Get the most value out of your APIs

with GraphQL and IBM StepZen

Core Problem:
Speed of app development is hindered by complicated service & data silos

"Our data is scattered in multiple SAP and Oracle systems, slowing down applications that need longitudinal view of our customers.": **CTO, Fortune 500 Enterprise Cloud Company**

How to access diverse, complicated silos?

?

SQL & NoSQL Databases & Data Lakes	Enterprise Apps	SaaS Apps	Microservices

SQL & NoSQL Databases & Data Lakes: PostgreSQL, MySQL, Microsoft SQL Server, ORACLE DATABASE, DataStax, IBM DB2, MongoDB, CouchDB, trino

Enterprise Apps: salesforce, ORACLE, Mavenlink, SAP S/4 HANA

SaaS Apps: stripe, shopify, contentful, HubSpot

**Microservices:** GraphQL, { REST }, XML, OPENAPI INITIATIVE

Solution:
A GraphQL API that abstracts the backends away

"A single GraphQL API that covers our Postgres, Salesforce and CMS systems and delivers an integrated view is what we need": Development Lead, Industrial Manufacturing Company

GraphQL APIs enable access to diverse sources

Frontend developers love GraphQL:
- Very easy to consume
- All backends get abstracted away

GraphQL

Backend developers like GraphQL:
- Lets everyone move faster
- But struggle with success:
  - skills
  - scale, support and maintenance

