

OT-RFC-13

TRAC Token deployment on Polkadot

Authors: OriginTrail Core Developers

Contributors: BRX, Milian, Guinesstache, Famous Amos

Version: 1

Date: 2022-10-04

[Introduction](#)

[TRAC as a multi-chain token](#)

[TRAC token deployment on Polkadot](#)

[OriginTrail Parachain TRAC Deployment requirements](#)

[Approach #1: Evolutionary deployment \(from EVM1.0 to EVM2.0\)](#)

[Approach #2: Direct Polkadot-native deployment \(EVM 2.0 only\)](#)

[Bridging TRAC to other ecosystems](#)

[Conclusion](#)

Introduction

The TRAC token is the utility token of the OriginTrail Decentralized Knowledge Graph that is getting deployed on Polkadot through the OriginTrail dedicated Parachain. Becoming a token spanning both Polkadot and Ethereum ecosystems has enormous potential, particularly in the benefits offered by the latest generation of blockchain features in Polkadot, such as native cross-chain interoperability and extensibility beyond the VM level.

The aim of this RFC is to outline the details of different approaches to technical deployment of the TRAC token on the OriginTrail Parachain and collect feedback from the OriginTrail community.

TRAC as a multi-chain token

Since its inception in January 2018, the TRAC token has been deployed as an ERC20 standardized token on the Ethereum mainnet blockchain as a limited supply, non-inflationary utility token fueling the OriginTrail Decentralized Knowledge Graph (DKG). Exactly 500 million TRAC have been created at the token generation event. Since then, the TRAC token has become crucial in enabling a variety of solutions used by notable global organizations as their data footprint constantly increases on the DKG.

As a two layer system, OriginTrail has always been based on two tokens - the TRAC token as the utility token of the OriginTrail DKG (Layer 2), and the gas token of the underlying blockchain used such as ETH, Matic and xDAI (Layer 1).

Earlier this year the OriginTrail community provided strong support for the launch of [OriginTrail specific blockchain on Polkadot - the OriginTrail Parachain](#), with its native OriginTrail Parachain token (OTP) providing blockchain utility on layer 1. Additionally, apart from OTP providing “gas” utility, its tokenomics are designed to directly incentivize network activity in the OriginTrail ecosystem. More information on the interplay between OriginTrail DKG and blockchain layers [can be found in the whitepaper](#), while detailed OTP tokenomics will be the subject of a separate RFC.

TRAC token deployment on Polkadot

With the launch of the OriginTrail Parachain, a highly scalable layer 1 blockchain tailored to the DKG has been successfully added to the OriginTrail ecosystem. This comes with the added benefit of native interoperability with the entire Polkadot ecosystem and any network adopting Polkadot’s communication standards.

Since the TRAC token is an Ethereum ERC20 token with significant adoption, it's important to carefully consider its implementation in the new Polkadot ecosystem. In the previous request for comments [OT-RFC-12 OriginTrail Parachain Bridges](#), the use of a third party bridge was deemed unsafe and a final decision alongside the community was made to teleport TRAC using the pre-audited SFC smart contract.

Polkadot enables advanced blockchain functionalities and customizability, and with the launch of OriginTrail Parachain the OriginTrail ecosystem became a direct beneficiary of features such as:

- **Cross-chain message passing via XCM** enabling Parachains to send arbitrary messages between each other in a trustless manner using the Polkadot Relay chain. This not only enables easy token transfers, but also execution of any arbitrary logic between chains. This opens up room for many integrations across the ecosystem, e.g. a transaction on the OriginTrail Parachain could execute a token swap on the Acala parachain or extend assets available on Moonbeam. More information on XCM [can be found here](#).
- **Secure bridging** between Parachains via the Relay chain, as well as strong, trustless bridges to other ecosystems built as parachains rather than multi-sig driven implementations, which have seen numerous hacks in the previous year. The OT-RFC-12 focuses specifically on this issue and the transitioning TRAC safely to the OriginTrail Parachain.
- **Extensibility through “pallets”**, enabling forkless upgrades directly on chain level (as opposed to being confined to the EVM environment and needing to fork the entire chain), allowing customizations and advanced features tailored to the use case of every parachain. Many pallets already exist for features such as governance, staking, etc.

More information on Polkadot features and implementation details can be found [here](#).

With this RFC we present two distinct approaches of TRAC token implementation on the OriginTrail Parachain (and therefore, Polkadot).

