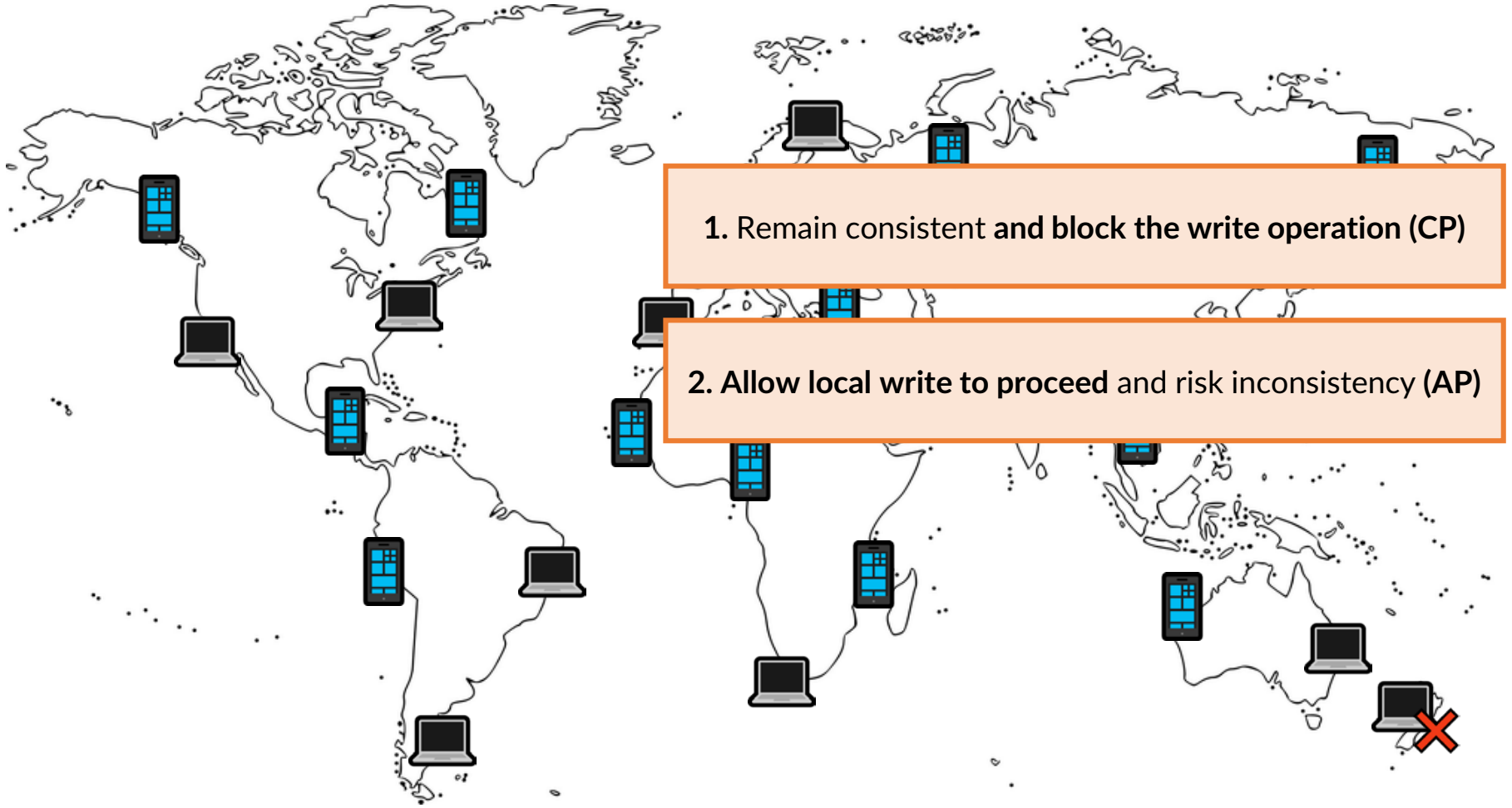
C8DB
Geo-Replicated, Conflict-Free Document Database

Christopher S. Meiklejohn
Macrometa

HPTS 2019

November 5th, 2019

Geo-Distribution Challenges



1. Remain consistent **and block the write operation (CP)**

2. Allow local write to proceed and risk inconsistency **(AP)**

Conflict-Free Replicated Data Types

Distributed data structures, mimic sequential counterparts
Designed for **convergence** under concurrency
Multiple flavors (e.g., **state-based**, delta-based

