“In blockchains we trust (... n't?)”

Demystifying the Technology that Supports NFTs

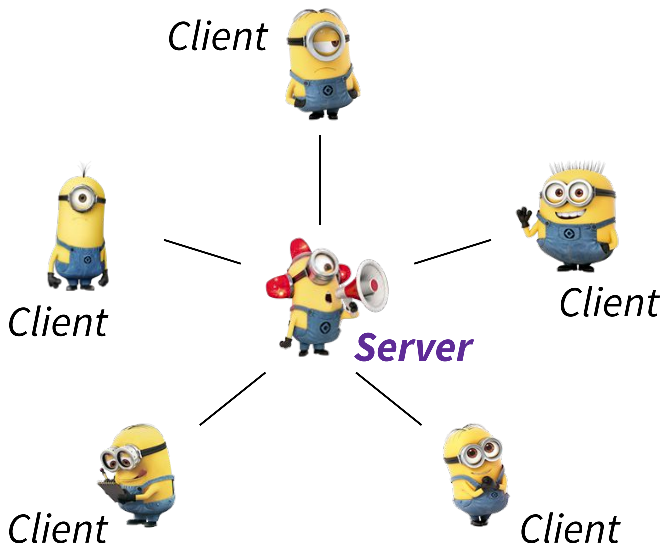
NYU 上海 SHANGHAI 纽约大学

Olivier Marin
Department of Computer Science & Engineering
https://wp.nyu.edu/omarin/

Trust Issues - Centralized Services

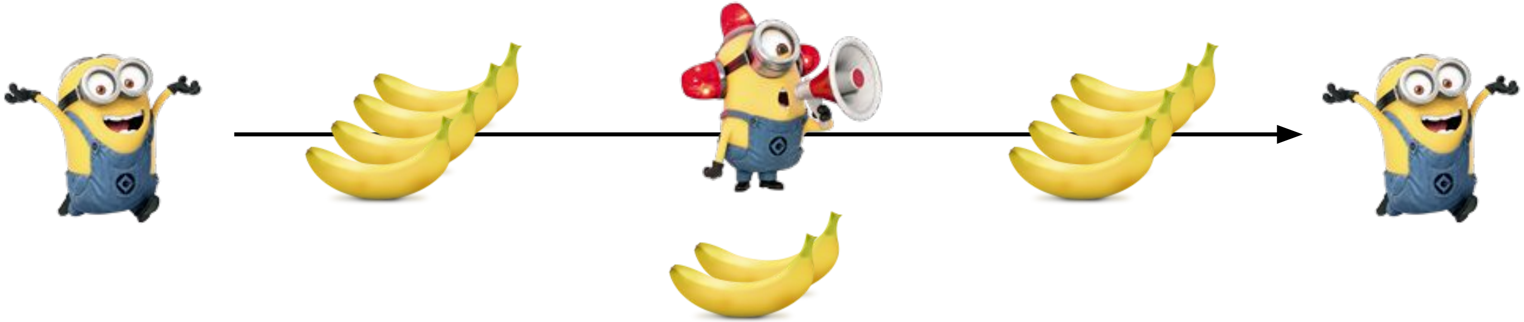
Billions of digital transactions every day

Traditional solutions rely on a **centralized** Trusted Third Party (TTP)



Trust Issues - Pros of Centralized TTPs

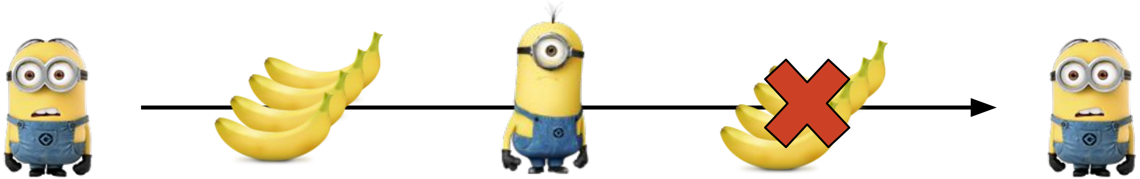
+ No other fully "reliable" solution
+ Simple to implement



Trust Issues - Cons of Centralized TTPs

- Single point of failure

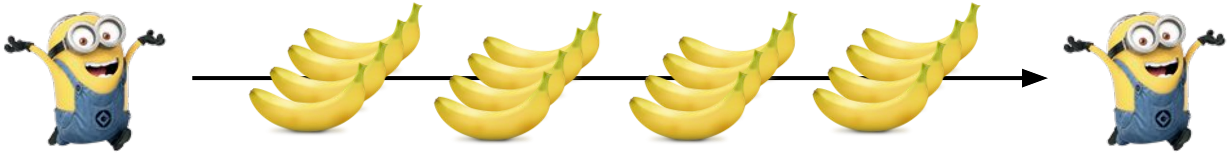
- Privacy



- Service is *never free*

Trust Issues - Cutting out the Middleman

Direct transaction



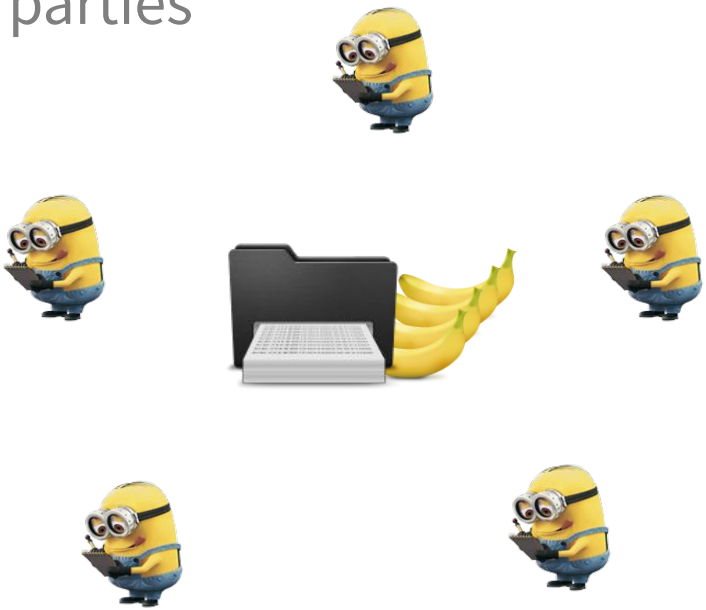
Digital world offers many opportunities for cheating

- ➢ Post-payment denial
- ➢ Double spending
- ➢ Theft (identity, ownership)
- ➢ ...

