

# **BPMN 2.0**

## **Introduction to the Level 1 Palette**

by Declan Chellar

# Scope of this slide deck

- In: an introduction to the shapes of the Level 1 Palette of BPMN 2.0
- In: textual explanations of the purpose of each shape
- In: stand-alone visual examples of the use of each shape
- Out: an example start-to-finish process model

# What is BPMN?

- “Business Process Model and Notation”
- A graphical technique for modelling business processes
- Based on flowcharting
- An expressive language with vocabulary, grammar and syntax
- Managed by the Object Management Group

# Why use BPMN?





- It is an international standard, the *lingua franca* of process modelling
- Easily understood by non-technical business people
- Caters well for event-based process modelling
- Caters well for exception scenarios
- Caters well for hierarchical modelling

# Why use the Level 1 Palette?

Any of the following is a good reason to stick to Level 1:

- You are not familiar with the shapes of the Level 2 Palette
- You are modelling an initial overview of a process
- You want a simple (non-executable) model of a process
- You don't need to model event-triggered behaviour

# First some symbols

- Instructions appear with this symbol: 
- Handy tips appear with this symbol: 
- Warnings appear with this symbol: 
- Quotes from the BPMN 2.0 Specification appear with this symbol:   
*(along with a reference to the relevant section in the specification)*

# Pools and Lanes



This is a **Pool**. It represents an entire Process from start to finish.



<p>[Process Name]</p>	
-----------------------	--

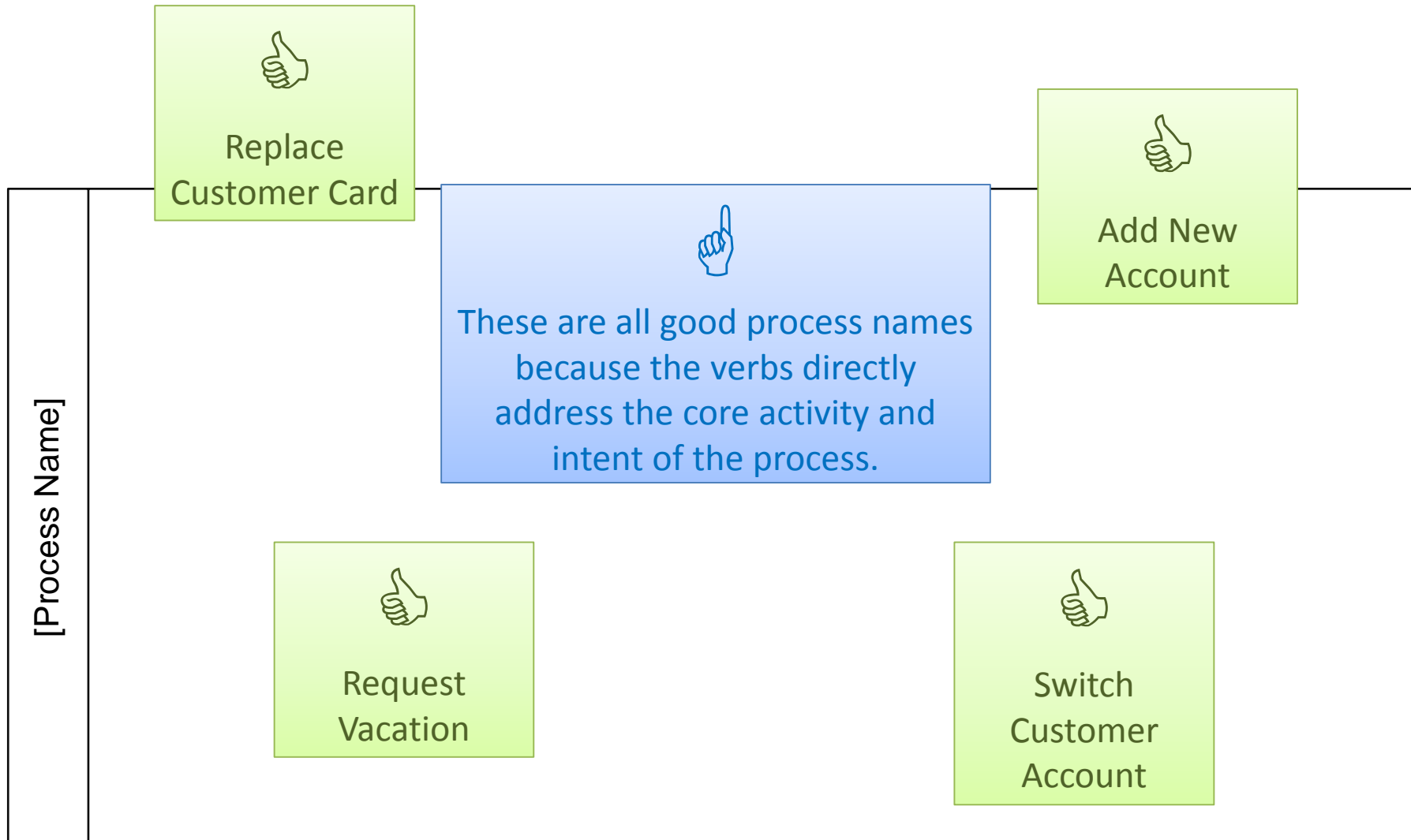


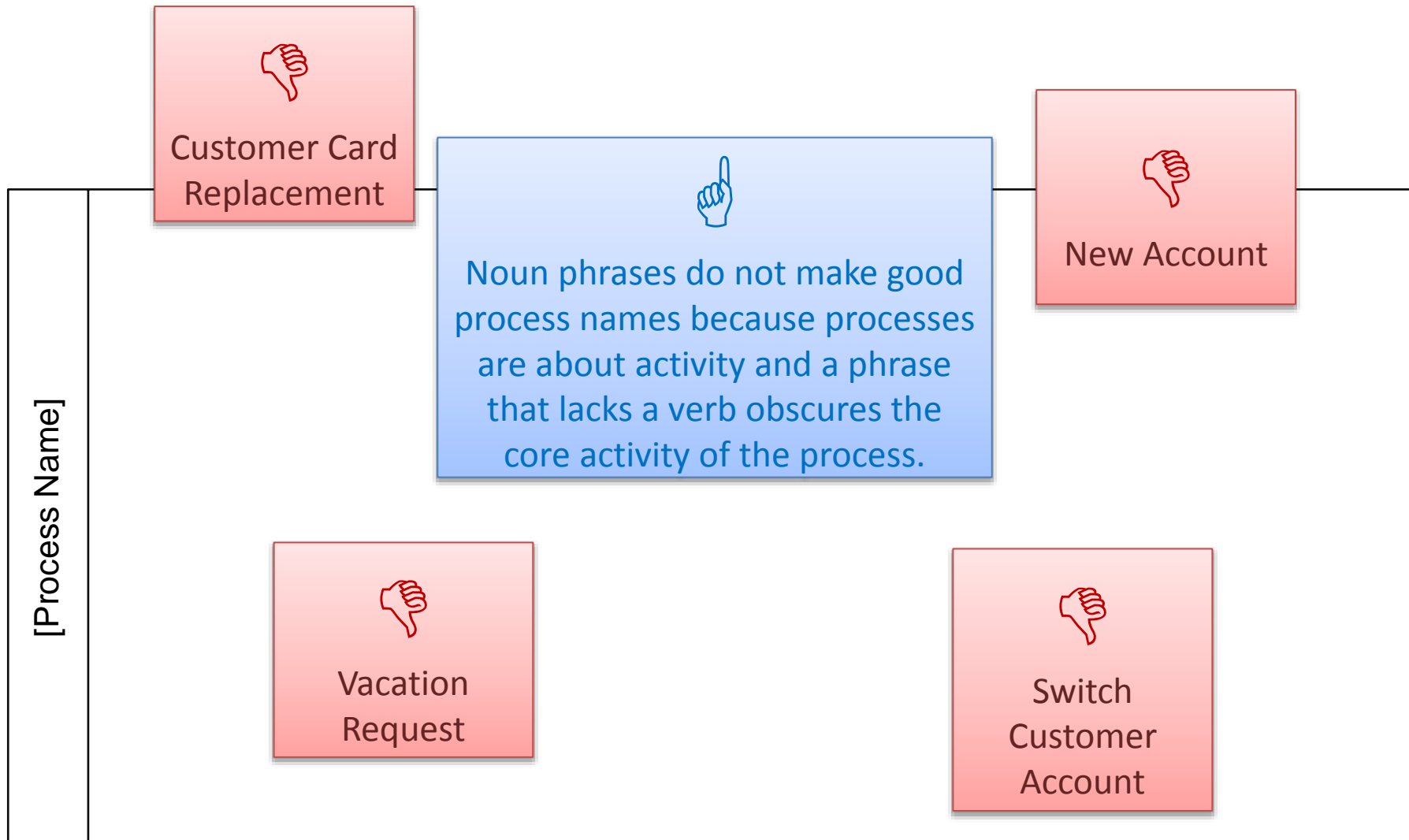
The process name  
goes here.

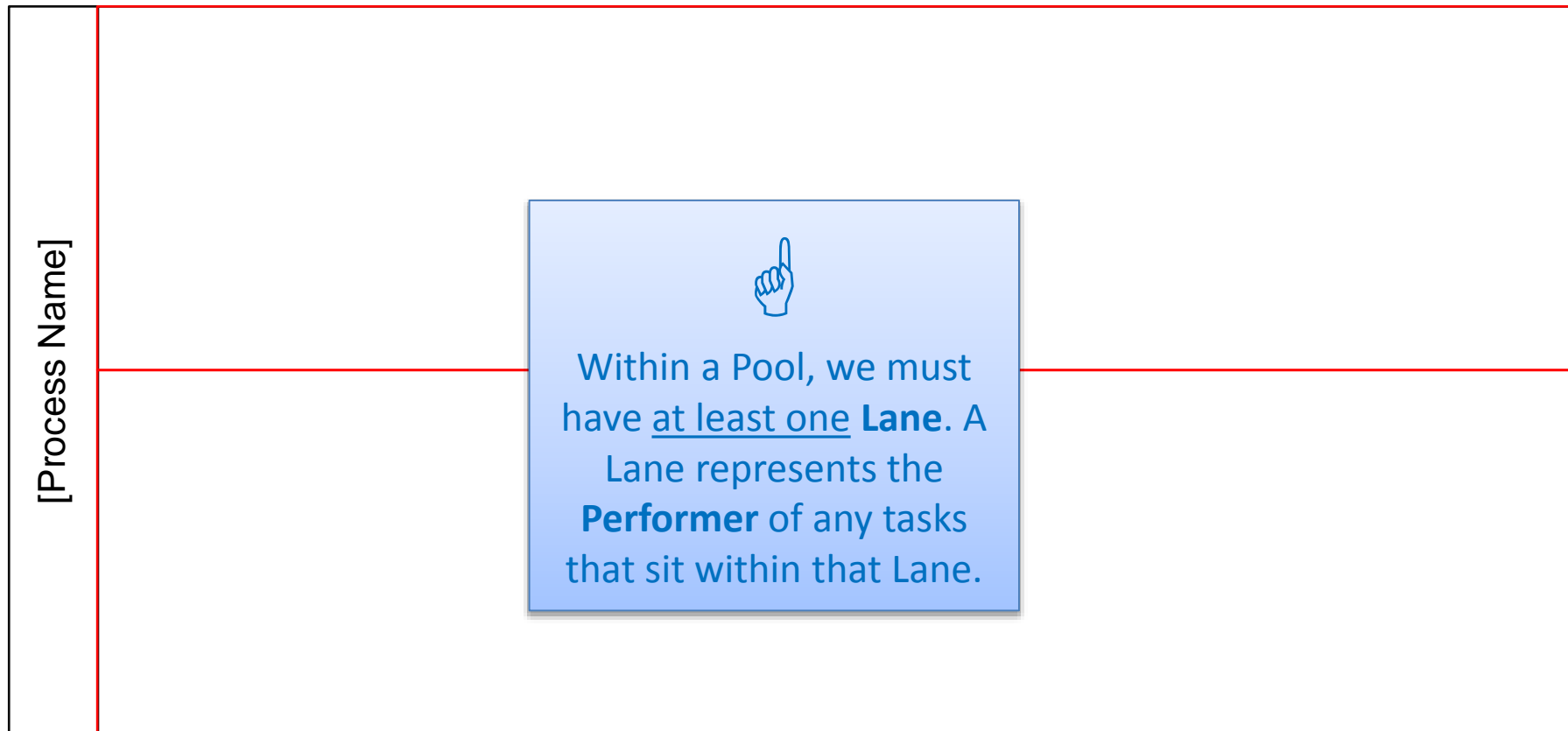
[Process Name]



Process names are always verb phrases and should reveal the intent of the process.







[Process Name]	
[Performer Name]	[Performer Name]




[Process Name]	<table><tr><td data-bbox="479 464 983 875">[Performer Name]</td><td data-bbox="983 464 2165 875"></td></tr><tr><td data-bbox="479 875 983 1285">[Performer Name]</td><td data-bbox="983 875 2165 1285"></td></tr></table>	[Performer Name]		[Performer Name]	
[Performer Name]					
[Performer Name]					






Performer names are always noun phrases and can represent individual roles, teams, departments or can represent a desire for Activities to be automated.




Personnel  
Manager



Accounts  
Department

[Process Name]	[Performer Name]	<div><p>These are all good Performer names.</p></div>	
	[Performer Name]	<div><p>Claims Adjuster</p></div>	<div><p>System (for Activities to be automated)</p></div>





[Process Name]	
[Performer Name]	<div data-bbox="945 642 1597 1113"><p>Those who perform activities within the same process must be modelled as Lanes within the <u>same</u> process Pool.</p></div>
[Performer Name]	


[Name]	
[Process Name]	[Performer Name]
	[Performer Name]

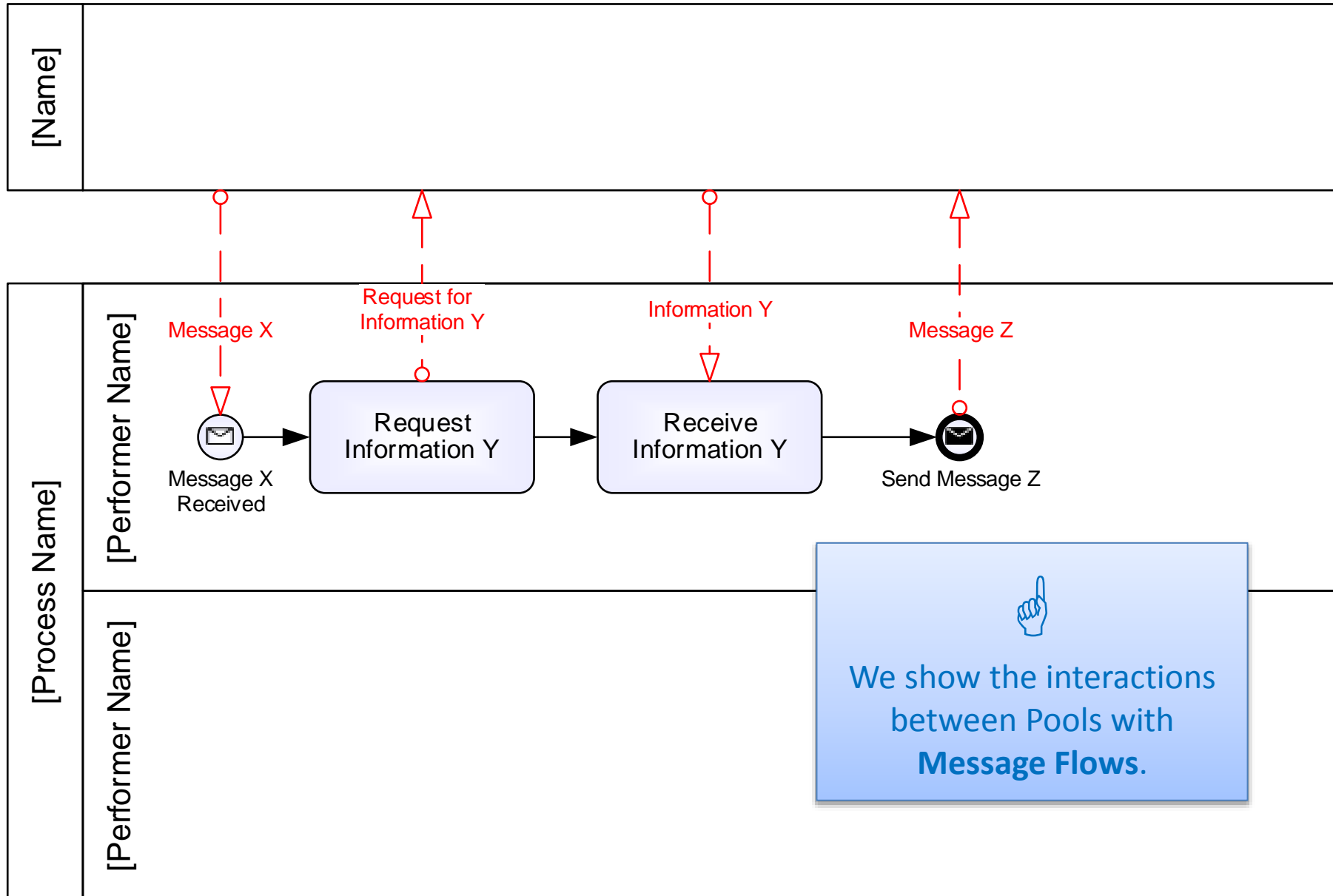


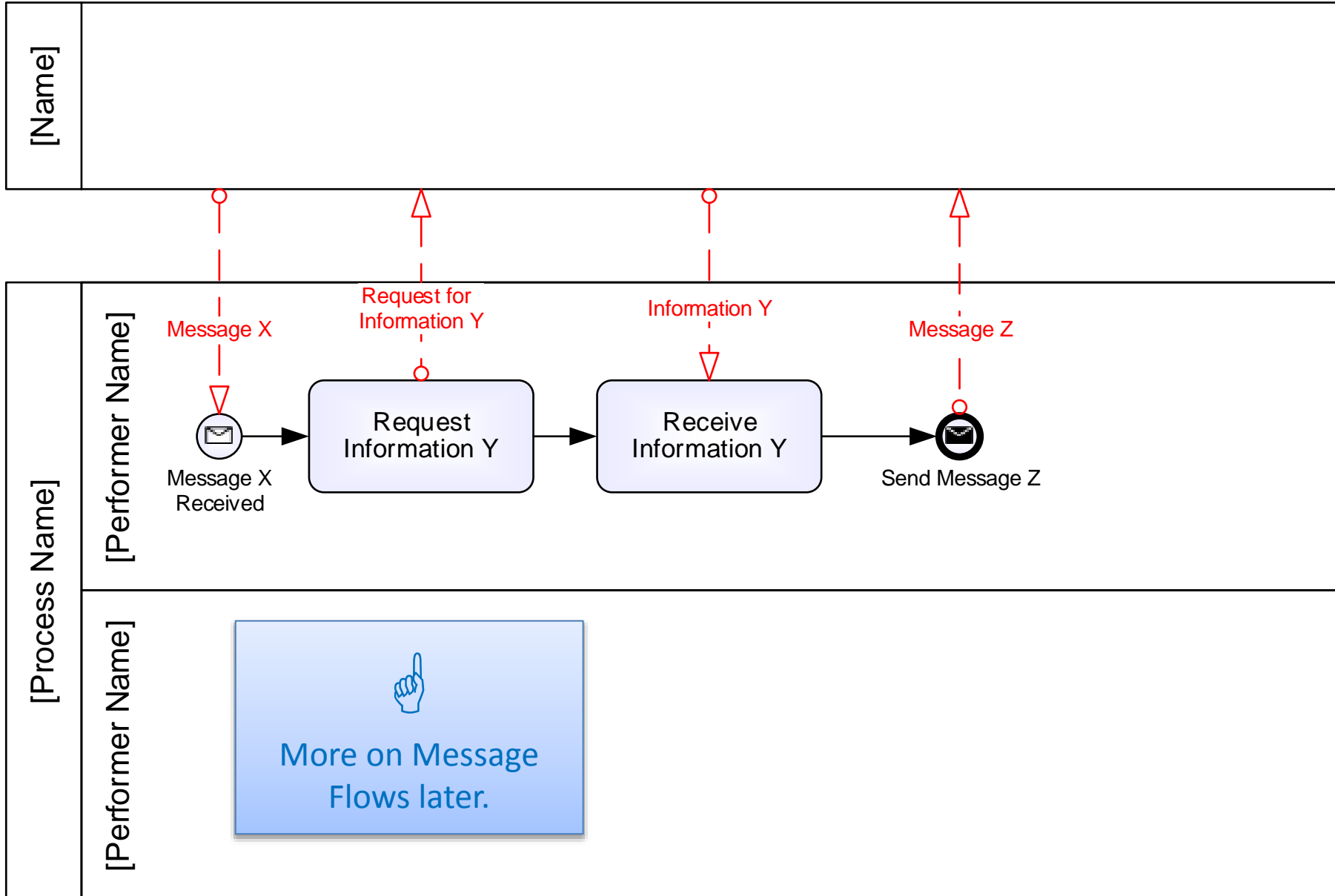
You can also have Pools outside the process, to indicate ***entities*** or ***other processes*** that interact with the process you are modelling.

