



Red Hat Process Automation Manager 7.12

Designing your decision management
architecture for Red Hat Process Automation
Manager

Red Hat Process Automation Manager 7.12 Designing your decision management architecture for Red Hat Process Automation Manager

Legal Notice

Copyright © 2022 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

This document describes basic concepts and options for a Red Hat Process Automation Manager implementation to consider when designing your decision management architecture. To verify the most suitable implementation for your environment, contact your Red Hat Technical Account Manager or Solutions Architect.

Table of Contents

PREFACE	3
MAKING OPEN SOURCE MORE INCLUSIVE	4
CHAPTER 1. INSTALLATION ENVIRONMENT OPTIONS FOR RED HAT PROCESS AUTOMATION MANAGER .	5
CHAPTER 2. DECISION-AUTHORING ASSETS IN RED HAT PROCESS AUTOMATION MANAGER	6
CHAPTER 3. PROJECT STORAGE AND BUILD OPTIONS WITH RED HAT PROCESS AUTOMATION MANAGER	10
CHAPTER 4. PROJECT DEPLOYMENT OPTIONS WITH RED HAT PROCESS AUTOMATION MANAGER ..	13
CHAPTER 5. ASSET EXECUTION OPTIONS WITH RED HAT PROCESS AUTOMATION MANAGER	15
CHAPTER 6. EXAMPLE DECISION MANAGEMENT ARCHITECTURES WITH RED HAT PROCESS AUTOMATION MANAGER	16
CHAPTER 7. ADDITIONAL RESOURCES	21
APPENDIX A. VERSIONING INFORMATION	22
APPENDIX B. CONTACT INFORMATION	23

PREFACE

As a business analyst, system administrator, or business rules developer, the types of Red Hat Process Automation Manager business assets that you create and the way in which you develop, store, execute, and deploy those assets varies based on the needs of your business automation environment. You can design a decision management architecture that uses Business Central or an external integrated development environment (IDE) for asset development, a built-in or external Git repository for storage and version control, a KIE Server or embedded application for deployment, or other implementation combinations. You can also deploy Red Hat Process Automation Manager into a Red Hat OpenShift Container Platform environment for an enhanced containerized implementation.

This document describes basic concepts and options for a Red Hat Process Automation Manager implementation to consider when designing your decision management architecture. To verify the most suitable implementation for your environment, contact your Red Hat Technical Account Manager or Solutions Architect.

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see [our CTO Chris Wright's message](#).

CHAPTER 1. INSTALLATION ENVIRONMENT OPTIONS FOR RED HAT PROCESS AUTOMATION MANAGER

With Red Hat Process Automation Manager, you can set up a development environment to develop business applications, a runtime environment to run those applications to support decisions, or both.

- **Development environment:** Typically consists of one Business Central installation and at least one KIE Server installation. You can use Business Central to design decisions and other artifacts, and you can use KIE Server to execute and test the artifacts that you created.
- **Runtime environment:** Consists of one or more KIE Server instances with or without Business Central. Business Central has an embedded Process Automation Manager controller. If you install Business Central, use the **Menu → Deploy → Execution servers** page to create and maintain containers. If you want to automate KIE Server management without Business Central, you can use the headless Process Automation Manager controller.

You can also cluster both development and runtime environments. A clustered development or runtime environment consists of a unified group or cluster of two or more servers. The primary benefit of clustering Red Hat Process Automation Manager development environments is high availability and enhanced collaboration, while the primary benefit of clustering Red Hat Process Automation Manager runtime environments is high availability and load balancing. High availability decreases the chance of data loss when a single server fails. When a server fails, another server fills the gap by providing a copy of the data that was on the failed server. When the failed server comes online again, it resumes its place in the cluster.



NOTE

Clustering of the runtime environment is currently supported on Red Hat JBoss EAP 7.4 and Red Hat OpenShift Container Platform only.

