EDIT

: We have stared implementing this concept under the name Hashi at:GitHub - gnosis/hashi: An EVM header oracle aggregator

Preface

We see a range of approaches for bridges. The L2beat team gave a goodoverview here.

The problem with the current approach is that the bridge landscape is very heterogeneous. For some chain combinations, there might be bridge solutions with lower trust requirements (e.g. a zk based light client bridge between Gnosis Chain and Ethereum - or this Berkely paper on zk-birdges that has been the starting point for the zk-collective). However - those bridges might only be available for a subset of chains and an application that needs to bridge to several chains might need to use a bridge solution that has worse security assumptions but can serve a greater number of chains (e.g. committee-based bridges like e.g. wormhole)

(As a side note: Those security assumptions are about the system assuming it works as intended, they do not take into consideration the possibility of bugs. Historically many bridge hacks did not have their root cause in e.g. committees getting compromised but instead simply in smart contract bugs. Taking that into account it is maybe impossible to do an objective ranging of security of different bridges)

We see currently 2 issues:

1. As new bridge designs are being developed they often let applications pick new security tradeoffs but as discussed above they are rarely strictly better than what existed before
2. Standardization: currently, absolutely most bridges have their own message formats. For an application to switch from one bridge to another is a major engineering task. Early approaches to find standards for cross chain messages (kind of IBC for EVM) have not yet succeeded.

This approach tries to solve both:

1. Create “additive” security. New bridge approaches can ideally increase the security of existing bridges instead of offering new tradeoffs
2. Standardization at least on the lowest level (block header) - this allows standards to emerge above that are independent of the underlying bridge/trust mechanism

[

Screenshot 2023-01-31 at 16.49.32

2738×1542 383 KB

](https://ethresear.ch/uploads/default/original/2X/d/d2aa1320f3ac50c24dd6ade24e92ee5a12d5c493.png)

Block header based bridges

At the center of this approach are “header storage contracts” - simple contracts that store ChainID->blocknumber->header

These header storage contracts need to get the headers from “header oracles” - they can be the different bridge we know today. Note, it is easily possible to convert any existing bridge that allows sending arbitrary messages into header oracles. If they do not support access to block headers directly, one can simply write a contract on the source chain that accesses a recent block header and sends it into the bridge.

We believe the block header abstraction layer is the best for standardization and aggregation. As block header oracles need governance, there might still be many per chain combination, but the role of governance can be quite minimized. After setting up a set of trusted bridge oracles, a minimized governance would only act as a conflict resolution mechanism. If all oracles report the same header for a block number, governance has no rights. Only in case oracles make conflicting reports governance would need to resolve it.

Once a reliable source for block headers from a foreign chain is established different tools can emerge on top.

a) Merkle proof for specific storage slots

b) Merkle proof for specific events being emitted

c) zk-based proof of (storage/ events/ previous headers/ …)

On top of these primitives, application specific contracts can emerge: Token bridges/ NFT bridges/ allowing smart contract wallets to control assets cross chain/…

As a side note: It might be even useful to have such a header oracle for your own chain. Currently within the EVM applications can only access the last 256 headers.

Drawbacks

The main drawback of this approach is higher gas costs and potentially slower bridging times. If n oracles are required for the header storage contract, the bridge time will be determined by the slowest. However - “liquidity protocols” like Hop or Connexed have emerged that can do much faster optimistic execution. They are expanding their scope to general messages. Those approaches still need a source of truth eventually which the described design can hopefully provide.