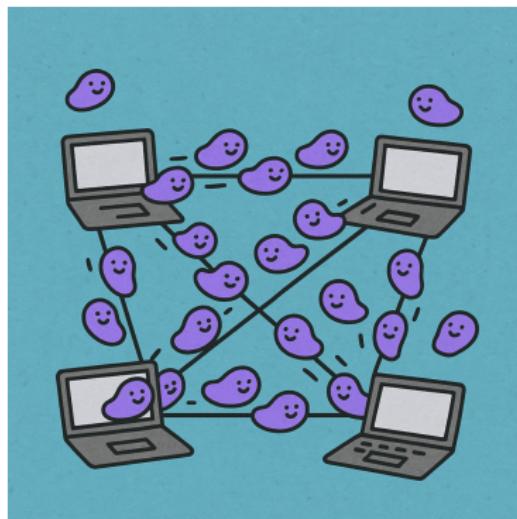


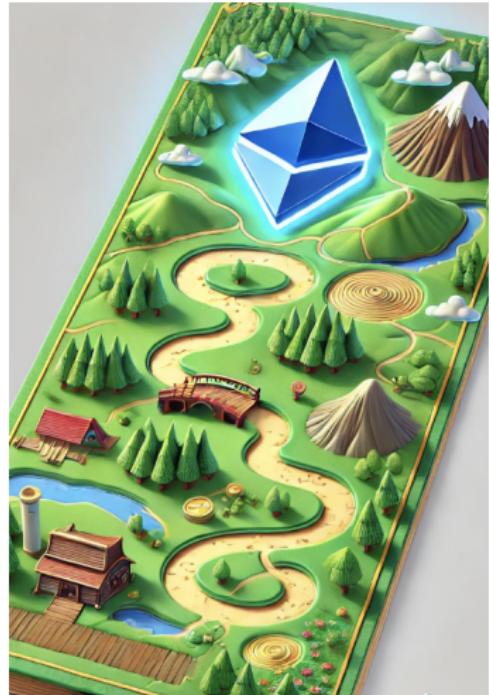
Blobs and the future of the blob mempool



mike neuder *with Julian Ma & Francesco D'Amato*
Monday, June 30th, 2025
ETHCC[8] – Cannes, France

Agenda

- Revisiting blob services
- Blob gossip & validation
(present)
- What changes with PeerDAS?
(short-term future)
- What can we do moving forward?
(long-term future)



Revisiting blob services

Confirmation rules

Definition (Confirmation rule (informal))

A signal indicating the status of a transaction.

