



Managing the Complexity of Microservices Deployments

Kenny Bastani (Principal Technologist)
Prithpal Bhogill (Product Management)

May 9, 2017

Proprietary + Confidential

Reminder: “This webinar may include mention of some future features. Please make any purchasing decisions based on currently GA features.

This information is Google confidential information, covered by the terms of Apigee and Google Cloud Platform License Agreement.”

Speaking Today



Prithpal Bhogill
Product Management



Kenny Bastani
Principal Technologist

Agenda

Monolith to Microservices

Why API Management?

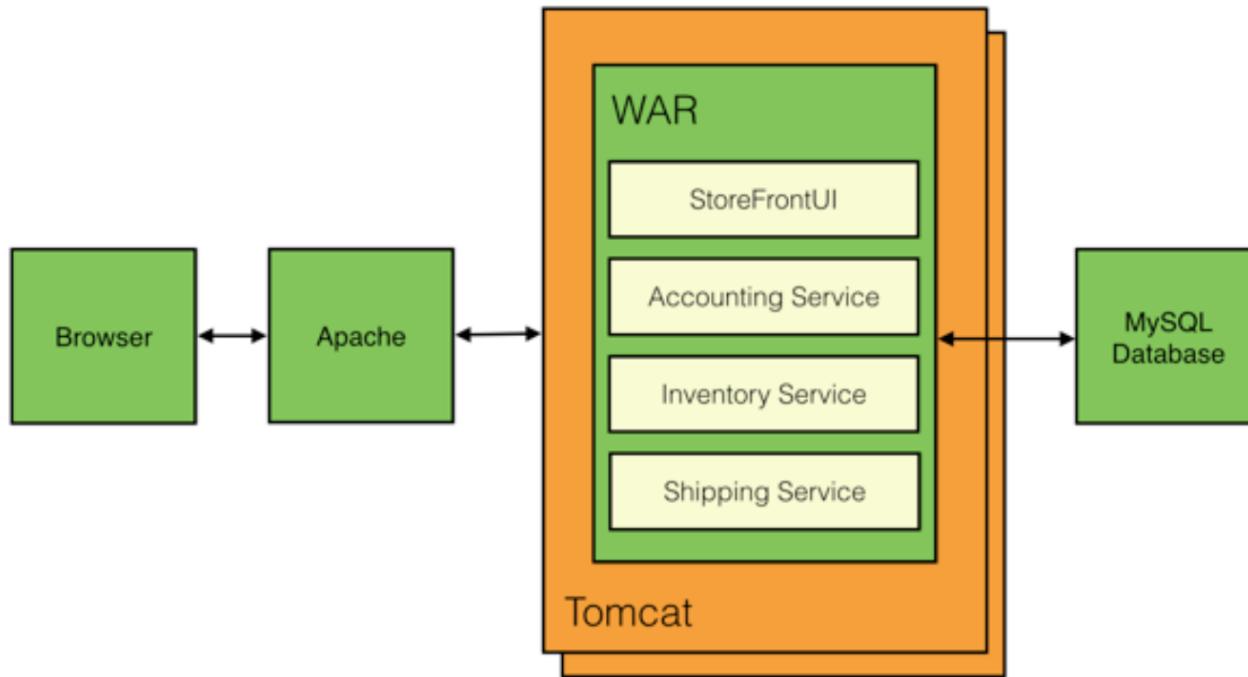
Apigee and Pivotal Cloud Foundry

Customer Benefits

Call to Action

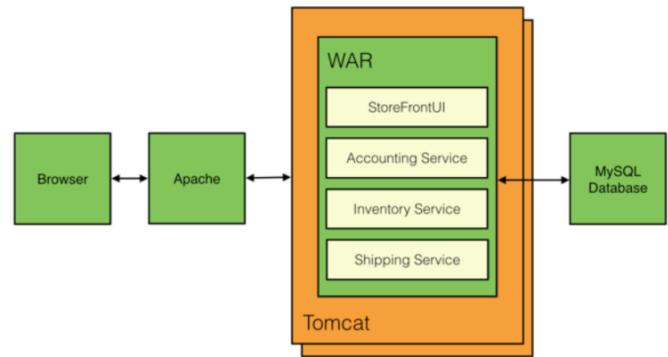
From Monoliths to Microservices

We started with the monolith



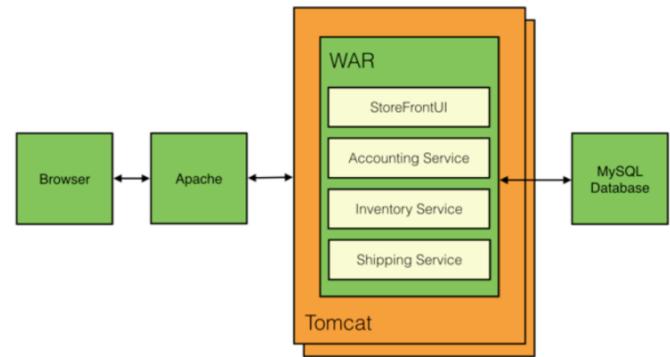
Cultural problems with monoliths

- Slows our velocity getting into production
- It takes too long to ramp up new engineers
- The code base is just too large for any one person to fully comprehend
- Centralized authority and change management slows progress (DBA, Ops)

