CIDR 26

Event Horizon

Asymmetric Dependencies for Fast Geo-Distributed Operations

Jonathan Arns, Harald Ng, Kyriakos Psarakis,
Asterios Katsifodimos, Paris Carbone

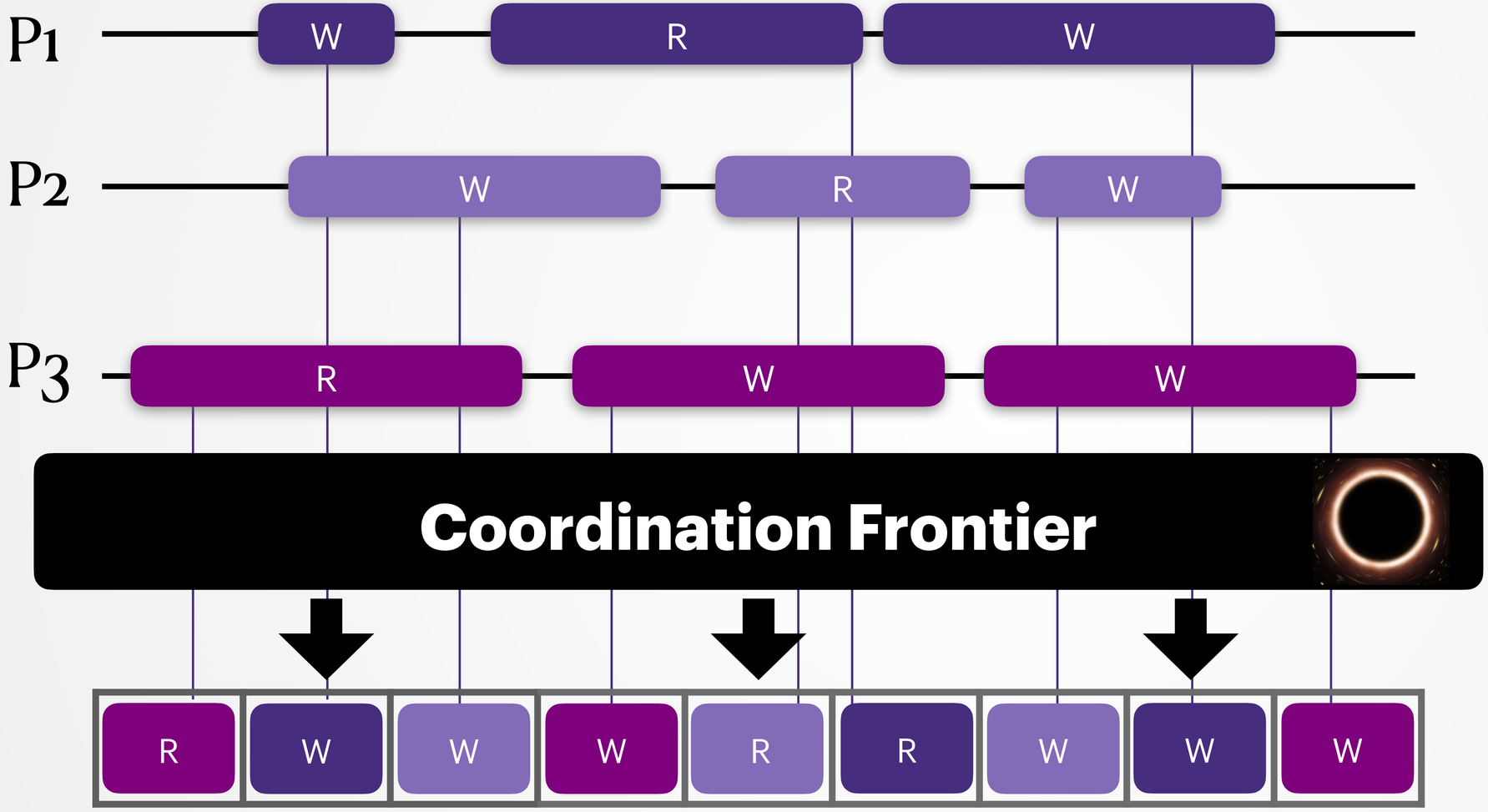
Linearizability ~ Strong Consistency

- ✦ All ops take effect **instantaneously** during their execution. 
- ✦ No stale reads, **no anomalies**

