Programming
Weak Synchronization
Models

Christopher S. Meiklejohn
Université catholique de Louvain, Belgium
Instituto Superior Técnico, Portugal




Distributed Runtime
Anabranch

work-in-progress

Anabranch

- Layered approach
  Cluster membership and state dissemination for large clusters
- Delta-state synchronization
  Efficient incremental state dissemination and anti-entropy mechanism [Almeida et al. 2016]
- Epsilon-invariants
  Lower-bound invariants, configurable at runtime
- Scalable
  Demonstrated high scalability in production Cloud environments

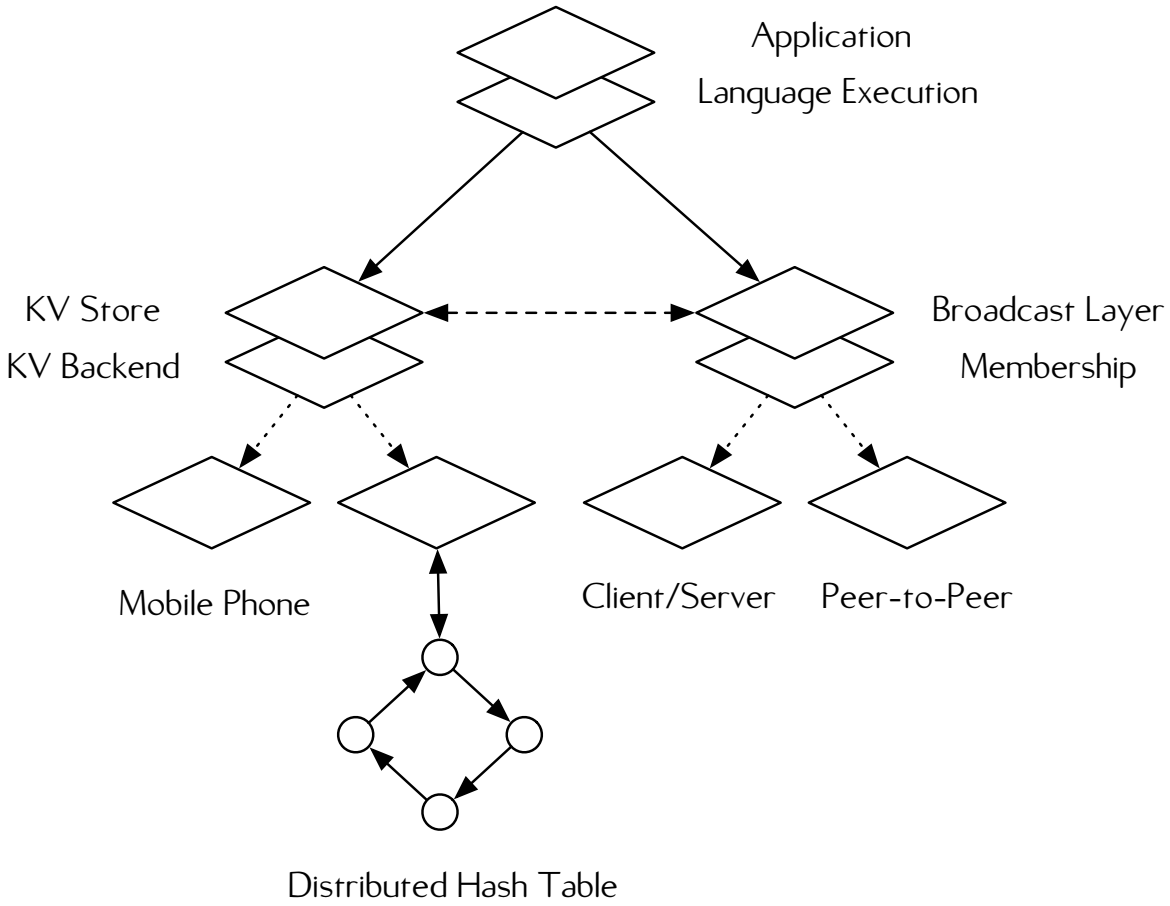
Anabranch
Layered Approach

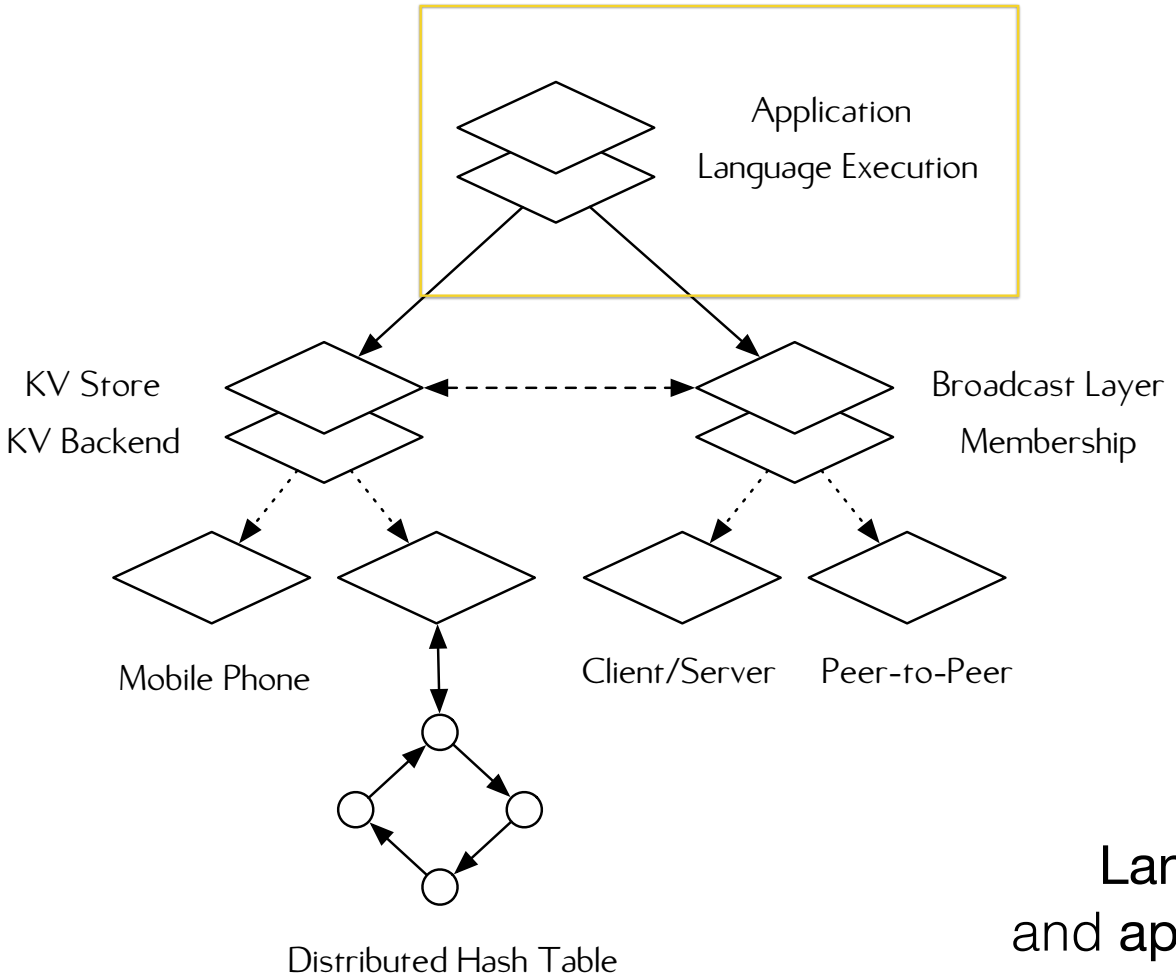
Layered Approach

- Backend
  Configurable persistence layer depending on application.

- Membership
  Configurable membership protocol which can operate in a client-server or peer-to-peer mode [Leitao et al. 2007]

