INNOVATIVE DIGITAL

Camunda 8 Training

Day 1: Camunda 8 Platform Overview and BPMN Basics

Presented By: Hasan Ghanem | Camunda Champion
hhghanem@innovative-digital.com.sa
https://forum.camunda.io/u/hassang/summary

Join **Slido** using the code:

6647755

Camunda Self-Managed (Local development environment)

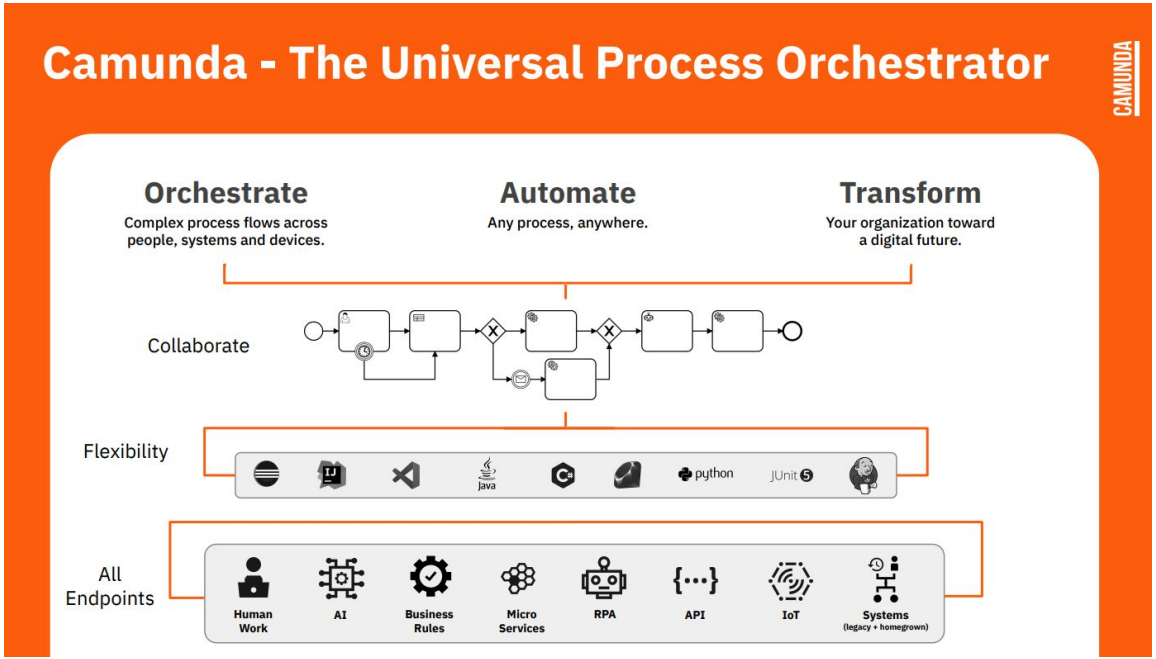
https://signup.camunda.com/self-managed-download

