redhat.

MONOLITHS TO MICROSERVICES: APP TRANSFORMATION

Hands-on Technical Workshop

Thomas Qvarnström
Sr. Technical Marketing Manager
Middleware BU

James Falkner
Sr. Technical Marketing Manager
Middleware BU

PART 3: MONOLITHS TO MICROSERVICES WITH JAVA EE AND SPRING BOOT

WHY MONOLITH TO MICROSERVICES

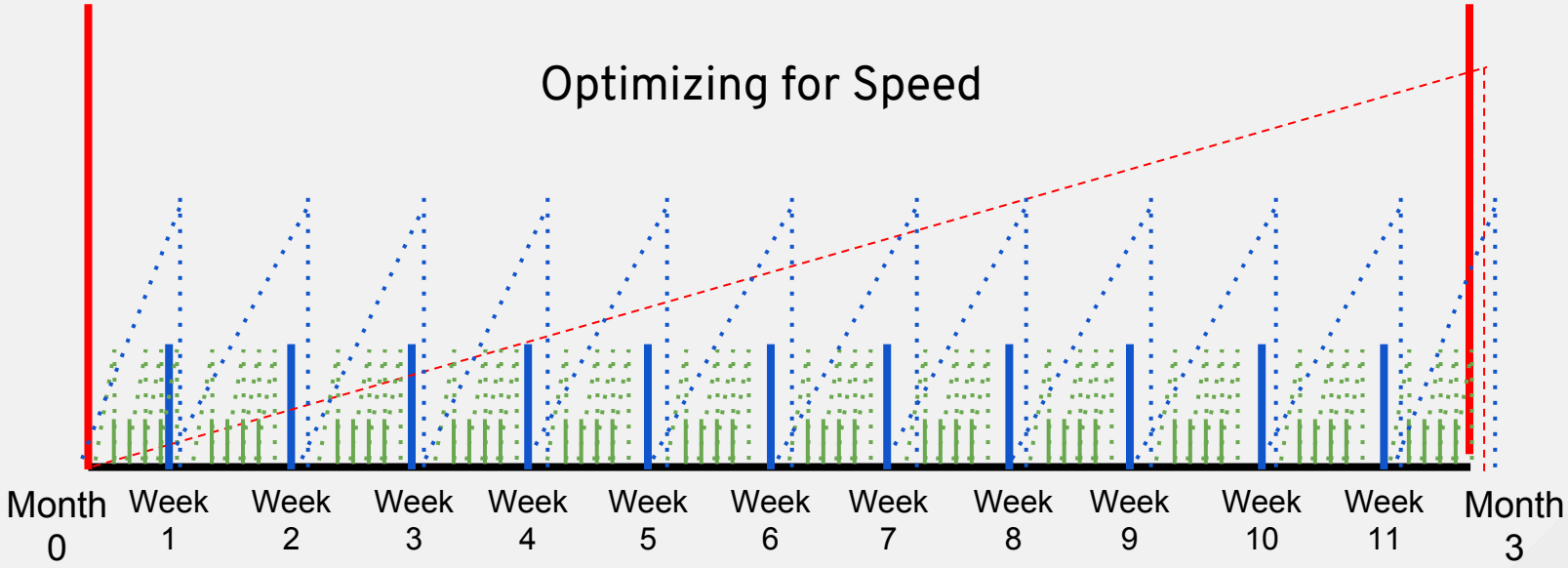
Break things down (organizations, teams, IT systems, etc) down into smaller pieces for greater parallelization and autonomy and focus on reducing time to value.

