Multi-site State Coordination Service (MUSIC)

ONAP Policy Meeting 6/6/2018

Presented by Bharath Balasubramanian

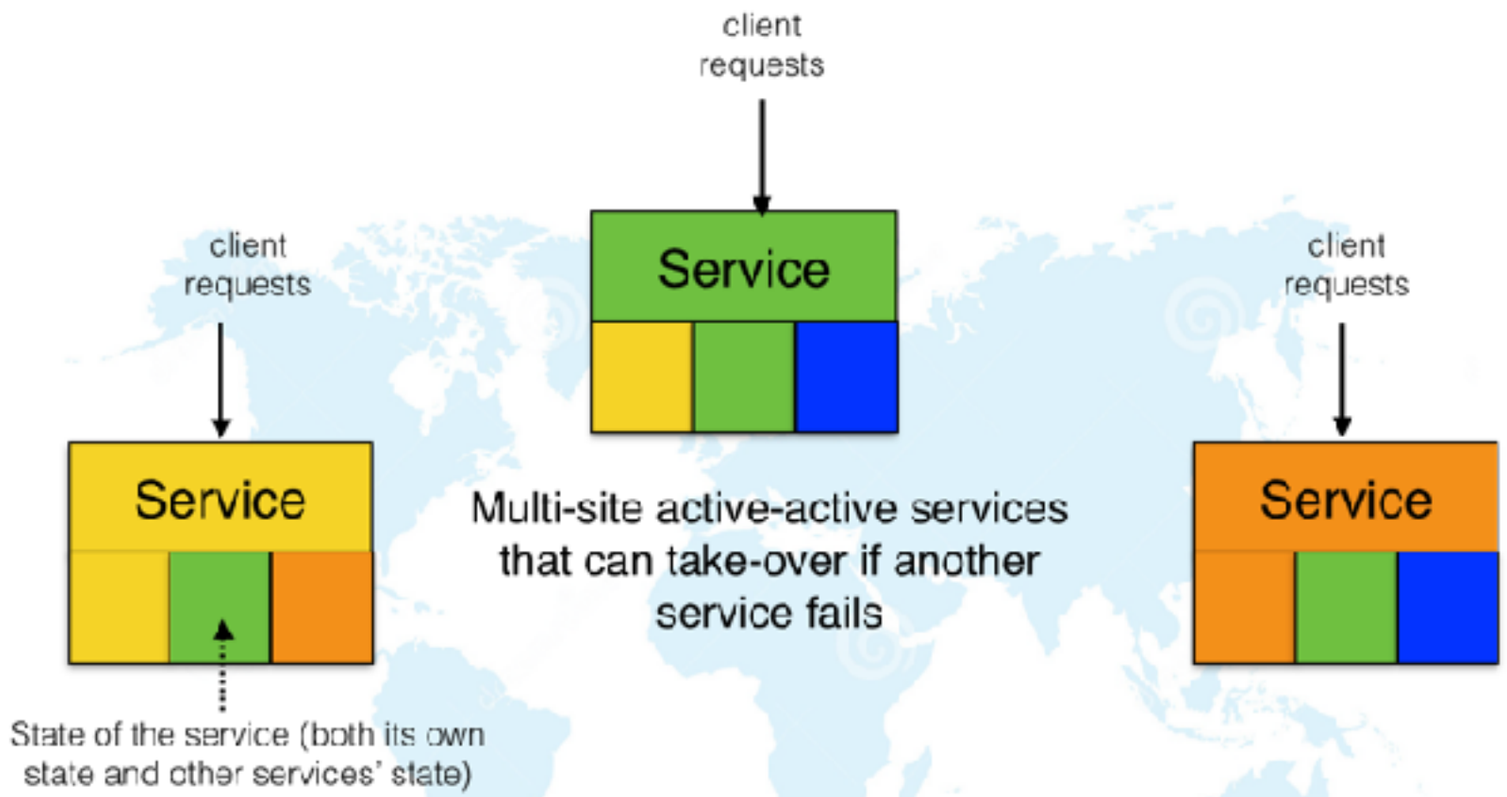
Goal

A common state-coordination/management platform (MUSIC) to build VNFs with 5 9s of availability on 3 9s (or lower) software and infrastructure in a cost-effective manner.

MUSIC Team

Responsibility	AVP	People
ONAP	NA	Bharath Balasubramanian (PTL, ATT), Viswanath Kumar Skand Priya (Verizon), Abbas Fazal (ATT), Thomas Nelson Junior (ATT), Greg Waines (Windriver), Brendan Tschaen (ATT)
Overall leadership and vision, D2 and ONAP evangelization	Oliver Spatscheck	Bharath Balasubramanian, Kaustubh Joshi
Design, architecture and prototypes	Oliver Spatscheck	Bharath Balasubramanian, Pamela Zave, Richard Schlichting, Shankar
Core development team	Gerald Karam	Abbas Fazal, Brendan Tschaen, Garima Mullick, Thomas Nelson, Chinnamma Charalel, Vaibhav Isanaka, Vikram Potturi, Inam Soomro, Srupane Kondreddy,
Integrating into ECOMP common software platform	Gerald Karam	Heather Robinett, Michael Howe
Benchmarking and testing	Catherine Lefevre	Chetan Doshi, Leonardo Bellini, Andrea Chiappa
Customers on board	Abbas Fazal and Gerald Karam (HAS-ATT/ONAP, Portal-ATT/ONAP, Valet (ATT), Sharon Chisholm, Jim Mount and Richard Trabedski (Amdocs POC for SDN-C-ATT)	

