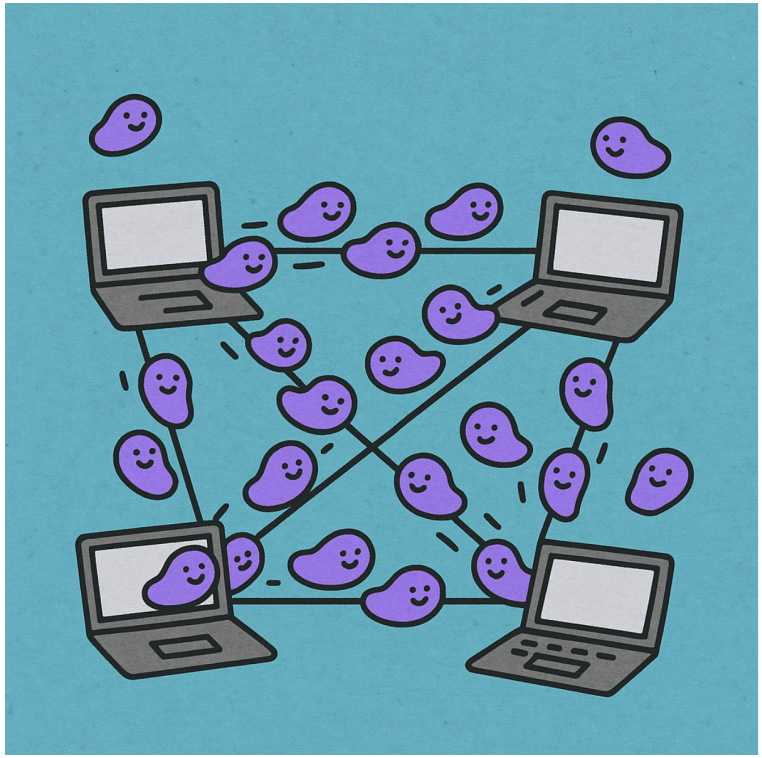
Blob futures and the blob mempool



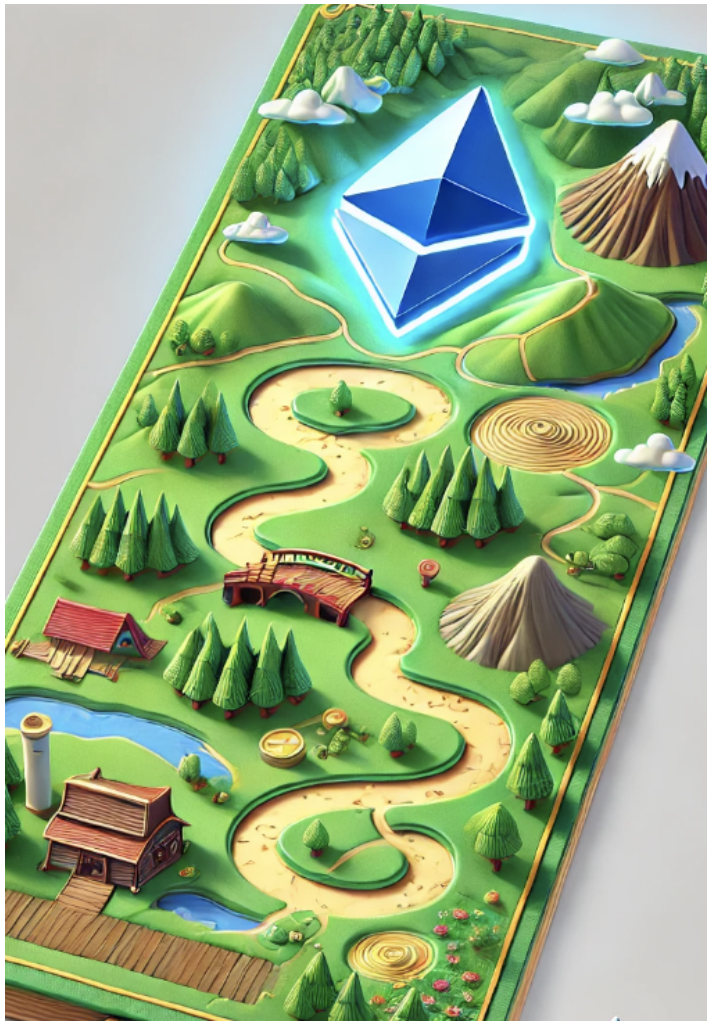
mike neuder *with Julian Ma & Francesco D'Amato*

Monday, June 30^{th}, 2025

Blockspace Futures Day – Cannes, France

Agenda

1. Blob futures as a scaling technology
   - Revisiting blob services
   - Blob gossip & validation
   - What changes with PeerDAS?
2. Blob futures use cases
   - Blob futures use case #1: *trustless L2→L1 bridging*
   - Blob futures use case #2: *preconf rollup interop*
   - Differences with blockspace futures

