BANCA D'ITALIA
EUROSISTEMA

ART
Applied Research Team

Certified Byzantine Consensus with Confidential Quorum for a Bitcoin-derived Permissioned DLT

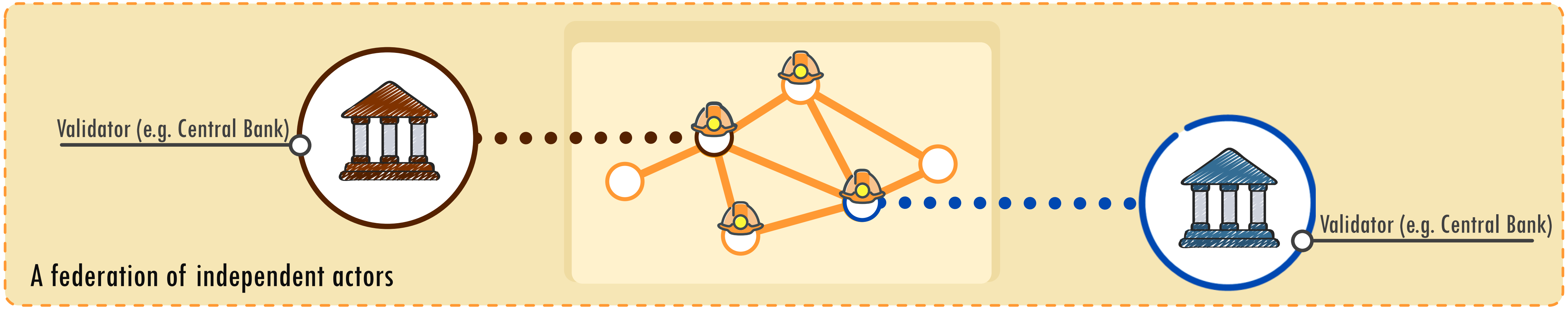
M. Benedetti, F. De Sclavis, M. Favorito, **G. Galano**, S. Giammusso, A. Muci, M. Nardelli

Applied Research Team (ART) - IT Department - Bank of Italy*

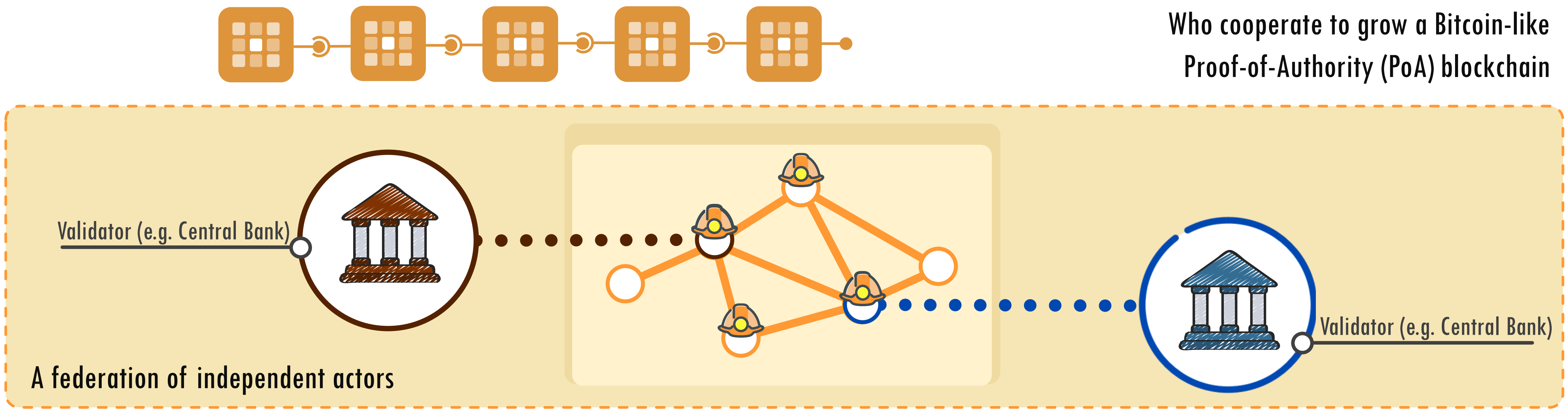
DLT23: 5th Distributed Ledger Technology Workshop, May 25-26, 2023, Bologna, Italy

*The views expressed in this presentation are those of the authors and do not necessarily reflect those of the Bank of Italy

Introduction

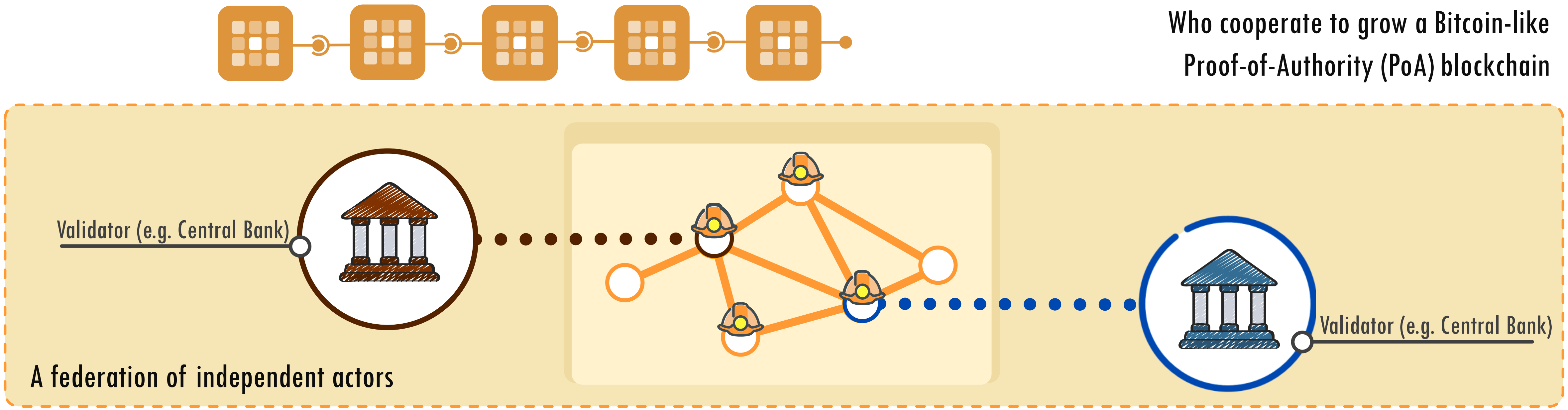

Validator (e.g. Central Bank)
Validator (e.g. Central Bank)
A federation of independent actors

Introduction

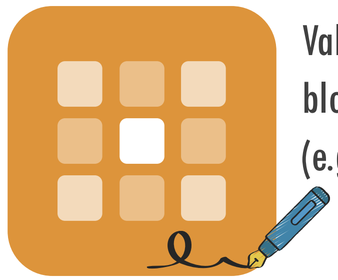

Who cooperate to grow a Bitcoin-like
Proof-of-Authority (PoA) blockchain
Validator (e.g. Central Bank)
Validator (e.g. Central Bank)
A federation of independent actors

Introduction

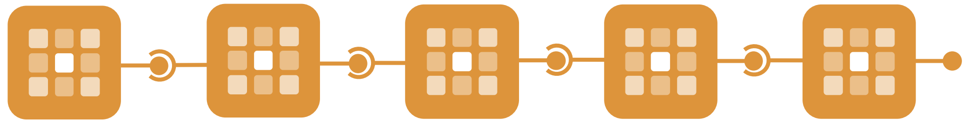
Proof-of-work is **disabled** (50% success probability). Blocks are valid iff they include a **solution** to a specific block **challenge**. A challenge can be a set of 4 public keys out of which 3 signatures are required (3-of-4 OP_CHECKMULTISIG), or a Taproot tweaked public key. A solution is the corresponding witness.


Who cooperate to grow a Bitcoin-like Proof-of-Authority (PoA) blockchain
Validator (e.g. Central Bank)
Validator (e.g. Central Bank)
A federation of independent actors

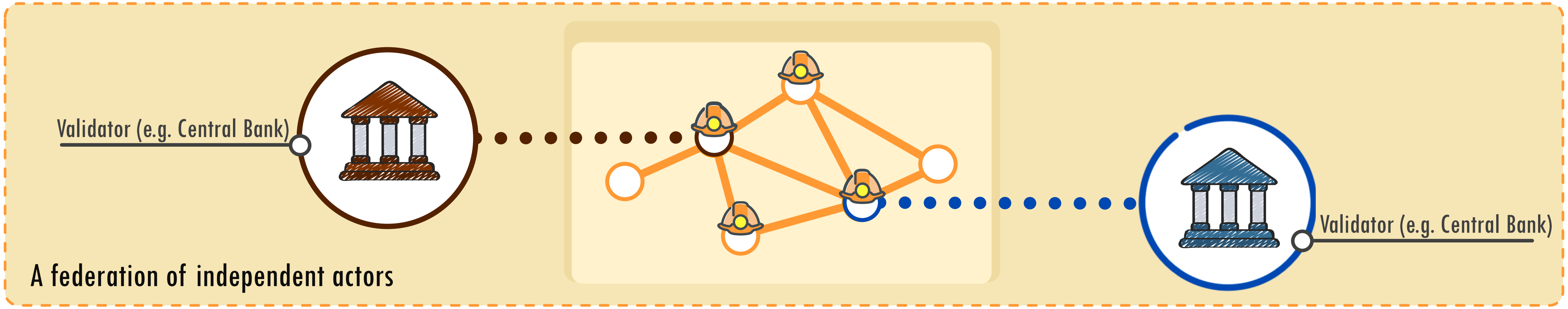
Introduction



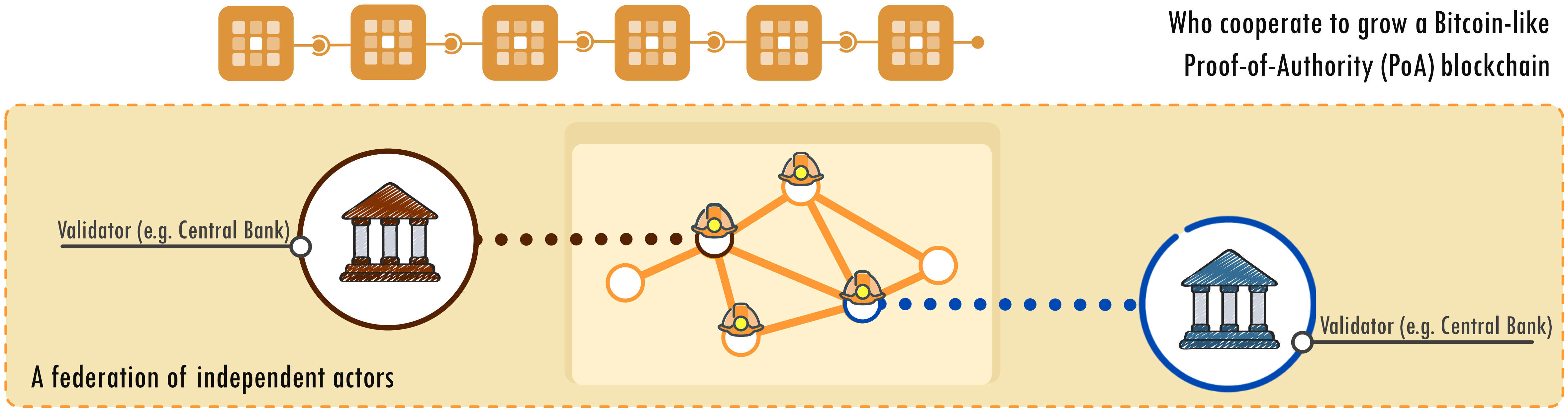
Validators jointly create blocks by adding the solution (e.g., signatures)



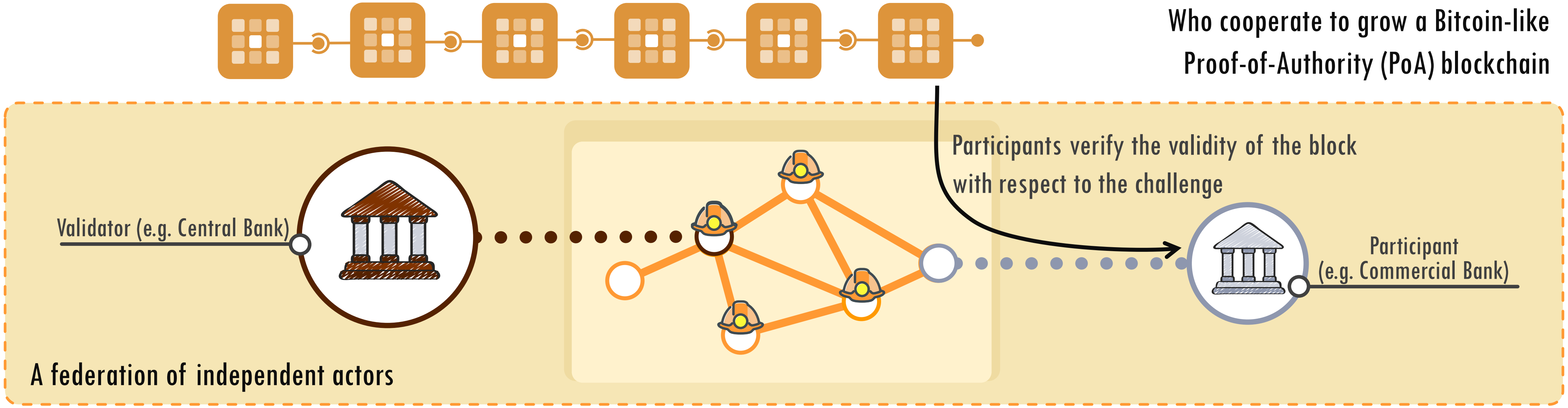
Who cooperate to grow a Bitcoin-like Proof-of-Authority (PoA) blockchain


Validator (e.g. Central Bank)
Validator (e.g. Central Bank)
A federation of independent actors

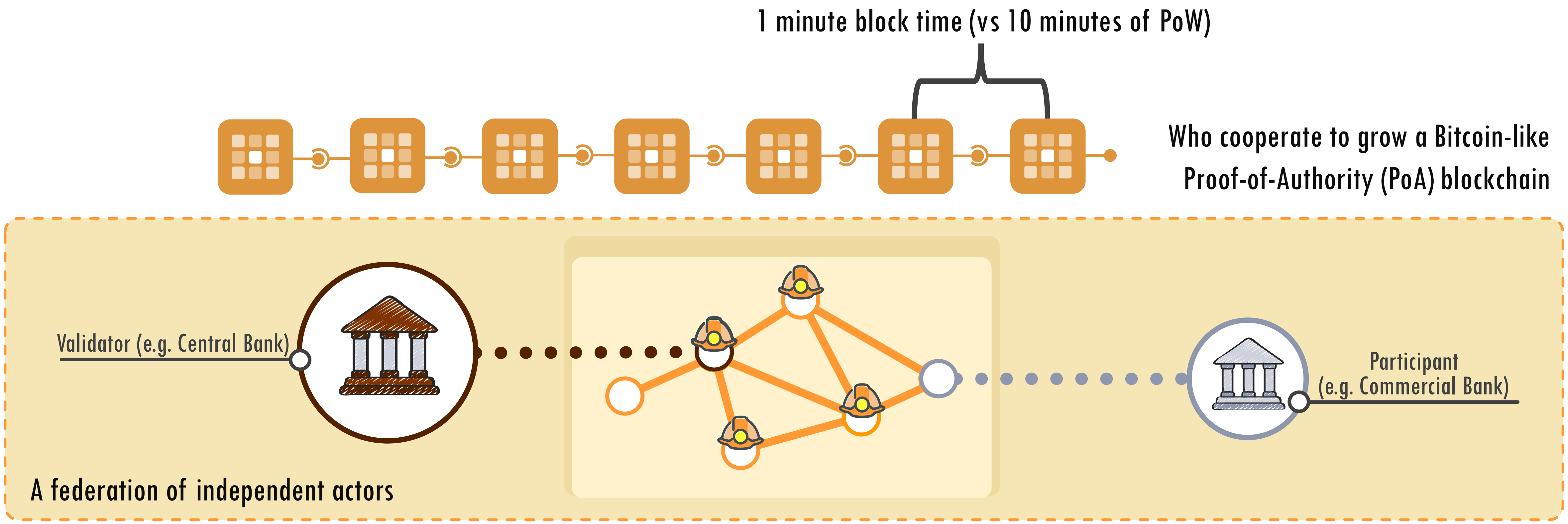
Introduction


Who cooperate to grow a Bitcoin-like
Proof-of-Authority (PoA) blockchain
Validator (e.g. Central Bank)
Validator (e.g. Central Bank)
A federation of independent actors