What do we do when **things don't commute?**

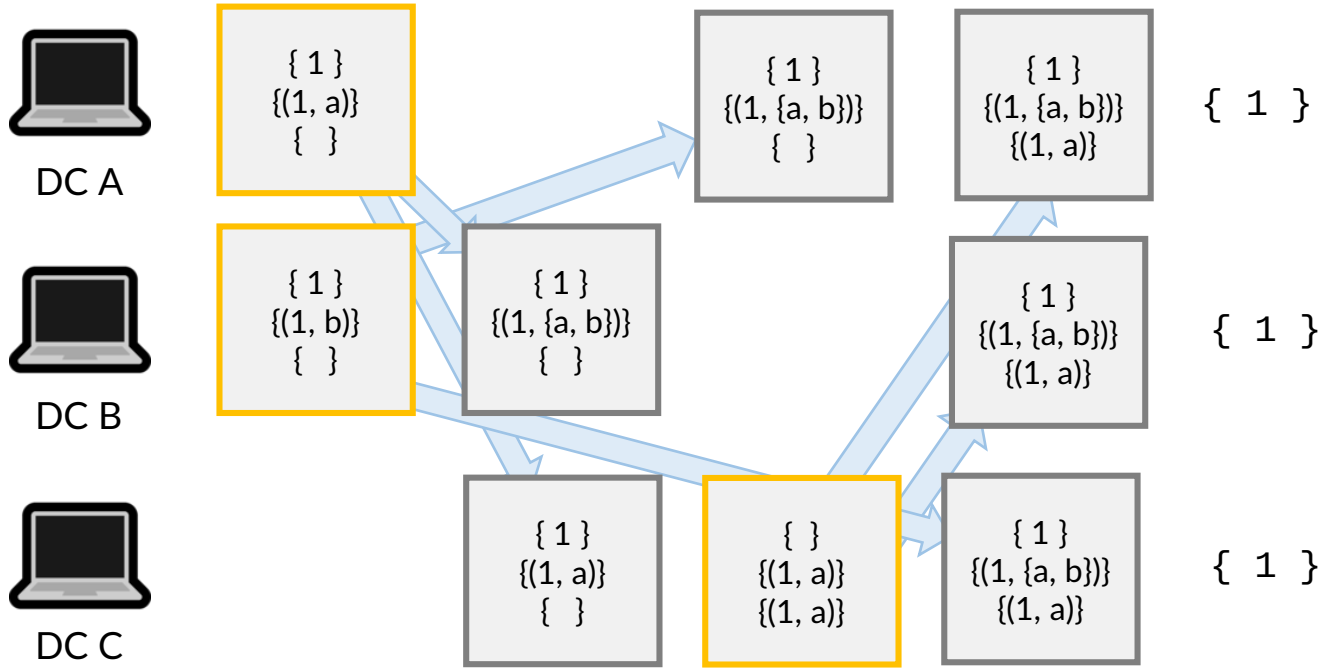
Counters
G-Counter
PN-Counter

Sets
G-Set
~~2P-Set~~
OR-Set
~~AW-Set~~
RW-Set

Registers
LWW-Register
MV-Register

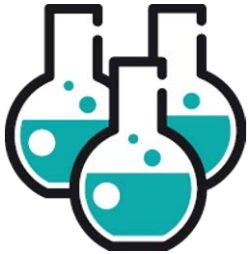
Flags

Sequence
Treedoc
RGA
RGA-Split



CRDTs in Production

Riak distributed database
First major production implementation of state-based CRDTs
Used by NHS, Riot Games, LO/JACK, bet365



AntidoteDB (HPTS '17)
Developer on operation-based CRDT database with Transactional Causal Consistency
Part of the LightKone, SyncFree research projects

Others
Mesosphere's DC/OS (Lashup, Minuteman), Adobe, Comcast, Macrometa
TomTom, Roshi, Automerge, Ditto, Cosmos DB, etc.

Macrometa: Overview

Document database for the edge
Documents are **JSON** and grouped into **collections**
(think: namespaces for each document, clients presented with dictionaries)

105 physical regions, 175 logical regions
36 **reserved core** regions, 12 available through public beta today
34 **customer** requested regions (on-prem, specific colo, etc.)
35 **burstable** regions available

Configurable consistency
Strong consistency in a single DC with propagation to remote DCs in order
Eventual consistency in an multi-master mode with all DCs receiving updates

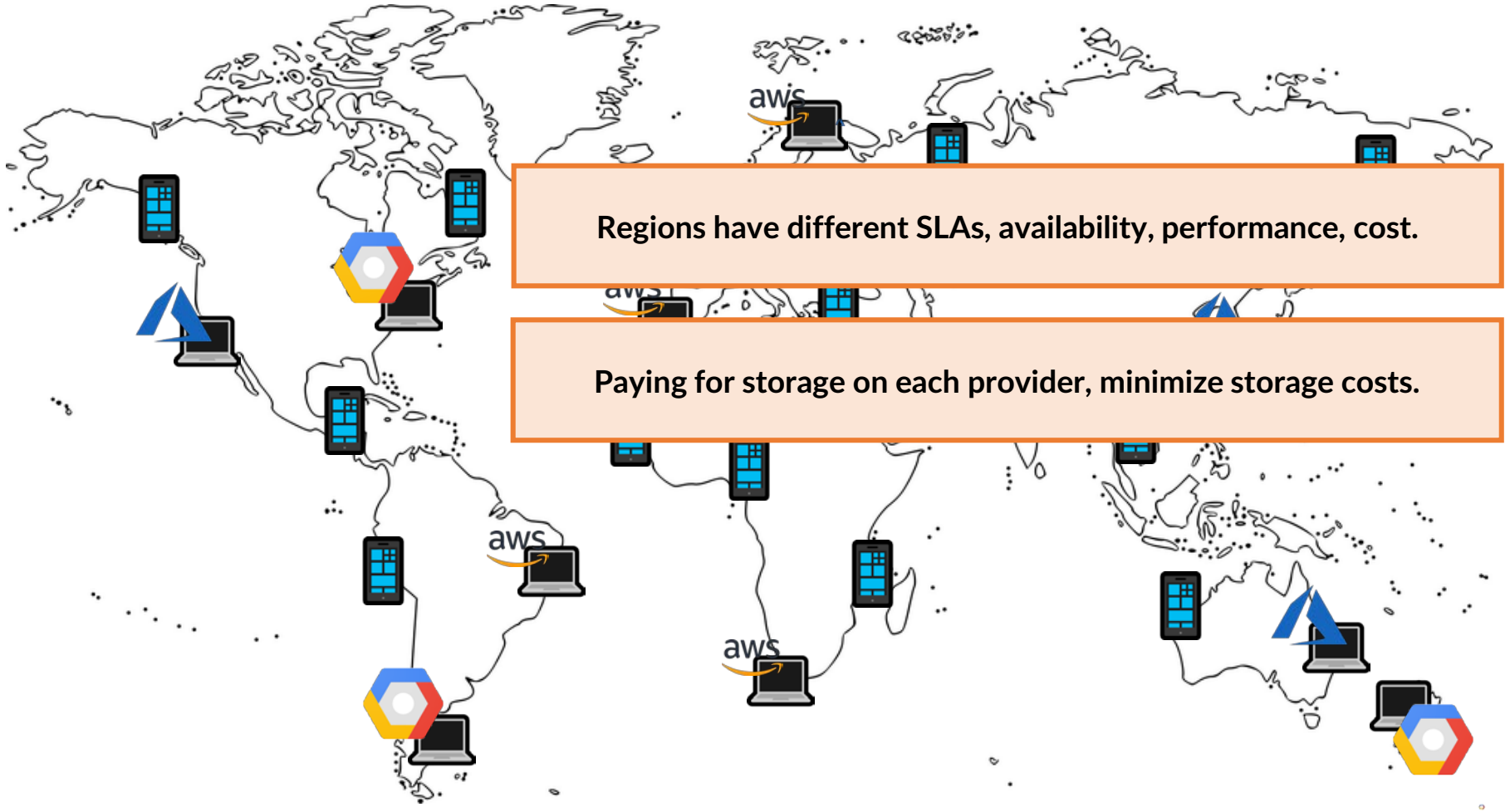
Macrometa: Architecture

Logging-based Architecture
Updates within a data-center are **durably written to a log**
Logs are maintained with **Zookeeper** and **BookKeeper**

Causal broadcast
There is a single log per destination DC delivered in **FIFO order**
Updates are buffered until log entries **dependencies are met**

Client operate on the **document** abstraction level
Clients submit documents that are **converted to updates** by Macrometa nodes
Macrometa nodes **materialize values** for keys based on incoming log updatges

Geo-Distribution Concerns



Regions have different SLAs, availability, performance, cost.