Additional resources

- [Planning a Red Hat Process Automation Manager installation](#)
- [Red Hat Process Automation Manager 7 Supported Configurations](#)
- [Red Hat Process Automation Manager 7 Component Details](#)

CHAPTER 2. DECISION-AUTHORING ASSETS IN RED HAT PROCESS AUTOMATION MANAGER

Red Hat Process Automation Manager supports several assets that you can use to define business decisions for your decision service. Each decision-authoring asset has different advantages, and you might prefer to use one or a combination of multiple assets depending on your goals and needs.

The following table highlights the main decision-authoring assets supported in Red Hat Process Automation Manager projects to help you decide or confirm the best method for defining decisions in your decision service.

Table 2.1. Decision-authoring assets supported in Red Hat Process Automation Manager

Asset	Highlights	Authoring tools	Documentation
Decision Model and Notation (DMN) models	<ul style="list-style-type: none"> • Are decision models based on a notation standard defined by the Object Management Group (OMG) • Use graphical decision requirements diagrams (DRDs) that represent part or all of the overall decision requirements graph (DRG) to trace business decision flows • Use an XML schema that allows the DMN models to be shared between DMN-compliant platforms • Support Friendly Enough Expression Language (FEEL) to define decision logic in DMN decision tables and other DMN boxed expressions • Can be integrated efficiently with Business Process Model and Notation (BPMN) process models • Are optimal for creating comprehensive, illustrative, and stable decision flows 	Business Central or other DMN-compliant editor	Designing a decision service using DMN models

Asset	Highlights	Authoring tools	Documentation
Guided decision tables	<ul style="list-style-type: none"> ● Are tables of rules that you create in a UI-based table designer in Business Central ● Are a wizard-led alternative to spreadsheet decision tables ● Provide fields and options for acceptable input ● Support template keys and values for creating rule templates ● Support hit policies, real-time validation, and other additional features not supported in other assets ● Are optimal for creating rules in a controlled tabular format to minimize compilation errors 	Business Central	Designing a decision service using guided decision tables
Spreadsheet decision tables	<ul style="list-style-type: none"> ● Are XLS or XLSX spreadsheet decision tables that you can upload into Business Central ● Support template keys and values for creating rule templates ● Are optimal for creating rules in decision tables already managed outside of Business Central ● Have strict syntax requirements for rules to be compiled properly when uploaded 	Spreadsheet editor	Designing a decision service using spreadsheet decision tables
Guided rules	<ul style="list-style-type: none"> ● Are individual rules that you create in a UI-based rule designer in Business Central ● Provide fields and options for acceptable input ● Are optimal for creating single rules in a controlled format to minimize compilation errors 	Business Central	Designing a decision service using guided rules

Asset	Highlights	Authoring tools	Documentation
Guided rule templates	<ul style="list-style-type: none"> ● Are reusable rule structures that you create in a UI-based template designer in Business Central ● Provide fields and options for acceptable input ● Support template keys and values for creating rule templates (fundamental to the purpose of this asset) ● Are optimal for creating many rules with the same rule structure but with different defined field values 	Business Central	Designing a decision service using guided rule templates
DRL rules	<ul style="list-style-type: none"> ● Are individual rules that you define directly in .drl text files ● Provide the most flexibility for defining rules and other technicalities of rule behavior ● Can be created in certain standalone environments and integrated with Red Hat Process Automation Manager ● Are optimal for creating rules that require advanced DRL options ● Have strict syntax requirements for rules to be compiled properly 	Business Central or integrated development environment (IDE)	Designing a decision service using DRL rules

Asset	Highlights	Authoring tools	Documentation
Predictive Model Markup Language (PMML) models	<ul style="list-style-type: none"> ● Are predictive data-analytic models based on a notation standard defined by the Data Mining Group (DMG) ● Use an XML schema that allows the PMML models to be shared between PMML-compliant platforms ● Support Regression, Scorecard, Tree, Mining, and other model types ● Can be included with a standalone Red Hat Process Automation Manager project or imported into a project in Business Central ● Are optimal for incorporating predictive data into decision services in Red Hat Process Automation Manager 	PMML or XML editor	Designing a decision service using PMML models

When you define business decisions, you can also consider using Red Hat build of Kogito for your cloud-native decision services. For more information about getting started with Red Hat build of Kogito microservices, see [Getting started with Red Hat build of Kogito in Red Hat Process Automation Manager](#) .

