Tezos, the self-amending crypto-ledger

Vincent Botbol – Nomadic Labs – Paris, France

February 11, 2019

General overview of the Tezos Project

Tezos is a distributed and decentralized ledger

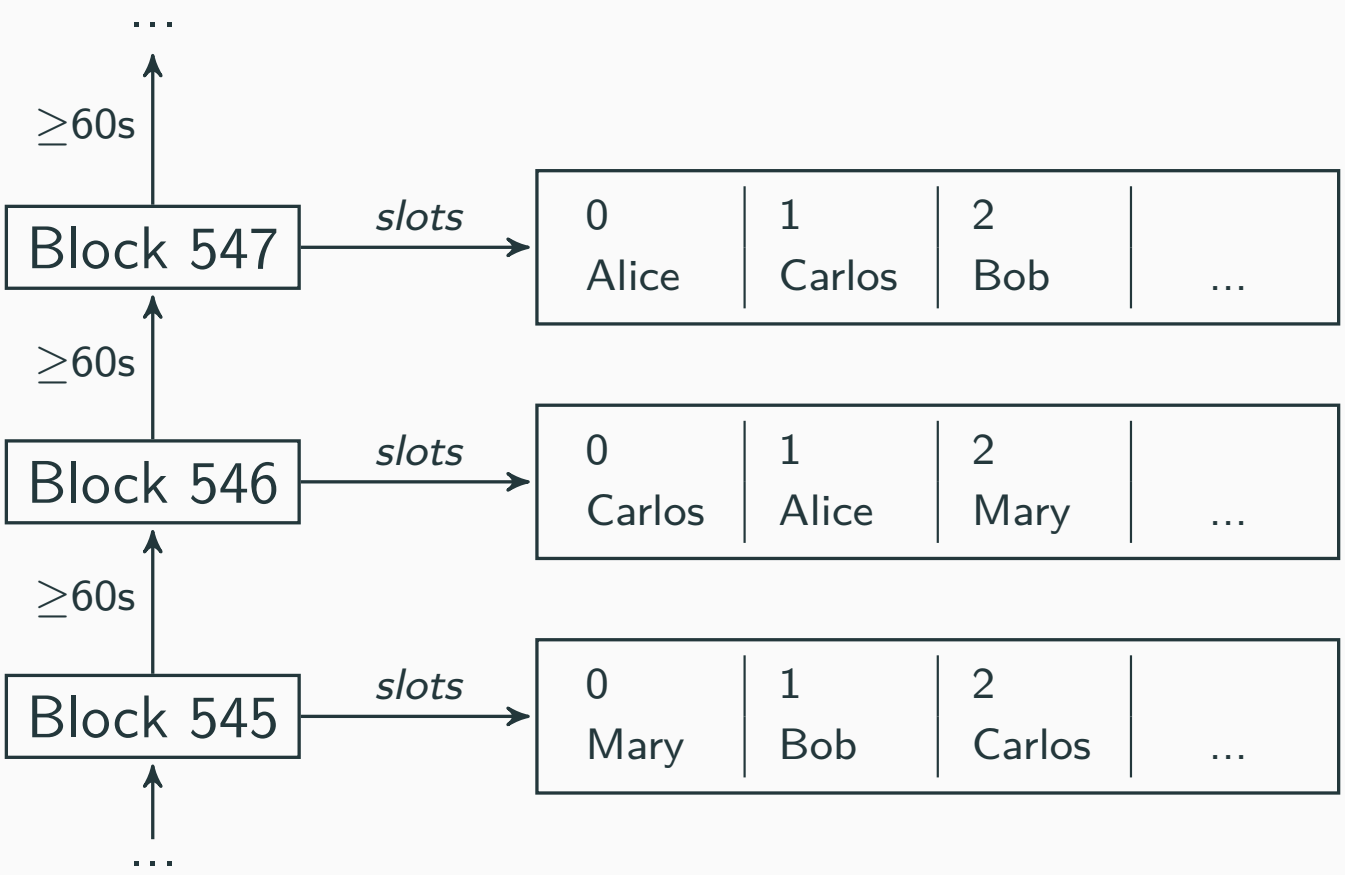
- Q3/2018 Tezos **mainnet** went live,
- Open-Source project (`http://gitlab.com/tezos/tezos`),
- Written in OCaml,
- Proof-of-Stake consensus,
- On-chain Governance mechanism,
- Aims to include state of the art formal verification.

