

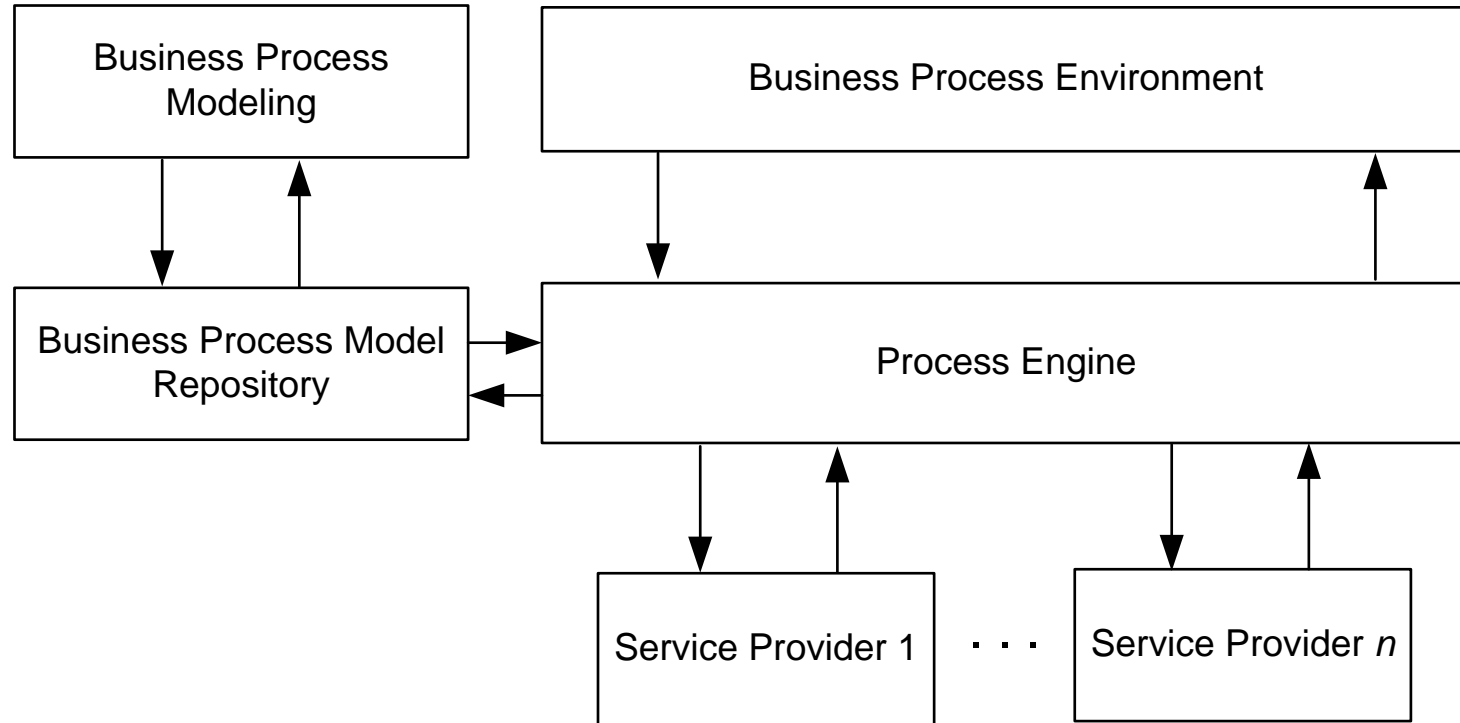


Tutorial: Introduction to Camunda Process Engine



Dr. Luise Pufahl | Chair of Software and Business Engineering
luise.pufahl@tu-berlin.de

