# Economic design of distributed protocols in the blockchain era

Keynote SERIAL@Middleware2018

**Sara Tucci-Piergiovanni, Ph.D.**

**Joint work with Yackolley Amoussou-Guenou, Bruno Bias, Antonella del Pozzo, Maria Potop Butucaru**

# HISTORICAL PERSPECTIVE ON THE BLOCKCHAIN

From the early 80s the vision of digital money has been around – but it took more than a quarter of century before a fully distributed solution became a reality.

**Electronic cash**
[Chaum 1982], [Law et al 1996]

**Untraceability**

Token forgery and multiple spending avoided by a **trusted third party**

**B-money, RPOW**
[Day 1998][Finney 2004]

**Minting** money through **PoW**

Token forgery and multiple spending avoided by **trusted entities**

**Bit Gold**
[Szabo 2003, 2005] [Mahlki, Reiter 1998]

**Byzantine quorum** system based on voting

**Decentralized** but vulnerable to Sybil attacks

**Bitcoin**
[Nakamoto 2008]

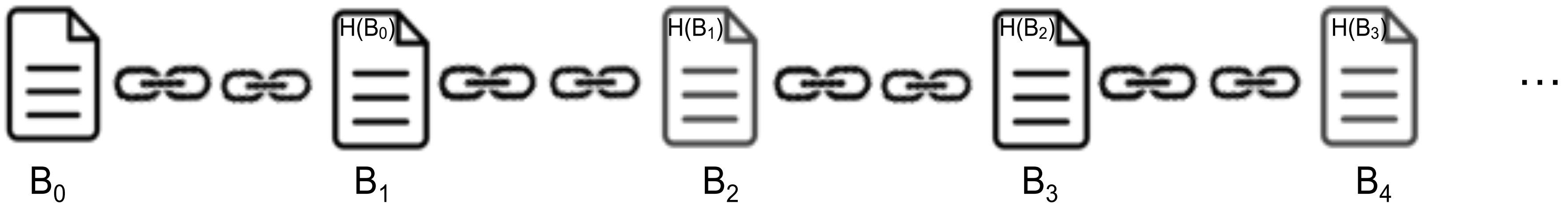
# BITCOIN [Nakamoto 2008]

Combination of all the abovementioned techniques for **full decentralization**

Proof-of-Work used to

- Limit the number of votes per entity (against Sybil Attack)
- Limit multiple spending (coupled with longest chain rule)
- Minting and Incentives for miners: miners as rational profit seekers, it must be profitable to follow the protocol

