apigee

# Which App Modernization Pattern is Right for You?

Shaun Anderson, App TX Practice Lead, Pivotal
Prithpal Bhogill, Product Management, Google Cloud

Google Cloud

***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

Shaun Anderson
AppTX Practice Lead

# Agenda

Why modernize?

Strangling the Monolith

Why API Management?

Apigee and Pivotal Cloud Foundry

Customer Benefits

Call to Action

# What is Pivotal Cloud Foundry?

# Overview of Cloud Foundry


Pivotal Cloud Foundry is a **platform** for developing and running cloud applications

# Cloud Foundry Principles


Developer Productivity

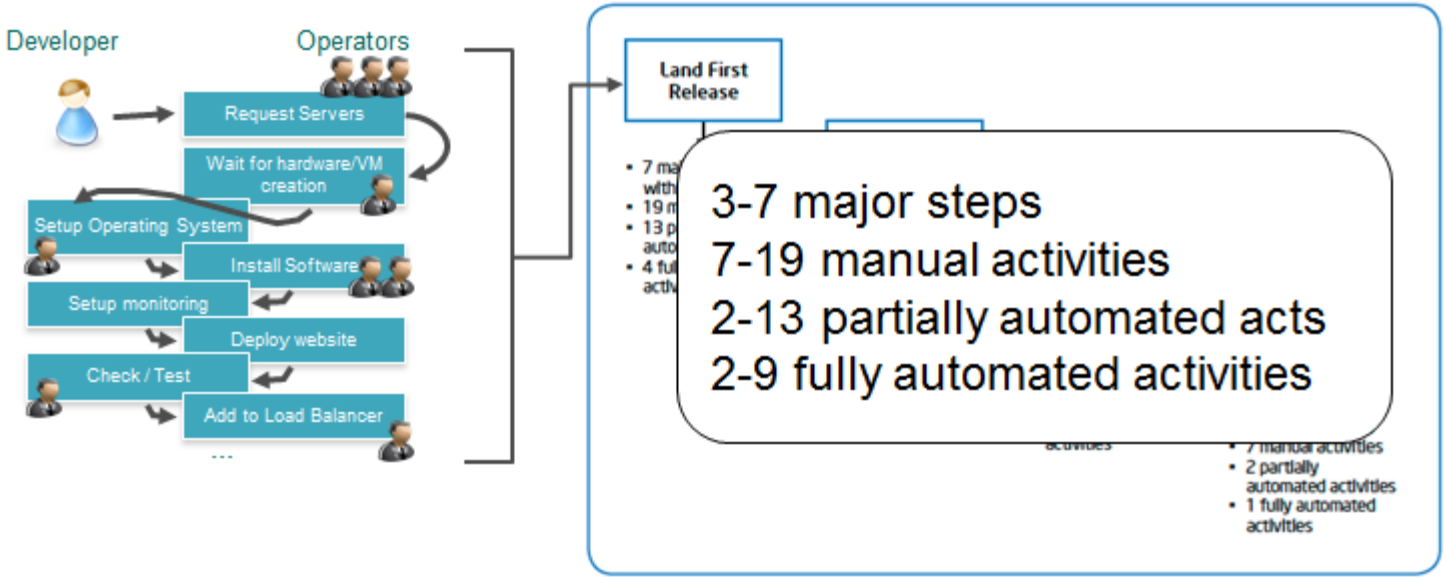
Operational Efficiency


Enhanced Security Posture

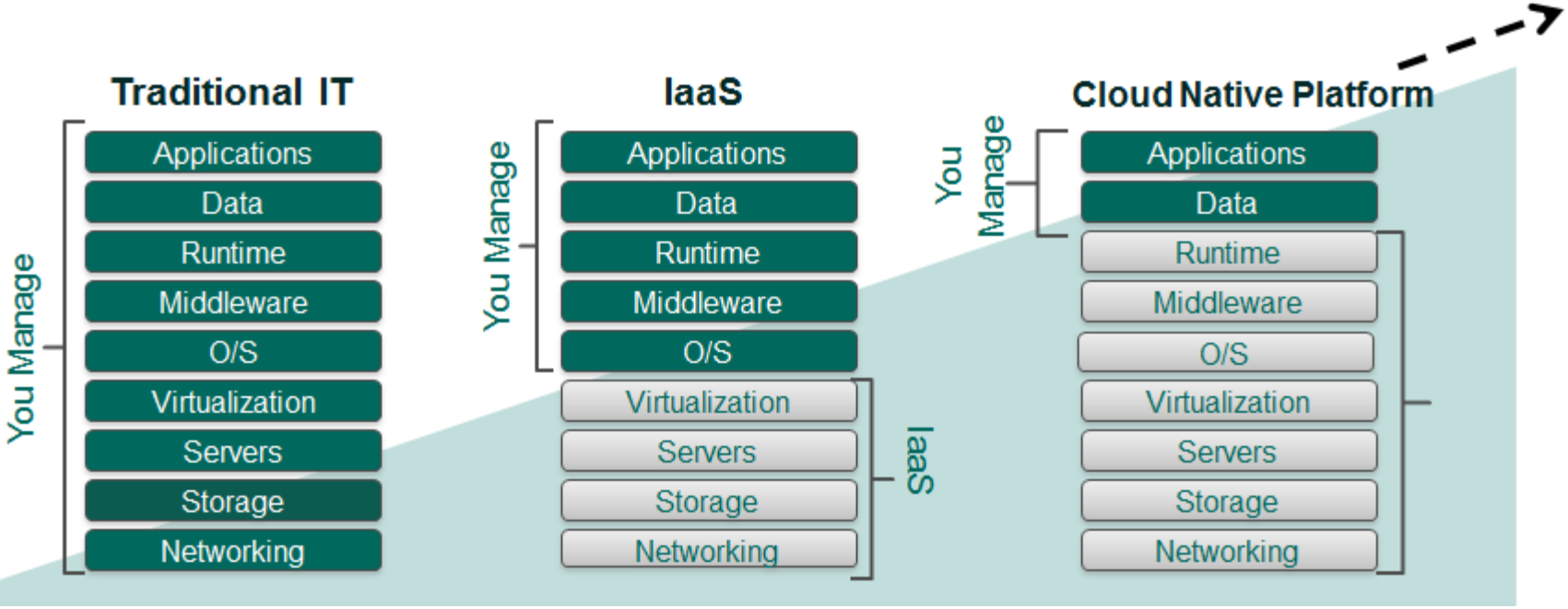
# Traditional App Lifecycle: Lengthy & Complex

Traditional Deployment:
8+ manual steps, developers out of the workflow

Traditional App Lifecycle:
Each phase compounds time and complexity, sacrificing agility



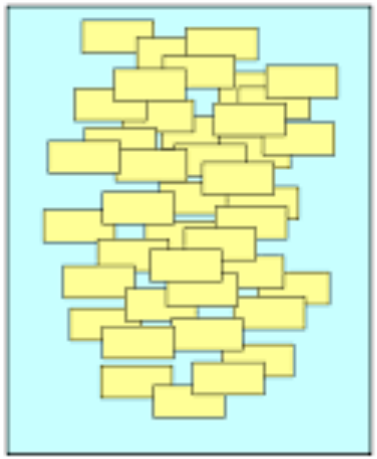
# The Power of Cloud Native Platform



# Why Modernize?

# Why Monoliths Can Be Bad

- Slow velocity and release cycle
- Knowledge is compartmentalized
- The code base is just too large for any one person to fully comprehend
- Centralized authority and change management slows progress (DBA, Ops)
- Inefficient infrastructure utilization
- Reliance on expensive software
- Non-standard toolsets



Bad monolith

Good microservices

# Why Replatform / Modernize

- Lower CapEx and OpEx
  - Moving away from full-blown application servers that are expensive to license, operate, and administrate has been a consistent market trend since 2004
- Increased scalability, provider portability
  - Vertical scalability is inherently limited. Cloud platforms can horizontally scale up or down in seconds, dynamically—even auto scale
- Optimize hardware resource usage, cost
  - Containerizing and virtualizing applications increases deployment density on the hardware or IaaS, ensuring maximum utilization of system resources, and deallocation or reallocation when idle
- Increased developer productivity without losing operations controls
  - When developers can self service, instantly provisioning an environment that uses platform-wide access controls and audit trails, it's a win for operations teams that can set consumption parameters
- Enabling DevOps with common management and monitoring tools
  - Modern clouds provide agentless health management that streams an integrated, near real-time view of key application, services, and platform metrics. Developers and operators gain a shared understanding of the same system health and availability data in cloud environments.

