Bitcoin:
Programming the Future of Money

Topics in Computer Science - ITCS 4010/5010, Spring 2025

Dr. Christian Kümmerle

Lecture 22

Scriptless Multisignatures & Consensus Changes

UNIVERSITY OF NORTH CAROLINA CHARLOTTE

Scriptless Multisignatures

MOTIVATION: COMPLEX REDEEM SCRIPTS

Consider following setup:

- Mohammed, a company owner in Dubai, operates an import/export business; he wishes to construct a company capital account with flexible rules. The scheme he creates requires different levels of authorization depending on timelocks. The participants in the multisig scheme are Mohammed, his two partners Saeed and Zaira, and their company lawyer.

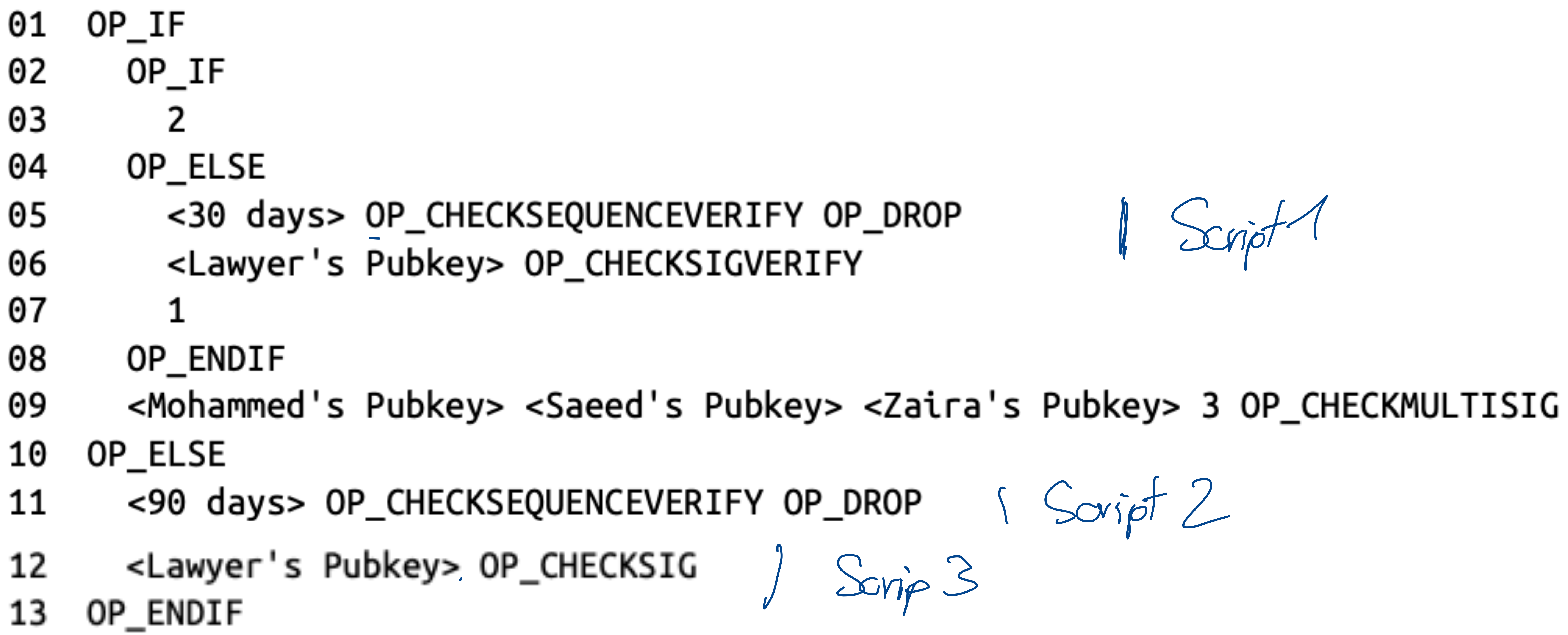
  The three partners make decisions based on a majority rule, so two of the three must agree. However, in the case of a problem with their keys, they want their lawyer to be able to recover the funds with one of the three partner signatures.

  Finally, if all partners are unavailable or incapacitated for a while, they want the lawyer to be able to manage the account directly after he gains access to the capital account's transaction records.

2-of-3 multisig

APPLICATION OF TAPROOT SPENDING CONDITION

Redeem script (as e.g. used in P2SH):



```
OP_IF
  OP_IF
    2
  OP_ELSE
    <30 days> OP_CHECKSEQUENCEVERIFY OP_DROP
    <Lawyer's Pubkey> OP_CHECKSIGVERIFY
    1
  OP_ENDIF
  <Mohammed's Pubkey> <Saeed's Pubkey> <Zaira's Pubkey> 3 OP_CHECKMULTISIG
OP_ELSE
  <90 days> OP_CHECKSEQUENCEVERIFY OP_DROP
  <Lawyer's Pubkey> OP_CHECKSIG
OP_ENDIF
```

Script 1

← pubkey

Script 2

Scrip 3

Q: How to encode script such that only the spending condition that is used needs to be revealed?

DIFFERENT APPROACHES TO MULTISIGNATURES

Goal: Set spending condition such that a subset of **at least k individuals** needs to sign off a transaction **among n possible individuals** (**“k-of-n multisig”**).
Approaches:

- **Scripted Multisignatures:** Set ScriptPubKey, e.g., as

  k <PublicKey1> <PublicKey2> ... <Public Key n> n OP_CHECKMULTISIG

- **Scriptless Multisignatures** (Schnorr-based):
  Implementation by aggregation of keys and aggregation of signatures of different individuals into one **public key (& associated signature)**.

