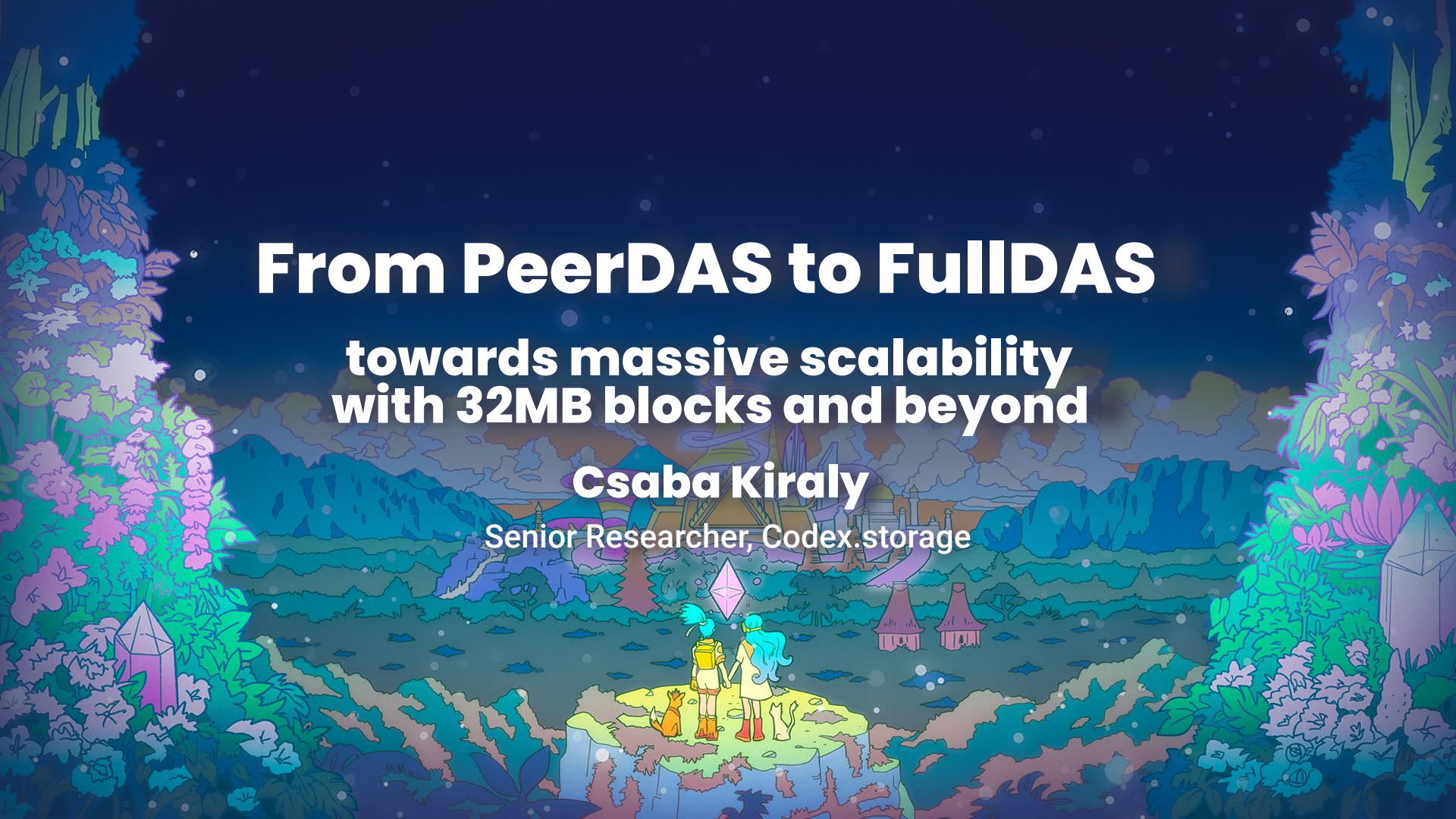
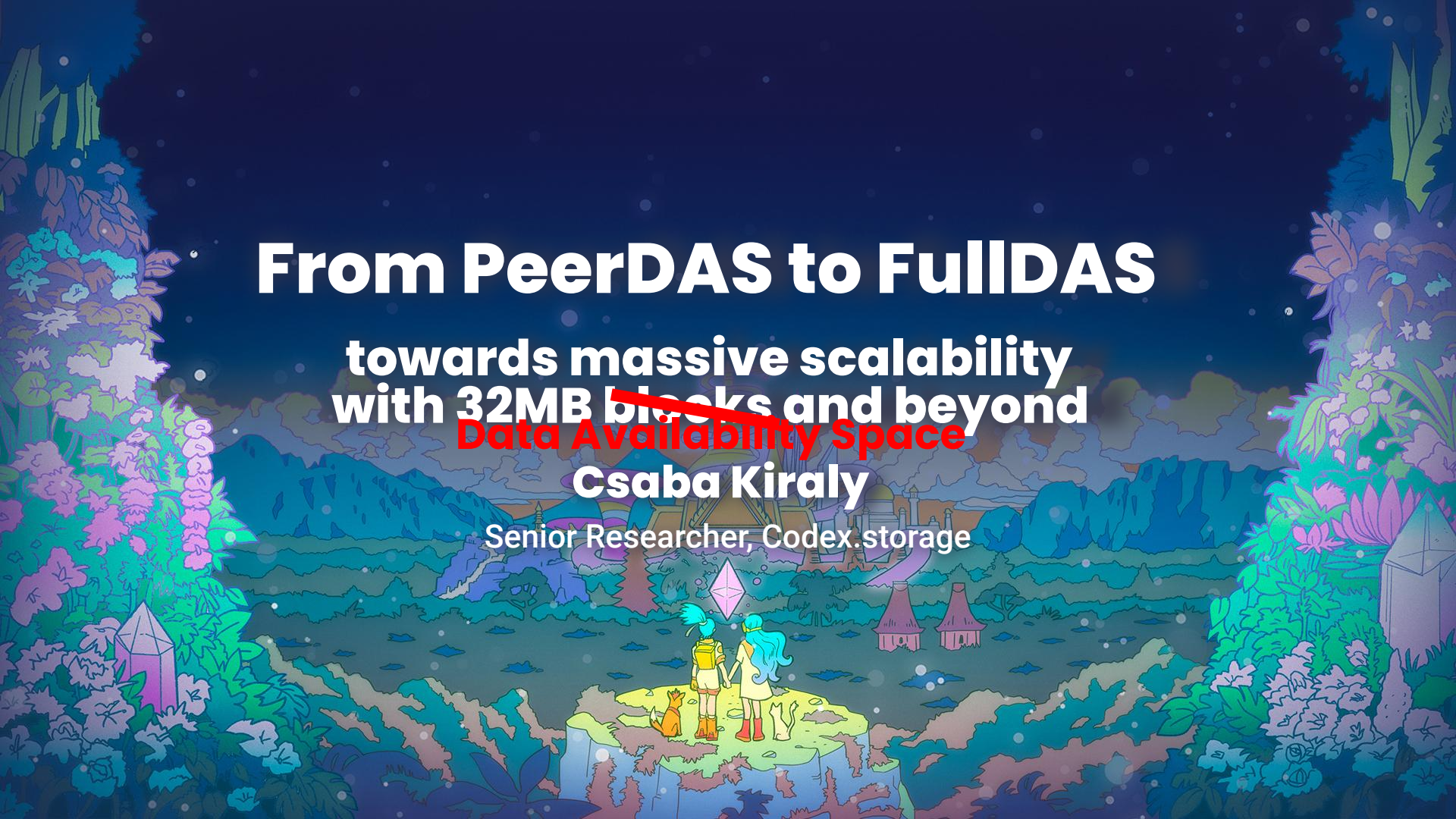