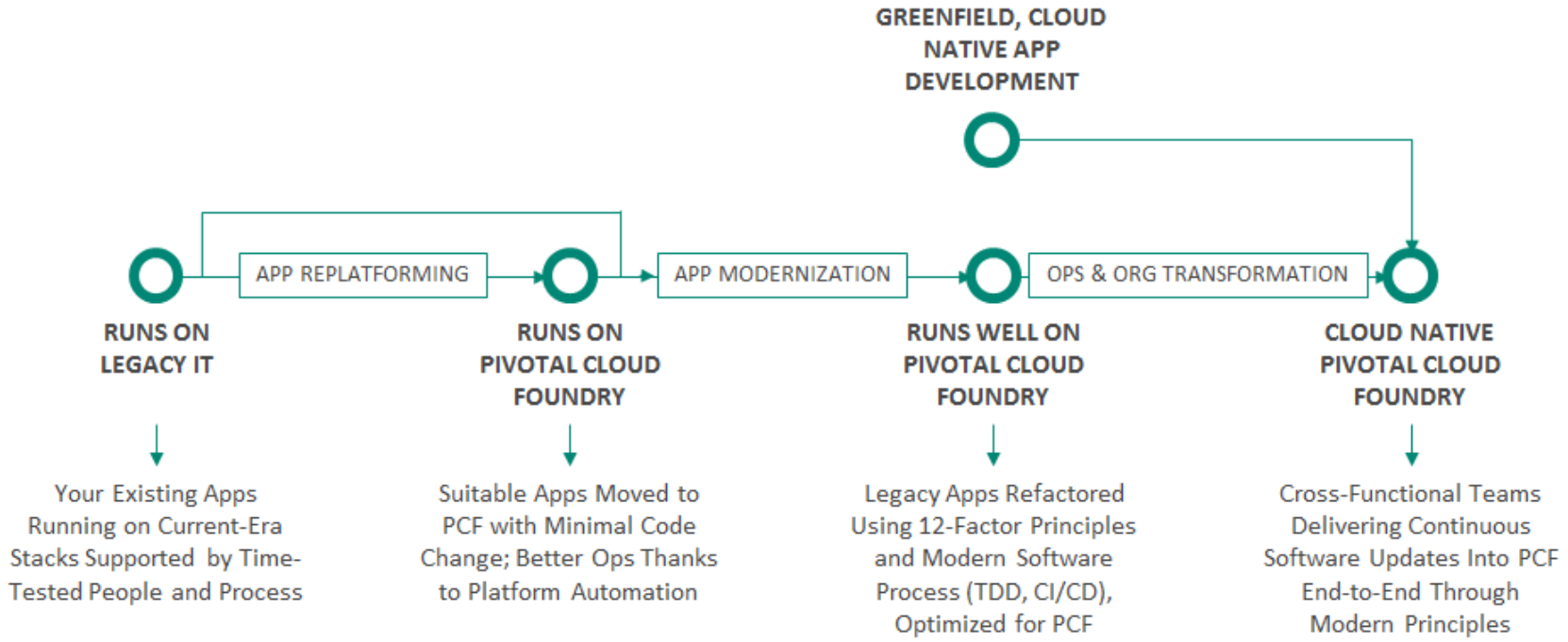
# Replatform vs Modernize (definitions)

**Replatforming** involves upgrading an application from its existing platform and adhering to the minimum possible Twelve factors to get it to run on the cloud, while preserving existing functionality.

**Modernization** is the process of organizing existing functionality into applications, microservices and development teams around business capabilities and domains. A main goal of modernization is to decompose an existing monolith into small, 12 Factor services.

*Digital transformation is how good companies become great. They invest in becoming software-driven to reduce time-to-value windows. They choose a meaningful project, combine modern methodologies and technologies to build high-quality smart applications that regularly deliver real-time insights, then continually iterate to ensure exceptional customer experiences.*

# The Cloud Native Application Journey



# Strangling the Monolith - Approach

Top-Down: Decomposition from a business perspective

- Event Storming
- Snap Analysis
- Boris (Spider) Diagram
- DDD
- Find Seams
- Iterate

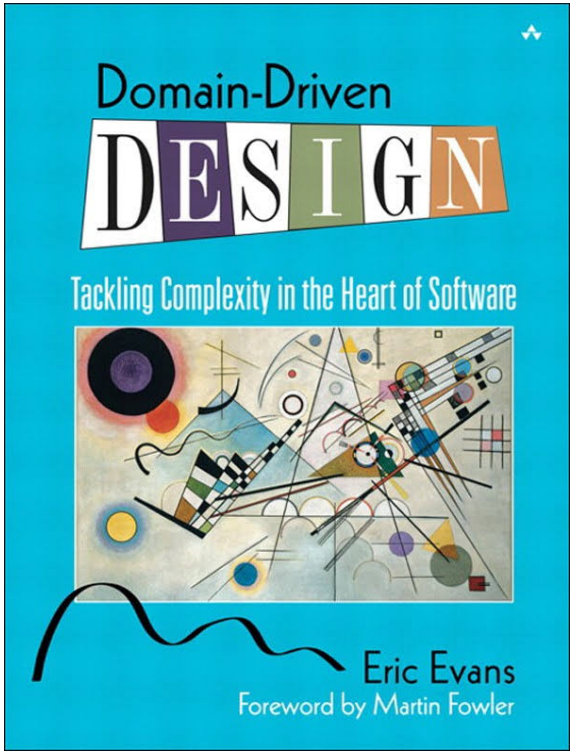
Bottom-Up: Decomposition driven by current pain points of the monolith

- Reverse engineering
- Code tracing
- Source text parsing and analytics

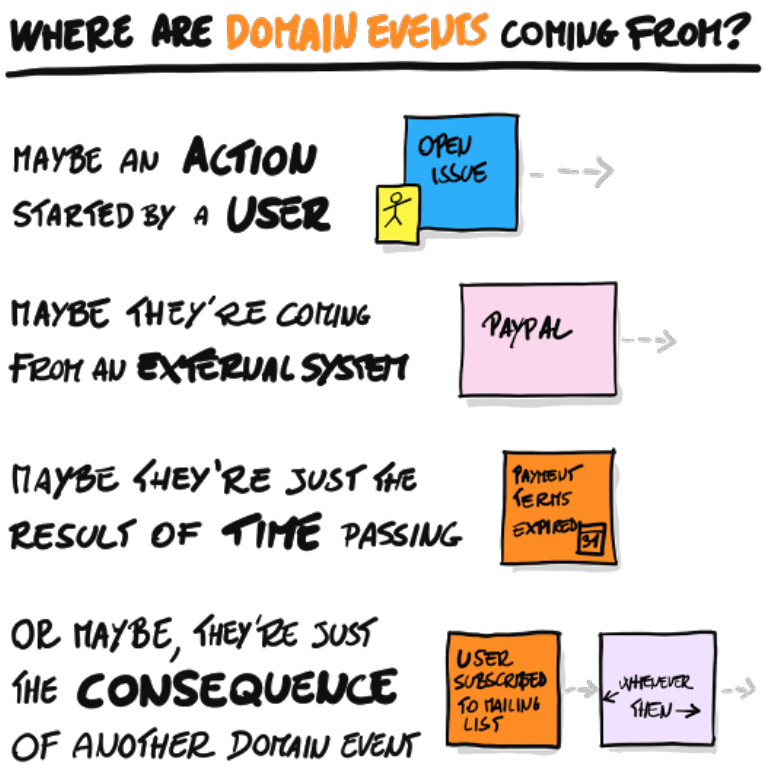
# Domain Driven Design (DDD)

**Wikipedia**: DDD is an approach to software development for complex needs by connecting the implementation to an evolving model.

