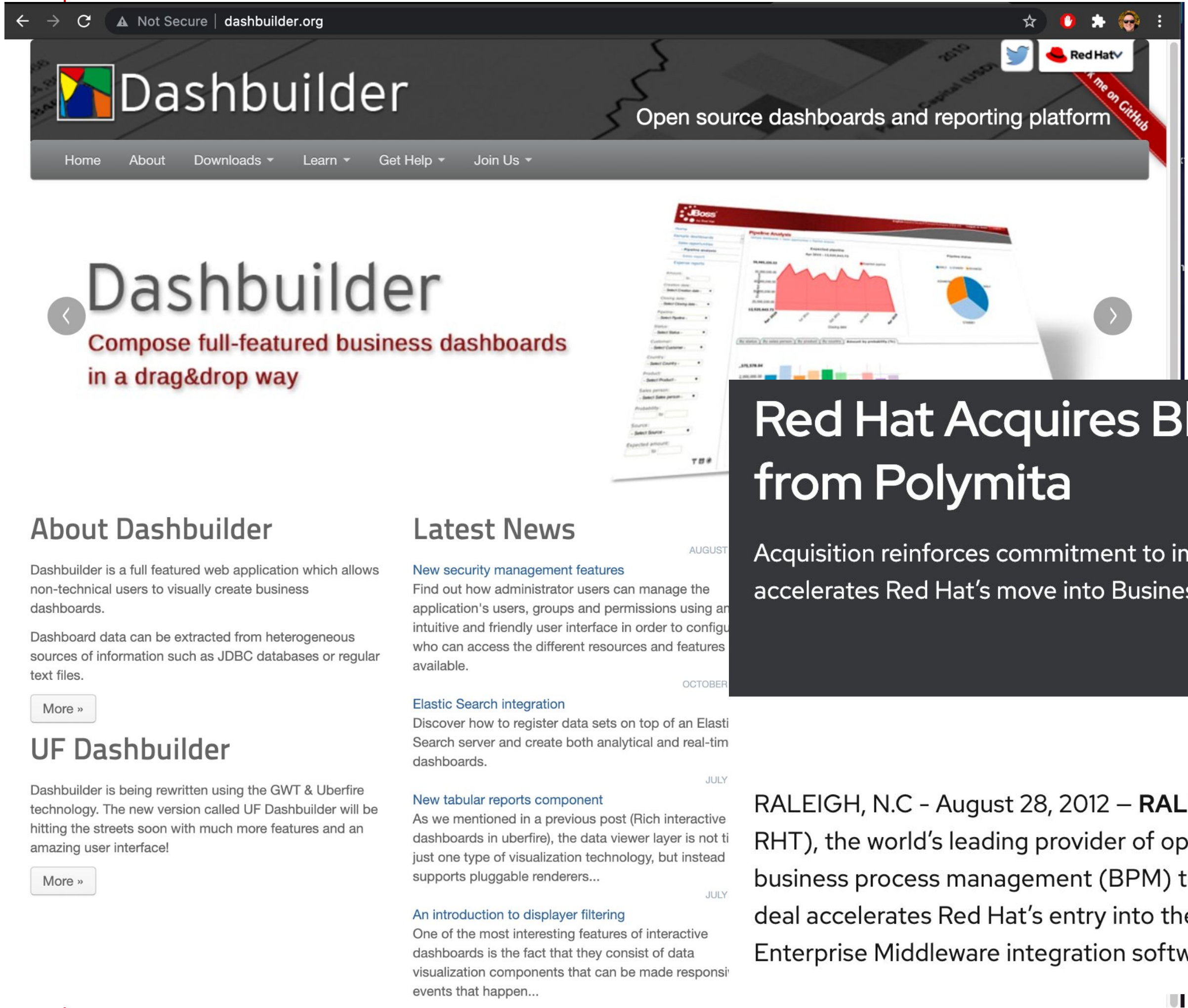
Migrating DashBuilder to Quarkus

Eder Ignatowicz
Principal Software Engineer
@ederign

William Siqueira
Senior Software Engineer
@William_Antonio



Dashbuilder

Open source dashboards and reporting platform

Dashbuilder

Compose full-featured business dashboards in a drag&drop way

About Dashbuilder

Dashbuilder is a full featured web application which allows non-technical users to visually create business dashboards.

Dashboard data can be extracted from heterogeneous sources of information such as JDBC databases or regular text files.

More »

UF Dashbuilder

Dashbuilder is being rewritten using the GWT & Uberfire technology. The new version called UF Dashbuilder will be hitting the streets soon with much more features and an amazing user interface!

More »

Latest News

AUGUST

New security management features

Find out how administrator users can manage the application's users, groups and permissions using an intuitive and friendly user interface in order to configu who can access the different resources and features available.

OCTOBER

Elastic Search integration

Discover how to register data sets on top of an Elasti Search server and create both analytical and real-tim dashboards.

JULY

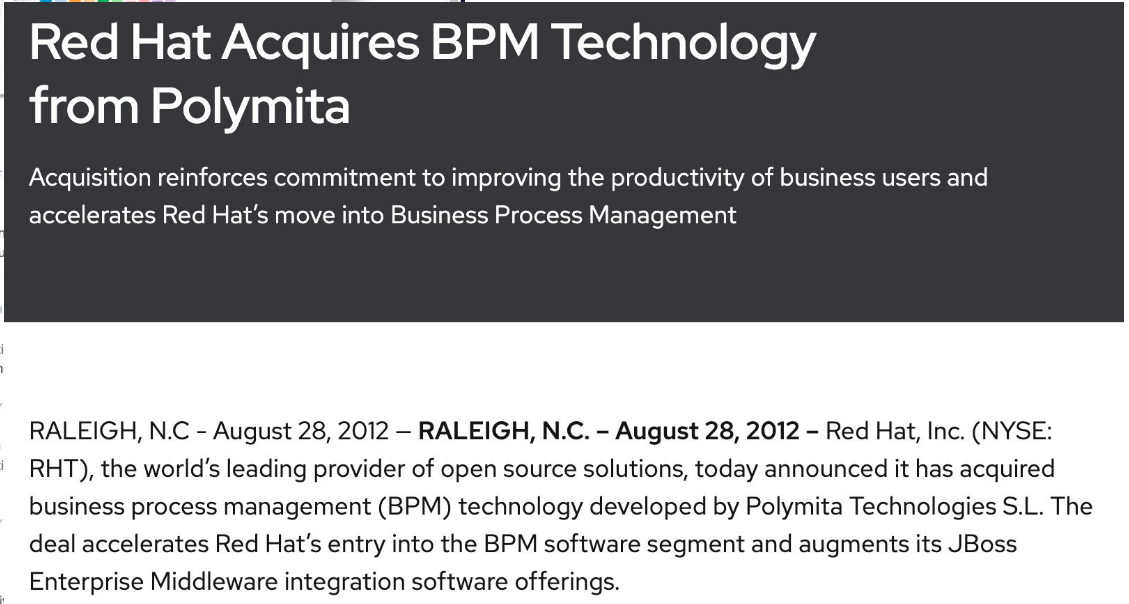
New tabular reports component

As we mentioned in a previous post (Rich interactive dashboards in uberfire), the data viewer layer is not ti just one type of visualization technology, but instead supports pluggable renderers...

JULY

An introduction to displayer filtering

One of the most interesting features of interactive dashboards is the fact that they consist of data visualization components that can be made responsi events that happen...



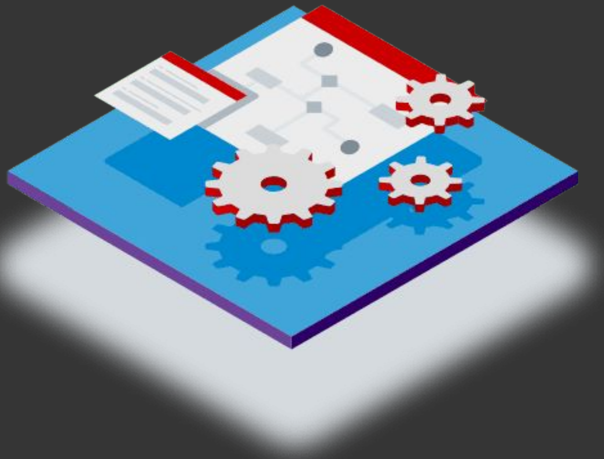
Red Hat Acquires BPM Technology from Polymita

Acquisition reinforces commitment to improving the productivity of business users and accelerates Red Hat's move into Business Process Management

RALEIGH, N.C - August 28, 2012 – **RALEIGH, N.C. – August 28, 2012 –** Red Hat, Inc. (NYSE: RHT), the world's leading provider of open source solutions, today announced it has acquired business process management (BPM) technology developed by Polymita Technologies S.L. The deal accelerates Red Hat's entry into the BPM software segment and augments its JBoss Enterprise Middleware integration software offerings.

