

Red Hat  
**Summit**

**Connect**

Modernizziamo le applicazioni

**Francesco Rossi**  
Solution Architect

**Matteo Grimaldi**  
Solution Architect

# Modernizing Applications

Are all modernization projects successful ?

**79%**  
**App  
Modernization  
Efforts Fail**

**\$1.5M**  
**Average Cost  
of a  
Modernization  
Project**

**16mo**  
**Average Time  
of a  
Modernization  
Project**

# Modernizing Applications

Why they might fail ?

**3**

**Failure to accurately set expectations**

**2**

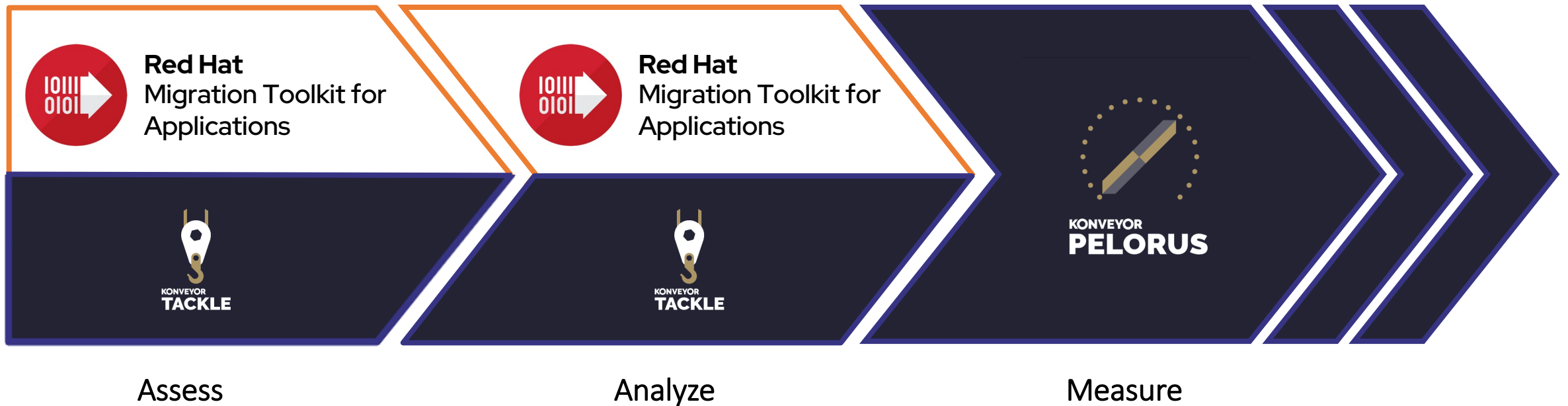
**Inadequate skills or training and complexity**

