Challenges of Going Cloud Native in the **Enterprise**

DevOps Enterprise
October 2014

Jason R McGee
IBM Fellow
CTO, Cloud Services

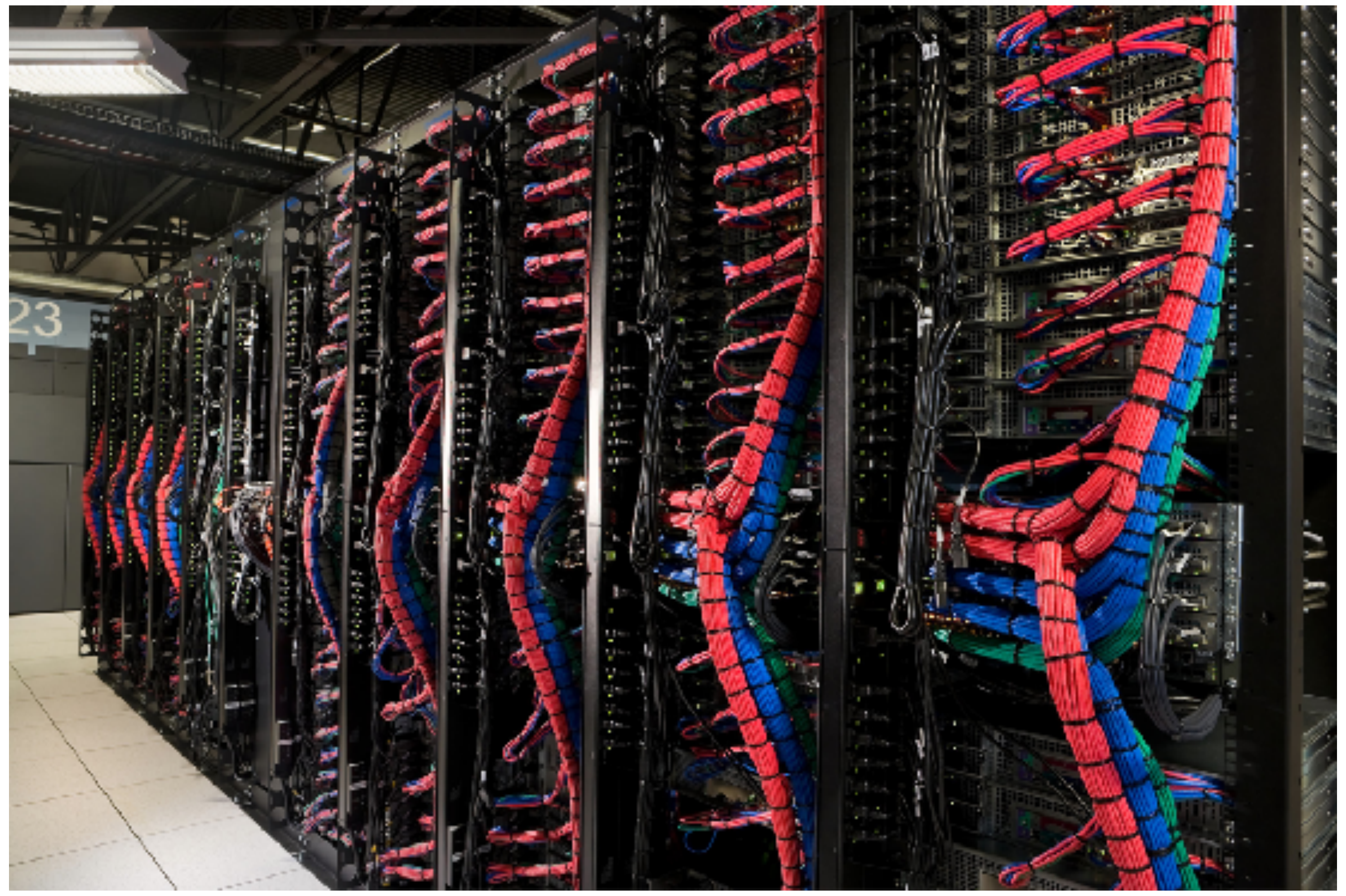
What are we going to talk about?