KIE

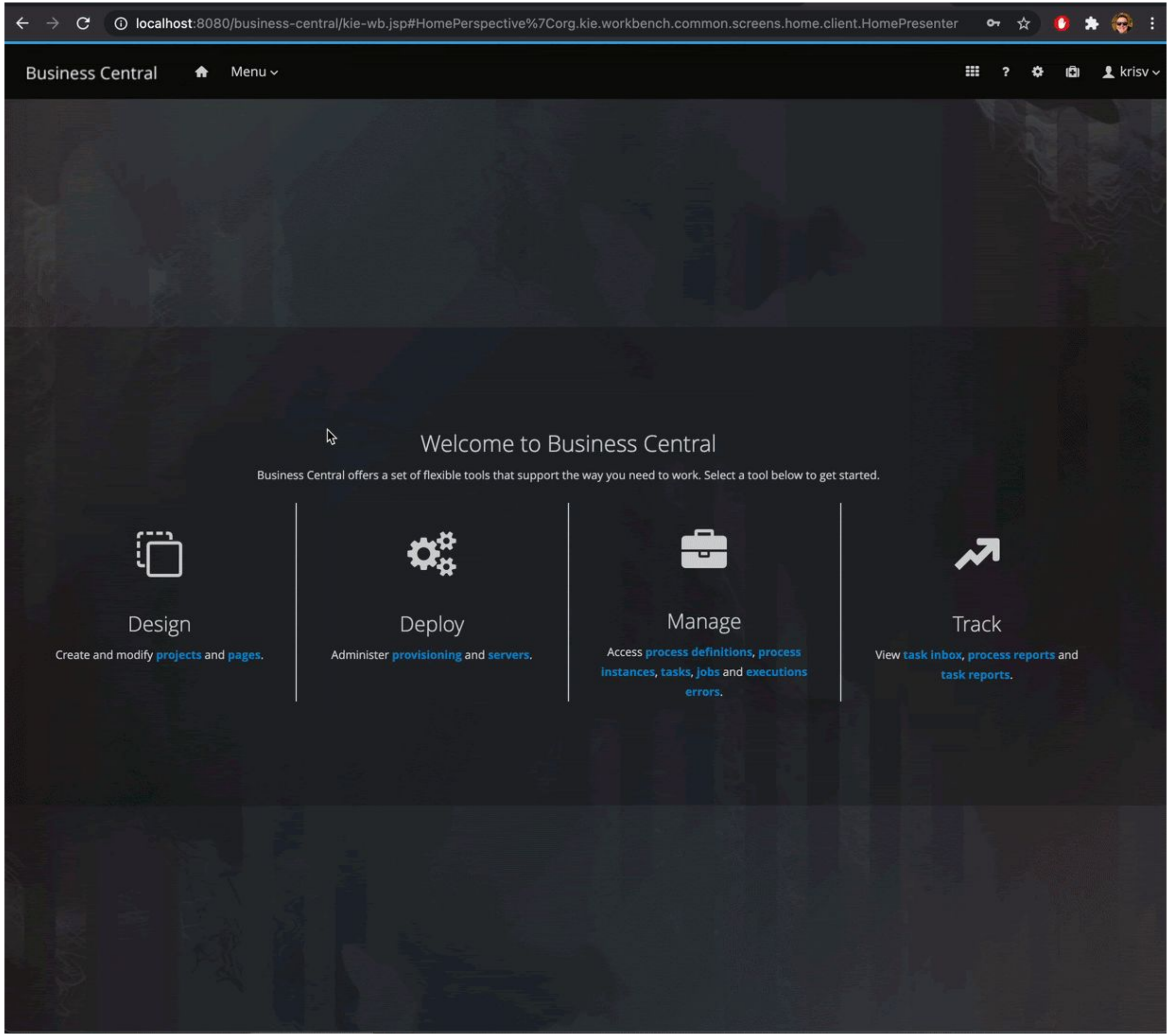
jBPM



Drools

OptaPlanner

Red Hat



localhost:8080/business-central/kie-wb.jsp#HomePerspective%7Corg.kie.workbench.common.screens.home.client.HomePresenter

Welcome to Business Central

Business Central offers a set of flexible tools that support the way you need to work. Select a tool below to get started.

Design

Create and modify projects and pages.

Deploy

Administer provisioning and servers.

Manage

Access process definitions, process instances, tasks, jobs and executions errors.

Track

View task inbox, process reports and task reports.

What is DashBuilder?

"DashBuilder is a tool for the building of reporting and monitoring business dashboards, licensed under the business-friendly Apache Software License (ASL)"

DashBuilder V7+ (August 2020)

New strategic features

Architectural Housekeeping

Cloud Native Dashbuilder "v7" (mid 2020)

Flexible Layout and Navigation

Prometheus, Kafka, Elasticsearch, CSV and JDBC Support

Victory Charts and Other Components

Time Series with Apex Charts

DashBuilder Lightweight Runtime and Multi-Tenancy

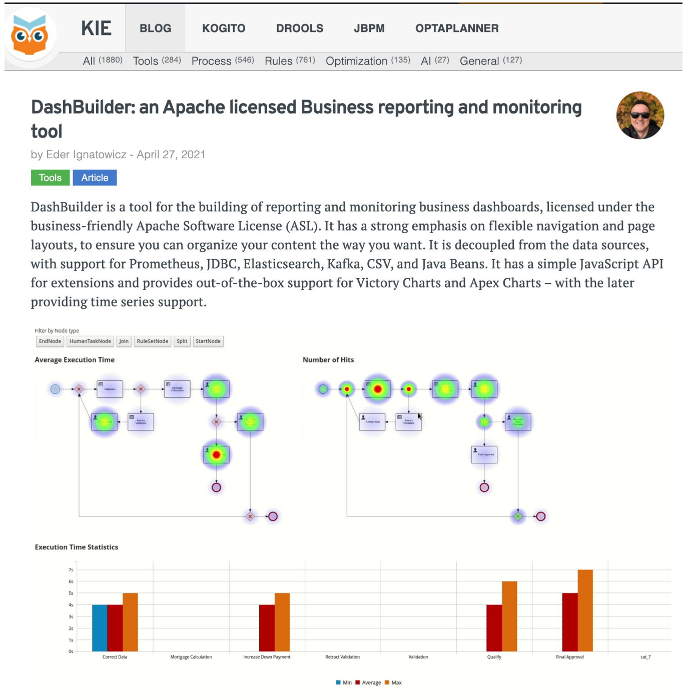
Easy way to import/export your dashboards

Embedded Mode

JavaScript API for Extensions

Declarative Programmatic API

Apache Software License (ASL)



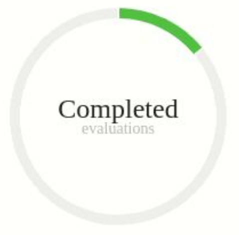
DashBuilder: an Apache licensed Business reporting and monitoring tool

