Cassandra Day
2022

Stargate – An OSS API Layer for C*

Sponsored by DataStax

Agenda

01

Stargate Data gateway
What, Why and How

02 {REST}

Exploring Apis
Rest DML and DDL

03

Exploring Apis
Document oriented

04

Exploring Apis
GraphQL and gRPC

05

Tooling
SDK, K8ssandra

06

What's NEXT ?
v2 & revised arch

Agenda

01

Stargate Data gateway
What, Why and How

02 {REST}

Exploring Apis
Rest DML and DDL

03

Exploring Apis
Document oriented

04

Exploring Apis
GraphQL and gRPC

05

Tooling
SDK, K8ssandra

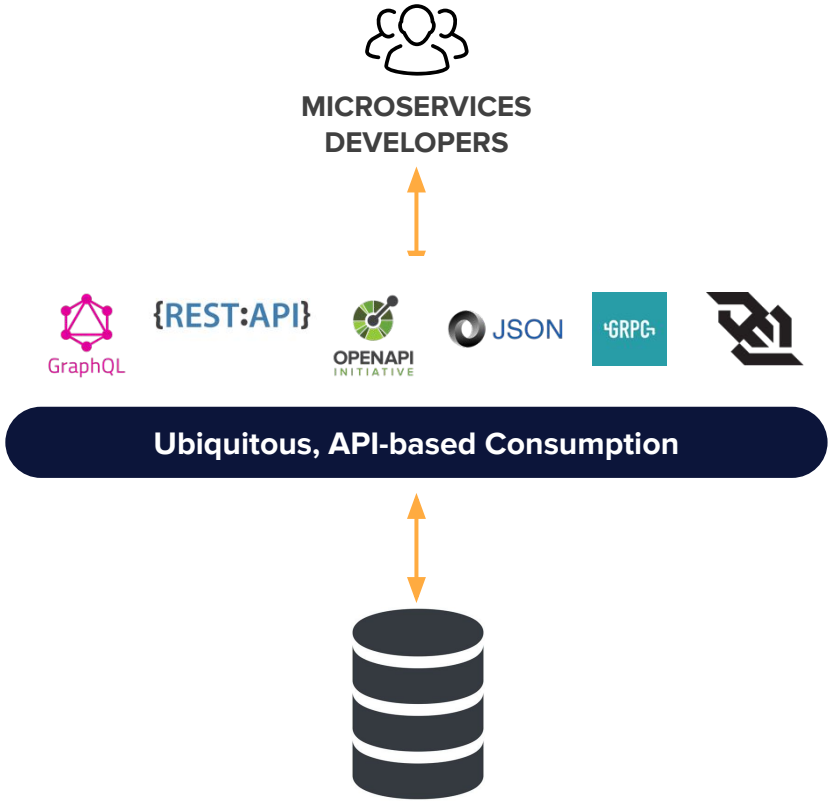
06

What's NEXT ?
v2 & revised arch

Data Gateway

> Developers want the option to use modern APIs and development gateways.

Cassandra is a database designed for the new standard and the associated APIs that facilitate the first choice of developers.

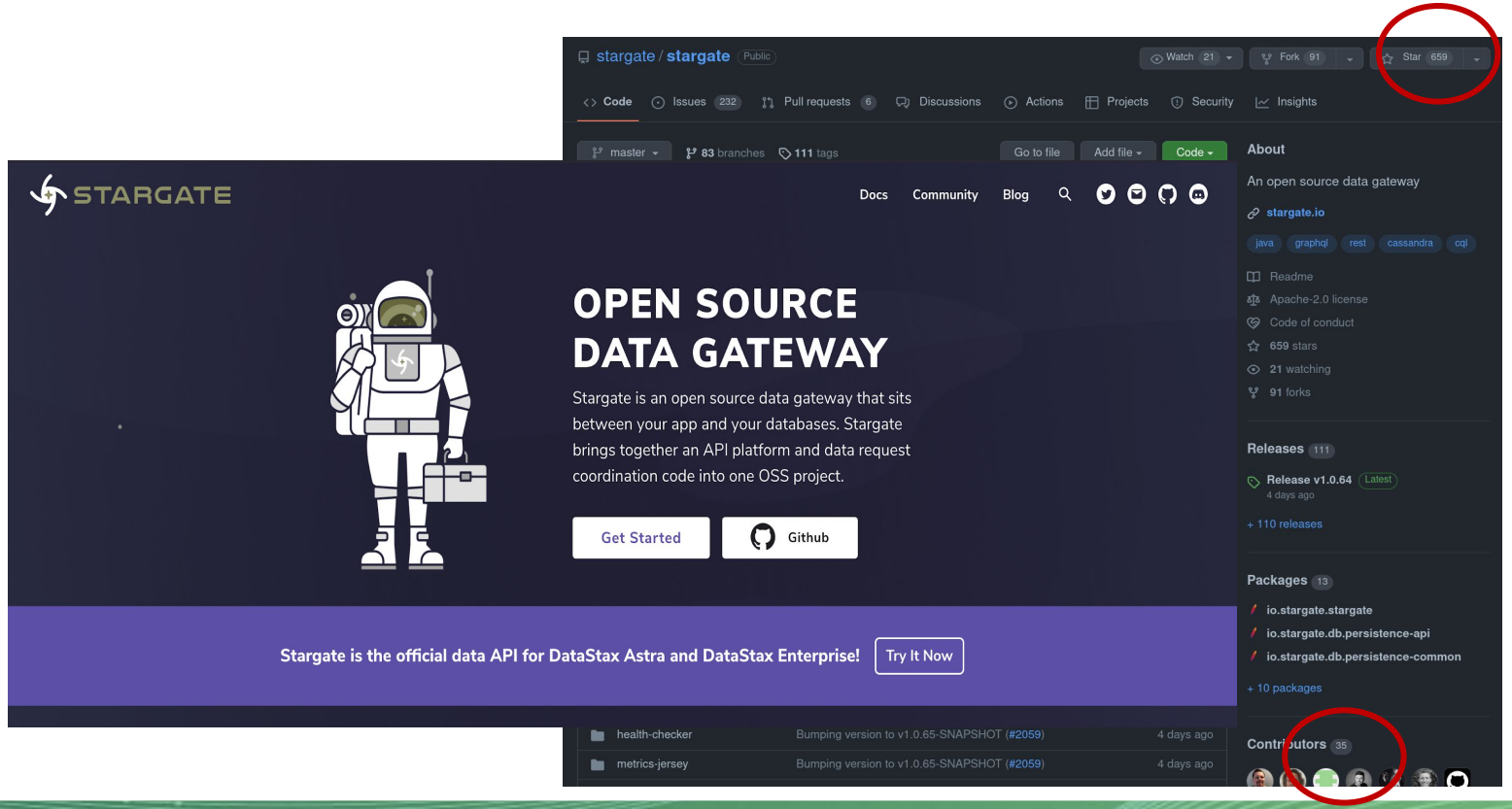


Stargate Overview

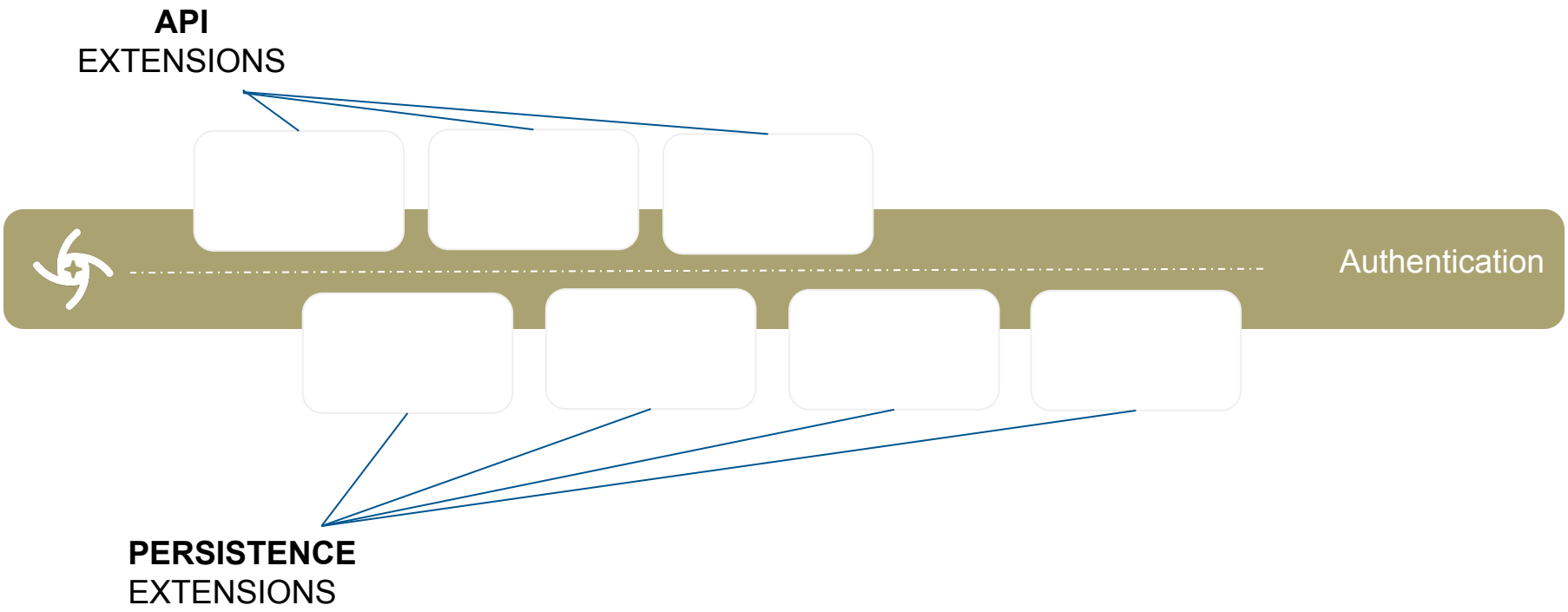
An open source API framework for data

Stargate makes it easy to use a database for any application workload by adding plugin support for new APIs, data types, and access methods