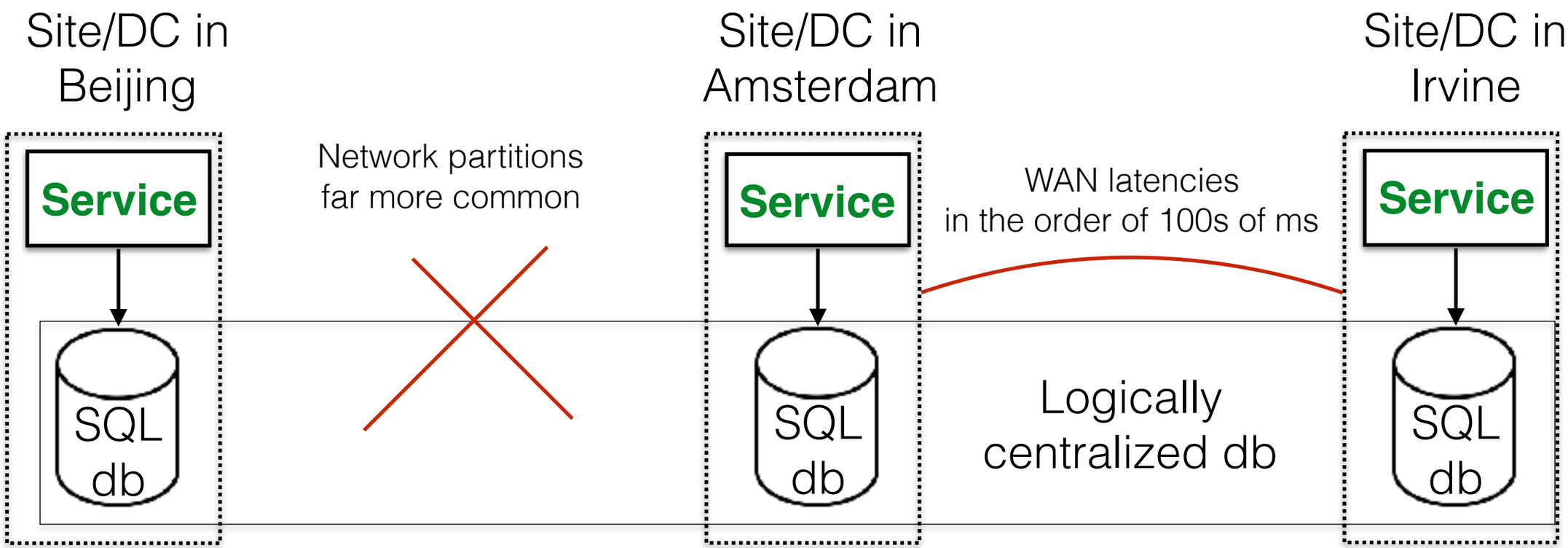
To achieve 5 9s availability network services (ONAP, VNFs, Edge/IoT services) need to support multi-site, active-active services with efficient failover.



Need of the hour

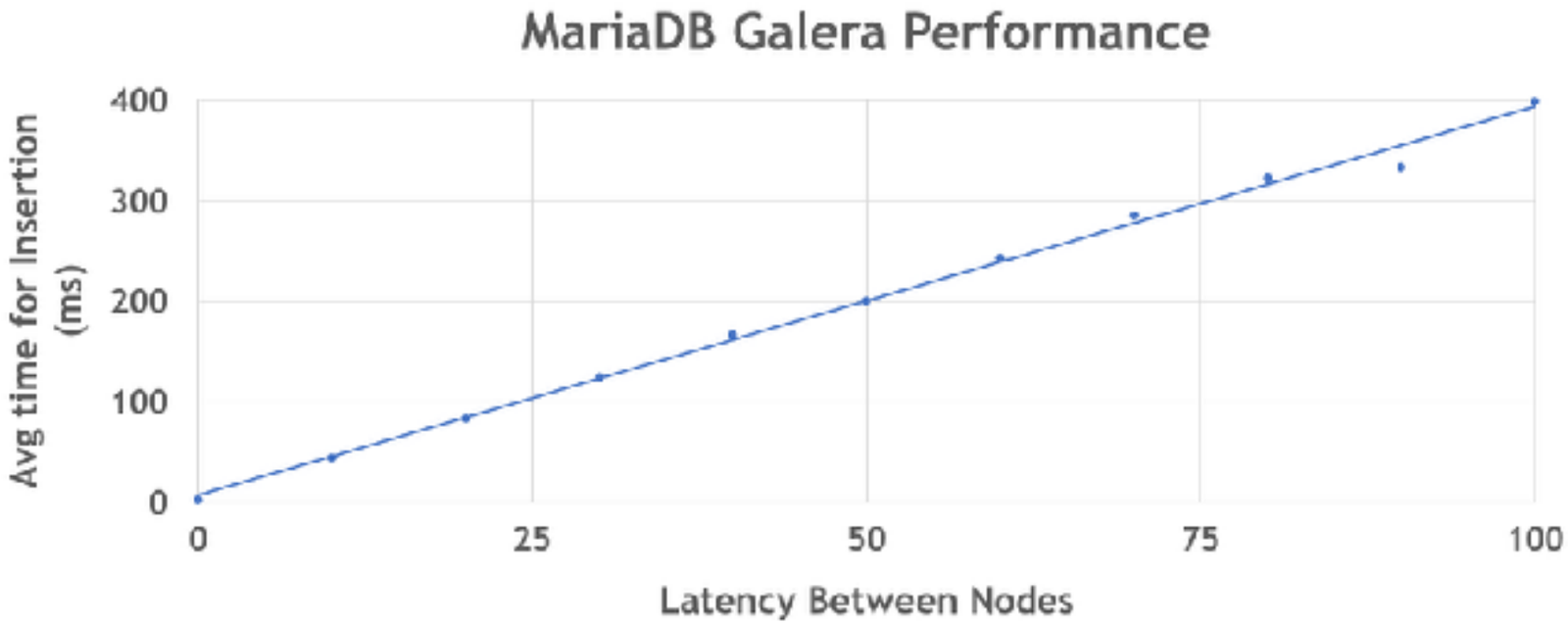
- Manage state of services across thousands of geo-distributed sites.
- Provide resiliency/coordination protocols to partition state across replicas and ensure correct and efficient failover of state during site-failures and site-partitions.

Current Practice: May not scale.