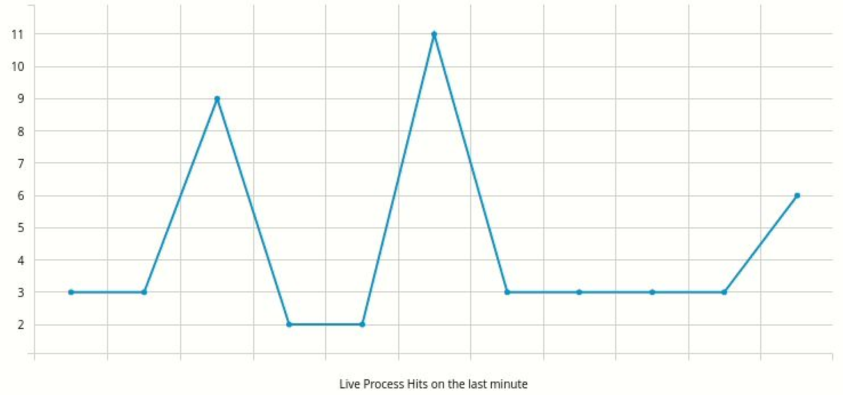
by Eder Ignatowicz - April 27, 2021

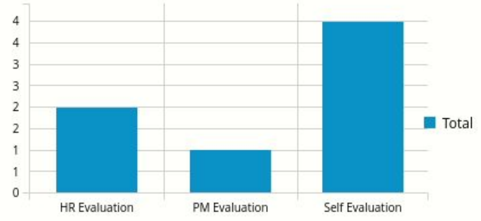
Tools Article

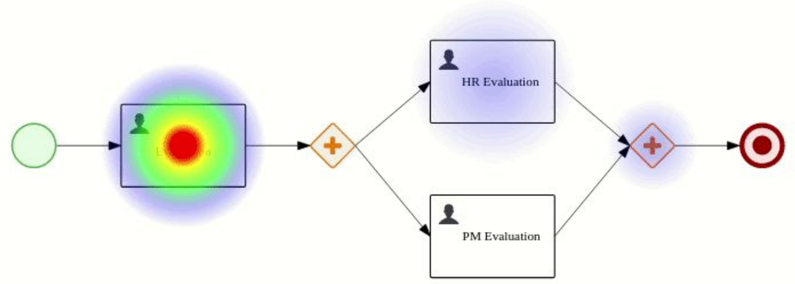
DashBuilder is a tool for the building of reporting and monitoring business dashboards, licensed under the business-friendly Apache Software License (ASL). It has a strong emphasis on flexible navigation and page layouts, to ensure you can organize your content the way you want. It is decoupled from the data sources, with support for Prometheus, JDBC, Elasticsearch, Kafka, CSV, and Java Beans. It has a simple JavaScript API for extensions and provides out-of-the-box support for Victory Charts and Apex Charts – with the later providing time series support.

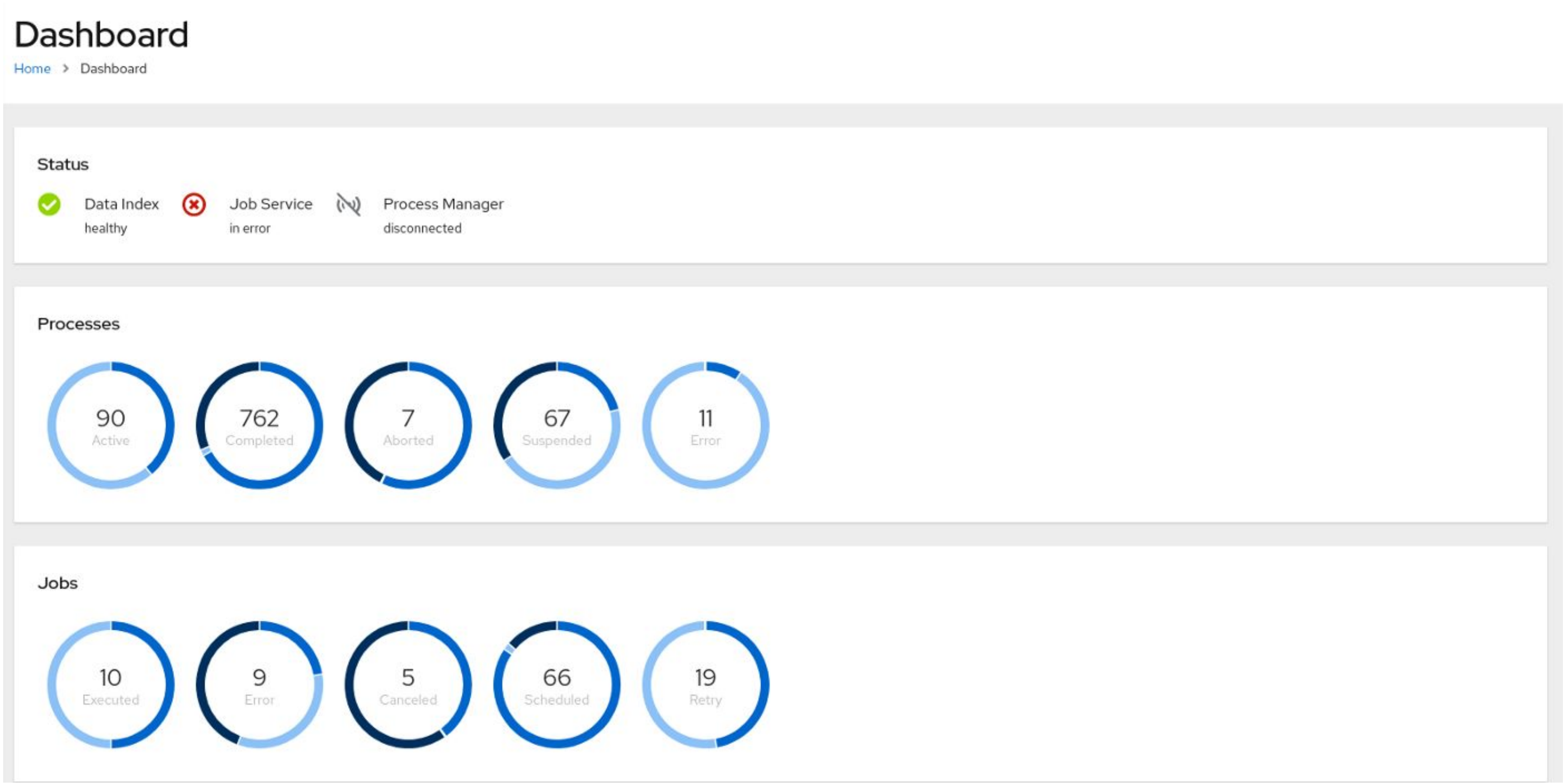
Employees Evaluation Progress











Dashboard

Home > Dashboard

Status

Data Index
healthy

Job Service
in error

Process Manager
disconnected

Processes

90 Active

762 Completed

7 Aborted

67 Suspended

11 Error

Jobs

10 Executed

9 Error

5 Canceled

66 Scheduled

19 Retry

```java
public class SimpleDashboard {

    public static void main(String[] args) {
        DataSet dataSet = newDataSetBuilder().column("Country", ColumnType.LABEL)
                                             .column("Population", ColumnType.NUMBER)
                                             .row("Brazil", "211")
                                             .row("United States", "328")
                                             .row("Cuba", "11")
                                             .row("India", "1366")
                                             .row("China", "1398")
                                             .buildDataSet();

        DisplayerSettings populationBar = newBarChartSettings().subType_Column()
                                                               .width(800)
                                                               .height(600)
                                                               .dataset(dataSet)
                                                               .column("Country")
                                                               .column("Population")
                                                               .buildSettings();
        Page page = page("Countries Population",
                         row("<h3> Countries Population </h3>"),
                         row(ComponentFactory.displayer(populationBar)));
        Navigation navigation = navigation(group("Countries Information", item(page)));

        Dashboard populationDashboard = DashboardFactory.dashboard(asList(page), navigation);

        DashboardExporter.get().export(populationDashboard,
                                       "/path/to/export.zip",
                                       ExportType.ZIP);
    }

}
```

Cloud-Native Applications

Cloud-Native

Cloud-native technologies empower organizations to build and run **scalable** applications in modern, dynamic environments such as **public, private, and hybrid clouds**. **Containers**, **service meshes**, **microservices**, **immutable infrastructure**, and declarative APIs exemplify this approach.

These techniques enable loosely coupled systems that are resilient, **manageable**, and **observable**. Combined with **robust automation**, they allow engineers to make high-impact changes frequently and predictably with minimal toil.