0x9dc54a8afc2d85d96446c67cbcc29a7f69fc77f57d90ca1ce122ddb430c12272 

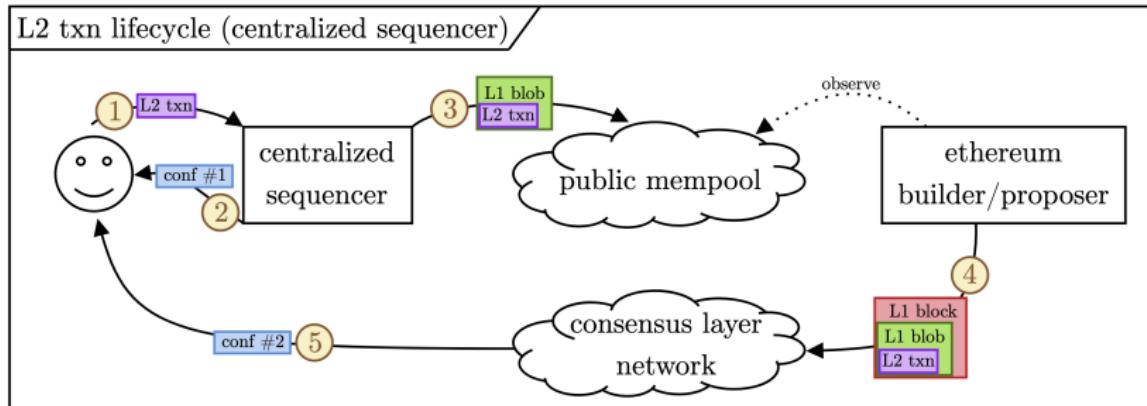
 Success

28499836 Confirmed by Sequencer

- *Blobs provide confirmation rules for L2 transactions.*

Revisiting blob services

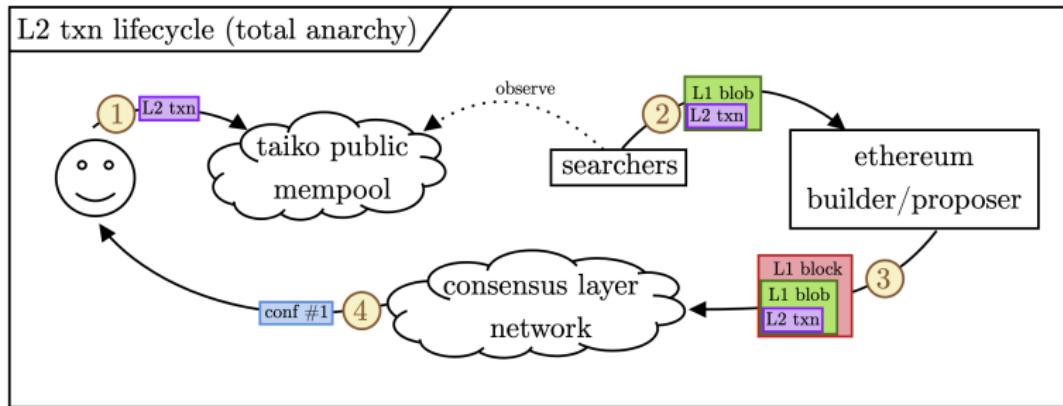
Centralized sequencers \implies patient blobs



- ▶ **tl;dr;** The first confirmation is sufficient for most users, thus there is no urgency to land the blob on the L1.

Revisiting blob services

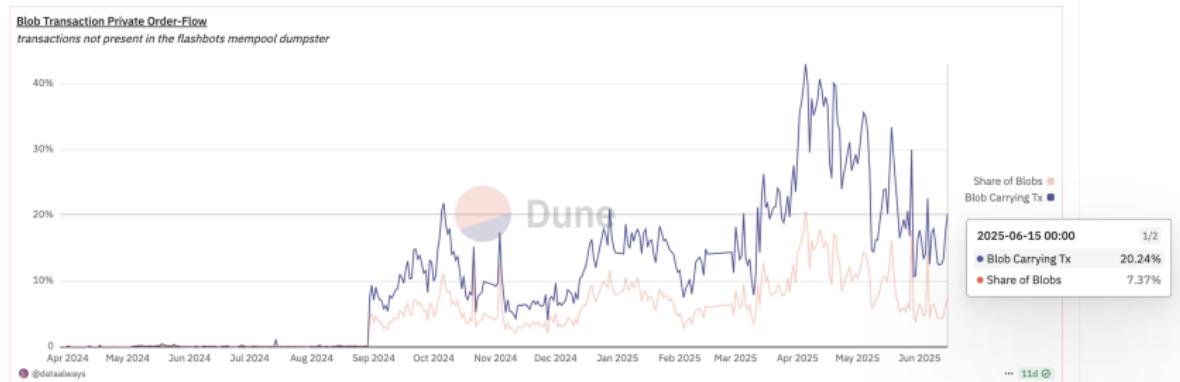
Total anarchy \implies impatient blobs



- ▶ **tl;dr;** The first confirmation is the blob inclusion on the L1, thus there is urgency (read: MEV) to land the blob.

Revisiting blob services

Summary



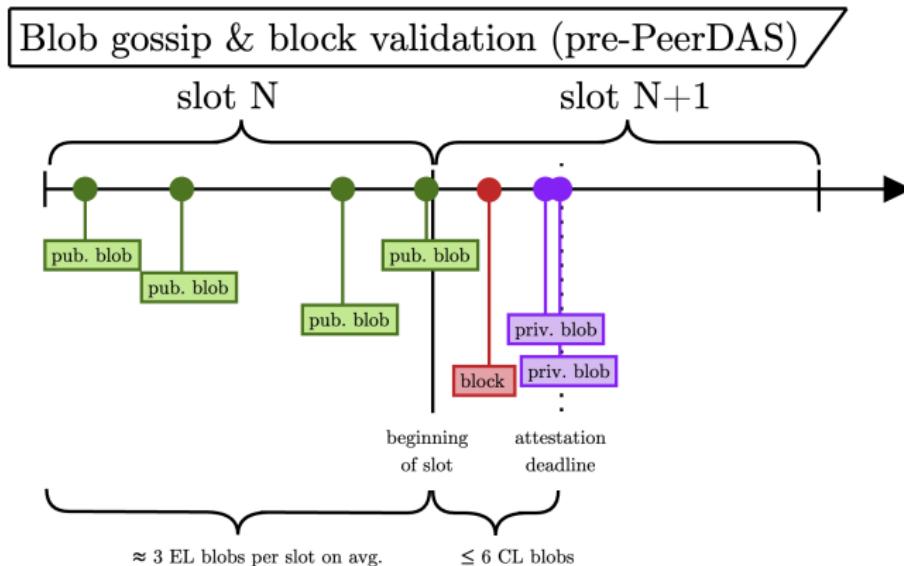
- ▶ We expect patient blobs to be content using the public mempool as long as it provides a relatively good inclusion service!
- ▶ **Foreshadow:** blobs distributed through the public mempool are probably better for builders and low-resourced attesters.

