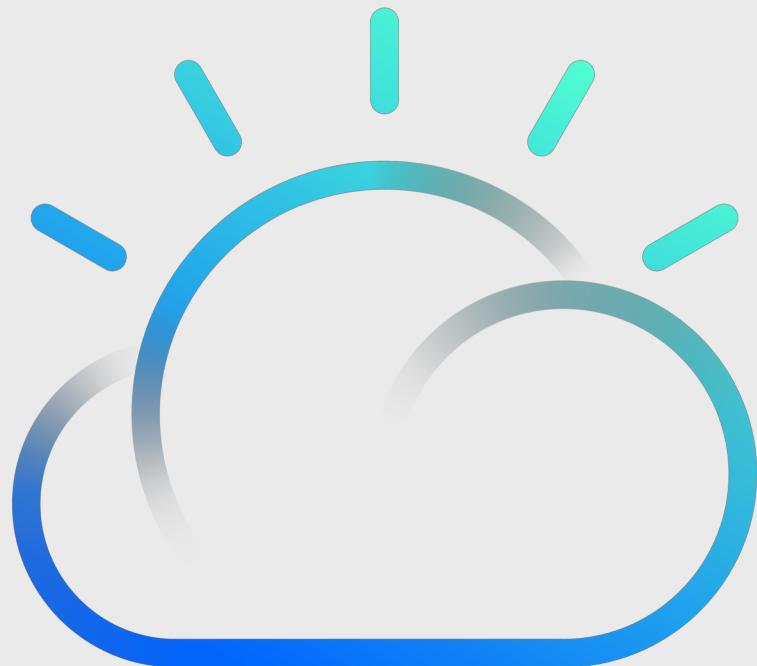


# **IBM Application Modernization**



**IBM Cloud**

**IBM**

# Application Modernization

Application modernization is **business modernization**

Application modernization is the **transition of existing applications to new approaches on the cloud**, helping you achieve the following business outcomes:

- Speed to market
- Rapid innovation
- Flexibility
- Cost savings

## Accelerate digital transformations

Driven by need to build new capabilities and deliver them quickly

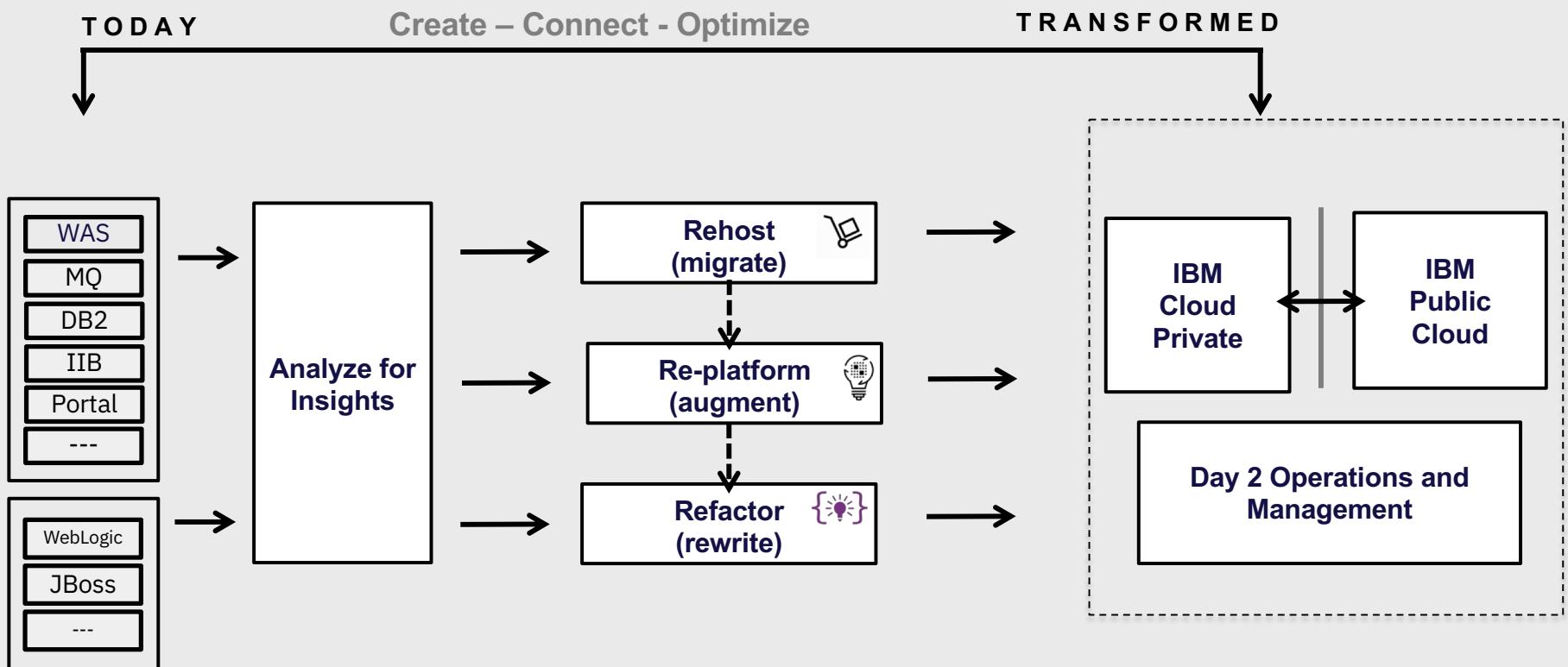
## Improve developer productivity

Developer self service via adoption of Cloud Native architecture and Containerization

## Increase operational efficiency and standardization

DevOps enablement drives a culture of automation and transformation of operations

# Application Modernization Framework



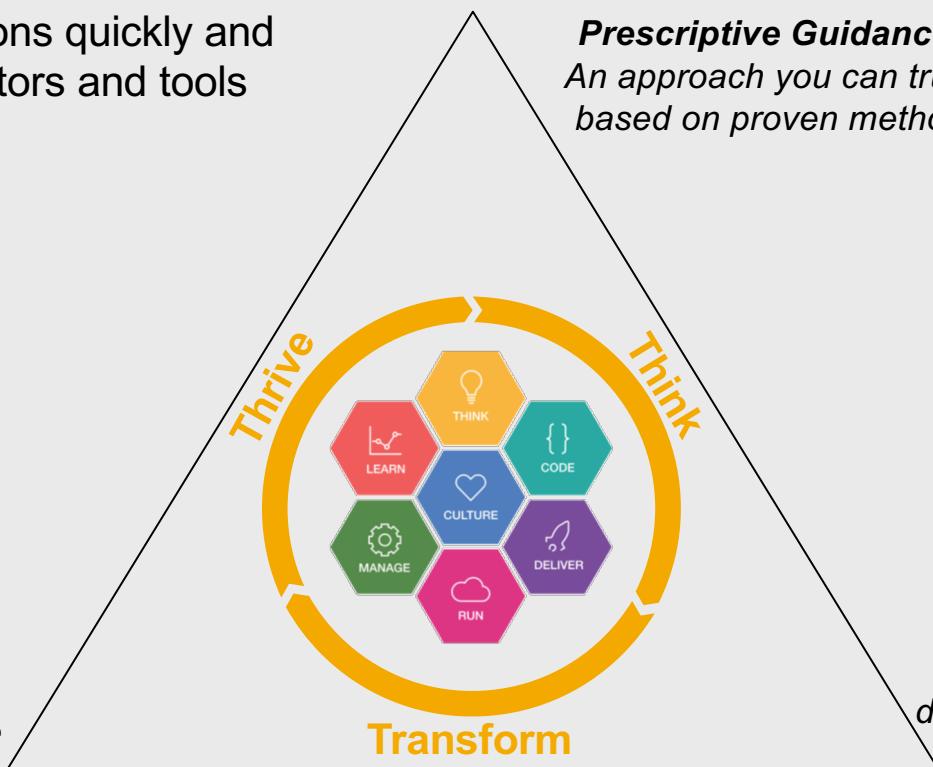
# IBM's AppMod Method

Modernize your applications quickly and safely using our accelerators and tools

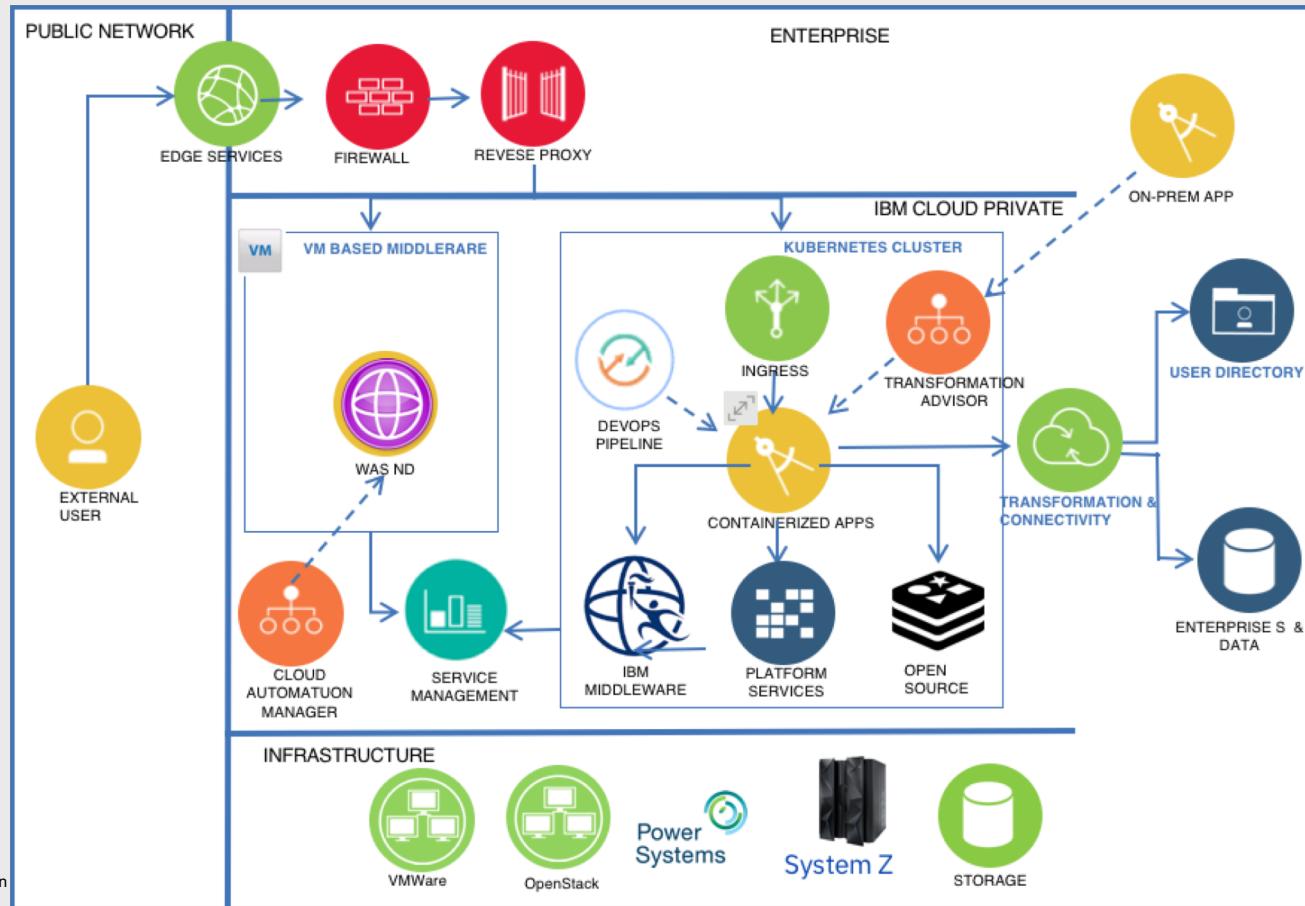
**Prescriptive Guidance:**  
*An approach you can trust based on proven method*

**Business value:**  
Accelerate value by leveraging existing estate

**Technology:**  
*Discover and automate deployment to speed time to market*



# Reference architecture for application modernization



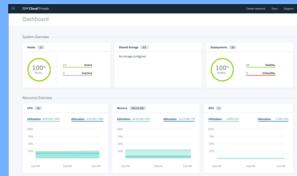
# Application modernization technology

## Garage Method

Architectures, best practices, and toolchains to jump-start modernization

### IBM Cloud Private

A transformative platform for building and running cloud-native applications and modernizing existing enterprise.



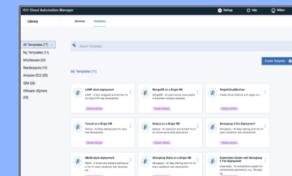
### Transformation Advisor

Assess and Deploy traditional apps into IBM Cloud Private



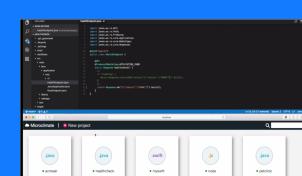
### Cloud Automation Manager

Multi-Cloud Provisioning  
Pre-Built Automation Content



### IBM Microclimate

End to end development environment that lets you rapidly create, edit, and deploy applications.



## Cloud Production

Deployment planning

Production environment

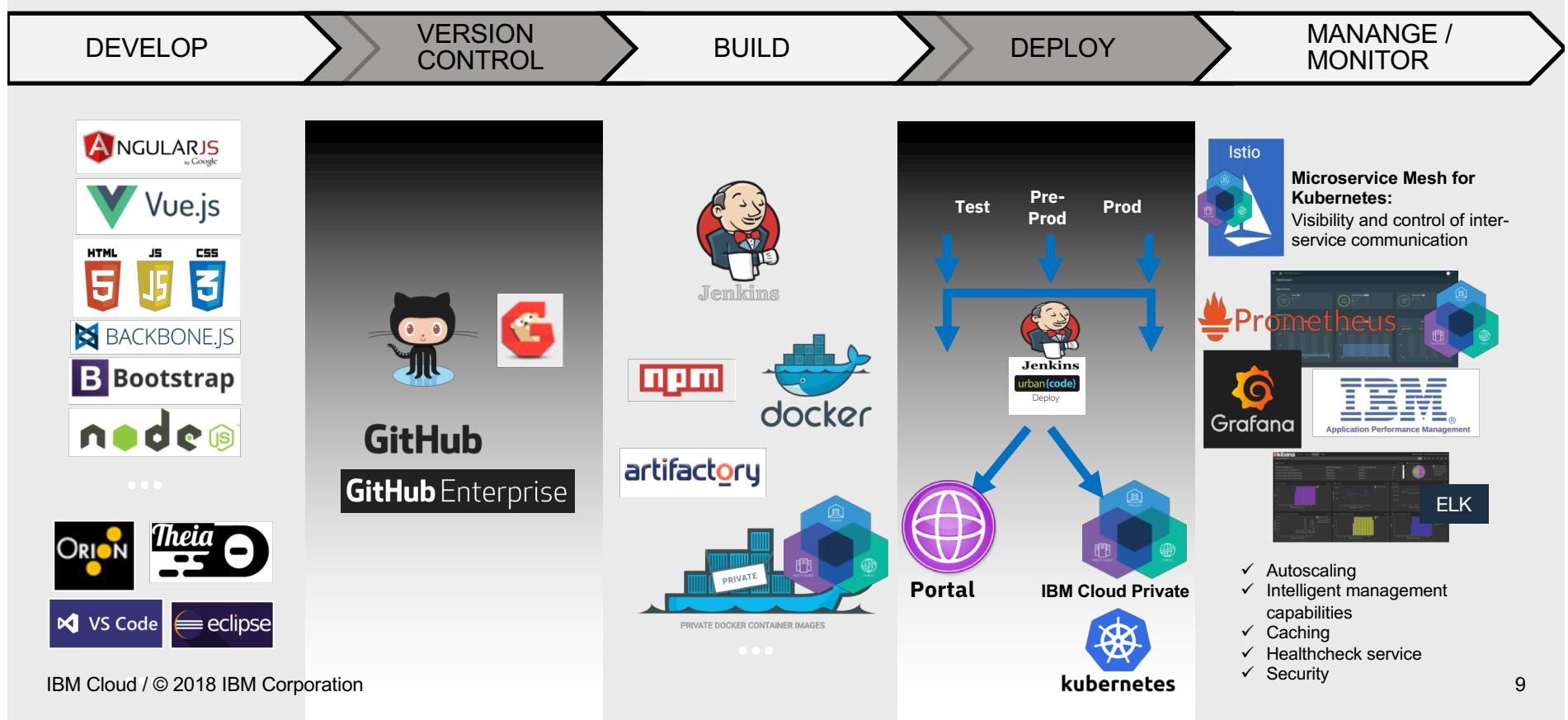
Production readiness

Service management

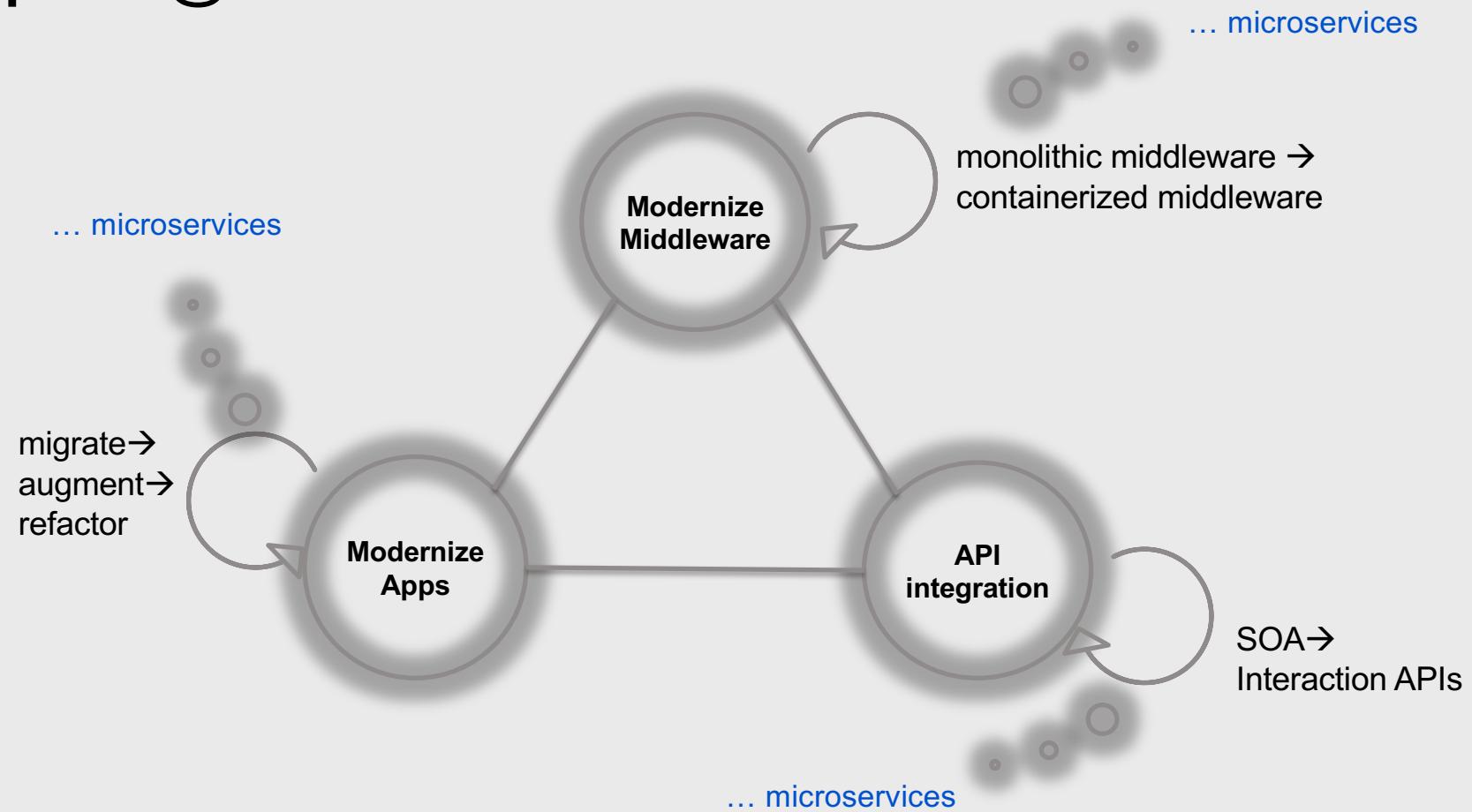
DevSecOps

Backup and recovery

# Development modernization and pre-integrated DevOps



# AppMod @3000ft

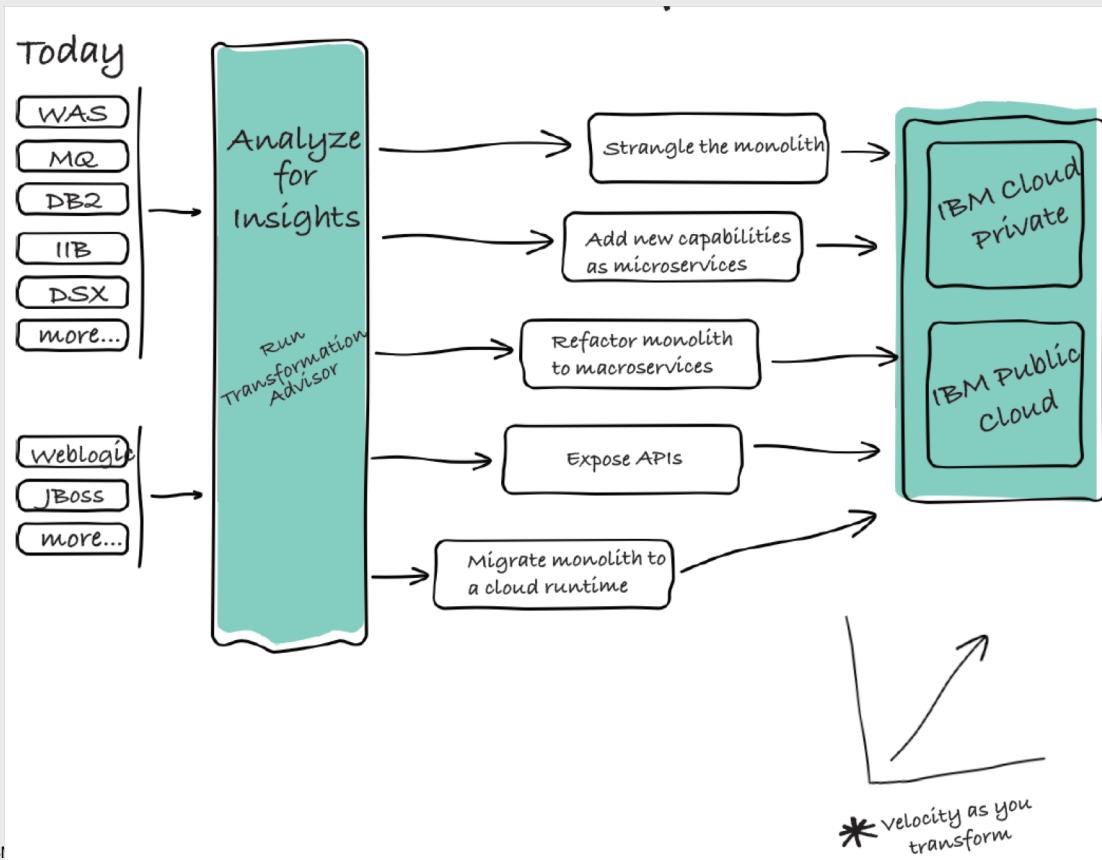


# **AppMod @3000ft**



**Modernize  
Apps**

# Application modernization journey



**Containerize the monolith.** Reduce costs and simplify operations.

**Expose on-prem assets with APIs.** APIs enable legacy assets that are difficult to cloud enable.

**Refactor into microservices.** Break down monoliths into deployable components.

**Add new microservices.** Innovate incrementally and establish success early.

**Strangle the monolith.** Incrementally sunset the monolith.

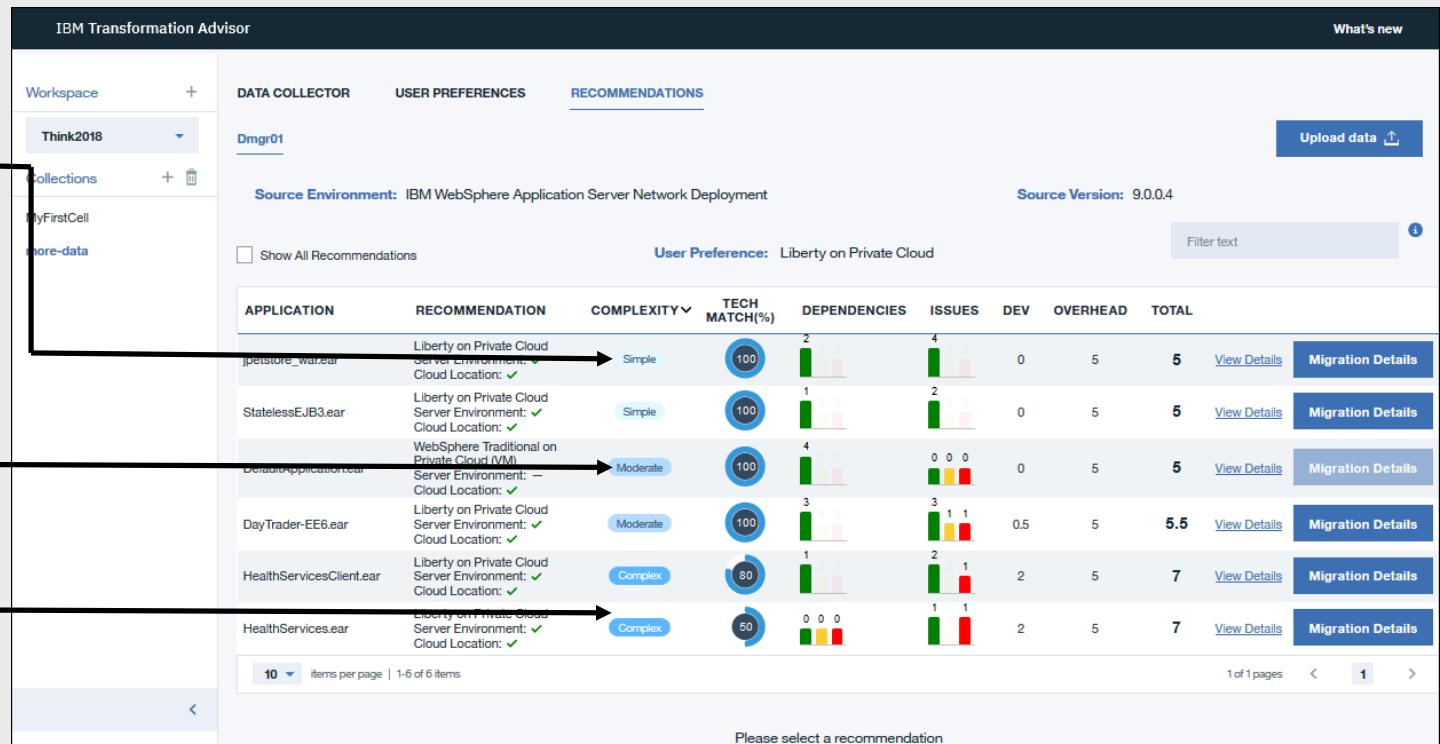
# IBM Transformation Advisor

Helps classify existing apps based on their modernization complexity

**Simple:**  
No code changes

**Moderate:**  
Some refactoring  
needed

**Complex:**  
May decide to run  
in WAS in VMs before  
re-engineering



# IBM Transformation Advisor

Generates artifacts to help you automate deployment onto IBM Cloud Platform(s)

The screenshot shows the IBM Transformation Advisor interface. On the left, there's a sidebar with a 'Move to Liberty' button and a 'Containerize and deploy' button. A bracket from these buttons points to the 'Migration Files' section in the main content area. The main content area displays a 'Your migration bundle is ready' message and a 'Migration Bundle' section. It lists 'Included in the bundle:' under 'Migration Files' (server.xml, Dockerfile, Helm Charts, deployment.yaml) and 'Application Dependencies' (APPLICATION Binary: jpetstore.war). To the right, there's an 'APPLICATION' section for 'jpetstore\_war.ear' (IBM WebSphere Application Server Network Deployment), a 'MIGRATE TO' section for 'IBM WebSphere Liberty on IBM Cloud Private', and a large blue 'Download Bundle' button. Below that is a 'HOW IT WORKS' section with two steps: ① Migrate to IBM WebSphere Liberty (with an icon of a stack of books) and ② Containerize WebSphere Liberty (with an icon of a stack of containers).

IBM Transformation Advisor

← Recommendations

Your migration bundle is ready

MIGRATION BUNDLE

The files included in your migration bundle help your migration to IBM WebSphere Liberty, create an image and help you package your application as a helm chart for easy deployment.

Included in the bundle:

Migration Files ⓘ

- ✓ server.xml ⓘ Download
- ✓ Dockerfile ⓘ Download
- ✓ Helm Charts ⓘ Download
- ✓ deployment.yaml ⓘ Download

Application Dependencies ⓘ

- ✓ APPLICATION Binary ⓘ jpetstore.war ⚙

APPLICATION

jpetstore\_war.ear

IBM WebSphere Application Server Network Deployment

MIGRATE TO

IBM WebSphere Liberty on IBM Cloud Private

Download Bundle

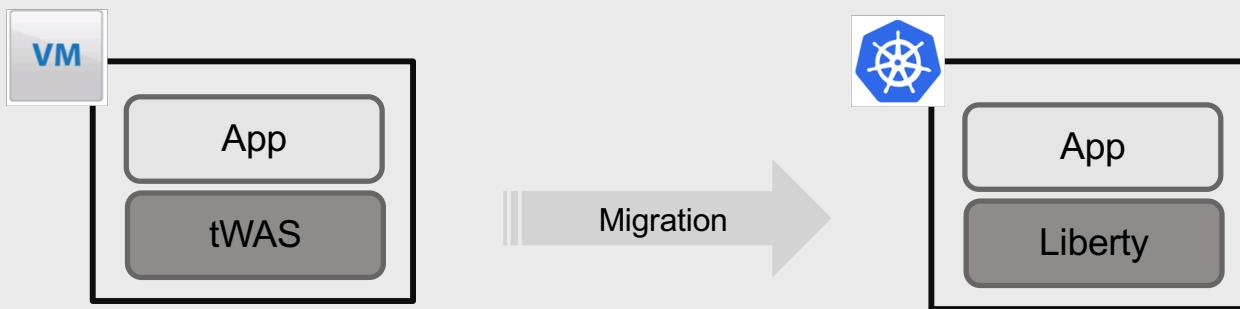
HOW IT WORKS

View the steps →

① Migrate to IBM WebSphere Liberty

② Containerize WebSphere Liberty

# Containerized Monolith



What

- Containerized runtime and middleware
- As little application change as possible
- Keep integrations and data on-prem
- Self-service developer access

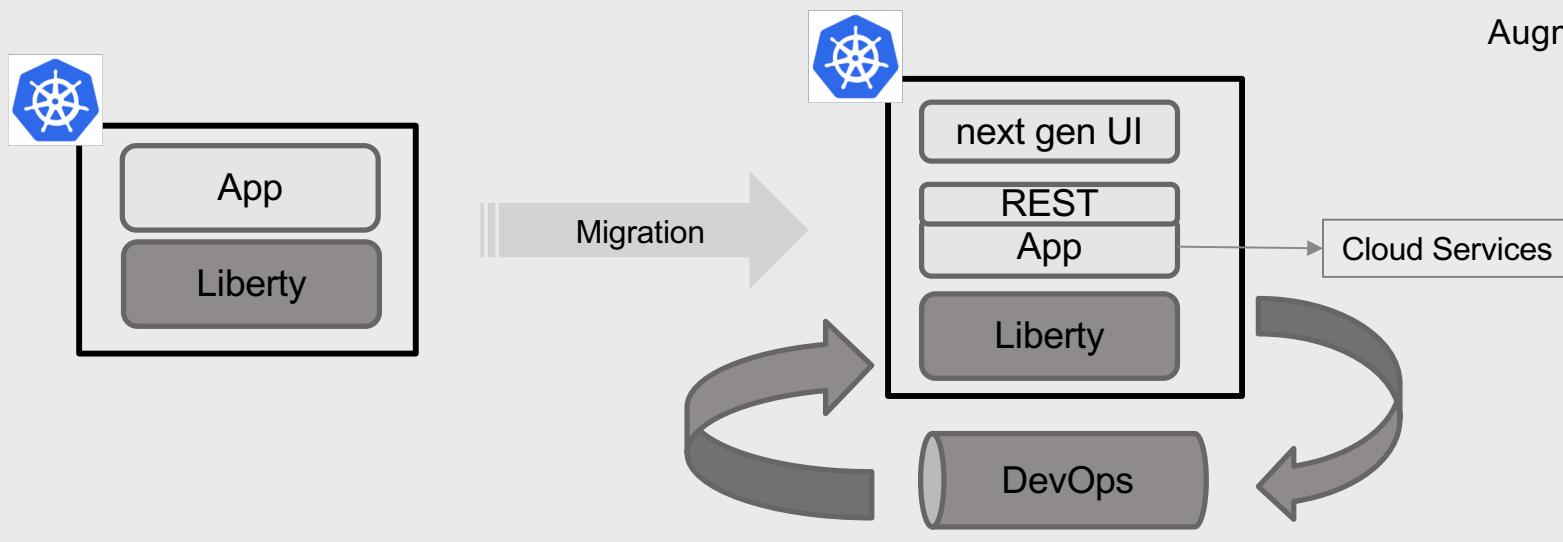
Why

- Runs on cloud
- Small runtime/ Fast startup
- “Zero” future version migration
- Cloud portability
- Lower operational costs (higher runtime density)

# Fast Monolith



Augment



## What

- Legacy is preserved, new capabilities added
- Modern DevOps
- Modern development tools

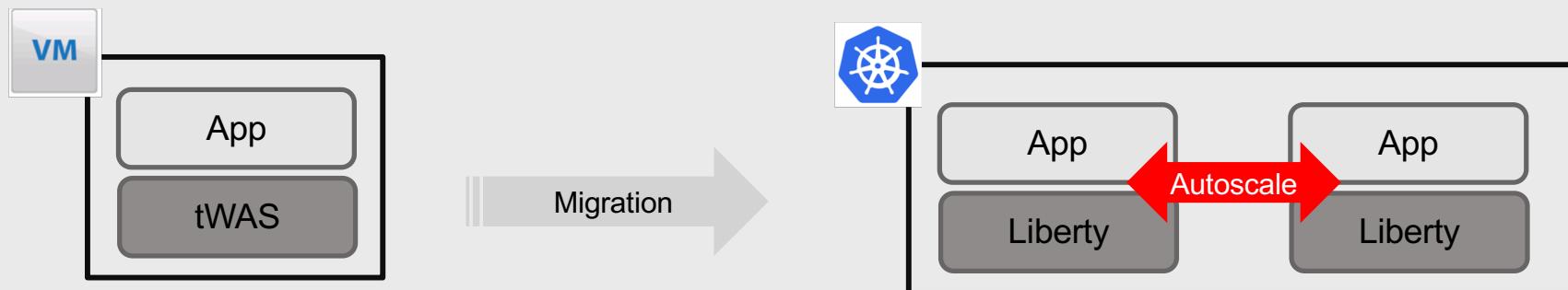
## Why

- Accelerate delivery cycle: months → week(s)  
Constant innovation
- next gen UI experience
  - Cloud services

# Scalable Monolith



Refactor



## What

- Augmented Monolith: 12 Factors
- Data/integrations stay on-prem

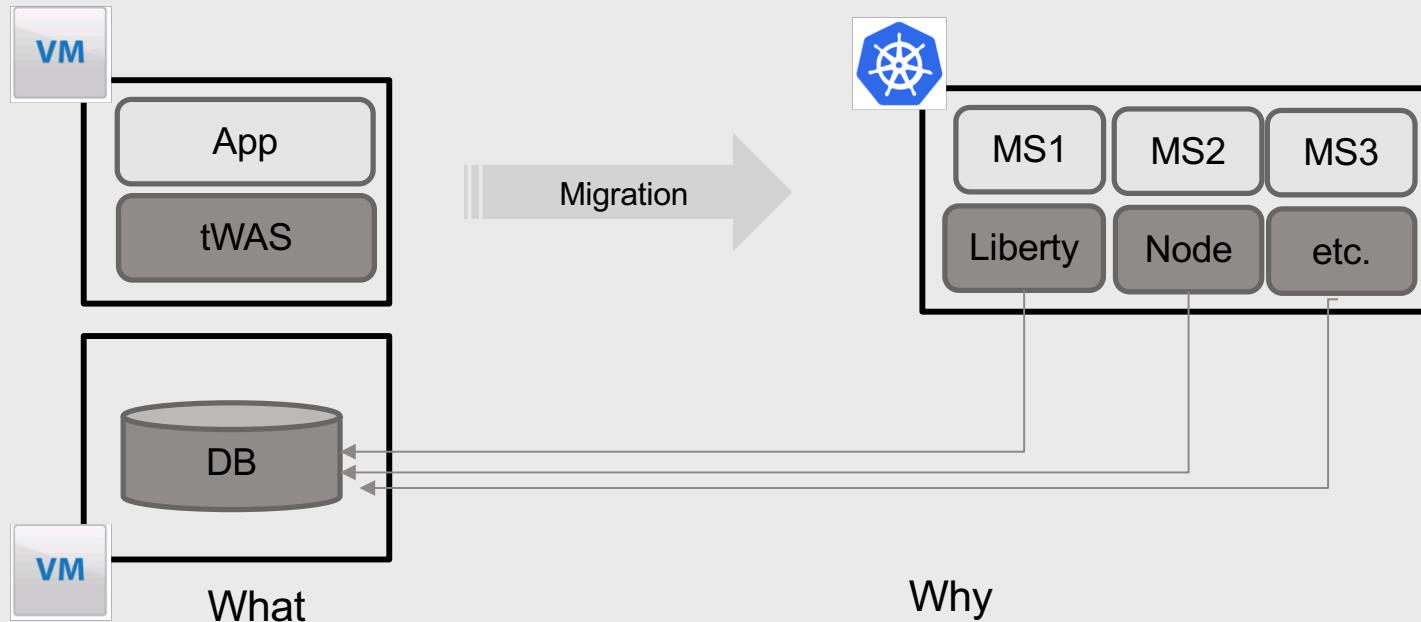
## Why

- Horizontally scalable
- Platform provided HA
- Substantial operational efficiencies

# Macro-services aka “Microlith”



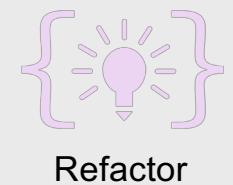
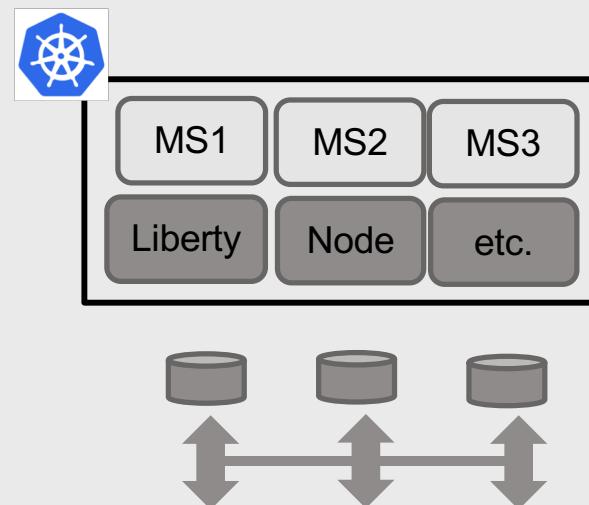
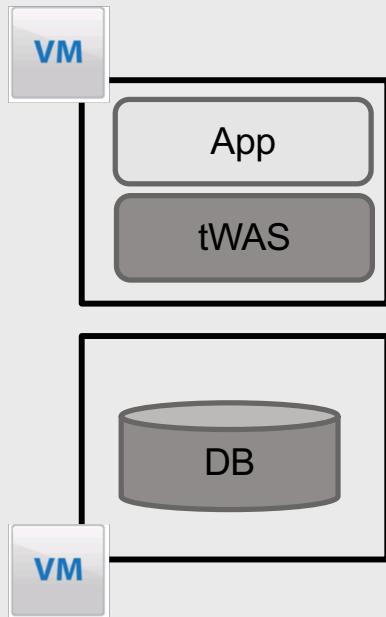
Refactor



- Break down application monolith into independently deployable modules
- Data remains a monolith