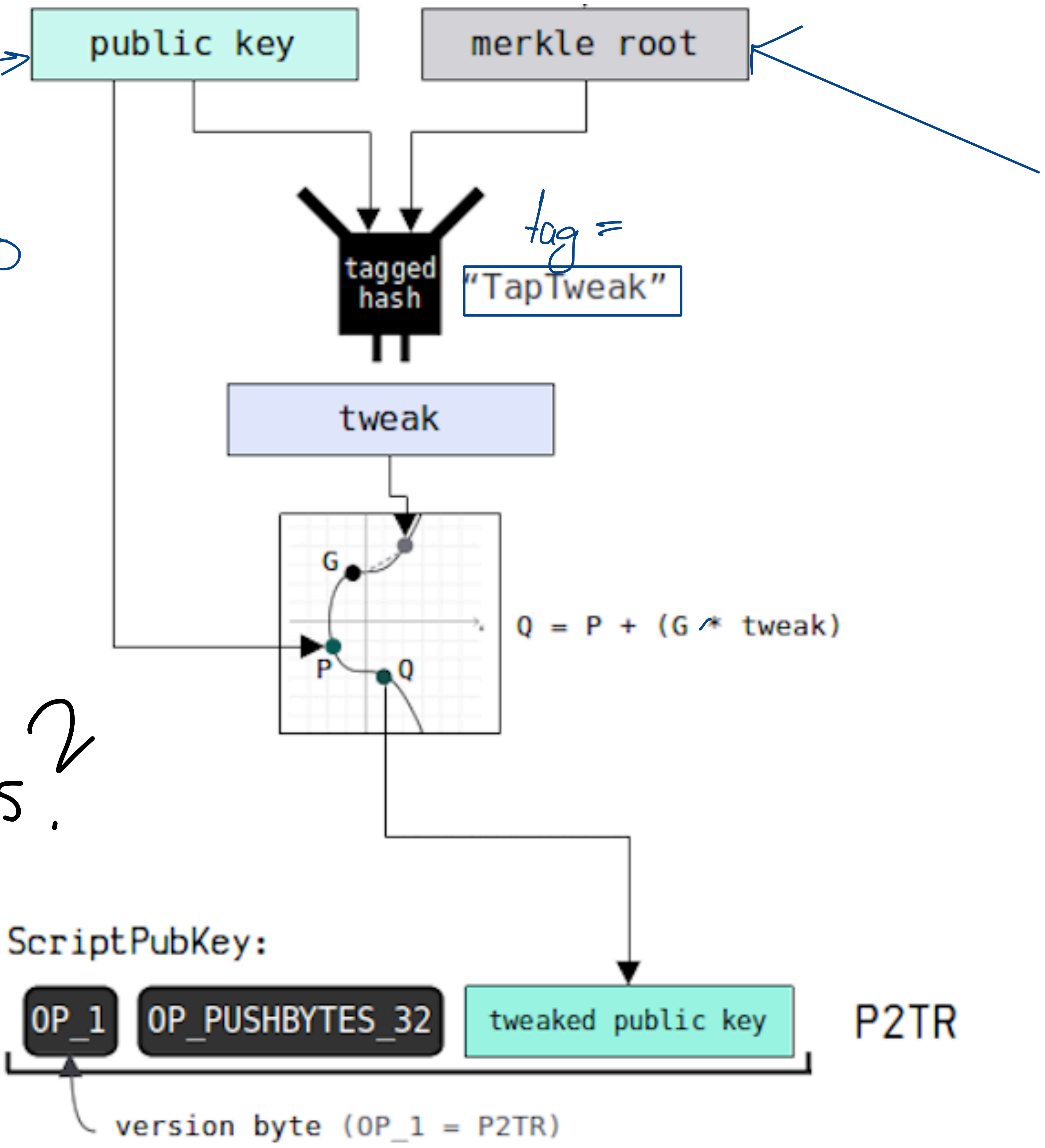
STRUCTURE OF P2TR SPENDING CONDITIONS

In example:

▷ Encode case that all three patners need to sign, i.e., a 3-of-3 multisig.

Q: How to implement this?

↳ Key aggregation protocol (e.g., MuSig, MuSig2)



▷ Encode information of last redeem script into a "tree of Scripts" Summarized by a Merkle root m.

RECAP: SCHNORR SIGNATURE SCHEME

```
SchnorrSign$(e, k, m)$
  ▷ $R = kG$
  ▷ Compute $z = hash(R || eG || m)$
  ▷ $s = k + z \cdot e$
  ▷ return $(s, R)$
```

private key P=eG is also concatenated here to avoid that a valid signature s' for "derived public child key" P+c can be created from valid signature s for public key P.

```
SchnorrVerify$(s, P, R, m)$
  ▷ Compute $sG$
  ▷ Compute $z = hash(\underset{agg}{R} || \underset{agg}{P} || m)$
  ▷ Compute $testval = zP + R$
  ▷ If $sG == testval$
        return True
    Else
        return False
```

Schnorr Signature scheme as defined in BIP340 (Bitcoin Improvement Protocol 340), used with secp256k1

NAIVE SCHNORR-BASED KEY & SIGNATURES AGGREGATION

Idea: Aggregated public key
$P_{agg} = P_0 + P_1 + \ldots + P_{n-1}$

Example: n-of-n multisig.

Assume: $d_0, \ldots, d_{n-1}$ are private keys of n individuals

$\hookrightarrow P_0 = d_0 \cdot G, \ldots, P_{n-1} = d_{n-1} \cdot G$ are public keys

1. For each $i \in \{0, \ldots, n-1\}$, sample random (private) nonce k_i, create public nonce $R_i = k_i \cdot G$
2. Individuals communicate their public keys P_i with each other
3. Individuals — " — public nonces R_i with each other
4. Each individual computes:
   - $P_{agg} = P_0 + P_1 + \ldots + P_{n-1}$
   - $R_{agg} = R_0 + R_1 + \ldots + R_{n-1}$
5. $\forall i \in \{0, \ldots, n-1\}$:
   - Create signature $s_i = k_i + hash(R_{agg} \| P_{agg} \| m) \cdot d_i$

   message to be signed

   ⚠ not secure!
6. - (s_i, R_i)

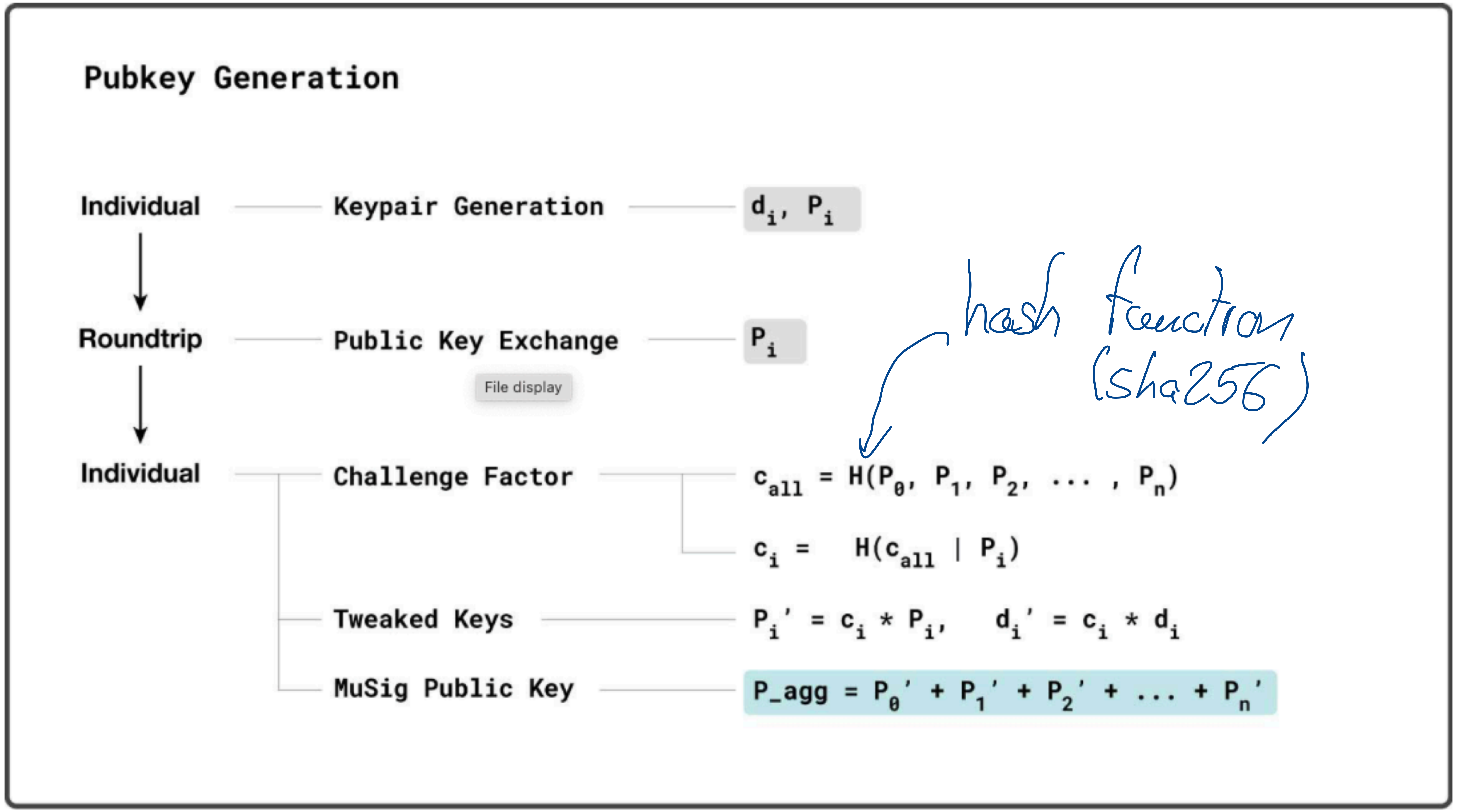
Goal: Aggregated signature $s_{agg} = s_0 + s_1 + \ldots + s_{n-1}$ is valid signature for P_{agg}

MUSIG: SIMPLE SCHNORR MULTI-SIGNATURES WITH APPLICATIONS TO BITCOIN

Proposed in 2018 by Maxwell, Poelstra, Seurin & Wuille

Setup Phase

- Create aggregated public key P_{agg}
- Involves one roundtrip communication round
- "Challenge factor" c_i unique for each participant



Pubkey Generation

Individual — Keypair Generation — d_i, P_i

Roundtrip — Public Key Exchange — P_i

Individual — Challenge Factor — $c_{all} = H(P_0, P_1, P_2, \ldots, P_n)$

$c_i = H(c_{all} \mid P_i)$

Tweaked Keys — $P_i' = c_i * P_i, \quad d_i' = c_i * d_i$

MuSig Public Key — $P\_agg = P_0' + P_1' + P_2' + \ldots + P_n'$

hash function (sha256)

