

# Applications Architecture



Joseph Anthony

@ansolabs | [www.ansolabs.net](http://www.ansolabs.net)

# Applications Architecture



Applications Architecture emphasizes on:

Taking strategic view of enterprise applications

Creating evolving standards

Creating policies and guidelines

Applications architectures inform and guide:

- Procurement
- Development
- Integration
- Deployment
- Delivery

# Module Focus

Future Directions & Trends

Foundational Concepts & Context

Why Applications Architecture?

# Module Focus

Why Applications  
Architecture?

Foundational  
Concepts & Context

Future Directions &  
Trends



Over the last few decades, businesses have invested heavily in business applications across all industry sectors



Every business is a software business



Enterprise applications architecture is a key determinant of how quickly and effectively business as a whole can adapt

# Case Study – Large Financial Services Organization

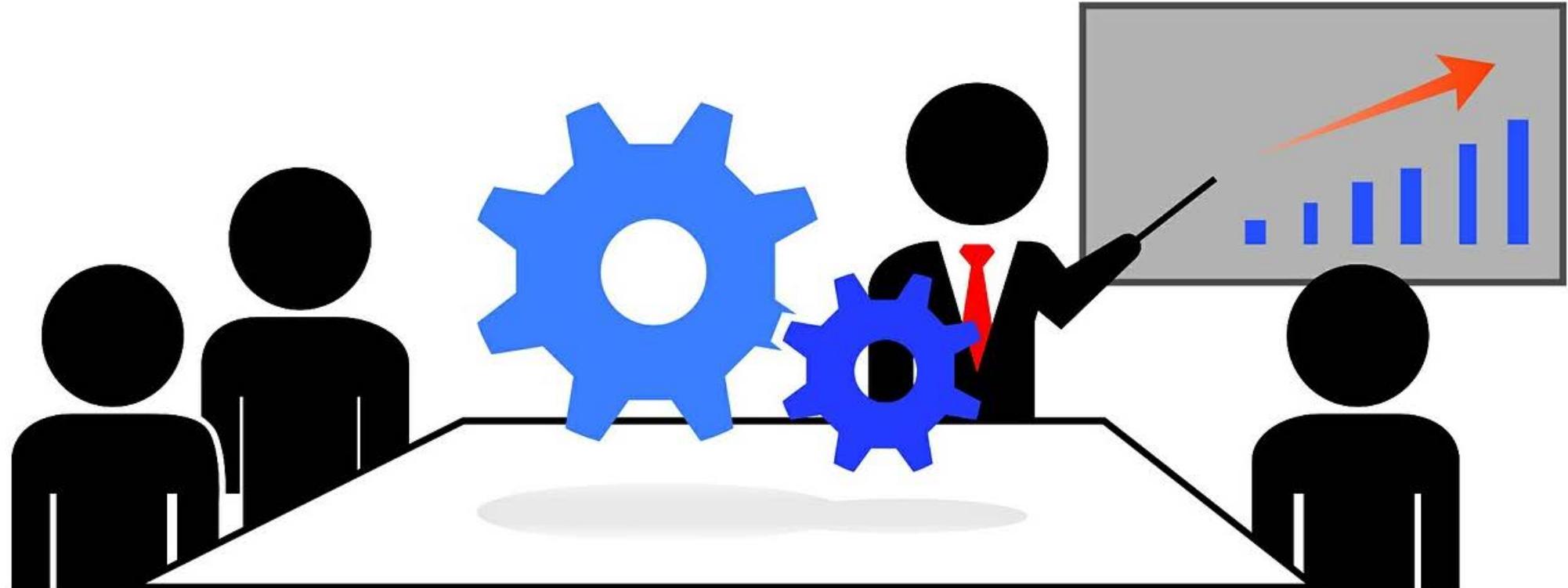


Regulatory changes to protect end-consumers obtaining financial advice  
The regulation mandated compliance by a hard end date  
Resulted in large number of application to be modified  
An effective and flexible applications architecture could have accomplished this at a fraction of the cost

# Case Study – Online Retail Organization



Expected to complete a customer centric redesign in 12 to 18 months  
The challenge of integrating more than 120 of its backend systems proved too much  
Development eventually completed in 5 years  
Overtaken by nimbler and faster moving competition



Overall architecture of the applications portfolio  
Consciously and continuously crafted and evolved principles, practices,  
policies and tools to realize an effective applications architecture can give it a  
huge competitive edge

# Scenarios



Mergers & Acquisitions



Business Operating Model



Compliance Management



Vendor Management



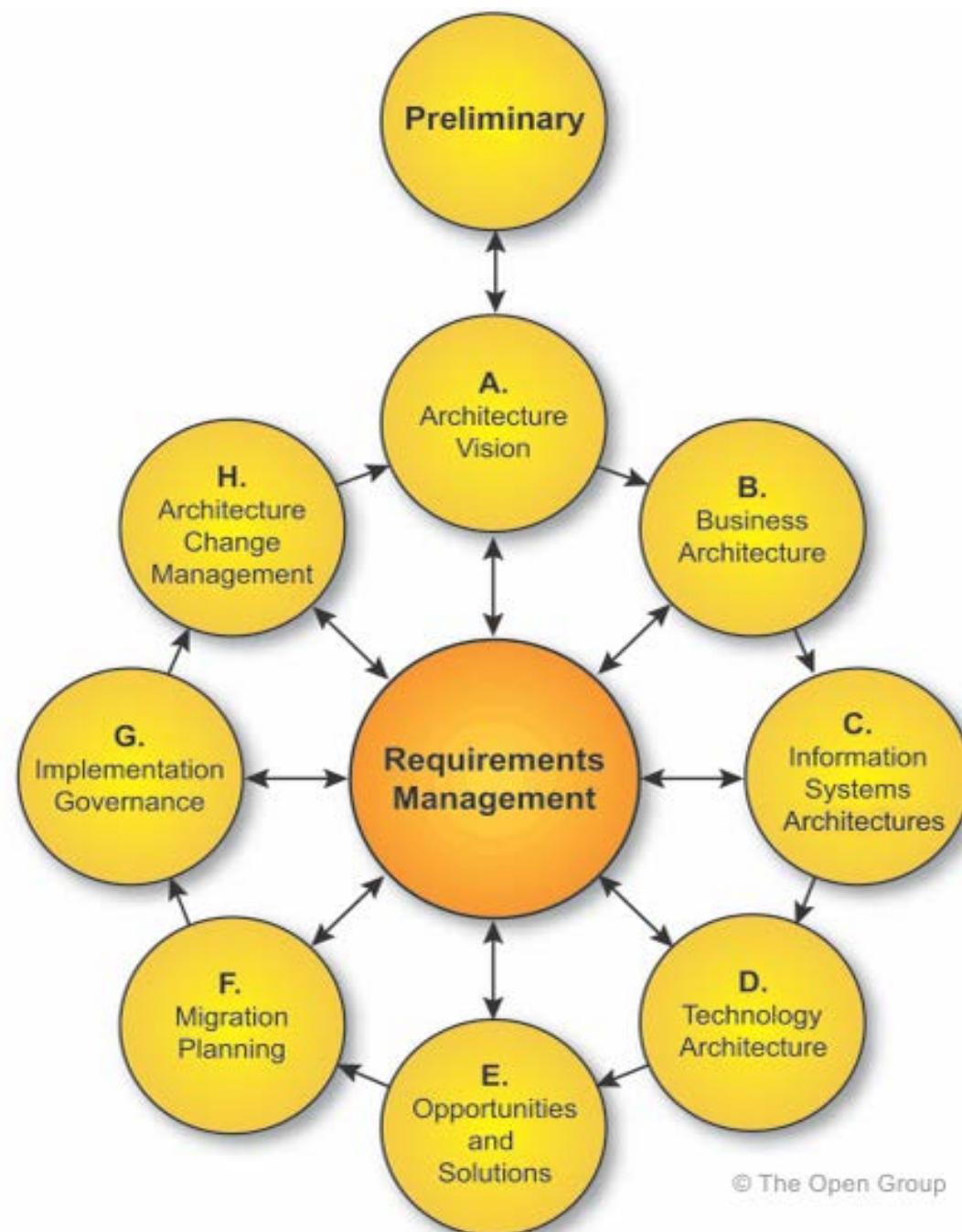
Risk Management

# Module Focus

Why Applications  
Architecture?