OriginTrail Parachain TRAC Deployment requirements

The end goal of TRAC token deployment on the OriginTrail Parachain is to achieve the following requirements:

1. **R1 - Human centric approach:** appropriate measures and planning has to be undertaken to maximize the ease of use and create a simple user experience of TRAC in order to not hinder the value propositions of the OriginTrail ecosystem.
2. **R2 - Maximized security:** special care must be taken to remove any unnecessary complexities and potential points of failure to ensure the TRAC token economy is robust and remains decentralized. This includes both the technical decisions and implementations, as well as decision making in an open, public discourse (such as through the OriginTrail RFCs).
3. **R3 - Native Polkadot interoperability:** ability to utilize TRAC token across the universe of Polkadot parachains and ecosystem tools (such as wallets), leveraging the innovations and value generated by the Polkadot ecosystem. This requires native Substrate integration through relevant pallets (such as the [Assets pallet](#)).

4. **R4 - Retained Ethereum compatibility:** TRAC on the OriginTrail Parachain should remain compatible with existing Ethereum infrastructure and tooling, retaining its standard ERC20 form. To achieve this in the context of the Substrate runtime, a set of precompile contracts need to be deployed on the OriginTrail Parachain (to reach the EVM2.0 deployment).
5. **R5 - Effective time of deployment:** The time resources required for TRAC implementation to provide utility (in the context of achieving above requirements) should be reasonably low (and thus effective).

In both proposed approaches, each of the above requirements will be graded with **NOT SATISFIED, EVENTUALLY SATISFIED, PARTIALLY SATISFIED** or **FULLY SATISFIED**.

Approach #1: Evolutionary deployment (from EVM1.0 to EVM2.0)

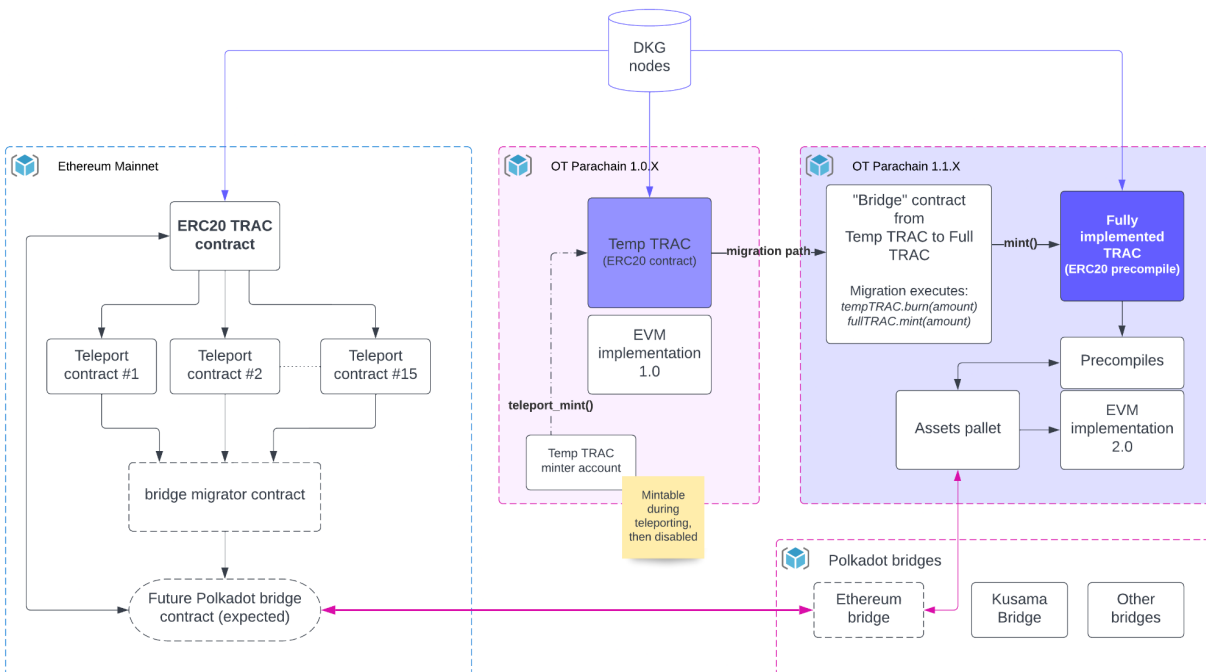
This approach would entail a transition from TRAC initially getting deployed as **ERC20 token** on the OriginTrail Parachain (indicated in scheme below as “Temp TRAC”), to reaching a **Polkadot native TRAC deployment (“Full TRAC”) in the next step**. This initial TRAC token would operate as a standard ERC20 token in the current EVM 1.0 environment on the OriginTrail Parachain (satisfying R4). However in such an environment, the token would not benefit from the additional features available from Polkadot technology (R3), and therefore would require a future token migration to a fully implemented TRAC in the EVM 2.0 environment on the OriginTrail Parachain.