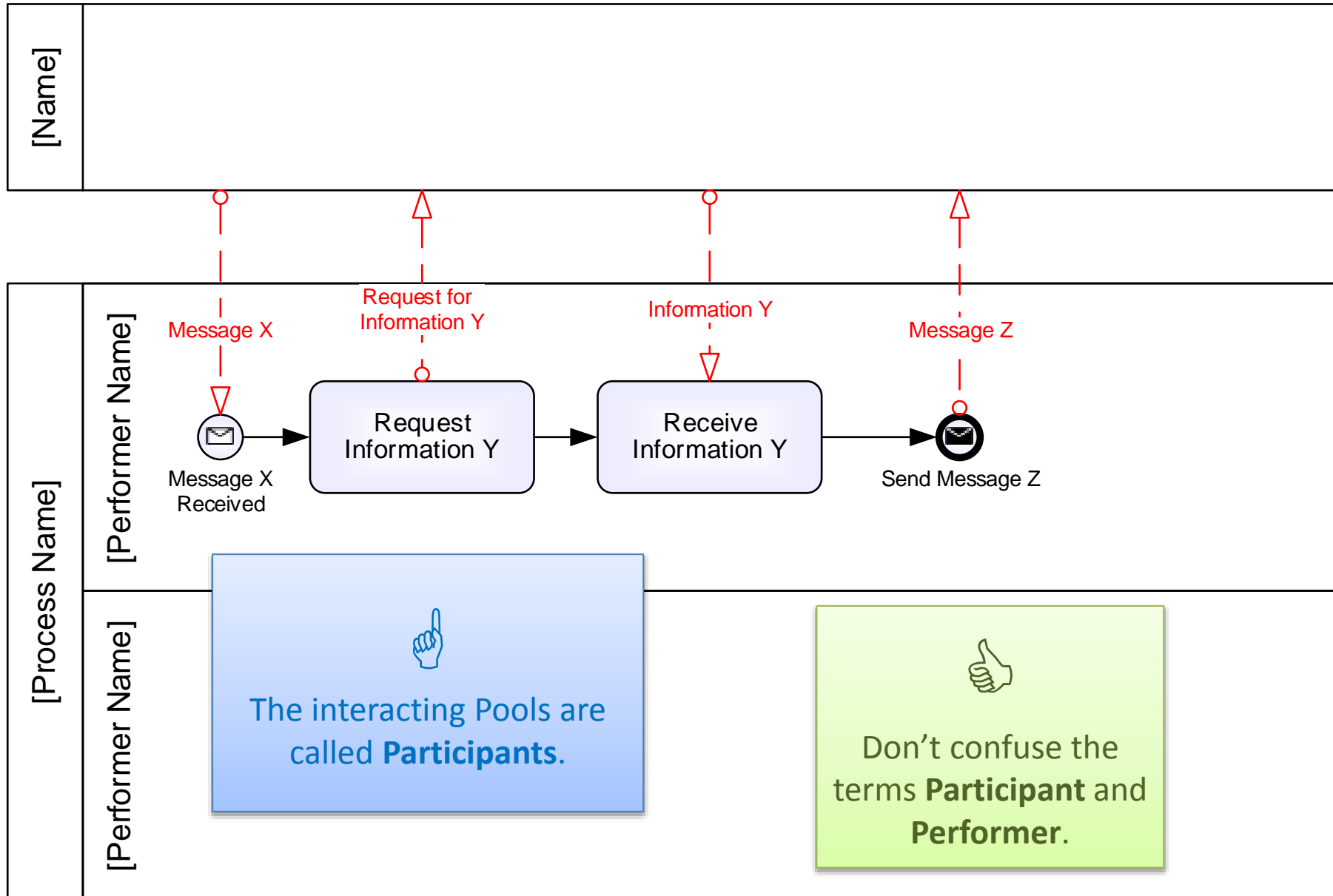
[Name]			
[Process Name]	[Performer Name]	<div><p>If we do <u>not</u> model what happens inside them, they are known as <b>Black Box</b> Pools.</p></div>	
	[Performer Name]		

[Name]			
[Process Name]	[Performer Name]	<div><p>We do not model what happens inside a Black Box Pool because:</p><ul style="list-style-type: none"><li>• we do not know</li><li>• or we do not care</li><li>• or we cannot control it</li></ul></div>	
	[Performer Name]		

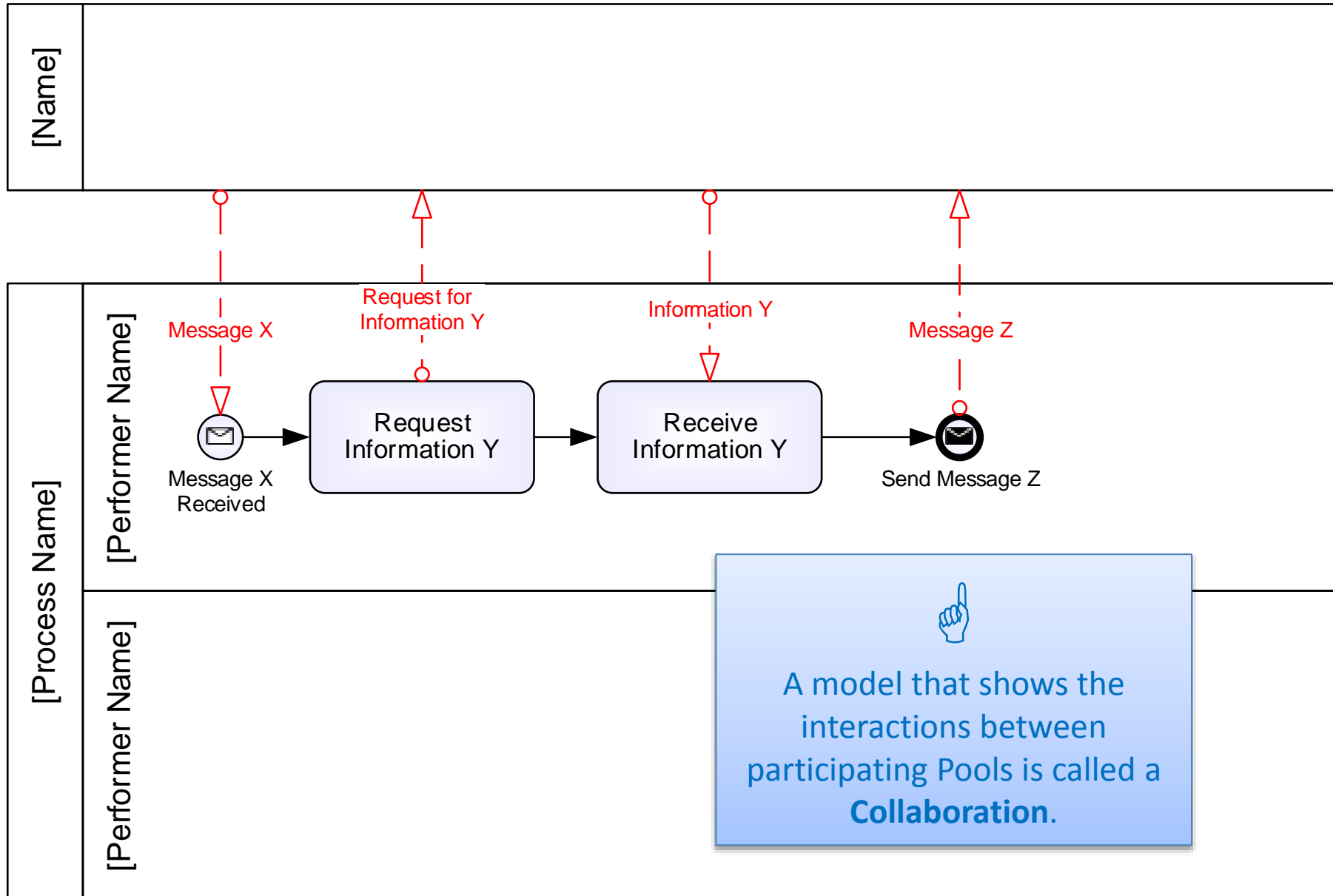
[Name]			
[Process Name]	[Performer Name]	<div><p>However, if we <u>do</u> know what goes on inside the other Pool <u>and</u> we can control it <u>and</u> we do care, then we model it as another <b>White Box</b> Pool.</p></div>	
	[Performer Name]		











# Events

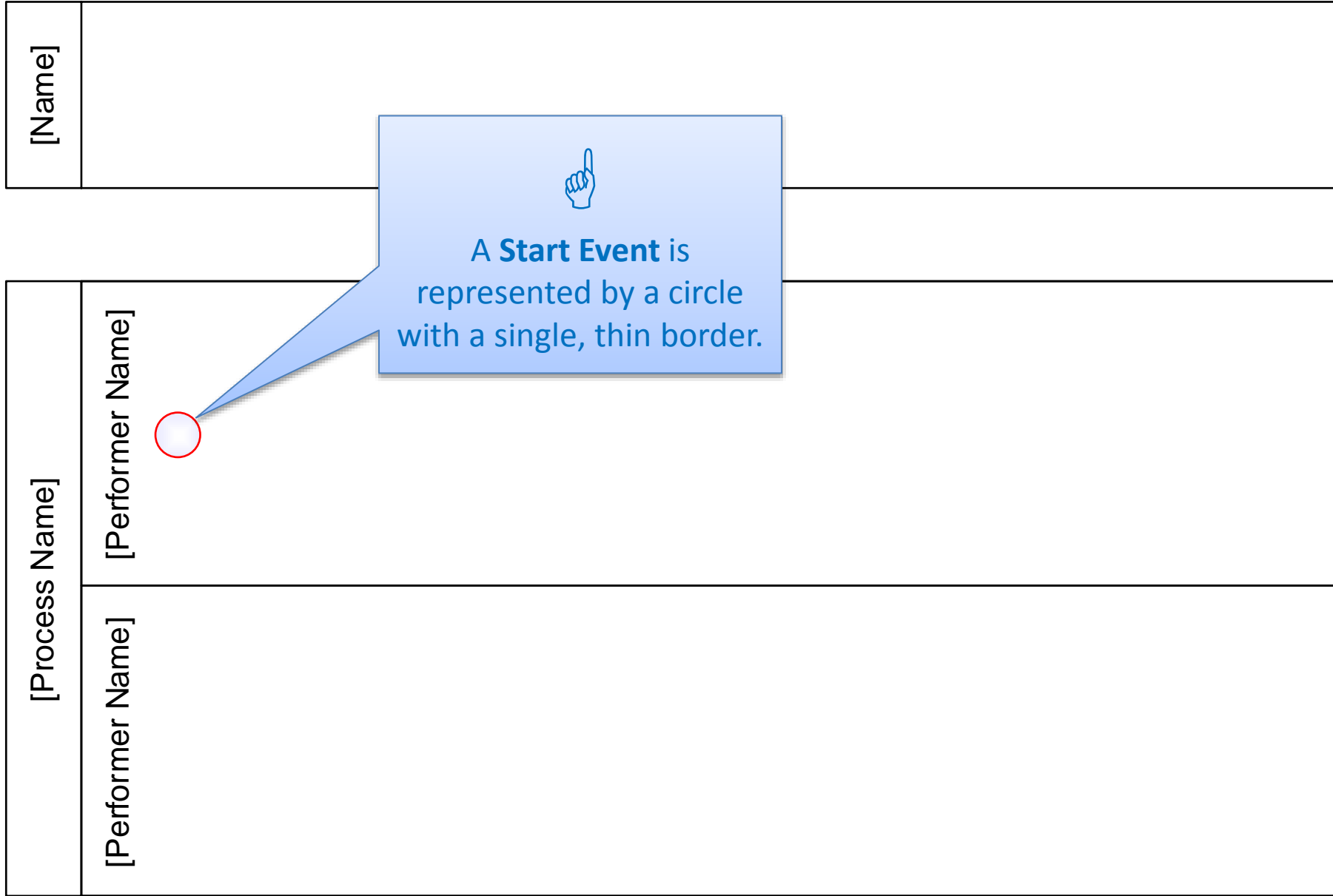
# Events




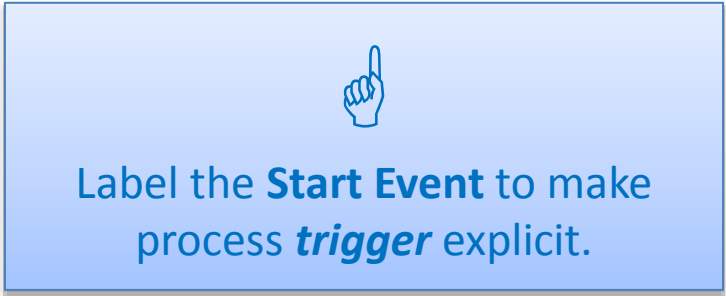
An **Event** is something that “happens” during the course of a **Process**... ***BPMN** has restricted the use of **Events** to include only those types of **Events** that will affect the sequence or timing of **Activities** of a **Process**.*  
(section 8.3.5)

[Name]	
--------	--

[Process Name]	<div data-bbox="507 498 563 841">[Performer Name]</div> <div data-bbox="575 636 644 701"></div> <div data-bbox="665 494 1304 848"><p>Every process has <i>at least one</i> <b>Event</b> which starts it.</p></div>
	<div data-bbox="507 906 563 1249">[Performer Name]</div>

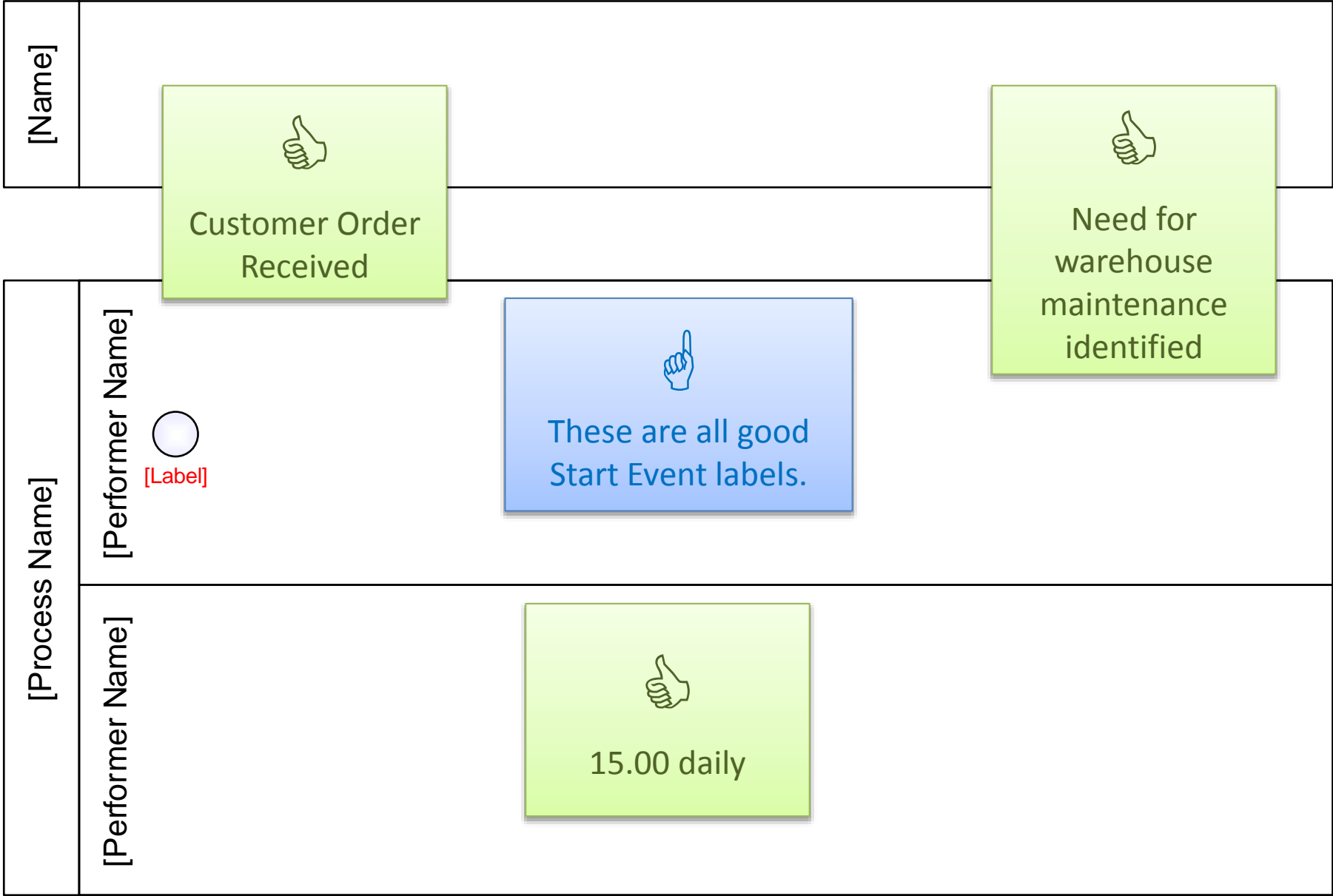


[Name]	
--------	--

[Process Name]	<div data-bbox="507 498 560 838">[Performer Name]</div> <div data-bbox="573 636 657 743"> [Label]</div> <div data-bbox="912 525 1633 818"><p>Label the <b>Start Event</b> to make process <i>trigger</i> explicit.</p></div>
	<div data-bbox="507 906 560 1246">[Performer Name]</div>

[Name]	
--------	--

[Process Name]	<div data-bbox="507 501 558 836">[Performer Name]</div> <div data-bbox="575 639 644 704"></div> <div data-bbox="565 714 654 742">[Label]</div>	<div data-bbox="1266 479 1358 591"></div> <div data-bbox="935 636 1699 798">“Start” is not a suitable label for a Start Event, as it adds no information about what <u>causes</u> the process to start.</div>
	<div data-bbox="507 912 558 1248">[Performer Name]</div>	



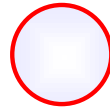




The three basic types  
of Start Event in  
BPMN 2.0 are:



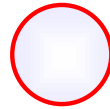
The three basic types  
of Start Event in  
BPMN 2.0 are:




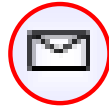
**“None”**: the process  
is started manually.




The three basic types  
of Start Event in  
BPMN 2.0 are:



  
“None”: the process  
is started manually.




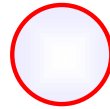
  
“**Message**”: the  
process starts when a  
Message is received  
from another Pool.




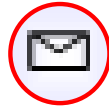
The three basic types  
of Start Event in  
BPMN 2.0 are:



A Message is not necessarily a letter  
or an email. It could be one person  
talking to another. A message  
represents nothing more than an  
interaction between two Pools.



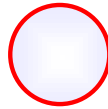
“None”: the process  
is started manually.



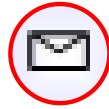
“Message”: the  
process starts when a  
Message is received  
from another Pool.



The three basic types  
of Start Event in  
BPMN 2.0 are:



“None”: the process  
is started manually.

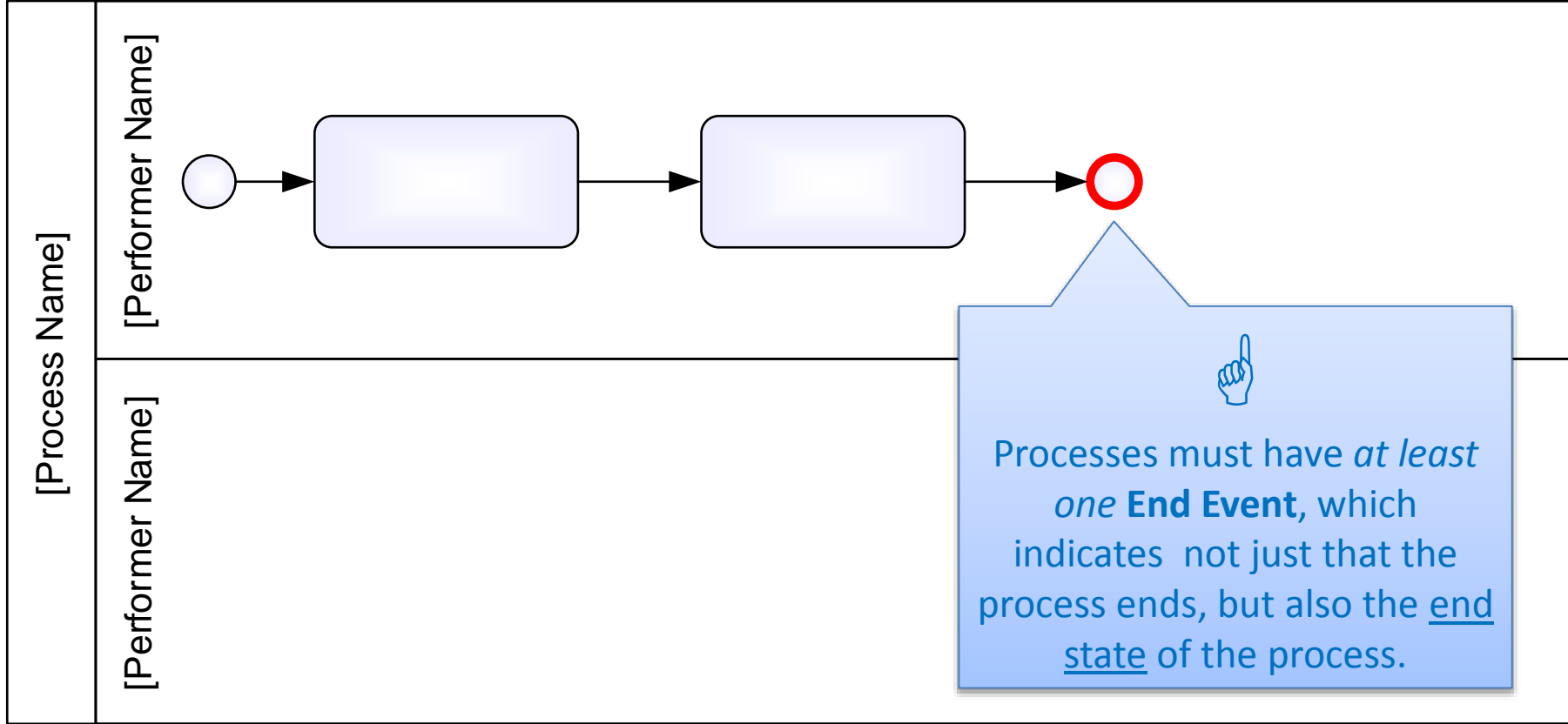


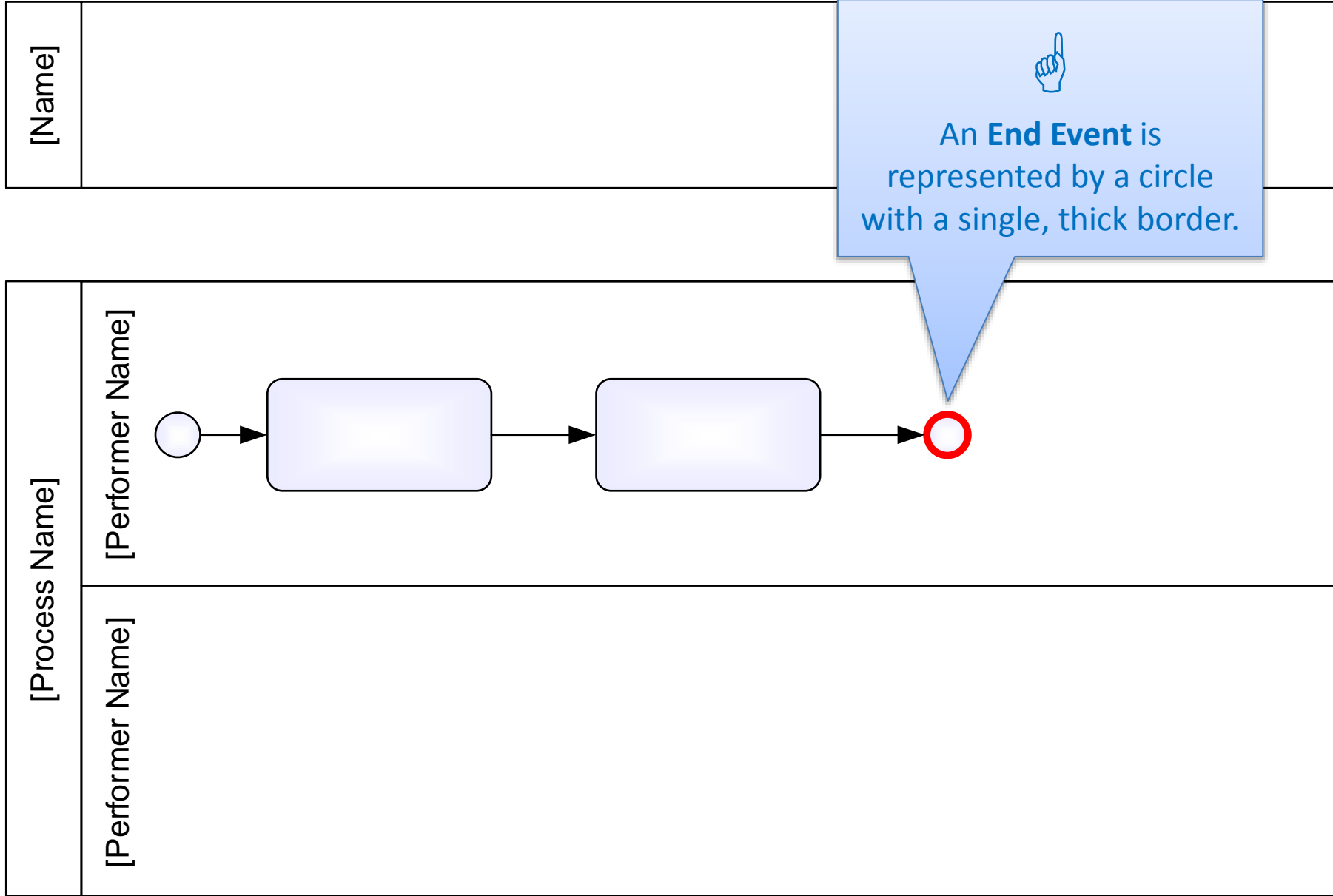
“Message”: the  
process starts when a  
Message is received  
from another Pool.