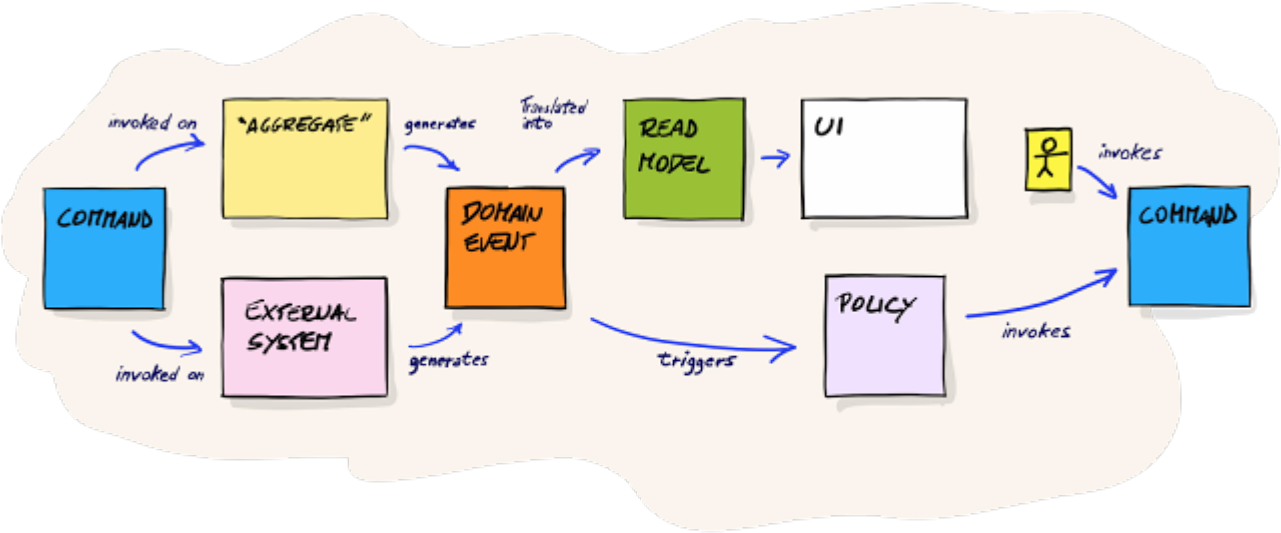
**Domain**: “A sphere of knowledge or activity.”



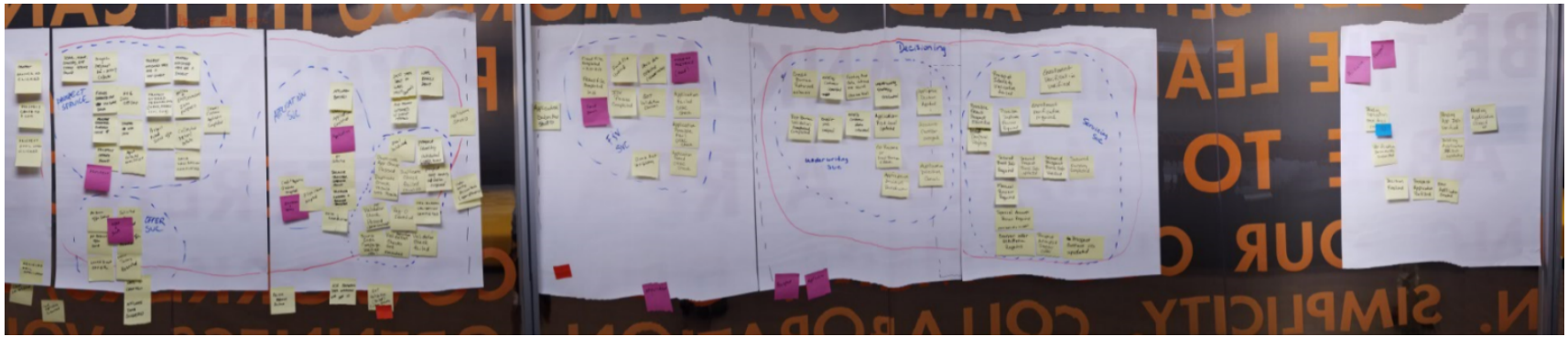
# Event Storming



# Event Storming

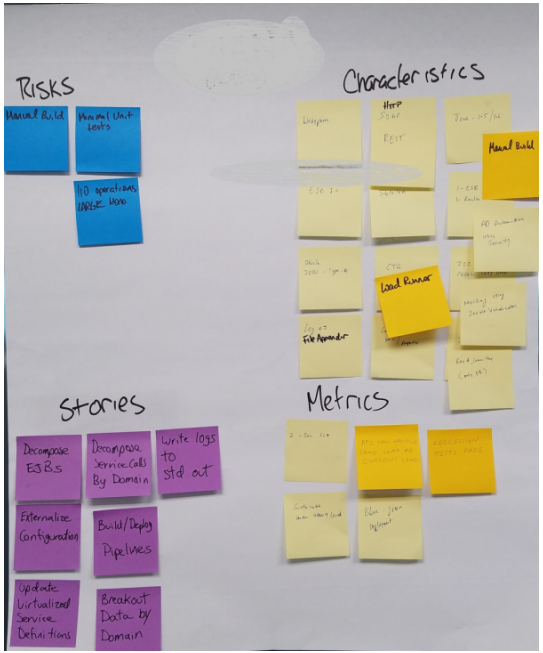


# Event Storming



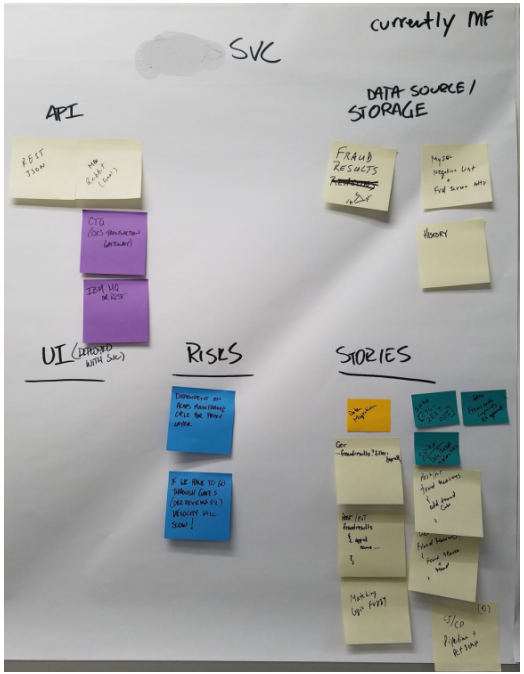
# Snap Analysis

Designed to make it easy to understand an application's complexity and suitability for replatforming to Cloud Foundry