Documentation: `http://tezos.gitlab.io`

Proof-of-Stake Consensus

Current Tezos protocol consensus

To push new block at a certain level, n validators (*bakers*) are randomly selected using a priority list



Specificities

- A baker must have a minimum of 10.000_{tz} (a *roll*) to get slots
- Slot attribution is proportional to the number of rolls
- If a participant does not wish to bake, it is possible to *delegate* its stake

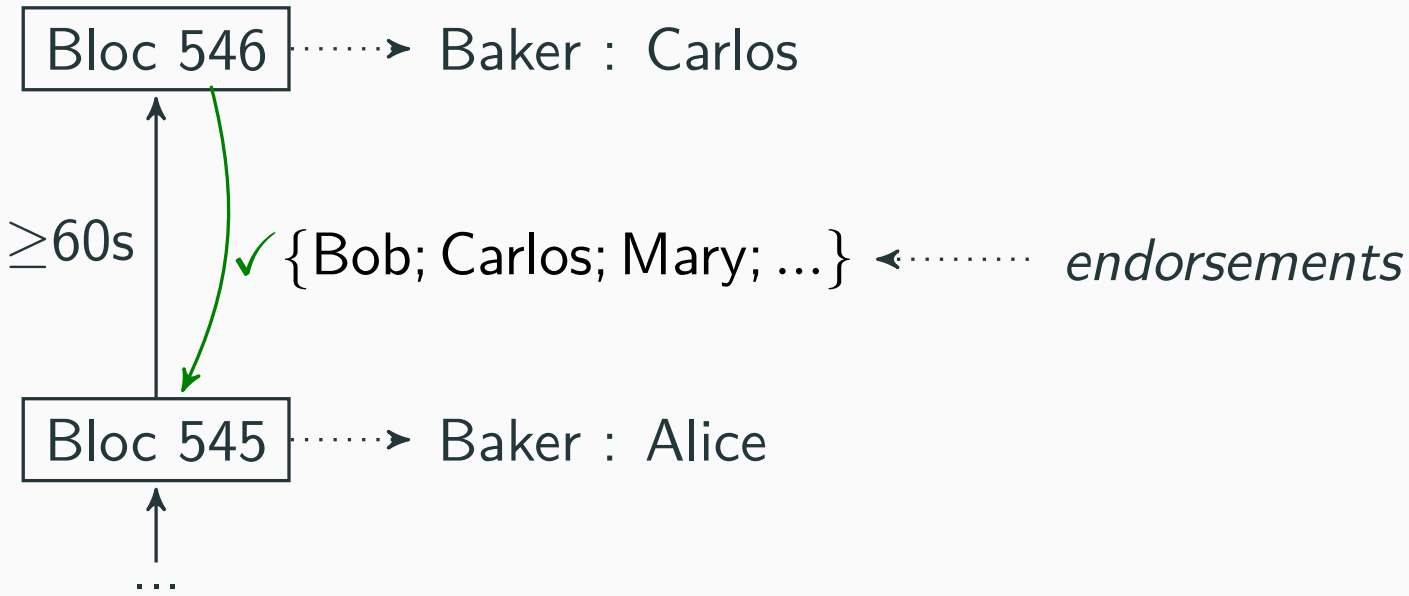
Endorsements

- In order to reach a finality faster, participants are incentivised to endorse blocks
- The highest block resulting score is considered head of the chain where the score is :

$$\texttt{score}(\mathcal{B}_{n+1}) = \texttt{score}(\mathcal{B}_n) + 1 + \texttt{nb\_endorsements}$$

with $\mathcal{B}_n$ a block at level n and `nb_endorsements`, the number of endorsements for $\mathcal{B}_n$ included in $\mathcal{B}_{n+1}$.

