


Starfish:

DAG-based Consensus via Encoded Cordial Dissemination

Nikita Polianskii (IOTA Foundation)

Joint work with **Sebastian Mueller** and **Ilya Vorobyev**

TUM Blockchain Conference, September 11-12, 2025.

IOTA

Outline


Consensus problem


DAG-based consensus

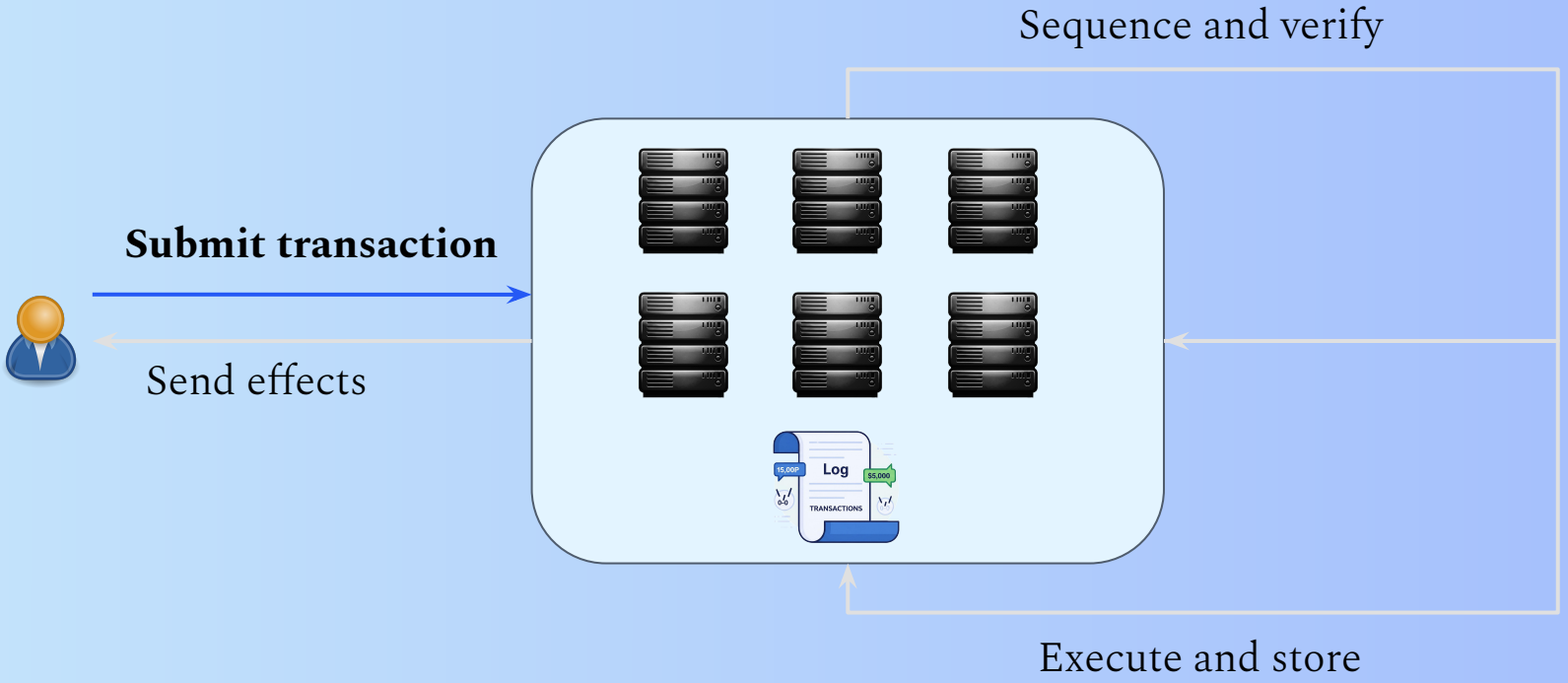

Uncertified DAG and motivation


Starfish

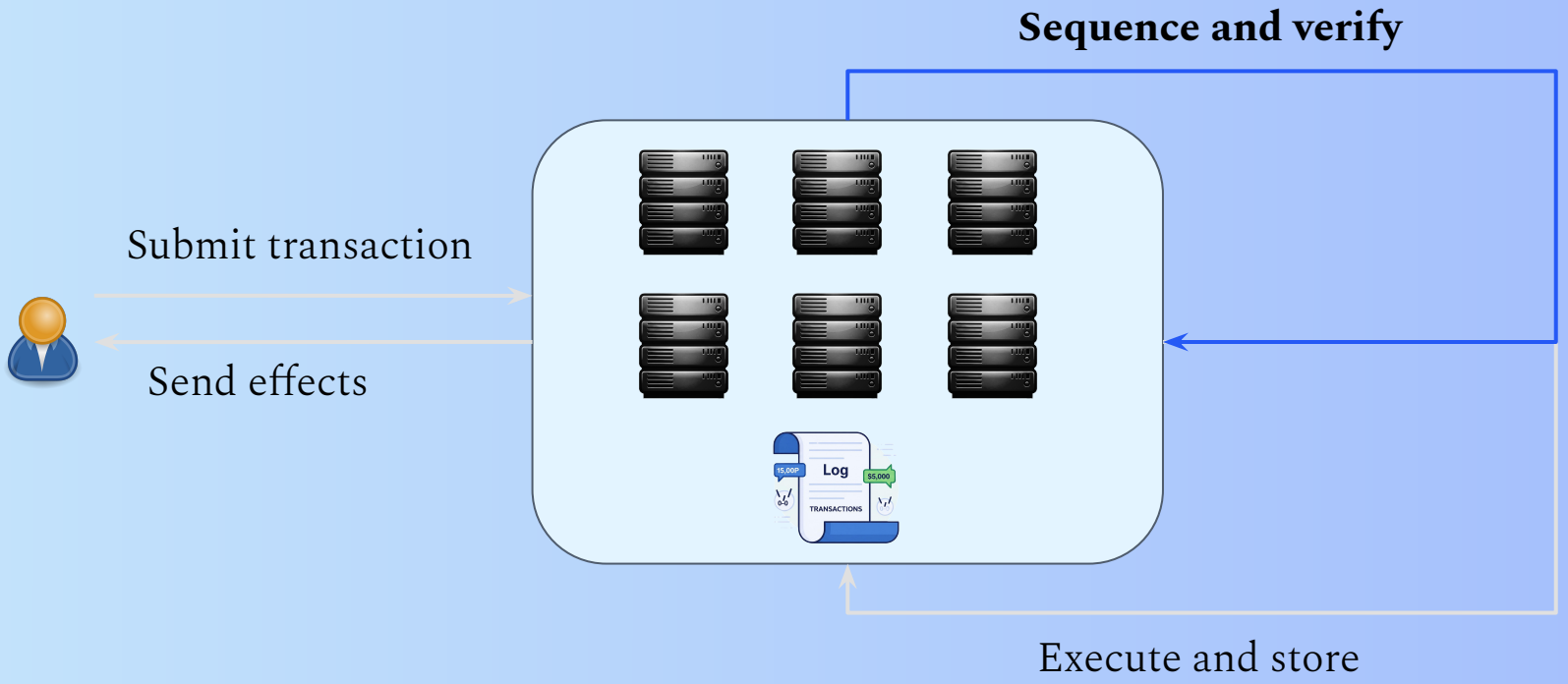

Performance and comparison

Consensus problem

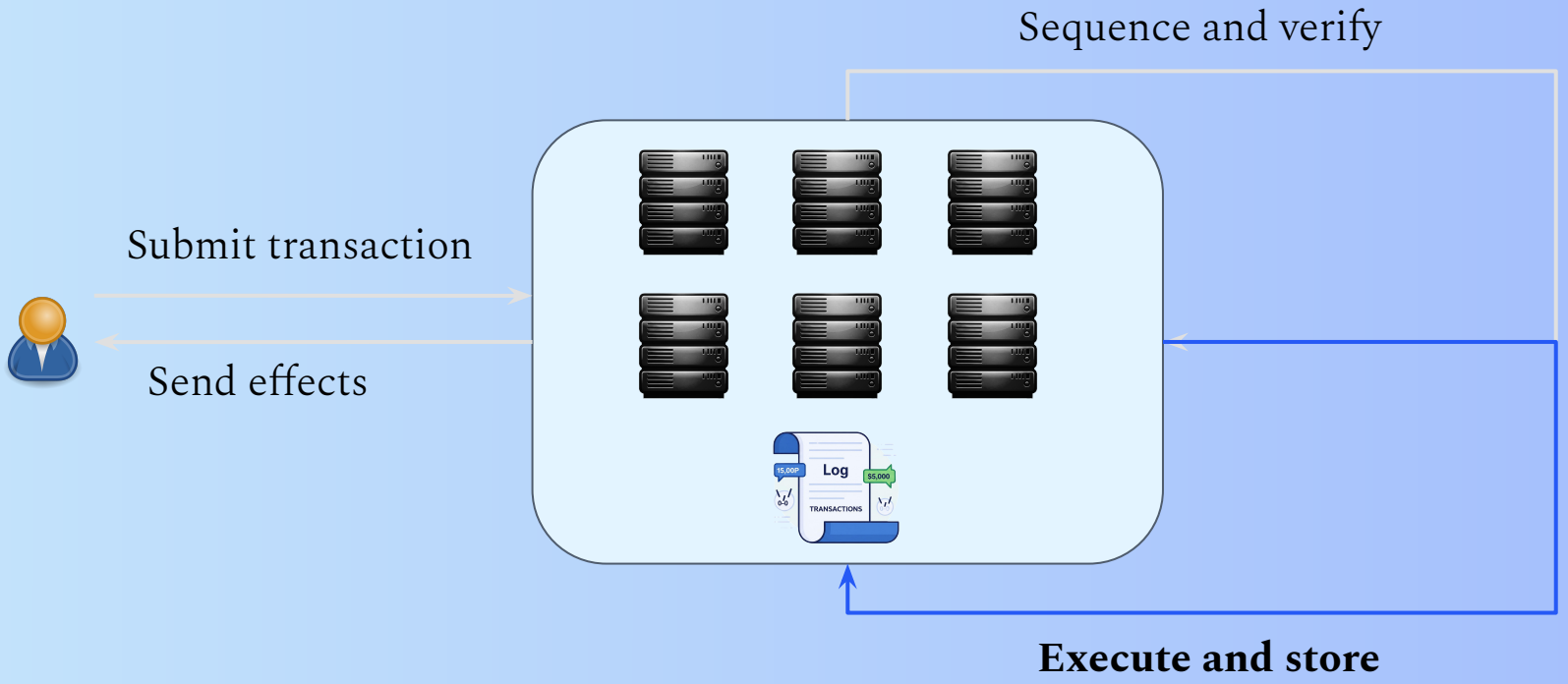
Consensus


Sequence and verify
Submit transaction
Send effects
Log
TRANSACTIONS
Execute and store

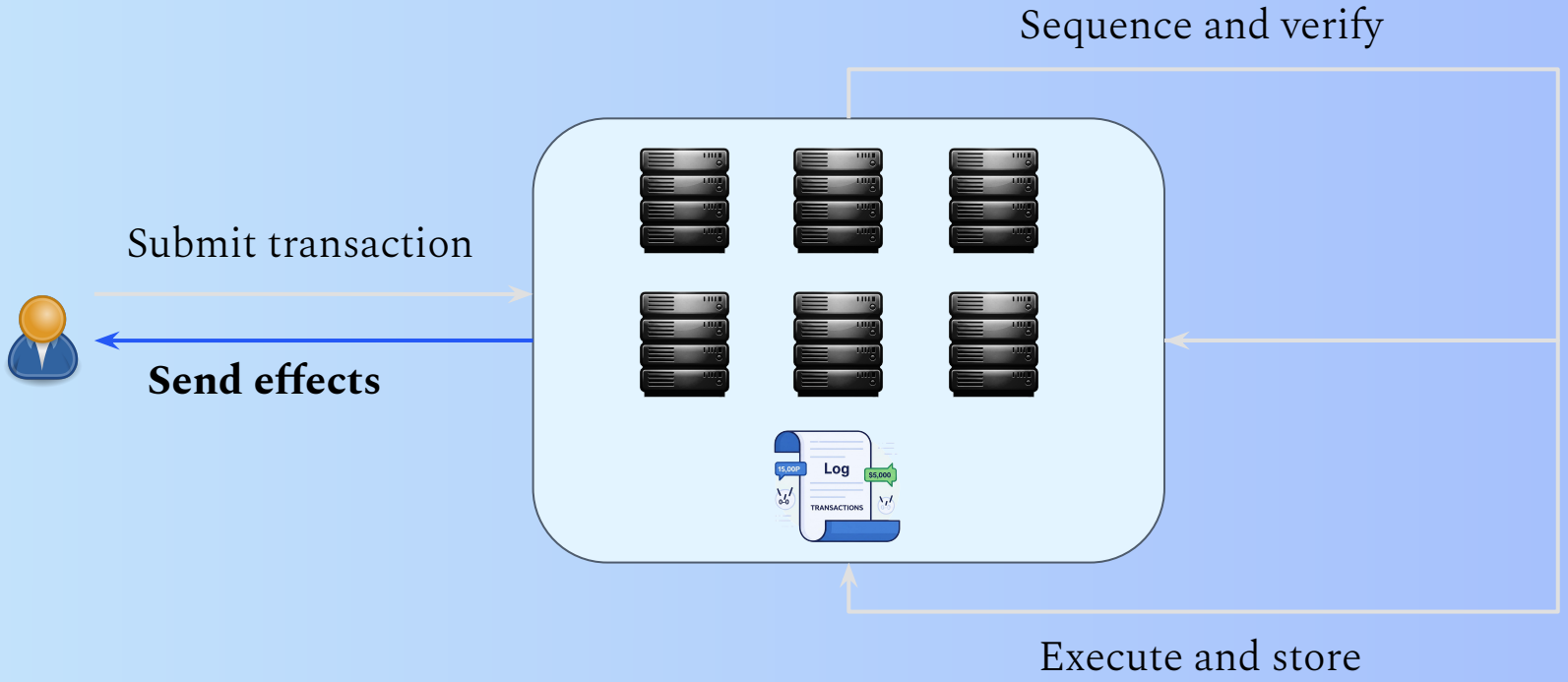
Consensus


Sequence and verify
Submit transaction
Send effects
Log
TRANSACTIONS
Execute and store

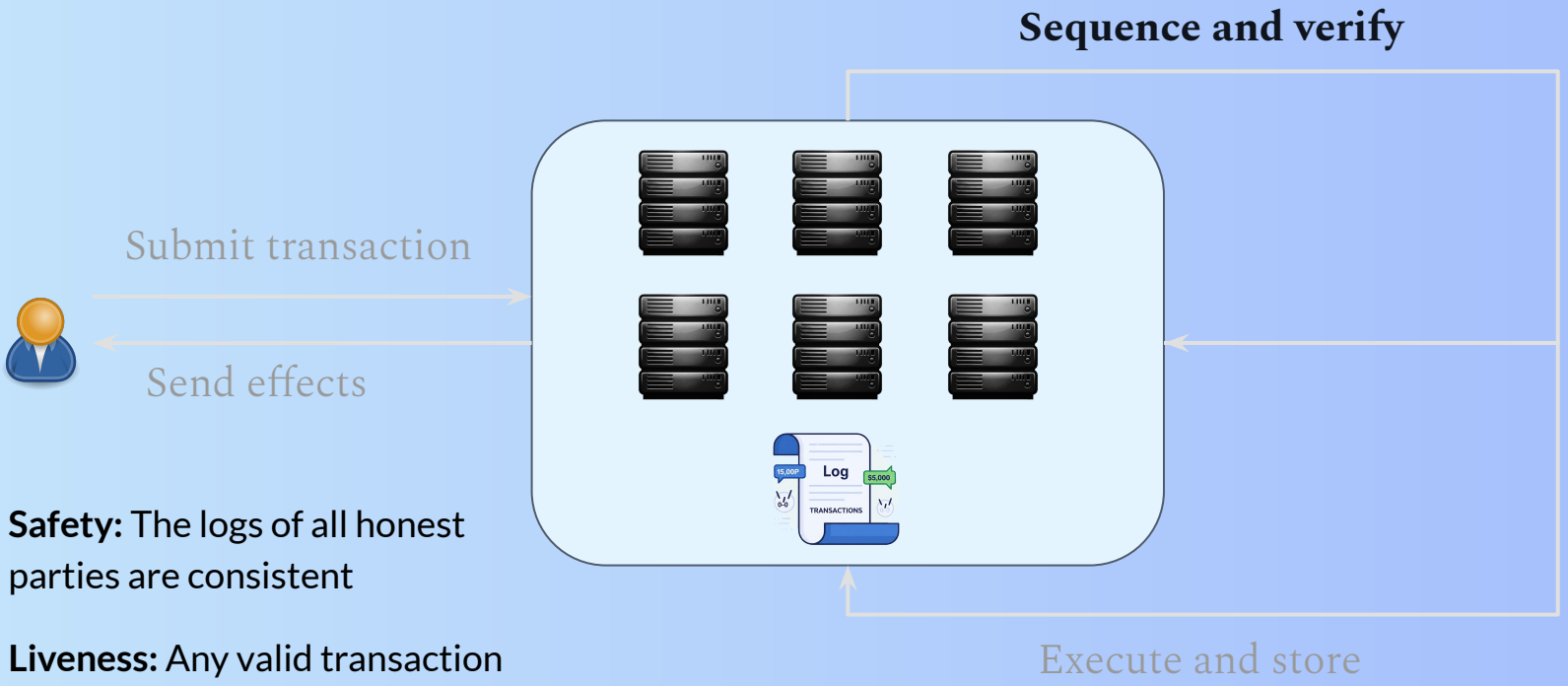
Consensus


Sequence and verify
Submit transaction
Send effects
Log
TRANSACTIONS
Execute and store

Consensus


Sequence and verify
Submit transaction
Send effects
Log
TRANSACTIONS
Execute and store

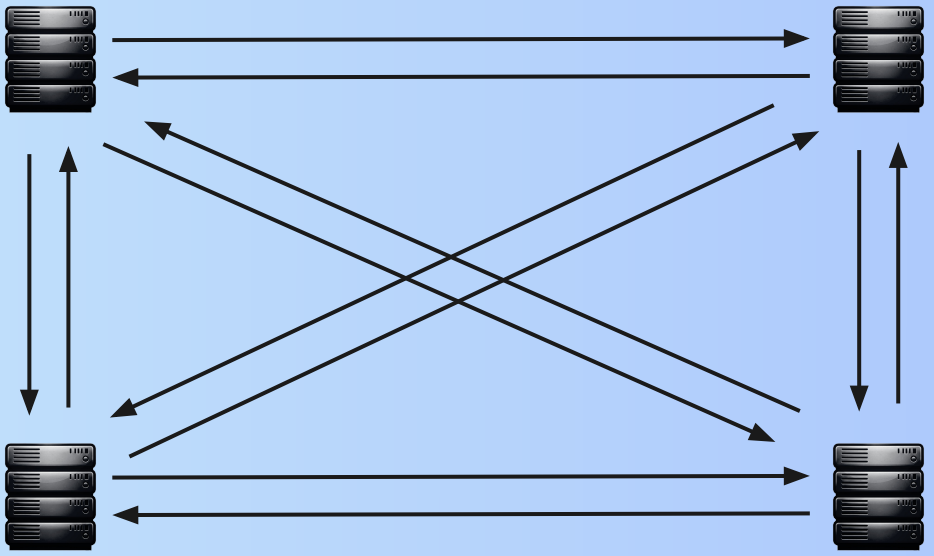
Consensus



Safety: The logs of all honest parties are consistent

Liveness: Any valid transaction submitted by an honest party is eventually included in the log.