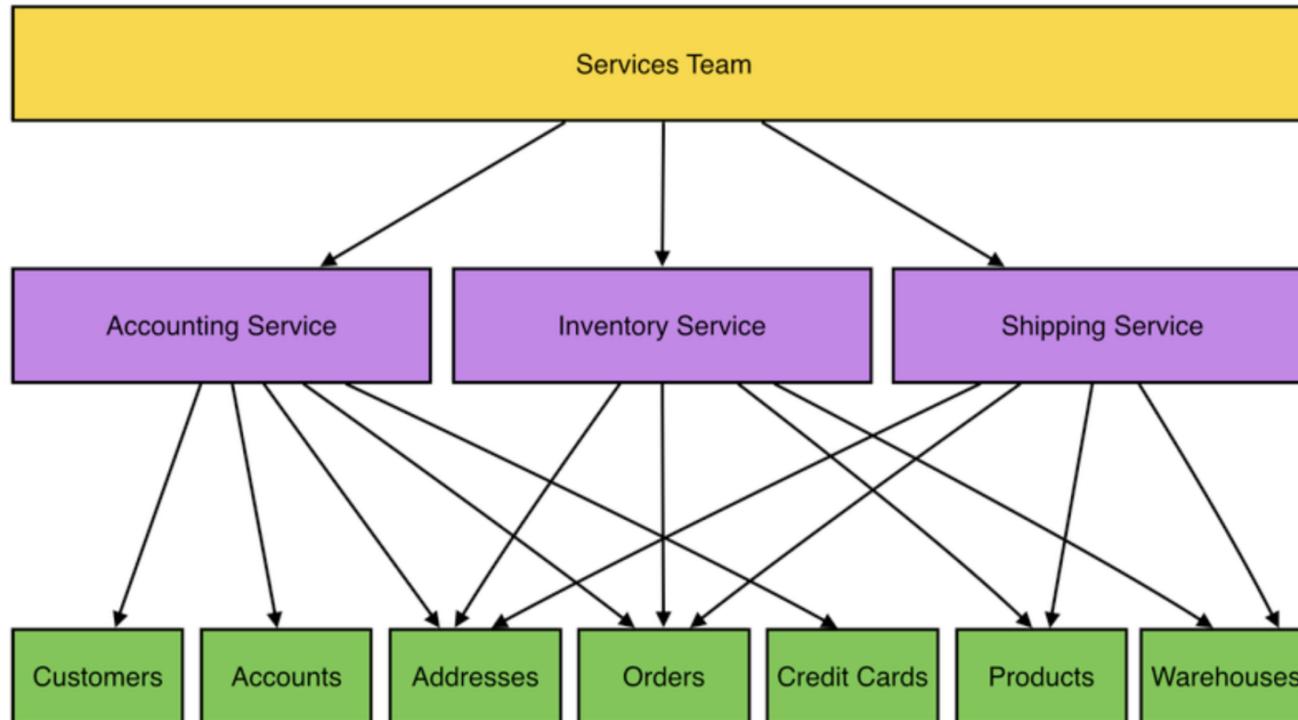


Operational problems with monoliths

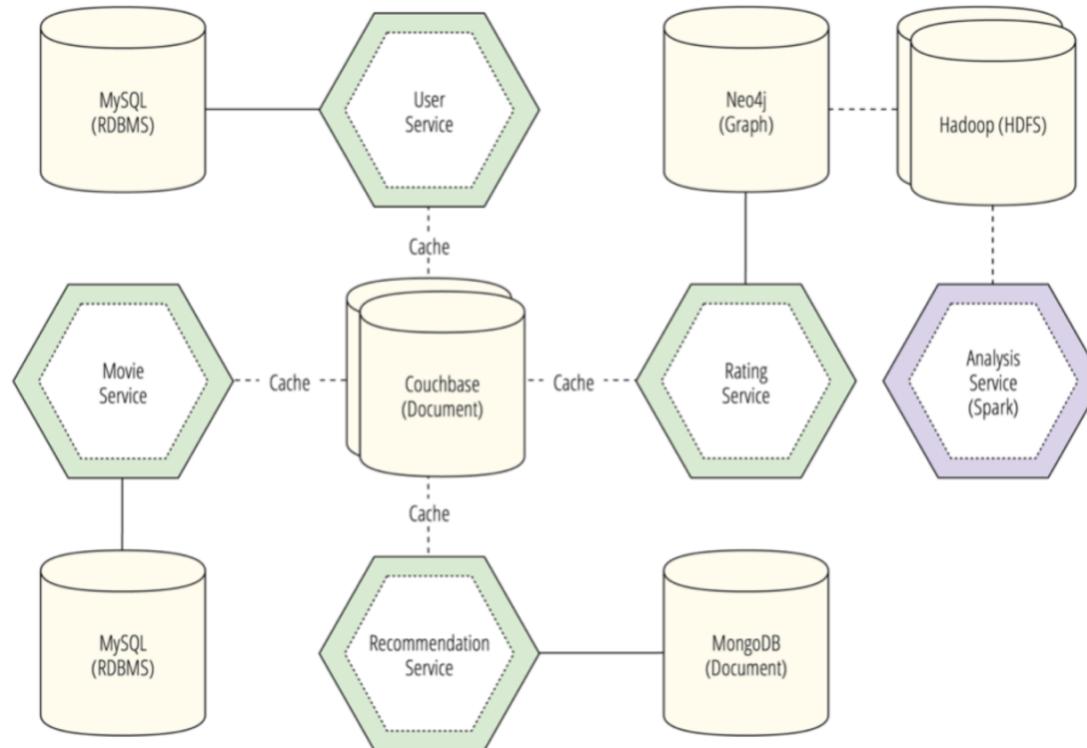
- Coordinated releases batch many changes together from different teams
- Operations drives the runtime environment of applications
- Operators take on all operational responsibility (including VM upgrades)
- *Deploy everything at once or nothing at all*