CHAPTER 3. PROJECT STORAGE AND BUILD OPTIONS WITH RED HAT PROCESS AUTOMATION MANAGER

As you develop a Red Hat Process Automation Manager project, you need to be able to track the versions of your project with a version-controlled repository, manage your project assets in a stable environment, and build your project for testing and deployment. You can use Business Central for all of these tasks, or use a combination of Business Central and external tools and repositories. Red Hat Process Automation Manager supports Git repositories for project version control, Apache Maven for project management, and a variety of Maven-based, Java-based, or custom-tool-based build options.

The following options are the main methods for Red Hat Process Automation Manager project versioning, storage, and building:

Table 3.1. Project version control options (Git)

Versioning option	Description	Documentation
Business Central Git VFS	Business Central contains a built-in Git Virtual File System (VFS) that stores all processes, rules, and other artifacts that you create in the authoring environment. Git is a distributed version control system that implements revisions as commit objects. When you commit your changes into a repository, a new commit object in the Git repository is created. When you create a project in Business Central, the project is added to the Git repository connected to Business Central.	NA
External Git repository	If you have Red Hat Process Automation Manager projects in Git repositories outside of Business Central, you can import them into Red Hat Process Automation Manager spaces and use Git hooks to synchronize the internal and external Git repositories.	Managing projects in Business Central

Table 3.2. Project management options (Maven)

Management option	Description	Documentation
-------------------	-------------	---------------


Management option	Description	Documentation
Business Central Maven repository	<p>Business Central contains a built-in Maven repository that organizes and builds project assets that you create in the authoring environment. Maven is a distributed build-automation tool that uses repositories to store Java libraries, plug-ins, and other build artifacts. When building projects and archetypes, Maven dynamically retrieves Java libraries and Maven plug-ins from local or remote repositories to promote shared dependencies across projects.</p> <div style="display: flex; align-items: center;">  <div> <p>NOTE</p> <p>For a production environment, consider using an external Maven repository configured with Business Central.</p> </div> </div>	NA
External Maven repository	<p>If you have Red Hat Process Automation Manager projects in an external Maven repository, such as Nexus or Artifactory, you can create a settings.xml file with connection details and add that file path to the kie.maven.settings.custom property in your project standalone-full.xml file.</p>	<p>Maven Settings Reference</p> <p>Packaging and deploying a Red Hat Process Automation Manager project</p>

Table 3.3. Project build options

Build option	Description	Documentation
Business Central (KJAR)	<p>Business Central builds Red Hat Process Automation Manager projects stored in either the built-in Maven repository or a configured external Maven repository. Projects in Business Central are packaged automatically as knowledge JAR (KJAR) files with all components needed for deployment when you build the projects.</p>	<p>Packaging and deploying a Red Hat Process Automation Manager project</p>
Standalone Maven project (KJAR)	<p>If you have a standalone Red Hat Process Automation Manager Maven project outside of Business Central, you can edit the project pom.xml file to package your project as a KJAR file, and then add a kmodule.xml file with the KIE base and KIE session configurations needed to build the project.</p>	<p>Packaging and deploying a Red Hat Process Automation Manager project</p>

