



Planetary Scale Systems

Matteo Monti

Distributed Computing Laboratory


Ecole Polytechnique Fédérale de Lausanne

Distributed Algorithms Course - 30/11/2020



Summary

Randomized algorithms
hold a promise for
planetary scale systems.

- 
- The origin.** Consensus-less Asset Transfer
 - The problem.** Byzantine Reliable Broadcast
 - The intuition.** Quorums vs. Samples
 - The challenge.** Games and Decorators
 - The future.** Towards a Planetary Database

Consensus-less Asset Transfer



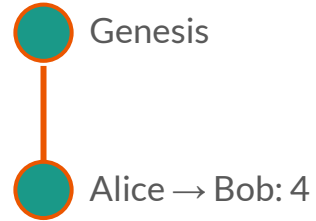
Asset Transfer (Atomic Broadcast)

Alice	5
Bob	6
Carl	0
Dave	3



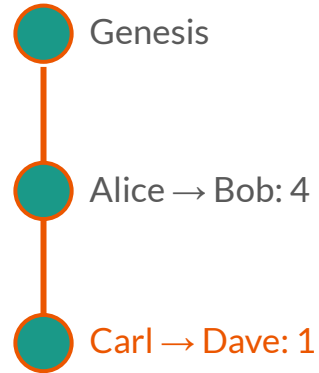
Asset Transfer (Atomic Broadcast)

Alice	1
Bob	10
Carl	0
Dave	3



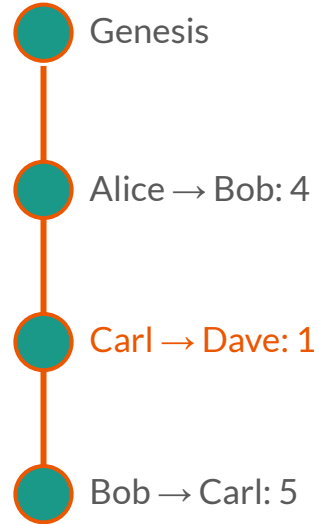
Asset Transfer (Atomic Broadcast)

Alice	1
Bob	10
Carl	0
Dave	3



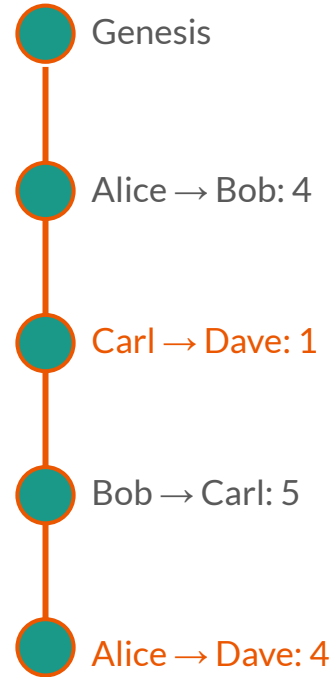
Asset Transfer (Atomic Broadcast)

Alice	1
Bob	5
Carl	5
Dave	3



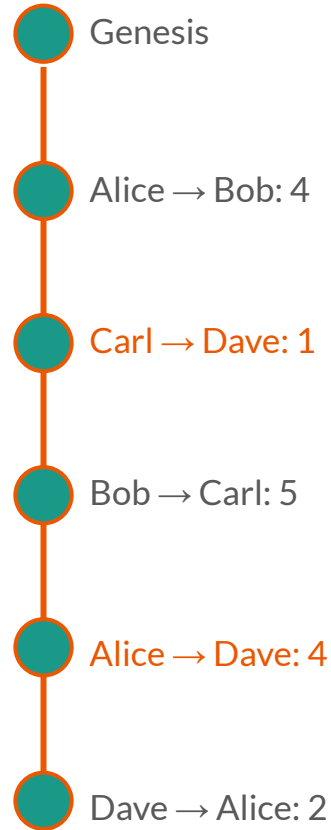
Asset Transfer (Atomic Broadcast)

Alice	1
Bob	5
Carl	5
Dave	3



Asset Transfer (Atomic Broadcast)

Alice	3
Bob	5
Carl	5
Dave	1



Asset Transfer (Source-Order Broadcast)



Alice	5	Carl	0
Bob	6	Dave	3

Asset Transfer (Source-Order Broadcast)



Alice	1	Carl	0
Bob	6	Dave	3

Asset Transfer (Source-Order Broadcast)



Alice	1	Carl	0
Bob	10	Dave	3

Asset Transfer (Source-Order Broadcast)



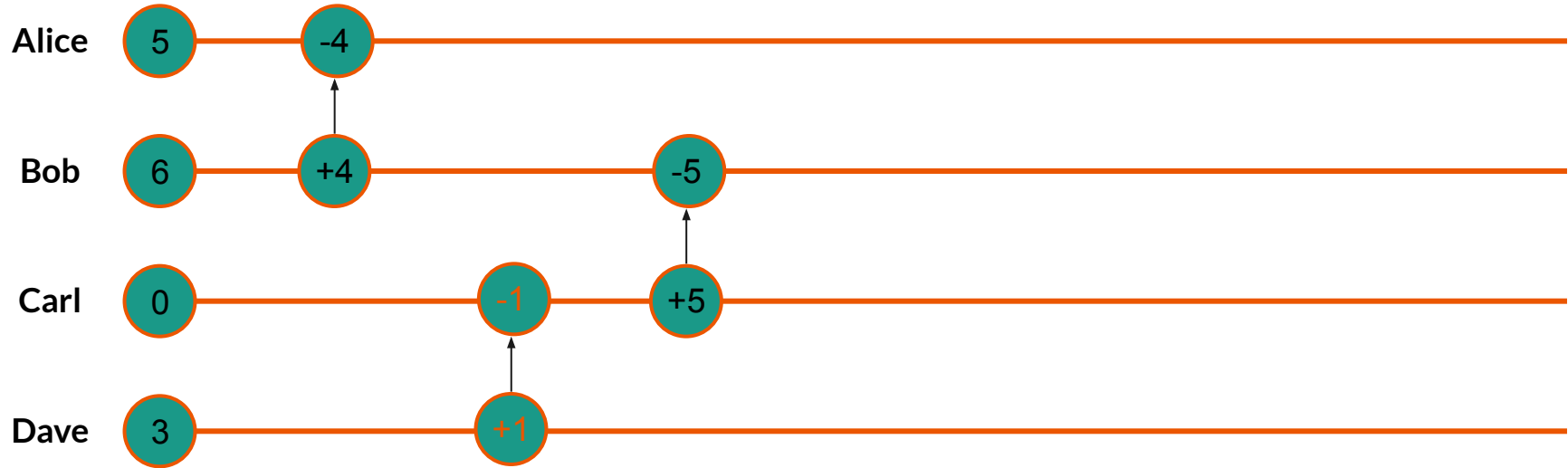
Alice	1	Carl	0
Bob	10	Dave	3

Asset Transfer (Source-Order Broadcast)



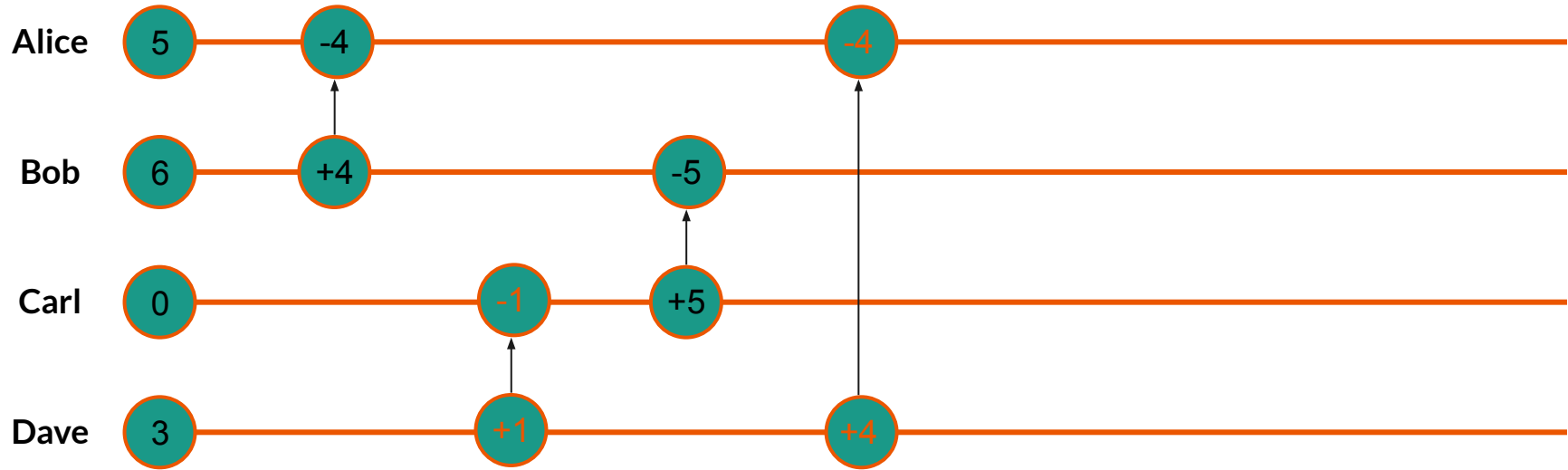
Alice	1	Carl	0
Bob	10	Dave	3

Asset Transfer (Source-Order Broadcast)



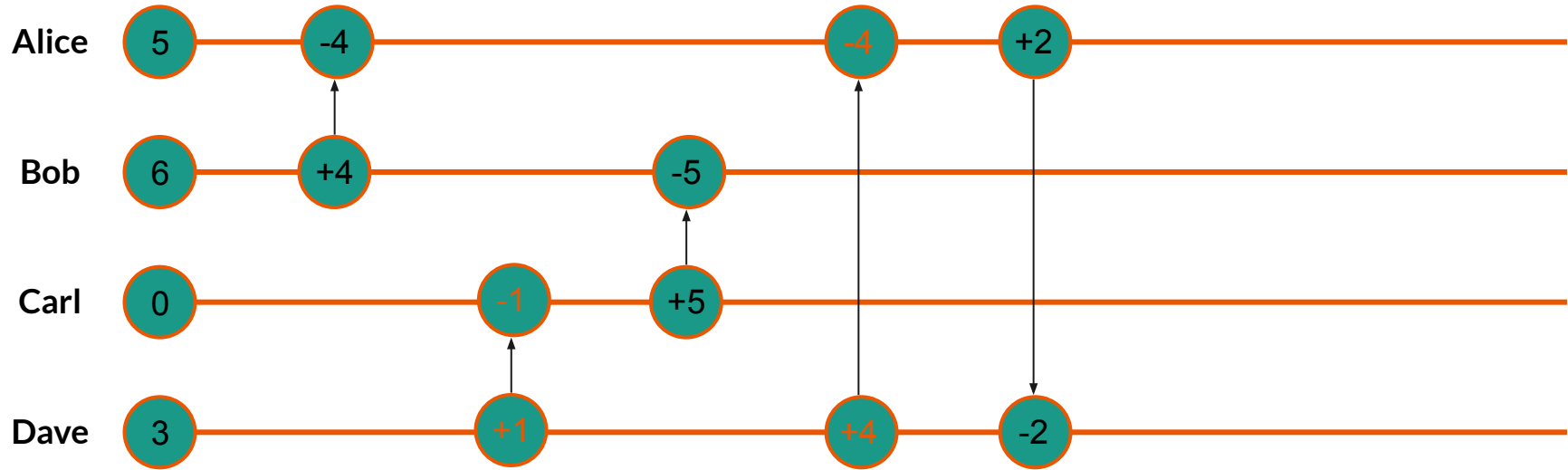
Alice	1	Carl	5
Bob	5	Dave	3

Asset Transfer (Source-Order Broadcast)



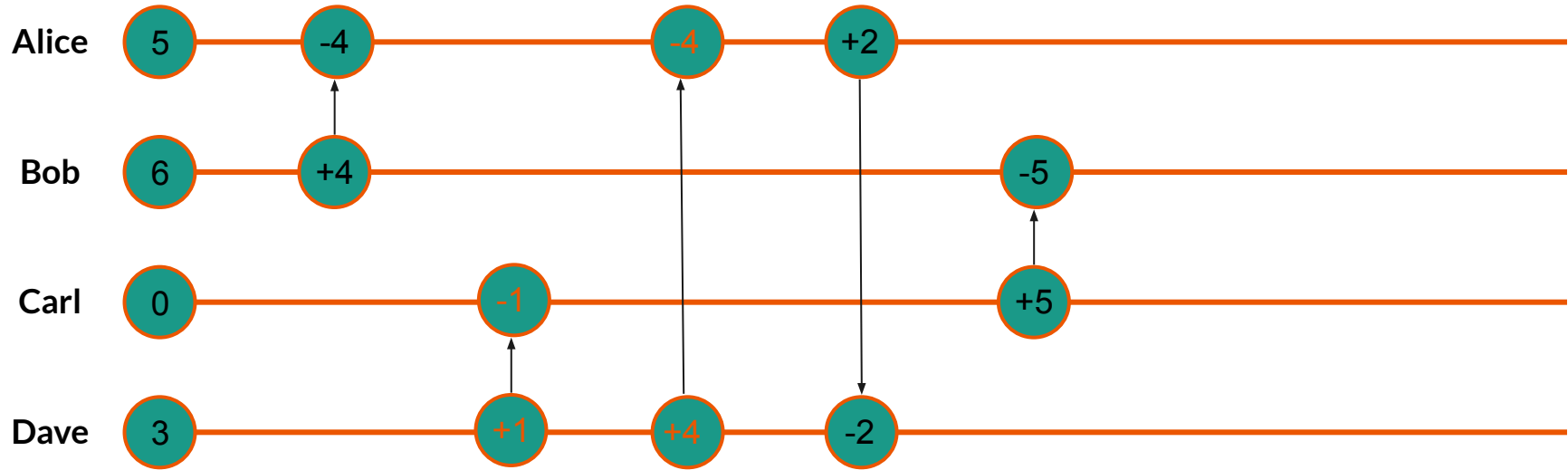
Alice	1	Carl	5
Bob	5	Dave	3

Asset Transfer (Source-Order Broadcast)



Alice	3	Carl	5
Bob	5	Dave	1

Asset Transfer (Source-Order Broadcast)



Alice	3	Carl	5
Bob	5	Dave	1

Solving Source-Order Broadcast

=

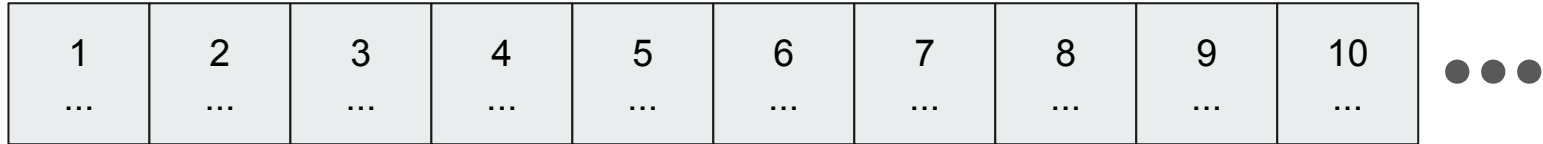
Solving Asset Transfer★

[★] Guerraoui, Rachid, et al. "The Consensus Number of a Cryptocurrency"
Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, 2019.

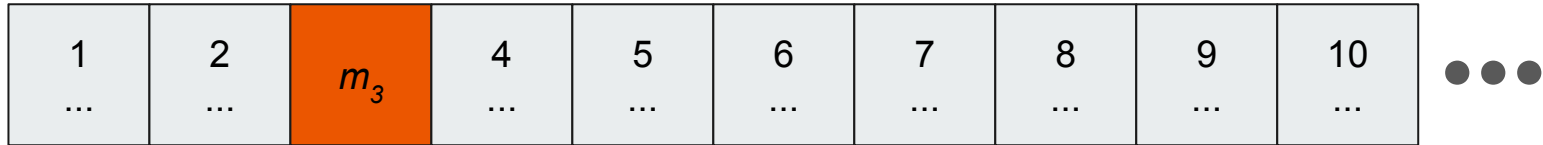
Per-source FIFO



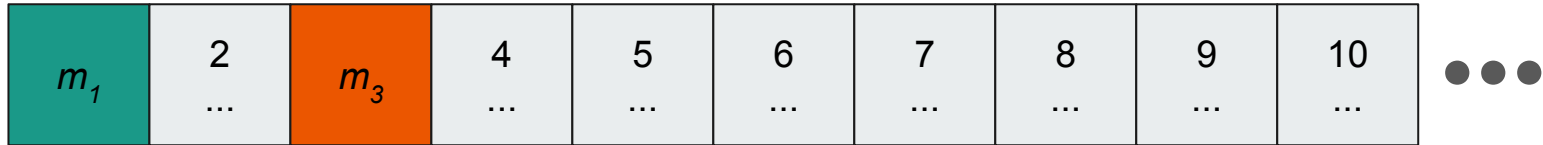
Reliable to FIFO



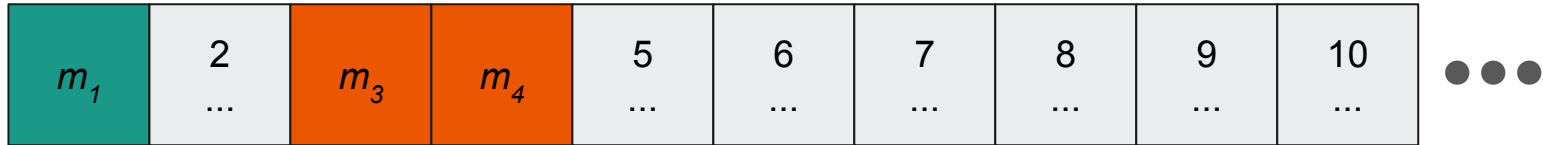
Reliable to FIFO



Reliable to FIFO



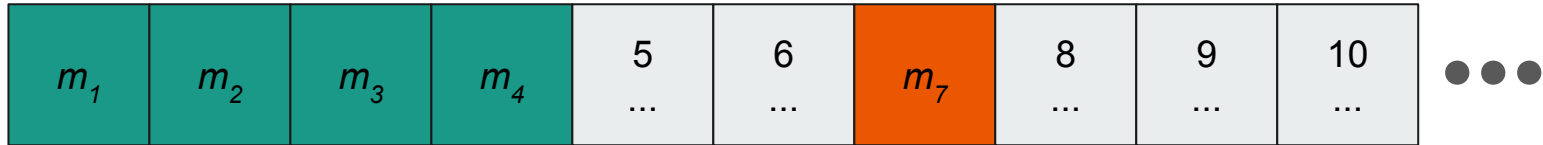
Reliable to FIFO



Reliable to FIFO



Reliable to FIFO



Solving Reliable Broadcast

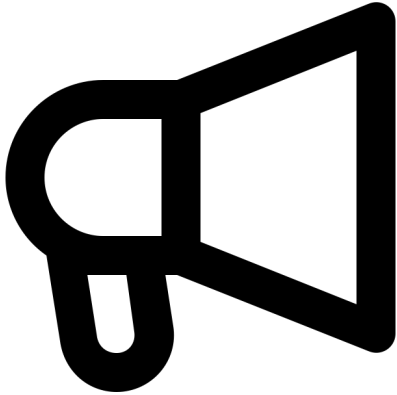
=

Solving Asset Transfer

Byzantine Reliable Broadcast

Interface

Broadcast
Request

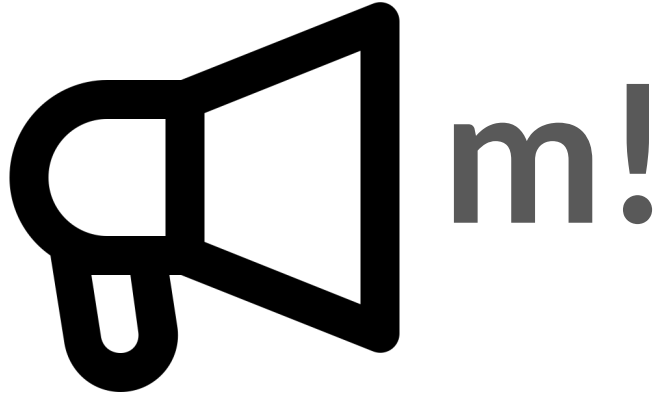


Deliver
Indication



Interface

Broadcast
Request

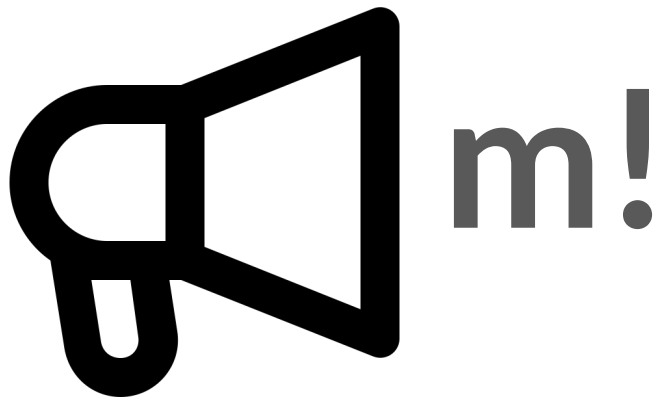


Deliver
Indication



Interface

Broadcast
Request



Deliver
Indication





Properties

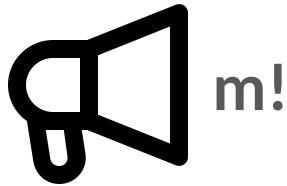
Deterministic case

Validity + **Consistency** + **Totality**

Validity

Consistency

Totality

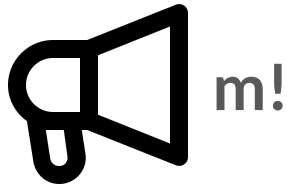


If the sender is correct, every correct process delivers its message.