Introduction

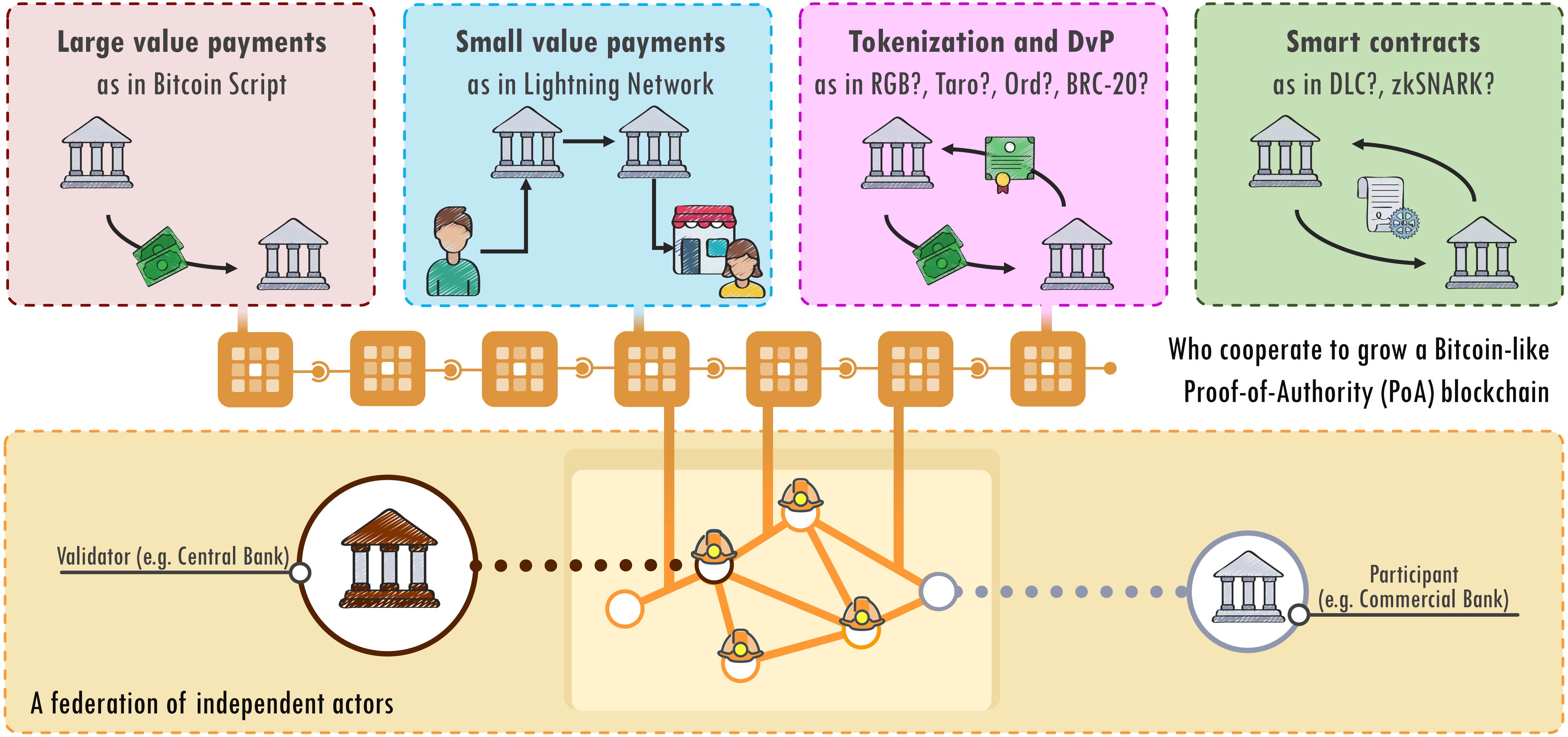

Who cooperate to grow a Bitcoin-like Proof-of-Authority (PoA) blockchain
Participants verify the validity of the block with respect to the challenge
Validator (e.g. Central Bank)
Participant (e.g. Commercial Bank)
A federation of independent actors

Introduction

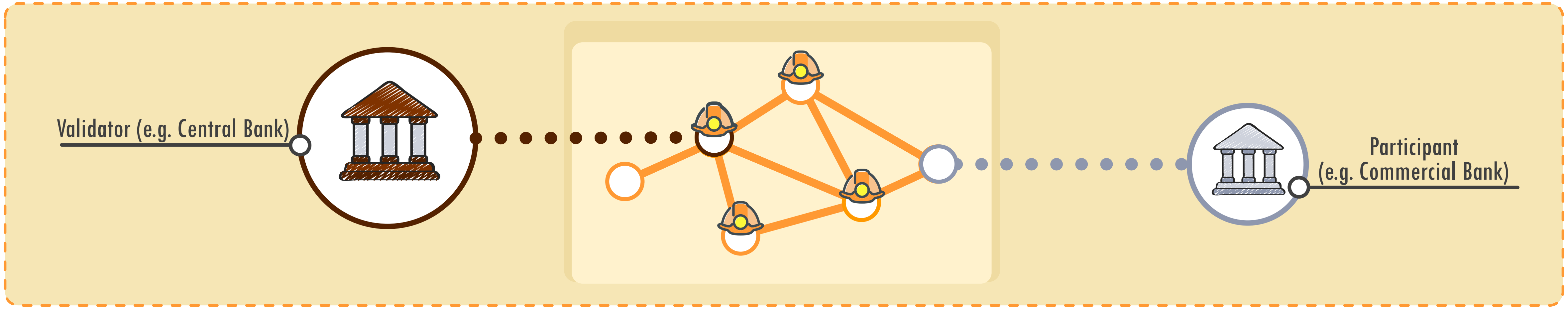

1 minute block time (vs 10 minutes of PoW)
Who cooperate to grow a Bitcoin-like
Proof-of-Authority (PoA) blockchain
Validator (e.g. Central Bank)
Participant
(e.g. Commercial Bank)
A federation of independent actors

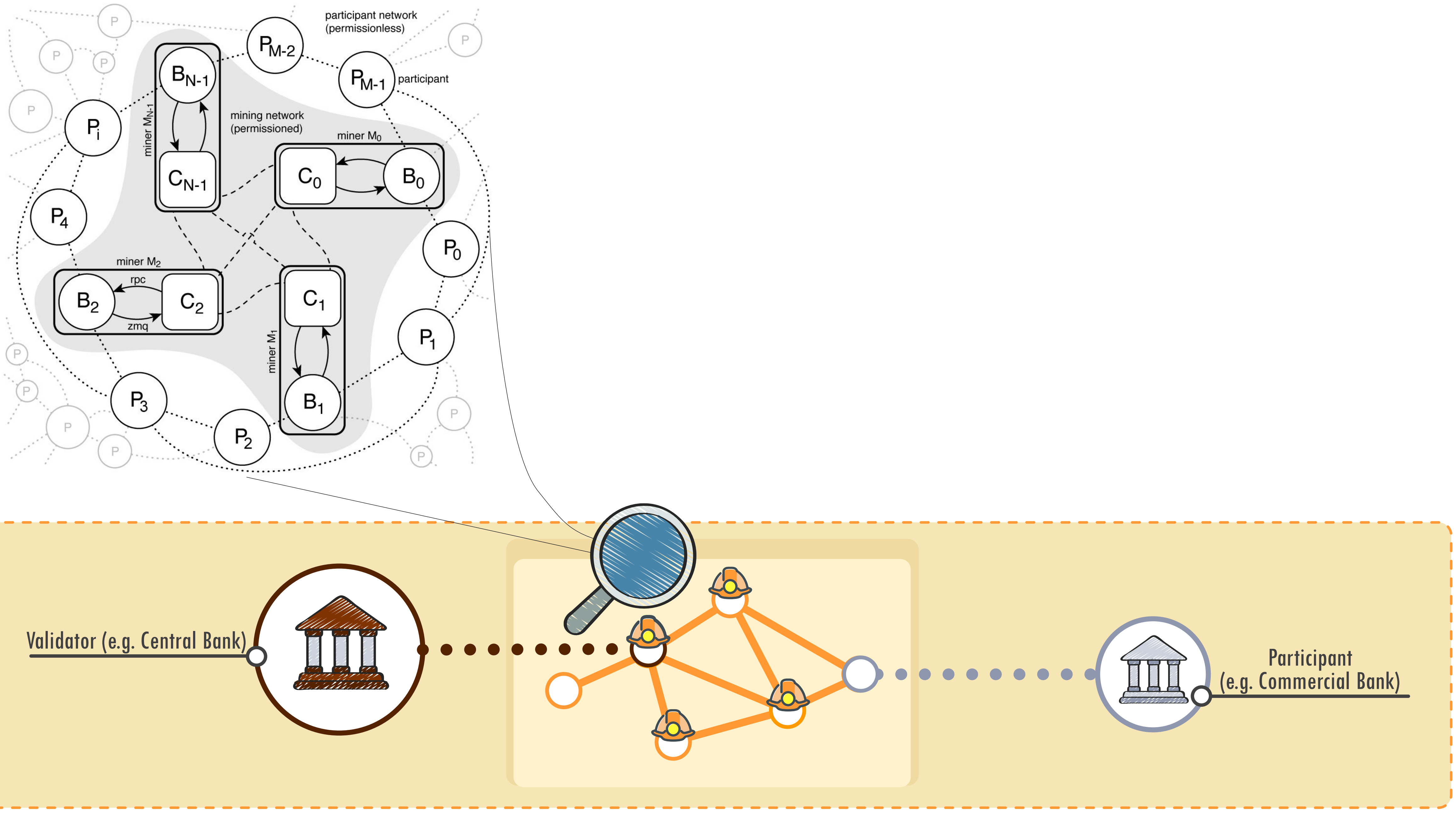
Introduction

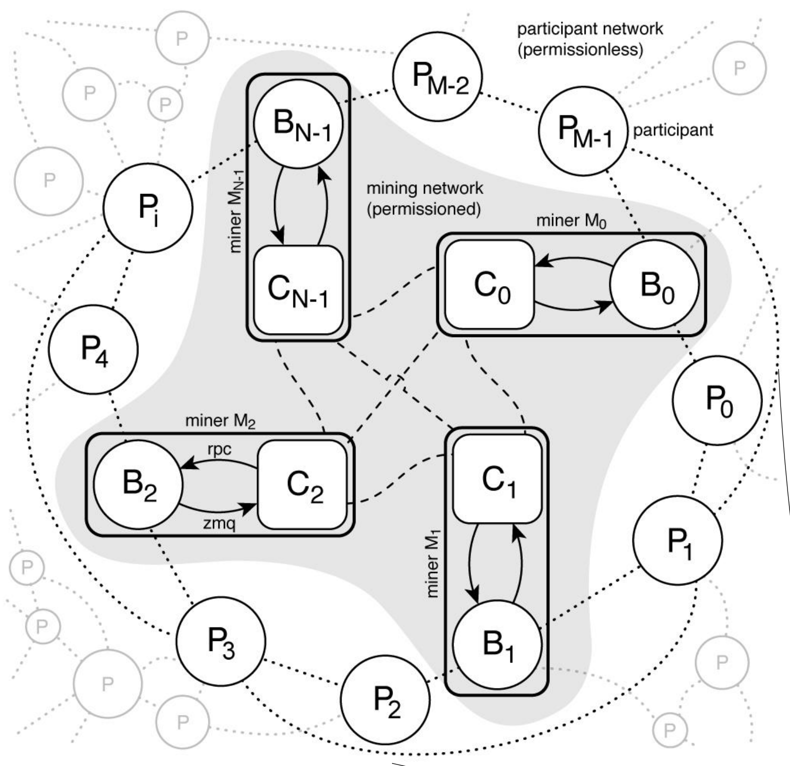
Enables the **reuse** of an existing and powerful technological stack in a permissioned setting



Who cooperate to grow a Bitcoin-like Proof-of-Authority (PoA) blockchain


Validator (e.g. Central Bank)
Participant
(e.g. Commercial Bank)


participant network
(permissionless)
participant
mining network
(permissioned)
miner M_0
miner M_{N-1}
miner M_1
miner M_2
rpc
zmq
B_{N-1}
C_{N-1}
C_0
B_0
C_1
B_1
B_2
C_2
P_{M-2}
P_{M-1}
P_i
P_4
P_0
P_1
P_3
P_2
P
Validator (e.g. Central Bank)
Participant
(e.g. Commercial Bank)



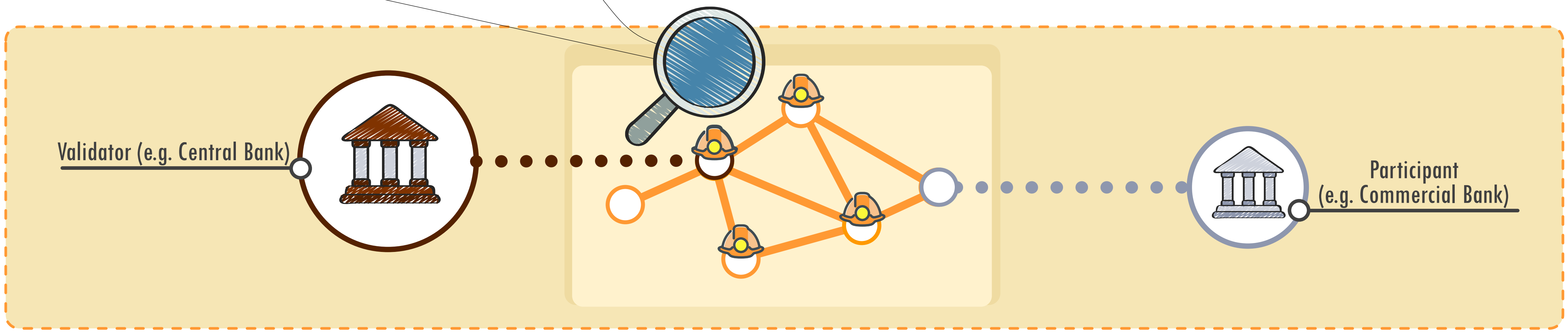
System model

Participants network with ~1000s of nodes, connected in a spontaneous and not predefined topology

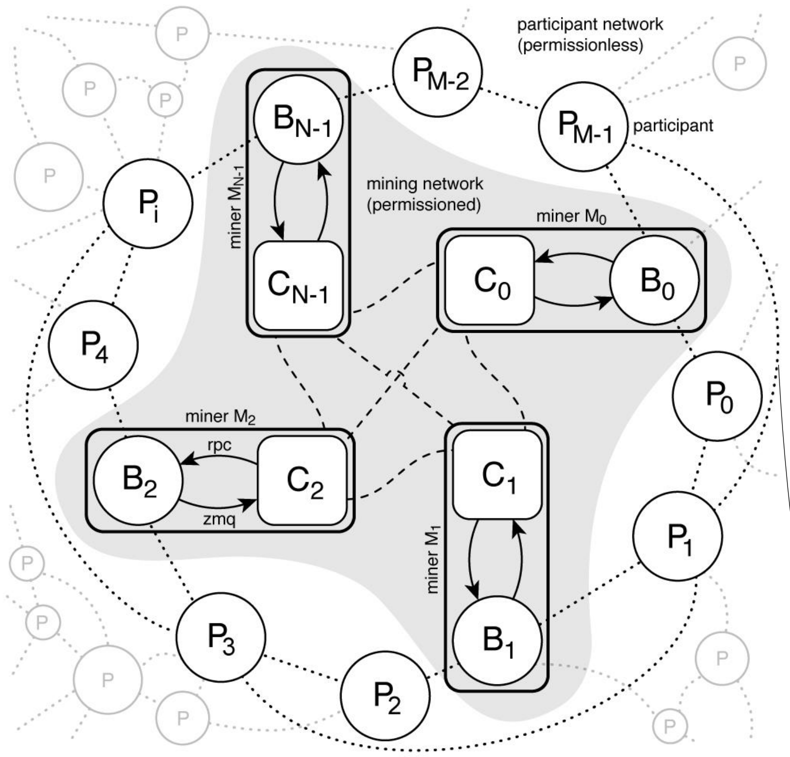
Permissioned "mining" network with N ∈ [4, ~20] nodes, geographically distributed and connected in a full-mesh topology.
Mining Node = Bridge Node + Consensus Node

