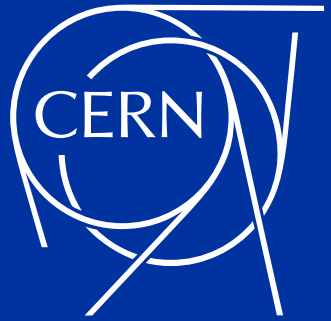


DISTRIBUTED CACHING AT CLOUD SCALE WITH APACHE IGNITE FOR THE C2MON FRAMEWORK

Tiago Oliveira

October 2021

- Introduction to C2MON
- Traditional Cache vs Distributed Cache
- Integration with Apache Ignite

Introduction to C2MON

Introduction to C2MON

Computing Center

Safety

Electricity

267k Endpoints

Cooling

QPS

Ventilation

Introduction to C2MON

Computing Center

Safety

Electricity

Cooling

QPS

Ventilation

Introduction to C2MON

Computing Center

Electricity

QPS

C2MON

- C2MON stands for CERN Control and Monitoring Platform
- Open-source platform
  https://cern.ch/c2mon/
- Industrial controls data acquisition
- Monitoring, control and data publishing
- It provides high availability and fault tolerance

Safety

Cooling

Ventilation

Introduction to C2MON

Computing Center

Electricity

QPS

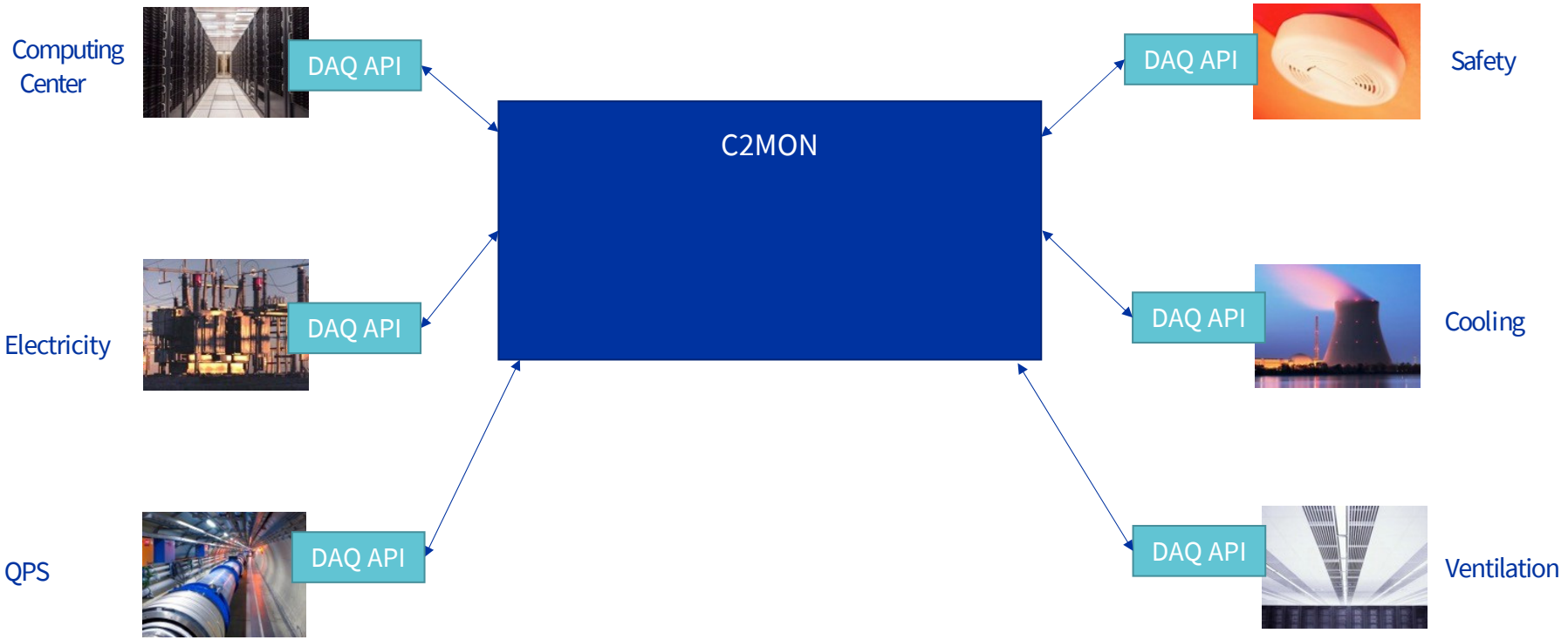
C2MON

Safety

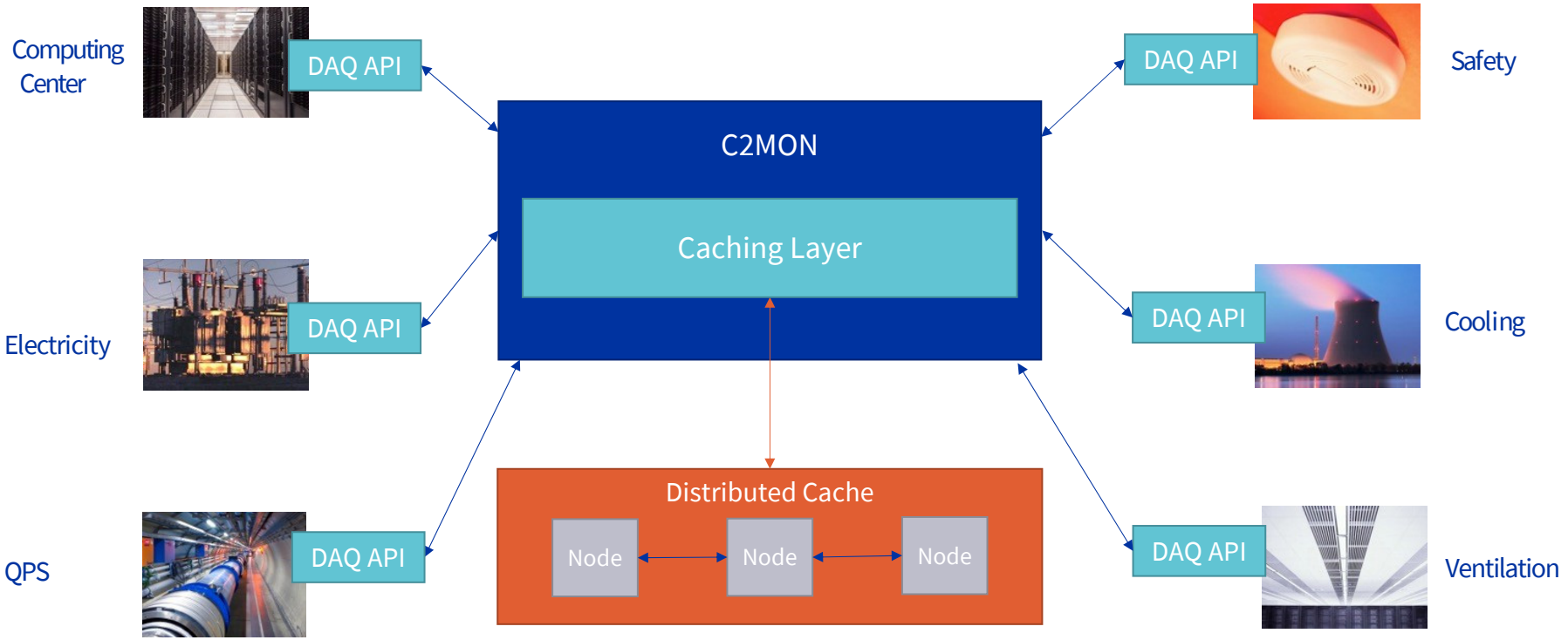
Cooling

Ventilation

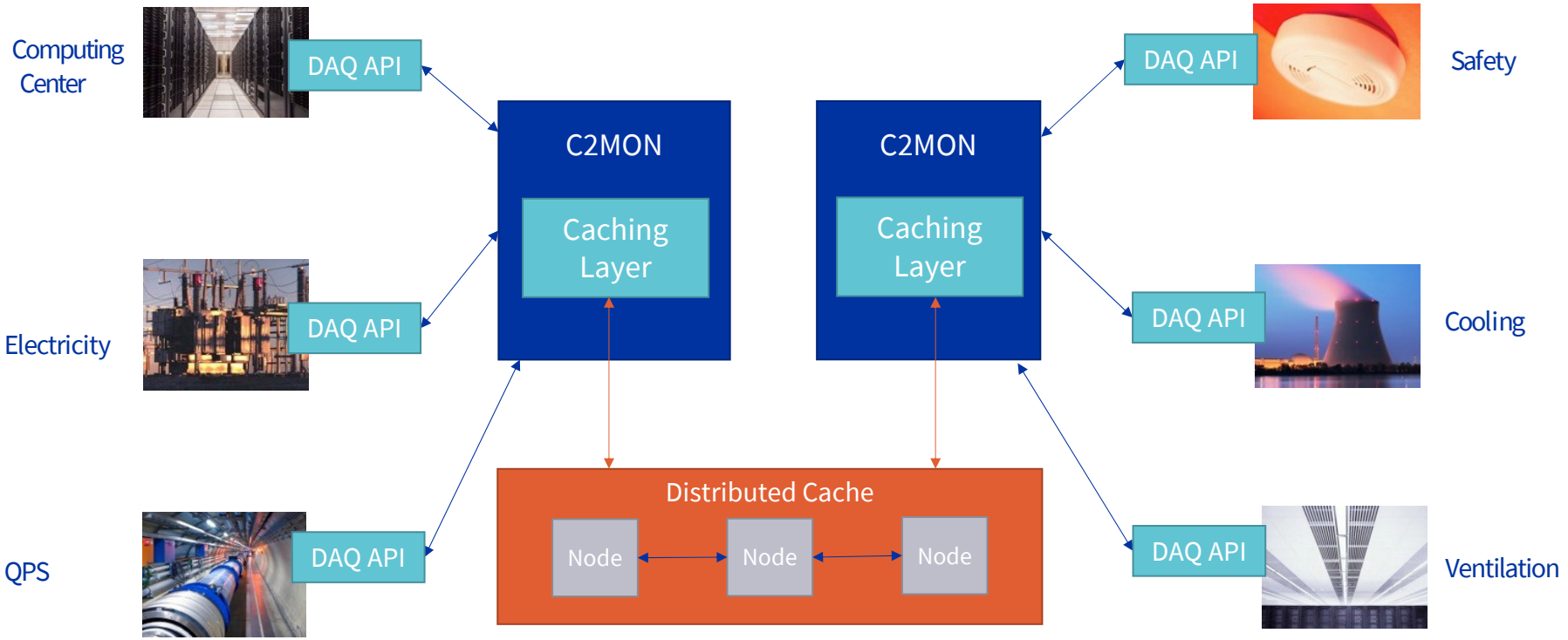
Introduction to C2MON



Introduction to C2MON

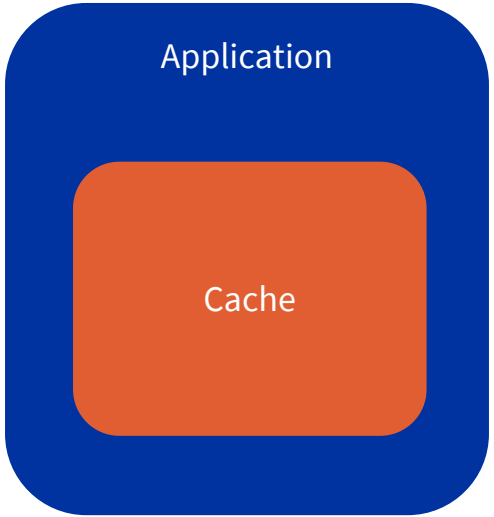


Introduction to C2MON

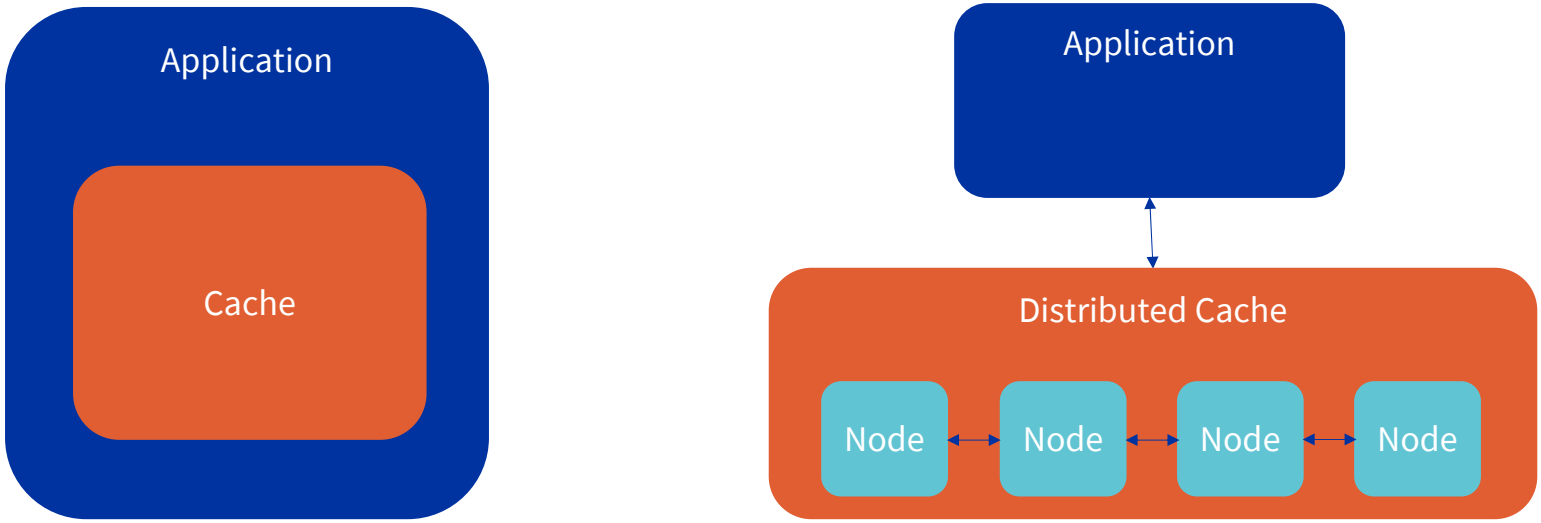


Traditional Cache vs Distributed Cache

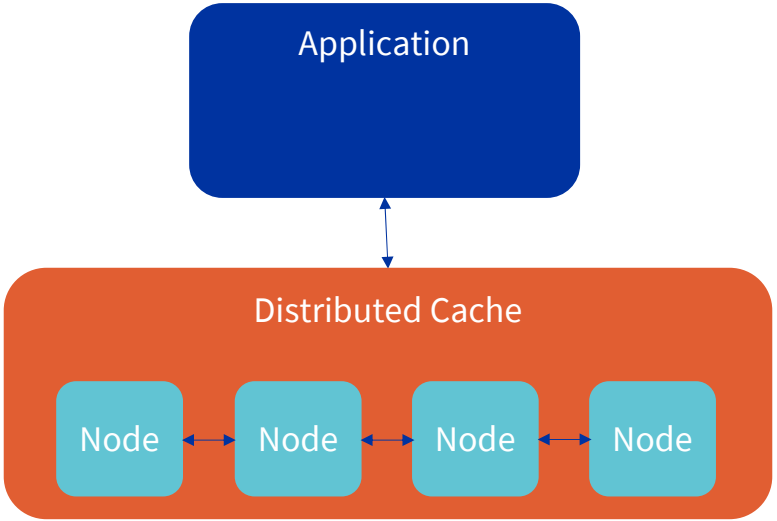
Traditional Cache vs Distributed Cache



