FRODO BAGGINS
BUT WITH FEDERATED GRAPHQL

WAIT, WHAT?

All right, then. Keep your secrets.

JUKBEN

JAKUB BENEŠ

STAFF ENGINEER AT PRODUCTBOARD

- jukben.codes
- @jukben
- AMA later

Agenda

- » GraphQL and why we have chosen it
- » GraphQL Federation in nutshell
- » What we did and how we did it

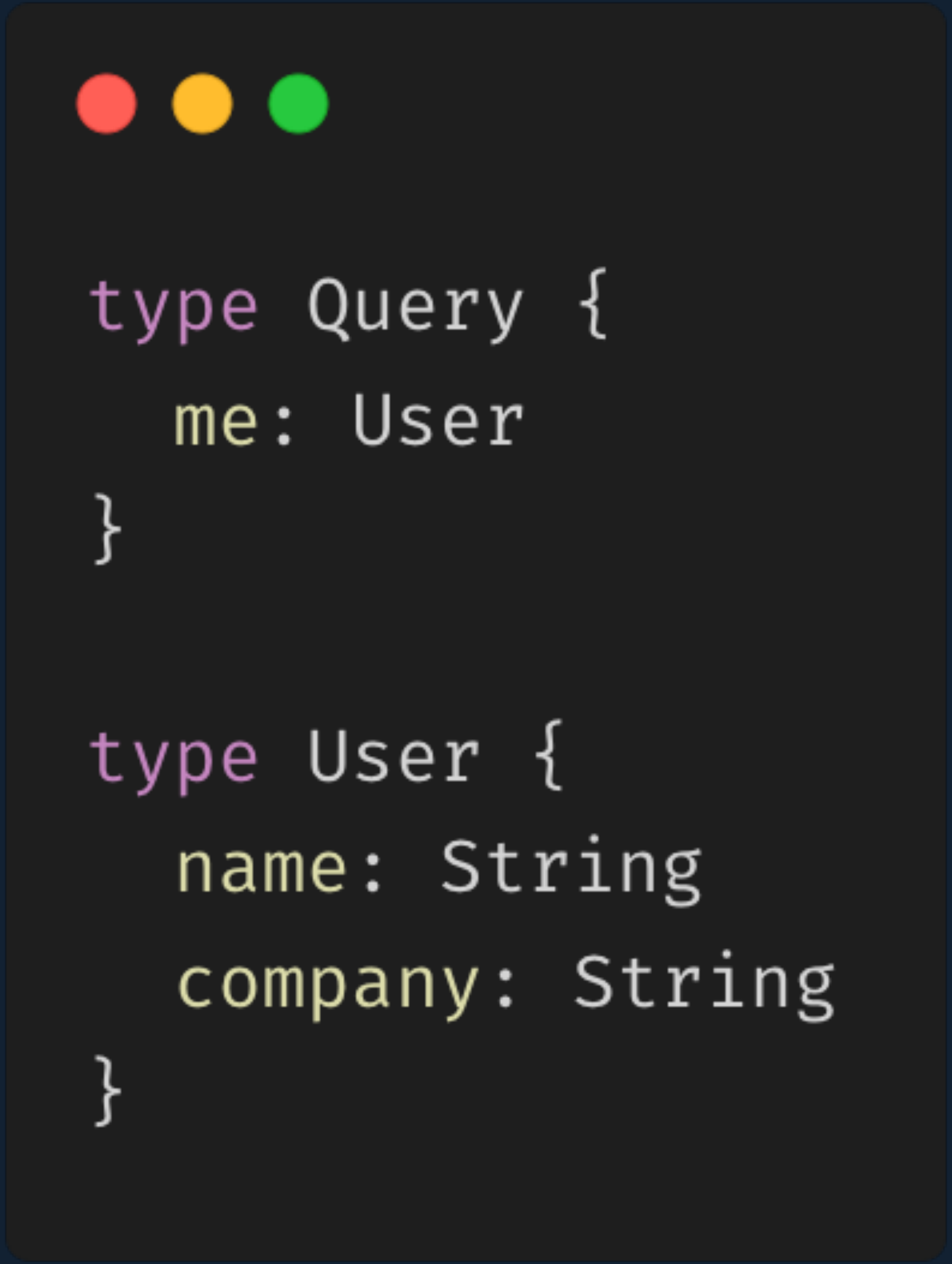
WHO KNOWS GRAPHQL?

GRAPHQL IN 180 SECONDS

QUERY LANGUAGE

Strongly Typed

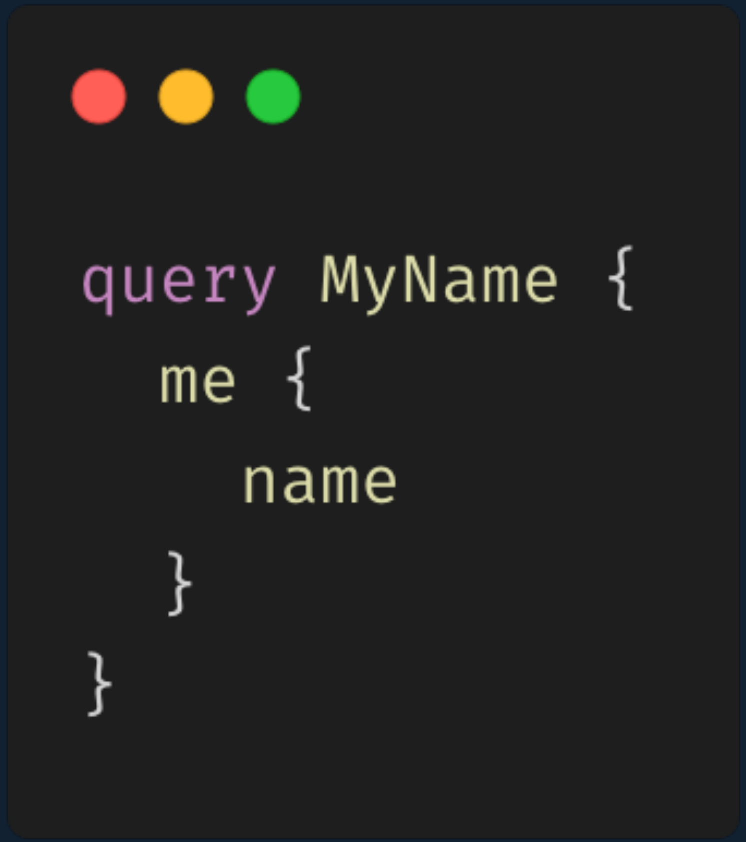
- » Nullable by default
- » Default scalars: Int, Float, String, Boolean, ID
- » Enums, Interfaces, Directives
- » https://spec.graphql.org/



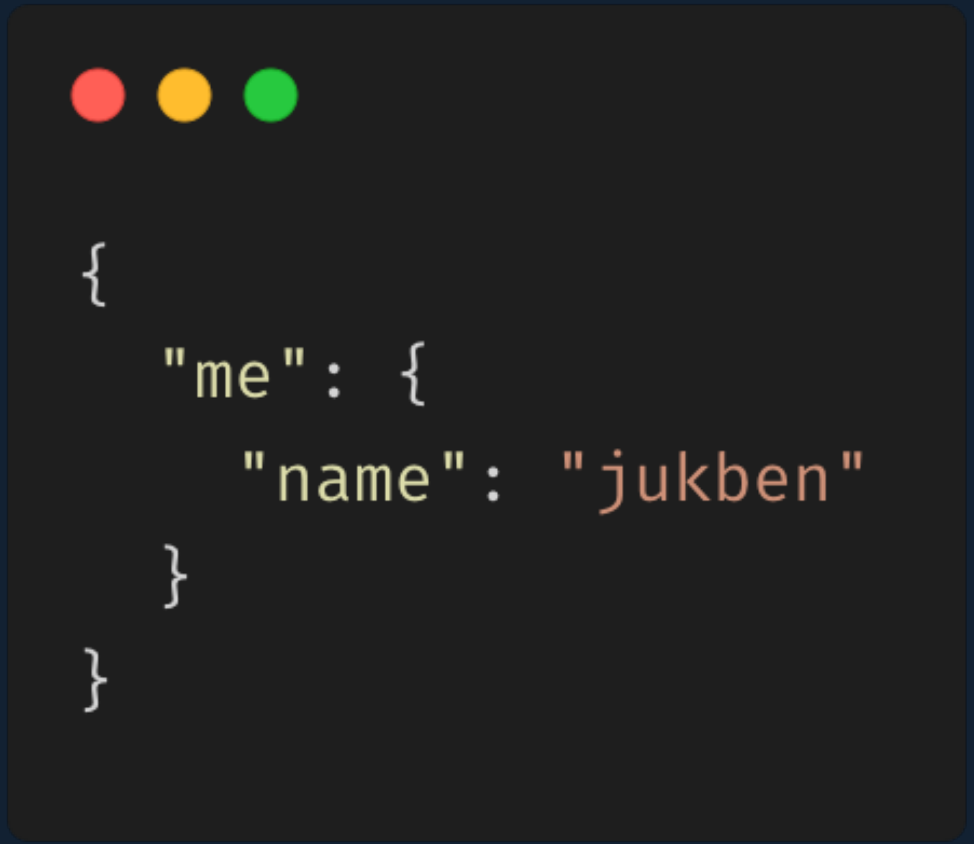
```
type Query {
  me: User
}

type User {
  name: String
  company: String
}
```

