

RED HAT
SUMMIT

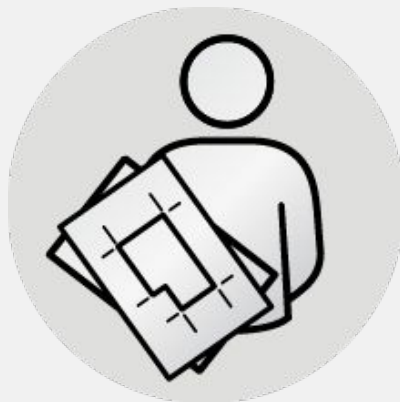
Red Hat Integration Platform as a Service

Integration Made Easy

Hiram Chirino
Platform Architect

Keith Babo
Product Manager

TODAY'S SPEAKERS



Hiram Chirino
Fuse Platform Architect
Apache Committer



Keith Babo
Fuse Product Manager
Reformed Engineer

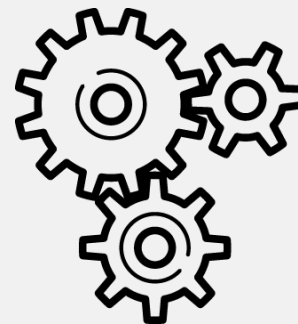
WHAT'S THIS IPAAS STUFF ABOUT?



User Engagement
Citizen Integrators



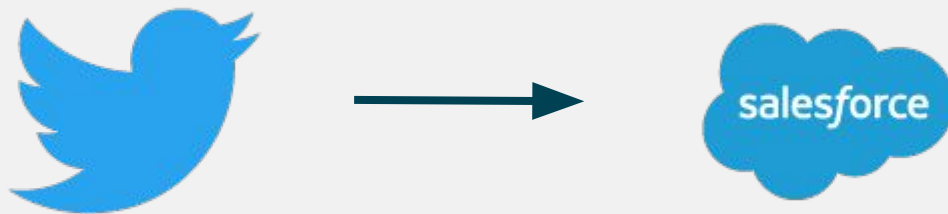
Cloud-Based
Public and Private Cloud



Core Integration
Connect, route, transform

FOR EXAMPLE ...

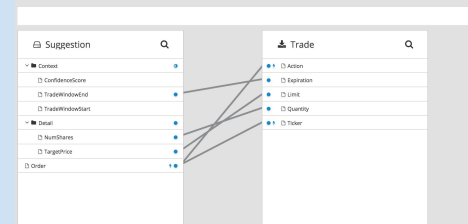
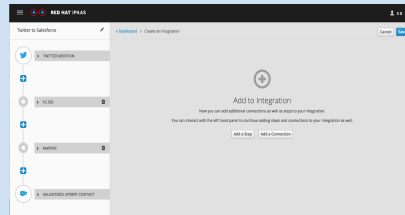
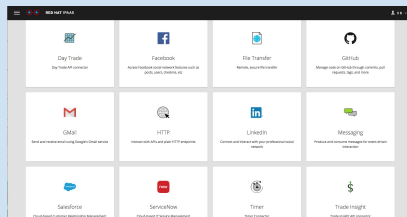
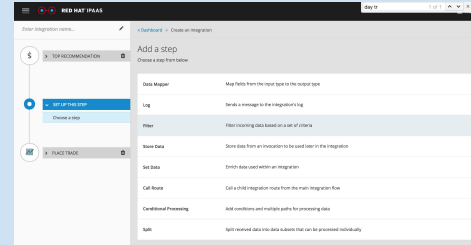
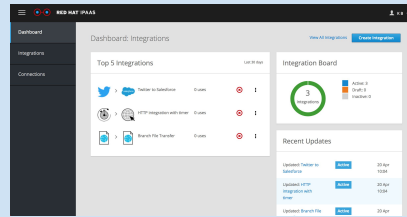
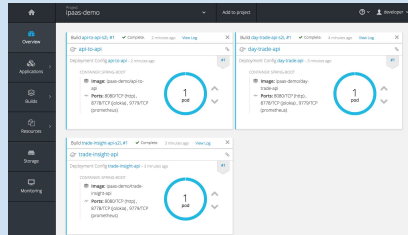
Goal : *Capture Twitter mentions related to a recent corporate event and create Salesforce contacts to follow up for sales opportunities.*



DEMO TIME!



LET'S BREAK THAT DOWN



BUT FIRST ...

