Quantstamp

OpenLedger - OFT Adapter

Executive Summary

This audit report was prepared by Quantstamp, the leader in blockchain security.

Type	Cross-Chain Token
Timeline	2025-07-09 through 2025-07-09
Language	Solidity
Methods	Architecture Review, Unit Testing, Functional Testing, Computer-Aided Verification, Manual Review
Specification	README.mdbridge.md
Source Code	• openledger-dev/openledger-layerzero #d7aed17
Auditors	• Hytham Farah Auditing Engineer • Andrei Stefan Auditing Engineer

Documentation quality	Undetermined
Test quality	Undetermined
Total Findings	0
High severity findings	0
Medium severity findings	0
Low severity findings	0
Undetermined severity findings	0
Informational findings	0

Summary of Findings

This repo implements a minimal bridge intended to be used for OpenLedger's Open token to become omnichain-compatible using LayerZero's OFTAdapter standard. It is designed to wrap an existing token (not deploy a new one), enabling secure cross-chain transfers in accordance with the Omnichain Fungible Token (OFT) standard. The repository is a fork of the LZ example implementation, and the changes made by the OpenLedger team are minimal. The focus here is on clarity, auditability, and minimalism. This gave auditors the space to focus more on deployment scripts and procedures.

We note that the test suite is robust with coverage metrics boasting 100% across all branches.

The audit identified no significant vulnerabilities; however, it left several suggestions around the documentation and deployment processes. All suggestions are strictly best-practice recommendations—chiefly around finalizing mainnet deployment scripts and setup—to ensure production readiness.

FIX-REVIEW UPDATE: The client has fixed all suggestions outlined in the report. These changes overall move the client towards production readiness with additional mainnet deployment configurations added.

Assessment Breakdown

Quantstamp's objective was to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices.

> **Disclaimer**
> Only features that are contained within the repositories at the commit hashes specified on the front page of the report are within the scope of the audit and fix review. All features added in future revisions of the code are excluded from consideration in this report.

Possible issues we looked for included (but are not limited to):

- Transaction-ordering dependence
- Timestamp dependence
- Mishandled exceptions and call stack limits
- Unsafe external calls
- Integer overflow / underflow

- Number rounding errors
- Reentrancy and cross-function vulnerabilities
- Denial of service / logical oversights
- Access control
- Centralization of power
- Business logic contradicting the specification
- Code clones, functionality duplication
- Gas usage
- Arbitrary token minting

Methodology

1. Code review that includes the following
   1. Review of the specifications, sources, and instructions provided to Quantstamp to make sure we understand the size, scope, and functionality of the smart contract.
   2. Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
   3. Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to Quantstamp describe.
2. Testing and automated analysis that includes the following:
   1. Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run those test cases.
   2. Symbolic execution, which is analyzing a program to determine what inputs cause each part of a program to execute.
3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarity, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.
4. Specific, itemized, and actionable recommendations to help you take steps to secure your smart contracts.

Scope

Files Included

Repo: https://github.com/openledger-dev/openledger-layerzero/commit/d7aed175d594df1ca9b7bc2245c97f01d0ad0a2b(d7aed175d594df1ca9b7bc2245c97f01d0ad0a2b) Files: contracts/OmnichainOpen.sol contracts/OPENOFTAdapter.sol
Repo: `https://github.com/openledger-dev/openledger-layerzero`

Files Excluded

Repo: https://github.com/openledger-dev/openledger-layerzero/commit/d7aed175d594df1ca9b7bc2245c97f01d0ad0a2b(d7aed175d594df1ca9b7bc2245c97f01d0ad0a2b) Files: contracts/mocks

Operational Considerations

1. The Open token inherits the `ERC20Burnable` contract, adding burn functionality to the token. It is therefore possible for a scenario to occur where there are locked tokens in the source chain that cannot be redeemed because they were burned on the destination chain. This is not a vulnerability, but it may complicate accounting.
2. The contract extends LayerZero's OFT, so it assumes the OFT implementation is correctly deployed, audited, and maintained. Any bugs, upgrades, or deprecations in the OFT library will directly affect token bridging behavior.
3. The `_lzEndpoint` address passed at construction must point to the correct LayerZero endpoint for the target chain. A misconfigured or incorrect endpoint will prevent all cross-chain operations.
4. Cross-chain transfers rely on LayerZero's messaging relayers and endpoint uptime. If the relayer network experiences outages, delays, or misconfigurations, token transfers will stall or revert until service is restored.
5. LayerZero message delivery does not guarantee strict ordering, and the OFT contracts do not implement sequence number validation. Successful transfer from a user could be delivered in different orders.