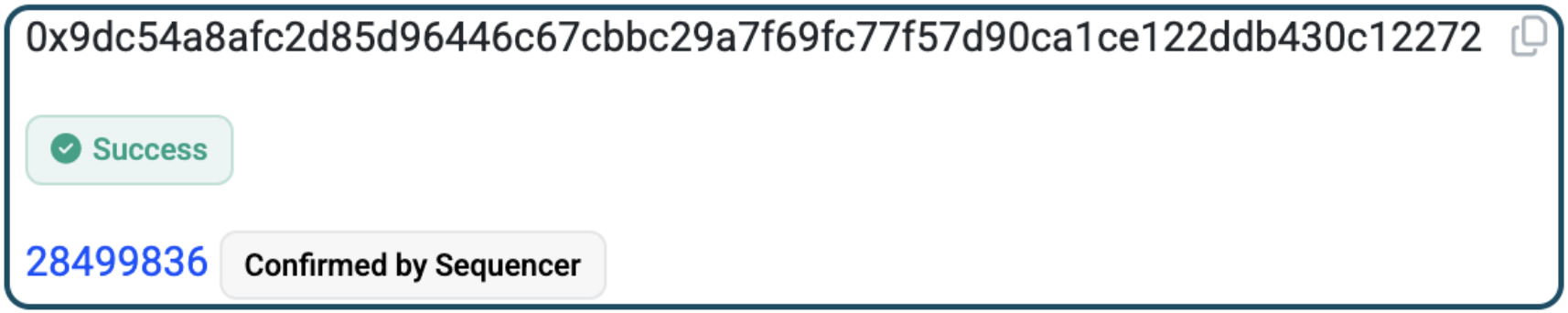


Revisiting blob services

Confirmation rules

Definition (Confirmation rule (informal))

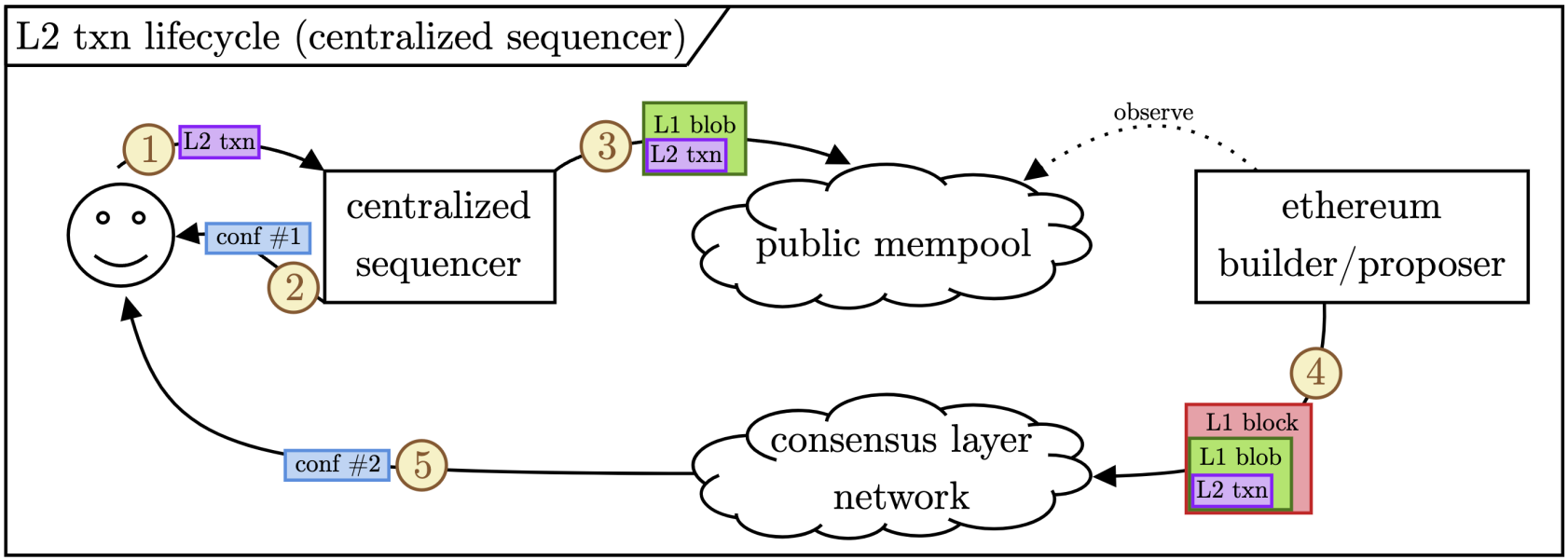
A signal indicating the status of a transaction.