**1**

**Lack of intelligent tools**

# Konveyor and Red Hat Migration Toolkits

Red Hat contributions and products



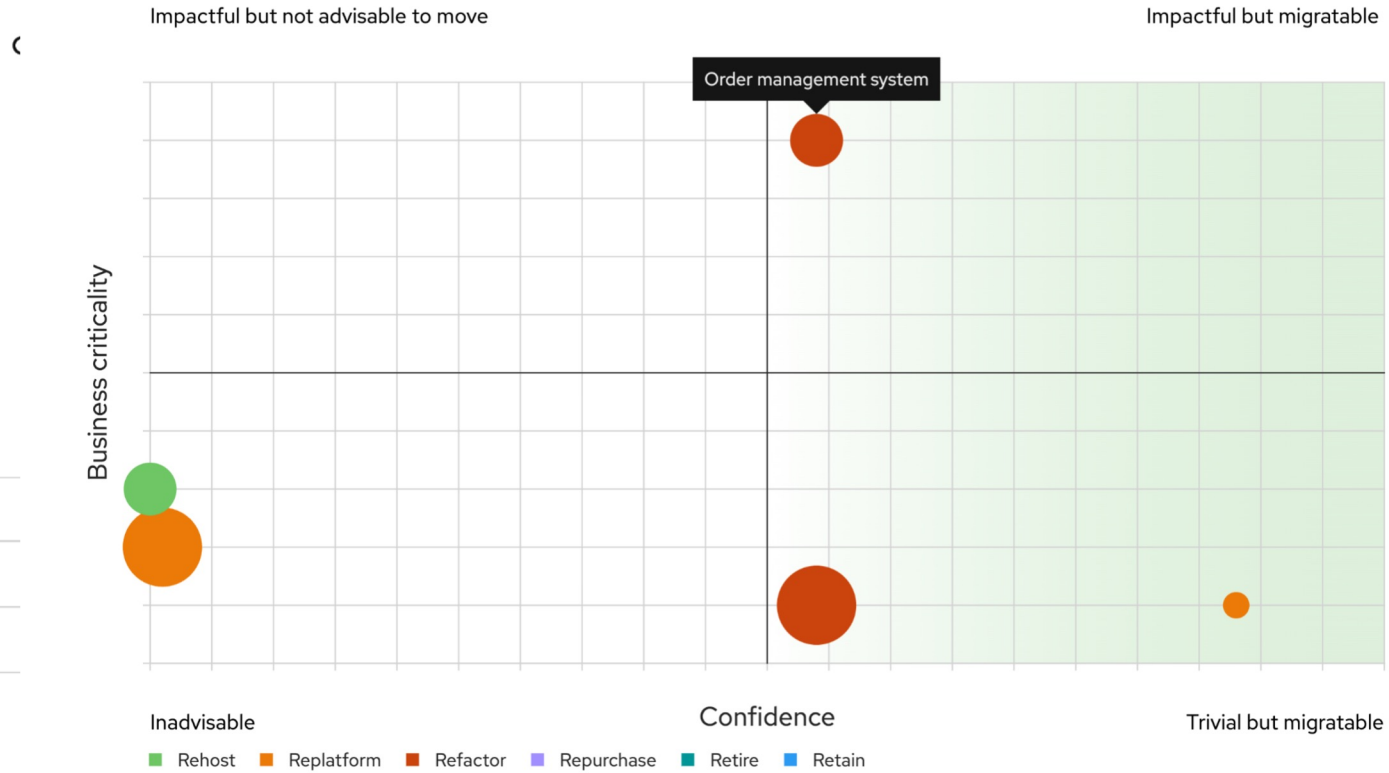


