Bitlayer Bridge
Audit Report

Thu Nov 20 2025

contact@bitslab.xyz https://twitter.com/scalebit_

ScaleBit

Bitlayer Bridge Audit Report

1 Executive Summary

1.1 Project Information

Description	Bitlayer Bridge is the official bridge for Bitlayer, connecting it to multiple blockchains—including Bitcoin, EVM and Non-EVM chains—via various bridges. You can access it here: https://www.bitlayer.org/bridge
Type	Bridge
Auditors	Sprig, Leon@BitsLab
Timeline	Wed Nov 19 2025 - Wed Nov 19 2025
Languages	Move, Solidity
Platform	Sui,EVM Chains
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/bitlayer-org/bitlayer-bridge-evm https://github.com/bitlayer-org/bitlayer-bridge-sui
Commits	https://github.com/bitlayer-org/bitlayer-bridge-evm: 19d75a5c6655270c6dd1f318d744bee3199b0c0a https://github.com/bitlayer-org/bitlayer-bridge-sui: 5fca547e0ca415dc17aa7e5f2edb80dfd999de29

1.2 Files in Scope

The following are the SHA1 hashes of the original reviewed files.

ID	File	SHA-1 Hash
BLI	contracts/BridgeLimiter.sol	2886d6e7b3dcfbe02e5c3e033e673cdd727a8f79
BCO	contracts/BridgeCommittee.sol	237e352c73d65f3e1bfe9bad9049035b7303e9e0
MVE	contracts/utils/MessageVerifier.sol	97853980ae81092c4a2ebdb225f8f017d3f153ab
CUP	contracts/utils/CommitteeUpgradeable.sol	08430c7331a916fb3661f329f0b4d5c501423cc6
BUT	contracts/utils/BridgeUtils.sol	8abe19ea8fea6d030dc788fbfa4c56d2b503943d
BVA	contracts/BridgeVault.sol	d4838a4607d391e97789a675a3d9162474fbd1d6
BRI	contracts/Bridge.sol	7b75054e7dde0e82e25bb0a3580ea2f2e98eb914
MERC2T	contracts/token/MintERC20Token.sol	ae17bfee3a449e7b152dfb901b0d28d1991ec2f6
WNT	contracts/token/WrappedNativeToken.sol	276343d73a50cb158bf875f2aca073622de2d099
BCO1	contracts/BridgeConfig.sol	a7dc44148df14e6449fc2ef6ca60eeb608733d43

IBR	contracts/interfaces/IBridge.sol	c8d2bb455342491c864ecf7f293fba93f3b4e559
IBL	contracts/interfaces/IBridgeLimiter.sol	92eca2caf9ad666d6000351ae14b02af189d6cb3
IERC2M	contracts/interfaces/IERC20Mintable.sol	86e9280f97bac61087de02b2f63546a9fd3f7374
IBC	contracts/interfaces/IBridgeConfig.sol	55c1131736adf3ce1ff05c17a6cedce0501159f0
IBV	contracts/interfaces/IBridgeVault.sol	6380077f22f11b5ab39339bfc0f62d6803ce9bb9
IWBTC9	contracts/interfaces/IWBTC9.sol	5114eac50367110ec3c54d7ace209be527ff471f
IERC2B	contracts/interfaces/IERC20Burnable.sol	d696fd9a2df48f4b3366706ff7ef793e5eeb3779
IBC1	contracts/interfaces/IBridgeCommittee.sol	e770ab73db1408676502f144e0ab7e402978969b
BRI	sources/bridge.move	c589f59cd9b1b3c14d96370b63a479668285f9c3
CID	sources/chain_ids.move	8383021cfe03c1eb60d7259731d0f9c809102005
CRY	sources/crypto.move	c164c2ee7820bbc6b3b2758605e83b6a903c9aee
MTY	sources/message_types.move	6972f0c0fee03220a1d310f8e52e56f5f2881087

MES	sources/message.move	d636351c4a893048bac56d6a64e69aa8c85f4006
COM	sources/committee.move	c7c5038d20c328f78211627f9ad17fbefdb3f12c
TRE	sources/treasury.move	91f1cc355e5311eb5ecd30cc33dad9ac54fefff7
LIM	sources/limiter.move	9289af23713782144cec8d6b4bb9bf0915eb3c1e

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	1	0	1
Centralization	1	0	1
Critical	0	0	0
Major	0	0	0
Medium	0	0	0
Minor	0	0	0
Informational	0	0	0

1.4 ScaleBit Audit Breakdown

ScaleBit aims to assess repositories for security-related issues, code quality, and compliance with specifications and best practices. Possible issues our team looked for included (but are not limited to):

- Transaction-ordering dependence
- Timestamp dependence
- Integer overflow/underflow
- Number of rounding errors
- Unchecked External Call
- Unchecked CALL Return Values
- Functionality Checks
- Reentrancy
- Denial of service / logical oversights
- Access control
- Centralization of power
- Business logic issues
- Gas usage
- Fallback function usage
- tx.origin authentication
- Replay attacks
- Coding style issues

1.5 Methodology

The security team adopted the **"Testing and Automated Analysis"**, **"Code Review"** and **"Formal Verification"** strategy to perform a complete security test on the code in a way that is closest to the real attack. The main entrance and scope of security testing are stated in the conventions in the "Audit Objective", which can expand to contexts beyond the scope according to the actual testing needs. The main types of this security audit include:

(1) Testing and Automated Analysis

Items to check: state consistency / failure rollback / unit testing / value overflows / parameter verification / unhandled errors / boundary checking / coding specifications.