| SQL & NoSQL Databases & Data Lakes | Enterprise Apps | SaaS Apps | Microservices |
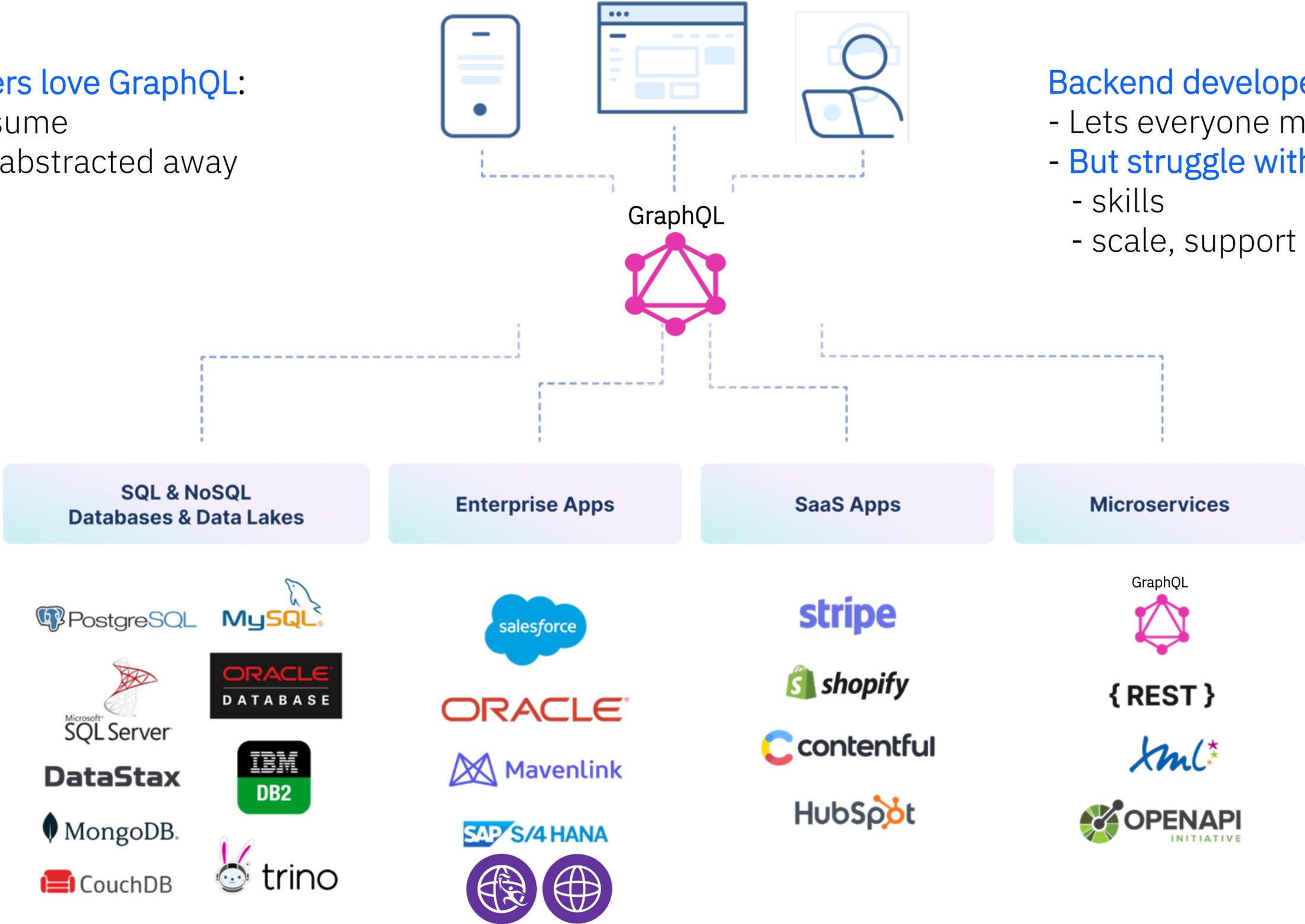
What are your challenges?
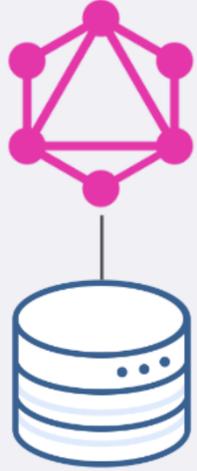
| GraphQL on a single database | GraphQL on multiple backends | GraphQL Federation |
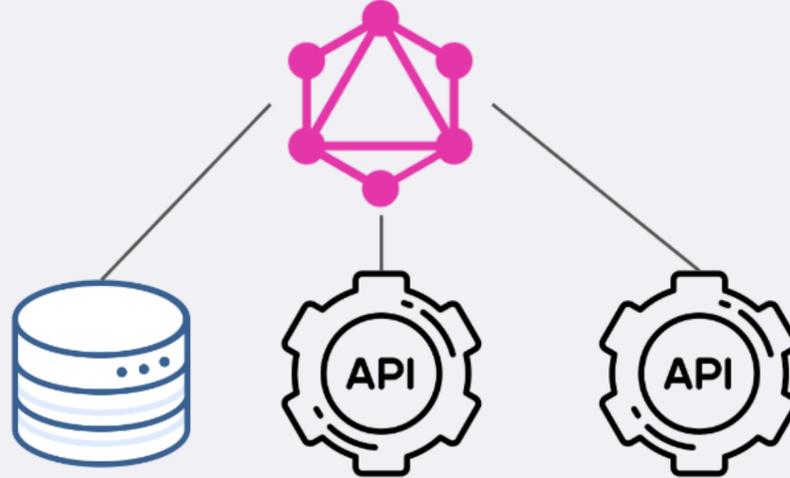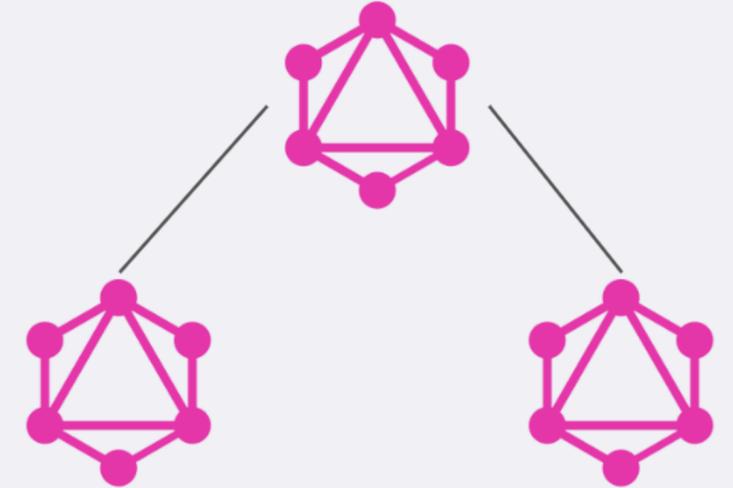
- DIY, using open source tools
- Apollo
- Hasura
- App Sync
- Cloud, and if so, which cloud?

- Experimenting, so not sure
- Performance
- Governance
- Change Management
- Security and Access Control

What is GraphQL anyway?

Ask for bits you want, get the bits you want

GraphQL is about the shape of the API

Consumers invoke like, but different from, REST

- Request data, drive actions, use JSON.

