

# Low-code: Building Automated Processes Without Getting Lost in Spaghetti of Code

Goldman  
Sachs



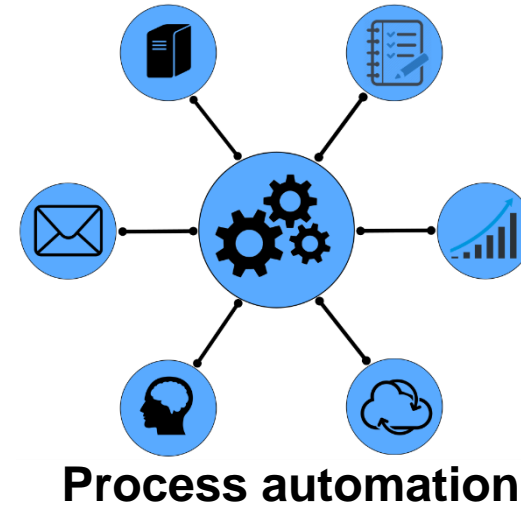
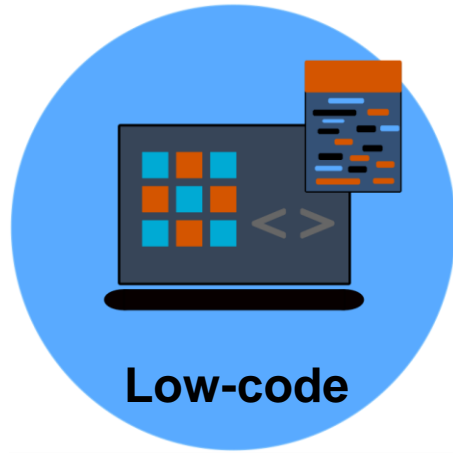
Anna Legierska  
Software Engineer  
[in](#) anna-legierska



Jadwiga Piechota  
Software Engineer  
[in](#) jadwiga-piechota

15 June 2023

# What we would talk about?



# Agenda

Goldman  
Sachs

1. Theory around process automation and low-code
2. Warm-up example
3. Define your own process

# Theory around process automation and low-code

# Why you use low-code?

Key concepts, features and benefits

Enable continuous  
collaboration between  
business and IT  
(developers)

---

## **Ease of communication**

Both speak the same language

Transparency

Both understand the process  
definition

# Why you use low-code?

Key concepts, features and benefits

Enable continuous  
collaboration between  
business and IT  
(developers)

## Ease of communication

Both speak the same language

Transparency

Both understand the process  
definition

Create model-driven  
process using  
visual tools

## Reduce the amount of code

Involvement in development  
without coding skills

Lower barrier of entry

Visual modelling

# Why you use low-code?

Key concepts, features and benefits

Enable continuous collaboration between business and IT (developers)

## Ease of communication

Both speak the same language

Transparency

Both understand the process definition

Create model-driven process using visual tools

## Reduce the amount of code

Involvement in development without coding skills

Lower barrier of entry

Visual modelling

Accelerate your application development cycle

## Actively redesign and update process without developers contribution

Respond quickly to stay competitive

Increase decision speed

Increase pace of development

Scalable environments

# Why you use low-code?

Key concepts, features and benefits

Enable continuous collaboration between business and IT (developers)

## Ease of communication

Both speak the same language

Transparency

Both understand the process definition

Create model-driven process using visual tools

## Reduce the amount of code

Involvement in development without coding skills

Lower barrier of entry

Visual modelling

Accelerate your application development cycle

## Actively redesign and update process without developers contribution

Respond quickly to stay competitive

Increase decision speed

Increase pace of development

Scalable environments

Use the power of a single platform

## Application lifecycle management

Collaboration tools

Reusable components

Data integration



# Why you use low-code?

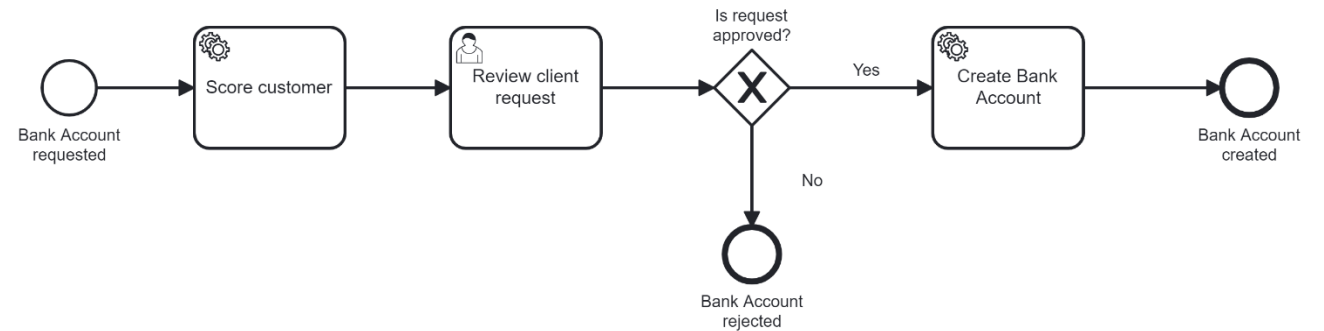
Key concepts, features and benefits



Low-code development provides a single platform where both non-technical users and professional developers can apply their skillsets, collaborate, and co-develop solutions.

