Red Hat's Integration Roadmap

Jack Britton
Product Manager - Messaging

Keith Babo
Product Manager - Integration

THE WORLD IS CHANGING

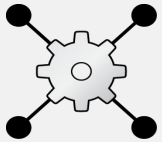
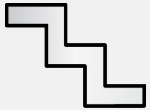
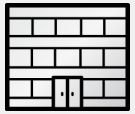
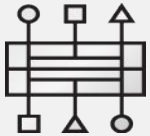
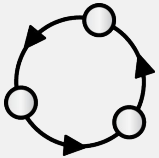
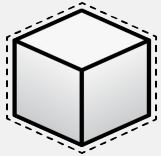
90%	80%	81%	62%	50 Billion
of all data created in last 2 years	of internet users use smartphones	of customers rely on social sites for purchasing advice	of total workload is running in the cloud	devices connected to internet by 2020
Big data	Mobile	Social	Cloud	IoT

INTEGRATION CONTINUES TO EVOLVE

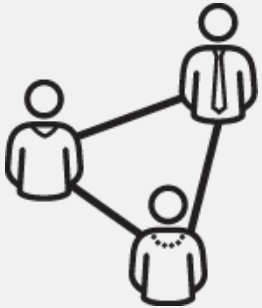
	1970	1980	1990	2000	2010
Technology	**Shared Mainframe Assets** ISAM VSAM IDMS	**Point to point** Proprietary TCP Sockets, FTP	**Hub and Spoke** Integration Brokers Messaging, CORBA	**Orchestration** SOA Web Services XML Open Source	**Service Mediation** REST JSON APIs
Business Problems	Data Consistency	Data Consistency Consolidated Reporting	Data Consistency Consolidated Reporting Electronic Ordering Case Management Straight Through Processing	All of those, ***plus*** Multi-step Processes Partner Managed Processes Composite Applications Mobile Applications	All of those, ***plus*** Migration to the Cloud APIs as a Business AI “Integration”

CHANGING LANDSCAPE FOR INTEGRATION

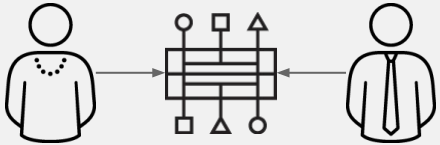
Enterprise IT is undergoing fundamental change. To remain competitive, businesses need an integration platform capable of supporting current ***and*** next generation architectures.

Service Endpoints	Architecture	Development Process	Deployment	Infrastructure
				
Webservices	Monolith	Waterfall	Server/VM	Data Center
				
APIs	Microservices	CI/CD	Container	Cloud

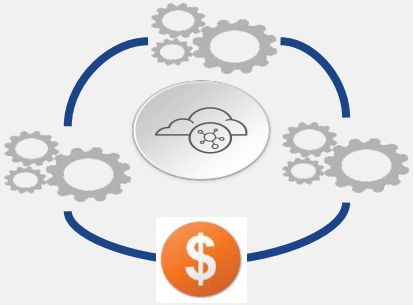
INTEGRATION IS EVERYWHERE



Integration is becoming pervasive within the enterprise with new actors involved (including non-technical)



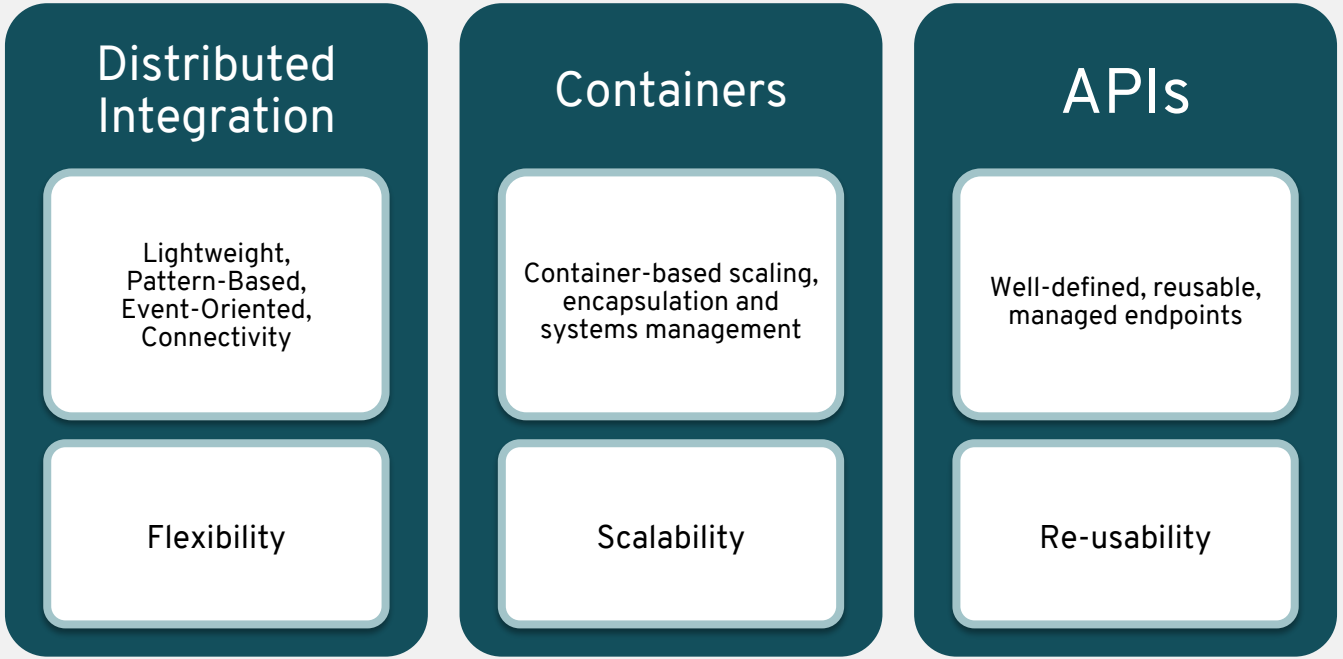
Integration now means not just “internal” connections but also “externally” (with customers and partners)



Integration is now central to business initiatives. Connecting it to revenue related projects

THREE PILLARS OF AGILE INTEGRATION

Key foundational capabilities needed by today's enterprises



TECHNOLOGY OF THE THREE PILLARS

Distributed integration

Lightweight, Pattern-Based, Event-Oriented, Connectivity

RED HAT® JBOSS® FUSE

RED HAT® JBOSS® A-MQ

Containers

Container-based scaling, encapsulation and systems management

APIs

Well-defined, reusable, managed endpoints

MESSAGING ROADMAP

Agenda

- AMQ Overview
  - Broker consolidation
  - Protocols and clients
  - AMQ 7 Broker features
  - AMQ Interconnect features
- Messaging as a Service introduction
- Roadmap timeline

What is AMQ 7?

- A suite of technologies to support a full range of common messaging patterns
  - Store and forward style queuing
  - Publish and subscribe
  - Direct, anycast, multicast, and request reply
  - Wide-area messaging networks
  - Elastic-scale cloud messaging: messaging as a service

- Open Standards support: AMQP 1.0 and MQTT

- Polyglot: Java/JMS. C++, .NET, Python, JavaScript (incl. Node.js)

Red Hat JBoss AMQ 7

Standard protocols

Common tooling

Flexible, standards-based messaging for the enterprise, cloud and Internet of Things

Broker

- New broker core with modern async architecture
- Improved performance and scalability

