



6 PATHS TO MODERN INFRASTRUCTURE

Microservices on Red Hat OpenShift Container Platform

Andrzej Kowalczyk
Senior Solution Architect
andrzej.kowalczyk@redhat.com

Agenda

- Introduction
- Red Hat microservices
- JBoss for microservices
- Red Hat Agile Integration

THE BUILDING BLOCKS

OPEN HYBRID CLOUD



THE BUILDING BLOCKS

CONTAINERS



OPEN HYBRID
CLOUD



THE BUILDING BLOCKS

MICROSERVICES



CONTAINERS

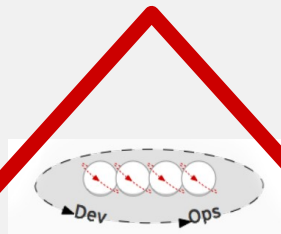


OPEN HYBRID
CLOUD



THE BUILDING BLOCKS

DevOps



MICROSERVICES



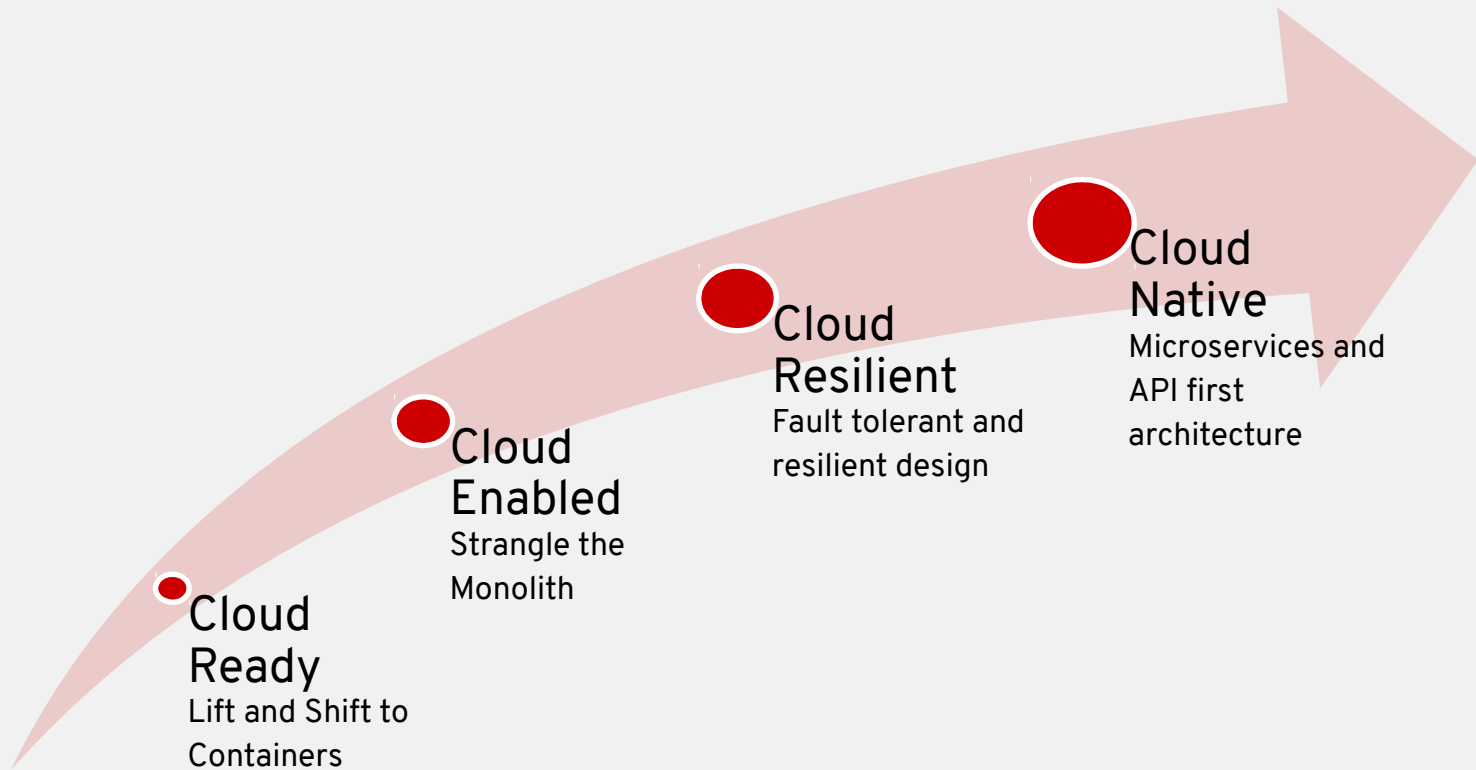
CONTAINERS



OPEN HYBRID
CLOUD



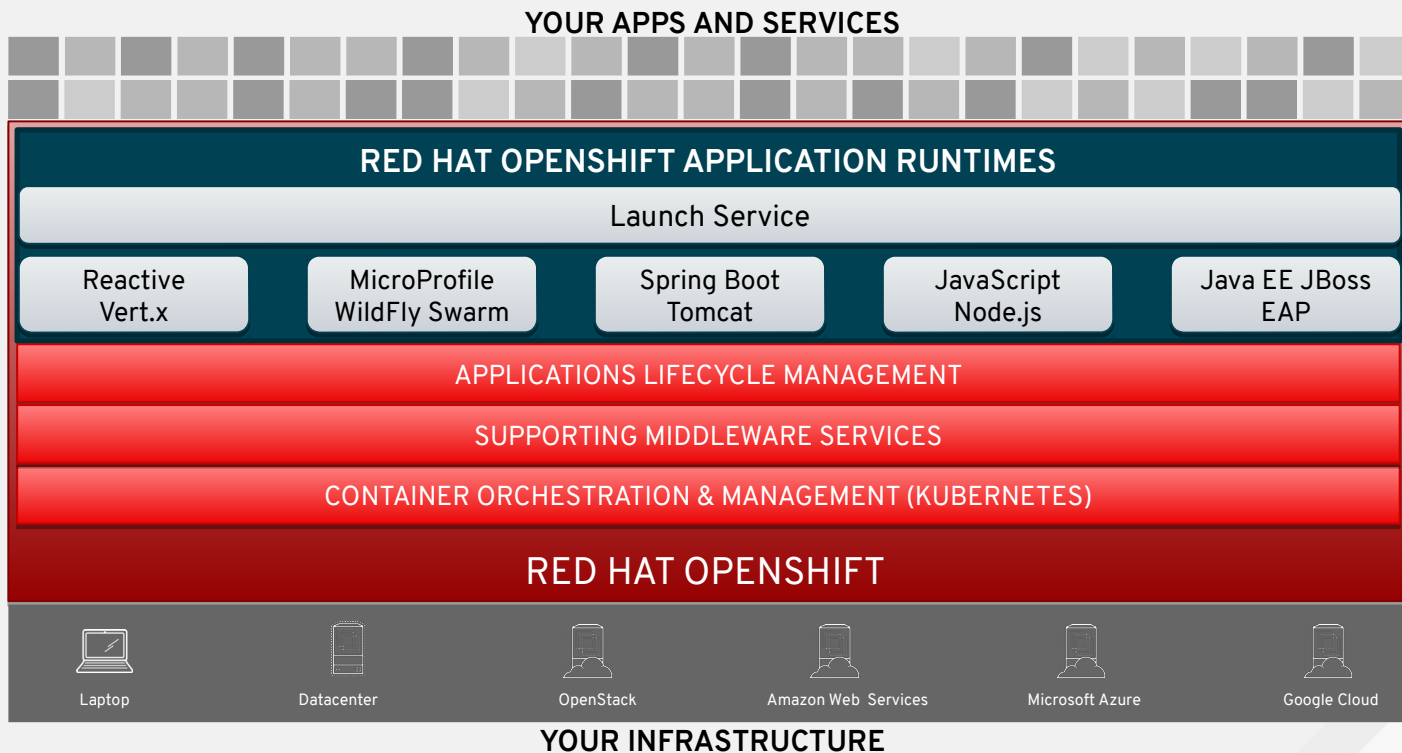
CLOUD NATIVE RAMP UP MODEL



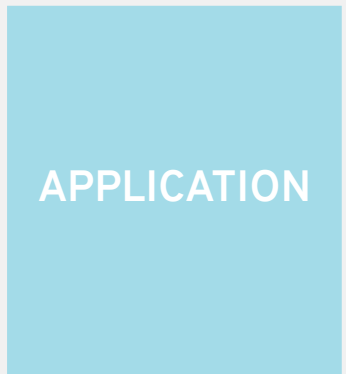
RED HAT OPENSIFT APPLICATION RUNTIMES

Providing curated set of integrated runtimes and frameworks *that standardizes Cloud Native App Dev*

