

# Week 4: App. Development \*

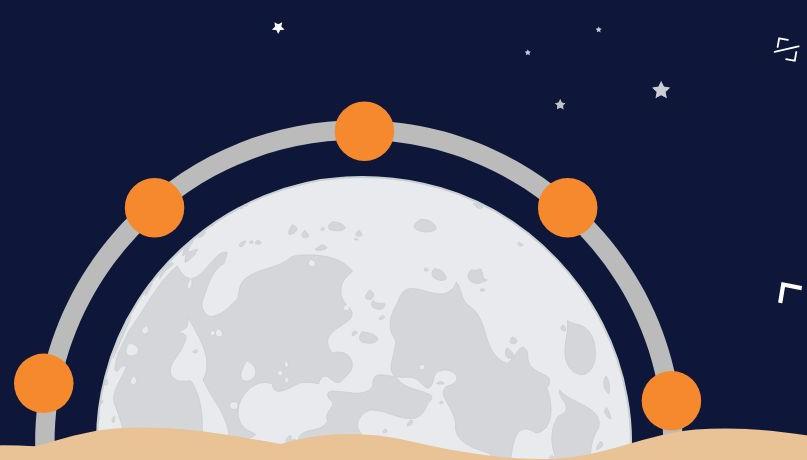


Apis and Microservices



Cassandra Cloud-Native  
Workshop Series

Building Cloud-Native apps with Cassandra Expertise



# DataStax Developer Special Unit !



David  
Jones-Gilardi



Eric  
Zietlow



Erick  
Ramirez



Cédrick  
Lunven



Bettina  
Swynnerton



Jack  
Fryer



Aleksandr  
Volochnev

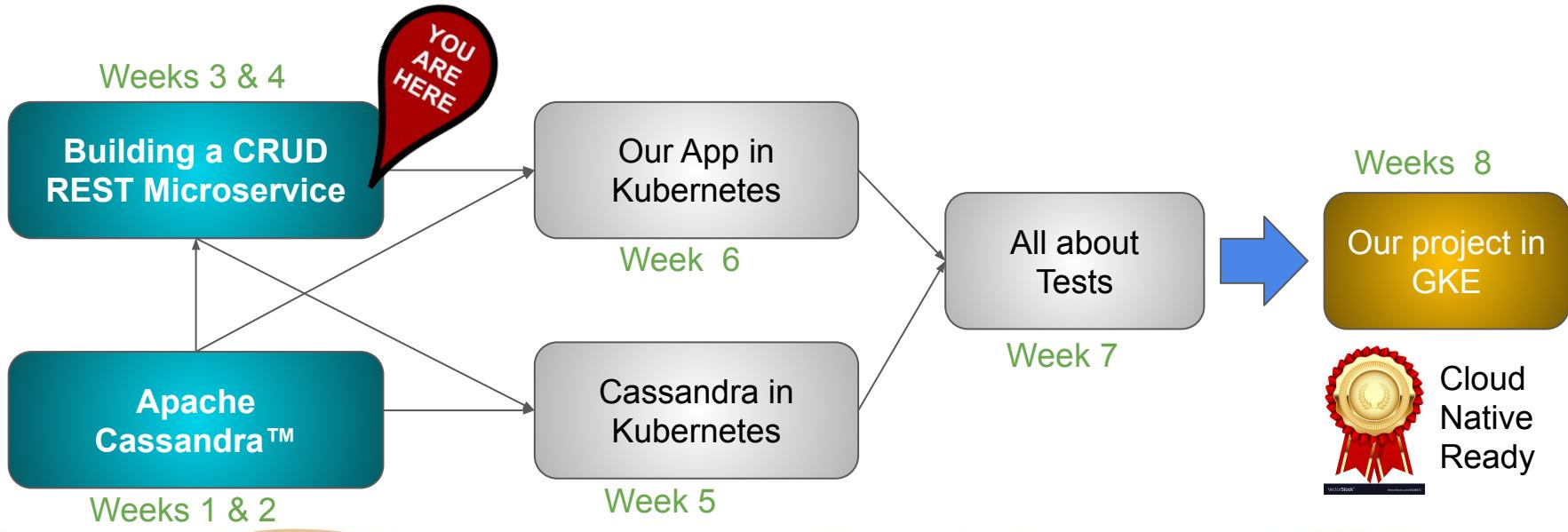


MATERIALS



[bit.ly/CassandraWorkshopMaterials](https://bit.ly/CassandraWorkshopMaterials)

# Workshops Series = Not only Cassandra



MATERIALS



[bit.ly/CassandraWorkshopMaterials](https://bit.ly/CassandraWorkshopMaterials)



# Developer Workshop Series **Week4**



What we will  
cover:

- Housekeeping
- Cassandra and Microservices
- APIs, Endpoint and Access Patterns
- Rest vs GraphQL vs GRPC
- Introduction to Quarkus (Java)
-  Surprise !!

MATERIALS



[bit.ly/CassandraWorkshopMaterials](https://bit.ly/CassandraWorkshopMaterials)

# HouseKeeping

- **Pre-requisites, we expect you:**
  - To have already created an Astra instance (week 1)
  - To have knowledge with Cassandra Data Modelling (week 2)
  - To know basics of one of the following languages:



- **You don't have to install anything**

## Livestreams

YouTube



Twitch



## Astra.datastax.com



cassandra

## Live Questions

YouTube



Discord



## Materials & Help



## Github



## Gitpod



MATERIALS → [bit.ly/CassandraWorkshopMaterials](https://bit.ly/CassandraWorkshopMaterials)

# menti.com

# 43 69 82



Available on the iPhone  
**App Store**

GET IT ON  
**Google play**



# Developer Workshop Series **Week4**

What we will cover:

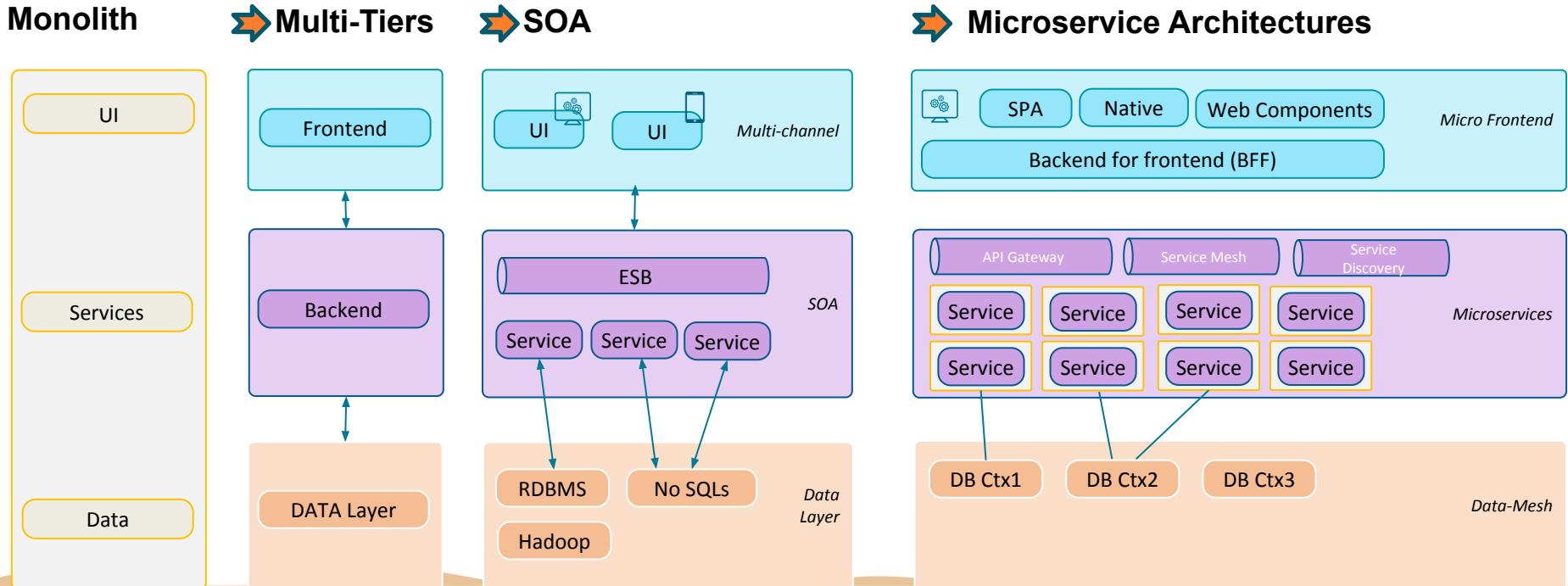
- Housekeeping
- Cassandra and Microservices
- APIs, Endpoint and Access Patterns
- Rest vs GraphQL vs GRPC
- Introduction to Quarkus (Java)
-  Surprise !!

