Tezos Basics

Öz, B., Hoops, F., & Matthes, F. (2022). "Blockchain-based Systems Engineering". Lecture Slides. TU Munich.

Chair of Software Engineering for Business Information Systems (sebis)
Faculty of Informatics
Technische Universität München
wwwmatthes.in.tum.de

Outline

History and Motivation

Tezos — a self-amending crypto-ledger

White paper

L.M Goodman

September 2, 2014

"Our argument is not flatly circular, but something like it."
— Willard van Orman Quine

Abstract

We present Tezos, a generic and self-amending crypto-ledger. Tezos can instantiate any blockchain based ledger. The operations of a regular blockchain are implemented as a purely functional module abstracted into a shell responsible for network operations. Bitcoin, Ethereum, Cryptonote, etc. can all be represented within Tezos by implementing the proper interface to the network layer.

- The white paper of Tezos was published in 2014.
- The motivation of Arthur Breitman, co-creator of Tezos, was **overcoming the overarching problems of the first generation blockchains** like Bitcoin.

  - Governance ➡ **On-chain governance**
  - Limited set of features ➡ **(Verifiable) smart contract**
  - Evolution management ➡ **Self-amendment process**
  } **Tezos**

- Contrary to most of the blockchains back in the day, Tezos adopted **Proof-of-Stake** (PoS) as its Sybil control mechanism.[1] To bootstrap a PoS chain, an initial distribution of tokens is required.

- Tezos held an open time-limited coin offering (i.e. ICO) to gather funds and issued pre-minted tokens to contributors. The motivation was to **incentivize token holders to participate in validation and governance** due to their financial involvement (in contrast to an airdrop).

- The system was initially designed to meet the needs of an ICO that gathers around 5 million USD. Eventually, the ICO collected more than 230 million USD and this led to a delay in the launch of the network and some legal issues with the investors.[2]

- The ironic part is that the ICO created an off-chain governance problem for a blockchain that promotes its on-chain governance mechanism.

Tezos White paper: https://tezos.com/whitepaper.pdf

[1] Refer back to "Consensus in Bitcoin" slides to understand why we call it a Sybil control mechanism, and not a consensus mechanism.
[2] You can read the following article to learn more: https://www.wired.com/story/tezos-blockchain-love-story-horror-story/

History and Motivation

"Tezos is an open-source platform that addresses key barriers facing blockchain adoption for assets and applications backed by a global community of validators, researchers, and builders. By design, Tezos embraces **long-term upgradability**, **open participation**, **collaboration**, *and* **smart contract safety***."*

https://tezos.com/

Foundation

*"**The Tezos Foundation** is one among many other entities in the Tezos ecosystem and stands as part of the community in **support of the Tezos protocol and ecosystem**. To do so, the Tezos Foundation deploys resources to entities and initiatives that will help to ensure the long-term success of Tezos."*

Tezos Foundation

- A **non-profit organization**, supervised by the Swiss Federal Foundation Supervisory Authority.

- Aims to **promote the Tezos protocol** through **grants** and other capital deployment vehicles.

- Between September 2021 and March 2022, Tezos Foundation has signed 67 grants, for grantees from 33 different countries, up to $243M in approved funds.

- See the [biannual report of March 2022](#) to get more insight.

Official website of the Tezos foundation: https://tezos.foundation/
Read more about the grants here: https://tezos.foundation/grants/

Outline

1. Introduction to Tezos

Sybil Control Mechanisms

■ To prevent Sybil attacks, where an attacker aims to gain the control of the network by generating new entities, Bitcoin employs Proof-of-Work. However, this comes with the drawback of **significant energy consumption**.

Carbon footprint	Electrical energy	Electronic waste
114 Megaton (Mt) CO2	204.50 TWh	32.48 kt
Comparable to the carbon footprint of Czech Republic.	Comparable to the power consumption of Thailand.	Comparable to the small IT equipment waste of the Netherlands.

■ Tezos solves Bitcoin's **intrinsic dilemma** (*network security vs energy consumption*), with its own Sybil control mechanism called **Liquid Proof-of-Stake** (LPoS).

■ LPoS aims to achieve security by true decentralization, thus, it has a **low barrier of entry** for validators.

  ■ Low up-front investment requirement

  ■ Participation through delegation

■ Unlike delegated Proof-of-Stake (dPoS) chains,[1] there is no fixed and limited validator set. Instead, the number of validators is only bounded by the total supply of tez.

[1] Example dPoS chains: EOS, Lisk

Bakers

In LPoS, **bakers** (i.e. validators) are the block proposer nodes.

- *Baking* is the act of signing and publishing blocks to Tezos.

- To become a baker, a node has to hold a minimum of **6,000 tez**.

- A baker is randomly selected to propose the next block **proportionally to their stake.**

  - Bakers' stakes can be **slashed** (i.e., penalized) in case of misbehavior.

- Besides the block proposer, a proportion of bakers are also selected as **validators** at each block.

- Participants of the network who don't want to become a baker can **delegate** their coins to bakers.

  - In return, they (delegators) get shares from bakers' earned revenue.

  - Bakers compete for delegations for increasing their chances to get selected as the next block proposer.

  - When a baker gets slashed, its delegators **do not** get slashed as well.

