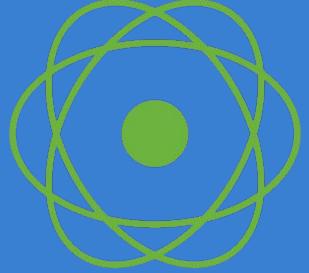
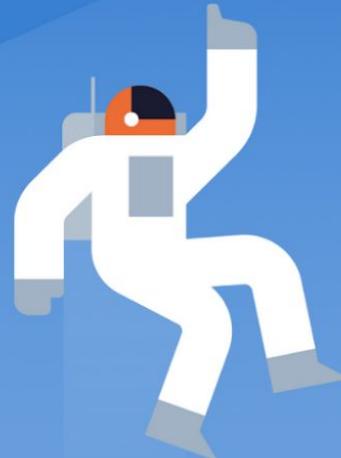


DataStax



# Build an Appointment Scheduling application for a Petclinic



LEVEL  
**UP**  
with the

DataStax

**Developers**



# Cédrick Lunven

## Director Developer Relations



- Trainer
- Public Speaker
- Developers Support
- Developer Applications
- Developer Tooling
  
- Creator of ff4j (ff4j.org)
- Maintainer for 8 years+
  
- Happy developer for 14 years
- Spring Petclinic Reactive & Starters
- Implementing APIs for 8 years



@clunven

# David Gilardi

## Senior Developer Advocate



**Developer Advocate,  
Major and our HOST  
TODAY**



- Training
- Teaching
- Reference Applications
- Support
- Public Speaking

➤ Certified !

➤ Advanced User



@SonicDMG



@SonicDMG

DataStax Developers

TO  
REVIEW  
BY DAVID

# DatastaxDevs: Developer Advocates Team



Cedrick  
Lunven

Aleksandr  
Volochnev

Jack  
Fryer

Kirsten  
Hunter

Stefano  
Lottini

David  
Gilardi

Ryan  
Welford

Rags  
Srinivas

Sonia  
Siganporia

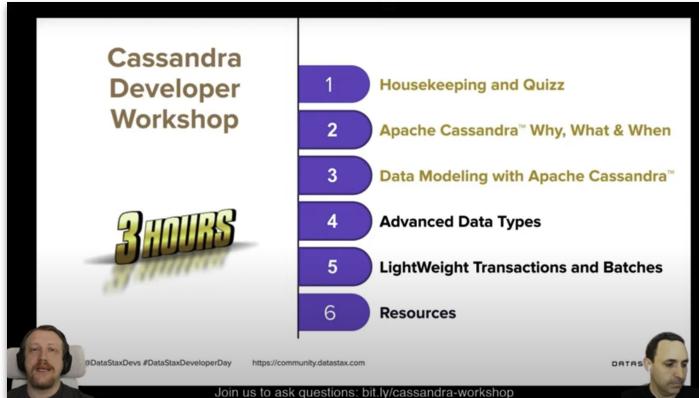
R

S

# Housekeeping #1: Attending the workshop

Live and interactive

**Livestream:** [youtube.com/DataStaxDevs](https://youtube.com/DataStaxDevs)



**YouTube**



**Twitch**

**Questions:** <https://dtsx.io/discord>



**Discord**

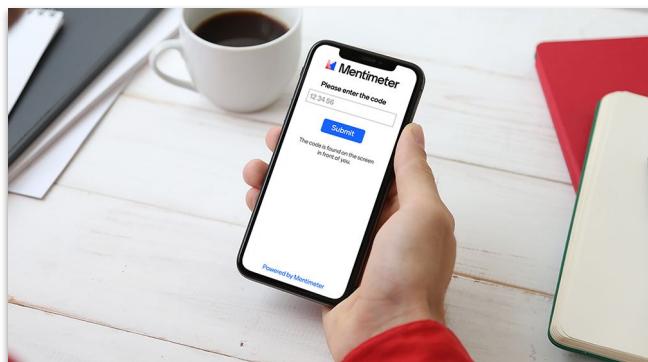


**YouTube**



Available on the iPhone  
App Store

**Games** [menti.com](https://menti.com)



**Mentimeter**

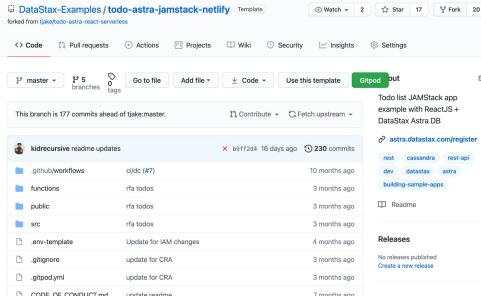


GET IT ON  
Google play

# Housekeeping #2: Doing Hands-On

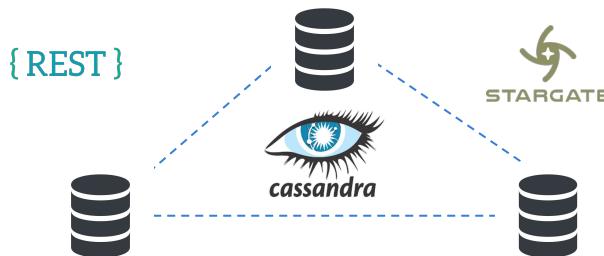
Nothing to install !

Source code + exercises + slides



GitHub

Database + GraphQL + PlayGround



DataStax

**Astra DB**

IDE

A screenshot of an IDE interface showing a Java file named 'StargateDemoApplication.java'. The code is as follows:

```
1 package com.datastax.demo.stargate;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5
6 @SpringBootApplication
7 public class StargateDemoApplication {
8
9     public static void main(String[] args) {
10         SpringApplication.run(StargateDemoApplication.class, args);
11     }
12 }
13
14 }
```

The IDE interface includes tabs for PROBLEMS, OUTPUT, TERMINAL, and DEBUG CONSOLE, with 'gitpod' listed under the TERMINAL tab.

Gitpod



**Maven**<sup>TM</sup>

**npm**

The Node.js logo, consisting of the word 'node' in a green hexagon and '.js' in a smaller hexagon.

**DataStax Developers**

# Get your Badge for today !



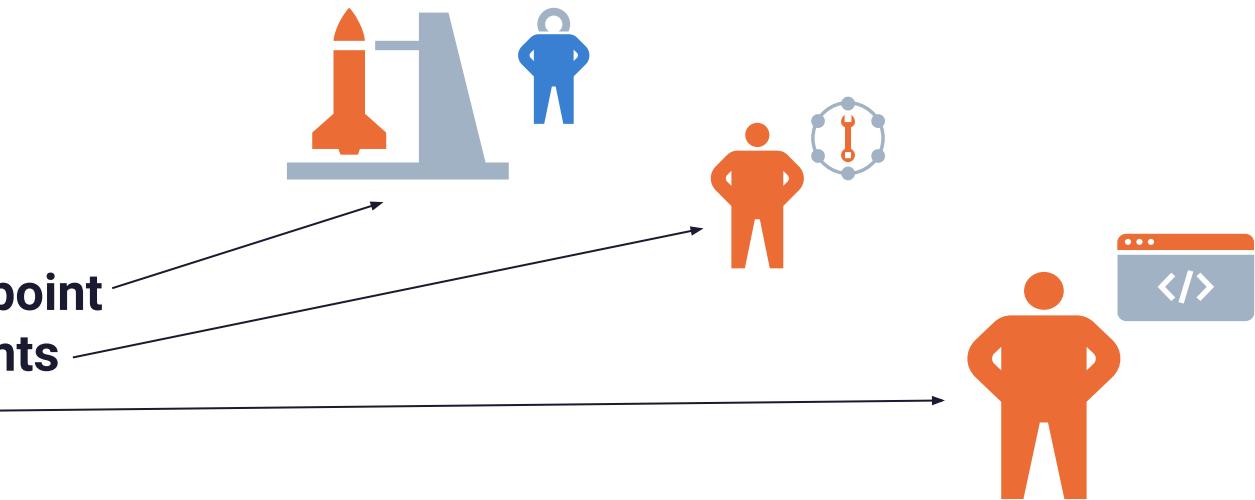


<https://dtsx.io/staxgiving>

500\$ + 250\$ + 250\$ AMAZON gift cards to win

#### CONDITIONS

- **#engage-with-us** : Tasks that will earn you 1 point
- **#work-with-us**: Tasks that will earn you 5 points
- **#build** : Tasks that will earn you 10 points.



#### DUE DATE

THANKSGIVING DAY => 25/11

Winners randomly chosen **live** on 12/8

# Agenda

01



PetClinic  
Architecture & Use Case

02



Cassandra Database  
The Art of Data Modelling

03



Reactive Drivers  
Reactive vs Async

04



Spring Reactive  
Boot and WebFlux

05



User Interface  
Angular

06



Game &  
Resources

# Welcome to PetClinic

Your mission is to understand and run the **Spring reference application with Cassandra as a backend with Reactive Programming with Angular user interface and without installing anything**

You have 2 hours  
Good LUCK.



# What is PetClinic ?

<https://projects.spring.io/spring-petclinic/>

Spring Petclinic

PetClinic demonstrates the use of a Spring Boot with Spring MVC and Spring Data. The PetClinic has an old and varied history dating right back to the beginning of the Spring Framework. It started life as a demonstration of nearly all the common things that you could do with Spring, back when it was possible to conceive of such a demonstration. These days it is a very small slice of what you could achieve, but the community has a spot for it, so it's nice to see it still going after all this time, so we hope you enjoy it too!

QUICK START

Name	Address	City	Telephone	Pets
Jeff Black	1450 Oak Blvd.	Monona	6085555387	Lucky
Jean Coleman	105 N. Lake St.	Monona	6085552654	Max Samantha
Betty Davis	638 Cardinal Ave.	Sun Prairie	6085551749	Basil
Harold Davis	563 Friendly St.	Windsor	6085553198	Iggy
Maria Escobito	345 Maple St.	Madison	6085557683	Mulligan
Carlos Estaban	2335 Independence La.	Waunakee	6085555487	Lucky Sly
George Franklin	110 W. Liberty St.	Madison	6085551023	Leo
Peter McTavish	2387 S. Fair Way	Madison	6085552765	George
Eduardo Rodriguez	2693 Commerce St.	McFarland	6085558763	Jewel Rosy
David Schroeder	2749 Blackhawk Trail	Madison	6085559435	Freddy

# What is Spring Petclinic Community ?

<https://spring-petclinic.github.io/>

The screenshot shows the GitHub repository page for 'spring-petclinic'. At the top left, there are 'Star' (5,186) and 'Fork' (14,079) counts, along with a 'Tweet' button. On the right, there are links for 'Docs', 'GitHub', 'Docker images', and 'Demo'. The main content area features a large green leaf icon and the text 'The Spring PetClinic Community'. Below this, it says 'Open Source sample applications based on the Spring stack'. To the right, there's a terminal window showing the command '\$ cd spring-petclinic \$ mvn package' and a 'Live Demo' button. At the bottom, there are three logos: the Spring logo ('spring by Pivotal'), the GitHub logo, and the Apache Software Foundation logo.

What is Spring PetClinic?

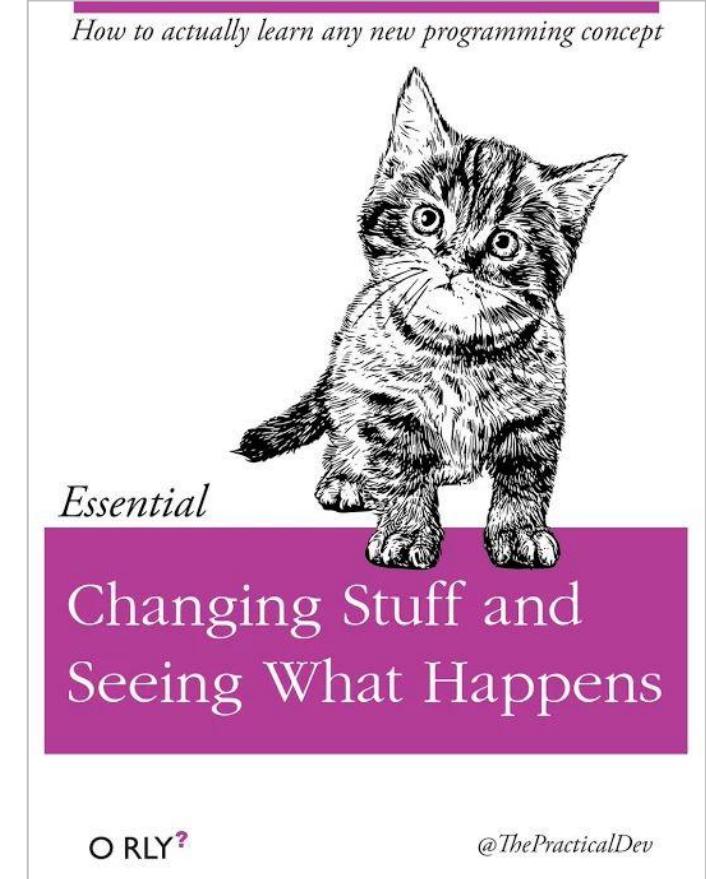
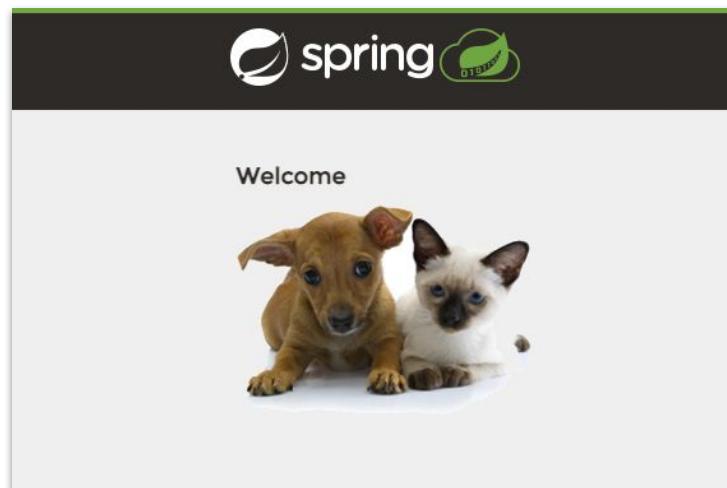
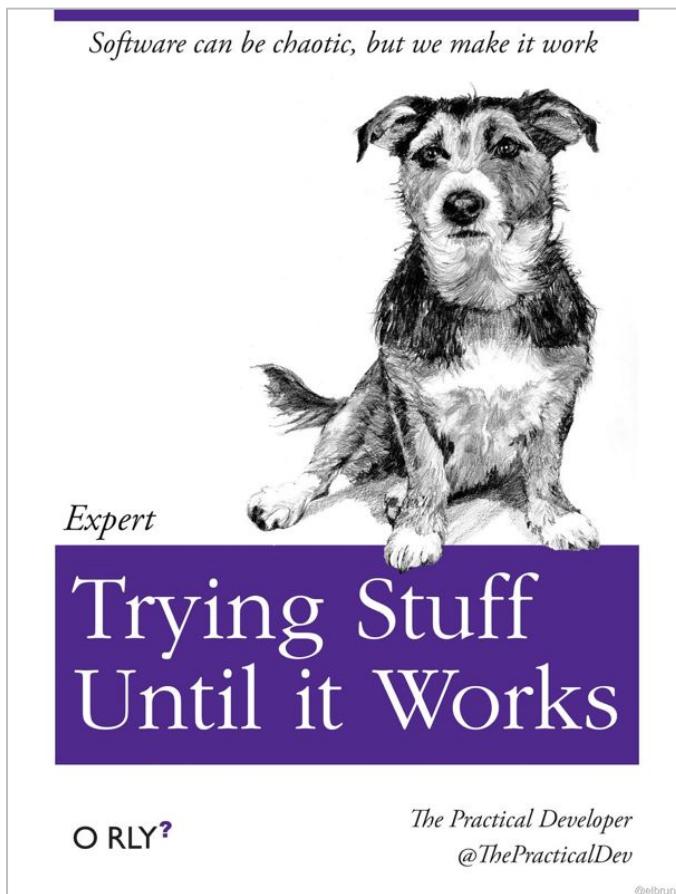
A legendary and official webapp

A GitHub Organization

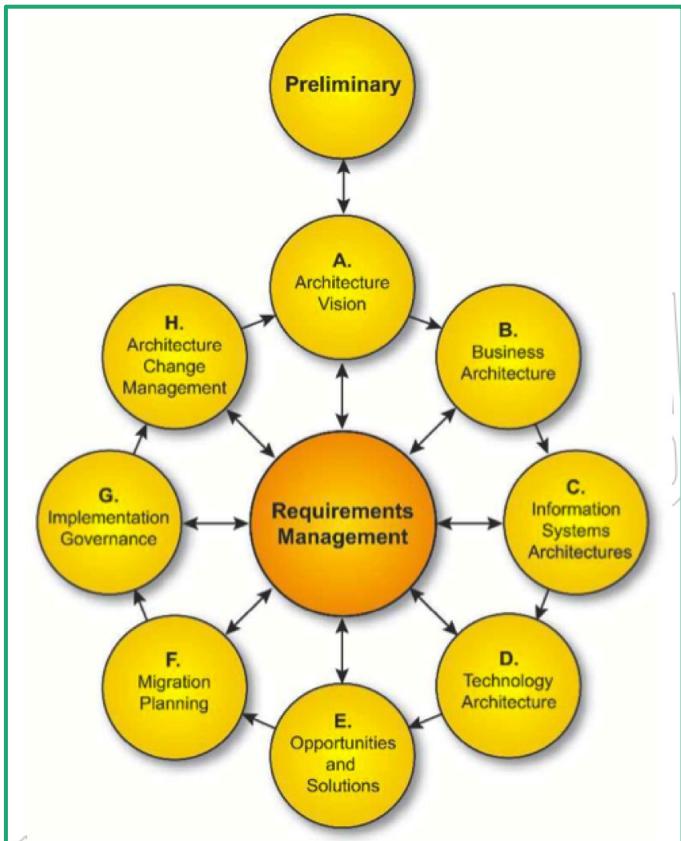
Apache 2.0 license

# Spring Pet Clinic – Demo

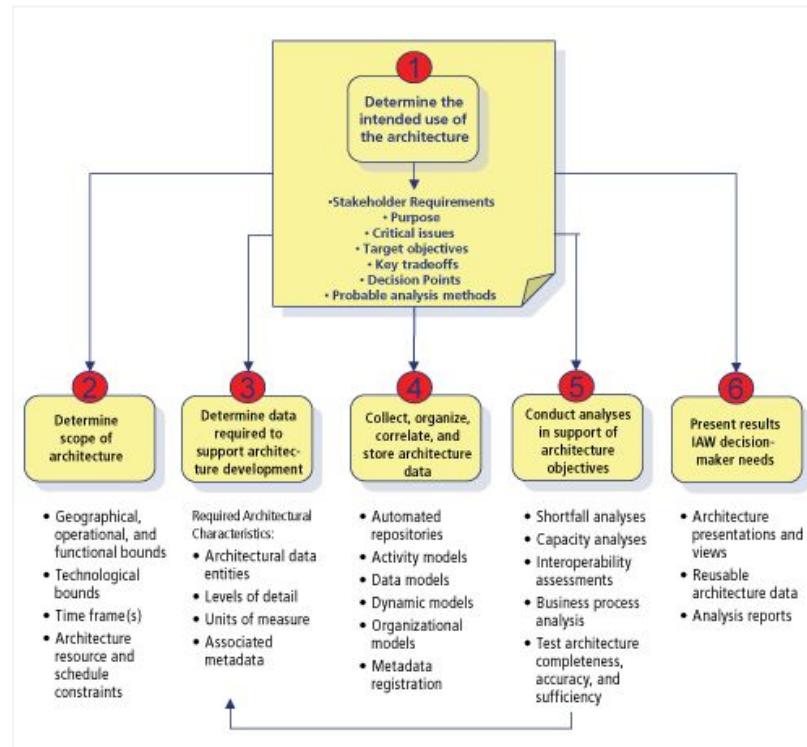
<https://spring-petclinic-community.herokuapp.com/>