Foundational  
Concepts & Context

Future Directions &  
Trends



© The Open Group

# Applications Instantiates Capabilities & Processes



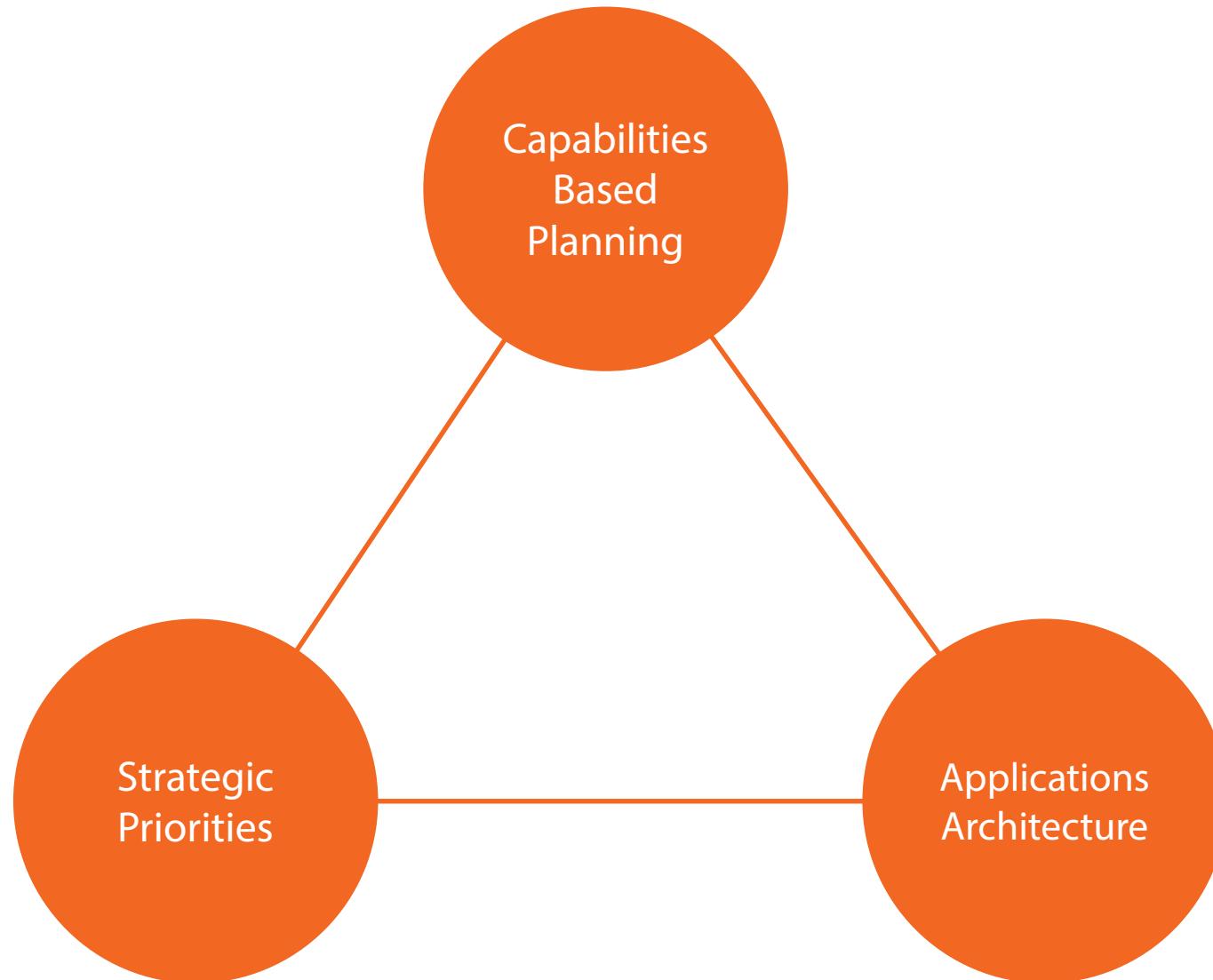
Key inputs to applications portfolio architecture include:

Strategic vision

Strategic business objectives

List of business capabilities

# Influences on Applications Architecture



# Applications Portfolio Analysis



Capability centric view of applications portfolio

Other architecture perspectives include:

Long term strategic objectives

Operational and tactical priorities

Business operating model

Organizational and political structures

A close-up photograph of a person's head and shoulders. They are wearing a dark flight helmet with a clear visor. The visor reflects a landscape of green fields and trees under a blue sky. The person is wearing a blue shirt. The background is blurred.

Enterprise applications architect creates target state applications portfolio roadmap, taking into consideration:

Total cost of change  
Return on investments  
Risks and path of least resistance

# Target State Might Include ...



- Identified gaps in application capabilities
- Decision to retire aging and low-value applications
- Modernizing legacy yet high-value applications
- Eliminating redundancy
- Standardizing on common technology platform
- Consolidating applications

# Governance and Grooming



Applications architecture and portfolio rationalization cannot be a one time initiative in today's business environment

There is a need for continuous grooming and on-going governance

Applications portfolio architecture management process need to be established

Should include executive level sponsorship to enable effective strategic decision making

The governance board should include cross-company business and IT leadership

# Module Focus

Why Applications  
Architecture?

Foundational  
Concepts & Context

Future Directions &  
Trends



How long would it take your  
organization to deploy a change that  
involves just one single line of code?

