Rosé: Flexible Replication With Strong Semantics For Partitioned Databases

Ioannis Zarkadas, Kelly Kostopoulou, Thomas Graham,
Junfeng Yang, Philip A. Bernstein, Asaf Cidon, Tamer Eldeeb

Columbia University, Microsoft Research

CIDR 2026

The wine series

Chardonnay

OSDI 2023

Fast single-region transactions with linearizable queries

Chablis

CIDR 2024

Fast cross-region transactions with linearizable queries

Rose

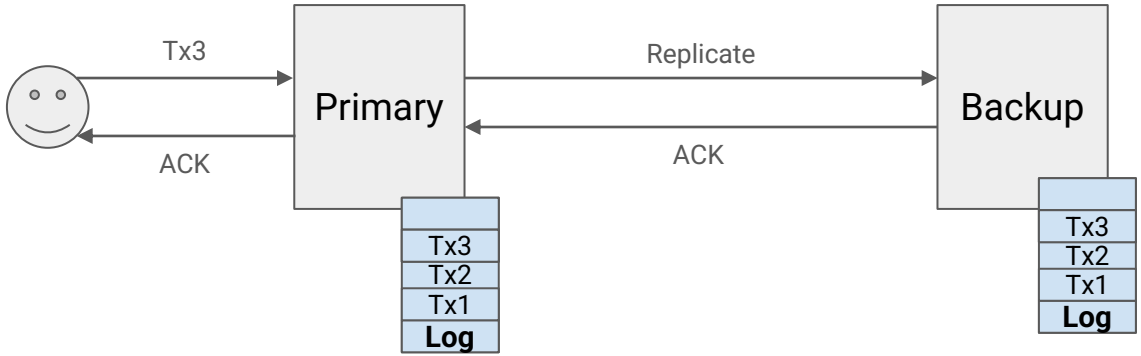
CIDR 2026

Better async replication

Replication is Ubiquitous

- Durability
- Availability
- Read Latency

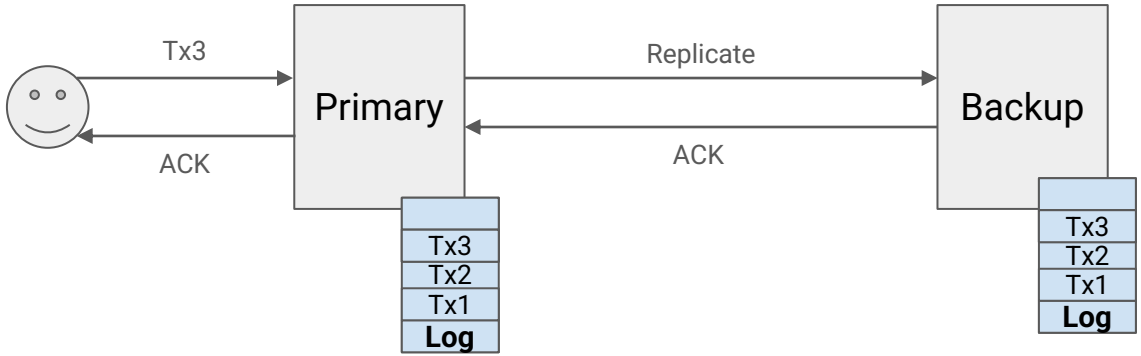
Synchronous Replication



High Durability, Linearizability, Fast Failover

Good fit for single-region

Asynchronous Replication



Low write latency, high availability

Good fit for cross-region

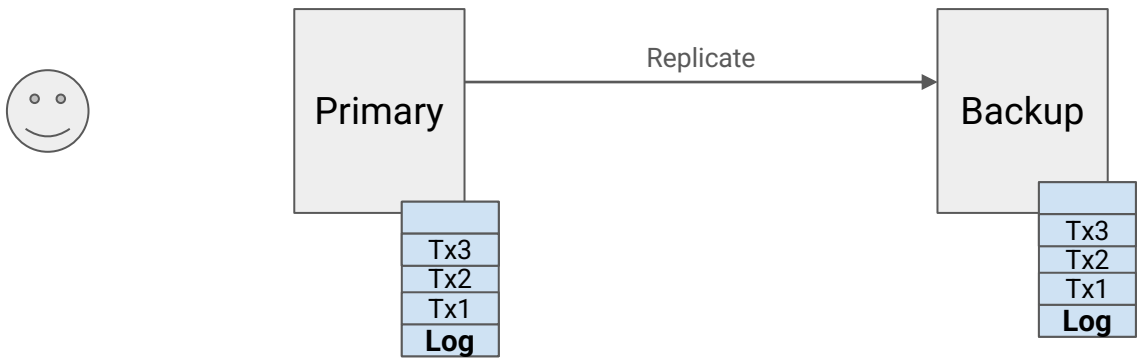
Async Replication is hard to work with

Challenge 1: Eventual consistency = no guarantees.