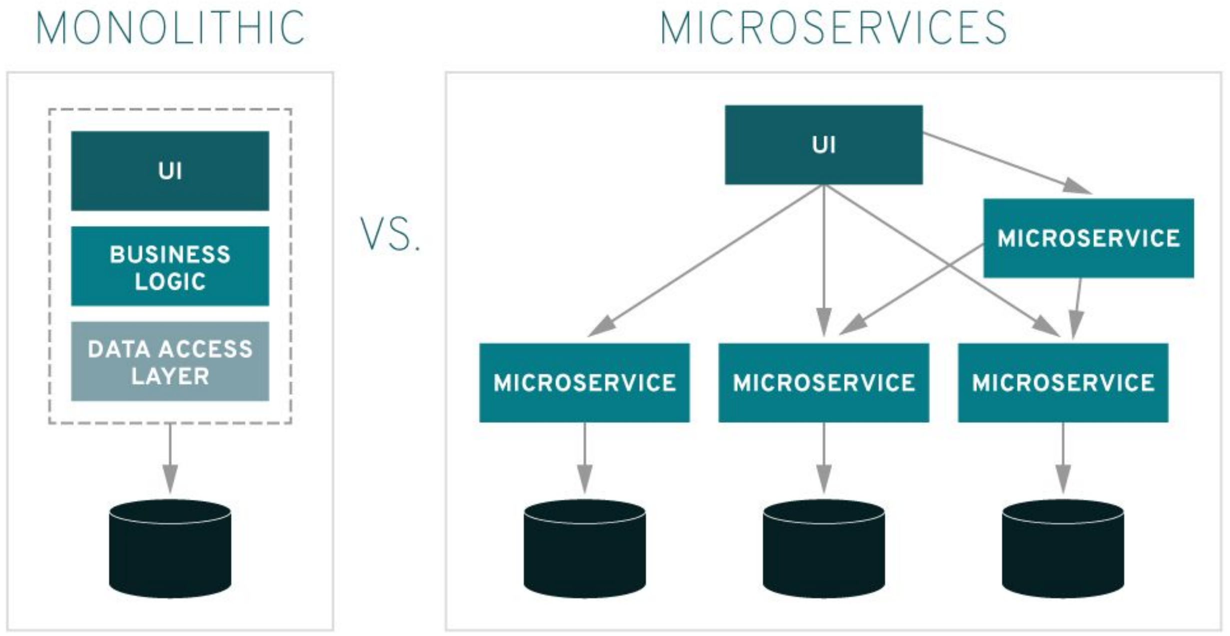
*Emphasis mine

https://github.com/cncf/foundation/blob/master/charter.md

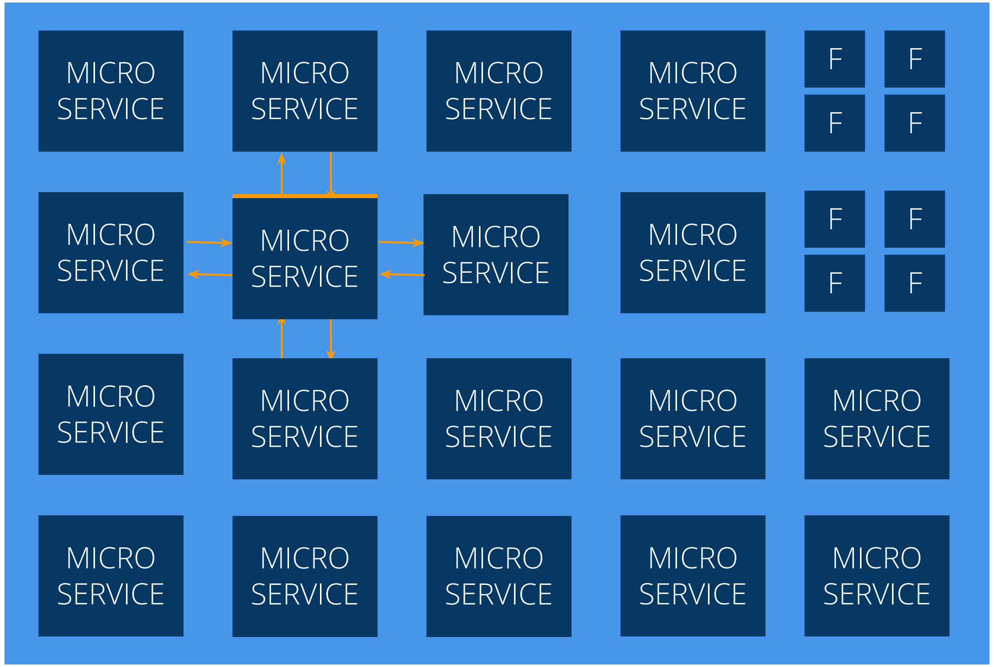
Cloud-Native Applications

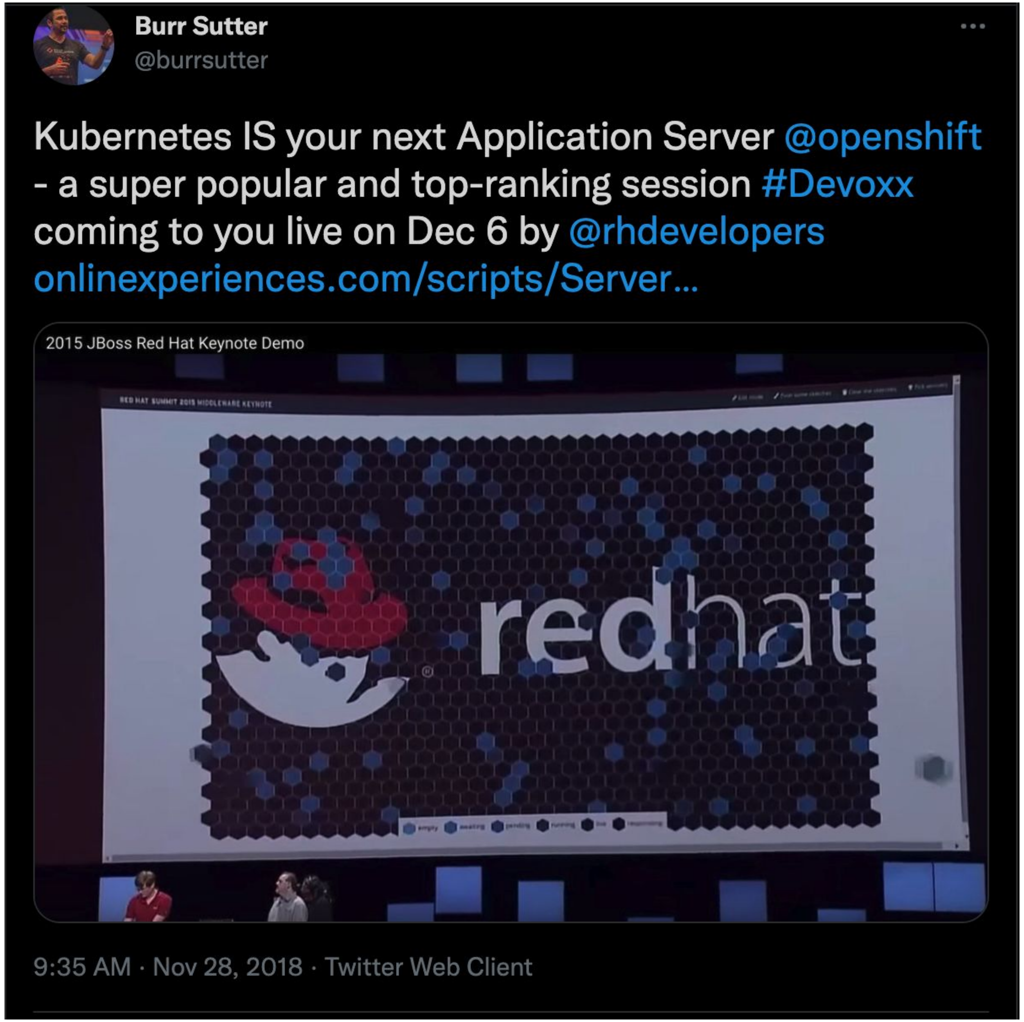
- Small, independent, and loosely coupled services
  - Microservices
- Container based
- Allows rapidly iteration to deliver business value
- Private, public, and hybrid clouds
- Scalable, resource efficient