# Assessing the Tree of Taste Application Inventory

Adoption candidate distribution

Table view | Graph view

Dependencies

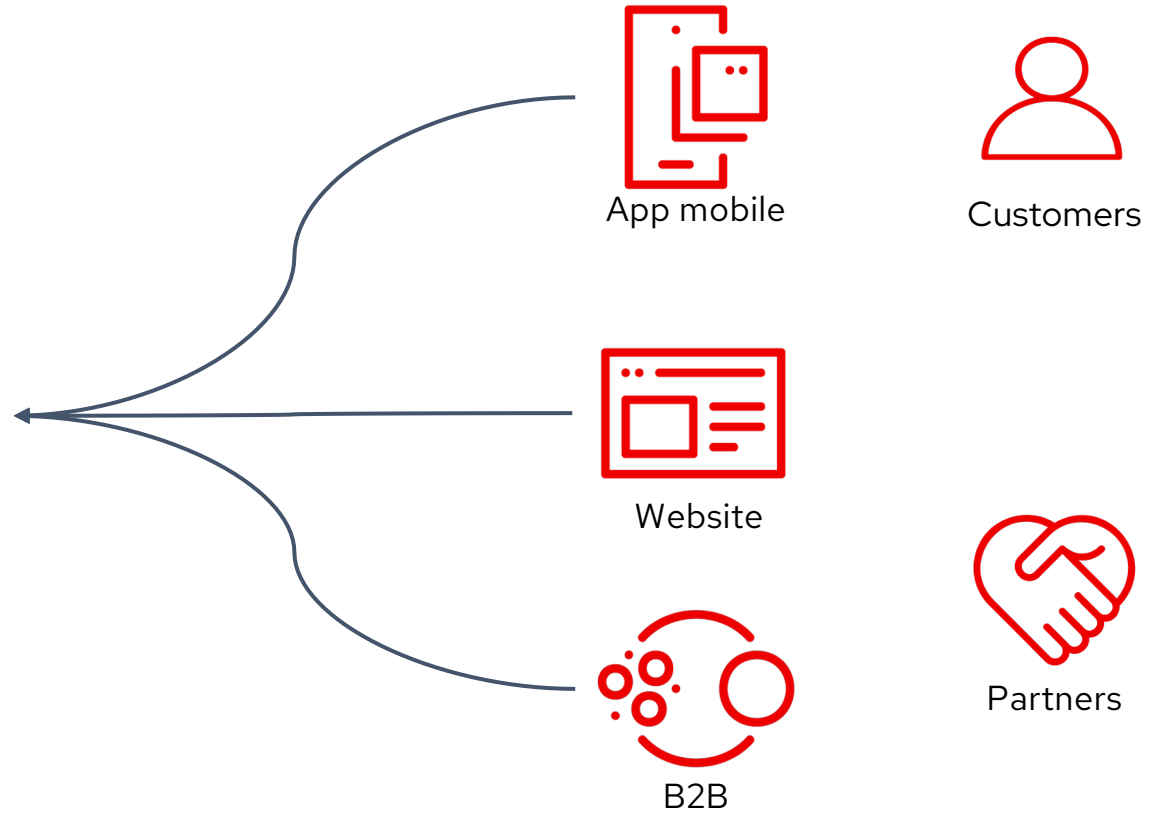
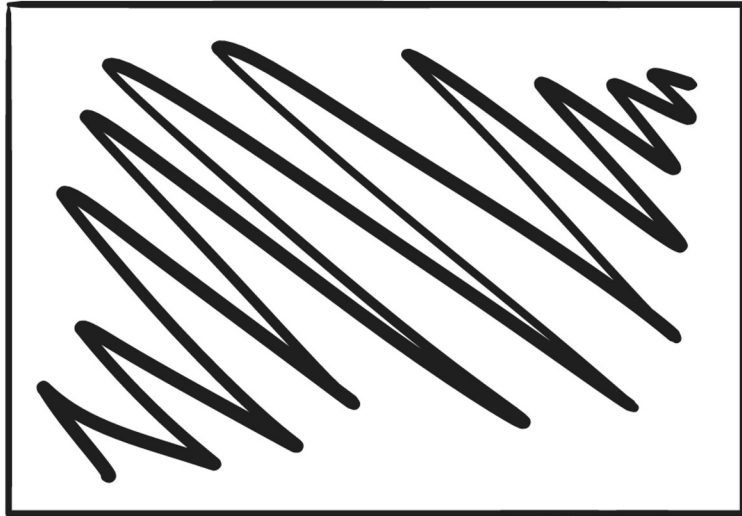