Build option	Description	Documentation
Embedded Java application (KJAR)	If you have an embedded Java application from which you want to build your Red Hat Process Automation Manager project, you can use a KieModuleModel instance to programmatically create a kmodule.xml file with the KIE base and KIE session configurations, and then add all resources in your project to the KIE virtual file system KieFileSystem to build the project.	Packaging and deploying a Red Hat Process Automation Manager project
CI/CD tool (KJAR)	If you use a tool for continuous integration and continuous delivery (CI/CD), you can configure the tool set to integrate with your Red Hat Process Automation Manager Git repositories to build a specified project. Ensure that your projects are packaged and built as KJAR files to ensure optimal deployment.	NA
S2I in OpenShift (container image)	If you use Red Hat Process Automation Manager on Red Hat OpenShift Container Platform, you can build your Red Hat Process Automation Manager projects as KJAR files in the typical way or use Source-to-Image (S2I) to build your projects as container images. S2I is a framework and a tool that allows you to write images that use the application source code as an input and produce a new image that runs the assembled application as an output. The main advantage of using the S2I tool for building reproducible container images is the ease of use for developers.	Creating Images in OpenShift

CHAPTER 4. PROJECT DEPLOYMENT OPTIONS WITH RED HAT PROCESS AUTOMATION MANAGER

After you develop, test, and build your Red Hat Process Automation Manager project, you can deploy the project to begin using the business assets you have created. You can deploy a Red Hat Process Automation Manager project to a configured KIE Server, to an embedded Java application, or into a Red Hat OpenShift Container Platform environment for an enhanced containerized implementation.

The following options are the main methods for Red Hat Process Automation Manager project deployment:

Table 4.1. Project deployment options

Deployment option	Description	Documentation
Deployment to an OpenShift environment	Red Hat OpenShift Container Platform combines Docker and Kubernetes and enables you to create and manage containers. You can install both Business Central and KIE Server on OpenShift. Red Hat Process Automation Manager provides templates that you can use to deploy a Red Hat Process Automation Manager authoring environment, managed server environment, immutable server environment, or trial environment on OpenShift. With OpenShift, components of Red Hat Process Automation Manager are deployed as separate OpenShift pods. You can scale each of the pods up and down individually, providing as few or as many containers as necessary for a particular component. You can use standard OpenShift methods to manage the pods and balance the load.	Deploying a Red Hat Process Automation Manager environment on Red Hat OpenShift Container Platform 4 using Operators Deploying a Red Hat Process Automation Manager environment on Red Hat OpenShift Container Platform 3 using templates
Deployment to KIE Server	KIE Server is the server provided with Red Hat Process Automation Manager that runs the decision services, process applications, and other deployable assets from a packaged and deployed Red Hat Process Automation Manager project (KJAR file). These services are consumed at run time through an instantiated KIE container, or <i>deployment unit</i> . You can deploy and maintain deployment units in KIE Server using Business Central or using a headless Process Automation Manager controller with its associated REST API (considered a <i>managed</i> KIE Server instance). You can also deploy and maintain deployment units using the KIE Server REST API or Java client API from a standalone Maven project, an embedded Java application, or other custom environment (considered an <i>unmanaged</i> KIE Server instance).	Packaging and deploying a Red Hat Process Automation Manager project Interacting with Red Hat Process Automation Manager using KIE APIs Managing and monitoring KIE Server

Deployment option	Description	Documentation
Deployment to an embedded Java application	If you want to deploy Red Hat Process Automation Manager projects to your own Java virtual machine (JVM) environment, microservice, or application server, you can bundle the application resources in the project WAR files to create a deployment unit similar to a KIE container. You can also use the core KIE APIs (not KIE Server APIs) to configure a KIE scanner to periodically update KIE containers.	KIE Public API

CHAPTER 5. ASSET EXECUTION OPTIONS WITH RED HAT PROCESS AUTOMATION MANAGER

After you build and deploy your Red Hat Process Automation Manager project to KIE Server or other environment, you can execute the deployed assets for testing or for runtime consumption. You can also execute assets locally in addition to or instead of executing them after deployment.

