

# Picsou: Enabling Replicated State Machines to Communicate Efficiently



Reginald Frank, Micah Murray,

Chawinphat Tankuranand, Junseo Yoo, Ethan Xu,

Natacha Crooks, Suyash Gupta<sup>†</sup>, Manos Kapritsos<sup>\*</sup>

University of California, Berkeley; †University of Oregon; \*University of Michigan

# Picsou: Enabling Replicated State Machines to Communicate Efficiently



Reginald Frank, Micah Murray,

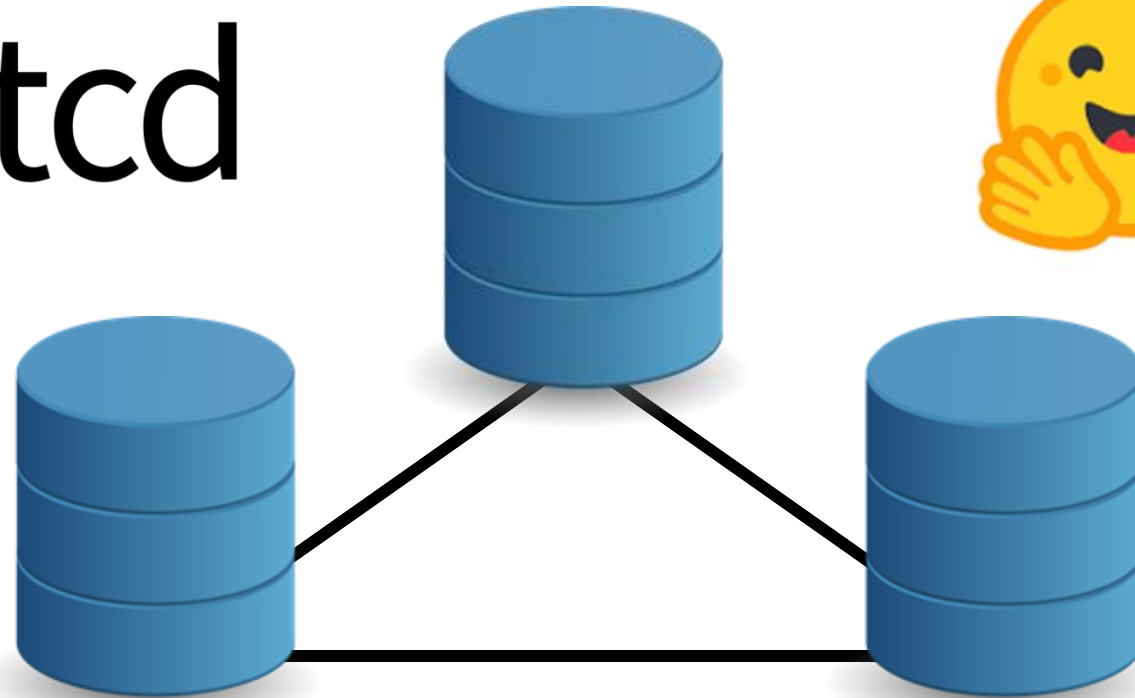
Chawinphat Tankuranand, Junseo Yoo, Ethan Xu,

Natacha Crooks, Suyash Gupta<sup>†</sup>, Manos Kapritsos<sup>\*</sup>

University of California, Berkeley; †University of Oregon; \*University of Michigan

# Distributed Systems and RSMs

 etcd

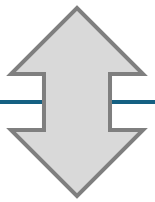


**Reliable  
Storage + Compute**

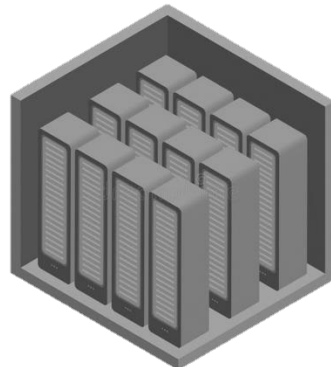
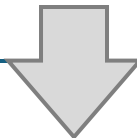
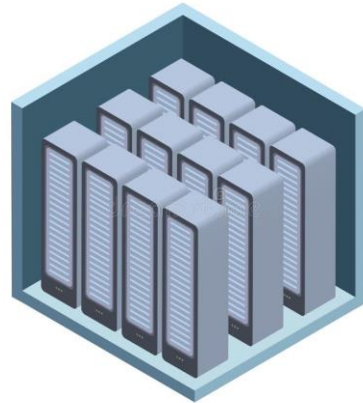


# RSMs Often Communicate

Cross-Org Collaboration



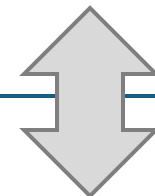
Disaster Recovery



Blockchain Bridges



ethereum



**Algorand**

Problem: There is no formal, scalable way for RSMs to communicate

---

*Existing solutions are ad-hoc, inefficient, or rely on trusted third parties*

What is required for reliable and efficient cross-RSM communication?

