Disciplina sobre Blockchain

Arlindo F. da Conceição (arlindo.conceicao@unifesp.br)

Consenso

Consenso no dia a dia

consenso

substantivo masculino

1. concordância ou uniformidade de opiniões, pensamentos, sentimentos, crenças etc., da maioria ou da totalidade de membros de uma coletividade.
   "o c. da cristandade"

Consenso em computação: leia-se "ordenação de eventos"

Permissioned *vs* Permissionless

- *Permissionless*: Ethereum, Bitcoin, etc.
  - Não autenticado
  - Anônimo
- *Permissioned*: Hyperledger, BigChainDB, Corda, etc.
  - Autenticação
  - Controle de acesso
  - Consenso bizantino: PBTF...
  - Escalabilidade aumentada pois **não usa *Proof of Work***. Utiliza *Proof of Stake* ou outro método...

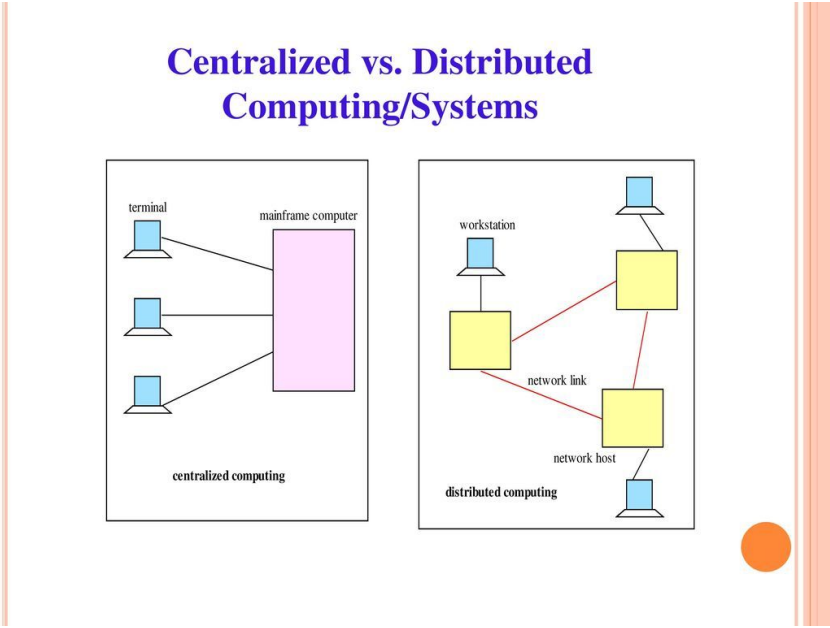
Consenso (do fim ao início)

- **Por que precisamos de protocolos de consenso?**
  - Para garantir a consistência da replicação
- Por que precisamos de replicação?
  - Para obter maior tolerância a falhas, disponibilidade e, sobretudo, escalabilidade
- Por que precisamos de escalabilidade?
  - Para atender mais clientes

Consenso (do início ao fim)

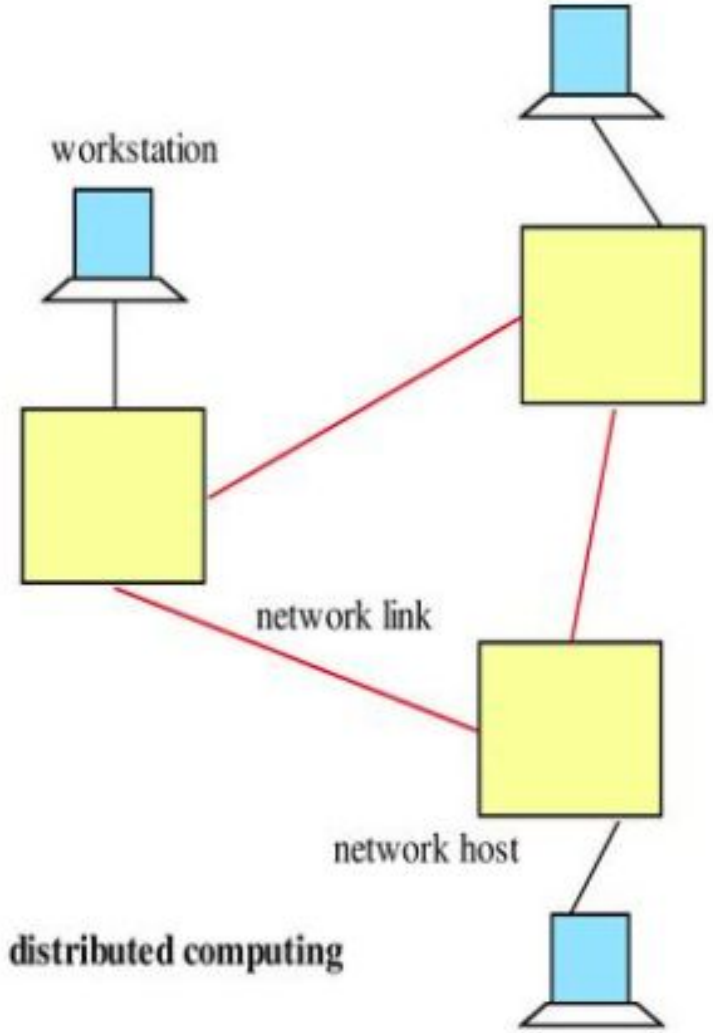
- Sistema centralizado
  - Ponto único de falha
  - Alto custo de expansão

