Camunda

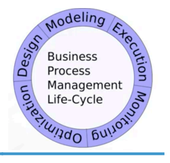
Business Process Management (BPM)

https://camunda.com/

A concept that assumes that Business Processes are organizational resources

Uses various methods to discover, model, analyze, measure, improve, optimize, and automate business processes

BPM life-cycle



Notations

BPM Systems

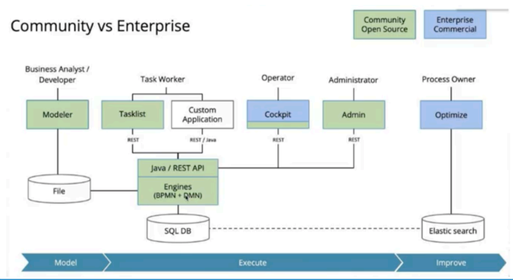
Why Camunda?

Other systems are very expensive and bring significant vendor-lock

Camunda is a fork of Activiti that is much more actively developed.

Camunda Community is open source (Apache 2.0 / MIT)

Camunda supports BPMN, DMN, CMMN notations



Custom apps can be written in any language - e.g. Golang for tasks execution, VueJS for user forms

Q: Is there only Request/Reply pattern from the custom apps? There is no Subscribe/Publish pattern over message bus?

Camunda deployment options

Camunda Platform (Camunda BPM)

As Process Engine library inside the Java app

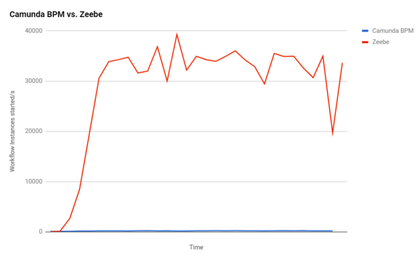
As Process Engine container service inside Java App server

As Process Engine standalone server

Can be accessed via REST, SOAP, JMS

As Process Engine clustered engine using shared database

Can also use multi-tenancy mode to isolate process groups



Zeebe is much more performant than regular Camunda BPM

https://blog.bernd-ruecker.com/zeebe-io-a-horizontally-scalable-distributed-workflow-engine-45788a90d549

Camunda Cloud (Zeebeworkflow engine, a horizontally-scalable deployment)

SaaS

Self-Managed

No PROD usage of Operate, Tasklist, and Optimize - we would need to create GUI or purchase the license

Camunda implementation guides

https://youtube.com/playlist?list=PLJG25HlmvsOX8TilGUZcVW-ez053YsOX0

How to run Camunda, implement business process, add Forms (inside Camunda), implement external tasks (using JS), use Timers, handle Error Events

JS external task code

SDK

https://github.com/camunda/camunda-external-task-client-js

```
const { Client, logger } = require("camunda-external-task-client-js");

// configuration for the Client:
//  - 'baseUrl': url to the Process Engine
//  - 'logger': utility to automatically log important events
const config = { baseUrl: "http://localhost:8080/engine-rest", use: logger };

// create a Client instance with custom configuration
const client = new Client(config);

// susbscribe to the topic: 'creditScoreChecker'
client.subscribe("creditScoreChecker", async function({ task, taskService }) {
  // Put your business logic
  // complete the task
  await taskService.complete(task);
});
```

How to use Camunda with docker, deploy it locally, implement business process, implement external task (using Golang), implement external GUI (using VueJS)

https://arifsetiawan.medium.com/my-journey-with-camunda-toc-3030da004511

Do NOT use built-in forms as they are very poor - always create standalone GUI (e.g. using React or Vue)

Use cases

When to use Camunda?

Automating business processes that requires HUMAN INTERACTION

HR processes

Hire

Vacation request

Purchases

Invoices

Mortgage

...

When NOT to use Camunda?

Automating service tasks that does NOT require human interactions

[Big] data batch processing / task orchestration

Scheduled import of data from one system to another

Better use Airflow or simple cron

Machine Learning models recalculation or data preparation

Better use Airflow or specific MLOps tools

CI/CD, including Automated Testing

Better use Gitlab or Jenkins some other CI tool

Realtime monitoring of services

Better use built-in features in k8s

Importing rare messages from one social network to another social network

Better use SaaS such as IFTTT, or Airflow

Event sourcing / a lot of small events

Collecting real-time market data (e.g. stock prices) and processing them to calculate derivative data

Better use message bus / event log such as Rabbit or Kafka or Nats

Collecting and processing user action logs

Better use specific databases such as ClickHouse directly or via Kafka