



# Making Data Work for Developers with Kubernetes & Cassandra

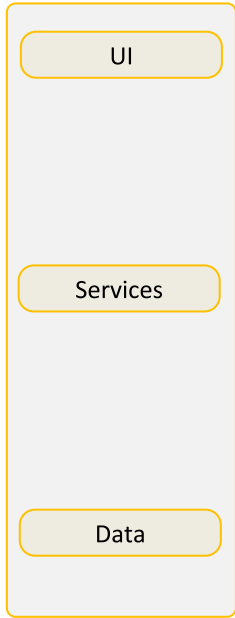
Chris Splinter | DataStax, Inc.

Patrick McFadin | DataStax, Inc.

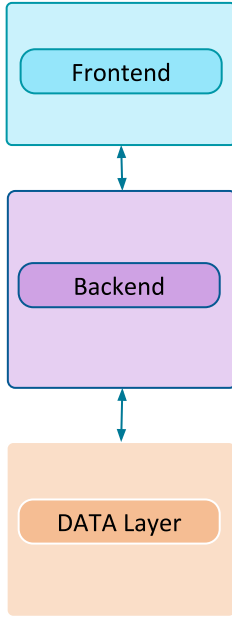
# From Monolith to Microservices

# All things distributed

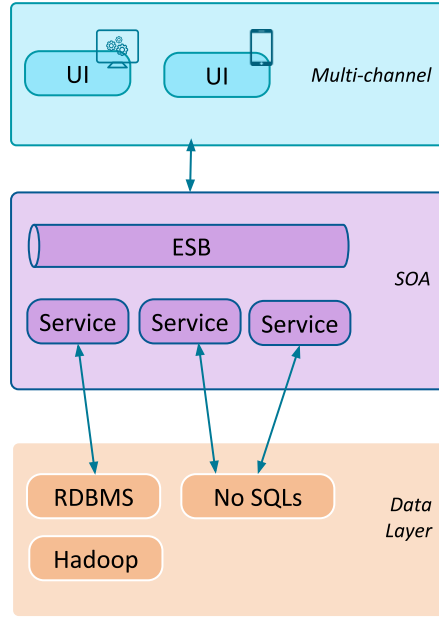
## Monolith



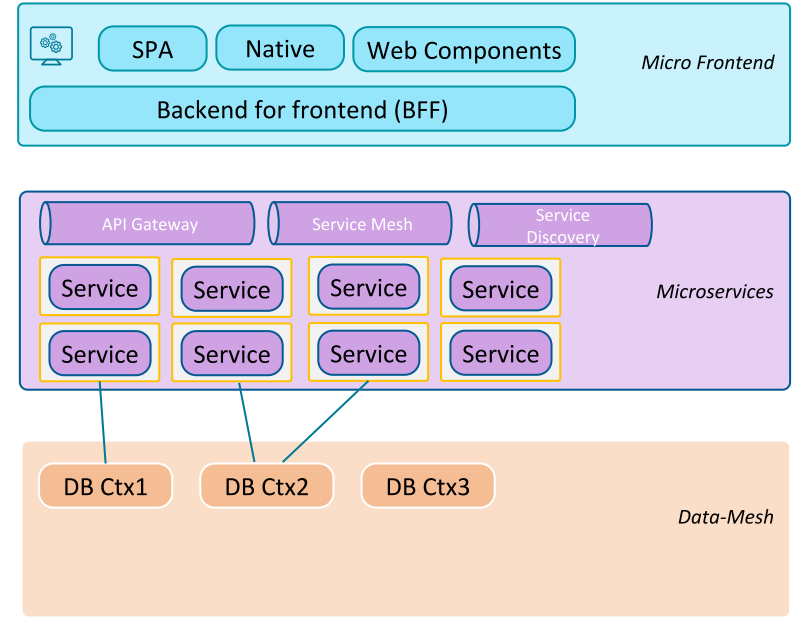
## ➡ Multi-Tiers

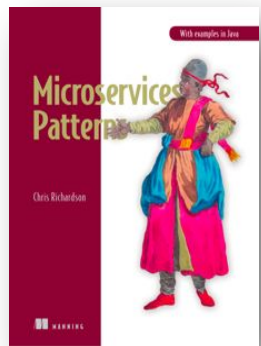
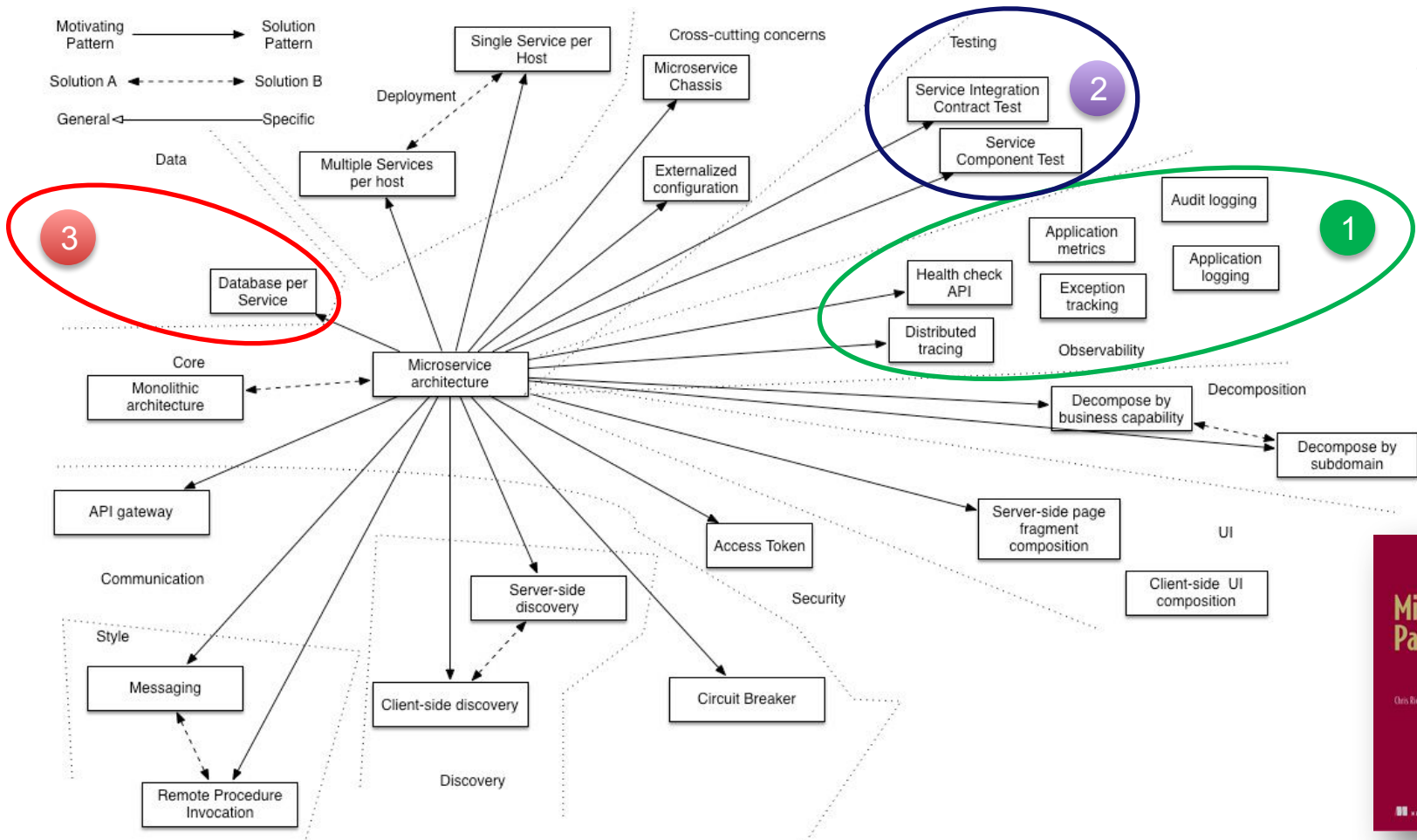