This talk will discuss some of the challenges faced taking built in the datacenter applications and architectures to the cloud. Challenges will be presented not only with regards to continuous delivery and devops, but also high availability/automatic recovery, elastic/web scale, operational visibility, and security/compliance.

The talk will discuss how IBM’s technologies as well as cloud native open source technology (such as OpenStack, CloudFoundry, NetflixOSS and Docker) have powered our evolution in a way that allows IBM to host our own IBM public cloud services with operational excellence.

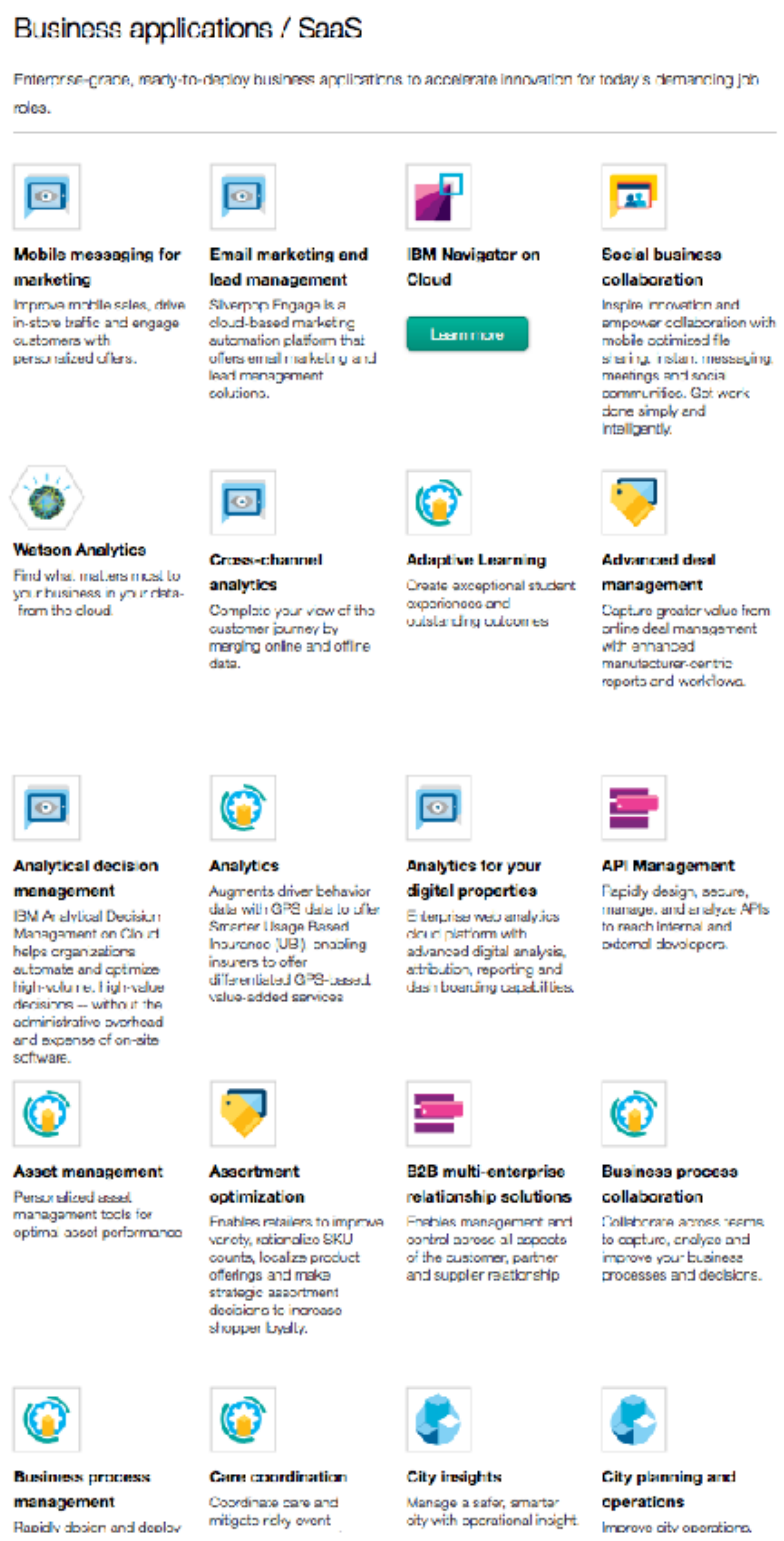
IBM and Cloud

openstack™
CLOUD SOFTWARE



SOFTLAYER®
an IBM Company

IBM Bluemix



Business applications / SaaS

Enterprise-grade, ready-to-deploy business applications to accelerate innovation for today's demanding job roles.