# Reliable Communication Requirements

1. Precise Guarantees
2. Low Overhead in Common Case
3. Robustness Under Failures
4. Support General RSM Communication
5. No Trusted Third Party

# Reliable Communication Requirements

## Cross-Cluster Consistent Broadcast (C3B)

## Picsou

1. Strong Guarantees

2. Low Overhead in Common Case

3. Robustness Under Failures

4. Support General RSM Communication

5. No Trusted Third Party

# Reliable Communication Requirements

## Picsou

2. Low Overhead in Common Case

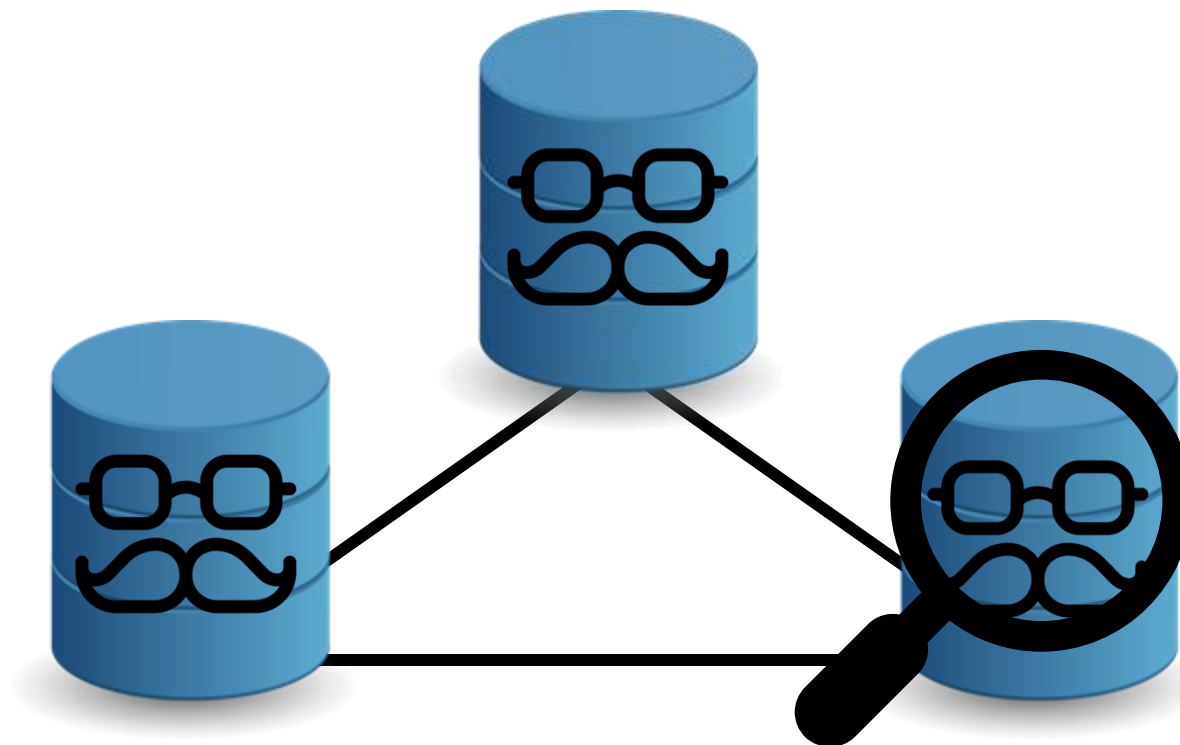
3. Robustness Under Failures

4. Support General RSM Communication

5. No Trusted Third Party



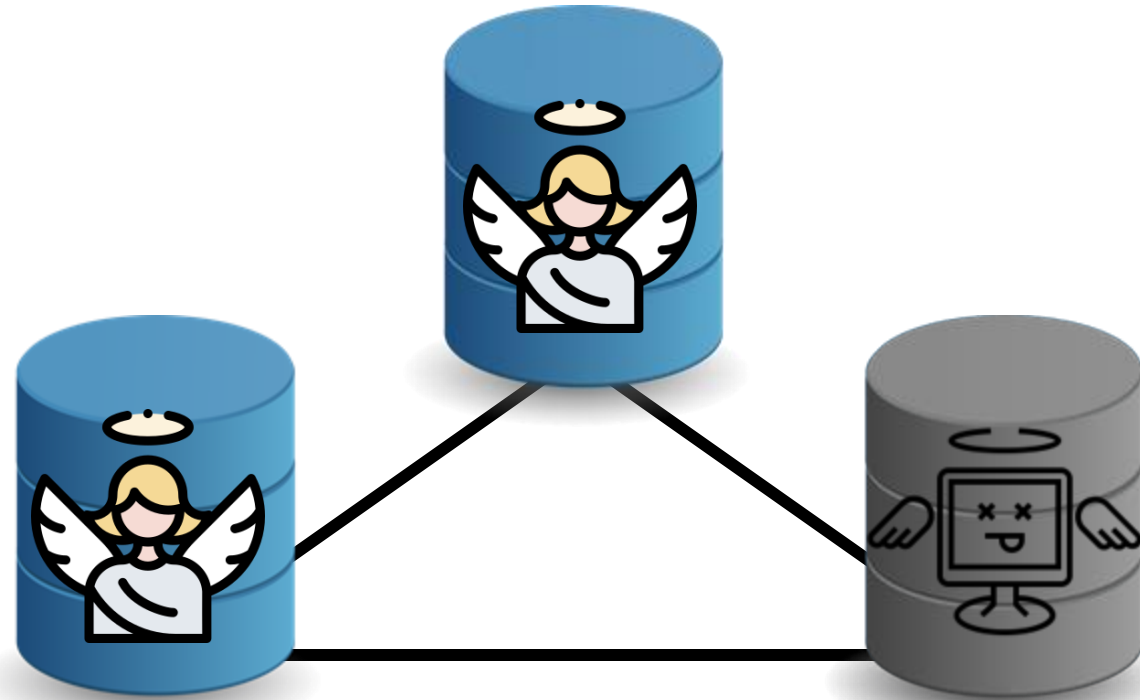
# Replicated State Machine Overview



$$N = 3$$

$$f = 1$$

# Replicated State Machine Overview

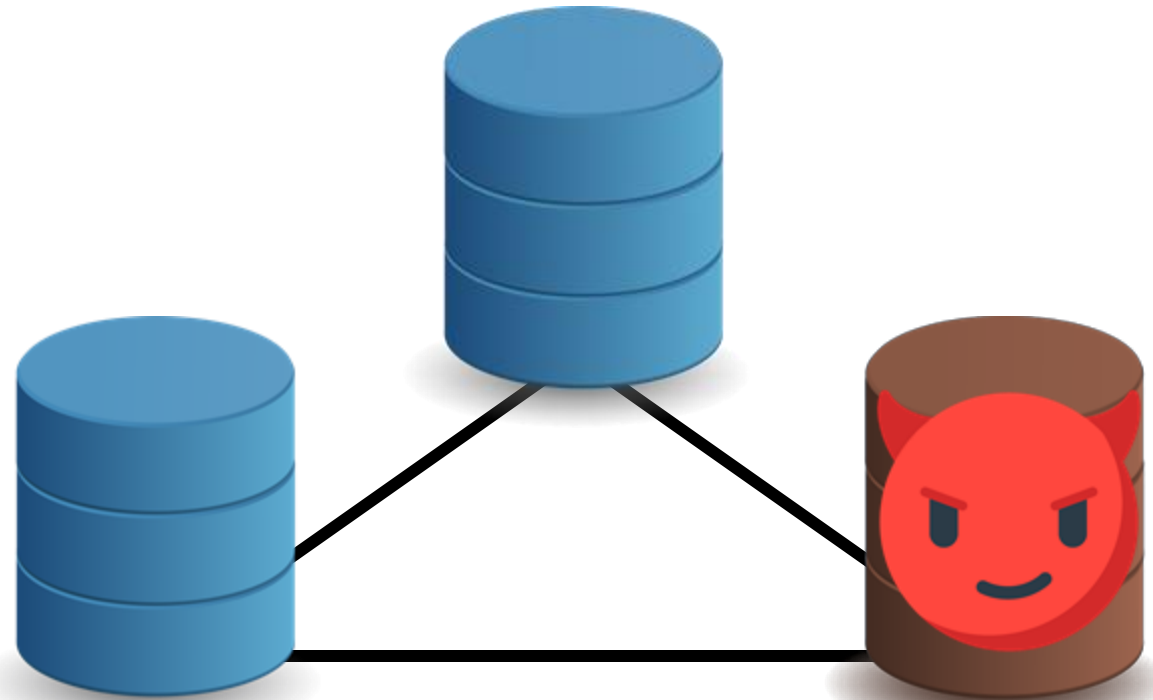


$$N = 3$$

$$f = 1$$

**Crash Fault Tolerant**

# Replicated State Machine Overview



$$N = 3$$

$$f = 1$$

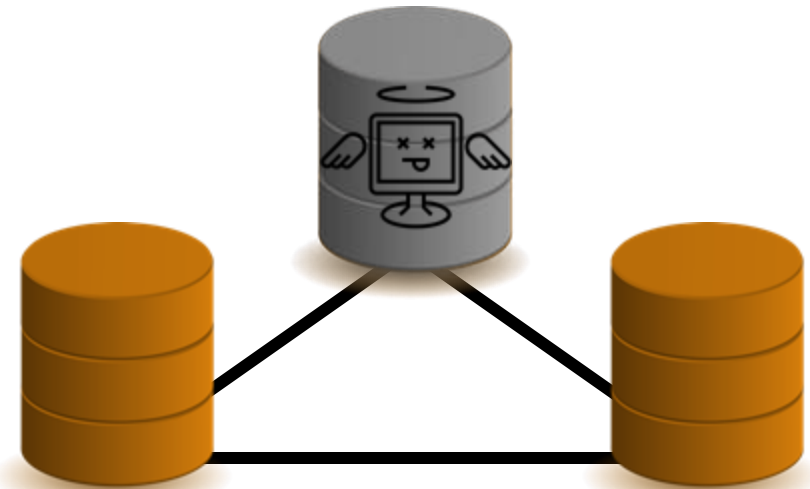
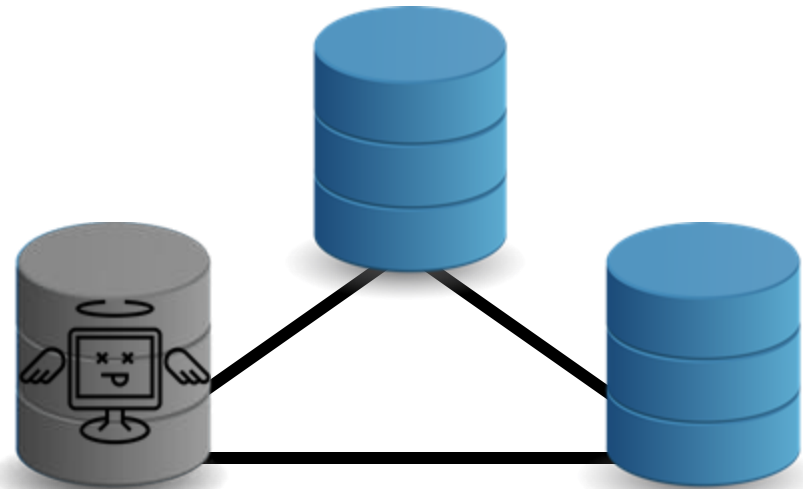
**Byzantine Fault Tolerant**