# BLOCKCHAIN



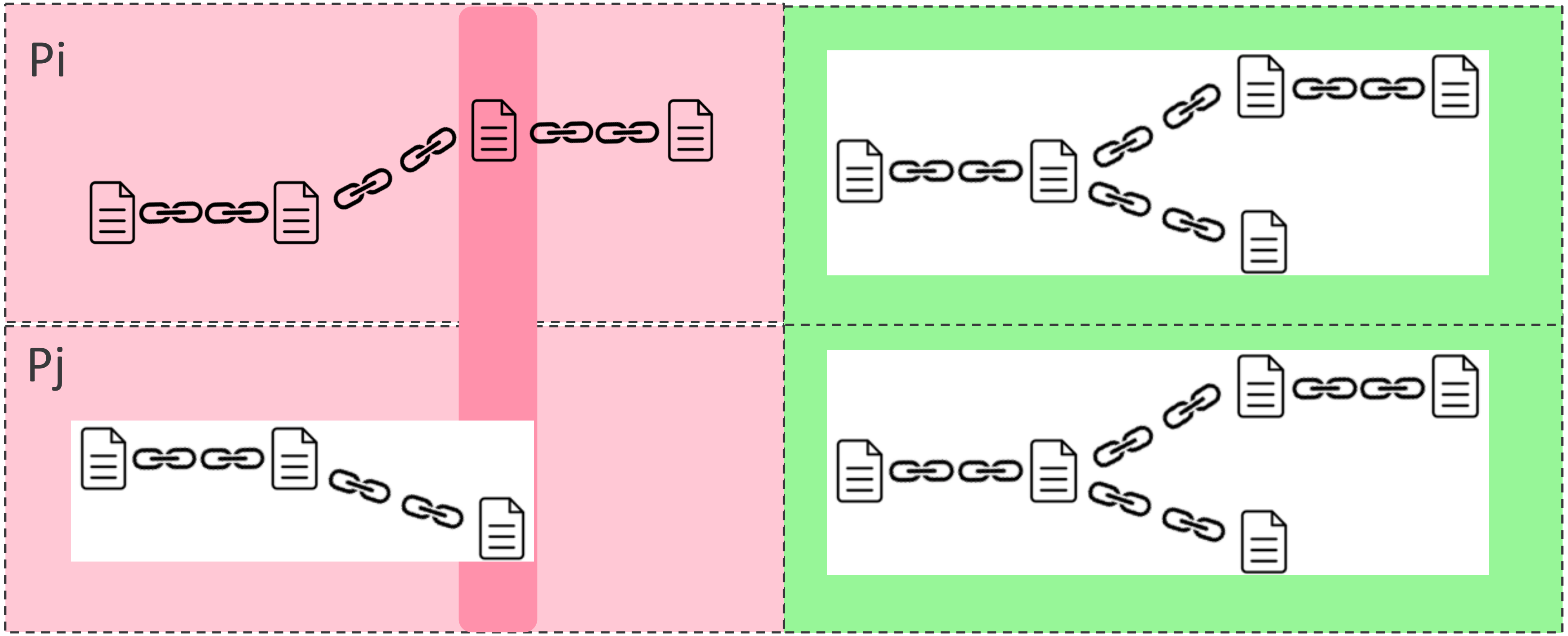

A Data Structure

- A sequence of blocks, each containing transactions, replicated at each process $\mathbf{p_i}$
- A block $\mathbf{B_h}$ at level **h** is linked to the block $\mathbf{B_{h-1}}$ at level **h-1** by containing the hash of $\mathbf{B_{h-1}}$

The (Bitcoin) Protocol to update the data structure at $\mathbf{p_i}$

- Make a block $\mathbf{B_h}$ solving PoW
- Broadcast $\mathbf{B_h}$
- Upon reception of $\mathbf{B_h}$ : verify $\mathbf{B_h}$ and locally append $\mathbf{B_h}$ if $\mathbf{B_h}$ is valid
- $\mathbf{B_h}$ contains the reward for the miner that made it

# CONSISTENCY ISSUES: FORKS




Forks are possible because

- More than one block produced for a given height
- Network delays and reordering

If all updates eventually arrive, then forks are solved with a local rule – reconciliation

# ECONOMIC-RELATED ISSUES

- **Monopoly**. In Bitcoin, we can take the idiom “rich gets richer” literally: it has been shown that the wealth of rich users increases faster than the wealth of users with low wealth [Kondor et al. 2013]

- **Waste** of computational power, and thus energy, without any intrinsic value

- **Participation failure**. The participants of Bitcoin pay the miners via fees
  - Each individual user’s (selfish) interest is to let others pay the fees. Users might therefore start to issue transactions without fees. If the majority acts this way, mining becomes unprofitable, and miners will give up [Bentov al 2014].
  - User fairness is compromised because waiting cost is not taken into account by miners [Gurcan et al 2017].

# Questioning

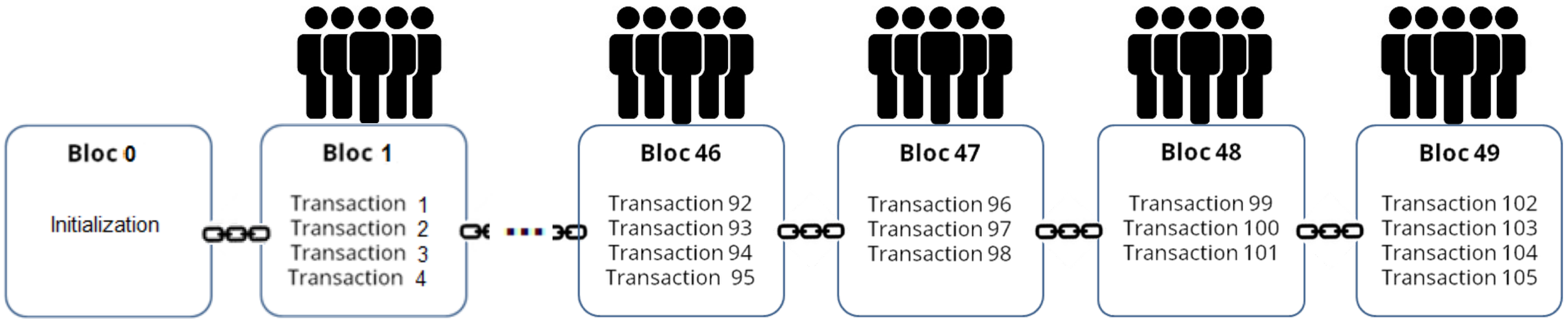
## eventual consistency & proof-of-work

and looking for alternatives considering the
basic requirements for an
**open** and **decentralized** system

1. The participants could join and leave at will
2. Consensus cannot hinder (too much) scalability
3. The block generation must be « expensive »
4. The participants should consider profitable to follow the protocol
5. Participants must not able to gain an over proportionally ability to mint coins

