Quantstamp Security Assessment Certificate

December 22nd 2021 — Quantstamp Verified

LayerZero

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

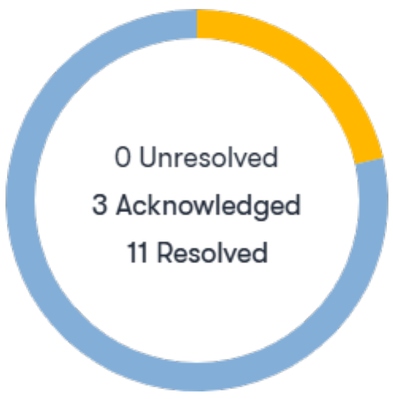
Executive Summary

Type	Omnichain interoperability protocol
Auditors	Souhail Mssassi, Research Engineer Hisham Galal, Research Engineer Cristiano Silva, Research Engineer
Timeline	2021-11-23 through 2021-11-28
EVM	Muir Glacier
Languages	Solidity
Methods	Architecture Review, Unit Testing, Functional Testing, Computer-Aided Verification, Manual Review
Specification	None
Documentation Quality	Undetermined
Test Quality	Undetermined

Source Code

Repository	Commit
stargate	22769fa

Total Issues	**14** (11 Resolved)
High Risk Issues	**3** (3 Resolved)
Medium Risk Issues	**3** (3 Resolved)
Low Risk Issues	**5** (3 Resolved)
Informational Risk Issues	**3** (2 Resolved)
Undetermined Risk Issues	0 (0 Resolved)



High Risk	The issue puts a large number of users' sensitive information at risk, or is reasonably likely to lead to catastrophic impact for client's reputation or serious financial implications for client and users.
Medium Risk	The issue puts a subset of users' sensitive information at risk, would be detrimental for the client's reputation if exploited, or is reasonably likely to lead to moderate financial impact.
Low Risk	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 post an immediate risk, but is relevant to security best practices or Defence in Depth.
Undetermined	The impact of the issue is uncertain.

Unresolved	Acknowledged the existence of the risk, and decided to accept it without engaging in special efforts to control it.
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).
Resolved	Adjusted program implementation, requirements or constraints to eliminate the risk.
Mitigated	Implemented actions to minimize the impact or likelihood of the risk.

Summary of Findings

ID	Description	Severity	Status
QSP-1	Usage Of transfer/transferFrom Instead Of safeTransfer/safeTransferFrom	High	Fixed
QSP-2	Front-Run on the setRouter	High	Fixed
QSP-3	Setters without Any Restrictions	High	Fixed
QSP-4	Approve Race Condition	Medium	Fixed
QSP-5	Owner Can Create Duplicate Pools	Medium	Fixed
QSP-6	Reward Miscalculation	Medium	Fixed
QSP-7	Owner Can Renounce Ownership	Low	Fixed
QSP-8	Missing Address Verification	Low	Fixed
QSP-9	Missing Value Verification	Low	Fixed
QSP-10	Gas Usage / `for` Loop Concerns	Low	Acknowledged
QSP-11	Block Timestamp Manipulation	Low	Acknowledged
QSP-12	CachedSwapLookup Cleared By Any User	Informational	Acknowledged
QSP-13	Unlocked Pragma	Informational	Fixed
QSP-14	'Dead' Code	Informational	Fixed

Quantstamp Audit Breakdown

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

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

The Quantstamp auditing process follows a routine series of steps:

1. Code review that includes the following
   i. Review of the specifications, sources, and instructions provided to Quantstamp to make sure we understand the size, scope, and functionality of the smart contract.
   ii. Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
   iii. 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:
   i. 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.
   ii. 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, clarify, 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.

Toolset

The notes below outline the setup and steps performed in the process of this audit.

Setup

Tool Setup:

- Slither v0.6.6
- Mythril v0.2.7

Steps taken to run the tools:

Installed the Slither tool: `pip install slither-analyzer` Run Slither from the project directory: `slither .` Installed the Mythril tool from Pypi: `pip3 install mythril` Ran the Mythril tool on each contract: `myth -x path/to/contract`

Findings

QSP-1 Usage Of transfer/transferFrom Instead Of safeTransfer/safeTransferFrom

Severity: *High Risk*

Status: Fixed

File(s) affected: `Staking.sol`, `LPStaking.sol`

Description: The ERC20 standard token implementation functions also return the transaction status as a Boolean. It's good practice to check for the return status of the function call to ensure that the transaction was successful. It is the developer's responsibility to enclose these function calls with require() to ensure that, when the intended ERC20 function call returns false, the caller transaction also fails. However, it is mostly missed by developers when they carry out checks; in effect, the transaction would always succeed, even if the token `transfer`/`transferFrom` didn't.

- The issue is in the `enter` function in the Staking contract (L40).
- Same issue in the `leave` function in the Staking contract (L51).
- Same issue in the `safeStargateTransfer` function in the LPStaking contract (L247,L249).

Recommendation: Use the `safeTransfer`/`safeTransferFrom` function from the `safeERC20` Implementation or put the transfer call inside and assert or require to verify that it returned true.

QSP-2 Front-Run on the setRouter

Severity: *High Risk*

Status: Fixed

File(s) affected: `Factory.sol`, `Bridge.sol`

Description: In the `setRouter` function located in the Factory contract , the `router` variable is set for one time . The problem here is that no restriction is done in order to prevent other users to call this function and update the `router` address.

- The issue is located in `setRouter` function of the Factory contract (L29).
- Same issue in the Bridge contract (L55).

Recommendation: Add a require to verify that the only the Owner who have the right to call this function.

QSP-3 Setters without Any Restrictions

Severity: *High Risk*

Status: Fixed

File(s) affected: `Bridge.sol`

Description: In the `Bridge` contract, we have setters that allow us to update the Oracle, Relayer address same thing with the `setBlockConfirmations` and `setLibraryVersion` functions, the problem here is that these functions don't have any modifiers thus any user can call this function and update the variables.

- The issue is located in the Bridge contract.

Update: Add a require to verify that the only the Owner who have the right to call these functions.

QSP-4 Approve Race Condition

Severity: *Medium Risk*

Status: Fixed

File(s) affected: `LPTokenERC20.sol`,

Description: The standard ERC20 implementation contains a widely-known racing condition in its approve function, wherein a spender is able to witness the token owner broadcast a transaction altering their approval and quickly sign and broadcast a transaction using `transferFrom` to move the current approved amount from the owner's balance to the spender. If the spender's transaction is validated before the owner's, the spender will be able to get both approval amounts of both transactions.

- The issue is located in `LPTokenERC20` contract (L65).

Recommendation: Use `increaseAllowance` and `decreaseAllowance` functions to modify the approval amount instead of using the `approve` function to modify it.

QSP-5 Owner Can Create Duplicate Pools

Severity: *Medium Risk*

Status: Fixed

File(s) affected: `LPStaking.sol`, `Factory.sol`

Description: The add() function is used to add a new pool, it turns out that it did not complete essential sanity checks to prohibit the creation of a new pool with duplicate LP tokens. If a new pool with a duplicate LP token is introduced, it is likely that an error in the reward distribution to the pools and staking will occur.

- The issue is located in the `add` function in LPStaking contract (L86).
- Same issue in the `createPair` function in the Factory contract (L42).

Recommendation: This might be avoided by defining a mapping from addresses to Booleans, such that once added, LP tokens are mapped to true. A require-statement might then be added to the method to prevent the same LP token from being added again.

QSP-6 Reward Miscalculation

Severity: *Medium Risk*

Status: Fixed

File(s) affected: `LPStaking.sol`

Description: The `totalAllocPoint` variable is used to determine the portion of total rewards minted that each pool will get, making it a critical part in the rewards calculation. As a result, if the `totalAllocPoint` variable is changed without first updating the pending awards, the payout for each pool is calculated improperly. The following `add()`and `set` functions modify the `totalAllocPoint` variable without updating the awards.

- The issue is located in the `add` and `set` function of the LPStaking contract (L91,L113).

Recommendation: Remove _withUpdate variable in the add() functions and always call the massUpdatePools() function before updating totalAllocPoint variable.

QSP-7 Owner Can Renounce Ownership

Severity: *Low Risk*

Status: Fixed

File(s) affected: `StargateToken.sol`, `Router.sol`, `Pool.sol`, `LPStaking.sol`, `Bridge.sol`

Description: Typically, the contract's owner is the account that deploys the contract. As a result, the owner is able to perform certain privileged activities on his behalf. The renounceOwnership function is used in smart contracts to renounce ownership. Otherwise, if the contract's ownership has not been transferred previously, it will never have an Owner, which is risky.

Recommendation: It is advised that the Owner cannot call renounceOwnership without first transferring ownership to a different address. Additionally, if a multi-signature wallet is utilized, executing the renounceOwnership method for two or more users should be confirmed. Alternatively, the RenounceOwnership functionality can be disabled by overriding it.

QSP-8 Missing Address Verification

Severity: *Low Risk*

Status: Fixed

File(s) affected: `Staking.sol`, `Router.sol`, `Pool.sol`, `LPStaking.sol`, `Bridge.sol`

Description: Certain functions lack a safety check in the address, the address-type argument should include a zero-address test, otherwise, the contract's functionality may become inaccessible or tokens may be burned in perpetuity.

- The `_stargate`variable in the constructor of of the Staking contract (L18).
- The `_bridge` and `factory` variables in the `setBridgeAndFactory` function of the Router contract (L63).
- The `_to` variable in the `addLiquidity` function of the Router contract (L91).
- The `_to` variable in the `removeLiquidityAndRedeemOnRemote` function of the Router contract (L106).
- The `_to` variable in the `removeLiquidity` function of the Router contract (L114).
- The `_refundAddress` variable in the `sendCredits` function of the Router contract (L121).
- The `_refundAddress` variable in the `retryRevert` function of the Router contract (L128).
- The `_srcAddress` variable in the `clearCachedSwap`function of the Router contract (L150).
- The `_to` and `_srcAddress`variables in the `removeLiquidityRemote` function of the Router contract (L172).
- The `_to` and `_srcAddress`variables in the `_swapAndMint` function of the Router contract (L188).
- The `_to` and `_srcAddress`variables in the `_swapRemote` function of the Router contract (L201).
- The `_token` variable in the `createPair` function of the Router contract (L227).
- The `_owner` variable in the `setXStargateFeeOwner` function of the Router contract (L241).
- The `_owner` variable in the `setMintFeeOwner` function of the Router contract (L245).
- The `_router` and `_token` variables in the constructor of the Pool contract (L79).
- The `_from` variable located in the `burnLocalAndRedeemOnRemote` function of the Pool contract (L236).
- The `_from` variable located in the `burnAndRedeem` function of the Pool contract (L248).
- The `_to` variable located in the `swapRemote` function of the Pool contract (L267).
- The `_to` variable located in the `swapAndMint` function of the Pool contract (L285).
- The `_devaddr` and `_stargate` variables located in the constructor of LPStaking contract (L66).
- The `_devaddr` variable located in the `dev` function of LPStaking contract (L254).
- The `_layerZeroEndpoint` located in the constructor of the Bridge contract (L51).

Recommendation: It's recommended to undertake further validation prior to user-supplied data. The concerns can be resolved by utilizing a whitelist technique or a modifier.

QSP-9 Missing Value Verification

Severity: *Low Risk*

Status: Fixed

File(s) affected: `Staking.sol`, `LPStaking.sol`

Description: Certain functions lack a safety check in the values, the values of the arguments should include some safety checks test, otherwise, the contract's functionality may get hurt.

- The `_amount` in the `enter` function of the staking contract should be greater than 0 (L23).
  . The `_share` in the `leave` function of the staking contract should be greater than 0 (L45).

- The `_startBlock` and `_bonusEndBlock` variables located in the constructor of LPStaking contract (L66).

Recommendation: It's recommended to undertake further validation prior to user-supplied data. The concerns can be resolved by utilizing require statements.

QSP-10 Gas Usage / `for` Loop Concerns

Severity: *Low Risk*

Status: Acknowledged

File(s) affected: `Pool.sol`, `LPStaking.sol`

Description: Gas usage is a main concern for smart contract developers and users, since high gas costs may prevent users from wanting to use the smart contract. Even worse, some gas usage issues may prevent the contract from providing services entirely. For example, if a `for` loop requires too much gas to exit, then it may prevent the contract from functioning correctly entirely. It is best to break such loops into individual functions as possible.

- The issue is located in the `_distributeFunds` function in the Pool contract (L110,L131,L139).
- Same issue located in the `createChainPath` function in the Pool contract (L318).
- Same issue located in the `setWeightForChainPath` function in the Pool contract (L330).
- Same issue located in the `massUpdatePools` function in the LPStaking contract (L163).

Recommendation: Avoid actions that involve looping across the entire data structure. If you really must loop over an array of unknown size, arrange for it to consume many blocs and thus multiple transactions.

