



# Process Modeling

MOMNA ZANEB

16 MARCH 2022



# Process Modeling

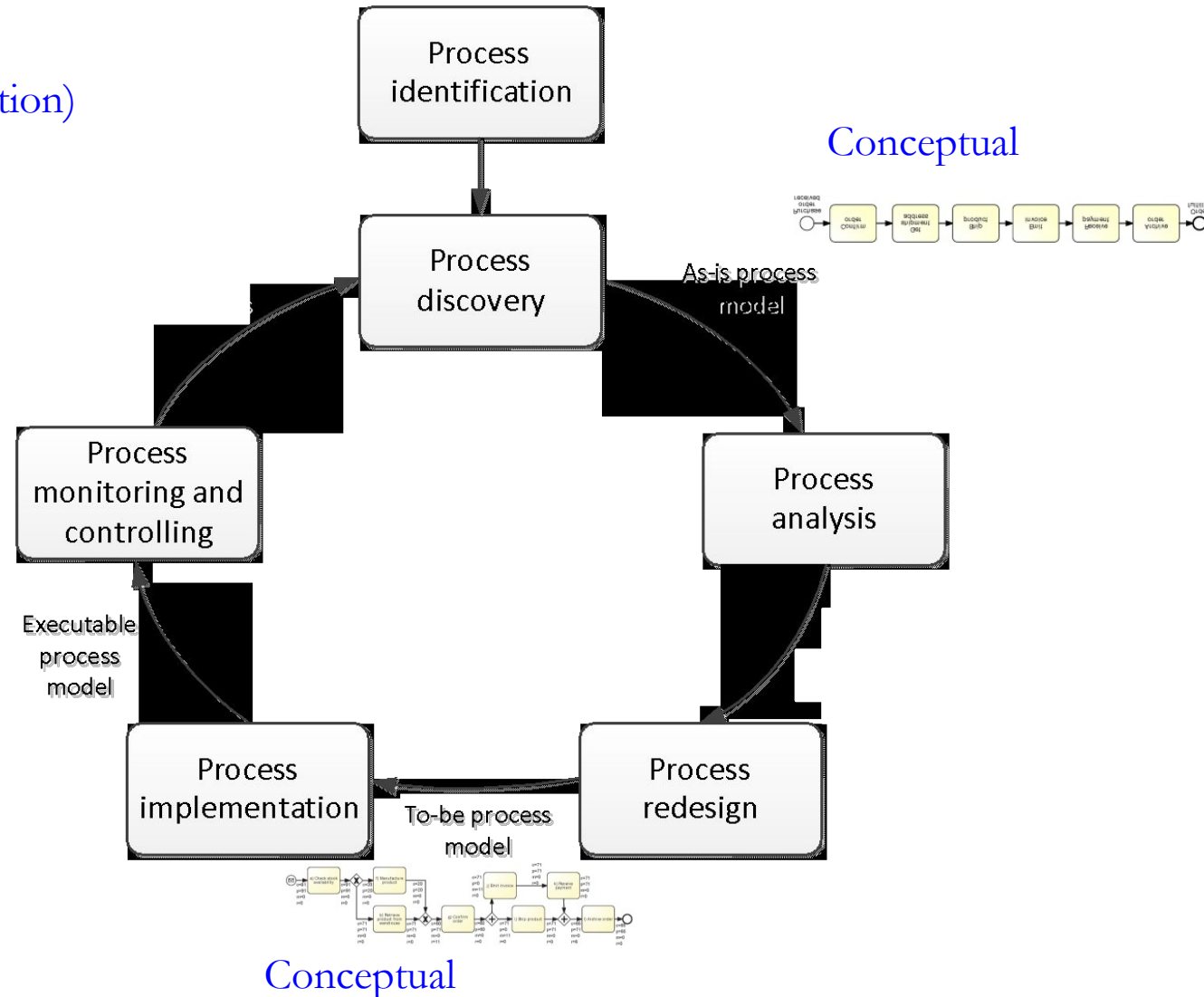
- Basic BPMN Notations
- Modeling Examples

# Purposes of process modeling

- Communication
- Documentation
- Analysis (e.g. simulation)

Executable

- Automation
- Testing



# Business Process Model and Notation (BPMN)

- Both for conceptual and executable models
- Supported by numerous tools: [bpmn.org](http://bpmn.org) lists over 70 tools, incl.
  - **Signavio** (we'll use the academic edition – [academic.signavio.com](http://academic.signavio.com))
  - Bizagi Process Modeler
  - Cameo Business Analyst
  - Camunda <https://camunda.com/>



# Reasons for Process Modeling

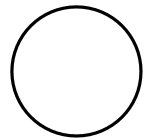
- To document an existing process clearly
- To use as a training aide
- To use as an assessment against standards and compliance requirements
- To understand how a process will perform under varying loads or in response to some anticipated change
- As the basis for analysis in identifying opportunities for improvement
- To design a new process or new approach for an existing process
- To provide a basis for communication and discussion
- To describe requirements for a new business operation

# BPMN from 10,000 miles...

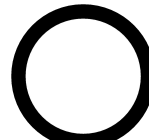
A BPMN process model is a graph consisting of four types of **core elements**:



activity

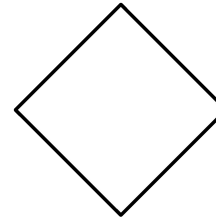


start

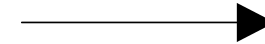


end

event



gateway



sequence flow



## BPMN -Events

- Event denotes something that happens
- Classifications –
  - Catching—triggered by external event
  - Throwing—generating an output
- Types –
  - Start Event-acts as a trigger for the process
  - End Event-represents the result of a process –
  - Intermediate Event-represents something that happens between the start and end events

# BPMN -Activities

- Activity describes the kind of work that must be done
- Types –
  - Task-represents a single unit of work that is not or cannot be broken down to a further level of business process detail
  - Sub-Process-used to hide or reveal additional levels of business process detail
  - Transaction-a form of sub-process in which all contained activities must be treated as a whole



## BPMN -Gateway

- BPMN -Gateway
- A Gateway determines forking and merging of paths depending on the conditions expressed

# Let's start modelling

## Order-to-cash

An order-to-cash process is triggered by the receipt of a purchase order from a customer. Upon receipt, the purchase order has to be checked against the stock to determine if the requested item(s) are available. Depending on stock availability the purchase order may be confirmed or rejected.

