



# Bridge

Ze Jin, Minqi Zhang

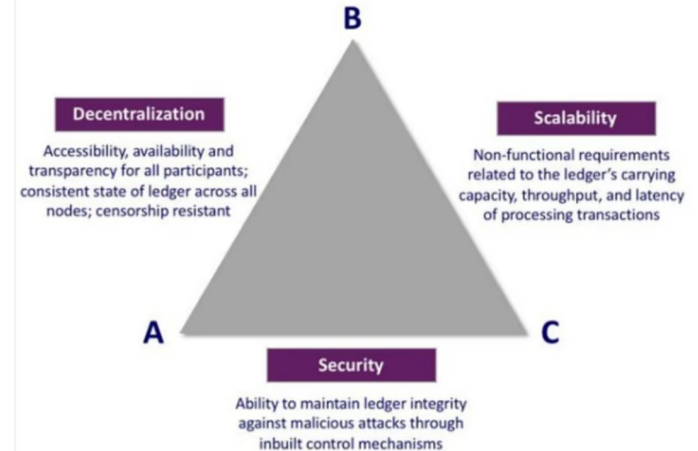


# What is a bridge

- Blockchain bridge **connects two blockchain** ecosystems.
- Bridges **facilitate communication** between blockchains through the **transfer of information and assets**

# Why Bridge

- All blockchains have their limitations(Trilema), we want to take advantage of different chains
- Blockchains cannot natively communicate.





## Why Bridge - High scalability, Low Gas fee

- Let's say you have ETH on Ethereum Mainnet but want cheaper transaction fees to explore different dapps.
- By bridging your ETH from the Mainnet to an Ethereum L2 rollup, you can enjoy lower transaction fees and high scalability.



## Why Bridge - Explore blockchain ecosystems

- Let's say you have ETH on Ethereum Mainnet and you want to explore Solana to try out their native dapps. You can use a bridge to transfer your ETH from Ethereum Mainnet to Solana.

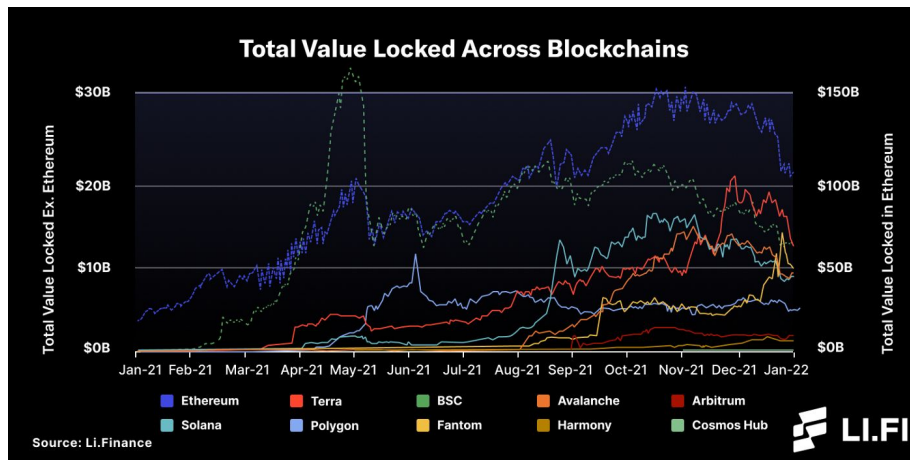


## Why Bridge - Dapps on other blockchains

- Let's say you've been using Aave on Ethereum Mainnet to lend USDT but the interest rate for lending USDT using Aave on Polygon is higher.

# Why Bridge - TVL

- Ethereum is the ecosystem where most of the Total Value Locked(TVL) in crypto is present, other smaller blockchains needs to connect to Ethereum to get fund.