Interconnect

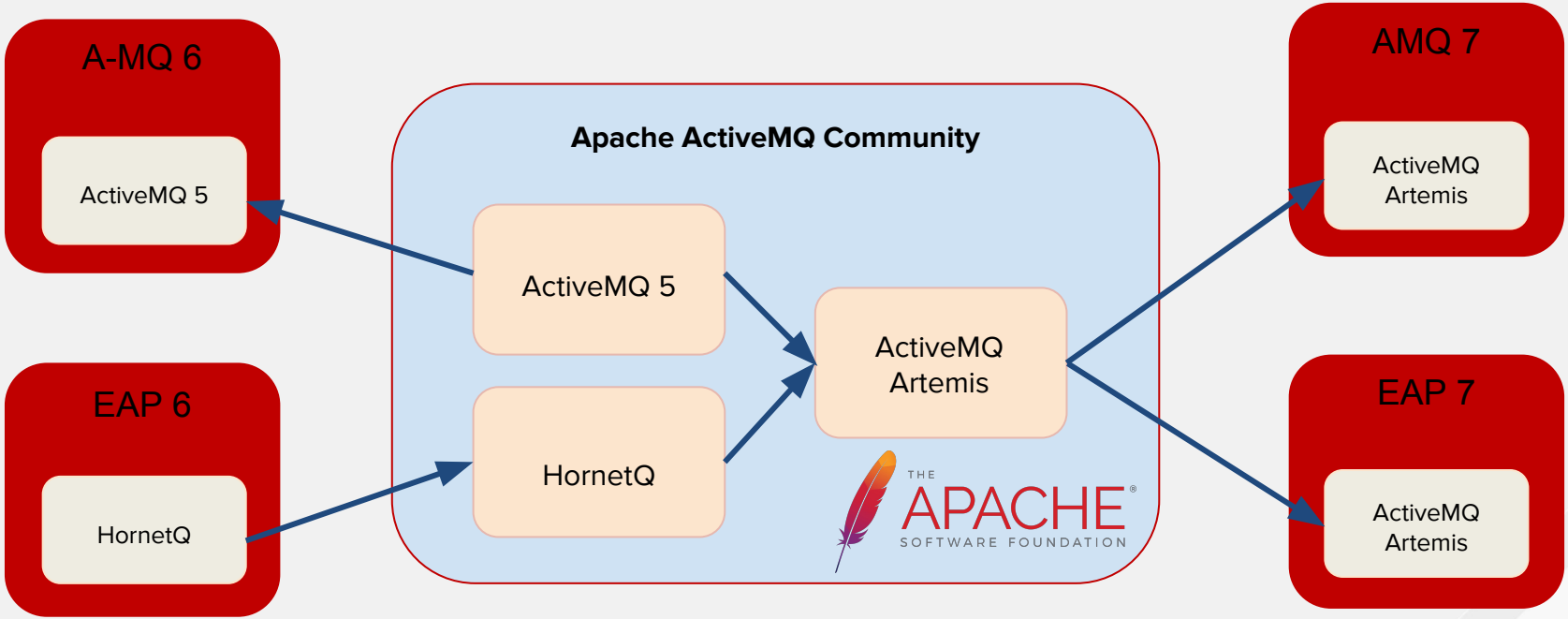
- Messaging Router
- High-performance direct messaging
- Distributed messaging backbone

Clients

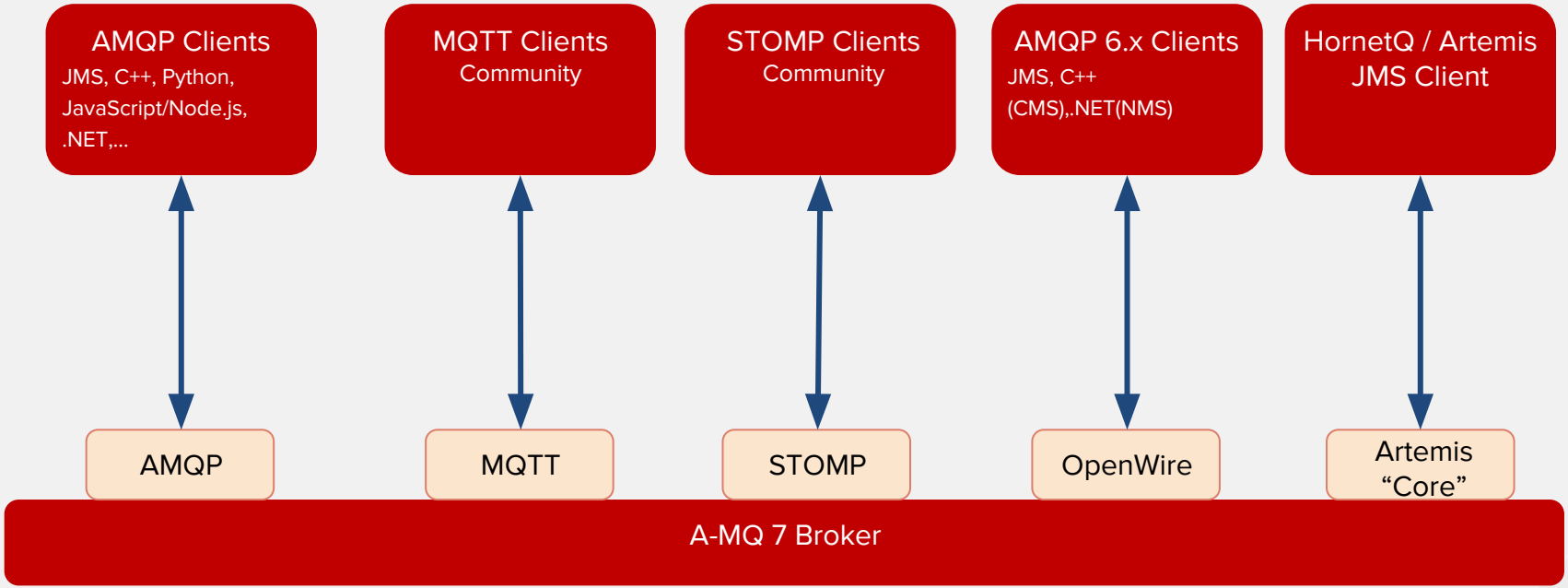
- New set of AMQP 1.0 clients
- Backward compatibility with A-MQ 6.x and HornetQ

AMQ 7 Broker

Broker Consolidation



Protocols and clients



AMQ 7 Broker vs. EAP Broker

Feature	A-MQ 7	EAP 7
High performance broker based on ActiveMQ Artemis	✔	✔
JMS 2.0 client library using “Core” protocol	✔	✔
AMQP 1.0 protocol support	✔	☐
MQTT protocol support	✔	☐
OpenWire protocol support	✔	☐

AMQ 7.0 Broker

New Features

- Faster throughput (completely non-blocking)
- JMS 2.0 support
- Shared-nothing HA
- Certified in AWS, GPE, and Azure (post GA)

Backward Compatibility

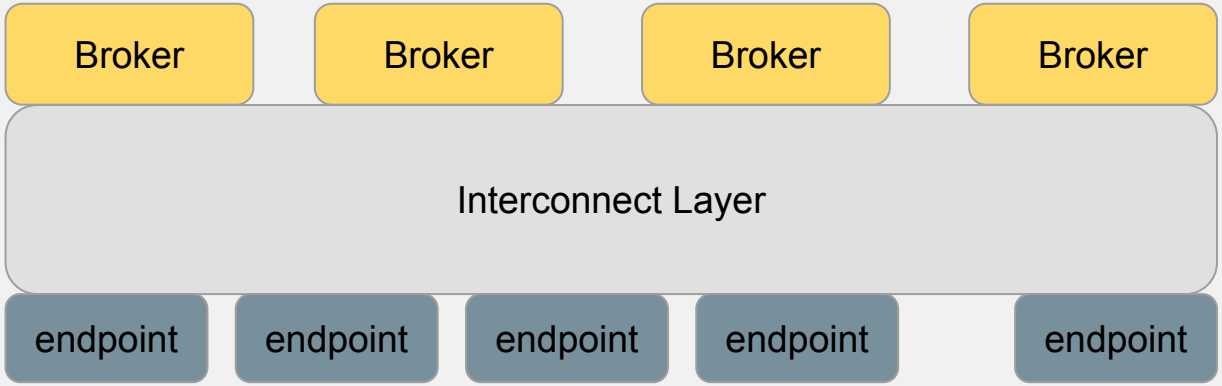
- Backward compatible with A-MQ 6.3
  - Openwire and AMQP
- Backward Compatible with HornetQ
  - Hornet Core clients
- Backward Compatible with MRG 3.2
  - AMQP 1.0 clients

AMQ 7 Interconnect

What is AMQ Interconnect?