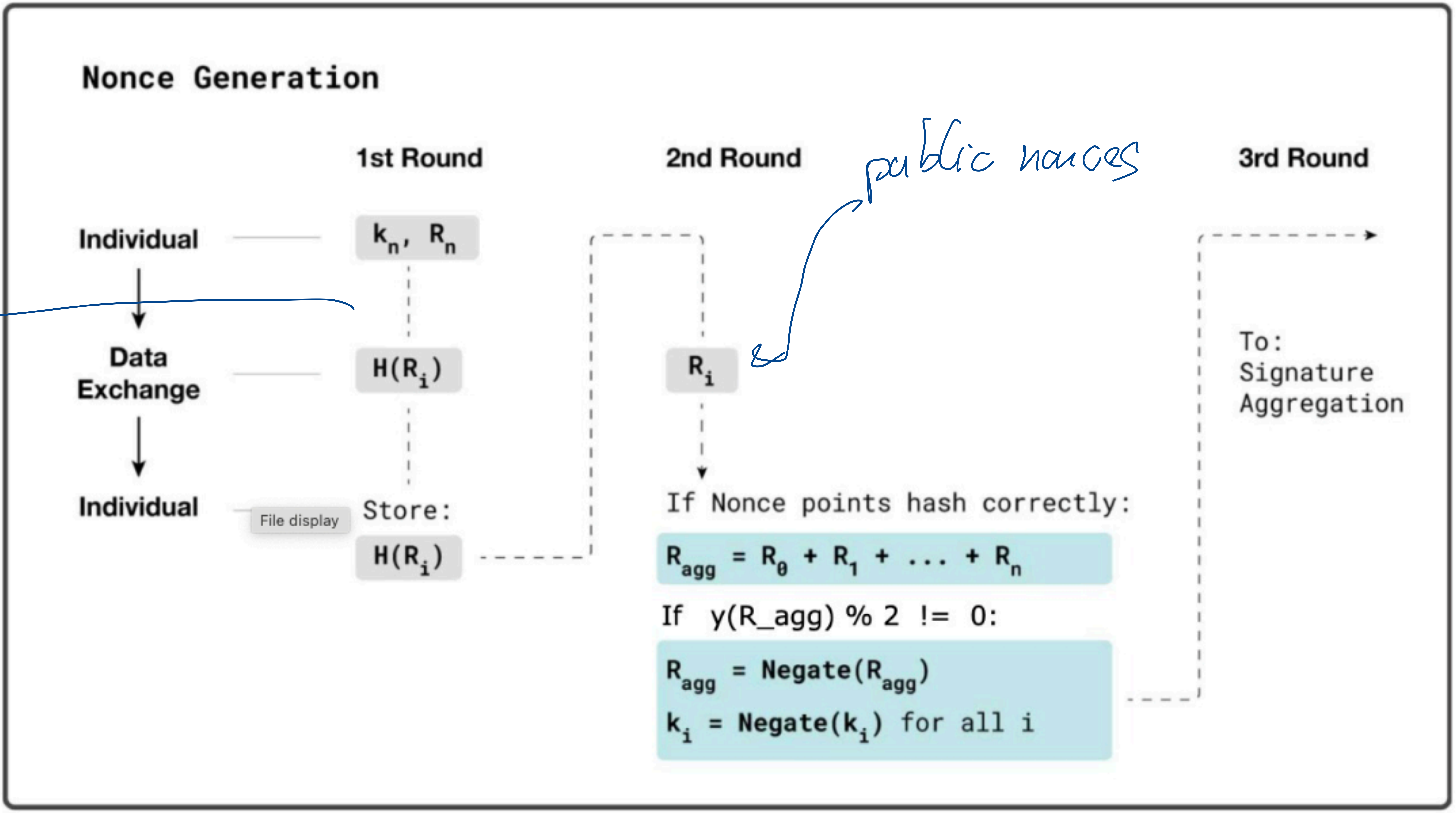
Previously: $P\_agg = P_0 + \ldots + P_n$

Nonce Generation

- Involves **three** rounds of communication

[illegible]

public nonces



Nonce Generation

1st Round

2nd Round

3rd Round

Individual

k_n, R_n

Data Exchange

$H(R_i)$

Individual

File display

Store:

$H(R_i)$

R_i

public nonces

If Nonce points hash correctly:

$R_{agg} = R_0 + R_1 + ... + R_n$

If y(R_agg) % 2 != 0:

R_{agg} = Negate(R_{agg})

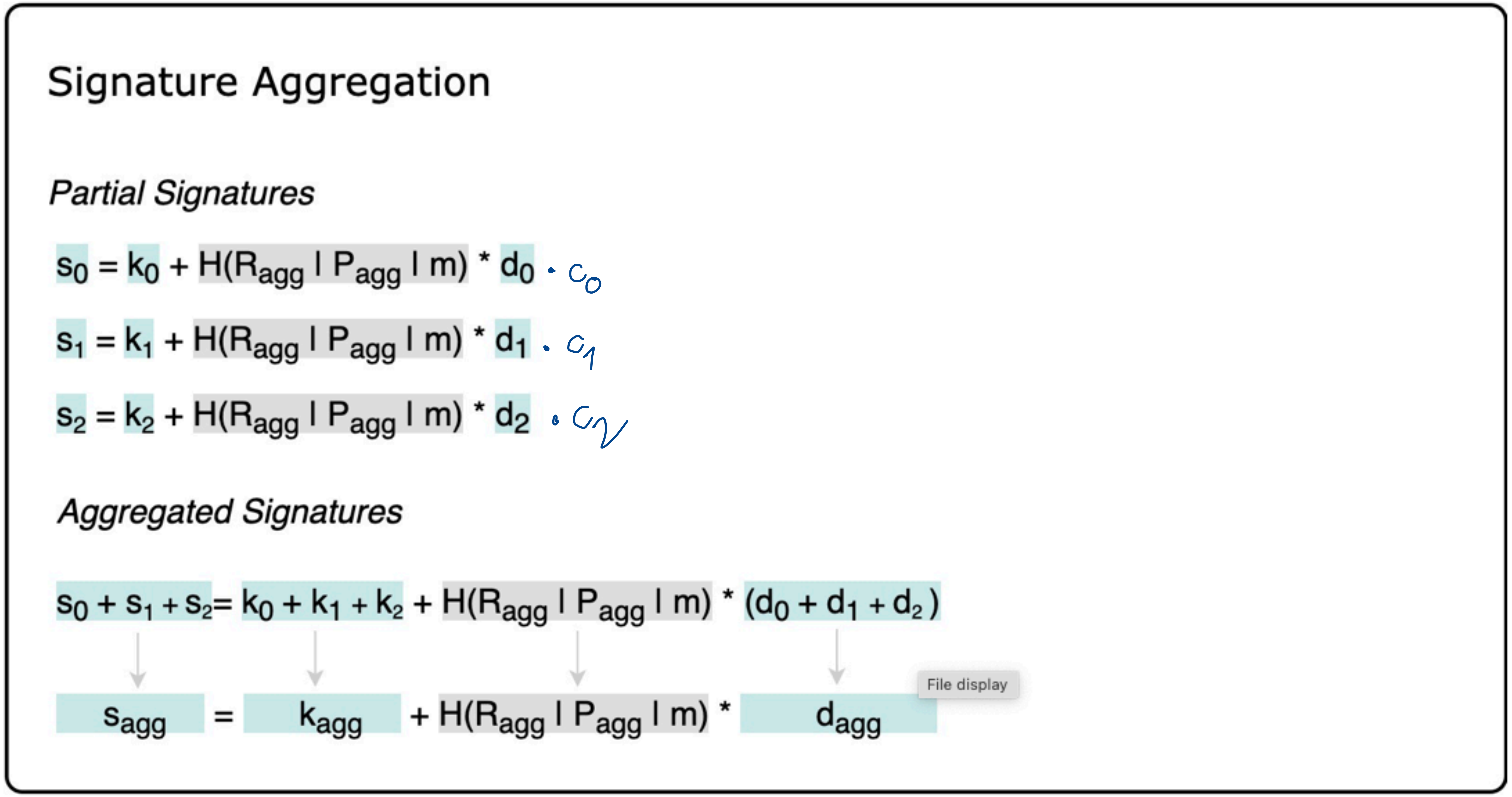
k_i = Negate(k_i) for all i

To: Signature Aggregation

Signature Generation

Example: 3-of-3 multisig

- Resulting signature $s = s_0 + s_1 + s_2$ is **valid Schnorr signature** for aggregated public key P_{agg}.