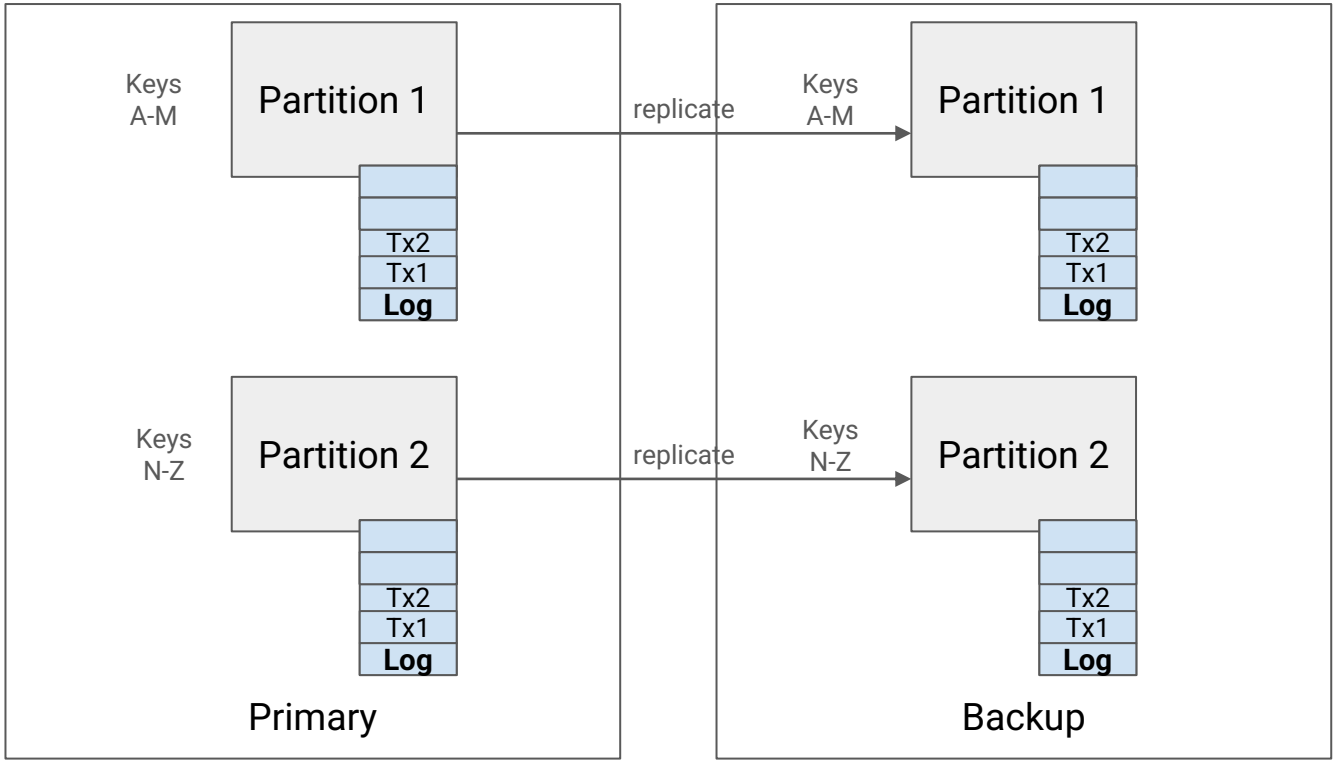
Challenge 2: Replication lag.

Challenge 3: Degraded performance after recovery.