Blob gossip & validation (present)

- ▶ Ethereum nodes run execution and consensus clients. Thus, they connect to different P2P networks.
- ▶ The blob mempool (abbr. blobpool) is an execution layer object that maintains the set of pending, but not included, blob transactions.
- ▶ The block validation and attesting flow is a consensus layer operation that checks for the availability of blob data.

Blob gossip & validation (present)

Public & private blob flows

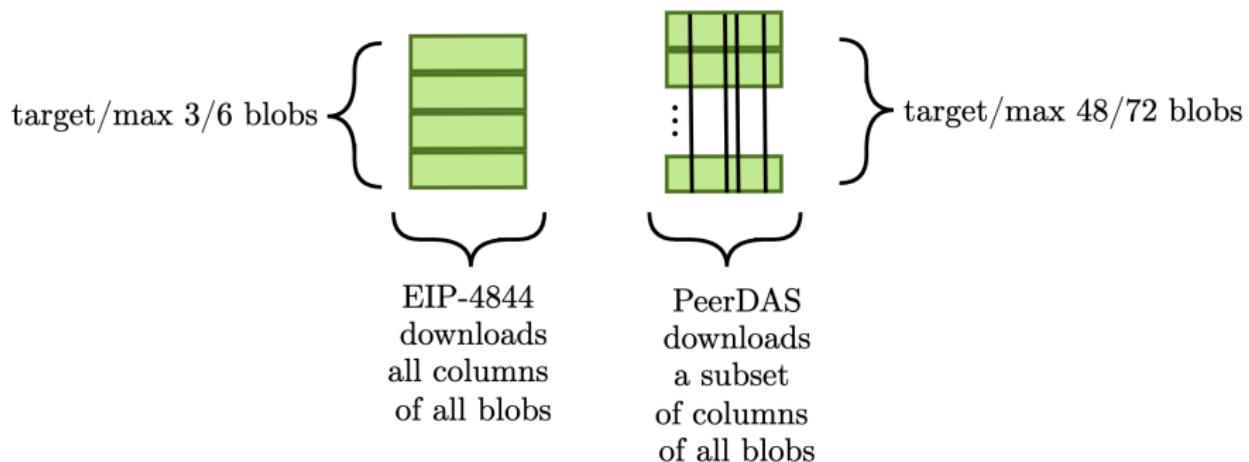


- ▶ Blobpool blobs *can be used for block validation.*
- ▶ There is a fundamental symmetry between execution and consensus, because the DA check needs the full blobs.

What changes with PeerDAS? (short-term future)

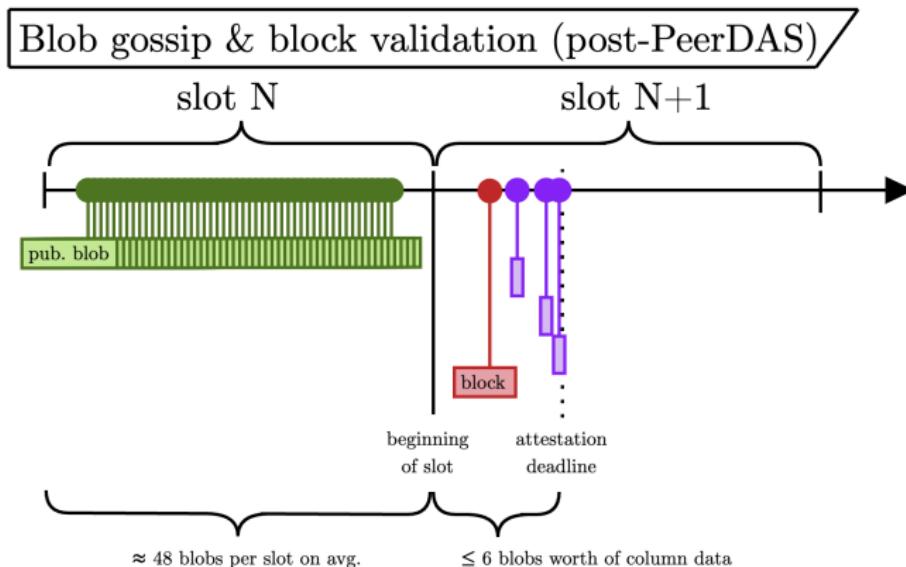
PeerDAS reminder

- ▶ Slated to be the headliner of the Fusaka hardfork – October 2025?
- ▶ Allows massive scaling of blob count by reducing the DA check to a subset of the columns of each blob.



