The Internal Lens: GraphQL Gateways From a Different Axis

Angel Svirkov
Software Engineer

trivago

Agenda

1. The Different Axis
2. Revelations
3. What's Next

The Typical GraphQL Story

- Multiple clients
- External consumers
- Over-, under-fetching
- Coordinated subgraphs

Before 2020

Admin Tool X

Rest API

SSO

Confluence

Content Tool Y

Custom APIs

Shared API key

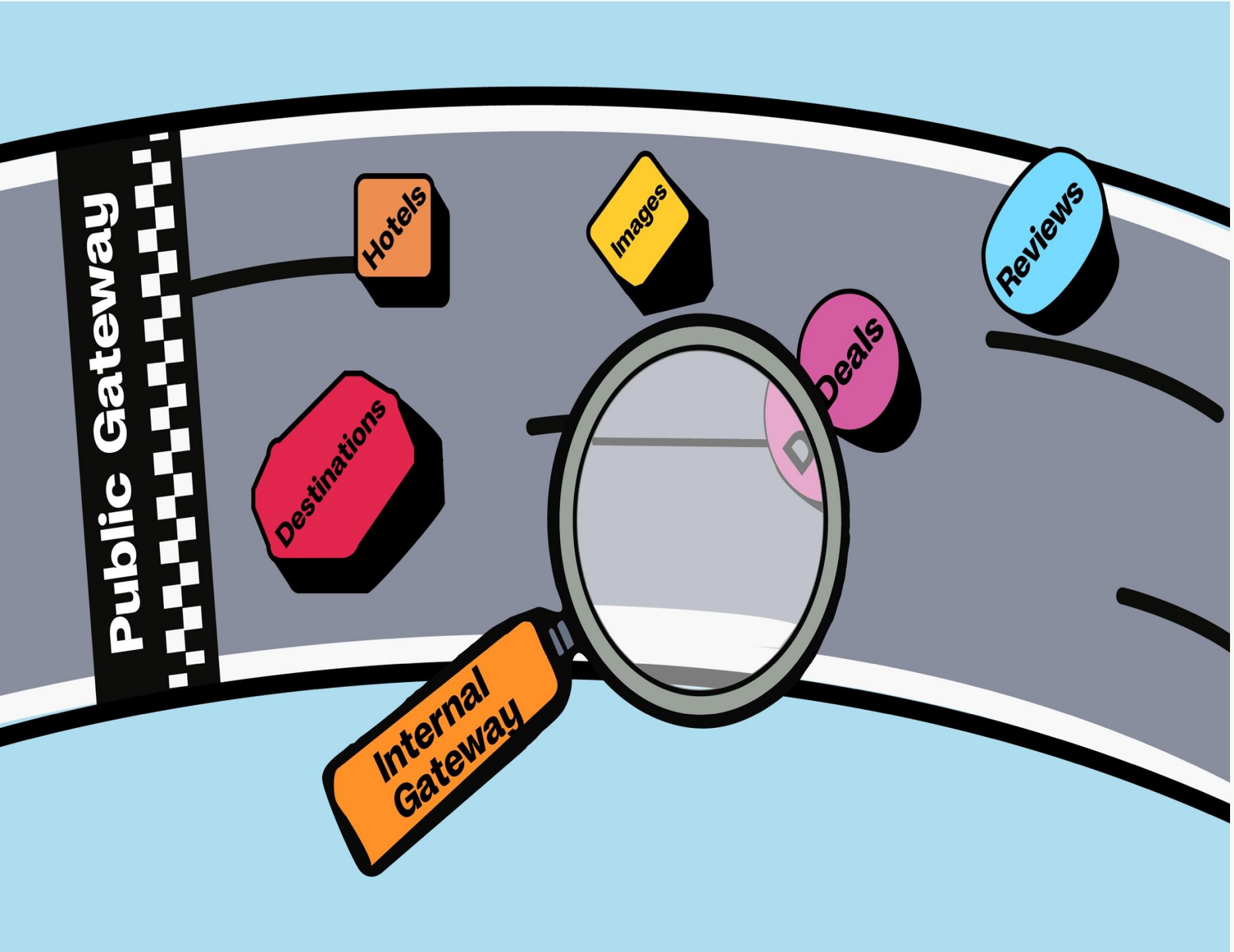
Strange URL

Rouge Workflow

MS Forms > Excel > SQL

Auth: ¯_(ツ)_/¯

Ask Michael



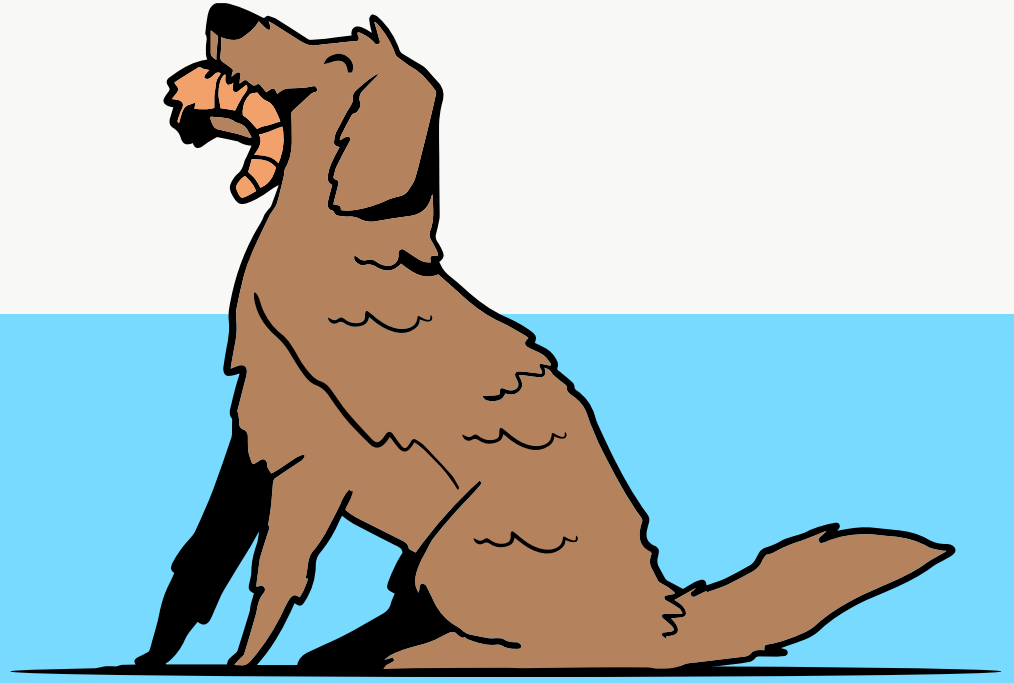
The Internal Gateway

- One client
- Internal consumers
- Fragmented APIs
- Uncoordinated services

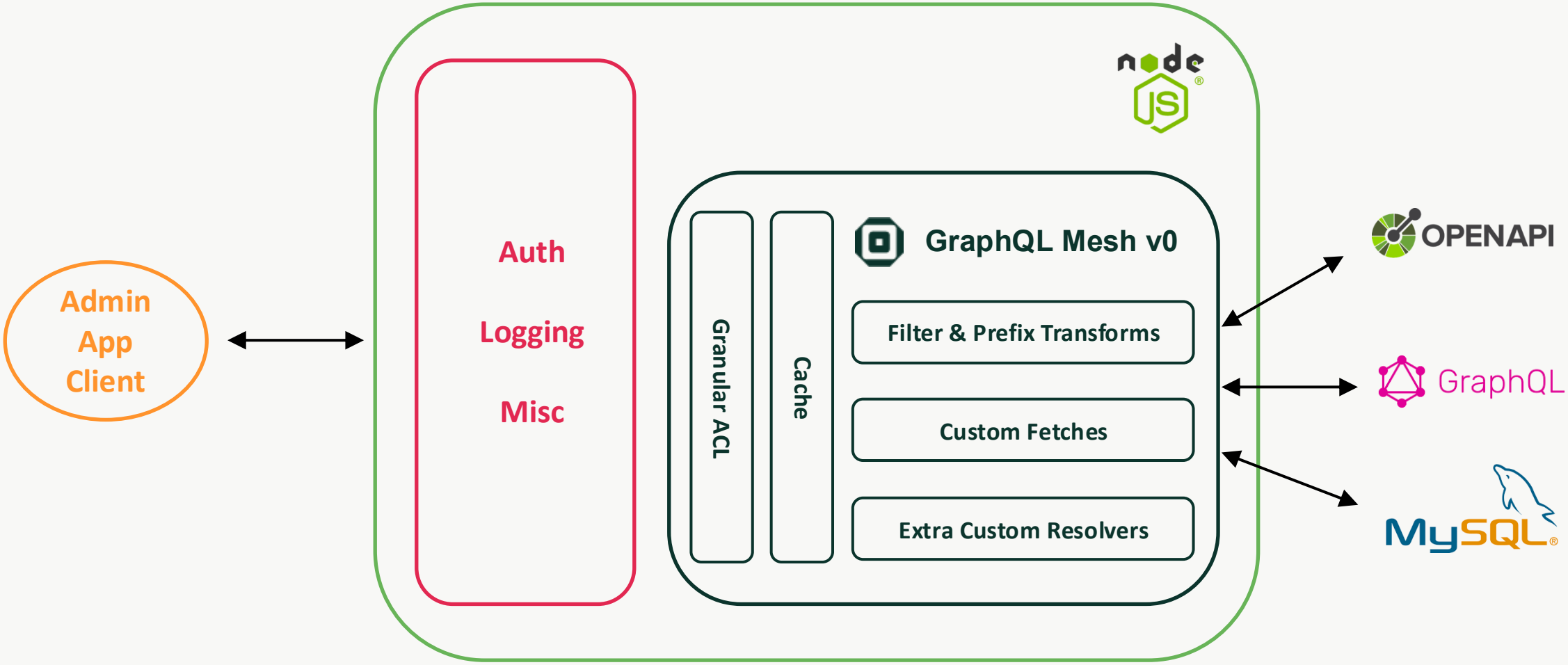
Give me a chance to work with GraphQL

trivago