New Architectures



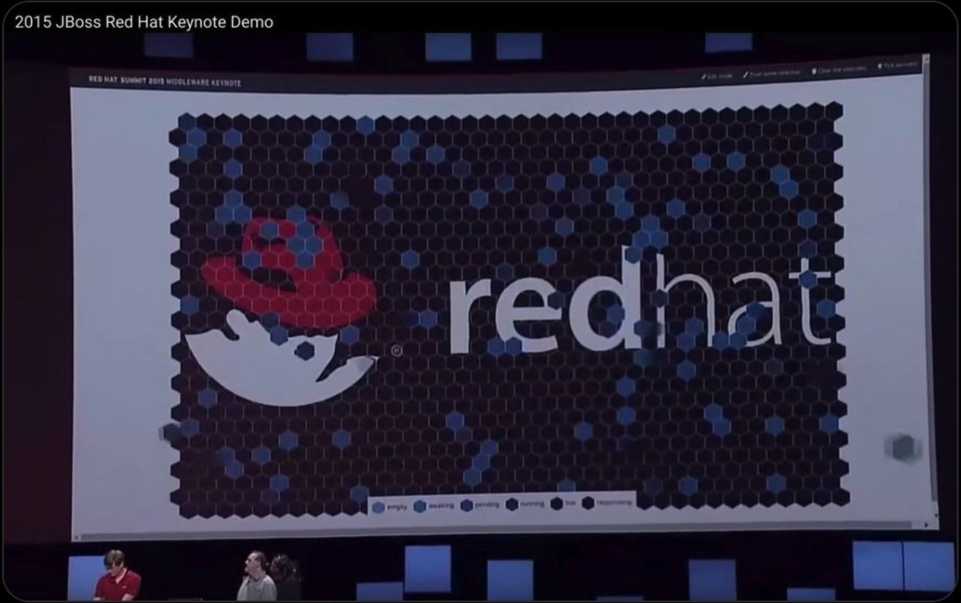
New Architectures





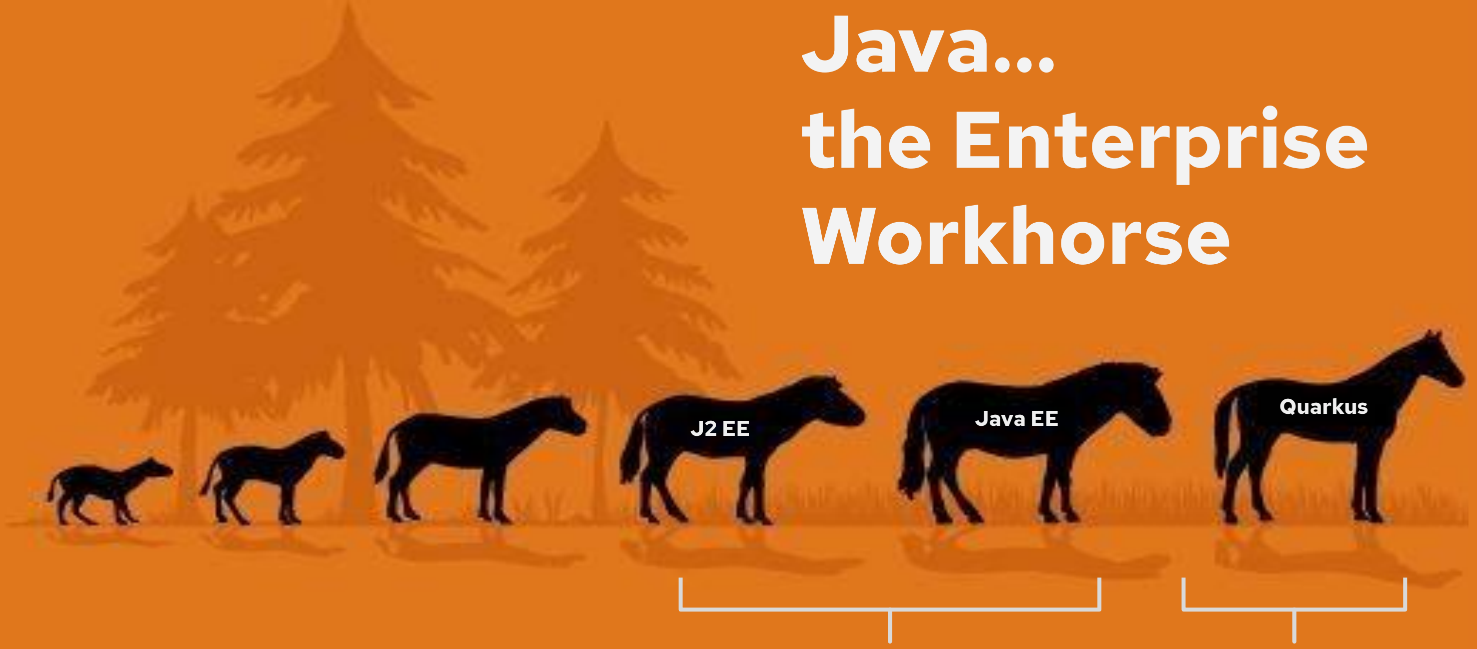
Burr Sutter
@burrsutter

Kubernetes IS your next Application Server @openshift - a super popular and top-ranking session #Devoxx coming to you live on Dec 6 by @rhdevelopers onlinexperiences.com/scripts/Server…


2015 JBoss Red Hat Keynote Demo
RED HAT SUMMIT 2015 MIDDLEWARE KEYNOTE
redhat

9:35 AM · Nov 28, 2018 · Twitter Web Client

Java...
the Enterprise Workhorse



Monolith

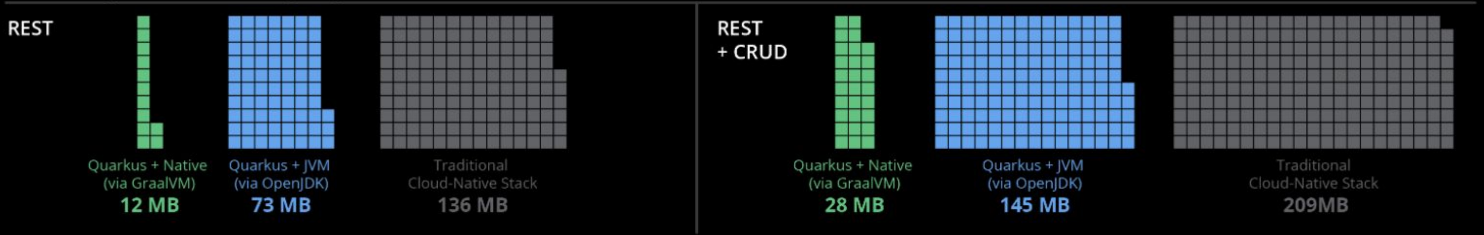
Microservices

Cloud-Native

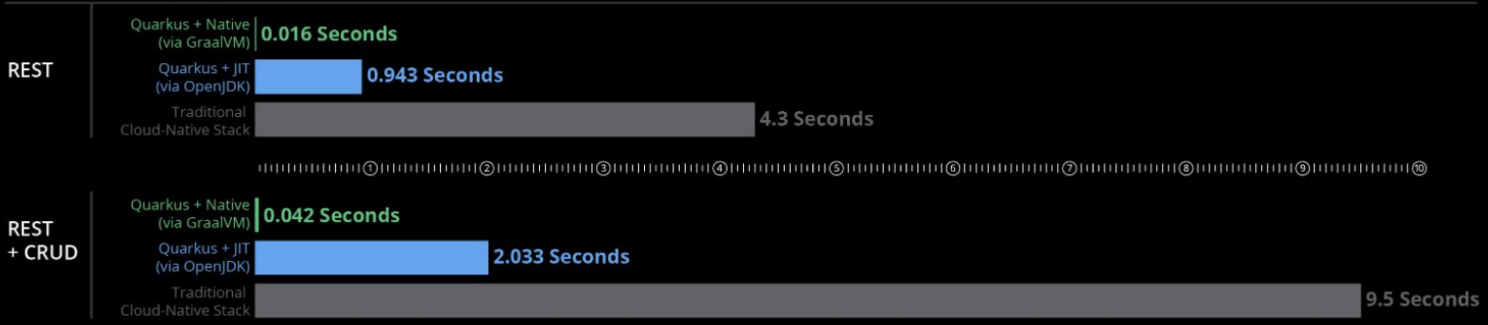
```
$ ./my-native-java-rest-app
Quarkus started in 0.008s
```

