RED HAT INTEGRATION ROADMAP - I

APIs, Events and Data: Your roadmap for Agile Integration with Red Hat

Keith Babo
Product Manager - Integration

David Ingham
Engineering Director - Integration

May 8, 2019
DOUBLE ROADMAP!!

PART I
APIs, events, and data—your roadmap for agile integration with Red Hat

Slides
Wednesday @ 10:30am
Room 160A

PART II
Best practices for developing modern applications with Red Hat Integration

Demos
Wednesday @ 11:30am
Room 157B
AGENDA

Modern application architectures
- Role of integration in modern applications
- Emerging trends

Red Hat Integration
- Introduction & Overview
- API Management with 3scale
- Flexible Messaging with AMQ
- Enterprise Integration with Fuse
- Data Integration with DV and CDC
THREE PILLARS OF AGILE INTEGRATION

Key foundational capabilities needed by today’s enterprises

DISTRIBUTED INTEGRATION

- Lightweight
- Pattern Based
- Event Oriented
- Community Sourced

FLEXIBILITY

MICROSERVICES CONTAINERS

- Cloud Native Solutions
- Lean Artifacts
- Individually Deployable
- Container Based Scaling and High Availability

SCALABILITY

APIs

- Well Defined
- Reusable
- Well Managed End-points
- Ecosystem Leverage

RE-USABILITY

#redhat #rhsummit
MODERN APPLICATION ARCHITECTURES
MODERN APP ARCHITECTURES

Ye Olde Architecture

Enlightened Postmodernism
MODERN APP ARCHITECTURES

- APIs
- Events
- Data
- Enterprise Integration Patterns
ELEMENTS OF MODERN APPS

- APIs
- Events
- EIPs
- Data
COMPOSITE USE CASES

Events

APIs

EIPs

Data

Data as APIs
API-Event Bridges
Change Data Capture
IoT
Legacy Integration (ACL)
Data Pipelines
Emerging Trends
"...an extension to Kubernetes exposing building blocks to build modern, source-centric, and container-based applications that can run anywhere".

**Build**
A pluggable model for building artifacts, like jar files, zips or containers from source code.

**Serving**
An event-driven model that serves the container with your application and can "scale to zero".

**Eventing**
Common infrastructure for consuming and producing events that will stimulate applications.
API MANAGEMENT and SERVICE MESH

API MANAGEMENT

- Business and people centric
- Resources are APIs and endpoints
- Subjects are apps and users
- Authentication & authorization
- Controls access and tracks usage

SERVICE MESH

- Infrastructure centric
- Resources are services, routes, pods
- Subjects are services and requests
- Authentication & authorization
- Controls access and tracks usage
ARE YOU DOING MULTICLOUD?

Multicloud is a cloud approach made up of more than 1 cloud service, from more than 1 cloud vendor—public or private.

https://www.redhat.com/en/topics/cloud-computing/what-is-multicloud
RED HAT INTEGRATION
INTEGRATION RELEASES

<table>
<thead>
<tr>
<th>CY 2018</th>
<th>CY 2019</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q4</td>
<td>Q3</td>
</tr>
<tr>
<td>Oct</td>
<td>TBD</td>
</tr>
<tr>
<td>Integration Q4</td>
<td>Integration Q3</td>
</tr>
<tr>
<td>Fuse 3scale AMQ</td>
<td>Fuse 3scale AMQ Data Virtualization Change Data Capture</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Q1</th>
<th>Q2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan</td>
<td>May</td>
</tr>
<tr>
<td>Integration Q1</td>
<td>Integration Q2</td>
</tr>
<tr>
<td>Fuse 3scale AMQ</td>
<td>Fuse 3scale AMQ Data Virtualization</td>
</tr>
</tbody>
</table>

TBD
Composite Use Cases

- End-to-end API lifecycle
- Red Hat Managed Integration
- Data as APIs
- Change Data Capture
- OpenShift 4.x certification

Operator-enabled platform
- Serverless Integration
- OpenAPI 3.0
- Kafka-native Camel Connectors

Red Hat Integration Roadmap

APIs
- 3scale

EIPs
- Fuse

Events
- AMQ

Data
- Data Virtualization
- Debezium
API Management with 3scale
## API MANAGEMENT