From PeerDAS to FullDAS
towards massive scalability
with 32MB blocks and beyond
Csaba Kiraly
Senior Researcher, Codex.storage

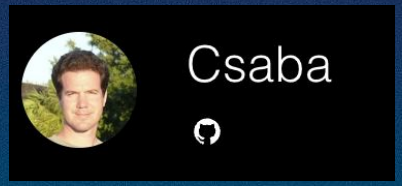

From PeerDAS to FullDAS
towards massive scalability
with 32MB blocks and beyond
Data Availability Space
Csaba Kiraly
Senior Researcher, Codex.storage

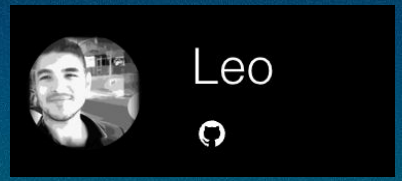
Codex

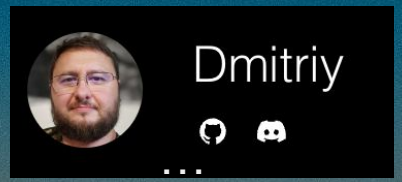
Data Durability Engine

Who we are

Codex Research







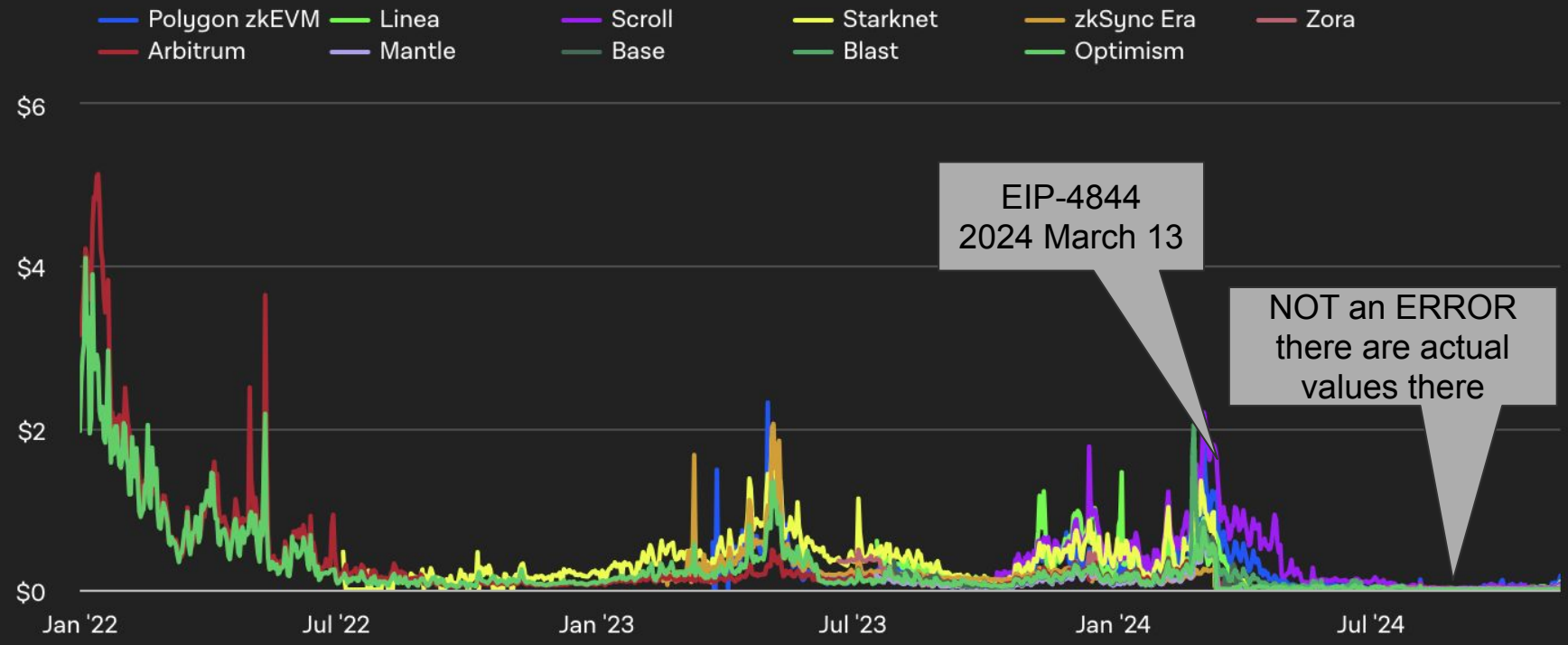
...

Working on DAS with

ethereum foundation grant

& CL client teams

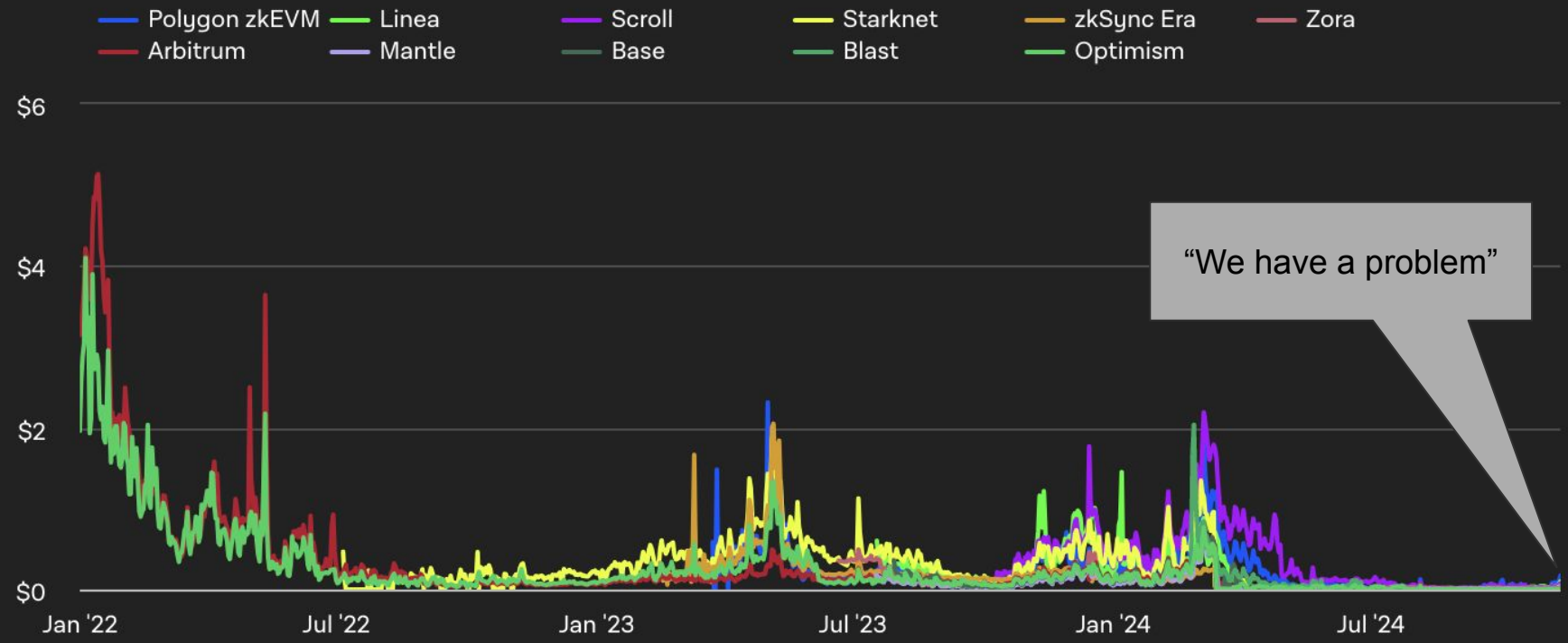
Layer 2 (L2) Median Transaction Fees (Daily)