FUSE 7.x

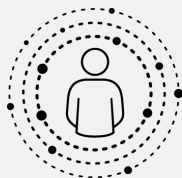
Three Fuse Product Initiatives



Fuse Standalone



Fuse Integration Services



Fuse iPaaS

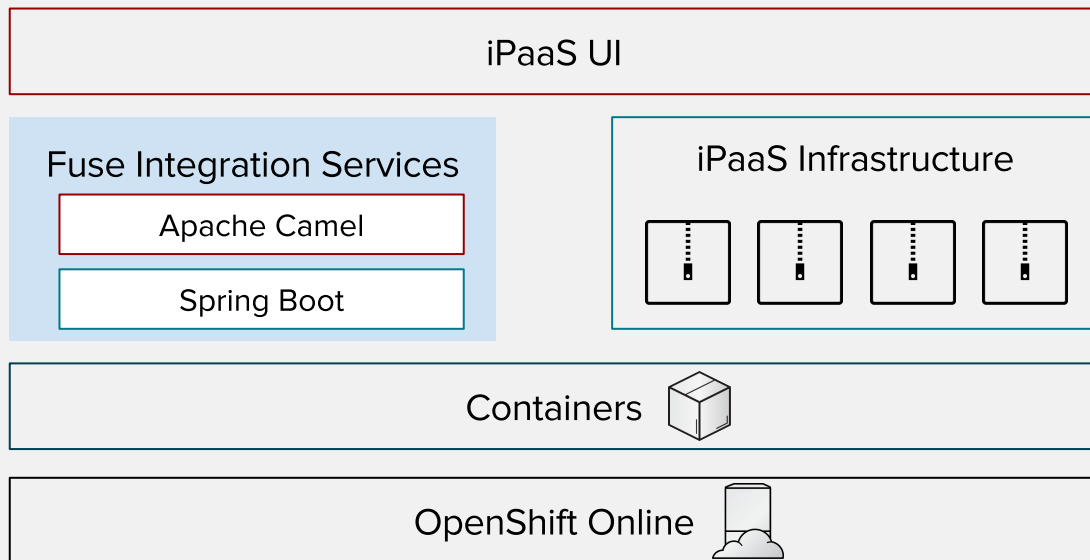
FUSE 7.x

One Comprehensive Product for Agile Integration

Fuse 7.0



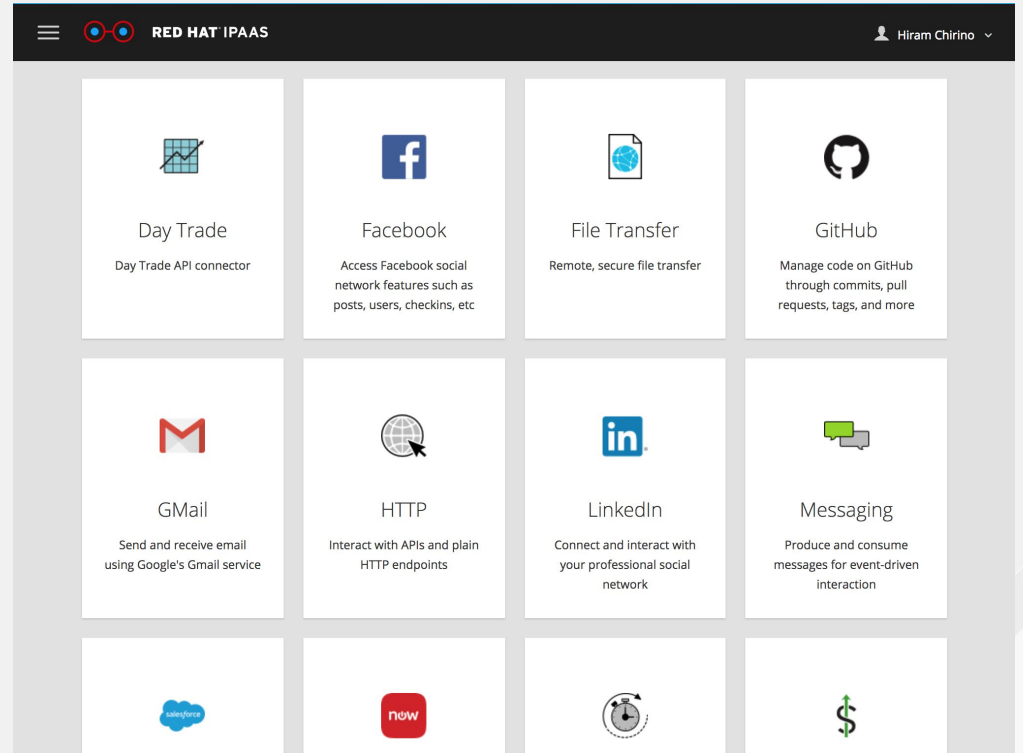
iPaaS Product Architecture



CONNECTORS

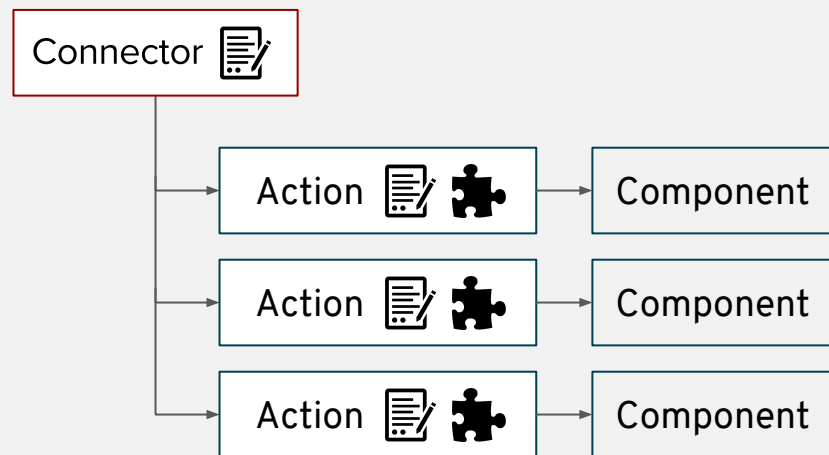
iPaaS Connectors

- Connectors are implemented with Apache Camel components under the covers
- Camel supports connecting to over 150 different Systems and or technologies



Connector Metadata

- Adds Metadata to Camel Components
- Defines General Connection Level Properties
- Multiple Actions:
 - Action Properties
 - Input/Output Data Shapes
- Simplifies Camel Connector Settings
 - Hides non-applicable properties
 - Per action defaults
 - Hard codes per action settings



iPaaS Connection


- Configures the general connection properties of a connector.
- Typically used to hold info like credentials
- Shared by Integrations
- Validation of connection settings is supported.

The screenshot shows the Red Hat iPaaS interface for creating a connection. The header includes the Red Hat logo and 'RED HAT IPAAS'. The user 'Hiram Chirino' is logged in. The breadcrumb trail is 'Home > Connections > Create Connection'. The wizard progress shows three steps: 'Select Connector' (1), 'Configure Connector' (2, currently active), and 'Name Connection' (3). The 'Configure Connector' step is titled 'Twitter Configuration' and contains four input fields: 'Access Token' (placeholder: 'The access token'), 'Access Token Secret' (placeholder: 'The access token secret'), 'Consumer Key' (placeholder: 'The consumer key'), and 'Consumer Secret' (placeholder: 'The consumer secret'). A 'Validate' button is located in the top right corner of the configuration box. Navigation buttons 'Cancel', '< Back', and 'Next >' are visible at the top right of the page.

Validation

Twitter Configuration

Validate

 Validation error: ✕

Could not authenticate you.

Access Token

Access Token Secret

Consumer Key

Consumer Secret

Validation

Twitter Configuration [Validate](#)

✔ **Twitter** has been successfully validated ✕

Access Token

Access Token Secret

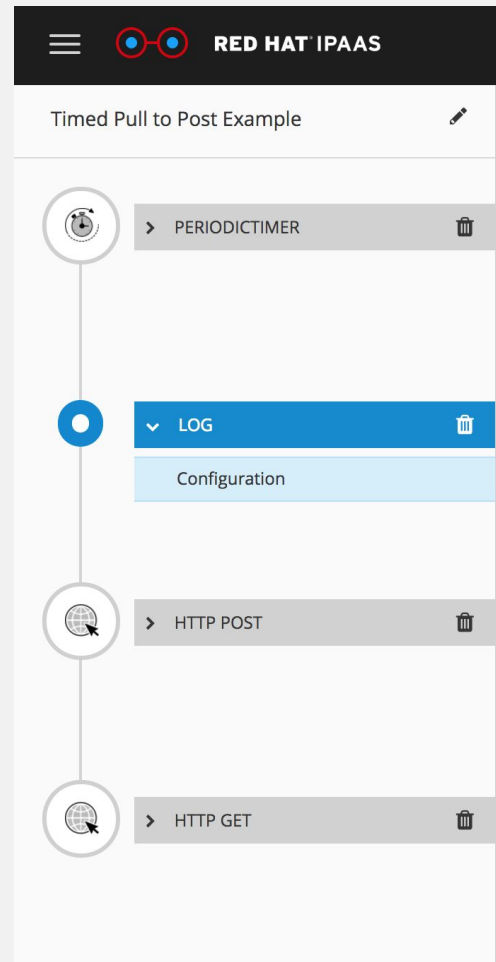
Consumer Key

Consumer Secret

INTEGRATIONS

What is an iPaaS Integration?

- A sequence of steps connect a source system with a target system.
- Use EIP style Steps between the systems
- Creates and Deploys a FIS style project for each Integration
- Can be steps (EIPs) or connections



The screenshot displays the Red Hat iPaaS interface. At the top, there is a navigation bar with a hamburger menu icon, the Red Hat logo, and the text "RED HAT IPAAS". Below this, the title "Timed Pull to Post Example" is visible, along with an edit icon. The main area shows a vertical sequence of four steps connected by a line. The second step, "LOG", is highlighted in blue and expanded to show a "Configuration" section. The other steps are "PERIODICTIMER", "HTTP POST", and "HTTP GET", each with a trash icon for deletion.