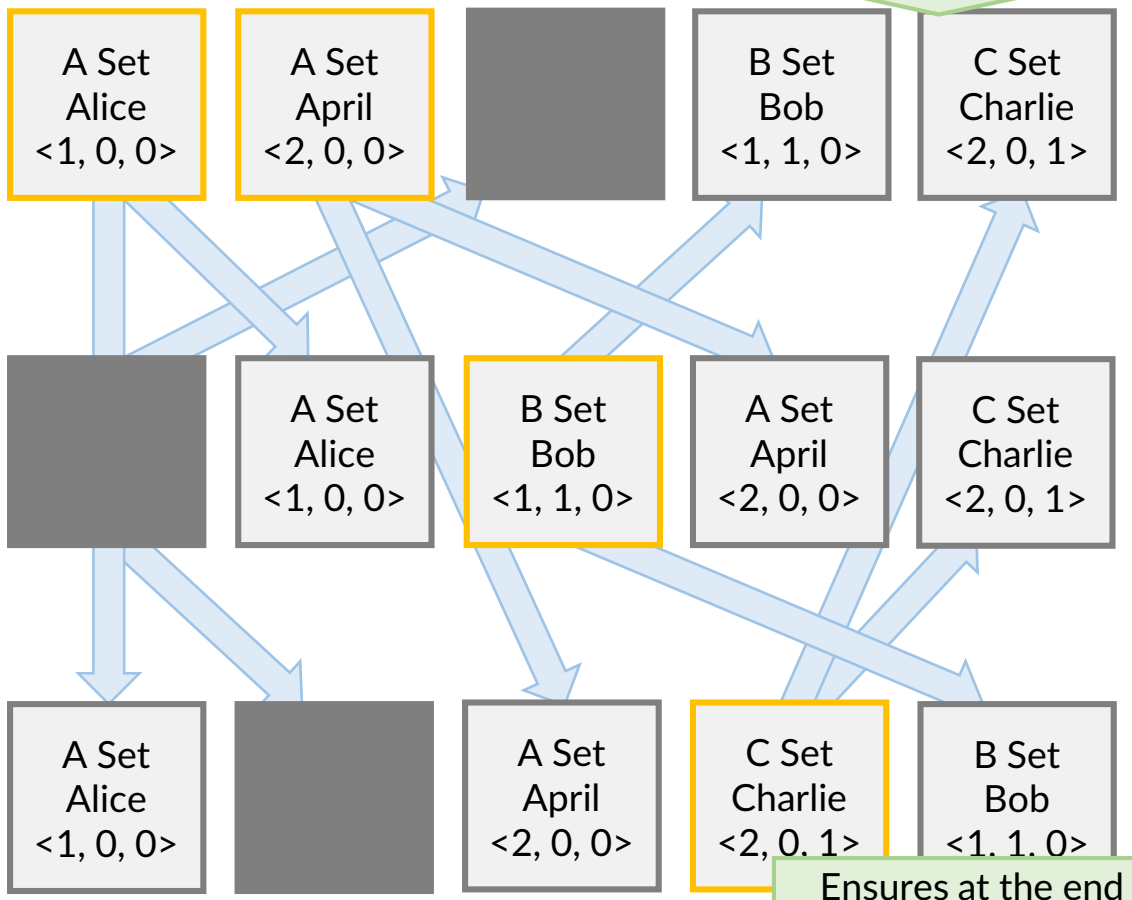
Paying for storage on each provider, minimize storage costs.