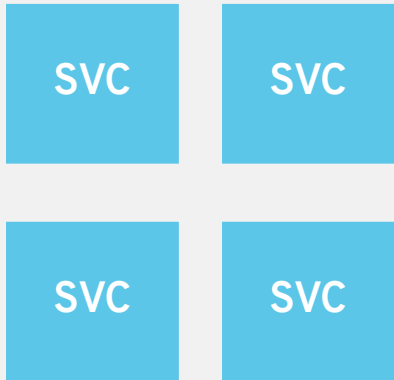
- ✓ Simplified development
- ✓ Strategic flexibility
- ✓ DevOps automation



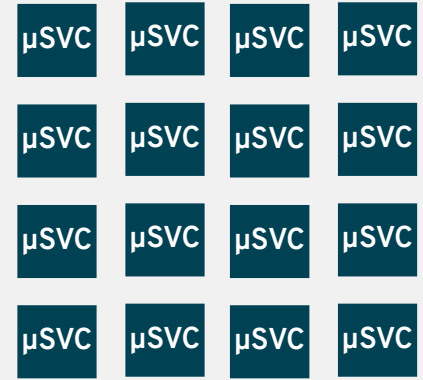
MICROSERVICES: MORE FLEXIBLE ARCHITECTURE



TRADITIONAL



SOA



MICROSERVICES

SIDE NOTE: MICROSERVICES ≠ ONE SIZE FITS ALL



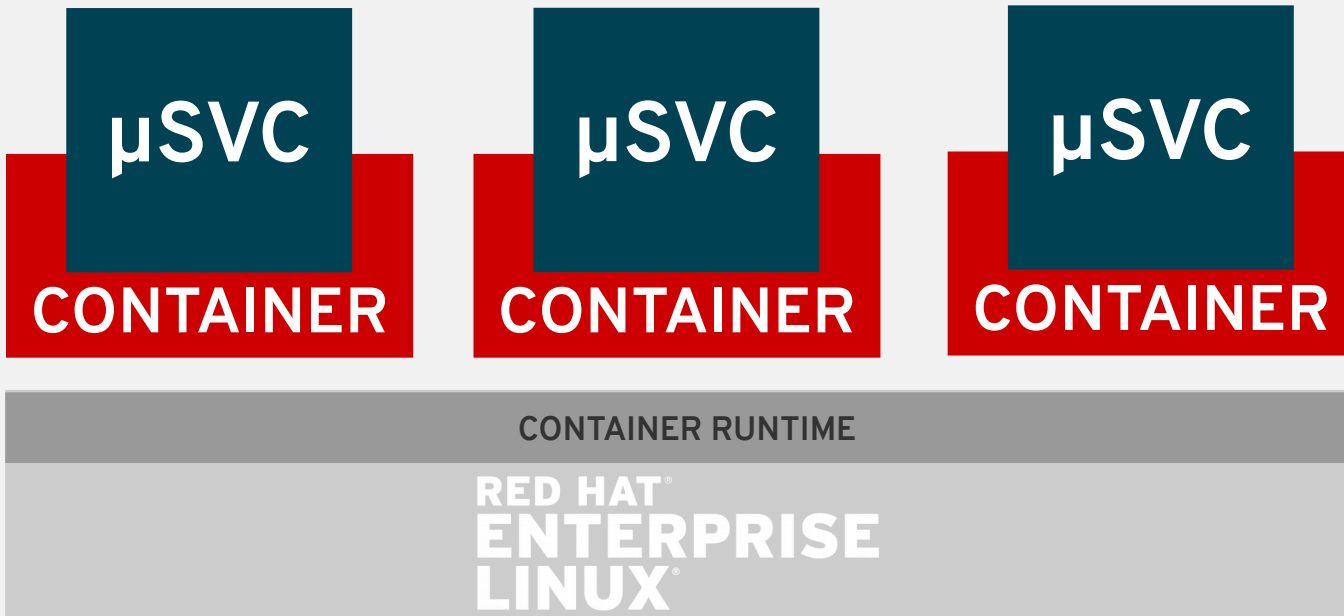
MICROSERVICE + CONTAINER



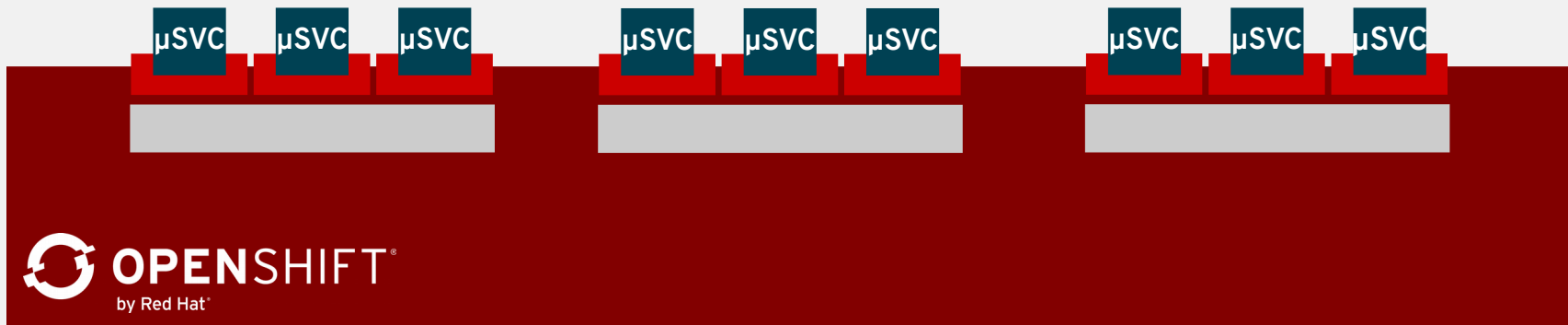
DEPLOYMENT MODEL IS EXPLODING AS STANDARD