But

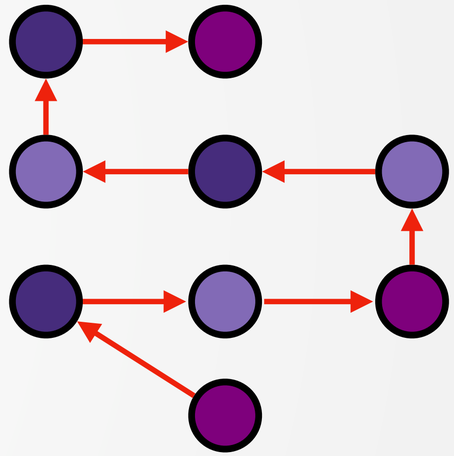
- ✳ **EVERY op** → globally ordered in real time
- ✳ **EVERY replica** → agrees on that order (typically using quorums)
- ✳ **Unavoidable** high latency, limited by speed of light
  - ✳ (~100-300ms in a geo-replicated setting)

What is Linearizability?



Global Total Order (real-time order)

Strong Consistency



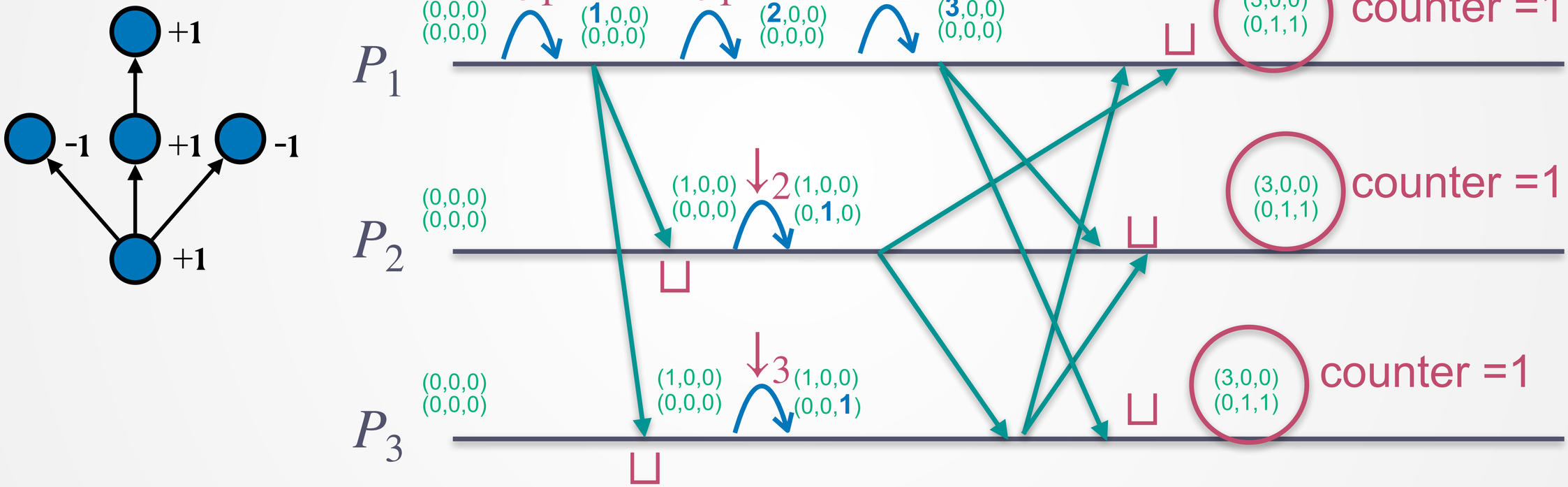
Weak Consistency

- **CALM:** If a computation only adds information, no agreement is necessary

- **SEC:** If two servers p_1, p_2 receive the exact same set of updates, then $p_1 \,.\, state = p_2 \,.\, state$, irrespective of delivery order.