- Sistema distribuído
  - Escalabilidade
  - Disponibilidade
  - Tolerância a falhas
  - Consistência?



Replicação e Consistência

- **Como garantir a consistência das cópias?**
- **O velho exemplo de transações na conta bancária...**
  - Saldo igual a 3
  - Saque de 5
  - Depósito de 2



Replicação e Consistência

- Ordenação de eventos dependentes
- Não estaríamos aqui conversando se existisse um relógio global

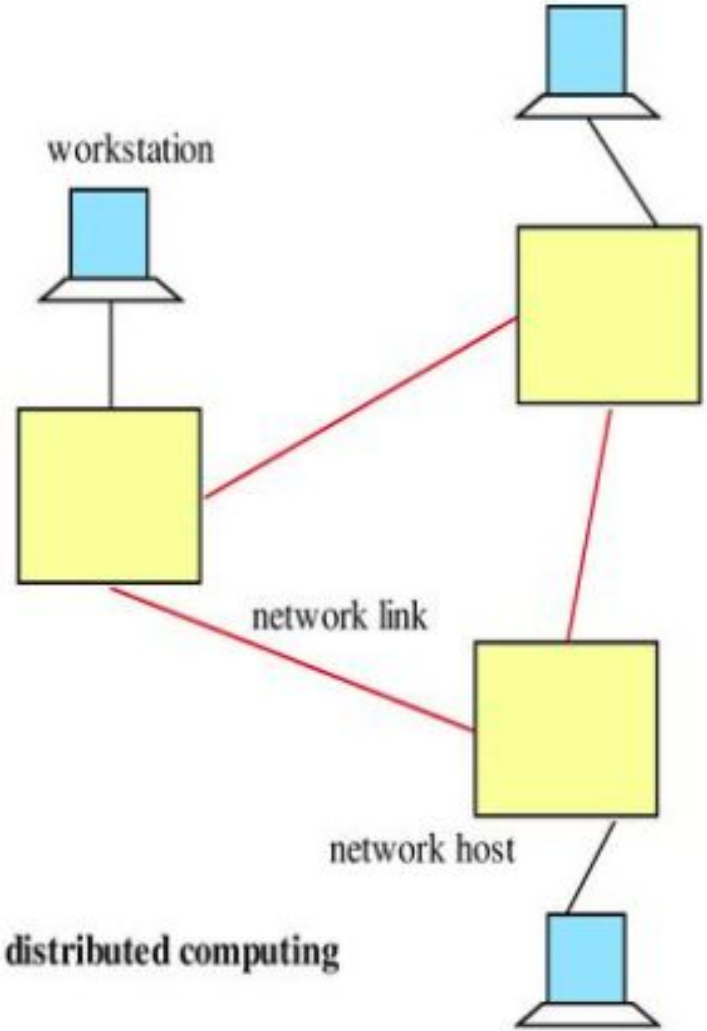
Replicação e Consistência

- Ordenação de eventos dependentes
- Não estaríamos aqui conversando se existisse um relógio global
- Alternativa: um emissor de *tickets*

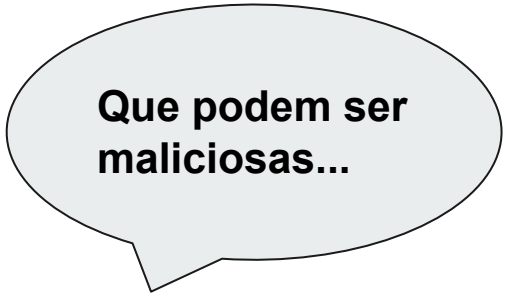
Emissor de *tíckets*

Problemas?

