

# Methods for the specification and verification of business processes

MPB (6 cfu, 295AA)

Roberto Bruni

<http://www.di.unipi.it/~bruni>

02 - Examples



# Insurance claim example

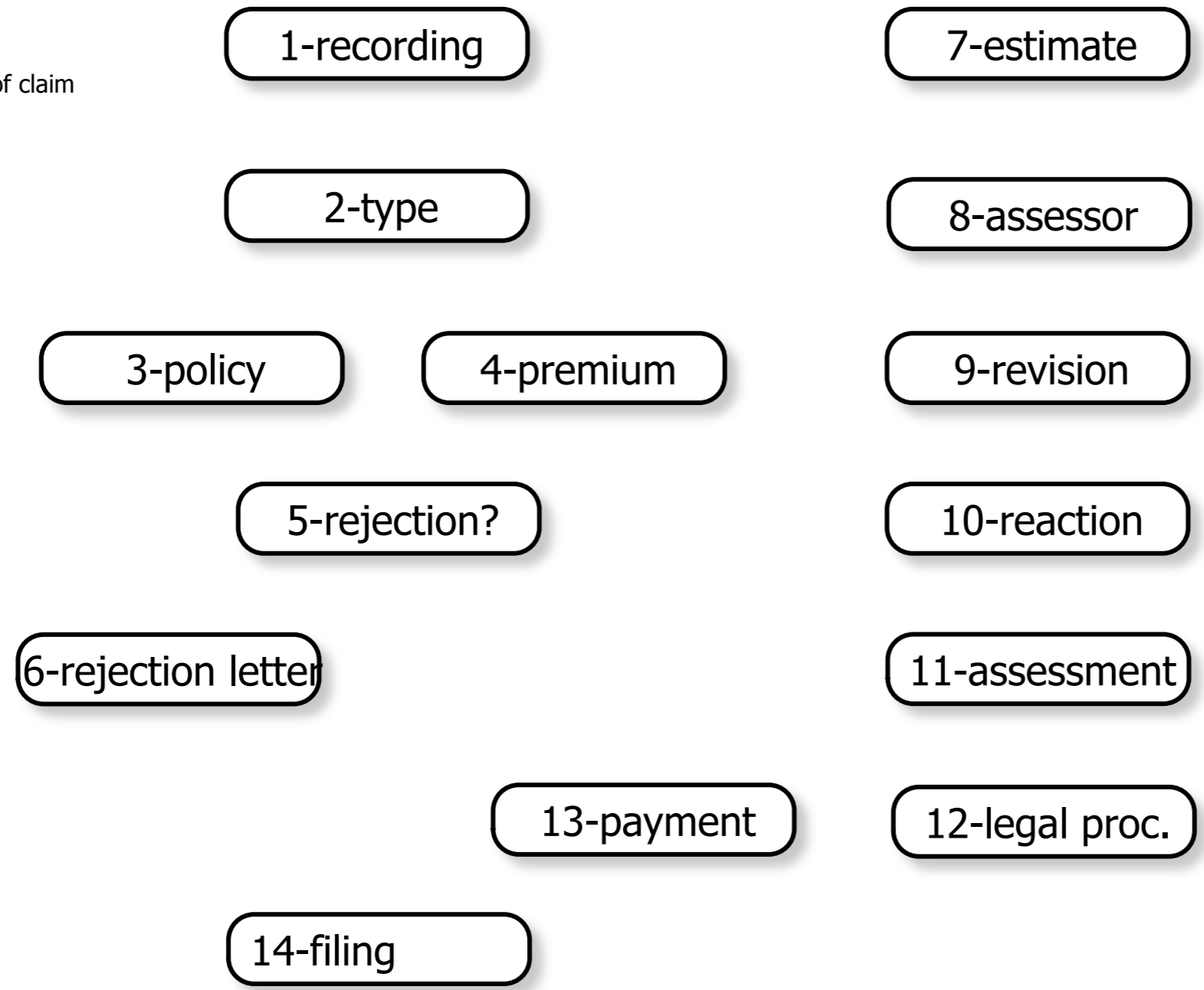
Sect.1.3 of Workflow Management: Models, Methods, and Systems

# An example: insurance claim

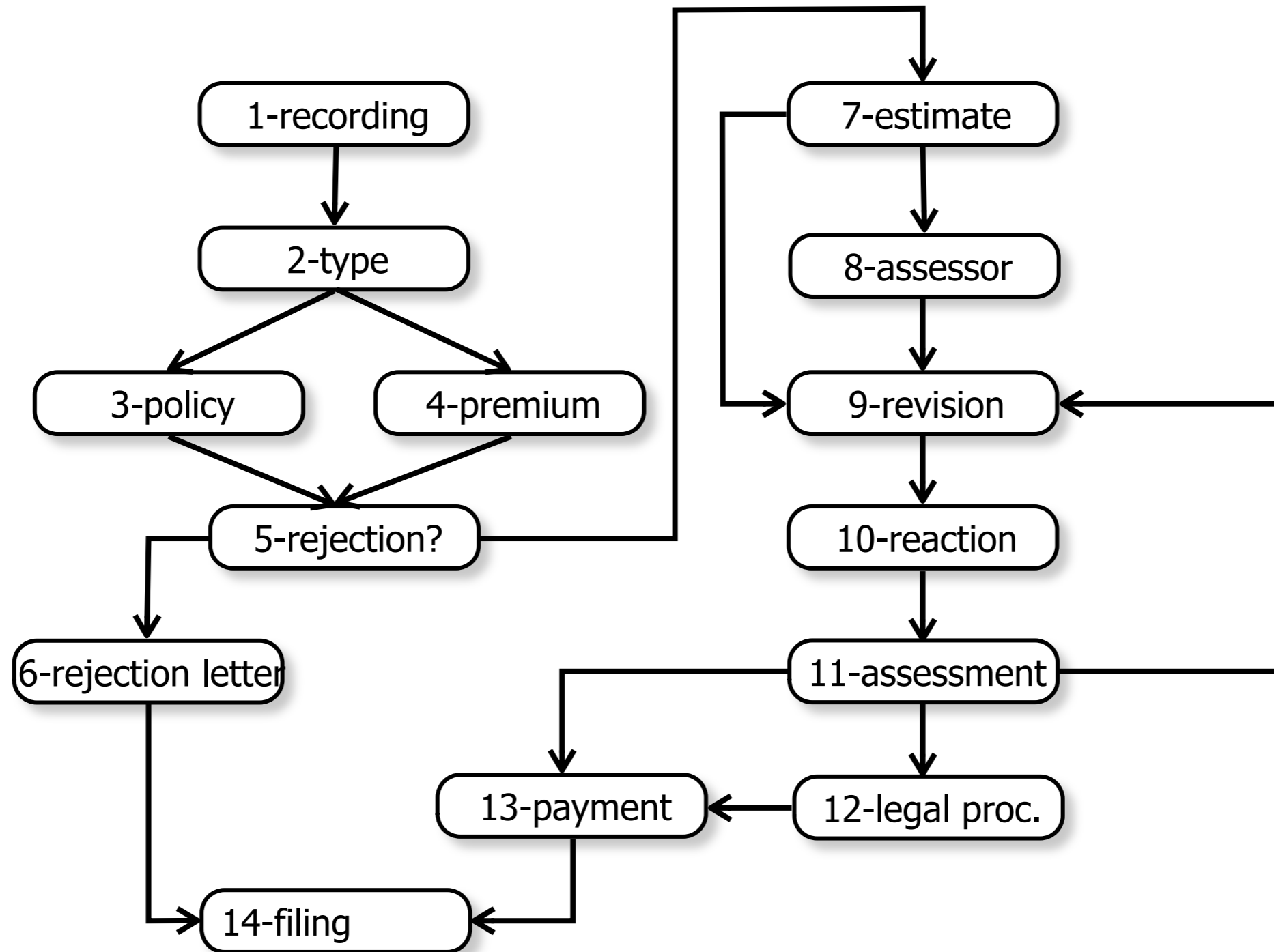
1. **recording** the receipt of the claim
2. establishing the **type** of the claim
3. checking covering of client's **policy**
4. checking the **premium** (payments up to date?)
5. **rejection**, if 3 or 4 has negative result
6. producing a **rejection letter**
7. roughly **estimate** the amount to be paid, if 3 & 4 have positive results
8. appointment of an **assessor**, if needed
9. **revision** of the amount offered to the client
10. recording client's **reaction**
11. **assessment** of objection: decision to revise 9 or take legal action 12
12. **legal proceedings**
13. **payment** of claim
14. **filing** and closure of claim

1. recording the receipt of the claim
2. establishing the **type** of the claim
3. checking covering of client's **policy**
4. checking the **premium** (payments up to date?)
5. **rejection**, if 3 or 4 has negative result
6. producing a **rejection letter**
7. roughly **estimate** the amount to be paid, if 3 & 4 have positive results
8. appointment of an **assessor**, if needed
9. **revision** of the amount offered to the client
10. recording client's **reaction**
11. **assessment** of objection: decision to revise 9 or take legal action
12. **legal proceedings**
13. **payment** of claim
14. **filing** and closure of claim