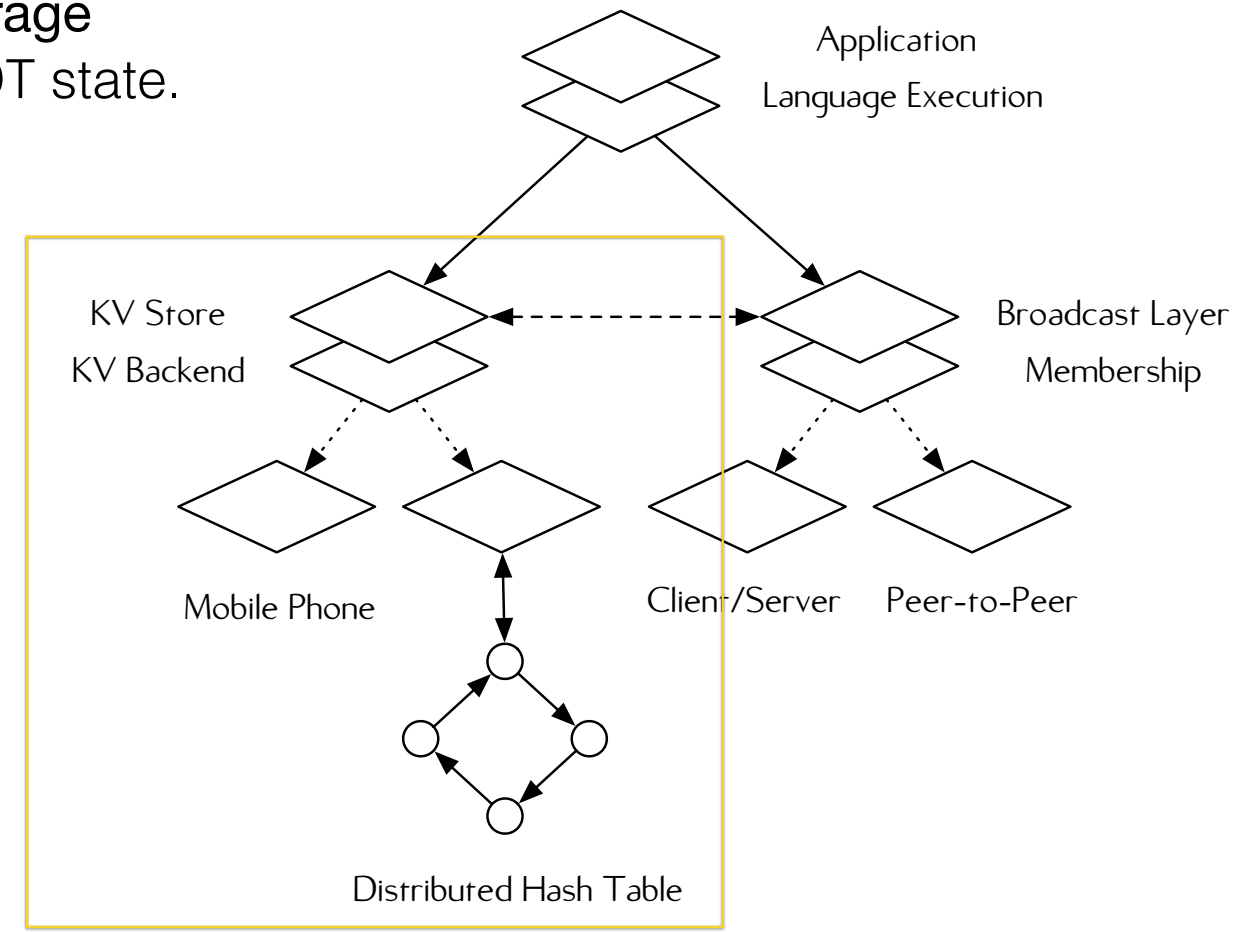
- Broadcast (via Gossip, Tree, etc.)
  Efficient dissemination of both program state and application state via gossip, broadcast tree, or hybrid mode [Leitao et al. 2007]