YOU GET WHAT YOU ASK FOR

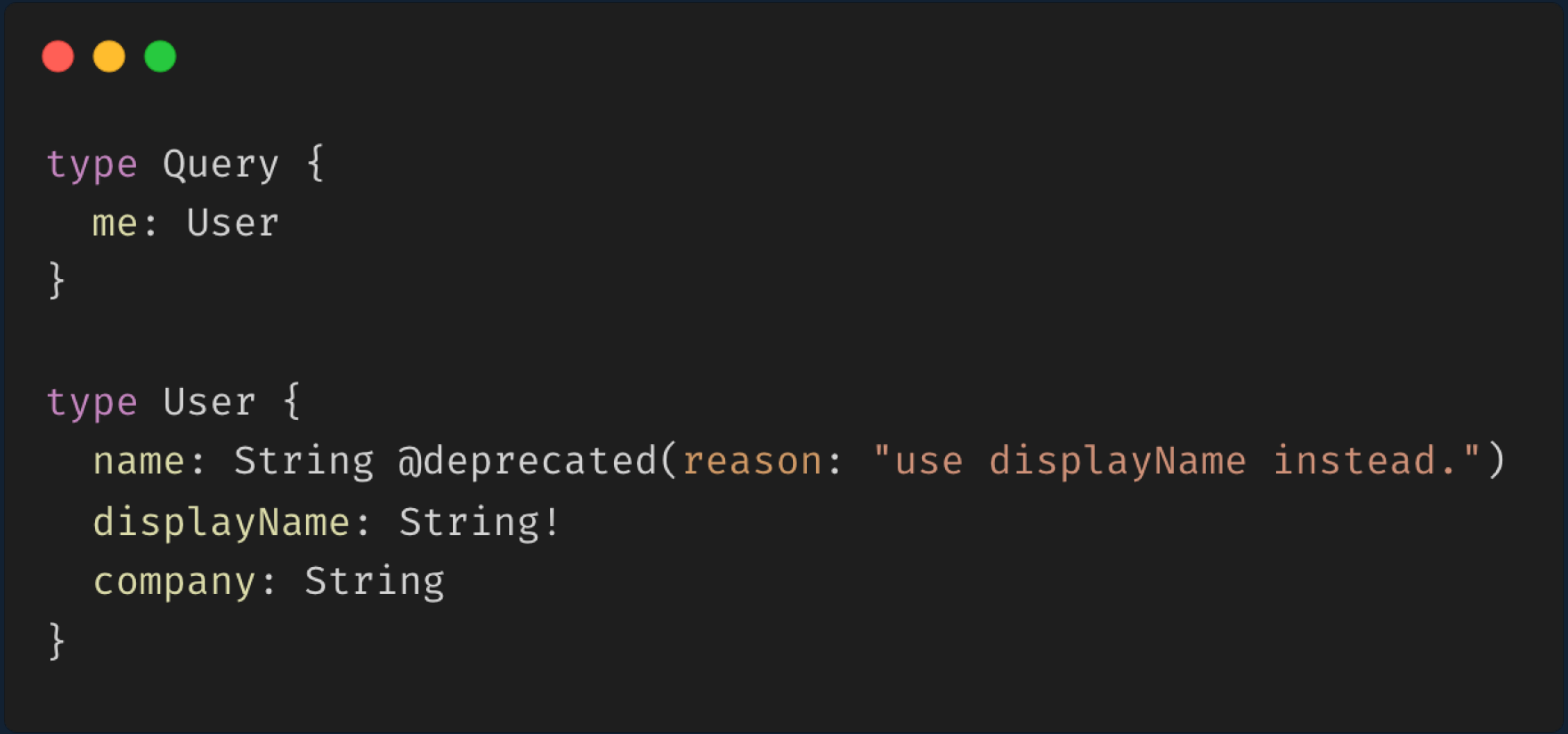


```graphql
query MyName {
  me {
    name
  }
}
```



```json
{
  "me": {
    "name": "jukben"
  }
}
```

NON-BREAKING CHANGES



```graphql
type Query {
  me: User
}

type User {
  name: String @deprecated(reason: "use displayName instead.")
  displayName: String!
  company: String
}
```