- *Blobs provide confirmation rules for L2 transactions.*

Revisiting blob services

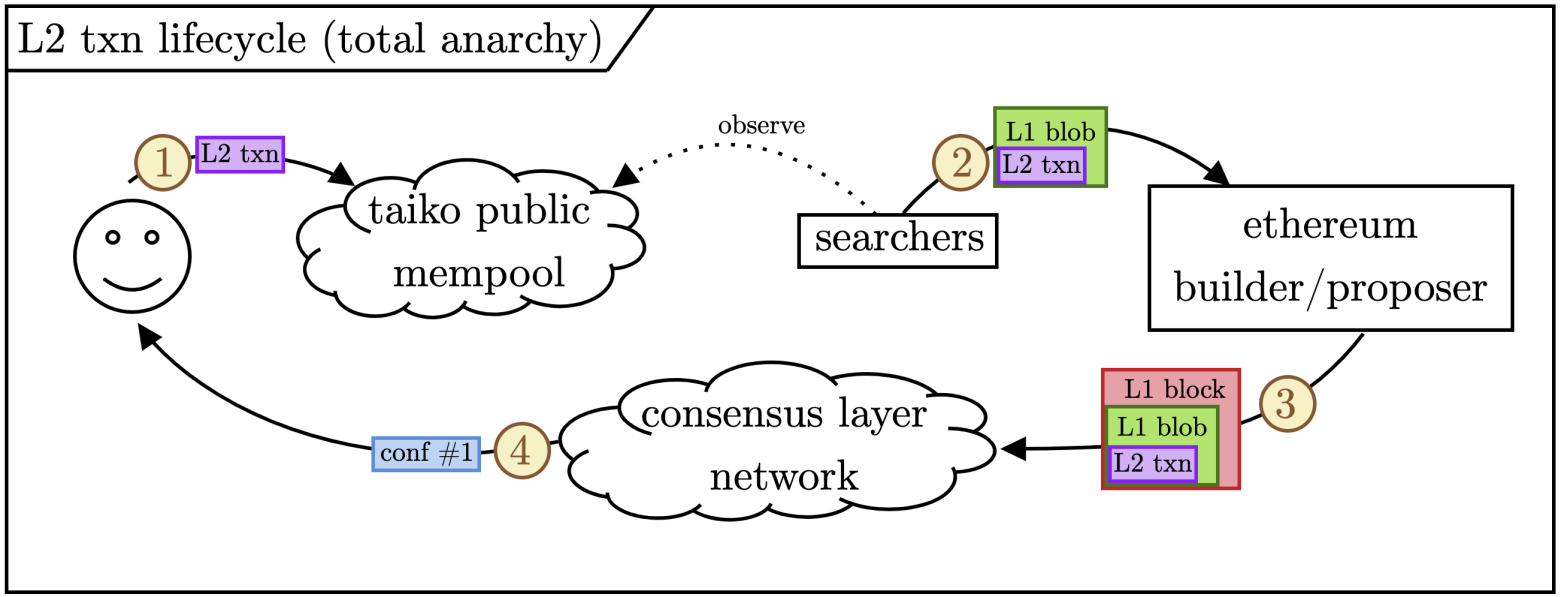
Centralized sequencers $\Longrightarrow$ patient blobs



- **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

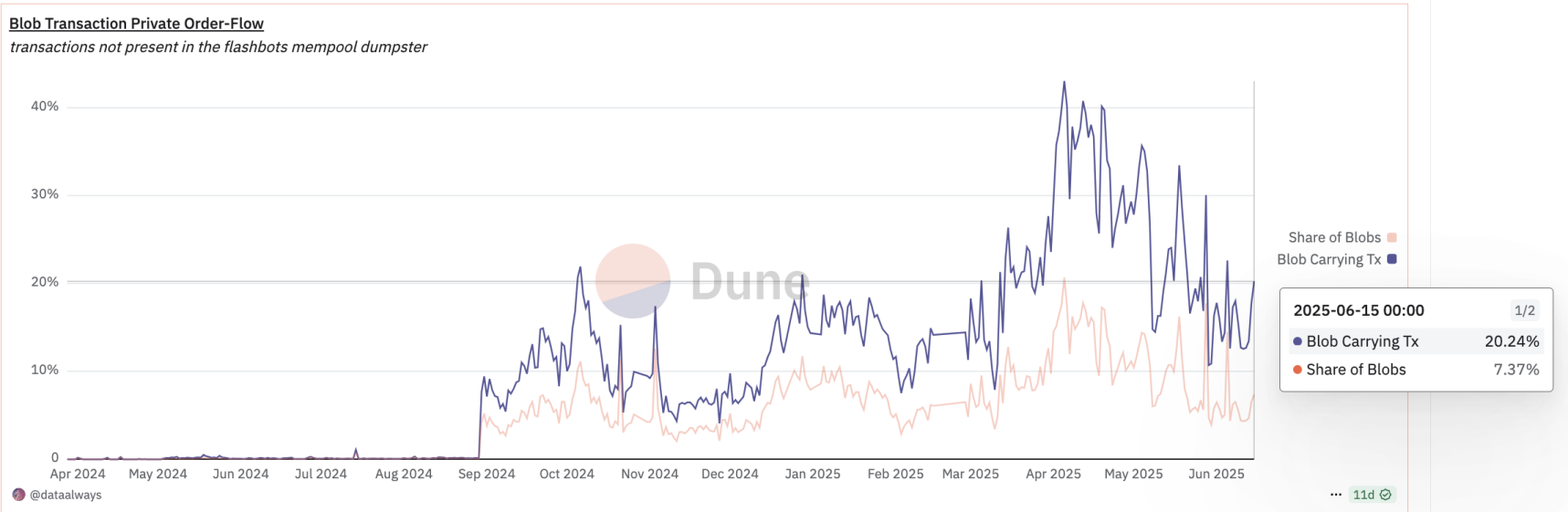
Total anarchy $\Longrightarrow$ impatient blobs



