State Management for the Telco's Edge

Bharath Balasubramanian, Principal Inventive Scientist

ATT Labs Research

NSF CPS Workshop, Nov 16, 2018

ATT’s Interaction with the Edge

Life cycle management of VNFs across thousands of network cloud sites formed by morphing central offices, customer premises, etc., into virtualized centers.

Orchestrator, Policy Engine, SDN Controller, etc. deployed across multiple sites.

Reference architecture for edge cloud.

Federated Akraino/
Edge Infrastructure Stack Controllers each managing hundreds of edge sites.

Virtualize the RAN and control it centrally.

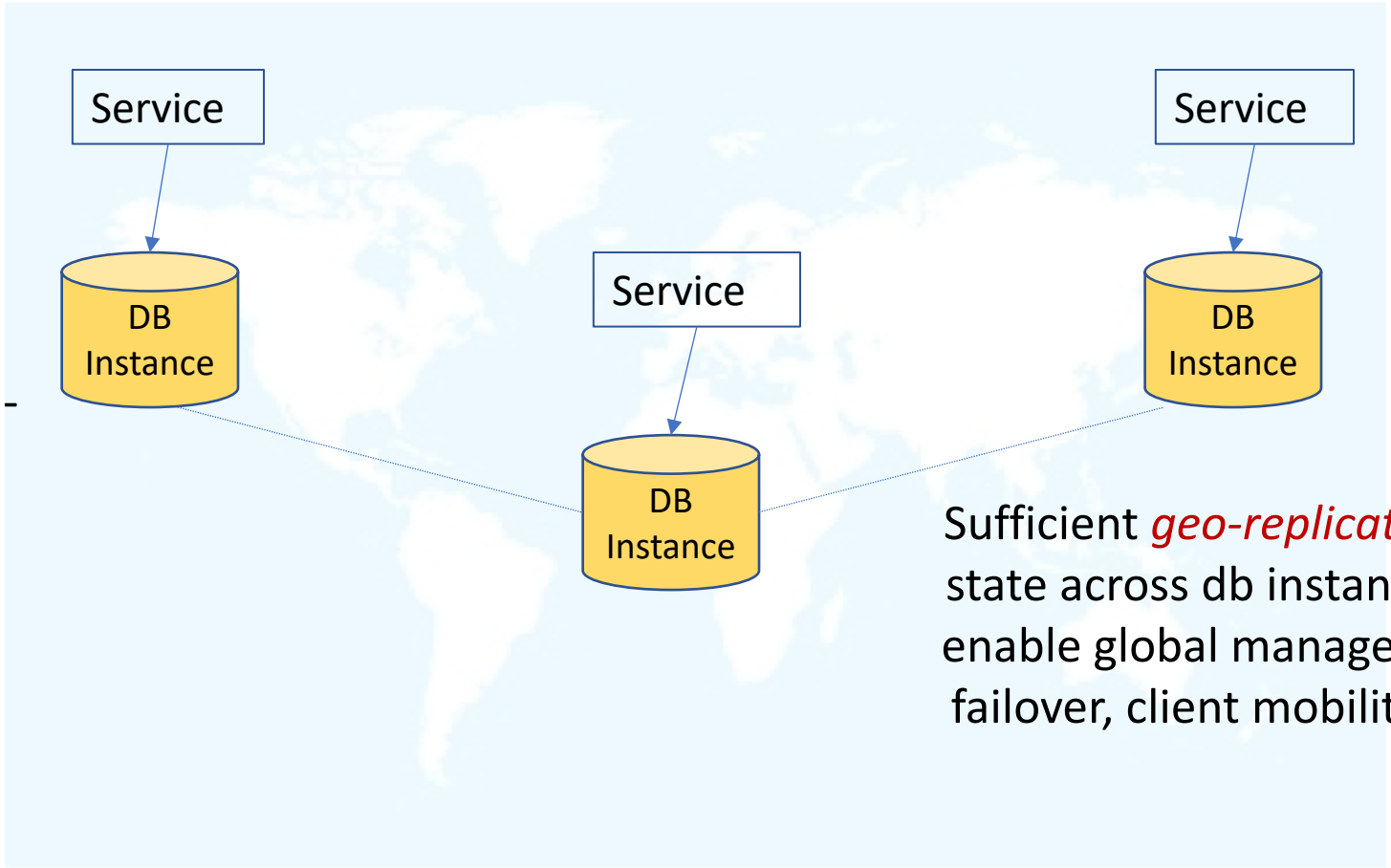
Controllers managing thousands of cell sites.