Rooted in the philosophy "the single-site solution should more or less work across geo-distributed sites". E.g. attempts to use MariaDB clustering as is across sites — may not scale and/or allow partitioned operation!

E.g: A preliminary study of MariaDB scalability across the world.



Simple 3 node cluster with latency using netem — nearly half a second average insertion time with 100 ms inter-node latency (typical of a US-East, US-West, Europe deployment of ONAP). Will be worse when we consider Irvine, Amsterdam, Beijing with > 200 ms latency.

Current Practice: Often wasteful and erroneous.

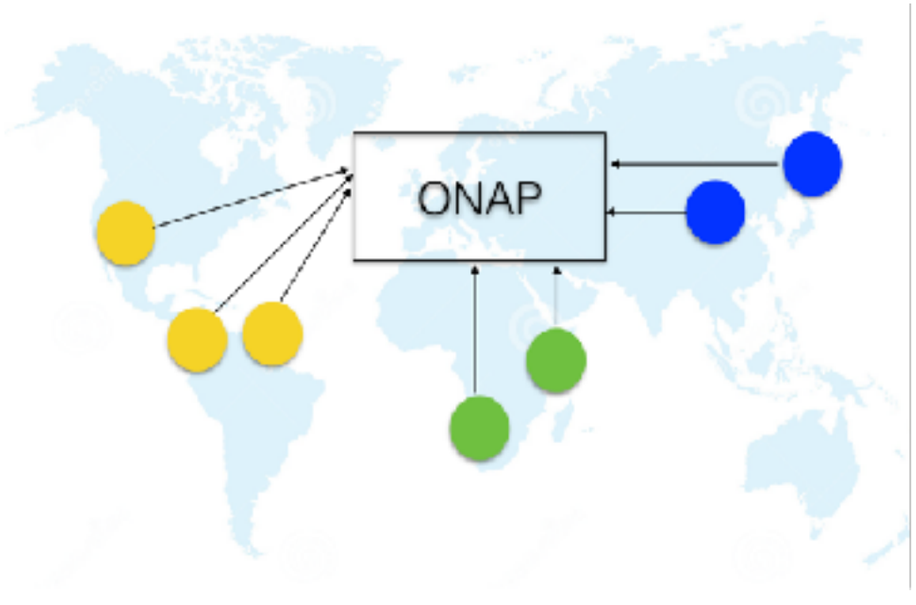
Each team building its own solution — wasteful and can often be erroneous due to complex distributed protocols, *replete with corner cases*. E.g.

- How do you access state in a mutually exclusive manner when required despite failures/split brain issues?
- Does the new active have the latest information on failover?

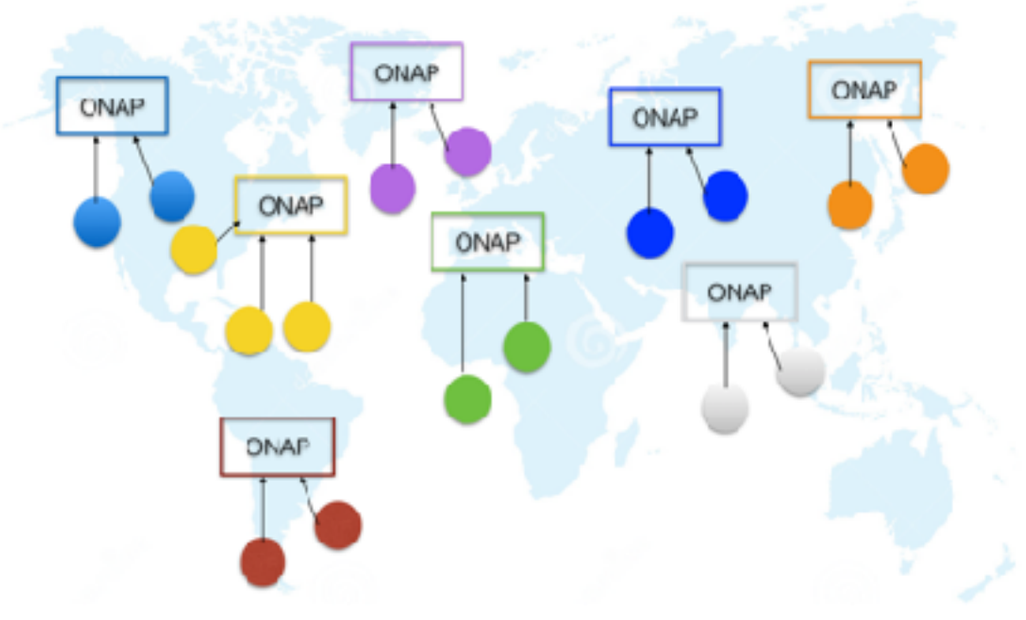
Current Practice: Does not address future needs.

Most current designs barely address active-active needs, and mostly ignore *federation* that is crucial for IoT/Edge.

E.g. ONAP for Edge/5G and in fact the entire network!



Monolithic design of
ONAP to control VNFs will not scale to the edge.



Federated design of
ONAP to control VNFs is much more practical.

Our solution: MUSIC

- A common state-management platform for ONAP components/micro-services specifically tailored for multi-site geo-distributed replication and federation.
- Provide rich resiliency/coordination recipes on top of MUSIC that ONAP components can simply configure and use — each team need not re-invent solutions.

Base solution

MUSIC maintains replicated state in an eventually-consistent data-store (Cassandra) wherein the access to the keys can be protected using a strongly consistent locking service (Zookeeper/Consul/etcd).

Why is it unique?

Despite site failures or network partitions, MUSIC guarantees that the lock-holder to a key always reads and writes to the latest value of the key (formally called entry-consistency*) - no other tool today achieves this.

Formally verified MUSIC protocols using model checking.

* Under review at the Principles of Distributed Computing Conference (PODC 2018)

Why is it useful?

Through the MUSIC abstractions and entry-consistency property, ONAP components/ services can achieve fine-grained flexible consistency on their replicated state across sites — acquire lock only when you need strong consistency.