Step Icon	Step Name	Configuration	Actions
🕒	PERIODICTIMER		🗑️
🟦	LOG	Configuration	🗑️
🌐	HTTP POST		🗑️
🌐	HTTP GET		🗑️

Starting/Ending Connections

RED HAT IPAAS

Hiram Chirino





Enter integration name...

« Dashboard » Create an Integration

Choose a Connection

Choose a connection to start.

Name Name

- 
Citizen Babo's Twitt...
Citizen Babo's Twitter Account
- 
HTTP Example
HTTP Connection
- 
Keith's Salesforce
Keith Babo's Salesforce Instance
- 
Timer Example
Timer Connection

Actions

The screenshot displays the Red Hat IPAAS interface. At the top, there is a navigation bar with a hamburger menu, the Red Hat logo, and the text 'RED HAT IPAAS'. On the right side of the navigation bar, the user's name 'Hiram Chirino' is displayed with a dropdown arrow. Below the navigation bar, there is a search bar for integration names and a breadcrumb trail: '< Dashboard > Create an Integration'. The main content area is titled 'Choose an Action' and includes the instruction 'Choose an action for the integration.' and a search filter 'Filter by name...'. On the left side, there is a sidebar with a Twitter icon and a plus sign. The sidebar shows a dropdown menu for 'CITIZEN BABO'S TWITTER' with a 'Choose an action' option, and a 'FINISH' button. The main content area lists five actions:

Mention	Search for tweets that mention you
Search	Search tweets based one or more keywords
Send a Tweet	Creates a new tweet
Send a Direct Message	Direct message a specific user
Get My Timeline	Retrieve tweets from my timeline

Steps

Data Mapper	Map fields from the input type to the output type
Log	Sends a message to the integration's log
Filter	Filter incoming data based on a set of criteria
Store Data	Store data from an invocation to be used later in the integration
Set Data	Enrich data used within an integration
Call Route	Call a child integration route from the main integration flow
Conditional Processing	Add conditions and multiple paths for processing data
Split	Split received data into data subsets that can be processed individually

Child Integrations

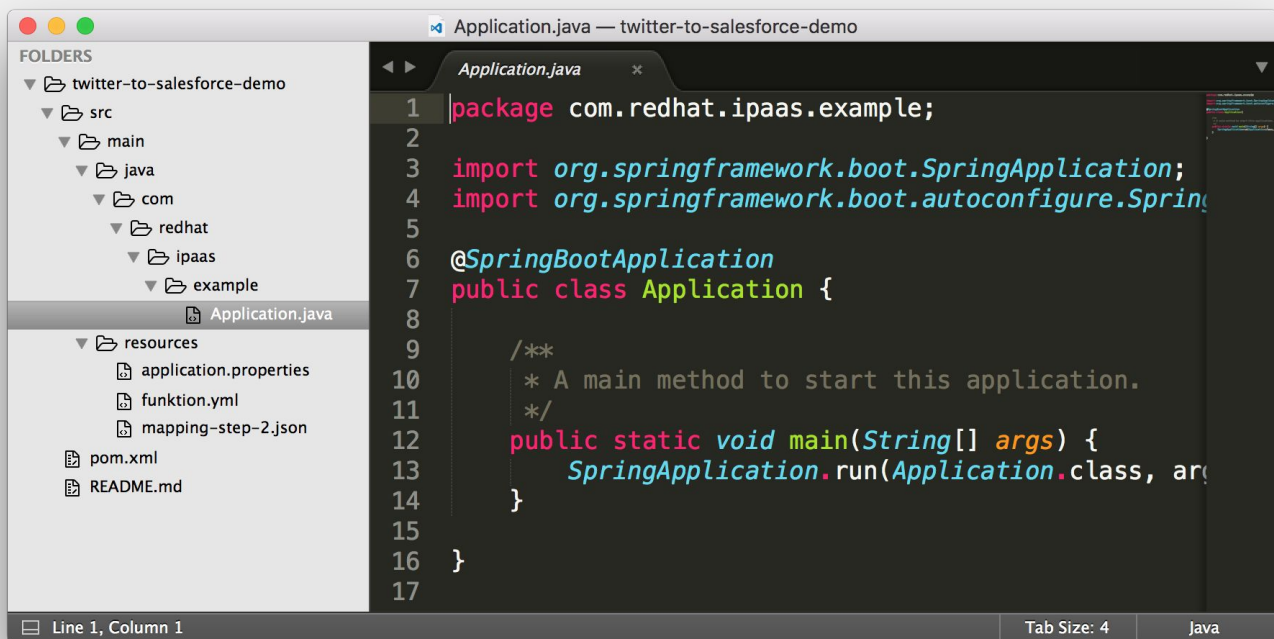
- Encapsulates more complex processing flows
- Partition integration logic
- Parent-child relationship

The screenshot displays the iPaaS interface for creating and configuring a child integration. The main workspace is divided into three sections:

- Left Panel (Create Integration):** Shows a vertical flow starting with a 'START' node (Facebook icon) labeled 'Facebook - photo'. This is followed by a 'Wire Tap 1' node (Twitter icon) labeled 'Child Integration 1'. The flow ends with a 'FINISH' node (Twitter icon) labeled 'Twitter - photo'.
- Middle Panel (Child Integration 1):** Shows a detailed view of the 'Child Integration 1' flow. It starts with a 'CHILD INTEGRATION START' node (Twitter icon) labeled 'Twitter message Connection' containing a 'Twitter message' step. This is followed by a 'Wire Tap 2' node (Twitter icon) labeled 'Child Integrations 2'. The flow ends with a 'CHILD INTEGRATION FINISH' node (Gear icon) labeled 'Set up this Connection'.
- Right Panel (Set up this Connection):** Contains a search bar and a grid of connection options. The options are categorized by platform and action:
 - Facebook:** 'Facebook - photo', 'Facebook - comment', and another 'Facebook - comment'.
 - Twitter:** 'Twitter - photo', 'Twitter - comment', and another 'Twitter - comment'.Each option includes a placeholder text: 'Ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo'. A 'Save + Continue' button is located at the bottom right of this panel.

Developer Perspective

What's in an iPaaS FIS Project?