Camunda Developer Hub

https://camunda.com/developers/

Introduction to Camunda 8

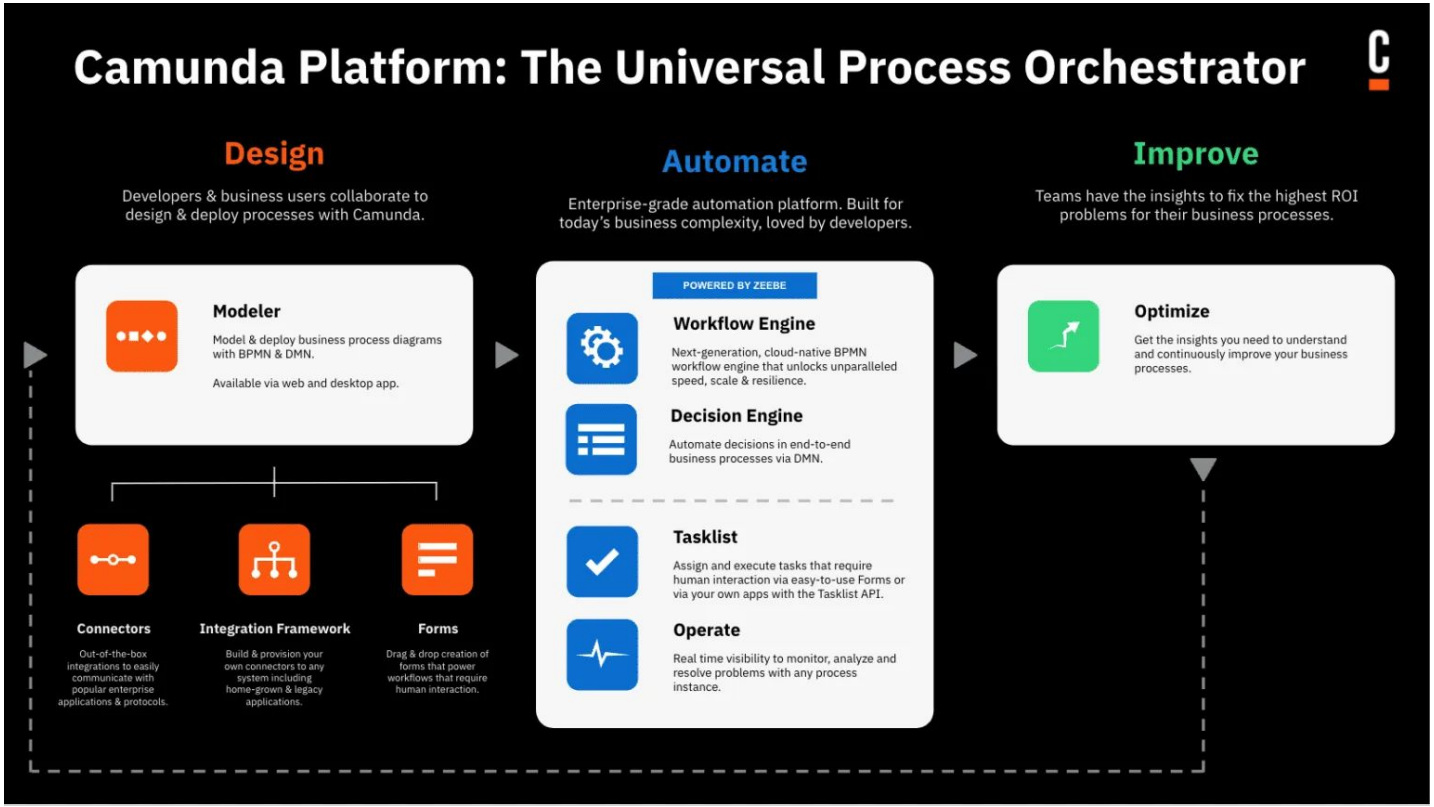
Orchestrate complex process flows across people, systems and devices.