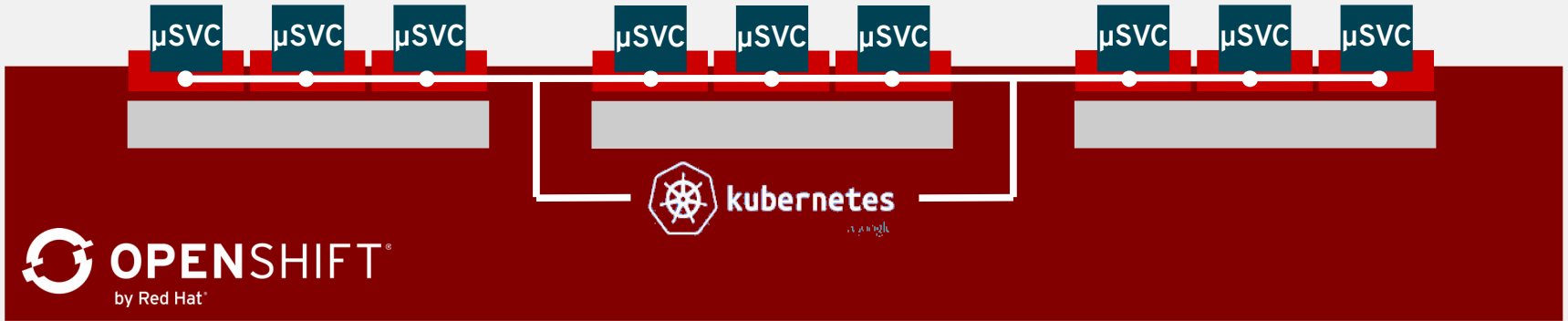
RHEL: THE WAY TO RUN CONTAINERS



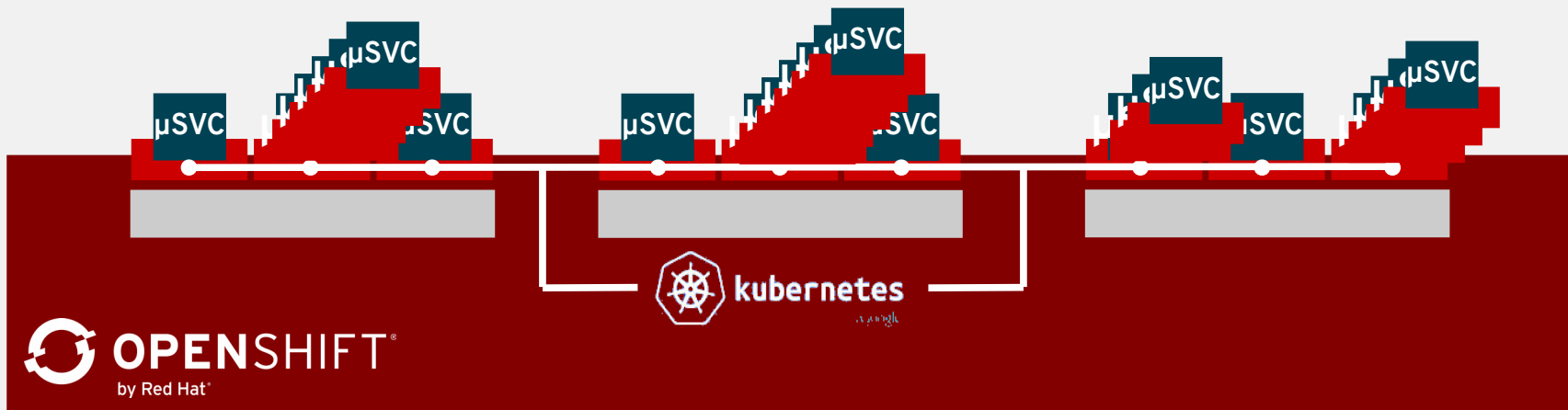
OPENSIFT TO MANAGE MANY MICROSERVICES



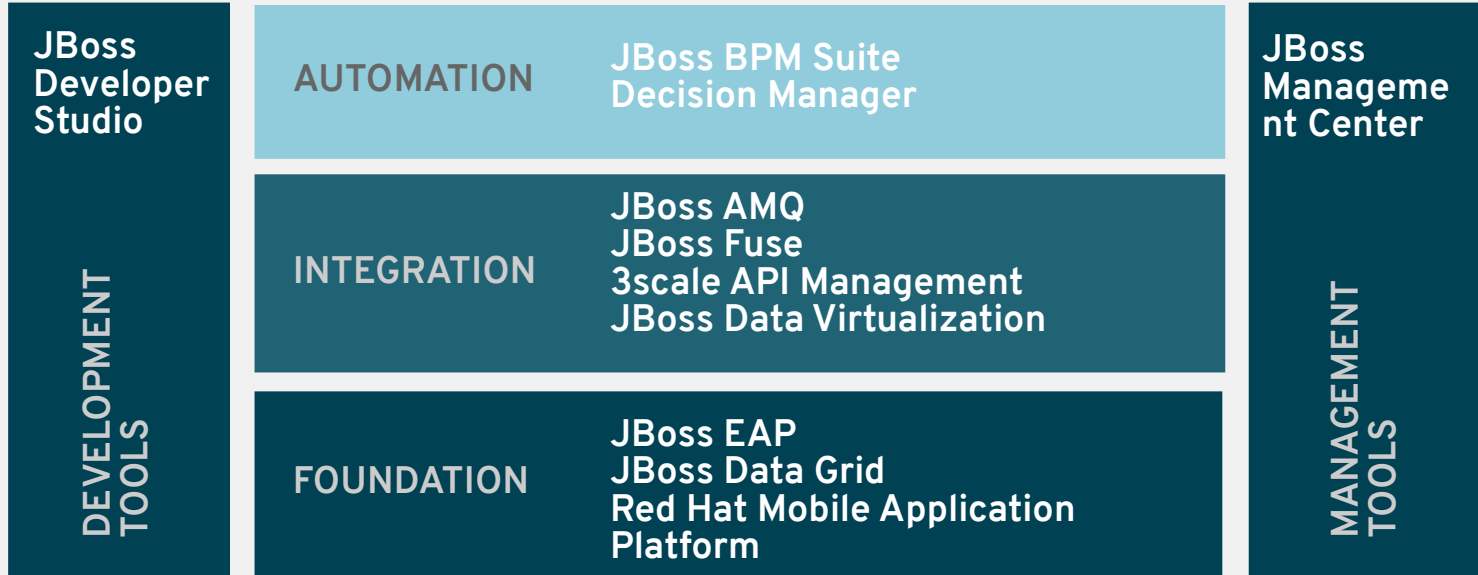
KUBERNETES FOR ORCHESTRATION



KUBERNETES FOR ORCHESTRATION



MIDDLEWARE FOR OPEN HYBRID CLOUD



PHYSICAL



VIRTUAL



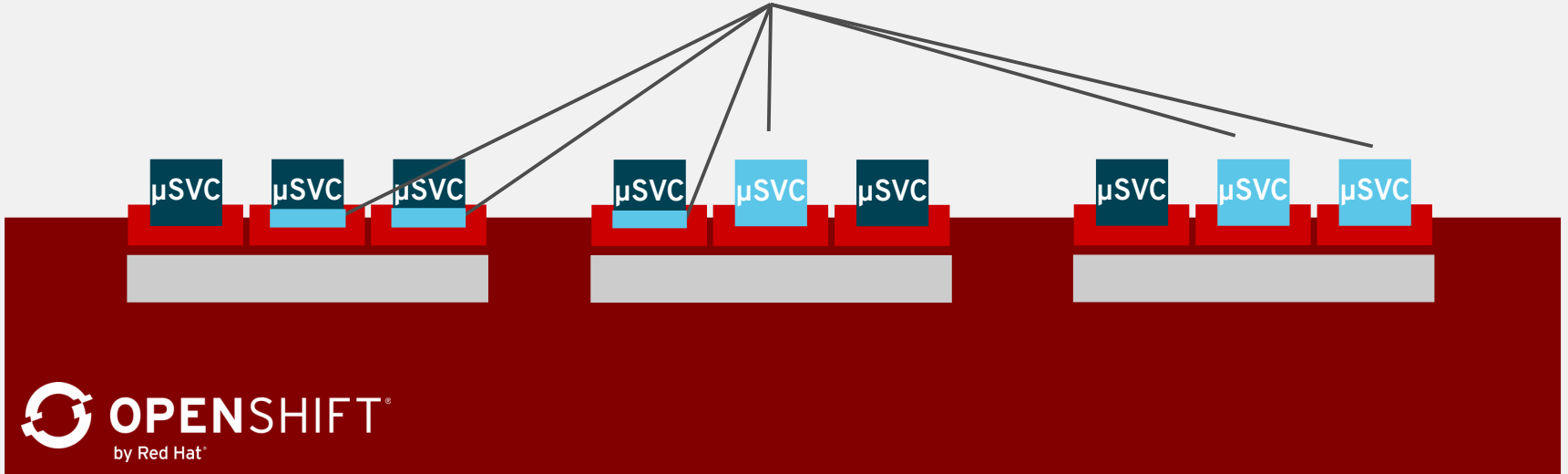
PRIVATE





PUBLIC

JBoss MIDDLEWARE CONTAINERIZED ON OPENS SHIFT

**RED HAT® JBOSS®
MIDDLEWARE**

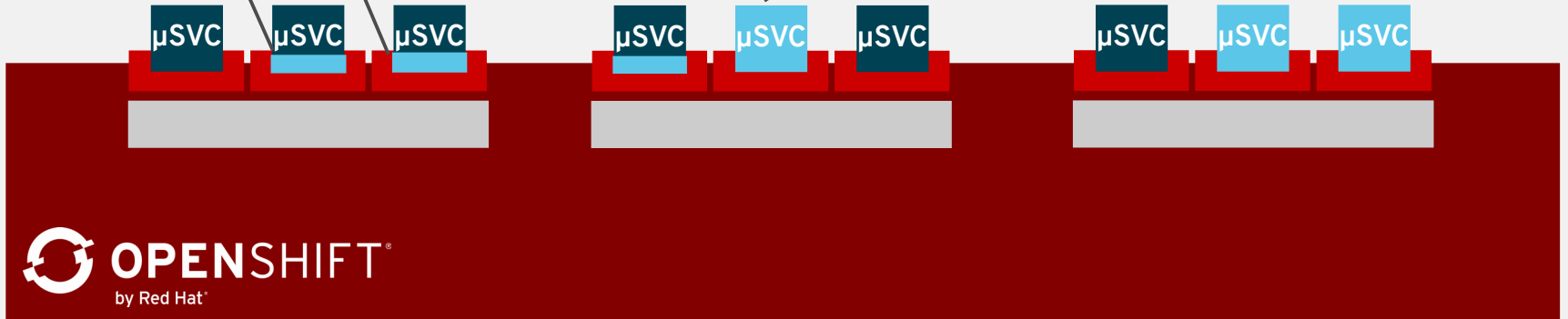


-  CUSTOM BUSINESS LOGIC
-  JBoss MIDDLEWARE

aPaaS

**RED HAT® JBOSS®
ENTERPRISE
APPLICATION PLATFORM**

**RED HAT® JBOSS®
DATA GRID**



■ CUSTOM BUSINESS LOGIC

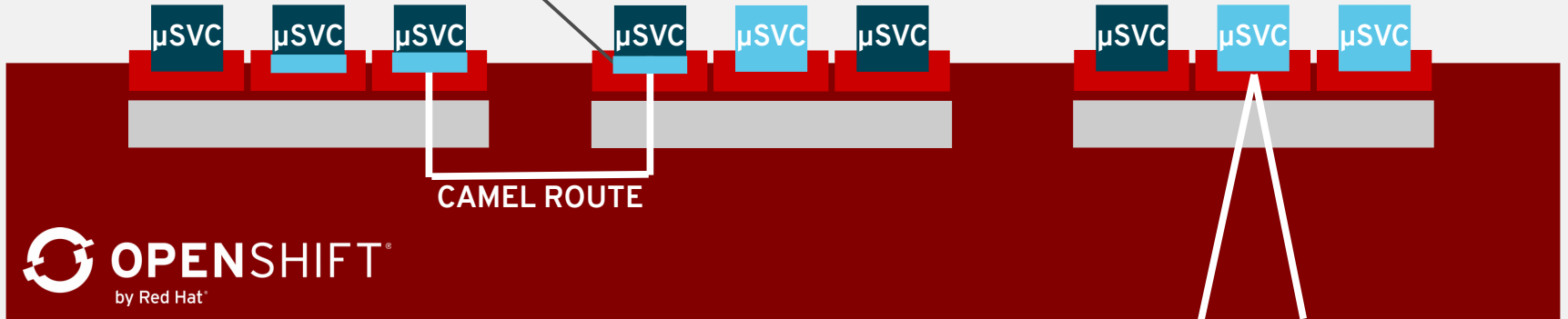
■ JBoss MIDDLEWARE

iPaaS

RED HAT® JBOSS®
FUSE

RED HAT® JBOSS®
A-MQ

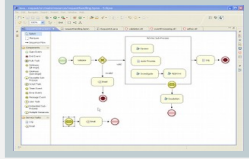
RED HAT® JBOSS®
DATA VIRTUALIZATION