— Mary Popen dieck



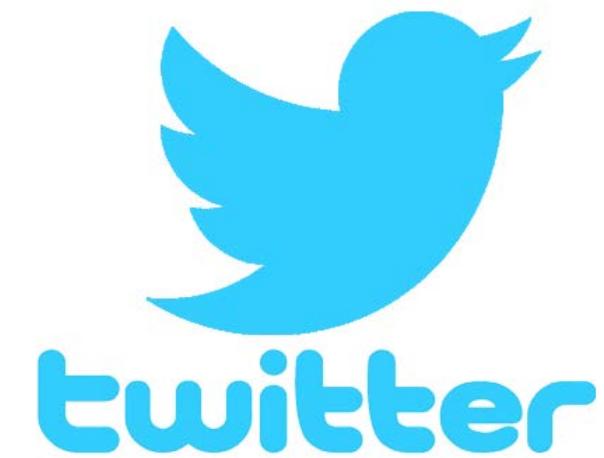
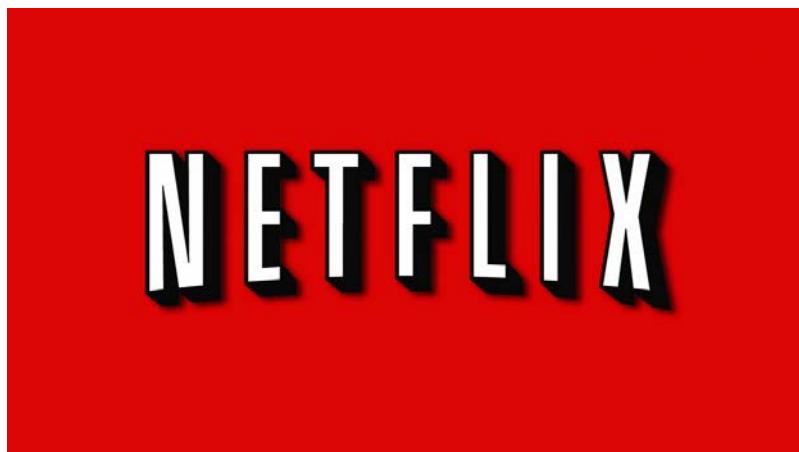
A photograph of a team of rowers in a boat, likely a dragon boat, in motion on water. The rowers are wearing life jackets and are pulling on green oars. The image is slightly blurred to convey movement. Overlaid on the bottom half of the image is a semi-transparent gray rectangular box containing the text.

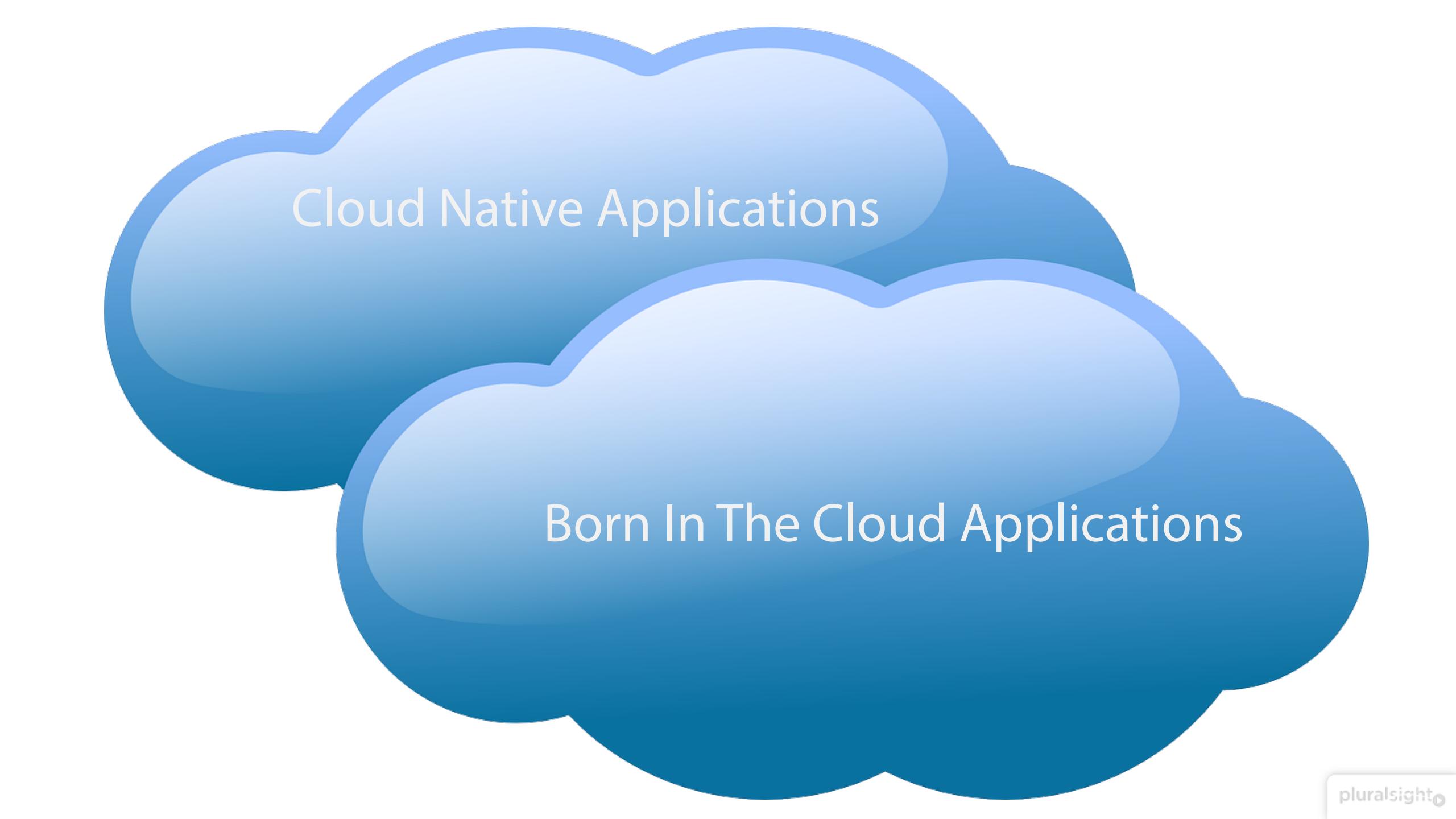
Businesses that are able to innovate, experiment and deliver software-based solutions quickly are outcompeting those that don't