MATERIALS



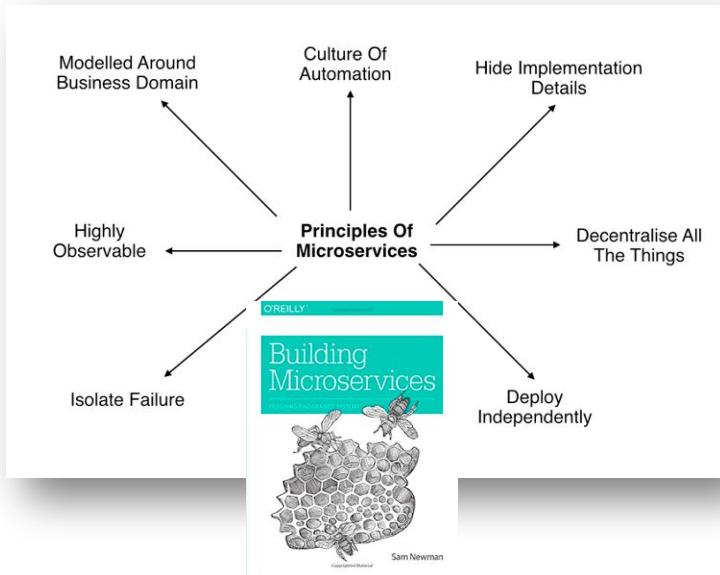
[bit.ly/CassandraWorkshopMaterials](https://bit.ly/CassandraWorkshopMaterials)

# All things distributed



MATERIALS → [bit.ly/CassandraWorkshopMaterials](https://bit.ly/CassandraWorkshopMaterials)

# Microservice



Martin Fowler



- ✓ Organized around **Business Capabilities**
- ✓ **Products not Projects**
- ✓ **Smart endpoints and dumb pipes**
- ✓ **Decentralized Governance**
- ✓ **Decentralized Data Management**
- ✓ **Infrastructure Automation**
- ✓ **Design for failure**
- ✓ **Evolutionary Design**

MATERIALS

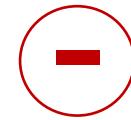


[bit.ly/CassandraWorkshopMaterials](http://bit.ly/CassandraWorkshopMaterials)



## Advantages

- **Cost Reductions**
  - Scale only what you need to
  - Design is simpler
  - Speed in Development
- **Risk Reductions**
  - Resiliency
  - Simple Developments
  - Security
  - Monitoring



## Drawbacks



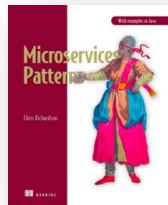
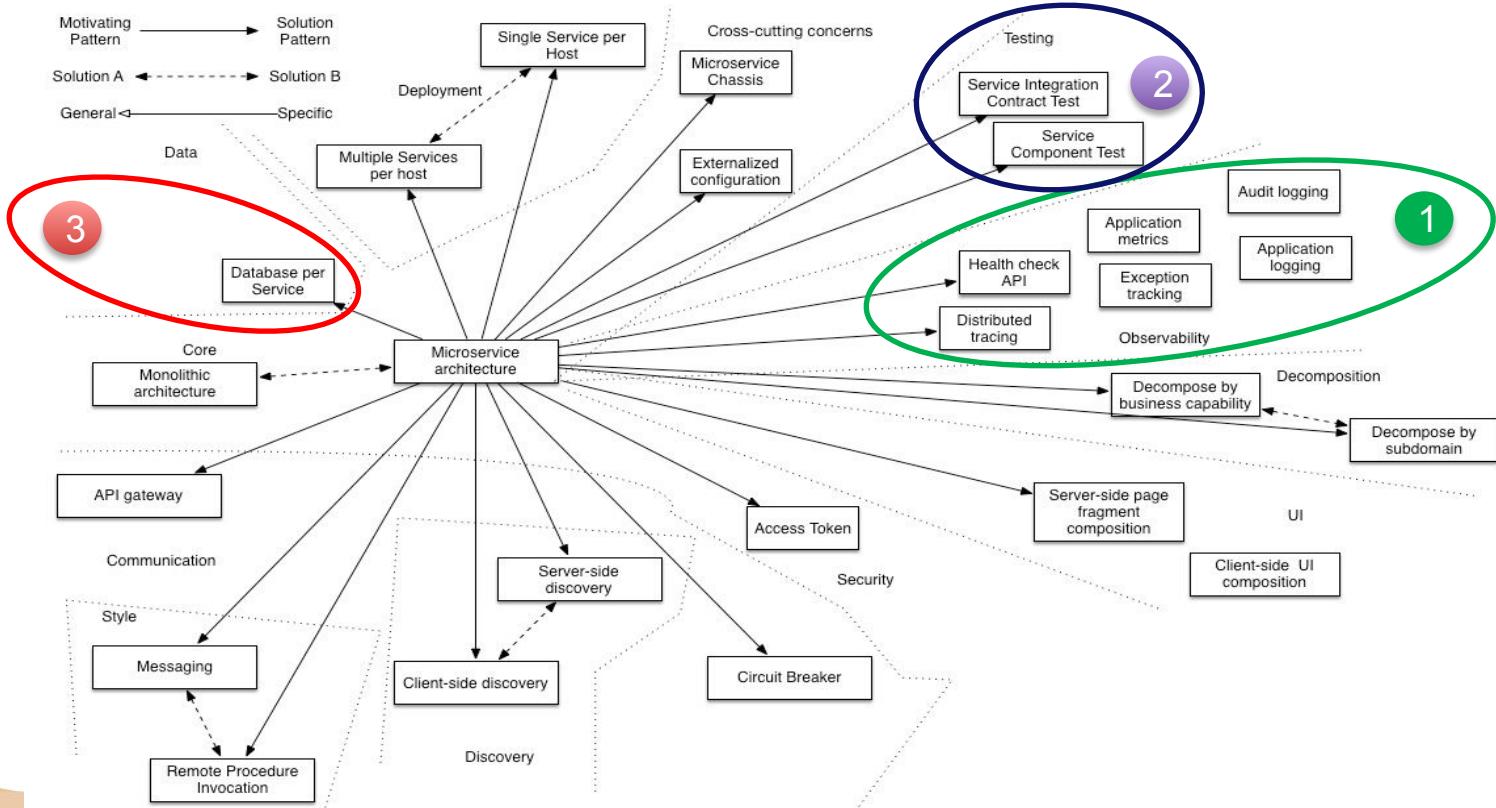
- **Introducing Complexity**

- Security
- Transactions Management
- Service Orchestrations

- **Culture Change**

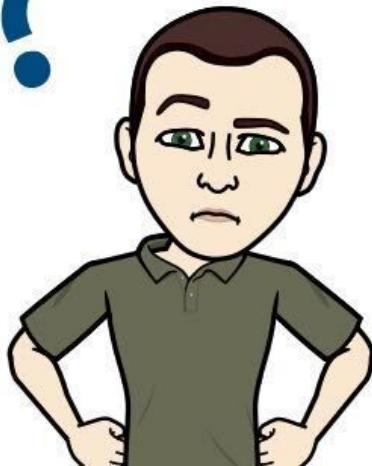
- CI/CD
- Failfast

- **Higher run footprint**



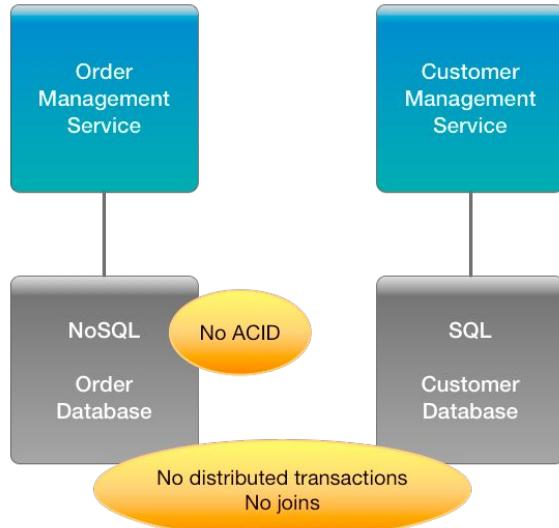
MATERIALS ➔ [bit.ly/CassandraWorkshopMaterials](https://bit.ly/CassandraWorkshopMaterials)

# 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 ≠ 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

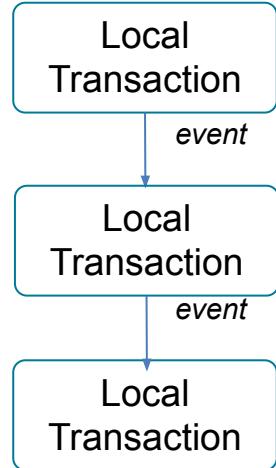
# Distributed arch are not “ACID” but “BASE”



- **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

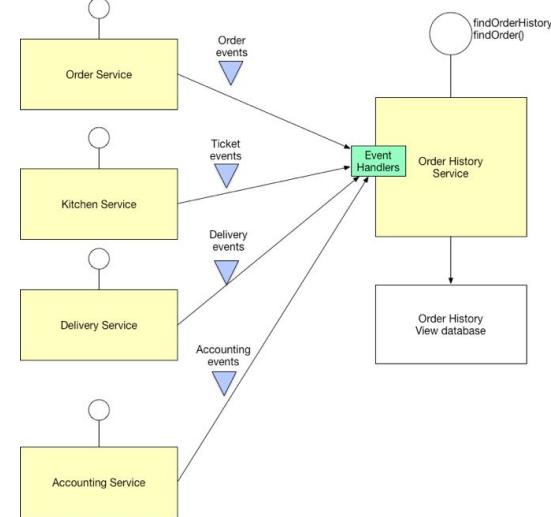
# Patterns to the rescue

## SAGA



- Event Driven
- Choreography

## CQRS (Command Query Responsibility Segregation)



- Split reads and writes
- Reads on “views” built through history (*event sourcing*)

MATERIALS



[bit.ly/CassandraWorkshopMaterials](http://bit.ly/CassandraWorkshopMaterials)

# 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



MATERIALS



[bit.ly/CassandraWorkshopMaterials](https://bit.ly/CassandraWorkshopMaterials)



# Getting Started with Astra Application

The screenshot shows the Astra Application interface. At the top, there's a navigation bar with the DataStax logo and the title "Getting Started with Astra". On the right side of the header is a "LAUNCH NEW JOURNEY" button. Below the header is a dark background with a starry space theme. In the center, there's a "Journey Checklist" section with two checked items: "Writing 1000 Rows to Astra" and "Reading 1000 Rows from Astra". Below the checklist is a "Replaying Rows" section with a small icon of a person. To the right of these sections is a large, stylized illustration of a white rocket ship with orange fins and a blue stripe, launching from a desert landscape. The rocket has a circular window on its side showing a view of the stars. At the bottom of the screen, there are three circular gauges and a chart area. The first gauge on the left is labeled "Temperature: 69.85 Fahrenheit" and "Astra Table: spacecraft\_temperature\_over\_time". The second gauge in the middle is labeled "Speed: 31536.95 km/h" and "Astra Table: spacecraft\_speed\_over\_time". The third gauge on the right is labeled "Pressure: 100.3 kPa" and "Astra Table: spacecraft\_pressure\_over\_time". To the right of these gauges is a chart titled "Current Spacecraft Location" with axes for X and Y coordinates in kilometers.