If the purchase order is confirmed, an invoice is emitted and the goods requested are shipped. The process completes by archiving the order.

# Let's start modeling – break it down

## Order-to-cash

- An order-to-cash process is triggered by the receipt of a purchase order from a customer.
- Upon receipt, the purchase order has to be checked against the stock to determine if the the requested item(s) are available.
- Depending on stock availability the purchase order may be confirmed or rejected.
- If the purchase order is confirmed, an invoice is emitted and the goods requested are shipped. The process completes by archiving the order.

# Let's start modeling – break it down

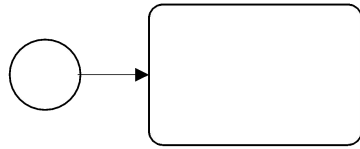
## Order-to-cash

- An order-to-cash process is triggered by the receipt of a purchase order from a customer.
- Upon receipt, the purchase order has to be checked against the stock to determine if the the requested item(s) are available.

# BPMN Model

## Order-to-cash

.....✓



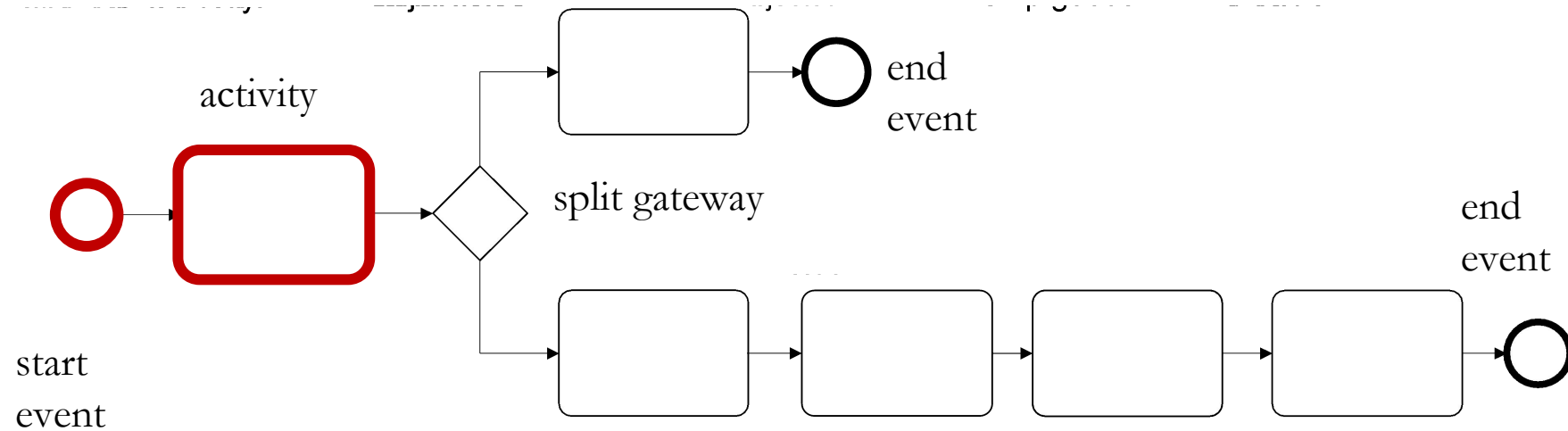
# Let's start modeling – break it down

## Order-to-cash

- An order-to-cash process is triggered by the receipt of a purchase order from a customer.
- Upon receipt, the purchase order has to be checked against the stock to determine if the the requested item(s) are available.
- **Depending on stock availability the purchase order may be confirmed or rejected.**
- **If the purchase order is confirmed, an invoice is emitted and the goods requested are shipped. The process completes by archiving the order.**

# BPMN Model

## Order-to-cash



### Naming conventions

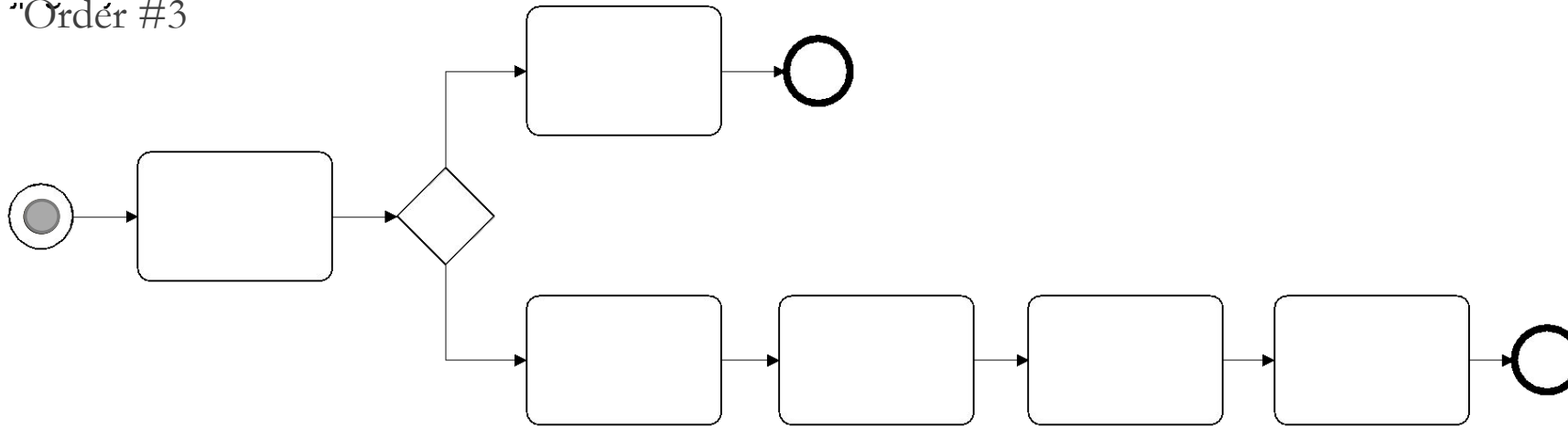
- Event: noun + past-participle verb (e.g. insurance claim lodged)
- Activity: verb + noun (e.g. assess credit risk)