QSP-11 Block Timestamp Manipulation

Severity: *Low Risk*

Status: Acknowledged

File(s) affected: `LPTokenERC20.sol`

Description: Projects may rely on block timestamps for various purposes. However, it's important to realize that miners individually set the timestamp of a block, and attackers may be able to manipulate timestamps for their own purposes. If a smart contract relies on a timestamp, it must take this into account.

- The issue is located in the `permit` function of the `LPTokenERC20` contract (L84).

Recommendation: Verify if a delay of 900 seconds won't destroy the logic of the staking contract.

QSP-12 CachedSwapLookup Cleared By Any User

Severity: *Informational*

Status: Acknowledged

File(s) affected: `Router.sol`

Description: In the `clearCachedSwap` function, we are clearing the mapping `cachedSwapLookup` to the default state, this mapping in fact will be set in the catch block of the `_swapRemote` function thus any one can reset this value after the call .

- Issue located in the `clearCachedSwap` function (L162).

Recommendation: Add a modifier to prevent some on to call this function.

QSP-13 Unlocked Pragma

Severity: *Informational*

Status: Fixed

File(s) affected: `interfaces/*`

Description: Every Solidity file specifies in the header a version number of the format `pragma solidity (^)0.7.6`. The caret (^) before the version number implies an unlocked pragma, meaning that the compiler will use the specified version *and above*, hence the term "unlocked".

Recommendation: For consistency and to prevent unexpected behavior in the future, it is recommended to remove the caret to lock the file onto a specific Solidity version.

QSP-14 'Dead' Code

Severity: Informational

Status: Fixed

File(s) affected: `Pool.sol`, `Bridge.sol`

Description: "Dead" code refers to code whose execution makes no impact on the final result. Dead code raises a concern, since either the code is unnecessary or the necessary code's results were ignored. Regardless, further investigation is required.

- The issue is located in the Pool contract (L198,L199,203,204).
- Same issue in the Bridge contract (L41-49).

Automated Analyses

Slither

LPStaking.safeStargateTransfer(address,uint256) (contracts/LPStaking.sol#261-268) ignores return value by stargate.transfer(_to,stargateBal) (contracts/LPStaking.sol#264) LPStaking.safeStargateTransfer(address,uint256) (contracts/LPStaking.sol#261-268) ignores return value by stargate.transfer(_to,_amount) (contracts/LPStaking.sol#266) Staking.enter(uint256) (contracts/Staking.sol#26-44) ignores return value by stargate.transferFrom(msg.sender,address(this),_amount) (contracts/Staking.sol#43) Staking.leave(uint256) (contracts/Staking.sol#48-57) ignores return value by stargate.transfer(msg.sender,what) (contracts/Staking.sol#56) Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unchecked-transfer

LPStaking.pendingStargate(uint256,address) (contracts/LPStaking.sol#150-174) performs a multiplication on the result of a division: -stargateReward = multiplier.mul(stargatePerBlock).mul(pool.allocPoint).div(totalAllocPoint) (contracts/LPStaking.sol#164-167) -accStargatePerShare = accStargatePerShare.add(stargateReward.mul(1e12).div(lpSupply)) (contracts/LPStaking.sol#168-170) LPStaking.updatePool(uint256) (contracts/LPStaking.sol#186-207) performs a multiplication on the result of a division: -stargateReward = multiplier.mul(stargatePerBlock).mul(pool.allocPoint).div(totalAllocPoint) (contracts/LPStaking.sol#197-200) -pool.accStargatePerShare = pool.accStargatePerShare.add(stargateReward.mul(1e12).div(lpSupply)) (contracts/LPStaking.sol#203-205) BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) performs a multiplication on the result of a division: -sstore(uint256,uint256)(_preBytes,fslot_concatStorage_asm_0 + mload(uint256)(_postBytes + 0x20) / 0x100 ** 32 - mlength_concatStorage_asm_0 * 0x100 ** 32 - newlength_concatStorage_asm_0 + mlength_concatStorage_asm_0 * 2) (contracts/libraries/BytesLib.sol#115-140) BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) performs a multiplication on the result of a division: -sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0) / mask_concatStorage_asm_0 * mask_concatStorage_asm_0) (contracts/libraries/BytesLib.sol#189) BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) performs a multiplication on the result of a division: -sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0) / mask_concatStorage_asm_0 * mask_concatStorage_asm_0) (contracts/libraries/BytesLib.sol#223) BytesLib.equalStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#446-516) performs a multiplication on the result of a division: -fslot_equalStorage_asm_0 = fslot_equalStorage_asm_0 / 0x100 * 0x100 (contracts/libraries/BytesLib.sol#473) Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply

LPStaking.updatePool(uint256) (contracts/LPStaking.sol#186-207) uses a dangerous strict equality: - lpSupply == 0 (contracts/LPStaking.sol#192) Staking.enter(uint256) (contracts/Staking.sol#26-44) uses a dangerous strict equality: - totalShares == 0 || totalStargate == 0 (contracts/Staking.sol#32) Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities

Reentrancy in LPStaking.add(uint256,IERC20,bool) (contracts/LPStaking.sol#90-113): External calls: - massUpdatePools() (contracts/LPStaking.sol#99) - stargate.mint(devaddr,stargateReward.div(10)) (contracts/LPStaking.sol#201) - stargate.mint(address(this),stargateReward) (contracts/LPStaking.sol#202) State variables written after the call(s): - poolInfo.push(PoolInfo(_lpToken,_allocPoint,lastRewardBlock,0)) (contracts/LPStaking.sol#105-112) - totalAllocPoint = totalAllocPoint.add(_allocPoint) (contracts/LPStaking.sol#104) Reentrancy in Router.clearCachedSwap(uint16,bytes,uint256) (contracts/Router.sol#337-361): External calls: - IStargateReceiver(cs.to).sgReceive(_chainId,_srcAddress,_nonce,cs.token,cs.amountLD,cs.payload) (contracts/Router.sol#345-352) State variables written after the call(s): - cachedSwapLookup[_chainId][_srcAddress][_nonce] = CachedSwap(address(0x0),0,address(0x0),) (contracts/Router.sol#355-360) Reentrancy in LPStaking.deposit(uint256,uint256) (contracts/LPStaking.sol#210-230): External calls: - updatePool(_pid) (contracts/LPStaking.sol#213) - stargate.mint(devaddr,stargateReward.div(10)) (contracts/LPStaking.sol#201) - stargate.mint(address(this),stargateReward) (contracts/LPStaking.sol#202) - safeStargateTransfer(msg.sender,pending) (contracts/LPStaking.sol#220) - stargate.transfer(_to,stargateBal) (contracts/LPStaking.sol#264) - stargate.transfer(_to,_amount) (contracts/LPStaking.sol#266) - pool.lpToken.safeTransferFrom(address(msg.sender),address(this),_amount) (contracts/LPStaking.sol#222-226) State variables written after the call(s): - user.amount = user.amount.add(_amount) (contracts/LPStaking.sol#227) - user.rewardDebt = user.amount.mul(pool.accStargatePerShare).div(1e12) (contracts/LPStaking.sol#228) Reentrancy in LPStaking.emergencyWithdraw(uint256) (contracts/LPStaking.sol#251-258): External calls: - pool.lpToken.safeTransfer(address(msg.sender),user.amount) (contracts/LPStaking.sol#254) State variables written after the call(s): - user.amount = 0 (contracts/LPStaking.sol#256) - user.rewardDebt = 0 (contracts/LPStaking.sol#257) Reentrancy in LPStaking.set(uint256,uint256,bool) (contracts/LPStaking.sol#116-129): External calls: - massUpdatePools() (contracts/LPStaking.sol#123) - stargate.mint(devaddr,stargateReward.div(10)) (contracts/LPStaking.sol#201) - stargate.mint(address(this),stargateReward) (contracts/LPStaking.sol#202) State variables written after the call(s): - poolInfo[_pid].allocPoint = _allocPoint (contracts/LPStaking.sol#128) - totalAllocPoint = totalAllocPoint.sub(poolInfo[_pid].allocPoint).add(_allocPoint) (contracts/LPStaking.sol#125-127) Reentrancy in LPStaking.updatePool(uint256) (contracts/LPStaking.sol#186-207): External calls: - stargate.mint(devaddr,stargateReward.div(10)) (contracts/LPStaking.sol#201) - stargate.mint(address(this),stargateReward) (contracts/LPStaking.sol#202) State variables written after the call(s): - pool.accStargatePerShare = pool.accStargatePerShare.add(stargateReward.mul(1e12).div(lpSupply)) (contracts/LPStaking.sol#203-205) - pool.lastRewardBlock = block.number (contracts/LPStaking.sol#206) Reentrancy in LPStaking.withdraw(uint256,uint256) (contracts/LPStaking.sol#233-248): External calls: - updatePool(_pid) (contracts/LPStaking.sol#237) - stargate.mint(devaddr,stargateReward.div(10)) (contracts/LPStaking.sol#201) - stargate.mint(address(this),stargateReward) (contracts/LPStaking.sol#202) - safeStargateTransfer(msg.sender,pending) (contracts/LPStaking.sol#243) - stargate.transfer(_to,stargateBal) (contracts/LPStaking.sol#264) - stargate.transfer(_to,_amount) (contracts/LPStaking.sol#266) State variables written after the call(s): - user.amount = user.amount.sub(_amount) (contracts/LPStaking.sol#244) - user.rewardDebt = user.amount.mul(pool.accStargatePerShare).div(1e12) (contracts/LPStaking.sol#245) Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1

Router._swapRemote(uint16,bytes,uint256,uint256,uint256,address,Pool.SwapObj,bytes).amountLD (contracts/Router.sol#500) is a local variable never initialized Router._swapRemote(uint16,bytes,uint256,uint256,uint256,address,Pool.SwapObj,bytes).reason (contracts/Router.sol#512) is a local variable never initialized Router.removeLiquidityRemote(uint16,bytes,uint256,uint256,uint256,bytes).swapAmountSD (contracts/Router.sol#385) is a local variable never initialized Router.removeLiquidityRemote(uint16,bytes,uint256,uint256,uint256,bytes).mintAmountSD (contracts/Router.sol#386) is a local variable never initialized Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

Bridge.quoteLayerZeroFee(uint16,uint8,bytes,bytes) (contracts/Bridge.sol#316-348) ignores return value by ILayerZeroEndpoint(layerZeroEndpoint).estimateNativeFees(_chainId,address(this),payload,useLayerZeroToken,gasAmountBuilderV1(gasLookup[_chainId][_functionType])) (contracts/Bridge.sol#341-347) Bridge.approveTokenSpender(address,address,uint256) (contracts/Bridge.sol#377-383) ignores return value by

IERC20(token).approve(spender,amount) (contracts/Bridge.sol#382) Router.addLiquidity(uint256,uint256,address) (contracts/Router.sol#136-144) ignores return value by pool.mint(_to,_amountLD) (contracts/Router.sol#143) Router.removeLiquidityRemote(uint16,bytes,uint256,uint256,uint256,bytes) (contracts/Router.sol#374-416) ignores return value by pool.withdrawRemote(_chainId,_amountSD) (contracts/Router.sol#384-415) Router._swapRemote(uint16,bytes,uint256,uint256,uint256,address,Pool.SwapObj,bytes) (contracts/Router.sol#488-551) ignores return value by pool.swapRemote(_to,_s) (contracts/Router.sol#500-550) Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Router.setXStargateFeeOwner(address)._owner (contracts/Router.sol#591) shadows: - Ownable._owner (node_modules/@openzeppelin/contracts/access/Ownable.sol#19) (state variable) Router.setMintFeeOwner(address)._owner (contracts/Router.sol#596) shadows: - Ownable._owner (node_modules/@openzeppelin/contracts/access/Ownable.sol#19) (state variable) Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

LPStaking.add(uint256,IERC20,bool) (contracts/LPStaking.sol#90-113) should emit an event for: - totalAllocPoint = totalAllocPoint.add(_allocPoint) (contracts/LPStaking.sol#104) LPStaking.set(uint256,uint256,bool) (contracts/LPStaking.sol#116-129) should emit an event for: - totalAllocPoint = totalAllocPoint.sub(poolInfo[_pid].allocPoint).add(_allocPoint) (contracts/LPStaking.sol#125-127) Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-arithmetic