- **CRDTs:** If states/operations of our data types allow commutativity and processes exchange information, eventually they converge to an identical view.

An Eventually Consistent Counter



Weak Consistency

- **NO** global ordering
- **EVERY** possible interleaving → valid state
- Ideal for collaborative editing, carts of items with unlimited stock

But

- Most applications feature **order-dependent** operations
- Cannot enforce **global invariants** without agreement
  - i.e., single auction winner, object ownership, no double-spending

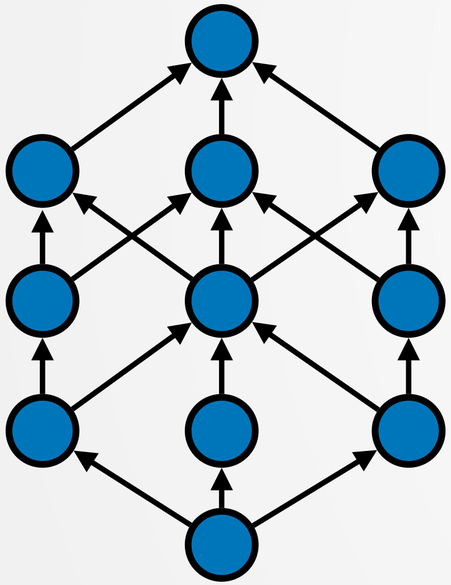
Dichotomy vs Spectrum

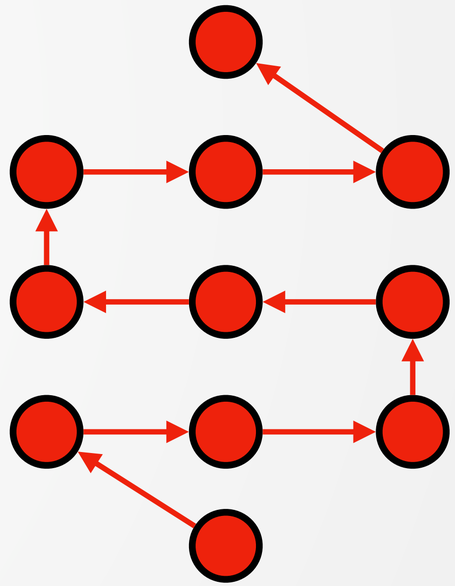
WEAK

Consistency

STRONG

Mixed / Hybrid Consistency





Classify operations as weak or strong… but “how”?