A tool for an overview of Tezos bakers: https://tzkt.io/bakers/all

Consensus Protocols
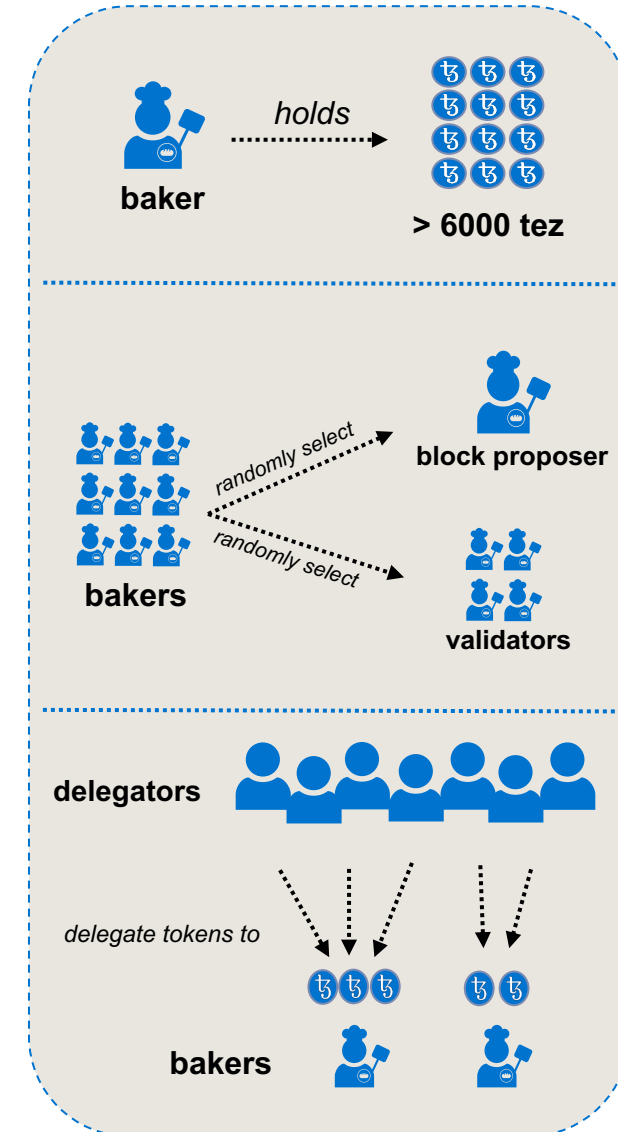
Consensus protocols can often be confused with Sybil control mechanisms (e.g., PoW, LPoS, …).
While Sybil control mechanisms deal with protecting the consensus protocol against Sybil attacks, consensus mechanisms focus on **reaching an agreement regarding the common version of the truth**.

- *How can the nodes in a network determine the right chain?* ➡️ Consensus Protocols

- The two main types of consensus protocols are:

  - **Nakamoto Consensus**: picks the longest, heaviest (i.e., the chain with the most blocks) chain (e.g., Bitcoin – picks the chain with most work/hash)[1]

    - 50% validation of the network is sufficient
    - Probabilistic finality
    - The network never stops operating (**liveliness first**)

  - **Byzantine Fault Tolerant (BFT) Consensus**: picks the block which has more than 2/3[2] of the validators' signatures

    - Deterministic finality
    - The network only operates when there is fully synchrony between nodes (**safety first**)

[1] *This is also shows that Sybil control mechanisms and consensus protocols are mutually dependent.*
2 *Remember that it is impossible for honest nodes to reach a consensus, if more than 1/3 of nodes are malicious. To read more, visit "04 Consensus in Bitcoin" slides.*

Consensus in Tezos

- In Tezos, the consensus protocol is called **Tenderbake**.[1]

- Tenderbake is a **BFT-style** consensus mechanism with **deterministic finality**.

    - Transaction **finality is reached after two blocks** (i.e. two confirmations) which takes around 1 minute currently.

    - This is pretty fast when compared to Bitcoin's transaction finality which requires 6 blocks (around 60 minutes).

- Tenderbake ensures that there is **no parallel block production** that can eventually revert transactions.

    - During an asynchronous period, where bakers cannot sync their actions (e.g., due to a shark biting the fibre optic on bottom of the Atlantic), the network degrades until the issue is fixed (**safety over liveness**).

    - Since all asynchronous periods are finite, the network recovers gracefully once the synchorony between nodes is established again.

    - In Nakamoto-style consensus mechanisms, the network continues to operate during asnychronous periods (**liveliness over safety**), which may then lead to forks or reverted transactions.

[1] Influenced from Tendermint, the first BFT-style algorithm (used by Cosmos)
Read more about Tenderbake here: https://research-development.nomadic-labs.com/a-look-ahead-to-tenderbake.html
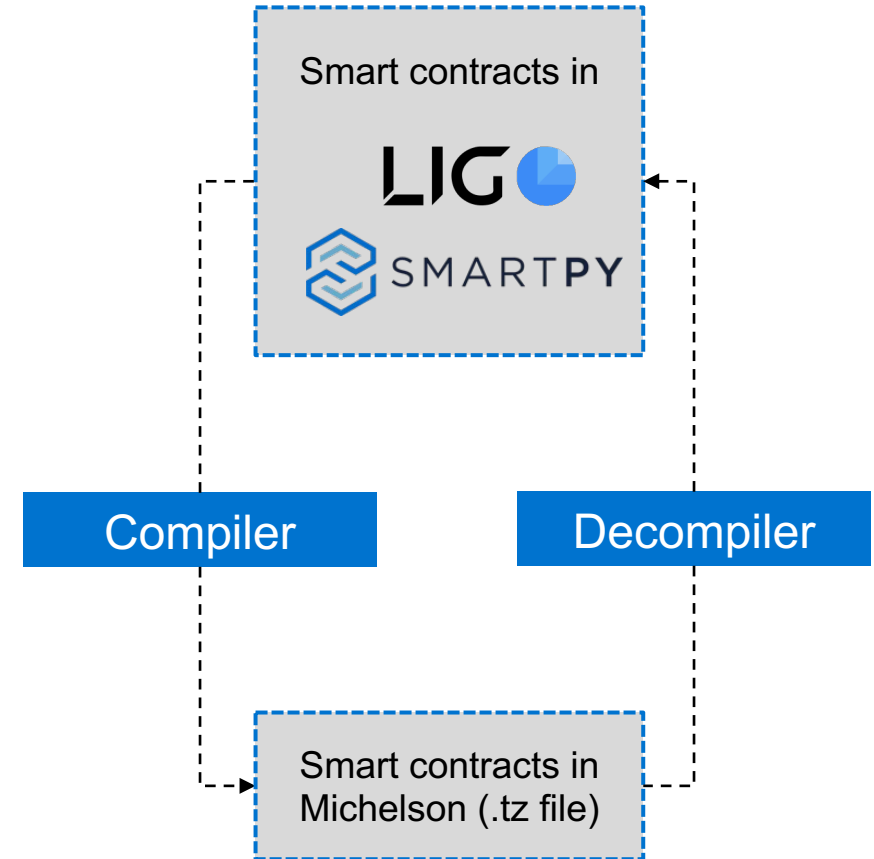
Formally Verifiable Smart Contracts

- Tezos aims to solve **two critical issues** that occur when engaging with smart contracts:

  - *Since smart contracts are programmed by humans, and humans make mistakes, is the smart contract doing what it should be?*

  - *Since the deployed version of a smart contract is not human readable, is it performing the intended use?*

- To address these issues, Tezos has its own functional language to execute formal verifications on smart contracts, **Michelson**, and a certified compiler.