Bridge.constructor(address)._layerZeroEndpoint (contracts/Bridge.sol#58) lacks a zero-check on : - layerZeroEndpoint = _layerZeroEndpoint (contracts/Bridge.sol#60) Factory.setRouter(address)._router (contracts/Factory.sol#29) lacks a zero-check on : - router = _router (contracts/Factory.sol#32) LPStaking.constructor(StargateToken,address,uint256,uint256,uint256)._devaddr (contracts/LPStaking.sol#70) lacks a zero-check on : - devaddr = _devaddr (contracts/LPStaking.sol#78) LPStaking.dev(address)._devaddr (contracts/LPStaking.sol#271) lacks a zero-check on : - devaddr = _devaddr (contracts/LPStaking.sol#274) Pool.constructor(uint256,address,address,uint256,uint256)._router (contracts/Pool.sol#128) lacks a zero-check on : - router = _router (contracts/Pool.sol#136) Pool.constructor(uint256,address,address,uint256,uint256)._token (contracts/Pool.sol#129) lacks a zero-check on : - token = _token (contracts/Pool.sol#137) Router.setXStargateFeeOwner(address)._owner (contracts/Router.sol#591) lacks a zero-check on : - xStargateFeeOwner = _owner (contracts/Router.sol#593) Router.setMintFeeOwner(address)._owner (contracts/Router.sol#596) lacks a zero-check on : - mintFeeOwner = _owner (contracts/Router.sol#598) Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Variable 'Bridge.lzReceive(uint16,bytes,uint64,bytes).poolId (contracts/Bridge.sol#129)' in Bridge.lzReceive(uint16,bytes,uint64,bytes) (contracts/Bridge.sol#107-212) potentially used before declaration: (poolId,creditAmountSD) = abi.decode(_payload,(uint8,uint256,uint256)) (contracts/Bridge.sol#163-166) Variable 'Bridge.lzReceive(uint16,bytes,uint64,bytes).creditAmountSD (contracts/Bridge.sol#131)' in Bridge.lzReceive(uint16,bytes,uint64,bytes) (contracts/Bridge.sol#107-212) potentially used before declaration: (poolId,creditAmountSD) = abi.decode(_payload,(uint8,uint256,uint256)) (contracts/Bridge.sol#163-166) Variable 'Bridge.lzReceive(uint16,bytes,uint64,bytes).to (contracts/Bridge.sol#133)' in Bridge.lzReceive(uint16,bytes,uint64,bytes) (contracts/Bridge.sol#107-212) potentially used before declaration: (poolId,creditAmountSD,amountSD,mintAmountSD,to) = abi.decode(_payload,(uint8,uint256,uint256,uint256,uint256,bytes)) (contracts/Bridge.sol#169-179) Variable 'Bridge.lzReceive(uint16,bytes,uint64,bytes).creditAmountSD (contracts/Bridge.sol#131)' in Bridge.lzReceive(uint16,bytes,uint64,bytes) (contracts/Bridge.sol#107-212) potentially used before declaration: (poolId,creditAmountSD,amountSD,mintAmountSD,to) = abi.decode(_payload,(uint8,uint256,uint256,uint256,uint256,bytes)) (contracts/Bridge.sol#169-179) Variable 'Bridge.lzReceive(uint16,bytes,uint64,bytes).poolId (contracts/Bridge.sol#129)' in Bridge.lzReceive(uint16,bytes,uint64,bytes) (contracts/Bridge.sol#107-212) potentially used before declaration: (poolId,creditAmountSD,amountSD,mintAmountSD,to) = abi.decode(_payload,(uint8,uint256,uint256,uint256,uint256,bytes)) (contracts/Bridge.sol#169-179) Variable 'Bridge.lzReceive(uint16,bytes,uint64,bytes).to (contracts/Bridge.sol#133)' in Bridge.lzReceive(uint16,bytes,uint64,bytes) (contracts/Bridge.sol#107-212) potentially used before declaration: toAddress = mload(uint256)(to + 20) (contracts/Bridge.sol#182) Variable 'Bridge.lzReceive(uint16,bytes,uint64,bytes).toAddress (contracts/Bridge.sol#147)' in Bridge.lzReceive(uint16,bytes,uint64,bytes) (contracts/Bridge.sol#107-212) potentially used before declaration: toAddress = mload(uint256)(to + 20) (contracts/Bridge.sol#182) Variable 'Bridge.lzReceive(uint16,bytes,uint64,bytes).to (contracts/Bridge.sol#133)' in Bridge.lzReceive(uint16,bytes,uint64,bytes) (contracts/Bridge.sol#107-212) potentially used before declaration: (poolId,creditAmountSD,amountSD,to) = abi.decode(_payload,(uint8,uint256,uint256,uint256,bytes)) (contracts/Bridge.sol#195-201) Variable 'Bridge.lzReceive(uint16,bytes,uint64,bytes).creditAmountSD (contracts/Bridge.sol#131)' in Bridge.lzReceive(uint16,bytes,uint64,bytes) (contracts/Bridge.sol#107-212) potentially used before declaration: (poolId,creditAmountSD,amountSD,to) = abi.decode(_payload,(uint8,uint256,uint256,uint256,bytes)) (contracts/Bridge.sol#195-201) Variable 'Bridge.lzReceive(uint16,bytes,uint64,bytes).poolId (contracts/Bridge.sol#129)' in Bridge.lzReceive(uint16,bytes,uint64,bytes) (contracts/Bridge.sol#107-212) potentially used before declaration: (poolId,creditAmountSD,amountSD,to) = abi.decode(_payload,(uint8,uint256,uint256,uint256,bytes)) (contracts/Bridge.sol#195-201) Variable 'Bridge.lzReceive(uint16,bytes,uint64,bytes).amountSD (contracts/Bridge.sol#173)' in Bridge.lzReceive(uint16,bytes,uint64,bytes) (contracts/Bridge.sol#107-212) potentially used before declaration: (poolId,creditAmountSD,amountSD,to) = abi.decode(_payload,(uint8,uint256,uint256,uint256,bytes)) (contracts/Bridge.sol#195-201) Variable 'Router.retryRevert(uint16,bytes,uint256,address).poolId (contracts/Router.sol#276)' in Router.retryRevert(uint16,bytes,uint256,address) (contracts/Router.sol#255-335) potentially used before declaration: (poolId,to,amountSD,mintAmountSD) = abi.decode(payload,(uint8,uint256,address,uint256,uint256)) (contracts/Router.sol#293-302) Variable 'Router.retryRevert(uint16,bytes,uint256,address).to (contracts/Router.sol#279)' in Router.retryRevert(uint16,bytes,uint256,address) (contracts/Router.sol#255-335) potentially used before declaration: (poolId,to,amountSD,mintAmountSD) = abi.decode(payload,(uint8,uint256,address,uint256,uint256)) (contracts/Router.sol#293-302) Variable 'Router.retryRevert(uint16,bytes,uint256,address).amountSD (contracts/Router.sol#277)' in Router.retryRevert(uint16,bytes,uint256,address) (contracts/Router.sol#255-335) potentially used before declaration: (poolId,to,amountSD,mintAmountSD) = abi.decode(payload,(uint8,uint256,address,uint256,uint256)) (contracts/Router.sol#293-302) Variable 'Router.retryRevert(uint16,bytes,uint256,address).mintAmountSD (contracts/Router.sol#278)' in Router.retryRevert(uint16,bytes,uint256,address) (contracts/Router.sol#255-335) potentially used before declaration: (poolId,to,amountSD,mintAmountSD) = abi.decode(payload,(uint8,uint256,address,uint256,uint256)) (contracts/Router.sol#293-302) Variable 'Router.retryRevert(uint16,bytes,uint256,address).poolId (contracts/Router.sol#276)' in Router.retryRevert(uint16,bytes,uint256,address) (contracts/Router.sol#255-335) potentially used before declaration: (poolId,dstGasForCall,to,s,p) = abi.decode(payload,(uint8,uint256,uint256,address,Pool.SwapObj,bytes)) (contracts/Router.sol#313-323) Variable 'Router.retryRevert(uint16,bytes,uint256,address).to (contracts/Router.sol#279)' in Router.retryRevert(uint16,bytes,uint256,address) (contracts/Router.sol#255-335) potentially used before declaration: (poolId,dstGasForCall,to,s,p) = abi.decode(payload,(uint8,uint256,uint256,address,Pool.SwapObj,bytes)) (contracts/Router.sol#313-323) Variable 'Router.removeLiquidityRemote(uint16,bytes,uint256,uint256,uint256,bytes).swapAmountSD (contracts/Router.sol#385)' in Router.removeLiquidityRemote(uint16,bytes,uint256,uint256,uint256,bytes) (contracts/Router.sol#374-416) potentially used before declaration: revertLookup[_chainId][_srcAddress][_nonce] = abi.encode(TYPE_SWAP_AND_MINT_REMOTE,_poolId,swapAmountSD,mintAmountSD,_to) (contracts/Router.sol#388-394) Variable 'Router.removeLiquidityRemote(uint16,bytes,uint256,uint256,uint256,bytes).mintAmountSD (contracts/Router.sol#386)' in Router.removeLiquidityRemote(uint16,bytes,uint256,uint256,uint256,bytes) (contracts/Router.sol#374-416) potentially used before declaration: revertLookup[_chainId][_srcAddress][_nonce] = abi.encode(TYPE_SWAP_AND_MINT_REMOTE,_poolId,swapAmountSD,mintAmountSD,_to) (contracts/Router.sol#388-394) Variable 'Router._swapRemote(uint16,bytes,uint256,uint256,uint256,address,Pool.SwapObj,bytes).amountLD (contracts/Router.sol#500)' in Router._swapRemote(uint16,bytes,uint256,uint256,uint256,address,Pool.SwapObj,bytes) (contracts/Router.sol#488-551) potentially used before declaration: IStargateReceiver(_to).sgReceive{gas: _dstGasForCall}(_chainId,_srcAddress,_nonce,pool.token(),amountLD,_payload) (contracts/Router.sol#503-538) Variable 'Router._swapRemote(uint16,bytes,uint256,uint256,uint256,address,Pool.SwapObj,bytes).amountLD (contracts/Router.sol#500)' in Router._swapRemote(uint16,bytes,uint256,uint256,uint256,address,Pool.SwapObj,bytes) (contracts/Router.sol#488-551) potentially used before declaration: ReceiveFailed(_chainId,_srcAddress,_nonce,pool.token(),amountLD,_payload,reason) (contracts/Router.sol#514-522) Variable 'Router._swapRemote(uint16,bytes,uint256,uint256,uint256,address,Pool.SwapObj,bytes).reason (contracts/Router.sol#512)' in Router._swapRemote(uint16,bytes,uint256,uint256,uint256,address,Pool.SwapObj,bytes) (contracts/Router.sol#488-551) potentially used before declaration: ReceiveFailed(_chainId,_srcAddress,_nonce,pool.token(),amountLD,_payload,reason) (contracts/Router.sol#514-522) Variable 'Router._swapRemote(uint16,bytes,uint256,uint256,uint256,address,Pool.SwapObj,bytes).amountLD (contracts/Router.sol#500)' in Router._swapRemote(uint16,bytes,uint256,uint256,uint256,address,Pool.SwapObj,bytes) (contracts/Router.sol#488-551) potentially used before declaration: cachedSwapLookup[_chainId][_srcAddress][_nonce] = CachedSwap(pool.token(),amountLD,_to,_payload) (contracts/Router.sol#526-528) Variable 'Router._swapRemote(uint16,bytes,uint256,uint256,uint256,address,Pool.SwapObj,bytes).amountLD (contracts/Router.sol#500)' in Router._swapRemote(uint16,bytes,uint256,uint256,uint256,address,Pool.SwapObj,bytes) (contracts/Router.sol#488-551) potentially used before declaration: CachedSwapSaved(_chainId,_srcAddress,_nonce,pool.token(),amountLD,_to,_payload) (contracts/Router.sol#529-537) Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sc_concatStorage_asm_0 = keccak256(uint256,uint256)(0x0,0x20) +