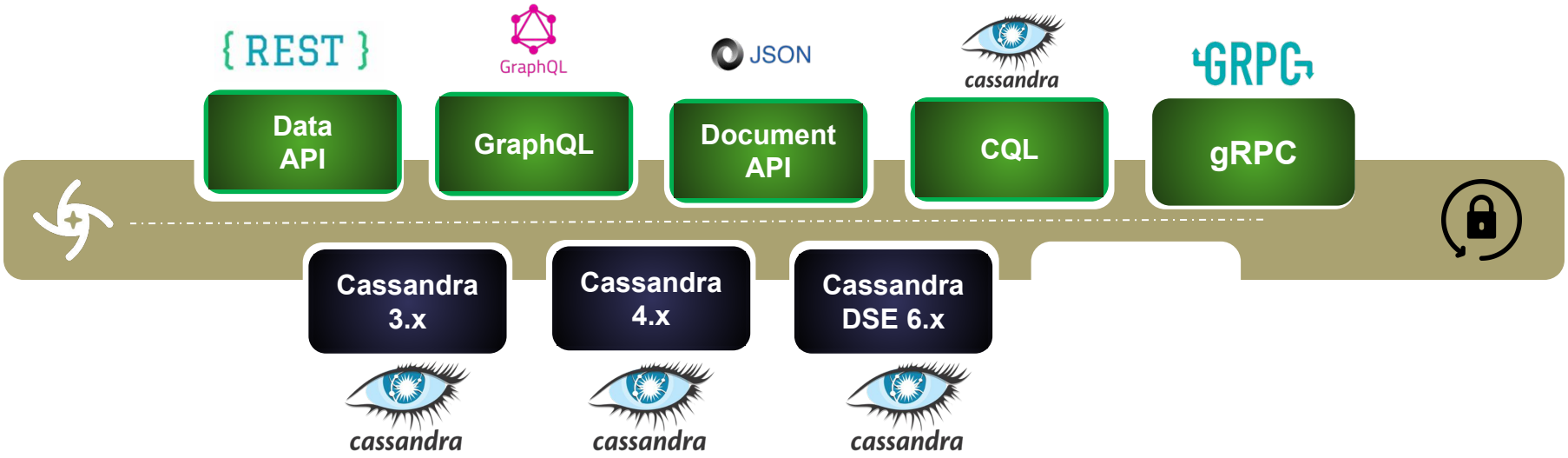
stargate.io

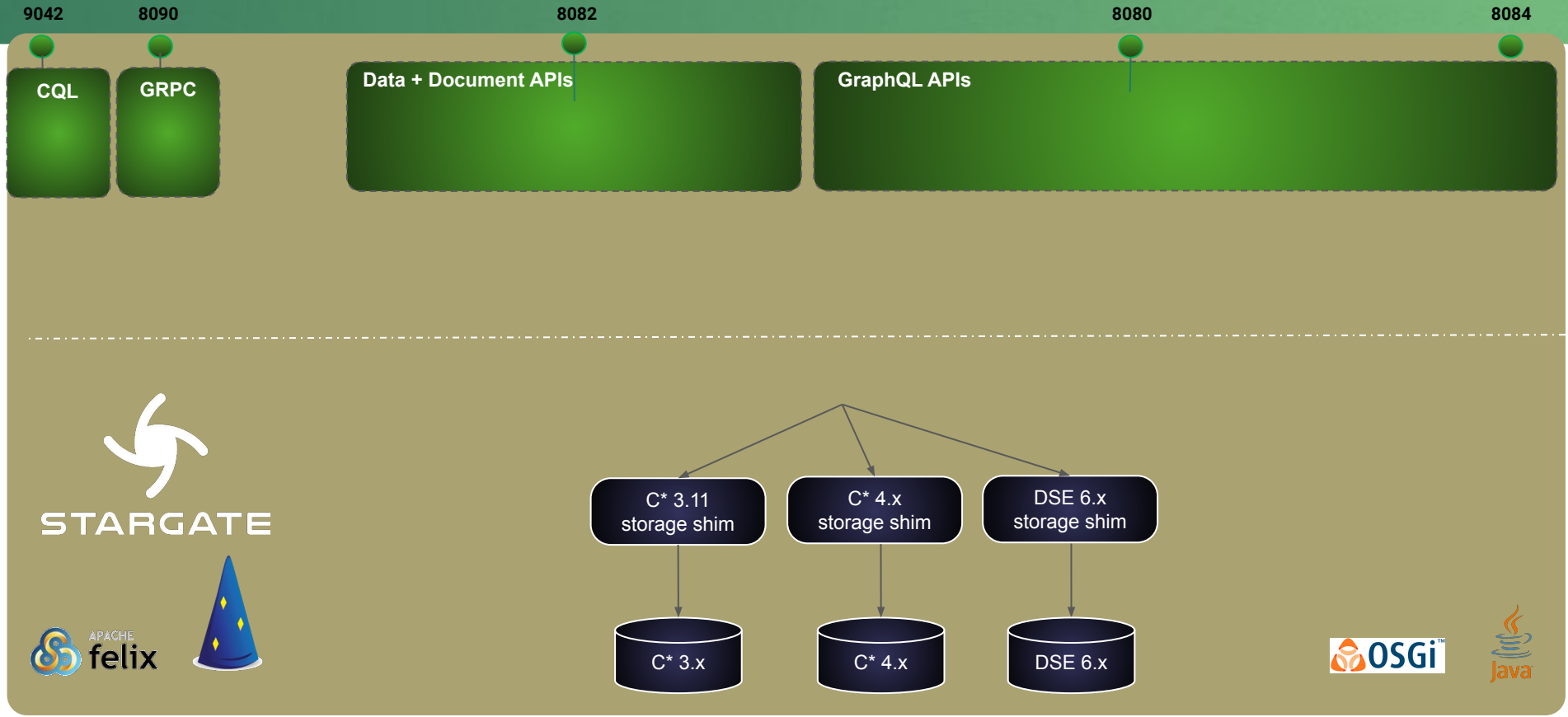

stargate / stargate
Public
Watch 21
Fork 91
Star 659
Code
Issues 232
Pull requests 6
Discussions
Actions
Projects
Security
Insights
master
83 branches
111 tags
Go to file
Add file
Code
About
An open source data gateway
stargate.io
java
graphql
rest
cassandra
cql
Readme
Apache-2.0 license
Code of conduct
659 stars
21 watching
91 forks
Releases 111
Release v1.0.64
Latest
4 days ago
+ 110 releases
Packages 13
io.stargate.stargate
io.stargate.db.persistence-api
io.stargate.db.persistence-common
+ 10 packages
Contributors 35
STARGATE
Docs
Community
Blog
OPEN SOURCE
DATA GATEWAY
Stargate is an open source data gateway that sits between your app and your databases. Stargate brings together an API platform and data request coordination code into one OSS project.
Get Started
Github
Stargate is the official data API for DataStax Astra and DataStax Enterprise!
Try It Now
health-checker
Bumping version to v1.0.65-SNAPSHOT (#2059)
4 days ago
metrics-jersey
Bumping version to v1.0.65-SNAPSHOT (#2059)
4 days ago

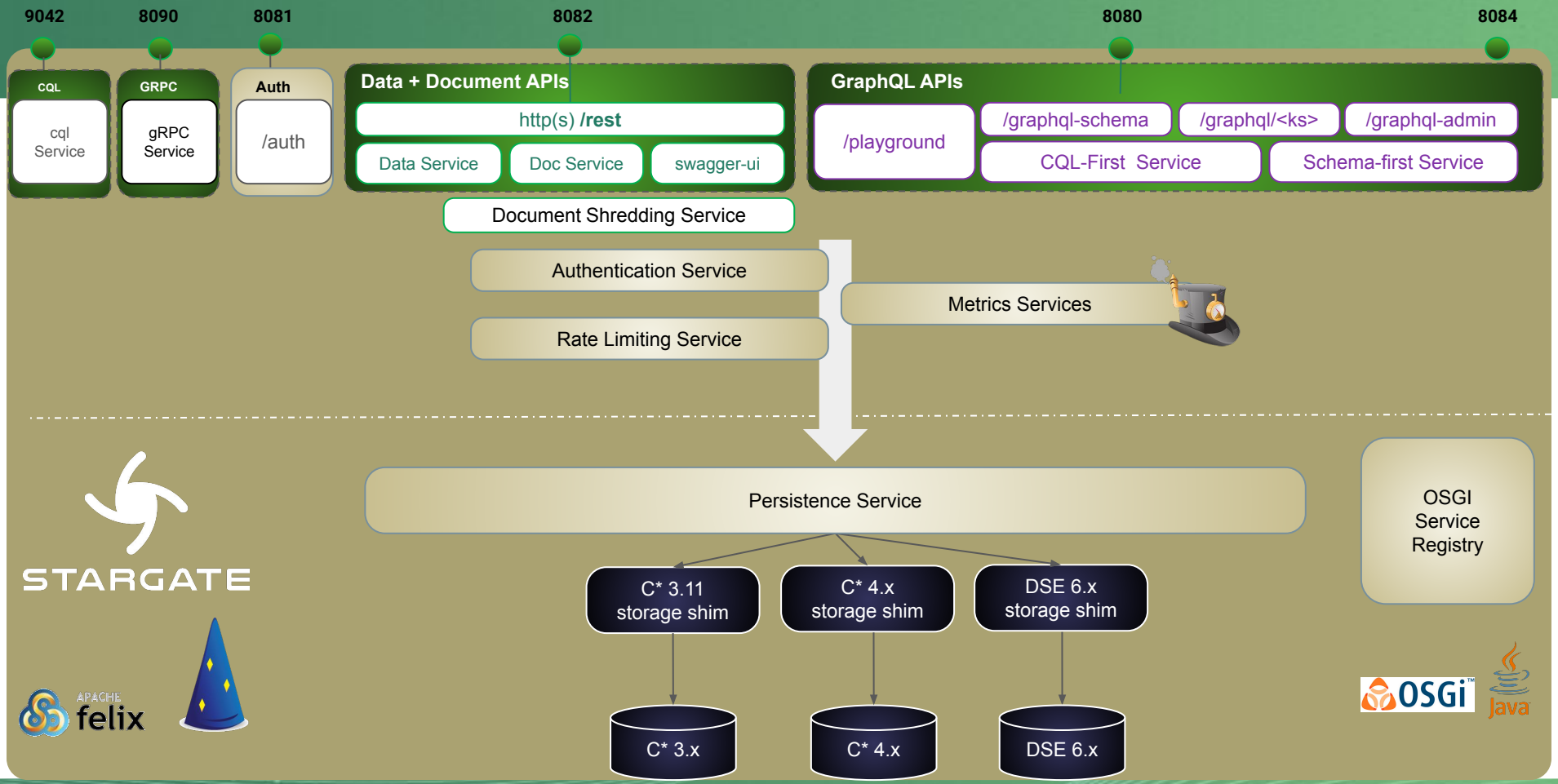
API Extensions and Persistence Extensions