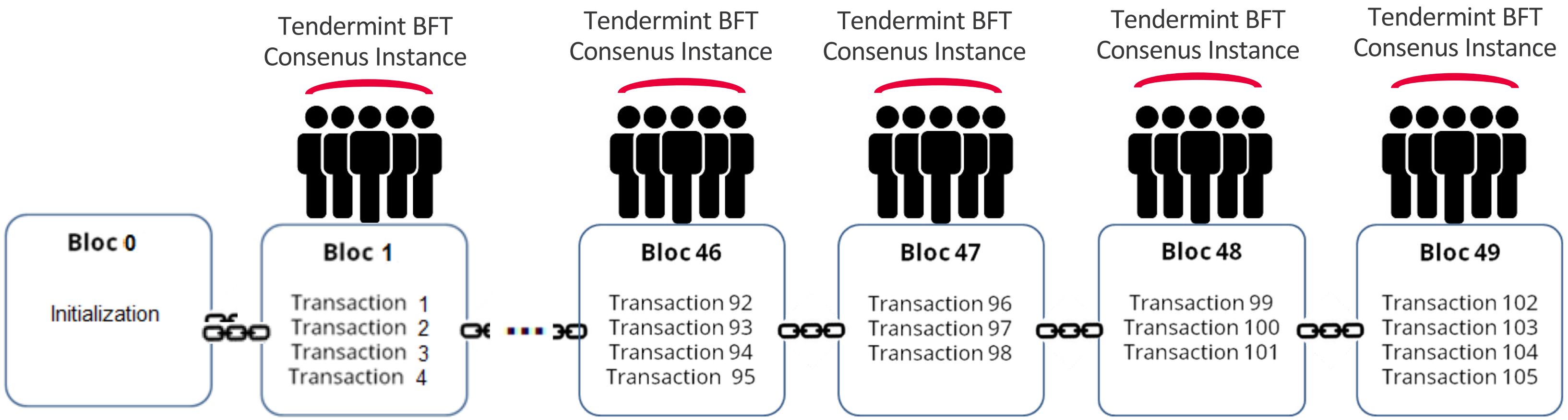
**Can we do it ?**

# COMMITTEE/CONSENSUS-BASED BLOCKCHAIN

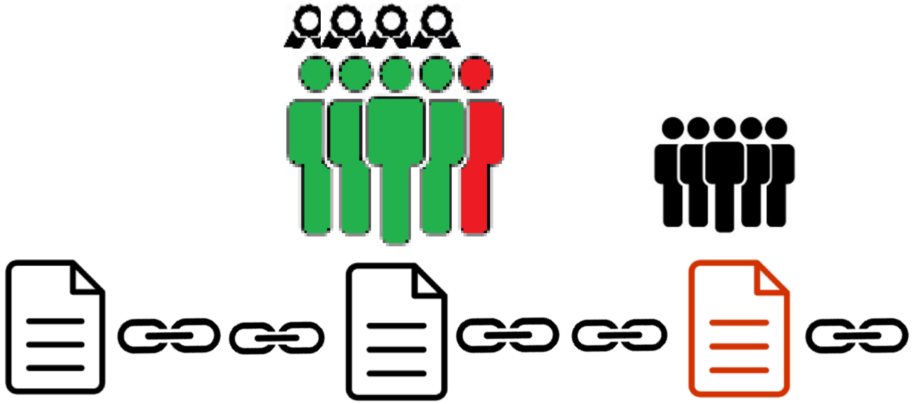



- Committee with a fixed number **N** of validators for height **h** run a Consensus to produce the next block, then broadcast to the network
- Be selected as validator should be expensive – i.e., locking funds
- Profitability and fairness depends on on how many times a participant is selected and rewarded for the work done to produce a block

# LET US TAKE ONE EXEMPLE: TENDERMINT




Selection made on same deterministic rule on the unique chain based on a merit parameter $\alpha$ in [0,1]



Reward is distributed by the **next committee to those that voted in the previous one**

# FAIRNESS IN CONSENSUS-BASED BLOCKCHAINS [Amoussou et al. 2018]

- ## Selection mechanism

We says that a selection mechanism is **fair** if process with merit parameter $\alpha$ will be selected at least $\alpha$ times in any sufficiently long window of the chain [Garay 2014]

| | | | |
|---|---|---|---|
| N=1 | ($p_0$; 0.20) ($p_1$; 0.80) | 110000000010000001000 | not fair |
| | | 101111111110011011111 | fair |

- ## Reward mechanism

We says that a reward mechanism is **fair** if all and only the ones that contributed to the block election are **rewarded**

Note that this definition of fairness works with a static merit parameter $\alpha$. This implies that rewarding does not change the merit parameter (for now it is an assumption).

# WHICH SYSTEM MODEL TO ASSUME?

## ALGORITHMS

Consensus → Algorithms in the committee +Broadcast in the network

Rewarding → Algorithms in the committee +Broadcast in the network

Selection → Algorithms in the network

## SYSTEM MODEL

**Participant behavior**

- Rational
- Byzantine/Correct
- Byzantine/Rational/Altruistic
  [Ayer et al SOSP 2005] BAR Model

**Network behavior**

- Synchronous
- Eventually synchronous

**Arrival Model** [Aguilera 2004]

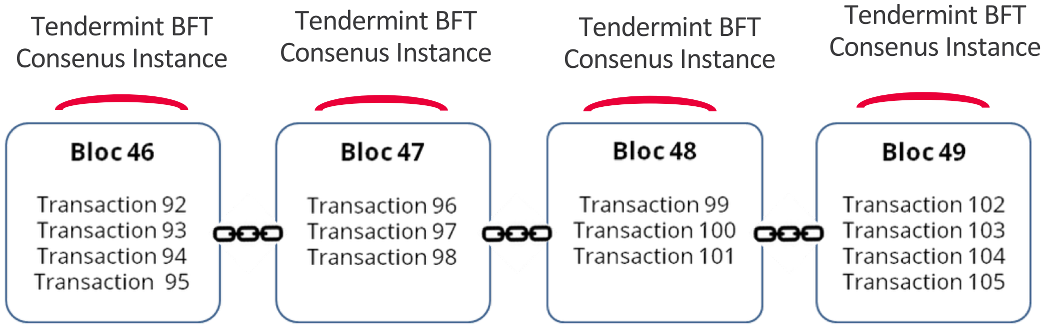
- Bounded finite arrival
- Finite arrival
- Infinite arrival