Quarterly enterprise release cycles thwarts  
creative experimentation

# New Model for Creating & Releasing Software ...





Cloud Native Applications

Born In The Cloud Applications

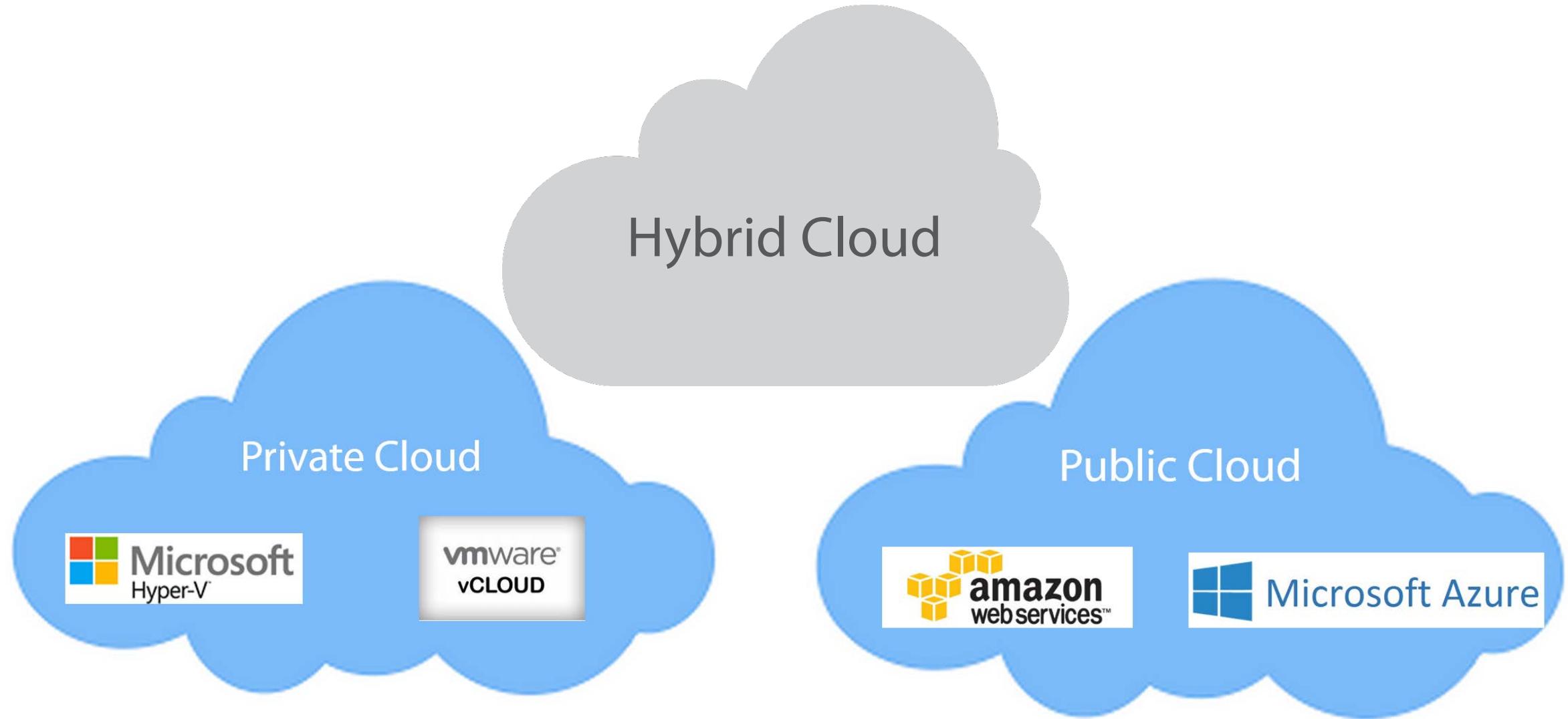
# Modern Business Applications Characteristics



- Fashioned for the cloud
- Support modern user interaction patterns
- Designed to scale elastically
- Optimized to support continuous delivery and/or deployment
- Works within architectural and operational safe guards
- They are antifragile

# Designed for the Cloud

# Modern Applications Architecture Are Cloud Native



# Automated Deployment

urban{code}

An IBM company

Release  
Management



Visual  
Studio

Microsoft

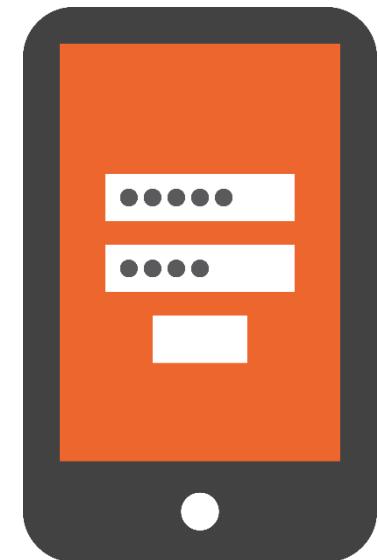
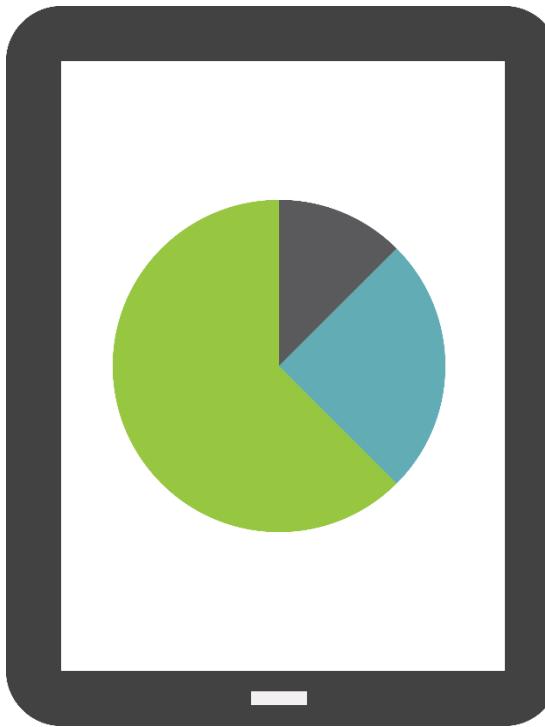


Jenkins

XL DEPLOY

# Designed to Support Modern End User Interaction Patterns

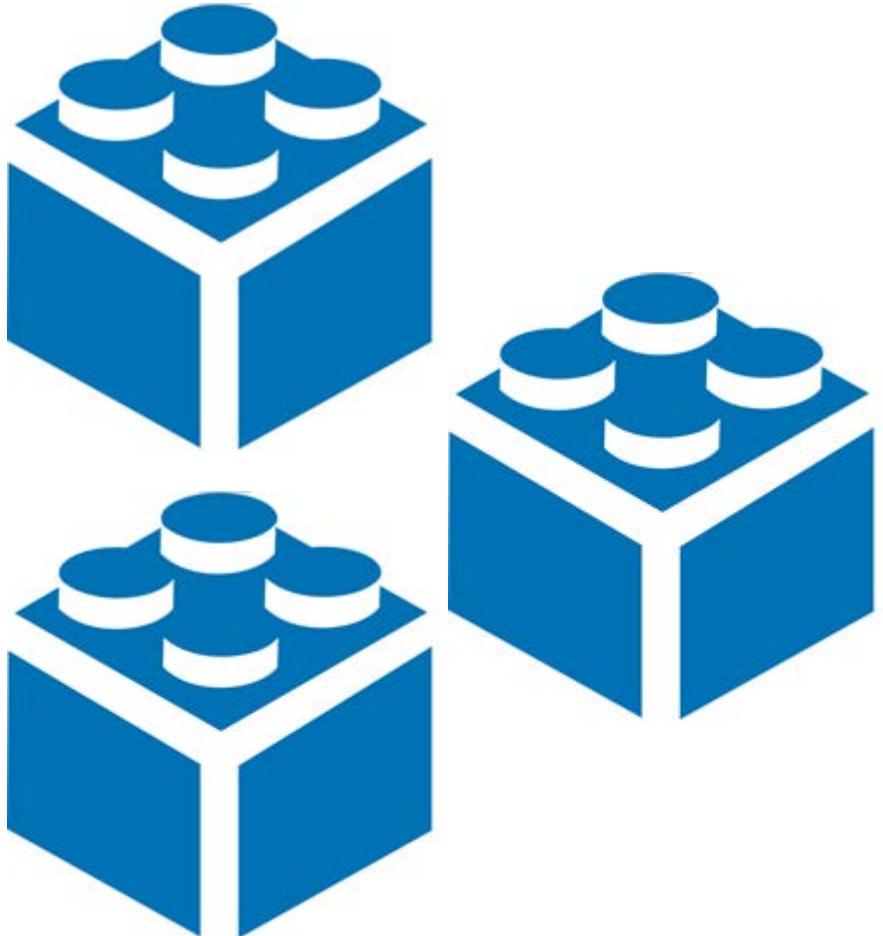
# Business Applications Expected to Support Multiple Device Types





Netflix supports 1000+ device types  
Developing business applications for multiple device types is a norm and presents an architectural challenge

