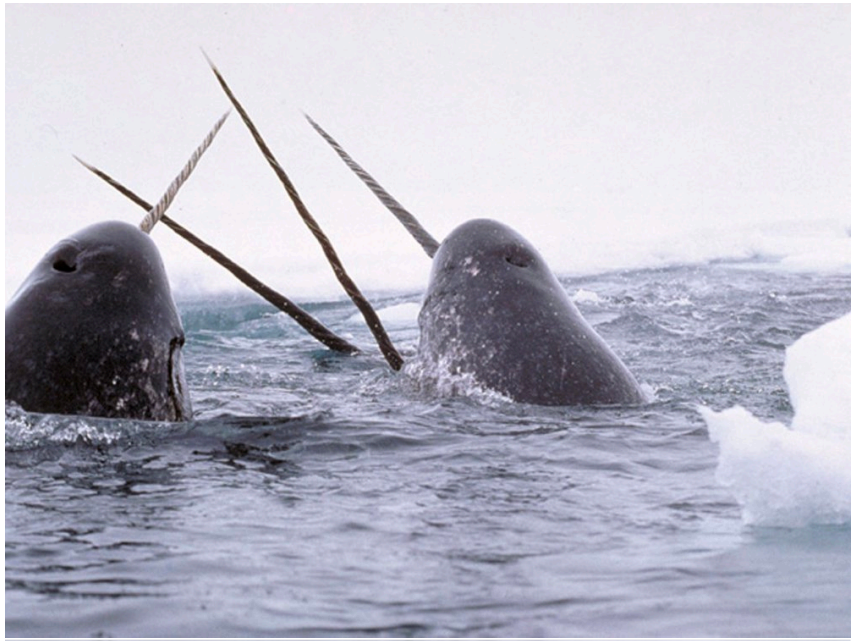
Narwhal and Tusk:
A DAG-based Mempool and Efficient BFT Consensus



Presented by:
Aditya Kumar Bej, Madhumitha Santhanakrishnan, Shreya Gundu and Shristi Suman

The slides are closely adapted from Alberto Sonnino's slides

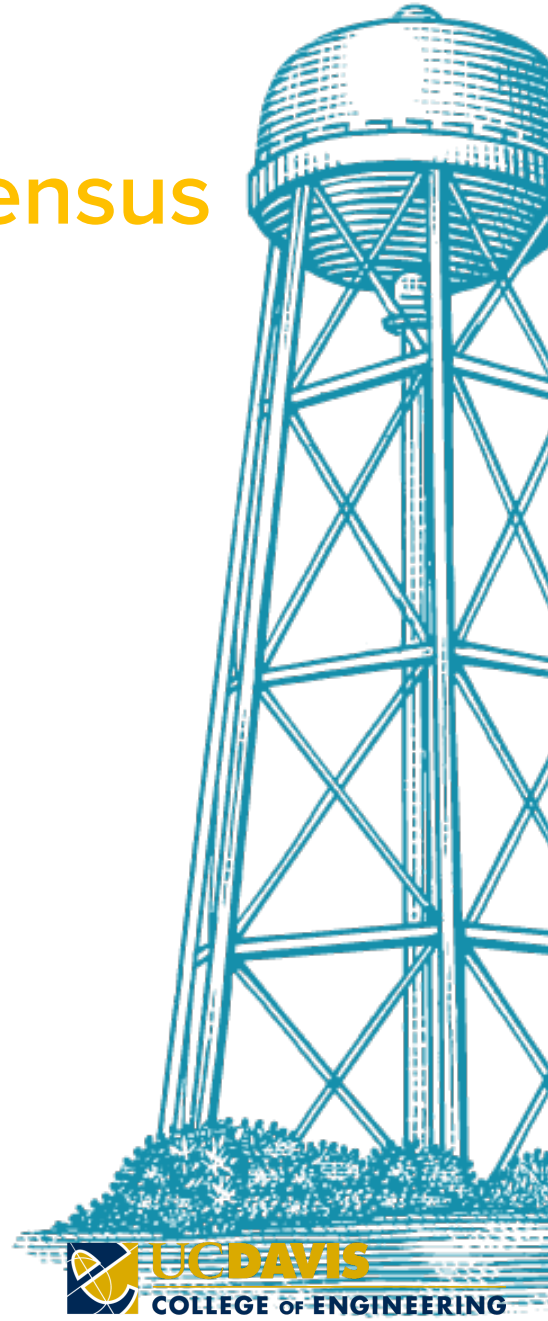


Table Of Contents

Goal of the Paper

To build (really) high performance blockchains

Goal of the Paper

To build (really) high performance blockchains

But how?

INTRODUCING NARWHAL AND TUSK

Narwhal

A Directed Acyclic Graph (DAG) based Mempool protocol

Tusk

Zero-message overhead asynchronous consensus protocol

NARWHAL AND TUSK

Narwhal

A Directed Acyclic Graph (DAG) based Mempool protocol

Questions

What is DAG?

Tusk

Zero-message overhead asynchronous consensus protocol

NARWHAL AND TUSK

Narwhal

A Directed Acyclic Graph (DAG) based Mempool protocol

Tusk

Zero-message overhead asynchronous consensus protocol

Questions

What is DAG?

What is Mempool?

NARWHAL AND TUSK

Narwhal

A Directed Acyclic Graph (DAG) based Mempool protocol

Tusk

Zero-message overhead asynchronous consensus protocol

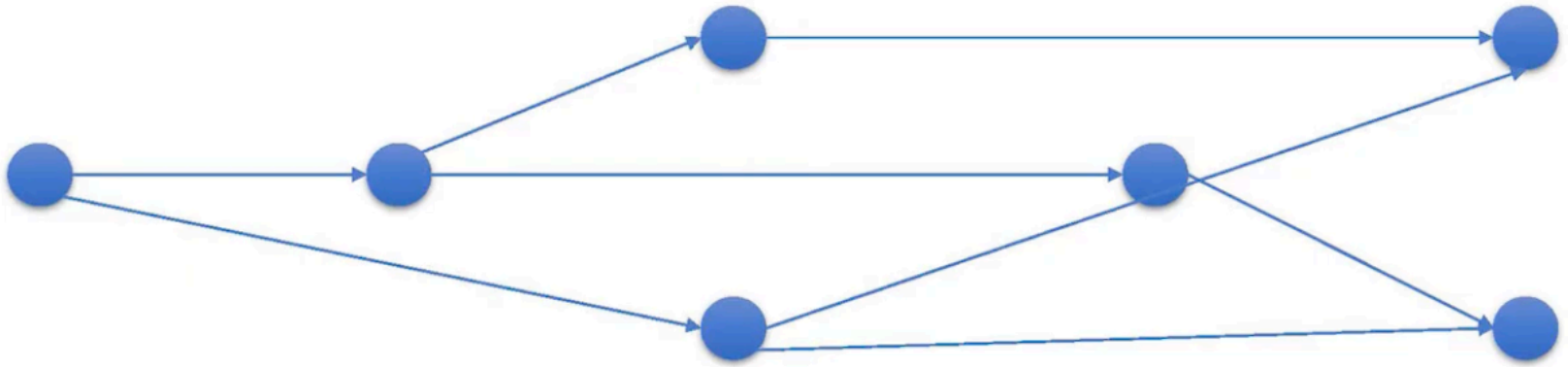
Questions

What is DAG?

What is Mempool?

What is zero-message overhead?

DAG - Directed Acyclic Graph



Graph - Node connected to other nodes
Directed - Connection between nodes have direction (A -> B != B -> A)
Acyclic - Non-circular. No closed loop in the connections.

Mempool 101

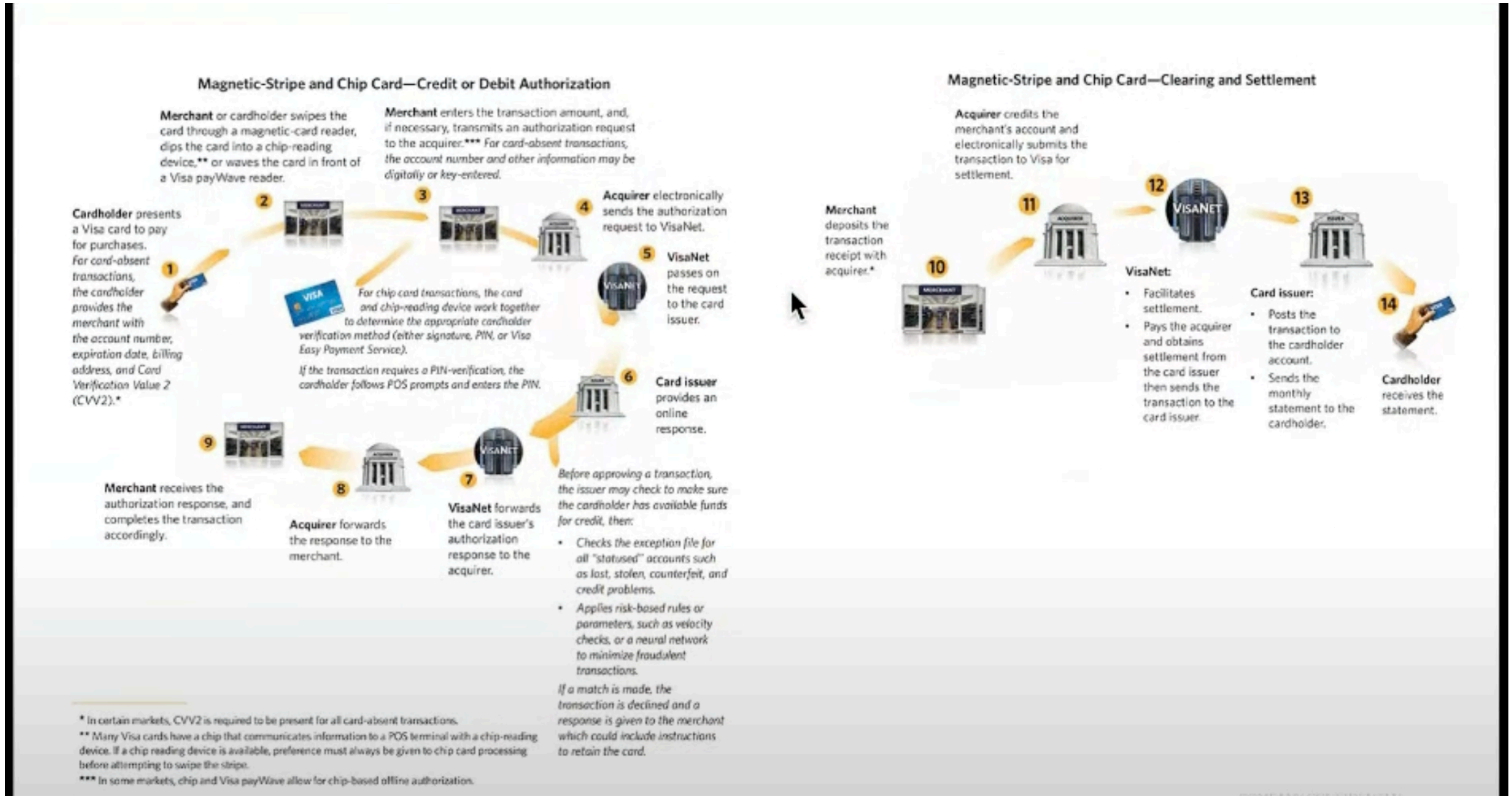
Fundamental Understanding

Transactions =
Complexity

What happens when you swipe and wait?