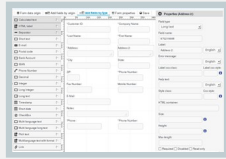
-  CUSTOM BUSINESS LOGIC
-  JBoss MIDDLEWARE

RED HAT® JBOSS® BPM SUITE

bpmPaaS



MODELING



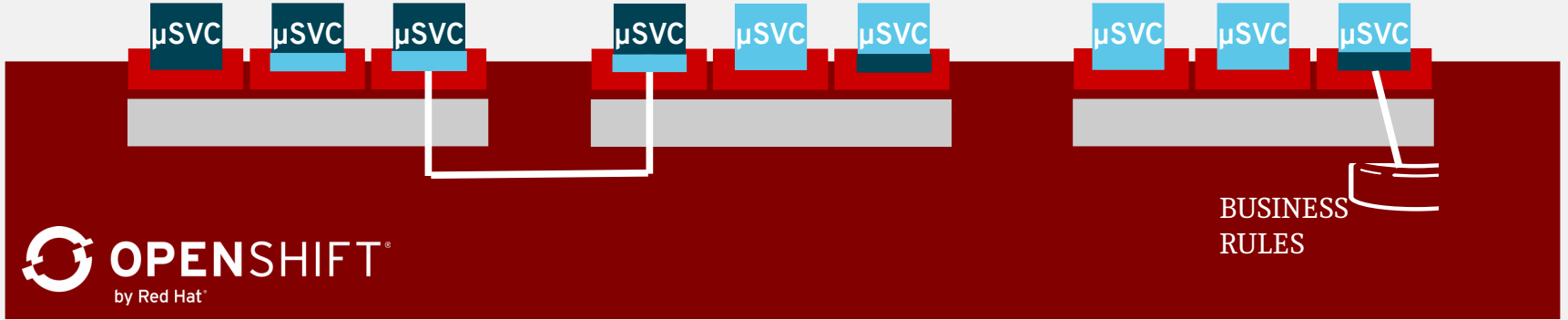
DEPLOYMENT




MANAGEMENT



ANALYTICS



 CUSTOM BUSINESS LOGIC

 JBoss MIDDLEWARE

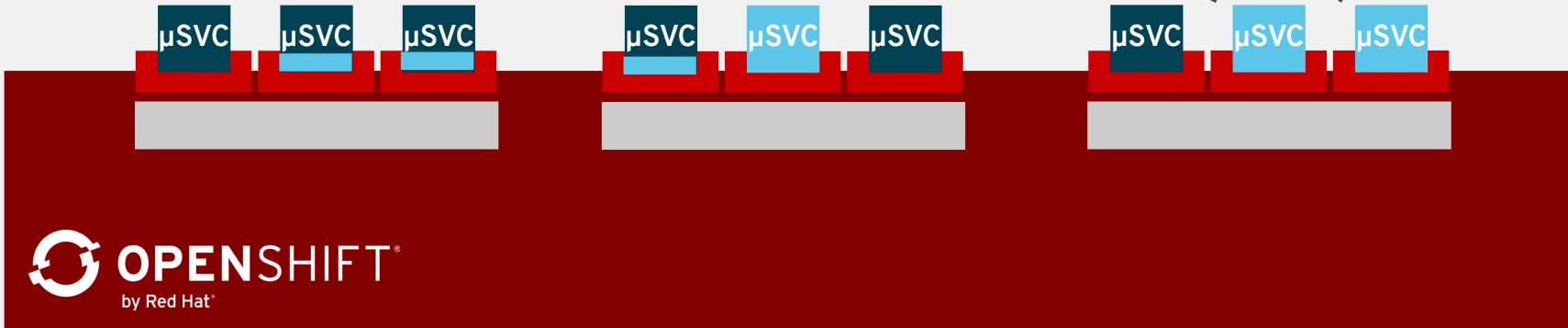
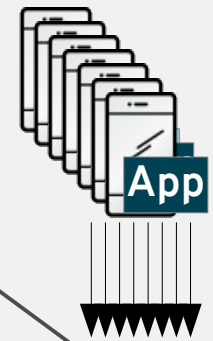
RED HAT® JBOSS®
BRMS

mPaaS

RED HAT® MOBILE APPLICATION PLATFORM

Device
SDKs
Form
Builder
Build Farm

Push
Sync
Security
App Mgt
API Mgt



-  CUSTOM BUSINESS LOGIC
-  JBoss MIDDLEWARE

Agile Integration

A person in a blue martial arts uniform is captured in a dynamic pose, performing a high kick against a stone wall. The person is wearing a blue gi with white wristbands and a white patch on the left wrist. The background is a clear blue sky with a bright sun flare on the right side. The word "Agility" is overlaid in large white text.