Up to F mining nodes **can fail arbitrarily** and N>3F

Network is **weakly synchronous**



Requirements



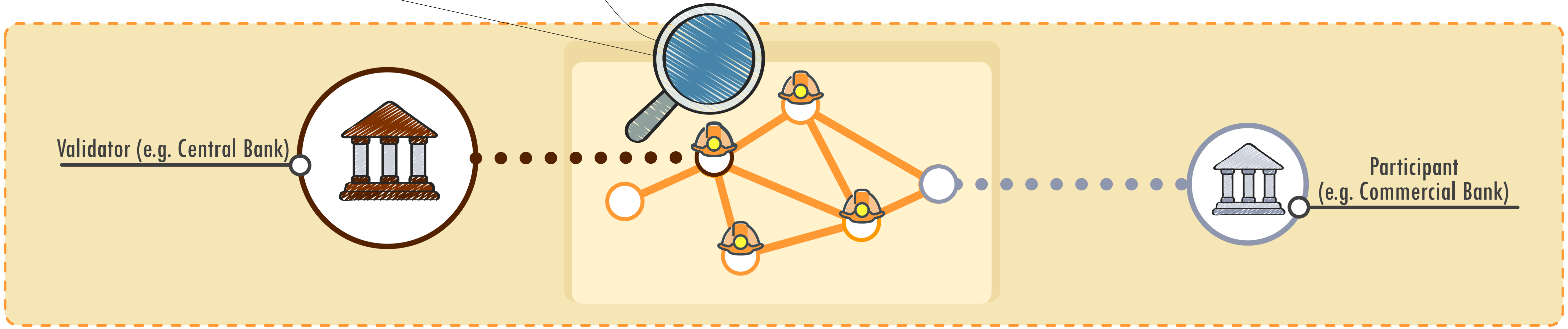
Correctness: transactions should be valid and properly authorized

Safety: finality of transactions should be deterministic (no forks)

Liveness: transactions should be validated also when F nodes fail

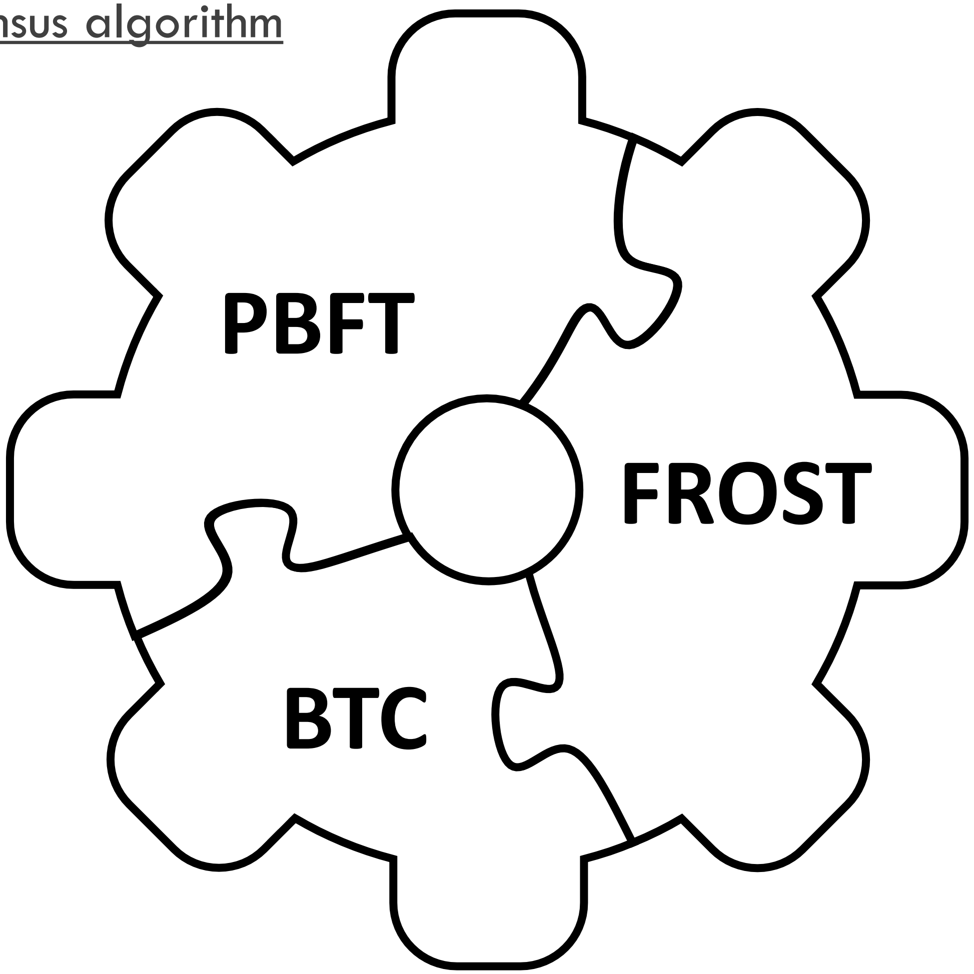
Calmness: blockchain should not grow too fast

Confidentiality (with no faulty miners): the block solution should not reveal information to participants about the miners configuration and quorum



Contribution: a combination of 3

1. Byzantine Fault Tolerant consensus algorithm



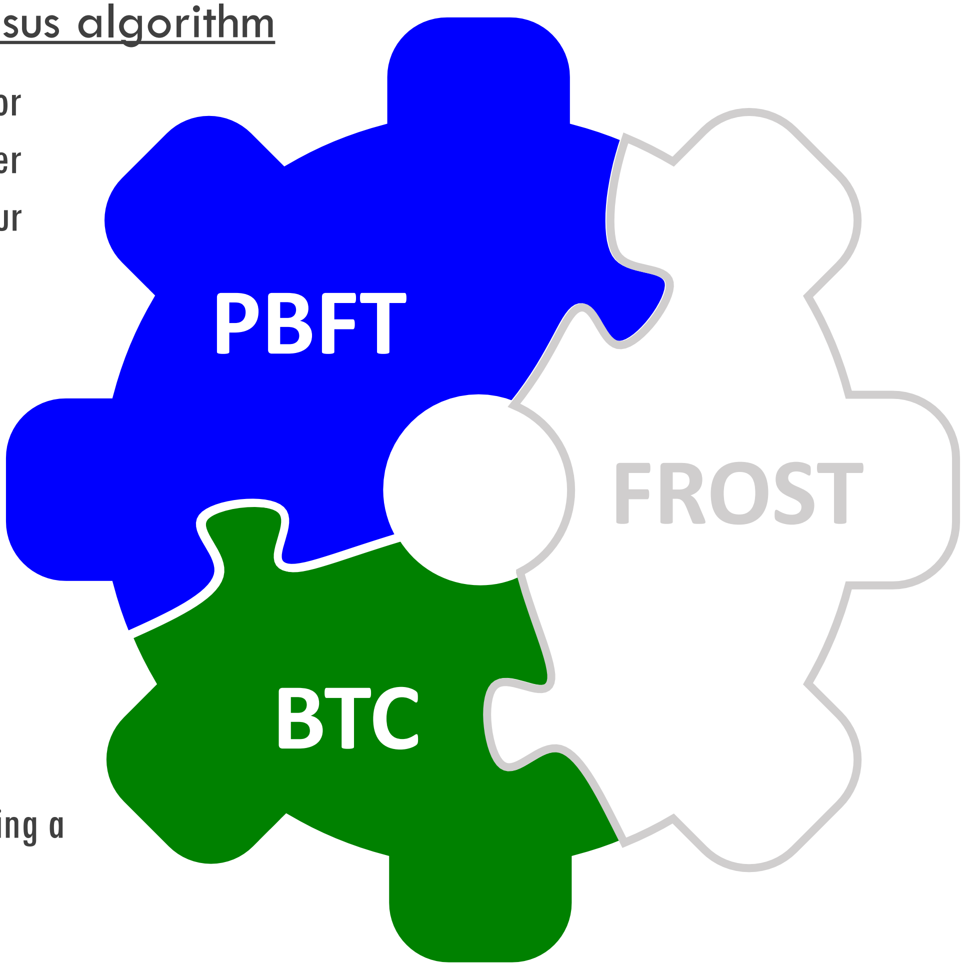
3. Threshold signature scheme

2. Blockchain platform

Contribution: a combination of 3

1. Byzantine Fault Tolerant consensus algorithm

Used to agree on the next block. PBFT is chosen for its "simple" implementation compared to newer and more scalable alternatives, because of our limited interest in on-ledger scalability.



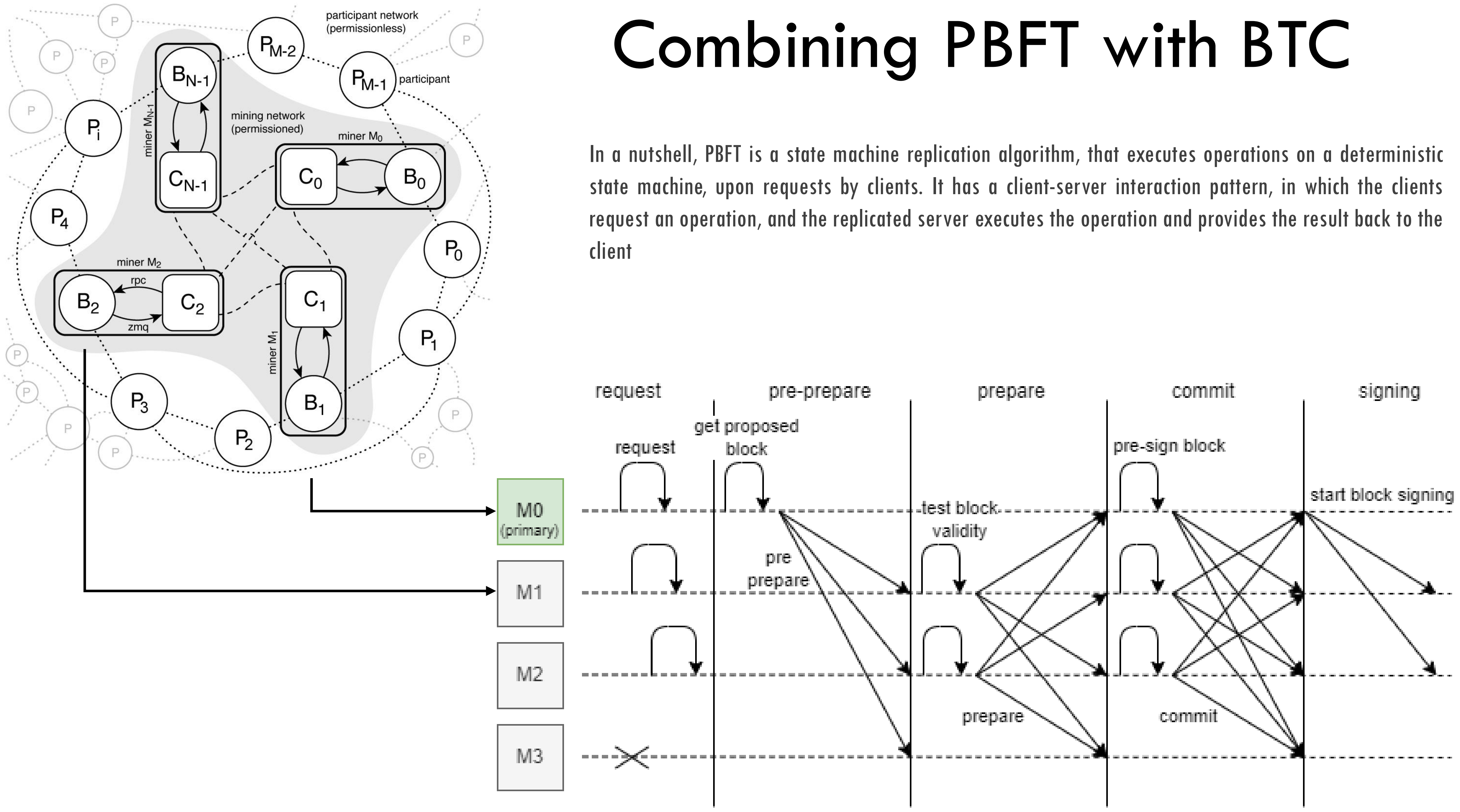
3. Threshold signature scheme

2. Blockchain platform

Used as the architectural solution for maintaining a shared ledger among participants.

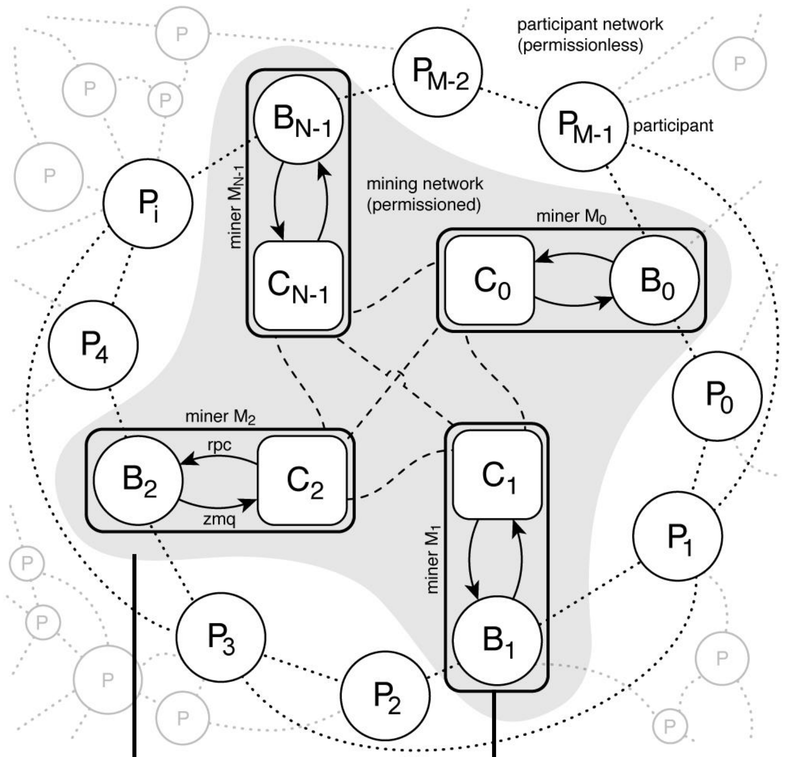
Combining PBFT with BTC

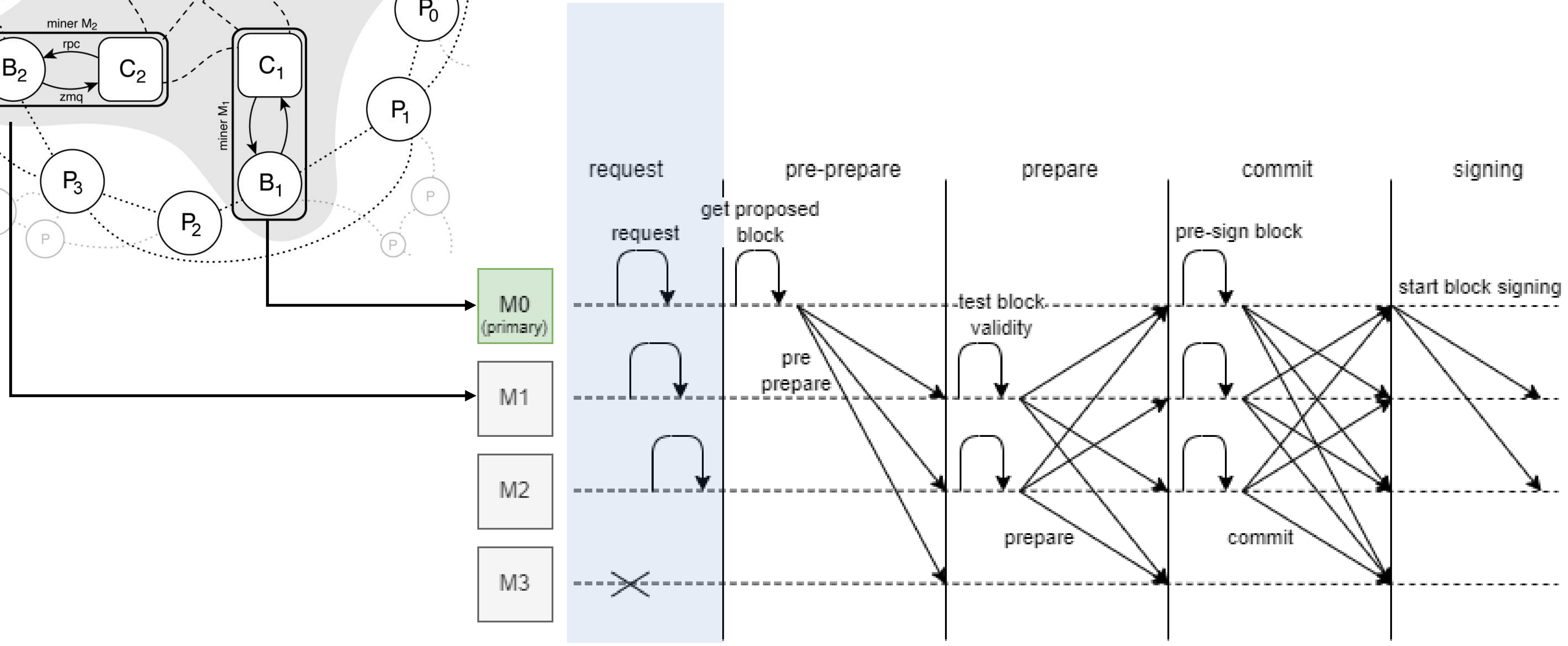
In a nutshell, PBFT is a state machine replication algorithm, that executes operations on a deterministic state machine, upon requests by clients. It has a client-server interaction pattern, in which the clients request an operation, and the replicated server executes the operation and provides the result back to the client