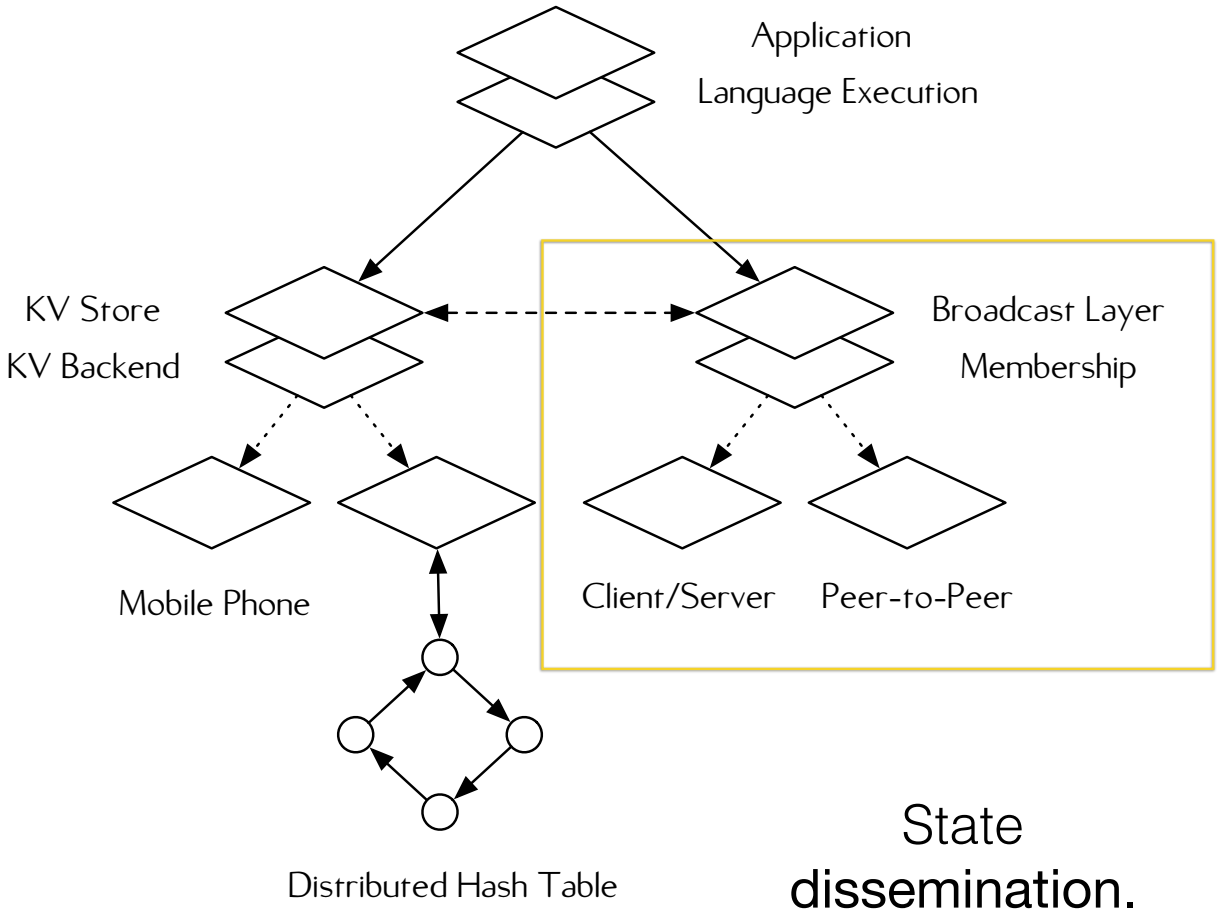

Application
Language Execution
KV Store
KV Backend
Broadcast Layer
Membership
Mobile Phone
Client/Server
Peer-to-Peer
Distributed Hash Table



Language
and applications.

Storage
for CRDT state.





State
dissemination.