# TENDERMINT ANALYSIS

We proved under

- **Byzantine/Correct**
- **Eventually synchronous**
- **Finite arrival model**

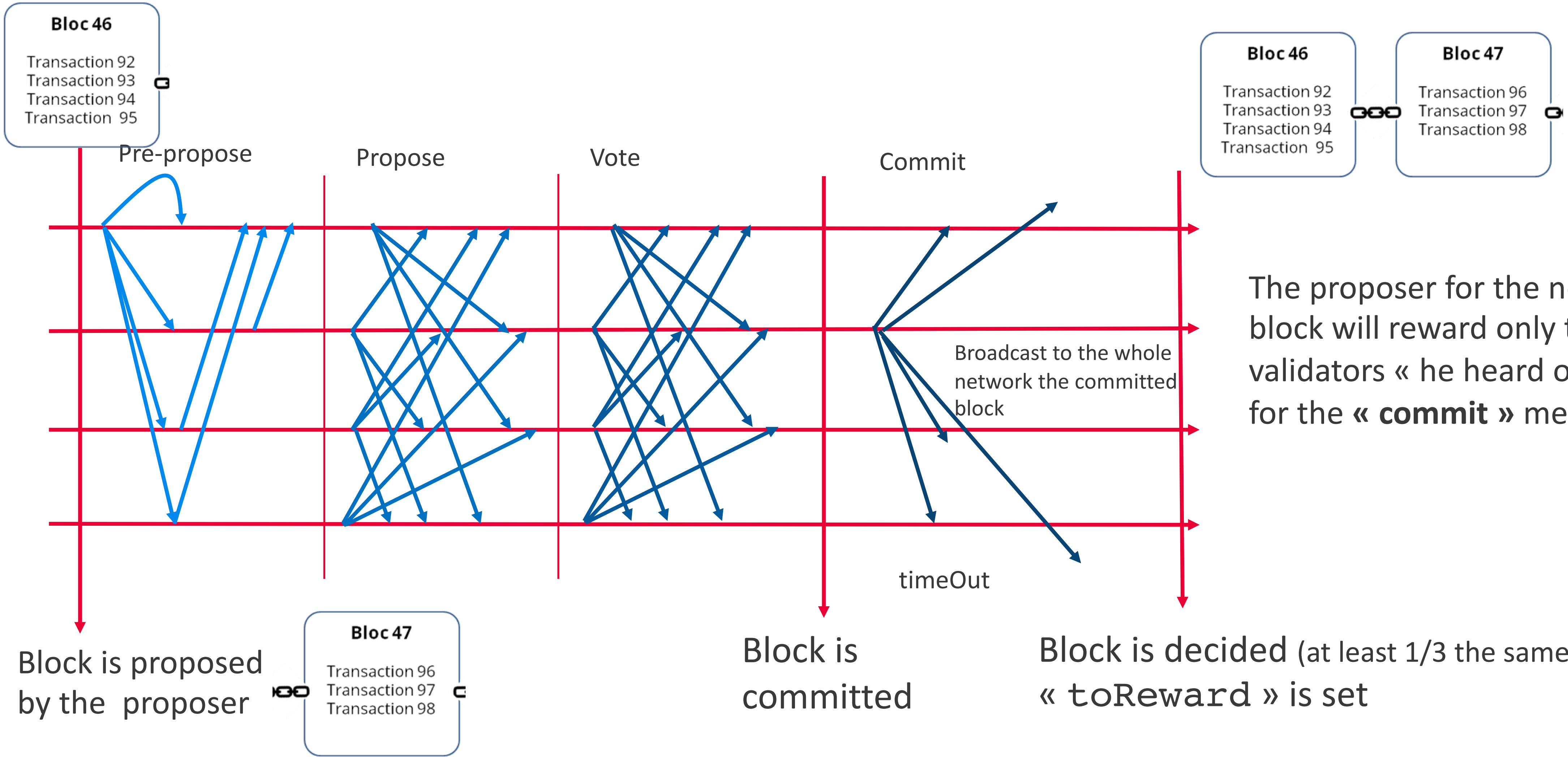
Tendermint BFT Consensus Correctness [Amoussou et al. OPODIS 2018]




and for Fairness [Amoussou et al. 2018]

- We proved that the rewarding mechanism cannot be fair in a non-synchronous network
- We weaken the definition to eventually fair. It is possible to get a rewarding mechanism eventually fair
- We proved that the Tendermint rewarding mechanism is not eventually fair