## Graphical representation brings a value

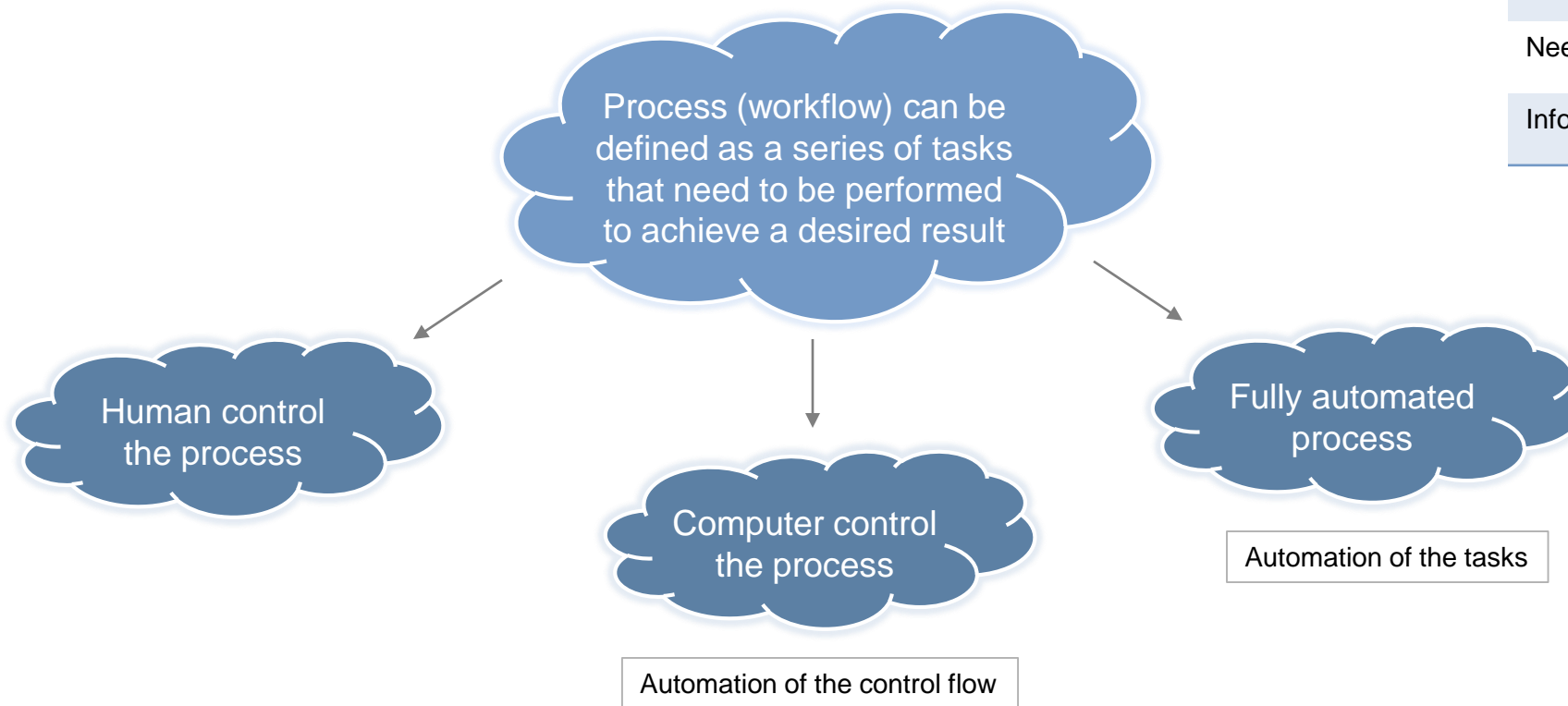
## Graphical representation brings a value



1. Model can be understood by business, stakeholders and developers.
2. Introduce increased visibility and comprehensibility of the model, even for developers.
3. Huge help when discussing requirements before and during implementation.
4. Troubleshooting easier – in case incidents happen.
5. Self-documented code.



# Automating process execution



## Main reasons that motivates automation

High number of repetitions

Standardization

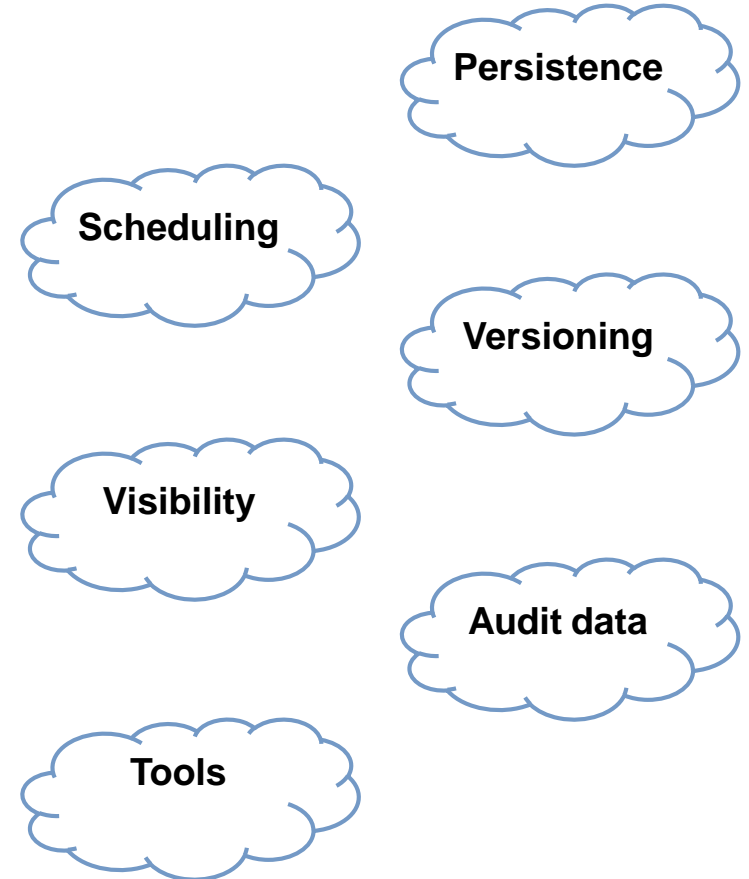
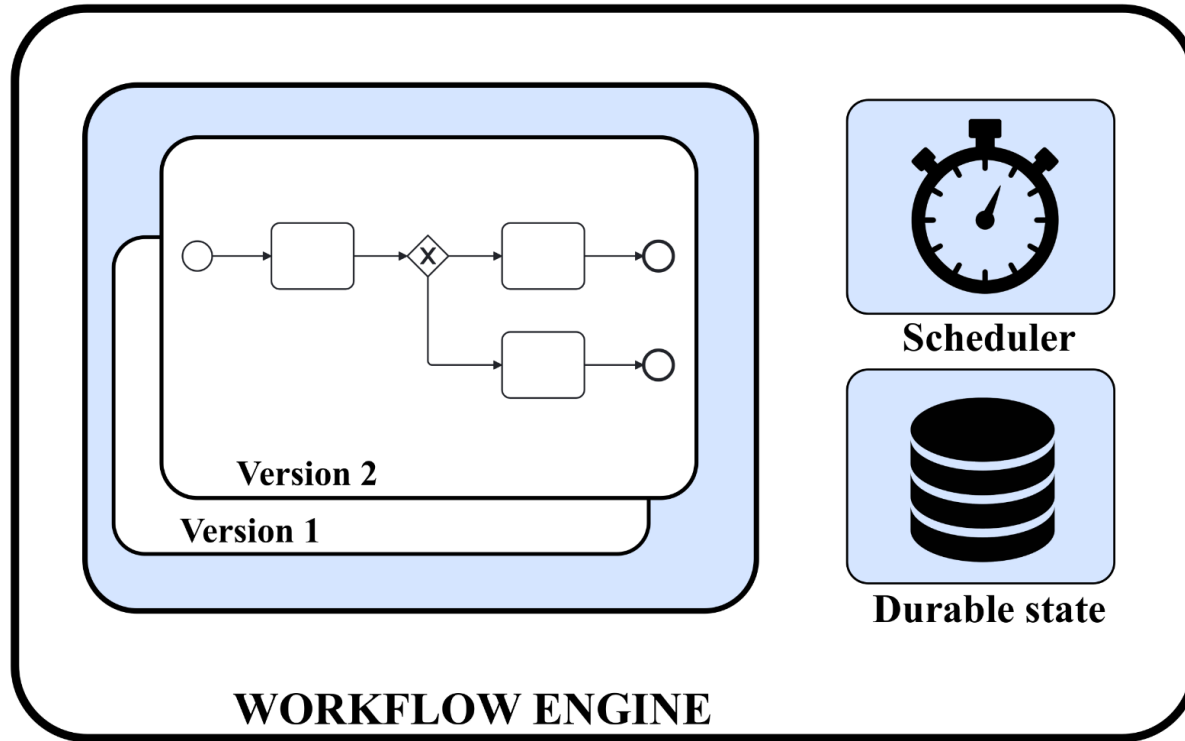
Compliance requirement