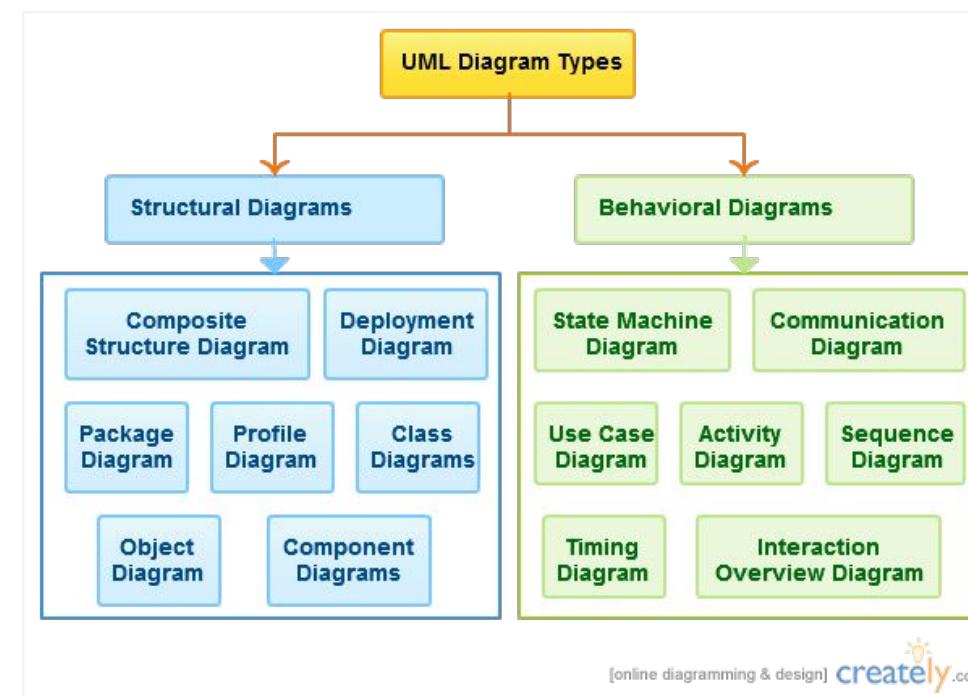
# PetClinic :: Architecture



ToGAF (Archimate)



DoDAF (DM2)

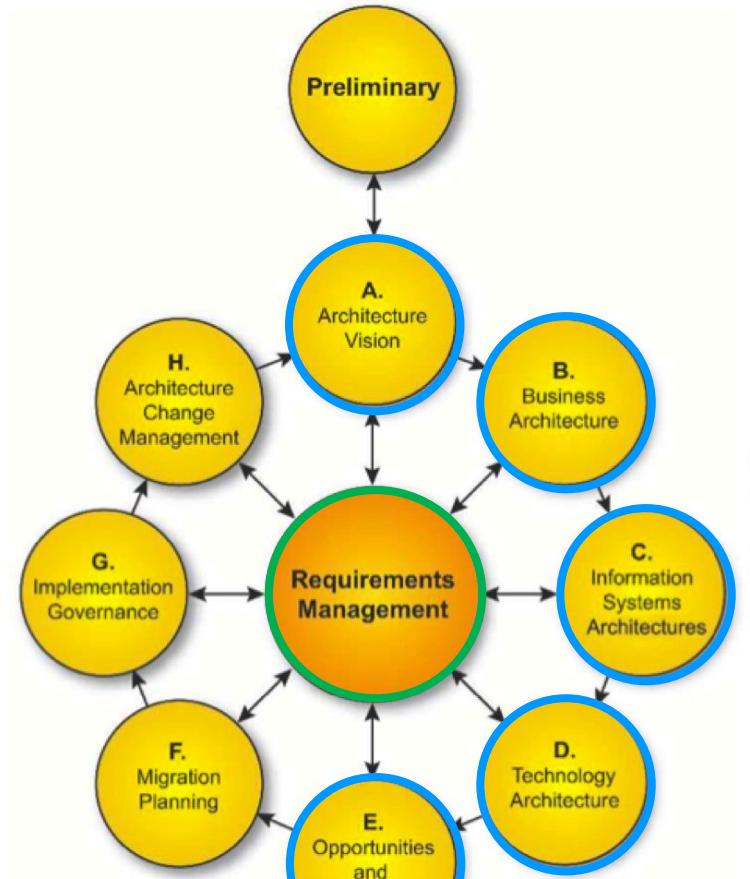
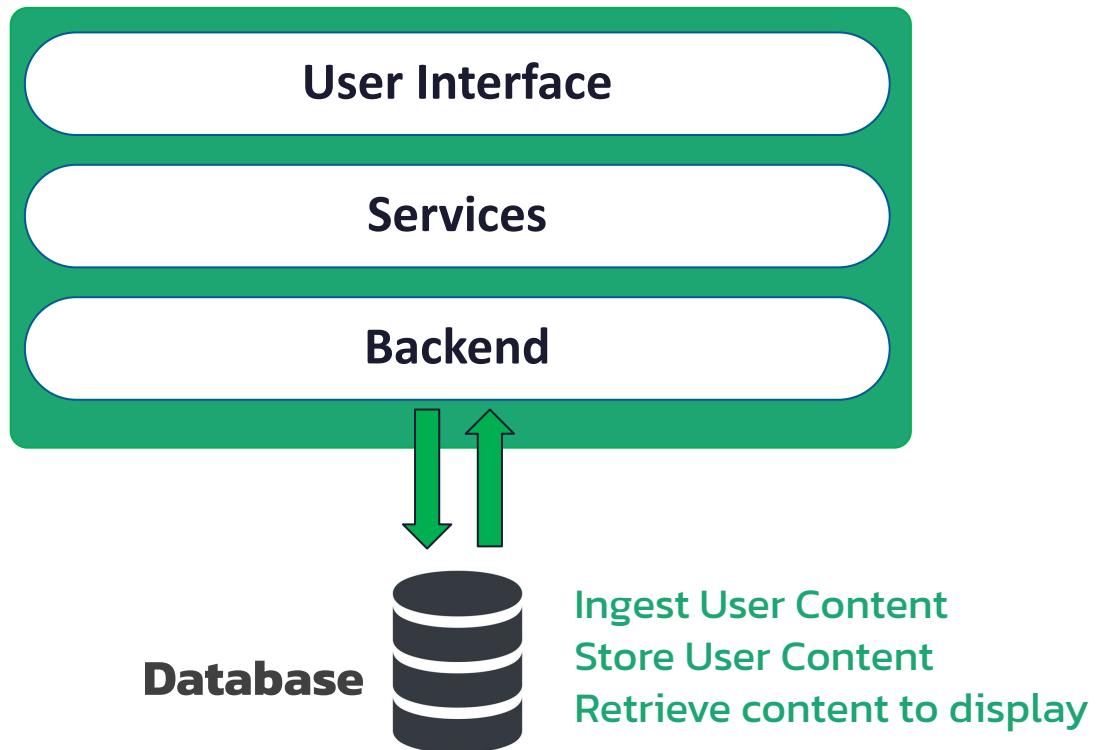


[online diagramming & design] [creately.com](#)

UML

# Petclinic :: (A) Architecture Vision

- **OLTP Create update delete operations** on multiple entities: Veterinarian, Owner, Pet, Visit with multiple **relationships**
- **Manage Reference lists** : Pet Family, Vet Specialties



# Petclinic :: (D) Technology Architecture

- Web and Mobile
- Javascript
- Angular
- Spring



SPRING BOOT



Which  
Technology ?

- Core Cassandra
- Spring Boot
- Spring-Data

Which  
Data Model  
Which Api ?



WHY  
NoSQL ?

Not  
only  
SQL

WHY  
Cassandra ?

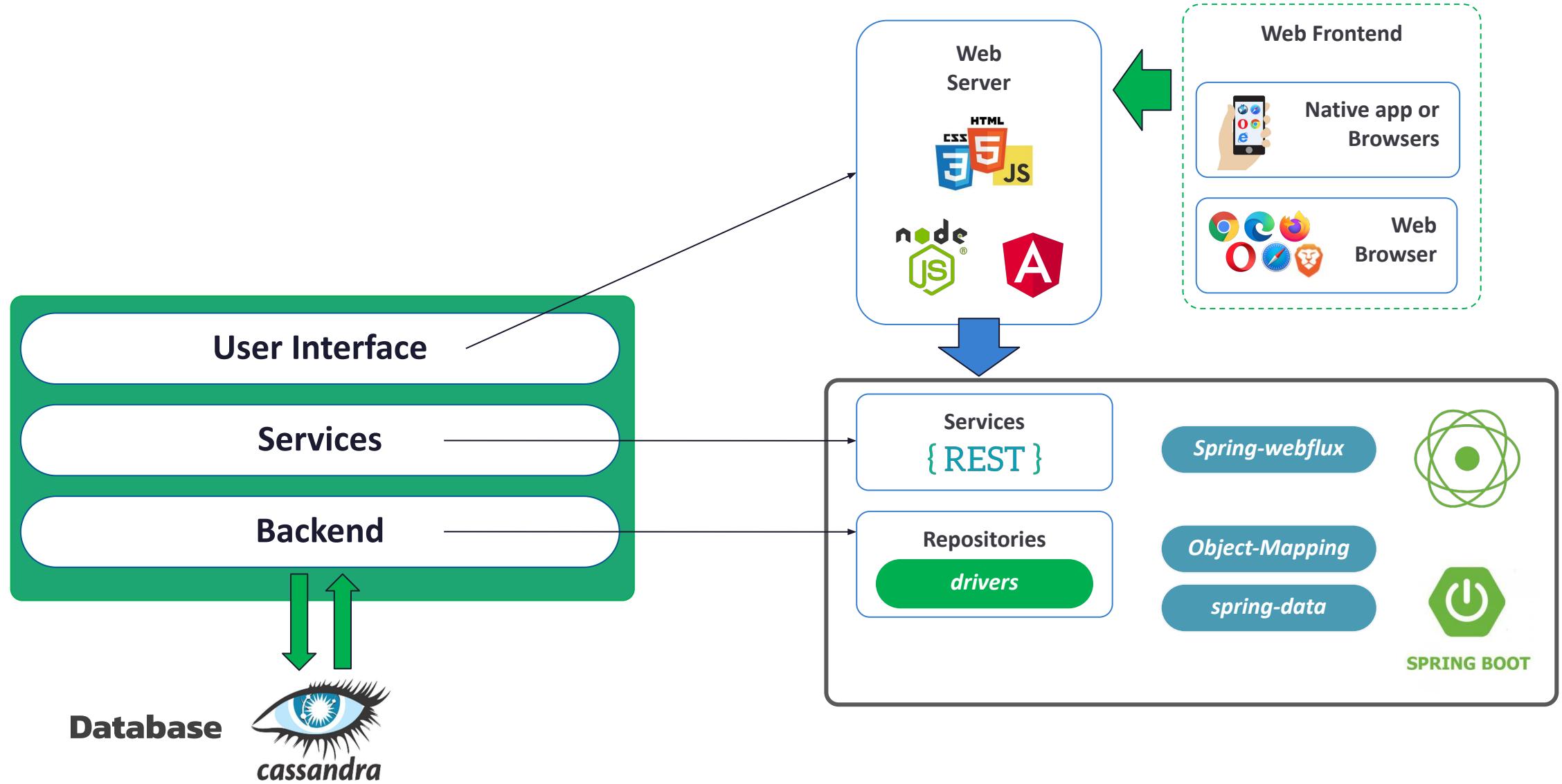
- Distributed System
- Variety of data
- High Volume
- High Throughput



cassandra

- Geographic Distribution
- Heavy Writes
- Maximum Scalability

# Petclinic :: (E) Solution and Opportunity



# Agenda

01



PetClinic  
Architecture & Use Case

02



Cassandra Database  
The Art of Data Modelling

03



Reactive Drivers  
Reactive vs Async

04



Spring Reactive  
Boot and WebFlux

05



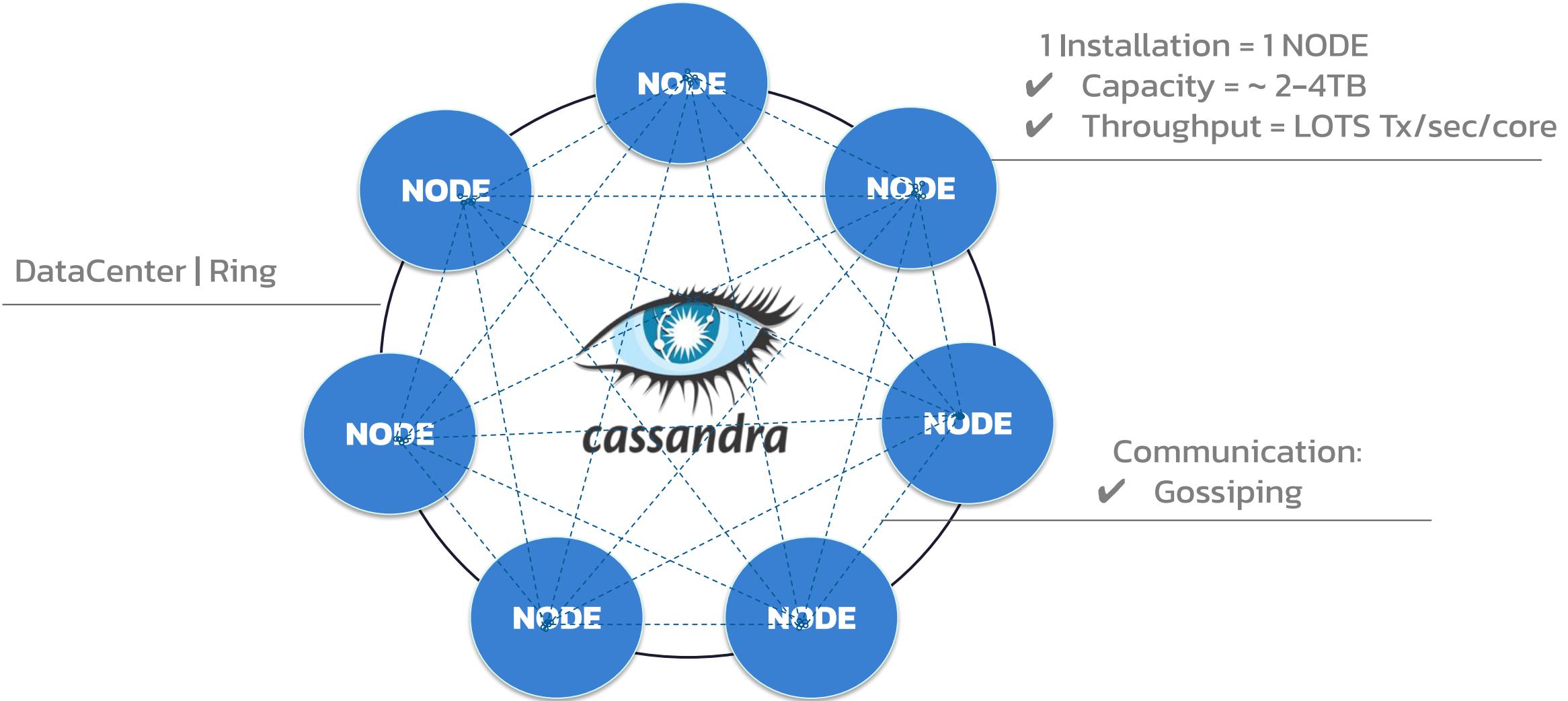
User Interface  
Angular

06

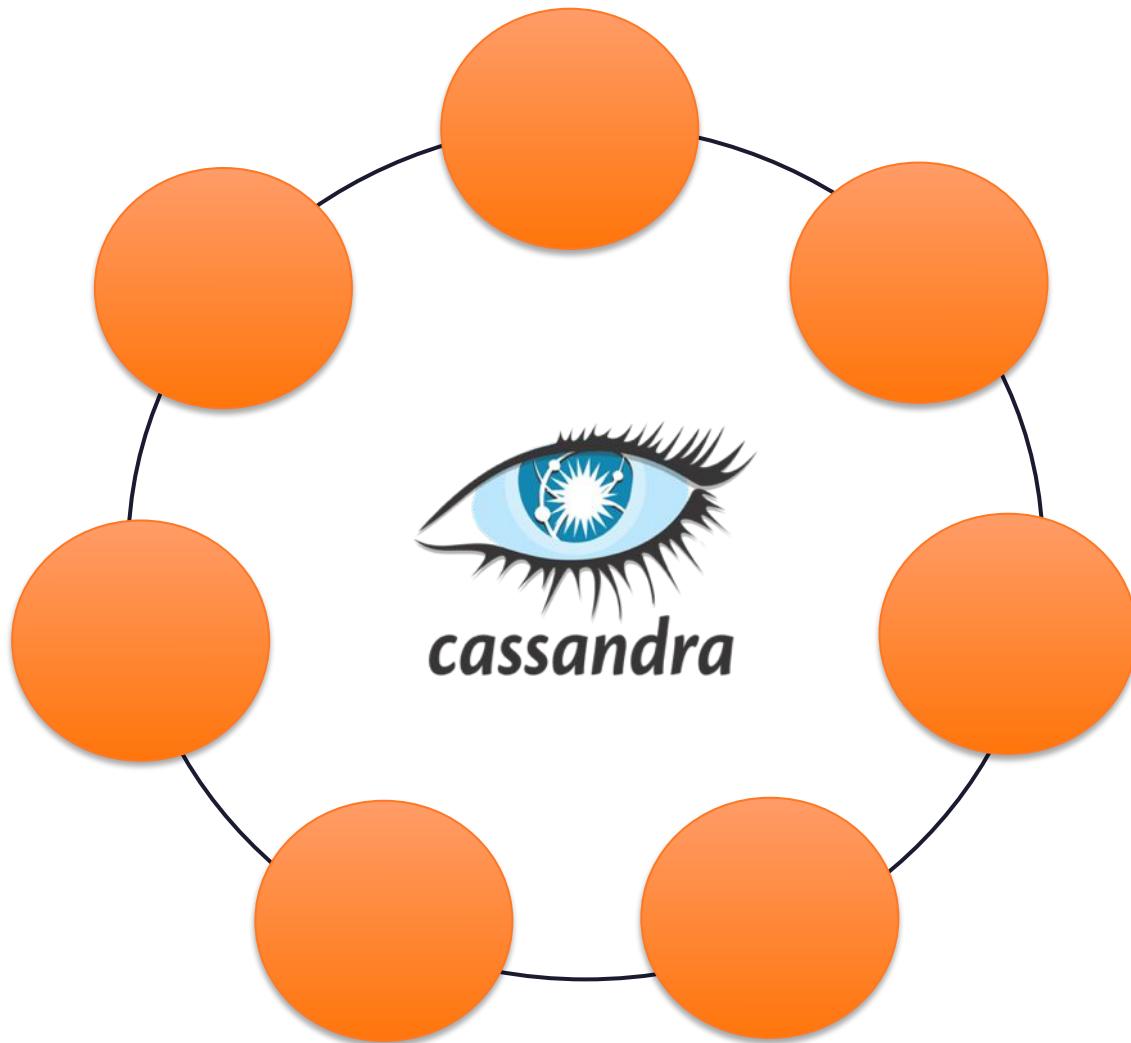


Game &  
Resources

# Apache Cassandra™ = NoSQL Distributed Database



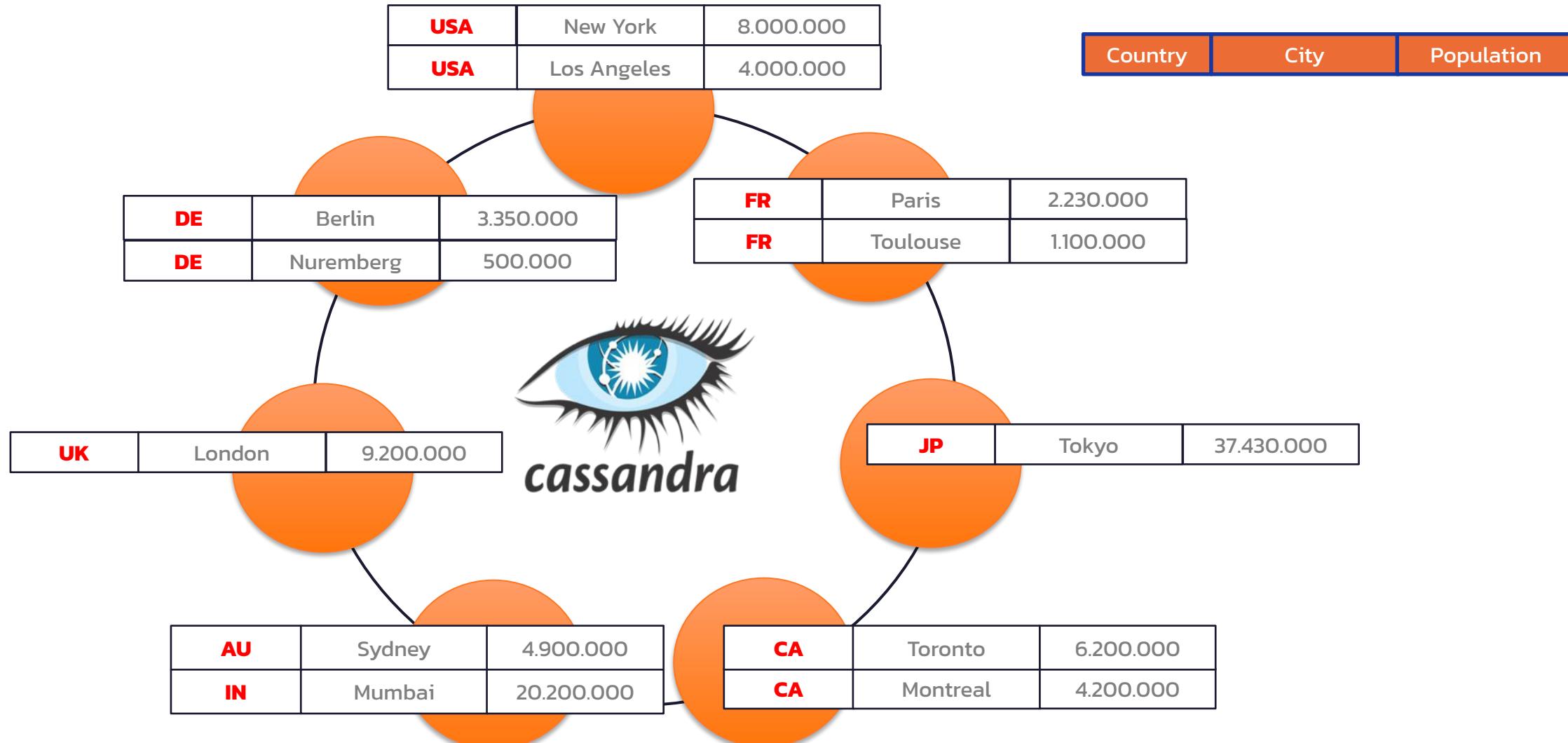
# Apache Cassandra™ :: Data is Distributed