# Tree of Taste Business Expectations

I want it all

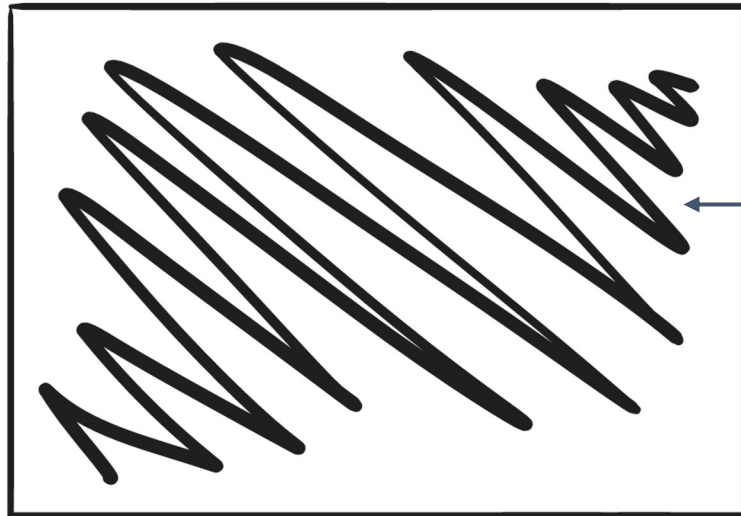
Order Management System



# Modernizing a legacy application

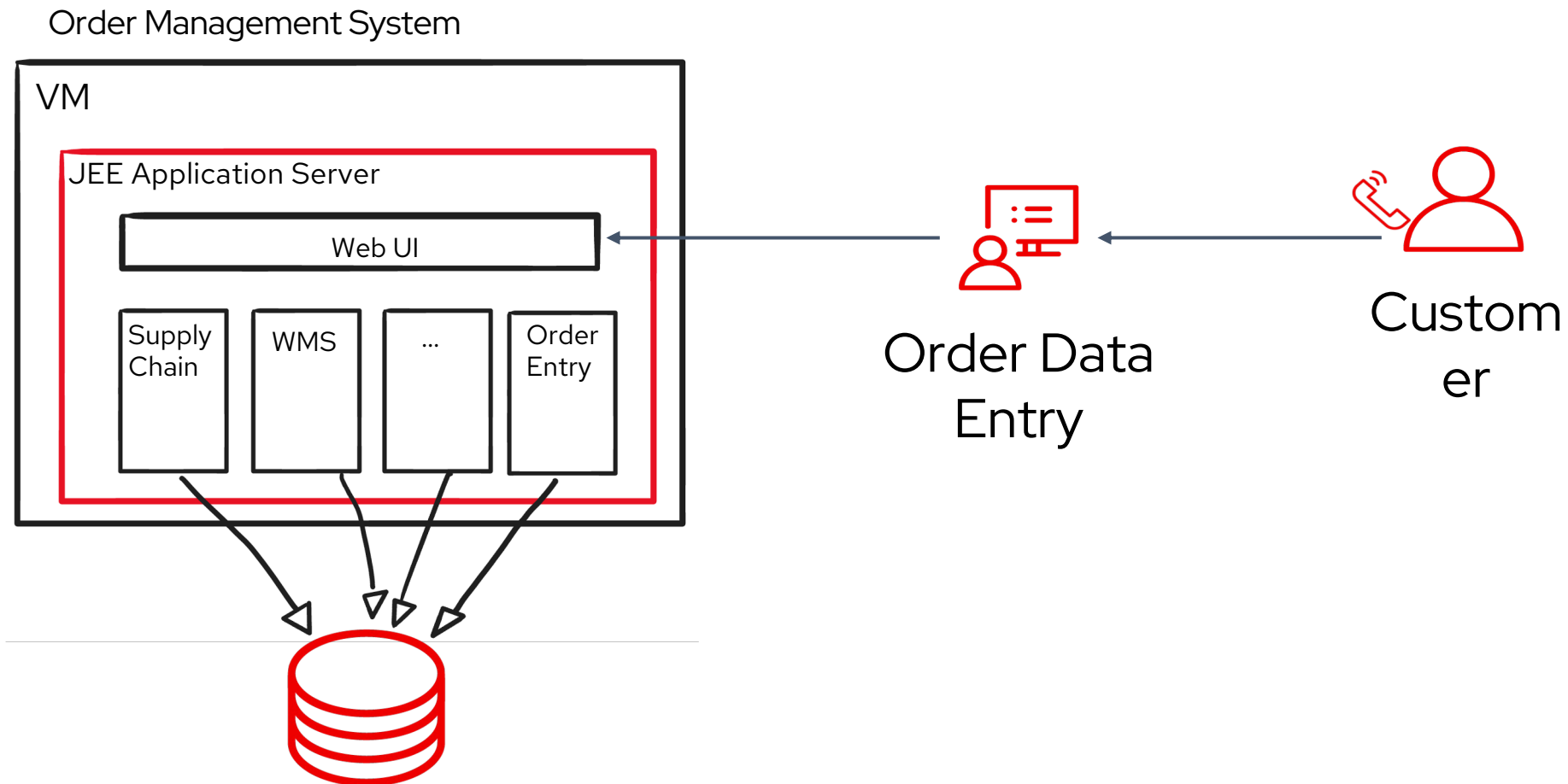
What's under the hood ?

Order Management System



# Modernizing a legacy application

## Internals

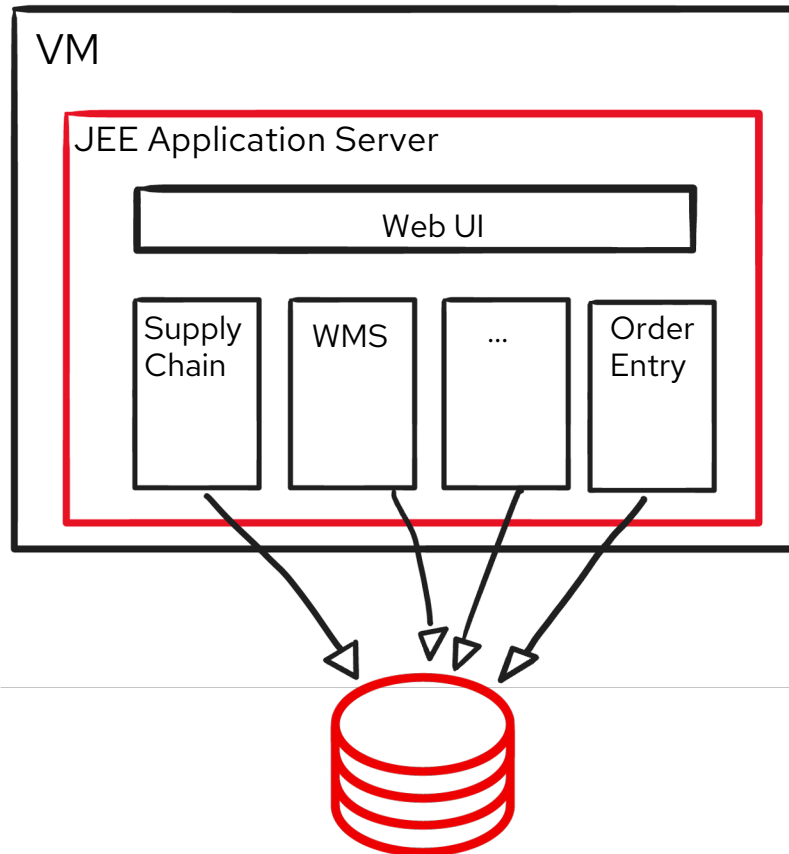




# Order Management System

What's preventing the innovation ?