Combining PBFT with BTC

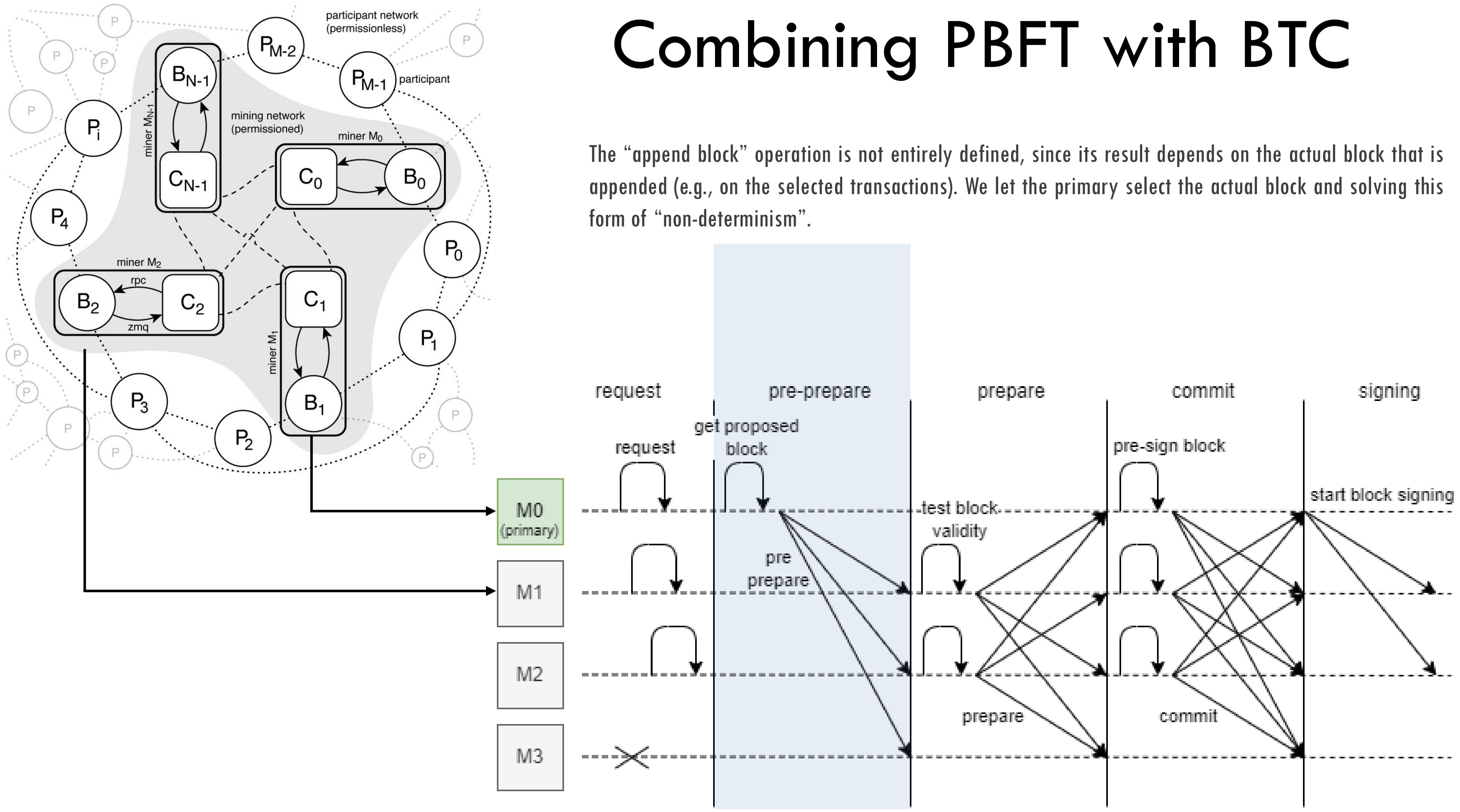
Our replicated state machine has one operation only, i.e., “append block”, and only one abstract client, i.e., the participants network, which invokes the operation every target block time. Requests can be **self-generated by replica**, avoiding trusted clients which would represent a single point of failure.





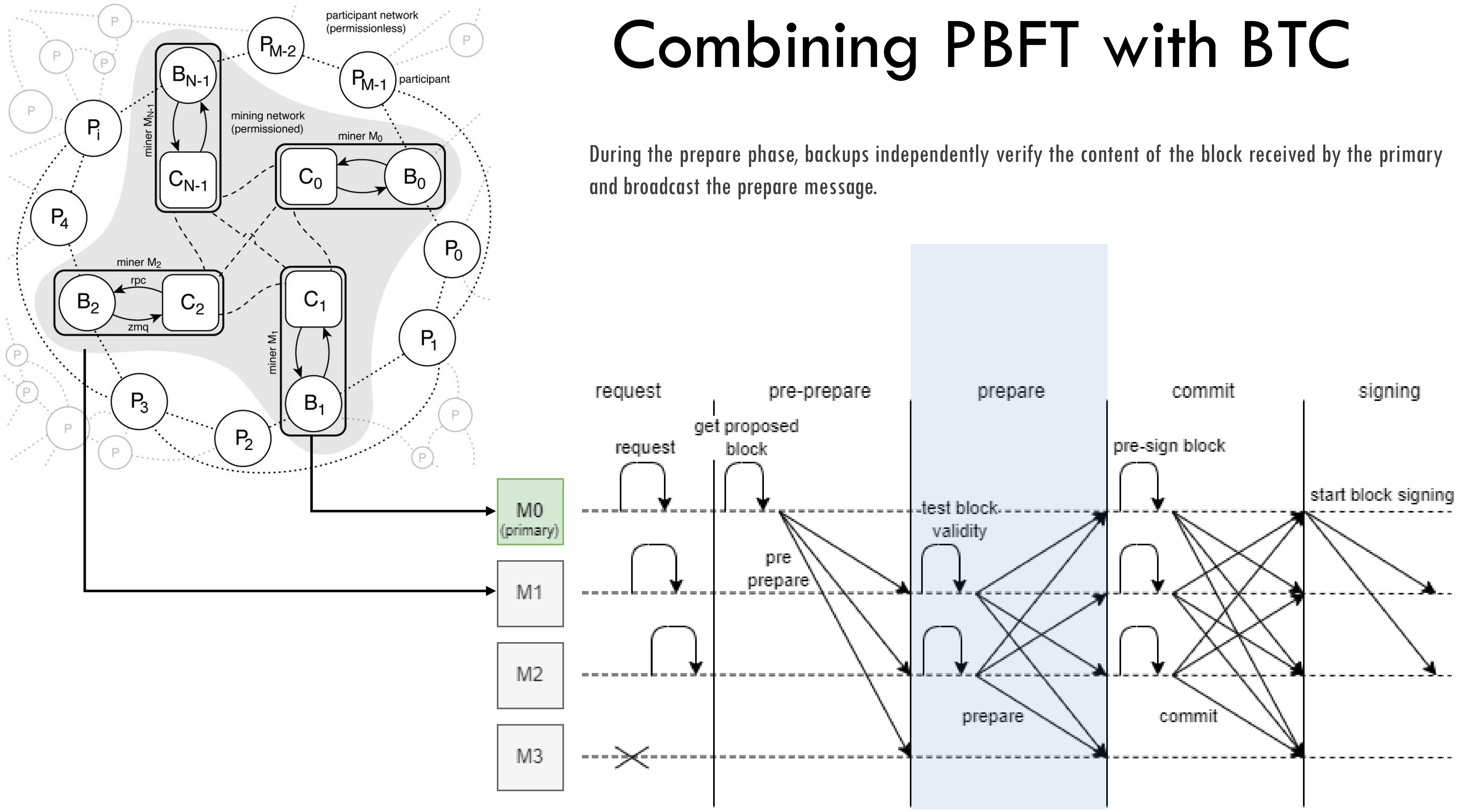
Combining PBFT with BTC

The "append block" operation is not entirely defined, since its result depends on the actual block that is appended (e.g., on the selected transactions). We let the primary select the actual block and solving this form of "non-determinism".



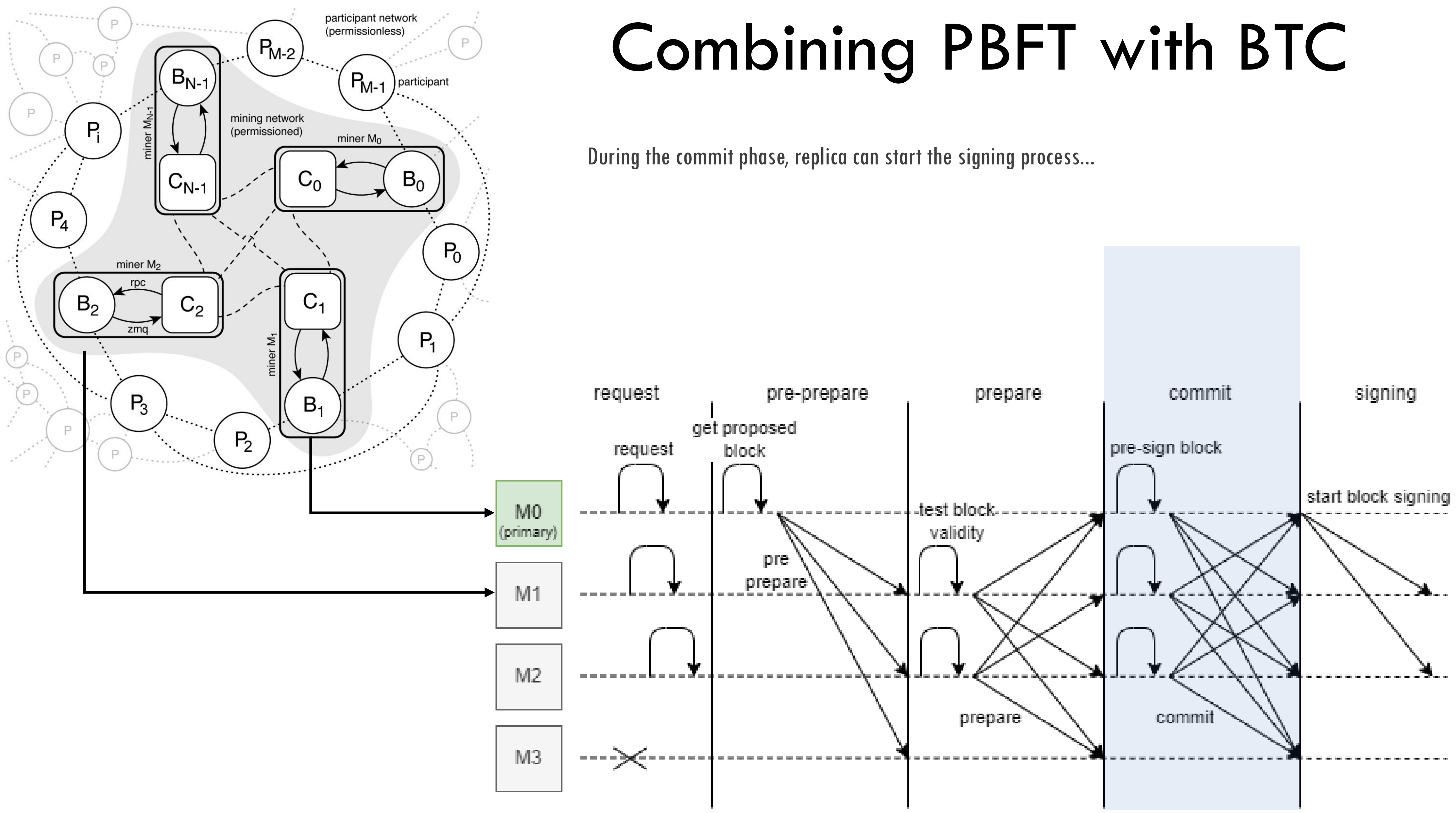
Combining PBFT with BTC

During the prepare phase, backups independently verify the content of the block received by the primary and broadcast the prepare message.



Combining PBFT with BTC

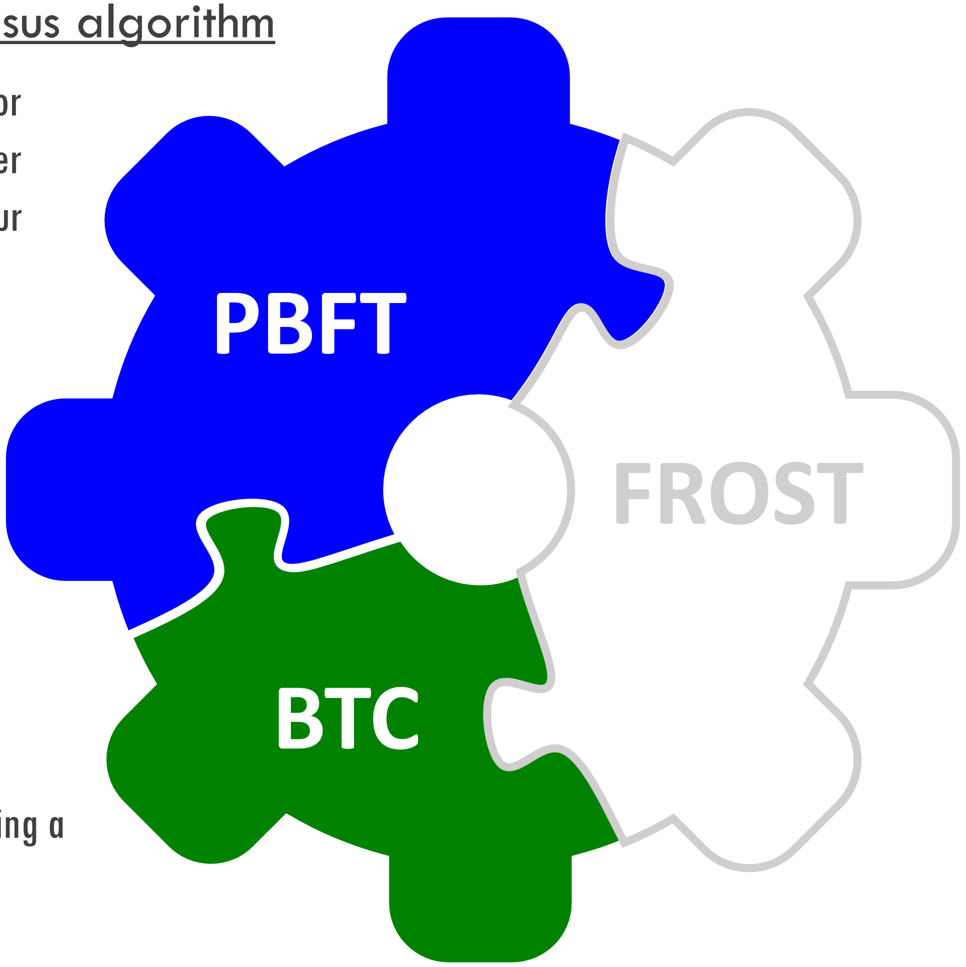
During the commit phase, replica can start the signing process…



Contribution: a combination of 3

1. Byzantine Fault Tolerant consensus algorithm

Used to agree on the next block. PBFT is chosen for its "simple" implementation compared to newer and more scalable alternatives, because of our limited interest in on-ledger scalability.



3. Threshold signature scheme

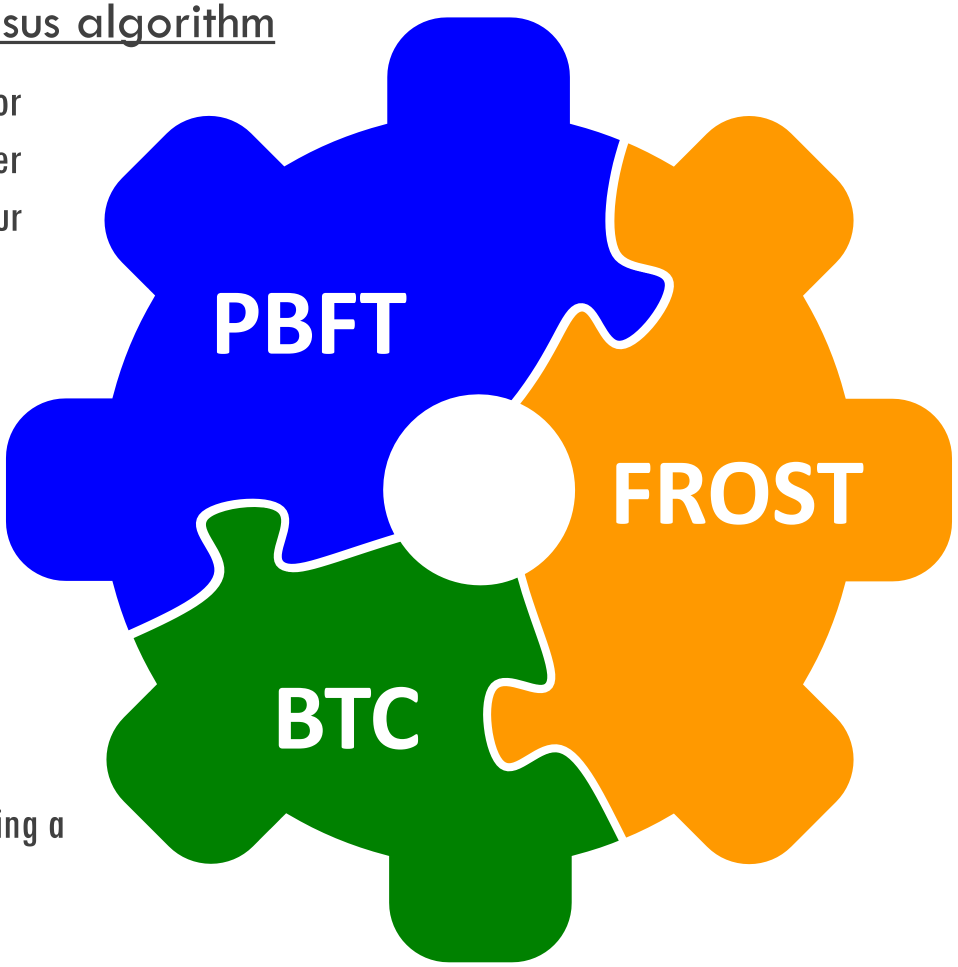
2. Blockchain platform

Used as the architectural solution for maintaining a shared ledger among participants.

Contribution: a combination of 3

1. Byzantine Fault Tolerant consensus algorithm

Used to agree on the next block. PBFT is chosen for its "simple" implementation compared to newer and more scalable alternatives, because of our limited interest in on-ledger scalability.



2. Blockchain platform

Used as the architectural solution for maintaining a shared ledger among participants.

3. Threshold signature scheme

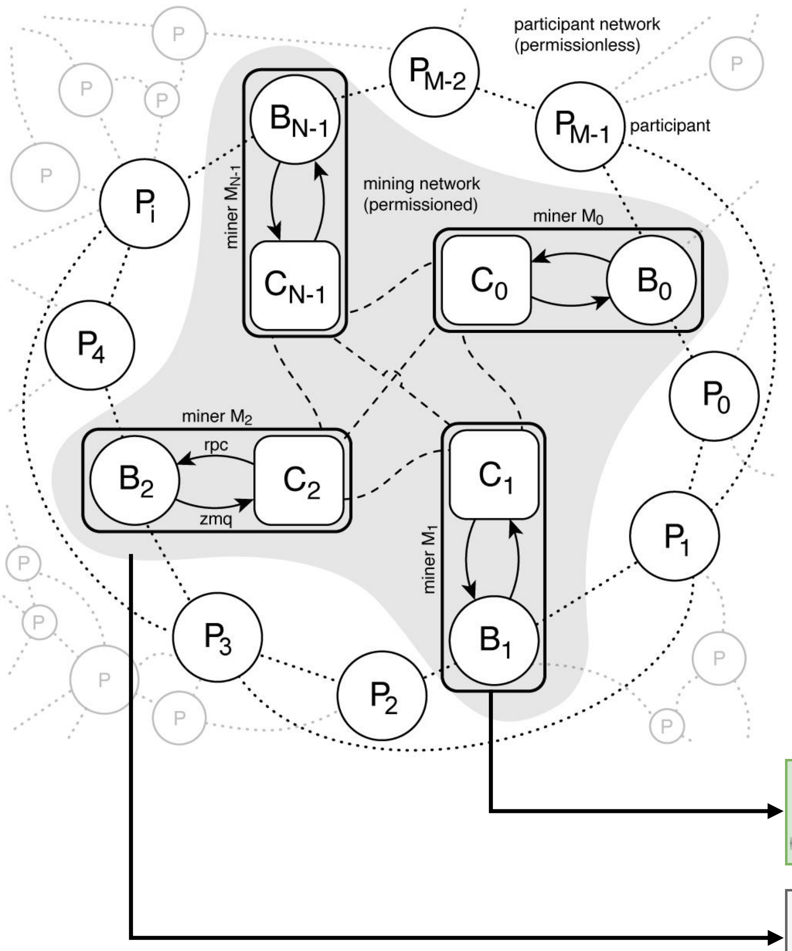
Used to aggregate a quorum of signature shares and produce a single signature that is used as a block solution. FROST* is chosen because it works with Schnorr signatures that have been recently introduced in our blockchain platform via Taproot

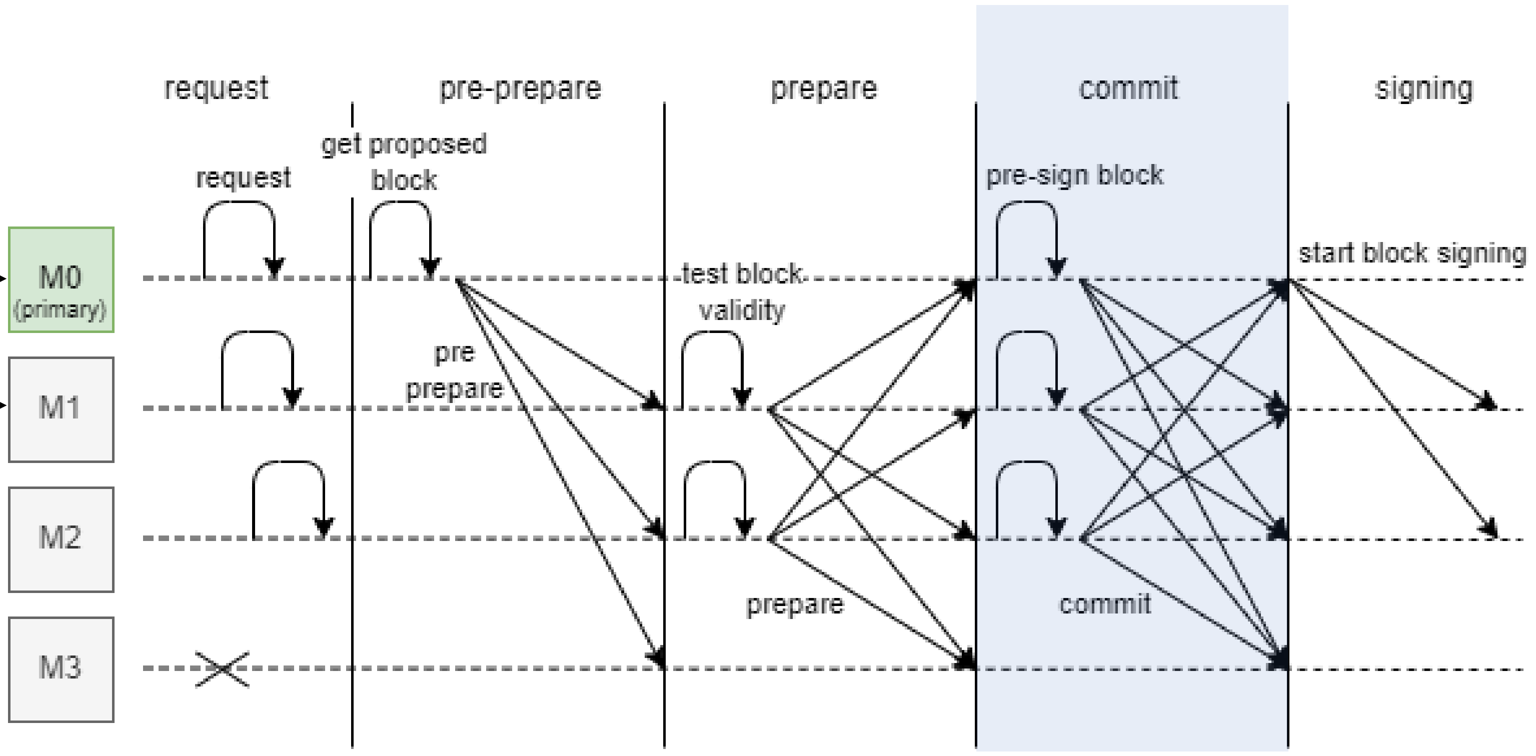
*Komlo, Chelsea, and Ian Goldberg. "FROST: flexible round-optimized Schnorr threshold signatures." *Selected Areas in Cryptography: 27th International Conference, Halifax, NS, Canada (Virtual Event), October 21-23, 2020, Revised Selected Papers 27*. Springer International Publishing, 2021.

Combining PBFT with FROST

During the commit phase, the block is already agreed and could already be signed by replicas. Indeed, a naïve PBFT implementation based on the concatenation of signatures can mingle commit and signing: each signature can be piggybacked to the commit message, and the first node gathering a Byzantine Quorum (2F+1) of signatures can assemble and broadcast a valid block.

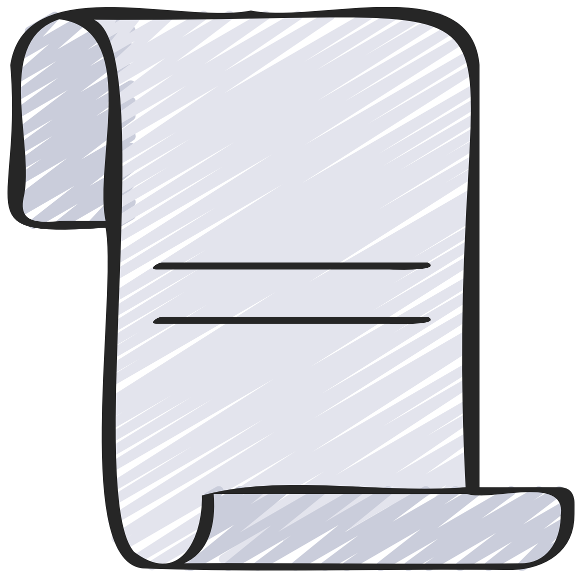
Nevertheless, when we move from signatures to signature shares to aggregate with FROST, an issue arises… The problem is known as **Frostland**.

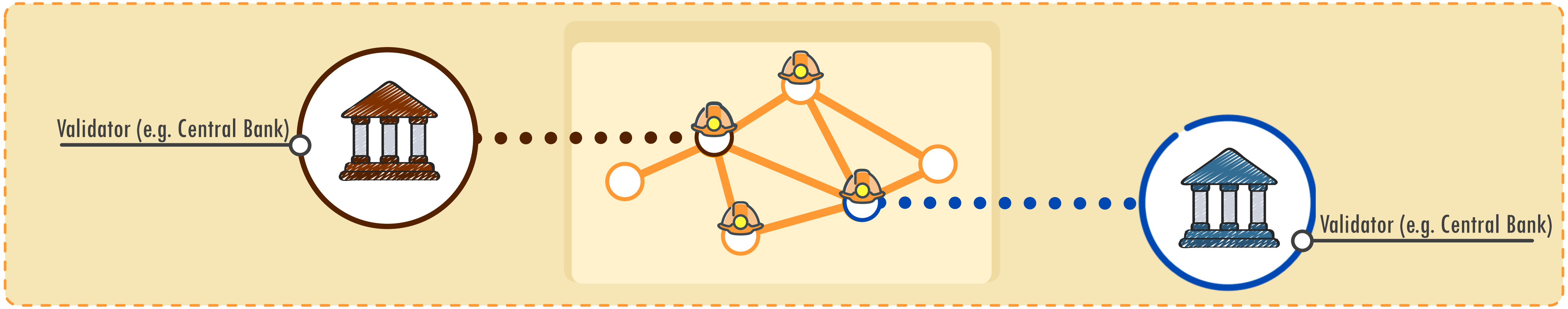




The Frostland problem

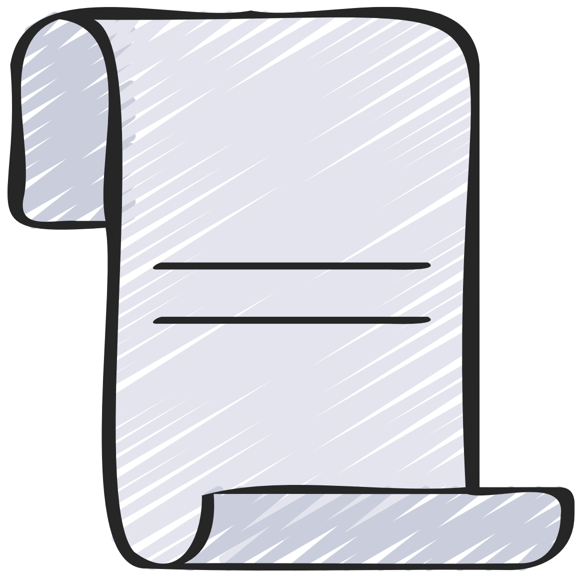
Ruffing, Tim, et al. "ROAST: Robust Asynchronous Schnorr Threshold Signatures." IACR Cryptol. ePrint Arch. 2022 (2022): 550.

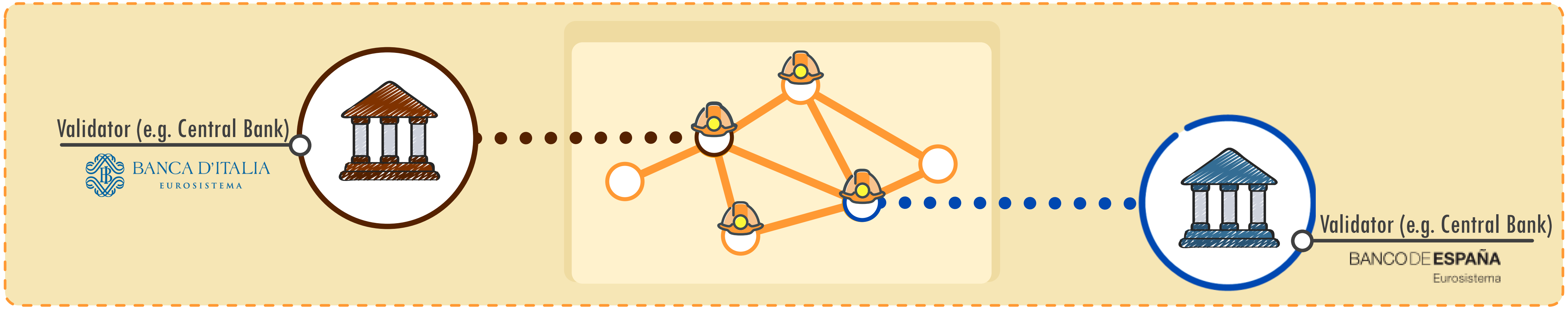




The Frostland problem

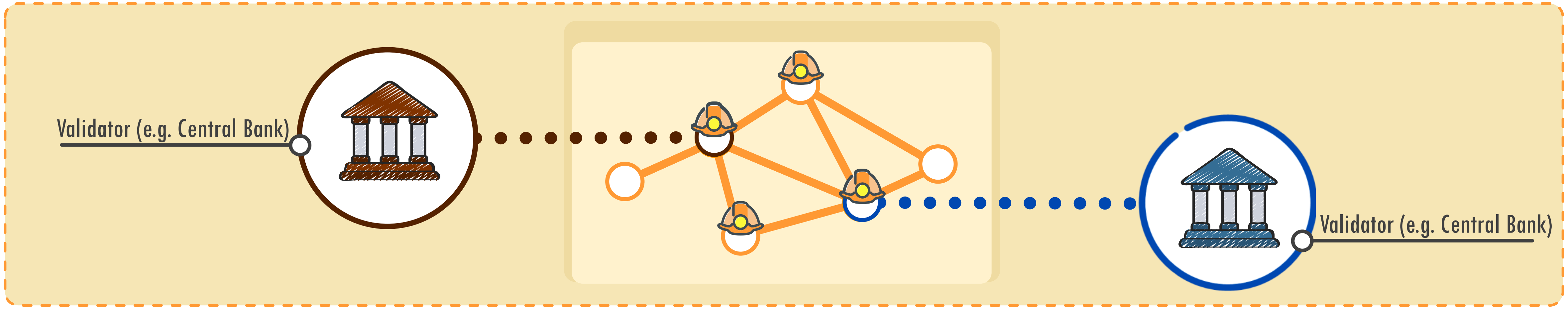
Ruffing, Tim, et al. "ROAST: Robust Asynchronous Schnorr Threshold Signatures." IACR Cryptol. ePrint Arch. 2022 (2022): 550.





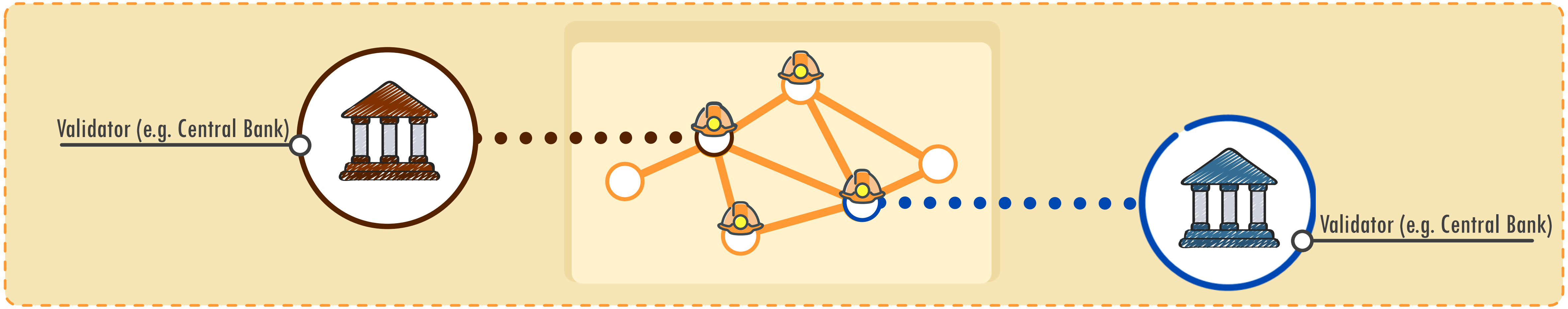
The Frostland problem

Ruffing, Tim, et al. "ROAST: Robust Asynchronous Schnorr Threshold Signatures." IACR Cryptol. ePrint Arch. 2022 (2022): 550.



The Frostland problem

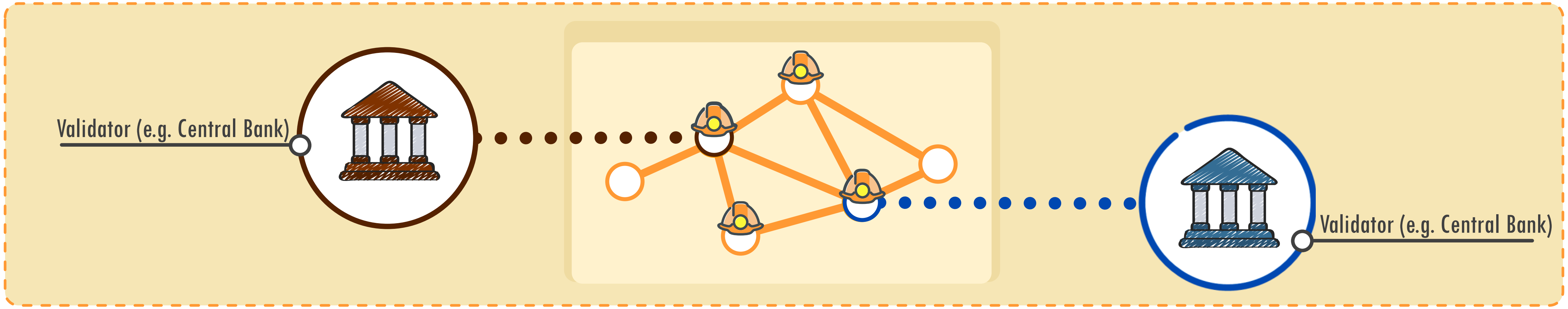
Ruffing, Tim, et al. "ROAST: Robust Asynchronous Schnorr Threshold Signatures." IACR Cryptol. ePrint Arch. 2022 (2022): 550.



The Frostland problem

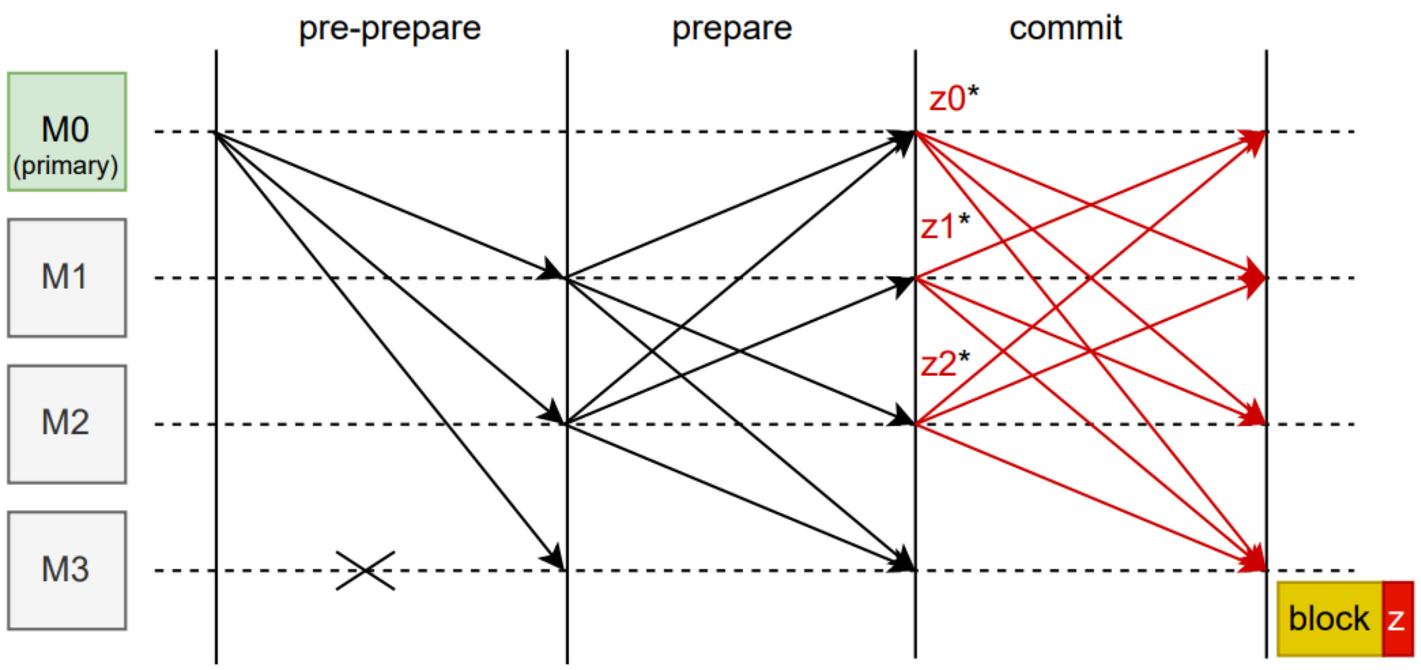
Ruffing, Tim, et al. "ROAST: Robust Asynchronous Schnorr Threshold Signatures." IACR Cryptol. ePrint Arch. 2022 (2022): 550.

Validators may refuse to sign even after their logo has been collected and stamped into the document!!



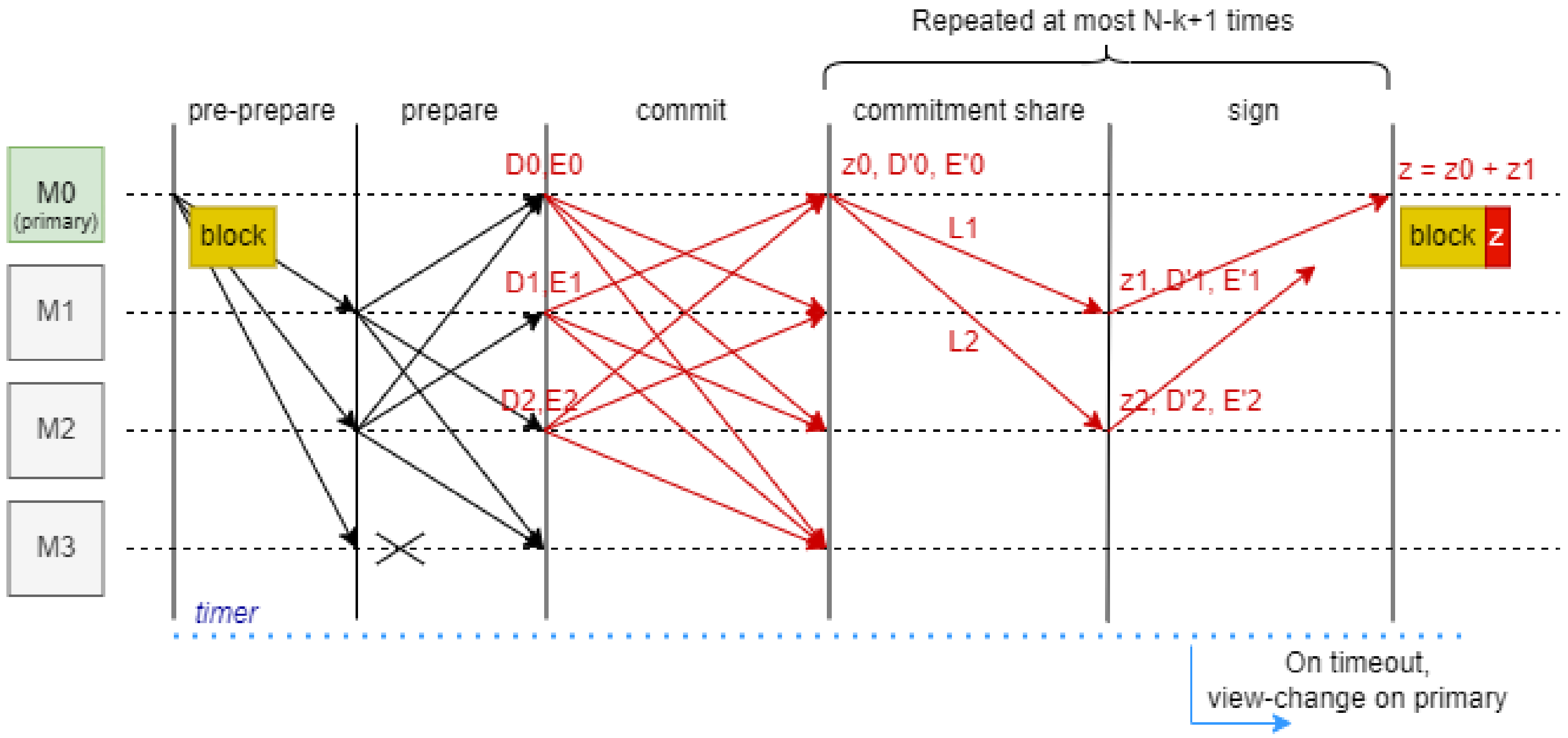
Signing blocks with FROST – 3FBFT

3FBFT: During the commit phase, each correct node creates a new signature share for all possible $\binom{N}{k}$ combinations of k actual signers out of N possible signers and appends all the shares to the commit message. Any node receiving a Byzantine quorum of commit messages can aggregate shares and broadcast the block. Optimizes the number of rounds but has a more than exponential complexity. Suitable only in very small mining networks.



Signing blocks with FROST - FBFT

FBFT: Appends two new rounds to PBFT, one to send the “document to sign” and one to collect signature shares from the signers. The primary aggregates shares and broadcasts the block. As demonstrated in ROAST, this terminates in a number of additional rounds that is *linear with N*, if the primary does not fail. Otherwise, timeout expires, and the view change is used to guarantee liveness.

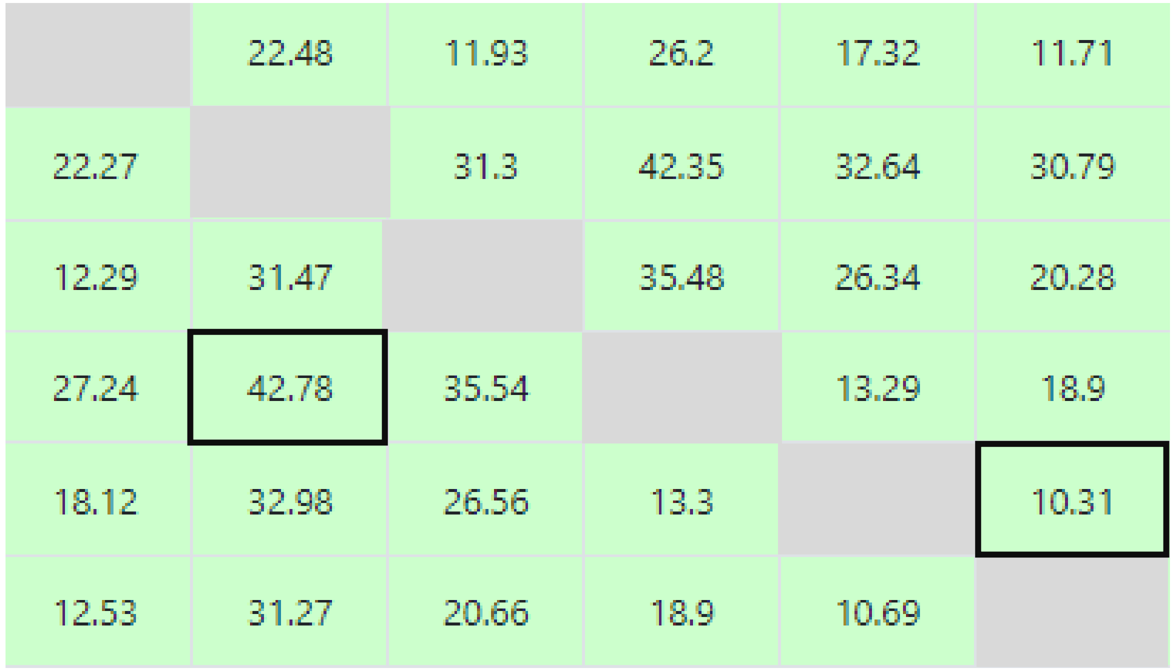


Experimental results

Geographically distributed benchmarking environment across 8 AWS European regions (Ireland, Germany, Italy, France, Sweden, England, Switzerland, Spain).

Experimental results

Geographically distributed benchmarking environment across 8 AWS European regions (Ireland, Germany, Italy, France, Sweden, England, Switzerland, Spain).

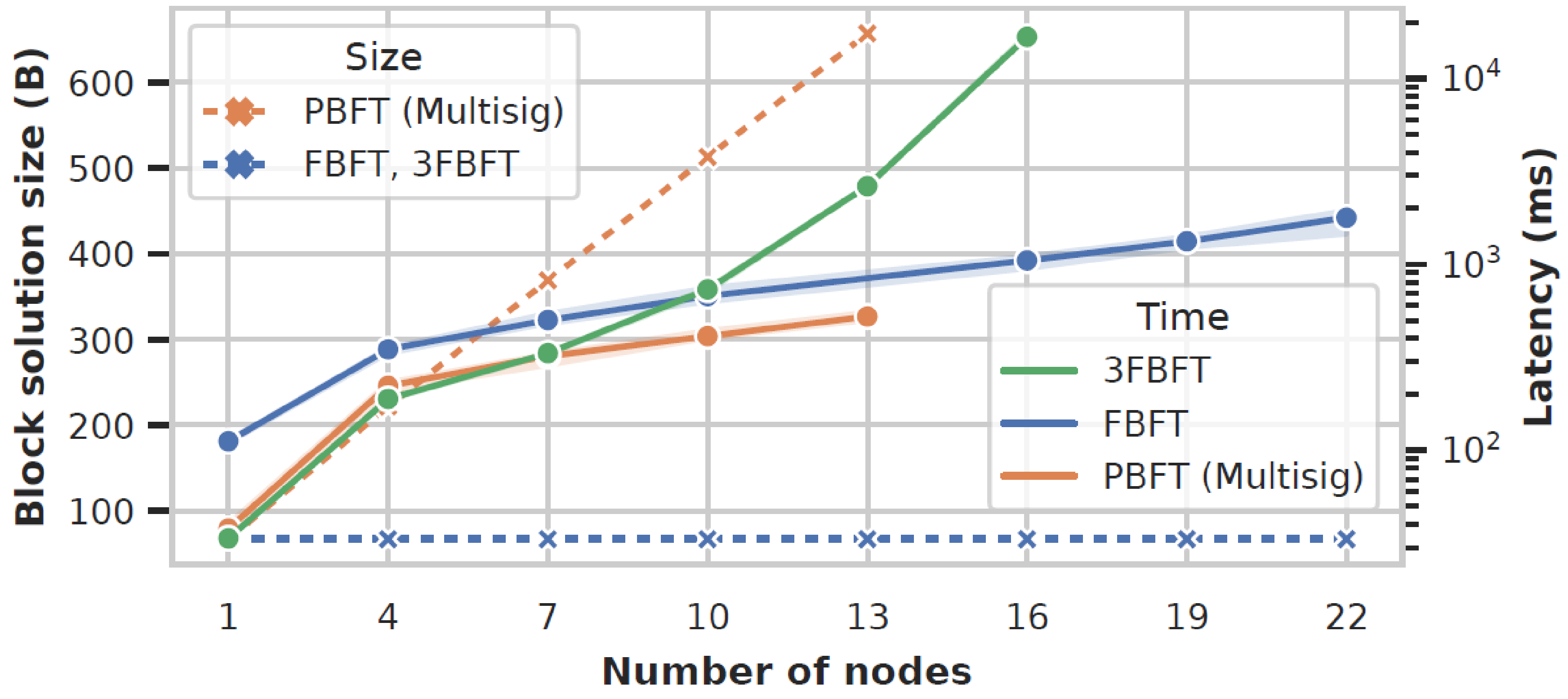


	22.48	11.93	26.2	17.32	11.71
22.27		31.3	42.35	32.64	30.79
12.29	31.47		35.48	26.34	20.28
27.24	42.78	35.54		13.29	18.9
18.12	32.98	26.56	13.3		10.31
12.53	31.27	20.66	18.9	10.69	

According to https://www.cloudping.co/grid inter-region latency in those European regions averaged between 10ms and 50ms in the last year.

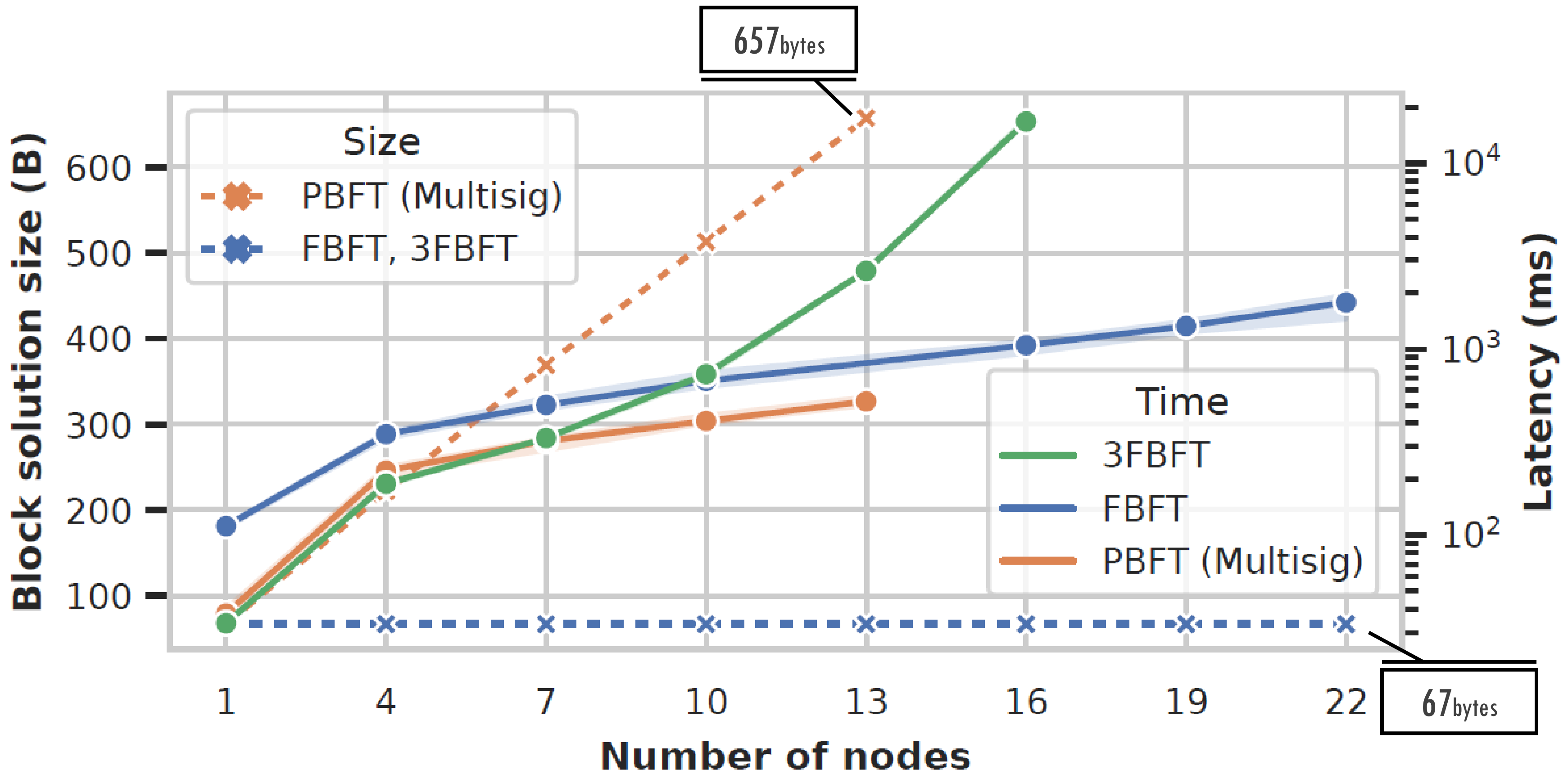
Experimental results, signature size and latency

The first results compare the block solution size and the latency of FBFT against PBFT (baseline) and 3FBFT, in absence of load



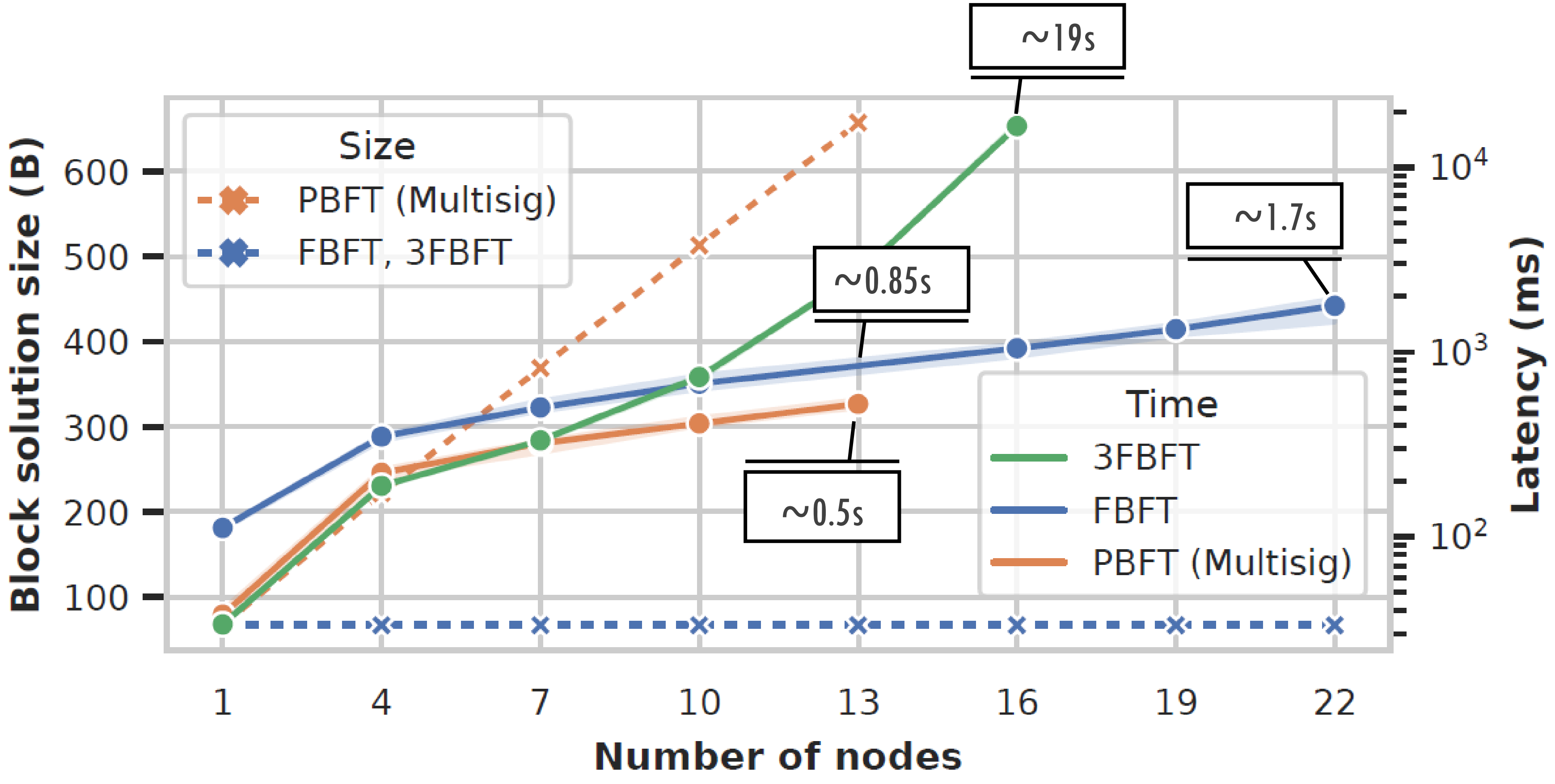
Experimental results, signature size and latency

When signature aggregation is not used, the block solution size grows linearly with the number of nodes. With FROST, the signature solution size is constant.

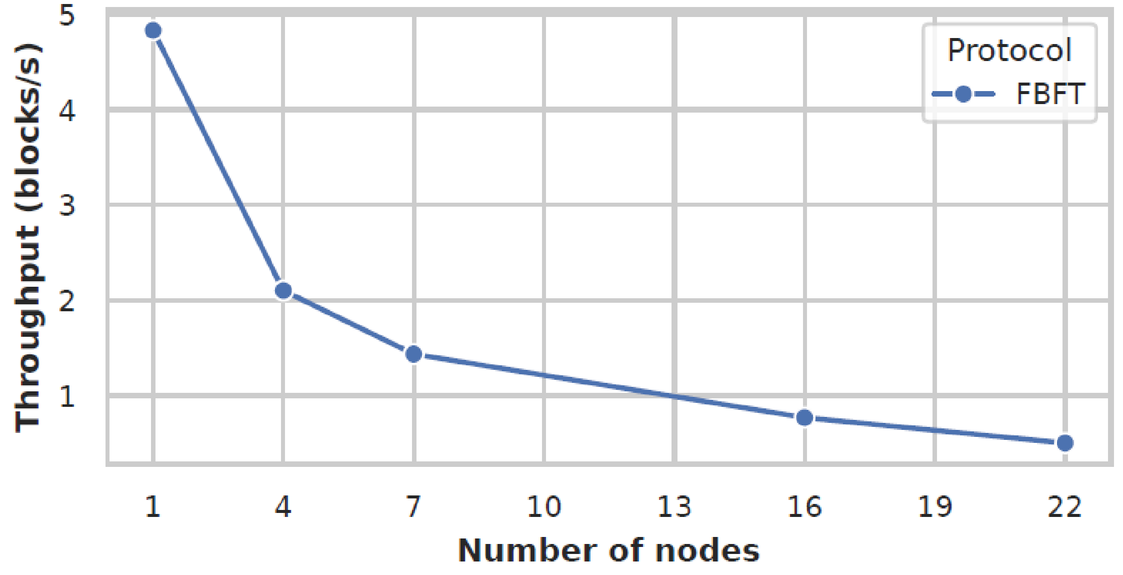


Experimental results, signature size and latency

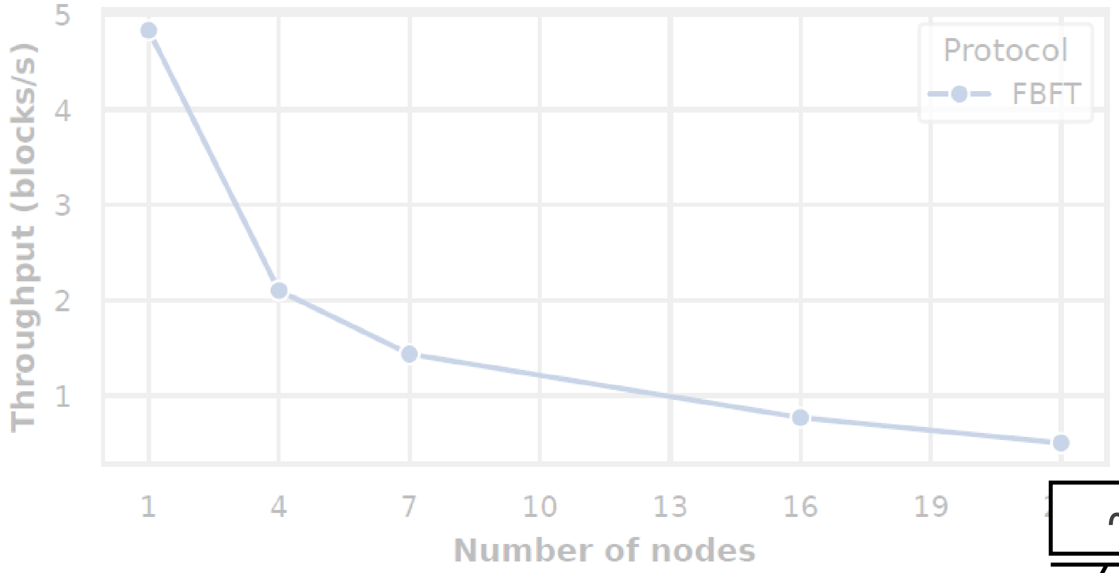
3FBFT has performances that are comparable with PBFT for small mining networks, but the latency sharply increase when the number of nodes grows. FBFT has just a reasonable increment in consensus latency with respect to PBFT.

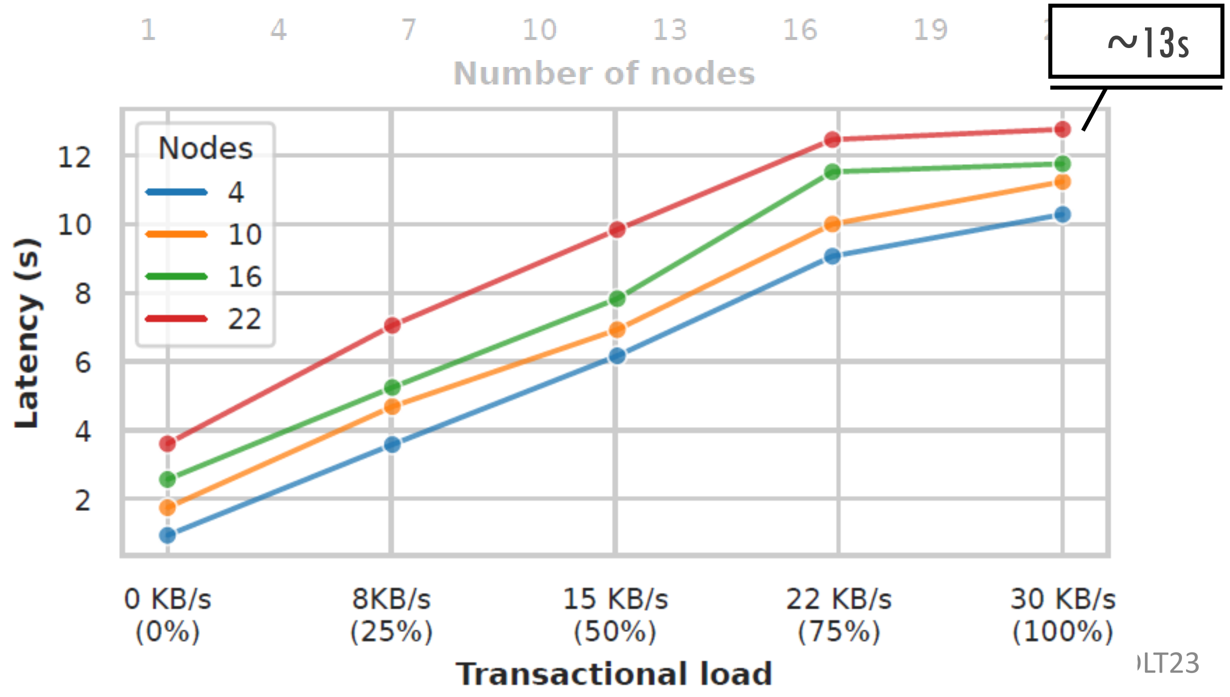


Experimental results, throughput

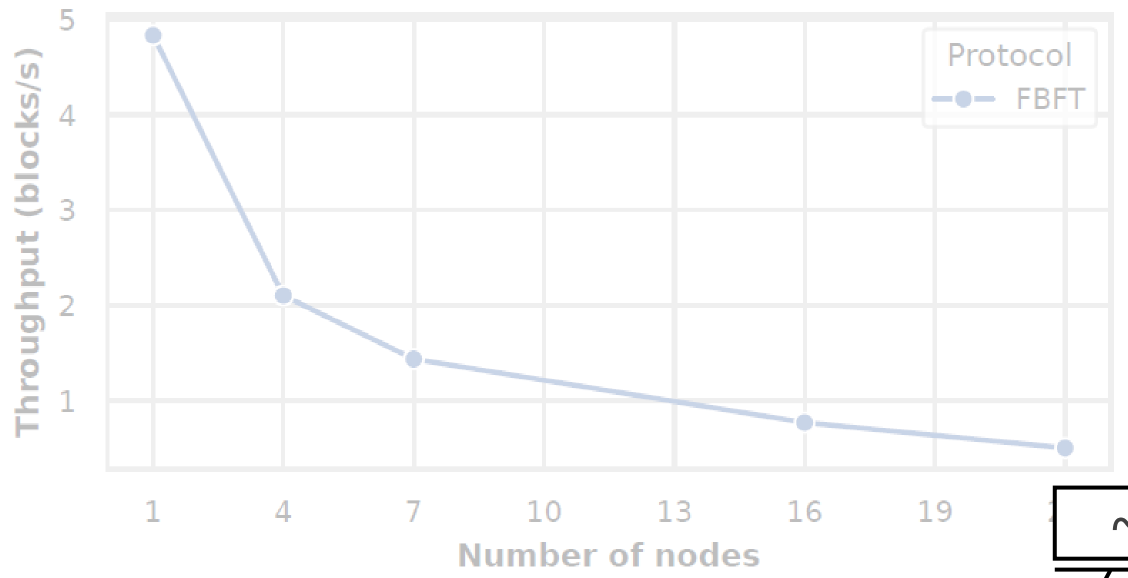


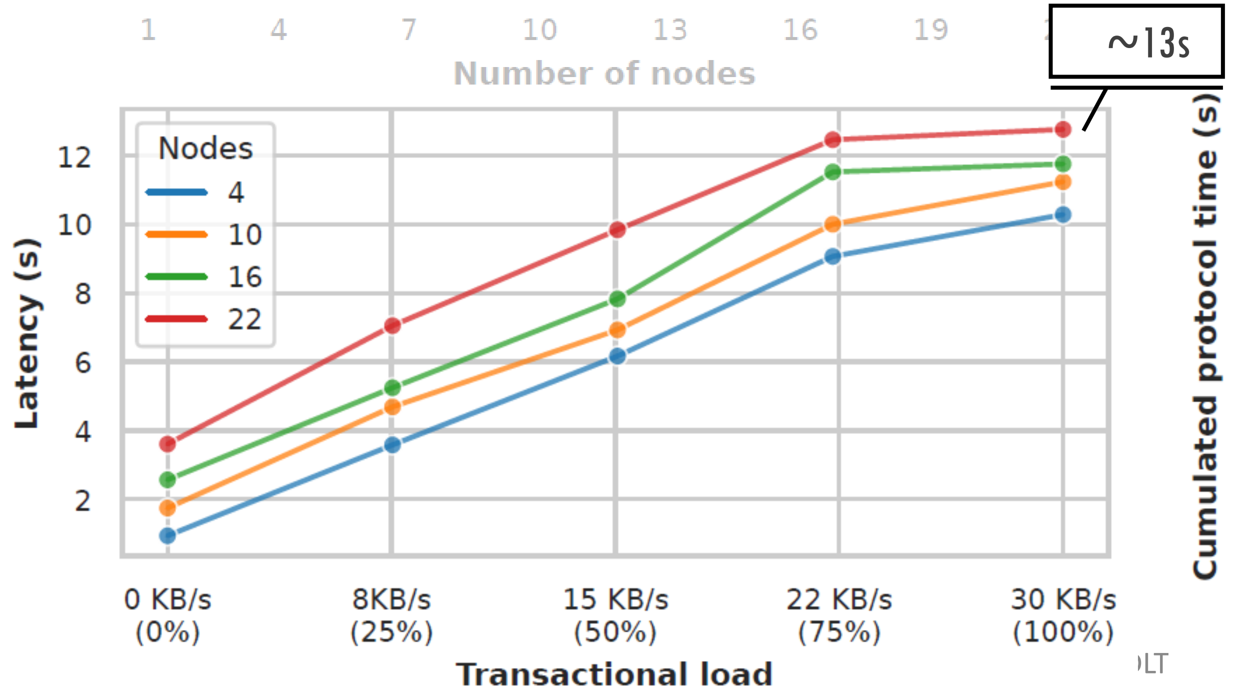
Experimental results, impact of transactional load

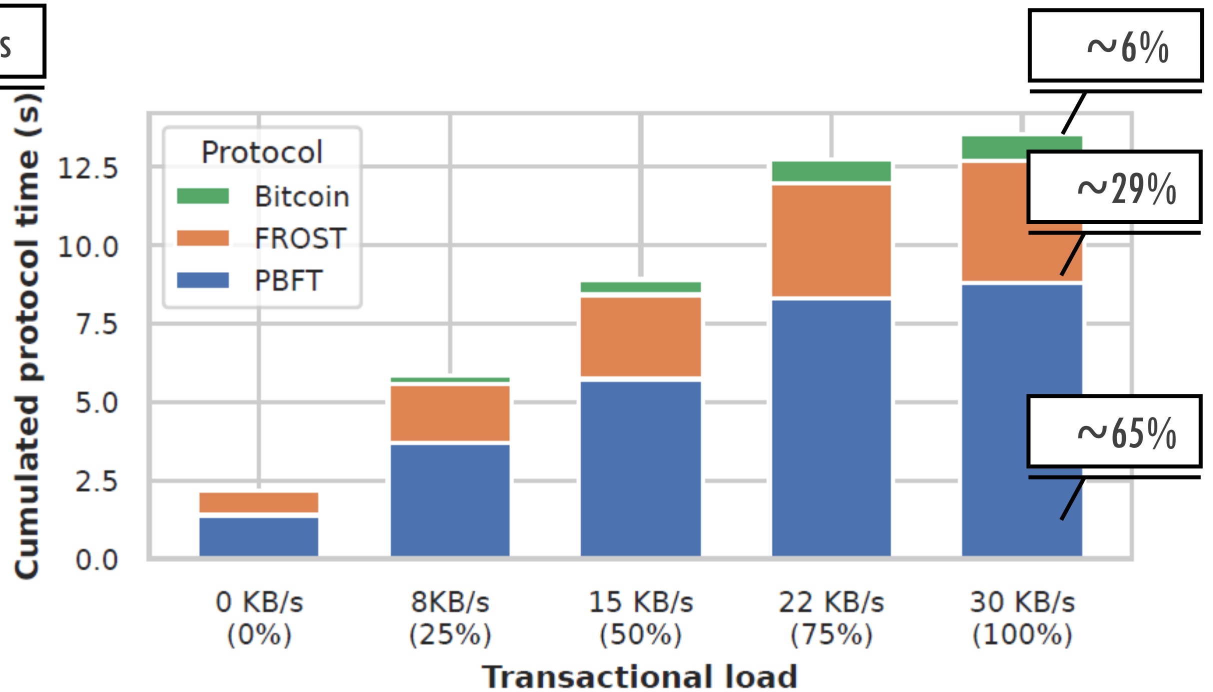




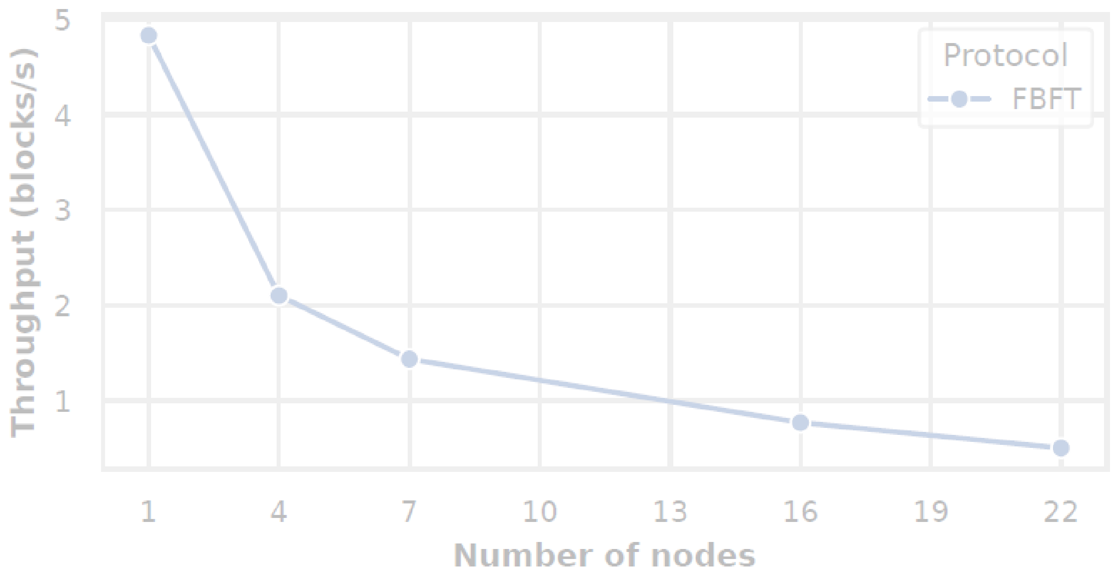
Experimental results, protocol dominance w/ load

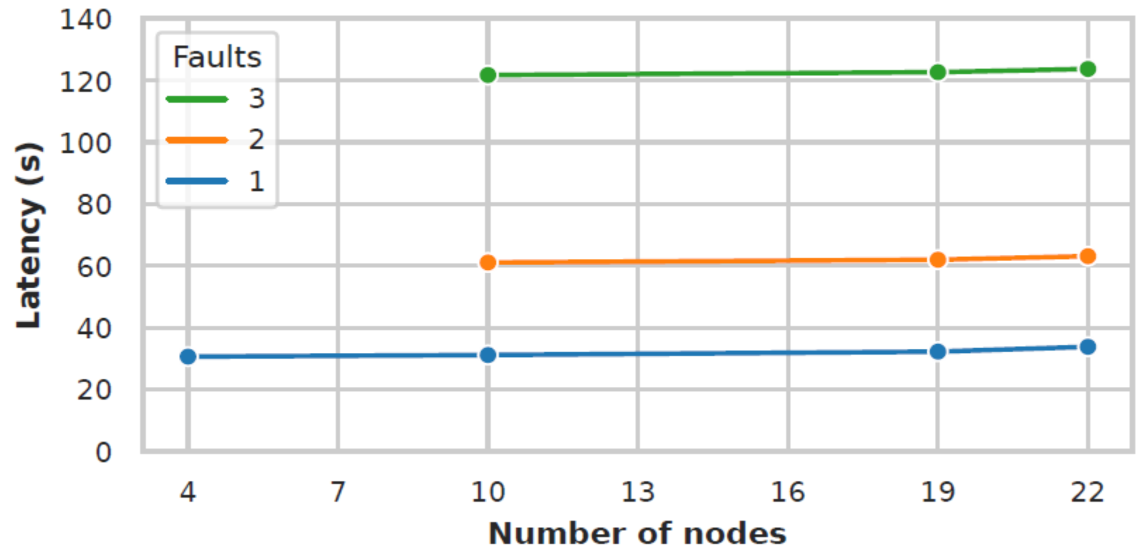


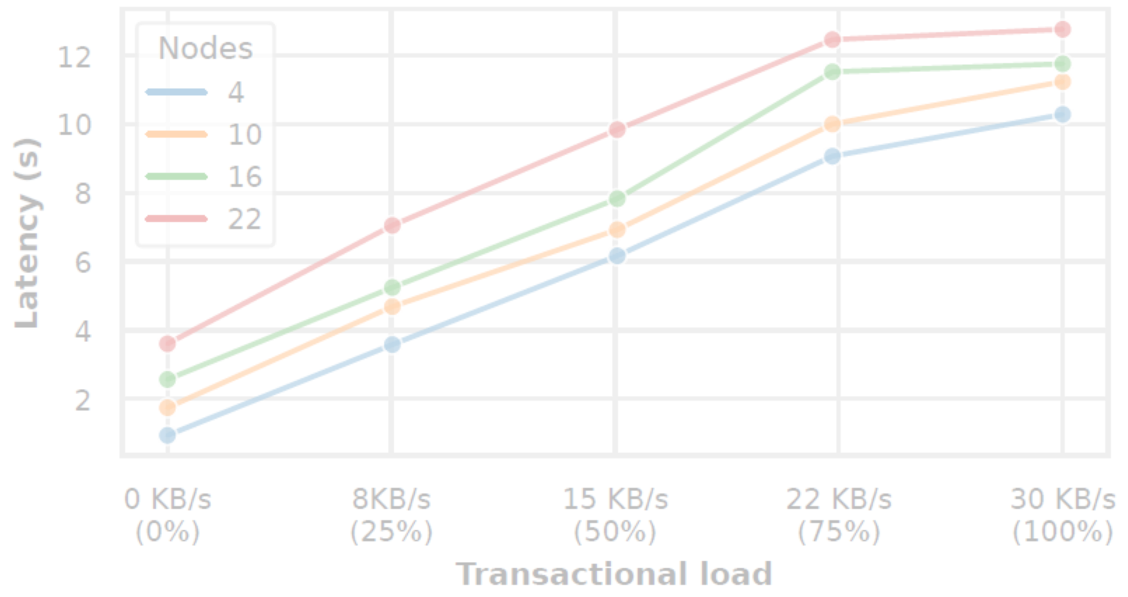


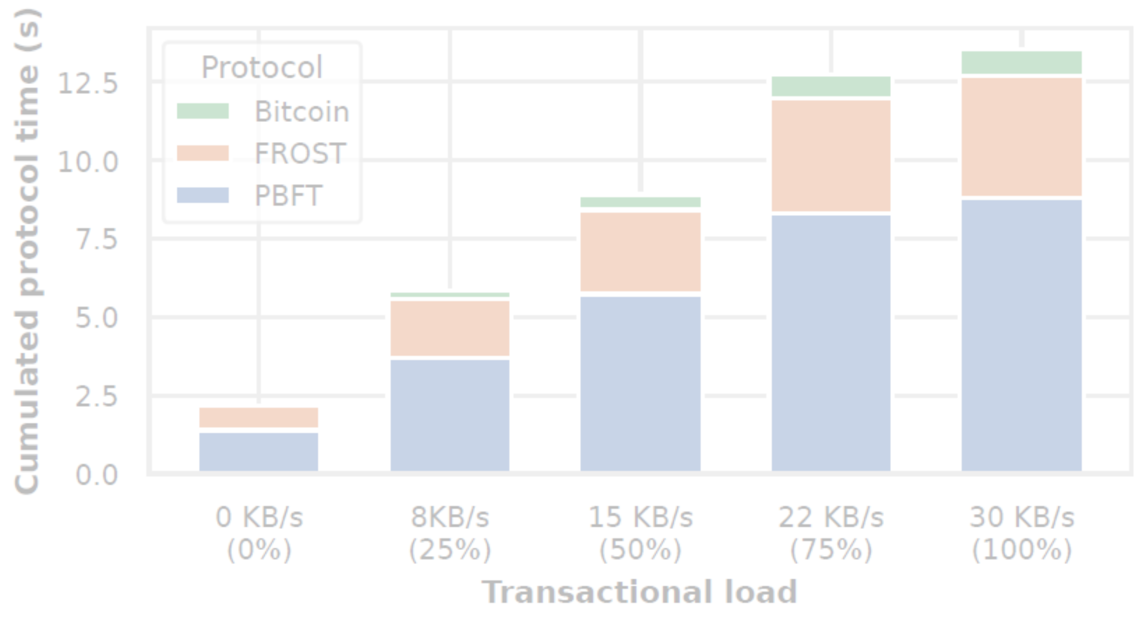


Experimental results, impact of failures (worst case)









Conclusions

- We presented a PoA Bitcoin-derived permissioned DLT, in which blocks are signed by a federation of independent actors and transactions enjoy deterministic finality;
- Using PBFT, the federation can operate also in presence of Byzantine failures of a subset of its members, providing high availability and fault tolerance;
- Using FROST, the signatures of the federation members are aggregated, improving the block space usage efficiency and preserving the confidentiality of the mining network configuration and its quorums;
- We evaluated our algorithm in a geographically distributed environment, showing that FBFT can satisfy all our requirements for just a reasonable increment in consensus latency;
- In the future, we plan to evolve our architecture towards different research directions: dynamic federations, fairness, privacy and scalability

Everything is open source!

https://bancaditalia.github.io/itcoin

Thanks for your attention!