Astra Table	Rows Written	Rows Read	Current Row Displayed
spacecraft_temperature_over_time	1000	1000	189
spacecraft_speed_over_time	1000	1000	189
spacecraft_pressure_over_time	1000	1000	189
spacecraft_location_over_time	1000	1000	189

MATERIALS



[bit.ly/CassandraWorkshopMaterials](https://bit.ly/CassandraWorkshopMaterials)

# Microservice Architecture



# Exercise



Start backend and API



The screenshot shows a terminal window with the following content:

```
File Edit Selection View Go Debug Terminal Workspace Help
EXPLORER: CASSANDRA-WORKSHOP-SERIES README.MD
> materials
> week1 - Getting Started with...
> week2-DataModel
> week3-AppDev-crud
> week4-AppDev-api
> getting-started-with-astra...
> getting-started-with-astra...
> getting-started-with-astra...
> getting-started-with-astra-ui
> images
> samples-codes
> HACKATHON.MD
> README.MD
> week5-Cass-in-k8s
> week6-App-in-k8s
> week7-Test-your-Apps
> week8-k8s-in-the-cloud
> .gitignore
> .gitpod.yml
> README.md
```

**Cassandra Workshop Series**  
Every Wednesday for Americas, Every Thursday for Europe and Asia Pacific

**Application Development - API REST**

Gitpod ready-to-code license Apache-2.0 chat 182 online

Cassandra Workshop Series are an interactive experience. Datastax Developer Advocates share some knowledge about Apache Cassandra™ NoSQL database and how you build Cloud Native applications. You interact with them through chats ([youtube](#) and [discord](#)), quizzes ([menti.com](#)), and exercises.

In this repository, you'll find everything you need related to week 3 of the **Cassandra Workshop Series**. For simplicity all exercise instructions will be listed in a single **README** for each language: Java, Javascript, Python, CSharp. We recommended you to have some previous experience with the language you choose.

Now if you are NOT familiar with any of those, we recommended to use Python, the is the simplest. Ready ? go !

**Workshop Materials**

Materials	Description and Links
Slidedeck	<a href="#">SLIDEDECK</a>
Chat with us on Discord	<a href="#">DISCORD</a>
Ask Question during week	<a href="#">COMMUNITY</a>

gitpod /workspace/cassandra-workshop-series \$

MATERIALS → [bit.ly/CassandraWorkshopMaterials](https://bit.ly/CassandraWorkshopMaterials)

# Developer Workshop Series **Week4**



What we will cover:

- Housekeeping
- Cassandra and Microservices
- Endpoint and Access Patterns
- API : Rest, GraphQL, GRPC
- Introduction to Quarkus (Java)
-  Surprise !!

MATERIALS

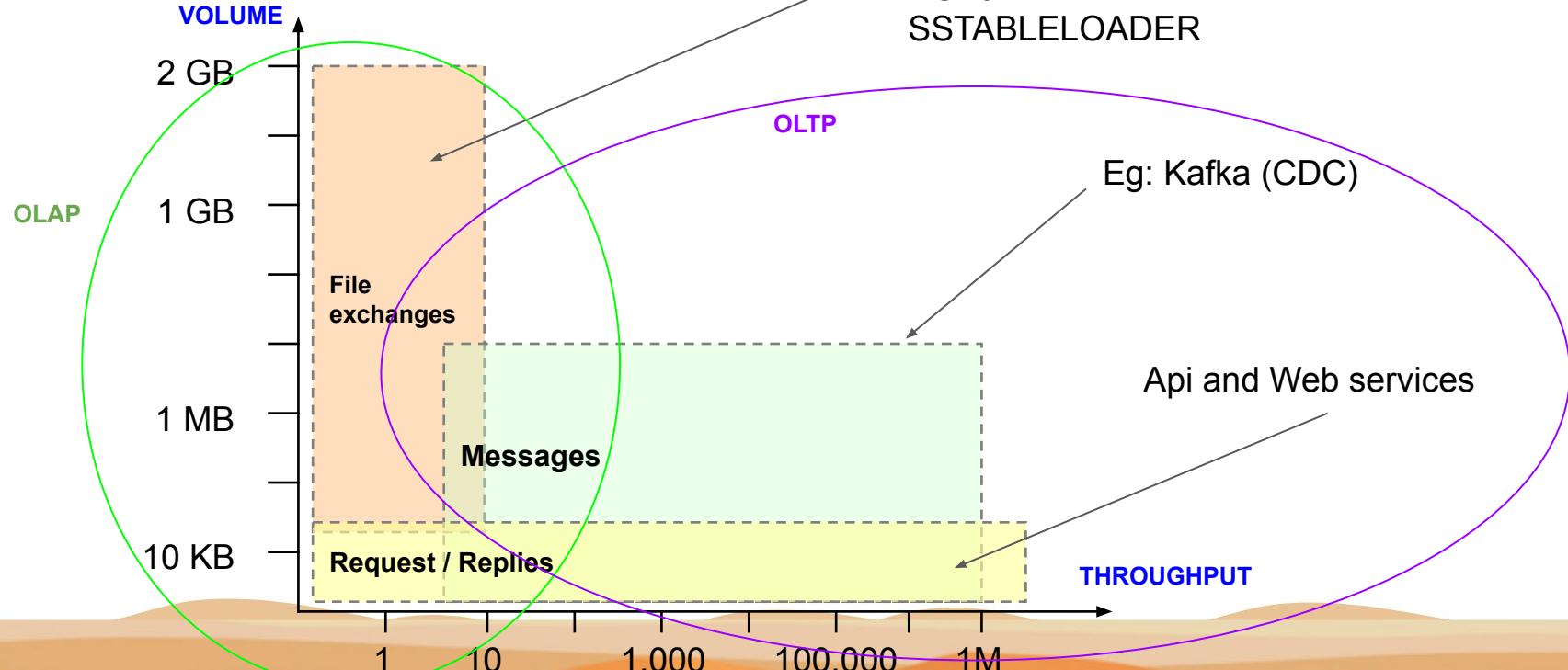


[bit.ly/CassandraWorkshopMaterials](https://bit.ly/CassandraWorkshopMaterials)

# Integrations Patterns



Cqlsh "COPY FROM"  
DSBulk  
SSTABLELOADER



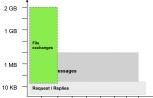
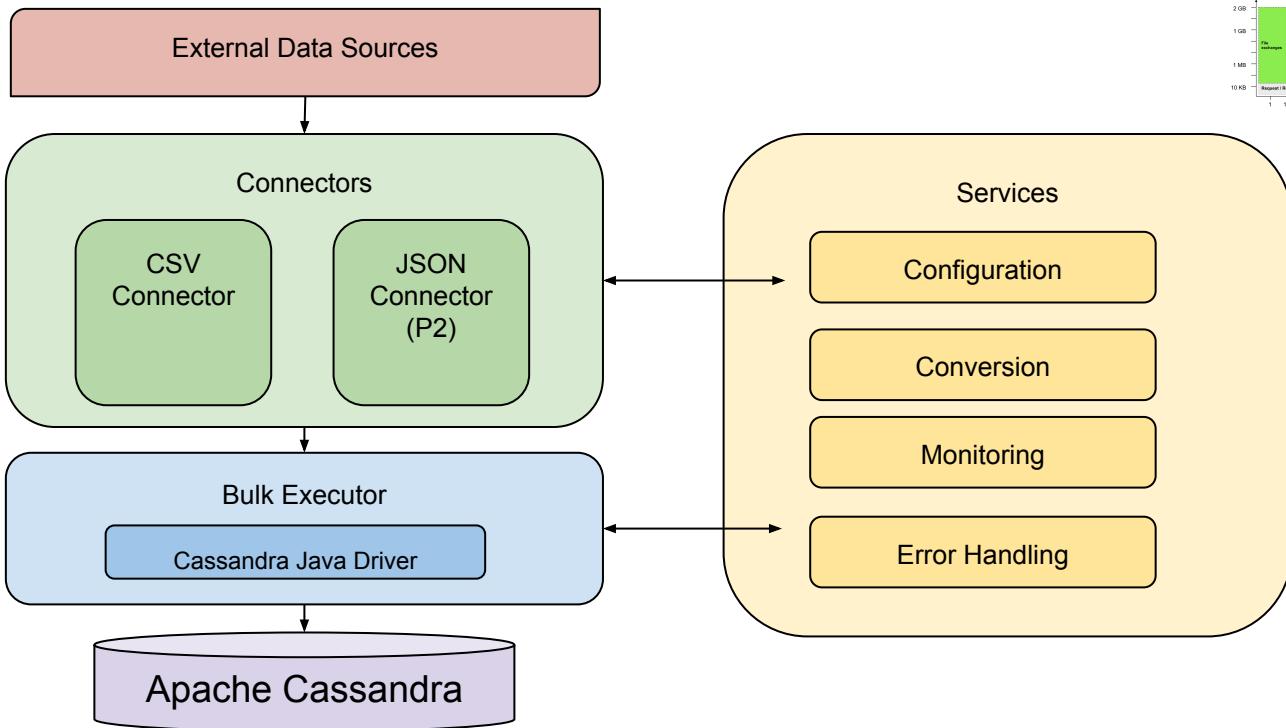
MATERIALS