Country	City	Population
USA	New York	8.000.000
USA	Los Angeles	4.000.000
FR	Paris	2.230.000
DE	Berlin	3.350.000
UK	London	9.200.000
AU	Sydney	4.900.000
DE	Nuremberg	500.000
CA	Toronto	6.200.000
CA	Montreal	4.200.000
FR	Toulouse	1.100.000
JP	Tokyo	37.430.000
IN	Mumbai	20.200.000

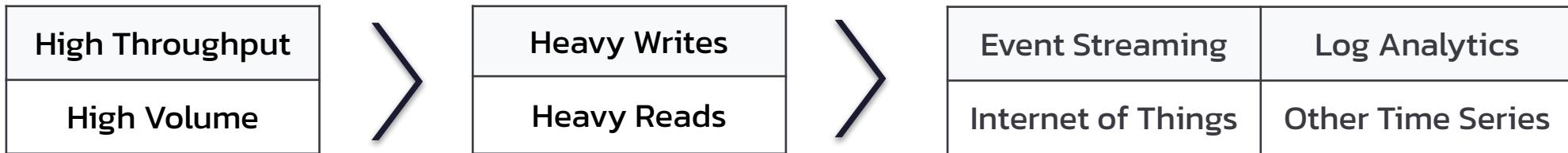
Partition Key

# Apache Cassandra™ :: Data is Distributed



# Apache Cassandra™ :: Use Cases

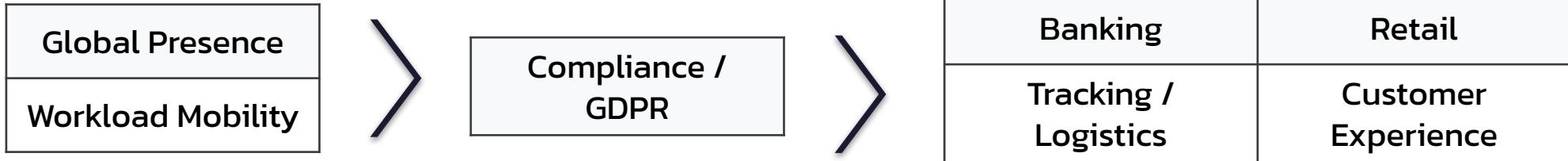
Scalability



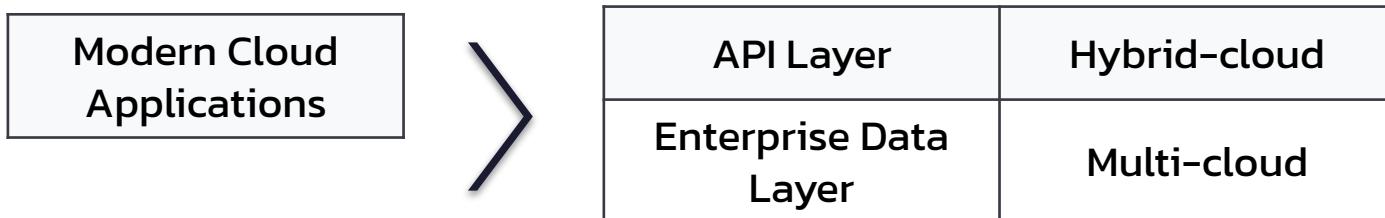
Availability



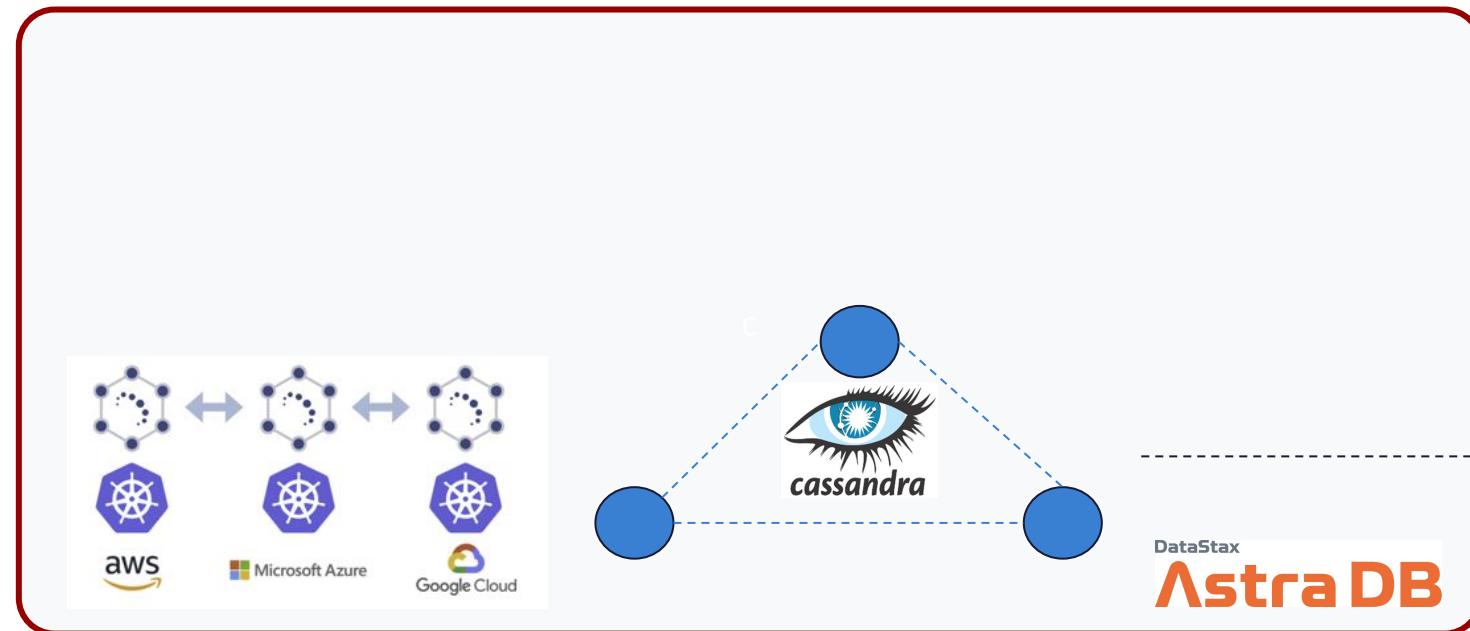
Distributed



Cloud-native



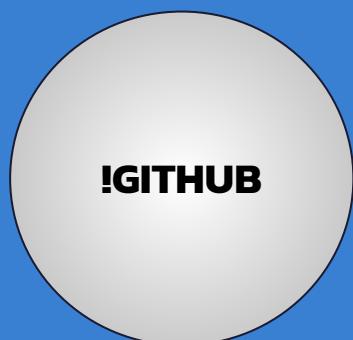
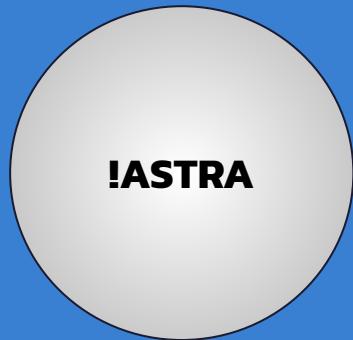
# Initiating the Database



**OSS Apache Cassandra**  
A Column oriented NoSQL  
Database



# HandsOn: #4. Create Instance



# Astra DB

**Get your instance here:**

<https://astra.dev/11-22>



**GitHub**

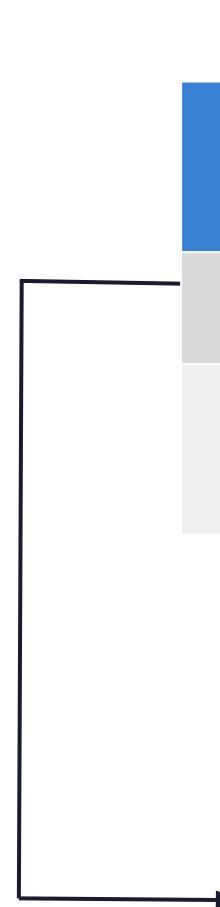
**Repository:**

[github.com/datastaxdevs/workshop-spring-reactive](https://github.com/datastaxdevs/workshop-spring-reactive)



# Normalization

"Database normalization is the process of structuring a relational database in accordance with a series of so-called normal forms in order to reduce data redundancy and improve data integrity. It was first proposed by Edgar F. Codd as part of his relational model."



The diagram illustrates a relationship between two tables: Employees and Departments. A vertical line connects the 'deptId' column of the Employees table to the 'departmentId' column of the Departments table, indicating a foreign key relationship.

Employees			
userId	deptId	firstName	lastName
1	1	Edgar	Codd
2	1	Raymond	Boyce

Departments	
departmentId	department
1	Engineering
2	Math

**PROS:** Simple write, Data Integrity

**CONS:** Slow read, Complex Queries

# Denormalization

"Denormalization is a strategy used on a database to increase performance. In computing, denormalization is the process of trying to improve the read performance of a database, at the expense of losing some write performance, by adding redundant copies of data"

**PROS:** Quick Read, Simple Queries

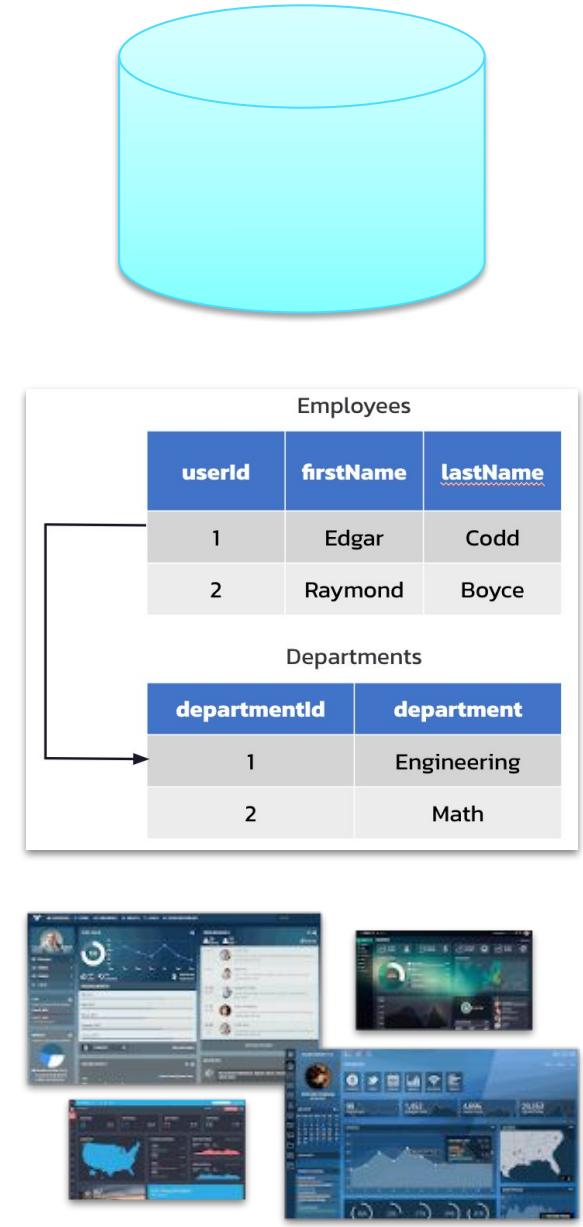
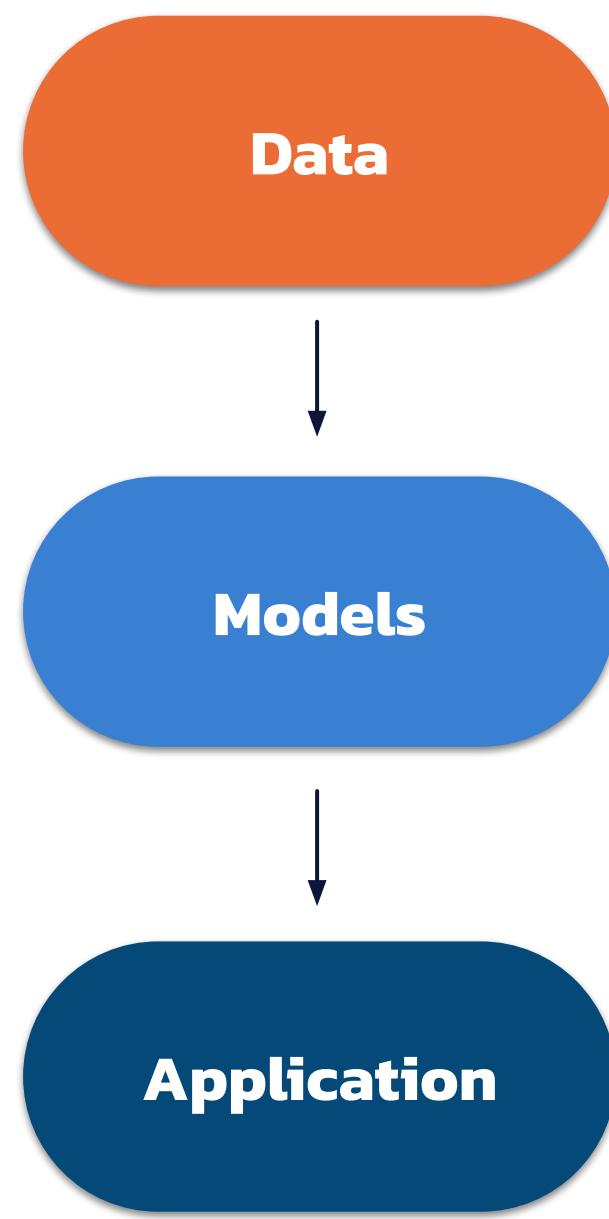
**CONS:** Multiple Writes, Manual Integrity

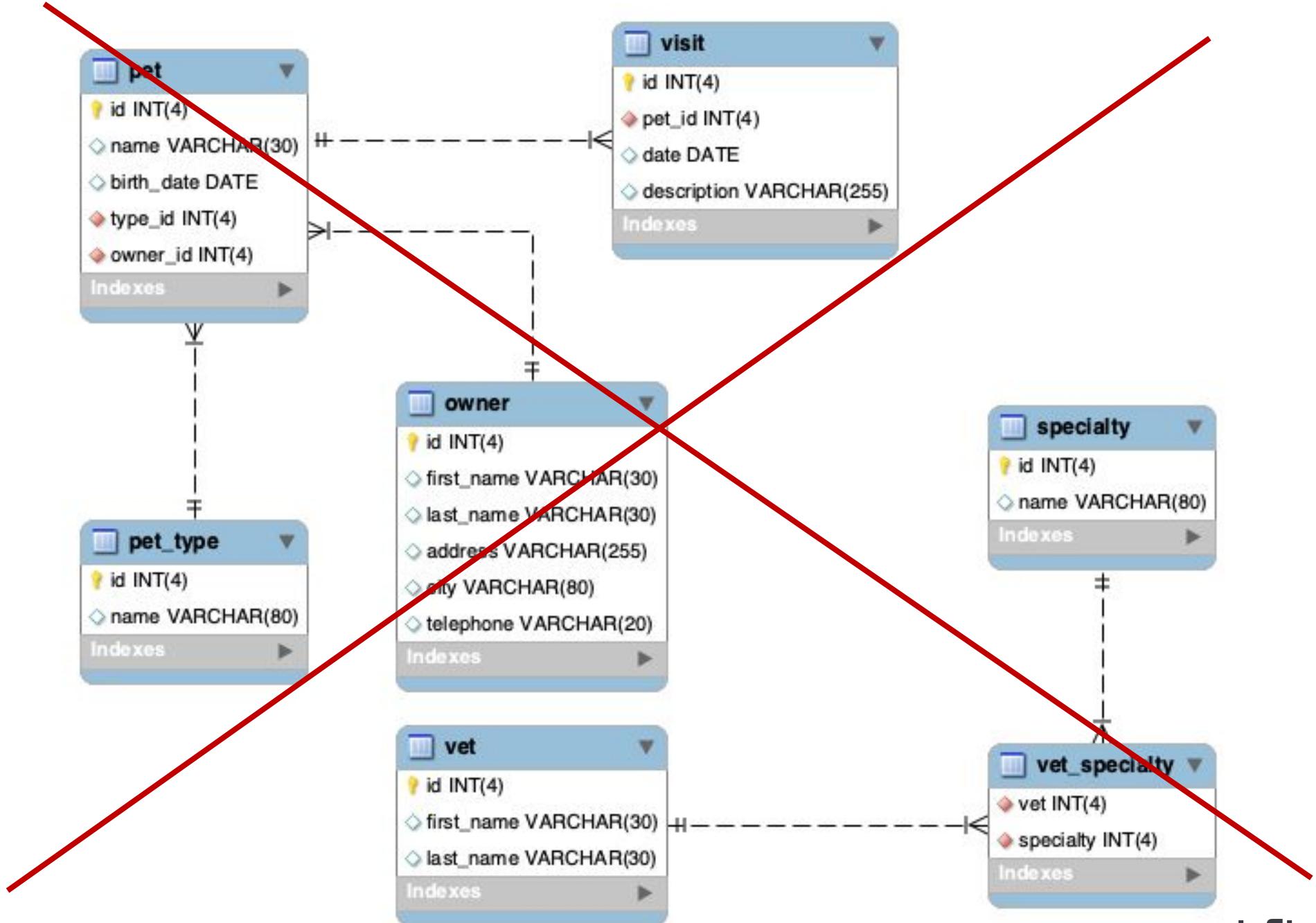
Employees

userId	firstName	lastName	department
1	Edgar	Codd	Engineering
2	Raymond	Boyce	Engineering
3	Sage	Lahja	Math
4	Juniper	Jones	Botany

