EthDA Lightpaper

version 0.9.2

Oct 2023

Contents

Abstract

EthDA is an Ethereum layer 2 ZK network with a permissionless set of decentralized sequencers based on a soft proof-of-stake consensus mechanism. Its main goal is to scale Ethereum's storage capability by leveraging blob-carrying transactions, which are proposed by EIP-4844 and generalized for storing any type of data. Blobs are sharded and permanently stored by sequencers using a Data Availability Sampling mechanism that mirrors the one used by Ethereum layer 1, in accordance with Ethereum's Danksharding scaling roadmap. Moreover, EthDA provides a set of application level protocols that allow users to store and retrieve files of arbitrary sizes based on Blobs, making it not only a suitable Data Availability network for emerging Ethereum L2s, but also an innovative fully-on-chain decentralized storage solution for the Ethereum ecosystem.

Motivation

Alt-DA

There are various Ethereum scaling solutions like sidechains, rollups, plasma and validium, etc, with different characteristics and trade-offs between security, decentralization, and throughput, etc. One of the most essential differences is Data Availability [1]. Using Ethereum L1 as DA derives security from Ethereum and that's the reason why rollups are widely regarded as the only trustless scaling solution in the medium and long term [2]. But posting L2 transaction data to L1 requires high gas consumption and may cause congestion on L1. This situation becomes even worse as L2 ecosystem prospers.

The long term and canonical solution to this DA dilemma is Danksharding [3], and the medium term solution is EIP-4844 [4], or Proto-Danksharding, which the Ethereum community is actively working on as part of the upcoming Cancun-Deneb upgrade. EIP-4844 introduces a new type of Blob TX that rollups could use to post their L2 transaction bundle to L1 in a gas efficient manner. Blob transaction is forwards compatible with Danksharding, and the carried Blobs will be persisted by the consensus layer for a limited period like 4 weeks.

However, since EIP-4844 takes time to implement and deploy (not to mention the full Danksharding), it's observed that some rollups are opting for Alt-DA solutions like Celestia, and Avail, etc [5][6]. Both Celestia and Avail are separate blockchains built on top of different technical stacks and providing their own Data Availability interface.

On the other hand, migrating along with Danksharding is not an easy task for existing rollups like Arbitrum One or Optimism Mainnet, due to reasons like:

- **Interface Change**. Rollups have to adapt their interfaces to use EIP-4844 blob-carrying transactions instead of regular transactions with calldata, otherwise they will not benefit from Ethereum data sharding.

- **Blob Storage Period Limit**. Blobs are only persisted in Ethereum beacon layer for a short period of time such as 4 weeks, and rollups have to either build their own solutions or rely on some third-party solutions for permanent storage. This will remain true even after Danksharding is fully implemented.

EthDA, as a contrast, is an Ethereum native Alt-DA solution, in that:

- EthDA itself is an Ethereum ZK validium, and is administered by a collection of smart contracts deployed on Ethereum L1
- EthDA provides a seamless Data Availability interface as L1 to Alt-DA chain users, allowing their sequencers to pay storage fees with native $ETH

Decentralized Storage

Decentralized storage (dStorage) is another key component of Ethereum and the wider Web3 ecosystem [7]. Unlike a conventional storage system that is run by a single entity or organization, decentralized storage system distributes large amount of data across a peer-to-peer network of operators located in different regions, creating a robust, trustless, and efficient file storage sharing system.