Need for quality

Information richness

# Core capabilities of workflow engine

Key component for automating the control flow



# BPMN & Camunda Platform



## **BPMN (Business Process Model and Notation)**

Process modelling language; open standard to diagram a business process. BPMN provides businesses with the capability of understanding their internal business procedures in a graphical notation with standardized manner.

## **Camunda Platform**

Process orchestration (workflow and decision automation) platform that allows developers to design, automate and improve processes.

Camunda Platform 8 that we use in this workshop is available in two options: SaaS and Self-Managed.

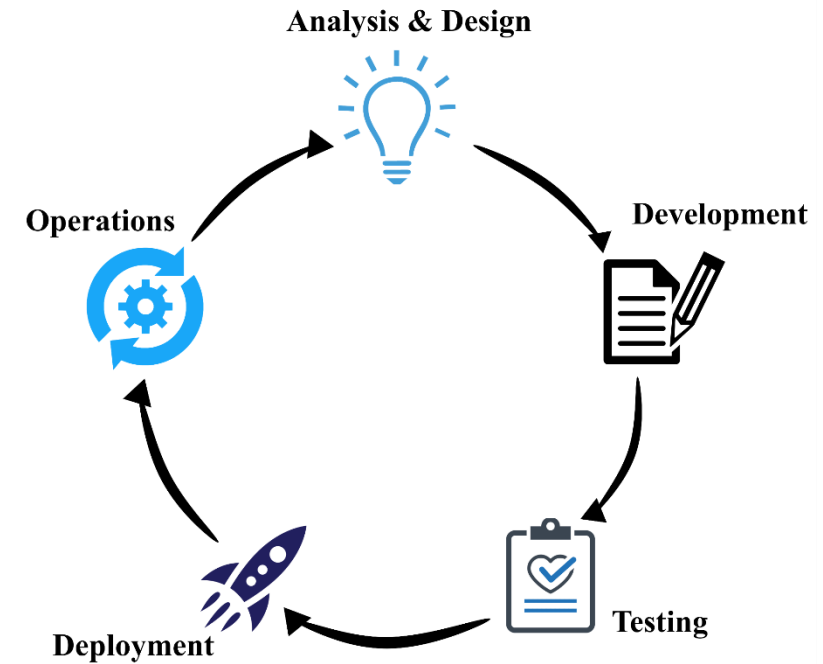
- SaaS – no technical setup or installation is required (cloud based, hosted by Camunda).
- Self-Managed option – has to be hosted, installed and setup on customer infrastructure.



# How it works?

## Process Lifecycle

1. Collect requirements for your process, analysis and design.
2. Define your process definition — design a blueprint that can be executed on a workflow engine.
3. Confront your model with business requirements — collaborate with business in order to define process.
4. Test your model.
5. Deploy process definition to the workflow engine (start with test environments).
6. Run at least one process instance (you can define many).
7. Monitor process instances (operate phase).
8. Improve your process when necessary.



# Warm-up example

# How to start with Camunda Cloud?

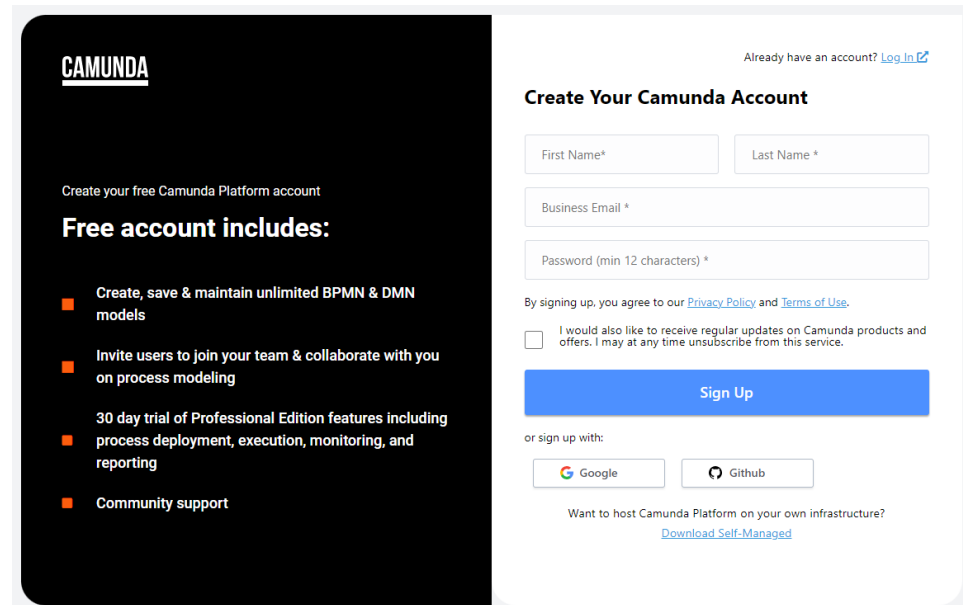
## Create an account

Open Camunda Platform: <https://camunda.com/platform/> and create an account. You will have 30 days of free trial period with one test cluster available.

Camunda Cloud offers two solutions to design your models:

- Web Modeler (embedded in Camunda Platform 8 Console)
- Desktop Modeler (desktop application).