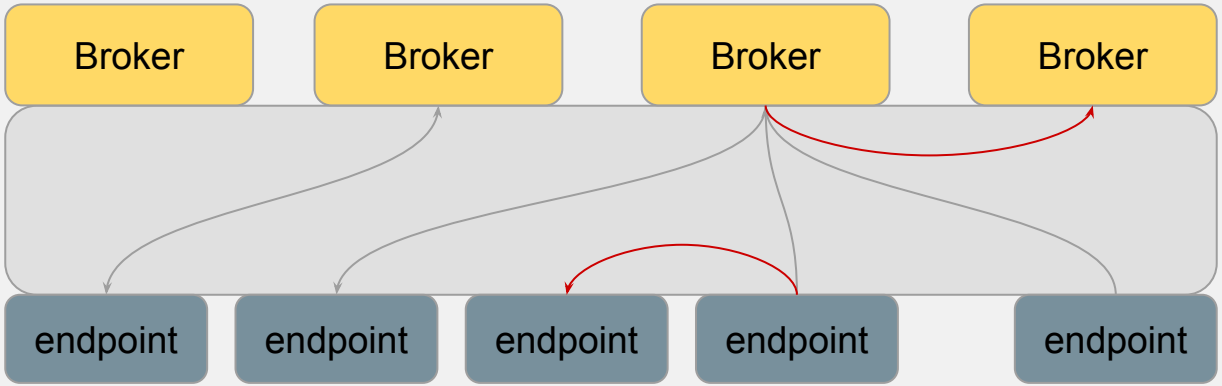
- A message router for the AMQP 1.0 protocol
- Separates messaging routing from message persistence
- Written in C and embedded python
- Built on the proton-C “engine” API
- Asynch architecture
- Routing protocol similar to OSPF (link-state routing)

AMQ Interconnect



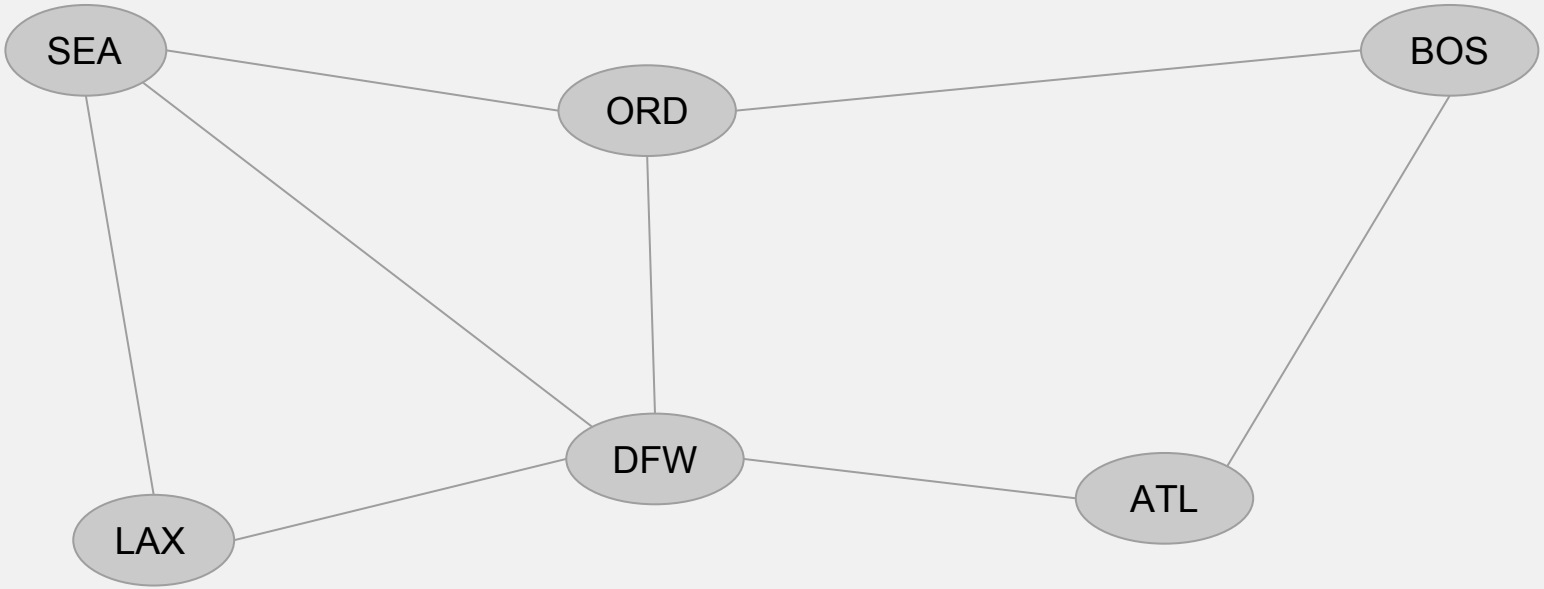
- Brokers are moved to the edge of the message bus (Interconnect layer)
- Endpoints make network connections to the Interconnect layer

AMQ Interconnect

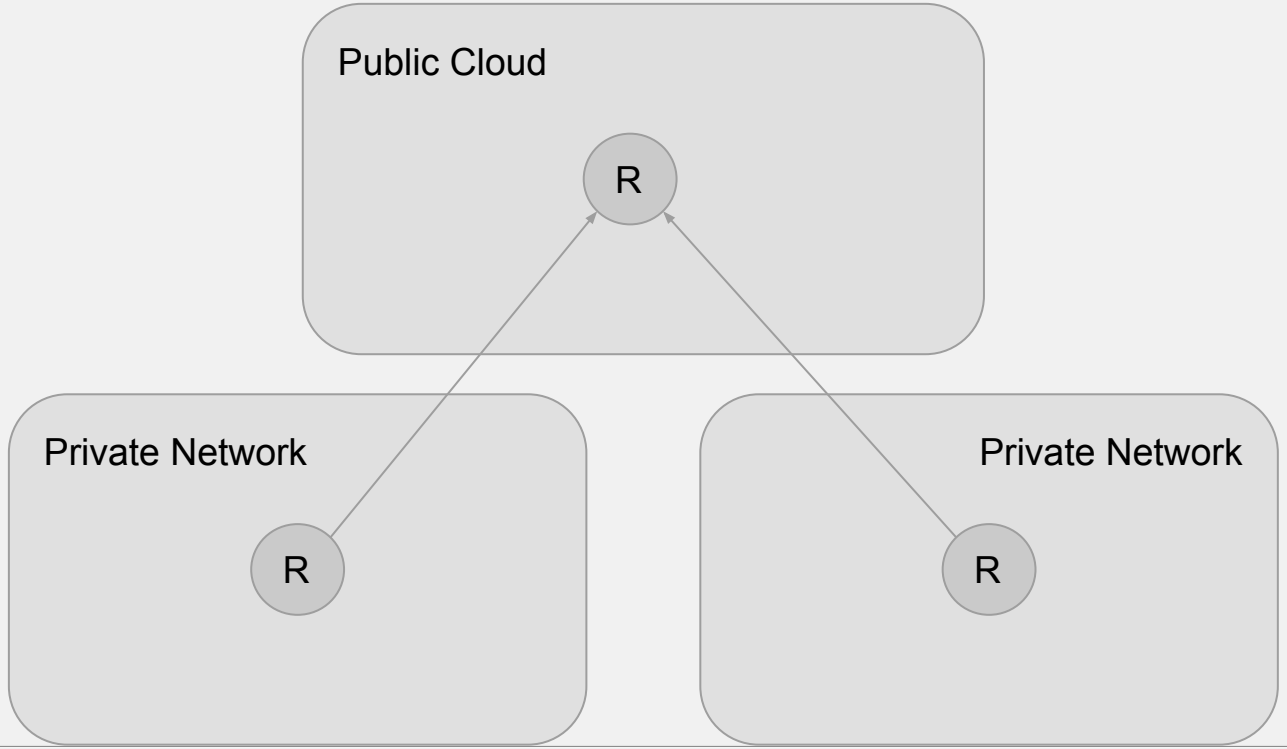


- At the AMQP level, endpoints may exchange messages with brokers
- And/or directly with other endpoints