- GraphQL enables requestor to select what exact information it needs


```
Describe your data            Ask for what you want          Get predictable results

type Project {                {                              {
  name: String                  project(name: "GraphQL") {     "project": {
  tagline: String                 tagline                        "tagline": "A query language for APIs"
  contributors: [User]         }                              }
}                             }                              }
```

Source: GraphQL Foundation

Access data like SQL

- Provides abstraction to multiple backends

- Enables access to diverse set of sources from one API


GraphQL

@dbquery   @rest   @graphql

GraphQL

**SQL
Backends**   **REST, SOAP, OData,
& NoSQL Backends**   **GraphQL
Backends**
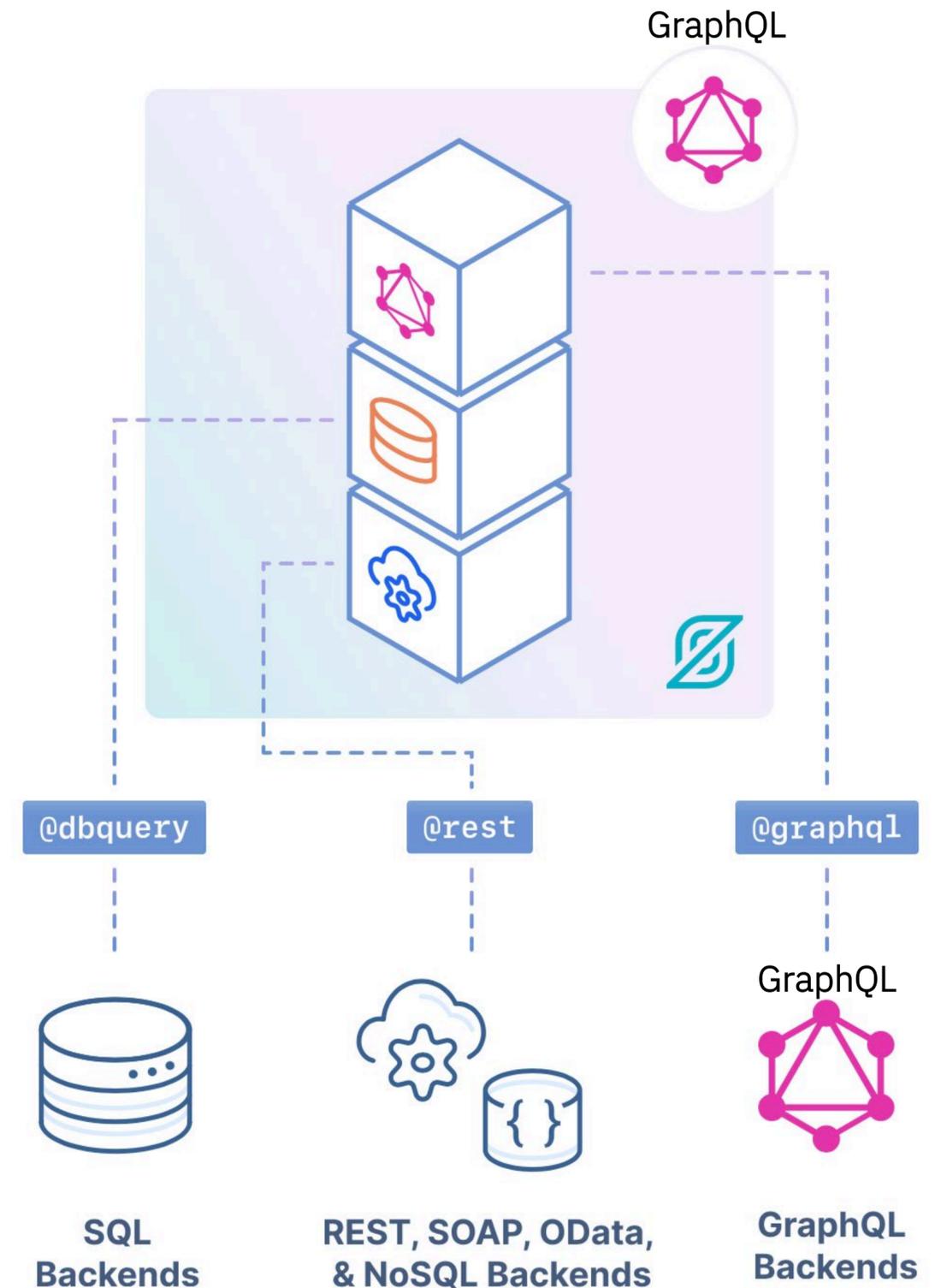
GraphQL enables you to ...

Leveraging existing investments...

