**Camunda Workflow:**

Why do we need a workflow in a banking scenario?  
Camunda Workflow is an open-source platform that uses a workflow engine and decision engine to automate business processes. Organizations can use it to automate business processes and decision processes.

Ex: Loan oriented – bank employee can collect all the documents and evaluate all the documents for the customer and take the manager approval for creating a loan account then finally approved the loan. These are the scenario we call it as workflow system.

What is meant by

**The Practical BPMN 2.0 Master Class (Udemy Course)**

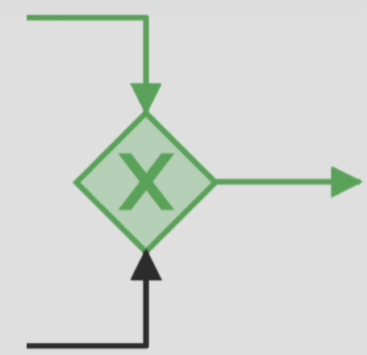
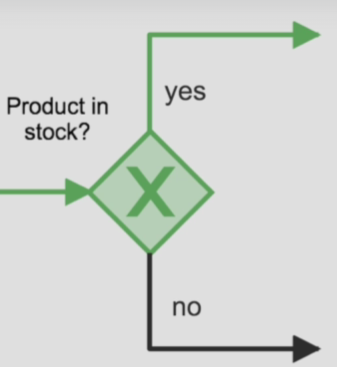
1. Why BPMN (Business process model notation)?

* Processes of algorithm of businesses. (Jakob Freund, Camunda CEO Said)
* Easy to understand, executable format and widely used.
* Learning BPMN 2.0 is always the right choice!

**Event, activities and gateways**:

Start and end event:   
\* The Start event triggers the process.  
\* End event defines the state that terminates the process

**Activity**: An Activity or task is a unit of work – the job to be done.

**Gateways**: Gateways add logic to the process flow!

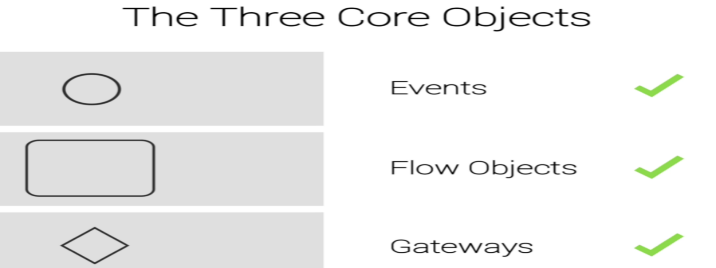
1. **Exclusive gateway**

* Defines a decision point.
* Only one path can be taken.

1. **Exclusive gate (Closing)**:

* Connects branches
* No logic involved

**Exclusive gateway** **Exclusive gate (Closing)**:

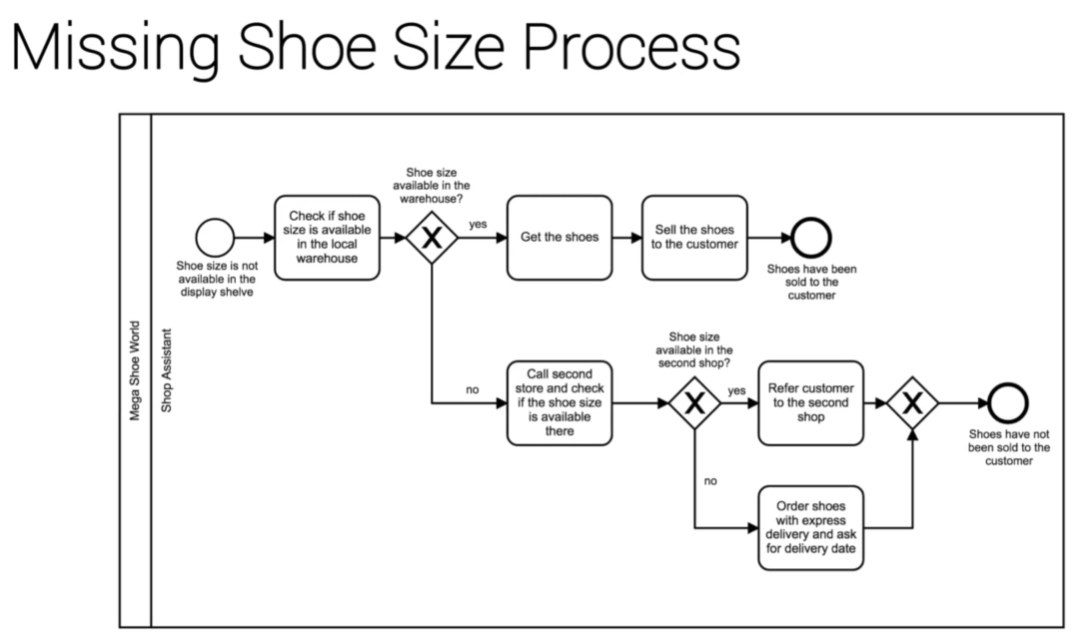
**There are three core objects are**:

1. Events
2. Flow objects
3. Gateways

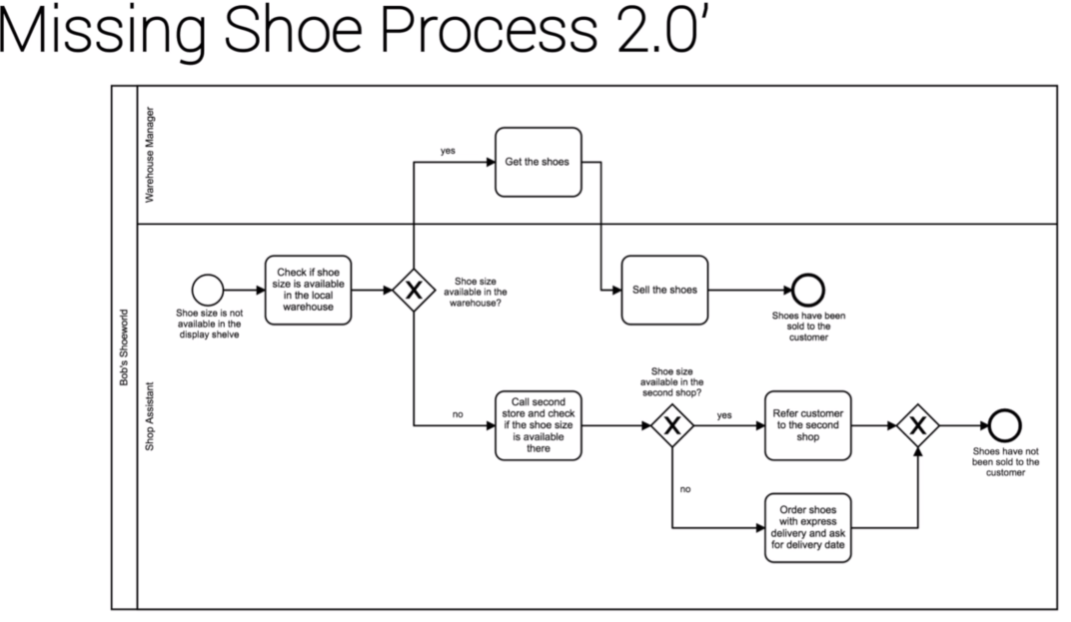
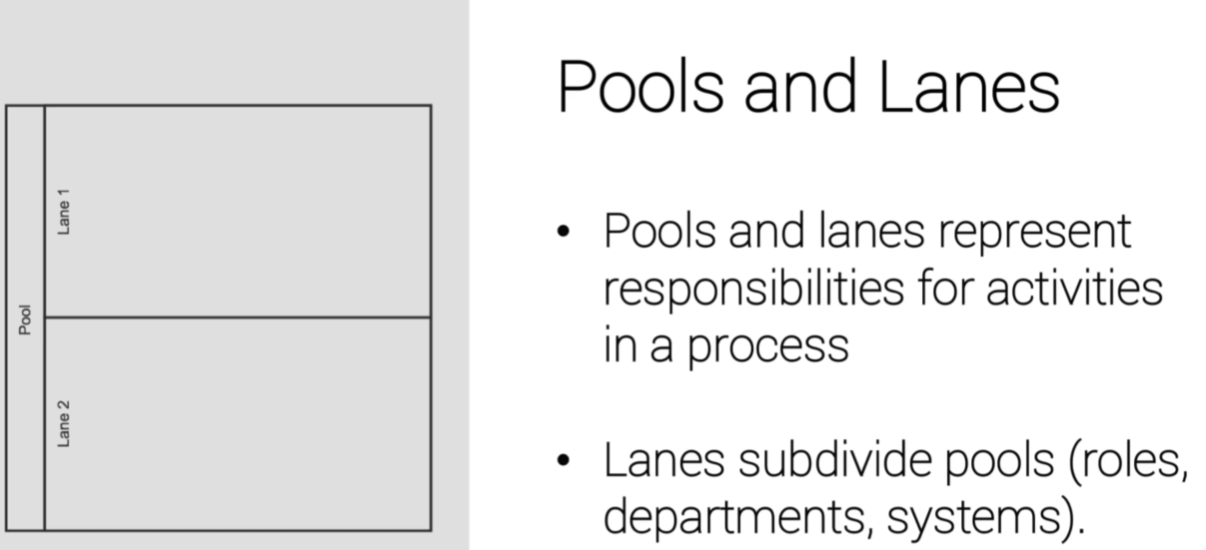
These three core objects are real backbone of the BPMN

**Example of why the need of BPMN?**

Ex: **Shoe shop** - One employee first week as a shop assistant his name is max. One of the Max’s customers can’t find the right shoe size! His favourite one shoes. Max wonders how to solve the situation. Then Max decided to look at the business process!