Validity

Consistency

Totality

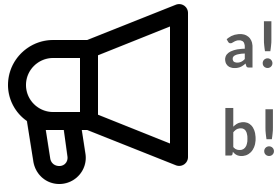


If the sender is correct, every correct process delivers its message.

Validity

Consistency

Totality

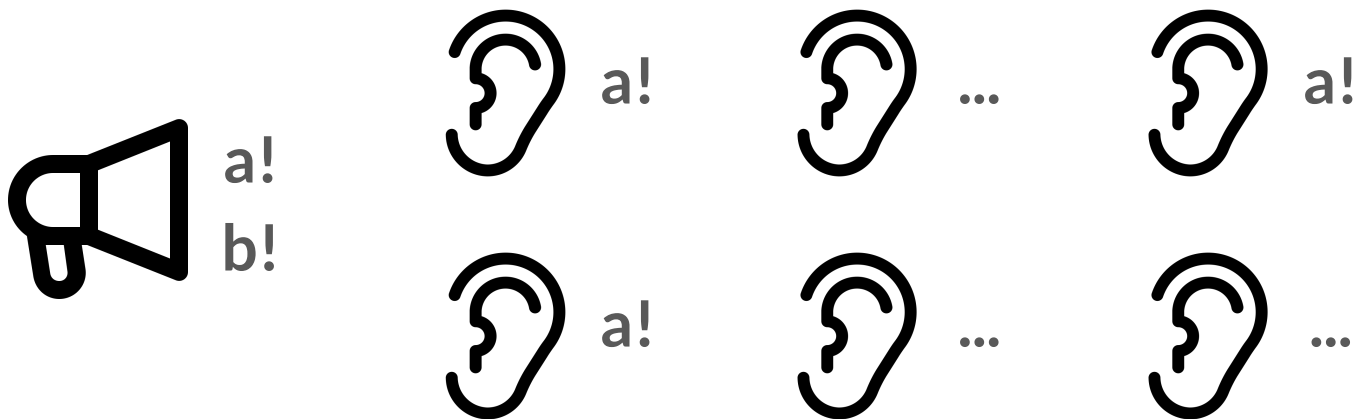


If two correct processes deliver a message, they deliver the same message.

Validity

Consistency

Totality

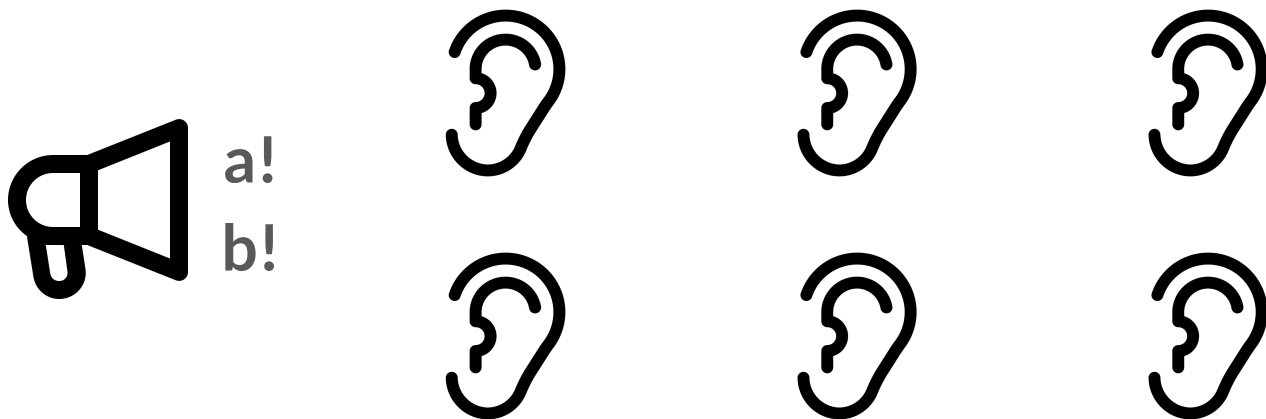


If two correct processes deliver a message, they deliver the same message.

Validity

Consistency

Totality

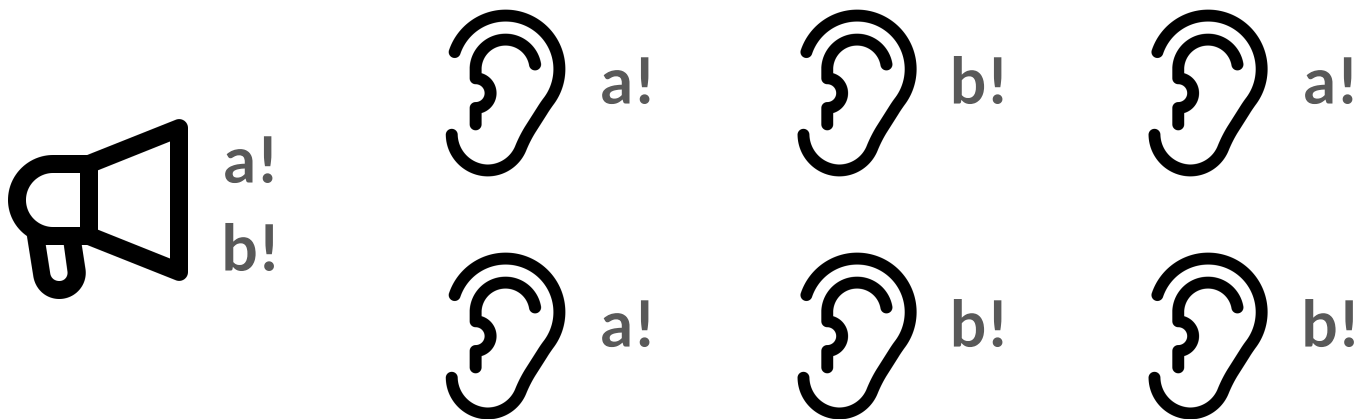


Either every correct process delivers a message, or no correct process delivers a message.

Validity

Consistency

Totality

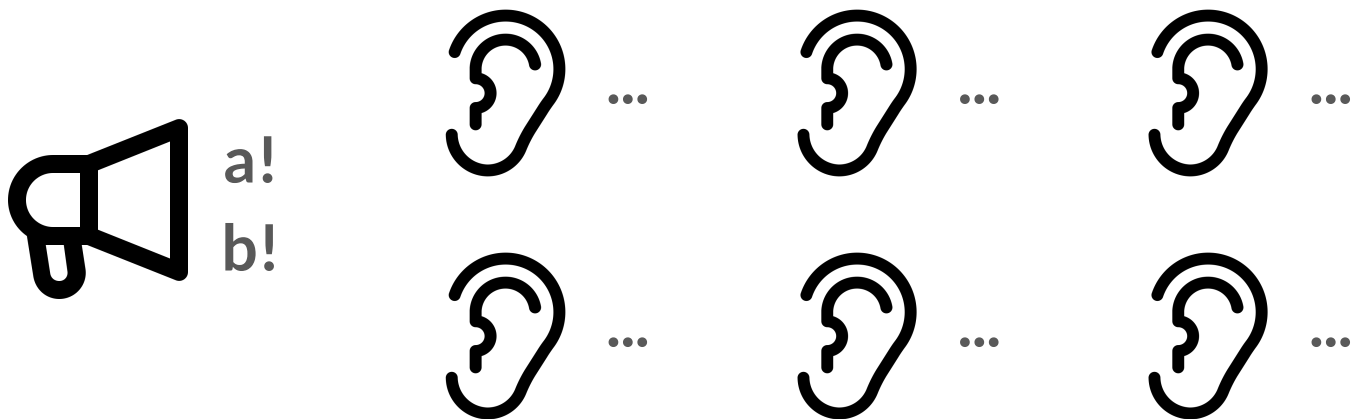


Either every correct process delivers a message, or no correct process delivers a message.

Validity

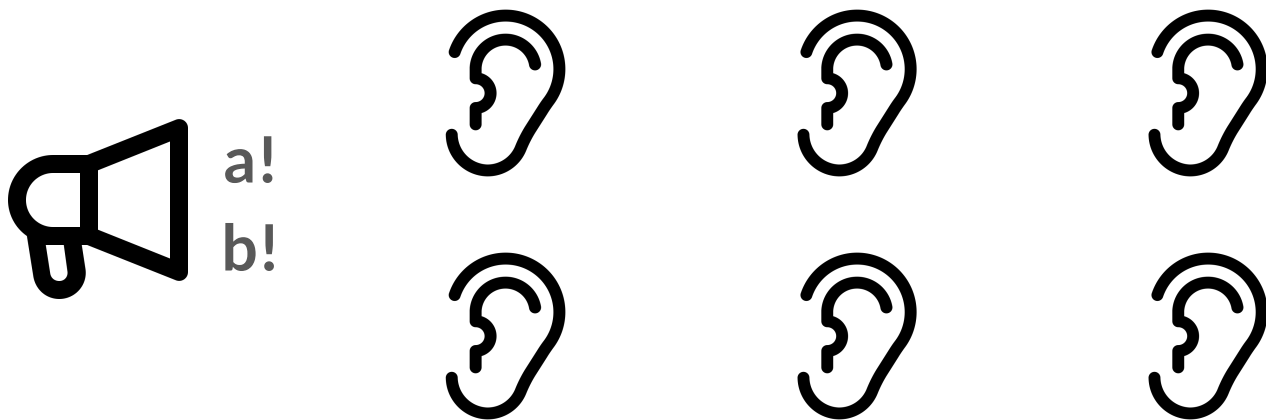
Consistency

Totality



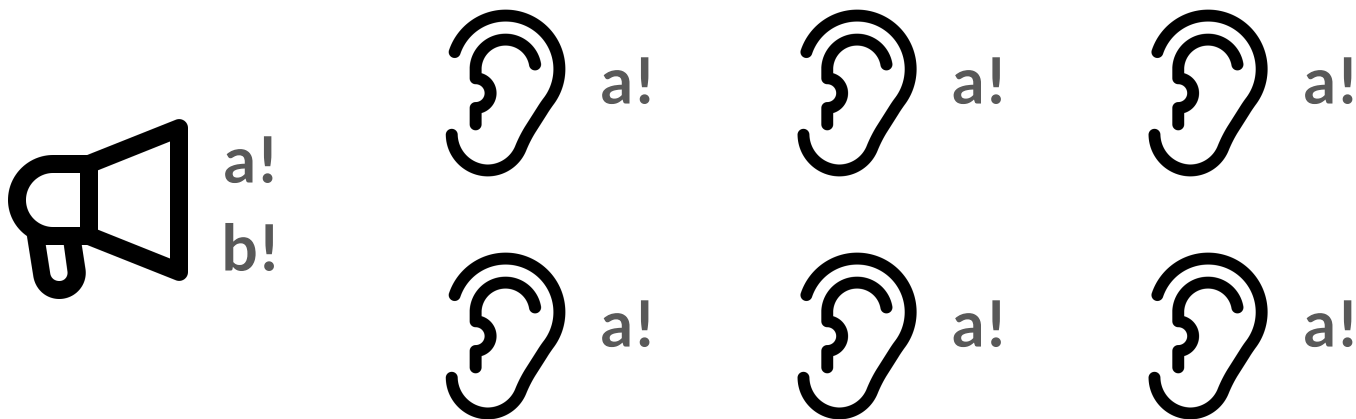
Either every correct process delivers a message, or no correct process delivers a message.

Validity + Consistency + Totality



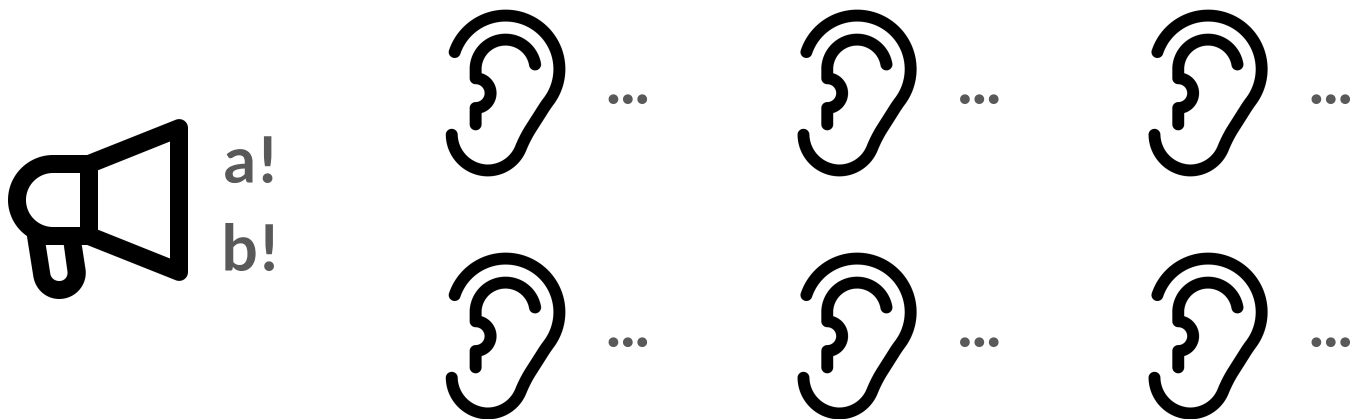
If the sender is correct, every correct process delivers its message. Either every correct process delivers the same message, or no correct process delivers any message.

Validity + Consistency + Totality



If the sender is correct, every correct process delivers its message. Either every correct process delivers the same message, or no correct process delivers any message.

Validity + Consistency + Totality



If the sender is correct, every correct process delivers its message. Either every correct process delivers the same message, or no correct process delivers any message.