# TENDERMINT REWARDING MECHANISM



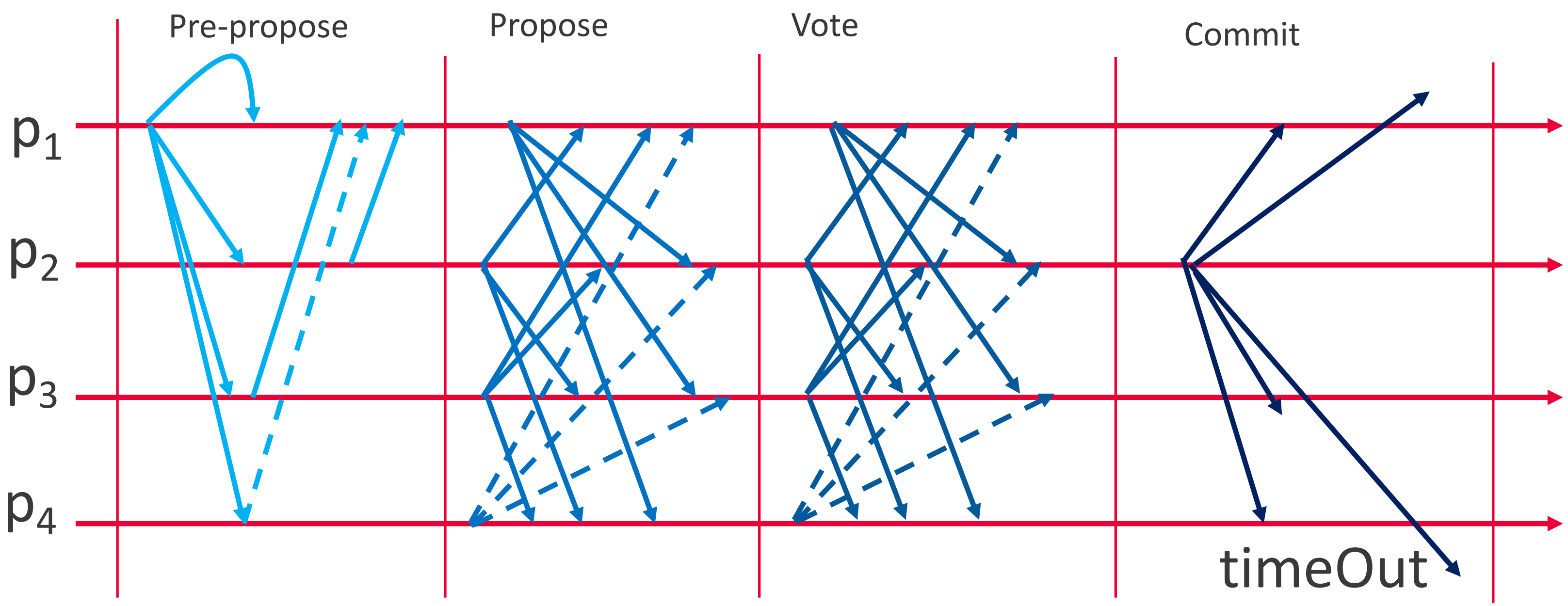

The proposer for the next block will reward only those validators « he heard of » for the **« commit »** message

Block is proposed by the proposer

Block is committed

Block is decided (at least 1/3 the same block)
« `toReward` » is set

# TENDERMINT REWARDING MECHANISM

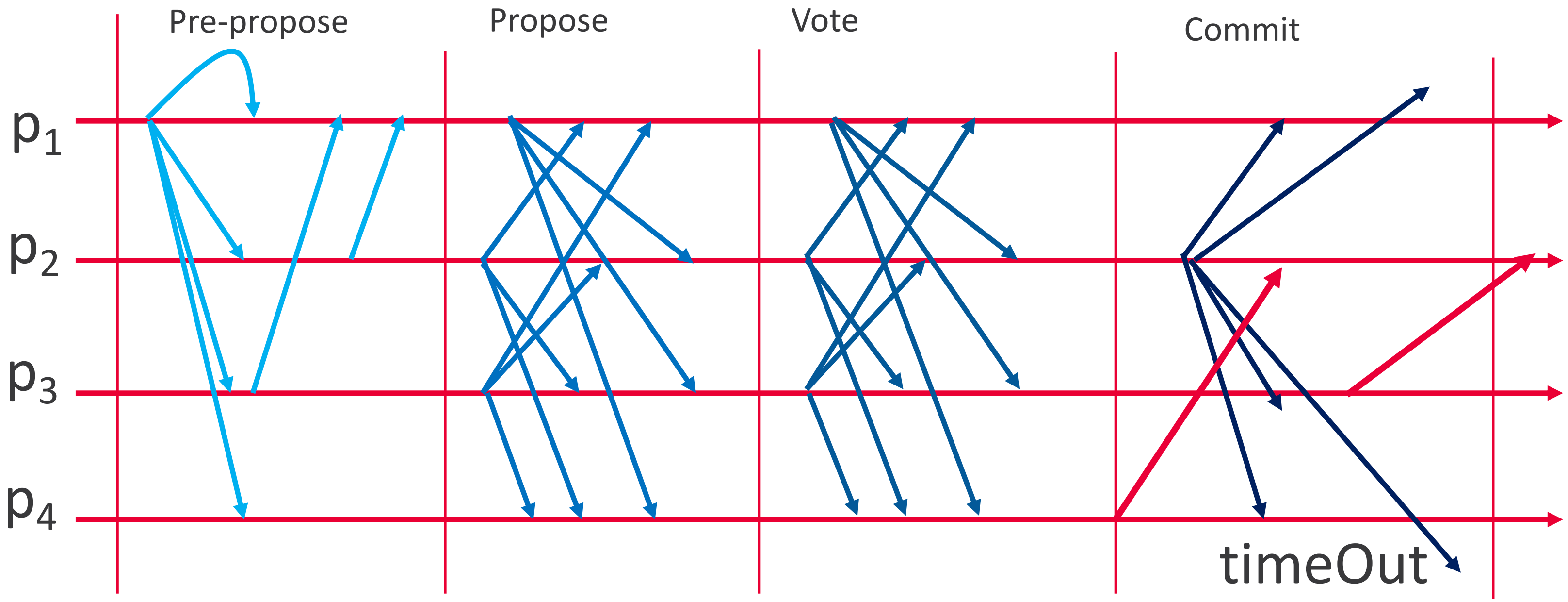



This scenario can happen an infinite number of times in an eventually synchronous system with a fixed timeout, a process that participated is never rewarded

