ERGO

January 2019

Dmitry Meshkov

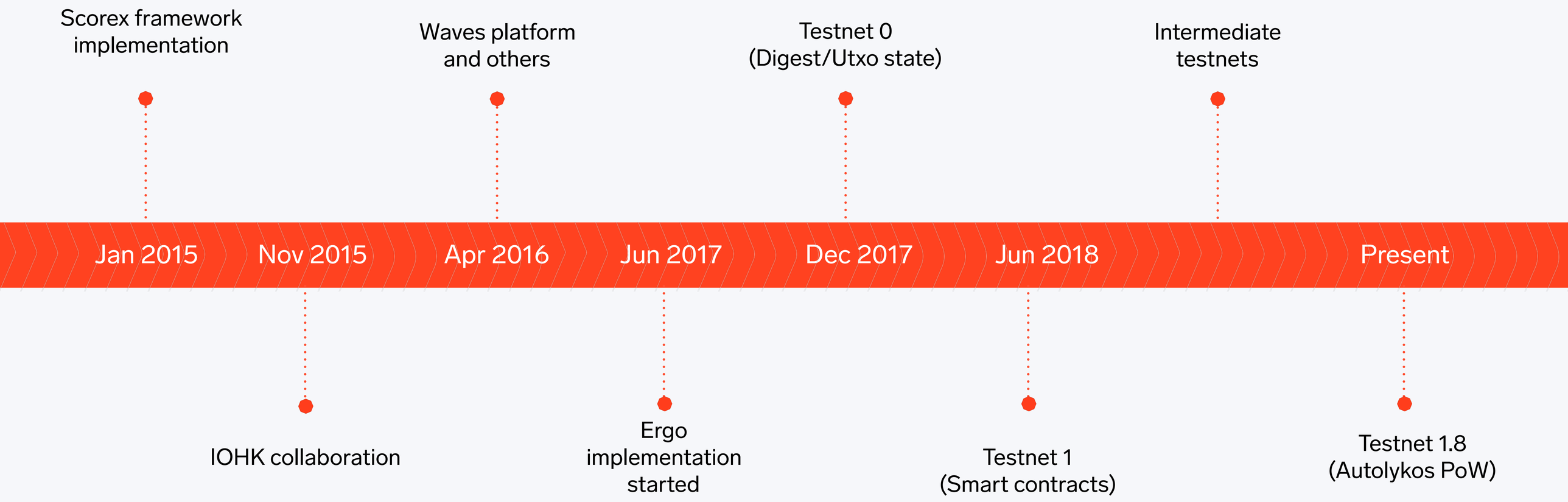
# Ergo platform: from prototypes to a survivable cryptocurrency

# Outline

- Ergo vision
- Consensus protocol
- Light clients
- Storage fee
- Voting system
- Smart contracts
- Monetary settings
- Roadmap

# Ergo Vision

# History


Scorex framework implementation
Waves platform and others
Testnet 0 (Digest/Utxo state)
Intermediate testnets
Jan 2015
Nov 2015
Apr 2016
Jun 2017
Dec 2017
Jun 2018
Present
IOHK collaboration
Ergo implementation started
Testnet 1 (Smart contracts)
Testnet 1.8 (Autolykos PoW)

# Vision

## Why to start a new cryptocurrency?

- Huge hype of cryptocurrencies, but technology stuck
- Blockchain 2.0, 3.0, ..., while actually we are still at 1.0
- New protocols are trying to achieve high throughput, complicated smart contracts, ...
- .. while sacrificing decentralization, promising that it will be achieved somewhere in the future

# Vision

## Ergo idea

- Blockchain 1.1 — a major update to blockchain technology without breaking changes
- Truly decentralized system
- Long-term survivability
- Fundamental approach
- Friendly for clients and applications

# Vision

## Who should be interested in Ergo

- Users - control over your money via trustless light clients in decentralized and stable permissionless-blockchain
- DApp developers - Ergo has secure and flexible smart contract language, native multi-token support and fast verification
- Long-term investors - survivabile cryptocurrency with no ICO and premine and strictly limited supply
- Miners - Ergo mined from zero with no-premine and gives miners on-chain voting and a strong role in governance
- And others...

