OT-RFC 10

Starfleet boarding (staking) smart contract specs

Document editors:

Branimir Rakić

Document authors:

Branimir Rakić, Tomaž Levak, Žiga Drev, Jurij Skornik (Trace Labs)

Versions:

- 2020-02-17 (v1.4 - after Quantstamp Final Security Audit)
    - Updated FR2 according to auditor recommendation
    - Update regarding additional points - QSP-14, QSP-15 and QSP-16
    - Formatting edits
- 2020-02-16 (v1.3)
    - Additional clarification on smart contract time manipulation related issues (QSP-12)
    - Implemented minor updates to the text according to inconsistencies suggested by bounty hunters
- 2020-02-13 (v1.2 - after Quantstamp Security Audit)
    - Additional clarification on the period (state) transition parameters in the Functional requirements overview table, as per QSP-1
    - Added section on "Post security audit remarks" addressing audit report unclarities
    - Updated functional requirements to reflect the latest decision by the Trace Alliance Starfleet Task Force on increasing the maximum cap from 50 million to 100 million TRAC tokens (FR10)
- 2020-12-16 (v1.1)
- 2020-12-01 (v1.0)
- 2020-11-11 (v0.1)

Table of contents

Abstract

The purpose of this document is to present the detailed technical specifications for the Starfleet boarding smart contracts, crafted by the OriginTrail core developers in order to enable the OriginTrail community to successfully board the newly launched Starfleet blockchain when ready.

Details of the Starfleet initiative have been described in previous OT-RFCs and this specific document focuses on the staking smart contract implementation. It is used for development and security auditing purposes.

Boarding mechanism explanation

In order to provide the necessary consensus layer functionalities for functioning of the ODN, the Starfleet blockchain needs to be provided with TRAC liquidity. Therefore to have the system operational at Starfleet genesis, the initial boarding process to Starfleet blockchain will take place in the form of a staking procedure, followed by the initial StarTRAC generation on Starfleet blockchain (at 1:1 parity).

The procedure consists of the following phases:

1. **Preparation period:** Participant registration and smart contract security auditing.
2. **Boarding period:** Staking TRAC to launch OriginTrail Knowledge Economy

3. **Starfleet launch window:** StarTRAC launched on Starfleet (at parity with staked TRAC on Ethereum mainnet).
4. **Bridge launch window:** With an active bridge, StarTRAC can be exchanged for TRAC and vice versa. Movements of TRAC between chains are enabled
5. **Fallback window.** In case of unsuccessful launch of the bridge, the token holders will be enabled to withdraw TRAC in the amount of their current StarTRAC holding on Starfleet blockchain at a designated Starfleet snapshot block, decided by the Trace Alliance Starfleet Task Force.

Smart contract requirements

The requirements in this document follow the IETF RFC 2119 recommendations for keyword requirement levels.

Functional requirements

- **FR1** Token holders SHALL be able to deposit TRAC during the boarding period, if the maximum staking threshold (MAX_THRESHOLD) has not been reached.
- **FR2** At any point in time, token holders SHALL be able to withdraw their deposited TRAC if the minimum staking threshold (MIN_THRESHOLD_REACHED) has not been reached.
- **FR3** Token holders SHALL NOT be able to withdraw TRAC tokens during the Starfleet launch window and Bridge launch window phases.
- **FR4** The contract manager SHALL be able to transfer TRAC during BRIDGE_PERIOD for bridge deployment.
- **FR5** The contract manager SHALL be able to perform direct accounting of StarTRAC in the smart contract during the fallback period (if the bridge has not been successfully launched during the bridge launch window).
- **FR6** Token holders SHALL be able to claim StarTRAC and withdraw TRAC in the same amount during fallback period (if the bridge has not been successfully launched during the bridge launch window).
- **FR7** The contract manager SHALL be able to change ownership of the contract to another address for bridge deployment or security reasons.
- **FR8** The smart contract SHOULD NOT be able to receive ETH.
- **FR9** The contract manager SHALL be able to set boarding period start time (t_zero) at contract deployment.
- **FR10** The maximum threshold (MAX_THRESHOLD) for staking is 100 million TRAC tokens, while the minimum threshold (MIN_THRESHOLD) is 20 million TRAC tokens .