Signature Aggregation

Partial Signatures

$s_0 = k_0 + H(R_{agg} \mid P_{agg} \mid m) * d_0 \cdot c_0$

$s_1 = k_1 + H(R_{agg} \mid P_{agg} \mid m) * d_1 \cdot c_1$

$s_2 = k_2 + H(R_{agg} \mid P_{agg} \mid m) * d_2 \cdot c_2$

Aggregated Signatures

$s_0 + s_1 + s_2 = k_0 + k_1 + k_2 + H(R_{agg} \mid P_{agg} \mid m) * (d_0 + d_1 + d_2)$

$s_{agg} = k_{agg} + H(R_{agg} \mid P_{agg} \mid m) * d_{agg}$

SCRIPTLESS MULTISIGNATURES

- Provably secure; advanced version of MuSig ("MuSig2") specified in BIP 327 and already used with Taproot addressed
- Can be extended to k-of-n multisig with k < n

fewer rounds of communication

also exists "FROST"

Advantages vs. scripted multisignatures:

- More compact / less on-chain footprint (-> fee savings)
- Lower verification cost for Bitcoin nodes
- Higher level of privacy

Downsides vs. scripted multisignatures:

- Additional communication necessary between parties
- Security issues can arise if nonces are reused
- For "threshold" script less multisignatures: Less accountabilities

(i.e., k-of-n multisig)

What determines the consensus of what Bitcoin is?

THE DECENTRALIZED CONSENSUS OF THE BITCOIN NETWORK

We distinguish:

Cast class

- **Technical Consensus:**
  Does this transaction follow the rules? Is this block a valid Bitcoin block? If there are multiple branches of the Bitcoin blockchain, which is the right one?

- **Social Consensus:**
  What are “the rules” in the first place?

“Emergent Consensus”

TECHNICAL CONSENSUS

- Independent **validation** of each **transaction** by all full nodes
- Independent **aggregation** of (mempool) **transactions** into new blocks by miners
- Independent **validation** of each **new block** by all full nodes
- Independent **selection of the chain with the most cumulative proof-of-work** enforced by all full nodes

VALIDATION OF (NEW) TRANSACTIONS

For a Tx to be valid, the following rules need to be satisfied:

- Tx's syntax and data structure must be correct.
- **No inputs** nor **outputs** are **empty**.
- Transaction is **not too large** (in vbytes) to fit in a block.
- Each **output** value needs to be **in range of values** (0 < and < 21,000,000).
- Lock time is INT_MAX or follows rules of BIP68 (correct abs. or rel. lock time).
- **Scripts** of each **input must validate against respective output script** of referred UTXO.
- Not more **signature operations** than the set limit (up to 80000 per block).
- All rel. or abs. lock times are fulfilled, if Coinbase input, 100 confirmations until spendable

VALIDATION OF BLOCKS

For a block to be valid, the following rules need to be satisfied:

- Syntax of the block data structure needs to be correct (see also here).
- Block header hash is less than the target.
- Block time stamp is above the **Median Time Past** (See BIP113) (median time last 11 blocks in the chain).
- Block time stamp is below **Network Adjusted Time** plus two hours.
- Block size is below 1,000,000 vbytes.
- (Only) first transaction in transaction Merkle tree is the **coinbase transaction**.
- All transactions in block are valid.

SELECTION OF VALID BLOCKCHAIN

- **Key Rule:**
  The chain of blocks with largest cumulative proof-of-work is the valid one.

- Can be checked by adding up the hashes of all blocks.

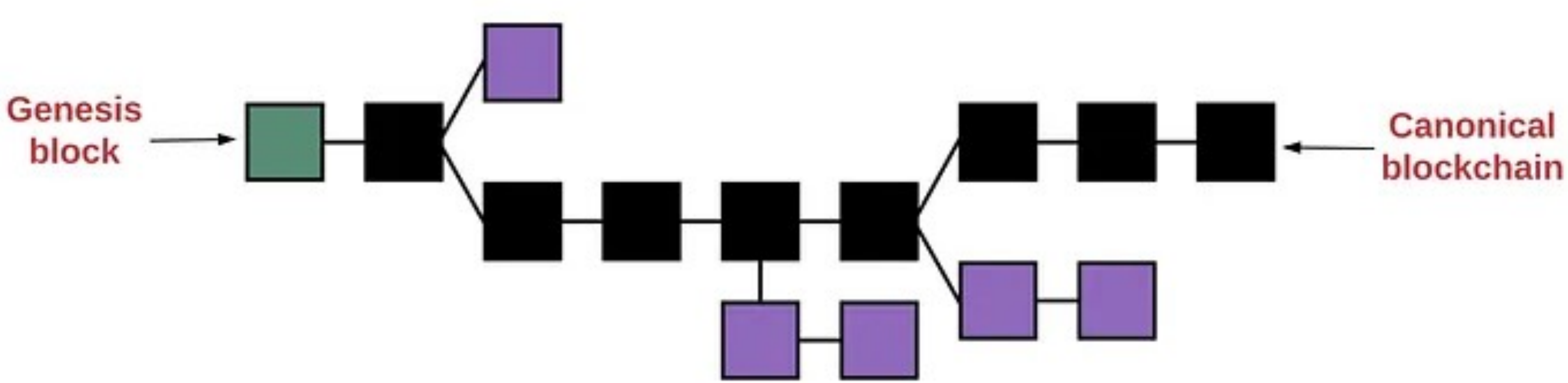
- Consequence:
  It is not uncommon that (short) **chain reorganizations** take place!

HASHRATE ATTACKS

How can malicious (group of) miners endanger the consensus?

- Empty-Block Attack
- Double Spend Attack
- Seesaw Attack

Usually require >51% of hashrate (or more than 30%).



Changes of Social Consensus

HARD AND SOFT FORKS

We distinguish:

- **Hard forks:**
  Change rules such that **new blocks / transaction types are not valid in the old rule set.**

-> Typically leads to chain split

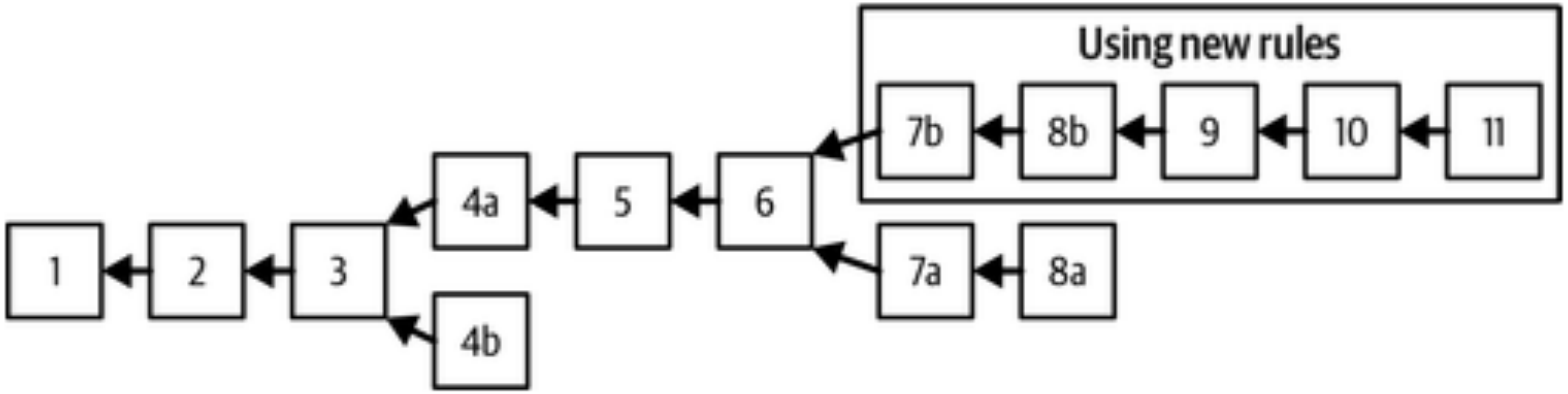
- **Soft forks:**
  Change rules such that **new blocks / transaction types are still valid in the old rule set, but old transaction/block rules are not (necessarily) valid in the new rule set.**