Mobile messaging for marketing
Improve mobile sales, drive in-store traffic and engage customers with personalized offers.

Email marketing and lead management
Silverpop Engage is a cloud-based marketing automation platform that offers email marketing and lead management solutions.

IBM Navigator on Cloud
Learn more

Social business collaboration
Inspire innovation and empower collaboration with mobile optimized file sharing, instant messaging, meetings and social communities. Get work done simply and intelligently.

Watson Analytics
Find what matters most to your business in your data - from the cloud.

Cross-channel analytics
Complete your view of the customer journey by merging online and offline data.

Adaptive Learning
Create exceptional student experiences and outstanding outcomes

Advanced deal management
Capture greater value from online deal management with enhanced manufacturer-centric reports and workflows.

Analytical decision management
IBM Analytical Decision Management on Cloud helps organizations automate and optimize high-volume, high-value decisions — without the administrative overhead and expense of on-site software.

Analytics
Augments driver behavior data with GPS data to offer Smarter Usage Based Insurance (UBI), enabling insurers to offer differentiated GPS-based value-added services

Analytics for your digital properties
Enterprise web analytics cloud platform with advanced digital analysis, attribution, reporting and dashboarding capabilities.

API Management
Rapidly design, secure, manage, and analyze APIs to reach internal and external developers.

Asset management
Personalized asset management tools for optimal asset performance

Assortment optimization
Enables retailers to improve variety, rationalize SKU counts, localize product offerings and make strategic assortment decisions to increase shopper loyalty.

B2B multi-enterprise relationship solutions
Enables management and control across all aspects of the customer, partner and supplier relationship

Business process collaboration
Collaborate across teams to capture, analyze and improve your business processes and decisions.

Business process management
Rapidly design and deploy

Care coordination
Coordinate care and mitigate risky event

City insights
Manage a safer, smarter city with operational insight.

City planning and operations
Improve city operations.

100+ SaaS Application

IBM Bluemix

SOLUTIONS CATALOG PRICING DOCS COMMUNITY SIGN UP LOG IN

Services // *The building blocks of any great app*

Watson

Build cognitive apps that help enhance, scale, and accelerate human expertise

Concept Expansion
IBM BETA

Language Identification
IBM BETA

Machine Translation
IBM BETA

Message Resonance
IBM BETA

Question and Answer
IBM BETA

Relationship Extraction
IBM BETA

User Modeling
IBM BETA

Mobile

Quickly get started with your next app

Mobile Application Sec...
IBM

Mobile Data
IBM

MobileQualityAssurance
IBM

Push
IBM

Twilio
Third Party

Web and Application

Deliver new web and apps

Business Rules
IBM

Data Cache
IBM

Gamification
IBM BETA

MQ Light
IBM

RapidApps
IBM BETA

Session Cache
IBM

1) Existing Application Portfolio

Designed largely for On-Premise Enterprise Deployment

Designed for dedicated operations staff

The Challenges

Richly Configurable (resulting in mass customization)

Optimized for Scalability, Performance and Flexibility - Not Operational Efficiency and Simplicity