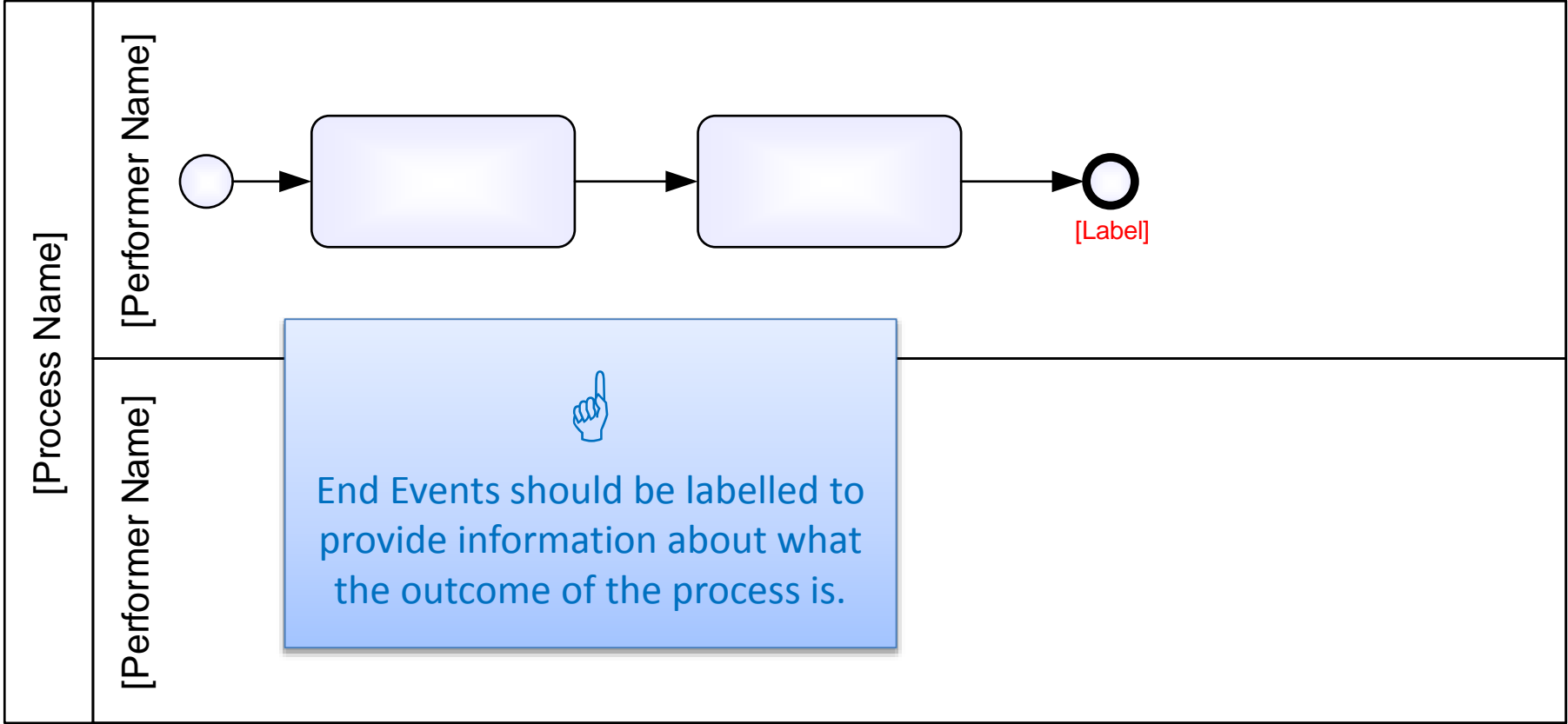
“**Timer**”: the process  
starts at a particular  
time, or date, or  
frequency, etc.

[Name]	
--------	--

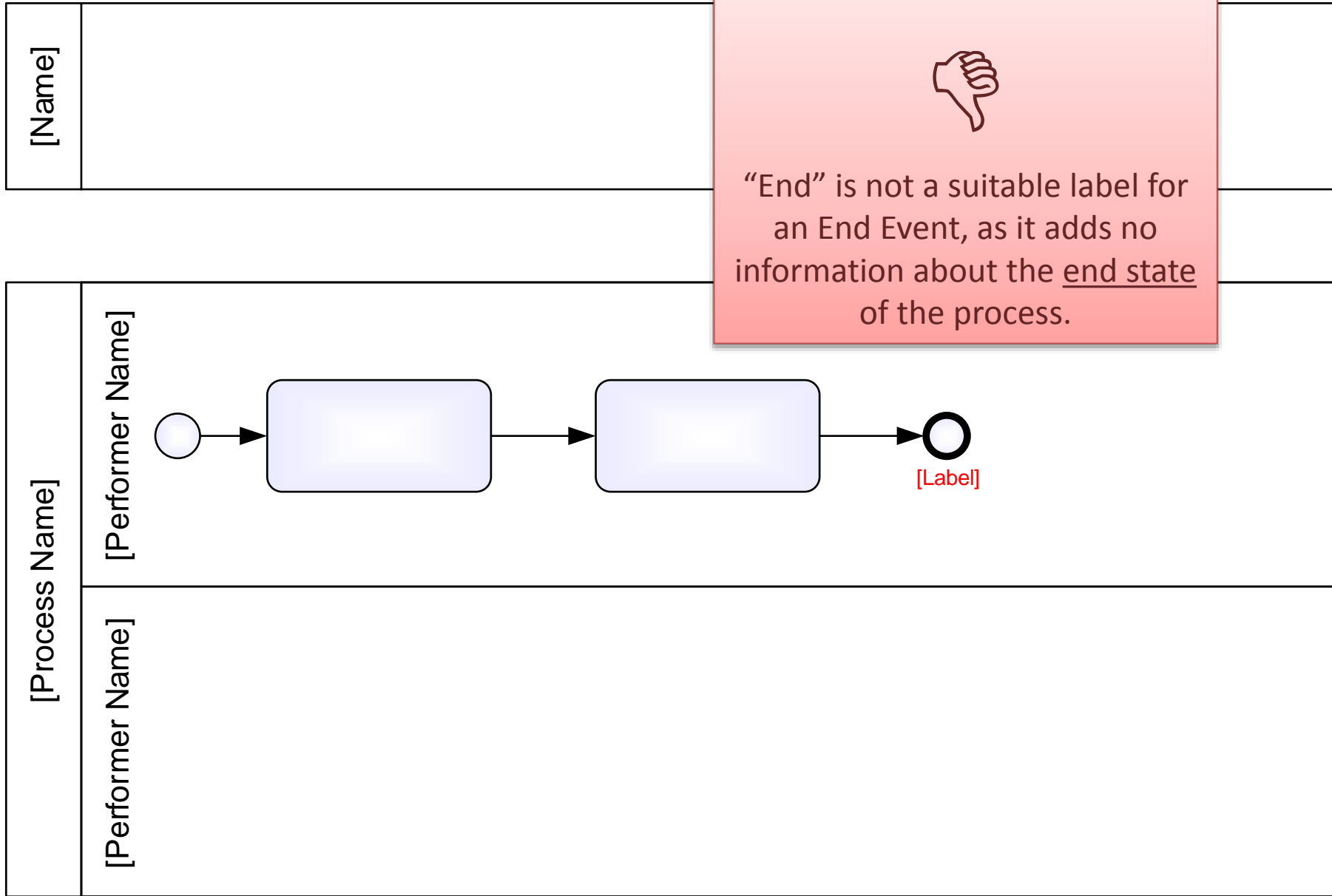


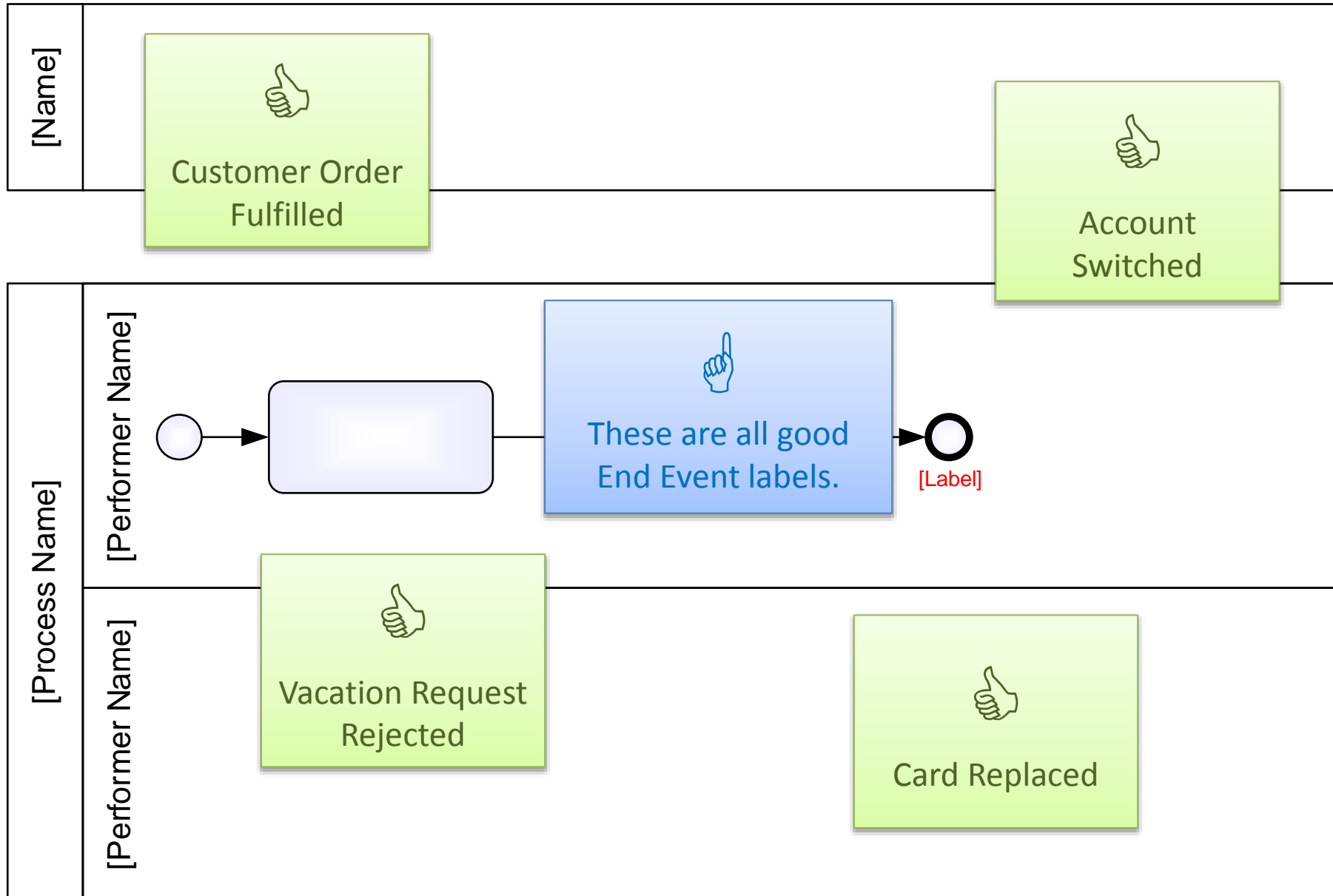


[Name]	
--------	--

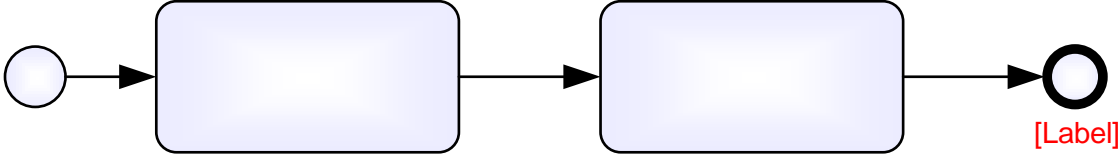



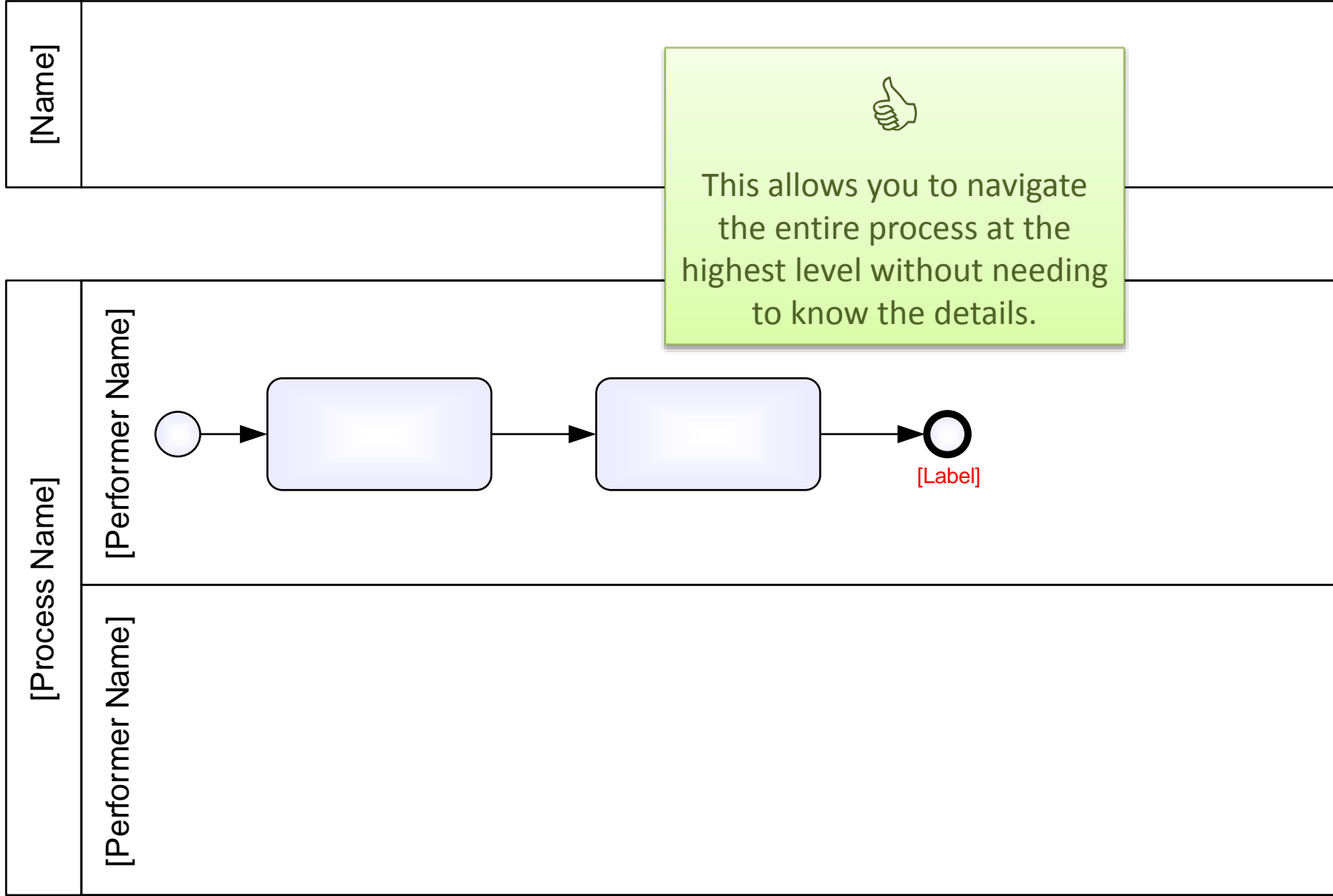






[Name]	
--------	--

[Process Name]	<div data-bbox="507 498 558 836">[Performer Name]</div> 
	<div data-bbox="507 908 558 1246">[Performer Name]</div> <div data-bbox="871 795 1592 1218">  <p><i>All possible Start Events and <u>all</u> possible End Events <u>must</u> be shown in the process Pool at the top level of your model.</i></p> </div>





The three basic types  
of End Event in BPMN  
2.0 are:



The three basic types  
of End Event in BPMN  
2.0 are:



**“None”**: the process  
throws no signal at  
the end.



The three basic types  
of End Event in BPMN  
2.0 are:



“None”: the process  
throws no signal at  
the end.



“**Message**”: the  
process throws a  
Message signal *to  
another Pool.*



The three basic types  
of End Event in BPMN  
2.0 are:



“None”: the process  
throws no signal at  
the end.



“Message”: the  
process throws a  
Message signal *to*  
***another Pool.***



“**Terminate**”: all  
active paths *at the*  
*same process level*  
are aborted.



# Activities

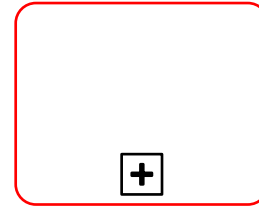
# Activities

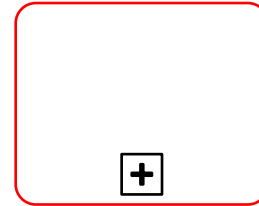


An **Activity** is work that is performed  
within a **Business Process**.  
(section 10.2)

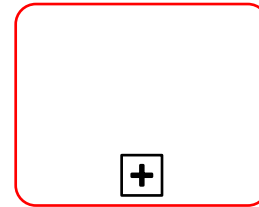


The performance of work  
can *only* be represented  
by **Activity** shapes.

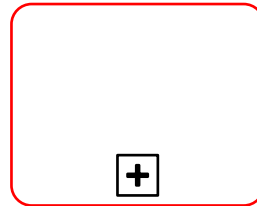




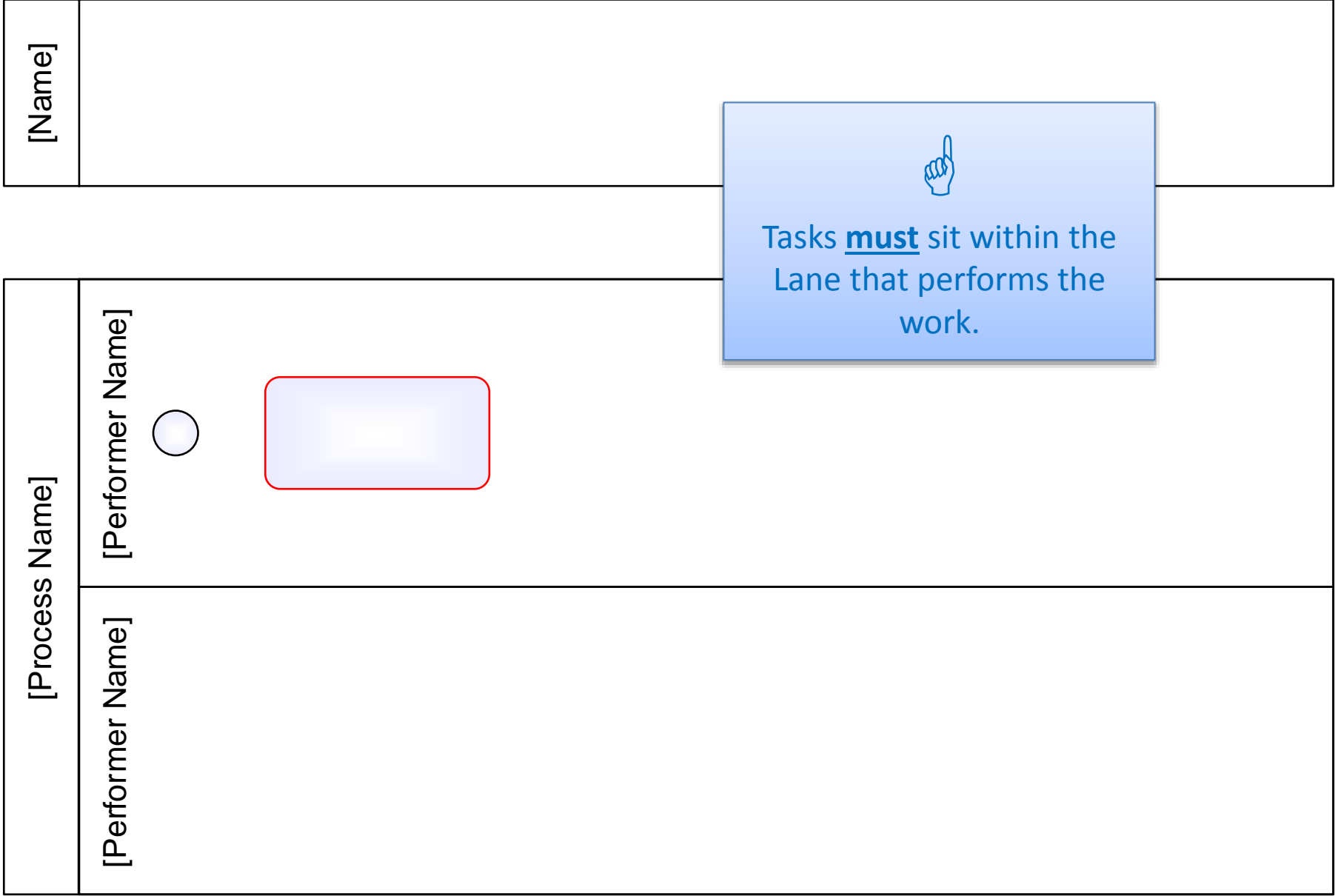
*Atomic* activities cannot be broken down into further steps and are represented by the **Task** shape.

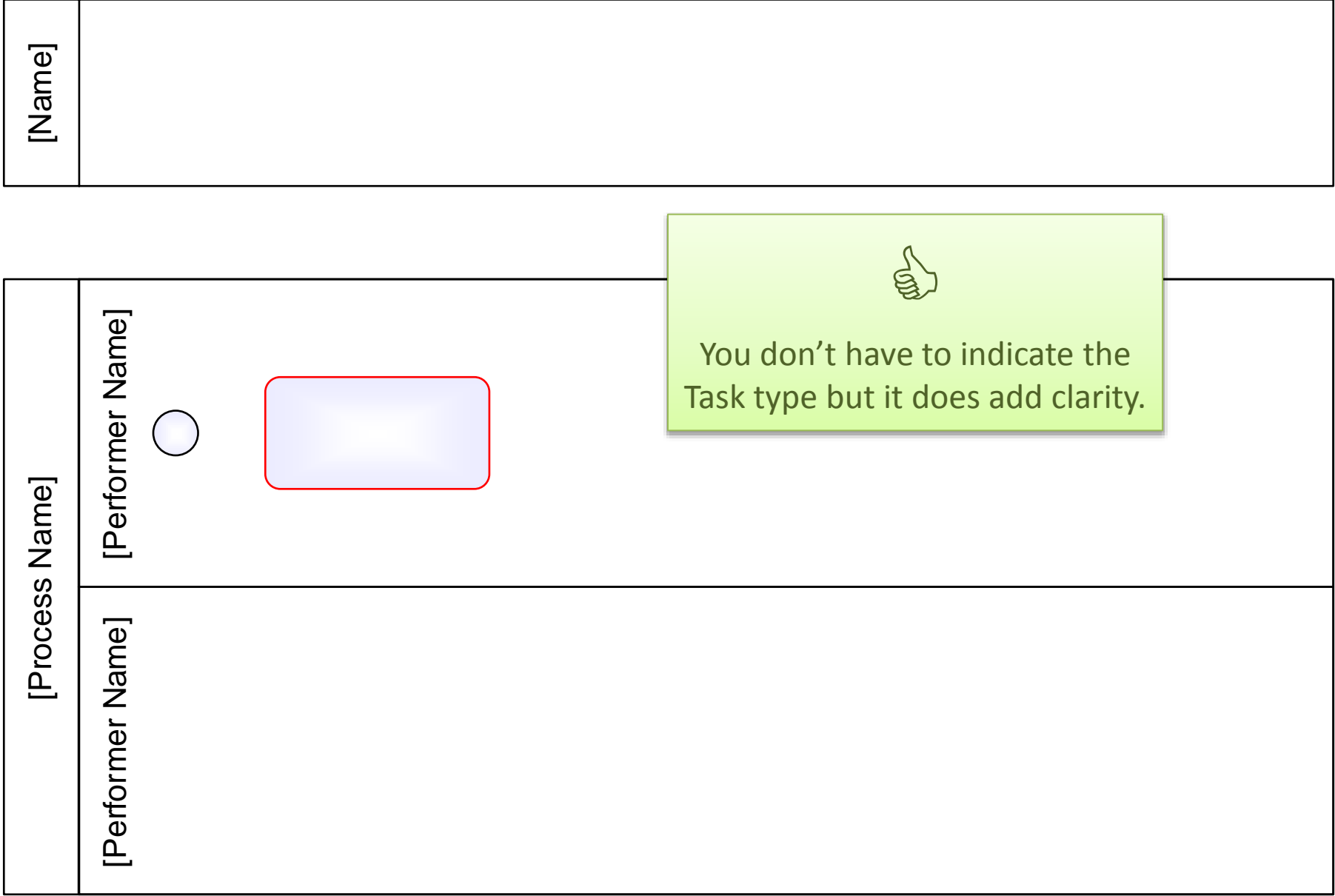


The **Sub Process** shape is used for Activities that are broken down into further Tasks or Sub Processes.



More on Sub  
Processes shortly.





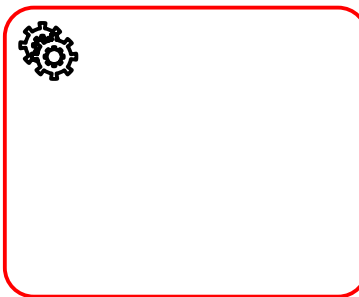




### **User (person symbol):**

In the Level 1 palette, the User Task type represents any atomic activity carried out by a *human*.





