



redhat.

# Планы по развитию интеграционных продуктов

Дмитрий Володин  
Системный архитектор, Red Hat  
[dvolodin@redhat.com](mailto:dvolodin@redhat.com)

Партнерский день, 18 Апреля 2018 года

# ЭВОЛЮЦИЯ ТЕХНОЛОГИЙ ИНТЕГРАЦИИ

	1970	1980	1990	2000	2010
Технологии	<b>Общие ресурсы на мэйнфреймах</b> ISAM VSAM IDMS	<b>Точка-точка</b> Проприетарные TCP сокеты, FTP	<b>Концентратор</b> Брокеры сообщений, CORBA	<b>Оркестрация</b> SOA Web-сервисы XML <i><b>Открытый исходный код</b></i>	<b>Медиация сервисов</b> REST JSON API's
Проблемы бизнеса	Согласованность данных	Согласованность данных Консолидированная отчетность	Предыдущие, <b>плюс</b> Электронные заказы Управление кейсами Сквозные процессы	Предыдущие, <b>плюс</b> Многошаговые и управляемые внешними системами процессы Композитные и мобильные приложения	Предыдущие, <b>плюс</b> Миграция в “облака” API как бизнес сервис “Интеграция” с AI

# ИЗМЕНЕНИЕ ЛАНДШАФТА ДЛЯ ИНТЕГРАЦИИ

ИТ предприятий претерпевают серьезные изменения. Чтобы оставаться конкурентоспособными, им нужна платформа интеграции, способная поддерживать архитектуру текущего и **следующего** поколений.

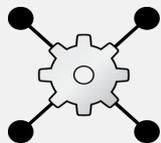
Конечные точки

Архитектура

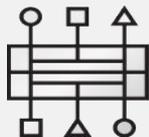
Процесс разработки

Развертывание

Инфраструктура



Web-сервисы



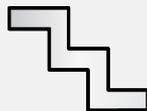
API



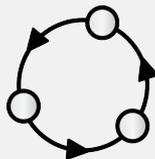
Монолитная



Микросервисная



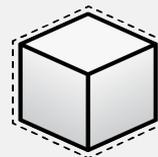
Модель "Водопад"



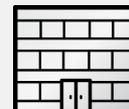
CI/CD



Сервера/VM



Контейнеры



ЦОД



Облако

# ТРИ СОСТАВЛЯЮЩИЕ ГИБКОЙ ИНТЕГРАЦИИ

Базовые возможности, необходимые современным предприятиям

## Распределенная интеграция

Легковесные, основанные на шаблонах, ориентированные на события, механизмы подключения

Гибкость

## Контейнеры

Контейнерное масштабирование, инкапсуляция и управление системами

Масштабируемость

## API

Точно определенные и документируемые, повторно используемые, управляемые конечные точки

Повторное использование

# ТЕХНОЛОГИИ ДЛЯ НАДЕЖНОЙ ОСНОВЫ

## Распределенная интеграция

Легковесные, основанные на шаблонах, ориентированные на события, механизмы подключения

RED HAT® JBOSS®  
**FUSE**

RED HAT® JBOSS®  
A-MQ

## Контейнеры

Контейнерное масштабирование, инкапсуляция и управление системами

 RED HAT®  
OPENSIFT

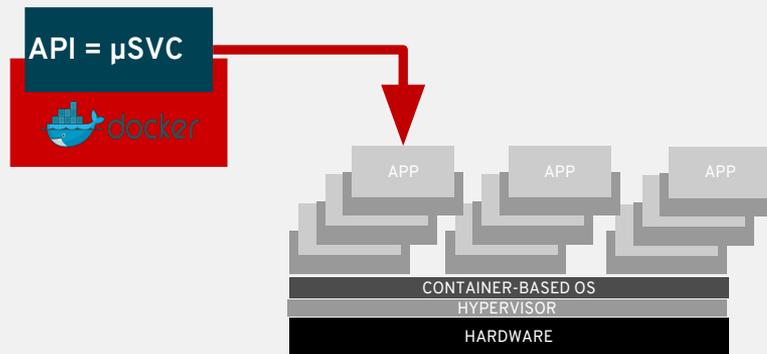
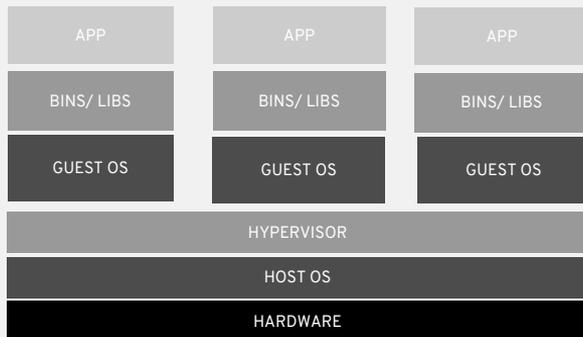
## API

Точно определенные и документируемые, повторно используемые, управляемые конечные точки

 **3scale**  
BY RED HAT®

# Интеграция $\mu$ -сервисов: ESB нового поколения

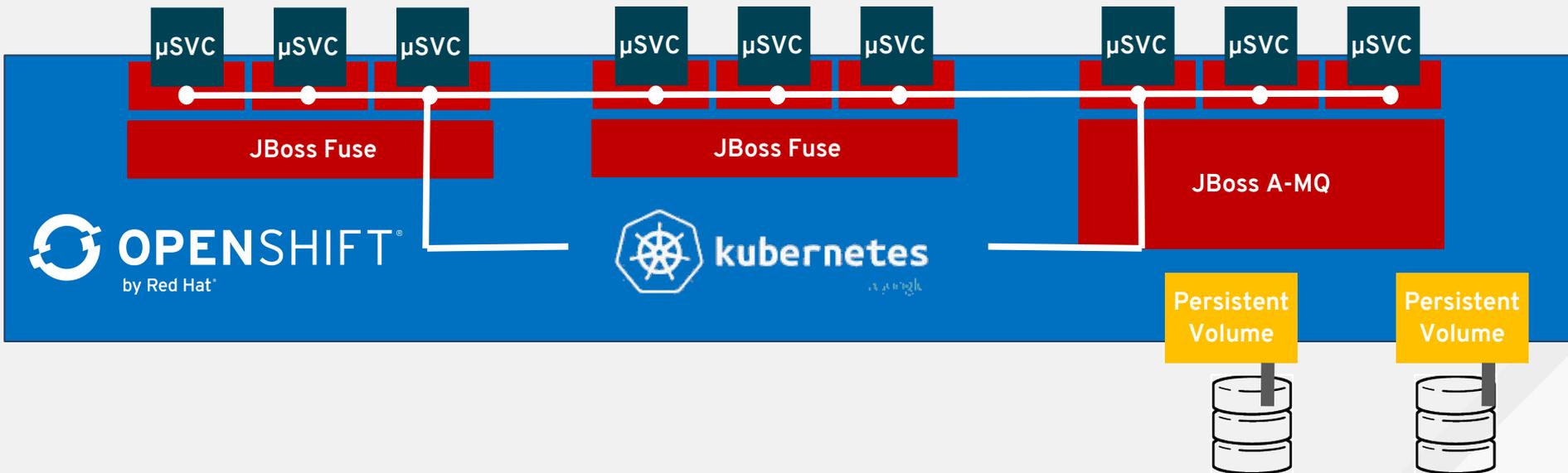
# Интеграция $\mu$ -сервисов: ESB нового поколения