API Extensions and Persistence Extensions



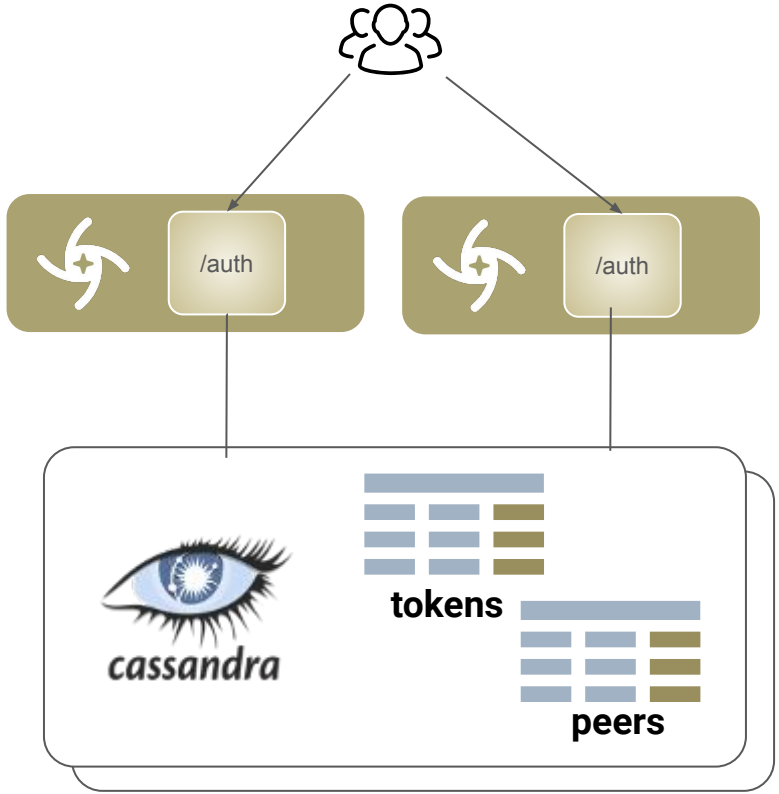

9042
8090
8082
8080
8084
CQL
GRPC
Data + Document APIs
GraphQL APIs
STARGATE
C* 3.11 storage shim
C* 4.x storage shim
DSE 6.x storage shim
C* 3.x
C* 4.x
DSE 6.x
APACHE felix
OSGi
Java


9042
8090
8081
8082
8080
8084
CQL
cql Service
GRPC
gRPC Service
Auth
/auth
Data + Document APIs
http(s) /rest
Data Service
Doc Service
swagger-ui
GraphQL APIs
/playground
/graphql-schema
/graphql/<ks>
/graphql-admin
CQL-First Service
Schema-first Service
Document Shredding Service
Authentication Service
Rate Limiting Service
Metrics Services
STARGATE
Persistence Service
OSGI Service Registry
C* 3.11 storage shim
C* 4.x storage shim
DSE 6.x storage shim
C* 3.x
C* 4.x
DSE 6.x
APACHE felix
OSGi
Java

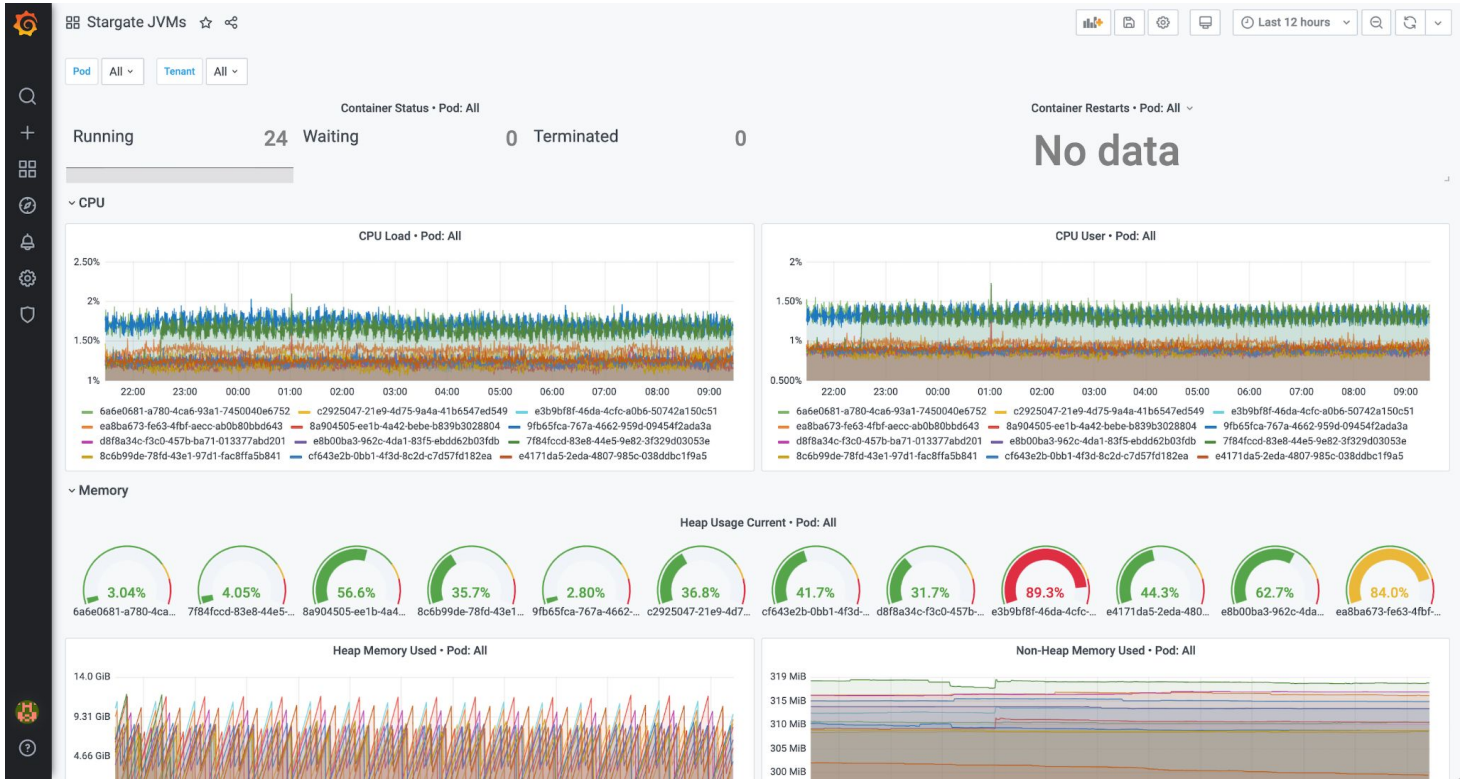
Authentication / Authorization

- Use Auth and get tokens (user+pwd)
- Tokens shared across instances
- Leverage on Cassandra roles (RBAC)
- `-Dstargate.auth_tokenttl` =X

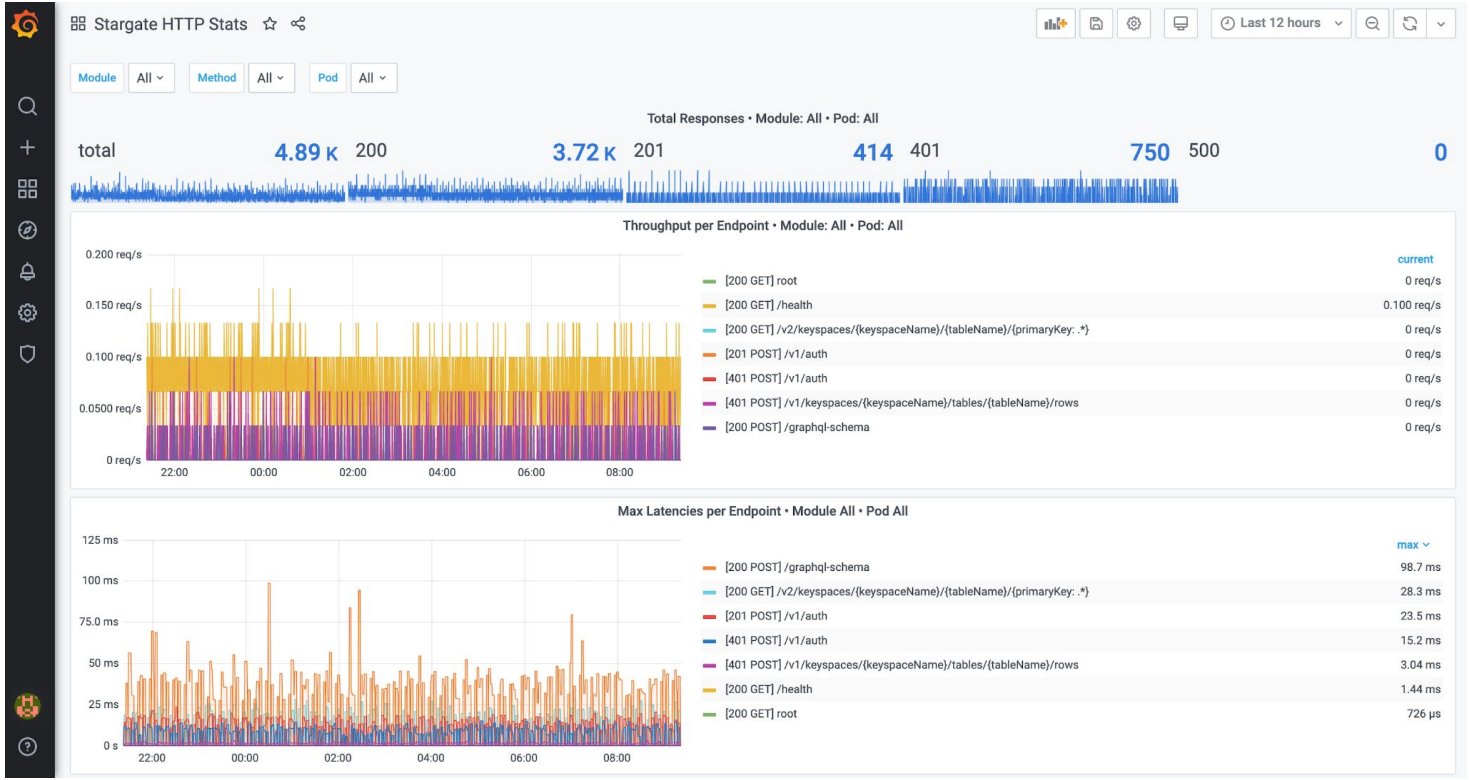
```
curl -L -X POST 'http://localhost:8081/v1/auth' \
  -H 'Content-Type: application/json' \
  --data-raw '{
    "username": "cassandra",
    "password": "cassandra"
}'
```