# Emergence of API Centric Architecture



API centric architectures sprung up in response to modern application interaction patterns

They follow simple design tenets:

Restful design

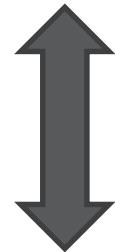
JSON-based data

Simple Versioning

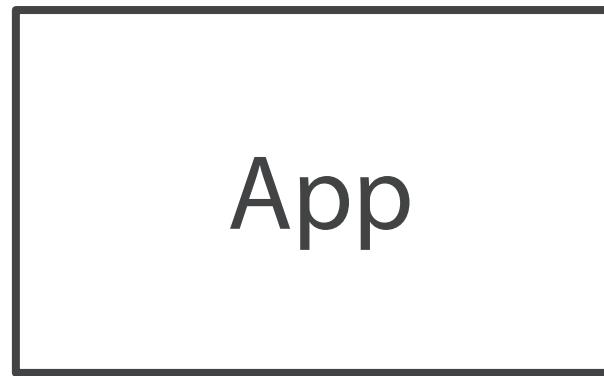
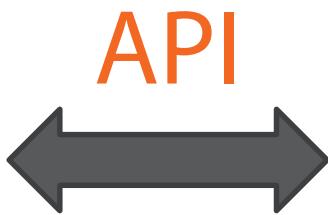
Key-based access control

The same basic APIs now support single page applications and native client user interfaces

Front End



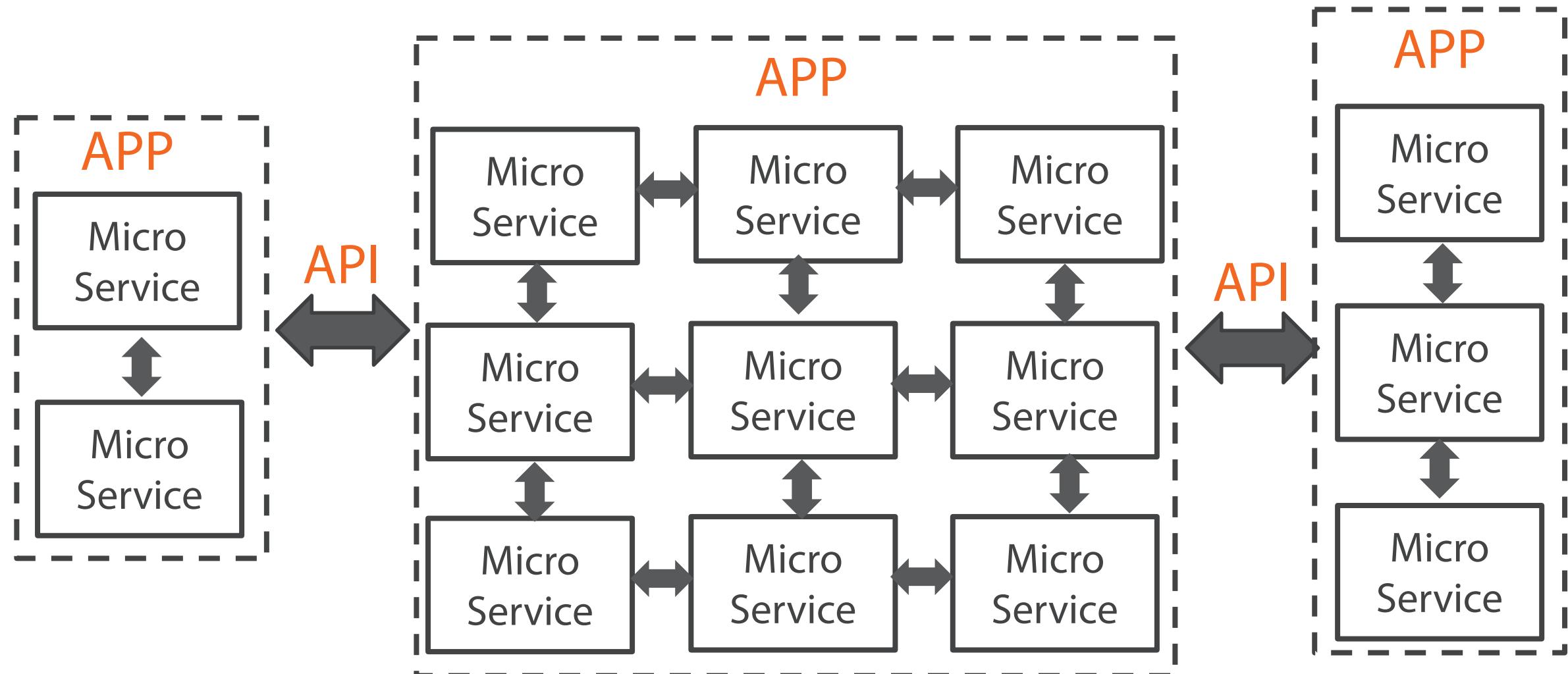
API



Backend

Application to application integrations leverage APIs

# Microservices Architecture



Microservices imply composing applications with fine-grained services  
Microservices interact through well-defined APIs

# Benefits of Microservices Architecture



Development teams organized around microservices benefit from:

Higher autonomy

Greater cohesion

Quicker feedback cycle and decision making

Other benefits include:

Fault isolation

Auto recovery

Lower dependencies

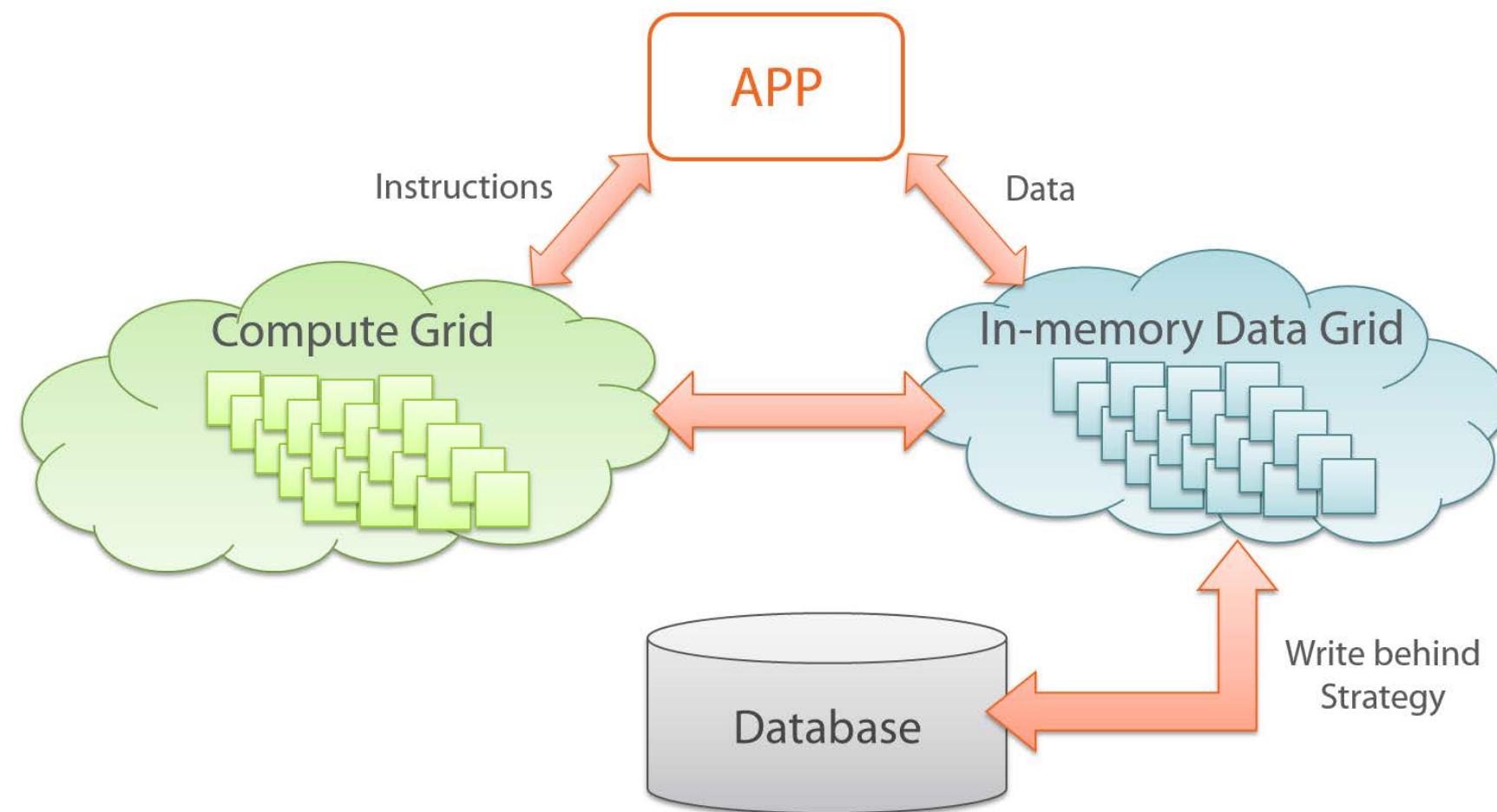
Ability to scale parts of the applications selectively