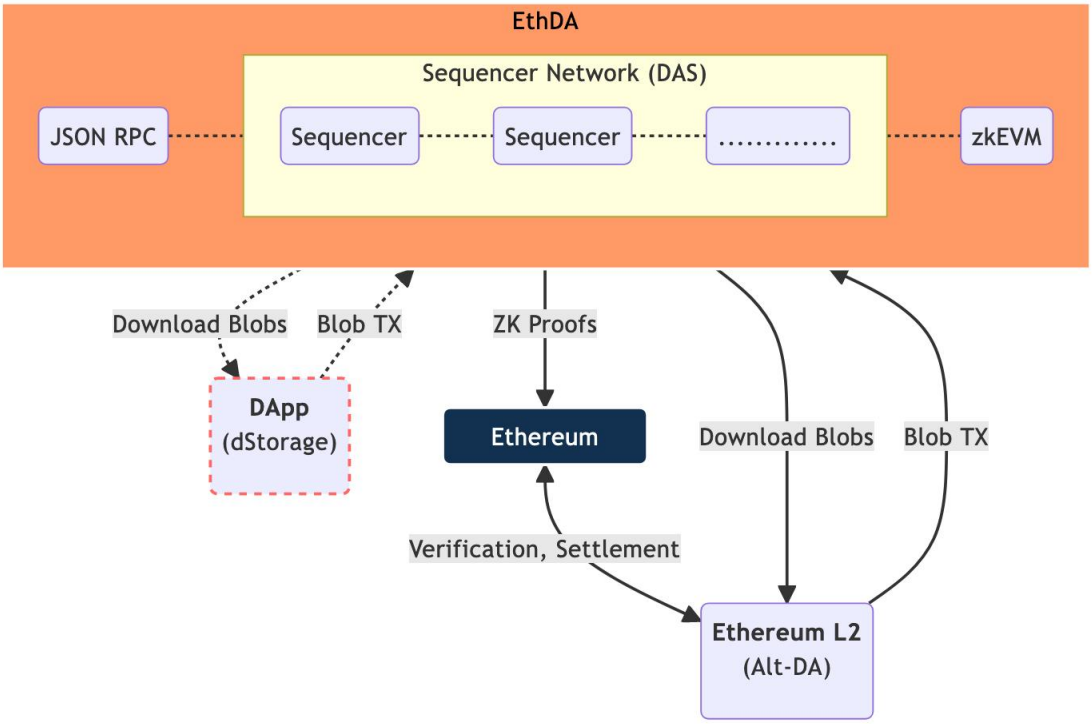
Ethereum itself is an example of a decentralized storage system, and perhaps the most ideal one for DApps of Ethereum ecosystem. However, it's only designed to store small amounts of data such as smart contract bytecode, and the cost is high due to gas fees. For larger amounts of data such as DApp websites, NFT high-resolution images, or even bigger Web2 files, some alternative solutions such as Arweave, Filecoin or Storj are needed. While these are all valuable dStorage solutions, from an Ethereum ecosystem perspective, they have some drawbacks such as lack of on-chain storage attestations, heterogeneous application interfaces and storage fees payment, etc.

Danksharding could be seen as an attempt to scale up Ethereum's data storage capacity, but it is only aimed at storing compressed transaction data of rollups for a limited period of time. This, however, could be extended to support a decentralized storage solution that can store data of any size permanently in an Ethereum native way.

System Overview

EthDA is an Ethereum layer 2 scaling network with native blob-carrying transactions support, aiming to serve as:

- **Alt-DA** for layer 2 chains, which could roll up to EthDA using blob-carrying transactions similarly to how they would roll up to Ethereum layer 1
- **Decentralized Storage** chain for fully-on-chain DApps, or potential replacements for traditional centralized cloud storage



Protocol Overview

EthDA is designed and built based on a set of principles or philosophies.

- **Ethereum Compatibility**: EthDA aims to be an extension or scaling solution for Ethereum, not a totally different technical stack. We will keep maximal compatibility with Ethereum at both the protocol layer and the application interface layer.

- **Simplicity**: We prefer to use existing battle-tested codebase and infrastructures in Ethereum ecosystem whenever feasible, and concentrate on innovating new features on top of that. For instance, we may opt for using Polygon CDK or OP Stack to build EthDA and may reuse Ethereum client modules if feasible.

- **Evolvability**: EthDA will evolve in alignment with Ethereum ecosystem for long-term sustainability.

The whole protocol stack consists of a collection of protocols that could be divided into two layers:

- **Core Protocols**: The ZK validium protocol that makes EthDA as a canonical Ethereum validium chain using Ethereum as settlement layer, along with a set of protocols that enables EthDA to be an alternative Data Availability solution for other Ethereum L2s. Blobs can be stored on EthDA with EIP-4844 blob-carrying transactions, and will be sharded and permanently stored by decentralized sequencers.

- **Peripheral Protocols**: A set of application level protocols that equips EthDA with decentralized storage capabilities for applications. Infrastructures and toolset will be devised to facilitate large sized file storage based on Blobs.

Core Protocols

The core protocols define EthDA as an Ethereum layer 2 ZK validium with a permissionless set of decentralized sequencers based on a soft proof-of-stake consensus mechanism, a DAS scheme among sequencers for blob sharding and storage, as well as a Blobweave scheme to ensure blobs are stored permanently.

L2 Protocols

Similar to other validium, EthDA scales Ethereum by processing transactions off-chain, and publishing ZK proofs on-chain for verification.

The architecture of EthDA consists of a set of on-chain contracts and a group of nodes with different roles.

Layer 1 Contracts

EthDA is essentially administered by a set of smart contracts deployed on Ethereum, including the main contract storing state commitments, and the verifier contract verifying ZK proofs submitted by sequencer nodes. The layer 1 contracts enable EthDA to derive security from Ethereum by using Ethereum as its settlement layer.