... through a Uniform Interface

Be **consistent** across backends

@dbquery

@rest

@graphql

SQL
Backends

REST, SOAP, OData,
& NoSQL Backends
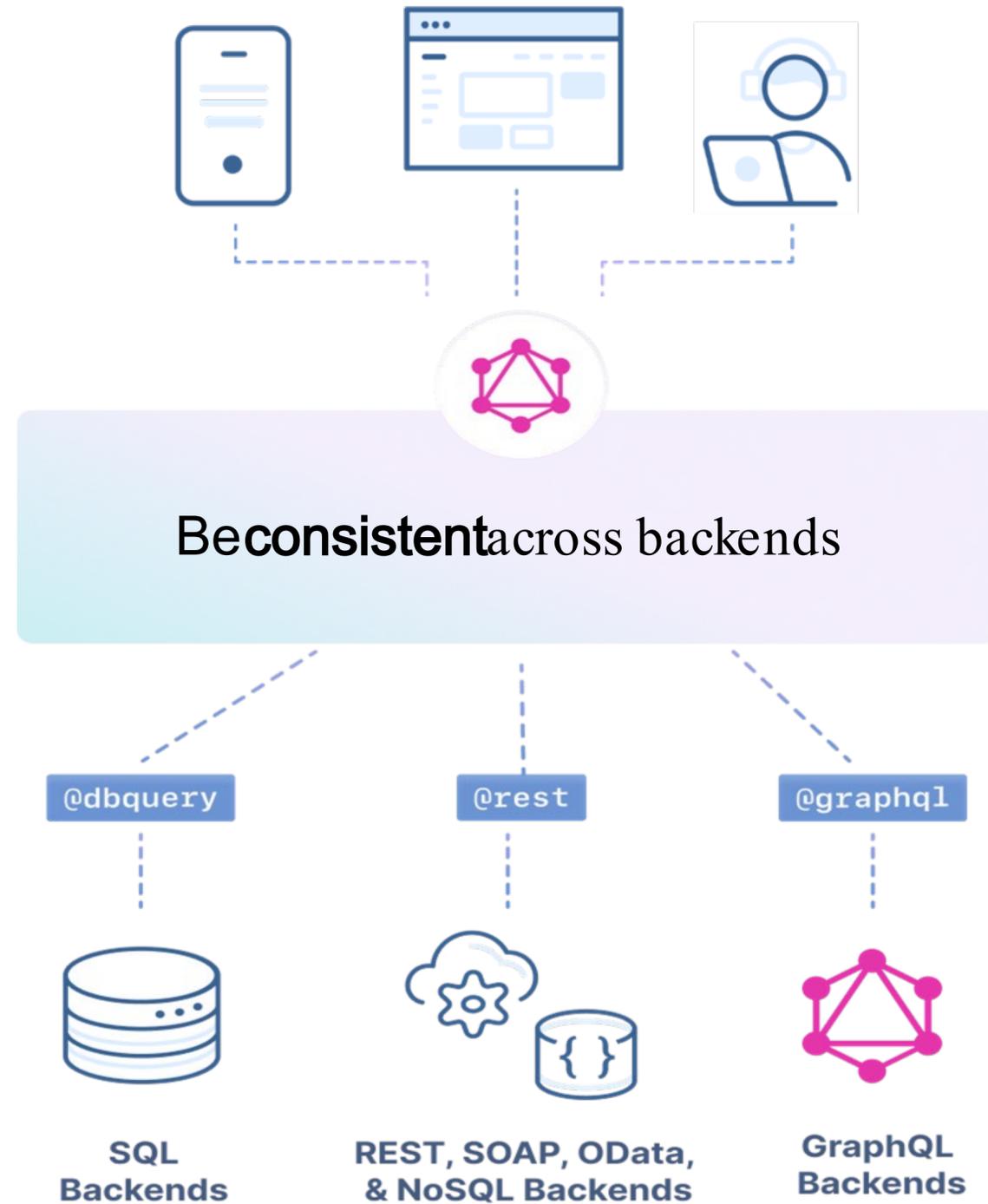
GraphQL
Backends

GraphQL has its challenges

How would you...

- optimize database performance?

- deploy in private clouds?

- manage changed schema?

- handle slower backends?

- federate across domains?

- secure access to the API? And secure access to backends?

StepZen makes GraphQL easy

StepZen solves hard problems, so you can focus on value

StepZen is a GraphQL engine

- Build GraphQL APIs declaratively (no code)
- Introspect diverse backends to easily surface data and services new apps need
- GraphQL server runtime
- Optimizes the hard problems for you