# Execution of a process model

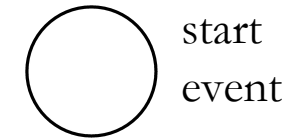
## The “token game”

- Order #1
- Order #2
- Order #3

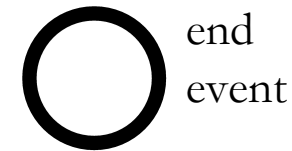


## A little bit more on events...

A *start event* triggers a new process instance by generating a token that traverses the sequence flow (“tokens source”)

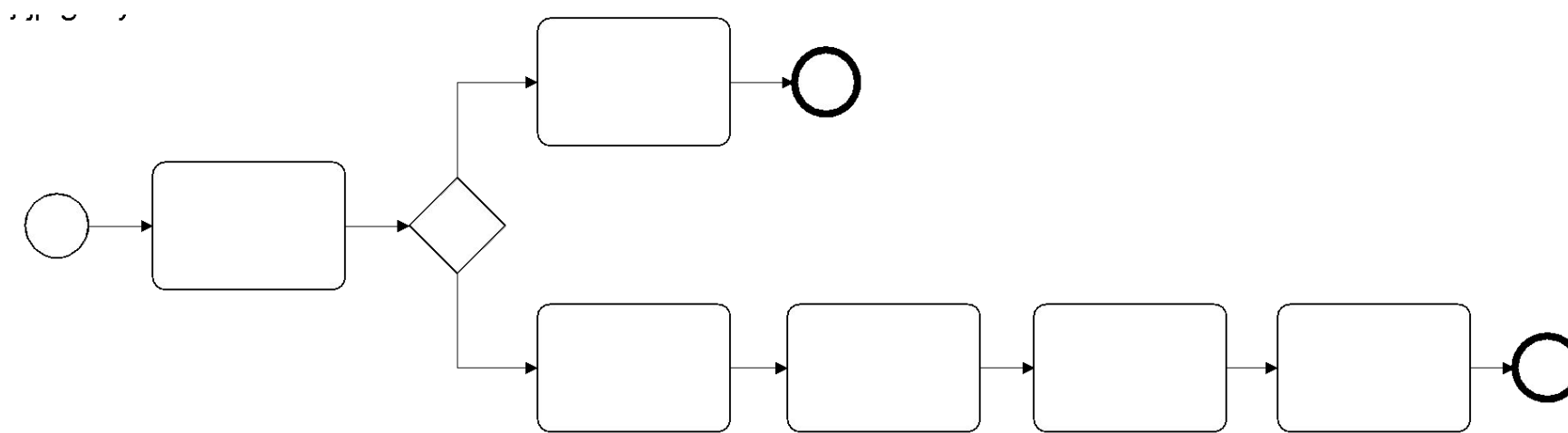


An *end event* signals that a process instance has completed with a given outcome by consuming a token (“tokens sink”)



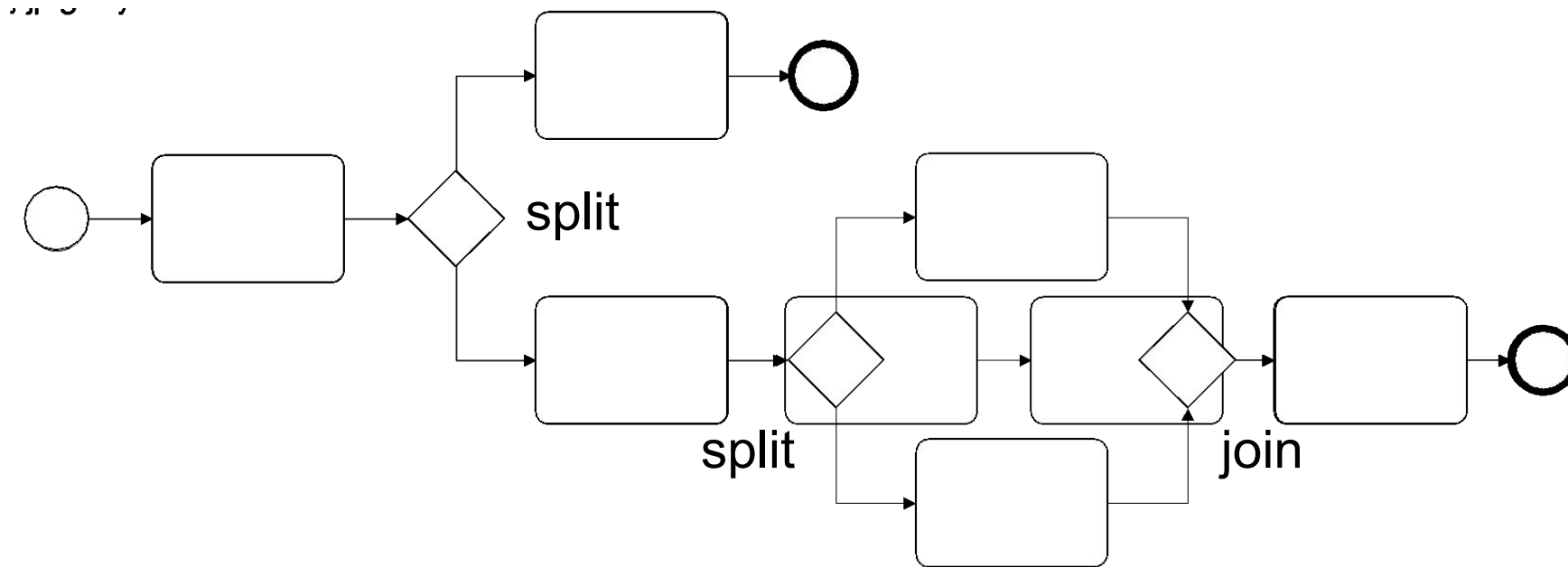
## Order-to-cash example revisited...