slength_concatStorage_asm_0 / 32 (contracts/libraries/BytesLib.sol#195) Variable 'BytesLib.concatStorage(bytes,bytes).submod_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#161)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: submod_concatStorage_asm_0 = 32 - slengthmod_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#204) Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mc_concatStorage_asm_0 = _postBytes + submod_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#205) Variable 'BytesLib.concatStorage(bytes,bytes).submod_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#161)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mc_concatStorage_asm_0 = _postBytes + submod_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#205) Variable 'BytesLib.concatStorage(bytes,bytes).end_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#163)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: end_concatStorage_asm_0 = _postBytes + mlength_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#206) Variable 'BytesLib.concatStorage(bytes,bytes).submod_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#161)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mask_concatStorage_asm_0 = 0x100 ** submod_concatStorage_asm_0 - 1 (contracts/libraries/BytesLib.sol#207) Variable 'BytesLib.concatStorage(bytes,bytes).mask_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#164)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mask_concatStorage_asm_0 = 0x100 ** submod_concatStorage_asm_0 - 1 (contracts/libraries/BytesLib.sol#207) Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,sload(uint256)(sc_concatStorage_asm_0) + mload(uint256)(mc_concatStorage_asm_0) & mask_concatStorage_asm_0) (contracts/libraries/BytesLib.sol#209) Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,sload(uint256)(sc_concatStorage_asm_0) + mload(uint256)(mc_concatStorage_asm_0) & mask_concatStorage_asm_0) (contracts/libraries/BytesLib.sol#209) Variable 'BytesLib.concatStorage(bytes,bytes).mask_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#164)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,sload(uint256)(sc_concatStorage_asm_0) + mload(uint256)(mc_concatStorage_asm_0) & mask_concatStorage_asm_0) (contracts/libraries/BytesLib.sol#209) Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sc_concatStorage_asm_0 = sc_concatStorage_asm_0 + 1 (contracts/libraries/BytesLib.sol#212) Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mc_concatStorage_asm_0 = mc_concatStorage_asm_0 + 0x20 (contracts/libraries/BytesLib.sol#213) Variable 'BytesLib.concatStorage(bytes,bytes).end_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#163)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mc_concatStorage_asm_0 < end_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#214) Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mc_concatStorage_asm_0 < end_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#214) Variable 'BytesLib.concatStorage(bytes,bytes).end_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#163)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mask_concatStorage_asm_0 = 0x100 ** mc_concatStorage_asm_0 - end_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#221) Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mask_concatStorage_asm_0 = 0x100 ** mc_concatStorage_asm_0 - end_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#221) Variable 'BytesLib.concatStorage(bytes,bytes).mask_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#164)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mask_concatStorage_asm_0 = 0x100 ** mc_concatStorage_asm_0 - end_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#221) Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0) / mask_concatStorage_asm_0 * mask_concatStorage_asm_0) (contracts/libraries/BytesLib.sol#223) Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0) / mask_concatStorage_asm_0 * mask_concatStorage_asm_0) (contracts/libraries/BytesLib.sol#223) Variable 'BytesLib.concatStorage(bytes,bytes).mask_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#164)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0) / mask_concatStorage_asm_0 * mask_concatStorage_asm_0) (contracts/libraries/BytesLib.sol#223) Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0)) (contracts/libraries/BytesLib.sol#218) Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0)) (contracts/libraries/BytesLib.sol#218) Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: sc_concatStorage_asm_0 = sc_concatStorage_asm_0 + 1 (contracts/libraries/BytesLib.sol#215) Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (contracts/libraries/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) potentially used before declaration: mc_concatStorage_asm_0 = mc_concatStorage_asm_0 + 0x20 (contracts/libraries/BytesLib.sol#216) Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in Router._swapAndMint(uint16,bytes,uint256,uint256,address,uint256,uint256) (contracts/Router.sol#438-464): External calls: - pool.swapAndMint(_to,_amountSD,_mintAmountSD) (contracts/Router.sol#449-463) State variables written after the call(s): - revertLookup[_chainId][_srcAddress][_nonce] = abi.encode(TYPE_SWAP_AND_MINT_LOCAL,_poolId,_to,_amountSD,_mintAmountSD) (contracts/Router.sol#450-456) Reentrancy in Router._swapRemote(uint16,bytes,uint256,uint256,uint256,address,Pool.SwapObj,bytes) (contracts/Router.sol#488-551): External calls: - pool.swapRemote(_to,_s) (contracts/Router.sol#500-550) - IStargateReceiver(_to).sgReceive{gas: _dstGasForCall}(_chainId,_srcAddress,_nonce,pool.token(),amountLD,_payload) (contracts/Router.sol#503-538) State variables written after the call(s): - cachedSwapLookup[_chainId][_srcAddress][_nonce] = CachedSwap(pool.token(),amountLD,_to,_payload) (contracts/Router.sol#526-528) Reentrancy in Router._swapRemote(uint16,bytes,uint256,uint256,uint256,address,Pool.SwapObj,bytes) (contracts/Router.sol#488-551): External calls: - pool.swapRemote(_to,_s) (contracts/Router.sol#500-550) State variables written after the call(s): - revertLookup[_chainId][_srcAddress][_nonce] = abi.encode(TYPE_SWAP_REMOTE_LOCAL,_poolId,_dstGasForCall,_to,_s,_payload) (contracts/Router.sol#541-548) Reentrancy in Router.removeLiquidityRemote(uint16,bytes,uint256,uint256,uint256,bytes) (contracts/Router.sol#374-416): External calls: - pool.withdrawRemote(_chainId,_amountSD) (contracts/Router.sol#384-415) State variables written after the call(s): - revertLookup[_chainId][_srcAddress][_nonce] = abi.encode(TYPE_SWAP_AND_MINT_REMOTE,_poolId,swapAmountSD,mintAmountSD,_to) (contracts/Router.sol#388-394) - revertLookup[_chainId][_srcAddress][_nonce] = abi.encode(TYPE_SWAP_AND_MINT_REMOTE,_poolId,0,_amountSD,_to) (contracts/Router.sol#402-408) Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in Router._swapAndMint(uint16,bytes,uint256,uint256,address,uint256,uint256) (contracts/Router.sol#438-464): External calls: - pool.swapAndMint(_to,_amountSD,_mintAmountSD) (contracts/Router.sol#449-463) Event emitted after the call(s): - Revert(TYPE_SWAP_AND_MINT_LOCAL,_chainId,_srcAddress,_nonce) (contracts/Router.sol#457-462) Reentrancy in Router._swapRemote(uint16,bytes,uint256,uint256,uint256,address,Pool.SwapObj,bytes) (contracts/Router.sol#488-551): External calls: - pool.swapRemote(_to,_s) (contracts/Router.sol#500-550) - IStargateReceiver(_to).sgReceive{gas: _dstGasForCall}(_chainId,_srcAddress,_nonce,pool.token(),amountLD,_payload) (contracts/Router.sol#503-538) Event emitted after the call(s): - CachedSwapSaved(_chainId,_srcAddress,_nonce,pool.token(),amountLD,_to,_payload) (contracts/Router.sol#529-537) - ReceiveFailed(_chainId,_srcAddress,_nonce,pool.token(),amountLD,_payload,reason) (contracts/Router.sol#514-522) Reentrancy in Router._swapRemote(uint16,bytes,uint256,uint256,uint256,address,Pool.SwapObj,bytes) (contracts/Router.sol#488-551): External calls: - pool.swapRemote(_to,_s) (contracts/Router.sol#500-550) Event emitted after the call(s): - Revert(TYPE_SWAP_REMOTE_LOCAL,_chainId,_srcAddress,_nonce) (contracts/Router.sol#549) Reentrancy

in LPStaking.deposit(uint256,uint256) (contracts/LPStaking.sol#210-230): External calls: - updatePool(_pid) (contracts/LPStaking.sol#213) - stargate.mint(devaddr,stargateReward.div(10)) (contracts/LPStaking.sol#201) - stargate.mint(address(this),stargateReward) (contracts/LPStaking.sol#202) - safeStargateTransfer(msg.sender,pending) (contracts/LPStaking.sol#220) - stargate.transfer(_to,stargateBal) (contracts/LPStaking.sol#264) - stargate.transfer(_to,_amount) (contracts/LPStaking.sol#266) - pool.lpToken.safeTransferFrom(address(msg.sender),address(this),_amount) (contracts/LPStaking.sol#222-226) Event emitted after the call(s): - Deposit(msg.sender,_pid,_amount) (contracts/LPStaking.sol#229) Reentrancy in LPStaking.emergencyWithdraw(uint256) (contracts/LPStaking.sol#251-258): External calls: - pool.lpToken.safeTransfer(address(msg.sender),user.amount) (contracts/LPStaking.sol#254) Event emitted after the call(s): - EmergencyWithdraw(msg.sender,_pid,user.amount) (contracts/LPStaking.sol#255) Reentrancy in Router.removeLiquidityRemote(uint16,bytes,uint256,uint256,uint256,bytes) (contracts/Router.sol#374-416): External calls: - pool.withdrawRemote(_chainId,_amountSD) (contracts/Router.sol#384-415) Event emitted after the call(s): - Revert(TYPE_SWAP_AND_MINT_REMOTE,_chainId,_srcAddress,_nonce) (contracts/Router.sol#395-400) - Revert(TYPE_SWAP_AND_MINT_REMOTE,_chainId,_srcAddress,_nonce) (contracts/Router.sol#409-414) Reentrancy in Pool.swapAndMint(address,uint256,uint256) (contracts/Pool.sol#415-427): External calls: - _safeTransfer(token,_to,amountLD) (contracts/Pool.sol#425) - (success,data) = _token.call(abi.encodeWithSelector(SELECTOR,_to,_value)) (contracts/Pool.sol#237-239) Event emitted after the call(s): - SwapAndMint(_to,_amountSD,_amountToMintSD) (contracts/Pool.sol#426) Reentrancy in Pool.swapRemote(address,Pool.SwapObj) (contracts/Pool.sol#386-407): External calls: - _safeTransfer(token,_to,amountLD) (contracts/Pool.sol#400) - (success,data) = _token.call(abi.encodeWithSelector(SELECTOR,_to,_value)) (contracts/Pool.sol#237-239) Event emitted after the call(s): - SwapRemote(_to,_s.amount.add(_s.srcReward),_s.xStargateFee,_s.dstFee) (contracts/Pool.sol#401-406) Reentrancy in LPStaking.withdraw(uint256,uint256) (contracts/LPStaking.sol#233-248): External calls: - updatePool(_pid) (contracts/LPStaking.sol#237) - stargate.mint(devaddr,stargateReward.div(10)) (contracts/LPStaking.sol#201) - stargate.mint(address(this),stargateReward) (contracts/LPStaking.sol#202) - safeStargateTransfer(msg.sender,pending) (contracts/LPStaking.sol#243) - stargate.transfer(_to,stargateBal) (contracts/LPStaking.sol#264) - stargate.transfer(_to,_amount) (contracts/LPStaking.sol#266) - pool.lpToken.safeTransfer(address(msg.sender),_amount) (contracts/LPStaking.sol#246) Event emitted after the call(s): - Withdraw(msg.sender,_pid,_amount) (contracts/LPStaking.sol#247) Reentrancy in Pool.withdrawMintFeeBalance(address) (contracts/Pool.sol#544-555): External calls: - _safeTransfer(token,_to,amountOfLD) (contracts/Pool.sol#552) - (success,data) = _token.call(abi.encodeWithSelector(SELECTOR,_to,_value)) (contracts/Pool.sol#237-239) Event emitted after the call(s): - WithdrawMintFeeBalance(_to,amountOfLD) (contracts/Pool.sol#553) Reentrancy in Pool.withdrawXStargateFeeBalance(address) (contracts/Pool.sol#531-542): External calls: - _safeTransfer(token,_to,amountOfLD) (contracts/Pool.sol#539) - (success,data) = _token.call(abi.encodeWithSelector(SELECTOR,_to,_value)) (contracts/Pool.sol#237-239) Event emitted after the call(s): - WithdrawXStargateFeeBalance(_to,amountOfLD) (contracts/Pool.sol#540) Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

