smart state

Web3 security easier than ever

MAGIC SQUARE
SQR token
Smart contract audit report

December 24, 2023

Table of contents

Methodology

- Manual code analysis
- Best code practices
- ERC20/BEP20 compliance (if applicable)
- Locked ether
- Pool Asset Security (backdoors in the underlying ERC-20)
- FA2 compliance (if applicable)
- Logical bugs & code logic issues
- Error handling issues
- General Denial Of Service(DOS)
- Cryptographic errors
- Weak PRNG / Random number generators issues
- Protocol and header parsing errors
- Private data leaks
- Using components with known vulnerabilities
- Unchecked call return method
- Code with no effects
- Unused vars
- Use of deprecated functions
- Authorization issues
- Re-entrancy
- Arithmetic Overflows / Underflows
- Hidden Malicious Code
- External Contract Referencing
- Short Address/Parameter Attack
- Race Conditions / Front Running
- Uninitialized Storage Pointers
- Floating Points and Precision
- Signatures Replay

Summary

This audit encompasses the examination of the smart contracts of the SQR token - Magic Square ecosystem token which can be used for the utilities within the Magic Square ecosystem, and can also be traded on DEX and CEX for other currencies or fiat.

Disclaimer

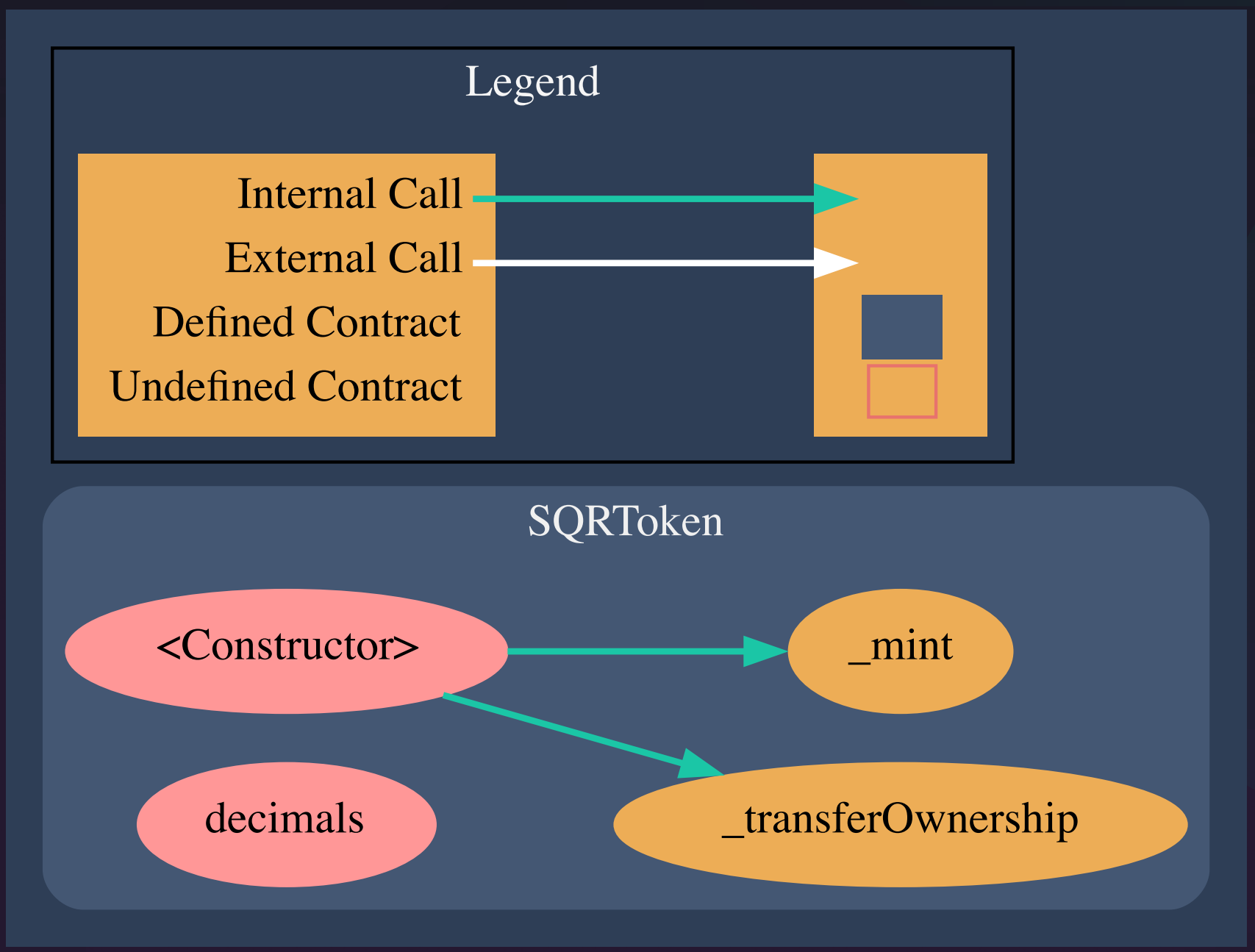
This is the final and public version of the security audit report and doesn't include vulnerabilities that might have been found and resolved during the audit process.
An audit does not provide any warranties regarding the code security. We presume that a single audit cannot be considered totally sufficient and always recommend several independent audits and a public bug bounty program to ensure code security.
Please do not consider this report as investment and / or financial advice of any kind.