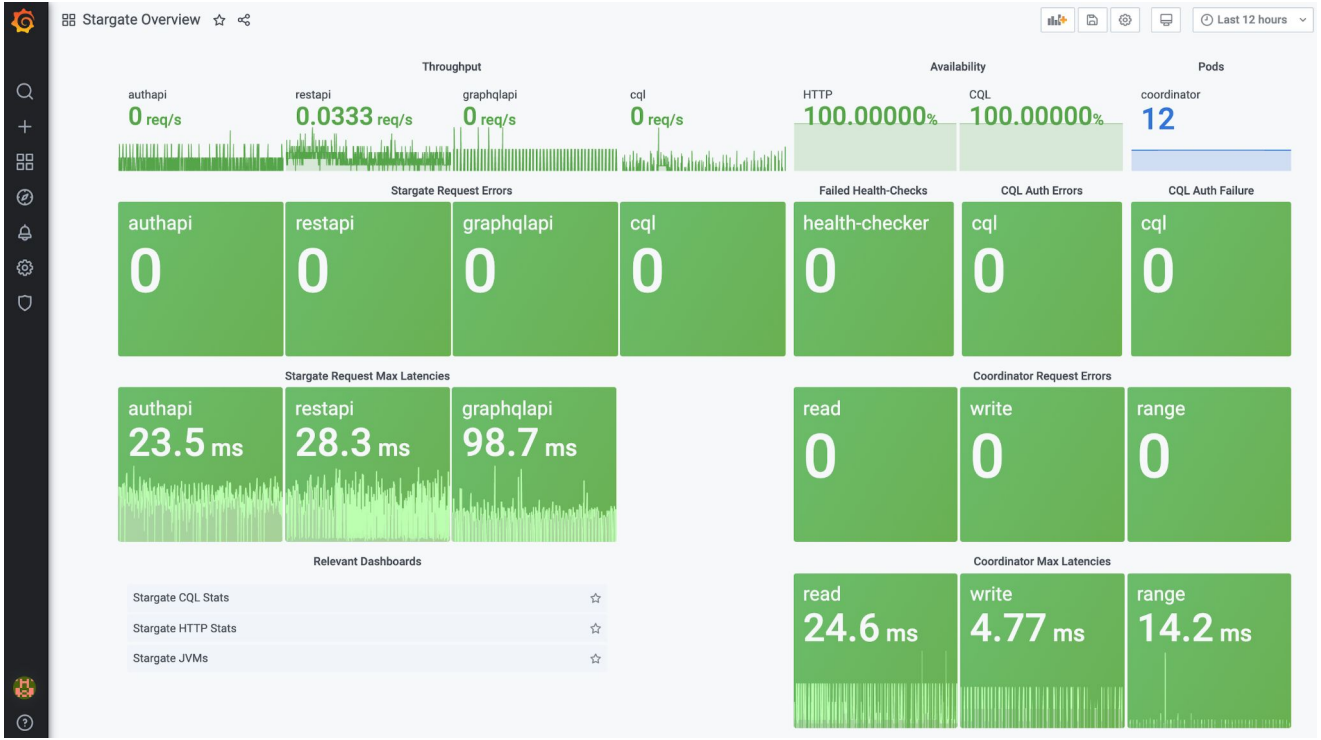
Monitoring (JVM)



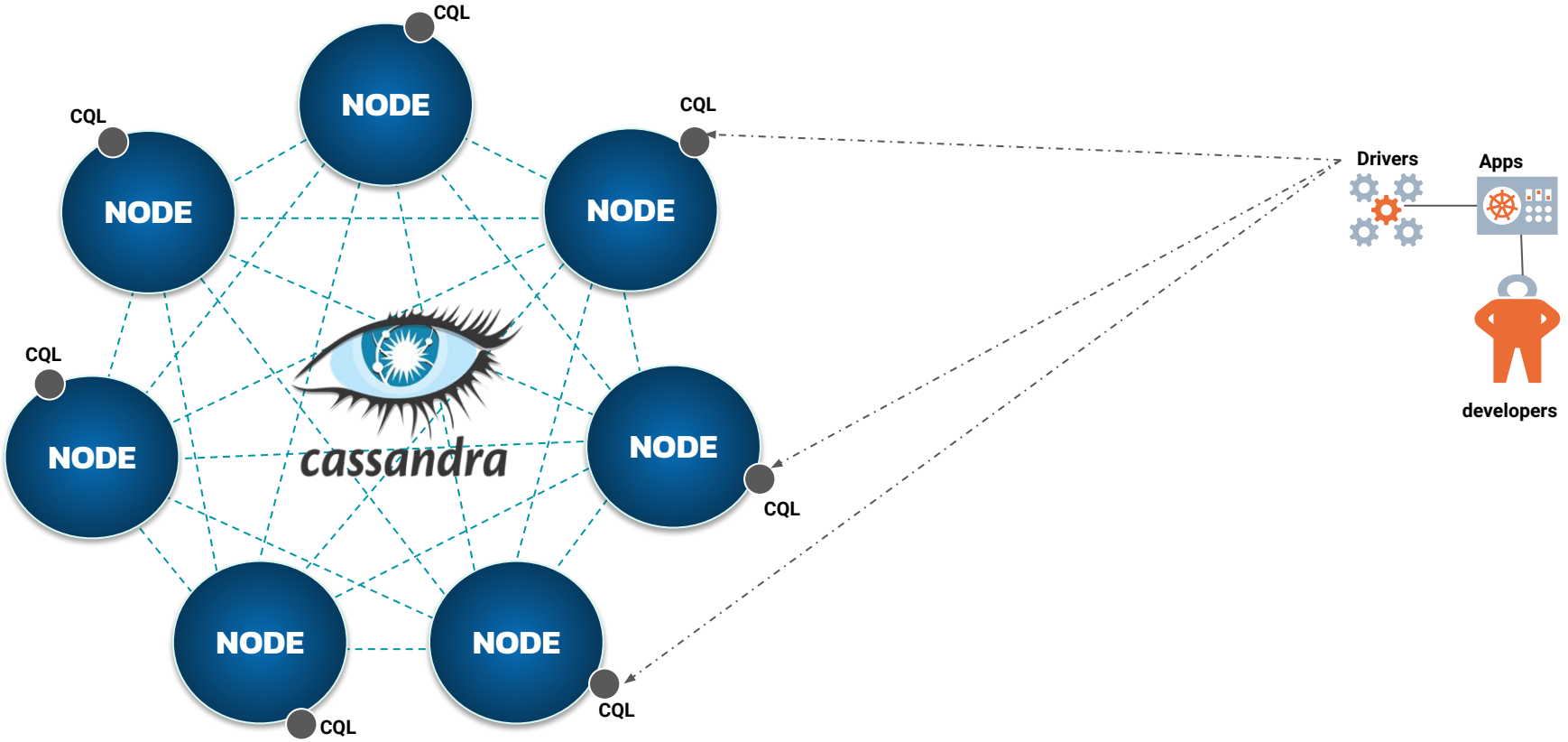
Monitoring (HTTP Traffic)



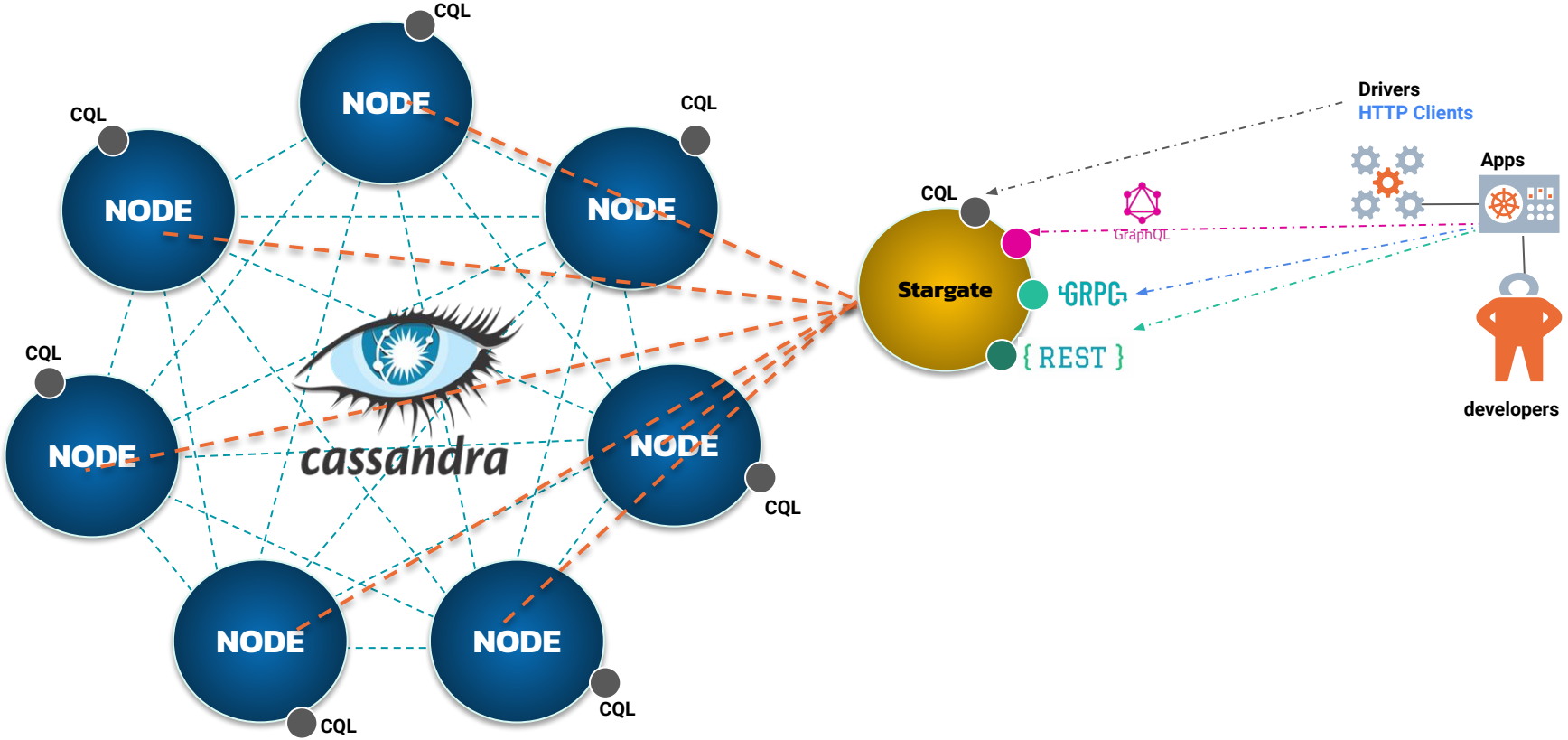
Monitoring (Services)