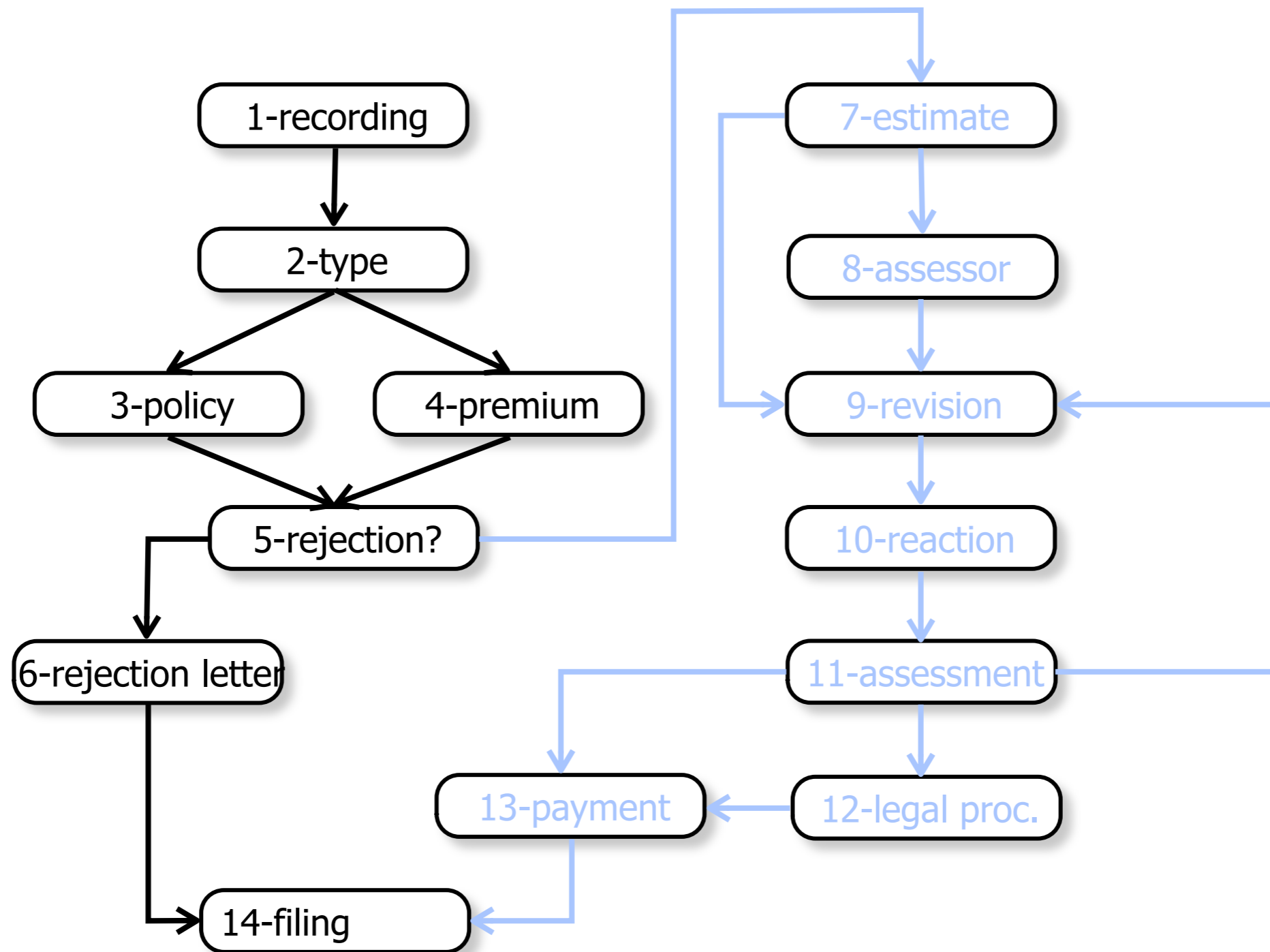
# Tasks



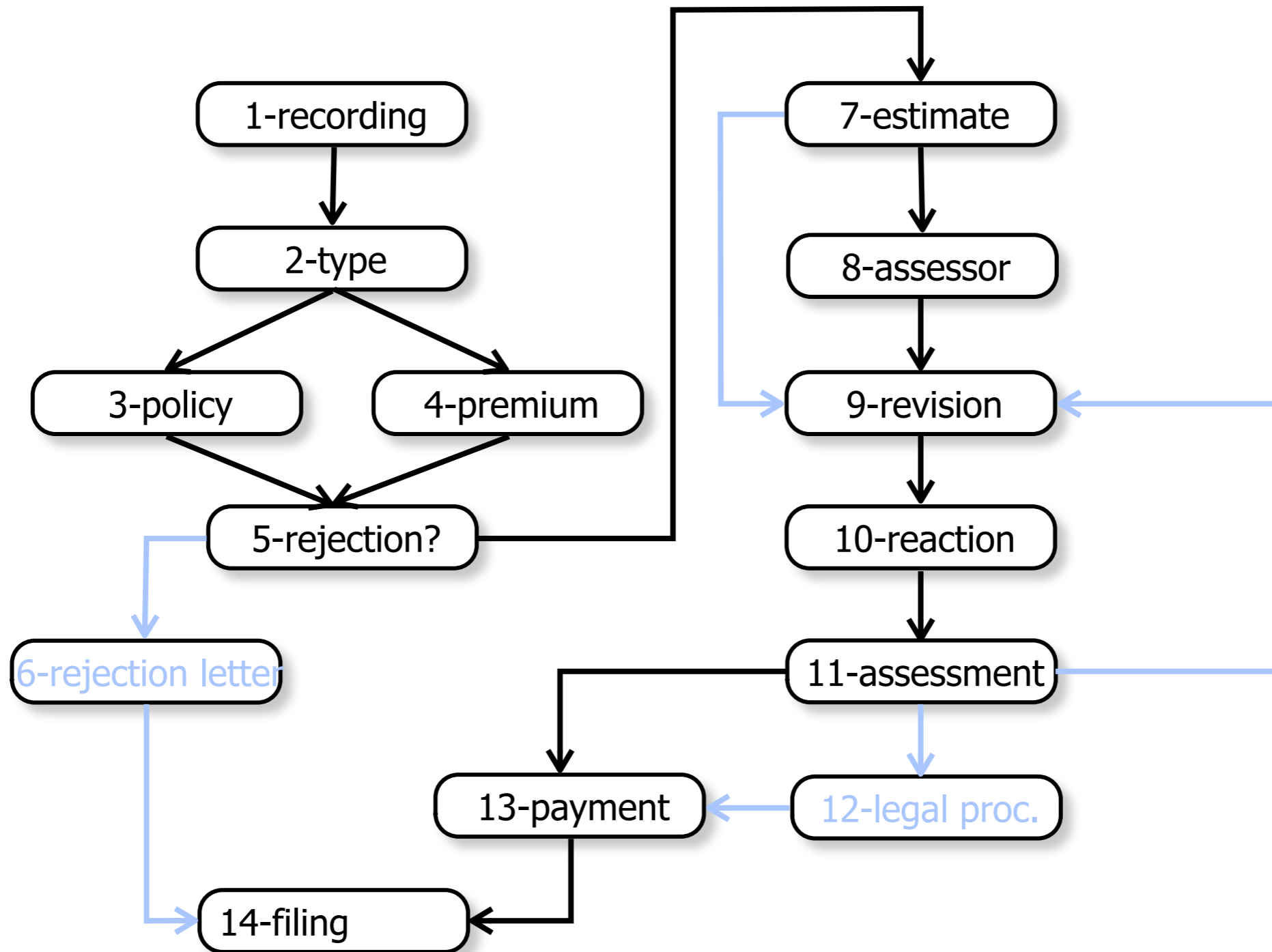
# Order / links



# A process instance



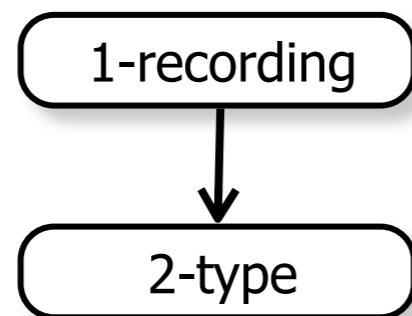