Memory (RSS) in Megabytes*

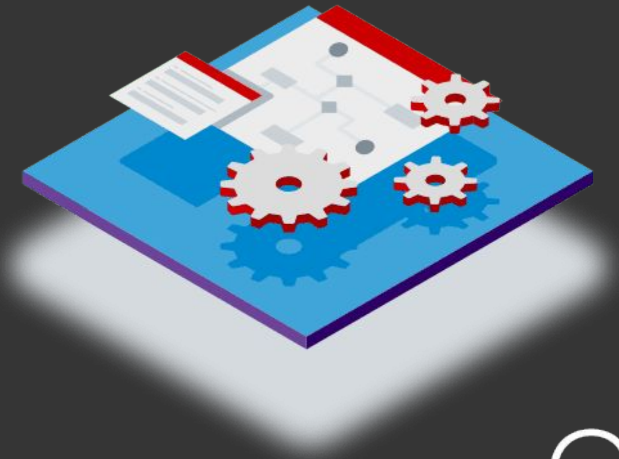
*Tested on a single-core machine



BOOT + First Response Time



KIE



jBPM

Drools

Kogito

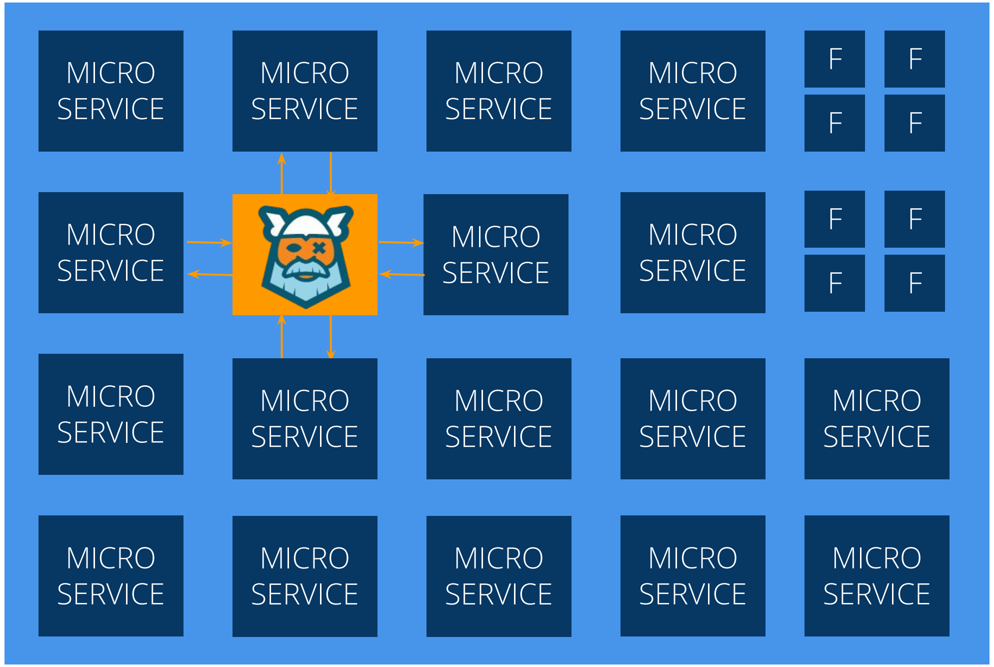
OptaPlanner

Dashbuilder

Red Hat

Kogito ergo automate
CLOUD-NATIVE BUSINESS AUTOMATION FOR BUILDING INTELLIGENT APPLICATIONS, BACKED BY BATTLE-TESTED CAPABILITIES.

New Architectures



What does this means to DashBuilder?

DashBuilder V8 - Iteration 1 (May 21)

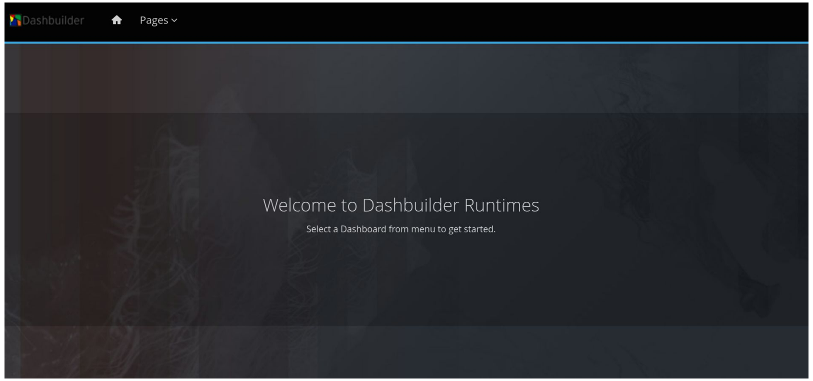
Cloud Native DashBuilder

Quarkus Based

What we've learned and built

Cloud Native DashBuilder

DashBuilder Runtime is a **immutable** and **stateless** cloud native web application that can run dashboards



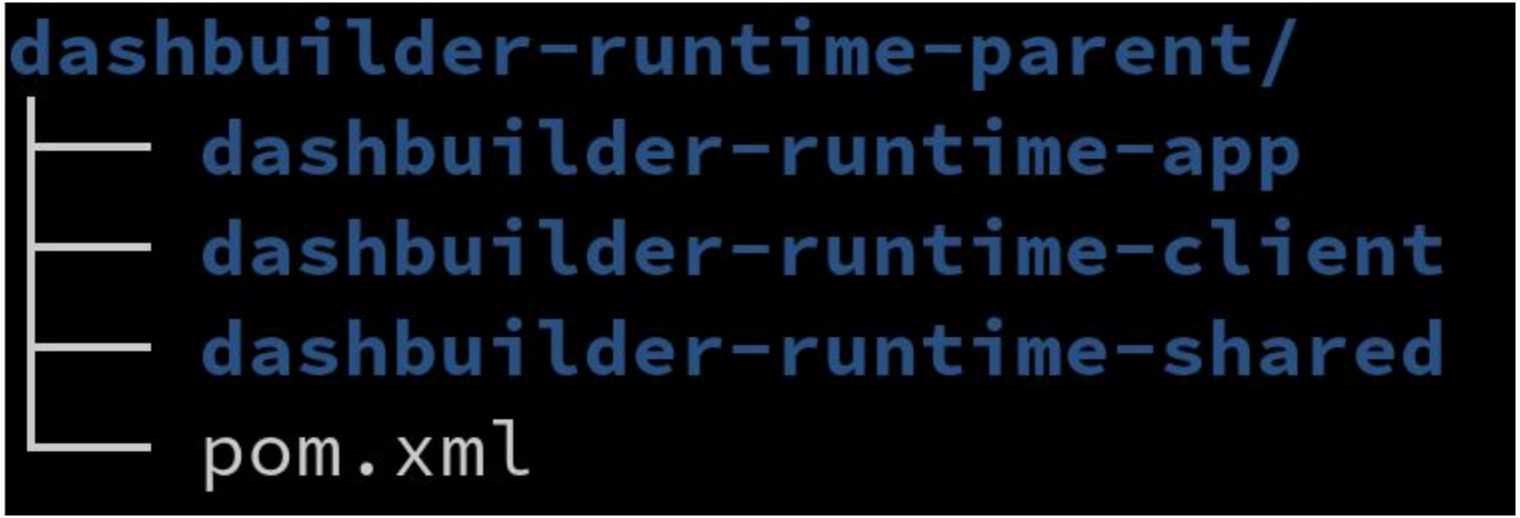
Welcome to Dashbuilder Runtimes

Select a Dashboard from menu to get started.

Quarkus Migration

Key points to consider

- Change project Structure
  - Errai RPC to REST endpoints
- Native compilation
- Java 8 to Java 11
- Security



```
dashbuilder-runtime-parent/
├── dashbuilder-runtime-app
├── dashbuilder-runtime-client
├── dashbuilder-runtime-shared
└── pom.xml
```

Moving to Quarkus

Java EE to Quarkus

- CDI and JAX-RS: Supported by Microprofile
  - If the API is implemented on Microprofile, it's super easy
- EJB replaced by CDI
  - Executors were replaced by Microprofile Context Propagation
  - https://download.eclipse.org/microprofile/microprofile-context-propagation-1.0/microprofile-context-propagation.html
- Application Lifecycle annotations replaced by Quarkus lifecycle
  - Observes CDI event: ***ShutdownEvent*** and ***StartupEvent*** from ***io.quarkus.runtime***

EJB

```java
import javax.enterprise.concurrent.ManagedExecutorService;

@Startup // internal annotation
@ApplicationScoped
public class RuntimeModelWatcherServiceManager {

    @Resource
    private ManagedExecutorService executorService;

    @PostConstruct
    public void start() {
      executorService.execute(() -> {
          // async work
      });
    }
}
```

Microprofile

```java
import org.eclipse.microprofile.context.ManagedExecutor;
import java.util.concurrent.Future;

@ApplicationScoped
public class RuntimeModelWatcherServiceManager {

    @Inject
    ManagedExecutor executor;

    private Future<?> watcherTask;

    @PostConstruct
    public void start(@Observes StartupEvent startupEvent) {
      watcherTask = executor.submit(() -> {
          // async work
      });
    }
}
```