SOURCE: GROWTHEPIE
UPDATED: NOV 13, 2024

ZOOM ALL YTD 12M 3M 1M

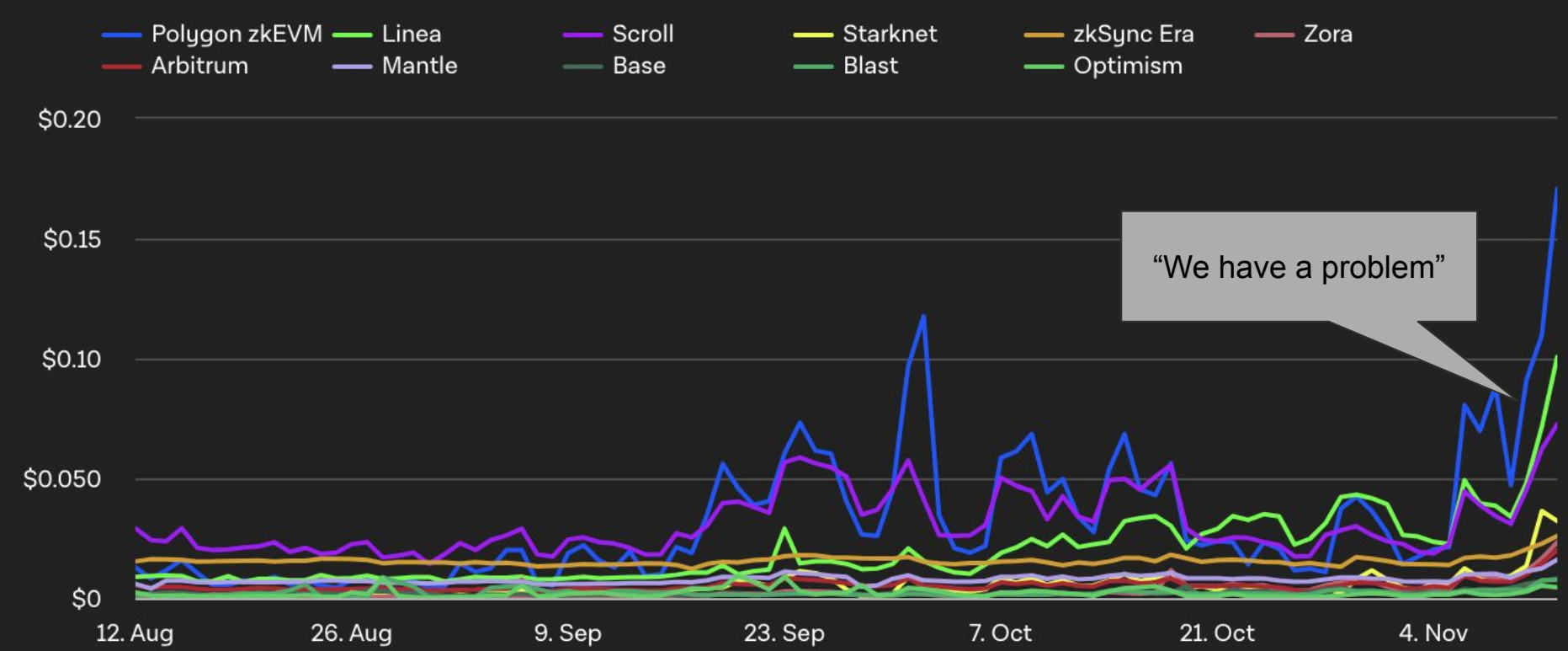
Layer 2 (L2) Median Transaction Fees (Daily)



SOURCE: GROWTHEPIE
UPDATED: NOV 13, 2024

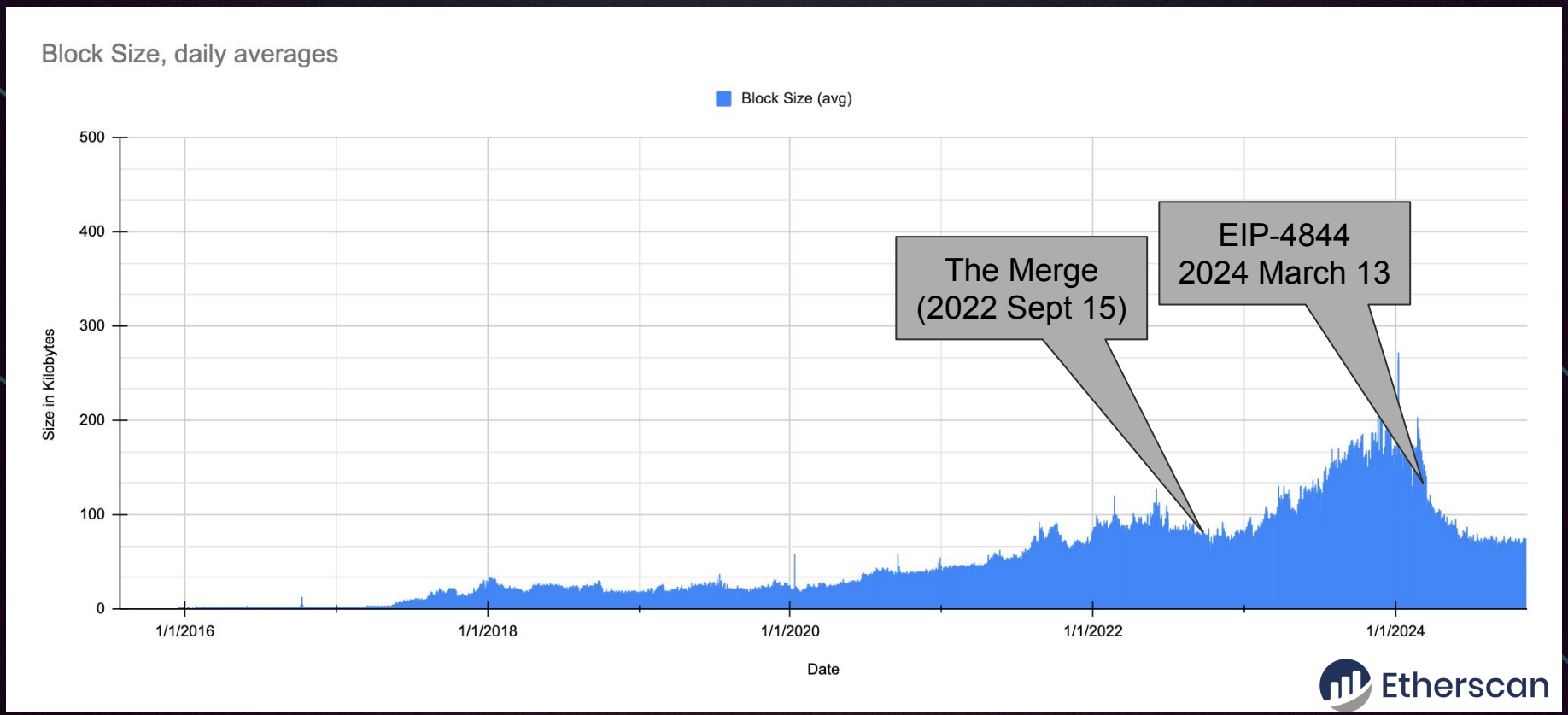
ZOOM ALL YTD 12M 3M 1M

Layer 2 (L2) Median Transaction Fees (Daily)

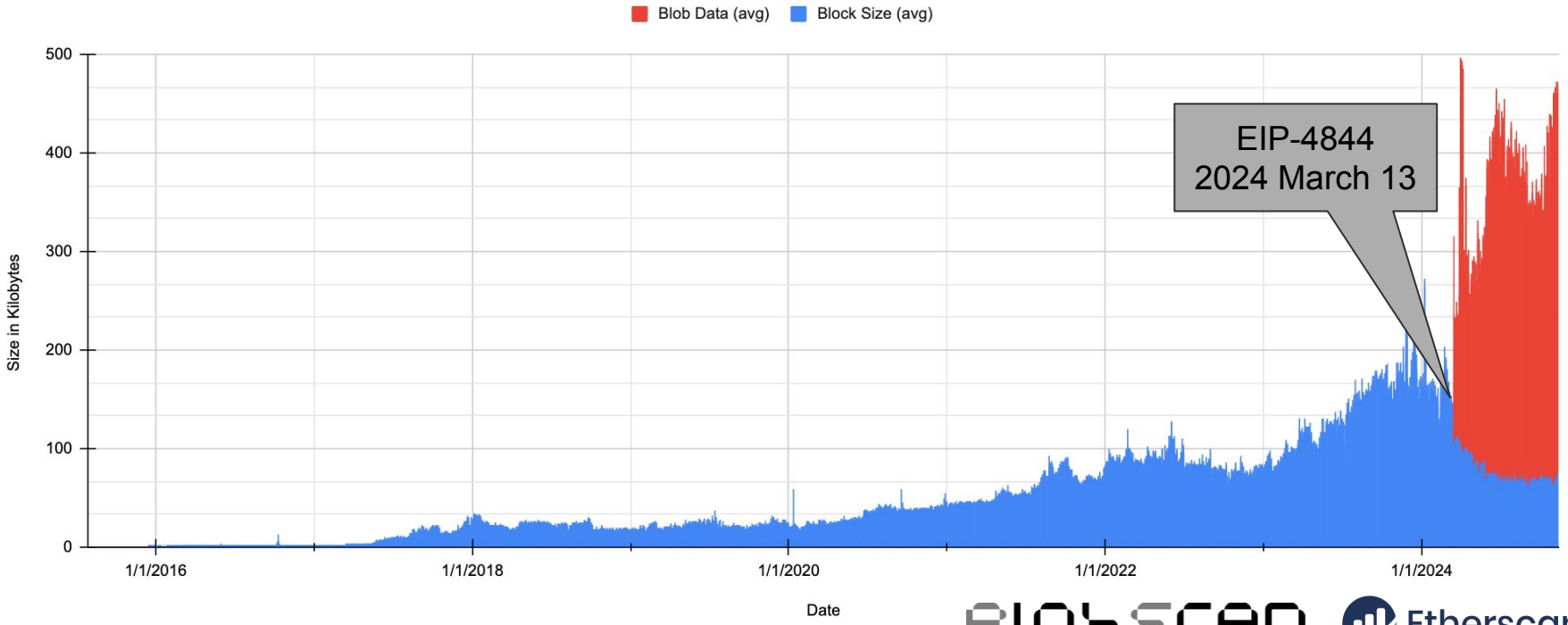


SOURCE: GROWTHEPIE
UPDATED: NOV 13, 2024

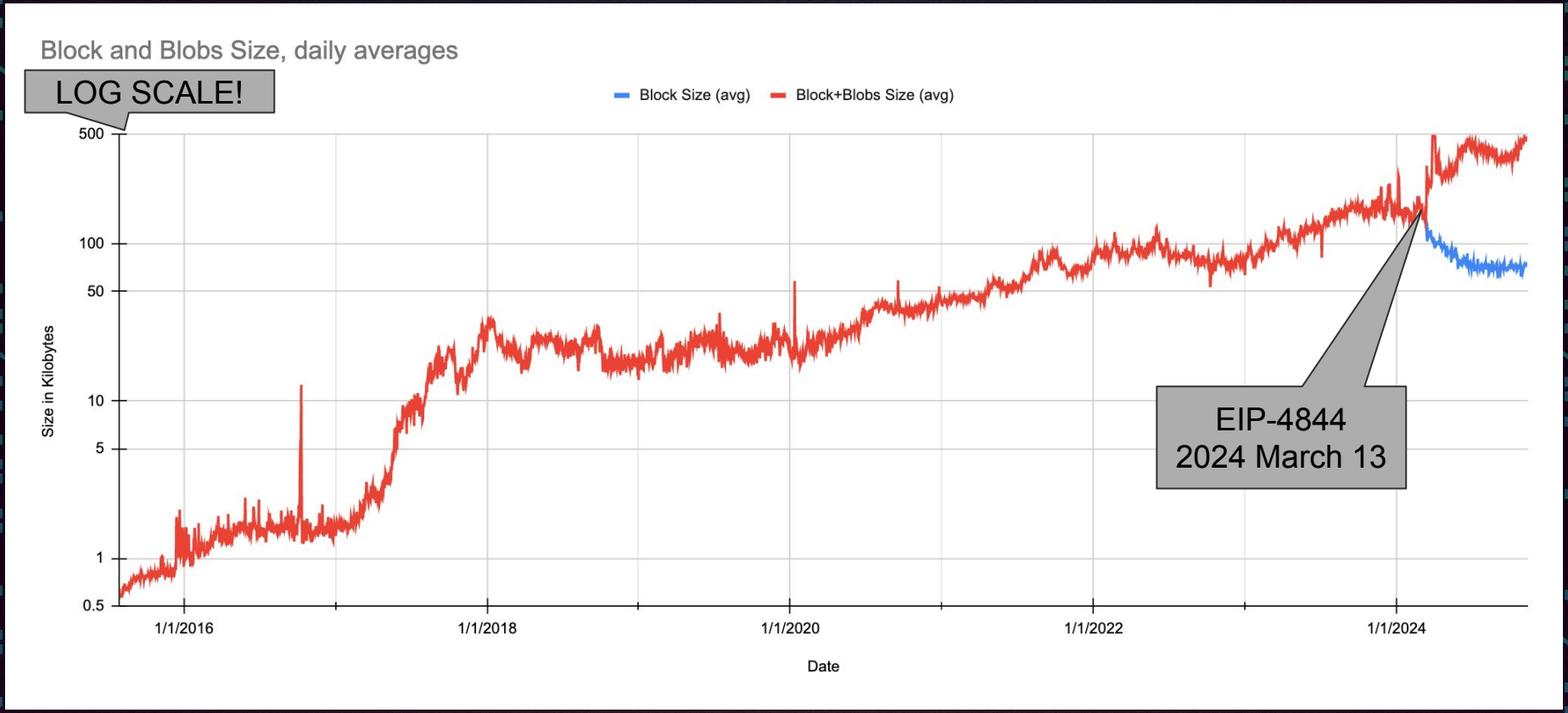
ZOOM ALL YTD 12M 3M 1M


Block Size, daily averages
Block Size (avg)
500
400
300
200
100
0
Size in Kilobytes
1/1/2016
1/1/2018
1/1/2020
1/1/2022
1/1/2024
Date
The Merge
(2022 Sept 15)
EIP-4844
2024 March 13
Etherscan