<table>
<thead>
<tr>
<th>CONTROL</th>
<th>VISIBILITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Securing APIs</td>
<td>Developer onboarding and engagement</td>
</tr>
<tr>
<td>Traffic flow control via policy</td>
<td>Traffic and policy alerts</td>
</tr>
<tr>
<td>Policy extensibility</td>
<td>Use analytics to understand how APIs are tracking against business objectives</td>
</tr>
<tr>
<td></td>
<td>FLEXIBILITY</td>
</tr>
<tr>
<td>Centralized management and distributed enforcement</td>
<td>API management architecture must span multiple environments</td>
</tr>
</tbody>
</table>
API MANAGEMENT AS CODE

ARTIFACTS
- OpenAPI Spec
- Implementation code
- Infrastructure code
- Mock service metadata
- Service metadata by environment
- Integration config
- API Management config
- API Gateway config
- Deployment config
- Service catalog / Developer portal

BENEFITS
- Version artifacts independently and as an integrated whole
- Deployment as a logical unit to eliminate sequential steps
- Immutable – roll back is a redeploy
- Standard governance and change/config process easily applied
- Full end-to-end automation for agility
API MANAGEMENT ROADMAP THEMES

- Extend API Policy coverage
- API Lifecycle Management
- Open Source
- Microservices
- API Management as Code
- Containers, Kubernetes, OpenShift
3scale ROADMAP

Last ‘6’

- Integration UX alignment
- API / Service Discovery
- 3scale Operator (TP)
- Istio Mixer Adapter (TP)
- APIcast policies: Routing, IP check, TLS client validation
- CLI Toolbox

Next ‘6’

- OpenShift 4.1 certification
- 3scale Operator (GA)
- APIs as a product
- Full end-to-end API Lifecycle automation
- Policy extension discovery
- APIcast policies: retry, upstream connection
- Istio Mixer Adapter (GA)

All roadmap dates are projections and subject to change.
DOUBLE CLICK ON 3scale

Wed, May 8, 3:30 p.m. Marius Bogoevici, Kavitha Srinivasan
   API management for asynchronous APIs, streaming APIs, and event-driven architectures

May 8, 4:00 p.m. Claus Ibsen, Mary Cochran, Dusty Humphries
   Learning to use the Camel Rest DSL with 3scale and OpenShift

Thu, May 9, 11:00 a.m. Monica Hockelberg, David Codelli
   API strategy demystified: Creating an API program your developers will love

Thu, May 9, 12:45 p.m. Christina Lin, Simon Green
   A day in the life: A workshop on APIs and integration
Flexible Messaging with AMQ
Messaging ≠ Messaging ≠ Messaging

- Low-latency pub/sub
- Multi-cloud backbone
- Replayable streams
- Load levelling
- Load balancing
- IoT device connectivity
- Enterprise application integration
- Temporal decoupling
- Database change data capture
- Geo-aware routing
- Event sourcing
- Message-driven beans
- Event-driven microservices
- Long-term message storage
- Temporal decoupling
- Database change data capture
- Geo-aware routing
- Event sourcing
- Message-driven beans
- Event-driven microservices
- Long-term message storage
Red Hat AMQ

Flexible messaging for the enterprise, cloud and Internet of Things

**Standard Protocols**
- Queuing and pub/sub
- Rich feature set
- JMS 2.0 compliance
- Best-in-class perf
- Based on Apache ActiveMQ Artemis

**Interconnect**
- Message routing
- Secure messaging backbone for hybrid cloud
- Based on Apache Qpid Dispatch Router

**Streams**
- Durable pub/sub
- Replayable streams
- Highly scalable
- Based on Apache Kafka

**Polyglot Clients**

**Common Management**
- Scalable, “self-service” messaging-as-a-service utility based on OpenShift
- Available for self-managed and Red Hat-managed deployments
AMQ INTERCONNECT
HYBRID CLOUD EVENT MESH

AMQ Interconnect

• Overlay messaging network spanning private and public cloud for seamless inter-service communication

• Secure
  ○ Mutual TLS between routers with dedicated CA
  ○ No inbound TCP connections to private cloud

• Logical addressing, not hosts & ports

Use-cases

• Expose data securely from private to public cloud (APIs and Events)
• Topology-aware request routing
• Cloud bursting
• Fault-tolerance
AMQ STREAMS

- Enterprise distribution of Apache Kafka
- Simplified deployment and management on OpenShift
- Provides:
  - Container images for running Apache Kafka, Zookeeper, Connect and MirrorMaker
  - Kafka Consumer, Producer and Admin clients
  - Kafka Streams
  - Operators for managing and configuring Apache Kafka clusters, topics, and users
- Based on OSS project called Strimzi (strimzi.io)
KAFKA ON OPENSIGHT
WITH AMQ STREAMS