Agility

a·gil·i·ty*

/ə'jɪlədē/

noun

“ability to move quickly and easily.”

“The most critical business capability today”

Src: https://www.bcgperspectives.com/content/articles/it_strategy_it_transformation_it_enabled_business_agility/



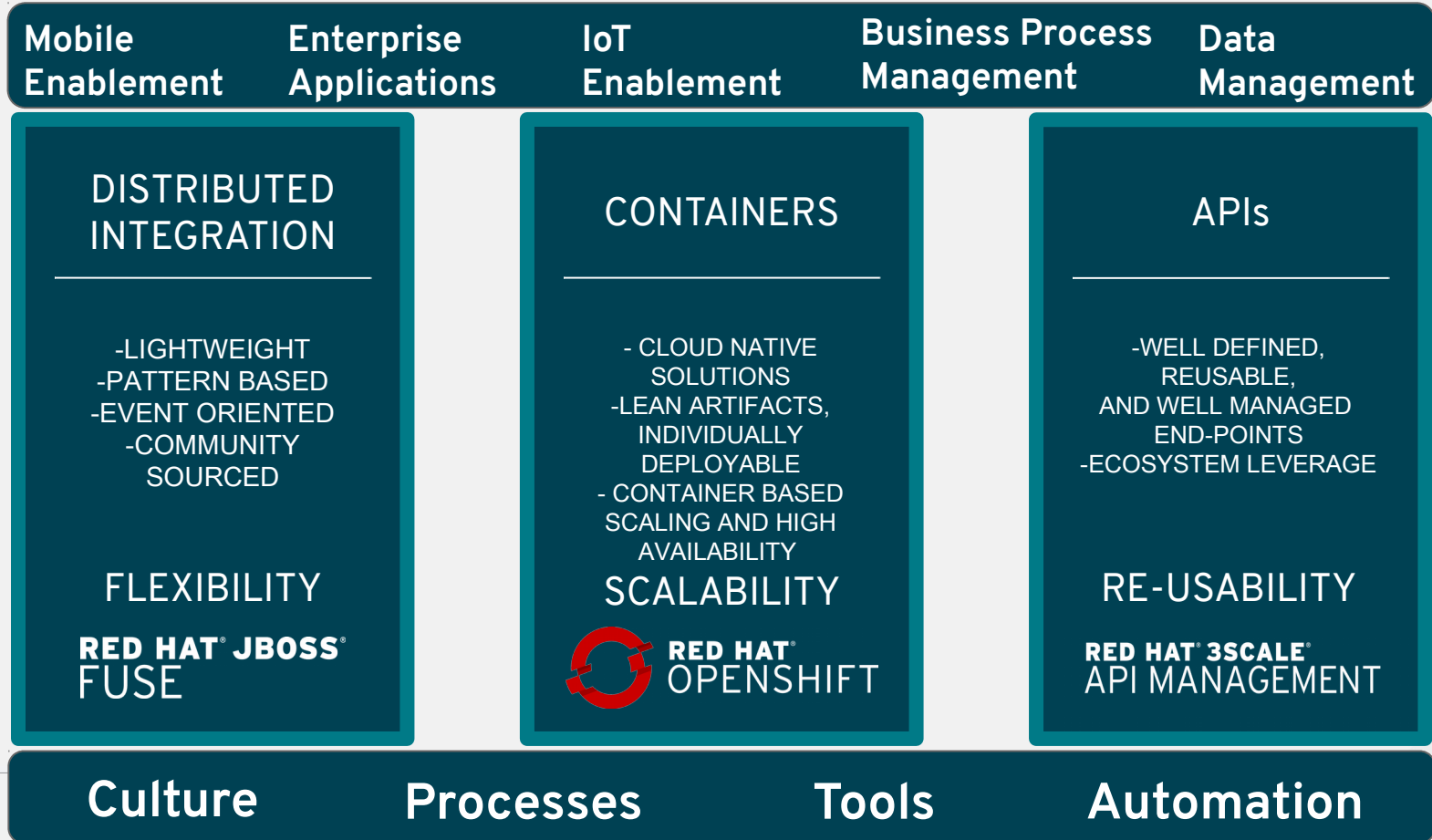
How Red Hat technology matches the vision

Typical Integration Challenges

Increase Internal Agility	Build consistent Omni-channel Experience	Leverage Existing Data/Services Better
Build Customer Or Partner Ecosystems	Integrate With Cloud Native/Mobile/IoT Apps	Enable SaaS Integrations Across Hybrid Environments

For details refer to the Agile Integration Whitepaper here: << [LINK](#) >>

Red Hat's Agile Integration Concept

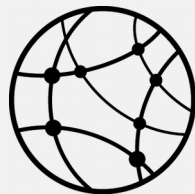


Pillar 1: Two Dimensions of Distributed Integration

DISTRIBUTED
INTEGRATION

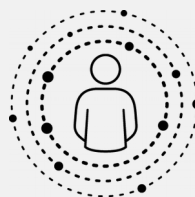
RED HAT® JBOSS®
FUZE

FLEXIBILITY



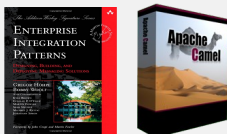
WHERE INTEGRATION IS USED

Data & Service integration at the center or the edge of your enterprise architecture.

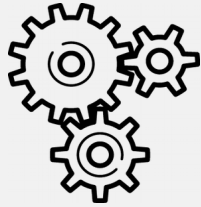


WHO PERFORMS INTEGRATION

Unlocking integration capability for new categories of integration personas.



Pillar 1: Distributed Integration (Fuse Variants)



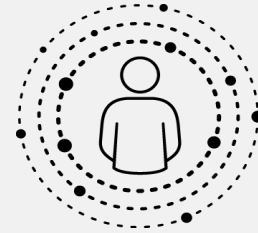
Fuse
Standalone

- Single JVM Fuse, EAP based
- Integration Specialist
- Integration where you need it
- **“Classic” integration**



Fuse
Integration Services

- Scale out Fuse
- Integration Specialist
- Optimised for OpenShift
- **“Cloud native” integration**



Fuse
Online (iPaaS)

- Low/no-code UX
- Citizen Integrator
- 100% cloud-based via OpenShift
- Integration through a browser
- **“Hybrid” integration**

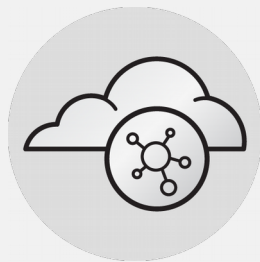
Pillar 2: Containers (OpenShift)

Single platform and toolchain across cloud environments provides consistency and flexibility for current and future deployment plans.