GraphQL Mesh



Internal Gateway



Different Axis, Different Mindset

- Small team, manageable codebase
- No upstream modifications
- Fast iteration over perfect design
- Fill gaps wherever needed
- Changelog is a must-have

2000 types

120 queries

100 mutations

7 levels deep

15+ upstream sources

Changelog?

- No guarantee the upstream change succeeded
- Mutations may trigger async pipelines
- The actual change might happen hours later

More accurately: User Intent Log!

Audit Logging – User Intent

- Client provides extra metadata
- Bundle mutations into one log
- High-level domain scope
- Reason for the change
- State before and after

```
mutation UpdateHotelTranslations {
  de: updateHotel(
    id: "123",
    input: { lang: "de", name: "Hotel A" }
  ) {
    translatedName
  }
  jp: updateHotel(
    id: "123",
    input: { lang: "jp", name: "ホテルA" }
  ) {
    translatedName
  }
  logAuditEvent(
    scope: "Hotel"
    changeReason: "Reported with Task-123"
    input: [{
      expectedSuccessPath: "de.translatedName"
      expectedSuccessValue: "Hotel A"
      oldValue: "...", newValue: "Hotel A"
    }, {
      # ...
    }]
  ) { success }
}
```

2. Revelations

We Expected a Tool. We Got a Lens.

We expected a tool.
We got a lens.

What did the lens reveal?

- Fields nobody could explain
- Two teams maintaining overlapping solutions
- Data relationship assumptions
- Knowledge locked in people‘s heads

We expected a tool. We got a lens.