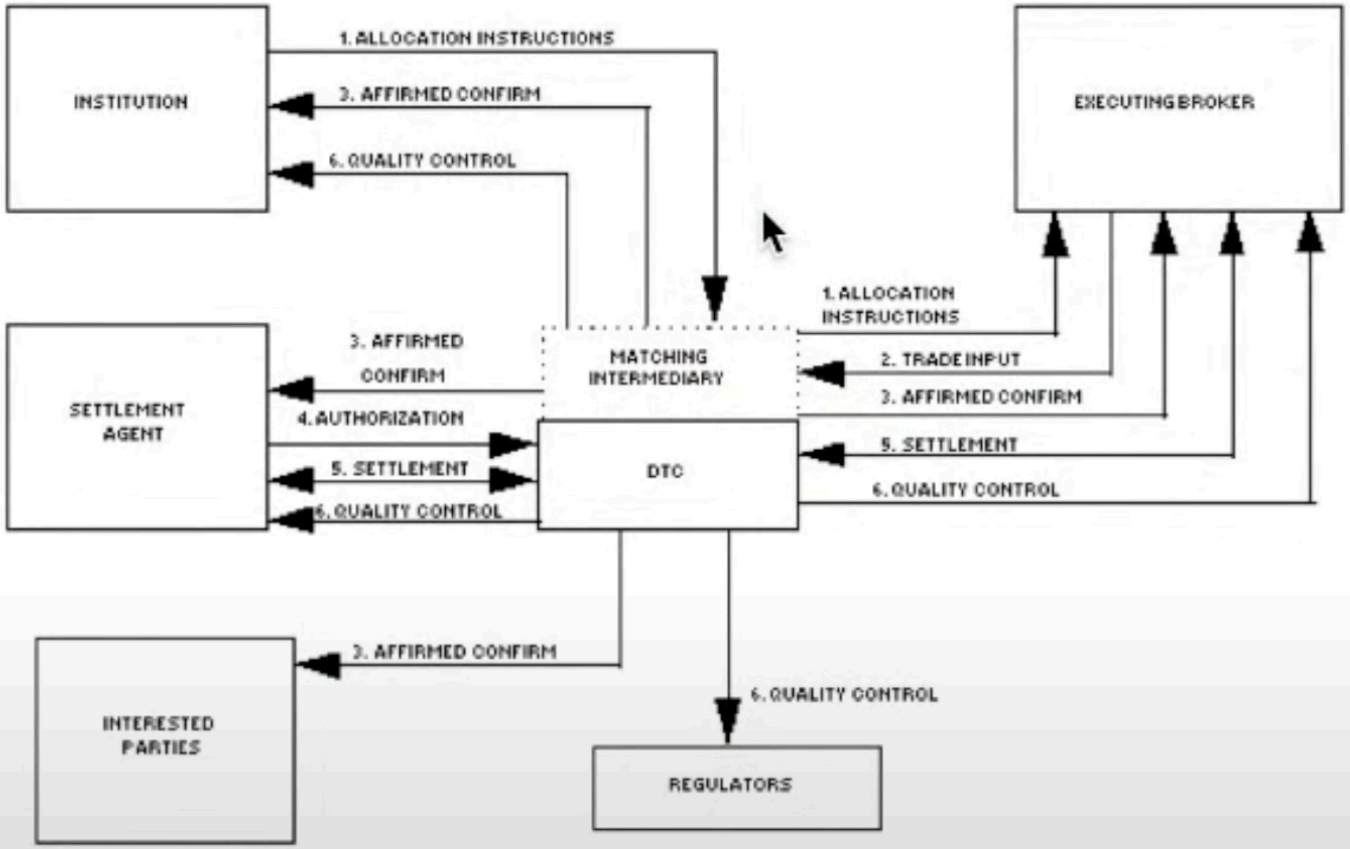
UCDAVIS
COLLEGE OF ENGINEERING

This happens

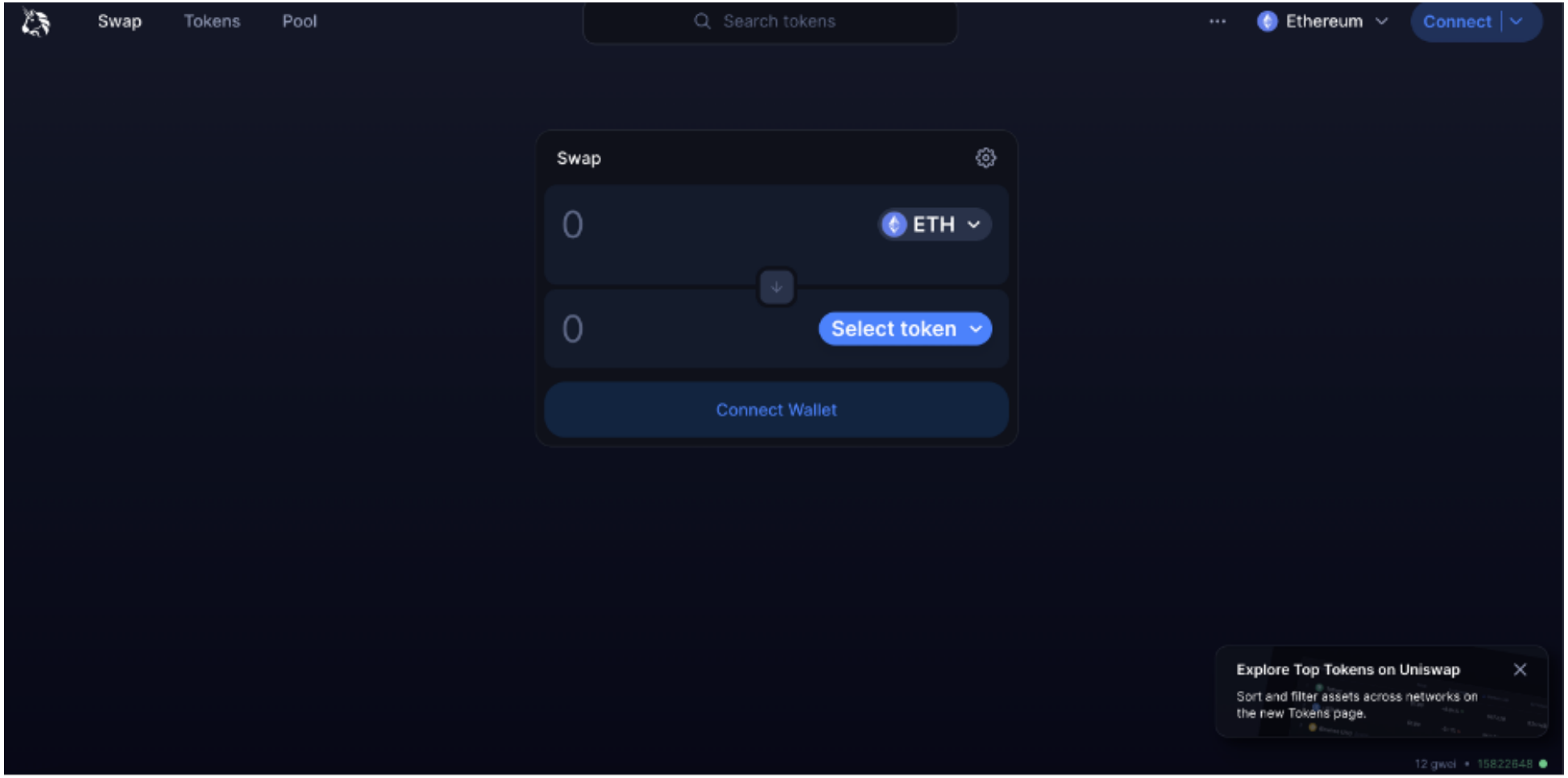


UC DAVIS COLLEGE OF ENGINEERING

Same is true for a stock market application

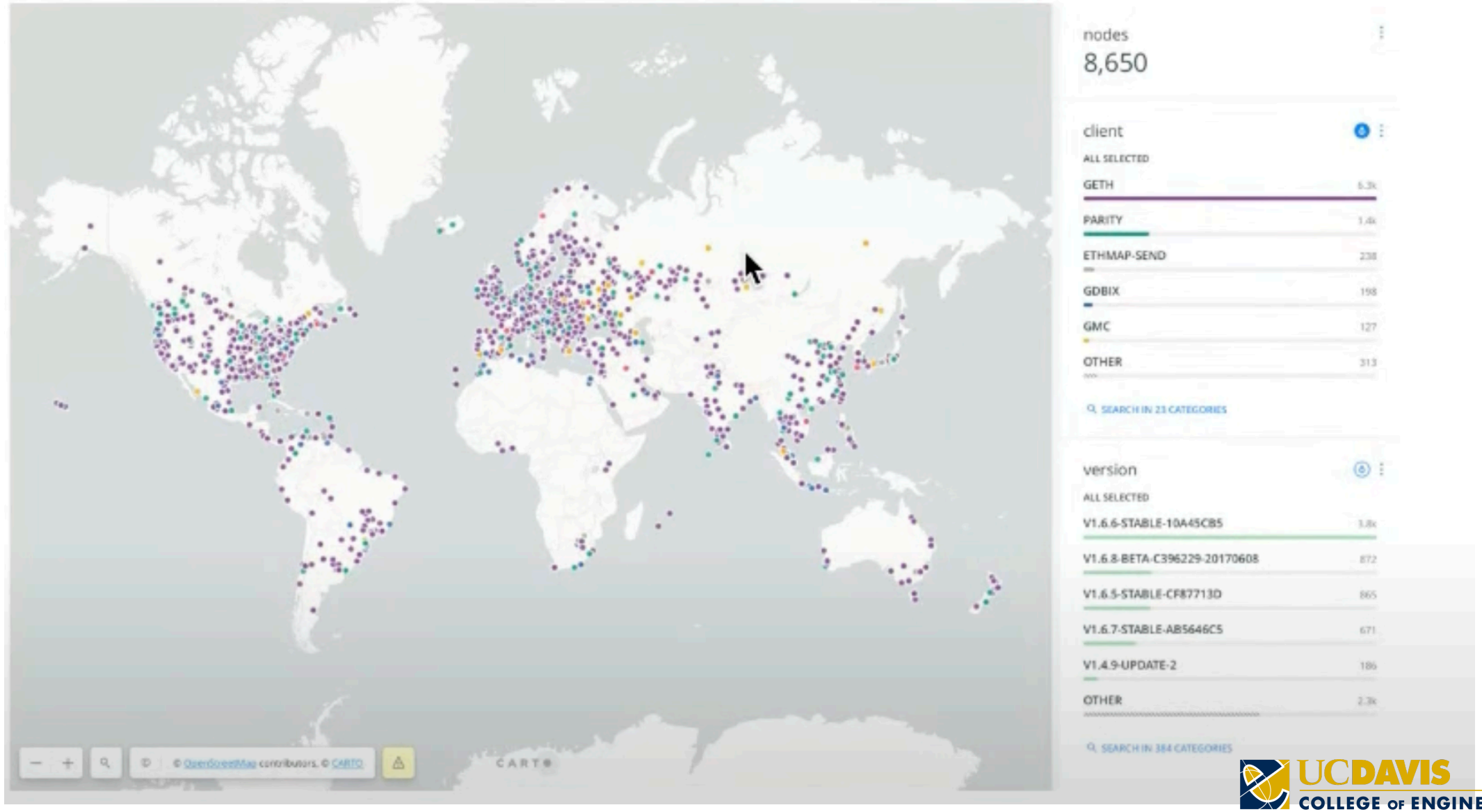


Onto a Crypto Trading application



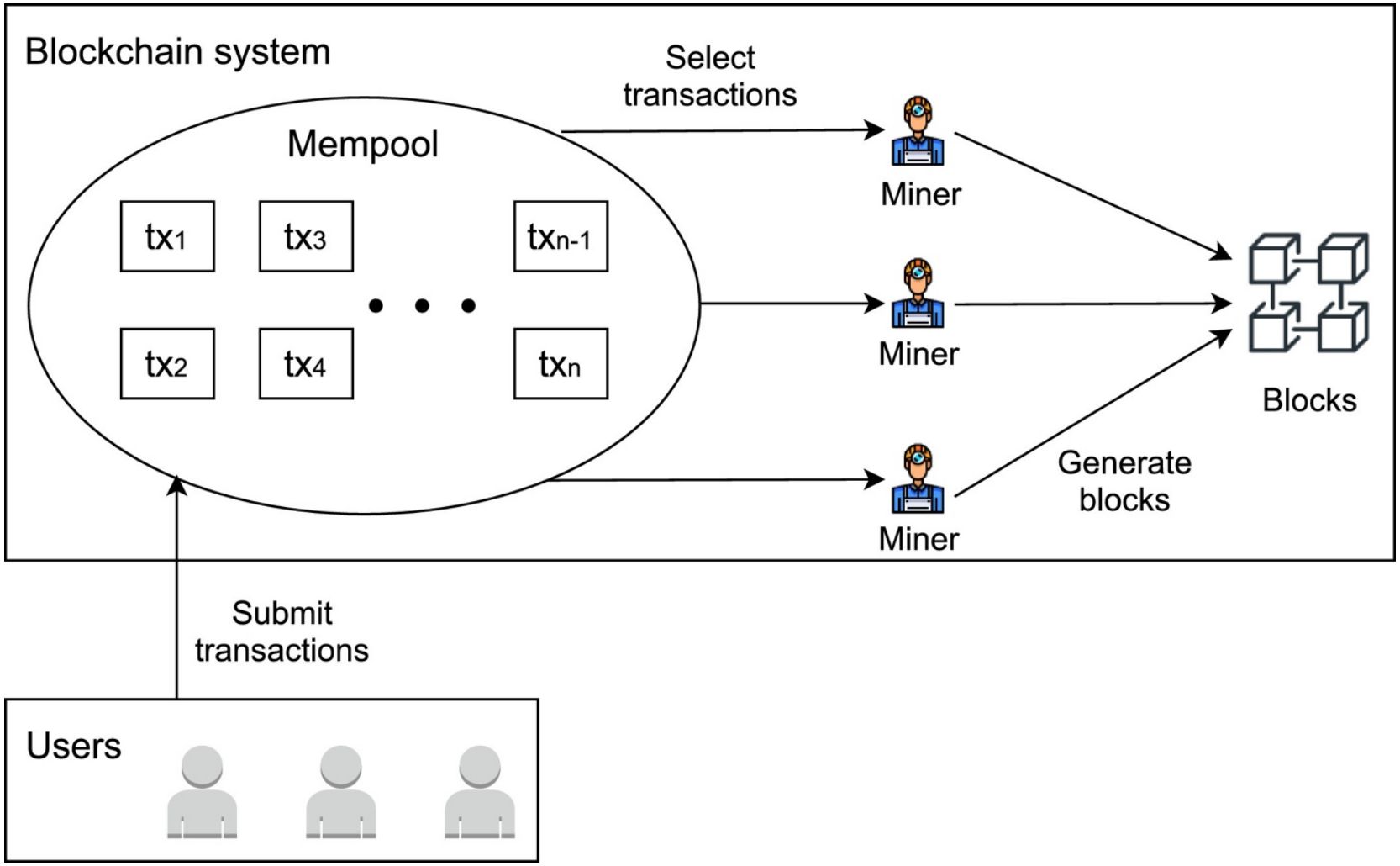
UC DAVIS COLLEGE OF ENGINEERING

Ethereum Network



UC DAVIS COLLEGE OF ENGINEERING

Mempool



1
Mempool = PRE-CONSENSUS

2

There is **no single** mempool

3

Each mempool has its **own unique set of transactions**

4

Each mempool has its **own unique characteristics** based on client type, configuration, peers

5

The mempool is mutable due to Replacement Transactions, including Speed ups & Cancels

6

The mempool is an **inconsistent** ‘**fabric**’ that grows lumpy during congestion/pressure

7

Your transaction competes in the mempool for block space

Traditional Transactions

Like a ballroom dance

- Very orderly
- Plays by the rules

Blockchain Transactions: Expectations

Like a flash mob

- More spontaneous
- Organic

UCDAVIS COLLEGE OF ENGINEERING

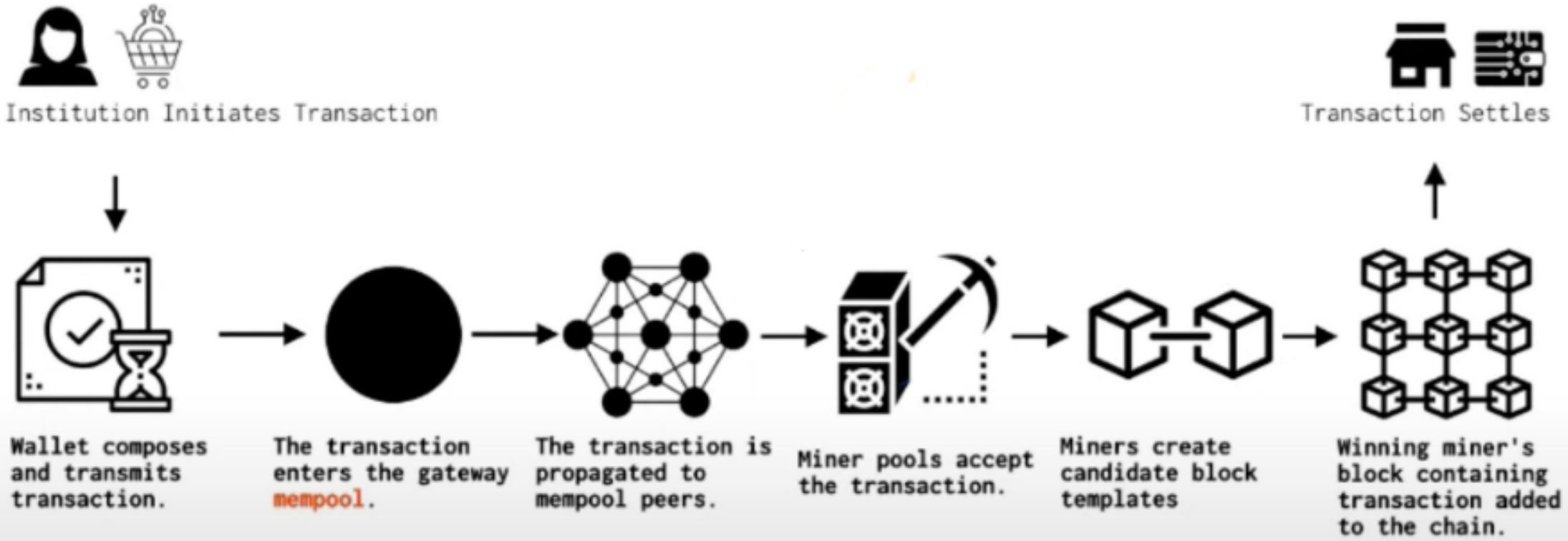
Blockchain Transactions: Reality

Like a mosh pit

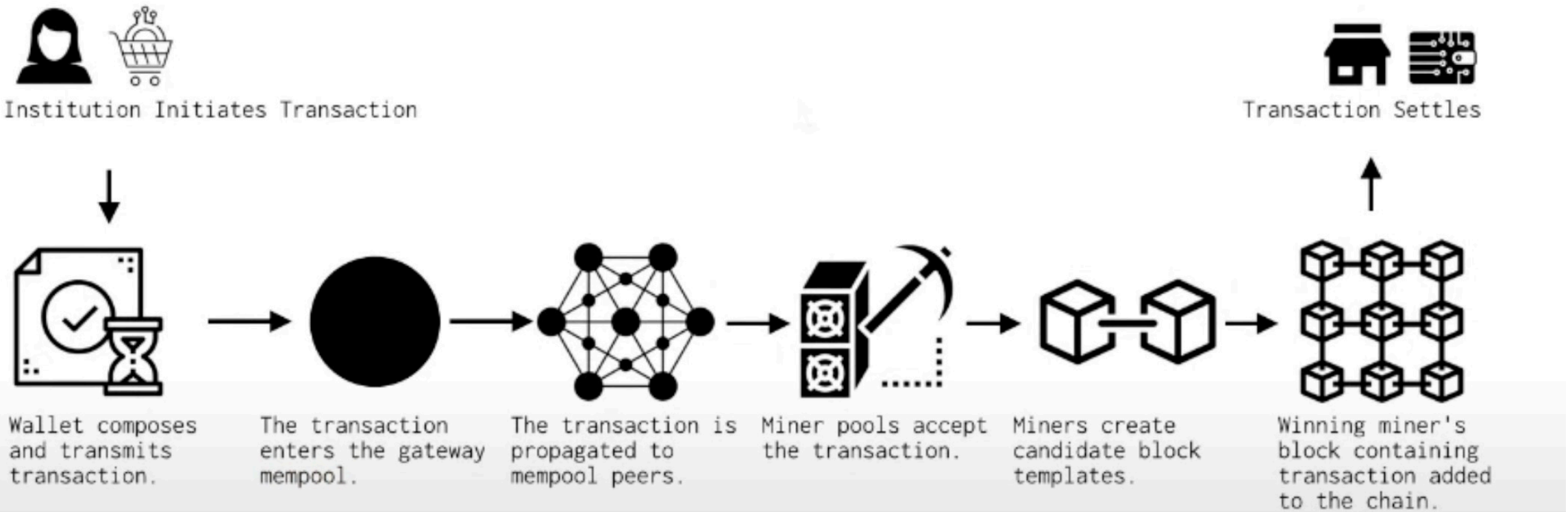
- Txs competing for confirmation
- Un-orchestrated fashion

Behind the scenes of Blockchain Transaction Operations and our **Area of Focus**

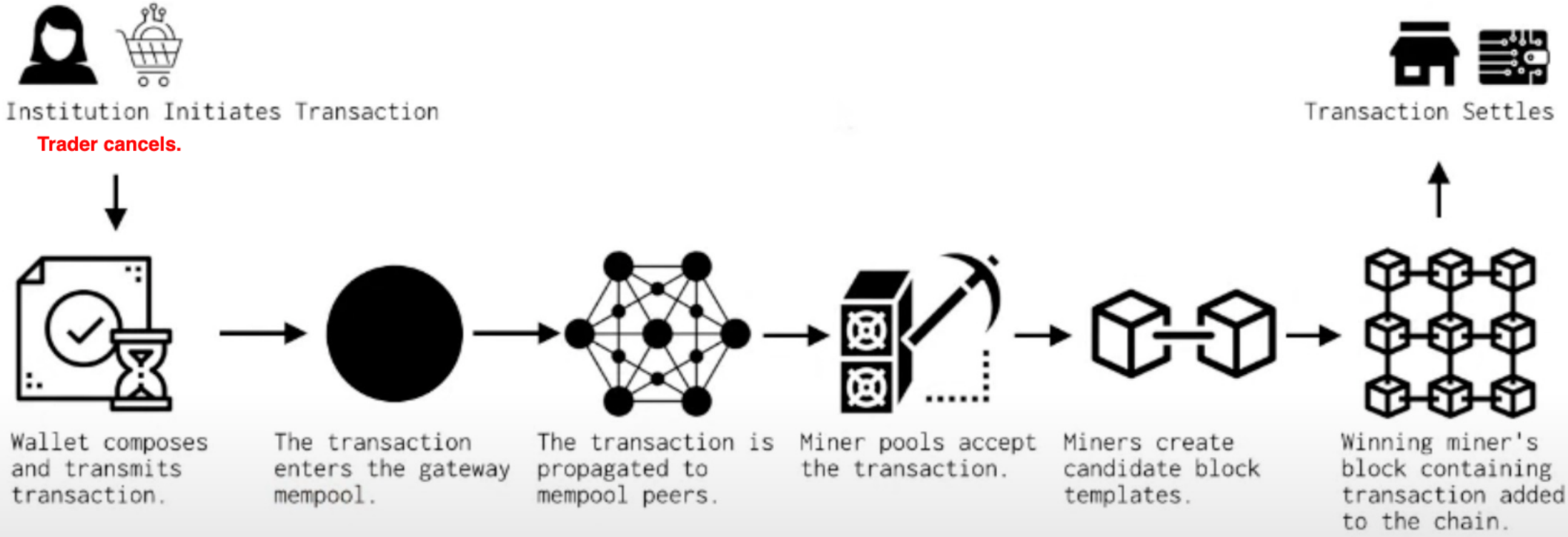
Blockchain Transaction Operations:
Behind The Scenes



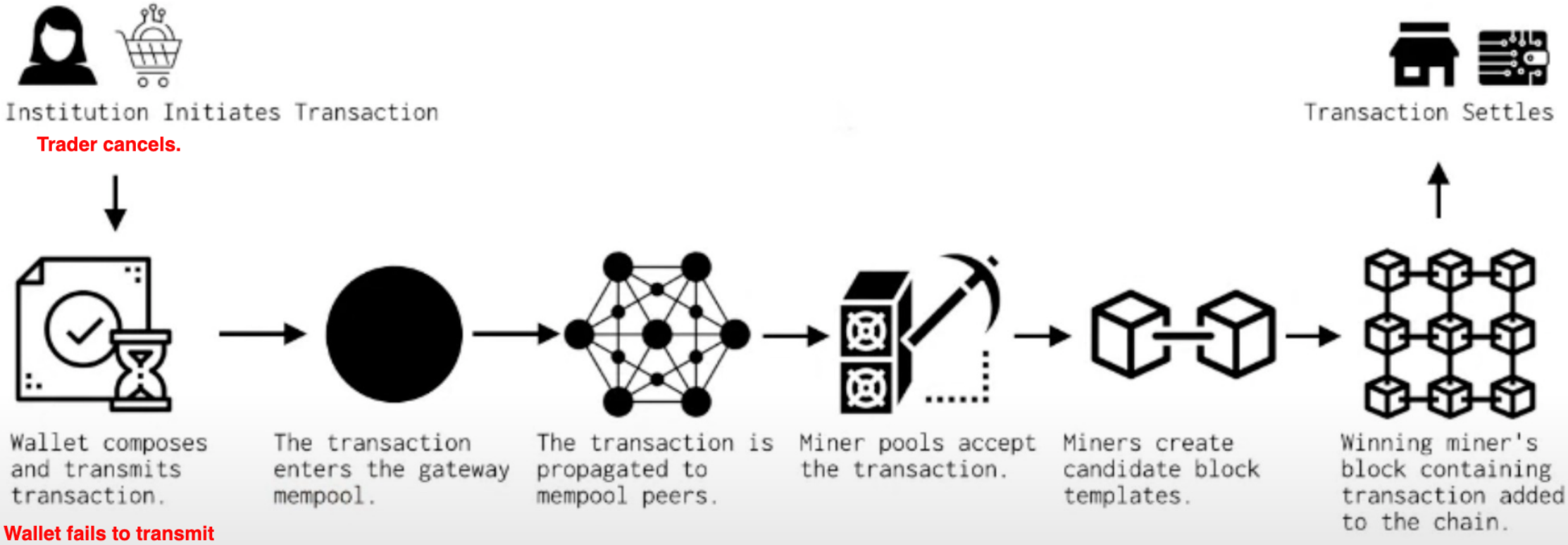
Blockchain Transaction Operations:
Risks



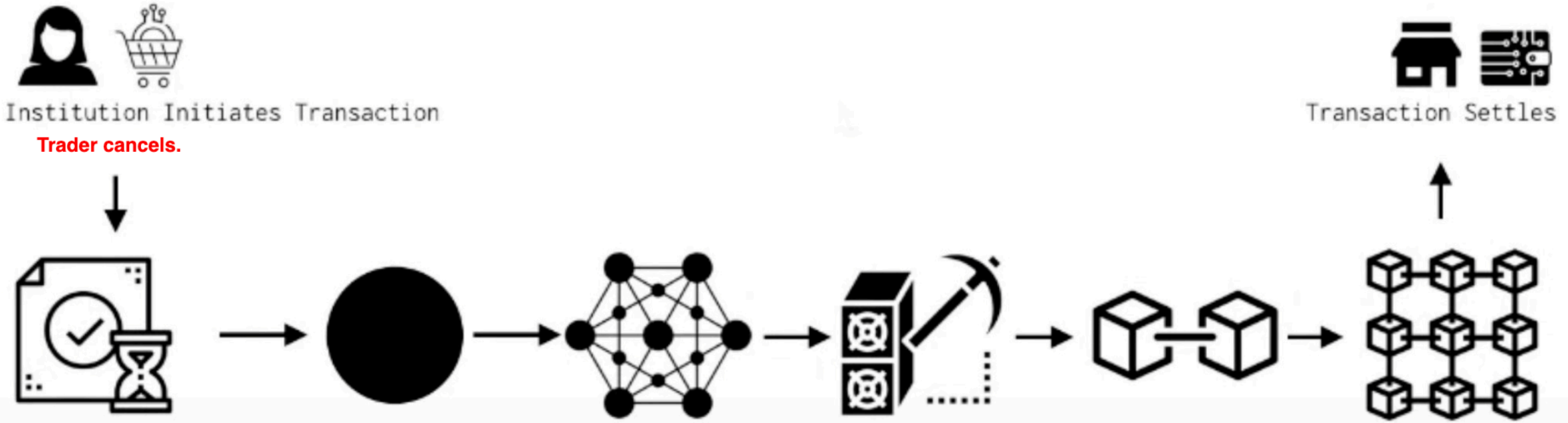
Blockchain Transaction Operations:
Risks