General Architecture of BPMS

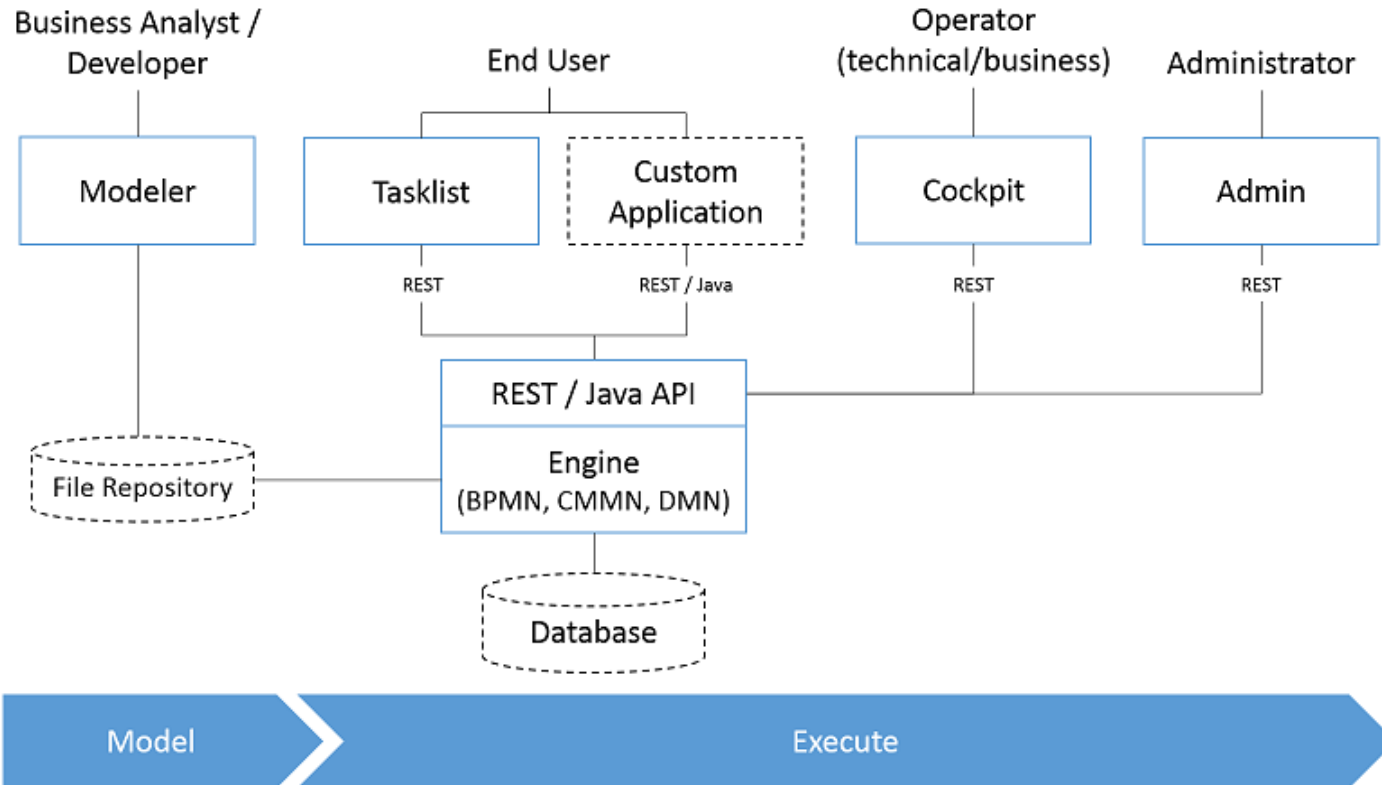


from M. Weske: Business Process Management, © Springer-Verlag Berlin Heidelberg 2007

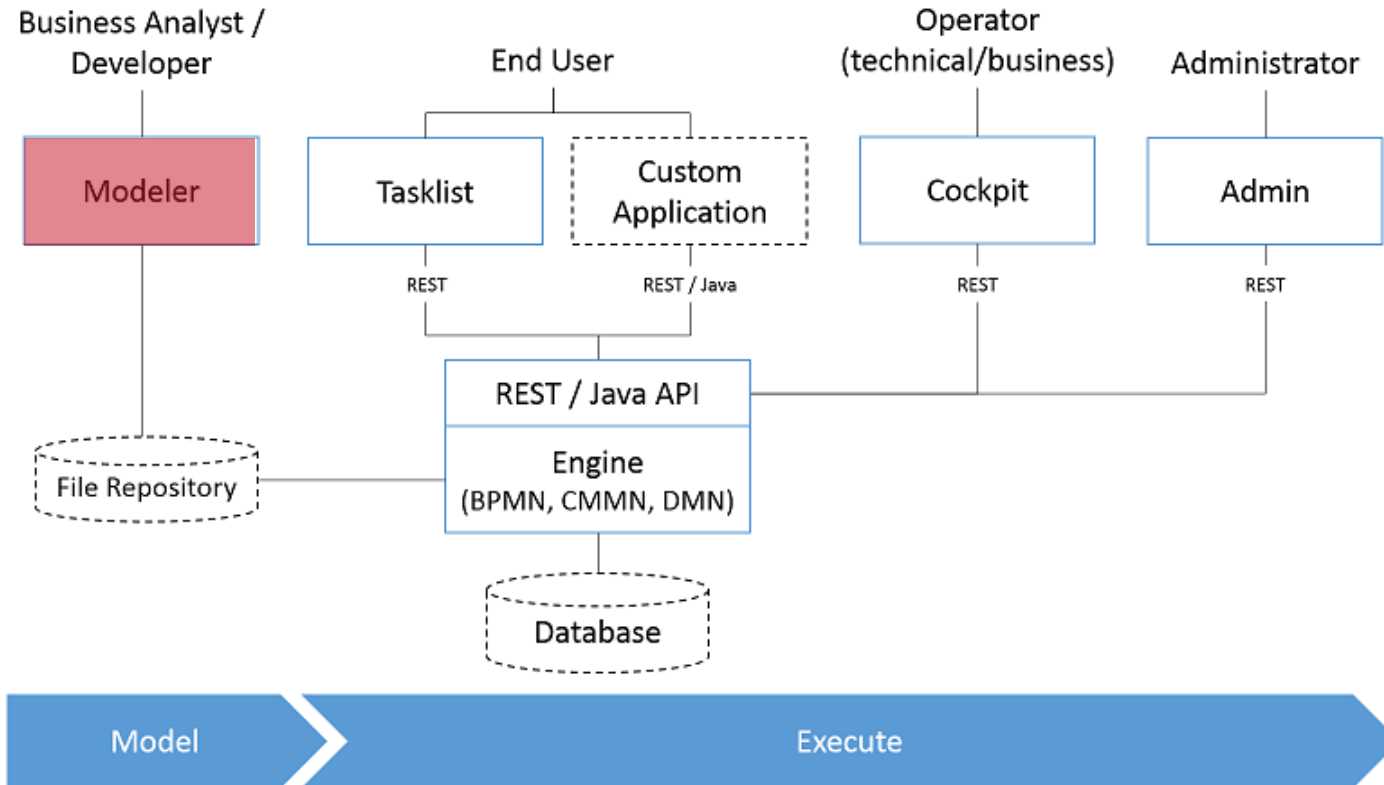
Why could we select Camunda?

