Introduction

We're excited to see and engage with the open RFP for wstETH on BNB. Discussion, knowledge-gathering, and feedback cycles around Lido's cross-chain expansion has been robust and encouraging to all of us building in the interoperability space.

LayerZero Labs submits this proposal to seek the Lido DAO's formal recognition of this deployed wstETH token on BNB Chain as canonical. Should this proposal receive positive community feedback and DAO approval, LayerZero Labs will transfer ownership of the wstETH contracts to the Lido DAO and support Lido technical contributors with the configurations recommended below.

This proposal is written in accordance with the Network Expansion Workgroup's (NEW) unofficial guidelines for bridging solutions. Our implementation and engagement with the DAO would not be possible without the ongoing feedback and encouragement from members of Lido NEW.

Overview

Formally recognizing a bridged wstETH token on BNB Chain opens up enormous growth opportunities for Lido by simplifying both user experience and developer experience and increasing DeFi composability across multiple chains.

Per community feedback and the unofficial guidelines for bridging solutions, the ideal wstETH on BNB implementation achieves the following:

- Enables the Lido DAO to easily add or remove bridges and other validators (ex. zkLightClient, zkOracle, DON)
- Minimizes liveness concerns by using a variety of validation mechanisms
- Leverages battle-tested code to handle the substantial influx of wstETH into BNB without exposing Lido to an exploit or new surface attack vectors
- Endows the Lido DAO with full ownership and control of the contracts
- Gas efficient and gas-abstracted for end-users; it should be inexpensive and simple to bridge

Technical Implementation

Primary Components:

wstETH contracts -

utilizes the Omnichain Fungible Token (OFT) open-source reference implementation contracts. The OFT Standard is an extension of the ERC20 Standard and is built on LayerZero protocol through a mint and burn mechanism. OFTs can be sent/received across any blockchain LayerZero supports while maintaining unified liquidity – though projects maintain rights to pick and choose which blockchains those are.

This standard implementation has been live and in production for 18 months, securing over $4.5 billion in transfer volume across 45+ chains. In addition, the OFT Standard has been extensively audited multiple times with no critical issues found.

Endpoint -

immutable smart contract libraries deployed on each supported chain that can never be upgraded by the deployer, LayerZero Labs.

Endpoint libraries are append-only and accommodate new technologies without forced upgrades on applications. Developers enjoy a unified interface on all chains.

[

1600×539 27.8 KB

](https://europe1.discourse-cdn.com/business20/uploads/lido/original/2X/3/39ef07d6c18e2879eb94f47a4e5b8936587f9833.png)

Oracle Configuration –

contract owners always maintain the ability to configure the x/n validation layer. Note: expanding x/n increases overall gas cost. For the wstETH on BNB implementation, we recommend configuring the contracts to select at least the:

(1) zkLightClient and

(2) one of the aforementioned bridges, per the preference of the DAO

Explanation for Oracle Configuration:

The LayerZero protocol is specifically solving the transportation layer, not the validation layer.

LayerZero protocol features a single immutable endpoint smart contract on each chain. From there, LayerZero protocol has an append-only registry for libraries, which allows OApps to configure a relayer/oracle pairing, along with other security parameters, while the Ultra Light Node (ULN) acts as an on-chain validation contract within the Message Library (MsgLib). Each message sent through LayerZero protocol is nonce-order enforced. Messages are confirmed by an oracle and a relayer, who are assumed to be independent, and executed via a relayer.

Within this architecture, LayerZero protocol simply acts as a conduit for teams like the Lido DAO to build applications, choose a security setup, and then begin transmitting messages.

The emphasis here is that the LayerZero protocol does not seek to solve the unique and opinionated problem of validation methodology. Rather, LayerZero's open and permissionless validation layer (aka the Oracle entity) is designed to support any set of methodologies. In fact, the validation methodologies of third party bridges such as Axelar, Wormhole, and CCIP could be leveraged directly within the LayerZero protocol.