StepZen

an IBM Company

Optimization
Protection
Scale & Low
Latency
Deployment
Caching
Geo Distribution

1 **Developer** GraphQL-izes backends, *easily*

2 **Developer** "stitches" the GraphQL-ized backends, *easily*

3 **StepZen** does the hard things, *transparently*

StepZen is a GraphQL server using composition

1 **Build**
the GraphQL schema using declarative building blocks

@dbquery   @rest

Customer Data   Order Data

2 **Deploy**
the schema into an API using the CLI

3 **StepZen serves the GraphQL API**
built-in optimization & scale

Public, Private
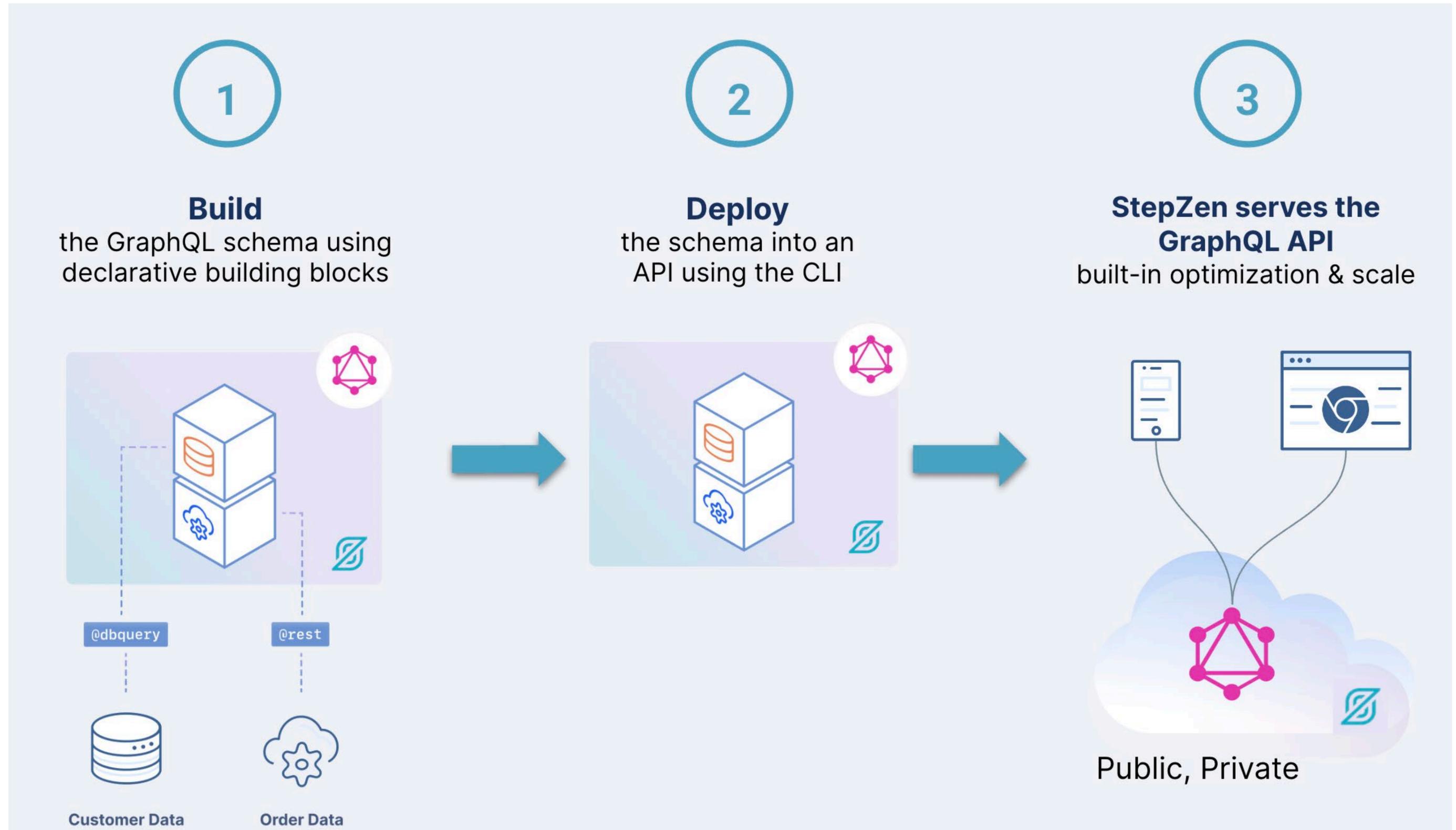
Composition for teams, and team of teams
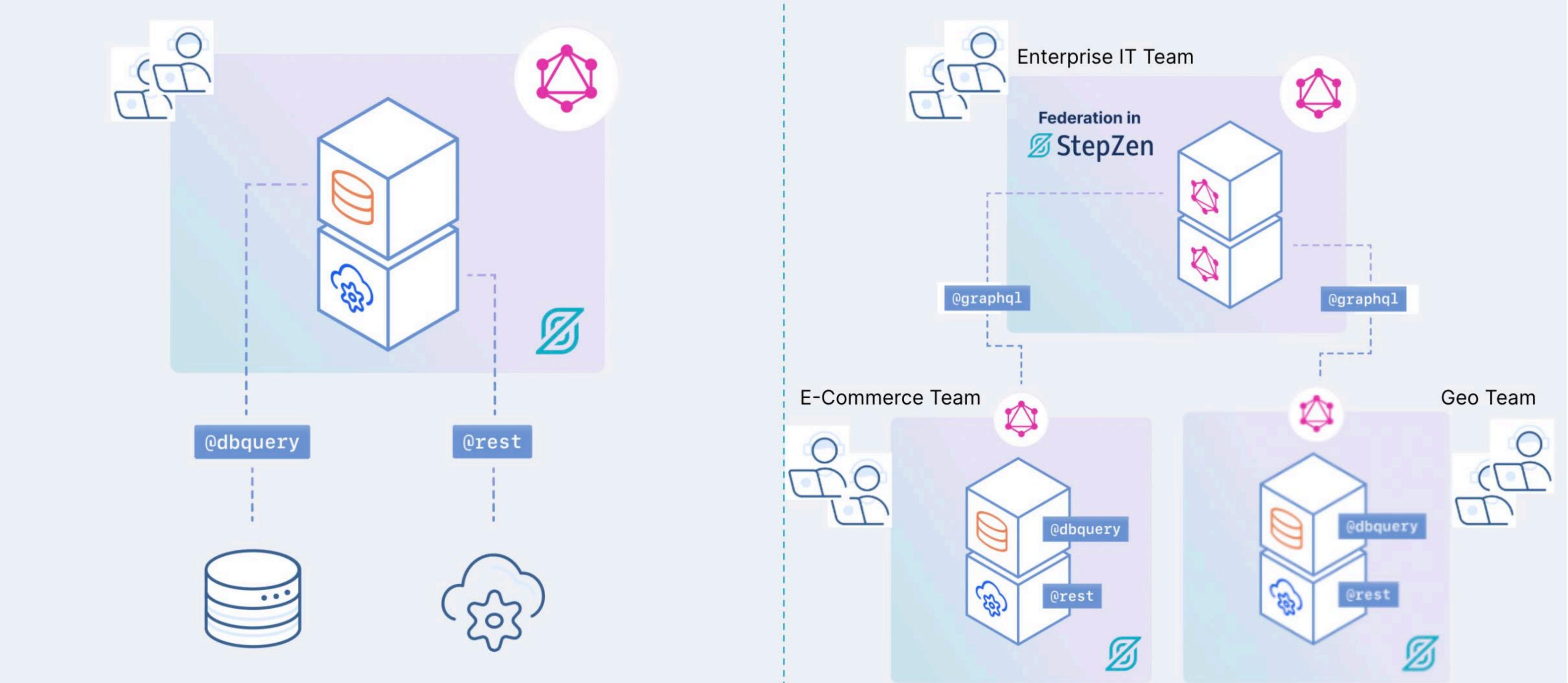
GraphQL + API Management