* This is the way procedures can be communicated in a **clear** and **expressive** way.
* Processes change typically because of growth or technological change.
* All new shop assistants have no clue how to handle their new tasks.
* The process can also **change because of growth**.
* For example, a warehouse manager is hired.

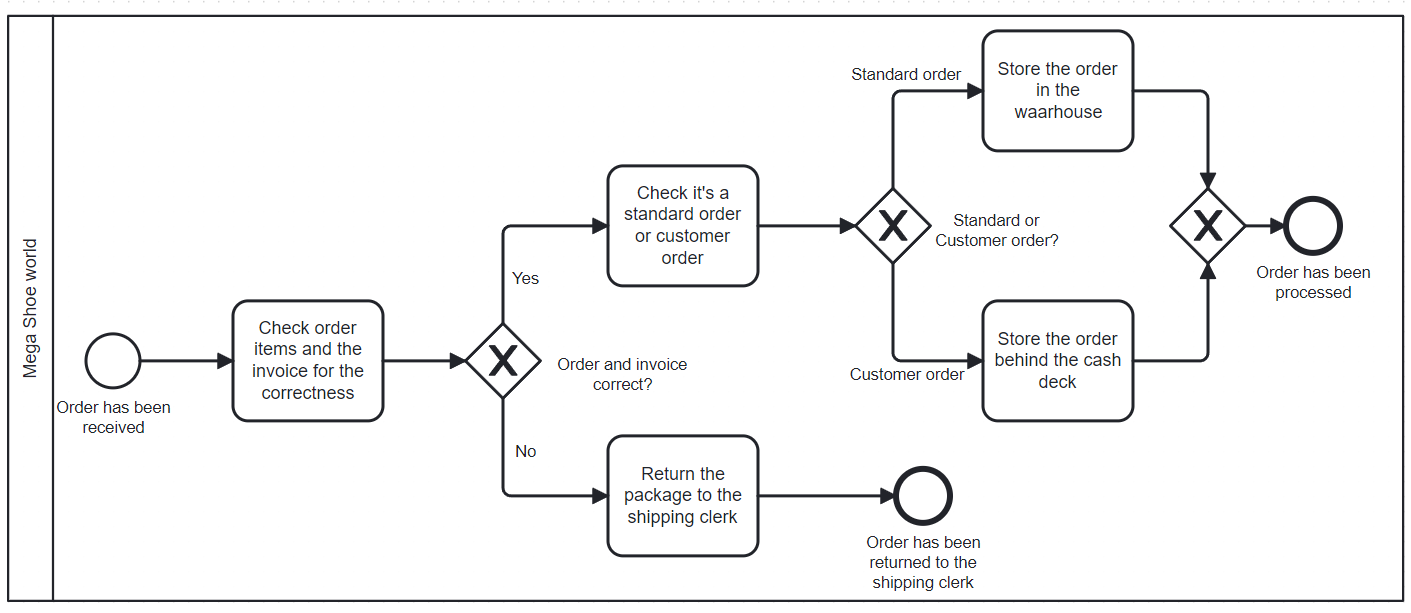
**Modelling Tool Introduction**

<https://demo.bpmn.io/> This is the official website where we create our BPMN diagram through online

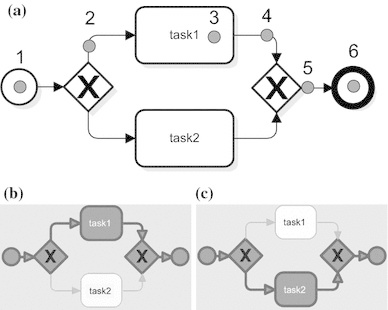
Max is facing a new challenge. He receives a shipment from the wholesaler and doesn’t know what do next. His boss tells him:

No worries max. First you check weather the order and the invoice are correct. If not, you give the package back to the shipping clerk. If it’s correct you check if it’s a standard order to refill the stock, or if it’s an order for a specific customer. If it’s a standard order you simply put the shoes into the warehouse. If it’s an order for a customer, put the shoes behind the cash deck and that’s it.