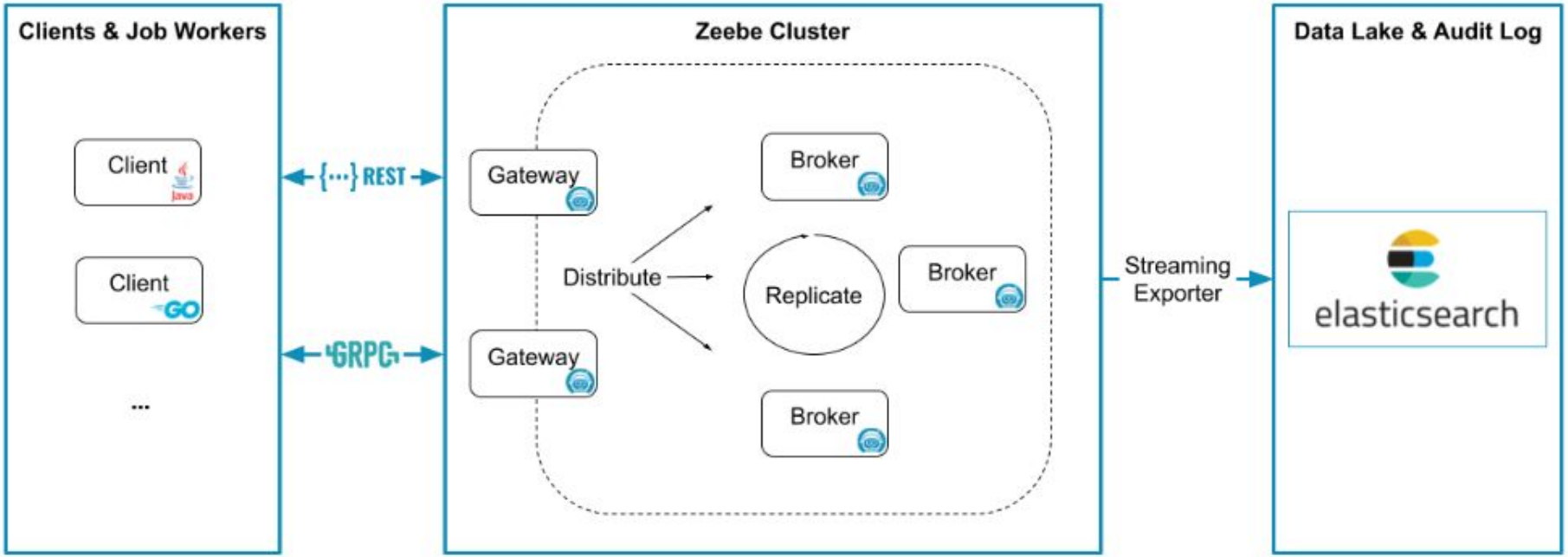
Camunda 8 is a cloud-native, stateless workflow engine based on Zeebe. It is designed for distributed systems and high scalability, using event sourcing and streaming for data persistence.

The primary difference between centralized and distributed systems is the communication pattern between the system's nodes. The state of a centralized system is contained within a central node. Nodes of a centralized system all access the central node. A centralized system has a single point of failure while a distributed system has no single point of failure.

