Tezos: the OCaml Crypto-Ledger

Benjamin Canou, Grégoire Henry, Pierre Chambart, Fabrice Le Fessant, Cagdas Bozman, Vincent Bernardoff, Guillem Rieu, Mohamed Iguernelala, Alain Mebsout, Arthur Breitman

Tezos: the OCaml Crypto-Ledger

Benjamin Canou, Grégoire Henry,
Pierre Chambart, Fabrice Le Fessant,
Çagdas Bozman, Vincent Bernardoff,
Guillem Rieu, Mohamed Iguernlala,
Alain Mebsout
OCamlPro
tezos@ocamlpro.com

Arthur Breitman
Dynamic Ledger Solutions
arthurb@tezos.com

Abstract

In this talk, we will present the story of the Tezos project. Tezos is a crypto-ledger, i.e. a distributed blockchain with a language to express smart contracts, with two specific characteristics: the first one is its ability for self-amendment, that allows a majority of stake-holders to update the behavior of Tezos without risking a fork; the second one is a design and implementation (in OCaml) that took into account safety and security from the very beginning, and in particular the use of static typing and formal methods when possible. For example, Michelson, the smart contract language of Tezos, is the only such language with a formal semantics, a type system ensuring no runtime errors, and an implementation that uses OCaml GADTs to ensure its correctness.

1 Context

A crypto-ledger is an append-only distributed data structure, with a consensus mechanism to avoid inconsistencies in the view of the most up-to-date version. It is also often called a Blockchain, popularized by the Bitcoin crypto-currency, and the tens of new crypto-currencies appearing every year. Among them, Ethereum was probably the most innovative one (and the main challenger for Bitcoin), with an expressive language to write smart contracts, i.e. programs that are executed when they are added to the blockchain or when they are called by other contracts. However, Ethereum is also famous for the hack of the TheDAO smart contract, on June 17, 2016, when an attacker stole around 50 Million dollars of Ether in a few minutes, using a known vulnerability in the smart contract code. Following this attack, the blockchain was updated to remove the attack, but the update was followed by only 85% of the miners, leading to a fork of the network in two separate networks, Ethereum and Ethereum classic.

2 The Tezos project

The Tezos project was started in 2014, with the publication of a white paper on a self-amending crypto-ledger[1]. The paper showed, well before fork of Ethereum, the importance of governance in a crypto-ledger, and proposed a mechanism to avoid forks of the network. A

first prototype of Tezos was implemented in a few months in OCaml, and has been developed since, to be finally released and run by the end of 2017.

The choice of OCaml was made, from the very beginning, to benefit from the safety and security of a statically-typed language, while still being able to execute code pretty fast. It also provides a well-designed module system, that helps encapsulate the economic protocol (the part of the crypto-ledger that needs to be updated) and abstract it from the other parts of the implementation.

The architecture itself relies on popular and well-tested OCaml libraries, such as Lwt for concurrency and networking, Irmin for a git-based management of the context, and ocplib-json-typed and ocplib-resto for REST RPCs. The client also embeds the OCaml compiler to be able to compile downloaded updates when they have been successfully voted for by the stake-holders.

3 The Michelson Language

Smart contracts are both the strongest features and the weakest point of modern crypto-ledgers. In Tezos, smart contracts are written in Michelson, a language that was designed from the beginning to ease the formal verification of these smart contracts. Contrary to most other smart contract languages, Michelson is a stack-based functional language with a formal semantics, and is statically typed to avoid runtime-errors. It provides advanced data structures as basic types (unbounded integers, lists, maps and sets). The interpreter of Michelson itself is statically-typed using OCaml GADTs.

4 Conclusion

In this talk, we will present how the most advanced features of OCaml were used to implement a modern crypto-ledger with a verification-friendly smart contract language.

References

[1] L. Goodman. Tezos: A self-amending crypto-ledger, 2014. https://tezos.com/static/papers/position_paper.pdf.