  - To minimize the probability of errors, Michelson utilizes mathematical proofs

  - The certified compiler helps to compare the human-readable version of a smart contract with the deployed version

To read more about Michelson, visit: https://tezos.gitlab.io/active/michelson.html

Formally Verifiable Smart Contracts (cont.)

- Smart contracts that are executed on virtual machines are **attack vectors** for blockchains:

    - Overflow/underflow
    - Re-entrancy ([Ethereum DAO hack](#))
    - Imperfect Features ([Parity](#))
    - Honeypots
    - …

- Michelson VM was custom-made to avoid these bugs through formal verification.

- Some other properties of Michelson[1] are:

    - **Statically typed stack language** (no variables, but high-level primitives such as maps, sets, crypto primitives,…)

    - **Efficient** calculation of gas costs and fast contract execution

    - **Readability** through the expressive representation of a smart contract on the blockchain

Formally Verifiable Smart Contracts (cont.)

- Michelson achieves efficiency and security since it is an assembly-like language. However, to increase readability, it also supports **high-level languages**.

- LIGO and SmartPy are two of the most popular **smart contract development languages** in Tezos, that can be **compiled to Michelson**.

  - Programming in these high-level languages comes more natural to most smart contract programmers

  - For example, LIGO supports different syntaxes that aim to resemble Pascal (PascaLIGO), Ocaml (CameLIGO), and ReasonML (ReasonLIGO)

- The certified compiler can be used by third parties who want to ensure that compiled version of the high-level code matches the low-level code deployed on the blockchain.

Outline

1. Introduction to Tezos

2. Architecture Overview
- Overview
- Sybil Control Mechanisms
- Consensus Protocols
- Formally Verifiable Smart Contracts

3. Governance
- Overview
- Amendments Process

4. Ecosystem

Overview

- Governance has been a known issue for the blockchain ecosystem as previously mentioned in the lecture as well.

- In popular blockchains such as Ethereum, while most of the governance process takes place **off-chain** (e.g. through social discussion),[1] the changes are achieved through forks.

- Forks have serious issues regarding **coordination in a decentralized ecosystem** and **favoring decisions based on merits** (*dictate of the loud minority*).

  - Hard forks can enable **replay attacks** if the new chain is not made incompatible with the legacy chain (e.g., by defining a new genesis block)

- With on-chain governance, changes are decided **based on the votes of stakeholders** (e.g., token holders).

  - Voting takes place on the chain

  - Changes can be automatically activated by all nodes in the network (replacing the old protocol)

  - Proposals can be **evaluated based on merit rather than herd behavior**

  - Decreases the probability of hard forks[2]

[1] There is an off-chain discussion process also in on-chain governance, but the actual approval procedure takes place on the chain.
[2] Read this article by Arthur Breitman to learn more about why there will always be hard forks.

Governance in Tezos

- Tezos employs on-chain governance which is implemented in an **amendment process**.

- Each amendment consists of **five periods** and each period lasts **5 baking cycles**.

    - *1 baking cycle = 4,096 blocks = 4,096 x 30 sec. (approx.) ≈ 2,048 minutes*

    - *1 period = 5 baking cycles = 20,480 blocks ≈ 7 days*

   - *1 amendment cycle = 5 periods = 25 baking cycles ≈ 1 month and 5 days*

Remember

- Voting rights are assigned to bakers depending on their staking balance
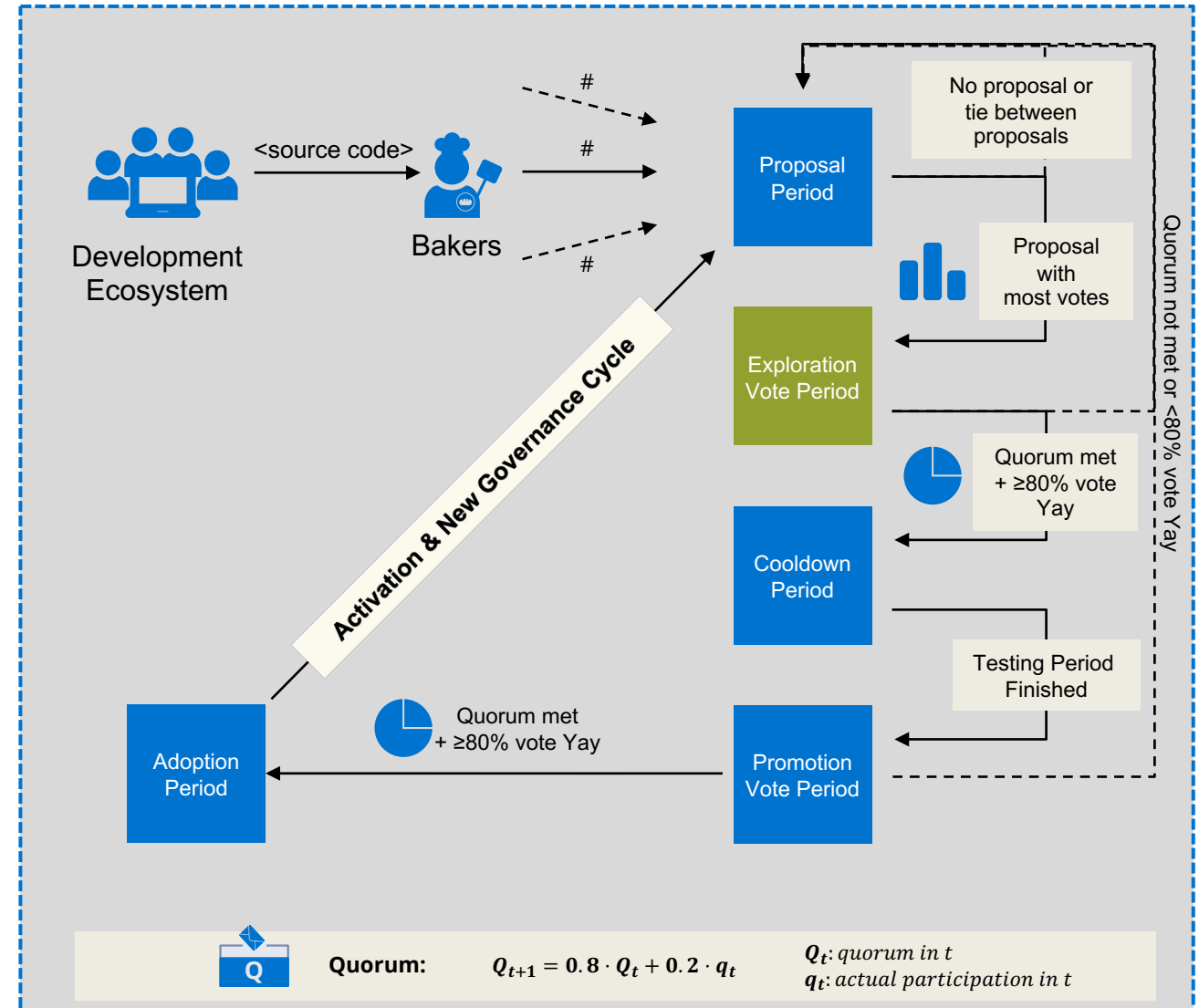
The Amendment Process