## Why Bridge - Own native crypto assets

- Let's say you have ETH on Ethereum, you want to own native Bitcoin (BTC). To gain exposure to BTC on Ethereum, you can buy Wrapped Bitcoin (WBTC). However, WBTC is an ERC-20 token native to the Ethereum network, to own native BTC, you can bridge your assets from Ethereum to Bitcoin using a bridge. This will bridge your WBTC and convert it into native BTC.
- Alternatively, you might own BTC and want to use it in Ethereum DeFi protocols. This would require bridging the other way, from BTC to WBTC which can then be used as an asset on Ethereum.
- You can also do all of the above using an exchange. However, unless your funds are already on an exchange, it would involve multiple steps, and you'd likely be better off using a bridge.



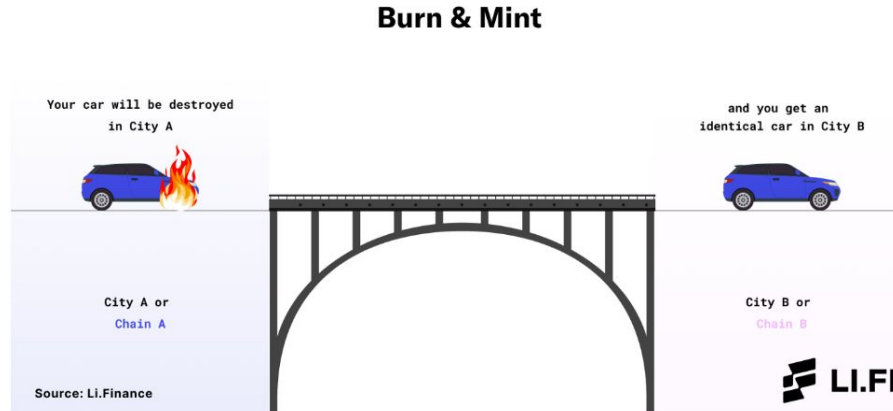
# How Native Bridge work - Lock & Mint

- These bridges lock assets on the source chain and mint assets on the destination chain. Examples: Polygon's PoS bridge, Avalanche Bridge (AB), wrapped BTC, wMonero.



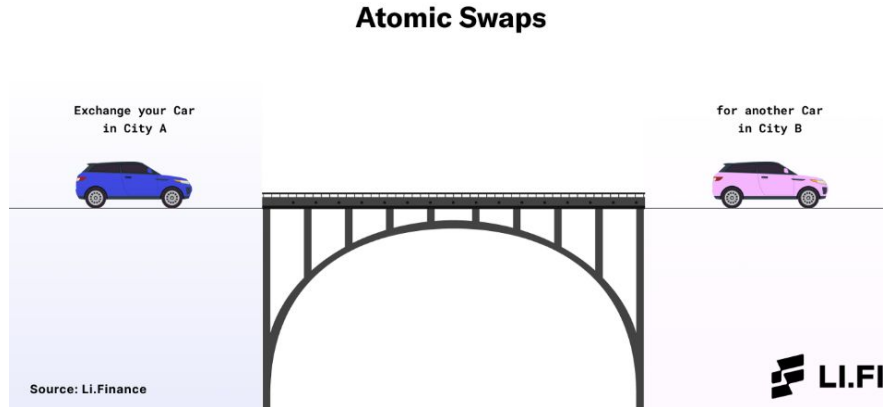
# How Native Bridge work - Burn & Mint

- These bridges burn assets on the source chain and mint assets on the destination chain. Examples: Hop, Across.



# How Native Bridge work - Atomic Swaps

- These bridges swap assets on the source chain for assets on the destination chain.
- Generally more trustless because they **rely on self-executing smart contracts for asset swaps and remove the requirement for a trusted third-party** necessary in lock & mint or burn & mint mechanisms. Examples: cBridge, Connex.





## Arbitrary Message Bridge

- Token transfers are just scratching the surface of what can be communicated between blockchains. Cross-chain bridges can do much more than pass token X from chain A to chain B.
- For example, bridges can be used to facilitate cross-chain governance, cross-chain token launches, cross-chain contract calls, gaming experiences, etc.



