Table Of Contents

Summary

TZIP-018 proposes mechanisms for defining upgradeable contracts. More precisely, we describe how to define administrator-forced upgrades and user-defined upgrades.

Motivation

During the past several years, bugs and vulnerabilities in smart contracts caused millions of dollars to get stolen or lost forever. Such cases may even require manual intervention in blockchain operation to recover the funds. As a result, the community starts to acknowledge the need for upgradeable smart-contracts.

Implementing an upgradeable contract is not a trivial task, there are many considerations to be had, such as address immutability and trust between users and administrators. This TZIP addresses those issues.

Abstract

This proposal describes how to implement two common patterns for upgradeable contracts:

1. Contracts capable of being upgraded by an administrator, without the users' consent, whilst retaining the existing contract's address. We call these "administrator-forced upgrades".
2. Contracts capable of being upgraded by an administrator, and letting the user choose if/when they want to transfer "value" (e.g. their tokens) to the new version. We call these "user-defined upgrades".

The design proposed for administrator-forced upgrades supports:

- adding/removing entrypoints;
- changing an entrypoint's code and parameter type;
- changing the storage's content and type;
- address immutability (i.e. the contract's address does not change upon being upgraded).

Thanks to address immutability, this mechanism facilitates pushing upgrades to all users without requiring them to update the contract's address in their wallets. This is convenient for fixing critical bugs when a vulnerability is discovered in the contract's implementation.

However, administrator-forced upgrades are not applicable to some use-cases because they require a certain degree of trust in the person or organization that manages the contract. The users of the contract have to trust that the managing party handles the wallet keys safely and takes all the necessary precautions to prevent their leakage. Incidents like the Bancor hack have been reported, showing practical evidence that upgrade administrator wallets can be compromised. For this reason, we also describe how to implement contracts supporting user-defined upgrades.

The main idea of user-defined upgrades is that the user of the contract is the only party that can choose whether to upgrade and transfer value to the next version of the contract or not. This paradigm offers the additional benefit, in case of project hard fork, of letting the user decide which of the new versions to follow. On the other hand, if the original contract has a bug or vulnerability that threatens the users' funds or renders the contract unusable, user-defined upgrades can be less effective than administrator-forced ones.