- “Half-way house” to microservices

# Microservice



Refactor

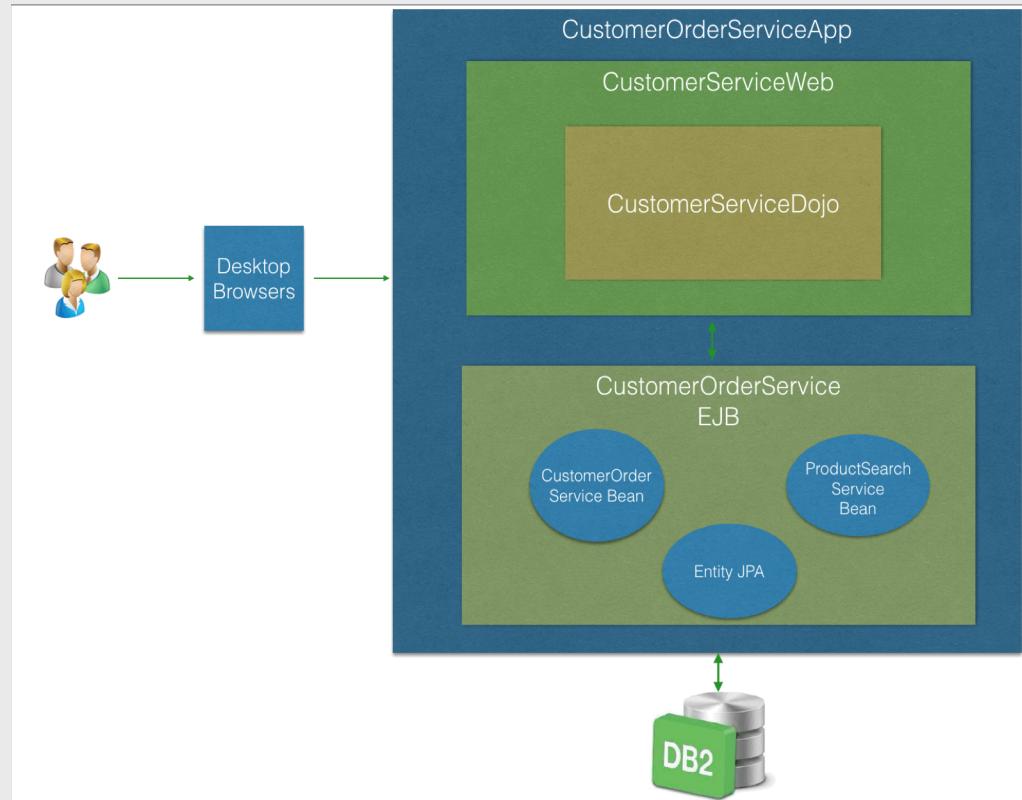
# Example: Refactoring a monolith to microservices

## Original architecture

- Single relational DB schema
  - Products, Customer, Ordering, etc.
- Single deployment unit (EAR file)

## Business problems

- Limited searching capabilities for products.
- Little knowledge of customer for targeted experience
- Ordering system is complex
  - Difficult to add product and customer analytics to site without breaking Order System.



# Modernization outcomes

Catalog data imported into Elastic Search

- Clients delighted by new fuzzy search

Customer data modeled and stored in document NoSQL store with analytic and social data

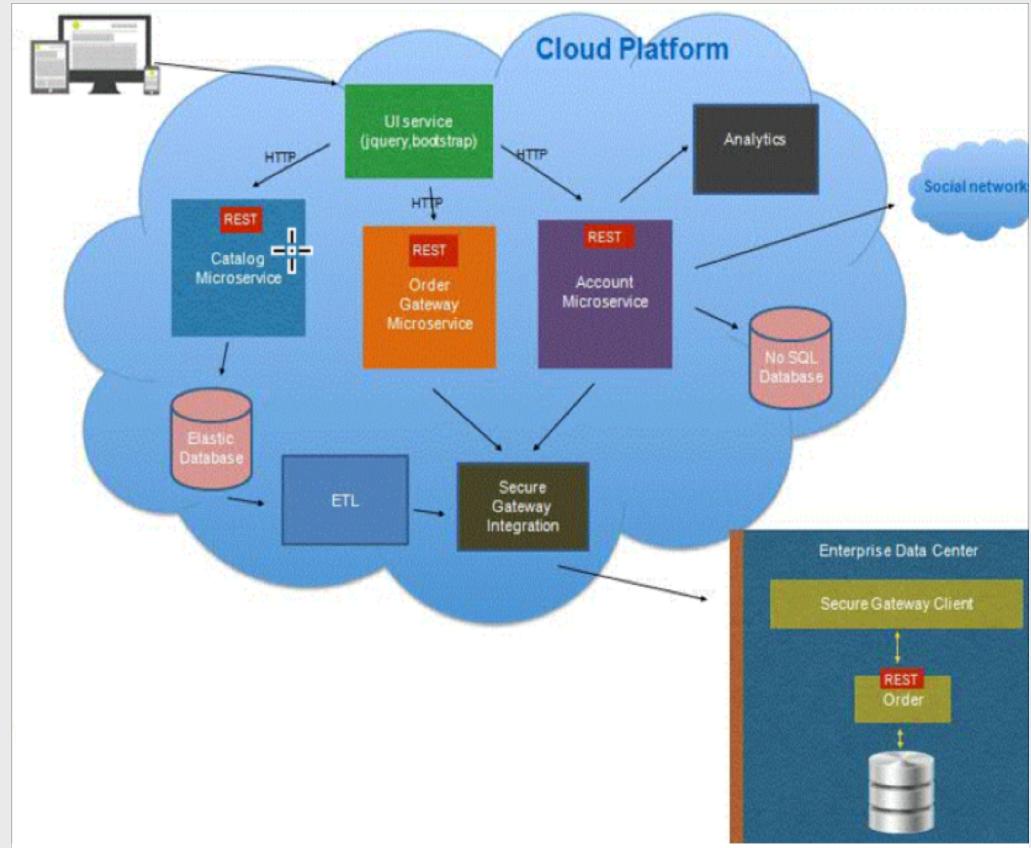
- Improved targeted experience

Order microservice wraps on-prem ordering and uses integration

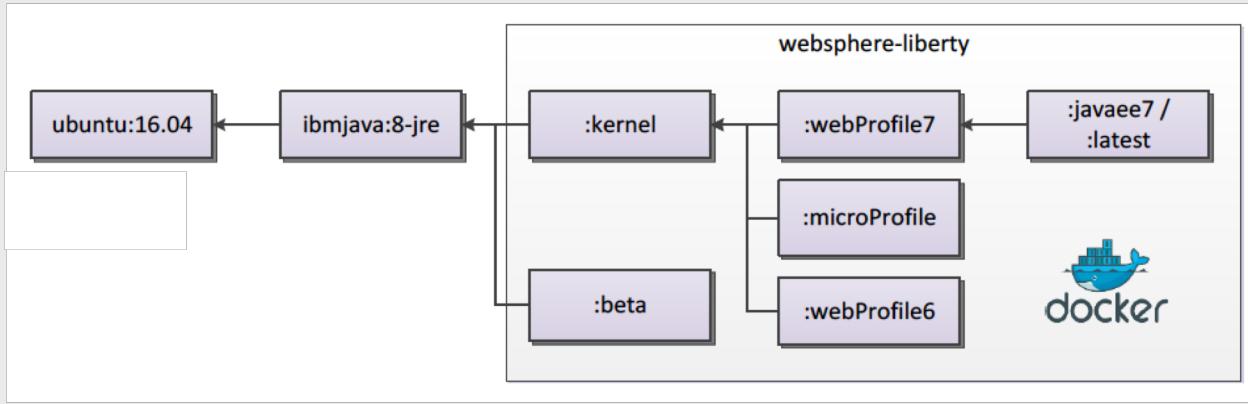
Omnichannel functions:

- New Mobile App uses new microservices
- Existing Website used with routing /

Strangler pattern to evolve



# Why WebSphere Liberty?

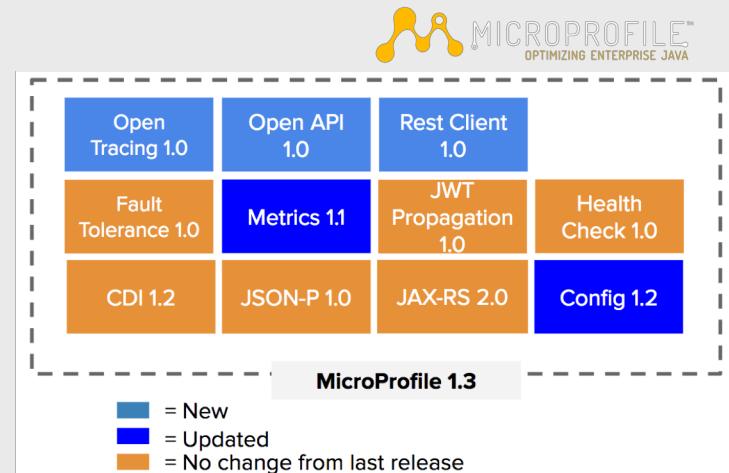


**Migrate** to Cloud  
enabled full Java EE7  
runtime

Migration automation  
and acceleration with  
Transformation Advisor

**Modernize** with MicroProfile for microservice refactoring

**Open Liberty** - open source Liberty with Java EE7 and MicroProfile capabilities



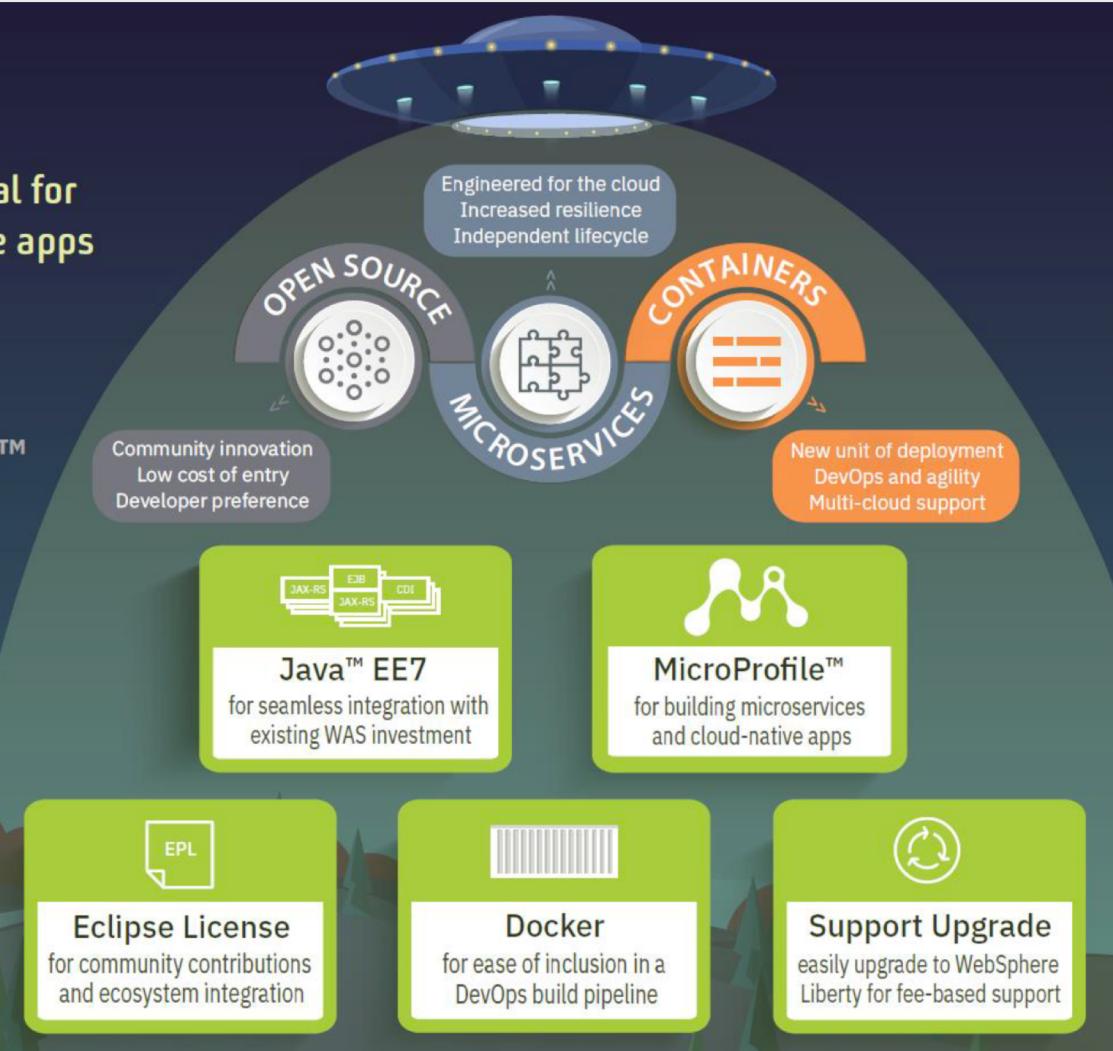


# Open Liberty

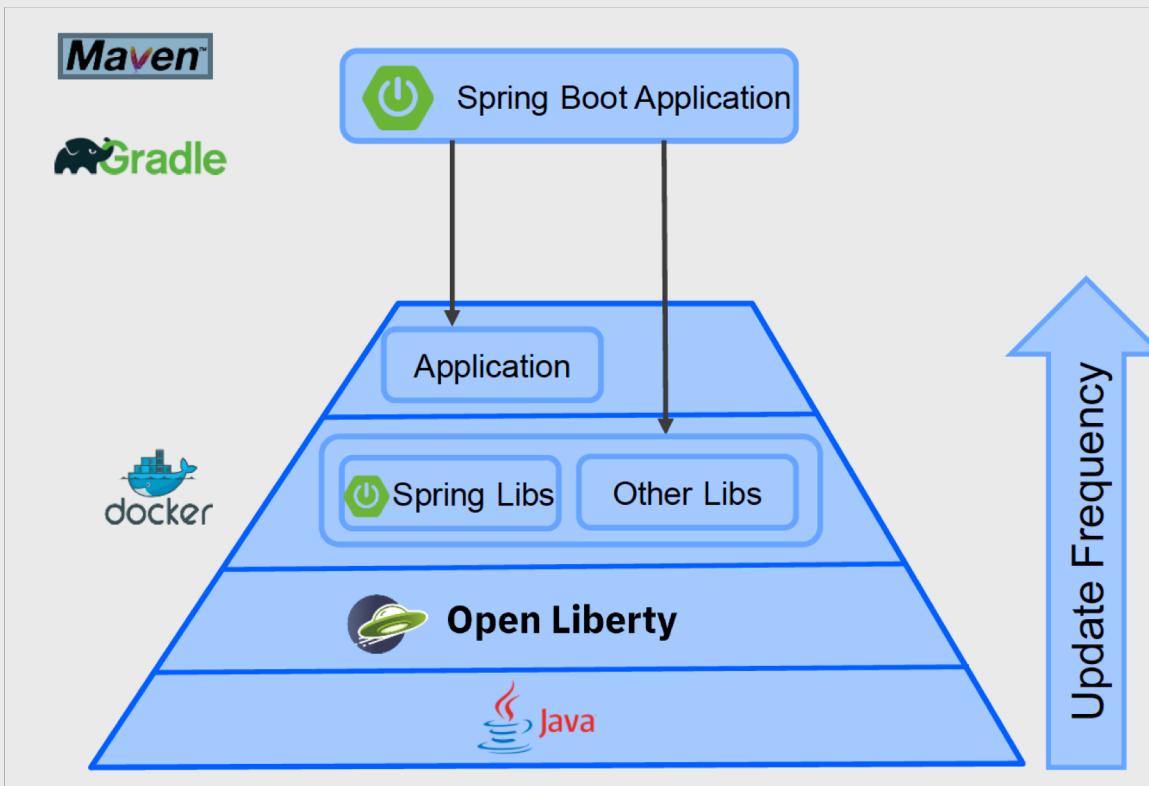
A lightweight open source server runtime ideal for building Java™ microservices and cloud-native apps

- Easy to consume
- Deploy on any cloud for Java™
- Seamlessly transition to WebSphere

<https://openliberty.io/>



# Spring Boot applications on Liberty



Liberty Spring Boot Starter to put Liberty in your Boot app.