Key Actors And Their Capabilities

Owner

Responsibility

- Administer cross-chain token configuration and LayerZero messaging parameters to ensure secure, reliable transfers.
- Oversee contract governance actions such as updating remote chain mappings and adapter settings.
- Manage ownership lifecycle (transfer or renunciation) to control access to all owner-only functionality.

Trust Assumption

- Owner will only point the contract at legitimate LayerZero endpoints and counterparty contracts.
- Owner will not misconfigure gas parameters or adapter settings in a way that blocks or censors user transfers.
- Owner will act in good faith when transferring or renouncing ownership, and will not lock out or abandon the protocol maliciously.

Exclusive Functions

All the exclusive functions are inherited from the LayerZero contracts as follows:

1. **LayerZero Admin:**
    1. `setTrustedRemote()`
       Map a source chain ID to its trusted remote contract address path, ensuring only messages from that path are accepted.
    2. `setUseCustomAdapterParams()`
       Enable or disable passing custom adapter parameters (for advanced gas/payment controls) on sends.
    3. `setMinDstGas()`
       Configure the minimum gas reservation at the destination chain for various message types.
    4. `setSendVersion()`
       Upgrade the LayerZero “send” library version to match endpoint expectations.
    5. `setReceiveVersion()`
       Upgrade the LayerZero “receive” library version for incoming messages.
    6. `forceResumeReceive()`
       Clear a blocked message state for a given source chain and address so processing can resume.

Auditor Suggestions

S1 There Is No Mainnet Deployment Configuration

Fixed

> ### Update
> Marked as "Fixed" by the client.
> Addressed in: `eea82c8e6679b6a12f634f582bf0403c36147e33`.

> ### Update
> Mainnet configurations have been added; however, the zero address is used as the Ethereum mainnet `OFTAdapter` token.

File(s) affected: `hardhat.config.ts`

Description: There are no mainnet setup configurations present in the repository.

Recommendation: Add mainnet deployment configuration

S2 Incorrect Contract Names used in Configurations

Fixed

> ### Update
> Marked as "Fixed" by the client.
> Addressed in: `9a7b7301ae3719152ba114ad154c025394c50be6`.

File(s) affected: `layerzero.config.ts`

Description: The `layerzero.config.ts` file still uses the default contract names instead of the actual deployed contract name. This can cause confusion or misconfiguration during deployment. Since these are all test networks, the impact so far is negligible and, at most, may result in misleading tests.

Recommendation: Update all `contractName` to match the actual implementation name: `OPENOFTAdapter`.

Definitions

- **High severity** – High-severity issues usually put a large number of users' sensitive information at risk, or are reasonably likely to lead to catastrophic impact for client's reputation or serious financial implications for client and users.
- **Medium severity** – Medium-severity issues tend to put a subset of users' sensitive information at risk, would be detrimental for the client's reputation if exploited, or are reasonably likely to lead to moderate financial impact.
- **Low severity** – The risk is relatively small and could not be exploited on a recurring basis, or is a risk that the client has indicated is low impact in view of the client's business circumstances.
- **Informational** – The issue does not pose an immediate risk, but is relevant to security best practices or Defence in Depth.
- **Undetermined** – The impact of the issue is uncertain.
- **Fixed** – Adjusted program implementation, requirements or constraints to eliminate the risk.
- **Mitigated** – Implemented actions to minimize the impact or likelihood of the risk.

- **Acknowledged** – The issue remains in the code but is a result of an intentional business or design decision. As such, it is supposed to be addressed outside the programmatic means, such as: 1) comments, documentation, README, FAQ; 2) business processes; 3) analyses showing that the issue shall have no negative consequences in practice (e.g., gas analysis, deployment settings).

Appendix

File Signatures

The following are the SHA-256 hashes of the reviewed files. A file with a different SHA-256 hash has been modified, intentionally or otherwise, after the security review. You are cautioned that a different SHA-256 hash could be (but is not necessarily) an indication of a changed condition or potential vulnerability that was not within the scope of the review.