## Types of bridge - Trusted

- Trusted bridges depend upon a **central entity or system** for their operations.
- They have **trust assumptions** with respect to the custody of funds and the security of the bridge. Users mostly **rely on the bridge operator's reputation**.
- Users need to **give up control** of their crypto assets.



## Types of bridge - Trustless

- Trustless bridges operate using **smart contracts and algorithms**.
- They are **trustless**, i.e., the **security of the bridge is the same as that of the underlying blockchain**.
- Through smart contracts, trustless bridges enable **users to remain in control of their funds**.

## Off-chain actor

- Bridges use different mechanisms, or actors, that play the role of **verifiers between blockchains to enable communication and overcome the trust boundaries**. Without these off-chain actors, communication between blockchains will not be possible.





# Consensus





# Bridge Consensus Process

1. Confirm the transaction did happen on source chain
2. Execute the transaction on destination chain



## Bridge Consensus Mechanism - Optimistic(Nomad)

- Step 1 – Users or dApps post data to the home contract on the source chain, where all the **message/data is added and committed to a Merkle tree on source chain**.
- Step 2 – An off-chain agent called an **Updater signs the root of the Merkle tree** with the data.
- Step 3 – This **root is read and forwarded by the Relayer to the destination chain** in an “update”, posted to the replica contract.
- Step 4 – Once posted, a **30-minute fraud-proof window** opens up, during which a **Watcher can prove fraud** and stop the data from going through.
- Step 5 – If no fraud proofs are submitted within the 30-minute window, a **Processor submits the Merkle proof** of the data in the replica contract on the destination chain.



## Bridge Consensus Mechanism - Proof of stake(Axelar)

- Step 1 – A user requesting a cross-chain transfer of information waits for either the token deposit or **transaction to be confirmed by Validators on source chain.**
- Step 2 –source chain **Validators** observe their source chain nodes and **cast votes on whether the transaction occurred on source chain.**
- Step 3 – If the number of **votes surpasses the set threshold**, the source chain **transaction is confirmed by the source chainNetwork.**
- Step 4 – **The source chain Validators votes also act as signatures** for the Gateway smart contract on Axelar, which uses multi-party cryptography to secure the issuance of tokens/data across chains. Once enough Validators confirm a transaction on chain A, **the requisite data on destination chain is relayed to the deposit address.**



## Bridge Consensus Mechanism - Oracle(Layer Zero)




- Step 5 – The Network sends the identifier for the smart contract on destination chain along with the block ID of the transaction on source chain to the Oracle. As a result, the **Oracle is notified to fetch the block header for the current block on source chain.**
- Step 8 – The **Oracle waits for a certain number of block confirmations before confirming that the block has been successfully committed on source chain.** Post confirmation, the **Oracle sends the block header to the Network on destination chain.**