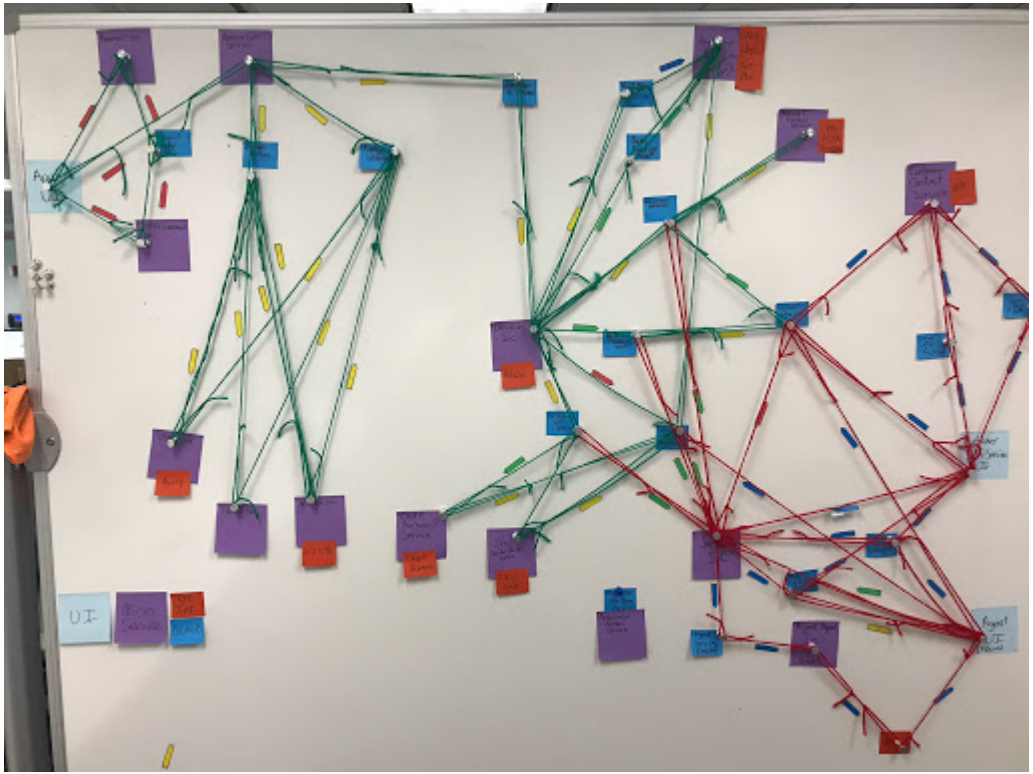


# Enhanced Snap Analysis

Used to describe a modernized service -- usually this service consists of functionality from one or more monoliths.



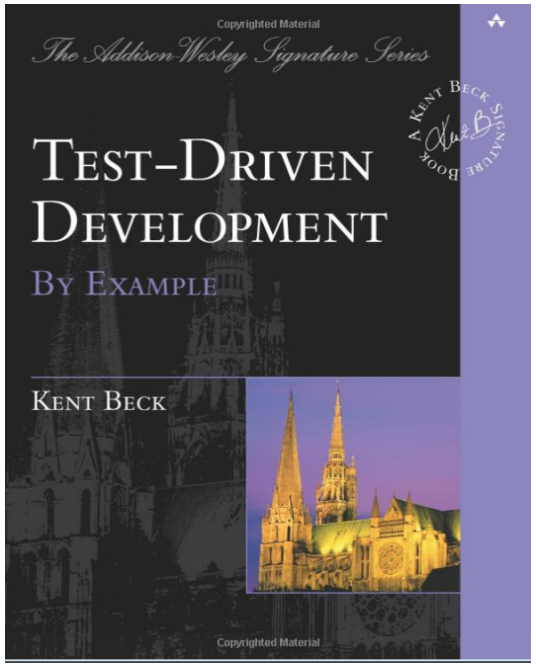
# Spider Diagrams



# Test Driven Development (TDD)

Test-Driven Development (TDD) is a technique for building software that guides software development by writing tests. It was developed by Kent Beck in the late 1990's as part of Extreme Programming. In essence you follow three simple steps repeatedly:

Write a test for the next bit of functionality you want to add.
Write the functional code until the test passes.
Refactor both new and old code to make it well structured.

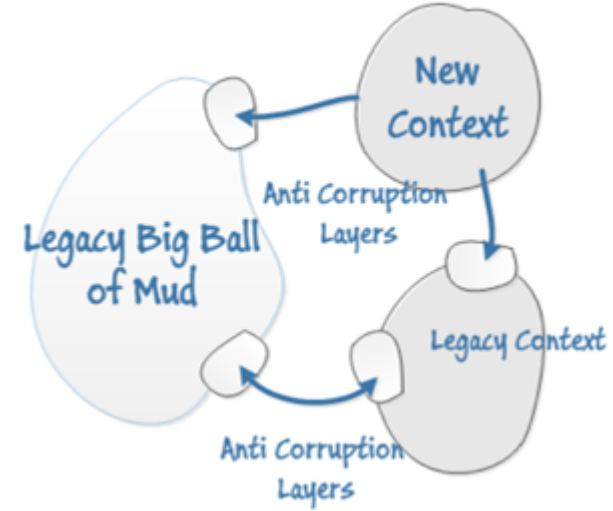


# Finding Seams

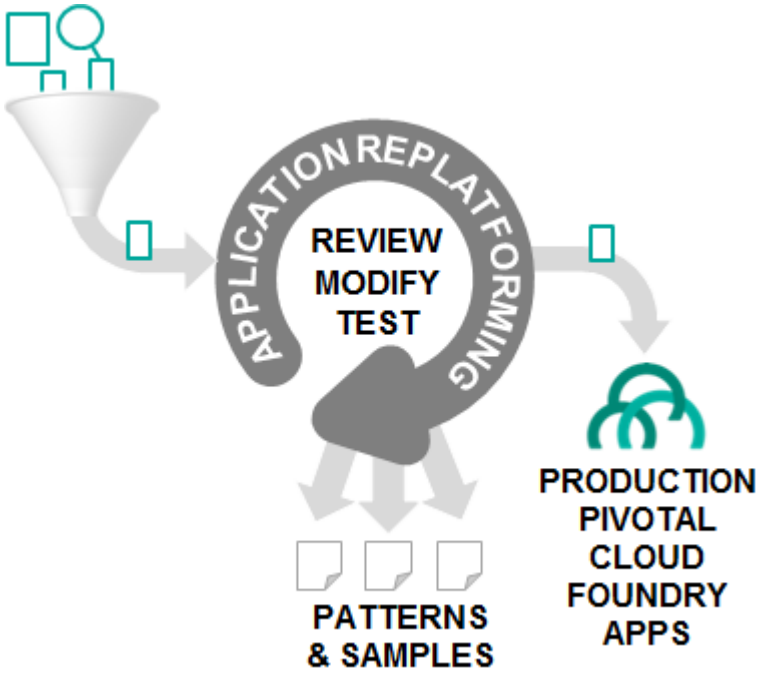
**The Goal:** Find areas of an existing monolith that can be pulled apart and reorganized **without impacting the functionality of the existing system**.