We will use Web Modeler for the purpose of this workshop.



The screenshot shows the Camunda Cloud account creation interface. On the left, a dark sidebar contains the Camunda logo and a list of features for a free account: unlimited BPMN & DMN models, team collaboration, a 30-day trial of Professional Edition features, and community support. The main area is white and titled 'Create Your Camunda Account'. It includes a 'Log In' link for existing users, input fields for First Name, Last Name, Business Email, and Password (minimum 12 characters), and a 'Sign Up' button. Below the sign-up button, there are links for Google and GitHub sign-in, and a link to download the self-managed version for those who want to host it on their own infrastructure.

**CAMUNDA**

Create your free Camunda Platform account

**Free account includes:**

- Create, save & maintain unlimited BPMN & DMN models
- Invite users to join your team & collaborate with you on process modeling
- 30 day trial of Professional Edition features including process deployment, execution, monitoring, and reporting
- Community support

Already have an account? [Log In](#)

**Create Your Camunda Account**

First Name\*

Last Name \*

Business Email \*

Password (min 12 characters) \*

By signing up, you agree to our [Privacy Policy](#) and [Terms of Use](#).

☐ I would also like to receive regular updates on Camunda products and offers. I may at any time unsubscribe from this service.

**Sign Up**

or sign up with:

[Google](#) [Github](#)

Want to host Camunda Platform on your own infrastructure?  
[Download Self-Managed](#)



## Process requirements

Define a process that needs to automate bank account creation.

Customer is requesting a bank account. Next, the bank employee needs to review the request in order to decide if request can be processed. If the decision is successful, the other employee has to perform some steps in order to provision the account. Otherwise, request is rejected and account is not created.

## Resources

Instruction & code required for the workshop:

<https://tinyurl.com/4jn5r8zt>

# How to start with Camunda Cloud?

## Create cluster


Cluster is a group of nodes that run in parallel to achieve a common goal.

Console

Dashboard

Clusters

Organization



### Processes run on Clusters

Click **Create new Cluster** to have a place to execute your processes

Create new Cluster +

[View The Getting Started Guide](#)

Clusters /

Create a cluster

Cluster name

Test cluster

Select a cluster type

Trial Cluster

Select region

☐ Toronto, North America (northamerica-northeast2)

☐ South Carolina, North America (us-east1)

☐ Iowa, North America (us-central1)

☐ Sydney, Australia (australia-southeast1)

☒ Belgium, Europe (europe-west1)

Pre-selection is based on the best current load and performance.

Channel

Stable

Alpha

The Stable channel provides the latest feature and patch releases ready for most users at minimal risk. The releases follow semantic versioning and can be updated to the next minor or patch release without data loss.

Generation

☒ Zeebe 8.2.4

☐ Zeebe 8.1.5

☐ Zeebe 8.0.11

☐ Zeebe 8.0.6

Summary

1 x Trial Cluster

Region  
Belgium, Europe (europe-west1)

Channel  
Stable

Generation  
Zeebe 8.2.4

Create cluster

Cancel

Need help? [Talk to an expert](#)

Clusters

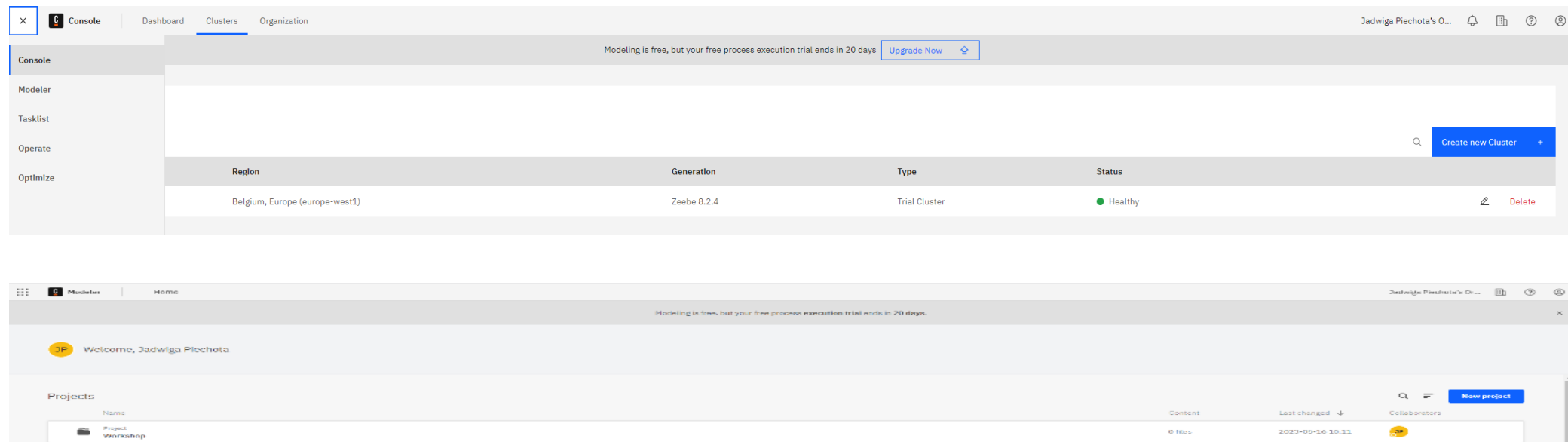
Create new Cluster +

Name	Region	Generation	Type	Status	
Test cluster	Belgium, Europe (europe-west1)	Zeebe 8.2.4	Trial Cluster	Creating	<a href="#">Edit</a> <a href="#">Delete</a>