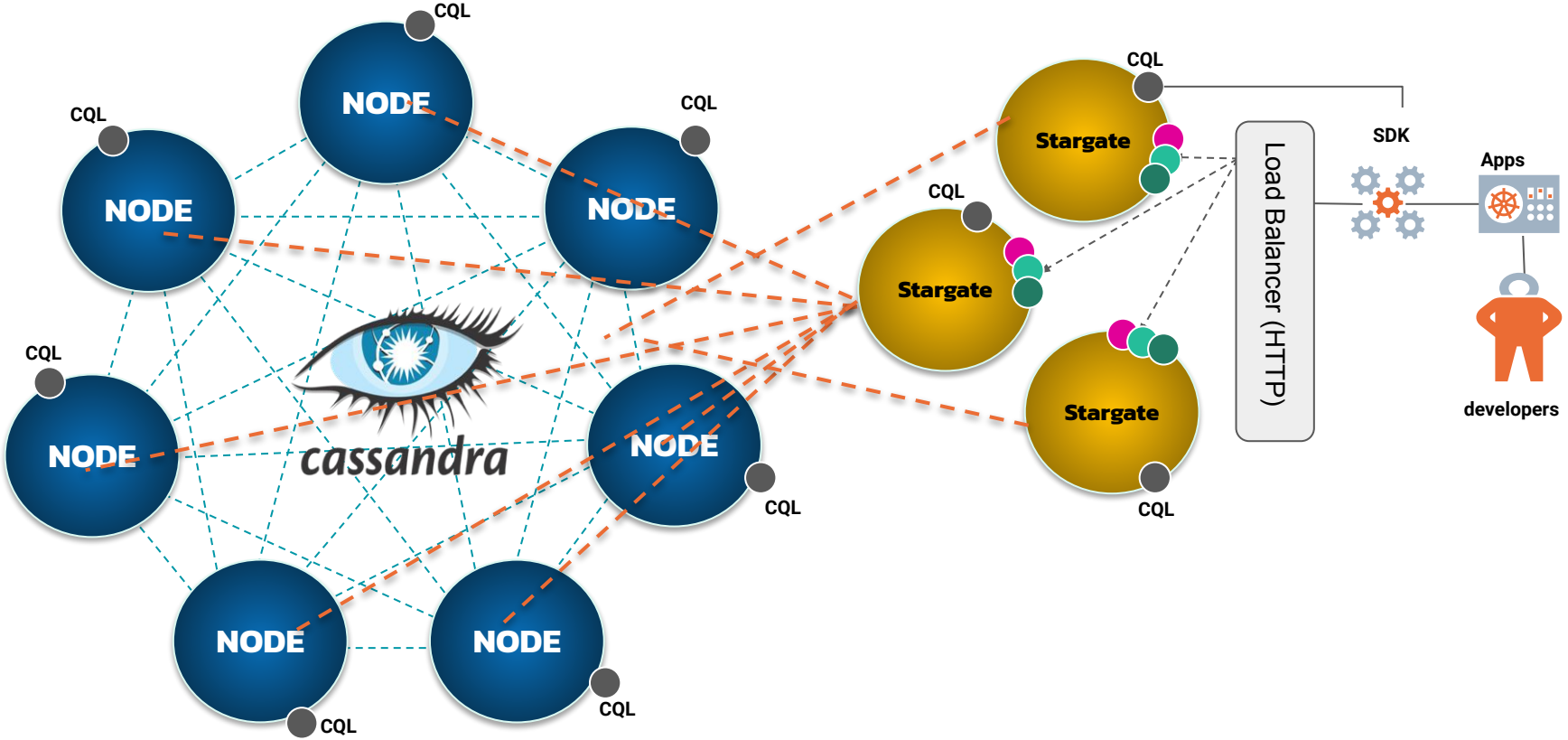
Connecting to your cluster (Before)



Connecting to your cluster (with Stargate)



Connecting to your cluster (with Stargates)



Agenda

01

Stargate Data gateway
What, Why and How

02 {REST}

Exploring Apis
Rest DML and DDL

03

Exploring Apis
Document oriented

04

Exploring Apis
GraphQL and gRPC

05

Tooling
SDK, K8ssandra

06

What's NEXT ?
v2 & revised arch

Stargate "REST" Api(s)

- Expose existing CQL schema as a RESTful endpoint, or create new CQL schema
- Most familiar API style for most teams
- Swagger integration

Quick start: https://stargate.io/docs/stargate/1.0/quickstart/quick_start-rest.html

REST Api(s) expose DDL and best practices DML

some CQL capabilities left out

GET	/v2/schemas/keyspaces/{keyspaceName}/tables/{tableName}/columns	Get all columns
POST	/v2/schemas/keyspaces/{keyspaceName}/tables/{tableName}/columns	Create a column
GET	/v2/schemas/keyspaces/{keyspaceName}/tables/{tableName}/columns/{columnName}	Get a column
PUT	/v2/schemas/keyspaces/{keyspaceName}/tables/{tableName}/columns/{columnName}	Update a column
DELETE	/v2/schemas/keyspaces/{keyspaceName}/tables/{tableName}/columns/{columnName}	Delete a column
DELETE	/v2/schemas/keyspaces/{keyspaceName}/tables/{tableName}/indexes/{indexName}	Drop an index from keyspace
GET	/v2/schemas/keyspaces/{keyspaceName}/tables/{tableName}/indexes	Get all indexes for a given table
POST	/v2/schemas/keyspaces/{keyspaceName}/tables/{tableName}/indexes	Add an index to a table's column
GET	/v2/schemas/keyspaces/{keyspaceName}	Get a keyspace
DELETE	/v2/schemas/keyspaces/{keyspaceName}	Delete a keyspace
GET	/v2/schemas/keyspaces	Get all keyspaces
POST	/v2/schemas/keyspaces	Create a keyspace
GET	/v2/schemas/keyspaces/{keyspaceName}/tables/{tableName}	Get a table
PUT	/v2/schemas/keyspaces/{keyspaceName}/tables/{tableName}	Replace a table definition

DDL (Schema)

DML (data)

GET	/v2/keyspaces/{keyspaceName}/{tableName}/rows	Retrieve all rows
GET	/v2/keyspaces/{keyspaceName}/{tableName}	Search a table
POST	/v2/keyspaces/{keyspaceName}/{tableName}	Add row
GET	/v2/keyspaces/{keyspaceName}/{tableName}/{primaryKey}	Get row(s)
PUT	/v2/keyspaces/{keyspaceName}/{tableName}/{primaryKey}	Replace row(s)
DELETE	/v2/keyspaces/{keyspaceName}/{tableName}/{primaryKey}	Delete row(s)
PATCH	/v2/keyspaces/{keyspaceName}/{tableName}/{primaryKey}	Update part of a row(s)

Agenda

01

Stargate Data gateway
What, Why and How

02 {REST}

Exploring Apis
Rest DML and DDL

03

Exploring Apis
Document oriented

04

Exploring Apis
GraphQL and gRPC

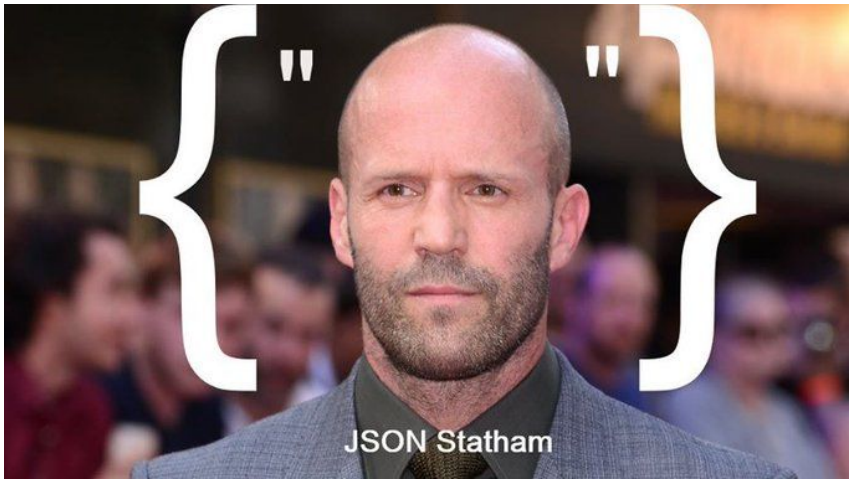
05

Tooling
SDK, K8ssandra

06

What's NEXT ?
v2 & revised arch

"Schemaless"



"What rules ?"

- You want to insert and retrieve any JSON documents efficiently
- Allow "schemaless" (Validation-less)
- Write to a single document is a single batch of statements
- Read from a single document is a single SELECT statement.
- Limit Tombstones with range deletes

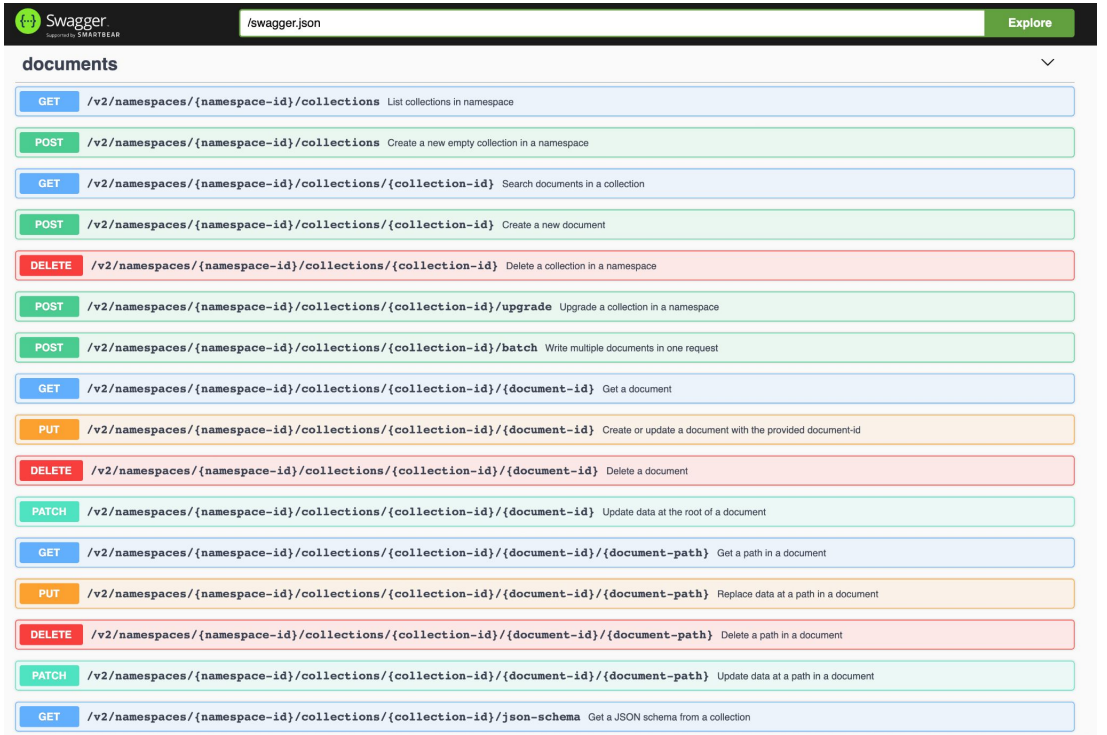
Different JSON shapes

```
{
  "comment": "found the Golden Fleece",
  "group": "The Argonauts",
  "properties": {
    "class": "myth",
    "nationality": "Greece"
   }
}
```

```
{
  "movies": [
    "The Expendables",
    "Fast and Furious"
  ],
  "properties": {
    "birthdate": 1967,
    "class": "person",
    "nationality": "English"
  },
  "surname": "Statham"
}
```

Stargate Document API

- Schemaless! (validation-less)
- Add any JSON document to a collection
- Uses "Document shredding" approach



Quick start: https://stargate.io/docs/stargate/1.0/quickstart/quick_start-document.html

Document Shredding (1 / 3)

```
create table <name> (
  key text,
  p0 text,
  … p[N] text,
  bool_value boolean,
  txt_value text, d
  bl_value double, leaf text
)
```

Document Shredding (2 / 3)

```
{"a": { "b": 1 }, "c": 2}
```

The document would be “shredded” into rows looking like this:

key	p0	p1	dbl_value
x	a	b	1
x	c	null	2

Document Shredding (3 / 3)

For data with an array, such as:

```
{"a": { "b": 1 }, "c": [{"d": 2}]}
```

there would be two rows, like so:

key	p0	p1	p2	dbl_value
x	a	b	null	1
x	c	[0]	d	2

Agenda

01

Stargate Data gateway
What, Why and How

02 {REST}

Exploring Apis
Rest DML and DDL

03

Exploring Apis
Document oriented

04

Exploring Apis
GraphQL and gRPC

05

Tooling
SDK, K8ssandra

06

What's NEXT ?
v2 & revised arch

History

GraphQL

- 2012 — Created by Facebook
  Used internally for mobile apps
- 2015 — Facebook give talk at ReactJs Conf
- 2015 — Facebook open sourced graphQL
- 2016 — Github announced move to GQL

Definition

GraphQL is an application programming interface (API) query language and server-side runtime that prioritises giving customers precisely the data they request.