# Communication between RSMs



# Communication between RSMs

 etcd



# Cross-Cluster Consistent Broadcast (C3B)

***Transmit.***

***Deliver.***

**Eventual Delivery**

**Integrity**

# Cross-Cluster Consistent Broadcast (C3B)

*Deliver.*

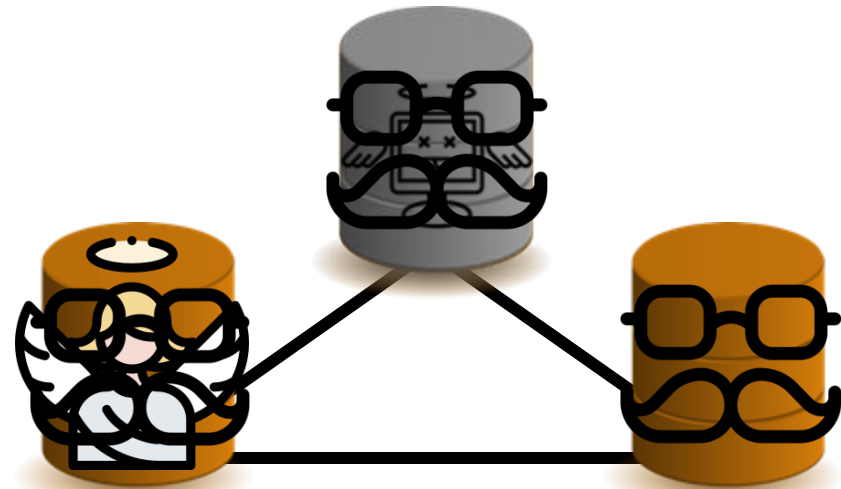
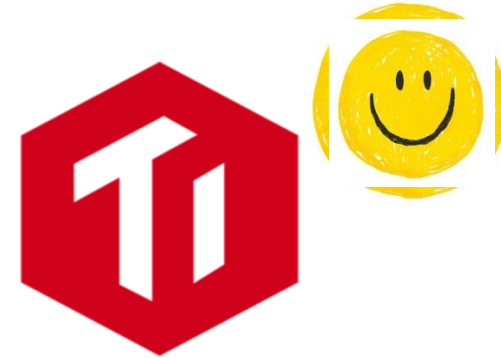
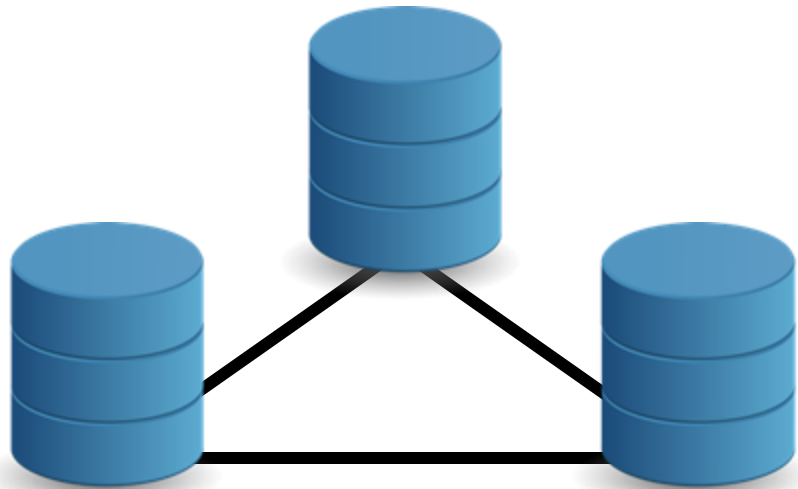
If a **correct** replica in RSM  $\mathcal{R}_r$  outputs message  $m$ ,

then we say RSM  $\mathcal{R}_r$

Delivers  $m$  from RSM  $\mathcal{R}_s$ .

