Together with Vaibhav Chellani from Socket I would like to present a risk framework proposal for assessing the security profile of different bridge architectures. The overall goal is similar to the risk framework for different L2s - to be able to quickly “classify” a given solution to a particular class of solutions having similar characteristic, while at the same time be detailed enough to present users with the security assumptions that they need to accept when using these bridges. Our focus is primarily on the bridges to/from Ethereum to other chain, as these we plan to present shortly on l2beat.com, however the basic reasoning about the security of these solutions apply to bridges from any chain to another chain. At this point we are looking for a broad community feedback on the proposed framework.

Bridge Type

[

Diagram showing hierarchy

2693×1248 141 KB

](https://europe1.discourse-cdn.com/standard20/uploads/l2beat/original/1X/1e7f130afa2a1db8542faeed49a307c3a15137cf.jpeg)

For the end user an asset bridge is something that accepts deposits for an asset on source-chain and gives user the asset on the destination chain. For eg: the typical flow is, Alice transfers funds to the bridge-contract on chainA, Alice received funds on chainB, simple.

Broadly speaking this happens in 2 ways:

- Message based Token Bridges
- These are bridges that allow liquidity flow via messages across-chains. They typically allow for minting assets on a destination chain upon locking or burning an asset on a source chain
- Examples: Rollup Bridges, Polygon Native bridge, Anyswap(anyCall), Axelar Network
- Examples: Rollup Bridges, Polygon Native bridge, Anyswap(anyCall), Axelar Network
- Liquidity Networks
- These are bridges that swap already minted assets. They allow users to move assets to other chains assuming these assets already were bridged there using “Message” bridges
- Examples: Connext on top of Nomad bridge, Hop on top of Hop Optimistic Bridge, some other HTLC and conditional transfers(nova etc)
- Examples: Connext on top of Nomad bridge, Hop on top of Hop Optimistic Bridge, some other HTLC and conditional transfers(nova etc)

Security for Message Bridges

In this section we try to explain different ways of validating a cross-chain message which has been leveraged by several bridging protocols out there. Token bridges leverage the security of the messaging bridge as shown on the diagram in the section above.

- Light client Verifying Validity of the State
- Description: Bridges that verify the validity of a state-transitions for source chain on destination chain. This is done via either zero-knowledge proof validation (state transition is accompanied with a zero-knowledge proof) or a fraud-proof system allowing independent Validators to dispute the validity of a new state root
- Examples: All rollups are an example of this, L1 validates state-transtitions for L2 either via FraudProof or ValidityProof.
- Description: Bridges that verify the validity of a state-transitions for source chain on destination chain. This is done via either zero-knowledge proof validation (state transition is accompanied with a zero-knowledge proof) or a fraud-proof system allowing independent Validators to dispute the validity of a new state root
- Examples: All rollups are an example of this, L1 validates state-transtitions for L2 either via FraudProof or ValidityProof.
- Light client Verifying Consensus
- Description: Bridges that validate the consensus of a source chain on a destination chain. Depending on the consensus being used, this typically involves checking the quorum signatures of the current Validator’s Committee if

the source chain uses a PBFT-style propose-and-vote consensus protocol (Tendermint, HotStuff, Casper FFG), or checking the longest chain with a relevant fork-choice rule if the source chain uses PoW or “longest-chain” - style PoS protocol (Ouroboros, ETH 2.0 LMD Ghost, etc…)