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

Revisiting blob services

Summary



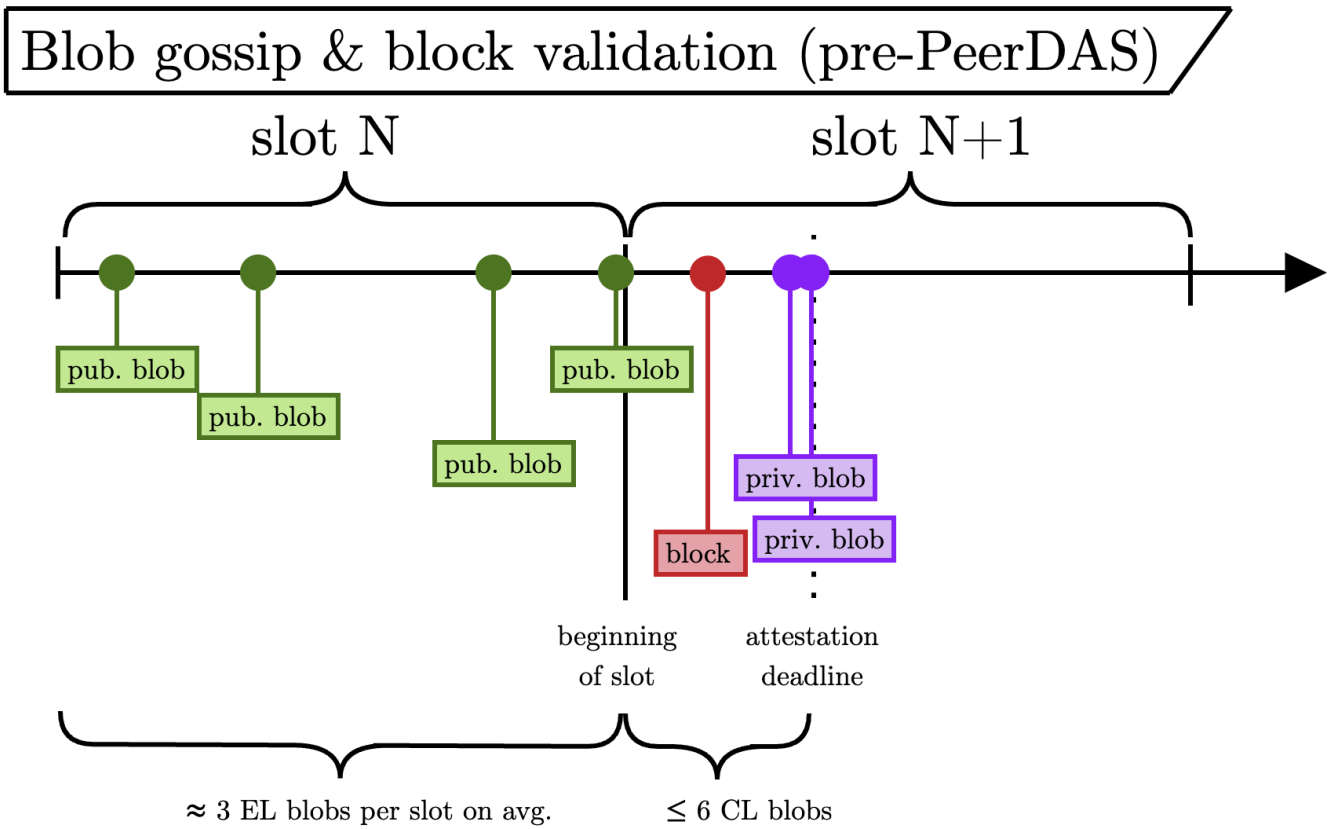
- We expect patient blobs to be content using the public mempool as long as it provides a relatively good inclusion service!
- **Foreshadow:** blobs distributed through the public mempool are probably better for builders and low-resourced attesters.