REDUCING TIME TO VALUE

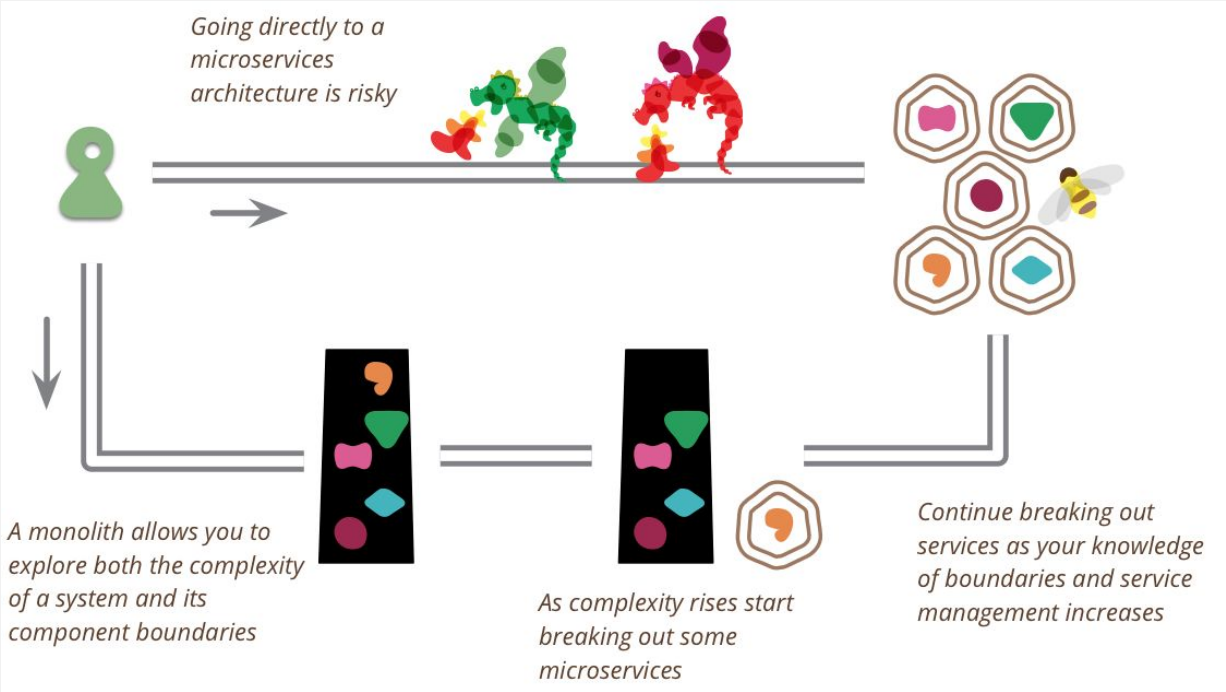
Monolith Lifecycle

Fast Moving Monolith

Microservices



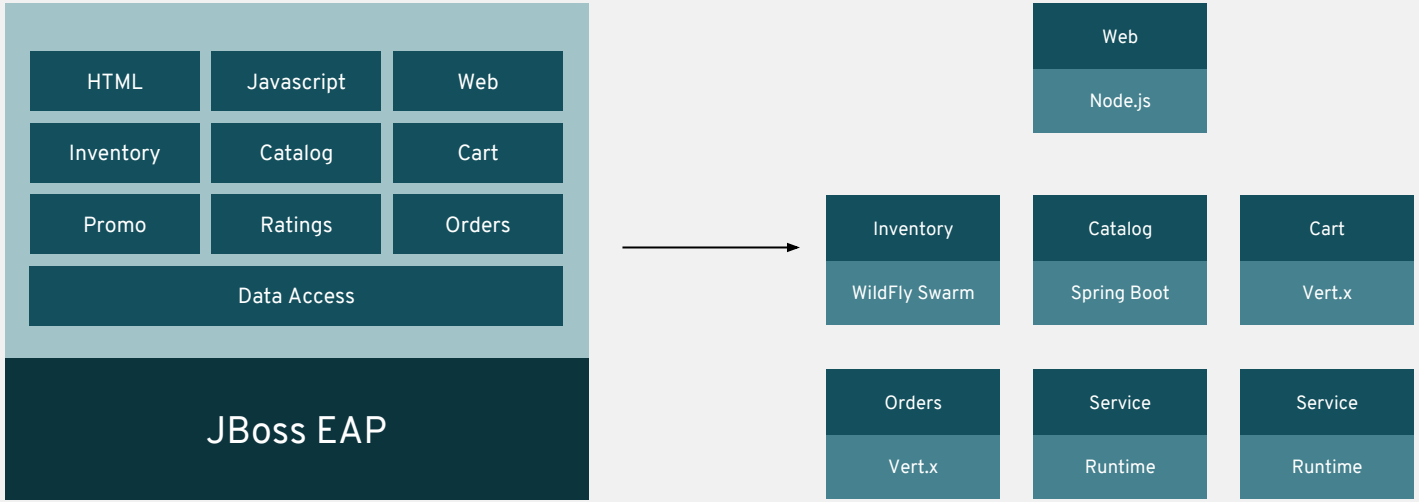
Monolith First?