The screenshot shows an IDE window titled "Application.java — twitter-to-salesforce-demo". The left sidebar displays the project structure under "FOLDERS":

- twitter-to-salesforce-demo
 - src
 - main
 - java
 - com
 - redhat
 - ipaas
 - example
 - Application.java

The main editor displays the following Java code:

```
1 package com.redhat.ipaas.example;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.Spring
5
6 @SpringBootApplication
7 public class Application {
8
9     /**
10      * A main method to start this application.
11      */
12     public static void main(String[] args) {
13         SpringApplication.run(Application.class, args);
14     }
15
16 }
17
```

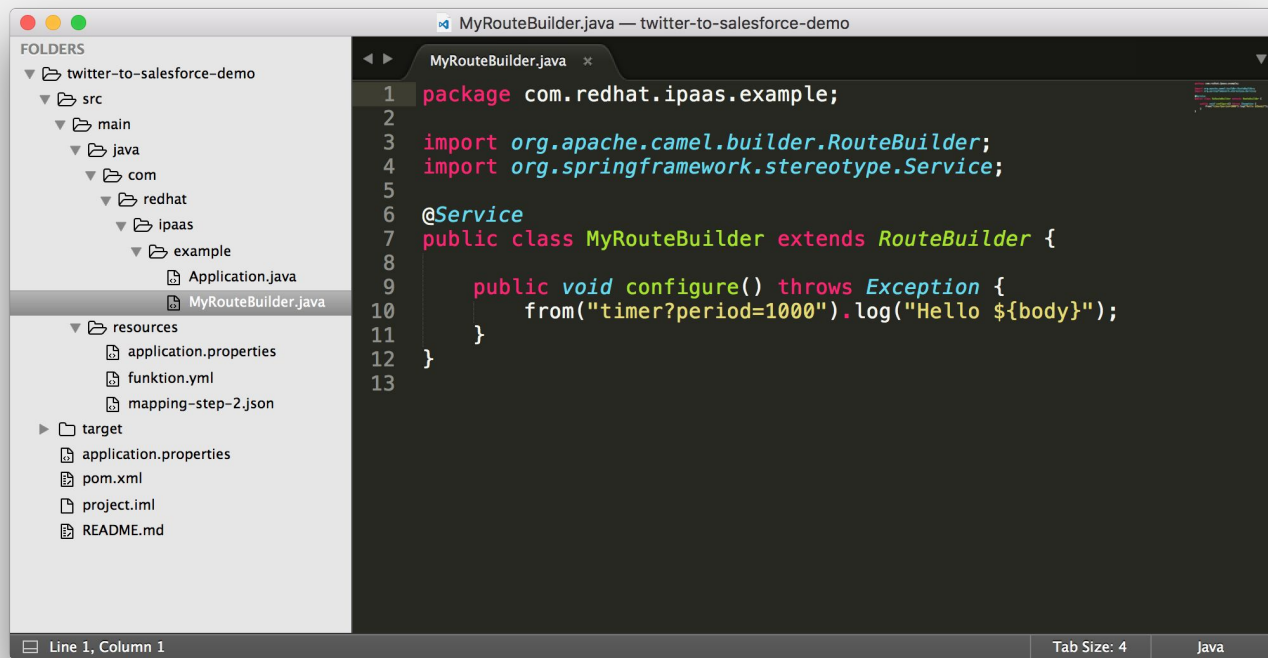
The status bar at the bottom indicates "Line 1, Column 1", "Tab Size: 4", and "Java".

function.yml

```
1 ---
2 flows:
3 - steps:
4   - kind: endpoint
5     uri: twitter-mention
6   - kind: endpoint
7     uri: atlas:mapping-step-2.json
8   - kind: endpoint
9     uri: salesforce-upsert-contact
10
```

Line 1, Column 1 Spaces: 2 YAML

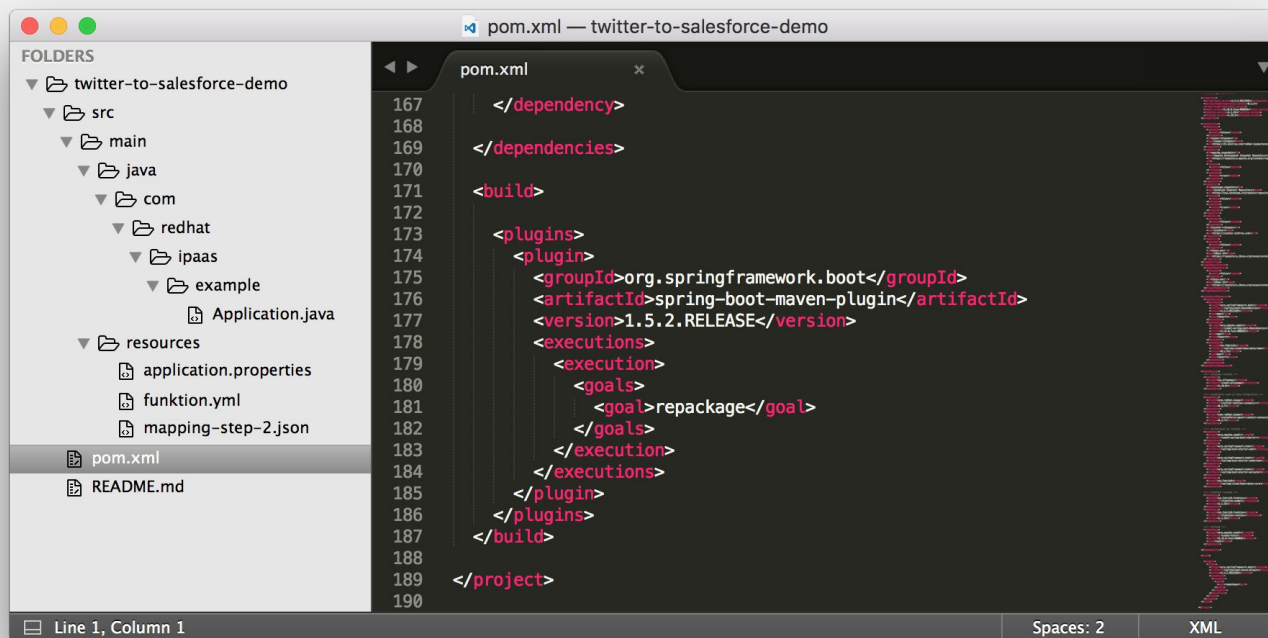
Custom Routes



```
1 package com.redhat.ipaas.example;
2
3 import org.apache.camel.builder.RouteBuilder;
4 import org.springframework.stereotype.Service;
5
6 @Service
7 public class MyRouteBuilder extends RouteBuilder {
8
9     public void configure() throws Exception {
10         from("timer?period=1000").log("Hello ${body}");
11     }
12 }
13
```

Line 1, Column 1 Tab Size: 4 Java

Maven Configuration



```
167     </dependency>
168
169   </dependencies>
170
171   <build>
172
173     <plugins>
174       <plugin>
175         <groupId>org.springframework.boot</groupId>
176         <artifactId>spring-boot-maven-plugin</artifactId>
177         <version>1.5.2.RELEASE</version>
178         <executions>
179           <execution>
180             <goals>
181               <goal>repackage</goal>
182             </goals>
183           </execution>
184         </executions>
185       </plugin>
186     </plugins>
187   </build>
188
189 </project>
190
```

