issue in monolithic blockchains (e.g., Bitcoin - where full nodes are responsible for all execution, consensus, settlement and data availability), since every full node downloads a copy of the entire blockchain, and each of them will discard fragmented blocks that do not adhere the protocol rules. All data included in blocks should be available to the public, and each full node should be able to verify whether the proposed blocks are legitimate by independently executing the transactions.

However, ensuring DA becomes a bit more intricate when it comes to L2 roll-ups (e.g., Optimism, zkSync). L2 roll-ups scale up the transaction processing progress of L1 blockchains by executing bundles of transactions off-chain, and then submitting the aggregated result (i.e., the newest state) to their respective base layers. However, to trust the result, DA regarding the L2 transactions used to create that newest state must be ensured. In other words, L2 transactions should be stored in an accessible manner, enabling anyone to individually reconstruct the newest state of L2 in case L2 sequencer goes down or incorrect state is submitted.

To guarantee DA for clients, roll-ups prior to EIP-4844 (i.e, Proto-Danksharding) stored the L2 transaction data permanently in L1 calldata storage of roll-up sequencer inbox contract (this implementation may vary across different roll-up services). Although this is still cheaper than processing transactions directly on L1, the fact that the majority of roll-up transaction fees are used for L1 calldata storage fees (60~85% for Optimism), and permanent storage of L2 data which has already undergone sufficient validity checks on L1 poses a potential bottleneck. This highlights the need of considering more cost-effective data storage methods.

Figure 2: Dominance of L1 storage fees before EIP-4844

Source: Optimism



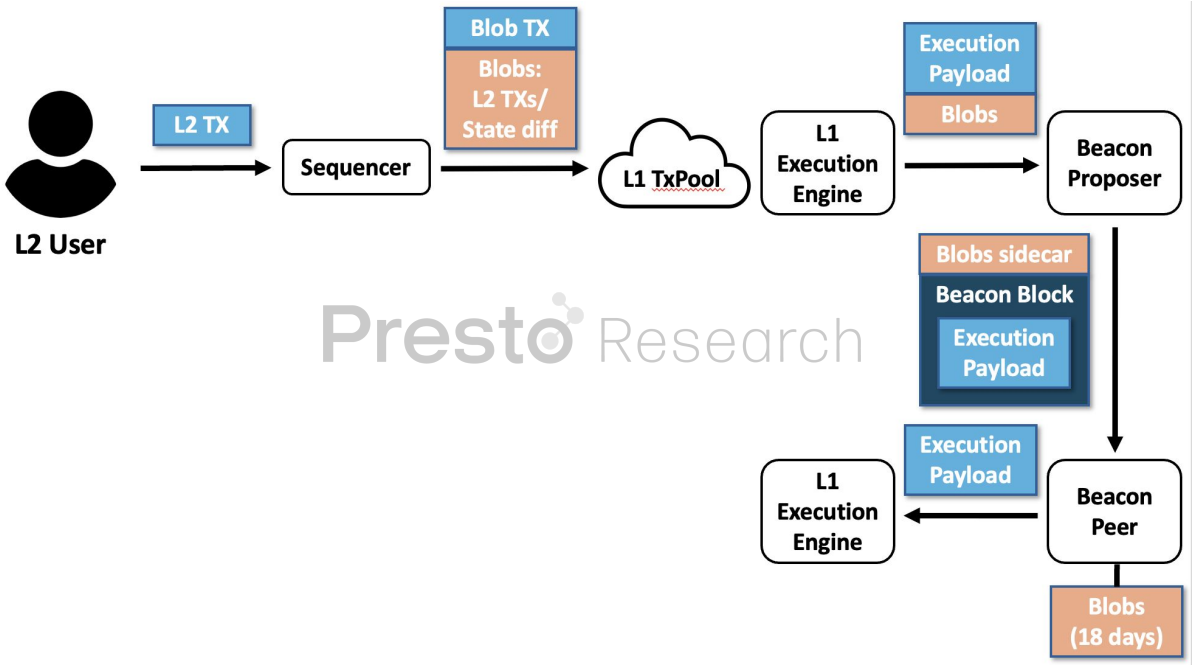
EIP-4844: Cornerstone of Roll-up Centric Roadmap

