An introduction to Tezos

Pietro Abate - Nomadic Labs

12/02/2020

Introduction to Tezos

- Motivations
- Design goals
- Self Amending Cryto Ledger
- Shell Vs Protocol
- General Protocol description and terminology

Quick history of Tezos development

- 2014: Arthur Breitman posts a white paper outlining some flaws in bitcoin and ethereum and proposes an alternative solution.
- 2014 Breitman contacts OcamlPro to develop the first Tezos prototype.
- 2016 Breitman secures the first private investment to develop Tezos.
- 2017 The Tezos foundation is established.
- 2017 Tezos launches the most successful fund-raiser in history: 232 million dollars.
- 2018 Nomadic Labs is created, a France based company uniquely devoted to Tezos development.
- Q2/2018: The Tezos **betanet** goes live.
- Q3/2018: The Tezos **mainnet** goes live.
- May Athens amendment
- July Babylon proposal with Cryptium Labs and Ligo
- Carthage Sept proposal with Cryptium Labs

Tezos / Nomadic Labs

Tezos is a distributed and decentralized ledger. The organization governing the development and innovation of Tezos follows a similar philosophy and it is composed of many entities:

- Tezos Foundation: the umbrella organization devoted to the development of the Tezos ecosystem.
- **Nomadic Labs**: French company working on the Tezos technology
- Tezos South-East Asia
- Other companies developing software for Tezos such as CryptiumLabs
- Research / Academic partners pushing the state of the art in many fields related to the blockchain

Tezos Foundation

The Tezos foundation is devoted to research, development and education.

The three main axes are :

- Research and Education: Research and education that furthers the Tezos protocol and related technologies.
- Community Organizations: Efforts to strengthen and nurture the Tezos community.
- Tools and Applications: Development of tools and applications to support the Tezos ecosystem.
  The Tezos Foundation encourages community members, such as educational and research institutions or developers, in contributing to Tezos.

Key point of Blockchain technology

There are three key aspects to consider:

- Immutability: Single transactions are grouped together in blocks and blocks are arranged together in a chain. Information cannot be changed.
- Decentralization: There is no central authority : Consensus
- Security: The system cannot be subverted by evil actors

Applications using blockchain technology drives the possible trade-off of these three main characteristics.

The building Blocks(!) of a Blockchain

At an abstract level, we can see a blockchain as an immutable database that operates in a decentralized network. It is based on *simple* yet revolutionary ideas.

- Public Key Cryptography / Digital signature / Cryptographic Hash Functions
- A probabilistic solution to the byzantine generals problem for consensus among all nodes
- A p2p / gossip network for low level communication

Blockchains are often called `crypto-ledgers`, that is, an electronic book recording transactions where the identity of users and immutability of the book is cryptographically ensured.

Design Goals

Tezos is a **self-amending crypto-ledger** in the sense that the protocol that guides how transactions are recorded on the chain can be amended using an on-chain governance mechanism.

The main focus leading the Tezos development were :

- Governance : a process to upgrade the protocol over time through on-chain voting.
- Security / Formal verification : a design effort to mathematically prove part of the code, algorithms and smart contract language.

Governance

We define Governance as the process to upgrade the protocol over time through on-chain voting.

The main idea of Tezos is to establish a true digital commonwealth among the Tezos players using a voting mechanism built in the system.

- Reduce Hard Forks and fraction/friction in the community
- Voting allows to amend the mechanism that governs the blockchain
- Voting rewards innovation, and benefits developers

Why we need a governance mechanism today?

- Explicit governance = rule of law
  - Bitcoin blocksize debate (Bitcoin -> Bitcoin Cash -> Bitcoin SV)
  - Different pow algorithm (Bitcoin -> Bitcoin Gold)
  - Ethereum hard fork (Ethereum -> Ethereum Classic)
  - Reduce community friction
- Reward innovation
  - Provides incentive for collaboration over competition
  - Rewards contributors
- Stimulate innovation by faster community adoption
- Who wins ? Schelling point: each person's expectation of what the other expects him to expect to be expected to do.

How governance currently works in Tezos

- An upgrade is proposed
- First round of approval voting to select the upgrade candidates
- Super-majority vote to approve the final upgrade proposal
- The entire voting process is about 3 months long, divided in periods of 3 weeks.

Blockchains: state-fullness in a stateless world.

The main data structure is

- a concurrent data-structure
- and it is represented by a **shared mutable singleton**
- as a linked list of blocks of operations

We use a monadic representation of a mutable state through a series of operations.

Blockchains: state-fullness in a stateless world.

The main data structure is

- a concurrent data-structure
- and it is represented by a **shared mutable singleton**
- as a linked list of blocks of operations

We use a monadic representation of a mutable state through a series of operations.

Ex: Bitcoin

- State = set of unspent outputs (utxo) + total work + block number
- Operations = transactions

Blockchains: state-fullness in a stateless world.

The main data structure is

- a concurrent data-structure
- and it is represented by a **shared mutable singleton**
- as a linked list of blocks of operations

We use a monadic representation of a mutable state through a series of operations.

Ex: Bitcoin

- State = set of unspent outputs (utxo) + total work + block number
- Operations = transactions

Generally speaking we can formalize with a function

$$\text{apply} : S \mapsto O \mapsto S$$

How does it look like?

In general, a blockchain can be a tree.

But how do we pick a canonical leaf?