The following options are the main methods for Red Hat Process Automation Manager asset execution:

Table 5.1. Asset execution options

Execution option	Description	Documentation
Execution in KIE Server	If you deployed Red Hat Process Automation Manager project assets to KIE Server, you can use the KIE Server REST API or Java client API to execute and interact with the deployed assets. You can also use Business Central or the headless Process Automation Manager controller outside of Business Central to manage the configurations and KIE containers in the KIE Server instances associated with your deployed assets. For process definitions, you can use Business Central directly to execute process instances.	Interacting with Red Hat Process Automation Manager using KIE APIs
Execution in an embedded Java application	If you deployed Red Hat Process Automation Manager project assets in your own Java virtual machine (JVM) environment, microservice, or application server, you can use custom APIs or application interactions with core KIE APIs (not KIE Server APIs) to execute assets in the embedded engine.	KIE Public API
Execution in a local environment for extended testing	As part of your development cycle, you can execute assets locally to ensure that the assets you have created in Red Hat Process Automation Manager function as intended. You can use local execution in addition to or instead of executing assets after deployment.	"Executing rules" in Designing a decision service using DRL rules



SMART ROUTER (KIE SERVER ROUTER)

Depending on your deployment and execution environment, you can use a Smart Router to aggregate multiple independent KIE Server instances as though they are a single server. Smart Router is a single endpoint that can receive calls from client applications to any of your services and route each call automatically to the KIE Server that runs the service. For more information about Smart Router, see [Installing and configuring Red Hat Process Automation Manager in a Red Hat JBoss EAP clustered environment](#).

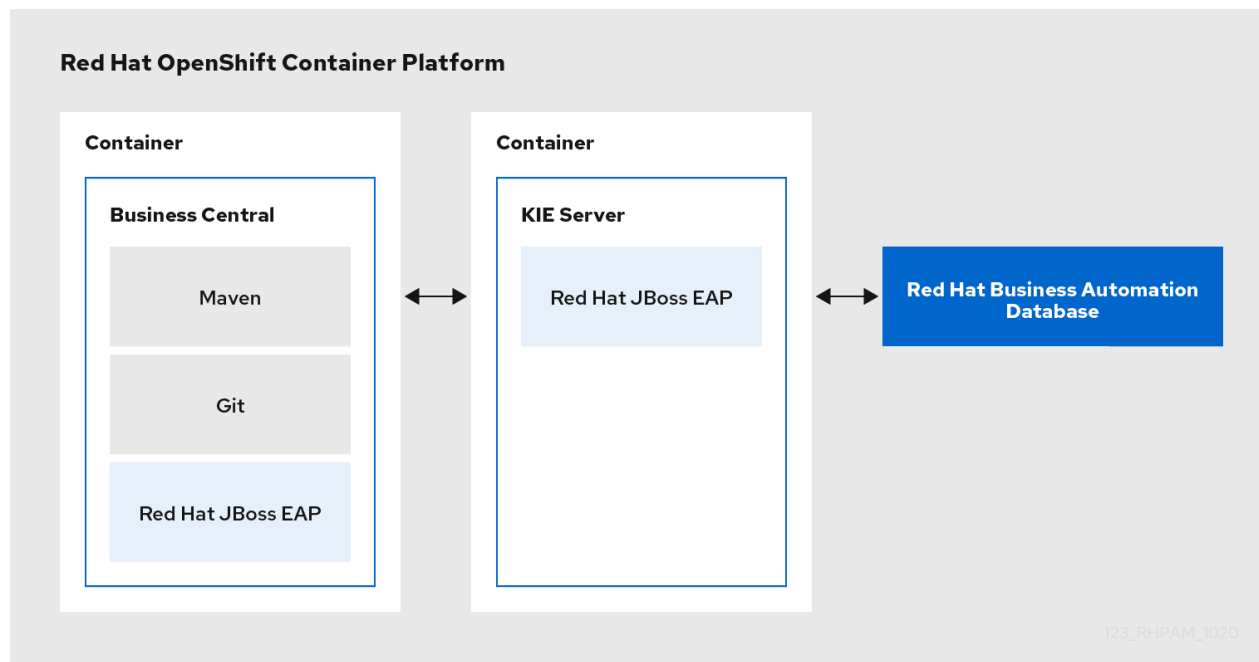
CHAPTER 6. EXAMPLE DECISION MANAGEMENT ARCHITECTURES WITH RED HAT PROCESS AUTOMATION MANAGER

The following scenarios illustrate common variations of Red Hat Process Automation Manager installation, asset authoring, project storage, project deployment, and asset execution in a decision management architecture. Each section summarizes the methods and tools used and the advantages for the given architecture. The examples are basic and are only a few of the many combinations you might consider, depending on your specific goals and needs with Red Hat Process Automation Manager.