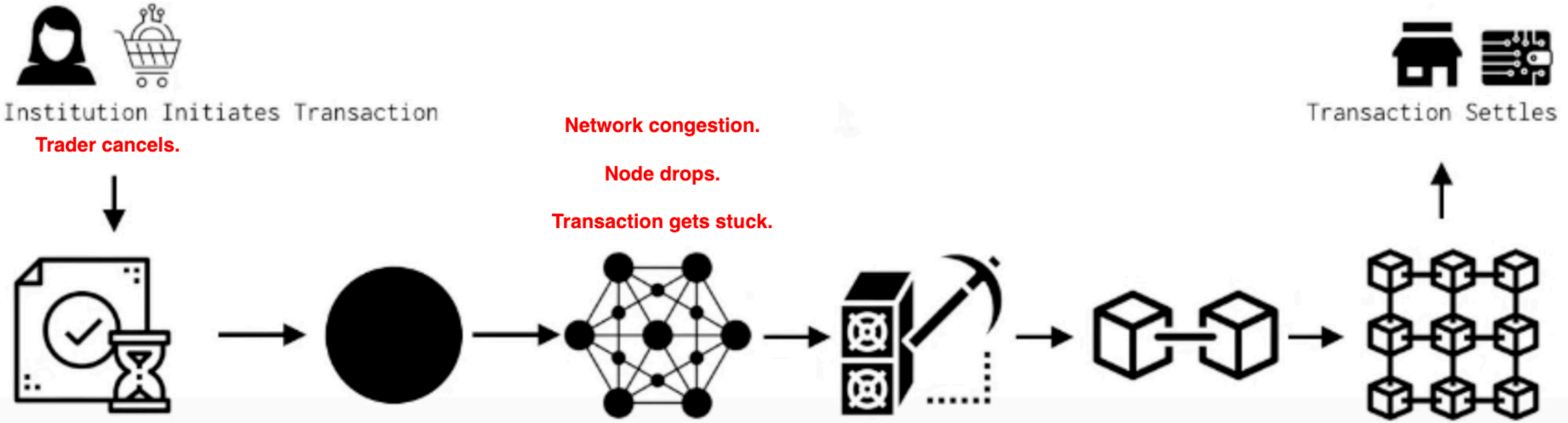
Blockchain Transaction Operations:
Risks



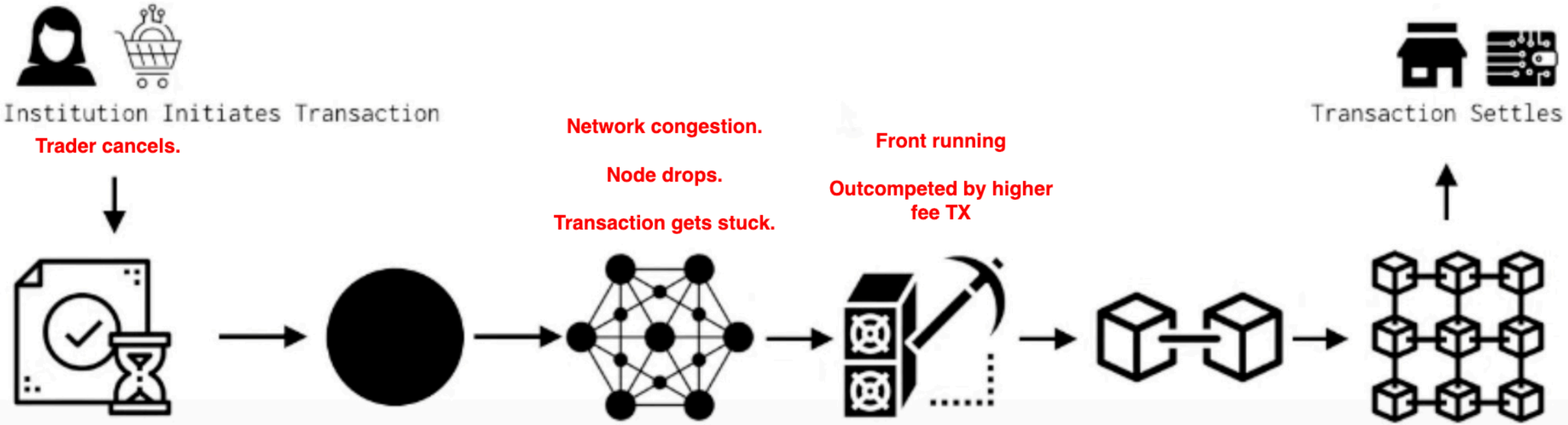
Blockchain Transaction Operations:
Risks



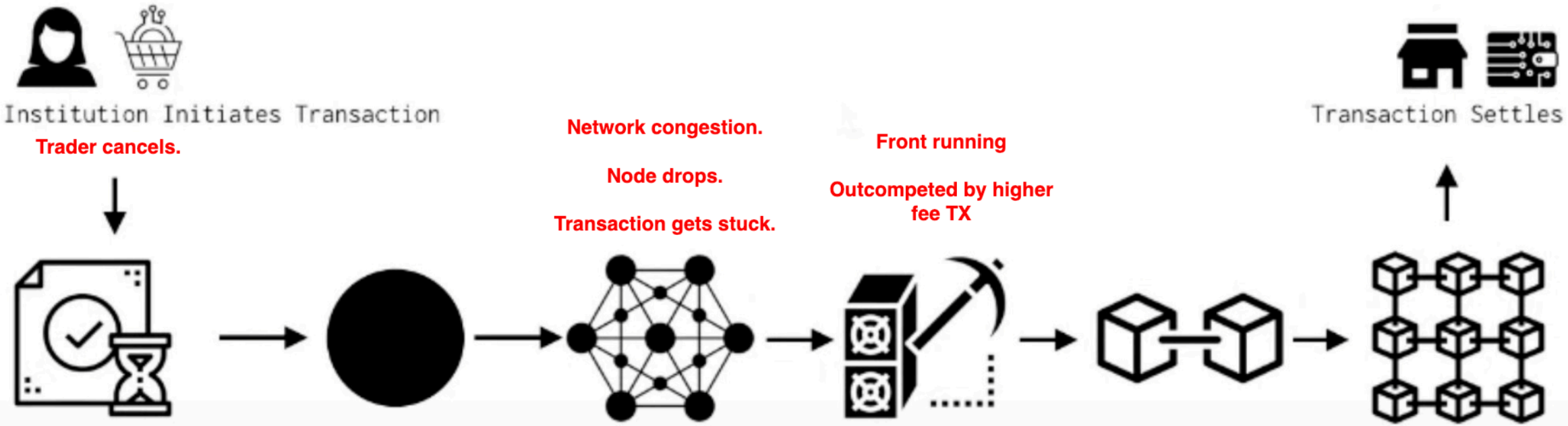
Blockchain Transaction Operations:
Risks



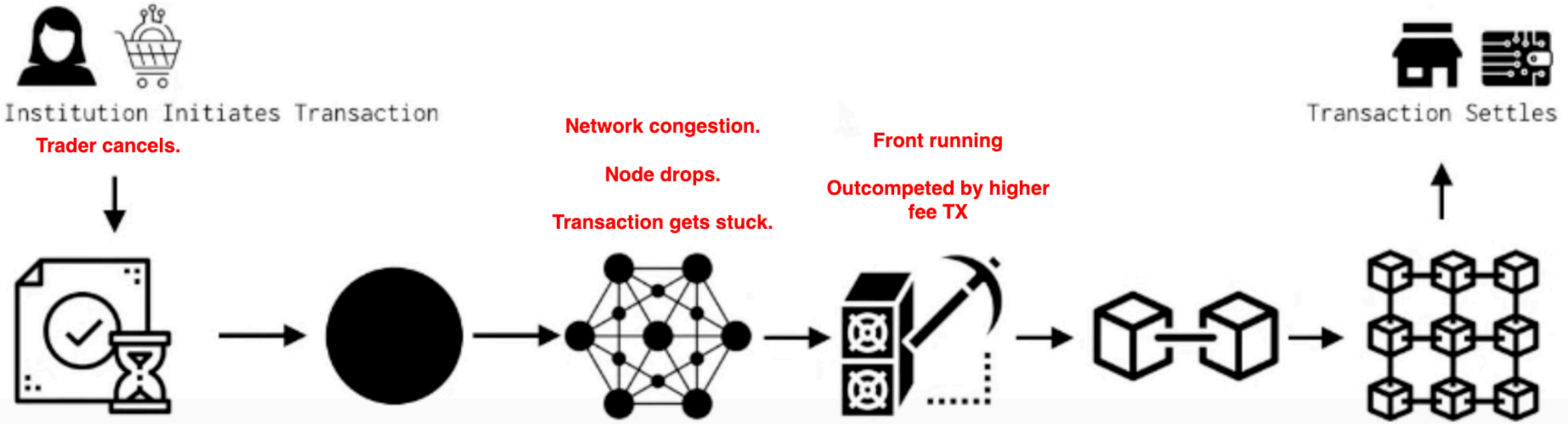
Blockchain Transaction Operations:
Risks



Blockchain Transaction Operations:
Risks



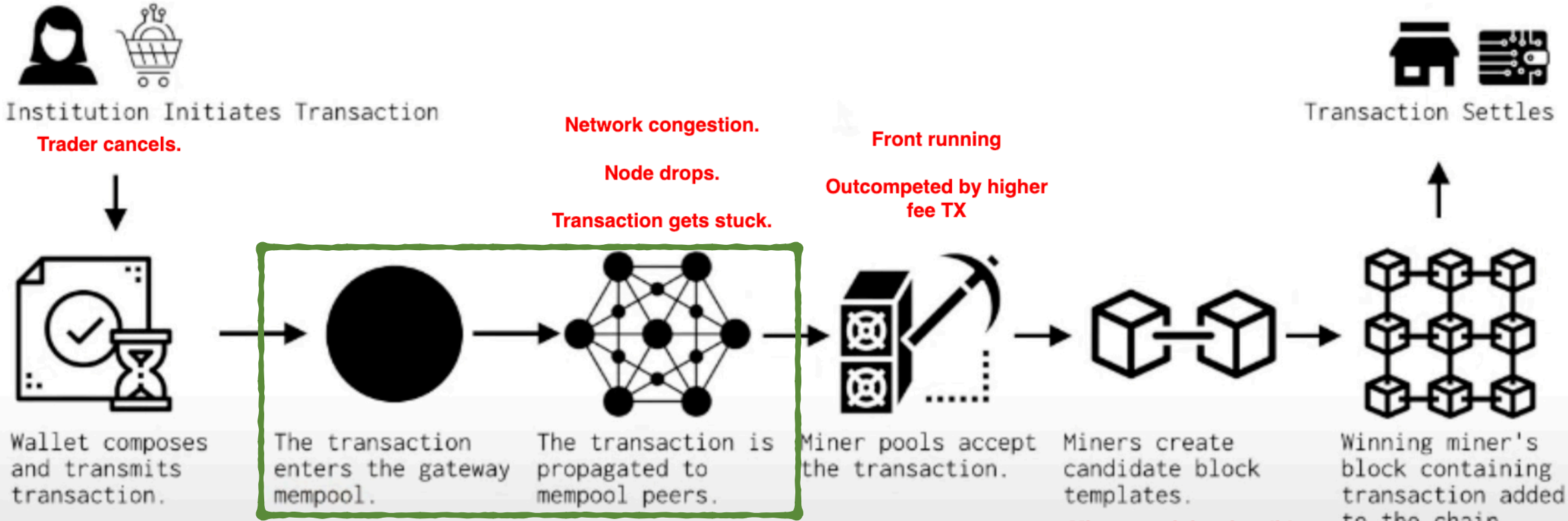
Blockchain Transaction Operations:
Risks



UCDAVIS COLLEGE OF ENGINEERING

Blockchain Transaction Operations:

Risks



NARWHAL AND TUSK

Narwhal

A Directed Acyclic Graph (DAG) based Mempool protocol

Tusk

Zero-message overhead asynchronous consensus protocol

Questions

What is DAG?

What is Mempool?

What is zero-message overhead?

NARWHAL AND TUSK

Narwhal

A Directed Acyclic Graph (DAG) based Mempool protocol

Tusk

Zero-message overhead asynchronous consensus protocol

Questions

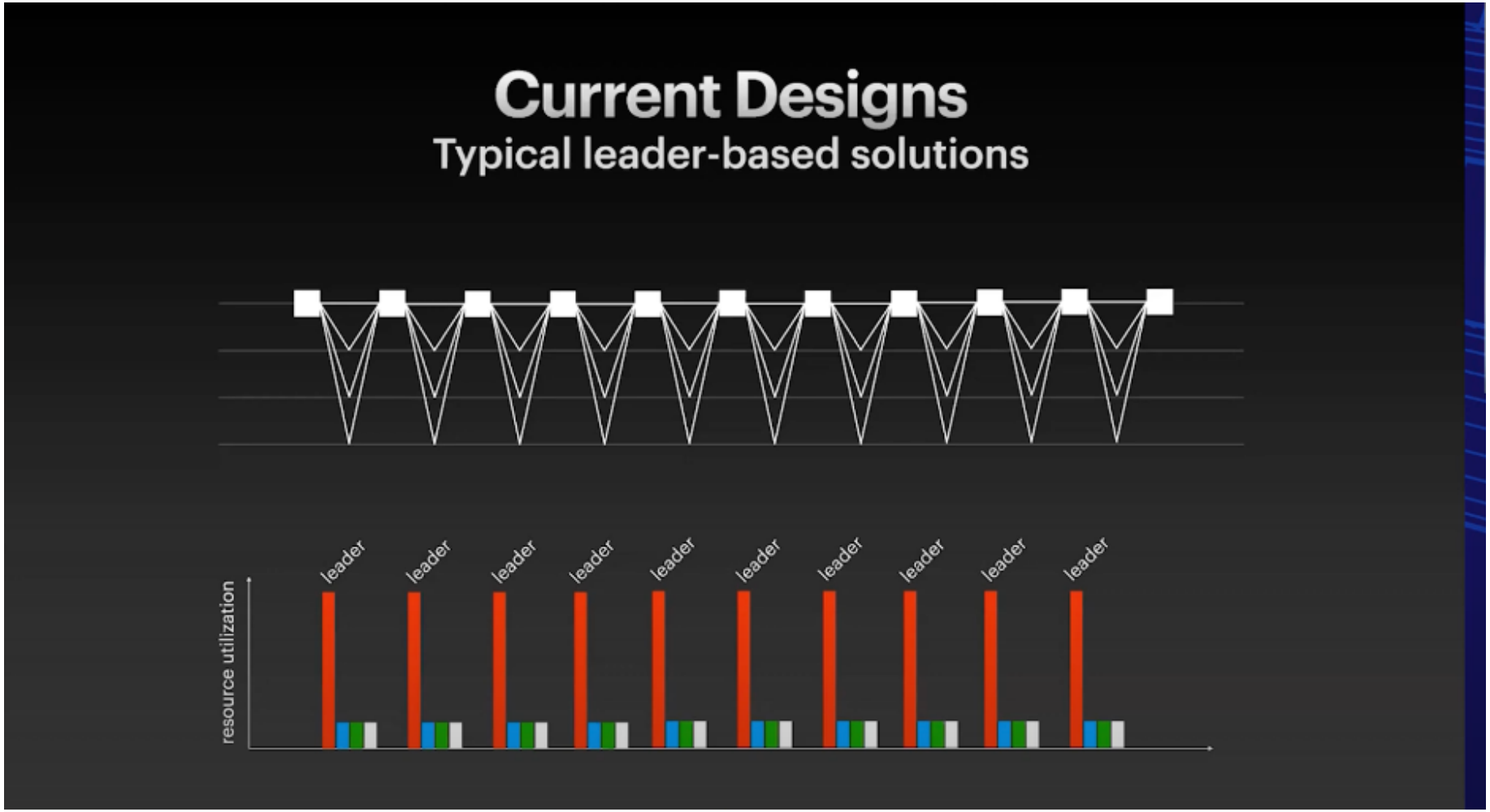
What is DAG?

What is Mempool?

What is zero-message overhead?

Consensus engine **does not send any message** at all. Hence the term.