Line 1, Column 1 Spaces: 2 XML

DATA MAPPER

Data Mapper

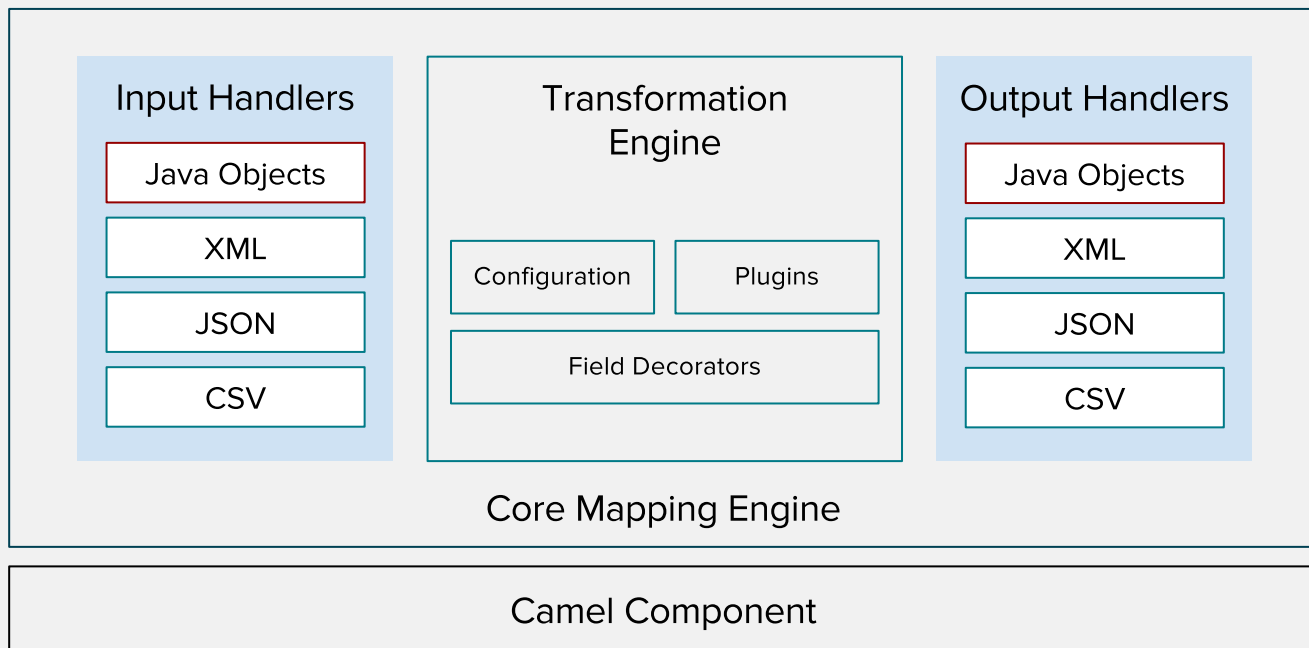
- Based on: Atlasmap
- Multi Data format
 - Java Objects
 - XML
 - Json

The screenshot displays the Red Hat IPAAS Data Mapper interface. The top navigation bar includes the Red Hat logo, the text 'RED HAT IPAAS', and a user profile for 'Hiram Chirino'. The main content area is titled 'Configure mapper' and shows two data sources: 'Status' and 'Contact'. The 'Status' source has a list of fields, with 'Name' selected. The 'Contact' source has a list of fields, with 'FirstName' and 'LastName' selected. Blue lines indicate the mapping from 'Name' in the 'Status' source to 'FirstName' and 'LastName' in the 'Contact' target. On the right, the 'Data Transformation' panel is visible, showing 'Sources' (User.Name), 'Action' (Separate), 'Separator' (Space), and 'Targets' (FirstName) with an 'Index' of 1.

Data Shapes

- Actions
- Schema
- Example XML or JSON

Data Mapper Architecture



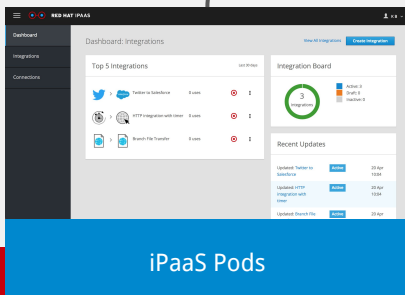
Data Mapper Features

- Multiple Source Documents
- Type auto conversions
- Separate
- Combine
- Lookups
- Constants
- Source/Target Field Functions: Case Manipulation, etc.

RUNTIME

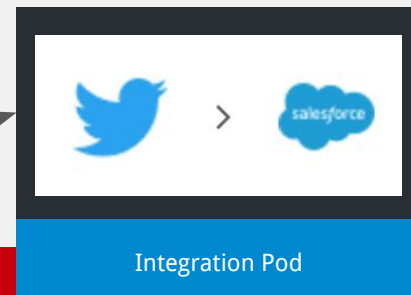
Publishing an Integration

Creates a Git Project
Per Integration



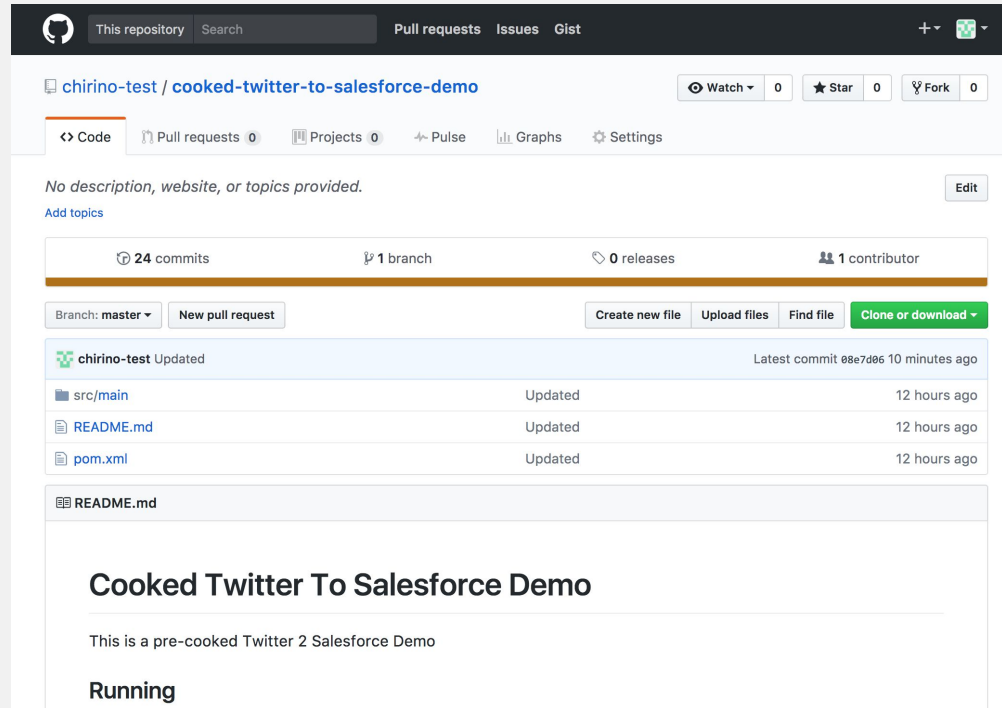
Configures
OpenShift

Builds &
Deploys



OpenShift

A Project shows up in GitHub



The screenshot shows a GitHub repository page for 'chirino-test / cooked-twitter-to-salesforce-demo'. The repository has 24 commits, 1 branch, 0 releases, and 1 contributor. The repository is currently on the 'master' branch. The file list shows 'src/main', 'README.md', and 'pom.xml', all updated 12 hours ago. The README.md file is open, showing the title 'Cooked Twitter To Salesforce Demo' and the text 'This is a pre-cooked Twitter 2 Salesforce Demo' and 'Running'.