Provides a natural starting point for federation — the lock holder for a bunch of keys owns the state corresponding to those keys.

Recipes

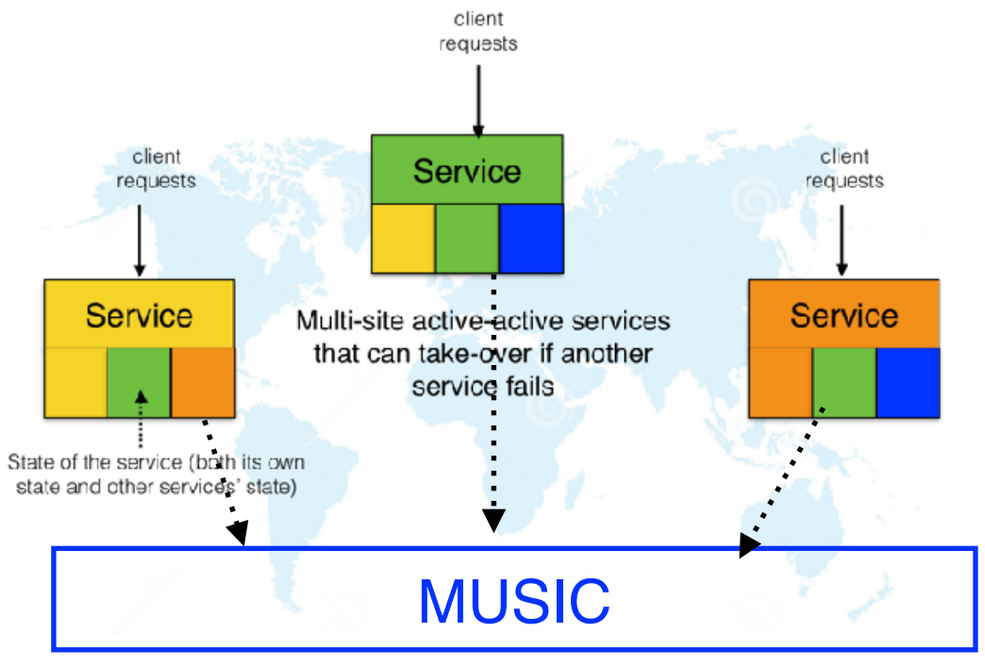
- mdbc: A plugin for components to migrate seamlessly from SQL usage to MUSIC
- prom: Policy driven ownership management of state to partition and failover state in a consistent manner
- musicCAS: Distributed compare and set across keys to perform atomic updates
- musicQ: A queue API across sites in which key management is carefully done to ensure efficient sorting

MUSIC Vision of federation

- MUSIC ensures through prom that each service replica is the “owner” for a particular set of state corresponding to certain requests and assigns a few backups for this state (completely policy driven) on other sites

- The owner can now update state locally either directly to MUSIC or through mdbc for SQL-based components doing only quorum writes to other sites (strong consistency without per operation locking!)

- On site failure prom will transfer state ownership to another service on a different site and ensure the new owner has latest state.



Service = Any ONAP component/micro service/VNF

MUSIC Implementation

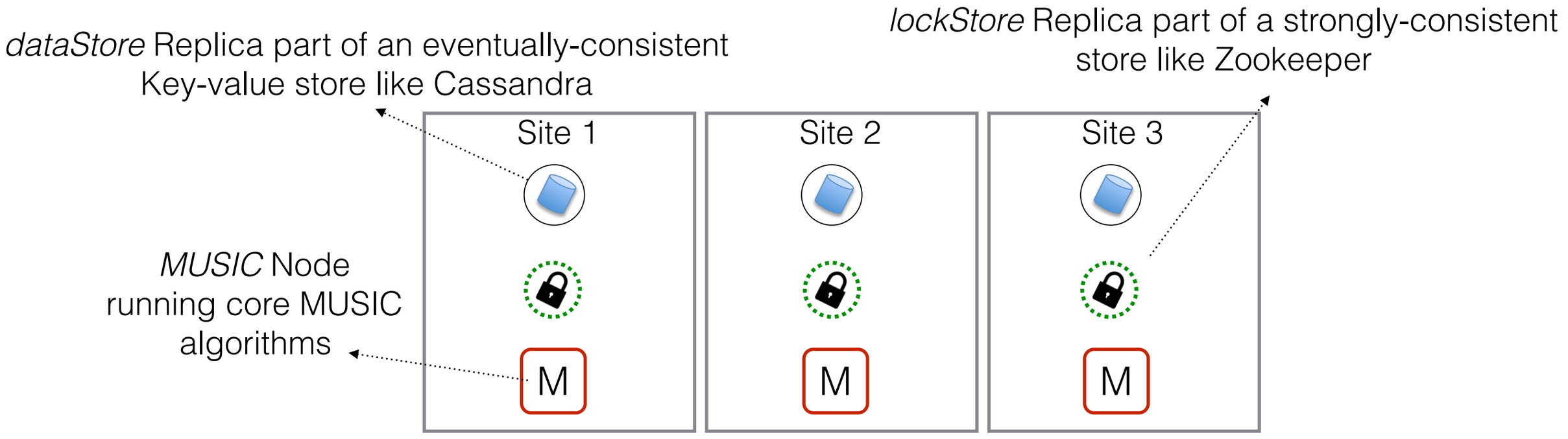
Thin shim layer (<10,000 lines) over two production tested open source tools — Apache Zookeeper and Cassandra provides you with 5 9s of availability!