Linearizable State

Eventually Consistent State

Globally Agreed Dependency

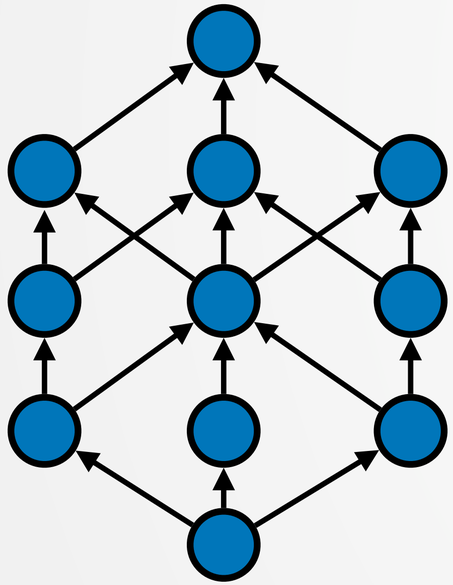
Causal Dependency

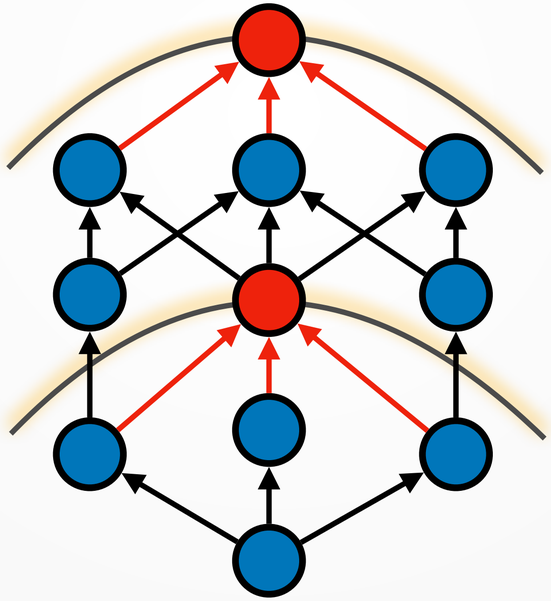
The Event Horizon Approach

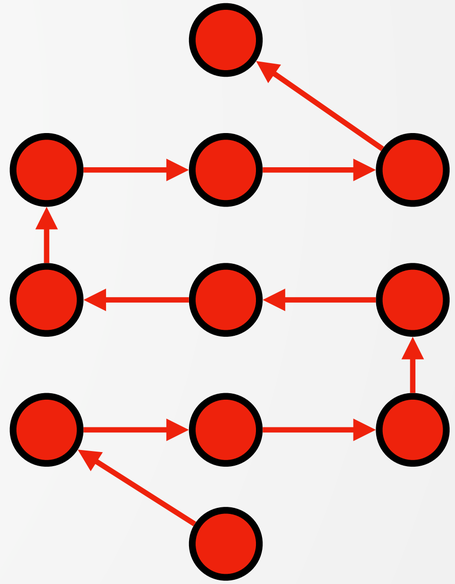