LPTokenERC20.permit(address,address,uint256,uint256,uint8,bytes32,bytes32) (contracts/LPTokenERC20.sol#107-140) uses timestamp for comparisons Dangerous comparisons: - require(bool,string)(deadline >= block.timestamp,Bridge: EXPIRED) (contracts/LPTokenERC20.sol#117) Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Address.isContract(address) (node_modules/@openzeppelin/contracts/utils/Address.sol#26-35) uses assembly - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#33) Address._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#171-188) uses assembly - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#180-183) Bridge.lzReceive(uint16,bytes,uint64,bytes) (contracts/Bridge.sol#107-212) uses assembly - INLINE ASM (contracts/Bridge.sol#121-124) - INLINE ASM (contracts/Bridge.sol#148-150) - INLINE ASM (contracts/Bridge.sol#181-183) LPTokenERC20.constructor() (contracts/LPTokenERC20.sol#31-47) uses assembly - INLINE ASM (contracts/LPTokenERC20.sol#33-35) Router.retryRevert(uint16,bytes,uint256,address) (contracts/Router.sol#255-335) uses assembly - INLINE ASM (contracts/Router.sol#269-271) BytesLib.concat(bytes,bytes) (contracts/libraries/BytesLib.sol#13-89) uses assembly - INLINE ASM (contracts/libraries/BytesLib.sol#23-86) BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) uses assembly - INLINE ASM (contracts/libraries/BytesLib.sol#92-225) BytesLib.slice(bytes,uint256,uint256) (contracts/libraries/BytesLib.sol#228-293) uses assembly - INLINE ASM (contracts/libraries/BytesLib.sol#243-290) BytesLib.toAddress(bytes,uint256) (contracts/libraries/BytesLib.sol#295-305) uses assembly - INLINE ASM (contracts/libraries/BytesLib.sol#300-302) BytesLib.toUint8(bytes,uint256) (contracts/libraries/BytesLib.sol#307-317) uses assembly - INLINE ASM (contracts/libraries/BytesLib.sol#312-314) BytesLib.toUint16(bytes,uint256) (contracts/libraries/BytesLib.sol#319-329) uses assembly - INLINE ASM (contracts/libraries/BytesLib.sol#324-326) BytesLib.toUint32(bytes,uint256) (contracts/libraries/BytesLib.sol#331-341) uses assembly - INLINE ASM (contracts/libraries/BytesLib.sol#336-338) BytesLib.toUint64(bytes,uint256) (contracts/libraries/BytesLib.sol#343-353) uses assembly - INLINE ASM (contracts/libraries/BytesLib.sol#348-350) BytesLib.toUint96(bytes,uint256) (contracts/libraries/BytesLib.sol#355-365) uses assembly - INLINE ASM (contracts/libraries/BytesLib.sol#360-362) BytesLib.toUint128(bytes,uint256) (contracts/libraries/BytesLib.sol#367-377) uses assembly - INLINE ASM (contracts/libraries/BytesLib.sol#372-374) BytesLib.toUint256(bytes,uint256) (contracts/libraries/BytesLib.sol#379-389) uses assembly - INLINE ASM (contracts/libraries/BytesLib.sol#384-386) BytesLib.toBytes32(bytes,uint256) (contracts/libraries/BytesLib.sol#391-401) uses assembly - INLINE ASM (contracts/libraries/BytesLib.sol#396-398) BytesLib.equal(bytes,bytes) (contracts/libraries/BytesLib.sol#403-444) uses assembly - INLINE ASM (contracts/libraries/BytesLib.sol#406-441) BytesLib.equalStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#446-516) uses assembly - INLINE ASM (contracts/libraries/BytesLib.sol#456-513) Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used: - Version used: ['0.7.6', '>=0.6.0<0.8.0', '>=0.6.2<0.8.0', '^0.7.6'] - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/access/Ownable.sol#3) - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/math/SafeMath.sol#3) - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#3) - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#3) - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/SafeERC20.sol#3) - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/utils/Address.sol#3) - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#3) - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#3) - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/ReentrancyGuard.sol#3) - 0.7.6 (contracts/Bridge.sol#3) - ABIEncoderV2 (contracts/Bridge.sol#4) - 0.7.6 (contracts/Factory.sol#3) - ABIEncoderV2 (contracts/Factory.sol#4) - 0.7.6 (contracts/LPStaking.sol#3) - 0.7.6 (contracts/LPTokenERC20.sol#3) - 0.7.6 (contracts/Pool.sol#3) - v2 (contracts/Pool.sol#4) - 0.7.6 (contracts/Router.sol#3) - v2 (contracts/Router.sol#4) - 0.7.6 (contracts/Staking.sol#3) - 0.7.6 (contracts/StargateToken.sol#3) - ^0.7.6 (contracts/interfaces/ILayerZeroEndpoint.sol#3) - ^0.7.6 (contracts/interfaces/ILayerZeroReceiver.sol#3) - ^0.7.6 (contracts/interfaces/ILayerZeroUserApplicationConfig.sol#3) - ^0.7.6 (contracts/interfaces/IStargateReceiver.sol#3) - ^0.7.6 (contracts/interfaces/IStargateRouter.sol#3) - 0.7.6 (contracts/libraries/BytesLib.sol#9) Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

BytesLib.concat(bytes,bytes) (contracts/libraries/BytesLib.sol#13-89) is never used and should be removed BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#91-226) is never used and should be removed BytesLib.equalStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#446-516) is never used and should be removed BytesLib.slice(bytes,uint256,uint256) (contracts/libraries/BytesLib.sol#228-293) is never used and should be removed BytesLib.toAddress(bytes,uint256) (contracts/libraries/BytesLib.sol#295-305) is never used and should be removed BytesLib.toBytes32(bytes,uint256) (contracts/libraries/BytesLib.sol#391-401) is never used and should be removed BytesLib.toUint128(bytes,uint256) (contracts/libraries/BytesLib.sol#367-377) is never used and should be removed BytesLib.toUint16(bytes,uint256) (contracts/libraries/BytesLib.sol#319-329) is never used and should be removed BytesLib.toUint256(bytes,uint256) (contracts/libraries/BytesLib.sol#379-389) is never used and should be removed BytesLib.toUint32(bytes,uint256) (contracts/libraries/BytesLib.sol#331-341) is never used and should be removed BytesLib.toUint64(bytes,uint256) (contracts/libraries/BytesLib.sol#343-353) is never used and should be removed BytesLib.toUint8(bytes,uint256) (contracts/libraries/BytesLib.sol#307-317) is never used and should be removed BytesLib.toUint96(bytes,uint256) (contracts/libraries/BytesLib.sol#355-365) is never used and should be removed Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/access/Ownable.sol#3) is too complex Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/math/SafeMath.sol#3) is too complex Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#3) is too complex Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#3) is too complex Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/SafeERC20.sol#3) is too complex Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/utils/Address.sol#3) is too complex Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#3) is too complex Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#3) is too complex Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/ReentrancyGuard.sol#3) is too complex Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#53-59): - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts/utils/Address.sol#57) Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-121): - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#119) Low level call in Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#139-145): - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#143) Low level call in Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#163-169): - (success,returndata) = target.delegatecall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#167) Low level call in Pool._safeTransfer(address,address,uint256) (contracts/Pool.sol#232-244): - (success,data) = _token.call(abi.encodeWithSelector(SELECTOR,_to,_value)) (contracts/Pool.sol#237-239) Low level call in Router._safeTransferFrom(address,address,address,uint256) (contracts/Router.sol#118-132): - (success,data) = token.call(abi.encodeWithSelector(0x23b872dd,from,to,value)) (contracts/Router.sol#125-127) Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Parameter Bridge.setRouter(Router)._router (contracts/Bridge.sol#63) is not in mixedCase Parameter Bridge.gasAmountBuilderV1(uint256)._gasAmount (contracts/Bridge.sol#72) is not in mixedCase Parameter Bridge.lzReceive(uint16,bytes,uint64,bytes)._srcChainId (contracts/Bridge.sol#108) is not in mixedCase Parameter Bridge.lzReceive(uint16,bytes,uint64,bytes)._srcAddress (contracts/Bridge.sol#109) is not in mixedCase Parameter Bridge.lzReceive(uint16,bytes,uint64,bytes)._nonce (contracts/Bridge.sol#110) is not in mixedCase Parameter Bridge.lzReceive(uint16,bytes,uint64,bytes)._payload (contracts/Bridge.sol#111) is not in mixedCase Parameter Bridge.swap(uint16,uint256,address,uint256,Pool.SwapObj,uint256,bytes,bytes)._chainId (contracts/Bridge.sol#217) is not in mixedCase Parameter Bridge.swap(uint16,uint256,address,uint256,Pool.SwapObj,uint256,bytes,bytes)._poolId (contracts/Bridge.sol#218) is not in mixedCase Parameter Bridge.swap(uint16,uint256,address,uint256,Pool.SwapObj,uint256,bytes,bytes)._refundAddress (contracts/Bridge.sol#219) is not in mixedCase Parameter Bridge.swap(uint16,uint256,address,uint256,Pool.SwapObj,uint256,bytes,bytes)._creditAmountSD (contracts/Bridge.sol#220) is not in mixedCase Parameter Bridge.swap(uint16,uint256,address,uint256,Pool.SwapObj,uint256,bytes,bytes)._s (contracts/Bridge.sol#221) is not in mixedCase Parameter Bridge.swap(uint16,uint256,address,uint256,Pool.SwapObj,uint256,bytes,bytes)._dstGasForCall (contracts/Bridge.sol#222) is not in mixedCase Parameter Bridge.swap(uint16,uint256,address,uint256,Pool.SwapObj,uint256,bytes,bytes)._to (contracts/Bridge.sol#223) is not in mixedCase Parameter Bridge.swap(uint16,uint256,address,uint256,Pool.SwapObj,uint256,bytes,bytes)._payload (contracts/Bridge.sol#224) is not in mixedCase Parameter Bridge.removeLiquidityAndRedeemOnRemote(uint16,uint256,address,uint256,uint256,bytes)._chainId (contracts/Bridge.sol#245) is not in mixedCase Parameter Bridge.removeLiquidityAndRedeemOnRemote(uint16,uint256,address,uint256,uint256,bytes)._poolId (contracts/Bridge.sol#246) is not in mixedCase Parameter Bridge.removeLiquidityAndRedeemOnRemote(uint16,uint256,address,uint256,uint256,bytes)._refundAddress (contracts/Bridge.sol#247) is not in mixedCase Parameter Bridge.removeLiquidityAndRedeemOnRemote(uint16,uint256,address,uint256,uint256,bytes)._creditAmountSD (contracts/Bridge.sol#248) is not in mixedCase Parameter Bridge.removeLiquidityAndRedeemOnRemote(uint16,uint256,address,uint256,uint256,bytes)._amountSD (contracts/Bridge.sol#249) is not in mixedCase Parameter Bridge.removeLiquidityAndRedeemOnRemote(uint16,uint256,address,uint256,uint256,bytes)._to (contracts/Bridge.sol#250) is not in mixedCase Parameter Bridge.swapAndMint(uint16,uint256,address,uint256,uint256,uint256,bytes)._chainId (contracts/Bridge.sol#265) is not in mixedCase Parameter Bridge.swapAndMint(uint16,uint256,address,uint256,uint256,uint256,bytes)._poolId (contracts/Bridge.sol#266) is not in mixedCase Parameter Bridge.swapAndMint(uint16,uint256,address,uint256,uint256,uint256,bytes)._refundAddress (contracts/Bridge.sol#267) is not in mixedCase Parameter Bridge.swapAndMint(uint16,uint256,address,uint256,uint256,uint256,bytes)._creditAmountSD (contracts/Bridge.sol#268) is not in mixedCase Parameter Bridge.swapAndMint(uint16,uint256,address,uint256,uint256,uint256,bytes)._amountSD (contracts/Bridge.sol#269) is not in mixedCase Parameter Bridge.swapAndMint(uint16,uint256,address,uint256,uint256,uint256,bytes)._mintAmountSD (contracts/Bridge.sol#270) is not in mixedCase Parameter Bridge.swapAndMint(uint16,uint256,address,uint256,uint256,uint256,bytes)._to (contracts/Bridge.sol#271) is not in mixedCase Parameter Bridge.removeLiquidity(uint16,uint256,address,uint256,uint256,bytes)._chainId (contracts/Bridge.sol#285) is not in mixedCase Parameter Bridge.removeLiquidity(uint16,uint256,address,uint256,uint256,bytes)._poolId (contracts/Bridge.sol#286) is not in mixedCase Parameter Bridge.removeLiquidity(uint16,uint256,address,uint256,uint256,bytes)._refundAddress (contracts/Bridge.sol#287) is not in mixedCase Parameter Bridge.removeLiquidity(uint16,uint256,address,uint256,uint256,bytes)._creditAmountSD (contracts/Bridge.sol#288) is not in mixedCase Parameter Bridge.removeLiquidity(uint16,uint256,address,uint256,uint256,bytes)._amountSD (contracts/Bridge.sol#289) is not in mixedCase Parameter Bridge.removeLiquidity(uint16,uint256,address,uint256,uint256,bytes)._to (contracts/Bridge.sol#290) is not in mixedCase Parameter Bridge.sendCredits(uint16,uint256,address,uint256)._chainId (contracts/Bridge.sol#303) is not in mixedCase Parameter Bridge.sendCredits(uint16,uint256,address,uint256)._poolId (contracts/Bridge.sol#304) is not in mixedCase Parameter Bridge.sendCredits(uint16,uint256,address,uint256)._refundAddress (contracts/Bridge.sol#305) is not in mixedCase Parameter Bridge.sendCredits(uint16,uint256,address,uint256)._creditAmountSD (contracts/Bridge.sol#306) is not in mixedCase Parameter Bridge.quoteLayerZeroFee(uint16,uint8,bytes,bytes)._chainId (contracts/Bridge.sol#317) is not in mixedCase Parameter Bridge.quoteLayerZeroFee(uint16,uint8,bytes,bytes)._functionType (contracts/Bridge.sol#318) is not in mixedCase Parameter Bridge.quoteLayerZeroFee(uint16,uint8,bytes,bytes)._toAddress (contracts/Bridge.sol#319) is not in mixedCase Parameter Bridge.quoteLayerZeroFee(uint16,uint8,bytes,bytes)._transferAndCallPayload (contracts/Bridge.sol#320) is not in mixedCase Parameter Bridge.setBridge(uint16,bytes)._chainId (contracts/Bridge.sol#352) is not in mixedCase Parameter Bridge.setBridge(uint16,bytes)._bridgeAddress (contracts/Bridge.sol#352) is not in mixedCase Parameter Bridge.setGasAmount(uint16,uint8,uint256)._chainId (contracts/Bridge.sol#365) is not in mixedCase Parameter Bridge.setGasAmount(uint16,uint8,uint256)._functionType (contracts/Bridge.sol#366) is not in mixedCase Parameter Bridge.setGasAmount(uint16,uint8,uint256)._gasAmount (contracts/Bridge.sol#367) is not in mixedCase Parameter Bridge.setOracle(uint16,address)._chainId (contracts/Bridge.sol#394) is not in mixedCase Parameter Bridge.setOracle(uint16,address)._oracle (contracts/Bridge.sol#394) is not in mixedCase Parameter Bridge.setRelayer(uint16,address)._chainId (contracts/Bridge.sol#405) is not in mixedCase Parameter Bridge.setRelayer(uint16,address)._relayer (contracts/Bridge.sol#405) is not in mixedCase Parameter Bridge.setBlockConfirmations(uint16,uint256)._chainId (contracts/Bridge.sol#416) is not in mixedCase Parameter Bridge.setBlockConfirmations(uint16,uint256)._blockConfirmations (contracts/Bridge.sol#416) is not in mixedCase Parameter Bridge.setLibraryVersion(uint16,uint16)._chainId (contracts/Bridge.sol#425) is not in mixedCase Parameter Bridge.setLibraryVersion(uint16,uint16)._libraryVersion (contracts/Bridge.sol#425) is not in mixedCase Parameter Bridge.getOracle(uint16,address)._chainId (contracts/Bridge.sol#434) is not in mixedCase Parameter Bridge.getRelayer(uint16,address)._chainId (contracts/Bridge.sol#447) is not in mixedCase Parameter Bridge.getBlockConfirmations(uint16,address)._chainId (contracts/Bridge.sol#460) is not in mixedCase Parameter Bridge.getLibraryVersion(uint16,address)._chainId (contracts/Bridge.sol#473) is not in mixedCase Parameter Factory.setRouter(address)._router (contracts/Factory.sol#29) is not in mixedCase Parameter Factory.createPair(uint256,address,uint8,uint8)._poolId (contracts/Factory.sol#44) is not in mixedCase Parameter Factory.createPair(uint256,address,uint8,uint8)._token (contracts/Factory.sol#45) is not in mixedCase Parameter Factory.createPair(uint256,address,uint8,uint8)._sharedDecimals (contracts/Factory.sol#46) is not in mixedCase Parameter Factory.createPair(uint256,address,uint8,uint8)._localDecimals (contracts/Factory.sol#47) is not in mixedCase Parameter LPStaking.add(uint256,IERC20,bool)._allocPoint (contracts/LPStaking.sol#91) is not in mixedCase Parameter LPStaking.add(uint256,IERC20,bool)._lpToken (contracts/LPStaking.sol#92) is not in mixedCase Parameter LPStaking.add(uint256,IERC20,bool)._withUpdate (contracts/LPStaking.sol#93) is not in mixedCase Parameter LPStaking.set(uint256,uint256,bool)._pid (contracts/LPStaking.sol#117) is not in mixedCase Parameter LPStaking.set(uint256,uint256,bool)._allocPoint (contracts/LPStaking.sol#118) is not in mixedCase Parameter LPStaking.set(uint256,uint256,bool)._withUpdate (contracts/LPStaking.sol#119) is not in mixedCase Parameter LPStaking.getMultiplier(uint256,uint256)._from (contracts/LPStaking.sol#132) is not in mixedCase Parameter LPStaking.getMultiplier(uint256,uint256)._to (contracts/LPStaking.sol#132) is not in mixedCase Parameter LPStaking.pendingStargate(uint256,address)._pid (contracts/LPStaking.sol#150) is not in mixedCase Parameter LPStaking.pendingStargate(uint256,address)._user (contracts/LPStaking.sol#150) is not in mixedCase Parameter LPStaking.updatePool(uint256)._pid (contracts/LPStaking.sol#186) is not in mixedCase Parameter LPStaking.deposit(uint256,uint256)._pid (contracts/LPStaking.sol#210) is not in mixedCase Parameter LPStaking.deposit(uint256,uint256)._amount (contracts/LPStaking.sol#210) is not in mixedCase Parameter LPStaking.withdraw(uint256,uint256)._pid (contracts/LPStaking.sol#233) is not in mixedCase Parameter LPStaking.withdraw(uint256,uint256)._amount (contracts/LPStaking.sol#233) is not in mixedCase Parameter LPStaking.emergencyWithdraw(uint256)._pid (contracts/LPStaking.sol#251) is not in mixedCase Parameter LPStaking.safeStargateTransfer(address,uint256)._to (contracts/LPStaking.sol#261) is not in mixedCase Parameter LPStaking.safeStargateTransfer(address,uint256)._amount (contracts/LPStaking.sol#261) is not in mixedCase Parameter LPStaking.dev(address)._devaddr (contracts/LPStaking.sol#271) is not in mixedCase Variable LPTokenERC20.DOMAIN_SEPARATOR (contracts/LPTokenERC20.sol#18) is not in mixedCase Parameter Pool.amountSDtoLD(uint256)._amount (contracts/Pool.sol#147) is not in mixedCase Parameter Pool.amountLDtoSD(uint256)._amount (contracts/Pool.sol#152) is not in mixedCase Parameter Pool.mint(address,uint256)._to (contracts/Pool.sol#251) is not in mixedCase Parameter Pool.mint(address,uint256)._amountLD (contracts/Pool.sol#251) is not in mixedCase Parameter Pool.swap(uint16,address,uint256,uint256)._chainId (contracts/Pool.sol#268) is not in mixedCase Parameter Pool.swap(uint16,address,uint256,uint256)._from (contracts/Pool.sol#269) is not in mixedCase Parameter Pool.swap(uint16,address,uint256,uint256)._amountLD