For better data space utilization for L2 transactions, the Ethereum foundation had recently (as of March 13, 2024) mounted EIP-4844 (Proto-Danksharding) protocol update during the Dencun mainnet upgrade, as the basic milestone for its roll-up centric roadmap. EIP-4844 introduces a new format of transaction called “blob-carrying transaction” with the short-term goal of reducing the storage fees of L2 transactions and the long-term goal of establishing the cornerstone for future integration with Danksharding.

The reason blob-carrying transactions could bring such advancement is due to its two key features; firstly, storing L2 transactions in a non-EVM accessible data availability space called “blob”, and secondly, temporarily storing that blob in the consensus layer to alleviate the load on each nodes. The diagram below depicts the transaction lifecycle in Ethereum network with EIP-4844 implemented. At first an L2 user initiates a transaction and submits it to the sequencer; the sequencer then packages the submitted L2 transactions into “blobs”, and sends it to L1 transaction pool along with “blob transaction” which contains the commitments of the data.

Figure 3: Lifecycle of L2 transaction in EIP-4844

Source: Presto Research



An important point to note is that the blob transaction (i.e., the commitment) is executed by EVM and delivered to beacon nodes in a form included in the execution payload, while the blob itself remains unexecuted by the EVM and is simply forwarded as it is. The execution payload containing the blob transaction will get included in the beacon block as usual and permanently stored on the blockchain, but in contrast the blob is stored as a sidecar attached to the block for about 18 days by beacon peers and is then automatically deleted. This reflects the direction of Ethereum's roadmap to utilize Ethereum not as full data storage, but rather as a real-time bulletin board where leaving room for other protocols to do long-term storage instead.

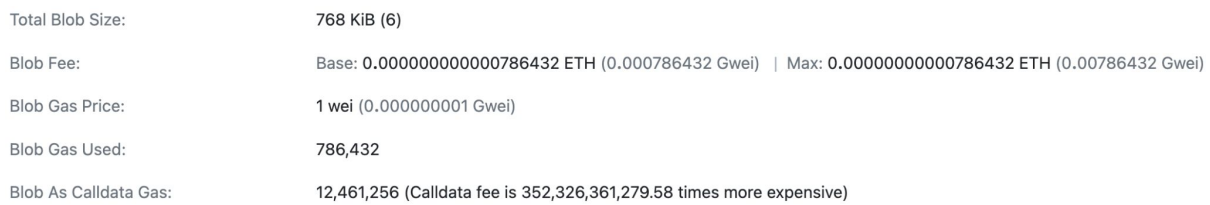
L2 transaction fee advantages from EIP-4844

Since blob data is temporarily stored by the beacon nodes in a form inaccessible from the EVM, the amount of gas required per byte (storage cost) is much cheaper than storing L2 transactions in L1 calldata. According to the current EIP-4844 specifications, each blob requires 131,072 gas. With each blob's size set at 128KiB, one can infer that blob data consumes 1 gas per byte. This is approximately 16 times cheaper compared to the method of storing L2 transactions in calldata, which consumes 16 gas per byte.

That's not the end of it; another noteworthy aspect of EIP-4844 is that blobs follow an independent gas fee accounting rule (i.e., *"blob gas - new type of gas"*). This is somewhat akin to EIP-1559; the blob gas price is determined proportionally based on the network's blob usage. The initial setting value of the blob gas fee was established at 1 wei (not 1 Gwei!), making the early roll-up costs nearly free. Presently, EIP-4844 is designed to take an average of three blobs per block, with a maximum of six. Even in a scenario where six blobs are attached to a single block, the storage cost for these six blobs would be 131,072 gas * 6 = 786,432 gas. As of March 26, 2024, the blob gas price was still set at 1 wei (= 10^-18 ETH); the blob fee per block in this case would be 786,432 gas * 1 wei = 0.000000000000786432 ETH.