**Service (gears symbol):**  
In the Level 1 palette, the  
Service Task type represents any  
***automated***, atomic activity.

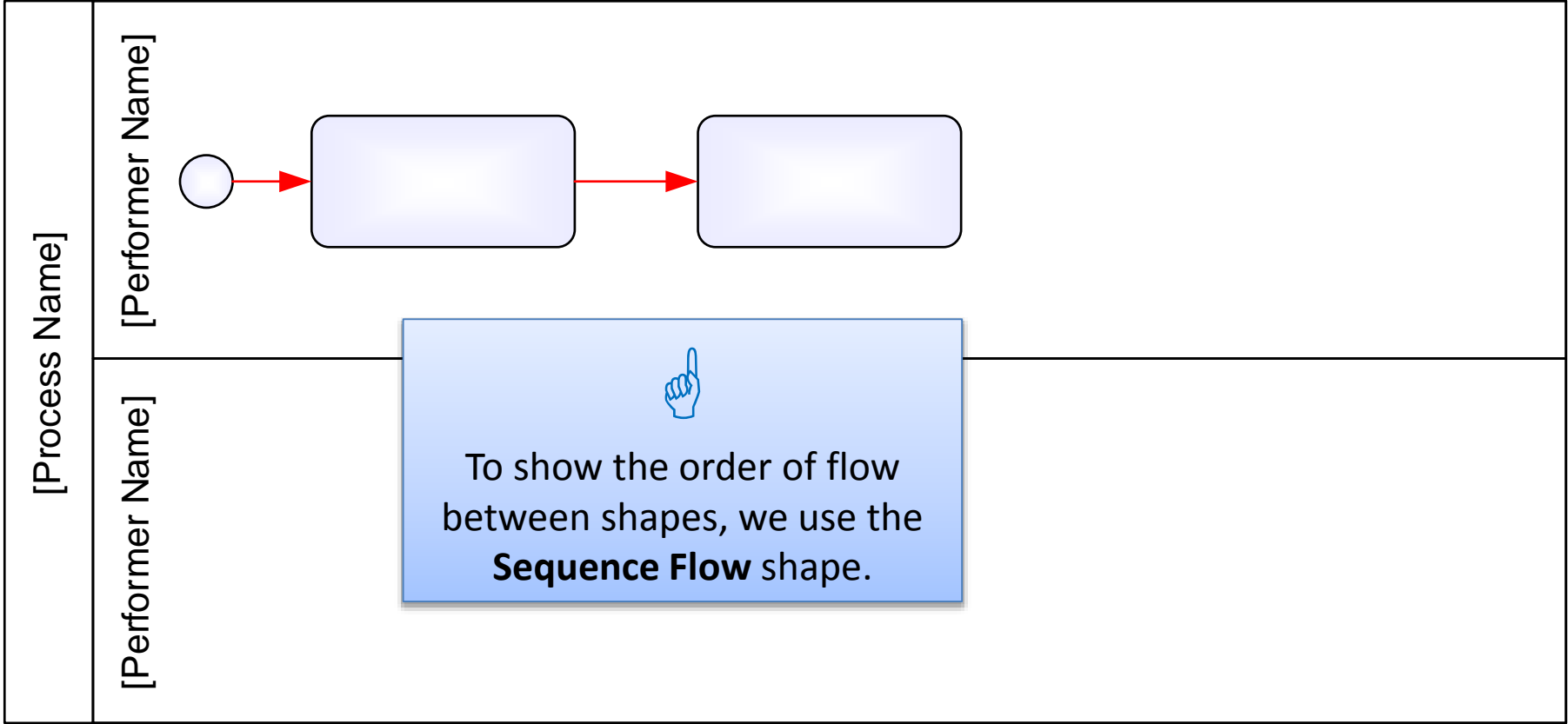
# Sequence Flows

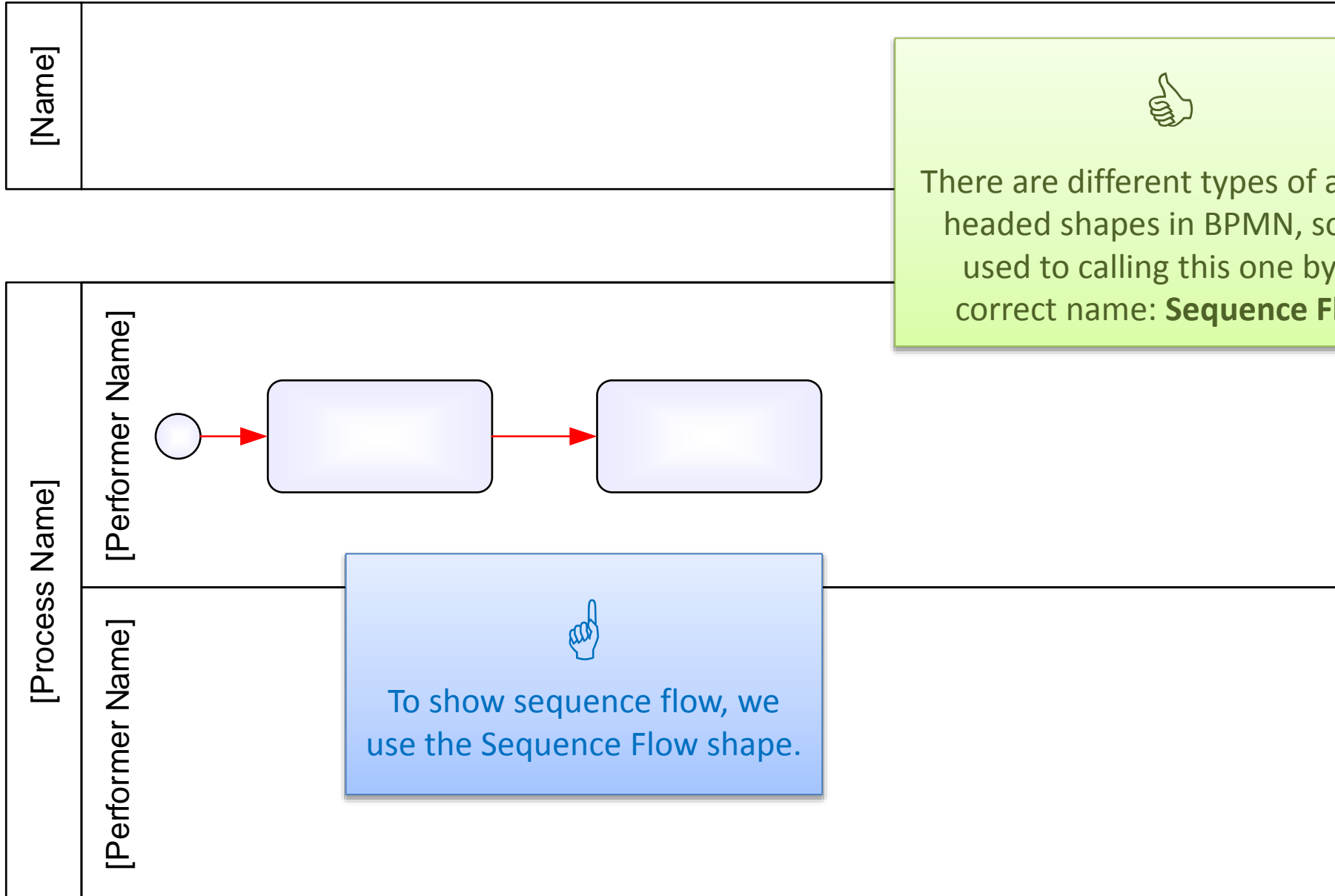
# Sequence Flows



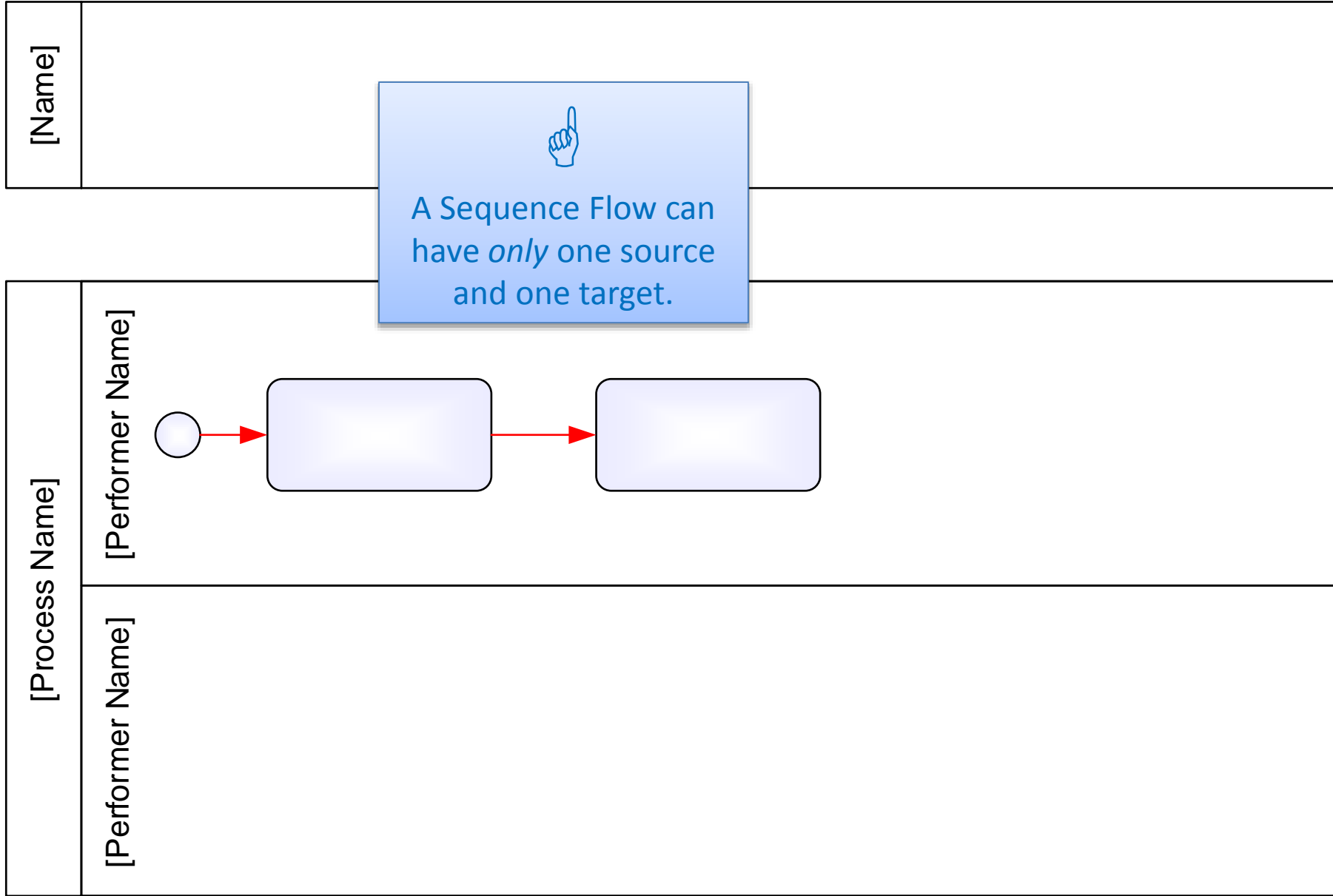
A **Sequence Flow** is used to show the order of Flow Elements in a Process.  
(section 8.3.13)

[Name]	
--------	--

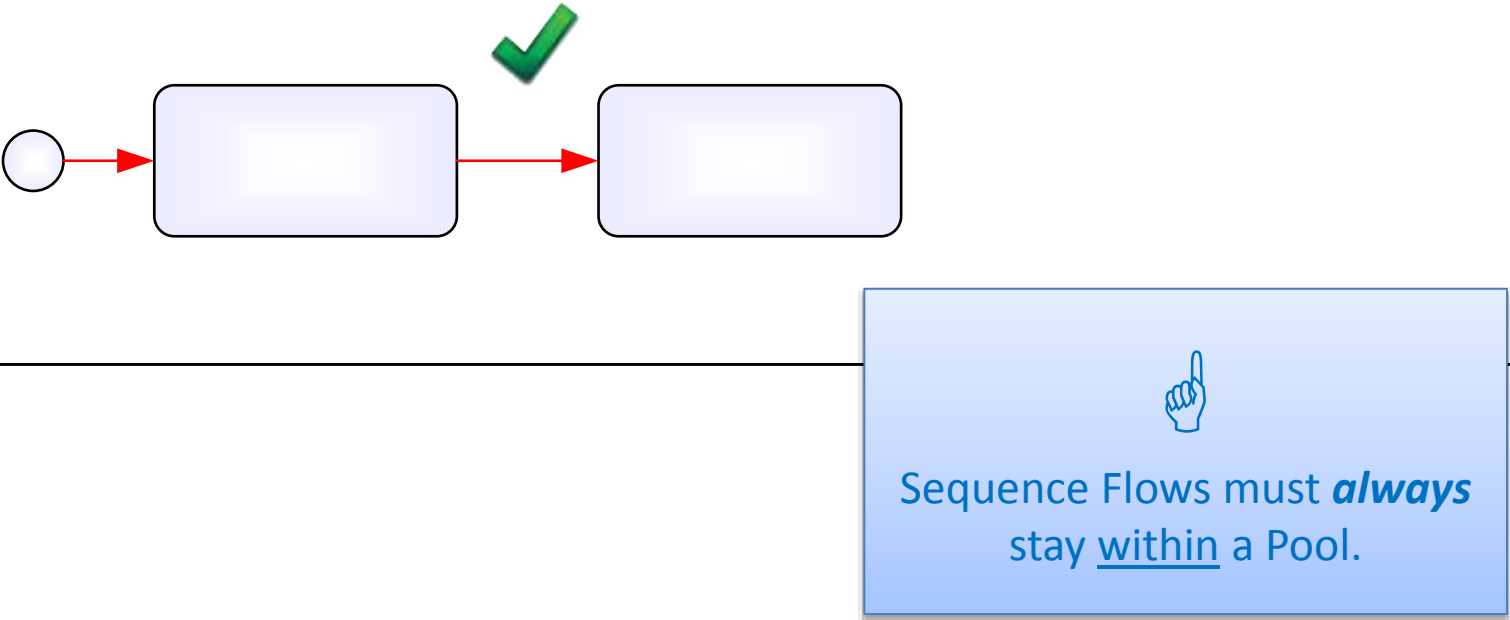




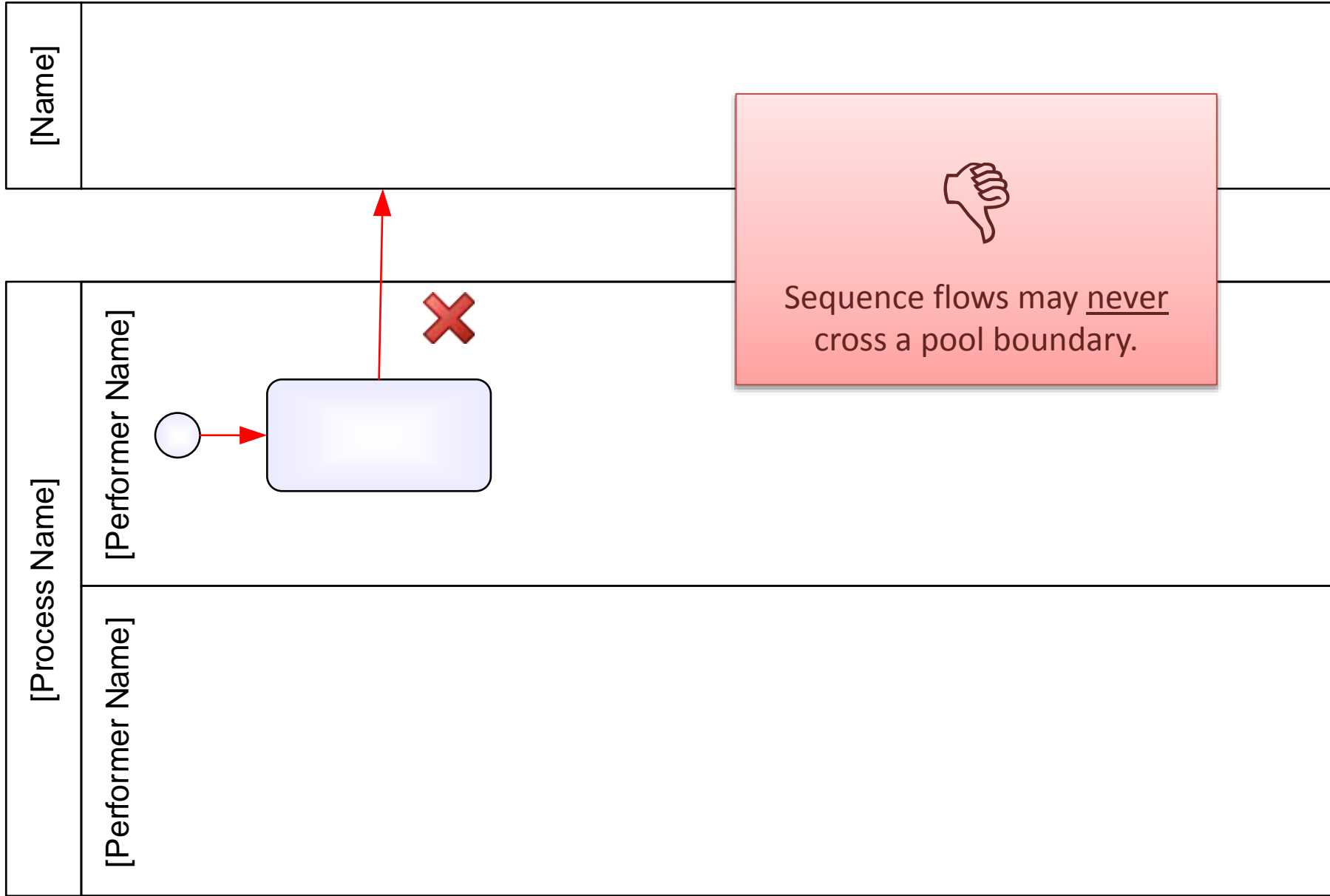
There are different types of arrow-headed shapes in BPMN, so get used to calling this one by its correct name: **Sequence Flow**