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

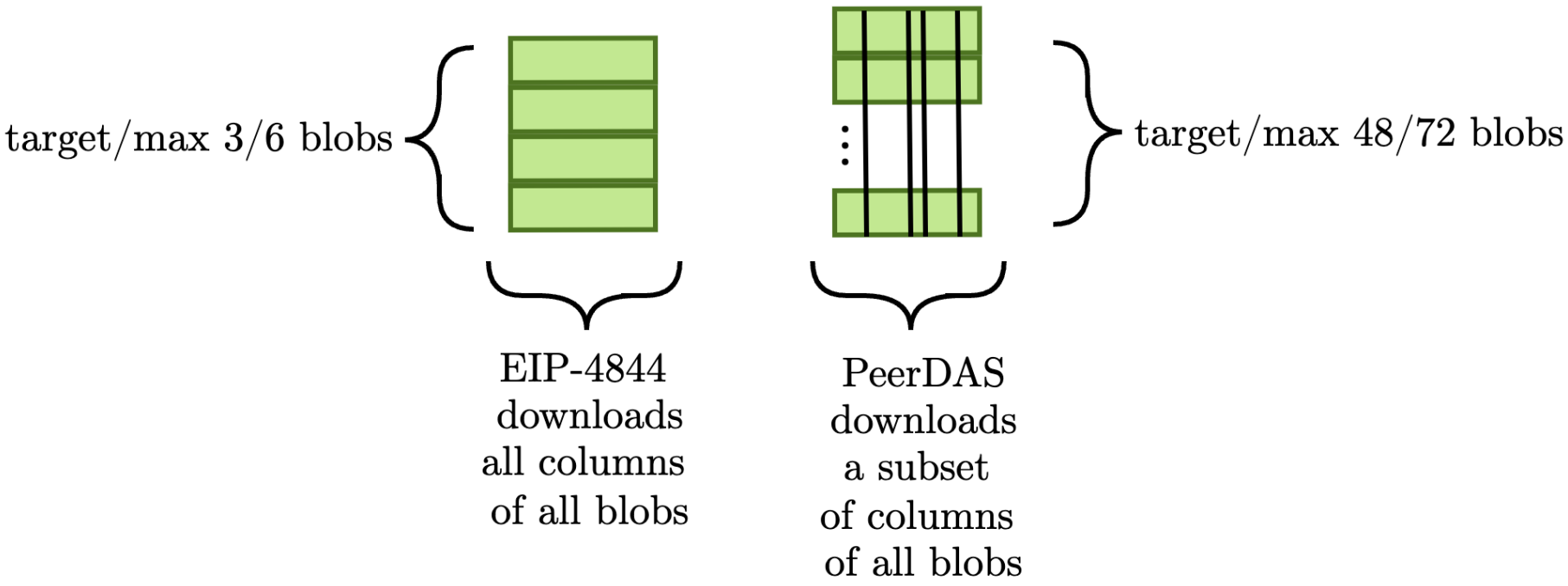


- Blobpool blobs *can be used for block validation*.
- There is a fundamental symmetry between execution and consensus, because the DA check needs the full blobs.

What changes with PeerDAS?

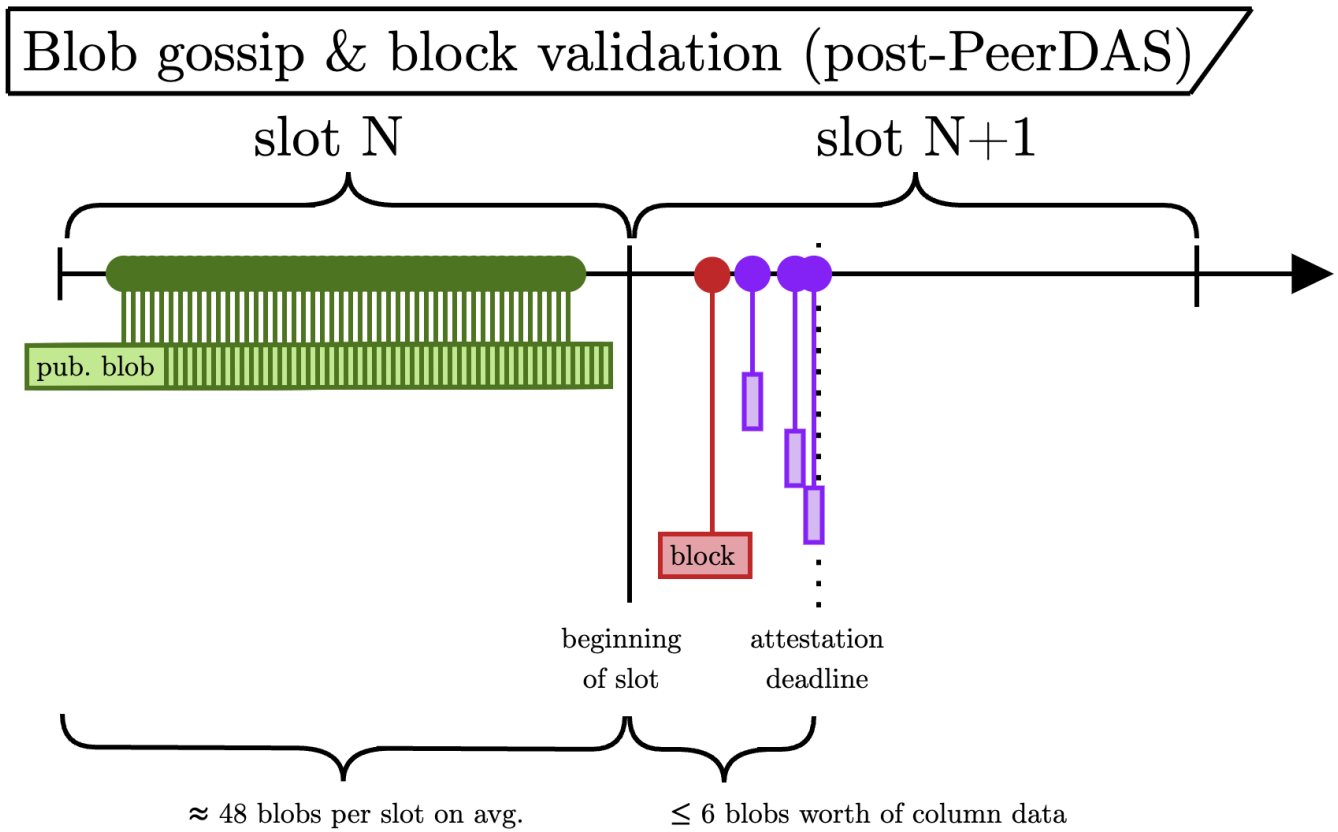
PeerDAS reminder

- 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?

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.

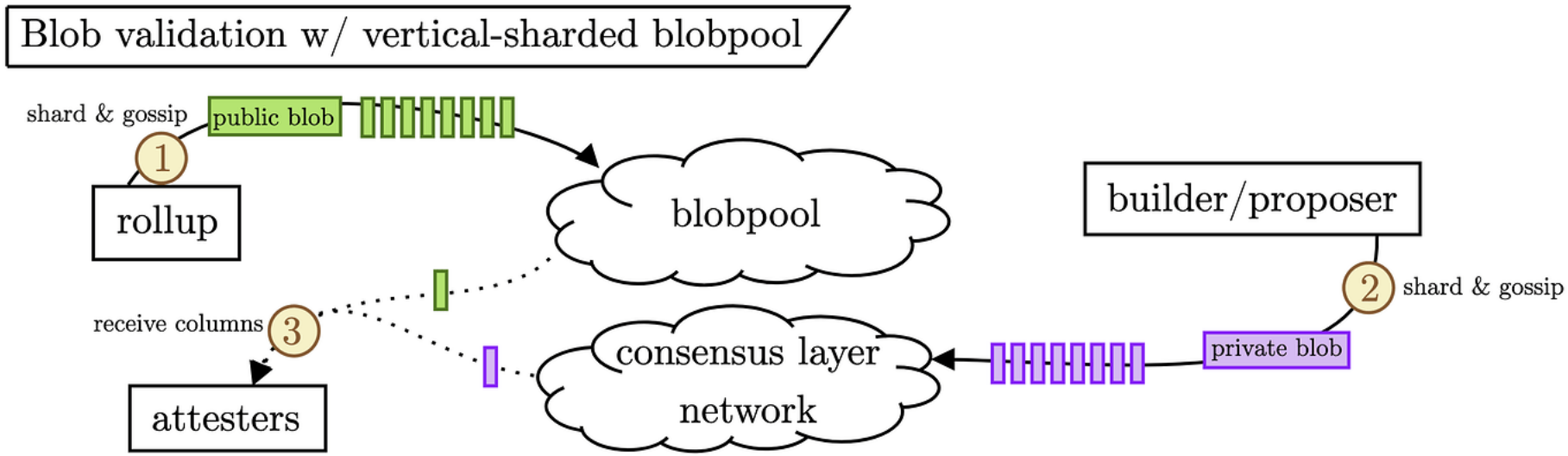
What changes with PeerDAS?

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 changes with PeerDAS?

Strawman proposal: gossip columns



- *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.

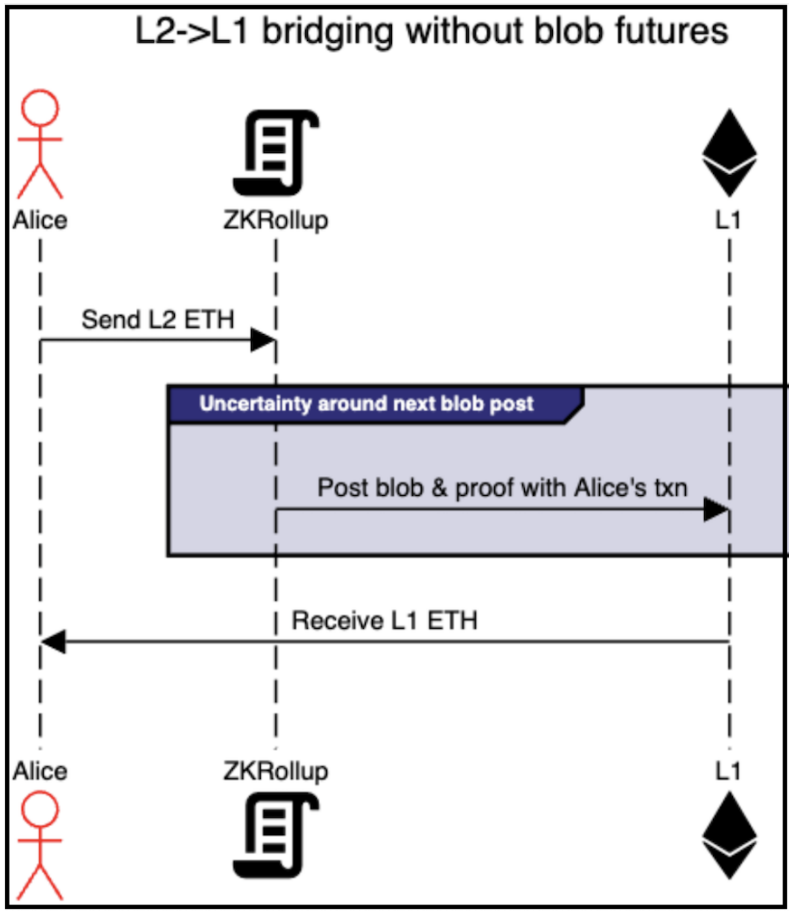
What changes with PeerDAS?