Control Patterns

- Event shunting / starving
- Decorator
- Bridge
- Router
- Proxy
- Facade
- Edge Gateway - Apigee



# Iterate vs Big Bang rewrite

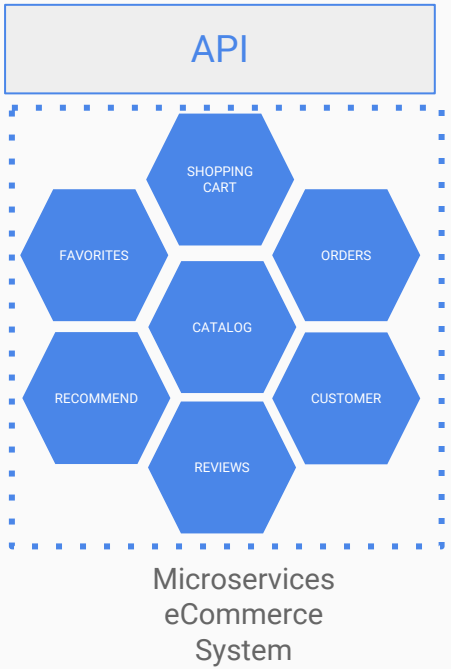



**Iterate**

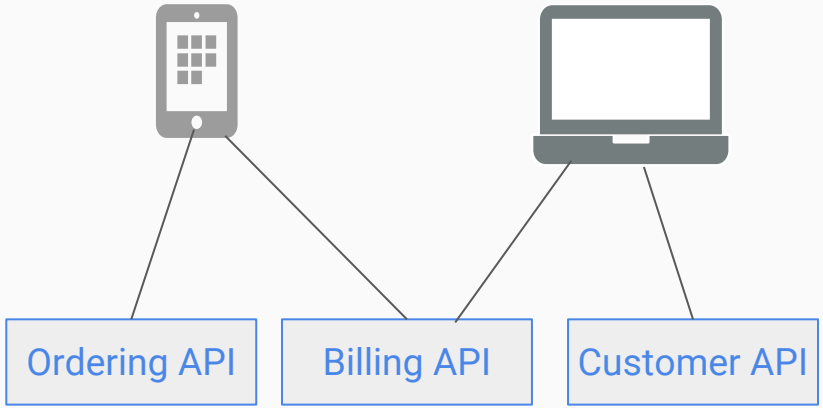
- Succeed Fast / Fail Fast
- Allows for innovation
- Conforms to changing requirements
- Allows for dynamic and balanced teams
- Pairing enables Dev Team Mitosis