# Communication between RSMs

 etcd

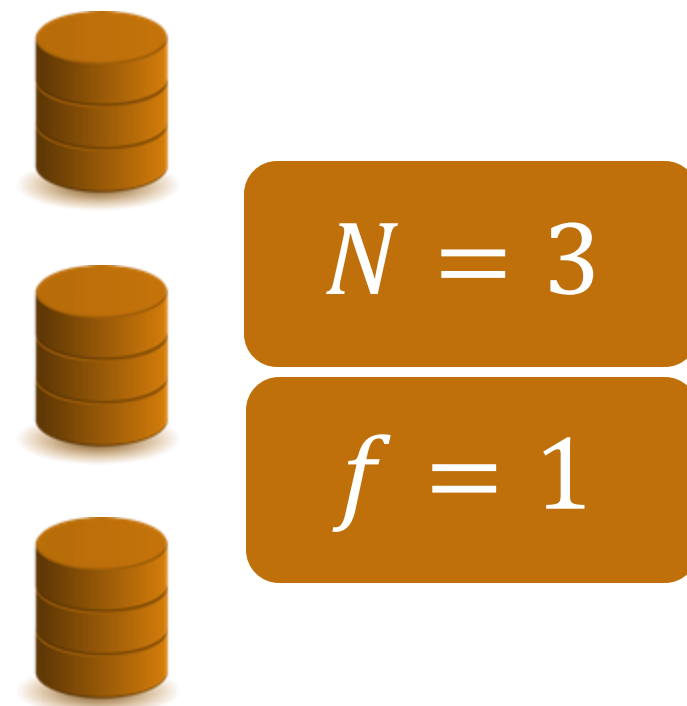
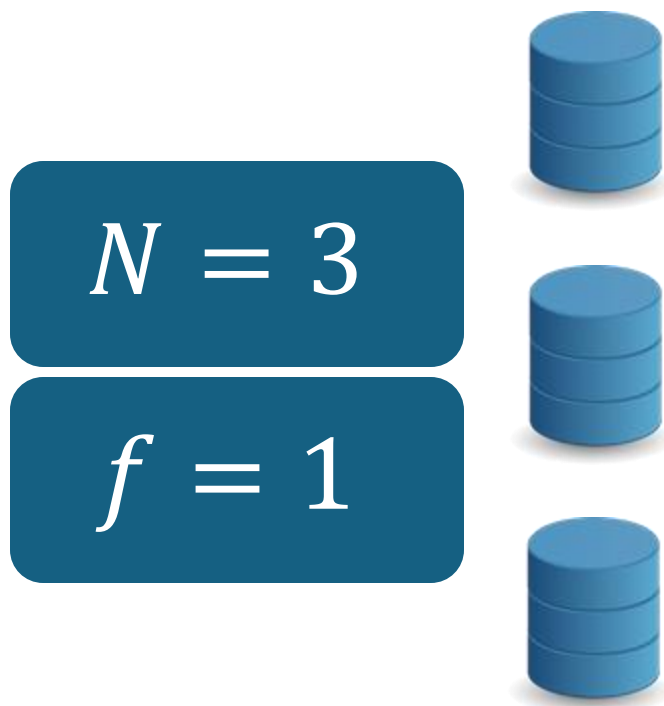