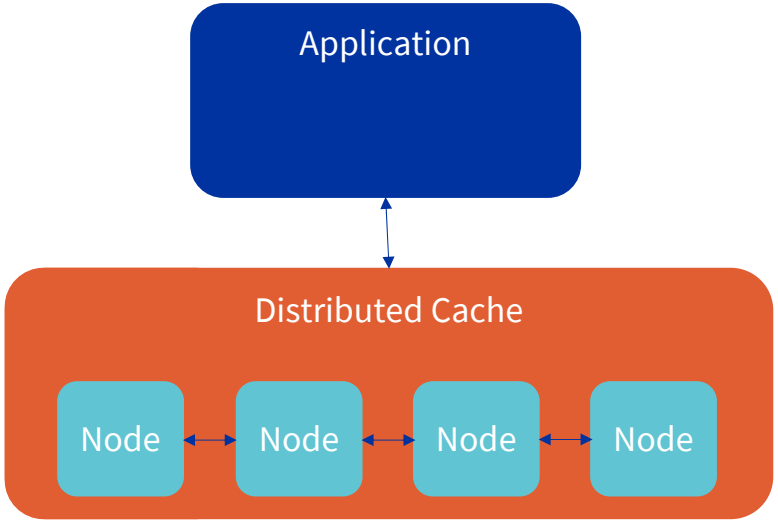
Traditional Cache vs Distributed Cache



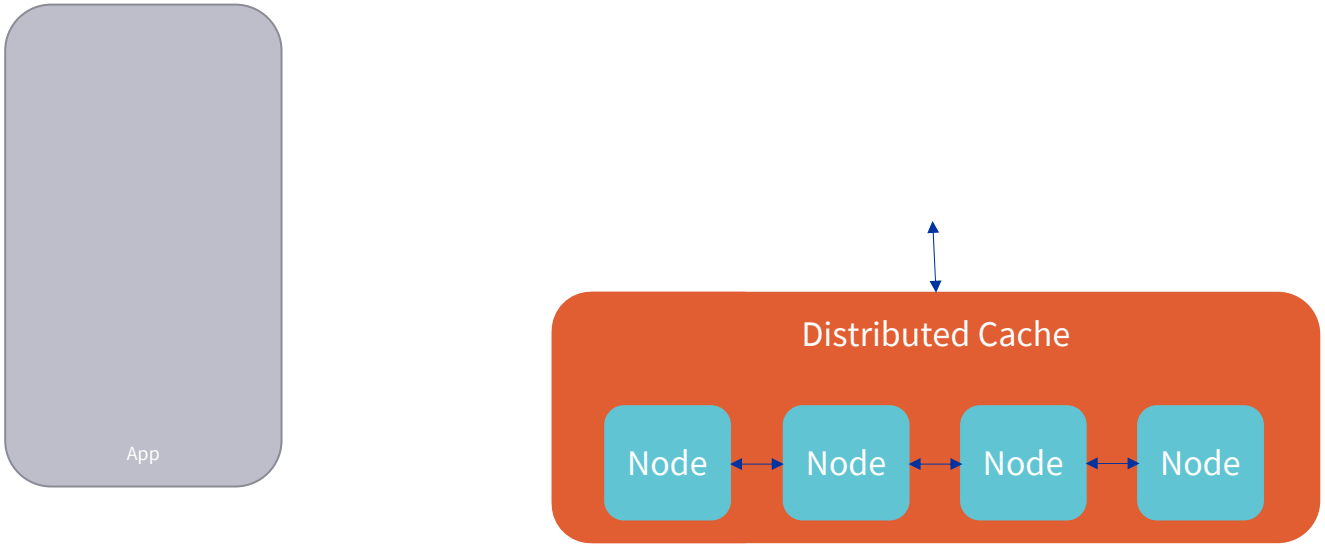
Traditional Cache vs Distributed Cache



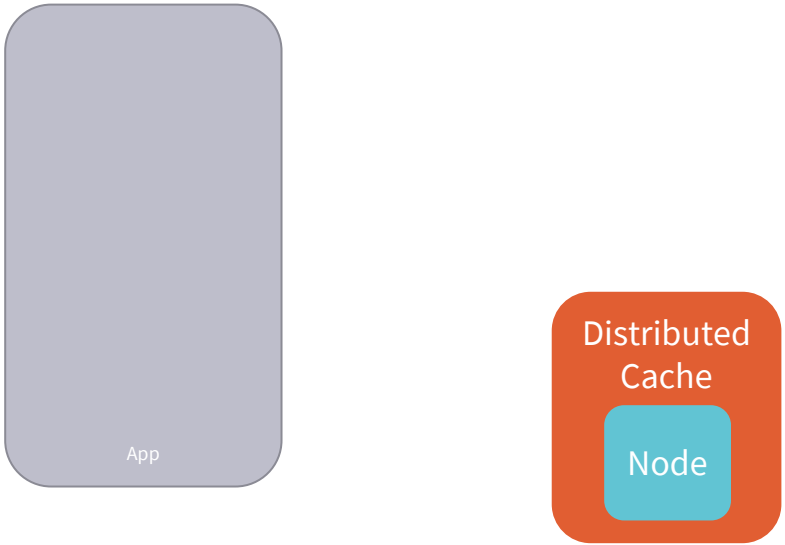
How Distributed Caches Work



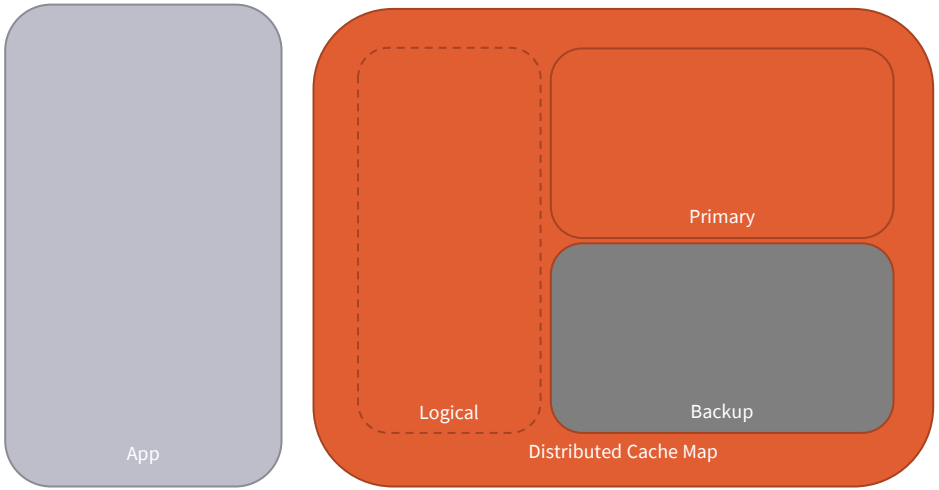
How Distributed Caches Work



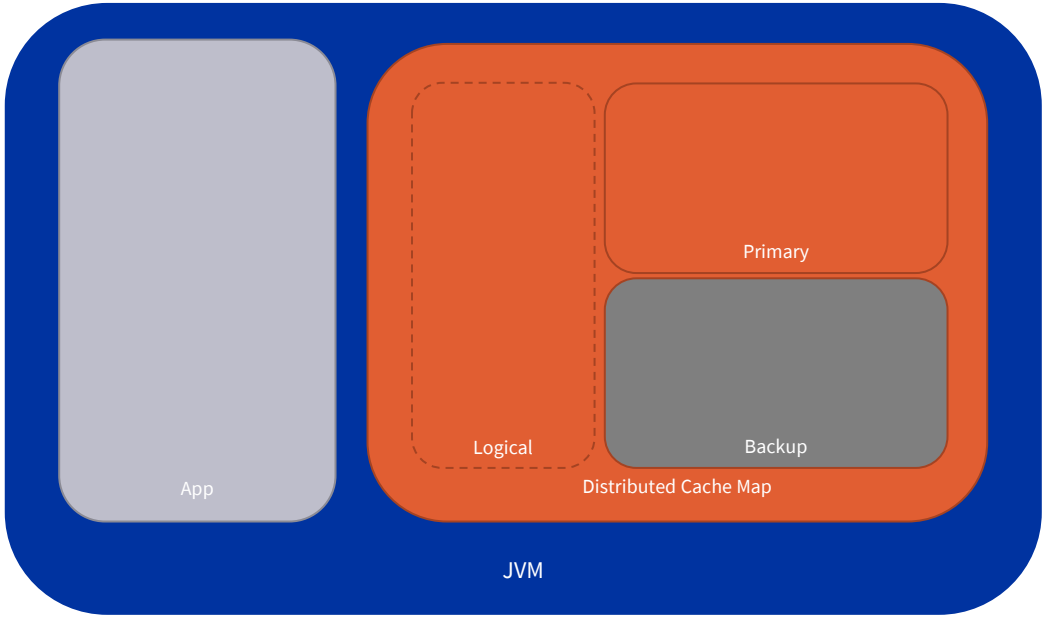
How Distributed Caches Work



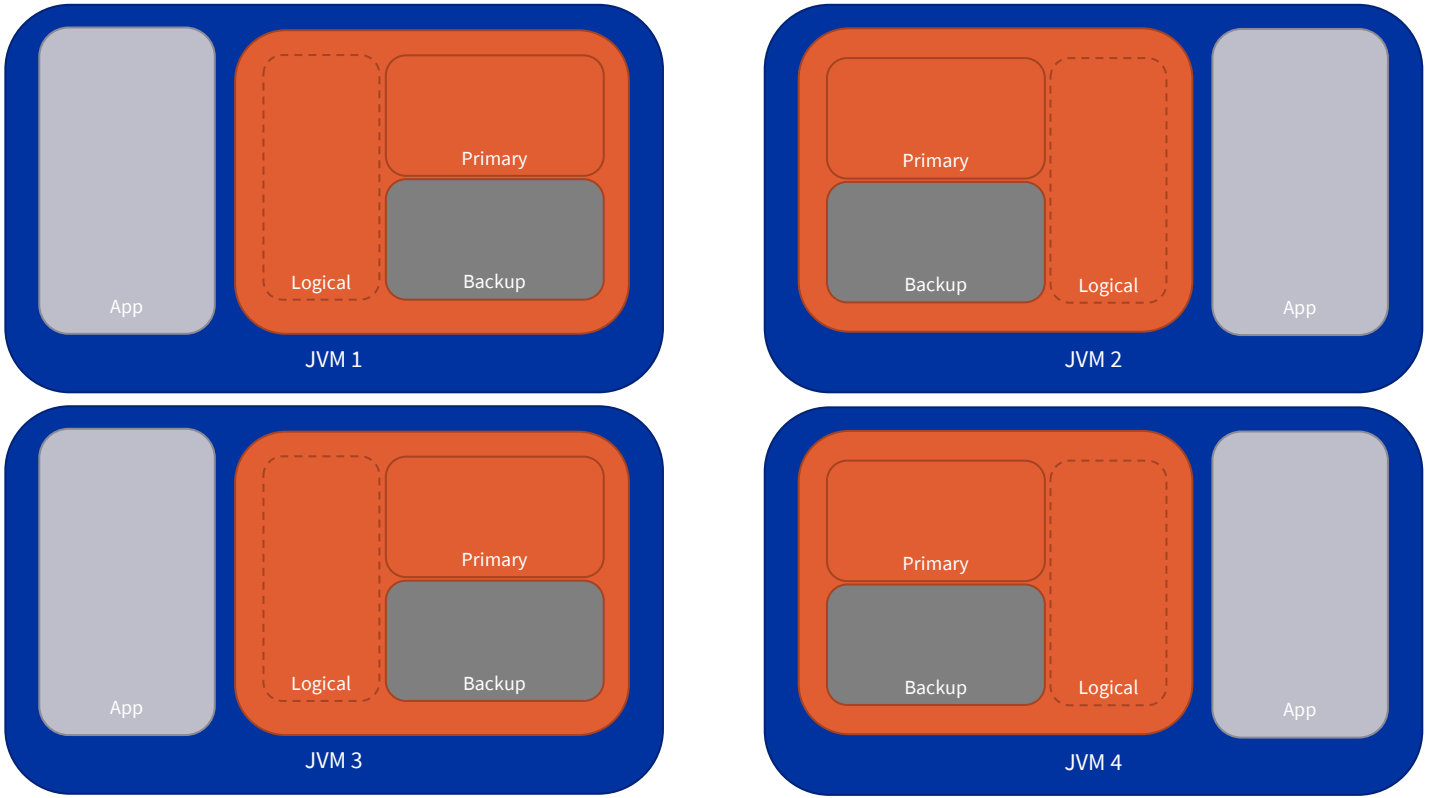
How Distributed Caches Work



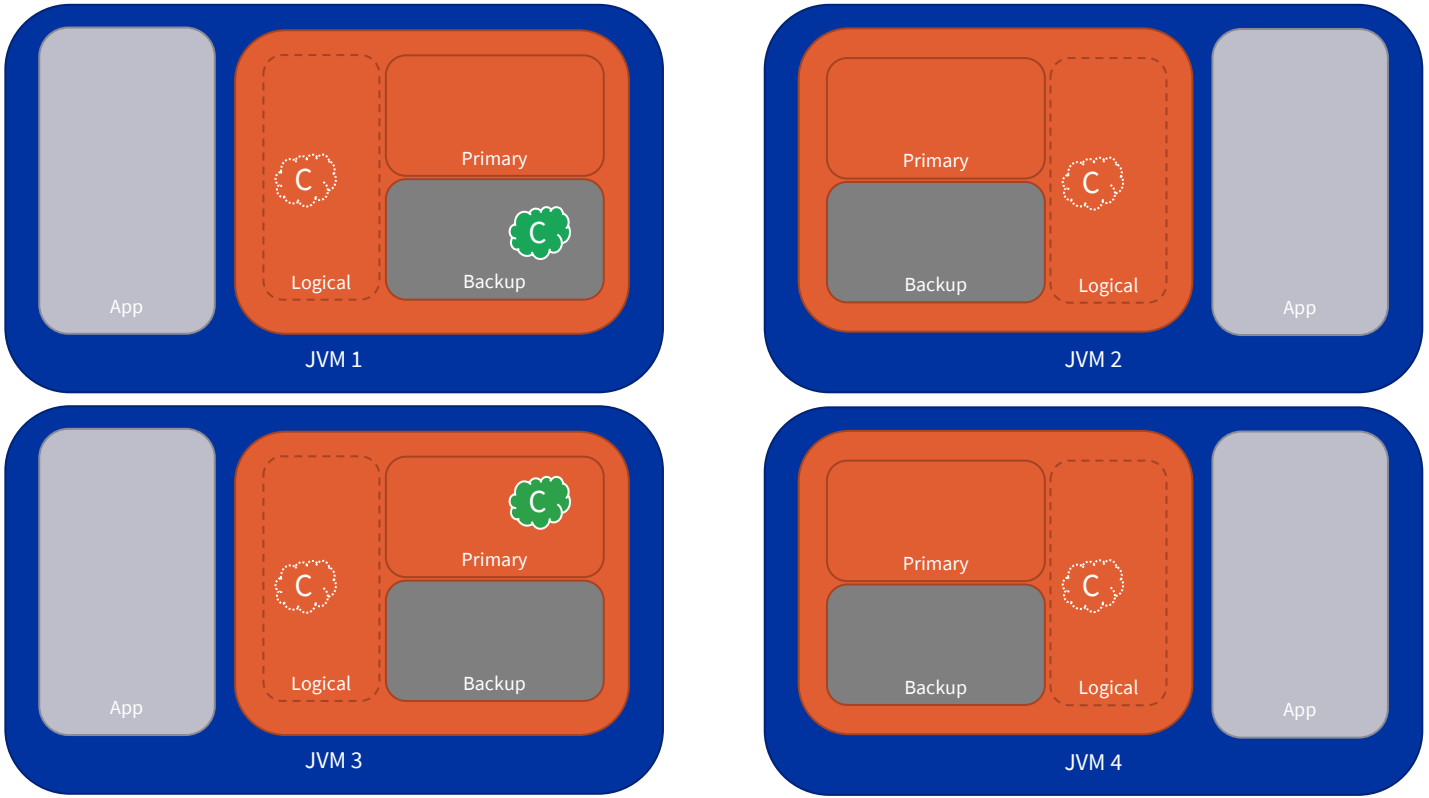
How Distributed Caches Work



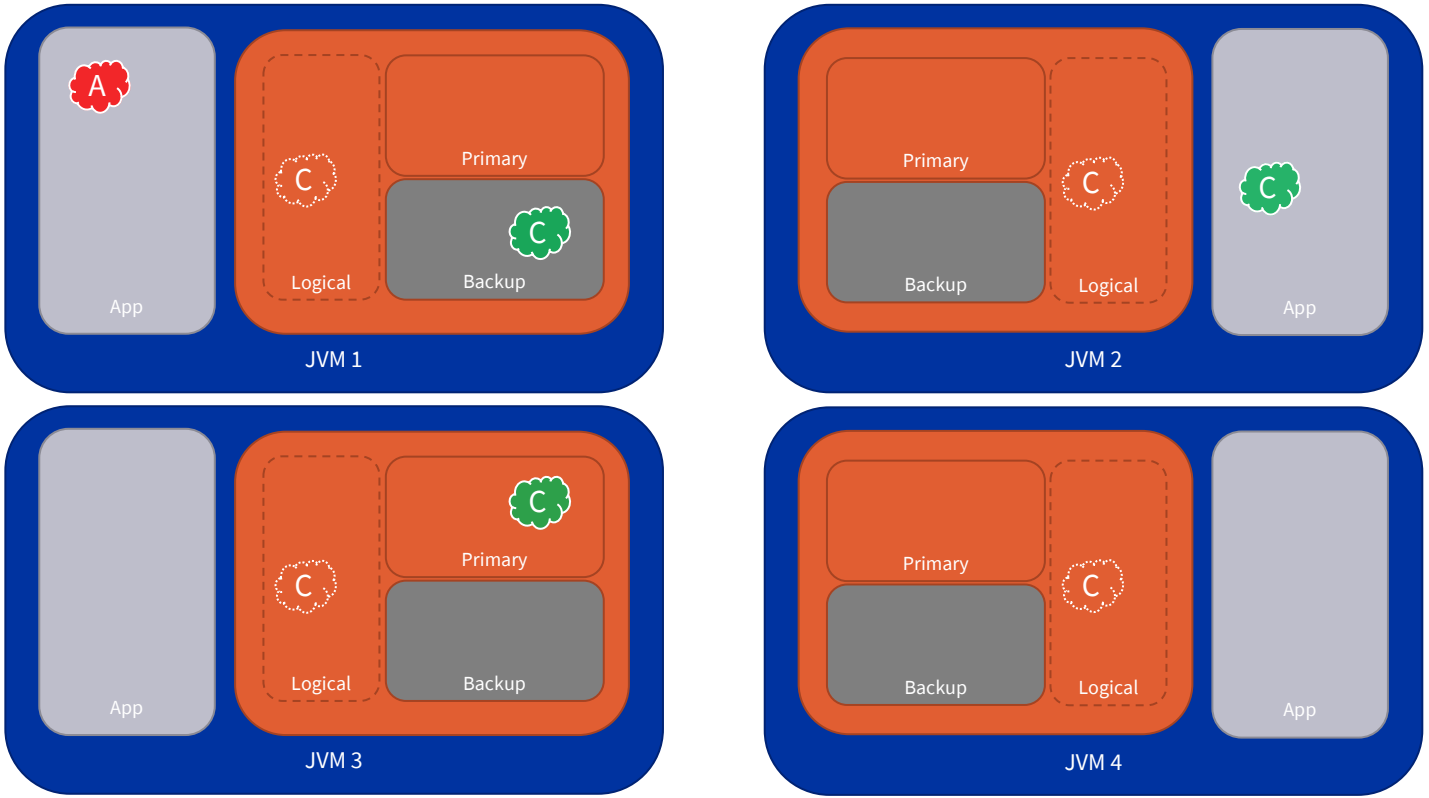
How Distributed Caches Work



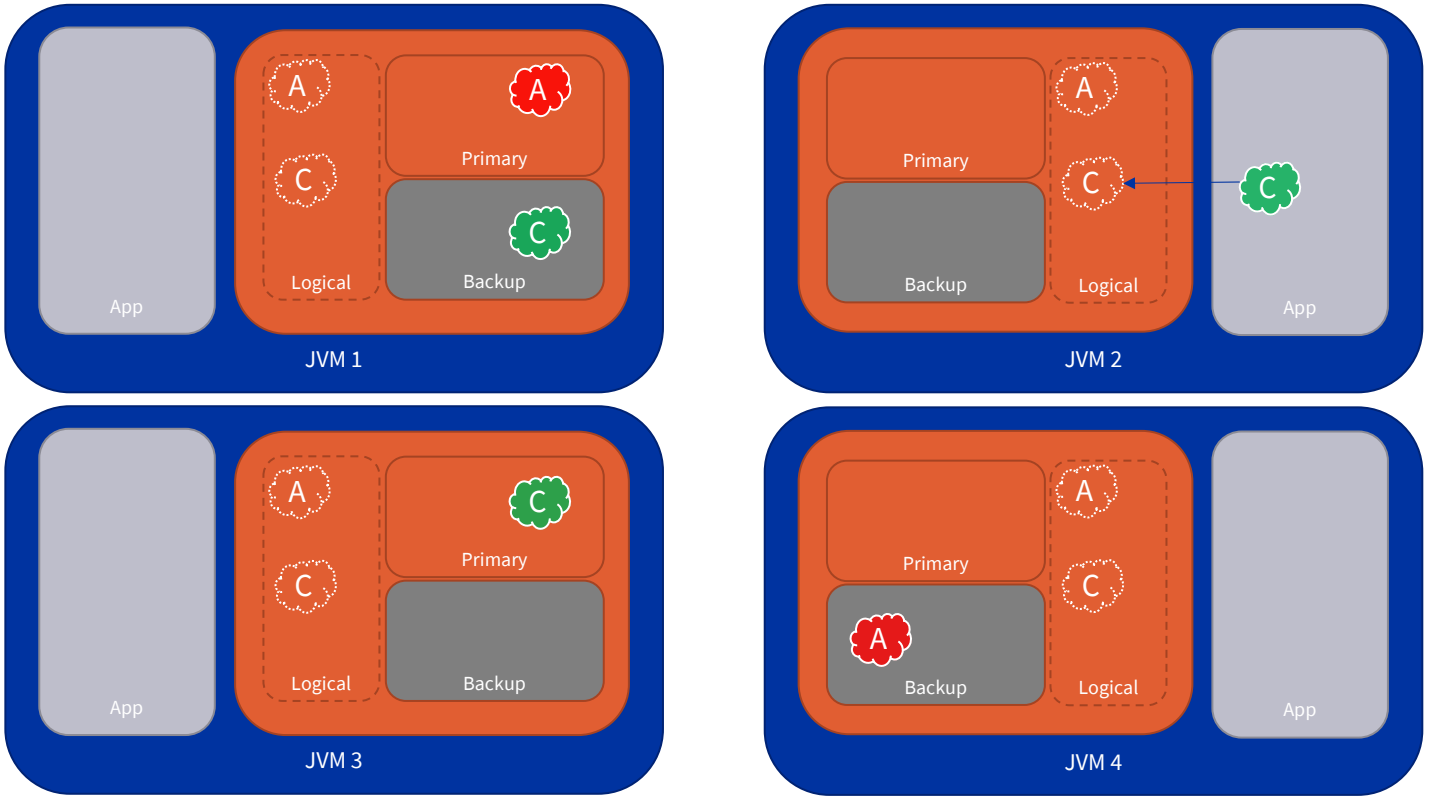