# Relational Data Model

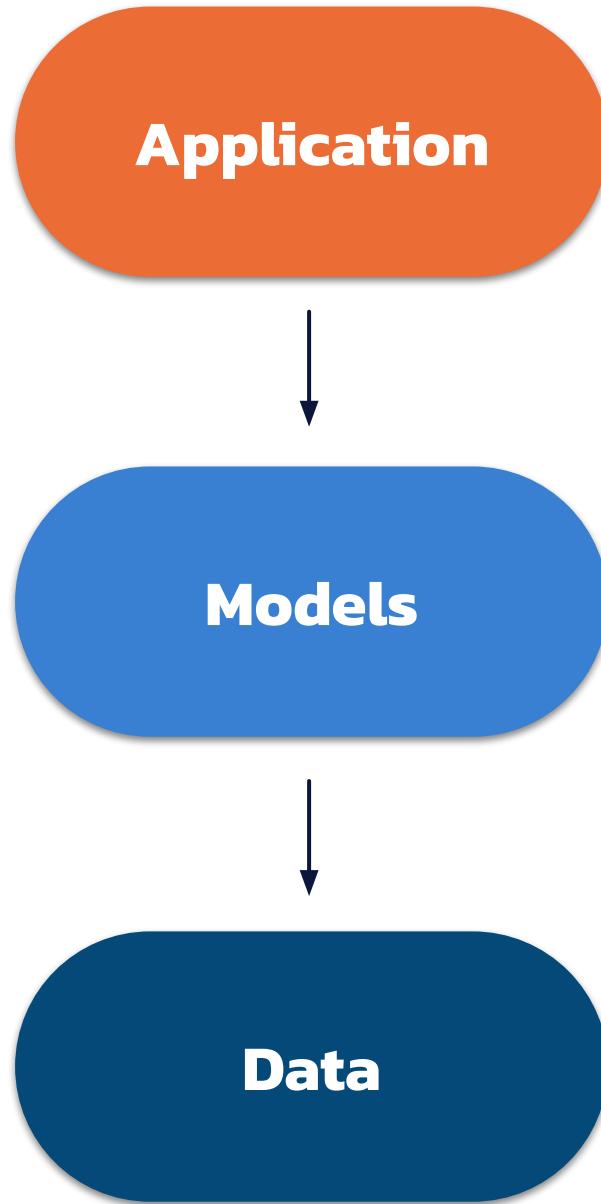
1. Analyze raw data
2. Identify entities, their properties and relations
3. Design tables, using **normalization** and foreign keys.
4. Use JOIN when doing queries to join normalized data from multiple tables



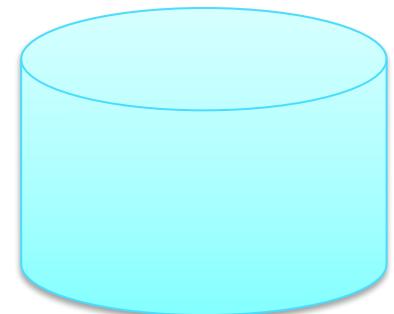


# NoSQL Data Model

1. Analyze user behaviour  
(customer first!)
2. Identify workflows, their dependencies and needs
3. Define Queries to fulfill these workflows
4. Knowing the queries, design tables, using **denormalization**.
5. Insert and update multiple copies of data that may have resulted due to denormalization

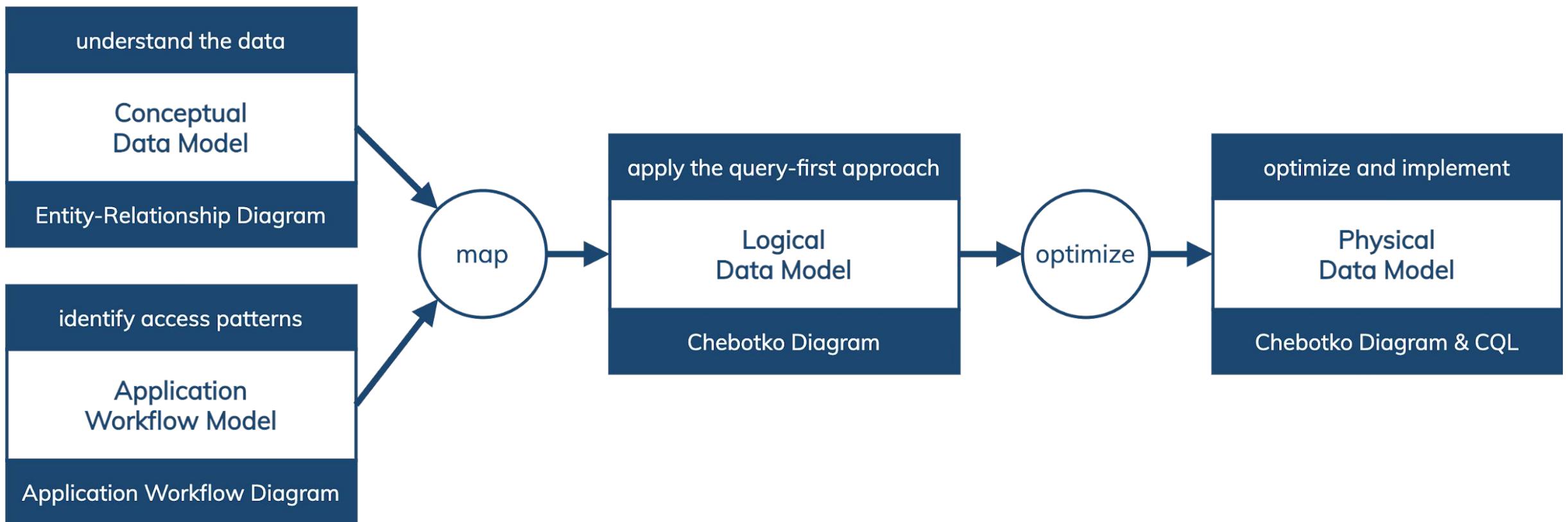


Employees			
userId	firstName	lastName	department
1	Edgar	Codd	Engineering
2	Raymond	Boyce	Math
3	Sage	Lahja	Math
4	Juniper	Jones	Botany

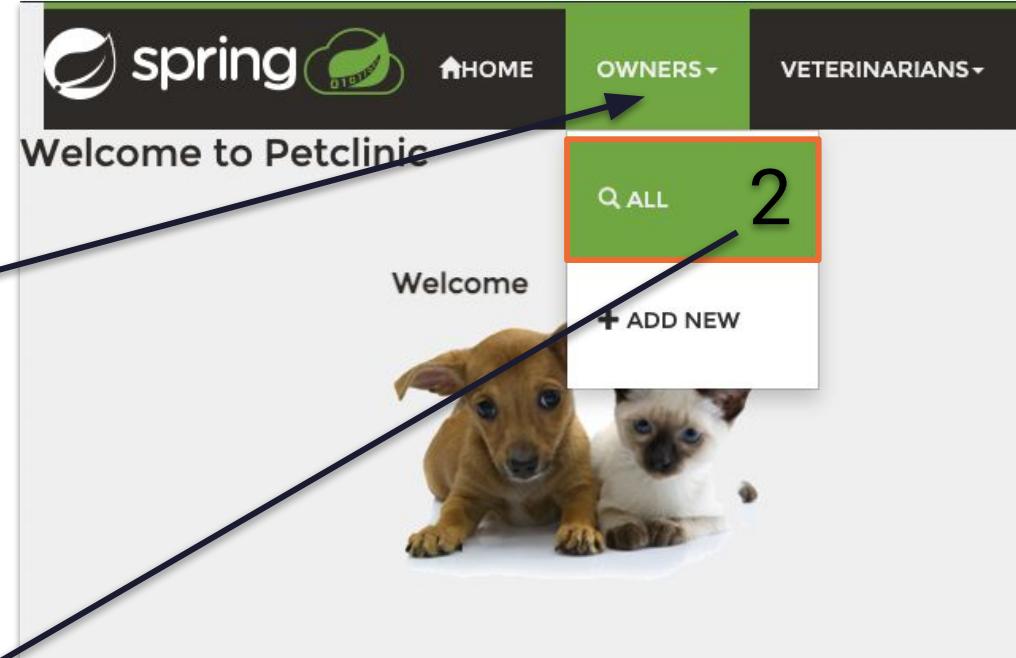


# Cassandra Data Modeling Methodology

**Four objectives, 4 models, 2 transitions**



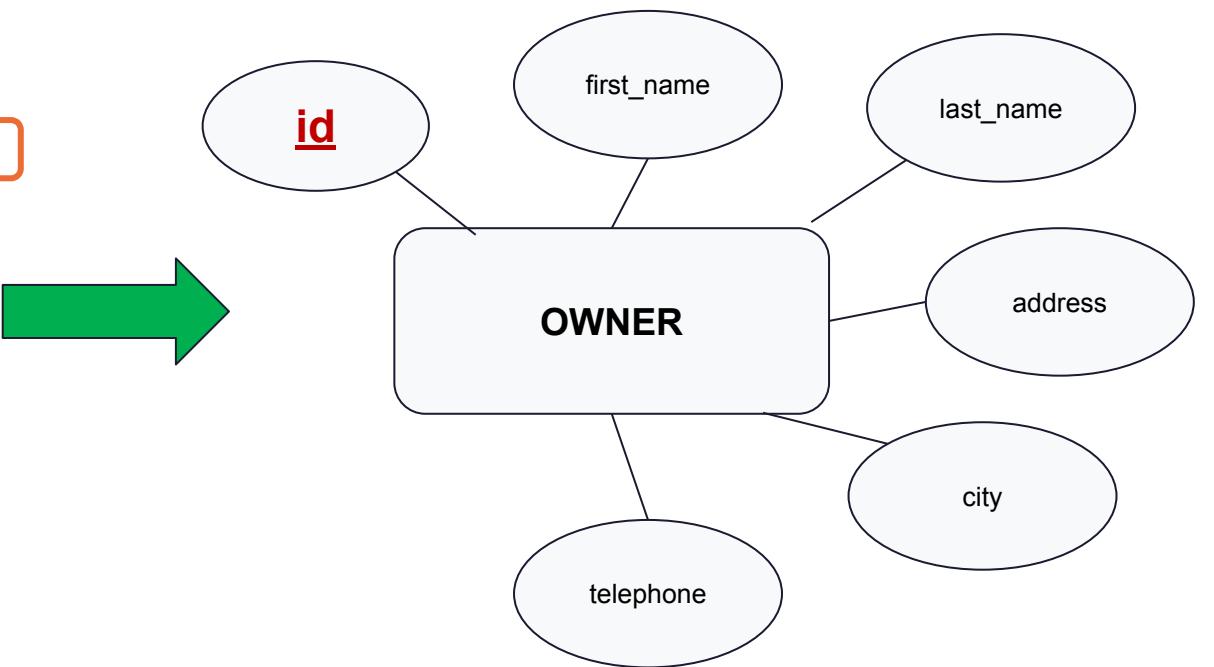
# #1 – List Owners



Owners				
Name	Address	City	Telephone	Pets
Heather Jones-Gilardi	AnotherPlace Dr.	Orlando	987654321	
David Gilardi	Nowhere St.	Orlando	123456789	Carrabba Shocker
Add Owner				

# #1 – List Owners

owner	
!	<b>id INT(4)</b>
◊	first_name VARCHAR(30)
◊	last_name VARCHAR(30)
◊	address VARCHAR(255)
◊	city VARCHAR(80)
◊	telephone VARCHAR(20)
Indexes	



petclinic_owner		
<b>id</b>	<b>uuid</b>	K
address	text	
city	text	
first_name	text	
last_name	text	
telephone	text	

## Use-Case:

- A User opens the owner “ALL” page

## Workflow:

- List all owners

## #2 – Get Pets of Owner (1 to many)

### Owners

Name	Address	City	Telephone	Pets
Heather Jones-Gilardi	AnotherPlace Dr.	Orlando	987654321	
David Gilardi	Nowhere St.	Orlando	123456789	Carrabba Shocker

Add Owner



# #2 – Get Pets of Owner (1 to many)

## Owner Information

Name David Gilardi  
Address Nowhere St.  
City Orlando  
Telephone 123456789

< Back    Edit Owner    Add New Pet

## Pets and Visits

Name Shocker  
Birth Date 2021/03/29  
Type dog

Edit Pet    Delete Pet    Add Visit

Visit Date	Description	Actions	
2019/03/25	yearly visit	Edit Visit	Delete Visit
2019/03/25	yearly visit	Edit Visit	Delete Visit

Name Carrabba  
Birth Date 2014/05/23  
Type dog

Edit Pet    Delete Pet    Add Visit

Visit Date	Description	Actions	
2020/03/25	yearly visit	Edit Visit	Delete Visit
2019/03/25	yearly visit	Edit Visit	Delete Visit

## #2 – Get Pets of Owner (1 to many)

### Owner Information

Name  
Address  
City  
Telephone

David Gilardi  
Nowhere St.  
Orlando  
123456789

< Back   Edit Owner   Add New Pet

One owner

### Pets and Visits

Name Shocker  
Birth Date 2021/03/29  
Type dog

Edit Pet   Delete Pet   Add Visit

Name Carrabba  
Birth Date 2014/05/23  
Type dog

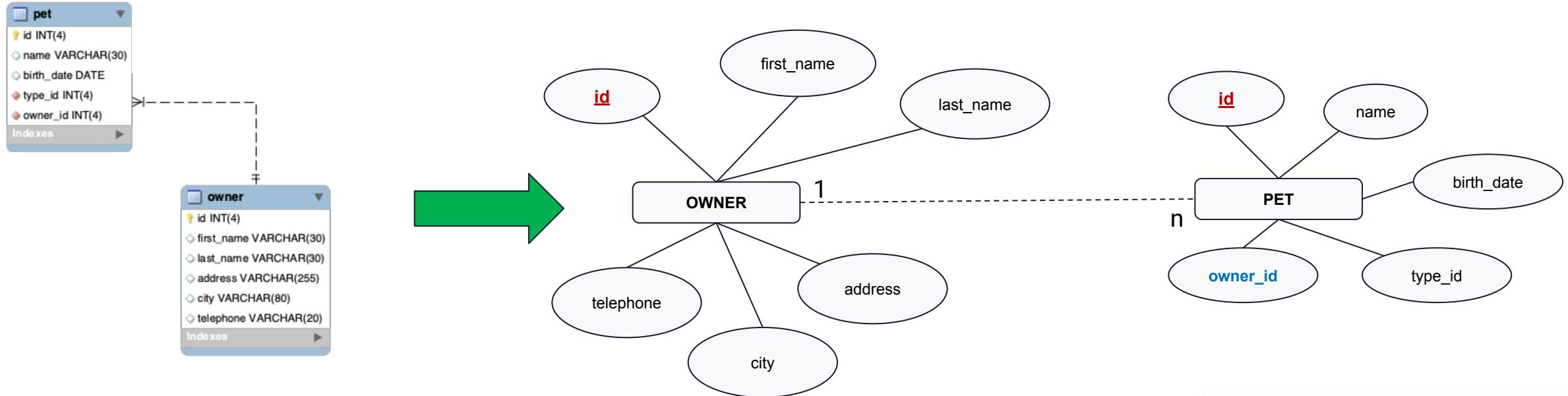
Edit Pet   Delete Pet   Add Visit

Visit Date	Description	Actions	
2019/03/25	yearly visit	Edit Visit	Delete Visit
2019/03/25	yearly visit	Edit Visit	Delete Visit

Visit Date	Description	Actions	
2020/03/25	yearly visit	Edit Visit	Delete Visit
2019/03/25	yearly visit	Edit Visit	Delete Visit

Multiple pets

## #2 – Get Pets of Owner (1 to many)



### Use-Case:

- A User opens the owner detail page

### Workflow:

- Find owner and any pets related to target owner using its identifier (owner\_id)

petclinic_pet_by_owner		
owner_id	uuid	K
pet_id	uuid	C↑
birth_date	date	
name	text	
pet_type	text	

### petclinic\_visit\_by\_pet

pet_id	uuid	K
visit_id	uuid	C↑
visit_date	date	
description	text	
last_name	text	
telephone	text	

## #3 – Get Visits of Pets (1 to many)

### Owner Information

Name  
Address  
City  
Telephone

< Back   Edit Owner   Add New Pet

David Gilardi  
Nowhere St.  
Orlando  
123456789

One owner

### Pets and Visits

Name Shocker  
Birth Date 2021/03/29  
Type dog

Edit Pet   Delete Pet   Add Visit

Name Carrabba  
Birth Date 2014/05/23  
Type dog

Edit Pet   Delete Pet   Add Visit

Multiple visits per each pet

#### Visit Date   Description   Actions

2019/03/25   yearly visit  
2019/03/25   yearly visit

Edit Visit   Delete Visit

Edit Visit   Delete Visit

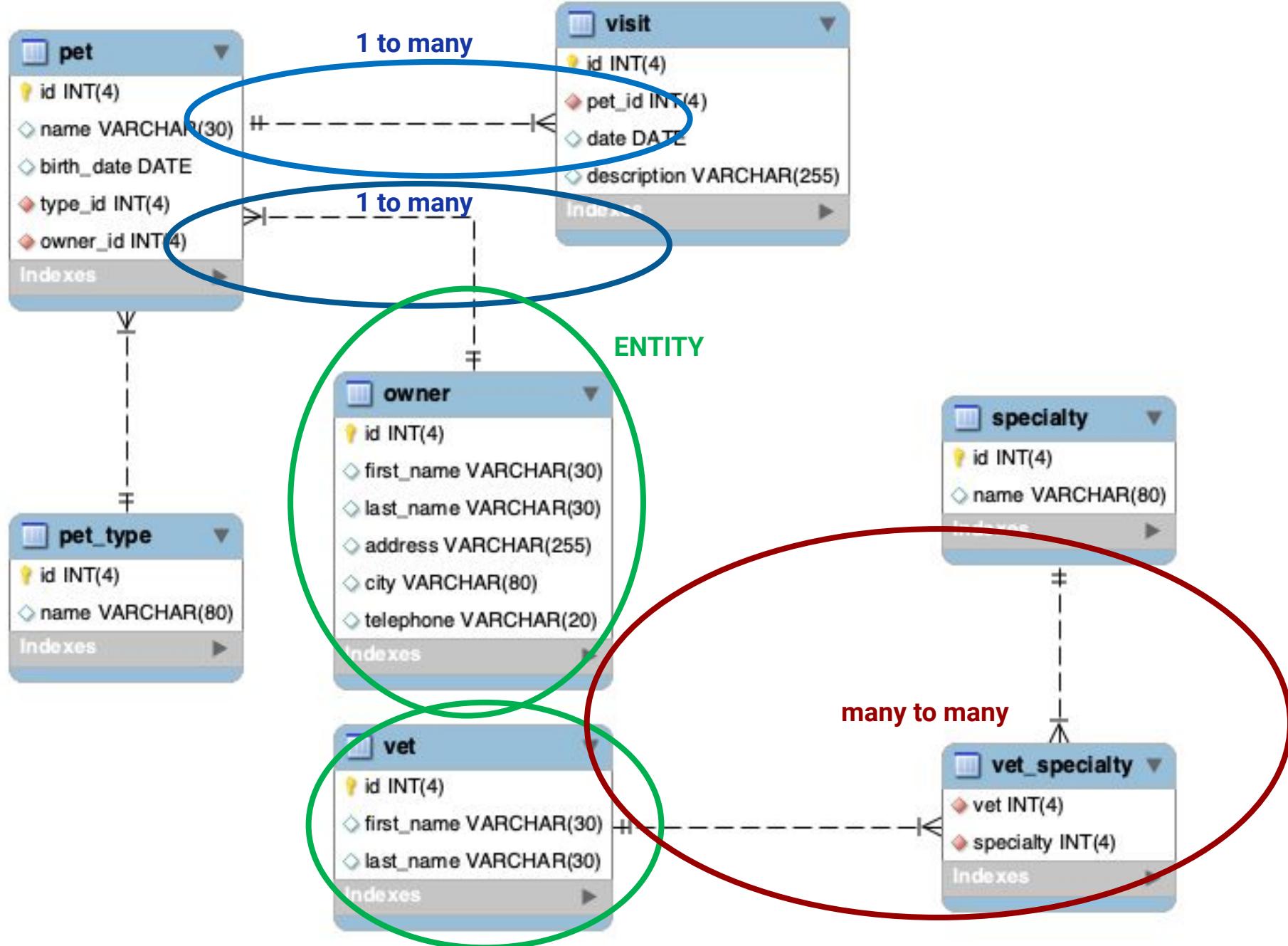
#### Visit Date   Description   Actions

2020/03/25   yearly visit  
2019/03/25   yearly visit

Edit Visit   Delete Visit

Edit Visit   Delete Visit

Multiple pets



# Complete Model

petclinic_owner		
<b>id</b>	uuid	K
address	text	
city	text	
first_name	text	
last_name	text	
telephone	text	

petclinic_pet_by_owner		
<b>owner_id</b>	uuid	K
<b>pet_id</b>	uuid	C↑
birth_date	date	
name	text	
pet_type	text	