# One to One Communication

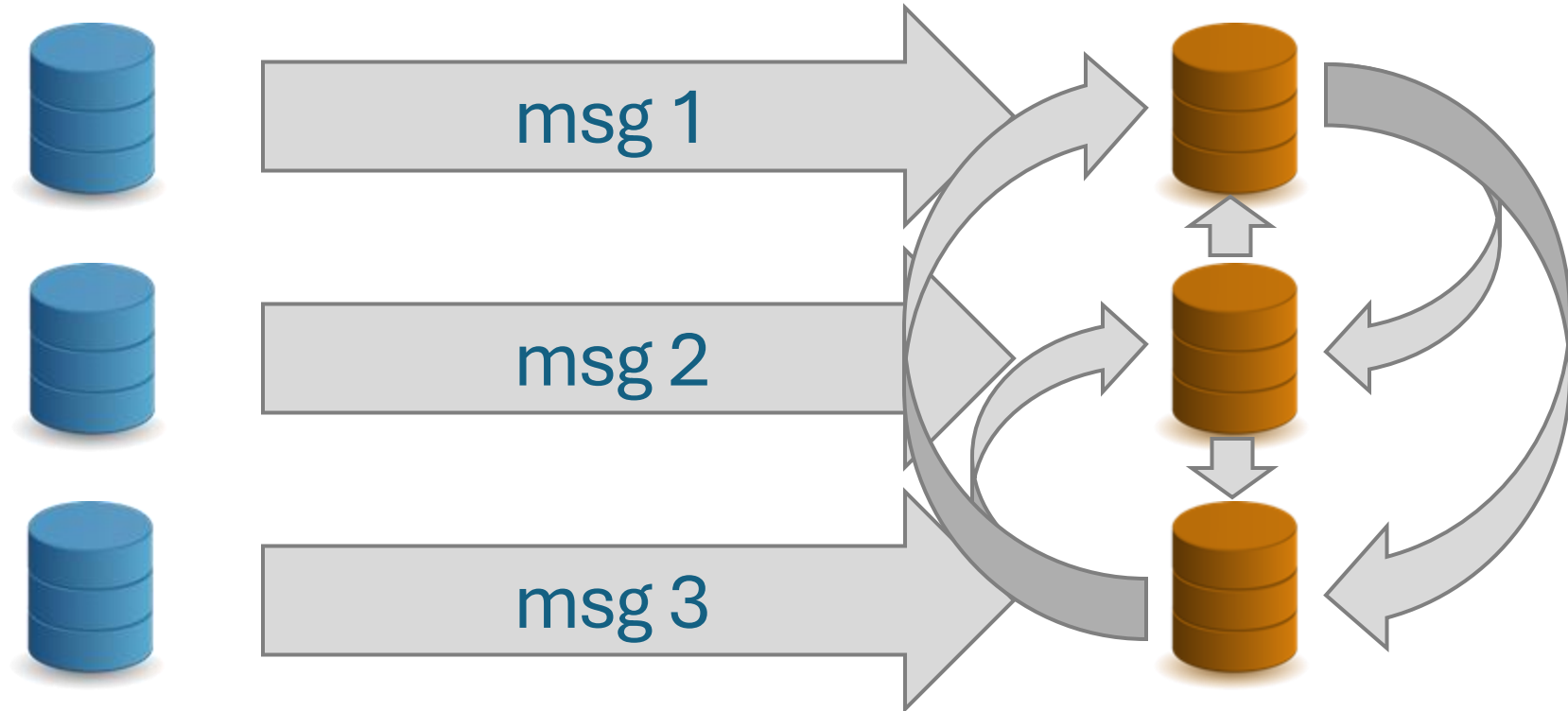


Cumulative Acknowledgements

# General Communication

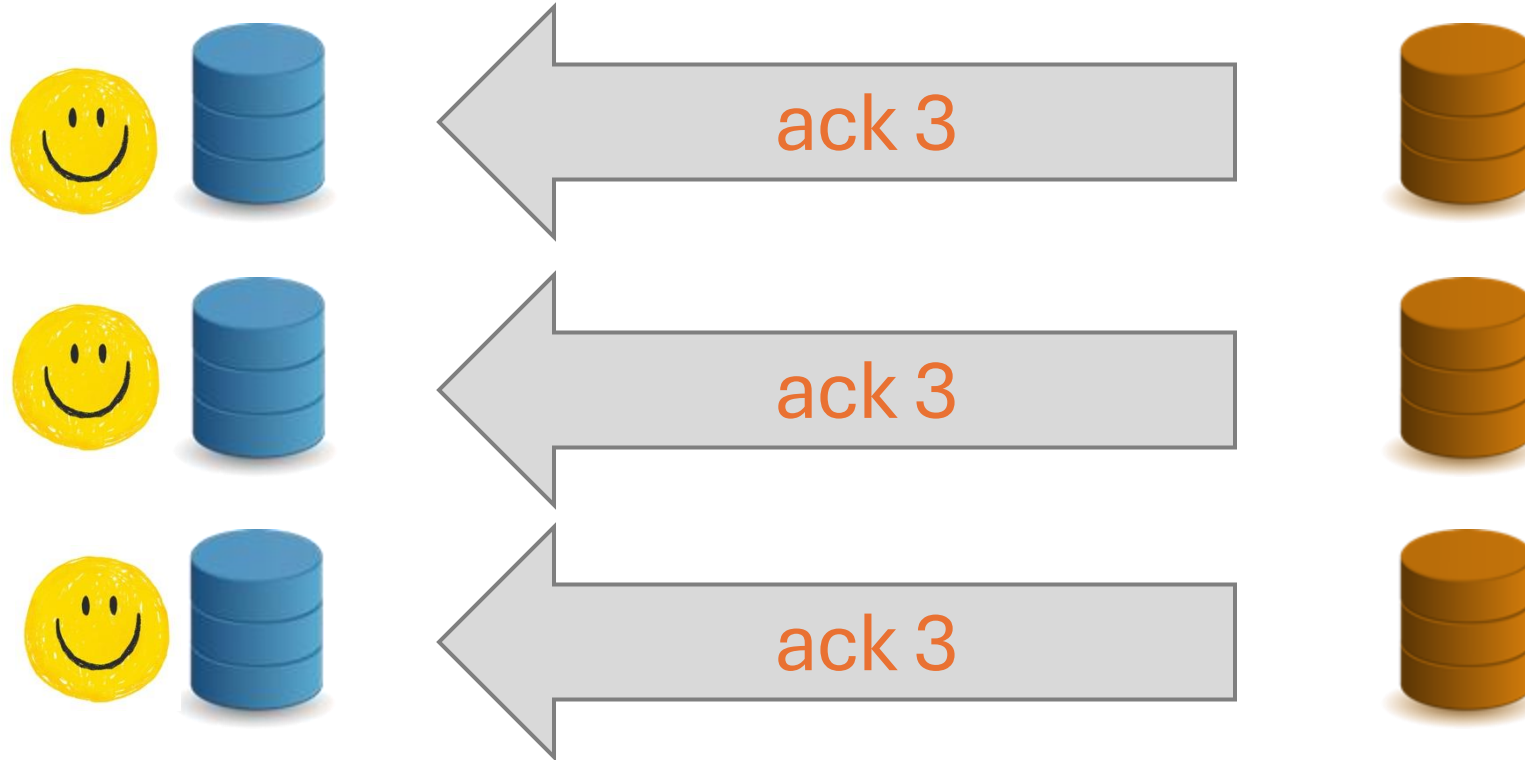


# General Communication

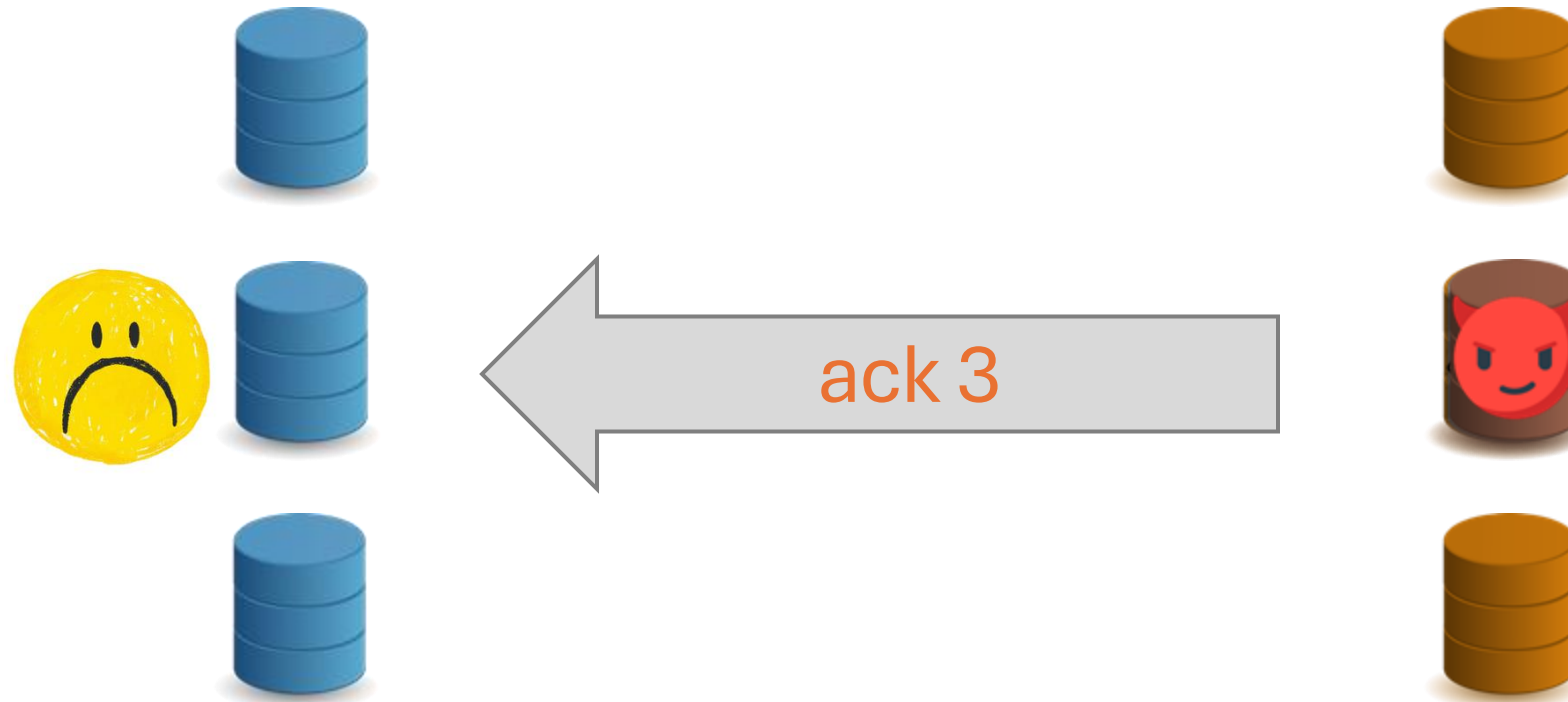


Round Robin Scheduling

# General Communication



# Delivered or Dropped?



***Detect Reliable Delivery***

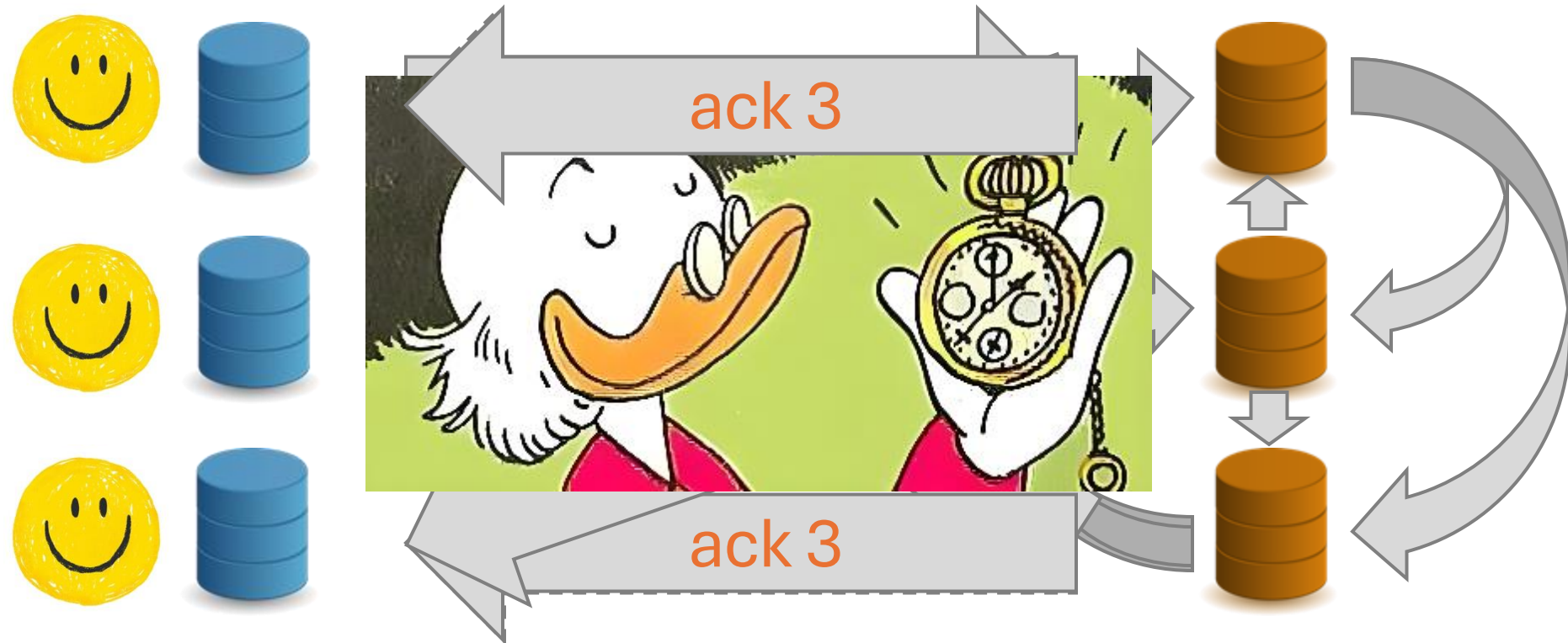