- Open-source BPMN engine
- Rich documentation
- Community
- Several extensions

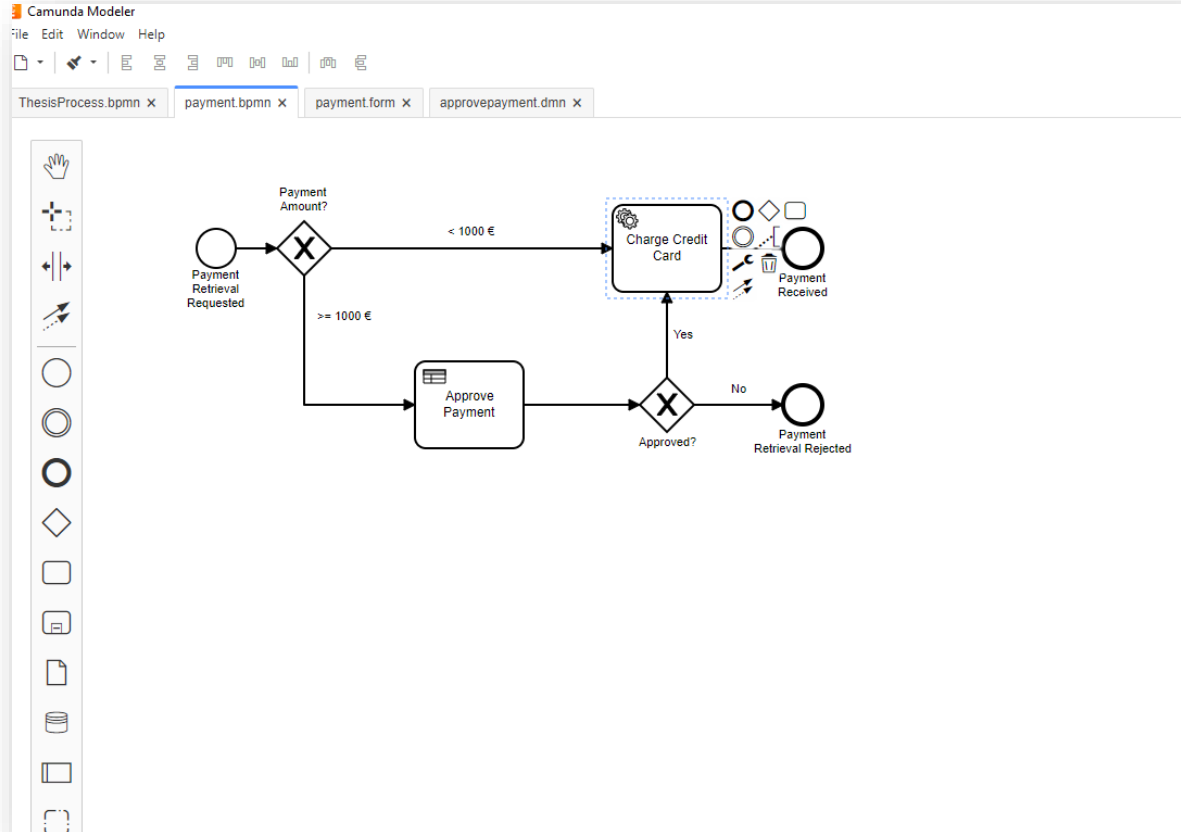
Architecture of Camunda Plattform



Architecture of Camunda Plattform



- BPMN
- DMN
- Forms



Java Maven Project for Service Tasks

eclipse-workspace - charge-card-worker/src/main/java/org/camunda/bpm/getstarted/chargecard/ChargeCardWorker.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- charge-card-worker
 - src/main/java
 - org.camunda.bpm.getstarted.chargecard
 - ChargeCardWorker.java
- JRE System Library [JavaSE-1.8]
- Maven Dependencies
- src
- target
- pom.xml
- test_ProMPlugin

module-info.java ProM with UfTopia (MiningResourceProfilesPlugin).launch ChargeCardWorker.java

```
24 * Copyright Camunda Services GmbH and/or licensed to Camunda Services GmbH
17 package org.camunda.bpm.getstarted.chargecard;
18
19 import java.awt.Desktop;
20
21 public class ChargeCardWorker {
22     private final static Logger LOGGER = Logger.getLogger(ChargeCardWorker.class.getName());
23
24     public static void main(String[] args) {
25         ExternalTaskClient client = ExternalTaskClient.create()
26             .baseUrl("http://localhost:8080/engine-rest")
27             .asyncResponseTimeout(10000) // long polling timeout
28             .build();
29
30         // subscribe to an external task topic as specified in the process
31         client.subscribe("charge-card")
32             .lockDuration(1000) // the default lock duration is 20 seconds, but you can override this
33             .handler((externalTask, externalTaskService) -> {
34                 // Put your business logic here
35
36                 // Get a process variable
37                 String item = externalTask.getVariable("item");
38                 Integer amount = externalTask.getVariable("amount");
39                 LOGGER.info("Charging credit card with an amount of " + amount + "€ for the item " + item + "...");
40
41                 try {
42                     Desktop.getDesktop().browse(new URI("https://docs.camunda.org/get-started/quick-start/complete"));
43                 } catch (Exception e) {
44                     e.printStackTrace();
45                 }
46
47                 // Complete the task
48                 externalTaskService.complete(externalTask);
49             })
50             .open();
51     }
52 }
53
54
55
56
```