MUSIC Status

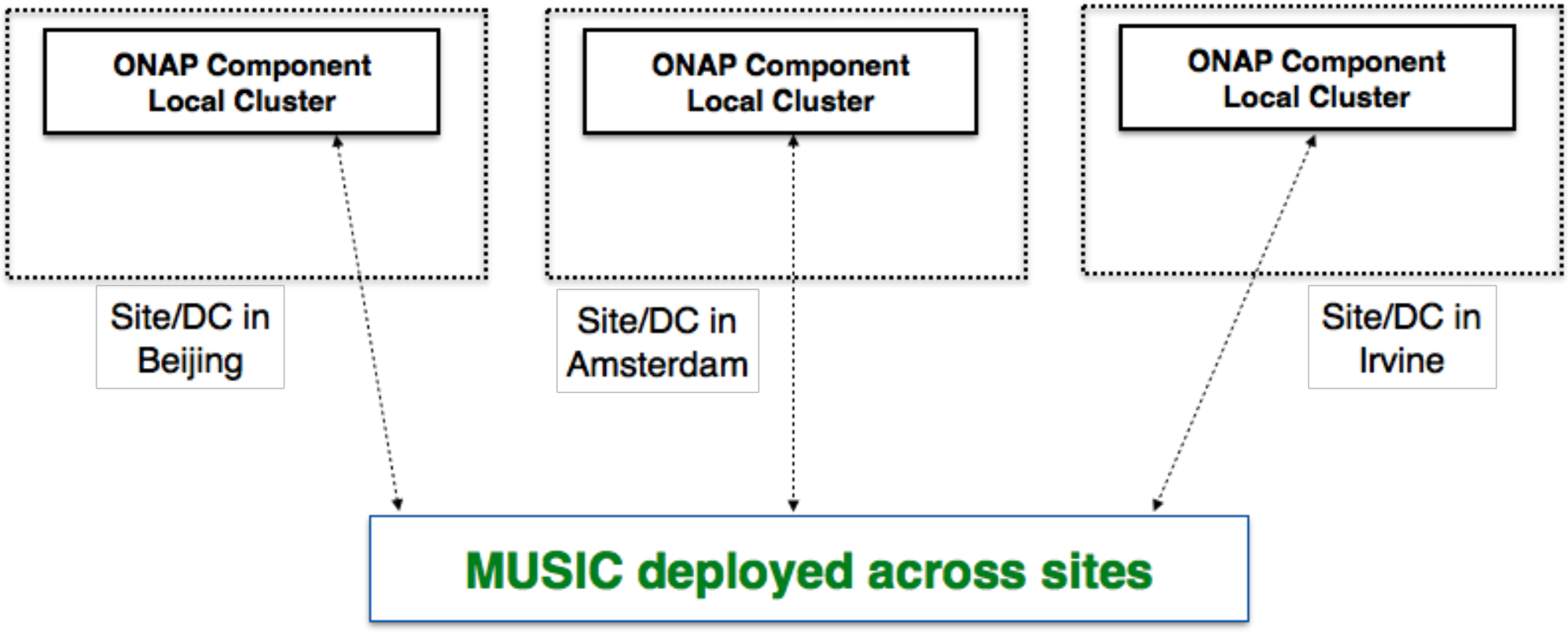
Use-Case	Status
ECOMP	• MUSIC in production 17/10 as the state management service of the ECOMP Homing Service (HAS) for SD-WAN. • Currently integrating with Portal for 18/06 • SDN-C POC with Amdocs for 18/06 • Working actively with TechDev partners to make this the resiliency platform for all ECOMP services (18/19/xx)
ONAP	• Official ONAP Project recommended by architecture subcommittee for multisite state management • R2 Beijing Release commitment from ONAP HAS and ONAP Portal
VNFs	Socialized the tool with Roman Pacewicz's team to solicit VNF
IoT/Edge Computing	Working with University partners to design and implement massively geo-distributed state-management architectures for IoT/Edge use-cases.