Below is the flow diagram for the above scenario.



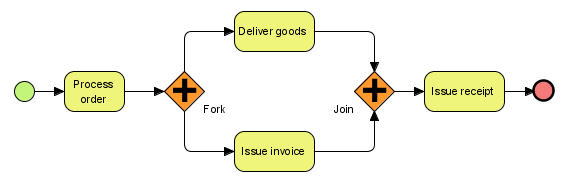
**Token concept:**

Imagine the token as a ball or marble. The **token** rolls through the process, from start to end event. The **token** passes all kinds of tasks and gateways that impact it.

The real **strength** of the **token** **concept** is to **visualize** how different **gateways** work.

Note: Build **mental bridges** to clarify **distinguish** the **gateways**.

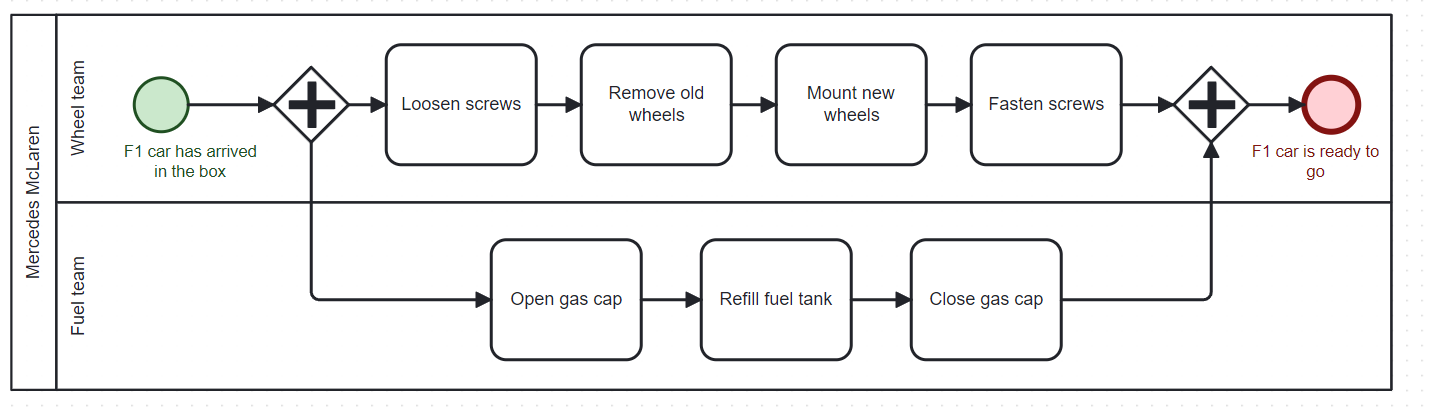
**Parallel Gateway**:



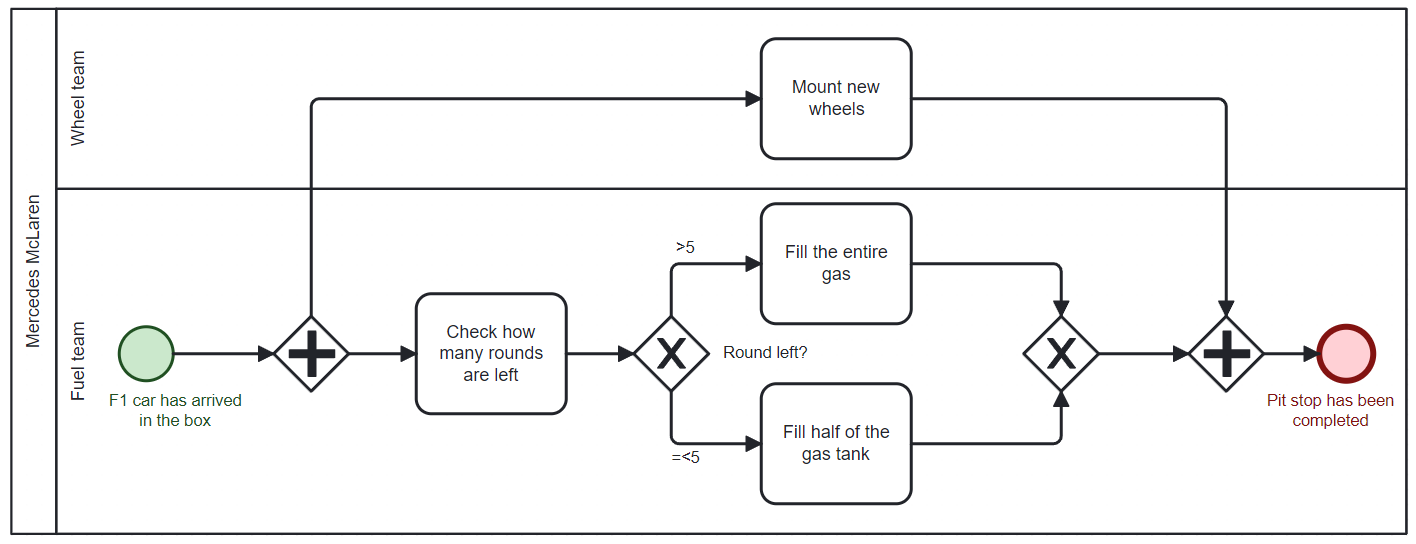
* No decision making
* All outgoing branches are activated.
* The token is cloned. One cloned token per outgoing branch.