[...] If the purchase order is confirmed, **an invoice is emitted and the goods requested are shipped (in any order)**. The process completes by archiving the order. [...]

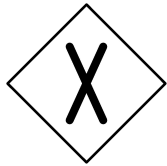


# First try

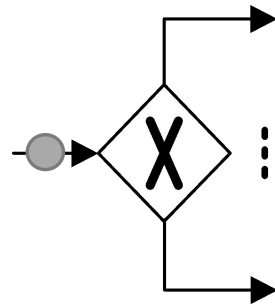
## Order-to-cash



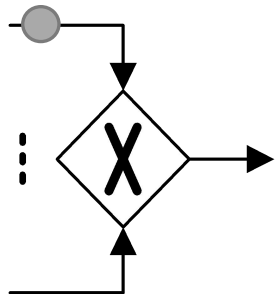
## A little more on gateways: XOR Gateway



An *XOR Gateway* captures decision points (XOR-split) and points where alternative flows are merged (XOR-join)



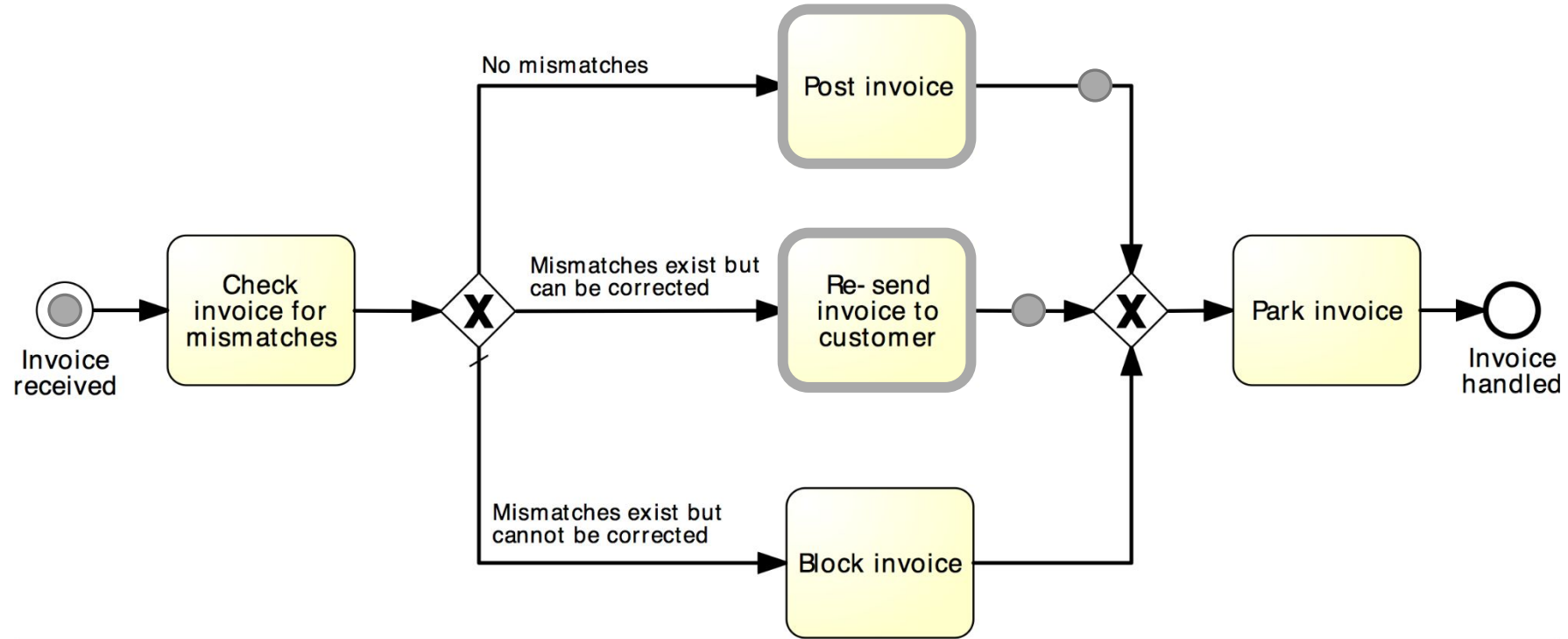
*XOR-split* takes **one** outgoing branch



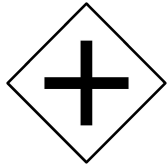
*XOR-join* proceeds when **one** incoming branch has completed

# Example: XOR Gateway

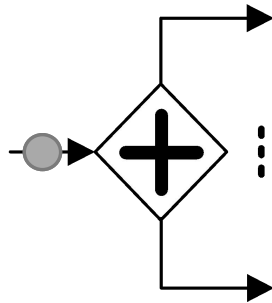
## Invoice checking process



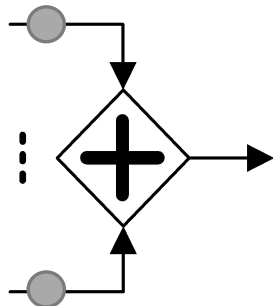
## A little more on gateways: AND Gateway



An *AND Gateway* provides a mechanism to create and synchronize “parallel” flows.