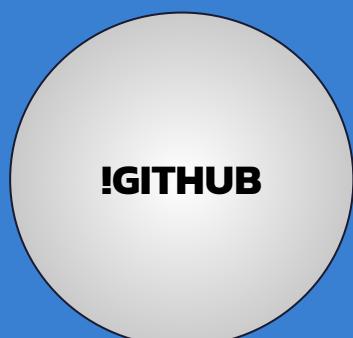
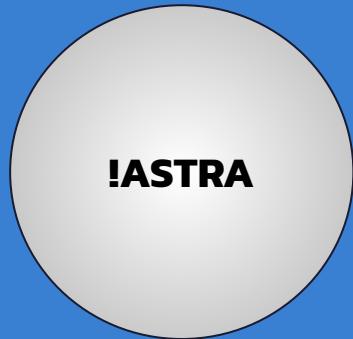
petclinic_visit_by_pet		
<b>pet_id</b>	uuid	K
<b>visit_id</b>	uuid	C↑
visit_date	date	
description	text	
last_name	text	
telephone	text	

petclinic_vet		
<b>id</b>	uuid	K
first_name	text	
last_name	text	
specialties	set<text>	

petclinic_vet_by_specialty		
<b>specialty</b>	text	K
<b>vet_id</b>	uuid	C↑
first_name	text	
last_name	text	

petclinic_reference_lists		
<b>list_name</b>	text	K
values	set<text>	

# HandsOn: 5. Create your schema



# Astra DB

**Get your instance here:**

<https://astra.dev/11-22>



**GitHub**

**Repository:**

[github.com/datastaxdevs/workshop-spring-reactive](https://github.com/datastaxdevs/workshop-spring-reactive)



# Agenda

01



PetClinic  
Architecture & Use Case

02



Cassandra Database  
The Art of Data Modelling

03



Reactive Drivers  
Reactive vs Async

04



Spring Reactive  
Boot and WebFlux

05



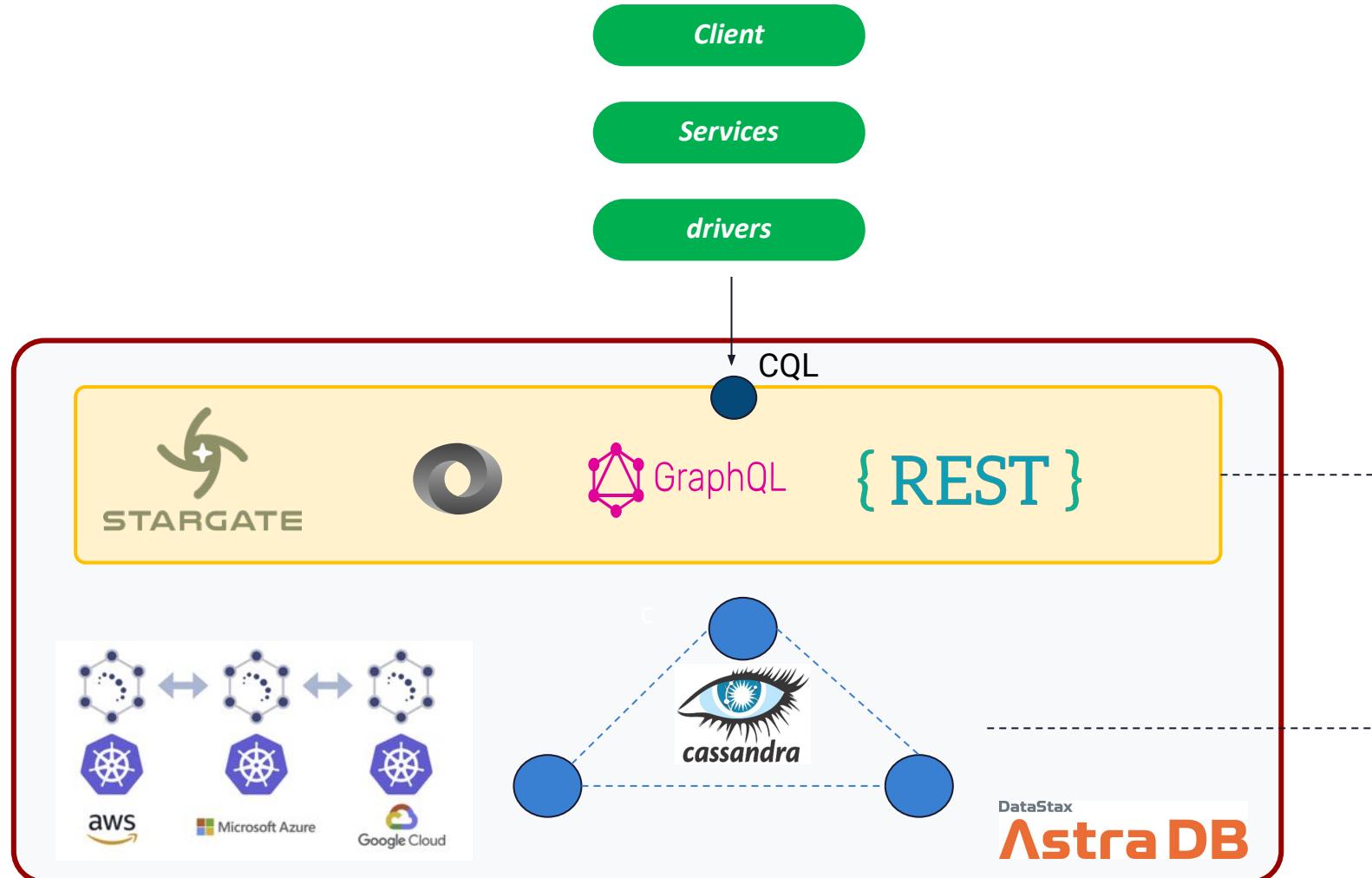
User Interface  
Angular

06



Game &  
Resources

# Accessing the Database



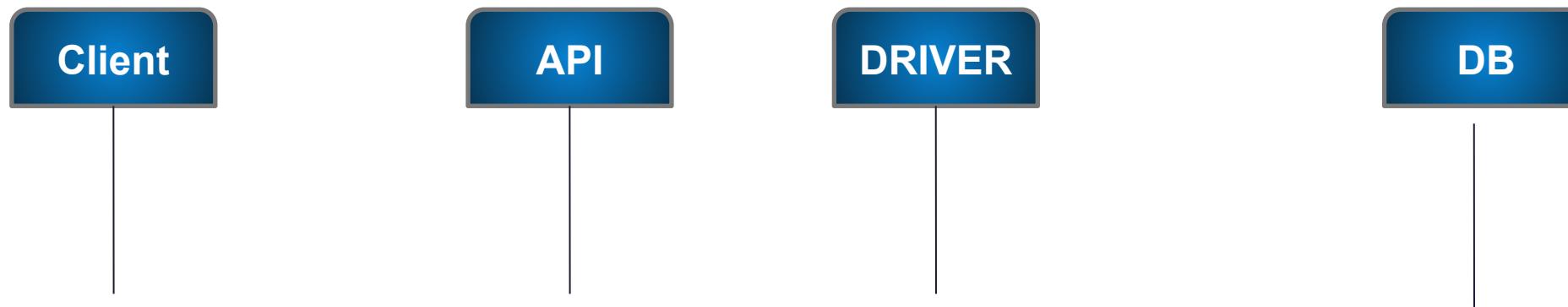
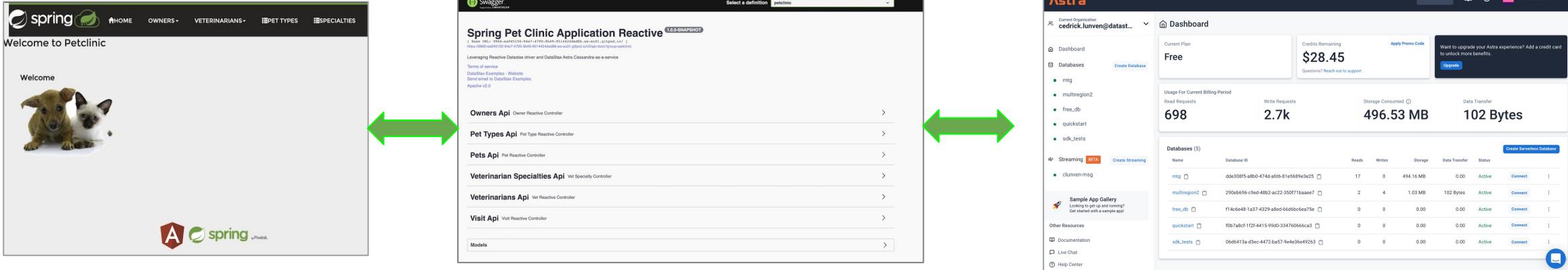
**OSS Stargate.io**  
A data gateway to allow  
multiple usages



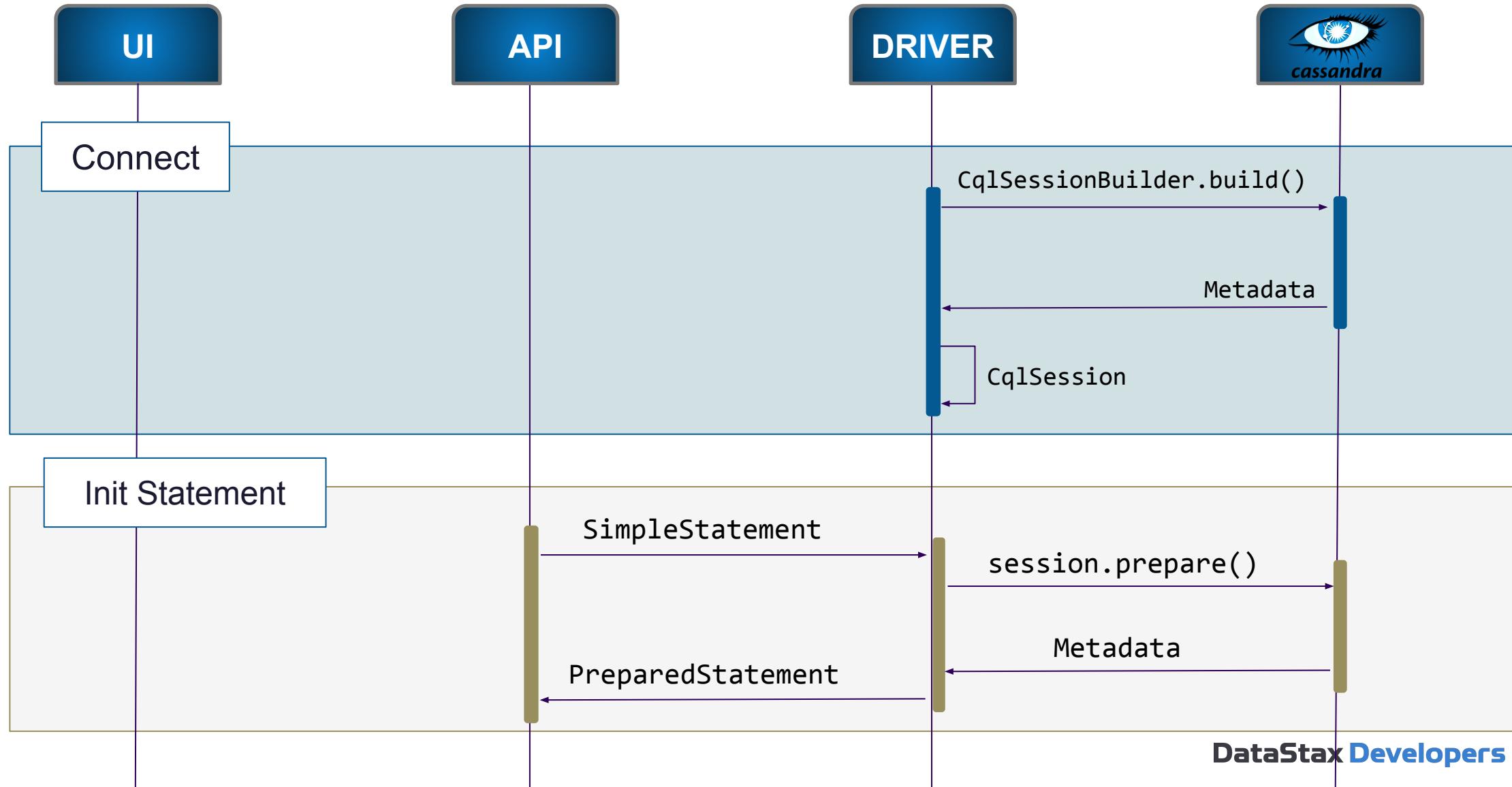
**OSS Apache Cassandra**  
A Column oriented NoSQL  
Database

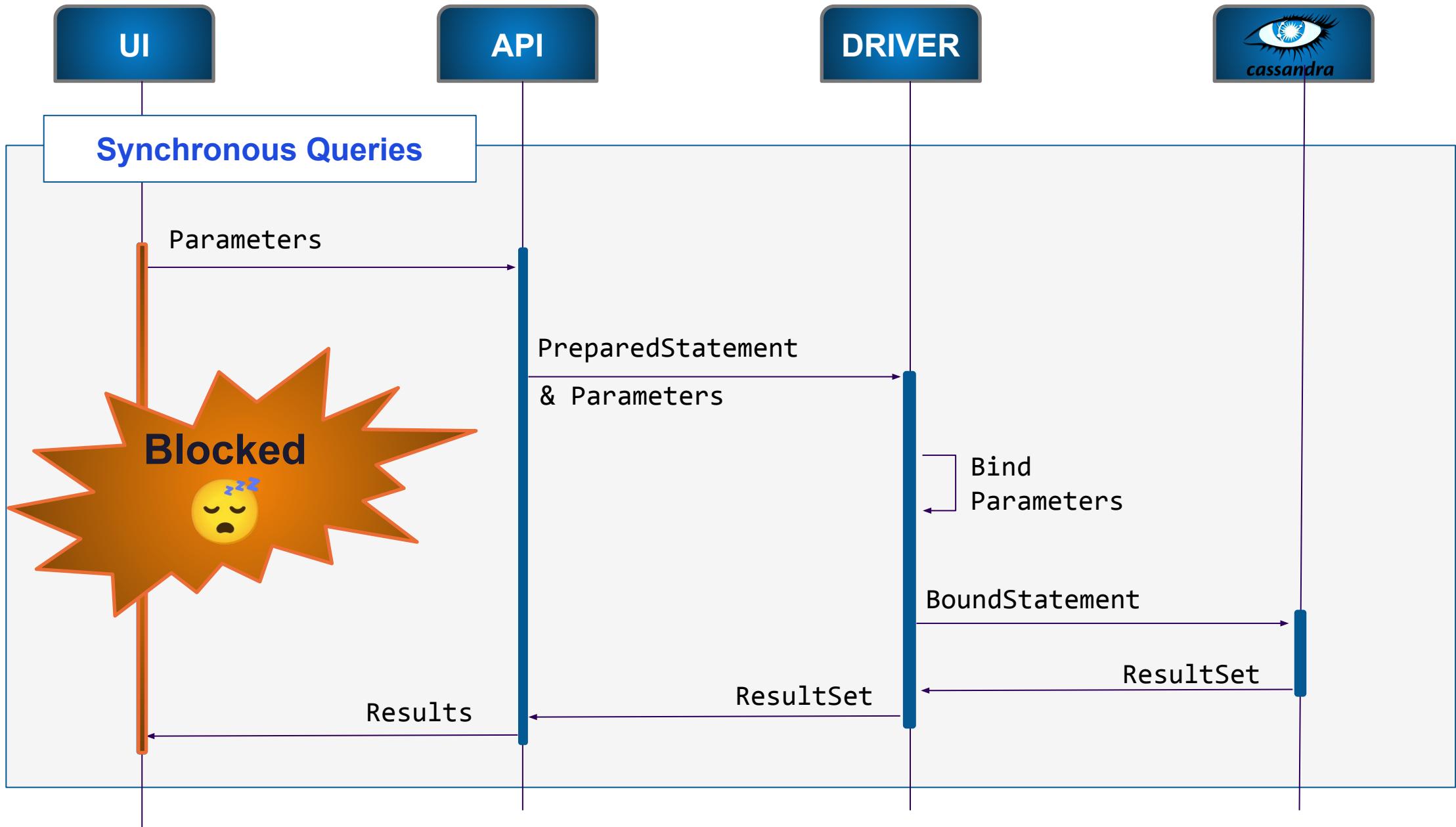


# Sequence Diagram (1)



# Initializations





# Synchronous

## Strengths

Simple

## Weaknesses

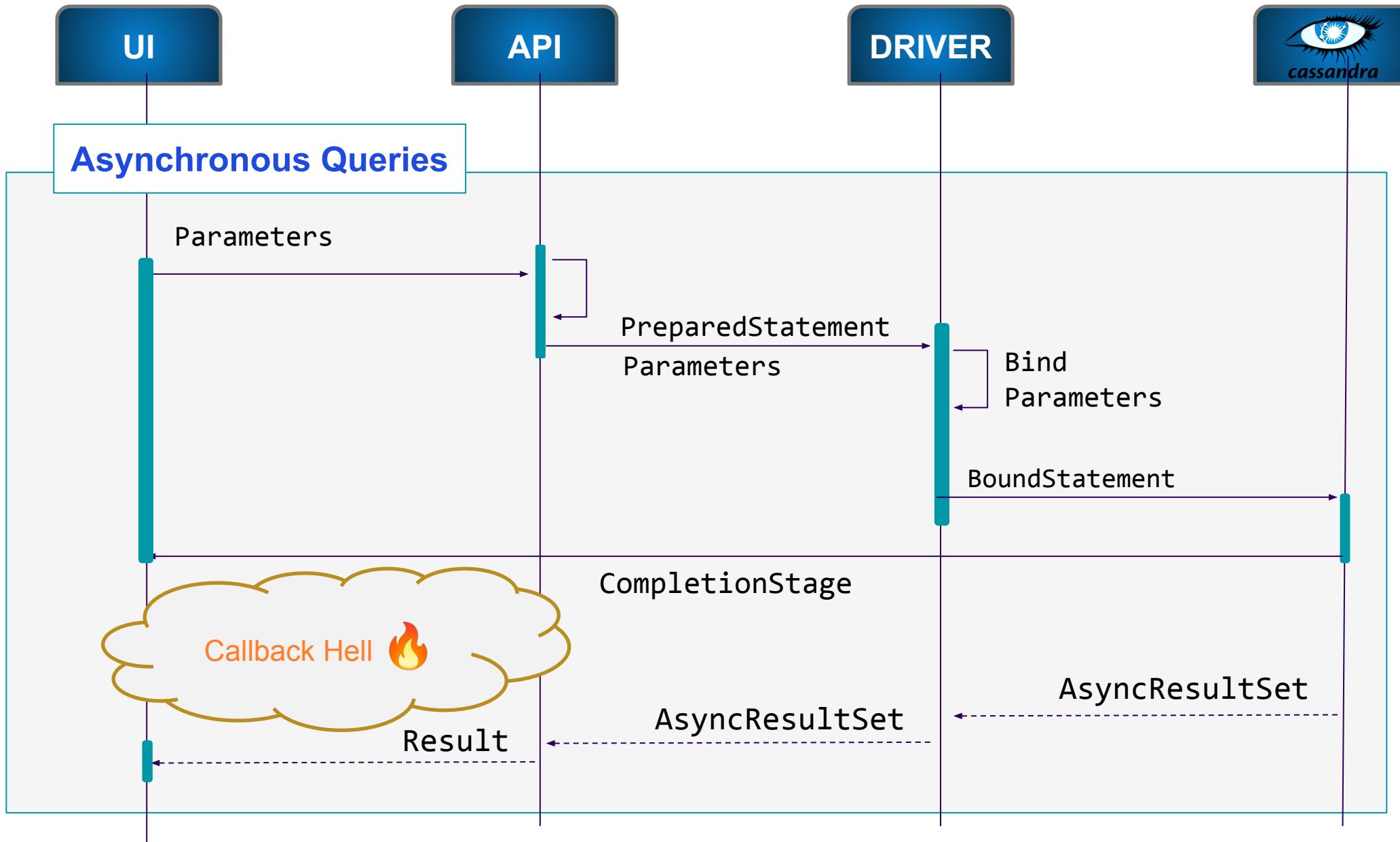
Blocking 😴

## Opportunities

Easy to Test & Maintain

## Threats

Scalability !