Order Management System



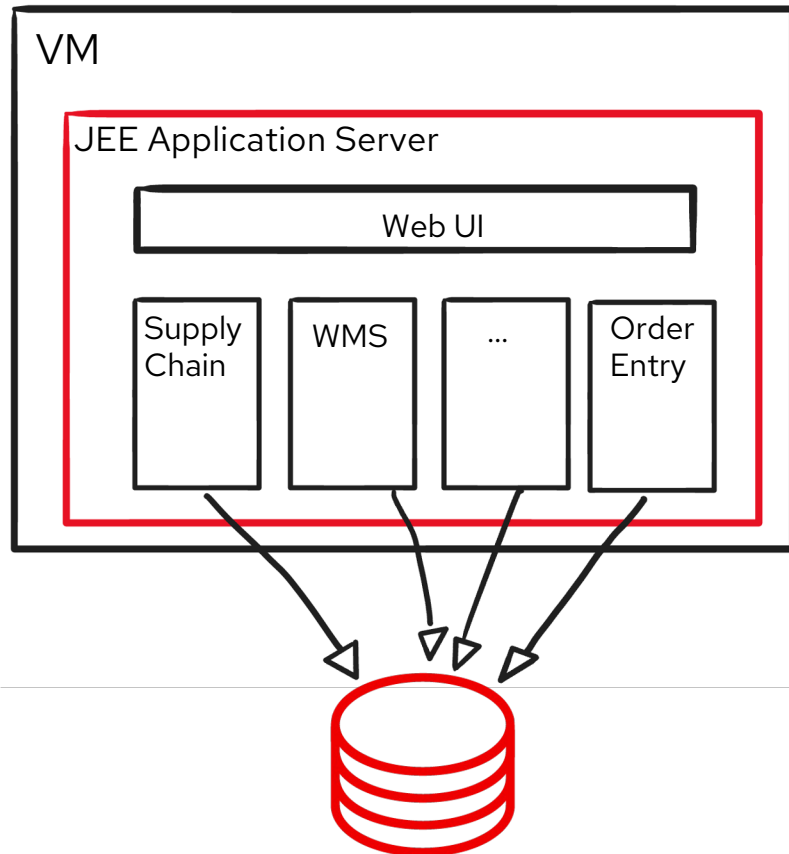
Virtual Machines

Monolithic  
legacy  
applications  
and Monolithic  
RDBMS

# Order Management System

What's preventing the innovation ?

Order Management System



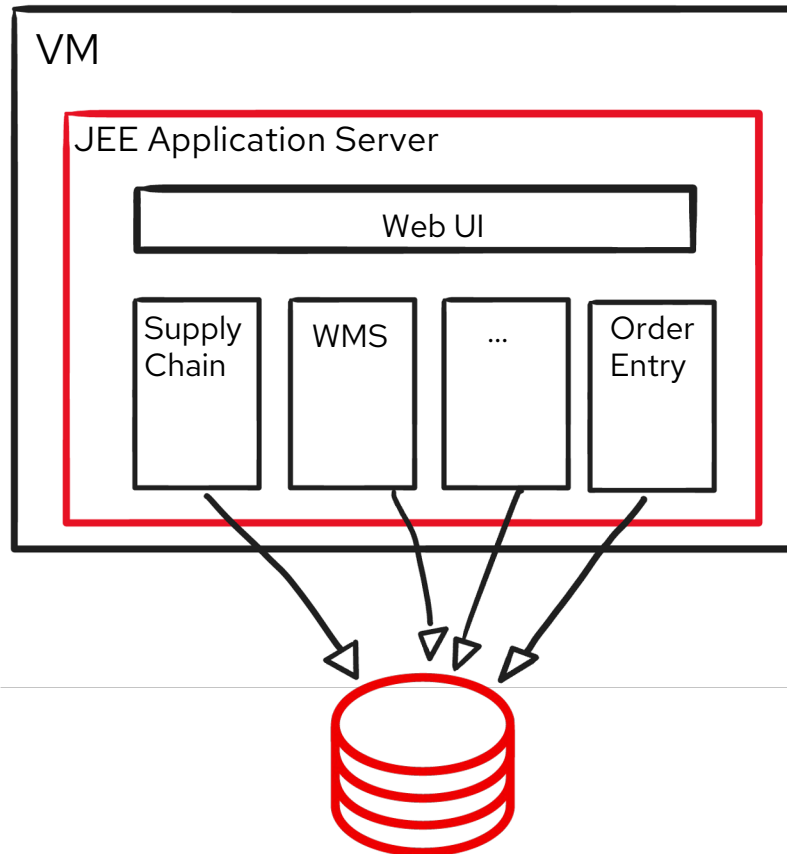
- High costs
- Complex maintenance
- Can't scale well