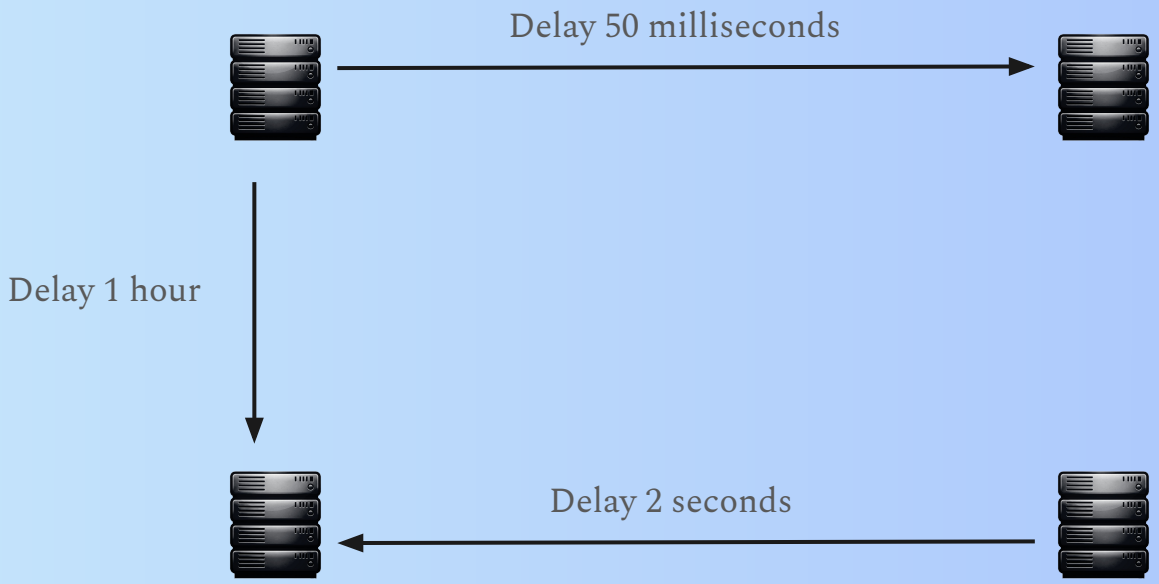
Network



Point-to-point communication

Each message is eventually delivered

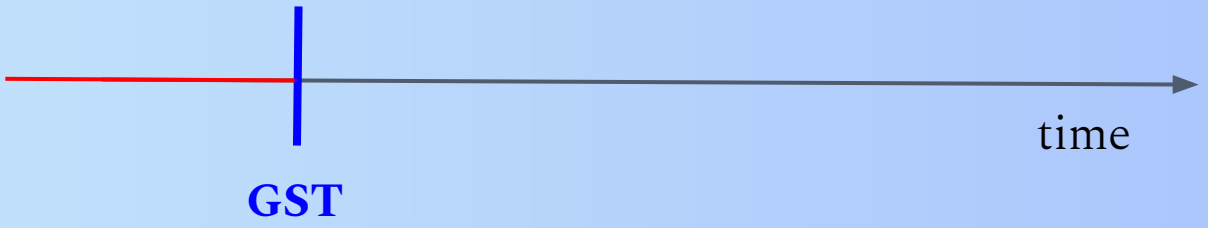
Network



Point-to-point communication

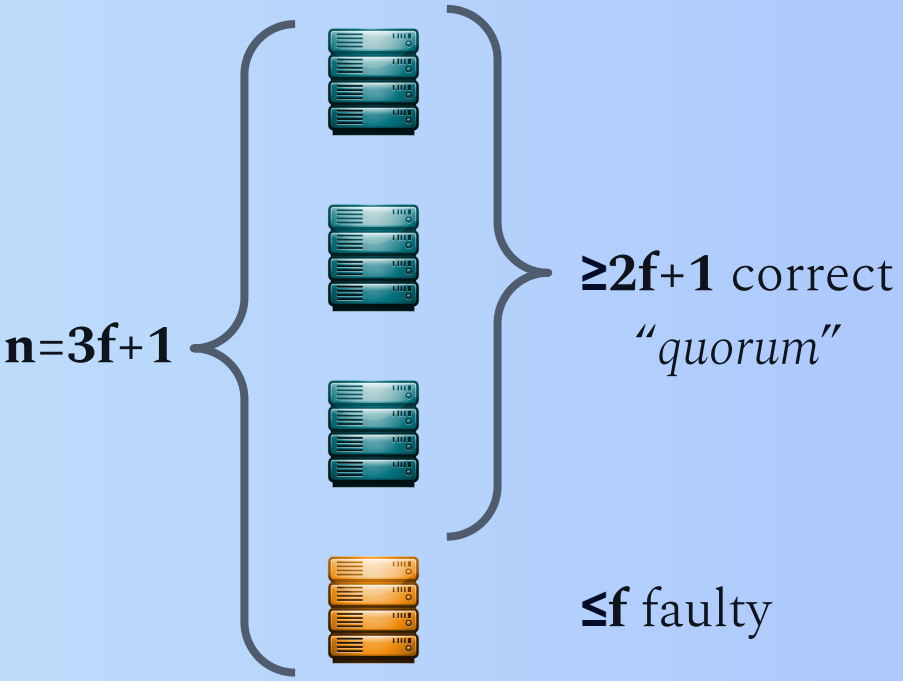
Each message is eventually delivered

Partially synchrony



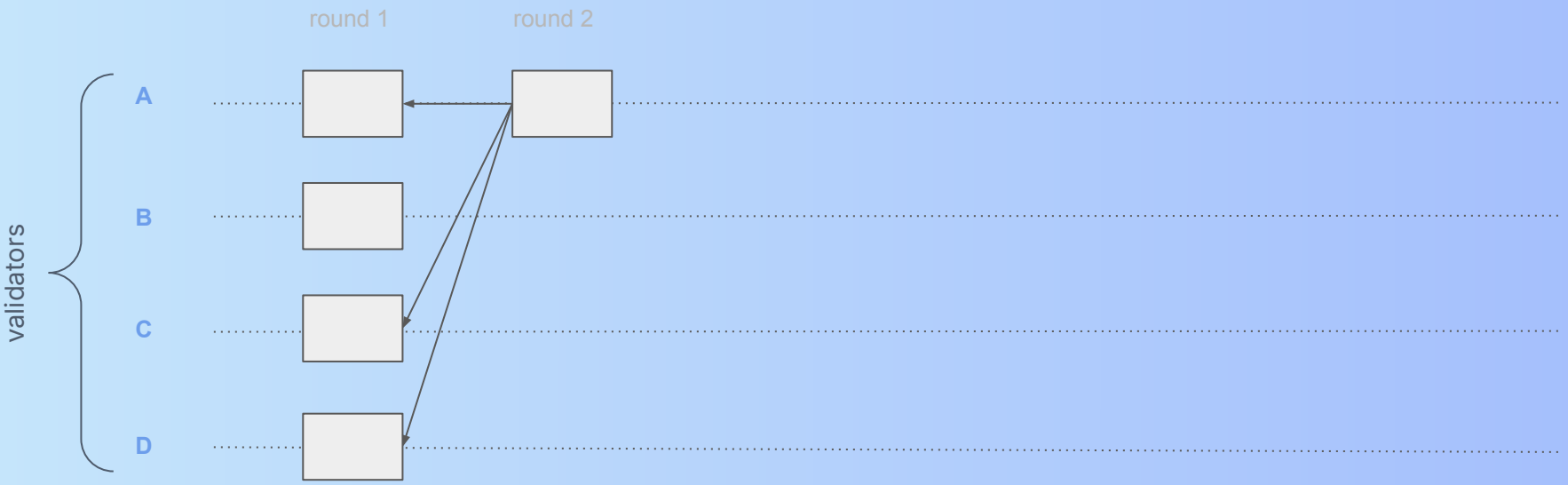
After *unknown* Global Stabilization Time (GST),
communication is synchronous with *known* delay **Δ**

Byzantine fault tolerance

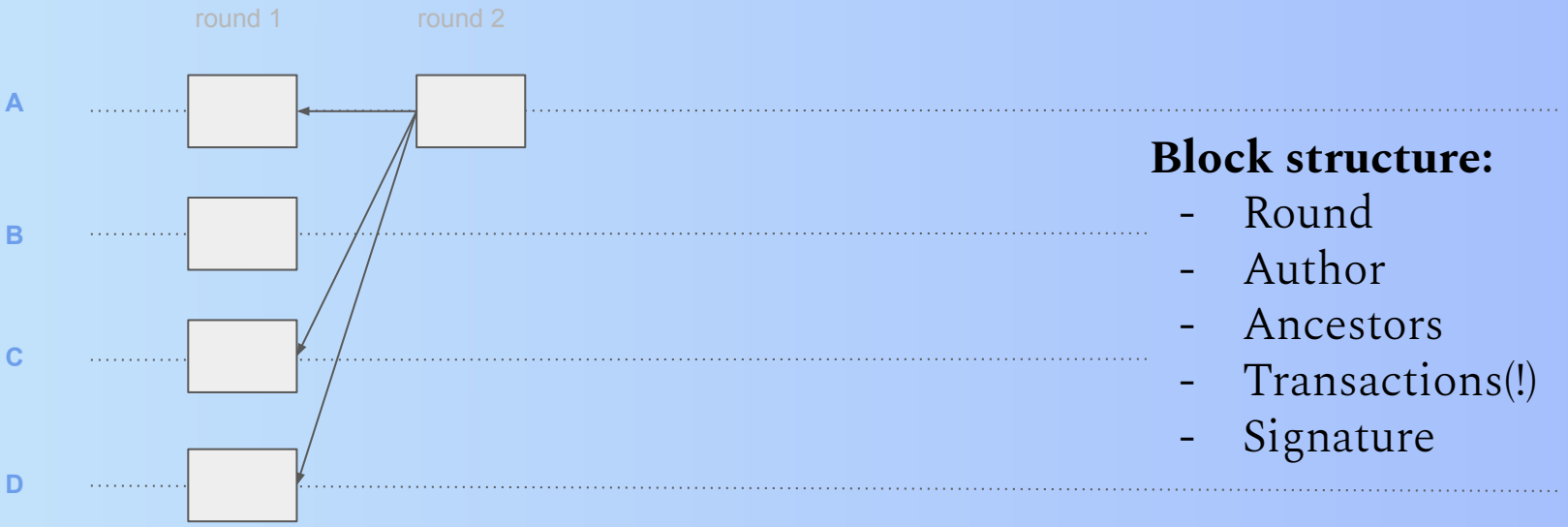


DAG-based consensus

DAG as communication and logical layer



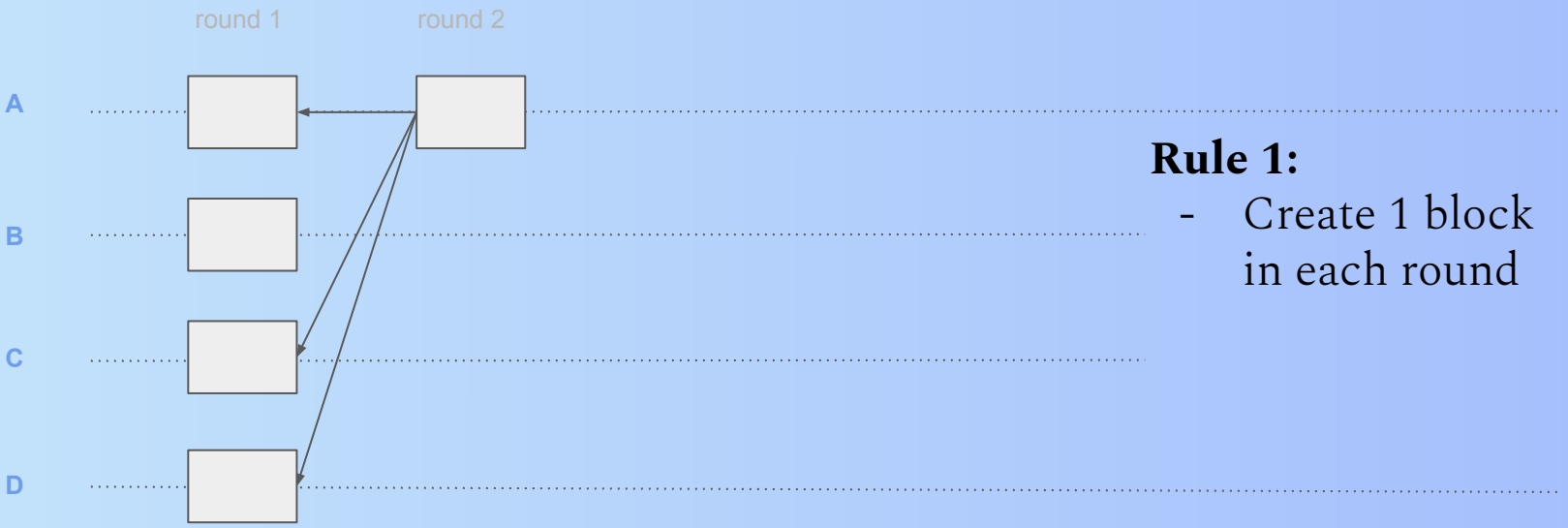
DAG as communication and logical layer



Block structure:

- Round
- Author
- Ancestors
- Transactions(!)
- Signature

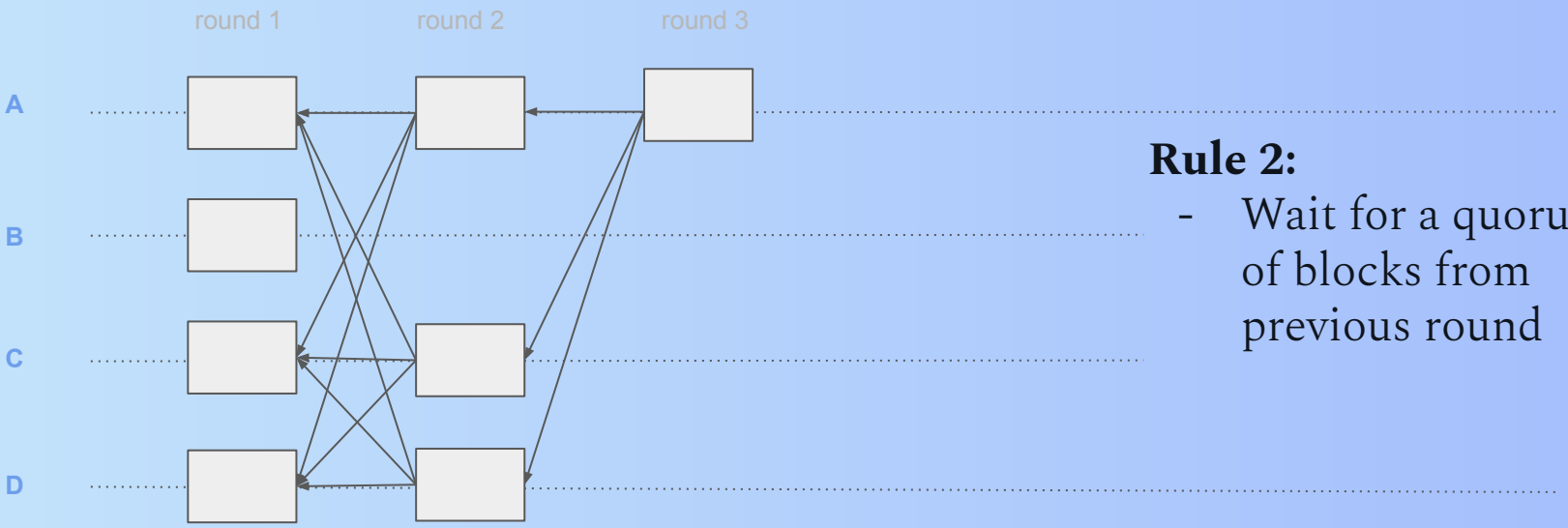
DAG as communication and logical layer



Rule 1:

- Create 1 block in each round

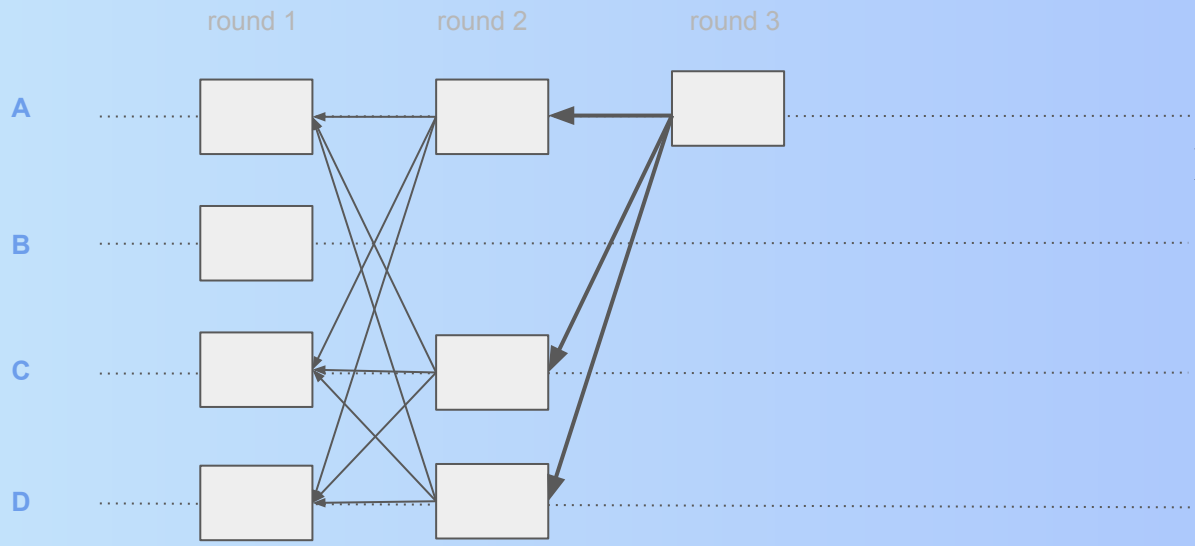
DAG as communication and logical layer



Rule 2:

- Wait for a quorum of blocks from previous round

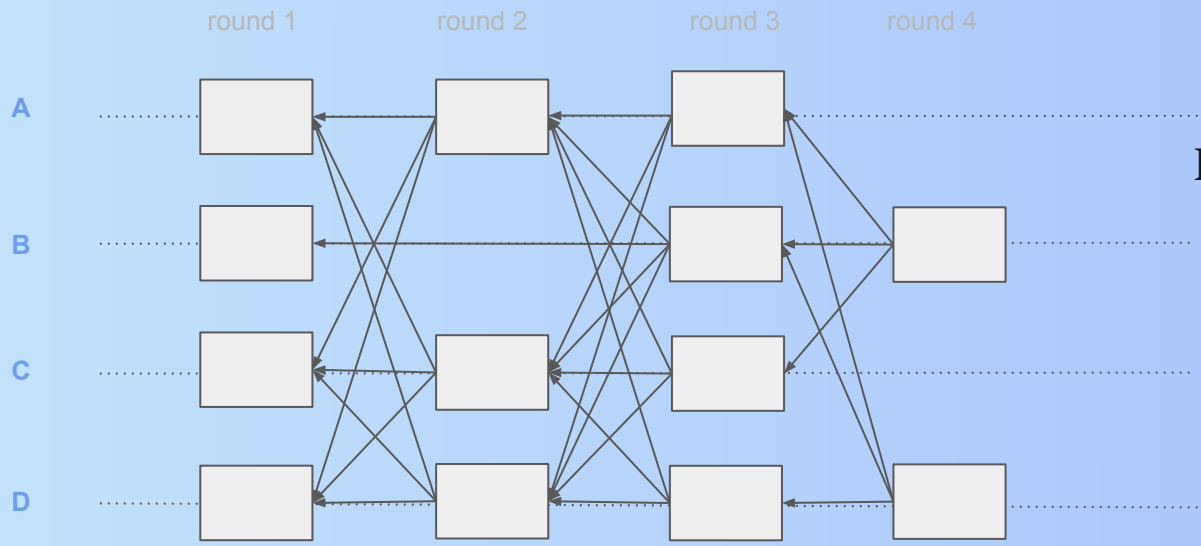
DAG as communication and logical layer



Rule 2:

- Wait for a quorum of blocks from previous round

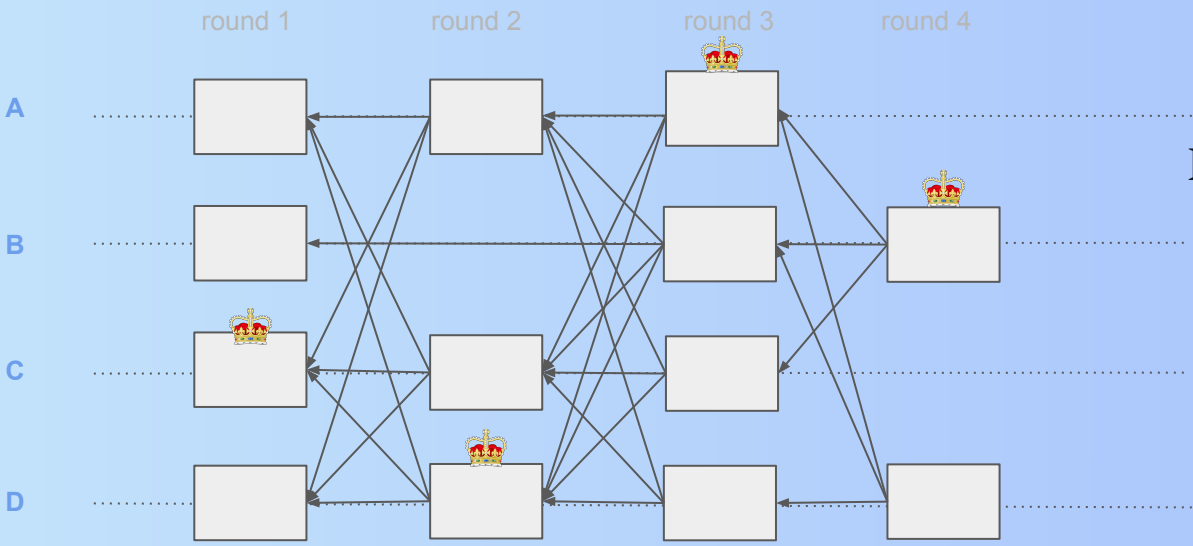
DAG as communication and logical layer



Rule 3:

- Wait for a leader in the previous round

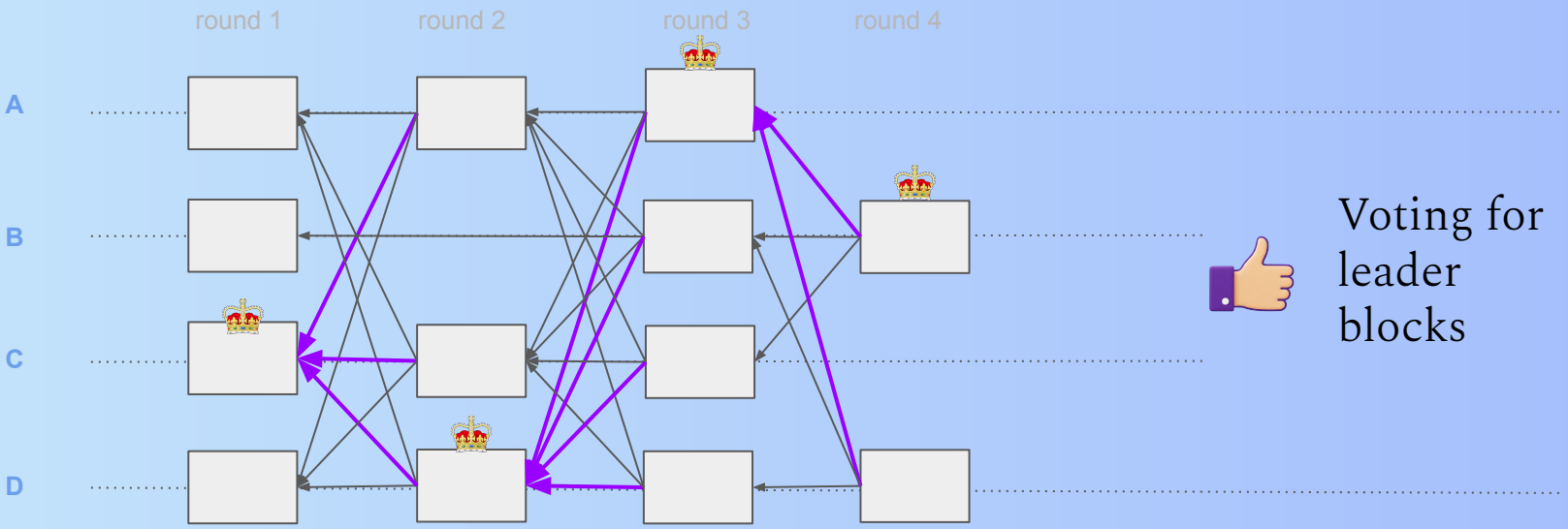
DAG as communication and logical layer



Rule 3:

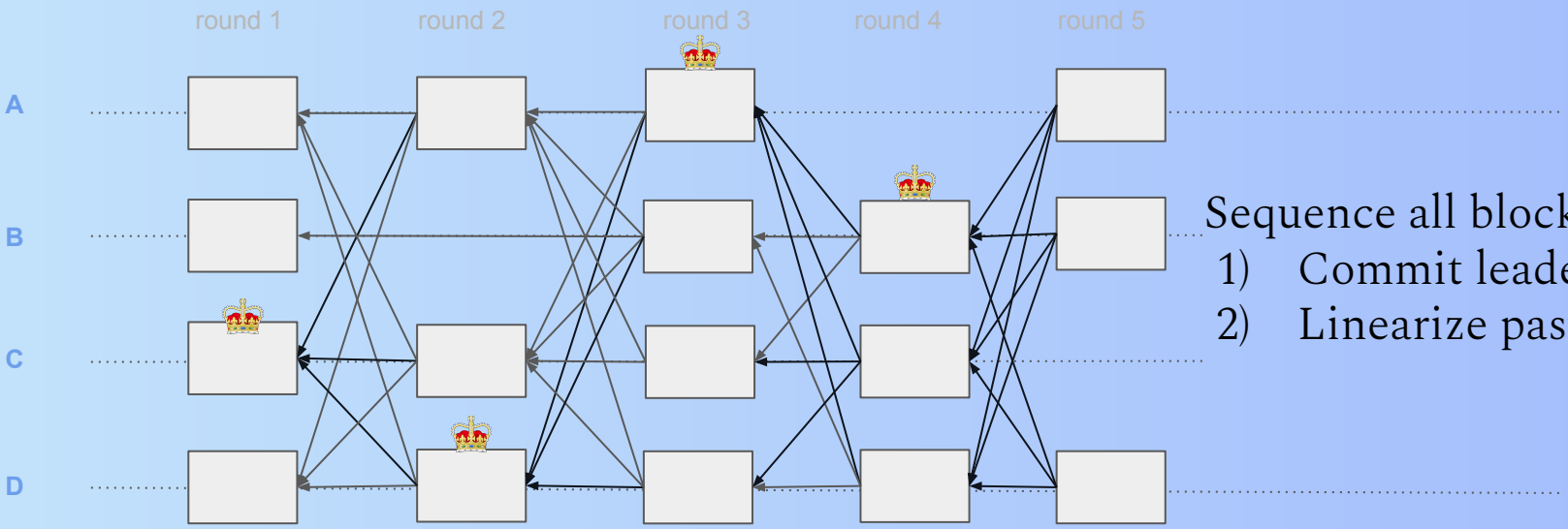
- Wait for a leader in the previous round

DAG as communication and logical layer



Voting for leader blocks

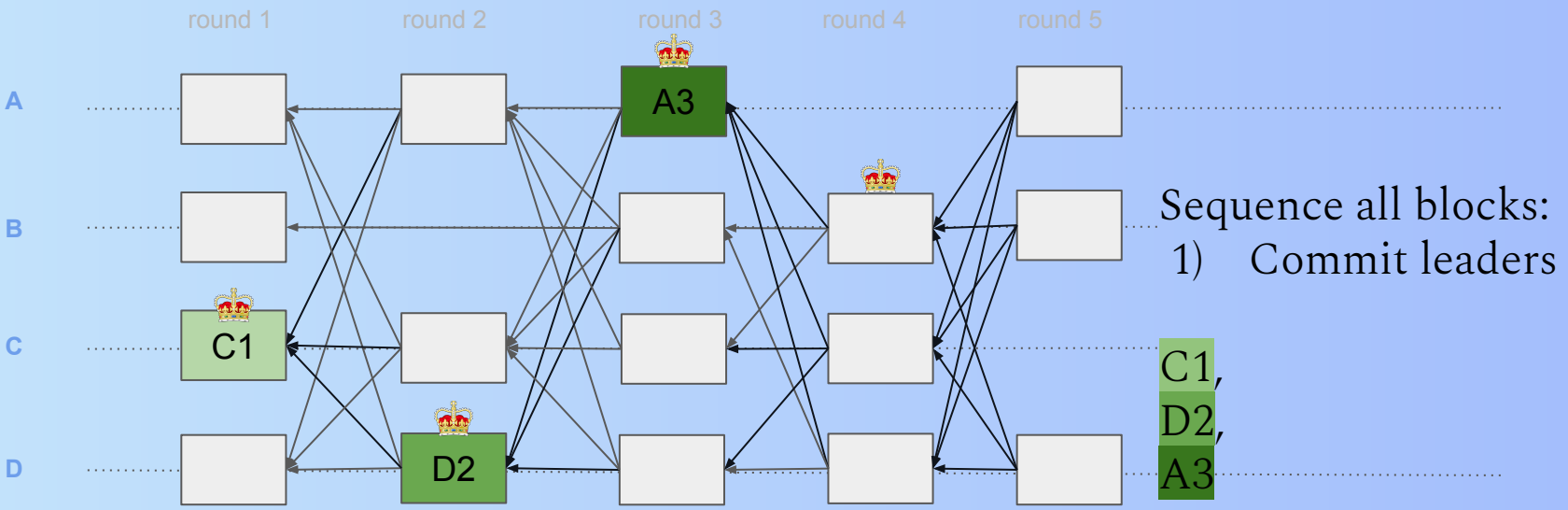
DAG as communication and logical layer



Sequence all blocks:

1) Commit leaders
2) Linearize past

DAG as communication and logical layer

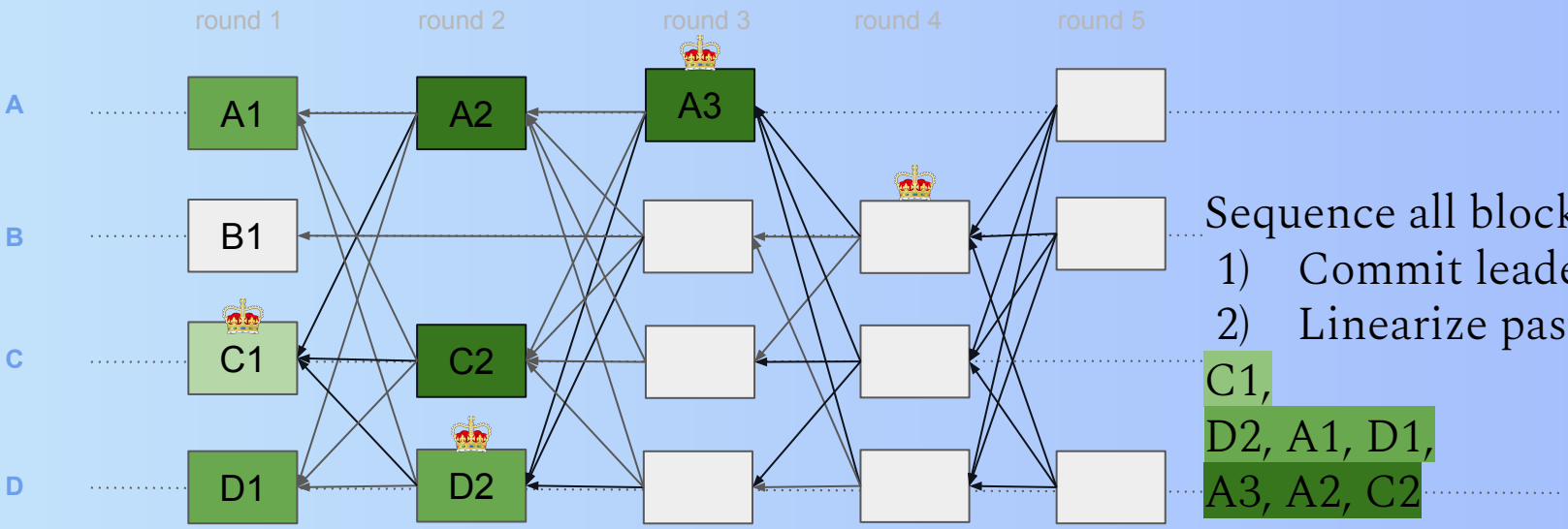


Sequence all blocks:

1) Commit leaders

C1,
D2,
A3

DAG as communication and logical layer

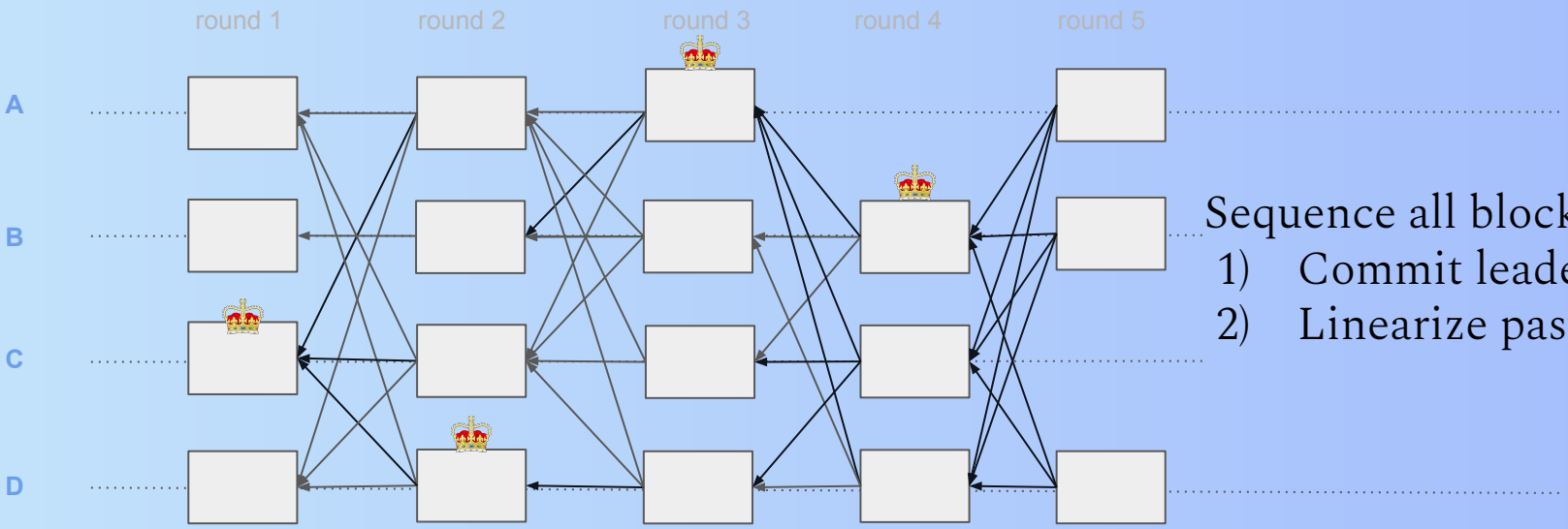


Sequence all blocks:

1) Commit leaders
2) Linearize past

C1,
D2, A1, D1,
A3, A2, C2

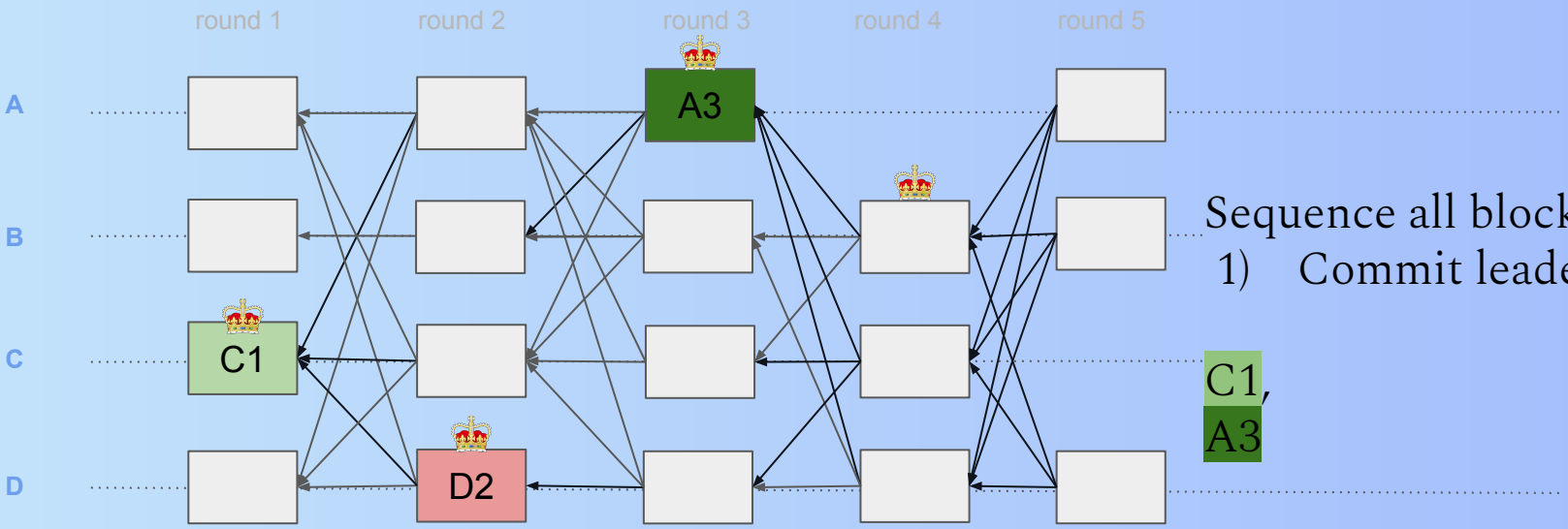
DAG as communication and logical layer



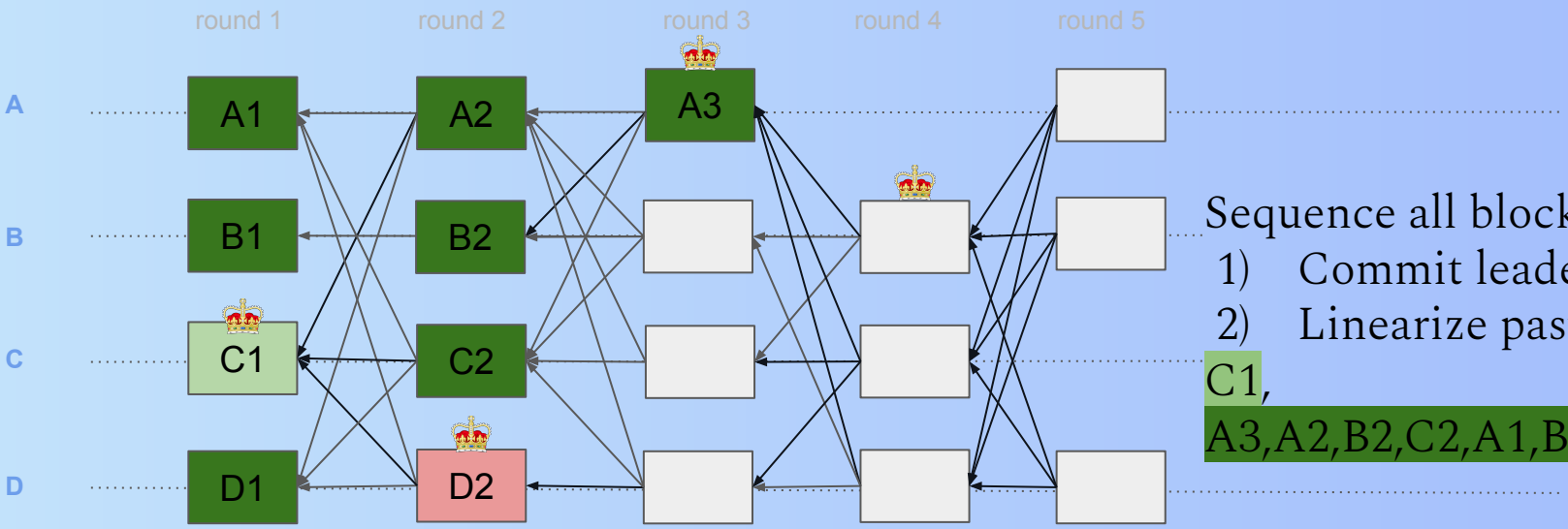
Sequence all blocks:

1) Commit leaders
2) Linearize past

DAG as communication and logical layer



DAG as communication and logical layer



Sequence all blocks:

1) Commit leaders
2) Linearize past

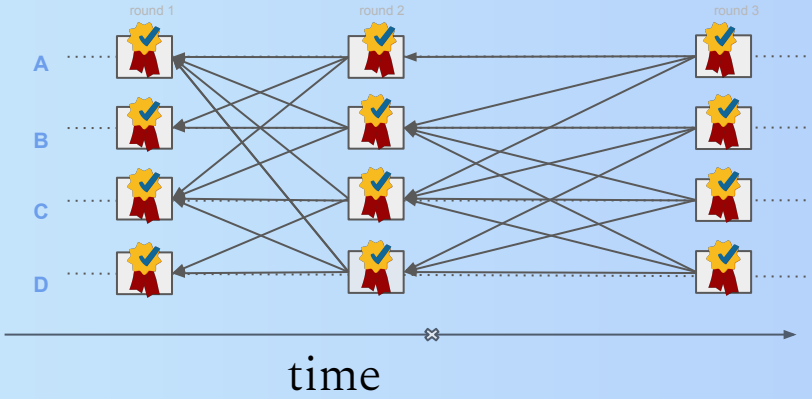
C1,

A3,A2,B2,C2,A1,B1,A1

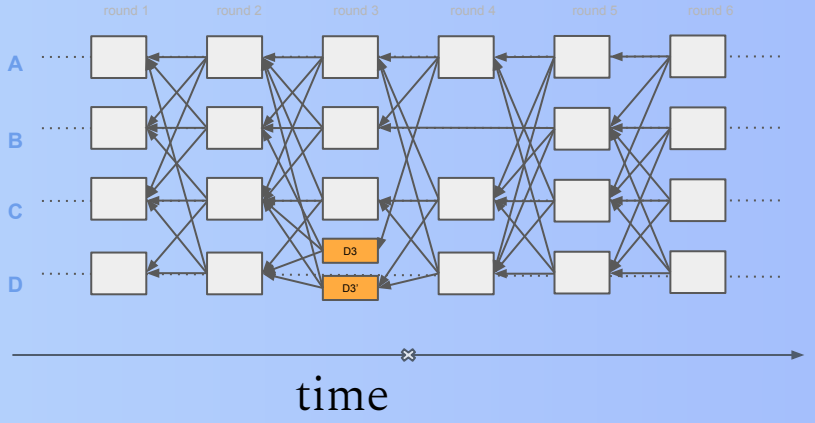
Uncertified DAG and motivation

Certified DAG vs Uncertified DAG

Each block includes a **certificate** — a quorum of 2f+1 signatures confirming its availability and uniqueness.



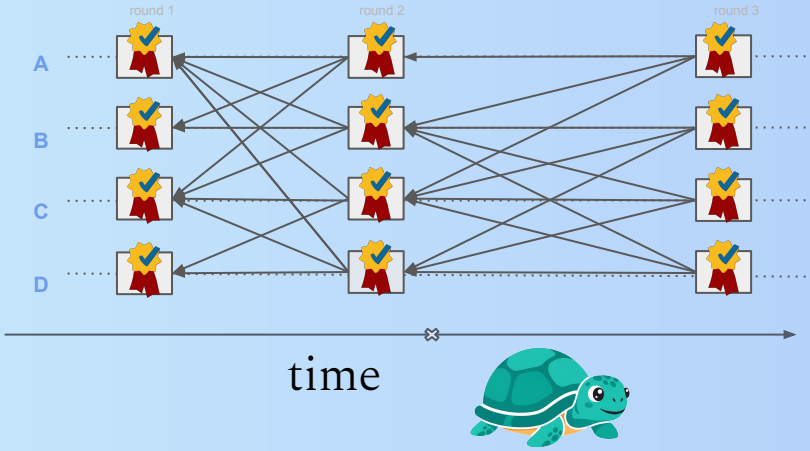
Each block is added **optimistically** after verifying the correctness of its causal past.



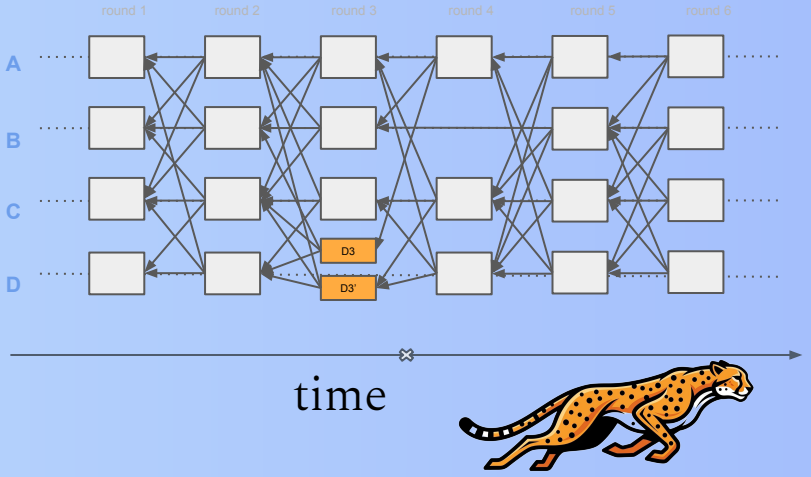
IOTA

Certified DAG vs Uncertified DAG

Each block includes a **certificate** — a quorum of 2f+1 signatures confirming its availability and uniqueness.



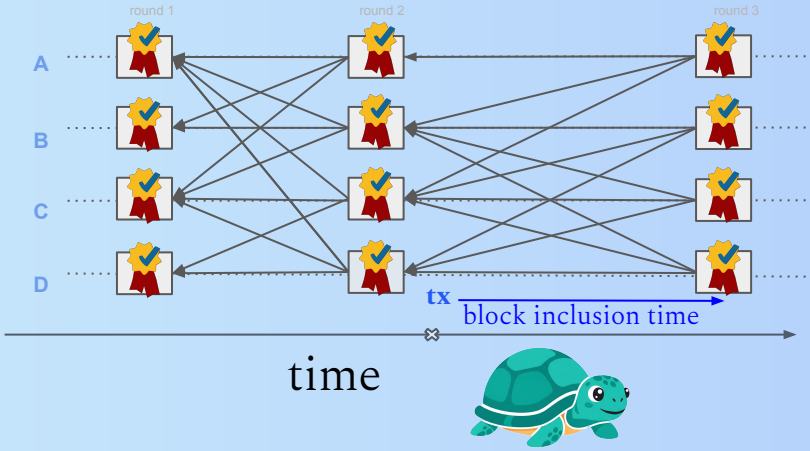
Each block is added **optimistically** after verifying the correctness of its causal past.



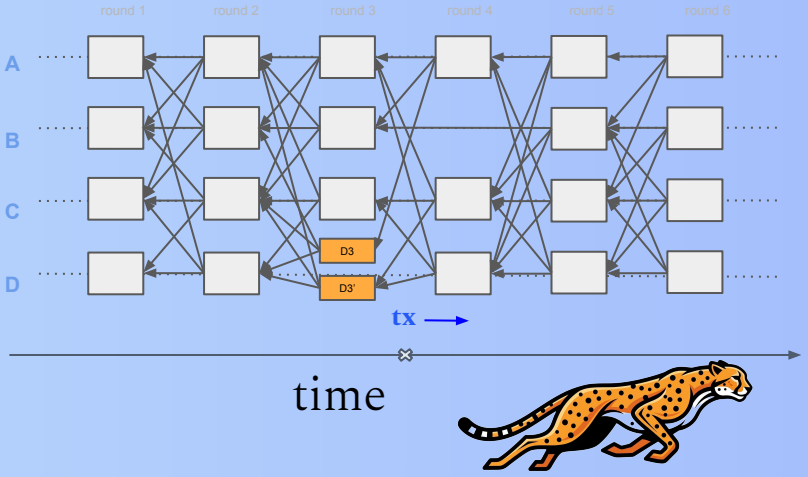
IOTA

Certified DAG vs Uncertified DAG

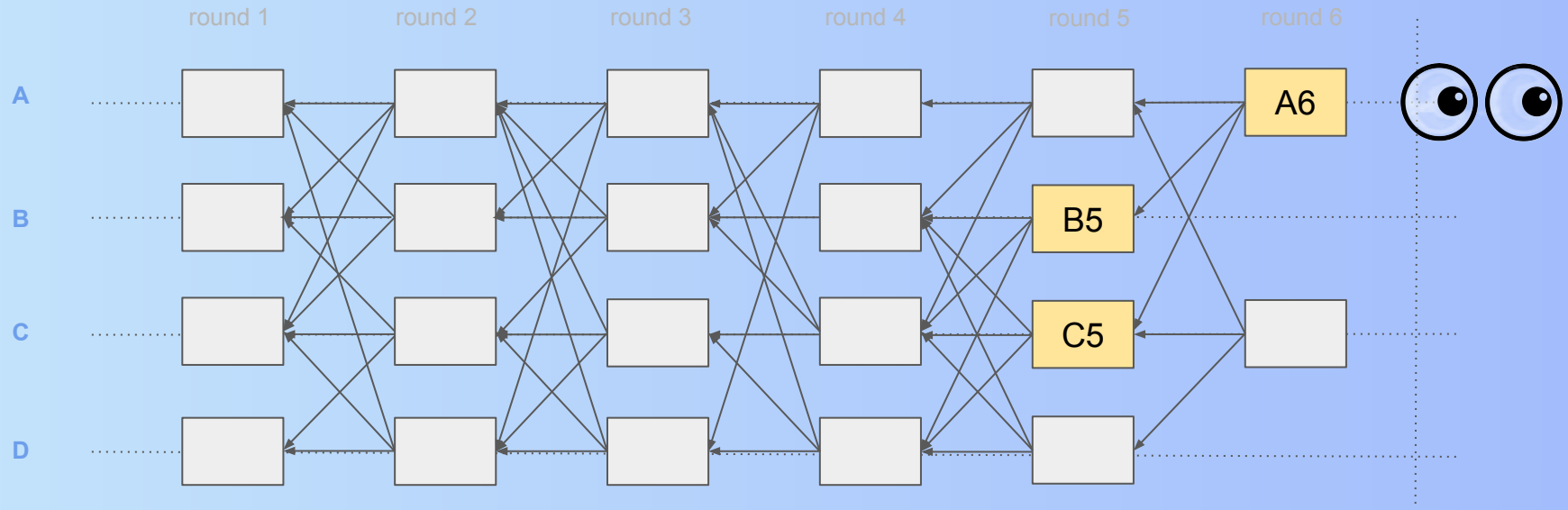
Each block includes a **certificate** — a quorum of 2f+1 signatures confirming its availability and uniqueness.



Each block is added **optimistically** after verifying the correctness of its causal past.



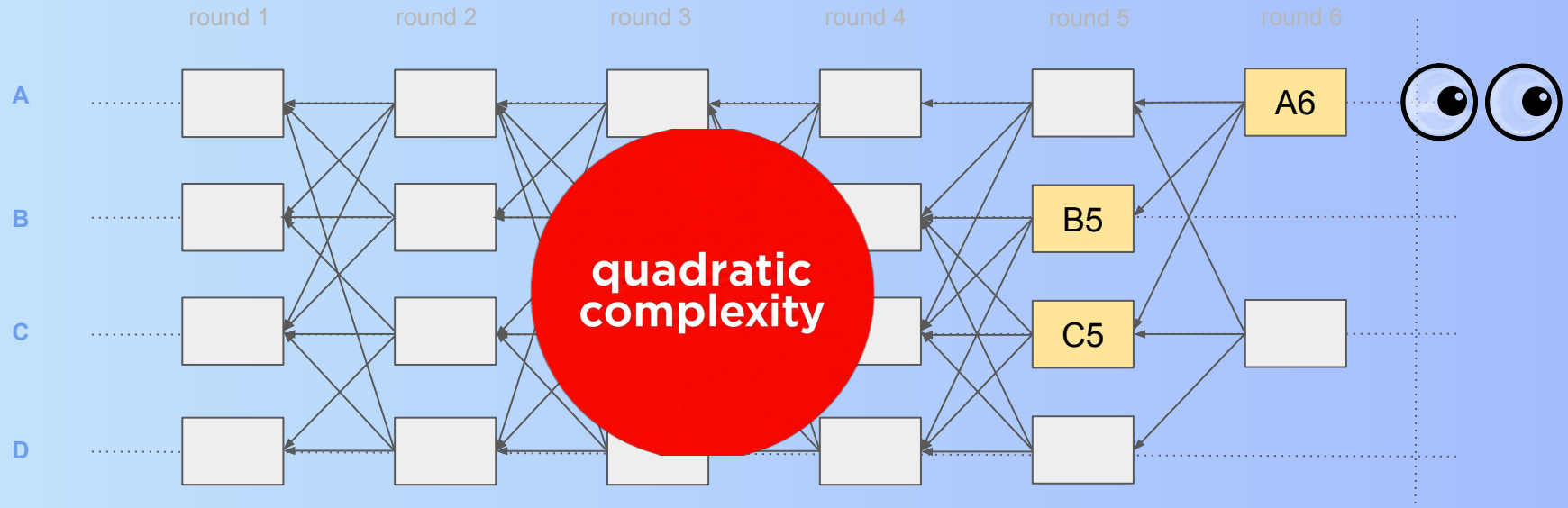
Broadcasting blocks (theory)



``**Push** to others blocks you know and think they need..." [†]
(example: A will send yellow blocks to D)

[†] *Cordial Miners*. DISC 2023

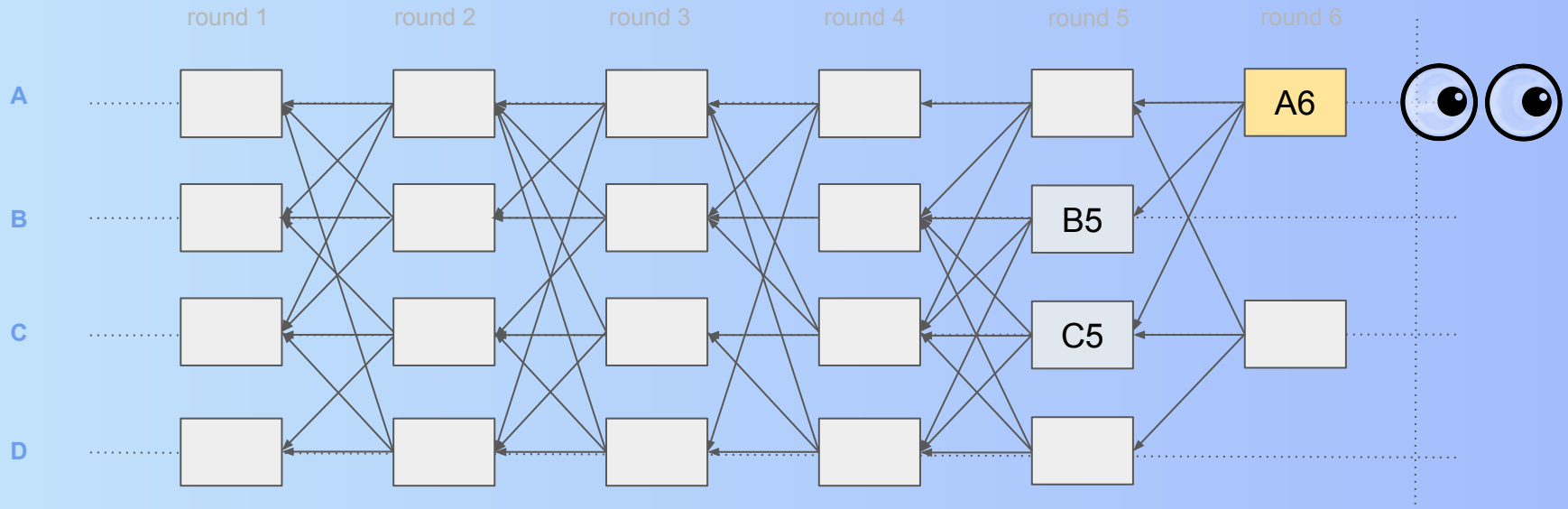
Broadcasting blocks (theory)



``**Push** to others blocks you know and think they need..." [†]
(example: A will send yellow blocks to D)

[†] *Cordial Miners*. DISC 2023

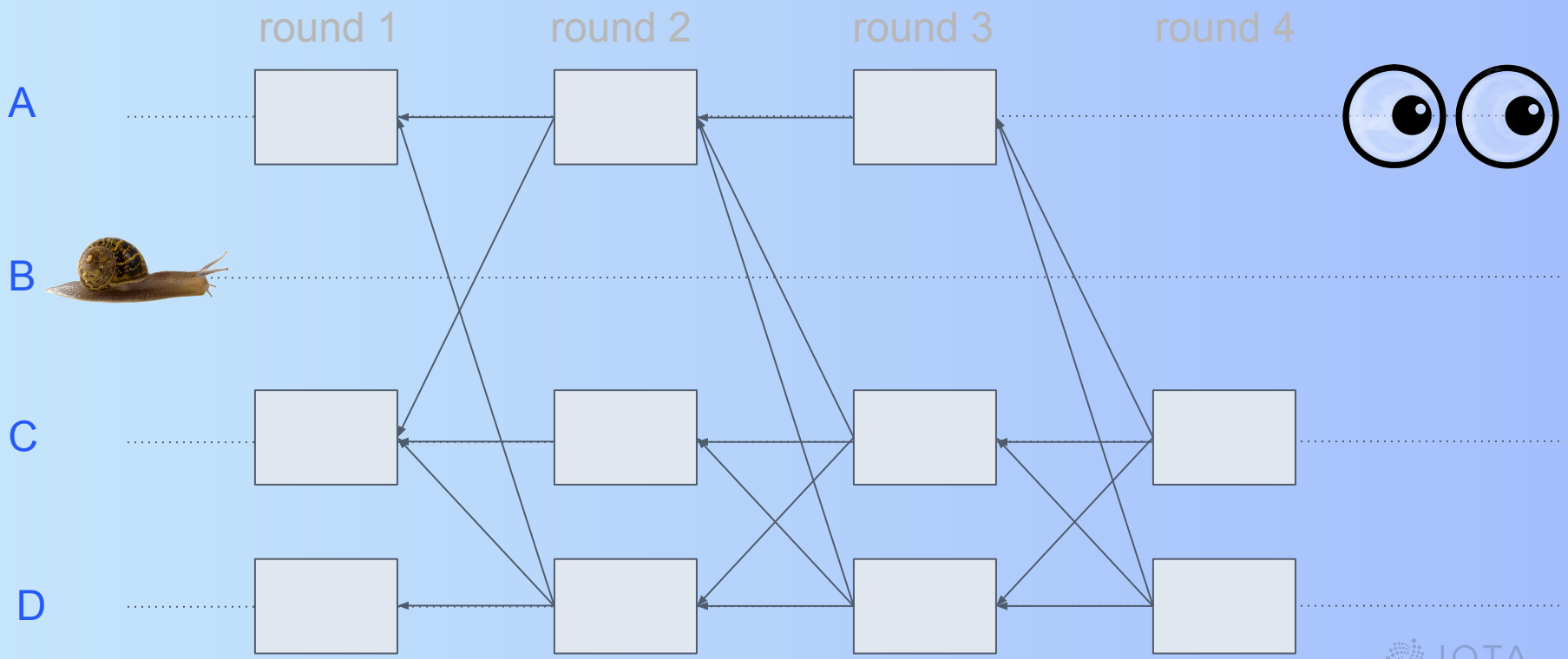
Broadcasting blocks (implementation)



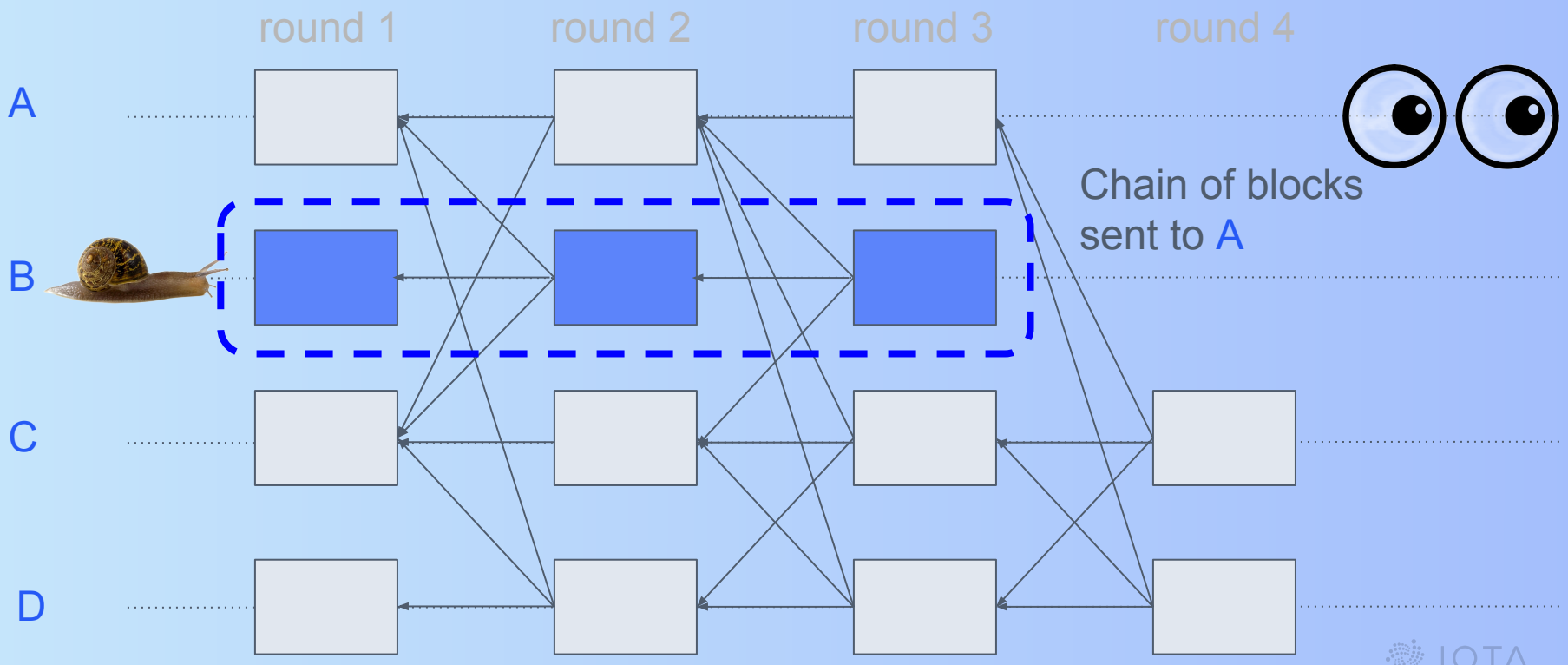
``Send to others only your own blocks and **pull** the unknown ancestors..."

Mysticeti. NDSS 2025

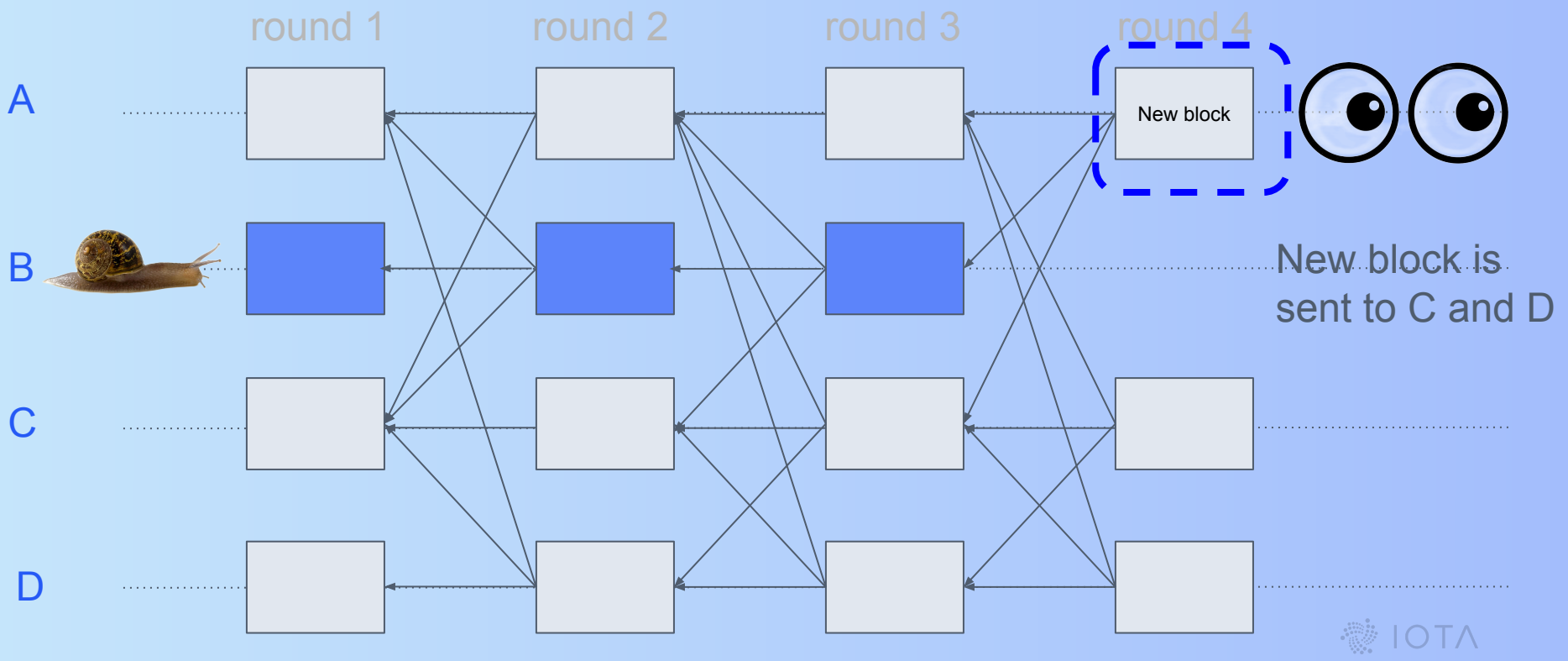
Pull-based dissemination issue



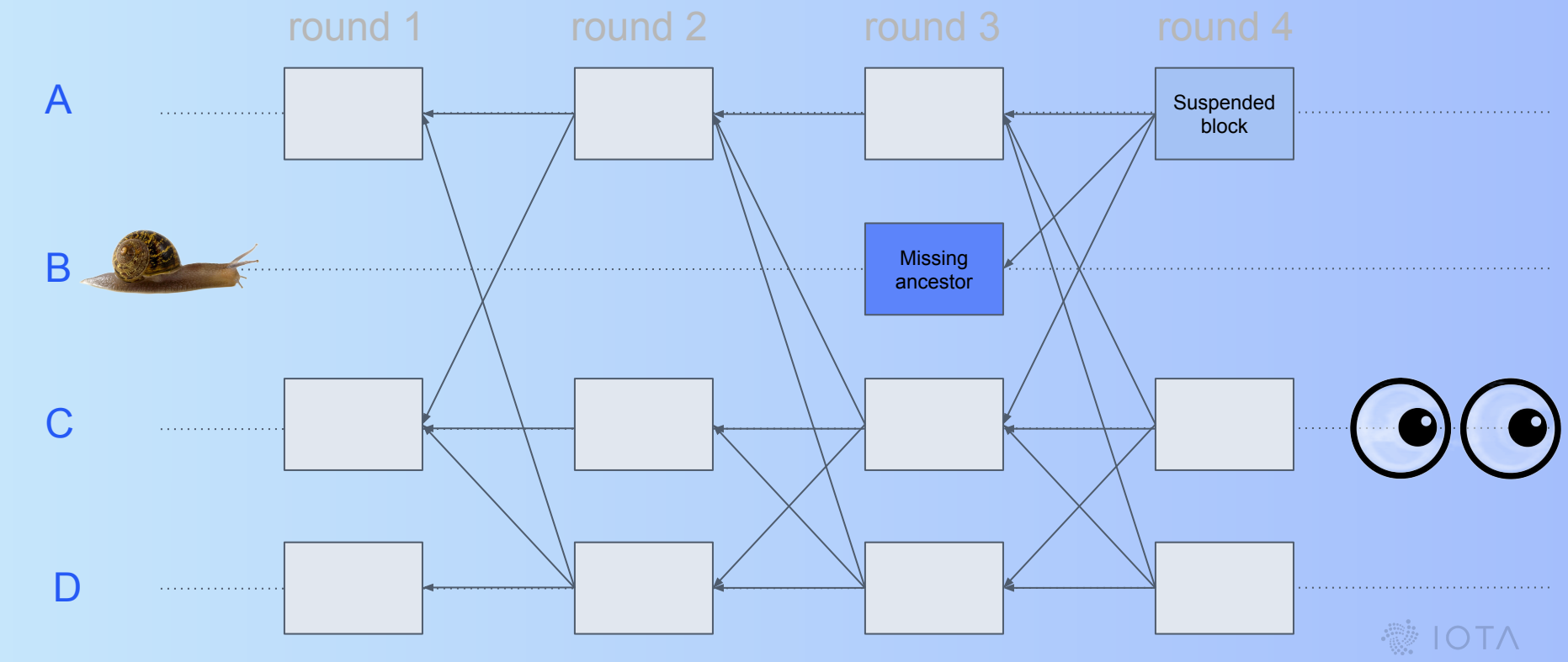
Pull-based dissemination issue



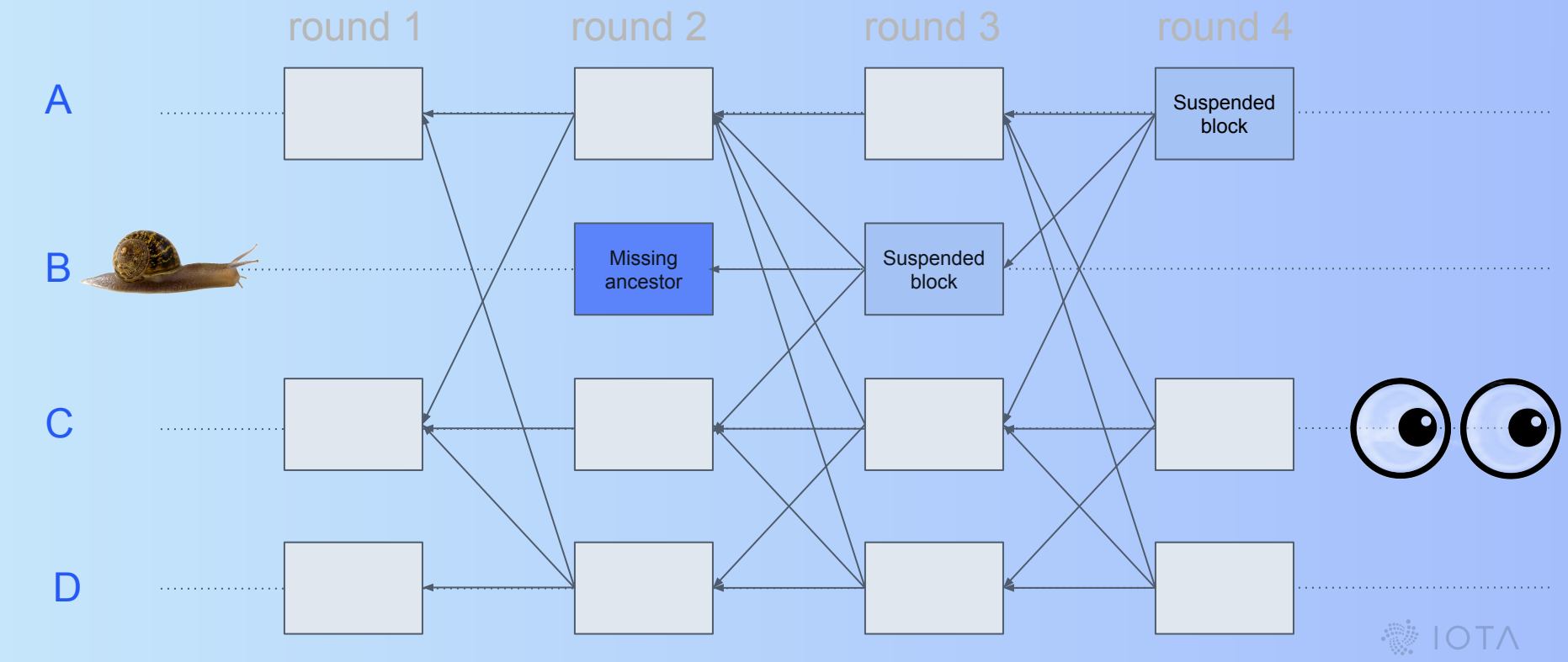
Pull-based dissemination issue



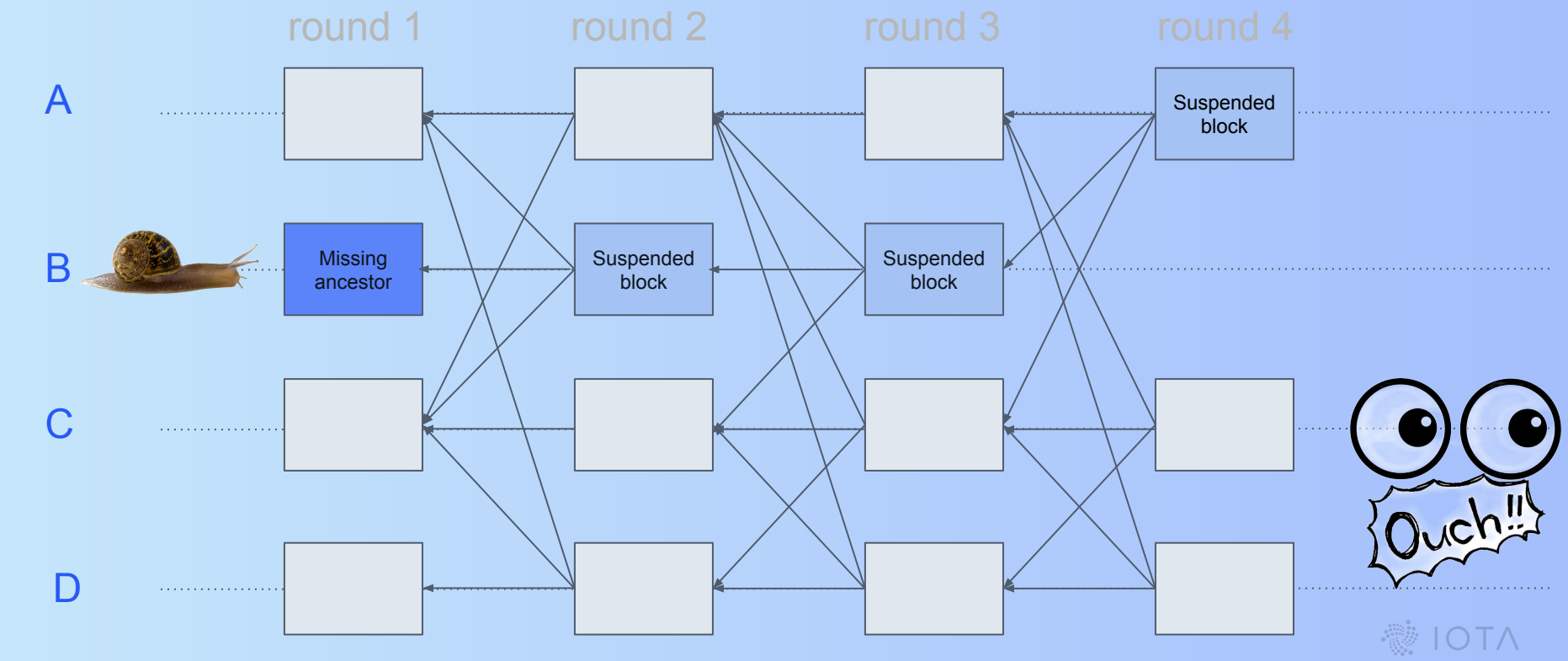
Pull-based dissemination issue



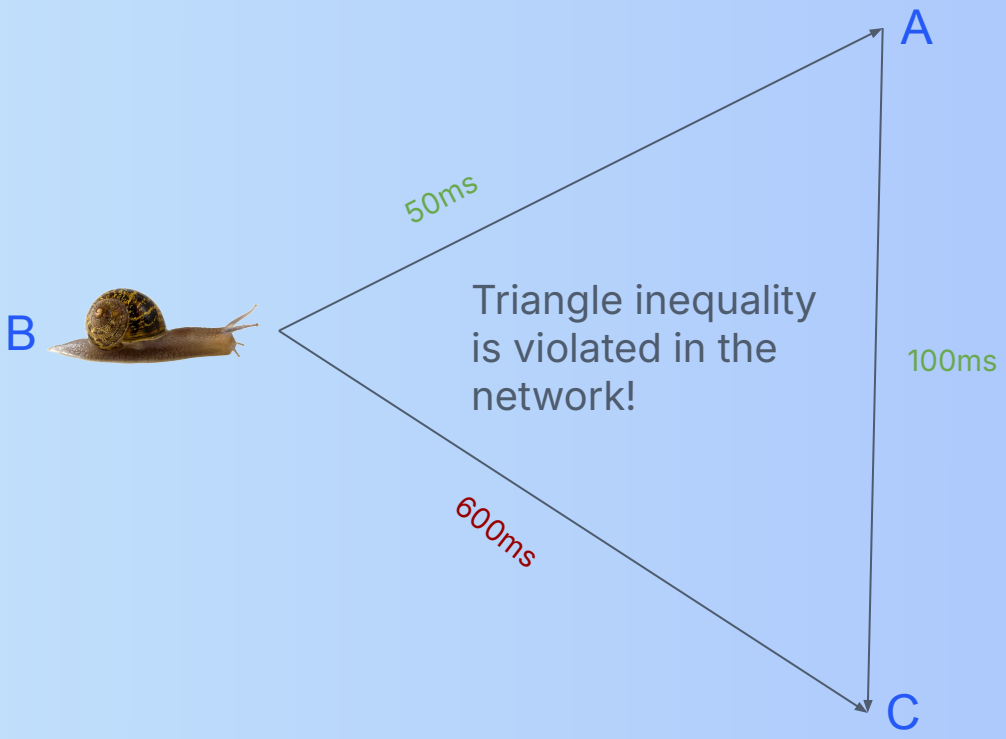
Pull-based dissemination issue



Pull-based dissemination issue



Pull-based dissemination issue



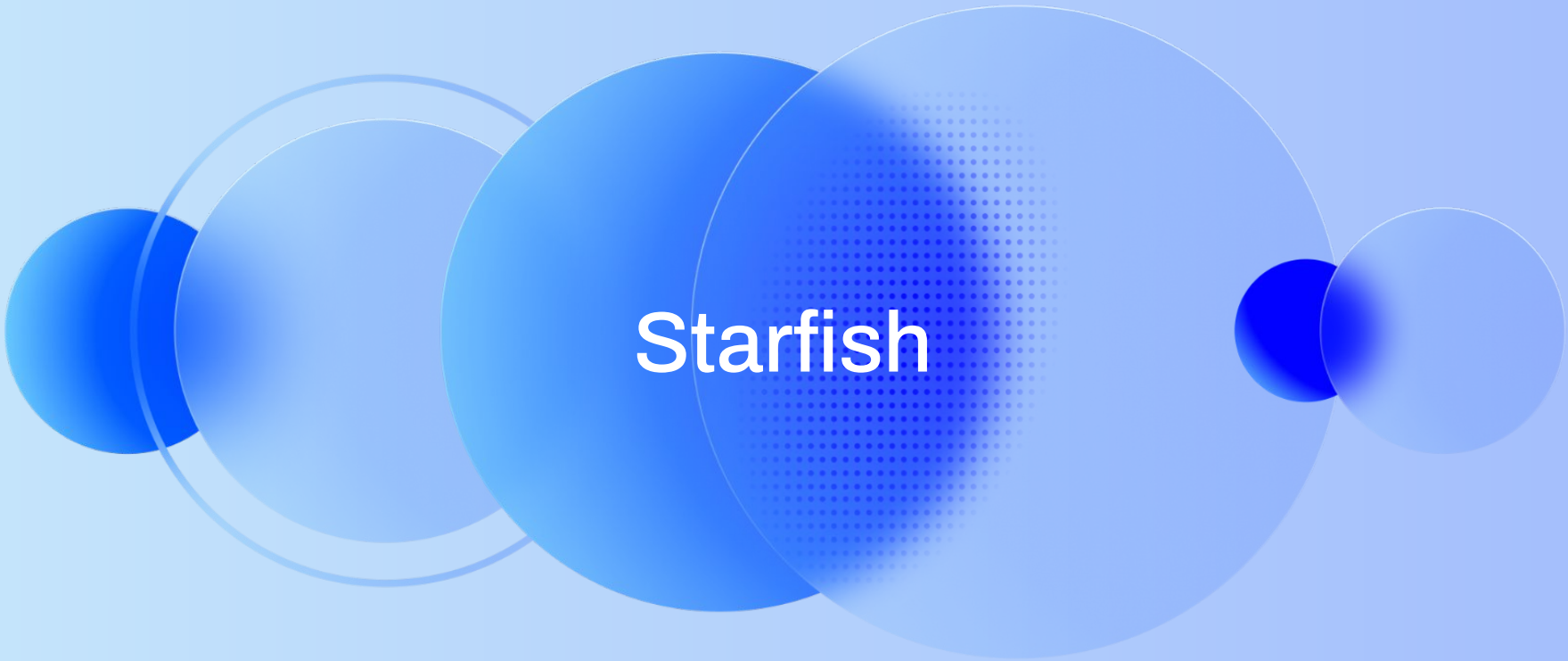
Motivation of Starfish

1. **Mitigate triangle inequality issue in network**

   Ensure that your blocks can be accepted in the local DAG of a peer upon their reception

2. **Keep bandwidth usage linear with tx data**

   Achieve linear communication complexity for transaction data even in Byzantine network


Starfish

Two ideas of Starfish

1. Decouple transaction data from block header metadata
    a. **Why:** pushing block headers is feasible because of the size
    b. **Purpose of headers:** driving DAG construction and consensus commits
    c. **When to sequence tx data:** at least 2f+1 nodes *acknowledged* that data is available

Block header:
- Round
- Author
- Ancestors
- Acknowledgements
- Tx data commitment
- Signature

Block:
- Block header
- Tx data

Two ideas of Starfish

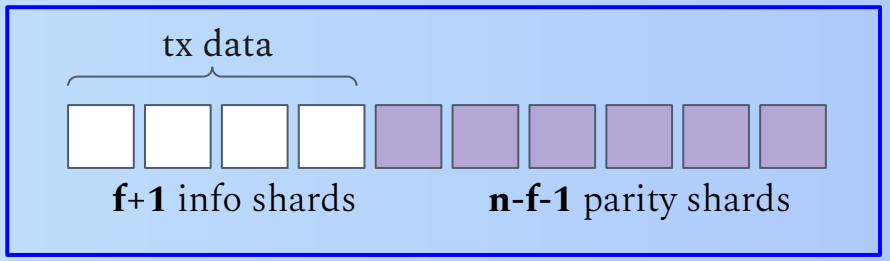
2. Encode transaction data with Reed-Solomon erasure codes

   a. **How to encode:** divide tx data into f+1 shards and encode with [n,f+1] RS codes

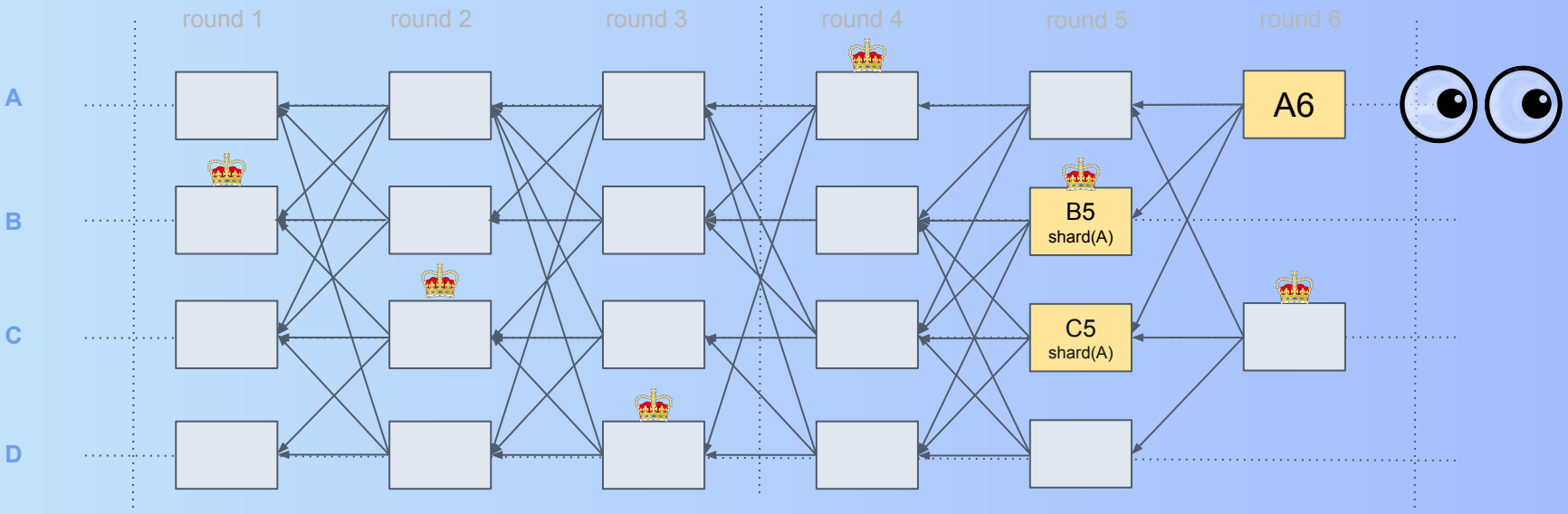
   b. **What to send (your data):** tx data + encoded tx commitment

   c. **What to send (data of other nodes):** one shard + shard proof

   d. **How to decode:** either received full data, or decode any f+1 shards



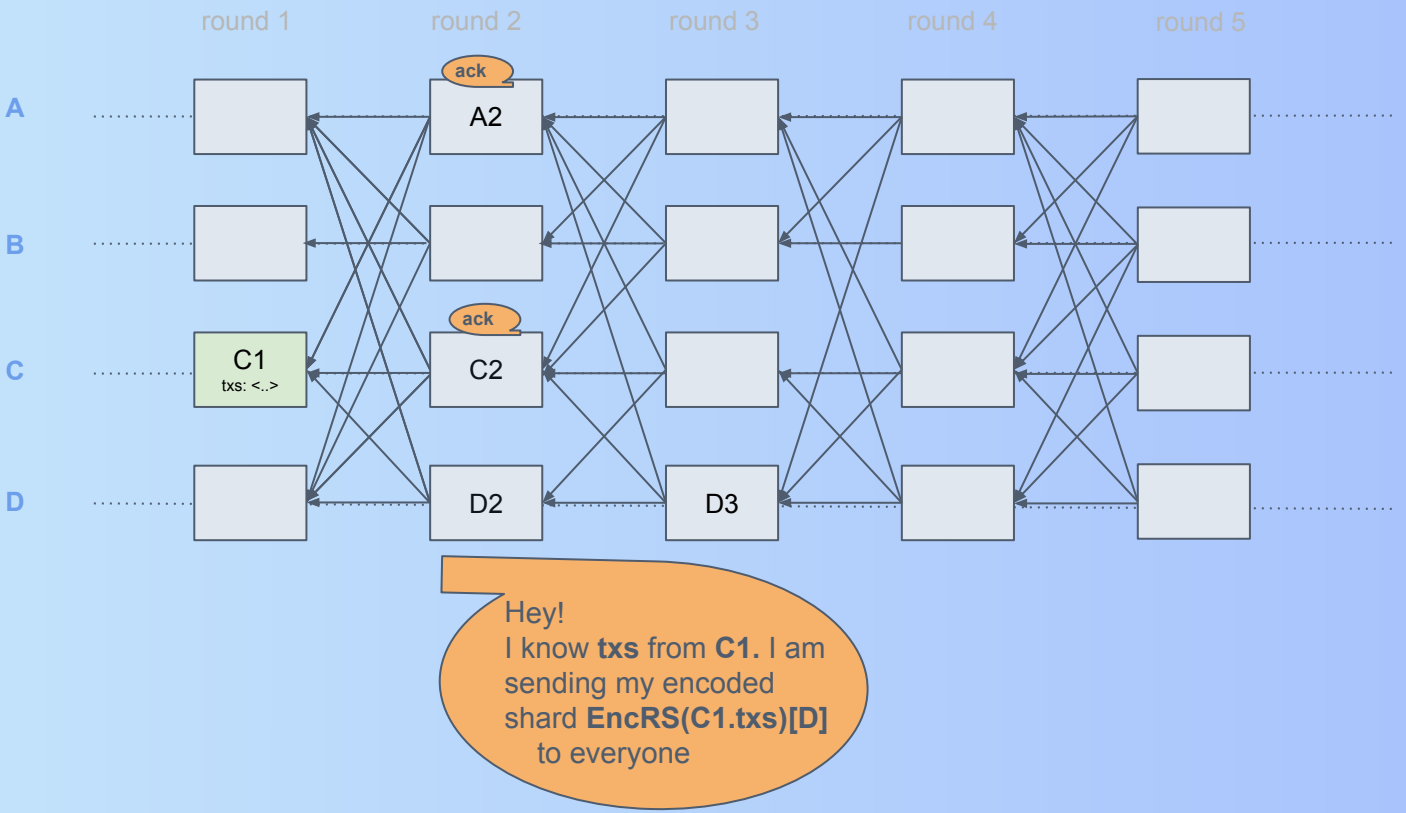
Broadcast rule in Starfish



``Send to others **block headers** (!) you know and think they need...

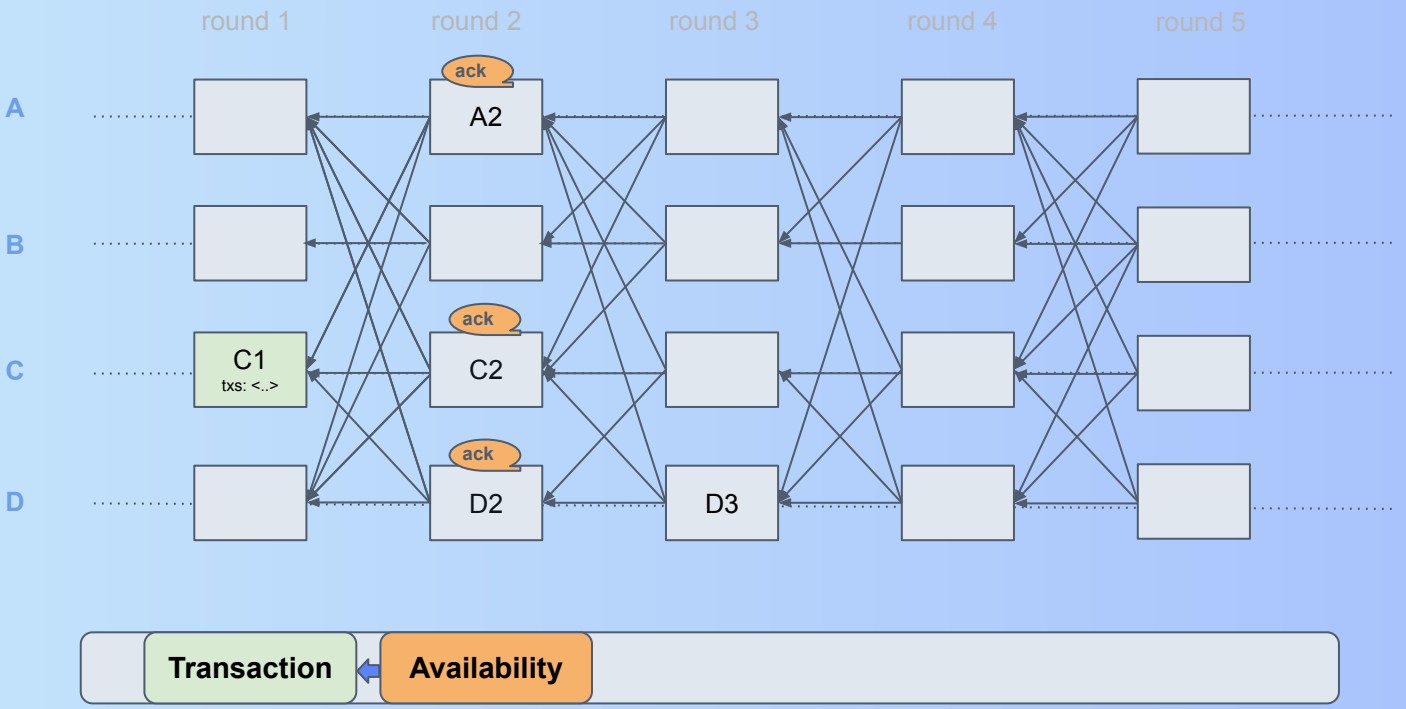
In addition, send your **encoded shard** when the transaction data is available"

Sequencing tx data in Starfish

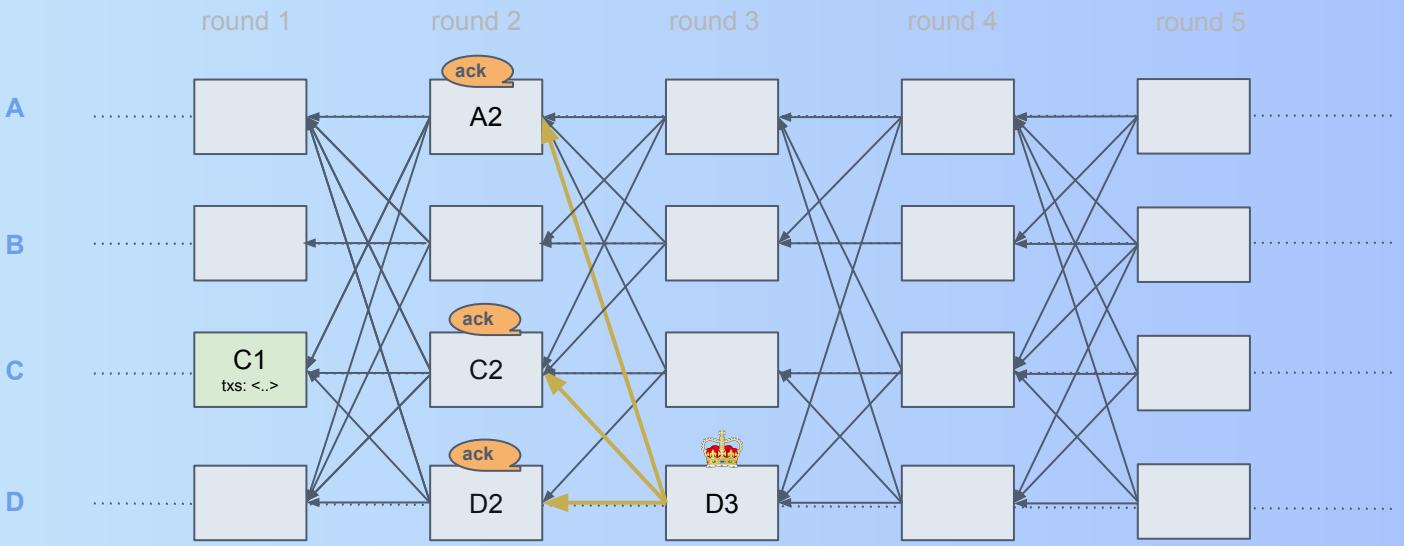


IOTA

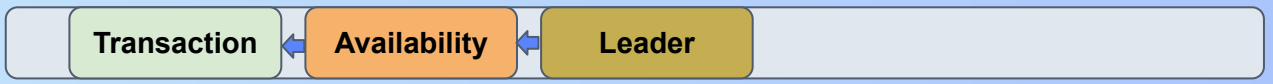
Sequencing tx data in Starfish



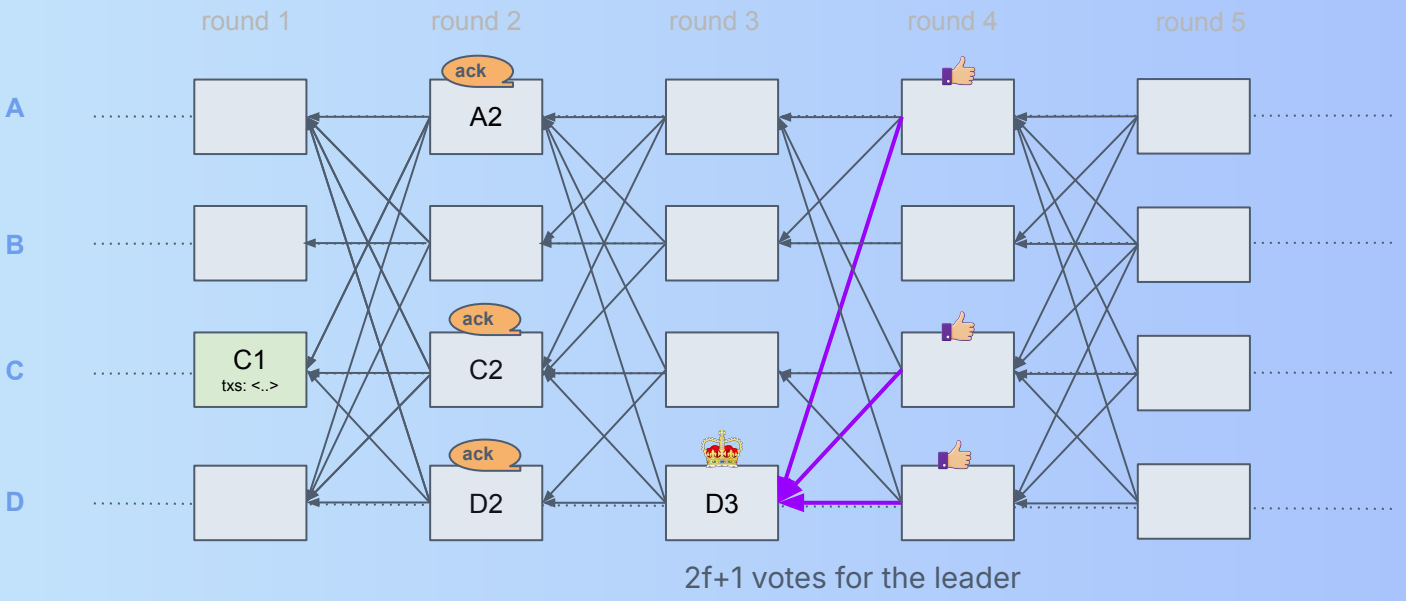
Sequencing tx data in Starfish



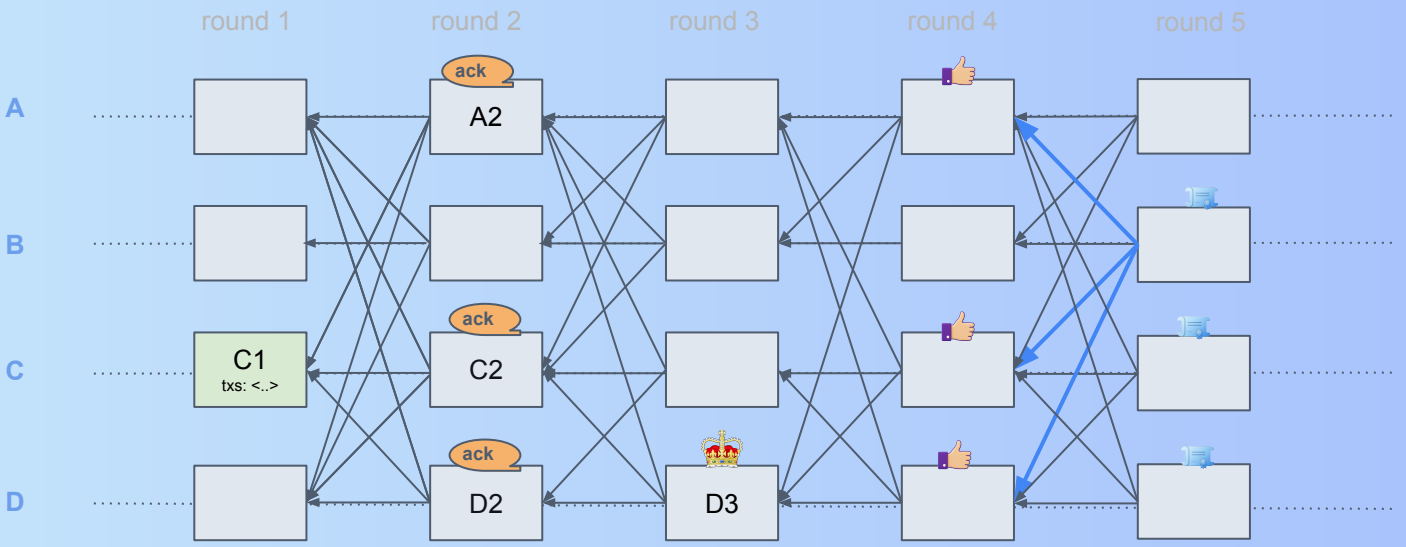
2f+1 acks are visible from the leader

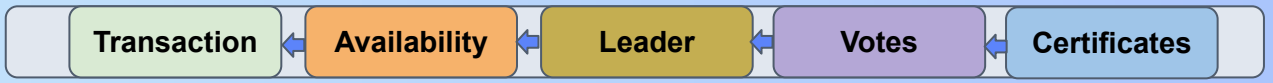


Sequencing tx data in Starfish



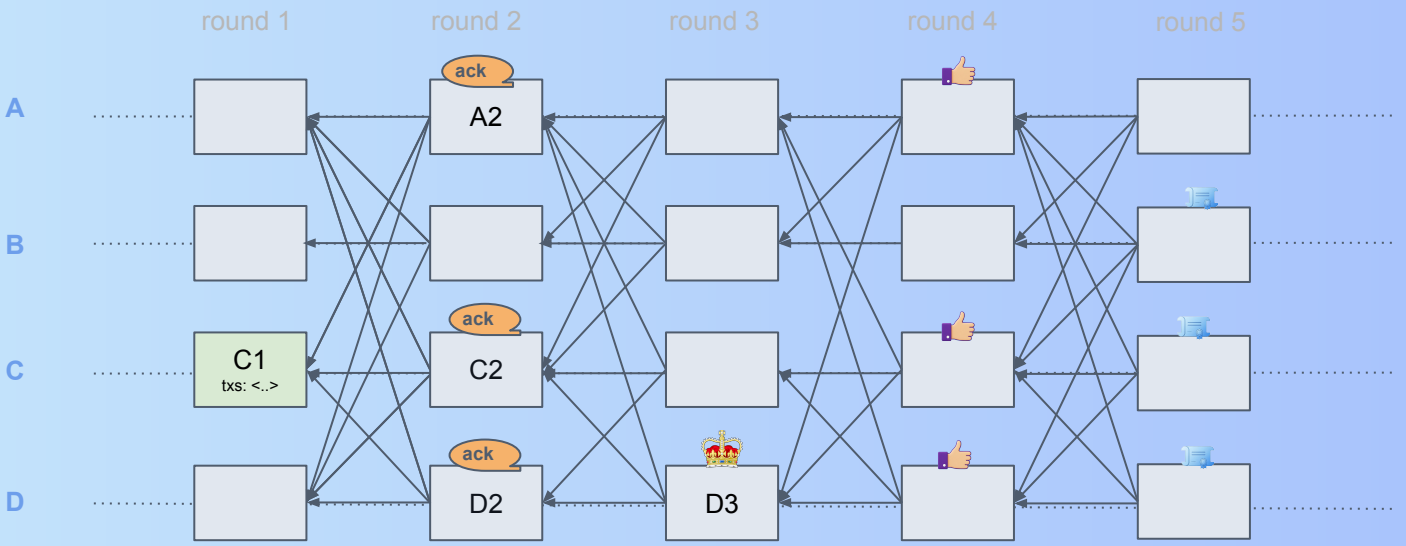
Sequencing tx data in Starfish

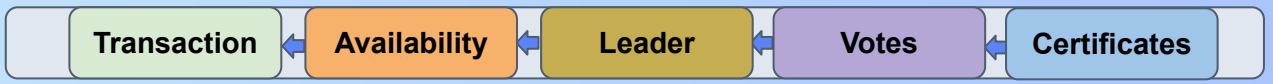




IOTA

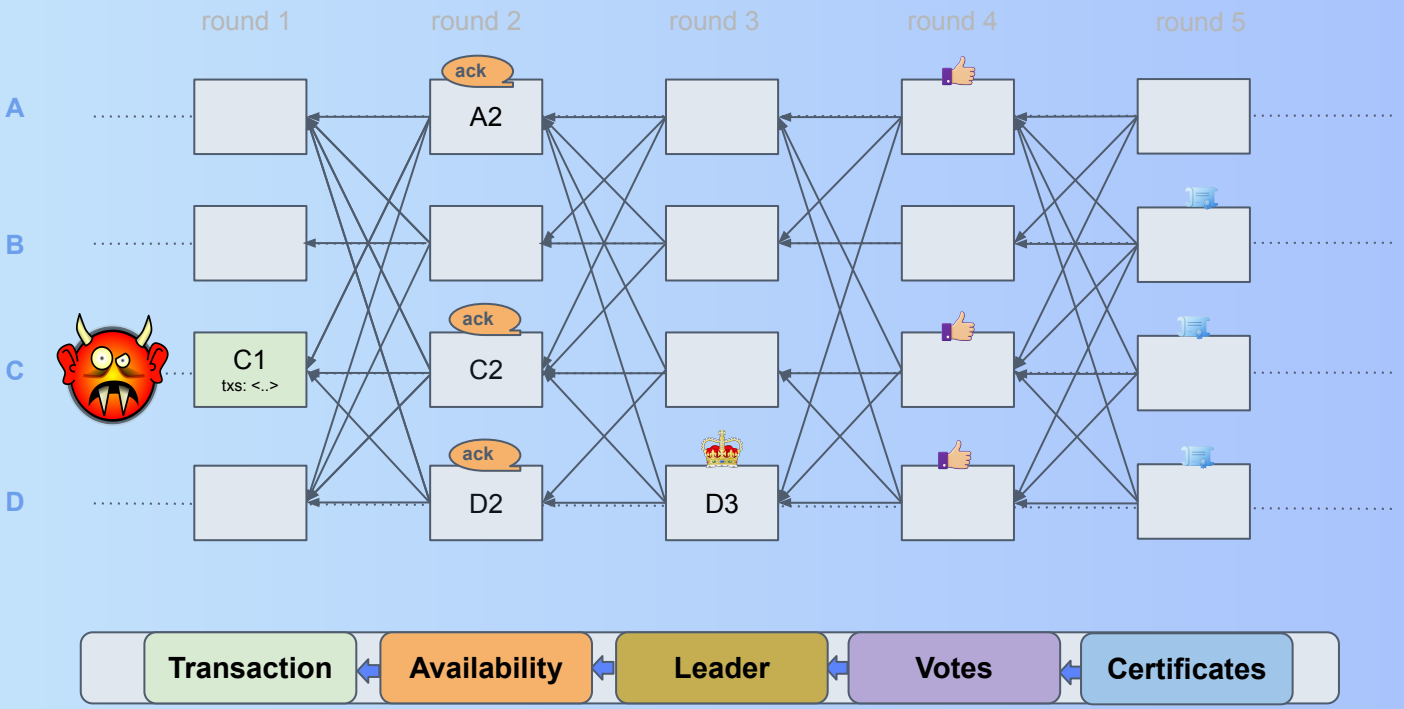
Sequencing tx data in Starfish





IOTA

Recoverability of committed transaction data



B didn't get from C transactions in **C1**...

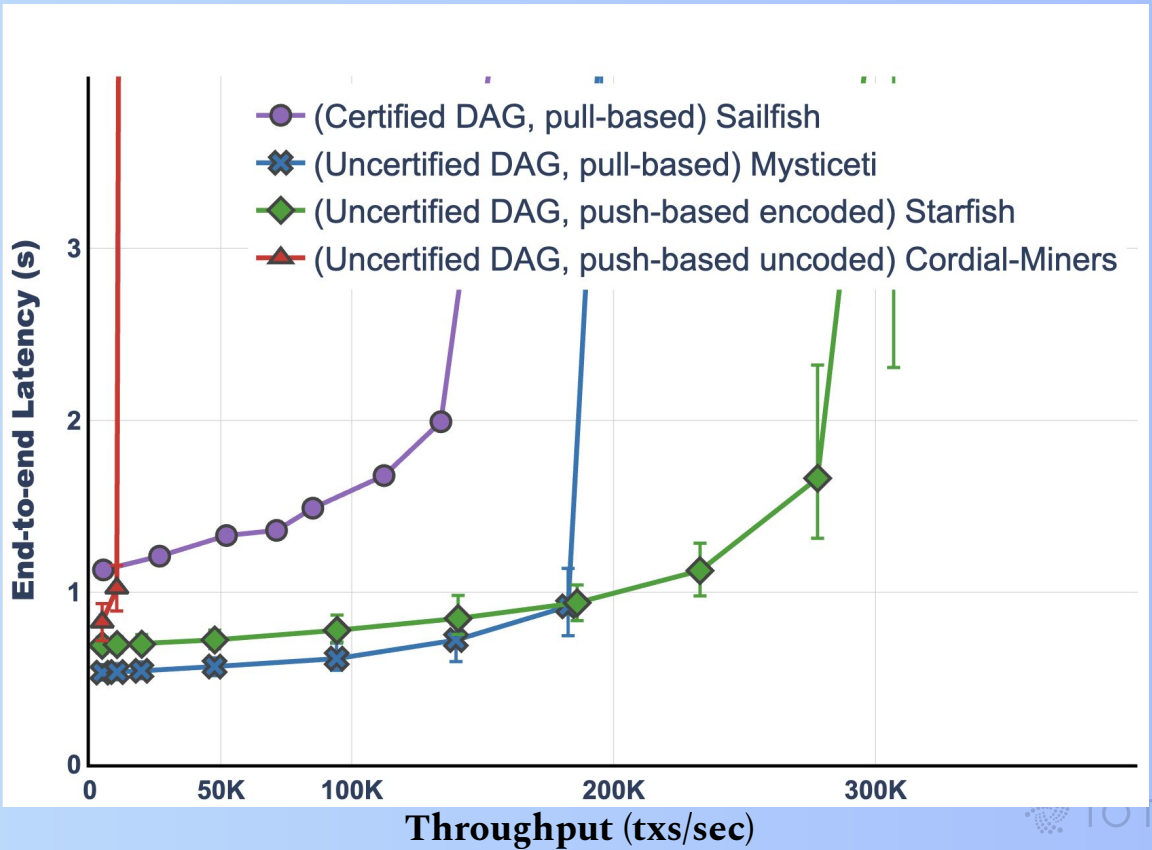
But recover from shards sent by A and D

IOTA

Performance and comparison

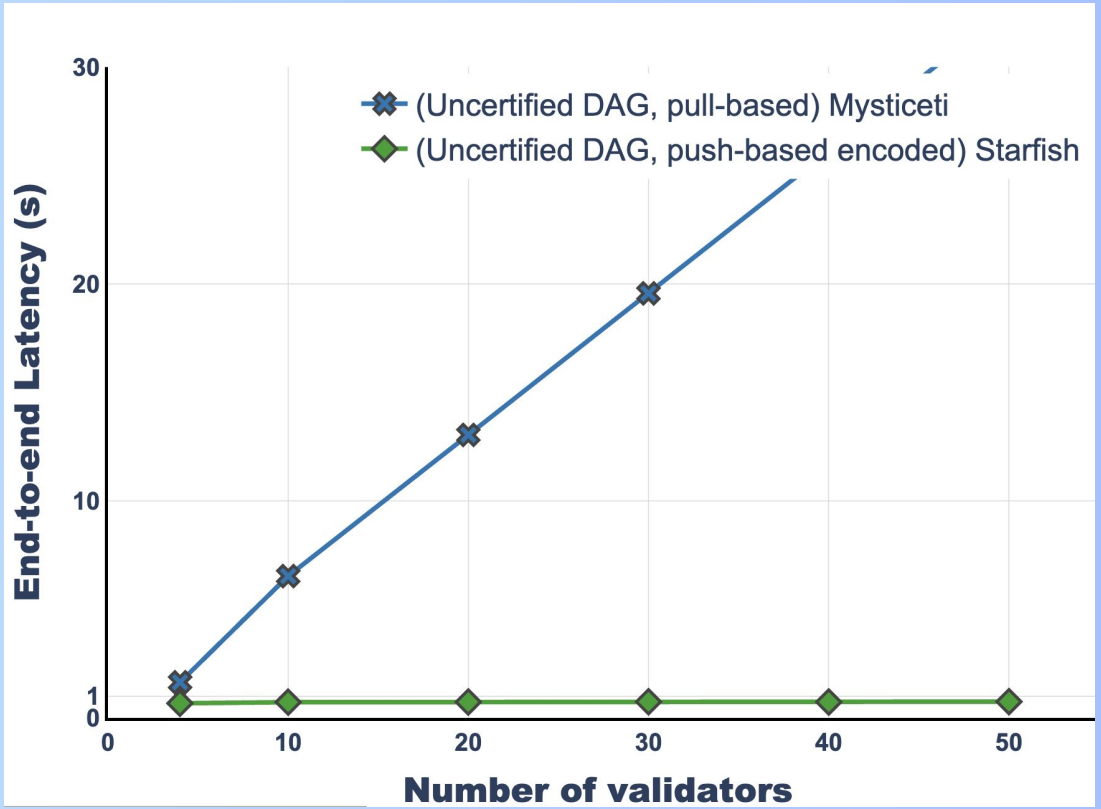
Baseline (no Byzantine validators)

- 100 validators
- geo-distributed network over 10 regions
- 512B-sized transactions
- Amazon EC2 m5d.4xlarge machines

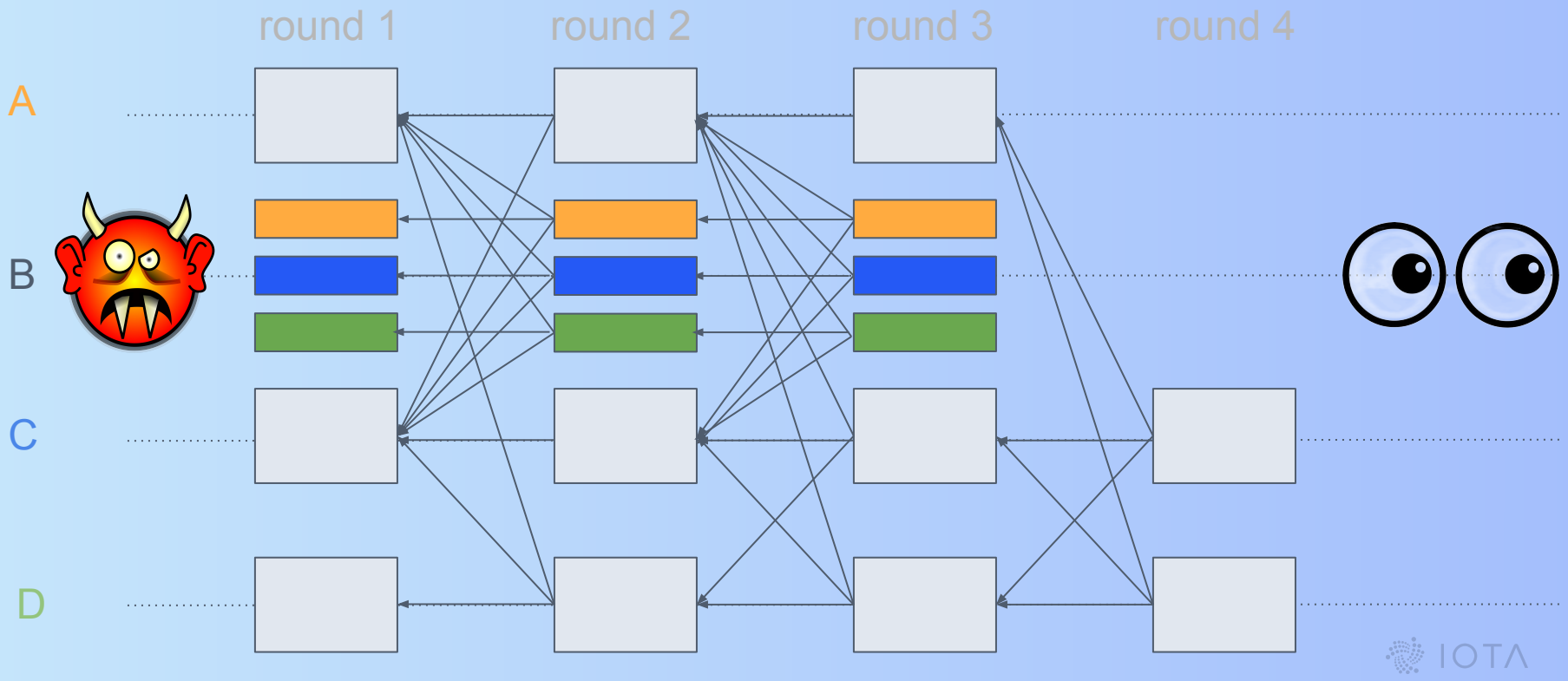


Equivocating chains bomb (only 1 Byzantine validator)

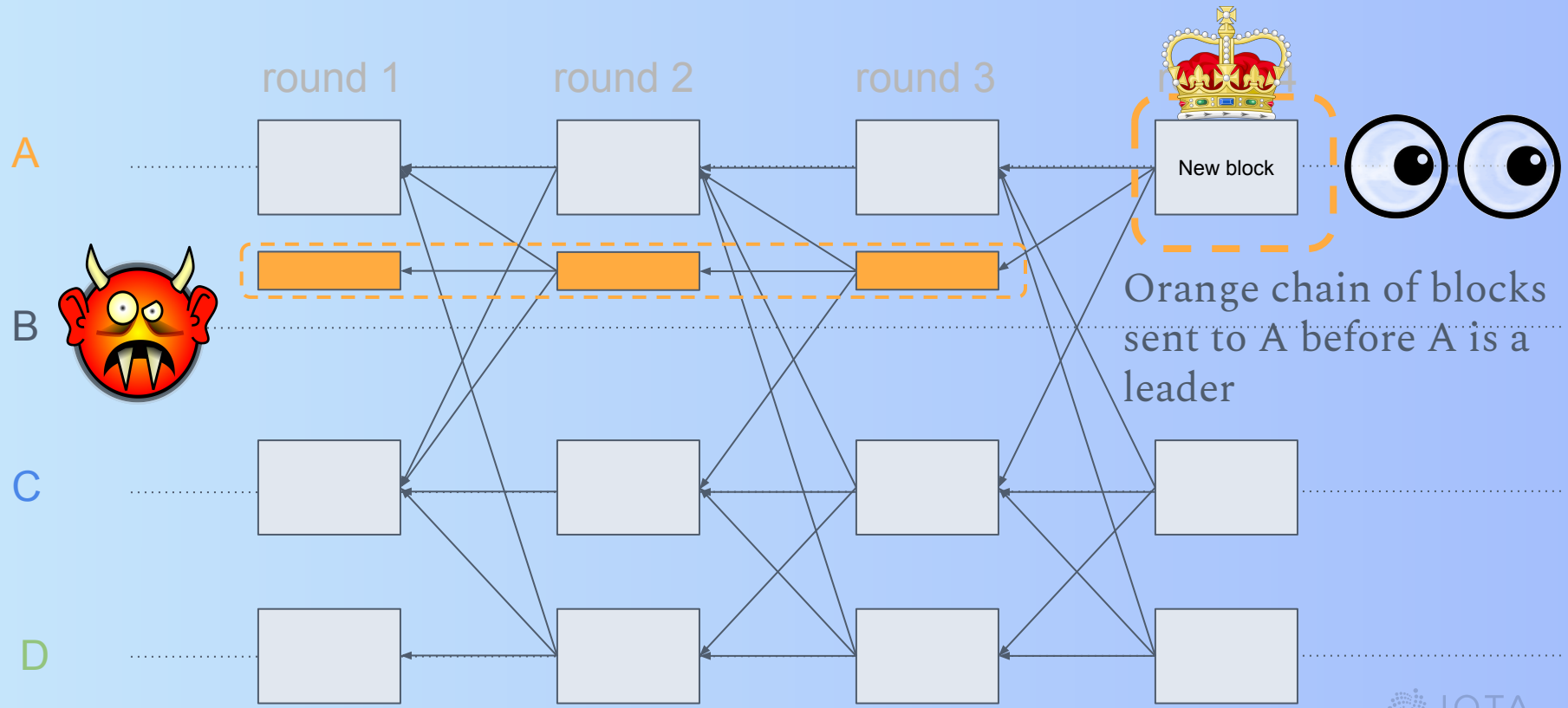
- 10-50 validators
- geo-distributed network over 10 regions
- 512B-sized transactions
- Target load 10,000 txs / sec
- Amazon EC2 m5d.4xlarge machines



Equivocating chains bomb attack

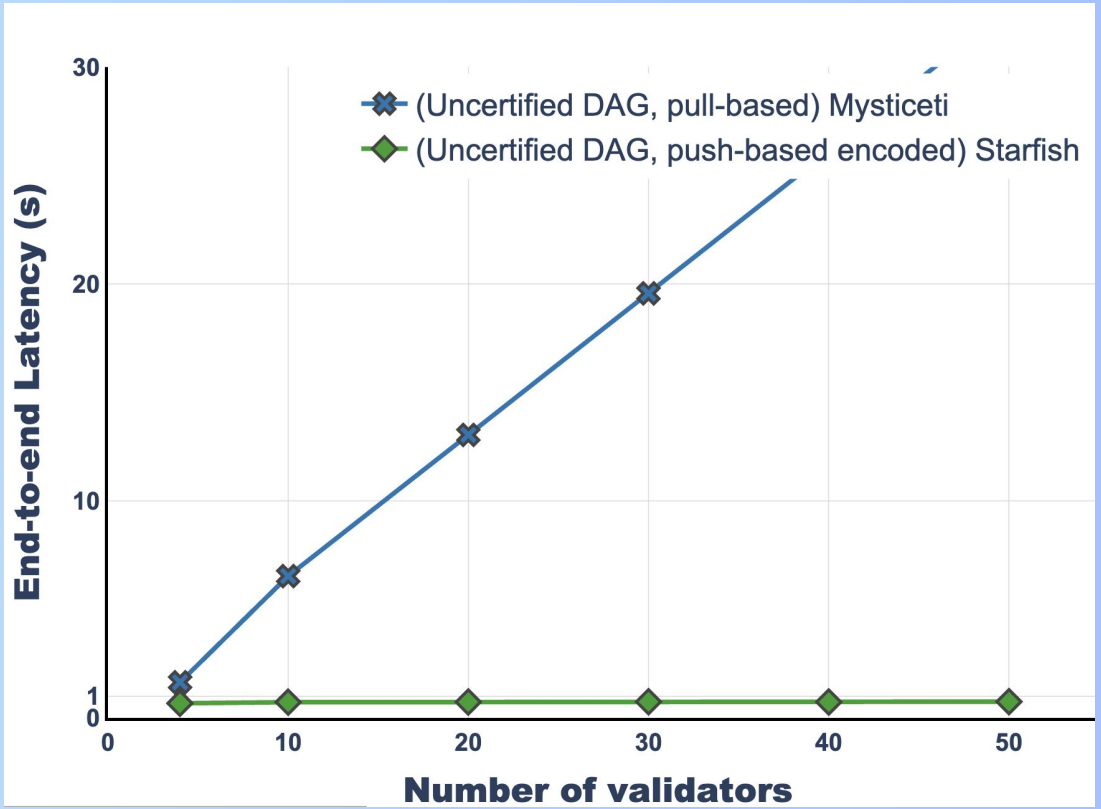


Equivocating chains bomb attack



Equivocating chains bomb (only 1 Byzantine validator)

- 10-50 validators
- geo-distributed network over 10 regions
- 512B-sized transactions
- Target load 10,000 txs / sec
- Amazon EC2 m5d.4xlarge machines



Comparison

	Byzantine broadcast primitive	Avg. block latency	Avg. e2e latency	One failure leader	Amort. comm. complexity
Bullshark	RBC for all blocks	14δ	$+2\delta$	$+(8\Delta + 8\delta)$	$O(n)$
Shoal	RBC for all blocks	12δ	$+2\delta$	$+(8\Delta + 4\delta)$	$O(n)$
Shoal++	RBC for all blocks	5δ	$+0.5\delta$	$+(8\Delta + 4\delta)$	$O(n^2)$
Sailfish 1	RBC for all blocks	9δ	$+2\delta$	$+(8\Delta + 2\delta)$	$O(n)$
Sailfish 2	RBC for all blocks	5δ	$+\delta$	$+(4\Delta + 2\delta)$	$O(n^2)$
BBCA-chain	BBCA for leaders	4δ	$+1.5\delta$	$+4\Delta$	$O(n^2)$
Cordial Miners	None	5δ	$+0.5\delta$	$+6\Delta$	$O(n^2)$
Mysticeti	None	4δ	$+0.5\delta$	$+4\Delta$	$O(n^2)$
Starfish	**None**	5δ	$+0.5\delta$	$+2\Delta$	$O(n)$

Summary

Starfish ...

- is a **high-performance** partially synchronous BFT protocol
- is built on **uncertified DAG**
- relies on **push-based** block dissemination strategy with **encoding**
- achieves **lowest latency** in the class of DAG-based protocols with **linear amortized communication complexity**
- has significantly better **robustness** against **Byzantine attacks** than existing uncertified DAG protocols

More info:

Thank you!

Questions?