> Ergo may appeal to the entire spectrum of crypto currency users given its diverse set of features and fundamental focus on decentralization and security.

# Autolykos consensus protocol

# Consensus: Why Proof-of-Work?

1. Widely studied and tested
2. Have high security guarantees
3. Allows new members to join the network
4. Light validation allows to use the blockchain without third parties

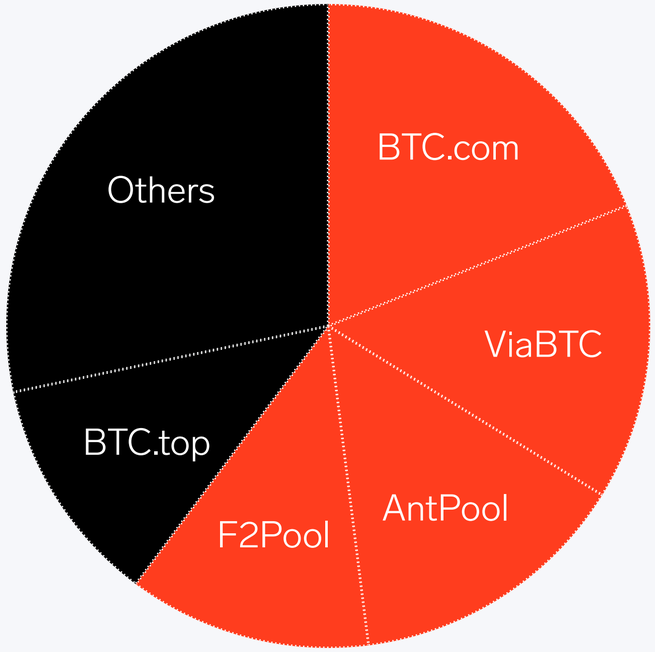
# Consensus: Known Proof-of-Work Drawbacks

1 ASICs – centralize the network around ASICs manufacturers

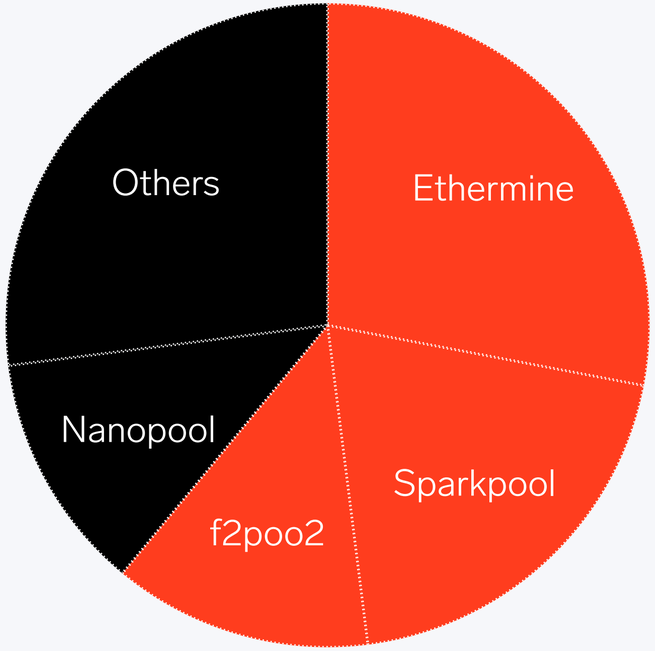
2 Mining pools - centralize the network around pool operators