Economic incentive & Rewarding (1/2)



Rewards:

- Baking a block: 16_{tz}
- Endorsing a block: $2_{tz} \times 32$ (depending on the slot)

Economic incentive & Rewarding (2/2)

When a baker emits a new block or endorsement, a deposit bond is frozen for ~2 weeks (256_{tz} /64_{tz})

Double-baking

- A baker injects two **different** blocks for a same level

Double-endorsing

- A baker endorses two **different** blocks for a same level

If a baker is caught cheating, the deposit and all pending rewards are forfeited.

On-chain governance

Self-amendment

We define self-amendment as the process to upgrade the protocol over time through on-chain voting:

- Reduce Forks and fraction/friction in the community
- Voting allows to amend the mechanism that governs the blockchain

Exemples of protocol amendments:

- Switch to a different consensus,
- Extend the smart-contract language,
- Modify the rewarding system,
- Anonymous transactions, ...

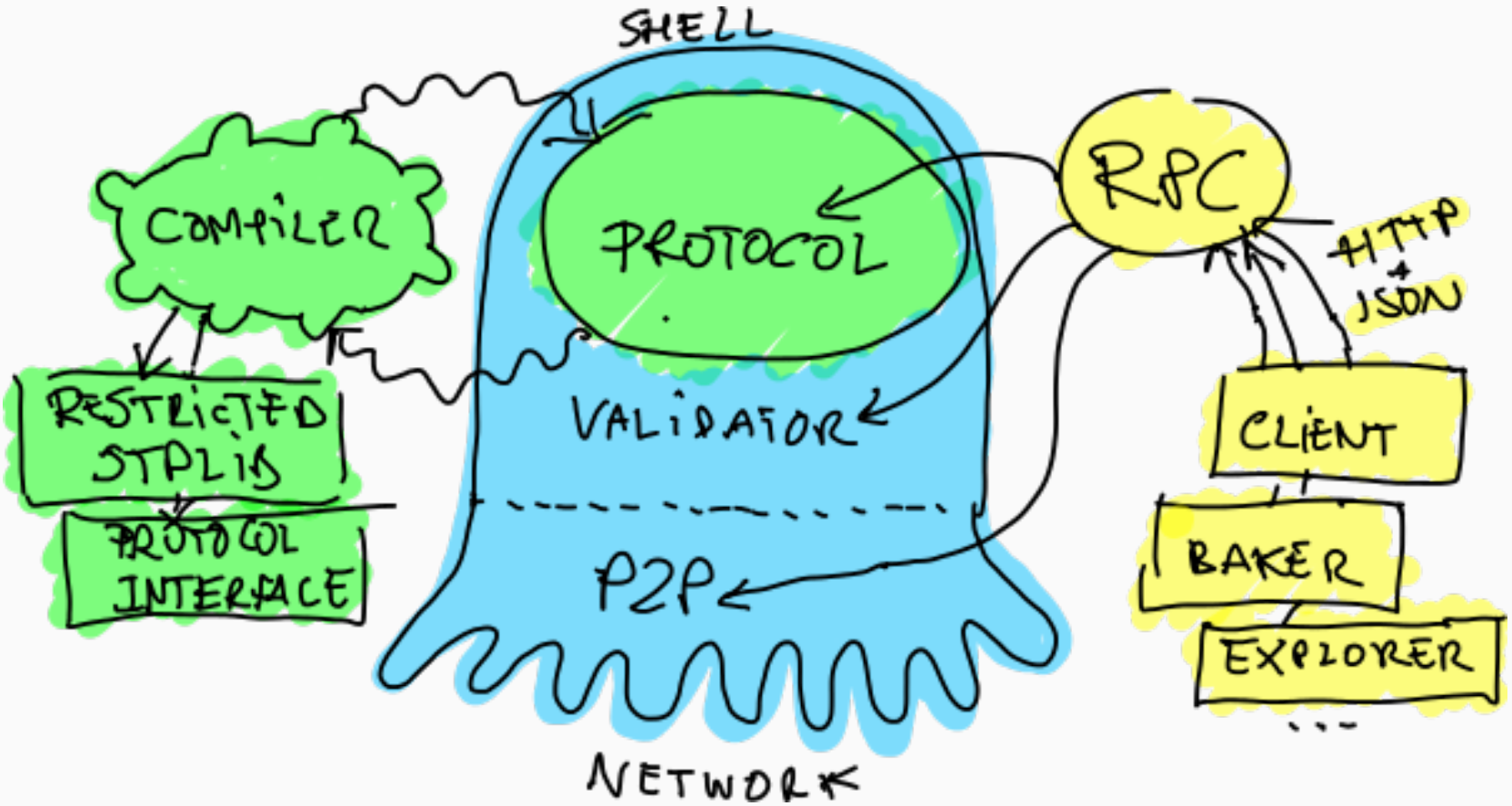
Tezos current voting process

The voting process is split in 4 periods of ~3 weeks each:

1. Participants submit a new protocol proposal
   (i.e. hash of the protocol proposal's source files)
2. A first voting happens for every submitted proposal
3. A side test chain spawns with the elected protocol
4. A final vote occurs to act the upgrade (80% of positive votes)

If the final vote is successful, every participant will automatically switch to the new protocol.

How does it work?



Isolating the protocol

The Tezos node is split in two.

1. The shell:
   - Fetch and propagate blocks & operations,
   - Download and prepare everything for the protocol,
   - Including the protocols,
   - Could have multiple implementations behaving differently.
2. The protocol:
   - The self-amendable part,
   - 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.

Protocol generic interface

We narrow down the protocol interface as much as possible:

- Increases modularity