Candidate mechanism: Blob futures auction

- **Core idea:** auction off blobspace in advance. Only allow gossip of pre-allocated blobs.
  1. `Slot n-2`: Bids are submitted for the `Slot n` blobs.
  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.
- **Note:** this is a redesign of the blob inclusion pipeline. It is possible that only a subset of blobs are sold ex-ante.
- The deisgn space here is large.

Blob futures use case #1: trustless L2→L1 bridging

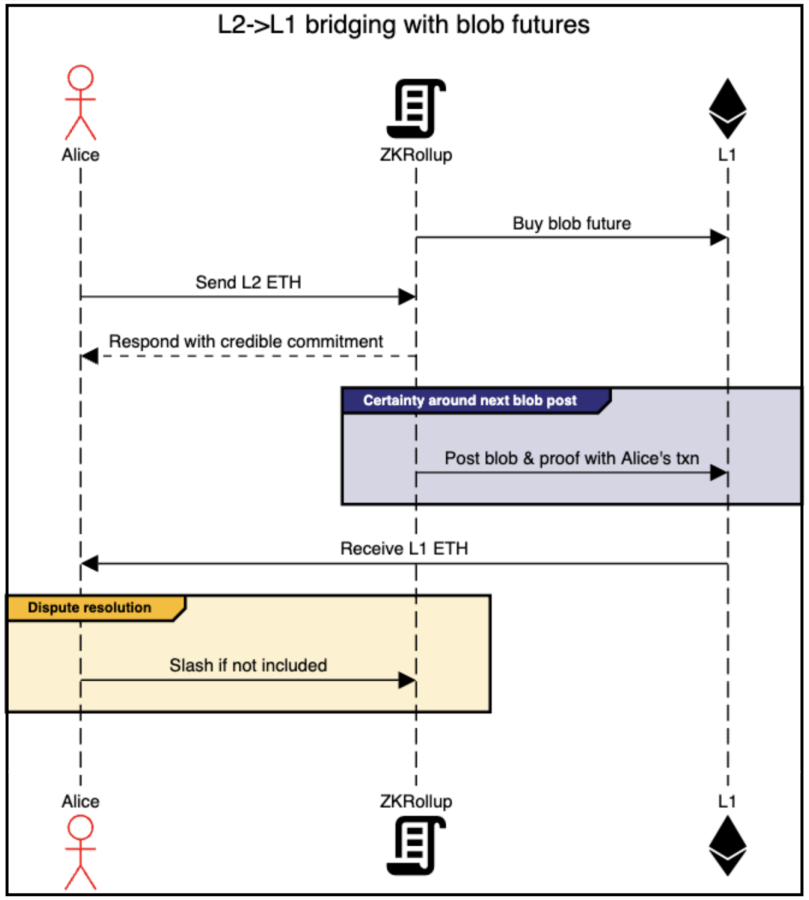
Without blob futures



- **Key point #1:** The rollup cannot guarantee timely blob posting.
- **Key point #2:** The user can't hold the rollup accountable.