(contracts/Pool.sol#270) is not in mixedCase Parameter Pool.swap(uint16,address,uint256,uint256)._minAmountLD (contracts/Pool.sol#271) is not in mixedCase Parameter Pool.sendCredits(uint16)._chainId (contracts/Pool.sol#322) is not in mixedCase Parameter Pool.burnLocalAndRedeemOnRemote(uint16,address,uint256)._chainId (contracts/Pool.sol#339) is not in mixedCase Parameter Pool.burnLocalAndRedeemOnRemote(uint16,address,uint256)._from (contracts/Pool.sol#340) is not in mixedCase Parameter Pool.burnLocalAndRedeemOnRemote(uint16,address,uint256)._amountLP (contracts/Pool.sol#341) is not in mixedCase Parameter Pool.burnAndRedeem(address,uint256)._from (contracts/Pool.sol#355) is not in mixedCase Parameter Pool.burnAndRedeem(address,uint256)._amountLP (contracts/Pool.sol#355) is not in mixedCase Parameter Pool.addToBalanceFromRemote(uint16,uint256)._chainId (contracts/Pool.sol#372) is not in mixedCase Parameter Pool.addToBalanceFromRemote(uint16,uint256)._amountSD (contracts/Pool.sol#372) is not in mixedCase Parameter Pool.swapRemote(address,Pool.SwapObj)._to (contracts/Pool.sol#386) is not in mixedCase Parameter Pool.swapRemote(address,Pool.SwapObj)._s (contracts/Pool.sol#386) is not in mixedCase Parameter Pool.swapAndMint(address,uint256,uint256)._to (contracts/Pool.sol#416) is not in mixedCase Parameter Pool.swapAndMint(address,uint256,uint256)._amountSD (contracts/Pool.sol#417) is not in mixedCase Parameter Pool.swapAndMint(address,uint256,uint256)._amountToMintSD (contracts/Pool.sol#418) is not in mixedCase Parameter Pool.withdrawRemote(uint16,uint256)._chainId (contracts/Pool.sol#433) is not in mixedCase Parameter Pool.withdrawRemote(uint16,uint256)._amountSD (contracts/Pool.sol#433) is not in mixedCase Parameter Pool.createChainPath(uint16,uint256)._chainId (contracts/Pool.sol#457) is not in mixedCase Parameter Pool.createChainPath(uint16,uint256)._weight (contracts/Pool.sol#457) is not in mixedCase Parameter Pool.setWeightForChainPath(uint16,uint16)._chainId (contracts/Pool.sol#476) is not in mixedCase Parameter Pool.setWeightForChainPath(uint16,uint16)._weight (contracts/Pool.sol#476) is not in mixedCase Parameter Pool.setFee(uint256,uint256,uint256)._lpFeeBP (contracts/Pool.sol#499) is not in mixedCase Parameter Pool.setFee(uint256,uint256,uint256)._xStargateFeeBP (contracts/Pool.sol#500) is not in mixedCase Parameter Pool.setFee(uint256,uint256,uint256)._mintFeeBP (contracts/Pool.sol#501) is not in mixedCase Parameter Pool.withdrawXStargateFeeBalance(address)._to (contracts/Pool.sol#531) is not in mixedCase Parameter Pool.withdrawMintFeeBalance(address)._to (contracts/Pool.sol#544) is not in mixedCase Parameter Router.setBridgeAndFactory(Bridge,Factory)._bridge (contracts/Router.sol#88) is not in mixedCase Parameter Router.setBridgeAndFactory(Bridge,Factory)._factory (contracts/Router.sol#88) is not in mixedCase Parameter Router.getPool(uint256)._poolId (contracts/Router.sol#107) is not in mixedCase Parameter Router.addLiquidity(uint256,uint256,address)._poolId (contracts/Router.sol#137) is not in mixedCase Parameter Router.addLiquidity(uint256,uint256,address)._amountLD (contracts/Router.sol#138) is not in mixedCase Parameter Router.addLiquidity(uint256,uint256,address)._to (contracts/Router.sol#139) is not in mixedCase Parameter Router.swap(uint16,uint256,address,uint256,uint256,uint256,bytes,bytes)._chainId (contracts/Router.sol#147) is not in mixedCase Parameter Router.swap(uint16,uint256,address,uint256,uint256,uint256,bytes,bytes)._poolId (contracts/Router.sol#148) is not in mixedCase Parameter Router.swap(uint16,uint256,address,uint256,uint256,uint256,bytes,bytes)._refundAddress (contracts/Router.sol#149) is not in mixedCase Parameter Router.swap(uint16,uint256,address,uint256,uint256,uint256,bytes,bytes)._amountLD (contracts/Router.sol#150) is not in mixedCase Parameter Router.swap(uint16,uint256,address,uint256,uint256,uint256,bytes,bytes)._minAmountLD (contracts/Router.sol#151) is not in mixedCase Parameter Router.swap(uint16,uint256,address,uint256,uint256,uint256,bytes,bytes)._dstGasForCall (contracts/Router.sol#152) is not in mixedCase Parameter Router.swap(uint16,uint256,address,uint256,uint256,uint256,bytes,bytes)._to (contracts/Router.sol#153) is not in mixedCase Parameter Router.swap(uint16,uint256,address,uint256,uint256,uint256,bytes,bytes)._payload (contracts/Router.sol#154) is not in mixedCase Parameter Router.removeLiquidityAndRedeemOnRemote(uint16,uint256,address,uint256,bytes)._chainId (contracts/Router.sol#178) is not in mixedCase Parameter Router.removeLiquidityAndRedeemOnRemote(uint16,uint256,address,uint256,bytes)._poolId (contracts/Router.sol#179) is not in mixedCase Parameter Router.removeLiquidityAndRedeemOnRemote(uint16,uint256,address,uint256,bytes)._refundAddress (contracts/Router.sol#180) is not in mixedCase Parameter Router.removeLiquidityAndRedeemOnRemote(uint16,uint256,address,uint256,bytes)._amountLP (contracts/Router.sol#181) is not in mixedCase Parameter Router.removeLiquidityAndRedeemOnRemote(uint16,uint256,address,uint256,bytes)._to (contracts/Router.sol#182) is not in mixedCase Parameter Router.removeLiquidity(uint16,uint256,address,uint256,bytes)._chainId (contracts/Router.sol#204) is not in mixedCase Parameter Router.removeLiquidity(uint16,uint256,address,uint256,bytes)._poolId (contracts/Router.sol#205) is not in mixedCase Parameter Router.removeLiquidity(uint16,uint256,address,uint256,bytes)._refundAddress (contracts/Router.sol#206) is not in mixedCase Parameter Router.removeLiquidity(uint16,uint256,address,uint256,bytes)._amountLP (contracts/Router.sol#207) is not in mixedCase Parameter Router.removeLiquidity(uint16,uint256,address,uint256,bytes)._to (contracts/Router.sol#208) is not in mixedCase Parameter Router.sendCredits(uint16,uint256,address)._chainId (contracts/Router.sol#226) is not in mixedCase Parameter Router.sendCredits(uint16,uint256,address)._poolId (contracts/Router.sol#227) is not in mixedCase Parameter Router.sendCredits(uint16,uint256,address)._refundAddress (contracts/Router.sol#228) is not in mixedCase Parameter Router.quoteLayerZeroFee(uint16,uint8,bytes,bytes)._chainId (contracts/Router.sol#242) is not in mixedCase Parameter Router.quoteLayerZeroFee(uint16,uint8,bytes,bytes)._functionType (contracts/Router.sol#243) is not in mixedCase Parameter Router.quoteLayerZeroFee(uint16,uint8,bytes,bytes)._toAddress (contracts/Router.sol#244) is not in mixedCase Parameter Router.quoteLayerZeroFee(uint16,uint8,bytes,bytes)._transferAndCallPayload (contracts/Router.sol#245) is not in mixedCase Parameter Router.retryRevert(uint16,bytes,uint256,address)._chainId (contracts/Router.sol#256) is not in mixedCase Parameter Router.retryRevert(uint16,bytes,uint256,address)._srcAddress (contracts/Router.sol#257) is not in mixedCase Parameter Router.retryRevert(uint16,bytes,uint256,address)._nonce (contracts/Router.sol#258) is not in mixedCase Parameter Router.retryRevert(uint16,bytes,uint256,address)._refundAddress (contracts/Router.sol#259) is not in mixedCase Parameter Router.clearCachedSwap(uint16,bytes,uint256)._chainId (contracts/Router.sol#338) is not in mixedCase Parameter Router.clearCachedSwap(uint16,bytes,uint256)._srcAddress (contracts/Router.sol#339) is not in mixedCase Parameter Router.clearCachedSwap(uint16,bytes,uint256)._nonce (contracts/Router.sol#340) is not in mixedCase Parameter Router.addLiquidityRemote(uint16,uint256,uint256)._chainId (contracts/Router.sol#366) is not in mixedCase Parameter Router.addLiquidityRemote(uint16,uint256,uint256)._poolId (contracts/Router.sol#367) is not in mixedCase Parameter Router.addLiquidityRemote(uint16,uint256,uint256)._amountSD (contracts/Router.sol#368) is not in mixedCase Parameter Router.removeLiquidityRemote(uint16,bytes,uint256,uint256,uint256,bytes)._chainId (contracts/Router.sol#375) is not in mixedCase Parameter Router.removeLiquidityRemote(uint16,bytes,uint256,uint256,uint256,bytes)._srcAddress (contracts/Router.sol#376) is not in mixedCase Parameter Router.removeLiquidityRemote(uint16,bytes,uint256,uint256,uint256,bytes)._nonce (contracts/Router.sol#377) is not in mixedCase Parameter Router.removeLiquidityRemote(uint16,bytes,uint256,uint256,uint256,bytes)._poolId (contracts/Router.sol#378) is not in mixedCase Parameter Router.removeLiquidityRemote(uint16,bytes,uint256,uint256,uint256,bytes)._amountSD (contracts/Router.sol#379) is not in mixedCase Parameter Router.removeLiquidityRemote(uint16,bytes,uint256,uint256,uint256,bytes)._to (contracts/Router.sol#380) is not in mixedCase Parameter Router.swapAndMint(uint16,bytes,uint256,uint256,address,uint256,uint256)._chainId (contracts/Router.sol#419) is not in mixedCase Parameter Router.swapAndMint(uint16,bytes,uint256,uint256,address,uint256,uint256)._srcAddress (contracts/Router.sol#420) is not in mixedCase Parameter Router.swapAndMint(uint16,bytes,uint256,uint256,address,uint256,uint256)._nonce (contracts/Router.sol#421) is not in mixedCase Parameter Router.swapAndMint(uint16,bytes,uint256,uint256,address,uint256,uint256)._poolId (contracts/Router.sol#422) is not in mixedCase Parameter Router.swapAndMint(uint16,bytes,uint256,uint256,address,uint256,uint256)._to (contracts/Router.sol#423) is not in mixedCase Parameter Router.swapAndMint(uint16,bytes,uint256,uint256,address,uint256,uint256)._amountSD (contracts/Router.sol#424) is not in mixedCase Parameter Router.swapAndMint(uint16,bytes,uint256,uint256,address,uint256,uint256)._mintAmountSD (contracts/Router.sol#425) is not in mixedCase Parameter Router.swapRemote(uint16,bytes,uint256,uint256,uint256,address,Pool.SwapObj,bytes)._chainId (contracts/Router.sol#467) is not in mixedCase Parameter Router.swapRemote(uint16,bytes,uint256,uint256,uint256,address,Pool.SwapObj,bytes)._srcAddress (contracts/Router.sol#468) is not in mixedCase Parameter Router.swapRemote(uint16,bytes,uint256,uint256,uint256,address,Pool.SwapObj,bytes)._nonce (contracts/Router.sol#469) is not in mixedCase Parameter Router.swapRemote(uint16,bytes,uint256,uint256,uint256,address,Pool.SwapObj,bytes)._poolId (contracts/Router.sol#470) is not in mixedCase Parameter Router.swapRemote(uint16,bytes,uint256,uint256,uint256,address,Pool.SwapObj,bytes)._dstGasForCall (contracts/Router.sol#471) is not in mixedCase Parameter Router.swapRemote(uint16,bytes,uint256,uint256,uint256,address,Pool.SwapObj,bytes)._to (contracts/Router.sol#472) is not in mixedCase Parameter Router.swapRemote(uint16,bytes,uint256,uint256,uint256,address,Pool.SwapObj,bytes)._s (contracts/Router.sol#473) is not in mixedCase Parameter Router.swapRemote(uint16,bytes,uint256,uint256,uint256,address,Pool.SwapObj,bytes)._payload (contracts/Router.sol#474) is not in mixedCase Parameter Router.createPair(uint256,address,uint8,uint8)._poolId (contracts/Router.sol#558) is not in mixedCase Parameter Router.createPair(uint256,address,uint8,uint8)._token (contracts/Router.sol#559) is not in mixedCase Parameter Router.createPair(uint256,address,uint8,uint8)._sharedDecimals (contracts/Router.sol#560) is not in mixedCase Parameter Router.createPair(uint256,address,uint8,uint8)._localDecimals (contracts/Router.sol#561) is not in mixedCase Parameter Router.createChainPath(uint256,uint16,uint256)._poolId (contracts/Router.sol#574) is not in mixedCase Parameter Router.createChainPath(uint256,uint16,uint256)._chainId (contracts/Router.sol#575) is not in mixedCase Parameter Router.createChainPath(uint256,uint16,uint256)._weight (contracts/Router.sol#576) is not in mixedCase Parameter Router.setWeightForChainPath(uint256,uint16,uint16)._poolId (contracts/Router.sol#583) is not in mixedCase Parameter Router.setWeightForChainPath(uint256,uint16,uint16)._chainId (contracts/Router.sol#584) is not in mixedCase Parameter Router.setWeightForChainPath(uint256,uint16,uint16)._weight (contracts/Router.sol#585) is not in mixedCase Parameter Router.setXStargateFeeOwner(address)._owner