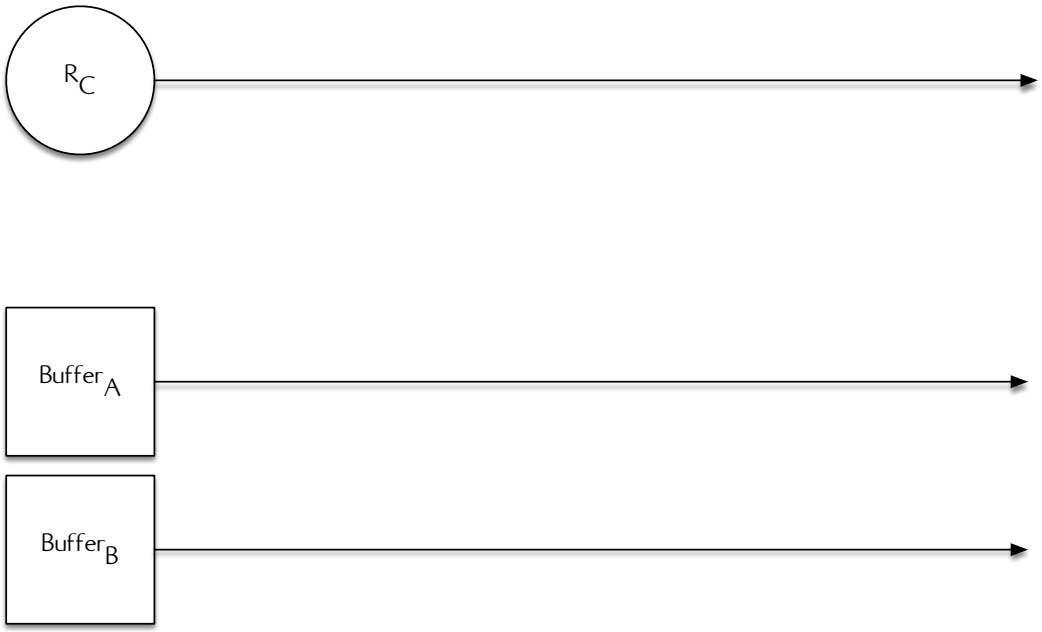
Anabranch
Delta-state CRDTs

Delta-based Dissemination

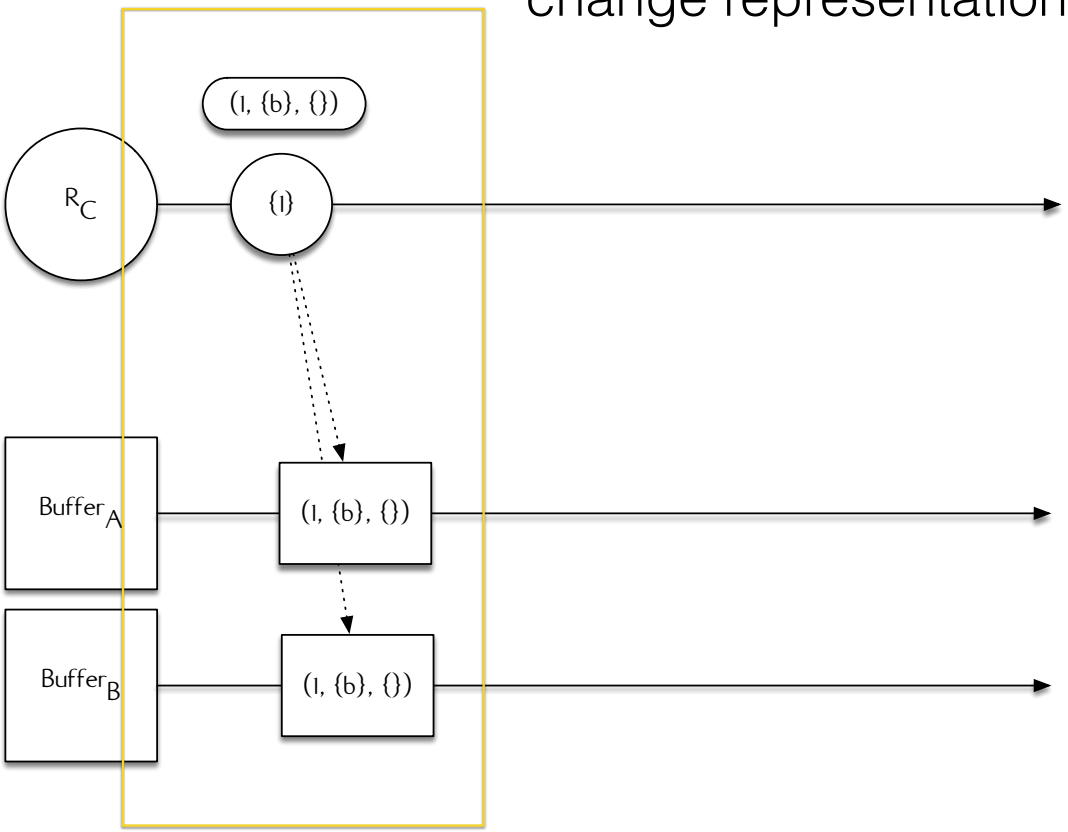
- **Delta-state based CRDTs**
  Reduces state transmission for clients