# Asynchronous

## Strengths

Non Blocking

## Weaknesses

Callback Hell 

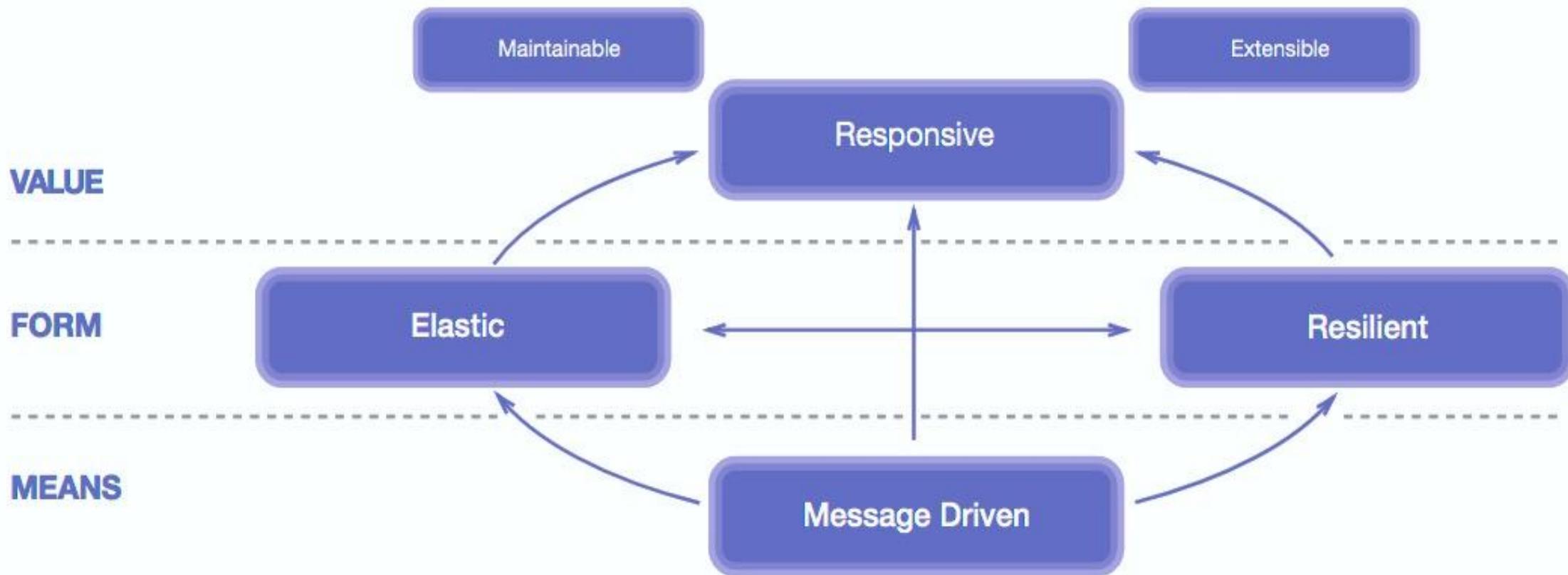
## Opportunities

Scalability

## Threats

Maintainability 

# Reactive Manifesto

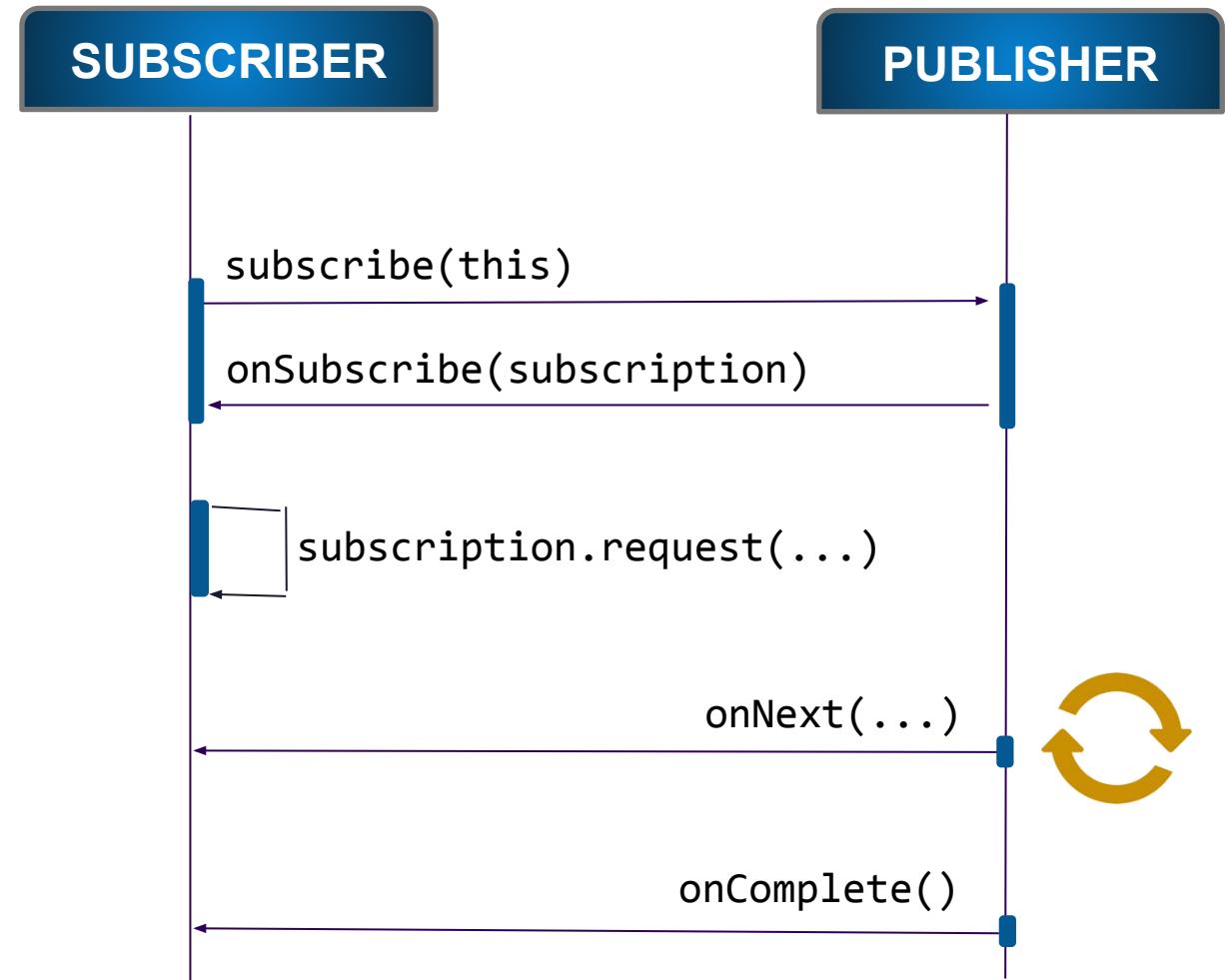


# Reactive Stream Api

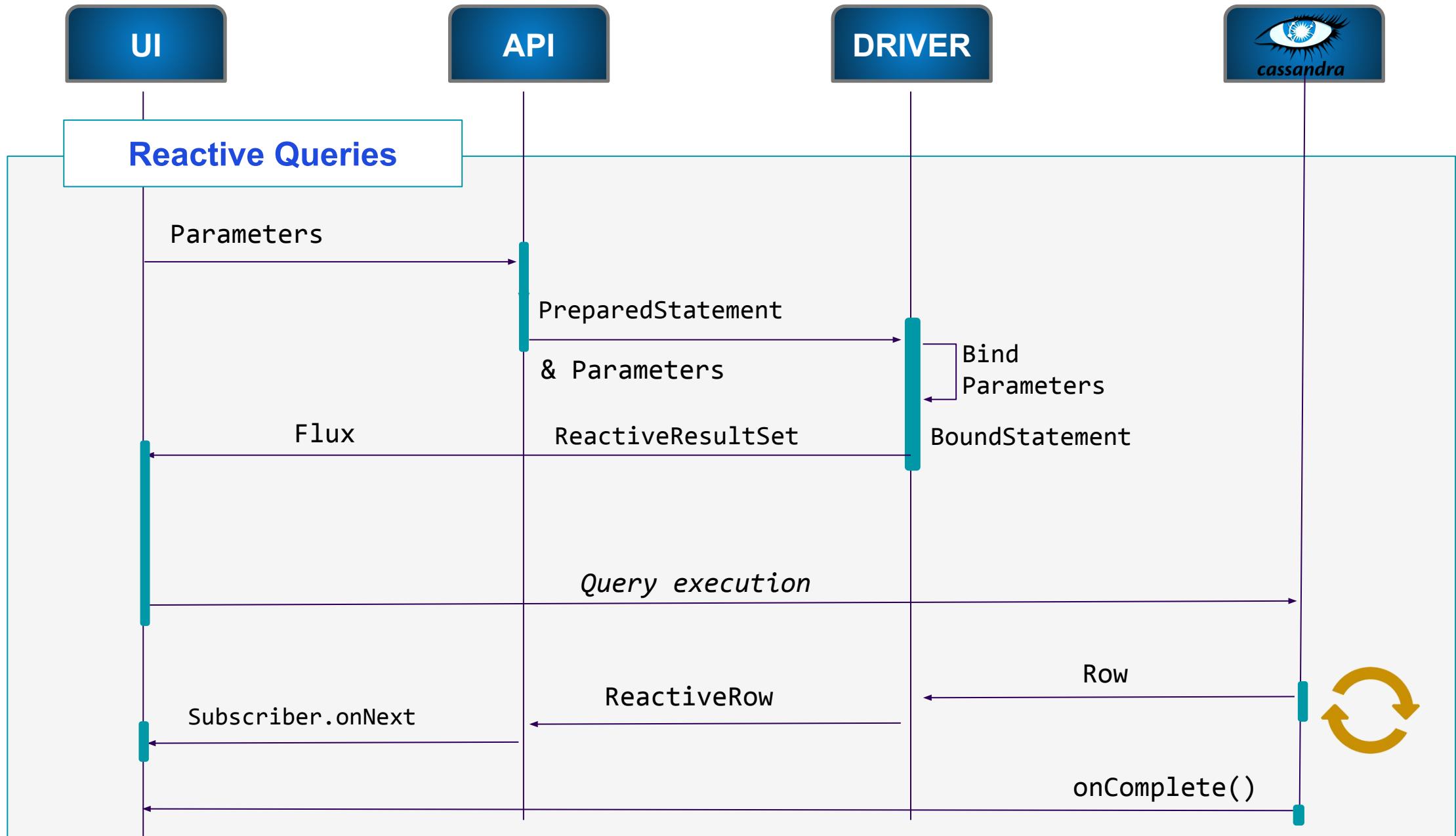


VERT.X

akka



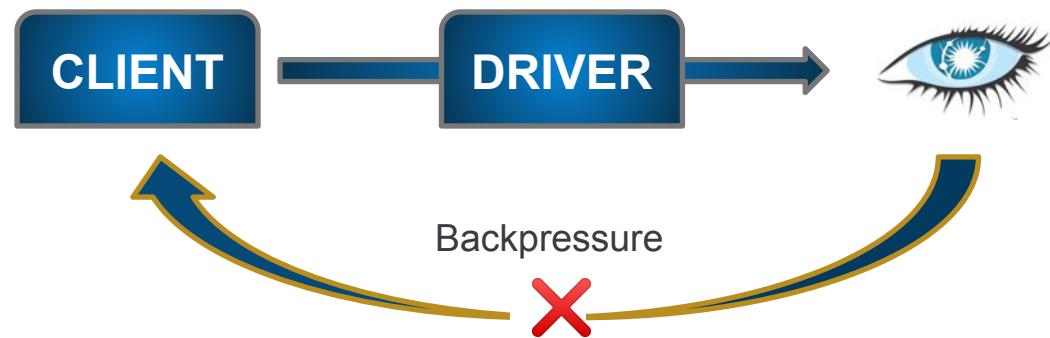
<http://www.reactive-streams.org/>



# Back Pressure

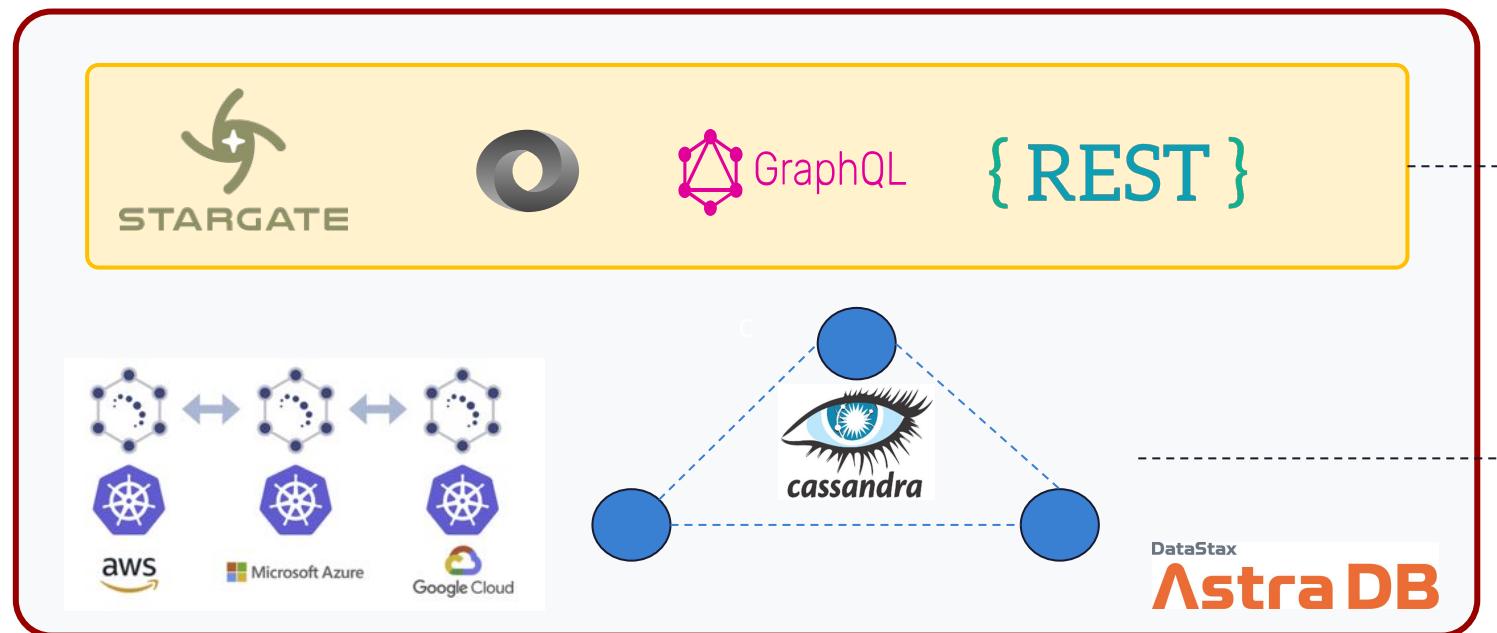


READ



WRITE

# Astra Connectivity



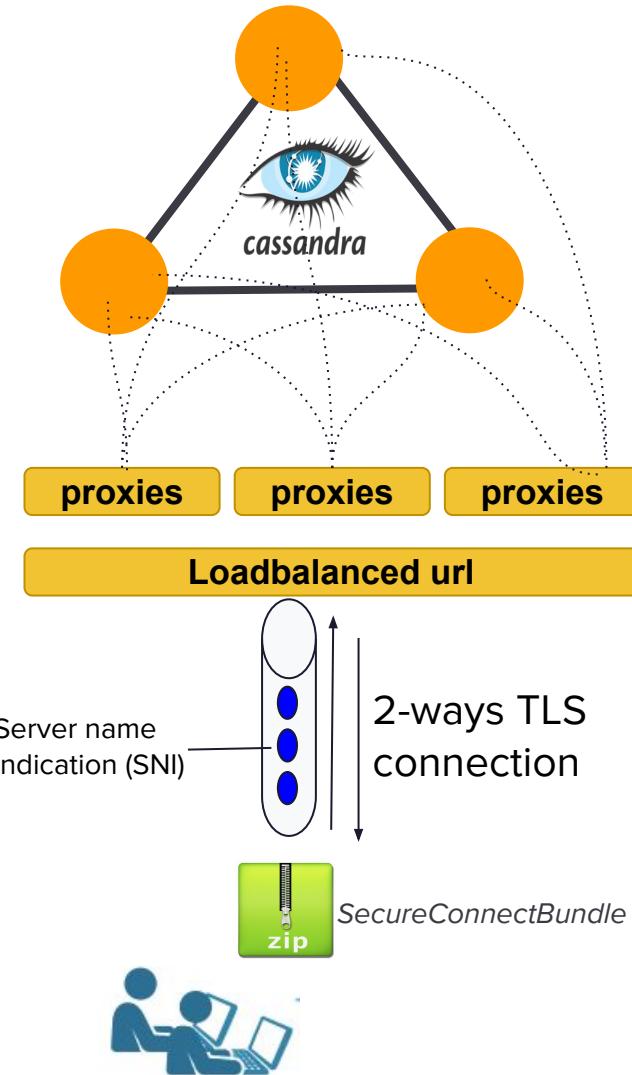
**OSS Stargate.io**  
A data gateway to allow  
multiple usages



**OSS Apache Cassandra**  
A Column oriented NoSQL  
Database



# Astra Connectivity

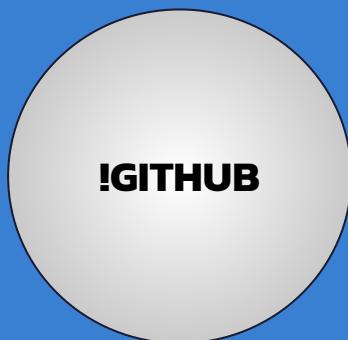


# HandsOn: 6, 7 and 8

6. Create your Token

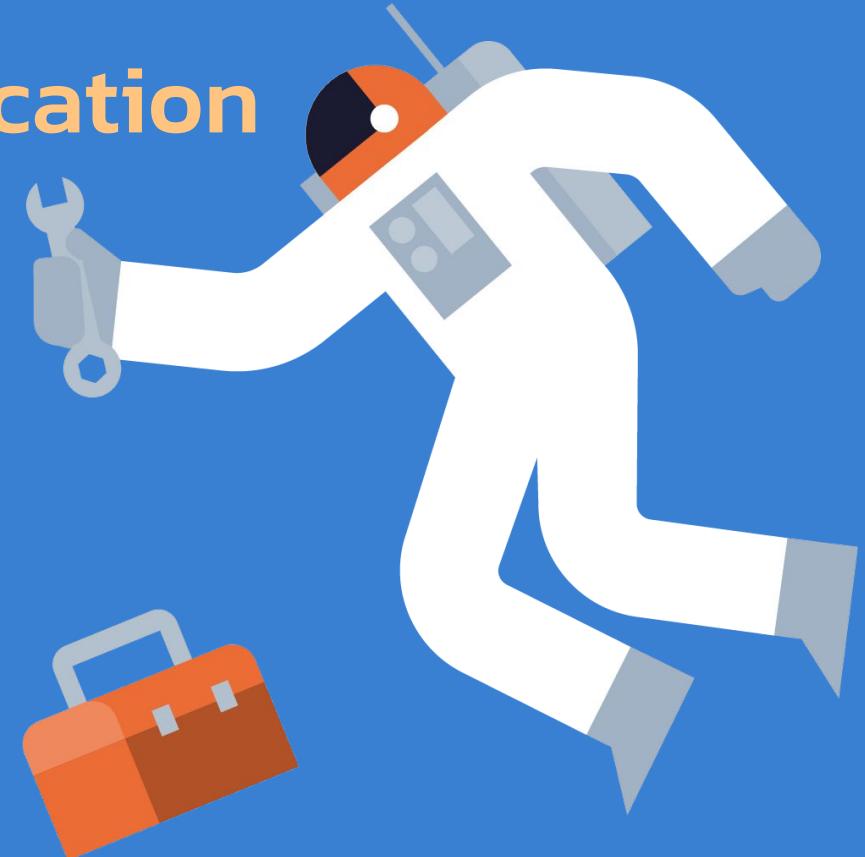
7. Start Gitpod and setup your application

8. Working with Cassandra Drivers



**Repository:**

[github.com/datastaxdevs/workshop-spring-reactive](https://github.com/datastaxdevs/workshop-spring-reactive)



# Agenda

01



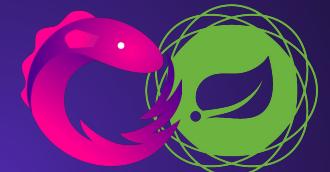
PetClinic  
Architecture & Use Case

02



Cassandra Database  
The Art of Data Modelling

03



Spring Reactive  
Reactive vs Async

04



Spring Reactive  
Boot and WebFlux

05



User Interface  
Angular

06



Game &  
Resources

# Spring Framework : Dependency Injection



## Spring Framework

Provides core support for dependency injection, transaction management, web apps, data access, messaging, and more.