The Challenges

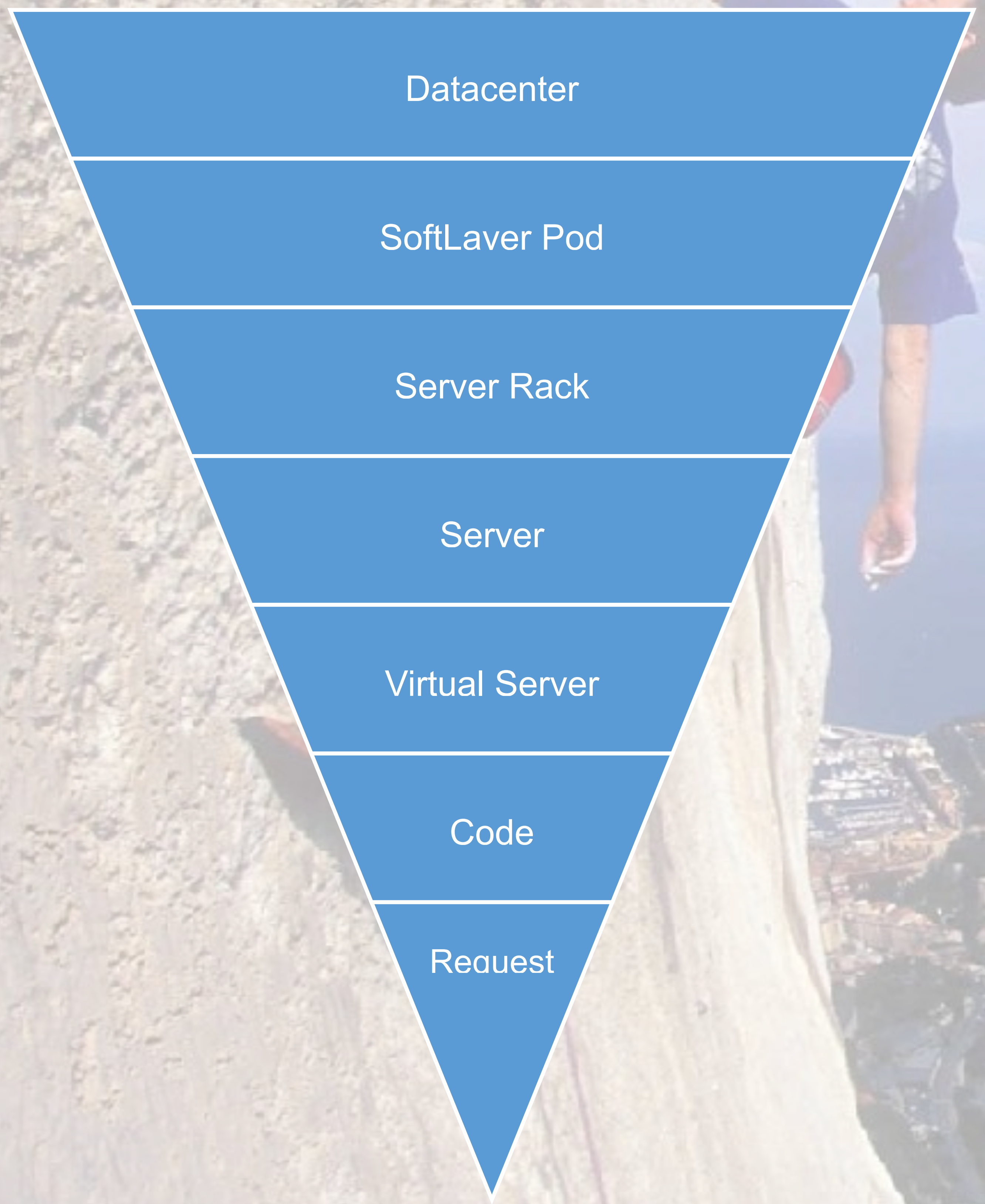
2) Transitioning from Software Developers to Cloud Service Operators

Most of the development team spent their career delivering packaged software

Focused on relatively small number of “releases” each year…. agile development with infrequent delivery

No “operations” culture… No one who has had to respond to PagerDuty

The Challenges



Datacenter

SoftLayer Pod

Server Rack

Server

Virtual Server

Code

Request

3) Understanding and Handling Failures

Failures will happen at any time and their root causes can be complex

Need to reduce the likelihood of correlated failures

Need to handle all kinds of failures

- An application could respond slowly, return a poor result or have a poor user experience

- These “failures” are harder to monitor and remediate

How did we get there?