PRIVATE CLOUD

Deploy on-premise



PUBLIC CLOUD

*Deploy on public cloud
provider*



MANAGED CLOUD

*Deployed and managed
by Red Hat*



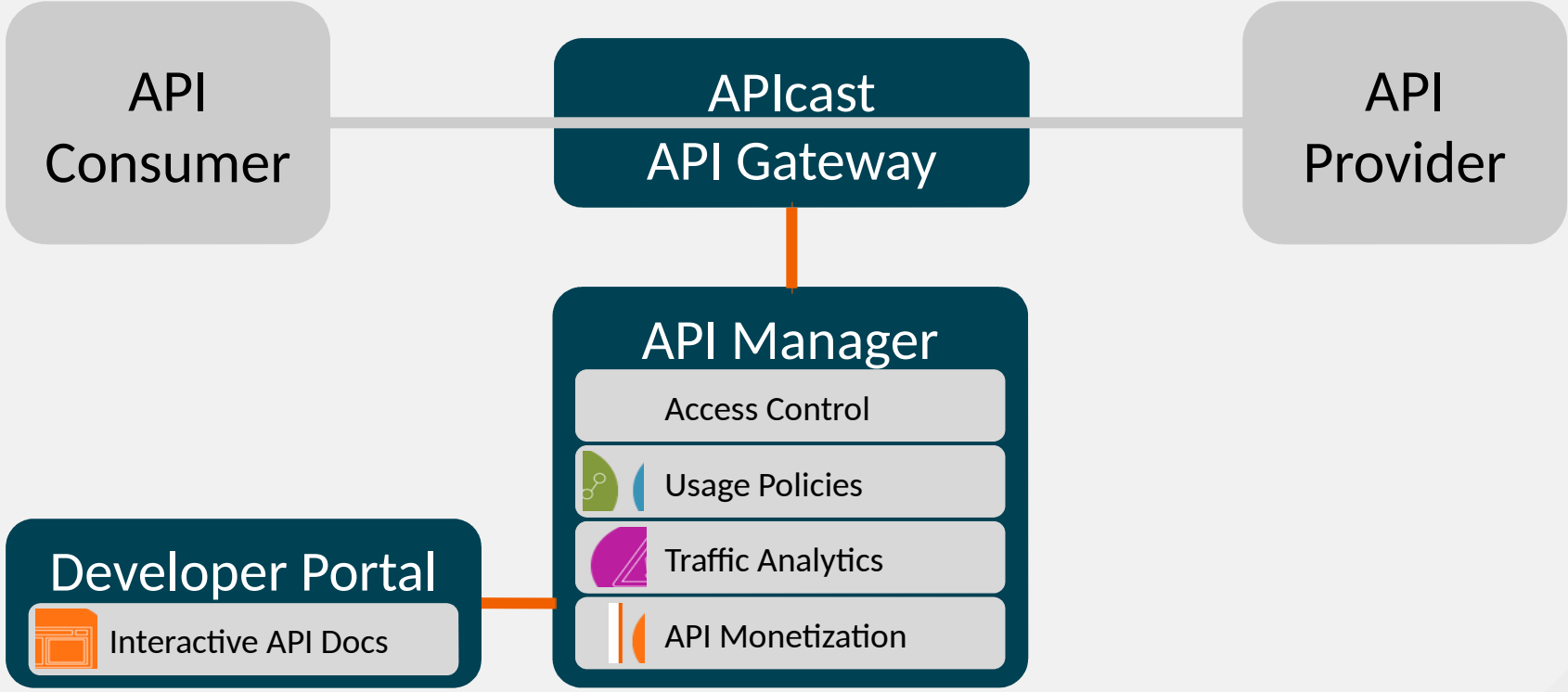
Pillar 2: Containers (OpenShift)

CONTAINERS



- Cloud Native
 - Multi-tenancy
 - Security
 - Networking
 - Containers
 - Self healing
- Application centric
 - Cloud deployment support
 - Service lookup

Pillar 3: APIs / API Management (3scale)





Interactive session

Red Ridding Hood wants to open Start-up

Food like Grandma Used to Make



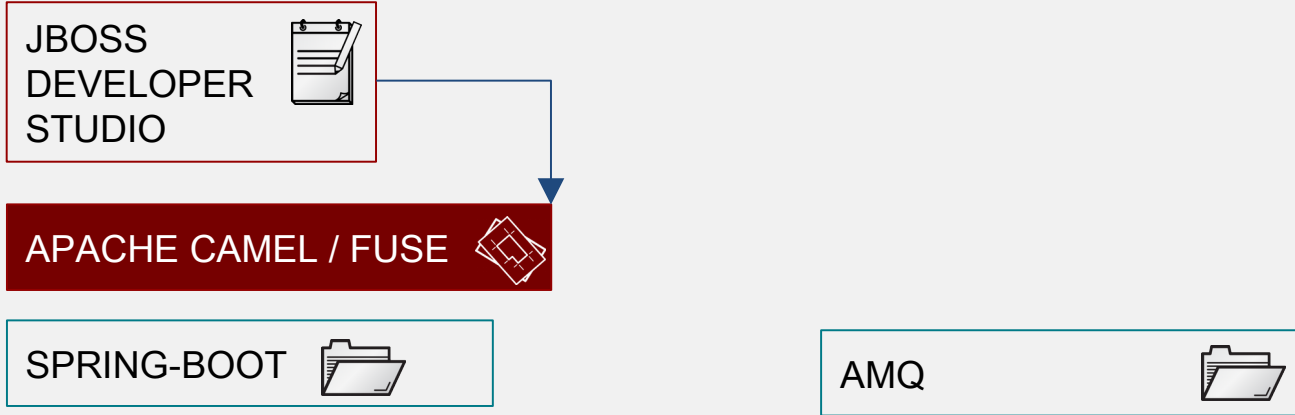
I want to offer home made food via many different web shops.

I want to expose my microservices via API like list of customers, offer, etc.

I want to have async and XML based communication with my logging system via fully HA messaging system.

I don't afraid of big wolfs – I will use Open Source solutions.

Red Ridding Hood wants to open Start-up *Food like Grandma Used to Make*



Red Hat JBoss Developer Studio – developer tool

Red Hat JBoss Fuse - agile integration tool on Spring-boot

Red Hat JBoss AMQ – messaging system

Red Ridding Hood wants to open Start-up

Food like Grandma Used to Make



Why Red Hat JBoss Fuse – based on Apache Camel, many connectors, mature product


Why SpringBoot – simple, good performance, easiness of usage: Java, Drag&Drop with GUI

Why AMQ – supports JMS, MQTT, integrated with Fuse, many HA architectures easy to use