WEAK

Consistency

STRONG

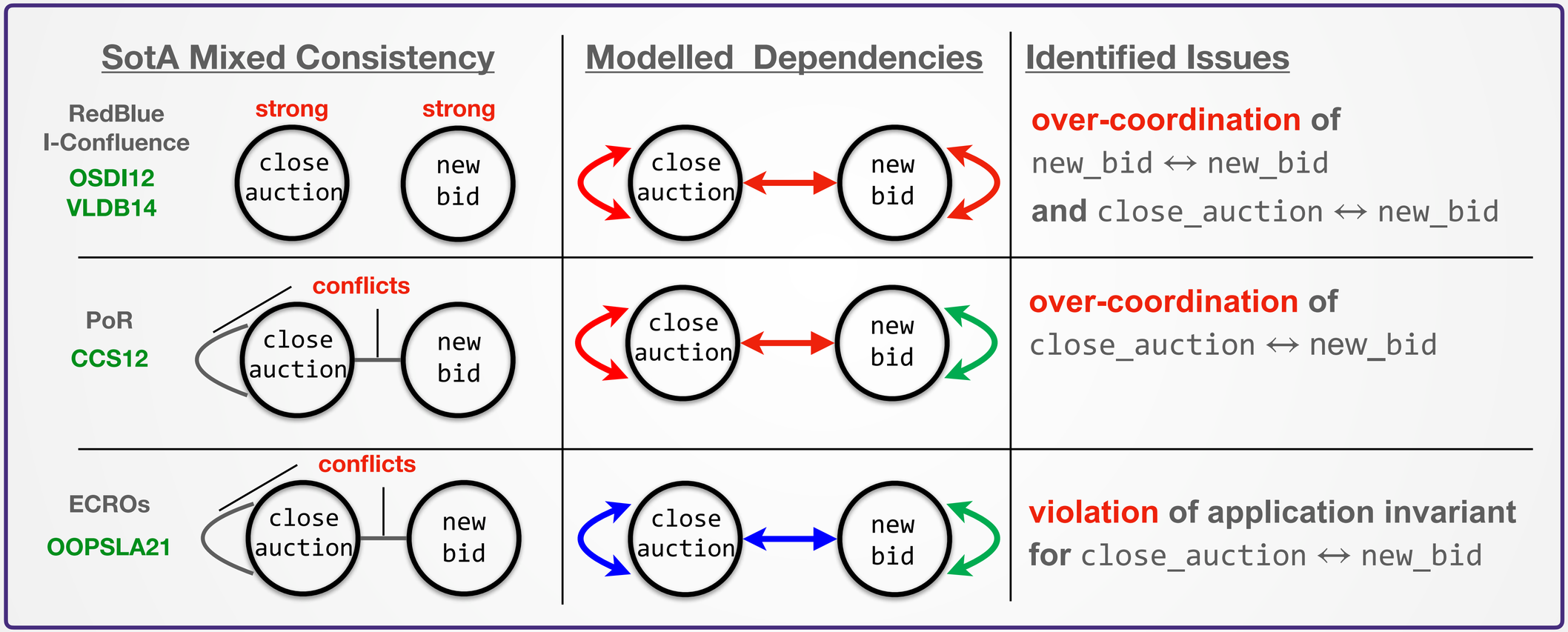






Where Symmetry Fails

Example: "Single Auction Winner = agreeing on highest **new_bid** at **close_auction**

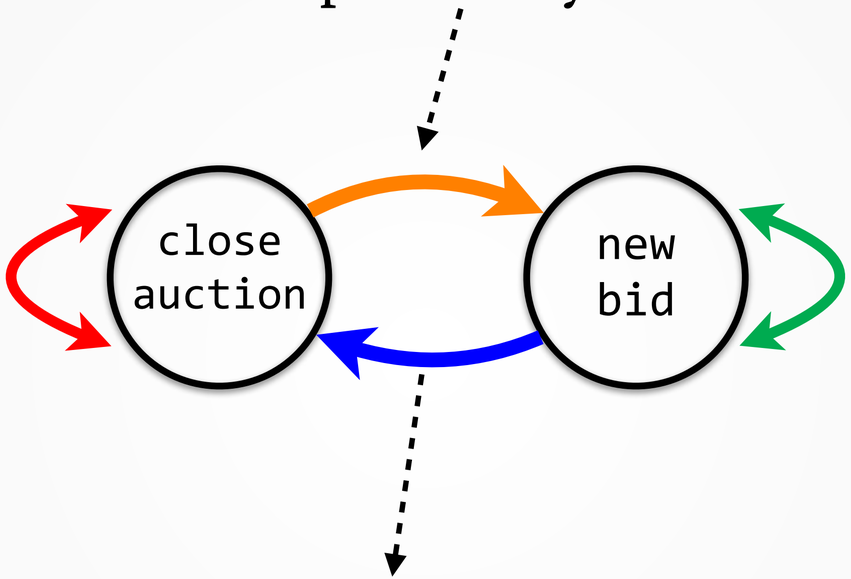