We then moved towards SOA

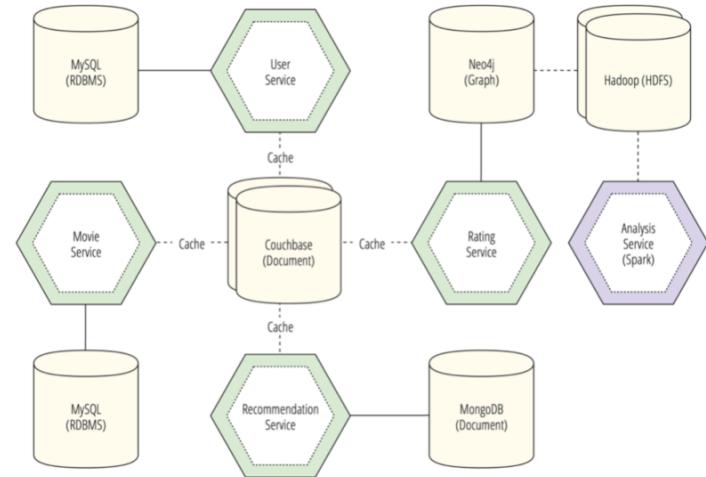


We have now arrived at microservices

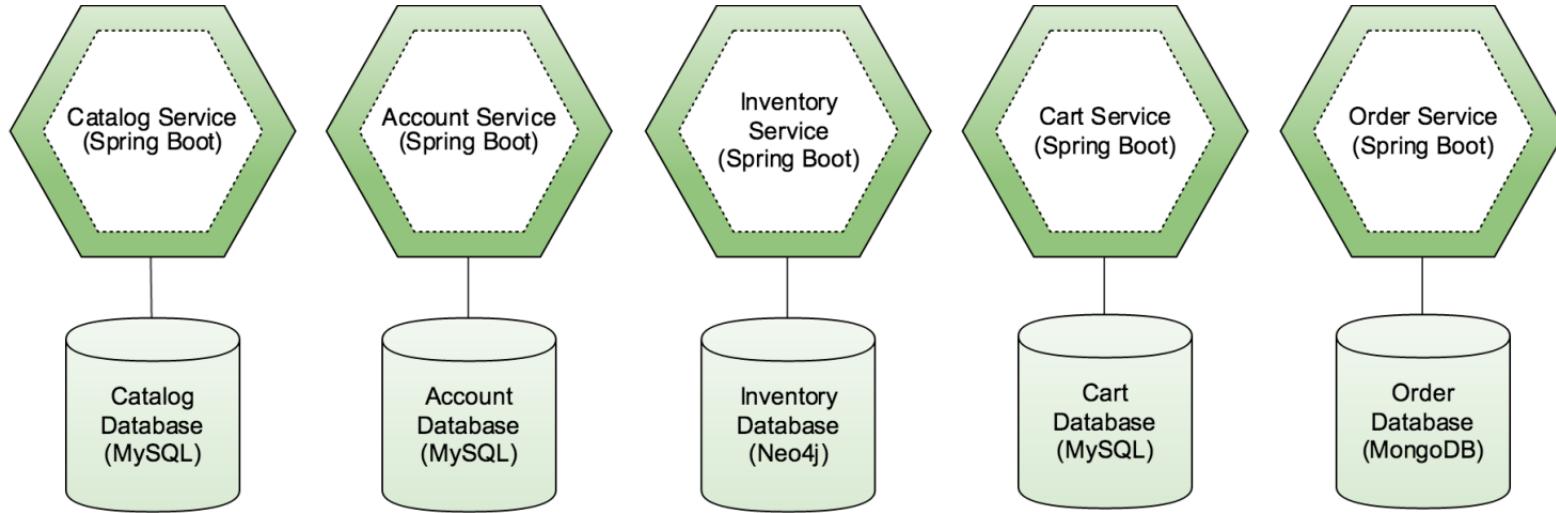


Microservices

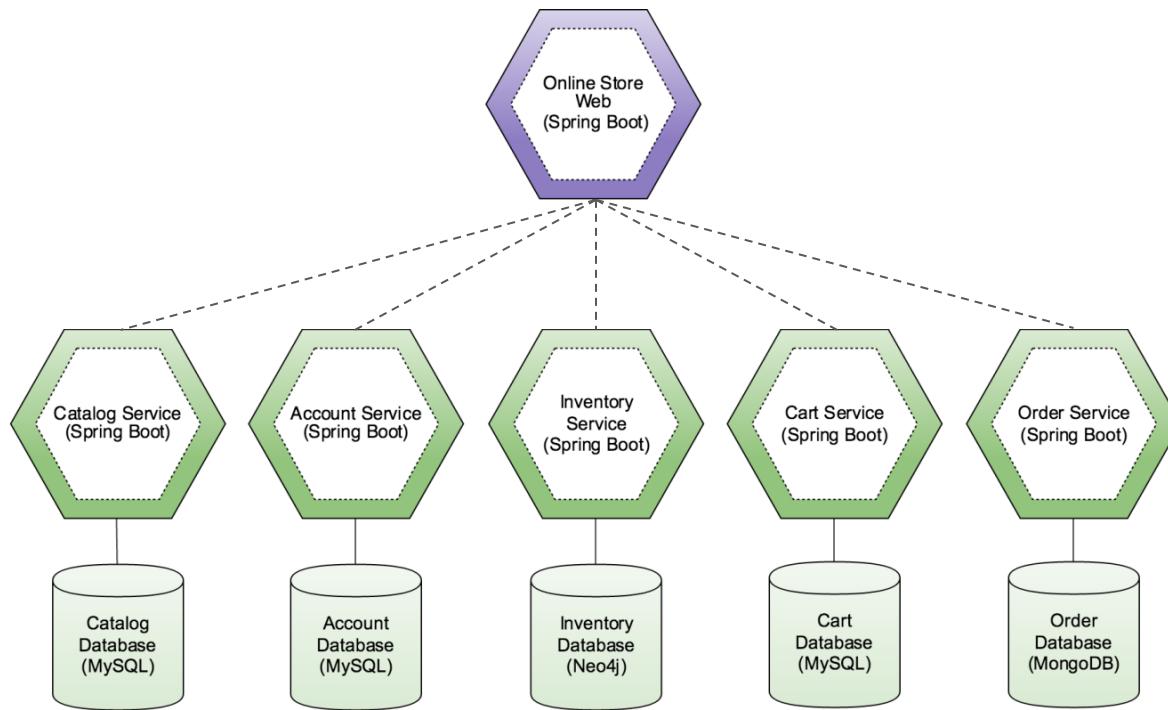
- Small teams organized around business capabilities
- Independently deployable
- Teams produce and consume APIs
- Teams can choose the best tools for the job