Monolithic legacy applications and Monolithic RDBMS

# Order Management System

What's preventing the innovation ?

Order Management System



- High costs
- Complex maintenance
- Can't scale well

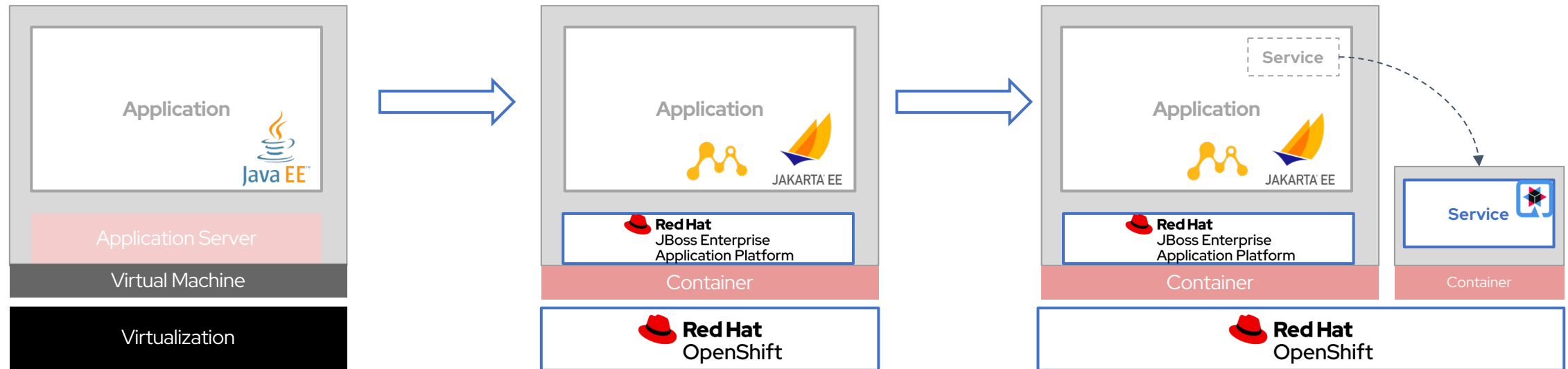
- Tech debt
- Slow change process
- Coupling in SDLC

So, how do we modernize ?