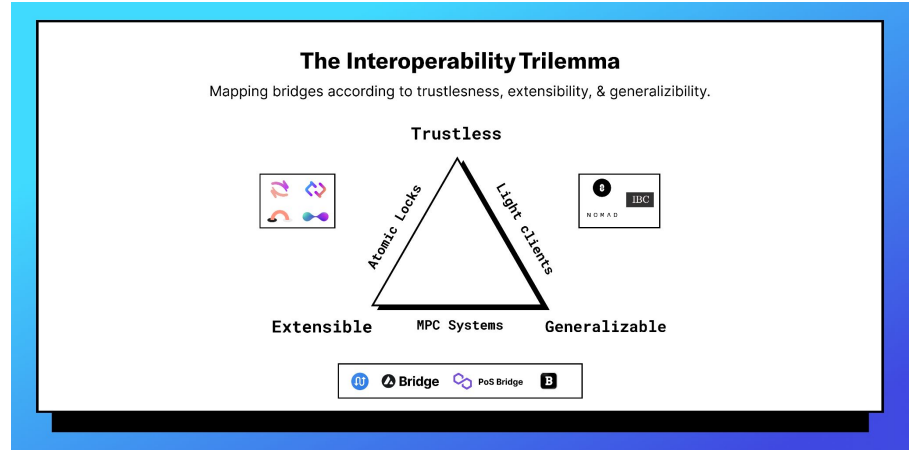
## How to hack/collude a bridge

- hack smart contract
- centralized - hack/collude the Guardians. e.g 13/19 Guardians on Wormhole
- proof of stake - hack/collude validators e.g 33/48 validators on Axelar
- optimistic - hack/collude updater and watcher
- oracle - hack/collude oracle and relayer
- forge signature

# How to secure a bridge

| Messaging Bridge   | Consensus Mechanism  |
|--|--|
|  Axelar   | Delegated Proof of Stake + Weighted Threshold Signature Scheme |
|  Nomad    | Optimistic   |
|  Wormhole | Multi-Sig  |

|   |   |
|---|---|
|  LayerZero | Independent Oracle and Relay  |
|  Celer IM  | Specialized Proof of Stake or Optimistic Rollup-like model                          |
|  anyCall   | Secure Multi-Party Computation (SMPC) + Equally-Weighted Threshold Signature Scheme |
|  Hyperlane | Delegated Proof of Stake + Sovereign Consensus                                      |





## How to secure a bridge

- **Audits and bounty system**— Wormhole has been audited by Neodyme and Kudelski (x2). It has audits by OtterSec, Certik, Halborn, Trail of Bits, and Coinspect scheduled for Q3 2022. Moreover, it has the largest bounty in the crypto space via a \$10 million offer on Immunefi.
- **White-hat hackers to contribute** — Wormhole has developed several strategies to make it easier for white-hat hackers to find security bugs in Wormhole, disclose them, and help secure the network. For instance, whitehats can review Wormhole's existing unit and integration test and disclose vulnerabilities.



# Our Bridge





# Goal of our bridge

- **Minimize Trust**

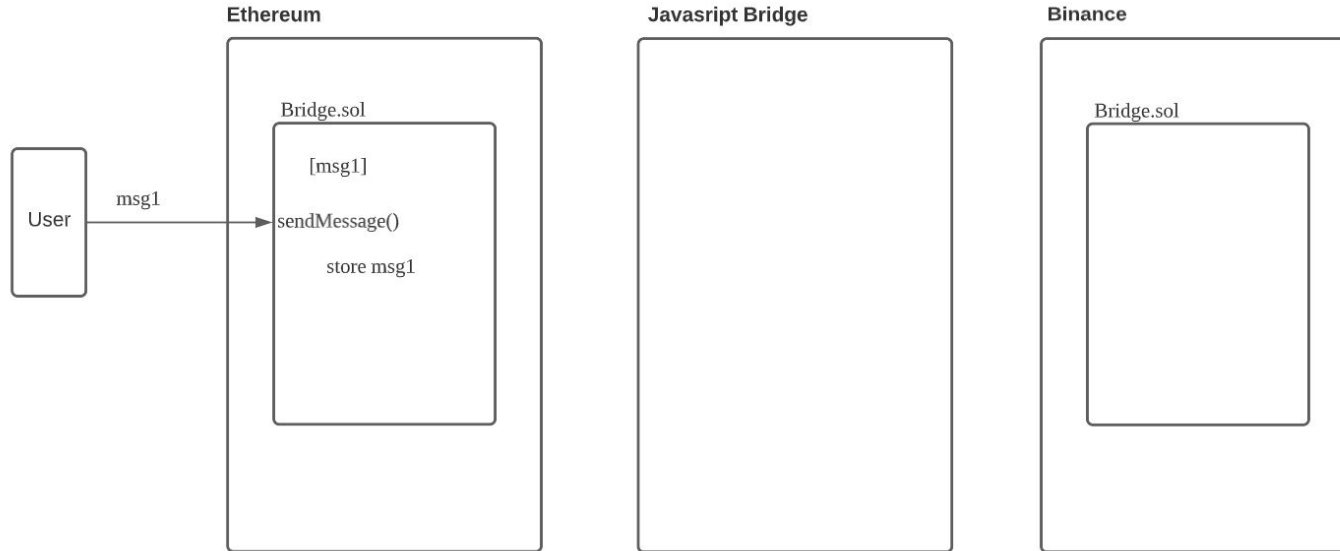
=> Do as little work on the bridge as possible