Add new data to a Distributed Cache



Add new data to a Distributed Cache

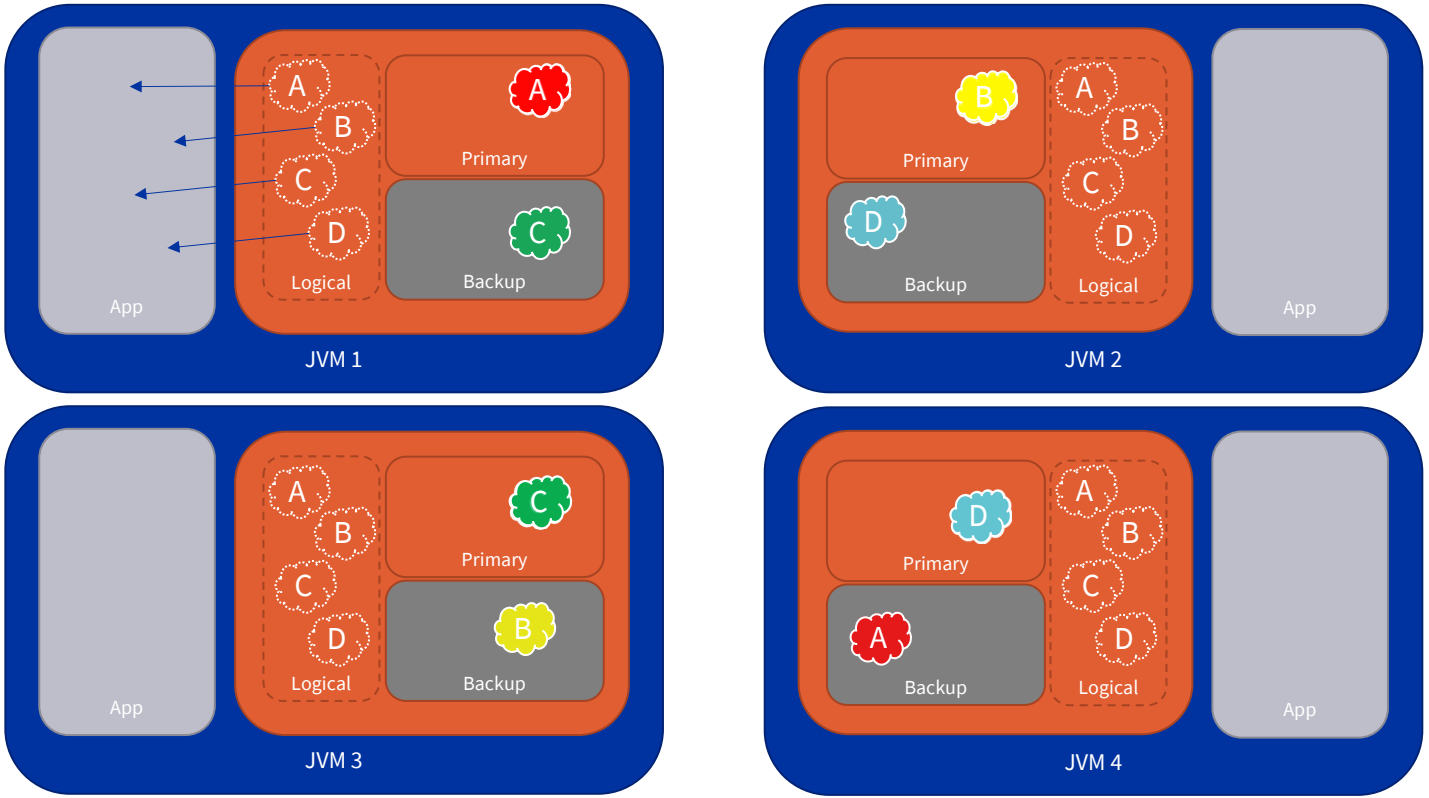


Add new data to a Distributed Cache

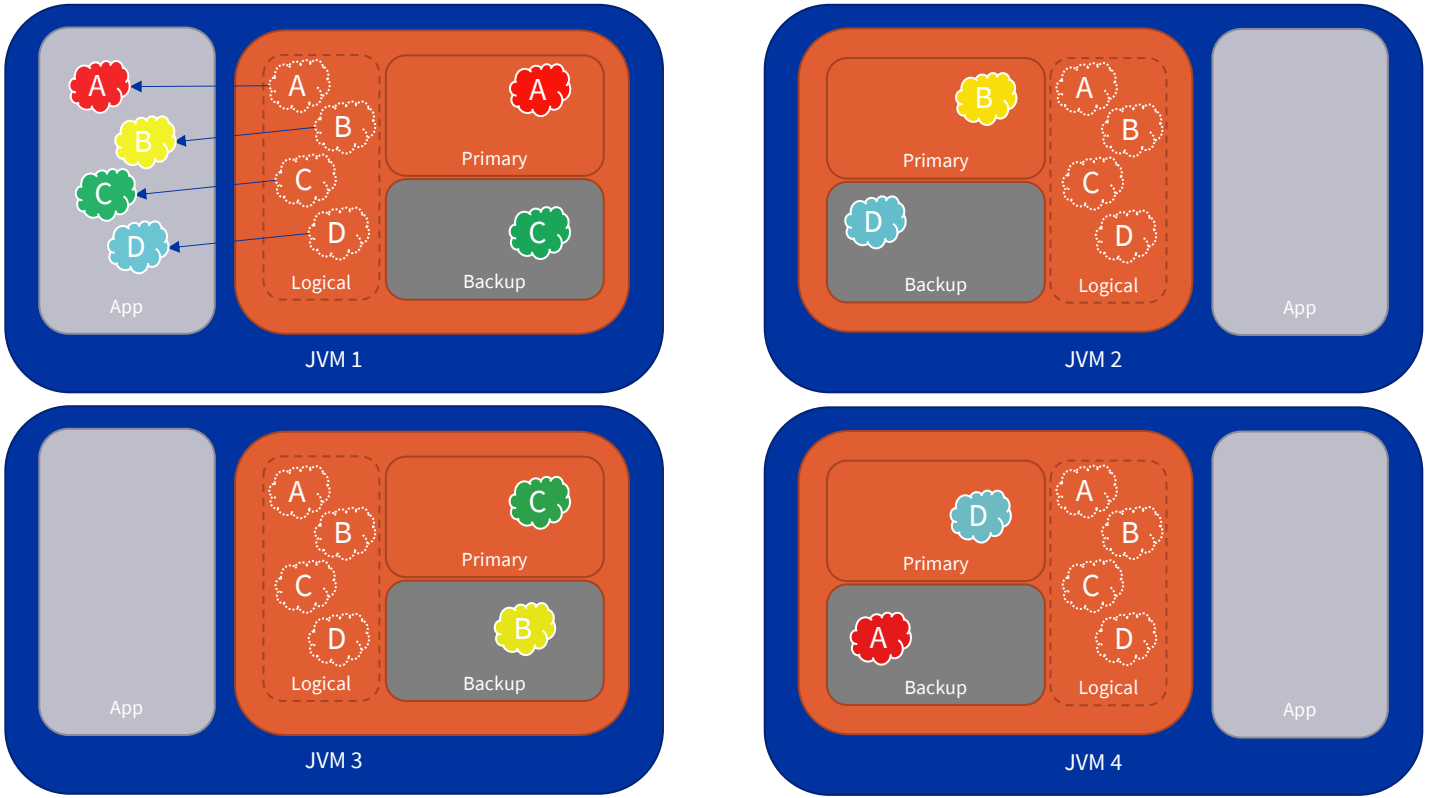


Add new data to a Distributed Cache

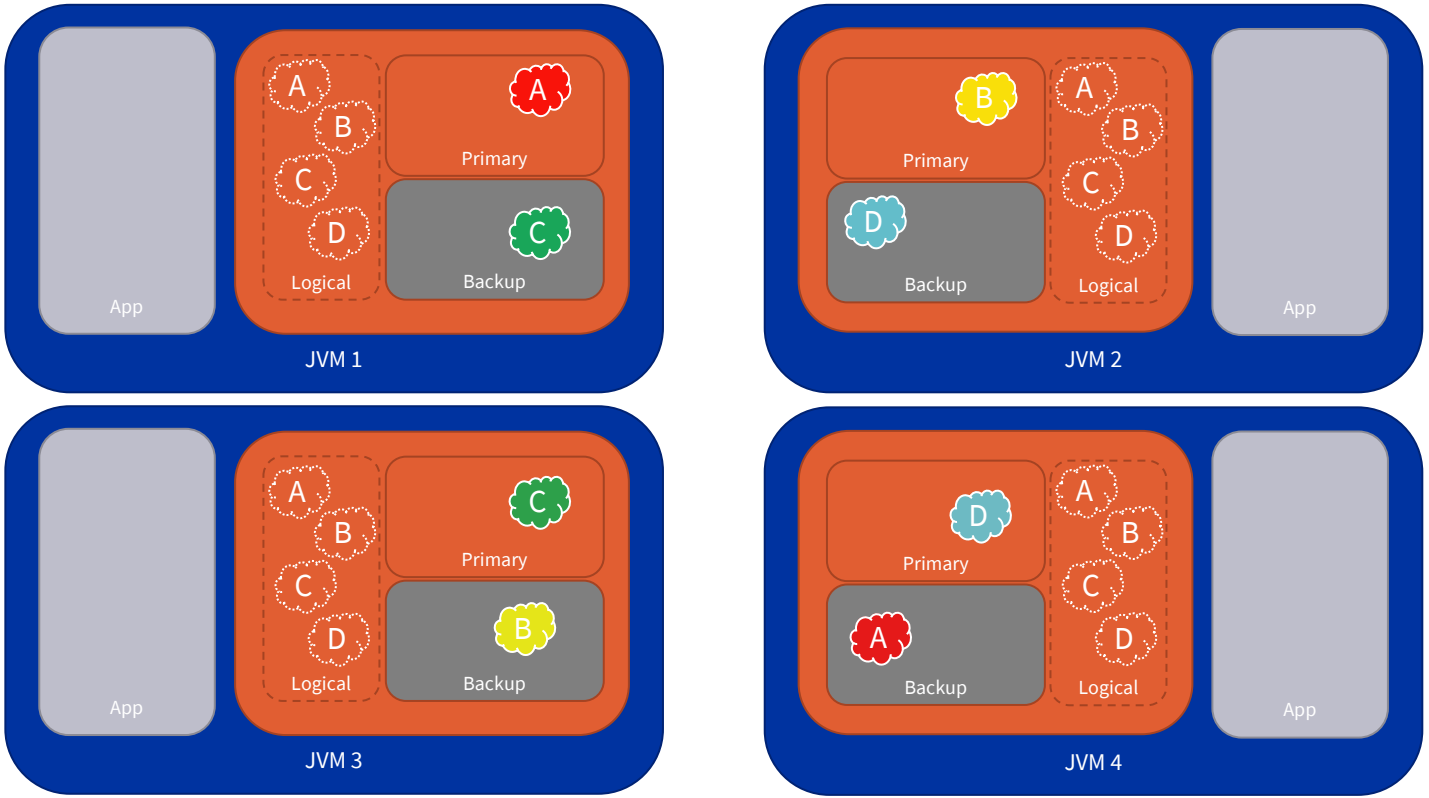
Retrieve data in a Distributed Cache



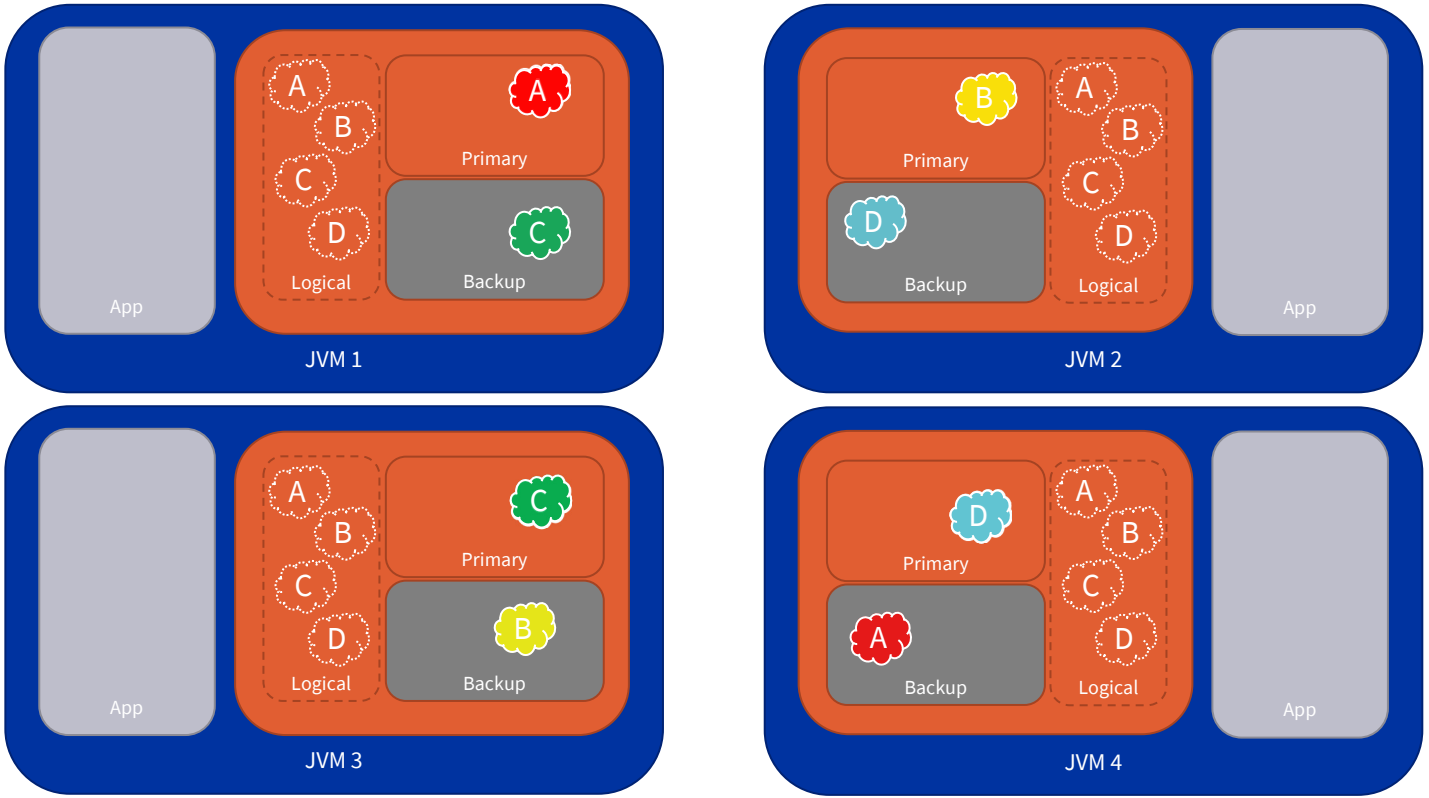
Retrieve data in a Distributed Cache



Retrieve data in a Distributed Cache



Distributed Cache on Failover



Distributed Cache on Failover

Distributed Cache on Failover

Traditional Cache vs Distributed Cache

Scalability