# Architected for Elastic Scaling

## Shared Nothing (SN) Architecture

Distributed computing architecture in which each node is independent and self-sufficient, and none of the nodes share memory or disk storage which could potentially cause contention

# In-Memory Data Grids



**ORACLE®**  
Coherence

**IBM**

 **XAP**  
In-Memory Computing Platform

# Optimized for Continuous Delivery/ Deployment

# Continuous Delivery vs. Continuous Deployment

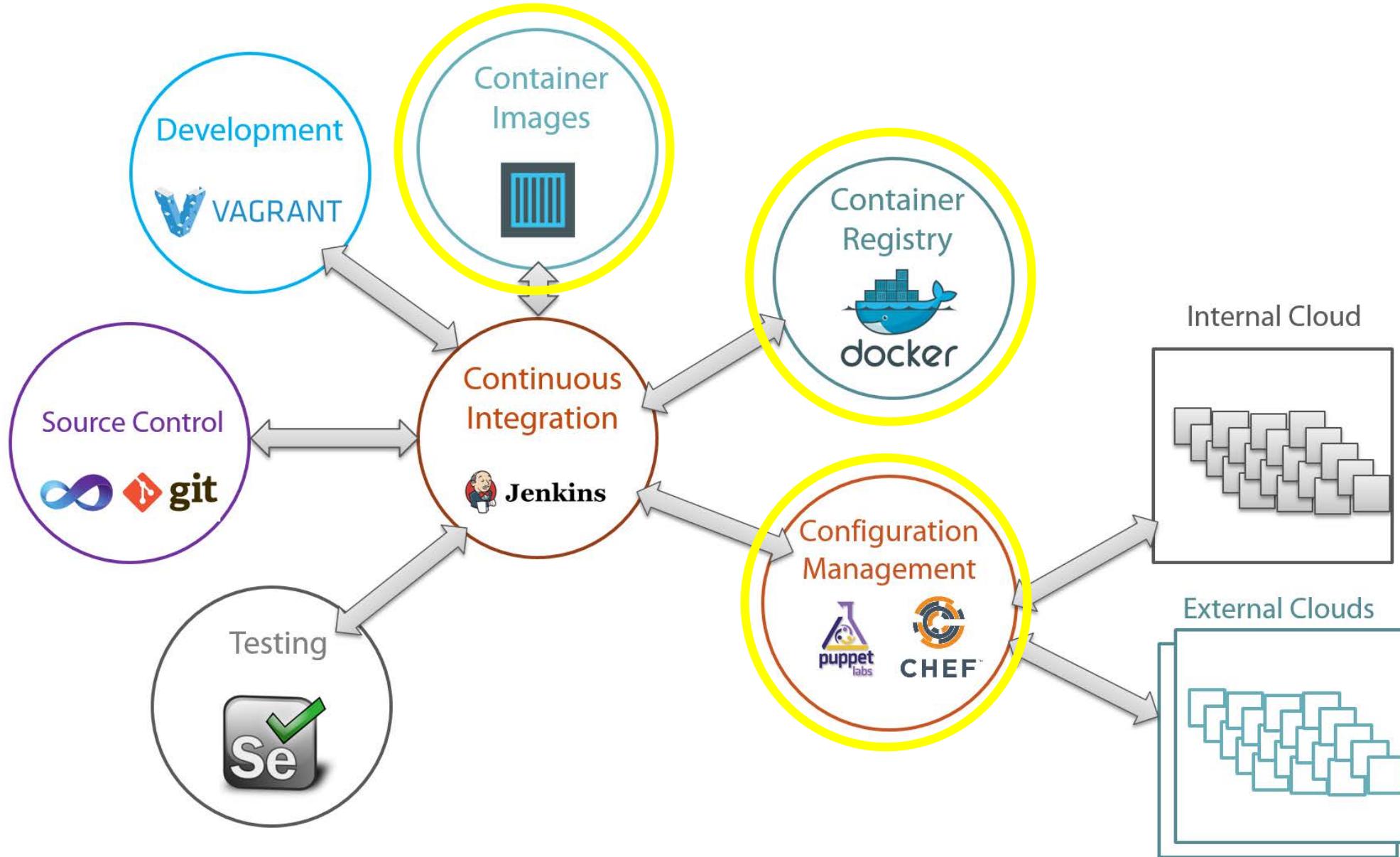
## Continuous Delivery

- Development to production deployment workflow is optimized to the extent that the checked in code is built, verified, tested and deployed each time to a production-like environment
- Ready to be pushed into production quite literally at the press of a button when business is ready for it

## Continuous Deployment

- Continuous deployment implies that the code is actually deployed to production environment every single time