Trust Issues - Distributed Ledger

Decentralized solution

- Share a common ledger among all parties



Blockchains are all the hype

Ingredients of a Blockchain

Cryptographic hashing

- Unique, verifiable keys
- Means for detecting that data has been tampered with

Peer-to-peer system

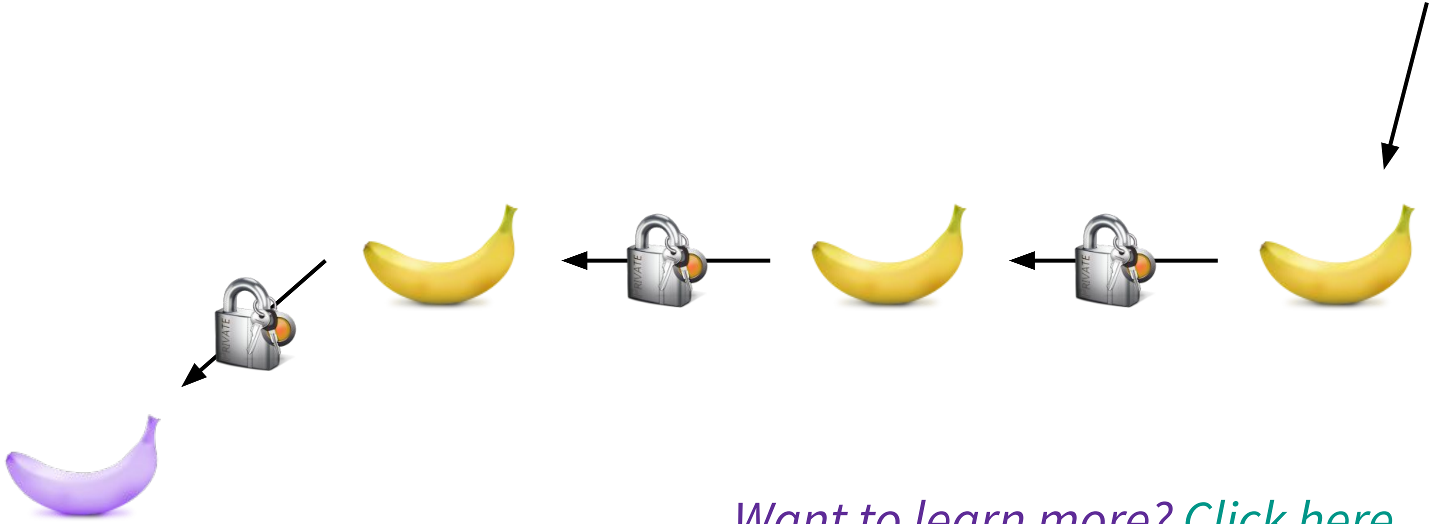
- Scalable, dynamic network
- Means for sharing information on a massive scale

Consensus algorithm

- Decentralized consistency protocol
- Means for agreeing on a common decision

Cryptographic Hashing

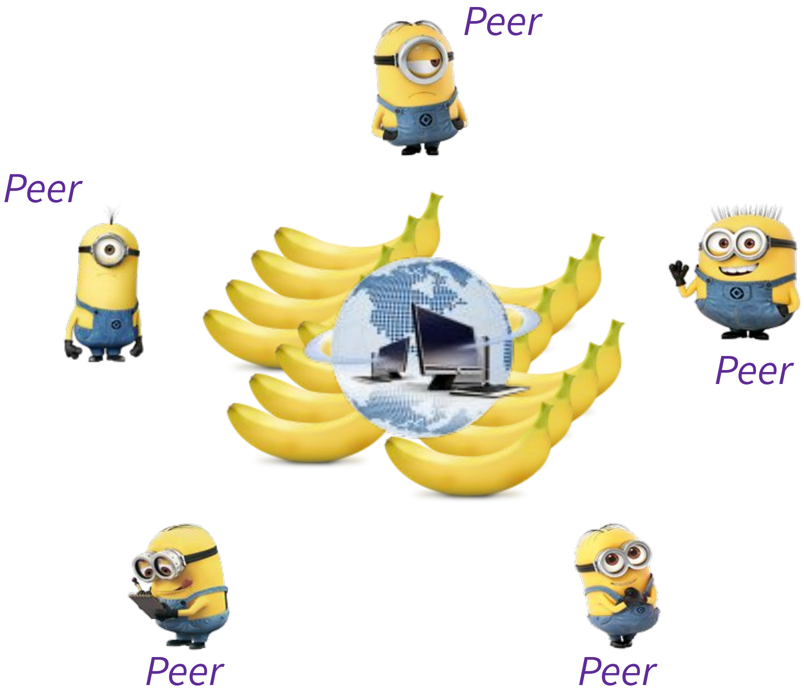
Linked list of transactions with hash pointers => "**Block Chain**"



Genesis

Want to learn more? [Click here]

Peer-to-Peer (P2P) Systems



The Small-World Problem

Experiment (Travers & Milgram, 1967) [1]

Letter sent to 150 random subjects in the midwest of the USA (Nebraska and Kansas)

Envelope lists summary information about a recipient on the east coast

Destination is in Cambridge, Massachusetts

Goal: get the letter to its recipient

Rules of transmission

- info. on the envelope indicates possible connection to the recipient
- only passed on to close acquaintances (first-name basis)
- write identity of every forwarder on the envelope (prevents loops)

The Small-World Problem

Results

Average number of hops = **6** (between 2 and 10)

~200 million inhabitants at the time => **fully scalable**

Based on network of acquaintances => **no centralization**

Malicious transmission only restarts the route => **reliable**

One major downside: this best-effort policy holds no guarantees!