Outline

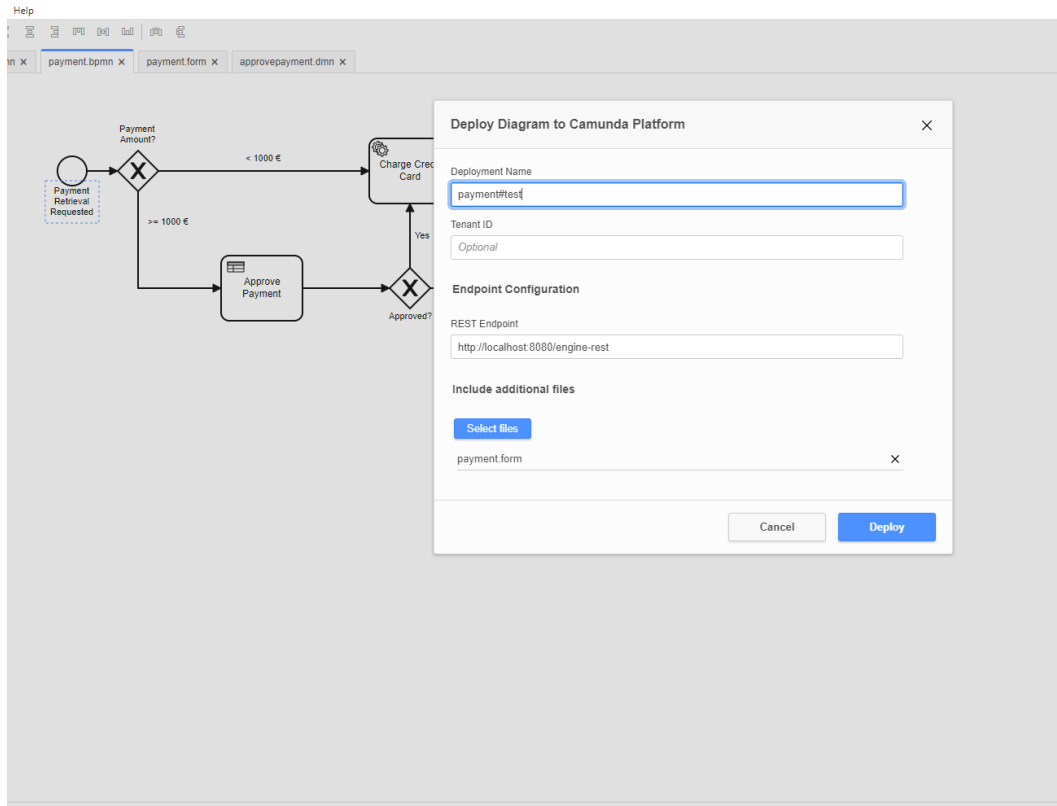
- org.camunda.bpm.getstarted.chargecard
 - ChargeCardWorker
 - LOGGER: Logger
 - main(String[]): void

Problems Javadoc Declaration

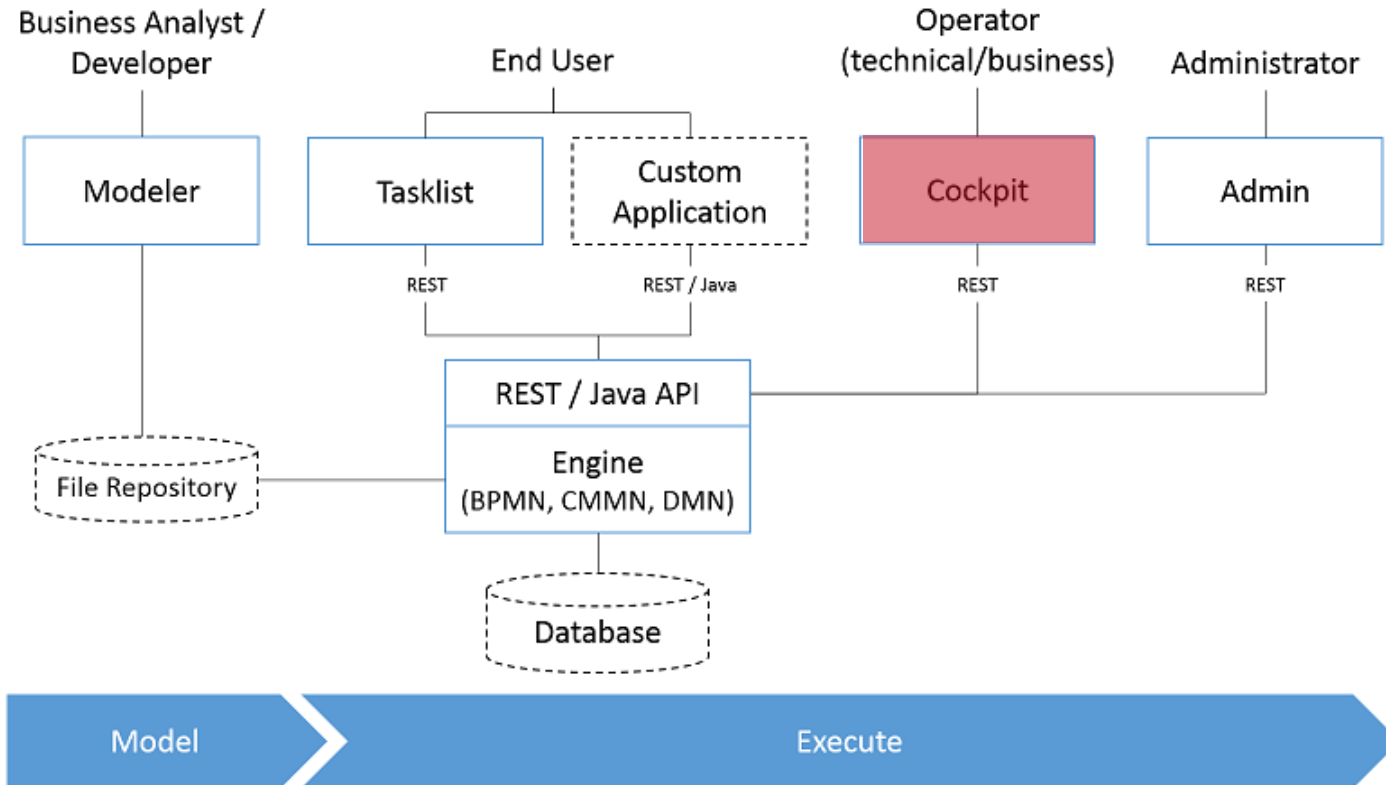
9 errors, 2 warnings, 0 others

Description	Resource	Path	Location	Type
-------------	----------	------	----------	------

Deployment



Architecture of Camunda Plattform



Camunda Cockpit Processes Decisions Human Tasks More ▾ Demo Demo

Dashboard » Processes » Invoice Receipt : Runtime

Definition Version: 2 ▾

Version Tag: V2.0

Definition ID: invoice:2:29d179f9-3b02-11ec-b707...