Decentralized Sequencer Network

EthDA uses a permissionless decentralized sequencer network to avoid the risk of a single sequencer defrauding the network or censoring users. A sequencer is responsible for accepting and ordering user transactions, constructing and executing layer 2 blocks, and submitting state updates and validity proofs to layer 1.

Sequencers are permissionless, meaning anyone could become a sequencer as long as they provide a bond as a proof-of-stake. The bond will be slashed if the sequencer behaves maliciously such as posting invalid layer 2 transactions to layer 1.

For every epoch, a sequencer will be randomly selected to bundle user transactions, execute them in layer 2 zkEVM, and roll up state update and validity proofs to Ethereum layer 1.

Blob Storage

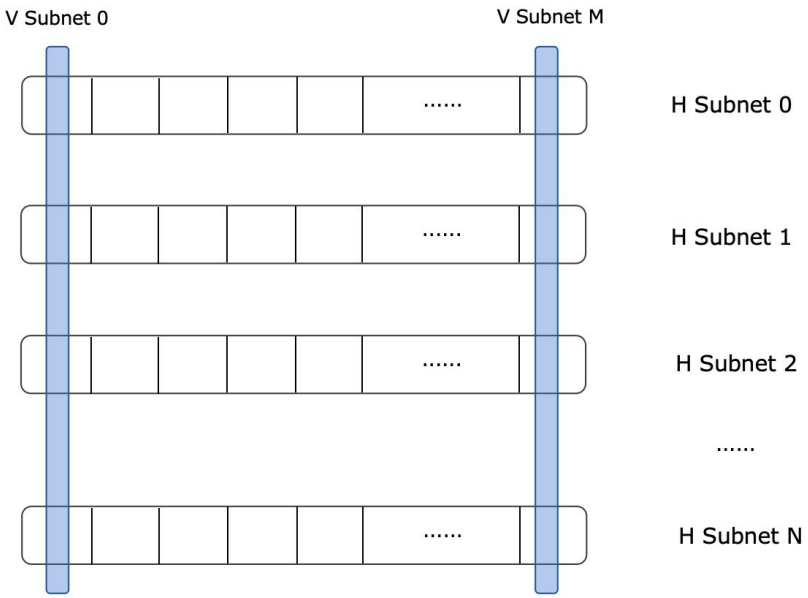
Data Availability Sampling

Data Availability Sampling on EthDA is a mirror of DAS on Ethereum, only that Blobs are permanently stored instead of being purged after a short period of time like 4 weeks.

Sequencers are randomly shuffled and split into N subnets. Since a maximal number of 64 Blobs might be attached to a block per Danksharding, imagine that N = 64. Each Blob is published to a subnet based on their index, and that subnet will be responsible for storing the Blob and ensuring its availability as a whole.

Sequencer node, however, does not need to store the whole Blob assigned to the subnet it belongs to. Instead, a Blob is split into multiple slices and sequencers will randomly sample each other to make sure the Blob is stored by the whole subnet.

The whole subnet architecture is a grid structure of N horizontal subnets and M vertical subnets.

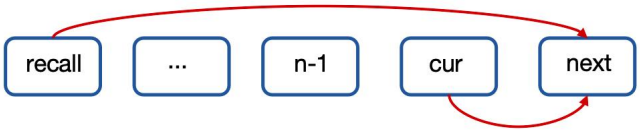


Blobweave

As mentioned above, there is a soft consensus mechanism among EthDA's sequencer network to produce L2 blocks (aka transaction bundles, or sequences) and submit ZK proofs to L1. To support Blob permanent storage by the sequencer network, EthDA uses a scheme called Blobweave similar to Arweave's Blockweave.

Basically, the sequencer network maintains the L2 block hash list, a list of the hashes of all

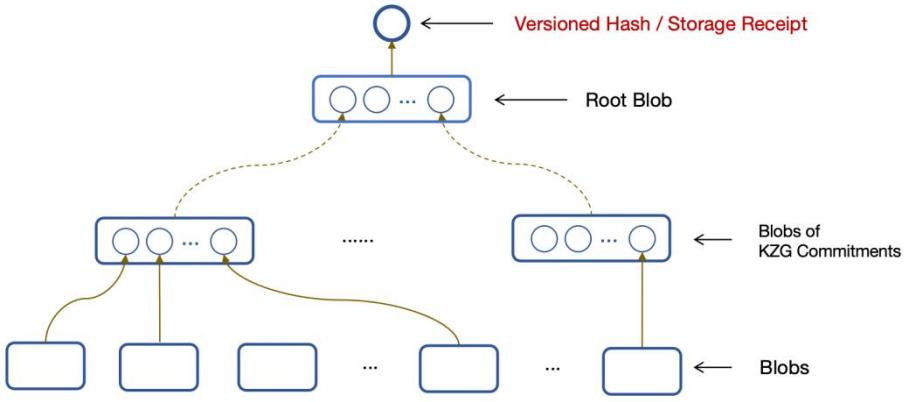
previous L2 blocks. The L2 block hash is calculated based on transaction data bundled into the block, including the carried Blobs. Moreover, a historical block with Blob TX is randomly selected to be the recall block. Transactions of the recall block are hashed alongside to generate the new block hash, thus forming a weave of blobs which we call Blobweave.