# Why API Management?

# APIs and Microservices are complementary

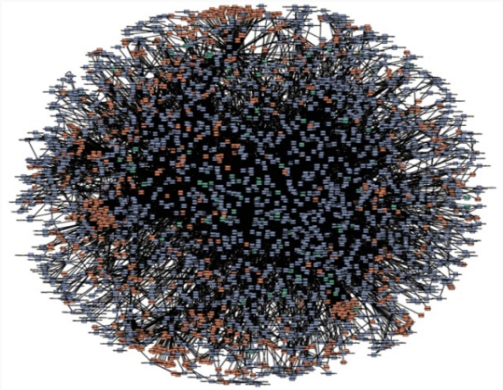



# APIs shield consumers from Microservices Complexity




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

# Microservices will fail without API Management



## API SPRAWL

Difficult to discover & reuse

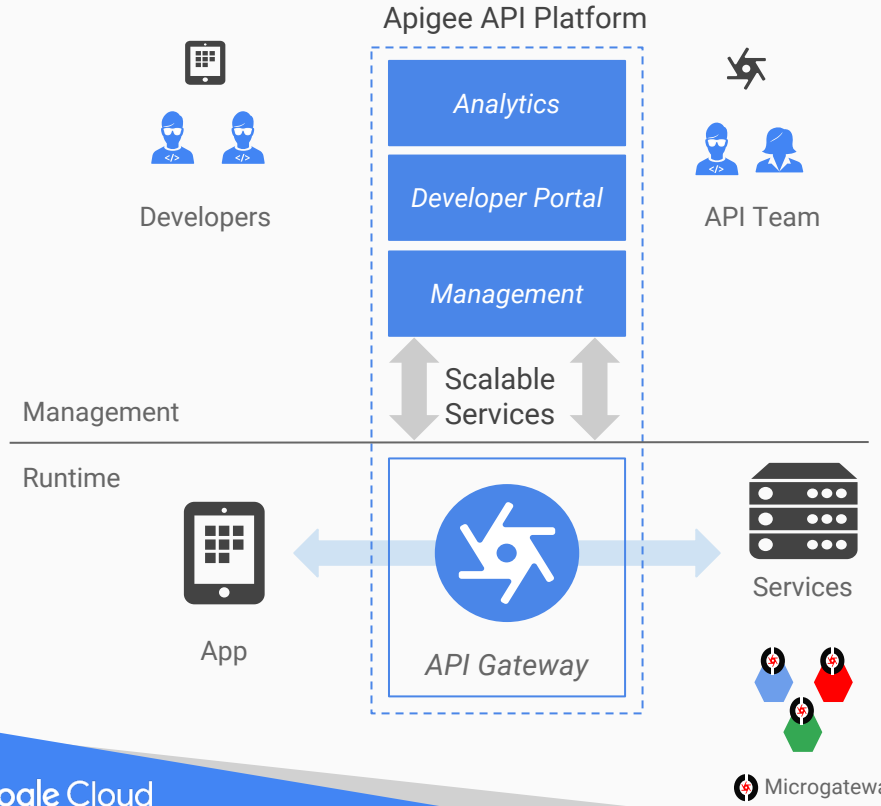
## CYBER THREATS

Limited built-in security

## INSUFFICIENT VISIBILITY

Limited use & performance visibility

# Technology powering the Apigee API Platform

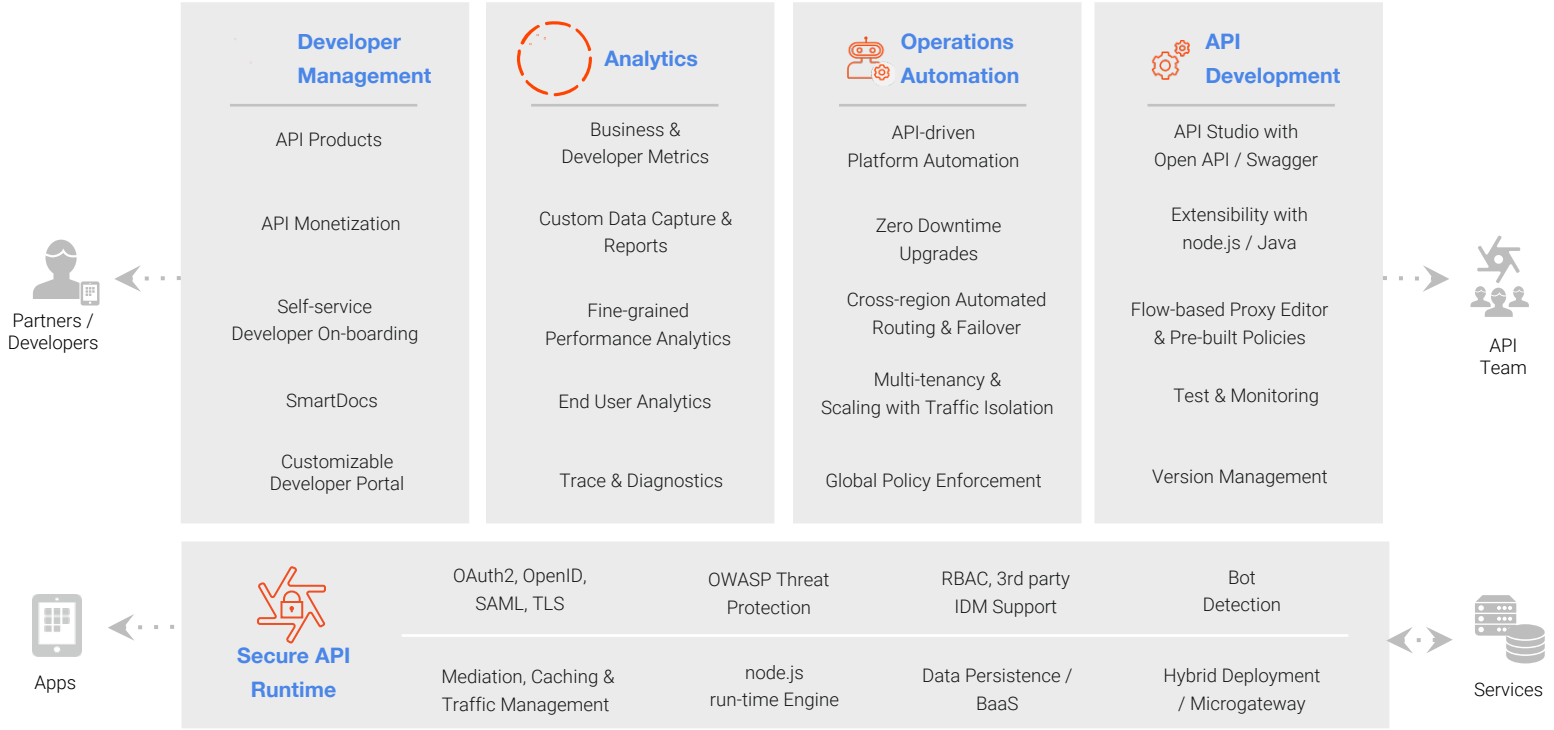



Available as
Apigee managed
+
Customer managed

A family of federated gateways

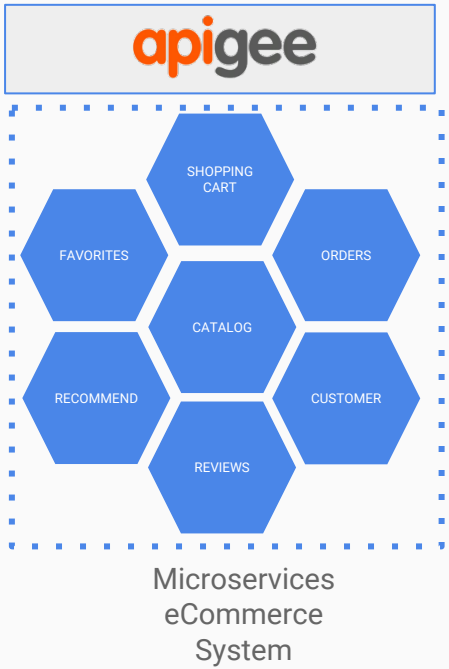
Available native integration capability with respective cloud

# Comprehensive API Management



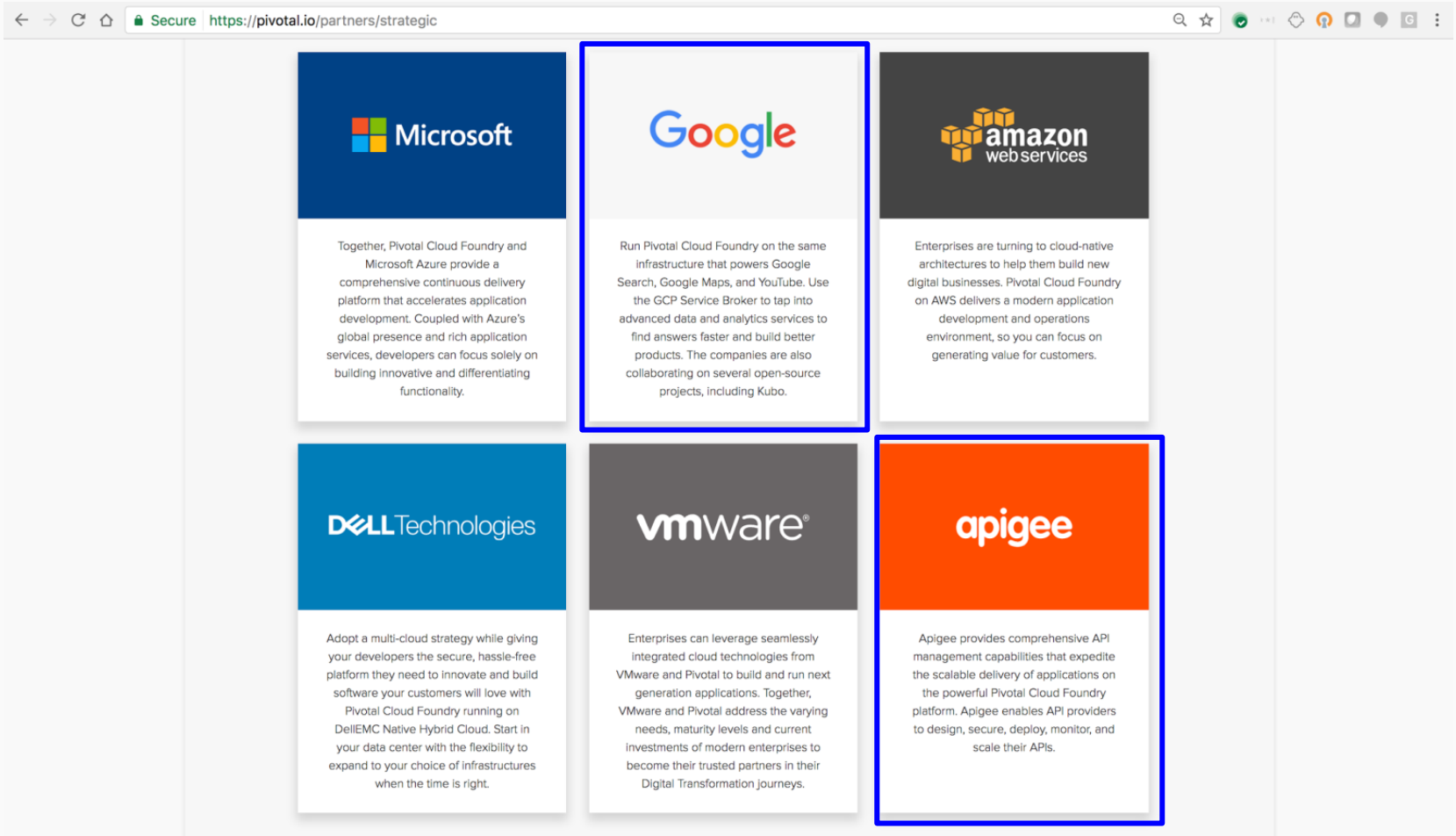
# Apigee provides the API layer for Microservices

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





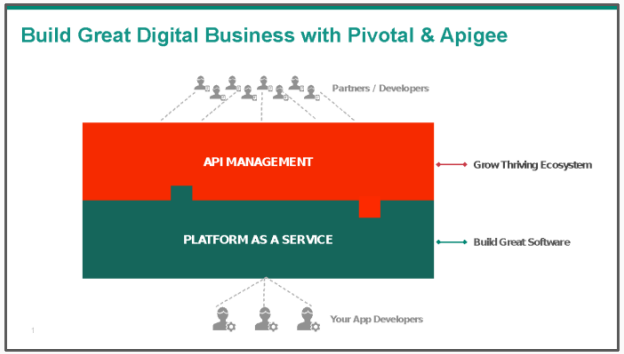
Apigee's microgateway provides security, throttling and analytics for individual microservices as well.