The main advantage of such an approach would be the immediate availability of TRAC on OriginTrail Parachain as no updates to the current Parachain codebase are necessary (OriginTrail Parachain 1.0.x codebase versions). The main disadvantage of the approach is the need to perform a token migration once all features necessary for the Parachain native implementation have been put in place (1.1.x codebase versions). There are two general approaches to such token migration, both introducing a new set of challenges:

1. **User-performed migration (open ended):** token holders would “swap” between the initial and fully fledged TRAC token versions through a “one-way bridge” contract, which would be responsible for minting “full TRAC” tokens. While satisfying criterion R4, this approach most likely leads to an “open ended migration” (both token implementations “living” on the OriginTrail Parachain for an undetermined period of time) as users could decide not to migrate for whatever reason, which is problematic in terms of Polkadot compatibility requirement (R3). Additionally, this approach requires significant adaptations to the implementation of OriginTrail DKG nodes which will have to manage working with both tokens simultaneously. The core developers assess this **as the least favorable option with regards to time resource expenditure (criterion R5)**. Finally, the existence of two implementations for the same token would likely cause user experience hurdles for less tech-savvy users (e.g. asset publishers), creating **problems in terms of requirement R1**

and **likely R2** due to additional complexity.

2. **Developer-performed migration (close ended):** in this option the core developers would retain the capability of “freezing” the “Temp TRAC” token contract implementation for a small amount of time in order to perform a “snapshot migration” from one contract to another (temp TRAC to full TRAC). This option removes some of the downsides of having two simultaneously living TRAC token implementations on the OriginTrail Parachain, however for the cost of necessarily pausing the operations of the entire TRAC economy on the OriginTrail Parachain for the amount of time needed for migration. This particular approach would still require careful planning and execution from the entire ecosystem involved in the TRAC economy on the OriginTrail Parachain. It is estimated that this migration approach would require significant time expenditure from all actors in the community and **is negatively assessed in the context of criterion R5**, based on the experience in previous token staking and teleporting activities. Moreover, the pausing capability introduces a problematic component into the token implementation due to enabling additional powerful token control capability on the smart contract level (potential single point of failure), **problematic in context of requirement R2** for the period of time until the migration to a native Polkadot implementation (full TRAC). Finally, even with developer lead initiative in this approach, it would still require effort and caution from the TRAC community which is **not favorable in the context of requirement R1**.



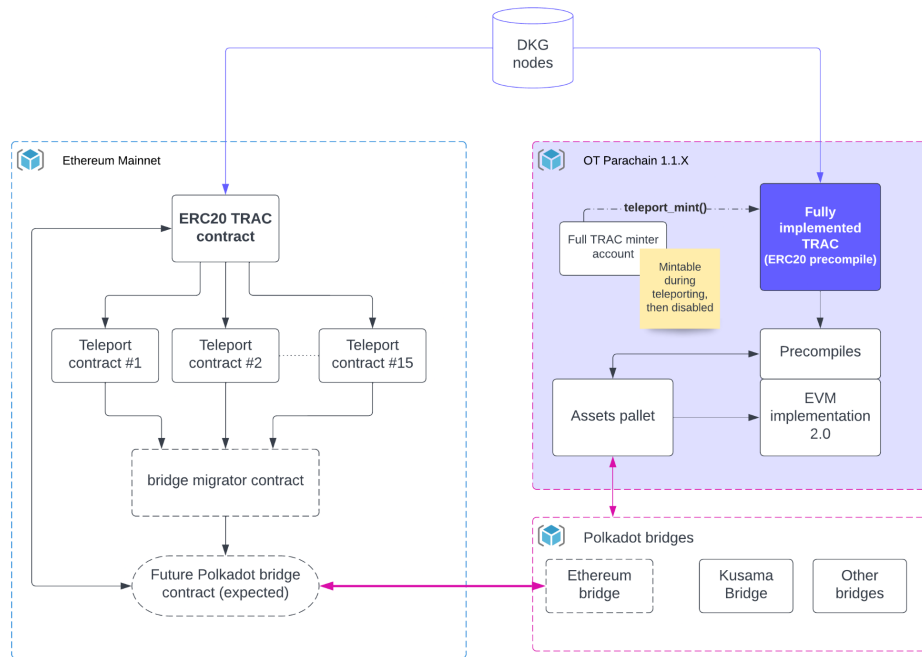
Approach #1: TRAC token migration path from EVM1.0 to EVM2.0 (1a involving the “Bridge” contract, 1b performing a snapshot transition)