[Name]	
--------	--

[Process Name]	
[Performer Name]	<div></div>





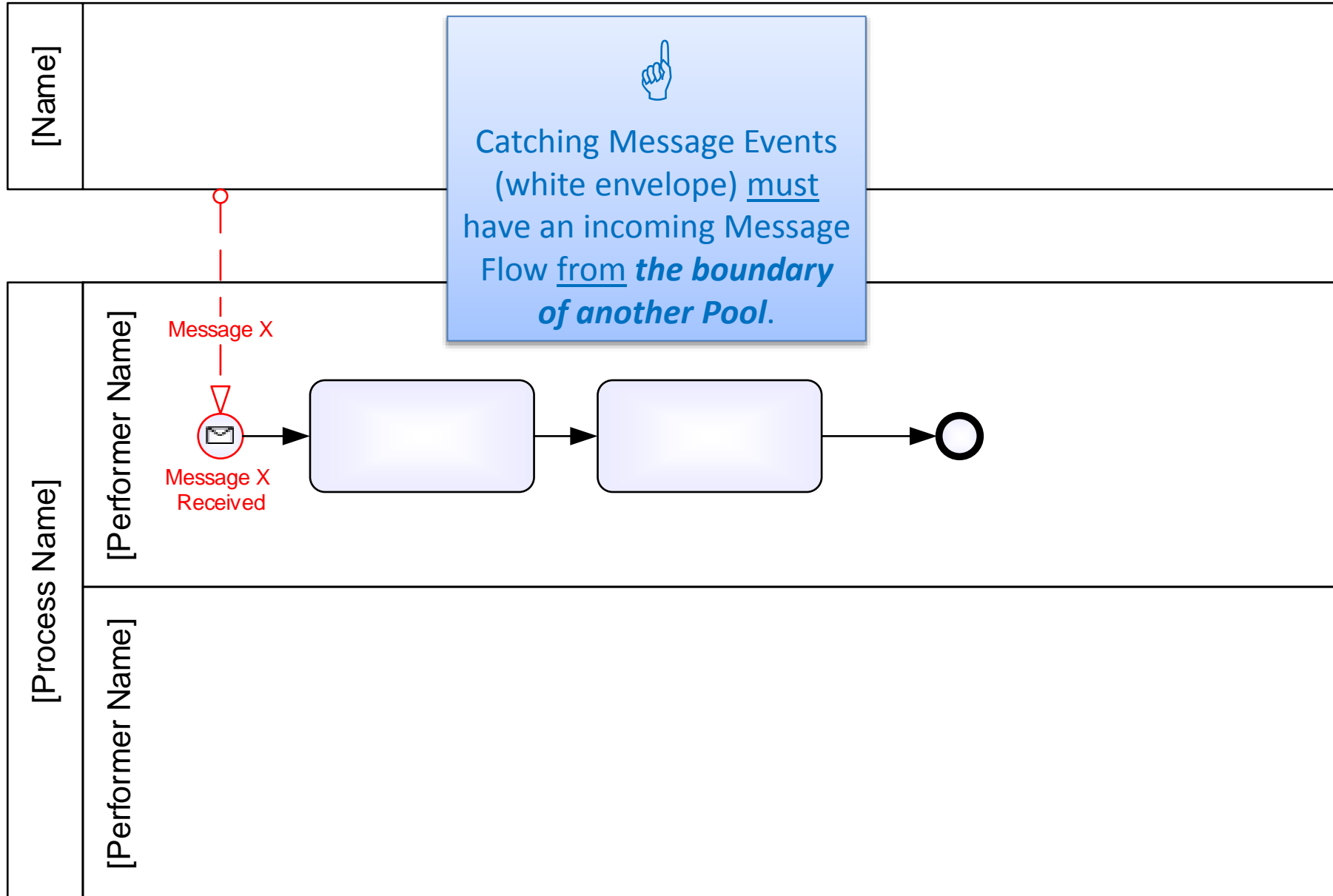
# Message Flows

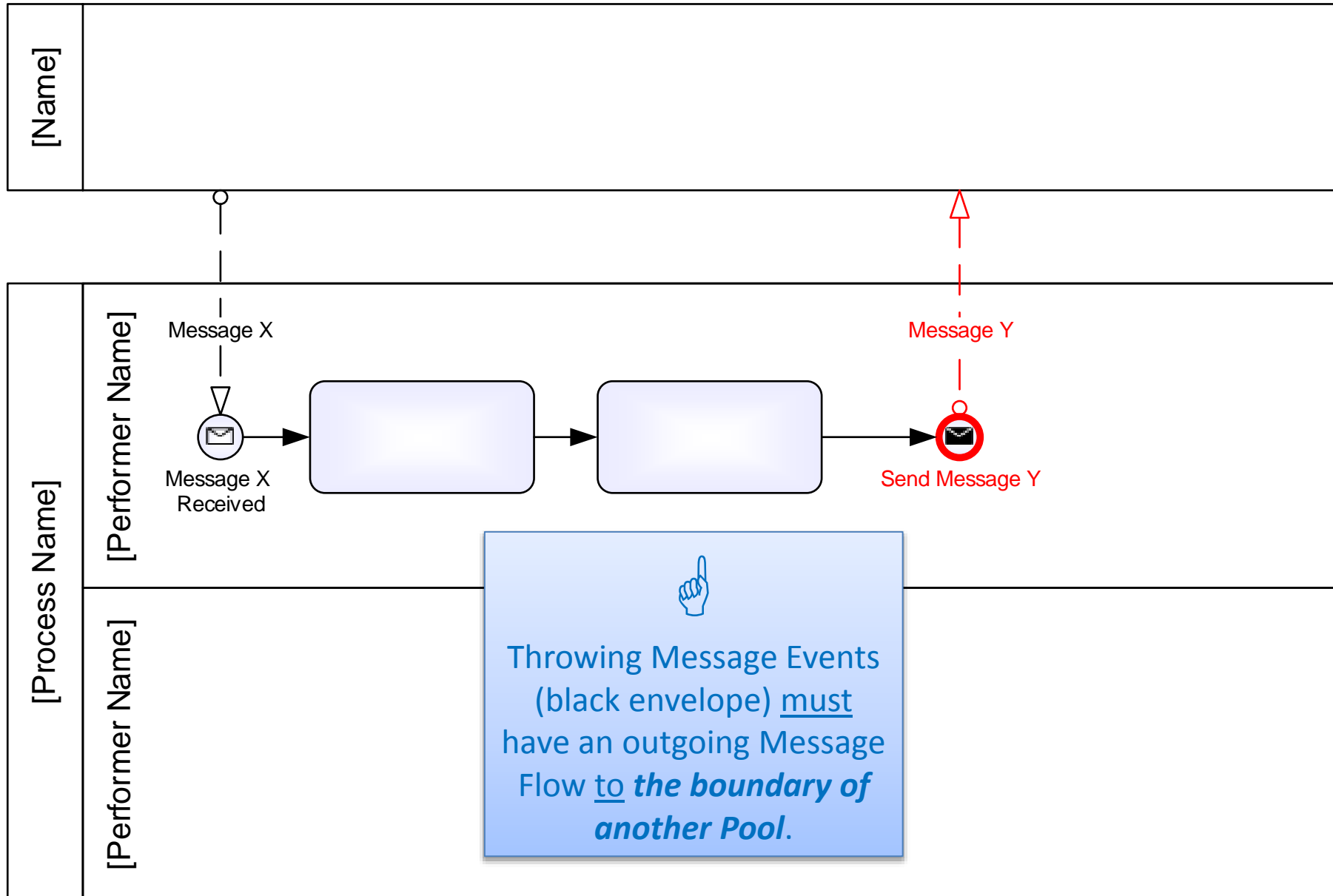
# Message Flows

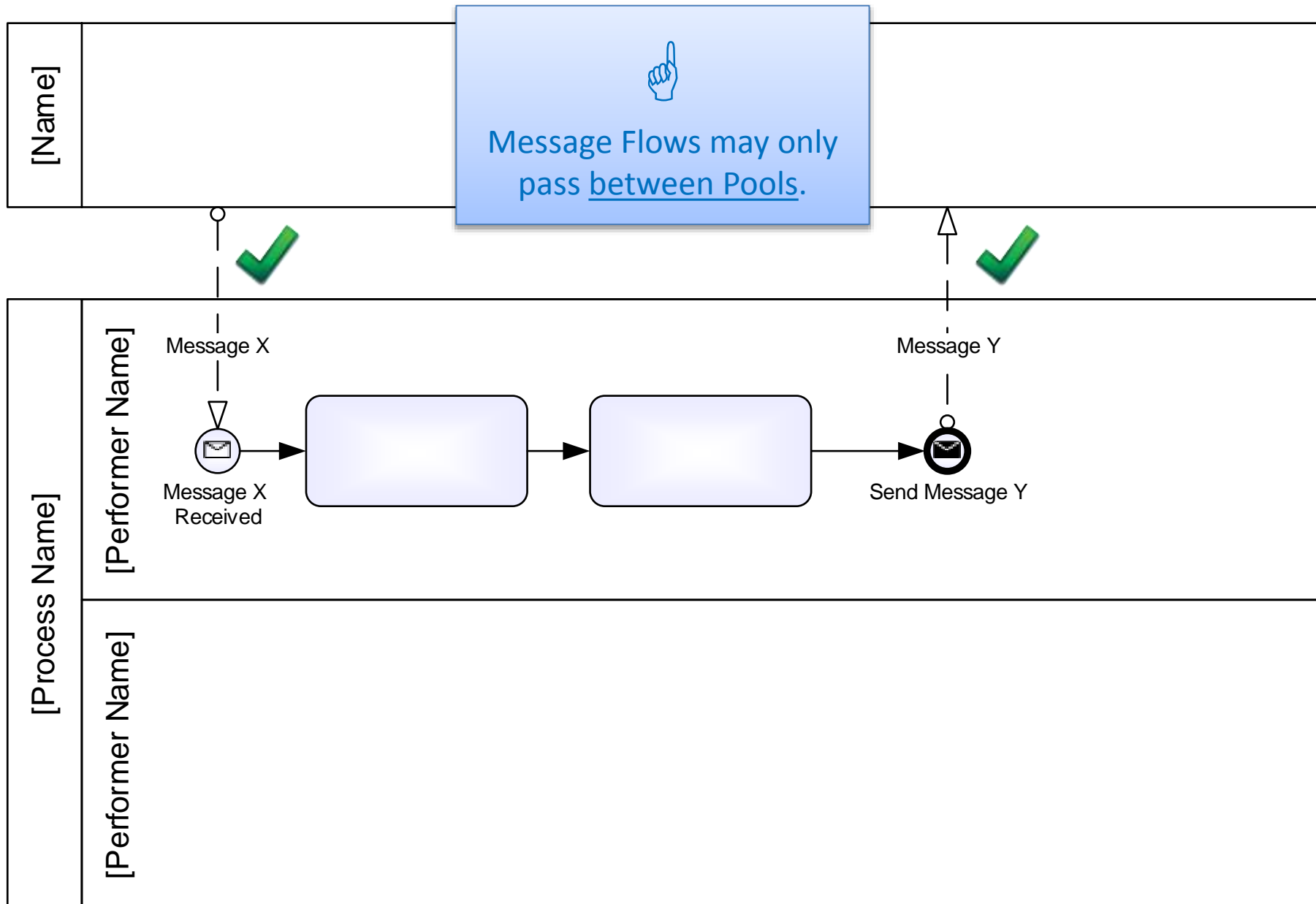


A **Message Flow** is used to show the flow of **Messages** between two Participants that are prepared to send and receive them.

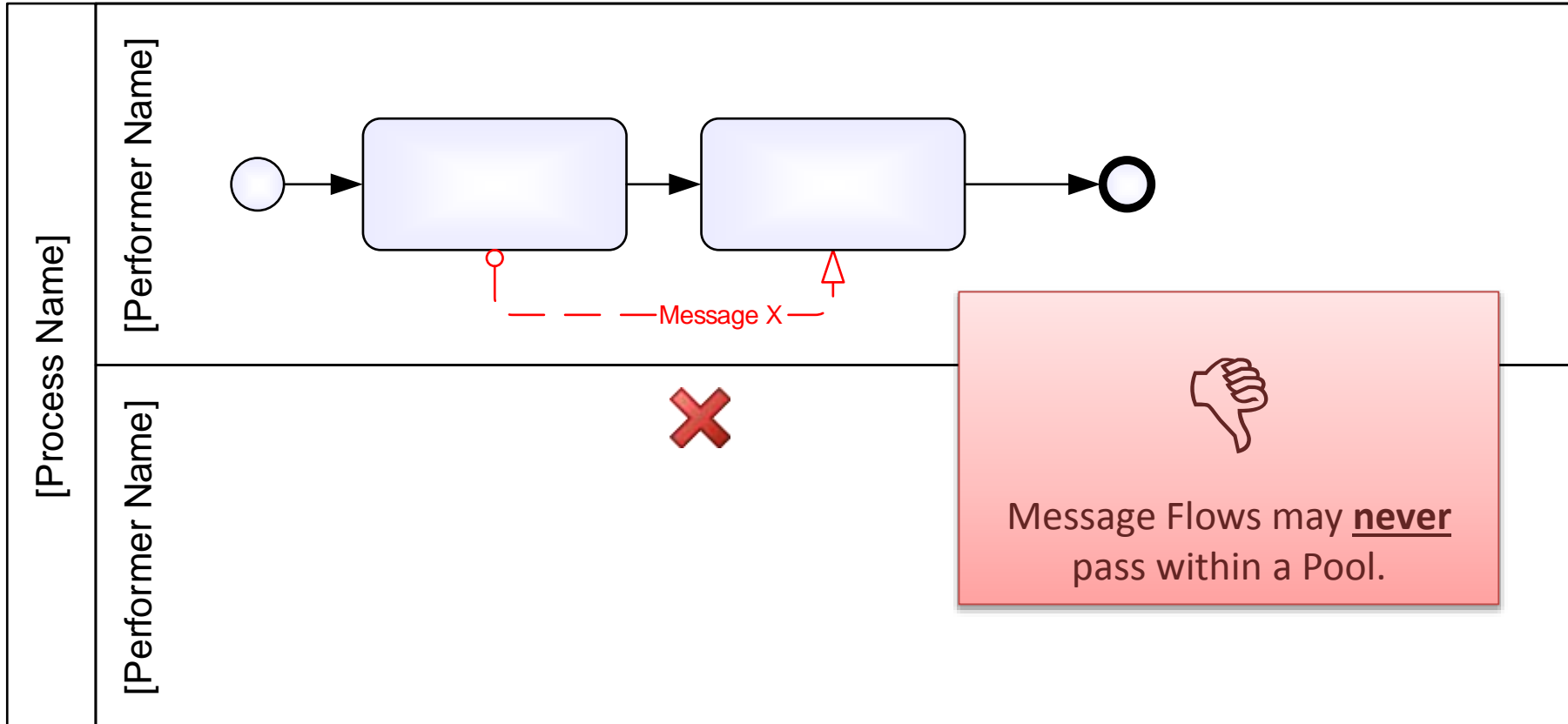
(section 9.3)

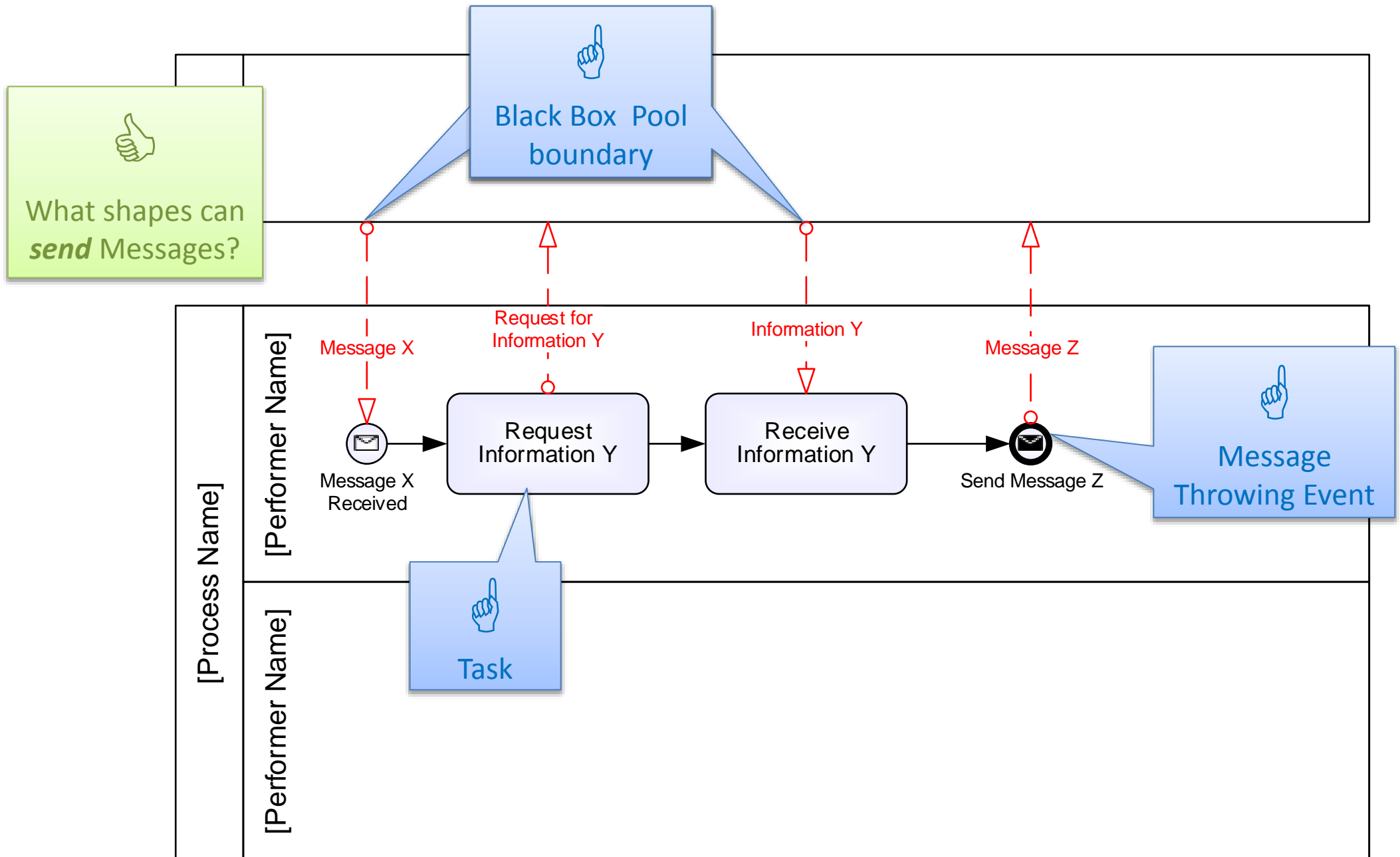




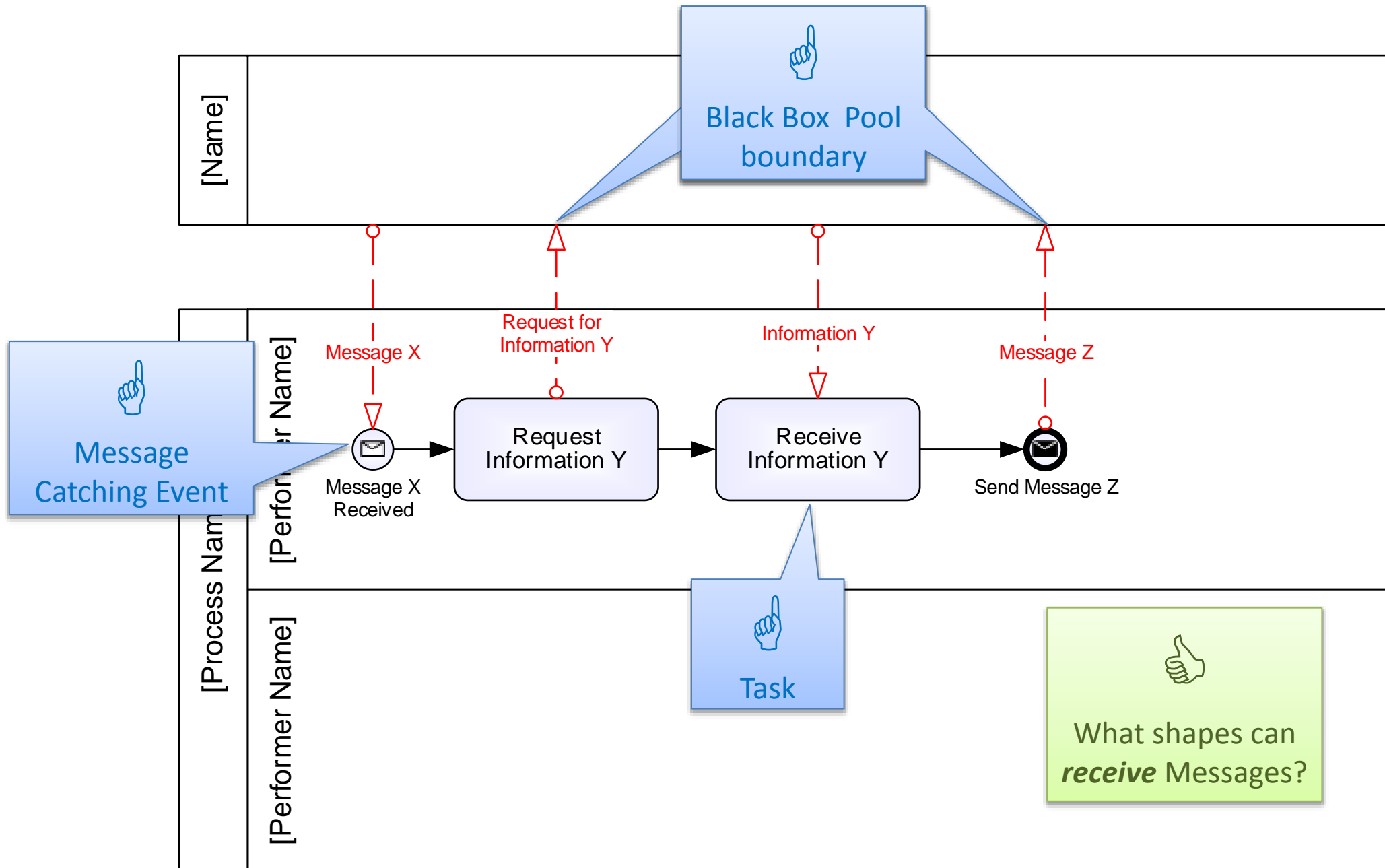


[Name]	
--------	--









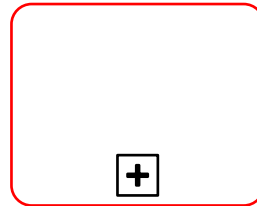
# Sub Processes

# Sub Processes



A **Sub-Process** is an **Activity** whose internal details have been modeled using **Activities**, **Gateways**, **Events**, and **Sequence Flows**. A **Sub-Process** is a graphical object within a **Process**, but it also can be “opened up” to show a lower-level **Process**.

(section 8.3.13)

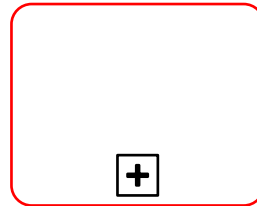


Sub Processes allow you to  
model the details of a Process  
***hierarchically.***



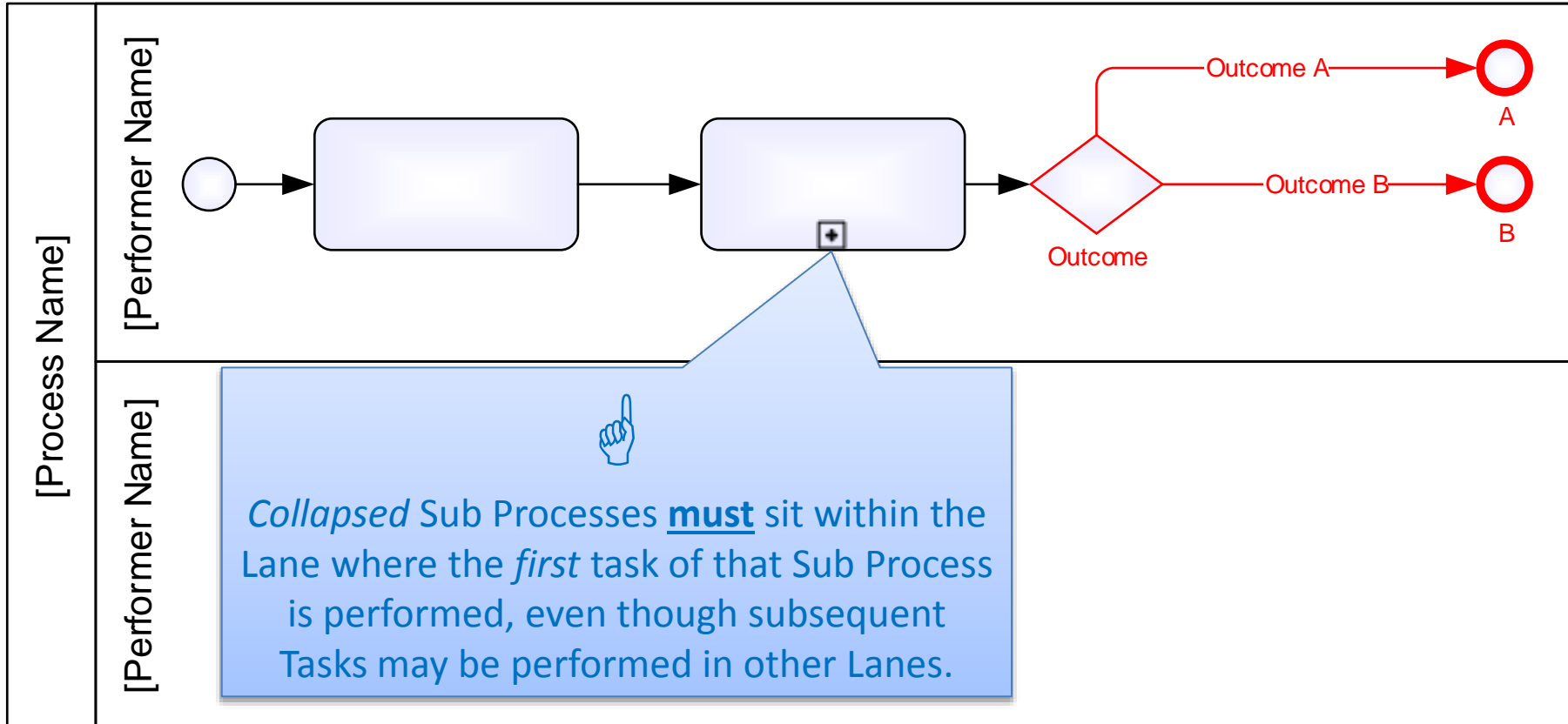
Sub Processes also allow you to group activities logically for the sake of tidiness.

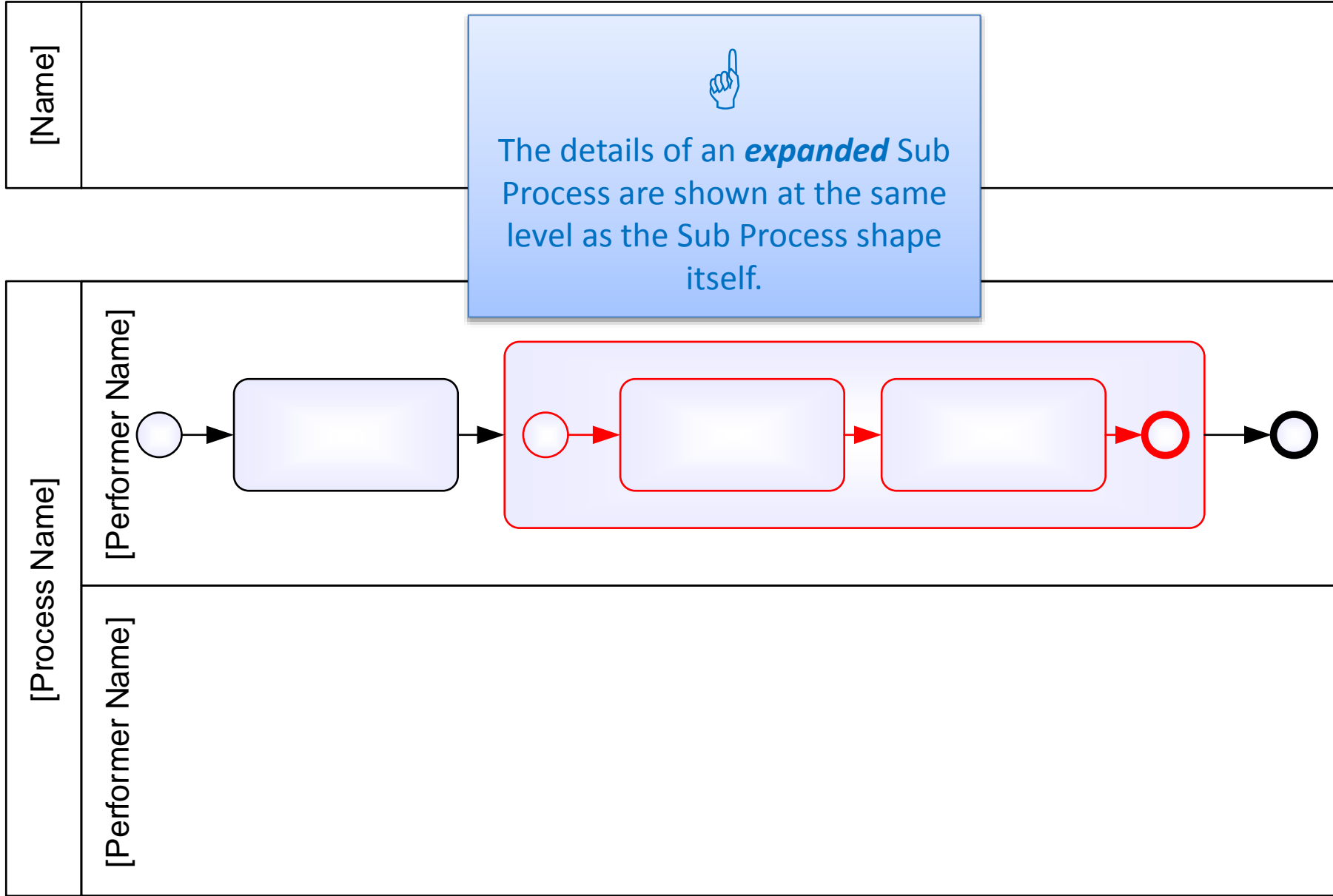




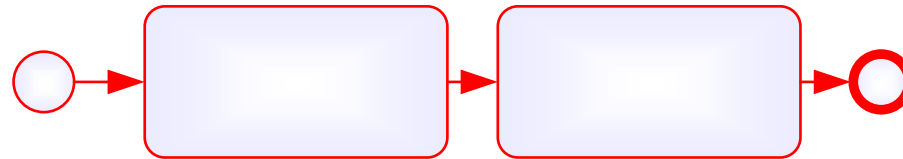
The details of a ***collapsed***  
Sub Process are shown on  
another (child) canvas.