Block Size and Blob Data, daily averages

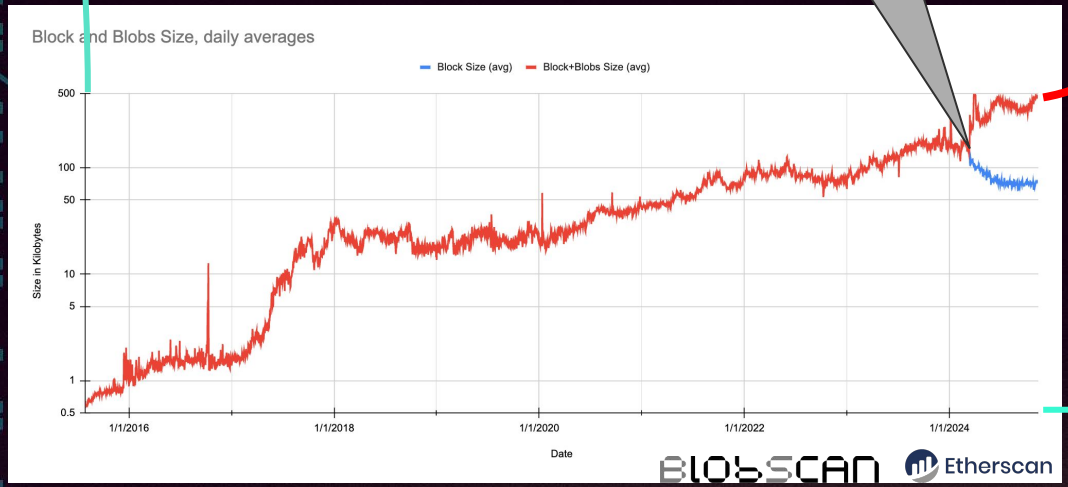


BLOBSCAN Etherscan


Block and Blobs Size, daily averages
LOG SCALE!
Block Size (avg)
Block+Blobs Size (avg)
500
100
50
10
5
1
0.5
Size in Kilobytes
1/1/2016
1/1/2018
1/1/2020
1/1/2022
1/1/2024
Date
EIP-4844
2024 March 13

EIP-4844
2024 March 13

More Blob
Space



UNKNOWN SCALE

EIP-4844
2024 March 13

PeerDAS
[date]

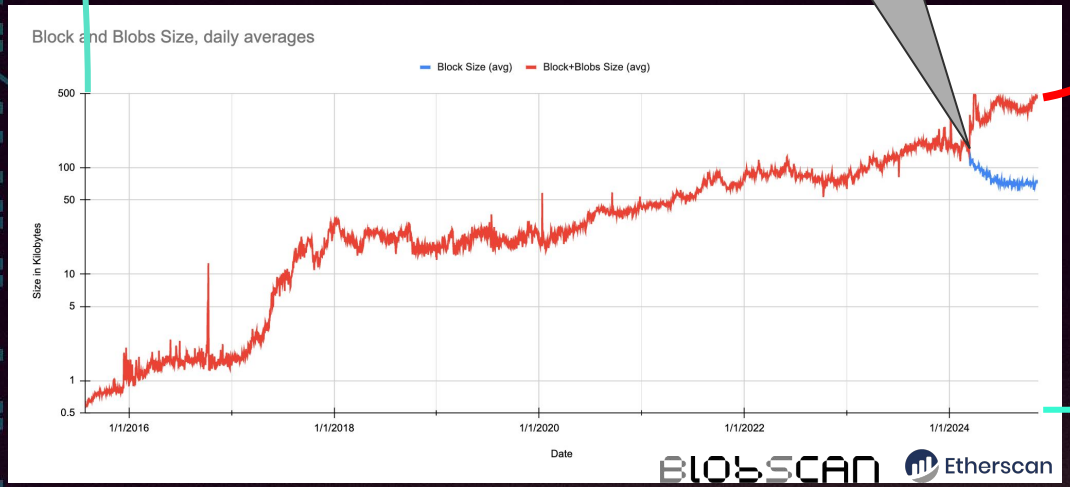
FullDAS
[another date]

