Reviewing wPOKT Token Migration

tl;dr Let's go with the original proposal of using Hyperlane.

Purpose of this Doc

1. Unblock the wPOKT migration work from kicking off
2. Capture & summarize conversations & key points that temporarily blocked this work

Primary References

1. Raid Guild Proposal
2. wpokt-migration discord thread
3. Messaging protocol comparison

Context, Consideration & Concerns

1. Building & Maintaining Custom Software

Morse (v0) context

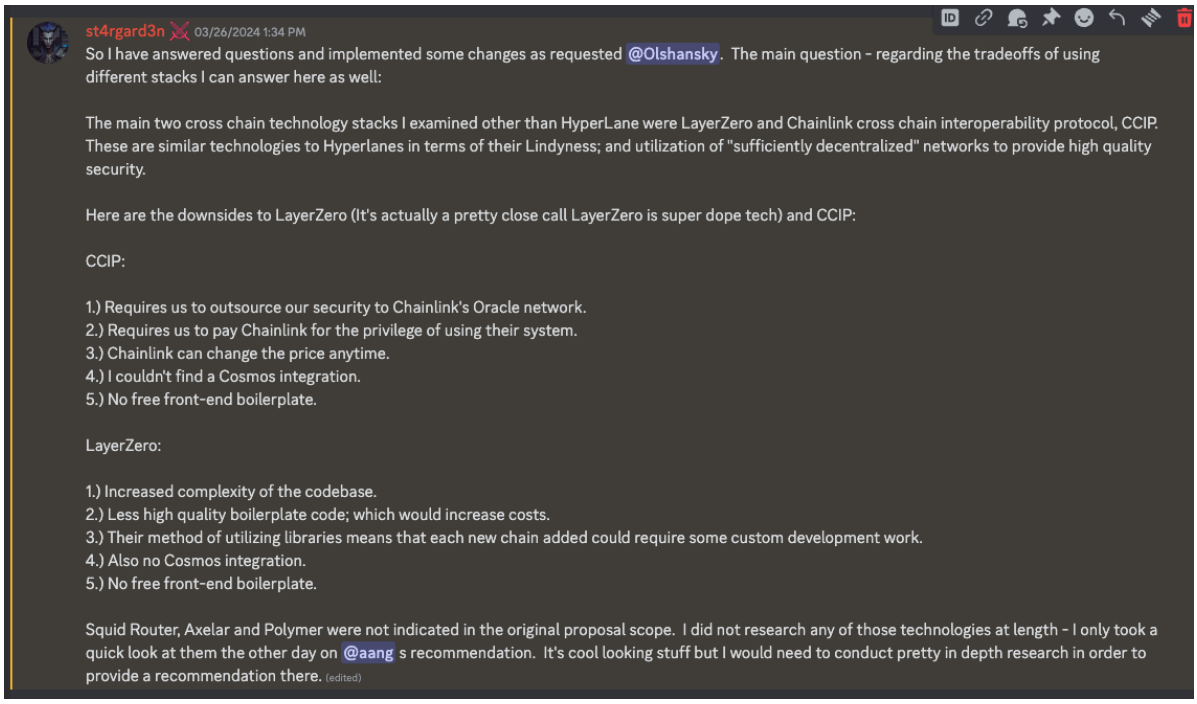
- Morse wPOKT bridge was built in-house due to the legacy frameworks & infrastructure it runs on.
  - 3 year old Tendermint fork
  - ABCI app but not Cosmos SDK compatible
  - Not IBC compatible
- Building a custom bridge (i.e. rolling our own) was the correct solution

Shannon (v1) context

- Cosmos SDK has a large developer ecosystem & community
- Bridging is a very common problem for all blockchains
- Key question: Is there something free, easy and off-the-shelf?

2. Exhaustive exploration of the bridging industry

- Source of concerns:
  1. Olshansky suggested to look into hyperlane and it ended up being selected as the final solution. Given Olshansky's lack of expertise and off-the-cuff suggestion, this raised concerns w.r.t what else may be better and out there.
  2. Hyperlane was only compared (in the proposal) against a custom solution in the original proposal. Intuition guides a custom solution to be off the table given that we are working off of mainline Cosmos.
- RaidGuild provided a deep comparison of the two following projects upon request:
  - CCIP
  - LayerZero



- Other solutions not researched in depth include:
  - “vanilla” IBC
  - Axelar (Custom independent network)
  - Squid Router (router on axelar)
  - Wormhole (custom independent network)
  - Polymer (EVM L2 to L2)
  - Union (zk but immature)
  - Router Protocol
  - Synapse
  - Piccaso
  - ???

3. Potential lock-in & future work

- Token transfer for POKT liquidity is the first and key goal

- The selected solution should not preclude future initiatives for general message passing to unblock deeper integrations; e.g. governance, EVM integration, etc...
- The selected 3rd party solution should not create vendor lock-in with a specific project/sdk/ecosystem

4. Lack of understanding of the current solution

- Lack of visuals of deep expertise of the existing solution made it hard to evaluate the scope of work that should be done w.r.t:
  - Maintaining & updating existing infra
  - Replacing existing infra
  - Integrating with existing infra
- This was resolved here: https://github.com/pokt-network/wpokt-validator/commit/7b48937b9aa96d0a25a10994af6762bd2651d0bb
- High level
  - `POKT <- Custom Infra -> wPOKT ERC20`
  - `wPOKT <- HyperLane -> Everything else`

Follow-up notes

The following projects ARE PROD READY but DO NOT USE IBC:

- Layer Zero
- Hyperlane
- Axelar
- Wormhole

Note: Discussions with ecosystem members showed that some projects "sell themselves" as IBC compatible but are not actually using IBC behind the scenes.

Benefits of IBC that most other protocols do not provide:

- Native ACKs / callback
- Native timeouts
- Refunds on failed transfers
- App (token) specific payment channels between specific SRC & TGT

tl;dr Why are we still going with the original Hyperlane proposal?

1. The cross-chain ecosystem is hard to navigate and more immature than it seems from the outside
2. Very few inter-chain protocol are ***IBC-native*** but rather just use IBC in some fashion in their stack
3. ***vanilla-IBC*** is available for free with the Cosmos SDK but would require maintaining extra infrastructure that Hyperlane provides
4. Hyperlane enables Pocket to maintain it's own bridging validator set and "potentially" move it elsewhere in the future.

Bonus

Request for a discussion to increase scope such that:

- Listeners & callbacks are added to listen on changes in Pocket's L1 validator set
- The entire Pocket validator set is used to sign the following bridge `POKT <– Custom Infra –> wPOKT ERC20`

Why?

- Increase the security of the bridge
- Make the security of the bridge as permissionless as pocket itself
- Potentially have the most secure bridge in the industry

When?

- Not during the migration work
- After Shannon mainnet launch