OST
Eastern Switzerland
University of Applied Sciences

Distributed Systems Advanced & Blockchain

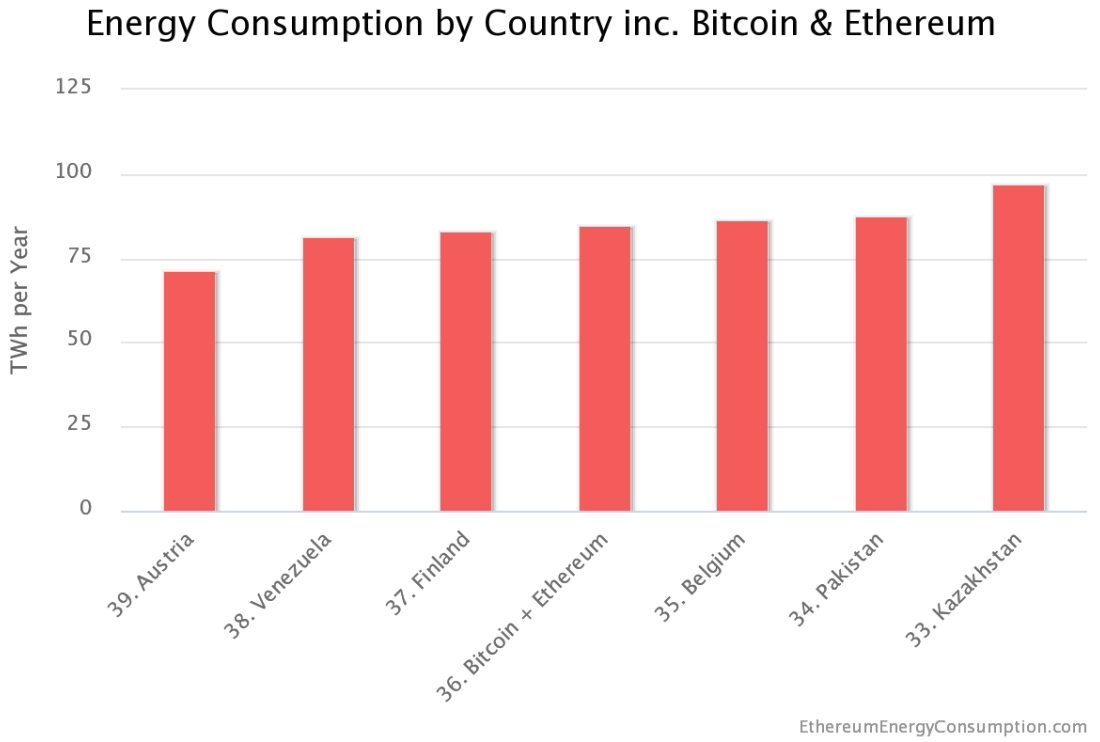
Tezos

Thomas Bocek

18 October 2020

About

- Tezos is a blockchain “similar” as Bitcoin, Ethereum
- Bitcoin: 09.01.2009, Ethereum: 30.06.2015, Tezos: 30.06.2018
- Bitcoin, Ethereum PoW, energy
- Tezos: proof-of-stake (PoS) , currency: XTZ
  - Described in whitepaper, 02.09.2014
  - Baking: > 8000XTZ (18K USD), 475 Bakers, 5.5-6%
    - Baking = staking, creating blocks, you can delegate
    - Slashing if double baking, can be expensive…
  - Wallets: AirGap, Atomic Wallet, Atomex, …
- Onchain governance: stakeholder can vote on protocol change
  - Proposal: “hey, staking should be much lower, who is in favor?” (its a bit more complicated), e.g., testing phase



History

- Tezos registered in Zug
  - Foundation: fundraising, promoting and sponsoring of Tezos protocol
  - Company: developing the Tezos protocol
- Fundraising during the ICO craze (2017)
  - Raised 232$ millions in 2 weeks, 01.07.2017-14.07.2017
  - 65’681 BTC and 361’122 ETH [stats]
- Development open source, GitLab
  - OCaml: functional, imperative and object-oriented styles, maintained by Inria
  - Dune Fork - coinmarketcap
- Foundation issues
  - 18.10.2017: The Path Forward
    - Slow development
  - 19.10.2017: Tezos Fights Over ICO Millions
    - “Tezos plans to pay its shareholders a 8.5 percent cut”
    - “Gevers claims the Breitmans are unwilling to let the foundation operate independently”
    - “relationship soured when Gevers tried to help himself to a bonus out of the ICO proceeds”
  - 22.02.2018: Johann Gevers steps down from Tezos Foundation board
    - 27.8.2018: “Es ist ruhig geworden um die Tezos Foundation aus Zug. Und das ist ein gutes Zeichen für das Blockchain-Projekt.”