↔ strictly ordered ↔ commutative ↔ eventually ordered

Symmetric models force both directions to pay the same cost.

Introducing Asymmetric Dependencies

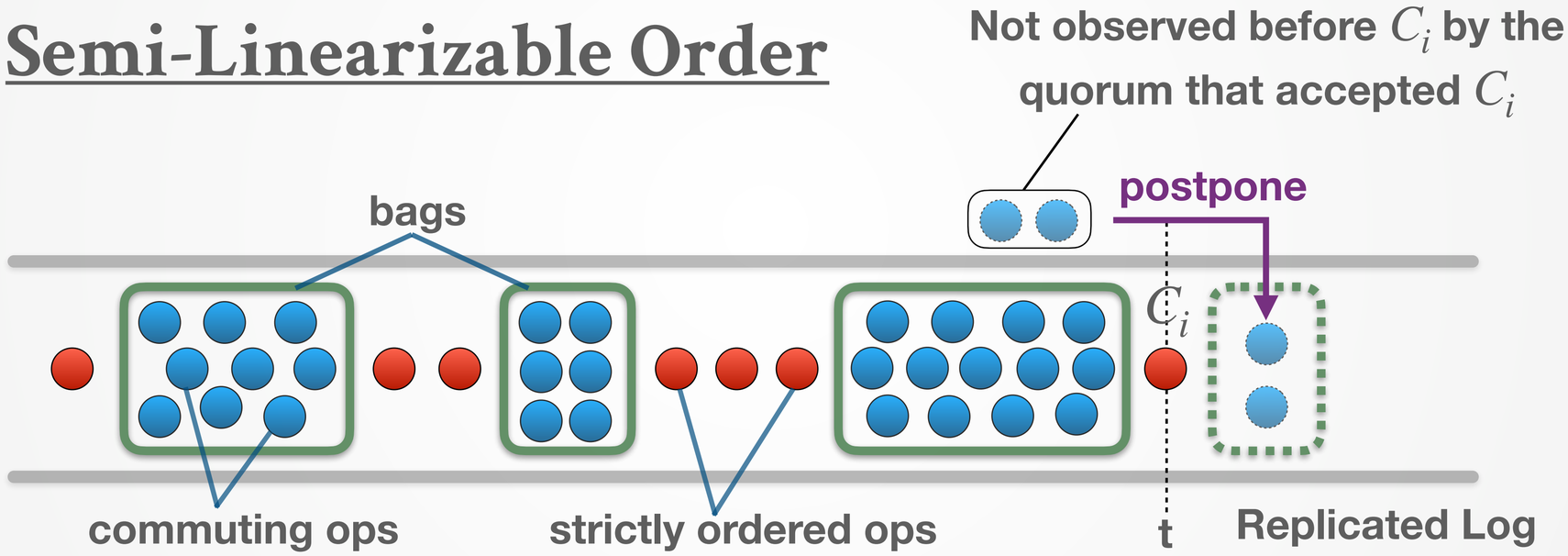
Closing an auction, commits all previously observed new_bid ops