If adaptive timeout, the protocol can catch up and $\mathbf{p_3}$ is rewarded

The commit message does not keep track of those that participated in the previous phases. A process that did not participate can always be included (e.g. $\mathbf{p_4}$). The rewarding mechanism is not fair.

# TENDERMINT REWARDING MECHANISM




toReward = $\{p_2, p_4\}$

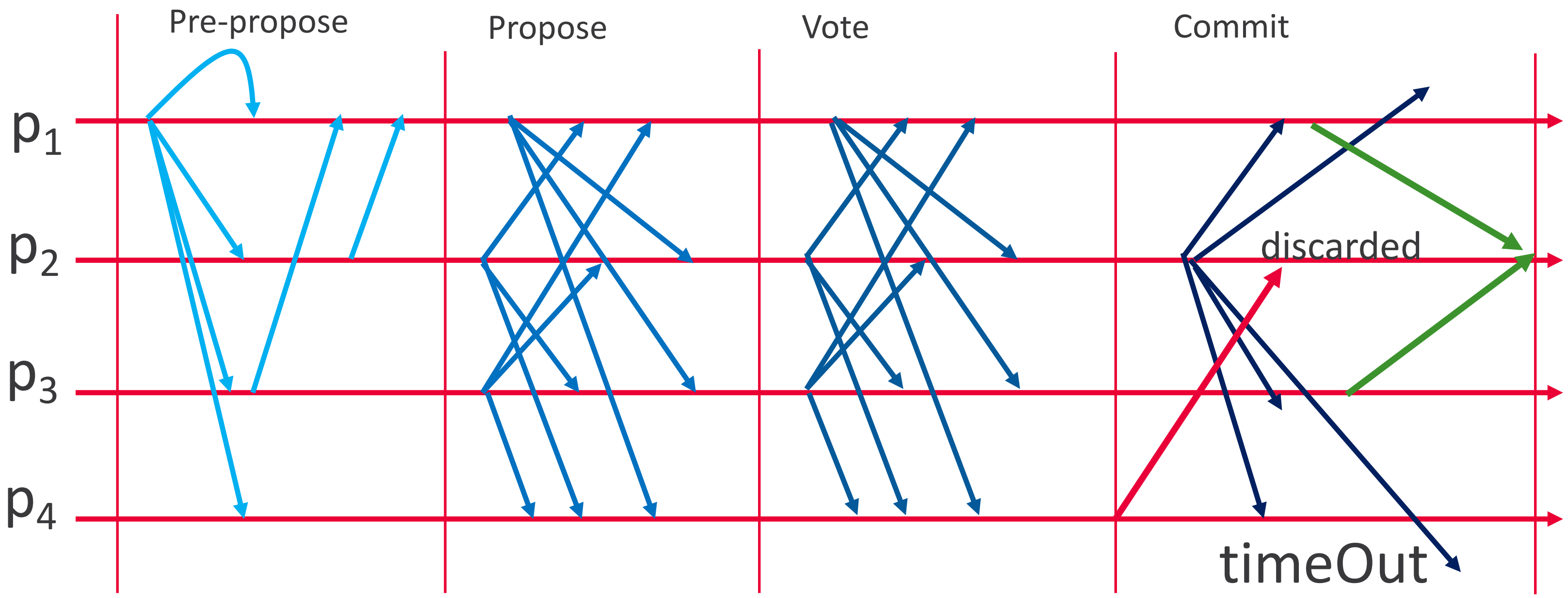
$p_3 \notin$ toReward

This scenario can happen an infinite number of times in an eventually synchronous system with a fixed timeout, a process that participated is never rewarded

If adaptive timeout, the protocol can catch up and $\mathbf{p_3}$ is rewarded

The commit message does not keep track of those that participated in the previous phases. A process that did not participate can always be included (e.g. $\mathbf{p_4}$). The rewarding mechanism is not fair.

# TENDERMINT REWARDING REVISED




toReward = {$p_2$, $p_3$}

Adaptive timeout, the protocol can catch up and $\mathbf{p_3}$ is rewarded

The commit message must keep track of those that participated in the previous phases.
Each process $\mathbf{p_i}$ in the COMMIT message includes a digitally signed list of those "he heard of" during the three phases
*Endorsement*: the process $\mathbf{p_i}$ is included in the `toReward` list only if at least one third of COMMIT messages includes $\mathbf{p_i}$

# ASSUMING RATIONAL BEHAVIOR

- Rational processes are self-interested and seek to maximize their benefit according to a known utility function

- Rational processes will deviate from the « suggested » protocol if and only if doing so increases their net utility

- The utility function must account for a process' costs (e.g., sending messages) and benefits (e.g., reward of a block) for participating in a system

- **If we consider that all processes are rational we study Nash equilibria**

## Tragedy of the commons

"A **dilemma** arising from the situation in which multiple individuals, acting **independently** and **rationally** consulting their own **self-interest** will deplete a shared resource, even when it is clear that it is not in anyone's long-term interest for this to happen."