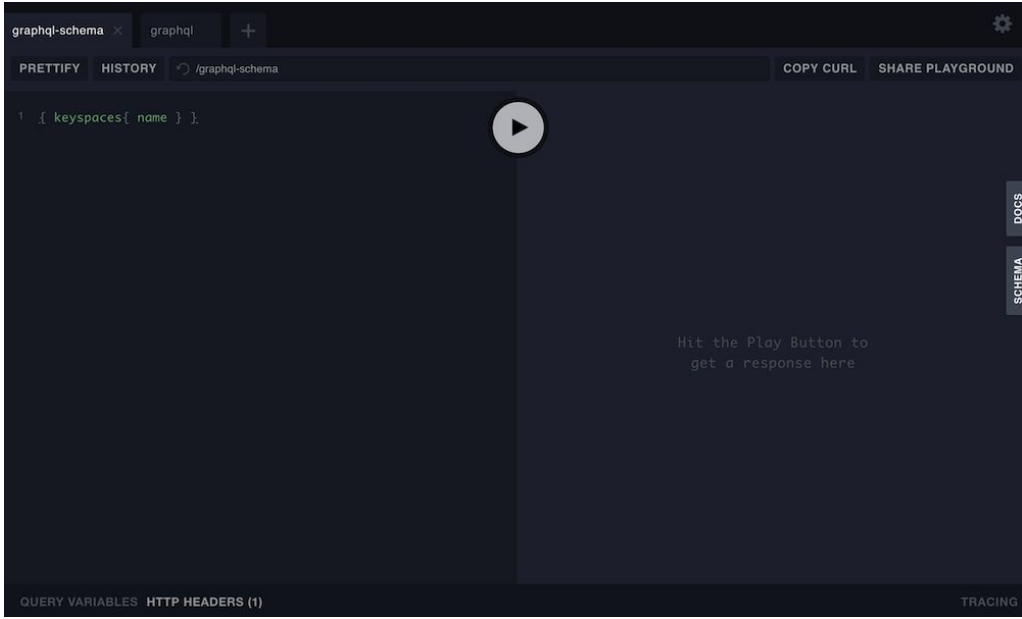
landscape.graphql.org/?category=graph-ql-adopter&grouping=category&style=borderless

Why?


GraphQL
Describe your data
type Project {
name: String
tagline: String
contributors: [User]
}
Ask for what you want
{
project(name: "GraphQL") {
tagline
}
}
Get predictable results
{
"project": {
"tagline": "A query language for APIs"
}
}

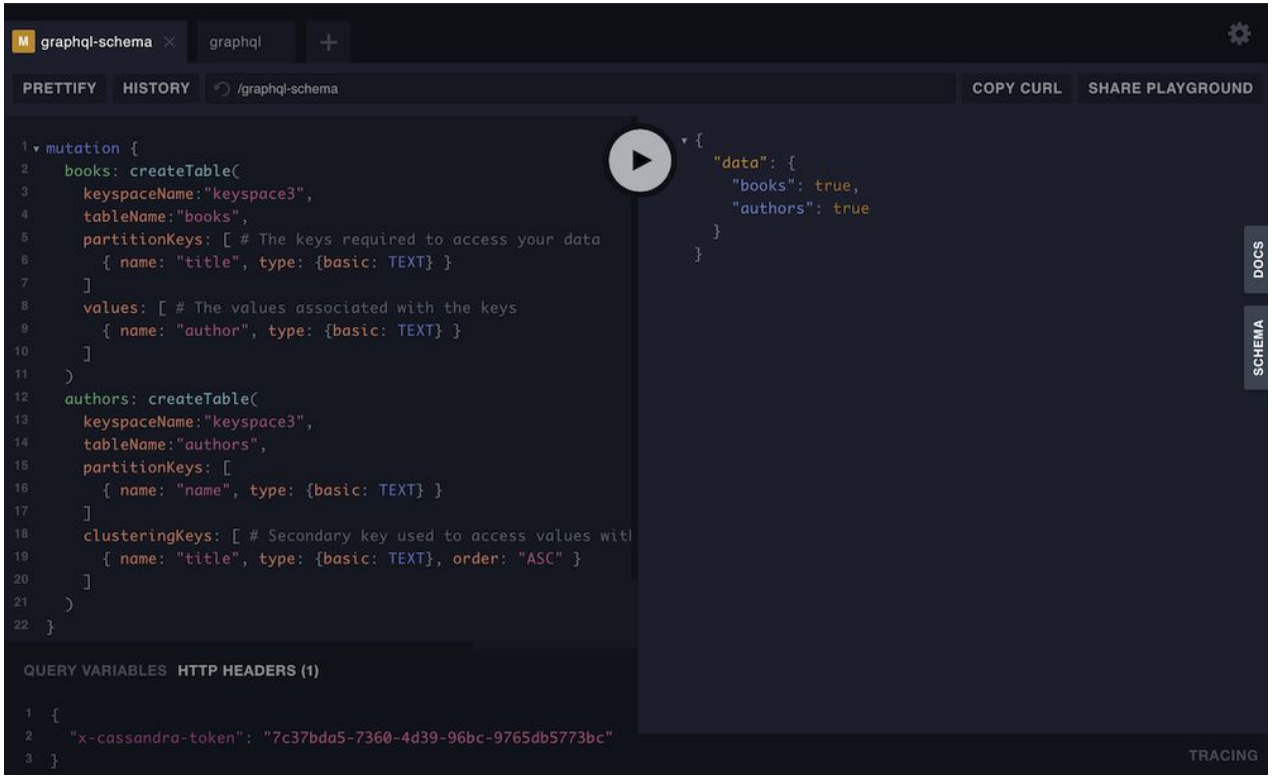
Stargate GraphQL API

- Expose existing CQL schema as a GraphQL endpoint
- Allows more fine-grained control of what fields returned
- Compatible with (Apollo) federation
- Easy experimentation in GraphQL playground

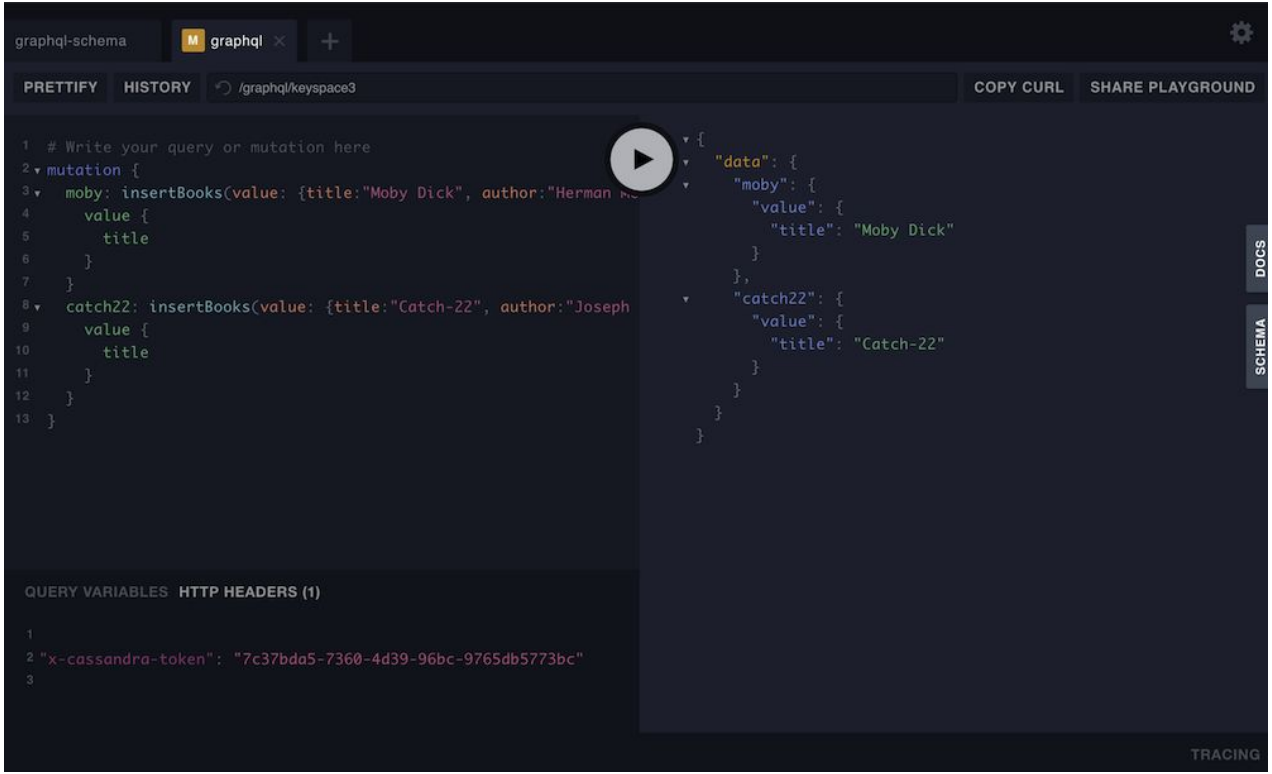


Quick start: stargate.io/docs/latest/quickstart/qs-graphql-cql-first.html

GraphQL "CQL-first"



GraphQL "CQL-first"