We need well-ordering over branches.
In practice, for efficiency purposes, we require

$$\text{score} : S \mapsto \mathbb{N}$$

All existing blockchain implementations can be subsumed by a pair

$$\begin{cases} \text{apply} : & S & \mapsto & O & \mapsto & S \\ \text{score} : & S & \mapsto & \mathbb{N} \end{cases}$$

How is score implemented?

Several models are possible

- Bitcoin: total difficulty of the branch
- Centralised: presence of trusted third parties signature
- Proof-of-stake: validate / count signatures

Self-amending blockchain

To make Tezos *self-amending* means to be able to change the function `apply` and `score` as follows

1. Pack `apply` and `score` inside the state
2. Add a new operation to change `(apply, score)`
3. Introspect into protocols

The *protocol* is the heart of a Tezos node

A Tezos node works as follows :

- The network protocol discovers blocks and broadcasts transactions.
- The transaction protocol specifies what makes a transaction valid and creates new blocks.
- The consensus protocol forms consensus around a unique chain.

In Tezos, we call the consensus and transaction protocol as the *economic protocol*.

The Tezos framework is split in two.

The (economic) protocol:

- Validates blocks and operations,
- Can trigger a protocol update,
- Runs exactly the same way on all nodes,
- Expects all needed data to be present when run.

The shell:

- Selects the blocks to validate,
- Downloads and prepares everything for the protocol,
- Includes the protocols,
- Could have multiple implementations behaving differently.

Tezos Blockchain Framework

- Generic (other blockchains platforms can be implemented on our framework)
- Consensus protocol agnostic (possibly extracted code from specification ?)
- Strong emphasis on verified components
- Rigorous software engineering practices
- Written in OCaml
- Already used in production and continually evolving

Tezos Blockchain Framework (cont)

- Modular architecture (nodes, clients, miners/bakers)
- P2p distribution network
- Powerful and extensible RPC services
- Certified cryptography (HACL*)
- Protocol compiler (in the future certified compilation?)

Current Protocol (Athens)

- May 29th first upgrade Athens (a first for the blockchain space)
- reduced the minimum amount of stake needed to participate in consensus from 10k to 8k
- incresed the gas limit
- invoicing example to fund innovation (and pay us a round of beers)

Current Protocol Proposals (Babylon)

- Proposed in July together with Cryptium Labs
- Introduces Emmy+
- Reorganizes accounts (delegation from tz accounts, no more KT1 replaced by verified manager.tz)
- Improvements to smart contracts (entrypoints, multiple bigmaps, gas update)
- Small changes to voting procedure (proposal quorum, quorum caps)

Future Protocols

- Tendermint inspired consensus
  - Tendermint is a recent BFT algorithm
  - With a formal proof
  - Finality
  - More work on incentives
- Privacy
  - We can make a shielded transactions contract
  - We can make a private asset contract
  - Private delegation
  - Secrecy / voting

Tenderbake : https://arxiv.org/abs/2001.11965

Smart Contract Platform

- Popularized by Ethereum as general purpose programs
- The blockchain is the trusted intermediary (think escrow) that replaces many financial, notarial or insurance related functions
- In Tezos are simple programs to automate business logic
- With formal verification in mind

Why Michelson?

Michelson is Tezos smart contract language.

- Generic
- Safe
- Readable
- Easy gas accounting

Inherent tension:

- Generic and easy gas accounting suggest an assembly-like language
- Safe and readable suggest a high-level, functional, language

Michelson as a compromise

A low-level language with high-level primitives.

- Stack-based for easier gas calculation (no variables)
- Statically typed
- Functional
- Lispy

Michelson example

```
storage    (map string nat) ;
parameter  string ;
return     unit ;
code       { AMOUNT @sent; PUSH @required tez "5.00" ;
             COMPARE ; GT ; IF { FAIL } {} ;
             DUP ; DIP { CDR ; DUP } ;
             CAR ; DUP ;
             DIP {
                   GET ; IF_NONE { FAIL } {} ;
                   PUSH nat 1; ADD ; SOME
                 } ;
             UPDATE ; PUSH unit Unit; PAIR }
```

Michelson example (cont)

Initialisation: sets the list of candidates

```
Map (Item "Tacos" 0)
    (Item "Baguette" 0)
    (Item "Kimchi" 0)
```

Tezos vs Ethereum vs BitCoin

The main differences :

- Tezos has an on-chain governance system. Changes of Bitcoin and Ethereum are driven by the lead developers / foundation.
- Tezos is based on Delegated Proof-of-stake while Bitcoin and Ethereum on Proof-of-work (Ethereum has plans to move to a PoS consensus algorithm in the future)
- Tezos has Michelson while Ethereum has the EVM.

Solidity vs Michelson

- High Level (js-like) vs Low Level (with high-level primitives)
- Bytecode vs Readable code
- Virtual Machine (EVM) vs Interpreter

I don't compare Michelson with Bitcoin Scripts as they are two completely different tools.

High level languages

- Michelson is a low level programming language
- It is efficient, and easily verifiable, but can be difficult for a programmer
- There are many ongoing efforts to create high-level languages that compile down to Michelson
- Examples of such languages are {Pascal,CamI,Reason}Ligo or SmartPy

Conclusions and Take-aways

- Tezos is a *self-amending crypto-ledger*
- We develop the Tezos framework that is protocol-agnostic
- The protocol will evolve based on contributions and ideas from the community
- Tezos protocol amendment system can be used to soften problems related to governance
- Tezos can be used as a playground to experiment with consensus algorithms

Question?

Figure: ?

Contact Us

Pietro Abate pietro.abate@nomadic-labs.com