(contracts/Router.sol#591) is not in mixedCase Parameter Router.setMintFeeOwner(address)._owner (contracts/Router.sol#596) is not in mixedCase Parameter Router.setFees(uint256,uint256,uint256,uint256)._poolId (contracts/Router.sol#602) is not in mixedCase Parameter Router.setFees(uint256,uint256,uint256,uint256)._lpFeeBP (contracts/Router.sol#603) is not in mixedCase Parameter Router.setFees(uint256,uint256,uint256,uint256)._xStargateFeeBP (contracts/Router.sol#604) is not in mixedCase Parameter Router.setFees(uint256,uint256,uint256,uint256)._mintFeeBP (contracts/Router.sol#605) is not in mixedCase Parameter Router.withdrawMintFee(uint256,address)._poolId (contracts/Router.sol#613) is not in mixedCase Parameter Router.withdrawMintFee(uint256,address)._to (contracts/Router.sol#613) is not in mixedCase Parameter Router.withdrawXStargateFee(uint256,address)._poolId (contracts/Router.sol#621) is not in mixedCase Parameter Router.withdrawXStargateFee(uint256,address)._to (contracts/Router.sol#621) is not in mixedCase Parameter Staking.enter(uint256)._amount (contracts/Staking.sol#26) is not in mixedCase Parameter Staking.leave(uint256)._share (contracts/Staking.sol#48) is not in mixedCase Parameter StargateToken.mint(address,uint256)._to (contracts/StargateToken.sol#22) is not in mixedCase Parameter StargateToken.mint(address,uint256)._qty (contracts/StargateToken.sol#22) is not in mixedCase Parameter BytesLib.concat(bytes,bytes)._preBytes (contracts/libraries/BytesLib.sol#14) is not in mixedCase Parameter BytesLib.concat(bytes,bytes)._postBytes (contracts/libraries/BytesLib.sol#15) is not in mixedCase Parameter BytesLib.concatStorage(bytes,bytes)._preBytes (contracts/libraries/BytesLib.sol#91) is not in mixedCase Parameter BytesLib.concatStorage(bytes,bytes)._postBytes (contracts/libraries/BytesLib.sol#91) is not in mixedCase Parameter BytesLib.slice(bytes,uint256,uint256)._bytes (contracts/libraries/BytesLib.sol#229) is not in mixedCase Parameter BytesLib.slice(bytes,uint256,uint256)._start (contracts/libraries/BytesLib.sol#230) is not in mixedCase Parameter BytesLib.slice(bytes,uint256,uint256)._length (contracts/libraries/BytesLib.sol#231) is not in mixedCase Parameter BytesLib.toAddress(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#295) is not in mixedCase Parameter BytesLib.toAddress(bytes,uint256)._start (contracts/libraries/BytesLib.sol#295) is not in mixedCase Parameter BytesLib.toUint8(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#307) is not in mixedCase Parameter BytesLib.toUint8(bytes,uint256)._start (contracts/libraries/BytesLib.sol#307) is not in mixedCase Parameter BytesLib.toUint16(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#319) is not in mixedCase Parameter BytesLib.toUint16(bytes,uint256)._start (contracts/libraries/BytesLib.sol#319) is not in mixedCase Parameter BytesLib.toUint32(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#331) is not in mixedCase Parameter BytesLib.toUint32(bytes,uint256)._start (contracts/libraries/BytesLib.sol#331) is not in mixedCase Parameter BytesLib.toUint64(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#343) is not in mixedCase Parameter BytesLib.toUint64(bytes,uint256)._start (contracts/libraries/BytesLib.sol#343) is not in mixedCase Parameter BytesLib.toUint96(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#355) is not in mixedCase Parameter BytesLib.toUint96(bytes,uint256)._start (contracts/libraries/BytesLib.sol#355) is not in mixedCase Parameter BytesLib.toUint128(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#367) is not in mixedCase Parameter BytesLib.toUint128(bytes,uint256)._start (contracts/libraries/BytesLib.sol#367) is not in mixedCase Parameter BytesLib.toUint256(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#379) is not in mixedCase Parameter BytesLib.toUint256(bytes,uint256)._start (contracts/libraries/BytesLib.sol#379) is not in mixedCase Parameter BytesLib.toBytes32(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#391) is not in mixedCase Parameter BytesLib.toBytes32(bytes,uint256)._start (contracts/libraries/BytesLib.sol#391) is not in mixedCase Parameter BytesLib.equal(bytes,bytes)._preBytes (contracts/libraries/BytesLib.sol#403) is not in mixedCase Parameter BytesLib.equal(bytes,bytes)._postBytes (contracts/libraries/BytesLib.sol#403) is not in mixedCase Parameter BytesLib.equalStorage(bytes,bytes)._preBytes (contracts/libraries/BytesLib.sol#447) is not in mixedCase Parameter BytesLib.equalStorage(bytes,bytes)._postBytes (contracts/libraries/BytesLib.sol#448) is not in mixedCase Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (node_modules/@openzeppelin/contracts/utils/Context.sol#21)" inContext (node_modules/@openzeppelin/contracts/utils/Context.sol#15-24) Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

Variable Bridge.lzReceive(uint16,bytes,uint64,bytes).creditAmountSD_scope_1 (contracts/Bridge.sol#163) is too similar to Bridge.lzReceive(uint16,bytes,uint64,bytes).creditAmountSD_scope_3 (contracts/Bridge.sol#172) Variable Bridge.lzReceive(uint16,bytes,uint64,bytes).creditAmountSD_scope_1 (contracts/Bridge.sol#163) is too similar to Bridge.lzReceive(uint16,bytes,uint64,bytes).creditAmountSD_scope_7 (contracts/Bridge.sol#198) Variable Bridge.lzReceive(uint16,bytes,uint64,bytes).creditAmountSD_scope_3 (contracts/Bridge.sol#172) is too similar to Bridge.lzReceive(uint16,bytes,uint64,bytes).creditAmountSD_scope_7 (contracts/Bridge.sol#198) Variable Bridge.lzReceive(uint16,bytes,uint64,bytes).poolId_scope_0 (contracts/Bridge.sol#163) is too similar to Bridge.lzReceive(uint16,bytes,uint64,bytes).poolId_scope_2 (contracts/Bridge.sol#171) Variable Bridge.lzReceive(uint16,bytes,uint64,bytes).poolId_scope_0 (contracts/Bridge.sol#163) is too similar to Bridge.lzReceive(uint16,bytes,uint64,bytes).poolId_scope_6 (contracts/Bridge.sol#197) Variable Bridge.lzReceive(uint16,bytes,uint64,bytes).poolId_scope_2 (contracts/Bridge.sol#171) is too similar to Bridge.lzReceive(uint16,bytes,uint64,bytes).poolId_scope_6 (contracts/Bridge.sol#197) Variable Router.retryRevert(uint16,bytes,uint256,address).poolId_scope_0 (contracts/Router.sol#295) is too similar to Router.retryRevert(uint16,bytes,uint256,address).poolId_scope_4 (contracts/Router.sol#315) Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

BytesLib.toAddress(bytes,uint256) (contracts/libraries/BytesLib.sol#295-305) uses literals with too many digits: - tempAddress = mload(uint256)(_bytes + 0x20 + _start) / 0x1000000000000000000000000 (contracts/libraries/BytesLib.sol#301) Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

renounceOwnership() should be declared external: - Ownable.renounceOwnership() (node_modules/@openzeppelin/contracts/access/Ownable.sol#54-57) transferOwnership(address) should be declared external: - Ownable.transferOwnership(address) (node_modules/@openzeppelin/contracts/access/Ownable.sol#63-67) name() should be declared external: - ERC20.name() (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#64-66) symbol() should be declared external: - ERC20.symbol() (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#72-74) decimals() should be declared external: - ERC20.decimals() (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#89-91) balanceOf(address) should be declared external: - ERC20.balanceOf(address) (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#103-105) transfer(address,uint256) should be declared external: - ERC20.transfer(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#115-118) allowance(address,address) should be declared external: - ERC20.allowance(address,address) (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#123-125) approve(address,uint256) should be declared external: - ERC20.approve(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#134-137) transferFrom(address,address,uint256) should be declared external: - ERC20.transferFrom(address,address,uint256) (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#152-156) increaseAllowance(address,uint256) should be declared external: - ERC20.increaseAllowance(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#170-173) decreaseAllowance(address,uint256) should be declared external: - ERC20.decreaseAllowance(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#189-192) createPair(uint256,address,uint8,uint8) should be declared external: - Factory.createPair(uint256,address,uint8,uint8) (contracts/Factory.sol#43-65) add(uint256,IERC20,bool) should be declared external: - LPStaking.add(uint256,IERC20,bool) (contracts/LPStaking.sol#90-113) set(uint256,uint256,bool) should be declared external: - LPStaking.set(uint256,uint256,bool) (contracts/LPStaking.sol#116-129) deposit(uint256,uint256) should be declared external: - LPStaking.deposit(uint256,uint256) (contracts/LPStaking.sol#210-230) withdraw(uint256,uint256) should be declared external: - LPStaking.withdraw(uint256,uint256) (contracts/LPStaking.sol#233-248) emergencyWithdraw(uint256) should be declared external: - LPStaking.emergencyWithdraw(uint256) (contracts/LPStaking.sol#251-258) dev(address) should be declared external: - LPStaking.dev(address) (contracts/LPStaking.sol#271-275) enter(uint256) should be declared external: - Staking.enter(uint256) (contracts/Staking.sol#26-44) leave(uint256) should be declared external: - Staking.leave(uint256) (contracts/Staking.sol#48-57) Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external . analyzed (23 contracts with 75 detectors), 412 result(s) found

Test Results

Test Suite Results

```
Bridge
   ✓ setBridge()
   ✓ setBridge() reverts when bridge already set
   ✓ setBridge() reverts for non owner
   ✓ setGasAmount() reverts for non owner
   ✓ setGasAmount() reverts for invalid function type
   ✓ approveTokenSpender() reverts for non owner
   ✓ approveTokenSpender() approves amount
   ✓ setUseLayerZeroToken() reverts for non owner

Factory
   ✓ allPairsLength()
```

```
    ✓ setRouter() reverts if router address has been set
    ✓ createPair() reverts if creatPair() is called for existing _poolId
    ✓ createPair() increments allPairsLength()

  Pool
    ✓ Should return the proper pool settings
    ✓ Should create a proper pool connection
    ✓ mint() reverts when called by non Router
    ✓ Should fail to mint with no chainpaths
    ✓ Should fail to mint with no weights for chainpaths
    ✓ Should mint to the user
    ✓ Should set weight for chain
    ✓ Should properly allocation to two pools based on weights
    ✓ Should add to balance for remote chain
    ✓ burnAndRedeem()
    ✓ addToBalanceFromRemote() emits event
    ✓ swapRemote() emits event
    ✓ swapAndMint() emits event
    ✓ withdrawRemote() emits event
    ✓ createChainPath() and setWeightForChainPath() emit correct event(s)
    ✓ setFee() emits correct event
    ✓ swap() reverts when called by non Router
    ✓ sendCredits() reverts when called by non Router
    ✓ burnLocalAndRedeemOnRemote() reverts when called by non Router
    ✓ burnAndRedeem() reverts when called by non Router
    ✓ addToBalanceFromRemote() reverts when called by non Router
    ✓ swapRemote() reverts when called by non Router
    ✓ swapAndMint() reverts when called by non Router
    ✓ withdrawRemote() reverts when called by non Router
    ✓ createChainPath() reverts when called by non Router
    ✓ setWeightForChainPath() reverts when called by non Router
    ✓ setWeightForChainPath() reverts when no chainPaths have been created yet
    ✓ setFee() reverts when called by non Router
    ✓ setFee() reverts cumulative fee exceeds 100%
    ✓ withdrawXStargateFeeBalance() reverts when called by non Router
    ✓ withdrawMintFeeBalance() reverts when called by non Router

  Router
    ✓ addLiquidity() reverts for non existant pool
    ✓ swap() - TODO
    ✓ removeLiquidityAndRedeemOnRemote() - TODO
    ✓ removeLiquidity() - TODO
    ✓ sendCredits() - TODO
    ✓ quoteLayerZeroFee() - TODO
    ✓ retryRevert() - reverts when theres nothing to try to retry
    ✓ retryRevert() - TYPE_SWAP_AND_MINT_REMOTE - TODO
    ✓ retryRevert() - TYPE_SWAP_AND_MINT_LOCAL - TODO
    ✓ retryRevert() - TYPE_SWAP_REMOTE_LOCAL - TODO
    ✓ clearCachedSwap() reverts when nothing to clean
    ✓ addLiquidityRemote() reverts when caller is no Bridge
    ✓ removeLiquitidyRemote() reverts when caller is no Bridge
    ✓ swapAndMint() reverts when caller is no Bridge
    ✓ swapRemote() reverts when caller is no Bridge
    ✓ createPair() reverts when caller is not the dao
    ✓ createChainPath() reverts when caller is not the dao
    ✓ setWeightForChainPath() reverts when caller is not the dao
    ✓ setXStargateFeeOwner() reverts when caller is not the dao
    ✓ setMintFeeOwner() reverts when caller is not the dao
    ✓ setFees() reverts when caller is not the dao
    ✓ withdrawMintFee() reverts when caller is not the mintFeeOwner
    ✓ withdrawXStargateFee() reverts when caller is not the mintFeeOwner

  Stargate singlechain w/ LayerZeroEndpointMock
    ✓ addLiquidity()
    ✓ lets swap em up!

  Stargate LPer (singlechain w/ LayerZeroEndpointMock)
    ✓ test remove liquidity from both sides

  StargateToken
    ✓ name
    ✓ symbol
    ✓ decimals
    ✓ mints deployer some initial supply
    ✓ mint() reverts when called by non Owner
```

Code Coverage

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
contracts/	55.8	46.09	59.43	57.49	
Bridge.sol	63.08	50	54.17	67.19	… 454,467,480
Factory.sol	100	83.33	100	100	
LPStaking.sol	0	0	0	0	… 266,273,274
LPTokenERC20.sol	30.77	0	33.33	33.33	… 134,135,139
Pool.sol	88.28	68.42	86.36	88.46	… 551,552,553
Router.sol	63.22	53.85	78.57	65.52	… 616,626,627
Staking.sol	0	0	0	0	… 50,52,55,56
StargateToken.sol	50	100	50	50	23
contracts/interfaces/	100	100	100	100	
ILayerZeroEndpoint.sol	100	100	100	100	
ILayerZeroReceiver.sol	100	100	100	100	
ILayerZeroUserApplicationConfig.sol	100	100	100	100	
IStargateReceiver.sol	100	100	100	100	
IStargateRouter.sol	100	100	100	100	
contracts/libraries/	4.26	0	7.14	4.92	
BytesLib.sol	4.26	0	7.14	4.92	… 454,456,515
contracts/mocks/	92.31	100	88.89	93.33	
LayerZeroEndpointMock.sol	90	100	80	91.67	56
MockToken.sol	100	100	100	100	
MockTokenNoInitialMint.sol	100	100	100	100	
All files	**51.61**	**34.71**	**55.81**	**51.97**	

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.

Contracts

```
fc684dddcbf76207753d9b3d52ae8505660808bced734908421584d6eb761042 ./contracts/Bridge.sol
7e1ecc0f5519c0ac81d06e94d10693f2978222387fb568c13790760f2384dfdb ./contracts/Factory.sol
a96cca143f632437b7b16662ace1ce352d6c67daafd5874107c3ec34c989c081 ./contracts/LPStaking.sol
acdd3c8ae6a2aa8910c4f20bc92b5d82938e8b50ed7852800532c29ee38e3a85 ./contracts/LPTokenERC20.sol
9bf1a5cdf834b747872792fc7472d690bca1ed4ff7f3c393c64c5d25af684240 ./contracts/Pool.sol
0b63eef7c2e826adfcdc6ce1cf7c398b7db5f085a3cde2e95ac0e3bb5e47de84 ./contracts/Router.sol
76ed839ef311863e9d8890661c3cef182e9a19c72b81bc483dbc84c792631507 ./contracts/Staking.sol
79ea4ca1df4bcf5421afa4ca2bb2c698f1795a9891b0c91ed4855f3e7ccbb412 ./contracts/StargateToken.sol
fa77bc4d637ce07a506308da0a501b2cc2402dd1abcaa705239652bb38e9e897 ./contracts/mocks/LayerZeroEndpointMock.sol
8eeea6cc702c05a33b7f4f4e282a62442b3d155923e8b825981cc7776484ae5f ./contracts/mocks/MockToken.sol
e21b938af81c0a66b5d22bb2fe7ed245c65fd9e746774e9f1f5f98839206e86b ./contracts/mocks/MockTokenNoInitialMint.sol
25f1ee730aa23848d7874b4a073a1b9aa7f31895b97759a32e59f2c48b1084da ./contracts/libraries/BytesLib.sol
8c3a8d39cb0cbae02aa181cd64d1af04e6454d784c24ce665bdf20a0cafa6864 ./contracts/interfaces/ILayerZeroEndpoint.sol
27b5d9e1d2775c903f1dbb9ad7bd22278a333de2fe7921aec263f3f583440886 ./contracts/interfaces/ILayerZeroReceiver.sol
eae8f47261c08b0478b0a6e3a4576faa0a3e1b414e8d4db8dc3fc44928ab5a6a ./contracts/interfaces/ILayerZeroUserApplicationConfig.sol
99f5ffafc47709384bd8fd024608186989e7fe439e2178896b47886ac8021d05 ./contracts/interfaces/IStargateReceiver.sol
3b7cf748310cf7362e40ca925137937fb81da589af2604cbd3a78e8ee5b9eec6 ./contracts/interfaces/IStargateRouter.sol
```

Tests

```
bd24582d5d6601639a8448e0b5dae7faac5324778a66b912d0ffb9118f2ebb6b ./test/Bridge.test.js
278bb4e048883cf884bd13e2fceaec5361faa050a928d5b673a7272e0de03acd ./test/Factory.test.js
1281233d0c8a7f4b5cb0be368e9e0de11fb9f638bc4761166148c3f945c3a91c ./test/Pool.test.js
c20b6b4187da37a93766ec31a93df3f26422be3b828dc90631ba0b212448dba1 ./test/Router.test.js
6761cc5e343aa73c89bf31e9f237750b580bf756876f8ee926bf71fc00222706 ./test/Stargate.test.js
b5a430635350a0309378d12775b6dbf0c69571bd59651a7ba941671d7db7636d ./test/StargateToken.test.js
```

Changelog

- 2021-11-28 - Initial report

About Quantstamp

Quantstamp is a Y Combinator-backed company that helps to secure blockchain platforms at scale using computer-aided reasoning tools, with a mission to help boost the adoption of this exponentially growing technology.

With over 1000 Google scholar citations and numerous published papers, Quantstamp's team has decades of combined experience in formal verification, static analysis, and software verification. Quantstamp has also developed a protocol to help smart contract developers and projects worldwide to perform cost-effective smart contract security scans.

To date, Quantstamp has protected $5B in digital asset risk from hackers and assisted dozens of blockchain projects globally through its white glove security assessment services. As an evangelist of the blockchain ecosystem, Quantstamp assists core infrastructure projects and leading community initiatives such as the Ethereum Community Fund to expedite the adoption of blockchain technology.

Quantstamp's collaborations with leading academic institutions such as the National University of Singapore and MIT (Massachusetts Institute of Technology) reflect our commitment to research, development, and enabling world-class blockchain security.