TON Smart Contracts Vulnerabilities

Matvey Mishuris

Smirnova Elizaveta

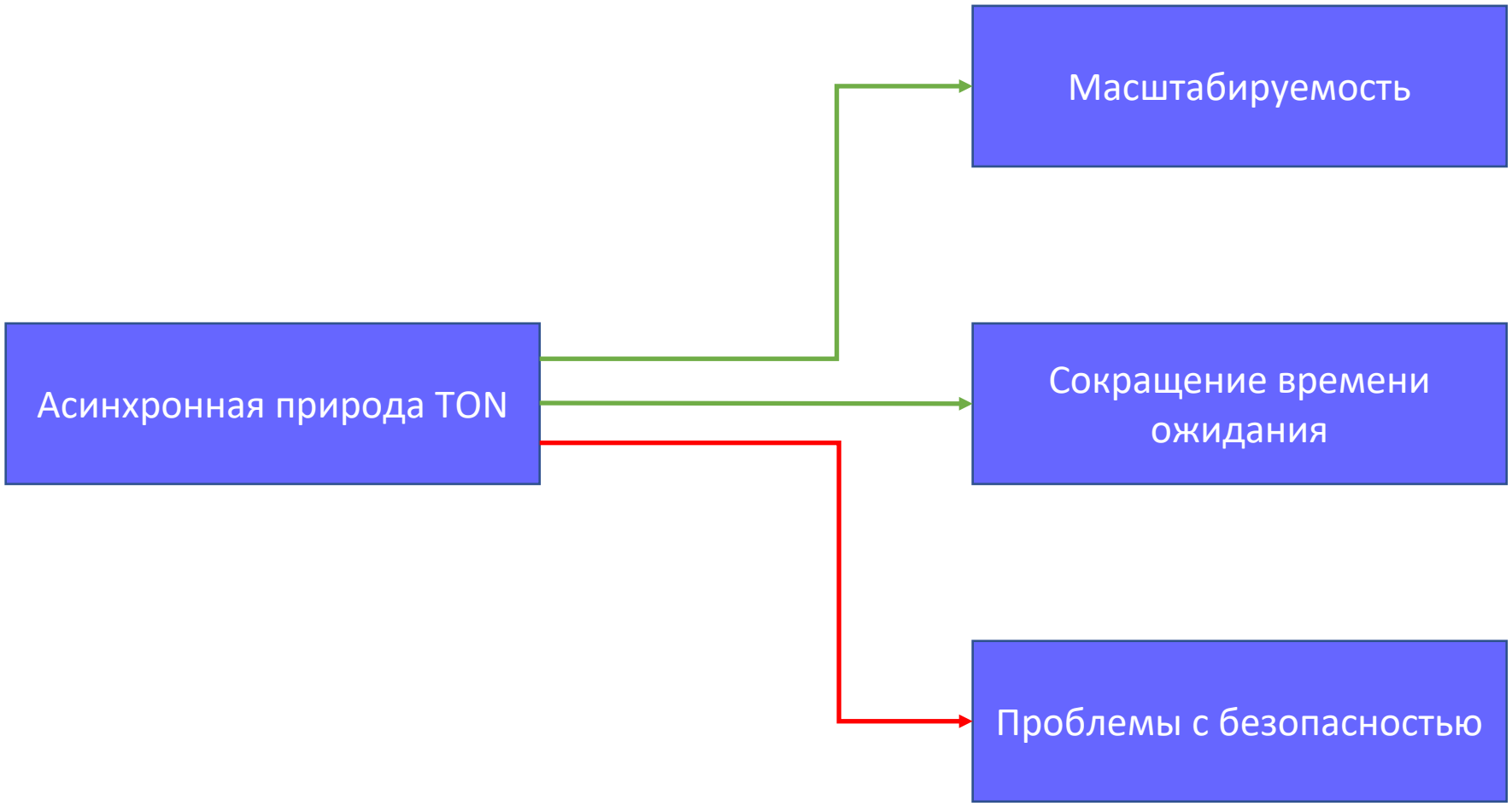
Yury Yanovich

Effectiveness Scalability Efficiency

The Open Network

Smart contracts Decentralized Finance Blockchain

TON


Асинхронная природа TON
Масштабируемость
Сокращение времени ожидания
Проблемы с безопасностью