# Another instance



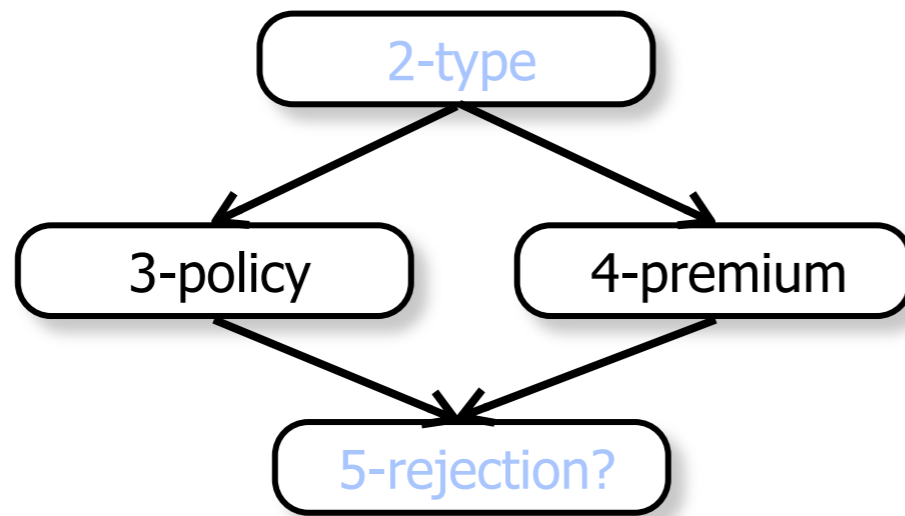
# Some link patterns



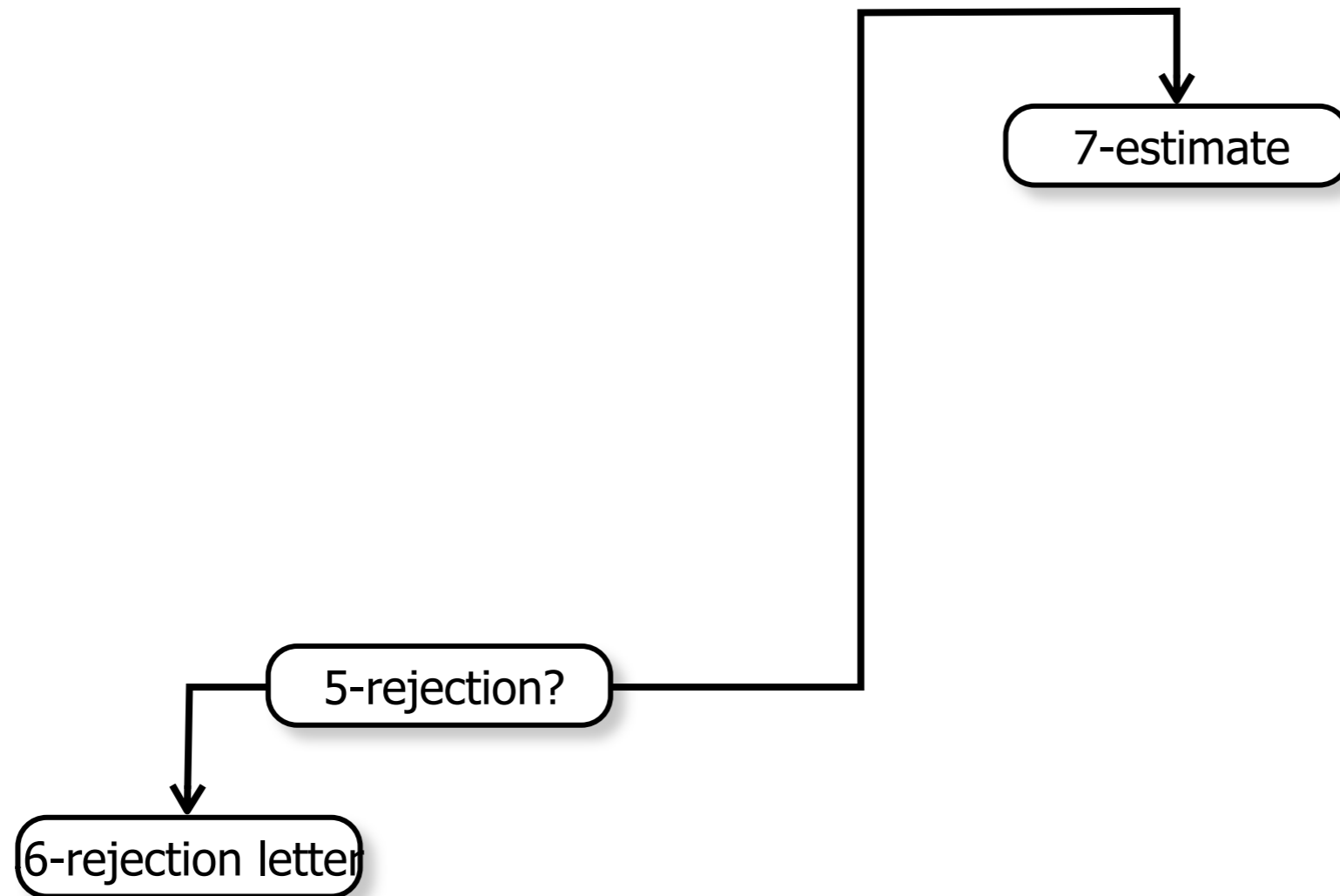
# Sequence



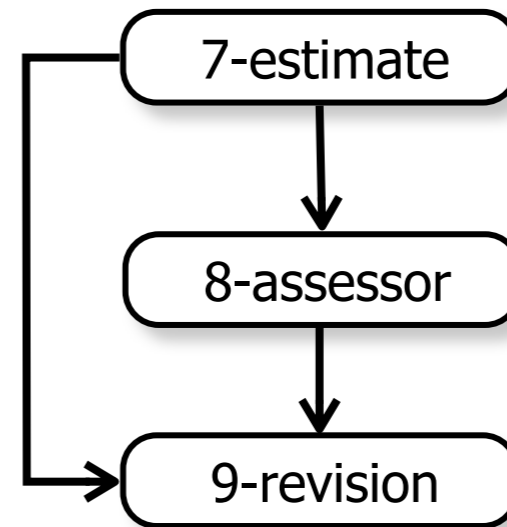
# Parallel



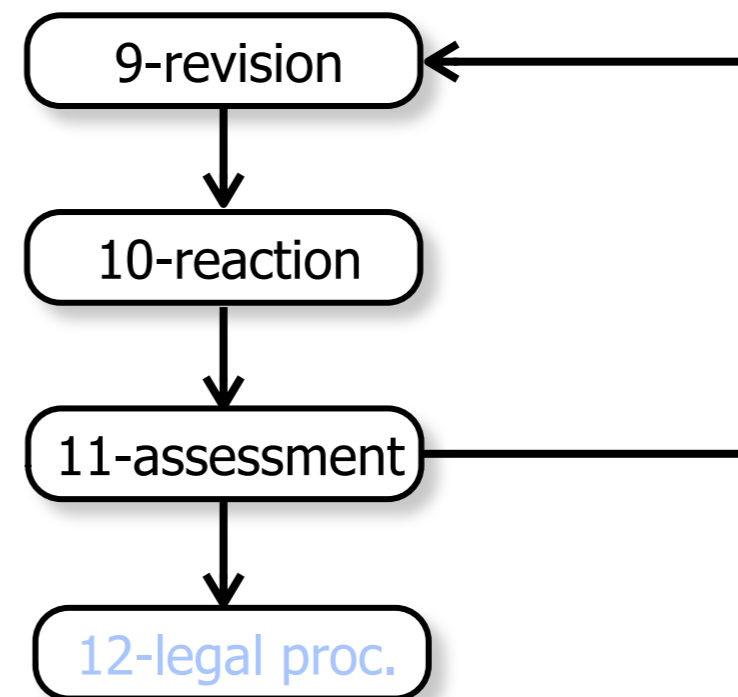