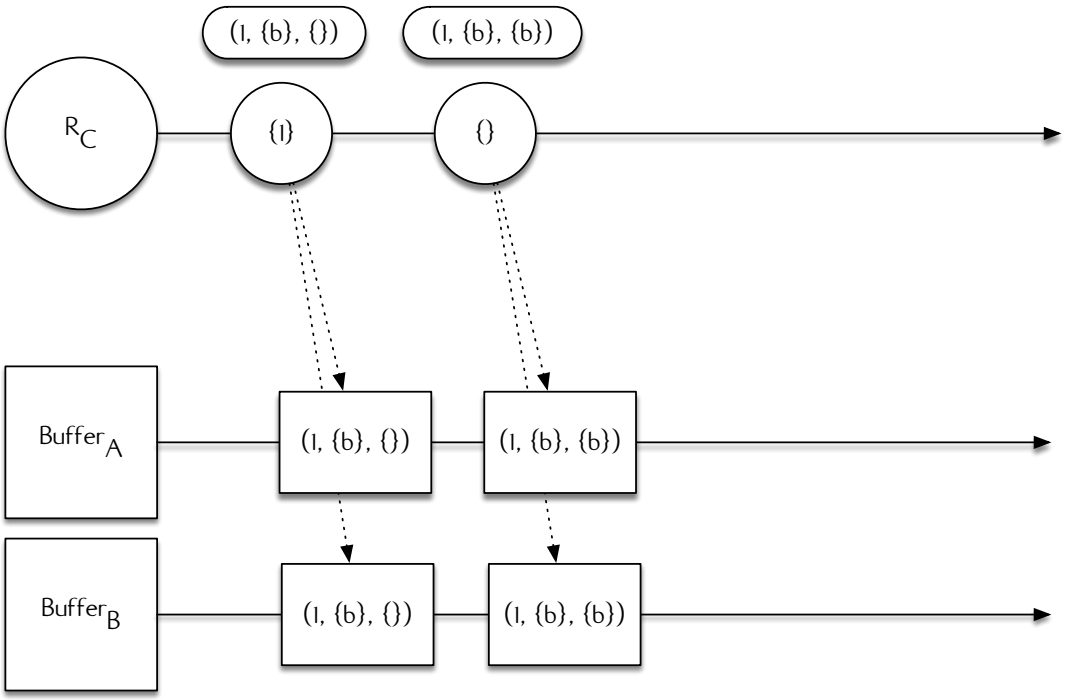
- **Operate locally**
  Objects are mutated locally; delta's buffered locally and periodically gossiped

- **Only fixed number of clients**
  Clients resort to full state synchronization when they've been partitioned too long