[Name]	
--------	--







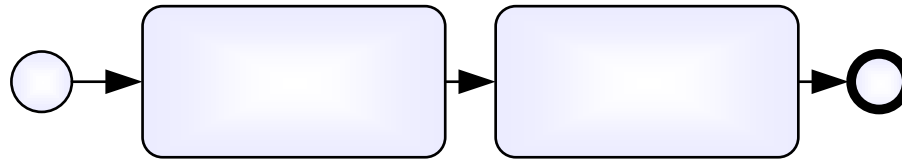


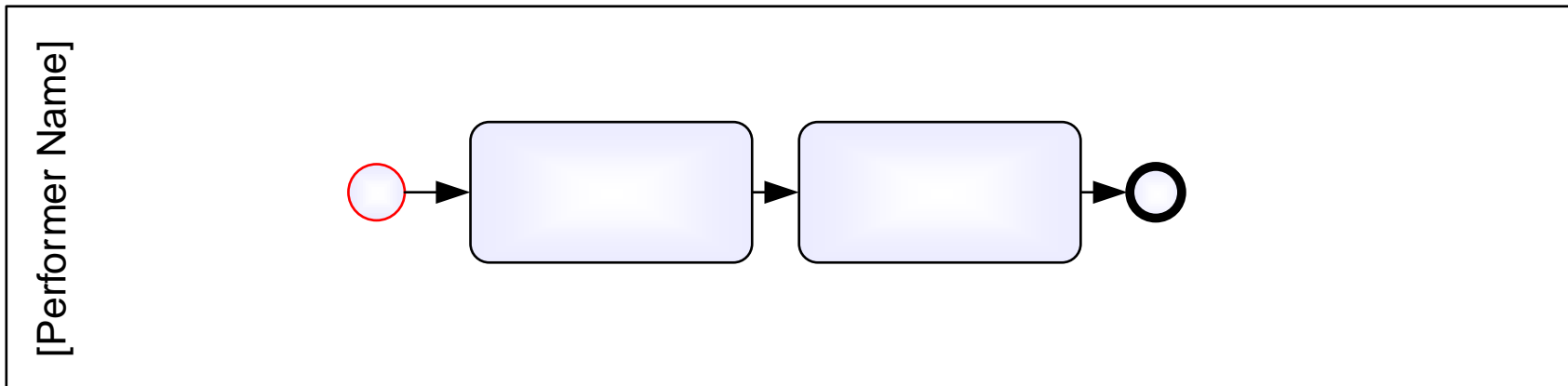
A Sub Process inherits its Pool from the parent, so you are only required to show the Pool at the top level of your model.



However, you should  
show any relevant Lanes  
at the Sub Process level.

[Performer Name]



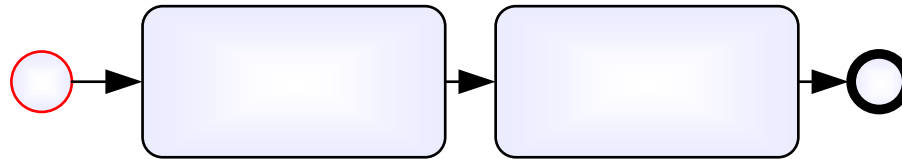


Sub Processes must have one  
*un-labelled* Start Event of the  
type “None”.



This is because the trigger for the Sub Process is already known: it is the ***previous Activity*** at the parent level.

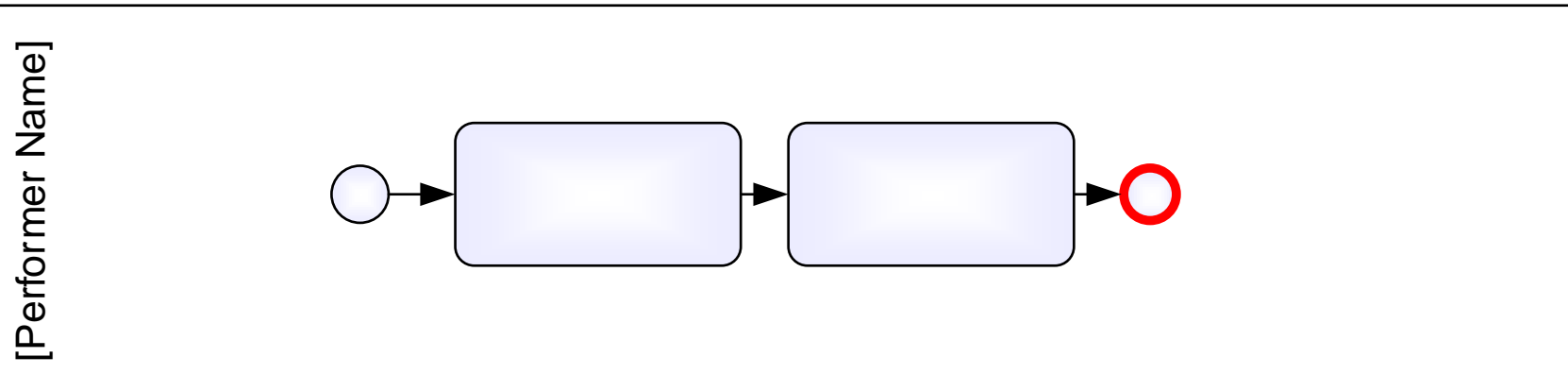
[Performer Name]



Sub Processes must have one *un-labelled* Start Event of the type “None”.



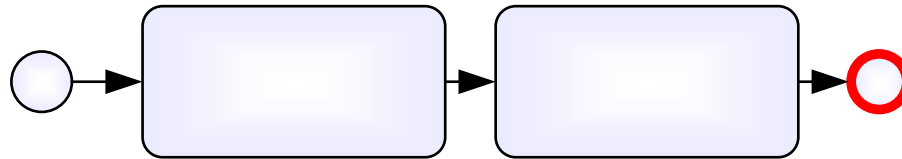
Sub Processes must *at least one* End Event.



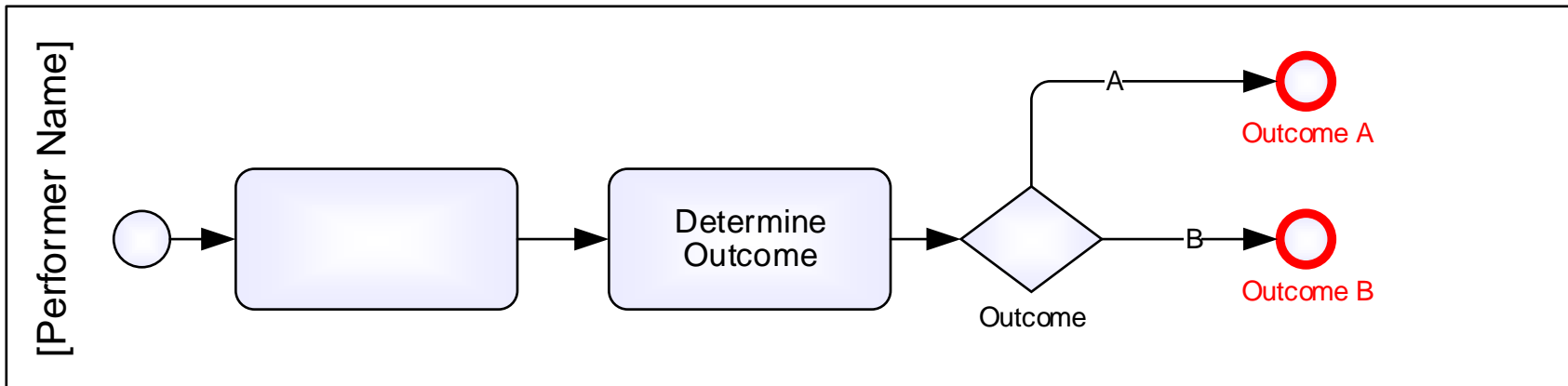


Sub Processes must *at least one* End Event.

[Performer Name]



There is no need to label  
a Sub Process End Event  
*if there is only one.*



You must label Sub Process  
End Events if there are  
more than one.

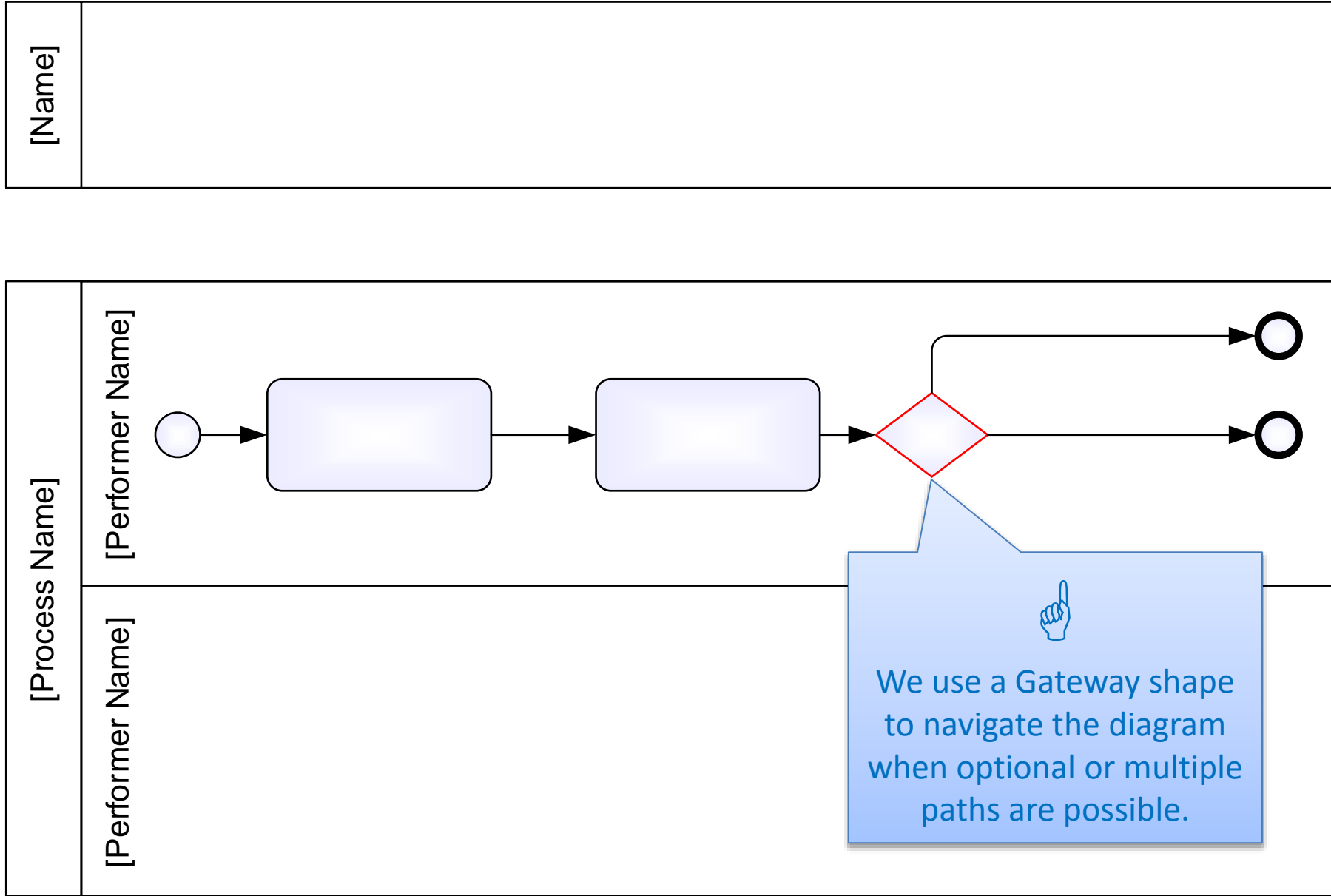
# Gateways



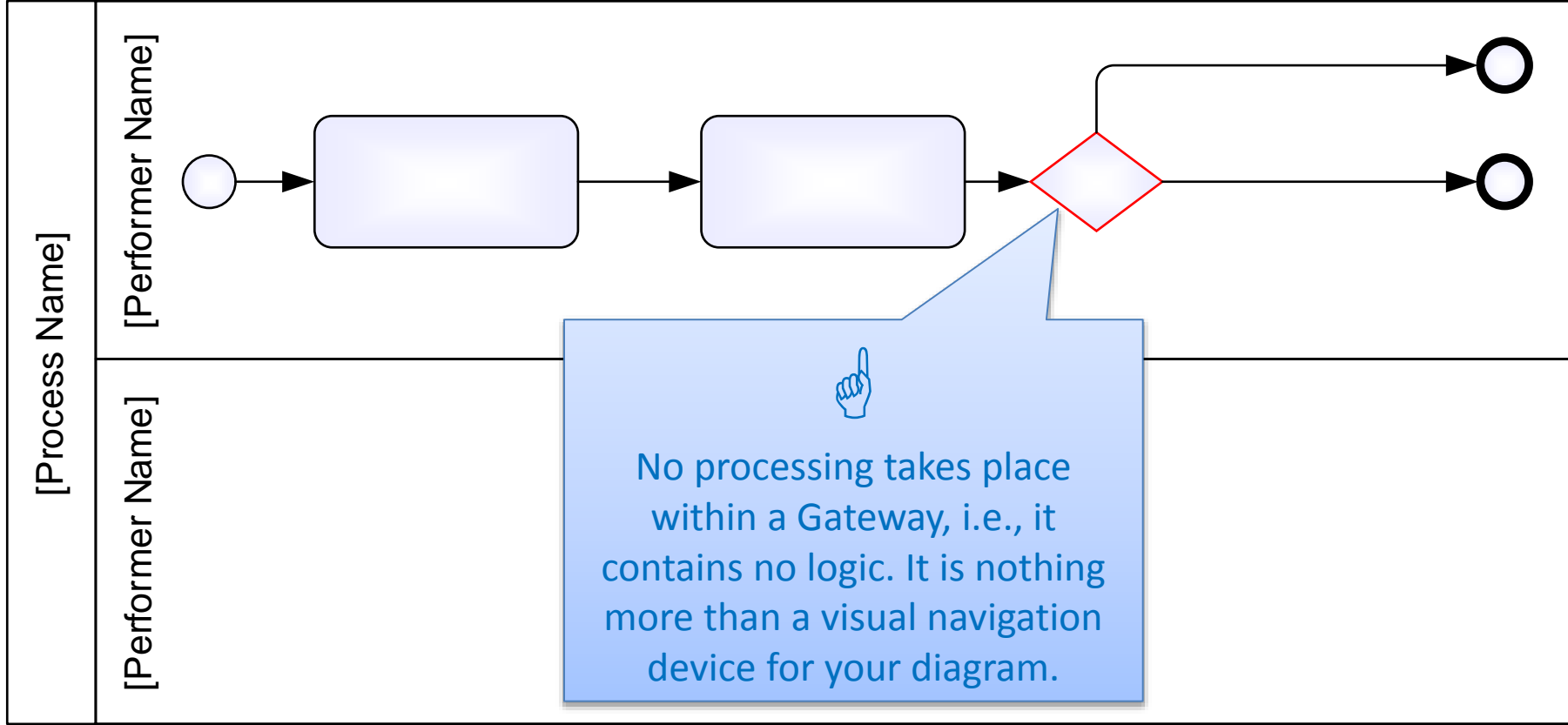
# Gateways

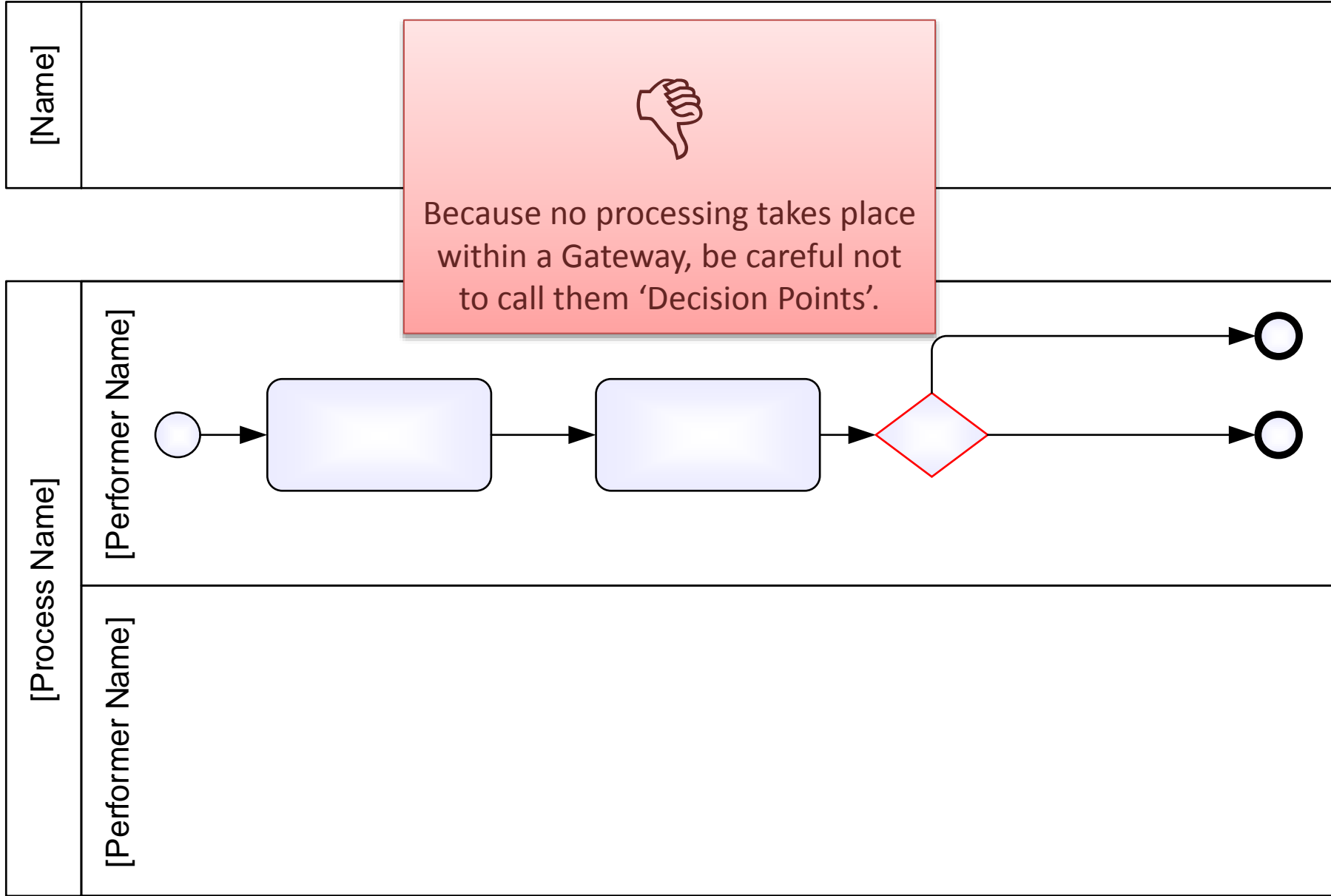


**Gateways** are used to control how **Sequence Flows** interact as they converge and diverge within a **Process**.  
(section 8.3.13)