Red Hat Process Automation Manager authoring and managed server environments on OpenShift with Business Central and KIE Server

- **Installation environment:** Red Hat Process Automation Manager on Red Hat OpenShift Container Platform, using the `rhcam712-authoring.yaml` and `rhcam712-kieserver.yaml` template files
- **Project storage and build environment:** External Git repository for project versioning synchronized with the Business Central Git repository using Git hooks, and external Maven repository for project management and building configured with KIE Server
- **Asset-authoring tool:** Business Central
- **Main asset types:** Decision Model and Notation (DMN) models for decisions and Business Process Model and Notation (BPMN) models for processes
- **Project deployment and execution environment:** KIE Server on Red Hat OpenShift Container Platform
- **Scenario advantages:**
 - Containerized implementation of Red Hat Process Automation Manager in a cloud-based development environment. Components of Red Hat Process Automation Manager are deployed as separate OpenShift pods that you can scale up and down individually, providing as few or as many containers as necessary for a particular component. You can use standard OpenShift methods to manage the pods and balance the load.
 - Access to the repositories, assets, asset designers, and project build options in Business Central.
 - Standardized asset-authoring approach using DMN and BPMN for optimal integration and stability.
 - Access to KIE Server functionality and KIE APIs for asset deployment and execution.

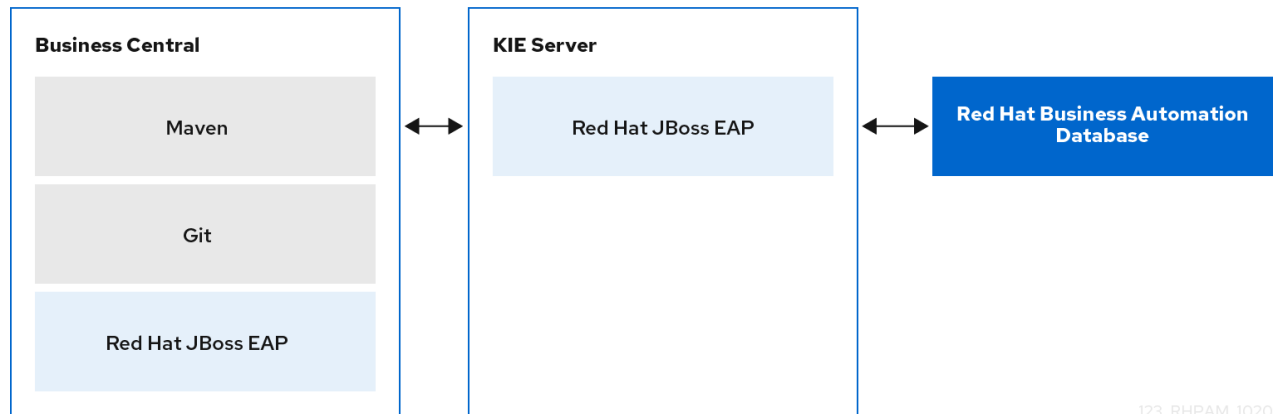
Figure 6.1. Red Hat Process Automation Manager on OpenShift with Business Central and KIE Server



Red Hat Process Automation Manager on Red Hat JBoss EAP with Business Central and KIE Server

- **Installation environment:** Red Hat Process Automation Manager on Red Hat JBoss Enterprise Application Platform
- **Project storage and build environment:** External Git repository for project versioning synchronized with the Business Central Git repository using Git hooks, and external Maven repository for project management and building configured with KIE Server
- **Asset-authoring tool:** Business Central
- **Main asset types:** Decision Model and Notation (DMN) models for decisions and Business Process Model and Notation (BPMN) models for processes
- **Project deployment and execution environment:** KIE Server
- **Scenario advantages:**
 - Stable implementation of Red Hat Process Automation Manager in an on-premise development environment
 - Access to the repositories, assets, asset designers, and project build options in Business Central
 - Standardized asset-authoring approach using DMN and BPMN for optimal integration and stability
 - Access to KIE Server functionality and KIE APIs for asset deployment and execution