http://martinfowler.com/bliki/MonolithFirst.html

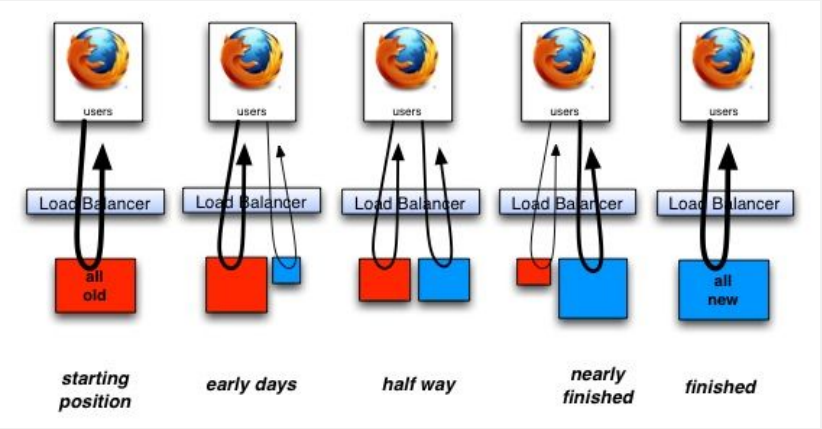
STRANGLING THE MONOLITH

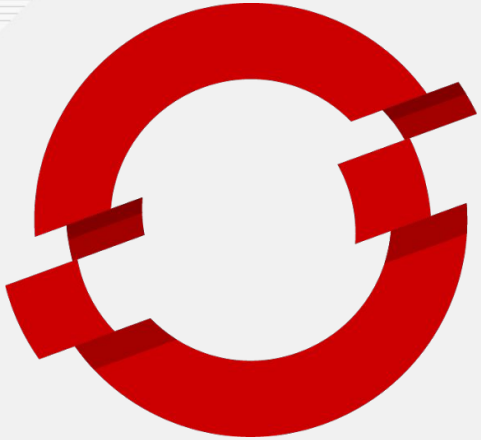
- In this lab, you will begin to ‘strangle’ the coolstore monolith by implementing its services as external microservices, split along business boundaries
- Once implemented, traffic destined to the original monolith’s services will be redirected (via OpenShift software-defined routing) to the new services



STRANGLING THE MONOLITH

- Strangling - **incrementally** replacing functionality in app with something better (cheaper, faster, easier to maintain).
- As functionality is replaced, “dead” parts of monolith can be removed/retired.
- You can also wait for all functionality to be replaced before retiring anything!
- You can optionally include new functionality during strangulation to make it more attractive to business stakeholders.