- A distributed caching architecture scales horizontally whereas a traditional caching architecture only supports vertical scaling

Redundancy

- In a single instance caching, there is no redundancy of data

Reliability

- Traditional caches use the same heap space as the application so it’s bound to the constraints of memory
- Distributed caches run as independent processes across multiple nodes in a cluster which makes it transparent to the program is if one node dies or is killed

Integration with Apache Ignite

Steps to integrate Apache Ignite in C2MON

- **Create simple serializable and stable POJOs**
  - Avoid breaking changes

```
public class Equipment implements Serializable{

  private String address;

  private Long processId;

  private final SupervisionEntity enitity = SupervisionEntity.EQUIPMENT;
```

Steps to integrate Apache Ignite in C2MON

- **Create simple serializable and stable POJOs**
  - Avoid breaking changes

```
public class Equipment implements Serializable{

  private String address;

  private Long processId;

  private final SupervisionEntity enitity = SupervisionEntity.EQUIPMENT;
```

- **Use of Java Generics**
  - Restrict what types of objects can be persisted

```
public class IgniteC2monCache<V extends Cacheable>
```

Steps to integrate Apache Ignite in C2MON

- **Create simple serializable and stable POJOs**
  - Avoid breaking changes

```
public class Equipment implements Serializable{

  private String address;

  private Long processId;

  private final SupervisionEntity enitity = SupervisionEntity.EQUIPMENT;
```

- **Use of Java Generics**
  - Restrict what types of objects can be persisted

```
public class IgniteC2monCache<V extends Cacheable>
```

- **Use Spring to interact with Apache Ignite**
  - @Transactional annotations

```
@Transactional(propagation = Propagation.REQUIRED)
<T> T executeTransaction(Supplier<T> callable);
```

Orchestration of C2MON integrated with Apache Ignite

- **Rolling Updates**
- **Load Balancing**
- **Scalability**

Orchestration of C2MON integrated with Apache Ignite

- **Rolling Updates**
- **Load Balancing**
- **Scalability**

Thursday, October 21st

14:00-14:15 THBL03 **The State of Containerization in CERN Accelerator Controls**

Rémi Voirin (CERN, Geneva)

Summary

Summary

- C2MON is a critical piece of infrastructure which needs to run 24/7/365
- Scalability, consistency of data and rolling upgrades are crucial
- Distributed cache is a good solution to achieve those goals
- The integration of Apache Ignite with Spring makes it easier to setup and configure
- This paves the way to move to the cloud