GraphQL APIs, like REST APIs, benefit from API Management

Lifecycle Management

Package, publish, promote and reuse APIs across environments with full lifecycle control

Govern & Secure

Apply enterprise-wide governance and security policies, consistency across the API estate
(REST, GraphQL, and more)
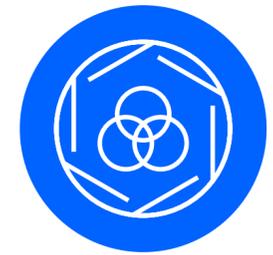
Socialize APIs

Empower API consumers to discover, test, & consume APIs building your digital brand

StepZen + IBM API Connect

Create
Build APIs your consumers will love: declaratively build GraphQL APIs.

Manage, Secure & Socialize
Package, publish, promote and socialize. Control and mediate delivery with API Gateway.

StepZen
an IBM Company

IBM API Connect

Build here

Manage there

SOFTWARE DEVELOPER

Build and consume APIs with multiple languages & devices
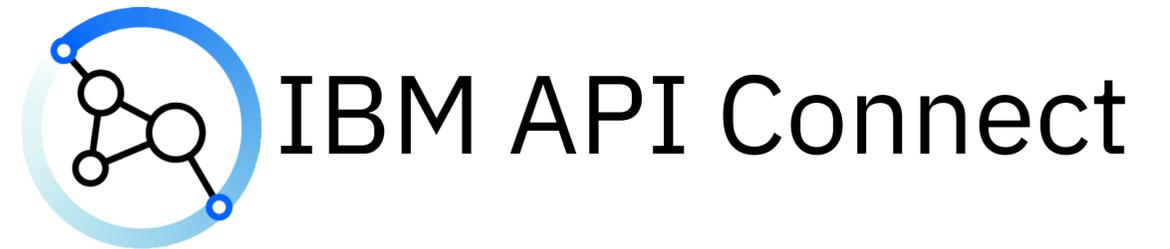
# API Management across the API estate