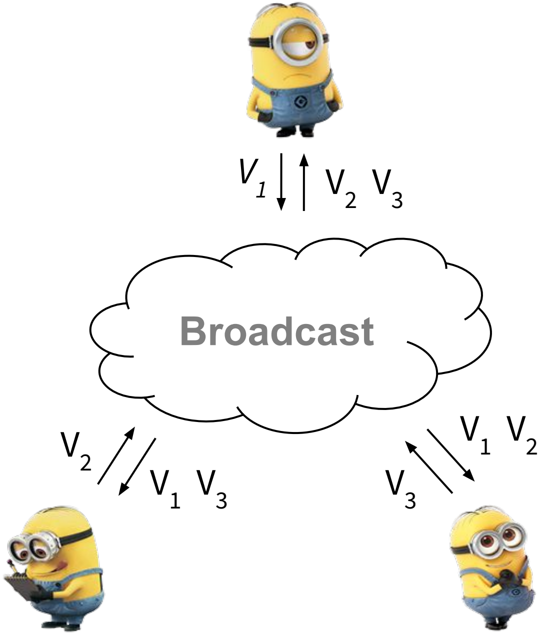
Consensus

Everyone pitches in towards the addition of a block/transaction

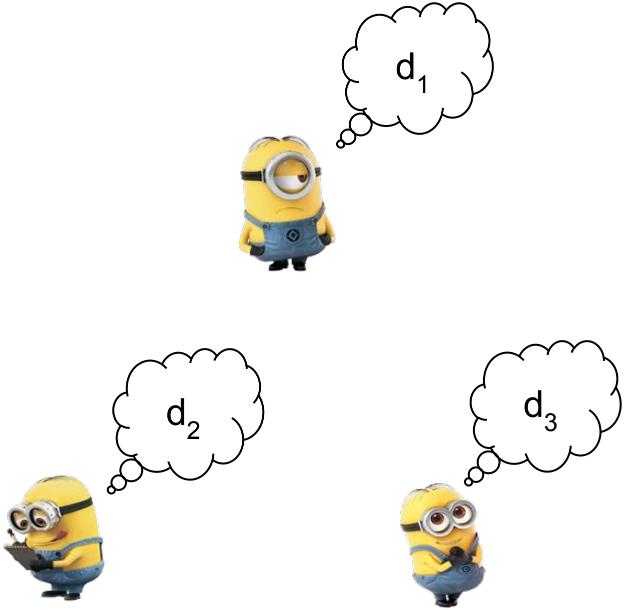
Computer nodes agree on:

- the validity of blocks
- which block gets added next

Anatomy of a Consensus



Step 1: Propose



Step 2: Decide

Properties of a Consensus

Termination

All processes eventually decide on a result ***D***

Validity

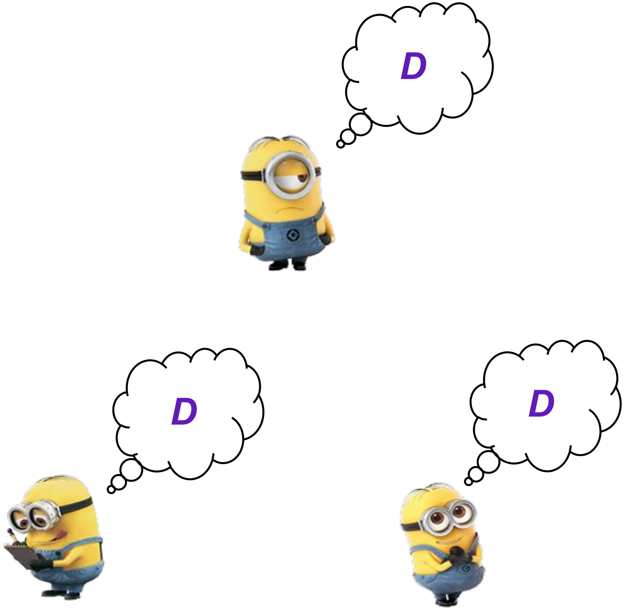
D is a proposed value

Agreement

D is the same for all processes

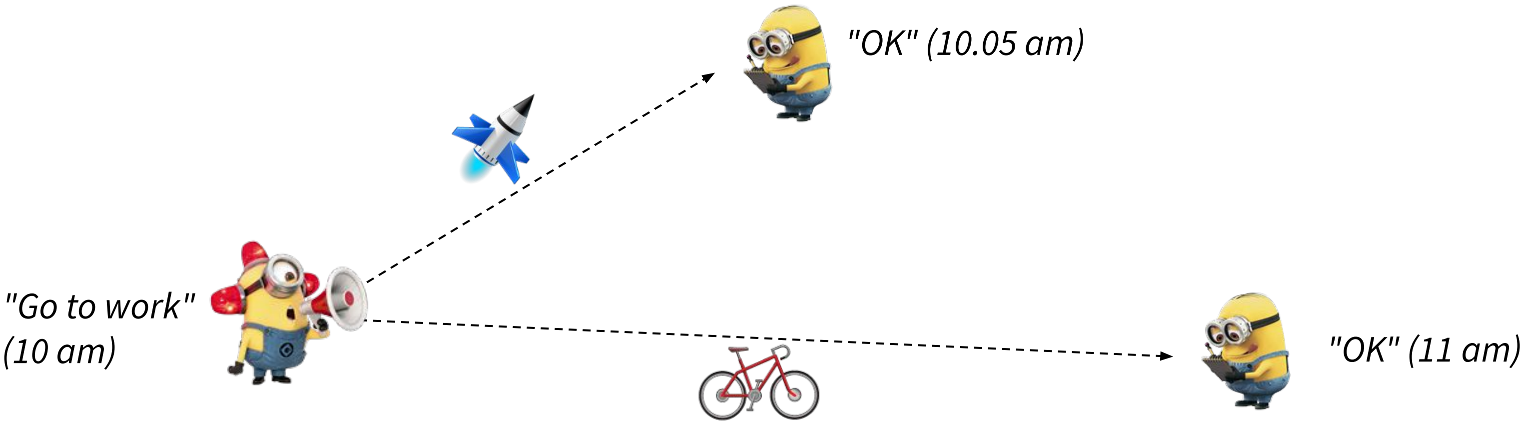
Integrity

Once a process decides ***D***, it cannot switch to ***D'***



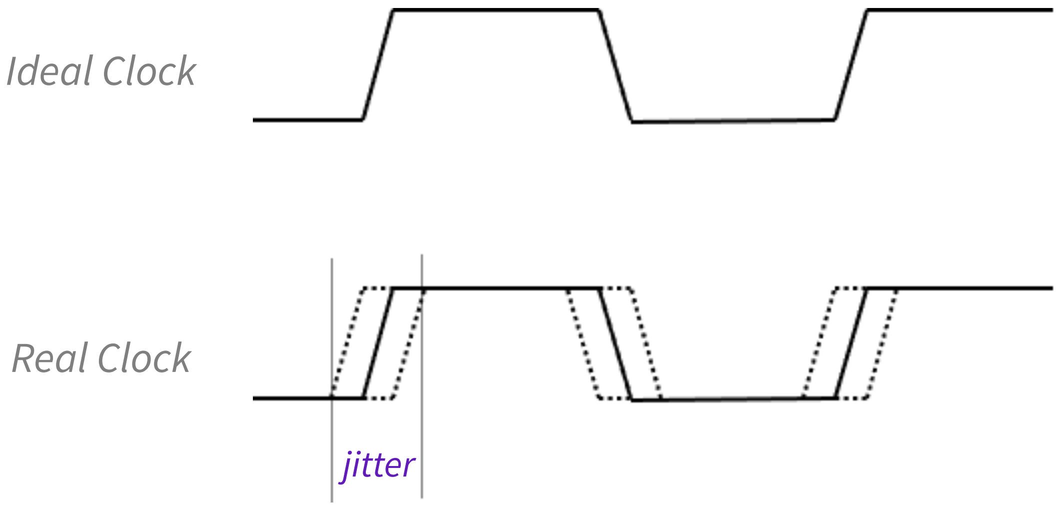
Time is Relative!

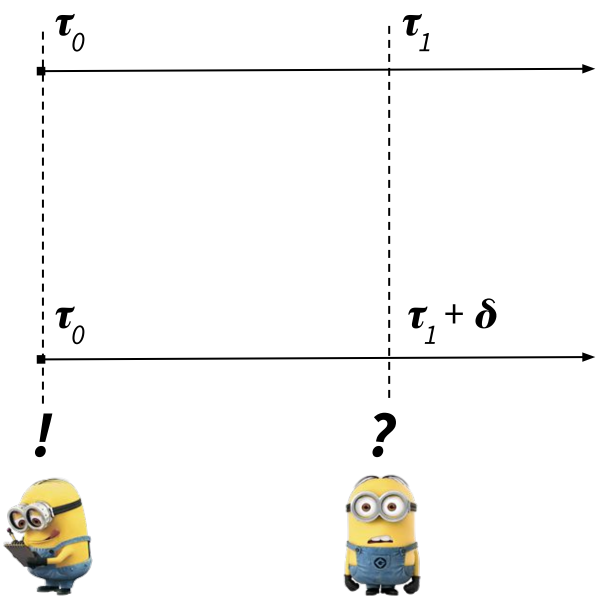
Clock skew



Time is Relative!

Clock jitter





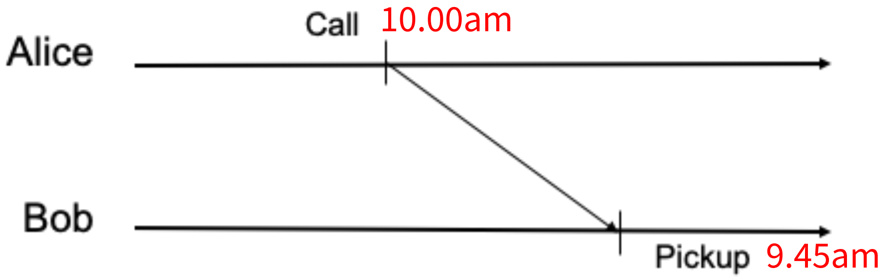
Time is Relative!

Jitter and skew induce clock drift

The drift varies independently for every node

Clock drifts may break causality

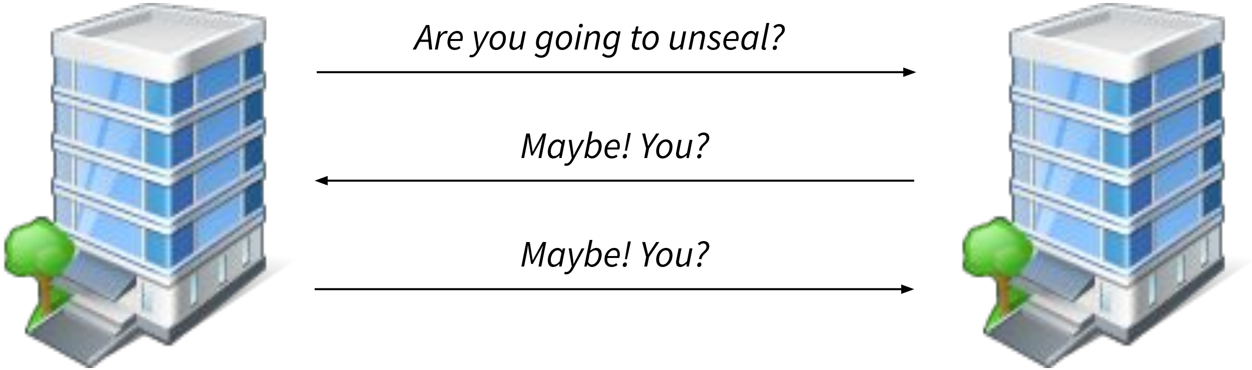
Alice and Bob both wake up at the same time in the real world
At that point their clocks display 8am
Alice's clock drifts forward very fast; Bob's clock drift is marginal
Alice is meant to call Bob at 10 am…