*AND-split* takes **all** outgoing branches

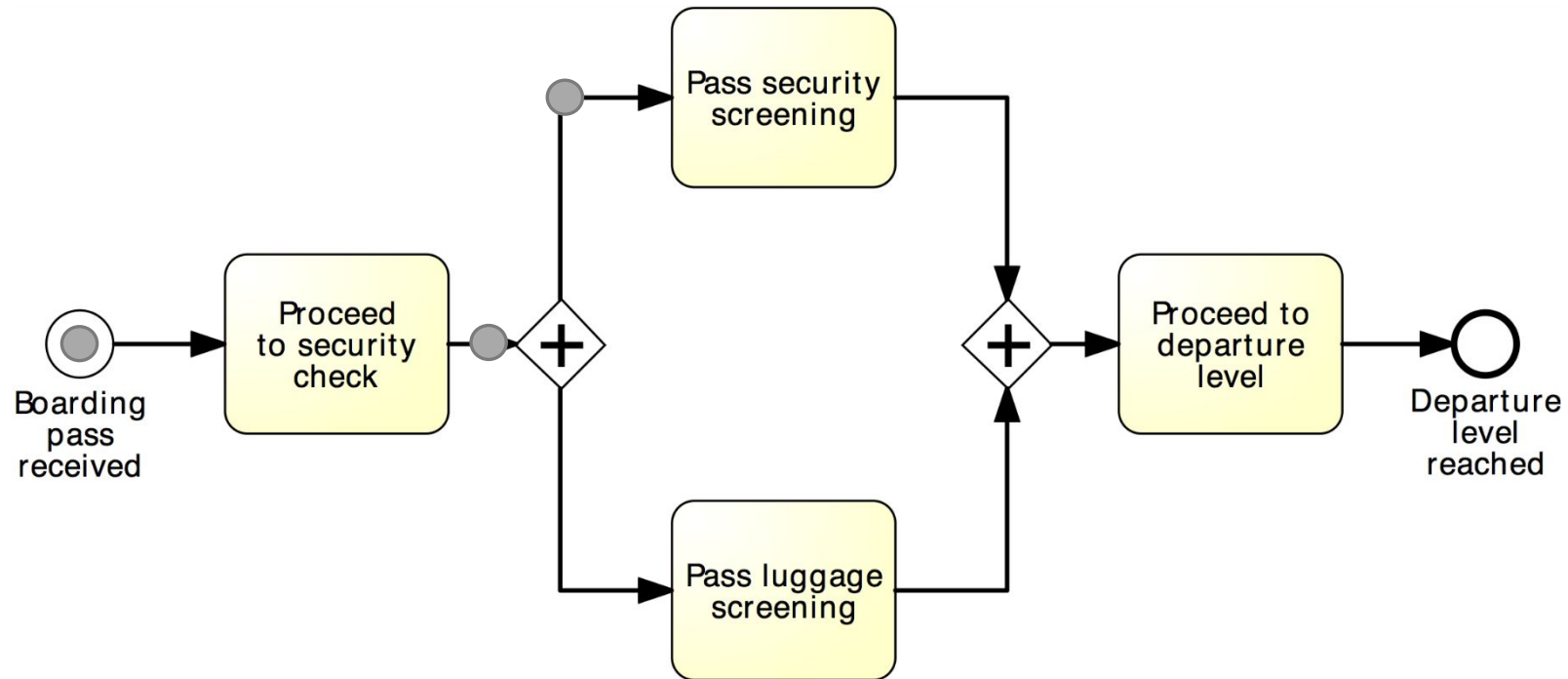


*AND-join* proceeds when **all** incoming branches have completed

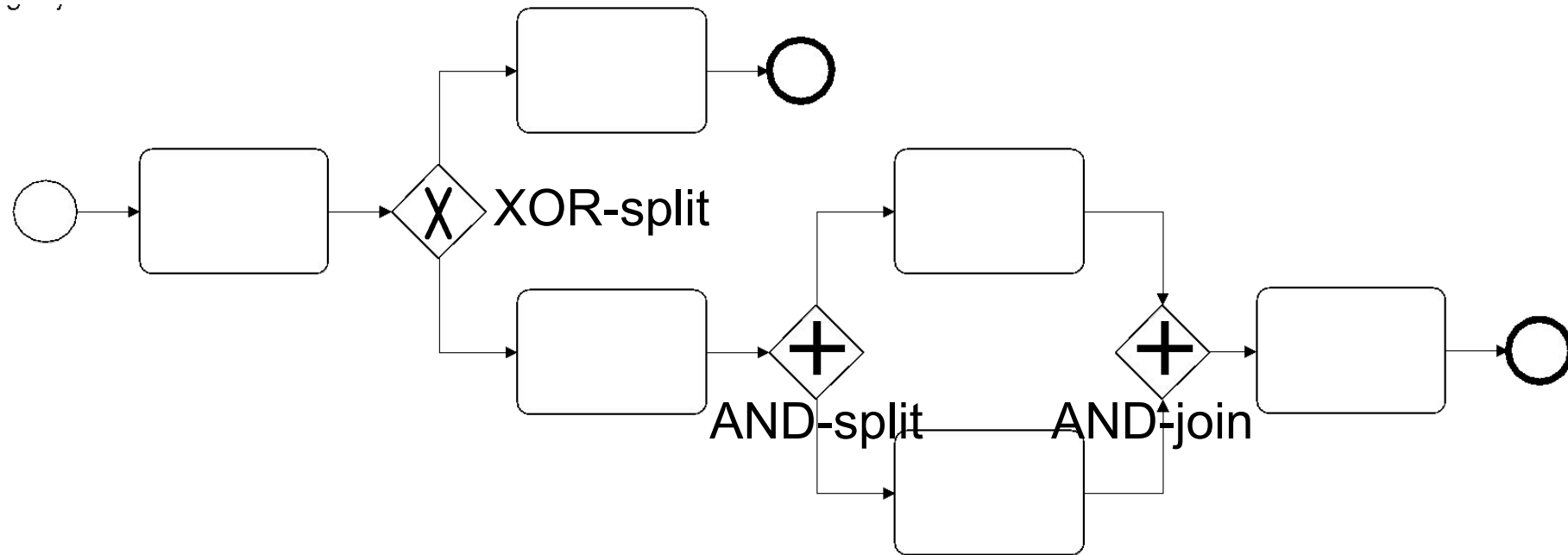


# Example: AND Gateway

## Airport security check



# Revised order-to-cash process model



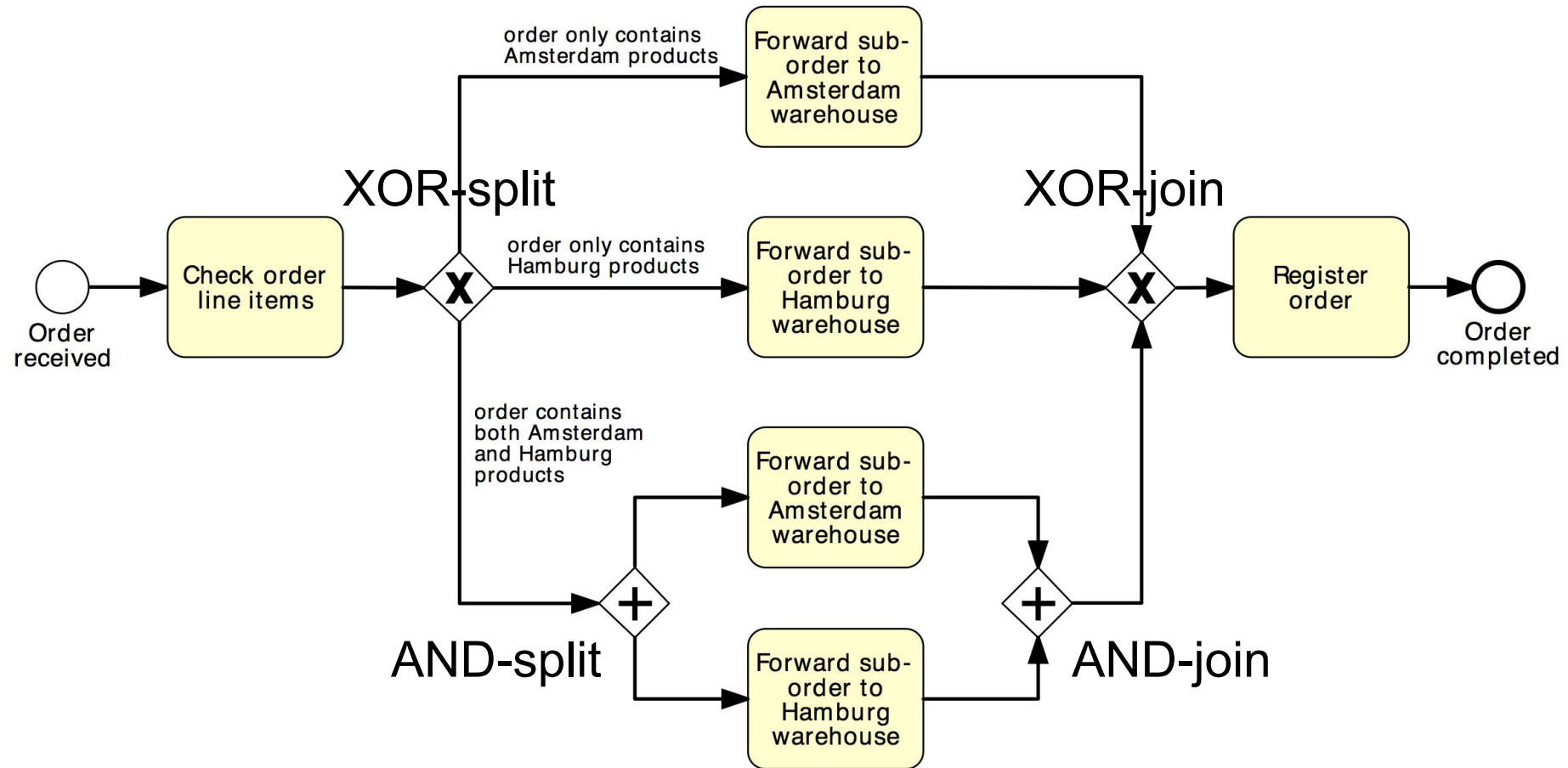
## Between XOR and AND

### **Order distribution process**

A company has two warehouses that store different products: Amsterdam and Hamburg. When an order is received, it is distributed across these warehouses: if some of the relevant products are maintained in Amsterdam, a sub-order is sent there; likewise, if some relevant products are maintained in Hamburg, a sub-order is sent there. Afterwards, the order is registered and the process completes.

