

Drools & jBPM

Workshop 2013

October · London

Agenda

- Day 1: Tooling Technology Stack
 - Architectural Overview
 - UberFire
- Day 2: Tooling
 - KIE Platform Intro
 - Drools Tooling
 - jBPM Tooling

Short Disclaimer

The tooling that we will be showing is a SNAPSHOT/Nightly build version of the project master branch. It can be built downloading the community projects hosted in github.
com/droolsjbpm.

If you are using Windows, you can expect some issues that are still being fixed before the final release.

Technology Stack

- GWT
- GWT-Bootstrap
- Errai
- Uberfire
- Contributing modules
 - guvnor
 - kie-wb-common
 - jBPM Console
 - jBPM Process Designer
 - jBPM Form Modeller
 - Drools Workbench

Stack - GWT

<http://www.gwtproject.org/>

- Legacy components
- Cross browser support
- Widget set
- GWT Bootstrap

Stack - GWT-Bootstrap

<http://gwtbootstrap.github.io/>

- Integration Bootstrap with GWT
- Common widget set
- Striving for better consistency

Stack - Errai

<http://errai.github.io/>

- Client side CDI
- Federated Event Bus
- Remote services
- Marshalling
- WELD

Stack - Uberfire

<http://droolsjbpm.github.io/uberfire/>

- Workbench
- Annotation driven development
- Unified Security API
- Virtual File System (Java nio2 backport)
- Indexed resources
- Concurrency support

Stack - Uberfire

- Workbench
 - WorkbenchPerspective
 - WorkbenchPanel
 - WorkbenchPart
- PanelManager
- Drag & Drop
- JavaScript plugins
- State persistence

Stack - Uberfire

- Workbench
 - WorkbenchPanel
 - WorkbenchPart
- PlaceRequest
 - @WorkbenchPerspective
 - @WorkbenchScreen
 - @WorkbenchEditor
 - @WorkbenchPopup

Stack - Uberfire

- Annotations

- @WorkbenchPerspective
- @WorkbenchScreen
- @WorkbenchEditor
- @WorkbenchPopup

Stack - Uberfire

- Annotations - @WorkbenchPerspective
 - @Perspective
 - @OnStartup
 - @OnOpen
 - @OnClose

Stack - Uberfire

- Annotations - @WorkbenchScreen
 - @WorkbenchPartTitle
 - @WorkbenchPartView (or implement IsWidget)
 - @OnStartup
 - @OnOpen
 - @OnMayClose
 - @OnClose
 - @OnFocus
 - @OnLostFocus

Stack - Uberfire

- Annotations - @WorkbenchEditor
 - @WorkbenchPartTitle
 - @WorkbenchPartView (or implement IsWidget)
 - @OnStartup
 - @OnOpen
 - @OnMayClose
 - @OnClose
 - @OnFocus
 - @OnLostFocus
 - @IsDirty
 - @OnSave

Stack - Uberfire

- Annotations - @WorkbenchEditor
 - Path
 - ResourceTypeDefinition
 - ClientResourceType

Stack - Uberfire

- Annotations - @WorkbenchPopup
 - @WorkbenchPartTitle
 - @WorkbenchPartView (or extend PopupPanel)
 - @OnOpen

Stack - Uberfire

- Workbench - Menus and Toolbars
- Hierarchical and contextual
 - Application
 - @WorkbenchPerspective
 - @WorkbenchScreen
 - @WorkbenchEditor
 - @WorkbenchMenu
 - @WorkbenchToolBar

Stack - Uberfire

- Unified Security API

- @Roles
- @All
- @Deny
- RuntimeResource
 - MenuItem
 - ToolBar / ToolBarItem
 - OrganizationalUnit
 - Repository
 - Project
 - Activity

Stack - Uberfire

- Virtual File System

- Java 7 NIO2 backport
 - git
- Clustering - Apache Zookeeper / Helix
- IOWatchService
- Meta Data

Stack - Uberfire

- Indexed Resources

- Lucene
- IOSearchService
- DublinCoreAttributes

Stack - Uberfire

- Concurrency support
 - ResourceAddedEvent
 - ResourceUpdatedEvent
 - ResourceDeletedEvent
 - ResourceRenamedEvent
- Client-side optimistic lock support
 - ObservablePath

Creating a new Screen

```
@WorkbenchScreen(identifier = "org.jboss.example.screen")
public class ScreenPresenter {
    ...
    @Inject
    public ScreenPresenter.View view;

    @OnStartup
    public void onStartup( final PlaceRequest placeRequest ) {
        ...
    }

    @WorkbenchPartTitle
    public String getTitle() {
        return "My title";
    }

    @WorkbenchPartView
    public IWidget getWidget() {
        return view;
    }
}
```

Creating a new Editor

```
@WorkbenchEditor(identifier = "org.jboss.example.editor", supportedTypes =
{MyResourceType.class} )
public class EditorPresenter {

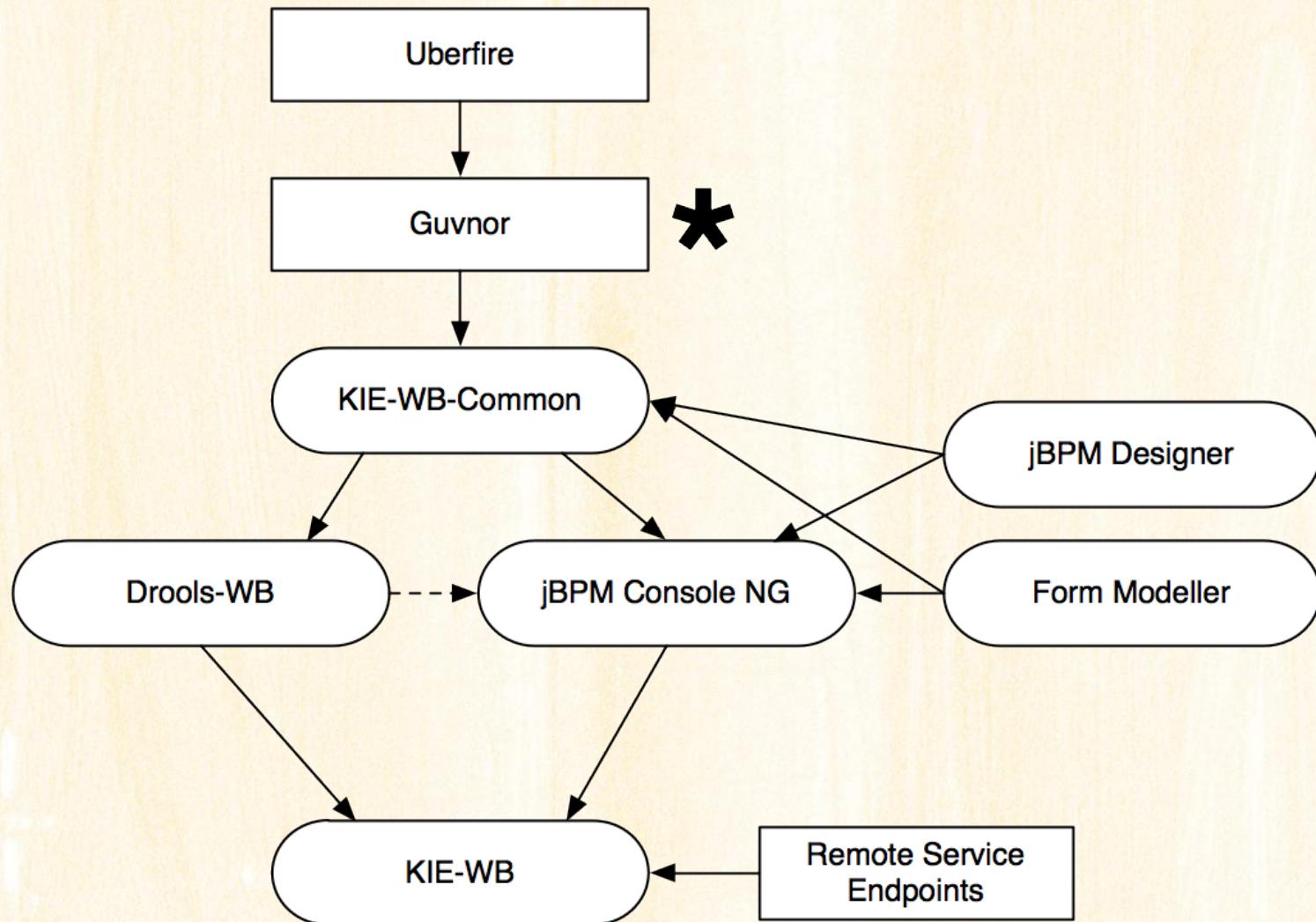
    ...
    @Inject
    public EditorPresenter.View view;

    @OnStartup
    public void onStartup( final Path path,
                          final PlaceRequest placeRequest ) {
        //load content from Path
    }

    @IsDirty
    public boolean isDirty() {
        return isDirty;
    }

    @OnSave
    public void onSave() {
        //Save content to Path
    }
}
```

Modules

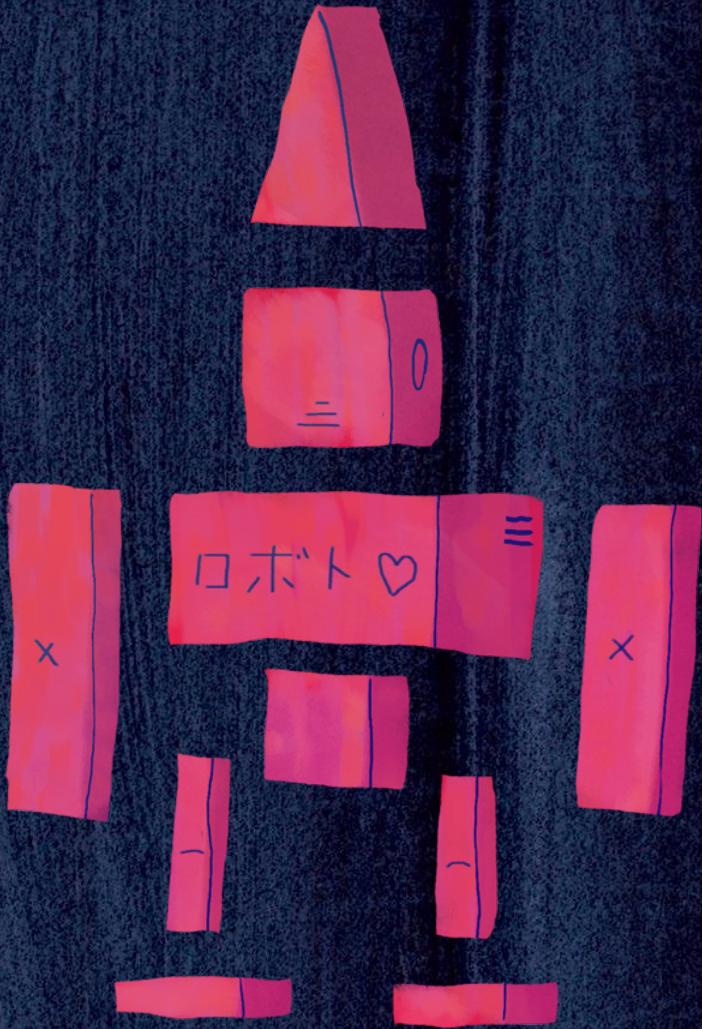


Modules

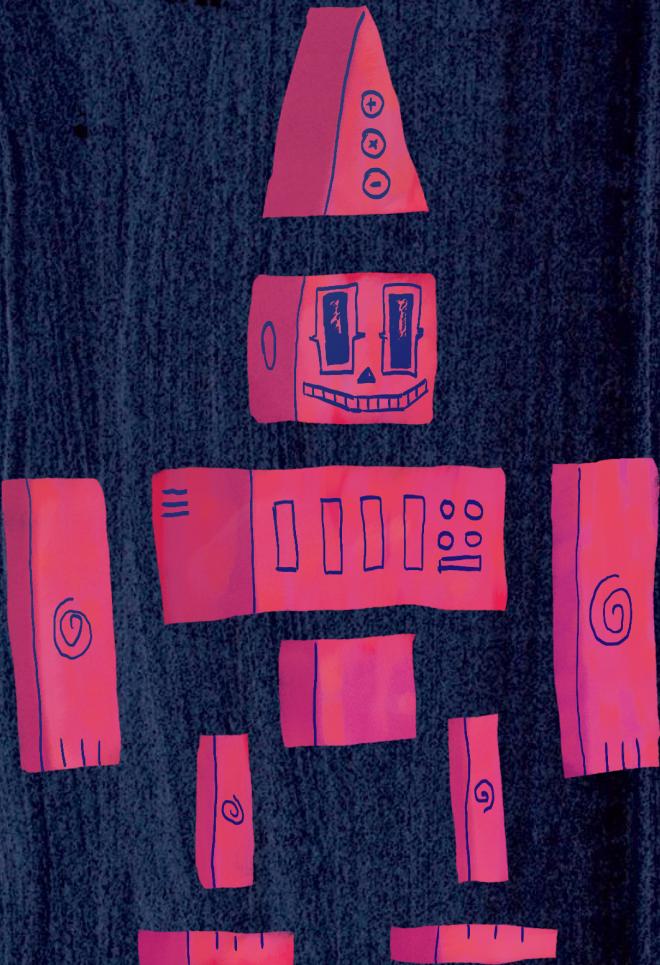
- Guvnor
- KIE-WB-Common
- jBPM Console NG
- jBPM Designer
- Form Modeller
- Drools Workbench
- KIE Workbench

Modules - Demo of drools-wb

Exercise - Write a screen



See you
tomorrow



Drools & jBPM

Workshop 2013

October · London

Day 2 Agenda: KIE = Drools + jBPM

- KIE Platform introduction
 - KIE Jars/ KIE Projects
 - kmodule.xml
 - Example Project
 - Tests
- jBPM Tooling
 - Creating a KIE Project
 - Deploying a Kjar into the platform runtime
 - Executing a business process
 - Rest APIs

Short Disclaimer

The tooling that we will be showing is a SNAPSHOT/Nightly build version of the project master branch. It can be built downloading the community projects hosted in github.
com/droolsjbpm.

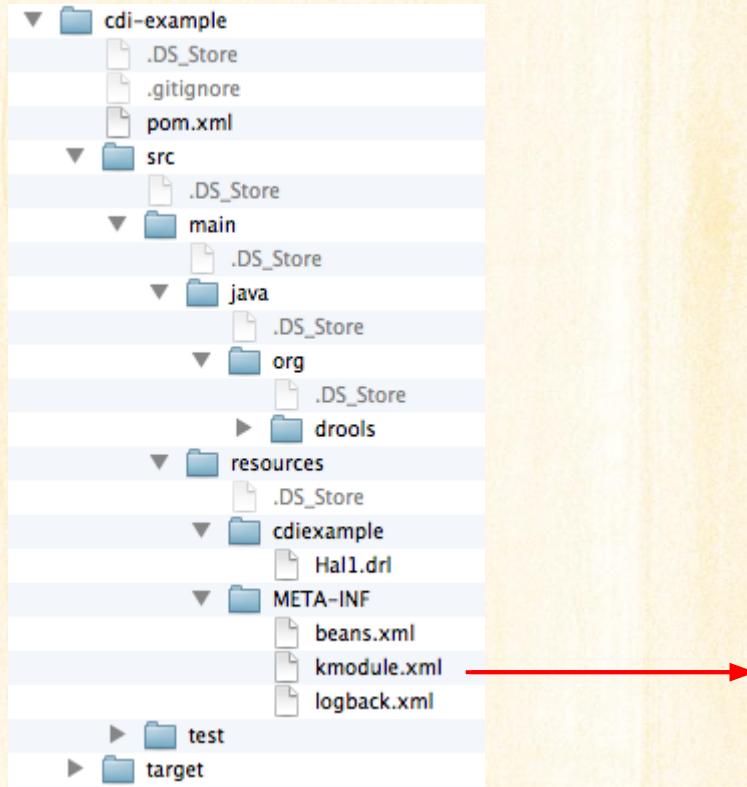
If you are using Windows, you can expect some issues that are still being fixed before the final release.

New Things

- KIE prefix
 - Stands for ***Knowledge is Everything***
 - Replaces the Knowledge* prefixes and packages
 - A background compatibility jar will be provided for migration
- KIE Projects
 - Provides the Drools and jBPM projects structure
 - Maven based / standard project structure
 - Integrated with CDI
 - Defines a Deployment Units for jBPM

CDI + Drools + KIE Project

- From: <https://github.com/droolsjbpm/drools/tree/master/drools-examples-cdi>



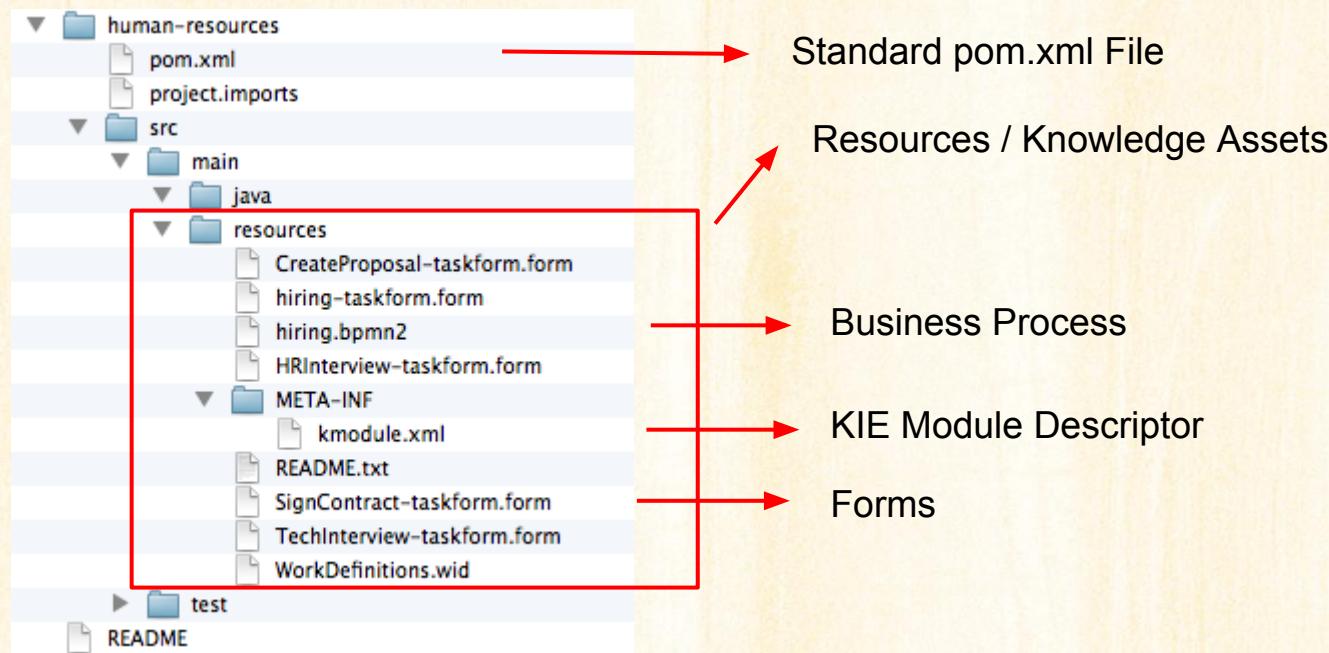
```
<kmodule
    xmlns="http://jboss.org/kie/6.0.0/kmodule">

    <kbase name="cdiexample">
        <ksession name="ksession1"/>
    </kbase>

</kmodule>
```

KIE Project

- Structure (Maven Standard + kmodule.xml)



KModule

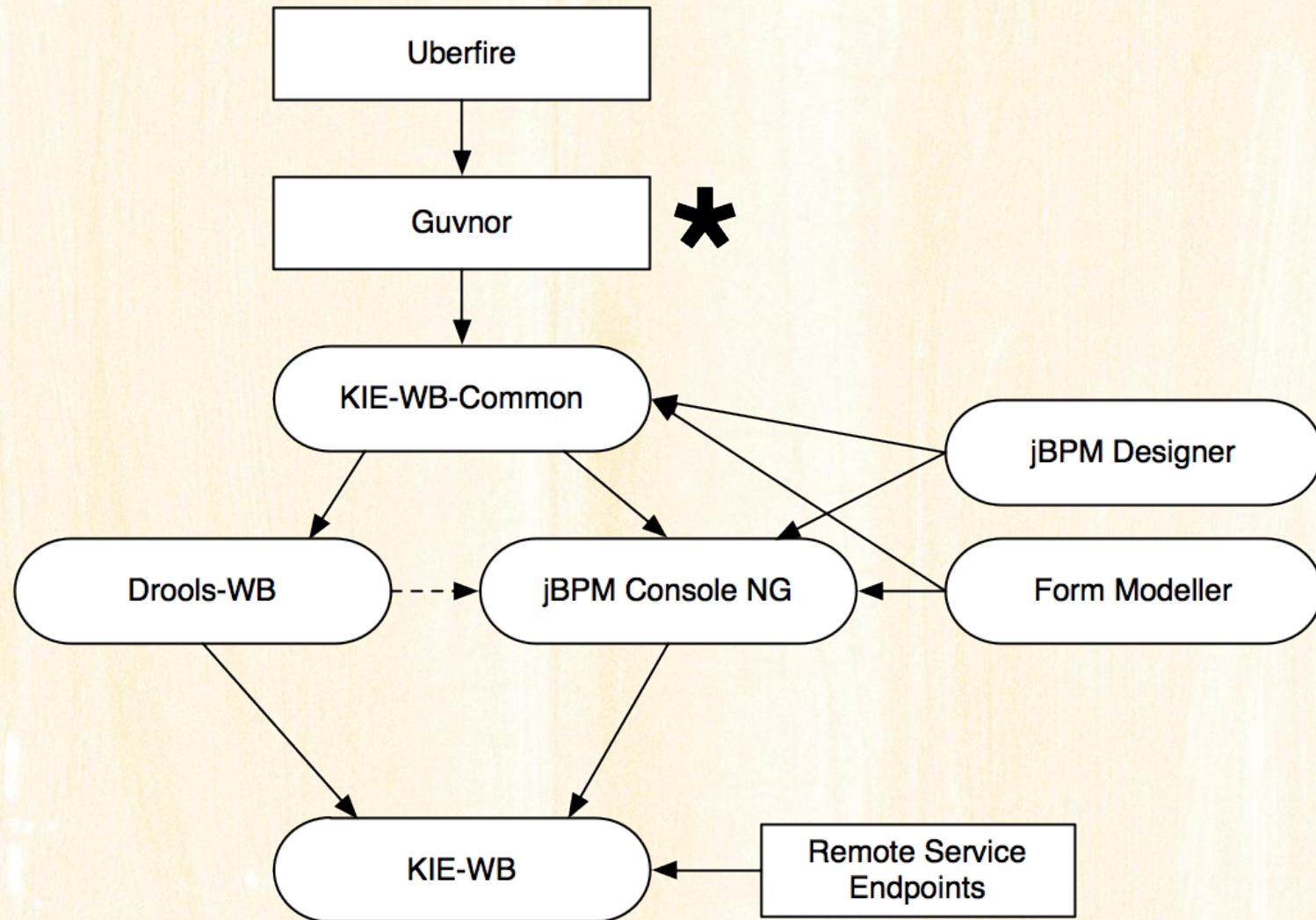
- Convention Driven

```
1 <kmodule xmlns="http://jboss.org/kie/6.0.0/kmodule"
2   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
3
4 </kmodule>
5
```

- Or you can add custom configurations

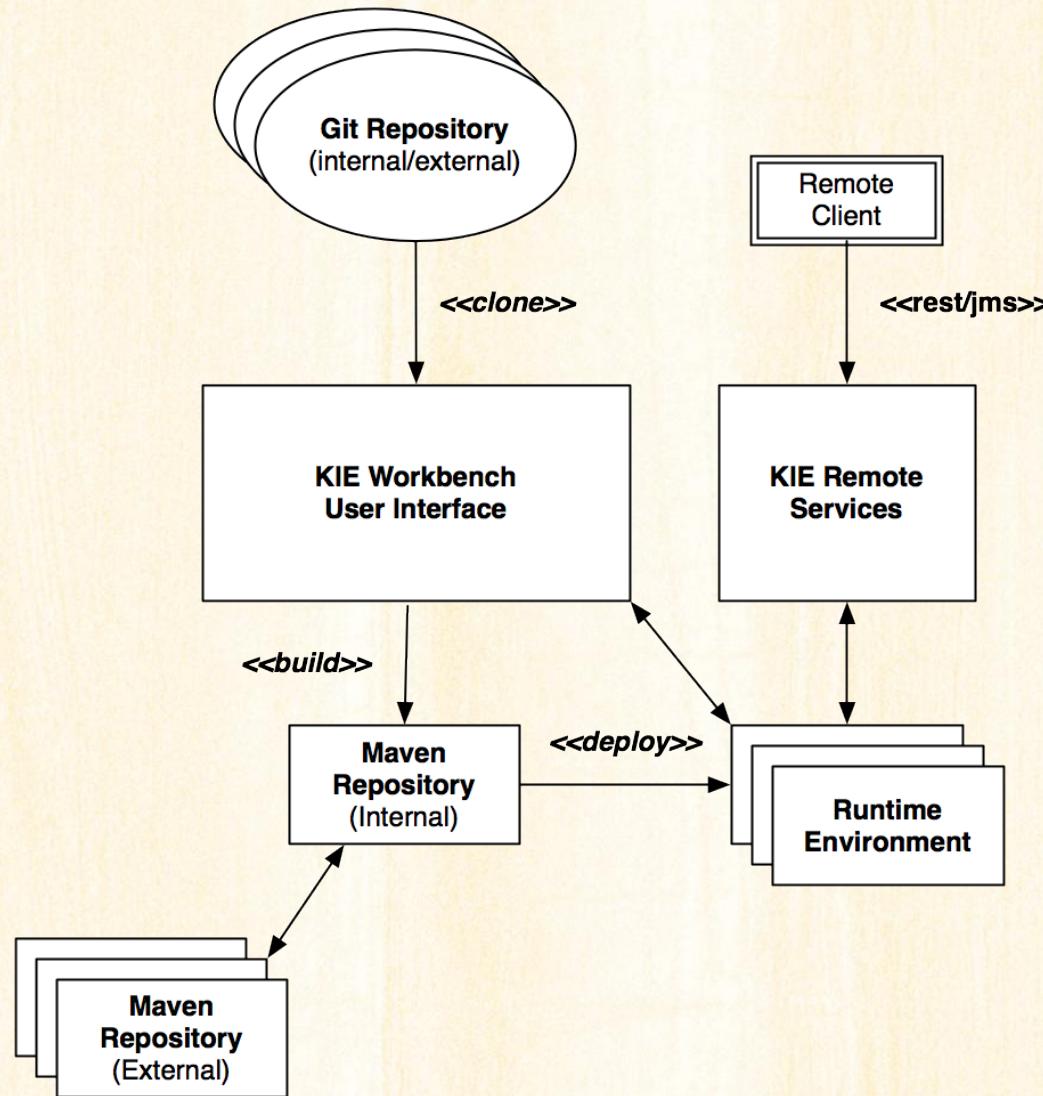
```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <kmodule xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xmlns="http://jboss.org/kie/6.0.0/kmodule">
4
5   <kbase name="org.kie.kbase1" equalsBehavior="equality" eventProcessingMode="stream">
6     <ksession name="org.kie.kbase1.ksession1" type="stateful" clockType="pseudo"/>
7     <ksession name="org.kie.kbase1.ksession2" type="stateless" clockType="realtime"/>
8   </kbase>
9
10  <kbase name="org.kie.kbase2" equalsBehavior="equality" eventProcessingMode="stream">
11    </kbase>
12
13  <kbase name="org.kie.kbase3" scope="SessionScoped" equalsBehavior="equality" eventProcessingMode="stream">
14    <ksession name="org.kie.kbase3.ksession1" scope="javax.enterprise.context.ApplicationScoped" type="stateful">
15      <ksession name="org.kie.kbase3.ksession2" scope="SessionScoped" type="stateful" clockType="realtime"/>
16    </kbase>
17 </kmodule>
```

Applications and Modules Distributions



KIE Workbench Components

Interactions



BPM Side Agenda

- Home
- Authoring Perspective
 - Modelling a process
 - Modelling forms
 - Registering Domain Specific Tasks
- Building & Deploying a Project
- Starting a Process Instance
- Task Management
- General Configurations
- REST endpoints

Home

- The Knowledge Lifecycle
 - Authoring
 - Runtime Configurations
 - Process Management
 - Task Management
 - Business Activity Monitoring

The Knowledge Life Cycle

Authoring
[Project Authoring](#)
[Asset repository](#)
[Administration](#)

Deploy
[Deployments](#)
[Jobs](#)

Process Management
[Process Definitions](#)
[Process Instances](#)

Tasks
[Tasks List](#)

Dashboards
[Process & Task Dashboard](#)
[Business Dashboards](#)

Authoring Perspective

- Authoring Assets
 - Creating KIE Projects
 - Creating Knowledge Assets
- Administration
 - Creating a new repository
 - Cloning an existing repository
 - <https://github.com/droolsjbpm/jbpm-playground>

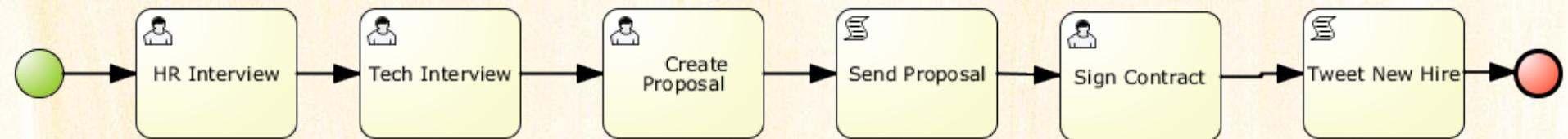
Building & Deploying a Project

- Project Editor
 - Project Dependencies
 - Build & Deploy
 - Deployment Perspective

Human Resources Example

From: <https://github.com/droolsjbpm/jbpm-playground>

Article: <http://salaboy.com/2013/10/11/using-the-jbpm-console-ng-hr-example/>

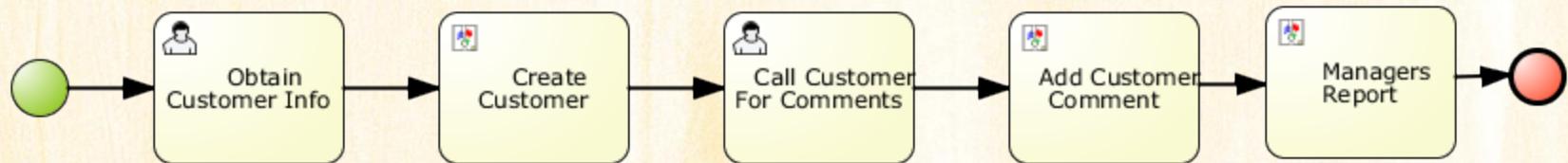


Customer Relationships Example

From: <https://github.com/droolsjbpm/jbpm-playground>

Uses: <http://www.service-repository.com/operation/operations?id=30>

Article: <http://salaboy.com/2013/10/22/kie-wb-jbpm-console-ng-configurations/>



Process Management

- Process Definitions
 - Shows only Deployed Process Definitions
 - Create new Process Instances
- Process Instances
 - You can inspect the details of your process instance
 - You can look into the Process Variables (simple types for now)

Task Management

- Task Lists
 - Work in your assigned tasks
 - Group Tasks
 - Personal Tasks
- Task Forms
 - FTLs -> Freemarker templates
 - Forms -> Graphically Designed Forms using Form Modeller

Business Activity Monitoring

- Dashboard
 - Generic Business Dashboard
 - Business Process Dashboard
 - Task Dashboard

General Configurations

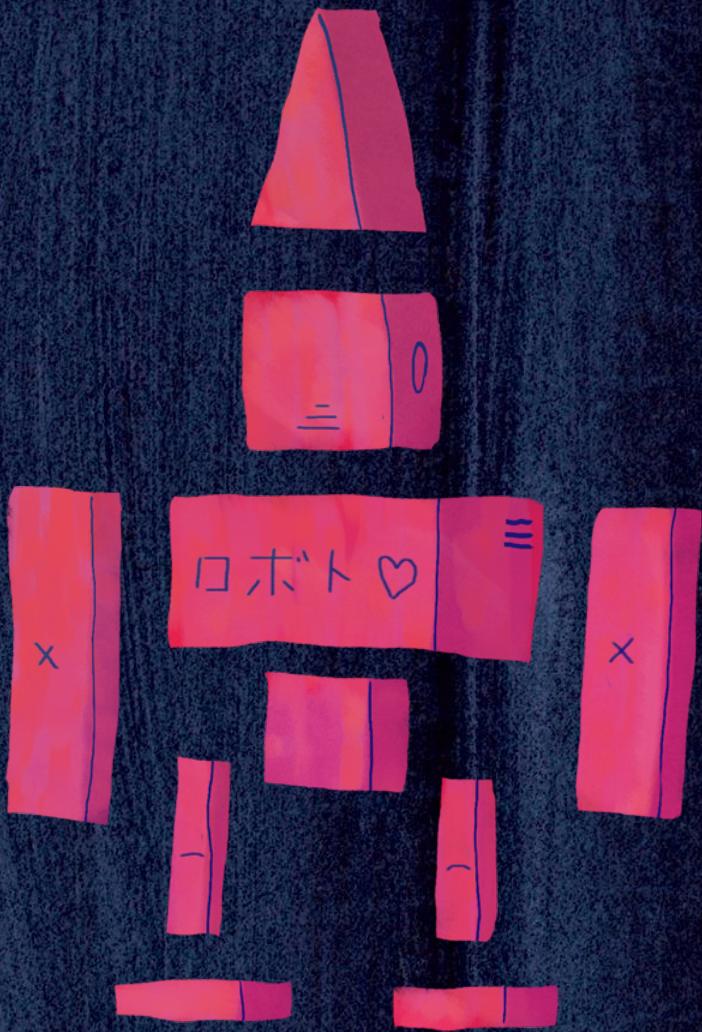
- User/Groups/Roles
 - Where should I configure it?
 - Look here: <http://salaboy.com/2013/10/22/kie-wb-jbpm-console-ng-configurations/>
- Work Item Handlers (Service Connectors)
 - What do I need?
 - Look here: <http://salaboy.com/2013/10/22/kie-wb-jbpm-console-ng-configurations/>
- Getting access to the GIT repositories
 - How do I access to the local GIT repository?
 - Let's do it!

REST Endpoints

- For the BPM Side you can interact with the Process Runtimes using a REST endpoint (<https://github.com/droolsjbpm/droolsjbpm-integration/wiki>)
- You can execute your operations against a KIE WB running instance
- Project: ***jbpm-rest-client*** also from the jbpm-playground repo

Community / Further Reading

- IRC: irc.freenode.net
 - #drools
 - #jbpm
- jBPM Forums / Drools Mailing List
 - www.drools.org and www.jbpm.org
- Blogs
 - blog.athico.com
 - kverlaen.blogspot.com
 - mswiderski.blogspot.com
 - salaboy.com



Good Bye