Definition Key: invoice

Definition Name: Invoice Receipt

History Time To Live: 45 days ✕

Tenant ID: null

Deployment ID: 29b58d86-3b02-11ec-b707-98af65...

Instances Running:

- current version: 4
- all versions: 7

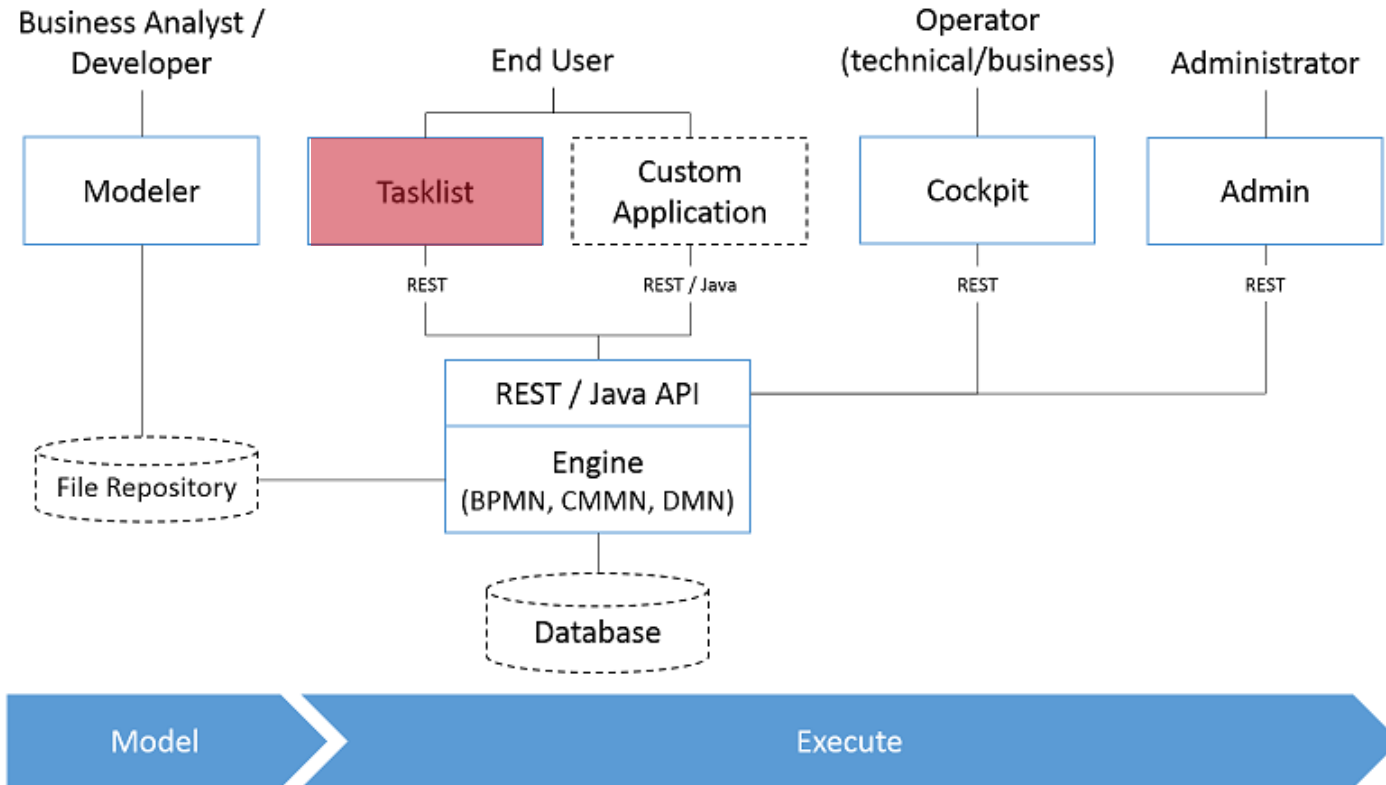
Activity Instance Statistics: on

Process Instances Incidents Called Process Definitions Job Definitions

Add criteria

State	ID	Start Time	Business Key
✓	ad2fe669-3bc6-11ec-b603-98af65178ca6	2021-11-02T11:57:31	
✓	2ee6e5df-3b02-11ec-b707-98af65178ca6	2021-11-01T11:55:10	
✓	2f0c6eb8-3b02-11ec-b707-98af65178ca6	2021-10-27T11:55:10	
✓	2ef3421f-3b02-11ec-b707-98af65178ca6	2021-10-18T11:55:10	

Architecture of Camunda Plattform



Tasklist

Camunda Tasklist

Keyboard Shortcuts Create task Start process default Jonny Prosciutto

Create a filter +

Created +

Filter Tasks 30

My Tasks

My Group Tasks

Soon due tasks (30)