MUSIC basics

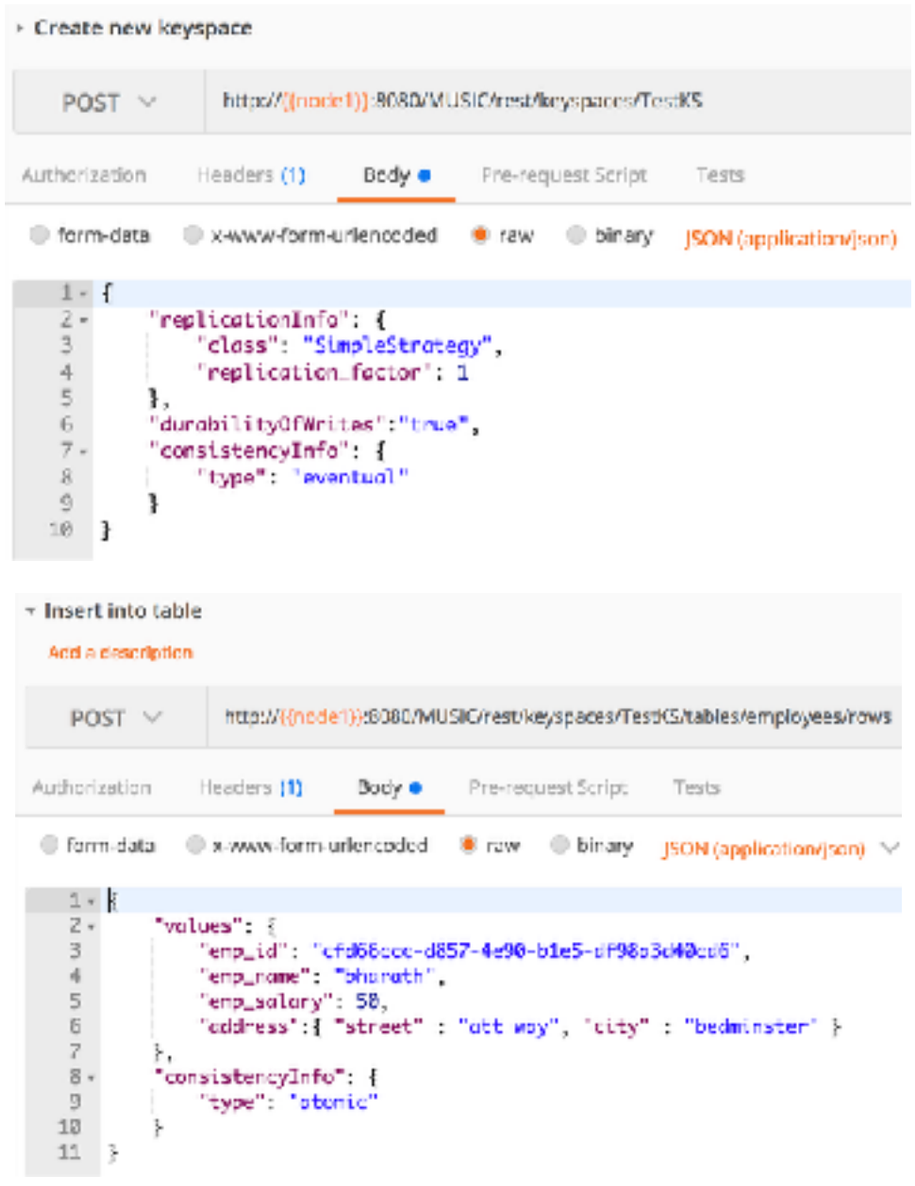
Architecture

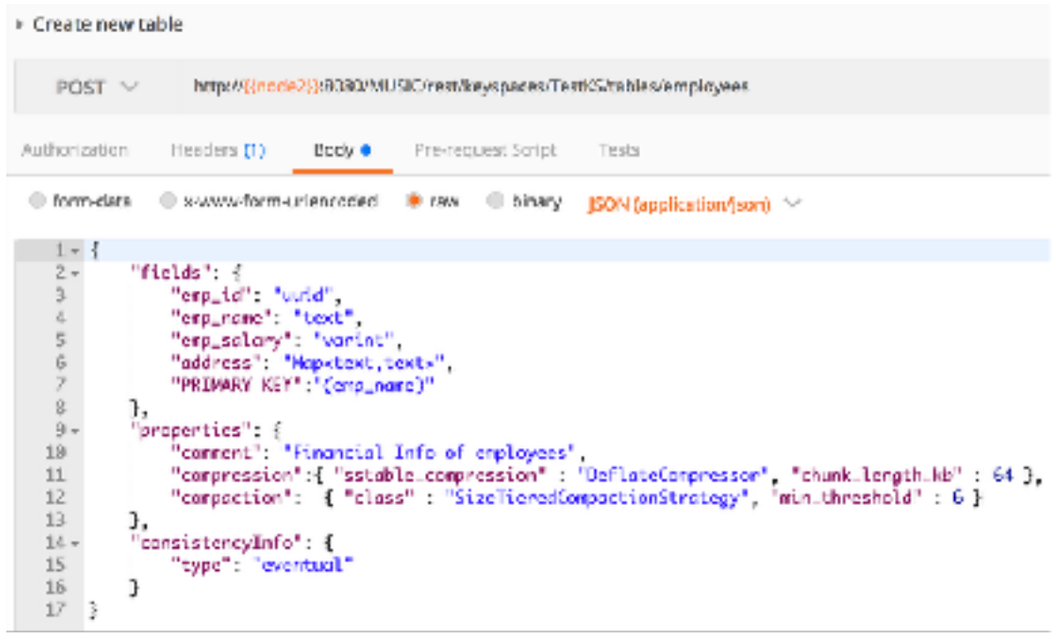


Usage


ONAP Component
Local Cluster
ONAP Component
Local Cluster
ONAP Component
Local Cluster
Site/DC in
Beijing
Site/DC in
Amsterdam
Site/DC in
Irvine
MUSIC deployed across sites