-> Typically does not lead to chain split

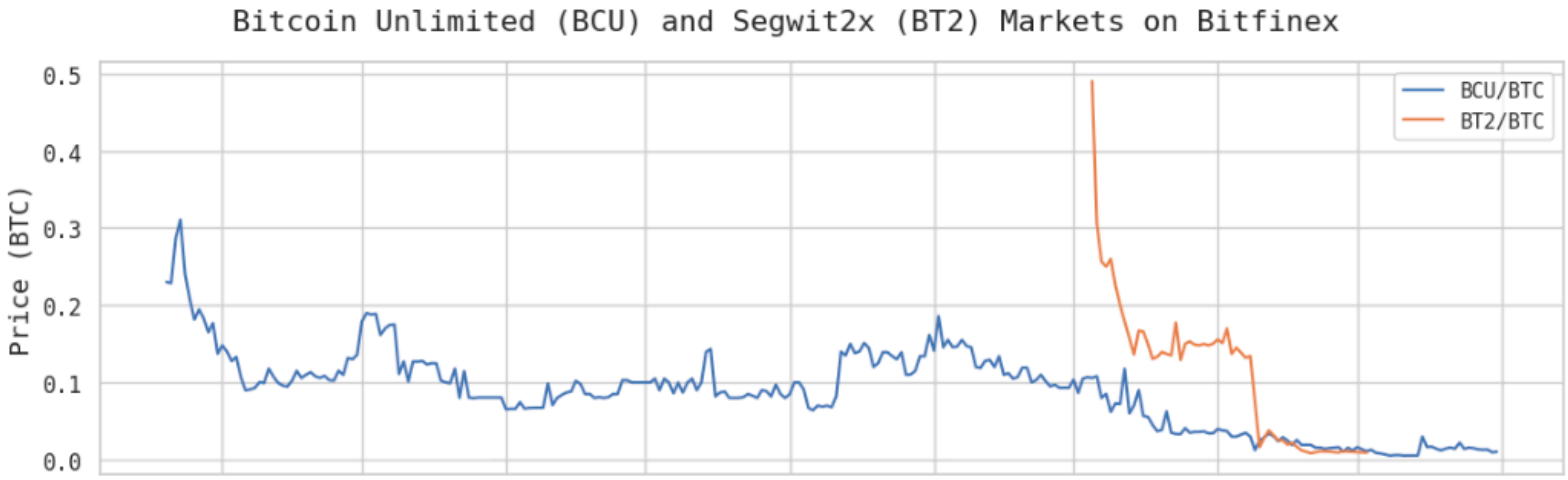
Some hard forks of Bitcoin

- 2010: Addition of OP_NOP opcodes by Satoshi Nakamoto (no chain split)
- 2016: Bitcoin Classic
- 2017: Bitcoin Cash ("The Blocksize Wars")
- 2018: Bitcoin-Satoshi's Vision

title of a book



MARKET VALUES OF FORK COINS



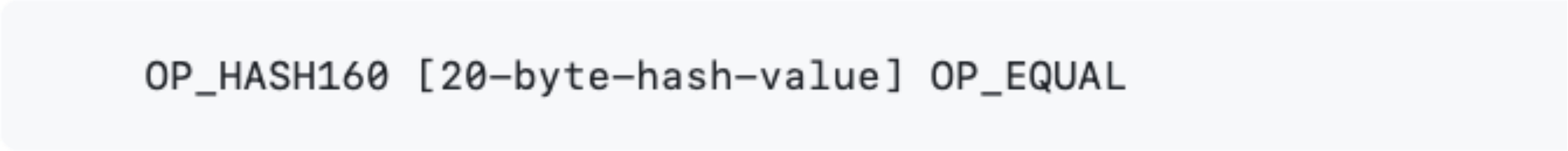
Some important soft forks of Bitcoin

- April 2012: Activation of P2SH format (BIP16)
- 2016/2017: SegWit Update
- November 2021: Taproot Update (BIP 340,BIP 341,BIP 342)
  Schnorr Signatures, Merklized Alternative Script Trees, Taproot Scripts

EXAMPLE SOFTFORK: ACTIVATION OF P2SH (BIP16)

Standard script validation rule before BIP16:

- ScriptPubKey



```
OP_HASH160 [20-byte-hash-value] OP_EQUAL
```

can be satisfied by providing the script whose hash160 image is [20-byte-hash-value]

Script validation rule after BIP16:

- As above, but also execute script <RedeemScript> and interpret remaining of ScriptSig information as Stack elements that need to successfully execute <RedeemScript>

How do we implement a soft fork?

SOFT FORKS OF BITCOIN

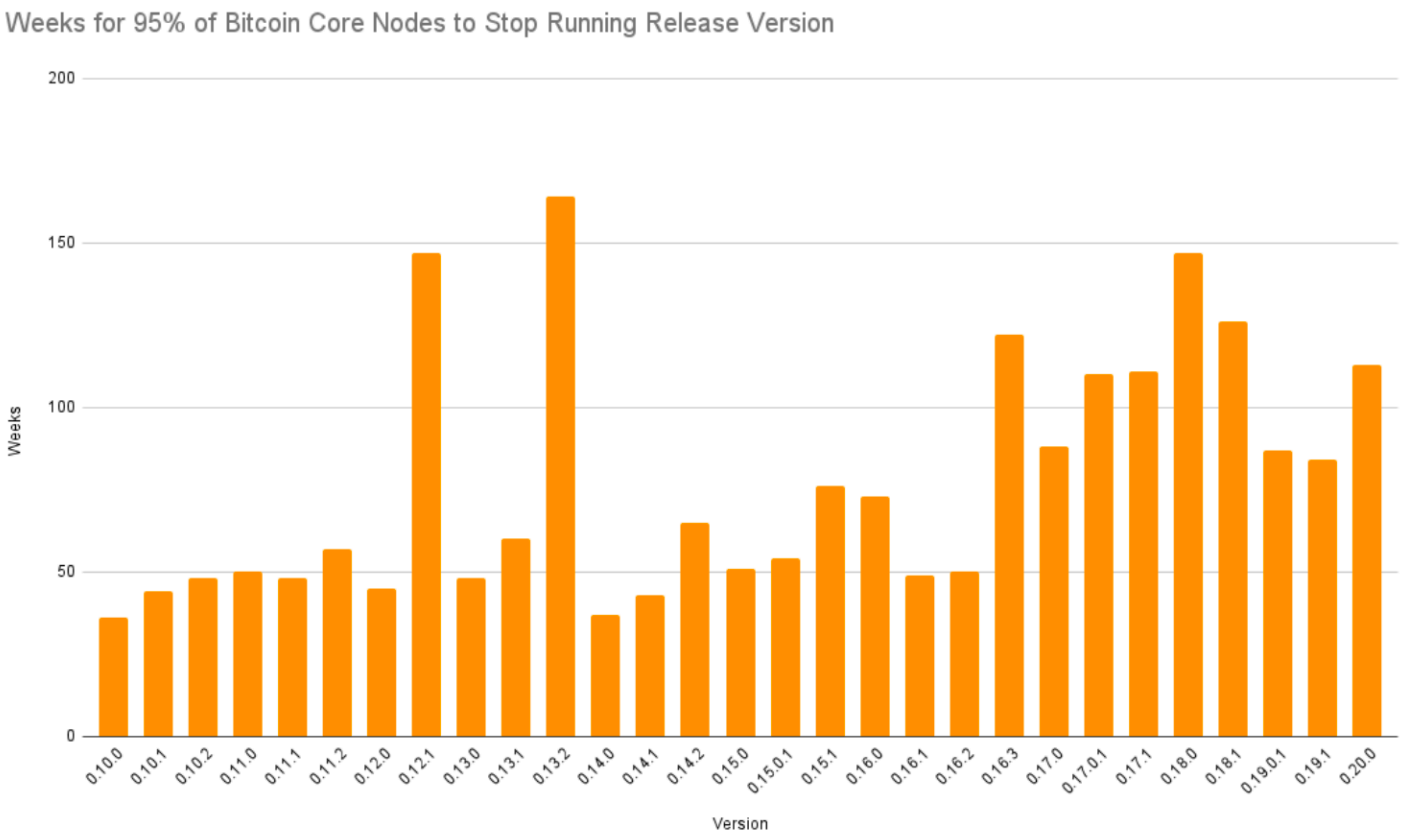


Figure 1: Weeks taken for 95% of Bitcoin Core Nodes to Upgrade

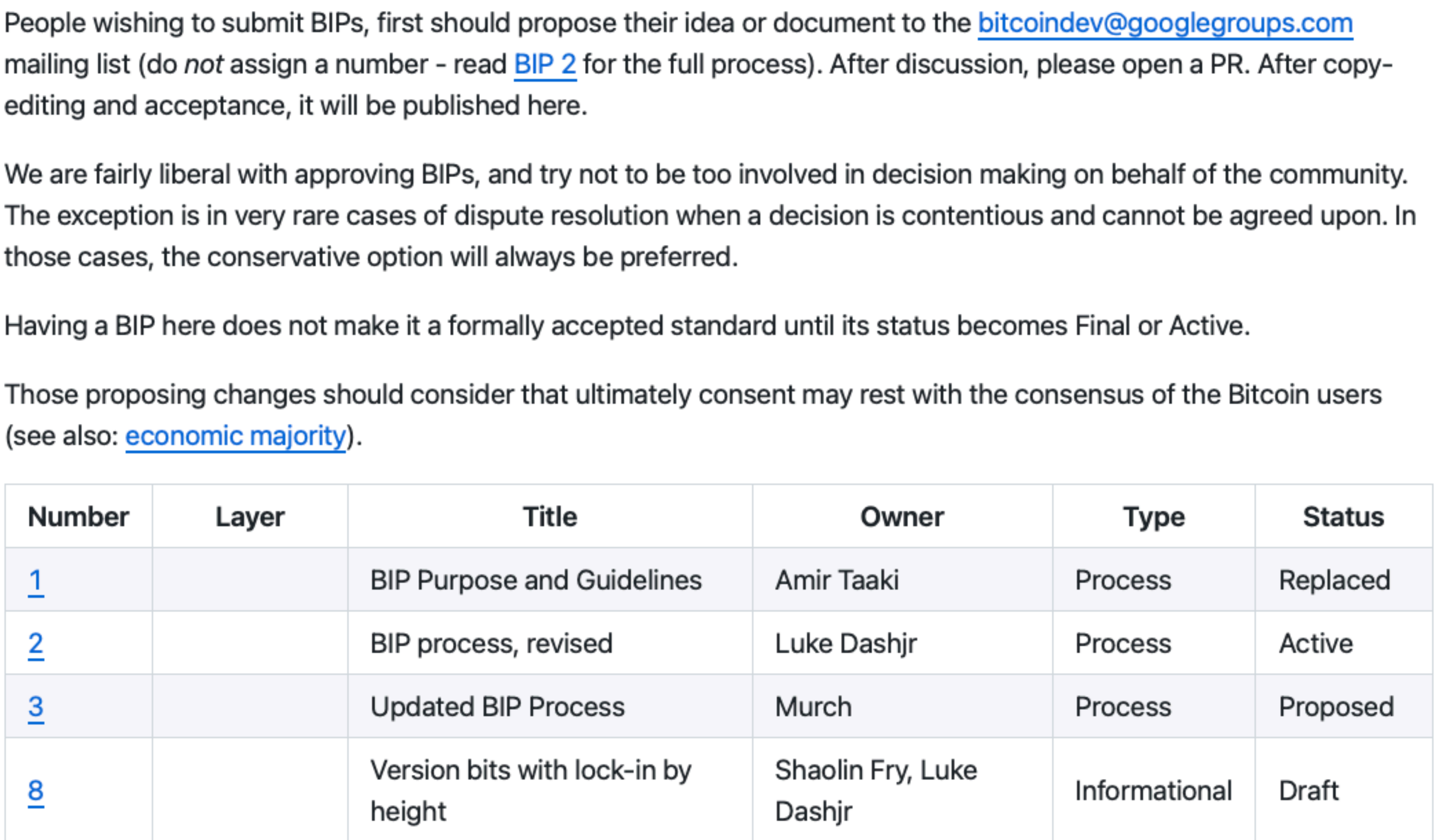
STAKE HOLDERS IN THE SOCIAL CONSENSUS

- Economic Nodes
- Investors
- Media Influencers
- Miners
- Protocol Developers
- Users and Application Developers
- Governments?

To understand this better, check paper:
"Analyzing Bitcoin Consensus: Risks in Protocol Upgrades", Ren Crypto Fish, Steve Lee, Lyn Alden, November 2024, also available at https://github.com/bitcoin-cap/bcap
Linked in Readings of Week 12

THE BITCOIN IMPROVEMENT PROPOSAL PROCESS

See: https://github.com/bitcoin/bips/tree/master



People wishing to submit BIPs, first should propose their idea or document to the bitcoindev@googlegroups.com mailing list (do *not* assign a number - read BIP 2 for the full process). After discussion, please open a PR. After copy-editing and acceptance, it will be published here.

We are fairly liberal with approving BIPs, and try not to be too involved in decision making on behalf of the community. The exception is in very rare cases of dispute resolution when a decision is contentious and cannot be agreed upon. In those cases, the conservative option will always be preferred.

Having a BIP here does not make it a formally accepted standard until its status becomes Final or Active.

Those proposing changes should consider that ultimately consent may rest with the consensus of the Bitcoin users (see also: economic majority).

Number	Layer	Title	Owner	Type	Status
1		BIP Purpose and Guidelines	Amir Taaki	Process	Replaced
2		BIP process, revised	Luke Dashjr	Process	Active
3		Updated BIP Process	Murch	Process	Proposed
8		Version bits with lock-in by height	Shaolin Fry, Luke Dashjr	Informational	Draft

THE BITCOIN IMPROVEMENT PROPOSAL PROCESS

Type of BIPs:

- Process BIP
- Standard BIP:
  Propose useful convention standard that does not involve consensus change
  (e.g.: BIP 39 on mnemonic backup phrase standard)
- Consensus BIP:
  Change of consensus rules, activation by network necessary

Traditional prerequisite for activation:

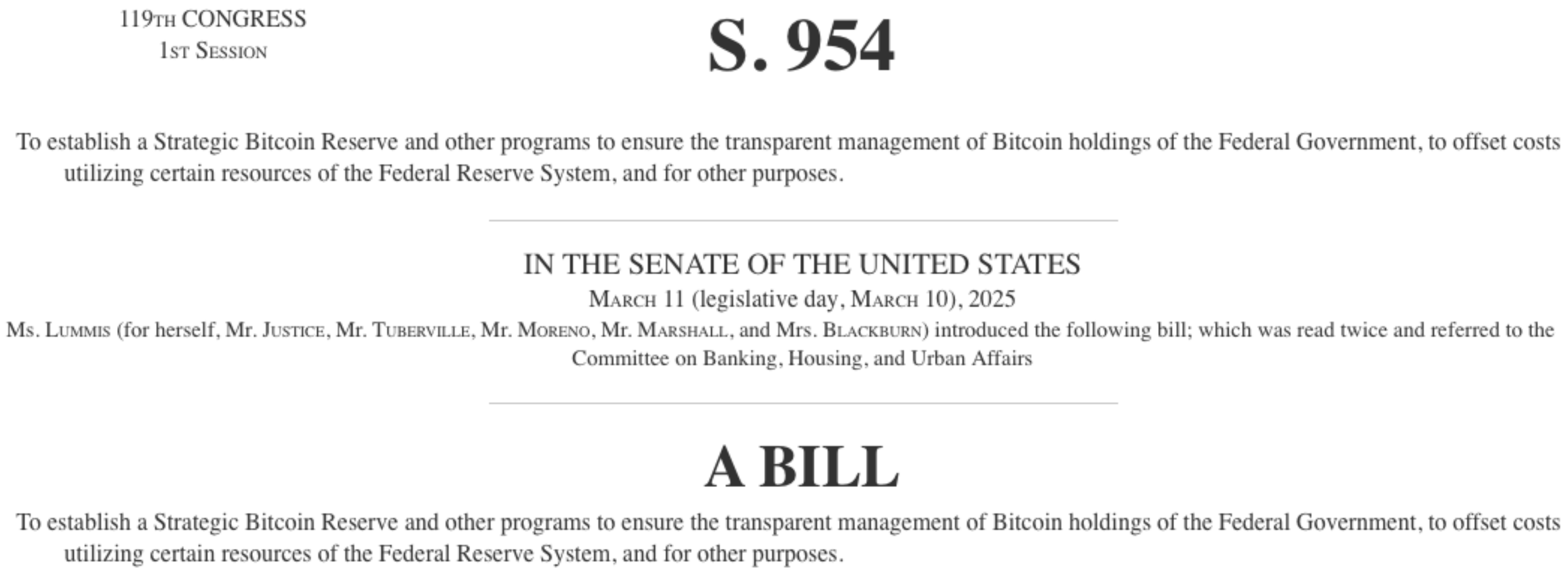
▷ Certain percentage of mined blocks in a time period indicates approval

THE BITCOIN ACT OF 2025: IN U.S. SENATE BANKING COMMITTEE

Proposed by Sen. Cynthia Lummis (R.): "Boosting Innovation, Technology, and Competitivenes through Optimized In Nationwide (BITCOIN) Act of 2025"

Goals:

- **Strategic Bitcoin Reserve**:
  Secured across geographically distributed storage facilities, holding period of 20 years
- **Bitcoin Purchase Program:**
  Purchase 200,000 Bitcoins per year for 5 years, total of 1,000,000 BTC



119th CONGRESS
1st Session

S. 954

To establish a Strategic Bitcoin Reserve and other programs to ensure the transparent management of Bitcoin holdings of the Federal Government, to offset costs utilizing certain resources of the Federal Reserve System, and for other purposes.

IN THE SENATE OF THE UNITED STATES

March 11 (legislative day, March 10), 2025

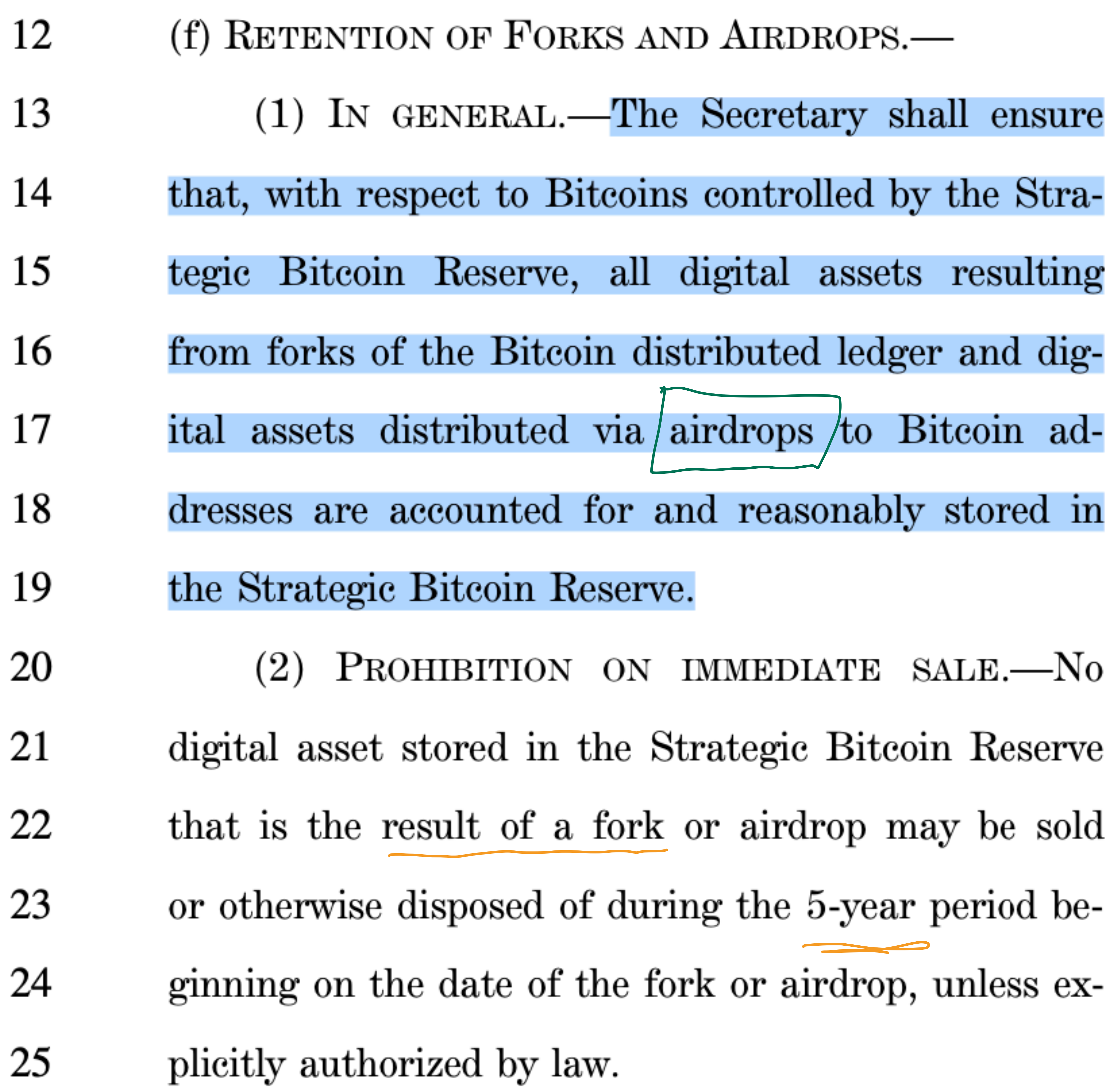
Ms. Lummis (for herself, Mr. Justice, Mr. Tuberville, Mr. Moreno, Mr. Marshall, and Mrs. Blackburn) introduced the following bill; which was read twice and referred to the Committee on Banking, Housing, and Urban Affairs

A BILL

To establish a Strategic Bitcoin Reserve and other programs to ensure the transparent management of Bitcoin holdings of the Federal Government, to offset costs utilizing certain resources of the Federal Reserve System, and for other purposes.

Source: https://www.congress.gov/bill/119th-congress/senate-bill/954/text

STAKE HOLDERS IN THE SOCIAL CONSENSUS



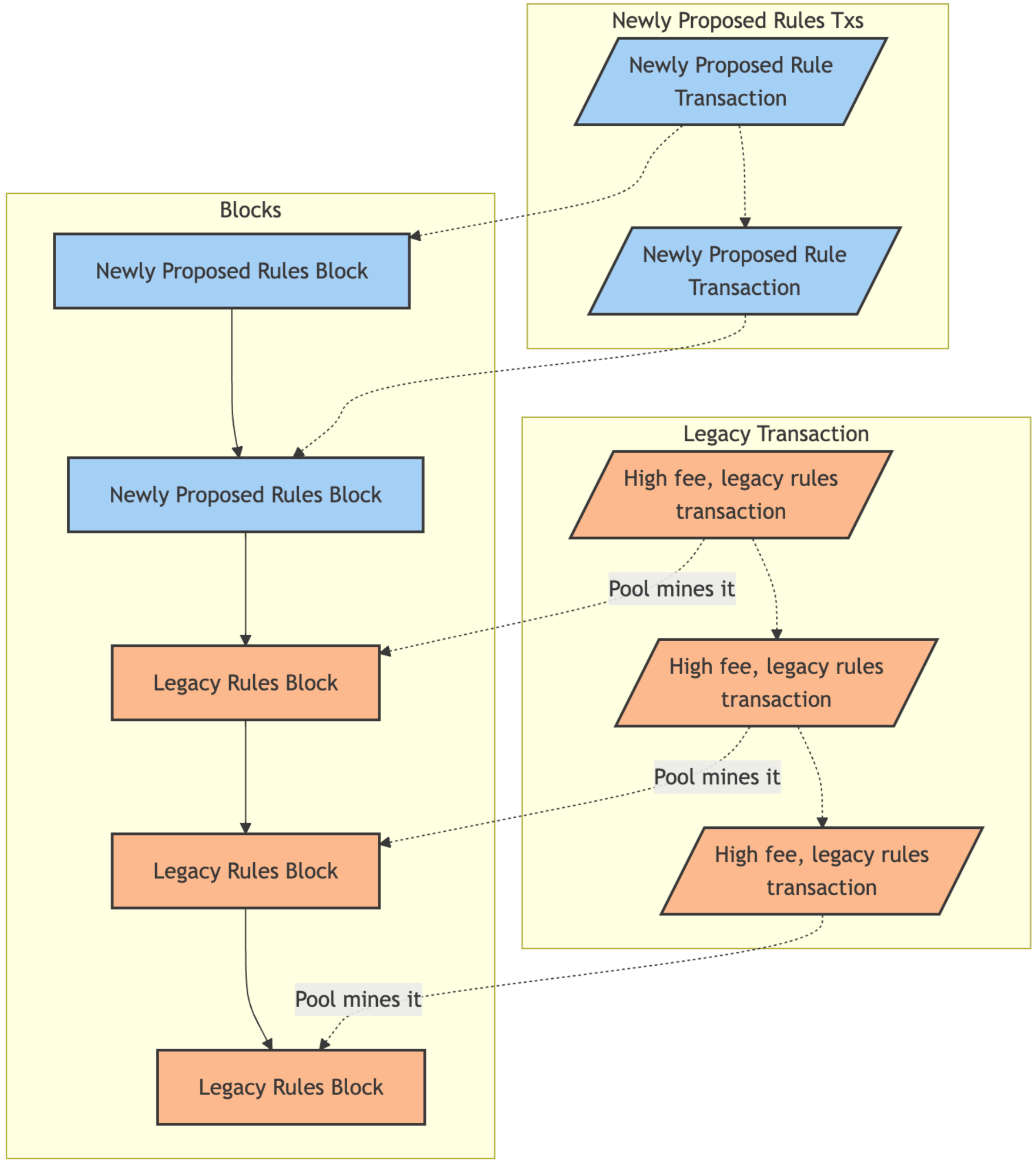
(f) Retention of Forks and Airdrops.—

(1) In general.—The Secretary shall ensure that, with respect to Bitcoins controlled by the Strategic Bitcoin Reserve, all digital assets resulting from forks of the Bitcoin distributed ledger and digital assets distributed via airdrops to Bitcoin addresses are accounted for and reasonably stored in the Strategic Bitcoin Reserve.

(2) Prohibition on immediate sale.—No digital asset stored in the Strategic Bitcoin Reserve that is the result of a fork or airdrop may be sold or otherwise disposed of during the 5-year period beginning on the date of the fork or airdrop, unless explicitly authorized by law.

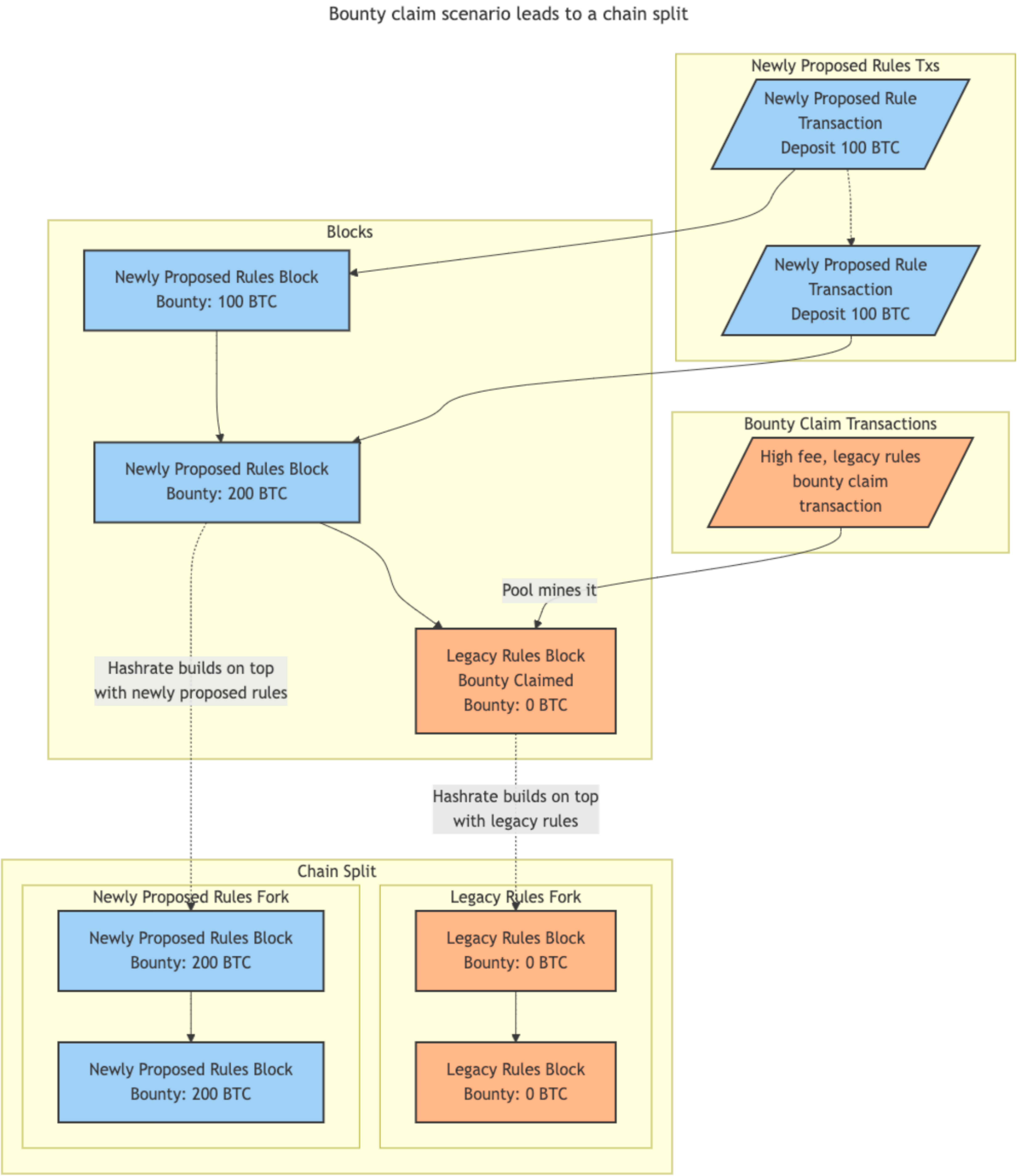
Scenario: Partially adopted Soft Fork

Legacy rules blocks can be built on top of newly proposed rules blocks


Newly Proposed Rules Txs
Newly Proposed Rule Transaction
Newly Proposed Rule Transaction
Blocks
Newly Proposed Rules Block
Newly Proposed Rules Block
Legacy Rules Block
Legacy Rules Block
Legacy Rules Block
Legacy Transaction
High fee, legacy rules transaction
Pool mines it
High fee, legacy rules transaction
Pool mines it
High fee, legacy rules transaction
Pool mines it

Scenario: Partially adopted Soft Fork

"Bounty Claim": Assets locked by new type of scripts that are freely spendable in old rule set.



Bounty claim scenario leads to a chain split
Newly Proposed Rules Txs
Newly Proposed Rule Transaction Deposit 100 BTC
Newly Proposed Rule Transaction Deposit 100 BTC
Blocks
Newly Proposed Rules Block Bounty: 100 BTC
Newly Proposed Rules Block Bounty: 200 BTC
Bounty Claim Transactions
High fee, legacy rules bounty claim transaction
Pool mines it
Legacy Rules Block Bounty Claimed Bounty: 0 BTC
Hashrate builds on top with newly proposed rules
Hashrate builds on top with legacy rules
Chain Split
Newly Proposed Rules Fork
Newly Proposed Rules Block Bounty: 200 BTC
Newly Proposed Rules Block Bounty: 200 BTC
Legacy Rules Fork
Legacy Rules Block Bounty: 0 BTC
Legacy Rules Block Bounty: 0 BTC