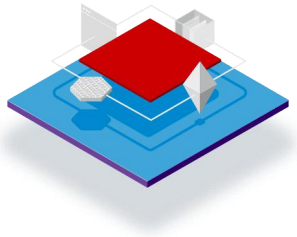


Red Hat Fuse – 7.x

A Distributed, Cloud-native Integration Platform

– Avadhut

$ whoami

- I'm Avadhut!
- Open-source enthusiast & Ex-redhatter ;)
- Active community member for couple of open-source projects
- Working as Senior Integration Developer, Consultant & Architect for different clients
- Worked on Middleware Integration using Fuse, Camel, Karaf, Kafka and messaging platforms for quite a bit
- Did full production deployments, architecture review and performance tuning for couple of employers and lot of Red Hat customers

** You can find me on:

https://kodtodya.github.io

Pre-Requisite for Fuse Course

- Ability to use command line
- Knowledge of Java or overall programming (We will use Java-8 for this course)
- Knowledge of Spring framework and Spring Boot is mandatory
- Linux(Any flavor) and Mac are strongly preferred, avoid Windows if possible
- Knowledge of Maven is mandatory too..
- Knowledge of Webservice, JMS and database is also required

- Willingness to learn an awesome technology... ☺

Who is this course for?

- **Developer**: who would like to learn how to write and run an application that integrates various systems and servers

- **Architects**: who want to understand the role of Fuse in the enterprise integration with/using micro-services

- **DevOps**: who want to understand how Fuse works with regards to various systems, protocols, components and its infrastructural setup

Expectations & Target

- Participants should understand what is Fuse and how to convert business requirements to Fuse design.
- Participants should be able to develop micro-services using spring-boot and camel.
- Participants should understand the Enterprise Integration Patterns in integration development.
- Participants should be able to follow the coding standards while designing the services
- Participants should be able to deploy application in development environment using openshift.

Welcome...!!!

A warm welcome to Red Hat Fuse-7.x Rapid Track course..

Course Structure

- **Part – 1 : Fundamentals**
  - What is Integration and it's need
  - History of integration
  - Real world problems with traditional deployment
  - What is container?
  - What is (RHJB) EAP?
  - What is (RH)Fuse?
  - What is (RH)A-MQ?
  - What is Spring framework?
  - What is Spring Boot?
  - What is Apache Camel?
  - What is Karaf?
  - What is Openshift?

- **Part – 2 : Integration in Action**
  - Integration options
  - History of Fuse
  - What is Enterprise Integration patterns?
  - What is OSGi? (Quick intro)

Course Structure

- **Part – 3 : Core Concepts**
  - Core Concepts (To be covered in subsequent slides)
  - Fuse Eco-system
  - Fuse Architecture
  - Role of Spring Boot
  - Role of Apache Camel
  - Fuse Management
  - Fuse Operations
  - Fuse Management
  - Fuse Operations

- **Part – 4 : Fuse in Action**
  - Introduction to Fuse-7.x
  - Fuse sub-systems
  - Management HawtIO
  - Deployment strategies

Exceptions

What I am not going to cover

- Openshift clustering
- Fuse Online/Ignite
- Your UAT and production issues
- Fuse Migrations to latest version
- Your enterprise application issues
- CI/CD jobs creation & nexus setup
- Hotfixes and patches to any environment
- Your enterprise project architecture & migration consultation
- Third party license software & tool usage or consultation about solution with it.

If you have any of the above use-case,

Please reach out to me for best pricing for solution at earliest:

https://kodtodya.github.io/talks

Red Hat Fuse Theory

Fundamentals

What is Integration?