# Consensus: Mining pools

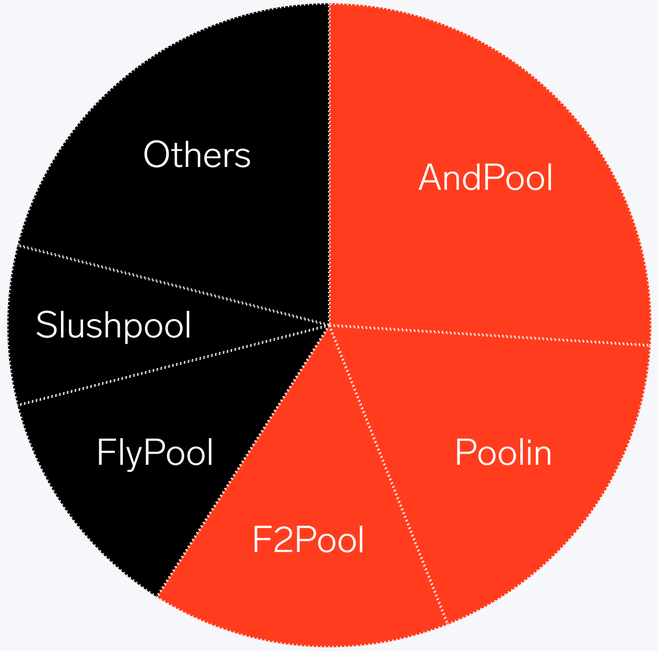
Regardless of the PoW algorithm, 2-4 pools control the network




BTC




ETH




ZEC

Hashrate distributions for 06.11.2018 (for 24 hours), taken from blockchain.com, etherchain.org, explorer.zcha.in

# Consensus: Autolykos

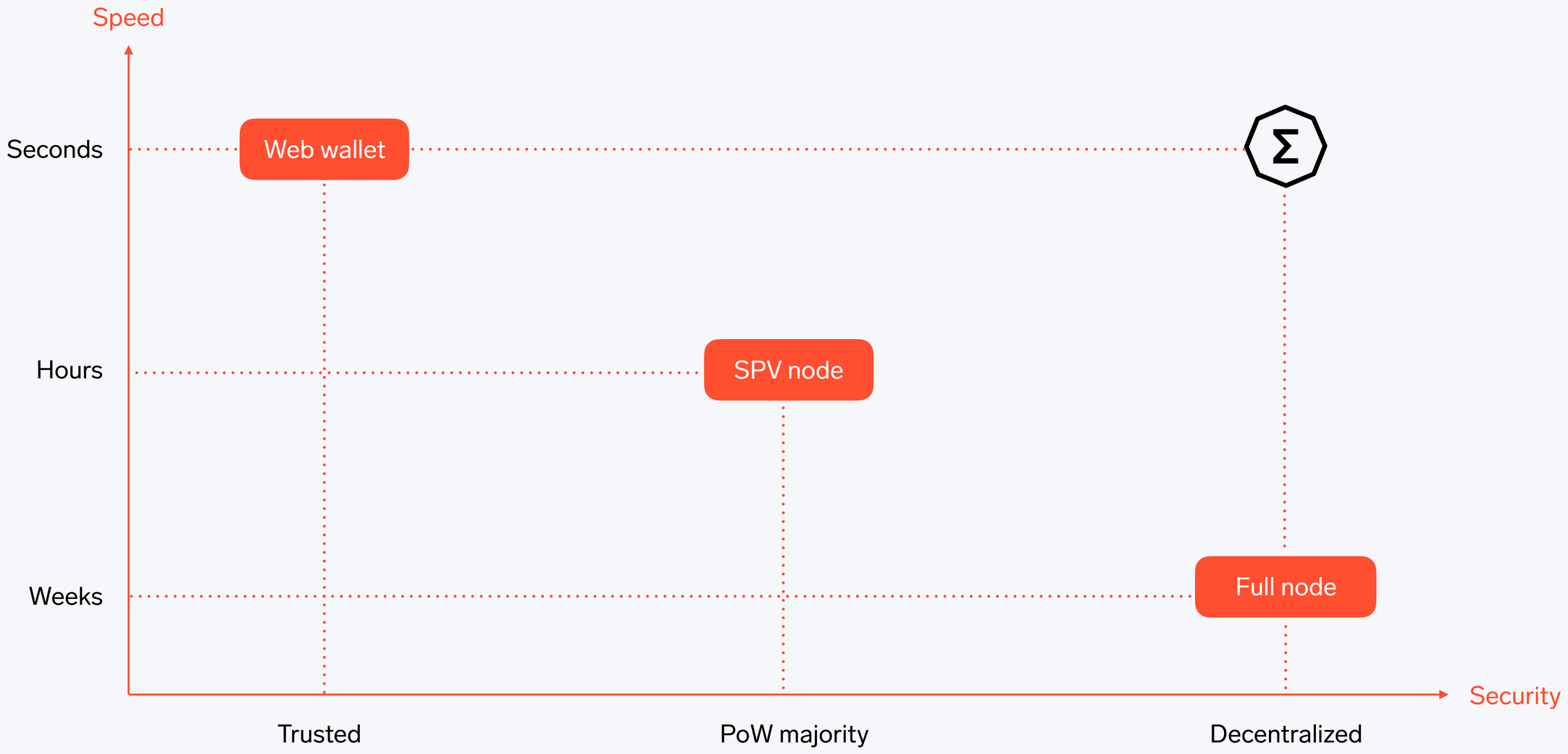
- Combination of Equihash and Schnorr signatures
- Solution search requires 2Gb of memory and is done over secret keys
- Solution verification requires 2Kb of memory and is done over public keys
- Efficient in terms of solution size and verification time