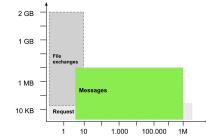
[bit.ly/CassandraWorkshopMaterials](http://bit.ly/CassandraWorkshopMaterials)



# DSBulk

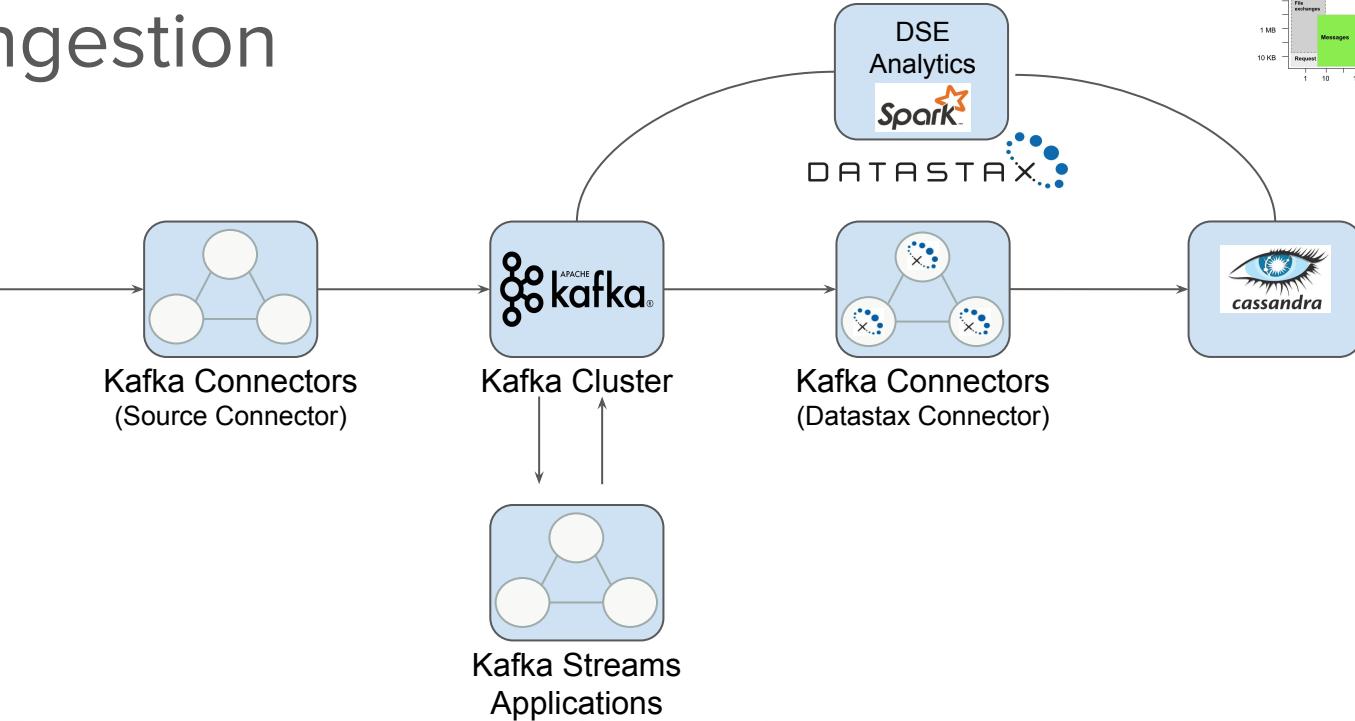


MATERIALS → [bit.ly/CassandraWorkshopMaterials](https://bit.ly/CassandraWorkshopMaterials)