# Selection



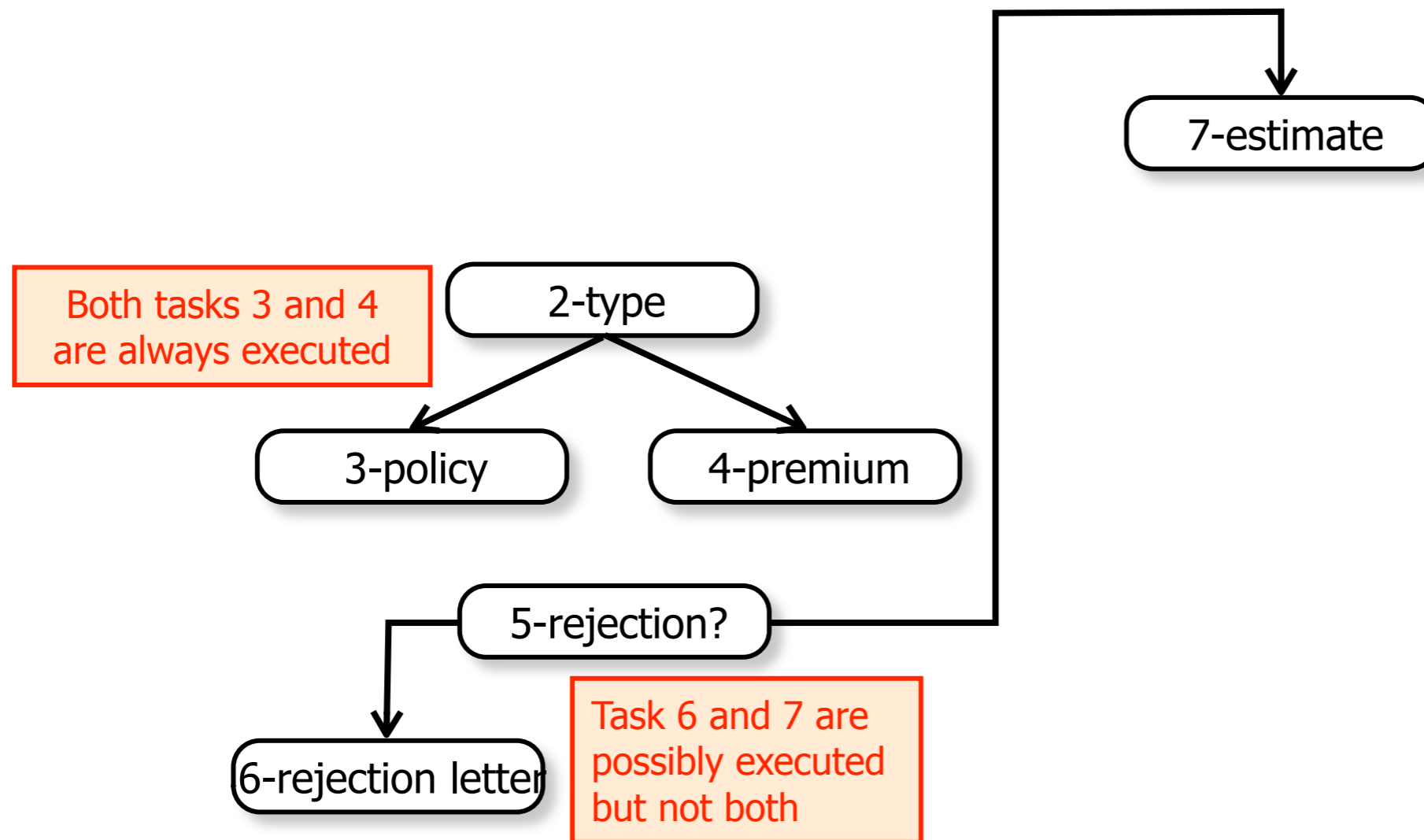
# Another selection



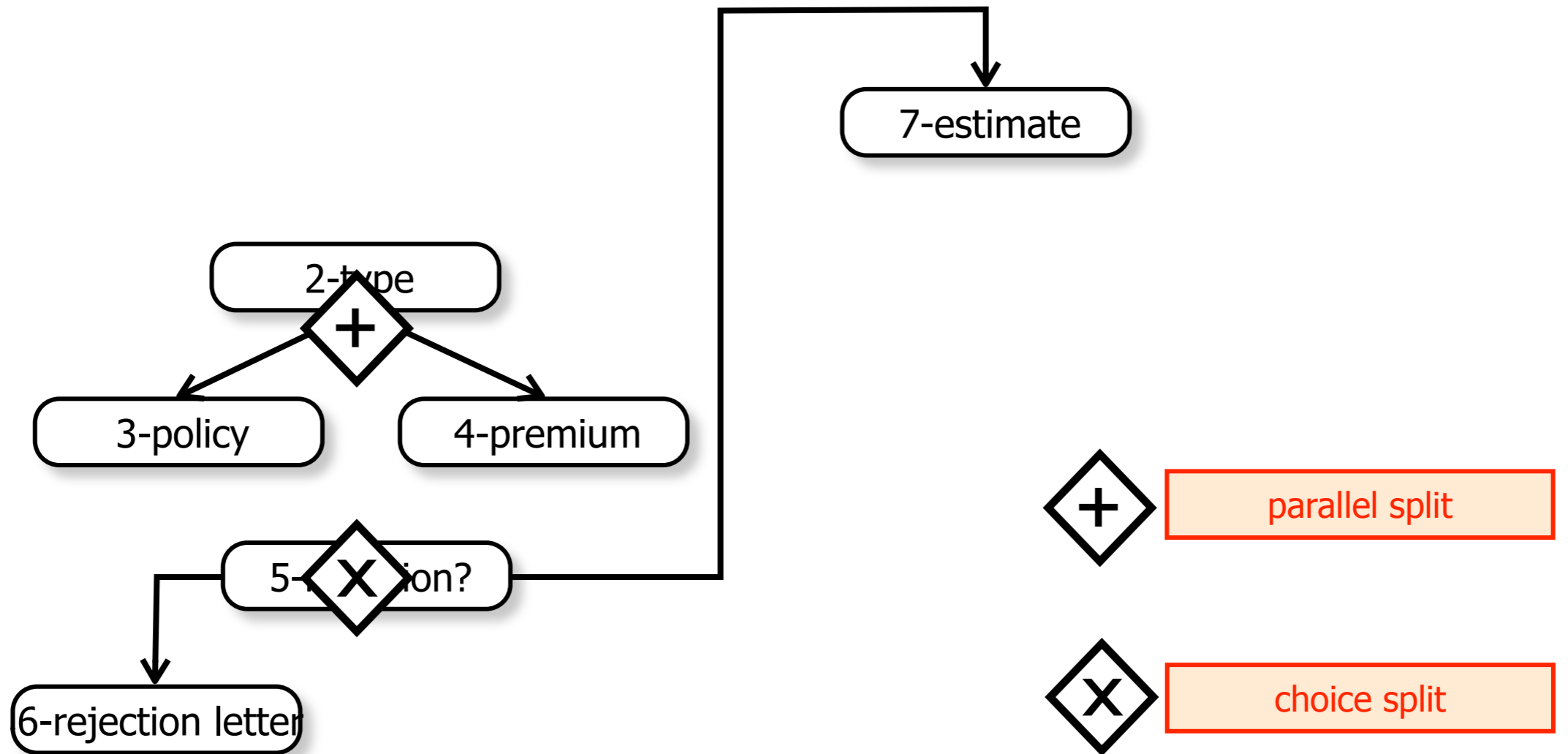
# Iteration



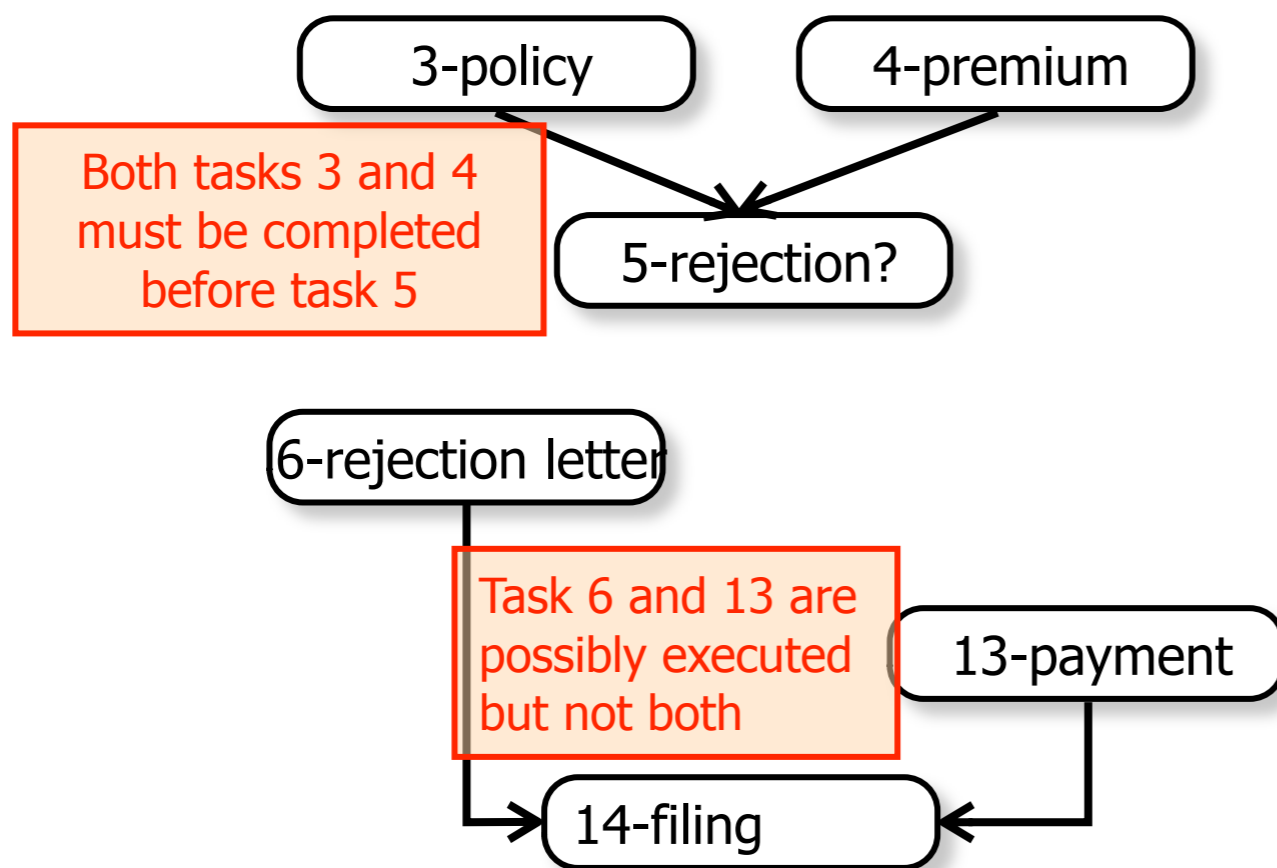
# Ambiguity!