- PUF
- Escalabilidade



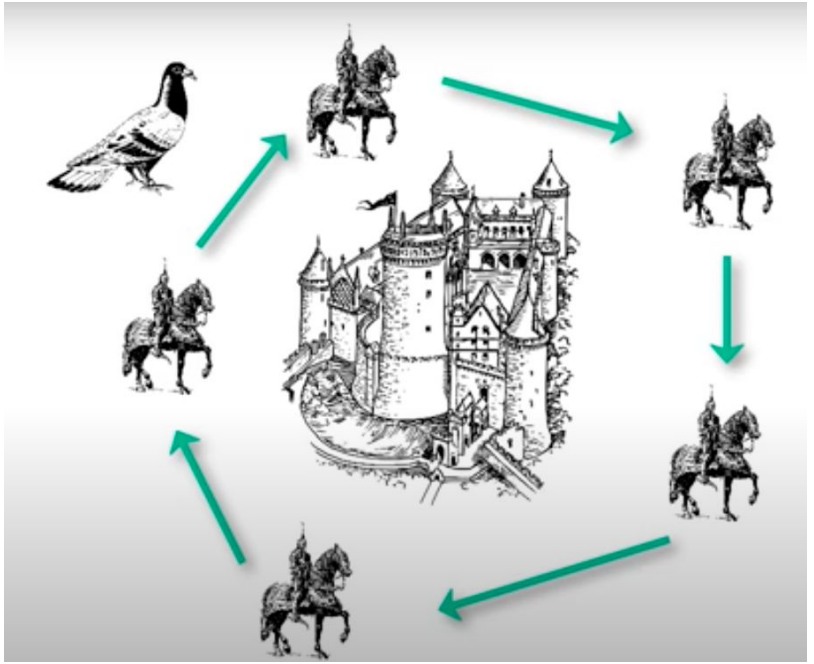
Teorema CAP (Brewer)



- **O teorema CAP diz que em caso de falhas eu preciso escolher entre consistência e disponibilidade**
- **Limites da replicação. Não posso ter 3 propriedades ao mesmo tempo:**
  - **C**onsistency: Every read receives the most recent write or an error
  - **A**vailability: Every request receives a (non-error) response – without guarantee that it contains the most recent write
  - **P**artition tolerance: The system continues to operate despite an arbitrary number of messages being dropped (or delayed) by the network between nodes

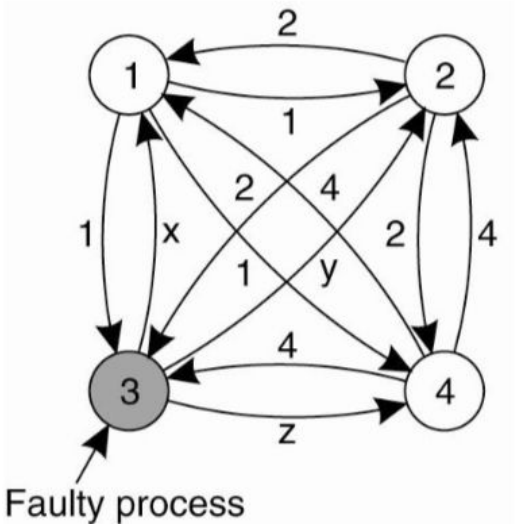
Problema dos Generais Bizantinos

O problema dos generais bizantinos é tomar uma decisão distribuída de atacar ou de recuar frente a possibilidade de alguns generais serem traidores e frente a possibilidade de mensagens entre eles se perderem.
O tempo para a decisão é finito.



Detecção e Tolerância a Falhas

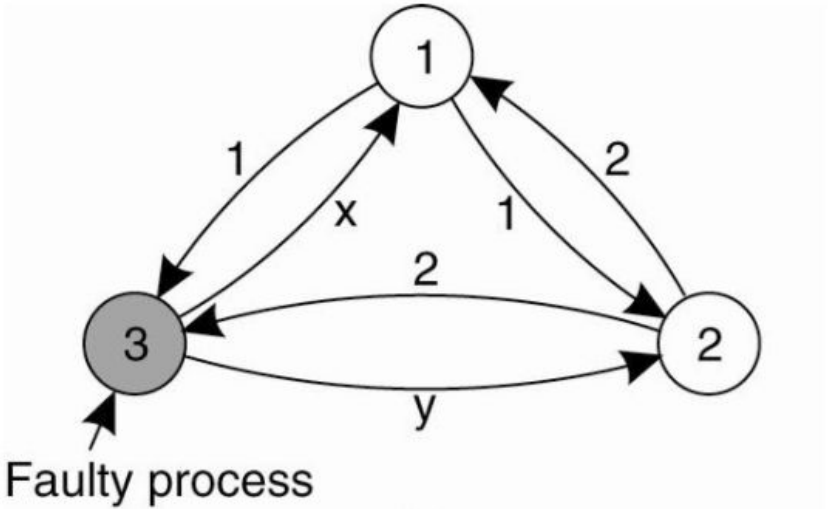
Fonte: Tanenbaum & Van Steen, Distributed Systems: Principles and Paradigms, 2e, (c) 2007



		1 Got	2 Got	4 Got
1	Got(1, 2, x, 4)	(1, 2, y, 4)	(1, 2, x, 4)	(1, 2, x, 4)
2	Got(1, 2, y, 4)	(a, b, c, d)	(e, f, g, h)	(1, 2, y, 4)
3	Got(1, 2, 3, 4)	(1, 2, z, 4)	(1, 2, z, 4)	(i, j, k, l)
4	Got(1, 2, z, 4)			

Detecção e Tolerância a Falhas

Fonte: Tanenbaum & Van Steen, Distributed Systems: Principles and Paradigms, 2e, (c) 2007



1 Got(1, 2, x)
2 Got(1, 2, y)
3 Got(1, 2, 3)

(b)

1 Got	2 Got
(1, 2, y)	(1, 2, x)
(a, b, c)	(d, e, f)

Detecção e Tolerância a Falhas

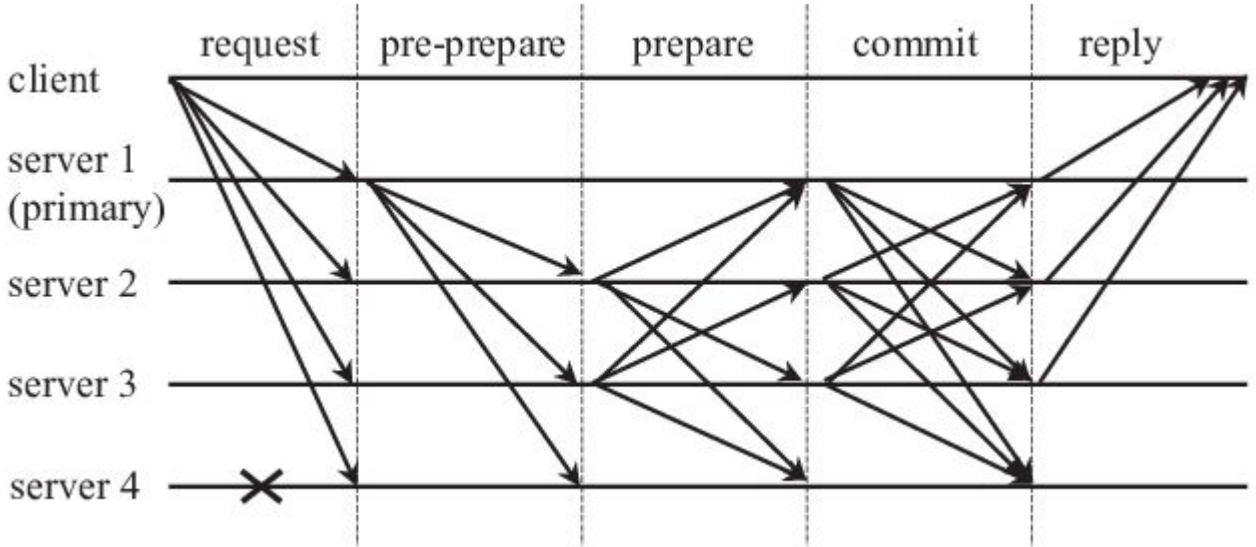
Fonte: Tanenbaum & Van Steen, Distributed Systems: Principles and Paradigms, 2e, (c) 2007

- Resultado forte:
  - É preciso *n* ou mais nós para tolerar *f* nós bizantinos:

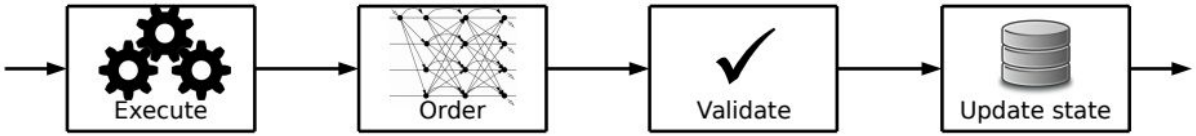
$$n >= 3*f+1$$

Algoritmos de Consenso

Practical Byzantine Fault Tolerance (**PBFT**)



Hyperledger Fabric (Androulaki et al. 2018)



- Simulate trans. and endorse
- Create rw-set
- Collect endorse-ments

- Order rw-sets
- Atomic broadcast (consensus)
- Stateless ordering service

- Validate endorse-ments & rw-sets
- Eliminate invalid and conflicting trans.

- Persist state on all peers

	avg	st.dev	99%	99.9%
(1) endorsement	5.6 / 7.5	2.4 / 4.2	15 / 21	19 / 26
(2) ordering	248 / 365	60.0 / 92.0	484 / 624	523 / 636
(3) VSCC val.	31.0 / 35.3	10.2 / 9.0	72.7 / 57.0	113 / 108.4
(4) R/W check	34.8 / 61.5	3.9 / 9.3	47.0 / 88.5	59.0 / 93.3
(5) ledger	50.6 / 72.2	6.2 / 8.8	70.1 / 97.5	72.5 / 105
(6) validation (3+4+5)	116 / 169	12.8 / 17.8	156 / 216	199 / 230
(7) end-to-end (1+2+6)	371 / 542	63 / 94	612 / 805	646 / 813

Consenso é uma atividade cara

	avg	st.dev	99%	99.9%
(1) endorsement	5.6 / 7.5	2.4 / 4.2	15 / 21	19 / 26
(2) ordering	248 / 365	60.0 / 92.0	484 / 624	523 / 636
(3) VSCC val.	31.0 / 35.3	10.2 / 9.0	72.7 / 57.0	113 / 108.4
(4) R/W check	34.8 / 61.5	3.9 / 9.3	47.0 / 88.5	59.0 / 93.3
(5) ledger	50.6 / 72.2	6.2 / 8.8	70.1 / 97.5	72.5 / 105
(6) validation (3+4+5)	116 / 169	12.8 / 17.8	156 / 216	199 / 230
(7) end-to-end (1+2+6)	371 / 542	63 / 94	612 / 805	646 / 813

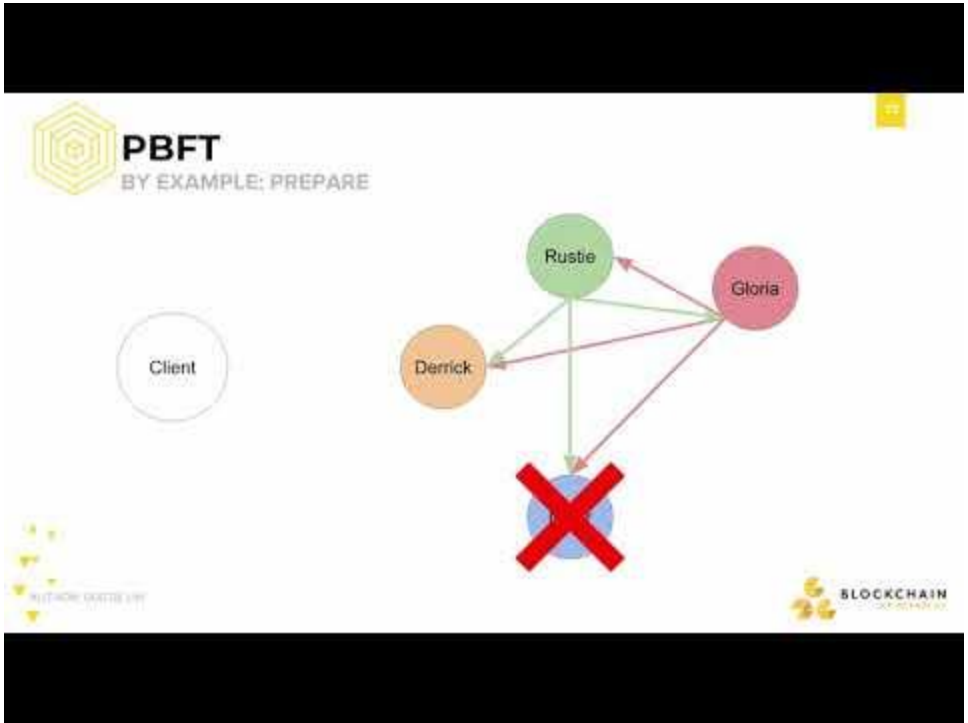
Table 1: Latency statistics in milliseconds (ms) for MINT and SPEND, broken down into five stages at a 32-vCPU peer with 2MB blocks. Validation (6) comprises stages 3, 4, and 5; the end-to-end latency contains stages 1–5.

EuroSys 2018

Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains

Elli Androulaki, Artem Barger, Vita Bortnikov — IBM
Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart — IBM
Christopher Ferris, Gennady Laventman, Yacov Manevich — IBM
Srinivasan Muralidharan*, Chet Murthy*, Binh Nguyen* — State Street Corp.
Manish Sethi, Gari Singh, Keith Smith, Alessandro Sorniotti — IBM
Chrysoula Stathakopoulou, Marko Vukolić, Sharon Weed Cocco, Jason Yellick — IBM

PBTF

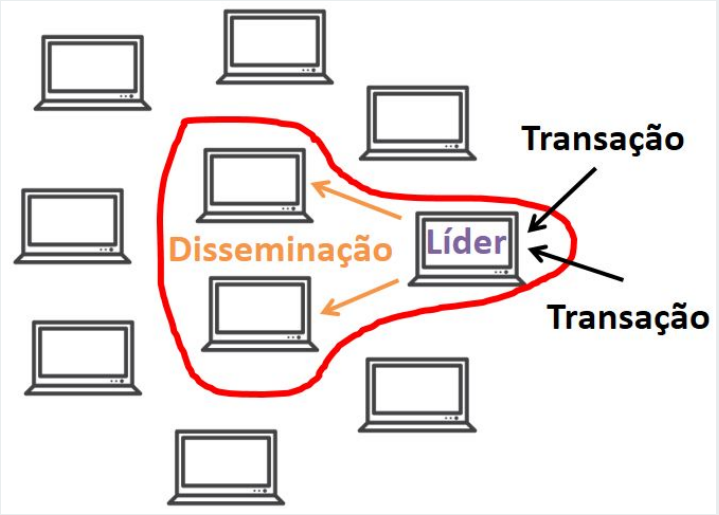

PBFT
BY EXAMPLE: PREPARE
Client
Rustie
Gloria
Derrick
BLOCKCHAIN

Como funciona: Consenso via Paxos

Como funciona: Consenso via Paxos

Como se adiciona um novo bloco no final da cadeia?

- **Alguns** participantes (bem definidos) serão encarregados por chegar ao acordo
- Escolha de um **líder** entre esses participantes
- Quando escolhido, o **líder** será o responsável por receber as transações, criar o bloco e disseminar essa informação aos outros.
- Não há recompensa associada nesse processo



Diferentes tipos de mecanismo de consenso

- Proof-of-work (PoW)
- Proof-of-stake (PoS)
  - Delegated-proof-of-stake (DPoS)
  - Proof-of-importance (PoI), Xem
- Proof-of-activity (PoA), PoW+PoS
- Proof-of-burn (PoB), gasta moedas existentes, Slimcoin
- Proof-of-deposit (PoD), lastro em moedas
- Proof-of-capacity (PoC), demonsta capacidade instalada, Burstcoin
- Proof-of-elapsed-time (PoET), Intel (considered reliable)

Source: Preprint, https://www.preprints.org/manuscript/201809.0136/v1 and https://www.coindesk.com/the-creator-of-proof-of-stake-thinks-he-finally-figured-it-out/

Resumo sobre Consenso

- Trata-se de um acordo que pode ocorrer mesmo na presença de falhas
- Não temos um relógio global, então precisamos de mecanismos de consenso...
- Estudado desde os anos 70, Tanenbaum e Lamport
- Necessário para obter escalabilidade
    - Teorema CAP (consistência, disponibilidade e tolerância ao particionamento)
- Algoritmos:
    - PBFT (Hyperledger), PAXOS, RAFT, PoW (bitcoin), PoS, PoET, etc.

Mais sobre modelos de Consenso

Livro Texto: https://sbseg2019.ime.usp.br/minicursos

Outros tópicos de interesse:

IOTA tangle

Solana Proof of History

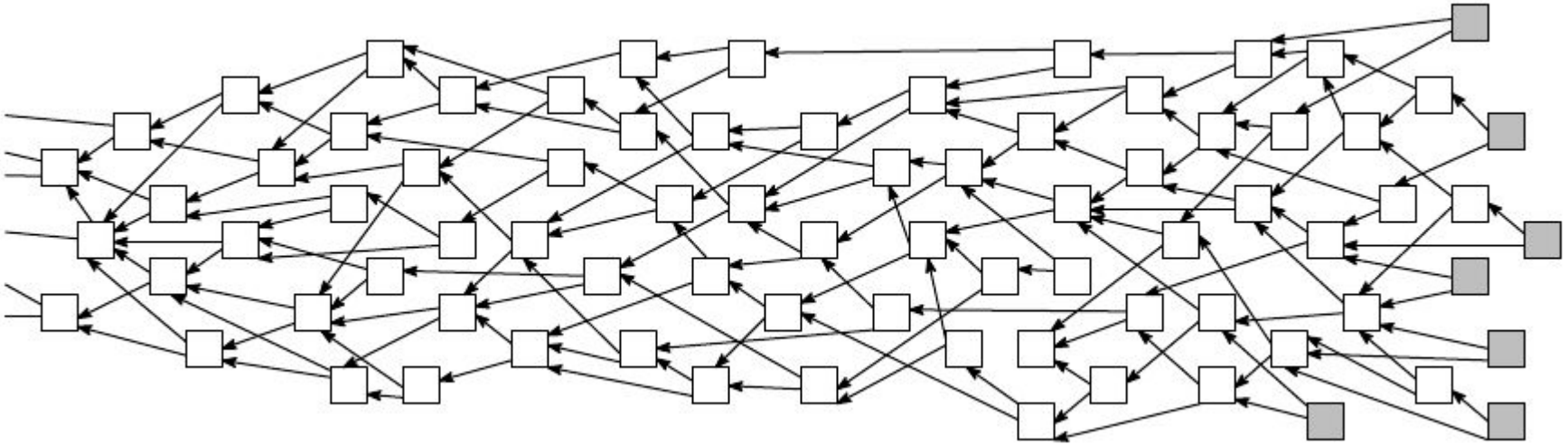
Swirlds Hashgraph

Avanços recentes

- Blockchain 1.0
  - Bitcoin
- Blockchain 2.0
  - Principal representante: Ethereum e Hyperledger
  - Contratos inteligentes
- Blockchain 3.0
  - Principais representantes: IOTA (Tangle) e Swirlds (Hashgraph)
    - Não baseados em Blockchain :-)
  - Melhor desempenho e mais segurança

IOTA Tangle

https://explorer.iota.org/mainnet/visualizer/



Hashgraph

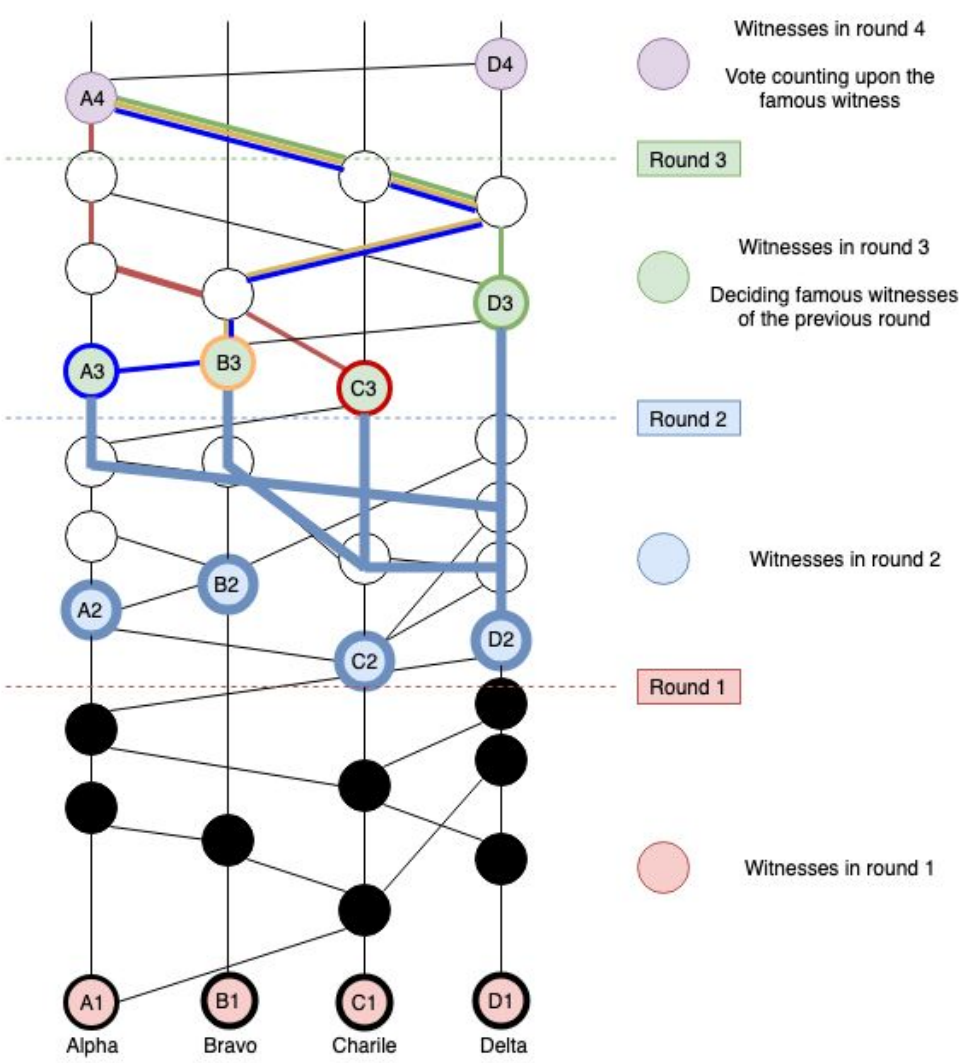
Baseado em fofocas

Patenteado

Dizem que suporta 100Ktps…

Comparativo

https://merehead.com/blog/difference-hedera-h

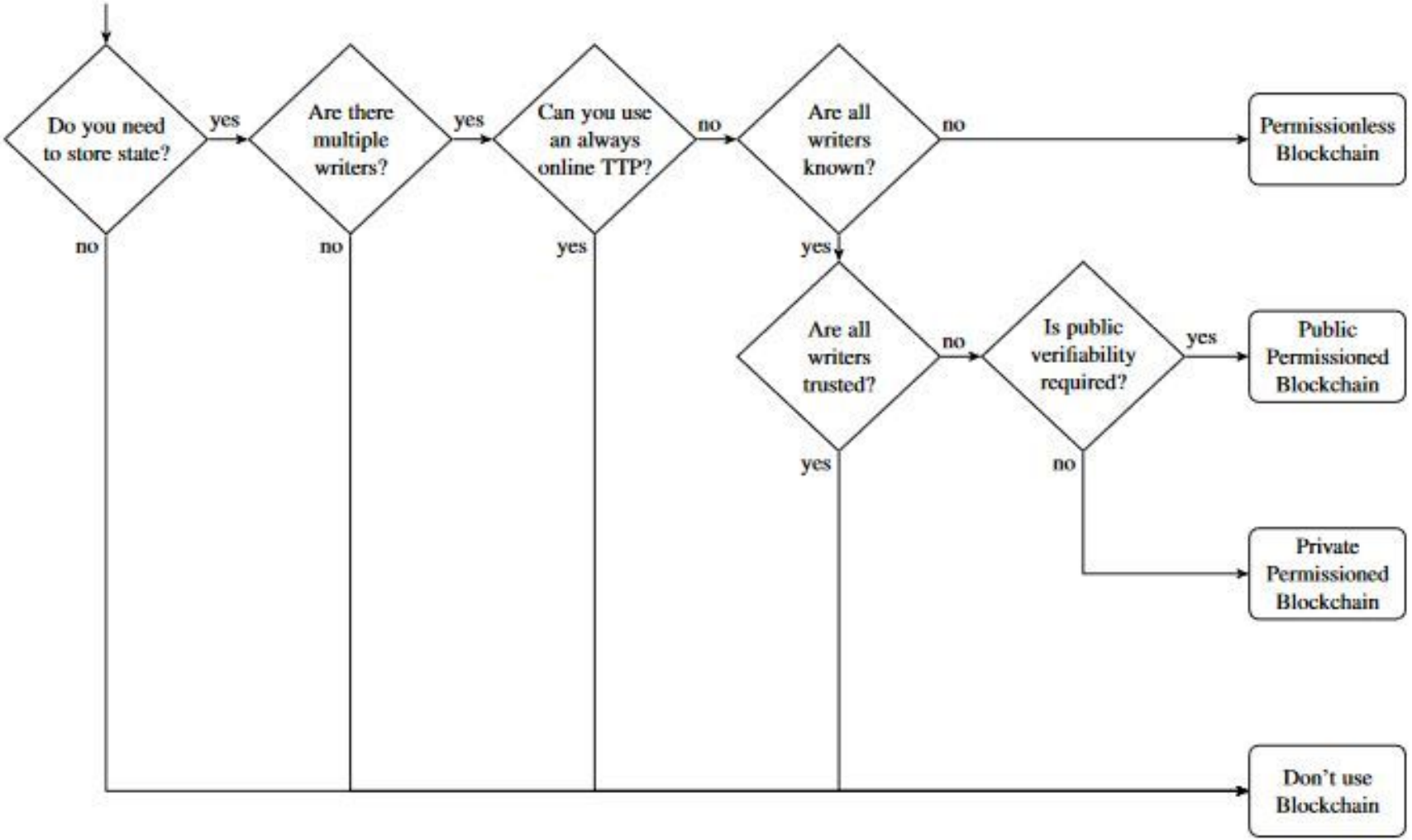


Fim sobre consenso

Antes de definir os projetos

Quando usar Blockchain?

- Fonte:
  Do you need a Blockchain? Karl Wüst, Arthur Gervais, *IACR Cryptology ePrint Archive*, 2017.


Do you need to store state?
yes
Are there multiple writers?
yes
Can you use an always online TTP?
no
Are all writers known?
no
Permissionless Blockchain
no
no
yes
yes
Are all writers trusted?
no
Is public verifiability required?
yes
Public Permissioned Blockchain
yes
no
Private Permissioned Blockchain
Don't use Blockchain