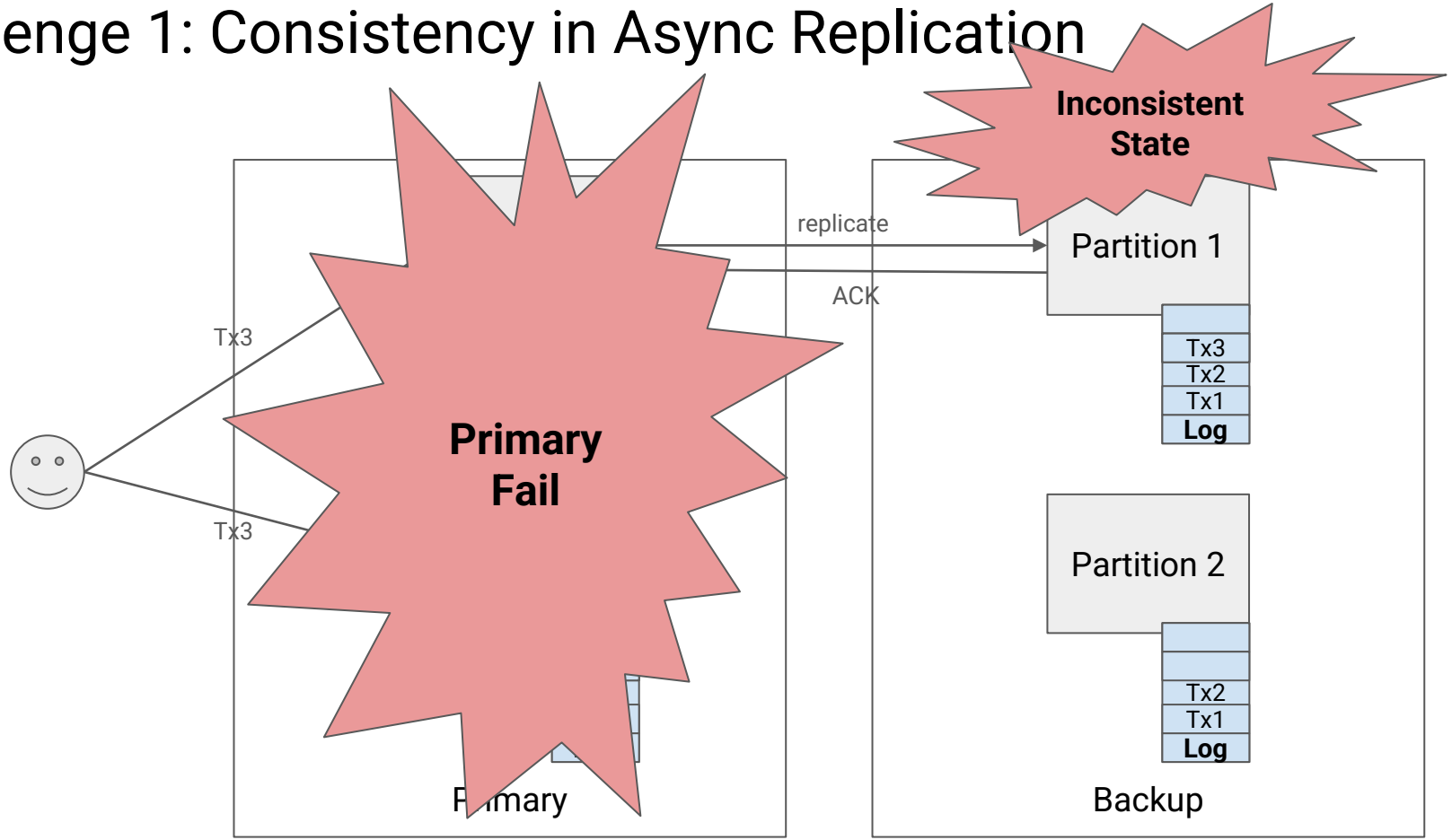
Our model database



Our model database



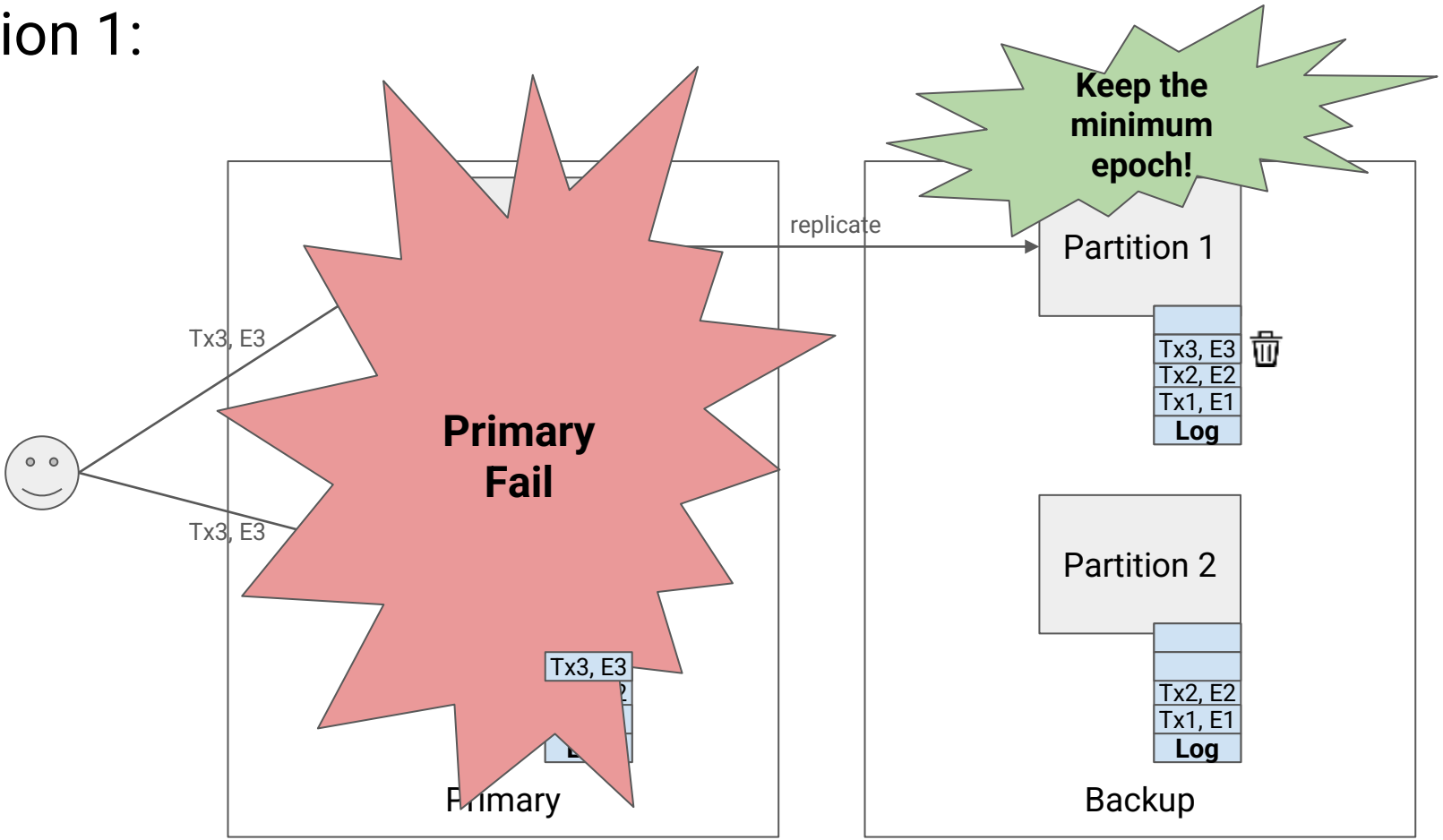
Challenge 1: Consistency in Async Replication