# How to start with Camunda Cloud?

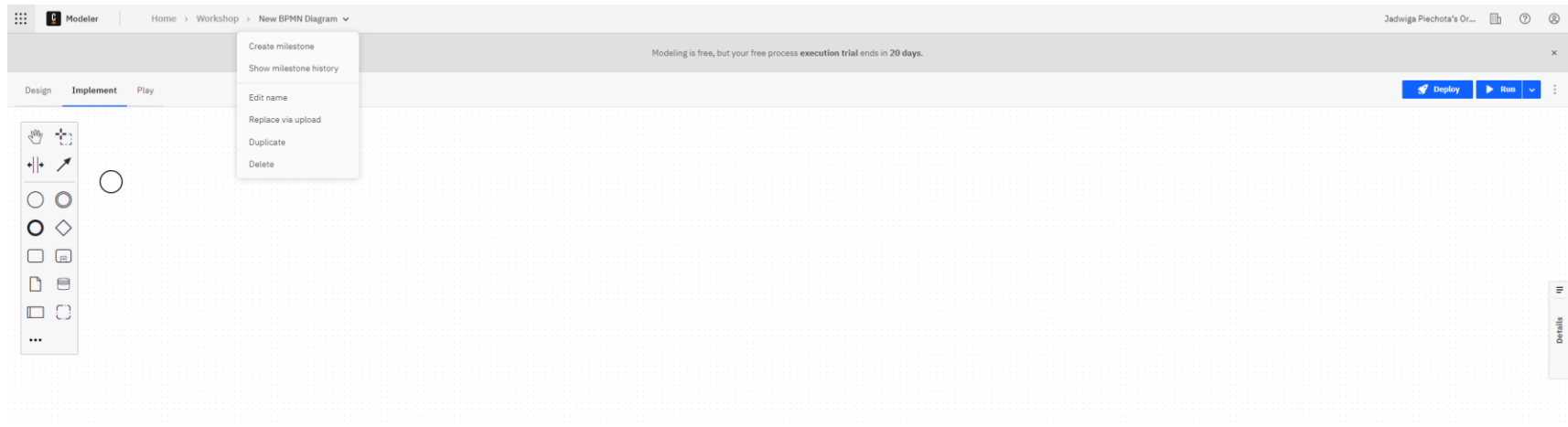
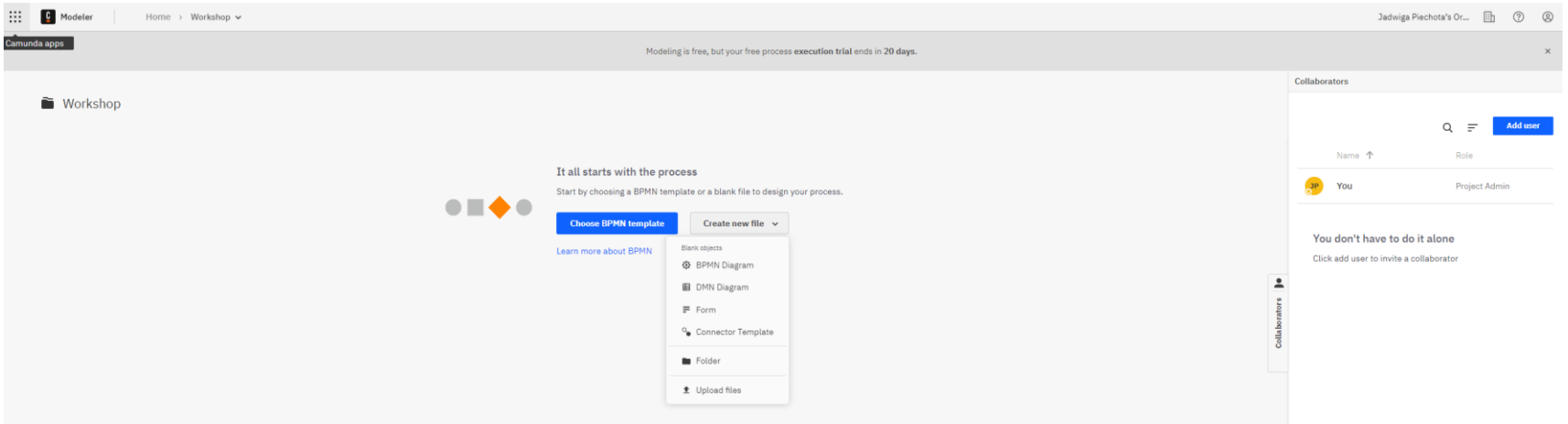
## Create a project

Project is a container for your process components – BPMN diagrams, DMN diagrams, forms and folders.

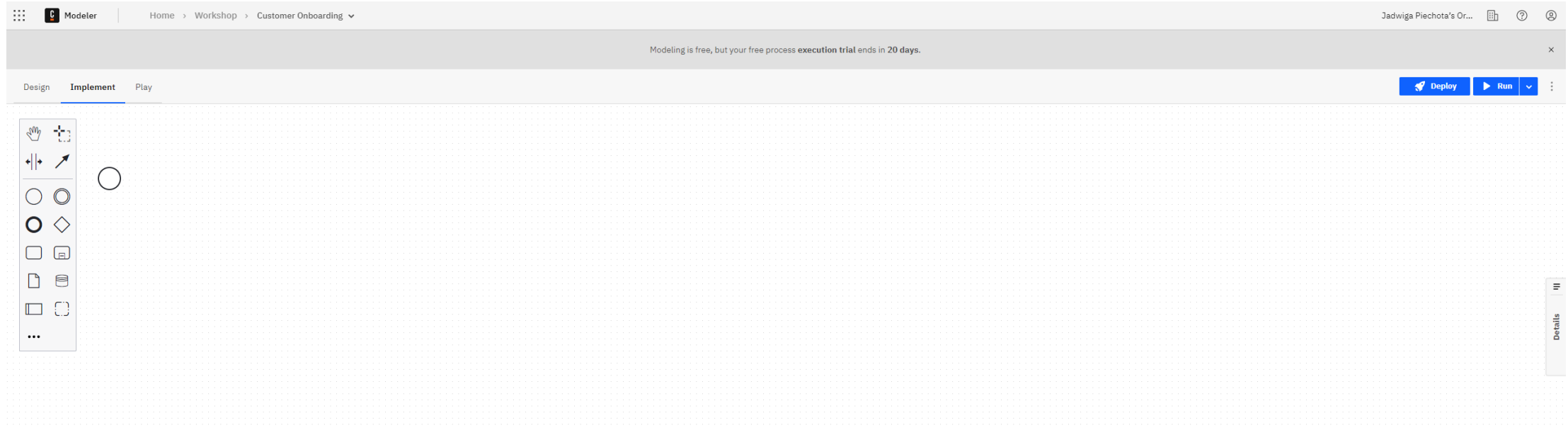


You ready to start designing your process!