AMQ Interconnect - Wide Area Message Bus



AMQ Interconnect - Wide Area Message Bus

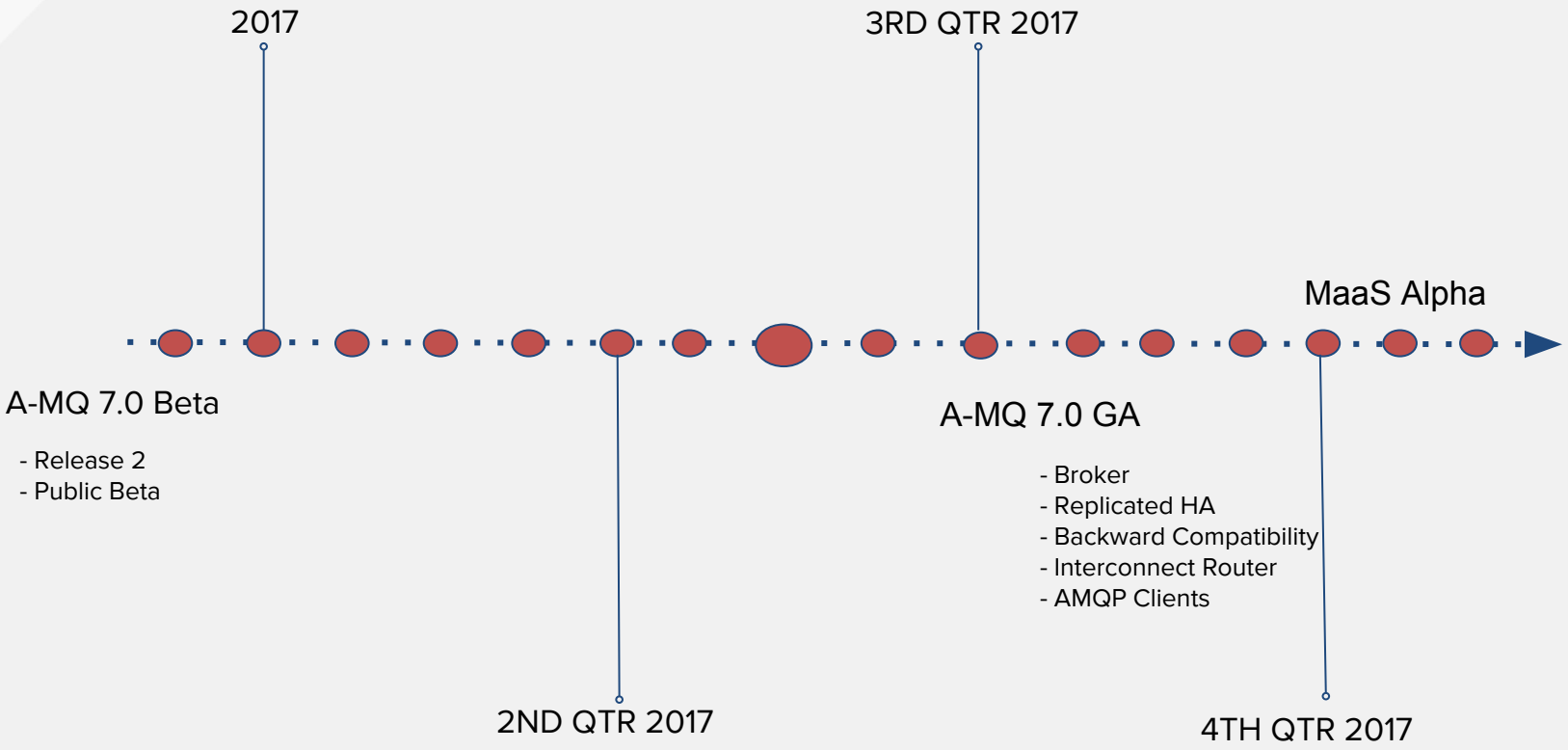


Messaging as a Service - Coming Soon

Messaging as a Service

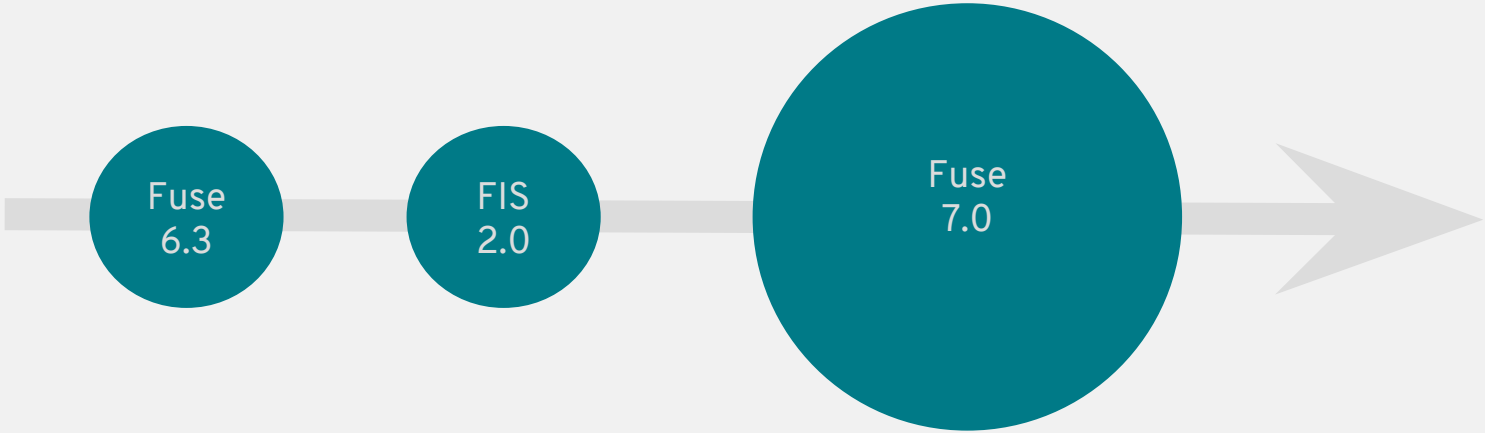
AMQ 7 built for OpenShift

- Enables applications deployed to an openshift cluster to include a scalable messaging component within them
- Expose service deployed on an openshift cluster to external applications
- Provide a service for communication between separate applications/projects on an openshift cluster
- Provide message fabric between OpenShift clusters on-prem and in the cloud


2017
3RD QTR 2017
MaaS Alpha
A-MQ 7.0 Beta
- Release 2
- Public Beta
A-MQ 7.0 GA
- Broker
- Replicated HA
- Backward Compatibility
- Interconnect Router
- AMQP Clients
2ND QTR 2017
4TH QTR 2017

INTEGRATION ROADMAP

WHAT HAVE WE BEEN UP TO ?