**Example**: Process racing

The pit stop process: To become the champion Max must win today.  
Max’s team must: change the tyre and fill the fuel tank simultaneously. So, let’s see how the situation can be expressed in BPMN diagram.

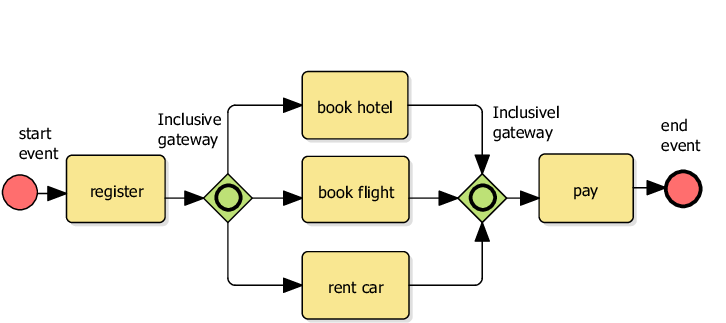
Suppose parallel gateway means the action needs to both the ways. If wheel team complete them work, they should wait for until fuel team completes them job vies versa. Ex: First token completes the job wait for second token or if second token completes first wait for the first token vies versa.The token concept empowers to understand even complex business processes.

Max has an idea. As his fuel team acted a bit slow the last time, he talks to his manager and tells him: When I arrive at the box, both teams start to work in parallel as usual. The wheel team simply changes the wheels. The fuel team however, first checks how many rounds are left. If 5 or less rounds are left, the fuel team will fill only half of the gas tank. If more than 5 round are left, the entire gas tank needs to be filled. So, the fuel refill is faster when 5 or less rounds are left. Your job now is to help the head of mechanics to map the process so he can effectively communicate the procedure with his team.

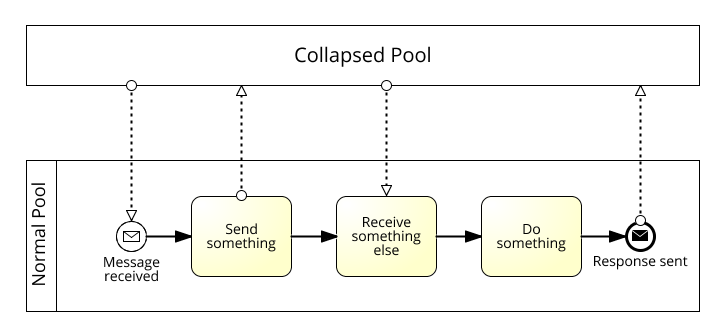


**Inclusive gateway | Message Event | Collapsed Pool**

**Inclusive gateway**: Defines a decision any combination of paths can be taken. Is informed about all preceding token flows Example below:



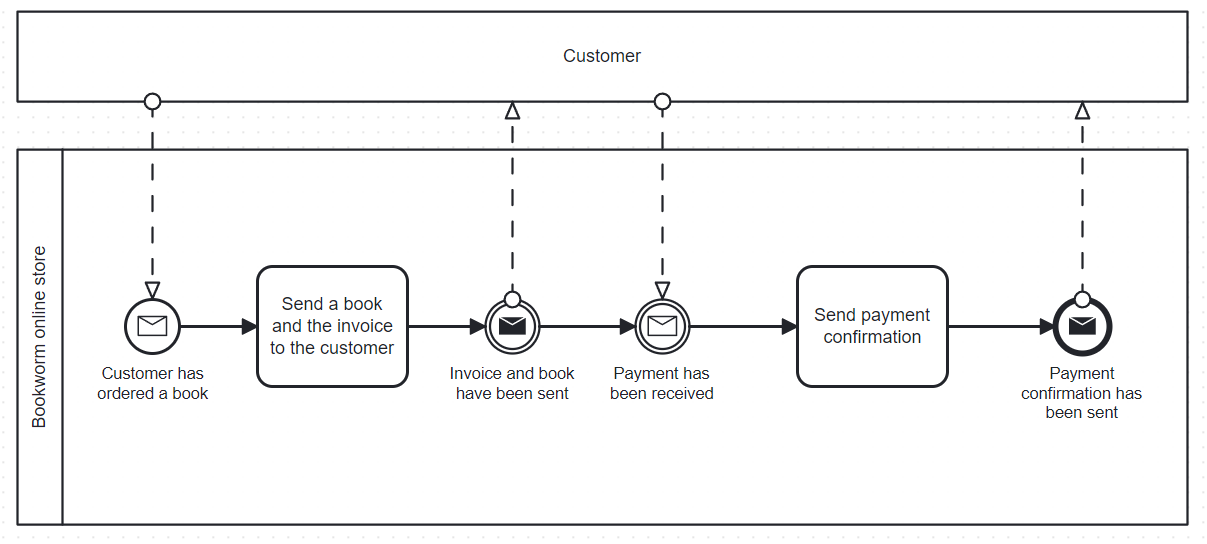
**Collapsed pool**: Don’t show the process flow – like a black box. Collapsed pools are used to represent external process participants.



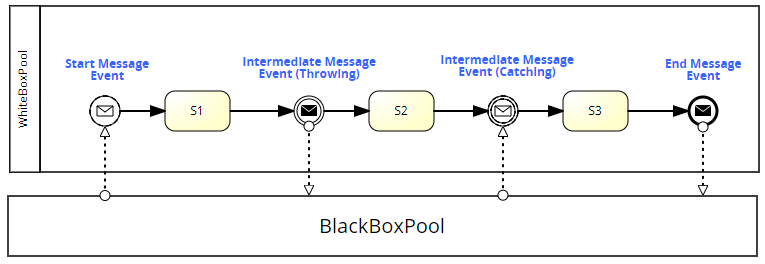
**Message Event**: It represents incoming or outgoing messages. Is also used for sending or receiving goods, money or other things. There are 4 types of message events.

Why do we need 4 message event types?  
A message can be sent or received and occur at different stages. So, if our process is completed with shipment goods. The end message event is the perfect choice.

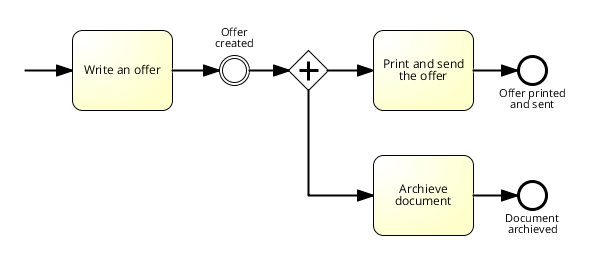
Let’s apply the 4 message event types to a small process:



* Tasks and Events are fundamentally different!
* The task describes the action of sending the book and invoice
* The event states that book and invoice have been sent. No actions are taken!
* The throwing intermediate event is “nice to have”.
* The catching intermediate event defines a waiting point!

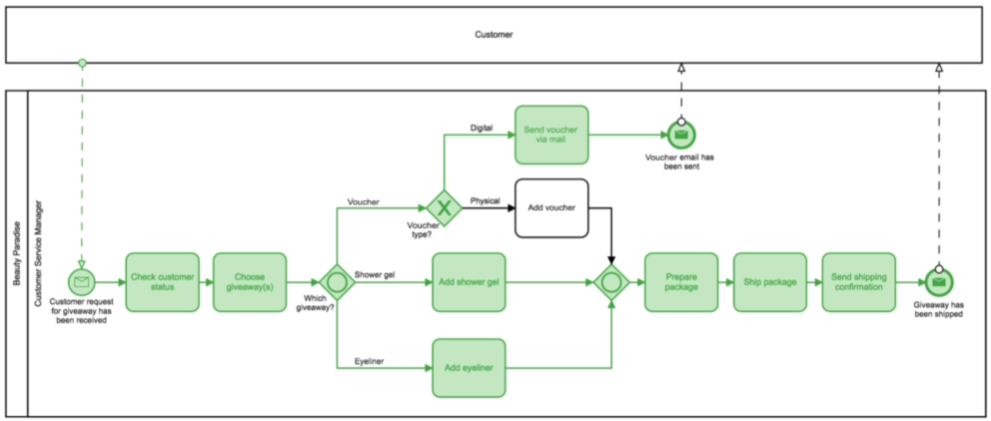


**Plain Intermediate Event:**



**Practical - Beauty Process (Part 1)**

It’s time to apply the new BPMN elements!   
The process takes place in the “**Beauty Paradise**”, a large cosmetics manufacturer. Susan is responsible for customer satisfaction. Susan responds customer who ask for a product sample. It’s very repetitive. Susan automates some steps:

The documentation of the process is the first step to improve and automate it.

