

# Red Hat Fuse – 7.x

A Distributed, Cloud-native Integration Platform

– **Avadhut**

# Course Structure

## ● Part – 3 : Core Concepts

- Core Concepts (To be covered in subsequent slides)
- Fuse flavors/Offerings
- Fuse Eco-system
- Fuse Sub-systems
- Fuse Architecture
- Role of Spring Boot
- Role of Apache Camel





# Red Hat Fuse Theory

## Core Concepts



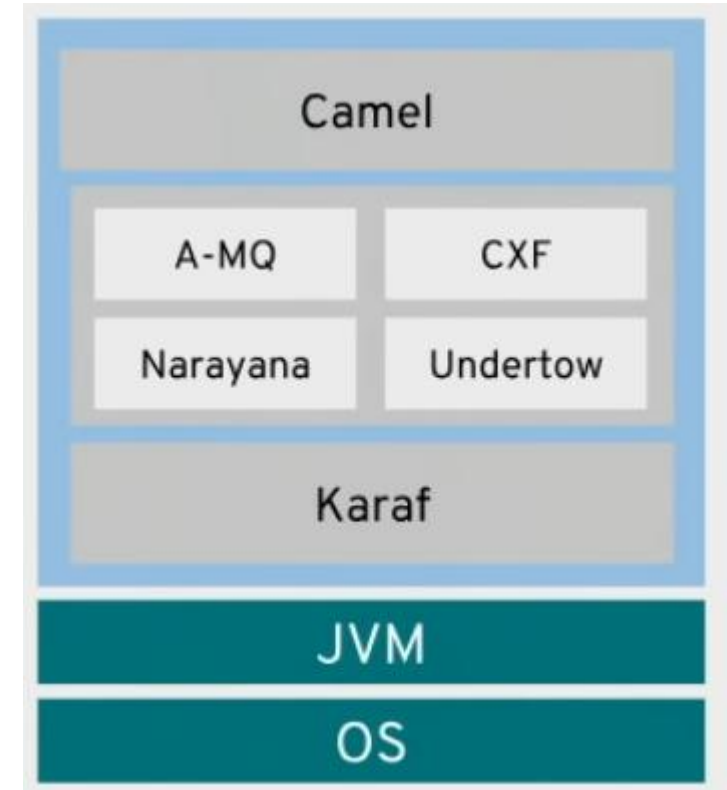
- **Fuse Standalone**
  - ◆ **Fuse on Karaf**
  - ◆ **Fuse on EAP**
  - ◆ **Fuse on Spring Boot**
- **Fuse on Openshift**
- **Fuse Online/Fuse Ignite**



# Fuse Standalone

## Fuse on Karaf

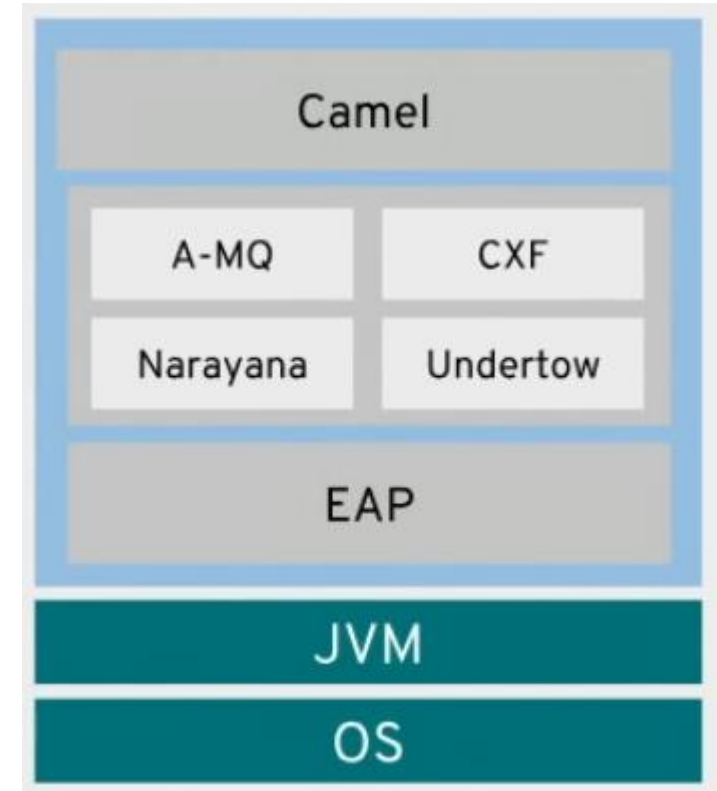
- Standalone OSGi R6 compliant/Karaf 4 container
- Classloader Isolation
- Highly dynamic module system
- Run the integration applications in local Karaf container
- Can generate small footprint servers
- Clustering is not possible in Karaf containers in standalone Fuse distribution



# Fuse Standalone

## Fuse on EAP

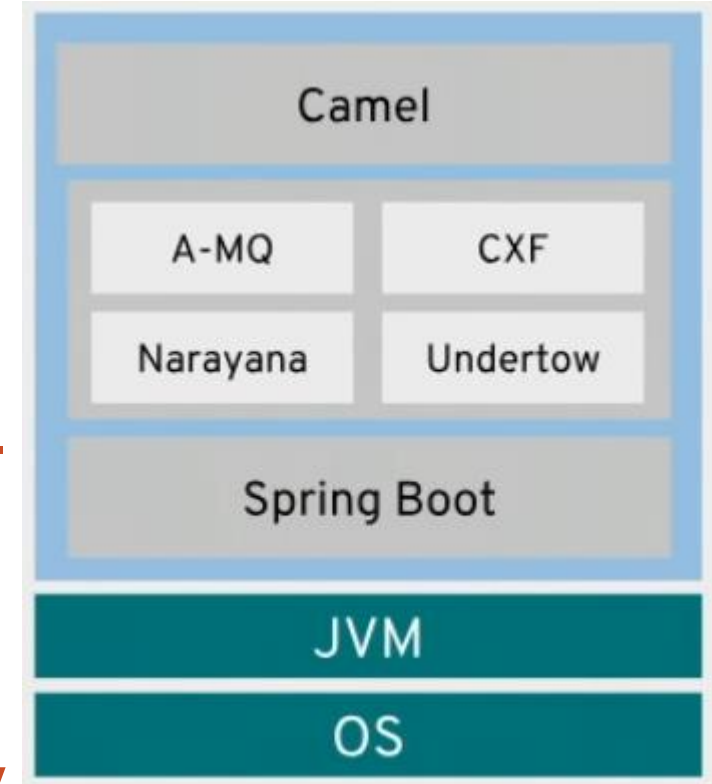
- J2EE/EAP-7.1 and OSGi compliant container
- Camel via Module System
- Integration logic can be written in web applications
- Slip Application WARs
- Deployment on JBoss EAP container
- Smooth Patching and upgrade experience
- Fuse on EAP offers enterprise J2EE applications to carry integration logic



# Fuse Standalone

## Fuse on Spring Boot

- Spring Boot application that provides embedded runtime tomcat container
- Spring Boot 1.5.x
- Flexible Integration Injection
- Generates flat classpath runtime
- Integration logic can be plugged as micro-services
- Deployment is can be done as same as running JAR
- Spring boot offers micro-services strategy to implement integration



# Use Fuse Standalone When...

- I'm Integration developer or architect looking for maximum flexibility and control over how integration is deployed
- Camel based applications targeting single JVM deployment
- High level of control over integration runtime
- Adaptable to traditional (e.g. ESB) and embedded integration architecture
- Provisioning, clustering and automation handled outside of Fuse
- When you prefer pets over cattle



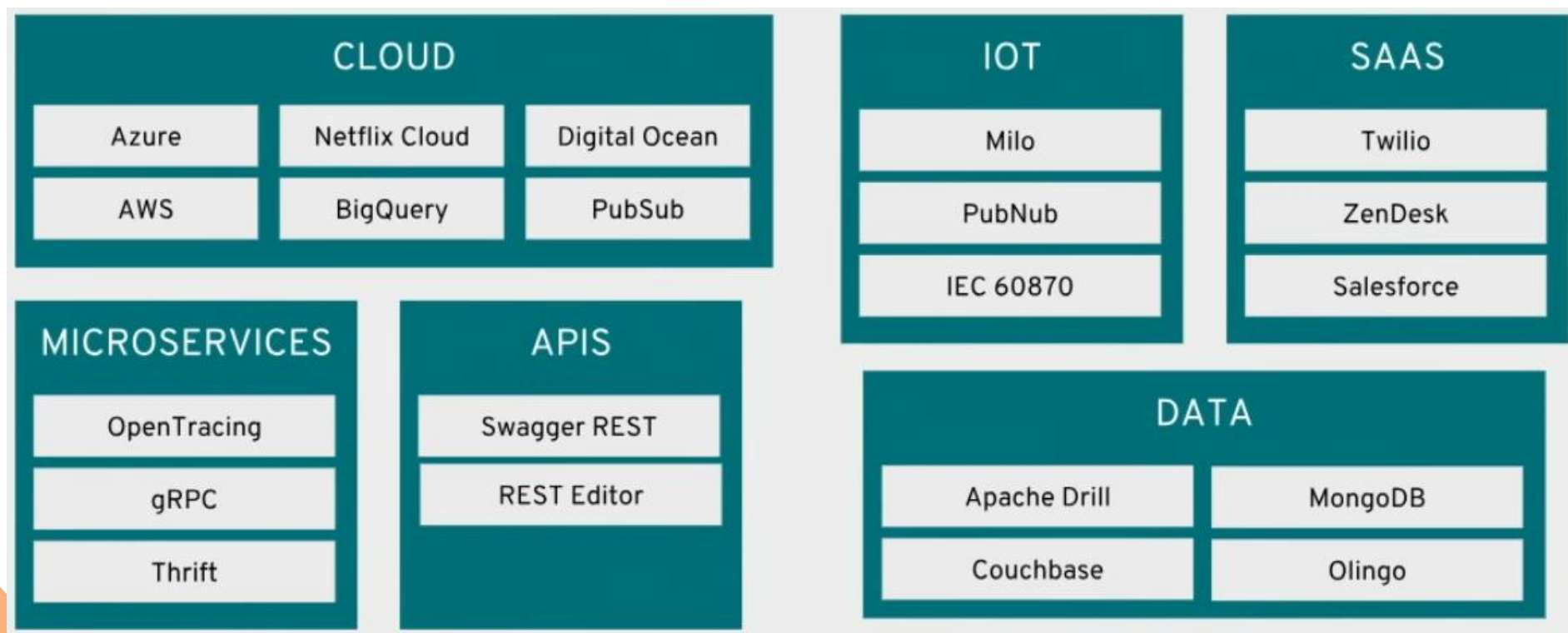


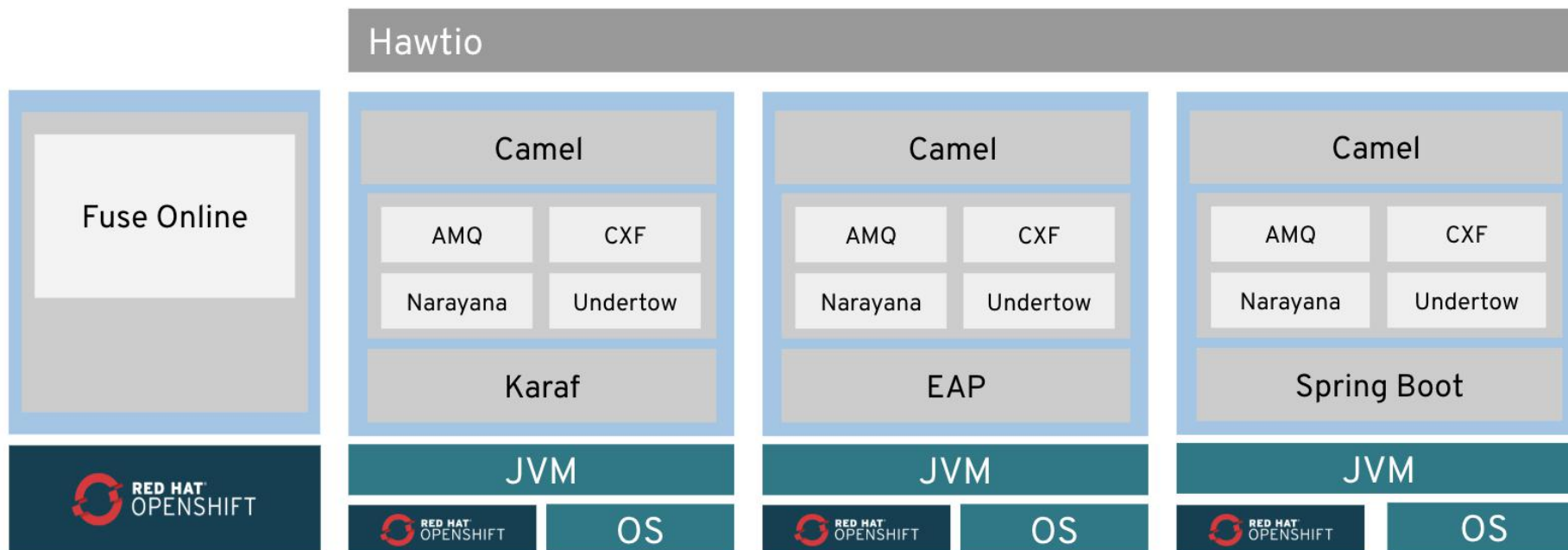
# Fuse Standalone Highlights

- **Update to latest version of Camel (2.21)**
  - ◆ 35+ new connectors – including IoT, SaaS, and API
  - ◆ Micro-services/Container native technologies (Hystrix, Zipkin, Open Tracing, Sagas)
- **Updated standalone management console based on HawtIO**
- **Connector parity across runtimes (EAP, Karaf, Spring Boot)**
- **JDK-8 supported with Fuse-7.x; JDK-11 is under consideration in upcoming 7.x releases**
- **Shared supporting services : Narayana, Undertow, CXF**



# Selection of New Capabilities





- Spring Boot application development with containerization
- Run the integration on OpenShift
- Deployment can be done on private openshift premises or Red Hat hosted openshift premises



- I'm developer with integration requirements targeting cloud native applications that allow for self service, scalability and continuous delivery

- **Running Integrations in Cloud Native Way**

- Reproducible Deployments via Immutable Images
- Service Discovery
- Load Balancing

- **Scale out and Management of Integrations**

- Centralized Logging
- Centralized Metrics
- Health Checks



- **Fuse on Openshift replaces Fabric v1 in Fuse 7.x**
- **Significant focus on management and monitoring of Fuse containers**
  - Centralized HawtIO console
  - Prometheus Metrics Integration
- **Scalability and Performance critical with mature adoption**
  - No breaking changes from Fuse-6.3/FIS
  - JVM Memory optimizations

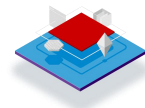


# What is Fuse Online

- Fuse Online is browser based integration interface with Zero\* coding.
- You can obtain data from application or service, operate on that data if you need to, and then send the data to a completely different application or service.
- This allow you integrate two or more different applications or services without writing code.
- It also provides features that allow you to introduce code if it is needed for complex use cases.
- Lets you enable data transfer between different applications



- Business experts in, for example, finance, human resources, or marketing, who do not want to write code in order to share data between two different applications.
- Their use of a variety of software-as-a-service (SaaS) applications gives them an understanding of business requirements, work-flows, and relevant data.



# Benefits of Fuse Online

- **Integrate data from different applications or services without writing code.**
- **Run the integration on OpenShift Online in the public cloud or on OpenShift Container Platform on site.**
- **Use the visual data mapper to map data fields in one application to data fields in another application.**
- **Leverage all the benefits of open source software. You can extend features, and customize interfaces. If Fuse Online does not provide a connector for an application or service that you want to integrate then a developer can create the connector that you need.**





# Fuse Online/Ignite Look

The screenshot displays the 'RED HAT FUSE ONLINE Ignite' web interface. At the top, a navigation bar includes a hamburger menu, the product name, and a user profile 'kbabo1'. Below this, a breadcrumb trail shows 'Home > Connections > Create Connection', with a 'Cancel' button on the right. The main area features a three-step wizard: 'Select Connector' (step 1), 'Configure Connection' (step 2), and 'Name Connection' (step 3). Under the 'Select Connector' step, there are two search filters: 'Name' with a dropdown and 'Filter by Name...' text input, and another 'Name' dropdown with a sort icon. The connectors are presented in a 2x4 grid:

Connector Icon	Connector Name	Description
	Dropbox	File upload and download from Dropbox
	FTP	Upload and download files using File Transfer Prot...
	HTTP	Http Connector
	HTTPS	Https Connector
	MQTT	Publish and Subscribe Messages to a MQTT broker.
	Salesforce	Cloud-based Customer Relationship Management
	SFTP	Upload and download files using Secure File Transf...
	Slack	Send messages to Slack





# Red Hat Fuse

Installation - for Karaf Based Server only



- **Download Fuse-7.2 from Red Hat Portal**
- **Extract it to installation directory**
- **Start your Fuse container first time**

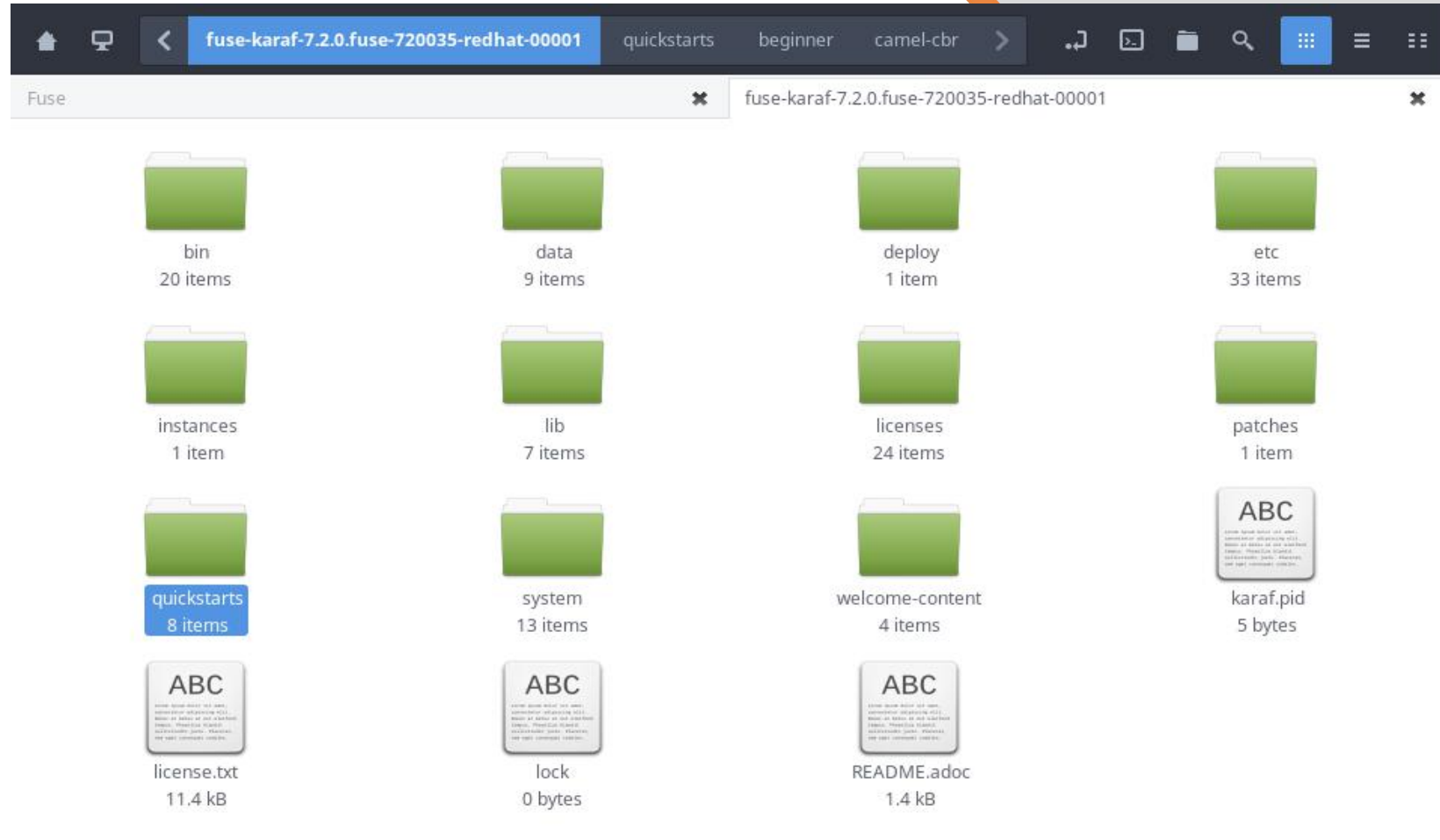




# Red Hat Fuse

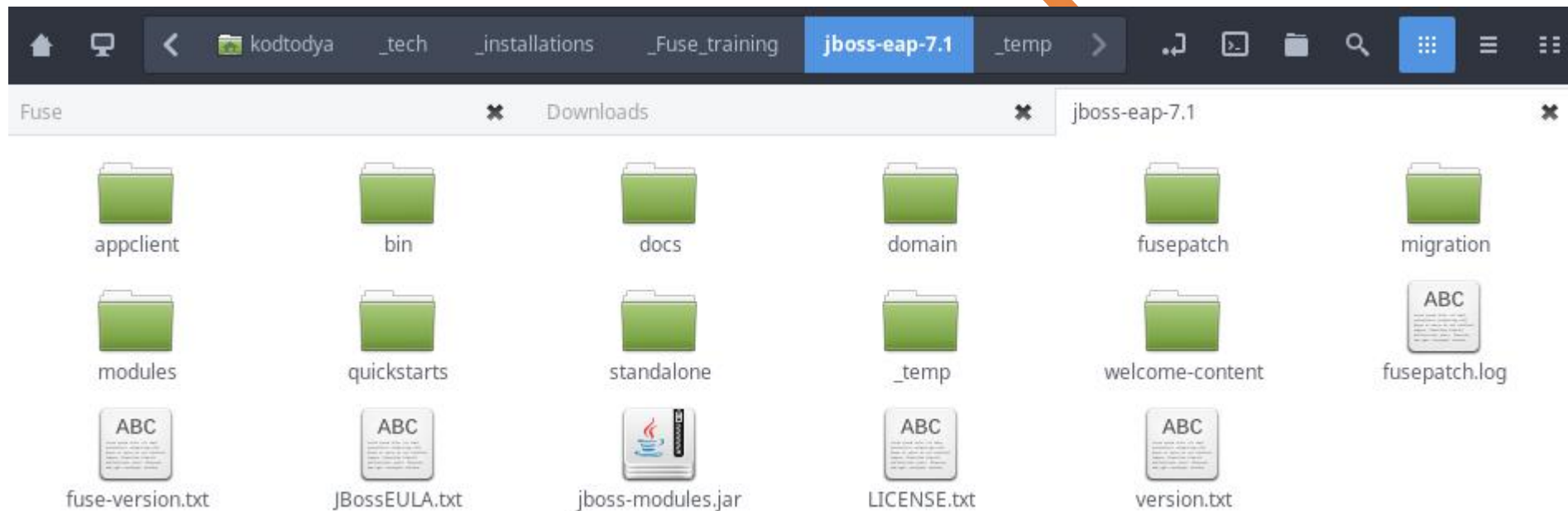
## Eco-systems





## Standalone Fuse(Karaf)





## Standalone Fuse on EAP





# Red Hat Fuse

## Roles of Sub-systems



- **Karaf** : OSGi container to run your application
- **Spring Boot** : Java framework to simplify bootstrapping and development of new Spring applications
- **Camel** : Java framework for integration





- As we know, Spring boot provides us embedded tomcat run-time container
- People are free to use whatever they are happy with; though our preferred approach to integration and micro-services is to use either Spring Boot or WildFly Swarm by default
- Spring boot is additional top-up to Spring framework that actually takes care of bootstrapping and development of new applications
- Spring BOOT makes development and testing easier" & "avoiding XML configurations



# Role of Apache Camel

- As we know, Apache Camel is java based integration framework and can be integrated with most of the well known Java frameworks
- Apache Camel offers us huge number of components to have connections with modern and legacy system
- Apache Camel comes with easy configuration of endpoint URIs that can help you to connect with more than 400 systems
- Camel is open-source and can be easily accommodated with any well known integration pattern and can be deployed as per any preferred approach of the user



# Let's revise Part-3

## ● Part – 3 : Core Concepts

- Core Concepts (To be covered in subsequent slides)
- Fuse flavors/Offerings
- Fuse Eco-system
- Fuse Sub-systems
- Fuse Architecture
- Role of Spring Boot
- Role of Apache Camel



# Questions ?





# Thank you..!!!

LinkedIn, GitHub, GitLab, Twitter: [@kodtodya](#)

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