## ➡ SOA



## ➡ Microservice Architectures

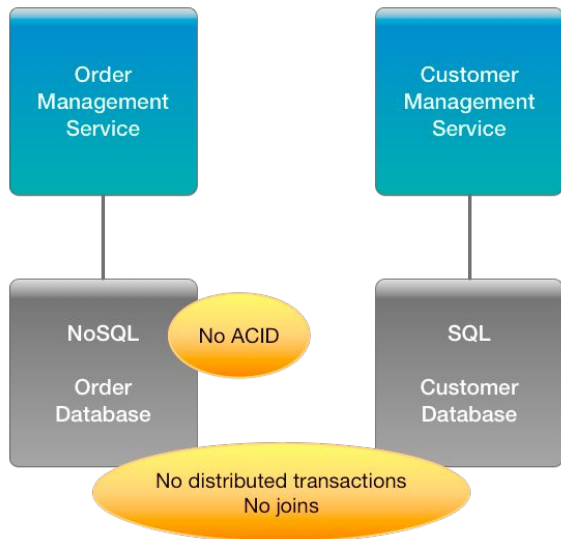




# 1 database per service ?

- ❑ What you need
  - ❑ Services must be loosely coupled
  - ❑ Each service is in charge of its data and isolated
- ❑ How does this relate to Cassandra ?
  - ❑ Coupling Data : Shared Database  $\neq$  shared Data
    - ❑ Data Isolation per Keyspace (set of tables)
    - ❑ Data Isolation per Table (1 query = 1 table)
    - ❑ Data Isolation per Role Based Access Control (RBAC)
  - ❑ Coupling Infrastructure:
    - ❑ Multiple Rings, Replicator factor, no SPOF

# Distributed arch are not “ACID” but “BASE”



- Atomicity Consistency Isolation Durability (ACID) does not work anymore
- Distributed transactions / 2 phases commit (2PC) does not work anymore
- BASE (Basic Availability, Soft-State, Eventual Consistency)
  - Availability has higher priority than Consistency
  - Event Sourcing: Saving messages and not final state
  - Idempotence: Messages should be replayable

# Apache Cassandra Microservices

- **REALTIME REQUESTS & SCALABILITY AT CORE**
- **DISTRIBUTED ARCHITECTURES**
  - From ACID to BASE (**B**asic **A**vailability, **S**oft-State, **E**ventual Consistency)
  - Implementations: CQRS, Event Sourcing
  - Colocate service and Data
- **DECOUPLING BY DESIGN**
  - 1 KEYSPACE = DOMAIN
  - 1 QUERY = 1 TABLE

# Case Studies: Cassandra + Microservices





# monzo

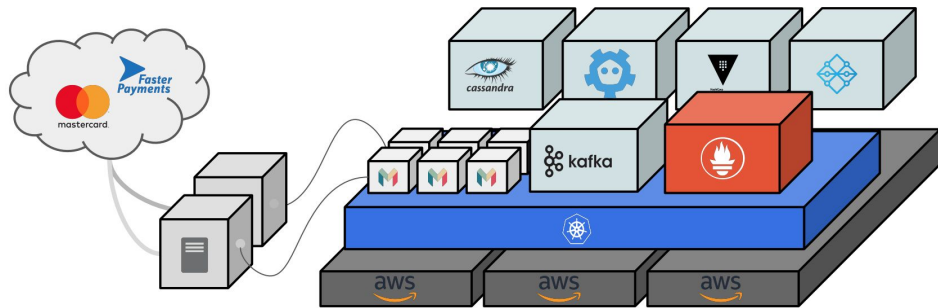
"We've made a big bet on Cassandra because it can **scale linearly**, which has let us nearly **quadruple our customer base every year** without an issue. At peak time, we can now handle **300,000 reads per second**."