GraphQL "Schema-first"

```
type Book @key @cql_entity(name: "book") @cql_input {
  title: String! @cql_column(partitionKey: true, name: "book_title")
  isbn: String @cql_column(clusteringOrder: ASC)
  author: [String] @cql_index(name: "author_idx", target: VALUES)
}

type Address @cql_entity(target: UDT) @cql_input {
  street: String
  city: String
  state: String
  zipCode: String @cql_column(name: "zip_code")
}

#...

type Query {
   bookByTitleAndIsbn(title:String!, isbn:String): [Book]
   readerByNameAndUserid(name:String!, user_id:Uuid): [Reader]
}

type Mutation {
  insertBook(book:BookInput!): Book
  updateBook(book:BookInput!): Boolean @cql_update
  deleteBook(book:BookInput!): Boolean
  insertReader(reader:ReaderInput!): Reader
  deleteReader(reader:ReaderInput!): Boolean
  insertLibCollection(libColl: LibCollectionInput!): LibCollection
  deleteLibCollection(libColl: LibCollectionInput!): Boolean
}
```

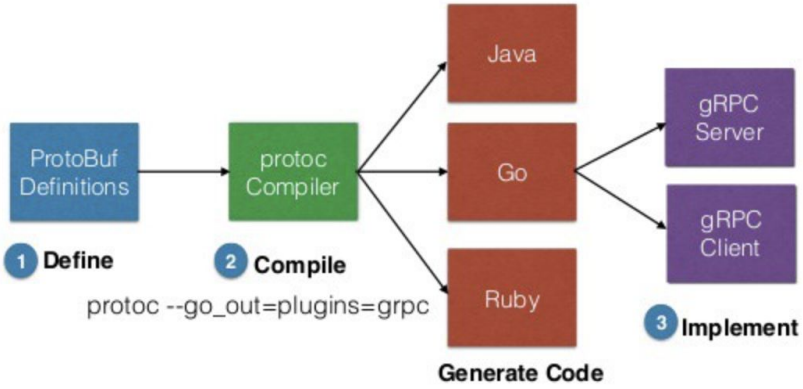
GraphQL "Schema-first"

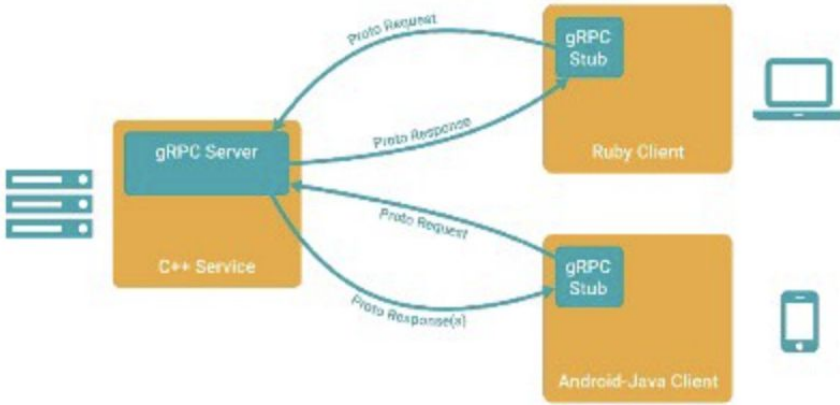
```
mutation {
  deploySchema(
    keyspace: "library"
    expectedVersion: "1da4f190-b7fd-11eb-8258-1ff1380eaff5"
    schema: """
    # Stargate does not require definition of fields in @key,
    # it uses the primary key
    type Book @key @cql_entity(name: "book") @cql_input {
      title: String! @cql_column(partitionKey: true, name: "book_title")
      isbn: String @cql_column(clusteringOrder: ASC)
      author: [String] @cql_index(name: "author_idx", target: VALUES)
    }

    ...
  }
```

GraphQL "Schema-first"

```
query fetchBook {
  book(title: "Native Son") {
    title
    author
  }
}

type Query {
bookGT(
  title: String
  isbn: String @cql_where(field: "isbn", predicate: GT)
): [Book]
}

type Query {
  readerCONTAINS(
    reviews: ReviewInput! @cql_where(field: "reviews", predicate: CONTAINS)
  ): [Reader]
}

type Query {
  booksIn(
    title: [String!] @cql_where(field: "title", predicate: IN)
  ): [Book]
}
```

gRPC





gRPC

@hostirosti @grpcio

Comparing Stargate APIs

Drivers | **Open API**

Cassandra Query Language

SQL like Table Model

Structured Data

Key-Value Data

Strong Types

Minimal query overhead

gRPC

gRPC

Structured Data (CQL)