- Easy scalability
  - Running Kafka on bare metal has a high bar (ops competency, physical servers, scaling up/down, etc.)
- Automation
  - Configuration as code and automated ops via Operators
  - Tedious ops actions like rolling updates and software upgrades are greatly simplified
- High availability
  - Restoration of Kafka nodes by rescheduling pods in the event of failure
- Messaging use cases are often latency sensitive
  - Can provision cluster/topics as the same time as the application
DEPLOYING A CLUSTER

Kafka
Custom
Resource

Cluster
operator

Topic & User
operators

Zookeeper
cluster

Kafka
cluster

#redhat #rhsummit
UPDATING A CLUSTER

Kafka Custom Resource

Cluster operator

Topic & User operators

Zookeeper cluster

Kafka cluster

#redhat #rhsummit
AMQ ONLINE

- Elastic-scale messaging-as-a-service built using AMQ components to run on OpenShift
- Separates the developer’s use of messaging from the administration of the infrastructure
- Allows messaging to be created “on demand” by development teams
- Multi-tenant
- Scale from small applications to huge volumes
- Declaratively define application messaging requirements
- Operator model with configuration as code
- Based on OSS project called Enmasse (enmasse.io)
AMQ ONLINE

AMQ Online Console
Create an address
Select type & name it
Choose plan
View summary
Address is created
Namespace view

#redhat #rhsummit
## AMQ Roadmap

### Last ‘6’
- Major AMQP perf improvements across Broker and Clients
- Client configuration conveniences for container usage
- AMQ Interconnect
  - Edge Router
  - Prometheus metrics
  - Improved logging and tracing
- AMQ Streams GA
- AMQ Online GA

### Next ‘6’
- AMQ Broker
  - HA/DR reference architecture
  - OpenShift Operator
- AMQ Interconnect
  - OpenShift Operator
- AMQ Streams
  - Kafka 2.1.1
  - Prometheus, Grafana, Alerts
  - HTTP & AMQP gateways
- AMQ Online
  - Console improvements
  - Kafka address spaces

All roadmap dates are projections and subject to change.
DOUBLE CLICK ON AMQ

Wed, May 8, 3:30 p.m. Marius Bogoevici & Kavitha Srinivasan, Red Hat
API management for asynchronous APIs, streaming APIs, and event-driven architectures

Wed, May 8, 4:00 p.m. Hugo Guerrero, Christina Wei, et al., Red Hat
Next-gen technologies at scale: Building solutions to manage tomorrow’s workloads (LAB)

Wed, May 8, 4:30 p.m. Edson Yanaga & Clement Escoffier, Red Hat
Post-REST microservices and event-driven architecture

Thu, May 9, 11:00 a.m. Tom Bentley & Jakub Scholz, Red Hat
Data streaming with Apache Kafka using AMQ streams

Thu, May 9, 2:00 p.m. Joseph Butler & Richard Stroop, Red Hat
Where do messages go in my cloud? (BOF)
Enterprise Integration with Fuse
RED HAT FUSE

Flexibility to distribute enterprise integration where it’s needed

Fuse Standalone
- Single JVM Fuse
- Developer-focused
- Integration where you need it
- “Classic” integration

Fuse on OpenShift
- Scale out Fuse
- Replaces Fabric v1
- Developer-focused
- “Cloud native” integration

Fuse Online (iPaaS)
- Low/no-code UX
- 100% cloud-based
- Integration through a browser
- Fuse for the rest of us - “Ad Hoc” Integration
FUSE ONLINE

- Self-service
- API Design & Implementation
- Data Mapping
- Connectivity
- Intelligent Routing
- Extensibility
APACHE CAMEL K

- A platform for directly running integrations on OpenShift and Kubernetes
- Based on Operator SDK
- Apache-based, community-driven project
- A subproject of Apache Camel started on August 31st, 2018

https://github.com/apache/camel-k
1. Create an integration definition

```java
// Lookup every second the 'www.google.com' domain name and log the output
from("timer:dns?period=1s")
    .routeId("dns")
    .setHeader("dns.domain")
        .constant("www.google.com")
    .to("dns:ip")
    .to("log:dns");
```

2. Run it

```bash
$ kamel run integration.groovy
```
APACHE CAMEL K

Tailored for cloud-native development experience

Dev Environment

Remote Cloud

“Integration” Custom Resource

Camel K Operator

Running Pod

kamel CLI

Live updates!