# Disambiguation

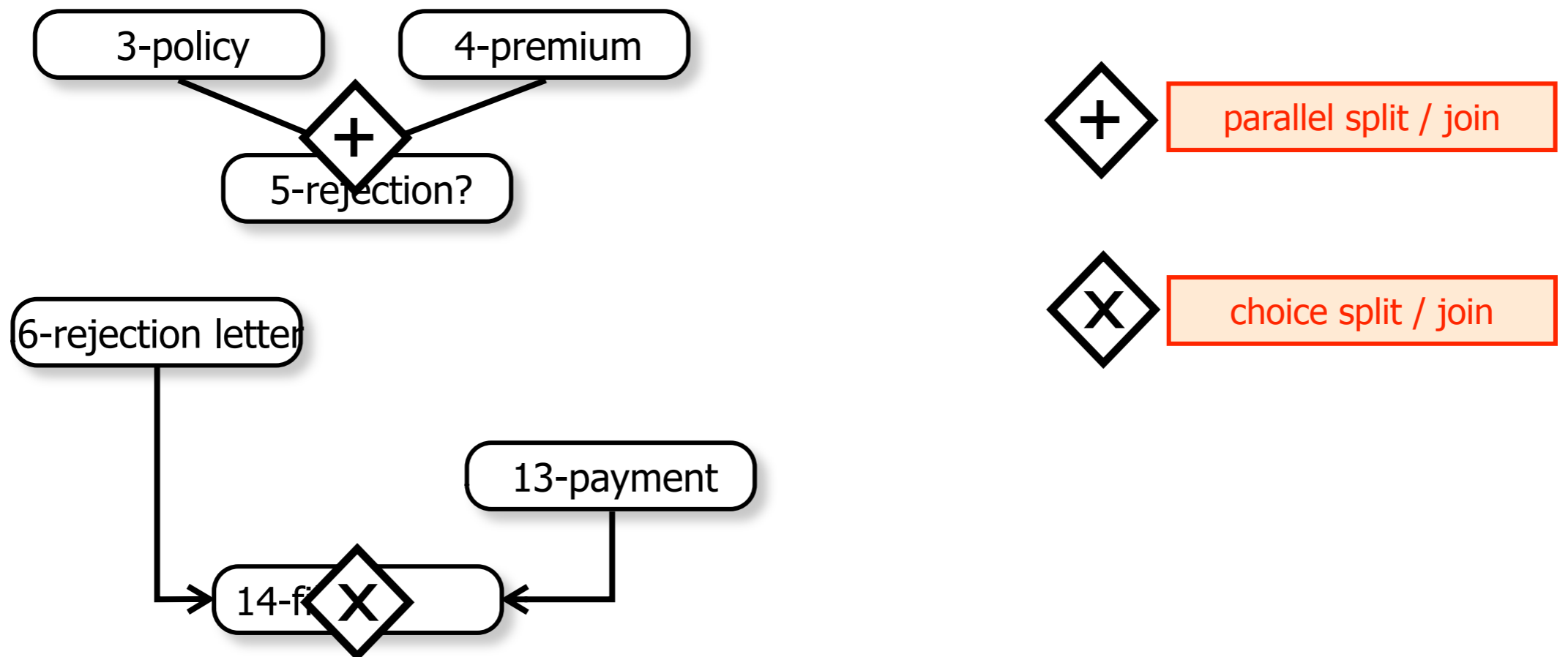


# Ambiguity!





# Disambiguation



# Orchestration

Business process models are performed in a single organization by definition

Thus, the **ordering of activities** can be controlled by a **business process management system** as a **centralized** software component run by the organization

This kind of control is called **orchestration**

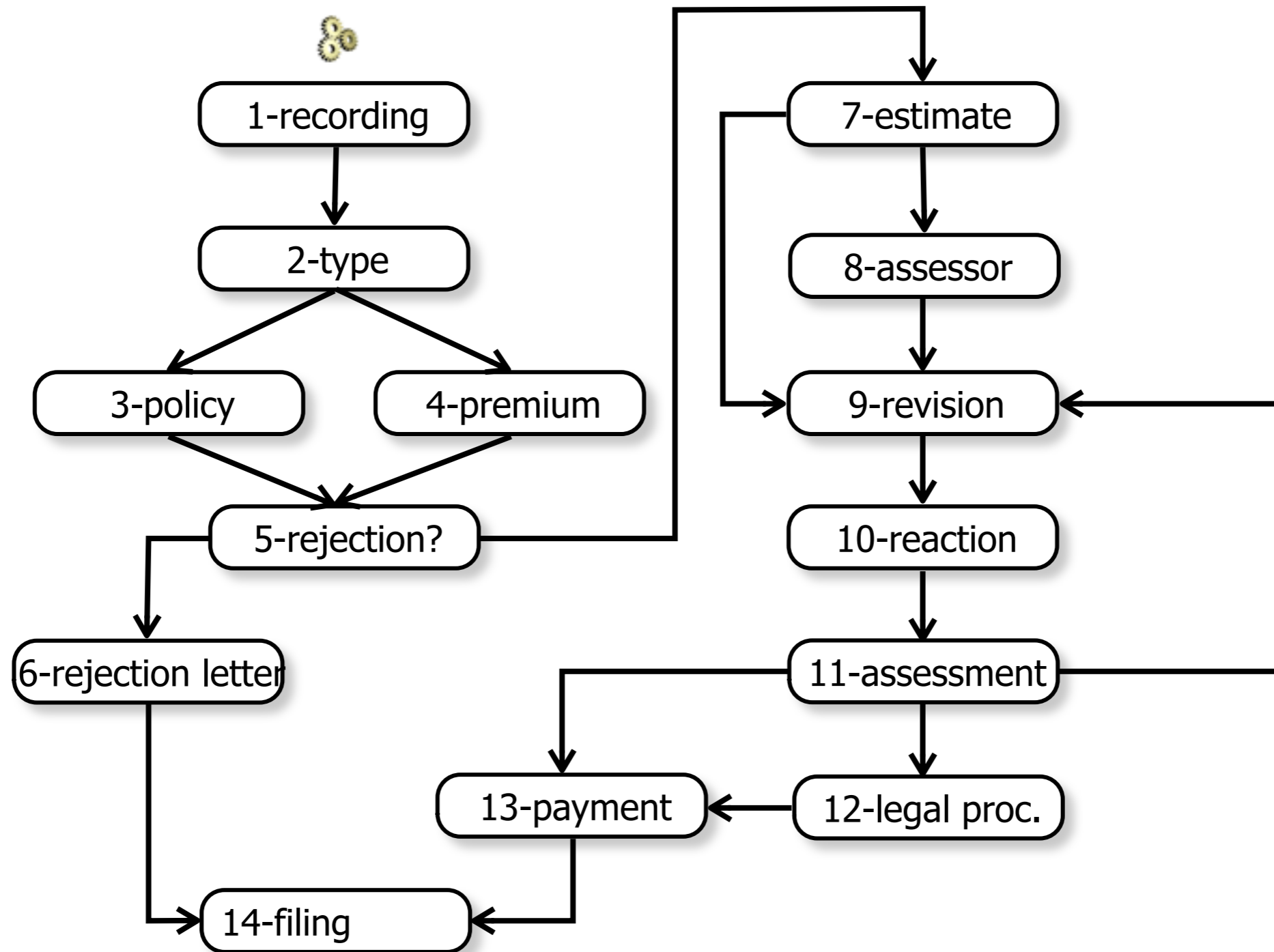
# Orchestration

Orchestration is about describing and executing a single view point model

The analogy is with the conductor who centrally controls the musicians in an orchestra