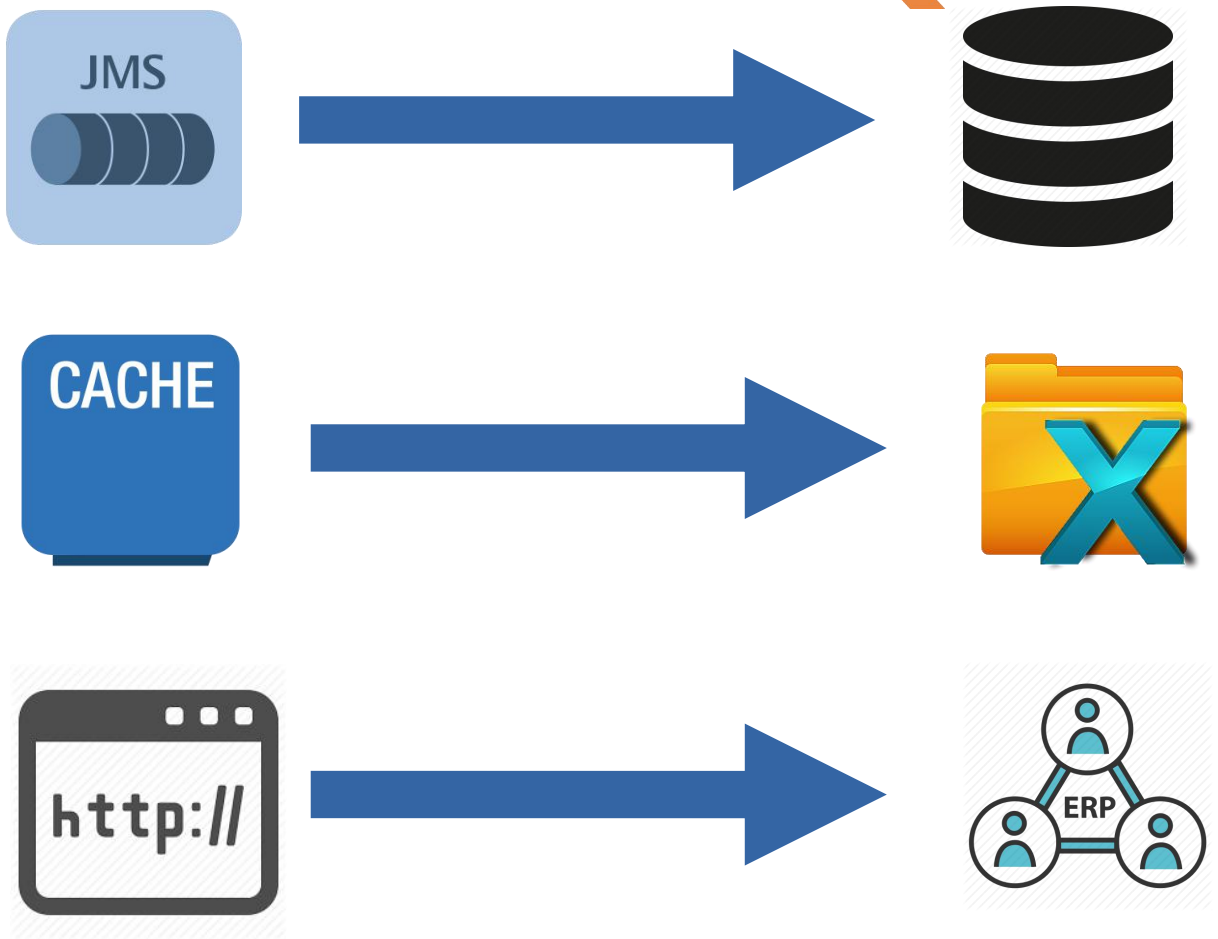
Act of bringing together smaller components into a single system that functions as one

- For us, it refers to the end result of a process that aims to stitch together different, often disparate, subsystems so that the data contained in each becomes part of a larger, more comprehensive system that, ideally, quickly and easily shares data when needed