Achieve **Operational Excellence** by:

Structuring services to minimize friction between developers and deployment

Teach the Zen of Cloud

Actively coding applications to embrace scale and minimize impact of failure

A/B testing at all levels to understand the impact of any change for the customer

Operational Excellence Checklist

Continuous Delivery and Integration: Changes delivered frequently with zero downtime

Best High Availability possible: No single point of failures, all components at least triple clustered

Automatic Recovery: Partial failure should be recovered by system

Elastic Scalability: Solution can scale out easily

Operational Visibility: Operators have live view of contextualized state of system

IBM

Technical Approach

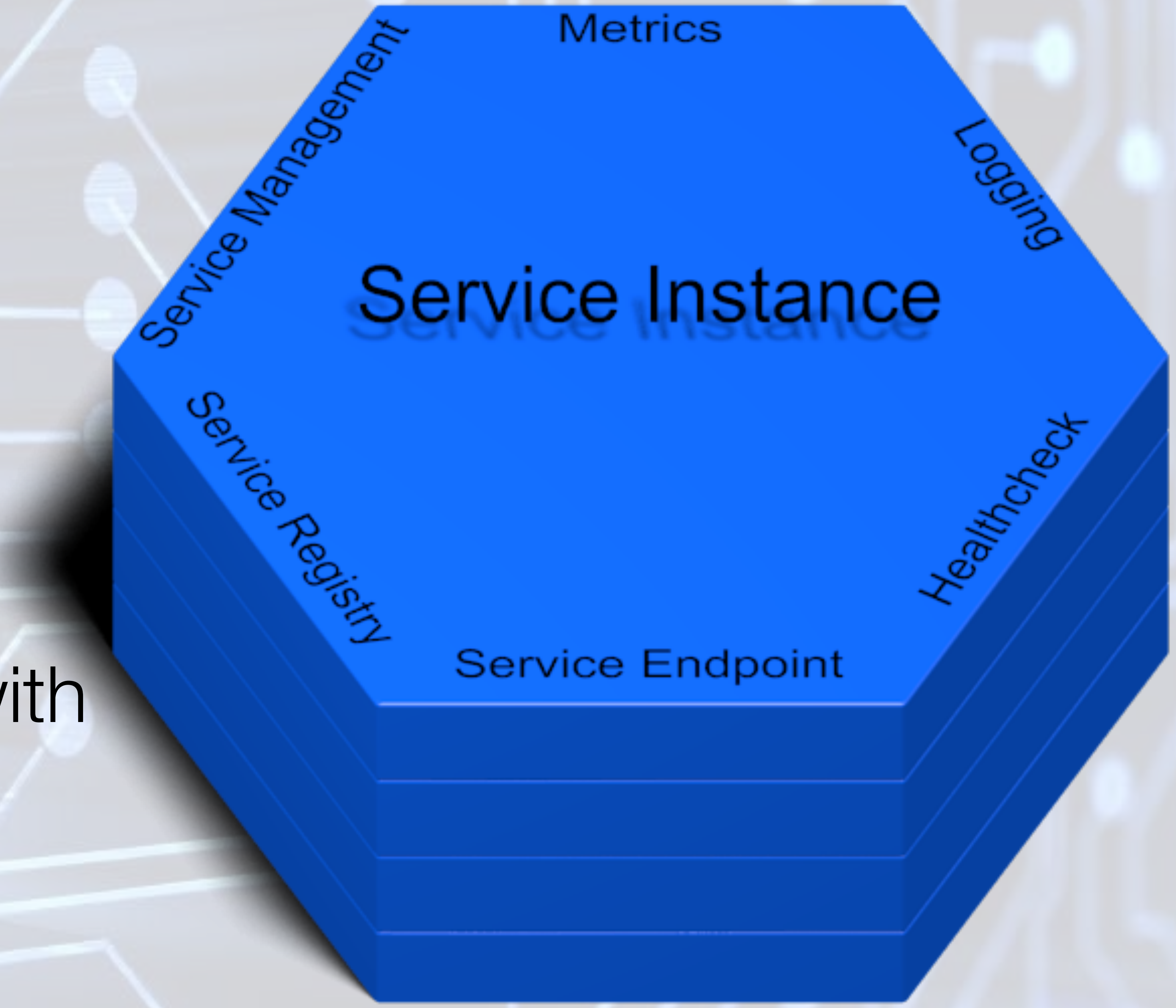
Standard Character of a Service Instance

Service Instances are ephemeral and immutable

Code and configuration are persisted at the time image is created

Future changes are made by deploying new instances

Service instances need to be either stateless or with the ability to scale up and down seamlessly