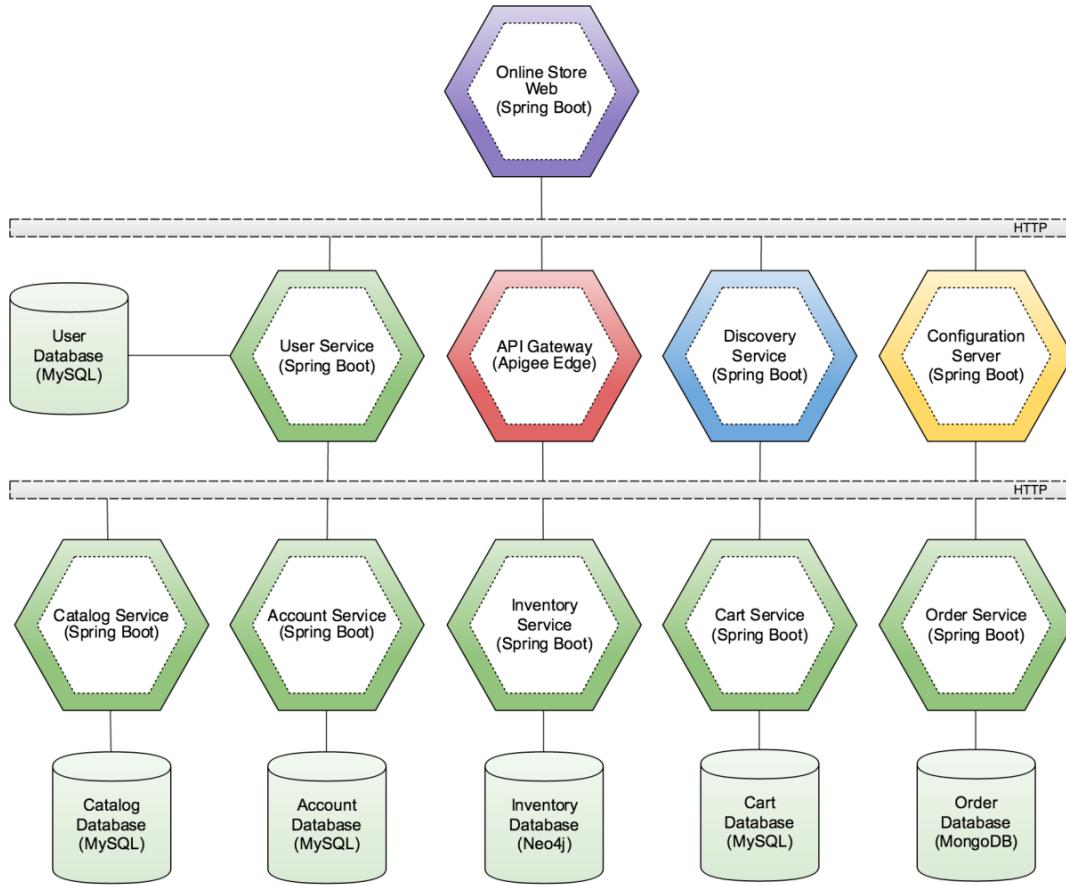


Online Store Microservices

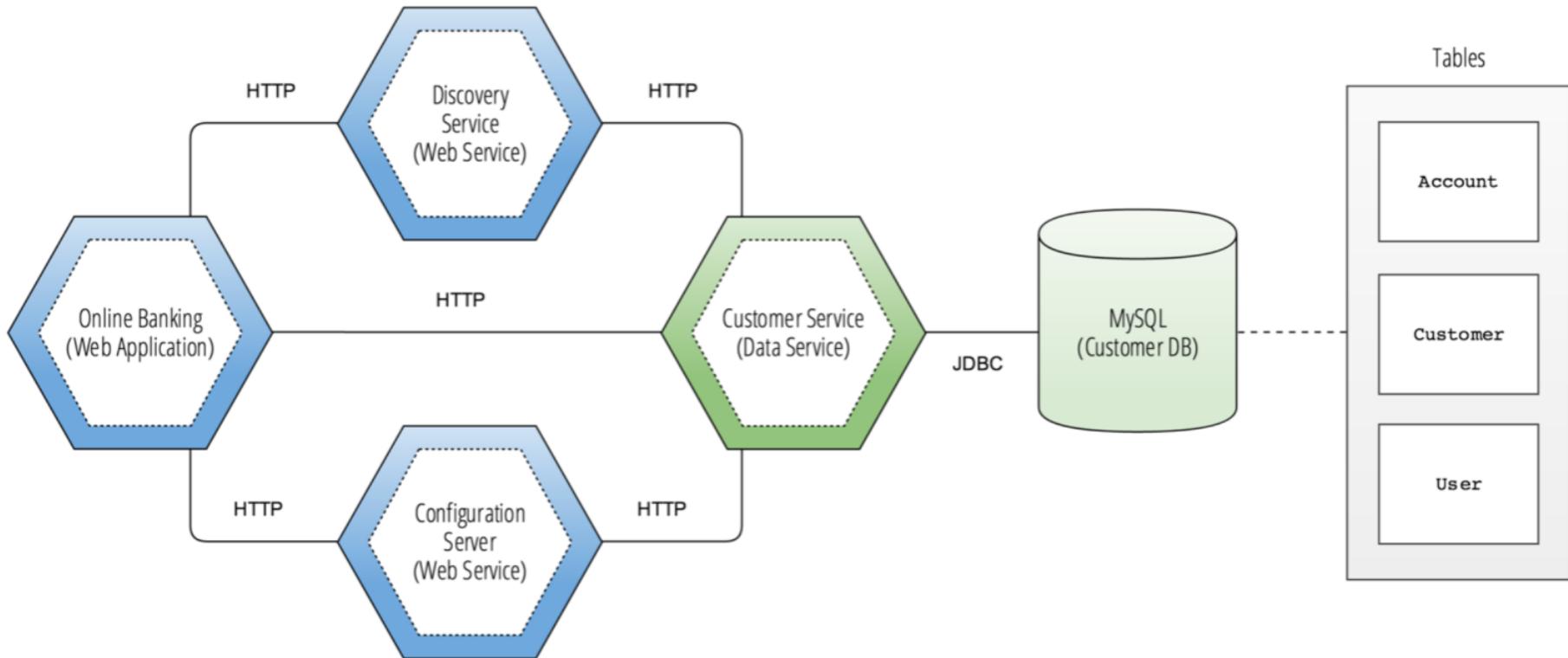


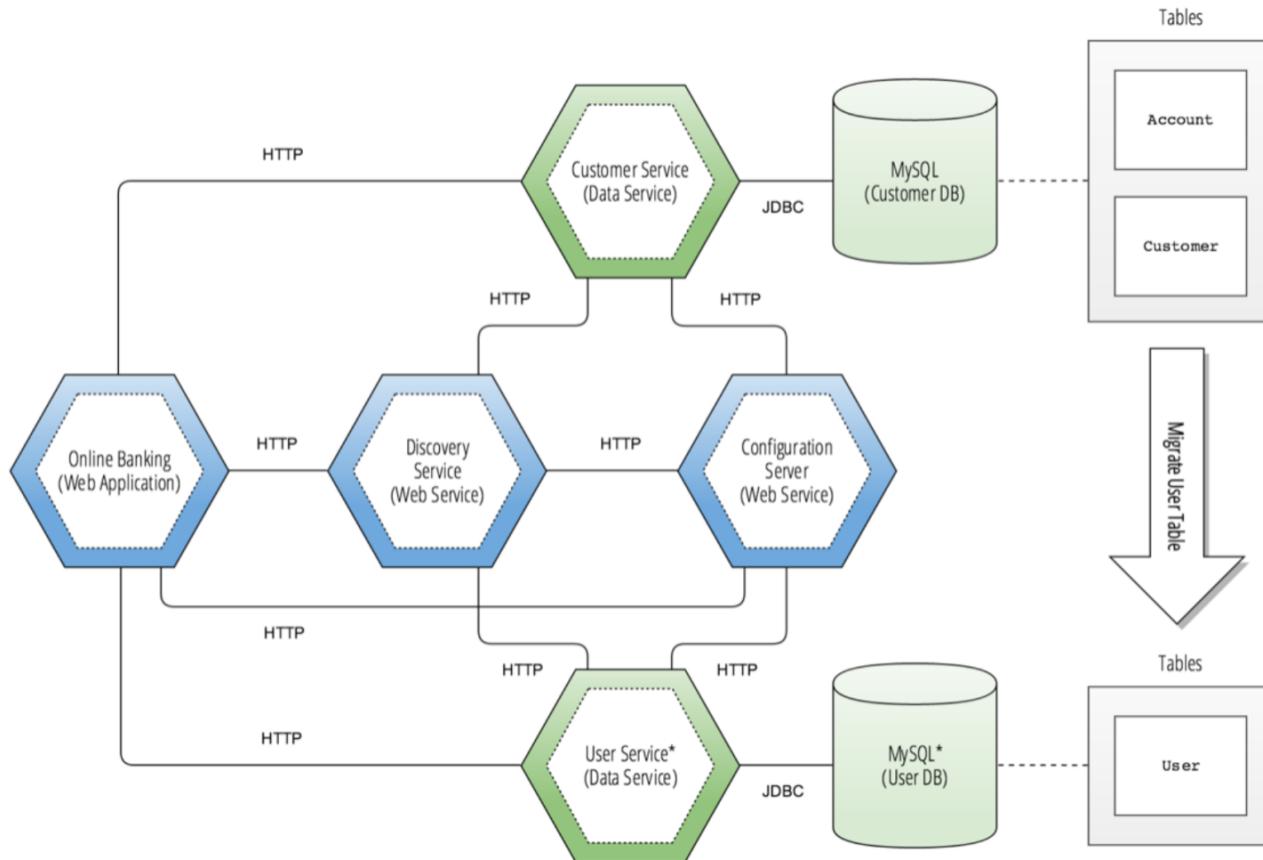
Online Store Microservices