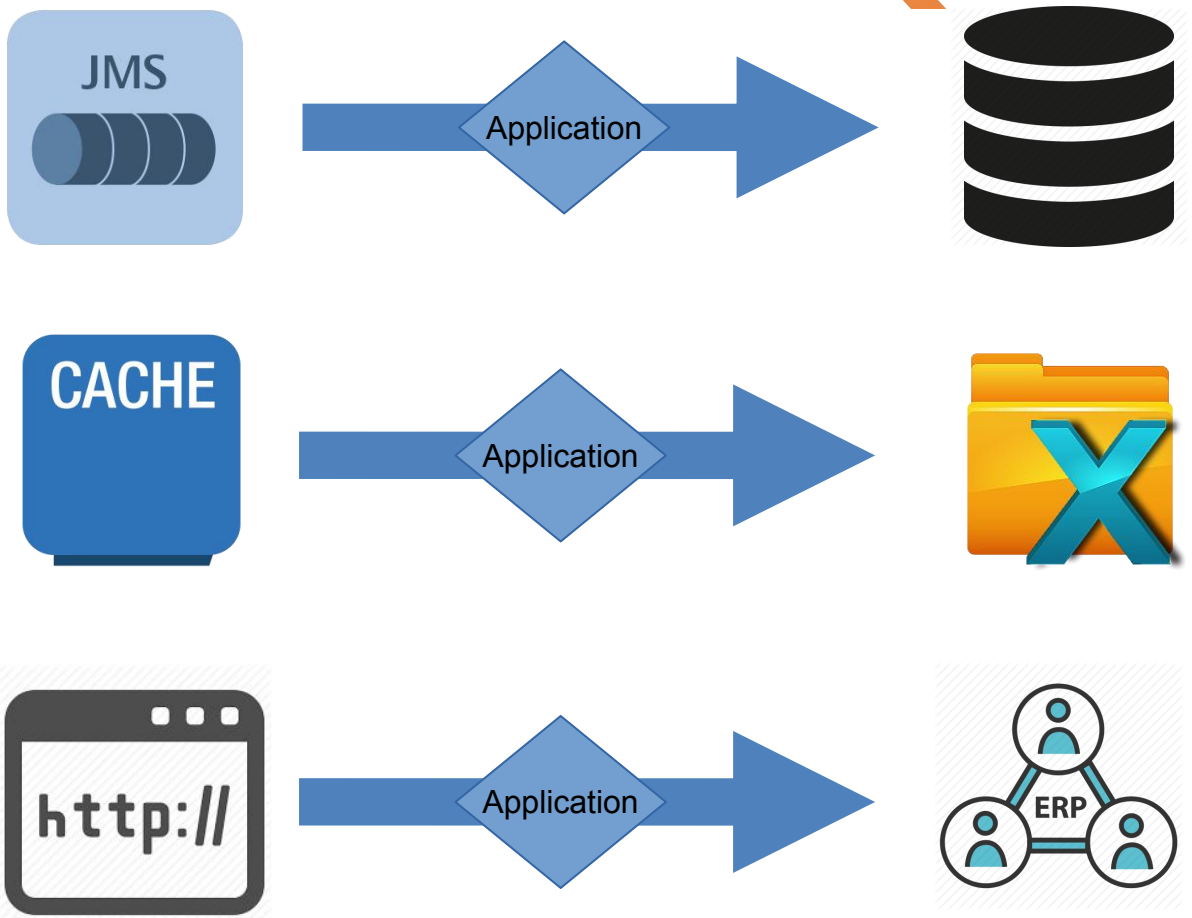
Need of Integration

- Connect with each other
- Incorporate in the system
- Amalgamation of components/sub-systems
- Healthy communication seamlessly in components/sub-systems
- Consolidation of components
- Assimilation
- Share data in sub-systems

History of Integration

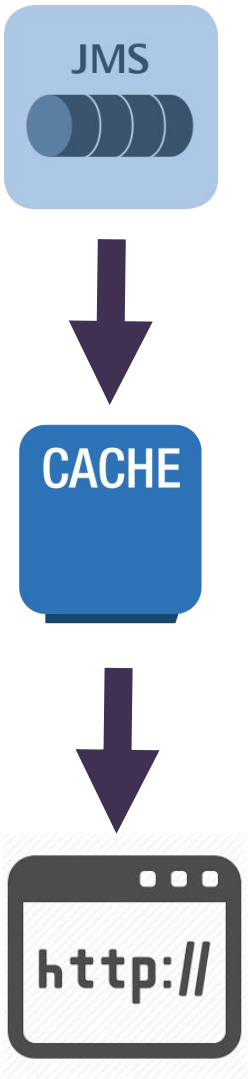


History of Integration



Real world Problems of Integration

- Real issue is actually here..
- What if, my requirement is changed and I need to change my integration in different way
- What if, my one of the system is communicating with unusual protocol and to handle that you need another certain stuffs



Real-world Problems In Integration

History of Integration

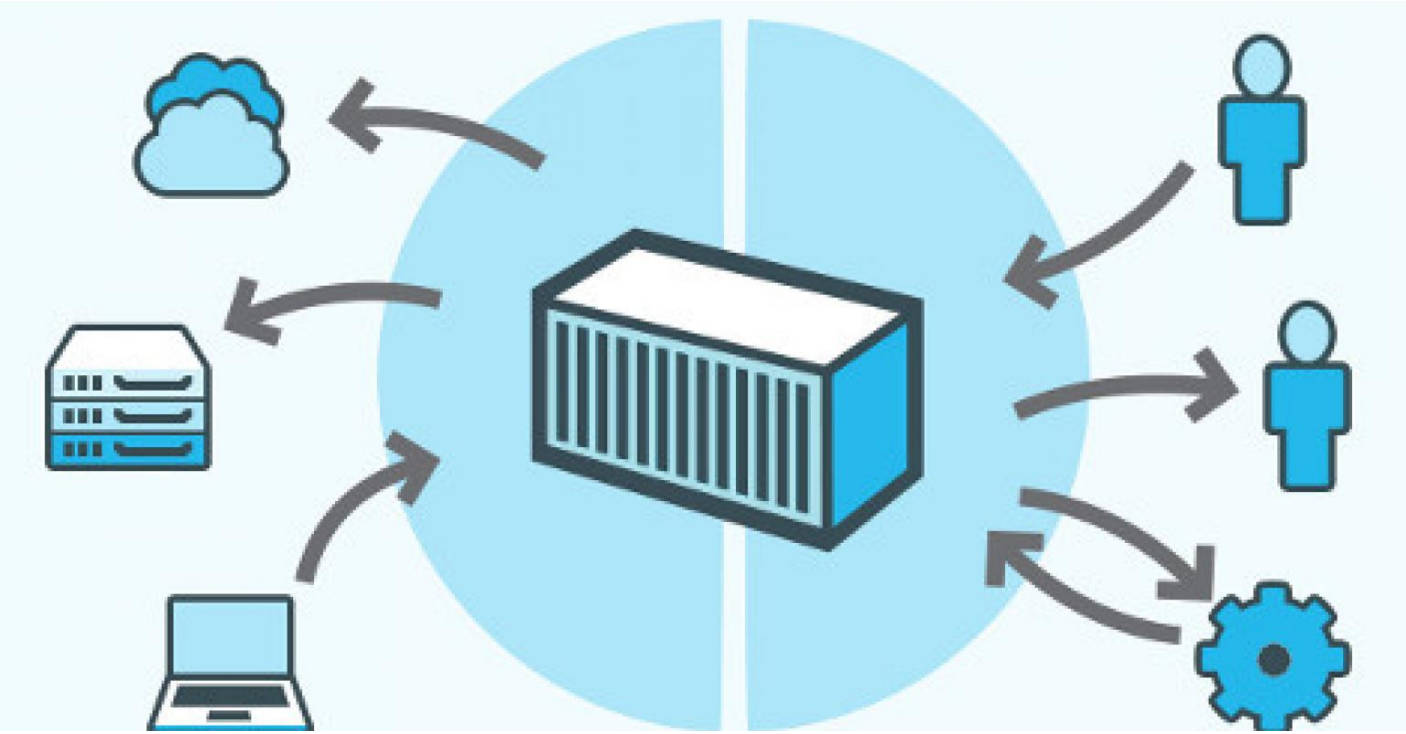
- To get out of this complication, you obviously need to write new code that will connect one system to another
- Now, this application should be also aware that what is output type of source system and what is input type of target system. i.e. Feed, File, JSON, DB record, POJO, cache, etc.
- Now, this also should have standard exception handling and even performance tuning too..