NON-EGALITARIAN RESOURCE UTILIZATIONS

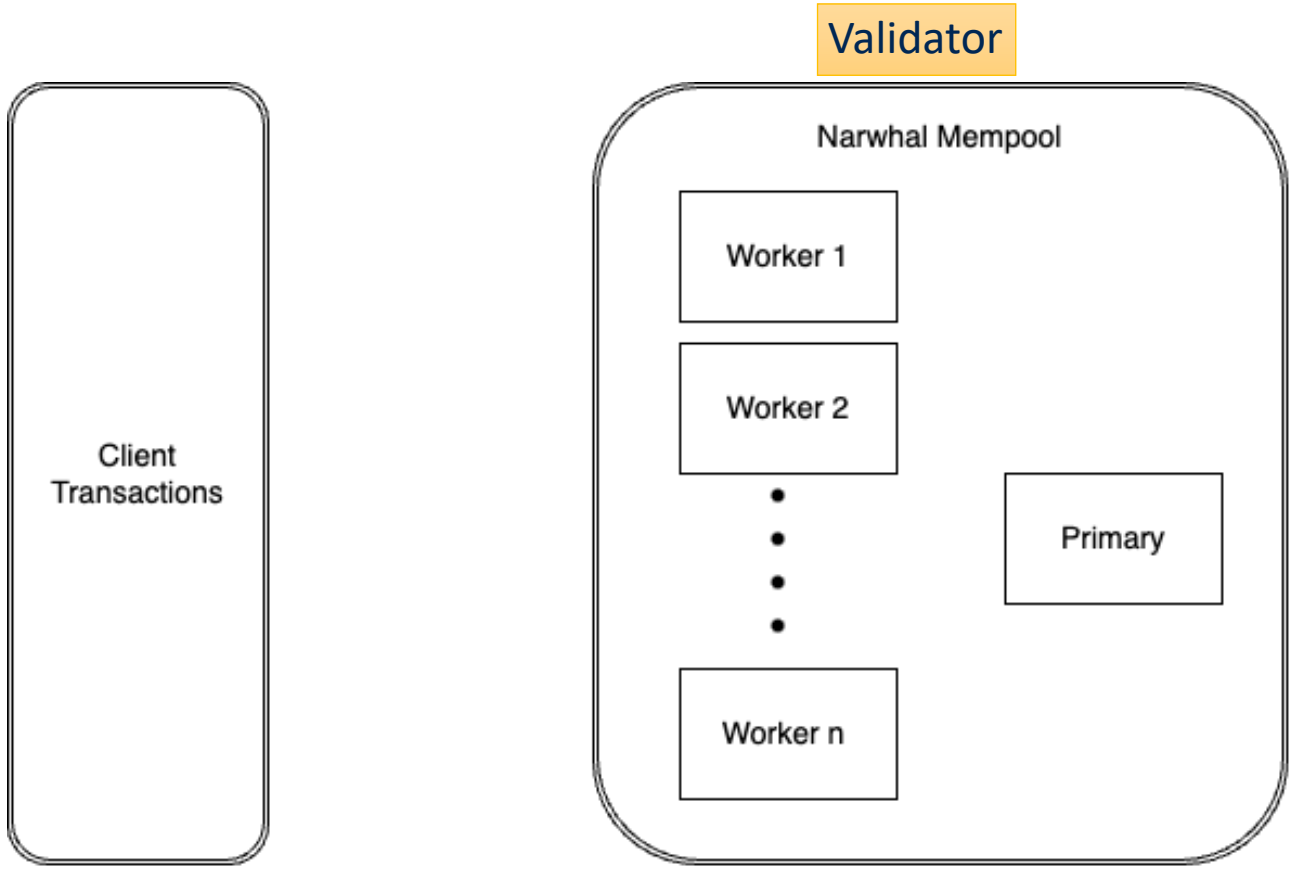


MEMPOOL IS THE KEY

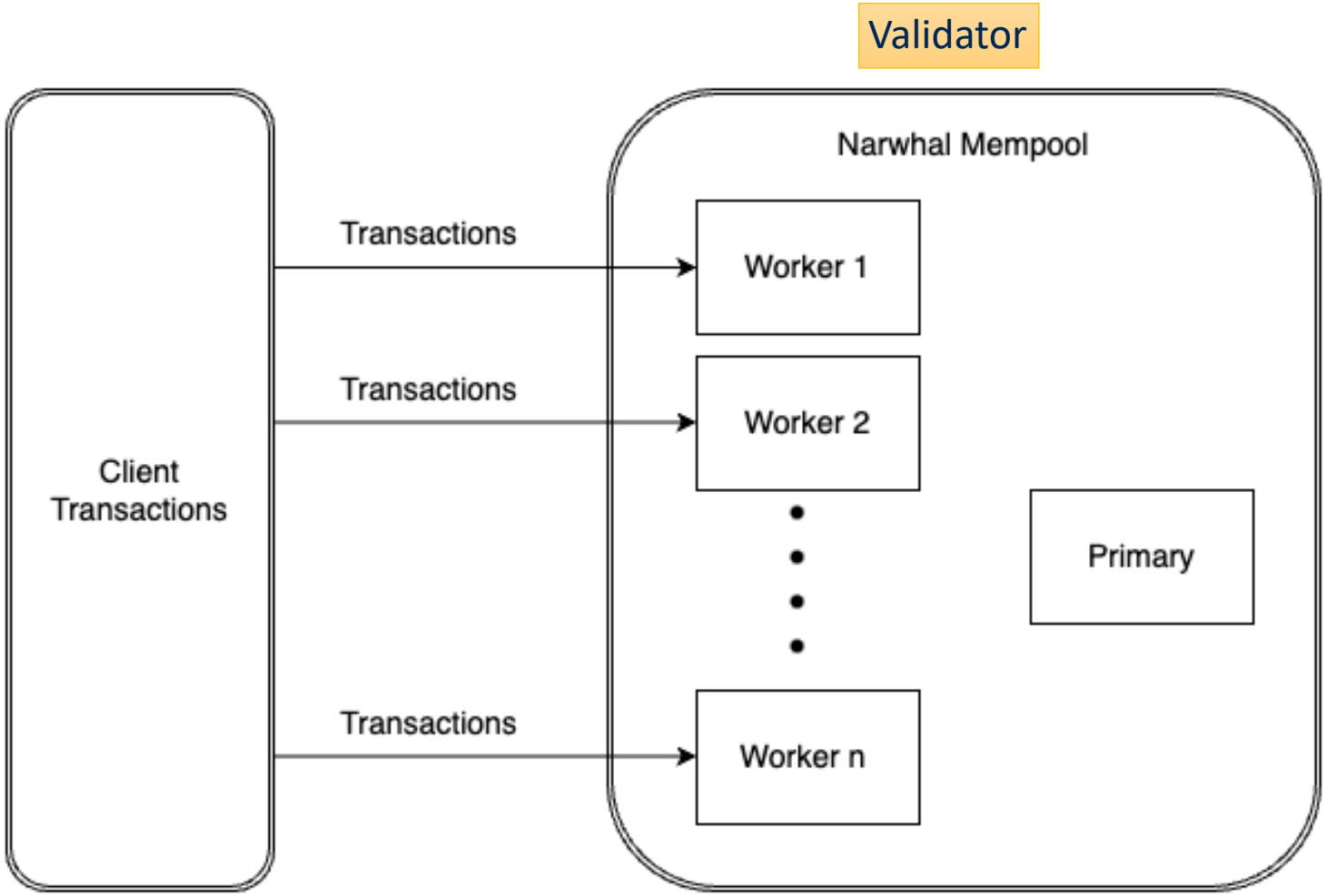
NARWHAL

Narwhal

The workers and the primary in Mempool

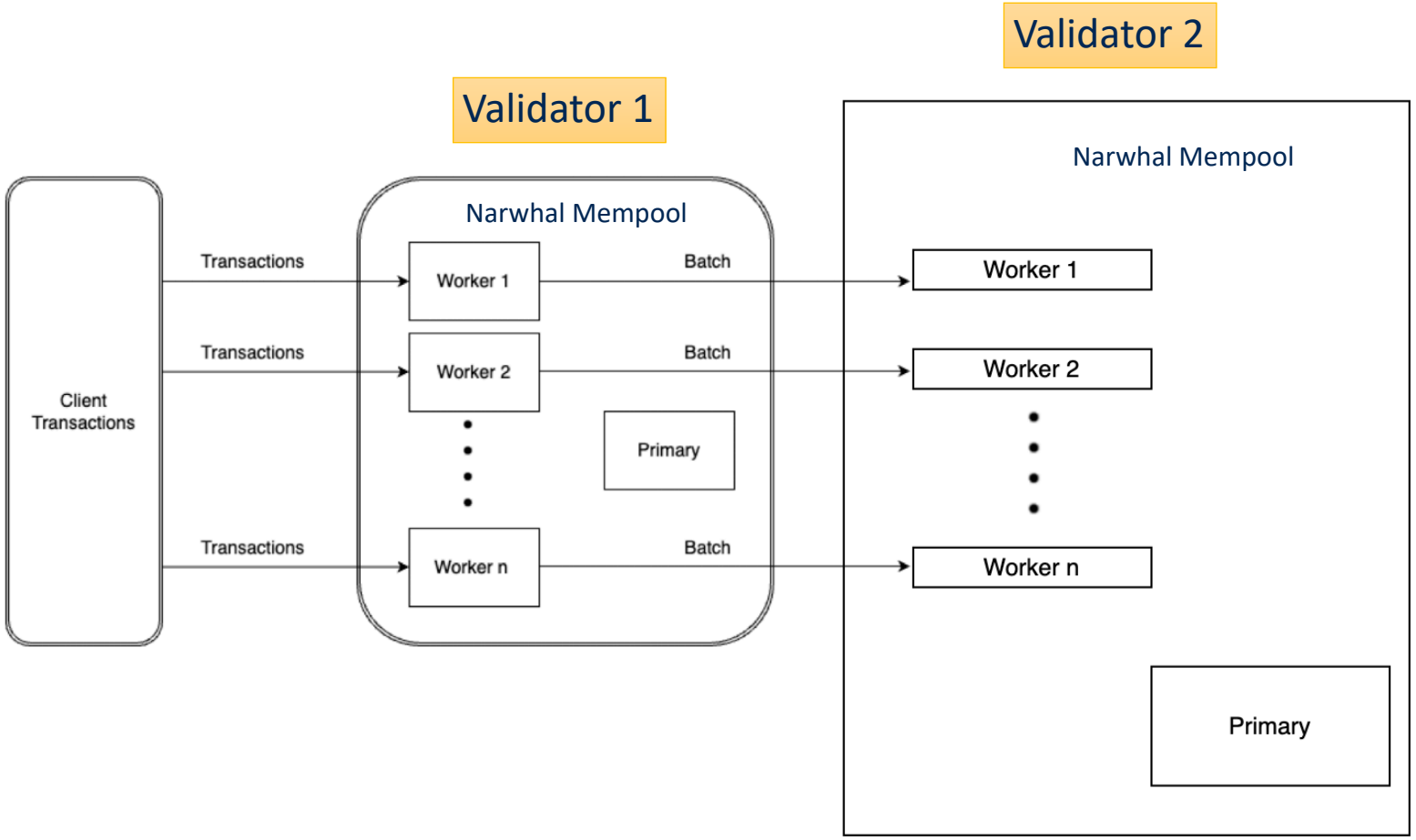


Client sends transactions

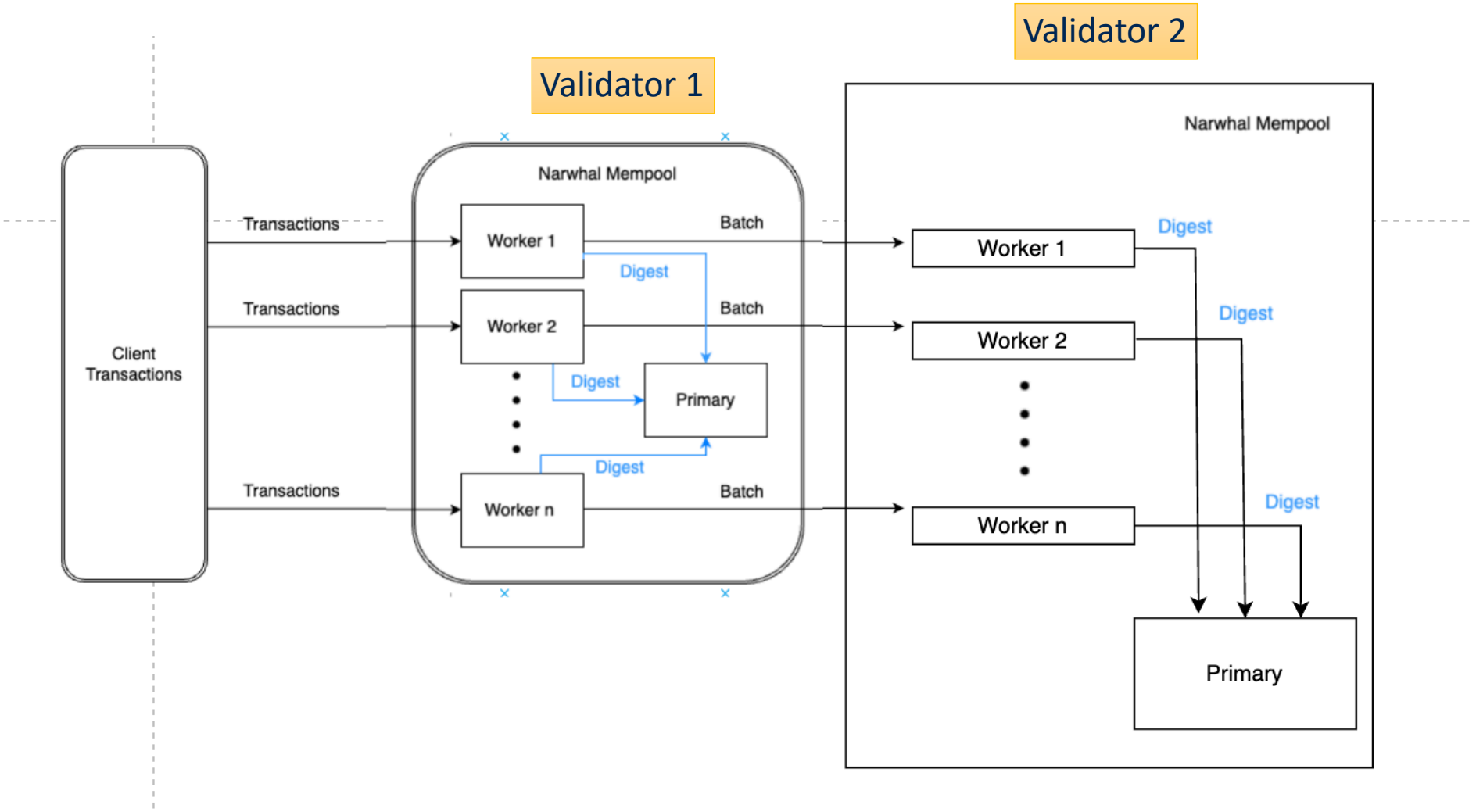


UCDAVIS COLLEGE OF ENGINEERING

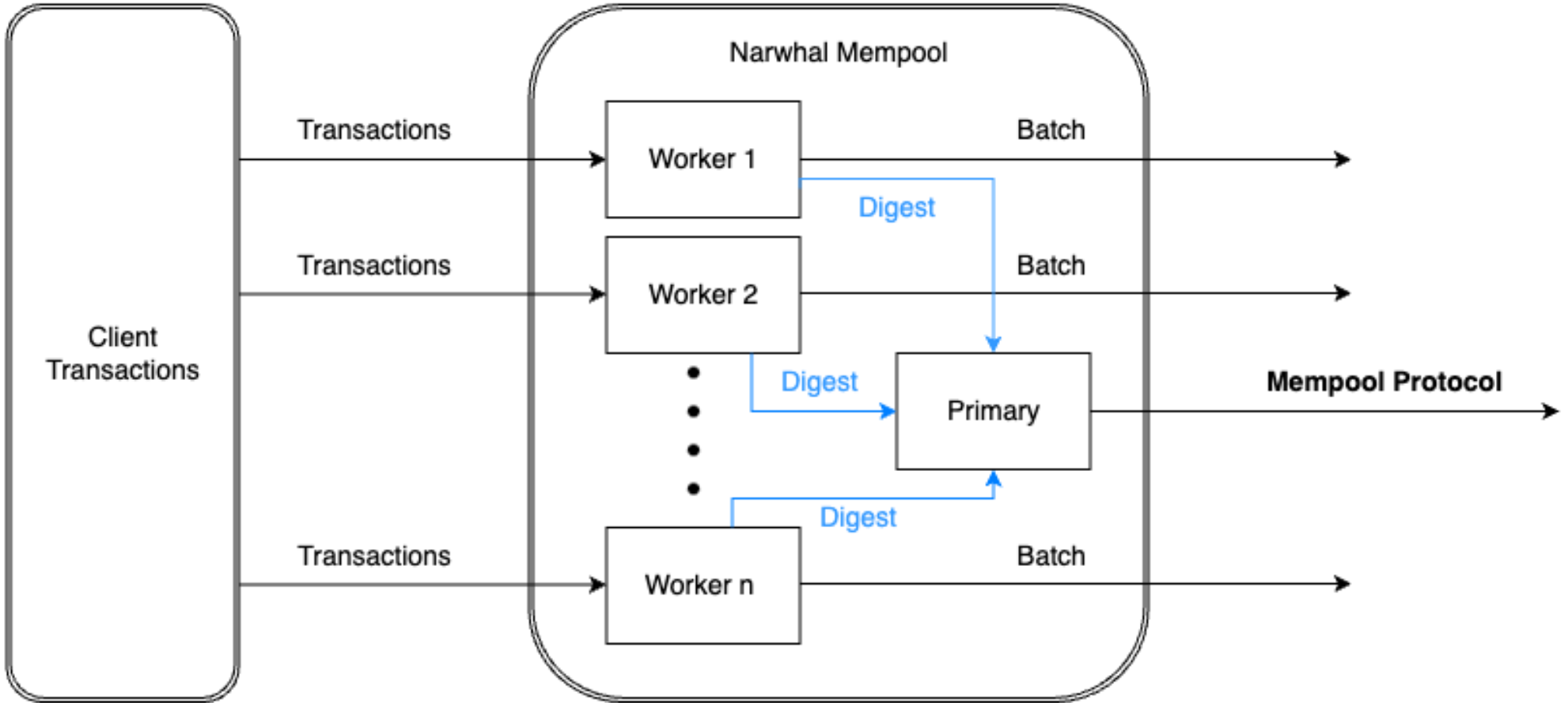
Batching Transactions



Digest sent to primary



Mempool Protocol Initialization



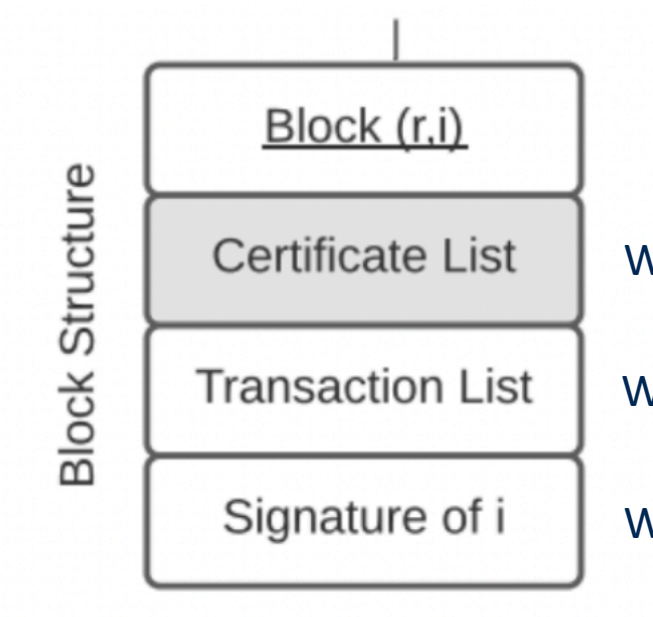
UCDAVIS COLLEGE OF ENGINEERING

Every Primary creates a Block

Every Primary creates a Block

What does it contain?

Block Structure



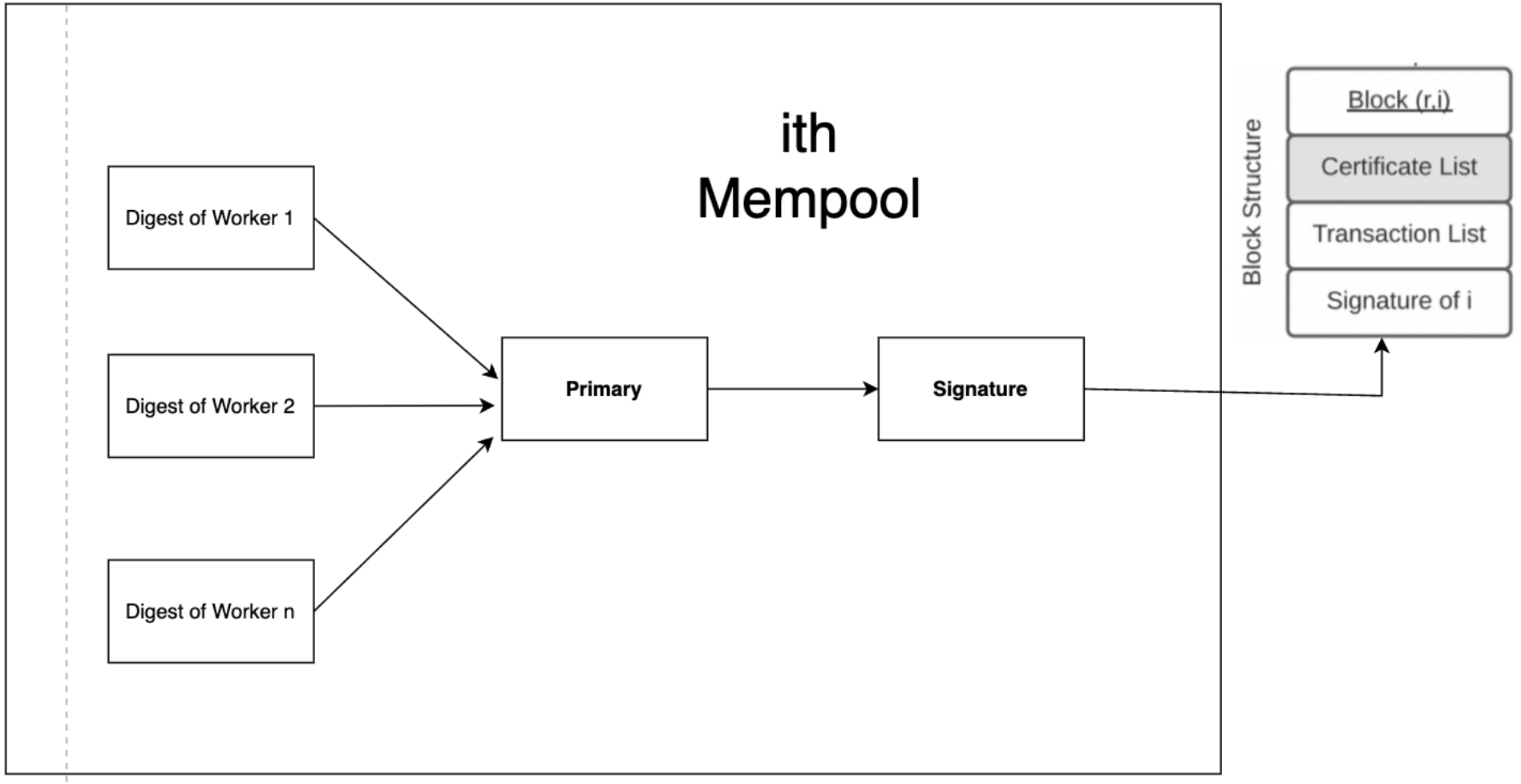
Where is this coming from?