IBM API Connect
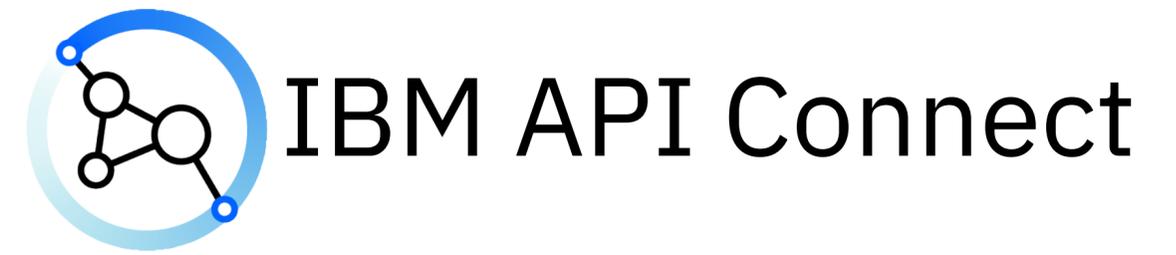
Consistently access, manage, secure and socialize
StepZen-built GraphQL APIs alongside the rest of your APIs

Share your API estate through developer portal

API gateway

StepZen
an IBM Company

Developer Portal

API Manager

SQL & NoSQL Databases & Data Lakes

PostgreSQL
MySQL
Microsoft SQL Server
ORACLE DATABASE
DataStax
IBM DB2
MongoDB
CouchDB
trino