All Tasks

Accounting Tasks

Prepare Bank Transfer

Invoice Receipt

Due 7 days ago, Created 14 days ago

50

Prepare Bank Transfer

Invoice Receipt

Due 7 days ago, Created 14 days ago

50

Do more work

Cornercases Process

Due 5 months ago, Created 5 months ago

50

Do more work

Cornercases Process

Due 5 months ago, Created 5 months ago

50

Do more work

Cornercases Process

Due 5 months ago, Created 5 months ago

50

Prepare Bank Transfer

Invoice Receipt

Set follow-up ... 7 days ago Accounting Claim

Form History Diagram Description

Please prepare the bank transfer for the following invoice

Invoice Document invoice.pdf

Creditor Bobby's Office Supplies

Amount 900

Invoice Number BOS-43934

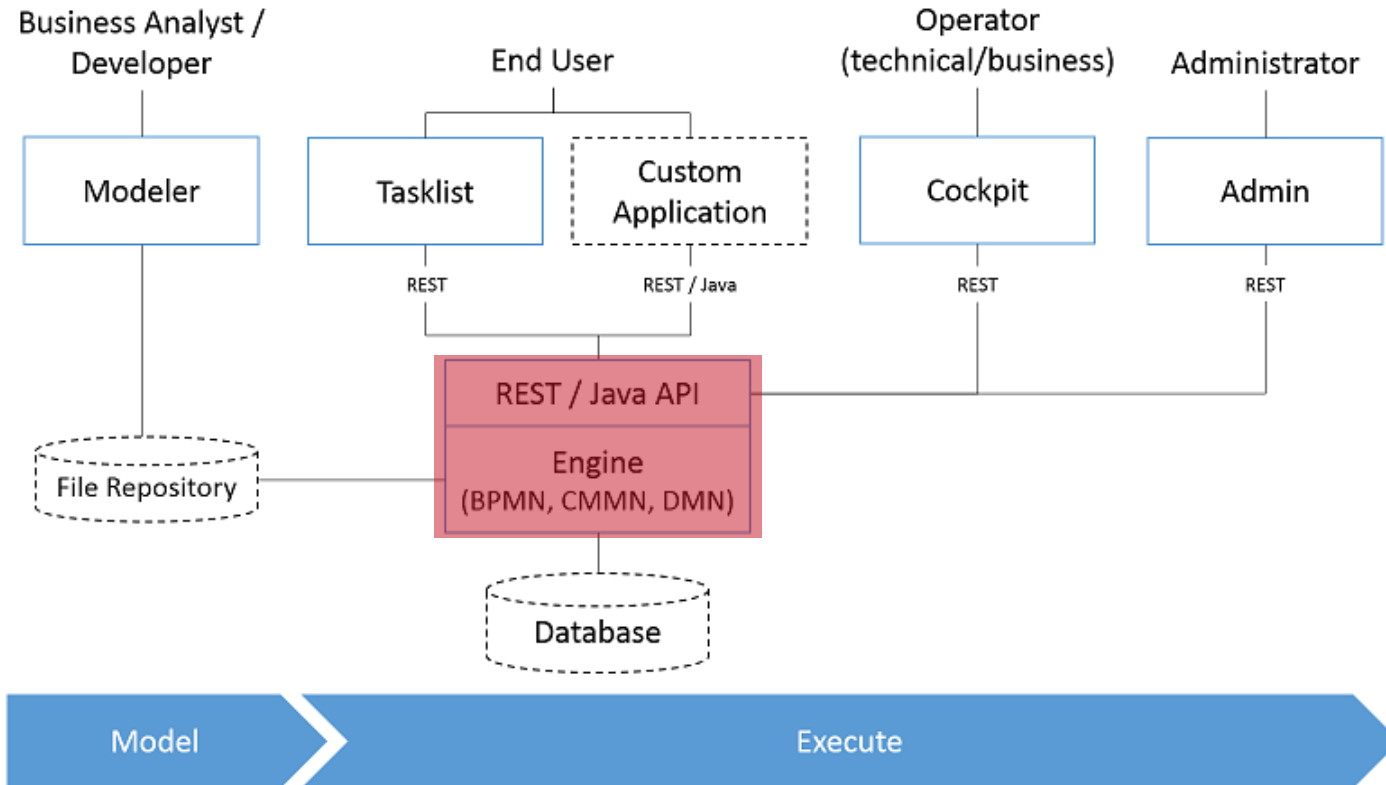
Approved by demo

Save Complete

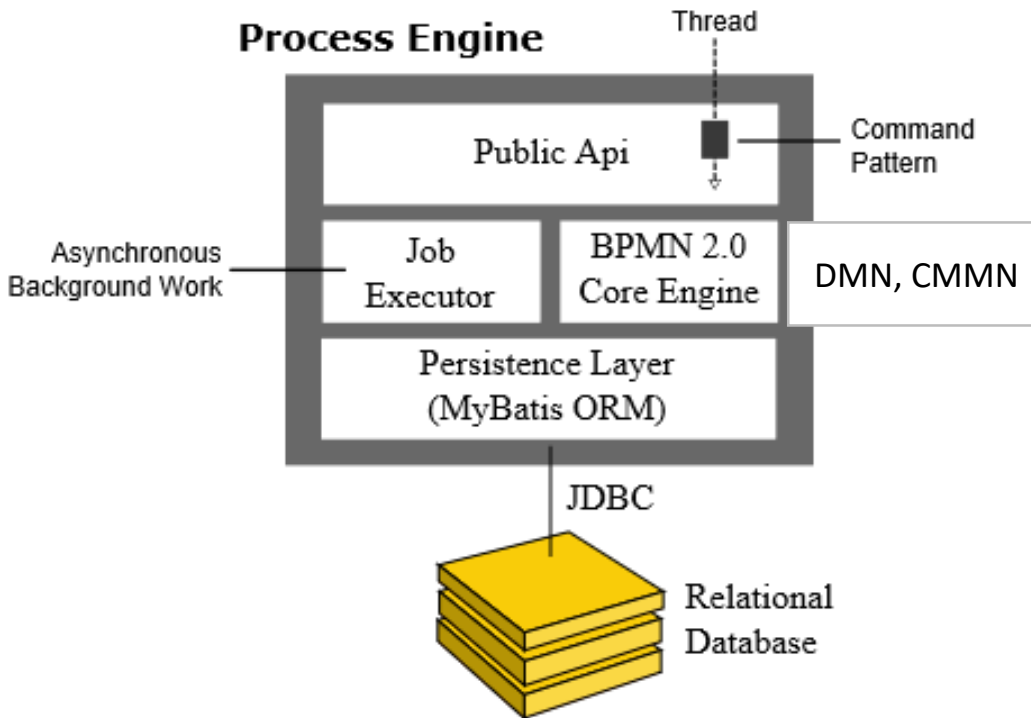
Date and Time displayed in local timezone: Europe/Berlin