This repository

Search

Pull requests Issues Gist

chirino-test / cooked-twitter-to-salesforce-demo

Watch 0 Star 0 Fork 0

Code Pull requests 0 Projects 0 Pulse Graphs Settings

No description, website, or topics provided. Edit

Add topics

24 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

chirino-test Updated Latest commit 08e7d06 10 minutes ago

src/main	Updated	12 hours ago
README.md	Updated	12 hours ago
pom.xml	Updated	12 hours ago

README.md

Cooked Twitter To Salesforce Demo

This is a pre-cooked Twitter 2 Salesforce Demo

Running

Build Configuration

Project ipaas ▼ Add to project ? ▼ developer ▼

Builds [Learn More](#)

Filter by label Add

Name	Last Build	Status	Duration	Created	Type	Source
twitter-to-salesforce-demo	#1	Running	3 minutes, 22 seconds	3 minutes ago	Source	https://github.com/chirino-test/twitter-to-salesforce-demo.git

Secrets

Project ipaas ▼ Add to project ? ▼ developer ▼

[Secrets](#) » twitter-to-salesforce-demo

twitter-to-salesforce-demo created 12 minutes ago Actions ▼

Opaque [Hide Secret](#)

application.properties

```
#
#Wed Apr 26 21:52:12 UTC 2017
twitter-mention.consumerSecret=
twitter-mention.accessToken=
salesforce-upsert-contact.loginUrl=
twitter-mention.consumerKey=
salesforce-upsert-contact.userName=
salesforce-upsert-contact.password=
salesforce-upsert-contact.clientId=
twitter-mention.accessTokenSecret=
salesforce-upsert-contact.clientSecret=
```

[Show Annotations](#)

Deployment Configuration

The screenshot displays the OpenShift console interface for a deployment configuration. The top navigation bar shows the project name 'ipaas' and the user 'developer'. The main content area is titled 'twitter-to-salesforce-demo' and includes a 'Deploy' button and an 'Actions' dropdown menu. The 'Configuration' tab is active, showing the following settings:

- Selectors:** integration=twitter-to-salesforce-demo
- Replicas:** 1 replica
- Strategy:** Rolling
- Timeout:** 600 sec
- Update Period:** 1 sec
- Interval:** 1 sec
- Max Unavailable:** 25%
- Max Surge:** 25%

The 'Template' section contains a warning message: "This container has no health checks to ensure your application is running correctly. [Add Health Checks](#)". Below this, the container details for 'CONTAINER: TWITTER-TO-SALESFORCE-DEMO' are listed:

- Image:** ipaas/twitter-to-salesforce-demo e2d6ef4 274.2 MiB
- Build:** twitter-to-salesforce-demo, #1
- Source:** Updated c39822a authored by chirino-test
- Ports:** 8778/TCP
- Mount:** secret-volume → /deployments/config read-only

The 'Volumes' section shows 'secret-volume' with a 'Remove' link.

Integration Pods

The screenshot displays the Red Hat IPAAS dashboard interface. At the top, there is a navigation bar with a home icon, a dropdown menu for 'Project ipaas', an 'Add to project' button, and user information for 'developer'. The left sidebar contains navigation options: Overview, Applications, Builds, Resources, and Storage. The main content area shows a deployment configuration for 'twitter-to-salesforce-demo' created 12 minutes ago. It lists the container name, image path, and ports. A central circular gauge indicates that there is 1 pod running. A URL for the application is also visible.

Project ipaas Add to project ? developer

> REDHAT IPAAS <https://ipaas.192.168.64.2.xip.io/v2/atlas>

Deployment Config [twitter-to-salesforce-demo](#) – 12 minutes ago #1

CONTAINER: TWITTER-TO-SALESFORCE-DEMO

Image: ipaas/twitter-to-salesforce-demo

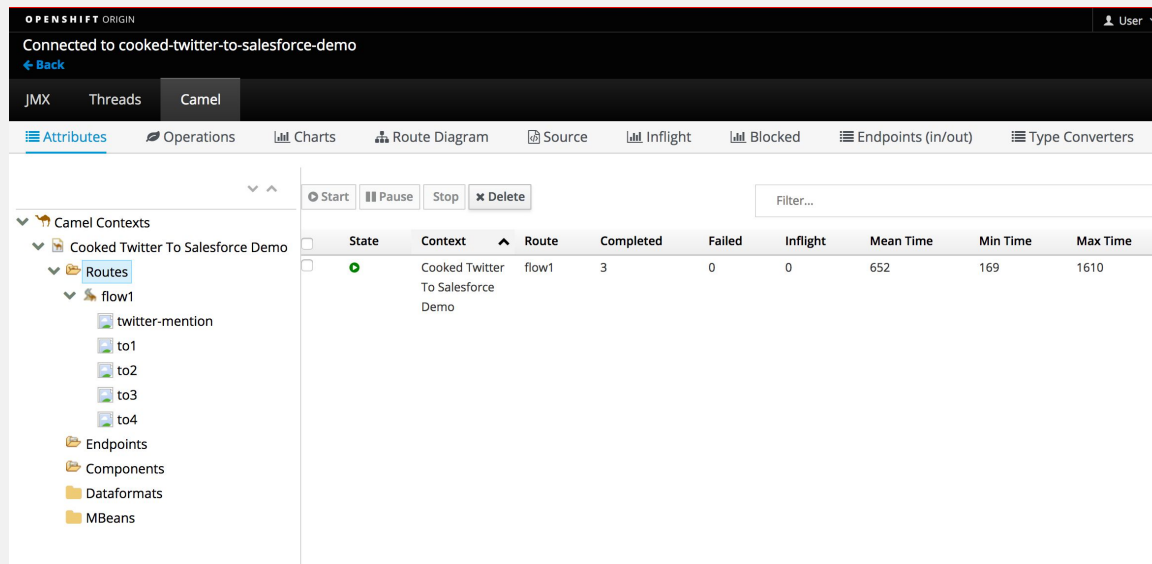
Ports: 8778/TCP

1 pod

MANAGEMENT & MONITORING

Metrics

- JVM Metrics
- Camel Metrics



The screenshot displays the OpenShift Origin console interface for a Camel context. The top navigation bar includes 'Attributes', 'Operations', 'Charts', 'Route Diagram', 'Source', 'Inflight', 'Blocked', 'Endpoints (in/out)', and 'Type Converters'. The left sidebar shows a tree view of Camel contexts, with 'Cooked Twitter To Salesforce Demo' expanded to show 'Routes' and 'flow1'. The main panel shows a table of metrics for the 'flow1' route.