Proposal Period

- Bakers submit proposals that contain the hash of the source code of an amendment/protocol.

- Each baker can submit up to 20 proposals.

- Bakers also specify a vote while submitting a proposal (proportionally to their stake).

- The most voted proposal continues to the *Exploration Period*.

- If there are no proposals submitted by the end of the period or if there is a tie, the amendment process reverts to the beginning.
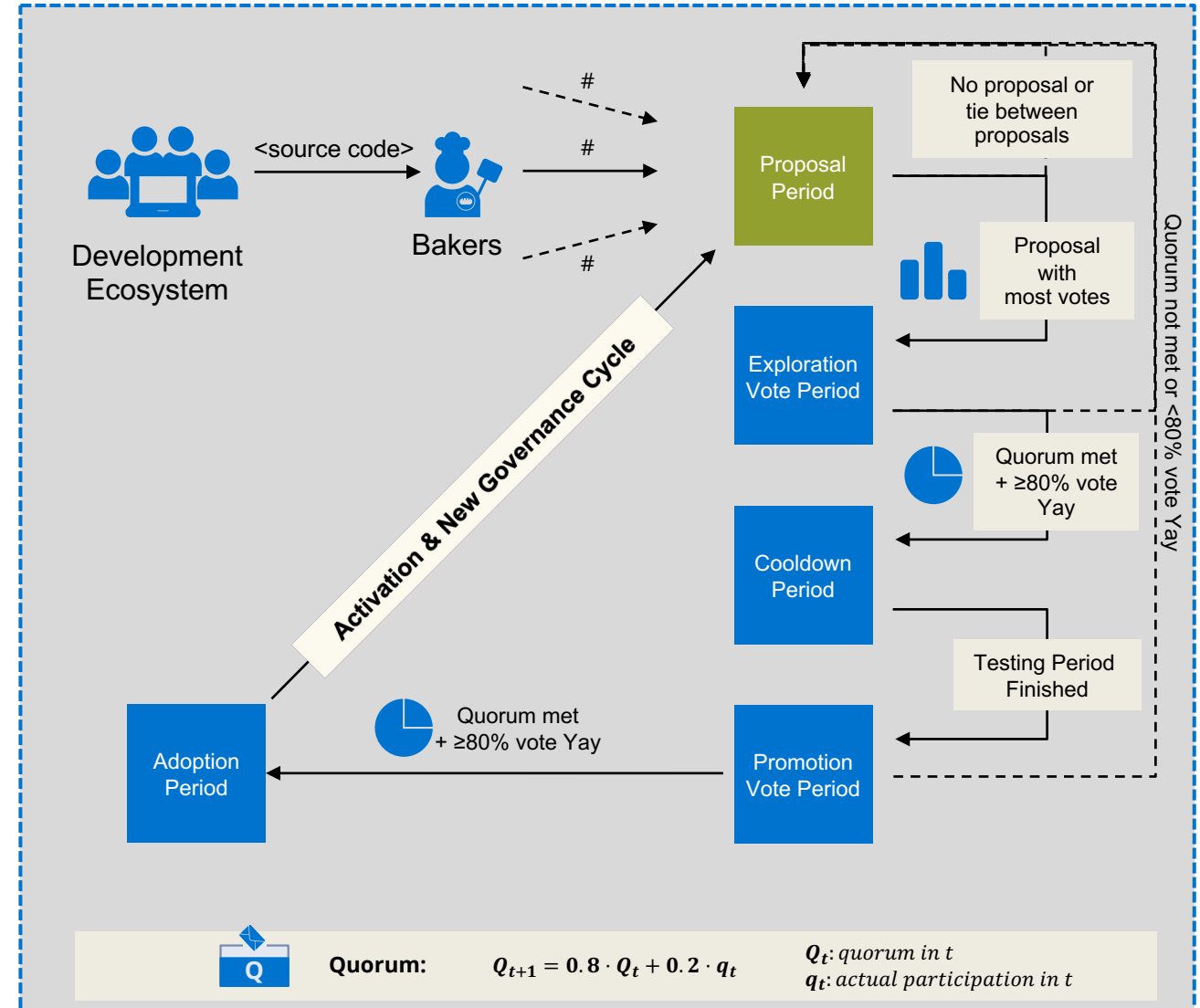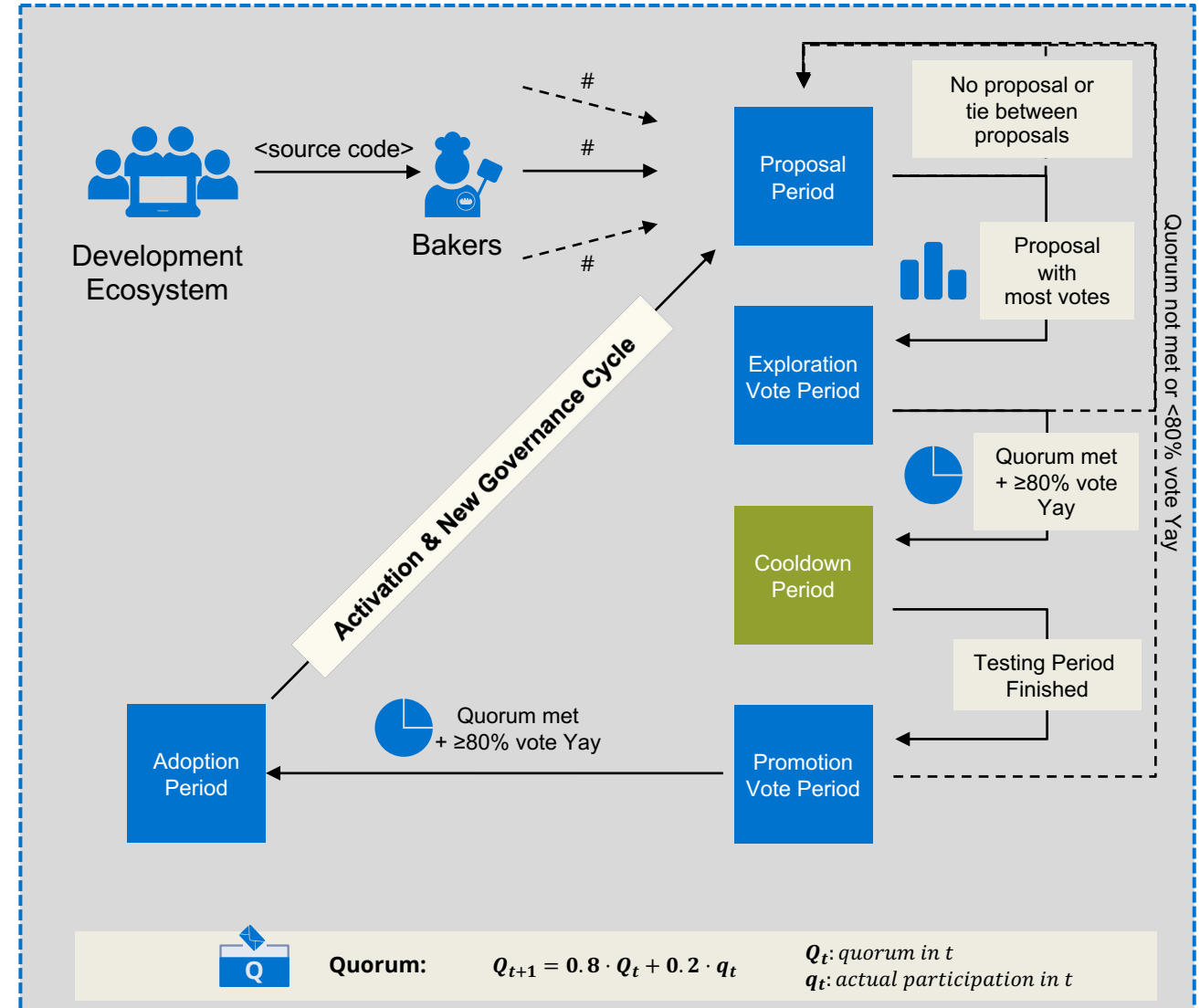
The Amendment Process