How to (Un)Seal a Deal

The Lockdown Conundrum (aka the *Two Generals' Problem* [3])

- A neighborhood with 2 buildings is in lockdown
- No new cases in the last 7 days
- Each building has a manager, both want to avoid risks at all costs
- They communicate by WeChat to decide whether they should lift the lockdown

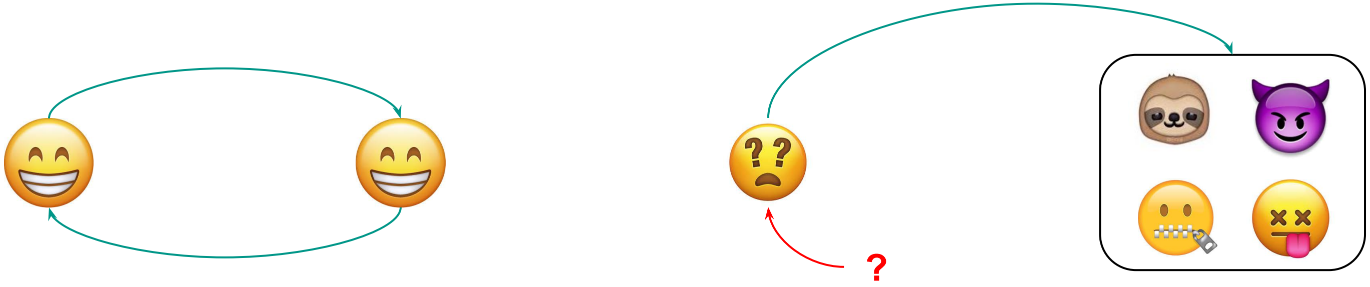


The Internet Can't Exist: Here's Why!

Scaling paradox

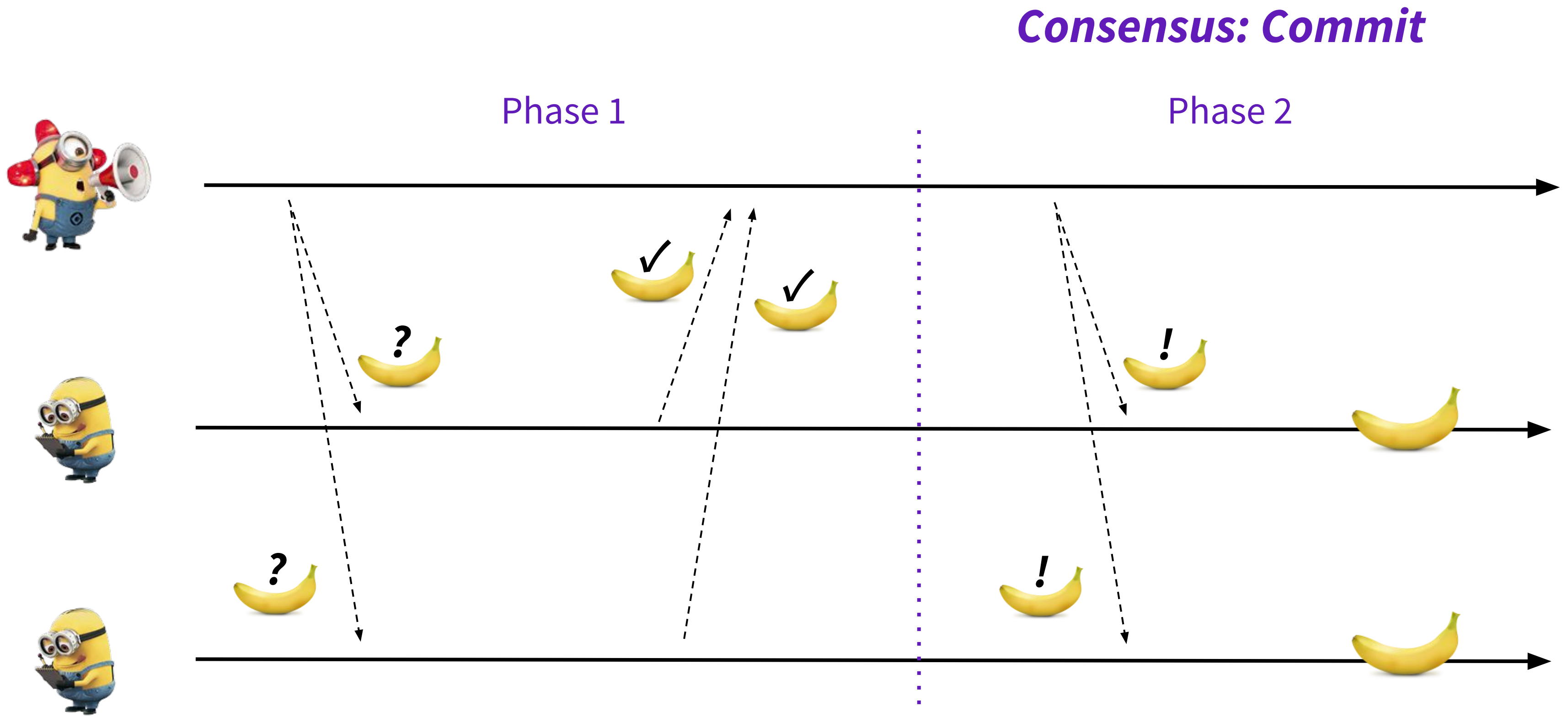
1. Probability of failures increases with the number of nodes
2. Dead nodes can bring down the entire system