FUSE 6.3

FUSE 6.3 RELEASE GOALS

REFRESH

Camel 2.17
23 new Camel components
CXF 3.1
Jetty 9

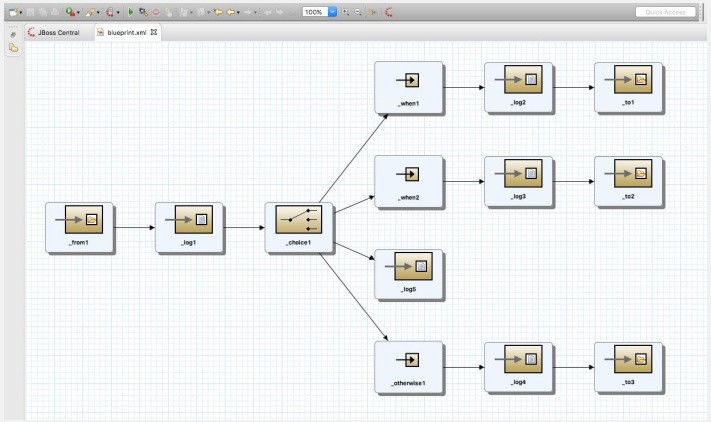
USABILITY

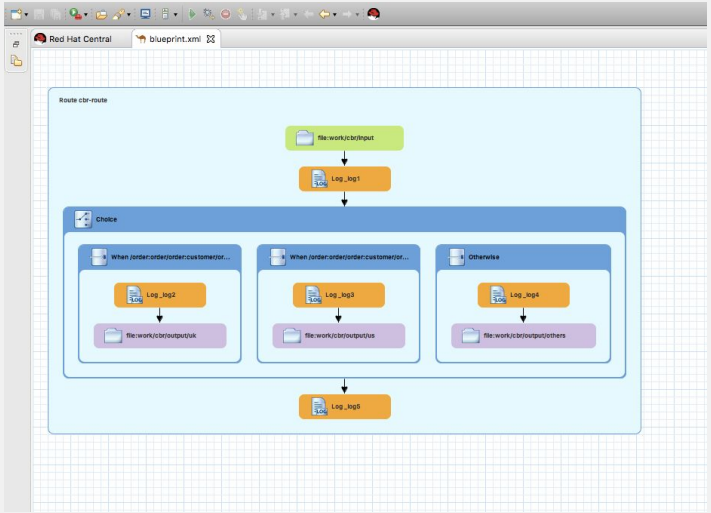
Revamped EIP Editor
New Getting Started Experience
Maven Repository Unification

STABILITY

2500 issues fixed across all projects
1191 issues fixed in product stream
204 issues fixed in Fabric v1

REVAMPED EIP EDITOR





FUSE LIFECYCLE UPDATES

Fuse 6.3 will be the long-term support stream for 6.x

Red Hat JBoss Fuse and Red Hat JBoss AMQ					
				Extended Life Support	
Family	GA	End of Full Support	End of Maintenance Support	End of ELS-1	End of ELS-2
6.x	Apr 2013	Jul 2018	Jan 2022	Jan 2024	Jan 2025

Fuse Integration Services

WHAT IS FUSE INTEGRATION SERVICES?

Fuse Integration Services (FIS) is a distribution of JBoss Fuse that provides tooling and runtime support for creating containerized integration services on OpenShift, including

- Docker-formatted container images
- Tooling to create, develop and build containerized Fuse applications
- Self-service deployment templates for common integration scenarios
- Native integration with Kubernetes for service discovery, clustering, and configuration management
- All based on the core technologies available in JBoss Fuse

LIGHTWEIGHT INTEGRATION RUNTIME

Docker-formatted container images for Karaf and Spring Boot provide the foundation for a built-for-purpose containerized integration runtime.

SPRING BOOT RUNTIME

`fis-java-openshift`

- Convention over configuration, bean-driven container
- Supports Spring and Java DSL for Apache Camel
- Autowired configuration
- Based on Camel 2.18, ideal for development of lightweight integration microservices

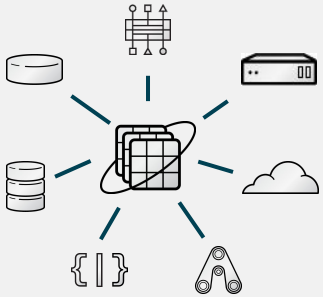
KARAF RUNTIME

`fis-karaf-openshift`

- Karaf-based OSGi container
- Supports Blueprint for Apache Camel
- Version-aligned to Fuse 6.3 to ease transition from standalone/Fabric-based Fuse deployments to OpenShift

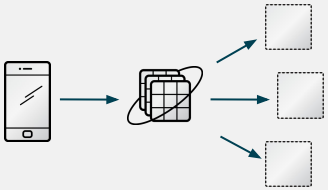
MULTIPLE INTEGRATION STYLES

A single technology stack to satisfy traditional and next generation integration requirements.



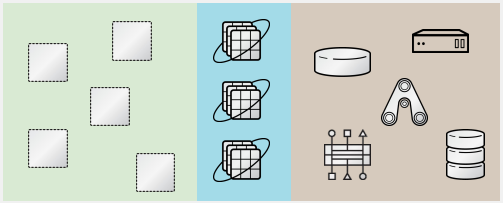
TRADITIONAL INTEGRATION

Pattern-oriented integration for on-premise and cloud-based resources.



INTEGRATION MICROSERVICES

Create and compose microservices using API and event-driven interactions.

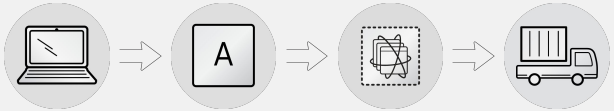


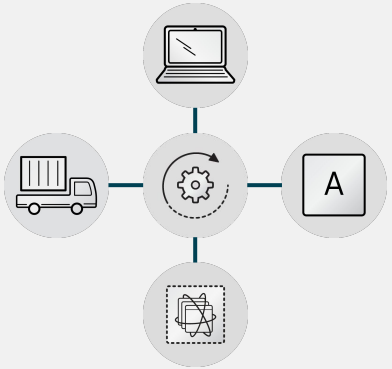
TRANSITIONAL INTEGRATION

Blend greenfield and brownfield to deliver next generation services.

CONTAINER-NATIVE TOOLCHAIN

Comprehensive tooling across all stages of delivery provides out-of-the-box support for continuous delivery with the flexibility to integrate with existing CI/CD solutions.