Why OpenSource – I can start right now, don't need ask

AGILE INTEGRATION

Distributed Integration


FUSE 


APACHE KARAF


SPRING BOOT 


Flexibility

Container

OPENSIFT 


KUBERNETES 


DOCKER 

FABRIC8 

Scalability

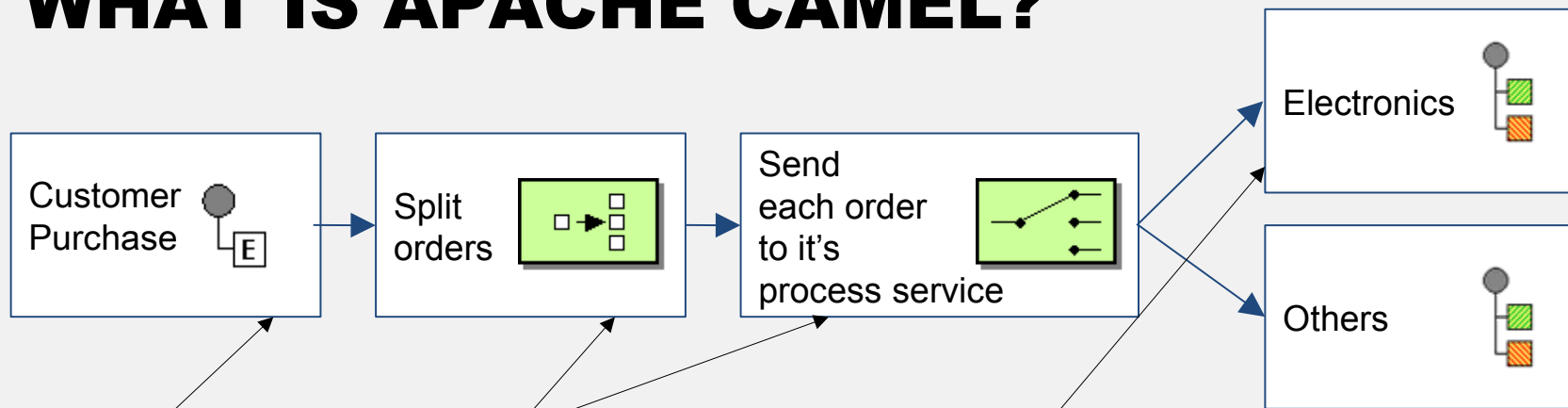
API

FUSE 

3SCALE 

Re-Usability

WHAT IS APACHE CAMEL?



```
from("file:work/cbr/input")
  .split(xpath("//orders"))
  .choice()
    .when(xpath("/order:order/order:type = 'E'"))
      .to("activemq:queue:electronic/us")
    .otherwise()
      .recipientList(simple("http4://otherservice"));
```

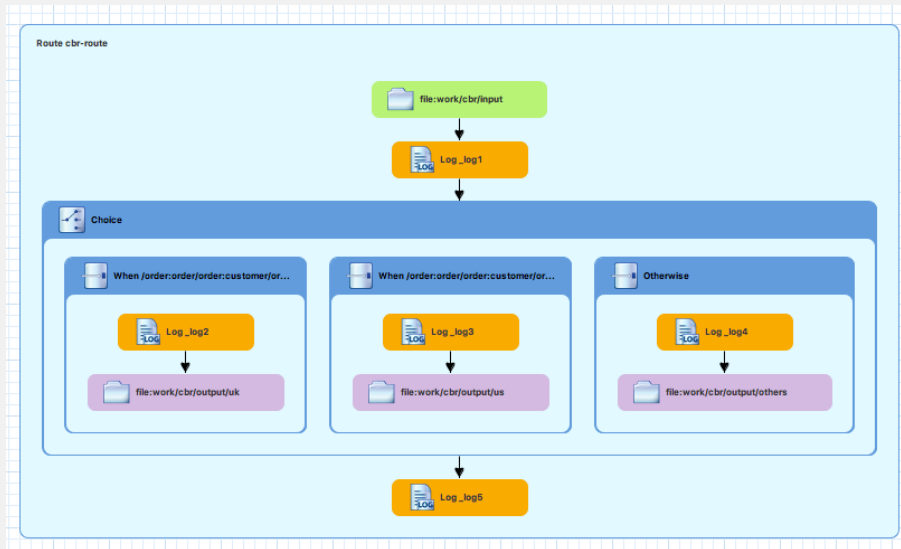
CAMEL DSL

JBOSS
DEVELOPER
STUDIO

Java DSL

Blueprint DSL (XML)

Spring DSL (XML)



```
from("file:work/cbr/input")
  .log("Receiving order ${file:name}")
  .choice()
    .when(ns.xpath("//c:order/c:customer/c:country[text() = 'UK']"))
      .log("Sending order ${file:name} to the UK")
      .to("file:work/cbr/output/uk")
    .when(ns.xpath("//c:order/c:customer/c:country[text() = 'US']"))
      .log("Sending order ${file:name} to the US")
      .to("file:work/cbr/output/us")
    .otherwise()
      .log("Sending order ${file:name} to another country")
      .to("file:work/cbr/output/others")
  .log("Done processing ${file:name}");
```

Red Ridding Hood wants to open Start-up

Food like Grandma Used to Make



What if my *Food like Grandma Used to Make* will be great success?

How can I serve huge load? I need to have auto-scaling functionality!

Hmm... I should be ready for a cloud solution.