Figure 4: Blob fees during happy days (March 26, 2024)

Source: Etherscan



Potential limitations of EIP-4844

EIP-4844 has effectively fulfilled its primary objective of reducing L2 transaction fees and easing the data availability space strain on L1. However, as implied by its name (*PROTO*-danksharding), this is not the final step towards enhancing Ethereum's scalability. Future updates such as PBS (Proposer-Builder Separation) and full Danksharding are poised to further increase TPS and reduce transactions fees. Nevertheless, the implementation of Danksharding on the mainnet is still expected to take considerable time; thus it is needed to analyze where EIP-4844 has room for improvement and consider ways to utilize it in a better way in the meantime.

Let's first delve into the discussion about gas prices. Some might say, "Wait, didn't this guy just mention that the blob gas price is overwhelmingly cheaper compared to the transaction gas price? What's the problem?" Well, it appears in that way at first glance—however, it's challenging to guarantee that blob gas prices will always remain cheaper than calldata fees in the future. Surprisingly, that concerning situation arose just two weeks after the Dencun update. The blob gas price, which had been stagnant at 1 wei, surged to a peak of 595.10 "Gwei (= 10^9 wei)" on March 27, 2024, and has since consistently maintained levels over 30 Gwei. How could the blob price surge by nearly 100 billion times in just one day? Let's analyze the reasons by examining at the blob gas price determination policy outlined in the EIP-4844 official documentation.

According to EIP-4844 specs, the blob gas price is determined solely by the exponential of "excess blob gas". Excess blob gas is defined as the accumulated excess blob gas from the block that first included blobs after the EIP-4844 update up to the latest block. Here, the term 'excess' refers to the difference between the total gas actually used and the total gas intended to be used. EIP-4844 has intended for a consumption of 393,216 blob gas per block, which is gas amount equivalent to 3 blobs.

In other words, if there were a total of n blocks on the chain that could accommodate blobs, and X blobs have been attached in total, excess blob gas would be equivalent to (X - 3n) blobs. If 6 blobs were consistently included in every block from the first block onwards, the blob gas price would increase by a factor of 1.125 per block. With an additional 200 blocks created in this manner, and considering Ethereum's block time of approximately 12 seconds, the blob gas price, initially 1 wei, could reach around 17 Gwei just in 40 minutes.

After the EIP-4844 update, initially, not all blocks did include blobs, resulting in the total gas consumed by blobs rarely exceeding the target value. However, since March 27th, 2024, the service “BlobScriptions”, which directly inscribing data onto blobs went viral. Consequently, blobs over target amount had consistently been submitted per block, leading to an exponential increase in blob gas price. As a result, even though it’s just for a short while, blob gas fees have nearly reached parity with calldata gas fees.