Secure https://pivotal.io/partners/strategic
Microsoft
Together, Pivotal Cloud Foundry and Microsoft Azure provide a comprehensive continuous delivery platform that accelerates application development. Coupled with Azure's global presence and rich application services, developers can focus solely on building innovative and differentiating functionality.
Google
Run Pivotal Cloud Foundry on the same infrastructure that powers Google Search, Google Maps, and YouTube. Use the GCP Service Broker to tap into advanced data and analytics services to find answers faster and build better products. The companies are also collaborating on several open-source projects, including Kubo.
amazon web services
Enterprises are turning to cloud-native architectures to help them build new digital businesses. Pivotal Cloud Foundry on AWS delivers a modern application development and operations environment, so you can focus on generating value for customers.
DELL Technologies
Adopt a multi-cloud strategy while giving your developers the secure, hassle-free platform they need to innovate and build software your customers will love with Pivotal Cloud Foundry running on DellEMC Native Hybrid Cloud. Start in your data center with the flexibility to expand to your choice of infrastructures when the time is right.
vmware
Enterprises can leverage seamlessly integrated cloud technologies from VMware and Pivotal to build and run next generation applications. Together, VMware and Pivotal address the varying needs, maturity levels and current investments of modern enterprises to become their trusted partners in their Digital Transformation journeys.
apigee
Apigee provides comprehensive API management capabilities that expedite the scalable delivery of applications on the powerful Pivotal Cloud Foundry platform. Apigee enables API providers to design, secure, deploy, monitor, and scale their APIs.

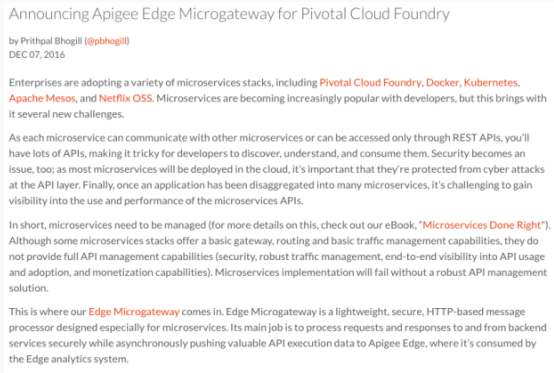
# What we are doing together?

apigee

Pivotal Cloud Foundry




We've integrated our products…



Announcing Apigee Edge Microgateway for Pivotal Cloud Foundry

by Prithpal Bhogill (@pbhogill)
DEC 07, 2016

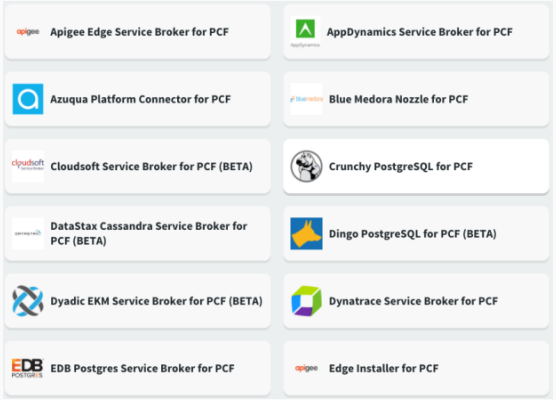
Enterprises are adopting a variety of microservices stacks, including Pivotal Cloud Foundry, Docker, Kubernetes, Apache Mesos, and Netflix OSS. Microservices are becoming increasingly popular with developers, but this brings with it several new challenges.

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.

added support for Edge Microgateway...