Exploration Period

- Bakers vote on the most voted proposal from the *Proposal Period*.

- They can vote either "*Yay*", "*Nay*, or "*Pass*" on a specific proposal.
  - "*Pass*" indicates that the baker has no preference

- If more than 80% of the bakers (supermajority) vote "*Yay*" and a dynamically determined quorum (calculated based on the previous quorum) is met, then the proposal continues to the *Cooldown Period*.

- If the conditions are not met, the amendment process reverts to the beginning.

$$Q_{t+1} = 0.8 \cdot Q_t + 0.2 \cdot q_t$$

Q_t: quorum in t
q_t: actual participation in t

Cooldown Period

- Nothing specific happens on-chain during this period.

- Off-chain, the community analyze and discuss finer points of the proposal.

- Developers perform additional tests.[1]

Quorum:
$$Q_{t+1} = 0.8 \cdot Q_t + 0.2 \cdot q_t$$

Q_t: quorum in t
q_t: actual participation in t

Promotion Period

- After the testing in the *Cooldown Period*, the network decided whether to adopt the amendment.

- Again, bakers submit their votes based on their stakes.

- At the end of the period, if the supermajority vote "*Yay*" and the quorum is met, then the amendment gets activated on the mainnet.

- If the conditions are not met, the amendment process reverts to the beginning.

$$Q_{t+1} = 0.8 \cdot Q_t + 0.2 \cdot q_t$$

Quorum: Q_t: quorum in t
q_t: actual participation in t
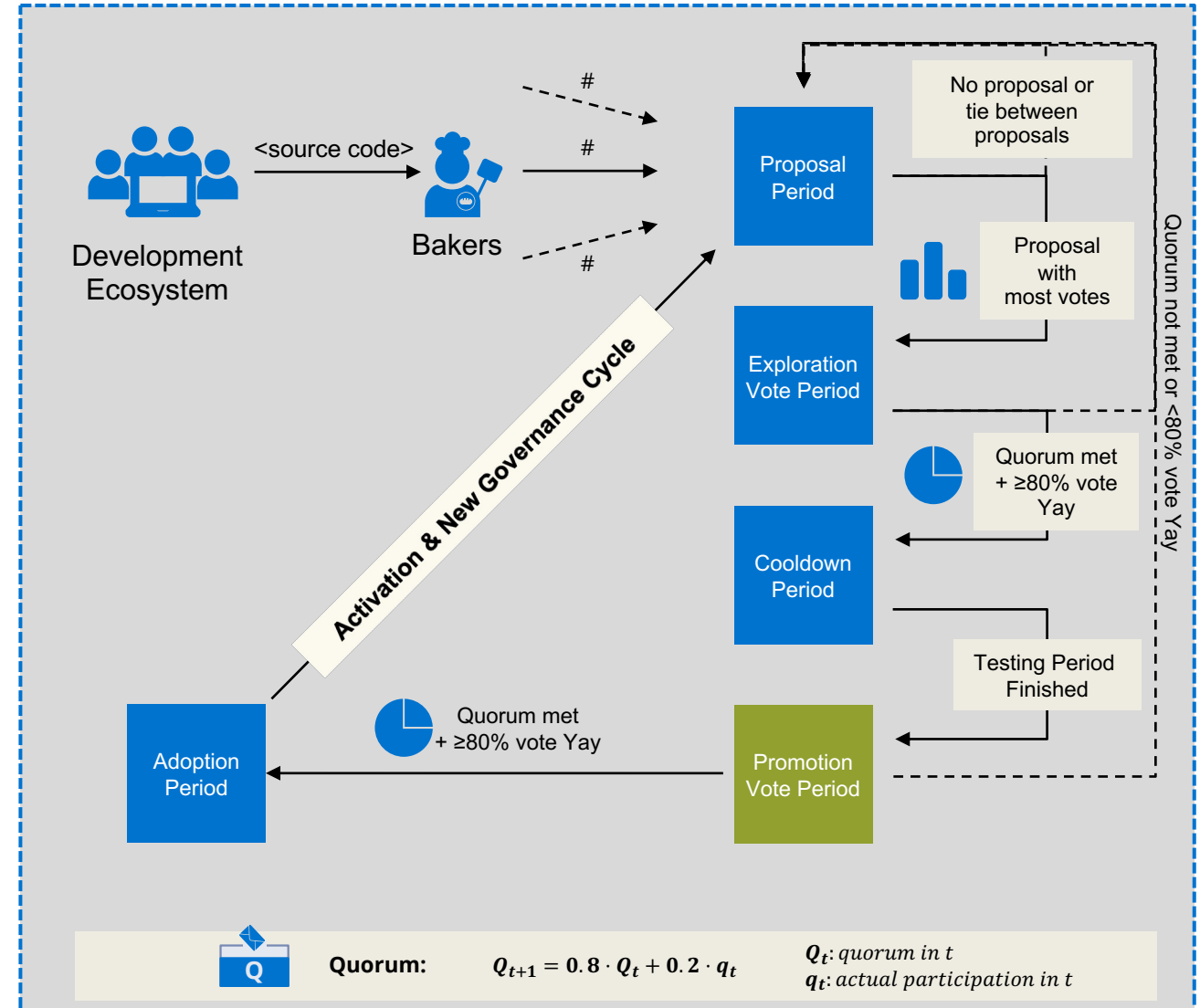
Adoption Period