- Facilitates reasoning about its behavior

The interface is primarily:

- apply: $\mathcal{S} \times \mathcal{B} \to \mathcal{S}$
- score: $\mathcal{S} \to \mathbb{N}$

$\mathcal{S}$ is an immutable (Key $\times$ Value) store and $\mathcal{B}$ is a block

A few other functions are exposed:

- For efficiency (block size, partial score computations, ...),
- For documenting errors,
- Protocol dependent RPCs (Remote Procedure Calls).

Protocol environment

The protocol has restricted access to the standard library:

- No I/Os, no threads,
- No unsafe languages traits, ...

And access to specific libraries:

- Cryptographic libraries,
- Database abstraction,
- High level RPC service definition,
- High level binary and JSON serializers, ...

Formal Verification

Cryptography

Tezos uses HACL* – A cryptographic library formally verified using the F* language

- Verified extraction to `C` and `OCaml`,
- Cryptographic primitives: Ed25519, SHA2 (256,384,512), ...

Smart-contracts

Smart-Contract Language : Michelson

- Stack based for good intuition on gas consumption,
- Statically typechecked,
- No side-effects: can only access its own storage.

One of the main design goals is to simplify the application of formal methods:

- Data-Flow Analysis,
- Model Checking,
- Deductive Verification, ...

Protocol Verification

Critical part of Tezos blockchain

- Establish a rigorous formal specification of the protocol and validate it using a F* or Coq implementation.

Allow the verification of high-level properties – e.g.:

- No unexpected coin creation,
- Chain liveness, ...

Thank you!

Nomadic Labs' first protocols proposals for Tezos

Proposition 1:

- Small tuning and improvements
- Increase smart-contracts gas limit
- Reduce the size of rolls from $10K_{tz}$ to $8K_{tz}$

Proposition 2:

- Small tuning and improvements
- Increase smart-contracts gas limit

Simple propositions – Allow us to test the procedure in a real-life context and polish the tools