# Light clients

# Light clients: Now

- You must set up a node or trust someone
- Node synchronization is slow, unreliable and resource intensive
- Regular users resort to trusted solutions
- If service provider is hacked (or become malicious), users may lose their funds
- And may not even notice this, because they use trusted block explorer
- Better alternatives (e.g. SPV nodes in Bitcoin) exist, but only allows to validate some subset of rules

# Light clients: Ergo



# Light clients: Ergo

- Ergo block header supports Non-Interactive Proofs of Proof-of-Work, that allows to synchronize the network, by downloading < 1Mb of data
- Ergo state is authenticated, which enables verification of transactions without any trust and without keeping the entire state
- Flexible configuration for facilitated node regimes

# Node regimes: Ergo

It is possible to use Ergo from a smartphone without any trust.

It is possible to join the network and start mining within a few hours.

No material performance degradation over time.

# Demurrage component

# Demurrage: Cryptocurrency fees

- Transaction utilizes 3 types of resources: network, CPU, storage
- Bitcoin - fees for transaction size (network resources)
- Ethereum – fees for gas consumed (CPU resources)
- In both system user can put a data for miners storage forever
- Ergo - fees for storage consumed

# Demurrage: Storage rent

- Demurrage – payment from users to miners for keeping their data in the state
- Similar to regular cloud storage payment is proportional to space*time
- Payment is collected from the box once per 4 years
- If there are not enough coins in the box at this point – it is removed from the state.
- Storage price may be changed via miner votes

# Demurrage: Effects

- Upper-bound of the state size become predictable
- Prevent circulating supply decrease due to lost keys, incorrect contracts, etc.
- Stabilizes mining by providing additional fixed reward
- Incentivizes people to use their money

# Voting system

# Development

- Environment is not static, therefore the network should also be changeable
- But how to make these changes?
- A decentralized cryptocurrency should avoid a dedicated "core" team
- The network should achieve long-term survivability without promised further changes

# Development: Voting protocol

- Ergo allows to change a lot of parameters via miners voting:
  block size, contract costs, demurrage coefficients and more...
- Parameters are changed with a small step (1% per 1024 blocks)
- But it allows to make big changes step by step

# Smart contracts

# Smart contracts: Smart money

- 2 main directions: protecting script (e.g. BTC) vs preform computations (e.g. ETH)
- Ergo – protecting script, platform for smart money
- Complicated protecting scripts (like multisig) are natural
- Computations (onchain or offchain) are also possible