- Serves as a buffer for the bakers and developers to adopt the new protocol version and do the necessary updates to their code base and infrastructure.

- At the end of the period, the amendment cycle restarts with the next *Proposal Period*.
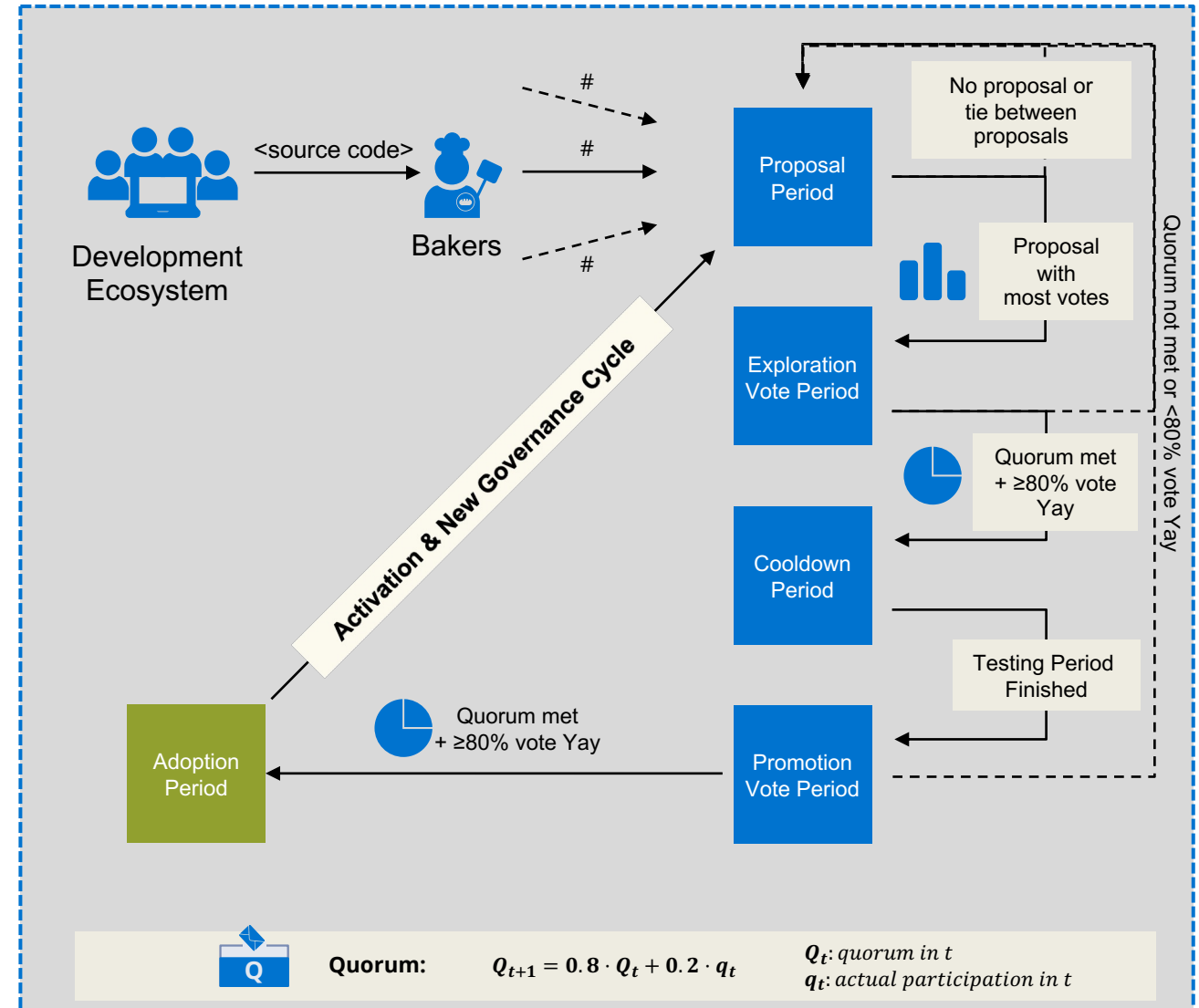
Amendment History

- Up to this day, there have been 9 amendments that successfully went through all of the periods in the amendment process.

- An amendment can include different types of changes (e.g., bug fixes, new features) and updates to the current protocol.

- [Tezos Agora](#) is a tool for monitoring the state of the amendment process and browsing details about all periods that occurred so far (e.g., proposals, descriptions, voting results).

- Tezos Agora also serves as a forum for the community to discuss current and future proposals (off-chain).