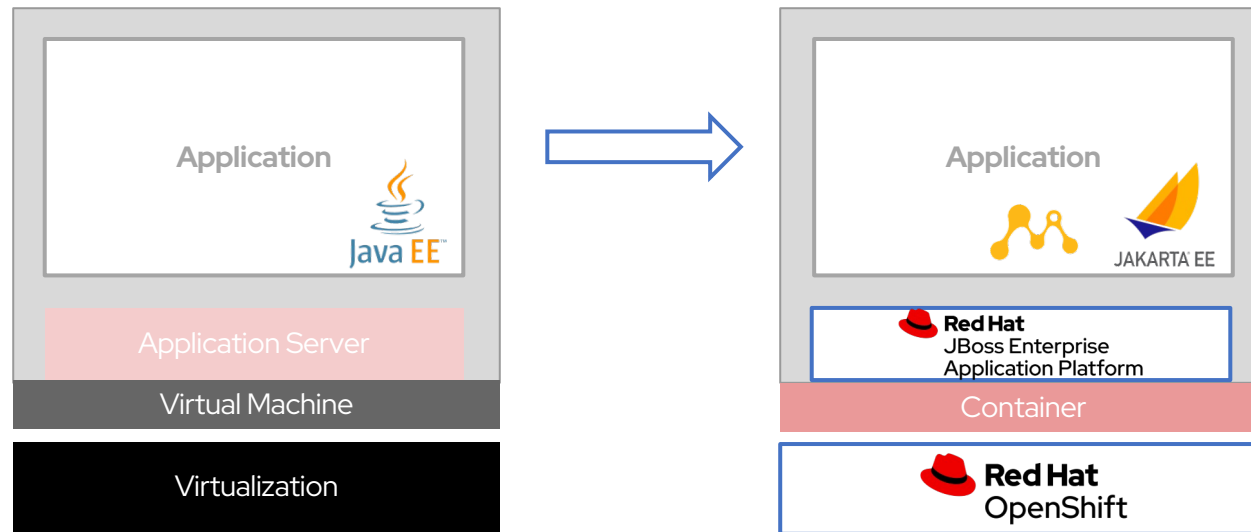
# Replatform and Refactor

Leverage an incremental approach



# Replatforming on OpenShift

## First step



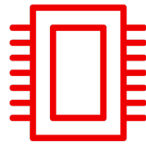
# Replatforming on OpenShift

## Benefits of deploying JBoss EAP on OpenShift



### Lower Operational Costs

- ▶ Streamlined configuration deployment and management



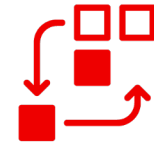
### Reduced Resource Usage

- ▶ Increased app density
- ▶ Decreased memory footprint



### Improved developer experience

- ▶ Developer-centric user interface and tools
- ▶ Closer to cloud-native tooling and development



### Access platform Additional features

- ▶ Advanced deploy options
- ▶ Clustering support
- ▶ Platform Autoscaling
- ▶ Graceful shutdowns