# STRATEGIES AND NASH EQUILIBRIA

A **strategy** of a process *i* for a height *h* is a function $\sigma_i^h : N \rightarrow \{0, 1\}$ which given a round *r*, selects if the process sends a message (1) or not (0).

- $\sigma_i^h(r) = 1$, *i* sends the message during the round r.
- $\sigma_i^h(r) = 0$, *i* does not send the message during the round r.

A **strategy profile** is the vector $\sigma^h(r) = [\sigma_1^h(r), \ldots, \sigma_n^h(r)]$

Let $U_i$ : Strat → R be a **utility function** for the process *i*.

Let us denote with **$(\sigma{-}i, \sigma_i')$**$(r)$ the fact that i deviates from σ by doing $\sigma_i'$.

Nash Equilibrium : a Nash equilibrium is a strategy profile where no player can increase its utility by deviating alone from the strategy profile.

A strategy profile σ is a pure Nash Equilibrium iff for each i, and for all strategies $\sigma_i'$ of i, : $U_i(\sigma{-}i, \sigma_{i'}) \leq U_i(\sigma)$.

# SOME PRELIMINARY NON-OBVIOUS RESULTS (STILL WORKING IN PROGRESS)

| | Reward all | Reward only Senders |
|---|---|---|
| ***ν***=1 | Exactly one message is sent | All processes send a message<br>Inefficient: too costly |
| ***ν***>1 | Multiple equilibria<br>- No message is sent. Coordination failure<br>- Exaclty ***ν*** are sent | Multiple equilibria<br>- No message is sent. Coordination failure<br>- All processes send a message. Inefficient: too costly |

**We simplify the original protocol to just one phase: vote messages**
**The block is produced if *ν* messages are sent**
**Messages cannot be lost and arrive at the end of the round**

# CONCLUSIONS

Committee based blockchains are important for strong consistency (no-fork), however economical properties for those class of protocols must be **defined** and **carefully analyzed under clear system model assumptions**

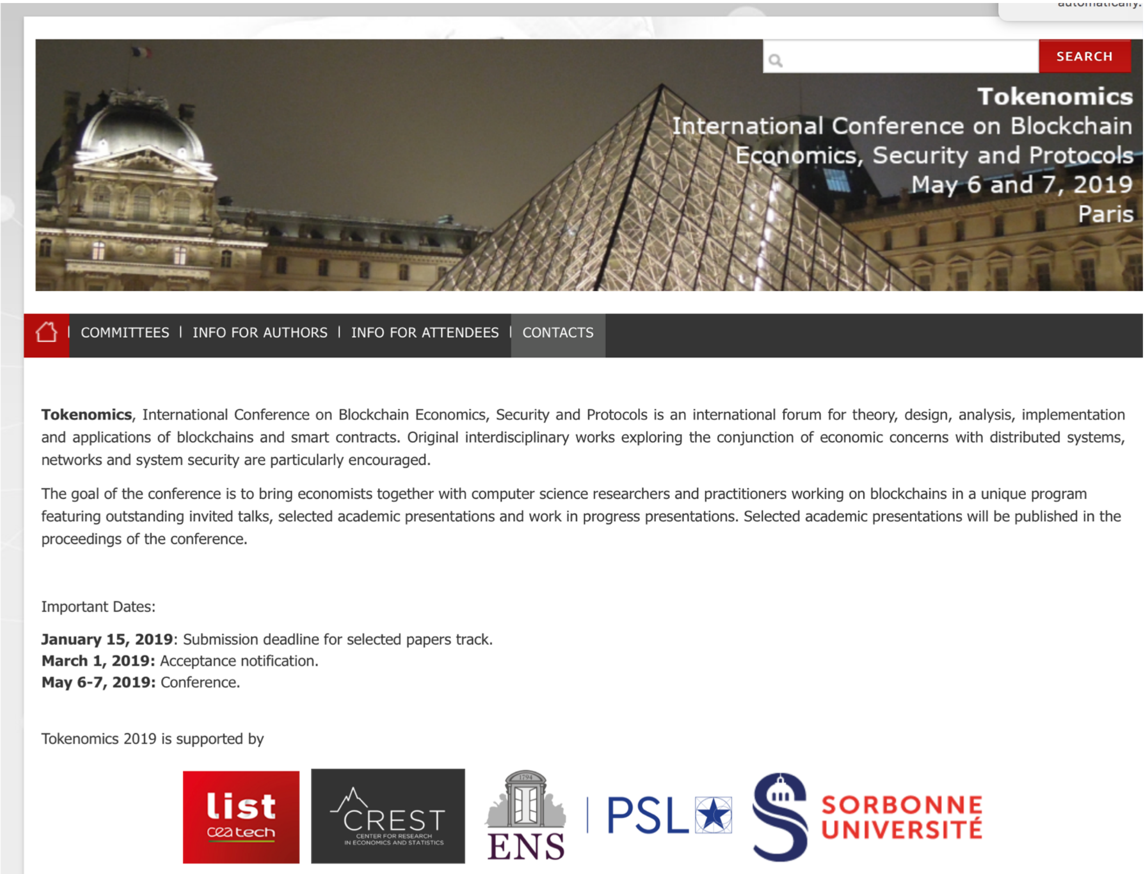
- Notion of fairness in Consensus-based Blockchains should separate the fairness of the selection mechanism from the fairness of the rewarding mechanism
- The effect of the network behavior has an impact on rewarding, analysis assuming a synchronous system is too limited
- Rational behavior analysis should  complement the Byzantine/correct one
- Rational behavior analysis should help to select the “right” reward function