Vulnerabilities found by type

Info	2
Warning	0
Warning	0
Total	2

1.1 Structure of contract:

SQRToken



pic.1.1 SQRToken

1.2 SQRToken contract methods analysis:

```
constructor(
    string memory name_,
    string memory symbol_,
    address newOwner,
    uint256 initMint,
    uint8 decimals_
  )
```

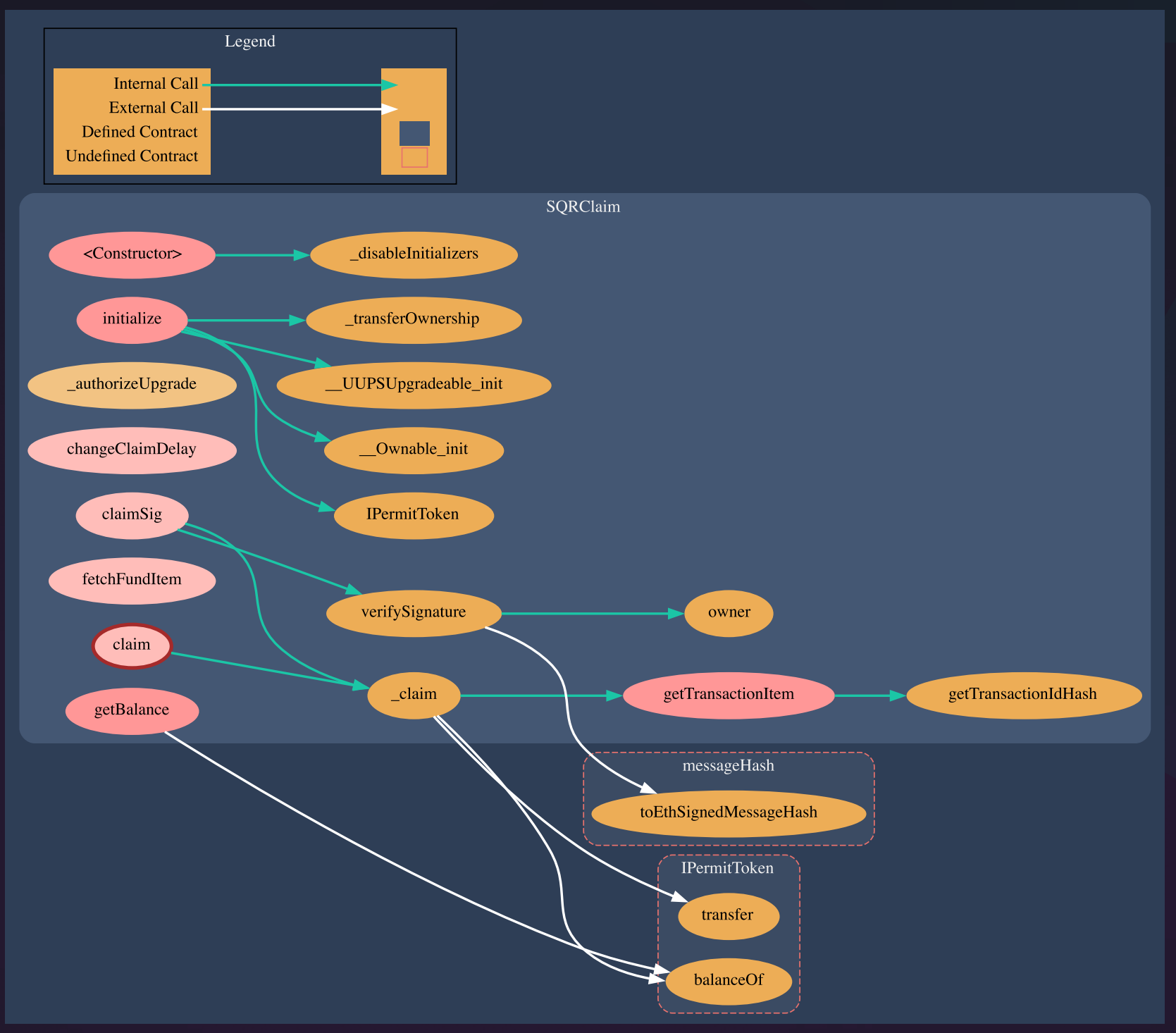
Vulnerabilities not detected

decimals()

Vulnerabilities not detected

2.1 Structure of contract:

SQRClaim



pic.2.1 SQRClaim

2.2 SQRClaim contract methods analysis:

constructor()
Vulnerabilities not detected

INFO

initialize(address _newOwner, address _sqrToken, uint32 _claimDelay)
_sqrToken parameter lacks 0 address check

_authorizeUpgrade(address newImplementation)
Vulnerabilities not detected

INFO

changeClaimDelay(uint32 _claimDelay)
Function should emit an event

getBalance()
Vulnerabilities not detected

getTransactionIdHash(string memory transactionId)
Vulnerabilities not detected

fetchFundItem(address account)
Vulnerabilities not detected

2.2 SQRClaim contract methods analysis:

```
getTransactionItem(
    string memory transactionId
  )
```

Vulnerabilities not detected

```
_claim(
    address account,