Modernizziamo le applicazioni

# Hands-on Application analysis with Tackle

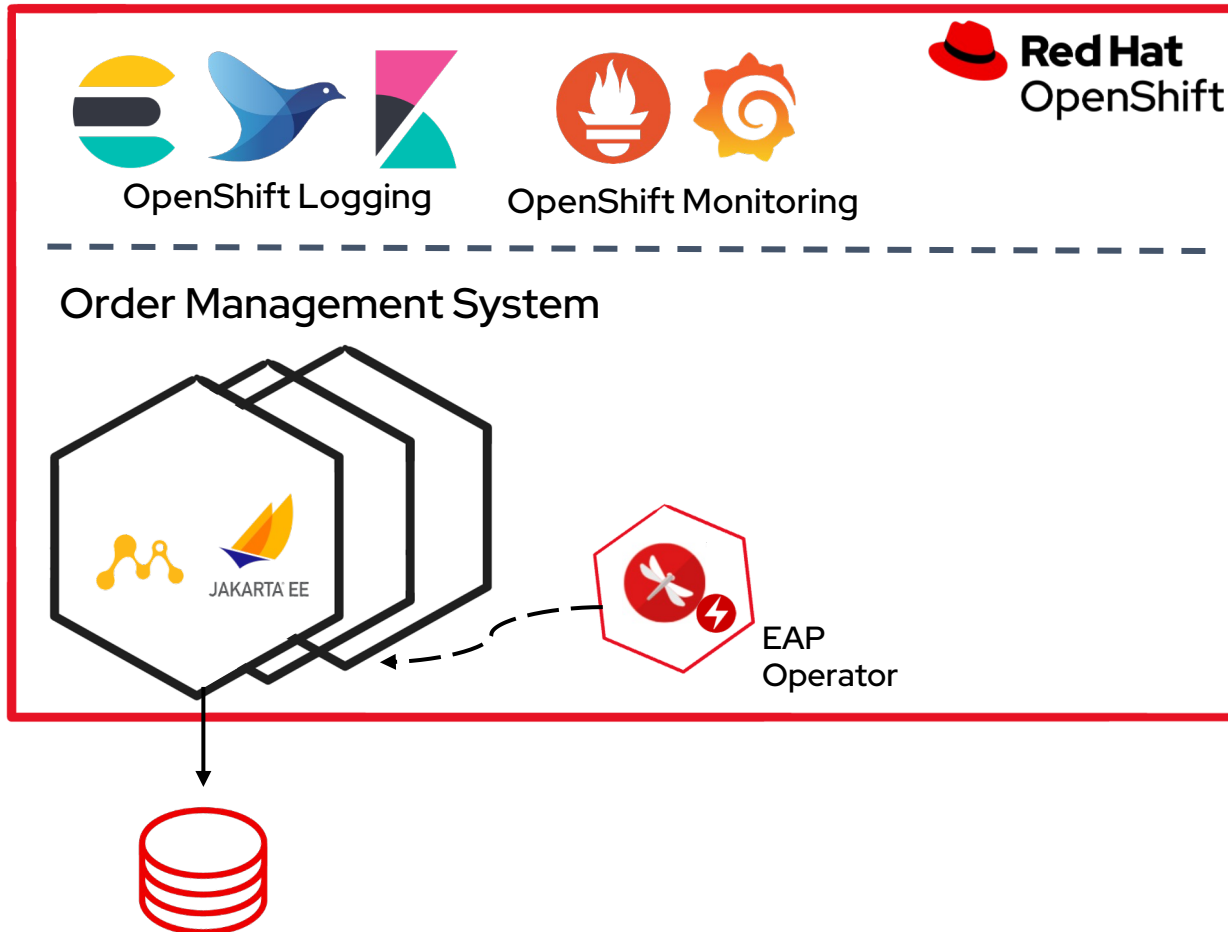


Modernizziamo le applicazioni

# Hands-on Replatforming JBoss EAP

# Replatforming on OpenShift

## Result



### Quick wins

#### Workload management

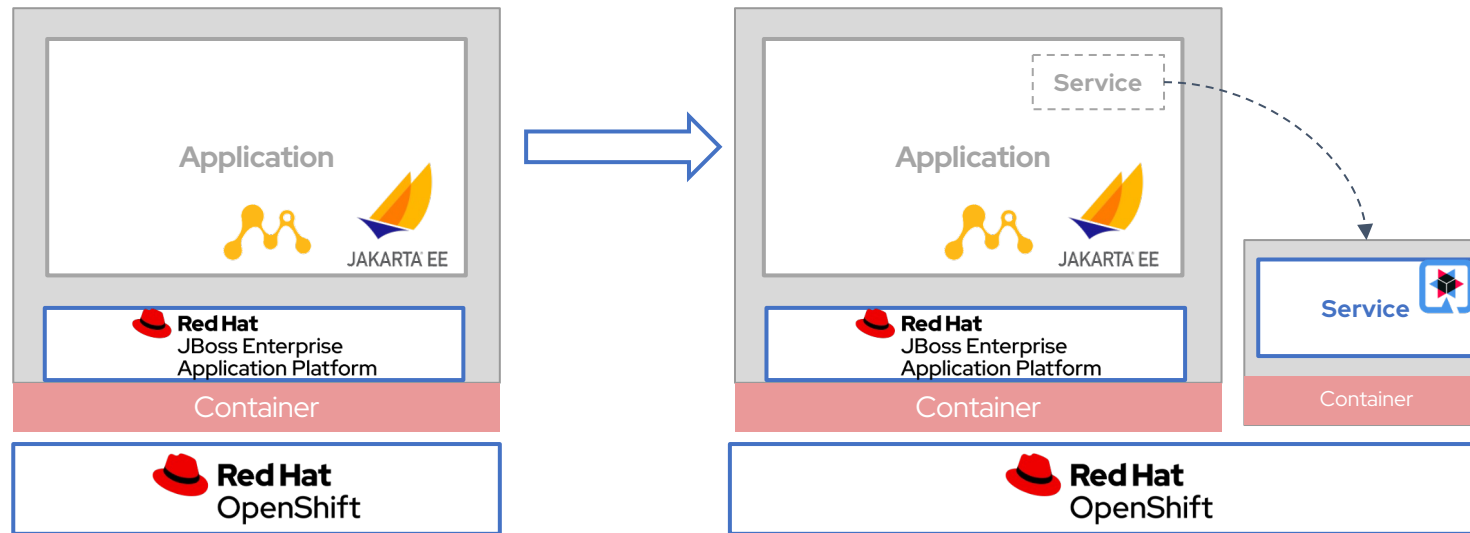
- ▶ Operator Support
- ▶ Platform Logging
- ▶ Platform Monitoring
- ▶ Autoscaling

#### Developer productivity

- ▶ Developer perspective
- ▶ Operator Hub & Marketplace
- ▶ Self-service access to platform services

# Refactoring with Quarkus

## Second step



# Refactoring with Quarkus

Supersonic. Subatomic. Java



## Container First

- ▶ Tailors your app for HotSpot & GraalVM
- ▶ Fast boot time and low RSS memory
- ▶ Serverless fit



## Unifies Imperative and Reactive

- ▶ Combines blocking and non-blocking
- ▶ Built-in event bus



## Best of breed

- ▶ 400+ extensions
- ▶ Microprofile based



## Developer Joy

- ▶ Live coding
- ▶ Continuous testing
- ▶ Dev UI
- ▶ DevServices

Modernizziamo le applicazioni

# Hands-on Refactoring with Quarkus

# Refactoring with Quarkus

## Result