# PERSPECTIVES

- Rational participants with message losses
- Mixing rational and byzantine behavior (BAR model)
- Selection mechanism is still an issue, need to define exact assumptions on the system model [Kiayias et al 2017] [Gilad et al 2017]
- Challenge: selection mechanism coupled with a reward mechanism that impacts the merit parameter. The merit parameter is dynamic and monopoly situations must be avoided

# TOKENOMICS

- http://www.tokenomics2019.org/tokenomics/



SEARCH

**Tokenomics**
International Conference on Blockchain
Economics, Security and Protocols
May 6 and 7, 2019
Paris

COMMITTEES | INFO FOR AUTHORS | INFO FOR ATTENDEES | CONTACTS

**Tokenomics**, International Conference on Blockchain Economics, Security and Protocols is an international forum for theory, design, analysis, implementation and applications of blockchains and smart contracts. Original interdisciplinary works exploring the conjunction of economic concerns with distributed systems, networks and system security are particularly encouraged.

The goal of the conference is to bring economists together with computer science researchers and practitioners working on blockchains in a unique program featuring outstanding invited talks, selected academic presentations and work in progress presentations. Selected academic presentations will be published in the proceedings of the conference.

Important Dates:

**January 15, 2019**: Submission deadline for selected papers track.
**March 1, 2019:** Acceptance notification.
**May 6-7, 2019:** Conference.

Tokenomics 2019 is supported by

# Thank you
# Questions ?

# REFERENCES


[Chaum 1982] David Chaum. Blind Signatures for Untraceable Payments. In CRYPTO ’82: Proceedings of the 2nd Conference on Advances in Cryptology. 199–203.

[Law et al. 1996] Law, Sabett and Solinas. How to Make a Mint: The Cryptography of Anonymous Electronic Cash. American University Law Review 46, 4 (1996), 1131–1162

[Dai 1998] Wei Dai. 1998. B-Money. (1998). http://www.weidai.com/bmoney

[Finney 2004] Hal Finney. 2004. RPOW. (2004). http://cryptome.org/rpow.htm

[Szabo 2003] Nick Szabo. 2003. Advances in Distributed Security. 2003).

[Szabo 2005] Nick Szabo. 2005. Bit Gold. (2005). http://unenumerated.blogspot.de/2005/12/bit- gold.html

[Malkhi and Reiter 1998] Dahlia Malkhi and Michael Reiter. 1998. Byzantine quorum systems. Distributed Computing 11, 4 (1998), 203–213.

[Nakamoto 2008a] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. (2008)

[Kondor et al. 2013] Kondor, Posfai, Csabai, and Vattay. 2013. Do the rich get richer? An empirical analysis of the BitCoin transaction network. arXiv preprint arXiv:1308.3892 (2013)

[Gurcan et al. 2017] Gurcan, Del Pozzo, Tucci-Piergiovanni. On the Bitcoin Limitations to Deliver Fairness to Users. In COOPIS 2017, 589-606

[Bentov 2014] Bentov and Kumaresan. How to Use Bitcoin to Design Fair Protocols. In CRYPTO ’14: Proceedings of the 34th Annual Conference on Advances in Cryptology. 421–439.

[Garay 2014] J. A. Garay, A. Kiayias, and N. Leonardos. The bitcoin backbone protocol: Analysis and applications. In Proc. of the EUROCRYPT International Conference, 2015.

[Aiyer et al 2005] Aiyer, Alvisi, Clement, Dahlin, Martin, Porth. BAR fault tolerance for cooperative services. SOSP 2005: 45-58

[Aguilera 2004] Aguilera. A pleasant stroll through the land of infinitely many creatures. ACM Sigact News, 35(2):36–59, 2004.

[Amoussou et al OPODIS 2018] Amoussou-Guenou, Del Pozzo, Potop-Butucaru, Tucci-Piergiovanni. Correctness of Tendermint-core Blockchains. OPODIS 2018

[Amoussou et al 2018] Amoussou-Guenou, Del Pozzo, Potop-Butucaru, Tucci-Piergiovanni. Correctness and Fairness of Tendermint-core Blockchains. CoRR abs/1805.08429 (2018)

[Kiayias et al 2017] Kiayias, Russell, David and Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. In Advances in Cryptology - CRYPTO 2017 357–388, 2017

[Gilad et al 2017] Gilad, Hemo, Micali, Vlachos and Zeldovich: Algorand: Scaling Byzantine Agreements for Cryptocurrencies. SOSP 2017: 51-68