Blob futures use case #1: trustless L2→L1 bridging

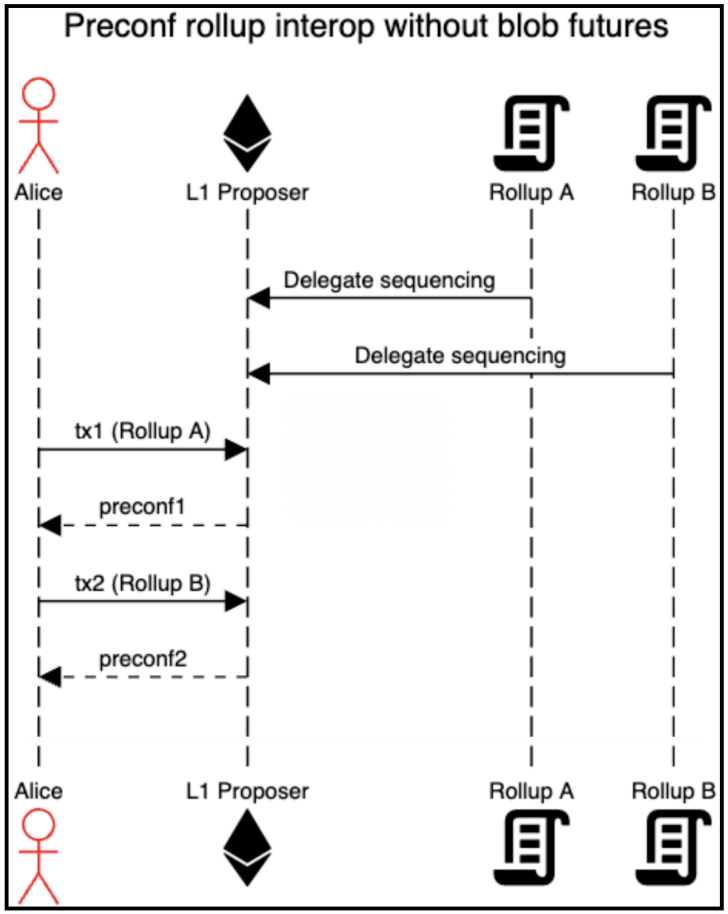
With blob futures



- **Key difference #1:** The rollup *can* guarantee timely blob posting.
- **Key difference #2:** The user *can* hold the rollup accountable.

Blob futures use case #2: preconf rollup interop

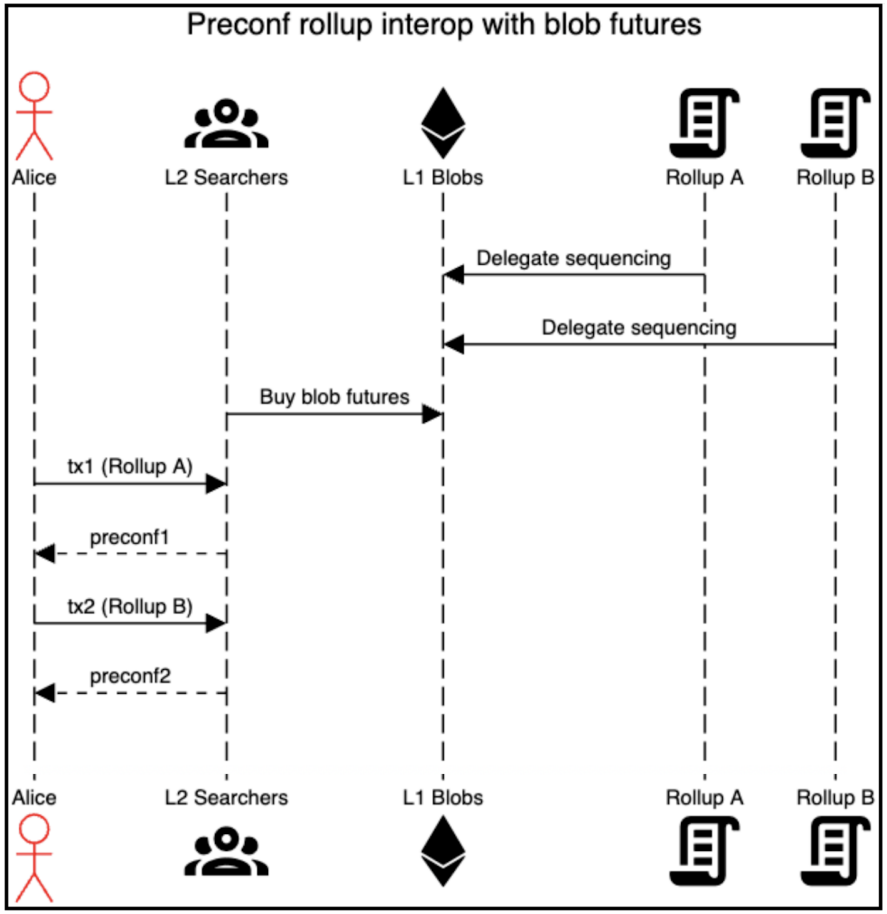
Without blob futures



- **Key point #1:** All based rollups are sequenced by the L1 proposer.
- **Key point #2:** Based sequencing competes with L1 execution.

Blob futures use case #2: preconf rollup interop

With blob futures



- **Key difference #1:** L2 searchers *compete* to sequence based rollups.
- **Key difference #2:** Based sequencing *doesn't* compete with L1 execution.

Block futures vs blob futures

- **Protocol-based block futures**
  1. Protocol could sell the right to build a block ahead of time (e.g., using an auction or a ticketing mechanism).
  2. **Goal:** Reduce impact of MEV on staking rewards.
  3. **Issue:** Multi-slot MEV.
  4. **Blessing & curse:** Removing the value-in-flight auction.
- **Protocol-based blob futures**
  1. *Could* prevent based rollup MEV from impacting validator economics.
  2. *Could* lead to multi-slot MEV for based rollups.

tl;dr;

- PeerDAS introduces a fundamental asymmetry between how blobs are gossiped and validated.
- Blob futures are a market-based mechanism to eliminate this asymmetry.
- The set of features that blob futures offer rollups is an important design consideration.

thanks :)

questions?

[1] https://hackmd.io/@mikeneuder/blob-gossip-and-validation

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