Solution 1: Reuse Snapshotting Mechanisms

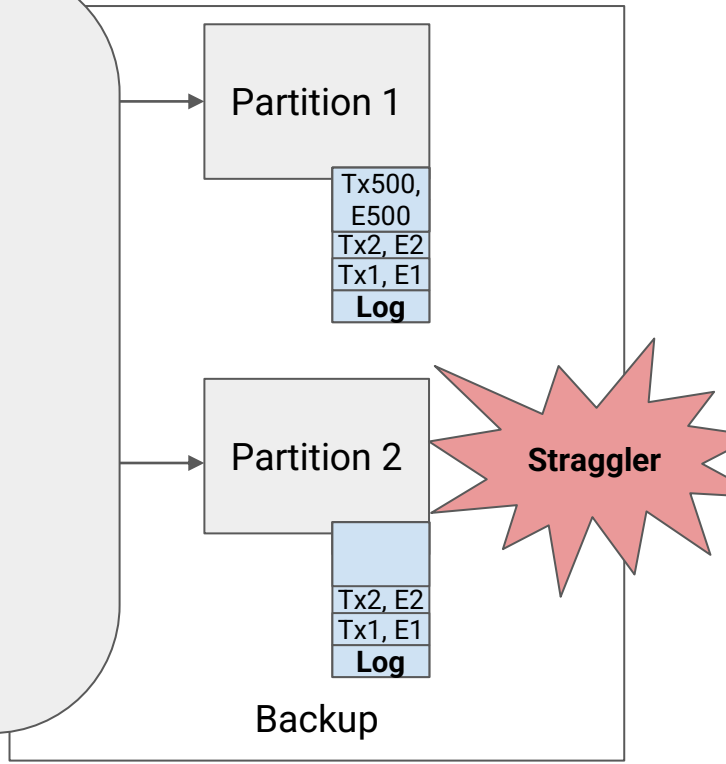
- Monotonic Prefix Consistency = Best possible for async
- Modern distributed databases often support global read-only snapshots
  - Can be based on time (Spanner), epochs (Chablis), HLC (Yugabyte), …
- Reuse snapshotting mechanism for replication

Solution 1:

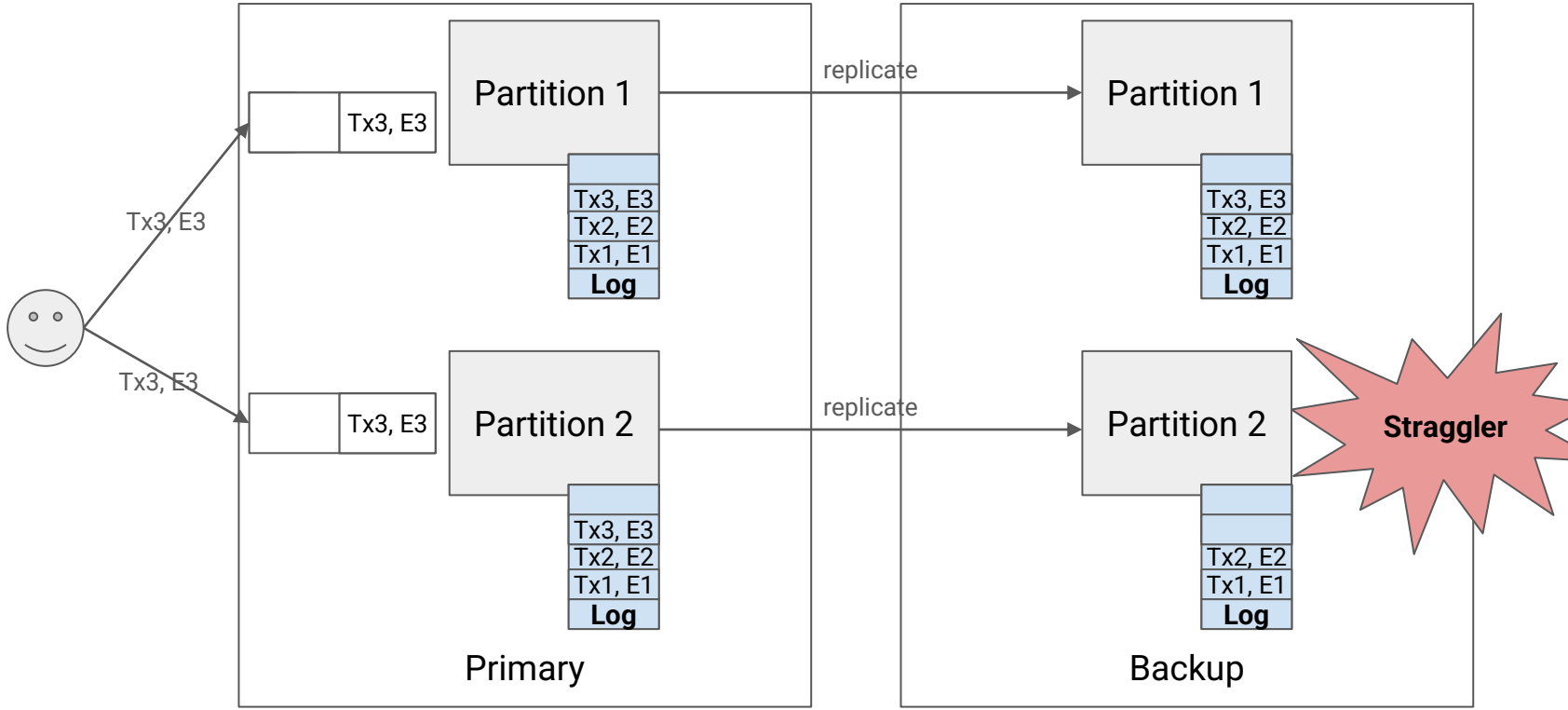

Keep the minimum epoch!
replicate
Partition 1
Tx3, E3
Tx2, E2
Tx1, E1
Log
Tx3, E3
Tx3, E3
Primary Fail
Partition 2
Tx2, E2
Tx1, E1
Log
Tx3, E3
Primary
Backup

Challenge 2: Stragglers are detrimental

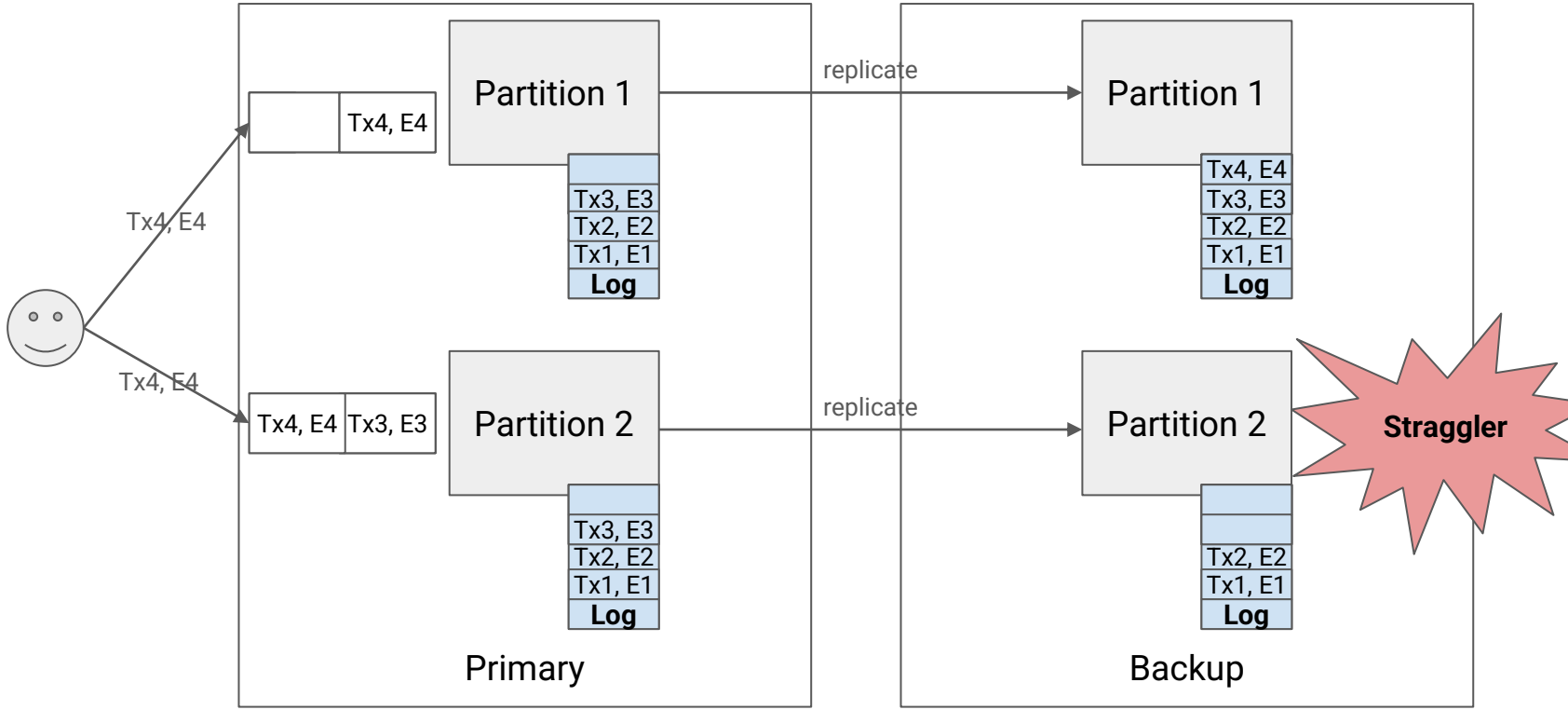
- Durability suffers. If primary fails, backup is consistent only up to epoch 2.
- Stale reads (primary at epoch 500, backup at epoch 2).