# Create BPMN Diagram

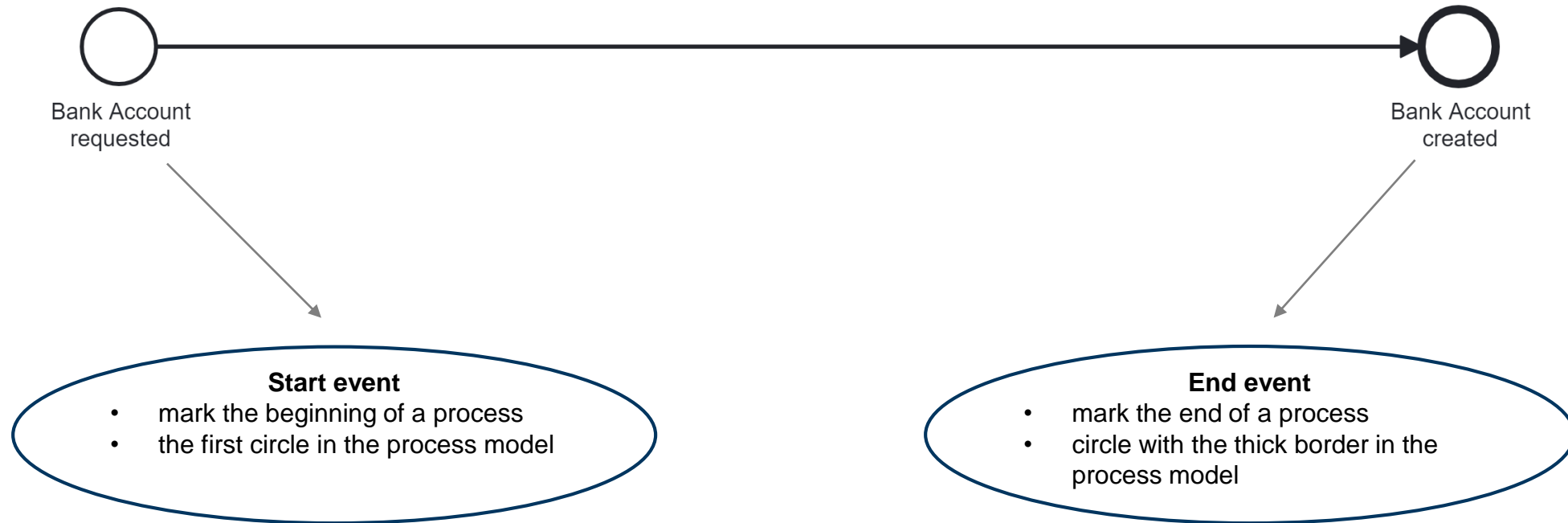


# BPMN Diagram created



# Step 1

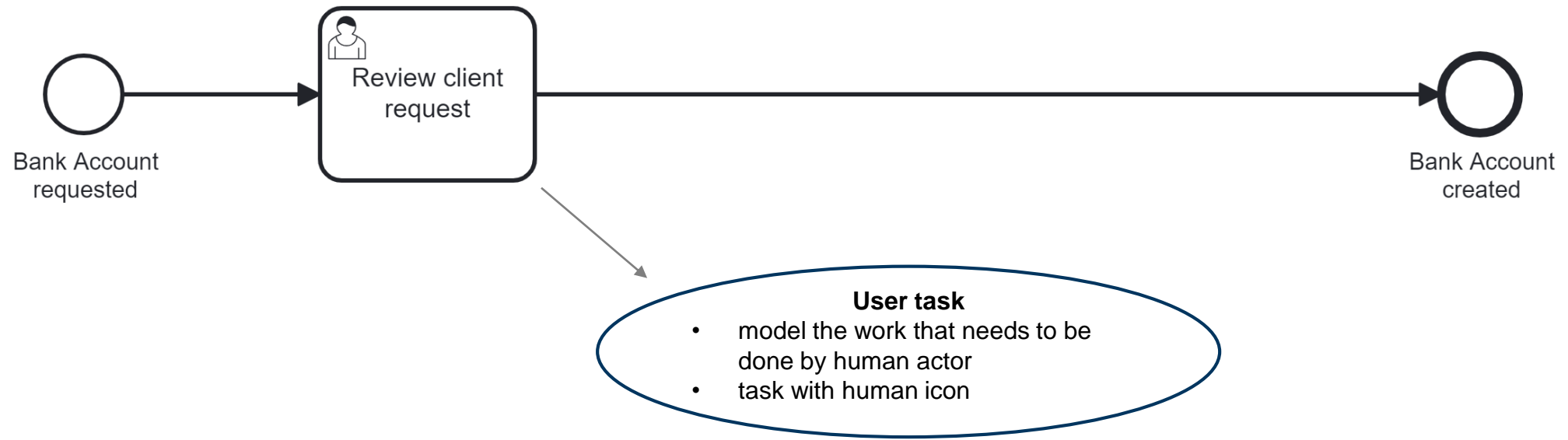
Customer is requesting a bank account. Next, the bank employee needs to review the request in order to decide if request can be processed. If the decision is successful, the other employee has to perform some steps in order to provision the account. Otherwise, request is rejected and account is not created.



Naming events: **object + verb reflecting the state**  
(when process is about to leave the event)

## Step 2

Customer is requesting a bank account. Next, the bank employee needs to review the request in order to decide if request can be processed. If the decision is successful, the other employee has to perform some steps in order to provision the account. Otherwise, request is rejected and account is not created.

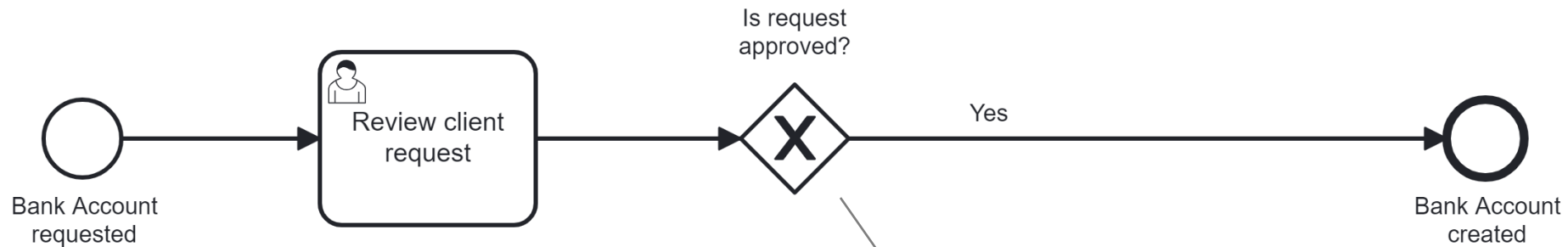


Naming tasks: **object + verb in the infinitive**  
(by doing this you consistently describe what you do with an object)



## Step 3

Customer is requesting a bank account. Next, the bank employee needs to review the request in order to decide if request can be processed. If the decision is successful, the other employee has to perform some steps in order to provision the account. Otherwise, request is rejected and account is not created.