Approach #1 in terms of defined criteria then can be assessed in the following way:

1. **R1 - Human centric approach - grade: PARTIALLY SATISFIED** - as both approach #1a and #1b introduce the need to manage two implementations of the same token and require additional activity from all end users of the OriginTrail Parachain, especially in approach #1a which doesn't have a time constraint on the token migration. Given that OriginTrail Parachain is a new environment (with new tools, such as wallets) the community is slowly getting accustomed to, having any additional friction is seen as problematic.
2. **R2 - Maximized security - grade: FULLY SATISFIED** - While significant measures need to be taken to ensure safe transition between two token implementations, the approach does satisfy the security requirements as the process can be entirely based on smart contracts. However, additional steps and complexity inherently do add an additional risk component in the process.
3. **R3 - Native Polkadot interoperability - grade: EVENTUALLY SATISFIED** - with the evolutionary approach TRAC would start as a "non-native" Polkadot asset. The additional features provided by Polkadot would only be available post-migration.
4. **R4 - Retained Ethereum compatibility - grade: FULLY SATISFIED** - in both approach #1a and #1b TRAC starts as an ERC20-compatible token.
5. **R5 - Effective time of deployment - grade: PARTIALLY SATISFIED** - in order to reach the fully deployed state of TRAC on Polkadot, a significant time investment would be required to plan and execute the migration activities in detail, both by the core developers as well as by the OriginTrail community (estimated at several months). The partially satisfied grading of the requirements stems from the fact that it allows immediate deployment of the (non-Polkadot) version of TRAC tokens as no additional updates to the OriginTrail Parachain are needed.

Approach #2: Direct Polkadot-native deployment (EVM 2.0 only)

The second approach would be to avoid the deployment of the TRAC token contract in the EVM1.0 environment (Parachain version 1.0.x) entirely, and **proceed to direct Polkadot native deployment of TRAC tokens**. With such an approach, the TRAC token would from the get-go have all of the benefits offered by Polkadot (as listed above) and would minimize the necessary coordination required by the migration activities detailed in approach #1, all of which would benefit from a **better user experience and less friction**.



Approach #2: TRAC token immediate deployment to EVM2.0 Parachain environment

This approach is favorable with regards to all requirements presented above:

1. **R1 - Human centric approach - grade: FULLY SATISFIED** - since only one TRAC token deployment would ever exist on the OriginTrail Parachain there would be no migrations or potentially difficult operations the TRAC community would need to perform.
2. **R2 - Maximized security - grade: FULLY SATISFIED** - higher security is achieved through minimizing complexity and need of coordination (compared to approach #1).
3. **R3 - Native Polkadot interoperability - grade: FULLY SATISFIED** - advanced Polkadot features will be available immediately from the moment of token deployment, as the token will be deployed as a Polkadot asset.
4. **R4 - Retained Ethereum compatibility - grade: FULLY SATISFIED** - the TRAC token implementation will be performed in such a way that, apart from being a Polkadot asset from the Substrate perspective, it will remain an ERC20 standardized token from the EVM perspective.
5. **R5 - Effective time of deployment - grade: PARTIALLY SATISFIED** - even though approach 2 is not entailing the immediate deployment of TRAC on the OriginTrail Parachain, we estimate that significantly lower amount of resources are required for a fully fledged TRAC implementation in approach #2 (compared to approach #1), from both the development team and the TRAC community, due to no migration activities and code adaptations required from the DKG node side. The estimated deployment time is under 1 month.

Due to the dynamics presented above, approach #2 is the preferred option of the core development team.

Bridging TRAC to other ecosystems

Due to Polkadot's native interoperability features, once implemented, the TRAC token will be available to the entirety of Polkadot and its parachains, as the Relay chain acts as a de facto bridge between Polkadot parachains. This will open up new opportunities for both the token holders as well as builders in the ecosystem to deploy OriginTrail in ways previously not seen.

The implementation of trustless bridges to other blockchain ecosystems is an ongoing effort of multiple teams and has been discussed in the previous OT-RFC-12. The necessary preparation and discussions with ecosystem developers and Parity have been ongoing in anticipation of the bridge deployment between Ethereum and Polkadot, as indicated in the scheme above.

Conclusion

This RFC presents the requirements and approaches to implementing the TRAC token on the OriginTrail Parachain, including the details of their execution according to the requirements. As core developers we invite the wider OriginTrail community to provide feedback and comments in the dedicated discussion on the OriginTrail RFC repository.