Lighter weight

Native driver alternative

Low query overhead

GraphQL

GraphQL

Hierarchy of types and fields

Structured Data

Key-Value Data

Low query overhead

{REST}

REST

Row based

Structured Data

Key-Value Data

Weaker Types

High query overhead

JSON

Document

JSON Documents

Semi-Structured Data

Weaker Types

High query overhead

More Performant ⟷ **More Flexible**

Agenda

01

Stargate Data gateway
What, Why and How

02 {REST}

Exploring Apis
Rest DML and DDL

03

Exploring Apis
Document oriented

04

Exploring Apis
GraphQL and gRPC

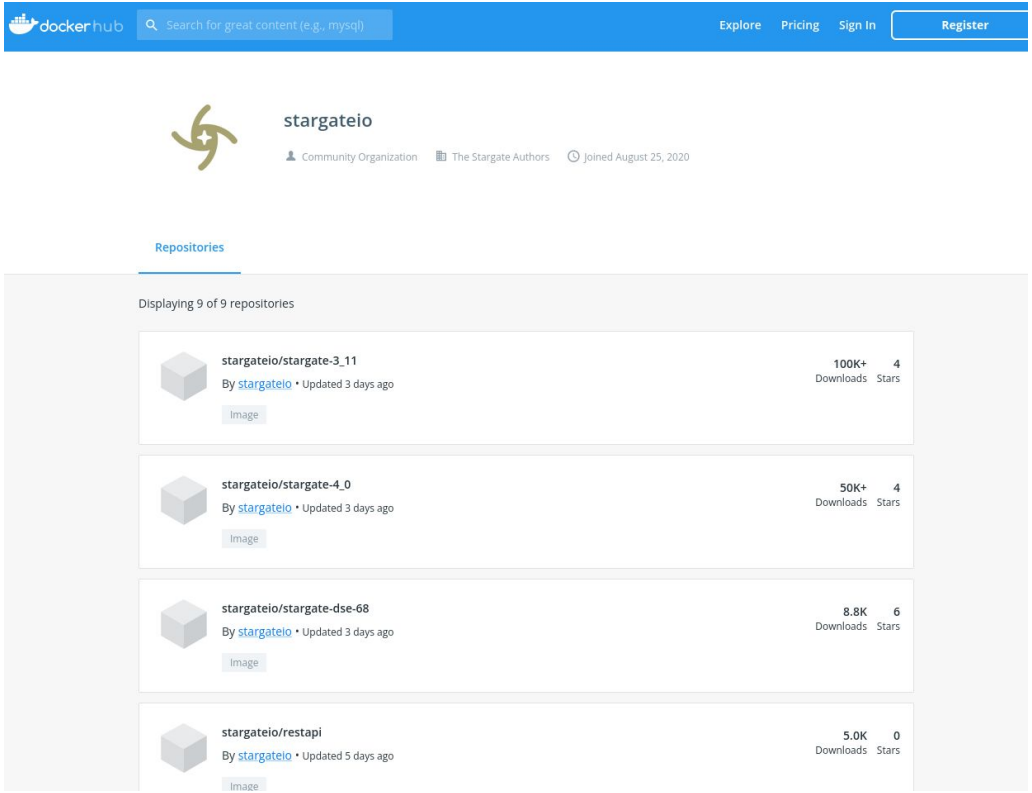
05

Tooling
SDK, K8ssandra

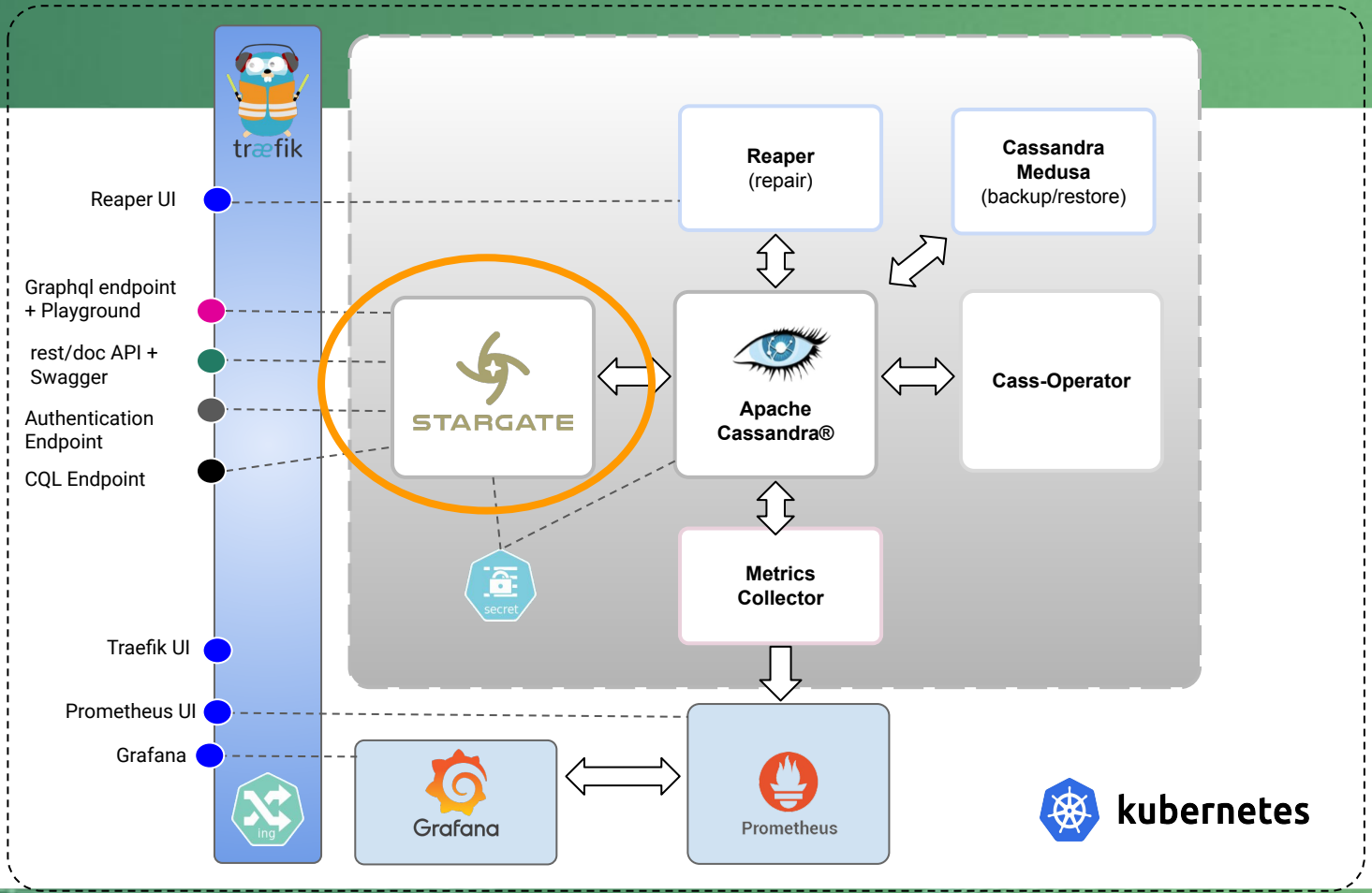
06

What's NEXT ?
v2 & revised arch

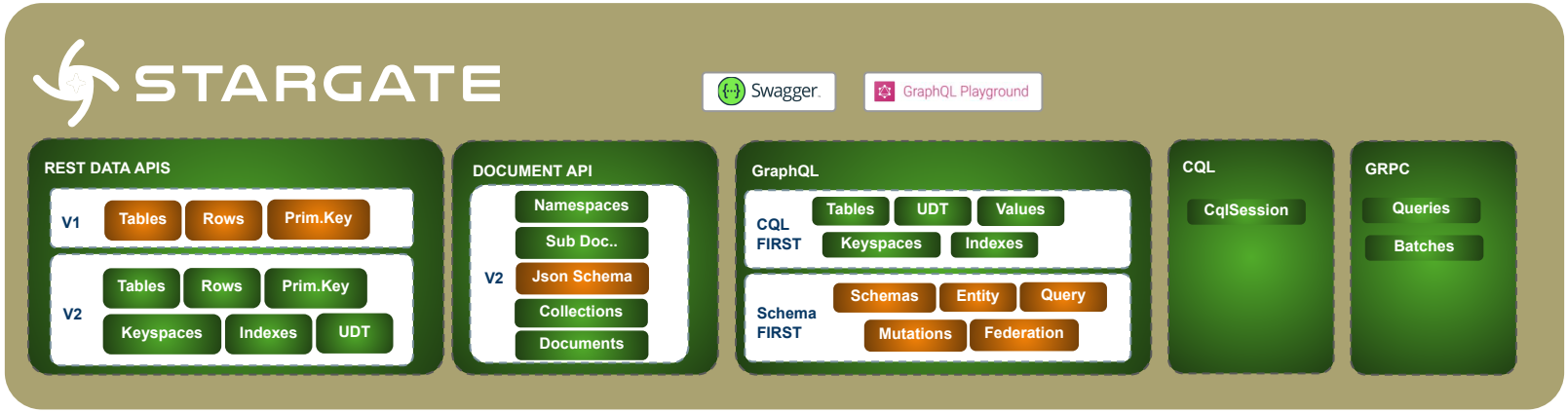
Docker


docker hub
Search for great content (e.g., mysql)
Explore
Pricing
Sign In
Register
stargateio
Community Organization
The Stargate Authors
Joined August 25, 2020
Repositories
Displaying 9 of 9 repositories
stargateio/stargate-3_11
By stargateio • Updated 3 days ago
Image
100K+ Downloads
4 Stars
stargateio/stargate-4_0
By stargateio • Updated 3 days ago
Image
50K+ Downloads
4 Stars
stargateio/stargate-dse-68
By stargateio • Updated 3 days ago
Image
8.8K Downloads
6 Stars
stargateio/restapi
By stargateio • Updated 5 days ago
Image
5.0K Downloads
0 Stars

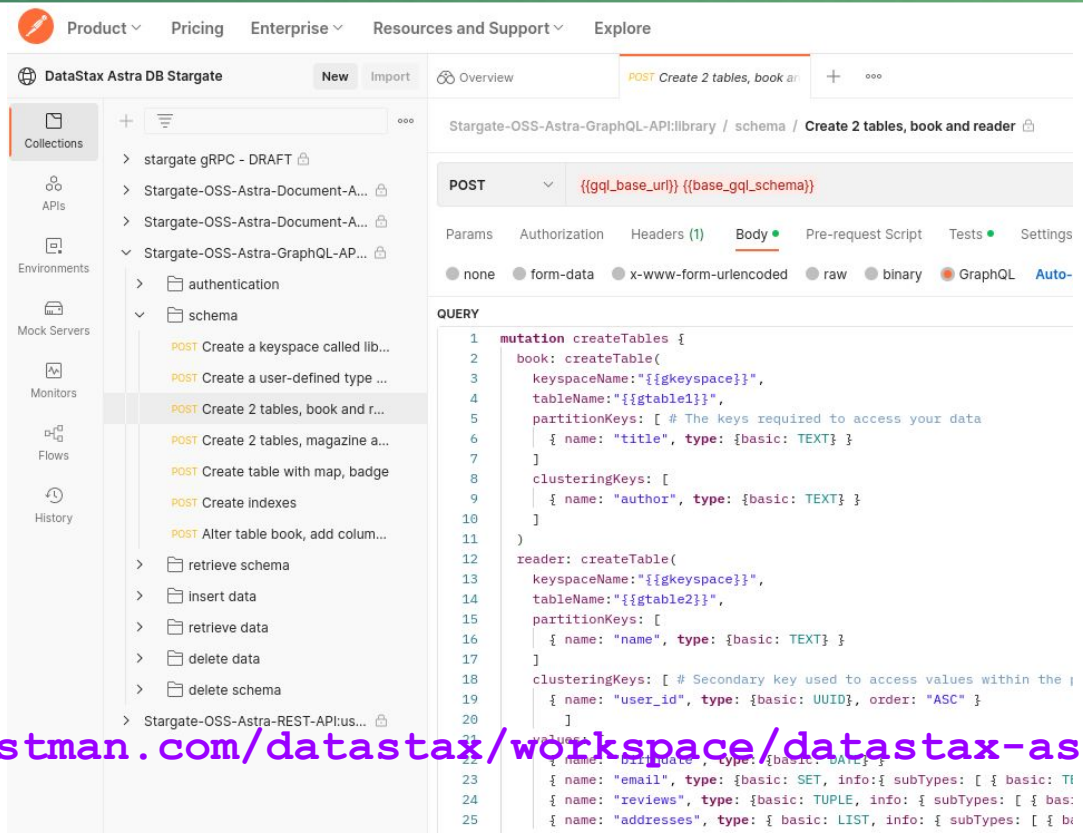
K8SSANDRA


traefik
Reaper UI
Graphql endpoint
+ Playground
rest/doc API +
Swagger
Authentication
Endpoint
CQL Endpoint
Traefik UI
Prometheus UI
Grafana
ing
STARGATE
secret
Reaper
(repair)
Cassandra
Medusa
(backup/restore)
Apache
Cassandra®
Cass-Operator
Metrics
Collector
Grafana
Prometheus
kubernetes

SDK (Java, Javascript, Python)



Postman



https://www.postman.com/datastax/workspace/datastax-astra-db-stargate

Agenda

01

Stargate Data gateway
What, Why and How

02 {REST}

Exploring Apis
Rest DML and DDL

03

Exploring Apis
Document oriented

04

Exploring Apis
GraphQL and gRPC

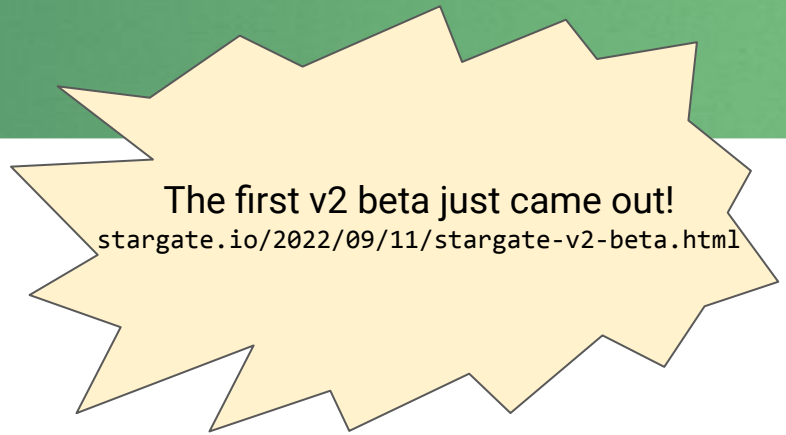
05

Tooling
SDK, K8ssandra

06

What's NEXT ?
v2 & revised arch

Next: upcoming Stargate V2

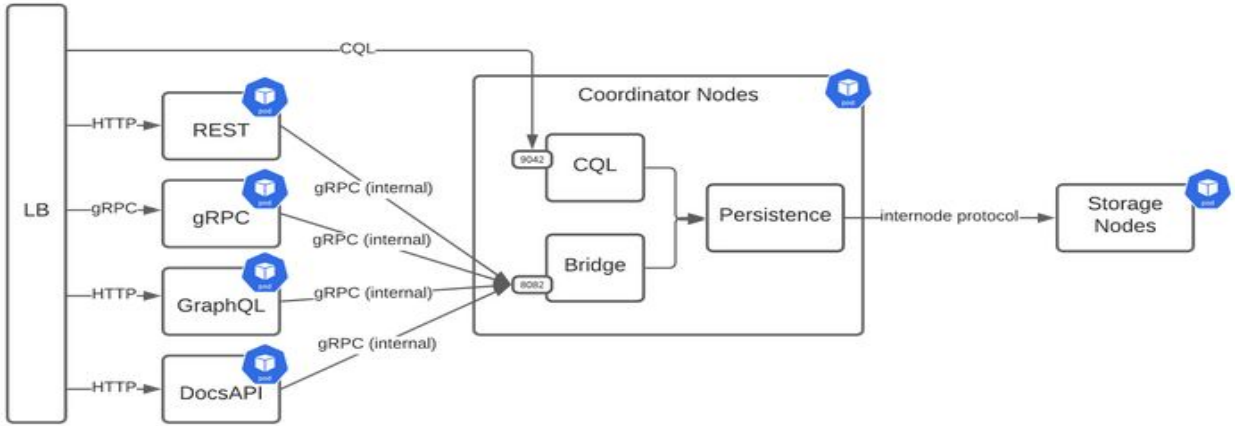


Each service in its pod	can scale them independently
Use gRPC internally	a new "bridge" introduced
From Spring Boot to Quarkus	native compilation, faster binary
Document API revision	also long-term plans for new Doc API features

V2, redesigned architecture

Meet the winning design

After weighing the pros and cons of the other potential designs, we finally settled on what we believe is the best option for Stargate v2. Take a look at the diagram below.



Source: stargate.io/2021/11/02/introducing-the-design-for-stargate-v2.html

Find out more & get involved

Stargate Discord	discord.com/invite/5gY8GDB
Website	stargate.io
Github	github.com/stargate/stargate
Twitter	@stargateio
Linkedin	linkedin.com/groups/9091327

Apache CASSANDRA

Thank you!

Sponsored by DataStax