### Exclusive (XOR) gateway

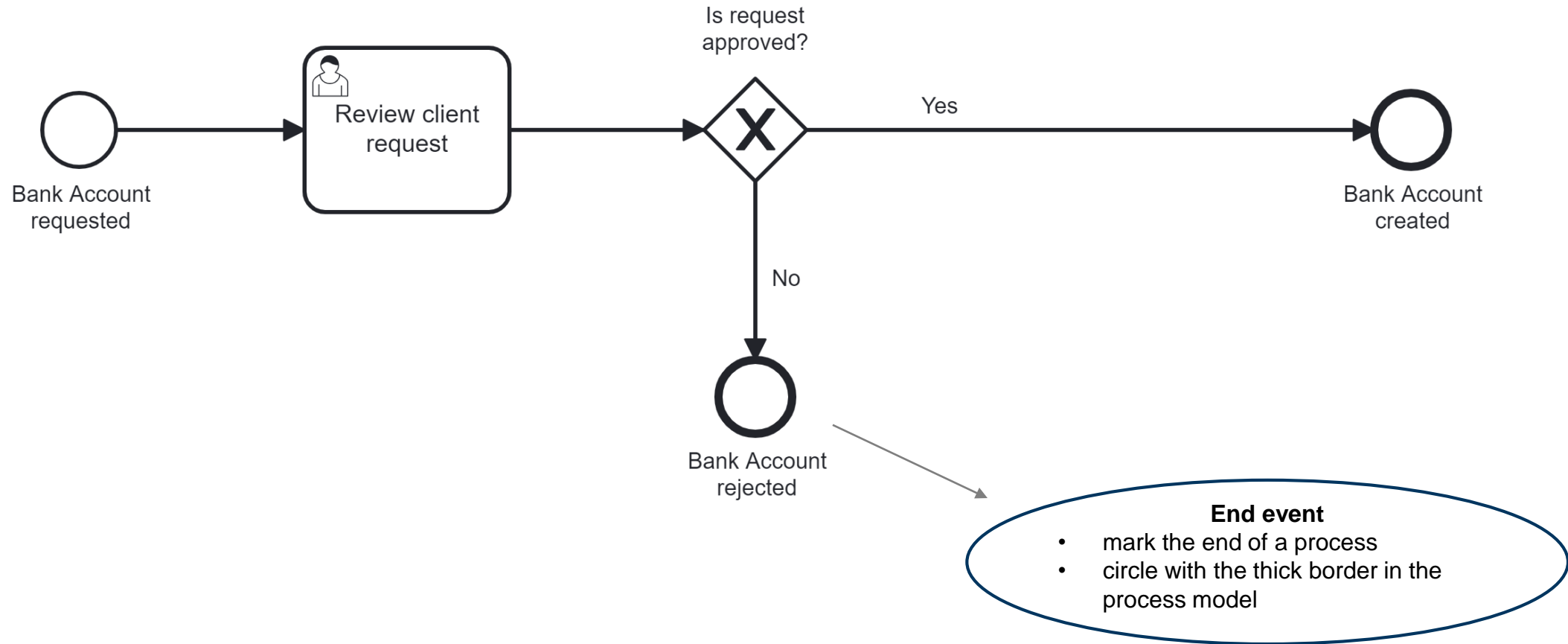
- allows you to make a decision based on data (i.e. process instance variables)
- condition is evaluated on entry (boolean expression)



Naming gateways: **label with a question**  
(label the outgoing sequence flows with the conditions they are executed under – formulate to answer the question posed at the gateway)

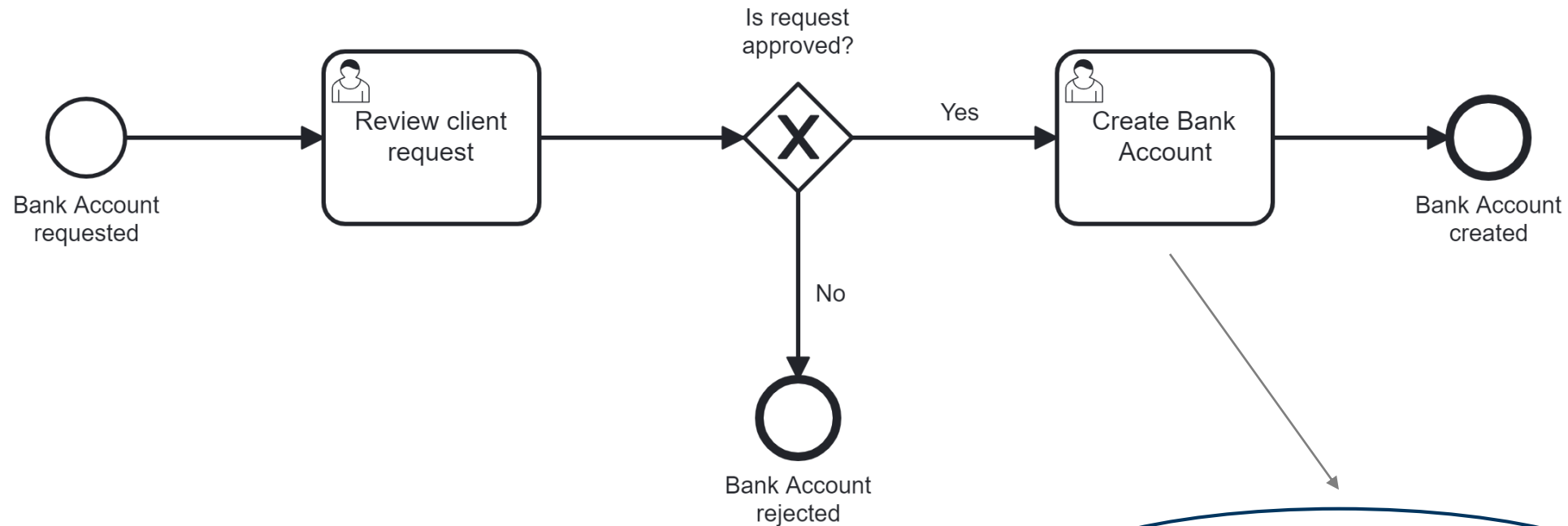
## Step 4

Customer is requesting a bank account. Next, the bank employee needs to review the request in order to decide if request can be processed. If the decision is successful, the other employee has to perform some steps in order to provision the account. Otherwise, request is rejected and account is not created.



## Step 5

Customer is requesting a bank account. Next, the bank employee needs to review the request in order to decide if request can be processed. If the decision is successful, the other employee has to perform some steps in order to provision the account. Otherwise, request is rejected and account is not created.