***No Additional Messages***

# Detecting Reliable Delivery



Quorum Acknowledgments (Quacks)

# Generating Quorum Acknowledgments



Receiver Rotation + Quacks

# Problems Solved

---

Good Case Message Scheduling

---

Using Quacks to track message delivery

---

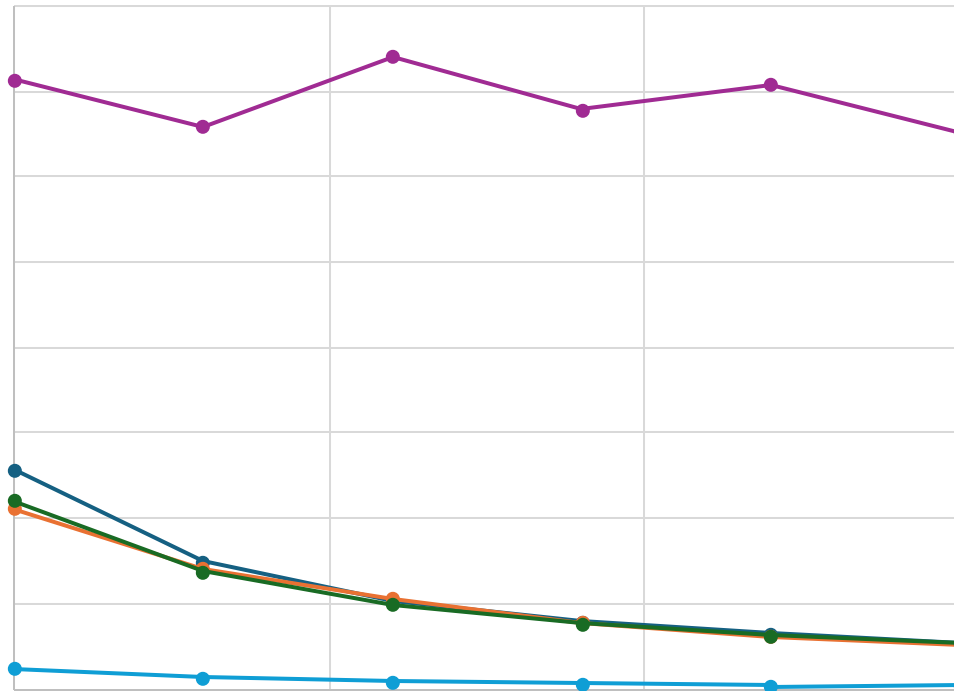
Tracking delivery of messages in parallel

---

Proof of Stake Systems

# Alternative C3B Strategies

# Strategy Throughput -- 1 MB Messages



# Picsou: Enabling RSM Communication!

Applications need Cross-RSM Communication

C3B formalizes reliable Cross-RSM Communication

Picsou implements C3B efficiently and reliably



reginaldfrank77@berkeley.edu