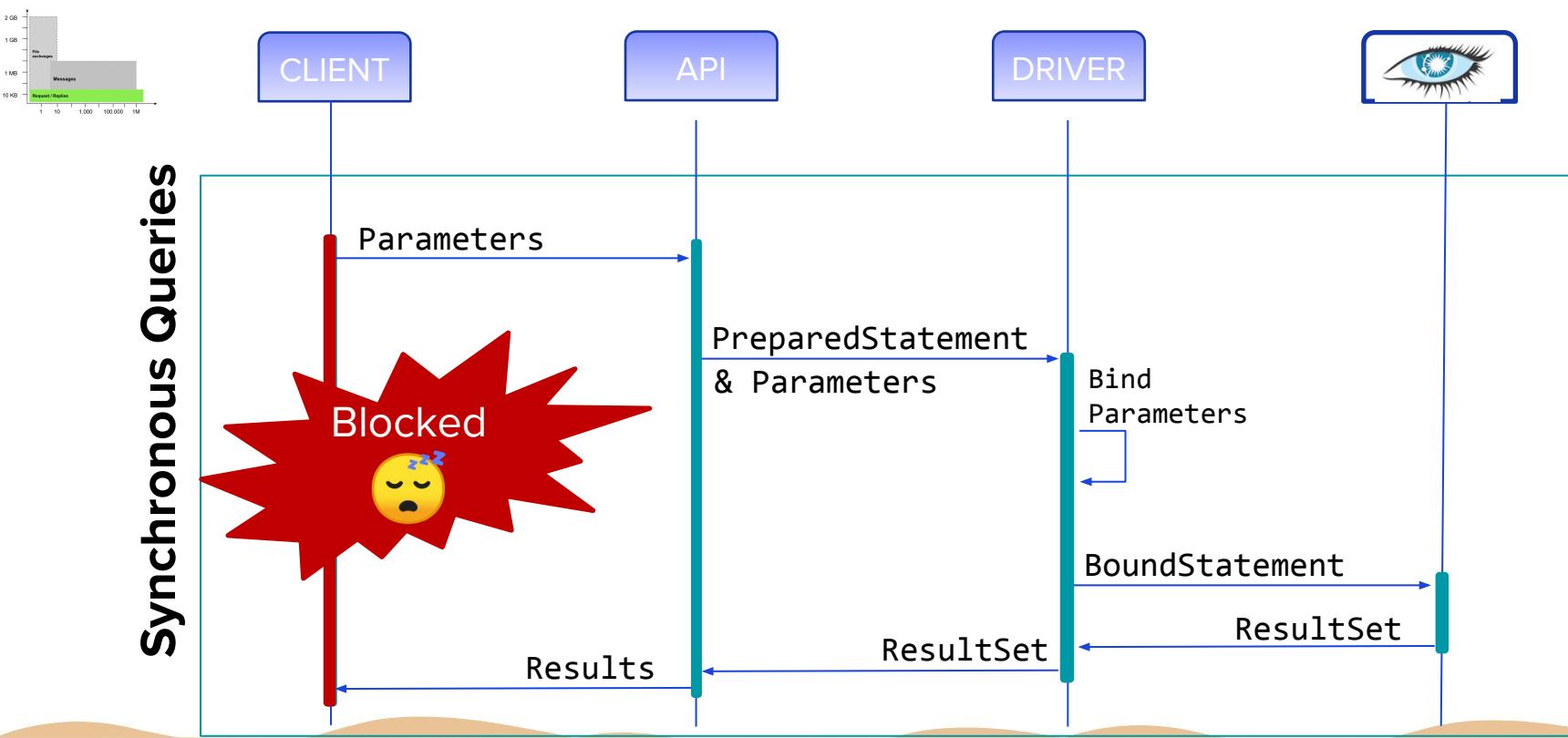


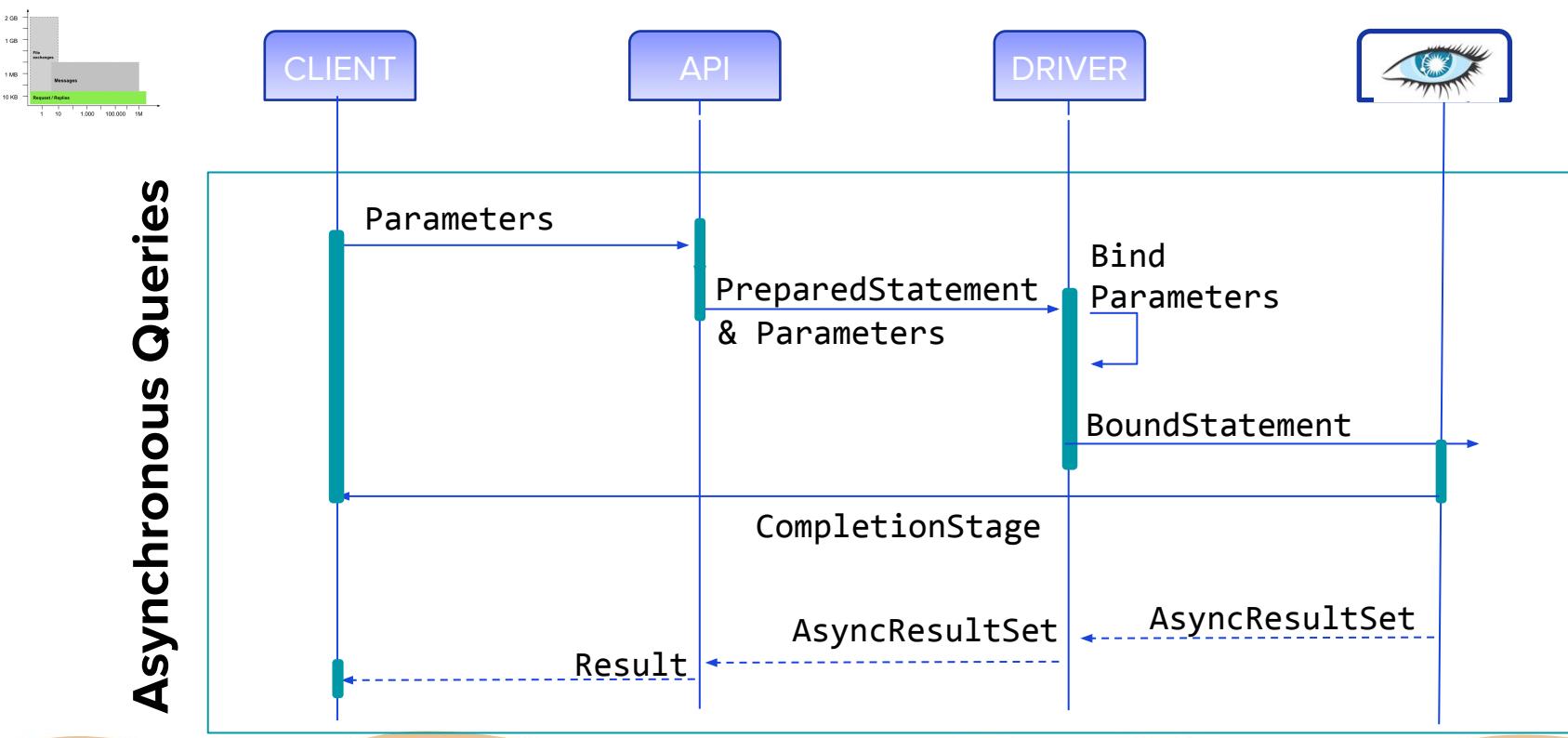
# Message Ingestion

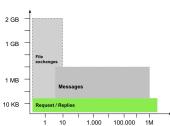
Data Sources



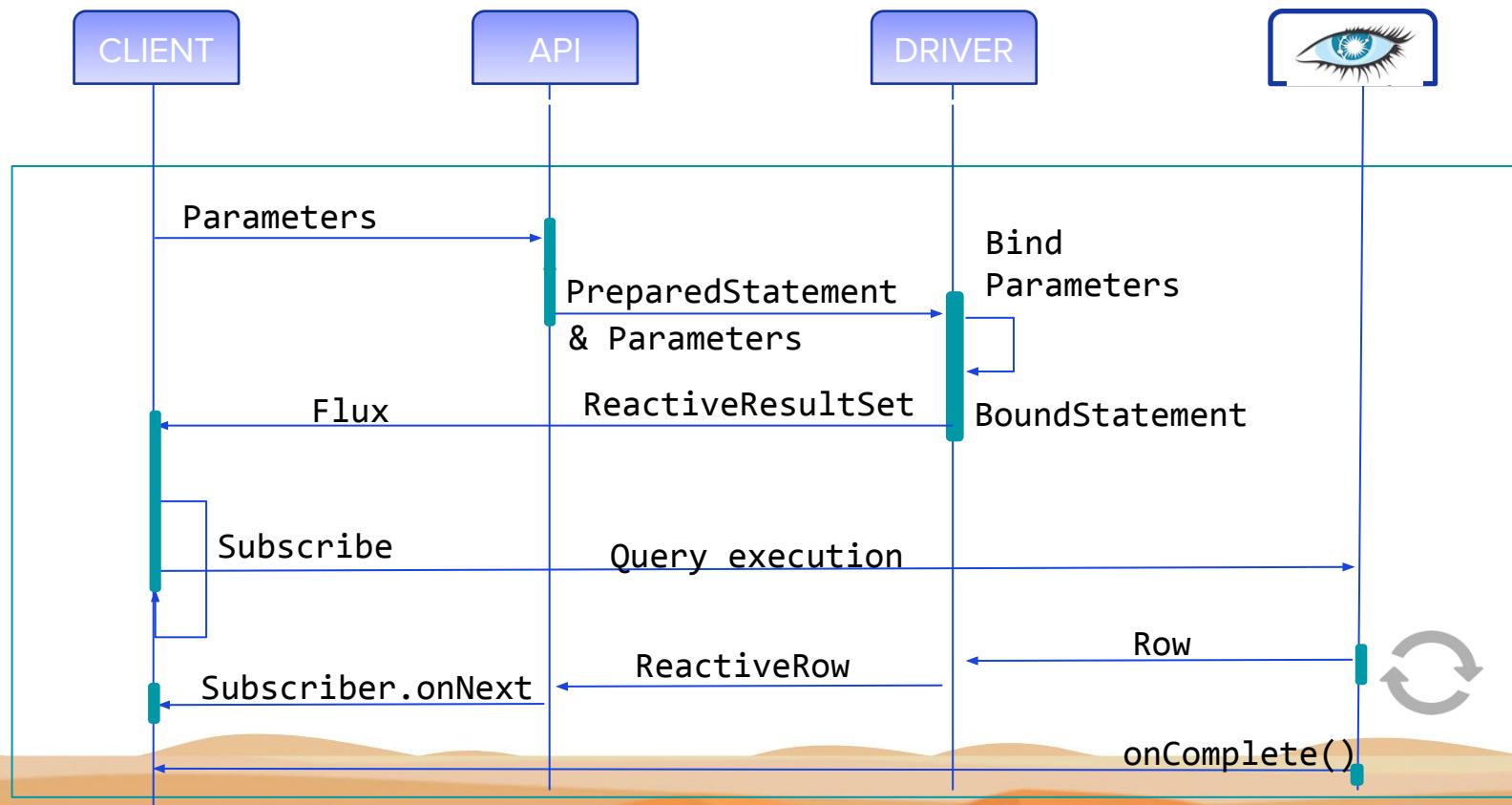
MATERIALS → [bit.ly/CassandraWorkshopMaterials](https://bit.ly/CassandraWorkshopMaterials)







## Reactive Queries



MATERIALS

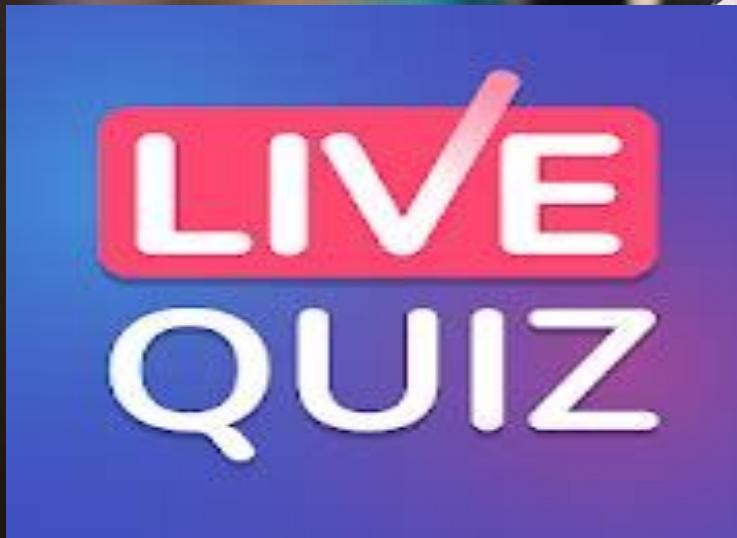


[bit.ly/CassandraWorkshopMaterials](http://bit.ly/CassandraWorkshopMaterials)



# menti.com

# 43 69 82



Available on the iPhone  
**App Store**

GET IT ON  
**Google play**

# Developer Workshop Series **Week4**



- Housekeeping
- Cassandra and Microservices
- Endpoint and Access Patterns
- API : Rest, GraphQL, GRPC
- Introduction to Quarkus (Java)
-  Surprise !!

MATERIALS



[bit.ly/CassandraWorkshopMaterials](https://bit.ly/CassandraWorkshopMaterials)



# GraphQL



## Shipping Faster

Don't write a ton of code or rely on new rigid API endpoints when you want to fetch data for a new screen in an app.



## Better Apps

Build features with the best data and services possible, not the API available that day. GraphQL helps you put personalization into every app.



## Parity Across Platforms

Make your apps consistent across all channels. Move data-related functionality that is common between platforms into the shared GraphQL layer.



## Powerful Partner APIs

Get new partners onto your API without custom changes, while ensuring a high-quality experience for your mutual customer.



## Visibility Into Your Data

Give your development teams real-time visibility into exactly what services are available for them to build on top of.



## Positive Control

Get a single point of control to secure and analyze all access to your data and see how it's used.

MATERIALS



[bit.ly/CassandraWorkshopMaterials](https://bit.ly/CassandraWorkshopMaterials)

# What is gRPC?

<https://grpc.io/>



gRPC is a modern open source high performance RPC framework that can run in any environment.

