

Smart Contract Test Plan

1. 测试环境介绍

本项目测试的合同包含4个，分别为DataUsage Contract, Agreement Contract, Log Contract和Verification Contract。采用Web3.js + Truffle的方式进行测试。从功能测试，非功能测试以及安全测试三个方面进行测试。

//测试结束后补充

2. 功能测试

此项目的功能测试分为单元测试，集成测试和系统测试。单元测试旨在测试合约的各个功能是否以需求为导向正常工作。集成测试旨在测试智能合约之间交互和其它相关系统组件的行为是否符合预期。系统测试旨在将测试无误的智能合约通过Metamask部署到如Goerli或Sepolia等以太网公共区块链测试网络进行测试。

2.1 单元测试

此项目将从如下几个方面进行单元测试。

- 状态检查：例如合约部署后检查初始状态是否正确，函数调用后检查状态是否改变且符合预期。一般状态检查为读取view函数。
- 事件触发：基本上，合约中的关键操作都应该触发事件进行相应追踪。在单元测试中测试事件是否触发，抛出的参数是否正确。
- 交易重置：在测试一些非预期条件时，交易应当重置并给出相应的原因。使用单元测试检测是否重置及错误原因是否相同。
- 函数计算：例如要计算不同条件下某函数的返回值，需要重复调用某个函数并输入不同的参数，看是否结果相符或符合需求预期。
- 完全功能测试：测试智能合约中的各项功能是否能够正常运行且符合预期。
- 测试覆盖率：包含代码覆盖率，函数覆盖率和分支覆盖率。