But how to deliver new functionality without disturbing other

I will use Linux containers and Blue/Green deployment

Red Ridding Hood wants to open Start-up

Food like Grandma Used to Make

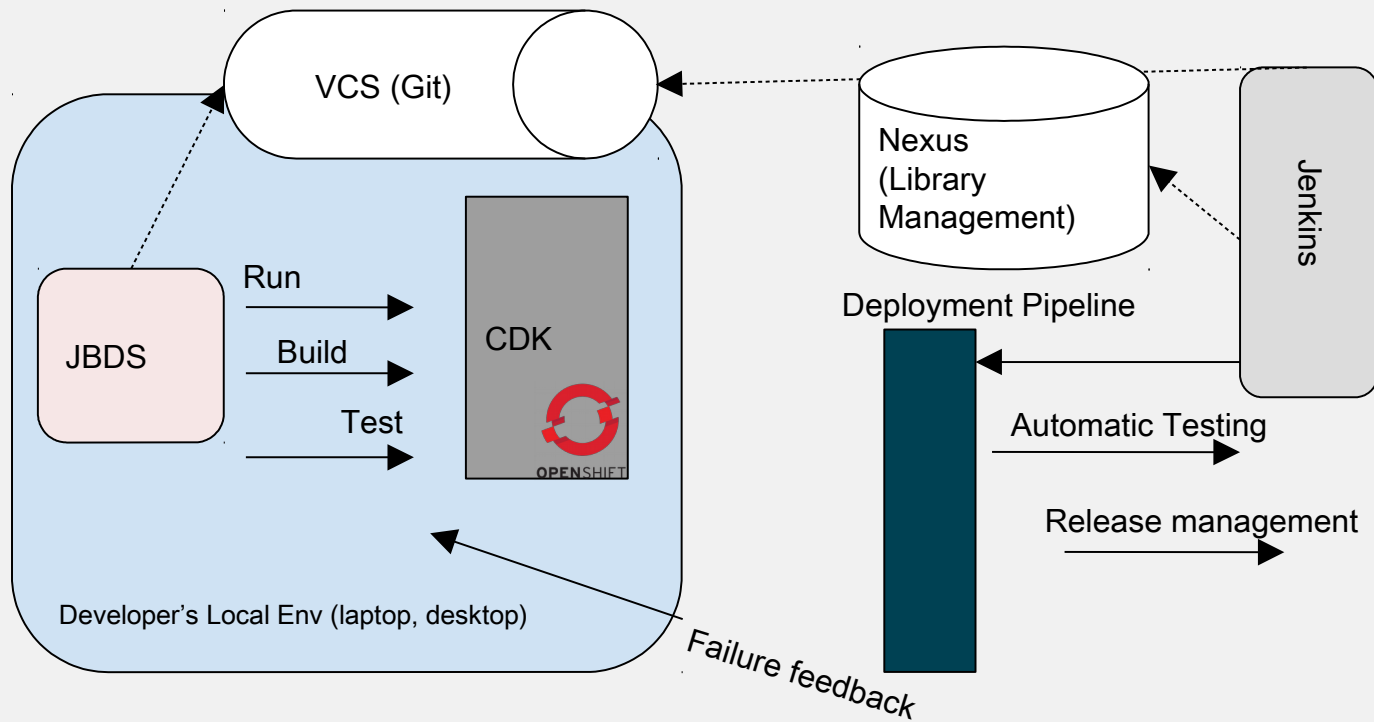


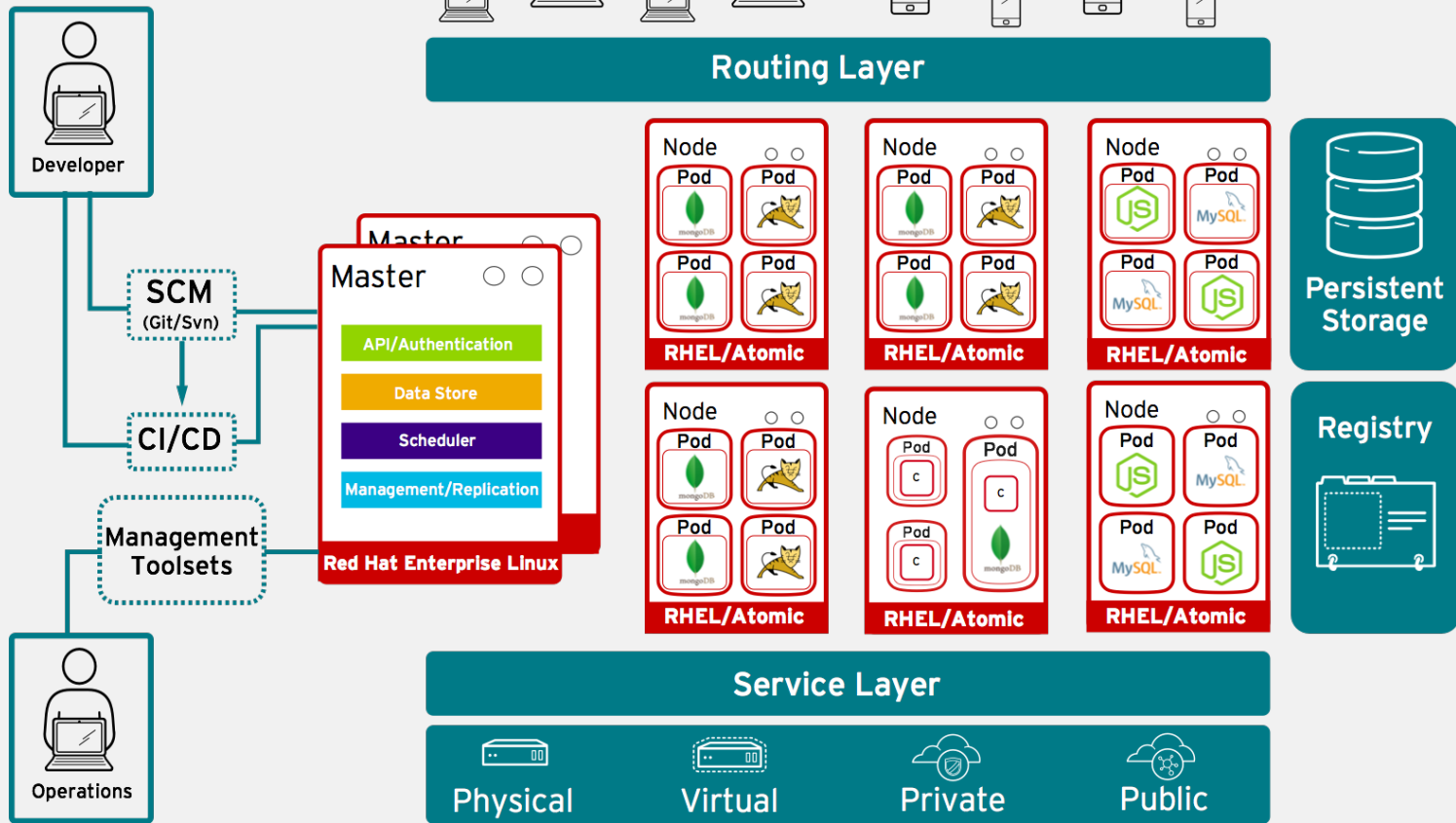
I will use containers for app packaging & delivery and portability

I will use OpenShift – ready for on-prem & cloud, can orchestrate containers and delivers CI/CD functionality

I can easily create new features - OpenShift supports Canary testing, A/B or Blue/Green deployment

CONTAINER





Red Ridding Hood wants to open Start-up

Food like Grandma Used to Make



I will sell products via different channels not via my own eshop

But how to create different channels for different web shop?

I need to offer free and paid microservice.

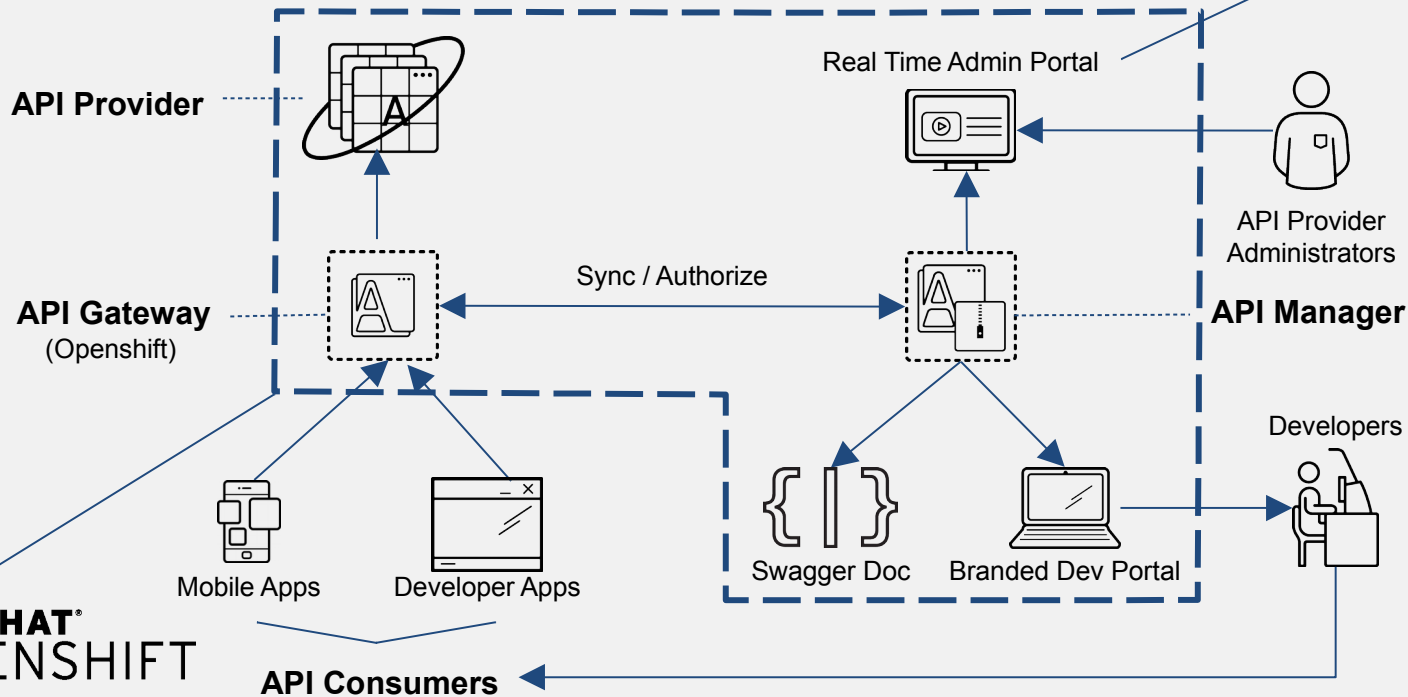
How to protect my back-end against overload?

It will be good to have developer portal for my customers & monetization module, integrated with payment system.

I will use 3scale AMP!!!

3scale AMP

Full On-Premise





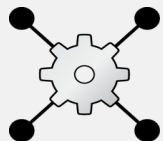


Microservices on Red Hat OpenShift Container Platform (demo)

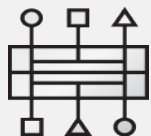
Agile Integration: Reinvent Enterprise Architecture

To remain competitive, businesses need an integration platform capable of supporting current *and* next generation architectures.

Service Integration

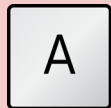


Webservices



APIs

Architecture

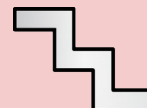


Monolith

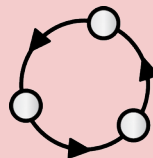


Microservices

Development Process



Waterfall

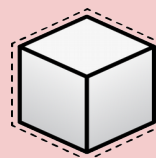


CI/CD

Deployment



Server/VM



Container

Infrastructure



Data Center



Cloud

-- End --



Questions?