"Every service has a different Cassandra 'keyspace',

We have **data isolation** between different microservices by default,

so most services never read data written by another service"

Source: Jack Kleeman ([Monzo blog](#))



Source: Matt Heath, Suhail Patel ([QCon](#))



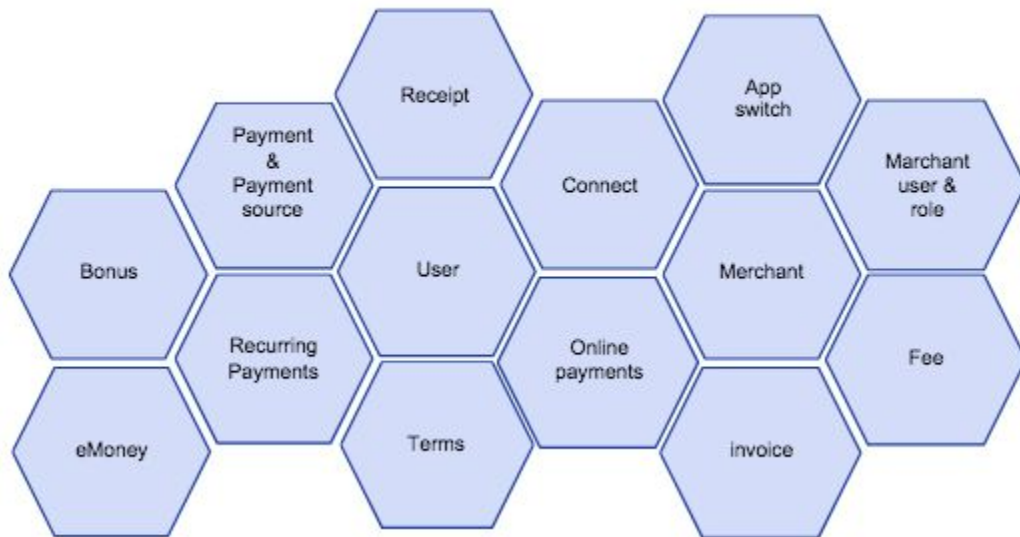
Different teams per service

Bounded Contexts

Completely isolated

Active - Active

## MobilePay domain



Source: Brian Nielsen, [DataStax Webinar](#)



Cluster per business unit

## Highlights and Challenges

- We serve billions of requests during the tax season
- We support roughly 100K TPS during TAX peak hours
- We manage Peta Bytes of data in S3
- What does all this mean for us ?
  - We have to be highly-available
  - We have to be highly-performant
  - We have to be highly-secure

One client impacted means a call to us

intuit

Intuit Confidential and Proprietary 4

Source: Achal Kumar, Larry Raab - [DataStax Accelerate](#)



## Keyspace per service

---

Predefine dedicated Keyspace/User combination per micro-service

---

Sharing lower non-prod DSE clusters in NON-PROD by groups of environments (DEV/QE/DEMO/BUG-FIX, UAT/SIT, E2E/PROD-FIX)

---

Deployment of CQL objects/DDL code and seed data as part of CICD DevOps applications integration tools

---

Over a 1000 table objects in lower non-prod clusters across all keyspaces (**watch heap space**)