The wstETH contracts enable contract owners to configure security parameters of x/n validators as an Oracle, thereby creating a meta-Oracle of sorts. Therefore, the Lido DAO can require multiple validation methodologies to sign off on each message going between chains. Within LayerZero's current setup, this means requiring signatures from some combination of entities among Oracles (validation methodologies) like Chainlink DON, Polyhedra's zkLightClient, Google Cloud Oracle and future validators like optimistic oracles and zkOracles. Other validation techniques – like third party bridges – can also be used in the Oracle as well.

In light of this proposal, LayerZero Labs recommends that the Lido DAO configure its Oracle to require two entities – Polyhedra's zkLightClient and a bridge of its own choice.

However, if Lido DAO wanted to, it could go so far as to point to multiple 3rd party bridges as validation mechanisms, thereby essentially achieving messaging aggregation inside the LayerZero protocol framework. This would allow the Lido DAO to leverage the (1) middle-chain security model of one protocol, (2) the guardian validator set of another, and (3) full Ethereum consensus via a zkLightClient, all while preventing a centralized dependency on the liveness of one of the aggregated protocols for the integrity of the token transfer.

Furthermore, if the Lido DAO wished to include a (4) delegated service provider (e.g. P2P, Nethermind, etc) the DAO could vote to extend the x/n to include it at any time. This further backstops the Lido DAO from malicious behaviour arising from corrupted validation mechanisms. A quick showcase of this configuration is shown below:

[

1582×878 181 KB

](https://europe1.discourse-cdn.com/business20/uploads/lido/original/2X/e/ecce39984996eea6478e8d3b30ef48b78faec977.jpeg)

To summarize briefly, the validation layer within the LayerZero framework empowers contract owners (e.g. Lido DAO) to expand and define their level of decentralization without writing any additional code. The above configuration example would result in wstETH which uses the full security of two independent bridges, zero-knowledge proofs, and a backstop of a community-run validator.

Transaction Lifecycle

User transfers wstETH from Ethereum to BNB:

1. User pays for gas upfront in ETH.
2. The user interacts with the wstETH proxyOFT contract on Ethereum, which sends a message to the wstETH contract on BNB.
3. The user's native stETH tokens lock in the wstETH ProxyOFT contract on Ethereum owned by the Lido DAO.
4. The message is sent to the Endpoint and is passed to the contract-configured ULN, which emits an event containing the message.
5. Upon the emission of the event containing the message, the Lido DAO validators and Relayer (both entities listening for the event) will trigger the fetching of the corresponding block hash and transaction proof respectively.
6. After the block confirmations specified within the wstETH contracts have passed, the selected validators will deliver the block header to the destination chain's contract-configured ULN.

7. After the block header is submitted by the selected validators, the ULN will emit a notification stating it received the block header.
8. The contract-selected Relayer will be listening for the event and upon seeing it will submit the corresponding transaction proof.
9. Upon receiving the transaction proof from the Relayer, the ULN verifies that the transaction proof matches the block hash provided by the validators.
10. If they match, the message is delivered to the LayerZero Endpoint. From there, the Endpoint tells the wstETH contract on BNB Chain to mint and deliver wstETH to the end user.
11. If they do not match, the message will be deemed invalid and the transaction will revert; only one single honest party is needed for a malicious message to revert.

User transfers wstETH from BNB to Ethereum:

Same as the above steps, except users pay upfront in the native gas token of the source chain and the wstETH token is burned on BNB and the locked stETH tokens are released on Ethereum.

Why is this implementation best suited for the Lido DAO's needs?