We know where this is from. The Client

Where is this coming from?

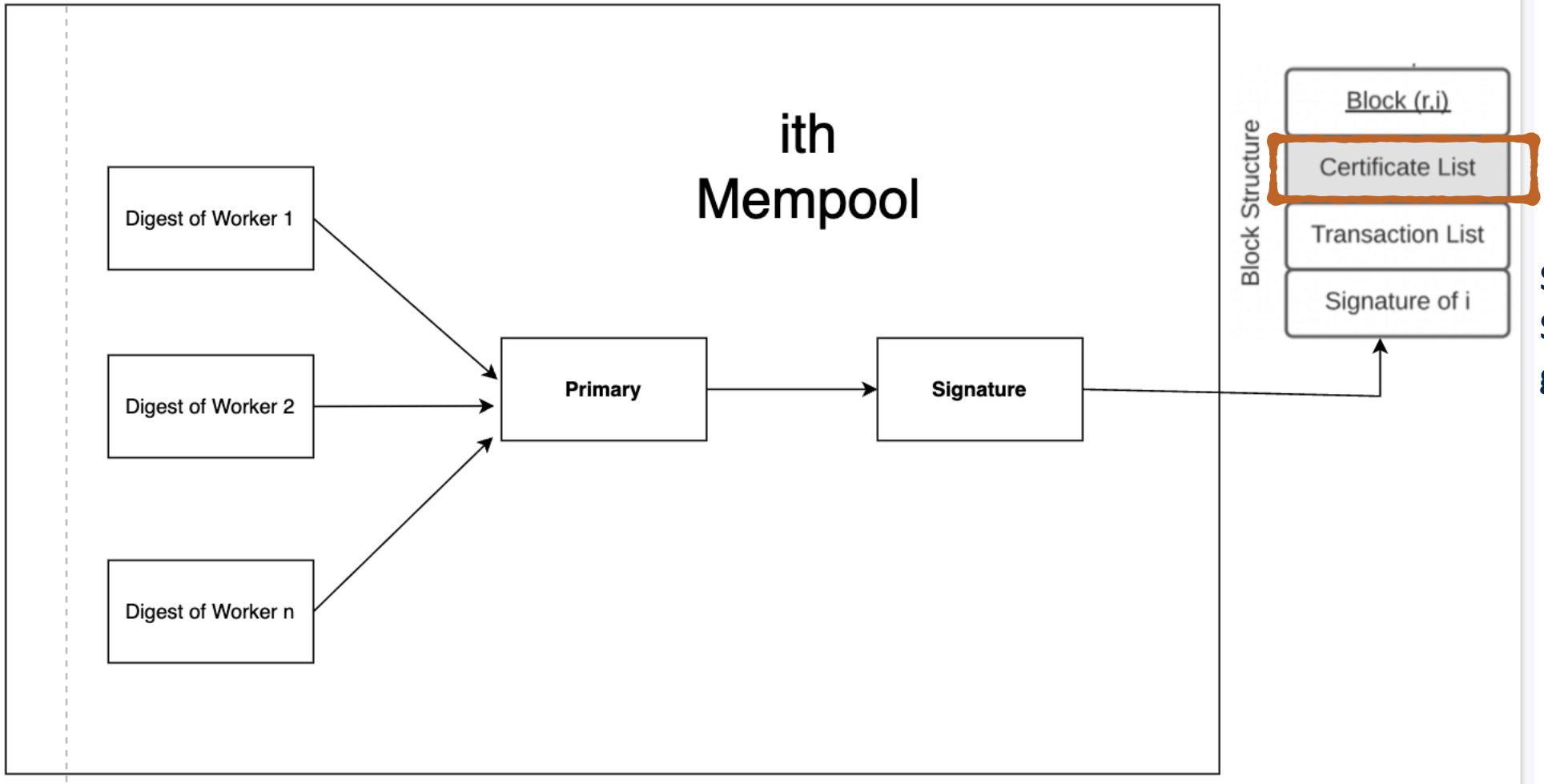
UCDAVIS COLLEGE OF ENGINEERING

Signature of the Block



So this is how Signature is generated

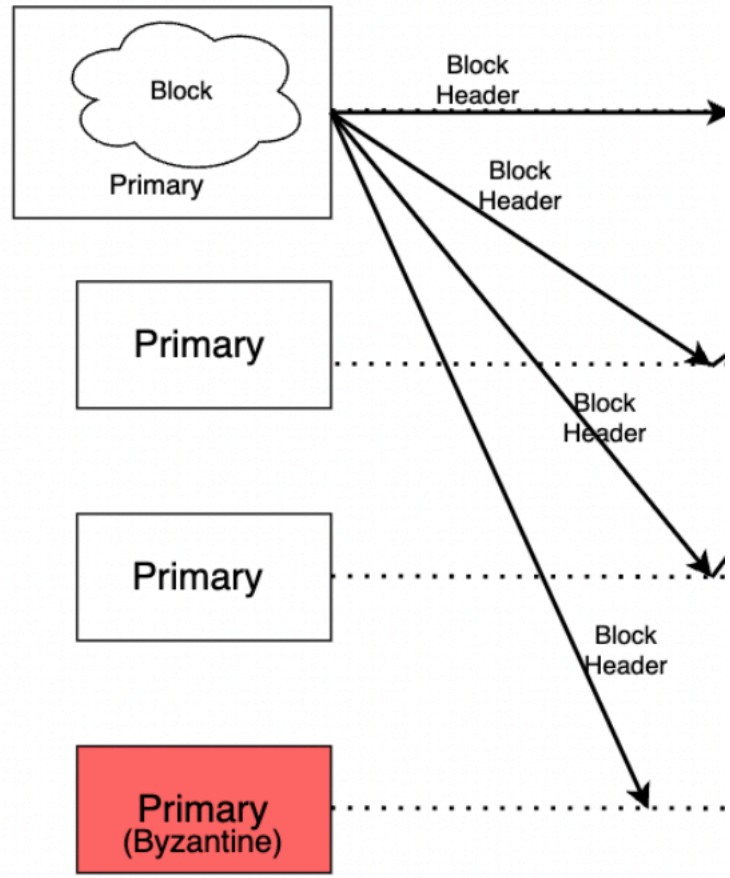
Signature of the Block

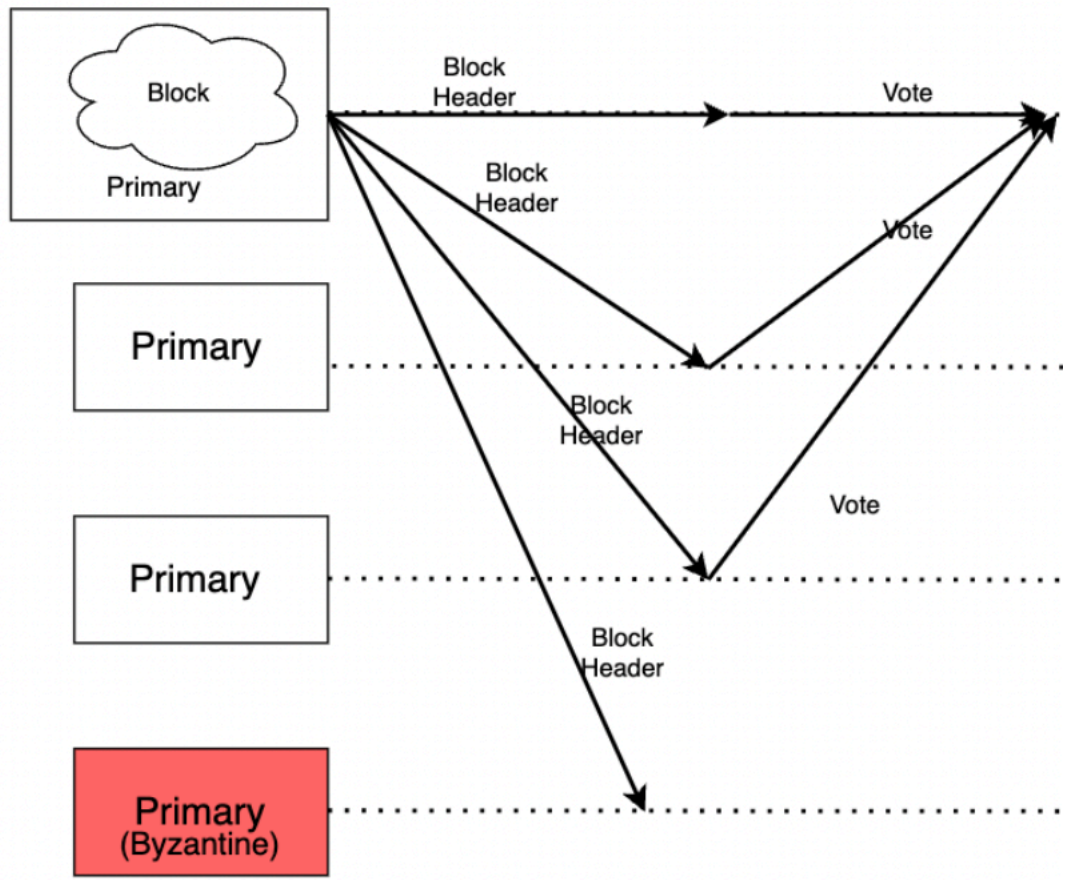


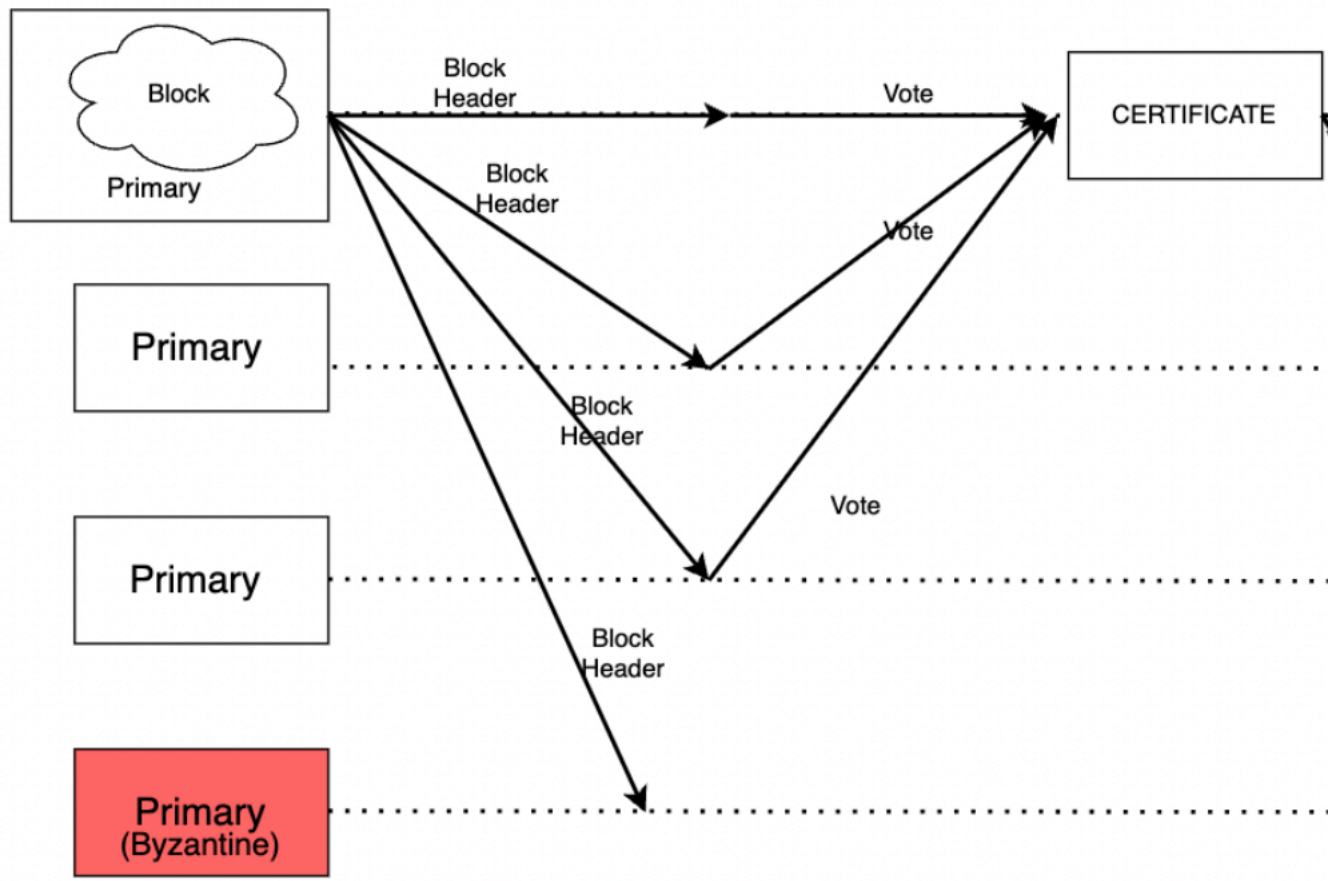
What about this?

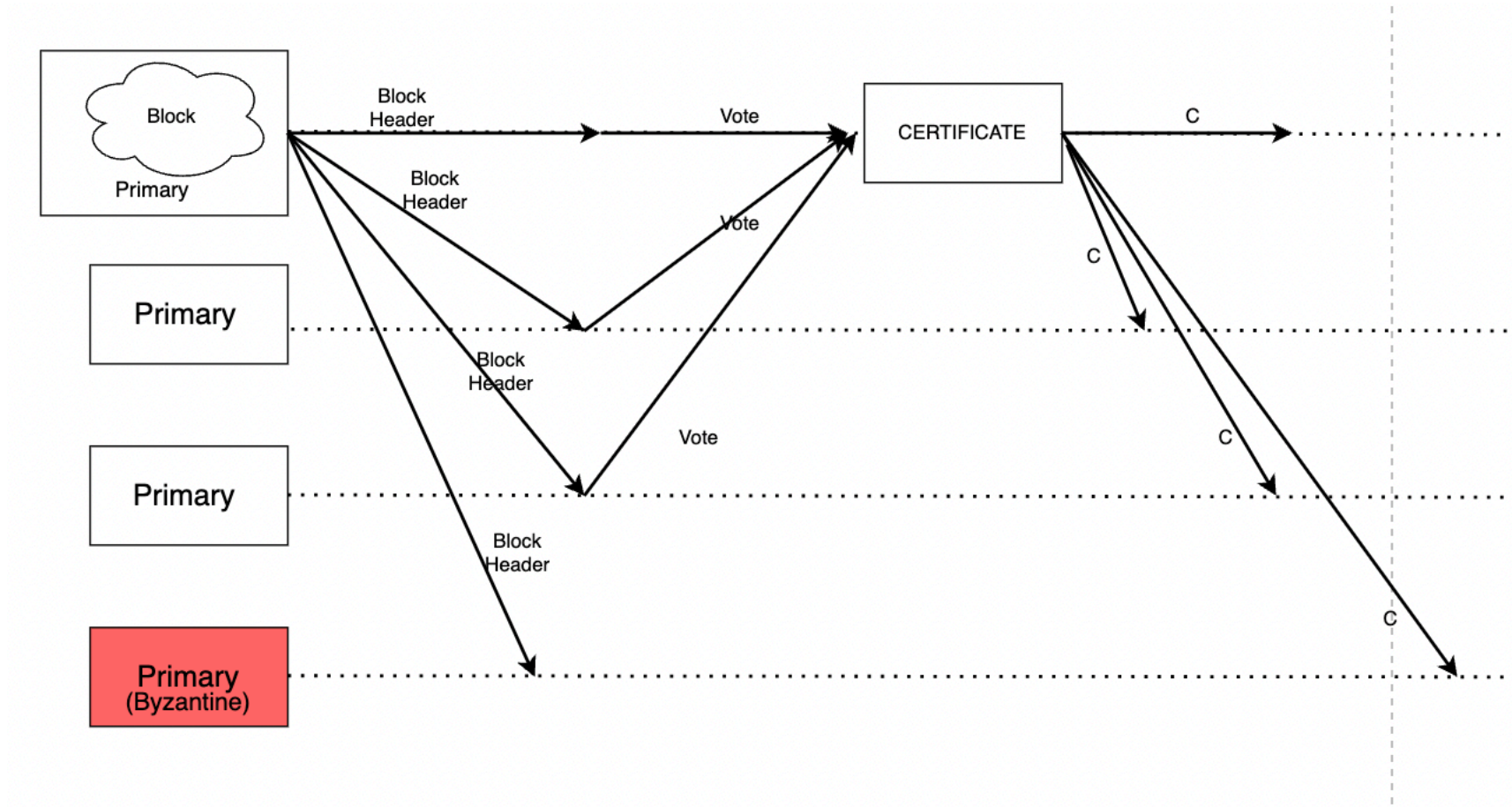
So this is how Signature is generated

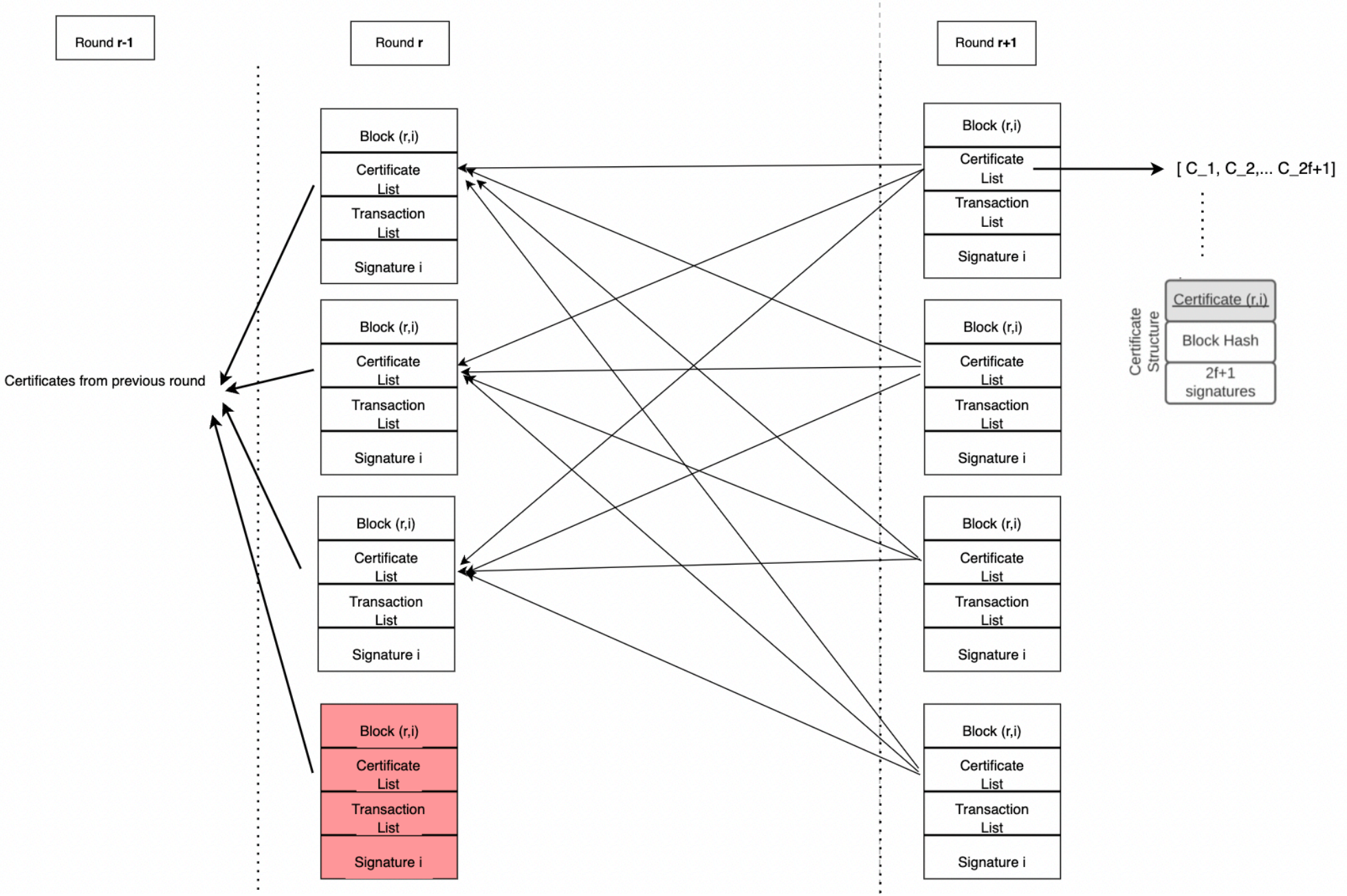
Where are these certificates coming from?