Powered by Camunda Platform / v7.16.0-ee

Architecture of Camunda Plattform



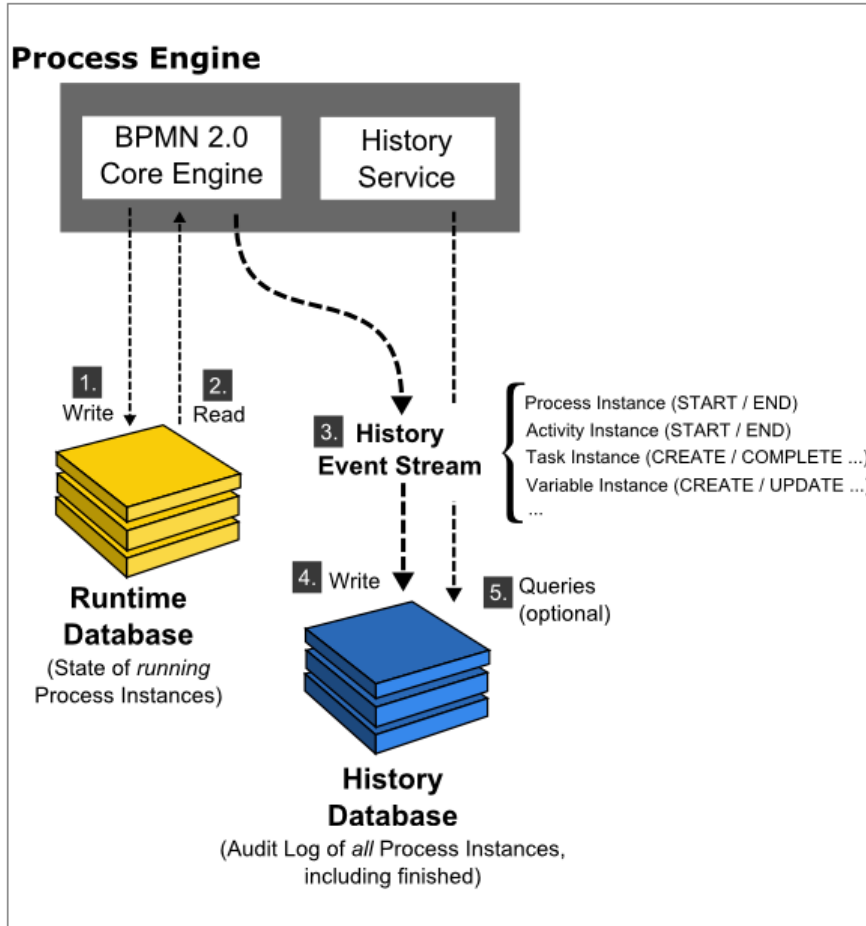
Process Engine Architecture



master camunda-bpm-platform / engine / src / main / java / org / camunda / bpm / engine

koevskinikola and mboskamp chore(engine): fix typo in method argument ...	
..	
AbstractBpmnActivityBehavior.java	feat(engine): throw a BpmnError from listeners
BoundaryConditionalEventActivityBehavior.java	chore(platform): update license headers
BoundaryEventActivityBehavior.java	chore(platform): update license headers
BpmnActivityBehavior.java	chore(engine): fix typo in method argument
BpmnBehaviorLogger.java	feat(engine): handle escalation for user tasks
CallActivityBehavior.java	chore(platform): update license headers
CallableElementActivityBehavior.java	chore(platform): update license headers
CancelBoundaryEventActivityBehavior.java	chore(platform): update license headers
CancelEndEventActivityBehavior.java	chore(platform): update license headers
CaseCallActivityBehavior.java	chore(platform): update license headers
ClassDelegateActivityBehavior.java	chore(platform): update license headers
CompensationEventActivityBehavior.java	chore(platform): update license headers
ConditionalEventBehavior.java	chore(platform): update license headers
CustomActivityBehavior.java	chore(platform): update license headers
DmnBusinessRuleTaskActivityBehavior.java	chore(platform): update license headers

Execution data stored



Camunda Docs

Get Started | Camunda Platform | Optimize | Camunda Enterprise | Security

Search...

Camunda.org

Logging

This page provides information about logging in Camunda.

SLF4J

Most Camunda modules, including the Camunda engine, use [slf4j](#) as logging "facade". This allows users to direct logging output to the logging "backend" of their choice, such as [logback](#) or [log4j](#).

Preconfigured Logging with a Shared Process Engine

When installing Camunda as a shared process engine in an application server, Camunda logging is pre-configured.

On all application servers except JBoss and Wildfly, logging is pre-configured using the [slf4j-jdk14](#) bridge. This means that Camunda effectively re-directs all its logging to Java Util Logging. Both [SLF4J API](#) and the [slf4j-jdk14](#) bridge are available in the classpath of all applications deployed on these servers.