Properties

Deterministic case

Validity + Consistency + Totality

~~Termination~~

Properties

Deterministic case





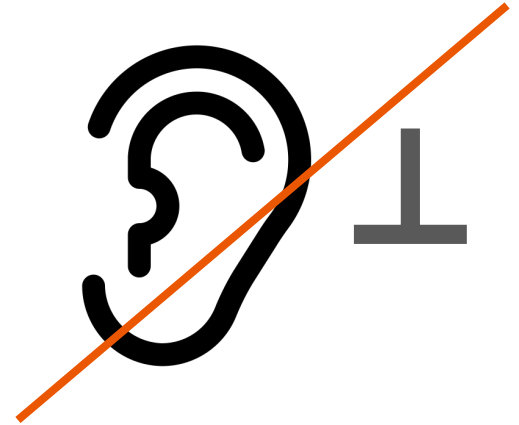
Properties

Deterministic case



Properties

Deterministic case





Properties

Deterministic case

Validity + **Consistency** + **Totality**

Properties

Probabilistic case

Validity + Consistency + Totality

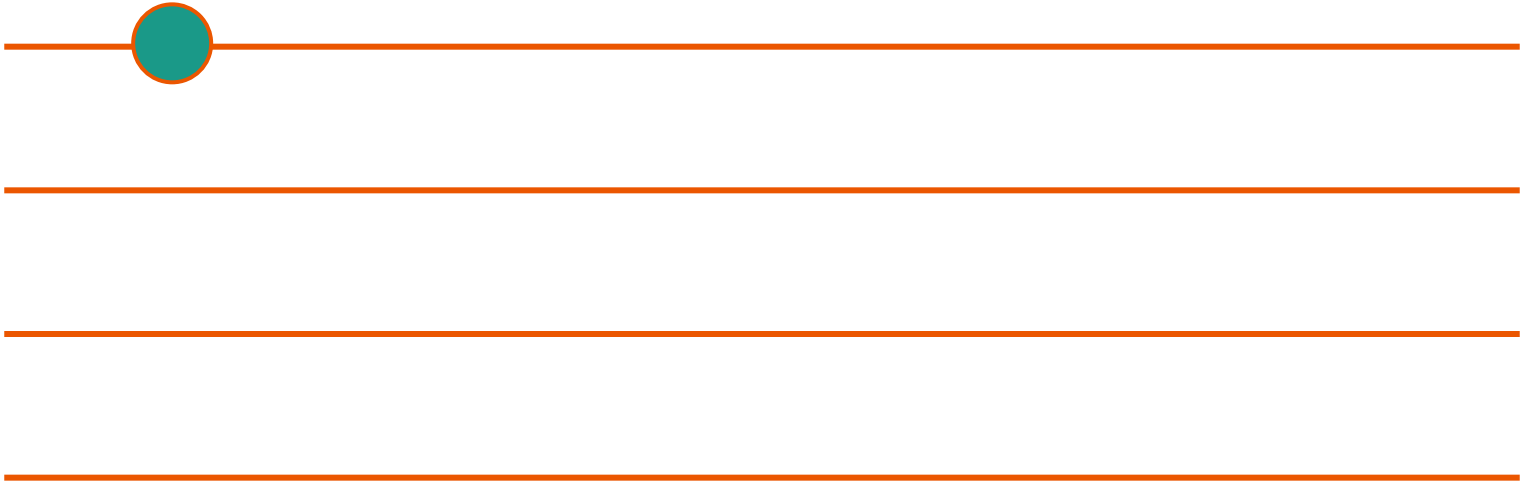


with probability
 $\geq (1 - \epsilon)$

Quorums vs. Samples

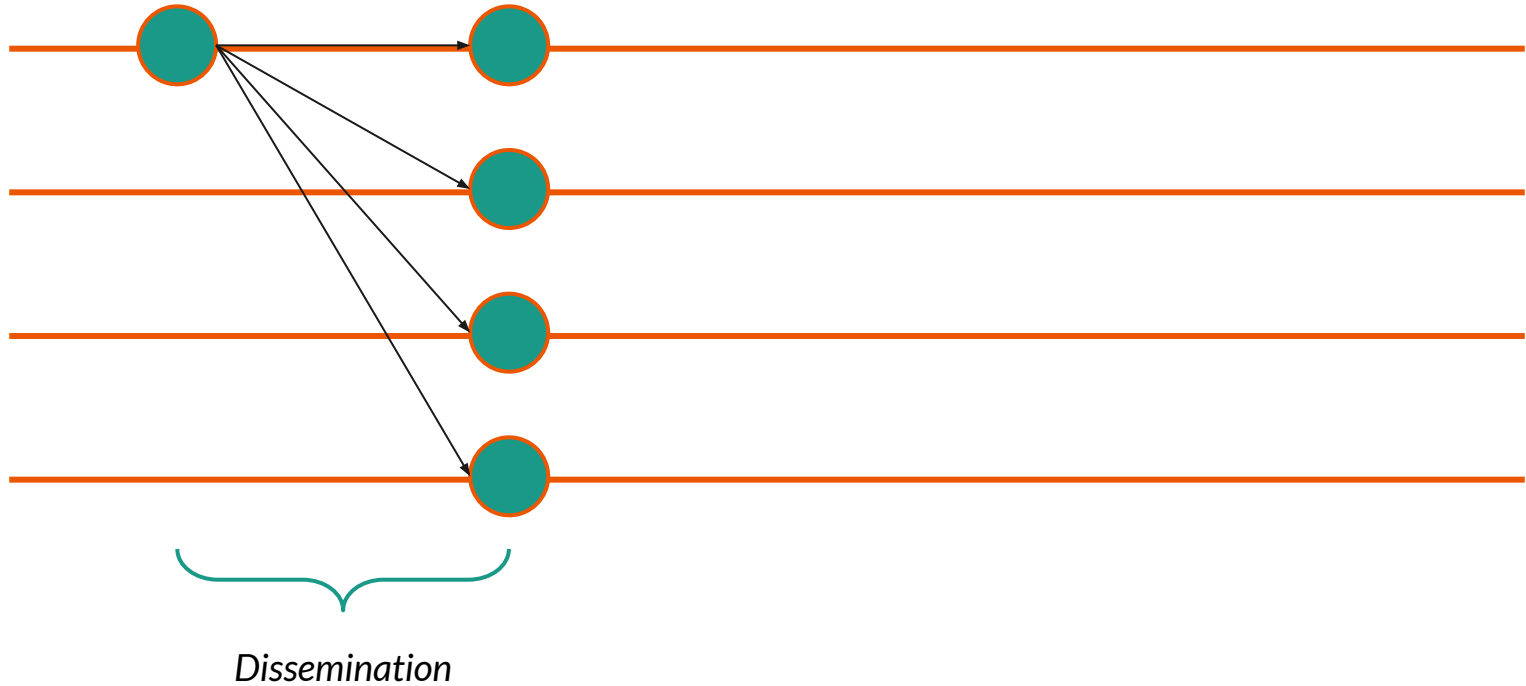


Double-Echo Broadcast

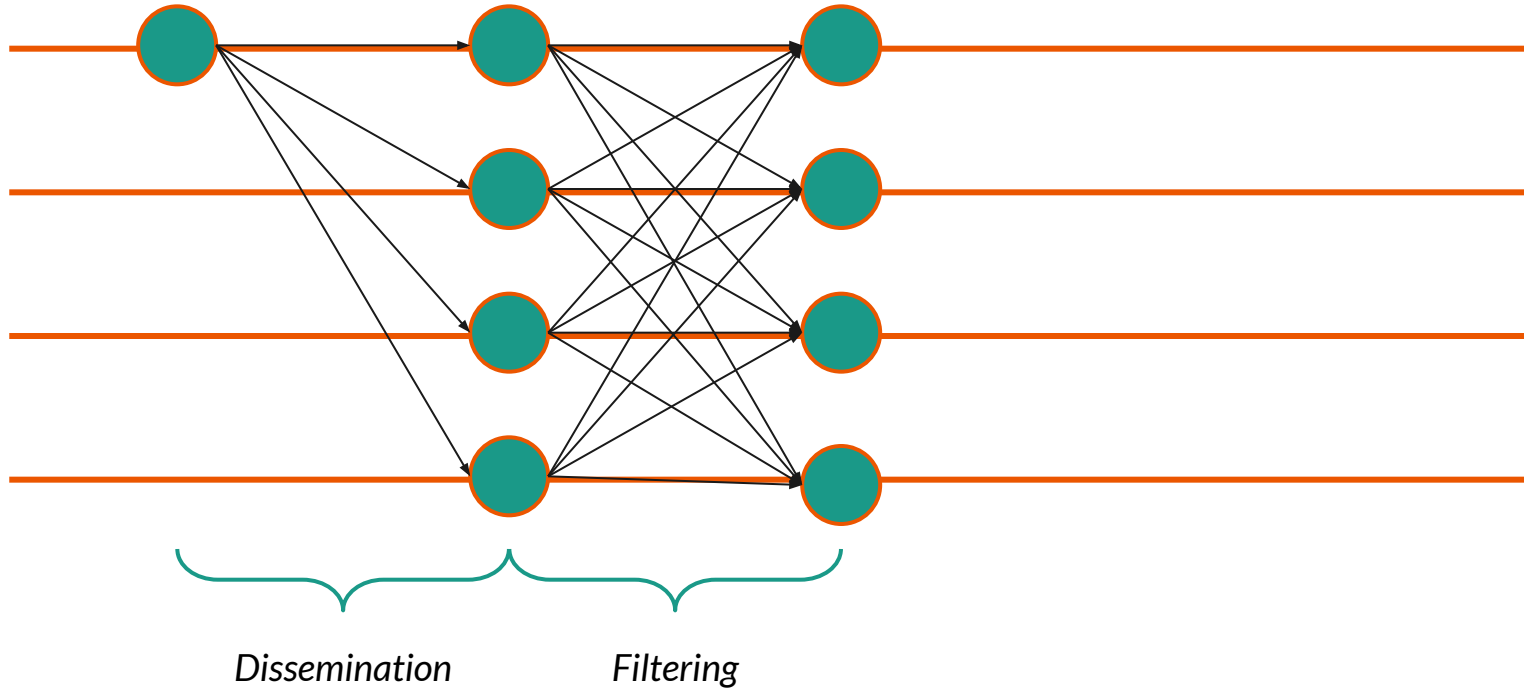


[★] Bracha, Gabriel "Asynchronous Byzantine Agreement Protocols"
Information and Computation, 1987.

