EthStorage

Scaling Ethereum Storage via Layer 2 and Data Availability

Qiang Zhu
EthStorage.io

EthStorage: Motivation

- Major goal: Reuse Ethereum security properties and extend Ethereum storage capabilities

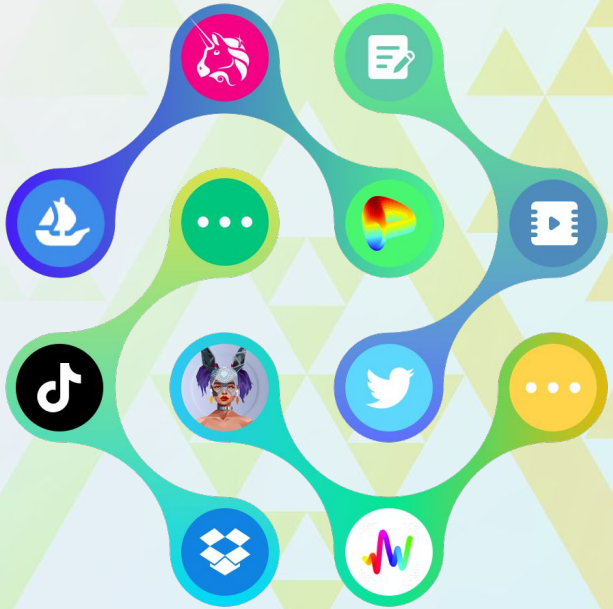
- Key Idea: Build a **storage-specific** Rollup vs **EVM-specific** Rollup
  - Reduce Ethereum storage cost to ~**1/1000x** (vs SSTORE)
  - Scale storage capacity to **PB** (vs ~TB SSTORE) with **CRUD** KV store semantics
  - Storage fee paid with ETH
  - Reuse Ethereum security properties with **proof of storage on L2 data**

	EthStorage L2	EVM-Specific Rollups
Type of Off-Chain Tx	CRUD operation with a large-sized value	EVM Tx
L2 World State	A list of BLOBs written by off-chain Txs	MPT of the off-chain EVM
L1 Rollup Batch Data	List of CRUD operations with large-sized values (each value is a DA BLOB)	List of off-chain EVM Txs as DA BLOBs
L1 Rollup State Root	Hash of each value (BLOB)	Root Hash of the MPT
Require a sequencer?	No, a user can directly submit Tx to L1	Yes, a user must submit a Tx to a sequencer
L2 World State Availability Assumption	At least one L2 node is honest	At least one L2 node is honest
Number of Physical Replicas of L2 World State	**~100 physical replicas**	No such enforcement

	Filecoin	Arweave	Ethereum SSTORE/SLOAD	EthStorage
Store Object	Static Files	Static Files	KV Store	
Semantics	CRD	CR	CRUD	
On-Chain Programmable	∅	∅	✓	✓
Proof of Storage	Proof of Space-Time with Challenge	Succinct Proof of Random Access	Fully Replicated	Dynamic Sharding with Proof of Random Access
Replication Guarantee	High	Median	Very High	High
Storage Cost	Very Low	Low	Very High	Low
Capacity	~ EB	~ 100 TB (Currnet)	~ 1 TB	~ PB
Access Protocol	ipfs://	N/A	web3://	
Wallet	Filecoin Wallet	ArWallet	ETH-Compatible	

Applications

- Long-term storage solution for other Rollups
  - OR, ZKR
- Fully Decentralized frontend with dynamic websites
  - FILECOIN/AR can only do static ones
- Native storage for NFTs