These software components have state and need to be fault-tolerant, highly available and performant.

Existing state management solutions are perfectly suited to manage state within a site or across a few sites. What about edge scale?

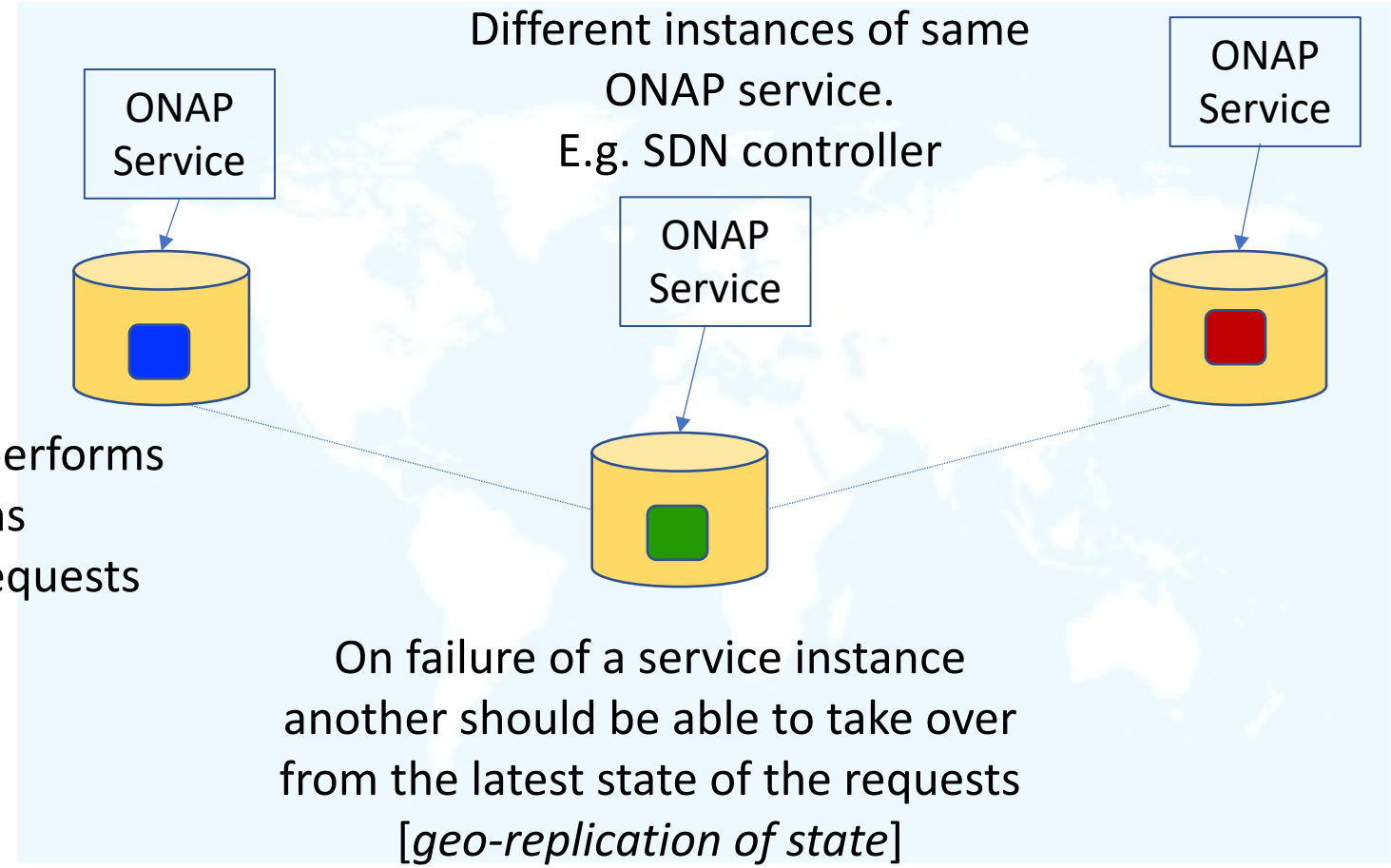
Basic Requirements of an edge-scale state management service

Efficient performance for reads and writes at local (or within data-center/site) db.



Sufficient *geo-replication* of state across db instances to enable global management, failover, client mobility etc.