Registers

DC A

DC B

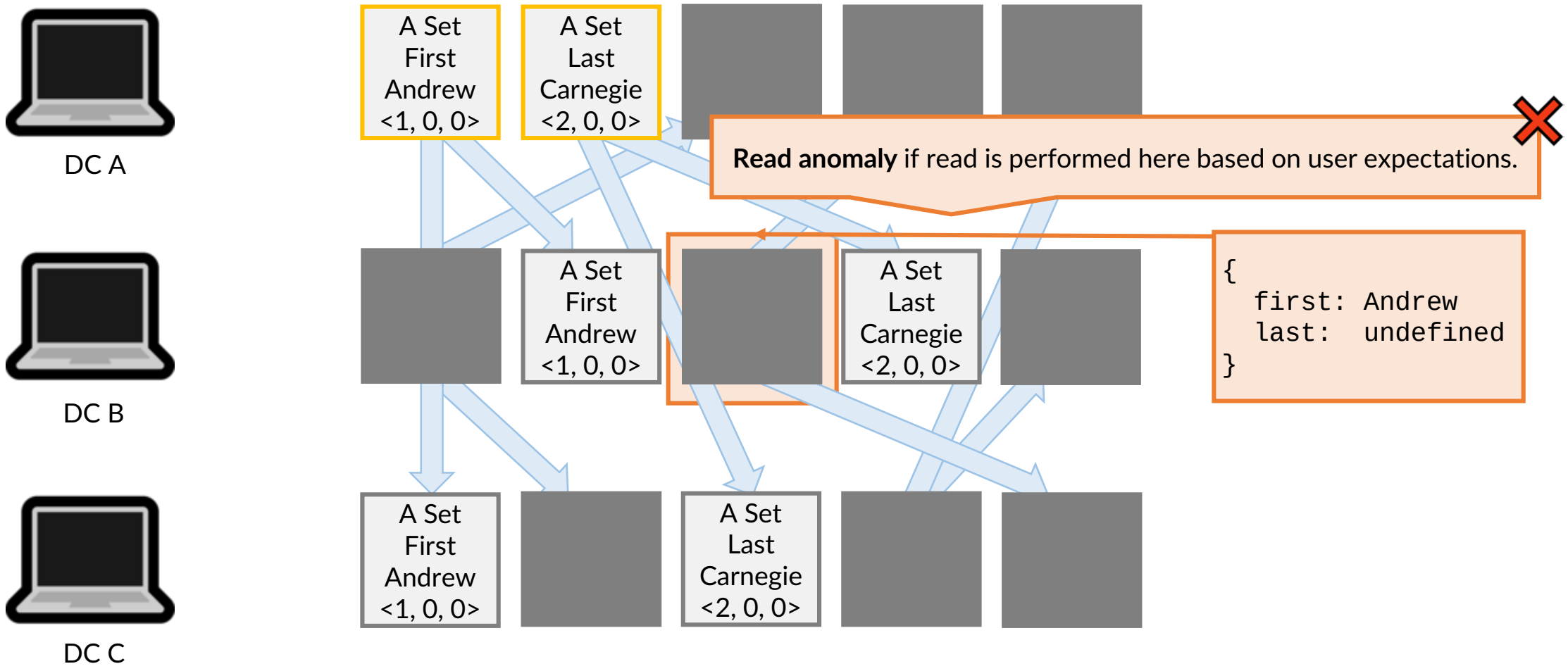
DC C



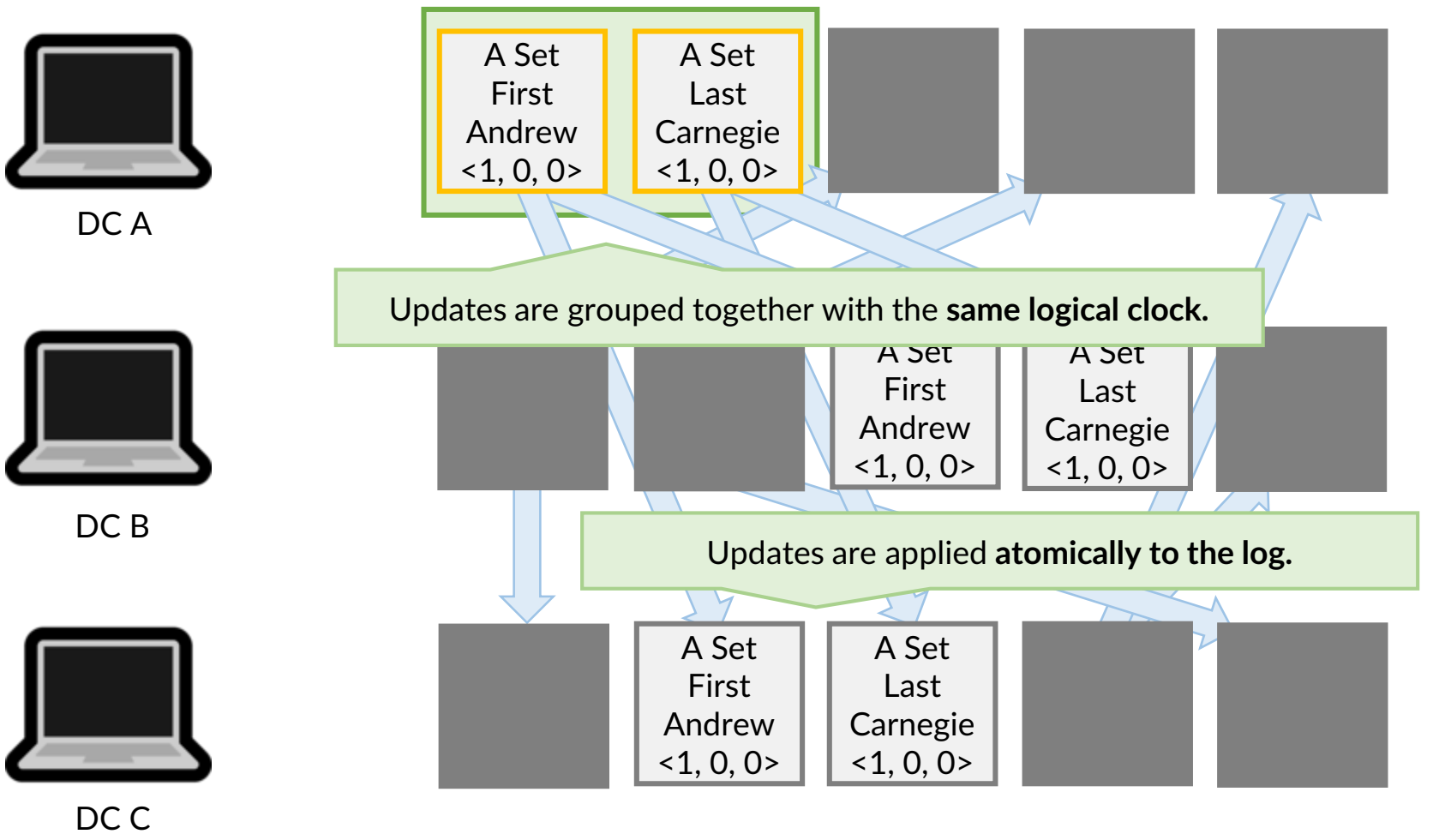
Concurrent updates merge arbitrate based on the **maximum DC identifier.**

Ensures at the end of convergence we need to only keep **a single update in the log for each field in the document.**