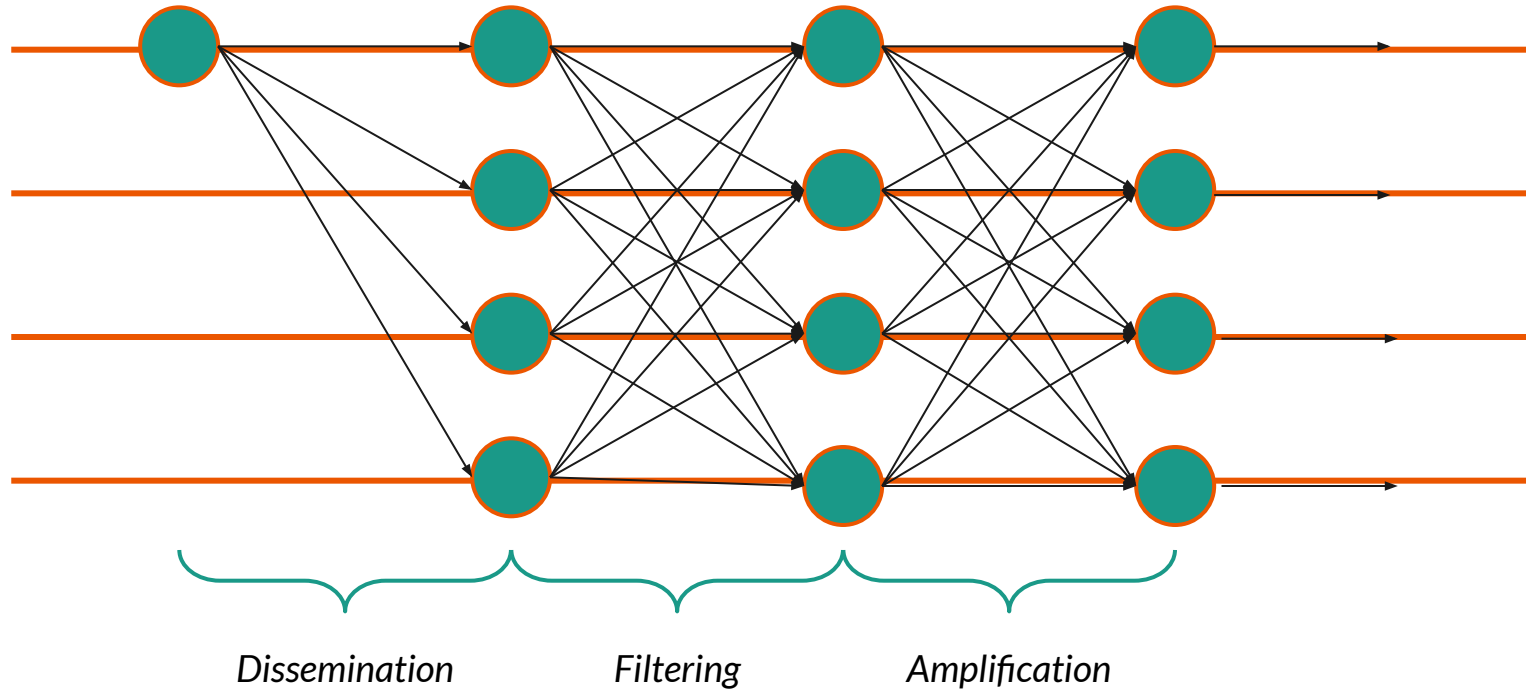
Double-Echo Broadcast



Double-Echo Broadcast



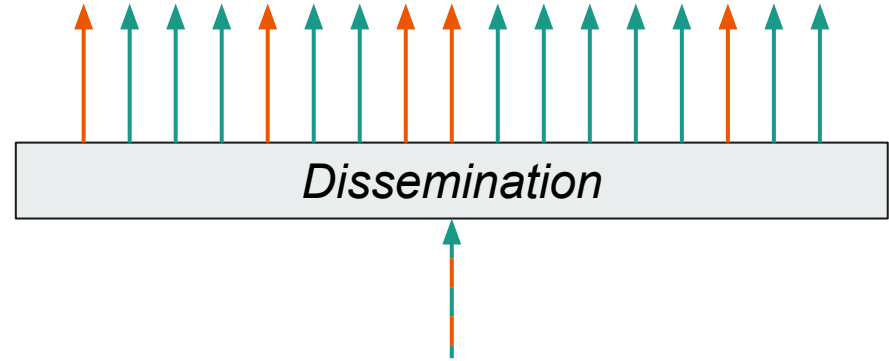
Double-Echo Broadcast





Dissemination

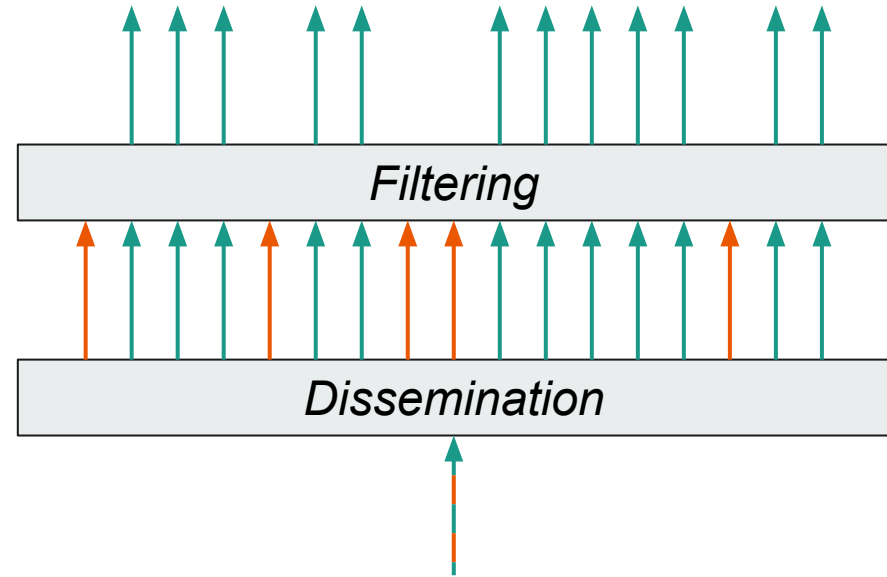
Validity + Totality





Filtering

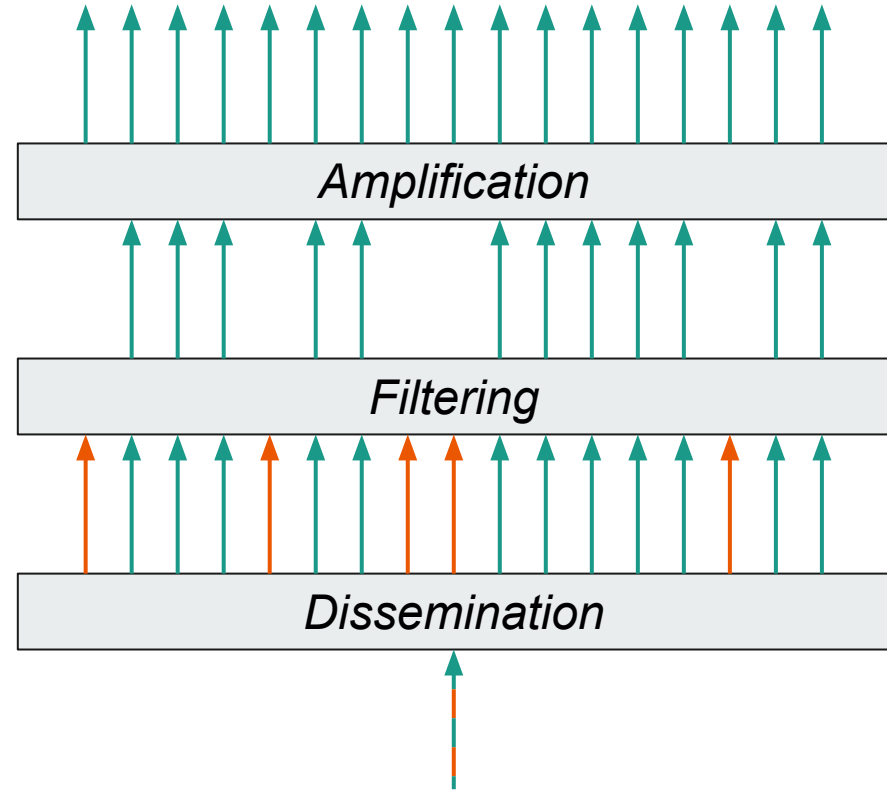
Validity + Consistency





Amplification

Validity + Consistency + Totality



*What fraction q of the
processes satisfy
a property P ?*

Quorum Voting





Quorum Voting

$$q = 17 / 24$$

Quorum Voting



*Gathering a quorum
costs $O(N)$ messages*

Sample Voting

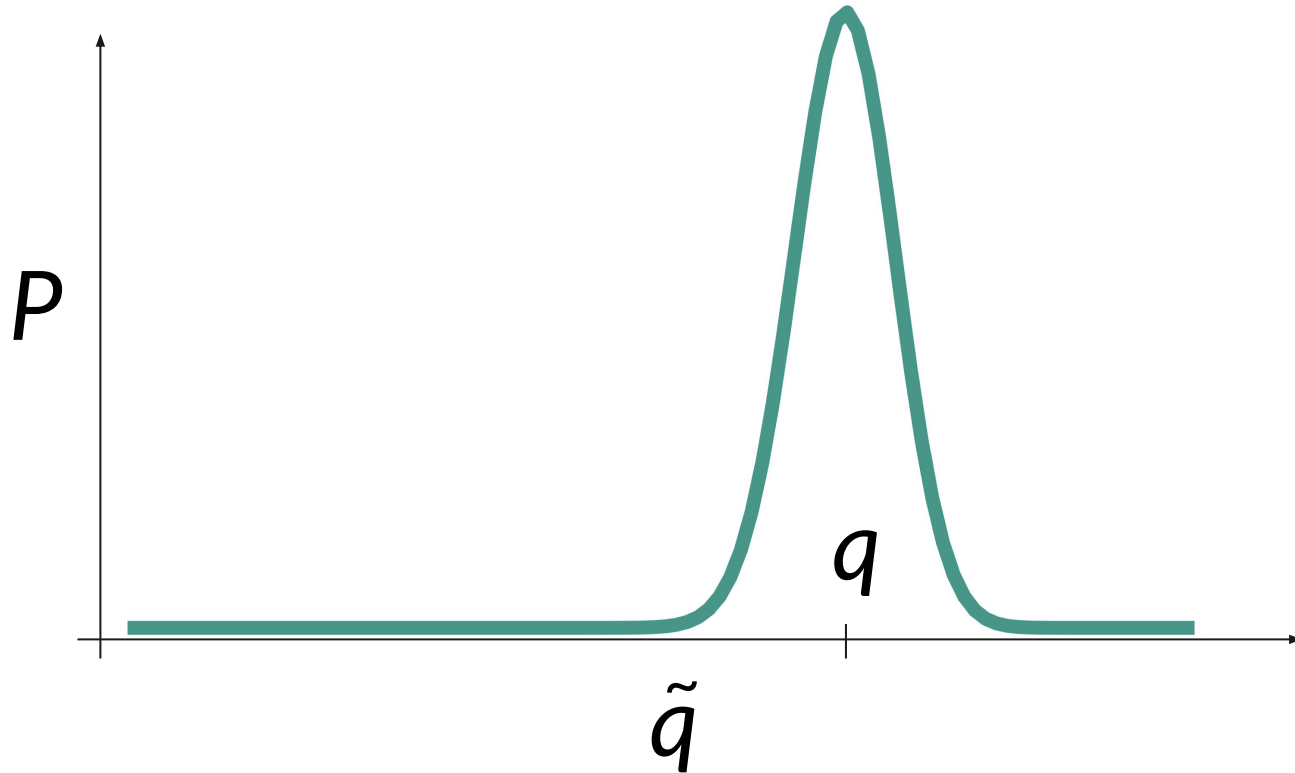


Sample Voting

$$\tilde{q} = 6 / 8$$

$$\approx 17 / 24 = q$$

Outcome Distribution



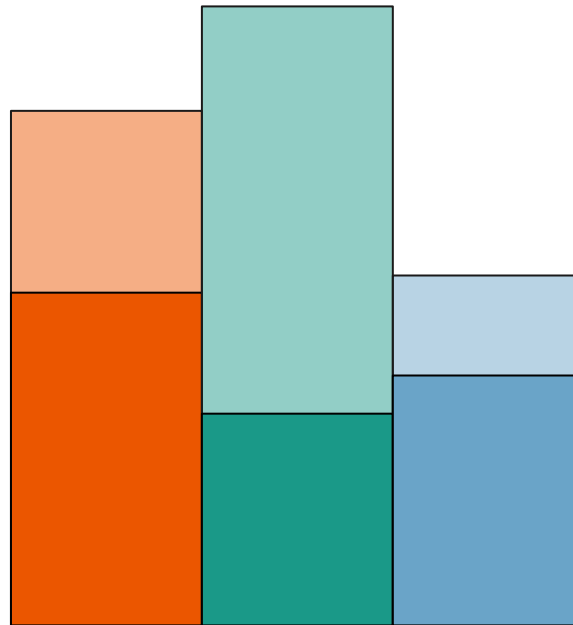


Chernoff Bound

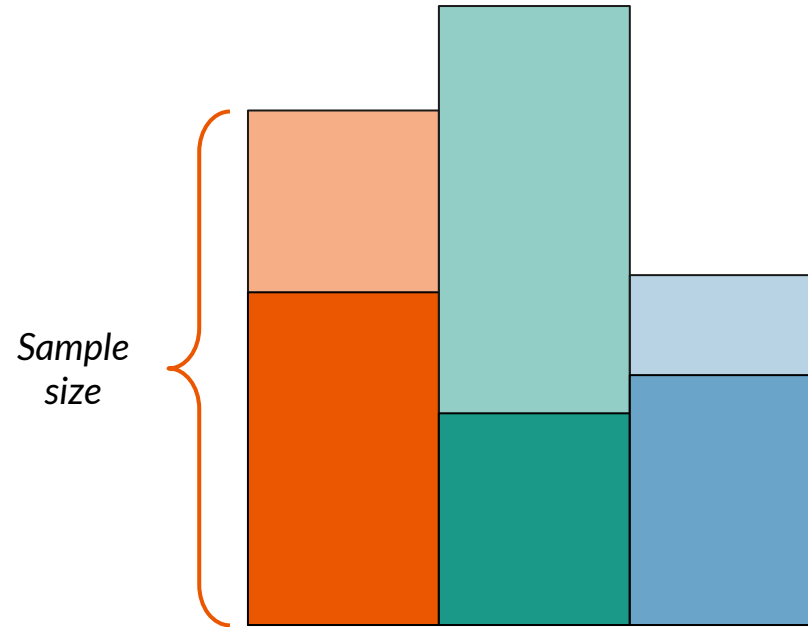
$$P[|\tilde{q} - q| > d] = h(d)^S$$

*The challenge is to include
Byzantine behavior
into the equation.*

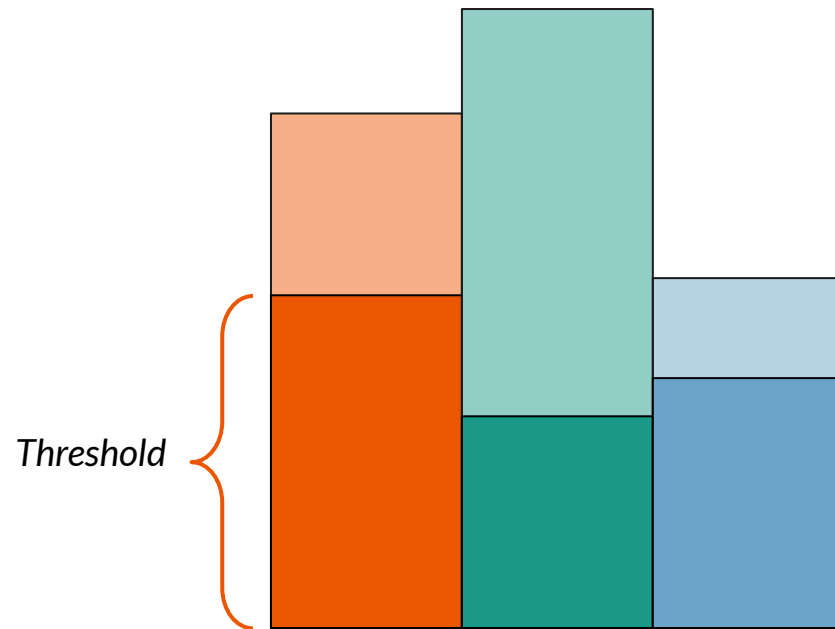
Security analysis



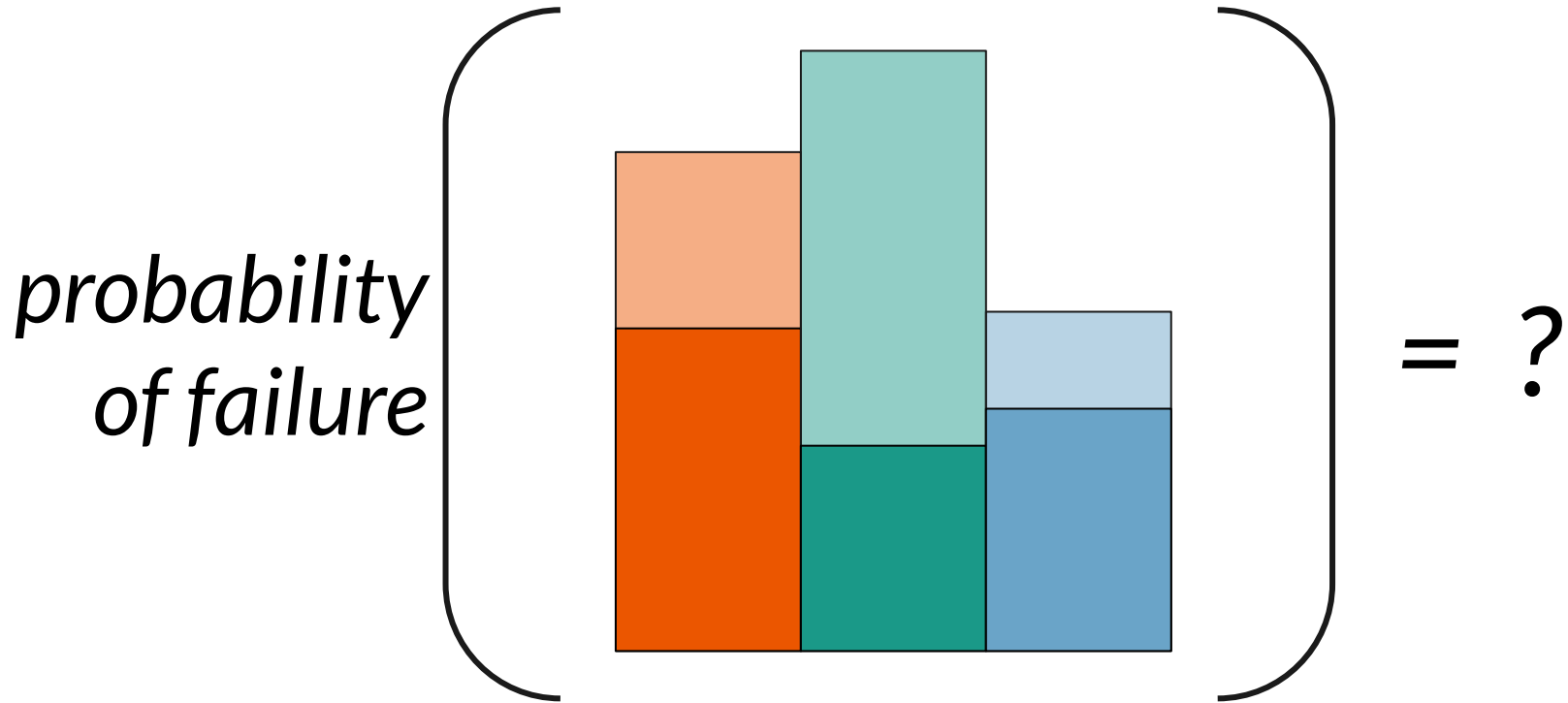
Security analysis



Security analysis

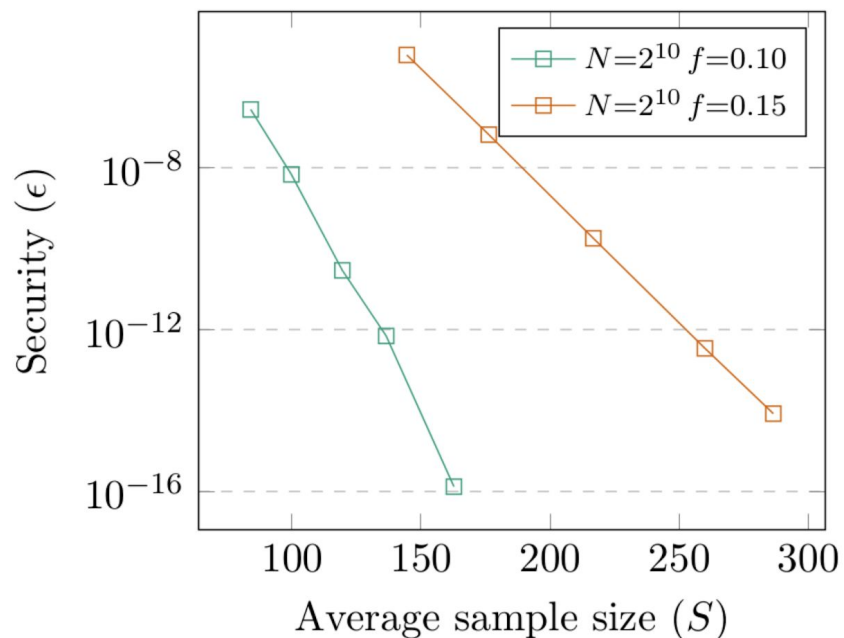


Security analysis

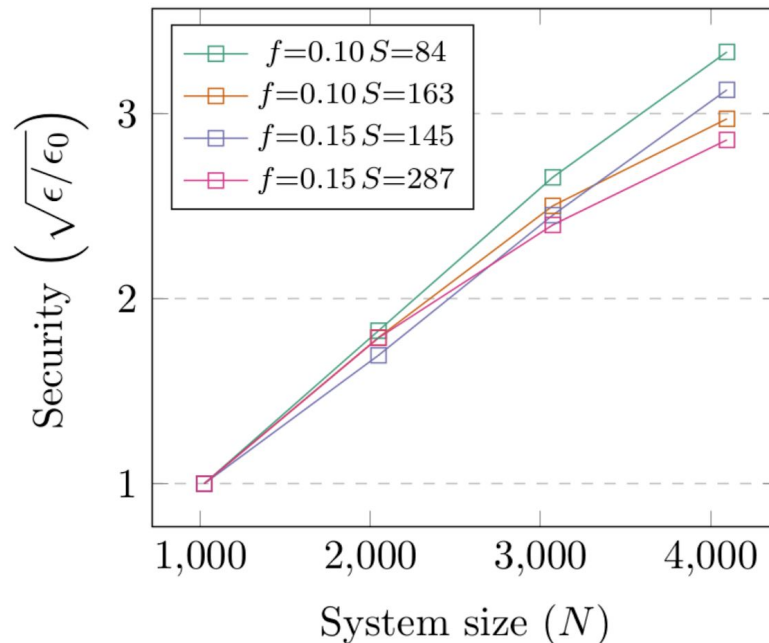


Security analysis

Security vs. Average sample size



Security vs. System size



*Gathering a sample
costs $O(\log N)$ messages★*

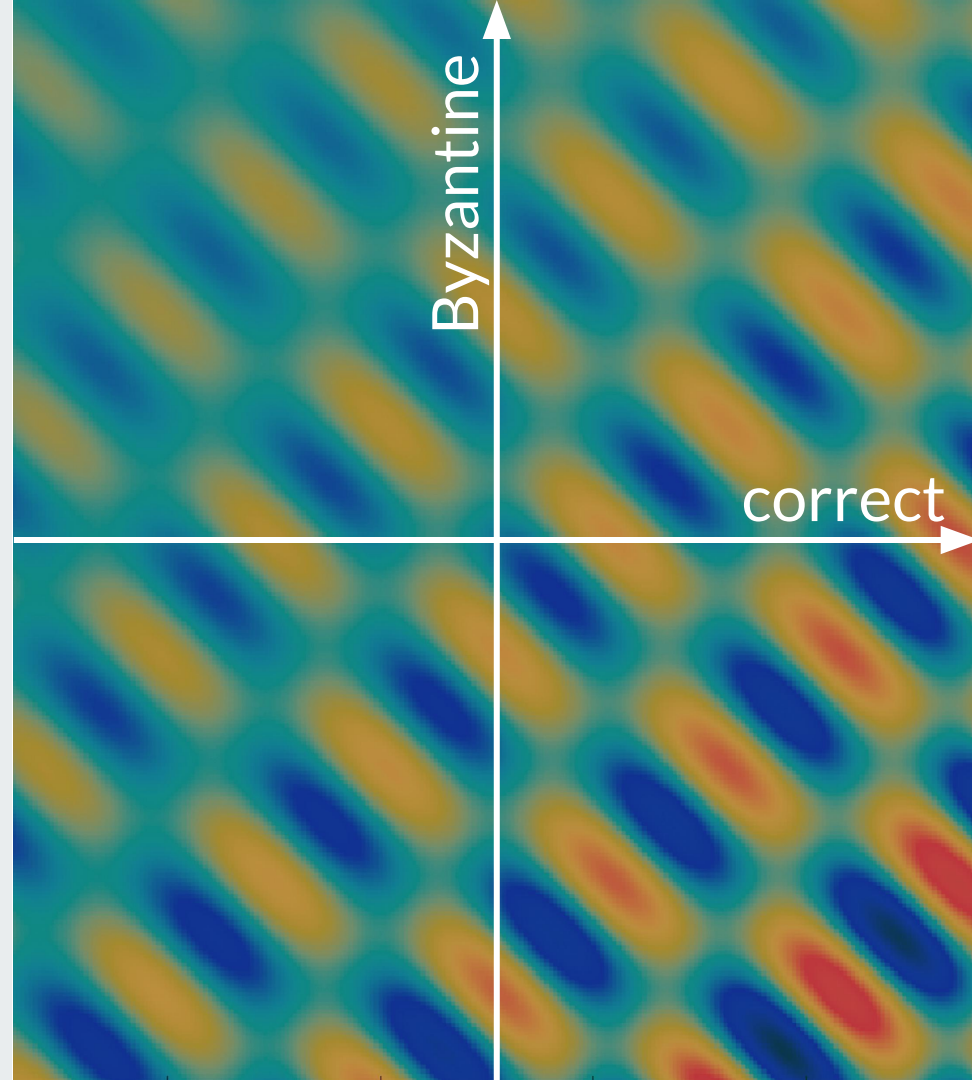
[★] Guerraoui, Rachid, et al. "Scalable Byzantine Reliable Broadcast"
Proceedings of the 33rd International Symposium on Distributed Computing, 2019

Games and Decorators



Distribution of outcomes

How does Byzantine behavior correlate to correct behavior?



***“Arbitrary”
is not the same as
“Random”!***



Solution 1

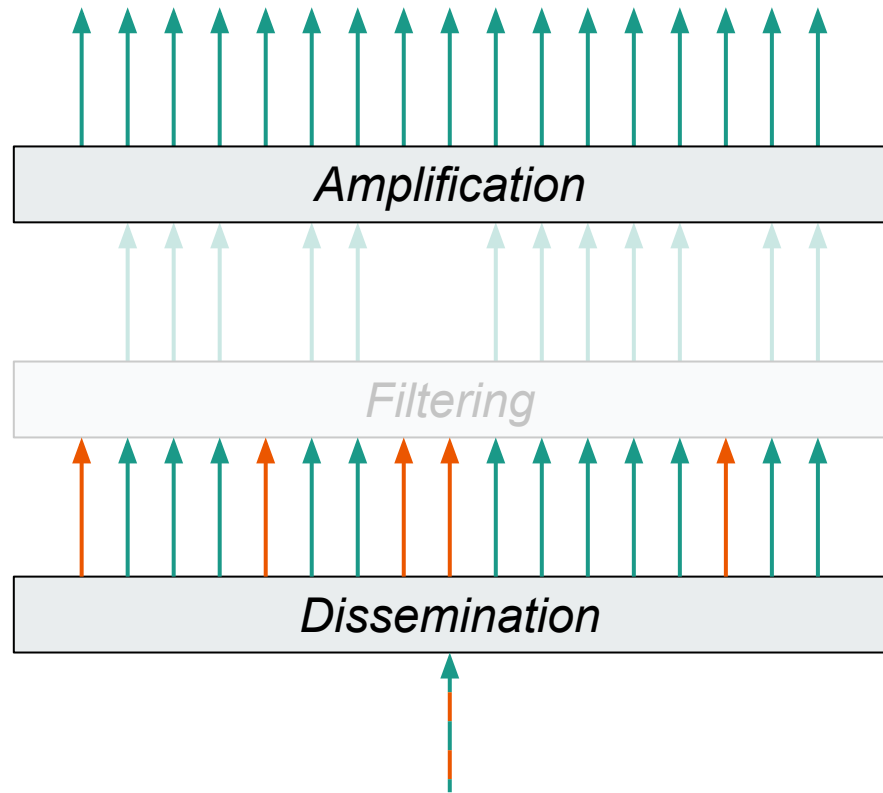
Design an algorithm whose outcome does not depend on adversarial behavior.

Byzantine

correct

Solution 1

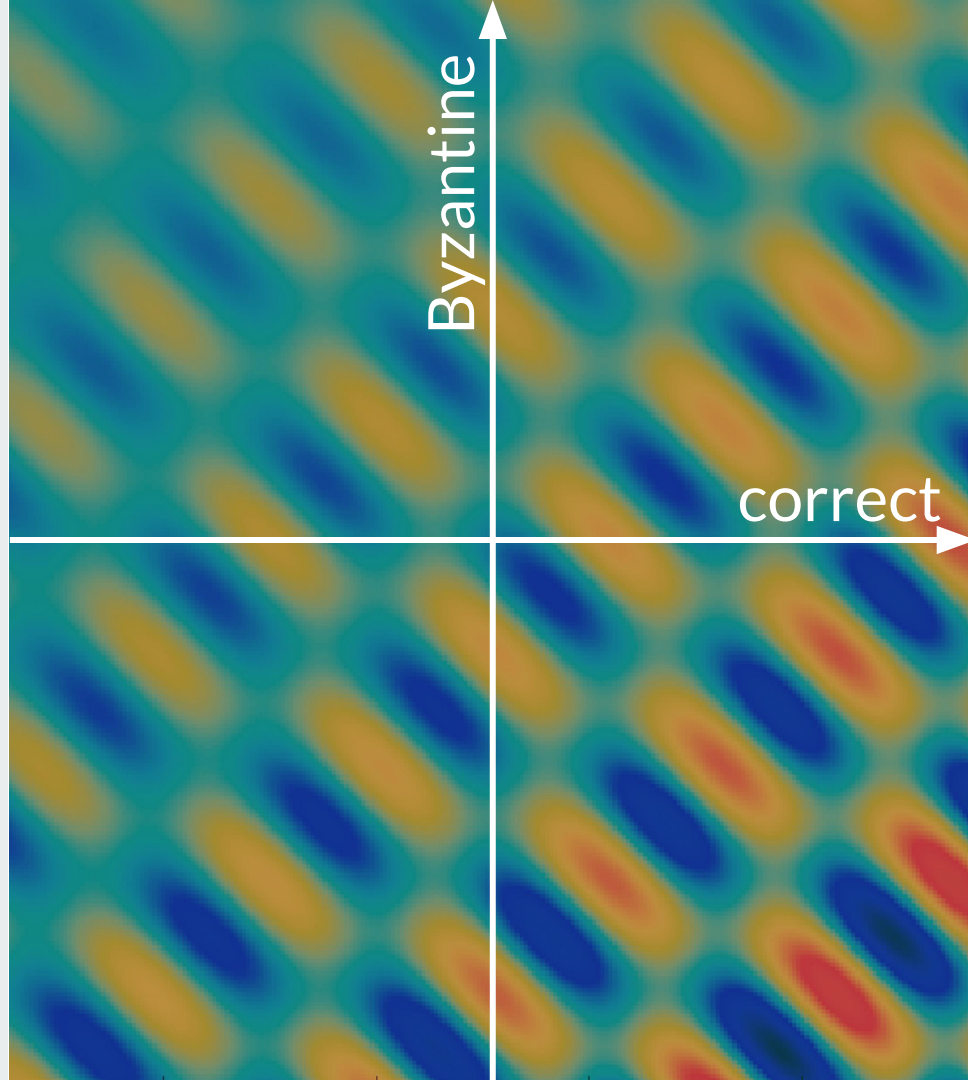
Design an algorithm whose outcome does not depend on adversarial behavior.