Or get modern and skinny down your fat boot jar:

- Deploy Spring Boot Apps on Open Liberty without change
- Use the tools of your choice
- Automatically manage dependencies separately from the app content
- Optimized Docker deployment by keeping the application layer small and efficient to update

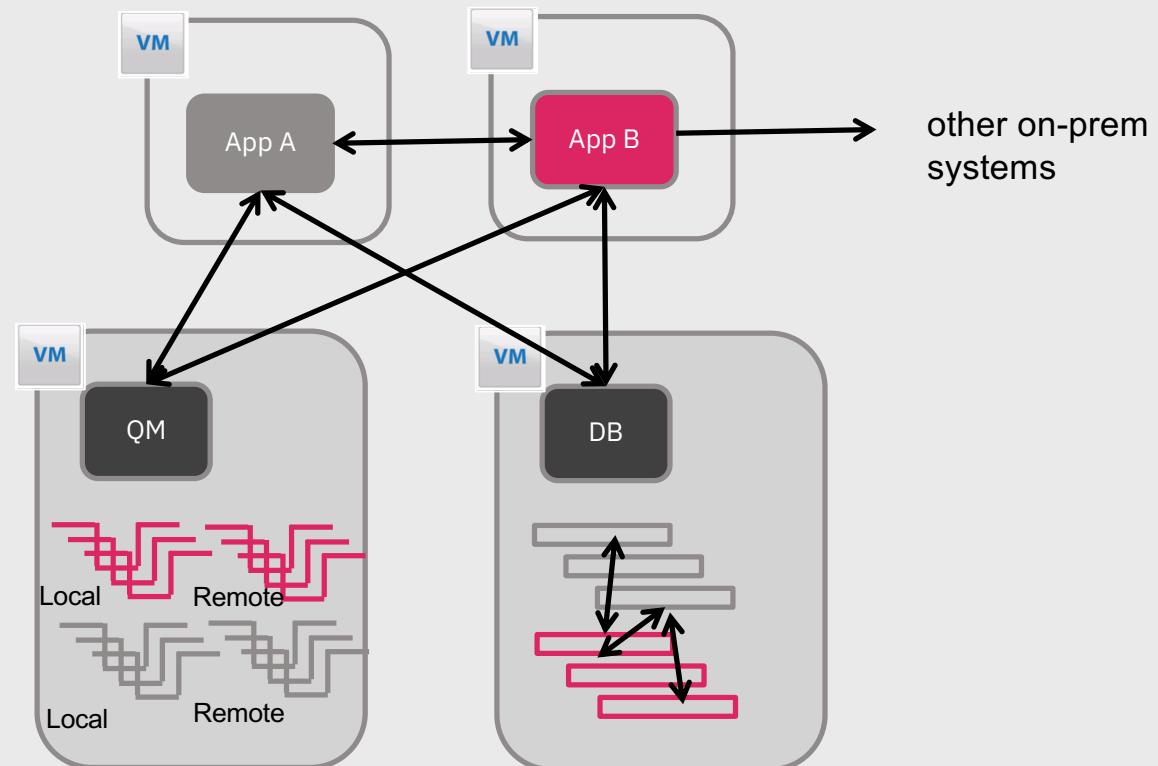
# **AppMod @3000ft**



**Modernize  
Middleware**

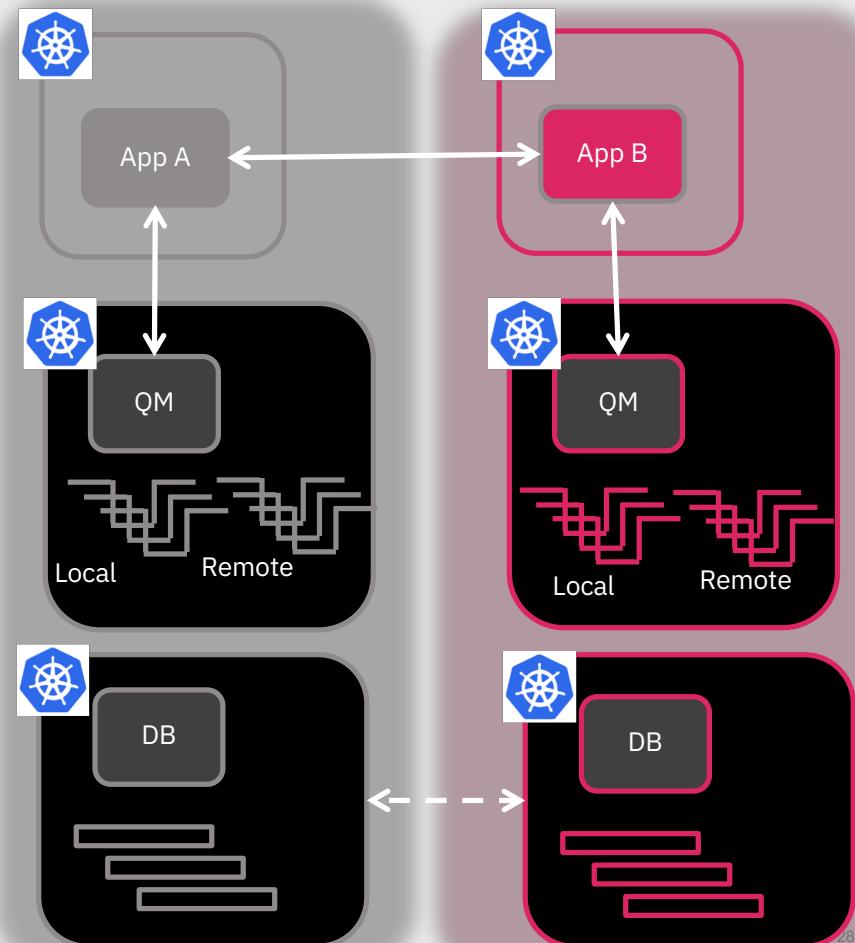
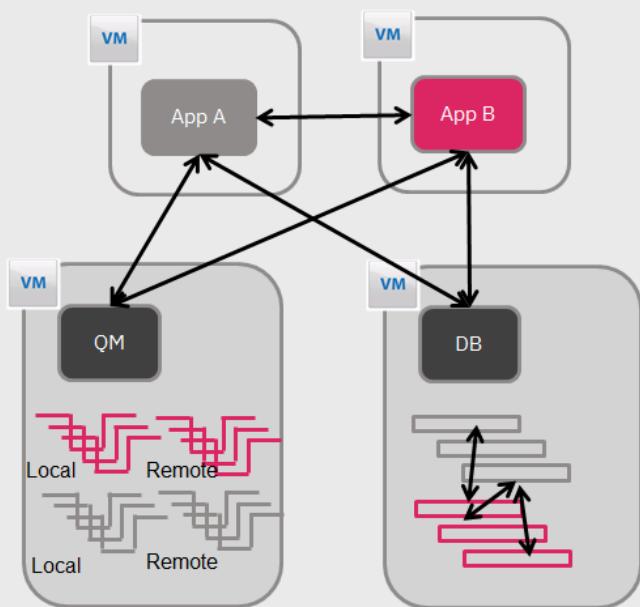
## Traditional middleware deployment “Hairball”

**.. is what business  
calls “an app”**

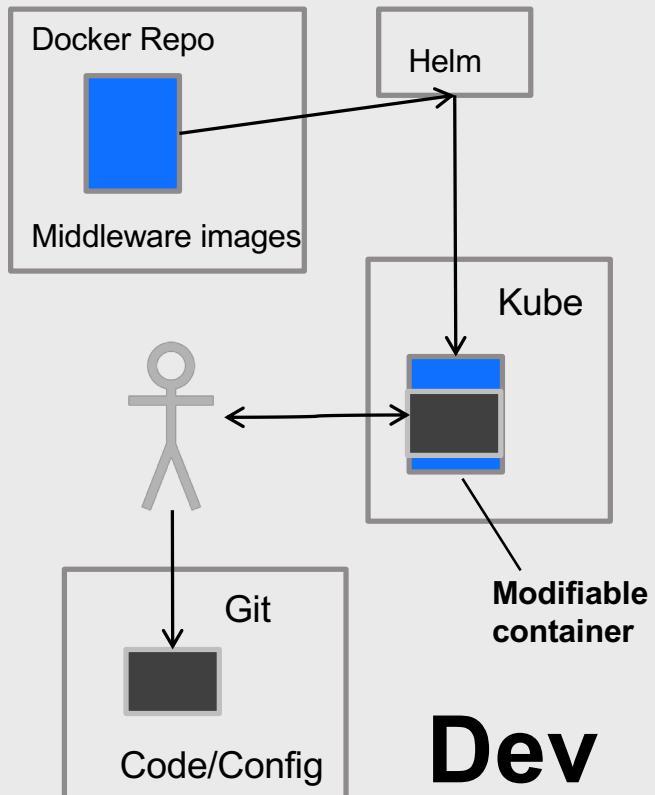


Data “hairball” to  
rule them all

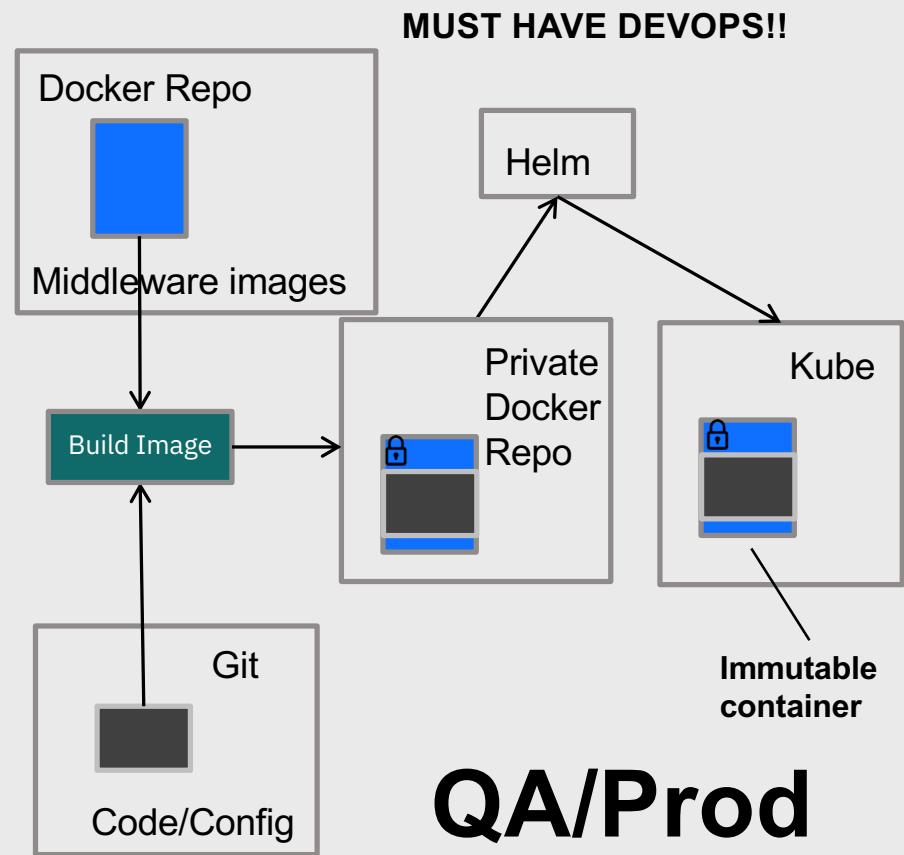
# Untangling the hairball with containerized middleware microservices