Block
Primary
Block Header
Block Header
Primary
Block Header
Primary
Block Header
Primary
(Byzantine)

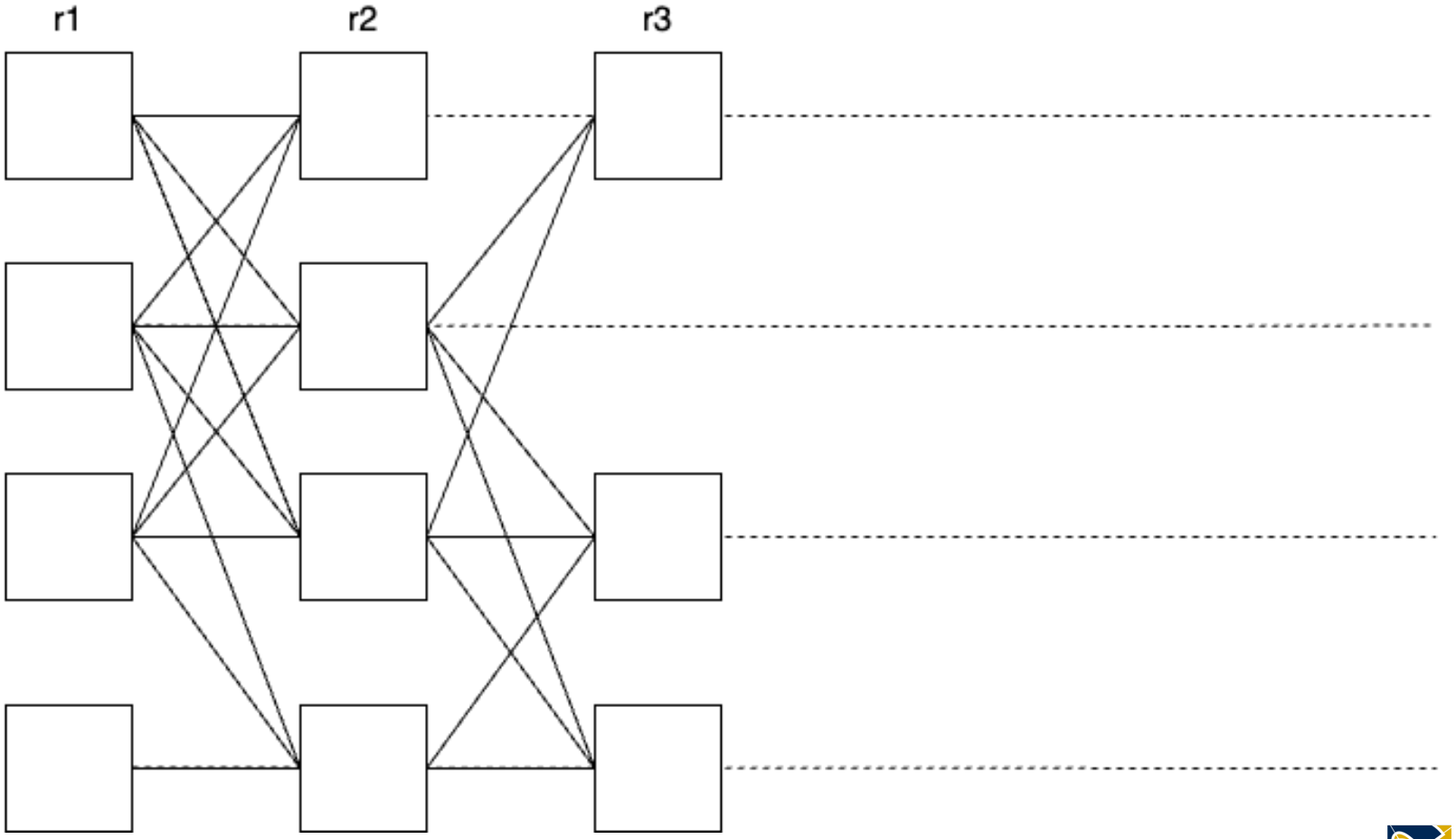

Block
Primary
Block Header
Vote
Block Header
Vote
Primary
Block Header
Vote
Primary
Block Header
Primary
(Byzantine)


Block
Primary
Block Header
Vote
CERTIFICATE
Block Header
Vote
Primary
Block Header
Vote
Primary
Block Header
Primary
(Byzantine)


Block
Primary
Block Header
Vote
CERTIFICATE
C
Block Header
Vote
C
Primary
Block Header
Vote
C
Primary
Block Header
C
Primary
(Byzantine)

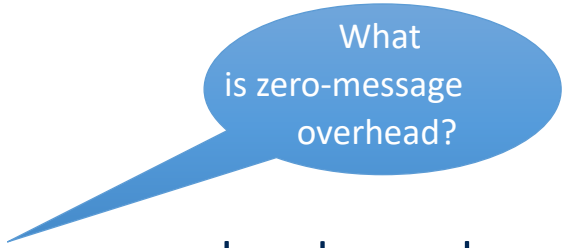

Round r-1
Round r
Round r+1
Certificates from previous round
Block (r,i)
Certificate List
Transaction List
Signature i
[C_1, C_2,... C_2f+1]
Certificate Structure
Certificate (r,i)
Block Hash
2f+1 signatures

This is the DAG created in Narwhal



UCDAVIS COLLEGE OF ENGINEERING

Tusk

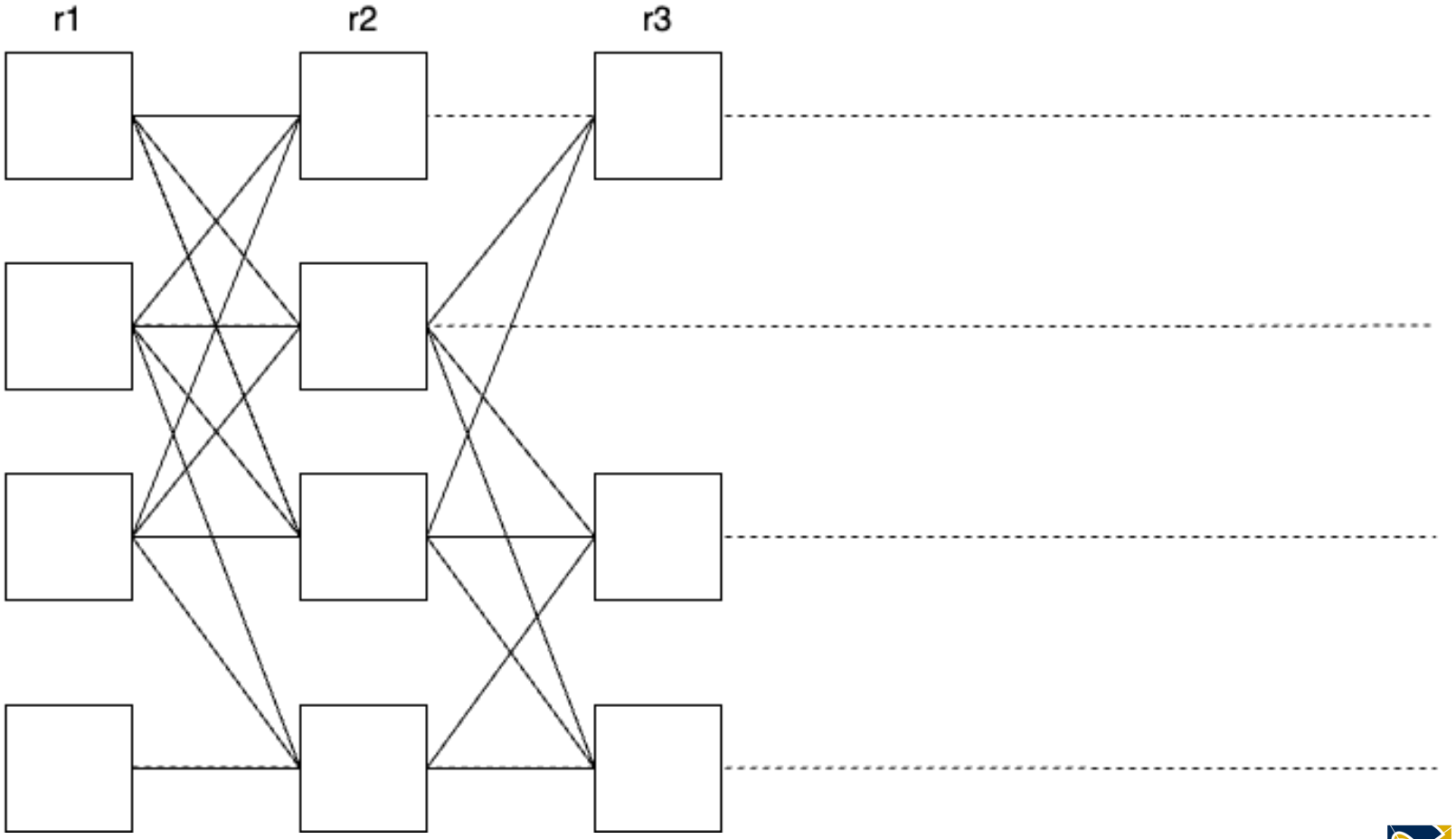


- A zero-message overhead asynchronous consensus protocol
- DAG —> Starting point
- Three consecutive rounds
  - First round - Each validator proposes its block and **causal** history (**PROPOSE**)
  - Second round - Each validator votes on the proposal by including them in their block (**VOTE**)
  - Third round - Validators produce randomness to elect a random leader's block.

UCDAVIS COLLEGE OF ENGINEERING

Tusk

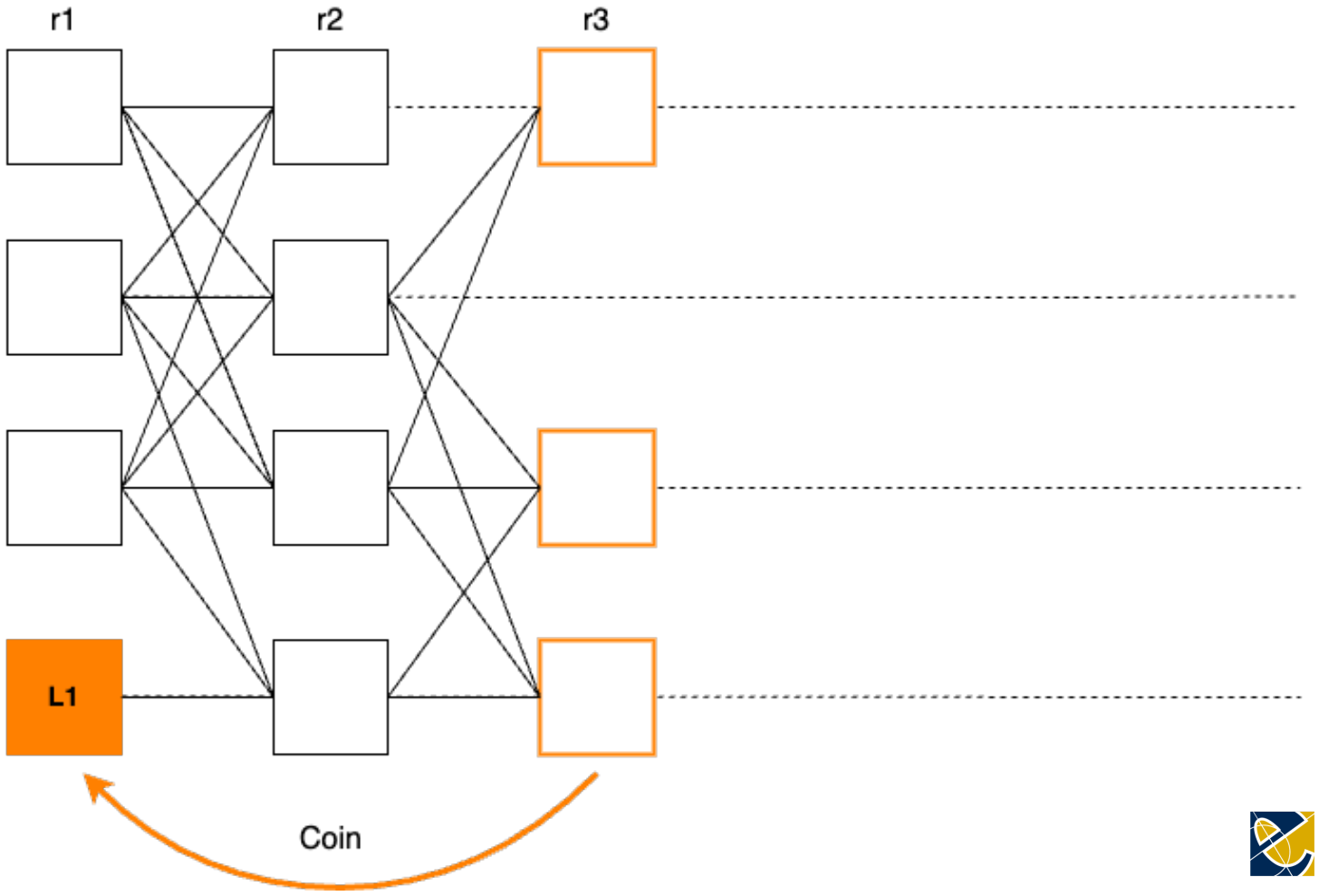
Just interpret the DAG created in Narwhal



UCDAVIS COLLEGE OF ENGINEERING

Tusk

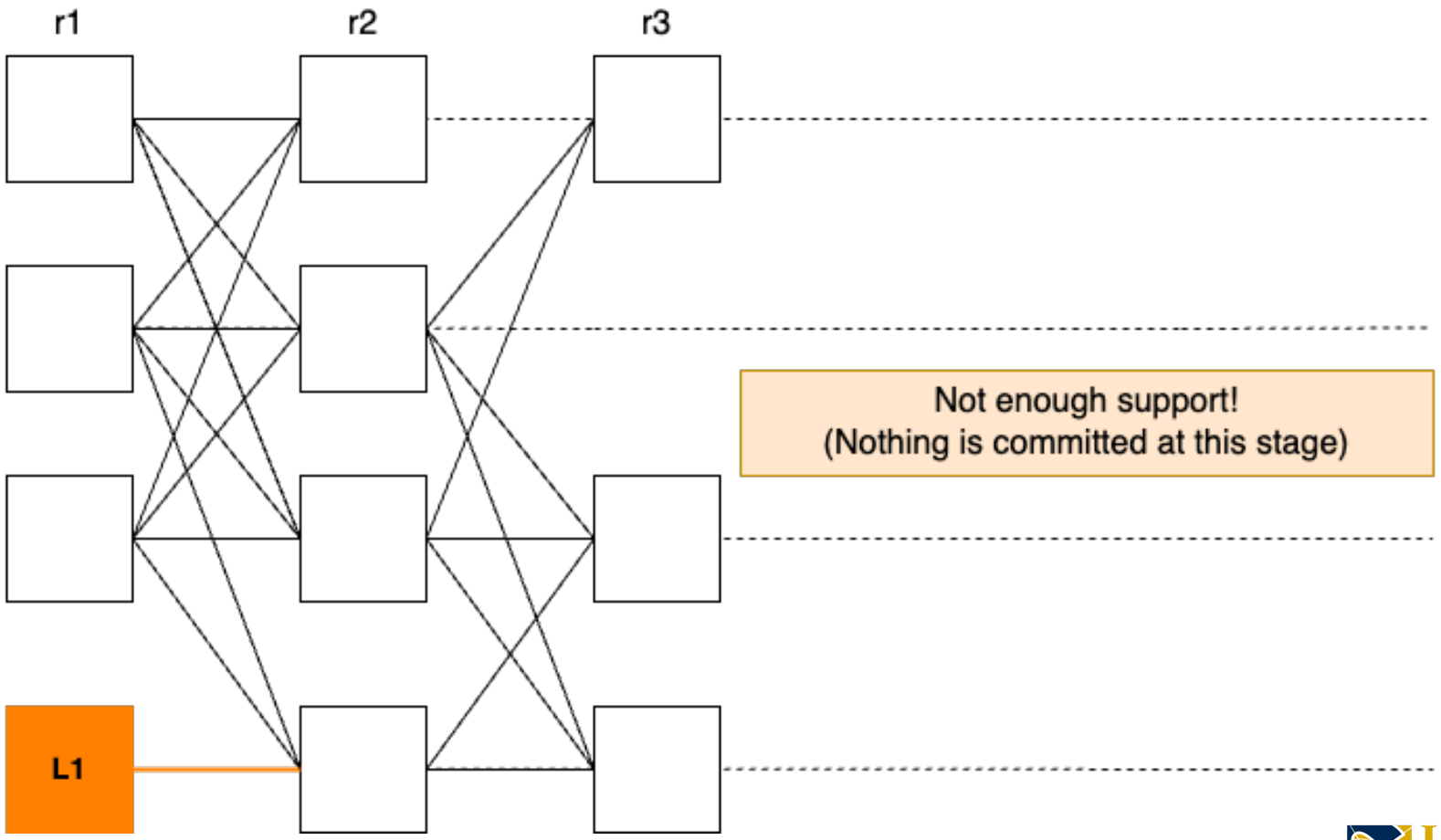
The random coin elects the leader of r-2



UCDAVIS COLLEGE OF ENGINEERING

Tusk

The chosen leader needs **f+1** links from round **r-1 to get committed**

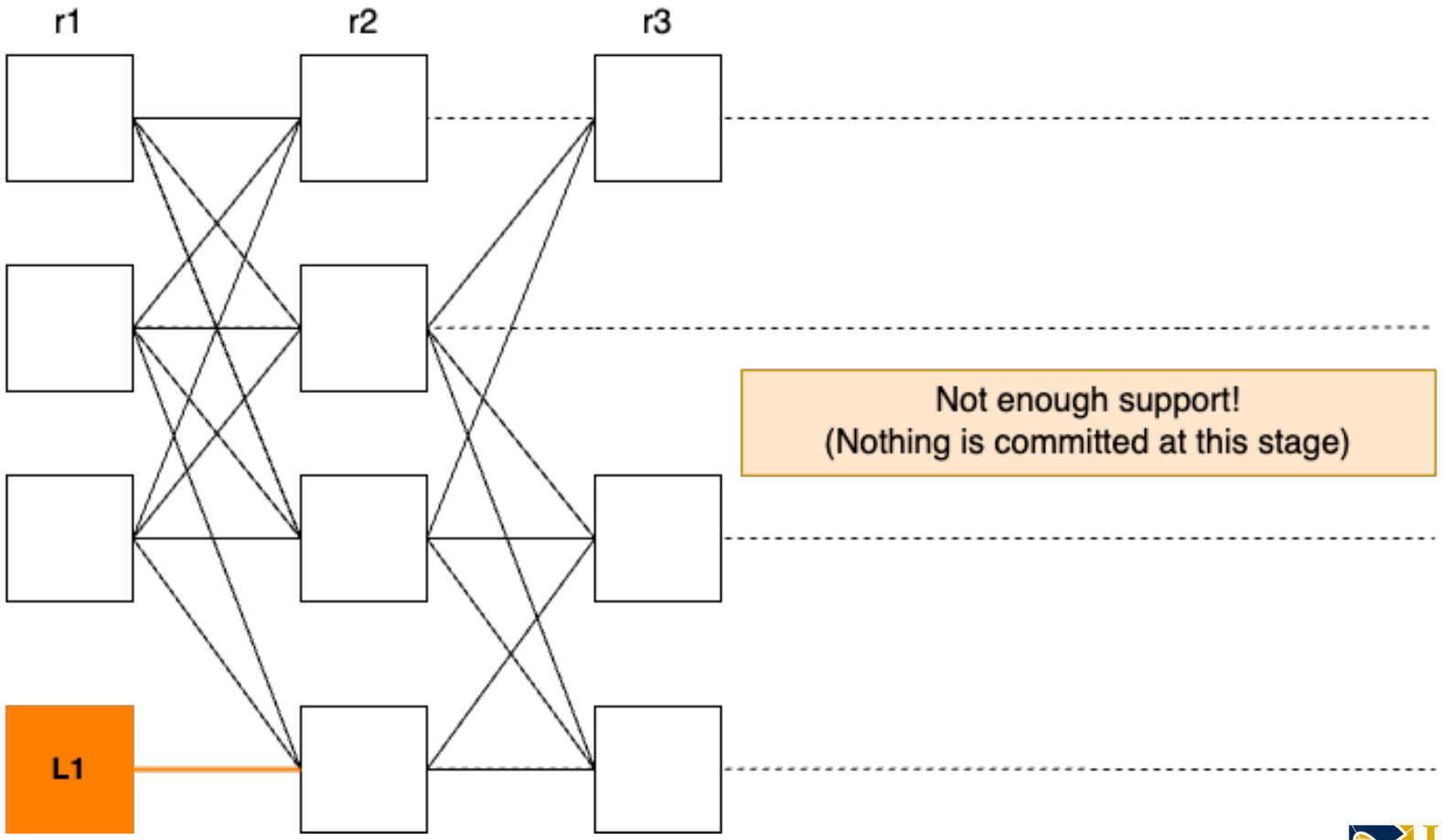


UCDAVIS COLLEGE OF ENGINEERING

Tusk

Let us discuss the intuition behind f+1 later.