- **Simple service definition:** Define your service using Protocol Buffers, a powerful binary serialization toolset and language
- **Start quickly and scale:** Install runtime and dev environments with a single line and also scale to millions of RPCs per second with the framework
- **Works across languages and platforms:** Automatically generate idiomatic client and server stubs for your service in a variety of languages and platforms
- **Bi-directional streaming and integrated auth:** Bi-directional streaming and fully integrated pluggable authentication with HTTP/2-based transport

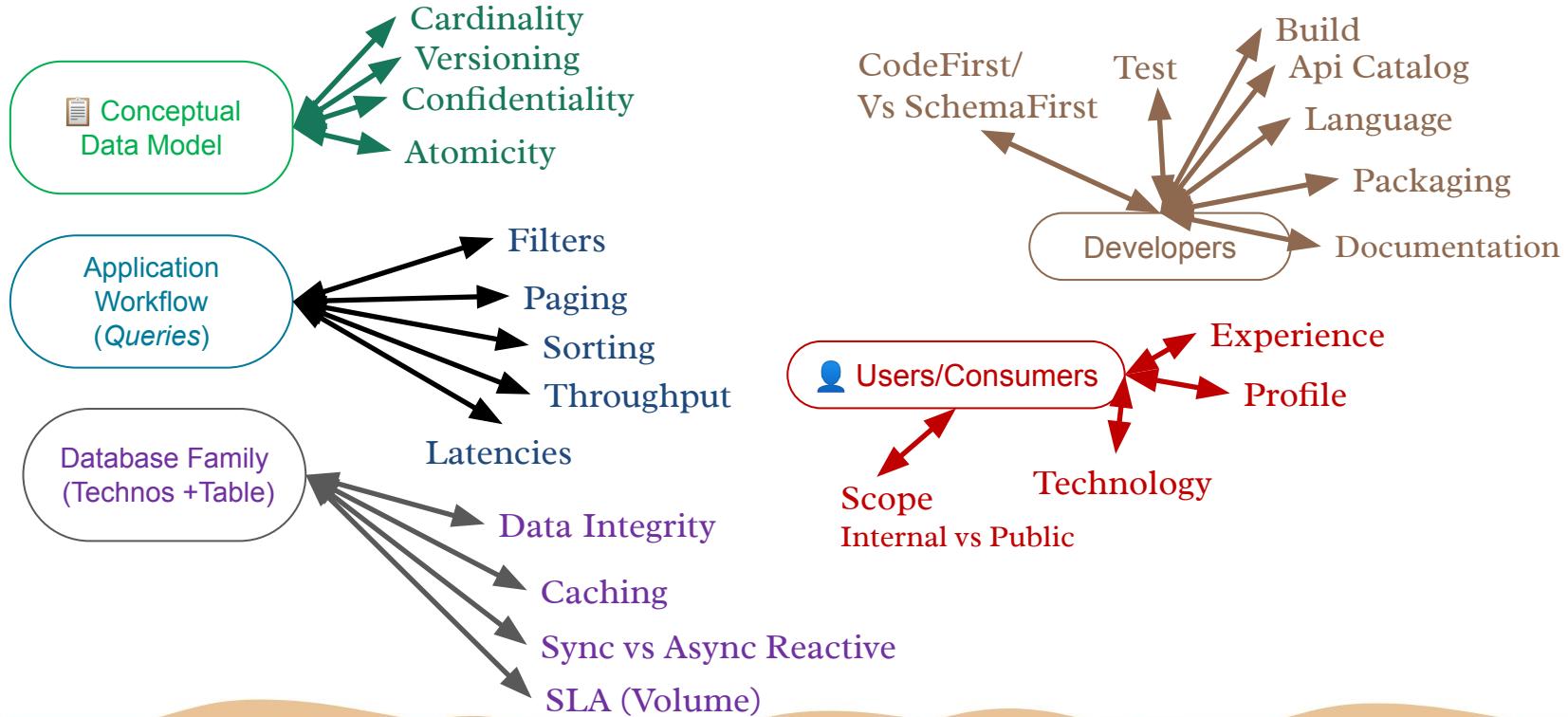


MATERIALS



[bit.ly/CassandraWorkshopMaterials](http://bit.ly/CassandraWorkshopMaterials)





MATERIALS



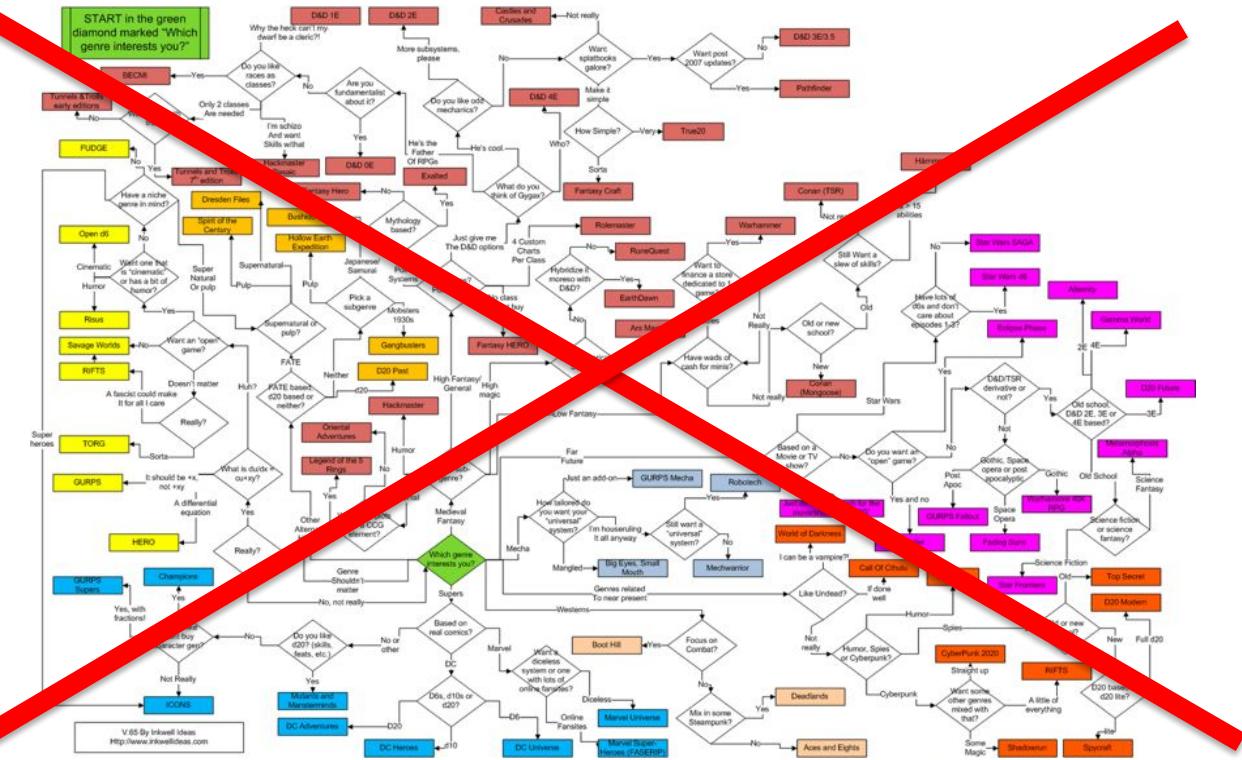
[bit.ly/CassandraWorkshopMaterials](http://bit.ly/CassandraWorkshopMaterials)

# Analysis Matrix

Concept	GraphQL	REST	gRPC
<b>Modèle conceptuel de données (Entités, Relations)</b>			
Cardinalité	1	5	3
Versionning	1	8	9
Confidentialité	6	9	1
Atomicité	6	7	0
<b>Parcours Applicatifs</b>			
Filtres	2	3	4
Pagination	3	3	6
Tris	4	4	6
Throughput	4	3	6
Latence	6	4	6
Base de données	2	3	6
Intégrité	6	4	7
Cache	7	0	6
Sync vs Async vs Reactive	1	2	4
SLA	6	3	4
<b>Development</b>			
Language	1	4	5
Api Catalog	6	7	8
Test	3	2	0
Build	4	5	6
Packaging	4	6	7
Code first / Schema Frist	5	7	8
Documentation	2	5	1
<b>Client</b>			
XP	1	4	7
Profil	3	3	9
Techno	4	5	7
Interne vs Extern	4	5	8

MATERIALS → [bit.ly/CassandraWorkshopMaterials](https://bit.ly/CassandraWorkshopMaterials)

# Decision Tree



MATERIALS → [bit.ly/CassandraWorkshopMaterials](https://bit.ly/CassandraWorkshopMaterials)

# { REST }



- ❖ Decoupling Client / Server (*Schema on read*)
- ❖ Flexibility: Sync, Async, Reactive + Multi payload
- ❖ Api Lifecycle (*Versioning*)
- ❖ Tooling (*API Management, Serverless*)
- ❖ Verbose payloads (*json, xml*)
- ❖ No discoverability
- ❖ Not best fit for command-like (functions) API (RPC)



- ❖ CRUD superstar
- ❖ Relevant for OLTP mutations and statuses
- ❖ Public and web APIs



- ❖ High Performances (*http/2 – binary serialisation*)
- ❖ Multiple stubs : Sync, Async, Streaming
- ❖ Multi languages (*Interoperability*)



- ❖ Strongly coupled (*schema with proto files*)
- ❖ No discoverability
- ❖ *Protobuf* serialization format



- ❖ Distributed network of services (no waits)
- ❖ High throughput & streaming use cases
- ❖ Command-like, RPC



# GraphQL



- ❖ Discoverability, documentation
- ❖ Custom payloads
- ❖ Match standards (Json | Http)