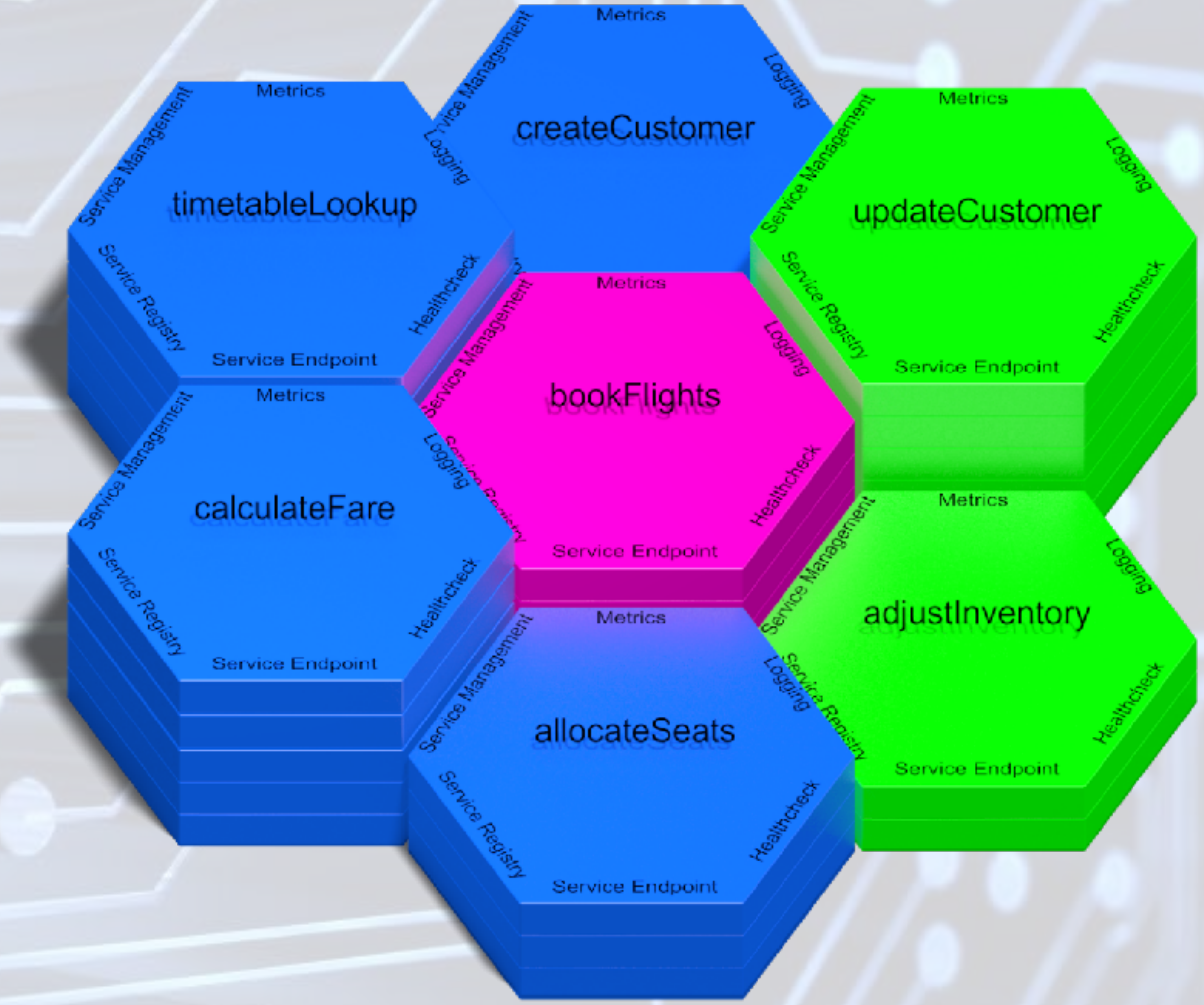
Microservices Architecture

Consists of a set of narrowly focused, independently deployable services

Used by Netflix, eBay & Amazon.