=> Do as much work as possible on the Chain

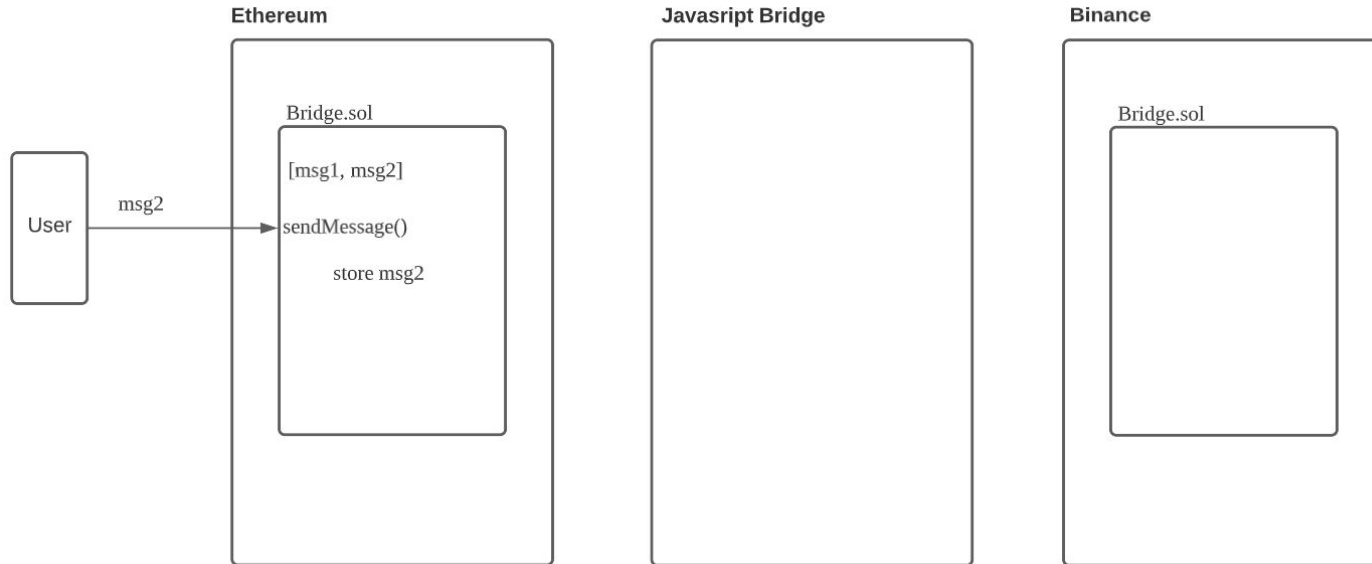
- **Arbitrary Message**

- **Bi-directional, Ethereum ↔ Binance**