Peripheral Protocols

Blob Tree

An arbitrary sized file could be split into Blobs and organized into a Blob tree. The versioned hash of the root Blob could be used as the storage receipt of the file.



blob:// Protocol

A Blob access protocol could be defined as below:

blob:///[blob-versioned-hash]:[chain-id]

Blob data could be auto-detected to see whether it contains KZG commitments of child Blobs, and if it does, the whole Blob tree could be parsed recursively to compose the whole chunk of data.

Infrastructures

A set of infrastructures and toolset are defined and provided to facilitate developers and users of dStorage users.

dStorage Contracts

Smart contracts to record and verify all storage orders.

Client SDK

A set of client SDK implemented in various languages (JavaScript, Python, etc) to support file storage and retrieval.

A complete e2e scenario comprises of the following stages:

- Preprocess: Split file into Blobs, compute intermediary Blobs of KZG commitments, and the top Blob's versioned hash as the file's storage receipt.
- Upload: Upload Blobs one by one to EthDA, each via a blob-carrying transaction. And post storage receipt to dStorage contract.
- Retrieval: Parse and download Blobs recursively, and compose the file on the client side.

Retrieval Gateway

An infrastructure to facilitate users to retrieve or render Blobs or multi-Blob files using https protocol. Imagine it like IPFS Gateway. Anyone could download the precompiled image and operate their own gateways, for scenarios like:

- File download.
- DApp rendering. DApp website could be hosted on EthDA, and be accessed via https URL in a browser with the help of a retrieval gateway.

Browser Extension

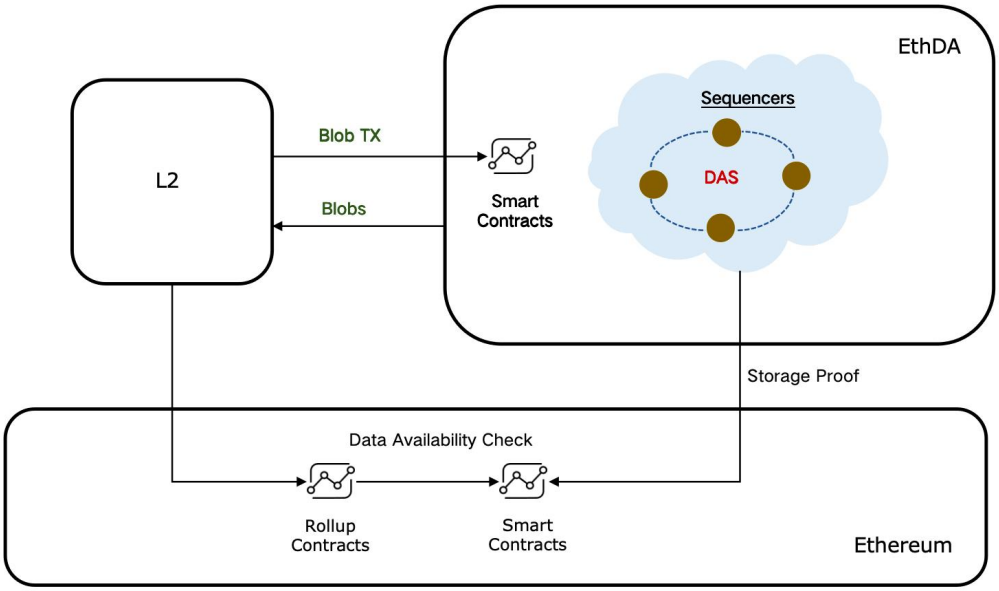
An alternative to retrieval gateway, but enable Blobs to be accessed directly via blob:// protocol.

Alt-DA

EthDA is specially designed for Ethereum L2s as an alternative Data Availability layer. L2s should store transaction data to EthDA using EIP-4844 blob-carrying transactions.