(2) Code Review

The code scope is illustrated in section 1.2.

(3) Audit Process

- Carry out relevant security tests on the testnet or the mainnet;
- If there are any questions during the audit process, communicate with the code owner in time. The code owners should actively cooperate (this might include providing the latest stable source code, relevant deployment scripts or methods, transaction signature scripts, exchange docking schemes, etc.);
- The necessary information during the audit process will be well documented for both the audit team and the code owner in a timely manner.

2 Summary

This report has been commissioned by Bitlayer Labs to identify any potential issues and vulnerabilities in the source code of the Bitlayer Bridge smart contract, as well as any contract dependencies that were not part of an officially recognized library. In this audit, we have utilized various techniques, including manual code review and static analysis, to identify potential vulnerabilities and security issues.

During the audit, we identified 1 issues of varying severity, listed below.

ID	Title	Severity	Status
BCO-1	Centralization Risk	Centralization	Acknowledged

3 Participant Process

Here are the relevant actors with their respective abilities within the Bitlayer Bridge Smart Contract :

Admin

- The admin can add one or more tokens through `addToken()` and `addTokens()` .
- The admin can update the token state through `updateTokenState()` .
- The admin can update the `wbtc9` address through `setWBTC9()` .
- The admin can update the chainID state through `updateChainID()` .
- The admin can update the `tokenFeePercentages` through `setFeePercentage()` .
- The admin can update the min amount through `setTokenMinAmount()` .
- The admin can add the stake amount of committees through `addCommitteeStake()` .
- The admin can update the submitter list through `updateSubmitterlist()` .
- The admin can pause/unpause the bridge with signatures through `executeEmergencyOp()` .
- The admin can update the block state of committees with signatures through `updateBlocklist()` .
- The admin can update the token price through `updateTokenPrice()` .
- The admin can add tokens with signatures through `addTokens()` .
- The admin can update the limit with signatures through `updateLimit()` .
- The admin can update the block list through `updateBlocklist()` .
- The admin can transfer tokens with signatures through `transferBridgedTokens()` .
- The admin can update the mininum through `setTokenMinAmount()` .
- The admin can update the fee percentage through `setFeePercentage()` .
- The admin can withdraw assets through `withdraw_treasury()` .

Users

- Users can bridge native tokens or ERC20 tokens through `bridgeNativeToken()` and `bridgeERC20()` .

4 Findings

BCO-1 Centralization Risk

Severity: Centralization

Status: Acknowledged

Code Location:

contracts/BridgeCommittee.sol

Descriptions:

Centralization risk is identified in the smart contract:

- The admin can add one or more tokens through `addToken()` and `addTokens()`.
- The admin can update the token state through `updateTokenState()`.
- The admin can update the `wbtc9` address through `setWBTC9()`.
- The admin can update the chainID state through `updateChainID()`.
- The admin can update the `tokenFeePercentages` through `setFeePercentage()`.
- The admin can update the min amount through `setTokenMinAmount()`.
- The admin can add the stake amount of committees through `addCommitteeStake()`.
- The admin can update the submitter list through `updateSubmitterlist()`.
- The admin can update the config address through `initializeConfig()`.
- The admin can pause/unpause the bridge with signatures through `executeEmergencyOp()`.
- The admin can update the block state of committees with signatures through `updateBlocklist()`.
- The admin can update the token price through `updateTokenPrice()`.
- The admin can add tokens with signatures through `addTokens()`.

- The admin can update the limit with signatures through `updateLimit()`.
- The admin can update the block list through `updateBlocklist()`.
- The admin can transfer tokens with signatures through `transferBridgedTokens()`.
- The admin can update the mininum through `setTokenMinAmount()`.
- The admin can update the fee percentage through `setFeePercentage()`.
- The admin can withdraw assets through `withdraw_treasury()`.

Suggestion:

It is recommended that measures be taken to reduce the risk of centralization, such as a multi-signature mechanism.

Appendix 1

Issue Level

- **Informational** issues are often recommendations to improve the style of the code or to optimize code that does not affect the overall functionality.
- **Minor** issues are general suggestions relevant to best practices and readability. They don't post any direct risk. Developers are encouraged to fix them.
- **Medium** issues are non-exploitable problems and not security vulnerabilities. They should be fixed unless there is a specific reason not to.
- **Major** issues are security vulnerabilities. They put a portion of users' sensitive information at risk, and often are not directly exploitable. All major issues should be fixed.
- **Critical** issues are directly exploitable security vulnerabilities. They put users' sensitive information at risk. All critical issues should be fixed.

Issue Status

- **Fixed:** The issue has been resolved.
- **Partially Fixed:** The issue has been partially resolved.
- **Acknowledged:** The issue has been acknowledged by the code owner, and the code owner confirms it's as designed, and decides to keep it.

Appendix 2

Disclaimer

This report is based on the scope of materials and documents provided, with a limited review at the time provided. Results may not be complete and do not include all vulnerabilities. The review and this report are provided on an as-is, where-is, and as-available basis. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your own risk. A report does not imply an endorsement of any particular project or team, nor does it guarantee its security. These reports should not be relied upon in any way by any third party, including for the purpose of making any decision to buy or sell products, services, or any other assets. TO THE FULLEST EXTENT PERMITTED BY LAW, WE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, IN CONNECTION WITH THIS REPORT, ITS CONTENT, RELATED SERVICES AND PRODUCTS, AND YOUR USE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NOT INFRINGEMENT.