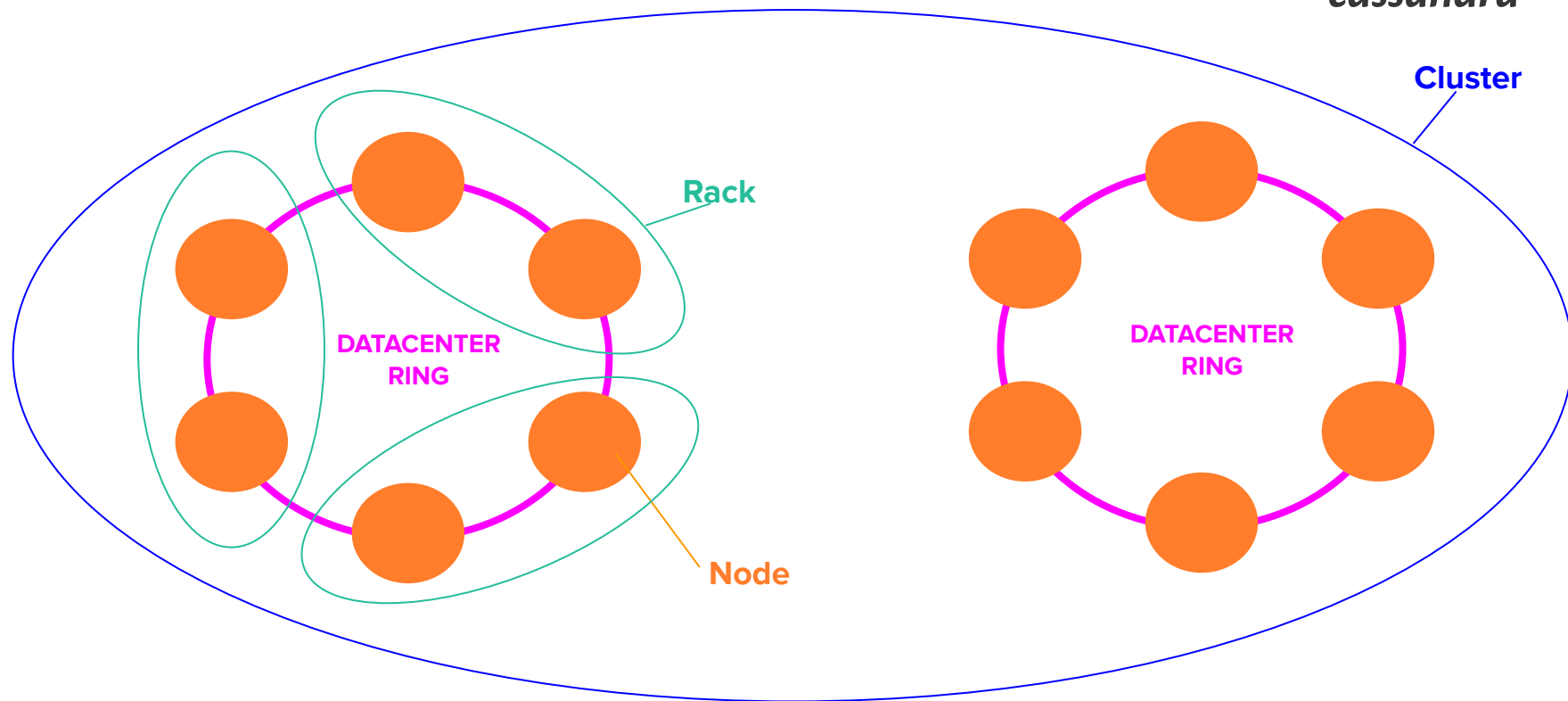
Source: Sergiy Smyrnov - [DataStax Accelerate](#)

# Takeaways

1. Microservices can bring agility but also complexity
2. Do what's right for your business and organization
3. Users choose Cassandra to match the scale and availability of their applications

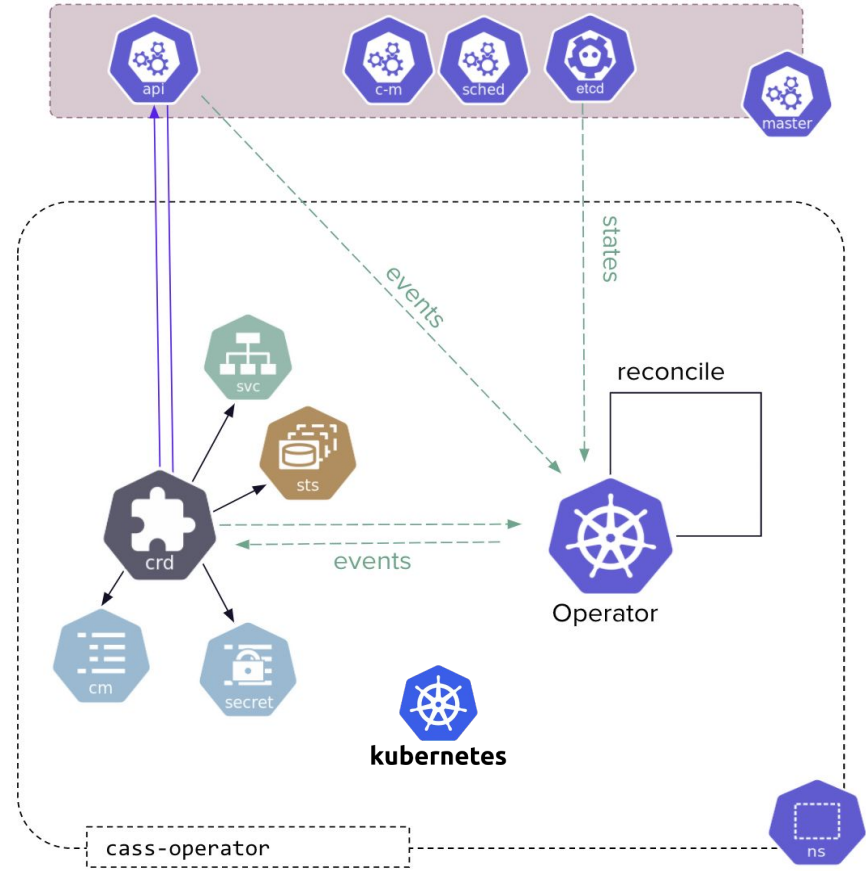
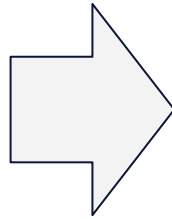
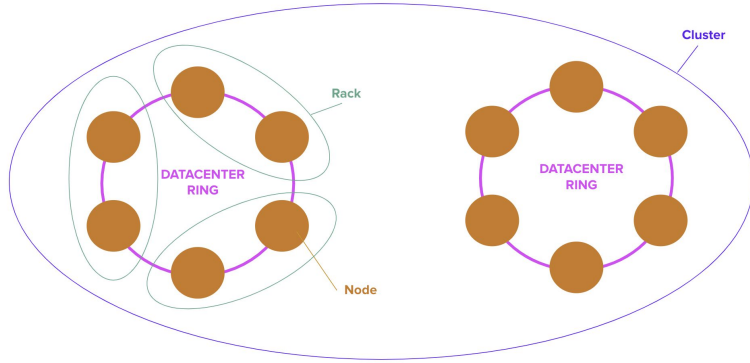
# Cassandra Operator for Kubernetes

# Apache Cassandra™ Vocabulary





**cassandra**





# Cass Operator : Features

- Proper token **ring initialization**, with only one node bootstrapping at a time
- **Seed node** management -
  - one per rack, or three per datacenter, whichever is more
- Server configuration integrated into the **CassandraDatacenter CRD**
  - Rolling reboot nodes by changing the CRD
  - Store data in a rack-safe way - one replica per cloud AZ
  - Scale up racks evenly with new nodes
  - Replace dead/unrecoverable nodes
- Multi DC clusters (limited to one Kubernetes namespace)

# CRD CassandraDataCenter

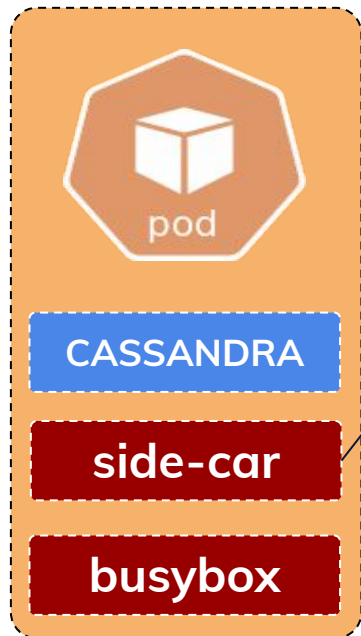
```
apiVersion: cassandra.datastax.com/v1beta1
kind: CassandraDatacenter
metadata:
  name: dc1
spec:
  clusterName: cluster1
  serverType: cassandra
  serverVersion: "3.11.6"
  managementApiAuth:
    insecure: {}
  size: 1
  storageConfig:
    cassandraDataVolumeClaimSpec:
      storageClassName: server-storage
      accessModes:
        - ReadWriteOnce
      resources:
        requests:
          storage: 5Gi
  config:
    cassandra-yaml:
      authenticator: org.apache.cassandra.auth.PasswordAuthenticator
      authorizer: org.apache.cassandra.auth.CassandraAuthorizer
      role_manager: org.apache.cassandra.auth.CassandraRoleManager
    jvm-options:
      initial_heap_size: "800M"
      max_heap_size: "800M"
```

```
apiVersion: cassandra.datastax.com/v1beta1
kind: CassandraDatacenter
metadata:
  name: multi-rack
spec:
  clusterName: multi-rack
  serverType: cassandra
  serverVersion: 3.11.6
  managementApiAuth:
    insecure: {}
  size: 9
  racks:
    - name: us-east1-b
      zone: us-east1-b
    - name: us-east1-c
      zone: us-east1-c
    - name: us-east1-d
      zone: us-east1-d
  storageConfig:
    cassandraDataVolumeClaimSpec:
      storageClassName: standard
      accessModes:
        - ReadWriteOnce
      resources:
        requests:
          storage: 5Gi
```