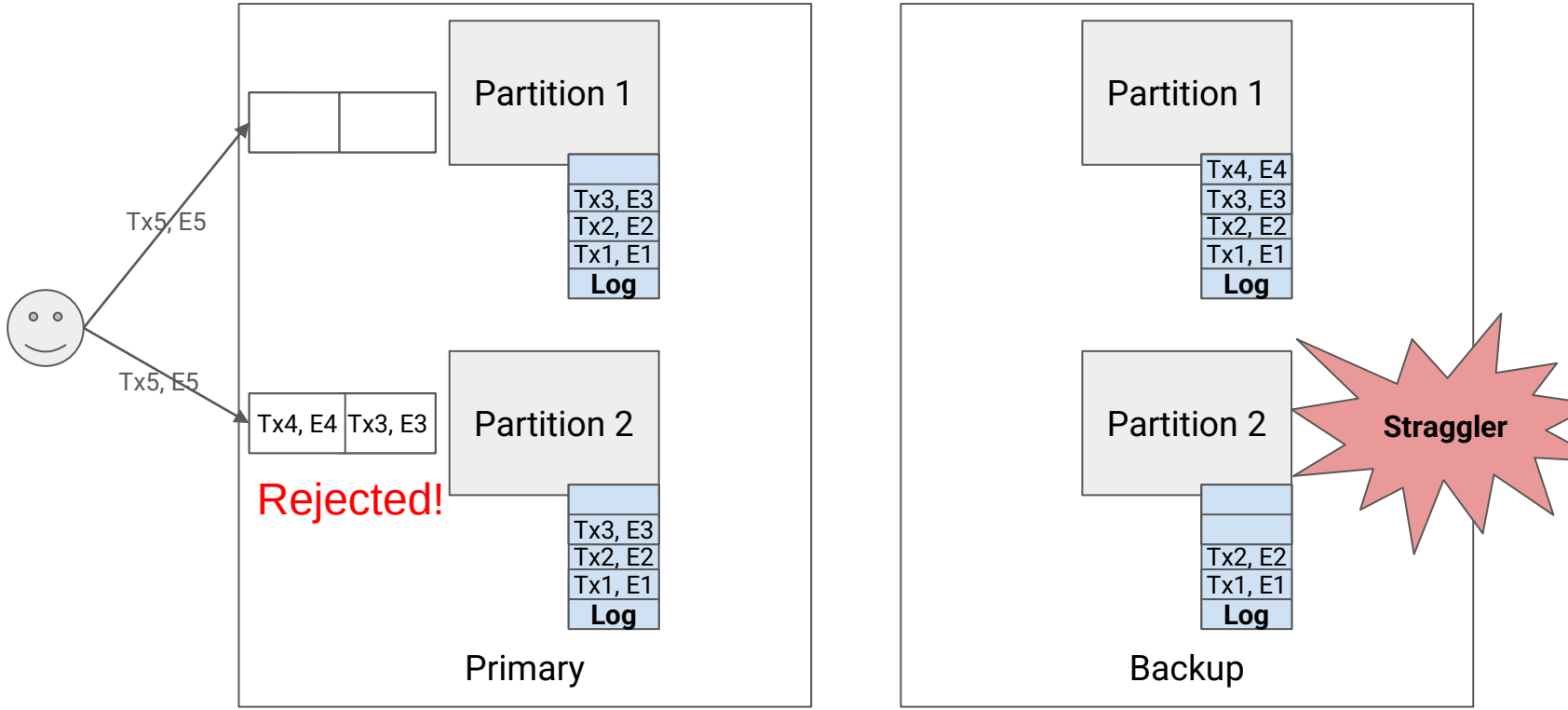
Solution 2: Backpressure to avoid stragglers



Solution 2: Backpressure to avoid stragglers



Solution 2: Backpressure to avoid stragglers

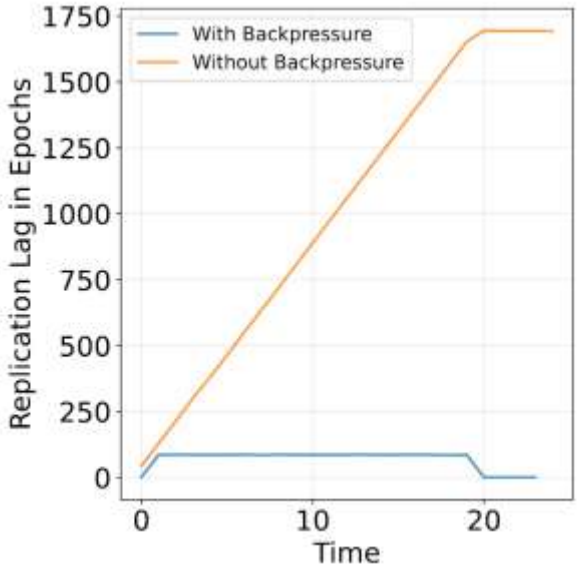


What about availability?

