Assure DeFi®

THE VERIFICATION GOLD STANDARD

Security Assessment

TIXER

Date: 27/06/2025

Audit Status: PASS

Audit Edition: Advanced+

tixer

ASSURE DEFI®

THE VERIFICATION GOLD STANDARD

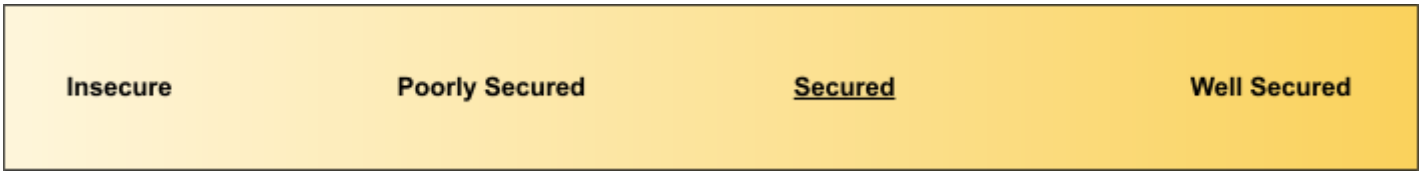
Risk Analysis

Vulnerability summary

Classification	Description
High	High-level vulnerabilities can result in the loss of assets or manipulation of data.
Medium	Medium-level vulnerabilities can be challenging to exploit, but they still have a considerable impact on smart contract execution, such as allowing public access to critical functions.
Low	Low-level vulnerabilities are primarily associated with outdated or unused code snippets that generally do not significantly impact execution, sometimes they can be ignored.
Informational	Informational vulnerabilities, code style violations, and informational statements do not affect smart contract execution and can typically be disregarded.

Executive Summary

According to the Assure assessment, the Customer's smart contract is **Secured.**



Scope

Target Code And Revision

For this audit, we performed research, investigation, and review of the TIXER contracts followed by issue reporting, along with mitigation and remediation instructions outlined in this report.

Target Code And Revision

Project	Assure
Language	Solidity
Codebase	AssetBettingV3.sol [SHA256]: eba86a1f15ce4fba3a912bccd84162ac06be78cd5c5e74ccaa33ed94f740967a Fixed version AssetBettingV4.sol [SHA256]: 40b9a89c50d4e64857aeafa250f3dfc64376a7491ea88be4575e39d2ff15805a Version 5: AssetBettingV5.sol [SHA256]: 2f094bafd7dce1b360506f89762a0f25fd9f77b8457baedddcb5f272ed888555
Audit Methodology	Static, Manual

Attacks made to the contract

In order to check for the security of the contract, we tested several attacks in order to make sure that the contract is secure and follows best practices.

Category	Item
Code review & Functional Review	• Compiler warnings. • Race conditions and Reentrancy. Cross-function race conditions. • Possible delays in data delivery. • Oracle calls. • Front running. • Timestamp dependence. • Integer Overflow and Underflow. • DoS with Revert. • DoS with block gas limit. • Methods execution permissions. • Economy model. • Private user data leaks. • Malicious Event log. • Scoping and Declarations. • Uninitialized storage pointers. • Arithmetic accuracy. • Design Logic. • Cross-function race conditions. • Safe Zeppelin module. • Fallback function security. • Overpowered functions / Owner privileges

AUDIT OVERVIEW

1. Fee Logic Missing [Fixed ✅]

Function: -

Issue: FEE_PERCENTAGE & feeCollector declared but never applied, so no fees are ever collected.

Recommendation: Either remove the unused fee parameters or implement fee deduction (for example deduct amount * FEE_PERCENTAGE / BASIS_POINTS in deposit/withdrawFor and send to feeCollector).

Fix: Unused parameters were removed.

2. Unrestricted Emergency Drain [Fixed ✅]

Function: recoverERC20,distributeTokens()

Issue: Owner can drain all USDC/USDT user deposits at any time, stealing funds.

Recommendation: Restrict recoverERC20 to only non-USDC/USDT tokens, or require a timelock/multisig, or split into recoverFees that only withdraws accrued fees tracked separately.

Fix: Unrestricted emergency-drain vectors have been removed/restricted. Users retain their deposits (and even emergencyWithdraw remains user-only after a 24 h delay), and the owner no longer has a unilateral drain all function.

3. Unbounded Recipients Array [Fixed ✅]

Function: distributeTokens(address,uint256,recipients[])

Issue: Large recipient arrays can exhaust gas and revert, potentially locking funds.

Recommendation: Enforce a maximum recipients.length (for example ≤ 20), or paginate distributions or consider to remove dynamic array and use only hard-coded recipients.

Fix: The dynamic-parameter–based distributeTokens() is gone.

V4 introduces a single-owner–set, in-contract DistributionRecipient[] distributionRecipients, populated only once at initialization, with:

```
require(_investors.length > 0, 'No recipients provided');
require(_investors.length <= 5, 'Too many recipients');
```

The owner can no longer pass in a custom array on each call there is a fixed list (max 5) enforced on setup, so gas usage is bounded and predictable.

4. No ERC20 Validation [Fixed ✅]

Function: updateToken

Issue: Owner could set a non-ERC20 address, breaking deposits/withdrawals and locking funds.

Recommendation: After newTokenAddress != address(0), perform a low-gas check like IERC20(newTokenAddress).balanceOf(address(this)) to confirm ERC20 compliance (reverting if not).
Fix: There is no longer any updateToken or similar owner-only setter. USDC and USDT token addresses are supplied once in the constructor (with a require(_usdcToken != address(0)/_usdtToken != address(0)) check) and cannot be changed thereafter so removing the mutable setter entirely, there's no pathway for the owner to point at a non-ERC20 contract.

5. No separate Fee Accounting [Fixed ✅]

Function: -

Issue: If fees are implemented, fee funds mix with user balances, risking accidental withdrawal.

Recommendation: Maintain a distinct mapping(address => uint256) feeBalances; and update it on each fee capture, so that user withdrawals can never touch feeBalances.

Fix: As described in [1.], fees were completely removed.

6. Centralized Withdrawal Authority [Fixed ✅]

Function: withdrawFor

Issue: Only owner can withdraw for users funds can be censored or permanently locked if owner fails.

Recommendation: Introduce a user-initiated withdraw(uint256 amount) entrypoint gated by off-chain validation or a timelock fallback allowing users to withdraw themselves after X days.

Fix: The withdrawFor function has been removed entirely.

Users now withdraw themselves via two permissionless entry points:

1. processWithdrawals() anyone (including the user) can submit a Merkle proof batch to trigger on-chain payouts.
2. emergencyWithdraw(token) after a 24 h delay from the last cycle's Merkle root, any user can pull out their entire balance without owner involvement.

7. Integer Rounding Loss [Fixed ✅]

Function: distributeTokens…hardcoded

Issue: (totalAmount * pct) / 10000 can leave a residual remainder stranded in contract.

Recommendation: Track distributed sum in the loop and, on the last recipient, assign amount = totalAmount - distributed to ensure full distribution.