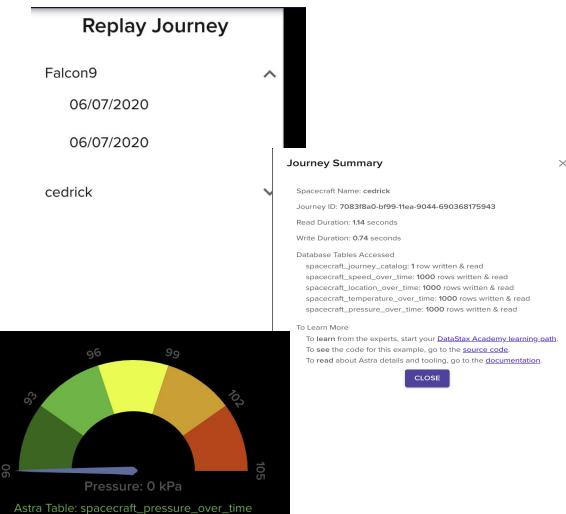
- ❖ Single endpoint (*versioning, monitoring, security*)
- ❖ Complex implementation (*tooling, still young*)
- ❖ Nice for customers nasty for DB (*N+1 select*)



- ❖ BFF : Backend for frontend
- ❖ Service aggregation | composition (*joins*)
- ❖ When bandwidth matters (*mobile phones*)

# Designing REST API with Application Workflow

- **Space crafts catalog queries**
  - Look up all of the journeys for a spacecraft
  - Look up the state of a journey
  - Create a new journey
- **Sensor readings queries** : Speed, Pressure, Temperature, Location
  - Save readings over time
  - Analyze each dimension independently
  - Analyze data per journey
  - Less than 100.000 records per journey per dimension



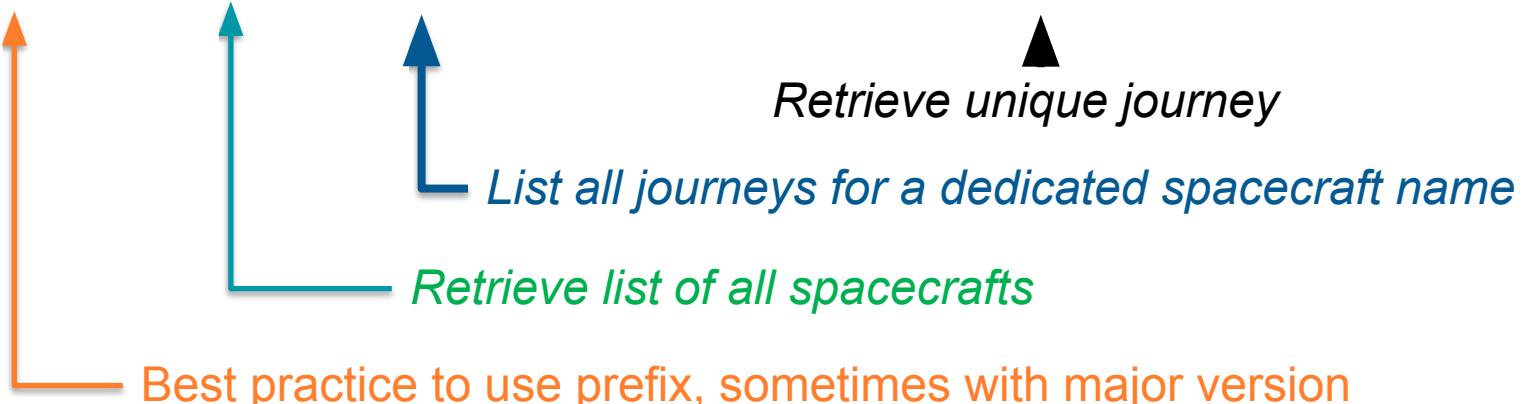
MATERIALS



[bit.ly/CassandraWorkshopMaterials](https://bit.ly/CassandraWorkshopMaterials)

# Designing your API : Journey Catalog

`/api/spacecraft/{spaceCraftName}/{journeyId}`



# Designing your API : Journey Catalog

Attribute	Value
<b>Method</b>	GET
<b>URL</b>	/api/v1/spaceship/{spaceshipName}
<b>Request</b>	We consider no paging (could add request queries)
<b>Response</b>	<ul style="list-style-type: none"><li>• 200 List of journeys</li><li>• 204 No content</li><li>• 400 Bad parameter</li><li>• 404 spaceship not found</li><li>• 500 Internal Error</li></ul>

# Designing your API : Journey Catalog

Attribute	Value
Method	GET
URL	/api/v1/spacecraft/{spacecraftName}/{journeyId}
Request	N/A
Response	<ul style="list-style-type: none"><li>• 200 The journey</li><li>• 400 Bad parameter</li><li>• 404 Journey or spacecraft not found</li><li>• 500 internal Error</li></ul>

# Designing your API; Journey Catalog

## spacecraft-controller Operations on spacecrafts catalog

GET

/api/spacecraft List all spacecrafts and journeys

GET

/api/spacecraft/{spacecraftName} List all journeys for a dedicated spacecraft name

POST

/api/spacecraft/{spacecraftName} Create a new Journey for a Spacecraft

GET

/api/spacecraft/{spacecraftName}/{journeyId} Retrieve a journey from its spacecraftname and journeyid

MATERIALS



[bit.ly/CassandraWorkshopMaterials](https://bit.ly/CassandraWorkshopMaterials)



# Designing your API

**instruments-controller** Works with Instruments

**GET** /api/spacecraft/{spacecraftName}/{journeyId}/instruments/location Retrieve location reading for a journey

**POST** /api/spacecraft/{spacecraftName}/{journeyId}/instruments/location Save location reading for a journey

**GET** /api/spacecraft/{spacecraftName}/{journeyId}/instruments/pressure Retrieve pressure reading for a journey

**POST** /api/spacecraft/{spacecraftName}/{journeyId}/instruments/pressure Save pressure reading for a journey

**GET** /api/spacecraft/{spacecraftName}/{journeyId}/instruments/speed Retrieve speed reading for a journey

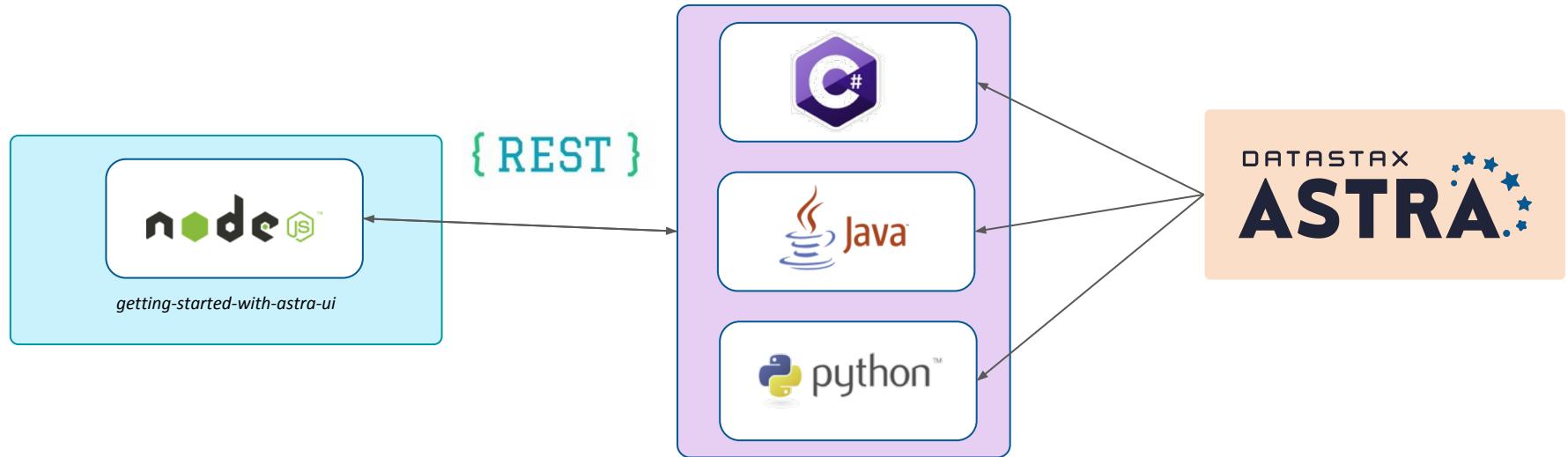
**POST** /api/spacecraft/{spacecraftName}/{journeyId}/instruments/speed Save speed reading for a journey

**GET** /api/spacecraft/{spacecraftName}/{journeyId}/instruments/temperature Retrieve temperature reading for a journey

**POST** /api/spacecraft/{spacecraftName}/{journeyId}/instruments/temperature Save temperature reading for a journey

MATERIALS → [bit.ly/CassandraWorkshopMaterials](https://bit.ly/CassandraWorkshopMaterials)

# Logical Architecture



MATERIALS → [bit.ly/CassandraWorkshopMaterials](https://bit.ly/CassandraWorkshopMaterials)

# Exercise



Start UI and demo



## Application Development - API REST

Cassandra Workshop Series are an interactive experience. Datastax Developer Advocates share some knowledge about Apache Cassandra™ NoSQL database and how you build Cloud Native applications. You interact with them through chats ([youtube](#) and [discord](#)), quizzes ([menti.com](#)), and exercises.

In this repository, you'll find everything you need related to week 3 of the Cassandra Workshop Series. For simplicity all exercise instructions will be listed in a single README for each language: Java, Javascript, Python, CSharp. We recommend you to have some previous experience with the language you choose.

Now if you are NOT familiar with any of those, we recommended to use Python, the is the simplest. Ready ? go !