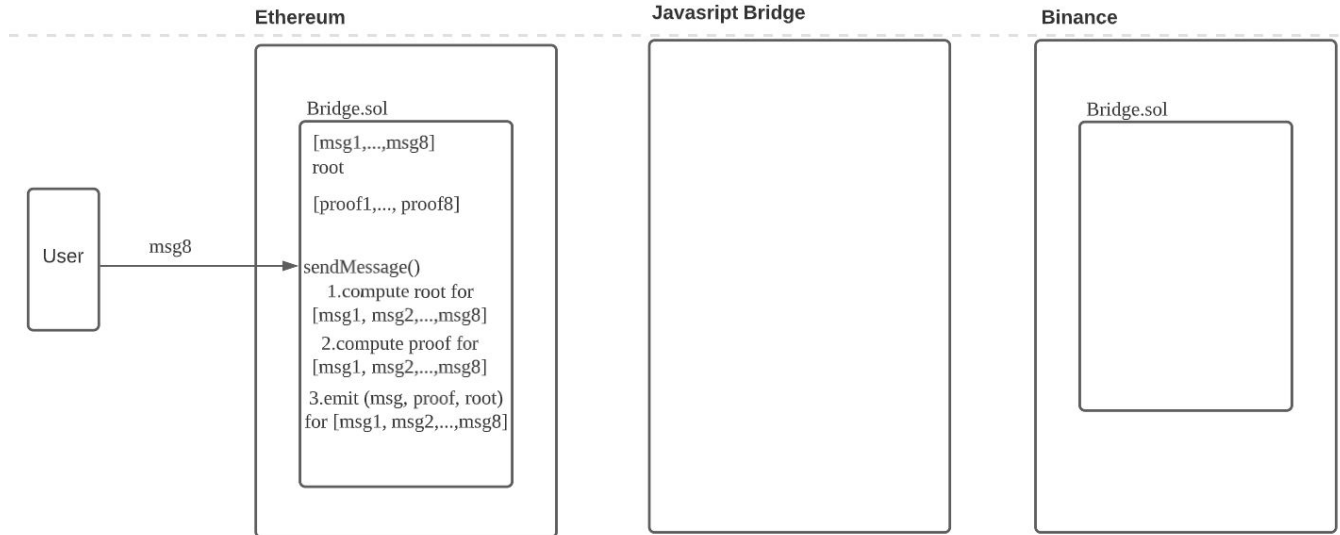
## How our bridge work - before batch size



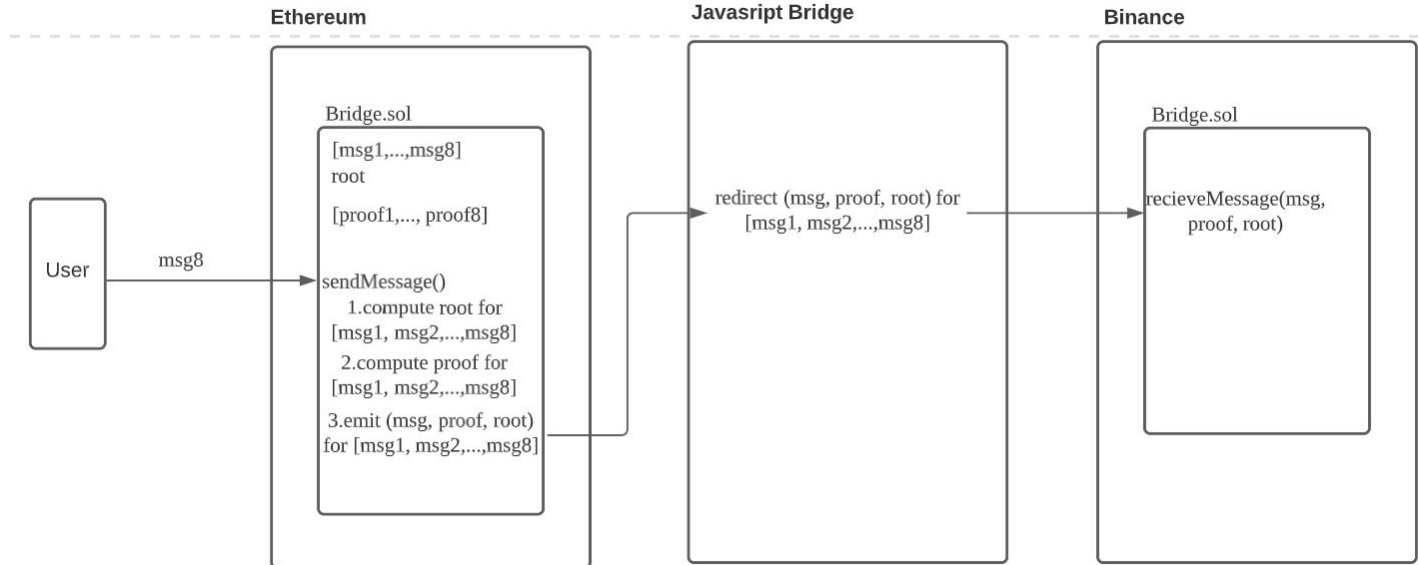
## How our bridge work - before batch size