State	Context	Route	Completed	Failed	Inflight	Mean Time	Min Time	Max Time
<input type="checkbox"/>	Cooked Twitter To Salesforce Demo	flow1	3	0	0	652	169	1610

Logs

The screenshot shows a Kubernetes dashboard interface. At the top, the project name is 'ipaas'. The main content area displays the pod 'cooked-twitter-to-salesforce-demo-1-q0pd5' created 2 hours ago. Below the pod name, there are several environment labels: 'deployment', 'cooked-twitter-to-salesforce-demo-1', 'deploymentconfig', 'cooked-twitter-to-salesforce-demo', 'integration', and 'cooked-twitter-to-salesforce-demo'. The 'Logs' tab is selected, showing the container logs for 'cooked-twitter-to-salesforce-demo'. The logs indicate that the Java application is starting using Jolokia, and the Spring Boot version is (v1.5.2.RELEASE). The dashboard also features a sidebar with navigation options: Overview, Applications, Builds, Resources, Storage, and Monitoring.

Project ipaas

Pods » cooked-twitter-to-salesforce-demo-1-q0pd5

cooked-twitter-to-salesforce-demo-1-q0pd5 created 2 hours ago

deployment cooked-twitter-to-salesforce-demo-1 deploymentconfig cooked-twitter-to-salesforce-demo integration cooked-twitter-to-salesforce-demo

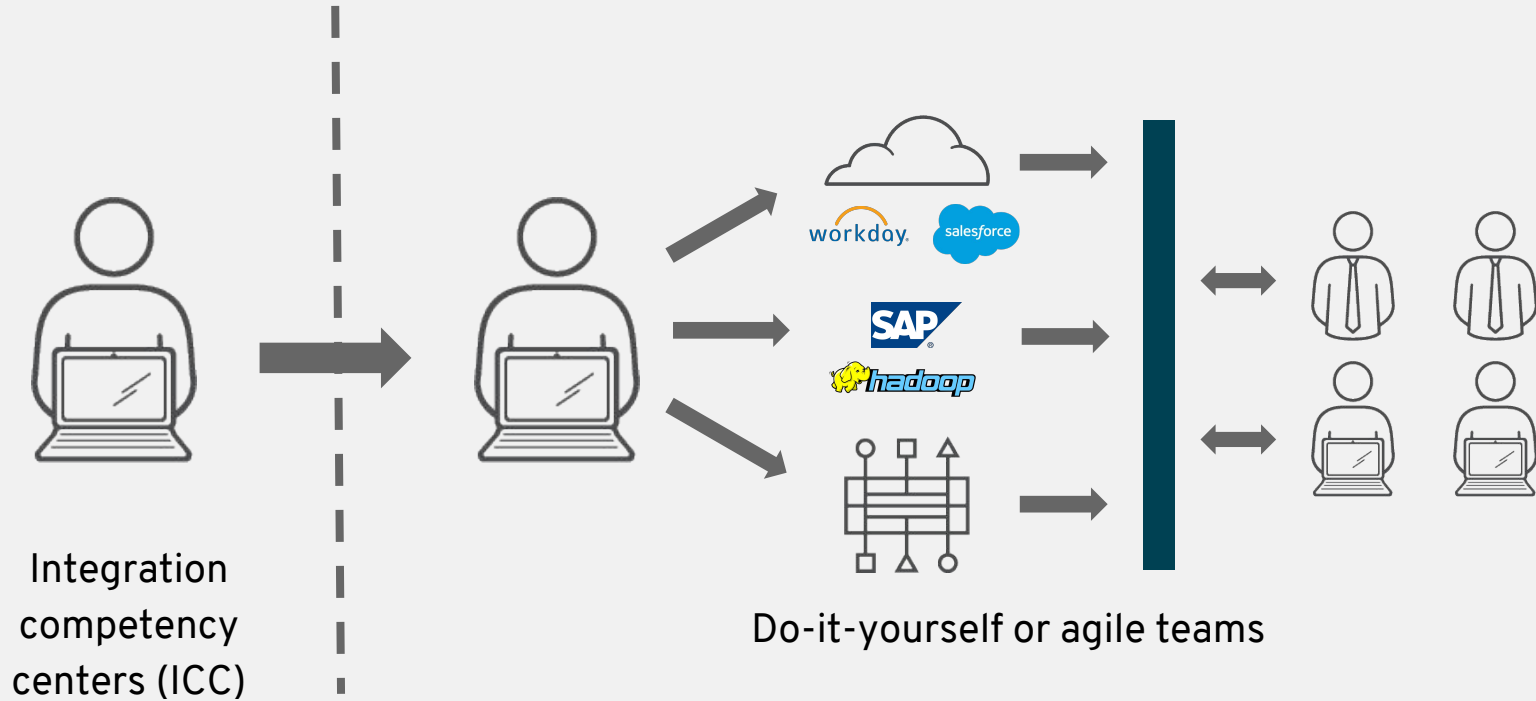
Details Environment Logs Terminal Events

Container:cooked-twitter-to-salesforce-demo — Running Log from Apr 28, 2017 8:48:23 AM

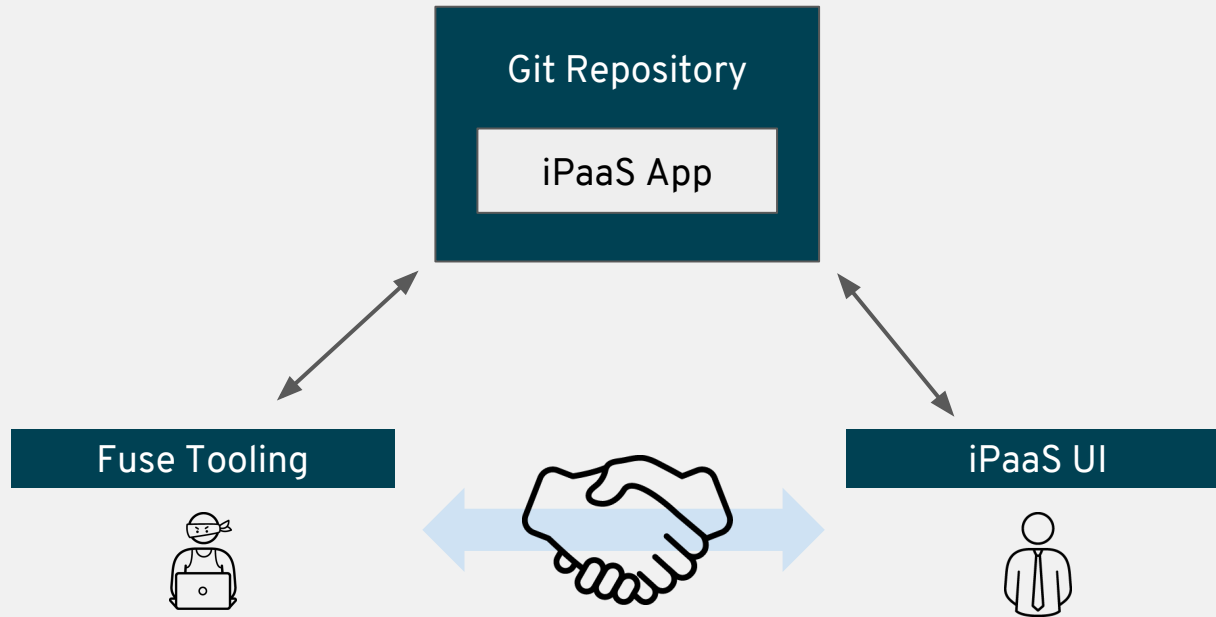
```
1 Starting the Java application using /opt/run-java/run-java.sh ...
2 exec java -javaagent:/opt/jolokia/jolokia.jar=config=/opt/jolokia/etc/jolokia.properties -cp . -jar /deployments/project-
0.1-SNAPSHOT.jar
3 I> No access restrictor found, access to any MBean is allowed
4 Jolokia: Agent started with URL https://172.17.0.2:8778/jolokia/
5
6
7  .  _ _ _ _ _
8  / \ / \ _ _ _ _ _ ( ) _ _ _ _ \ \ \ \ \
9  ( ( ) \ _ _ | ' | ' | ' | ' \ \ \ \ \
10 \ \ _ _ | _ | | | | | | | | ( | | ) ) ) )
11 ' | _ | _ | _ | _ | _ | _ | _ / / / / /
12 =====|_|=====|_|_/ / / / /
13 :: Spring Boot :: (v1.5.2.RELEASE)
```