Each deployed service is managed and scaled separately

Embrace immutability of code, deployment states and environment



Microservices Architecture

Benefits

- The size which allows developers to be most productive
- Easier to comprehend and test each service
- Correctly handle failure of any dependent service
- Reduces impact of correlated failures

Issues

- Monitoring and managing the complexity
- Teams need to agree the interfaces for each microservice
- Need to support multiple versions of each microservice api

Components of a Cloud Services Fabric

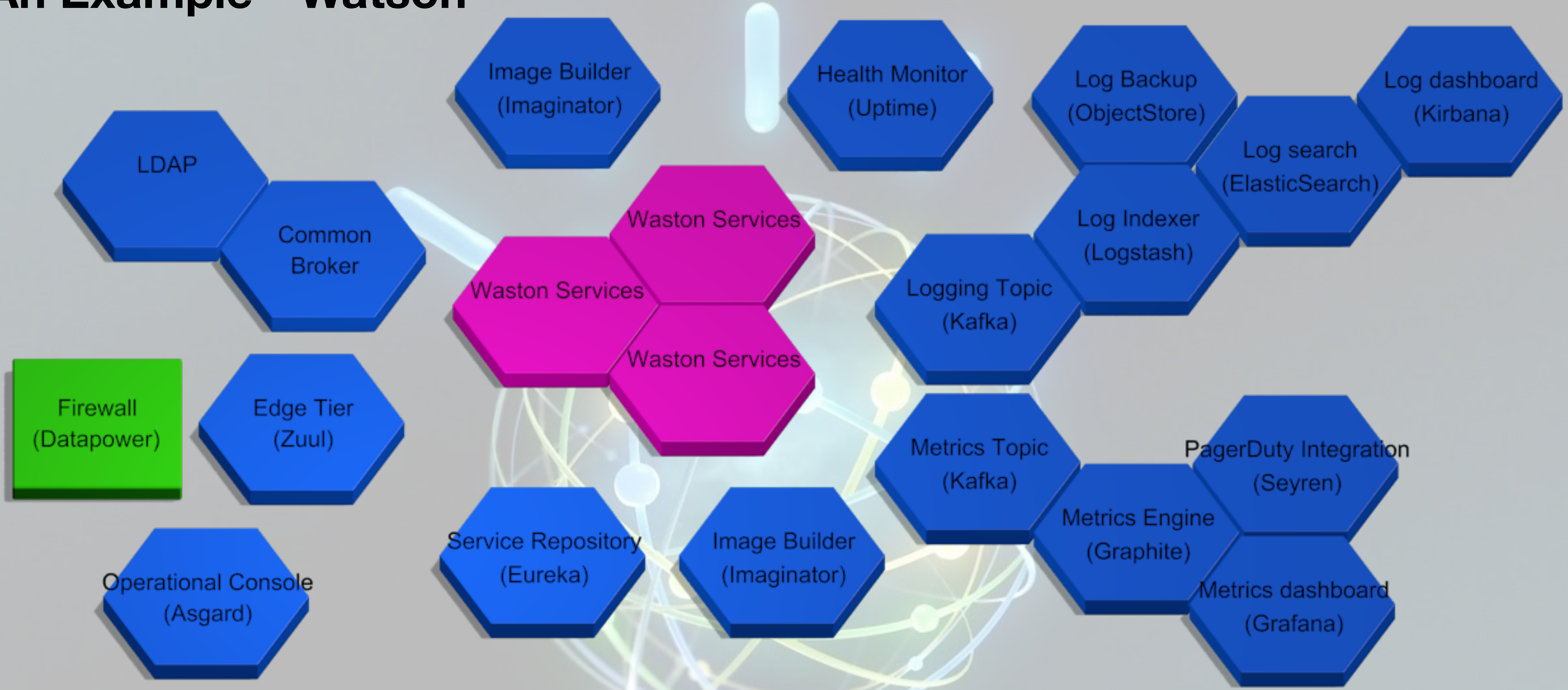
Sb CF Service Broker	A API Management								Fb Feedback
Cb Circuit Breaker	Pt Patterns	Ca Config Automation						Hm Health Monitoring	Ev Event Notification
Sp Service Proxy	Im Image Builder	Cf Configuration	Dc DevOps Control			Ac Access Control	Au Audit & Compliance	Mk Testing Monkeys	Lg Logging
Sd Service Discovery	Ob Object Store	Ar Archive & Backup	Eb Elastic Block Store	Sg Storage Gateway	Cn CDN	Sc Secure Connectivity	Dn DNS	Pd PagerDuty	Da Service Dashboard
Zk Zookeeper	Vm Virtual Machines	As^{v} AutoScaler (VM)	C Docker Containers	As^{c} AutoScaler (Container)	Se Security Groups	Fw Firewall	Lb IP Load Balancer	Up Uptime	Mc Monitoring Collection