# Deploying containerized middleware

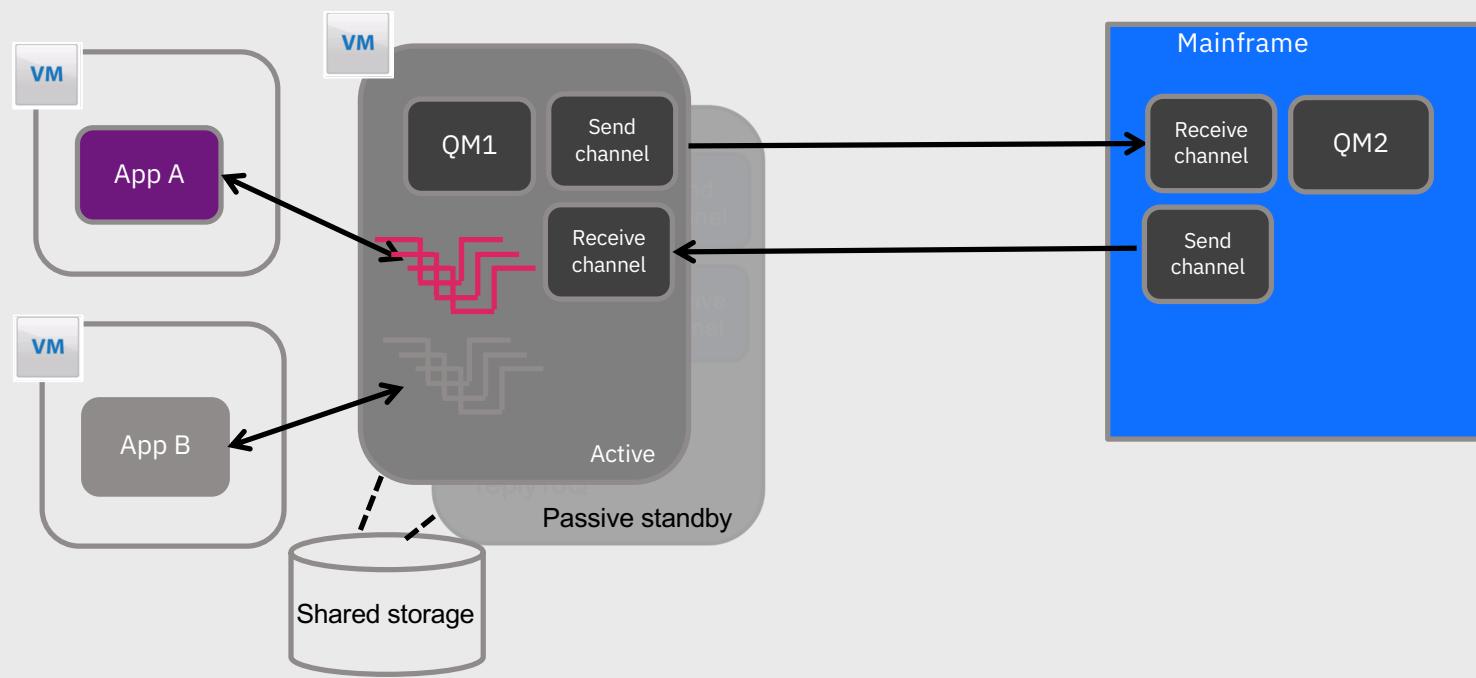


**Dev**

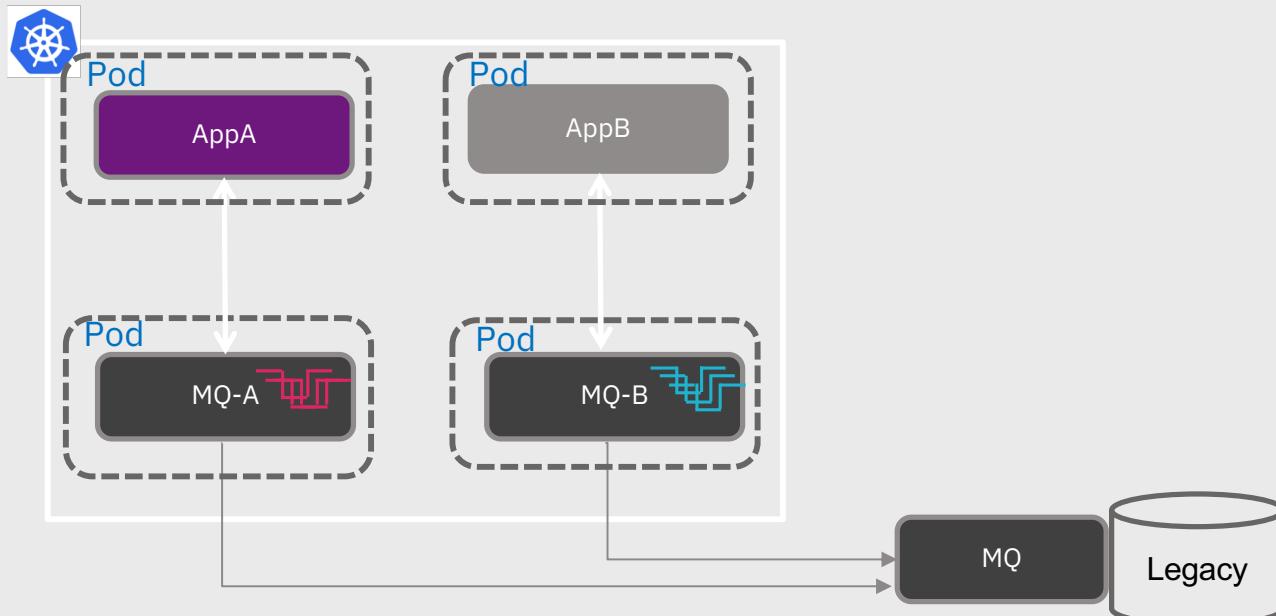


**QA/Prod**

## App is MQ Client – Traditional setup



# Disentangled messaging – MQ “microservices”



What

- Un-share MQ servers
- Use dedicated MQ containers for each app client
- Failover is just rescheduling MQ pod

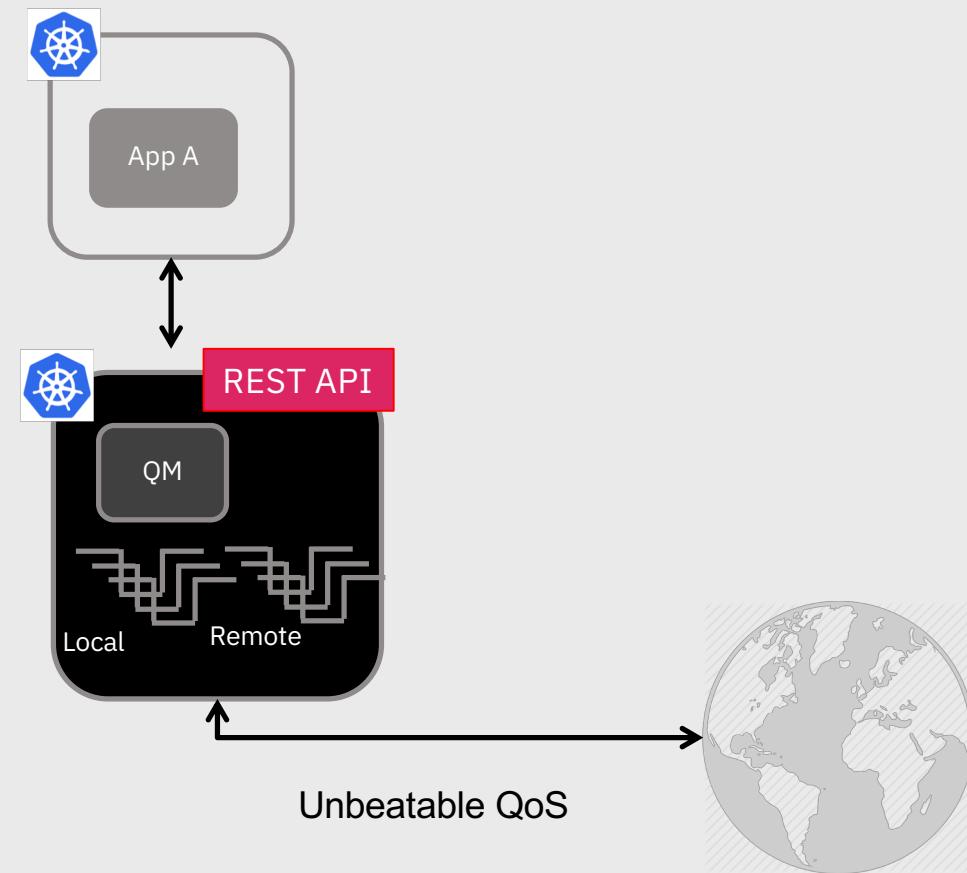
Why

- Running containers are immutable
- Cannot have “MQ admin” manage queues in running instance

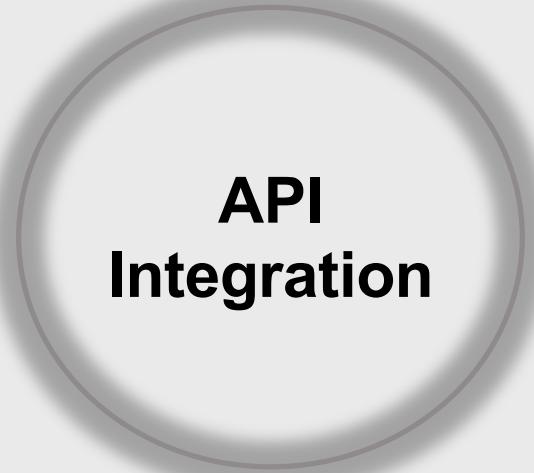
# Modernize cloud development experience with enterprise middleware microservices



- Do not want MQI, JMS
- Need a queue endpoint
- Advanced MQ TL;DR
- MQ REST API in 9.0.x



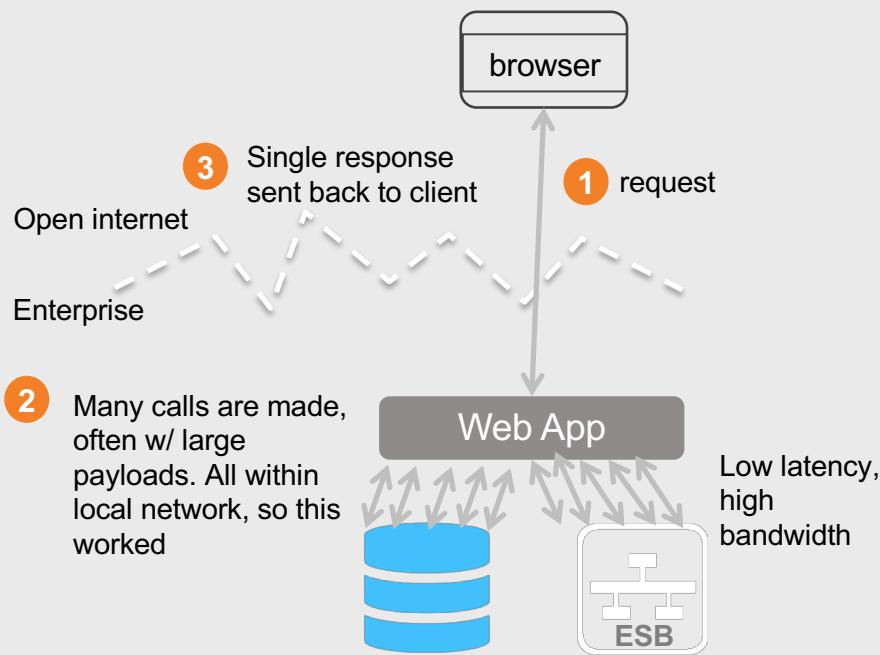
# **AppMod @3000ft**



**API  
Integration**

# Digital apps present new integration challenges

Traditional web apps

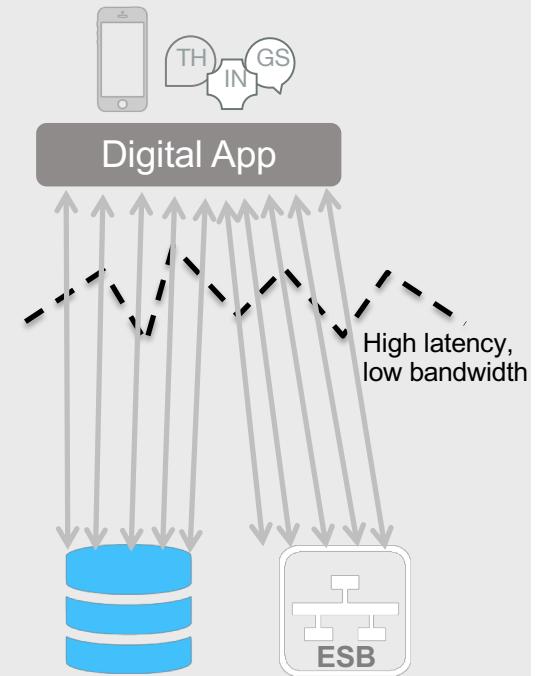


If digital apps used same approach...

Digital apps live in the internet (phones, IoT, dynamic web pages)

High latency make traditional integration approach untenable.

Cannot simply reuse existing services for new digital apps – need a new approach

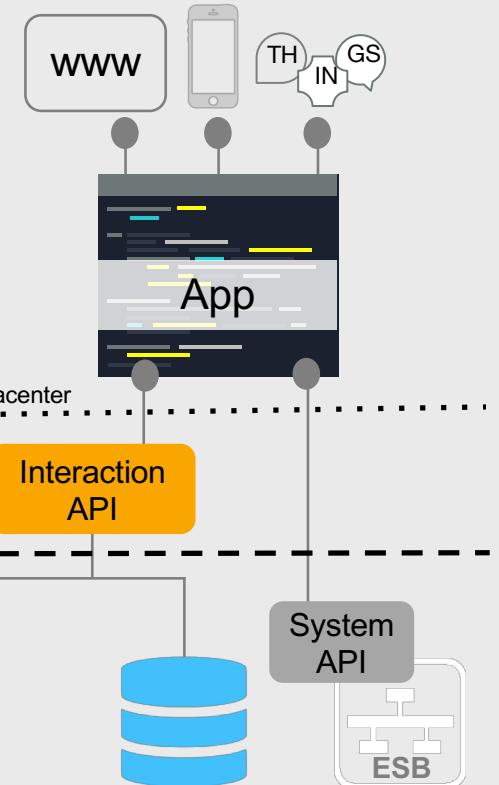


# Digital applications require a new interaction tier

## Digital Team / Line-of-Business

Measured on time to market. Motivated to be fast (e.g. “get it out, fix it in market”).