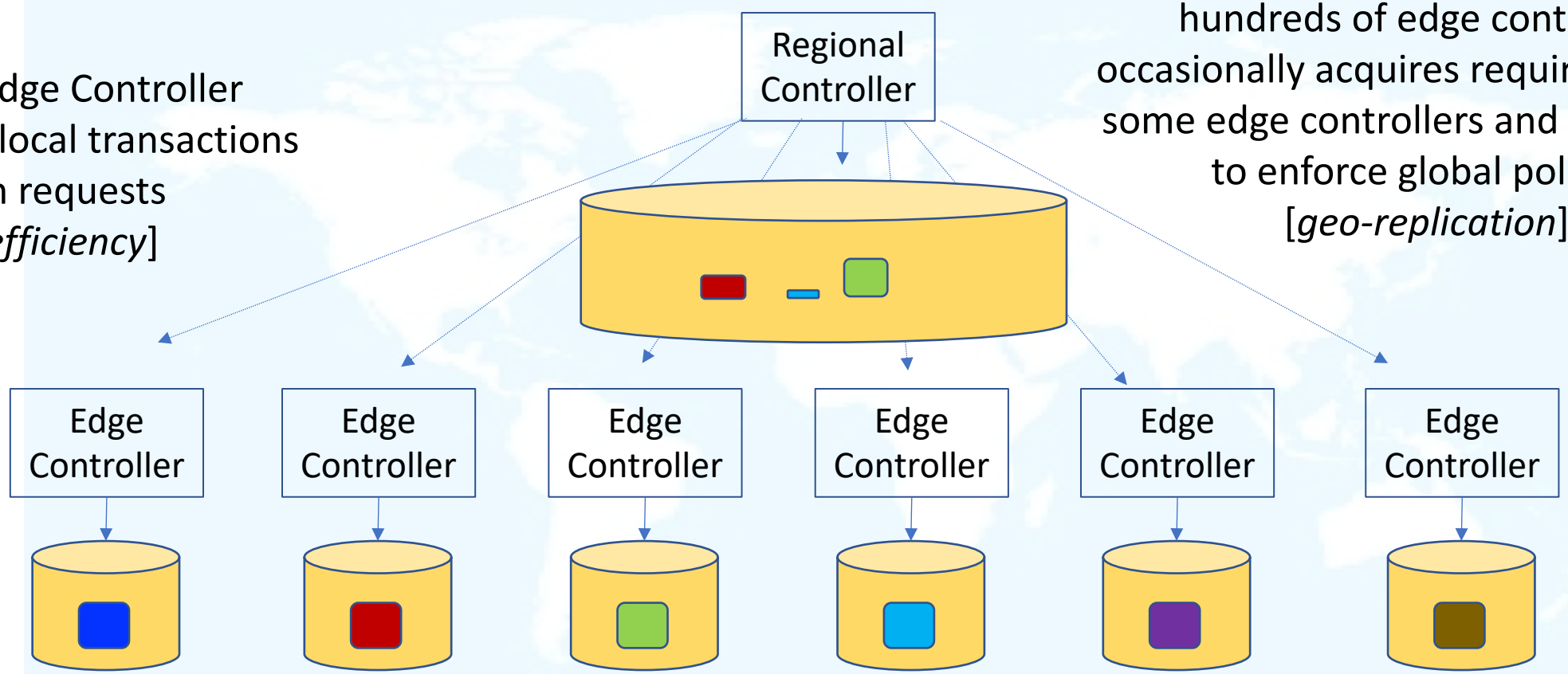
Example 1: Active Replication with Failover for ONAP Components



Example 2: Federated Controllers for Akraino

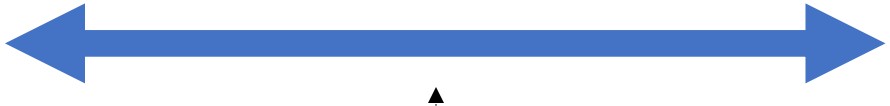
Each Edge Controller
Performs local transactions
on requests
[*efficiency*]

Regional Controller (s) each controlling
hundreds of edge controllers
occasionally acquires required state of
some edge controllers and modifies it
to enforce global policies
[*geo-replication*]



Problem: Finding the right balance between efficiency and geo-replication semantics

High efficiency but weakest cross-site guarantees. E.g. PostgreSQL, Cassandra async replication across sites – cannot obtain latest state for failover, global management



Strongest cross-site guarantees but costly protocols for geo-replication (2 PC, distributed consensus). E.g. Zookeeper, Fully transactional MariaDB Gallera, Spanner, CockrachDB,

Open questions in this quest for balance

- What are the right semantics for a state management service?
- How do basic assumptions on consistency change at the edge?
- What design patterns/recipes can we provide that enables better state management?
- Is there a CAP theorem for the edge?