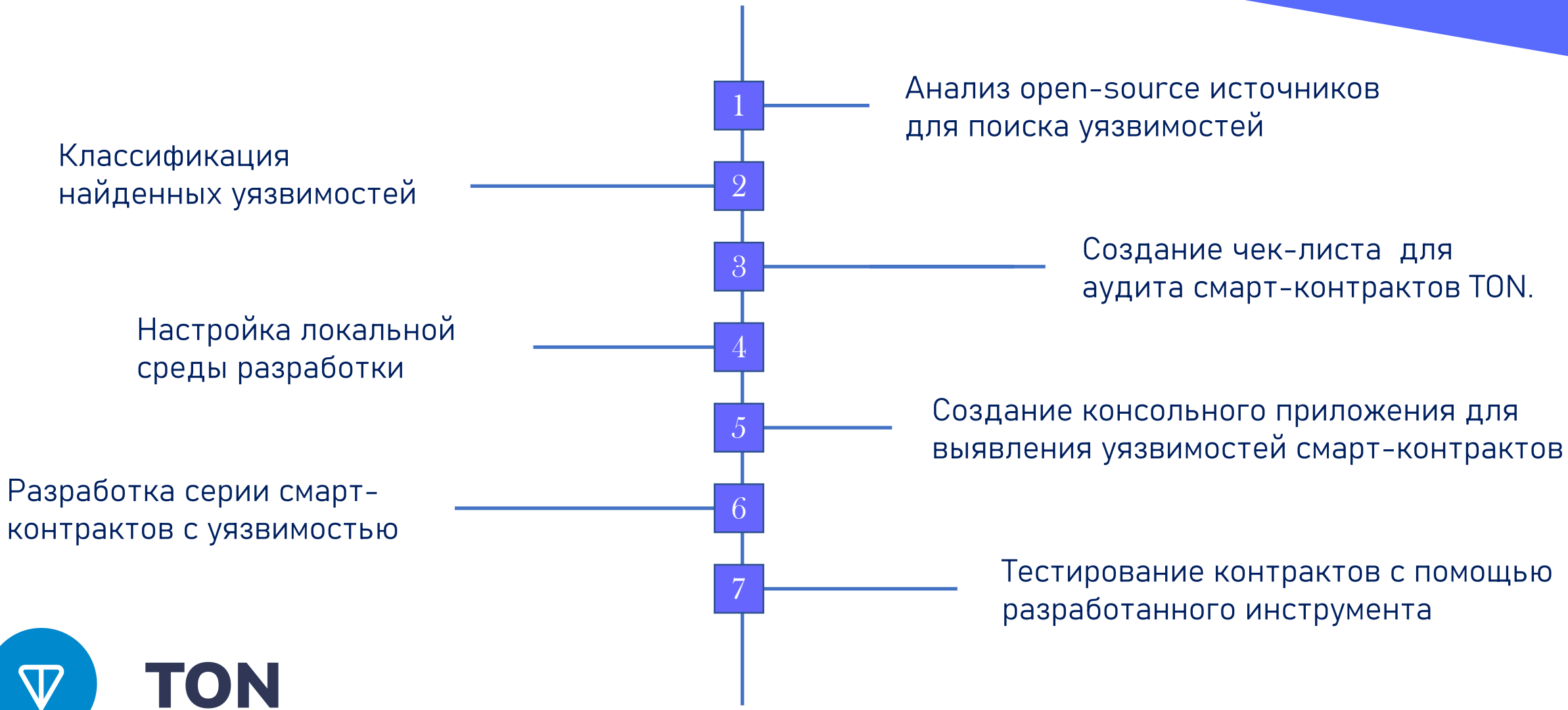
TON

Цель

Разработать инструмент для поиска и устранения уязвимостей в TON smart contracts

TON

План

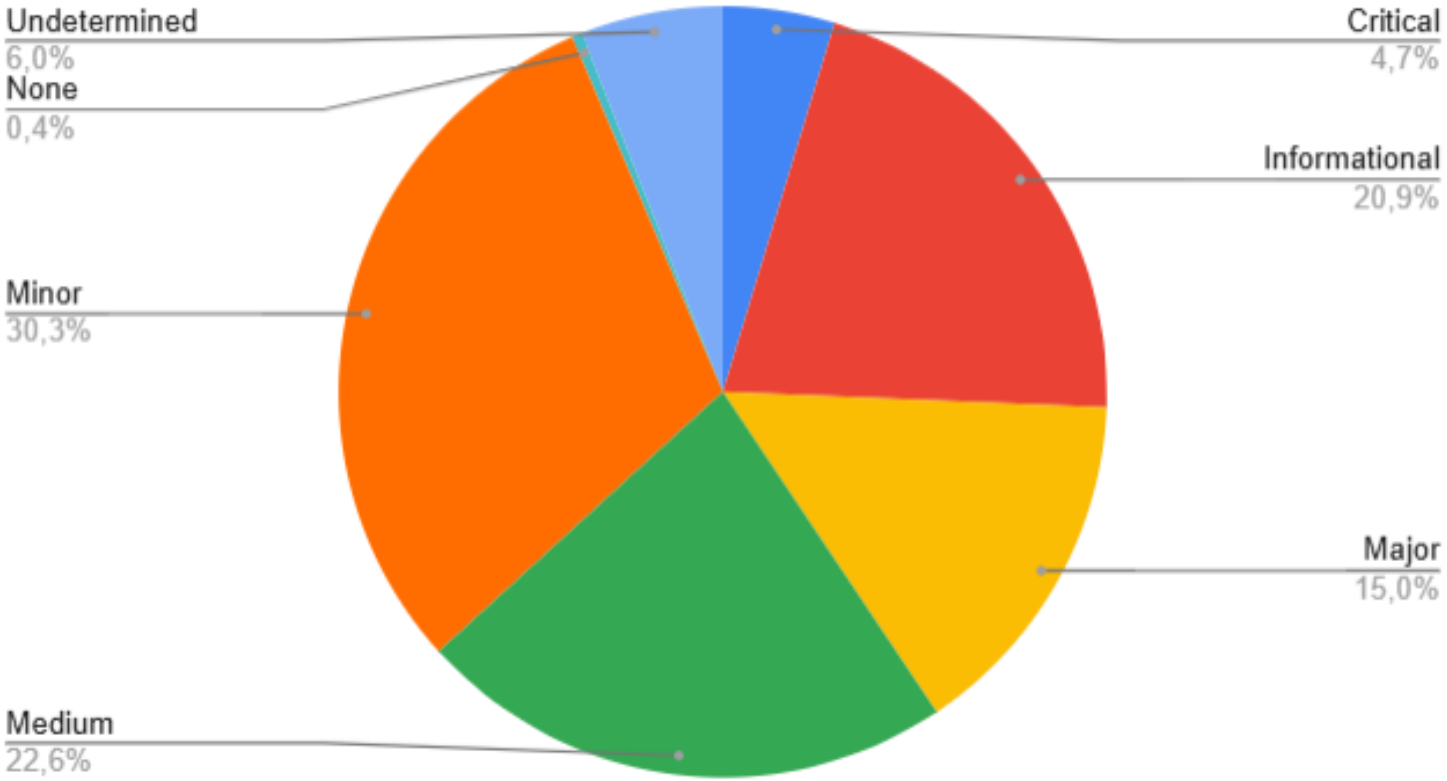

1
Анализ open-source источников для поиска уязвимостей
2
Классификация найденных уязвимостей
3
Создание чек-листа для аудита смарт-контрактов TON.
4
Настройка локальной среды разработки
5
Создание консольного приложения для выявления уязвимостей смарт-контрактов
6
Разработка серии смарт-контрактов с уязвимостью
7
Тестирование контрактов с помощью разработанного инструмента

TON

From Paradigm Shift to Audit Rift: Exploring Vulnerabilities and Audit Tips for TON Smart Contracts

TON

В результате обработки более 30 аудиторских отчётов в общей сложности было зарегистрировано более 230 уязвимостей.



Security Level	Vulnerability Count
Critical	11
Major	35
Medium	53
Low	71
Informational	49
Undetermined	14
Total	233

TON

Структура чек-листа

От общего знакомства с контрактом — к углубленному анализу потенциальных проблем и лучших практик. Основные разделы включают:

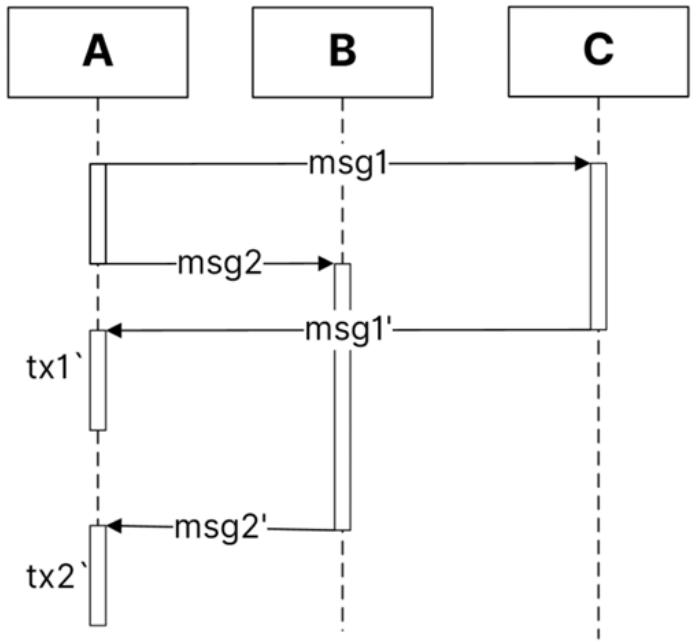
Type	Vulnerability Count
Asynchronous Execution	6
Best Practices	36
Common Errors	91
Contract Design	76
Gas Control	18
NA	1
Possible Errors in FunC	4
Random Number Generation in TON	1
Total	233

- Дизайн контракта (Contract Design)
- Асинхронное исполнение (Asynchronous Execution)
- Типовые уязвимости (Common Vulnerabilities)
- Управление газом (Gas Control)
- Генерацию случайных чисел (Random Number Generation)
- Ошибки, связанные с языком (Language-Specific Error)
- Общие рекомендации (Best Practices)

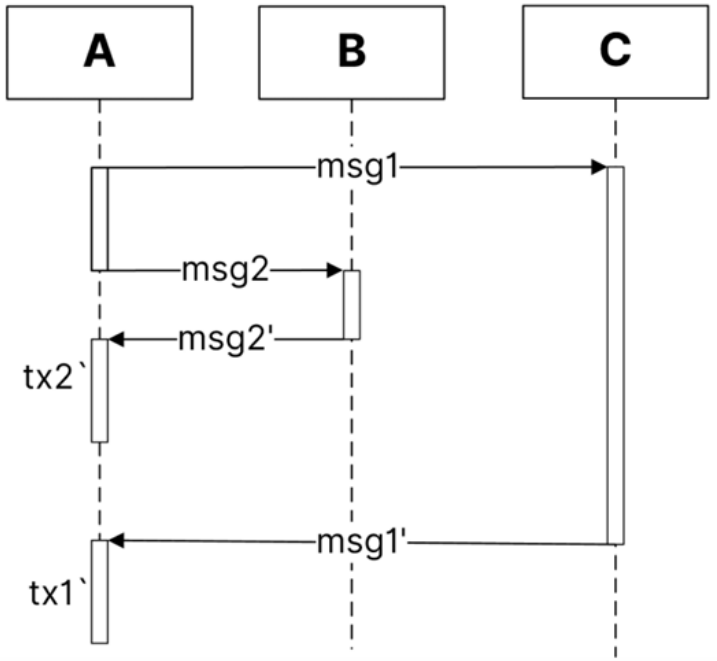
TON

BugMagnifier: TON Transaction Simulator to Reveal Smart Contract Vulnerabilities

TON: No Predefined Order



vs



TON

Эмулятор блокчейна

В ходе исследования были рассмотрены несколько доступных инструментов:

TON Sandbox
TVM-linker
TON Contract Executor

→ TON Sandbox

- Операции выполняются в изолированной среде.
- Локальная копия позволяет задать детерминированное начальное состояние контракта.
- После каждой транзакции сохраняется полное состояние контракта.
- Отсутствие задержек.
- Нет затрат на деплой контракта и gas.
- Локальная копия позволяет безопасно эмулировать спецефичные сценарии.

TON

BugMagnifier разработан для анализа безопасности смарт-контрактов в TON. Основным элементом архитектуры BugMagnifier является класс TONDebugConsole.