Technologies

Proof of Publication via Data Availability

Danksharding with KZG Commitment and Reed-Solomon Code

DA Research Grant from EF

EthStorage: External Data Retention L2 Network

Proof of Storage on Large Dynamic Datasets

~ PB Capacity with CRUD

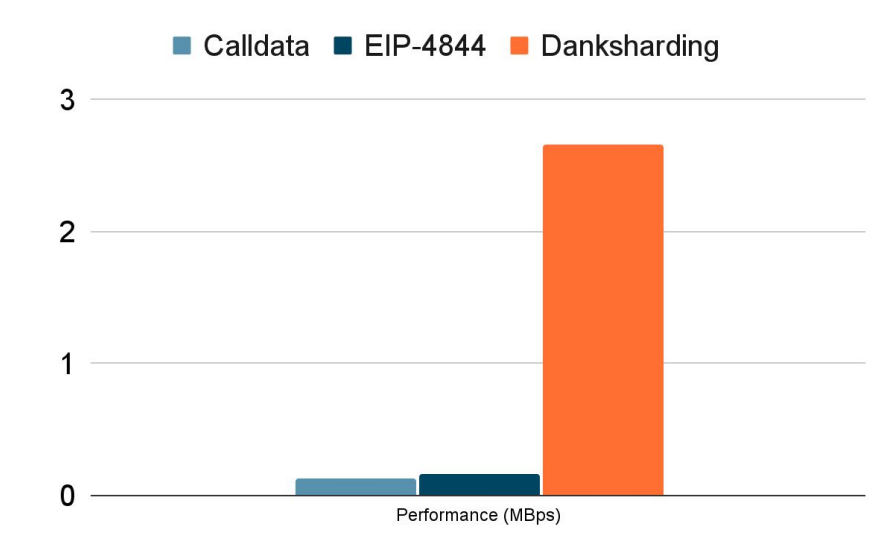
Proof of Storage Grant from EF

web3:// Access Protocol

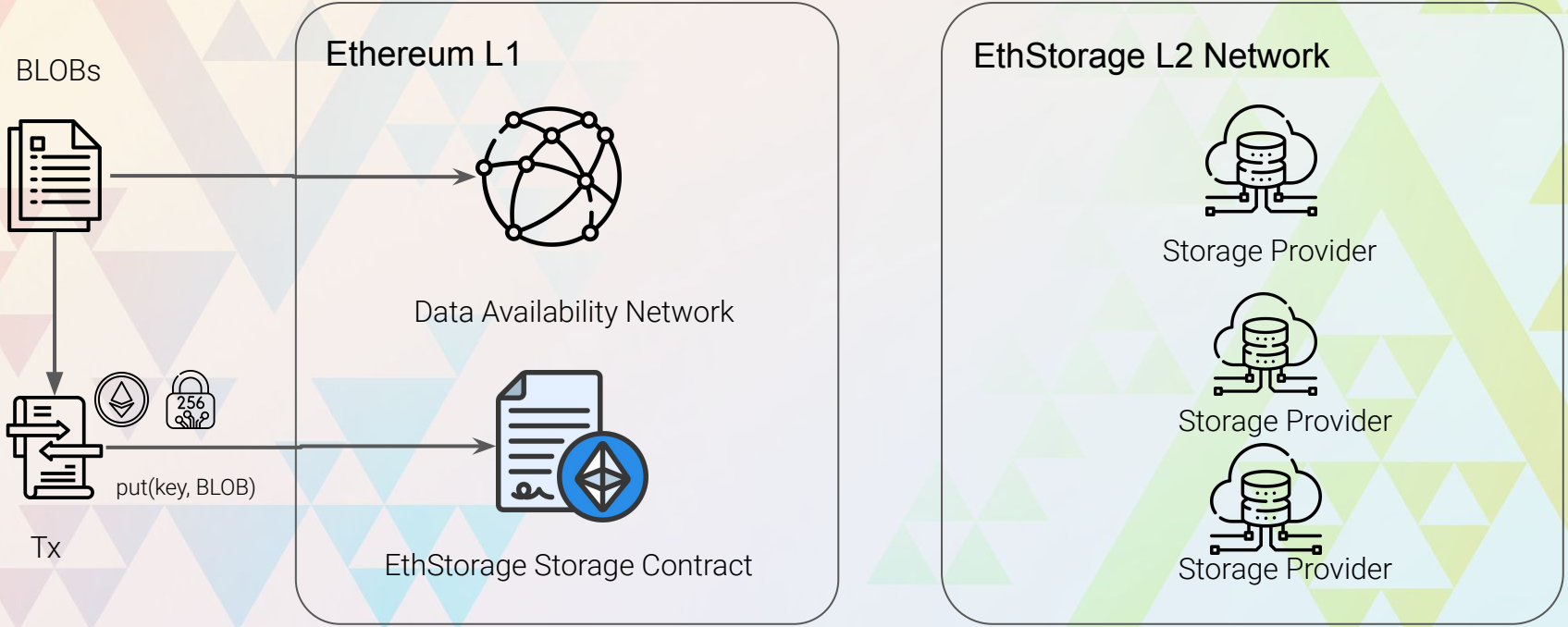
Decentralized Access to Dynamic Web Objects Hosted by Smart Contracts