### Workshop Materials

Materials	Description and Links
Slidedeck	<a href="#">SLIDEDECK</a>
Chat with us on Discord	<a href="#">DISCORD</a>
Ask Question during week	<a href="#">COMMUNITY</a>

A screenshot of a GitHub repository page for 'Application Development - API REST'. The title bar says 'Application Development - API REST'. Below the title is the same cartoon illustration of people with lightsabers. The main content area shows a file tree for 'README.MD' with sections like 'week1', 'week2', etc., and a 'gitpod.yml' file. A sidebar on the left shows repository stats: 'Gitpod ready-to-code', 'license Apache-2.0', and 'chat 182 online'. The bottom of the page shows a terminal window with the command 'gitpod /workspace/cassandra-workshop-series \$'.



MATERIALS → [bit.ly/CassandraWorkshopMaterials](https://bit.ly/CassandraWorkshopMaterials)

# Developer Workshop Series **Week4**



What we will cover:

- Housekeeping
- From Microservices to Cloud Native
- Access Patterns and Api Endpoints
- REST vs gRPC vs GraphQL vs Messages
- Introduction to Quarkus (Java)
-  Surprise !!

MATERIALS



[bit.ly/CassandraWorkshopMaterials](https://bit.ly/CassandraWorkshopMaterials)



QUARKUS

# Introducing Quarkus (quarkus.io)

- **First Class Support for Graal/SubstrateVM**
  - When an application is compiled down to a native image it starts much faster and can run with a much smaller heap than a standard JVM.
- **Build Time Metadata Processing**
  - As much processing as possible is done at build time, so your application will only contain the classes that are actually needed at runtime. In a traditional model all the classes required to perform the initial application deployment hang around for the life of the application, even though they are only used once.
- **Reduction in Reflection Usage**
  - As much as possible Quarkus tries to avoid reflection, reducing startup time and memory usage.
- **Native Image Pre Boot**
  - When running in a native image Quarkus pre-boots as much of the framework as possible during the native image build process. This means that the resulting native image has already run most of the startup code and serialized the result into the executable, resulting in even faster startup.



MATERIALS

[bit.ly/CassandraWorkshopMaterials](https://bit.ly/CassandraWorkshopMaterials)

 + 



# Quarkus Cassandra Extension Features

QUARKUS

- Native Quarkus Config
- Cassandra Driver Session Support
- Cassandra Driver Object Mapper Support
- Support for Mutiny Types (Reactive Types)
- Native Image Support
- Support for Astra (Cassandra dbaaS by DataStax)



MATERIALS



[bit.ly/CassandraWorkshopMaterials](https://bit.ly/CassandraWorkshopMaterials)



# Developer Workshop Series **Week4**



- Housekeeping
- From Microservices to Cloud Native
- Access Patterns and Api Endpoints
- REST vs gRPC vs GraphQL vs Messages
- Introduction to Quarkus (Java)
-  Surprise !!

MATERIALS



[bit.ly/CassandraWorkshopMaterials](https://bit.ly/CassandraWorkshopMaterials)

# Let's run an (Astra) HACKATHON

- With those 4 weeks **YOU** have the knowledge to go and start building your applications with Cassandra...**so let's do it !**
- Today we are launching the **ASTRAKATHON**
- 4 weeks (*end of submission August 26th 12pm PDT*)
  - BUILD a REAL app with ASTRA
  - Pick the language you are confident with
  - All source code will be shared with everyone on github
  - Real Money to win
  - Judges: P.McFadin, J Carpenter and whole advocate team



**WE WANT YOU**

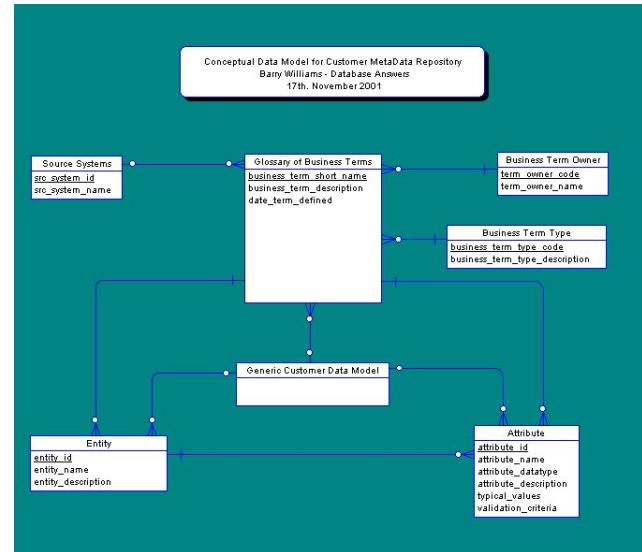
# What do I win ?



MATERIALS → [bit.ly/CassandraWorkshopMaterials](http://bit.ly/CassandraWorkshopMaterials)

# How do I get evaluated ?

- We will assess :
  - **The data model** : choices and hypothesis
  - **The source code** : quality and best practices
  - **Ease of use** (documentation, docker?) - should be launchable in 5 mins per docs
- No idea? Check this [data models collection](#), pick one and make it fit Cassandra
- We have created an alias for submissions :
  - [astrakathon@datastax.com](mailto:astrakathon@datastax.com)
  - Need to be registered to the series



# We are upgrading our game !

1. We are upgrading ACADEMY:
  - a. Migrating platform to real LMS
  - b. Introducing SSO
  - c. It will be offline for a few hours starting :  
16:00 Pacific | 19:00 Eastern | 20:00 Brazil |  
00:00 UK | 01:00 CEST | 04:30 IST | 07:00  
China | 09:00 Australia
2. We are changing certifications provider
  - a. No vouchers and exams schedule for the next 2 weeks.



The image shows the DataStax Academy landing page. The header features the text "DATASTAX ACADEMY" and a subtext: "Take advantage of the largest free library of technical courses about Apache Cassandra™ and DataStax products and tools." Below the header are three main call-to-action buttons: "Start Learning" (orange), "Browse by Topics" (purple), and "Get Certified" (dark blue). Each button has a corresponding icon: a play button, a list, and a checkmark.

HOT!



## 4.0 BETA: THE MOST STABLE RELEASE IN HISTORY

If you use a website or a smartphone today, you're touching a Cassandra-backed system.

## WHAT'S IN THE BETA?

The community contributed 1,000+ bug fixes, improvements and new features:

### 5X FASTER SCALING OPERATIONS

In cloud and Kubernetes environments with Zero Copy Streaming

### ENTERPRISE-GRADE SECURITY & OBSERVABILITY

With real time Audit Logging and Traffic Replay

### IMPROVING NODE AVAILABILITY

With Java 11 support and ZGC

### WHAT MAKES IT STABLE?

Performance tests ran on 200 node clusters and 100s of real world use cases and schemas:

- ✓ Replay testing
- ✓ Fuzz testing and property-based testing
- ✓ Distributed tests and fault-injection testing
- ✓ Upgrade testing

### ECOSYSTEM SUPPORT



### GET STARTED

DOWNLOAD: CASSANDRA.APACHE.ORG

### JOIN THE COMMUNITY



MATERIALS



[bit.ly/CassandraWorkshopMaterials](https://bit.ly/CassandraWorkshopMaterials)



# Homework Week 4 (cf community)

## 1. Learn

- a. Watch talk introducing Kafka and Cassandra (29min)
- b. Short course Kafka and Cassandra on academy (2H)
- c. (optional) Watch Accelerate Series, this is **FUN**
- d. (optional) [Webinar 10 tricks with Astra](#)



## 2. Practice

- a. Finish workshop exercises, if needed following guidance on Github.
- b. (optional) Quarkus example
  - i. <https://github.com/datastax/cassandra-quarkus/tree/master/quickstart>

## 3. Validation form of the week



MATERIALS



[bit.ly/CassandraWorkshopMaterials](http://bit.ly/CassandraWorkshopMaterials)



# Thank You



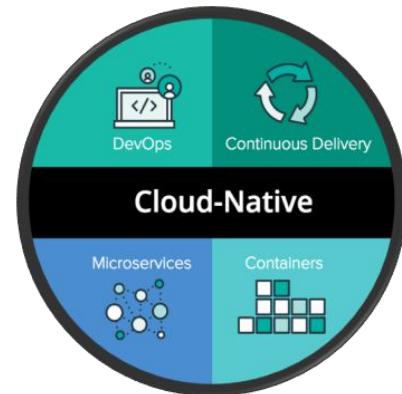
# Cloud Native Application 101

“Cloud native is an approach to building and running applications that fully exploit the advantages of the cloud computing model.” Source: What are Cloud-Native Applications?

- Pivotal

- **Containerized.** Each part (applications, processes, etc) is packaged in its own container. This facilitates reproducibility, transparency, and resource isolation.
- **Dynamically orchestrated.** Containers are actively scheduled and managed to optimize resource utilization.
- **Microservices-oriented.** Applications are segmented into microservices. This significantly increases the overall agility and maintainability of applications.”

More than just signing up with a cloud provider and using it to run your existing applications. It affects the **design, implementation, deployment, and operation** of your application.





# Building Cloud Native Applications

**Microservice architecture at its core :** Applications are segmented into microservices. Each service is independent, stateless and delegate data to storage unit.

**Container Oriented :** Each part is packaged in its own container.

- Stateless + Externalization of configuration
- Small footprint stack
- Standard endpoints to be invoker
- Standard monitoring to be administrate

**Functions oriented :Eventually split endpoints & implementations:**

- Going even smaller to the function levels.
- ENDPOINTS + IMPLEMENTATIONS
- SERVERLESS