- Examples: NEAR Rainbow bridge (disregarding an Optimistic component related to the complexity of validating NEAR sig scheme there), Polygon PoS bridge (checking consensus of the Heimdall chain), Cosmos IBC (verifying signatures of another Cosmos chain)
- Description: Bridges that validate the consensus of a source chain on a destination chain. Depending on the consensus being used, this typically involves checking the quorum signatures of the current Validator’s Committee if the source chain uses a PBFT-style propose-and-vote consensus protocol (Tendermint, HotStuff, Casper FFG), or checking the longest chain with a relevant fork-choice rule if the source chain uses PoW or “longest-chain” - style PoS protocol (Ouroboros, ETH 2.0 LMD Ghost, etc…)
- Examples: NEAR Rainbow bridge (disregarding an Optimistic component related to the complexity of validating NEAR sig scheme there), Polygon PoS bridge (checking consensus of the Heimdall chain), Cosmos IBC (verifying signatures of another Cosmos chain)
- External Validator Set
- Description: Bridges that use external Validators (i.e. Validators that form a separate committee than Validators on either source or destination chain) as source of truth. Depending on the implementation these Validators may use a basic MultiSig, run a consensus algorithm (typically from a propose-and-vote family), use Threshold Signature schemes, SGX, etc… Regardless of the technique used, they all lie in this bucket
- Examples: Wormmhole, Multichain, Axelar, DeBridge, Synapse, Stargate
- Description: Bridges that use external Validators (i.e. Validators that form a separate committee than Validators on either source or destination chain) as source of truth. Depending on the implementation these Validators may use a basic MultiSig, run a consensus algorithm (typically from a propose-and-vote family), use Threshold Signature schemes, SGX, etc… Regardless of the technique used, they all lie in this bucket
- Examples: Wormmhole, Multichain, Axelar, DeBridge, Synapse, Stargate
- Optimistic Validation
- Description: Bridges that have a challenge period where honest parties are supposed to prevent fraud lie in this bucket. There are however a few key parameters to consider here:
- Challenge Duration: The larger the better
- Watcher-set size: Permissionless > Permissioned
- Challenge Duration: The larger the better
- Watcher-set size: Permissionless > Permissioned
- Examples: Hop Protocol, Connext Amarok, Across, Nomad Token Bridge
- Description: Bridges that have a challenge period where honest parties are supposed to prevent fraud lie in this bucket. There are however a few key parameters to consider here:
- Challenge Duration: The larger the better
- Watcher-set size: Permissionless > Permissioned
- Challenge Duration: The larger the better
- Watcher-set size: Permissionless > Permissioned
- Examples: Hop Protocol, Connext Amarok, Across, Nomad Token Bridge
- Hybrid
- Description: There exists constructions out there that are a mixture of some of the buckets defined above.
- Description: There exists constructions out there that are a mixture of some of the buckets defined above.

Security for Liquidity Networks

Apart from actually sending an asset cross-chain there is an alternative method, cross-chain swaps, where no assets move across chains but simply change hands, performing a cross-chain swap.

A quick example of this is: Alice on chainA wants to move an asset to chainB. Bob(Liquidity Provider) already has the asset on chainB and offers to swap the asset on chainB for Alice's balance on chainA for a fee. At the end Alice gets the asset on chainB, Bob gets the assets on chainA+fees.

This section only describes the security of the "swapping" protocol i.e how likely is it that your LP who accepts deposits on the source chain is going to run away with your deposit. The asset holds the security of the message bridge it was minted from.

There are a few different ways to do this as well:

- HTLC

: Also known as Hash Timelock Contracts, can be leveraged to atomically

swap assets between 2 parties across-chains. Usually takes 2 actions from the user, once to lock and once to unlock. The failure case is, your funds are locked for a fixed "expiry" period * Examples: Connext NXTP, Liquality

- Examples: Connext NXTP, Liquality
- Conditional Transfers

: Allows a Liquidity provider to short circuit a message bridge such that the LP provides the funds instantly to the end user and accepts the funds from the message bridge whenever they are bridged. The failure case here is that a slow path is activated if LP is unavailable. * Examples: Hop, Connext Amarok, MakerDAO Teleport

- Examples: Hop, Connext Amarok, MakerDAO Teleport
- External Validator

: Allows users to transfer funds to a trusted bridging provider, who promised to release funds on the other side. The failure case here is that your funds are lost. * Examples: Binance

- Examples: Binance

Censorship Resistance

We will look at the security assumptions related to the possibility of censorship of an individual message by the bridge. More practically we will ask whether an individual message (token transfer) can be censored/ignored by the bridge, and if this happens, what will be the consequence to the users funds (will they be returned to the user, or they will be stuck "in transit"). Typical solutions:

- Leveraging censorship resistance of the underlying chain (e.g. some Rollups)
- Relying on the honesty of the validator set

General Liveness Failure

For a general liveness failure we will look at the consequence of "switching off" the bridge. For example, for bridges using external Validator Sets we will have a look at the security of users funds in the event of these Validators going down for an extended period of time (possibly indefinitely). Typical scenarios include:

- Slow Path Activated

: Defaults to slow path: No loss of funds

- Self Stake

: Users can stake, join the network, become a validator and self process stuck transfers.

- Frozen

: System paused, cannot progress until the bridge operator comes online.

Liquidity

In this section we will try to analyse the liquidity available for bridging assets. Can the bridge mint assets, are LPs required, can users always withdraw/move whatever amount of tokens they choose to move or they rely on external liquidity providers and the bridge can "run out of funds"

- Unlimited

(bridge can mint native/canonical tokens)

- Permissioned

(bridge-operator provided)

- Permissionless

(any LP can provide liquidity)

Additional Considerations and Metrics

- Upgradability
- Permissioned Actors
- Volume transferred in a last 24 hours
- Unique transfers in a last 24 hours
- Liquidity available
- Supported tokens/chains