What did the lens change?

- Gateway team became consultants
- Fostered collaboration across disconnected teams
- Created new organizational knowledge

What was the trade-off?

- Wide breadth of expertise
- Not easy to hire for

What Does Unification Even Mean?

What does unification even mean?

Does it mean source APIs behave the same way?

Does it mean they share the same logical entities?

Does it mean they share the same environments?

What does unification even mean?

Entity Dilemma

One Hotel entity
with
deep relationships

VS.

RawStorage_Hotel
and
HotelContent_Hotel

Flat Graph

What does unification even mean?

Further Realities of Unification

Different Query Contracts:

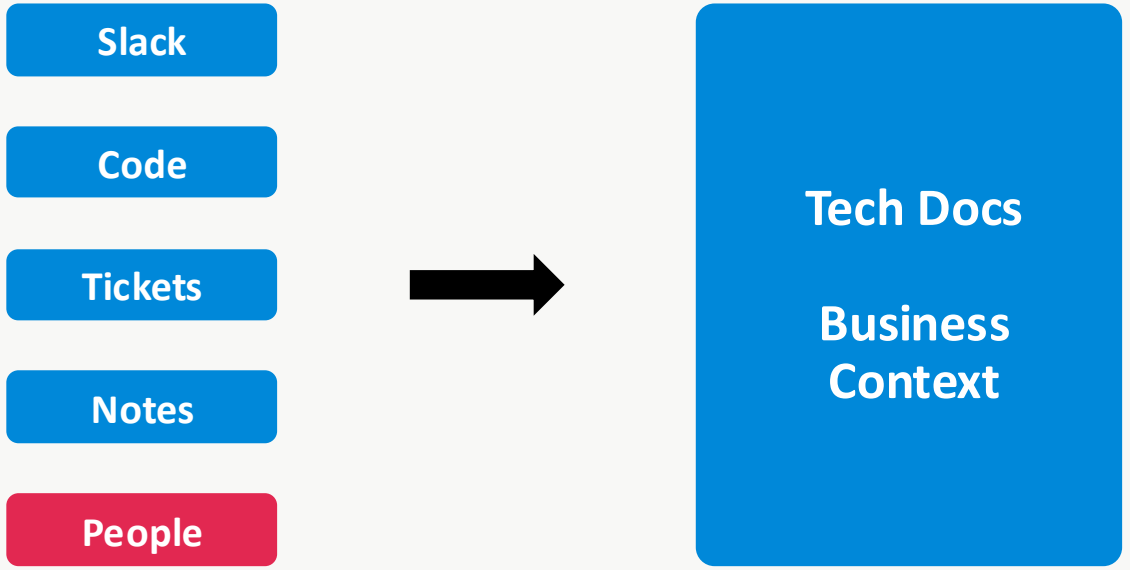
- Relay pagination
- Offset
- Custom filtering

Different Environments:

- Stage to Prod
- Prod to Stage

Documentation is Original Work

Documentation is original work



AI is artificial intelligence.

Feeding it with artificially generated documentation

produces a pretty artificial result.

Documentation is original work

GraphQL Introspection

- Types
- Queries
- Mutations
- Descriptions (if written)

What we need

- Why does this field exist?
- When to use a mutation?
- What it means to to the business?
- How do domains relate?

Schema registry from day one

Schema registry from day one

What Happened?

- Internal Gateway is a secondary concern
- APIs changed without informing us

What to do instead?

- Schema Registry integrated into CI/CD
- Early compatibility check

The Goal

- Not blocking deployments
- Awareness is key

3. What’s Next?

Beyond 2026

AI Generated Tool X

Which API
does it use?