- At worst as much as synchronous replication.
- Only straggler partition is affected.
- See paper for more details.

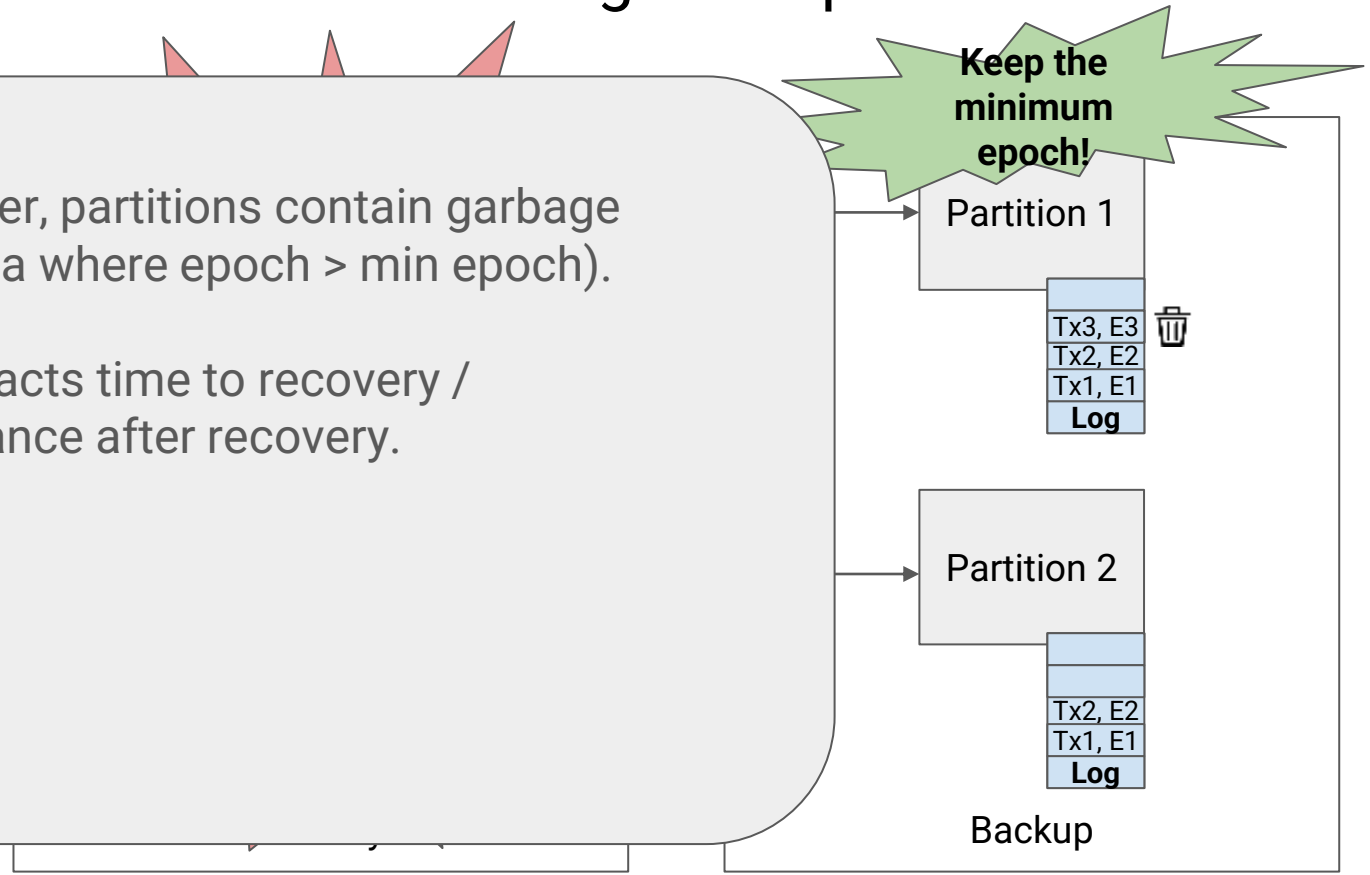
Rosé effectively caps replication lag

- Primary-backup, 2 partitions each.
- Backpressure vs plain asynchronous replication



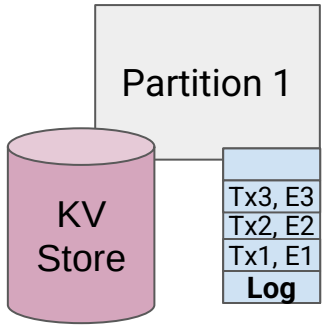
Challenge 3: Failover results in degraded performance

- On failover, partitions contain garbage data (data where epoch > min epoch).
- This impacts time to recovery / performance after recovery.