Registers: Read Anomalies

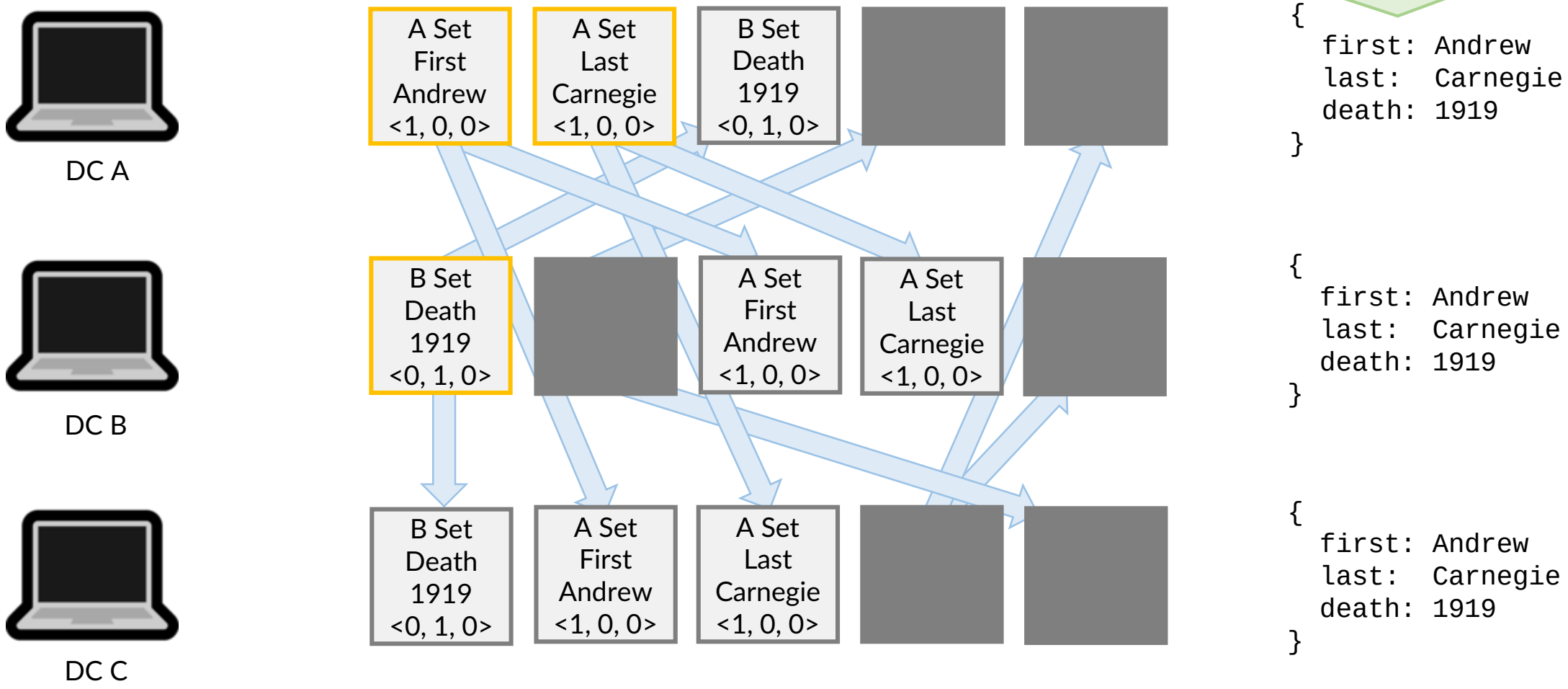


Transactions v1

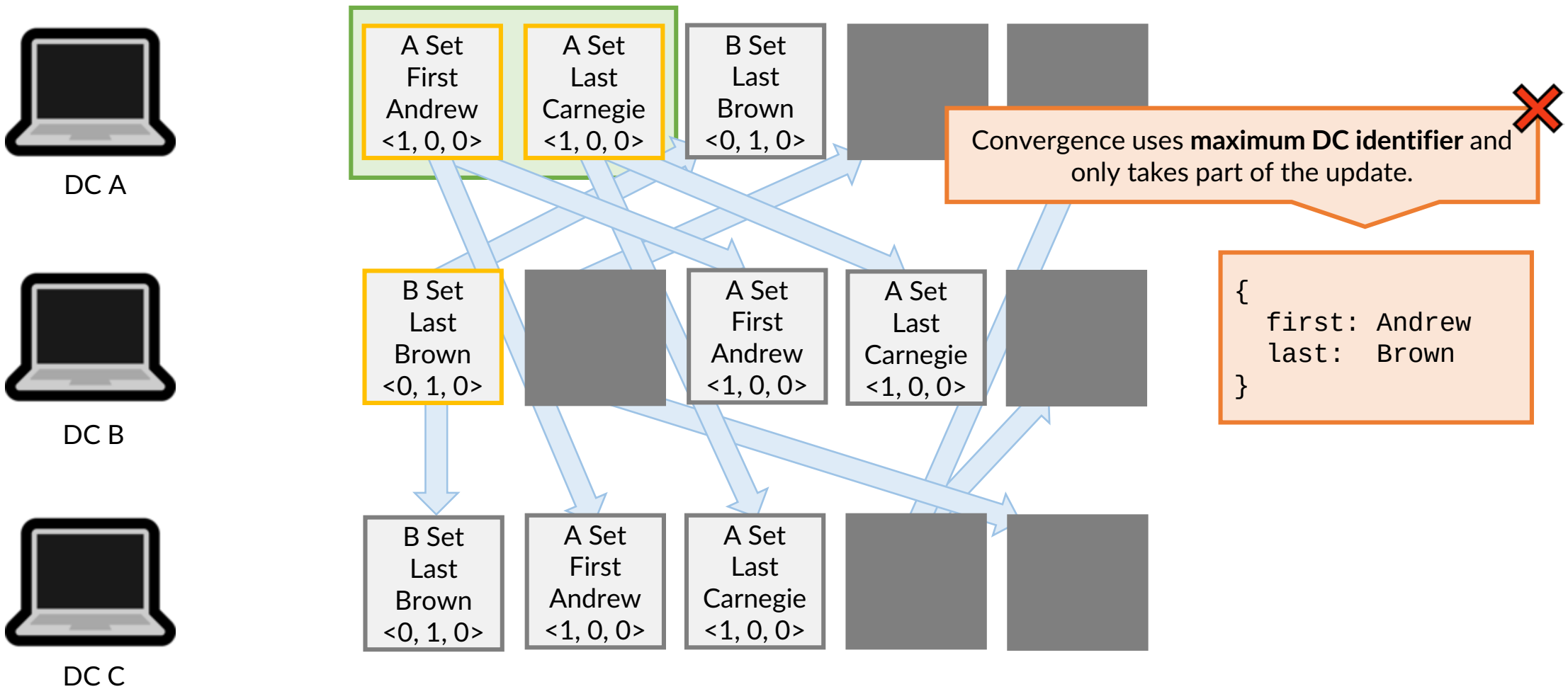


Transactions v1: Concurrency

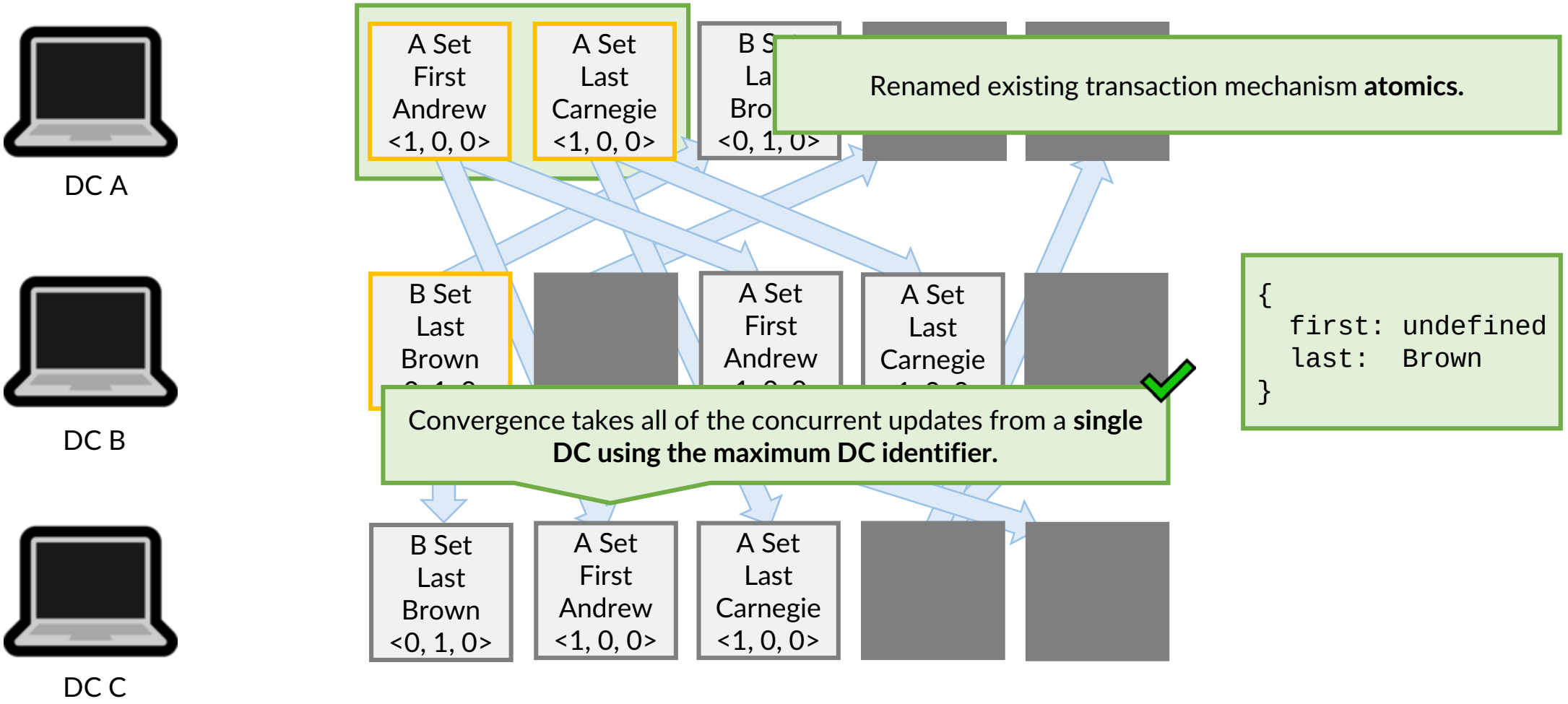
Concurrent operations are merged together using the existing convergence strategy by **maximum DC identifier.**