# Cass Operator Pod

## Cassandra Management API Service

<https://github.com/datastax/management-api-for-apache-cassandra>



### Management API for Apache Cassandra 0.1 OAS3

<https://raw.githubusercontent.com/datastax/management-api-for-apache-cassandra/master/management-api-server/doc/openapi.json>

This is a Restful service for operating Apache Cassandra. You can find out more about the Management API on [Github](#)

Apache 2.0

#### default

POST	/api/v0/ops/auth/role	Creates a new user role
GET	/api/v0/probes/liveness	Indicates whether this service is running
GET	/api/v0/probes/readiness	Indicates whether the Cassandra service is ready to service requests
GET	/api/v0/probes/cluster	Indicated whether the Cassandra cluster is able to achieve the specified consistency
POST	/api/v0/ops/seeds/reload	
POST	/api/v0/ops/keyspace/refresh	Load newly placed SSTables to the system without restart
POST	/api/v0/ops/keyspace/cleanup	Triggers the immediate cleanup of keys no longer belonging to a node. By default, clean all keyspaces
POST	/api/v0/lifecycle/start	
POST	/api/v0/lifecycle/stop	
POST	/api/v0/lifecycle/configure	
GET	/api/v0/lifecycle/pid	
GET	/api/v0/metadata/versions/release	Returns the Cassandra release version
GET	/api/v0/metadata/endpoints	Returns this nodes view of the endpoint states of nodes
POST	/api/v0/ops/node/drain	Drain the node (stop accepting writes and flush all tables)

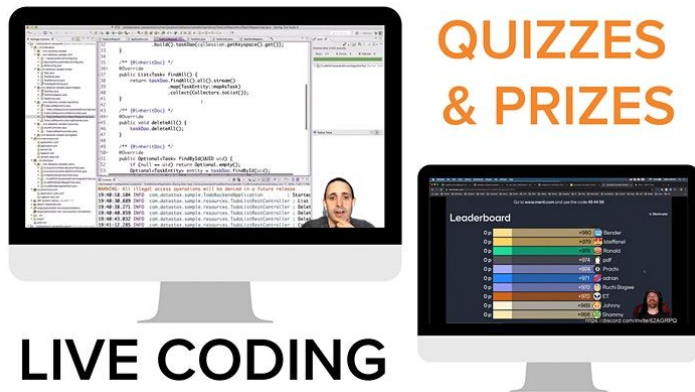
# Cassandra Community Effort

- Combine the many Cassandra Operators out there
- Closer alignment to the Apache Cassandra project



# Join Our Community and Learn More

## Cassandra Could Native Workshop Series



<https://www.datastax.com/company/events/cassandra-developer-workshop>

<https://www.datastax.com/dev>

# Cloud-native Cassandra-as-a-Service

## Try Now for Free

DataStax Astra simplifies cloud-native Cassandra application development. It reduces deployment time from weeks to minutes, removing the biggest obstacle to using Cassandra, which is behind many of the most heavily used applications in the world.

Get you **FREE** trial - Cloud-native Cassandra-as-a-Service Built on Apache Cassandra

- 10 GB free, no credit card needed.
- Built on open-source Apache Cassandra™ active-everywhere NoSQL.
- Eliminate the overhead to install and operate Cassandra.
- Enterprise-level reliability, security, and management.
- Deliver applications faster with CQL, REST and GraphQL APIs.
- Deploy dedicated clusters on AWS or GCP.
- Move data in and out of open-source Apache Cassandra.



**Try the free trial!**



Thank You!