Solution 2

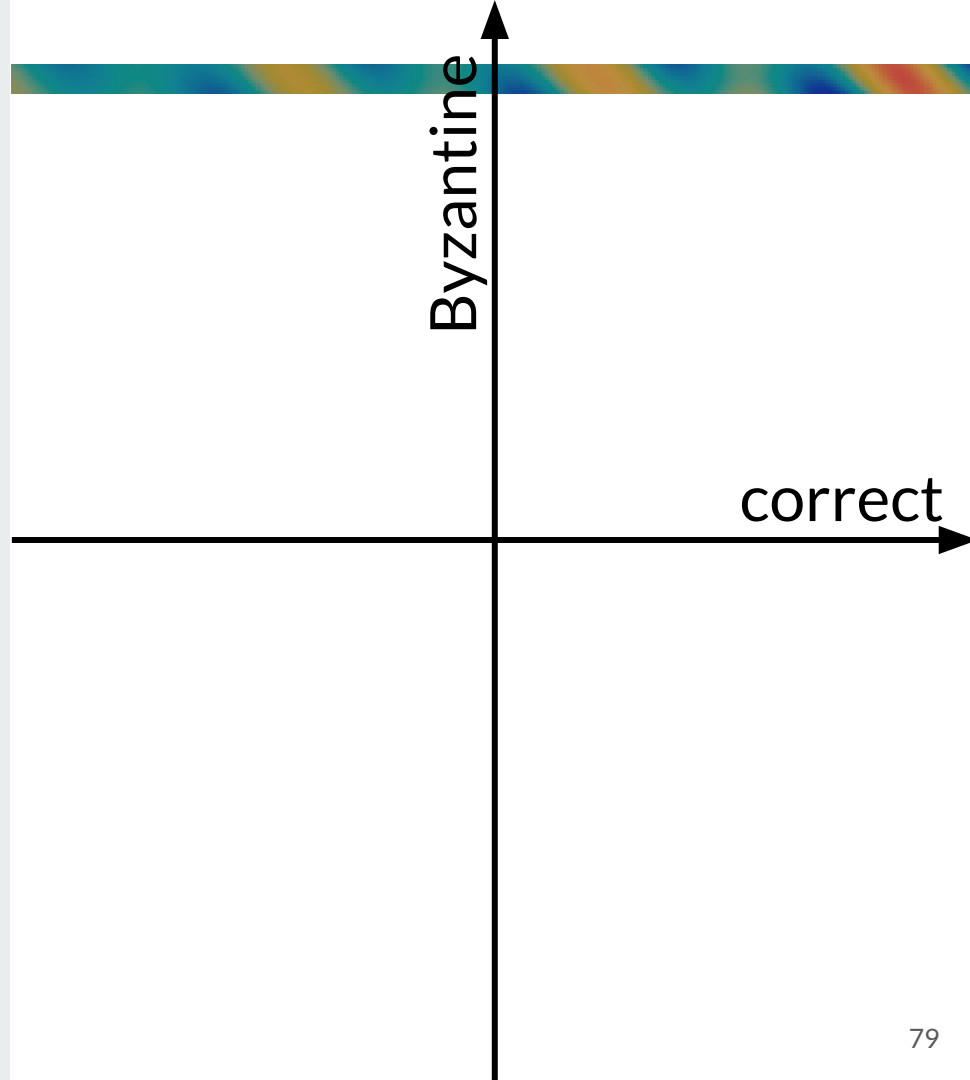
Provably find the *worst*
possible adversary.





Solution 2

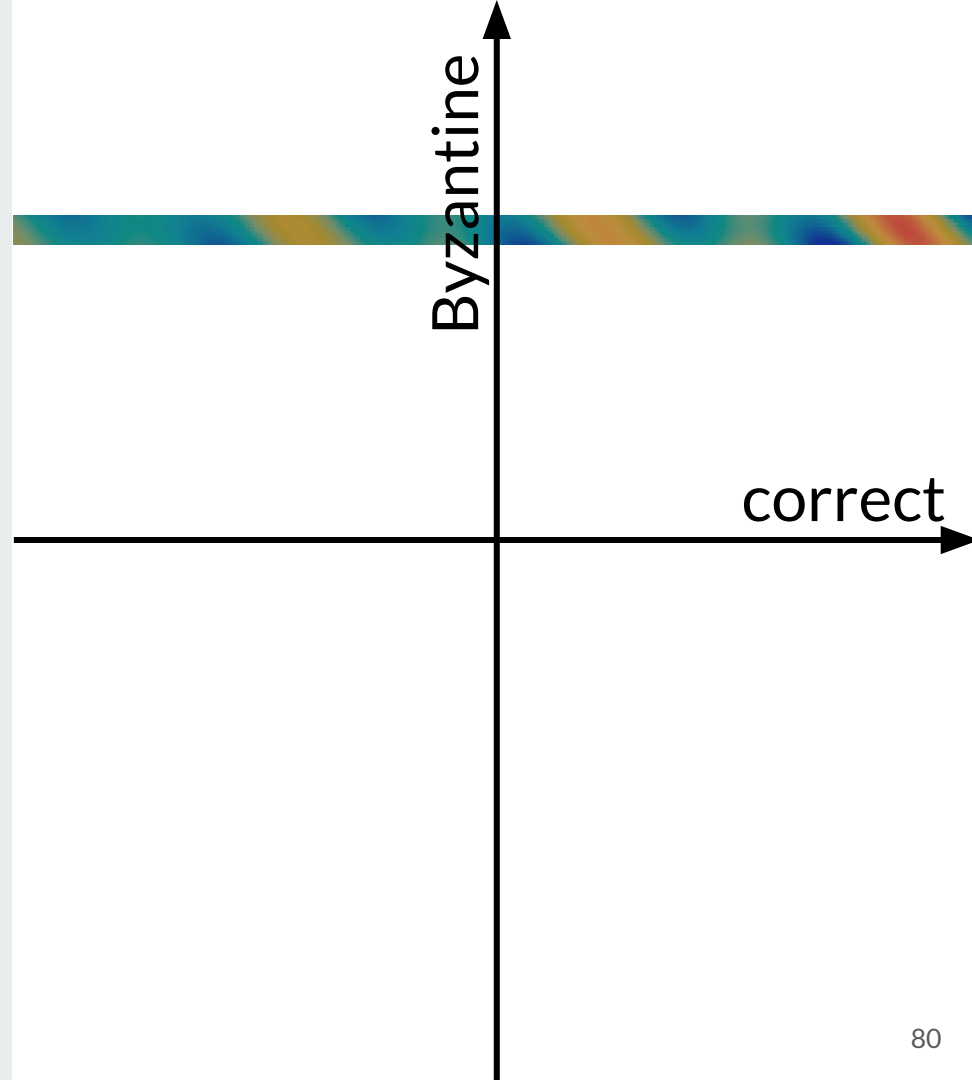
Provably find the *worst*
possible adversary.





Solution 2

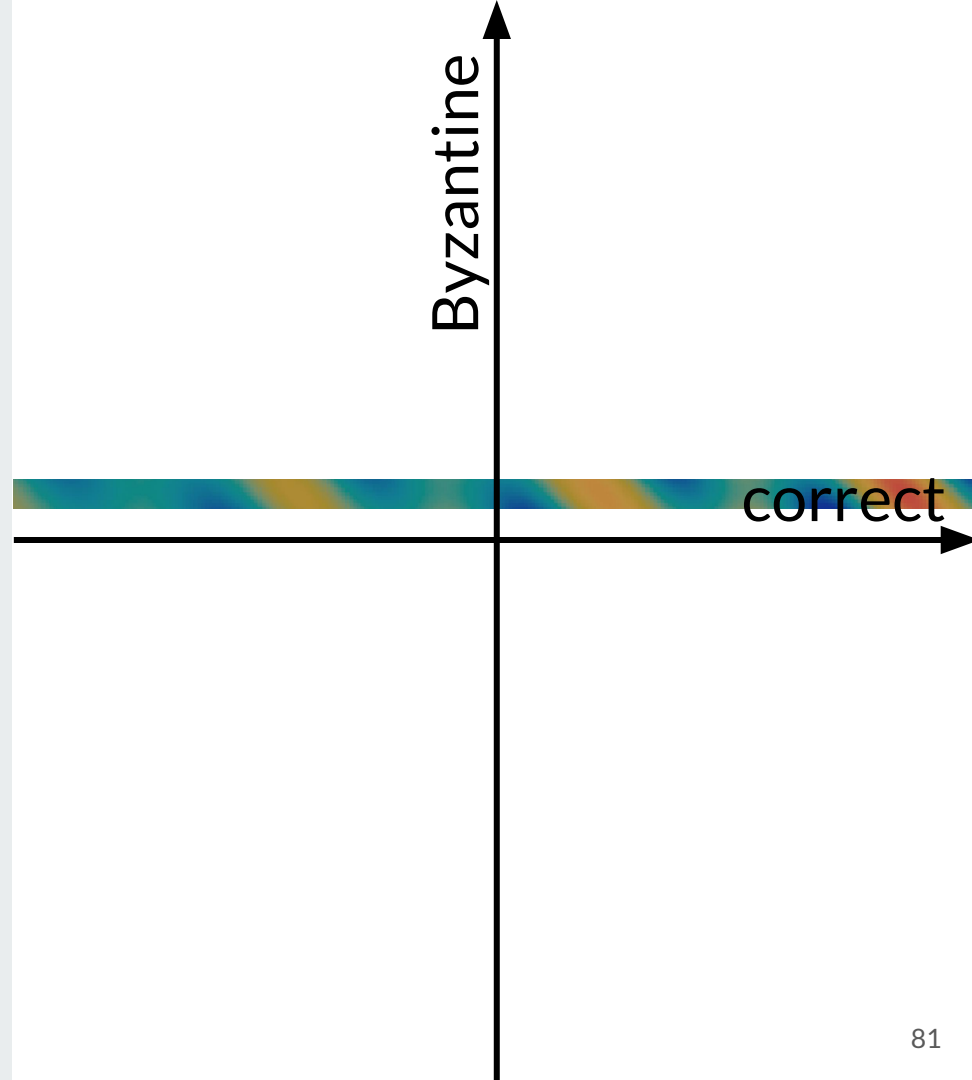
Provably find the *worst*
possible adversary.





Solution 2

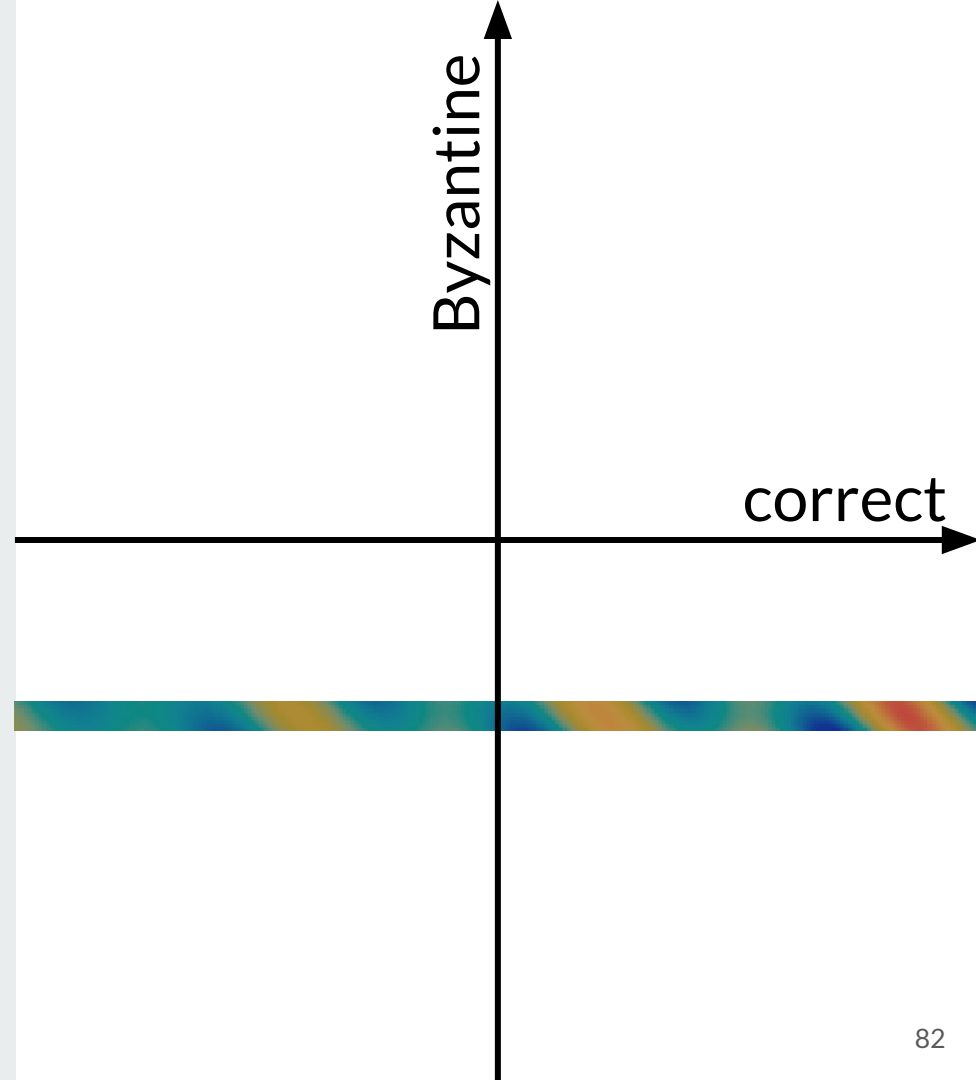
Provably find the *worst*
possible adversary.





Solution 2

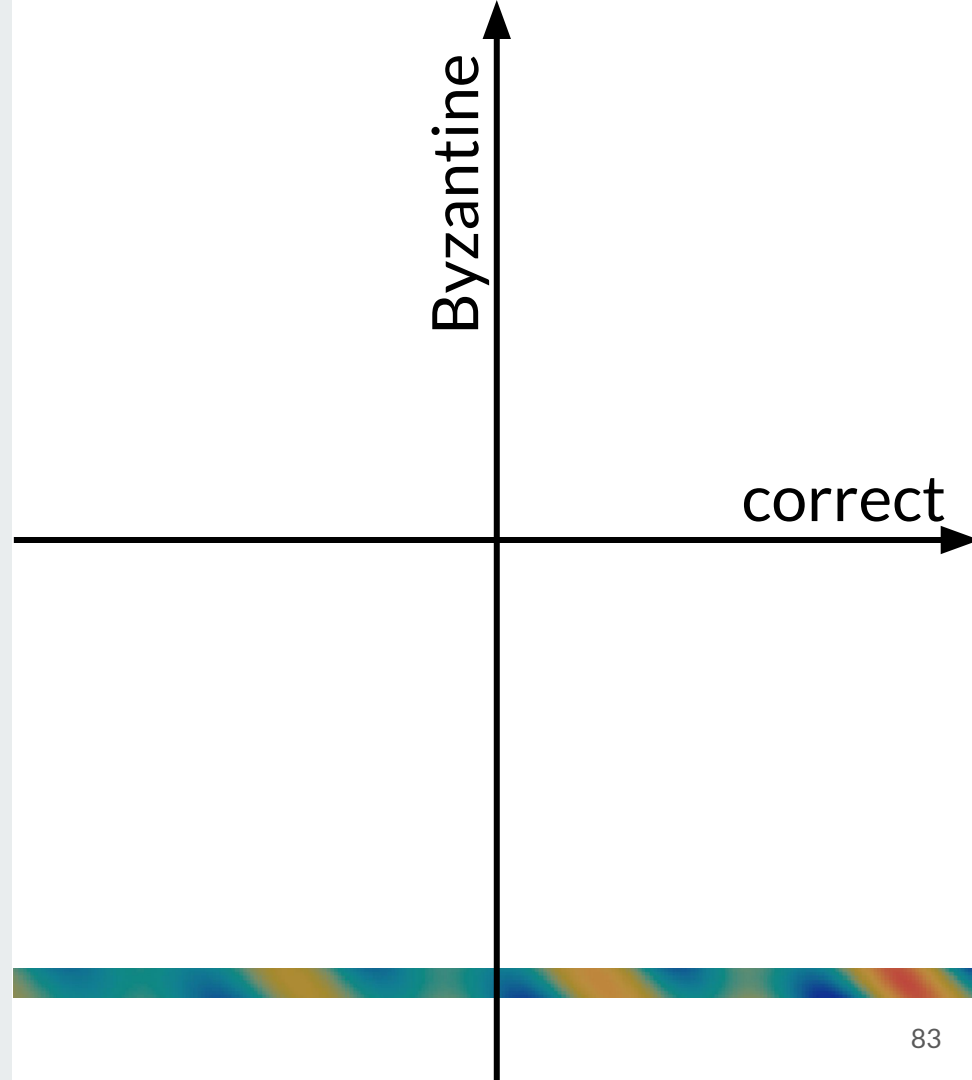
Provably find the *worst*
possible adversary.





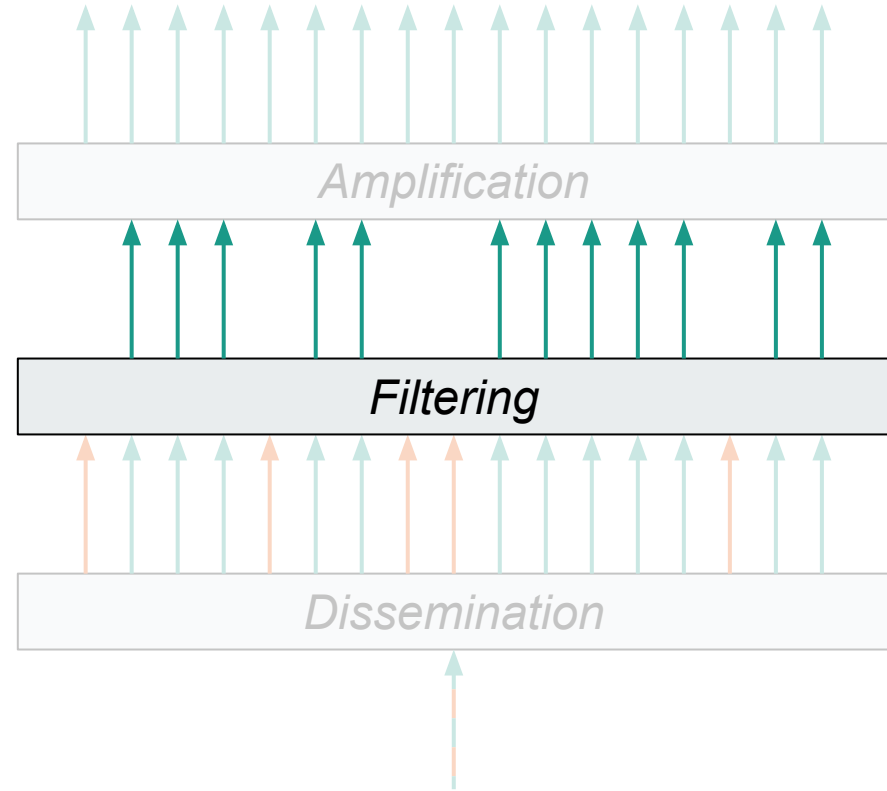
Solution 2

Provably find the *worst*
possible adversary.



Solution 2

Provably find the worst possible adversary.