Enterprise Apps

salesforce
ORACLE
Mavenlink
SAP S/4 HANA

SaaS Apps

stripe
shopify
Contentful
HubSpot

Microservices

GraphQL
{ REST }
XML
OPENAPI INITIATIVE
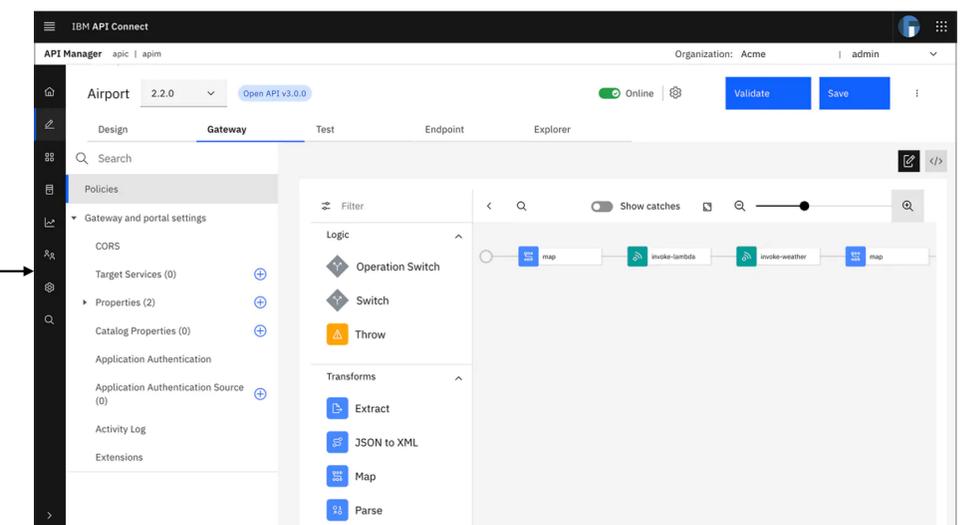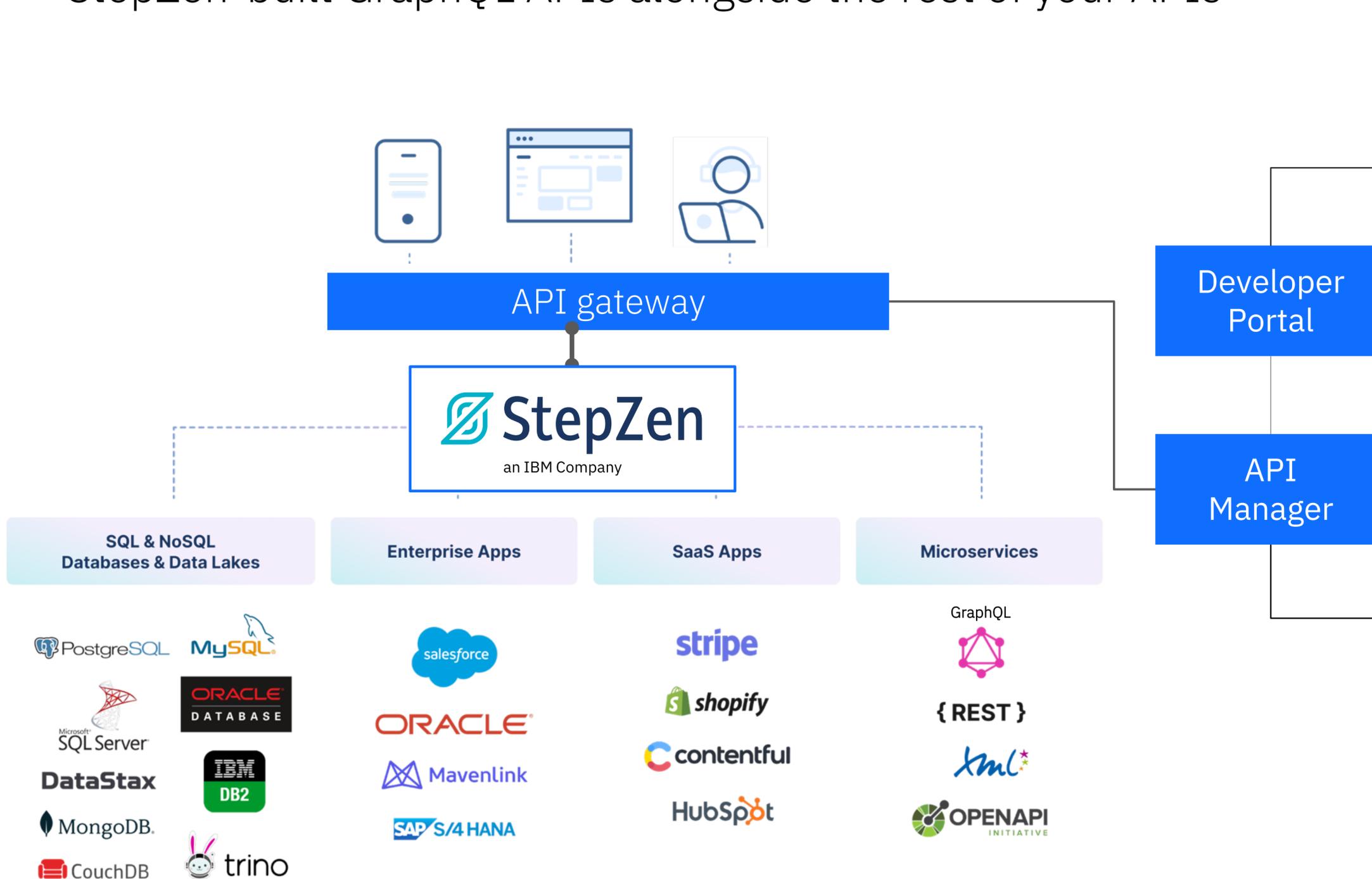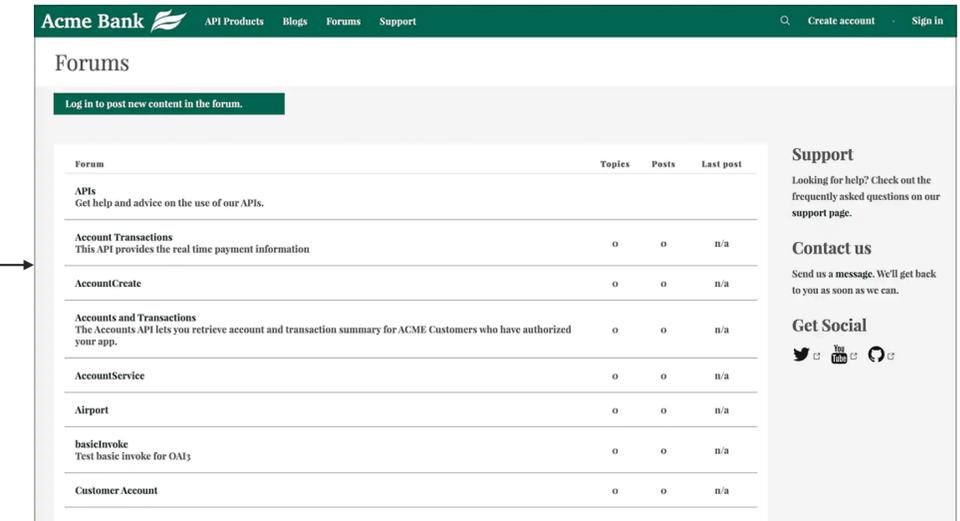
Secure and govern APIs with out of the box and custom policies

Thank you