    uint256 amount,
    string memory transactionId,
    uint32 timestampLimit
  )
```

Vulnerabilities not detected

```
claim(
    address account,
    uint256 amount,
    string memory transactionId,
    uint32 timestampLimit
  )
```

Vulnerabilities not detected

TOKEN FLOW	Tokens out, onlyOwner

```
verifySignature(
    address account,
    uint256 amount,
    string memory transactionId,
    uint32 timestampLimit,
    bytes memory signature
  )
```

Vulnerabilities not detected

2.2 SQRClaim contract methods analysis:

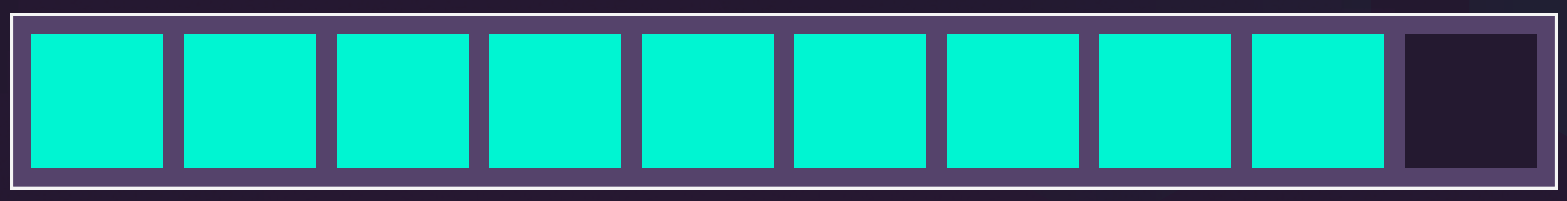
```
claimSig(
    address account,
    uint256 amount,
    string memory transactionId,
    uint32 timestampLimit,
    bytes memory signature
  )
```

Vulnerabilities not detected	
TOKEN FLOW	Tokens out, public

Verification checksums

Contract name	Bytecode hash(SHA-256)
SQRToken	becf674064291c4c66f3fb764ec5d394af02055efde104c9eabe2ad09981b03c
SQRClaim	a6468b8fce22a497d14aa27007dc031e6510d36cea7f0781b52d985502bfd2d4

Project evaluation



9/10

Get in touch 👋

@smartstatetech

@smartstate

@SmartStateAudit

@smartstatetech

@smartstate.tech

View this report on Smartstate.tech

info@smartstate.tech

smartstate.tech