Set of Adversaries

$\{ \textit{adversaries} \} \ni$

1	aaaaaaaaaaaa
2	aaaaaaaaaaaa
3	aaaaaaaaaaaa
4	aaaaaaaaaaaa
5	aaaaaaaaaaaa
6	aaaaaaaaaaaa

Set of Adversaries

$\{ \textit{adversaries} \} \ni$

1	aaaaaaaaaaaa
2	aaaaaaaaaaaa
3	aaaaaaaaaaaa
4	aaaaaaaaaaaa
5	aaaaaaaaaaaa
6	aaaaaaaaaab

Set of Adversaries

$\{ \textit{adversaries} \} \ni$

1	aaaaaaaaaaaa
2	aaaaaaaaaaaa
3	aaaaaaaaaaaa
4	aaaaaaaaaaaa
5	aaaaaaaaaaaa
6	aaaaaaaaaac

Set of Adversaries

$\{ \textit{adversaries} \} \ni$

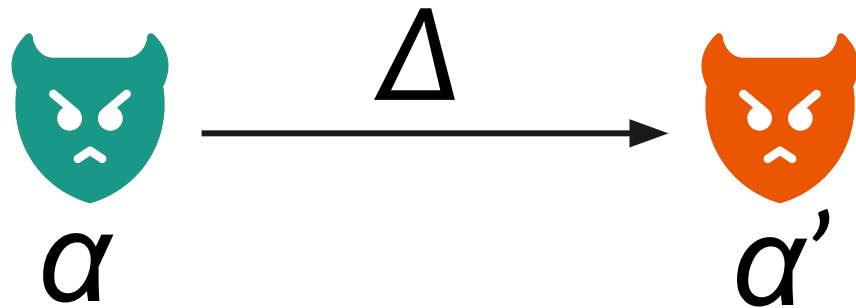
```
1  if sneak {  
2    loop {  
3      noise();  
4    }  
5  }  
6  // D'oh!
```


Set of Adversaries

$\{ \textit{adversaries} \} \ni$

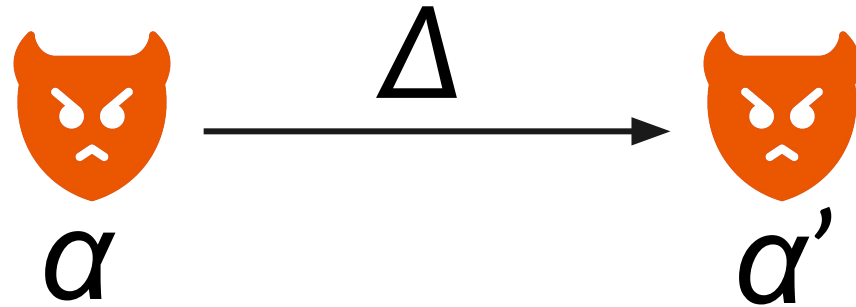
```
1  loop {  
2      adversary.  
3      settings.  
4      evil += 1;  
5  }  
6  // Mhuahua!
```

Decorator



$$P[\alpha' \text{ wins}] \geq P[\alpha \text{ wins}]$$

Decorator

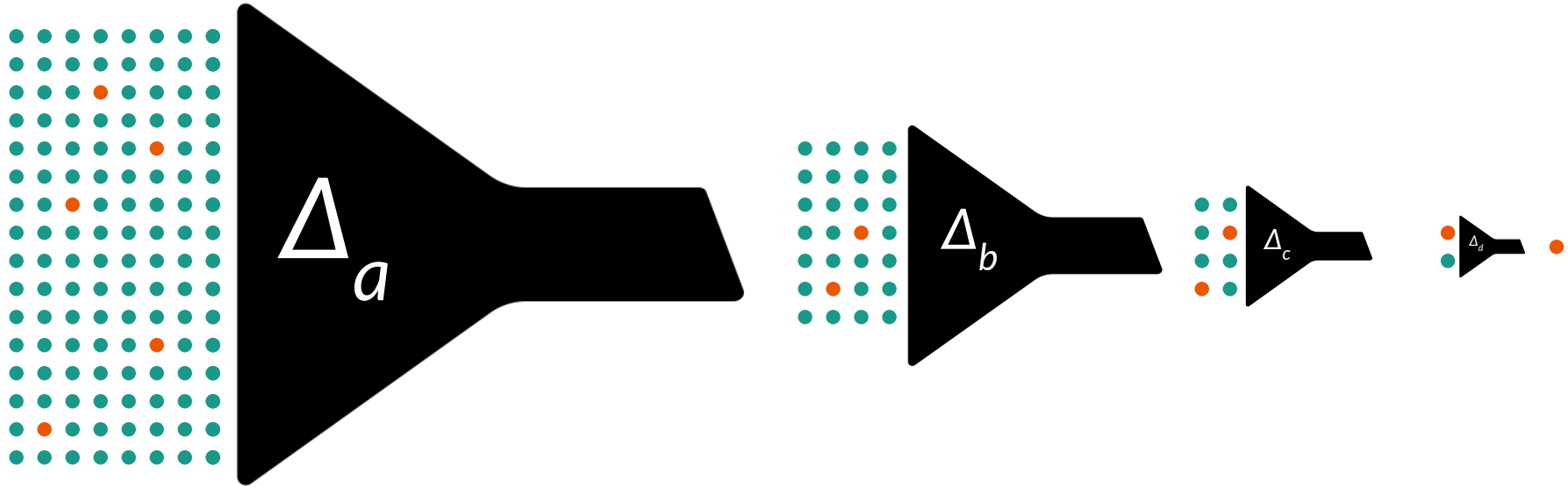


If α is optimal, α' is optimal.

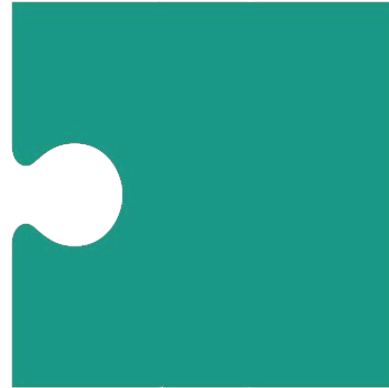
Codomain of a decorator



Composing decorators



Building a decorator

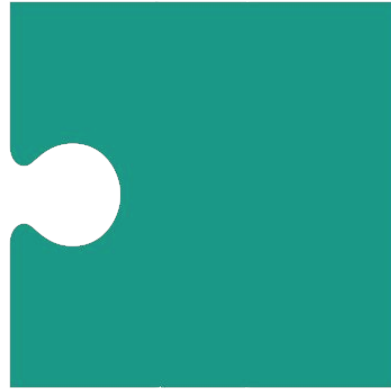


System

Building a decorator

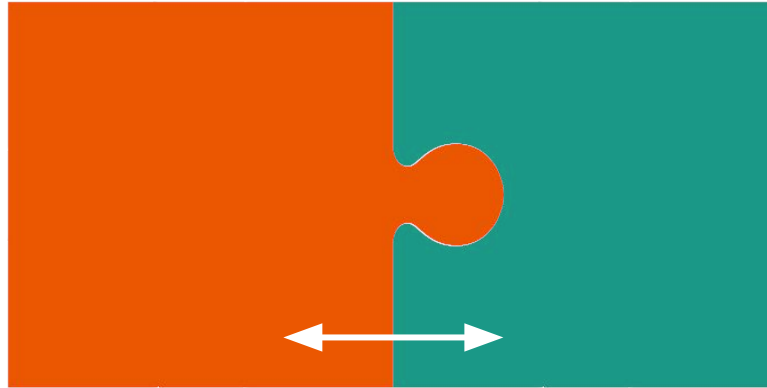


Adversary



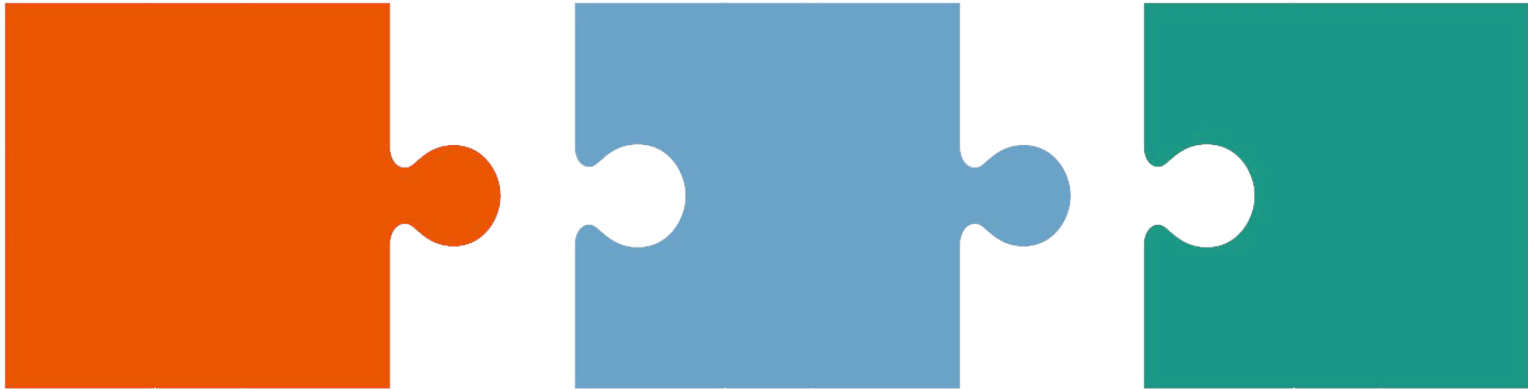
System

Building a decorator



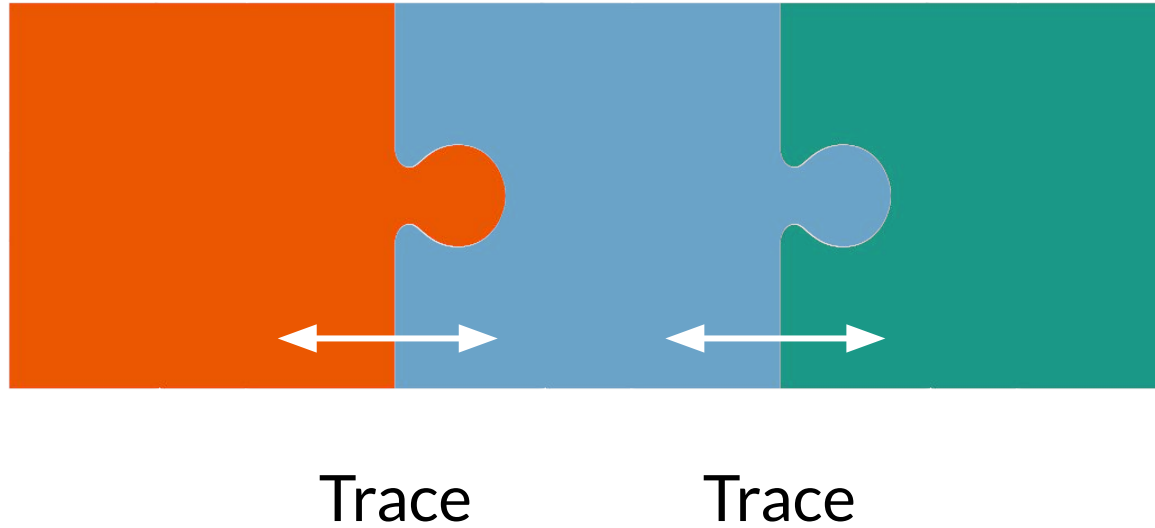
Trace

Building a decorator

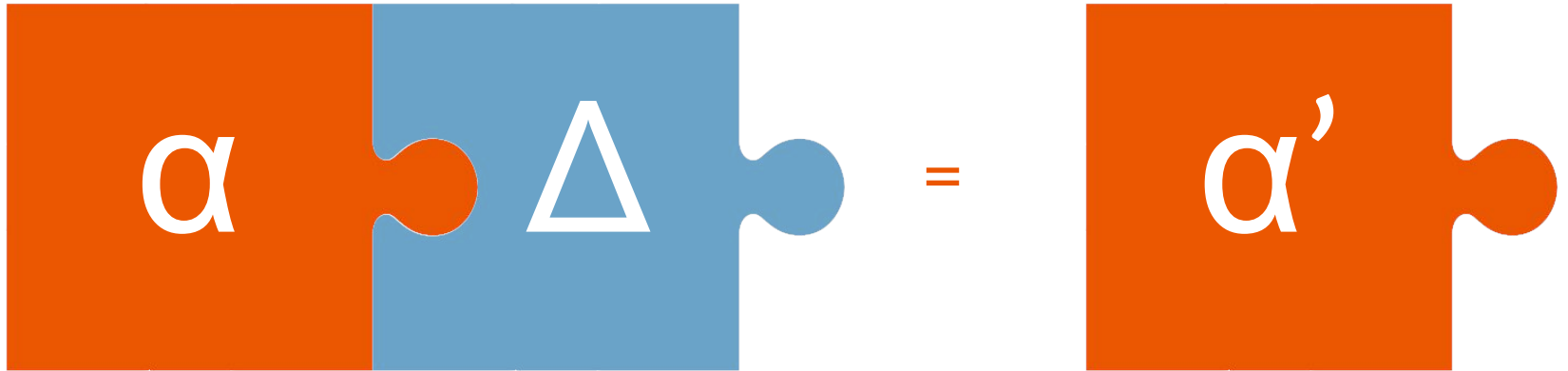


Decorator

Building a decorator



Building a decorator



Building a decorator



We can convince α to be playing against a “luckier” system σ' , while translating its calls to the original system σ !

Towards a Planetary Database

Solving Total Order Broadcast

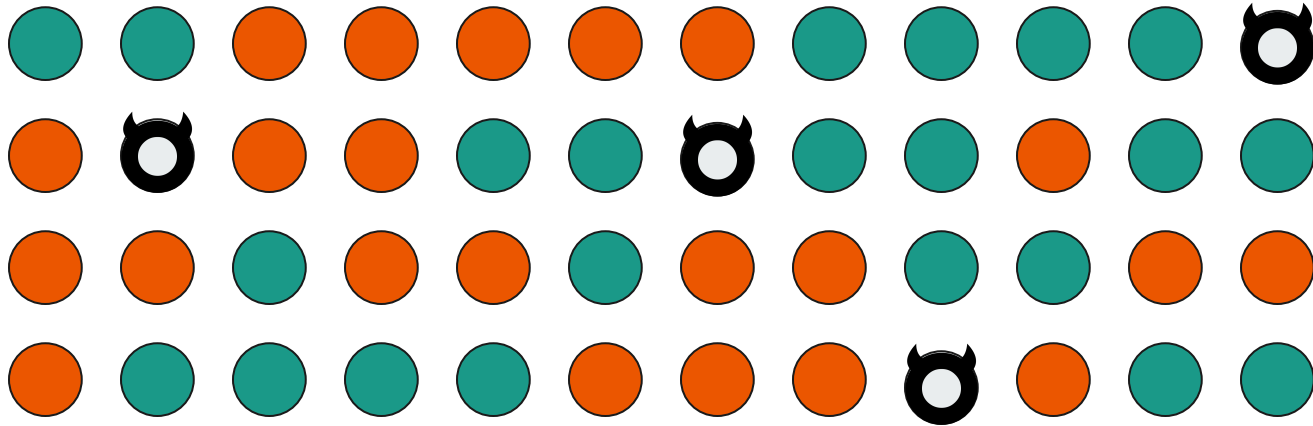
=

Solving Database

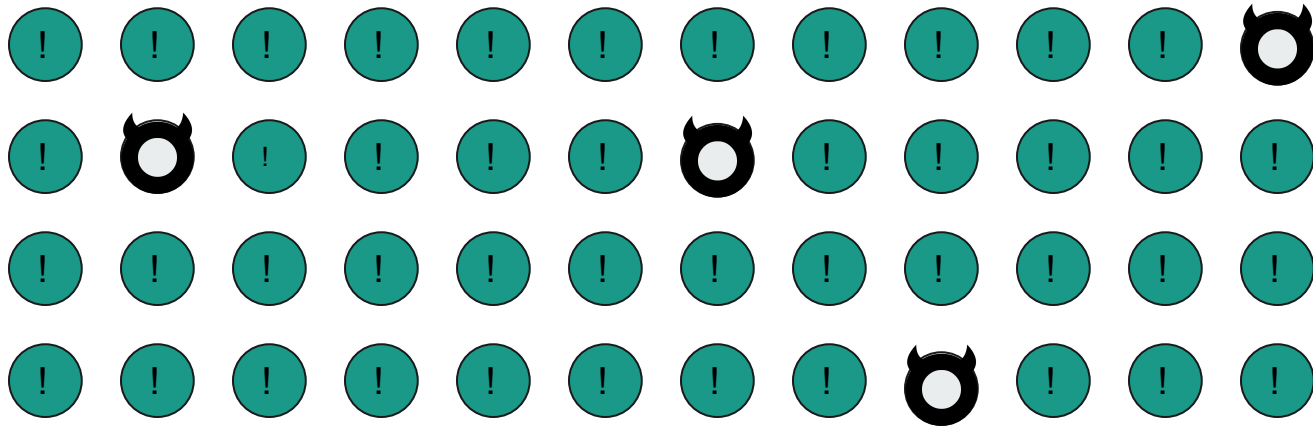
—

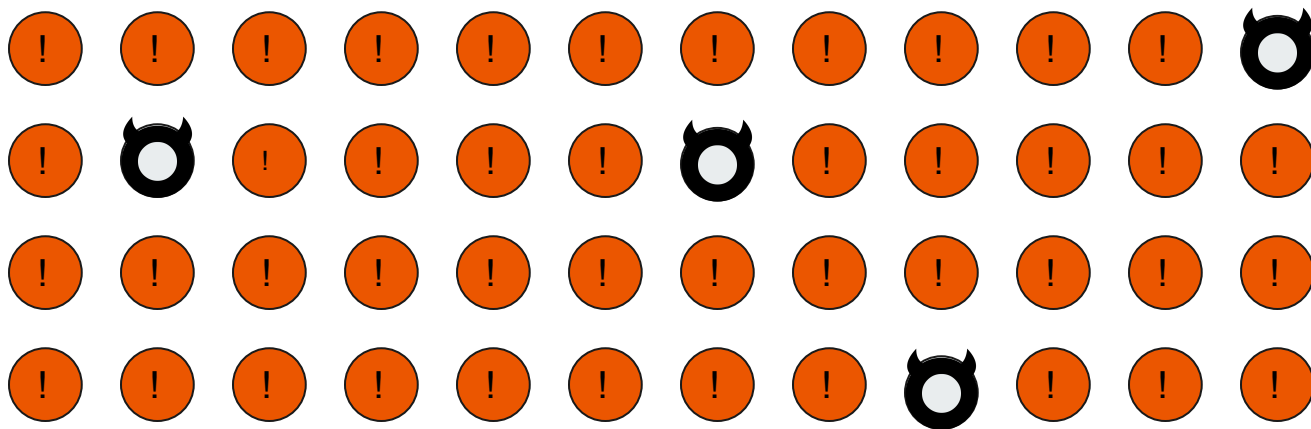
Solving Consensus
=
Solving Total Order Broadcast

Proposals..