# Ergo script: Idea

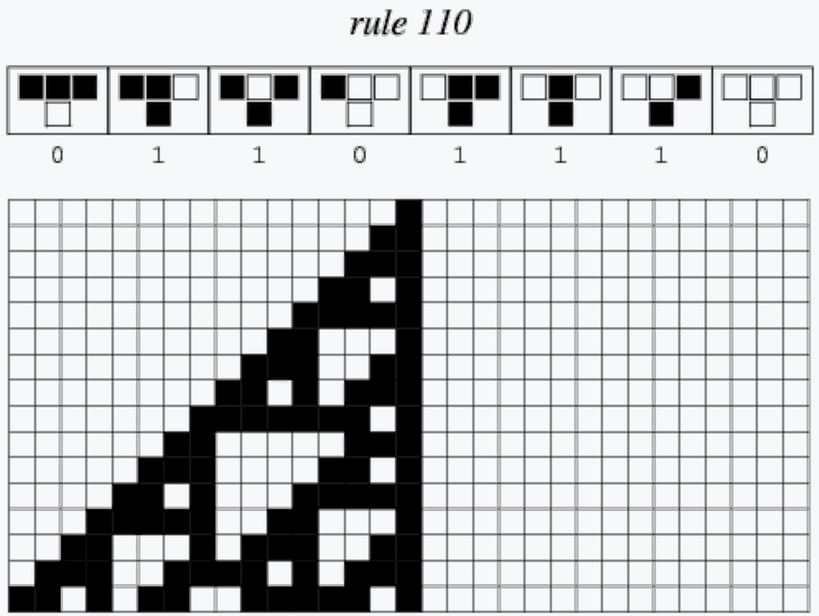
- Language for verifications rather than computations
- Strict upper-bound for computation time
- Only operations, that allow to estimate script complexity before execution
- Constant-time access to environment (few last headers)
- Based on Σ-protocols

# Smart contracts: Computations

- Even Bitcoin script allows to implement a lot of contracts: https://en.bitcoin.it/wiki/Contract
- But what is possible?
- Assumed to be not Turing complete (because of lack of loops)