Blobs are stored on EthDA using DAS and proof-of-random-access, and storage proofs are

submitted to layer 1 smart contracts. L2s' contracts on layer 1 could then use the storage proofs to check data availability. And verifiers could download transaction Blobs from EthDA on demand.



Fully-on-Chain DApp

A typical Web3 DApp comprises the following parts:

- **Smart Contracts**: Core component to keep track of user assets and operations, could be deployed on layer 1, layer 2 or any other EVM compatible chains

- **Website**: User interface to help interact with smart contracts. Could be deployed on a cloud server operated by some centralized organization, or on IPFS nodes which could be operated by some organizations like Pinata or be part of a decentralized network with incentive mechanisms like Crust Network.

- **Backend or Indexing Service**: Aggregating data from smart contracts, or performing some off-chain operations. This could be some Web2 like services, or some decentralized services like The Graph.

Also, some other infrastructures or points need to be considered:

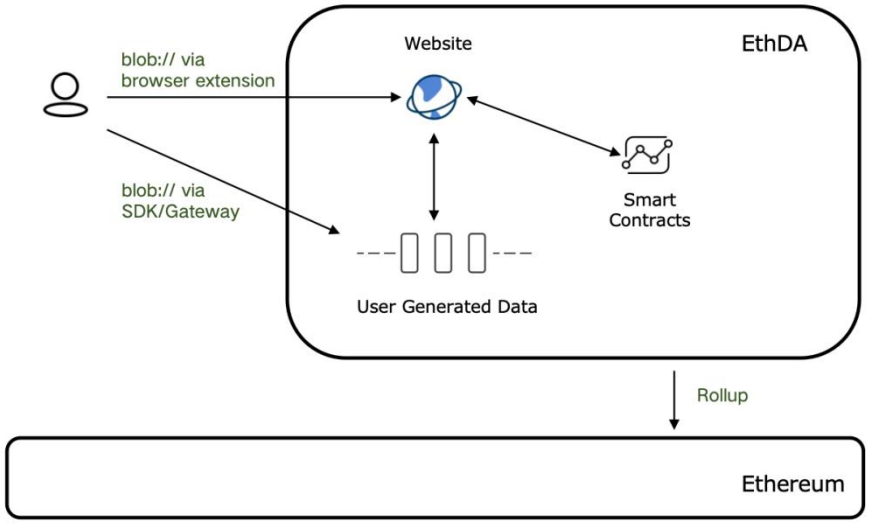
- **Domain Name**: A traditional domain plus some bridging solutions like DNSLink, or some Web3 protocols like ENS name, need to be used to bind to DApp address to facilitate user access [8].

- **UGC**: User Generated Contents also need some storage solutions to store. It could be a

S3 object storage service, some decentralized solutions like IPFS, or even blockchains like Arweave.

All these components or subsystems above, could be centralized or decentralized. And for the decentralized ones, they could be on-chain or off-chain. Nevertheless, there is no doubt a continuous effort in the community to bring DApps towards decentralization and ideally fully-on-chain from all aspects.

EthDA could help turn this vision into reality. Smart contracts and websites could be both deployed on EthDA, and user generated contents could also be stored on EthDA via the dStorage protocol. Backend service is optional, and the necessity is alleviated due to storage capacity extension. Also websites and UGC data could be accessed via blob:// protocol without binding to a domain or ENS name, making a fully-on-chain DApp totally feasible.



Conclusion

In this article, we have presented a new Ethereum layer 2 chain, EthDA, which is specially designed for scaling Ethereum as a Data Availability network for L2s and a decentralized storage system for applications. This is achieved by natively supporting blob-carrying transactions on layer 2, plus a mirror of Data Availability Sampling mechanism of Ethereum Danksharding for Blob sharding and permanent storage among a permissionless set of decentralized sequencers.

Also a set of peripheral protocols are designed for permanently storing arbitrary sized files based on Blobs, making EthDA an ideal solution to build fully-on-chain decentralized applications.

References

[1] Data Availability. https://ethereum.org/en/developers/docs/data-availability

[2] An Incomplete Guide to Rollups. https://vitalik.ca/general/2021/01/05/rollup.html

[3] Danksharding. https://ethereum.org/en/roadmap/danksharding/

[4] EIP-4844. https://www.eip4844.com/

[5] Quantum Gravity Bridge: Secure Off-Chain Data Availability for Ethereum L2s with Celestia. https://blog.celestia.org/celestiums/

[6] Avail-Powered Validiums. https://docs.availproject.org/about/introduction/validiums/

[7] Decentralized Storage. https://ethereum.org/en/developers/docs/storage/

[8] https://dnslink.dev/