What is container?

An object for holding or transporting something..

What is container?

For us, it is a standardized unit of software..

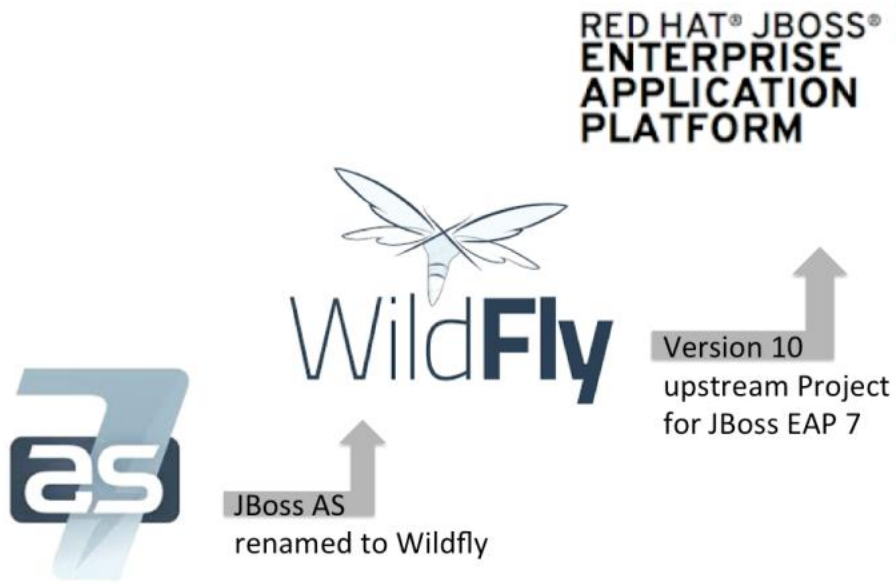


Is Container a future?

- We are using below types of containers in Fuse till now:

- Spring Boot containers
- Karaf Containers
- Fabric containers
- AMQ containers
- Openshift containers

What is EAP?



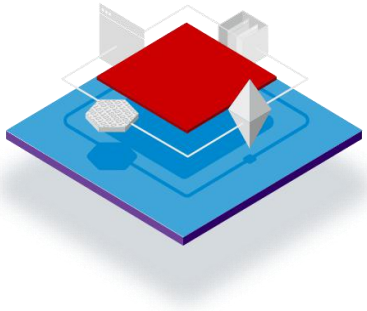
- One of the leading open-source Application server
- Community WildFly-10 is the upstream product of EAP-7
- Integration applications also can be deployed in EAP-7

What is AMQ?

RED HAT®
AMQ

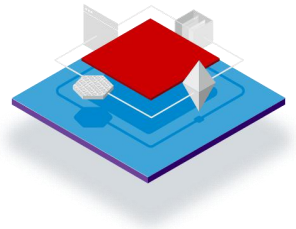
- One of the leading open-source messaging broker
- Community Apache ActiveMQ Artemis is the upstream product of (RH)AMQ-7
- This is provided as embedded broker in Fuse

What is (RH)Fuse?