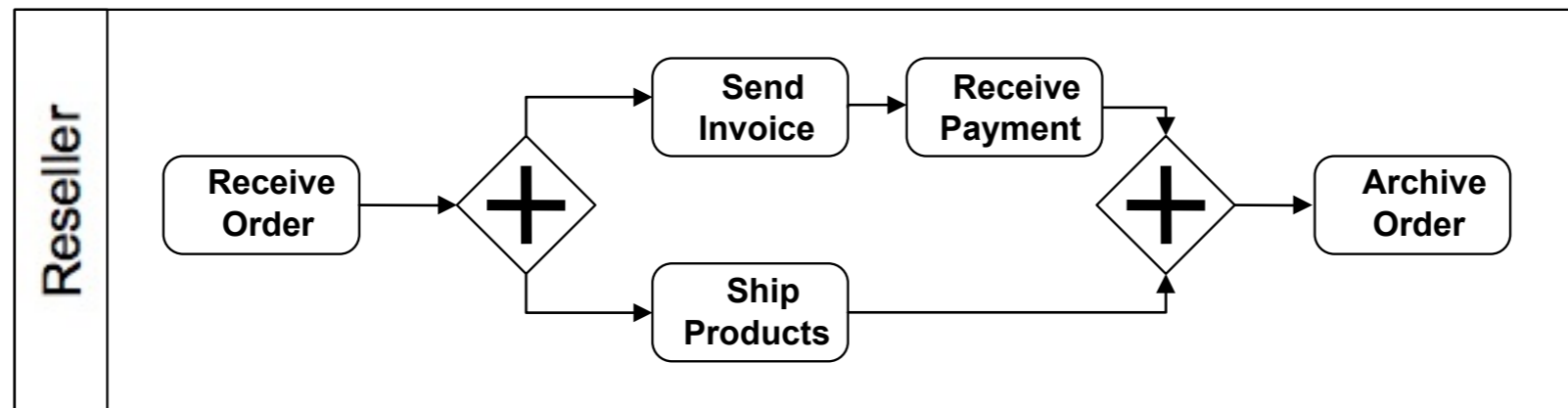
# Executing the model



# Buyer & Reseller example

Sect.1.1 of Business Process Management: Concepts, Languages, Architectures

# Example: Reseller



R<sub>1</sub>

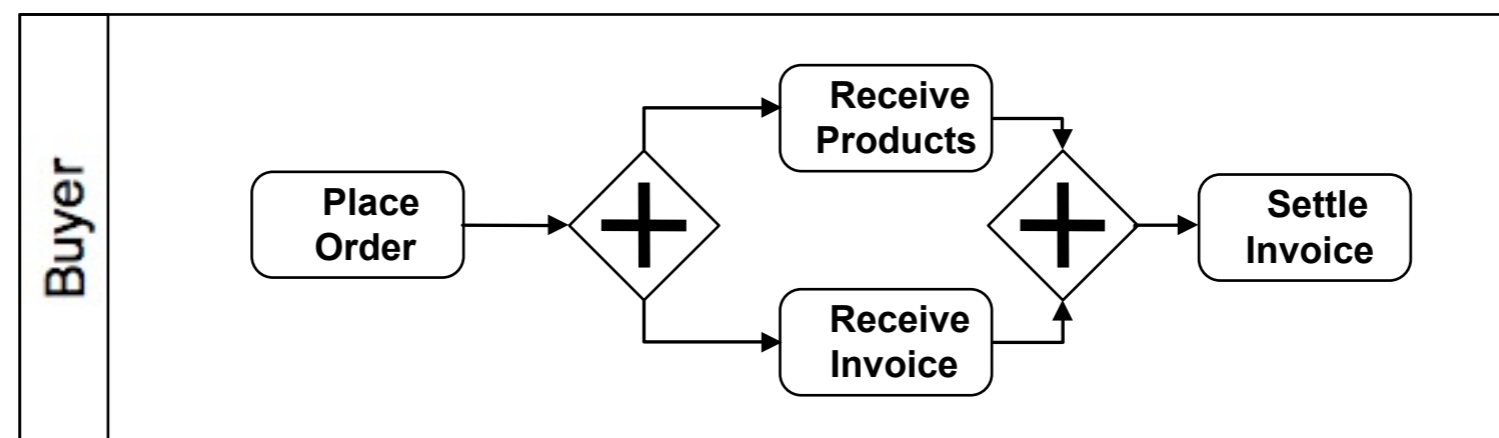
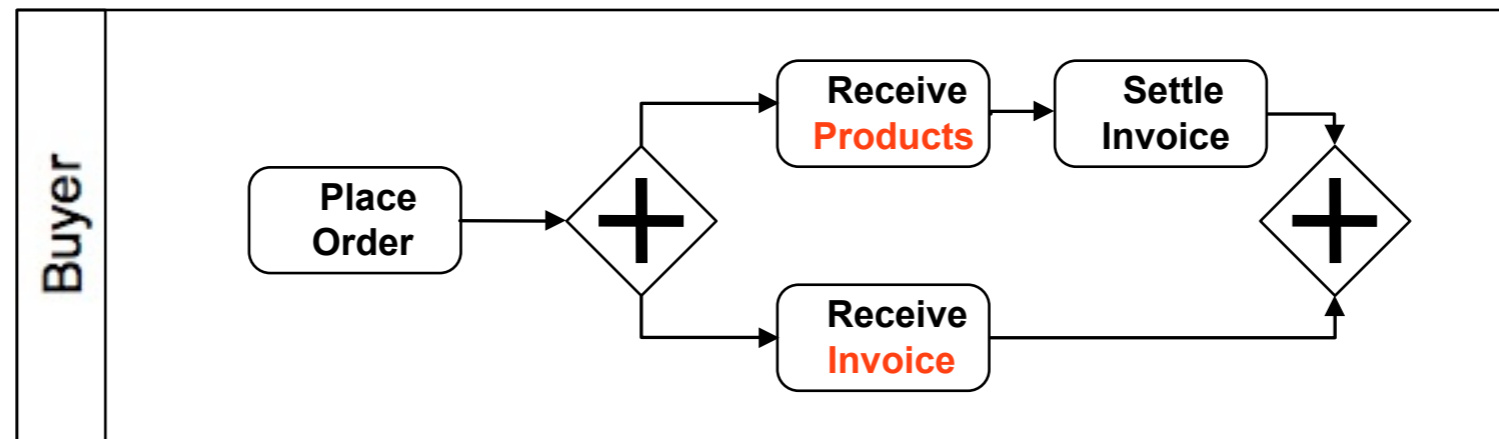
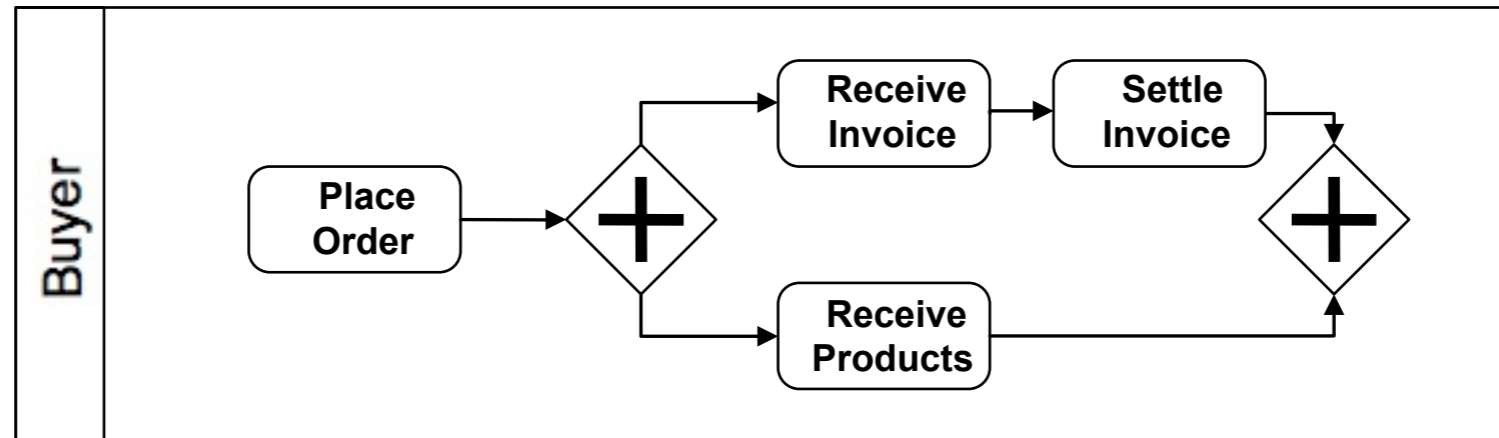
We move to  
BPMN-like syntax

A reseller can use the business process model above to configure the business process management system accordingly

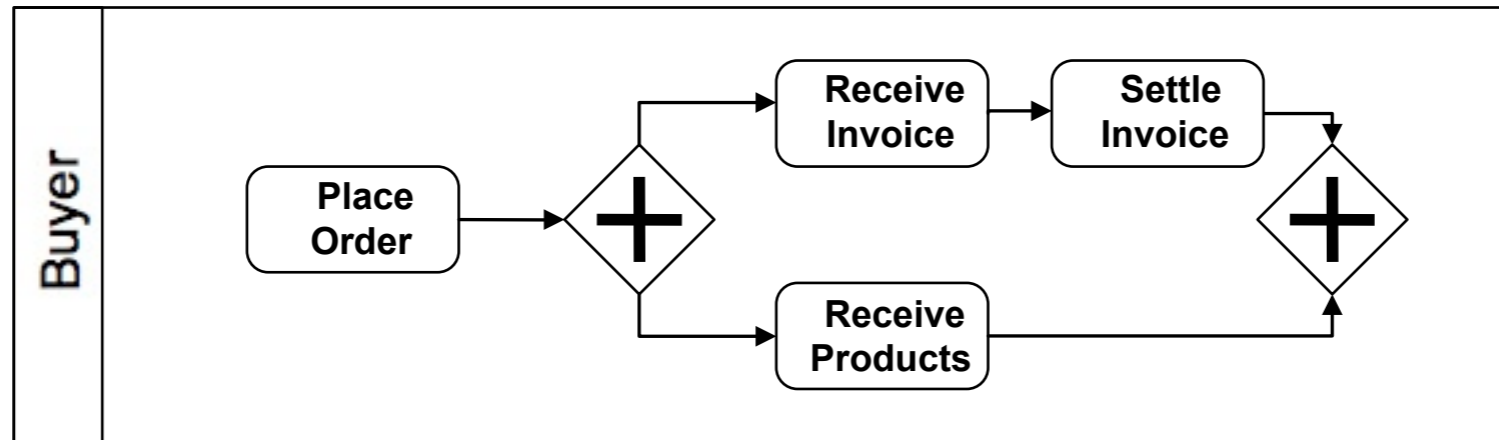
All instances will be executed as specified (after receiving the order, send and ship activities are concurrently executed)

# Example: Buyers

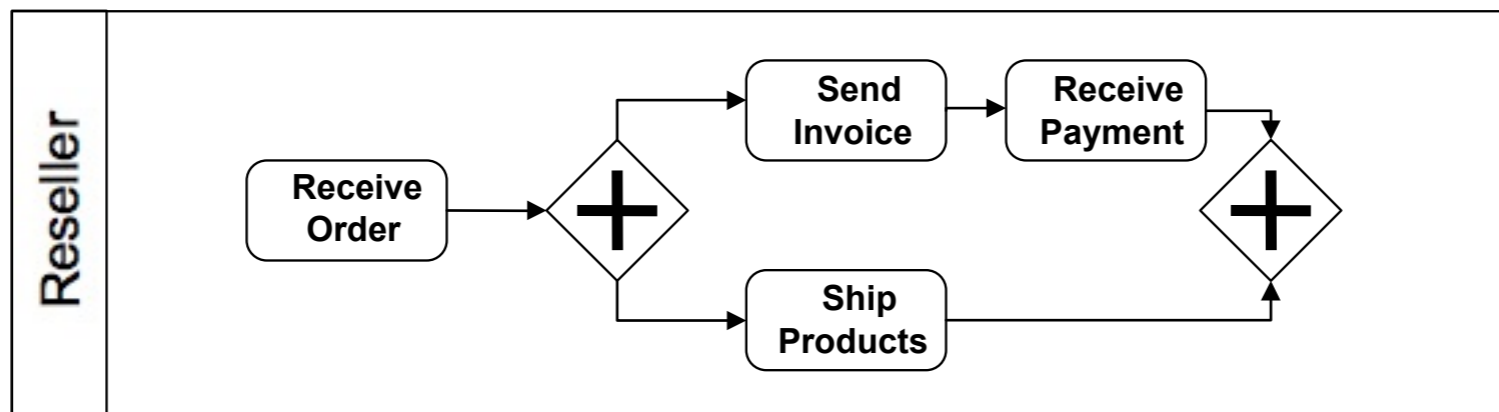
Do they all make sense?