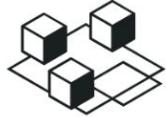
```
public interface GreetingServices {  
    String sayHello();  
}
```

```
@Component("greeting.fr")  
public class GreetingServicesFR implements GreetingServices {  
    public String sayHello() { return "Bonjour !"; }  
}
```

```
@Component("greeting.en")  
public class GreetingServicesEN implements GreetingServices {  
    public String sayHello() { return "Hello !"; }  
}
```

```
@RestController  
public class MyAPI {  
  
    @Autowired  
    @Qualifier("greeting.en") // <--  
    private GreetingServices greetingService;  
  
    @GetMapping  
    public String sayHello() {  
        return greetingService.sayHello();  
    }  
}
```

# What Spring Framework Can do



## Microservices

Quickly deliver production-grade features with independently evolvable microservices.



## Reactive

Spring's asynchronous, nonblocking architecture means you can get more from your computing resources.



## Cloud

Your code, any cloud—we've got you covered. Connect and scale your services, whatever your platform.



## Web apps

Frameworks for fast, secure, and responsive web applications connected to any data store.



## Serverless

The ultimate flexibility. Scale up on demand and scale to zero when there's no demand.



## Event Driven

Integrate with your enterprise. React to business events. Act on your streaming data in realtime.



## Batch

Automated tasks. Offline processing of data at a time to suit you.

# Spring Framework Projects



<https://spring.io/projects>



## Spring Boot

Takes an opinionated view of building Spring applications and gets you up and running as quickly as possible.



## Spring Data

Provides a consistent approach to data access – relational, non-relational, map-reduce, and beyond.



## Spring Security

Protects your application with comprehensive and extensible authentication and authorization support.



## Spring Batch

Simplifies and optimizes the work of processing high-volume batch operations.



## Spring Cloud

Provides a set of tools for common patterns in distributed systems. Useful for building and deploying microservices.



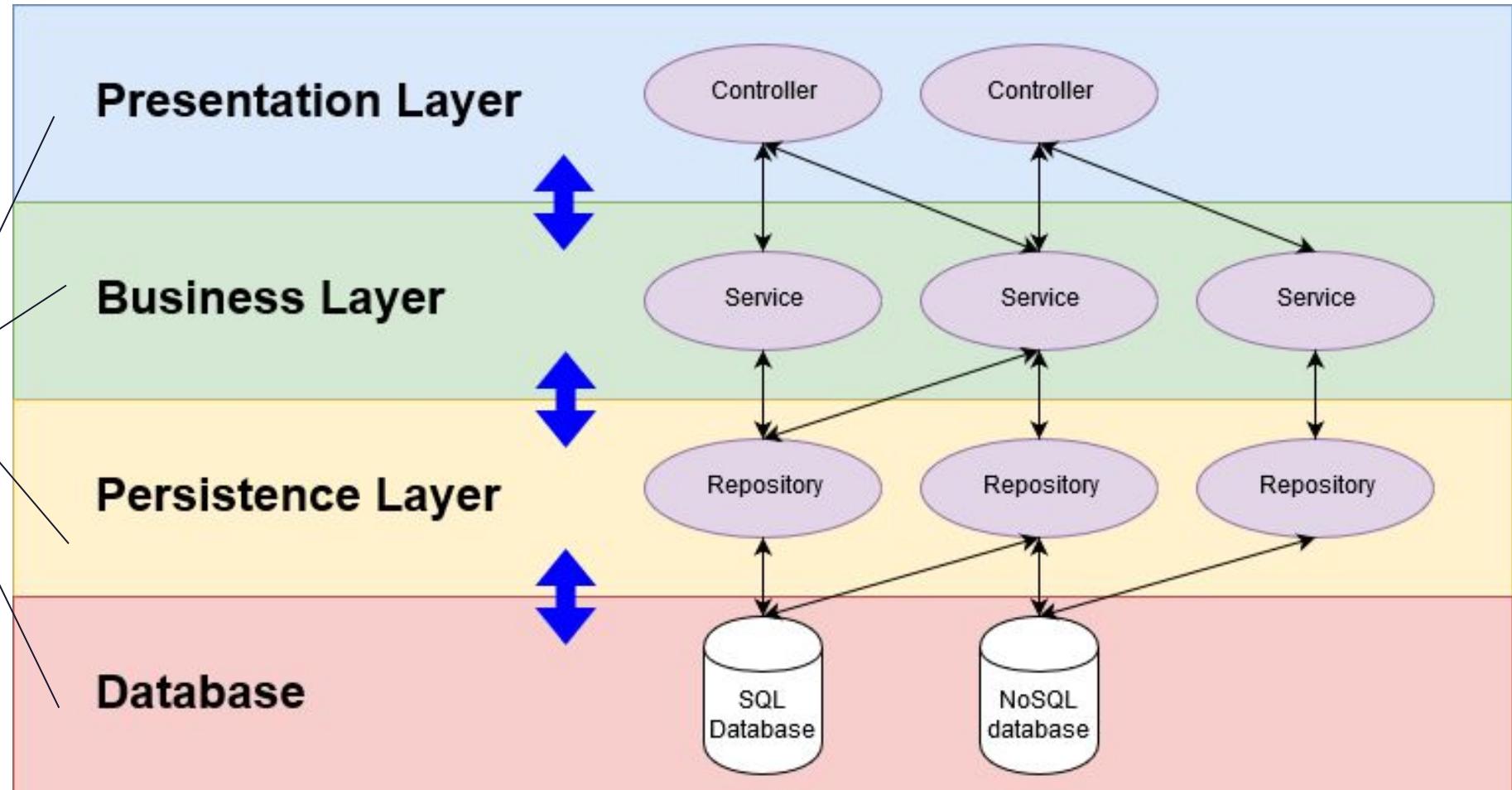
## Spring REST Docs

Lets you document RESTful services by combining hand-written documentation with auto-generated snippets produced with Spring MVC Test or REST Assured.