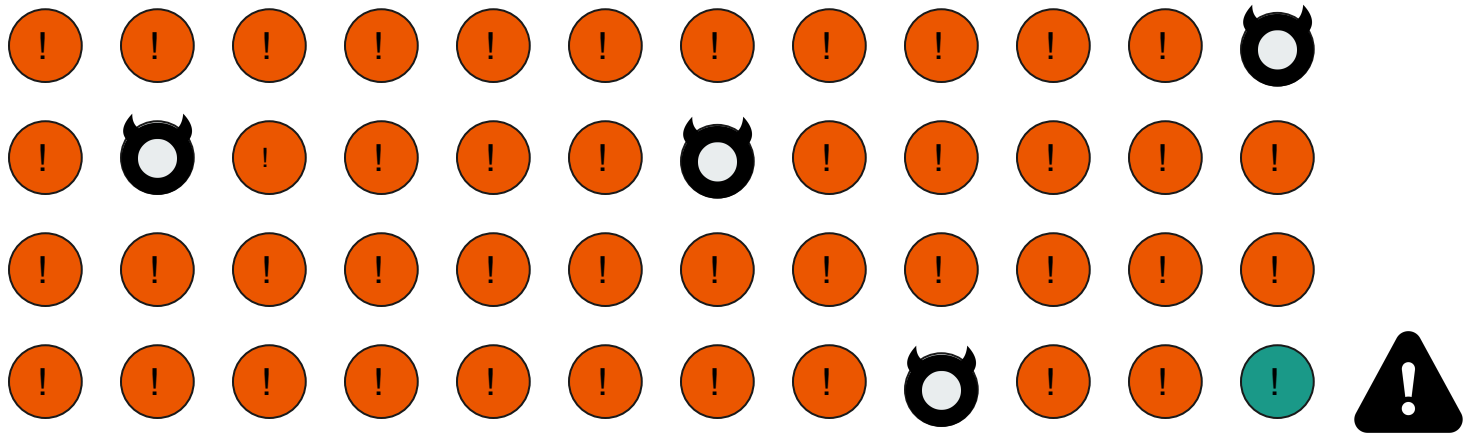


.. and decisions



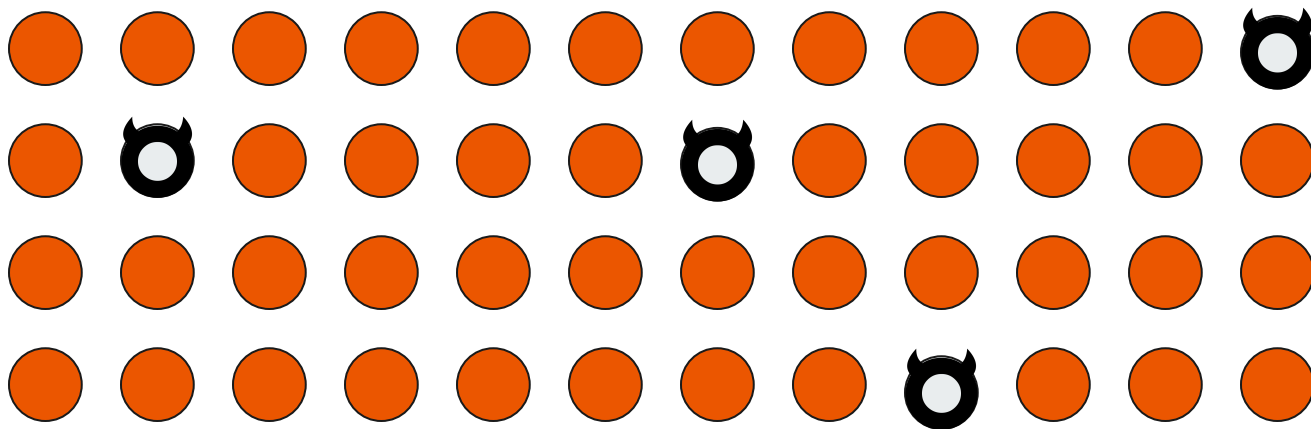


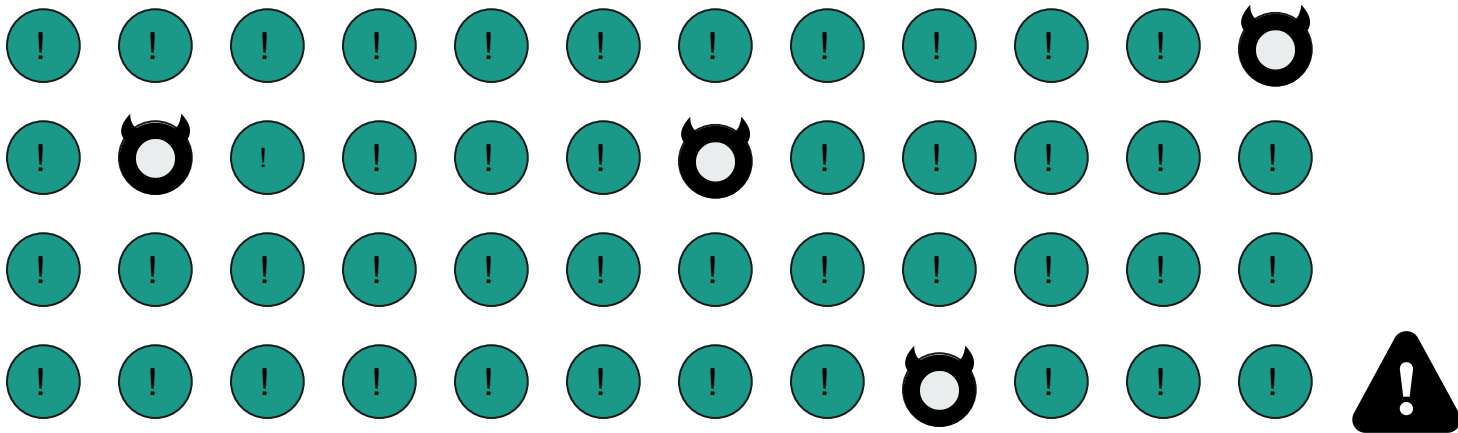
Agreement..

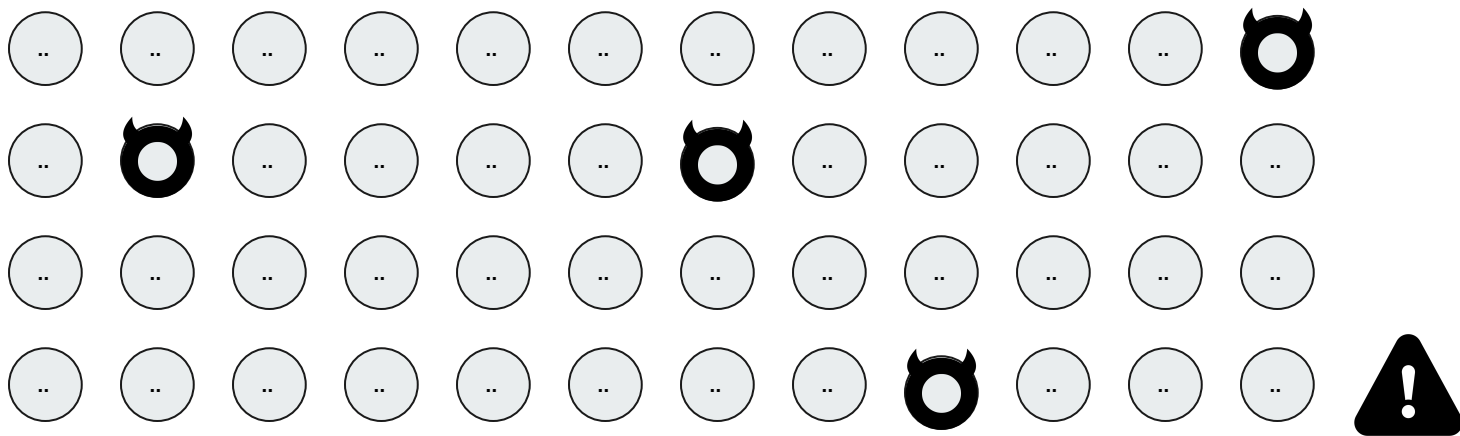




.. validity..





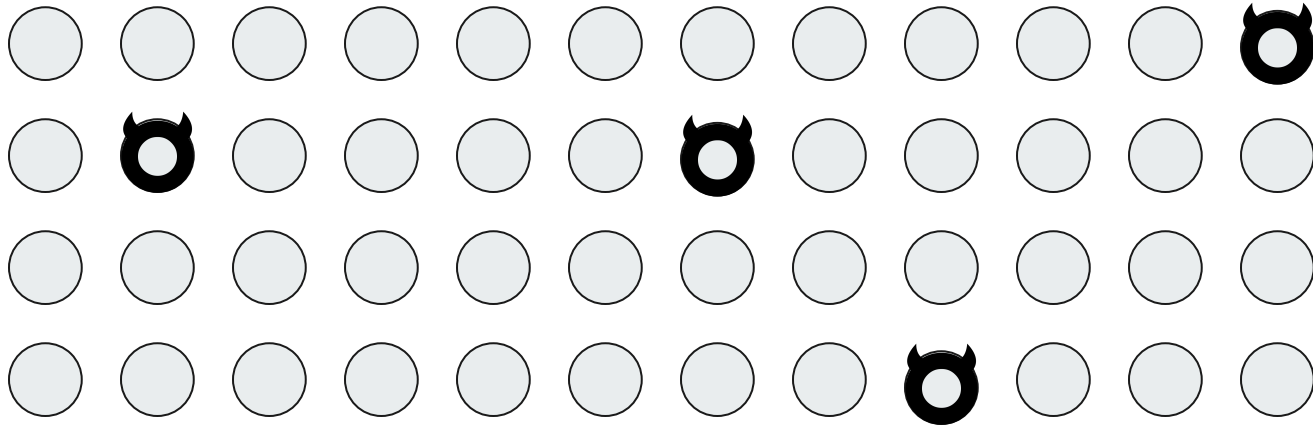


We can verify that we agree.

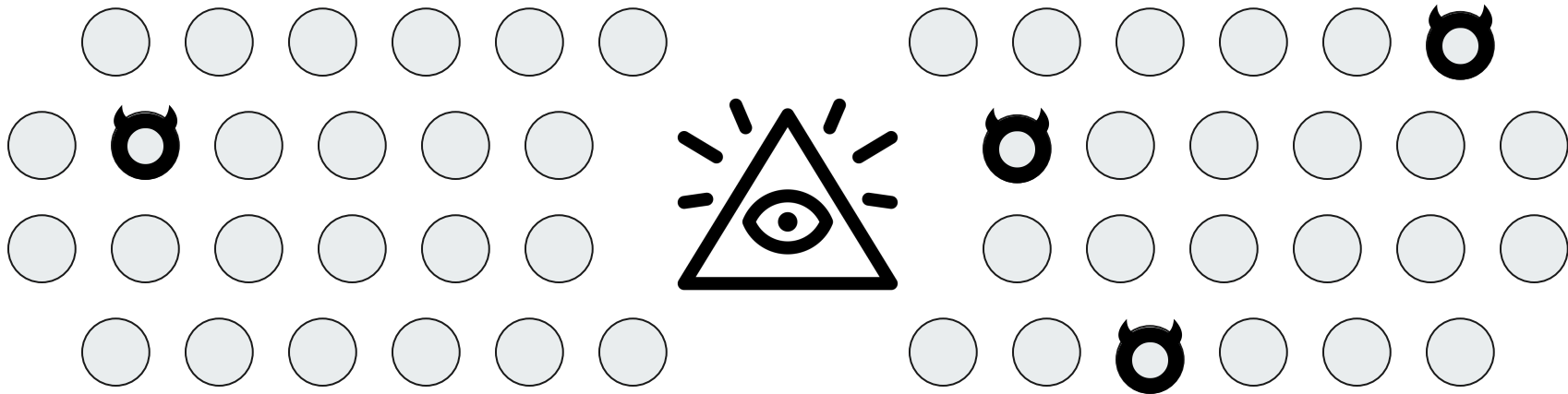
*If we don't agree, might as well
choose at random!*



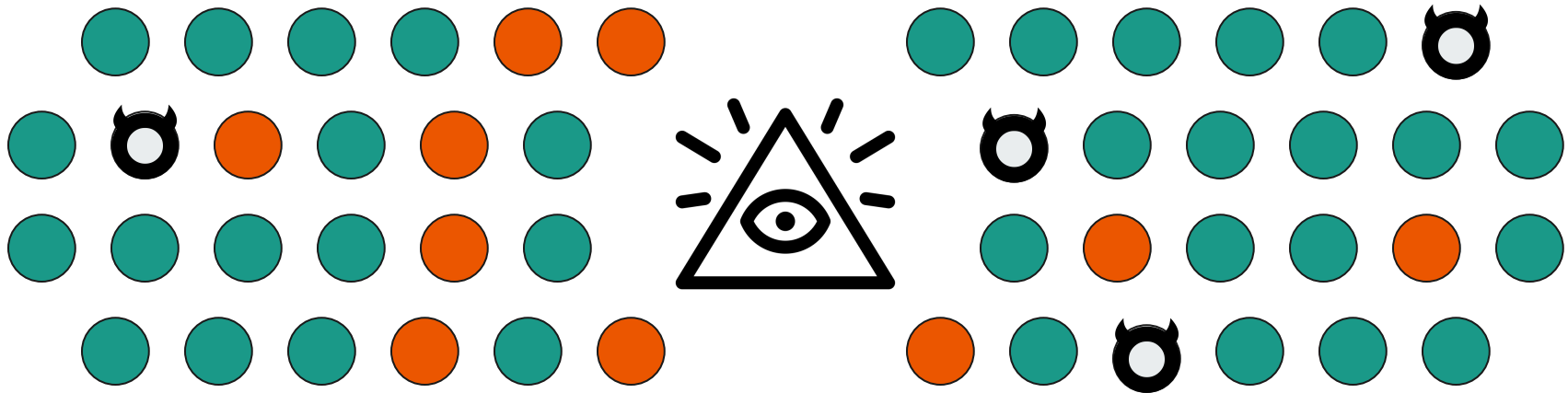
Processes..



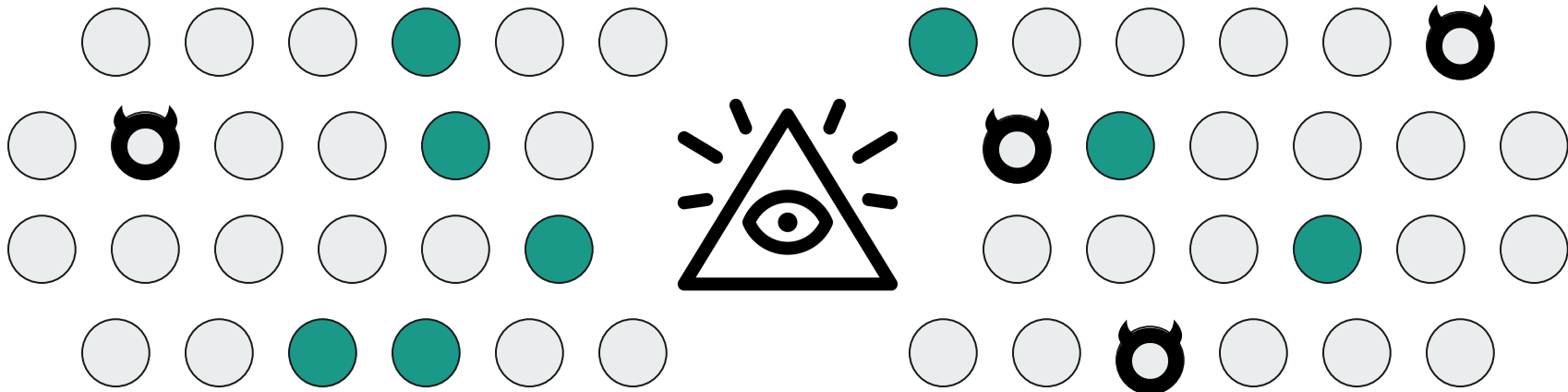
.. and a random oracle



Proposals..

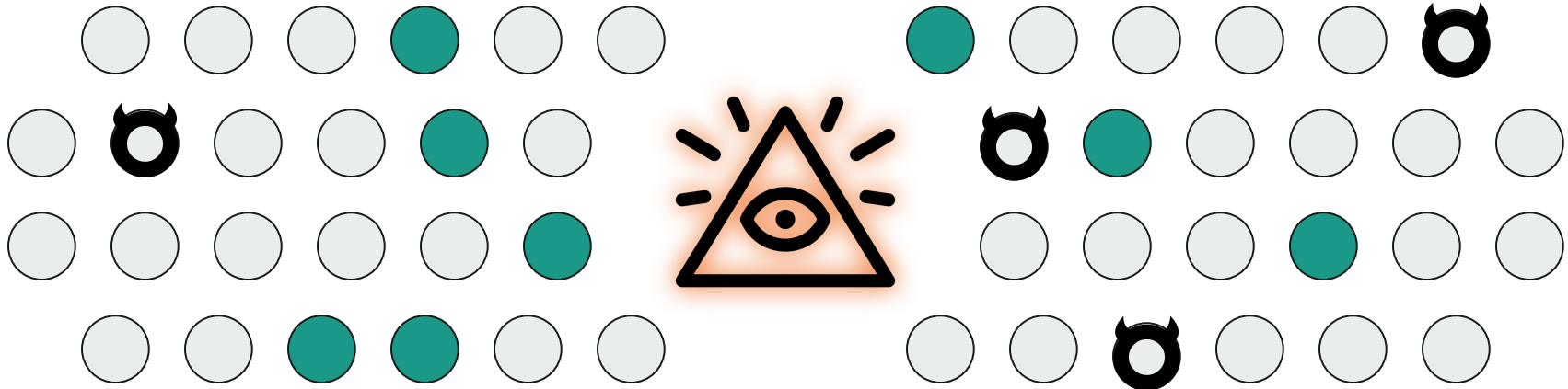


.. and broadcasts..



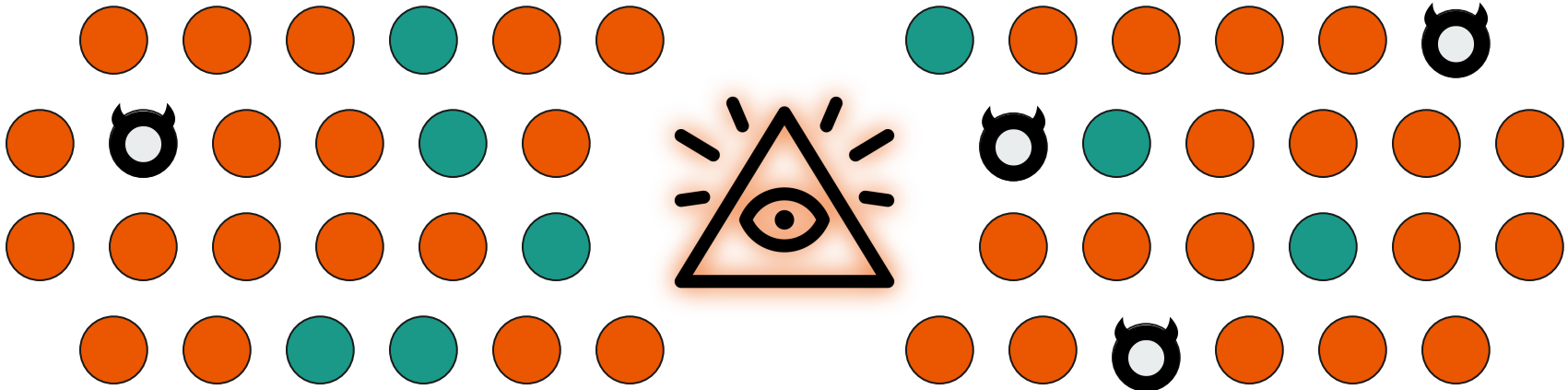


.. and coin flips..

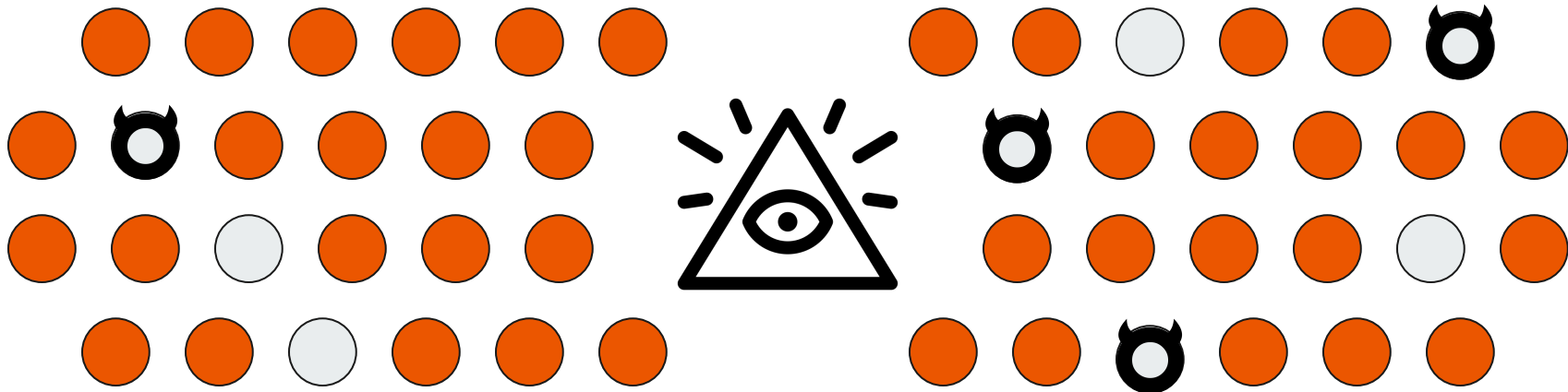




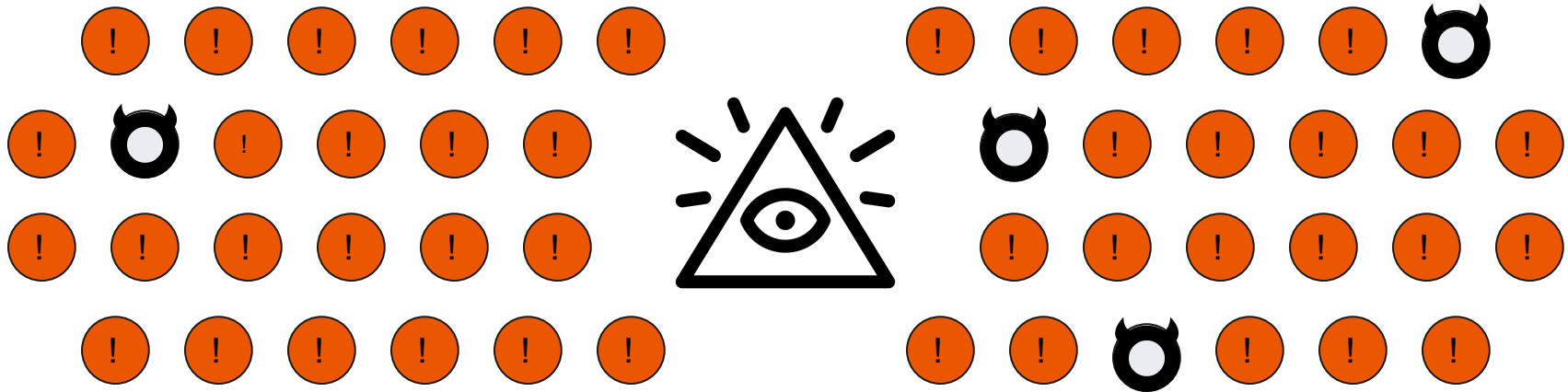
.. and coin flips..



