Contest Proposal: Everscalend Smart Contract Specification (Formal verification - Phase 1)

Short Description

The contestants shall provide the Phase 1 of the formal verification (functional specification) of the **Everscalend** smart contracts (the list is provided below), hereinafter referred to as “smart contracts”, where the detailed description of the “Functional specification” is described below. All debot contracts are excluded from the present contest.

Introduction

In the context of building high-assurance software systems, Formal Verification is a logically sound method of proving the correspondence between a provided program code and a system description, hereinafter referred to as The Specification.

The Specification contains the most important information regarding the system, such as: its general purpose, who are the users, usage scenarios, etc. Besides that, it states what kind of risks and what guarantees (if any) a system provides.

In this work, you are expected to conduct a specification on the program system in a form of well structured document. The document has to be written in the native language, maybe with sporadic mathematical notes. In our case, the document has to focus on the risks and properties that compensate for those risks - it is exactly those properties that will be formally (mathematically) proven on the next stage of the quality assurance process.

The document has to be easily comprehensible both by a Business Analyst and by a Formal Verification expert.

Here, Business Analyst is a role referring to someone from the program system development team having a general comprehension of the system business logic. They are able to clarify both general system behavior and its corner cases, if any.

Specification content

The following specification content is expected to be present:

1) High-level system description

- System purpose
- Terms of the system domain
- Kinds of System Users
- Architecture of the System
- Main usage scenarios
- Users capabilities
- Key system algorithms

The description better be accompanied by diagrams, such as:

- diagram of interaction between a user and a system
- diagram of interaction between different system components

2) Risks

Here, you state what main risks should be considered in the context of a system usage. For example, coins loss, ownership loss, etc.

3) System Properties

In this section, you cover the following:

- List of assumptions under which the system is expected to operate
- Threat model
- List of system properties, i.e. statements regarding a system behavior in different usage scenarios
- List of guarantees, i.e. a subset of system properties that mitigate main risks identified in the Section 2.

Assumptions convey the expected system operation conditions.

Threat model states what is expected from a perfect intruder, its capabilities, incentives, etc.

The guarantees are stated with an eye on the assumptions and the threat model.

Properties should be described mostly using terms introduced in Section 1.

The authors are free to add any kind of supplementary details enhancing the reader's understanding even further, such as lower level function descriptions, description of abstractions used, etc. It is strongly advised to add such details only in case it adds value to the document and doesn't clutter the main narrative.

4) Appendix

Description of the most important system APIs. The system description may refer to some of those APIs to tie the explanation with the program code.

Specification Purpose

The main purpose of the specification is to provide the Formal Verification expert with enough insight on the system purpose and its expected behavior in different scenarios, so they could further perform Formal Verification of the program system.

The properties must cover the main user risks. The formulations must use well-defined described terms.

We advise to group properties by some reasonable criteria, for example by their component origin, or by the risk they eliminate.

We advise to accompany properties with mnemonic names for easier navigation and further discussion.

Specification Evaluation

The specification evaluation criteria are:

- Logical coherence of the text
- General conciseness: convey more with less text
- Clarity and completeness of the system description
- How well the risks are identified
- Clarity and completeness of the system properties

Ideally, the specification receives a proof reading from Business Analyst before being published. The document that has been proof-read by the Analyst and received his/hers appreciation is expected to receive a higher score.

Generic rules

The contest should be compliant to the Generic Contest Rules.

Location

The source code is available at https://github.com/SVOIcom/everscalend-contracts at branch *main* with hash code equal to **8d24e268f9c44bd3e896fb6a28bbf8a42c7027a9.**

Contracts

The following contracts are subject to be audited:

- Giver
    - Giver.sol
- Market
    - libraries
        - MarketMath.sol
        - MarketOperations.sol
        - MarketPayloads.sol
    - MarketsAggregator.sol
- ModulesForMarket
    - BorrowModule.sol
    - LiquidationModule.sol
    - RepayModule.sol
    - SupplyModule.sol
    - WithdrawModule.sol
- Oracle
    - Oracle.sol
- TIP3Deployer
    - TIP3Deployer.sol
- UserAccount
    - UserAccount.sol
    - UserAccountManager.sol
- WalletController
    - WalletController.sol
- utils
    - Platform
        - Platform.sol
    - TIP3
        - RootTokenContract.sol
        - TONTokenWallet.sol
    - libraries
        - FloatingPointOperations.sol
        - TvmCellOperations.sol

Contest Terms

Contestants shall submit a document in PDF format that covers:
- All the errors found
- All the warnings found

Errors and warnings should be submitted to the developers as early as possible, during the contest, so that the code can be fixed immediately.

The document should also contain a high-level description of the system, and any information showing that the contest had a good understanding of the infrastructure and of the code.

Contest Dates

10 Feb 2022 - 11 Mar 2022

Proposed Prices

The total contest budget is **300 kTON**, whereby 270kTON are allocated to the contestant awards and 30 kTON are allocated to the jury reward.

The contestant awards are distributed as follows:
- Place 1 - 120 kTON
- Place 2 - 90 kTON
- Place 3 - 60 kTON

The Jury

The Jury shall be formed from acknowledged experts in the fields of security, smart contract audit, and formal verification fields, whereby:

- Jurors whose team(s) intend to provide submissions in this contest shall lose their right to vote in this contest
- Each juror shall vote by rating each submission on a scale of 0 to 10 (0 equalling rejection of the proposal); a juror may abstain from voting if they do not see themselves sufficiently qualified to judge such proposal
- A juror that has voted on a submission shall provide detailed feedback on it

Jury Guidelines

- The main goal of the jury is to check how the provided specification is accurate and full.
- The specification is intended to be understandable for an average IT professional but at the same time it must be evident it’s relatively easy to convert it to the formal one
- All the requirements mentioned above are considered as mandatory otherwise some points have to be taken from the corresponding application.

Jury Rewards

The jury reward is **3 000 EVER** for each voting for each juror.

This amount shall be granted for jurors who have voted **and** provided feedback on all submissions.