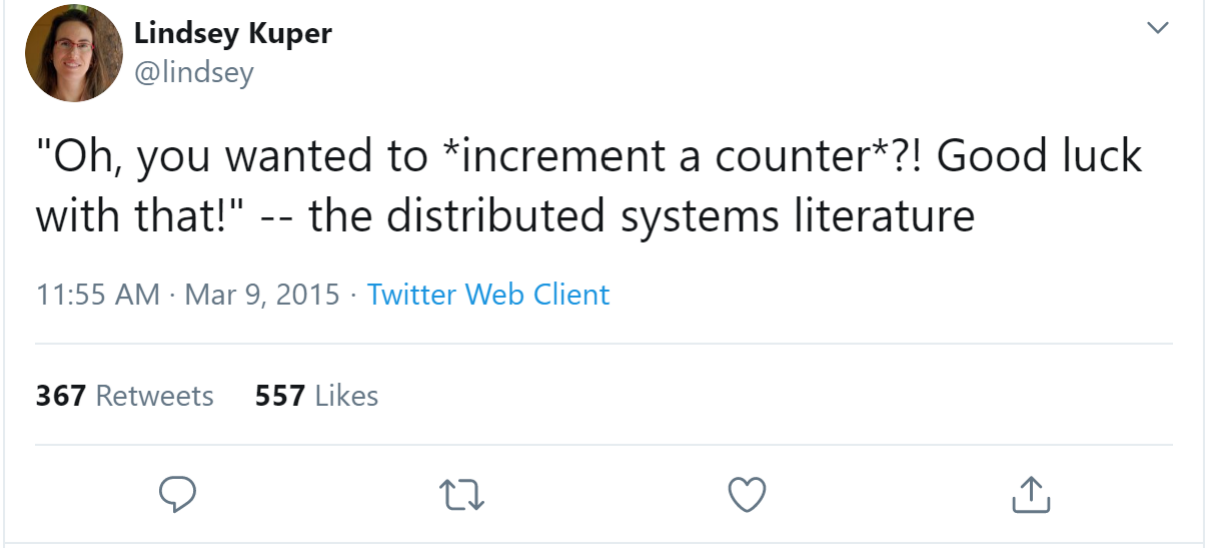
Transactions v1: Concurrency Anomalies



Transactions: v2



Counters?



Can't use a register, **assignment doesn't commute.**

Sum double counts, **max** under counts.

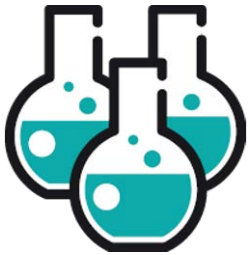
Store sum per node or…

…store individual updates.

Types of CRDT Counters

State-based CRDTs
Lattice-based, lower for storage overhead and require that developers ensure that all operations are always mergeable by proper design.
e.g., Riak

How do we find a solution that is **compatible** with the **transactions** that meets both the **garbage collection** and **storage** requirements?



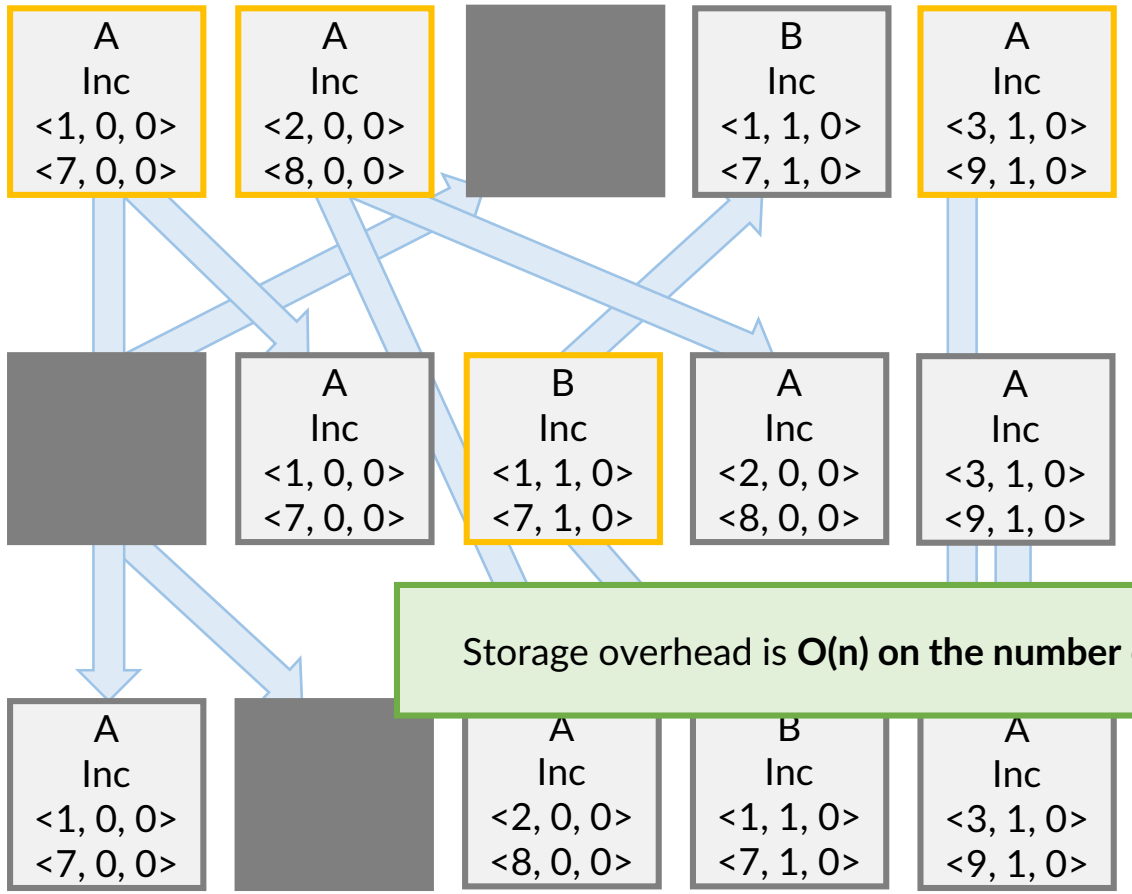
Operation-based CRDTs
Optimized for individual operations where all operations are designed to be commutative, but do not need for form a lattice. Also requires causal broadcast.
e.g. Antidote