Camunda 8 Components

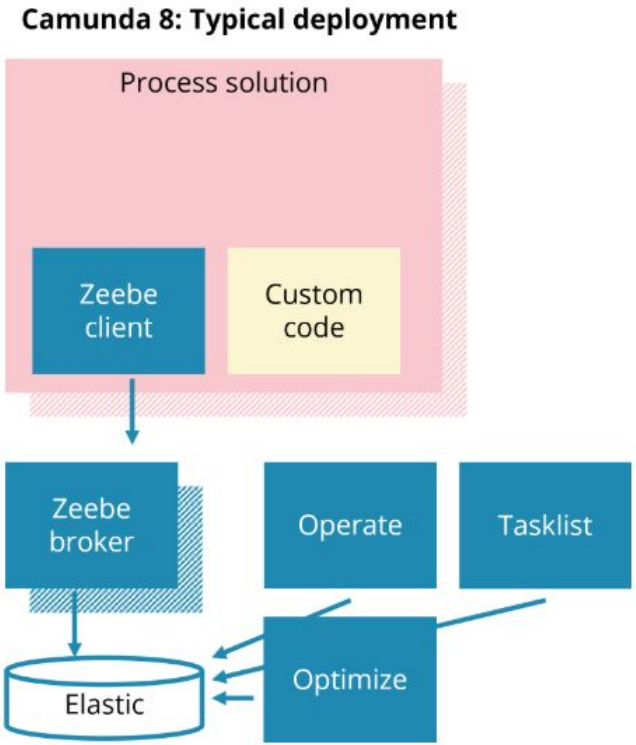


Architecture



https://docs.camunda.io/docs/components/zeebe/technical-concepts/architecture

Camunda 8: Typical Deployment



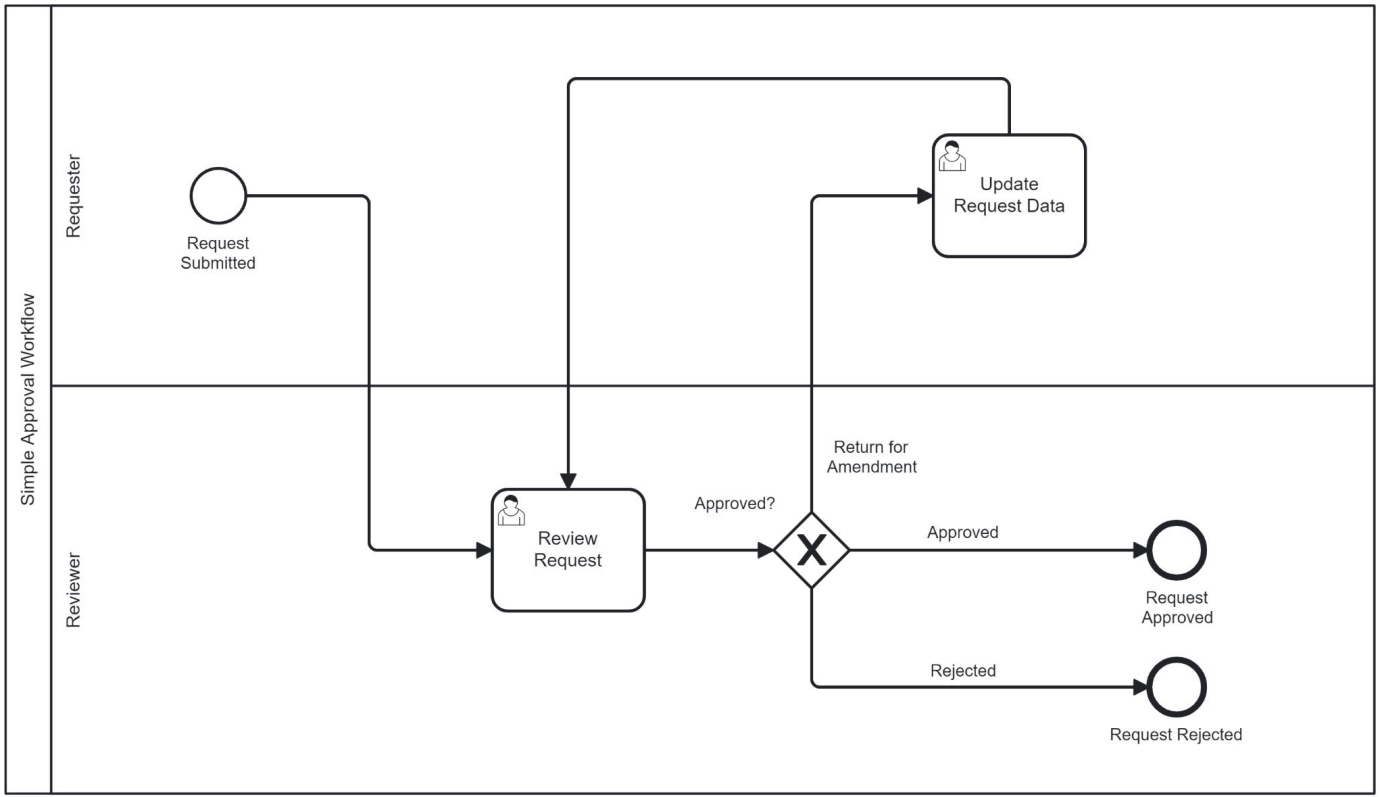
BPMN Basics: Modeling Workflows

In general, certain tasks have to be carried out during a process (**activities**), perhaps under certain conditions (**gateways**), and things may happen (**events**). What connects these **three flow objects** are **sequence flows**, but only within a **pool**. If connections cross pool boundaries, the process resorts to **message flows**.

Source: Real-Life BPMN (4th edition)

https://page.camunda.com/wp-bpmn-2-0-business-process-model-and-notation-en
https://docs.camunda.io/docs/components/modeler/bpmn/bpmn-coverage/

Create, Deploy and Test a Simple Workflow



INNOVATIVE DIGITAL

Camunda 8 Training

Day 2: Advanced BPMN Patterns and Worker Development

Presented By: Hasan Ghanem | Camunda Champion
hhghanem@innovative-digital.com.sa
https://forum.camunda.io/u/hassang/summary

Advanced Workflow Patterns

Explore common workflow patterns:

- Reference: Workflow Patterns Documentation. (https://docs.camunda.io/docs/components/concepts/workflow-patterns/)
- Error handling, compensation, and boundary events.
- Parallel and sequential workflows.