DX

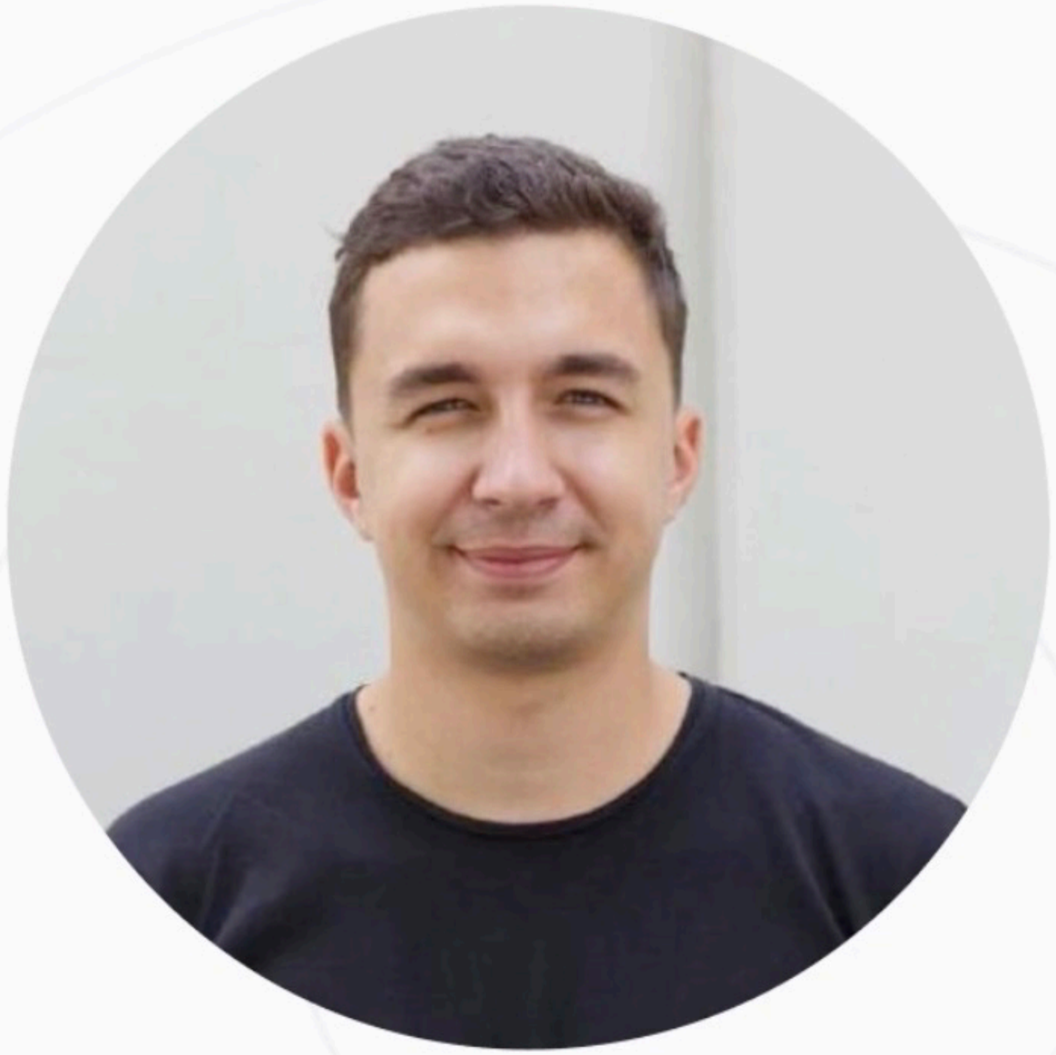
PRODUCTBOARD'S REASONS

S

» YouTube

productboard

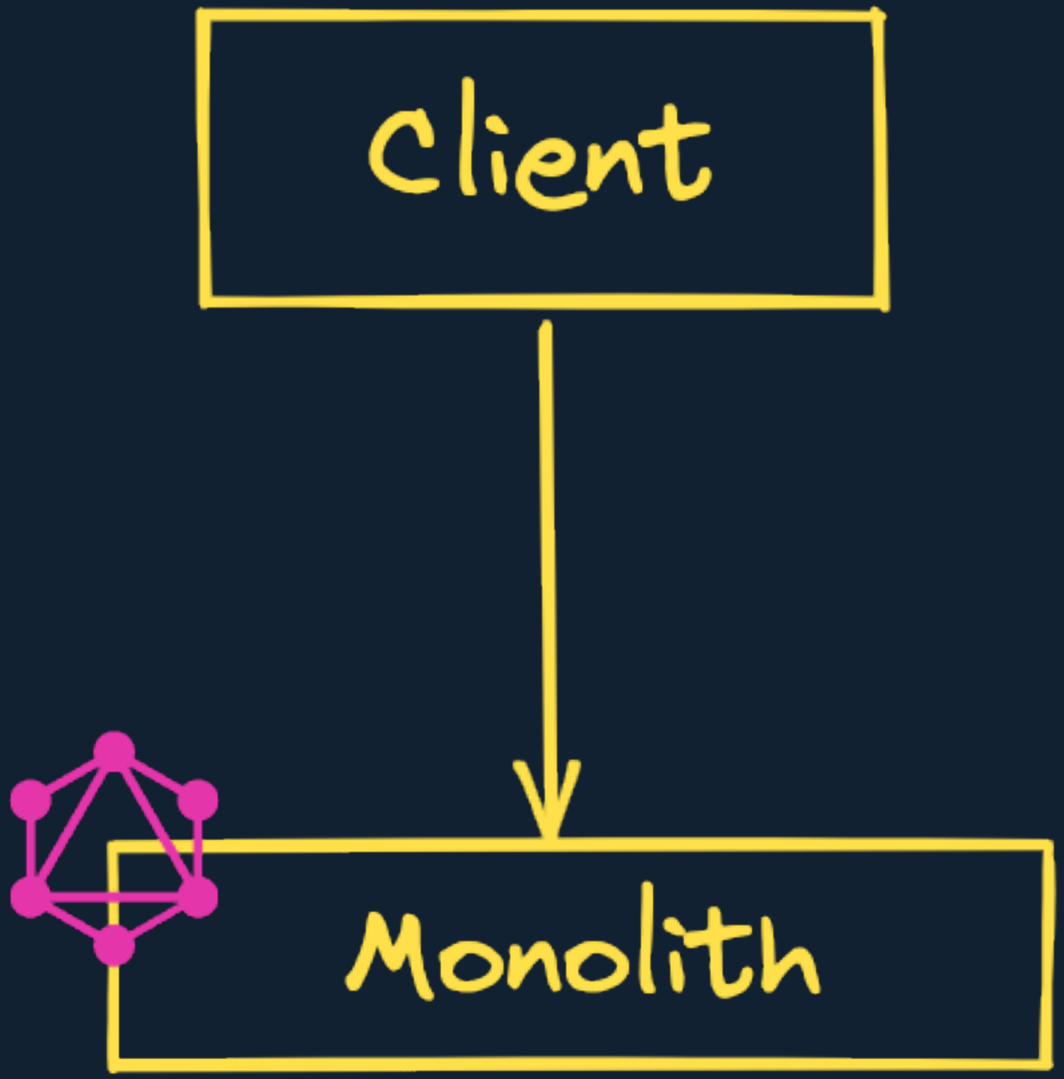
Loading 40 MB of JSON on initial load

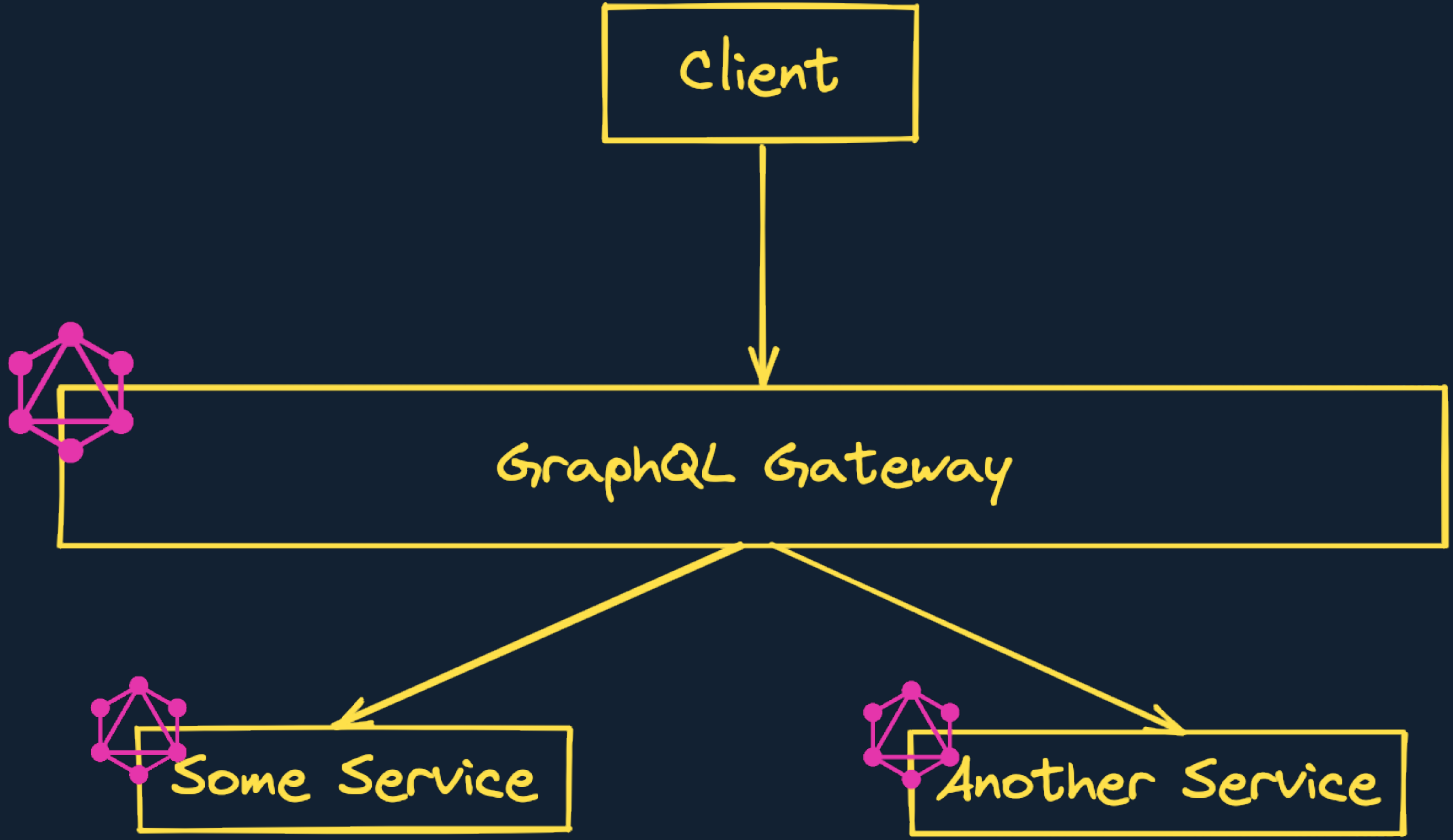


Lukáš Huvar

Senior Frontend Engineer at Productboard

FEDERATED GRAPHQL


Client
Monolith

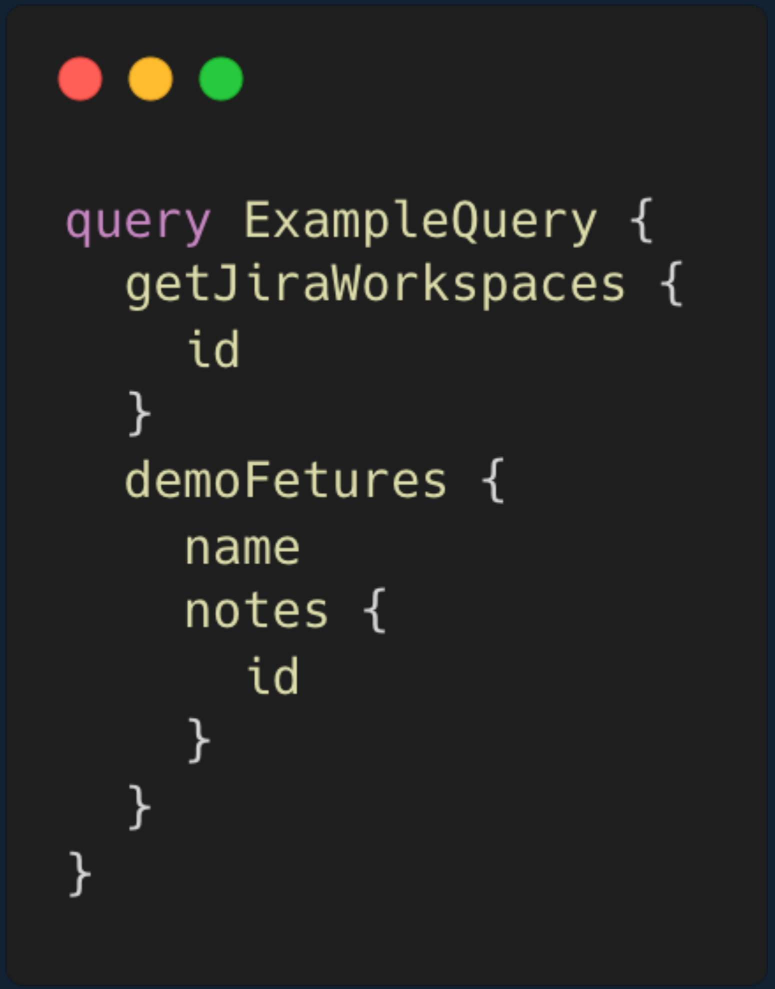

Client
GraphQL Gateway
Some Service
Another Service

Apollo Federation

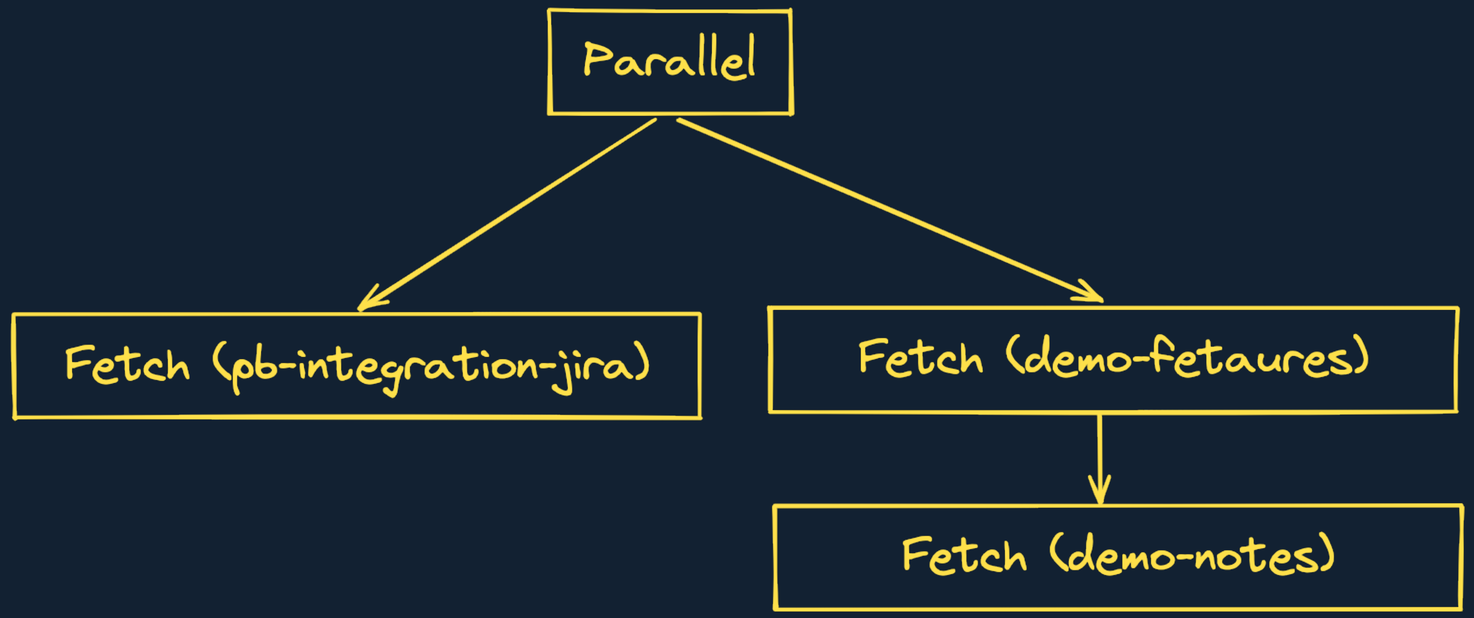
» https://www.apollographql.com/docs/federation/federation-spec