# Automated Deployment Pipeline



# Architectural and Operational Safeguards

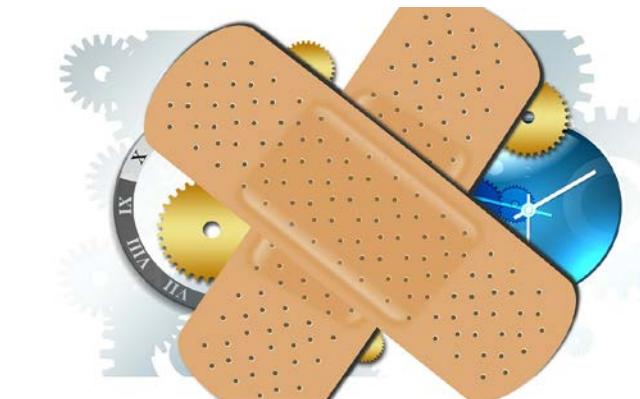
# Architectural/ Operational Safeguards



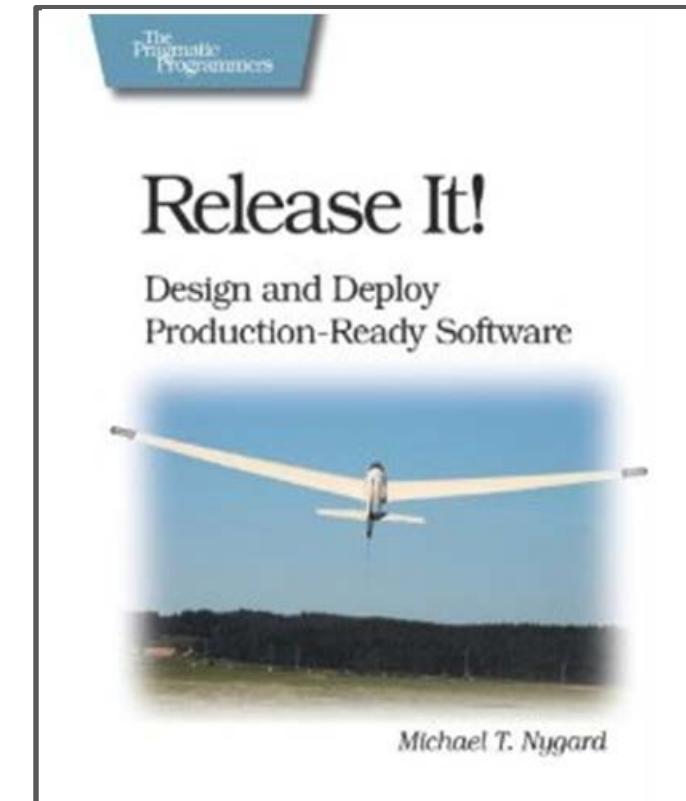
Continuous Monitoring &  
Health Checks



Fault Isolation

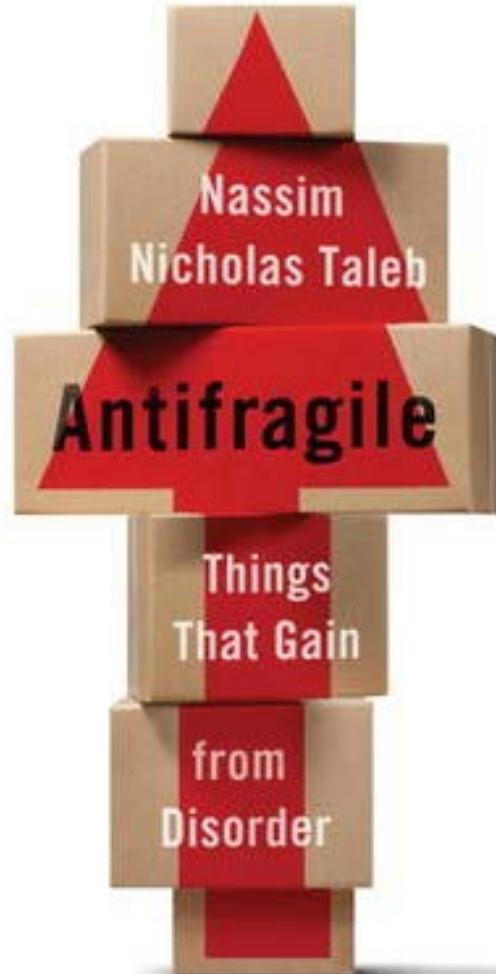


Auto-recovery



# Antifragility

NEW YORK TIMES BESTSELLING AUTHOR OF  
THE BLACK SWAN



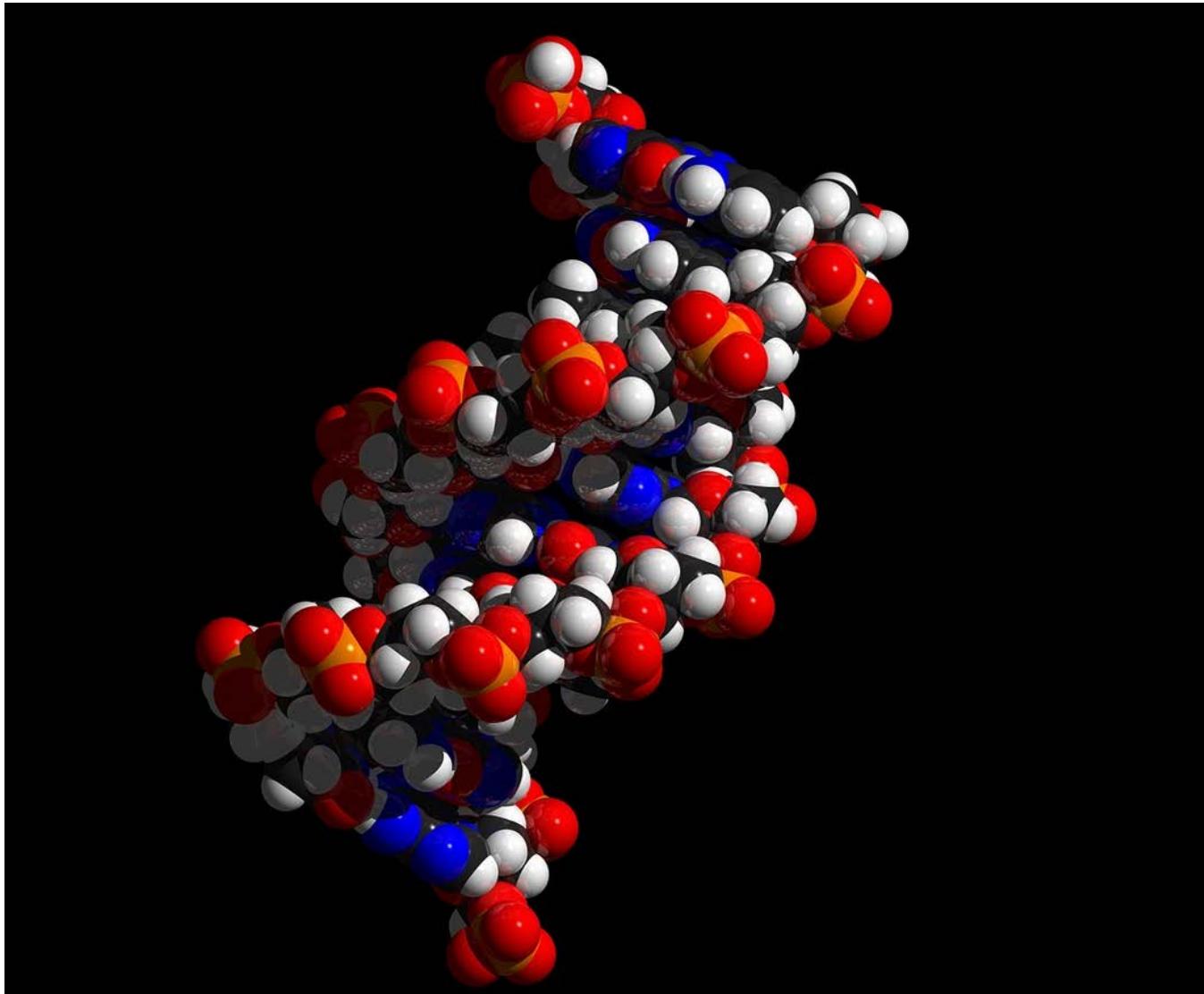
The characteristic that is the  
opposite of fragile