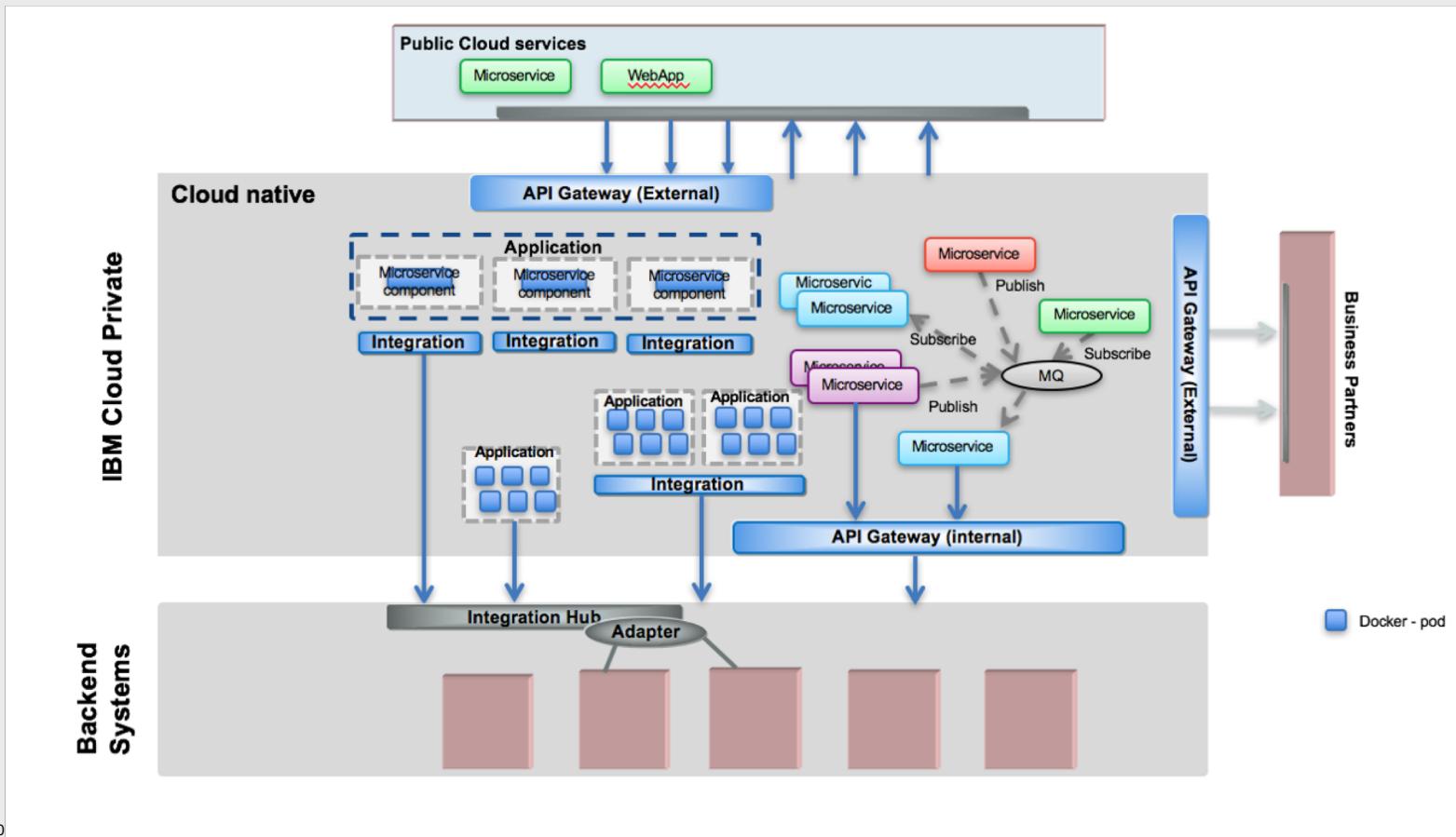
- Cannot simply call existing System API's from digital apps for performance reasons (see previous slide)
- Cannot afford to wait weeks/months for a new System API that exactly meets their needs
- **Needs a NEW tier** which sits close to the systems of record, but is controlled by the Digital Team/LoB



## Central IT Team

- Measured on resiliency & uptime.
- Motivated to be cautious.
- Generally takes weeks/months to turn around change requests (e.g. to support digital apps), due to change control & quality control processes.

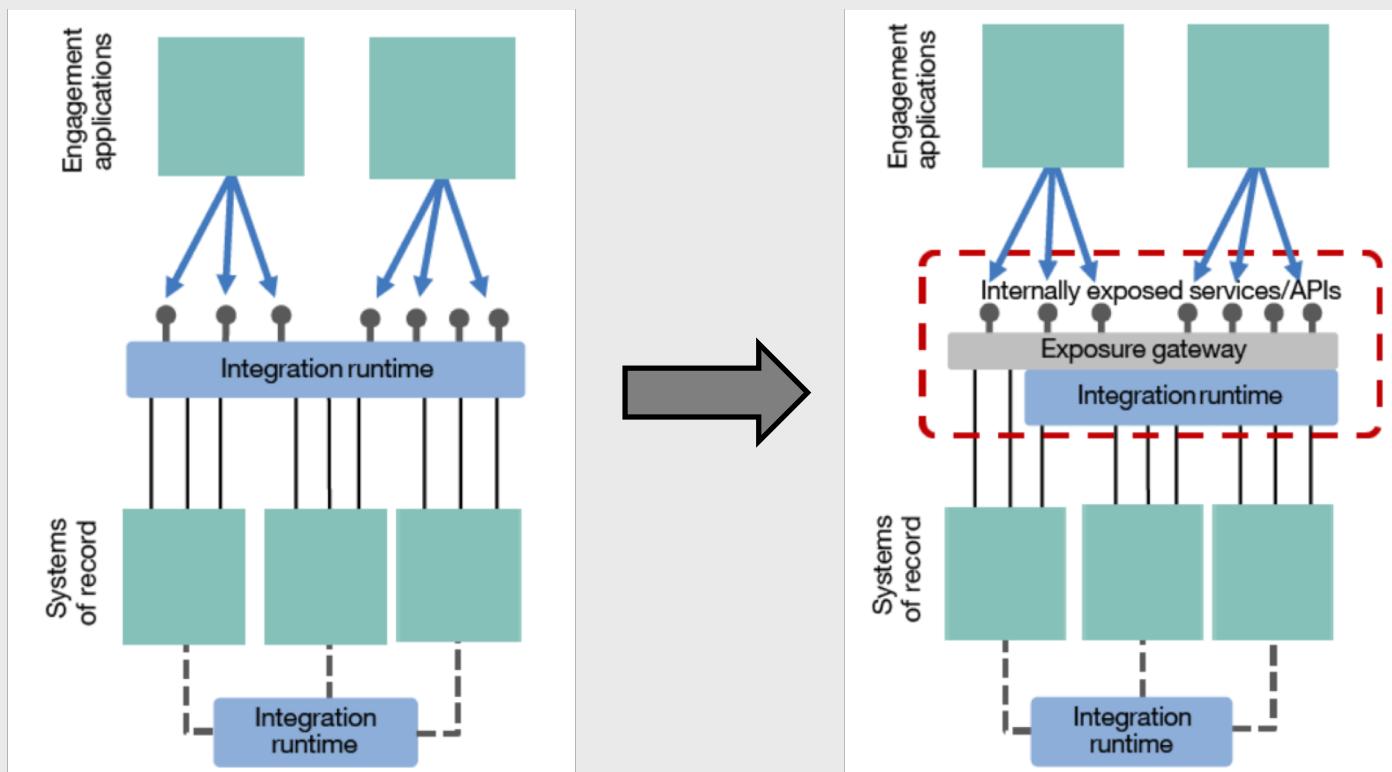
# API Integration Patterns



# Enhancing existing ESB pattern



Augment

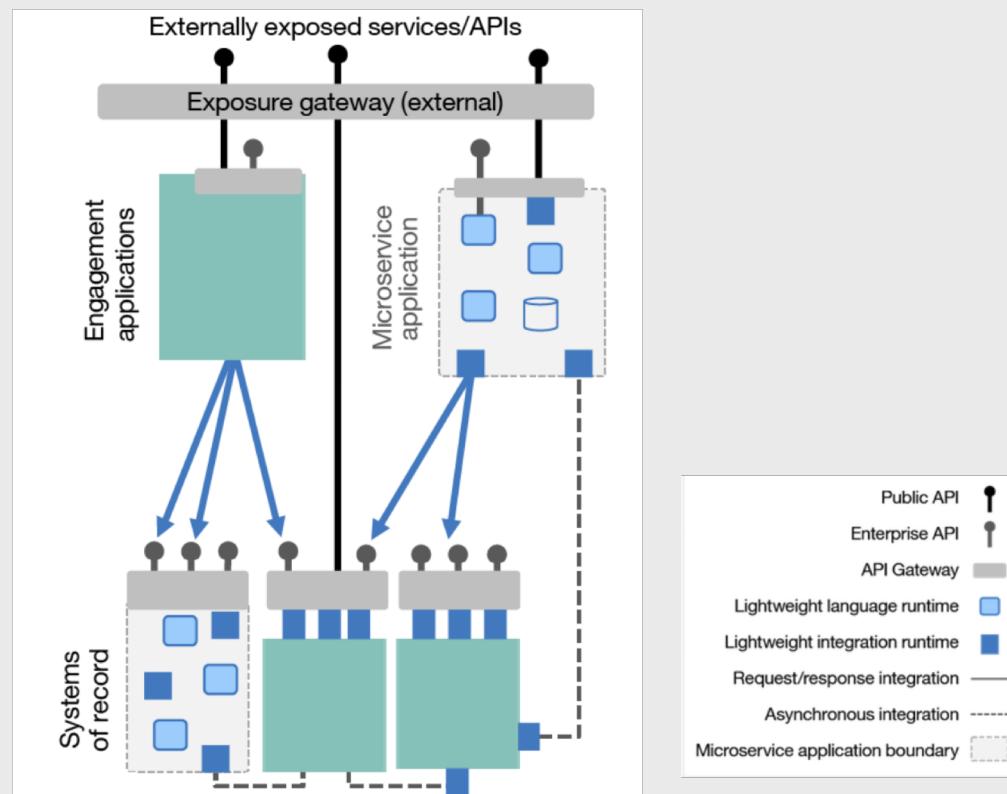


<https://www.ibm.com/developerworks/cloud/library/cl-lightweight-integration-1/>  
IBM Cloud / © 2018 IBM Corporation

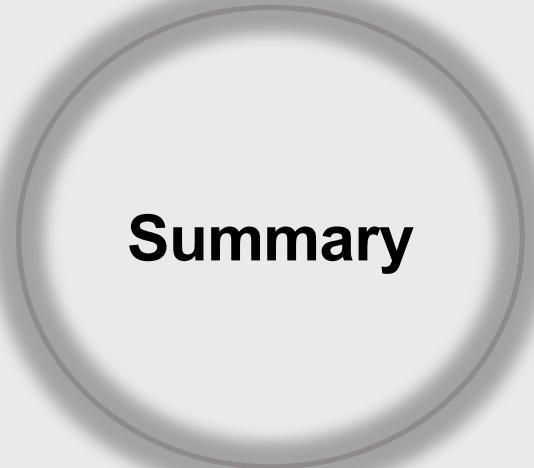
# Breaking up the centralized ESB into independently maintainable and scalable pieces