Screenshot taken from: https://www.tezosagora.org/period/64
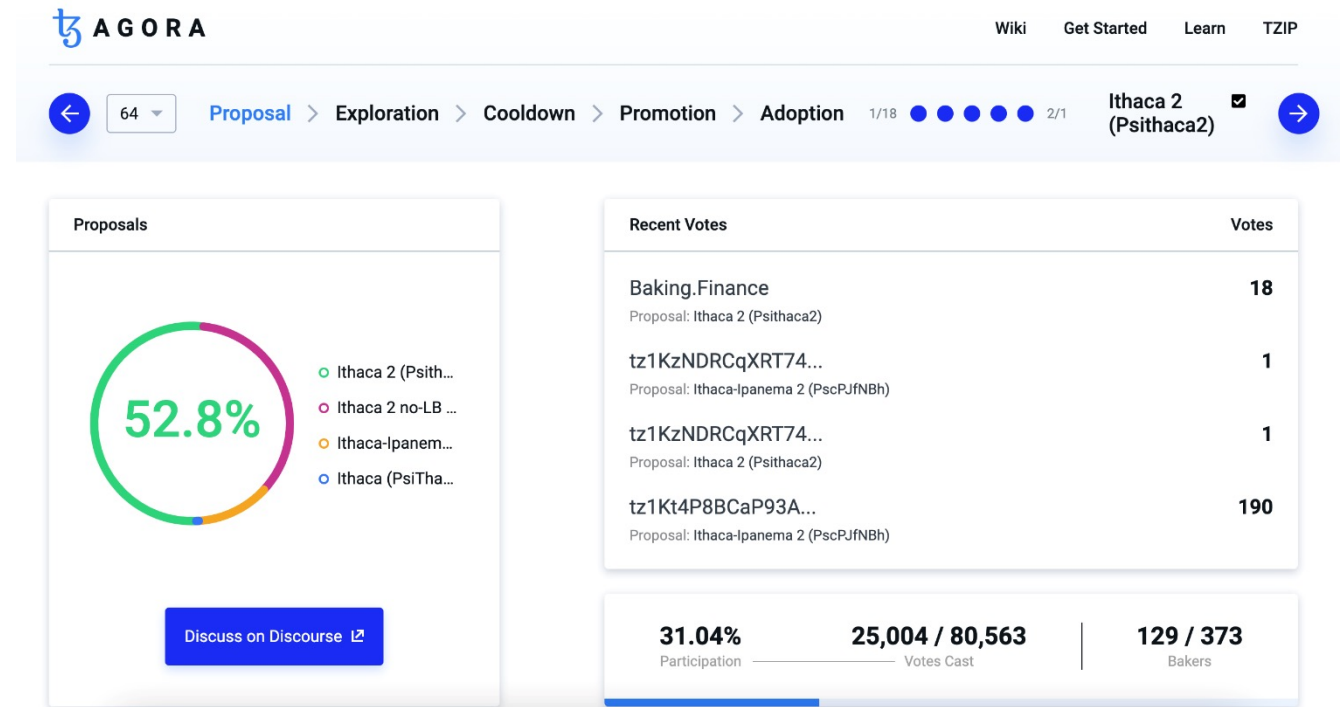
- **Athens – 05/19**
  - First successful amendment of the on-chain governance mechanism
  - Increase in gas limit per block and reduce in roll size to 8,000 tez
- **Babylon – 10/19**
  - Upgrade to consensus (*Emmy*+) and Michelson upgrades
- **Carthage – 03/20**
  - Increase in gas limit per block and improvement in the formula for baking and endorsement rewards
- **Delphi – 11/20**
  - Increase in the amount of computation per unit of gas, reduction in storage cost, and code refactoring
- **Edo – 02/21**
  - Privacy preserving transactions through Sapling
  - Reduction of periods to 5 cycles (20,480 blocks – 10 weeks) and addition of the *Adoption Period*
- **Florence – 05/21**
  - Increase in max operation size (16kB to 32 kB) and elimination of test chain in the third period (*Cooldown*)
- **Granada – 08/21**
  - Upgrade to consensus (*Emmy**) which reduces block time from 60 sec to 30 sec,
  - Liquidity baking (piggybacking of the liquidity and availability of Bitcoin – *tzBTC*)[1]

[1] Read more about liquidity baking here
More info about amendments and overall Tezos protocol: http://doc.tzalpha.net/index.html

Amendment History (cont.)

- **Hangzhou – 12/21**
  - RPC changes
  - Michelson on-chain views to read storage of other smart contracts
  - Introduction of Timelock opcodes to prevent bakers seeing the content of transactions and ordering them in a way that will yield a gain for bakers (also called *block proposer extractable value*)[1]
- **Ithaca – 04/22**
  - Introduction of Tenderbake (the current consensus mechanism), replacing Emmy* in order to provide deterministic finality
  - Reduction in the required tez amount for becoming a baker (from 8,000 tez to 6,000 tez)
  - Michelson updates
- **Jakartha – 08/22**
  - Transactional Optimistic Rollups (TORU) to provide a Layer 2 scaling solution
  - Liquidity baking vote to signal interest in liquidity baking

Switching Consensus Mechanisms On-Chain – An Open Heart Surgery

- With the Ithaca amendment, Tezos **switched its consensus type** from a Nakamoto consensus (Emmy*) to a BFT-style consensus (Tenderbake).

- There is no other network known that has established a succesful consensus mechanism switch on-chain.

- This is possible due to couple of factors:

    - A well designed amendment process

    - A community that is open to change

    - Extensive test network infrastructure and intensive software testing

- A change of this scale proves that Tezos has established a well functioning evolution management process.

Outline

1. Introduction to Tezos

2. Architecture Overview
   - Overview
   - Sybil Control Mechanisms
   - Consensus Protocols
   - Formally Verifiable Smart Contracts

3. Governance
   - Overview
   - Amendments Process

4. Ecosystem

Tezos Improvement Proposal (TZIP)

- Like Bitcoin (BIP) and Ethereum (EIP), Tezos also has an improvement proposal standard called **TZIP**.

- TZIP documents contain enhancements to the Tezos protocol such as new standards, features, and tools.

- Complements the on-chain governance process.