# Buyer & Reseller



Separately developed processes want to communicate!





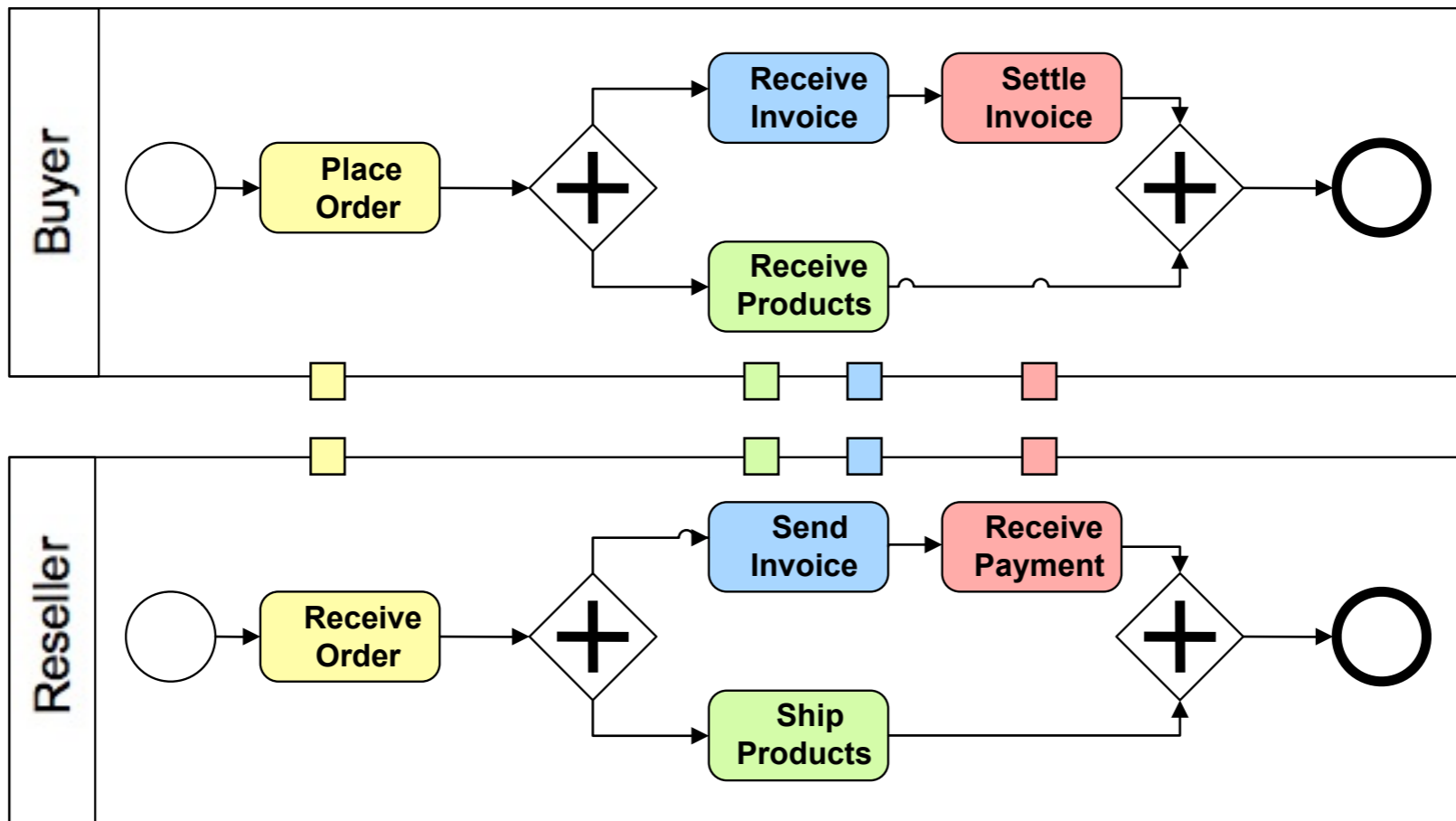
# Cross-organization interaction

Each business process is enacted by one  
organization

Business processes can interact with each other

Interacting activities of business processes must  
be related together

# Interacting processes

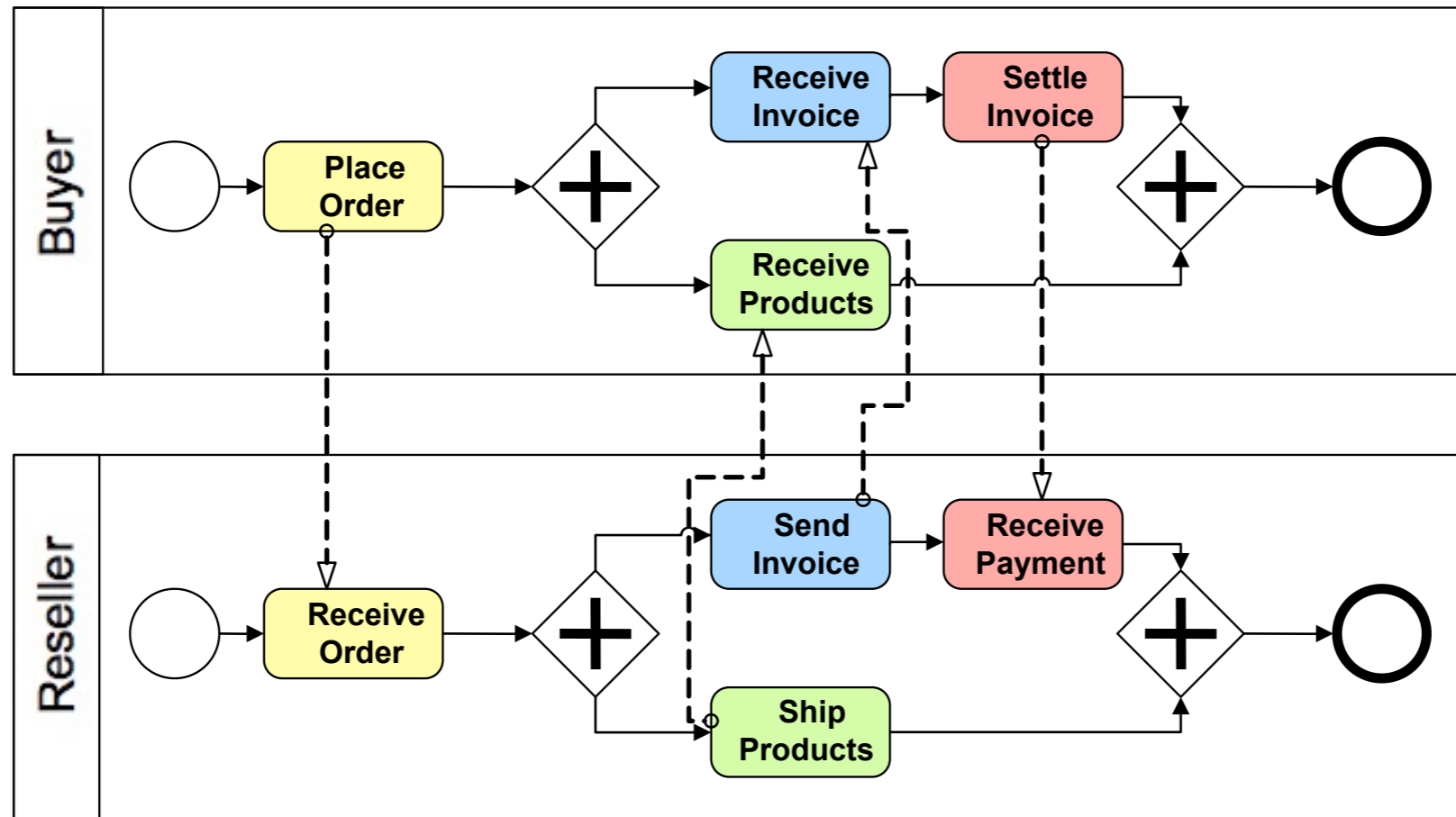


We move to BPMN-like syntax

Interacting processes can exchange information (electronic messages, physically transported objects)

# Interacting processes

We move to BPMN-like syntax



Message flow is represented by dotted arcs

# Choreography

The interactions of a set of business processes are specified in a **process choreography**

## **Difference w.r.t. orchestration:**

the absence of a central agent that controls the activities in the business processes involved

For the interaction to be realized correctly, the interacting business processes better **be aware and agree upon the choreography in advance**

# Choreography

Choreography is about describing a global model  
(multi-point view)

The analogy is with the dancers who behave  
autonomously, but follow their parts in the choreography