Smart Contracts

- Ethereum: Solidity, Bitcoin: Opcodes, Tezos: Michelson - functional oriented
- Two type of accounts
  - Contract account with smart contract and tokens, “regular” accounts with tokens only
  - Michelson: cannot distinguish → Solidity ERC-677
  - Light polymorphism supported: my advice - keep contracts simple
- Types (arbitrary precision!)

- Michelson online editor
- “anyone can provide any amount of funds. After the timestamp, the contract stops accepting new transactions and the winner is the person with the highest bid. The item is distributed off chain. I allow people to increase the bid by any amount, as an exercise, change this. … Signing a message with the winning key allows for the item to be delivered only to the correct person. ” [source, TLS!]

```
parameter key_hash;
storage (pair timestamp (pair tez key_hash));
return unit;
code { DUP; CDAR; DUP; NOW; CMPGT; IF {FAIL} {}; SWAP; # Check if auction has ended
       DUP; CAR; DIP{CDDR}; AMOUNT; PAIR; SWAP; DIP{SWAP; PAIR}; # Setup replacement storage
       DUP; CAR; AMOUNT; CMPLE; IF {FAIL} {};  # Check to make sure that the new amount is greater
       DUP; CAR;                               # Get amount of refund
       DIP{CDR; DEFAULT_ACCOUNT}; UNIT; TRANSFER_TOKENS; # Make refund
       PAIR}                                             # Calling convention
```

Smart Contracts

- Gas similar to Ethereum, block gas limit: 10.4m gas, 8 types of costs to calculate gas (optimize):
  1. Reading cost
  2. Deserialization cost
  3. Parsing cost
  4. Type comparison cost
  5. Interpreter cost
  6. Unparsing cost
  7. Serialization cost
  8. Writng cost
- You don't want to write EVM Opcodes
  - Bitcoin, you write opcodes…
  - Tezos you can: "Michelson is designed as a readable compilation target, though it can be handwritten" (tutorial)
- From VSS, FS2020:
  - Ivy - a non Turing complete higher-level language that compiles to Bitcoin Script - Experimental
  - Rootstock
    - Rootstock (RSK) is a smart-contract platform with a Turing Complete VM for Bitcoin.
    - Bitcoin Sidechain, 2-way pegging
    - Backward compatible with Ethereum VM
  - Simplicity - a typed, combinator-based, functional language without, by Blockstream
    - Definition fits on a t-shirt
  - Miniscript: language for writing Bitcoin scripts in a structured way

EVM	WASM	Michelson
256 ints	32 / 64 ints	Infinite precision ints
No data structures	No data structures	Persistent sets, maps, lists
Side effects	Side effects	No side effects
Purpose made	Standard	Purpose made
Ethereum	Dfinity, EOS	Tezos

https://learn.tqtezos.com/files/language.html

Smart Contracts

- LIGO: A friendly Smart Contract Language for Tezos
  - 3 dialects, Pascal, OCaml, Reason, documentation
  - VSCode extensions (syntax highlith, live compilation)

Smart Contracts

- Liquidity Lang
  - Functional language, works for Tezos and Dune
  - Targets Dune, 1 main contributor
  - LOVE: a New Smart Contract Language for the Dune Network
    - “Michelson: it has a limited expressiveness due to its reduced set of instructions and does not allow code sharing”
    - “Solidity: it is an expressive and feature-rich language, allowing to easily write a broad range of smart contracts, but does not offer strong safety guarantees.”
- Others
  - https://medium.com/protofire-blog/choosing-a-language-for-smart-contracts-on-tezos-125c4e50552e
    - Ligo, SmartPy (use Python), Morley (Haskell), Fi (Javascript, learn), Archetype
  - https://tezosukraine.medium.com/tezos-programming-languages-f5d1ed8884b
    - Albert
- So many languages, which to choose?

FA1.2 – ERC-20

- Approvable Ledger Interface
- Entrypoints:
  - (address :from, (address :to, nat :value)) %transfer
  - (address :spender, nat :value) %approve
  - (view (address :owner, address :spender) nat) %getAllowance
  - (view (address :owner) nat) %getBalance
  - (view unit nat) %getTotalSupply
- Only 3 Assets?
- Block Explorer, example tx fees: 3 cents

- Implementation in LIGO
- Implementation in Liquitidy → token.liq
- Implementation in SmartPy → FA1.2 token standard
- Implementation in Fi (n/a)