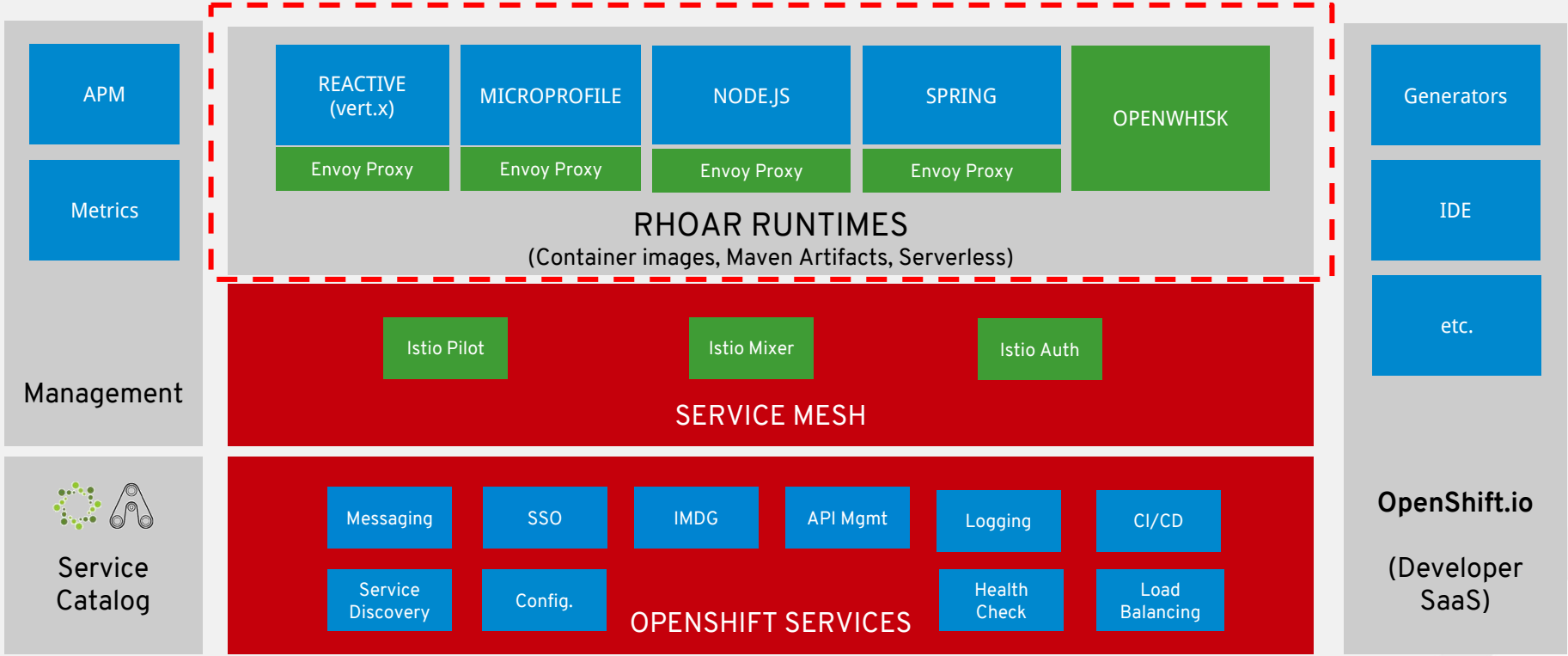




RED HAT® OPENSHIFT

Application Runtimes

RHOAR PRODUCT ARCHITECTURE



== Future

ENTERPRISE JAVA

RED HAT JBOSS
ENTERPRISE
APPLICATION PLATFORM

SPRING APPS

JAVA MICROSERVICES

JAVASCRIPT FLEXIBILITY

REACTIVE SYSTEMS

TOMCAT SIMPLICITY

RED HAT JBOSS
WEB SERVER

THE BIGGER PICTURE: THE PATH TO CLOUD-NATIVE APPS

A DIGITAL DARWINISM