Assumption – Inclusive gateways only communicate between each other. This assumption is wrong. It is possible that the number of expected tokens changes!

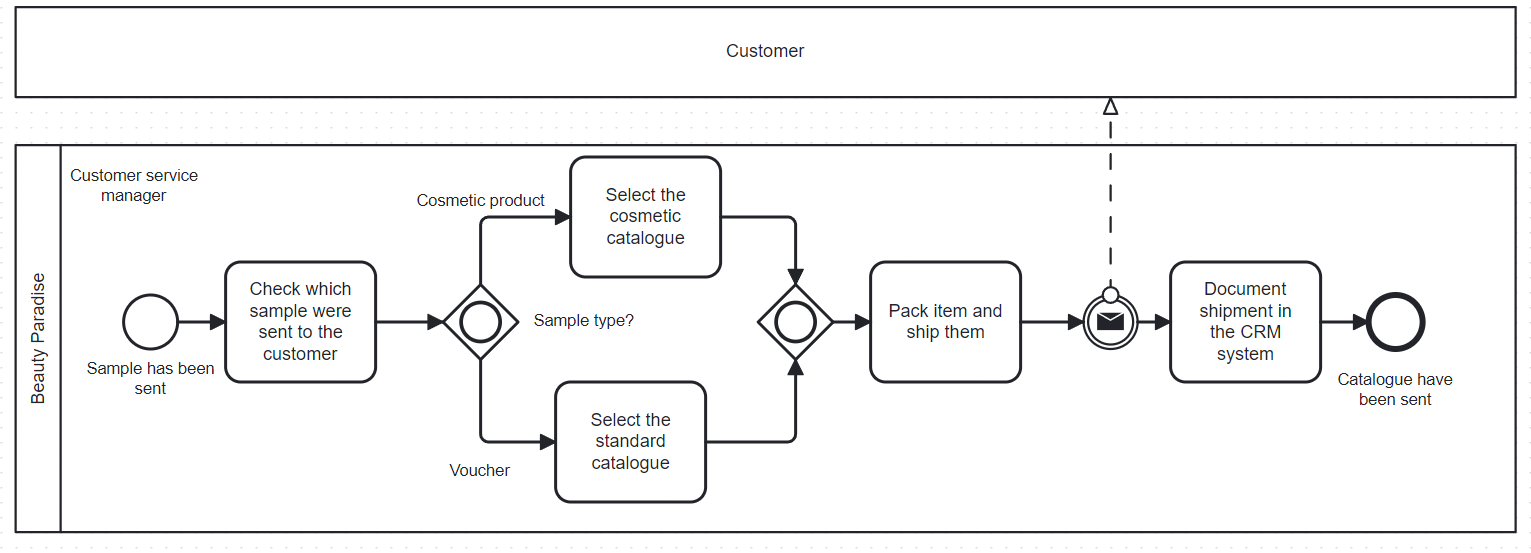
What about the more complex case? All three giveaways are selected. Are two active and events even allowed? Yes  
The process doesn’t end with the first end event being activated. Both tokens must arrive at their end events.

Why do we need to define two separate end events?  
Both end events define distinct situations which lead to different follow up activities. When shipping the giveaways activities like “paying the shipping company” and “track the package are” following. We need two end events because each of them triggers different follow up activities.

Susan is still keen on further optimizing the customer experience! Therefore, she chats with her boss and tells him that **she wants to develop a new process** that works like this:

After the sample has been sent, I will dispatch a product catalogue to the customer. We have only 2 product catalogues. I will check which samples I've sent to the customer. If he or she got a cosmetic product, I'll send our cosmetic catalogue. If he or she got a voucher, I'll send our standard catalogue. If he or she received both, I'll send both catalogues. After that, I'll document the catalogue shipment in our CRM system.

our job now is to **map this new process** so Susan can effectively communicate the procedure with her team.