Fast redeploy!
Less than 1 second!
INTEGRATION AND KNATIVE
**FUSE ROADMAP**

### Last ‘6’
- API Provider GA
- Split and Aggregate Steps in Fuse Online
- Public API for CI/CD in Fuse Online
- Operators for Fuse console and Apicurito
- Camel K as a runtime for Fuse Online (TP)
- Updated connectors: Google Calendar, Slack, FHIR, Kafka, Telegram, Timer, Concur, IRC, Kudu, Google Sheets, OData (TP), FHIR full schema (TP)
- Standalone Apicurio with generation

### Next ‘6’
- Spring Boot 2.x support
- Data Views (TP)
- Conditional Flows
- Standalone data mapper
- Atlasmap conditional mapping
- OpenShift 4.1 and L2 Operators
- Error handling support for API Provider
- Connectors for SQS, SNS, JIRA, Knative

*All roadmap dates are projections and subject to change.*
RED HAT FUSE

Flexibility to distribute enterprise integration where it’s needed

**Fuse Standalone**
- Single JVM Fuse
- Developer-focused
- Integration where you need it
- “Classic” integration

**Fuse on OpenShift**
- Scale out Fuse
- Replaces Fabric v1
- Developer-focused
- “Cloud native” integration

**Fuse Online (iPaaS)**
- Low/no-code UX
- 100% cloud-based
- Integration through a browser
- Fuse for the rest of us - “Ad Hoc” Integration
DOUBLE CLICK ON FUSE

Wed, May 8, 4:30 p.m.  Marius Bogoevici, Kavitha Srinivasan  
A blueprint for distributed integration from the world's largest financial network

Wed May 8, 3:30 p.m.  Alain Pham, Rachid Snoussi  
Agile integration escape room

Wed, May 8, 11:30 a.m.  Cyrille Sauvignac, Darius Matboo, Laurent Broudoux  
ATOS smart city data platform based on Red Hat technology

Thu, May 9, 11:00 a.m.  David Bericat, Erik Jacobs  
From cloud to edge: Why cloud-native application development matters in supporting IoT

Thu, May 9, 3:15 p.m.  Songuel Ballikaya, Bernd Semmerau, Janine Eichler, Jochen Cordes, Rainer Thom, Shrish Shrivastava  
Building a next-generation integration platform for Deutsche Telekom IT
Data Integration with Data Virtualization and Change Data Capture
DATA VIRTUALIZATION

- Core data federation and virtualization functions of Red Hat Data Virtualization
- Virtual databases deployed as container-native services within OpenShift
- Web-based environment for creating and managing data views
- OData access for data-driven APIs
- JDBC access for traditional clients
- Built-in integration with Fuse and 3scale for enterprise integration and API management
CHANGE DATA CAPTURE

- Change data capture (CDC) allows database changes (inserts, updates, and deletes) to be externalized as events.
- The event stream can be used for a variety of purposes including maintaining a cache, updating search indexes, updating UIs, and generating derived views etc.
Debezium

- Fully open-source Change Data Capture project
- Active community, led by Red Hat; see debezium.io
- Provides source connectors for popular databases
- Externalizes event stream to Apache Kafka topics

CDC in Red Hat Integration

- Debezium is being productized as part of the Red Hat Integration product
- Integrated with Apache Kafka using AMQ Streams
- A Developer Preview will be released soon with general availability (GA) planned for later this year
CDC WITH DEBEZIUM
DATA ROADMAP

Last ‘6’

Data virtualization technology preview
- ODBC/JDBC client support
- OData client support
- Open API generation for OData endpoints
- Data Sources: Relational, File, Excel, MongoDB, REST
- Discoverability of OData APIs via 3scale
- Prometheus-based metrics

Next ‘6’

- Data virtualization General Availability
- CDC with Debezium

All roadmap dates are projections and subject to change.
DOUBLE CLICK ON CDC

Thu, May 9, 11:00 a.m. Gunnar Morling, Red Hat
Change data streaming patterns for microservices with Debezium

#redhat #rhsummit
THINGS TO REMEMBER

1. Integration = APIs, Events, EIPs, and Data
2. Red Hat Integration brings these capabilities together into a unified offering
3. This talk was a high-level summary - check out the deep dive sessions for great coverage of individual components
4. Part 2 of this talk is really awesome!
THANK YOU

linkedin.com/company/Red-Hat

facebook.com/RedHatInc

youtube.com/user/RedHatVideos

twitter.com/RedHat