Ethereum Data Availability

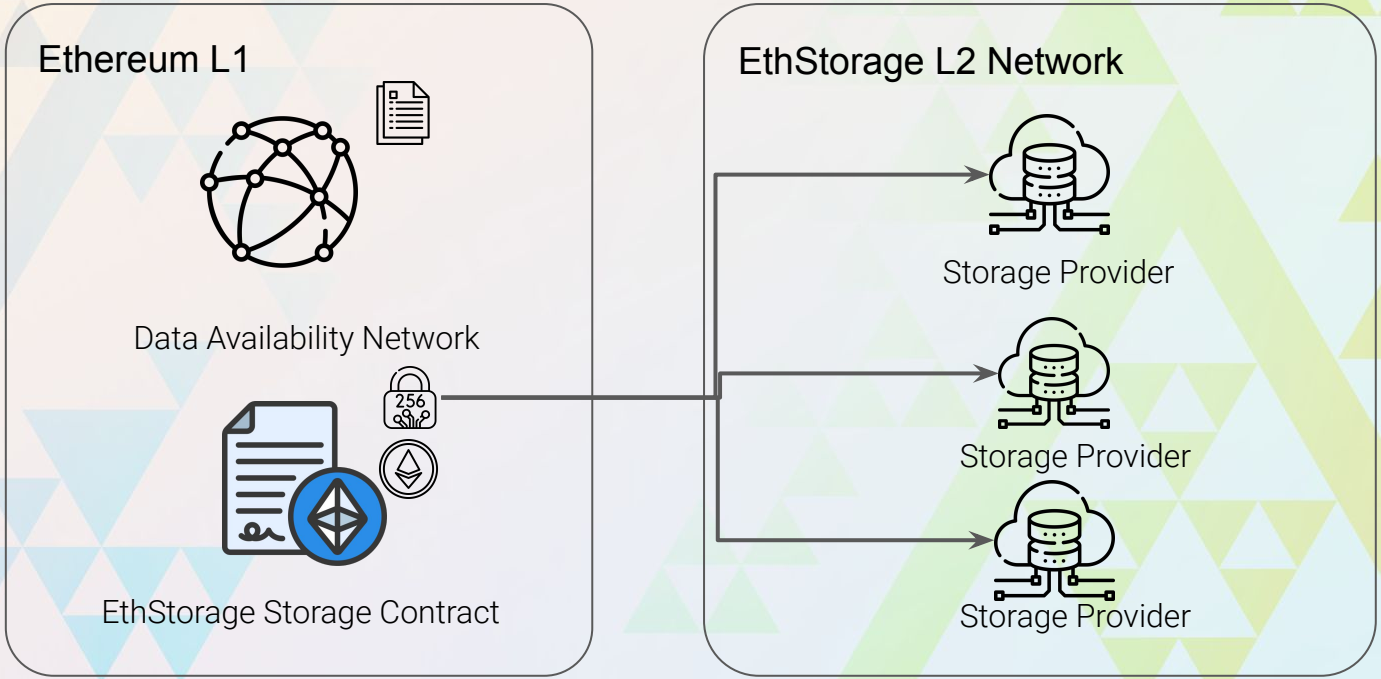
- Danksharding: Increase Ethereum L1 data upload speed
  - Using KZG commitment and Reed-Solomon code
  - **20x vs calldata**
- Danksharding DA will create **1.5TB BLOB per week**
- BLOBs will be expired in about **1 month**
- DAS grant: Optimize RS code reconstruction to **3x**
  - https://ethresear.ch/t/reed-solomon-erasure-code-recovery-in-n-log-2-n-time-with-ffts/3039