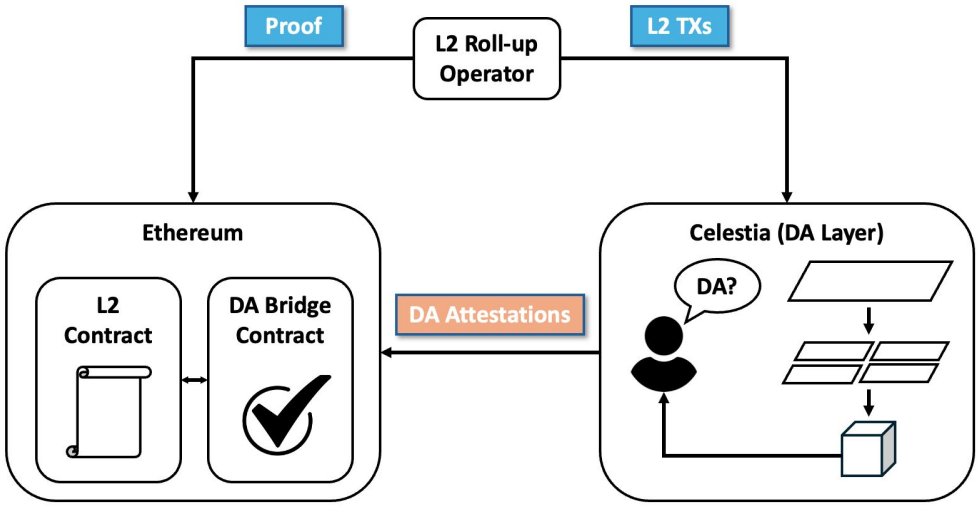
Another discussion pertains to TPS. As previously mentioned, EIP-4844 aims to incorporate an fixed size of three 128 KiB sized blobs per block. Given Ethereum's block generation rate of approximately one every 12 seconds on average, it can be inferred that storage space for around 3 * 128 KiB / 12 = 32 KiB of L2 transactions per second is made available. Prior to EIP-4844, statistics from Optimism indicate that each L2 transaction incurred roughly 3000 to 4000 gas of L1 calldata storage costs. Considering the gas fee for calldata at 16 gas per byte, it can be estimated that each L2 transaction occupied approximately 187 to 250 bytes in L1. By combining these two observations, an approximate TPS of 131 to 175 can be achieved. This figure exceeds mainstream L2 services by more than twice and surpasses Ethereum mainnet's TPS by more than tenfold, yet it still falls short compared to everyday payment methods such as Visa or Mastercard.

To sum up, due to its fixed blob space size and fee accounting policy, EIP-4844 faces the challenge of effectively regulating gas fees in response to the increasing demand of blob space from roll-up based projects.

DA Layers: Potential solutions for DA problem

Here, DA layers—separate blockchains or systems specialized for storing transaction data from other blockchains—offer a solution to the limitations posed by EIP-4844. Each in their own way, DA layers can adjust the size of storage and stabilize storage fees even as demand grows. In this context, we dive deeper into three prominent DA layer projects—Celestia, Avail, and EigenDA—and analyze how did they address the scalability problem for DAs, and what limitations do they face.

Figure 5: Overview of how Celestia works as a DA layer Source: Celestia, Presto Research



Presto Research

Celestia, Avail: Scaling through DAS

In the context of roll-up, where Celestia chain serves as the DA layer, L2 transactions are sent to Celestia while proofs are submitted to L2 contracts on Ethereum. Here, L2 transactions sent to Celestia undergo 2D Reed-Solomon encoding before being integrated into a block and disseminated to light nodes. Then, each light node can verify the DA of blocks by DAS (Data Availability Sampling) - which is DA verification process by conducting multiple rounds of downloading for only small portion of the block.

As a light node continues to sample block data across multiple rounds, its confidence for the DA becomes more promising. Once it achieves a predetermined confidence threshold (e.g. 99.9%), the light node deems the block data as available. This process is possible without downloading the entire data of the block. The scalability of Celestia comes from here; as more light nodes participate in the Celestia network, they can verify DA for larger datasets through DAS, enabling the expansion of block sizes in response to growing demands without increasing the hardware requirements or sacrificing security. This is how Celestia can stabilize fees.

Once the DA of transactions submitted to Celestia are ensured by DAS from light nodes, then Celestia validators submit the merkle root of signed available data to the Celestia's DA bridge contract on L1 in the form of "DA attestation". If a new state transition is submitted to the L2 contract, rather than relying on calldata or blobs, the DA verification takes the form of querying the DA bridge contract to check whether the corresponding data is available (i.e., the DA attestation is correctly made).