Variables and Data Flow

- Defining, passing, and manipulating variables in workflows.
- Best practices for managing data in workflows.

https://docs.camunda.io/docs/components/modeler/bpmn/data-flow/
https://docs.camunda.io/docs/components/best-practices/development/handling-data-in-processes/

Developing Workers

- Introduction to workers and their role in the Camunda platform.
- Writing and deploying workers in Java.

https://docs.camunda.io/docs/components/concepts/job-workers/
https://docs.camunda.io/docs/apis-tools/spring-zeebe-sdk/getting-started/
https://docs.camunda.io/docs/apis-tools/spring-zeebe-sdk/configuration/

Using Connectors and Execution Listeners

- Overview of built-in connectors for external system integrations.
- Using execution listeners.

https://docs.camunda.io/docs/components/connectors/introduction-to-connectors/
https://docs.camunda.io/docs/self-managed/connectors-deployment/install-and-start/#connector-templates

Built-In Connectors

Out-of-the-box (OOTB) Connectors accelerate solution implementation by providing pre-built, ready-to-use Connectors to popular external systems.

https://docs.camunda.io/docs/components/connectors/out-of-the-box-connectors/available-connectors-overview/

Execution Listeners

Some **use cases** could be:

- Generate a UID as soon as the process get started
- Invoke a backend delegate to insert task level information in external database as soon as the task is created
- Update business information related to tasks as soon as the task is completed
- Update process status in external database as soon as the process instance is completed

Execution Listeners

Execution listeners and service tasks in Camunda serve different roles:

1. **Execution listeners** handle general tasks like logging, metrics, or resource cleanup. They're great for small, reusable actions that don't affect the process flow and keep business logic clean.
2. **Service tasks** are for key process actions, such as external service calls or complex business logic, and appear in the BPMN diagram to make the process clear.

Use **execution listeners** for lightweight operations, cross-cutting tasks, or consistency across multiple processes. Use **service tasks** for major process steps, long-running tasks, or when error handling and retries are needed.

https://academy.camunda.com/c8-execution-listeners/2052065

INNOVATIVE DIGITAL

Camunda 8 Training

Day 3: Decision Models, Operations, and Best Practices

Presented By: Hasan Ghanem | Camunda Champion
hhghanem@innovative-digital.com.sa
https://forum.camunda.io/u/hassang/summary

DMN and FEEL Overview

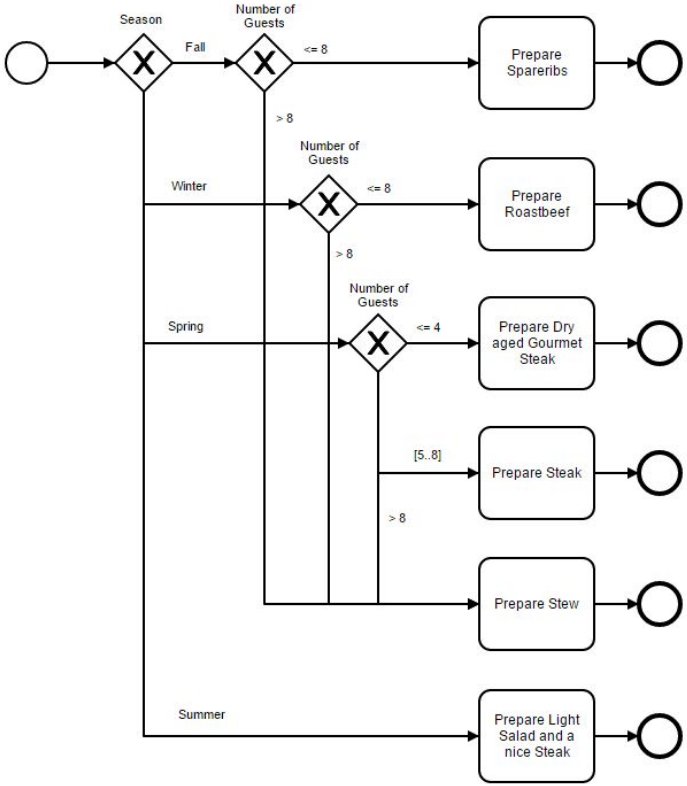
- Introduction to Decision Models (DMN).
- Basics of FEEL (Friendly Enough Expression Language).
- Creating simple decision tables and integrating them with BPMN workflows