- Использование контейнеров
- Изоляция без копирования ОС
- Быстрота и легкость
- Операционная эффективность
- Уровни абстракции
- Разделение приложений и их расположения
- Повторное использование образов и их слоев
- Применение методологии DevOps



# Оркестровка сервисов на уровне платформы



# Выставление сервисов через API



Routing layer

μSVC

μSVC

μSVC

μSVC

μSVC

μSVC

μSVC

μSVC

μSVC

JBoss Fuse

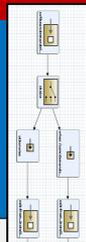
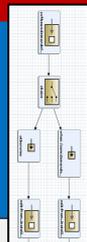
JBoss Fuse

JBoss A-MQ

Topics & Queues



Camel Routes



Persistent Volume

Persistent Volume

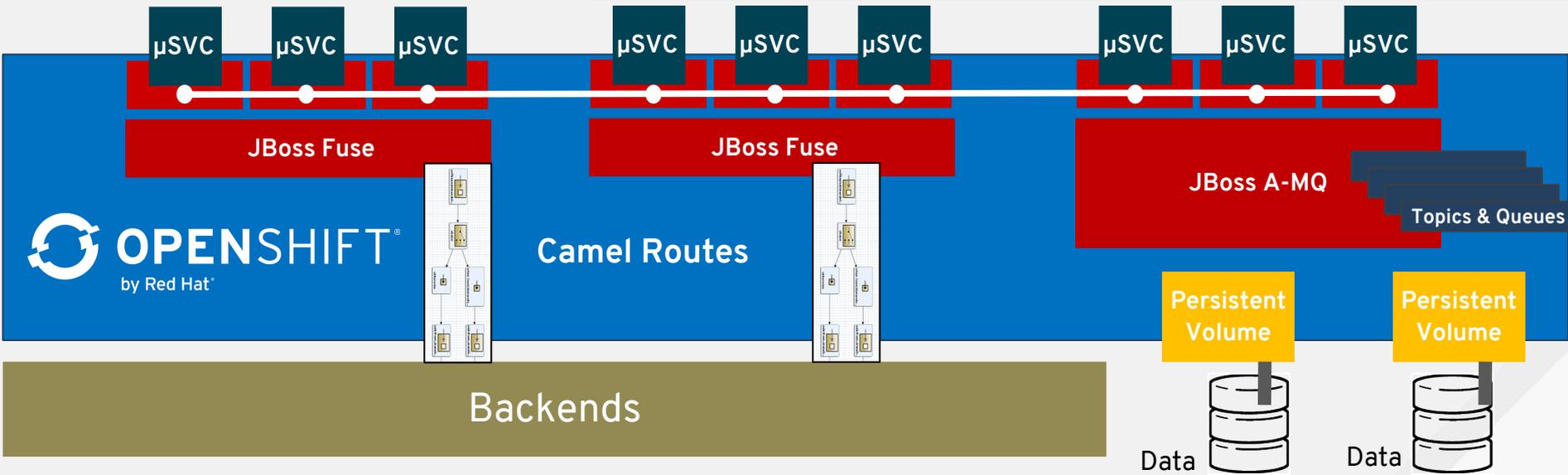
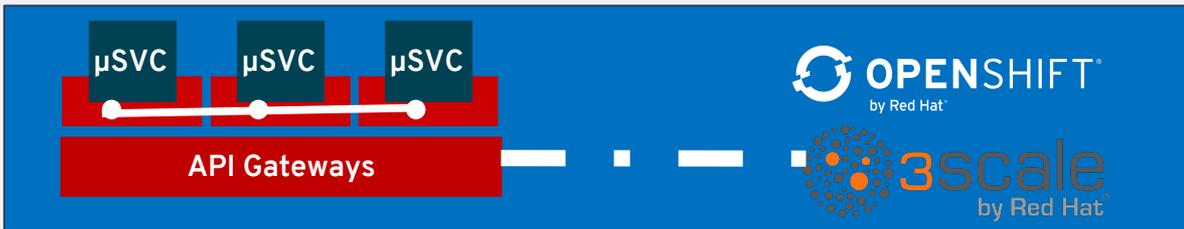


Backends

# Управление и безопасность API

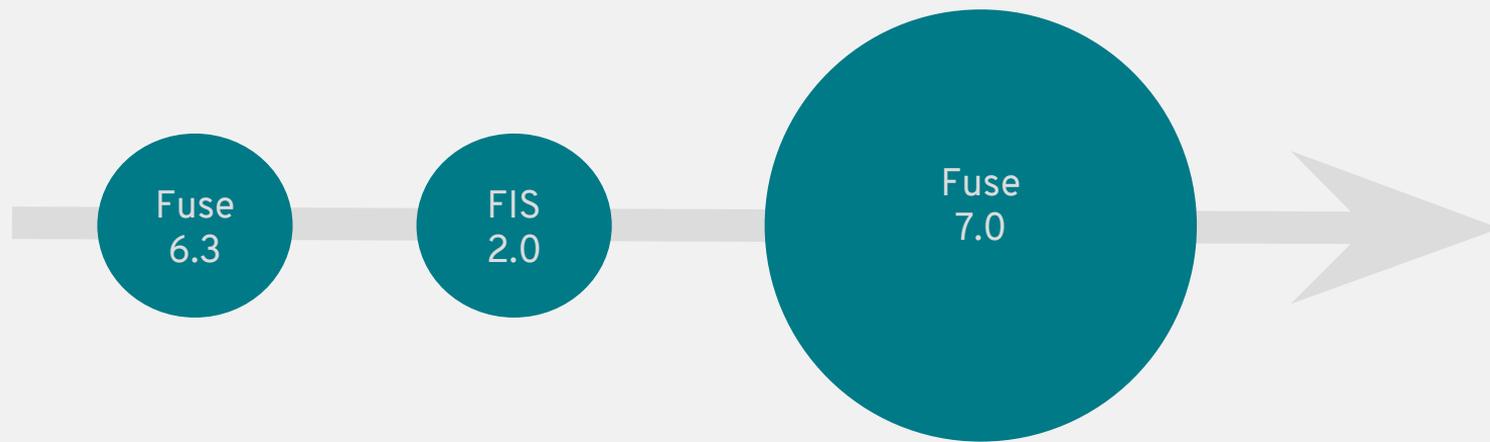


Routing layer



# **JBOSS FUSE**

# К ЧЕМУ МЫ ДОЛЖНЫ БЫТЬ ГОТОВЫ?



# FUSE 6.3

## ОБНОВЛЕНИЕ

Camel 2.17  
23 новых компонентов Camel  
CXF 3.1  
Jetty 9

## УДОБСТВО

Обновленный редактор потоков  
Новые примеры использования  
Унификация Maven репозитория

## СТАБИЛЬНОСТЬ

Исправлено 2500 проблем во всех  
подпроектах  
Исправлено 1191 проблем в продукте  
204 проблемы исправлено в Fabric8 v1

# ЧТО ТАКОЕ FUSE INTEGRATION SERVICES (FIS)?

Fuse Integration Services (FIS) - это вариант распространения JBoss Fuse, который обеспечивает поддержку инструментов и механизмов для создания контейнеризированных интеграционных сервисов на платформе OpenShift, включая:

- Образ контейнера в формате Docker
- Инструменты для создания, разработки и сборки контейнеризированных приложений Fuse
- Шаблоны для большинства интеграционных сценариев
- Непосредственная интеграция с Kubernetes для поиска сервисов, их кластеризации и конфигурации
- Все построено на базовых технологиях JBoss Fuse

# ЛЕГКОВЕСНАЯ СРЕДА ИСПОЛНЕНИЯ ИНТЕГРАЦИЙ

Образы контейнеров в формате Docker для Karaf и Spring Boot обеспечивают основу для построения контейнеризированных интеграционных сервисов.

## SPRING BOOT RUNTIME

Разработка кода и конфигурация

Специфические для приложения  
зависимости

Spring Boot

fis-java-openshift

- Простая конфигурация, bean-контейнер
- Поддержка Spring и Java DSL для Apache Camel
- Автоматическое связывание (Autowired)
- Основан на Camel 2.18

## KARAF RUNTIME

Разработка кода и конфигурация

Специфические для приложения  
зависимости

Apache Karaf

fis-karaf-openshift

- OSGi контейнер на базе Karaf
- Поддерживает Blueprint DSL для Apache Camel
- Выровнен по версиям с Fuse 6.3 для более простого перехода от развертывания Fuse на базе Fabric8 v1 к варианту развертывания на OpenShift

# FUSE 7.x

Три основных аспекта развития продукта



## Fuse Standalone



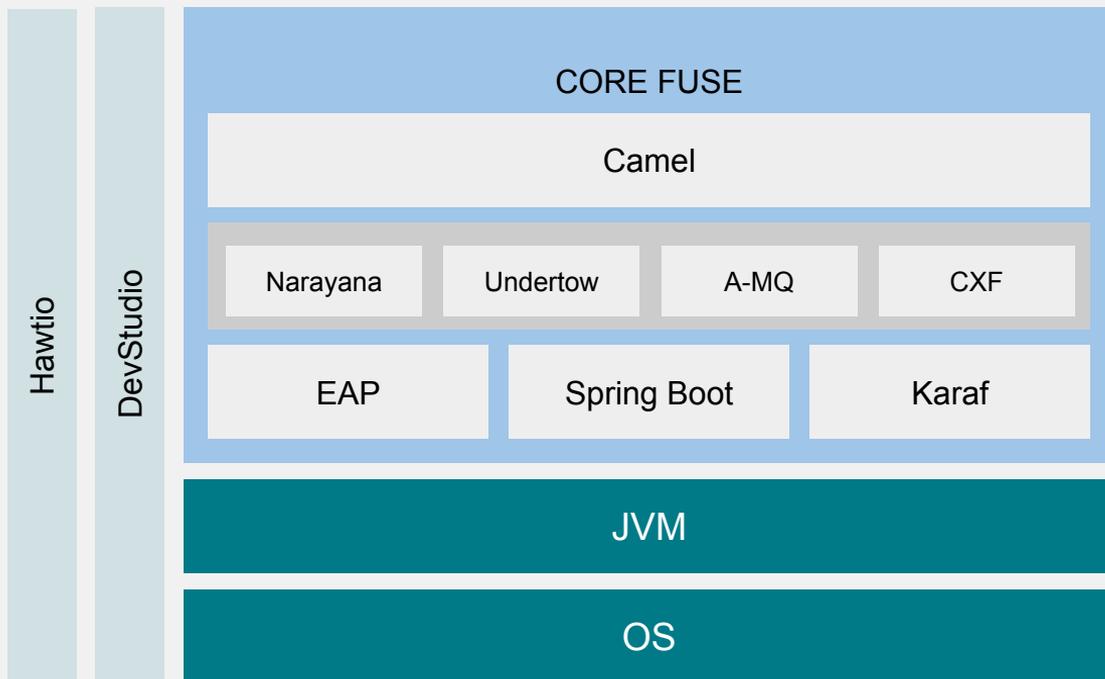
## Fuse Integration Services



## Fuse iPaaS



# FUSE STANDALONE



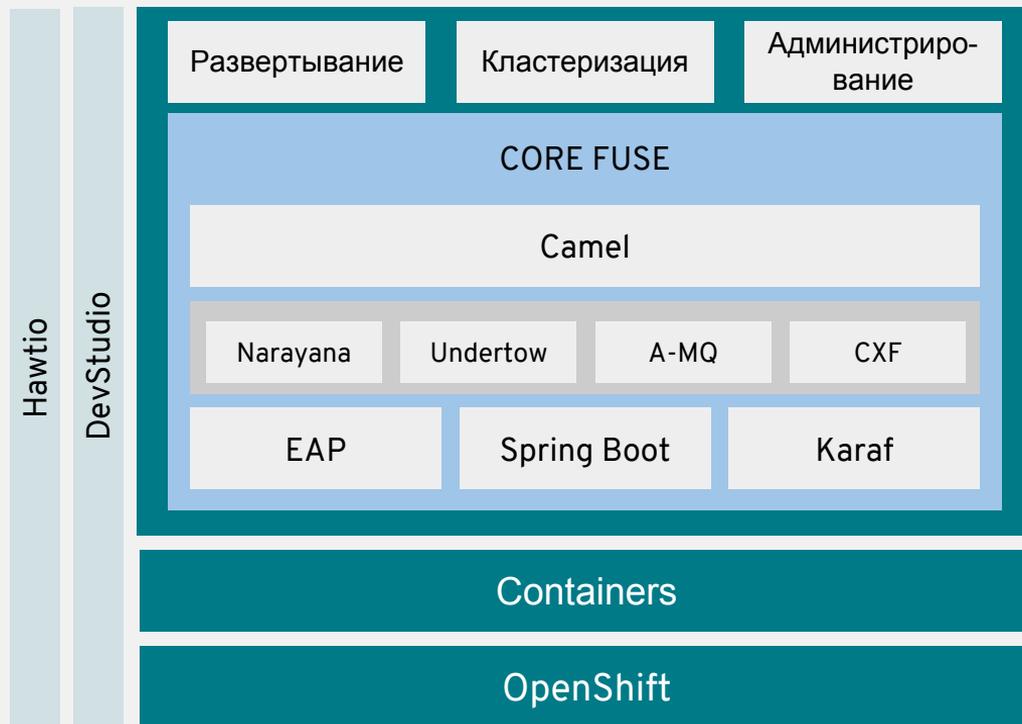


# FUSE STANDALONE

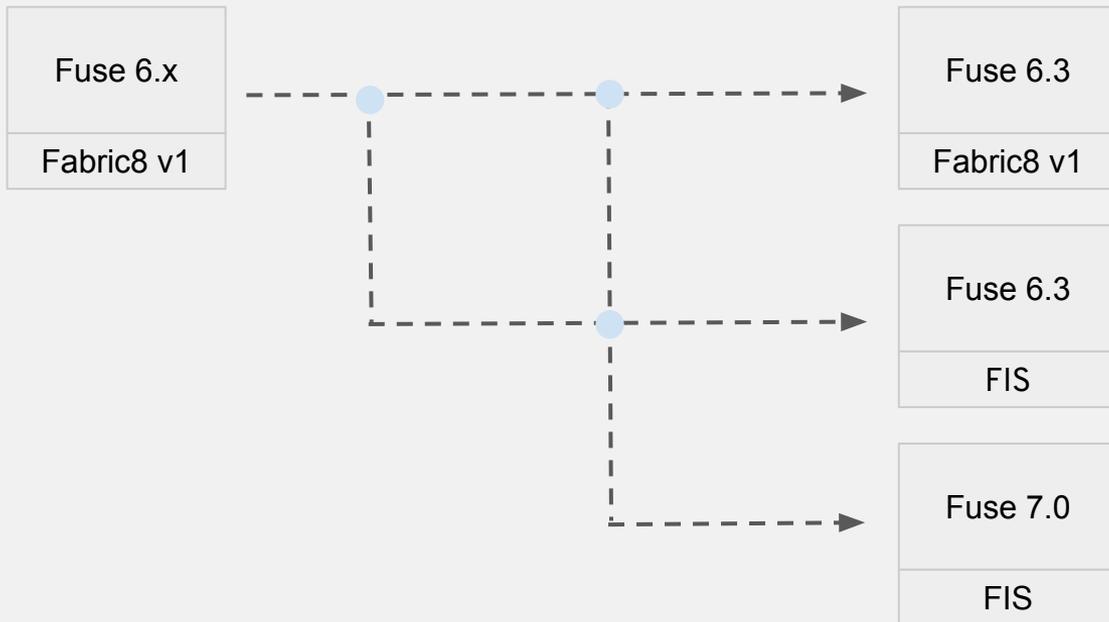
- Camel 2.21
- Karaf 4.0
- Поддержка Spring Boot
- Поддержка EAP 7.x
- Менеджер транзакций Narayana
  - Geronimo больше не поддерживается
- Web-контейнер Undertow
  - Jetty больше не поддерживается
- Сертификация с AMQ 6/7
- Дополнительные компоненты для EAP 7
- SwitchYard больше не поддерживается
- Интеграция с фокусом на API
  - Редактор REST DSL
  - SOAP <-> REST
  - Интеграция с Red Hat 3scale



# FUSE INTEGRATION SERVICES



# ПЕРЕХОД Fuse 6.x -> Fuse 7.0



Оставайтесь на той же версии Fuse (End of Maintenance Support - Jan 2022)

Используйте FIS 2.0, когда будете готовы

Ждите Fuse 7.0

# FUSE iPaaS



**...СКОРО...**



# Red Hat iPaaS - интеграция становится легче

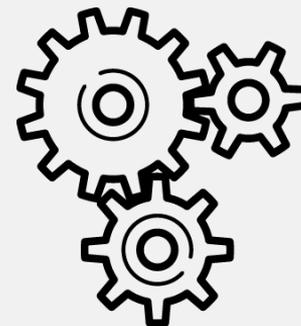
# ЧТО ТАКОЕ iPaaS И О ЧЕМ МЫ БУДЕМ СЕГОДНЯ ГОВОРИТЬ?



Вовлечение  
пользователей  
“Рядовые” интеграторы



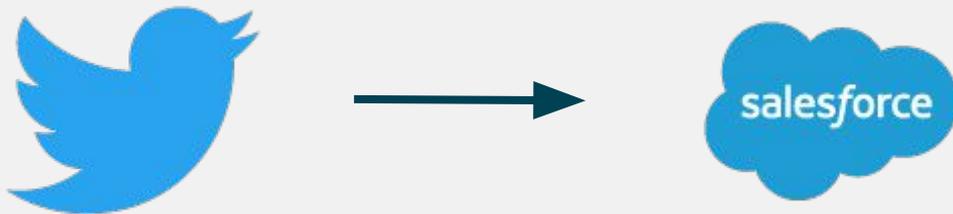
Облачные  
сервисы  
Публичные и частные



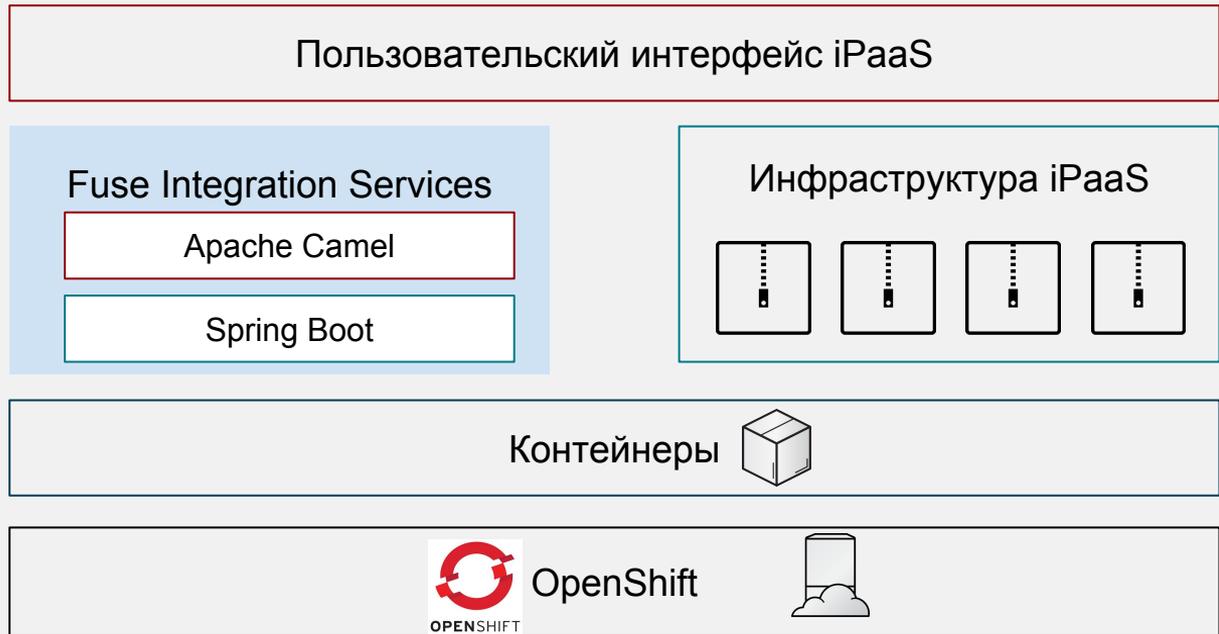
Базовые  
интеграции  
Соединение, трансформация,  
маршрутизация

# НАПРИМЕР ...

**Цель :** *Захват упоминаний о корпоративном мероприятии и создание контактов в Salesforce для последующего анализа потенциальных сделок.*



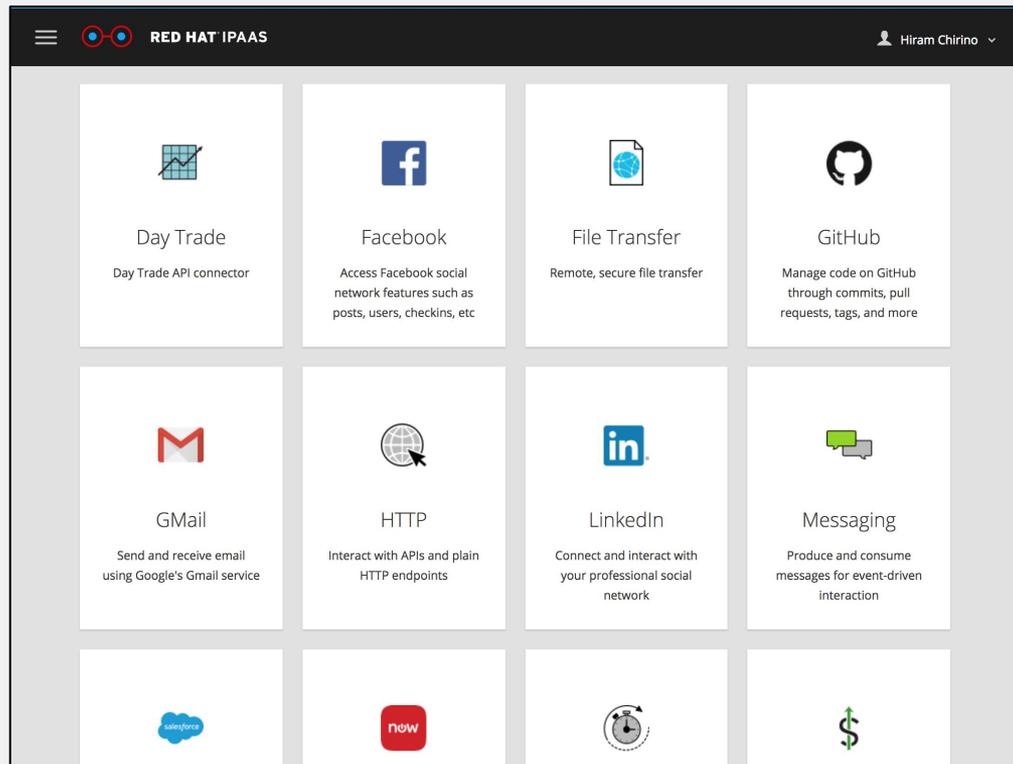
# АРХИТЕКТУРА ПРОДУКТА iPaaS



# КОННЕКТОРЫ

# КОННЕКТОРЫ iPaaS

- Коннекторы реализованы в виде компонентов Apache Camel
- Camel поддерживает подключение к более чем 150 различным системам и технологиям



# НАСТРОЙКА ПОДКЛЮЧЕНИЯ В iPaaS

- Настраивает основные параметры подключения к конечной системе
- Обычно используется для хранения информации, такой как учетные данные
- Общие для различных интеграций
- Поддерживается валидация настроек подключения

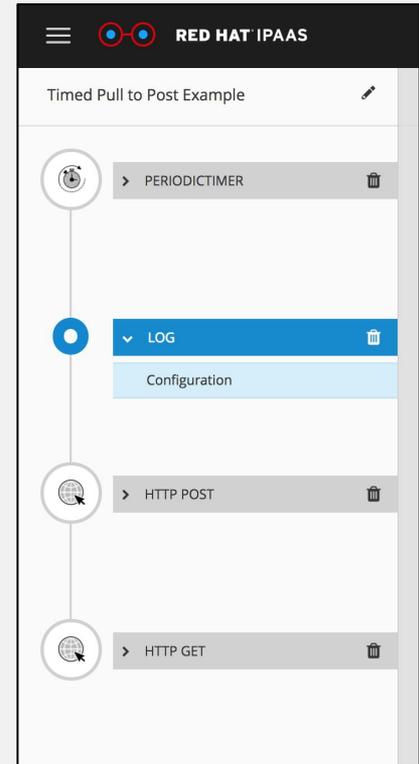
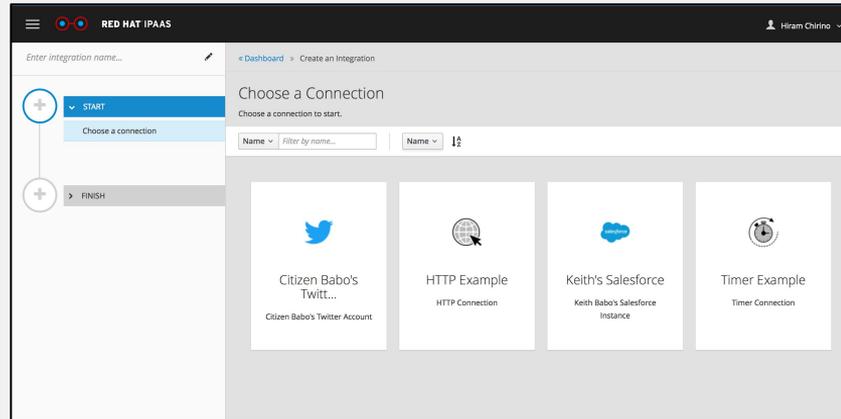
The screenshot shows the 'RED HAT IPAAS' interface for creating a connection. The breadcrumb trail is 'Home > Connections > Create Connection'. The user 'Hiram Chirino' is logged in. A progress indicator shows three steps: 'Select Connector' (1), 'Configure Connector' (2, currently active), and 'Name Connection' (3). The 'Configure Connector' step is titled 'Twitter Configuration' and includes a 'Validate' button. The configuration fields are:

- Access Token:
- Access Token Secret:
- Consumer Key:
- Consumer Secret:

# ИНТЕГРАЦИИ

# ИНТЕГРАЦИИ iPaaS

- Последовательности шагов для подключения систем источника и приемника
- Реализованы в виде шагов в стиле Enterprise Integration Patterns (EIP)
- Каждая интеграция - это отдельный проект в виде FIS



# ДЕЙСТВИЯ И ШАГИ

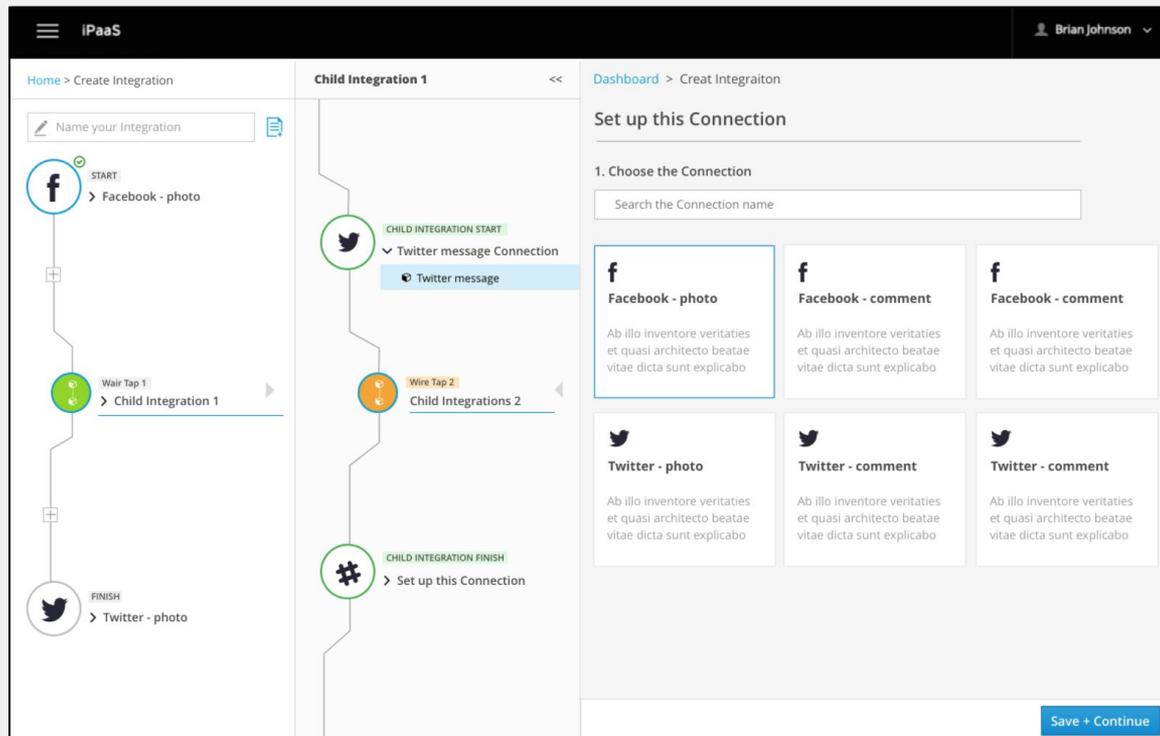
The screenshot shows the Red Hat IPAAS interface. At the top, there is a navigation bar with the Red Hat logo and the text 'RED HAT IPAAS'. On the right side of the navigation bar, the user's name 'Hiram Chirino' is displayed. Below the navigation bar, there is a search bar for integration names and a breadcrumb trail: '< Dashboard > Create an Integration'. The main content area is titled 'Choose an Action' and includes a sub-header 'Choose an action for the integration.' and a search filter 'Filter by name...'. On the left side, there is a vertical sidebar with a Twitter icon and a 'FINISH' button. The main content area lists several 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

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

# ДОЧЕРНИЕ ИНТЕГРАЦИИ

- Возможность для создания сложных потоков обработки
- Прослеживание взаимодействия между родительскими и дочерними процессами



# СВЯЗЫВАНИЕ ДАННЫХ

# СВЯЗЫВАНИЕ ДАННЫХ

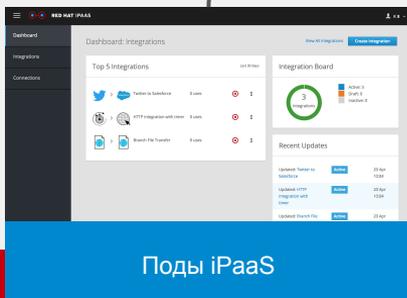
- Основан на проекте Atlasmap
- Поддержка различных форматов данных
  - Объекты Java
  - XML
  - JSON
  - CSV

The screenshot displays the 'RED HAT IPAAS' 'Configure mapper' interface. It features two source lists: 'Status' on the left and 'Contact' on the right. The 'Status' list includes fields like 'Lang', 'ListedCount', 'Location', 'MiniProfileImageUrl', 'MiniProfileImageUrlHttps', 'Name', 'OriginalProfileImageUrl', 'OriginalProfileImageUrlHttps', 'ProfileBackgroundColor', 'ProfileBackgroundImageUrl', 'ProfileBackgroundImageUrlHttps', 'ProfileBackgroundTiled', 'ProfileBannerIPadRetinaURL', 'ProfileBannerIPadURL', 'ProfileBannerMobileRetinaURL', and 'ProfileBannerMobileURL'. The 'Contact' list includes fields like 'EmailBouncedReason', 'Fax', 'FirstName', 'HomePhone', 'IsEmailBounced', 'Jigsaw', 'JigsawContactId', 'Languages\_c', 'LastCURequestDate', 'LastCUUpdateDate', 'LeadSource', 'Level\_c', 'MailingAddress', 'MailingCity', and 'MailingCountry'. The 'Name' field in the 'Status' list and the 'FirstName' and 'LastName' fields in the 'Contact' list are highlighted in blue. Blue lines connect 'Name' to 'FirstName' and 'Name' to 'LastName'. On the right, the 'Data Transformation' panel shows 'Sources' with 'User.Name' selected, 'Action' set to 'Separate', 'Separator' set to 'Space', and 'Targets' with 'FirstName' selected and 'Index' set to '1'. The top navigation bar includes a menu icon, the Red Hat logo, 'RED HAT IPAAS', and the user name 'Hiram Chirino'. The breadcrumb trail shows '< Dashboard > Configure mapper'.

# СРЕДА ИСПОЛНЕНИЯ

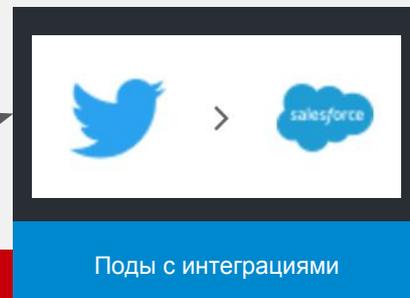
# ПУБЛИКАЦИЯ ИНТЕГРАЦИЙ

Создание проекта git  
(для каждой интеграции)



Настройка  
OpenShift

Сборка и  
развертывание



OpenShift

# СБОРКА И РАЗВЕРТЫВАНИЕ

Project ipaaS

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	<a href="https://github.com/chirino-test/twitter-">https://github.com/chirino-test/twitter-</a>

Project ipaaS

Overview

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

# УПРАВЛЕНИЕ И МОНИТОРИНГ

# МЕТРИКИ

- Метрики JVM
- Метрики Camel

The screenshot displays the OpenShift 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 main content area shows a tree view of Camel contexts on the left and a table of metrics on the right.

**Table Data:**

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

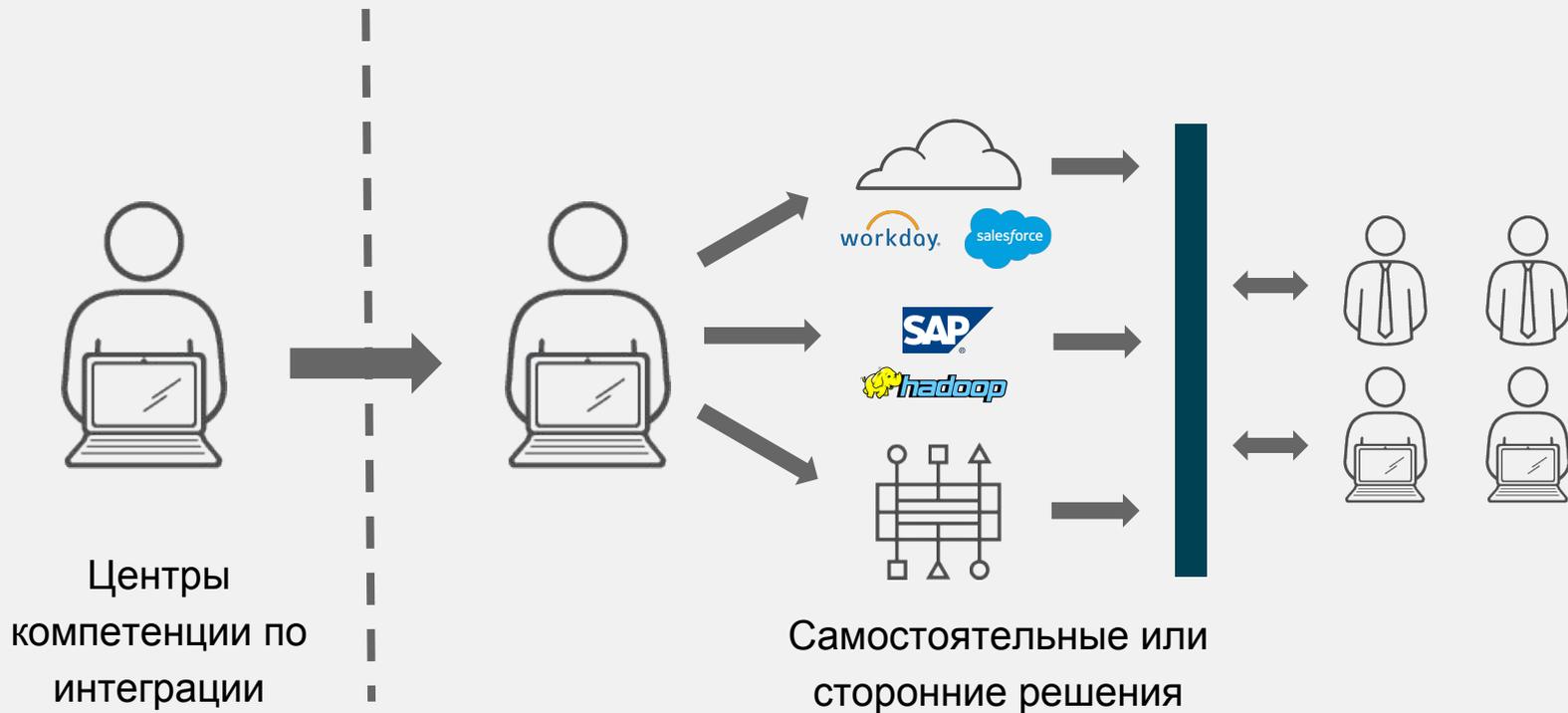
# ЛОГИ

The screenshot shows a Kubernetes dashboard interface. At the top, there's a navigation bar with a home icon, a dropdown menu for 'Project ipaaS', an 'Add to project' button, and a user profile for 'developer'. The main content area displays the pod 'cooked-twitter-to-salesforce-demo-1-q0pd5' created 2 hours ago. Below the pod name are several tabs: '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 details for the container 'cooked-twitter-to-salesforce-demo' which is in a 'Running' state. The log output shows the following:

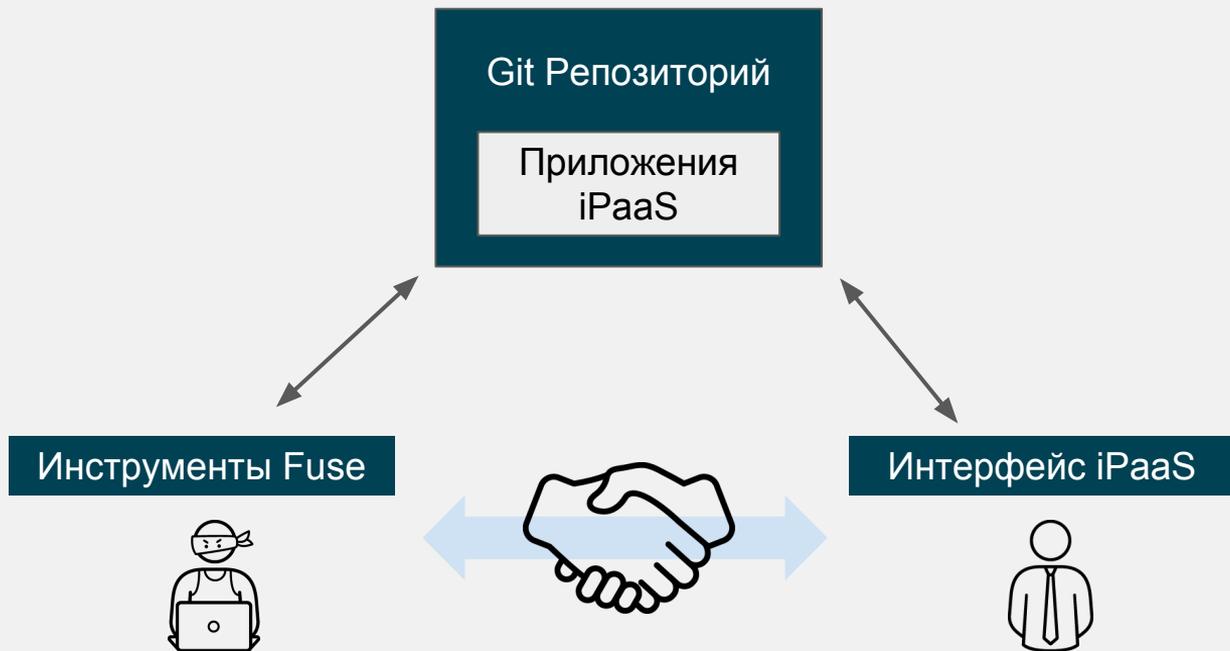
```
1 Starting the Java application using /opt/run-java/run-java.sh ...
2 exec java -javaagent:/opt/jolokia/jolokia.jar=config=/opt/jolokia/etc/jolokia.properties -cp . -jar /deployments/project-
  0.1-SNAPSHOT.jar
3 I> No access restrictor found, access to any MBean is allowed
4 Jolokia: Agent started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

# ГИБРИДНАЯ ИНТЕГРАЦИЯ

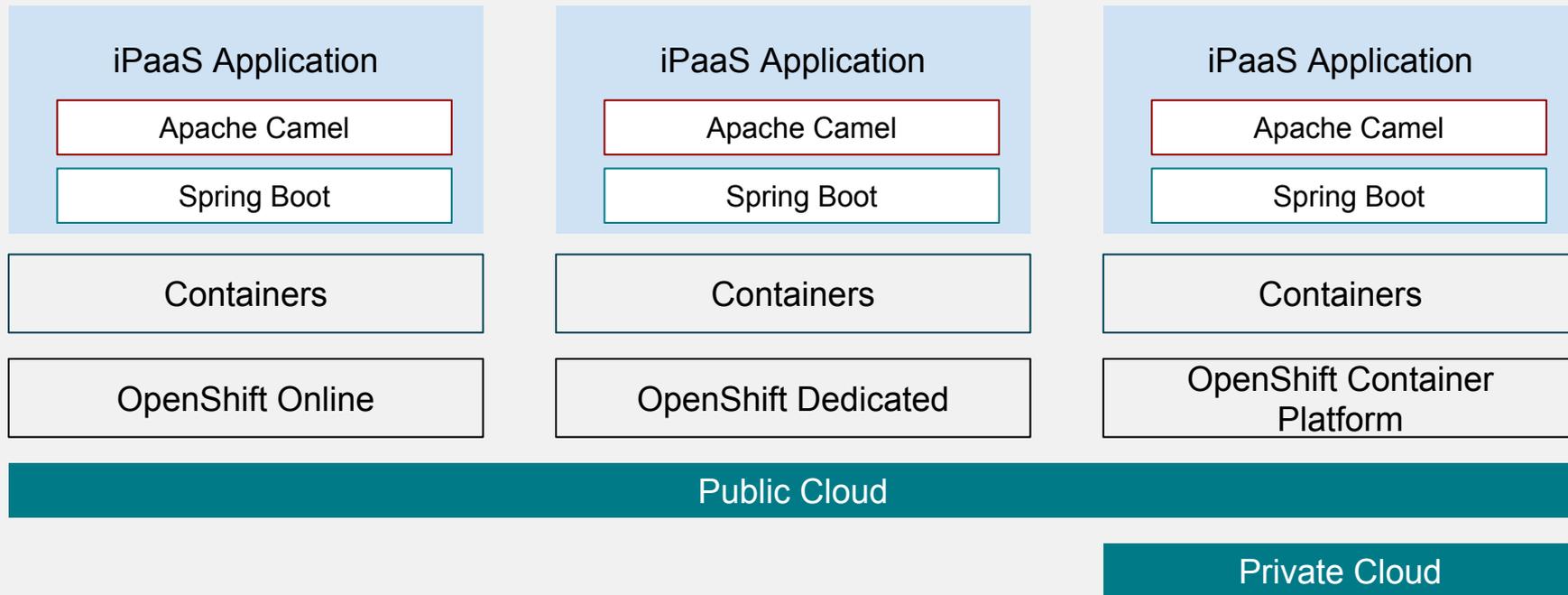
# ЭВОЛЮЦИЯ РАЗРАБОТКИ ИНТЕГРАЦИОННЫХ СЕРВИСОВ



# ГИБРИДНАЯ РАЗРАБОТКА

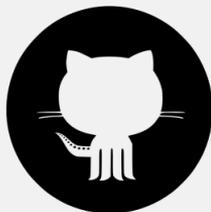


# ГИБРИДНОЕ ИСПОЛНЕНИЕ



# ДЕМОНСТРАЦИЯ

# ДЕМОНСТРАЦИЯ



## Red Hat Forum Moscow 2017 Demo Camel client and server

<https://github.com/dmvolod/rh-forum-moscow-2017>

The screenshot shows the OpenShift console interface for a project named 'demo-sb-camel'. On the left is a navigation sidebar with icons for Overview, Applications, Builds, Resources, and Storage. The main area displays two deployment configurations side-by-side. The left deployment is for 'grpc-sb-camel-client', showing it is complete 29 minutes ago. It uses the 'demo-sb-camel/grpc-sb-camel-client' image and has one pod running. The right deployment is for 'hello-grpc-service', showing it is complete 35 minutes ago. It uses the 'demo-sb-camel/grpc-sb-camel-server' image and also has one pod running. Both deployments are running on the 'SPRING-BOOT' container.

Project: demo-sb-camel

Build `grpc-sb-camel-client-s2i, #5` ✓ Complete. 29 minutes ago View Log

`grpc-sb-camel-client`

Deployment Config `grpc-sb-camel-client` - 29 minutes ago #5

CONTAINER: SPRING-BOOT

- Image: demo-sb-camel/grpc-sb-camel-client
- Ports: 8001/TCP (vcom-tunnel), 8778/TCP (jolokia), 9779/TCP (prometheus)

1 pod

Build `grpc-sb-camel-server-s2i, #10` ✓ Complete. 35 minutes ago View Log

`hello-grpc-service`

Deployment Config `grpc-sb-camel-server` - 35 minutes ago #8

CONTAINER: SPRING-BOOT

- Image: demo-sb-camel/grpc-sb-camel-server
- Ports: 8002/TCP (teradataordbms), 8778/TCP (jolokia), 9779/TCP (prometheus)

1 pod



# СПАСИБО ЗА ВНИМАНИЕ



[plus.google.com/+RedHat](https://plus.google.com/+RedHat)



[facebook.com/redhatinc](https://facebook.com/redhatinc)



[linkedin.com/company/red-hat](https://linkedin.com/company/red-hat)



[twitter.com/RedHatNews](https://twitter.com/RedHatNews)



[youtube.com/user/RedHatVideos](https://youtube.com/user/RedHatVideos)