Even More Blob Space

More Blob Space

Less Node Traffic

Even Less Node Traffic



UNKNOWN SCALE

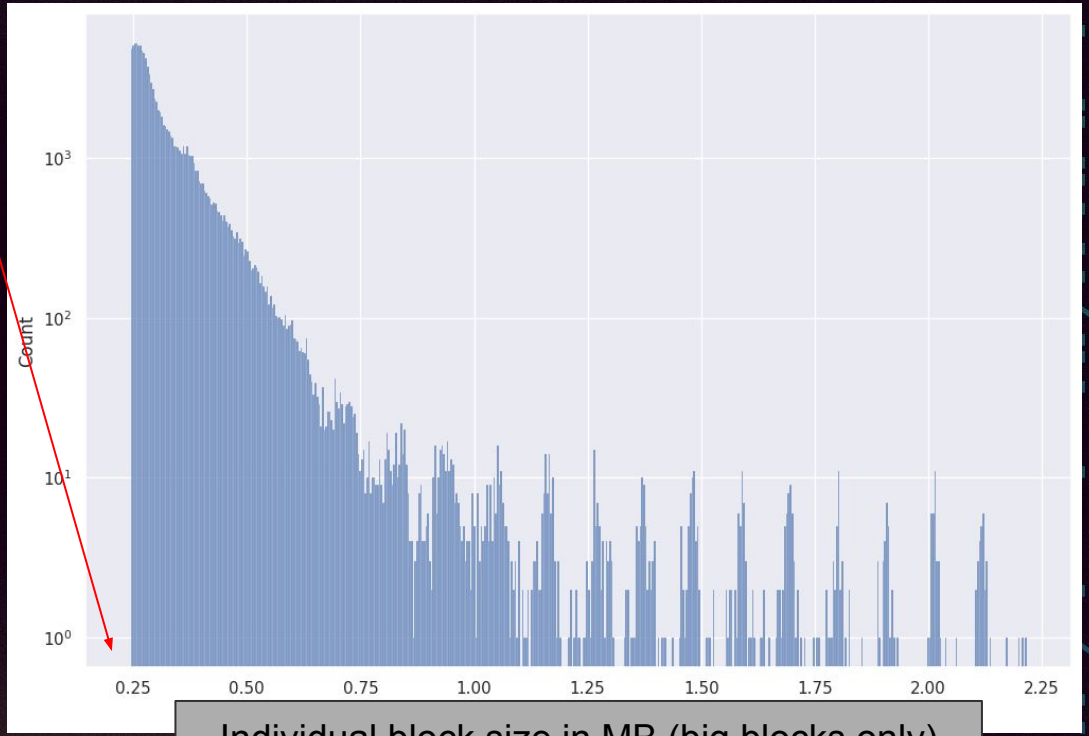
Scaling the L1 Network

Average block size ... and the tail distribution, without averaging

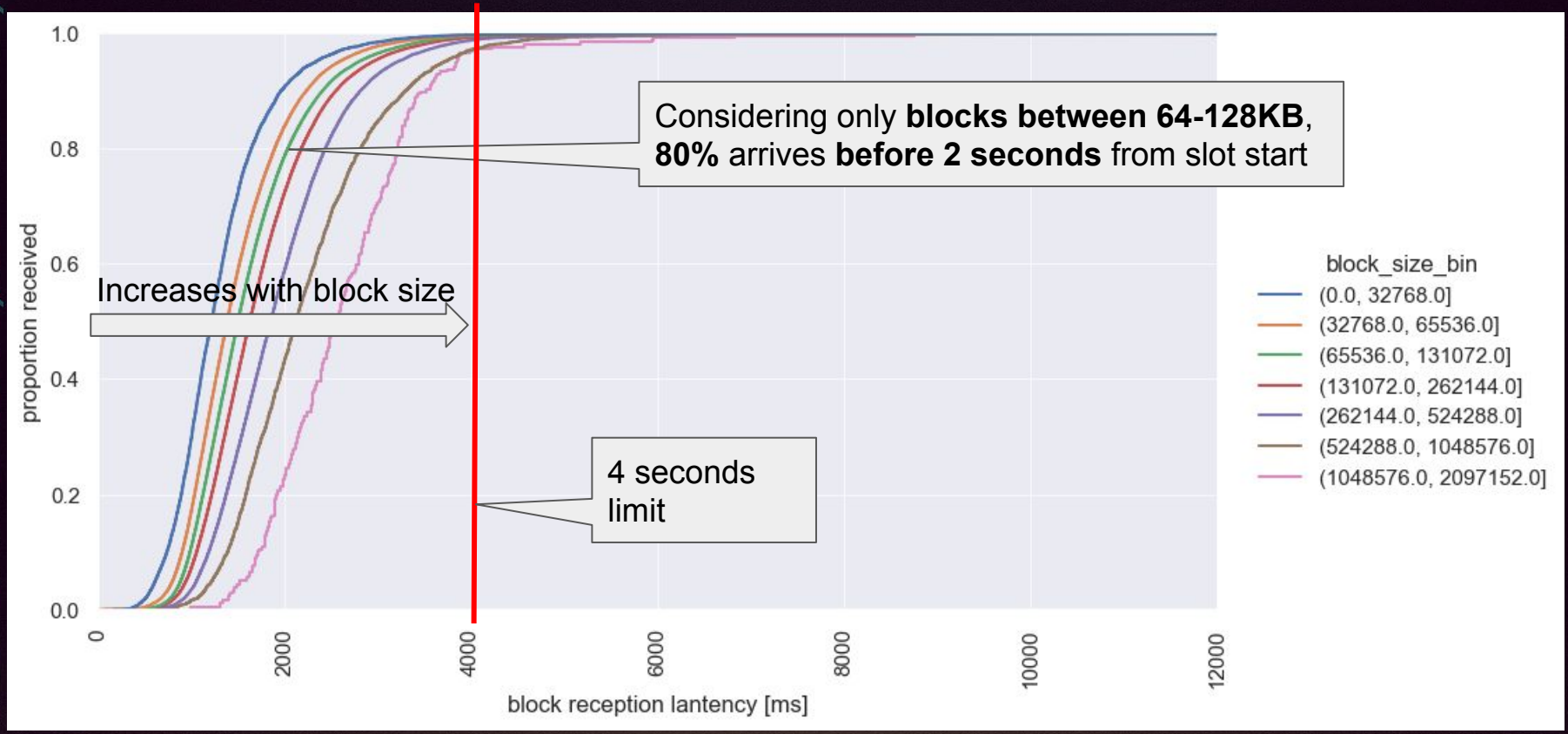
Average block size, ~100 KB

Big (>250KB) blocks on mainnet

- Big blocks injected for experimentation (2023 July)
- “Organic” big blocks observed later



Block Propagation Latency



Path to L1 Scalability

1. Changing the **slot timing window**, changing **resource requirements**
   - Controversial
   - Small gains, but lot to lose

2. **GossipSub** improvements for large messages
   - Important, but not my talk today
   - Relatively small gains

3. **Sharding** and **Data Availability Sampling** (DAS)
   - Large gains

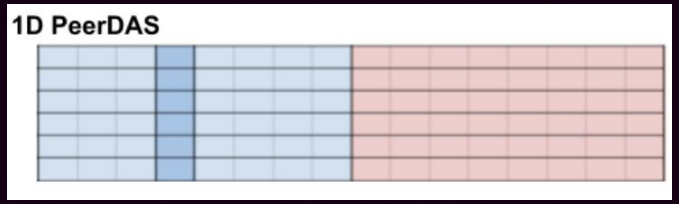
4. **Distributed Block Building**
   - Intertwined, but large gains
   - Rethinking data flow for the whole blob transaction lifecycle

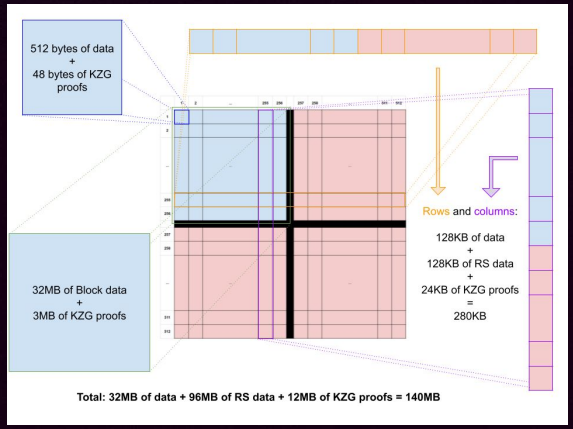
Sharding and DAS over the Network

DAS key concepts

Data Preparation

- **Blobs** (a.k.a. data availability space)
- **Encoding** (RS code and KZG commitments)
  - **Segment**
  - **Extend**
  - **Commit**





Over the network

- **Sampling** … from where? How?