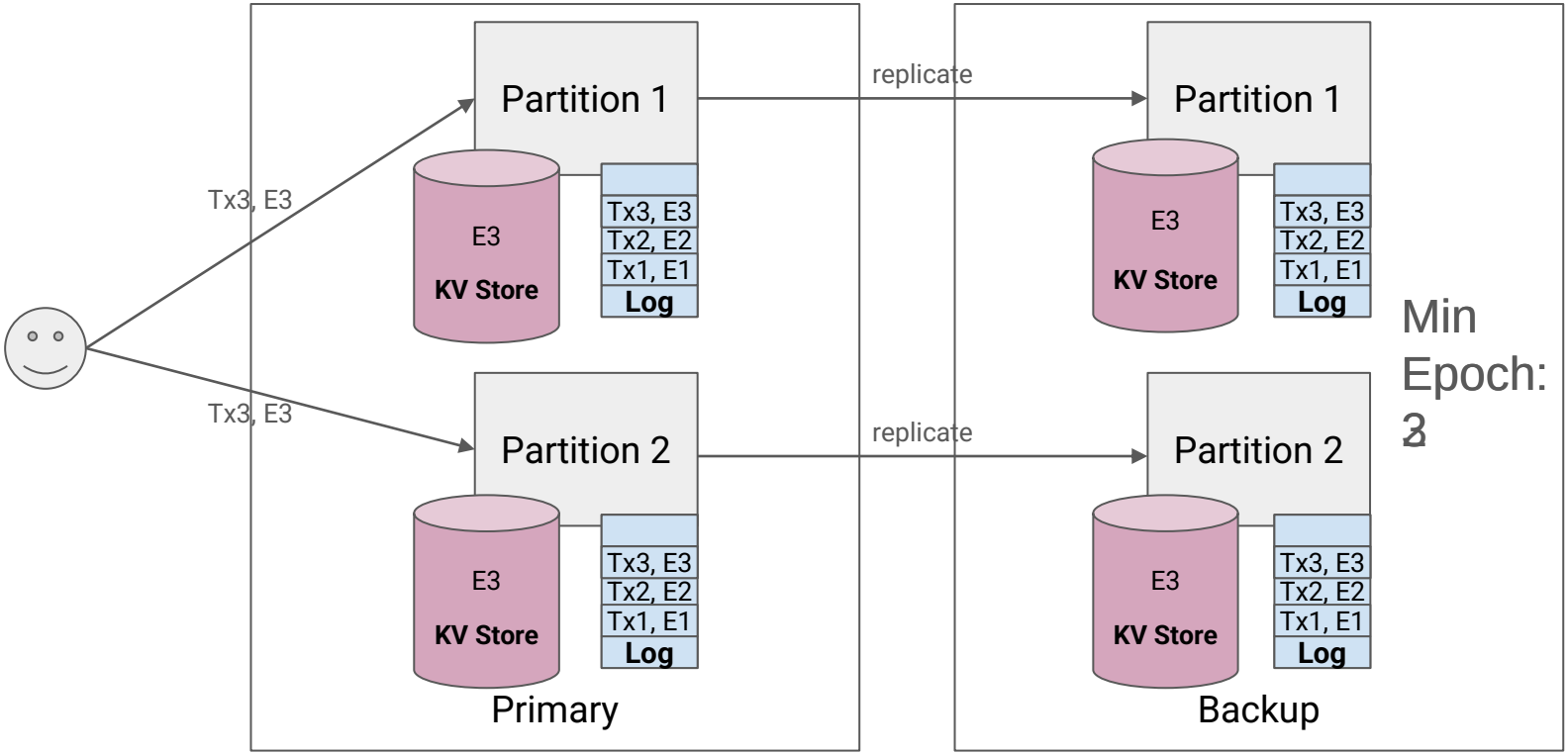


Challenge 3: Failover results in degraded performance

- Partitions persist data in log for durability.
- Then, they write the data in a key-value store for read performance and trim the log.
- Deleting epochs from the log is fast (stored in order).
- Deleting epochs from the key-value store is slow.



Solution 3: Coordinated Apply



Performance after failover remains the same

- Simulate a failover on a skewed workload.
- Measure performance after failover compared to before.
- Chablis with coordinated apply vs Yugabyte xCluster Replication.

Performance After Failover	Yugabyte	Rosé
Throughput Slowdown	22%	0%
P99 Latency Inflation	15%	0%

Rosè balances between fully sync and async

	Write Latency	Consistency	Durability	Availability	Perf after Recovery
Synchronous	High	Strong	High	Low	Same
Asynchronous	Low	Eventual	Low	High	Degraded
Rosè	Low	Monotonic-Prefix	Medium	Medium	Same

Thank you!

Summary:

- Async replication important for cross-region.
- Rosè offers asynchronous replication with strong guarantees.
- More in the paper!

Ioannis Zarkadas
iz2175@columbia.edu