HYBRID INTEGRATION

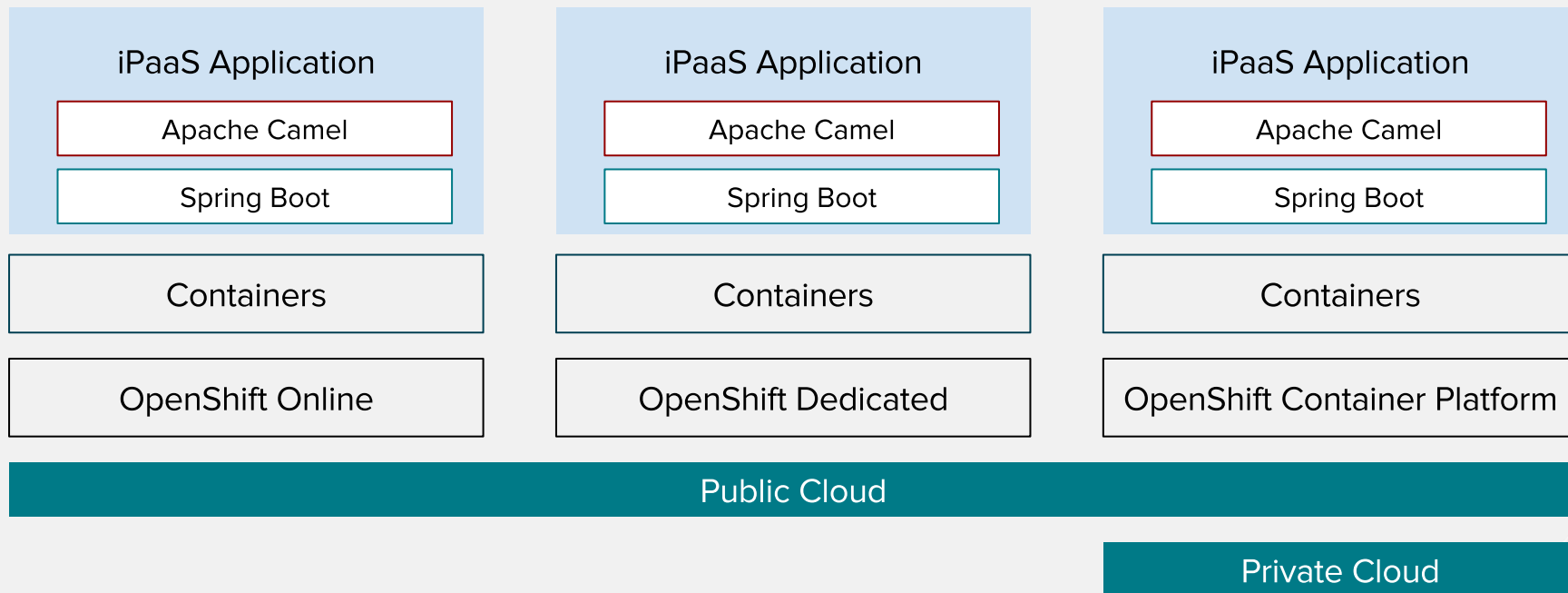
EVOLVING ORGANIZATIONAL MODEL



HYBRID DEVELOPMENT

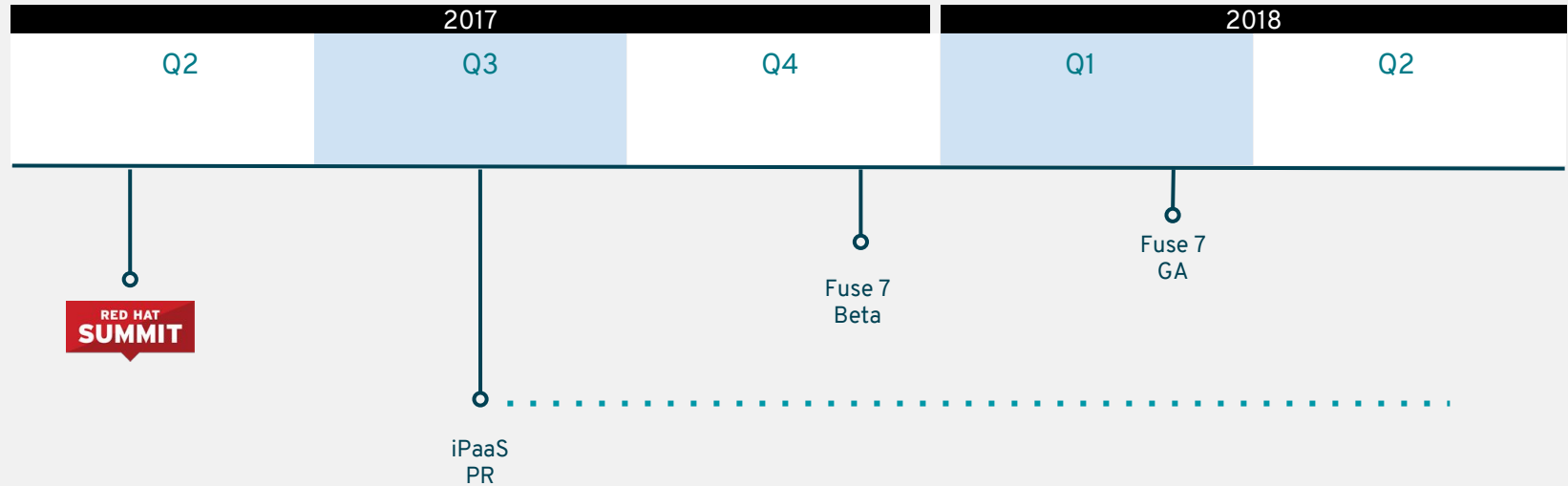


HYBRID DEPLOYMENT



PRODUCT ROADMAP

iPaaS TIMELINE





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

RED HAT
SUMMIT

THANK YOU



plus.google.com/+RedHat



facebook.com/redhatinc



linkedin.com/company/red-hat



twitter.com/RedHatNews



youtube.com/user/RedHatVideos

The logo consists of a red speech bubble shape pointing downwards, containing the text "RED HAT" in a smaller font above "SUMMIT" in a larger, bold font.

RED HAT
SUMMIT

LEARN. NETWORK.
EXPERIENCE
OPEN SOURCE.