Refactor

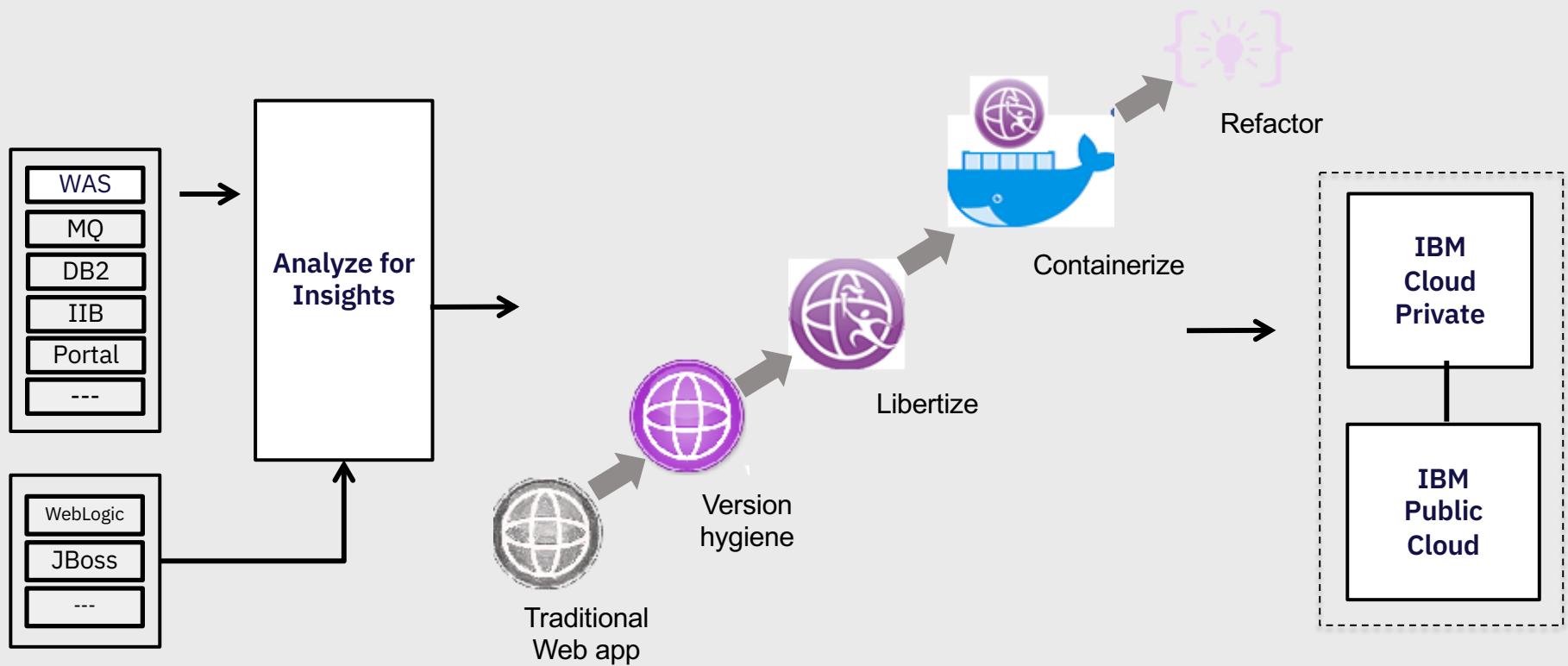


# **AppMod @3000ft**

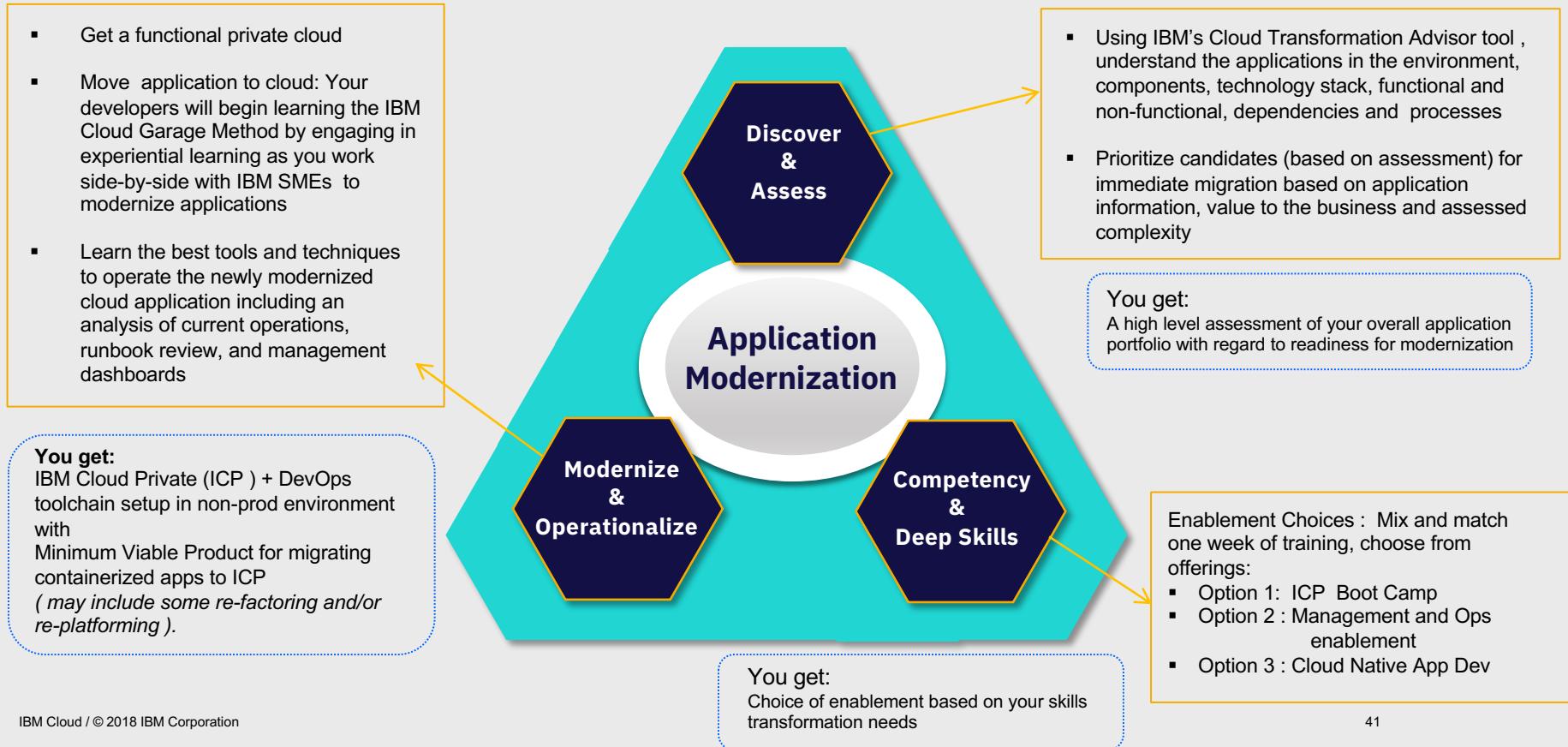


**Summary**

# Accelerate the cloud adoption journey



# Garage Services for Application Modernization



# IBM Transformation Advisor

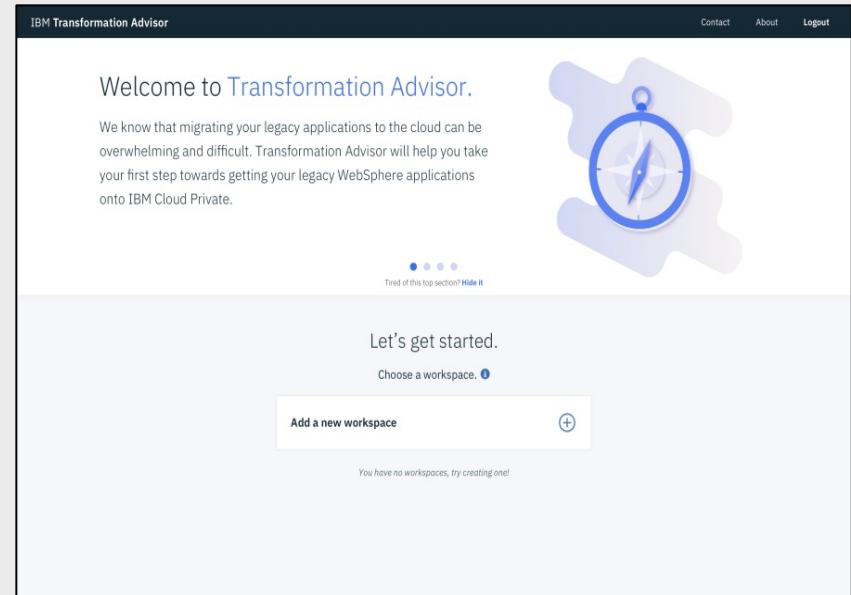
Collects information about an existing WebSphere environment and applications.  
Combines that with rules and insights gained from years of working with WebSphere and  
WebSphere applications, and provides recommendations.

## Challenges addressed:

- Leveraging existing application logic
- Need to accelerate application development and maintenance
- Monolithic applications that are complex and tightly coupled

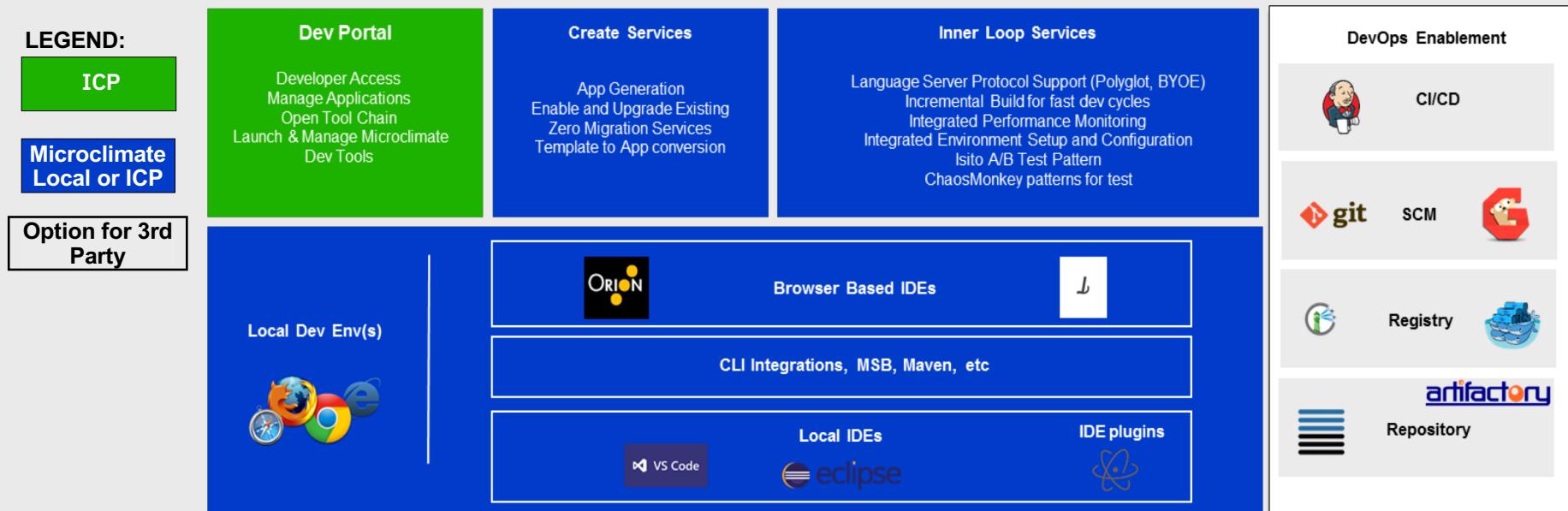
## Benefits:

- Included and deployed on IBM Cloud Private
- Introspects existing WebSphere Deployments
- Provides recommendations, guidance and artifacts for deployment in Liberty containers and Kubernetes clouds

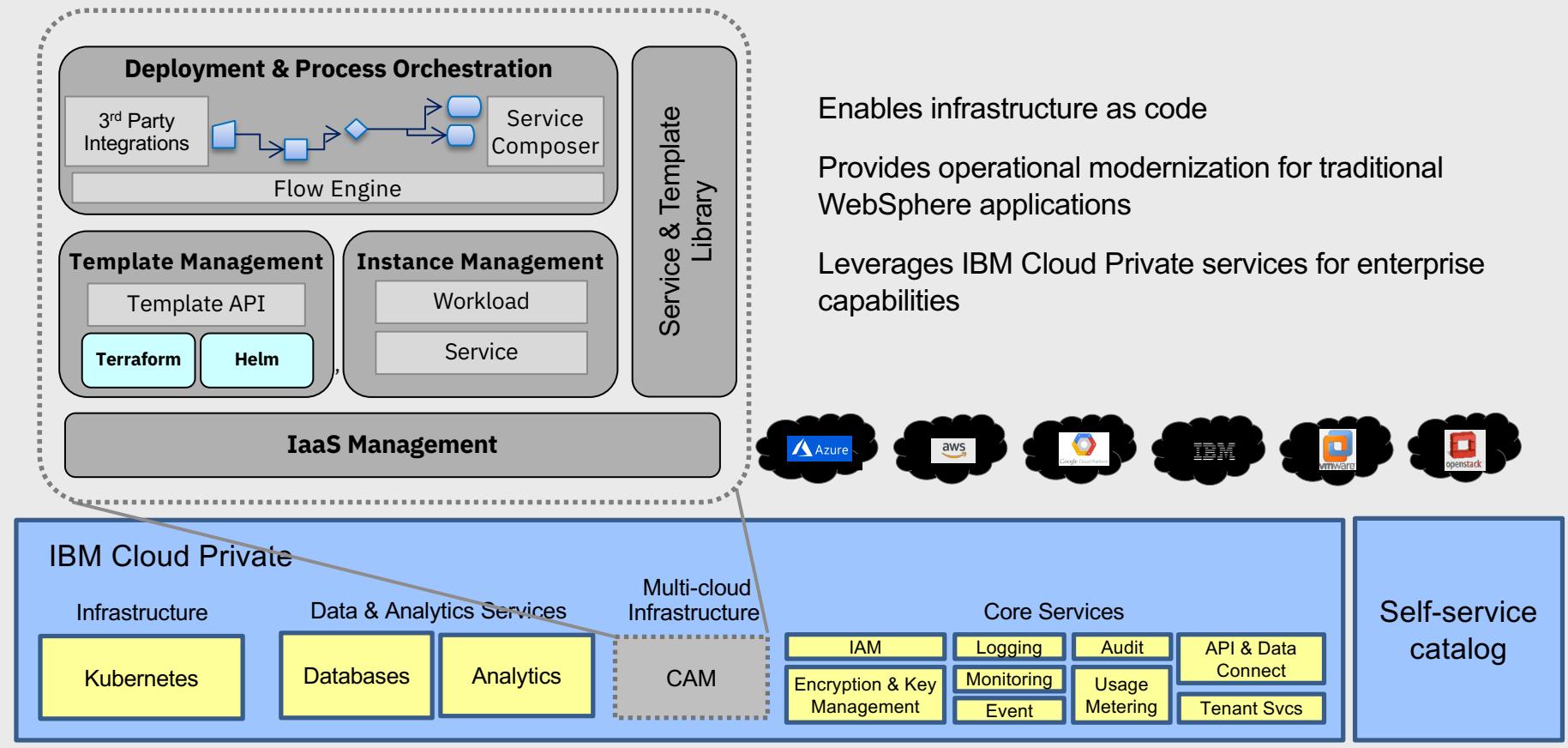


# IBM Microclimate

Microclimate is an end-to-end development environment that lets you rapidly create, edit, and deploy applications. Applications run in containers from day one and can be delivered into production on Kubernetes through an automated DevOps pipeline using Jenkins.



# Cloud Automation Manager



# References

IBM Transformation Advisor

<https://developer.ibm.com/app-modernization/>

Microclimate

<https://www.ibm.com/us-en/marketplace/microclimate>

WebSphere Liberty

<http://wasdev.net/>

IBM Cloud Automation Manager (CAM)

<https://developer.ibm.com/cloudautomation/>

Free online Garage Method Courses

<https://www.ibm.com/cloud/garage/content/course/websphere-on-cloud-private>

IBM App Modernization Field Guide -

<https://www.ibm.com/cloud/garage/content/culture/app-modernization-field-guide/>

WebSphere Application Server Migration Toolkit

<https://www.ibm.com/developerworks/library/mw-1701-was-migration/index.html>