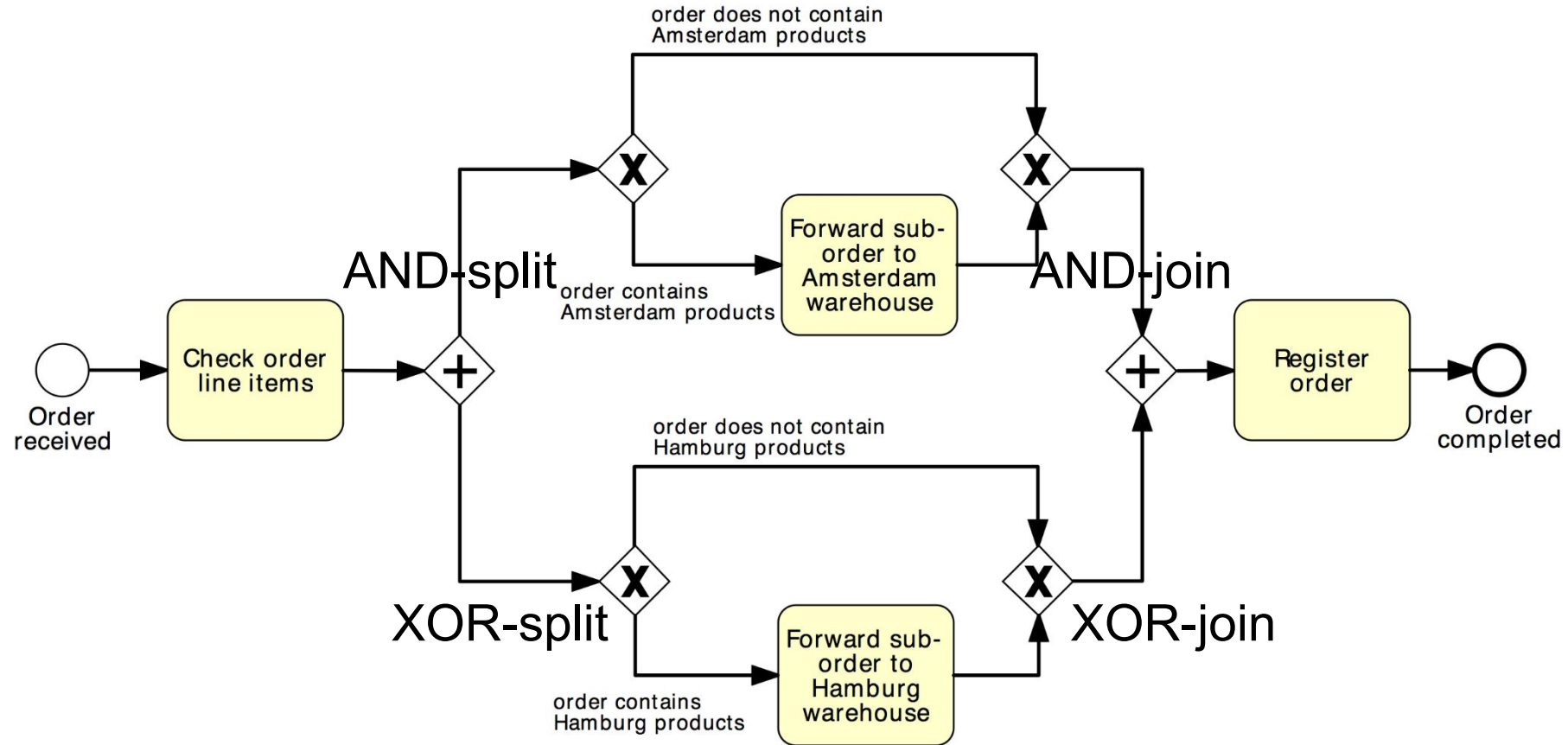
# Solution 1

## Order distribution process

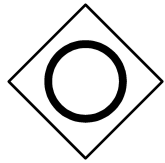


## Solution 2

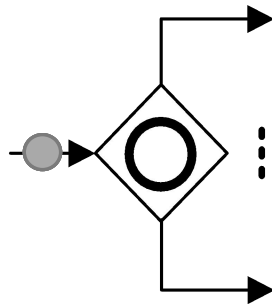
### Order distribution process



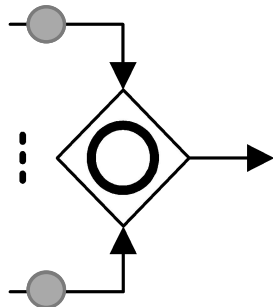
# OR Gateway



An *OR Gateway* provides a mechanism to create and synchronize  $n$  out of  $m$  parallel flows.



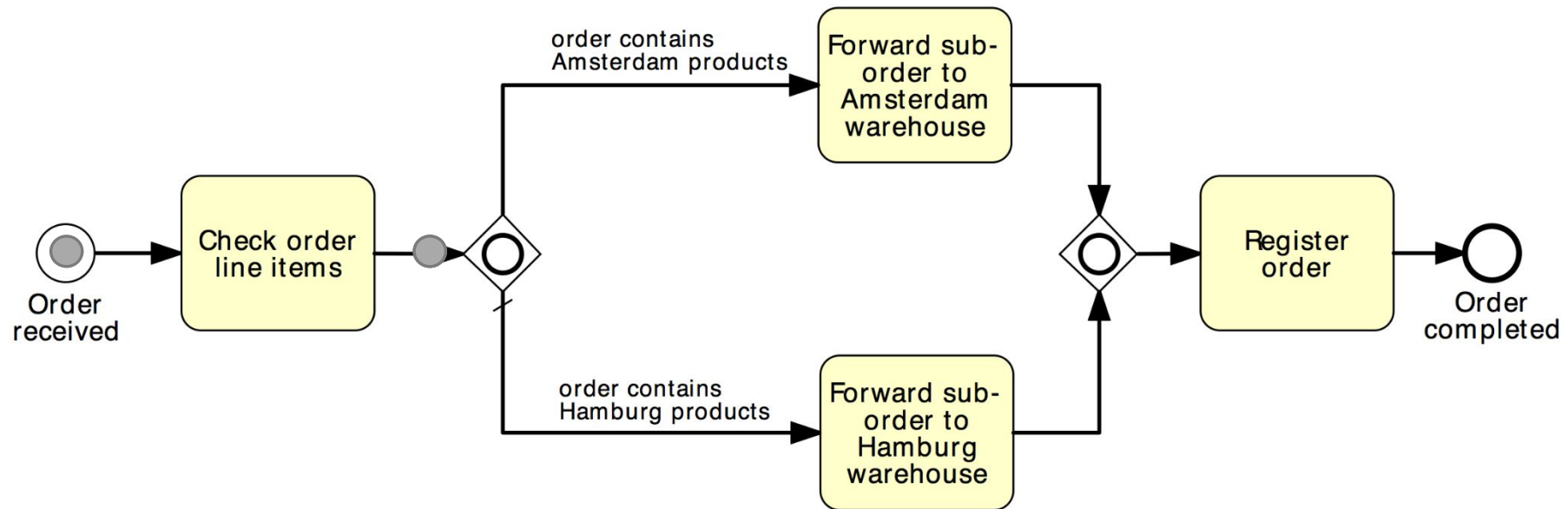
*OR-split* takes one or more branches depending on conditions