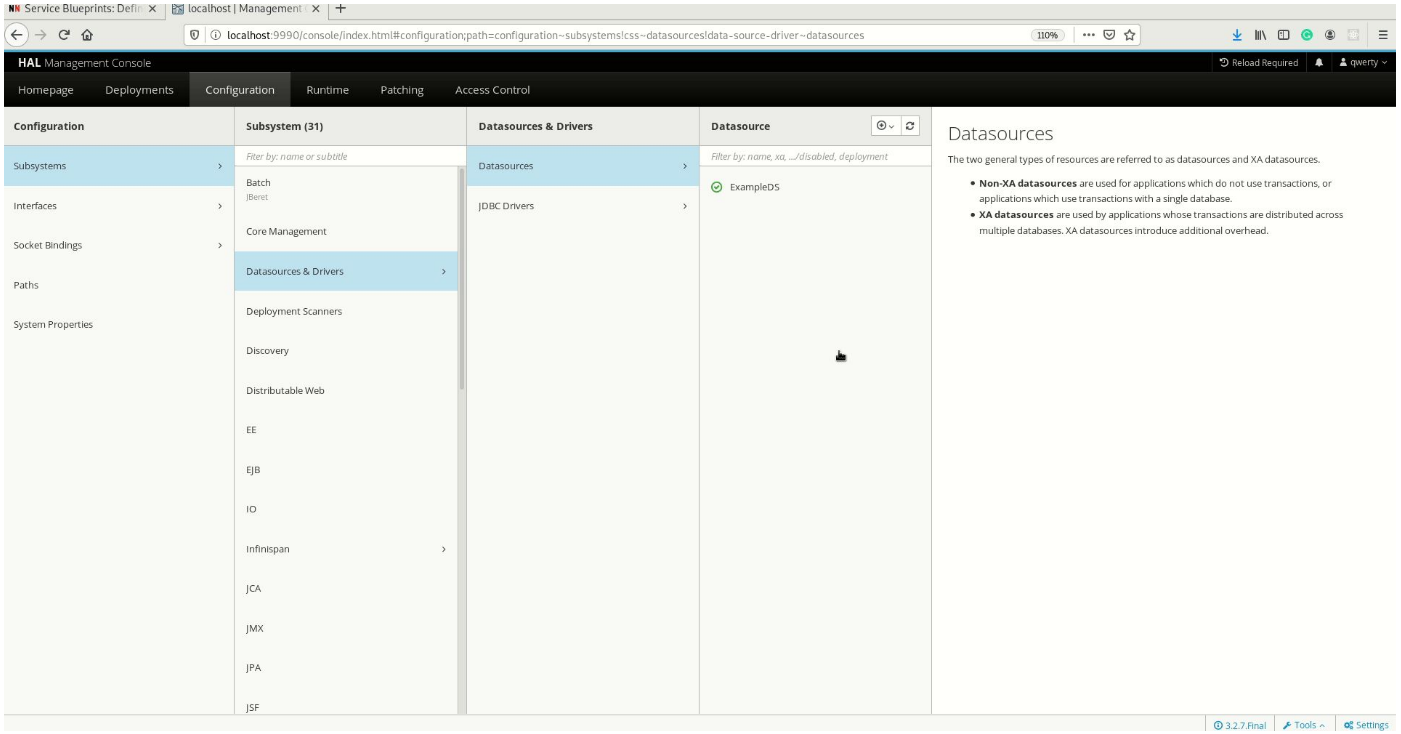
Red Hat

Moving to Quarkus

Database

- Data sets of type SQL should be able to access databases
    - https://blog.kie.org/2021/07/add-sql-datasource-for-authoring-dashboards.html
- In 7.x we used Wildfly datasources configuration
- Using Quarkus supported drivers we had to create a way to configure datasources
    - http://fxapps.blogspot.com/2021/06/database-query-server-using-quarkus.html

Adding WIldfly Datasource


Service Blueprints: Defin
localhost | Management
localhost:9990/console/index.html#configuration;path=configuration~subsystems!css~datasources!data-source-driver~datasources

HAL Management Console
Reload Required
qwerty
Homepage Deployments Configuration Runtime Patching Access Control

Configuration
Subsystems
Interfaces
Socket Bindings
Paths
System Properties

Subsystem (31)
Filter by: name or subtitle
Batch
JBeret
Core Management
Datasources & Drivers
Deployment Scanners
Discovery
Distributable Web
EE
EJB
IO
Infinispan
JCA
JMX
JPA
JSF

Datasources & Drivers
Datasources
JDBC Drivers

Datasource
Filter by: name, xa, .../disabled, deployment
ExampleDS

Datasources

The two general types of resources are referred to as datasources and XA datasources.

- **Non-XA datasources** are used for applications which do not use transactions, or applications which use transactions with a single database.
- **XA datasources** are used by applications whose transactions are distributed across multiple databases. XA datasources introduce additional overhead.

3.2.7.Final Tools Settings

Or using configuration files or JBoss CLI

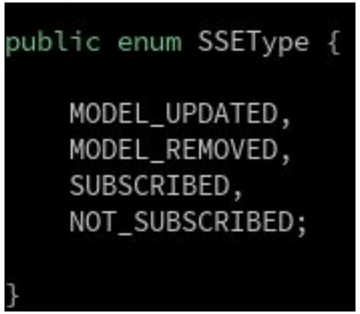
```
java \
-Ddashbuilder.datasources=sample \
-Ddashbuilder.datasource.sample.jdbcUrl={JDBC connection URL} \
-Ddashbuilder.datasource.sample.providerClassName={driver class name} \
-Ddashbuilder.datasource.sample.maxSize=10 \
-Ddashbuilder.datasource.sample.principal={user} \
-Ddashbuilder.datasource.sample.credential={password} \
 -jar dashbuilder-runtime-app-8.0.0.Alpha.jar
```

Name “sample” should match datasource name as in dataset configuration

Moving to Quarkus

Dev Mode

- Previous framework broadcast server-side events to client for “free”
- Quarkus: Server-Sent Event from JAX-RS
- Protocol to keep client up to date

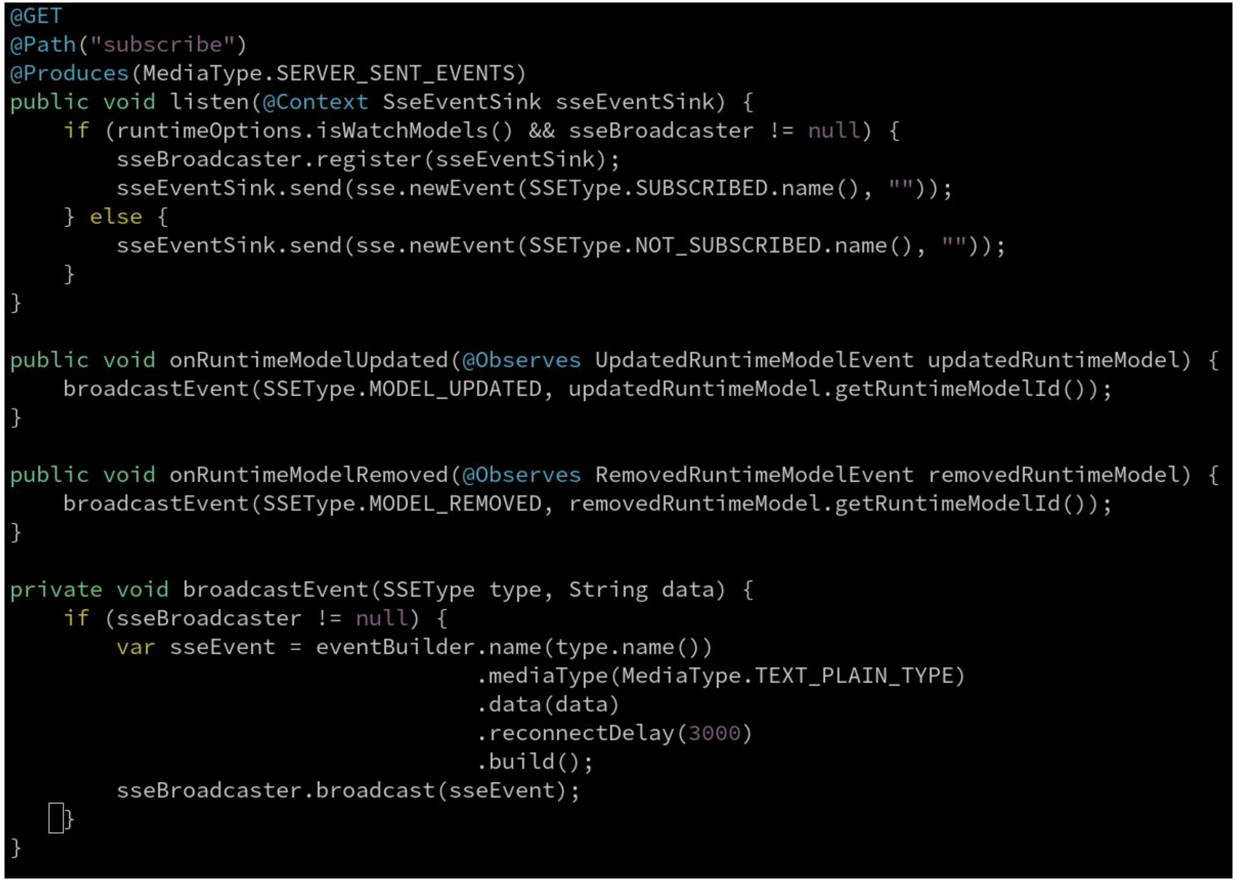


```
public enum SSEType {

    MODEL_UPDATED,
    MODEL_REMOVED,
    SUBSCRIBED,
    NOT_SUBSCRIBED;

}
```