2.1.1 DataUsage Contract

DataUsage Contract中主要包含的测试内容如Figure 1所示。

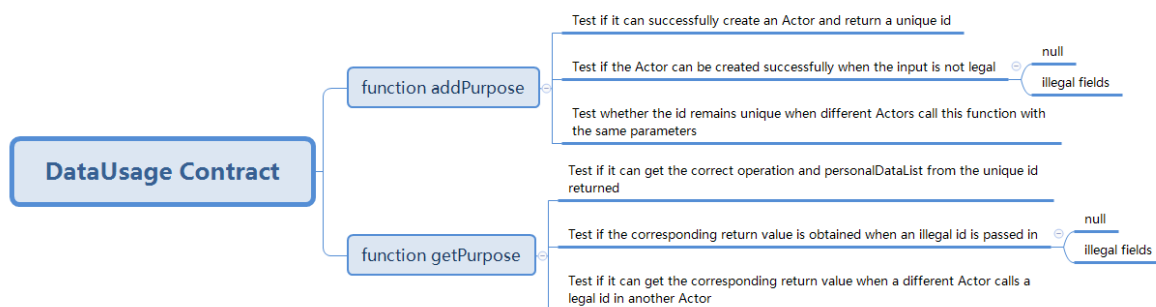


Figure 1. DataUsage Contract Test Plan

//以下为实际测试后填写内容。

1. function addPurpose

2. function getPurpose

2.1.2 Agreement Contract

Agreement Contract中主要包含的测试内容如Figure 2所示。

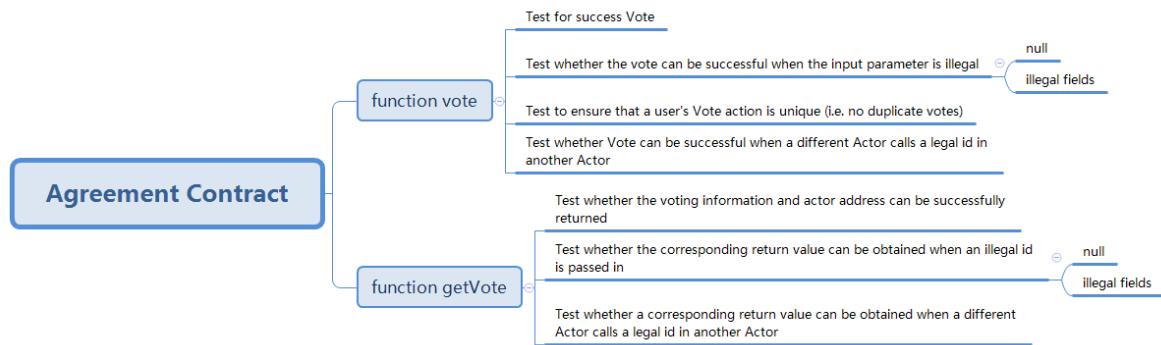


Figure 2. Agreement Contract Test Plan

1. function vote
2. function getVote

2.1.3 Log Contract

Log Contract中主要包含的测试内容如Figure 3所示。

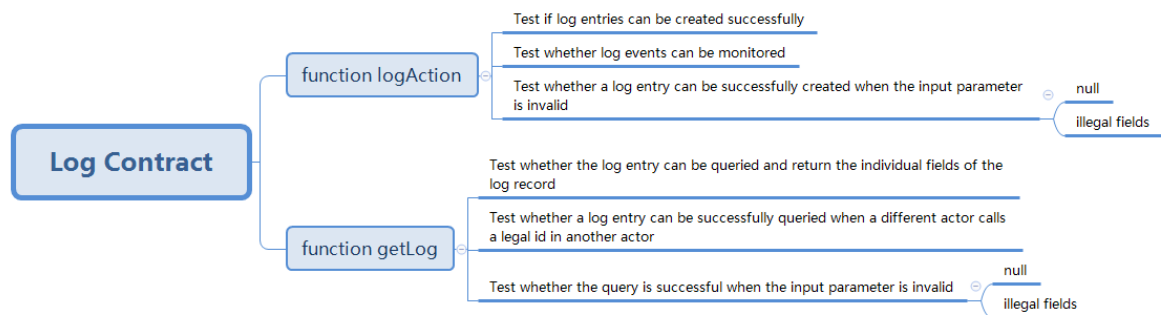


Figure 3. Log Contract Test Plan

1. function logAction
2. function getLog

2.1.4 Verification Contract

Verification Contract中主要包含的测试内容如Figure 4所示。

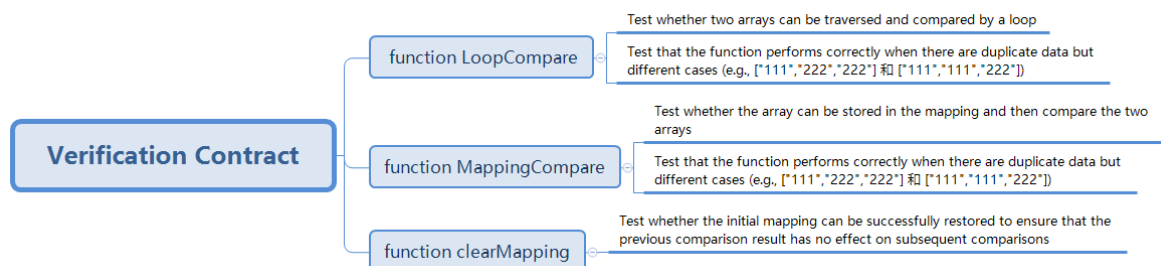


Figure 4. Verification Contract Test Plan