What changes with PeerDAS? (short-term future)

New block validation



- ▶ **Key observation:** The *types* of the blobpool and the DA check are now different (full blob vs. column).
- ▶ Without changing anything, the blobpool bandwidth increases by 8x, while the DA check data remains constant.

What can we do moving forward? (long-term future)

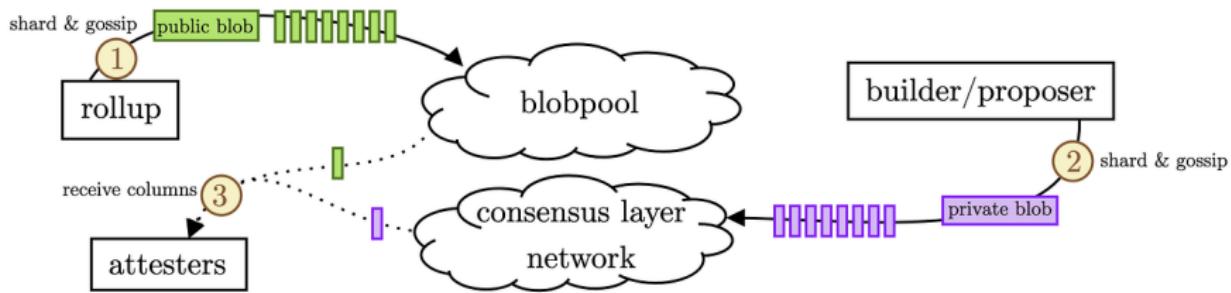
Do we actually want to do anything?

- ▶ **Fact #1:** No protocol-enshrined changes will be implemented for the blobpool in Fusaka.
- ▶ **Fact #2:** The number of validators self-building is so low that the blob inclusion service might not significantly degrade.
- ▶ **Fact #3:** Shifting to a predominately “private-blob regime” is not inevitable because public blobs are better for builders.
- ▶ **Claim #1:** It is worth keeping the public blobpool around as the default, neutral path for blobs.
- ▶ **Claim #2:** Solving the fundamental asymmetry of blob validation and gossip is important to continue scaling – especially for attesters.

What can we do moving forward? (long-term future)

Strawman proposal: gossip columns

Blob validation w/ vertical-sharded blobpool



- ▶ *Can we just gossip columns directly into the blobpool?*
- **No**, because this is a free DoS vector on the blobpool.
- ▶ We need to ensure that column data gossiped in the mempool pays for that service.

What can we do moving forward? (long-term future)

Candidate mechanism: Blobpool ticket auction with refunds

- ▶ **Core idea:** auction off write access to the blob mempool!
 1. Slot $n-2$: Bids are submitted for the Slot n access to the mempool.
 2. Slot $n-1$: The bids are processed onchain, and the write access is allocated accordingly.
 3. Slot n : Blob columns are gossiped, but must be accompanied with a winning bid.
 4. Slot $n+1$: Any blobs that land onchain with a winning bid are refunded.
- ▶ **Note:** we want write access to the blobpool to be free in expectation to avoid pushing L2s into sending private blobs.
- ▶ There are many considerations in this design space.

What can we do moving forward? (long-term future)

Broader considerations

- ▶ Horizontal sharding is much easier and may be worth the reduced complexity, but they fundamentally change the mempool assumptions.
- ▶ There are alternative ways to allocate blobpool write access in **Sybil-resistant** way (e.g., reputation systems).
- ▶ We could, alternatively, consider a wholesale **redesign of the blob fee mechanism** (e.g., blob inclusion enforcement with blobpool tickets to replace the priority fee).

What can we do moving forward? (long-term future)

Stakeholder questions

► *Questions for rollup teams:*

1. How do you think about public vs. private blobs?
2. How much latency are you OK with for landing blobs?
3. Would you participate in the blobpool auction?

► *Questions for builders:*

1. How much do you depend on the public blobpool when deciding which blobs to include in blocks?
2. What inclusion guarantees do you offer to private blobs? How much do these blobs impact the outcome of the MEV auction?

► *Questions for validators/staking operators:*

1. How do you plan on handling the bandwidth increase resulting from an increase in the number of blobs under PeerDAS?
2. Would you plan on continuing to download all blobs in either the horizontal or vertical sharding world?

thanks :)

questions?



¹<https://hackmd.io/@mikeneuder/blob-gossip-and-validation>

²<https://ethresear.ch/t/on-the-future-of-the-blob-mempool/22613>