A new bid can be committed before/after a concurrent or future close_auction

↔ strictly ordered ↔ commutative → eventually ordered → ordered

SEMI-LINEARIZABILITY (SL)

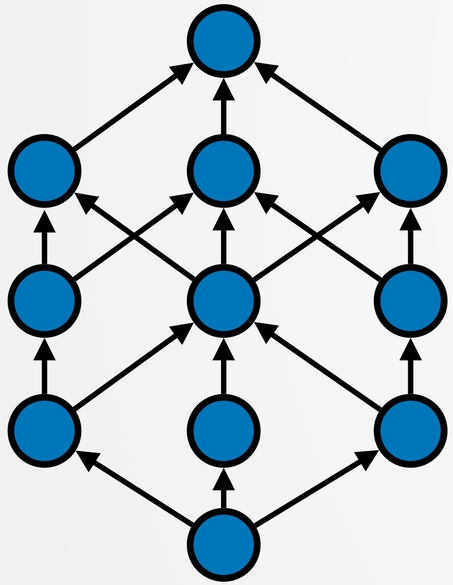


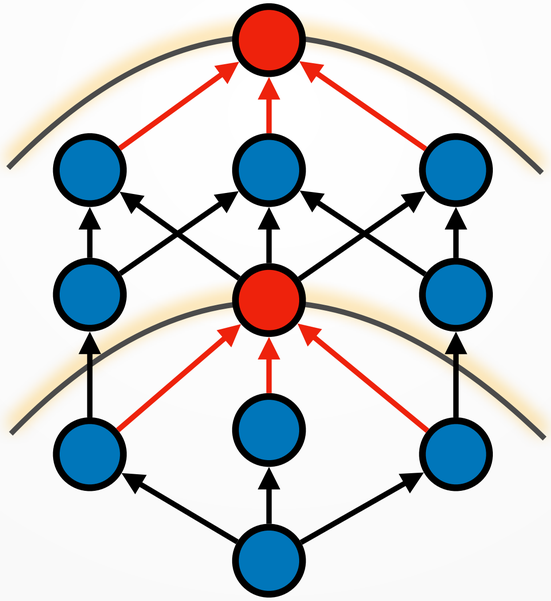
The Event Horizon Principle

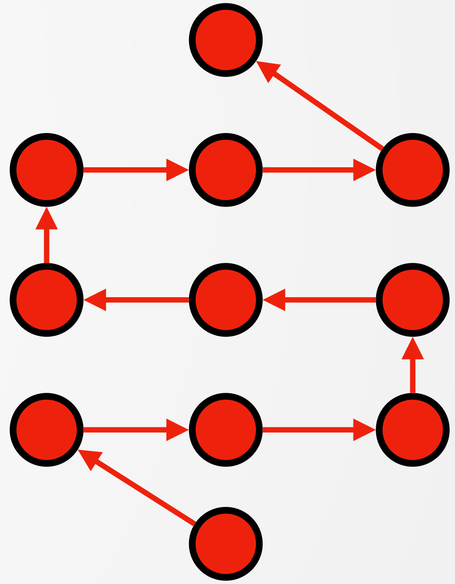
WEAK

Consistency

STRONG







Linearizable State

Eventually Consistent State

Globally Agreed Dependency

Causal Dependency

Event Horizon

DeMon

DUAL-PATH EXECUTION

Weak Operations Path

1. Execute locally at receiving replica
2. Respond to client immediately
3. Replicate via causal broadcast (async)

Strong Operations Path

1. Attach watermark (vector clock)
2. Decide via consensus (OmniPaxos)
3. Enforce stable state, respond to client

KEY MECHANISMS

Watermarks

Vector clocks summarizing which weak ops must be ordered before strong ops

Versioned State

Stable: committed behind event horizon

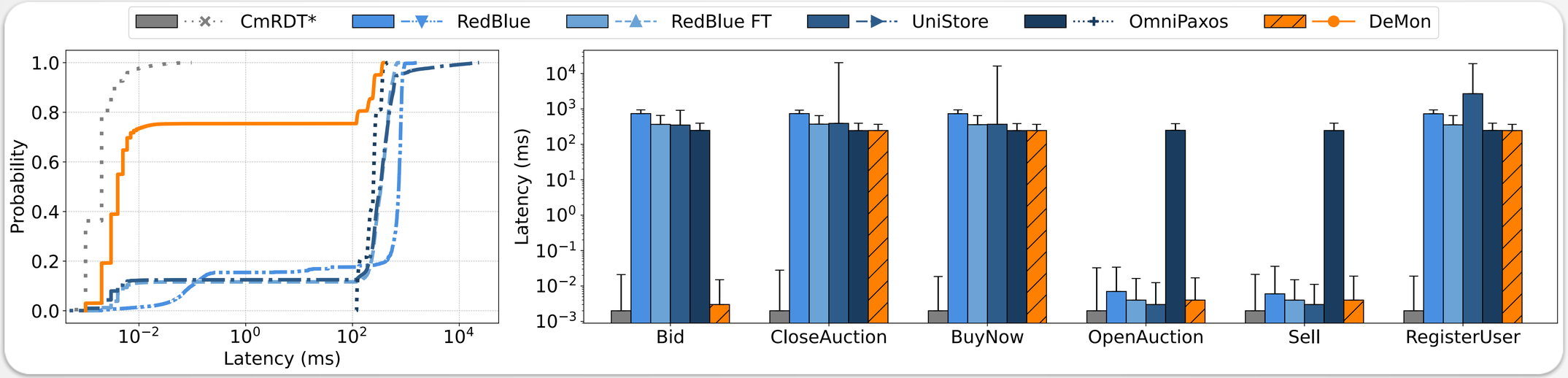
Unstable: recent weak ops (can be reordered)

Reordering

Weak ops arriving after a strong op commits are applied to unstable state and marked as commited after the event horizon

github.com/JonathanArns/demon

DeMon Eval



No Guarantees (CmRDT*): Causal broadcast only — fast but unsafe
RedBlue: Original RedBlue — not fault-tolerant
RedBlue FT: RedBlue using OmniPaxos for strong ops / FT
UniStore: PoR-based with optimistic commit
OmniPaxos: **linearizable / strictly serializable**