... and many more

DataStax Developers

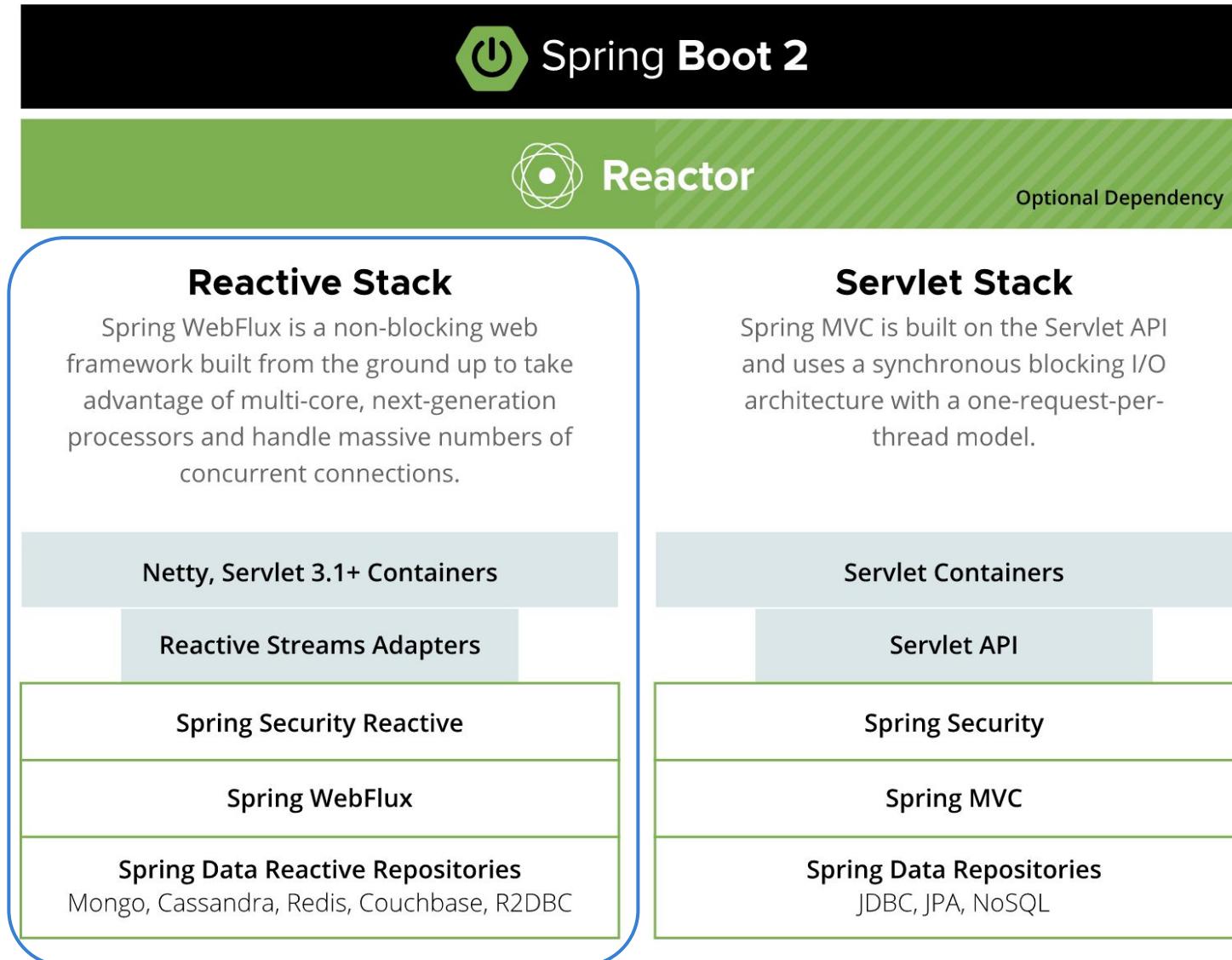
# Spring Boot Application



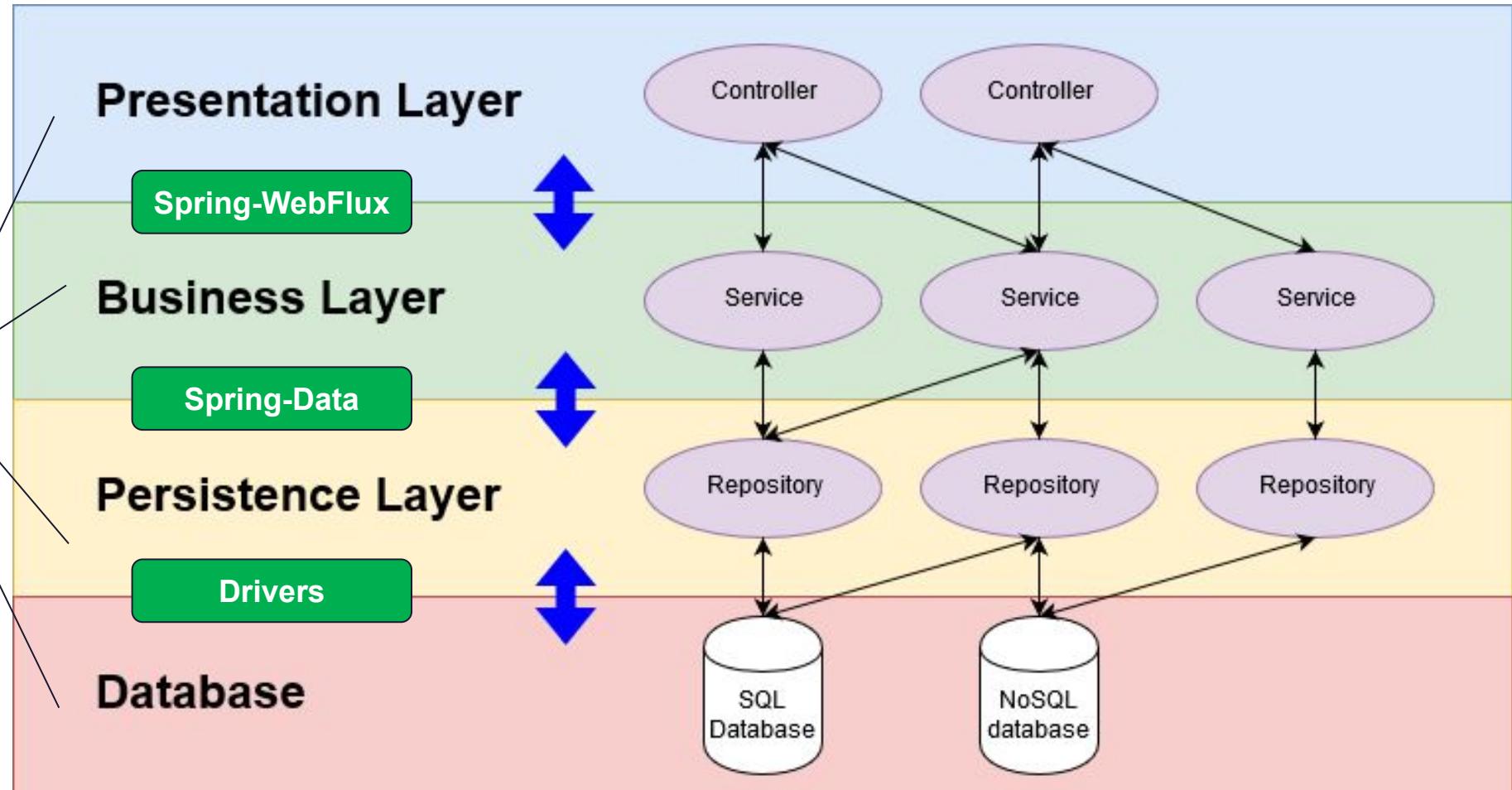
# Spring Reactive



<https://spring.io/reactive>



# Spring Boot Application



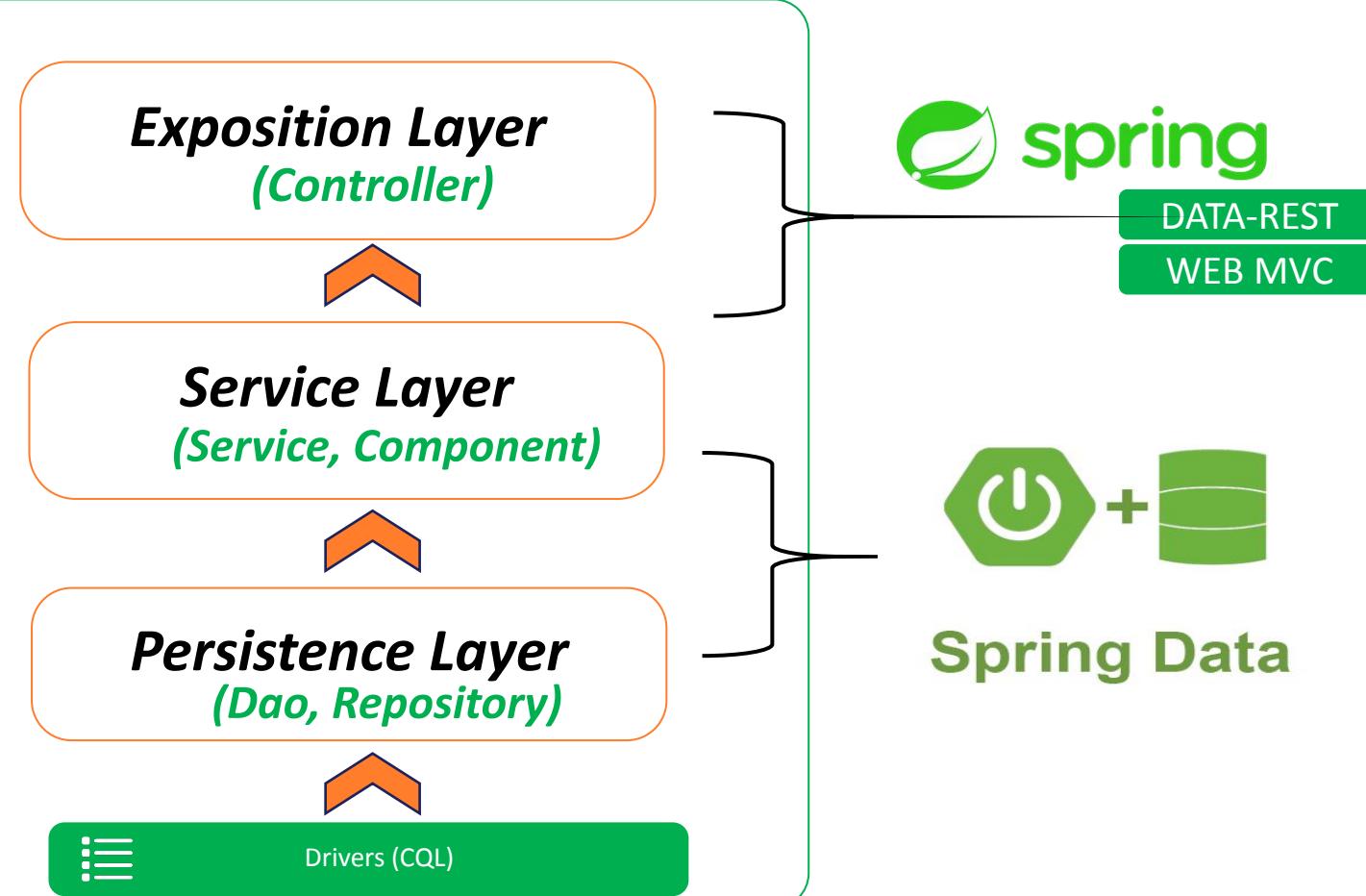
# Spring boot Starters



<https://spring.io/reactive>



- Security**
- Monitoring**
- Configuration**



# Spring Boot Application

Spring  
Boot

Spring  
security

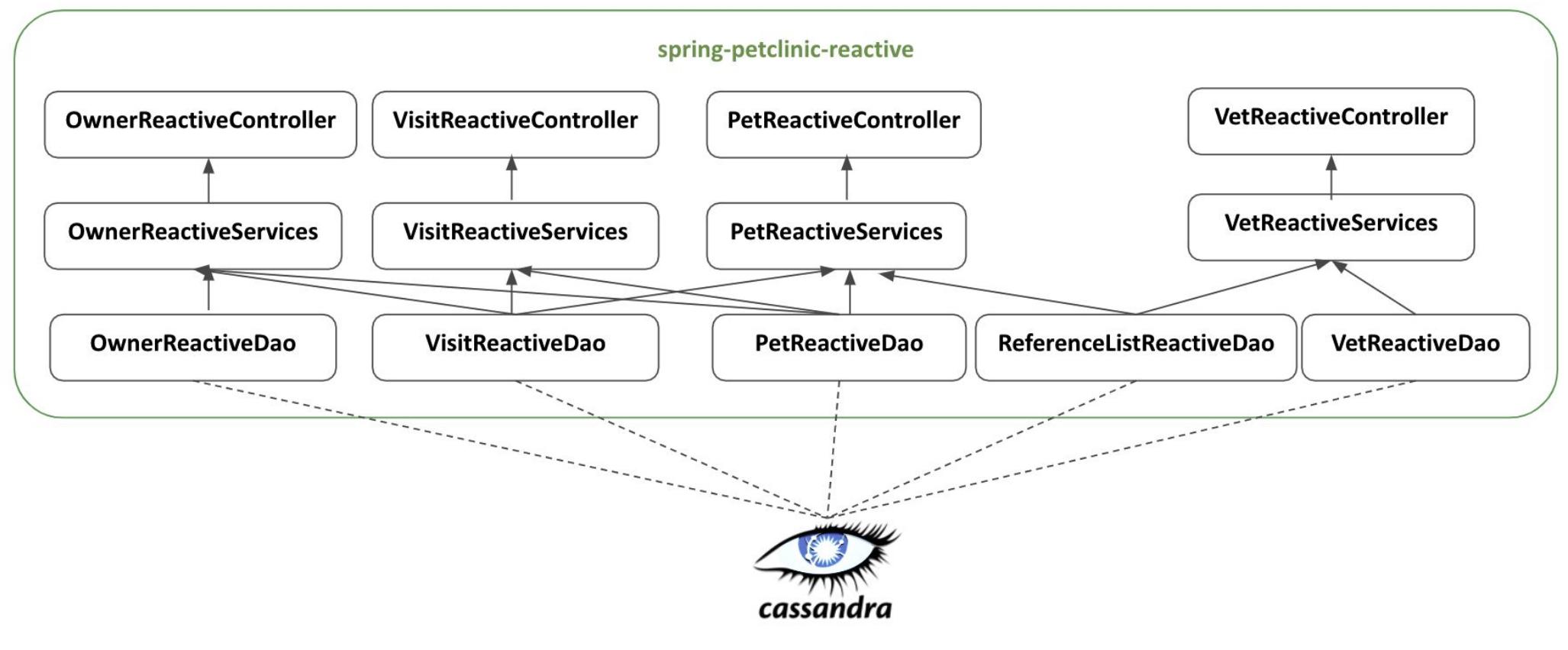
Spring  
Weblux

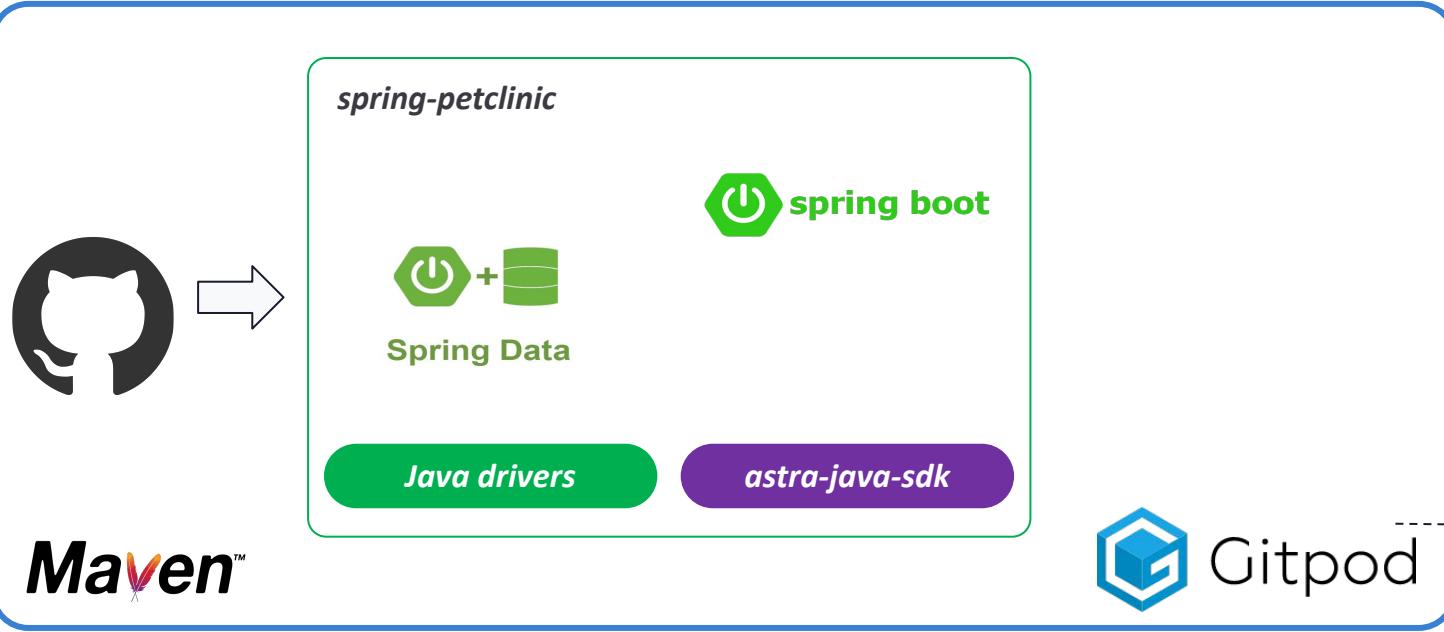
Spring  
Actuator

Spring  
Test

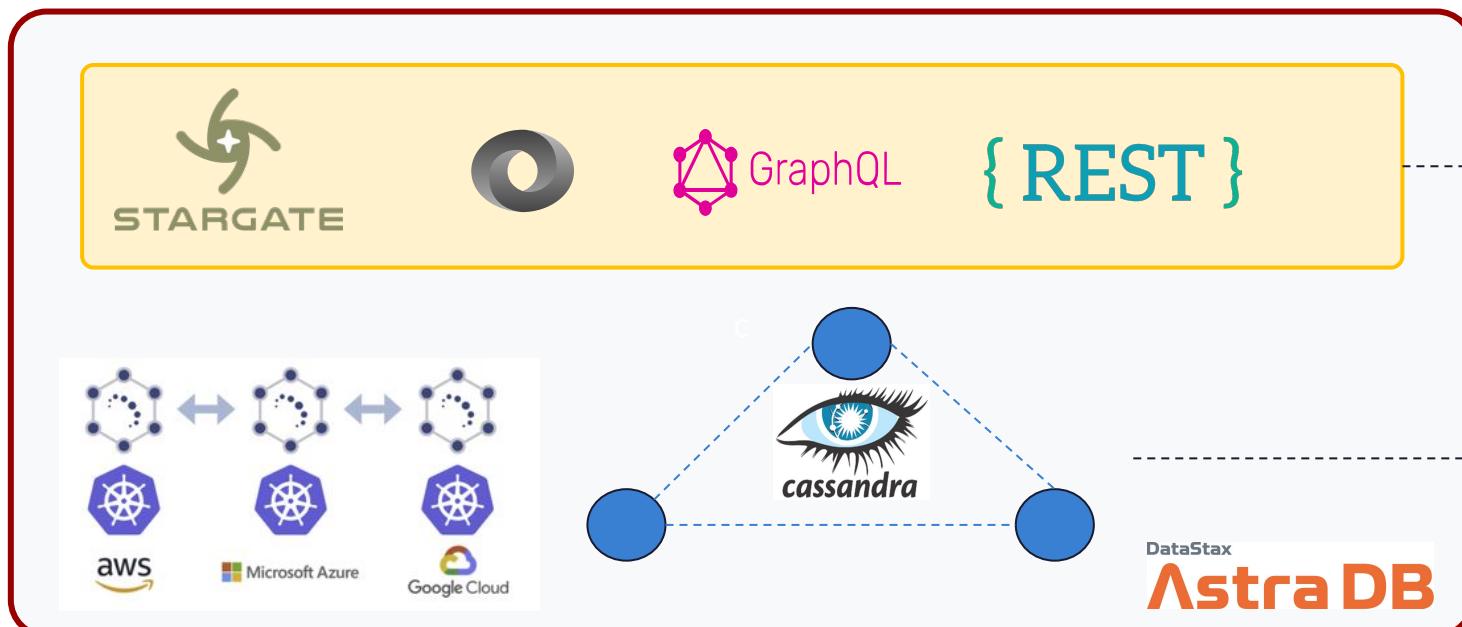
Spring  
Cloud

Springfox





**IDE Gitpod.io**  
A free cloud-based IDE



**OSS Stargate.io**  
A data gateway to allow multiple usages



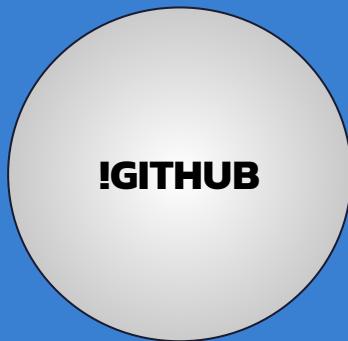
**OSS Apache Cassandra**  
A Column oriented NoSQL Database



# HandsOn: 9 and 10

9. Working with Spring Data

10. Working with Spring WebFlux



**Repository:**

[github.com/datastaxdevs/workshop-spring-reactive](https://github.com/datastaxdevs/workshop-spring-reactive)



# Agenda

01



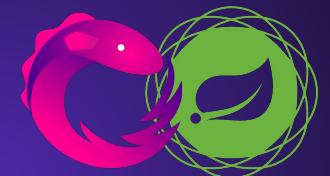
PetClinic  
Architecture & Use Case

02



Cassandra Database  
The Art of Data Modelling

03



Spring Reactive  
Reactive vs Async

04



Spring Reactive  
Boot and WebFlux

05



User Interface  
Angular

06



Game &  
Resources



# Spring Pet Clinic Angular

<https://github.com/spring-petclinic/spring-petclinic-angular>

The screenshot shows the 'Owners' section of the Spring Pet Clinic Angular application. At the top, there is a navigation bar with links for HOME, OWNERS, VETERINARIANS, PET TYPES, and SPECIALTIES. Below the navigation bar is a table titled 'Owners' with columns for Name, Address, City, Telephone, and Pets. The table contains eight rows of data. At the bottom left is a button labeled 'Add Owner'. At the bottom center is the Spring logo.

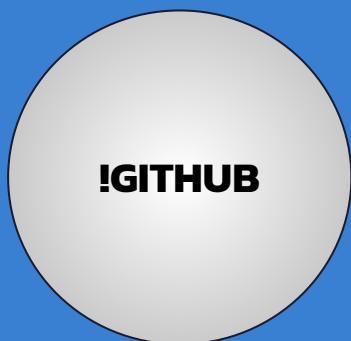
Name	Address	City	Telephone	Pets
George Franklin	110 W. Liberty St.	Madison	6085551023	Leo
Betty Davis	638 Cardinal Ave.	Sun Prairie	6085551749	Basil
Eduardo Rodriguez	2693 Commerce St.	McFarland	6085558763	Jewel Rosy
Harold Davis	563 Friendly St.	Windsor	6085553198	Iggy
Peter McTavish	2387 S. Fair Way	Madison	6085552765	George
Maria Escobito	345 Maple St.	Madison	6085557683	Mulligan
David Schroeder	2749 Blackhawk Trail	Madison	6085559435	Freddy
Carlos Estaban	2335 Independence La.	Waunakee	6085555487	Lucky Sly

<https://github.com/spring-petclinic/spring-petclinic-reactjs> exists



# HandsOn: 11

## 11. Working with Angular UI



**Repository:**

[github.com/datastaxdevs/workshop-spring-reactive](https://github.com/datastaxdevs/workshop-spring-reactive)



# Agenda

01



PetClinic  
Architecture & Use Case

02



Cassandra Database  
The Art of Data Modelling

03



Spring Reactive  
Reactive vs Async

04



Spring Reactive  
Boot and WebFlux

05



User Interface  
Angular

06



Game &  
Resources

menti.com



Mentimeter

# Claim your **FREE** Certification Voucher



Vouchers (normally 145\$ each)

- valid for 3 months,
- valid for 2 attempts

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

Claim using the link:  
<http://dtsx.io/workshop-voucher>



# LEARNING PATHS at the [academy.datastax.com](https://academy.datastax.com)





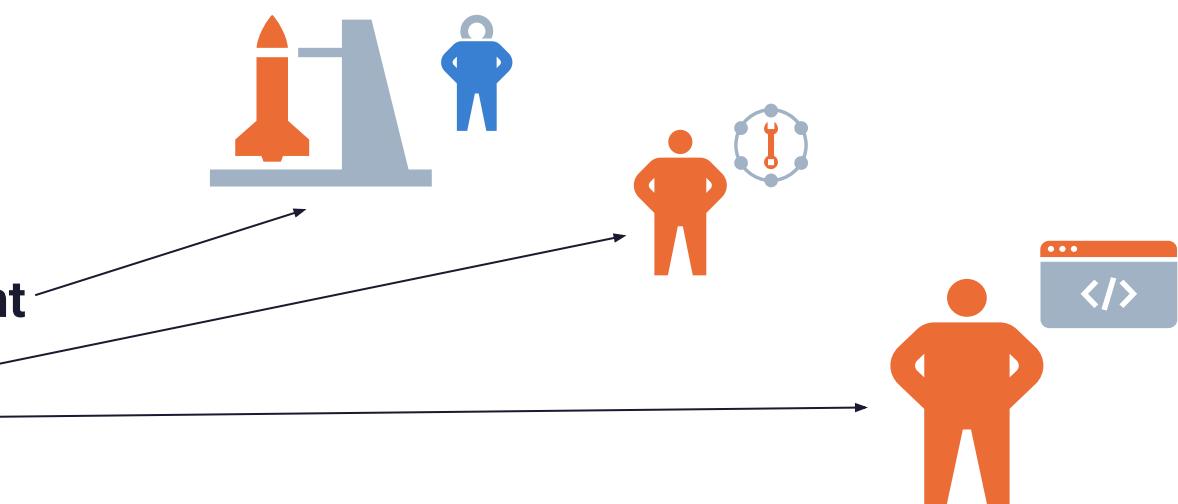
DataStax Developers

<https://dtsx.io/staxgiving>

500\$ + 250\$ + 250\$ AMAZON gift cards to win

#### CONDITIONS

- **#engage-with-us** : Tasks that will earn you 1 point
- **#work-with-us**: Tasks that will earn you 5 points
- **#build** : Tasks that will earn you 10 points.



#### DUE DATE

THANKSGIVING DAY => 25/11

Winners randomly chosen **live** on 12/8

# Weekly Workshops

[datastax.com/workshops](http://datastax.com/workshops)

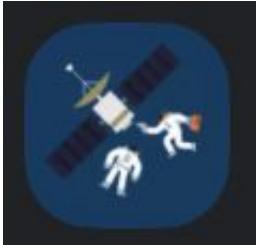
The screenshot shows the DataStax Developers YouTube channel page. At the top, there's a large banner with the text "LEVEL UP with the DataStax Developers". Below the banner, the channel name "DataStax Developers" and "8,1 k abonnés" are displayed. The main content area features a grid of video thumbnails for various workshops, such as "Building Microservices with Cassandra + Spring", "Advanced Data Modeling in Apache Cassandra™", and "Apache Cassandra™ Certification Preparation". Each thumbnail includes a small image of a developer character, the video title, duration, view count, and upload date.

The screenshot shows the DataStax Developers website's "Upcoming Live Events" section. It displays four event cards with the following details:

- Apache Cassandra™ Certification Preparation**: Multiple Dates | NoSQL | Beginner. Includes a "Register Now" button.
- Build Microservices with Apache Cassandra™!**: Feb 17 or Feb 18 | NoSQL | Beginner. Includes a "Register Now" button.
- Certification Exam Preparation Workshop**: Multiple Dates. Includes a "Register Now" button.
- Cloud-Native Workshop: Build Spring Microservices with Apache Cassandra™**: Multiple Dates. Includes a "Register Now" button.
- Learn how to build a Serverless Game!**: Feb 24 or Feb 25 | Game Development | Beginner. Includes a "START" button and a game controller icon.
- Build Microservices with Cassandra & Quarkus**: March 11 | Microservices | Beginner. Includes a "Register Now" button.
- Cloud-Native Workshop: Build a serverless game with the JAMStack!**: THU MAR 11 2021. Includes a "Register Now" button.

# Join our 17k Discord Community

## DataStax Developers



Screenshot of the DataStax Developers Discord server interface:

- Left Sidebar:** Shows categories like "DataStax Developers", ".WELCOME", "code-of-conduct", "introductions", "upcoming-events", "useful-resources", "memes", "your-ideas", ".WORKSHOPS", ".ASTRADB", ".APACHE CASSANDRA", and a "Cedrick Lun..." profile.
- Top Bar:** Includes a search bar, a link to a YouTube video, and user status indicators.
- # workshop-chat Channel:** Displays a message from a user named RIGGITYREKT about mixed version testing in DSE.
- Right Panel:** Lists the "PRESENTER - 1" (David Jones-Gilardi), "HELPER - 7" (012345, AaronP, B1nary, Chelsea Navo, Jeremy Hanna, John Sanda, Patrick\_McFadin), and "EN LIGNE - 560" members (including -samu-, 6304-42J8, Aahlya, Abdurahim, abhi3pathi, Abhiis.s, Abhineet, Abirsh).

[dtsx.io/discord](https://dtsx.io/discord)

DataStax Developers

# DataStax Developers

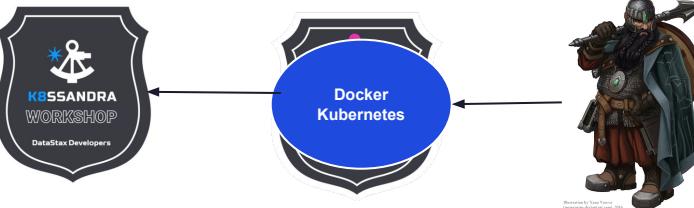
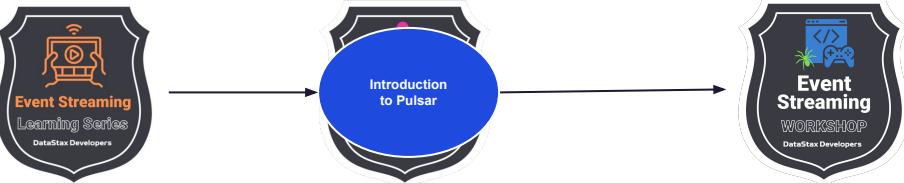
Thank you!



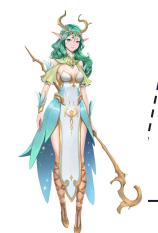
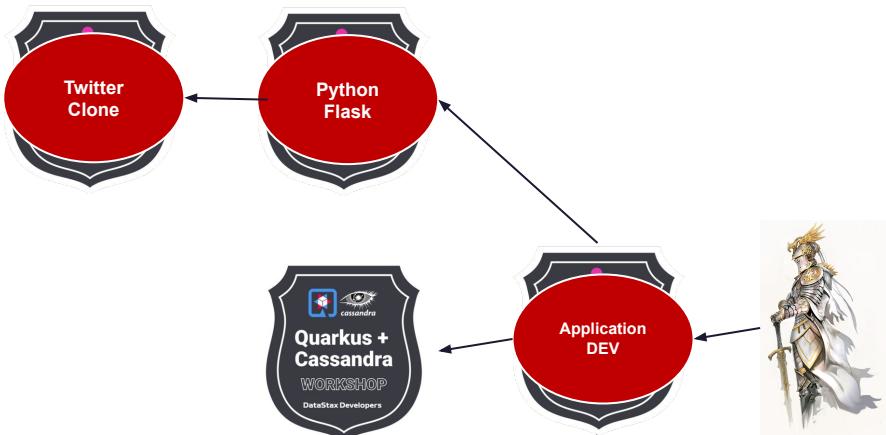
Subscribe



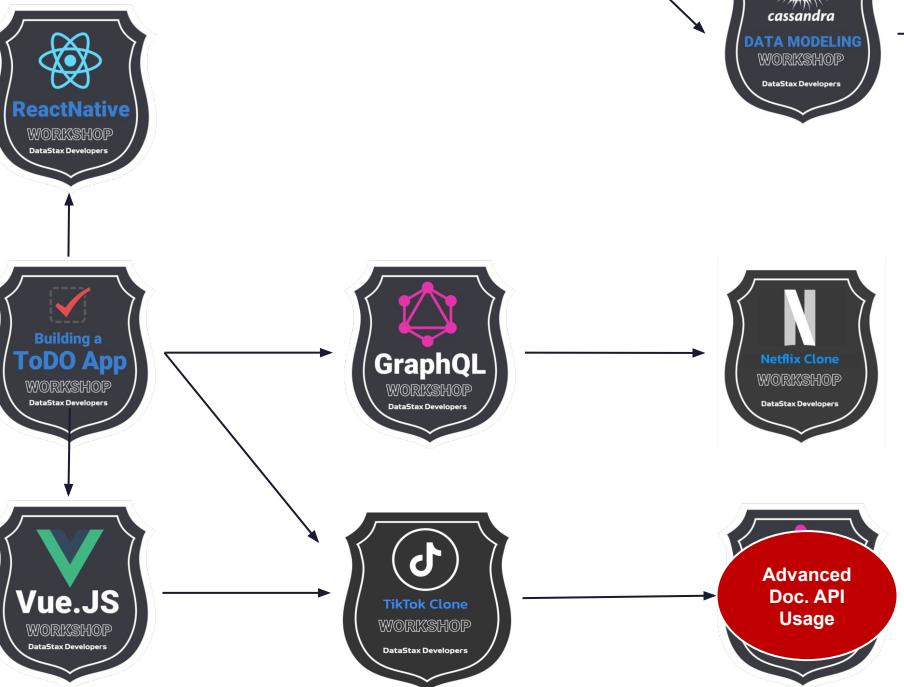
Streaming  
Engineer



Data  
Engineer



DEV  
FRONTEND



SRE



DEV  
BACKEND

Illustration by Yann Nézet-Séguin, 2021  
For the DataStax Community Foundation, a group of volunteers.