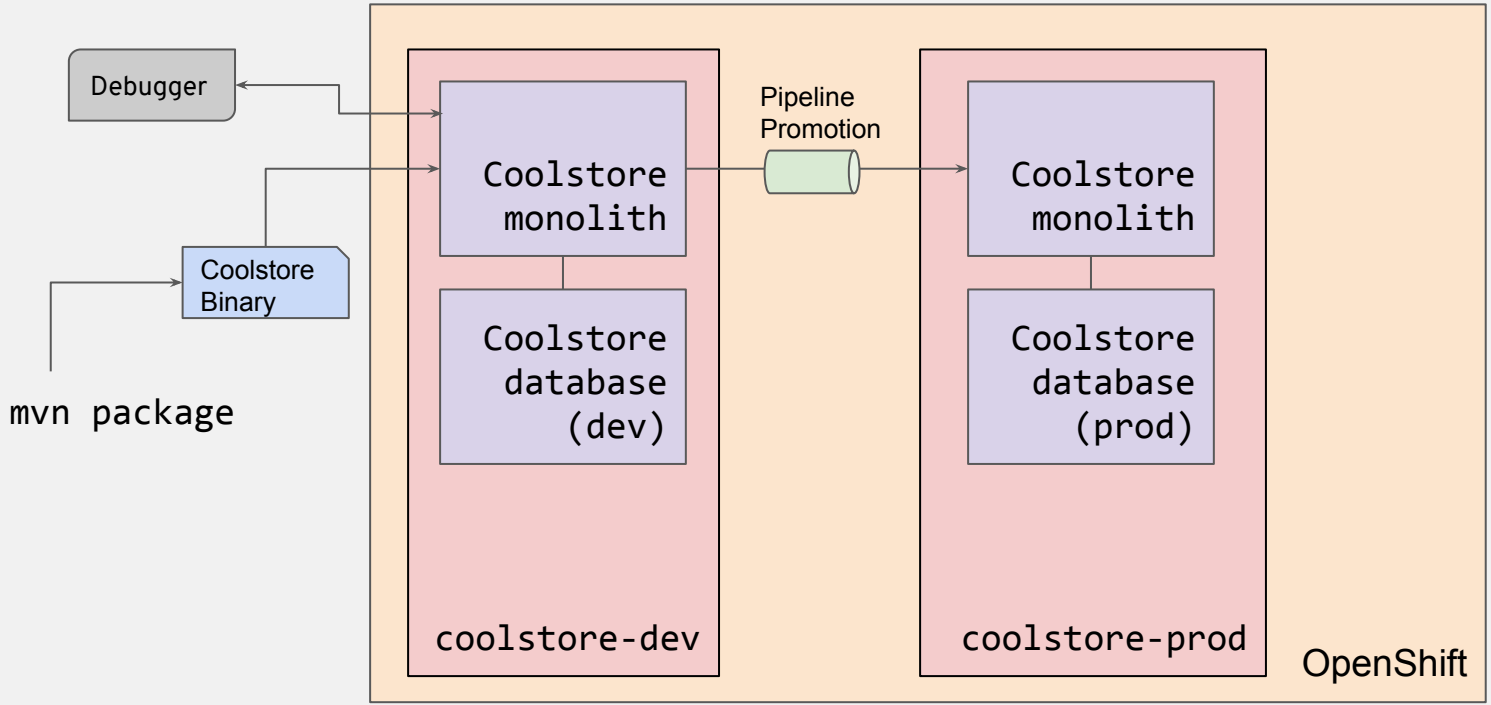
LAB: MONOLITHS TO MICROSERVICES WITH JAVA EE AND SPRING BOOT

GOAL FOR LAB

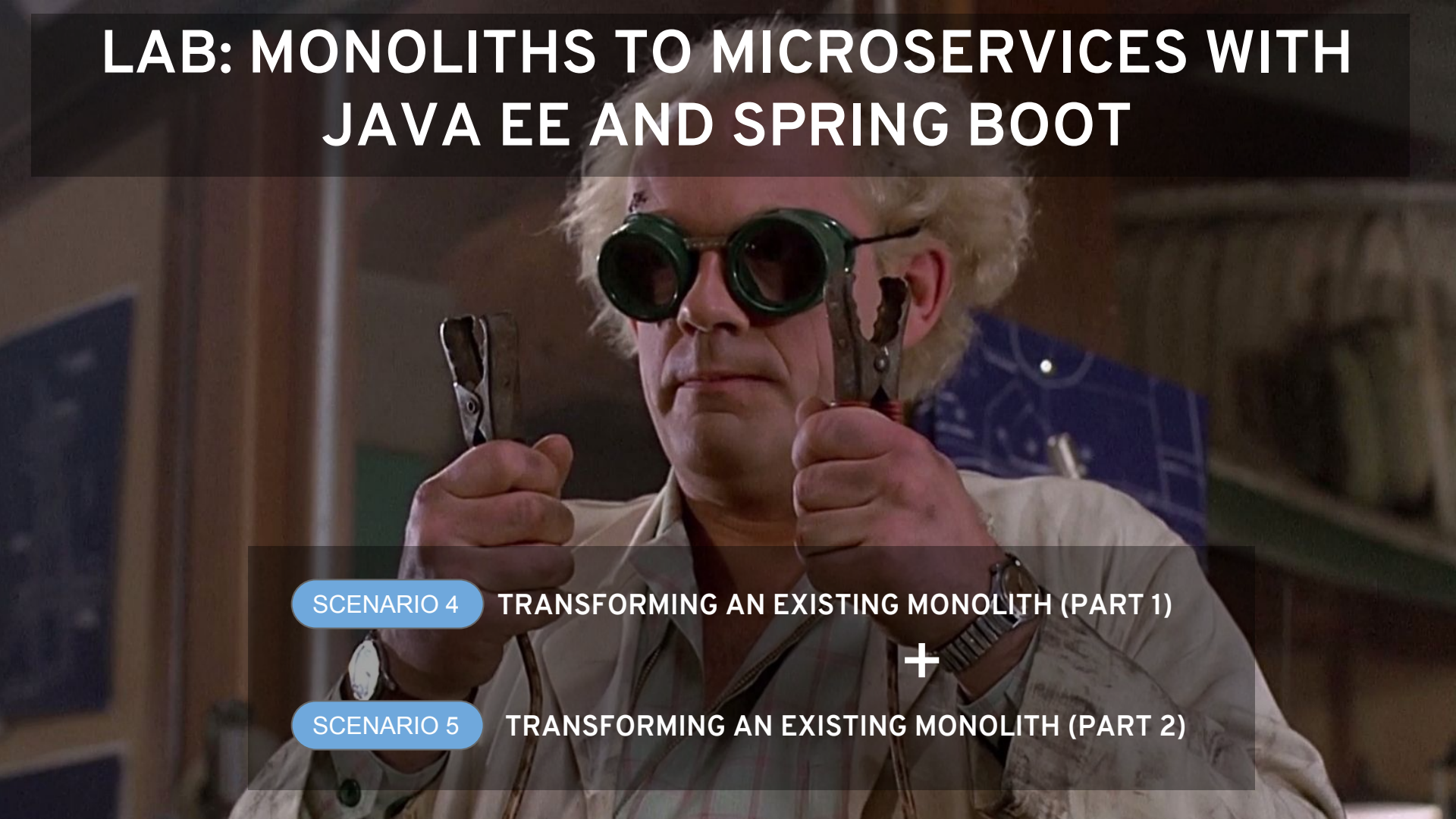
In this lab you will learn:

- How Red Hat OpenShift and Red Hat OpenShift Application Runtimes (RHOAR) help jumpstart app modernization
- Benefits and challenges of microservices
- How to transform existing monolithic applications to microservices using strangler pattern and 12-factor app patterns.
- Use modern app dev frameworks like WildFly Swarm and Spring Boot to implement microservice applications on OpenShift

CURRENT STATE - THE MONOLITH



LAB: MONOLITHS TO MICROSERVICES WITH JAVA EE AND SPRING BOOT



SCENARIO 4 TRANSFORMING AN EXISTING MONOLITH (PART 1)

+

SCENARIO 5 TRANSFORMING AN EXISTING MONOLITH (PART 2)