SUPPORT AT EACH STAGE OF DELIVERY

Develop, build, containerize, deploy

INCREASED AGILITY THROUGH CONTINUOUS DELIVERY

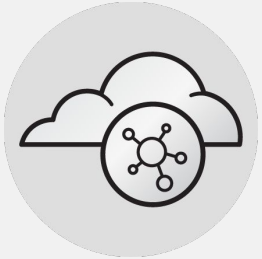
Automated delivery pipelines

HYBRID INTEGRATION PLATFORM

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

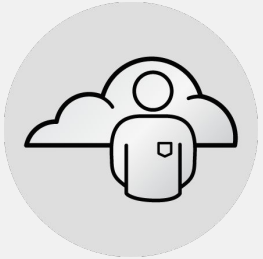
PRIVATE CLOUD

Deploy on-premise



PUBLIC CLOUD

Deploy on public cloud provider



MANAGED CLOUD

Deployed and managed by Red Hat

Fuse 7.0

FUSE 7.x

Three Fuse Product Initiatives

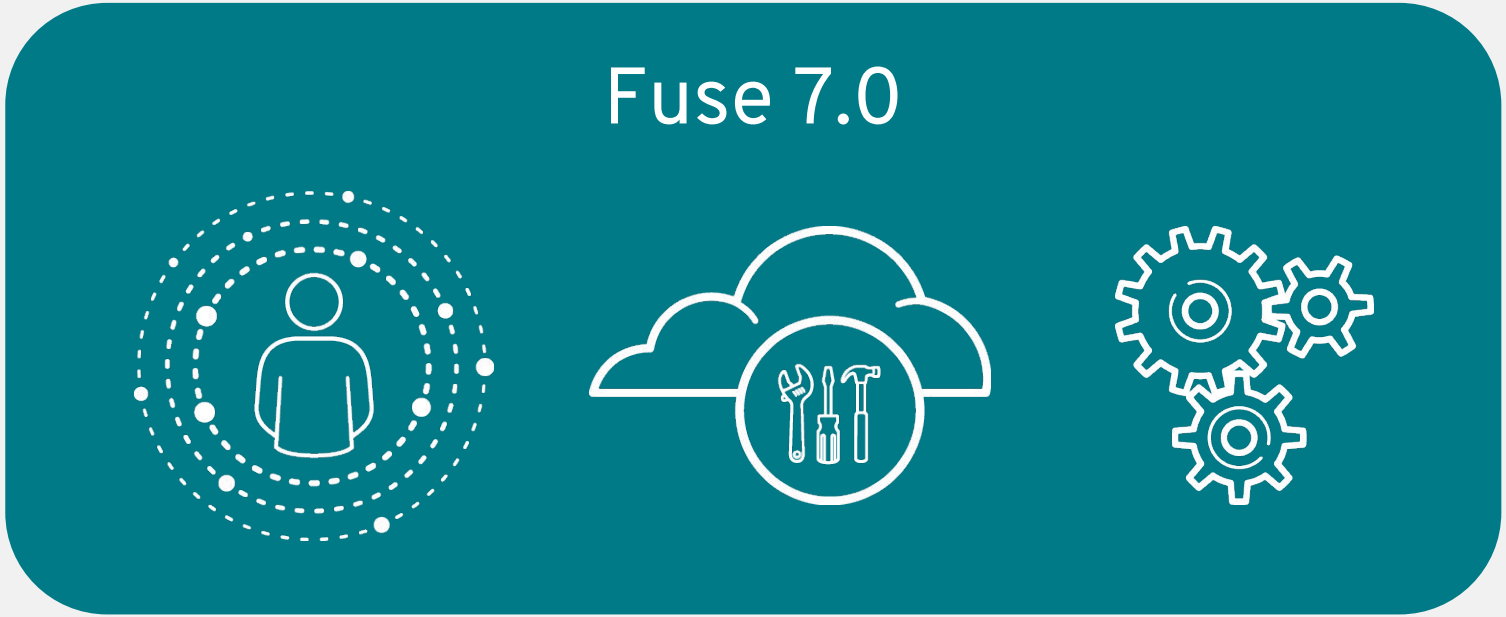
Fuse Standalone

Fuse Integration Services

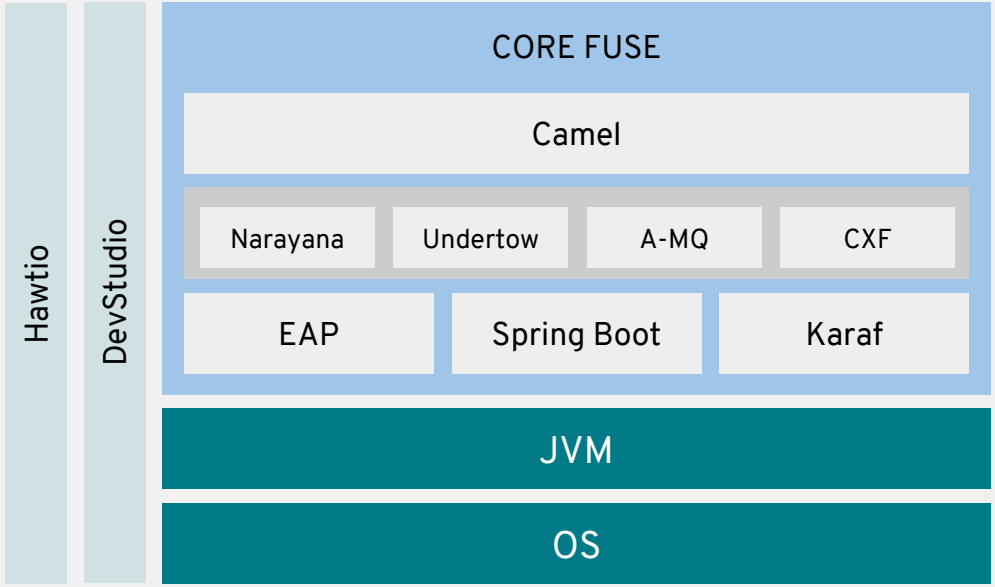
Fuse iPaaS

FUSE 7.x

One Comprehensive Product for Agile Integration



FUSE STANDALONE

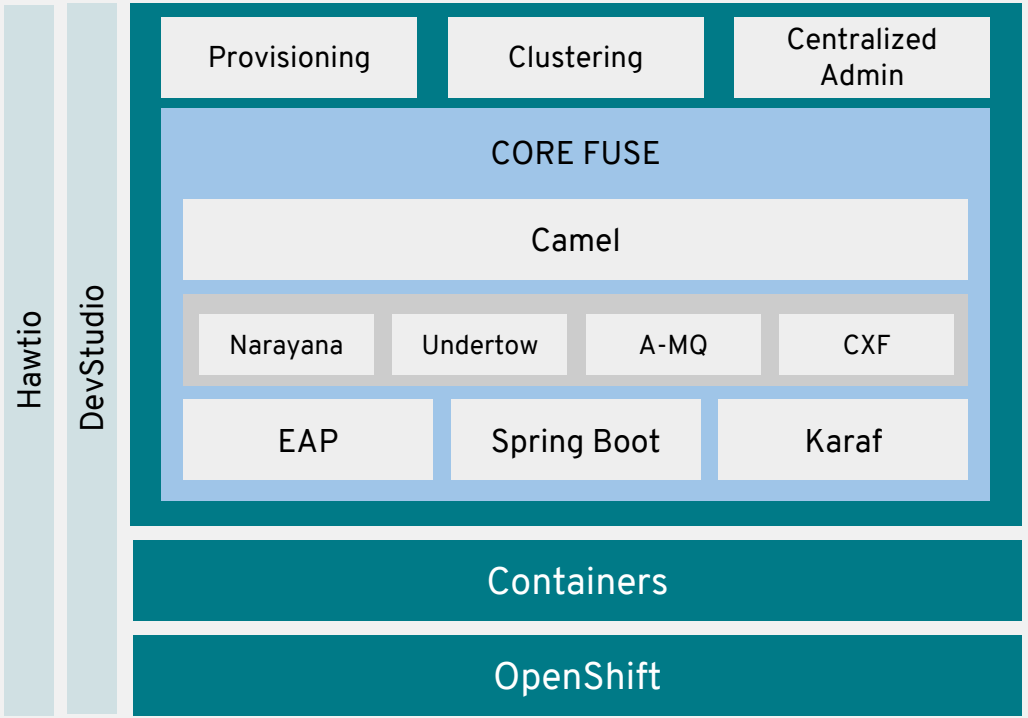


FUSE STANDALONE

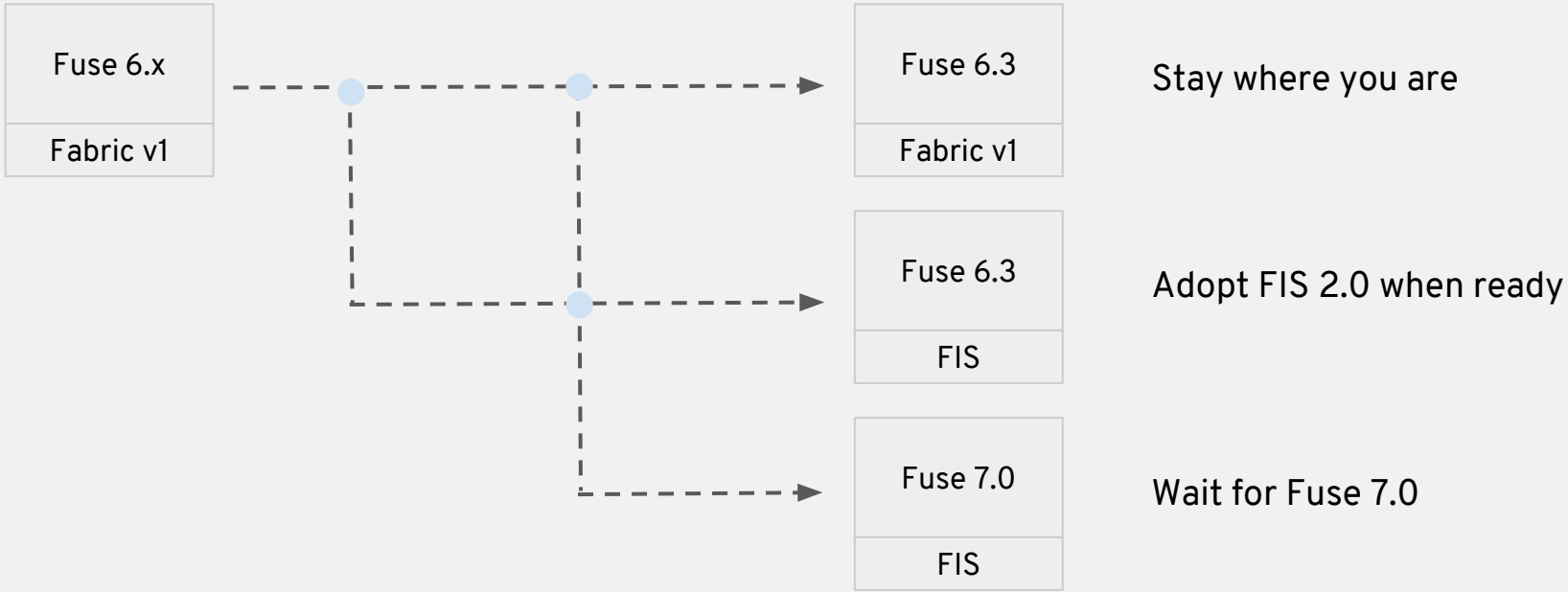
- Camel 2.20
- Karaf 4
- Add Spring Boot
- EAP 7.x Support
- Narayana transaction manager
  - Geronimo removed
- Undertow web container
  - Jetty deprecated