Theoretical proof that two nodes may never reach consensus [4]

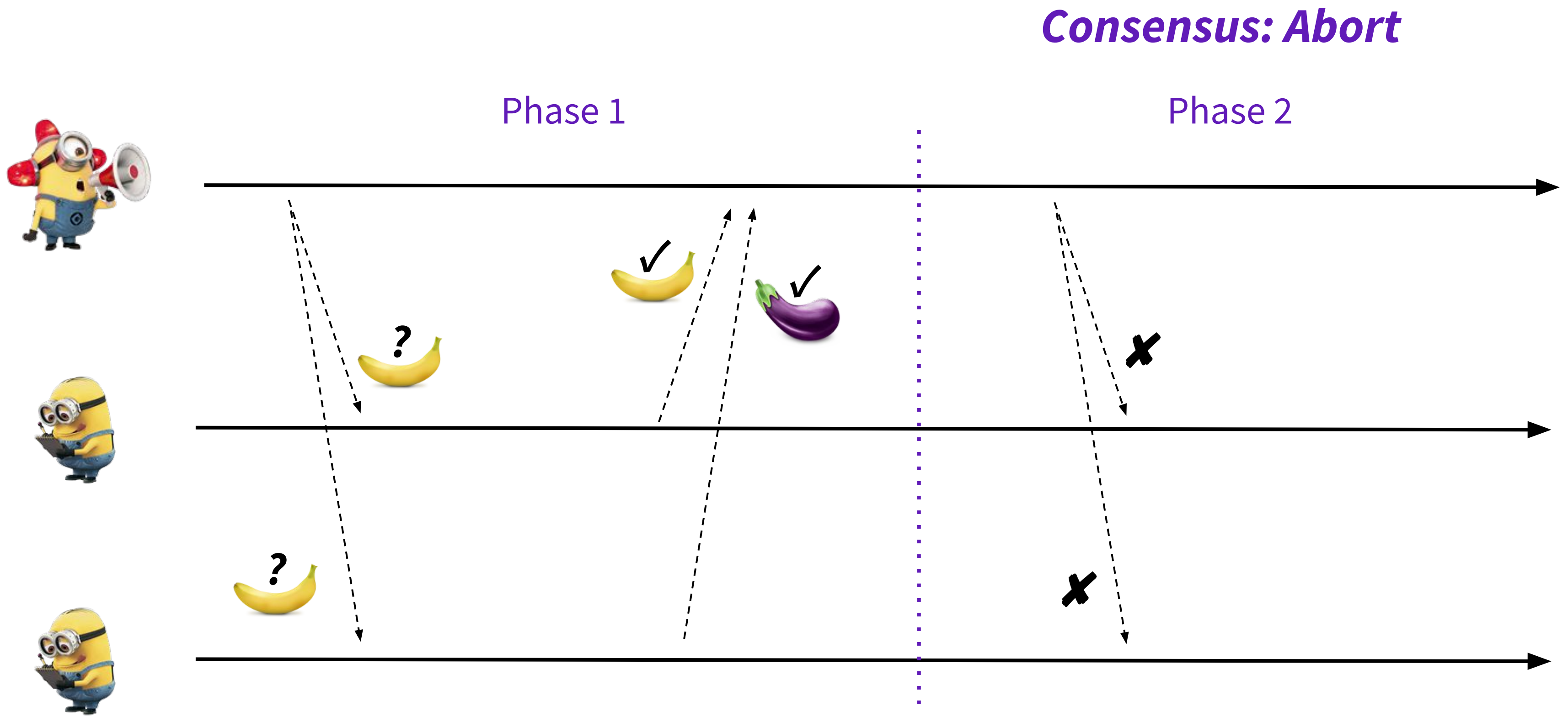


Solution: unreliable failure detectors [6]