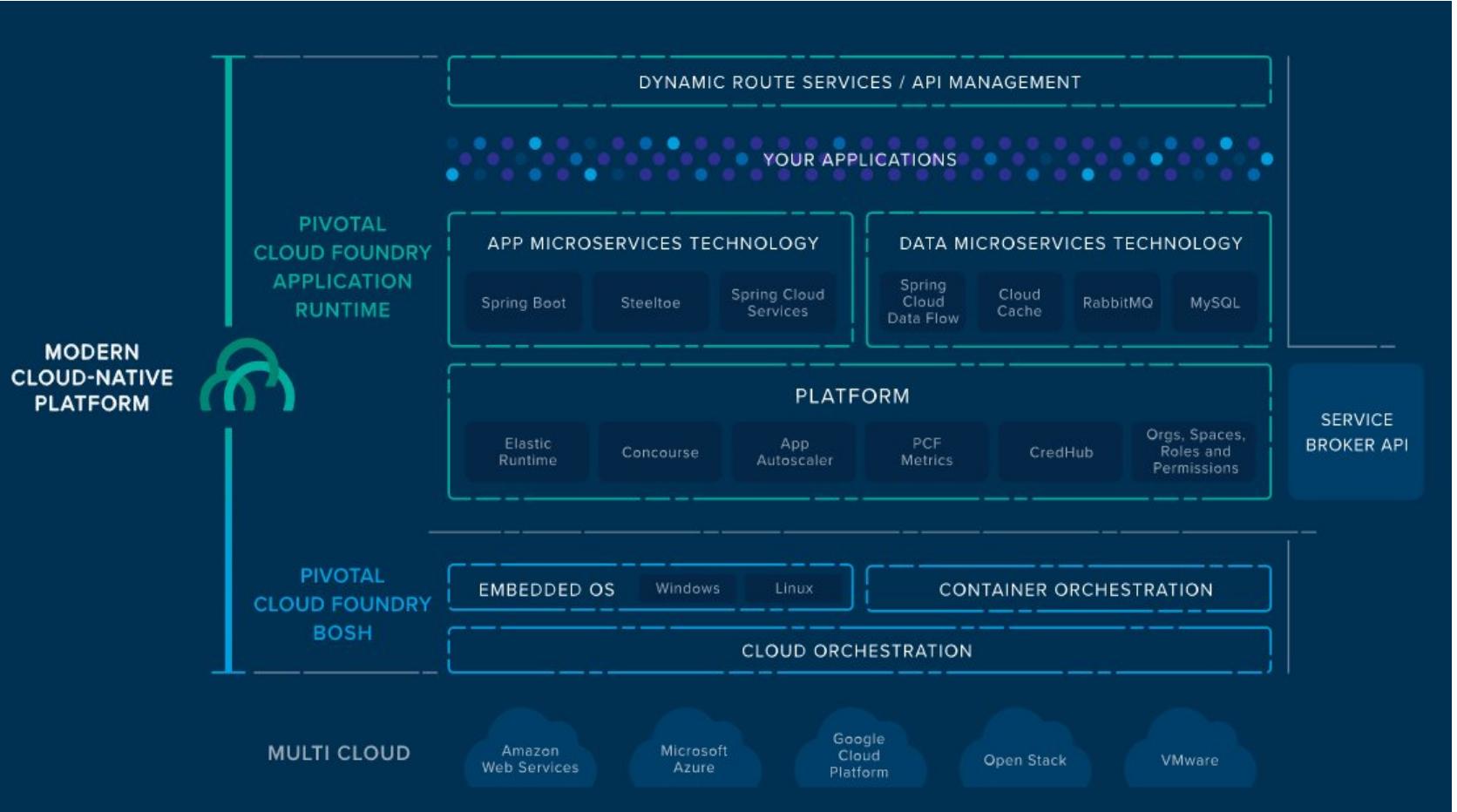


Splitting the monolith





Pivotal Cloud Foundry

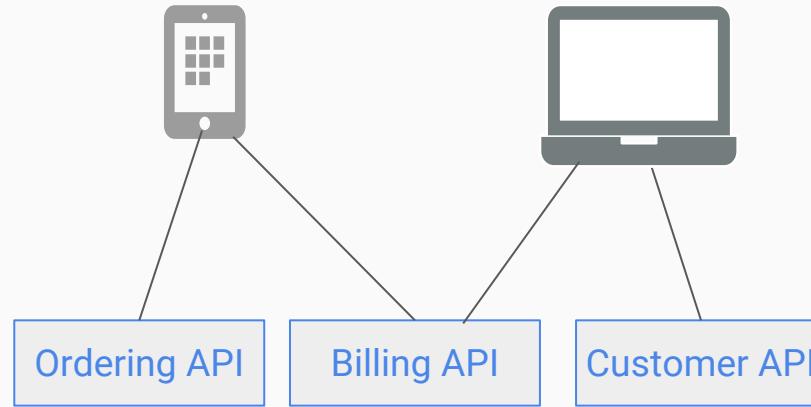


Why API Management?