Listen for changes: server side



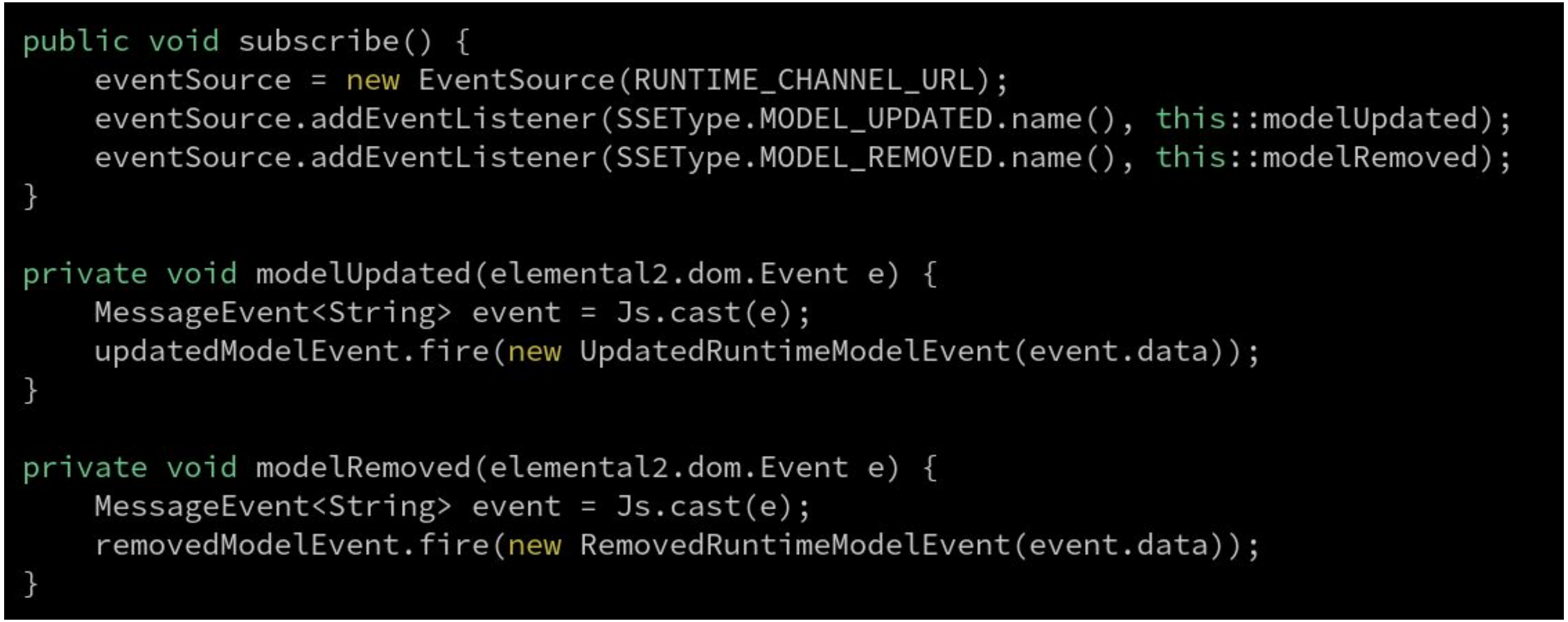
```java
@GET
@Path("subscribe")
@Produces(MediaType.SERVER_SENT_EVENTS)
public void listen(@Context SseEventSink sseEventSink) {
    if (runtimeOptions.isWatchModels() && sseBroadcaster != null) {
        sseBroadcaster.register(sseEventSink);
        sseEventSink.send(sse.newEvent(SSEType.SUBSCRIBED.name(), ""));
    } else {
        sseEventSink.send(sse.newEvent(SSEType.NOT_SUBSCRIBED.name(), ""));
    }
}

public void onRuntimeModelUpdated(@Observes UpdatedRuntimeModelEvent updatedRuntimeModel) {
    broadcastEvent(SSEType.MODEL_UPDATED, updatedRuntimeModel.getRuntimeModelId());
}

public void onRuntimeModelRemoved(@Observes RemovedRuntimeModelEvent removedRuntimeModel) {
    broadcastEvent(SSEType.MODEL_REMOVED, removedRuntimeModel.getRuntimeModelId());
}

private void broadcastEvent(SSEType type, String data) {
    if (sseBroadcaster != null) {
        var sseEvent = eventBuilder.name(type.name())
                                   .mediaType(MediaType.TEXT_PLAIN_TYPE)
                                   .data(data)
                                   .reconnectDelay(3000)
                                   .build();
        sseBroadcaster.broadcast(sseEvent);
    }
}
```

Listen for changes: client side



```java
public void subscribe() {
    eventSource = new EventSource(RUNTIME_CHANNEL_URL);
    eventSource.addEventListener(SSEType.MODEL_UPDATED.name(), this::modelUpdated);
    eventSource.addEventListener(SSEType.MODEL_REMOVED.name(), this::modelRemoved);
}

private void modelUpdated(elemental2.dom.Event e) {
    MessageEvent<String> event = Js.cast(e);
    updatedModelEvent.fire(new UpdatedRuntimeModelEvent(event.data));
}

private void modelRemoved(elemental2.dom.Event e) {
    MessageEvent<String> event = Js.cast(e);
    removedModelEvent.fire(new RemovedRuntimeModelEvent(event.data));
}
```

Moving to Quarkus

- Servlet used to compress all resources served by the application
- Quarkus: Compression enabled by a system property!

quarkus.http.enable-compression=true

Red Hat

Moving to Quarkus

- Multiple alternatives to package the application
- Simple executable JAR that can run with ***java -jar*** command

quarkus.package.type=uber-jar

Moving to Quarkus

Possible use of other Quarkus APIs

- Microprofile Configuration
  - Currently with Java System properties
- Microprofile Health
  - Implemented with JAX-RS
- Microprofile Metrics

Is it required to move everything to Quarkus APIs?

More code changes, more test (retest)

Demo

- WAR Deployment startup
  - Screenshot
- Quarkus Startup
  - REST endpoints calls
- Dev Mode

Red Hat

Summary

Pros

- Performance
- Smaller and easier to install
- Modern
- Cloud native
- Easy to go from Java EE -> Microprofile

Cons

- No more delegation of some features (e.g. SQL datasources)
- Project structure more complex (3 projects)

Dashbuilder V8 - Iteration 2 (Set 21)

Cloud Native Dashbuilder "v8" (mid 2020)

External Data Sources

YML based Cloud native deployments

New External Components

DashBuilder site revamp

A lot of new exciting features

A lot of opportunities to get involved with Open Source

DashBuilder

TRY IT NOW !!!

- Getting started
  - DashBuilder new features:
    1. https://blog.kie.org/2021/04/dashbuilder-an-apache-licensed-business-reporting-and-monitoring-tool.html
  - DashBuilder Getting Started
    1. https://blog.kie.org/2021/05/dashbuilder-getting-started-guide.html
  - Demos and samples
    1. https://github.com/jesuino/dashbuilder-dashboards
    2. https://github.com/jesuino/dashbuilder-docker
- Community
  - Chat http://kie.zulipchat.com/ #tooling channel
  - Blog https://blog.kie.org/
- Twitter
  - @ederign @william_antonio
- GitHub, JIRA
- Documentation
- Events

More on: Google for "Kie Live DashBuilder"

Thank you

Eder Ignatowicz

Principal Software Engineer

@ederign

William Siqueira

Senior Software Engineer

@william_antonio

linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat