Hi All,

I have a question regarding the approach Ethereum has regarding data availability vs data retrievability, specifically for Layer 2 networks. As a reminder the following are the definitions as per the Ethereum Docs:

· Data availability is the assurance that full nodes have been able to access and verify the full set of transactions associated with a specific block. It does not necessarily follow that the data is accessible forever.

· Data retrievability is the ability of nodes to retrieve historical information from the blockchain. This historical data is not needed for verifying new blocks, it is only required for syncing full nodes from the genesis block or serving specific historical requests.

As I understand it, currently Layer 2 solutions use the CALLDATA function to permanently store their batched compressed data with Optimistic rollups submitting compressed transaction data for peer review via the challenge period, and ZKPs submitting their proof which contains the mathematical validity of their state. The result of this is that state computation is moved off mainnet and fees are shared between several users within the batch, making each user's transaction significantly cheaper. However, what is published to mainnet is the state change and not any transaction metadata, i.e., who interacted with who. This transaction metadata is the responsibility of the Layer 2 to store and manage, as per the Data Retrievability ethos of Ethereum above.

Currently this state change data persists within CALLDATA but will move to blob storage as per EIP-4844, which will not affect the data availability of Ethereum as this data is only required temporarily, specifically the amount of time it takes independent verifiers to validate the state change. Once this period has passed the state change itself will only exist on the EVM and supporting information will only be held on the Layer 2.

My observation surrounding this is that Ethereum can be used to serve as a robust base layer to validate and hold the state of several Layer 2s, as the Layer 2s inherit the decentralisation and security of Ethereum as they do not have consensus methods themselves. However, it does mean that historical data and metadata surrounding transactions is the responsibility of the Layer 2 provider. Does this not create a problematic situation though? While the state of the Layer 2 can be validated, the log of transactions and interactions between wallets is lost, or is at least held in a non-standard centralised database, being the Layer 2 storage technique.

My analysis of this issue is that it gives the Layer 2 providers significant power which platform users may not be aware of. For example, in the situation a bad actor traverses a Layer 2 using illicit funds, and in the event the Layer 2 does not expose their block explorer, which is an abstraction of their centralised database containing transaction metadata, tracking of the illicit funds becomes significantly more difficult. This is because the metadata in the transaction is lost and not published on mainnet, only in a state change within a batch of other transactions. Does this then not mimic a Tornado Cash 2.0 situation as the deposit and withdrawal to/from the Layer 2 being public, but any transfer or interaction on Layer 2 being opaque, therefore lending itself to migration of illicit funds from a dirty wallet to a clean wallet on Layer 2.

In addition, the effect at an enterprise level is more substantial as each Layer 2 is responsible for their own transaction metadata, and due to non-compliance of critical infrastructure or lack of supporting controls, relevant data may be deleted. This is also likely to occur as the Layer 2 network ecosystem will consist of many sub-networks all with their own supporting Web2 stacks, increasing the risk. This deleted data makes the entire Layer 2 and any other enterprise entities using the network being left with a significant issue, the inability to perform a complete audit as the supporting evidence no longer exists, only the attestation that the Layer 2 is accurate and complete given mainnet holds the valid present state of the Layer 2, which is insufficient in an audit. This is because, in order to perform sampling or revenue testing more than attestation over the state is required, the metadata of each transaction will be required.

When EIP-4844 gets implemented (which I support) this enterprise issue becomes more problematic, as the chain of state change proofs becomes lost after the challenge window expires. This is the last audit log present, and my fear is that it will be damaging to the Ethereum ecosystem as it will not inherently support enterprise needs, being auditing. Furthermore, it also seems like a lost opportunity via the automation of smart contracts and the potential for automated on-chain audits. While I understand that the current proof being accurate results in past proofs being accurate, during an enterprise audit the auditors will also be interested in identifying and acknowledging any potential challenges or rollbacks.

I would be interested to hear the communities' thoughts regarding this and any fixes if they exist, which I have not identified yet. If no fix exists yet, my thinking is leaning towards a dedicated plasma sidechain to store the blob data indefinitely and/or transaction metadata of Layer 2s for the sole purpose of auditing. My hope is that this plasma side chain offers a consistent data storage mechanism for Layer 2s to use rather than their own centralised web2 stack which goes against the Ethereum ethos of decentralisation, robustness, traceability, and security, while also helping to support the enterprise development on Ethereum.