# Choreography diagram

Choreography diagrams allow for multiple concrete implementations, with different software support

**Old-fashioned order:** a buyer browses a paper catalogue of a reseller, then fills a postcard and sends it by snail mail and pay by bank transfer

**e-commerce:** a buyer browses an online web catalogue, fills a virtual basket and an electronic form (billing information) and presses the submit button. The goods themselves may be intangible (e-books, music, video, software)

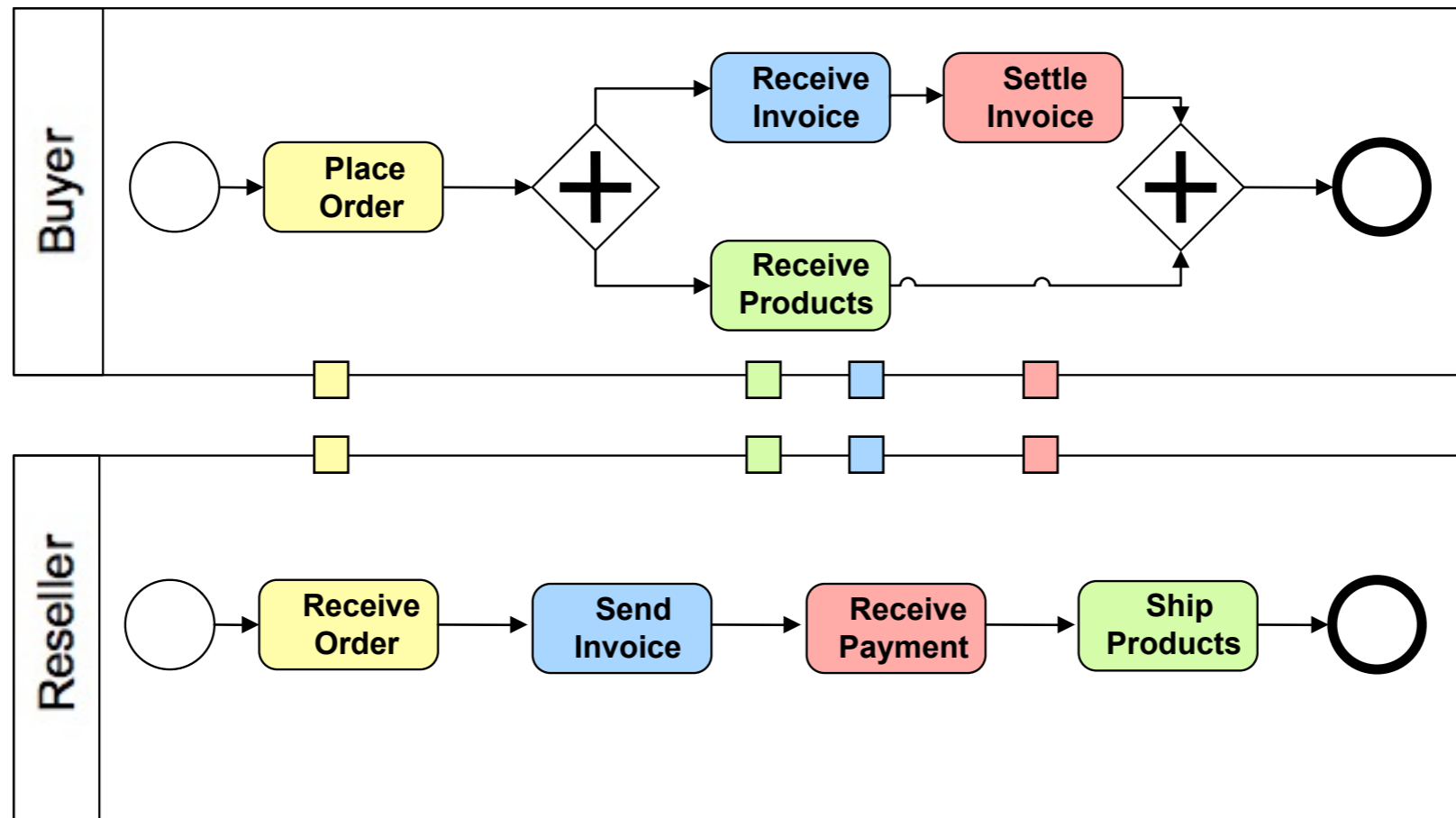
# Interaction issues

As said, interacting business processes must be aware and agree upon the choreography

In such cases, **the realization of business processes by participants can change without affecting the overall behaviour**

On the other hand, **if the change is not done correctly, then some problems may arise**

# Question time

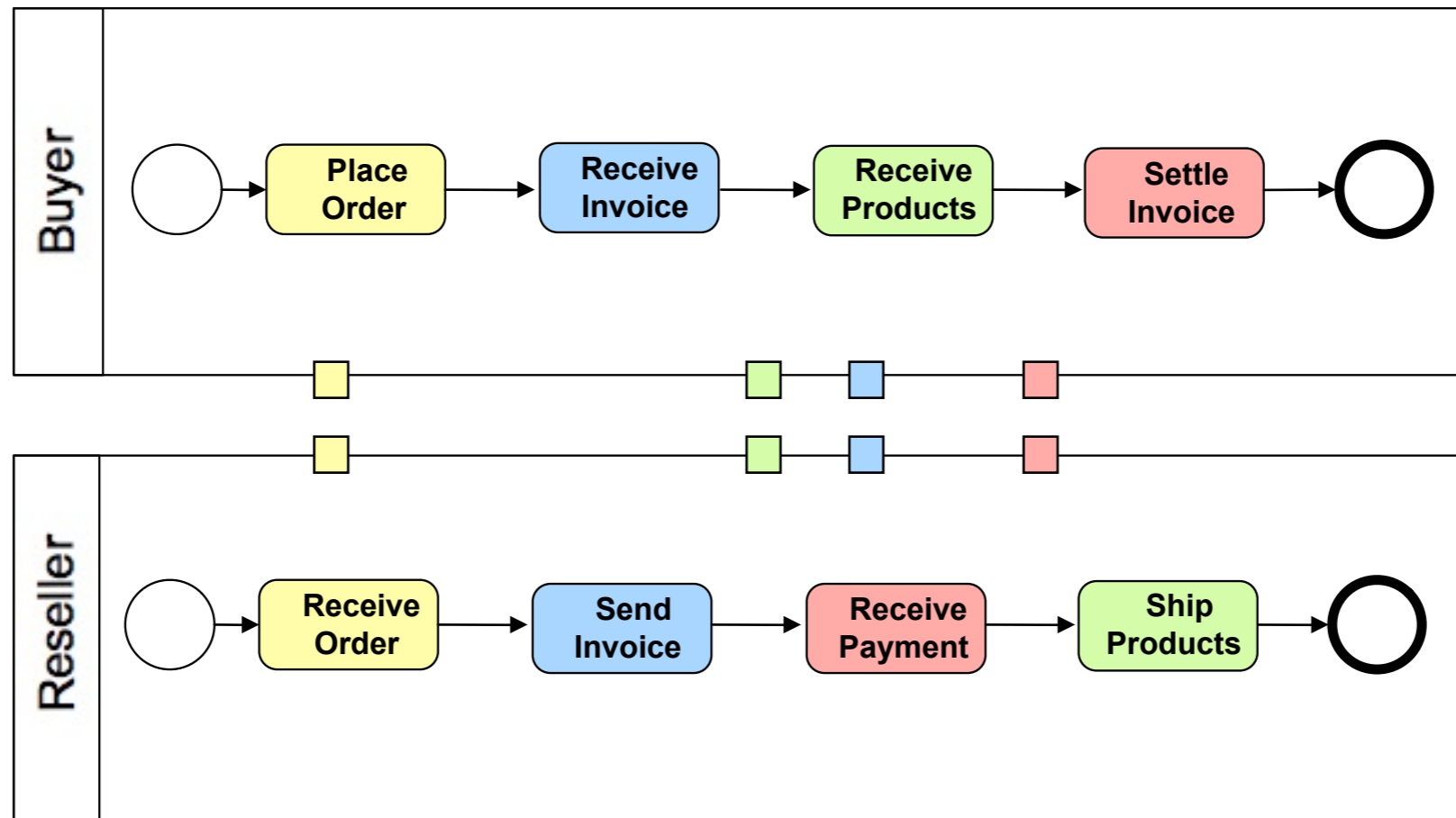


R<sub>2</sub>

Still working fine?



# Question time



B<sub>4</sub>

Still working fine?

# Exercises

In previous slides, we have seen many variants of business processes for resellers (two) and buyers (four).

Build a “compatibility” matrix with two rows and four columns and mark all the combinations for which some problems may arise during the interaction because activities are not implemented in the expected order.

You are also free to consider other process diagrams, by adding the corresponding rows and columns in the matrix.

	B <sub>1</sub>	B <sub>2</sub>	B <sub>3</sub>	B <sub>4</sub>
R <sub>1</sub>	ok			
R <sub>2</sub>	ok			no

# Exercises

**Travel agency:** define a series of task for booking a flight, a hotel and optionally a car, with the possibility to change dates, to cancel the booking, to confirm the booking. Then, draw a process diagram relating the tasks.

**Coffee break:** draw the process diagram for a vending machine that accepts a coin, then gives the possibility (1) to get a coffee or (2) to insert another coin and get either a cappuccino or a tea. Draw the process diagrams for a compatible and a "problematic" butler robot.