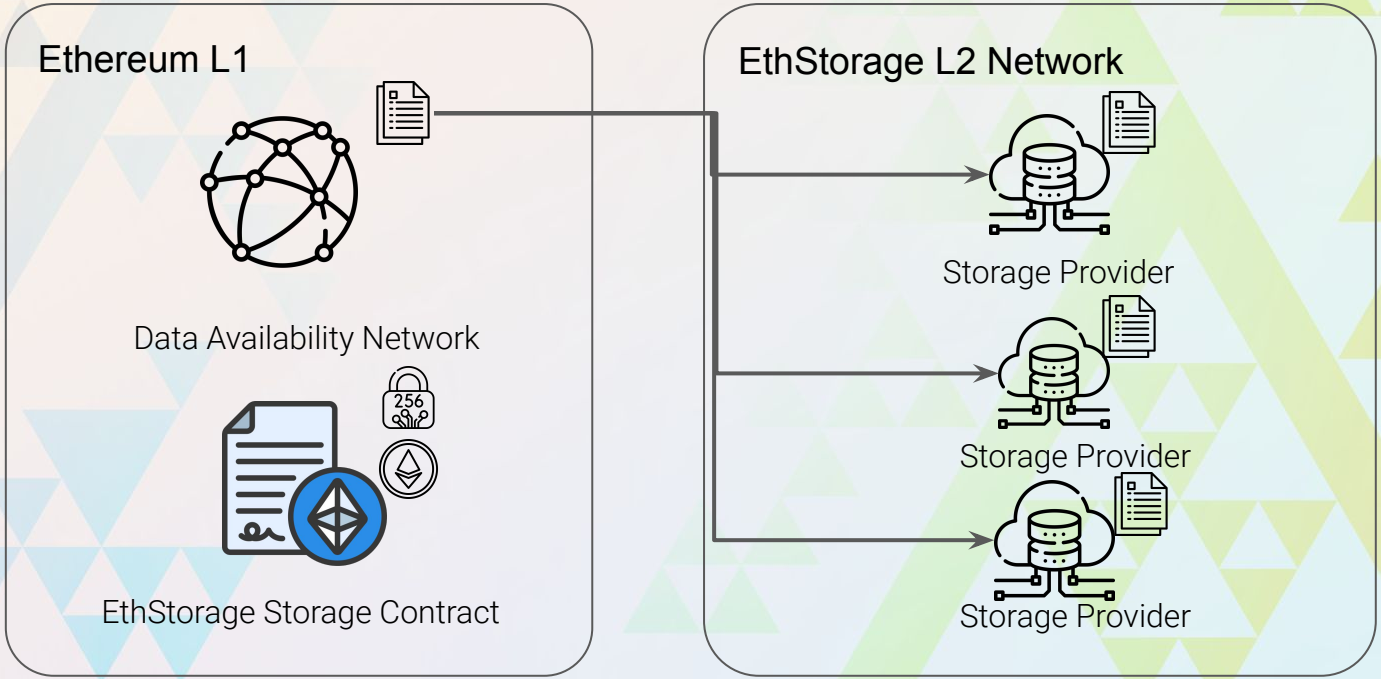
EthStorage: Data Upload with DA



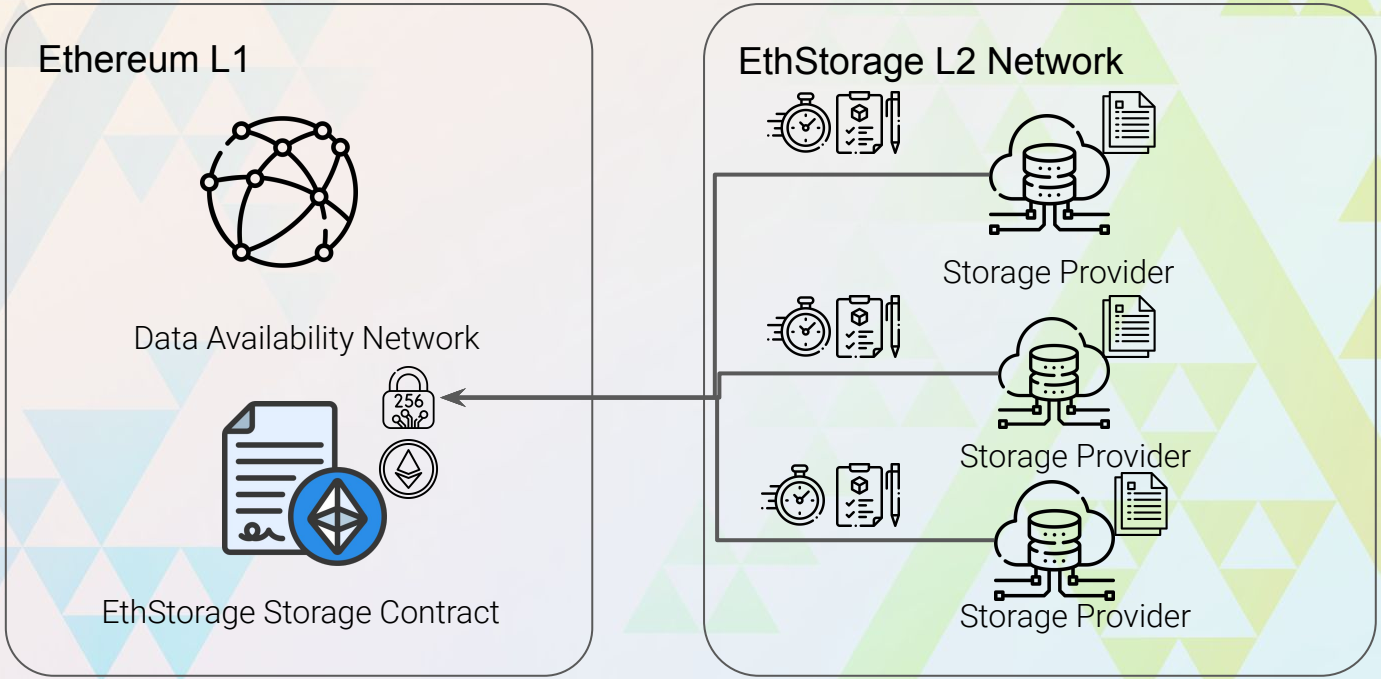
EthStorage: Data Upload with DA



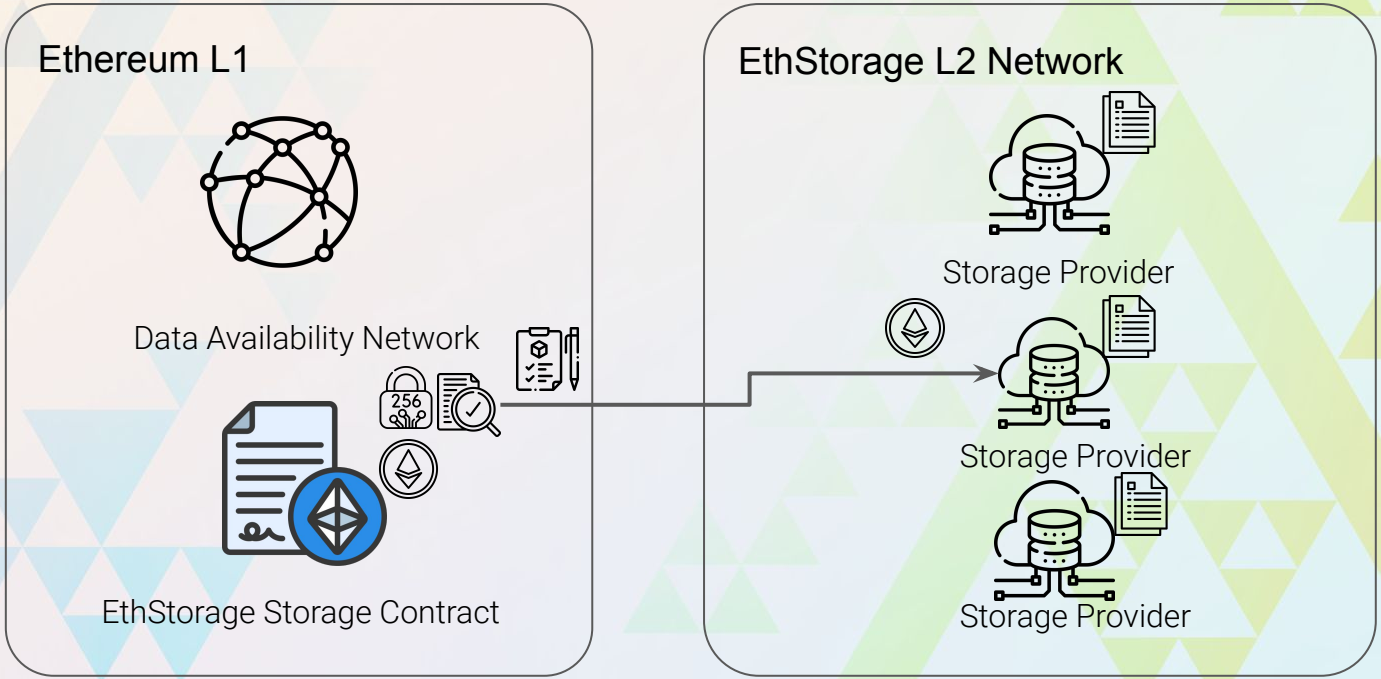
EthStorage: Data Upload with DA



EthStorage: Proof of Storage with L1 Contract



EthStorage: Proof of Storage with L1 Contract



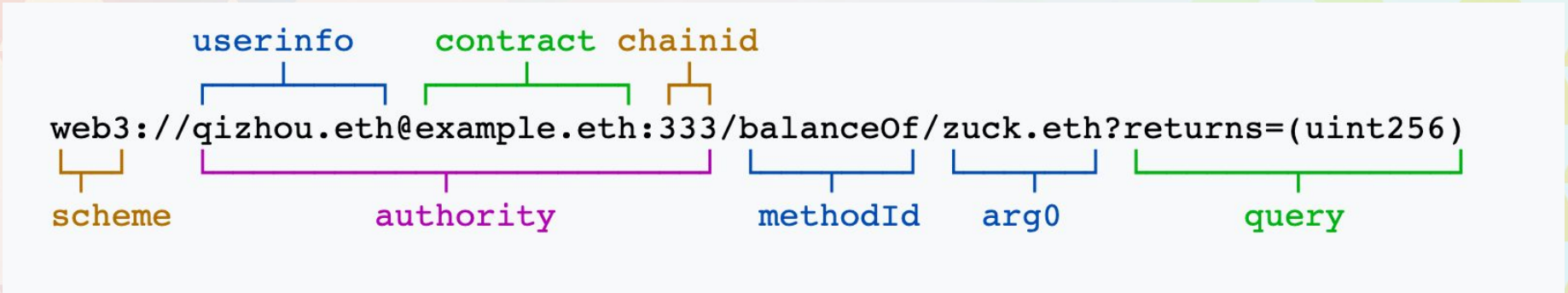
Proof of Storage on Dynamic L2 Datasets with an Ethereum L1 Contract

- Supported by ESP grant https://blog.ethereum.org/2023/02/14/layer-2-grants-roundup
- Keywords: **Dynamic datasets** and **verification on L1 contract**
- Key Technology: **zkNoSQL**

- Ethstorage - Proof of Storage on L2 Dynamic Datasets with an Ethereum L1 Contract
  - Given a list of commitments of BLOBs (e.g., KZG commitments from EIP-4844/Danksharding, indexed from 0...n-1) in an L1 contract, the project is to study an efficient proof system to efficiently verify on L1 that the BLOBs are stored off-chain with the desired redundancy (e.g., 30~50 physical replicas). By putting these storage nodes in an L2 network and assuming 1/m of nodes is honest, we could build an L2 storage network that reuses mainnet security while extending Ethereum scalability dramatically.
  - Further research will attempt to answer the questions of how to build the proof/verification system if the BLOBs and the commitments are constantly changed (e.g., new BLOBs are appended to the list or a BLOB of an index is changed); and how to build incentivized/payment system to ensure desired replication factor using ETH as payment; it is possible to ask a storage node to store partial BLOBs while maintaining the same level of security of Mainnet?

web3:// Access Protocol

- ERC-4804: Web3 URL Standard - an IANA Registered Scheme
- Render Web Objects Hosted by Smart Contracts (or hash linked to EthStorage)



Early Experiments and Community Ideas

- Vitalik's blog uploaded to Arbitrum Nova with 0.13ETH
  - https://www.reddit.com/r/ethereum/comments/107ok8e/upload_40mb_vitaliks_blog_to_a_smart_contract_on/

- Decentralized Reddit pixel war
  - https://twitter.com/qc_qizhou/status/1615233757207990272

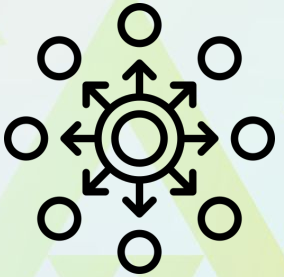
- git3 (decentralized git), w3box (decentralized dropbox), w3mail (decentralized email), ...

BlockEden.xyz

Future Web3 Vision

Ethereum Drives Most Innovations

Omnipresent to Everyone



Fully Decentralized

Timeline of EthStorage

2022
PoC Testnet

- ERC-4804: Web3 URL Standard
- Technical White Paper
- web3:// Gateway
- BFT + PoS Sidechain Consensus
- Devcon Talk

2023
Full Functional Testnet

- Storage Contacts
- Data Retention Network
- Permissionless Staking and Validator Change
- BLOB Libraries
- 2-3 Community Projects
- Verifiable Gateway

2024
Pre-Launch*

- Full Integration with Danksharding
- EthStorage CL + EL Clients
- web3:// Browser Native Integration

Thank you!

Qiang Zhu
zhuqiang@ethstorage.io

@qzhodl

CREDITS: Icons from **flaticon**