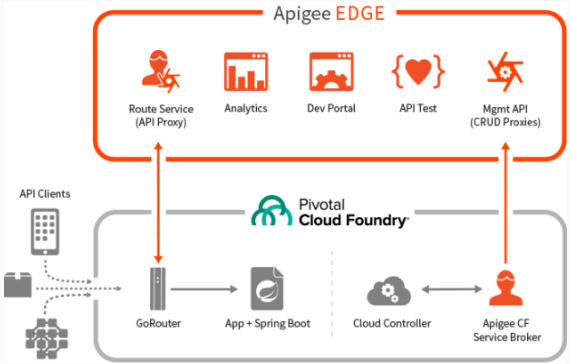

released BOSH Installers for Apigee Edge

# Integration Options - Flexibility & Choice

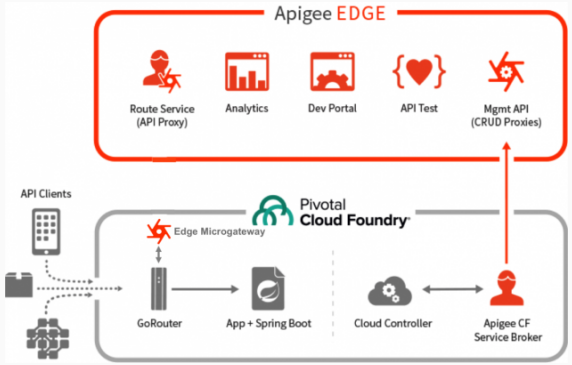
**Route Services**

**CF Meta Buildpack***

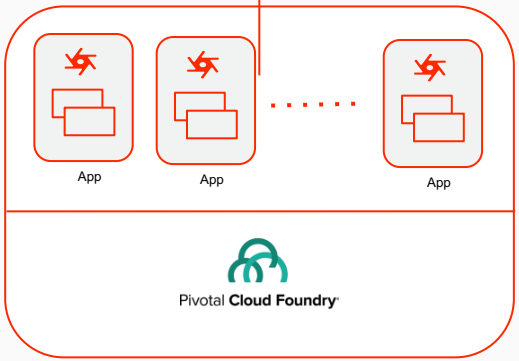
Apigee Edge Enterprise




Apigee Edge Microgateway

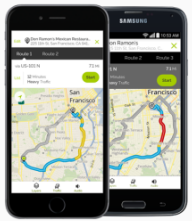



Apigee Edge Microgateway




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

# Application Modernization

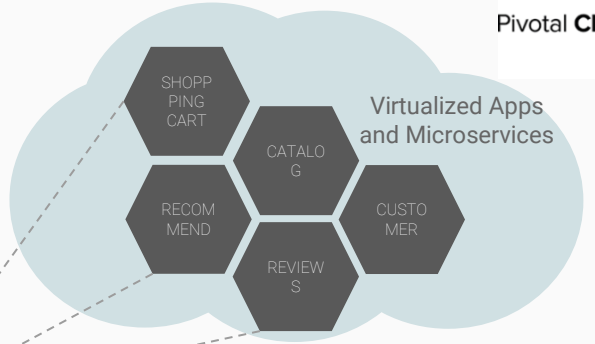


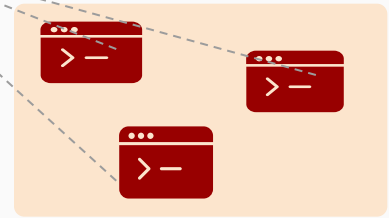
Mobile & Web apps

apigee edge

Routing
Security
Analytics

Pivotal Cloud Foundry



On-Premises, Legacy Apps

**Mitigate migration risk** with automated routing of API requests

**Monitor performance & usage** of both legacy and modern services

**Secure your services** from cyber threats and traffic spikes

# 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)

**Accelerate Application Modernization**

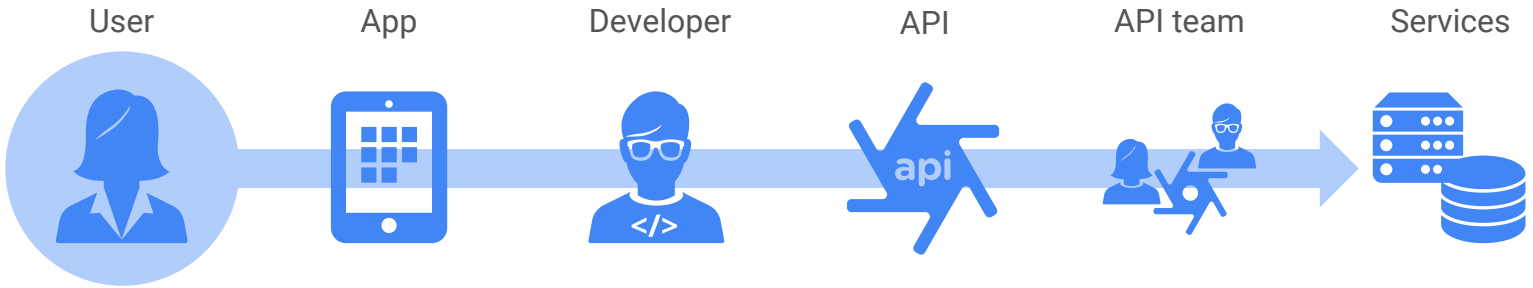
### Operators

- 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

**Ease Operational Concerns**

# 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 Cloud Foundry

Google Cloud

# Key Takeaways

Modernizing applications can have a huge ROI

It doesn’t have to be hard - start small

APIs and Microservices are complementary

Use API Management as you transition into Microservices

Enterprise customers are using this approach successfully

# Appendix

# Call to Action

| | |
|---|---|
| If you are PCF Developer or Operator | Learn more about Apigee's Service Broker with Edge Microgateway here<br>Ask your PCF rep about Apigee's free Microservices Starter Kit offer<br>Ask you 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<br>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<br>Ask your Apigee rep for an introduction to Pivotal Sales. |

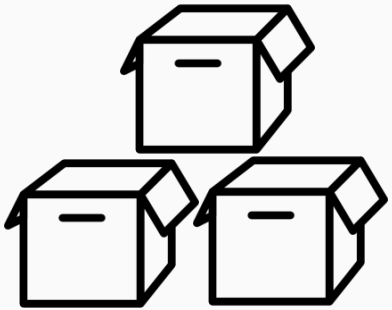
# REST APIs - Enable easy consumption

REST APIs

enable easy consumption

CLOUD

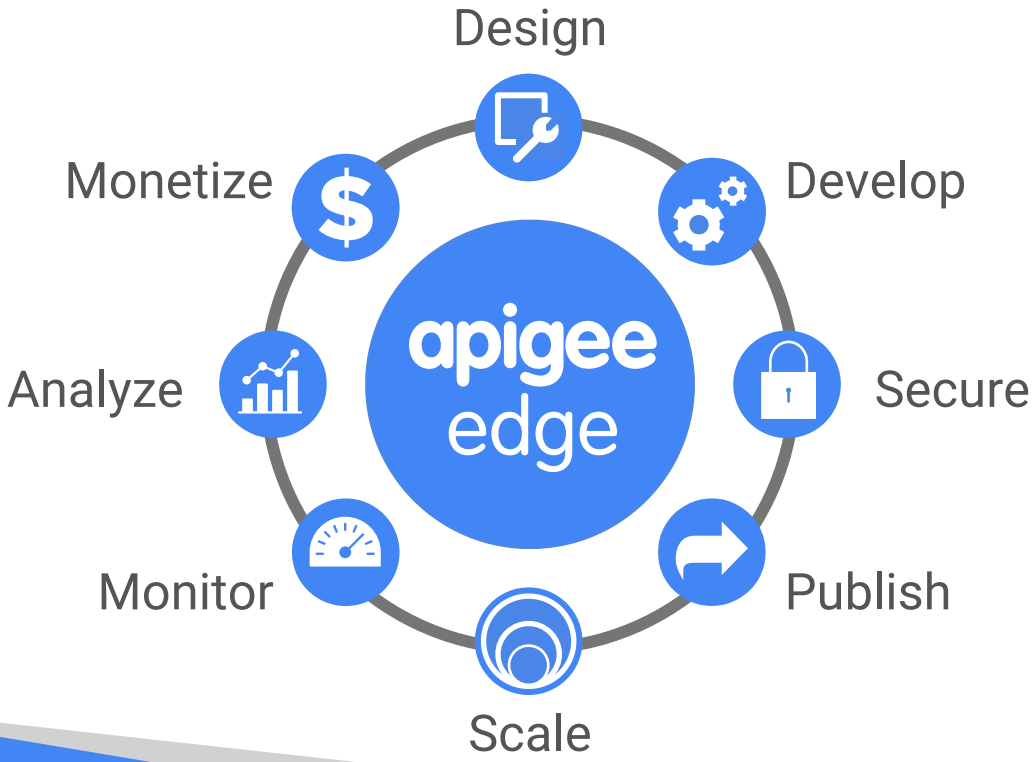
enable independent scaling



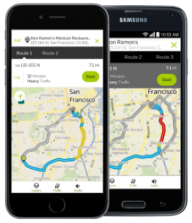
CONTAINERS

enable independent deployment

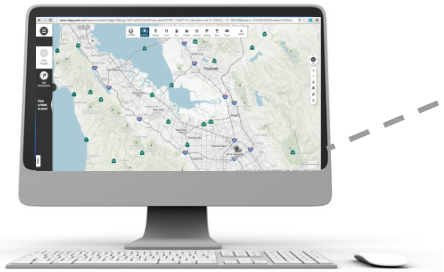
# For the Full API Lifecycle



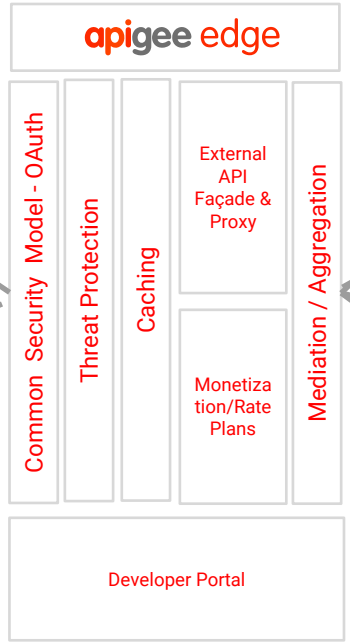
# Use API management as you transition to microservices

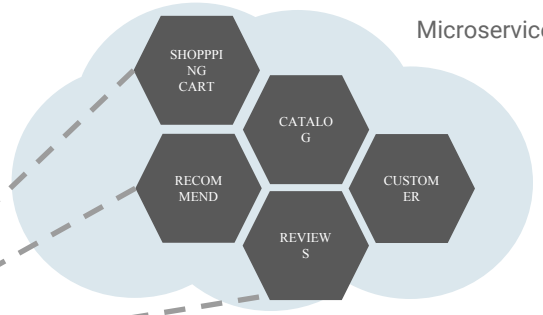


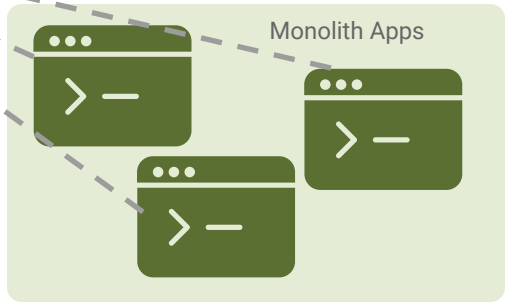
Mobile & Web apps



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




On-Premises or Private Cloud

TRUSTPILOT

magazineluiza

jcpenney

AUTODESK.

belly

Google Cloud