APIs and Microservices are complementary



APIs shield consumers from Microservices Complexity



Consumers Shouldn't
Care About What's
Behind the API

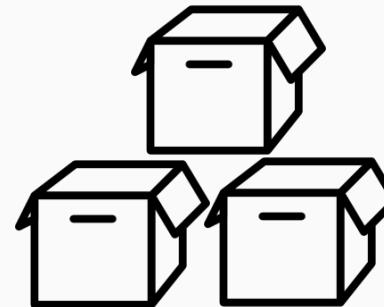
REST APIs - Enable easy consumption



CLOUD
enable independent scaling

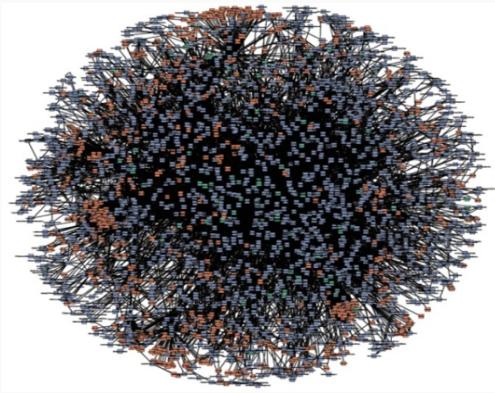


REST APIs
enable easy consumption



CONTAINERS
enable independent deployment

Microservices will fail without API Management



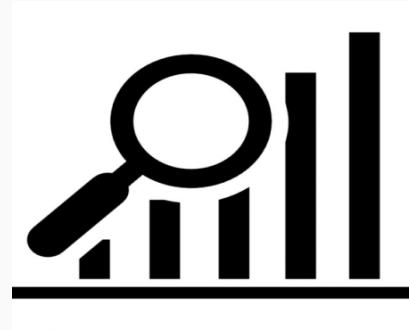
API SPRAWL

Difficult to discover & reuse



CYBER THREATS

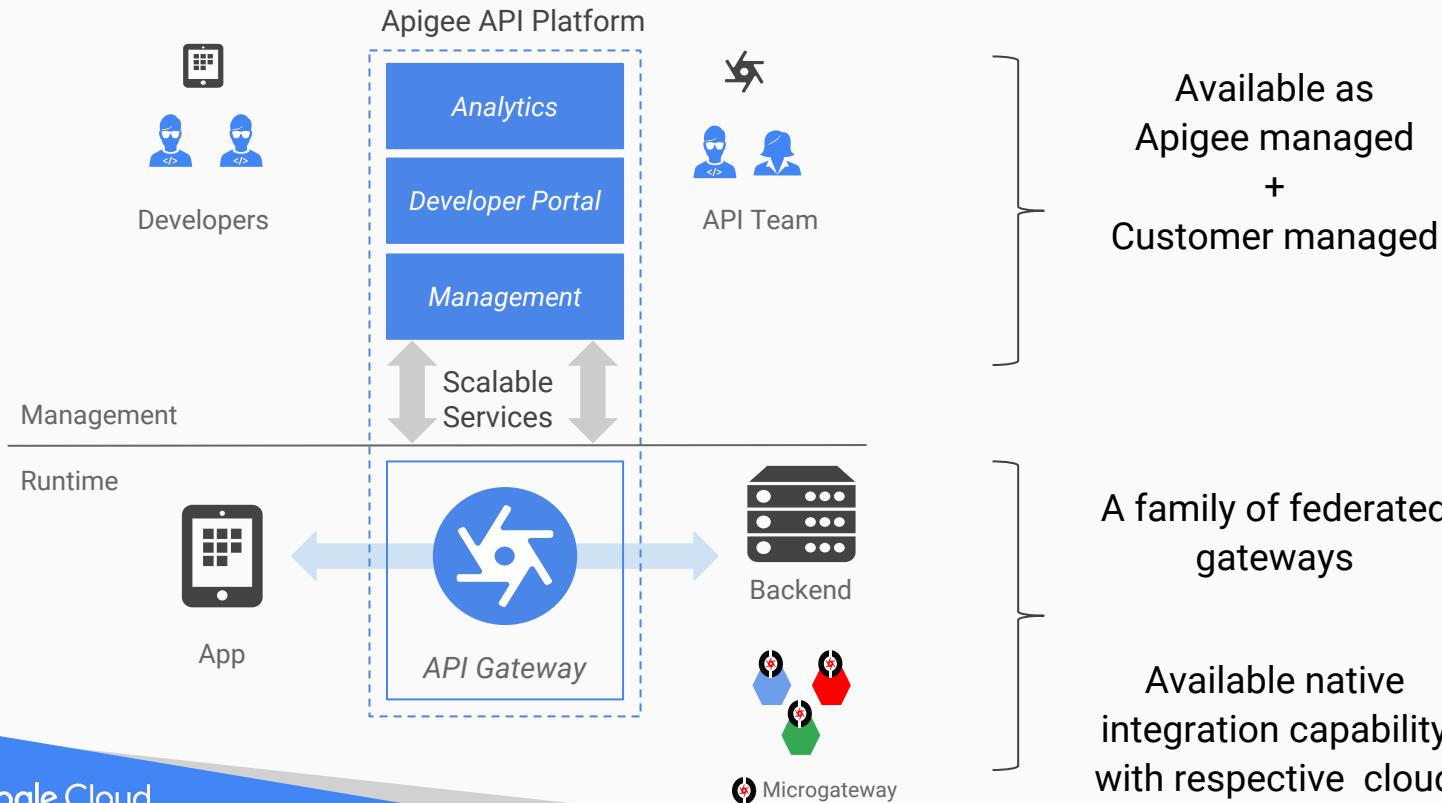
Limited built-in security



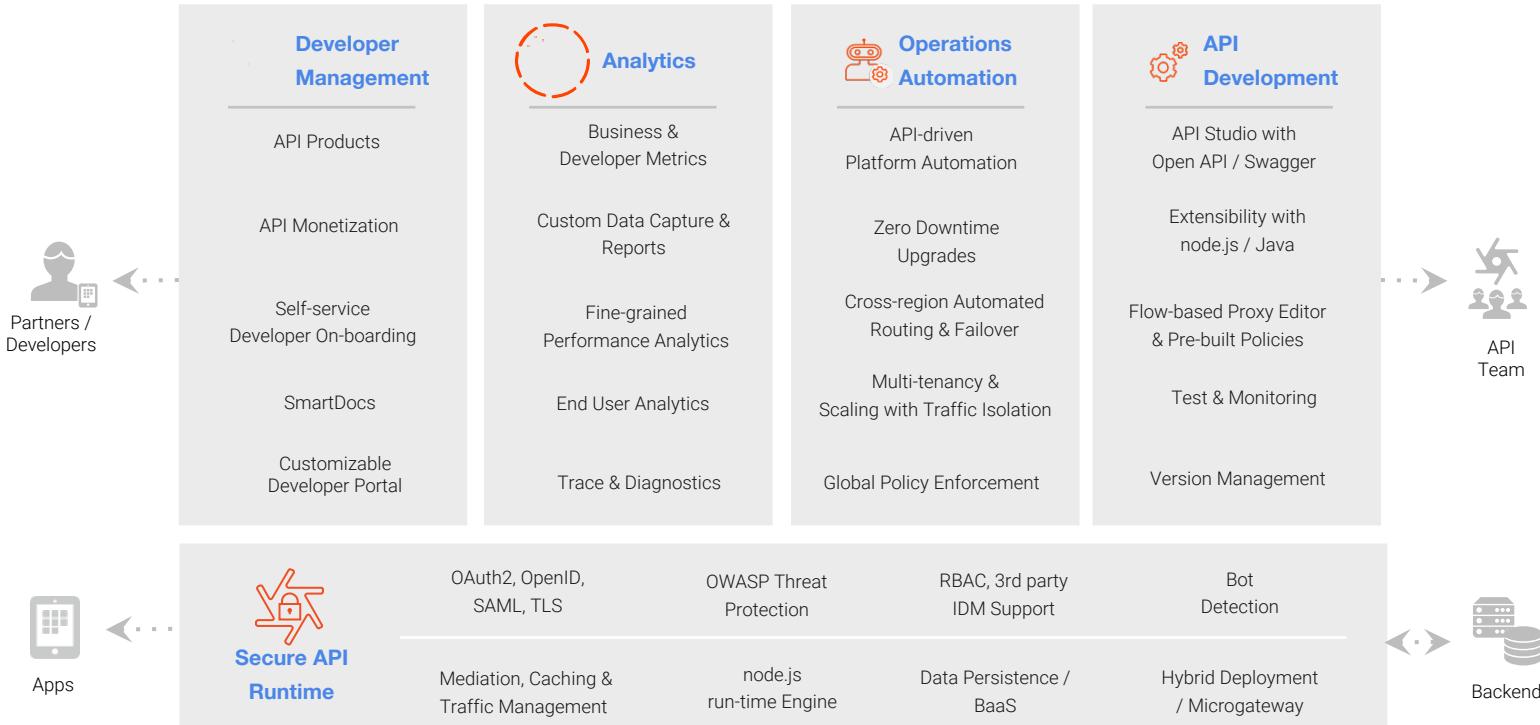
FLYING BLIND

Limited use & performance visibility

Technology powering the Apigee API Platform



Comprehensive API Management



For the Full API Lifecycle



Apigee provides the API layer for Microservices

Apigee's centralized gateway provides comprehensive API management and an API layer for your microservices



Apigee's microgateway provides security, throttling and analytics for individual microservices as well.

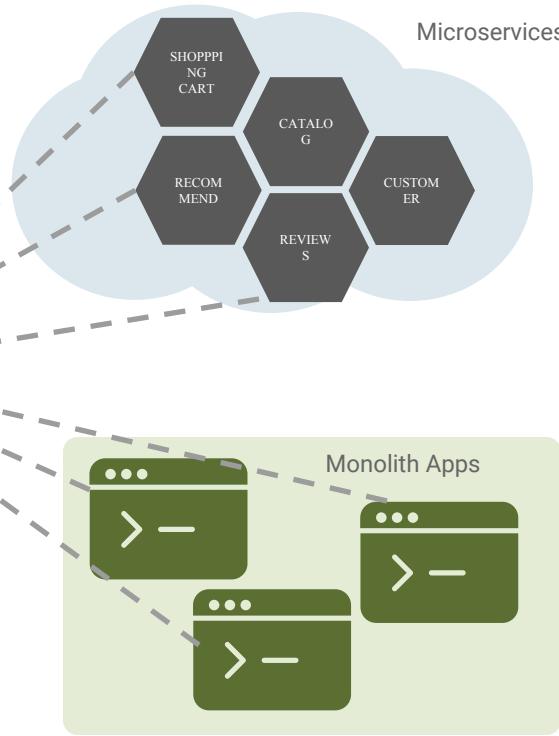
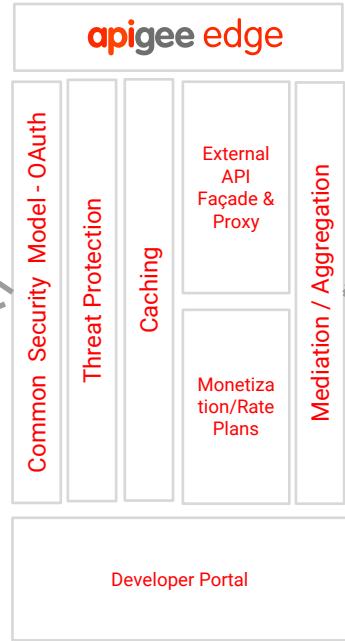
Use API management as you transition to microservices



Mobile & Web apps



Apigee authenticates, applies security policies and routes request to microservices. Also orchestrates to your legacy services/monolith apps



TRUSTPILOT

magazineluiza

Google Cloud

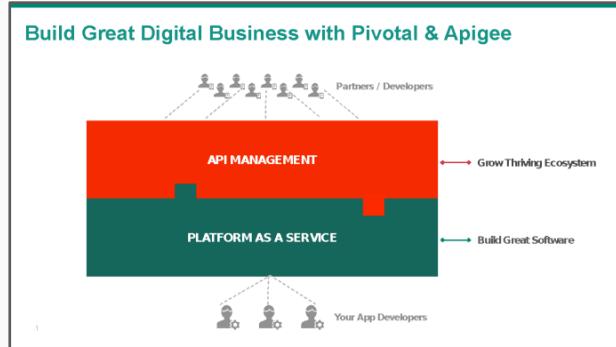
jcpenny

AUTODESK.

belly

Proprietary + Confidential

What we are doing together?



Announcing Apigee Edge Microgateway for Pivotal Cloud Foundry

by Prithpal Bhogil (@pbhogil)
DEC 07, 2016

Enterprises are adopting a variety of microservices stacks, including [Pivotal Cloud Foundry](#), [Docker](#), [Kubernetes](#), [Apache Mesos](#), and [Netflix OSS](#). Microservices are becoming increasingly popular with developers, but this brings with it several new challenges.

As each microservice can communicate with other microservices or can be accessed only through REST APIs, you'll have lots of APIs, making it tricky for developers to discover, understand, and consume them. Security becomes an issue, too: as most microservices will be deployed in the cloud, it's important that they're protected from cyber attacks at the API layer. Finally, once an application has been disaggregated into many microservices, it's challenging to gain visibility into the use and performance of the microservices APIs.

In short, microservices need to be managed (for more details on this, check out our eBook, "[Microservices Done Right](#)"). Although some microservices stacks offer a basic gateway, routing and basic traffic management capabilities, they do not provide full API management capabilities (security, robust traffic management, end-to-end visibility into API usage and adoption, and monetization capabilities). Microservices implementation will fail without a robust API management solution.