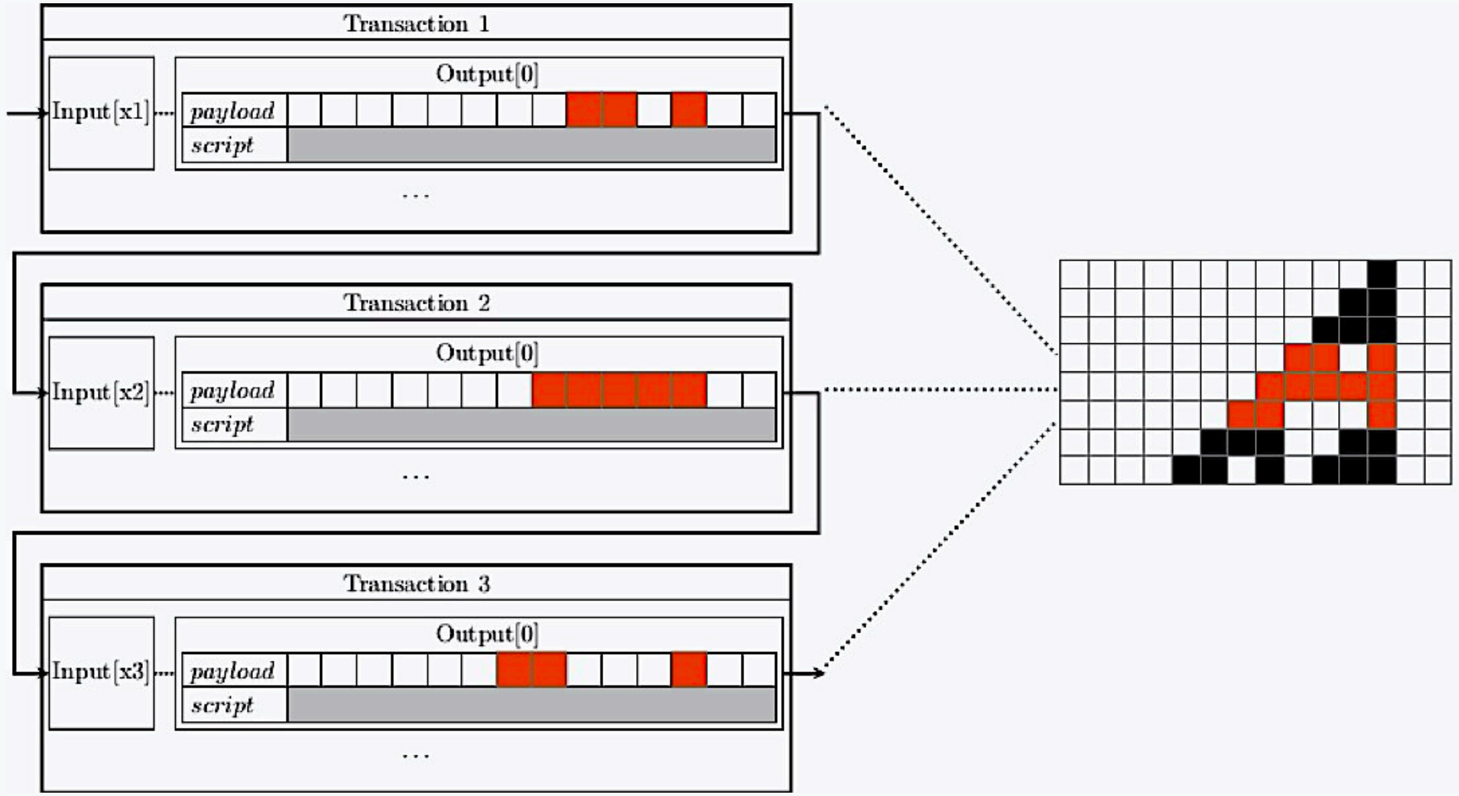
# Ergo script: Chaining

- But Turing-completeness may be achieved without loops inside the languages
- Turing completeness proof – implementation of known Turing complete system
- Rule 110 was implemented in Ergo script (see http://arxiv.org/pdf/1806.10116v1)