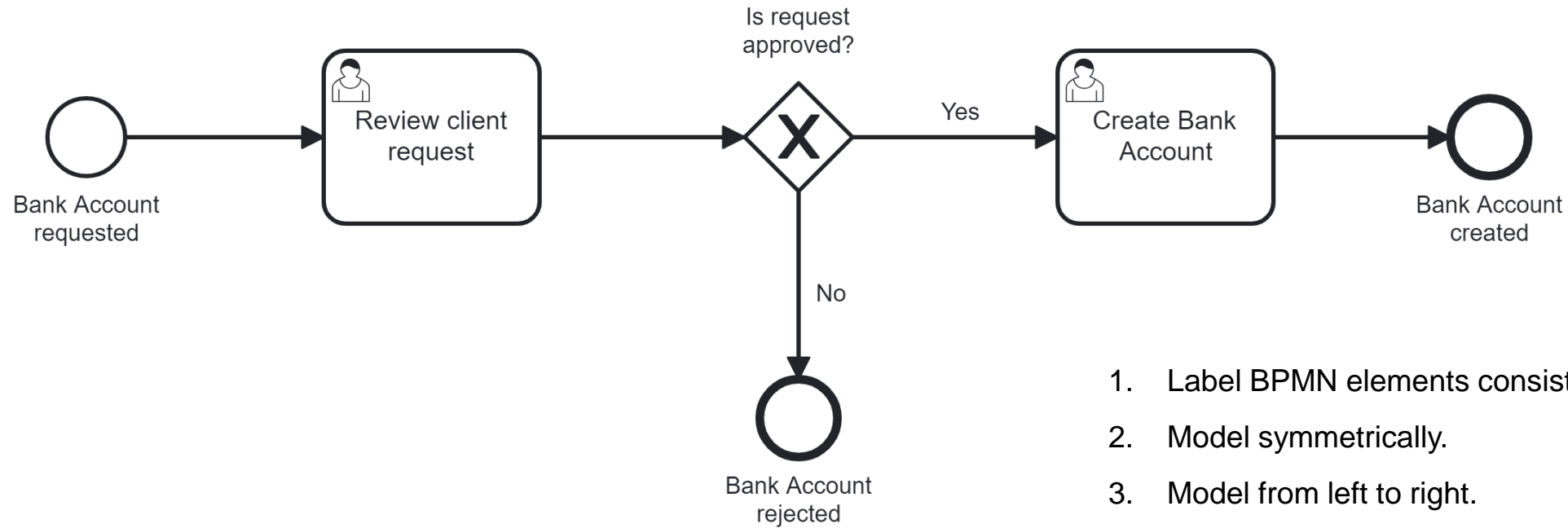


Naming tasks: **object + verb in the infinitive**  
(by doing this you consistently describe what you do with an object)

### User task

- model the work that needs to be done by human actor
- task with human icon

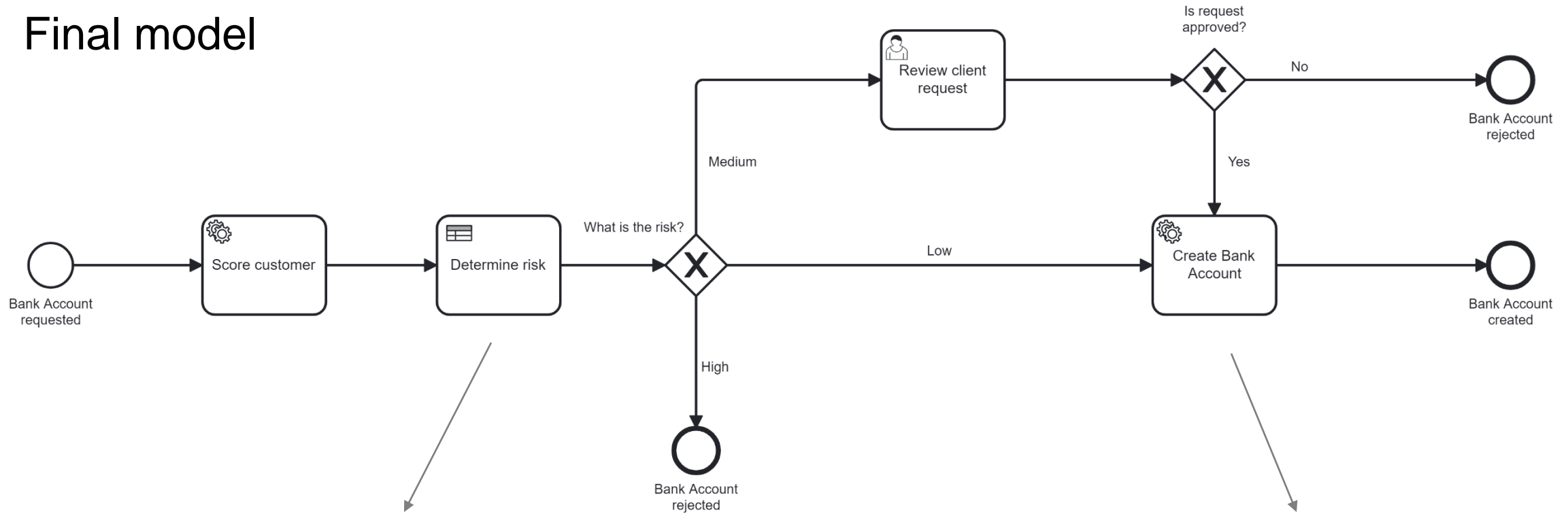
# Best practices



1. Label BPMN elements consistently.
2. Model symmetrically.
3. Model from left to right.
4. Create readable sequence flows.
5. Model explicitly.
6. Emphasize the happy path.
7. Avoid modelling retry behavior.
8. Avoid changes to symbol size and color.
9. Name technically relevant ID.

# Define your own process

# Final model



## Business rule task

- model the evaluation of a business rule – in our case decision modeled in DMN
- task with decision table

## Service task

- model the work item with a specific type – when service task is entered, corresponding job is created
- task with cog wheel

# Conclusion

1. Implementing such processes in Java or other equivalent programming language would be much less readable, more complex. It is much easier to design and define a process with visual form.
2. Monitoring tools are given – there is no need to prepare another solution for collecting and analyzing the logs.  
We can easily check at which step each process instance is.
3. Process definition can be understood by non-technical users.
4. No coding skills required in order to design a process.

# Thank you !

## Q & A



**Goldman  
Sachs**