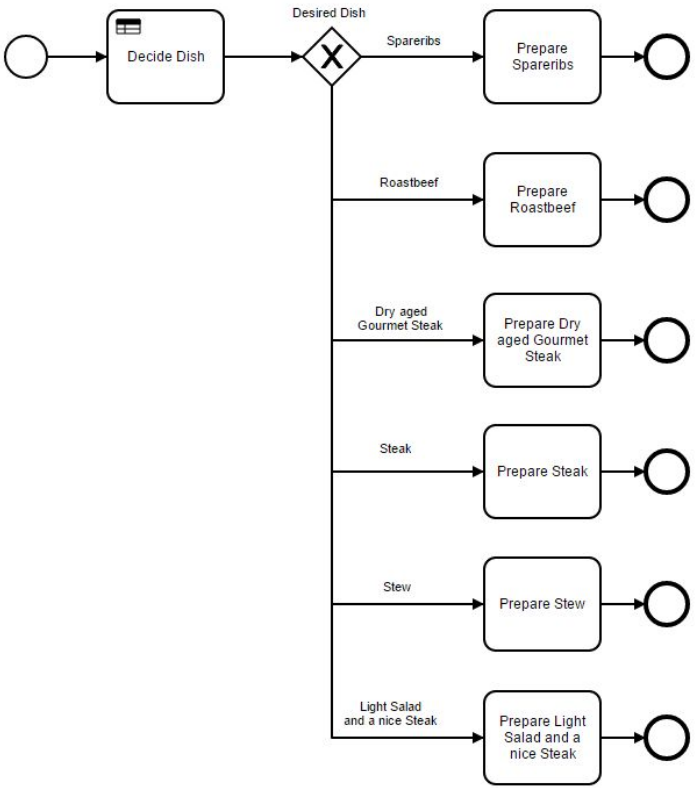
What is DMN?

Decision Model and Notation (DMN) is a modeling approach owned by an institution called the Object Management Group (OMG), which also operates worldwide standards for BPMN. In DMN, decisions are modeled and executed using a language both business analysts and developers can understand.

https://docs.camunda.io/docs/components/modeler/dmn/
https://docs.camunda.io/docs/components/modeler/feel/what-is-feel/

Without DMN vs With DMN





Decide Dish DMN Table

Dish | **Hit policy:** Unique

	When Season string	And How many guests integer	Then Dish string	Annotations
1	"Fall"	<= 8	"Spareribs"	
2	"Winter"	<= 8	"Roastbeef"	
3	"Spring"	<= 4	"Dry Aged Gourmet Steak"	
4	"Spring"	[5..8]	"Steak"	Save money
5	"Fall", "Winter", "Spring"	> 8	"Stew"	Less effort
6	"Summer"	-	"Light Salad and a nice Steak"	Hey, why not!?
+	-	-		

Operational Tasks

- Resolving Incidents:
  - Identifying incidents using Operate.
  - Resolving common errors and updating variables.
- Process Instance Modification:
  - Adding, removing, or skipping steps in running instances.
- Process Instance Migration:
  - Updating workflows and migrating instances between versions.

https://docs.camunda.io/docs/components/operate/userguide/resolve-incidents-update-variables/
https://docs.camunda.io/docs/components/operate/userguide/process-instance-modification/
https://docs.camunda.io/docs/components/operate/userguide/process-instance-migration/

Best Practices

https://docs.camunda.io/docs/components/best-practices/best-practices-overview/