TON

```
Command Reference

run next                              - Execute next message from queue
run message <id>                      - Execute specific message by ID
continue                              - Execute all remaining messages
queue list                            - Show message queue
set queue --order reverse/random      - Reorder queue
add messages <path>                   - Add messages from JSON file
delete message <id>                   - Remove message from queue
script load <path>                    - Load custom queue script
script run                            - Execute custom queue script

show state                            - Show current contract state
load state <path>                     - Load state from file
save state <path>                     - Save current state to file
diff <path1> <path2>                  - Compare two state files

show transactions                     - List executed transactions
show message log                      - Show executed messages log

help                                  - Show this help message
exit                                  - Exit the debug console
```

После успешной компиляции и валидации дополнительных параметров запускается интерактивная консоль со следующими ключевыми командами:

TON

```
tondebug> run next

        ╔═══════════════════════════════╗
        ║       Executing Message       ║
        ╚═══════════════════════════════╝

      Message ID:   1
      Name:         ENLIST Alice (1 TON)
      Type:         internal
      Value:        1000000000
      Sender:       1


        ╔═══════════════════════════════╗
        ║     Transaction Executed      ║
        ╚═══════════════════════════════╝

      Contract Address: 105235931327373489687787736238263148142578133606140344497867785826263094040558
      Current Balance:  1998647200

      Transaction Details:
        LT: 1000000
        Hash: c810960b6678c4b4f72ac9ef606d42f1ee749d5e3d5eecdd03ed640e8849b7ba
        Status: active
        Out Msgs: 0

      Previous Transaction:
        LT: 0
        Hash: 0

      Transaction fees:
```

(index)	op	valueIn	valueOut	totalFees	inForwardFee	outForwardFee	outActions	computeFee	exitCode	actionCode
0	'0x1'	'1 TON'	'0 TON'	'0.001353 TON'	'0 TON'	'N/A'	0	'0.001353 TON'	0	0

TON

run next

show state

```
Current Contract State

Balance: 1998647200
Status:  active

Code: b5ee9c7241010c0100a4000114ff00f4a413f4bcf2c80b0102012002 0b020148030a0202ce0407020120050 6008f3b68bb7efc0040fc0088b5d27087e38cc0b4c7cc08300063854c1400e80875d270
00250c407c00e4ccbc00f8b6cc780c4cc0b000a70071c1655c22c23c00f7b6cc7816e497c178a0001f3b51340835d27087e4b4c7e49c0078a00201200809000f3434c0cc7e900c200015007232c7c073c5b27b5520
0009a0a75be0030004f23074df22bc
Data: b5ee9c72410101010028000 04b3b9aca008000af32b6d3fcef8c34044128111b179efb915b004ab26f0214deaa977cdf53103076cdba0a

Last Transaction:
  LT:   1000000
  Hash: c810960b6678c4b4f72ac9ef606d42f1ee749d5e3d5eecdd03ed640e8849b7ba
```

TON

Технологический стек
Языки программирования

TypeScript

- Статическая типизация снижает риск ошибок при работе со сложными структурами TON (ячейки, транзакции).
- Упрощает рефакторинг и поддержку кода.
- Позволяет писать кастомные скрипты на TS, которые транспилируются в JS для выполнения.

FunC

- Официальный язык для разработки смарт-контрактов в TON.
- Компилируется в байт-код Fift, который исполняется в TVM (TON Virtual Machine).

TON

Технологический стек Библиотеки

@ton/core

Предоставляет базовые структуры данных TON: Cell, Address
Сериализация/десериализация ячеек (методы toBoc(), fromBoc())

@ton/sandbox

Эмуляция блокчейна TON в изолированной песочнице.

@ton-community/func-js

Компиляция FunC-кода в Boc-файлы (байт-код).
Интеграция с инструментом для автоматической компиляции контрактов при запуске тестов.

@ton/test-utils

Генерация тестовых данных

TON

Результаты

Достигнутые:

- Разработан чек-лист для аудита смарт-контрактов на основе составленной классификации
- Оформленная статья отправлена в научный журнал "BCRA" (Blockchain: Research and Applications)
- Разработан инструмент в виде интерактивной консоли для аудита смарт-контрактов
- Разработана и протестирована при помощи BugMagnifier серия контрактов с Race Condition

Будущие:

- В процессе финальной обработки находится статья по второму блоку работы

Ссылки:

- https://docs.ton.org/v3/documentation/smart-contracts/func/docs/functions
- https://github.com/ton-org/sandbox
- https://test.ton.org/tblkch.pdf
- https://github.com/ton-org/blueprint?tab=readme-ov-file#contract-development
- https://tonbit.xyz/reports/TonUP-Smart-Contract-Final-Audit-Report.pdf
- https://certificate.quantstamp.com/full/ton-locker-contract/6872997f-1110-45cc-b70f-2a4cd639da1f/index.html
- Разбор конкретных кейсов
- Тестирование
- Submit первой статьи

TON