- AMQ 6/7 certification
- Further component modularization for EAP 7
- SwitchYard removed
- API-focused integration
  - REST DSL editor
  - SOAP <-> REST
  - 3scale integration

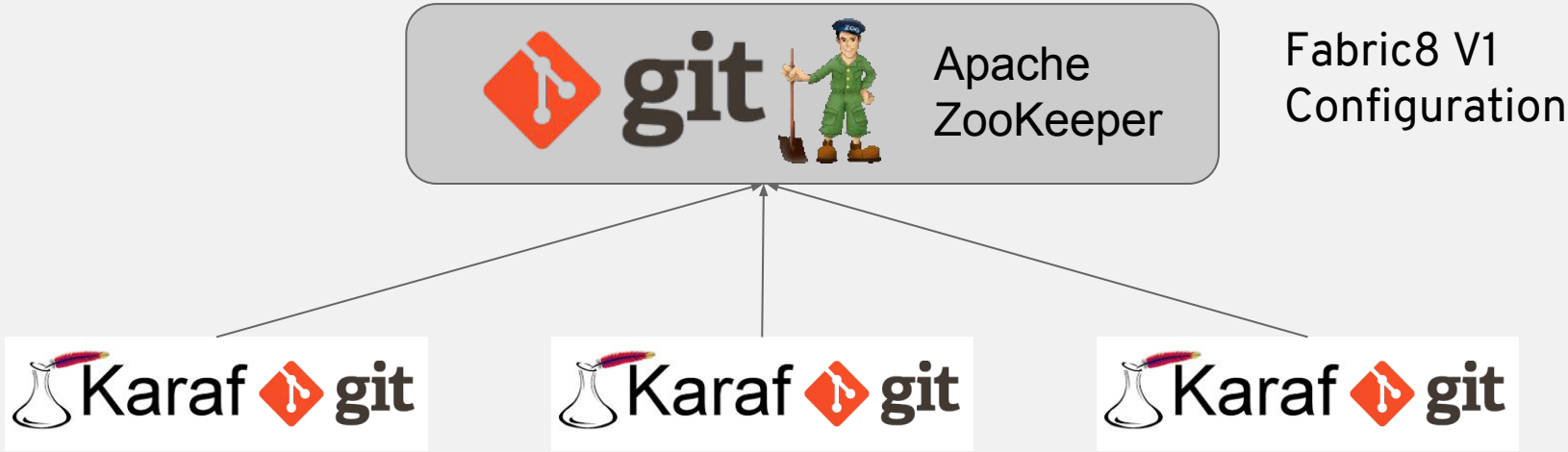
FUSE INTEGRATION SERVICES



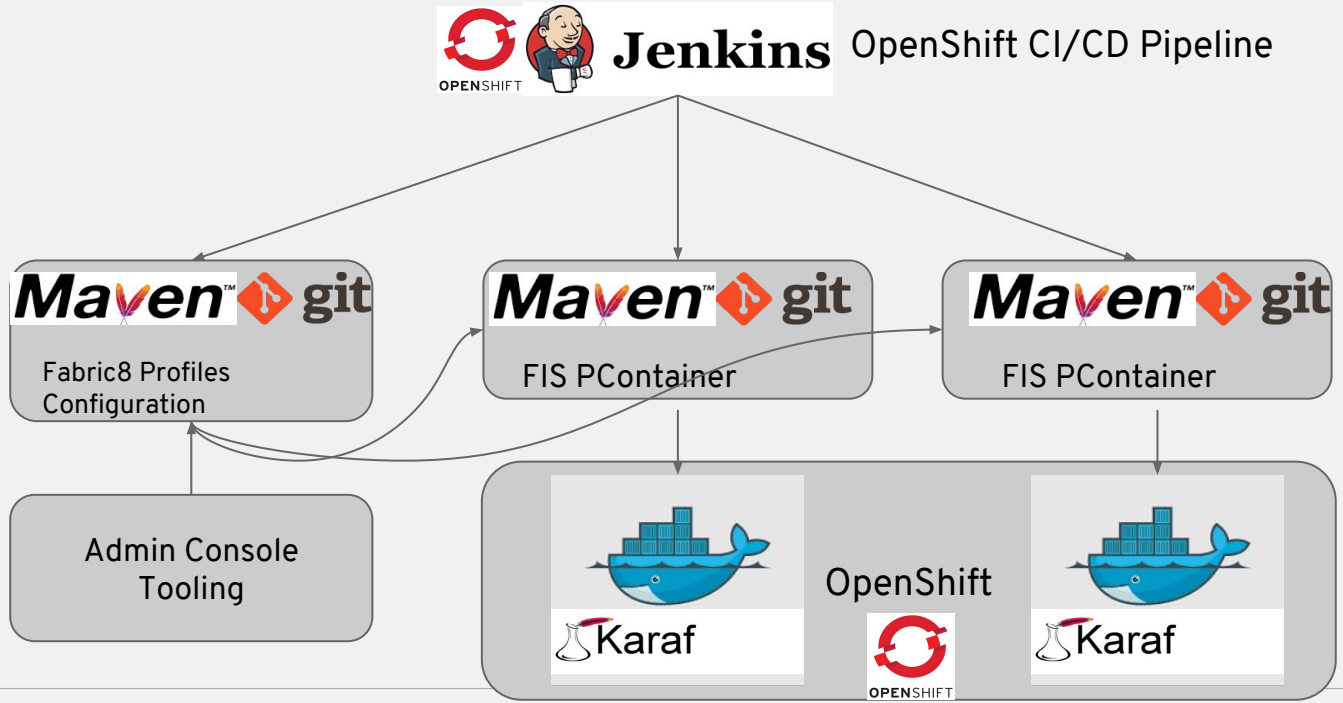
Fuse 6.x → Fuse 7.0



FABRIC v1 PROFILES



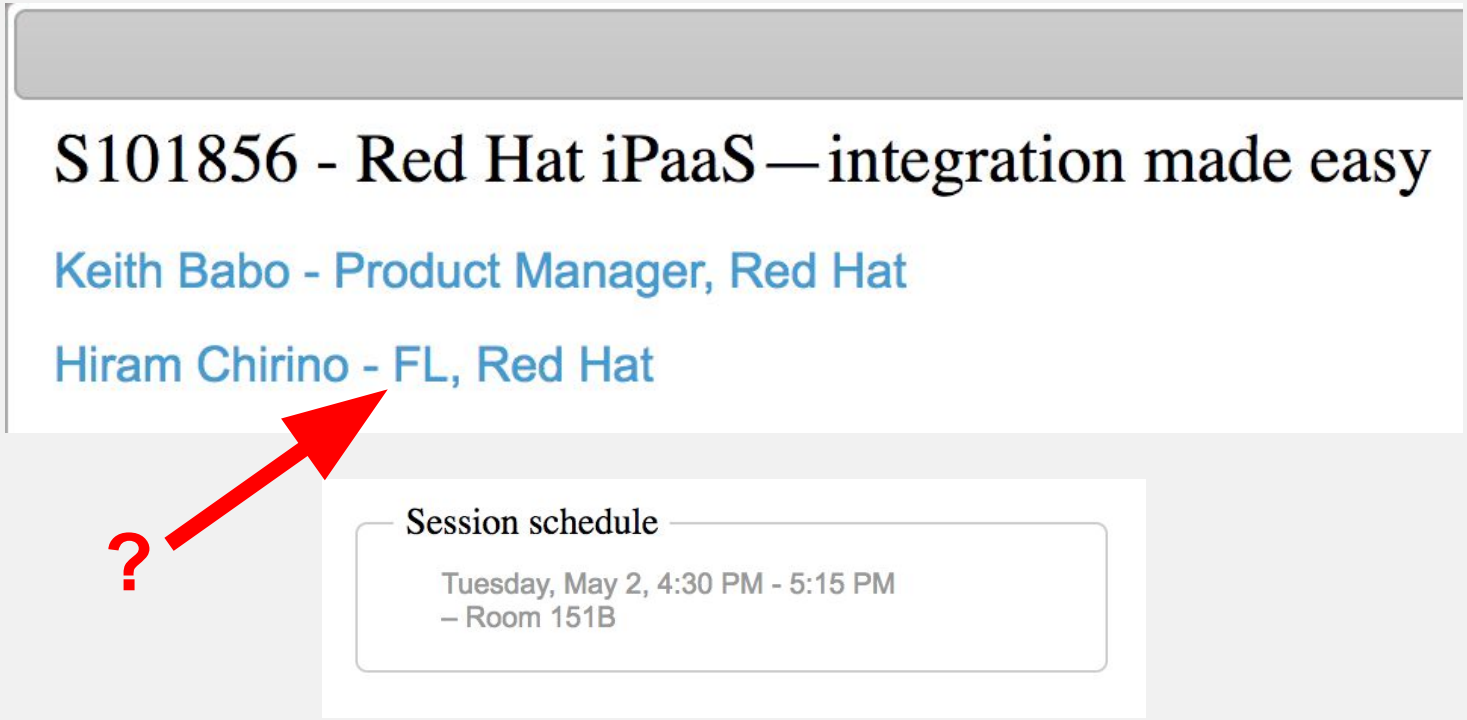
FUSE 7 PROFILES



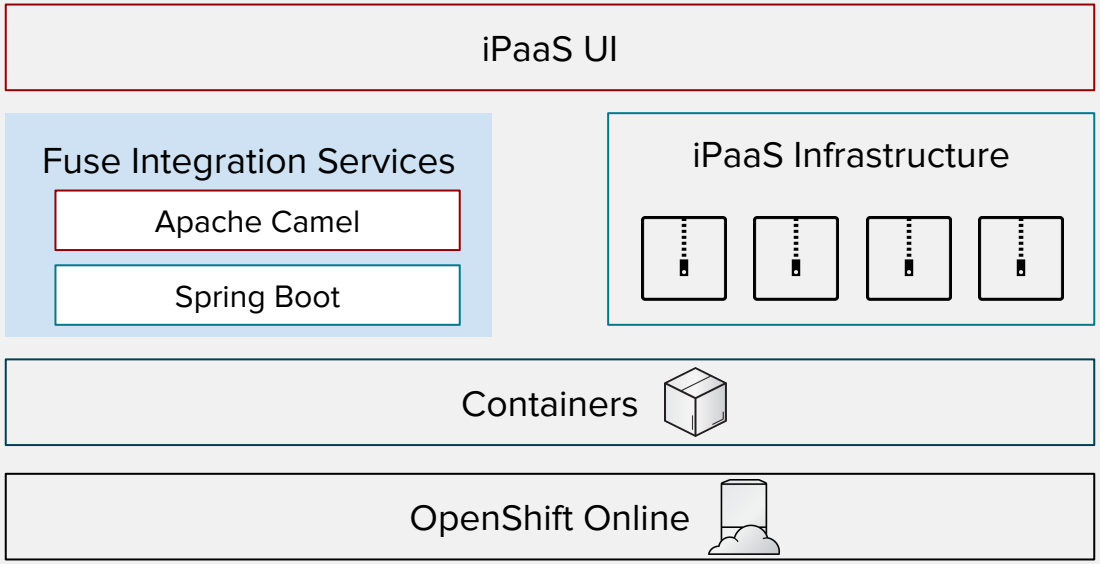
FUSE iPaaS

- No local install footprint
- 100% web-based interface
- Designed for everyday users
- Cloud, SaaS, and API connectivity
- Comprehensive coverage of integration lifecycle