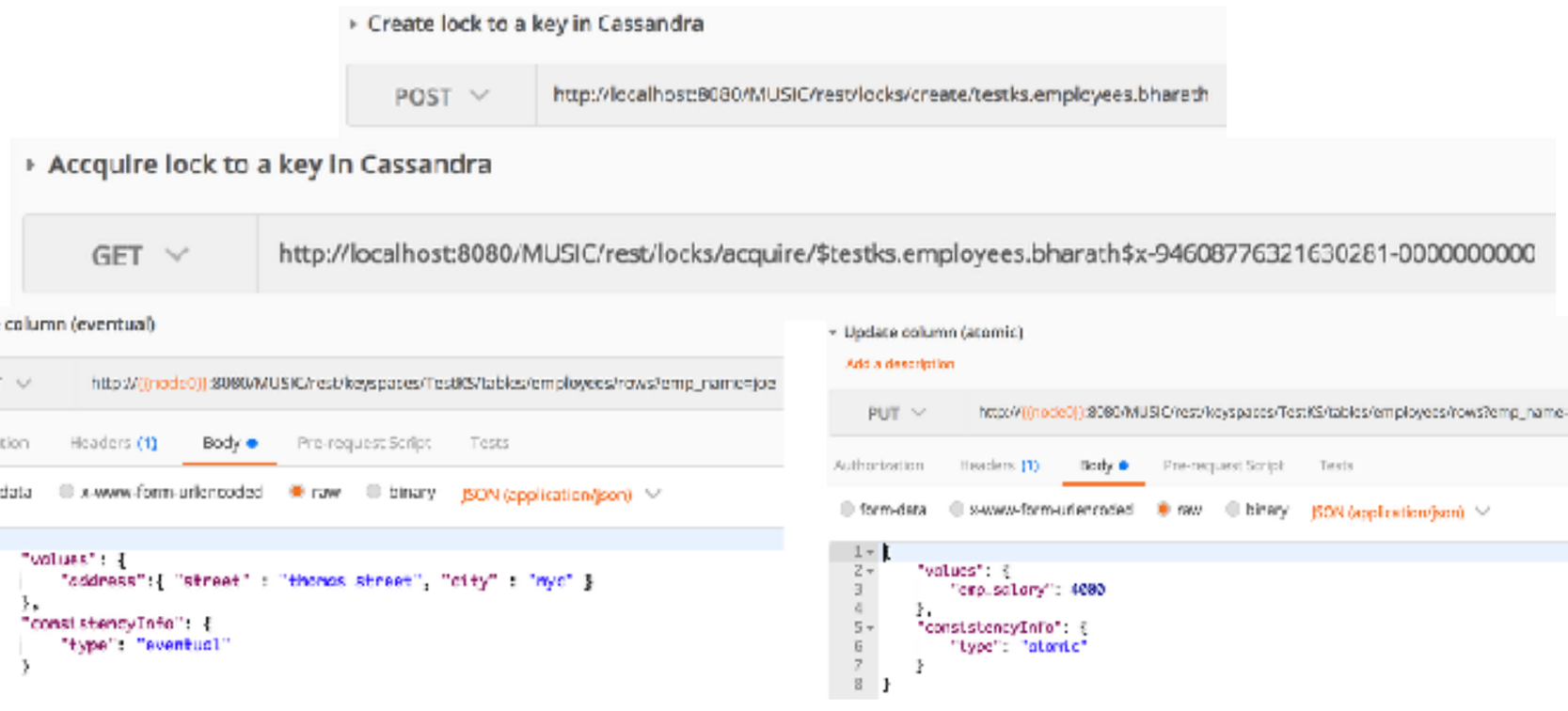
Data API





The basic REST API provides a REST+JSON wrapper around the standard Cassandra API. While this is useful in itself, there is more…

Locking API



The novel locking API allows the client to create locks on keys (that guarantees mutually exclusive access). Further, the client can chose between eventual operations (no locks) and atomic operations on keys (uses locks in a critical section)!

mdbc: multi-site database cache

mdbc Basics

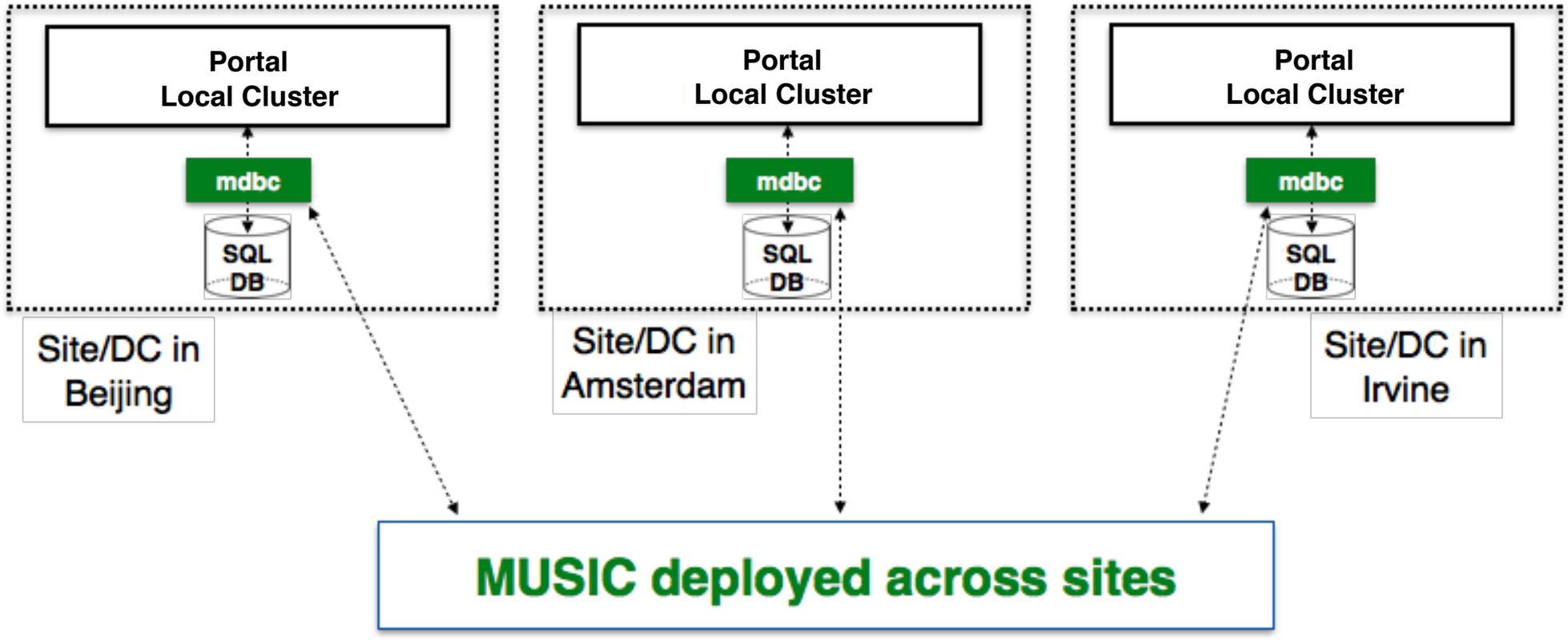
A plug-in for SQL-based components to seamlessly use MUSIC with no change to SQL code.

Ensures transactional guarantees within a site with per-key choice of sync or async replication across sites.

mdbc Design

- mdbc = local sql db + multi-site MUSIC deployment
- Service replicated across multiple sites; writes to and reads from the local mdbc sql database
- mdbc captures local sql writes and propagates it to MUSIC and captures local reads and serves it from MUSIC

mdbc Usage



SQL DB = within site
MariaDB Gallera cluster for Portal

mdbc config

Before:

```
//Register JDBC driver
Class.forName(“com.mysql.jdbc.Driver”);
//Open a connection
Connection conn =
DriverManager.getConnection(DB_URL,
connectionProps);
```

After:

```
//Register MDBC driver
Class.forName(“org.onap.mdbc.ProxyDriver");
//Open a connection
Connection conn =
DriverManager.getConnection(MDBC_DB_URL,
connectionProps);
```