*OR-join* proceeds when all **active** incoming branches have completed

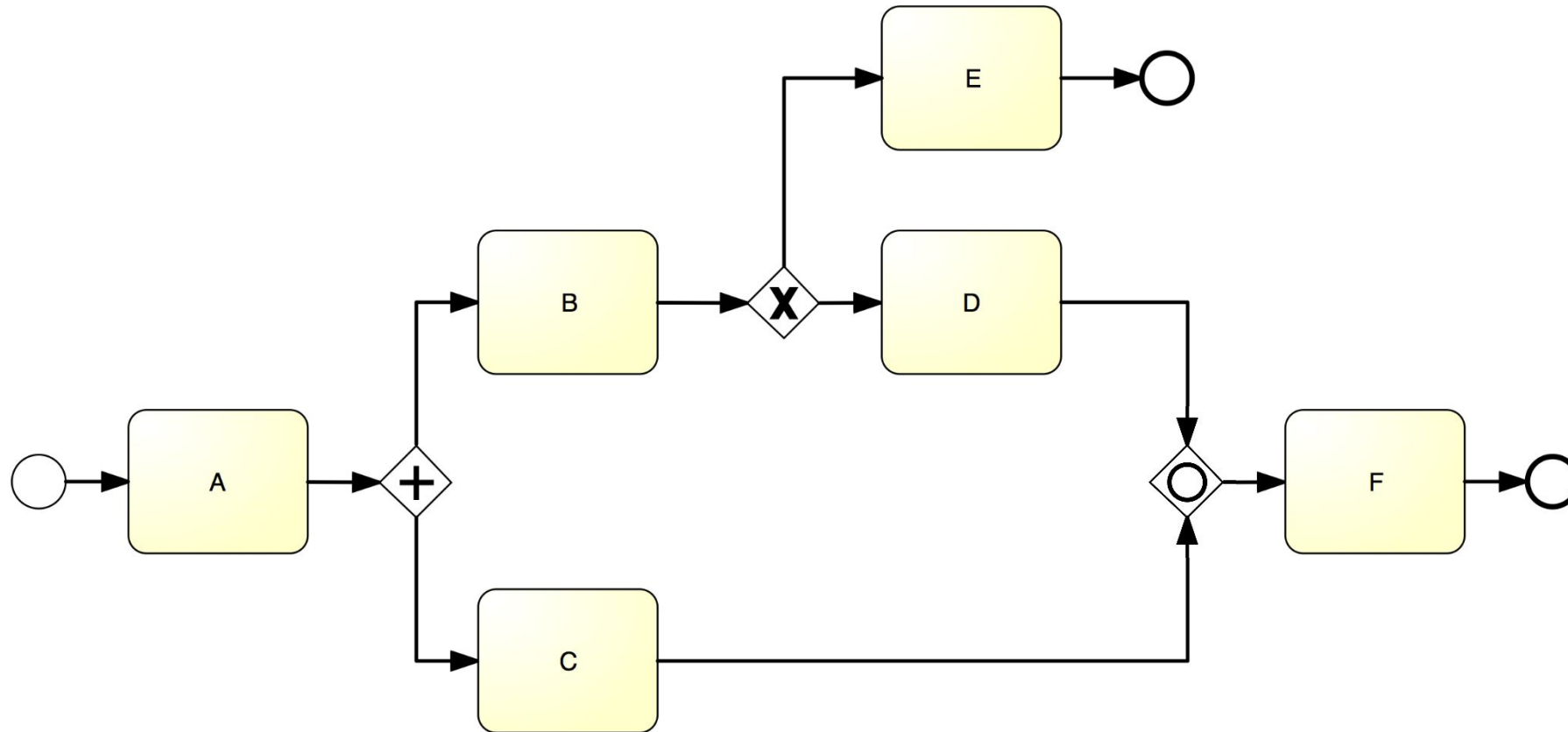
# Solution using OR Gateway

## Order distribution process

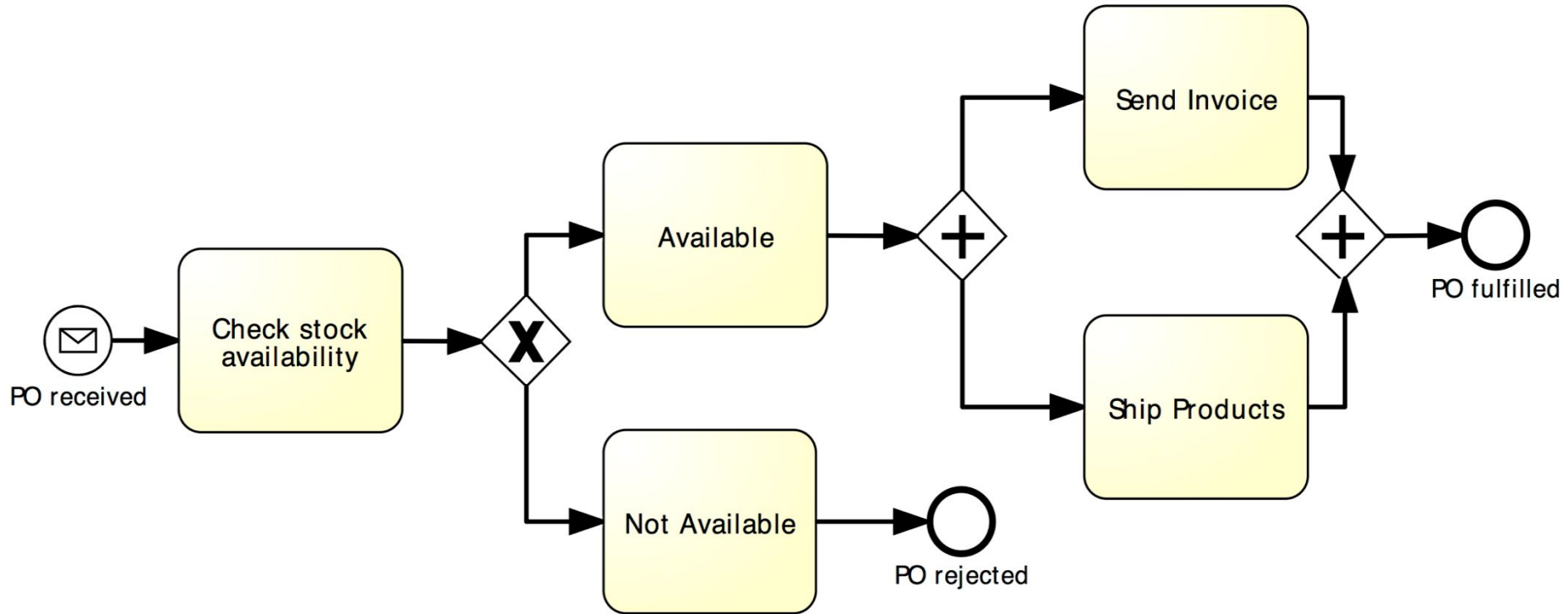




What join type do we need here?



# Beware: Beginner's Mistake...



# Guidelines: Naming Conventions

1. Give a name to every event and task
2. For tasks: verb followed by business object name and possibly complement
  - Issue Driver Licence, Renew Licence via Agency
3. For message events: object + past participle
  - Invoice received, Claim settled
4. Avoid generic verbs such as Handle, Record...
5. Label each XOR-split with a condition
  - Policy is invalid, Claim is inadmissible