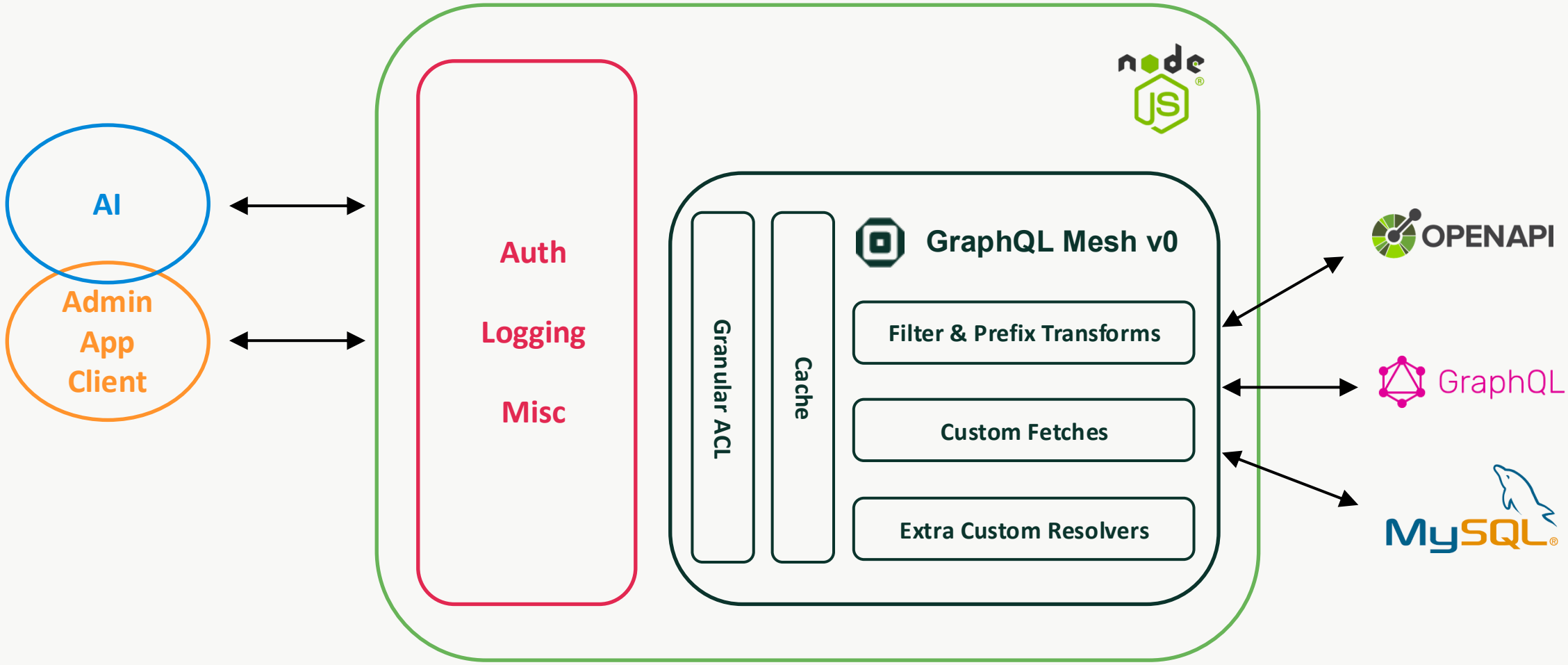
AI Generated Service Y

Who owns
this?

Vibe-Coded Workflows

Just use
my credentials

Internal Gateway



The Foundation for Future Workflows

1. Human: “This hotel’s pricing looks wrong”
2. AI Agent: Queries the gateway
3. Agent cross-references business documentation
4. Agent surfaces: Data diverges at enrichment step
5. Agent fixes via gateway mutations

The Foundation for Future Workflows

- Powerful Gateway Features
  (connectivity, logging, access control)
- Documented Business Context
  (human-written)
- GraphQL+
  (query contracts, semantics, metadata)

If SQL helps you understand a database,

GraphQL can help you understand your company.

Thank you!

tech.trivago.com –
Unifying Internal APIs: A Different Use Case for GraphQL Gateways