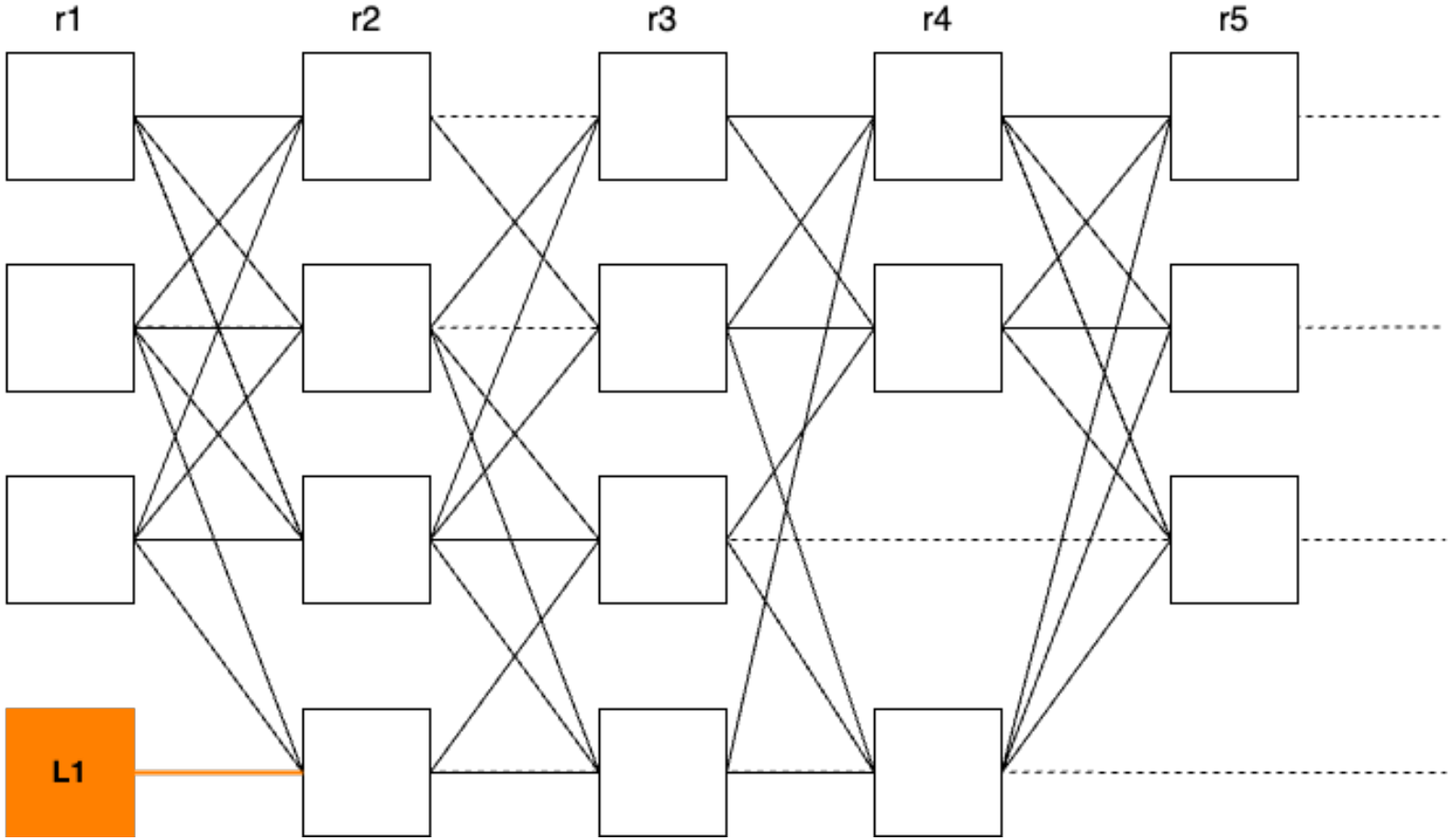
The chosen leader needs **f+1** links from round **r-1 to get committed**



UCDAVIS COLLEGE OF ENGINEERING

Tusk

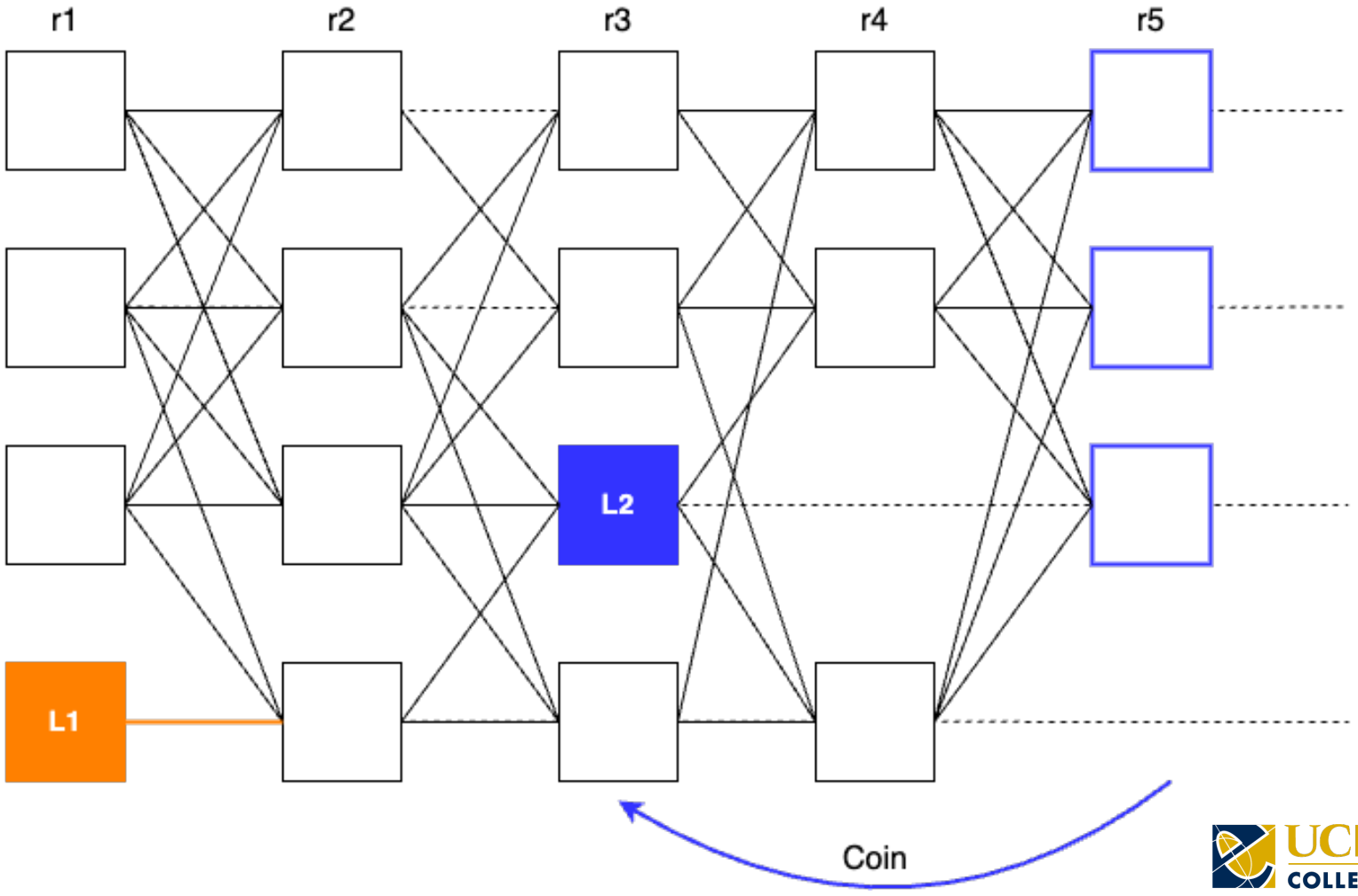
Nothing is committed so we keep building the DAG