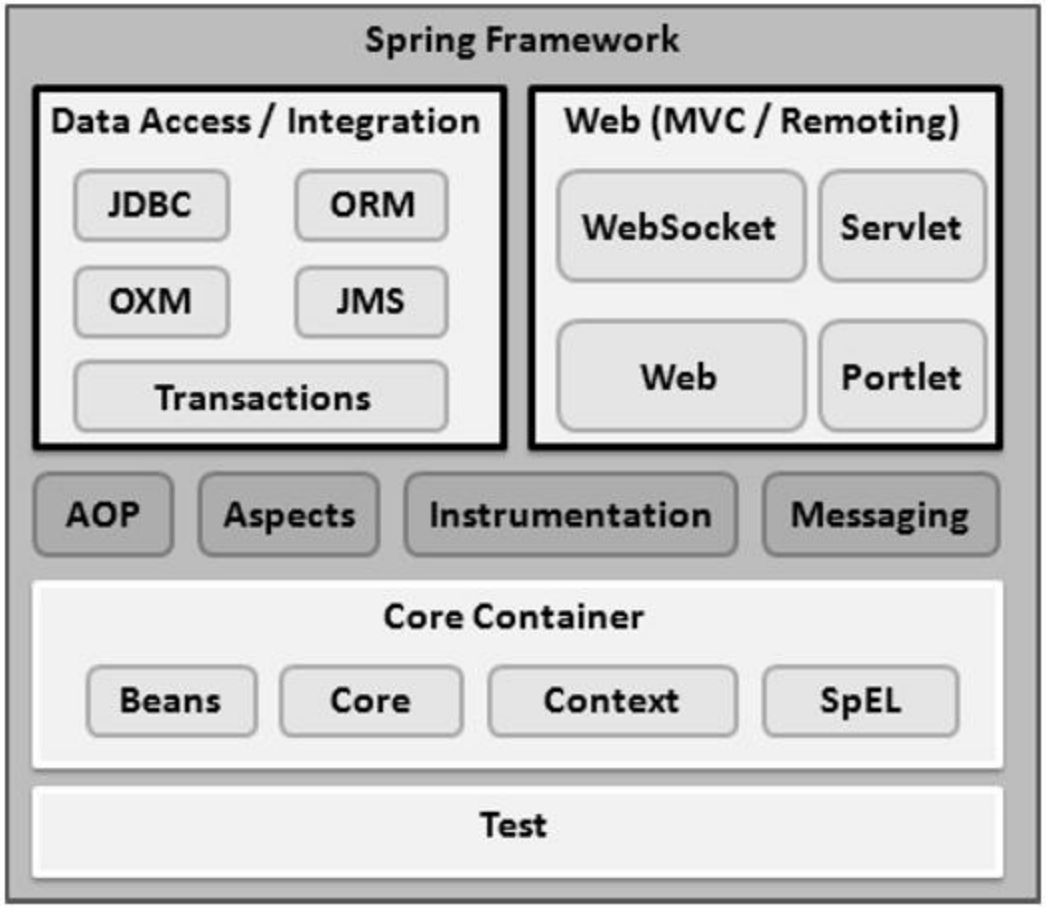
- Leading light-weight, flexible, open-source, OSGi Complaint and distributed integration platform based on Camel
- Subset of agile integration solution
- Set of best community products to provide highly scalable, API-centric, container-based architecture to decouple services so they can be created, extended, and deployed independently
- Provides a standardized methodology, infrastructure, and tools to integrate services, micro-services, and application components.
- Reduces the pain of connecting applications, services, processes and devices for comprehensive and efficient integration
- Can be installed and configured based on your requirement, budget and available resources. e.g. Fuse on Openshift, Fuse standalone, Fuse Online
- Fuse can be integrated with J2EE container like RH JBoss EAP

(RH)Fuse Benefits



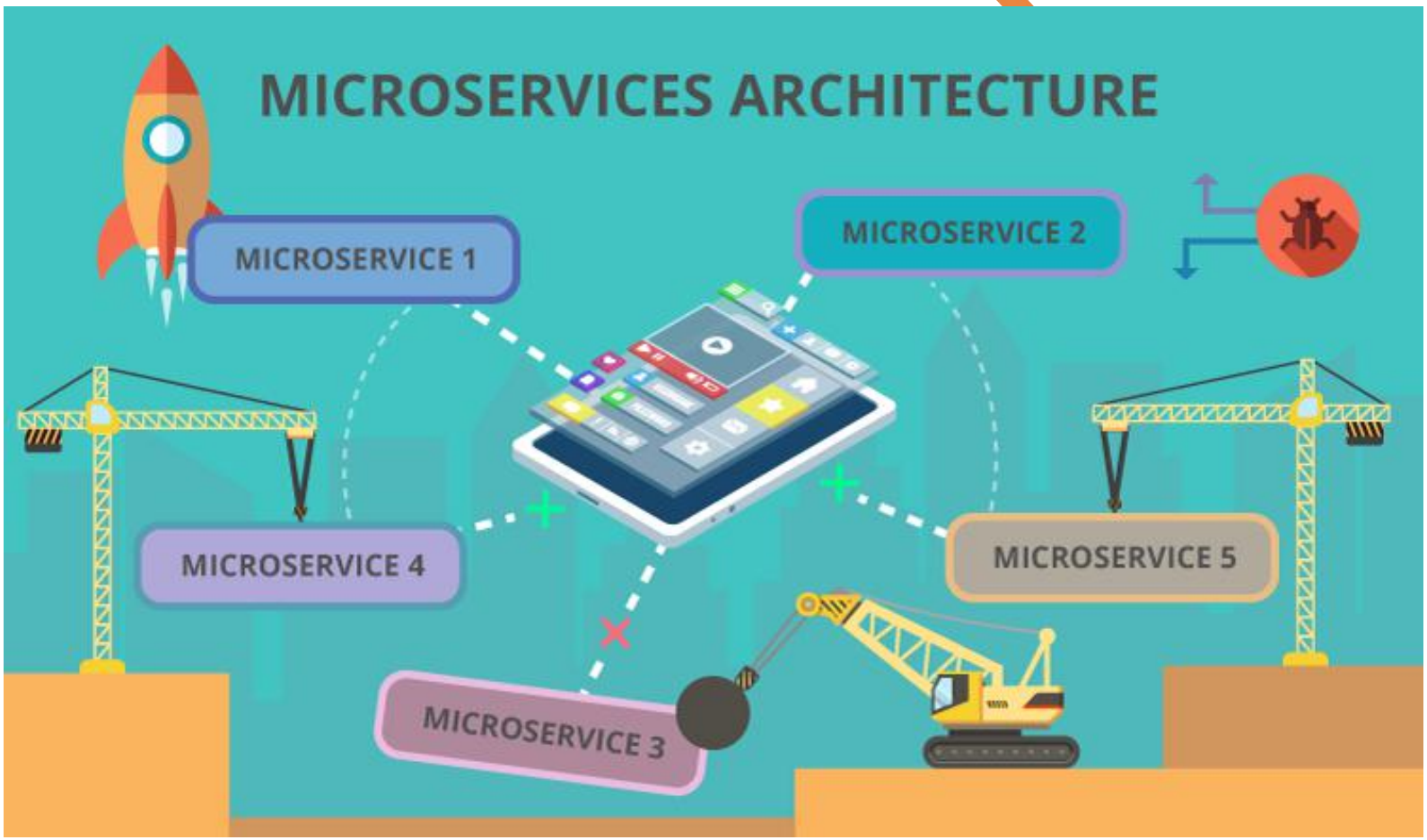
- Hybrid deployment
- Built-in iPaaS with low-code UI/UX
- Container based integration
- Integration everywhere
- Camel provides huge number of connectors, so reduces pain of connectivity
- Supports Micro-services architecture / distributed infrastructure
- Can be integrated with EAP(Application Server) container to fulfill business needs
- Low-code interface (Fuse tooling allows non-technical users to deal with integration)

What is Spring framework?

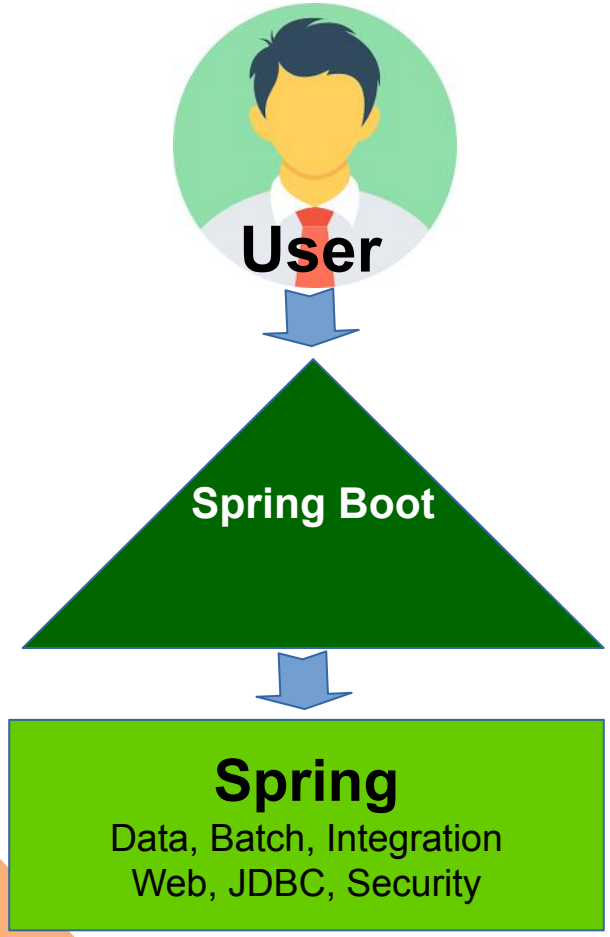


- Most popular application development framework for enterprise Java
- Useful to create high performing, easily testable, and reusable code
- Extreme lightweight and comes with different extension

What is Micro-service?

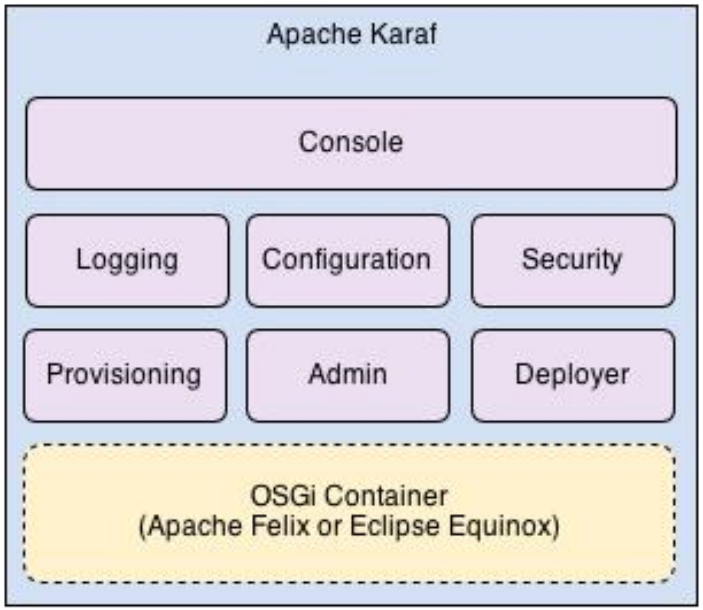


What is Spring Boot?



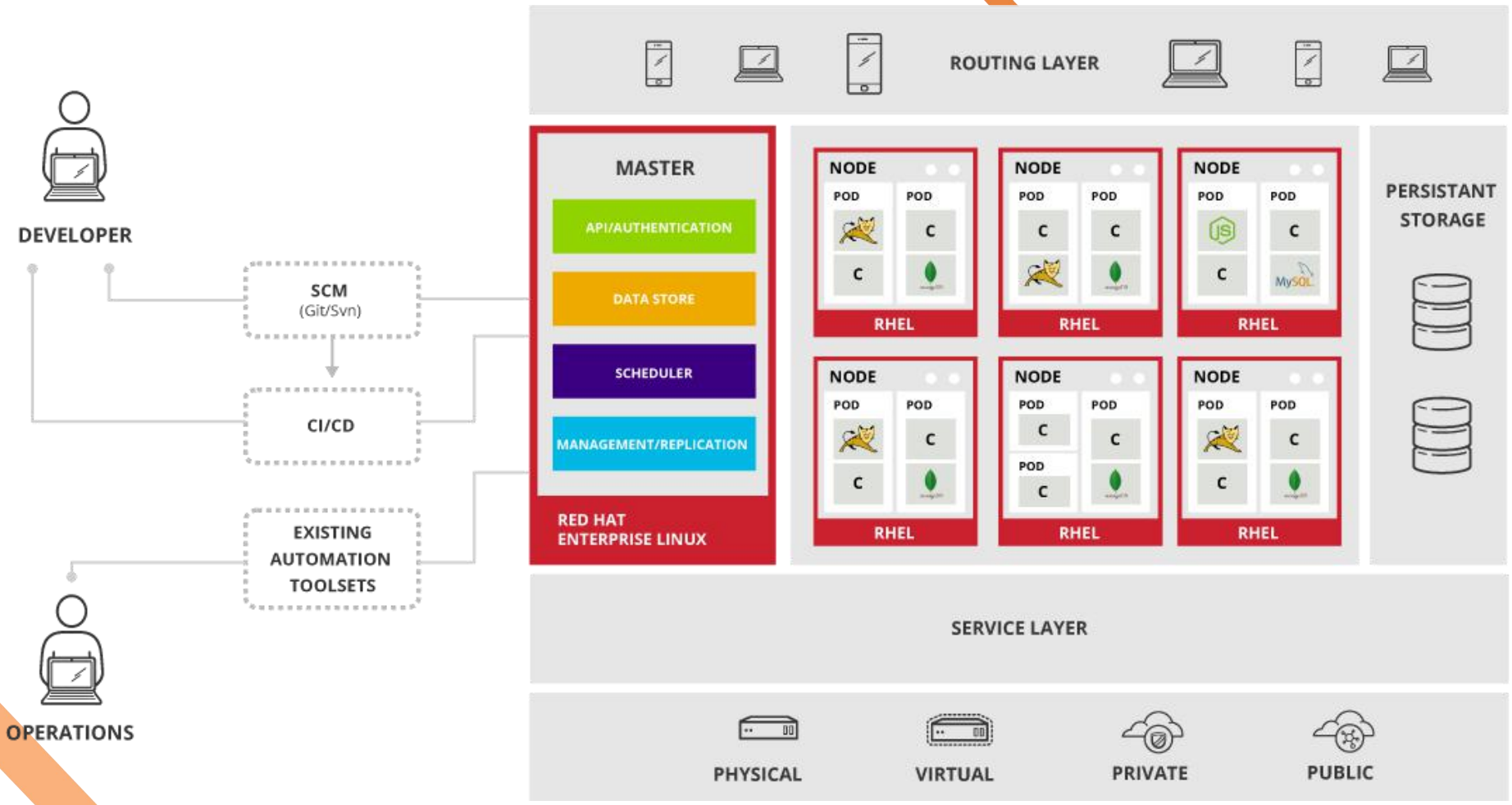
- Spring Boot is not a framework
- It is a way to ease to create stand-alone application with minimal or zero configurations
- Leverages existing spring projects as well as Third party projects to develop production ready applications
- Provides a set of Starter POM's or gradle build files which one can use to add required dependencies and also facilitate auto configuration

What is Apache Karaf?