State-based

DC A

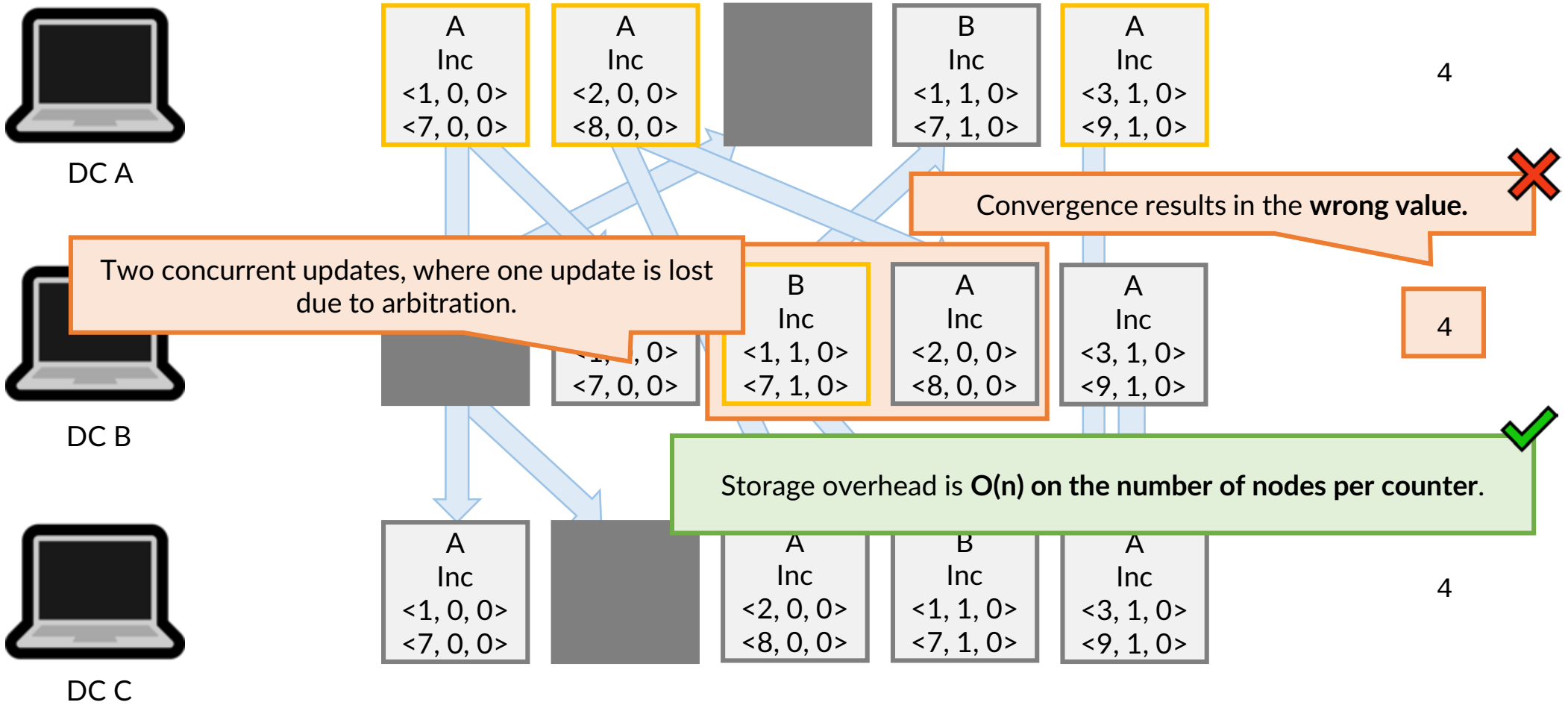
DC B

DC C

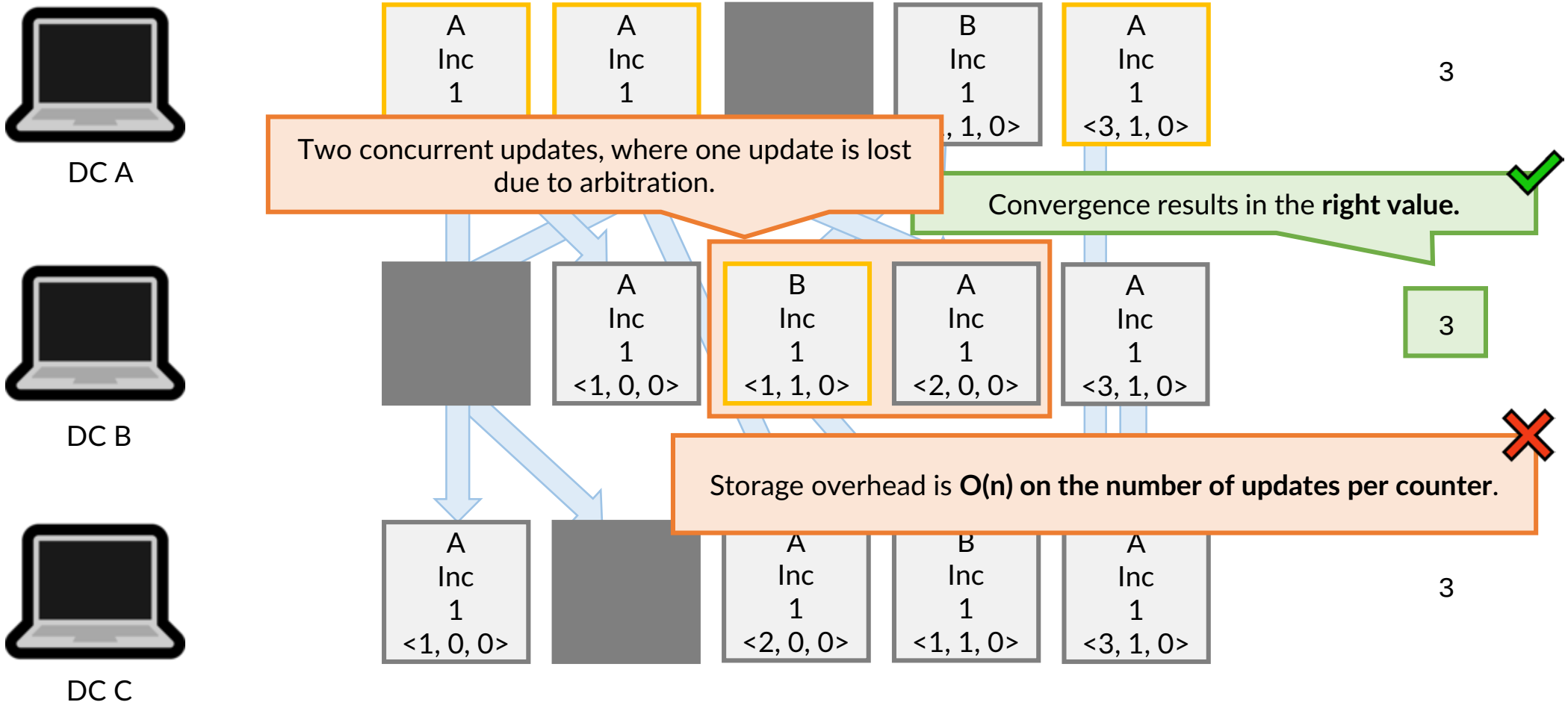


Storage overhead is **O(n) on the number of nodes per counter.**

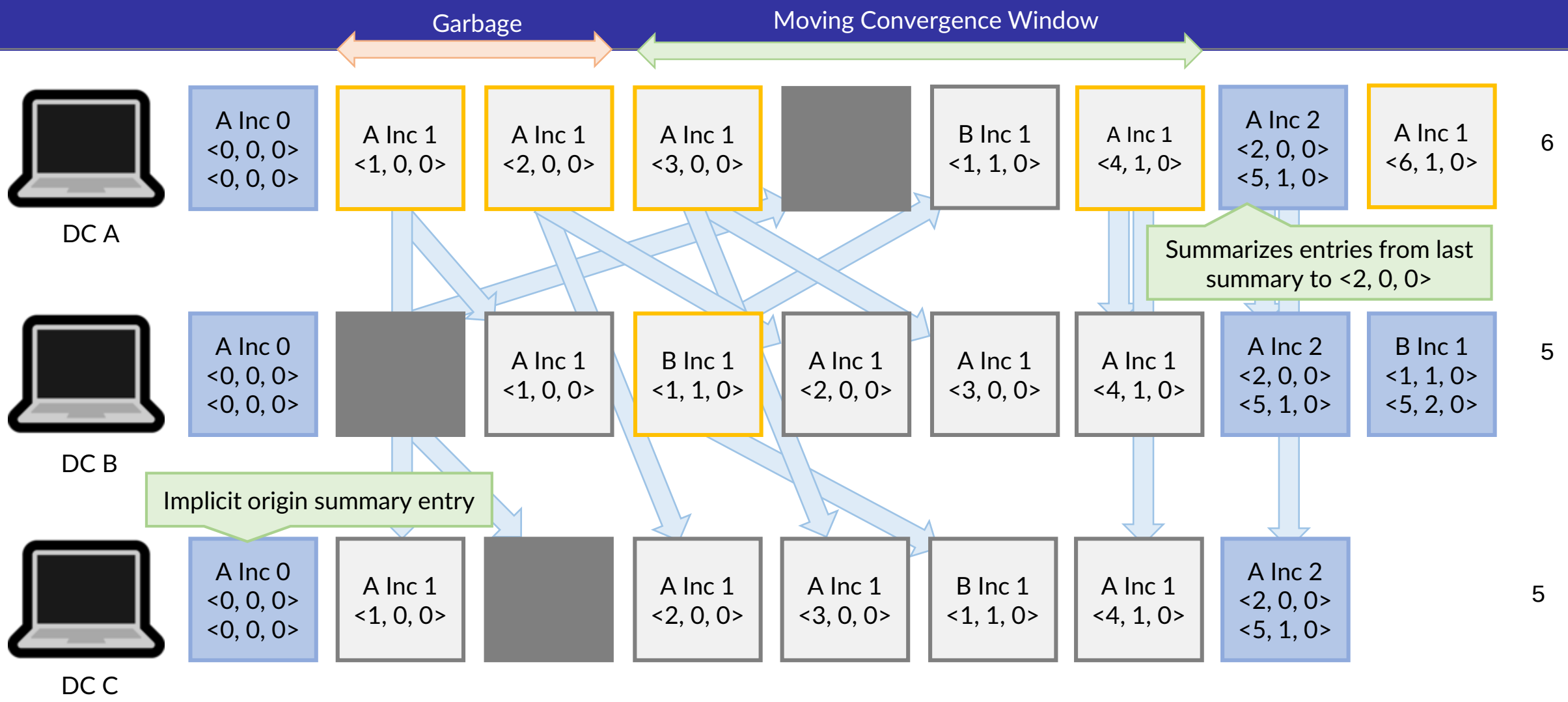
State-based and Transactions



Operation-based and Transactions



Counters


Garbage
Moving Convergence Window
DC A
DC B
DC C
A Inc 0 <0, 0, 0> <0, 0, 0>
A Inc 1 <1, 0, 0>
A Inc 1 <2, 0, 0>
A Inc 1 <3, 0, 0>
B Inc 1 <1, 1, 0>
A Inc 1 <4, 1, 0>
A Inc 2 <2, 0, 0> <5, 1, 0>
A Inc 1 <6, 1, 0>
6
Summarizes entries from last summary to <2, 0, 0>
A Inc 0 <0, 0, 0> <0, 0, 0>
A Inc 1 <1, 0, 0>
B Inc 1 <1, 1, 0>
A Inc 1 <2, 0, 0>
A Inc 1 <3, 0, 0>
A Inc 1 <4, 1, 0>
A Inc 2 <2, 0, 0> <5, 1, 0>
B Inc 1 <1, 1, 0> <5, 2, 0>
5
Implicit origin summary entry
A Inc 0 <0, 0, 0> <0, 0, 0>
A Inc 1 <1, 0, 0>
A Inc 1 <2, 0, 0>
A Inc 1 <3, 0, 0>
B Inc 1 <1, 1, 0>
A Inc 1 <4, 1, 0>
A Inc 2 <2, 0, 0> <5, 1, 0>
5

CRDTs, tho?

CRDTs?
Are these **actually CRDTs?** Probably not, need a new name.
Interesting point in the design space (c.f., Riak, Antidote)

Thanks!