- LayerZero remains the only interoperability protocol at-scale (85M+ messages delivered) to never experience an exploit.
- LayerZero has secured $30B+ in lifetime volume.
- LayerZero has consistently operated with the principle that security is the highest priority. Last year, we launched crypto's largest bug bounty - a $15 million bounty in collaboration with ImmuneFi - and commissioned $3m+ of security audits.
- LayerZero has secured $30B+ in lifetime volume.
- LayerZero has consistently operated with the principle that security is the highest priority. Last year, we launched crypto's largest bug bounty - a $15 million bounty in collaboration with ImmuneFi - and commissioned $3m+ of security audits.
- All applications utilizing the LayerZero protocol have the option to be protected by Pre-Crime. Pre-Crime is a proprietary security module that simulates incoming transactions against an application-defined rule set, ensuring that invariants are checked before the delivery of each message. LayerZero Labs created Pre-Crime in response to the common security failures of major cross-chain protocols.
- Using Pre-Crime, the Lido DAO can customize the specific set of checks to run prior to message transmission. If any rules within the set fail, the transaction will be blocked at the source. This adds a layer of additional security – a live risk management and monitoring system – without introducing any additional latency.
- Using Pre-Crime, the Lido DAO can customize the specific set of checks to run prior to message transmission. If any rules within the set fail, the transaction will be blocked at the source. This adds a layer of additional security – a live risk management and monitoring system – without introducing any additional latency.

[

1456×780 48.3 KB

](https://europe1.discourse-cdn.com/business20/uploads/lido/original/2X/7/7170c387184acb63a45f659a2843767db3b60e84.png)

Conclusion

While we applaud the thoughtful discourse, we believe that this proposed implementation is the best solution for wstETH. LayerZero protocol is built to be a transport layer and is inherently validation mechanism agnostic, which seems to be a trait much prioritized by the industry writ large. With this bridged token implementation and the architecture of the Endpoint, the Lido DAO will have complete control over wstETH and governance without being locked in to a single validation type.

To speak briefly on the Axelar and Wormhole proposal, LayerZero would like to emphasize that moving wstETH across chains under the Wormhole-Axlear architecture relies on powerful BridgeManager contracts written just for this purpose. Rather than using newly written, unaudited (or briefly audited) BridgeManager contracts – which effectively function as Endpoints– the LayerZero proposal puts forth an implementation which uses Endpoints that have secured $30B+ value over nearly 2 years across LayerZero protocol.

Additionally, with LayerZero, contract owners (e.g. Lido DAO) can swap or change the x/n configuration at any time with light engineering lift. In other words, the DAO maintains the flexibility to remove validation methodologies such as bridge providers, in addition to adding them per governance vote. This comes in stark contrast to the Wormhole-Axelar proposal, which effectively locks the Lido DAO into a 2/2 multi-sig between the bridges. If either bridge experiences liveness issues or is hacked, this could cause major disruptions for wstETH. Finally, the implementation proposed by Wormhole-Axelar uses code has never been live and in-production.

Regarding the Multi Message Aggregation (MMA) implementation, we recognize this exciting collaborative effort by Li.Fi, Kydo, Celer to be a thoughtful response to governance-specific use cases. MMA prioritizes governance, which limits user experience and functionality. Adding more features may require new audits, potentially weakening the original design and introducing additional risk to Lido.

Commitment to BNB Chain

LayerZero Labs believes commitment to BNB Chain is important to mention in this conversation – as any Endpoint must be maintained long-term, updated to match anything that occurs on BNB Chain, etc.

With that in mind, LayerZero protocol has been live on BNB Chain since mainnet launch in April 2022. While BNB Chain does not have a native bridge, LayerZero is proud to be the interoperability provider for the top 3 native protocols: Radiant, PancakeSwap, and Venus. Additionally, LayerZero protocol supports dApps like Stargate Finance, the most adopted bridge for users on BNB Chain (DefiLlama).

Timeline:

- As of December 5, 3+ audits have been completed for the underlying token contracts.
- January → Commission additional audits of the wstETH bridged token implementation after recommended configurations are made
- We intend to work with MixBytes and ChainSecurity on these audits
- We intend to work with MixBytes and ChainSecurity on these audits
- February → Pending DAO vote and NEW preferences, transfer contract ownership to the DAO

Appendix

- Contracts and Github
- Reference Diagrams
- Audits
- Bug Bounty
- Documentation