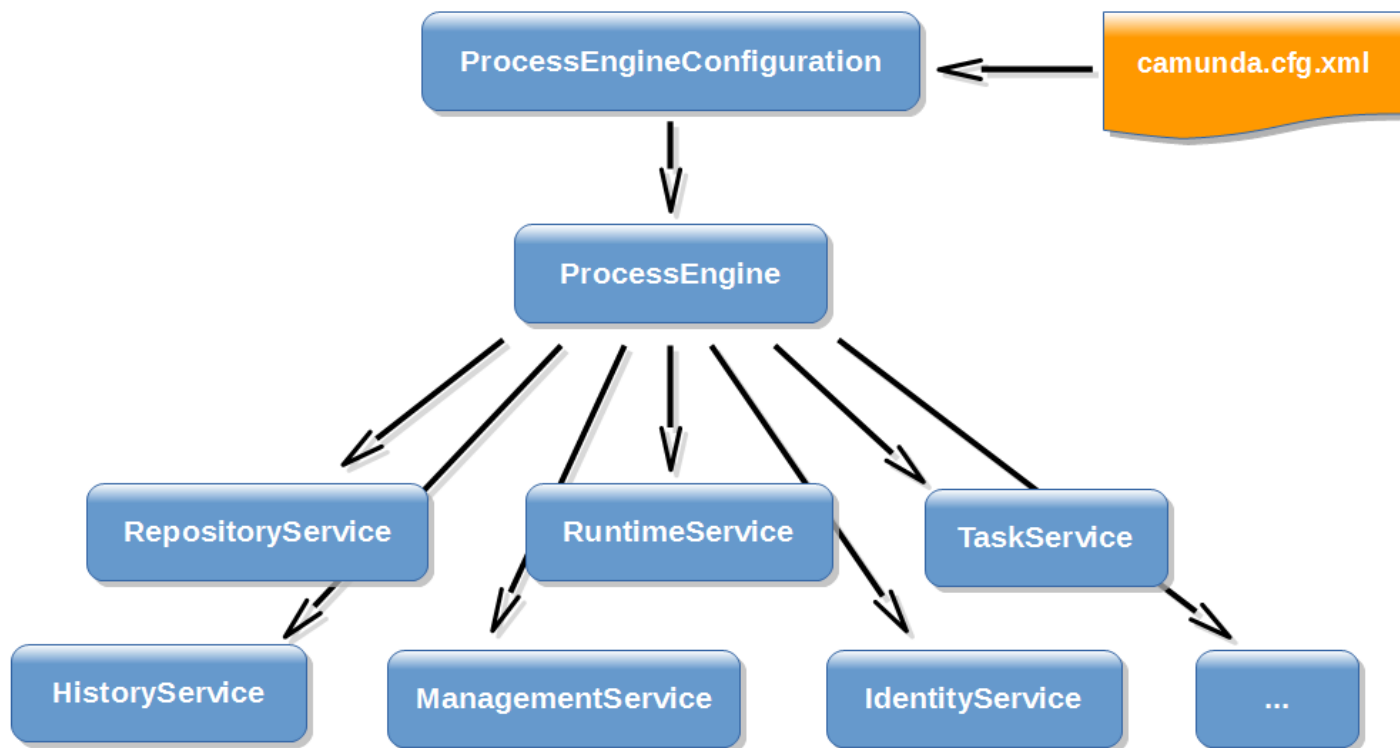
On JBoss/Wildfly, logging is directed to the JBoss logging infrastructure. The [SLF4J API](#) is not available in the classpath of custom applications by default.

Adding a Logging Backend for Embedded Use

When using the Camunda Maven modules in a custom application, only the [slf4j API](#) is pulled in transitively. If you do not provide any backend, nothing will actually be logged.

In the following, we provide two alternative examples of how to set up logging. See the [SLF4J](#)

- <https://arxiv.org/pdf/2009.06209.pdf>



How to Simulate Process Executions?

Camunda Docs

Get Started

Camunda Platform

Optimize

Cawemo

Enterprise

Security

Search...

Camunda.org

Introduction

User Guide

Process Engine

Process Applications

Runtime Container Integration

Camunda Platform Run

Spring Framework Integration

Spring Boot Integration

Quarkus Integration

CDI and Java EE Integration

Testing

Model API

Data Formats (XML, JSON, Other)

User Task Forms

DMN Engine

Logging

Security Instructions

Camunda License Keys

External Task Client

Camunda Platform RPA Bridge

Reference

Installation

Modeler

Web Applications

Examples

Update & Migration

Testing

Testing BPMN processes, CMMN cases (and also DMN decision tables) is a crucial part of the development process. This section explains how to write unit tests and integrate them into the CI pipeline. This section also explains some best practice and guidelines.

Unit Tests

Camunda supports both JUnit versions 3 and 4 styles of unit testing.

JUnit 4

Using the JUnit 4 style of writing unit tests, the `ProcessEngine` rule, the process engine and services are available through `ProcessEngineTestCase` (see above), including this rule will be on the classpath. Process engines are statically cached over the same configuration resource.

The following code snippet shows an example of using the JUnit 4 style of the `ProcessEngineRule`.

```
public class MyBusinessProcessTest {  
  
    @Rule  
    public ProcessEngineRule processEngineRule = new ProcessEngineRule();  
  
    @Test  
    @Deployment  
    public void ruleUsageExample() {  
        RuntimeService runtimeService = processEngineRule.getRuntimeService();  
        runtimeService.startProcessInstanceByKey("ruleUsageExample");  
    }  
}
```

Why GitHub? Team Enterprise Explore Marketplace Pricing

Search

Sign in Sign up

camunda-consulting / camunda-bpm-simulator

forked from nevries/camunda-bpm-simulator

<> Code Issues 3 Pull requests Actions Projects Wiki Security Insights

master 3 branches 10 tags

Go to file Code

About

No description, website, or topics provided.

Readme

Releases

10 tags

Packages

No packages published

Languages

Java 98.8% HTML 1.2%

camunda-bpm-simulator

Camunda process engine plugin to simulate process execution.

Purpose

Next Steps

- Identify how you want to log data, where, and in which format
 - Identify how you can get the data from the process engine
 - Decide how you want to simulate process executions
-