CAIRO SECURITY CLAN

ATOMIQ EXCHANGE - EVM

SECURITY ASSESEMENT REPORT

JULY 2025

Prepared for

ATOMIQ LABS

Contents

1 About Cairo Security Clan

Cairo Security Clan is a leading force in the realm of blockchain security, dedicated to fortifying the foundations of the digital age. As pioneers in the field, we specialize in conducting meticulous smart contract security audits, ensuring the integrity and reliability of decentralized applications built on blockchain technology.

At Cairo Security Clan, we boast a multidisciplinary team of seasoned professionals proficient in blockchain security, cryptography, and software engineering. With a firm commitment to excellence, our experts delve into every aspect of the Web3 ecosystem, from foundational layer protocols to application-layer development. Our comprehensive suite of services encompasses smart contract audits, formal verification, and real-time monitoring, offering unparalleled protection against potential vulnerabilities.

Our team comprises industry veterans and scholars with extensive academic backgrounds and practical experience. Armed with advanced methodologies and cutting-edge tools, we scrutinize and analyze complex smart contracts with precision and rigor. Our track record speaks volumes, with a plethora of published research papers and citations, demonstrating our unwavering dedication to advancing the field of blockchain security.

At Cairo Security Clan, we prioritize collaboration and transparency, fostering meaningful partnerships with our clients. We believe in a customer-oriented approach, engaging stakeholders at every stage of the auditing process. By maintaining open lines of communication and soliciting client feedback, we ensure that our solutions are tailored to meet the unique needs and objectives of each project.

Beyond our core services, Cairo Security Clan is committed to driving innovation and shaping the future of blockchain technology. As active contributors to the ecosystem, we participate in the development of emerging technologies such as Starknet, leveraging our expertise to build robust infrastructure and tools. Through strategic guidance and support, we empower our partners to navigate the complexities of the blockchain landscape with confidence and clarity.

In summary, Cairo Security Clan stands at the forefront of blockchain security, blending technical prowess with a client-centric ethos to deliver unparalleled protection and peace of mind in an ever-evolving digital landscape. Join us in safeguarding the future of decentralized finance and digital assets with confidence and conviction.

2 Disclaimer

Disclaimer Limitations of this Audit:

This report is based solely on the materials and documentation provided by you to Cairo Security Clan for the specific purpose of conducting the security review outlined in the Summary of Audit and Scoped Files. The findings presented here may not be exhaustive and may not identify all potential vulnerabilities. Cairo Security Clan provides this review and report on an "as-is" and "as-available" basis. You acknowledge that your use of this report, including any associated services, products, protocols, platforms, content, and materials, occurs entirely at your own risk.

Inherent Risks of Blockchain Technology:

Blockchain technology remains in its developmental stage and is inherently susceptible to unknown risks and vulnerabilities. This review is specifically focused on the smart contract code and does not extend to the compiler layer, programming language elements beyond the reviewed code, or other potential security risks outside the code itself.

Report Purpose and Reliance:

This report should not be construed as an endorsement of any specific project or team, nor does it guarantee the absolute security of the audited smart contracts. No third party should rely on this report for any purpose, including making investment or purchasing decisions.

Liability Disclaimer:

To the fullest extent permitted by law, Cairo Security Clan disclaims all liability associated with this report, its contents, and any related services and products arising from your use. This includes, but is not limited to, implied warranties of merchantability, fitness for a particular purpose, and non-infringement.

Third-Party Products and Services:

Cairo Security Clan does not warrant, endorse, guarantee, or assume responsibility for any products or services advertised by third parties within this report, nor for any open-source or third-party software, code, libraries, materials, or information linked to, referenced by, or accessible through this report, its content, and related services and products. This includes any hyperlinked websites, websites or applications appearing on advertisements, and Cairo Security Clan will not be responsible for monitoring any transactions between you and third-party providers. It is recommended that you exercise due diligence and caution when considering any third-party products or services, just as you would with any purchase or service through any medium.

Disclaimer of Advice:

FOR THE AVOIDANCE OF DOUBT, THIS REPORT, ITS CONTENT, ACCESS, AND/OR USE, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHOULD NOT BE CONSIDERED OR RELIED UPON AS FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER PROFESSIONAL ADVICE.

3 Executive Summary

This document presents the security review performed by Cairo Security Clan on the Atomiq Exchange.

atomiq.exchange is a fully trustless cross-chain decentralized exchange (DEX) allowing you to swap between smart chains and Bitcoin, without having to trust any intermediary in the process.

All transactions are processed atomically with strong security guarantees based on bitcoin light client (leveraging bitcoin's proof-of-work security) & submarine swaps (leveraging HTLCs - hash-time locked contracts over bitcoin's lightning network). With this approach atomiq.exchange is able to offer security guarantees far exceeding those of existing bridging or cross-chain swapping solutions. Learn more from docs and FAQ.

The audit was performed using

- manual analysis of the codebase,
- automated analysis tools,
- simulation of the smart contract,
- analysis of edge test cases

There are 6 points of attention, where 0 are classified as Critical, 1 as High, 2 as Medium, 0 as Low, 1 as Informational, 2 as Best Practices, and 0 as Undetermined. The issues are summarized in Fig. 1.

This document is organized as follows. Section 1 About Cairo Security Clan. Section 2 Disclaimer. Section 3 Executive Summary. Section 4 Summary of Audit. Section 5 Risk Classification. Section 6 Issues by Severity Levels. Section 7 Test Evaluation.

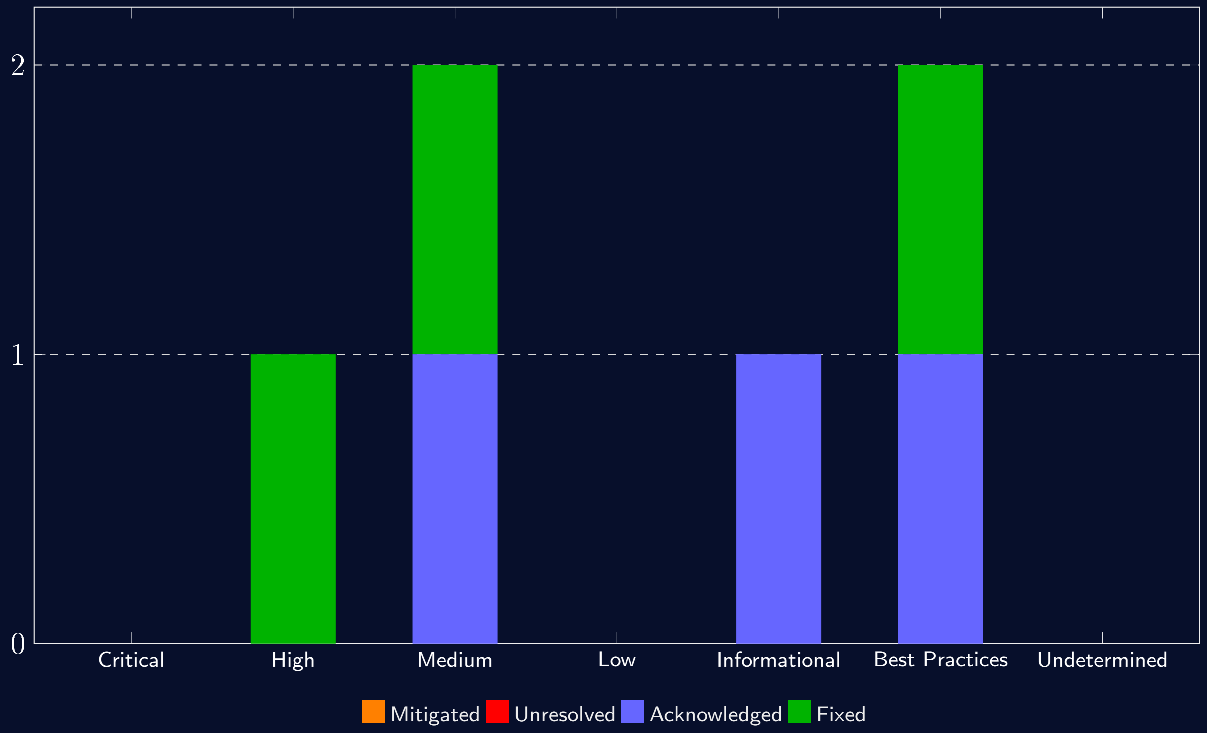

Fig 1: Distribution of issues: Critical (0), **High** (1), **Medium** (2), **Low** (0), **Informational** (1), **Best Practices** (2), **Undetermined** (0). **Distribution of status: Fixed** (3), **Acknowledged** (3), **Mitigated** (0), **Unresolved** (0).

4 Summary of Audit

Audit Type	Security Review
Solidity Version	0.8.28
Final Report	25/07/2025
Repository	atomiqlabs/atomiq-contracts-evm
Initial Commit Hash	d4e5f456d84a6eb1817df4a313c3c5264224e2f0
Final Commit Hash	e416d55e713c9e9da2daafe77e99df83b7c08f36
Documentation	Website documentation
Test Suite Assessment	High

4.1 Scoped Files

	Contracts
1	contracts/btc_nonced_output_claim_handler/BitcoinNoncedOutputClaimHandler.sol
2	contracts/btc_output_claim_handler/BitcoinOutputClaimHandler.sol
3	contracts/btc_relay/state/BtcRelayState.sol
4	contracts/btc_relay/structs/CompactBlockHeader.sol
5	contracts/btc_relay/structs/StoredBlockHeader.sol
6	contracts/btc_relay/utils/Difficulty.sol
7	contracts/btc_relay/utils/Nbits.sol
8	contracts/btc_relay/BtcRelay.sol
9	contracts/btc_relay/Constants.sol
10	contracts/btc_relay/Events.sol
11	contracts/btc_txid_claim_handler/BitcoinTxIdClaimHandler.sol
12	contracts/btc_utils/BitcoinMerkleTree.sol
13	contracts/btc_utils/BitcoinTx.sol
14	contracts/btc_utils/Endianness.sol
15	contracts/common/IClaimHandler.sol
16	contracts/common/IRefundHandler.sol
17	contracts/escrow_manager/components/EIP712Sighash.sol
18	contracts/escrow_manager/components/EscrowStorage.sol
19	contracts/escrow_manager/components/LpVault.sol
20	contracts/escrow_manager/components/ReputationTracker.sol
21	contracts/escrow_manager/state/EscrowState.sol
22	contracts/escrow_manager/state/ReputationState.sol
23	contracts/escrow_manager/structs/Escrow.sol
24	contracts/escrow_manager/EscrowManager.sol
25	contracts/escrow_manager/Events.sol
26	contracts/execution_contract/structs/Execution.sol
27	contracts/execution_contract/Events.sol
28	contracts/execution_contract/ExecutionContract.sol
29	contracts/execution_proxy/structs/ContractCall.sol
30	contracts/execution_proxy/structs/ExecutionAction.sol
31	contracts/execution_proxy/ExecutionProxy.sol
32	contracts/execution_proxy/Executor.sol
33	contracts/hashlock_claim_handler/HashlockClaimHandler.sol
34	contracts/spv_swap_vault/state/SpvVaultState.sol
35	contracts/spv_swap_vault/structs/BitcoinVaultTransactionData.sol
36	contracts/spv_swap_vault/structs/SpvVaultParameters.sol
37	contracts/spv_swap_vault/Events.sol
38	contracts/spv_swap_vault/SpvVaultManager.sol
39	contracts/spv_swap_vault/Utils.sol
40	contracts/timelock_refund_handler/TimelockRefundHandler.sol
41	contracts/transfer_utils/TransferUtils.sol
42	contracts/utils/MathUtils.sol

4.2 Issues

	Findings	Severity	Update
1	Front-running issue in `ExecutionContract.create()` allows blocking of legitimate SpvSwapVault transactions	High	Fixed
2	Success action execution failures do not revert the claim transaction	Medium	Acknowledged
3	Attacker can intentionally limit gas to sabotage the success action execution	Medium	Fixed
4	Claimer cannot initialize escrow with native ETH	Informational	Acknowledged
5	Misleading parameter name `src` in `_payOut()` function	Best Practices	Fixed
6	Deposit function should use safe arithmetic operations like withdraw function	Best Practices	Acknowledged

5 Risk Classification

The risk rating methodology used by Cairo Security Clan follows the principles established by the CVSS risk rating methodology. The severity of each finding is determined by two factors: **Likelihood** and **Impact**.

Likelihood measures how likely an attacker will uncover and exploit the finding. This factor will be one of the following values:

a) **High**: The issue is trivial to exploit and has no specific conditions that need to be met;

b) **Medium**: The issue is moderately complex and may have some conditions that need to be met;

c) **Low**: The issue is very complex and requires very specific conditions to be met.

When defining the likelihood of a finding, other factors are also considered. These can include but are not limited to Motive, opportunity, exploit accessibility, ease of discovery, and ease of exploit.

Impact is a measure of the damage that may be caused if an attacker exploits the finding. This factor will be one of the following values:

a) **High**: The issue can cause significant damage such as loss of funds or the protocol entering an unrecoverable state;

b) **Medium**: The issue can cause moderate damage such as impacts that only affect a small group of users or only a particular part of the protocol;

c) **Low**: The issue can cause little to no damage such as bugs that are easily recoverable or cause unexpected interactions that cause minor inconveniences.

When defining the impact of a finding other factors are also considered. These can include but are not limited to Data/state integrity, loss of availability, financial loss, and reputation damage. After defining the likelihood and impact of an issue, the severity can be determined according to the table below.

		Likelihood		
		High	**Medium**	**Low**
Impact	**High**	Critical	High	Medium
	Medium	High	Medium	Low
	Low	Medium	Low	Info/Best Practices

To address issues that do not fit a High/Medium/Low severity, Cairo Security Clan also uses three more finding severities: **Informational**, **Best Practices** and **Gas**

a) **Informational** findings do not pose any risk to the application, but they carry some information that the audit team intends to formally pass to the client;

b) **Best Practice** findings are used when some piece of code does not conform with smart contract development best practices;

c) **Gas** findings are used when some piece of code uses more gas than it should be or have some functions that can be removed to save gas.

6 Issues by Severity Levels

6.1 High

6.1.1 Front-running issue in `ExecutionContract.create()` allows blocking of legitimate SpvSwapVault transactions

File(s): contracts/execution_contract/ExecutionContract.sol

Description: The `ExecutionContract.create()` function is permissionless and allows anyone to create an execution commitment for any `owner` with any `salt`. When users claim or front in `SpvVaultManager` with execution actions, the contract calls `ExecutionContract.create()` using `data.recipient` as the owner and `btcTxHash` as the salt. Attackers can frontrun these transactions by calling `create()` with the same parameters, causing legitimate transactions to revert with "create: Already initiated" error. This completely blocks users from accessing their funds through the intended execution mechanism.

```
function create(address owner, bytes32 salt, Execution calldata execution) external payable {
    //Make sure execution not yet initialized
    require(_executionCommitments[owner][salt]==bytes32(0x0), "create:␣Already␣initiated");

    //Commit execution
    bytes32 executionHash = execution.hash();
    _executionCommitments[owner][salt] = executionHash;

    //Transfer token amount to the contract
    uint256 totalAmount = execution.amount + execution.executionFee;
    TransferUtils.transferIn(execution.token, msg.sender, totalAmount);

    //Emit event
    emit Events.ExecutionCreated(owner, salt, executionHash);
}
```

Recommendation(s): Consider modifying the salt generation in ExecutionContract to include additional information like msg.sender. For example, using the hash value of `salt` and `msg.value`.

Status: Fixed

Update from the client: Fixed in e416d55, using the recommended method of hashing the user-provided salt with the `msg.sender`, this making the actual salt unique on a per-sender basis. Also fixed in the Cairo contracts in 31e0deb using the same method.

6.2 Medium

6.2.1 Success action execution failures do not revert the claim transaction

File(s): contracts/escrow_manager/EscrowManager.sol

Description: When claiming an escrow with a success action, the `claimWithSuccessAction()` function executes the success action immediately via an external call to the ExecutionProxy contract. If the success action execution fails, the transaction does not revert. Instead, it only emits an `ExecutionError` event:

```
function claimWithSuccessAction(EscrowData calldata escrow, bytes calldata witness, ExecutionAction calldata
     successAction) external {
    require(escrow.successActionCommitment==successAction.hash(), "claim:␣invalid␣success␣action");
    bytes32 escrowHash = _claimWithoutPayout(escrow, witness);

    //Execute through execution proxy instead of paying out
    (bool success, bytes memory errorResult) = _execute(escrow.token, escrow.amount, successAction, escrow.
     claimer);
    if(!success) emit Events.ExecutionError(escrowHash, errorResult);
}
```

This means that even if the success action fails (due to insufficient gas, contract errors, or other execution issues), the escrow claim is still processed and the funds are transferred to the execution proxy. The success action is effectively lost with no mechanism to retry or recover it.

Comparing with Cairo version of this protocol, the claim action will queue the success action for execution in the ExecutionContract instead of executing it immediately during claiming.

```
if escrow.success_action.is_some() {
    //Transfer funds to execution contract in case success action was defined
    self._to_execution_contract(escrow.claimer, escrow.token, escrow.amount, escrow.is_pay_out(), escrow.
     success_action.unwrap(), escrow_hash);
} else {
    //Pay out directly to the claimer
    self._pay_out(escrow.claimer, escrow.token, escrow.amount, escrow.is_pay_out());
}

// [...]
fn _to_execution_contract(ref self: ContractState, dst: ContractAddress, token: ContractAddress, amount: u256,
     pay_out: bool, escrow_execution: EscrowExecution, escrow_hash: felt252) {
    if amount < escrow_execution.fee {
        //Fee is larger than the full amount, just pay out as a fallback
        self._pay_out(dst, token, amount, pay_out);
        return;
    }

    let execution_contract = IExecutionContractDispatcher{contract_address: self.execution_contract.read()};

    erc20_utils::approve(token, execution_contract.contract_address, amount);
    execution_contract.create(dst, escrow_hash, token, amount - escrow_execution.fee, escrow_execution.fee,
     escrow_execution.hash, escrow_execution.expiry);
}
```

Recommendation(s): Consider implementing a queuing mechanism similar to the Cairo version, where success actions are queued for later execution in the Execution Contract.

Status: Acknowledged

Update from the client: Better clarified in the code at commit 31a6152.

This is the expected behavior, when the success action execution fails the funds are just transfered to the user directly. Transfering to execution contract (like done in Cairo contracts) would have the same effect, there it would also just transfer funds directly to the user in case the call fails.

The reason that execution contract is not used in EVM but is used in Cairo, is that outside call error handling in Cairo is limited and you cannot:

- limit gas, making it possible for the calls to always run out of gas
- rely on the call to not revert the whole transaction, if the destination contract doesn't exist the call to it will revert the whole transaction instead of just throwing an error

In both of these cases the claiming of the escrow is blocked and claimer might loose funds by being unable to claim before timeout.

Compared to EVM where you can limit gas, and catch every possible error case in the code (contract revert, contract panic, contract doesn't exist, out of gas) - making sure that a failing call doesn't revert the whole claim transaction and doesn't block the claim of the escrow.

6.2.2 Attacker can intentionally limit gas to sabotage the success action execution

File(s): contracts/execution_proxy/Executor.sol

Description: When initializing an escrow, a success action can be specified to execute automatically upon a successful claim. This is performed via an external call to an `ExecutionProxy` contract. If no custom gas limit is set (`executionAction.gasLimit == 0`), the contract defaults to using the full remaining gas (`gasleft()`):

```
//Try to execute calls
(success, callError) = address(executionProxy).call{gas: executionAction.gasLimit==0 ? gasleft() :
     executionAction.gasLimit}(
    abi.encodeWithSelector(ExecutionProxy.execute.selector, executionAction.calls)
);
```

However, this call can be triggered by any external actor via `claimWithSuccessAction()`, including a malicious one. An attacker could deliberately invoke the function with insufficient gas, causing the success action to fail during execution.

Additionally, failure of the success action does not revert the transaction. Instead, it emits an `ExecutionError` event:

```
function claimWithSuccessAction(EscrowData calldata escrow, bytes calldata witness, ExecutionAction calldata
     successAction) external {
    require(escrow.successActionCommitment==successAction.hash(), "claim:␣invalid␣success␣action");
    bytes32 escrowHash = _claimWithoutPayout(escrow, witness);

    //Execute through execution proxy instead of paying out
    (bool success, bytes memory errorResult) = _execute(escrow.token, escrow.amount, successAction, escrow.
     claimer);
    if(!success) emit Events.ExecutionError(escrowHash, errorResult);
}
```

Because the failure is non-reverting and there's no retry mechanism, the success action is effectively lost once a low-gas claim is executed.

Recommendation(s): Restrict who can call `claimWithSuccessAction()` to only the `escrow.claimer` if there is no gas limit specified.

Status: Fixed

Update from the client: Fixed in 4d3f6d3 by disabling the feature to forward `gasleft()` when `gasLimit` is 0 and instead immediately failing when `gasLimit` equals 0

6.3 Informational

6.3.1 Claimer cannot initialize escrow with native ETH

File(s): contracts/escrow_manager/EscrowManager.sol

Description: The escrow manager contract prevents a claimer from initializing an escrow using native ETH due to a sender verification requirement in the `transferIn()` function.

Specifically, when transferring native ETH (`token == address(0)`), the following check is enforced:

```
function transferIn(address token, address src, uint256 amount) internal {
    if(token==address(0x0)) {
        //Native token transfer
        require(src==msg.sender, "transferIn:␣sender␣not␣src");
        require(msg.value >= amount, "transferIn:␣value␣too␣low");
    } else {
        // [...]
    }
}
```

This check fails if the claimer (i.e. not the offerer) tries to initialize the escrow, because the `src` address passed into `transferIn()` is set to `escrow.offerer`, while `msg.sender` would be the claimer.

Here's the relevant snippet from `initialize()`:

```
if(escrow.depositToken==escrow.token && escrow.isPayIn() && msg.sender==escrow.offerer) {
    //Transfer funds in one go
    _payIn(escrow.offerer, escrow.token, escrow.amount + depositAmount, true);
} else {
    //Transfer funds separatelly
    if(depositAmount > 0) TransferUtils.transferIn(escrow.depositToken, msg.sender, depositAmount);
    _payIn(escrow.offerer, escrow.token, escrow.amount, escrow.isPayIn());
}
```

In the else branch, `_payIn()` is called with `src = escrow.offerer`. If native ETH is used, and the caller is not `escrow.offerer`, the `transferIn()` call will revert.

Recommendation(s): Consider reviewing the logic to support the claimer to initialize escrow with native ETH. Otherwise, document this behavior clearly in the codebase.

Status: Acknowledged

Update from the client: Better clarification is provided in the commit 67fa15a, clearly stating that initializing an escrow by the claimer when `payIn = true` and using native token (ETH) will always fail.

6.4 Best Practices

6.4.1 Misleading parameter name src in _payOut() function

File(s): contracts/escrow_manager/EscrowManager.sol

Description: The _payOut() function uses the parameter name src to represent the recipient of funds, which is misleading and inconsistent with typical naming conventions. In payout operations, the address receiving the funds is usually referred to as dst (destination), while src (source) implies a sender.

```
function _payOut(address src, address token, uint256 amount, bool payOut) internal {
    if (payOut) {
        TransferUtils.transferOut(token, src, amount);
    } else {
        _LpVault_transferOut(token, src, amount);
    }
}
```

This naming may confuse developers reviewing or maintaining the code, especially since other functions (like transferOut()) correctly use dst to refer to the recipient.

Recommendation(s): Rename the src parameter to dst (or recipient) to better reflect its role as the payout destination.

Status: Fixed

Update from the client: Fixed in 1fa48ab

6.4.2 Deposit function should use safe arithmetic operations like withdraw function

File(s): contracts/spv_swap_vault/state/SpvVaultState.sol

Description: The deposit() function in SpvVaultState does not implement the same safety checks as the withdraw() function. While withdraw() uses safe arithmetic operations to prevent overflow, deposit() relies on unchecked operations:

Withdraw function (lines 80-85) - Uses safe operations:

```
//Make sure subtraction doesn't overflow
(bool token0AmountSuccess, uint64 token0Amount) = _token0Amount.checkedSubUint64(rawAmount0);
if(!token0AmountSuccess) return (false, _withdrawCount, "withdraw: amount 0");
(bool token1AmountSuccess, uint64 token1Amount) = _token1Amount.checkedSubUint64(rawAmount1);
if(!token1AmountSuccess) return (false, _withdrawCount, "withdraw: amount 1");
withdrawCount = _withdrawCount.saturatingAddOneUint32();
```

Deposit function (lines 123-125) - Uses unsafe operations:

```
_token0Amount += rawAmount0;
_token1Amount += rawAmount1;
depositCount = ++_depositCount;
```

The deposit function uses unchecked addition (+=) which can overflow when _token0Amount + rawAmount0 or _token1Amount + rawAmount1 exceeds type(uint64).max. Similarly, ++_depositCount can overflow when it reaches type(uint32).max.

Recommendation(s): Consider using safe arithmetic functions with proper error handling, similar to the withdraw() function, to avoid unexpected overflows.

Status: Acknowledged

Update from the client: Better clarified that this is the expected behavior at commit 7de2226

It is critical that we don't panic in the withdraw function (since that could lead to funds being locked up), but panicing in deposit is fine because that couldn't lead to loss of funds.

7 Test Compilation Evaluation

7.1 Compilation Output

```
 npx hardhat compile
Generating typings for: 103 artifacts in dir: typechain-types for target: ethers-v6
Successfully generated 300 typings!
Compiled 94 Solidity files successfully (evm target: cancun).
```

7.2 Tests Output

```
  BtcRelay
    [PASS] Valid constructor (514ms)
    [PASS] Invalid block PoW (blockhash not lower than target) (675ms)
    [PASS] Invalid block nbits (not same as last block)
    [PASS] Invalid block nbits at difficulty adjustment (invalidly computed)
    [PASS] Invalid block previous blockhash (321ms)
    [PASS] Invalid block median timestamps (56ms)
    [PASS] Invalid block future timestamp
[submitMainAndAssert] Gas usage, num headers: 20 gas used: 572328 gas per header: 28616
    [PASS] Valid main chain (383ms)
    [PASS] Invalid main chain, stored blockheader not committed (78ms)
[submitMainAndAssert] Gas usage, num headers: 4 gas used: 139280 gas per header: 34820
    [PASS] Invalid main chain, stored header is not the tip (97ms)
[submitMainAndAssert] Gas usage, num headers: 20 gas used: 572268 gas per header: 28613
    [PASS] Invalid short fork, stored header not in main chain (415ms)
[submitMainAndAssert] Gas usage, num headers: 20 gas used: 572292 gas per header: 28615
    [PASS] Invalid long fork, stored header not in main chain (424ms)
[submitMainAndAssert] Gas usage, num headers: 20 gas used: 572304 gas per header: 28615
[submitLongForkAndAssert] Gas usage, num headers: 4 gas used: 183198 gas per header: 45800
    [PASS] Invalid long fork, stored header not in fork state (416ms)
[submitMainAndAssert] Gas usage, num headers: 20 gas used: 572292 gas per header: 28615
[submitLongForkAndAssert] Gas usage, num headers: 4 gas used: 183210 gas per header: 45803
    [PASS] Invalid long fork, stored header not fork state tip (436ms)
[submitMainAndAssert] Gas usage, num headers: 135 gas used: 3685107 gas per header: 27297
[submitShortForkAndAssert] Gas usage, num headers: 134 gas used: 1371293 gas per header: 10234
    [PASS] Invalid main chain, re-org & try to build on top of now future blockheight (16080ms)
[submitMainAndAssert] Gas usage, num headers: 135 gas used: 3684975 gas per header: 27296
[submitShortForkAndAssert] Gas usage, num headers: 134 gas used: 1371317 gas per header: 10234
    [PASS] Invalid short fork, re-org & try to fork again from now future blockheight (16469ms)
[submitMainAndAssert] Gas usage, num headers: 135 gas used: 3684927 gas per header: 27296
[submitShortForkAndAssert] Gas usage, num headers: 134 gas used: 1371341 gas per header: 10234
    [PASS] Invalid long fork, re-org & try to fork again from now future blockheight (15775ms)
[submitMainAndAssert] Gas usage, num headers: 20 gas used: 572328 gas per header: 28616
[submitShortForkAndAssert] Gas usage, num headers: 20 gas used: 302068 gas per header: 15103
    [PASS] Valid short fork (696ms)
[submitMainAndAssert] Gas usage, num headers: 20 gas used: 572280 gas per header: 28614
[submitLongForkAndAssert] Gas usage, num headers: 20 gas used: 644808 gas per header: 32240
    [PASS] Valid long fork (683ms)
[submitMainAndAssert] Gas usage, num headers: 20 gas used: 572352 gas per header: 28618
[submitLongForkAndAssert] Gas usage, num headers: 4 gas used: 183198 gas per header: 45800
[submitLongForkAndAssert] Gas usage, num headers: 16 gas used: 544608 gas per header: 34038
    [PASS] Valid long fork in 2 txns (697ms)
[submitMainAndAssert] Gas usage, num headers: 135 gas used: 3684963 gas per header: 27296
[submitShortForkAndAssert] Gas usage, num headers: 134 gas used: 1371245 gas per header: 10233
    [PASS] Valid short fork, higher chainwork but shorter chain (15643ms)
[submitMainAndAssert] Gas usage, num headers: 20 gas used: 572268 gas per header: 28613
    [PASS] Invalid short fork, not enough length (345ms)
[submitMainAndAssert] Gas usage, num headers: 20 gas used: 572352 gas per header: 28618
[submitLongForkAndAssert] Gas usage, num headers: 5 gas used: 210816 gas per header: 42163
    [PASS] Invalid long fork, not enough length (384ms)
[submitMainAndAssert] Gas usage, num headers: 123 gas used: 3360399 gas per header: 27320
    [PASS] Invalid short fork, not enough chainwork, but long enough (7362ms)
[submitMainAndAssert] Gas usage, num headers: 123 gas used: 3360483 gas per header: 27321
[submitLongForkAndAssert] Gas usage, num headers: 124 gas used: 3477112 gas per header: 28041
    [PASS] Invalid long fork, not enough chainwork, but long enough (13718ms)

  EscrowManager
    Claim
      [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=false,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=false) (82ms)
      [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=false,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=false)
      [PASS] Valid claim (payOut=false,reputation=true,securityDeposit=false,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=false)
      [PASS] Valid claim (payOut=true,reputation=true,securityDeposit=false,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=false)
      [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=true,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=true,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=true,securityDeposit=true,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=true,securityDeposit=true,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=false,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=false,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=true,securityDeposit=false,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=true,securityDeposit=false,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=true,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=true,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=true,securityDeposit=true,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=true,securityDeposit=true,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=false,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=false,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=true,securityDeposit=false,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=true,securityDeposit=false,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=true,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=true,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=true,securityDeposit=true,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=true,securityDeposit=true,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=false,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=false,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=true,securityDeposit=false,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=true,securityDeposit=false,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=true,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=true,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=true,securityDeposit=true,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=true,securityDeposit=true,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=false,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=true,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=false,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=true,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=true,securityDeposit=false,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=true,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=true,securityDeposit=false,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=true,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=true,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=true,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=true,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=true,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=true,securityDeposit=true,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=true,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=true,securityDeposit=true,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=true,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=false,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=true,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=false,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=true,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=true,securityDeposit=false,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=true,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=true,securityDeposit=false,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=true,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=true,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=true,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=true,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=true,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=true,securityDeposit=true,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=true,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=true,securityDeposit=true,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=true,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=false,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=true,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=false,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=true,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=true,securityDeposit=false,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=true,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=true,securityDeposit=false,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=true,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=true,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=true,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=true,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=true,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=true,securityDeposit=true,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=true,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=true,securityDeposit=true,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=true,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=false,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=true,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=false,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=true,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=true,securityDeposit=false,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=true,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=true,securityDeposit=false,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=true,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=true,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=true,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=true,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=true,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=true,securityDeposit=true,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=true,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=true,securityDeposit=true,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=true,successActionSuccess=false,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=false,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=false,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=false,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=false,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=true,securityDeposit=false,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=false,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=true,securityDeposit=false,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=false,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=true,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=false,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=true,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=false,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=true,securityDeposit=true,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=false,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=true,securityDeposit=true,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=false,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=false,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=false,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=false,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=false,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=true,securityDeposit=false,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=false,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=true,securityDeposit=false,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=false,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=true,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=false,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=true,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=false,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=true,securityDeposit=true,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=false,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=true,securityDeposit=true,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=false,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=false,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=false,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=false,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=false,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=true,securityDeposit=false,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=false,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=true,securityDeposit=false,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=false,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=true,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=false,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=true,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=false,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=true,securityDeposit=true,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=false,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=true,securityDeposit=true,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=false,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=false,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=false,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=false,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=false,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=true,securityDeposit=false,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=false,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=true,securityDeposit=false,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=false,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=true,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=false,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=true,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=false,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=true,securityDeposit=true,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=false,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=true,securityDeposit=true,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=false,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=false,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=true,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=false,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=true,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=true,securityDeposit=false,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=true,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=true,securityDeposit=false,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=true,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=true,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=true,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=true,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=true,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=true,securityDeposit=true,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=true,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=true,securityDeposit=true,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=true,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=false,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=true,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=false,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=true,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=true,securityDeposit=false,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=true,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=true,securityDeposit=false,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=true,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=true,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=true,successActionSuccess=true,successActionError=false)
  [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=true,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=true,successActionSuccess=true,successActionError=false)
```

```
      [PASS] Valid claim (payOut=false,reputation=true,securityDeposit=true,claimerBounty=true,usesNativeToken=
     false,usesNativeTokenForDeposit=true,successActionSuccess=true,successActionError=false)
      [PASS] Valid claim (payOut=true,reputation=true,securityDeposit=true,claimerBounty=true,usesNativeToken=
     false,usesNativeTokenForDeposit=true,successActionSuccess=true,successActionError=false)
      [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=false,claimerBounty=false,usesNativeToken
     =true,usesNativeTokenForDeposit=true,successActionSuccess=true,successActionError=false)
      [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=false,claimerBounty=false,usesNativeToken=
     true,usesNativeTokenForDeposit=true,successActionSuccess=true,successActionError=false)
      [PASS] Valid claim (payOut=false,reputation=true,securityDeposit=false,claimerBounty=false,usesNativeToken=
     true,usesNativeTokenForDeposit=true,successActionSuccess=true,successActionError=false)
      [PASS] Valid claim (payOut=true,reputation=true,securityDeposit=false,claimerBounty=false,usesNativeToken=
     true,usesNativeTokenForDeposit=true,successActionSuccess=true,successActionError=false)
      [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=true,claimerBounty=false,usesNativeToken=
     true,usesNativeTokenForDeposit=true,successActionSuccess=true,successActionError=false)
      [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=true,claimerBounty=false,usesNativeToken=
     true,usesNativeTokenForDeposit=true,successActionSuccess=true,successActionError=false)
      [PASS] Valid claim (payOut=false,reputation=true,securityDeposit=true,claimerBounty=false,usesNativeToken=
     true,usesNativeTokenForDeposit=true,successActionSuccess=true,successActionError=false)
      [PASS] Valid claim (payOut=true,reputation=true,securityDeposit=true,claimerBounty=false,usesNativeToken=
     true,usesNativeTokenForDeposit=true,successActionSuccess=true,successActionError=false)
      [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=false,claimerBounty=true,usesNativeToken=
     true,usesNativeTokenForDeposit=true,successActionSuccess=true,successActionError=false)
      [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=false,claimerBounty=true,usesNativeToken=
     true,usesNativeTokenForDeposit=true,successActionSuccess=true,successActionError=false)
      [PASS] Valid claim (payOut=false,reputation=true,securityDeposit=false,claimerBounty=true,usesNativeToken=
     true,usesNativeTokenForDeposit=true,successActionSuccess=true,successActionError=false)
      [PASS] Valid claim (payOut=true,reputation=true,securityDeposit=false,claimerBounty=true,usesNativeToken=
     true,usesNativeTokenForDeposit=true,successActionSuccess=true,successActionError=false)
      [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=true,claimerBounty=true,usesNativeToken=
     true,usesNativeTokenForDeposit=true,successActionSuccess=true,successActionError=false)
      [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=true,claimerBounty=true,usesNativeToken=
     true,usesNativeTokenForDeposit=true,successActionSuccess=true,successActionError=false)
      [PASS] Valid claim (payOut=false,reputation=true,securityDeposit=true,claimerBounty=true,usesNativeToken=
     true,usesNativeTokenForDeposit=true,successActionSuccess=true,successActionError=false)
      [PASS] Valid claim (payOut=true,reputation=true,securityDeposit=true,claimerBounty=true,usesNativeToken=
     true,usesNativeTokenForDeposit=true,successActionSuccess=true,successActionError=false)
      [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=false,claimerBounty=false,usesNativeToken
     =false,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=true)
      [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=false,claimerBounty=false,usesNativeToken=
     false,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=true)
      [PASS] Valid claim (payOut=false,reputation=true,securityDeposit=false,claimerBounty=false,usesNativeToken=
     false,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=true)
      [PASS] Valid claim (payOut=true,reputation=true,securityDeposit=false,claimerBounty=false,usesNativeToken=
     false,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=true)
      [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=true,claimerBounty=false,usesNativeToken=
     false,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=true)
      [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=true,claimerBounty=false,usesNativeToken=
     false,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=true)
      [PASS] Valid claim (payOut=false,reputation=true,securityDeposit=true,claimerBounty=false,usesNativeToken=
     false,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=true)
      [PASS] Valid claim (payOut=true,reputation=true,securityDeposit=true,claimerBounty=false,usesNativeToken=
     false,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=true)
      [PASS] Valid claim (payOut=false,reputation=false,securityDeposit=false,claimerBounty=true,usesNativeToken=
     false,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=true)
      [PASS] Valid claim (payOut=true,reputation=false,securityDeposit=false,claimerBounty=true,usesNativeToken=
     false,usesNativeTokenForDeposit=false,successActionSuccess=false,successActionError=true)
      [PASS] Valid claim (contract call runs out of gas)
      [PASS] Invalid claim (negative answer from claim handler)
      [PASS] Invalid claim (claim uninitialized escrow)
      [PASS] Invalid claim (claim twice)
      [PASS] Invalid claim (no execution specified, but success action execution attempted)
      [PASS] Invalid claim (success action execution specified, but no success action execution attempted)
    Refund cooperative
      [PASS] Valid cooperative refund (payIn=false,reputation=false,securityDeposit=false,claimerBounty=false,
     usesNativeToken=false,usesNativeTokenForDeposit=false)
      [PASS] Valid cooperative refund (payIn=true,reputation=false,securityDeposit=false,claimerBounty=false,
     usesNativeToken=false,usesNativeTokenForDeposit=false)
      [PASS] Valid cooperative refund (payIn=false,reputation=true,securityDeposit=false,claimerBounty=false,
     usesNativeToken=false,usesNativeTokenForDeposit=false)
```

```
  [PASS] Valid cooperative refund (payIn=true,reputation=true,securityDeposit=false,claimerBounty=false,
usesNativeToken=false,usesNativeTokenForDeposit=false)
  [PASS] Valid cooperative refund (payIn=false,reputation=false,securityDeposit=true,claimerBounty=false,
usesNativeToken=false,usesNativeTokenForDeposit=false)
  [PASS] Valid cooperative refund (payIn=true,reputation=false,securityDeposit=true,claimerBounty=false,
usesNativeToken=false,usesNativeTokenForDeposit=false)
  [PASS] Valid cooperative refund (payIn=false,reputation=true,securityDeposit=true,claimerBounty=false,
usesNativeToken=false,usesNativeTokenForDeposit=false)
  [PASS] Valid cooperative refund (payIn=true,reputation=true,securityDeposit=true,claimerBounty=false,
usesNativeToken=false,usesNativeTokenForDeposit=false)
  [PASS] Valid cooperative refund (payIn=false,reputation=false,securityDeposit=false,claimerBounty=true,
usesNativeToken=false,usesNativeTokenForDeposit=false)
  [PASS] Valid cooperative refund (payIn=true,reputation=false,securityDeposit=false,claimerBounty=true,
usesNativeToken=false,usesNativeTokenForDeposit=false)
  [PASS] Valid cooperative refund (payIn=false,reputation=true,securityDeposit=false,claimerBounty=true,
usesNativeToken=false,usesNativeTokenForDeposit=false)
  [PASS] Valid cooperative refund (payIn=true,reputation=true,securityDeposit=false,claimerBounty=true,
usesNativeToken=false,usesNativeTokenForDeposit=false)
  [PASS] Valid cooperative refund (payIn=false,reputation=false,securityDeposit=true,claimerBounty=true,
usesNativeToken=false,usesNativeTokenForDeposit=false)
  [PASS] Valid cooperative refund (payIn=true,reputation=false,securityDeposit=true,claimerBounty=true,
usesNativeToken=false,usesNativeTokenForDeposit=false)
  [PASS] Valid cooperative refund (payIn=true,reputation=false,securityDeposit=false,claimerBounty=false,
usesNativeToken=false,usesNativeTokenForDeposit=true)
  [PASS] Valid cooperative refund (payIn=false,reputation=true,securityDeposit=false,claimerBounty=false,
usesNativeToken=false,usesNativeTokenForDeposit=true)
  [PASS] Valid cooperative refund (payIn=true,reputation=true,securityDeposit=false,claimerBounty=false,
usesNativeToken=false,usesNativeTokenForDeposit=true)
  [PASS] Valid cooperative refund (payIn=false,reputation=false,securityDeposit=true,claimerBounty=false,
usesNativeToken=false,usesNativeTokenForDeposit=true)
  [PASS] Valid cooperative refund (payIn=true,reputation=false,securityDeposit=true,claimerBounty=false,
usesNativeToken=false,usesNativeTokenForDeposit=true)
  [PASS] Valid cooperative refund (payIn=false,reputation=true,securityDeposit=true,claimerBounty=false,
usesNativeToken=false,usesNativeTokenForDeposit=true)
  [PASS] Valid cooperative refund (payIn=true,reputation=true,securityDeposit=true,claimerBounty=false,
usesNativeToken=false,usesNativeTokenForDeposit=true)
  [PASS] Valid cooperative refund (payIn=false,reputation=false,securityDeposit=false,claimerBounty=true,
usesNativeToken=false,usesNativeTokenForDeposit=true)
  [PASS] Valid cooperative refund (payIn=true,reputation=false,securityDeposit=false,claimerBounty=true,
usesNativeToken=false,usesNativeTokenForDeposit=true)
  [PASS] Valid cooperative refund (payIn=false,reputation=true,securityDeposit=false,claimerBounty=true,
usesNativeToken=false,usesNativeTokenForDeposit=true)
  [PASS] Valid cooperative refund (payIn=true,reputation=true,securityDeposit=false,claimerBounty=true,
usesNativeToken=false,usesNativeTokenForDeposit=true)
  [PASS] Valid cooperative refund (payIn=false,reputation=false,securityDeposit=true,claimerBounty=true,
usesNativeToken=false,usesNativeTokenForDeposit=true)
  [PASS] Valid cooperative refund (payIn=true,reputation=false,securityDeposit=true,claimerBounty=true,
usesNativeToken=false,usesNativeTokenForDeposit=true)
  [PASS] Valid cooperative refund (payIn=false,reputation=true,securityDeposit=true,claimerBounty=true,
usesNativeToken=false,usesNativeTokenForDeposit=true)
  [PASS] Valid cooperative refund (payIn=true,reputation=true,securityDeposit=true,claimerBounty=true,
usesNativeToken=false,usesNativeTokenForDeposit=true)
  [PASS] Valid cooperative refund (payIn=false,reputation=false,securityDeposit=false,claimerBounty=false,
usesNativeToken=true,usesNativeTokenForDeposit=true)
  [PASS] Valid cooperative refund (payIn=true,reputation=false,securityDeposit=false,claimerBounty=false,
usesNativeToken=true,usesNativeTokenForDeposit=true)
  [PASS] Valid cooperative refund (payIn=false,reputation=true,securityDeposit=false,claimerBounty=false,
usesNativeToken=true,usesNativeTokenForDeposit=true)
  [PASS] Valid cooperative refund (payIn=true,reputation=true,securityDeposit=false,claimerBounty=false,
usesNativeToken=true,usesNativeTokenForDeposit=true)
  [PASS] Valid cooperative refund (payIn=false,reputation=false,securityDeposit=true,claimerBounty=false,
usesNativeToken=true,usesNativeTokenForDeposit=true)
  [PASS] Valid cooperative refund (payIn=true,reputation=false,securityDeposit=true,claimerBounty=false,
usesNativeToken=true,usesNativeTokenForDeposit=true)
  [PASS] Valid cooperative refund (payIn=false,reputation=true,securityDeposit=true,claimerBounty=false,
usesNativeToken=true,usesNativeTokenForDeposit=true)
  [PASS] Valid cooperative refund (payIn=true,reputation=true,securityDeposit=true,claimerBounty=false,
usesNativeToken=true,usesNativeTokenForDeposit=true)
  [PASS] Valid cooperative refund (payIn=false,reputation=false,securityDeposit=false,claimerBounty=true,
usesNativeToken=true,usesNativeTokenForDeposit=true)
```

```
    [PASS] Valid cooperative refund (payIn=true,reputation=false,securityDeposit=false,claimerBounty=true,
  usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid cooperative refund (payIn=false,reputation=true,securityDeposit=false,claimerBounty=true,
  usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid cooperative refund (payIn=true,reputation=true,securityDeposit=false,claimerBounty=true,
  usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid cooperative refund (payIn=false,reputation=false,securityDeposit=true,claimerBounty=true,
  usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid cooperative refund (payIn=true,reputation=false,securityDeposit=true,claimerBounty=true,
  usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid cooperative refund (payIn=false,reputation=true,securityDeposit=true,claimerBounty=true,
  usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid cooperative refund (payIn=true,reputation=true,securityDeposit=true,claimerBounty=true,
  usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid refund cooperative (claimer is an erc1271 account)
    [PASS] Invalid refund cooperative (not initialized)
    [PASS] Invalid refund cooperative (try to refund twice)
    [PASS] Invalid refund cooperative (timed out refund authorization)
    [PASS] Invalid refund cooperative (sign different timeout)
    [PASS] Invalid refund cooperative (sign random swap hash)
    [PASS] Invalid refund cooperative (wrong signer)
  Refund
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=false,reputation=false,
  securityDeposit=false,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=true,reputation=false,
  securityDeposit=false,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=false,reputation=true,
  securityDeposit=false,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=true,reputation=true,
  securityDeposit=false,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=false,reputation=false,
  securityDeposit=true,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=true,payIn=false,reputation=false,
  securityDeposit=true,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=true,reputation=false,
  securityDeposit=true,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=true,payIn=true,reputation=false,
  securityDeposit=true,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=false,reputation=true,
  securityDeposit=true,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=true,payIn=false,reputation=true,
  securityDeposit=true,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=true,reputation=true,
  securityDeposit=true,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=true,payIn=true,reputation=true,securityDeposit
  =true,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=false,reputation=false,
  securityDeposit=false,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=true,payIn=false,reputation=false,
  securityDeposit=false,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=true,reputation=false,
  securityDeposit=false,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=true,payIn=true,reputation=false,
  securityDeposit=false,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=false,reputation=true,
  securityDeposit=false,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=true,payIn=false,reputation=true,
  securityDeposit=false,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=true,reputation=true,
  securityDeposit=false,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=true,payIn=true,reputation=true,securityDeposit
  =true,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=false)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=false,reputation=false,
  securityDeposit=false,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=true,reputation=false,
  securityDeposit=false,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=false,reputation=true,
  securityDeposit=false,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=true,reputation=true,
  securityDeposit=false,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=true)
```

```
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=false,reputation=false,securityDeposit=true,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=true,payIn=false,reputation=false,securityDeposit=true,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=true,reputation=false,securityDeposit=true,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=true,payIn=true,reputation=false,securityDeposit=true,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=false,reputation=true,securityDeposit=true,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=true,payIn=false,reputation=true,securityDeposit=true,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=true,reputation=true,securityDeposit=true,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=true,payIn=true,reputation=true,securityDeposit=true,claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=false,reputation=false,securityDeposit=false,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=true,payIn=false,reputation=false,securityDeposit=false,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=true,reputation=false,securityDeposit=false,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=true,payIn=true,reputation=false,securityDeposit=false,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=false,reputation=true,securityDeposit=false,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=true,payIn=false,reputation=true,securityDeposit=false,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=true,reputation=true,securityDeposit=false,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=true,payIn=true,reputation=true,securityDeposit=false,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=false,reputation=false,securityDeposit=true,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=true,payIn=false,reputation=false,securityDeposit=true,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=true,reputation=false,securityDeposit=true,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=true,payIn=true,reputation=false,securityDeposit=true,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=false,reputation=true,securityDeposit=true,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=true,payIn=false,reputation=true,securityDeposit=true,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=true,reputation=true,securityDeposit=true,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=true,payIn=true,reputation=true,securityDeposit=true,claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=false,reputation=false,securityDeposit=false,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=true,reputation=false,securityDeposit=false,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=false,reputation=true,securityDeposit=false,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=true,reputation=true,securityDeposit=false,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=false,reputation=false,securityDeposit=true,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=true,payIn=false,reputation=false,securityDeposit=true,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=true,reputation=false,securityDeposit=true,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=true,payIn=true,reputation=false,securityDeposit=true,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=false,reputation=true,securityDeposit=true,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=true,payIn=false,reputation=true,securityDeposit=true,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=true,reputation=true,securityDeposit=true,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=true,payIn=true,reputation=true,securityDeposit
  =true,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=false,reputation=false,
  securityDeposit=false,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=true,payIn=false,reputation=false,
  securityDeposit=false,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=true,reputation=false,
  securityDeposit=false,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=true,payIn=true,reputation=false,
  securityDeposit=false,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=false,reputation=true,
  securityDeposit=false,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=true,payIn=false,reputation=true,
  securityDeposit=false,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=true,reputation=true,
  securityDeposit=false,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=true,payIn=true,reputation=true,securityDeposit
  =false,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=false,reputation=false,
  securityDeposit=true,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=true,payIn=false,reputation=false,
  securityDeposit=true,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=true,reputation=false,
  securityDeposit=true,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=true,payIn=true,reputation=false,
  securityDeposit=true,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=false,reputation=true,
  securityDeposit=true,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=true,payIn=false,reputation=true,
  securityDeposit=true,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=false,payIn=true,reputation=true,
  securityDeposit=true,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid refund (securityDepositLargerThanClaimerBounty=true,payIn=true,reputation=true,securityDeposit
  =true,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Invalid refund (negative answer from refund handler)
    [PASS] Invalid refund (refund uninitialized escrow)
    [PASS] Invalid refund (refund twice)
  Initialize
    [PASS] Valid initialize (senderClaimer=false,payIn=false,reputation=false,securityDeposit=false,
  claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid initialize (senderClaimer=true,payIn=false,reputation=false,securityDeposit=false,
  claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid initialize (senderClaimer=false,payIn=true,reputation=false,securityDeposit=false,
  claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid initialize (senderClaimer=true,payIn=true,reputation=false,securityDeposit=false,claimerBounty
  =false,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid initialize (senderClaimer=false,payIn=false,reputation=true,securityDeposit=false,
  claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid initialize (senderClaimer=true,payIn=false,reputation=true,securityDeposit=false,claimerBounty
  =false,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid initialize (senderClaimer=false,payIn=true,reputation=true,securityDeposit=false,claimerBounty
  =false,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid initialize (senderClaimer=true,payIn=true,reputation=true,securityDeposit=false,claimerBounty=
  false,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid initialize (senderClaimer=false,payIn=false,reputation=false,securityDeposit=true,
  claimerBounty=false,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid initialize (senderClaimer=true,payIn=false,reputation=false,securityDeposit=true,claimerBounty
  =false,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid initialize (senderClaimer=false,payIn=true,reputation=false,securityDeposit=true,claimerBounty
  =false,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid initialize (senderClaimer=true,payIn=true,reputation=false,securityDeposit=true,claimerBounty=
  false,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid initialize (senderClaimer=false,payIn=false,reputation=true,securityDeposit=true,claimerBounty
  =false,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid initialize (senderClaimer=true,payIn=false,reputation=true,securityDeposit=true,claimerBounty=
  false,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid initialize (senderClaimer=false,payIn=true,reputation=true,securityDeposit=true,claimerBounty=
  false,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid initialize (senderClaimer=true,payIn=true,reputation=true,securityDeposit=true,claimerBounty=
  false,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid initialize (senderClaimer=false,payIn=false,reputation=false,securityDeposit=false,
  claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid initialize (senderClaimer=true,payIn=false,reputation=false,securityDeposit=false,
  claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid initialize (senderClaimer=false,payIn=true,reputation=false,securityDeposit=false,
  claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid initialize (senderClaimer=true,payIn=true,reputation=false,securityDeposit=false,claimerBounty
  =true,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid initialize (senderClaimer=false,payIn=false,reputation=true,securityDeposit=false,
  claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid initialize (senderClaimer=true,payIn=false,reputation=true,securityDeposit=false,claimerBounty
  =true,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid initialize (senderClaimer=false,payIn=true,reputation=true,securityDeposit=false,claimerBounty
  =true,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid initialize (senderClaimer=true,payIn=true,reputation=true,securityDeposit=false,claimerBounty=
  true,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid initialize (senderClaimer=false,payIn=false,reputation=false,securityDeposit=true,
  claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid initialize (senderClaimer=true,payIn=false,reputation=false,securityDeposit=true,claimerBounty
  =true,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid initialize (senderClaimer=false,payIn=true,reputation=false,securityDeposit=true,claimerBounty
  =true,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid initialize (senderClaimer=true,payIn=true,reputation=false,securityDeposit=true,claimerBounty=
  true,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid initialize (senderClaimer=false,payIn=false,reputation=true,securityDeposit=true,claimerBounty
  =true,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid initialize (senderClaimer=true,payIn=false,reputation=true,securityDeposit=true,claimerBounty=
  true,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid initialize (senderClaimer=false,payIn=true,reputation=true,securityDeposit=true,claimerBounty=
  true,usesNativeToken=false,usesNativeTokenForDeposit=false)
    [PASS] Valid initialize (senderClaimer=true,payIn=false,reputation=false,securityDeposit=false,
  claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid initialize (senderClaimer=false,payIn=true,reputation=false,securityDeposit=false,
  claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid initialize (senderClaimer=true,payIn=true,reputation=false,securityDeposit=false,claimerBounty
  =true,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid initialize (senderClaimer=false,payIn=false,reputation=true,securityDeposit=false,
  claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid initialize (senderClaimer=true,payIn=false,reputation=true,securityDeposit=false,claimerBounty
  =true,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid initialize (senderClaimer=false,payIn=true,reputation=true,securityDeposit=false,claimerBounty
  =true,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid initialize (senderClaimer=true,payIn=true,reputation=true,securityDeposit=false,claimerBounty=
  true,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid initialize (senderClaimer=false,payIn=false,reputation=false,securityDeposit=true,
  claimerBounty=true,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid initialize (senderClaimer=true,payIn=false,reputation=false,securityDeposit=true,claimerBounty
  =true,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid initialize (senderClaimer=false,payIn=true,reputation=false,securityDeposit=true,claimerBounty
  =true,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid initialize (senderClaimer=true,payIn=true,reputation=false,securityDeposit=true,claimerBounty=
  true,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid initialize (senderClaimer=false,payIn=false,reputation=true,securityDeposit=true,claimerBounty
  =true,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid initialize (senderClaimer=true,payIn=false,reputation=true,securityDeposit=true,claimerBounty=
  true,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid initialize (senderClaimer=false,payIn=true,reputation=true,securityDeposit=true,claimerBounty=
  true,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid initialize (senderClaimer=true,payIn=true,reputation=true,securityDeposit=true,claimerBounty=
  true,usesNativeToken=false,usesNativeTokenForDeposit=true)
    [PASS] Valid initialize (senderClaimer=false,payIn=false,reputation=false,securityDeposit=false,
  claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid initialize (senderClaimer=false,payIn=true,reputation=false,securityDeposit=false,
  claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid initialize (senderClaimer=false,payIn=false,reputation=true,securityDeposit=false,
  claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid initialize (senderClaimer=false,payIn=true,reputation=true,securityDeposit=false,claimerBounty
  =false,usesNativeToken=true,usesNativeTokenForDeposit=true)
    [PASS] Valid initialize (senderClaimer=false,payIn=false,reputation=false,securityDeposit=true,
  claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=true)
```

```
      [PASS] Valid initialize (senderClaimer=false,payIn=true,reputation=false,securityDeposit=true,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=true)
      [PASS] Valid initialize (senderClaimer=false,payIn=false,reputation=true,securityDeposit=true,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=true)
      [PASS] Valid initialize (senderClaimer=false,payIn=true,reputation=true,securityDeposit=true,claimerBounty=false,usesNativeToken=true,usesNativeTokenForDeposit=true)
      [PASS] Valid initialize (senderClaimer=false,payIn=false,reputation=false,securityDeposit=false,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=true)
      [PASS] Valid initialize (senderClaimer=false,payIn=true,reputation=false,securityDeposit=false,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=true)
      [PASS] Valid initialize (senderClaimer=false,payIn=false,reputation=true,securityDeposit=false,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=true)
      [PASS] Valid initialize (senderClaimer=false,payIn=true,reputation=true,securityDeposit=false,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=true)
      [PASS] Valid initialize (senderClaimer=false,payIn=false,reputation=false,securityDeposit=true,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=true)
      [PASS] Valid initialize (senderClaimer=false,payIn=true,reputation=false,securityDeposit=true,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=true)
      [PASS] Valid initialize (senderClaimer=false,payIn=false,reputation=true,securityDeposit=true,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=true)
      [PASS] Valid initialize (senderClaimer=false,payIn=true,reputation=true,securityDeposit=true,claimerBounty=true,usesNativeToken=true,usesNativeTokenForDeposit=true)
      [PASS] Valid initialize (claimer is an erc1271 account)
      [PASS] Invalid initialize not enough balance (payIn)
      [PASS] Invalid initialize not enough balance (not payIn)
      [PASS] Invalid initialize not enough native token sent to the contract
      [PASS] Invalid initialize (check correct handling of msg.value)
      [PASS] Invalid initialize not enough allowance (payIn erc-20)
      [PASS] Invalid initialize not enough deposit token balance (erc-20)
      [PASS] Invalid initialize not enough deposit token allowance (erc-20)
      [PASS] Invalid initialize wrong signer
      [PASS] Invalid initialize bad sign message
      [PASS] Invalid initialize bad sign message (extra data)
      [PASS] Invalid initialize 3rd party caller
      [PASS] Invalid initialize expired
      [PASS] Invalid initialize sign different timeout
      [PASS] Invalid initialize commit twice

  ExecutionContract
    [PASS] Valid create (erc-20)
    [PASS] Invalid create (erc-20 not enough balance)
    [PASS] Invalid create (erc-20 not enough allowance)
    [PASS] Valid create (native token)
    [PASS] Invalid create (native token invalid msg.value)
    [PASS] Invalid create twice the same execution
    [PASS] Valid refund expired
    [PASS] Invalid refund expired, not initiated
    [PASS] Invalid refund expired, not expired
    [PASS] Invalid refund expired, already processed
    [PASS] Valid refund by owner
    [PASS] Valid refund by owner (even though not expired yet)
    [PASS] Invalid refund by owner, caller not owner
    [PASS] Invalid refund by owner, not initiated
    [PASS] Invalid refund by owner, try to refund twice
    [PASS] Valid execute, empty calls
    [PASS] Valid execute, erc20 transfer calls
    [PASS] Valid execute, native transfer calls
    [PASS] Valid execute, dummy emit event
    [PASS] Valid execute, call reverted
    [PASS] Valid execute, out of gas
    [PASS] Valid execute, dummy emit event (drain tokens)
    [PASS] Valid execute, call reverted (drain tokens)
    [PASS] Invalid execute, wrong success action provided
    [PASS] Invalid execute, not scheduled
    [PASS] Invalid execute, try execute twice

  SpvVaultManager
    Open
      [PASS] Valid open vault (51ms)
      [PASS] Invalid open vault (already opened)
    Deposit
      [PASS] Valid deposit (token0Native=false,token1Native=false,token0Type=false,token1Type=false,thirdPartyDeposit=false,noToken0=false,noToken1=false)
      [PASS] Valid deposit (token0Native=true,token1Native=false,token0Type=false,token1Type=false,thirdPartyDeposit=false,noToken0=false,noToken1=false)
      [PASS] Valid deposit (token0Native=false,token1Native=true,token0Type=false,token1Type=false,thirdPartyDeposit=false,noToken0=false,noToken1=false)
      [PASS] Valid deposit (token0Native=true,token1Native=true,token0Type=false,token1Type=false,thirdPartyDeposit=false,noToken0=false,noToken1=false)
      [PASS] Valid deposit (token0Native=false,token1Native=false,token0Type=true,token1Type=false,thirdPartyDeposit=false,noToken0=false,noToken1=false)
      [PASS] Valid deposit (token0Native=false,token1Native=true,token0Type=true,token1Type=false,thirdPartyDeposit=false,noToken0=false,noToken1=false)
      [PASS] Valid deposit (token0Native=false,token1Native=false,token0Type=false,token1Type=true,thirdPartyDeposit=false,noToken0=false,noToken1=false)
      [PASS] Valid deposit (token0Native=true,token1Native=false,token0Type=false,token1Type=true,thirdPartyDeposit=false,noToken0=false,noToken1=false)
      [PASS] Valid deposit (token0Native=false,token1Native=false,token0Type=true,token1Type=true,thirdPartyDeposit=false,noToken0=false,noToken1=false)
      [PASS] Valid deposit (token0Native=false,token1Native=false,token0Type=false,token1Type=false,thirdPartyDeposit=true,noToken0=false,noToken1=false)
      [PASS] Valid deposit (token0Native=true,token1Native=false,token0Type=false,token1Type=false,thirdPartyDeposit=true,noToken0=false,noToken1=false)
      [PASS] Valid deposit (token0Native=false,token1Native=true,token0Type=false,token1Type=false,thirdPartyDeposit=true,noToken0=false,noToken1=false)
      [PASS] Valid deposit (token0Native=true,token1Native=true,token0Type=false,token1Type=false,thirdPartyDeposit=true,noToken0=false,noToken1=false)
      [PASS] Valid deposit (token0Native=false,token1Native=false,token0Type=true,token1Type=false,thirdPartyDeposit=true,noToken0=false,noToken1=false)
      [PASS] Valid deposit (token0Native=false,token1Native=true,token0Type=true,token1Type=false,thirdPartyDeposit=true,noToken0=false,noToken1=false)
      [PASS] Valid deposit (token0Native=false,token1Native=false,token0Type=false,token1Type=true,thirdPartyDeposit=true,noToken0=false,noToken1=false)
      [PASS] Valid deposit (token0Native=true,token1Native=false,token0Type=false,token1Type=true,thirdPartyDeposit=true,noToken0=false,noToken1=false)
      [PASS] Valid deposit (token0Native=false,token1Native=false,token0Type=true,token1Type=true,thirdPartyDeposit=true,noToken0=false,noToken1=false)
      [PASS] Valid deposit (token0Native=false,token1Native=false,token0Type=false,token1Type=false,thirdPartyDeposit=false,noToken0=true,noToken1=false)
      [PASS] Valid deposit (token0Native=true,token1Native=false,token0Type=false,token1Type=false,thirdPartyDeposit=false,noToken0=true,noToken1=false)
      [PASS] Valid deposit (token0Native=false,token1Native=true,token0Type=false,token1Type=false,thirdPartyDeposit=false,noToken0=true,noToken1=false)
      [PASS] Valid deposit (token0Native=true,token1Native=true,token0Type=false,token1Type=false,thirdPartyDeposit=false,noToken0=true,noToken1=false)
      [PASS] Valid deposit (token0Native=false,token1Native=false,token0Type=true,token1Type=false,thirdPartyDeposit=false,noToken0=true,noToken1=false)
      [PASS] Valid deposit (token0Native=false,token1Native=false,token0Type=true,token1Type=false,thirdPartyDeposit=false,noToken0=false,noToken1=true)
      [PASS] Valid deposit (token0Native=false,token1Native=true,token0Type=true,token1Type=false,thirdPartyDeposit=false,noToken0=false,noToken1=true)
      [PASS] Valid deposit (token0Native=false,token1Native=false,token0Type=false,token1Type=true,thirdPartyDeposit=false,noToken0=false,noToken1=true)
      [PASS] Valid deposit (token0Native=true,token1Native=false,token0Type=false,token1Type=true,thirdPartyDeposit=false,noToken0=false,noToken1=true)
      [PASS] Valid deposit (token0Native=false,token1Native=false,token0Type=true,token1Type=true,thirdPartyDeposit=false,noToken0=false,noToken1=true)
      [PASS] Valid deposit (token0Native=false,token1Native=false,token0Type=false,token1Type=false,thirdPartyDeposit=true,noToken0=false,noToken1=true)
      [PASS] Valid deposit (token0Native=true,token1Native=false,token0Type=false,token1Type=false,thirdPartyDeposit=true,noToken0=false,noToken1=true)
      [PASS] Valid deposit (token0Native=false,token1Native=true,token0Type=false,token1Type=false,thirdPartyDeposit=true,noToken0=false,noToken1=true)
      [PASS] Valid deposit (token0Native=true,token1Native=true,token0Type=false,token1Type=false,thirdPartyDeposit=true,noToken0=false,noToken1=true)
      [PASS] Valid deposit (token0Native=false,token1Native=false,token0Type=true,token1Type=false,thirdPartyDeposit=true,noToken0=false,noToken1=true)
      [PASS] Valid deposit (token0Native=false,token1Native=true,token0Type=true,token1Type=false,thirdPartyDeposit=true,noToken0=false,noToken1=true)
    [PASS] Valid deposit (token0Native=false,token1Native=false,token0Type=false,token1Type=true,thirdPartyDeposit=true,noToken0=false,noToken1=true)
    [PASS] Valid deposit (token0Native=true,token1Native=false,token0Type=false,token1Type=true,thirdPartyDeposit=true,noToken0=false,noToken1=true)
    [PASS] Valid deposit (token0Native=false,token1Native=false,token0Type=true,token1Type=true,thirdPartyDeposit=true,noToken0=false,noToken1=true)
    [PASS] Invalid deposit - not enough balance erc-20 (token0Native=false,token1Native=false,token0Type=false,token1Type=false,token0NotEnoughBalance=true,token1NotEnoughBalance=false)
    [PASS] Invalid deposit - not enough balance erc-20 (token0Native=false,token1Native=true,token0Type=false,token1Type=false,token0NotEnoughBalance=true,token1NotEnoughBalance=false)
    [PASS] Invalid deposit - not enough balance erc-20 (token0Native=false,token1Native=false,token0Type=true,token1Type=false,token0NotEnoughBalance=true,token1NotEnoughBalance=false)
    [PASS] Invalid deposit - not enough balance erc-20 (token0Native=false,token1Native=true,token0Type=true,token1Type=false,token0NotEnoughBalance=true,token1NotEnoughBalance=false)
    [PASS] Invalid deposit - not enough balance erc-20 (token0Native=false,token1Native=false,token0Type=true,token1Type=true,token0NotEnoughBalance=false,token1NotEnoughBalance=true)
    [PASS] Invalid deposit - not enough balance erc-20 (token0Native=false,token1Native=false,token0Type=false,token1Type=false,token0NotEnoughBalance=true,token1NotEnoughBalance=true)
    [PASS] Invalid deposit - not enough balance erc-20 (token0Native=false,token1Native=false,token0Type=true,token1Type=false,token0NotEnoughBalance=true,token1NotEnoughBalance=true)
    [PASS] Invalid deposit - not enough balance erc-20 (token0Native=false,token1Native=false,token0Type=false,token1Type=true,token0NotEnoughBalance=true,token1NotEnoughBalance=true)
    [PASS] Invalid deposit - not enough balance erc-20 (token0Native=false,token1Native=false,token0Type=true,token1Type=true,token0NotEnoughBalance=true,token1NotEnoughBalance=true)
    [PASS] Invalid deposit - not enough allowance erc-20 (token0Native=false,token1Native=false,token0Type=false,token1Type=false,token0NotEnoughAllowance=true,token1NotEnoughAllowance=false)
    [PASS] Invalid deposit - not enough allowance erc-20 (token0Native=false,token1Native=true,token0Type=false,token1Type=false,token0NotEnoughAllowance=true,token1NotEnoughAllowance=false)
    [PASS] Invalid deposit - not enough allowance erc-20 (token0Native=false,token1Native=false,token0Type=true,token1Type=false,token0NotEnoughAllowance=true,token1NotEnoughAllowance=false)
    [PASS] Invalid deposit - not enough allowance erc-20 (token0Native=false,token1Native=true,token0Type=true,token1Type=false,token0NotEnoughAllowance=true,token1NotEnoughAllowance=false)
    [PASS] Invalid deposit - not enough allowance erc-20 (token0Native=false,token1Native=false,token0Type=false,token1Type=false,token0NotEnoughAllowance=true,token1NotEnoughAllowance=true)
    [PASS] Invalid deposit - not enough allowance erc-20 (token0Native=false,token1Native=false,token0Type=true,token1Type=false,token0NotEnoughAllowance=true,token1NotEnoughAllowance=true)
    [PASS] Invalid deposit - not enough allowance erc-20 (token0Native=false,token1Native=false,token0Type=false,token1Type=true,token0NotEnoughAllowance=true,token1NotEnoughAllowance=true)
    [PASS] Invalid deposit - not enough allowance erc-20 (token0Native=false,token1Native=false,token0Type=true,token1Type=true,token0NotEnoughAllowance=true,token1NotEnoughAllowance=true)
    [PASS] Invalid deposit - not enough msg.value (token0Native=true,token1Native=false,token0Type=false,token1Type=false,token0NotEnoughMsgValue=true,token1NotEnoughMsgValue=false)
    [PASS] Invalid deposit - not enough msg.value (token0Native=true,token1Native=true,token0Type=false,token1Type=false,token0NotEnoughMsgValue=true,token1NotEnoughMsgValue=false)
    [PASS] Invalid deposit - not enough msg.value (token0Native=true,token1Native=false,token0Type=false,token1Type=true,token0NotEnoughMsgValue=true,token1NotEnoughMsgValue=false)
    [PASS] Invalid deposit - not enough msg.value (token0Native=false,token1Native=true,token0Type=false,token1Type=false,token0NotEnoughMsgValue=false,token1NotEnoughMsgValue=true)
    [PASS] Invalid deposit - not enough msg.value (token0Native=true,token1Native=true,token0Type=false,token1Type=false,token0NotEnoughMsgValue=false,token1NotEnoughMsgValue=true)
    [PASS] Invalid deposit - not enough msg.value (token0Native=false,token1Native=true,token0Type=true,token1Type=false,token0NotEnoughMsgValue=false,token1NotEnoughMsgValue=true)
    [PASS] Invalid deposit - not enough msg.value (token0Native=true,token1Native=true,token0Type=false,token1Type=false,token0NotEnoughMsgValue=true,token1NotEnoughMsgValue=true)
    [PASS] Invalid deposit - msg.value only enough for one deposit
    [PASS] Invalid deposit - vault not opened
    [PASS] Invalid deposit - vault got closed
  Front
    [PASS] Valid front (token0Native=false,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=false)
    [PASS] Valid front (token0Native=true,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=false)
    [PASS] Valid front (token0Native=false,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=false)
    [PASS] Valid front (token0Native=true,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=false)
    [PASS] Valid front (token0Native=false,token1Native=false,token0Type=false,token1Type=true,thirdPartyFront=true,noToken0=true,noToken1=false,usesExecution=false)
    [PASS] Valid front (token0Native=true,token1Native=false,token0Type=false,token1Type=true,thirdPartyFront=true,noToken0=true,noToken1=false,usesExecution=false)
  [PASS] Valid front (token0Native=false,token1Native=false,token0Type=true,token1Type=true,thirdPartyFront=true,noToken0=true,noToken1=false,usesExecution=false)
  [PASS] Valid front (token0Native=false,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=true,usesExecution=false)
  [PASS] Valid front (token0Native=true,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=true,usesExecution=false)
  [PASS] Valid front (token0Native=false,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=true,usesExecution=false)
  [PASS] Valid front (token0Native=true,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=true,usesExecution=false)
  [PASS] Valid front (token0Native=false,token1Native=false,token0Type=true,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=true,usesExecution=false)
  [PASS] Valid front (token0Native=false,token1Native=true,token0Type=true,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=true,usesExecution=false)
  [PASS] Valid front (token0Native=false,token1Native=false,token0Type=false,token1Type=true,thirdPartyFront=false,noToken0=false,noToken1=true,usesExecution=false)
  [PASS] Valid front (token0Native=true,token1Native=false,token0Type=false,token1Type=true,thirdPartyFront=false,noToken0=false,noToken1=true,usesExecution=false)
  [PASS] Valid front (token0Native=false,token1Native=false,token0Type=true,token1Type=true,thirdPartyFront=false,noToken0=false,noToken1=true,usesExecution=false)
  [PASS] Valid front (token0Native=false,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=false)
  [PASS] Valid front (token0Native=true,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=false)
  [PASS] Valid front (token0Native=false,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=false)
  [PASS] Valid front (token0Native=true,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=false)
  [PASS] Valid front (token0Native=false,token1Native=false,token0Type=true,token1Type=false,thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=false)
  [PASS] Valid front (token0Native=false,token1Native=true,token0Type=true,token1Type=false,thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=false)
  [PASS] Valid front (token0Native=false,token1Native=false,token0Type=false,token1Type=true,thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=false)
  [PASS] Valid front (token0Native=true,token1Native=false,token0Type=false,token1Type=true,thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=false)
  [PASS] Valid front (token0Native=false,token1Native=false,token0Type=true,token1Type=true,thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=false)
  [PASS] Valid front (token0Native=false,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=true)
  [PASS] Valid front (token0Native=true,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=true)
  [PASS] Valid front (token0Native=false,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=true)
  [PASS] Valid front (token0Native=true,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=true)
  [PASS] Valid front (token0Native=false,token1Native=false,token0Type=true,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=true)
  [PASS] Valid front (token0Native=false,token1Native=true,token0Type=true,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=true)
  [PASS] Valid front (token0Native=false,token1Native=false,token0Type=false,token1Type=true,thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=true)
  [PASS] Valid front (token0Native=true,token1Native=false,token0Type=false,token1Type=true,thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=true)
  [PASS] Valid front (token0Native=false,token1Native=false,token0Type=true,token1Type=true,thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=true)
  [PASS] Valid front (token0Native=false,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront=true,noToken0=false,noToken1=false,usesExecution=true)
  [PASS] Valid front (token0Native=true,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront=true,noToken0=false,noToken1=false,usesExecution=true)
  [PASS] Valid front (token0Native=false,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=true,noToken0=false,noToken1=false,usesExecution=true)
  [PASS] Valid front (token0Native=true,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=true,noToken0=false,noToken1=false,usesExecution=true)
  [PASS] Valid front (token0Native=false,token1Native=false,token0Type=true,token1Type=false,thirdPartyFront=true,noToken0=false,noToken1=false,usesExecution=true)
  [PASS] Valid front (token0Native=false,token1Native=true,token0Type=true,token1Type=false,thirdPartyFront=true,noToken0=false,noToken1=false,usesExecution=true)
  [PASS] Valid front (token0Native=false,token1Native=false,token0Type=false,token1Type=true,thirdPartyFront=true,noToken0=false,noToken1=false,usesExecution=true)
  [PASS] Valid front (token0Native=true,token1Native=false,token0Type=false,token1Type=true,thirdPartyFront=true,noToken0=false,noToken1=false,usesExecution=true)
  [PASS] Valid front (token0Native=false,token1Native=false,token0Type=true,token1Type=true,thirdPartyFront=true,noToken0=false,noToken1=false,usesExecution=true)
  [PASS] Valid front (token0Native=false,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront=false,noToken0=true,noToken1=false,usesExecution=true)
  [PASS] Valid front (token0Native=true,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront=false,noToken0=true,noToken1=false,usesExecution=true)
  [PASS] Valid front (token0Native=false,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=false,noToken0=true,noToken1=false,usesExecution=true)
  [PASS] Valid front (token0Native=true,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=false,noToken0=true,noToken1=false,usesExecution=true)
  [PASS] Valid front (token0Native=false,token1Native=false,token0Type=true,token1Type=false,thirdPartyFront=false,noToken0=true,noToken1=false,usesExecution=true)
  [PASS] Valid front (token0Native=true,token1Native=false,token0Type=false,token1Type=true,thirdPartyFront=false,noToken0=false,noToken1=true,usesExecution=true)
  [PASS] Valid front (token0Native=false,token1Native=false,token0Type=true,token1Type=true,thirdPartyFront=false,noToken0=false,noToken1=true,usesExecution=true)
  [PASS] Valid front (token0Native=false,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=true)
  [PASS] Valid front (token0Native=true,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=true)
  [PASS] Valid front (token0Native=false,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=true)
  [PASS] Valid front (token0Native=true,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=true)
  [PASS] Valid front (token0Native=false,token1Native=false,token0Type=true,token1Type=false,thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=true)
  [PASS] Valid front (token0Native=false,token1Native=true,token0Type=true,token1Type=false,thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=true)
  [PASS] Valid front (token0Native=false,token1Native=false,token0Type=false,token1Type=true,thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=true)
  [PASS] Valid front (token0Native=true,token1Native=false,token0Type=false,token1Type=true,thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=true)
  [PASS] Valid front (token0Native=false,token1Native=false,token0Type=true,token1Type=true,thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=true)
  [PASS] Invalid front - not enough balance erc-20 (token0Native=false,token1Native=false,token0Type=false,token1Type=false,token0NotEnoughBalance=true,token1NotEnoughBalance=false)
  [PASS] Invalid front - not enough balance erc-20 (token0Native=false,token1Native=true,token0Type=false,token1Type=false,token0NotEnoughBalance=true,token1NotEnoughBalance=false)
  [PASS] Invalid front - not enough balance erc-20 (token0Native=false,token1Native=false,token0Type=true,token1Type=false,token0NotEnoughBalance=true,token1NotEnoughBalance=false)
  [PASS] Invalid front - not enough balance erc-20 (token0Native=false,token1Native=true,token0Type=true,token1Type=false,token0NotEnoughBalance=true,token1NotEnoughBalance=false)
  [PASS] Invalid front - not enough balance erc-20 (token0Native=false,token1Native=false,token0Type=false,token1Type=true,token0NotEnoughBalance=true,token1NotEnoughBalance=false)
  [PASS] Invalid front - not enough balance erc-20 (token0Native=false,token1Native=false,token0Type=false,token1Type=true,token0NotEnoughBalance=false,token1NotEnoughBalance=true)
  [PASS] Invalid front - not enough balance erc-20 (token0Native=true,token1Native=false,token0Type=false,token1Type=true,token0NotEnoughBalance=false,token1NotEnoughBalance=true)
  [PASS] Invalid front - not enough balance erc-20 (token0Native=false,token1Native=false,token0Type=true,token1Type=true,token0NotEnoughBalance=false,token1NotEnoughBalance=true)
  [PASS] Invalid front - not enough balance erc-20 (token0Native=false,token1Native=false,token0Type=false,token1Type=false,token0NotEnoughBalance=true,token1NotEnoughBalance=true)
  [PASS] Invalid front - not enough balance erc-20 (token0Native=false,token1Native=false,token0Type=true,token1Type=false,token0NotEnoughBalance=true,token1NotEnoughBalance=true)
  [PASS] Invalid front - not enough balance erc-20 (token0Native=false,token1Native=false,token0Type=false,token1Type=true,token0NotEnoughBalance=true,token1NotEnoughBalance=true)
  [PASS] Invalid front - not enough balance erc-20 (token0Native=false,token1Native=false,token0Type=true,token1Type=true,token0NotEnoughBalance=true,token1NotEnoughBalance=true)
  [PASS] Invalid front - not enough allowance erc-20 (token0Native=false,token1Native=false,token0Type=false,token1Type=false,token0NotEnoughAllowance=true,token1NotEnoughAllowance=false)
  [PASS] Invalid front - not enough allowance erc-20 (token0Native=false,token1Native=true,token0Type=false,token1Type=false,token0NotEnoughAllowance=true,token1NotEnoughAllowance=false)
  [PASS] Invalid front - not enough allowance erc-20 (token0Native=false,token1Native=false,token0Type=true,token1Type=false,token0NotEnoughAllowance=true,token1NotEnoughAllowance=false)
  [PASS] Invalid front - not enough allowance erc-20 (token0Native=false,token1Native=true,token0Type=true,token1Type=false,token0NotEnoughAllowance=true,token1NotEnoughAllowance=false)
  [PASS] Invalid front - not enough allowance erc-20 (token0Native=false,token1Native=false,token0Type=false,token1Type=true,token0NotEnoughAllowance=true,token1NotEnoughAllowance=false)
```

```
    [PASS] Invalid front - not enough allowance erc-20 (token0Native=false,token1Native=false,token0Type=false,token1Type=false,token0NotEnoughAllowance=true,token1NotEnoughAllowance=true)
    [PASS] Invalid front - not enough allowance erc-20 (token0Native=false,token1Native=false,token0Type=true,token1Type=false,token0NotEnoughAllowance=true,token1NotEnoughAllowance=true)
    [PASS] Invalid front - not enough allowance erc-20 (token0Native=false,token1Native=false,token0Type=false,token1Type=true,token0NotEnoughAllowance=true,token1NotEnoughAllowance=true)
    [PASS] Invalid front - not enough allowance erc-20 (token0Native=false,token1Native=false,token0Type=true,token1Type=true,token0NotEnoughAllowance=true,token1NotEnoughAllowance=true)
    [PASS] Invalid front - not enough msg.value (token0Native=true,token1Native=false,token0Type=false,token1Type=false,token0NotEnoughMsgValue=true,token1NotEnoughMsgValue=false)
    [PASS] Invalid front - not enough msg.value (token0Native=true,token1Native=true,token0Type=false,token1Type=false,token0NotEnoughMsgValue=true,token1NotEnoughMsgValue=false)
    [PASS] Invalid front - not enough msg.value (token0Native=true,token1Native=false,token0Type=false,token1Type=true,token0NotEnoughMsgValue=true,token1NotEnoughMsgValue=false)
    [PASS] Invalid front - not enough msg.value (token0Native=false,token1Native=true,token0Type=false,token1Type=false,token0NotEnoughMsgValue=false,token1NotEnoughMsgValue=true)
    [PASS] Invalid front - not enough msg.value (token0Native=true,token1Native=true,token0Type=false,token1Type=false,token0NotEnoughMsgValue=false,token1NotEnoughMsgValue=true)
    [PASS] Invalid front - not enough msg.value (token0Native=false,token1Native=true,token0Type=true,token1Type=false,token0NotEnoughMsgValue=false,token1NotEnoughMsgValue=true)
    [PASS] Invalid front - not enough msg.value (token0Native=true,token1Native=true,token0Type=false,token1Type=false,token0NotEnoughMsgValue=true,token1NotEnoughMsgValue=true)
    [PASS] Invalid front - msg.value only enough for one deposit
    [PASS] Invalid front - vault not opened
    [PASS] Invalid front - vault got closed
    [PASS] Invalid front - already fronted
    [PASS] Invalid front - already claimed
    [PASS] Invalid front - amount0 overflow 64-bits
  Claim
    [PASS] Valid claim (token0Native=false,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=false)
    [PASS] Valid claim (token0Native=true,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=false)
    [PASS] Valid claim (token0Native=false,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=false)
    [PASS] Valid claim (token0Native=true,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=false)
    [PASS] Valid claim (token0Native=false,token1Native=false,token0Type=true,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=false) (38ms)
    [PASS] Valid claim (token0Native=false,token1Native=true,token0Type=true,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=false)
    [PASS] Valid claim (token0Native=false,token1Native=false,token0Type=false,token1Type=true,thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=false)
    [PASS] Valid claim (token0Native=true,token1Native=false,token0Type=false,token1Type=true,thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=false)
    [PASS] Valid claim (token0Native=false,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=true,usesExecution=false)
    [PASS] Valid claim (token0Native=true,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=true,usesExecution=false)
    [PASS] Valid claim (token0Native=false,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=true,usesExecution=false)
    [PASS] Valid claim (token0Native=true,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=true,usesExecution=false)
    [PASS] Valid claim (token0Native=false,token1Native=false,token0Type=true,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=true,usesExecution=false)
    [PASS] Valid claim (token0Native=false,token1Native=true,token0Type=true,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=true,usesExecution=false)
    [PASS] Valid claim (token0Native=false,token1Native=false,token0Type=false,token1Type=true,thirdPartyFront=false,noToken0=false,noToken1=true,usesExecution=false)
    [PASS] Valid claim (token0Native=true,token1Native=false,token0Type=false,token1Type=true,thirdPartyFront=false,noToken0=false,noToken1=true,usesExecution=false)
    [PASS] Valid claim (token0Native=false,token1Native=false,token0Type=true,token1Type=true,thirdPartyFront=false,noToken0=false,noToken1=true,usesExecution=false)
    [PASS] Valid claim (token0Native=false,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=false)
    [PASS] Valid claim (token0Native=true,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=false)
    [PASS] Valid claim (token0Native=false,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=false)
    [PASS] Valid claim (token0Native=true,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=
  true,noToken0=false,noToken1=true,usesExecution=false)
    [PASS] Valid claim (token0Native=false,token1Native=false,token0Type=true,token1Type=false,thirdPartyFront=
  true,noToken0=false,noToken1=true,usesExecution=false)
    [PASS] Valid claim (token0Native=false,token1Native=true,token0Type=true,token1Type=false,thirdPartyFront=
  true,noToken0=false,noToken1=true,usesExecution=false)
    [PASS] Valid claim (token0Native=false,token1Native=false,token0Type=false,token1Type=true,thirdPartyFront=
  true,noToken0=false,noToken1=true,usesExecution=false)
    [PASS] Valid claim (token0Native=true,token1Native=false,token0Type=false,token1Type=true,thirdPartyFront=
  true,noToken0=false,noToken1=true,usesExecution=false)
    [PASS] Valid claim (token0Native=false,token1Native=false,token0Type=true,token1Type=true,thirdPartyFront=
  true,noToken0=false,noToken1=true,usesExecution=false)
    [PASS] Valid claim (token0Native=false,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront
  =false,noToken0=false,noToken1=false,usesExecution=true)
    [PASS] Valid claim (token0Native=true,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront=
  false,noToken0=false,noToken1=false,usesExecution=true)
    [PASS] Valid claim (token0Native=false,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=
  false,noToken0=false,noToken1=false,usesExecution=true)
    [PASS] Valid claim (token0Native=true,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=
  false,noToken0=false,noToken1=false,usesExecution=true)
    [PASS] Valid claim (token0Native=false,token1Native=false,token0Type=true,token1Type=false,thirdPartyFront=
  false,noToken0=false,noToken1=false,usesExecution=true)
    [PASS] Valid claim (token0Native=false,token1Native=true,token0Type=true,token1Type=false,thirdPartyFront=
  false,noToken0=false,noToken1=false,usesExecution=true)
    [PASS] Valid claim (token0Native=false,token1Native=false,token0Type=false,token1Type=true,thirdPartyFront=
  false,noToken0=false,noToken1=false,usesExecution=true)
    [PASS] Valid claim (token0Native=true,token1Native=false,token0Type=false,token1Type=true,thirdPartyFront=
  false,noToken0=false,noToken1=false,usesExecution=true)
    [PASS] Valid claim (token0Native=false,token1Native=false,token0Type=true,token1Type=true,thirdPartyFront=
  false,noToken0=false,noToken1=false,usesExecution=true)
    [PASS] Valid claim (token0Native=false,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront
  =true,noToken0=false,noToken1=false,usesExecution=true)
    [PASS] Valid claim (token0Native=true,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront=
  true,noToken0=false,noToken1=false,usesExecution=true)
    [PASS] Valid claim (token0Native=false,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=
  true,noToken0=false,noToken1=false,usesExecution=true)
    [PASS] Valid claim (token0Native=true,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=
  true,noToken0=false,noToken1=false,usesExecution=true)
    [PASS] Valid claim (token0Native=false,token1Native=false,token0Type=true,token1Type=false,thirdPartyFront=
  true,noToken0=false,noToken1=false,usesExecution=true)
    [PASS] Valid claim (token0Native=false,token1Native=true,token0Type=true,token1Type=false,thirdPartyFront=
  true,noToken0=false,noToken1=false,usesExecution=true)
    [PASS] Valid claim (token0Native=false,token1Native=false,token0Type=false,token1Type=true,thirdPartyFront=
  true,noToken0=false,noToken1=false,usesExecution=true)
    [PASS] Valid claim (token0Native=true,token1Native=false,token0Type=false,token1Type=true,thirdPartyFront=
  true,noToken0=false,noToken1=false,usesExecution=true)
    [PASS] Valid claim (token0Native=false,token1Native=false,token0Type=true,token1Type=true,thirdPartyFront=
  true,noToken0=false,noToken1=false,usesExecution=true)
    [PASS] Valid claim (token0Native=false,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront
  =false,noToken0=true,noToken1=false,usesExecution=true)
    [PASS] Valid claim (token0Native=true,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront=
  false,noToken0=true,noToken1=false,usesExecution=true)
    [PASS] Valid claim (token0Native=false,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=
  false,noToken0=true,noToken1=false,usesExecution=true)
    [PASS] Valid claim (token0Native=true,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=
  false,noToken0=true,noToken1=false,usesExecution=true)
    [PASS] Valid claim (token0Native=false,token1Native=false,token0Type=true,token1Type=false,thirdPartyFront=
  false,noToken0=true,noToken1=false,usesExecution=true)
    [PASS] Valid claim (token0Native=false,token1Native=true,token0Type=true,token1Type=false,thirdPartyFront=
  false,noToken0=true,noToken1=false,usesExecution=true)
    [PASS] Valid claim (token0Native=false,token1Native=false,token0Type=false,token1Type=true,thirdPartyFront=
  false,noToken0=true,noToken1=false,usesExecution=true) (40ms)
    [PASS] Valid claim (token0Native=true,token1Native=false,token0Type=false,token1Type=true,thirdPartyFront=
  false,noToken0=true,noToken1=false,usesExecution=true)
    [PASS] Valid claim (token0Native=false,token1Native=false,token0Type=true,token1Type=true,thirdPartyFront=
  false,noToken0=true,noToken1=false,usesExecution=true) (38ms)
    [PASS] Valid claim (token0Native=false,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront
  =true,noToken0=true,noToken1=false,usesExecution=true)
    [PASS] Valid claim (token0Native=false,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront
  =true,noToken0=false,noToken1=true,usesExecution=true)
```

```
  [PASS] Valid claim (token0Native=true,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=true)
  [PASS] Valid claim (token0Native=false,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=true)
  [PASS] Valid claim (token0Native=true,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=true)
  [PASS] Valid claim (token0Native=false,token1Native=false,token0Type=true,token1Type=false,thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=true)
  [PASS] Valid claim (token0Native=false,token1Native=true,token0Type=true,token1Type=false,thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=true)
  [PASS] Valid claim (token0Native=false,token1Native=false,token0Type=false,token1Type=true,thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=true)
  [PASS] Valid claim (token0Native=true,token1Native=false,token0Type=false,token1Type=true,thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=true)
  [PASS] Valid claim (token0Native=false,token1Native=false,token0Type=true,token1Type=true,thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=true) (38ms)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=false)
  [PASS] Valid claim - fronted (token0Native=true,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=false)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=false)
  [PASS] Valid claim - fronted (token0Native=true,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=false)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=false,token0Type=true,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=false) (40ms)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=true,token0Type=true,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=false)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=false,token0Type=false,token1Type=true,thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=false)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront=false,noToken0=true,noToken1=false,usesExecution=false)
  [PASS] Valid claim - fronted (token0Native=true,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront=false,noToken0=true,noToken1=false,usesExecution=false)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=false,noToken0=true,noToken1=false,usesExecution=false)
  [PASS] Valid claim - fronted (token0Native=true,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=false,noToken0=true,noToken1=false,usesExecution=false)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=false,token0Type=true,token1Type=false,thirdPartyFront=false,noToken0=true,noToken1=false,usesExecution=false)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=true,token0Type=true,token1Type=false,thirdPartyFront=false,noToken0=true,noToken1=false,usesExecution=false)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=false,token0Type=false,token1Type=true,thirdPartyFront=false,noToken0=true,noToken1=false,usesExecution=false) (38ms)
  [PASS] Valid claim - fronted (token0Native=true,token1Native=false,token0Type=false,token1Type=true,thirdPartyFront=false,noToken0=true,noToken1=false,usesExecution=false)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=false,token0Type=true,token1Type=true,thirdPartyFront=false,noToken0=true,noToken1=false,usesExecution=false)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront=true,noToken0=true,noToken1=false,usesExecution=false)
  [PASS] Valid claim - fronted (token0Native=true,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront=true,noToken0=true,noToken1=false,usesExecution=false)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=true,noToken0=true,noToken1=false,usesExecution=false) (43ms)
  [PASS] Valid claim - fronted (token0Native=true,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=true,noToken0=true,noToken1=false,usesExecution=false)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=false,token0Type=true,token1Type=false,thirdPartyFront=true,noToken0=true,noToken1=false,usesExecution=false) (41ms)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=true,token0Type=true,token1Type=false,thirdPartyFront=true,noToken0=true,noToken1=false,usesExecution=false)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=false,token0Type=false,token1Type=true,thirdPartyFront=true,noToken0=true,noToken1=false,usesExecution=false) (38ms)
  [PASS] Valid claim - fronted (token0Native=true,token1Native=false,token0Type=false,token1Type=true,thirdPartyFront=true,noToken0=true,noToken1=false,usesExecution=false)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=false,token0Type=true,token1Type=true,thirdPartyFront=true,noToken0=true,noToken1=false,usesExecution=false)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=false,token0Type=false,token1Type=false,thirdPartyFront=false,noToken0=false,noToken1=true,usesExecution=false) (38ms)
  [PASS] Valid claim - fronted (token0Native=true,token1Native=true,token0Type=false,token1Type=false,thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=false)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=false,token0Type=true,token1Type=false,
thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=false)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=true,token0Type=true,token1Type=false,
thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=false)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=false,token0Type=false,token1Type=true,
thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=false)
  [PASS] Valid claim - fronted (token0Native=true,token1Native=false,token0Type=false,token1Type=true,
thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=false) (41ms)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=false,token0Type=true,token1Type=true,
thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=false)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=false,token0Type=false,token1Type=false,
thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=true)
  [PASS] Valid claim - fronted (token0Native=true,token1Native=false,token0Type=false,token1Type=false,
thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=true)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=true,token0Type=false,token1Type=false,
thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=true)
  [PASS] Valid claim - fronted (token0Native=true,token1Native=true,token0Type=false,token1Type=false,
thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=true)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=false,token0Type=true,token1Type=false,
thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=true)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=true,token0Type=true,token1Type=false,
thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=true)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=false,token0Type=false,token1Type=true,
thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=true) (38ms)
  [PASS] Valid claim - fronted (token0Native=true,token1Native=false,token0Type=false,token1Type=true,
thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=true)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=false,token0Type=true,token1Type=true,
thirdPartyFront=false,noToken0=false,noToken1=false,usesExecution=true)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=false,token0Type=false,token1Type=false,
thirdPartyFront=true,noToken0=false,noToken1=false,usesExecution=true)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=true,token0Type=false,token1Type=false,
thirdPartyFront=true,noToken0=true,noToken1=false,usesExecution=true)
  [PASS] Valid claim - fronted (token0Native=true,token1Native=true,token0Type=false,token1Type=false,
thirdPartyFront=true,noToken0=true,noToken1=false,usesExecution=true)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=false,token0Type=true,token1Type=false,
thirdPartyFront=true,noToken0=true,noToken1=false,usesExecution=true)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=true,token0Type=true,token1Type=false,
thirdPartyFront=true,noToken0=true,noToken1=false,usesExecution=true)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=false,token0Type=false,token1Type=true,
thirdPartyFront=true,noToken0=true,noToken1=false,usesExecution=true)
  [PASS] Valid claim - fronted (token0Native=true,token1Native=false,token0Type=false,token1Type=true,
thirdPartyFront=true,noToken0=true,noToken1=false,usesExecution=true)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=false,token0Type=true,token1Type=true,
thirdPartyFront=true,noToken0=true,noToken1=false,usesExecution=true)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=false,token0Type=false,token1Type=false,
thirdPartyFront=false,noToken0=false,noToken1=true,usesExecution=true)
  [PASS] Valid claim - fronted (token0Native=true,token1Native=false,token0Type=false,token1Type=false,
thirdPartyFront=false,noToken0=false,noToken1=true,usesExecution=true)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=true,token0Type=false,token1Type=false,
thirdPartyFront=false,noToken0=false,noToken1=true,usesExecution=true)
  [PASS] Valid claim - fronted (token0Native=true,token1Native=true,token0Type=false,token1Type=false,
thirdPartyFront=false,noToken0=false,noToken1=true,usesExecution=true)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=false,token0Type=true,token1Type=false,
thirdPartyFront=false,noToken0=false,noToken1=true,usesExecution=true) (48ms)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=true,token0Type=true,token1Type=false,
thirdPartyFront=false,noToken0=false,noToken1=true,usesExecution=true)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=false,token0Type=false,token1Type=true,
thirdPartyFront=false,noToken0=false,noToken1=true,usesExecution=true)
  [PASS] Valid claim - fronted (token0Native=true,token1Native=false,token0Type=false,token1Type=true,
thirdPartyFront=false,noToken0=false,noToken1=true,usesExecution=true)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=false,token0Type=true,token1Type=true,
thirdPartyFront=false,noToken0=false,noToken1=true,usesExecution=true)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=false,token0Type=false,token1Type=false,
thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=true)
  [PASS] Valid claim - fronted (token0Native=true,token1Native=false,token0Type=false,token1Type=false,
thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=true)
  [PASS] Valid claim - fronted (token0Native=false,token1Native=true,token0Type=false,token1Type=false,
thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=true)
  [PASS] Valid claim - fronted (token0Native=true,token1Native=true,token0Type=false,token1Type=false,
thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=true)
```

```
      [PASS] Valid claim - fronted (token0Native=false,token1Native=false,token0Type=true,token1Type=false,
    thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=true) (43ms)
      [PASS] Valid claim - fronted (token0Native=false,token1Native=true,token0Type=true,token1Type=false,
    thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=true)
      [PASS] Valid claim - fronted (token0Native=false,token1Native=false,token0Type=false,token1Type=true,
    thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=true)
      [PASS] Valid claim - fronted (token0Native=true,token1Native=false,token0Type=false,token1Type=true,
    thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=true)
      [PASS] Valid claim - fronted (token0Native=false,token1Native=false,token0Type=true,token1Type=true,
    thirdPartyFront=true,noToken0=false,noToken1=true,usesExecution=true)
      [PASS] Invalid claim - vault not opened
      [PASS] Invalid claim - invalid tx confirmations
      [PASS] Invalid claim - invalid merkle proof
      [PASS] Invalid claim - invalid merkle proof (position)
      [PASS] Invalid claim - btc tx empty inputs
      [PASS] Invalid claim - btc tx empty inputs
      [PASS] Invalid claim (vault close) - output 1 not found
      [PASS] Invalid claim (vault close) - input 1 not found
      [PASS] Invalid claim (vault close) - output 1 empty script
      [PASS] Invalid claim (vault close) - output 1 not OP_RETURN
      [PASS] Invalid claim (vault close) - output 1 invalid len
      [PASS] Invalid claim (vault close) - caller fee 0 overflow
      [PASS] Invalid claim (vault close) - fronting fee 0 overflow
      [PASS] Invalid claim (vault close) - execution fee 0 overflow
      [PASS] Invalid claim (vault close) - caller fee 1 overflow
      [PASS] Invalid claim (vault close) - fronting fee 1 overflow
      [PASS] Invalid claim (vault close) - amount 0 sum overflow
      [PASS] Invalid claim (vault close) - amount 1 sum overflow
      [PASS] Invalid claim (vault close) - withdraw too much token0
      [PASS] Invalid claim (vault close) - withdraw too much token1

  BitcoinNoncedOutputClaimHandler
    [PASS] Valid random witness (87ms)
    [PASS] Valid real witness (4322ms)
    [PASS] Invalid empty witness
    [PASS] Invalid incorrect commitment witness
    [PASS] Invalid block confirmations
    [PASS] Invalid merkle proof, root doesn't match
    [PASS] Invalid blockheader, provided header is not known to the btc relay
    [PASS] Invalid vout of bounds
    [PASS] Invalid txoHash doesn't match (due to wrong outputAmount)
    [PASS] Invalid txoHash doesn't match (due to wrong outputScript)
    [PASS] Invalid txoHash doesn't match (due to wrong nonce)

  BitcoinOutputClaimHandler
    [PASS] Valid random witness (106ms)
    [PASS] Valid real witness (6354ms)
    [PASS] Invalid empty witness
    [PASS] Invalid incorrect commitment witness
    [PASS] Invalid block confirmations
    [PASS] Invalid merkle proof, root doesn't match
    [PASS] Invalid blockheader, provided header is not known to the btc relay
    [PASS] Invalid vout of bounds
    [PASS] Invalid txoHash doesn't match (due to wrong outputAmount)
    [PASS] Invalid txoHash doesn't match (due to wrong outputScript)

  CompactBlockHeader
    [PASS] Valid verify out of bounds
    [PASS] Valid verify out of bounds with offset
    [PASS] Invalid verify out of bounds
    [PASS] Valid read values
    [PASS] Existing blockheaders (3335ms)

  Difficulty
    [PASS] Get chainwork random data
    [PASS] Compute new target real adjustments (9160ms)
    [PASS] Get chainwork real data (6726ms)

  Nbits
    [PASS] Nbits to target
    [PASS] Target to nbits
    [PASS] Bitcoin core test vector
    [PASS] Bitcoin core negative nbits test vector
    [PASS] Random targets

  StoredBlockHeader
    [PASS] Valid from calldata
    [PASS] Valid from calldata with offset
    [PASS] Invalid from calldata with offset
    [PASS] Valid read values
    [PASS] Existing blockheaders (6548ms)
    [PASS] Valid parse random (62ms)
    [PASS] Valid update random (567ms)
    [PASS] Valid update random block on PoW readjustment (1211ms)
    [PASS] Valid update random block on PoW readjustment, too fast (1496ms)
    [PASS] Valid update random block on PoW readjustment, too slow (796ms)
    [PASS] Valid update random block, with timestamp larger than median of last 11 blocks (591ms)
    [PASS] Invalid update random block, due to low PoW (3131ms)
    [PASS] Invalid update random block, due to wrong nBits (231ms)
    [PASS] Invalid update random block, due to wrong nBits during difficulty retarget (215ms)
    [PASS] Invalid update random block, due to wrong previous blockhash (589ms)
    [PASS] Invalid update random block, due to timestamp not being larger than median of last 11 blocks (547ms)
    [PASS] Invalid update random block, due to timestamp being too far in the future (420ms)
    [PASS] Valid update real (9008ms)
    [PASS] Valid update real on PoW readjustment (965ms)
    [PASS] Valid update real on PoW readjustment, too fast (964ms)

  BitcoinTxIdClaimHandler
    [PASS] Valid random witness (96ms)
    [PASS] Valid real witness (4334ms)
    [PASS] Invalid empty witness
    [PASS] Invalid incorrect commitment witness
    [PASS] Invalid block confirmations
    [PASS] Invalid merkle proof, root doesn't match
    [PASS] Invalid blockheader, provided header is not known to the btc relay

  BitcoinMerkleTree
    [PASS] Randomly generated tests
    [PASS] Real data tests (4171ms)

  BitcoinTx
    [PASS] Valid random bitcoin txs (515ms)
getRealRandomTransactionTest(): 455ef3447d4be1d1c5ea78082828a2a5b5abc52d13d21233d44baa919bc79bcb
getRealRandomTransactionTest(): acbd6110dc9e3ab7692631d46e045b5ca5d698ea2d1aee27d8d33e045ff2414b
getRealRandomTransactionTest(): 76075dc0b066ece54fba8d5634b4da7cdea7b94be66d7267440207de34522fcd
getRealRandomTransactionTest(): 45709232aa6257ef37be7d8d26a271307f481d72002e9e63f84c5b1f96d4b344
getRealRandomTransactionTest(): f74bec3437c4150341aae406346b77fc2c795de259ba5377da64a9e2c39e4a3a
getRealRandomTransactionTest(): dd86a8383f64bb8d57a8b84326ff3e01b3bde6d6dd9d3b007d745d05a5b776d9
getRealRandomTransactionTest(): 9ee76d0ec593361a6a1388e4b6e1c9f8c119482bce9cb71f364c54a3b56d0bba
getRealRandomTransactionTest(): 83e68be11340f6055477edb4e8a0fbf436e9fd90122bf22253ae6594311773bc
getRealRandomTransactionTest(): aaf0b8b717a8fba7c6d2a0d4aa42494c2ac4941583c5c14e9ef62bf7da59bf54
getRealRandomTransactionTest(): 9ab4423c919b861a510937df06a76ead71cef4c84e71af543fd6fa4fb19e45c6
    [PASS] Valid real bitcoin txs (5675ms)
    [PASS] Invalid witness not stripped
    [PASS] Invalid tx more than expected data
    [PASS] Invalid tx length exactly 64 bytes

  Endianness
    [PASS] Reverse uint32
    [PASS] Reverse uint64
    [PASS] Reverse bytes32

  EIP712Sighash
    [PASS] Random valid init (359ms)
    [PASS] Random valid refund (168ms)

  Escrow
    [PASS] Parse flags (49ms)
    [PASS] Total deposit calculation
    [PASS] Random hash
```

```

EscrowStorage
  [PASS] Commit
  [PASS] Invalid commit twice
  [PASS] Commit 2 different
  [PASS] Commit & finalize success
  [PASS] Commit & finalize not success
  [PASS] Invalid commit, finalize, try to re-commit
  [PASS] Invalid finalize, not committed
  [PASS] Commit 2 different, finalize 1

LpVault
  [PASS] Deposit
  [PASS] 2 Deposits
  [PASS] Invalid deposit not enough funds
  [PASS] Invalid deposit not enough allowance
  [PASS] Deposit ETH
  [PASS] Deposit ETH, more than required
  [PASS] Invalid deposit ETH, too low tx.value
  [PASS] Invalid deposit ETH, not enough balance
  [PASS] Withdraw
  [PASS] Withdraw ETH
  [PASS] Withdraw all
  [PASS] Invalid withdraw more than deposited
  [PASS] Invalid withdraw more than deposited (contract has enough token)
  [PASS] Transfer out
  [PASS] Transfer in
  [PASS] Transfer in all
  [PASS] Invalid transfer in, more than owned

ReputationState
  [PASS] Test updates
  [PASS] Test overflowing

ReputationTracker
  [PASS] Update reputation
  [PASS] Update reputation multiple types
  [PASS] Update reputation multiple tokens
  [PASS] Update reputation multiple claim handlers
  [PASS] Update reputation multiple tokens & claim handlers
  [PASS] Update reputation type out of bounds

Execution
  [PASS] Random hash

ExecutionProxy
  [PASS] Valid execute
  [PASS] Valid execute (payable)
  [PASS] Valid execute (revert)
  [PASS] Valid execute (contract doesn't exist)
  [PASS] Valid execute (out of gas) (16505ms)
  [PASS] Valid drainAll (erc-20) (257ms)
  [PASS] Valid drainAll (erc-20)
  [PASS] Valid drainAll (native)
  [PASS] Valid drainAll (native & 2x erc-20) (5949ms)
  [PASS] Valid drainAll (native & 2x erc-20, with contract having 0 balance of 1 of the token) (41ms)

Executor
  [PASS] Valid execute (success) (268ms)
  [PASS] Valid execute (success, multiple calls)
  [PASS] Valid execute (run out of gas)
  [PASS] Valid execute (rejection in contract call)
  [PASS] Valid execute (erc-20)
  [PASS] Valid execute (native token)
  [PASS] Valid execute payable (native token)
  [PASS] Valid execute payable (native token & other erc20)

HashlockClaimHandler
  [PASS] Valid witness
  [PASS] Invalid witness with more than 32 bytes
  [PASS] Invalid witness
  [PASS] Random valid witnesses

BitcoinVaultTransactionData
  [PASS] Parse valid (amount1 and execution) (74ms)
  [PASS] Parse valid (execution)
  [PASS] Parse valid (amount1)
  [PASS] Parse valid
  [PASS] Invalid single output only
  [PASS] Invalid 2nd output empty script
  [PASS] Invalid 2nd output not OP_RETURN
  [PASS] Invalid no input
  [PASS] Invalid just 1 input
  [PASS] Invalid 2nd output invalid length
  [PASS] Invalid caller fee 0 overflow
  [PASS] Invalid fronting fee 0 overflow
  [PASS] Invalid execution fee 0 overflow
  [PASS] Invalid caller fee 1 overflow
  [PASS] Invalid fronting fee 1 overflow
  [PASS] Valid get full amounts
  [PASS] Invalid get full amounts (overflow amount0)
  [PASS] Invalid get full amounts (overflow amount1)
  [PASS] Valid get hash

SpvVaultParameters
  [PASS] Valid from raw token0
  [PASS] Valid from raw token1
  [PASS] Valid from raw
  [PASS] Valid hash

SpvVaultState
  [PASS] Valid open (42ms)
  [PASS] Valid close
  [PASS] Valid is opened (true)
  [PASS] Valid is opened (false)
  [PASS] Valid check opened and params
  [PASS] Invalid check opened and params (not opened)
  [PASS] Invalid check opened and params (invalid params)
  [PASS] Valid withdraw
  [PASS] Invalid withdraw (amount 0 withdraw too much)
  [PASS] Invalid withdraw (amount 1 withdraw too much)
  [PASS] Valid deposit
  [PASS] Invalid deposit (amount 0 overflow)
  [PASS] Invalid deposit (amount 1 overflow)
  [PASS] Invalid deposit (deposit count overflow)

Utils
  [PASS] Valid pack address and vault id
  [PASS] Valid calculate fee
  [PASS] Invalid calculate fee (overflow)
  [PASS] Calculate fee random (142ms)

TimelockRefundHandler
  [PASS] Valid refund
  [PASS] Invalid refund, non-empty witness
  [PASS] Invalid refund, not expired yet

TransferUtils
  [PASS] Valid balance of (erc-20) (65ms)
  [PASS] Valid balance of (native token)
  [PASS] Valid transfer in (erc-20)
  [PASS] Valid transfer in (erc-20 not from caller)
  [PASS] Invalid transfer in, not enough balance (erc-20)
  [PASS] Invalid transfer in, not enough allowance (erc-20)
  [PASS] Valid transfer in (native token)
  [PASS] Invalid transfer in (native token not from caller)
  [PASS] Invalid transfer in not enough msg.value provided (native token)
  [PASS] Valid transfer out (erc-20)
  [PASS] Invalid transfer out not enough balance (erc-20)
  [PASS] Valid transfer out (native token)
```

```
    [PASS] Invalid transfer out not enough balance (native token)
    [PASS] Invalid transfer out, target reverted or ran out of gas (native token)
    [PASS] Valid transfer out (no revert) (erc-20)
    [PASS] Invalid transfer out not enough balance (no revert) (erc-20)
    [PASS] Valid transfer out (no revert) (native token)
    [PASS] Invalid transfer out not enough balance (no revert) (native token)
    [PASS] Invalid transfer out, target reverted or ran out of gas (no revert) (native token)

  MathUtils
    [PASS] Valid castToUint64
    [PASS] Valid castToUint64 overflow
    [PASS] Valid uncheckedAddUint64
    [PASS] Invalid uncheckedAddUint64, overflow (unchecked so simply just returns the lower bits of the value)
    [PASS] Valid checkedSubUint64
    [PASS] Valid checkedSubUint64, result 0
    [PASS] Valid checkedSubUint64, underflow
    [PASS] Valid checkedSubUint64, big underflow
    [PASS] Valid saturatingAddOneUint32
    [PASS] Valid saturatingAddOneUint32, saturated
    [PASS] Valid saturatingAddUint224
    [PASS] Valid saturatingAddUint224, exact saturated
    [PASS] Valid saturatingAddUint224, saturated
    [PASS] Valid saturatingAddUint224, saturated, big summand
    [PASS] Valid maxUint256, first is bigger
    [PASS] Valid maxUint256, second is bigger
    [PASS] Valid maxUint256, equal
    [PASS] Valid maxUint256, zero


  1246 passing (3m)
```