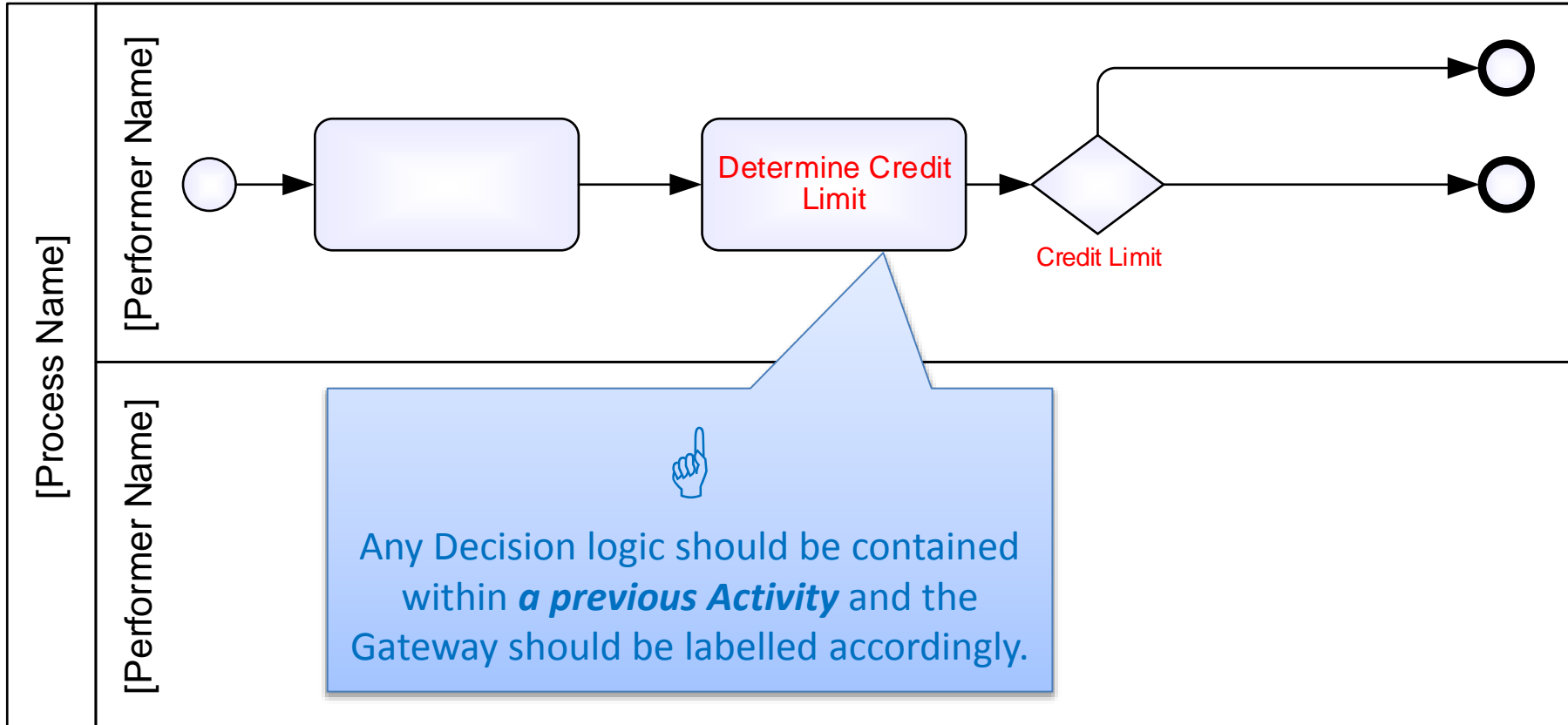


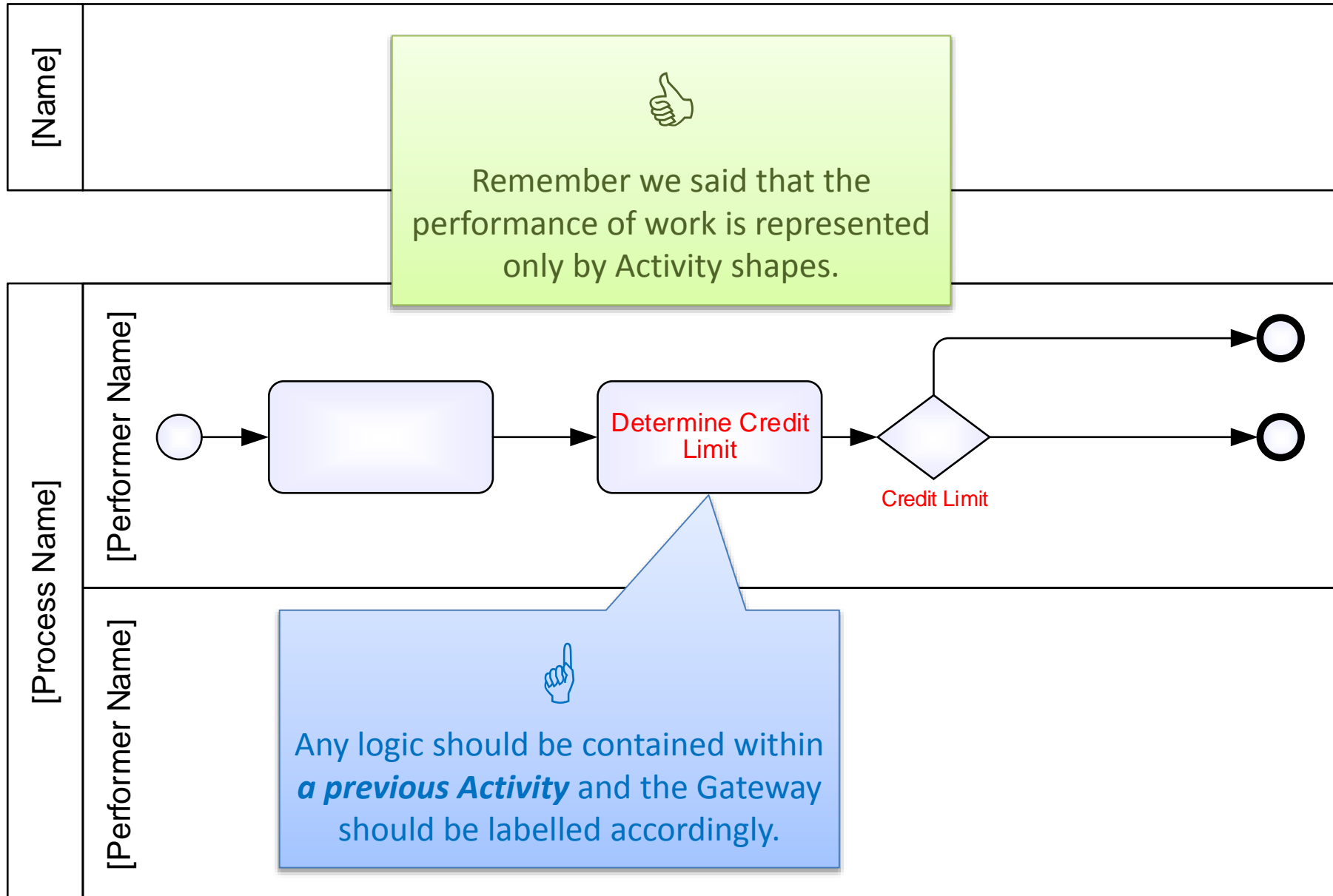
[Name]	
--------	--



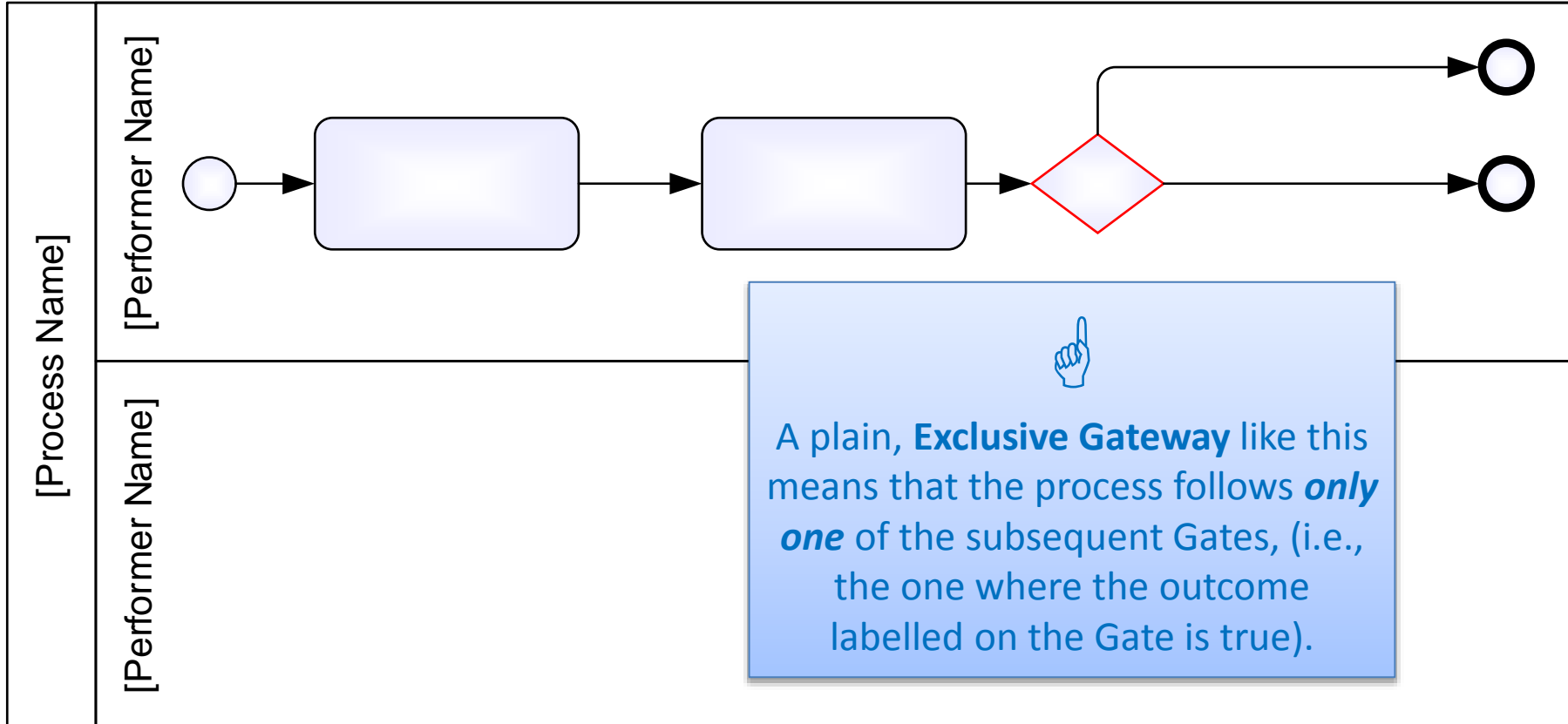


[Name]	
--------	--

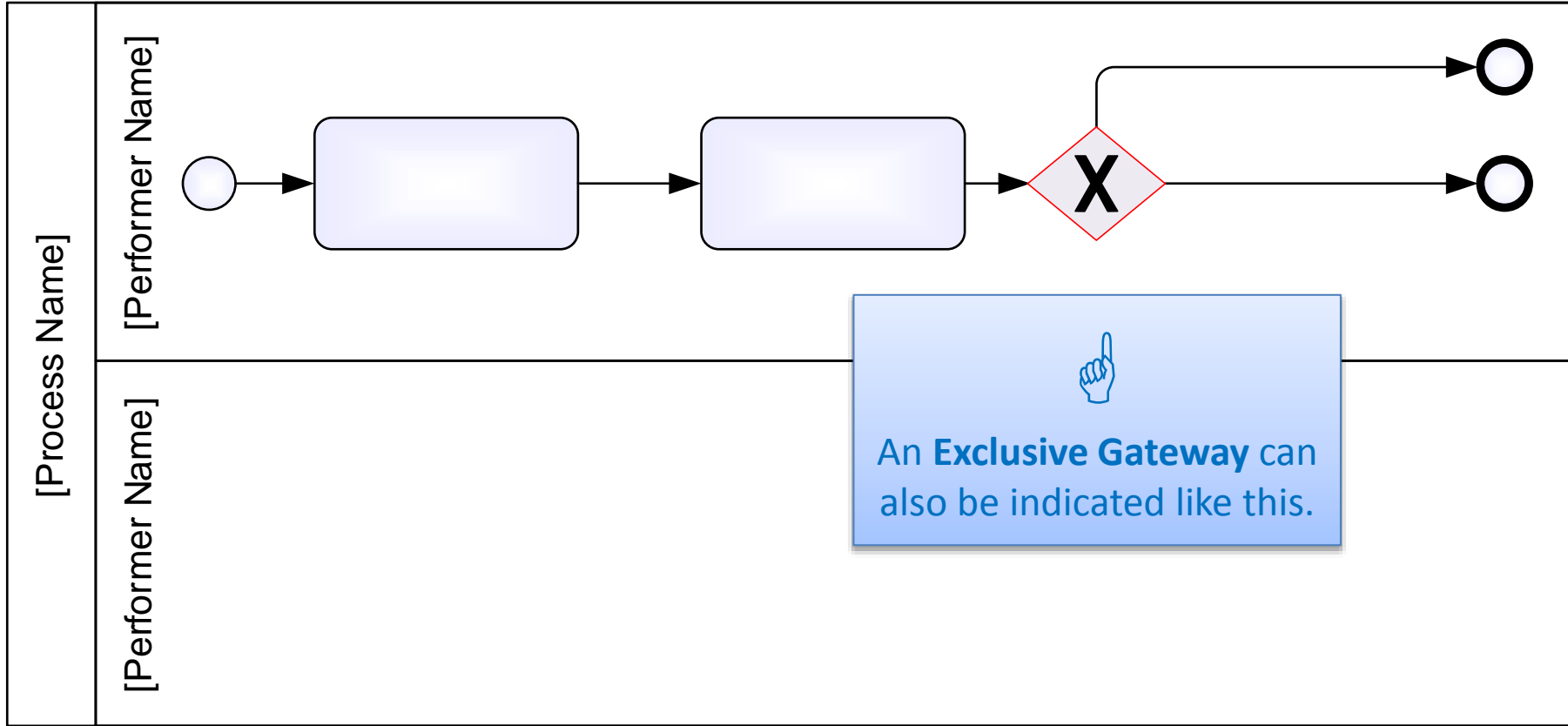




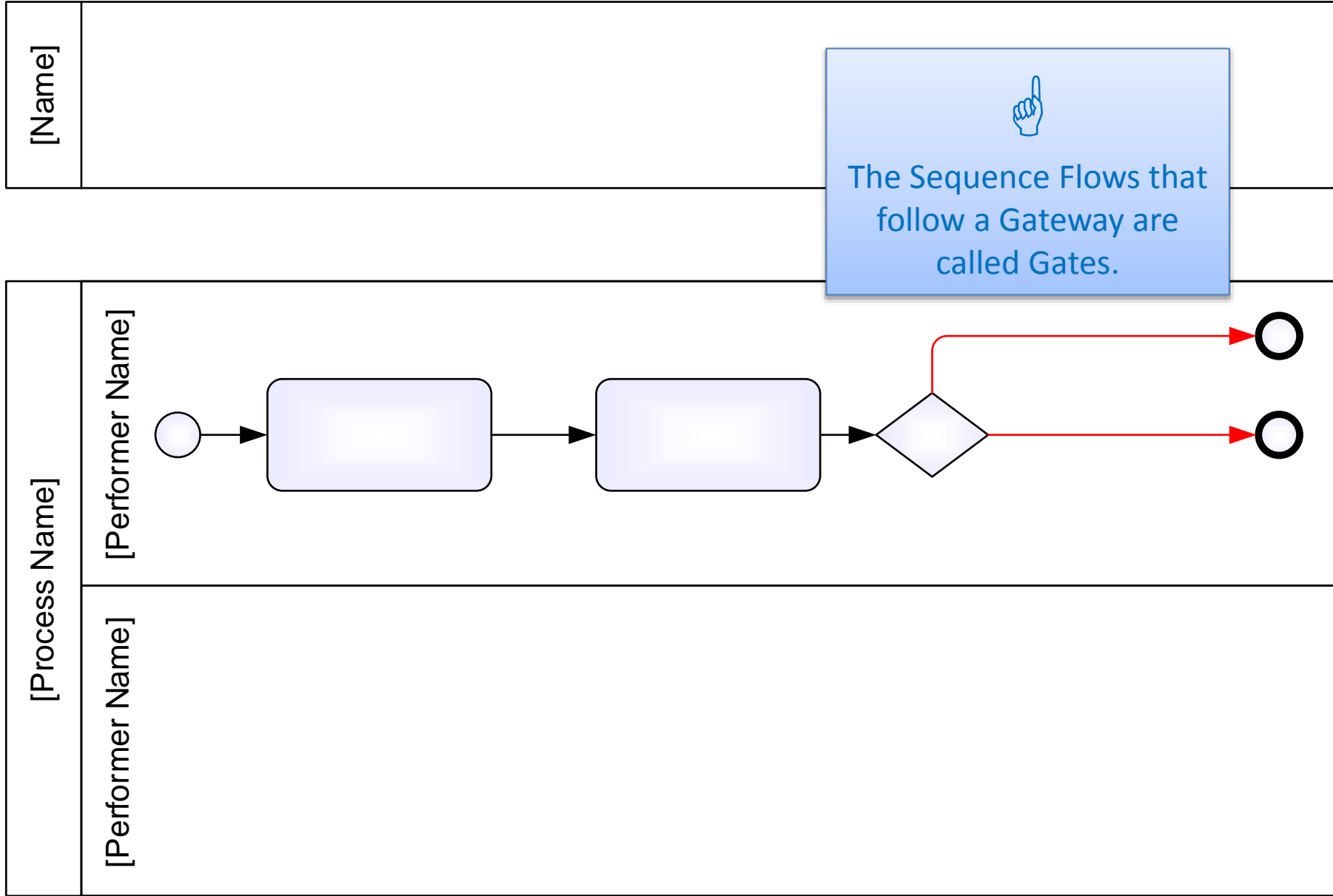
[Name]	
--------	--

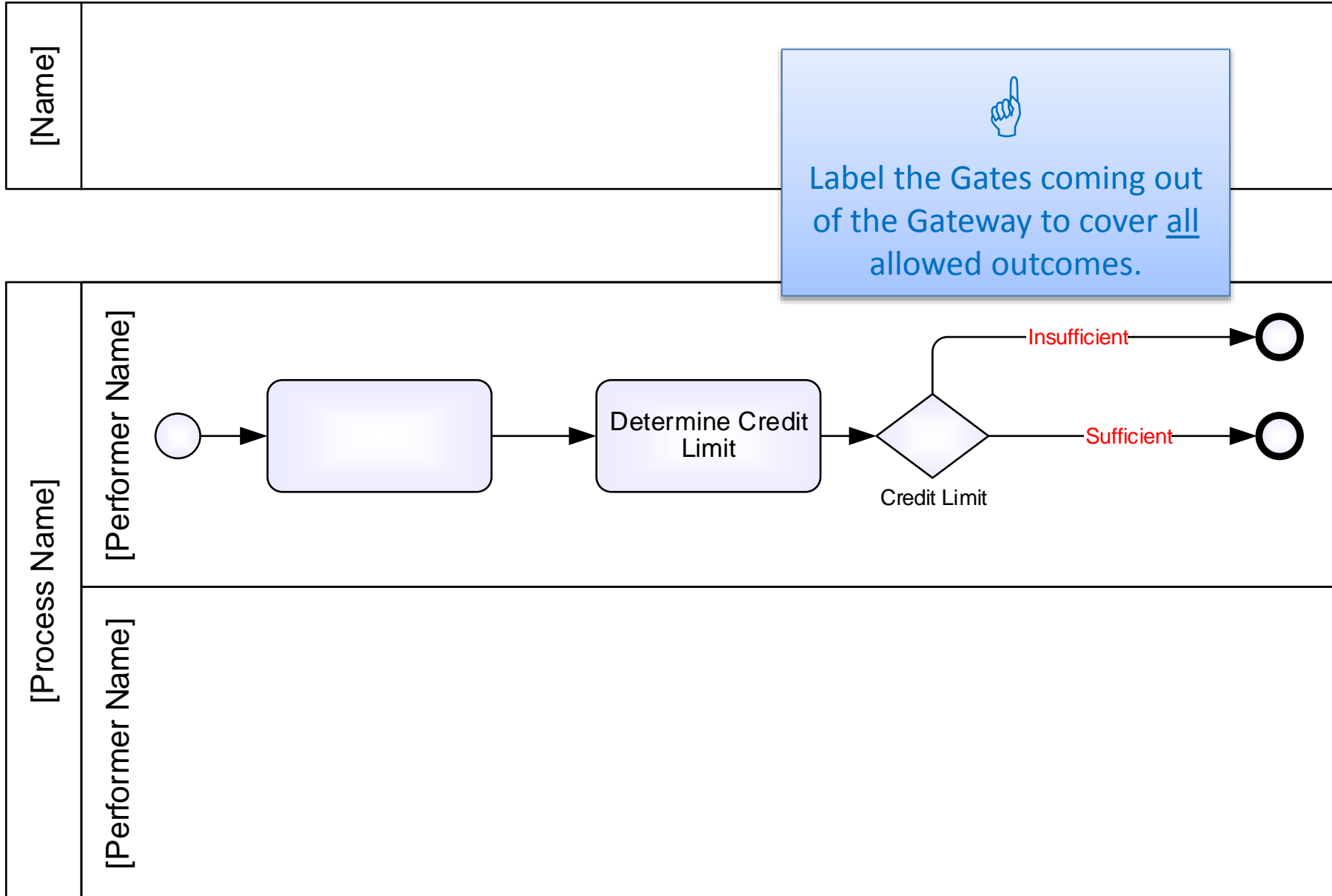


[Name]	
--------	--

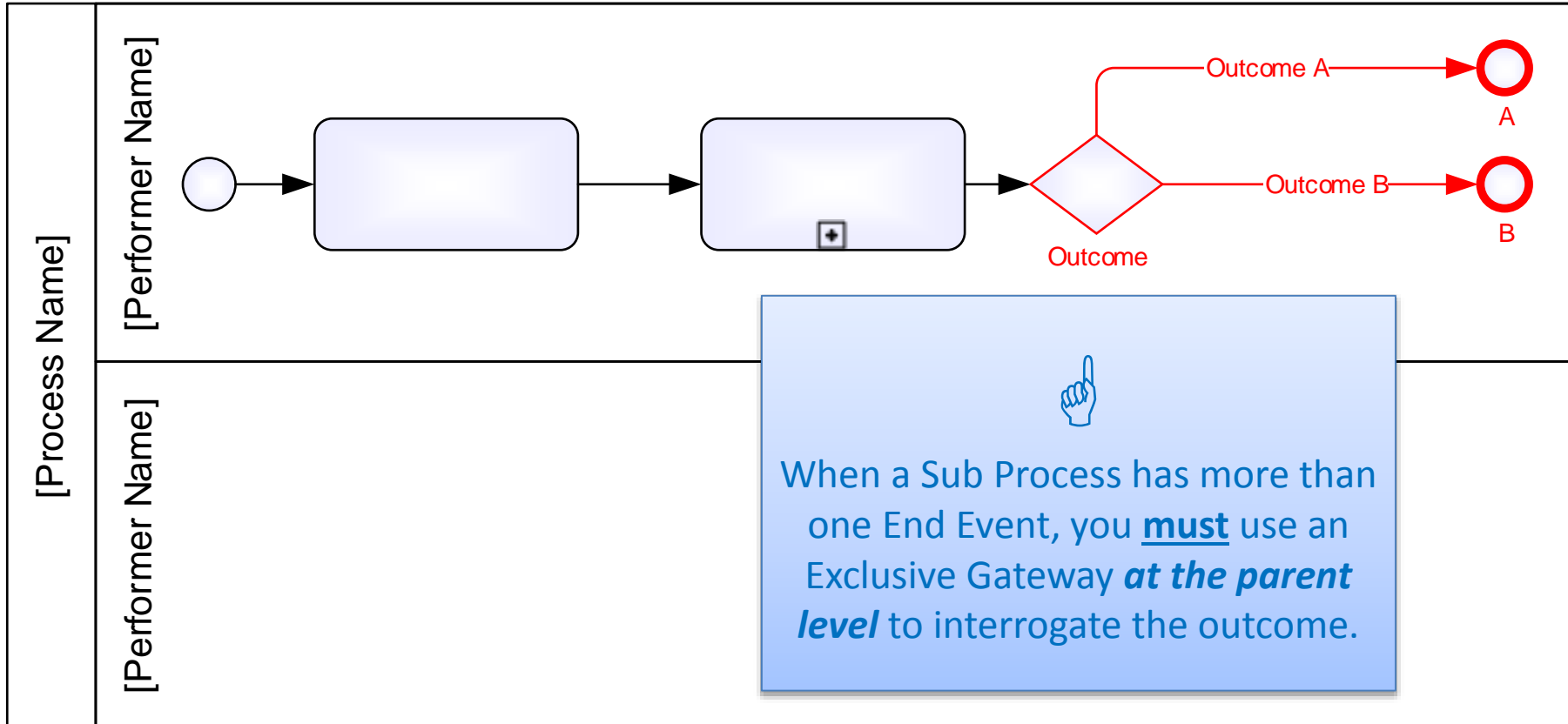




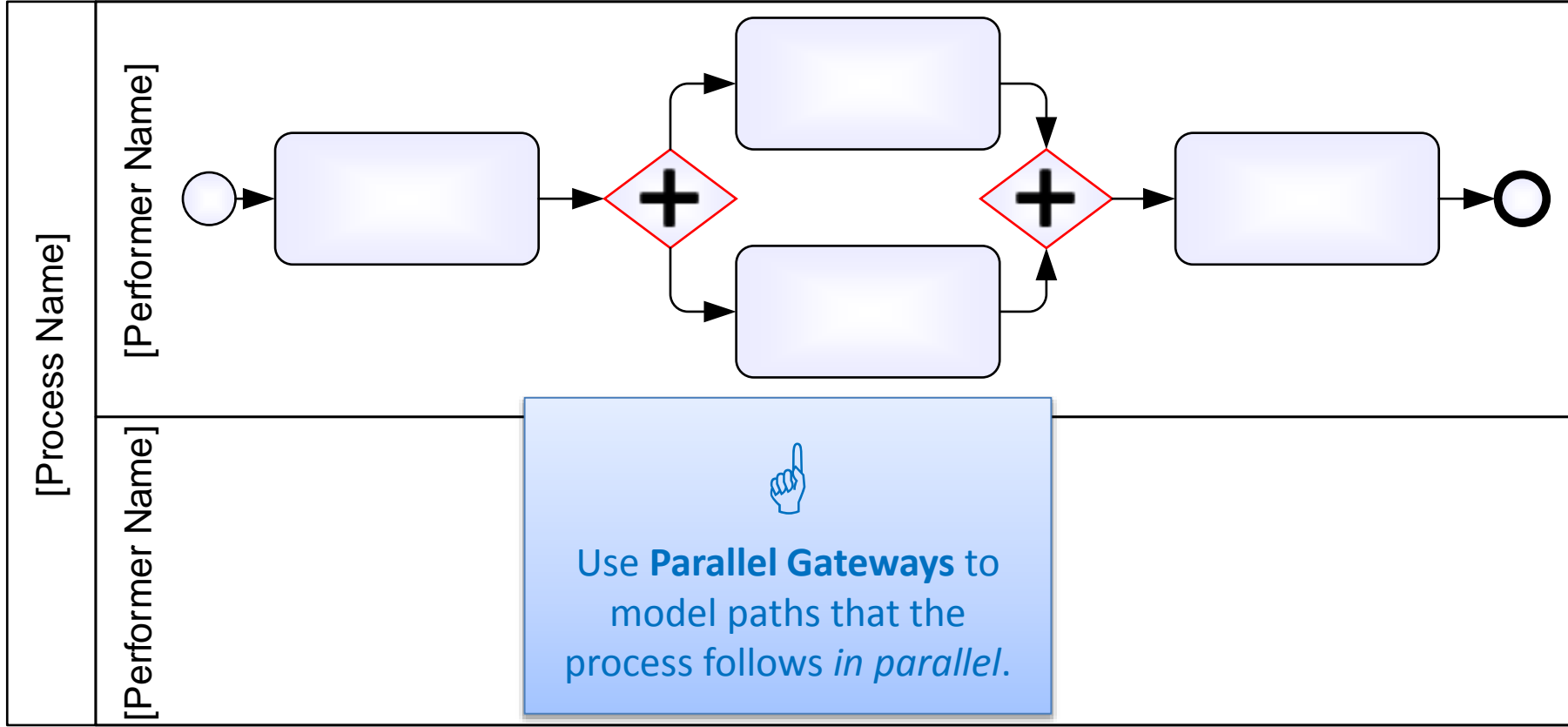


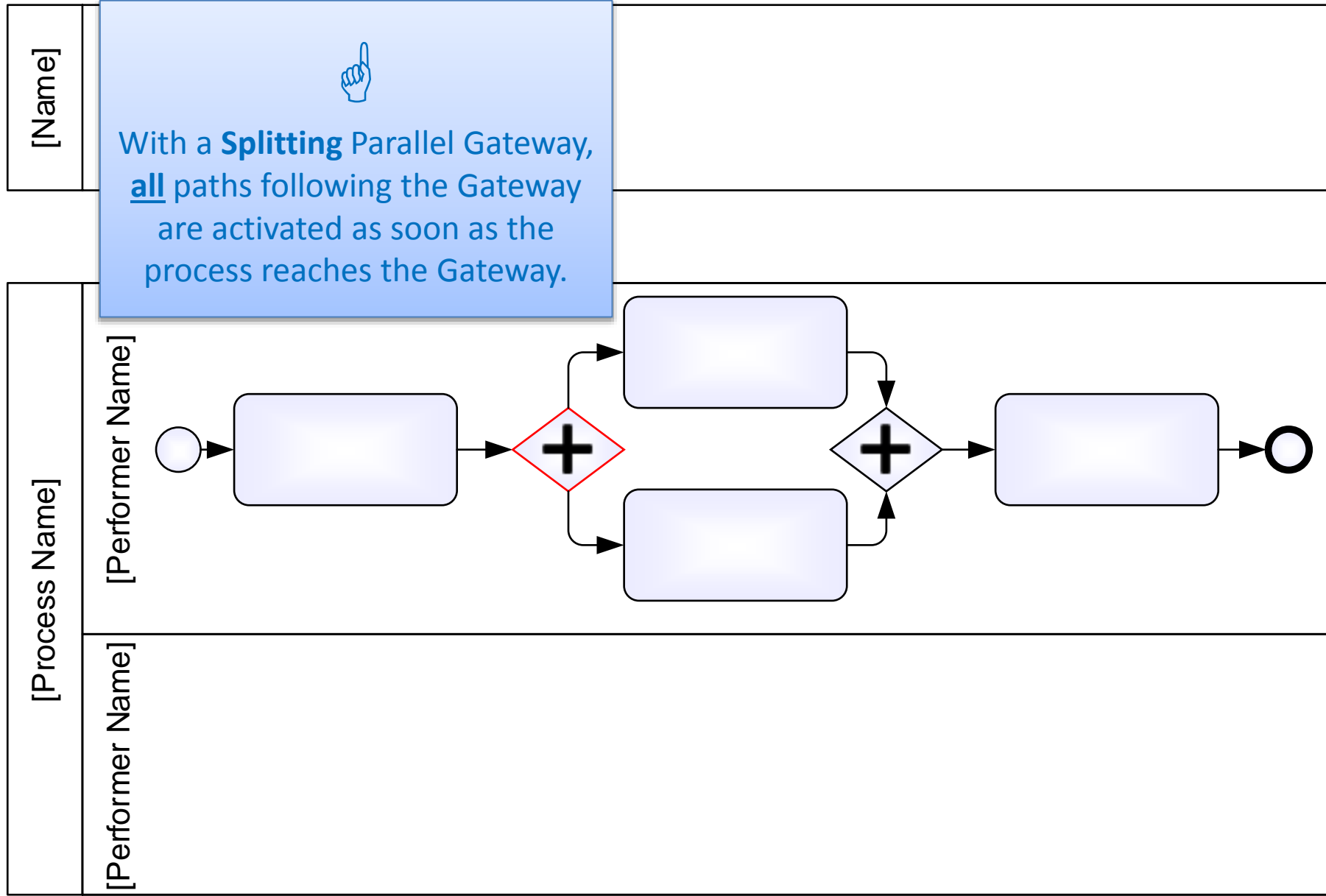