Functional requirements overview for each period

Period	Activities during period	Period active under following conditions	Functional requirements
Preparation period	Smart contract security auditing, Smart contract deployment, Staking not possible	*t < tZero*	**FR9**
Boarding Period, MIN_THRESHOLD not reached	Staking period, Withdrawal possible	*t >= tZero, t < tZero + BOARDING_PERIOD_LENGTH,* *min_threshold_reached === false*	**FR1, FR2, FR7, FR8**
Boarding Period, MIN_THRESHOLD reached	Staking period, **Boarding successful (minimum reached) but not completed** Withdrawal no longer possible	*t >= tZero, t < tZero + BOARDING_PERIOD_LENGTH,* *min_threshold_reached === true*	**FR1, FR7, FR8**
Unsuccessful boarding, MIN_THRESHOLD not reached within BOARDING_PERIOD_LENGTH	**Boarding unsuccessful**, none of the additional periods are initiated (Starfleet launch window, Bridge Launch window, Fallback period)	*t >= tZero + BOARDING_PERIOD_LENGTH,* *min_threshold_reached === false*	**FR2**
Starfleet Launch Window (Lock period)	**Boarding successful and completed (minimum reached)**, Starfleet blockchain launch, Launch of StarTRAC tokens in the amount of staked TRAC, TRAC tokens on Ethereum locked	*t >= tZero + BOARDING_PERIOD_LENGTH,* *t < tZero + BOARDING_PERIOD_LENGTH + LOCK_PERIOD_LENGTH,* *min_threshold_reached === true*	**FR3, FR7, FR8**
Bridge Launch Window	**Transporter bridge launch**, Two way TRAC <> StarTRAC bridge crossing enabled	*t >= tZero + BOARDING_PERIOD_LENGTH + LOCK_PERIOD_LENGTH,* *t < tZero + BOARDING_PERIOD_LENGTH + LOCK_PERIOD_LENGTH + BRIDGE_PERIOD_LENGTH,* *min_threshold_reached === true*	**FR3, FR4, FR7, FR8**
Fallback Period	**Transporter bridge launch not successful**, StarTRAC accounting performed by contract manager, TRAC withdrawal available for token holders	*t > tZero + BOARDING_PERIOD_LENGTH + LOCK_PERIOD_LENGTH + BRIDGE_PERIOD_LENGTH,* *min_threshold_reached === true*	**FR5, FR6, FR7, FR8**

Non-functional requirements

- **NFR1** The Starfleet staking smart contract SHALL be deployed on Ethereum mainnet and referencing TRAC token at address 0xaA7a9CA87d3694B5755f213B5D04094b8d0F0A6F.
- **NFR2** The Starfleet staking smart contract manager SHALL utilize a secure multisignature wallet for contract management.
- **NFR3** The Starfleet staking smart contract SHALL utilize standardized, proven and security tested Solidity libraries.

Update: Post security audit remarks

During the security auditing process performed by Quantstamp (in early February 2021), two main challenges have been discovered, which are reflected in the **QSP-1** and **QSP-2** issues as indicated in the audit report.

- The issue **QSP-1** reports an unclear definition of technical requirements when it comes to transitions between phases in the Starfleet staking process, which has been addressed in this document (from v1.2) by introducing additional information and clarity in the functional requirement overview table on the previous pages.
- The issue **QSP-2** focuses on the functionality of accounting for the state of StarTRAC (*accountStarTRAC*) on Starfleet blockchain, and the auditor recommending that it be removed. Since this proposition is not according to the functional requirements, we established that the auditor recommendation comes from the unclarity of requirements, leading to the recommendation of using the same function for TRAC and StarTRAC withdrawal in point 1 of recommendation). The intention behind the *accountStarTRAC* function is to enable the contract manager to copy a “snapshot” of the Starfleet blockchain ledger state on Ethereum, should the Fallback period be initiated (and Starfleet bridge launch fail). Since the token amounts for each participant address of staked TRAC in the smart contract at the end of the boarding phase, and acquired StarTRAC on Starfleet blockchain at the end of the Starfleet launch and Bridge phases can (and most likely will) differ due to StarTRAC being actively transacted with on the Starfleet blockchain for a longer period of time, it is necessary to use *accountStarTRAC* to create a snapshot of the Starfleet ledger state

at possible end-of-life of Starfleet blockchain, and *fallbackWithdrawTokens* then to be utilized by token holders to return their TRAC in the amount of StarTRAC that the snapshot associates with their account. These features are however only to be utilized in the case of Starfleet project failure, and the functions are deliberately implemented in such a way that they do not alter the *stake* list state to minimize complexity and potential problems. Recommendations 2 and 4 from QSP-2 have been included in the smart contract implementation as they contribute to the security and efficiency of the smart contract.

- The indicated issue **QSP-12** regarding timestamp manipulation has been acknowledged by the team, yet has not been addressed in the code due to the expected duration of the staking period. Participants in the staking process should be aware of the possibility of potential Ethereum miner timestamp manipulation - if you intend to stake TRAC, please do not wait for the last minutes of the staking period to perform the staking. It is advised to engage in the staking procedure as soon as possible in order not to be affected by this potential situation.
- Issues **QSP-3** to **QSP-13** have all been addressed as recommended in the draft audit report, as well as the best practice recommendations by the Quantstamp team.

There have been 3 additional issues presented by the Quantstamp team in the final version of the audit report from February 17th:

- **QSP-14** concerns potential issues in the usage of the *accountStarTRAC* function by the contract manager, should there be a need for it in the fallback period. As this issue is not related to the participants of the staking contract and cannot affect the staking process, it has been acknowledged by the developers and still left in original form as the intention is to perform all necessary checks and preparations outside of the scope of the smart contract. The development team acknowledges that there could be potential expenditure of ETH should they try to stack too many contributors in the input array.
- **QSP-15** indicated a potential low risk issue with the *transfer* function in the new *withdrawMisplacedEther* function, which is intended to withdraw any wrongfully sent ETH to the smart contract to the contract managers wallet. It has been resolved as per recommendation
- **QSP-16** recommends additional comments and checks to make sure accidental accounting rewrites do not happen, which have been resolved as recommended by the auditor.

The test suite has been increased, reaching a total number of 41 tests.

Conclusion & Next steps

This document presents the specific functional and nonfunctional requirements for the Starfleet boarding smart contracts and is to be used as input for developers, security researchers and the wider technical community to provide feedback and improvement proposals. The corresponding code implementation repository will be hosted on the OriginTrail Github when available.