DA + S

DA: Dispersal of blob data to **custody**

- Ensuring **Data Availability**, while **Sharding**

S: Sampling from **custody**

- **Testing** Data Availability

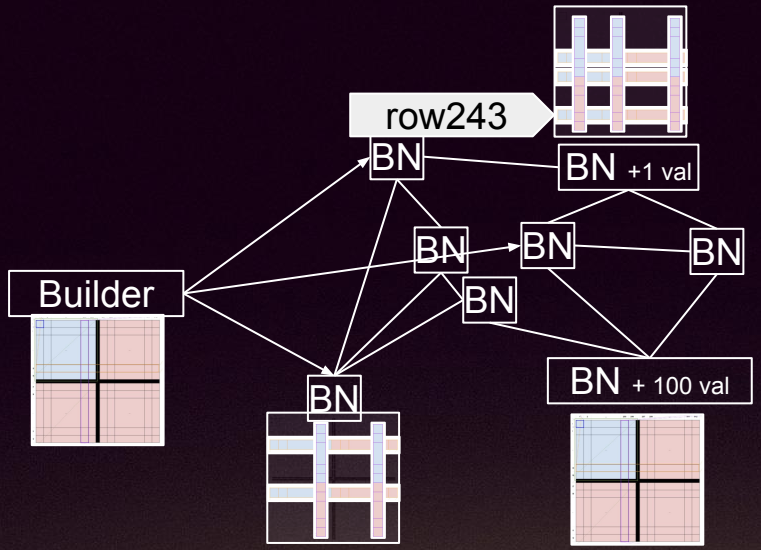
DA + S

DA: Dispersal of blob data to **custody**

- Ensuring **Data Availability**, while **Sharding**

S: Sampling from **custody**

- **Testing** Data Availability



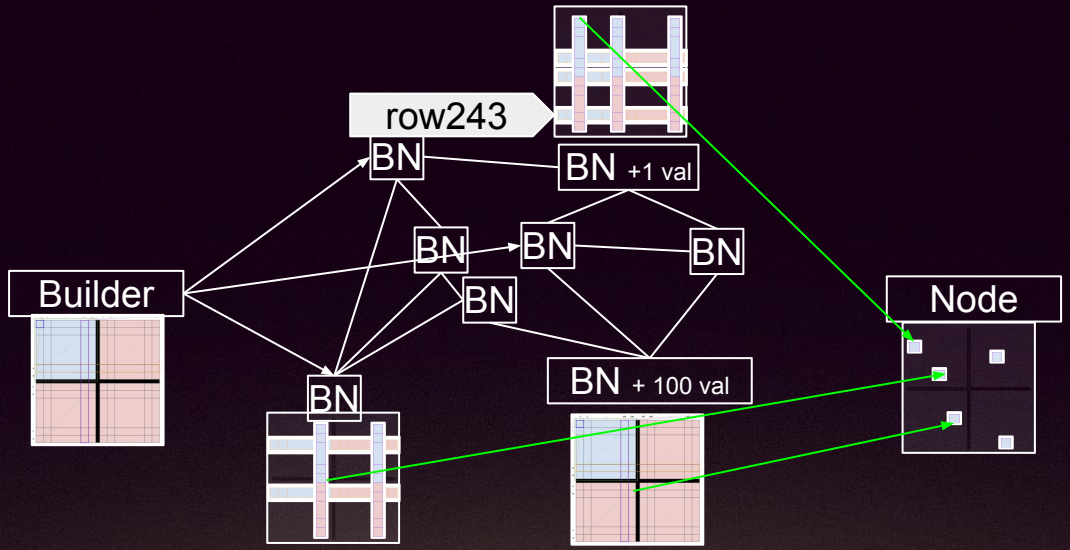
DA + S

DA: Dispersal of blob data to **custody**

- Ensuring **Data Availability**, while **Sharding**

S: Sampling from **custody**

- **Testing** Data Availability



DA + S

DA: **Dispersal** of blob data to **custody**

- Ensuring **Data Availability**, while **Sharding**

Robust

Secure

Fast

Cheap
Low traffic overhead

S: **Sampling** from **custody**

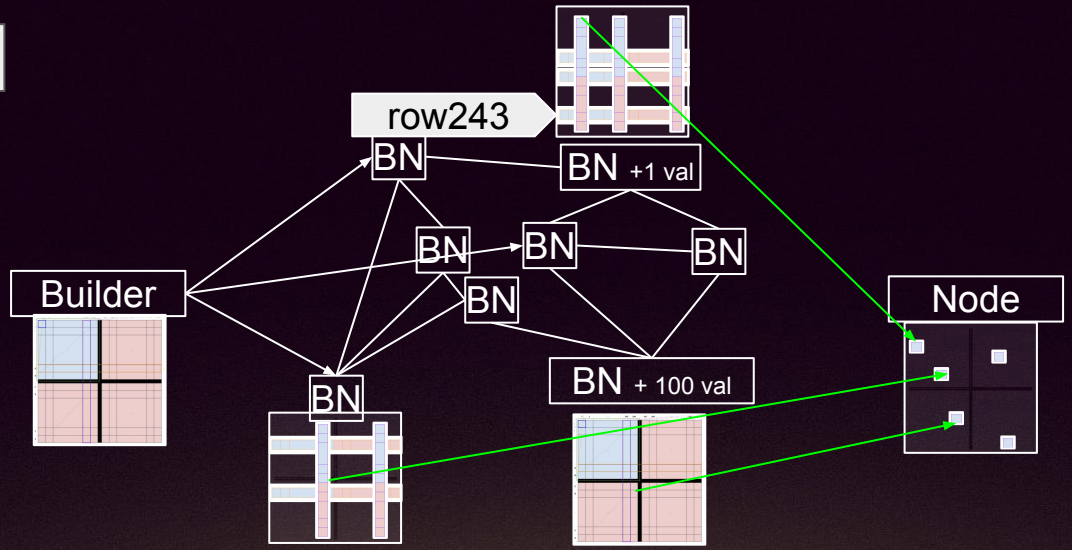
- **Testing** Data Availability

Secure

Robust

Fast

Cheap
Low traffic



PeerDAS to FullDAS: the 32MB challenge

LossyDAS, IncrementalDAS, and DiDAS to absorb false negatives

2D encoding with on-the-fly **Availability Amplification**

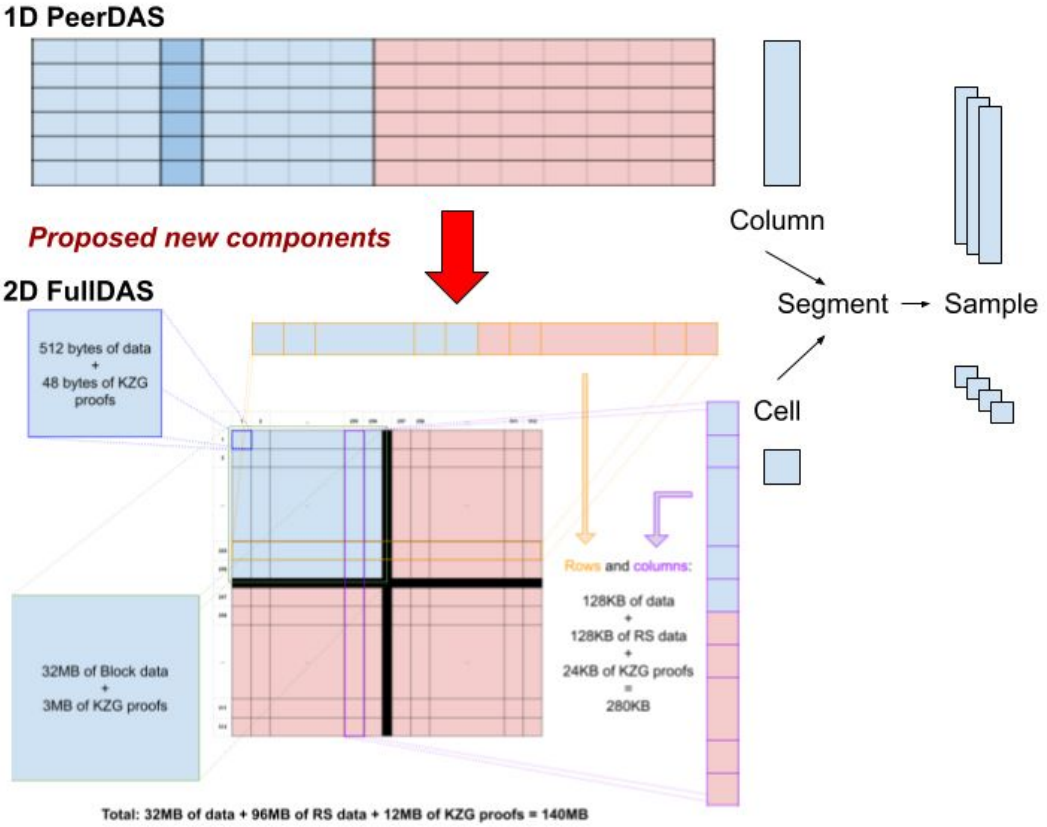
Enhanced and efficient discovery with **Topic Routing**

Ephemeral Connect for fast sampling

liteDAS for fast, robust, bandwidth efficient and reliable sampling

Rarest-first and non-TCP transport for fast segment dispersal

Push-pull state transition and IDONTWANT to reduce bandwidth



So what's the difference?

PeerDAS

FullDAS

So what's the difference?

PeerDAS	FullDAS
Cheap	Cheaper
Robust	More Robust
Fast	Faster
Secure	More Secure

So what's the difference?

PeerDAS

Cheap

Robust

Fast

Secure

FullDAS

Cheaper

Sampling at **cell level**, smaller traffic volume

More Robust

2D erasure code allows local repair, no need for large nodes

Faster

New pub-sub and sampling protocols make it faster

More Secure

Rethinking the data flow

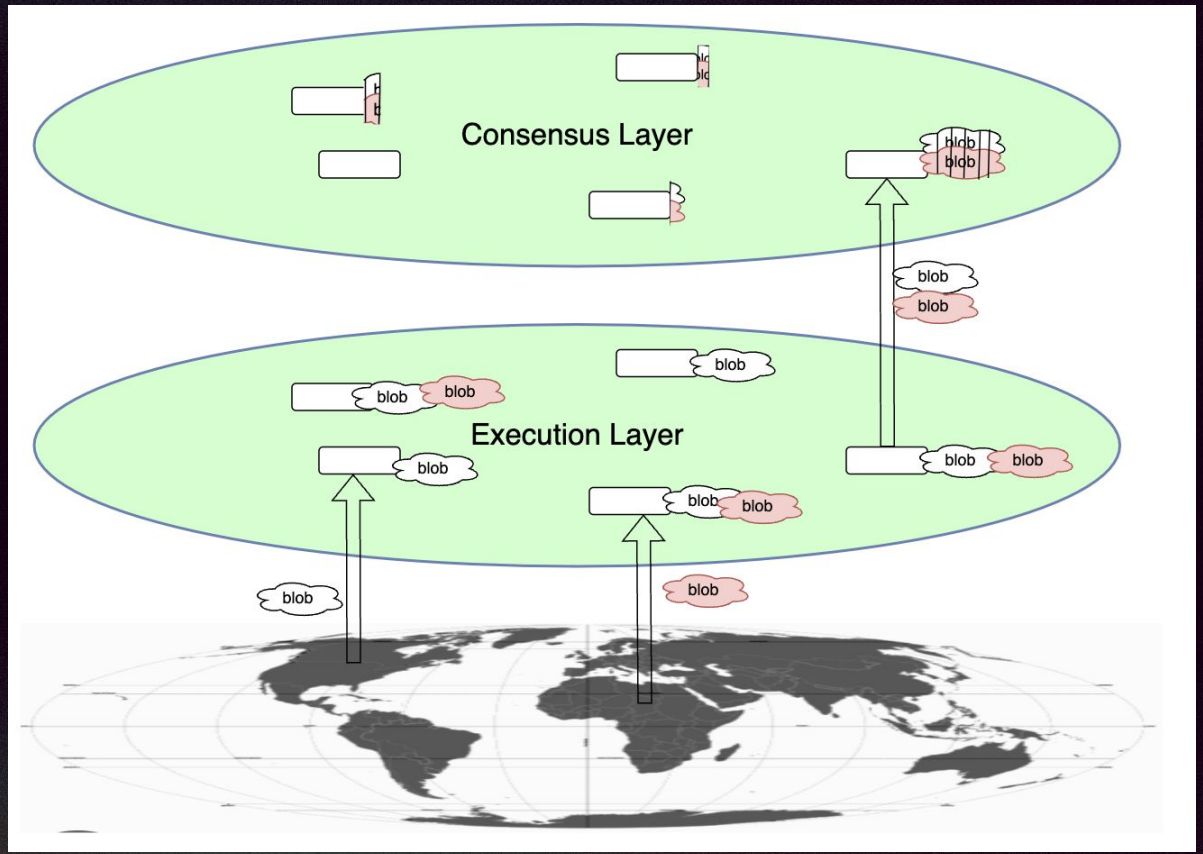
5, Dispersing blob pieces
(libp2p, push/pull)

4, Building the block

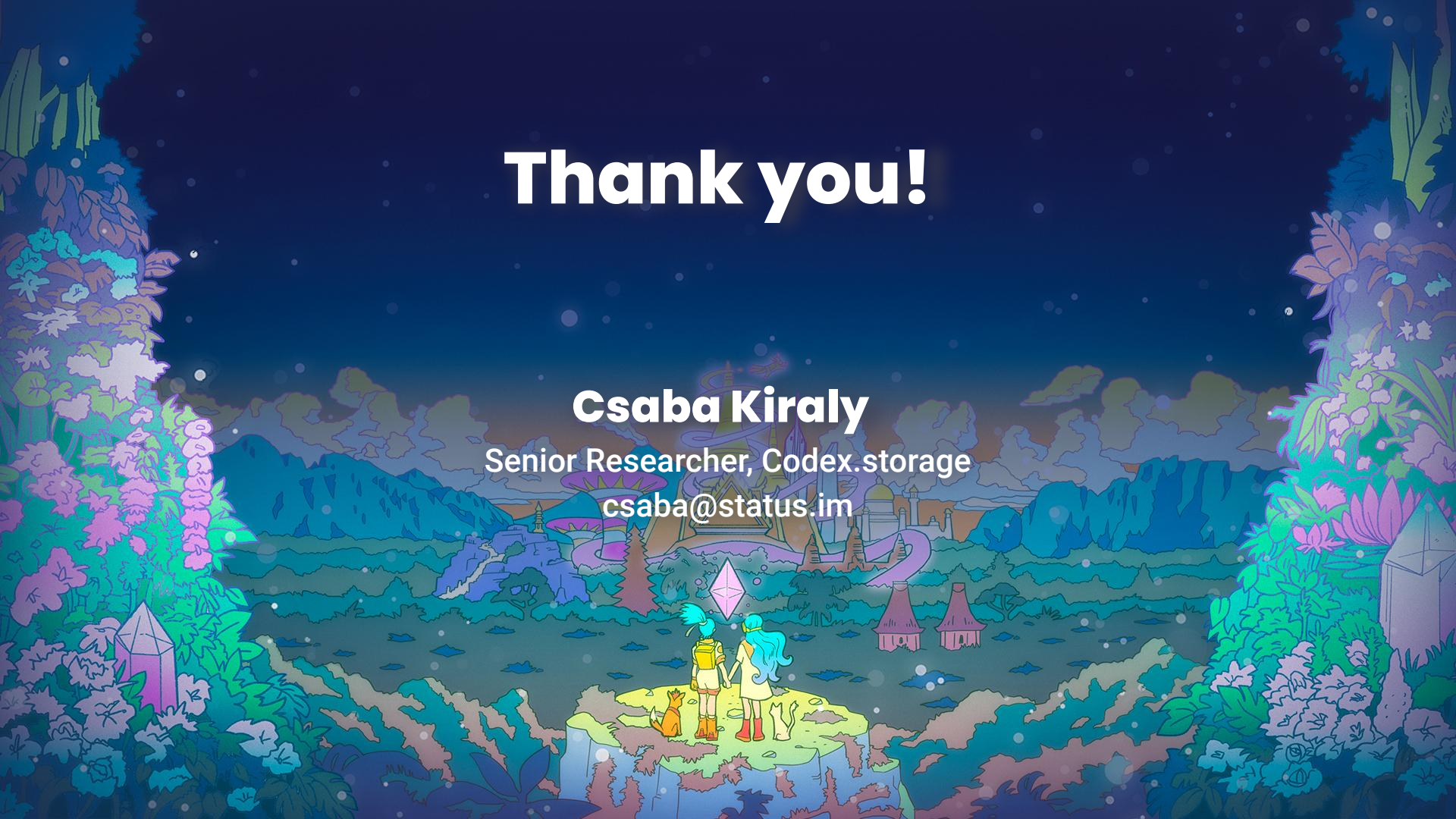
3, Mempool exchanges
(devp2p, pull)

2, A blob tx is sent

1, A blob tx is sent



Thank you!



Csaba Kiraly
Senior Researcher, Codex.storage
csaba@status.im