- Build on a truly *hybrid* integration platform

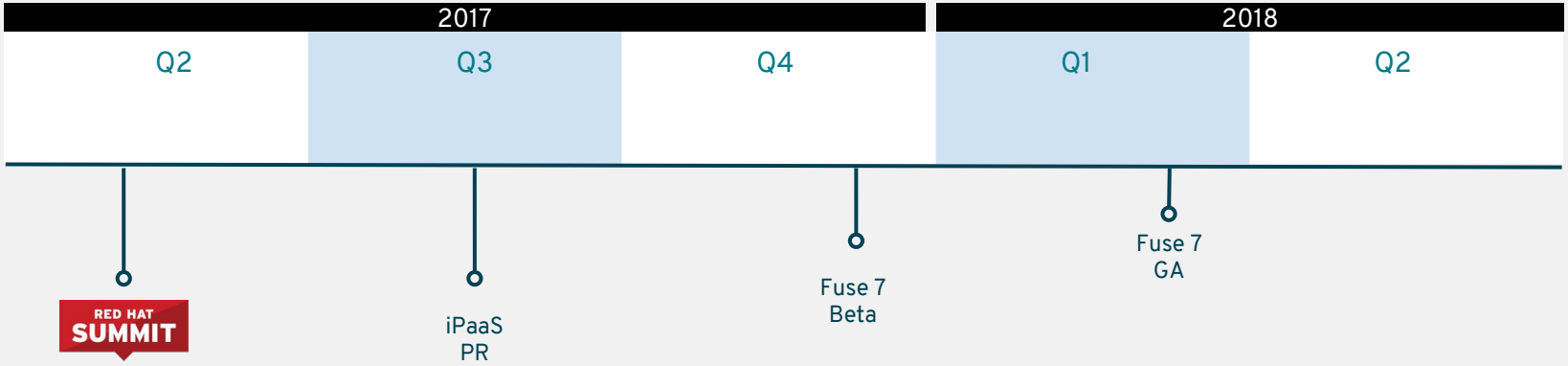
Integration Made Easy



Fuse iPaaS Architecture



Release Timeline



All roadmap dates are projections and subject to change.

UXD

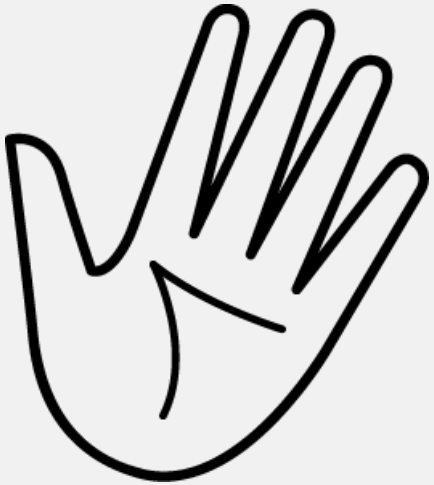
EXPERIENCE MATTERS

Interested in this topic?
Head over to the **User Experience Design** booth to learn more.

Partner Pavilion
Exhibit Hall A

Red Hat iPaaS Feedback

Dongni Wang



Questions?

RED HAT SUMMIT

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHatNews

youtube.com/user/RedHatVideos

LEARN. NETWORK.
EXPERIENCE
OPEN SOURCE.