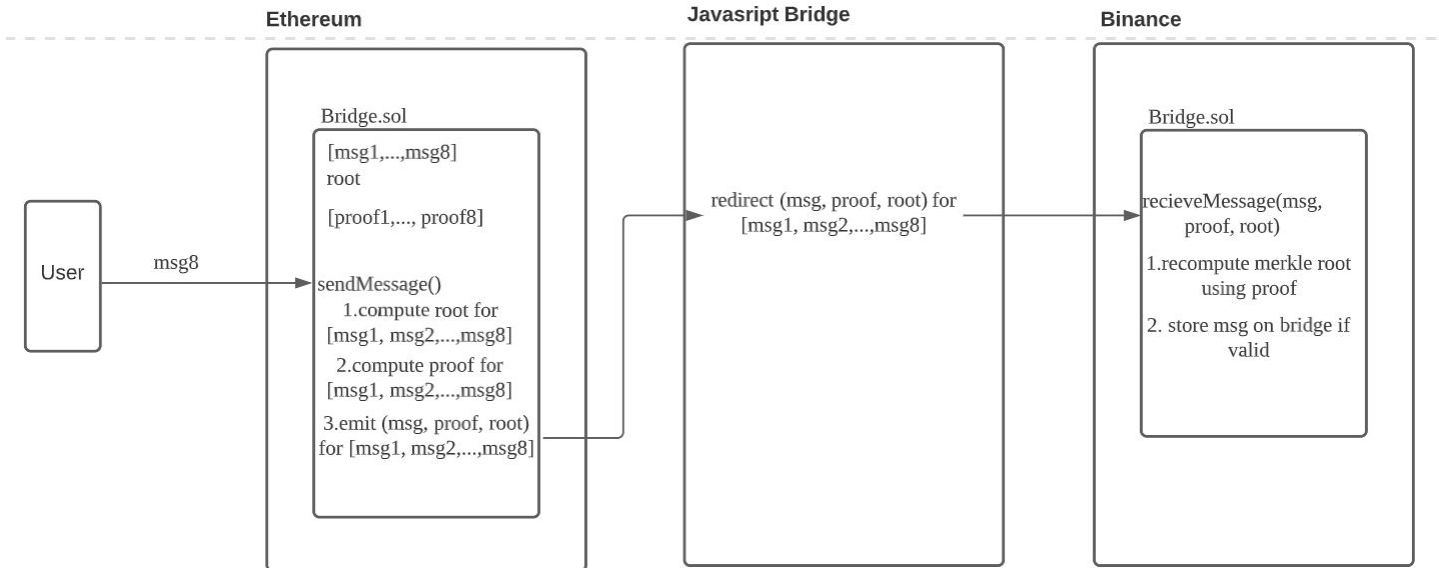
# How our bridge work - at batch size



# How our bridge work - at batch size



# How our bridge work - at batch size





## Reference

<https://li.fi/knowledge-hub/blockchain-bridges/>

<https://li.fi/knowledge-hub/blockchain-bridges-and-classification/>

<https://li.fi/knowledge-hub/the-evolution-of-native-bridges/>

<https://li.fi/knowledge-hub/navigating-arbitrary-messaging-bridges-a-comparison-framework/>