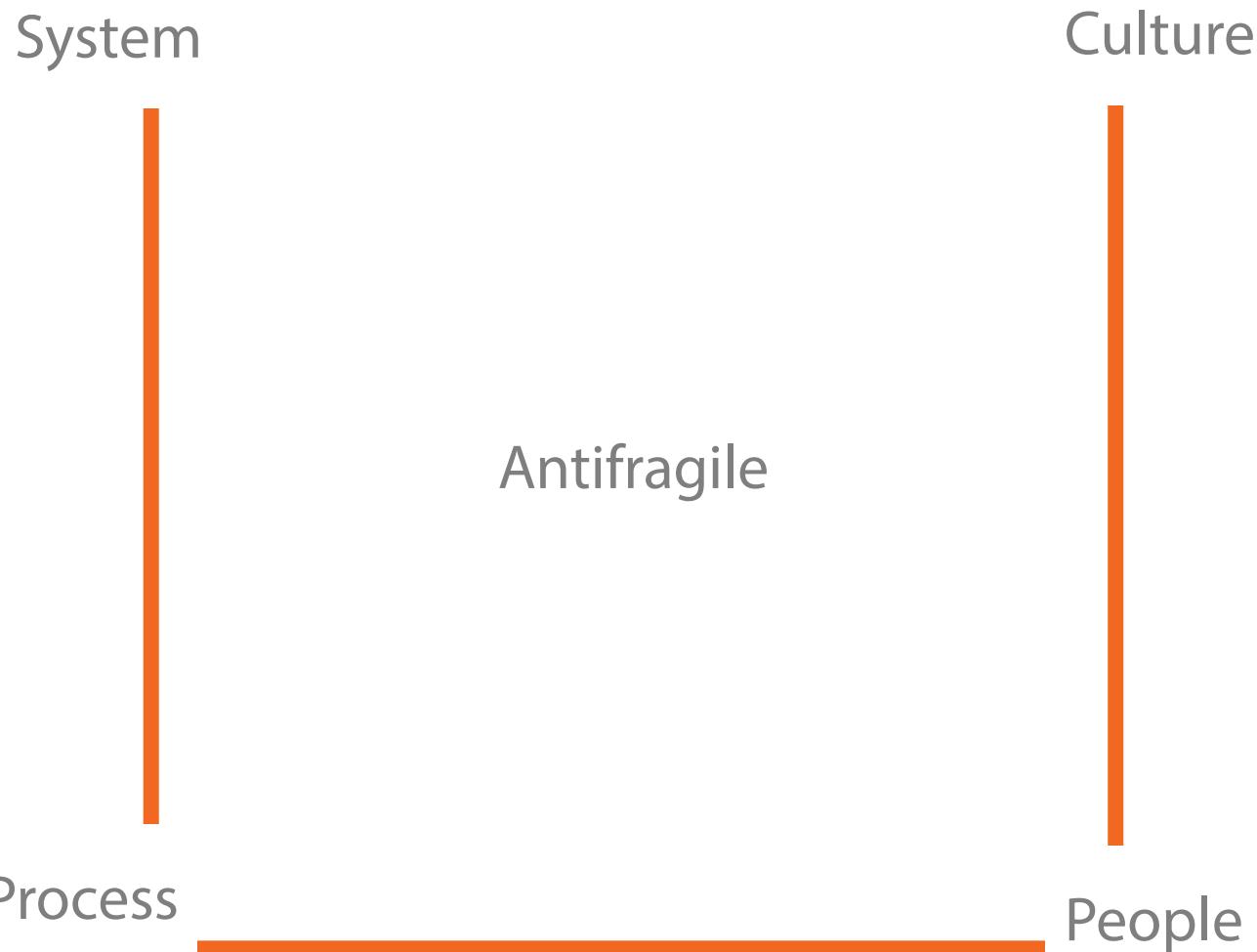
If you apply stress to a system  
that is fragile, it breaks



Robust does not imply enduring



Systems that get better with stress are referred to as antifragile



System

Culture

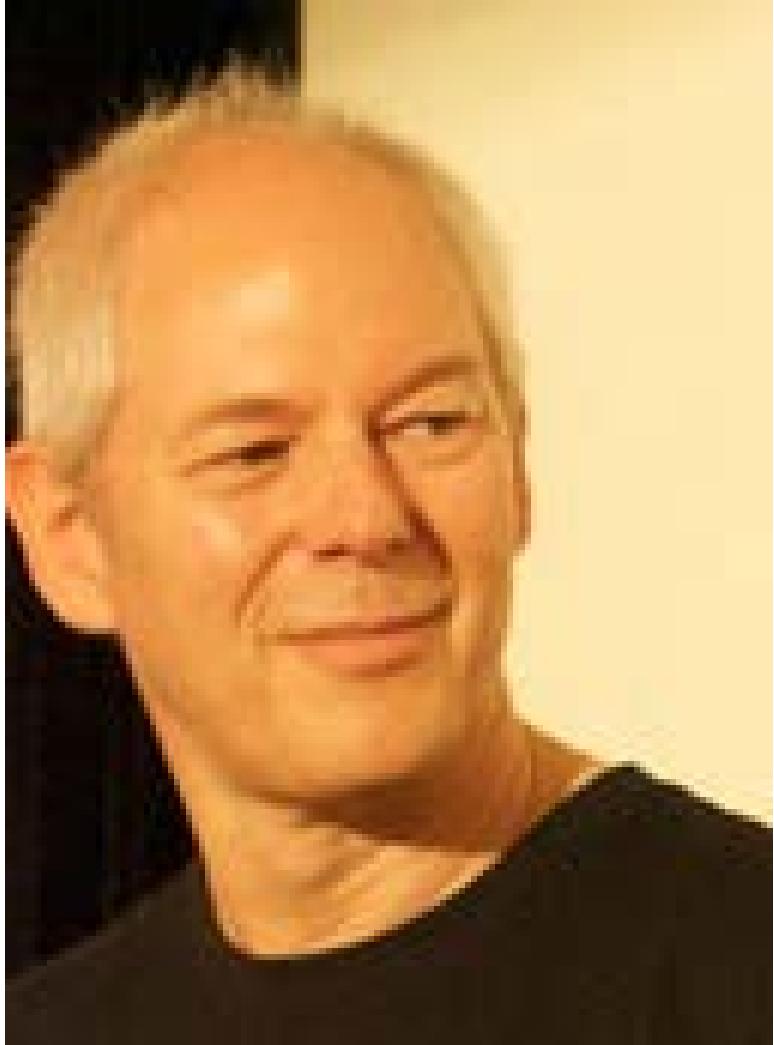
Process

People

Antifragile

# Netflix Simian Army Project





Adrian Cockcroft  
( Previously Cloud Architect – Netflix)

“When a user initiates an interaction with the Netflix application, that action typically invokes hundreds of connected services in the infrastructure. If any of them fails, that piece of functionality simply isn’t offered.”

# Lead Cast of Simians



Chaos Monkey

Chaos Gorilla

Chaos Kong

Janitor Monkey

Conformity Monkey

# Module Summary

Why Applications Architecture?

Context for Applications Architecture

Trends and Future Directions