This is where our [Edge Microgateway](#) comes in. Edge Microgateway is a lightweight, secure, HTTP-based message processor designed especially for microservices. Its main job is to process requests and responses to and from backend services securely while asynchronously pushing valuable API execution data to Apigee Edge, where it's consumed by the Edge analytics system.

Apigee Edge Service Broker for PCF	AppDynamics Service Broker for PCF
Azuqua Platform Connector for PCF	Blue Medora Nozzle for PCF
Cloudsoft Service Broker for PCF (BETA)	Crunchy PostgreSQL for PCF
DataStax Cassandra Service Broker for PCF (BETA)	Dingo PostgreSQL for PCF (BETA)
Dyadic EKM Service Broker for PCF (BETA)	Dynatrace Service Broker for PCF
EDB Postgres Service Broker for PCF	Edge Installer for PCF

We've integrated our products...

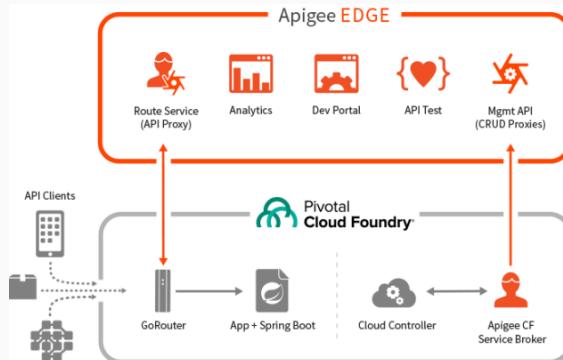
added support for Edge Microgateway...

released BOSH Installers for Apigee Edge

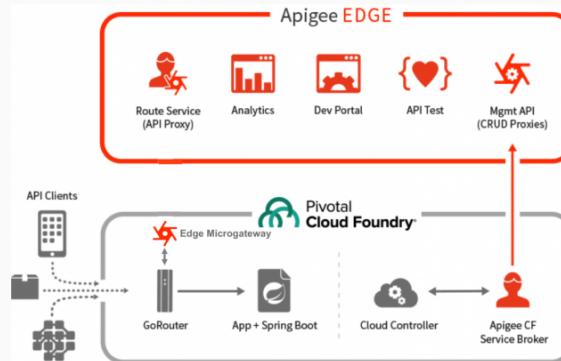
Integration Options - Flexibility & Choice

Route Services

Apigee Edge Enterprise

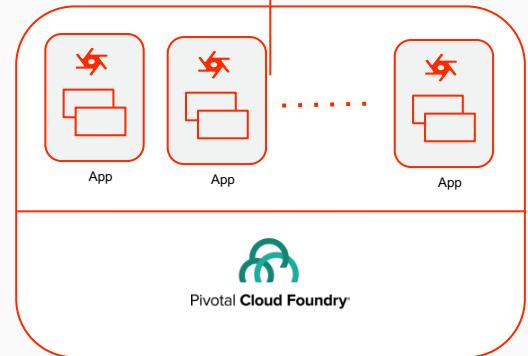


Apigee Edge Microgateway



CF Buildpack*

Apigee Edge Microgateway



* Edge Microgateway Buildpack available today as OSS on <https://github.com/swilliams11/edgemicro-decorator>.

Benefits for Developers & Operators

Developers

- Apply pre-built traffic management and security features to their app
- Apply pre-built pricing models to monetize their app
- Automatically expose their app (as services) to other developers
- Discover, test, and obtain access to other services (as APIs)

Operators

- Apply a set of security and traffic management features across projects
- Scale these features through PCF
- Obtain visibility into the usage and performance of APIs
- Install and manage Apigee Edge through BOSH

Accelerate Application Modernization

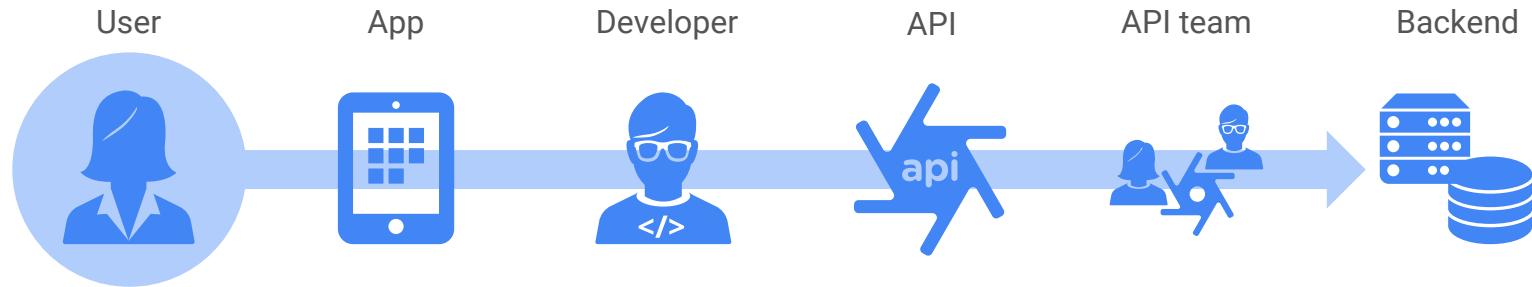
Ease Operational Concerns

Use Cases

Use Case	Microservices Development	Application Modernization	Enabling Ecosystems
Description	Apigee protects and enables authorized access to microservices	Apigee mitigates risk in app modernization/migration	Apigee enables APIs to be discovered and managed by app developers
Product	Apigee Edge Microgateway	Apigee Edge	Apigee Edge
Features	3 policies OOTB, specifically spike arrest, API key validation / OAuth, and quota limits Operational analytics Centralized API management	35+ policies OOTB, including transformation, mediation, & security Developer portal Operational analytics Centralized API management	35 policies OOTB, including transformation, mediation, & security Developer portal Operational analytics Centralized API management

Powering the “Digital Value Chain”

“Outside-in” – digital consumer view



Consuming digital products and digital experiences

Reduce time to develop connected experiences
Enable consumption of APIs

apigee edge

Reduce time to build better software
Reduce operational complexity

Pivotal CF™

Call to Action

If you are PCF Developer or Operator

Learn more about Apigee's Service Broker with Edge Microgateway [here](#)
Ask your PCF rep about Apigee's free Microservices Starter Kit offer
Ask your PCF rep for an introduction to Apigee Sales

If you are new to Apigee

Visit www.apigee.com and sign up for a trial account
Learn about Apigee's integrations with Pivotal [here](#)

If you are new to Pivotal

Visit www.pivotal.io to learn how to develop and deploy software with cloud native techniques
Ask your Apigee rep for an introduction to Pivotal Sales.

Thank You