.. and more broadcasts..

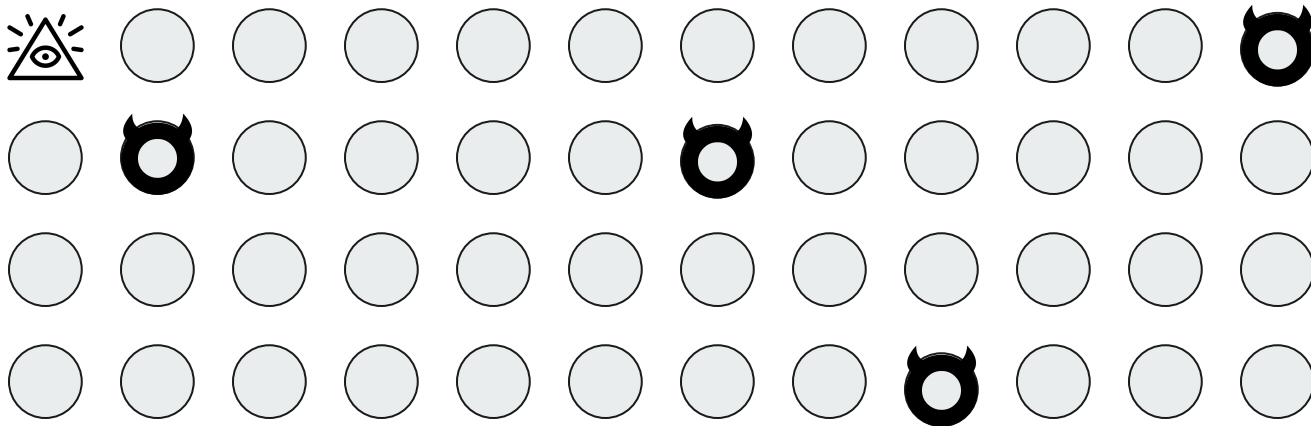


.. and decisions!

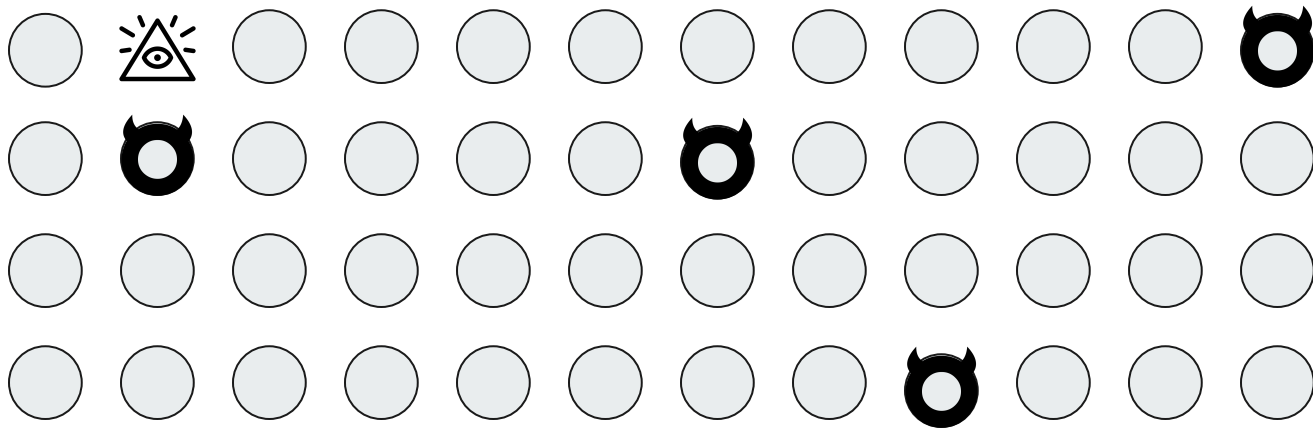


***Our goal is to scale
Common Coin***

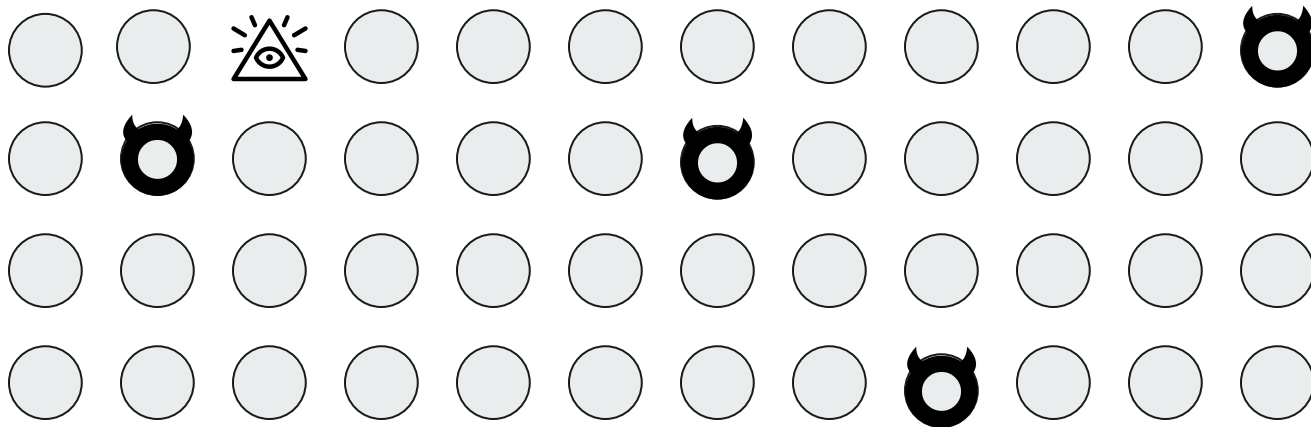
You choose!



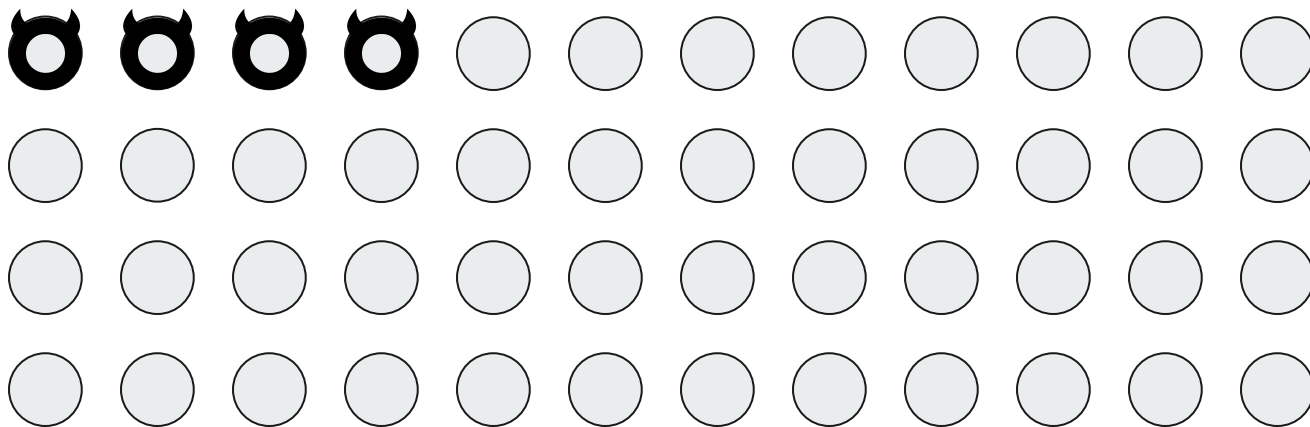
You choose!



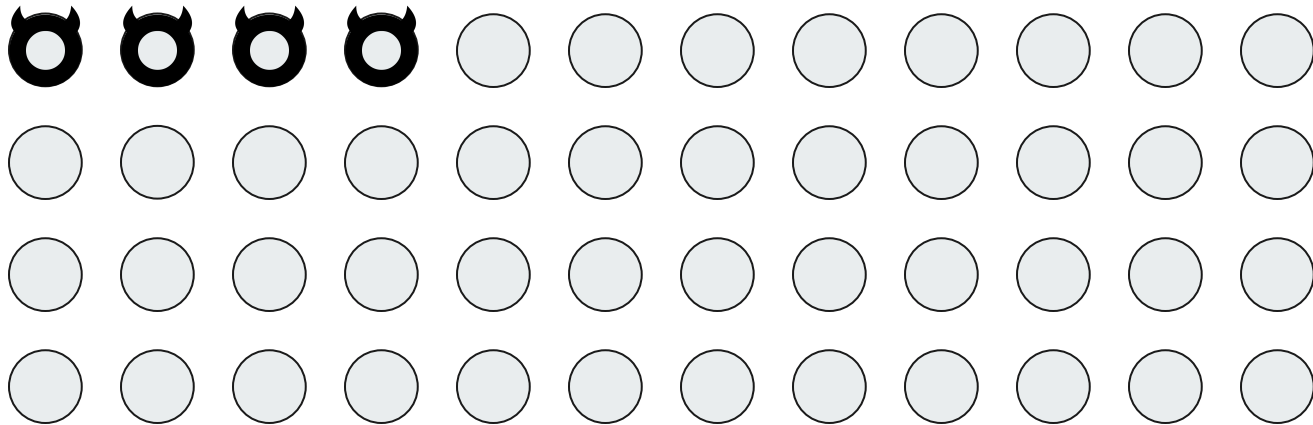
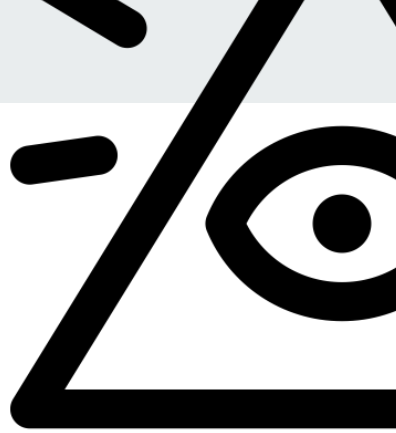
You choose!



You choose!



Pick.. at random..?





Verifiable Delay Functions

$$x \xrightarrow{\Delta} f(x)$$



Timeline





Timeline



Yo!



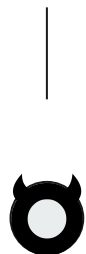
Timeline



Yo!



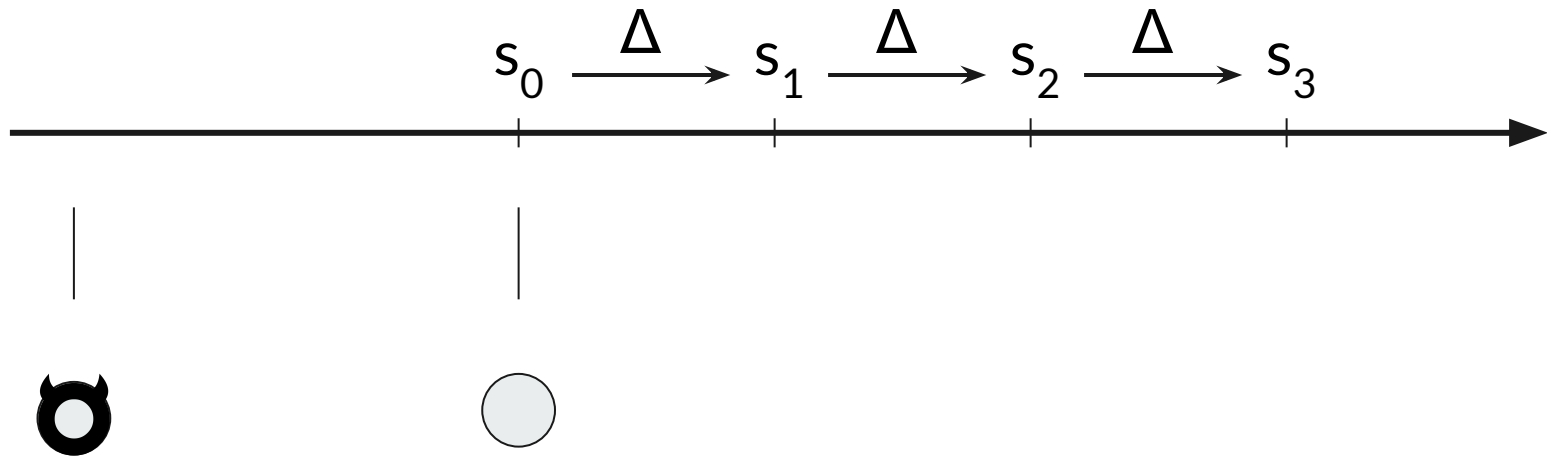
Timeline



Byzantine Grace Period

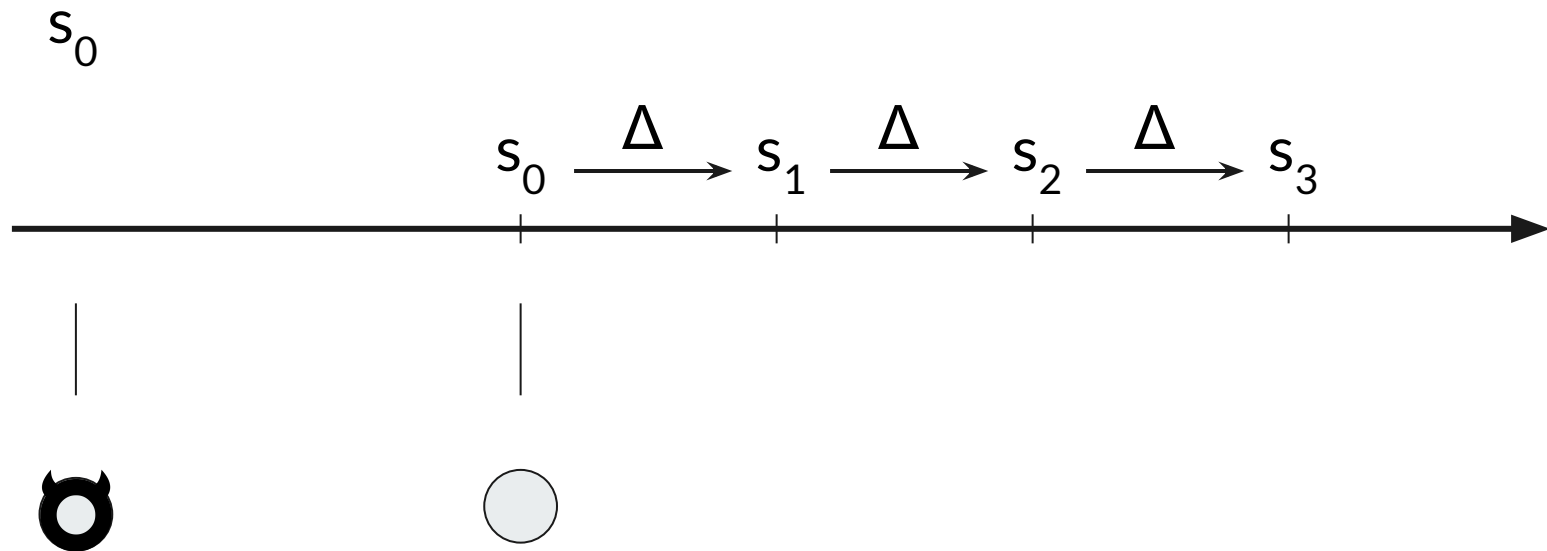


Precompute me..





Precompute me..





Precompute me..

$$s_0 \xrightarrow{\Delta} s_1$$

$$s_0 \xrightarrow{\Delta} s_1 \xrightarrow{\Delta} s_2 \xrightarrow{\Delta} s_3$$



.. if you can!

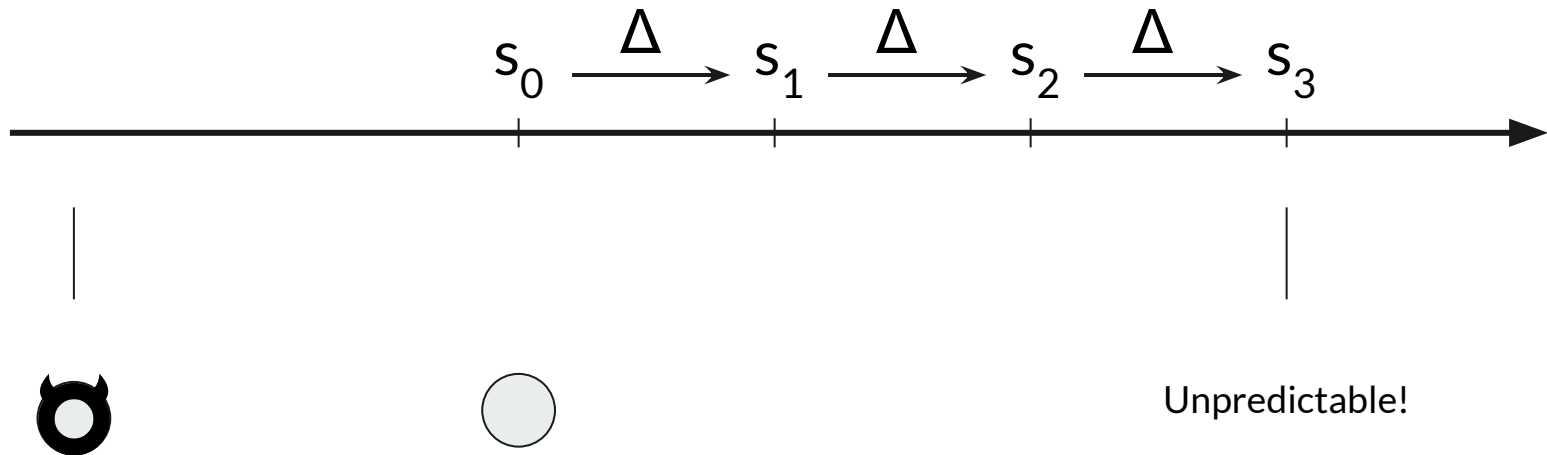
$$s_0 \xrightarrow{\Delta} s_1 \xrightarrow{\Delta} ??$$

$$s_0 \xrightarrow{\Delta} s_1 \xrightarrow{\Delta} s_2 \xrightarrow{\Delta} s_3$$





.. if you can!



Random



**Eventually
Unpredictable**

—



Questions?

Randomized algorithms
hold a promise for
planetary scale systems.

- 
- The origin.** Consensus-less Asset Transfer
 - The problem.** Byzantine Reliable Broadcast
 - The intuition.** Quorums vs. Samples
 - The challenge.** Games and Decorators
 - The future.** Towards a Planetary Database