# Ergo script: Chaining

Even if you don’t have infinite loop inside a block, you have it between blocks



# Ergo script: Chaining

**If you need some computation:**

- Estimate work done before execution
- Put it to one or multiple transactions

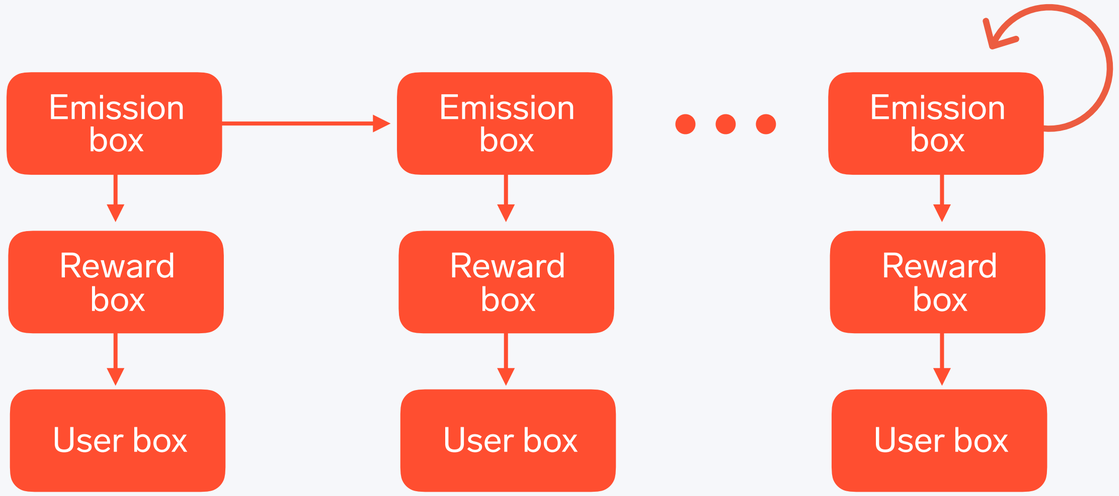
Number of blocks is infinite

↓

Possibly infinite loop

# Ergo script: Emission

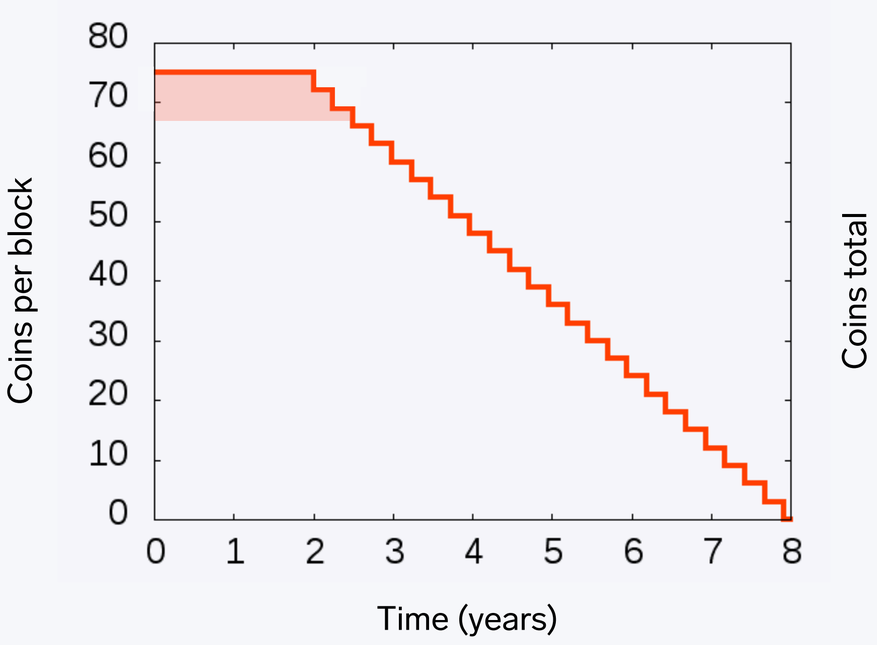
- Atomic swaps, DEX, crowdfunding, rule 110 and more at https://git.io/fpDhE
- Emission box: every block miner can take a part from it, returning the rest to the same script
- It should be spent in such a transaction, that has exactly one output, which creation height is current height, and proposition is: Height >= SELF.creationHeight + 720 && minerPk



# Monetary Settings

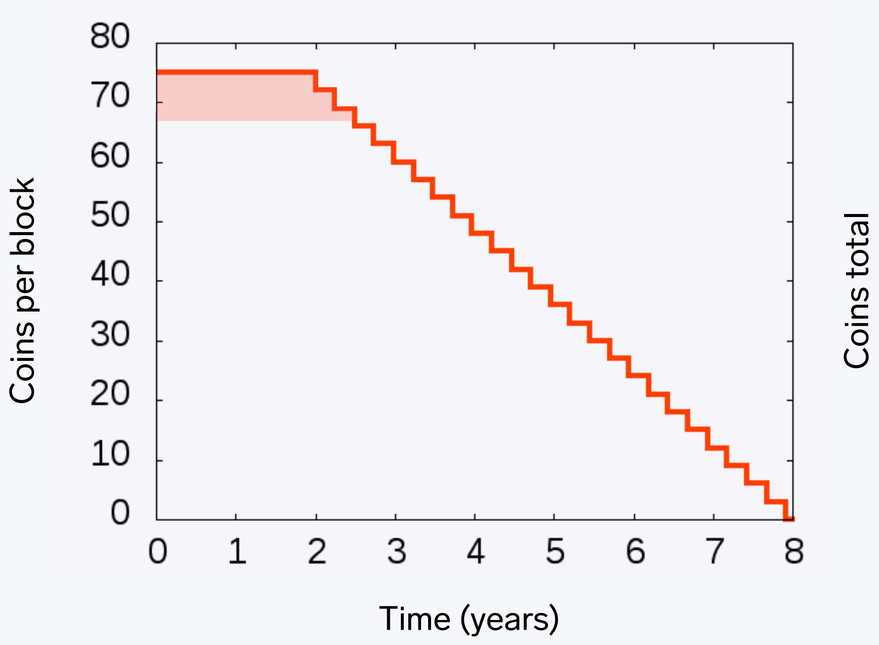
# Monetary settings

- No ICO and premine
- 97739925 coins total after 8 years emission
- Part of the emission goes to a treasury to fund the development
- For the first 2 years, block reward is 75 Erg, 7.5 Erg (10%) of them goes to foundation
- After that treasury part reduces for 3 coins every 3 months



# Monetary settings

- EFYT token was issued at a start of implementation
- First year foundation reward will be used to cover EFYT token with 1:1 rate
- After the first year the community will decide, where to spend these funds via voting



# Roadmap

# Roadmap


Testnet
(1.9-2.0-2.x)
releases soon...
Wallet
Foundation
Website, blog, ...
Mainnet release
in Q2 2019
GPU miner
Exchange listing
Security audit
& testing
Libraries
& tooling

# Thank You
The End

http://www.slideshare.net/DmitryMeshkov

https://twitter.com/DmitryMeshkov

catena@protonmail.com