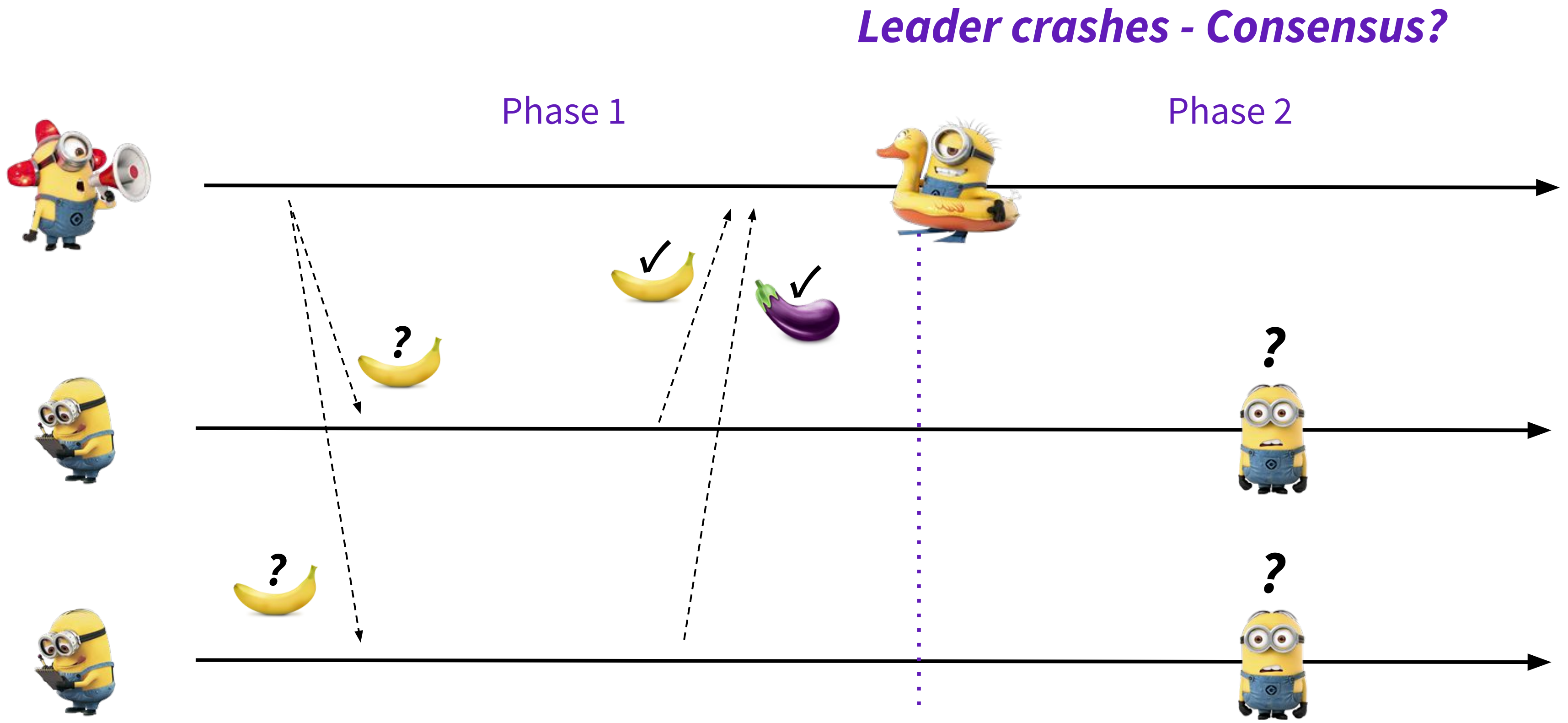
Two-Phase Commit



Two-Phase Commit

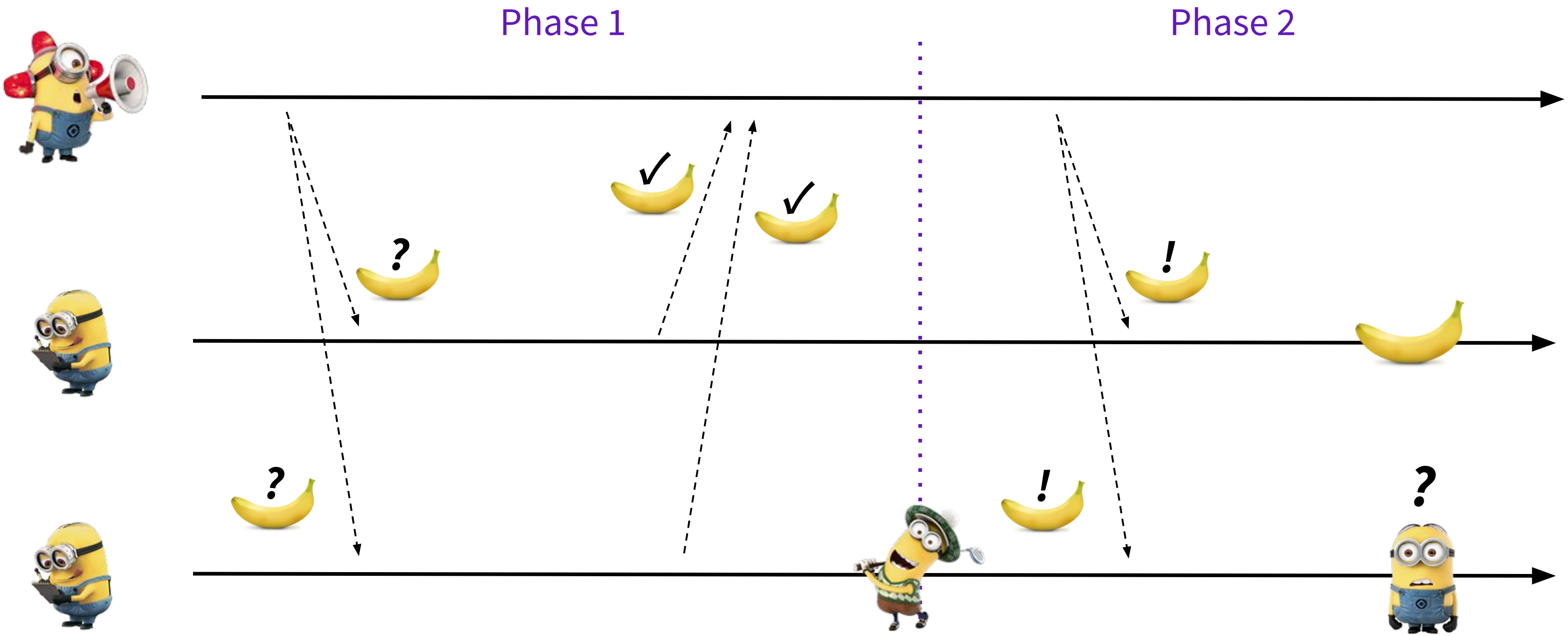


Two-Phase Commit



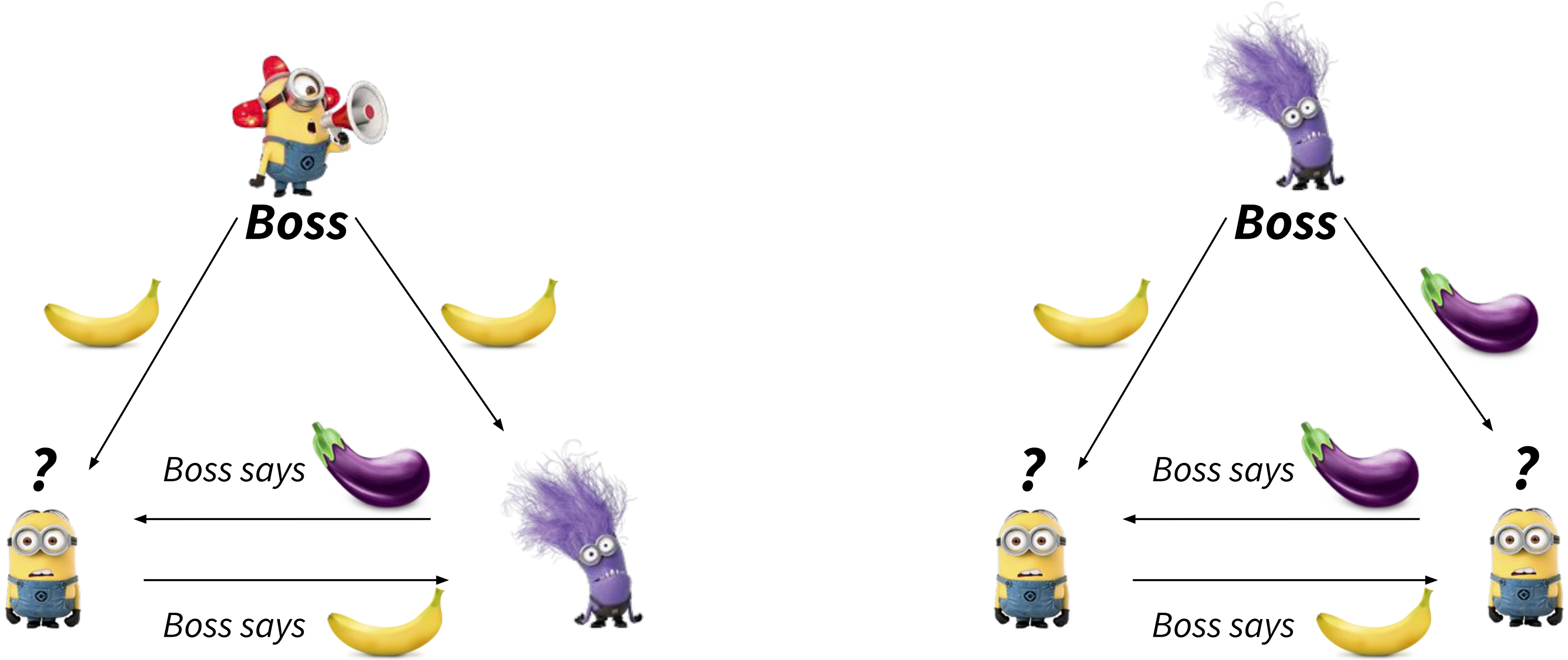
Two-Phase Commit

Follower crashes - Consensus?



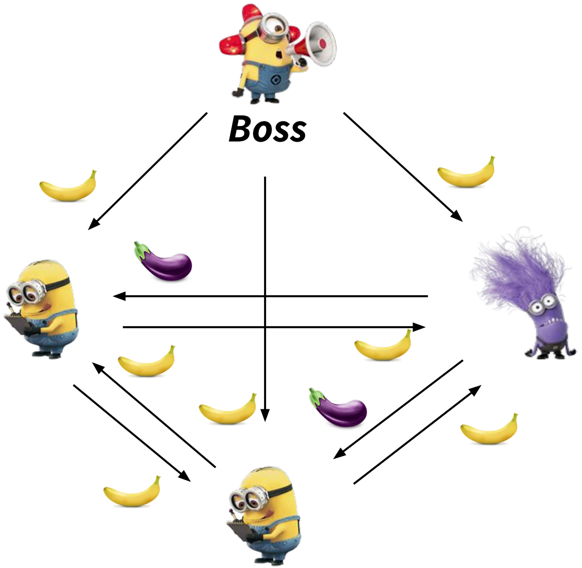
Consensus with byzantine failures

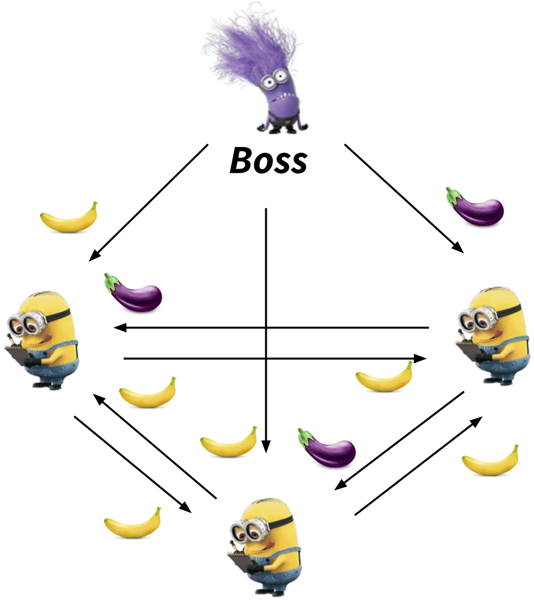
Impossibility: solving byzantine generals with 3 processes [5]



Consensus with byzantine failures

4-process solution for byzantine generals [5]





Takeaway points

None of the blockchain components are 100% reliable!

They can never be!

Failures are… ***inevitable***!

Paradoxically, reliability is found in numbers

A majority of honest nodes ensures safety and liveness!

References

[1] J. Travers, S. Milgram. An Experimental Study of the Small World Problem. Sociometry, 32(4). 1969. pp. 425–443.

[2] S. Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. 2009.

[3] E. A. Akkoyunlu, K. Ekanadham, and R. V. Huber. Some constraints and tradeoffs in the design of network communications. SIGOPS Operating Systems Review 9(5). 1975. pp. 67–74.

[4] M. Fischer, N. Lynch, and M. Paterson. Impossibility of Distributed Consensus with One Faulty Process. Journal of the ACM 32(2). 1985. pp. 374-382.

[5] L. Lamport, R. Shostak, and M. Pease. The Byzantine Generals Problem. ACM Transactions on Programming Languages and Systems 4(3). 1982. pp. 382–401.

[6] T. D. Chandra and S. Toueg. Unreliable failure detectors for reliable distributed systems. Journal of the ACM 43(2). 1996. pp. 225–267.