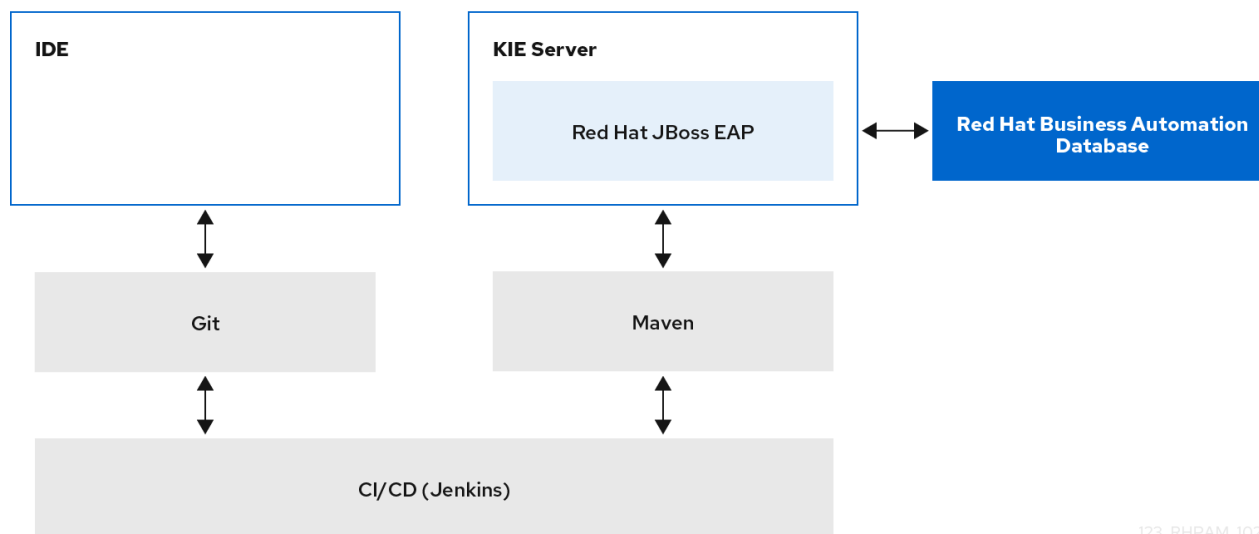
Figure 6.2. Red Hat Process Automation Manager on Red Hat JBoss EAP with Business Central and KIE Server



Red Hat Process Automation Manager on Red Hat JBoss EAP with an IDE and KIE Server

- **Installation environment:** Red Hat Process Automation Manager on Red Hat JBoss Enterprise Application Platform
- **Project storage and build environment:** External Git repository for project versioning (not synchronized with Business Central) and external Maven repository for project management and building configured with KIE Server
- **Asset-authoring tools:** Integrated development environment (IDE), such as Red Hat CodeReady Studio, and a spreadsheet editor or a Decision Model and Notation (DMN) modeling tool for other decision formats
- **Main asset types:** Drools Rule Language (DRL) rules, spreadsheet decision tables, and Decision Model and Notation (DMN) models for decisions, and Business Process Model and Notation (BPMN) models for processes
- **Project deployment and execution environment:** KIE Server
- **Scenario advantages:**
 - Flexible implementation of Red Hat Process Automation Manager in an on-premise development environment
 - Ability to define business assets using an external IDE and other asset-authoring tools of your choice
 - Access to KIE Server functionality and KIE APIs for asset deployment and execution