» Introduces a few of directives like: @key, @sharable to instruct Gateway how to do query planning.

QUERY PLANNING

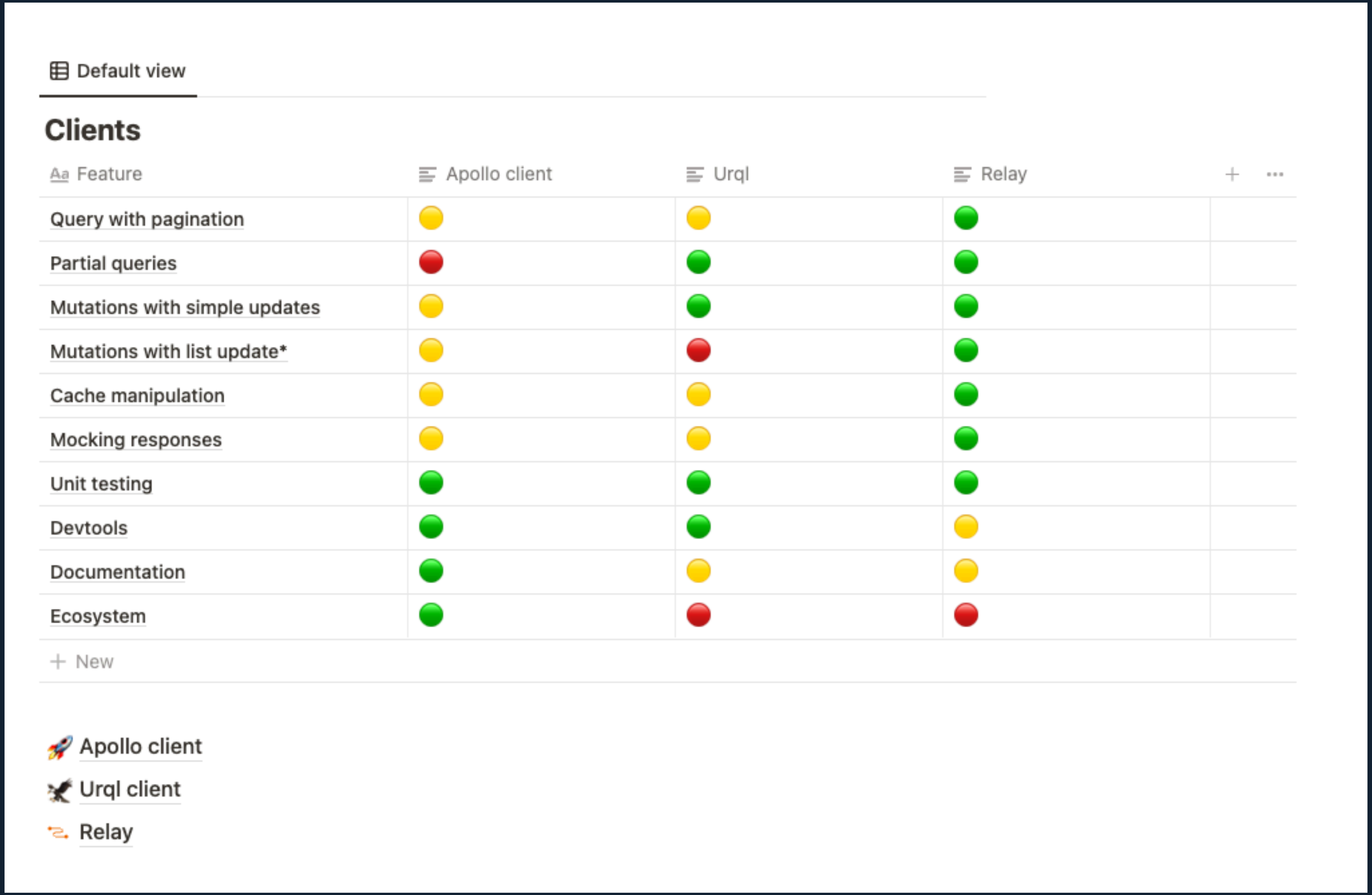


```
query ExampleQuery {
  getJiraWorkspaces {
    id
  }
  demoFetures {
    name
    notes {
      id
    }
  }
}
```



OUR IMPLEMENTATION

FRONTEND



Default view

Clients

Feature	Apollo client	Urql	Relay
Query with pagination	🟡	🟡	🟢
Partial queries	🔴	🟢	🟢
Mutations with simple updates	🟡	🟢	🟢
Mutations with list update*	🟡	🔴	🟢
Cache manipulation	🟡	🟡	🟢
Mocking responses	🟡	🟡	🟢
Unit testing	🟢	🟢	🟢
Devtools	🟢	🟢	🟡
Documentation	🟢	🟡	🟡
Ecosystem	🟢	🔴	🔴

+ New

🚀 Apollo client

Urql client

Relay

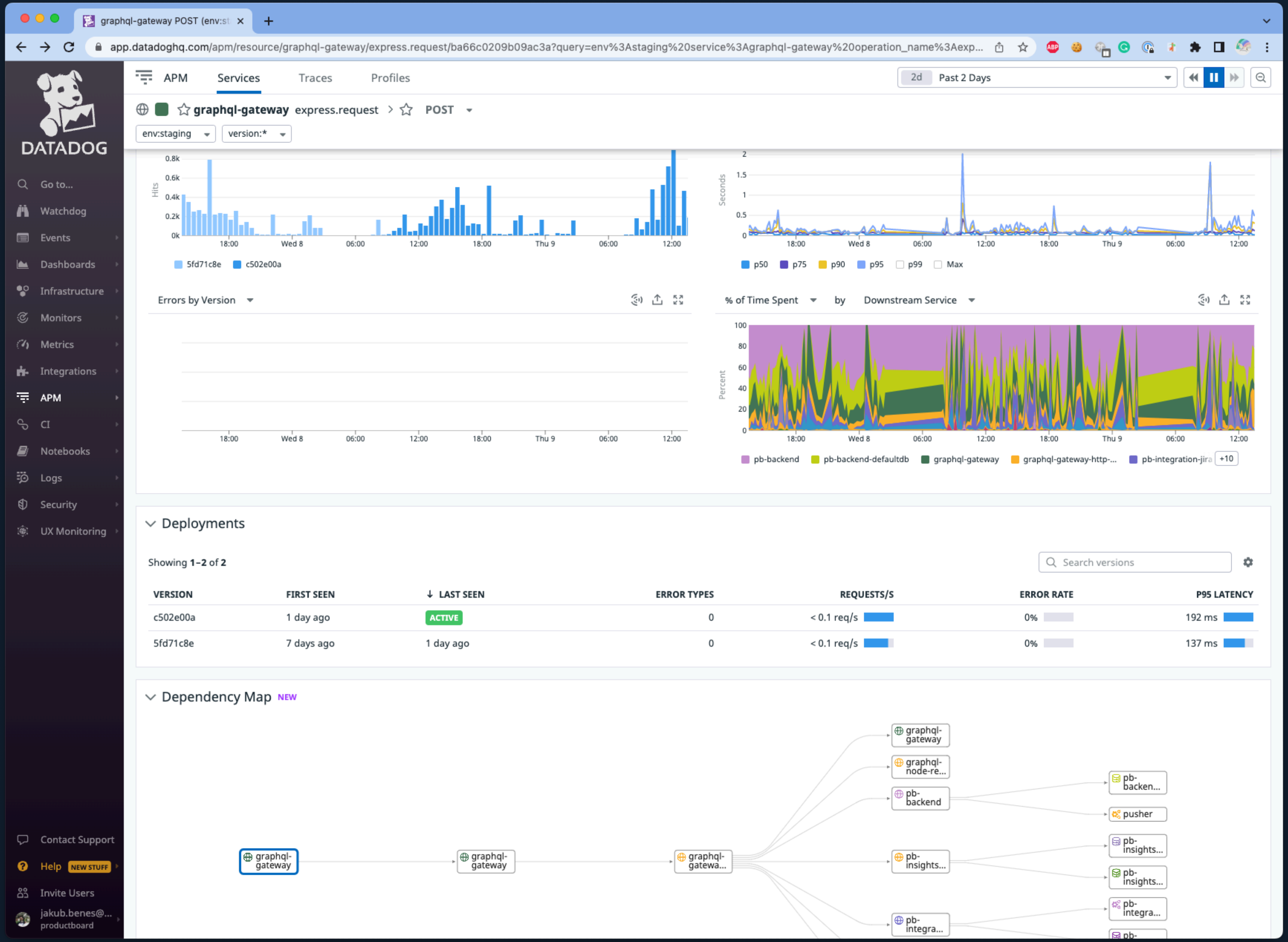
Backend

- https://netflix.github.io/dgs/ built on top of Spring Boot
- https://graphql-ruby.org
- Server needs to follow Federation Spec and Global ID and Pagination spec

Federation

- decided to go with managed Federation by Apollo Studio
- schema registry with schema checking

OBSERVABILITY: SERVICE

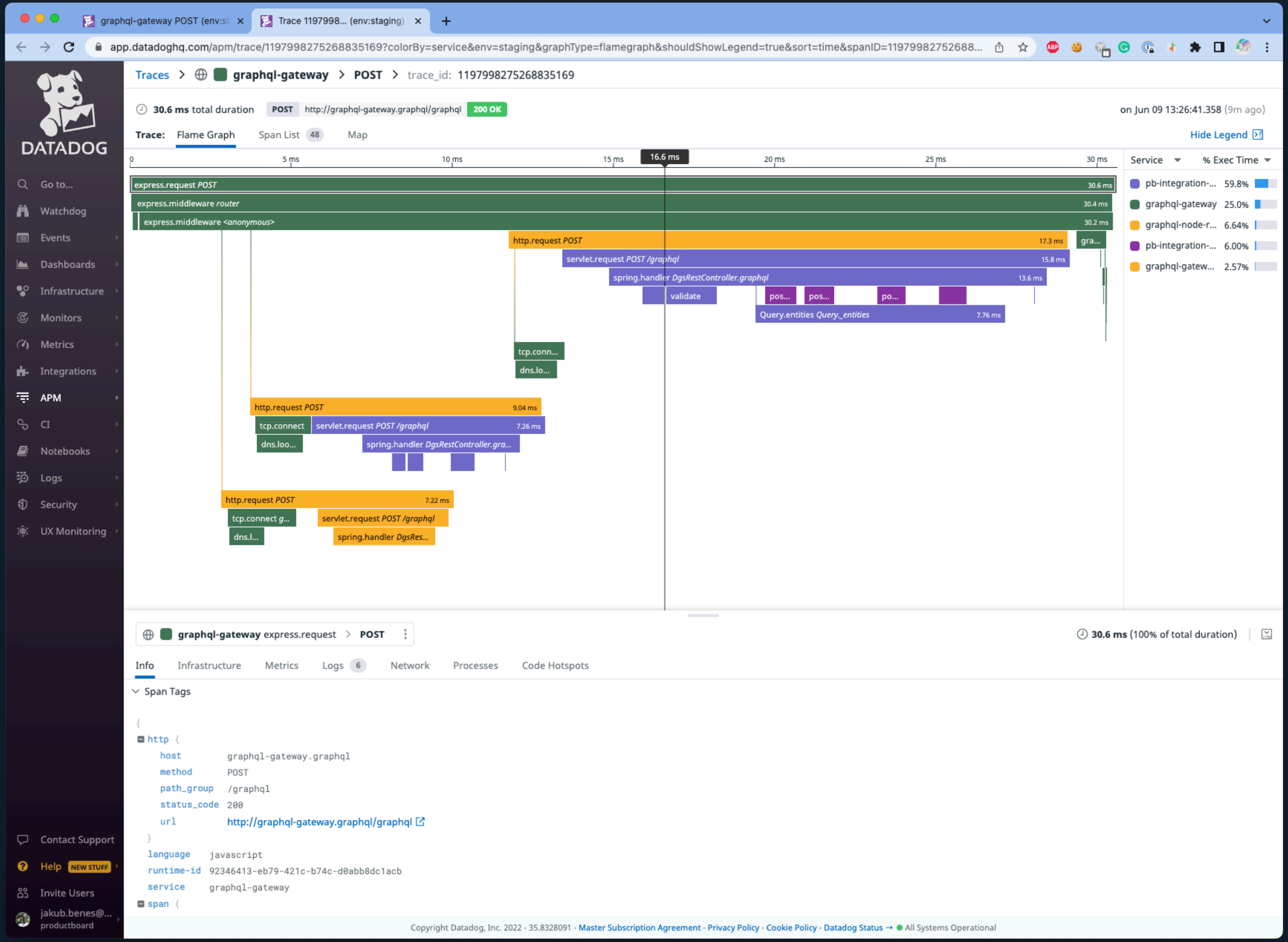


Deployments

Showing 1–2 of 2

VERSION	FIRST SEEN	↓ LAST SEEN	ERROR TYPES	REQUESTS/S	ERROR RATE	P95 LATENCY
c502e00a	1 day ago	ACTIVE	0	< 0.1 req/s	0%	192 ms
5fd71c8e	7 days ago	1 day ago	0	< 0.1 req/s	0%	137 ms

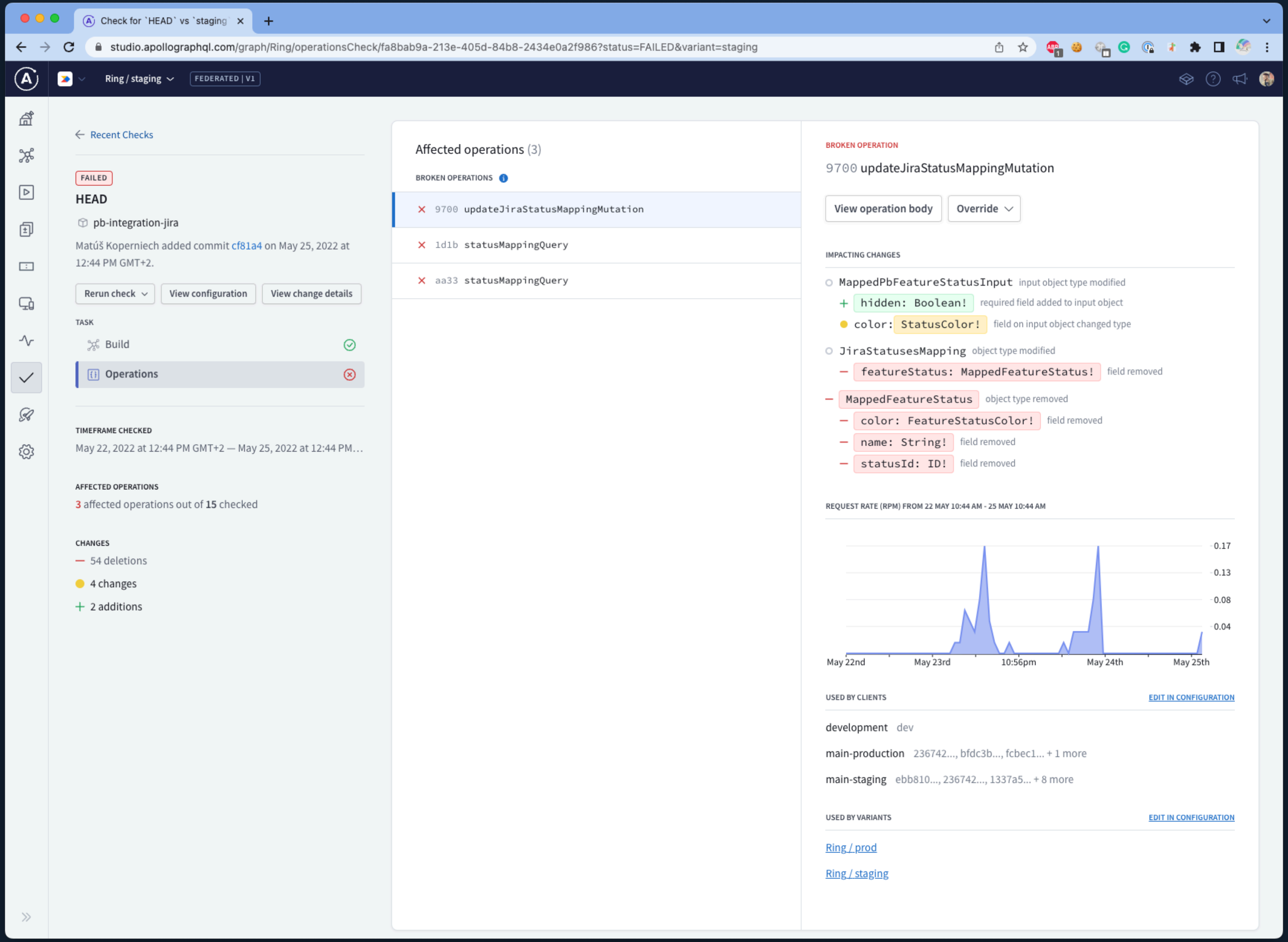
OBSERVABILITY: APM


graphql-gateway POST (env:st
Trace 1197998... (env:staging)
app.datadoghq.com/apm/trace/1197998275268835169?colorBy=service&env=staging&graphType=flamegraph&shouldShowLegend=true&sort=time&spanID=11979982752688...
DATADOG
Go to...
Watchdog
Events
Dashboards
Infrastructure
Monitors
Metrics
Integrations
APM
CI
Notebooks
Logs
Security
UX Monitoring
Contact Support
Help NEW STUFF
Invite Users
jakub.benes@...
productboard
Traces
graphql-gateway
POST
trace_id: 1197998275268835169
30.6 ms total duration
POST http://graphql-gateway.graphql/graphql 200 OK
on Jun 09 13:26:41.358 (9m ago)
Trace: Flame Graph
Span List 48
Map
Hide Legend
0
5 ms
10 ms
15 ms
16.6 ms
20 ms
25 ms
30 ms
express.request POST 30.6 ms
express.middleware router 30.4 ms
express.middleware <anonymous> 30.2 ms
http.request POST 17.3 ms
gra...
servlet.request POST /graphql 15.8 ms
spring.handler DgsRestController.graphql 13.6 ms
validate
pos...
pos...
po...
Query.entities Query._entities 7.76 ms
tcp.conn...
dns.lo...
http.request POST 9.04 ms
tcp.connect
servlet.request POST /graphql 7.26 ms
dns.loo...
spring.handler DgsRestController.gra...
http.request POST 7.22 ms
tcp.connect g...
servlet.request POST /graphql
dns.l...
spring.handler DgsRes...
Service
% Exec Time
pb-integration-... 59.8%
graphql-gateway 25.0%
graphql-node-r... 6.64%
pb-integration-... 6.00%
graphql-gatew... 2.57%
graphql-gateway express.request
POST
30.6 ms (100% of total duration)
Info
Infrastructure
Metrics
Logs 6
Network
Processes
Code Hotspots
Span Tags

```
{
http {
  host        graphql-gateway.graphql
  method      POST
  path_group  /graphql
  status_code 200
  url         http://graphql-gateway.graphql/graphql
}
language   javascript
runtime-id 92346413-eb79-421c-b74c-d0abb8dc1acb
service    graphql-gateway
span {
```

Copyright Datadog, Inc. 2022 - 35.8328091 - Master Subscription Agreement - Privacy Policy - Cookie Policy - Datadog Status → ● All Systems Operational

OBSERVABILITY: BREAKING CHANGES


Check for `HEAD` vs `staging`
studio.apollographql.com/graph/Ring/operationsCheck/fa8bab9a-213e-405d-84b8-2434e0a2f986?status=FAILED&variant=staging
Ring / staging
FEDERATED | V1
Recent Checks
Rerun check
View configuration
View change details
View operation body
Override
EDIT IN CONFIGURATION

FAILED

HEAD

pb-integration-jira

Matúš Koperniech added commit cf81a4 on May 25, 2022 at 12:44 PM GMT+2.

TASK

Build

Operations

TIMEFRAME CHECKED

May 22, 2022 at 12:44 PM GMT+2 — May 25, 2022 at 12:44 PM…

AFFECTED OPERATIONS

3 affected operations out of 15 checked

CHANGES

— 54 deletions

● 4 changes

+ 2 additions

Affected operations (3)

BROKEN OPERATIONS

```
× 9700 updateJiraStatusMappingMutation
× 1d1b statusMappingQuery
× aa33 statusMappingQuery
```

BROKEN OPERATION

9700 updateJiraStatusMappingMutation

IMPACTING CHANGES

```
○ MappedPbFeatureStatusInput  input object type modified
  + hidden: Boolean!  required field added to input object
  ● color: StatusColor!  field on input object changed type
○ JiraStatusesMapping  object type modified
  — featureStatus: MappedFeatureStatus!  field removed
— MappedFeatureStatus  object type removed
  — color: FeatureStatusColor!  field removed
  — name: String!  field removed
  — statusId: ID!  field removed
```

REQUEST RATE (RPM) FROM 22 MAY 10:44 AM - 25 MAY 10:44 AM

USED BY CLIENTS

development	dev
main-production	236742..., bfdc3b..., fcbec1... + 1 more
main-staging	ebb810..., 236742..., 1337a5... + 8 more

USED BY VARIANTS

Ring / prod

Ring / staging

MORPHING

AKA HOW TO ROLL IT OUT

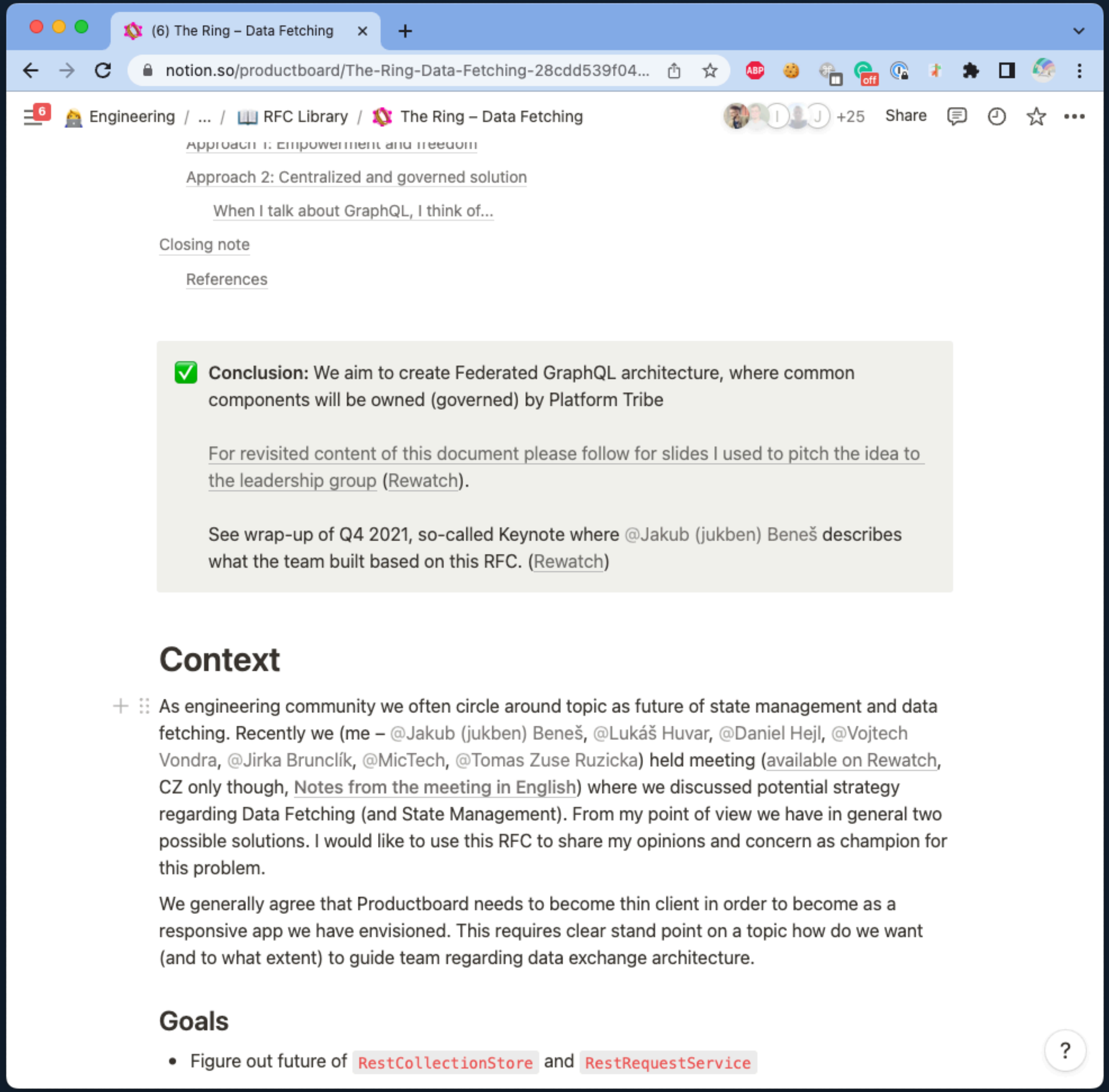
THE RING

Data Fetching

FELLOWSHIP

@huvik
@lukas_krecan
@balvajs

PROBLEM AND SOLUTION PROPOSAL


(6) The Ring – Data Fetching

notion.so/productboard/The-Ring-Data-Fetching-28cdd539f04...

Engineering / ... / RFC Library / The Ring – Data Fetching

Approach 1: Empowerment and freedom

Approach 2: Centralized and governed solution

When I talk about GraphQL, I think of...

Closing note

References

✅ **Conclusion:** We aim to create Federated GraphQL architecture, where common components will be owned (governed) by Platform Tribe

For revisited content of this document please follow for slides I used to pitch the idea to the leadership group (Rewatch).

See wrap-up of Q4 2021, so-called Keynote where @Jakub (jukben) Beneš describes what the team built based on this RFC. (Rewatch)

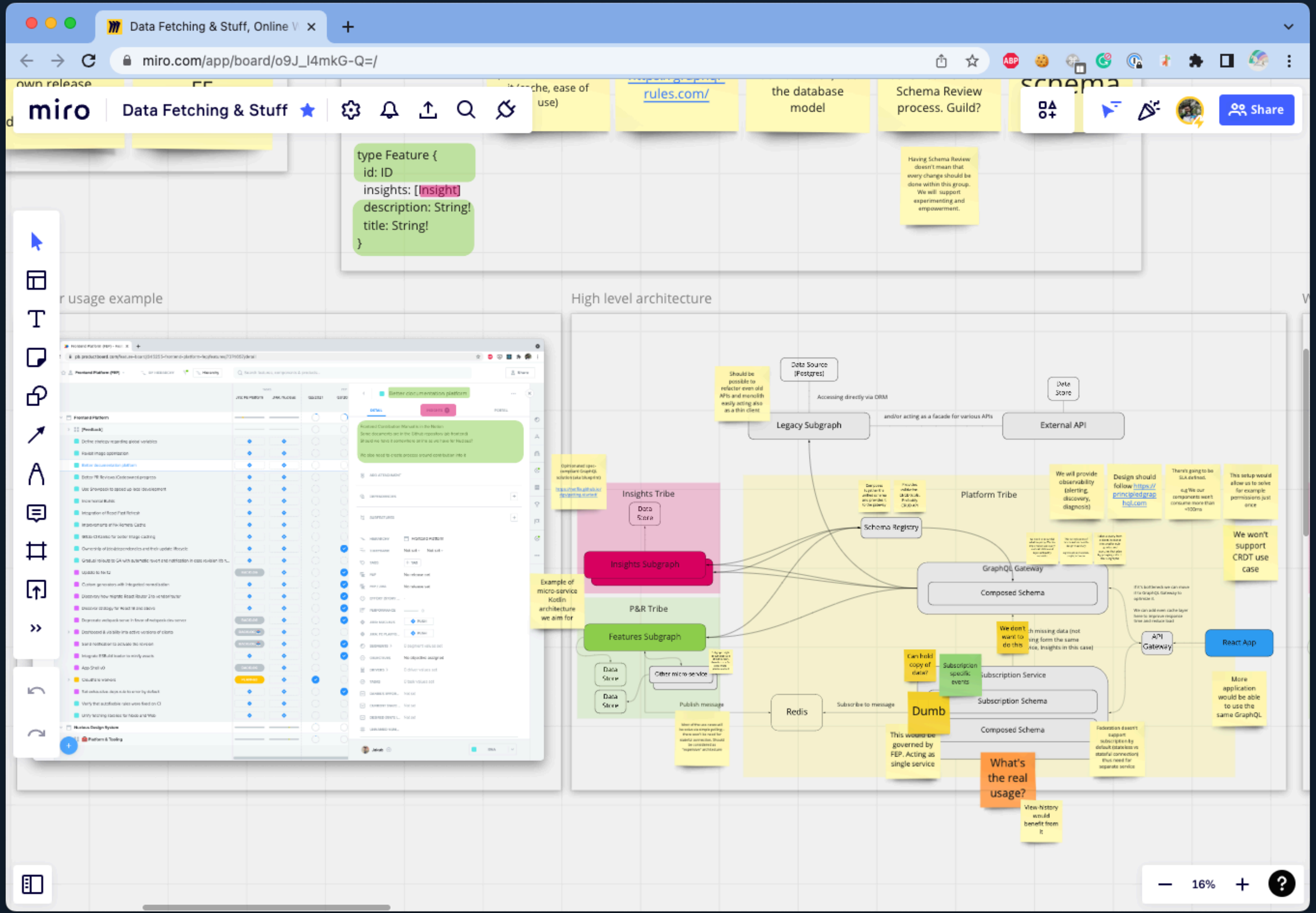
Context

As engineering community we often circle around topic as future of state management and data fetching. Recently we (me – @Jakub (jukben) Beneš, @Lukáš Huvar, @Daniel Hejl, @Vojtech Vondra, @Jirka Bruncllík, @MicTech, @Tomas Zuse Ruzicka) held meeting (available on Rewatch, CZ only though, Notes from the meeting in English) where we discussed potential strategy regarding Data Fetching (and State Management). From my point of view we have in general two possible solutions. I would like to use this RFC to share my opinions and concern as champion for this problem.

We generally agree that Productboard needs to become thin client in order to become as a responsive app we have envisioned. This requires clear stand point on a topic how do we want (and to what extent) to guide team regarding data exchange architecture.

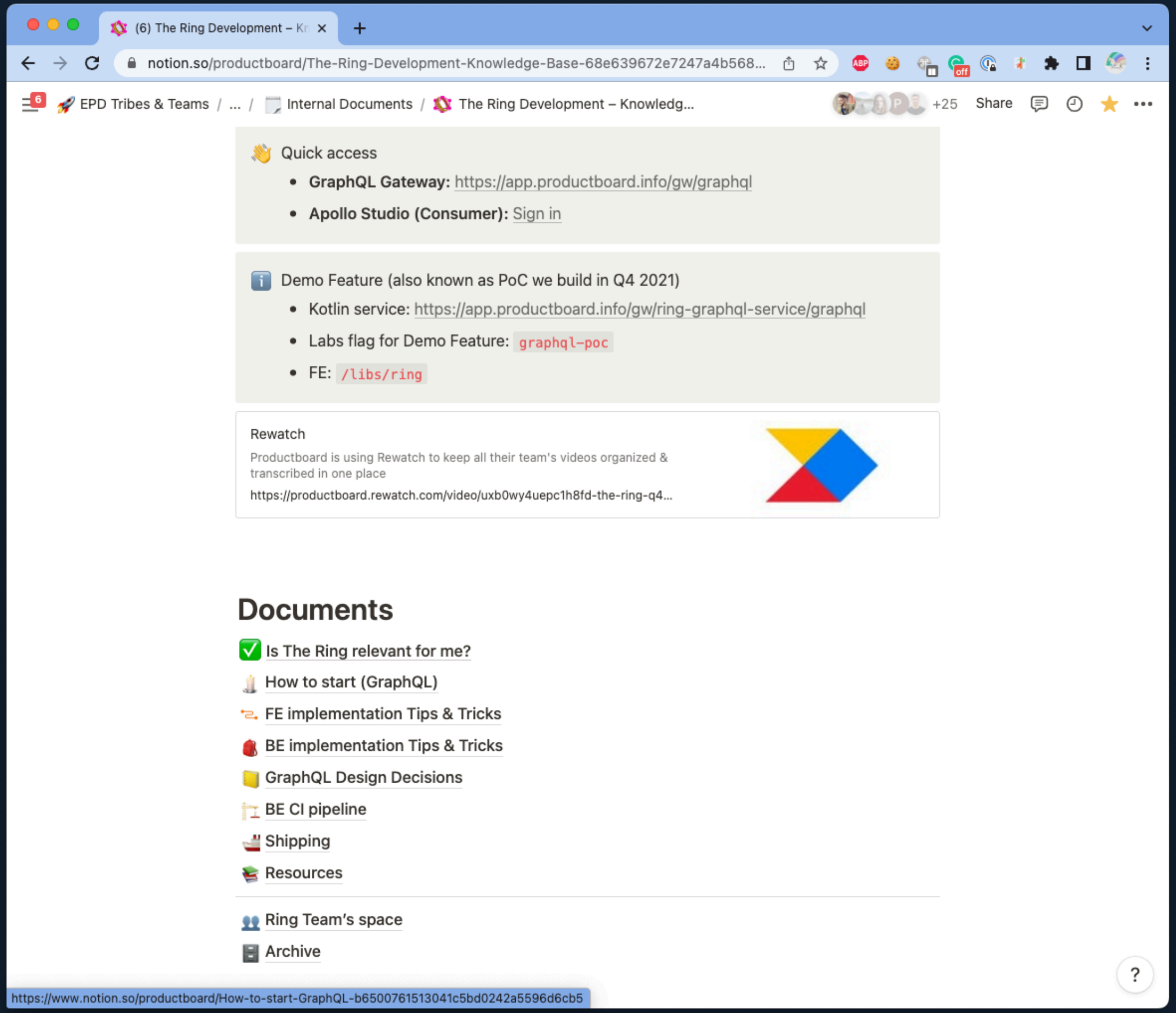
Goals

- Figure out future of `RestCollectionStore` and `RestRequestService`



```
type Feature {
  id: ID
  insights: [Insight]
  description: String!
  title: String!
}
```

TRANSPARENT (OVER)COMMUNICATION



(6) The Ring Development – Kn

notion.so/productboard/The-Ring-Development-Knowledge-Base-68e639672e7247a4b568...

EPD Tribes & Teams / ... / Internal Documents / The Ring Development – Knowledg...

+25 Share

👋 Quick access

- **GraphQL Gateway:** https://app.productboard.info/gw/graphql
- **Apollo Studio (Consumer):** Sign in

ℹ️ Demo Feature (also known as PoC we build in Q4 2021)

- Kotlin service: https://app.productboard.info/gw/ring-graphql-service/graphql
- Labs flag for Demo Feature: `graphql-poc`
- FE: `/libs/ring`

Rewatch

Productboard is using Rewatch to keep all their team's videos organized & transcribed in one place

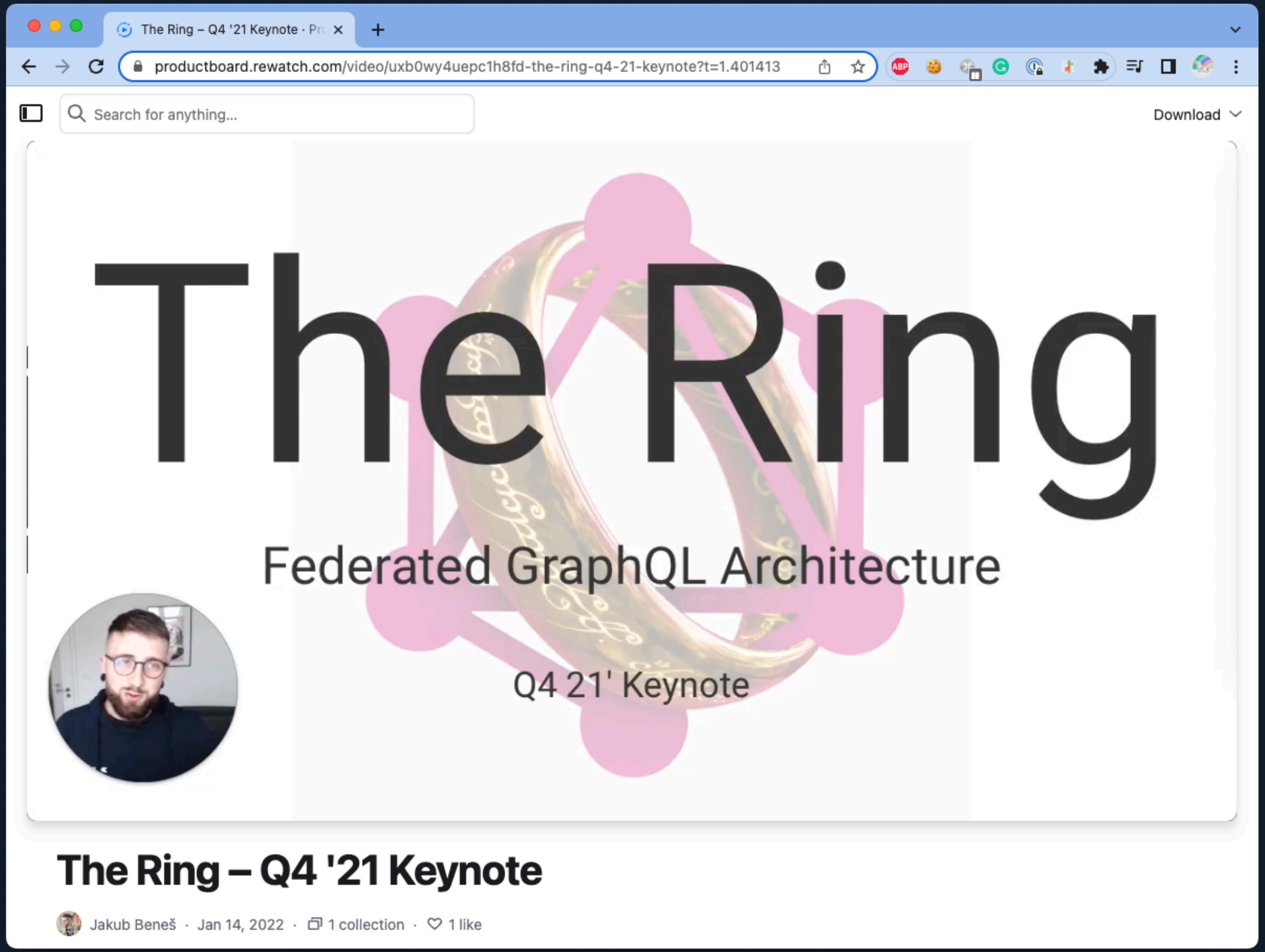
https://productboard.rewatch.com/video/uxb0wy4uepc1h8fd-the-ring-q4...

Documents

- ✅ Is The Ring relevant for me?
- How to start (GraphQL)
- FE implementation Tips & Tricks
- BE implementation Tips & Tricks
- GraphQL Design Decisions
- BE CI pipeline
- Shipping
- Resources

- Ring Team's space
- Archive

https://www.notion.so/productboard/How-to-start-GraphQL-b6500761513041c5bd0242a5596d6cb5



The Ring – Q4 '21 Keynote · Pr

productboard.rewatch.com/video/uxb0wy4uepc1h8fd-the-ring-q4-21-keynote?t=1.401413

Search for anything...

Download

The Ring

Federated GraphQL Architecture

Q4 21' Keynote

The Ring – Q4 '21 Keynote

Jakub Beneš · Jan 14, 2022 · 1 collection · 1 like

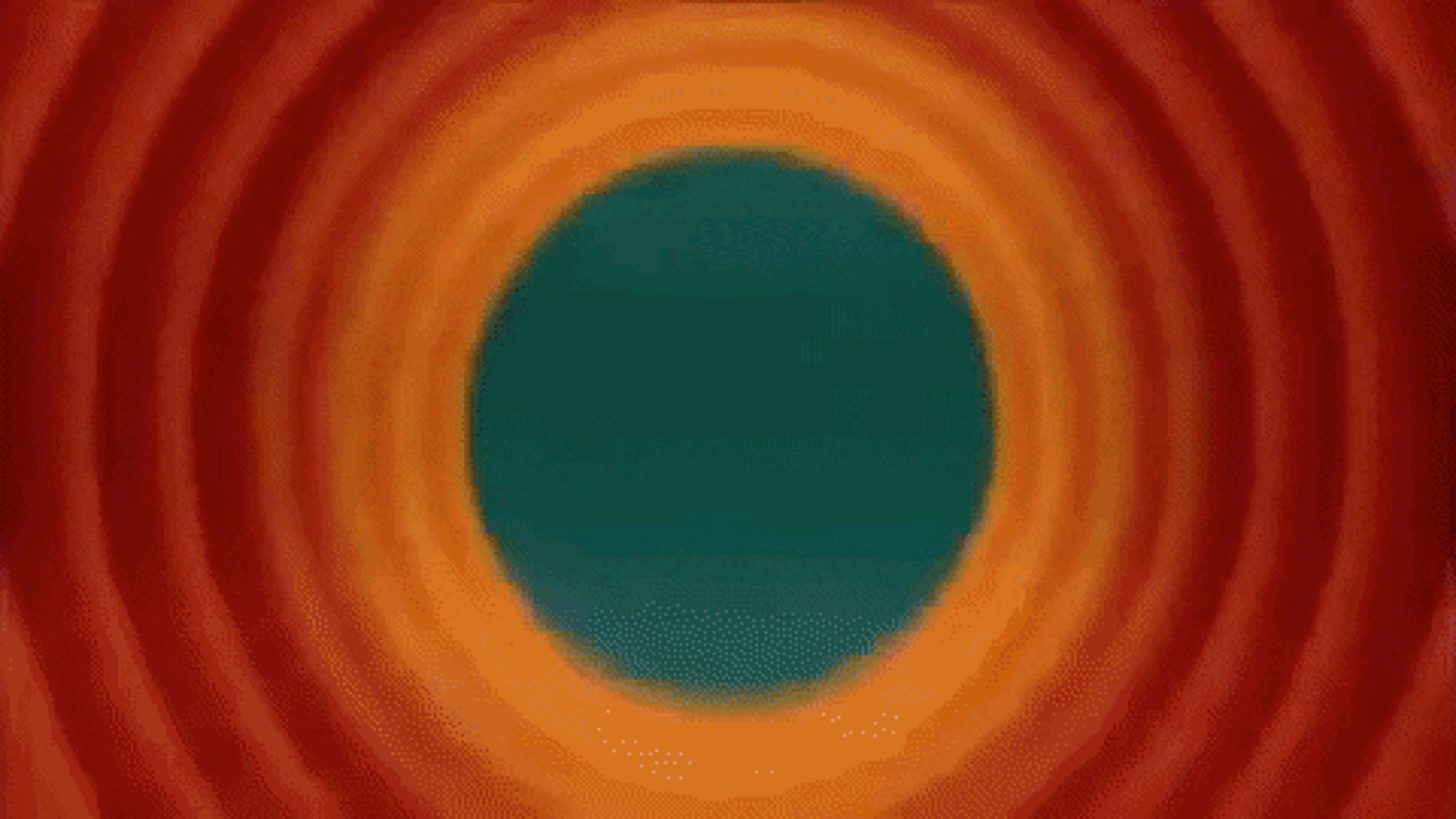
MAKE IT FUN

In Numbers

- We started in Q4 2021
- First product use-cases early this year
- Now we have 8 subgraphs connected
- 33 queries, 38 mutations, 150 types
- 500 RPM in peak

RECAP

- » Federated GraphQL is good especially if you want to iterate fast in distributed way
- » It's easier to run big thing if you name it
- » Big changes requires patience



THANK YOU!

Resources

- » Pretty Glonky
- » Apollo Federation
- » Relay and Relay's Global ID spec and Relay's Pagination spec
- » DGS
- » GraphQL Ruby
- » Production Ready GraphQL