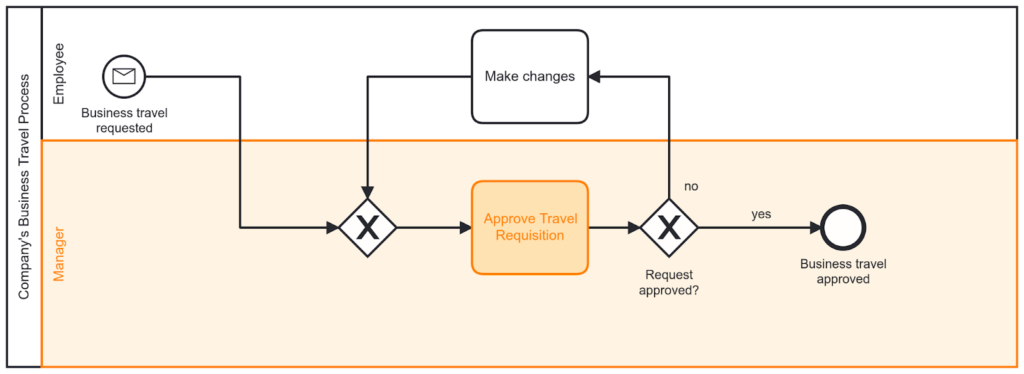
**BPMN Traps**

**Deadlock | multi-merge Trap**

The traps are caused by mixing up the gateways.

**Deadlock**: The deadlock describes the trap of getting “**stuck for ever”.**

* Opening Exclusive gateway
* Closing parallel gateway
* One token is issued. Two tokens are expected the process gets stuck.



**The multi-merge**: The Multi-merge trap is based on the misleading use of gateways.

* Opening parallel gateway
* Closing Exclusive gateway
* Two tokens are issued. Several tokens rolling through the entire process. Instead of being merged into a single token again.

**Multi-merges** are the opposite of **deadlocks**. They often occur when Gateways are used incorrectly and lead to multiple sequence flows merging at a single-flow element. Avoiding this is necessary, since the process often shows an unexpected behaviour within a **multi-merge**.

**Takeaways**:  
Multi-merge Deadlock

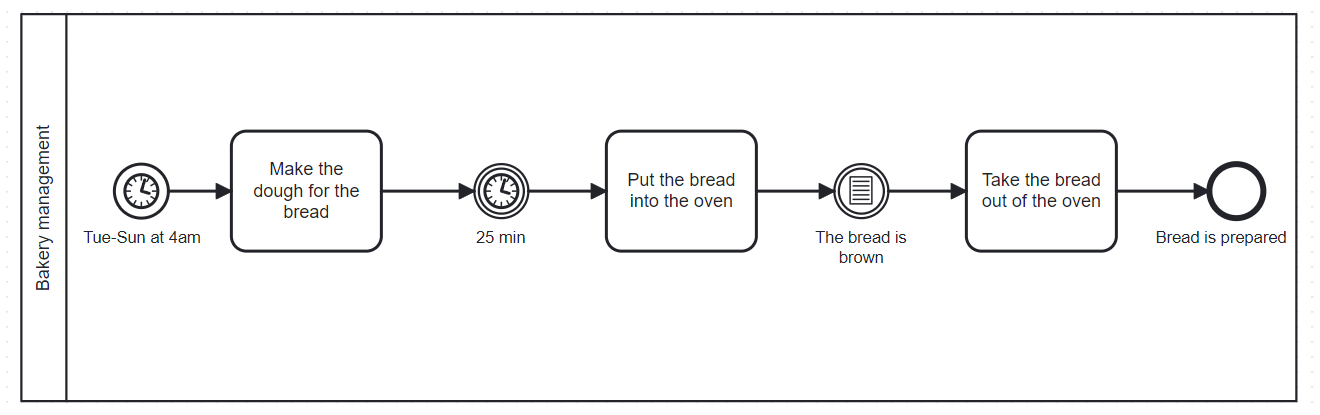
* Technically correct \* Technically incorrect
* Affects expressiveness of process model. \* Process becomes incompletable.
* Multiple tokens circulate. \* Tokens gets stuck.

**Timer and Conditional Event | Event Base Gateway | Attached Events**

**The Timer Event**: Represents a date or a period.

Start time event and Intermediate timer event only available, because somebody needs to dispatch of the message a person clicks on the sent button with a time event hour its different this time is something record manipulate at all its simply happens that’s why we don’t have any **throwing time events.**

**The Conditional Event**: Defines an event which is triggered if a given condition is evaluated is true. Eventually for conditional event also there is no throwing conditional event.  
The conditional event can also be waiting point. For example,



A screenshot of a diagram

Description automatically generated**Attached Events**: The activity gets cancelled through the occurrence of the event. If the event occurs, an alternative path is taken.

**Event Based Gateway**: Reacts of occurring events and only one path can be taken. Reuse of the closing exclusive gateway.

A diagram of a diagram

Description automatically generated