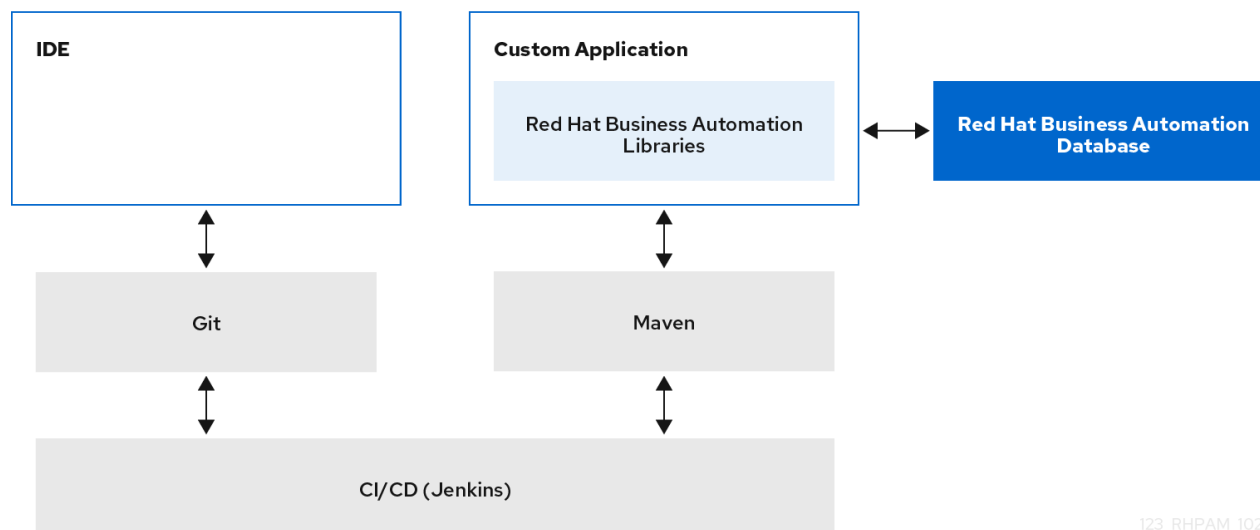
Figure 6.3. Red Hat Process Automation Manager on Red Hat JBoss EAP with an IDE and KIE Server



Red Hat Process Automation Manager with an IDE and an embedded Java application

- **Installation environment:** Red Hat Process Automation Manager libraries embedded within a custom application
- **Project storage and build environment** External Git repository for project versioning (not synchronized with Business Central) and external Maven repository for project management and building configured with your embedded Java application (not configured with KIE Server)
- **Asset-authoring tools:** Integrated development environment (IDE), such as Red Hat CodeReady Studio, and a spreadsheet editor or a Decision Model and Notation (DMN) modeling tool for other decision formats
- **Main asset types:** Drools Rule Language (DRL) rules, spreadsheet decision tables, and Decision Model and Notation (DMN) models for decisions, and Business Process Model and Notation (BPMN) models for processes
- **Project deployment and execution environment:** Embedded Java application, such as in a Java virtual machine (JVM) environment, microservice, or custom application server
- **Scenario advantages:**
 - Custom implementation of Red Hat Process Automation Manager in an on-premise development environment with an embedded Java application
 - Ability to define business assets using an external IDE and other asset-authoring tools of your choice
 - Use of custom APIs to interact with core KIE APIs (not KIE Server APIs) and to execute assets in the embedded engine

Figure 6.4. Red Hat Process Automation Manager with an IDE and an embedded Java application



CHAPTER 7. ADDITIONAL RESOURCES

- *[Planning a Red Hat Process Automation Manager installation](#)*
- *[Getting started with decision services](#)*
- *[Getting started with Red Hat build of OptaPlanner](#)*
- *[Getting started with process services](#)*
- *[Packaging and deploying a Red Hat Process Automation Manager project](#)*

APPENDIX A. VERSIONING INFORMATION

Documentation last updated on Wednesday, January 26, 2022.

APPENDIX B. CONTACT INFORMATION

Red Hat Process Automation Manager documentation team: brms-docs@redhat.com