UCDAVIS COLLEGE OF ENGINEERING

Tusk

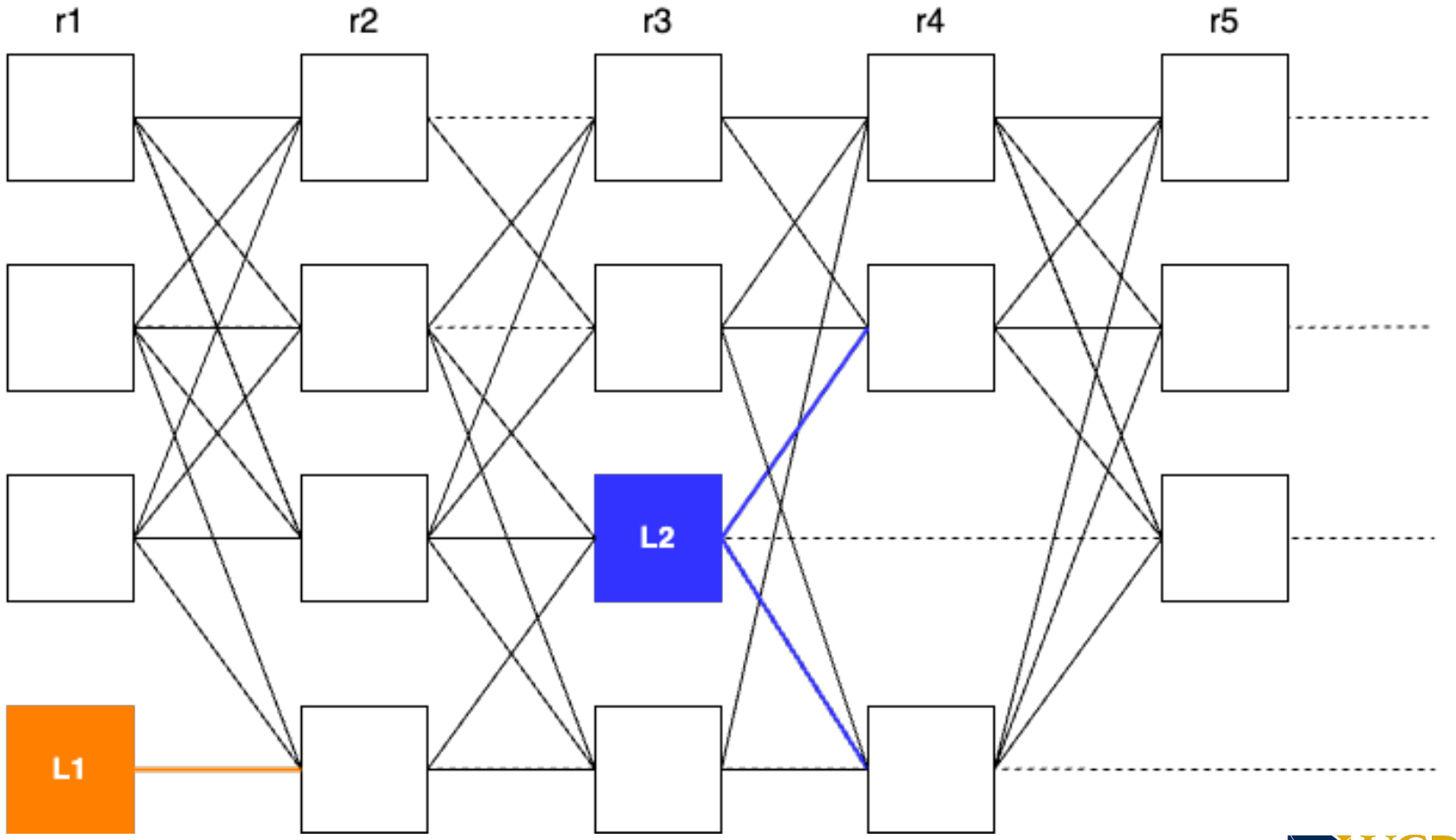
Elect the leader of r3



UCDAVIS COLLEGE OF ENGINEERING

Tusk

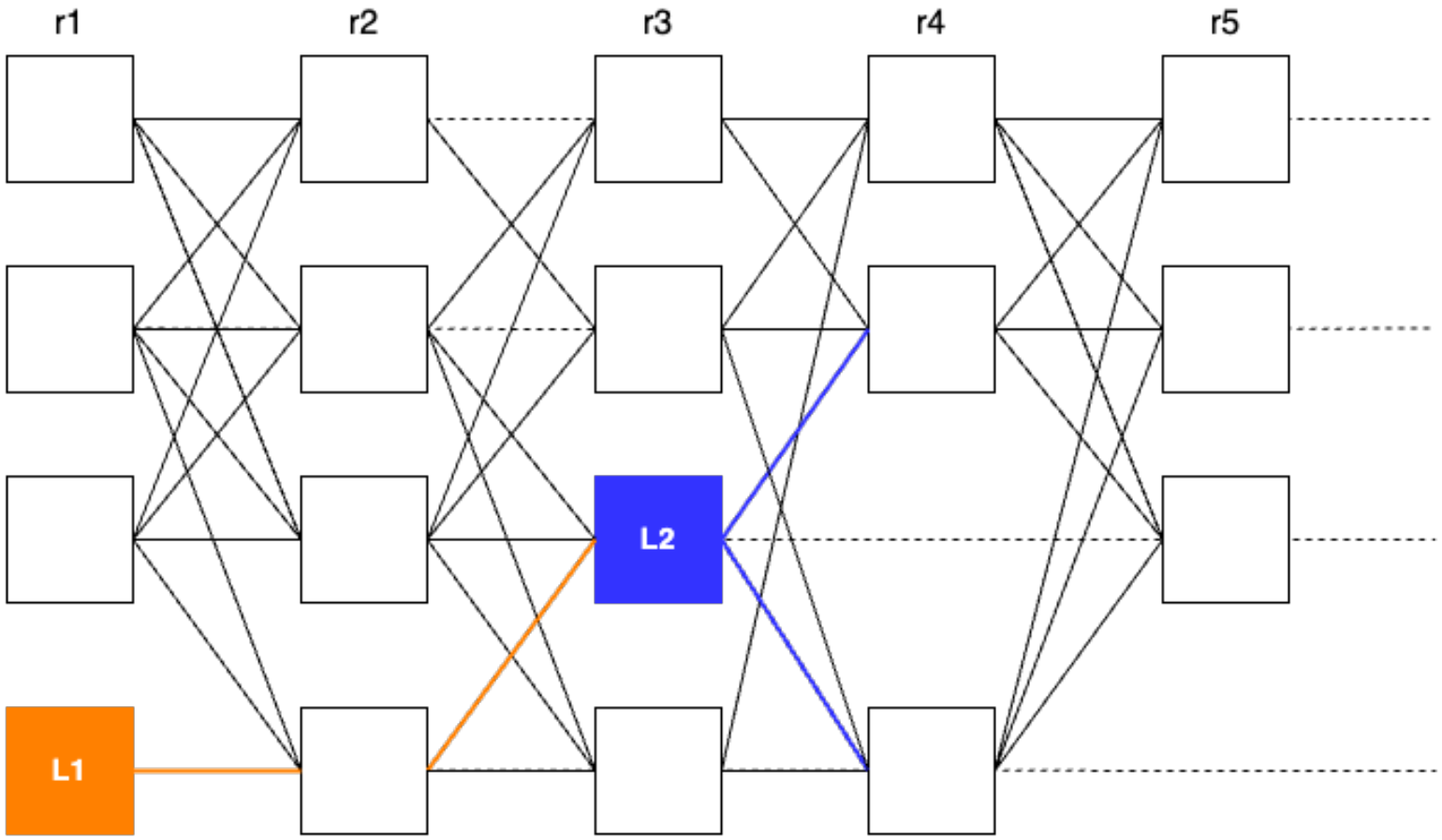
Leader L2 has enough support



UCDAVIS COLLEGE OF ENGINEERING

Tusk

Leader L2 has links to Leader L1



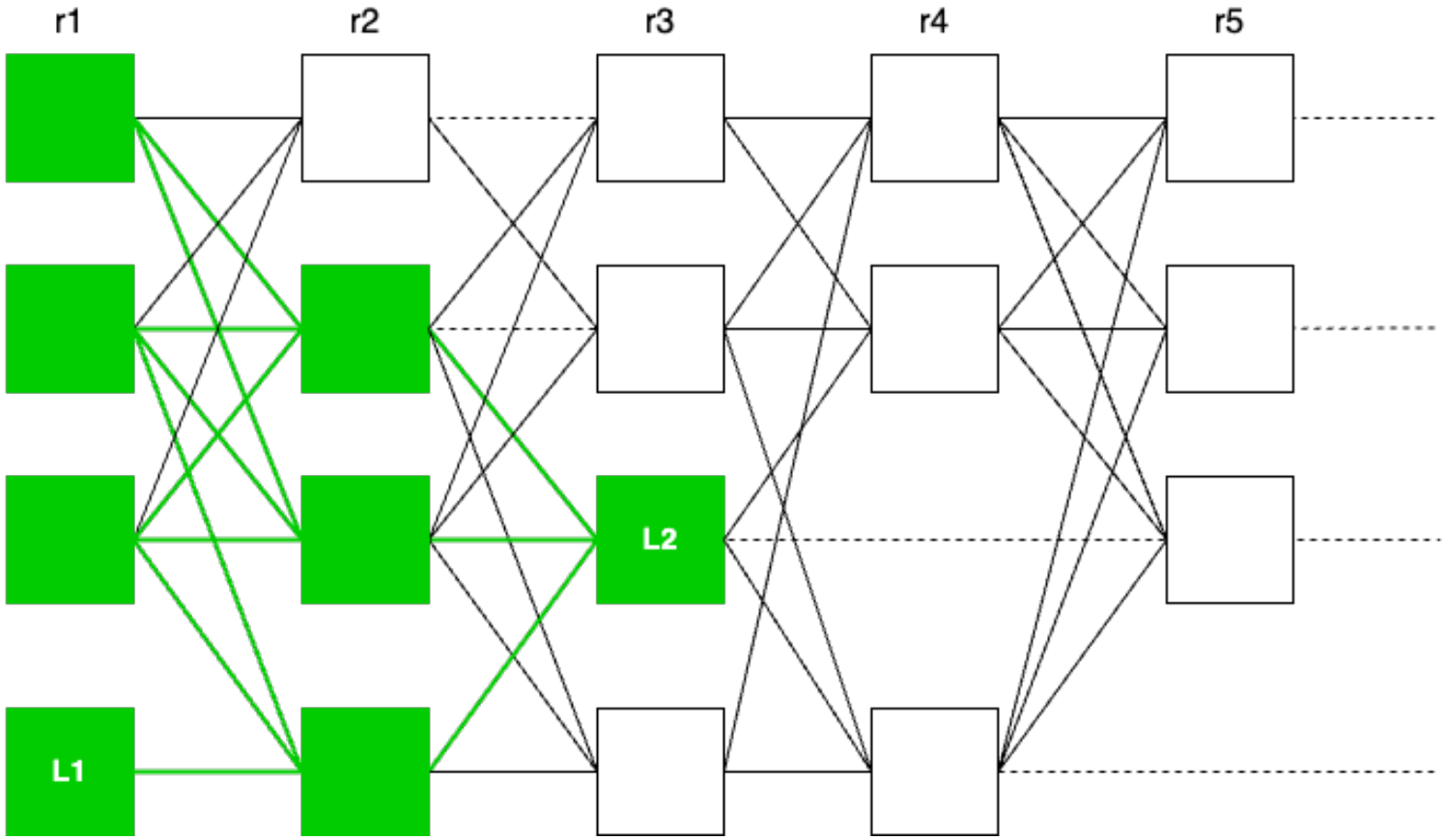
First commit L1

Then commit L2

UCDAVIS COLLEGE OF ENGINEERING

Tusk

Commit all the sub-DAG of the leader



UCDAVIS COLLEGE OF ENGINEERING

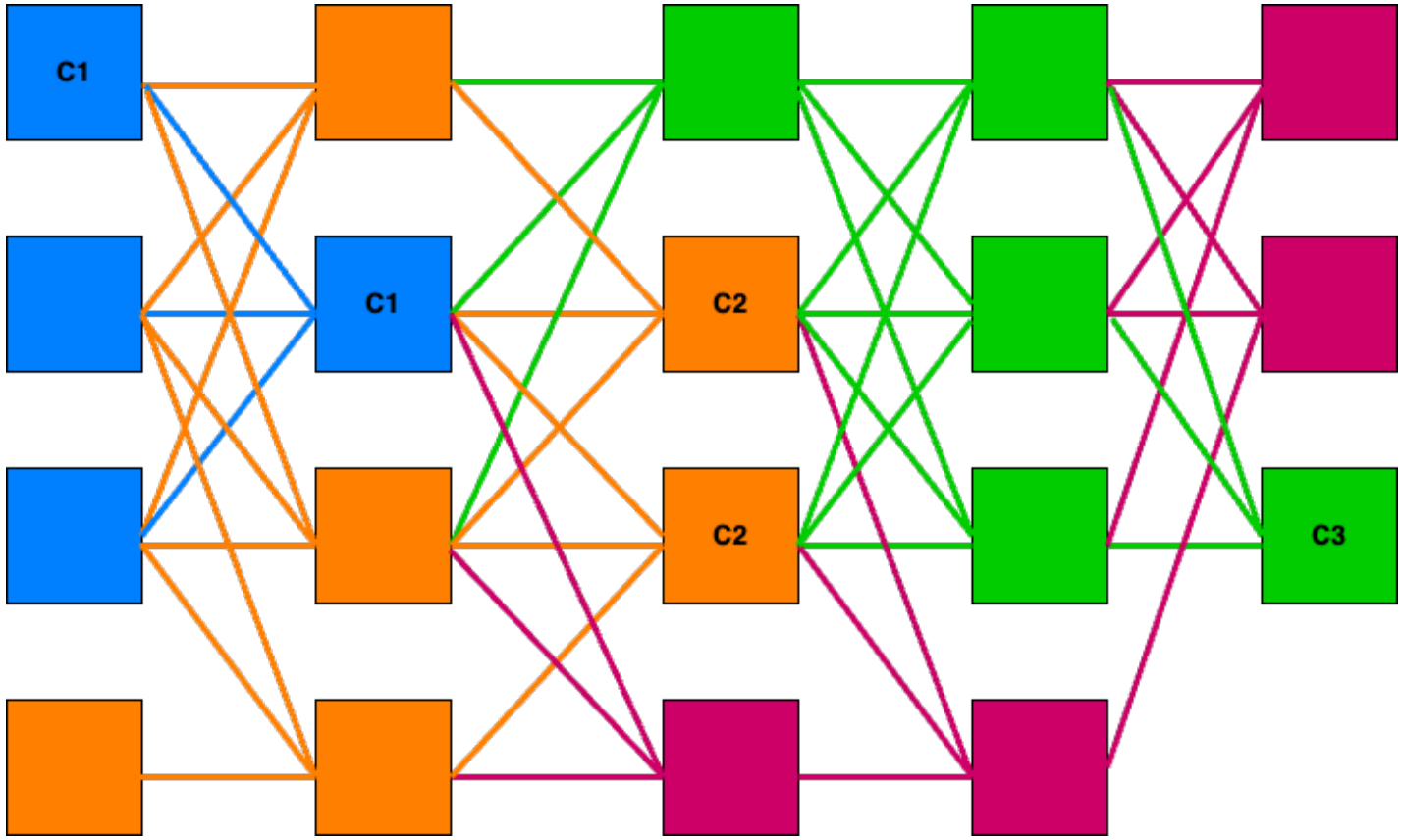
What is the intuition behind **f+1?**

What is the intuition behind **f+1?**

For safety

How to ensure **liveness**?

HotStuff on Narwhal



UCDAVIS
COLLEGE OF ENGINEERING

Implementation and Metrics

- Written in Rust
- Networking: Tokio (TCP)
- Storage: RocksDB
- Cryptography: ed25519-dalek

https://github.com/facebookresearch/narwhal

Evaluation - Experimental setup on AWS

UCDAVIS
COLLEGE OF ENGINEERING

Evaluation - Throughput latency graph

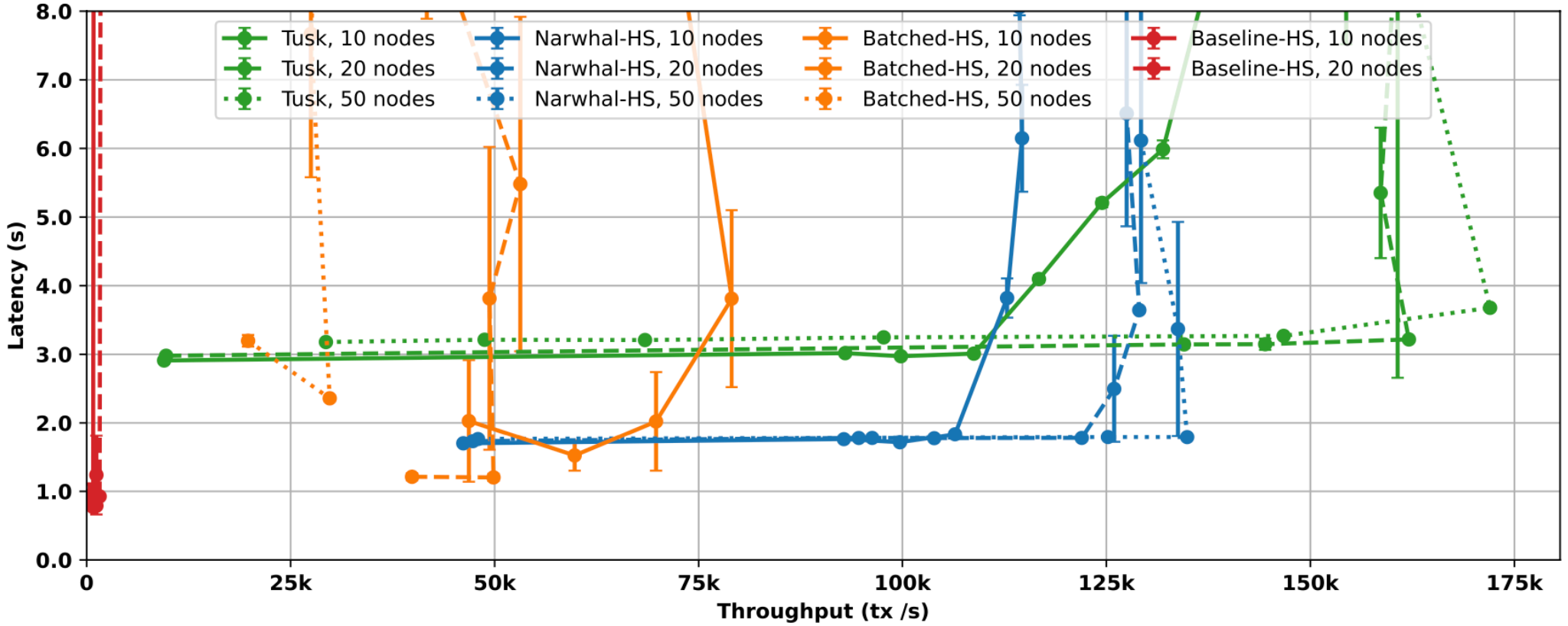
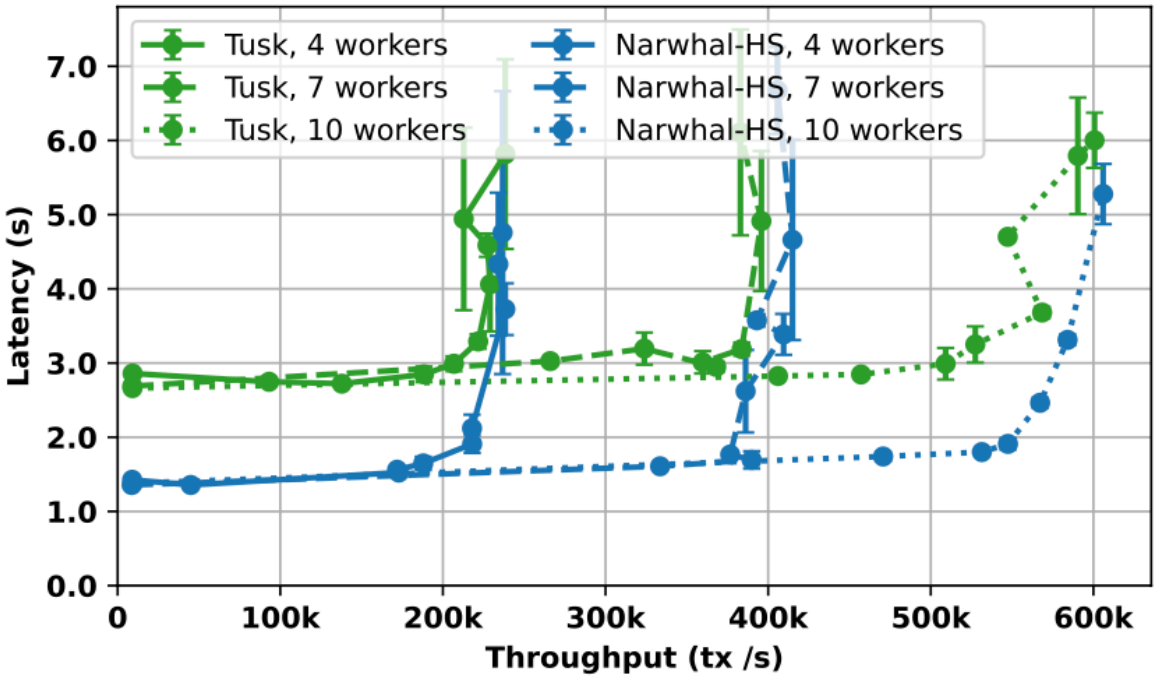
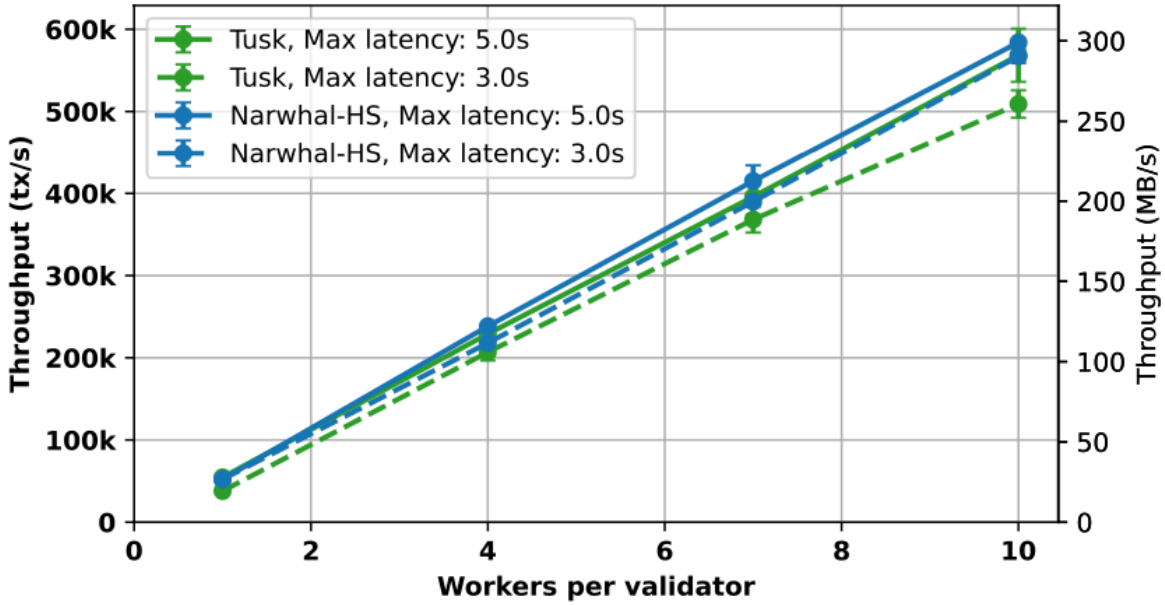


Figure 6. Comparative throughput-latency performance for the novel Narwhal-HotStuff, Tusk, batched-HotStuff and the baseline HotStuff. WAN measurements with 10, 20, and 50 validators, using 1 worker collocated with the primary. No validator faults, 500KB max. block size and 512B transaction size.

UCDAVIS COLLEGE OF ENGINEERING

Evaluation - Scalability





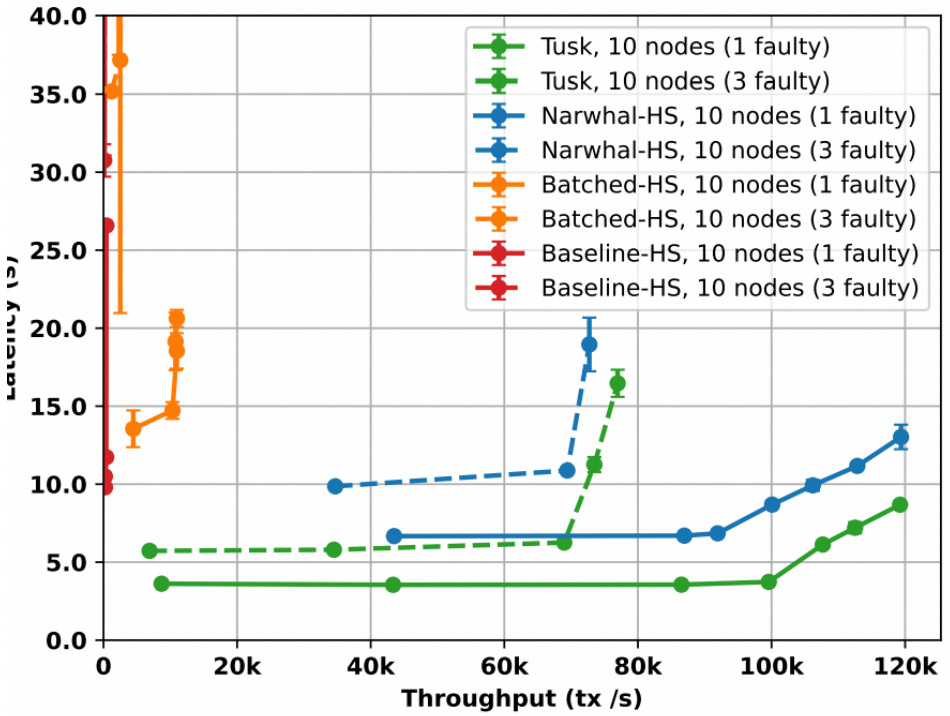
UCDAVIS COLLEGE OF ENGINEERING

Conclusion

NARWHAL AND TUSK

- Separate consensus and data dissemination for high performance.
- Scalable design, egalitarian resource utilizations.

Performance under faults

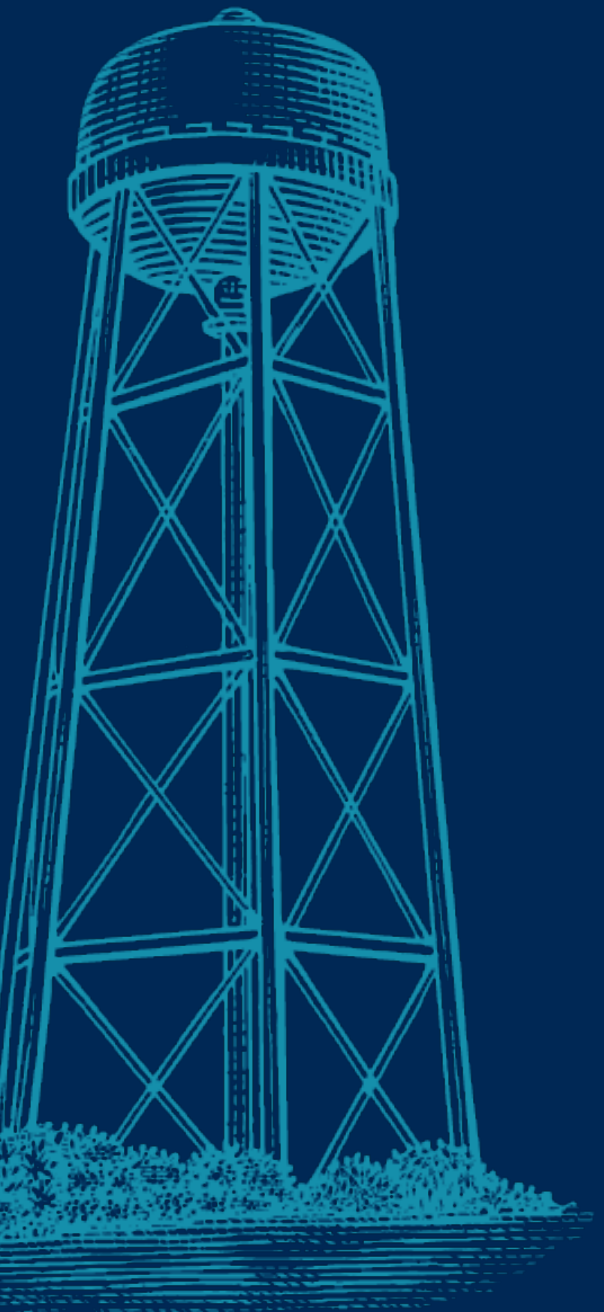


Comparative throughput-latency under faults. WAN measurements with 10 validators using 1 worker co-located with the primary. 1 and 3 faults, 500KB max block size, and 512B transaction size.

UCDAVIS COLLEGE OF ENGINEERING

References

- Danezis, G., Kogias, E. K., Sonnino, & Spiegelman, A. (2021). Narwhal and Tusk: A DAG-based Mempool and Efficient BFT Consensus. *arXiv*. https://doi.org/10.48550/arXiv.2105.11827
- https://decentralizedthoughts.github.io/2022-06-28-DAG-meets-BFT/
- https://research.protocol.ai/sites/consensusday21/programme/
- https://docs.sui.io/learn/architecture/consensus
- https://eprint.iacr.org/2018/992.pdf
- https://salemal.medium.com/a-review-of-narwhal-and-tusk-a-dag-based-mempool-and-efficient-bft-consensus-bf2099908c63
- https://www.blocknative.com/blog/mempool-intro
- https://blog.unibulmerchantservices.com/how-visas-payment-system-works/



THANK YOU!

ANY QUESTIONS? ASK AWAY!

UCDAVIS
COLLEGE OF ENGINEERING