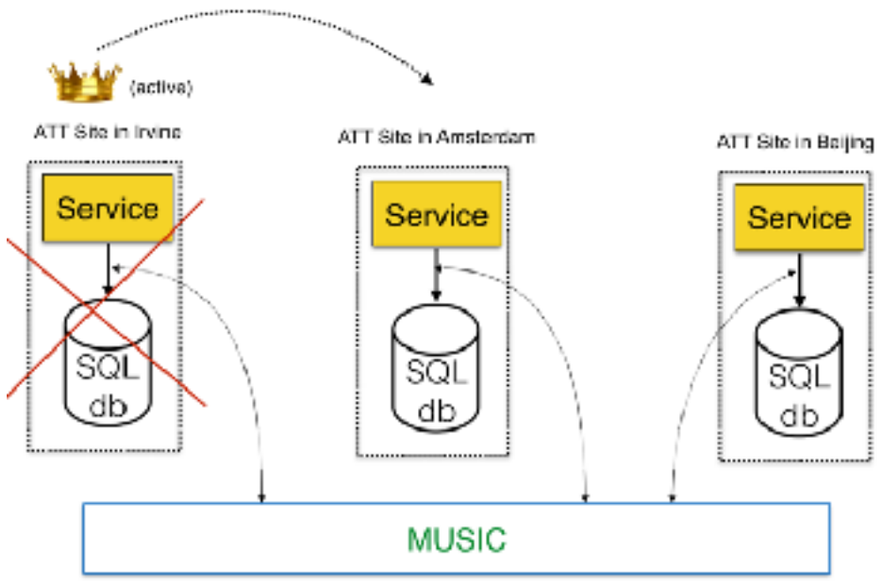
mdbc properties file:

```
music_address=1.2.3.4
music_sync_tables= task, job, process
music_r_factor=3
```

prom: policy-driven state ownership management

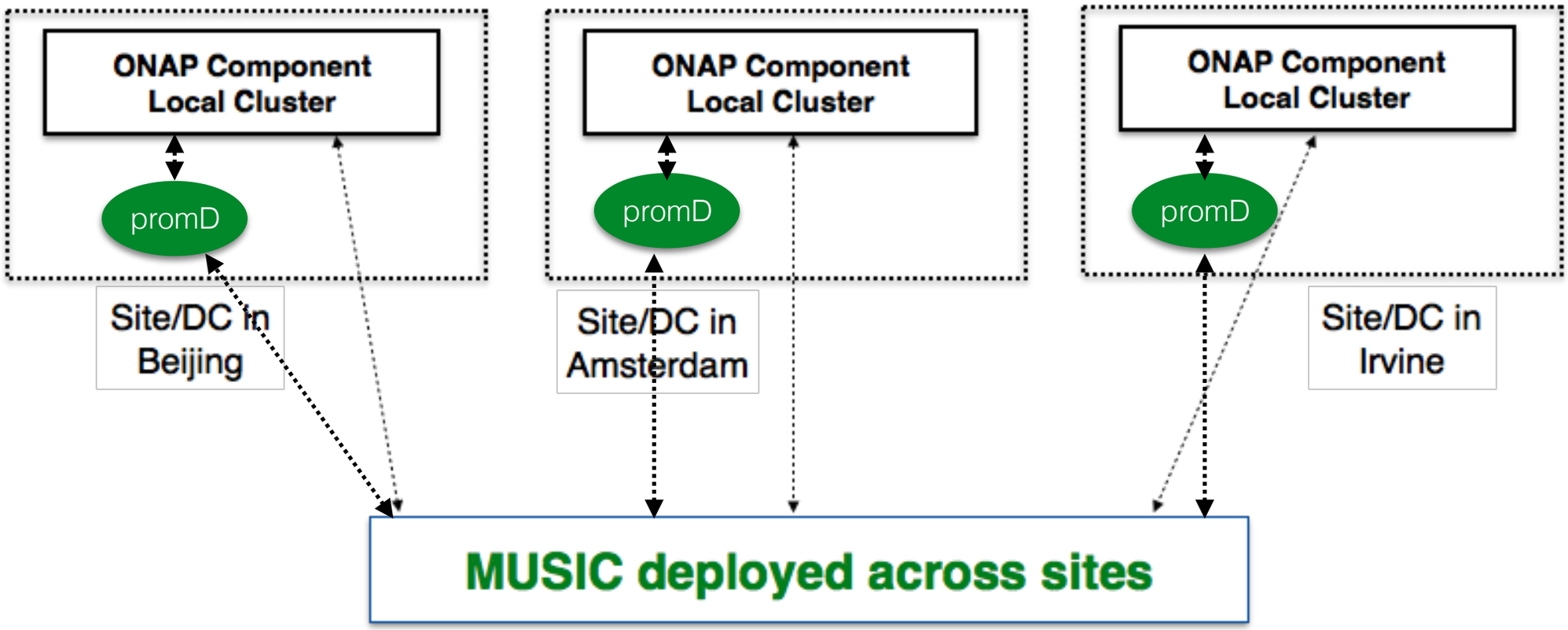
prom: policy-driven ownership management

Detecting failure of a service and transferring ownership of state to a standby on another site involves complex distributed system challenges, replete with corner cases.



PROM provides recipes on top of MUSIC that services can simply configure (policy-driven) to achieve different resiliency patterns: active-standby, active-active with failover etc.

prom Usage


ONAP Component
Local Cluster
promD
Site/DC in
Beijing
ONAP Component
Local Cluster
promD
Site/DC in
Amsterdam
ONAP Component
Local Cluster
promD
Site/DC in
Irvine
MUSIC deployed across sites

prom config

```
{
    "app-name":"sdnc",
    "aid":"389ce6ab-3aa0-47b9-9ad4-fa43cb2ce5ea",
    "namespace":"prom_sdnc",
    "userid":"promUID",
    "password":"promPW",
    "ensure-active-MT": "/home/bt054f/prom/sampleApp/ensureSdncActive.sh",
    "ensure-passive-MT":"/home/bt054f/prom/sampleApp/ensureSdncStandby.sh",
    "core-monitor-sleep-time":"1000",
    "prom-timeout":"5000",
    "no-of-retry-attempts":"3",
    "replica-id-list":[“MT", “BD“, ”NYC”],
    "music-location":[“10.127.253.18",“10.5.1.31"],
    "music-version":2
}
```