Development Services

Infrastructure Services

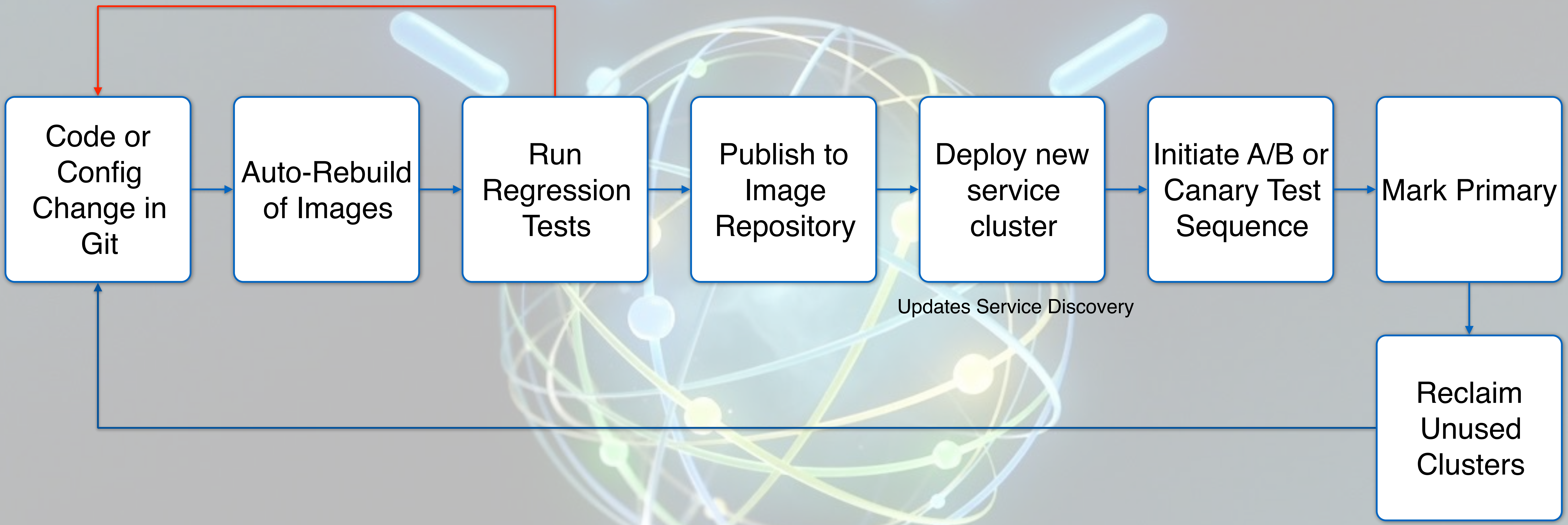
Operations Services

An Example - Watson



SOFTLAYER®
an IBM Company

The Update Sequence



Code or Config Change in Git

Auto-Rebuild of Images

Run Regression Tests

Publish to Image Repository

Deploy new service cluster

Updates Service Discovery

Initiate A/B or Canary Test Sequence

Mark Primary

Reclaim Unused Clusters

Active testing in production

A suite of tools called the **Simian Army**

Focused on managing & testing a cloud

Chaos Monkey

Will randomly terminate a server, during office hours to recreate outages

Chaos Gorilla

Will do the same thing to an availability zone or region

Janitor Monkey

Will sweep and mark unused resources

Owners notified, then removed

Conformity Monkey

Check instances are conforming to rules around security, software levels, coding standards etc.

Latency Monkey

Will introduce latency into calls, to test handling of error conditions

When disaster strikes….

What happens when your cloud reboots?

Learned that zookeeper is complicated, and developers built unnecessary dependencies on it

- Significant outage (before go live) was caused by this
- Fixed by changing dependency to more appropriate technologies

Team Dynamics

Who Owns the Production Environment??

- Developers – some wanted, some didn’t
- Operation
- Support

Our structure was a Development team with a small dedicated Ops team

Driving the Right Behaviors

- Developers carry the “pager”
- Developers have access to production environments
- Penalty Free Failure
- Transparency

Questions?