Files

Repo: `https://github.com/openledger-dev/openledger-layerzero`

- `020...208 ./.env-sample`
- `167...a16 ./.eslintignore`
- `9c3...106 ./.eslintrc.js`
- `435...1b1 ./.gitignore`
- `f22...4f3 ./.nvmrc`
- `beb...d6e ./.prettierignore`
- `73d...9e1 ./.prettierrc.js`
- `0ef...126 ./README.md`
- `919...674 ./contracts/OPENOFTAdapter.sol`
- `534...adc ./contracts/OmnichainOpen.sol`
- `8d1...b39 ./contracts/mocks/MyERC20Mock.sol`
- `b96...99b ./contracts/mocks/MyOFTAdapterMock.sol`
- `e0c...2b2 ./contracts/mocks/MyOFTMock.sol`
- `620...572 ./contracts/mocks/Open.sol`
- `d12...c35 ./deploy/MyERC20Mock.ts`
- `8ca...4c6 ./deploy/MyOFT.ts`
- `407...9b1 ./deploy/MyOFTAdapter.ts`
- `f1f...bb3 ./deploy/OpenMock.ts`
- `3b9...ef8 ./deployments/base-testnet/.chainId`
- `a77...d4d ./deployments/base-testnet/MyOFT.json`
- `d2f...fb4 ./deployments/base-testnet/solcInputs/85f673901b09a646dff1284a1194f121.json`
- `d6d...091 ./deployments/bnb-testnet/.chainId`
- `09a...9c1 ./deployments/bnb-testnet/MyOFT.json`
- `d2f...fb4 ./deployments/bnb-testnet/solcInputs/85f673901b09a646dff1284a1194f121.json`
- `a89...659 ./deployments/sepolia-testnet/.chainId`
- `1d4...8e5 ./deployments/sepolia-testnet/MyOFTAdapter.json`
- `d2f...fb4 ./deployments/sepolia-testnet/solcInputs/85f673901b09a646dff1284a1194f121.json`
- `6b7...01f ./docs/bridge.md`
- `c13...8f4 ./foundry.toml`
- `10b...bda ./hardhat.config.ts`
- `c40...780 ./layerzero.config.ts`
- `238...17d ./package-lock.json`
- `fed...c83 ./package.json`
- `2ca...798 ./solhint.config.js`
- `814...7a9 ./tasks/sendEvm.ts`
- `132...034 ./tasks/sendOFT.ts`
- `136...8d0 ./tasks/types.ts`
- `a21...46a ./tasks/utils.ts`
- `5a7...20c ./test/foundry/MyOFT.t.sol`
- `e53...c6a ./test/foundry/MyOFTAdapter.t.sol`
- `770...abe ./test/hardhat/MyOFT.test.ts`
- `37a...855 ./test/hardhat/MyOFTAdapter.test.ts`

- `9d6...8f0 ./test/mocks/ERC20Mock.sol`
- `f54...32d ./test/mocks/OFTAdapterMock.sol`
- `107...676 ./test/mocks/OFTComposerMock.sol`
- `af0...9cd ./test/mocks/OFTMock.sol`
- `555...0b1 ./tsconfig.json`
- `816...22f ./type-extensions.ts`

Test Suite Results

THe test suite supports testing with both foundry and hardhat. The data for both is provided below.

```
FOUNDRY

Ran 3 tests for test/foundry/MyOFT.t.sol:MyOFTTest
[PASS] test_constructor() (gas: 34825)
[PASS] test_send_oft() (gas: 696928)
[PASS] test_send_oft_compose_msg() (gas: 1449096)
Suite result: ok. 3 passed; 0 failed; 0 skipped; finished in 8.51ms (4.64ms CPU time)

Ran 3 tests for test/foundry/MyOFTAdapter.t.sol:MyOFTAdapterTest
[PASS] test_constructor() (gas: 42508)
[PASS] test_send_oft_adapter() (gas: 763047)
[PASS] test_send_oft_adapter_compose_msg() (gas: 1502289)
Suite result: ok. 3 passed; 0 failed; 0 skipped; finished in 8.53ms (4.56ms CPU time)

Ran 2 test suites in 207.27ms (17.04ms CPU time): 6 tests passed, 0 failed, 0 skipped (6 total tests)
```

```
HARDHAT

Version
=======
> solidity-coverage: v0.8.16

Instrumenting for coverage...
=============================

> mocks/MyERC20Mock.sol
> mocks/MyOFTAdapterMock.sol
> mocks/MyOFTMock.sol
> mocks/Open.sol
> OmnichainOpen.sol
> OPENOFTAdapter.sol

Compilation:
============

Compiled 47 Solidity files successfully (evm target: paris).

Network Info
============
> HardhatEVM: v2.25.0
> network:    hardhat



  MyOFT Test
    ✔ should send a token from A address to B address via each OFT (67ms)
  MyOFTAdapter Test
    ✔ should send a token from A address to B address via OFTAdapter/OFT (79ms)


  2 passing (397ms)
```

Code Coverage

Perfect coverage metrics across all branches and all lines. Coverage was generated using `npx hardhat coverage.