- OSGi powered modern and polymorphic container
- Lightweight, powerful, and enterprise ready platform
- Karaf is the perfect solution for microservices, systems integration, big data, and much more
- Can be scaled from a very lightweight container to a fully featured enterprise service
- very flexible and extensible container, covering all the major needs

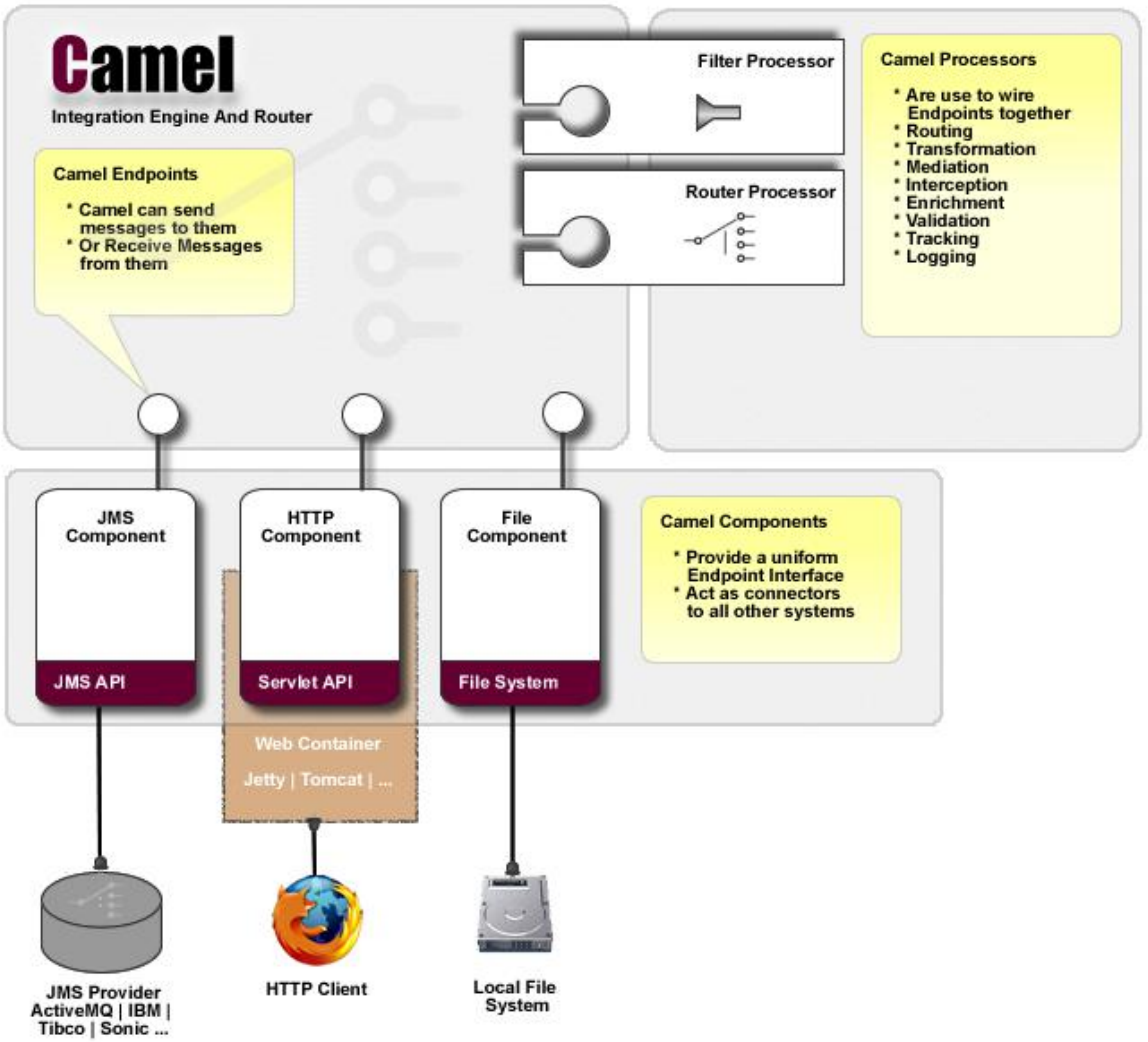
What is OpenShift?



What is OpenShift?

- Open, hybrid cloud Kubernetes platform to build, run, and scale container-based applications
- includes everything you need to manage your development life-cycle, including standardized workflows, support for multiple environments, continuous integration, and release management
- Built around a core of application containers powered by Docker, with orchestration and management provided by Kubernetes, on a foundation of Red Hat Enterprise Linux
- Openshift is much more than just a hybrid cloud

What is Apache Camel?



What is Apache Camel?

- Versatile open-source integration framework based on known Enterprise Integration Patterns

- Empowers you to define routing and mediation rules in a variety of domain-specific languages, including a Java API, Spring/Blueprint XML Configuration files, and a Scala DSL

- Open source framework for message-oriented middleware with a rule-based routing and mediation engine

Lets revise – Part 1

- **Part – 1 : Fundamentals**
  - What is Integration and it's need
  - History of integration
  - Real world problems with existing traditional deployment
  - What is container?
  - What is (RHJB) EAP?
  - What is (RH)Fuse?
  - What is (RH)A-MQ?
  - What is Spring framework?
  - What is Spring Boot?
  - What is Apache Camel?
  - What is Karaf?
  - What is Openshift?

Questions?

Thank you..!!!

LinkedIn, GitHub, GitLab, Twitter: @kodtodya

https://kodtodya.github.io/talks/