Avail shares a core philosophy with Celestia as a blockchain, but differs in its detailed implementation. Both use erasure coding for data recoverability, allow light nodes to verify block data's DA via DAS, and confirm DA through a DA bridge deployed on L1.

However, there is a key distinction in how they verify the correctness of the encoded block. Celestia employs a fraud-proof mechanism to detect incorrectly encoded blocks. The advantage of fraud-proof lies in its simplicity of implementation; Celestia's validators do not need to perform additional expensive work when producing blocks. However, there is a drawback of slight delay in confirming the finality of whether a block is correctly encoded, as Celestia's light nodes must wait until they receive fraud-proof information from full storage nodes.

In contrast, Avail utilizes a validity proof scheme to ensure block encoding correctness. Avail's validators provide KZG commitments alongside block proposal, enabling light nodes to immediately confirm block's correctness without relying on full nodes. This offers the advantage of reduced latency for light nodes but requires validators to have higher-performance hardware. Consequently, both DA layer services involve a trade-off between validator overhead and light node latency. Thus, developers should meticulously analyze the strengths and weaknesses of each DA layer to choose the most suitable option for their needs.

Figure 6: DA layer Comparison

Source: Presto Research

	EIP-4844	Celestia	Avail	EigenDA
Support DAS	No	Yes	Yes	No
Encoding proof scheme	Validity proofs	Fraud proofs	Validity proofs	Validity proofs
Block time	12 sec	15 sec	20 sec	N/A
Verification time	12-15 min	< 10 min	< 1 min	12-15 min
Ability to scale	No	Yes	Yes	Yes

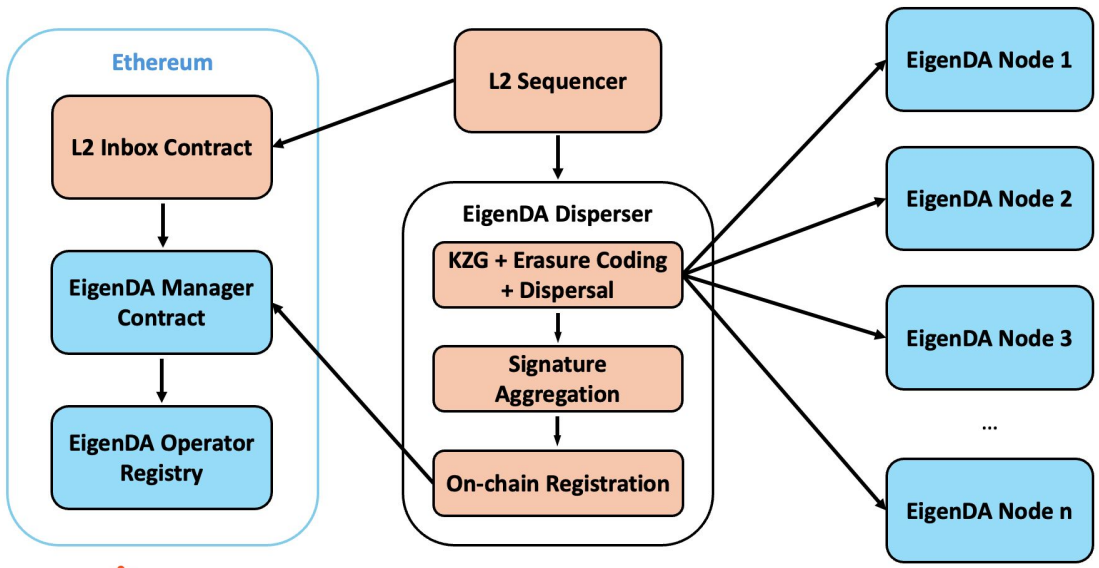
EigenDA: Scaling through DAC

EigenDA, the first Actively Validated Service (AVS) launched on EigenLayer, differs from Celestia and Avail that it does not take the form of a blockchain but instead resembles an efficient database. Unlike the two blockchains, which scaled by allowing each light node to verify the DA of blocks through DAS, EigenDA achieves horizontal scalability by forming a Data Availability Committee (DAC) with EigenDA nodes which store the fragments of data blobs for a predefined time period, and providing DA for each of them upon requests.