Changelog

- 2025-07-09 - Initial report
- 2025-07-17 - Final report

About Quantstamp

Quantstamp is a global leader in blockchain security. Founded in 2017, Quantstamp's mission is to securely onboard the next billion users to Web3 through its best-in-class Web3 security products and services.

Quantstamp's team consists of cybersecurity experts hailing from globally recognized organizations including Microsoft, AWS, BMW, Meta, and the Ethereum Foundation. Quantstamp engineers hold PhDs or advanced computer science degrees, with decades of combined experience in formal verification, static analysis, blockchain audits, penetration testing, and original leading-edge research.

To date, Quantstamp has performed more than 500 audits and secured over $200 billion in digital asset risk from hackers. Quantstamp has worked with a diverse range of customers, including startups, category leaders and financial institutions. Brands that Quantstamp has worked with include Ethereum 2.0, Binance, Visa, PayPal, Polygon, Avalanche, Curve, Solana, Compound, Lido, MakerDAO, Arbitrum, OpenSea and the World Economic Forum.

Quantstamp's collaborations and partnerships showcase our commitment to world-class research, development and security. We're honored to work with some of the top names in the industry and proud to secure the future of web3.

Notable Collaborations & Customers:

- Blockchains: Ethereum 2.0, Near, Flow, Avalanche, Solana, Cardano, Binance Smart Chain, Hedera Hashgraph, Tezos
- DeFi: Curve, Compound, Maker, Lido, Polygon, Arbitrum, SushiSwap
- NFT: OpenSea, Parallel, Dapper Labs, Decentraland, Sandbox, Axie Infinity, Illuvium, NBA Top Shot, Zora
- Academic institutions: National University of Singapore, MIT

Timeliness of content

The content contained in the report is current as of the date appearing on the report and is subject to change without notice, unless indicated otherwise by Quantstamp; however, Quantstamp does not guarantee or warrant the accuracy, timeliness, or completeness of any report you access using the internet or other means, and assumes no obligation to update any information following publication or other making available of the report to you by Quantstamp.

Notice of confidentiality

This report, including the content, data, and underlying methodologies, are subject to the confidentiality and feedback provisions in your agreement with Quantstamp. These materials are not to be disclosed, extracted, copied, or distributed except to the extent expressly authorized by Quantstamp.

Links to other websites

You may, through hypertext or other computer links, gain access to web sites operated by persons other than Quantstamp. Such hyperlinks are provided for your reference and convenience only, and are the exclusive responsibility of such web sites' owners. You agree that Quantstamp are not responsible for the content or operation of such web sites, and that Quantstamp shall have no liability to you or any other person or entity for the use of third-party web sites. Except as described below, a hyperlink from this web site to another web site does not imply or mean that Quantstamp endorses the content on that web site or the operator or operations of that site. You are solely responsible for determining the extent to which you may use any content at any other web sites to which you link from the report. Quantstamp assumes no responsibility for the use of third-party software on any website and shall have no liability whatsoever to any person or entity for the accuracy or completeness of any output generated by such software.

Disclaimer

The review and this report are provided on an as-is, where-is, and as-available basis. To the fullest extent permitted by law, Quantstamp disclaims all warranties, expressed implied, in connection with this report, its content, and the related services and products and your use thereof, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement. You agree that access and/or use of the report and other results of the review, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk. FOR AVOIDANCE OF DOUBT, THE REPORT, ITS CONTENT, ACCESS, AND/OR USAGE THEREOF, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER ADVICE. This report is based on the scope of materials and documentation provided for a limited review at the time provided. You acknowledge that Blockchain technology remains under development and is subject to unknown risks and flaws and, as such, the report may not be complete or inclusive of all vulnerabilities. The review is limited to the materials identified in the report and does not extend to the compiler layer, or any other areas beyond the programming language, or programming aspects that could present security risks. The report does not indicate the endorsement by Quantstamp of any particular project or team, nor guarantee its security, and may not be represented as such. No third party is entitled to rely on the report in any way, including for the purpose of making any decisions to buy or sell a product, service or any other asset. Quantstamp does not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party, or any open source or third-party software, code, libraries, materials, or information to, called by, referenced by or accessible through the report, its content, or any related services and products, any hyperlinked websites, or any other websites or mobile applications, and we will not be a party to or in any way be responsible for monitoring any transaction between you and any third party. As with the purchase or use of a product or service through any medium or in any environment, you should use your best judgment and exercise caution where appropriate.

Quantstamp