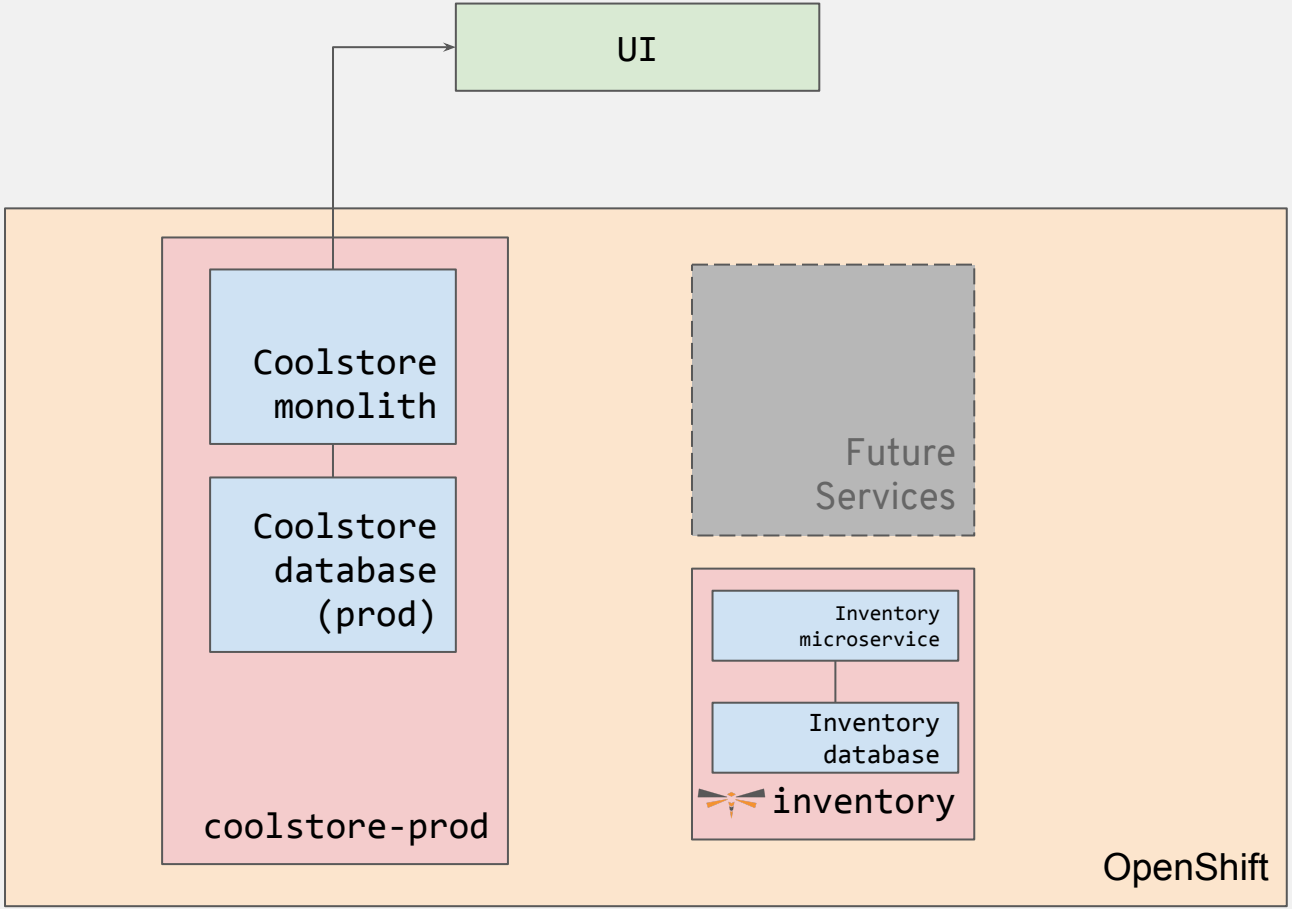
WRAP-UP AND DISCUSSION

RESULT OF LAB

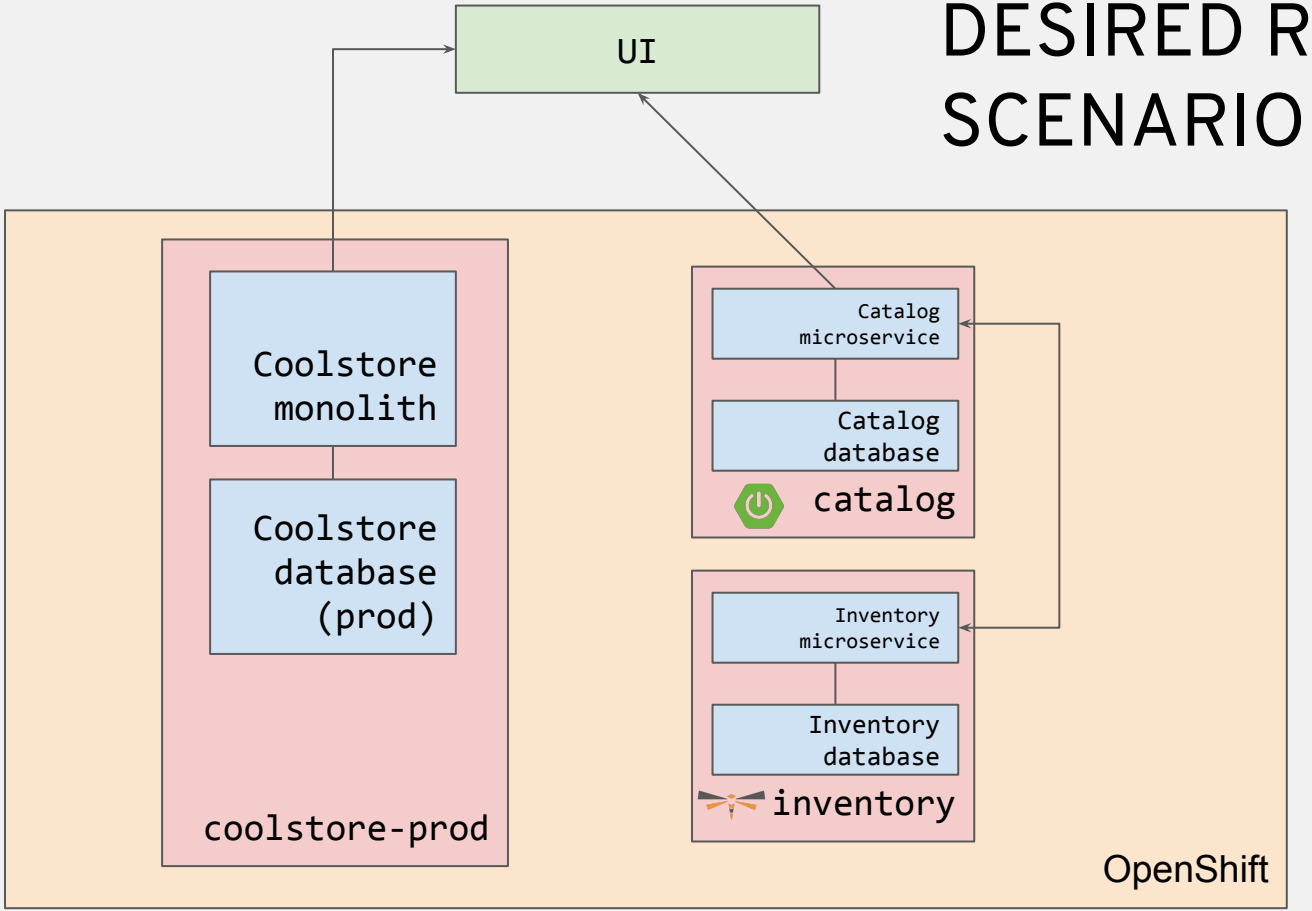
In this lab you learned how to:

- Implement a Java EE microservice using WildFly Swarm
- Implement a Java EE microservice using Spring Boot
- Develop container-based testing
- Add microservice concerns like Health checks, externalized configuration and circuit breaking
- Use the strangler pattern to slowly migrate functionality from monolith to microservices

RESULT OF



DESIRED RESULT OF SCENARIO 5



redhat.

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHatNews

youtube.com/user/RedHatVideos