In EigenDA, restakers of EigenLayer can delegate their stakes to EigenDA node operators responsible for data validation. The roll-up sequencer forwards ordered datablobs from L2 to the disperser, which applies erasure coding and divides them into chunks. Each chunk, along with KZG commitments and proofs, is then sent to EigenDA nodes. These nodes store the data and are obligated to provide DA of each chunks. After confirming the consistency of the received data and proofs, EigenDA nodes submit their signatures back to the disperser. After the disperser aggregates signatures, it “registers” the blob onchain by sending a transaction to the EigenDA Manager contract with the aggregated signature and blob metadata.

Figure 7: EigenDA Architecture

Source: EigenDA, Presto Research



Potential risks of DA layers: Cross-chain bridges

Although DA layers have the advantage of enhancing the scalability of existing roll-up services, they ultimately face a vulnerability due to their dependence on chains or services outside of Ethereum. This issue has been also pointed out by Vitalik Buterin, that transferring data from one zone of sovereignty to another makes it difficult for the data to be protected by protocol rules. For instance, in the case where Celestia were under 51% attack, attackers wouldn't be able to steal the Celestia users' native assets This is because 51% attackers could create or revert the history of blockchain, but honest users would not follow the corrupted chain anyway since it violates the protocol rules.

However, it becomes a different story if the target of the attack is a cross-chain bridge. As aforementioned, Celestia itself may have resistance to a 51% attack. However, the Ethereum network does not directly follow DAS executed on the Celestia network; instead, it verifies DAs for L2 transactions through querying DA bridge contracts located within Ethereum. If a 51% attack were to occur on Celestia, the attacker could gain the authority to manipulate the DA attestation data stored in the bridge contract through the majority of malicious validators under control. In this case, if L2 contracts send queries to the bridge contract for DAs, corrupted responses could be returned, leading to a situation where DAs cannot be ensured in L2 roll-ups. This can also happen to Avail and EigenDA, as Avail also uses data attestation bridge, and EigenDA uses untrusted disperser to upload the aggregated signature to Ethereum.

While the risks mentioned are noteworthy, orchestrating a 51% attack on a single bridge demands considerable resources and effort. Not all cross-chain bridges are immediately vulnerable to such attacks. However, the probability of becoming a target escalates with the concentration of funds in specific bridges. We've witnessed several cross-chain bridge exploits leading to substantial financial losses in recent years; thus the risk shouldn’t be overlooked.

Conclusion

In this article, we've delved into the importance of Data Availability (DA) in roll-ups, explored the attempts and limitations of EIP-4844 in addressing the DA problem, and examined the scalability and its constraints of DA layers that could serve as alternatives to EIP-4844. Each DA layer offers its unique advantages and clear limitations. Roll-up developers aiming to minimize DA failure probability may find EIP-4844's blob usage beneficial. For validators seeking to participate in consensus without heavy hardware requirements, Celestia could be an optimal choice. Avail might suit light node operators aiming for fast finality without full node dependency. If prioritizing DA layer bootstrapping, EigenDA could present a viable solution. Choosing the most superior DA layer is a subjective process, and one must carefully weigh the trade-offs associated with his/her usages.

About Presto

Presto is a Singapore-based algorithmic trading and financial services firm founded in 2014. Presto focuses on delivering exceptional value for clients through rigorous research-driven approach to investment and trade execution. With more than a 100 million trade executions in a day, Presto is a leading financial services firm in both digital assets and traditional finance markets.

Find out more at https://www.prestolabs.io.
Follow Presto for more content: X, LinkedIn

Authors

Jaehyun Ha, Research Analyst : X, LinkedIn

Required Disclosures