Fix: In distributePlatformTokens, V4 keeps a running distributed tally and on the last recipient does:

```
uint256 amount = (i == distributionRecipients.length - 1)
    ? totalAmount - distributed
    : (totalAmount * distributionRecipients[i].percentage) / 10000;
```

By assigning the final slice as totalAmount - distributed, any leftover from rounding gets absorbed on the last transfer, ensuring the full balance is always distributed.

8. Owner-Privileged Drain via Fee Claims [Acknowledge]

Function: updateInvestorsVault() and updateAffiliateBank()

Issue: Owner can updateInvestorsVault() and updateAffiliateBank() at any time. Owner can set arbitrary merkleRoot for any cycleId with setMerkleRoot(). _processPlatformFee() and _processAffiliateFee() do not reduce any liabilities (for example userBalances) and only check IERC20(token).balanceOf(this) >= amount before transferring. Thus, the owner can set vault/bank to an attacker-controlled address, set root with a huge fee leaf, and drain the entire token balance even if it backs userBalances.

Recommendation: Enforce solvency invariant: contractTokenBalance >= sum(userBalances) must always hold. Track total liabilities and surplus; allow platform/affiliate fee transfers only from surplus (PnL / house edge), never from user liabilities. Alternatively, segregate fee funds in a separate dedicated escrow (distinct token flow), or burn/skim from realized PnL buckets off-chain and deposit fee tokens before claims.

Add invariant checks in setMerkleRoot() and/or fee processing (for example require(contractBalance - feeAmount >= totalUserBalances)). Consider restricting admin keys (multisig/timelock) and add on-chain guardian caps on per-cycle fee amounts.

9. Double-spend via emergency withdrawal paying historical deposits [Fixed ✅]

Function: deposit(uint256 amount) processWithdrawals(WithdrawalData[] calldata) _processIndividualWithdrawal(...) emergencyWithdraw()

Issue: User balances are credited on deposit() but not reduced on normal Merkle withdrawals, or the emergency path paid from the historical deposited total. A user could first withdraw normally via the Merkle path and later call emergencyWithdraw() to receive (again) the same funds, resulting in a double-spend.

Recommendation: Always decrement per-user balances on normal withdrawals before transferring. In emergencyWithdraw(), set the user's balance to 0 before transfer (checks-effects-interactions). Track per-leaf / per-cycle claims (claimedWithdrawals, processedLeaves) to prevent replay with different leaves. Keep nonReentrant on external entry points.

Fix: Normal withdrawals now reduce balances and Emergency withdraw zeroes balance before transfer.

1. Unnecessary String Storage [Fixed ✅]

Function: deposit/withdrawFor

Issue: Storing a string transactionType in each record costs ~16 k gas more per tx.

Recommendation: Replace string transactionType with enum TxType { DEPOSIT, WITHDRAWAL } to reduce storage and simplify logic.

Fix: V4 eliminates the stored string entirely. There is no longer any transactionType field in the user-balance records or in any structs. Instead, V4 relies on distinct functions (and events) to distinguish deposits vs. withdrawals, removing the need to persist a string or even an enum on-chain.

2. Gas Heavy Arraw Shifting [Fixed ✅]

Function: _recordTransaction

Issue: Looping to shift elements on every tx costs ~45 k extra gas once full.

Recommendation: Use a circular buffer with a head pointer per user, overwriting at head % MAX_TRANSACTIONS to keep costs constant.

Fix: V4 removes any on-chain per-user transaction array and the _recordTransaction logic entirely. Instead of tracking every individual deposit/withdrawal in a shifting array, the contract only maintains aggregate balances and relies on Merkle-proof–driven distributions.

3. Overloaded Distribute Functions [Fixed ✅]

Function: -

Issue: Owner may call the wrong overload, leading to unexpected behavior.

Recommendation: Rename the two distributions for example distributeCustomRecipients vs. distributeConfiguredRecipients to avoid ABI confusion.

Fix: V4 exposes only one distribution entrypoint—distributePlatformTokens and has removed any other distribute overloads.

1. Missing Event Emission [Fixed ✅]

Function: recoverERC20

Issue: Emergency token recoveries are not logged on-chain.

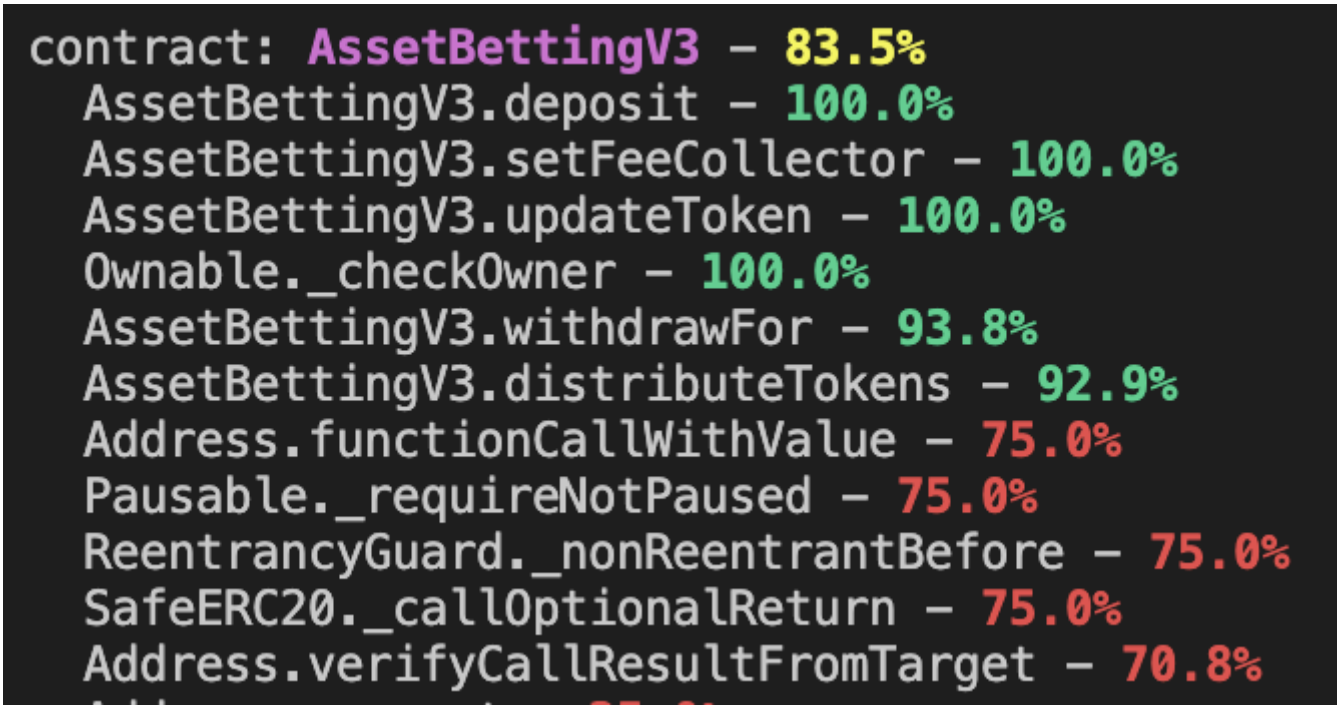
Recommendation: Emit a Recovered(address indexed token, uint256 amount) event at the end of recoverERC20.

Fix: V4 removes the recoverERC20 function entirely (eliminating the unlogged emergency-drain path).

No informational issues were found.

Testing coverage

During the testing phase, custom use cases were written to cover all the logic of contracts. *Check "Annexes" to see the testing code.*




```
tests/test_asset_betting.py::test_constructor RUNNING
Transaction sent: 0xc1e416c846a655cd01e35ae22dec6f592a1ade389309db0ffa09bfe32fda3735
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  ERC20Mock.constructor confirmed   Block: 1   Gas used: 619675 (5.16%)
  ERC20Mock deployed at: 0x3194cBDC3dbcd3E11a07892e7bA5c3394048Cc87

Transaction sent: 0xa495294734ab4b52ebfa20d23aaeefff79a9386632f40ef2c51585515d39925f
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  ERC20Mock.constructor confirmed   Block: 2   Gas used: 619675 (5.16%)
  ERC20Mock deployed at: 0x602C71e4DAC47a042Ee7f46E0aee17F94A3bA0B6

Transaction sent: 0x9f4974220bb011d04e9421fce480fe9bb2a5ea5ad3b010cfb72d5b6ec162cfe2
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 2
  AssetBettingV3.constructor confirmed (Invalid fee collector)   Block: 3   Gas used: 263099 (2.19%)

Transaction sent: 0x67549adae80d5b9010cfe8f2822fe9b24bed856a85444b9471a5520fed42a215
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 3
  AssetBettingV3.constructor confirmed (Invalid USDC token address)   Block: 4   Gas used: 263145 (2.19%)

Transaction sent: 0xa6b495dbaeefb815806161234e3b06d1f84412d9a4cf87ce6dfa61d01470b5be
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 4
  AssetBettingV3.constructor confirmed (Invalid USDT token address)   Block: 5   Gas used: 263180 (2.19%)

Transaction sent: 0x1cb390ede2bb29c45bc807f2d23716609aa9380034081b2f1034d9020360f8d6
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 5
  AssetBettingV3.constructor confirmed   Block: 6   Gas used: 2386114 (19.88%)
  AssetBettingV3 deployed at: 0x6b4BDe1086912A6Cb24ce3dB43b3466e6c72AFd3

tests/test_asset_betting.py::test_constructor PASSED
```

```
tests/test_asset_betting.py::test_deposit RUNNING
Transaction sent: 0x0794acaac0e75562a5f428ebf38ede583cf9be922462453c46a92ad45fd1b7b5
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 6
  ERC20Mock.constructor confirmed   Block: 7   Gas used: 619675 (5.16%)
  ERC20Mock deployed at: 0x9E4c14403d7d9A8A782044E86a93CAE09D7B2ac9

Transaction sent: 0x19da091acdf48464a4a39058eb7c39a90250c415b53fa3f44c5b4866881fb128
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 7
  ERC20Mock.constructor confirmed   Block: 8   Gas used: 619675 (5.16%)
  ERC20Mock deployed at: 0xcCB53c9429d32594F404d01fbe9E65ED1DCda8D9

Transaction sent: 0x1bf5eed8eae2b8f5f4171bc871c1d5db841390963f761c71e858cf028b42059e
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 8
  ERC20Mock.constructor confirmed   Block: 9   Gas used: 619687 (5.16%)
  ERC20Mock deployed at: 0x420b1099B9eF5baba6D92029594eF45E19A04A4A

Transaction sent: 0xf2d0812779dc1b5db5a481f5fce8ea1511b43e050f6695e830dde266fc9bffac
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 9
  AssetBettingV3.constructor confirmed   Block: 10   Gas used: 2386114 (19.88%)
  AssetBettingV3 deployed at: 0xa3B53dDCd2E3fC28e8E130288F2aBD8d5EE37472

Transaction sent: 0x03b49075fbad0b92e73eb78898648fa7bbdf2c36f1f1003807c210fdbdd27e48
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  AssetBettingV3.deposit confirmed (Unsupported token)   Block: 11   Gas used: 30507 (0.25%)

Transaction sent: 0x8c3ac869b4693e96e2d501f1e3fc26cffd9ead7ef30ff0f538a121914647d313
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  AssetBettingV3.deposit confirmed (Must deposit non-zero amount)   Block: 12   Gas used: 29629 (0.25%)

Transaction sent: 0xa2f60068575a5030c7e5b5e486ad7f43293876f25d729c6c5b7824303aa9b417
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 2
  AssetBettingV3.deposit confirmed (ERC20: insufficient allowance)   Block: 13   Gas used: 33169 (0.28%)

Transaction sent: 0xf4fb77d34c6f50f62a35a7f1453f907309e1fce0000df2c3908ce4bcf7e7c69a
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 10
  ERC20Mock.mint confirmed   Block: 14   Gas used: 65601 (0.55%)

Transaction sent: 0x51bdf20b06ce43b3a1ed1553dd8b2660511c46ac54f692fdefb1e6b0d05fefa1
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 11
  ERC20Mock.mint confirmed   Block: 15   Gas used: 65589 (0.55%)

Transaction sent: 0x24d7254f62165c801d45de4a59a1a8e6ae5f63ceba4175e699665d08d9e4872b
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 3
  ERC20Mock.approve confirmed   Block: 16   Gas used: 44124 (0.37%)

Transaction sent: 0x57de51862a2b53235e449f4310bc1a61f7521f732c7db3a67d844b1a9e9596ae
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  ERC20Mock.approve confirmed   Block: 17   Gas used: 44124 (0.37%)

Transaction sent: 0xf61b78bc184be2e2f5a7912ed6c73c529e931601ad96ec8c7965363fb1e176cb
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 4
  AssetBettingV3.deposit confirmed   Block: 18   Gas used: 238226 (1.99%)

Transaction sent: 0x837d59920a5777a66be67a1f5edc96cad11f05b7d4c6638672a8ec13af987bde
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  AssetBettingV3.deposit confirmed   Block: 19   Gas used: 239064 (1.99%)

Transaction sent: 0xabdc4f407949f3dc6a8ffbcaf440cfc9aa3b7e6ea27af491b3aa9662e4c5e3bb
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 5
  AssetBettingV3.deposit confirmed   Block: 20   Gas used: 178226 (1.49%)

Transaction sent: 0x06bba247a0a16b1233743f1b919057c69acd11b9f340c301d738aaf50e760f45
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 6
  AssetBettingV3.deposit confirmed   Block: 21   Gas used: 178226 (1.49%)

Transaction sent: 0x4b8ca51130cdf6d711cc33d65d1af1f9b1c15bc822ebfae40d12e52712ebdf2a
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 7
  AssetBettingV3.deposit confirmed   Block: 22   Gas used: 178226 (1.49%)

Transaction sent: 0xba4fe0a307eb27039d35b0c10a63913587117cf0d6d645b170fa3fc3397e5650
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 8
  AssetBettingV3.deposit confirmed   Block: 23   Gas used: 178214 (1.49%)

Transaction sent: 0xa2e9637c2459675db15fd2b30627632540a6e858472aac71aca2115159acec45
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 9
  AssetBettingV3.deposit confirmed   Block: 24   Gas used: 178226 (1.49%)

Transaction sent: 0x1997b85dcf23395fedd1e03d030a9f3a4d6335b0748ef924aca8822f29777547
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 10
  AssetBettingV3.deposit confirmed   Block: 25   Gas used: 178226 (1.49%)

Transaction sent: 0x28b6ff2ba0d35154533194db9f211af7c7ab10bfdcf8cb153a2b0eae68ddc81c
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 11
  AssetBettingV3.deposit confirmed   Block: 26   Gas used: 178226 (1.49%)

Transaction sent: 0xe33bb3cbae5b9045e9fad637d0e2e52abdf4b788ebb17344a3e7ffd02de1e803
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 12
  AssetBettingV3.deposit confirmed   Block: 27   Gas used: 178214 (1.49%)

Transaction sent: 0x38bc8c8e0d4cc7a99f9cc4e2d515b8f4c394eebc896ee47fce30422ac794930c
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 13
  AssetBettingV3.deposit confirmed   Block: 28   Gas used: 178226 (1.49%)

Transaction sent: 0xc22b02ee6e13b349024cc221fd4946962992465b723bf3563c5c3a68cf5ac9c9
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 14
  AssetBettingV3.deposit confirmed   Block: 29   Gas used: 240579 (2.00%)

Transaction sent: 0x4b48e3029198b898a115bdcb88eecb6f1d2ce5ed946292c4356e96f63870d6ca
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 15
  AssetBettingV3.deposit confirmed   Block: 30   Gas used: 240579 (2.00%)

Transaction sent: 0xb62c2d88ee3b2a75b9186fc8e095dbdba89aad30c47a9f2650c7382db9253362
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 16
  AssetBettingV3.deposit confirmed   Block: 31   Gas used: 240567 (2.00%)
```

```
tests/test_asset_betting.py::test_withdraw RUNNING
Transaction sent: 0xc15f7f668423fef1422650cd1ed8ed59c877fa9241cf2142e261069ad17469ef
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 12
  ERC20Mock.constructor confirmed   Block: 35   Gas used: 619675 (5.16%)
  ERC20Mock deployed at: 0x2c15A315610Bfa5248E4CbCbd693320e9D8E03Cc

Transaction sent: 0x5413e146f489b14a68974d72825e20fd3469d463454543c9f3d539694e2ef55a
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 13
  ERC20Mock.constructor confirmed   Block: 36   Gas used: 619675 (5.16%)
  ERC20Mock deployed at: 0xe692Cf21B12e0B2717C4bF647F9768Fa58861c8b

Transaction sent: 0x37c944f8116d40bb3708801a8988f20f30f9ade68dfd533156411320d4c43cbf
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 14
  ERC20Mock.constructor confirmed   Block: 37   Gas used: 619687 (5.16%)
  ERC20Mock deployed at: 0xe65A7a341978d59d40d30FC23F5014FACB4f575A

Transaction sent: 0xf091d0283978bfc963da7cc0561f2ebe286e78e7cab5a40cd052e7057b563d9e
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 15
  AssetBettingV3.constructor confirmed   Block: 38   Gas used: 2386114 (19.88%)
  AssetBettingV3 deployed at: 0x30375B532345B01cB8c2AD12541b09E9Aa53A93d

Transaction sent: 0x92b06544055a1021316de83afaf817123a52f4b136487bbd9bdb1da819d2ed50
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 16
  ERC20Mock.mint confirmed   Block: 39   Gas used: 65601 (0.55%)

Transaction sent: 0x11c3cc6cdbaa21da9fed14e5d895e49d538eb43b5cb568560c5d21c1e0e66ccc
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 17
  ERC20Mock.mint confirmed   Block: 40   Gas used: 65589 (0.55%)

Transaction sent: 0xa3930135e10c1fbc37fdbf1fc5f61afd58f605869e04007d34c52164deaba6b8
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 20
  ERC20Mock.approve confirmed   Block: 41   Gas used: 44124 (0.37%)

Transaction sent: 0x644db1363632f41d24e96b473c24d3f52c1b025e91c111e74702c5f584abfab2
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 2
  ERC20Mock.approve confirmed   Block: 42   Gas used: 44124 (0.37%)

Transaction sent: 0x01d609579d57bce02c29687fe134204fca40632a2d9815cd945f5676248bbcbd
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 21
  AssetBettingV3.withdrawFor confirmed (Ownable: caller is not the owner)   Block: 43   Gas used: 23402 (0.20%)

Transaction sent: 0xf35e0f3786a2da251f80a49f74087926df14987323b581449fdf87b09ddb56a6
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 18
  AssetBettingV3.withdrawFor confirmed (Unsupported token)   Block: 44   Gas used: 30993 (0.26%)

Transaction sent: 0x14e1f8bd474c3c1d7eb8f458b4366d84a8d51313ce86aa566d578c0e4e8d5712
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 19
  AssetBettingV3.withdrawFor confirmed (Must withdraw non-zero amount)   Block: 45   Gas used: 30115 (0.25%)

Transaction sent: 0x2fc5cf71b85b82e614dd48cce7b667d10496f021bc49a2838b74fcba84e32aac
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 20
  AssetBettingV3.withdrawFor confirmed (Insufficient balance)   Block: 46   Gas used: 31148 (0.26%)

Transaction sent: 0x6167967d1ab888c10c40093f5448f841ce5d32801185f0915e595c25b45baf72
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 22
  AssetBettingV3.deposit confirmed   Block: 47   Gas used: 238226 (1.99%)

Transaction sent: 0x76f3000f6d18abfb32a1582355a18862951ad78794c1851e5379cf23335c732f
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 21
  AssetBettingV3.withdrawFor confirmed   Block: 48   Gas used: 171364 (1.43%)

Transaction sent: 0x4a86cd38ca4138c25e2c6e9a85fdea8a8c41eb8f4c4c97a3d894682fd15585ad
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 22
  AssetBettingV3.withdrawFor confirmed   Block: 49   Gas used: 171364 (1.43%)

Transaction sent: 0xac780bf33e2186099848723f9c21fc330ff69d8a046b31cc7c213582a4f1e89f
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 23
  AssetBettingV3.withdrawFor confirmed (Insufficient balance)   Block: 50   Gas used: 31160 (0.26%)

tests/test_asset_betting.py::test_withdraw PASSED
tests/test_asset_betting.py::test_set_fee_collector RUNNING
Transaction sent: 0x1430e5c3135365e8735c81a7e9dded7bacbbd17c9e91afab246a86df92124091
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 24
  ERC20Mock.constructor confirmed   Block: 51   Gas used: 619675 (5.16%)
  ERC20Mock deployed at: 0xf9C8Cf55f2E520B08d869df7bc76aa3d3ddDF913

Transaction sent: 0xa9c6be77128afd6c6d4e54ee500b8b6a2d461e8be1ceffc4da8f393b0df0d622
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 25
  ERC20Mock.constructor confirmed   Block: 52   Gas used: 619675 (5.16%)
  ERC20Mock deployed at: 0x654f70d8442EA18904FA1AD79114f7250F7E9336

Transaction sent: 0x220c7da914a39a58b631e8d02d5ac98f943a96319f7368a917b97124c7b9f717
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 26
  AssetBettingV3.constructor confirmed   Block: 53   Gas used: 2386114 (19.88%)
  AssetBettingV3 deployed at: 0xADeD61D42dE86f9058386D1D0d739d20C7eAfC43

Transaction sent: 0xd4d5c08e2659fdabfb0d5e0c3e7262414ea8507f7df074f9f5a87584468d4edb
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 23
  AssetBettingV3.setFeeCollector confirmed (Ownable: caller is not the owner)   Block: 54   Gas used: 22754 (0.19%)

Transaction sent: 0x8ee678808f5dceef1c47fe096cfc4864550f4a7cd1f51b7da39f2921b395c63c
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 27
  AssetBettingV3.setFeeCollector confirmed (Invalid fee collector)   Block: 55   Gas used: 22573 (0.19%)

Transaction sent: 0x873677525151 69cb6e4f8bd00b68c77517243aea048b12002bafd2aab5f38334
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 28
  AssetBettingV3.setFeeCollector confirmed   Block: 56   Gas used: 30094 (0.25%)

tests/test_asset_betting.py::test_set_fee_collector PASSED
```

```
tests/test_asset_betting.py::test_update_token RUNNING
Transaction sent: 0x0580bfbfcc133e3cb196e3f4301130f03a50a7f6923d17ce541c5c3f9cc723d3
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 29
  ERC20Mock.constructor confirmed   Block: 57   Gas used: 619675 (5.16%)
  ERC20Mock deployed at: 0x42E8D004c84E6B5Bad559D3b5CE7947AADb9E0bc

Transaction sent: 0x925c8d41640fbbcaa80312b0ce6044e357093d0bdb911bd6b4064721aebbb15c
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 30
  ERC20Mock.constructor confirmed   Block: 58   Gas used: 619723 (5.16%)
  ERC20Mock deployed at: 0xF06D5f5BfFFCB6a52c84cfebc03AD35637728E73

Transaction sent: 0xc5ec84c54d9c60060b6aee011eb1f4c3ff1d5e25ea66321c23875453c9f87aca
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 31
  ERC20Mock.constructor confirmed   Block: 59   Gas used: 619675 (5.16%)
  ERC20Mock deployed at: 0x82c83b7f88aef2eD99d4869D547b6ED28e69C8df

Transaction sent: 0xae26234249430349f50e648bec195243d202ddeb0da4cf746d817242e6ead321
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 32
  ERC20Mock.constructor confirmed   Block: 60   Gas used: 619723 (5.16%)
  ERC20Mock deployed at: 0x724Ca58E1e6e64BFB1E15d7Eec0fe1E5f581c7bD

Transaction sent: 0xdf8781db4629ebcc7fabab1b2c706392c322d440fc344c9b738e040a1304d437
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 33
  AssetBettingV3.constructor confirmed   Block: 61   Gas used: 2386114 (19.88%)
  AssetBettingV3 deployed at: 0x34b97ffa01dc0DC959c5f1176273D0de3be914C1

Transaction sent: 0x14ebda6b55174259d99de01bd4dbb40c0993005c456ee498adb8e1c7bdfa236e
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 24
  AssetBettingV3.updateToken confirmed (Ownable: caller is not the owner)   Block: 62   Gas used: 22994 (0.19%)

Transaction sent: 0x2ca1cefc901ffb65af16fe036f2f7b34ce58ef7d411fe90403444cd371dd40f0
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 34
  AssetBettingV3.updateToken confirmed (Invalid token address)   Block: 63   Gas used: 22801 (0.19%)

Transaction sent: 0xf27e6e1e5c9c29e477c51bcbe44c391004ccfbb9f25e343b534e2eb87b8c240b
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 35
  AssetBettingV3.updateToken confirmed (Invalid token type)   Block: 64   Gas used: 23112 (0.19%)

Transaction sent: 0x8c22510eafa05b061d93a24032274f99354c1d12ae02b20f76ffd3143b0bd703
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 36
  AssetBettingV3.updateToken confirmed   Block: 65   Gas used: 30833 (0.26%)

Transaction sent: 0xb37ae67f4065bc1684217905c9a49f2b3c6c53326a8721dbaf6d72a227412aed
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 37
  AssetBettingV3.updateToken confirmed   Block: 66   Gas used: 30874 (0.26%)

tests/test_asset_betting.py::test_update_token PASSED
```

```
tests/test_asset_betting.py::test_distribute_tokens RUNNING
Transaction sent: 0xc3bb65e9f87cfafc26c91392f552ad9fdcca33eaa103f11ca98fa8757ca4abe0
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 38
  ERC20Mock.constructor confirmed   Block: 67   Gas used: 619675 (5.16%)
  ERC20Mock deployed at: 0xFE0F4Cf81B5c0a6Fd65a610FD9488F33aE9095cB

Transaction sent: 0xa30c0d04f95e49760d460e0d42767e465edc07a1410bf2710f414537ca7478a8
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 39
  ERC20Mock.constructor confirmed   Block: 68   Gas used: 619675 (5.16%)
  ERC20Mock deployed at: 0x4e3E7dC9D84dA7BE8f017f4C36153A61341736d4

Transaction sent: 0x5740069460c7e1960293bfb29dbff191b18c18bf81e02dd9df7c7ac826281fcd
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 40
  ERC20Mock.constructor confirmed   Block: 69   Gas used: 619687 (5.16%)
  ERC20Mock deployed at: 0x0AC45e945A008D3fc19da8f591be8601C1F63130

Transaction sent: 0xfff3c0e23970ebb0b8968ae086f4d0ccb709900f12e7f733e43c08dc7925a45c
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 41
  AssetBettingV3.constructor confirmed   Block: 70   Gas used: 2386114 (19.88%)
  AssetBettingV3 deployed at: 0x5847798CE8c89e3Fff59AE5fA30BEC0d406b5687

Transaction sent: 0xed673fc4d770ca27233a4977864cfa344b28f58d744a79d2b2158eb7f60d503a
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 42
  ERC20Mock.mint confirmed   Block: 71   Gas used: 65601 (0.55%)

Transaction sent: 0x760d31e614b10ed9576e5f3da32431a7e3dff6d4900dbf88d4810f6687d74bc4
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 43
  ERC20Mock.mint confirmed   Block: 72   Gas used: 65589 (0.55%)

Transaction sent: 0xb3f6832a44a205ae41928b5e762ed709e6cc13e1a62eccef82c12737966c8801
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 25
  ERC20Mock.approve confirmed   Block: 73   Gas used: 44124 (0.37%)

Transaction sent: 0x758320e662aca2e527c60c9113688c16da3802acd5a0995d30be3d8ef13be7ca
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 3
  ERC20Mock.approve confirmed   Block: 74   Gas used: 44124 (0.37%)

Transaction sent: 0xdf397e283e557a1ec443f0b0058eadb215223fd1e46f2fe3e4b554ccfe0a47f1
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 26
  AssetBettingV3.distributeTokens confirmed (Ownable: caller is not the owner)   Block: 75   Gas used: 22952 (0.19%)

Transaction sent: 0x82e43ffa69ad3c3a039b40474cfdb4026c246632109a74481f83123e72bd68f3
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 44
  AssetBettingV3.distributeTokens confirmed (Unsupported token)   Block: 76   Gas used: 30531 (0.25%)

Transaction sent: 0xabaca596aa6ca03fe2749e3bfc05b3893c4291079e8a53a67d8a7e31c40fa1dc
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 45
  AssetBettingV3.distributeTokens confirmed (Must distribute non-zero amount)   Block: 77   Gas used: 29665 (0.25%)

Transaction sent: 0x882343fc88945948a577ca4a66a5e36cc26764dcae340b22cd5977f432e77c82
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 46
  AssetBettingV3.distributeTokens confirmed (Insufficient contract balance)   Block: 78   Gas used: 32778 (0.27%)

Transaction sent: 0xfc063128b22dcaff6b06be1b9a0963068a77ab341b353f221e849906edc20a61
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 27
  AssetBettingV3.deposit confirmed   Block: 79   Gas used: 238226 (1.99%)

Transaction sent: 0x888bb92f447d3bf69cf740894e3435f361a9a5b6756b979956443bd6b528f395
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 47
  AssetBettingV3.distributeTokens confirmed   Block: 80   Gas used: 116862 (0.97%)

tests/test_asset_betting.py::test_distribute_tokens PASSED
```

```
tests/test_asset_betting.py::test_distribute_tokens_recipients RUNNING
Transaction sent: 0x61c68516da630afdd7f78637cfcd639a28c00713ce72bb0a211e0bfc4b5ad6da
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 48
  ERC20Mock.constructor confirmed   Block: 81   Gas used: 619675 (5.16%)
  ERC20Mock deployed at: 0x8b1B440724DCe2EE9779B58af841Ec59F545838B

Transaction sent: 0x14f5169e445a660722999faa9b3e6127f3e5771f8b6ebe7e867a758a225fb03c
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 49
  ERC20Mock.constructor confirmed   Block: 82   Gas used: 619675 (5.16%)
  ERC20Mock deployed at: 0xC6D563d5c2243b27e7294511063f563ED701EA2C

Transaction sent: 0xc3fdc4cb60571bb1b82d710af5b70b60e162ef5455e07a4c34c5220e017a2804
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 50
  ERC20Mock.constructor confirmed   Block: 83   Gas used: 619687 (5.16%)
  ERC20Mock deployed at: 0xD537bF4b795b7D07Bd5F4bAf7017e3ce8360B1DE

Transaction sent: 0x7d8849c13cacc8f432f1797ce19bd0711721de181bb8c421ad3fadae5eb4ddaf
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 51
  AssetBettingV3.constructor confirmed   Block: 84   Gas used: 2386114 (19.88%)
  AssetBettingV3 deployed at: 0x70bC6D873D110Da59a9c49E7485a27B0F605E5db

Transaction sent: 0x38a0bbbd6c0330f39577519dd5e740ecbce2b6e5dff44789dc8cbb25d26a00c0
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 52
  ERC20Mock.mint confirmed   Block: 85   Gas used: 65601 (0.55%)

Transaction sent: 0xf7e6f7069dd5ed4b0f4a2f0b9d7d32bc7688f9cf2db70b0b989a914a3a5a3ac4
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 53
  ERC20Mock.mint confirmed   Block: 86   Gas used: 65589 (0.55%)

Transaction sent: 0x9a496cce7d5499f31eb20557ba506c788dc6b82c84ebe483266d67debca1ceb3
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 28
  ERC20Mock.approve confirmed   Block: 87   Gas used: 44124 (0.37%)

Transaction sent: 0x138b78887043549b804117898fd87d2b3e80666f0cbfd91546060548386385d0
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 4
  ERC20Mock.approve confirmed   Block: 88   Gas used: 44124 (0.37%)

Transaction sent: 0xb12b697ff42ebdbeadc765a37048d2bee73ed27b3ad4971eba7ef9c18efa6fc3
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 29
  AssetBettingV3.distributeTokens confirmed (Ownable: caller is not the owner)   Block: 89   Gas used: 23418 (0.20%)

Transaction sent: 0x1317c1ee5d3f5006c8af82b77cc394608997402c0425e157f7375c3bb7f10730
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 54
  AssetBettingV3.distributeTokens confirmed (Unsupported token)   Block: 90   Gas used: 31009 (0.26%)

Transaction sent: 0x912bc7b39aed78d3aeddbea16ed7035b30f80e81a68c039fee9be9d33266caf1
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 55
  AssetBettingV3.distributeTokens confirmed (Must distribute non-zero amount)   Block: 91   Gas used: 30131 (0.25%)

Transaction sent: 0xd7946136eb0bf0472c6a6cac2e95df841053abfccaf432cf632e740c35f572a1
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 56
  AssetBettingV3.distributeTokens confirmed (Recipients array cannot be empty)   Block: 92   Gas used: 30187 (0.25%)

Transaction sent: 0xb40f3a58915a286ae75e9cb24bb74f5d6caeb6edea031376c5f97fbba7ded219
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 57
  AssetBettingV3.distributeTokens confirmed (Insufficient contract balance)   Block: 93   Gas used: 32964 (0.27%)

Transaction sent: 0x19b09c4506c5d820f234e892be496e4cbd78ac6a32ebfa378662055d3b08a9e9
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 30
  AssetBettingV3.deposit confirmed   Block: 94   Gas used: 238226 (1.99%)

Transaction sent: 0xdc696fbf0a37b95e7ef1572c5974f7f50be0de80722ca30cafb84aeb90725846
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 58
  AssetBettingV3.distributeTokens confirmed (Invalid recipient address)   Block: 95   Gas used: 33028 (0.28%)

Transaction sent: 0xb7ef880f84c246009d9b2a634ece0e247952e9b6333ea4998e45dbc5de6af528
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 59
  AssetBettingV3.distributeTokens confirmed (Percentage must be greater than 0)   Block: 96   Gas used: 33333 (0.28%)

Transaction sent: 0x28a77b1fbe4b526f395e572c018e2355a3f0bbb30a73a326641b309394c8c18f
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 60
  AssetBettingV3.distributeTokens confirmed (Percentages must sum to 100%)   Block: 97   Gas used: 33542 (0.28%)

Transaction sent: 0xd7eba30f88e5bf0e924369bf2a2a8616336374d3d59c2deba12c2cf41dda4e93
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 61
  AssetBettingV3.distributeTokens confirmed   Block: 98   Gas used: 48764 (0.41%)

tests/test_asset_betting.py::test_distribute_tokens_recipients PASSED
```

Annexes

Testing code:

```
from brownie import (
    reverts,
)

from scripts.helpful_scripts import (
    ZERO_ADDRESS,
    DAY_TIMESTAMP,
    get_account,
    get_timestamp,
    get_chain_number,
    increase_timestamp
)

from scripts.deploy import (
    deploy_erc,
    deploy_betting
)

def test_constructor(only_local):
    # Arrange
    owner = get_account(0)
    other = get_account(1)
    extra = get_account(2)
    fee_collector = get_account(3)

    usdc_token = deploy_erc(owner, "USDC token", "USDC")
    usdt_token = deploy_erc(owner, "USDT token", "USDT")

    with reverts("Invalid fee collector"):
        deploy_betting(owner, ZERO_ADDRESS, usdc_token.address, usdt_token.address)
```

```
    with reverts("Invalid USDC token address"):
        deploy_betting(owner, fee_collector, ZERO_ADDRESS, usdt_token.address)
    with reverts("Invalid USDT token address"):
        deploy_betting(owner, fee_collector, usdc_token.address, ZERO_ADDRESS)

    betting = deploy_betting(owner, fee_collector, usdc_token.address,
usdt_token.address)

def test_deposit(only_local):
    # Arrange
    owner = get_account(0)
    other = get_account(1)
    extra = get_account(2)
    fee_collector = get_account(3)

    usdc_token = deploy_erc(owner, "USDC token", "USDC")
    usdt_token = deploy_erc(owner, "USDT token", "USDT")
    random_token = deploy_erc(owner, "Random token", "RDM")
    betting = deploy_betting(owner, fee_collector, usdc_token.address,
usdt_token.address)

    with reverts("Unsupported token"):
        betting.deposit(random_token.address, 10e6, {"from": other})
    with reverts("Must deposit non-zero amount"):
        betting.deposit(usdc_token.address, 0, {"from": other})
    with reverts("ERC20: insufficient allowance"):
        betting.deposit(usdc_token.address, 10e6, {"from": other})

    # mint some tokens
    usdc_token.mint(other, 1000e6)
    usdt_token.mint(extra, 1000e6)
    usdc_token.approve(betting.address, 1000e6, {"from": other})
    usdt_token.approve(betting.address, 1000e6, {"from": extra})

    tx = betting.deposit(usdc_token.address, 10e6, {"from": other})
```

```
    assert tx.events['TransactionRecorded'][0]['user'] == other
    assert tx.events['TransactionRecorded'][0]['transactionType'] == 'DEPOSIT'
    assert tx.events['TransactionRecorded'][0]['token'] == usdc_token.address
    assert tx.events['TransactionRecorded'][0]['amount'] == 10e6
    assert tx.events['Deposited'][0]['user'] == other
    assert tx.events['Deposited'][0]['token'] == usdc_token.address
    assert tx.events['Deposited'][0]['amount'] == 10e6

    tx = betting.deposit(usdt_token.address, 100e6, {"from": extra})
    assert tx.events['TransactionRecorded'][0]['user'] == extra
    assert tx.events['TransactionRecorded'][0]['transactionType'] == 'DEPOSIT'
    assert tx.events['TransactionRecorded'][0]['token'] == usdt_token.address
    assert tx.events['TransactionRecorded'][0]['amount'] == 100e6
    assert tx.events['Deposited'][0]['user'] == extra
    assert tx.events['Deposited'][0]['token'] == usdt_token.address
    assert tx.events['Deposited'][0]['amount'] == 100e6

    # deposit many tokens to create multiple transactions
    for n in range(15):
        betting.deposit(usdc_token.address, (n + 1) * 1e6, {"from": other})

def test_withdraw(only_local):
    # Arrange
    owner = get_account(0)
    other = get_account(1)
    extra = get_account(2)
    fee_collector = get_account(3)

    usdc_token = deploy_erc(owner, "USDC token", "USDC")
    usdt_token = deploy_erc(owner, "USDT token", "USDT")
    random_token = deploy_erc(owner, "Random token", "RDM")
    betting = deploy_betting(owner, fee_collector, usdc_token.address,
usdt_token.address)

    # mint some tokens
```

```
    usdc_token.mint(other, 1000e6)
    usdt_token.mint(extra, 1000e6)
    usdc_token.approve(betting.address, 1000e6, {"from": other})
    usdt_token.approve(betting.address, 1000e6, {"from": extra})

    with reverts("Ownable: caller is not the owner"):
        betting.withdrawFor(other, usdc_token.address, 10e6, {"from": other})
    with reverts("Unsupported token"):
        betting.withdrawFor(other, random_token.address, 10e6, {"from": owner})
    with reverts("Must withdraw non-zero amount"):
        betting.withdrawFor(other, usdc_token.address, 0, {"from": owner})
    with reverts("Insufficient balance"):
        betting.withdrawFor(other, usdc_token.address, 10e6, {"from": owner})

    betting.deposit(usdc_token.address, 100e6, {"from": other})
    tx = betting.withdrawFor(other, usdc_token.address, 50e6, {"from": owner})
    assert tx.events['TransactionRecorded'][0]['user'] == other
    assert tx.events['TransactionRecorded'][0]['transactionType'] == 'WITHDRAWAL'
    assert tx.events['TransactionRecorded'][0]['token'] == usdc_token.address
    assert tx.events['TransactionRecorded'][0]['amount'] == 50e6
    assert tx.events['Withdrawn'][0]['user'] == other
    assert tx.events['Withdrawn'][0]['token'] == usdc_token.address
    assert tx.events['Withdrawn'][0]['amount'] == 50e6

    tx = betting.withdrawFor(other, usdc_token.address, 25e6, {"from": owner})
    assert tx.events['Withdrawn'][0]['user'] == other
    assert tx.events['Withdrawn'][0]['token'] == usdc_token.address
    assert tx.events['Withdrawn'][0]['amount'] == 25e6

    with reverts("Insufficient balance"):
        betting.withdrawFor(other, usdc_token.address, 50e6, {"from": owner})

def test_set_fee_collector(only_local):
    # Arrange
    owner = get_account(0)
```

```
    other = get_account(1)
    extra = get_account(2)
    fee_collector = get_account(3)

    usdc_token = deploy_erc(owner, "USDC token", "USDC")
    usdt_token = deploy_erc(owner, "USDT token", "USDT")
    betting = deploy_betting(owner, fee_collector, usdc_token.address,
usdt_token.address)

    with reverts("Ownable: caller is not the owner"):
        betting.setFeeCollector(extra, {"from": other})
    with reverts("Invalid fee collector"):
        betting.setFeeCollector(ZERO_ADDRESS, {"from": owner})

    assert betting.feeCollector() == fee_collector
    betting.setFeeCollector(extra, {"from": owner})
    assert betting.feeCollector() == extra

def test_update_token(only_local):
    # Arrange
    owner = get_account(0)
    other = get_account(1)
    fee_collector = get_account(3)

    usdc_token = deploy_erc(owner, "USDC token", "USDC")
    new_usdc_token = deploy_erc(owner, "New USDC token", "USDC")
    usdt_token = deploy_erc(owner, "USDT token", "USDT")
    new_usdt_token = deploy_erc(owner, "New USDT token", "USDT")
    betting = deploy_betting(owner, fee_collector, usdc_token.address,
usdt_token.address)

    with reverts("Ownable: caller is not the owner"):
        betting.updateToken(0, new_usdc_token.address, {"from": other})
    with reverts("Invalid token address"):
        betting.updateToken(0, ZERO_ADDRESS, {"from": owner})
```

```
    with reverts("Invalid token type"):
        betting.updateToken(2, new_usdc_token.address, {"from": owner})

    tx = betting.updateToken(0, new_usdc_token.address, {"from": owner})
    assert tx.events['TokenUpdated'][0]['token'] == new_usdc_token.address
    tx = betting.updateToken(1, new_usdt_token.address, {"from": owner})
    assert tx.events['TokenUpdated'][0]['token'] == new_usdt_token.address

def test_distribute_tokens(only_local):
    # Arrange
    owner = get_account(0)
    other = get_account(1)
    extra = get_account(2)
    fee_collector = get_account(3)

    usdc_token = deploy_erc(owner, "USDC token", "USDC")
    usdt_token = deploy_erc(owner, "USDT token", "USDT")
    random_token = deploy_erc(owner, "Random token", "RDM")
    betting = deploy_betting(owner, fee_collector, usdc_token.address,
usdt_token.address)

    # mint some tokens
    usdc_token.mint(other, 1000e6)
    usdt_token.mint(extra, 1000e6)
    usdc_token.approve(betting.address, 1000e6, {"from": other})
    usdt_token.approve(betting.address, 1000e6, {"from": extra})

    with reverts("Ownable: caller is not the owner"):
        betting.distributeTokens(usdc_token.address, 100e6, {"from": other})
    with reverts("Unsupported token"):
        betting.distributeTokens(random_token.address, 100e6, {"from": owner})
    with reverts("Must distribute non-zero amount"):
        betting.distributeTokens(usdc_token.address, 0, {"from": owner})
    with reverts("Insufficient contract balance"):
        betting.distributeTokens(usdc_token.address, 100e6, {"from": owner})
```

```
    betting.deposit(usdc_token.address, 100e6, {"from": other})

    tx = betting.distributeTokens(usdc_token.address, 100e6, {"from": owner})
    assert tx.events['TokensDistributed'][0]['token'] == usdc_token.address
    assert tx.events['TokensDistributed'][0]['totalAmount'] == 100e6
    assert tx.events['TokensDistributed'][0]['recipientCount'] == 3

def test_distribute_tokens_recipients(only_local):
    # Arrange
    owner = get_account(0)
    other = get_account(1)
    extra = get_account(2)
    fee_collector = get_account(3)

    usdc_token = deploy_erc(owner, "USDC token", "USDC")
    usdt_token = deploy_erc(owner, "USDT token", "USDT")
    random_token = deploy_erc(owner, "Random token", "RDM")
    betting = deploy_betting(owner, fee_collector, usdc_token.address,
usdt_token.address)

    # mint some tokens
    usdc_token.mint(other, 1000e6)
    usdt_token.mint(extra, 1000e6)
    usdc_token.approve(betting.address, 1000e6, {"from": other})
    usdt_token.approve(betting.address, 1000e6, {"from": extra})

    with reverts("Ownable: caller is not the owner"):
        betting.distributeTokens(usdc_token.address, 100e6, [], {"from": other})
    with reverts("Unsupported token"):
        betting.distributeTokens(random_token.address, 100e6, [], {"from": owner})
    with reverts("Must distribute non-zero amount"):
        betting.distributeTokens(usdc_token.address, 0, [], {"from": owner})
    with reverts("Recipients array cannot be empty"):
        betting.distributeTokens(usdc_token.address, 100e6, [], {"from": owner})
```

```
    with reverts("Insufficient contract balance"):
        betting.distributeTokens(usdc_token.address, 100e6, [[extra, 10000]],
{"from": owner})

    betting.deposit(usdc_token.address, 100e6, {"from": other})

    with reverts("Invalid recipient address"):
        betting.distributeTokens(usdc_token.address, 100e6, [[ZERO_ADDRESS, 10000]],
{"from": owner})
    with reverts("Percentage must be greater than 0"):
        betting.distributeTokens(usdc_token.address, 100e6, [[extra, 0]], {"from":
owner})
    with reverts("Percentages must sum to 100%"):
        betting.distributeTokens(usdc_token.address, 100e6, [[extra, 2000]],
{"from": owner})

    tx = betting.distributeTokens(usdc_token.address, 100e6, [[extra, 10000]],
{"from": owner})
    assert tx.events['TokensDistributed'][0]['token'] == usdc_token.address
    assert tx.events['TokensDistributed'][0]['totalAmount'] == 100e6
    assert tx.events['TokensDistributed'][0]['recipientCount'] == 1
```

Technical Findings Summary

Findings

Vulnerability Level	Total	Pending	Not Apply	Acknowledged	Partially Fixed	Fixed
High	9			1		8
Medium	3					3
Low	1					1
Informational						

Assessment Results

Score Results

Review	Score
Global Score	**85/100**
Assure KYC	Not completed
Audit Score	85/100

The Following Score System Has been Added to this page to help understand the value of the audit, the maximum score is 100, however to attain that value the project must pass and provide all the data needed for the assessment. Our Passing Score has been changed to 84 Points for a higher standard, if a project does not attain 85% is an automatic failure. Read our notes and final assessment below. The Global Score is a combination of the evaluations obtained between having or not having KYC and the type of contract audited together with its manual audit.

Audit PASS

Following our comprehensive security audit of the token contract for the TIXER project, we inform you that the project has met the necessary security standards.

Disclaimer

Assure Defi has conducted an independent security assessment to verify the integrity of and highlight any vulnerabilities or errors, intentional or unintentional, that may be present in the reviewed code for the scope of this assessment. This report does not constitute agreement, acceptance, or advocating for the Project, and users relying on this report should not consider this as having any merit for financial adTIXER in any shape, form, or nature. The contracts audited do not account for any economic developments that the Project in question may pursue, and the veracity of the findings thus presented in this report relate solely to the proficiency, competence, aptitude, and discretion of our independent auditors, who make no guarantees nor assurance that the contracts are entirely free of exploits, bugs, vulnerabilities or deprecation of technologies.

All information provided in this report does not constitute financial or investment adTIXER, nor should it be used to signal that any person reading this report should invest their funds without sufficient individual due diligence, regardless of the findings presented. Information is provided ‘as is, and Assure Defi is under no covenant to audit completeness, accuracy, or solidity of the contracts. In no event will Assure Defi or its partners, employees, agents, or parties related to the provision of this audit report be liable to any parties for, or lack thereof, decisions or actions with regards to the information provided in this audit report.

The assessment serTIXERs provided by Assure Defi are subject to dependencies and are under continuing development. You agree that your access or use, including but not limited to any serTIXERs, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies with high levels of technical risk and uncertainty. The assessment reports could include false positives, negatives, and unpredictable results. The serTIXERs may access, and depend upon, multiple layers of third parties.