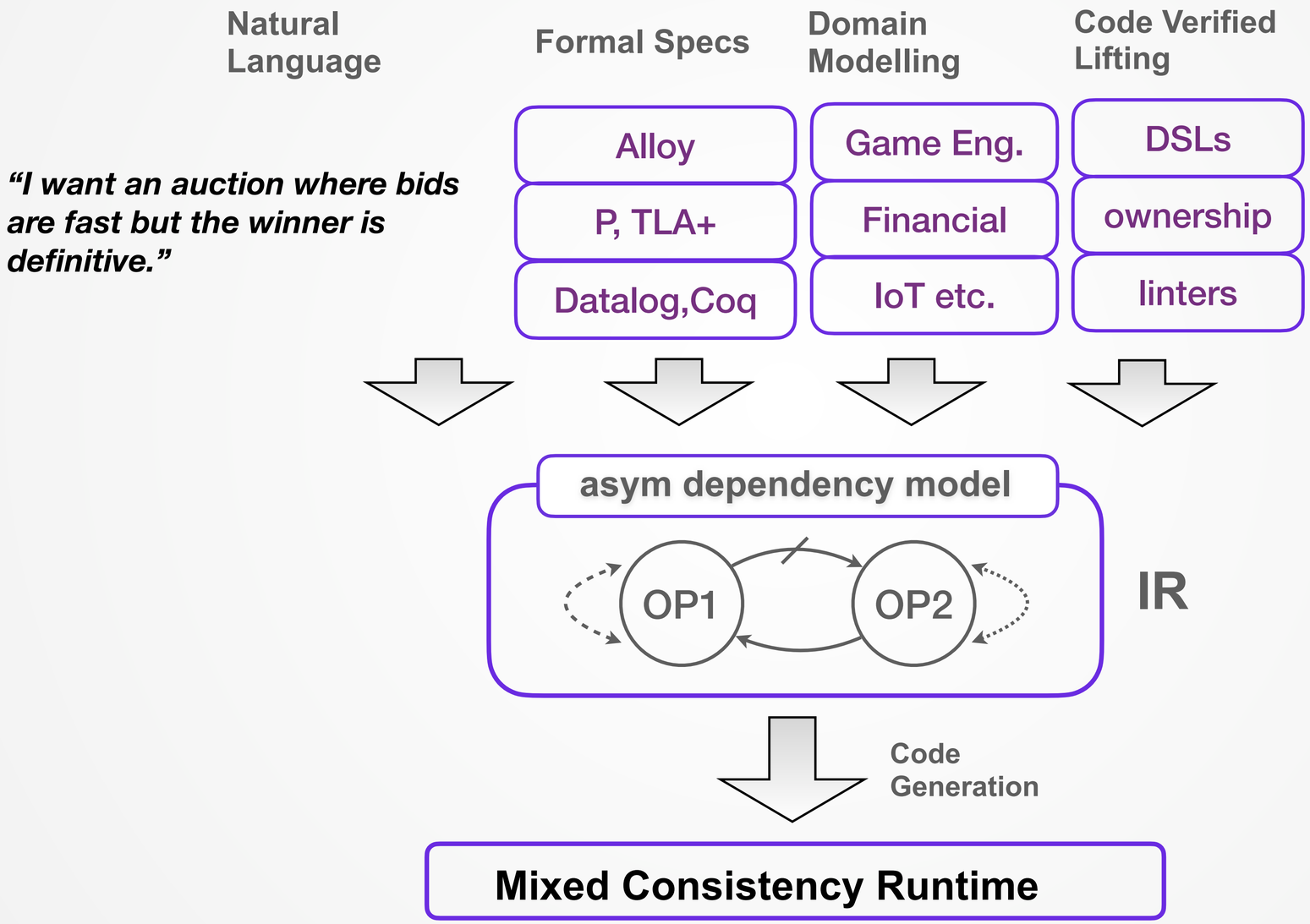
RuBis benchmark

new_bid: 60% of update workload

5 replicas across US-East, Finland, Brazil, US-West, and Singapore.

Round-trip latencies ˙ ~74-388ms. Primary is in US-East

Towards Automated Synthesis



Conclusion

Asymmetric Dependencies

Formalized directional ordering requirements between operations

Semi-Linearizability

New consistency model with event horizons that collapse histories when needed

DeMon Runtime

up to 3 orders of magnitude lower latency on frequent operations while preserving invariants

Asymmetric op. dependencies are **more nuanced** than binary conflict.
Leverage directional relationships to reduce coordination.