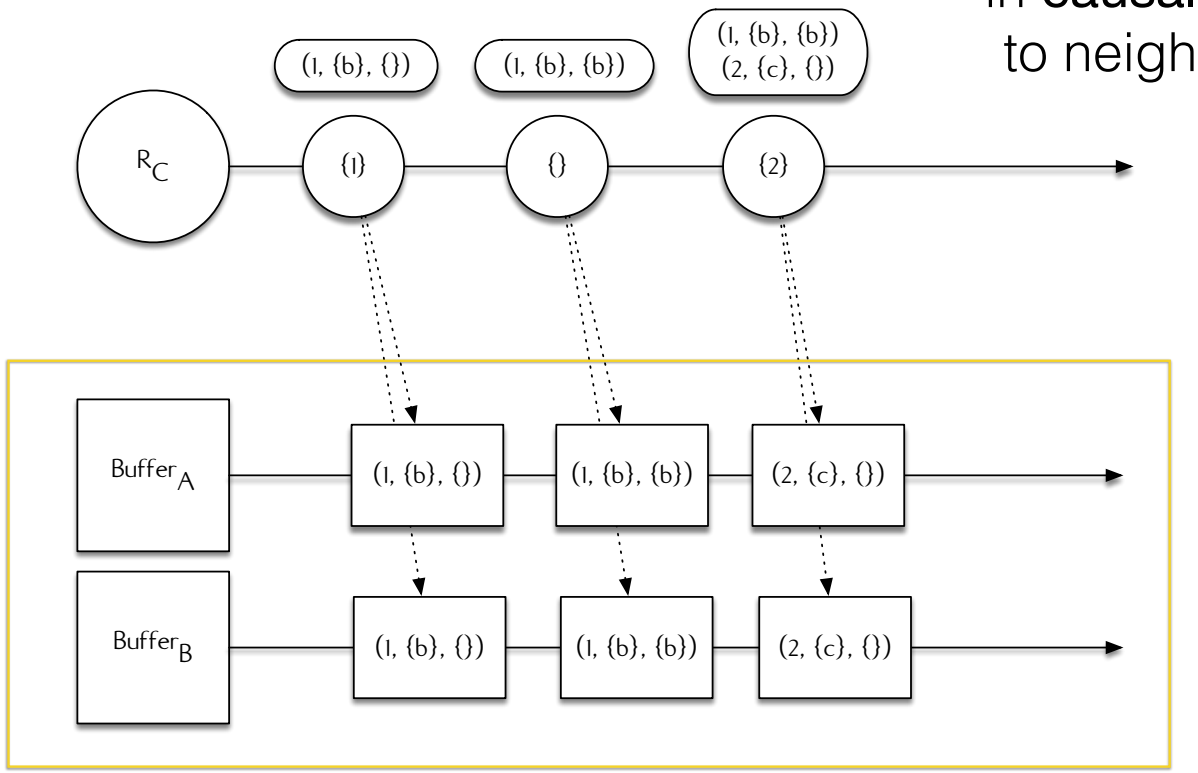

R_C
$Buffer_A$
$Buffer_B$

Buffer **minimal**
change representation…

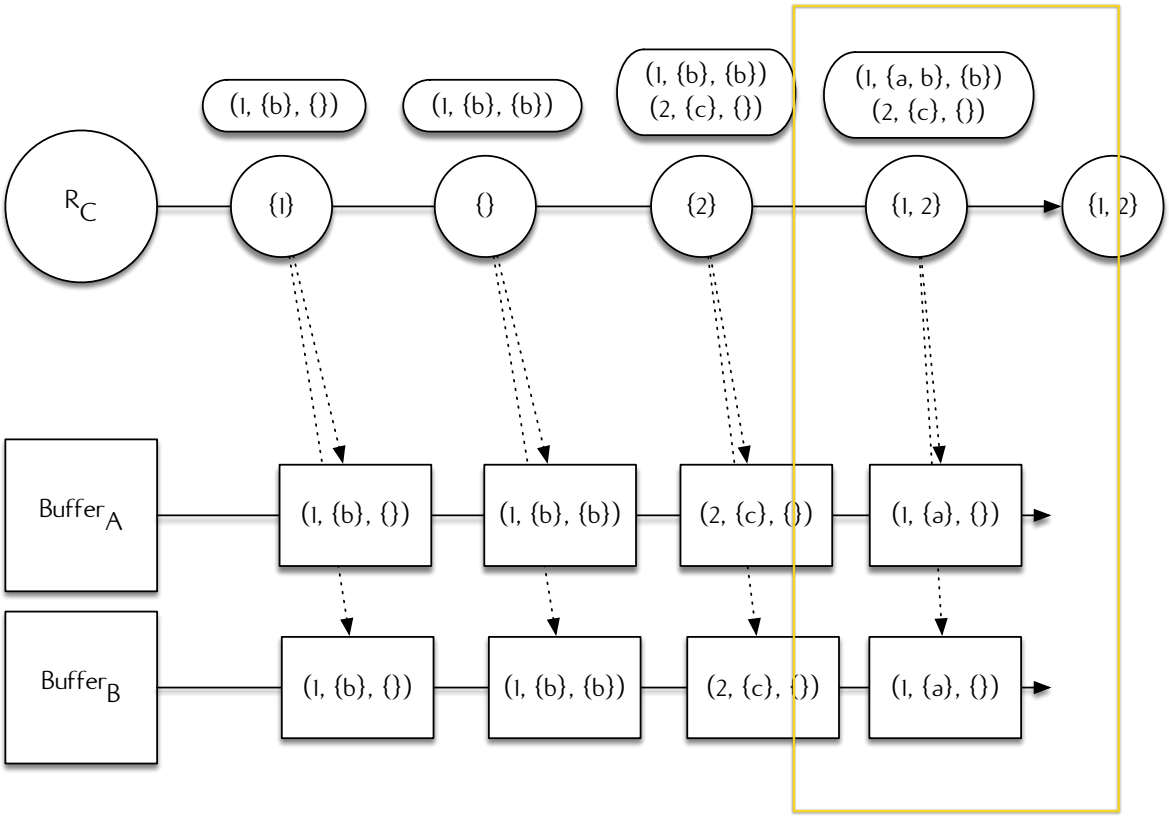



(I, {b}, {})
(I, {b}, {b})
R_C
{I}
{}
Buffer_A
(I, {b}, {})
(I, {b}, {b})
Buffer_B
(I, {b}, {})
(I, {b}, {b})

…then, disseminate state in **causal order** to neighbors.



Only ship **inflation**
from incoming state.



Anabranch
Scalability

Scalability

- 1024+ nodes
  Demonstrated scalability to 1024 nodes in Amazon cloud computing environment

- Modular approach
  Many of the components built and can be operated outside of Lasp to improve scalability of Erlang

- Automated and repeatable
  Fully automated deployment, scenario execution, log aggregation and archival of experimental results