- **TZIP-12[1]**
  - Provides token-agnostic (i.e., fungible, non-fungible) implementation standards (like ERC-1155). These standards are referred to as [FA2](#).
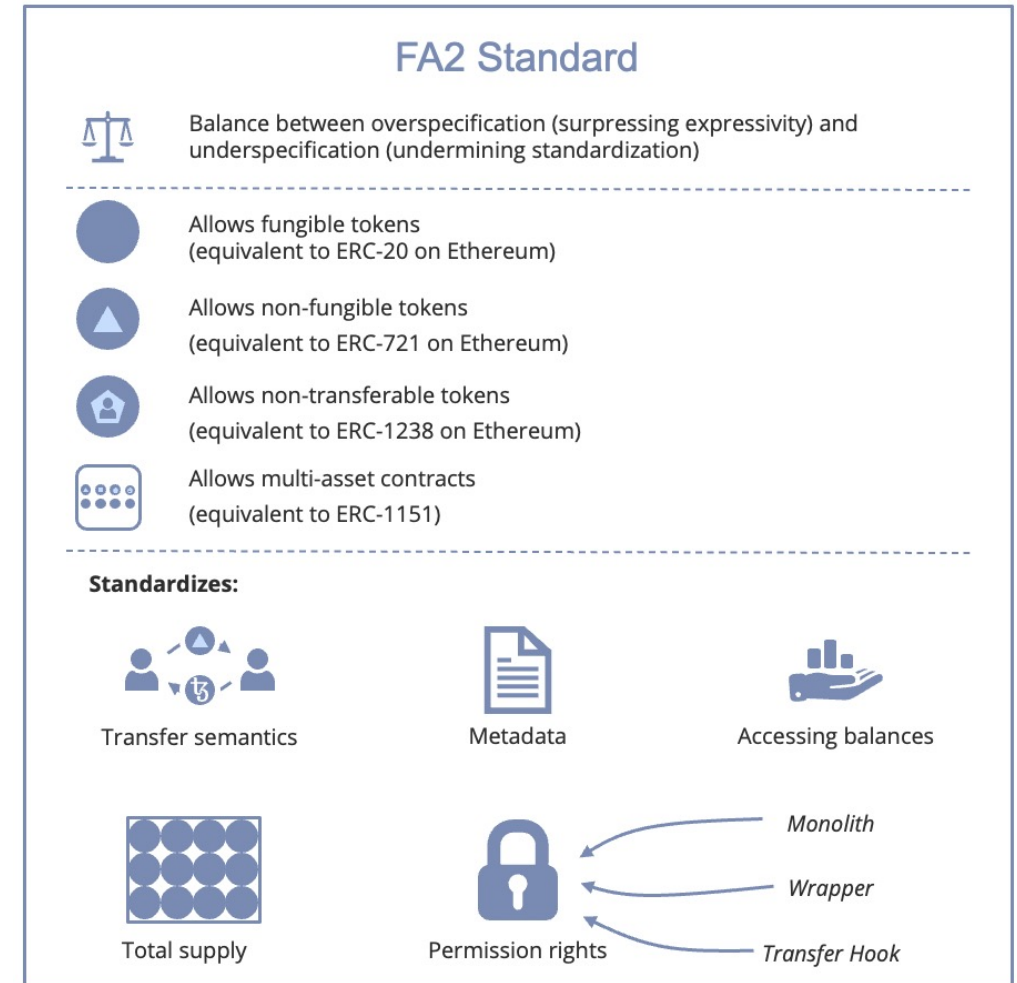
Screenshot taken from the [slides](#) of Niko Hildebrand (ASCS), prepared for the ENVITED academy

[1] More info about TZIP-12: https://tzip.tezosagora.org/proposal/tzip-12/
Here you can find the TZIP reference repository: https://gitlab.com/tezos/tzip/-/tree/master/proposals

Tezos Use Cases

Decentralized Finance (DeFi)

Decentralized Finance is the collection of "Centralized Finance" ecosystem services that exist in a decentralized, peer-to-peer network. Tezos enables some of the most popular DeFi components such as decentralized exchanges, price oracles, wrapped assets, and stablecoins.
Popular examples: Quipuswap, Plenty, Harbinger

Non-Fungible Tokens (NFTs)

At a very high level, NFTs are unique cryptographic tokens that are stored on a blockchain. There are many types of NFTs. They can be anything digital such as drawings, music, media, virtual fashion items, etc.
Popular examples: tzcolors, fxhash, Hicetnunc, TEIA

Decentralized Autonomous Organization (DAO)

DAOs are non-hierarchical business models that are collectively owned by a group of members. Think of it as a generalization of multi-signature wallets where all decisions are taken based on members' votes. Thus, no single member can unliterally take a decision (e.g., send a transaction). DAOs are useful for starting an organization with people you have limited trust (e.g., only communicating over the internet) since you only need to trust the system design and no other single member.
Popular examples: baseDAO, Homebase

Self-Sovereign Identity (SSI)

Self-sovereign identity technology enables control of digital identities by entities who own them (e.g., individuals, companies, …). Spruce Systems are currently developing a suite of tools for bringing SSI to Tezos.

Decentralized Applications in Tezos

- Decentralized applications (dApps) enable interaction with the Tezos network and smart contracts running on it.

- Beacon (TZIP-10) defines how a wallet interacts with a dApp.
    - A dApp can implement the Beacon SDK to create a p2p connection with a wallet instance (e.g., for signing transactions)

- Like Web3js in Ethereum, Taquito is a TypeScript library suite for connecting dApps to the Tezos blockchain.

- Some dApp examples on Tezos:
    - tzbutton, Tezos Domains, more


```
import { TezosToolkit } from '@taquito/taquito';
const Tezos = new TezosToolkit('http://localhost:8732');
const wallet = Tezos.setProvider({ wallet: walletOfYourChoice }); // use the wallet of your choice

const userBalance = await Tezos.tz.getBalance('tz_address');

const contract = await Tezos.wallet.at('contract_address');

const counter = await contract.storage();

const op = await contract.methods.increment(counter + 1).send();
await op.confirmation();
```

A code snippet for connecting to a full node and calling a method on a smart contract with Taquito

Key Takeaways

Benefits

- On-chain governance
- Proven ability of changing
- Active and supportive community
- Formally verifiable smart contracts
- Successfully running PoS
- Deterministic finality with Tenderbake
- Strong academic background

Drawbacks

- Fierce competition with other smart-contract enabled platforms such as Ethereum and Solana.

- No widespread commercial adoption yet.