1. function LoopCompare
2. function MappingCompare
3. function clearMapping

2.1.5 Test coverage

2.2 集成测试

本项目中集成测试主要体现在Verification Contract中，其主要测试内容如Figure 5所示。

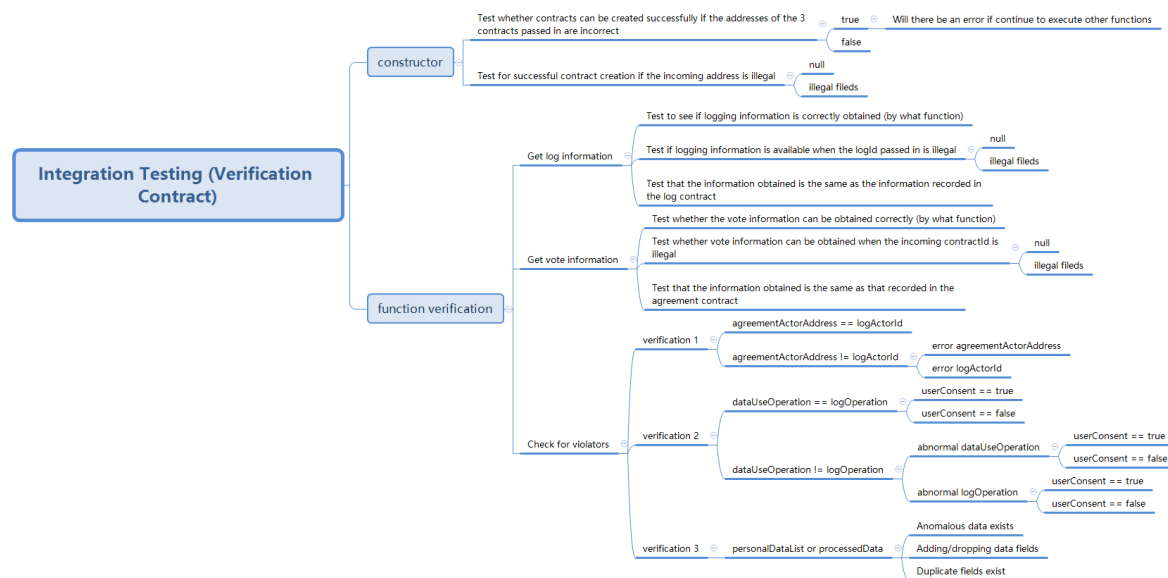


Figure 5. Integration Test Plan

2.2.1 constructor

2.2.2 function verification

1. Get log information
2. Get vote information
3. Check for violators
 - verification 1
 - verification 2
 - verification 3

2.3 系统测试

通过Web3.js脚本，Metamask中Goerli测试网络进行测试

3. 非功能性测试

此项目中非功能性测试将从易用性，可扩展性，可靠性，兼容性以及Gas消耗5方面进行测试并阐述。其中Gas消耗的测试指当Actor从1增长到10时合同的Gas消耗变化。

- 易用性
- 可扩展性
- 可靠性
- Gas消耗

4. 安全测试

此项目中非功能性测试将从边界值测试，重入攻击预防，加密算法的安全性，错误处理，静态代码安全测试5方面进行测试并阐述。其具体测试内容如Figure 6所示。

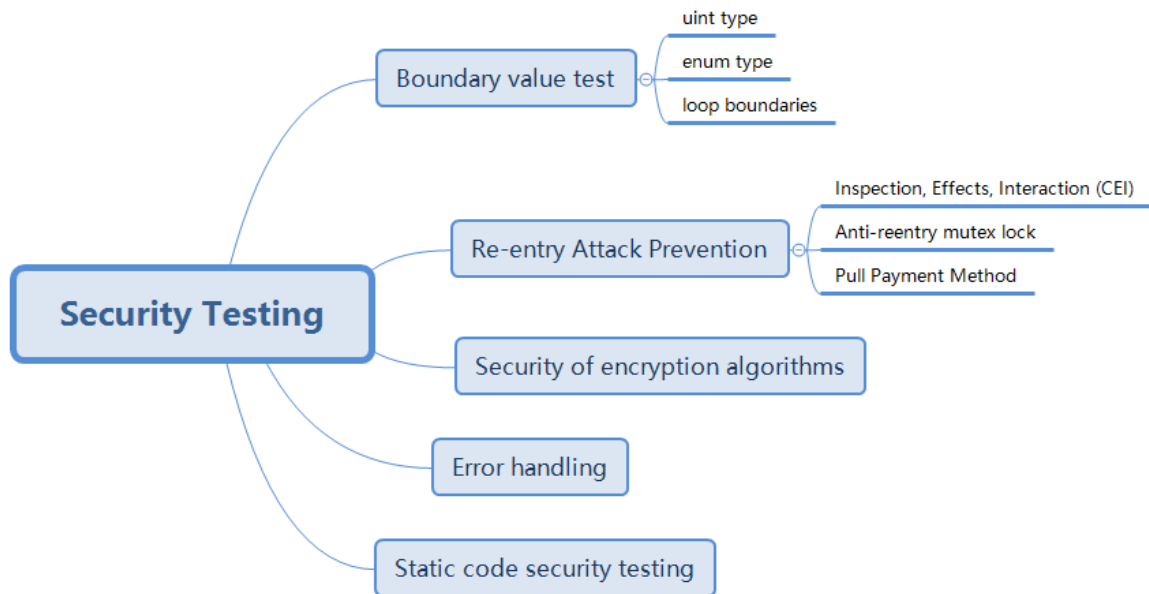


Figure 6. Security Test Plan