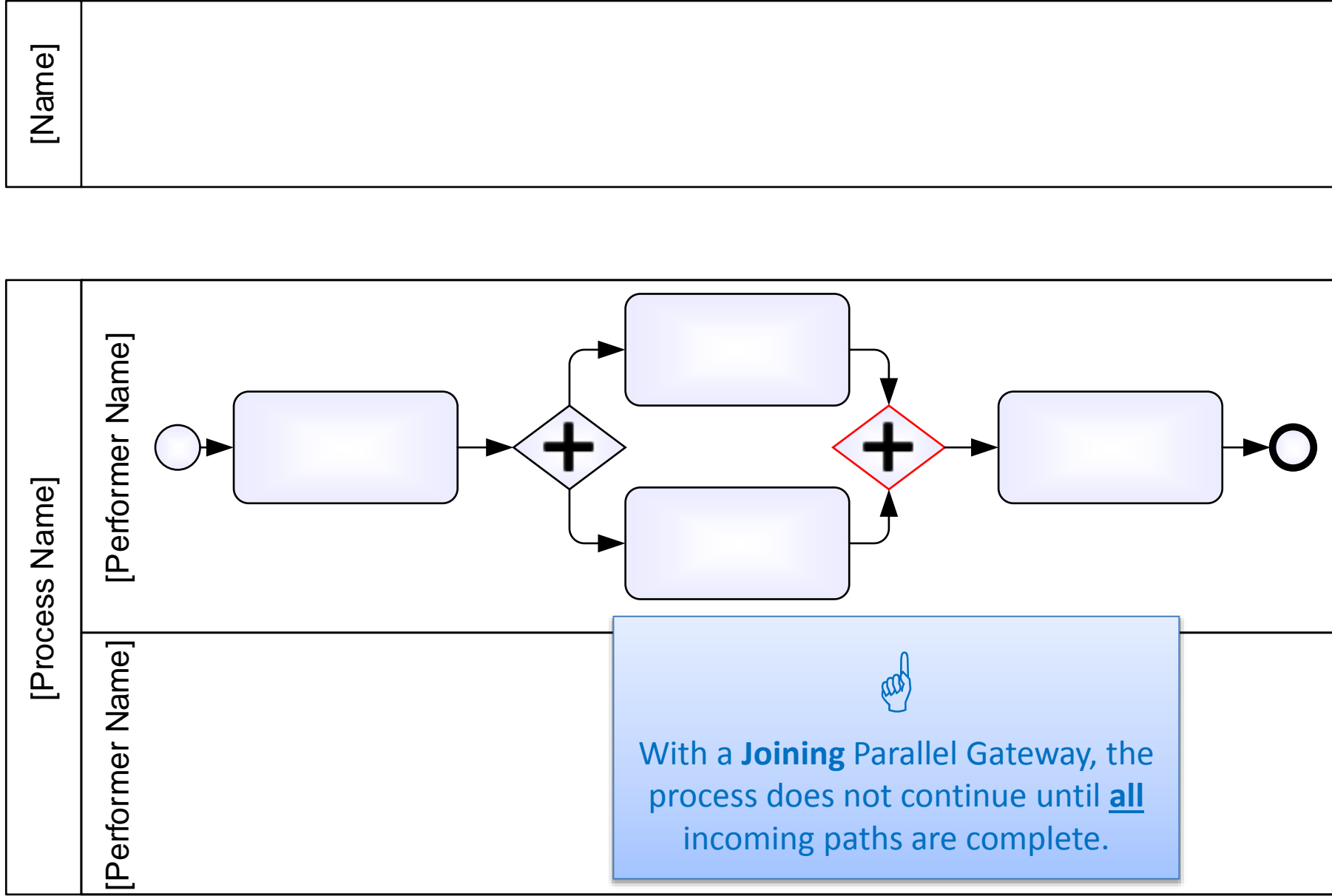
[Name]	
--------	--



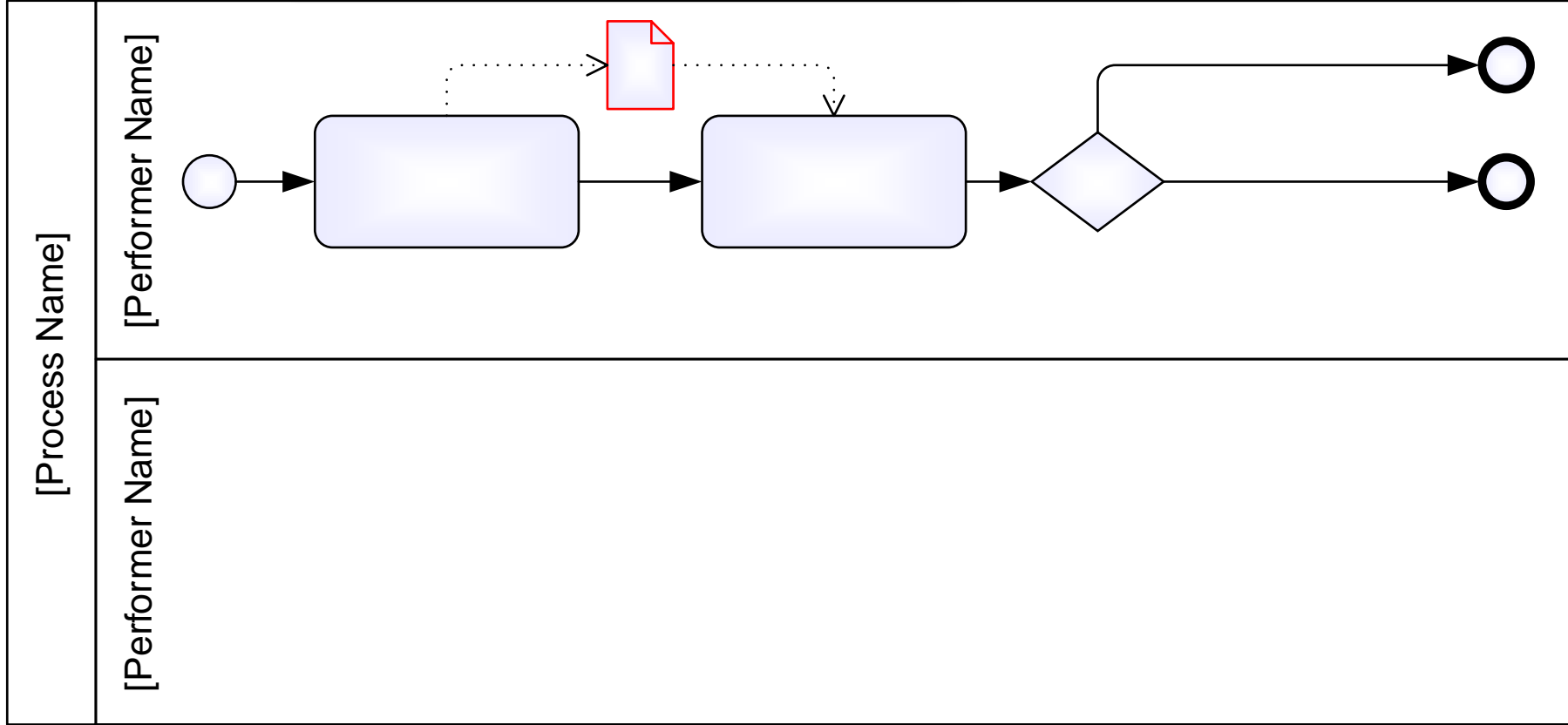
[Name]	
--------	--





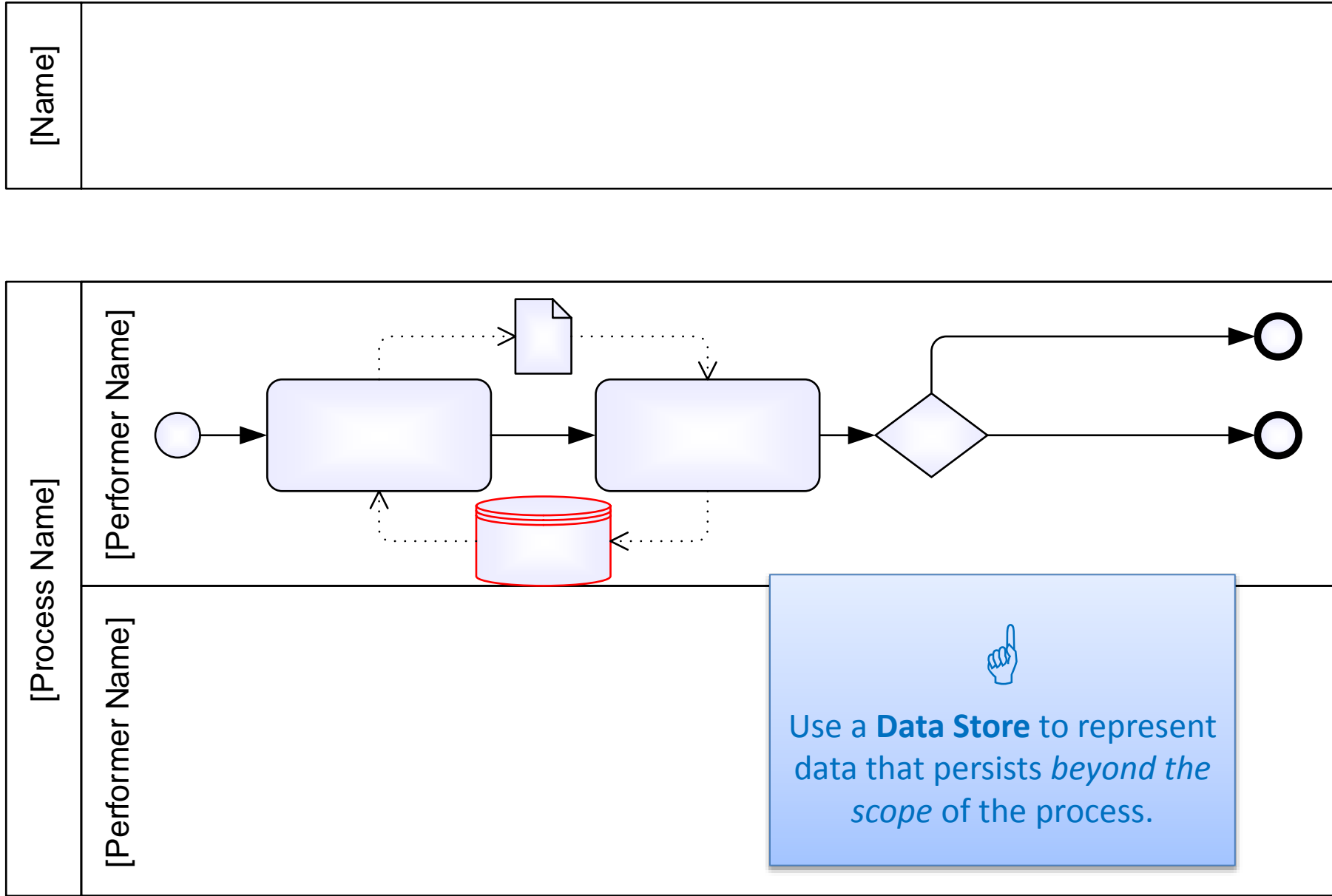


# Data Shapes



[Name]
<p>Use a <b>Data Object</b> to represent <u>transient</u> data that is passed <i>within</i> a process but which does not need to exist outside the process.</p>

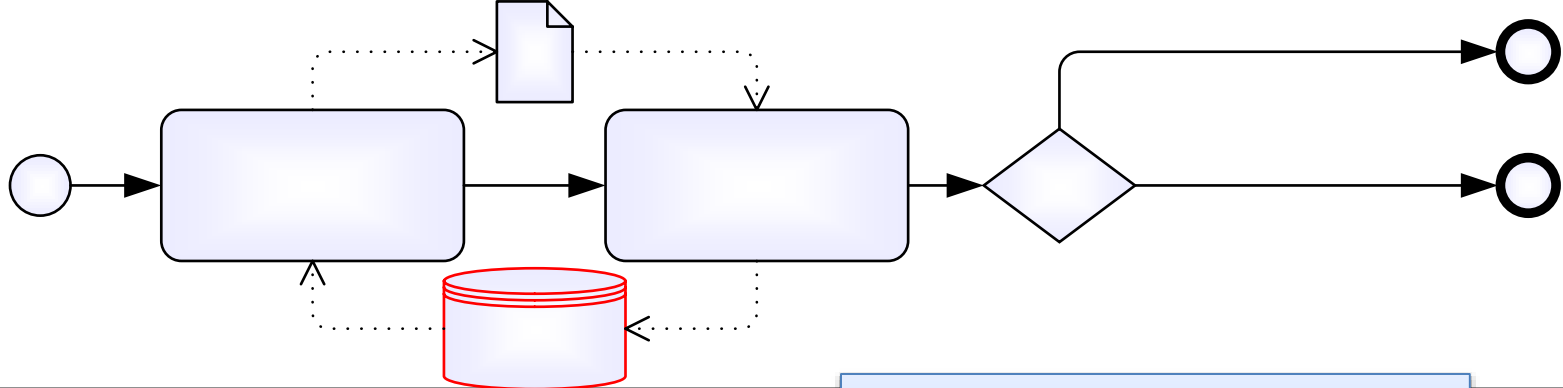


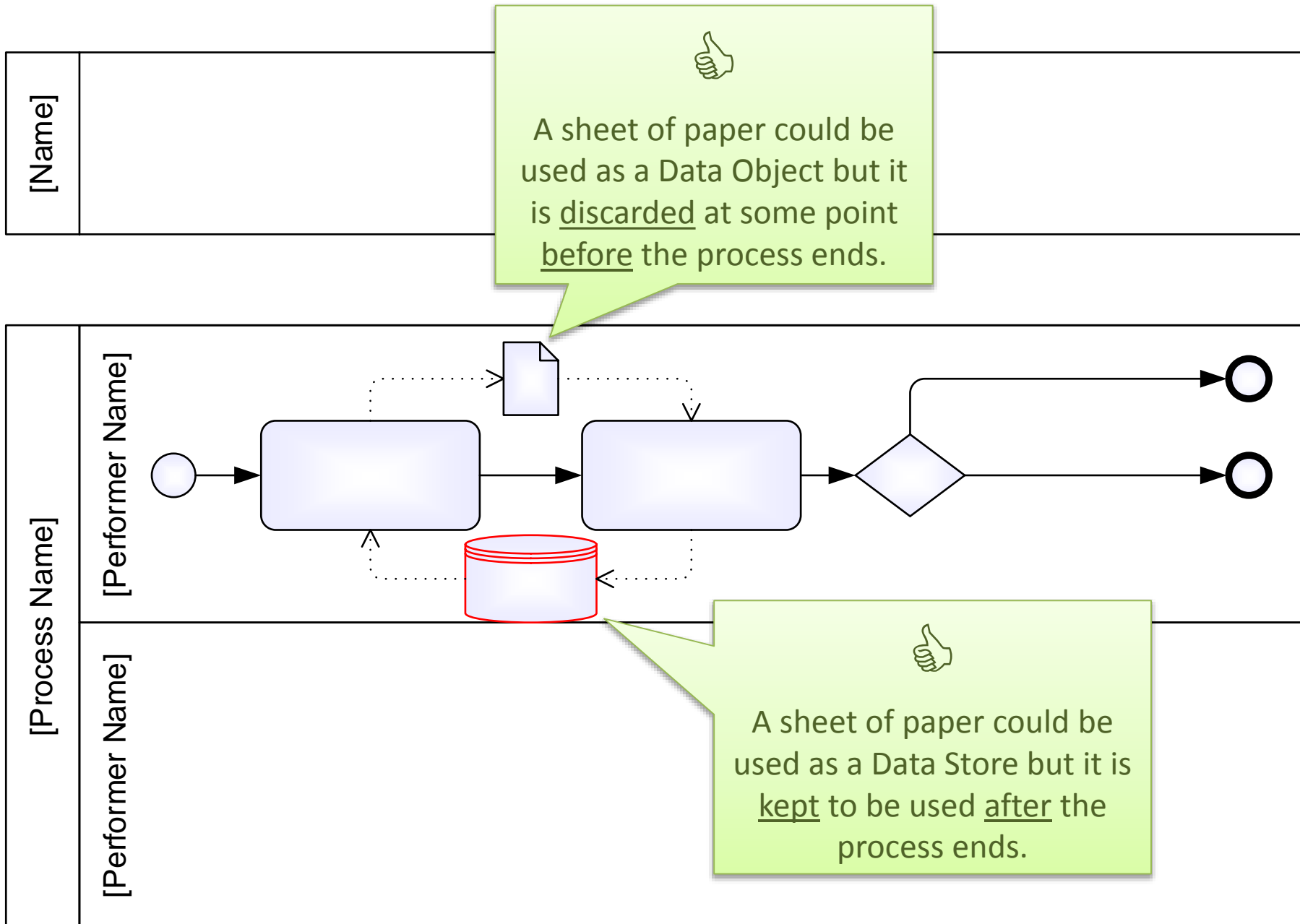


[Name]	
--------	--

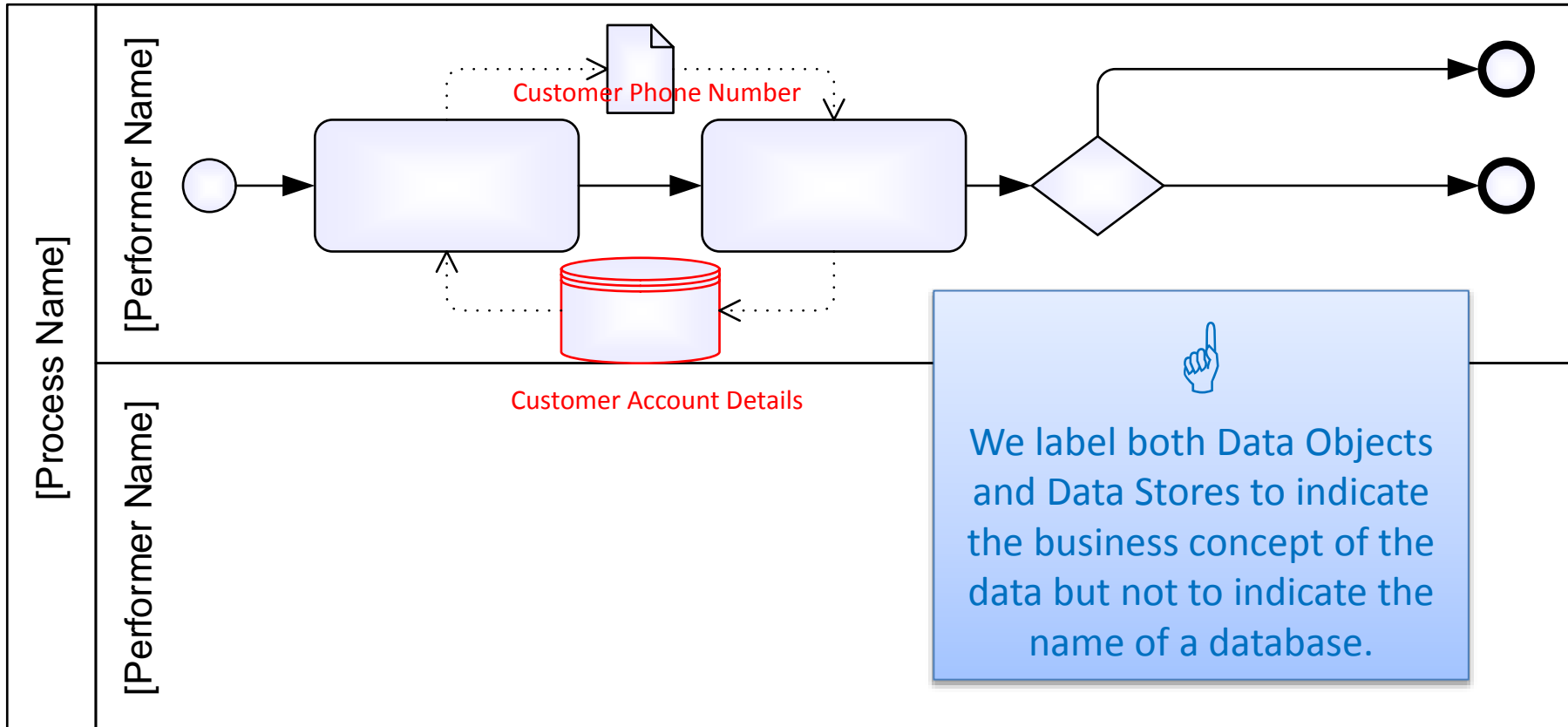


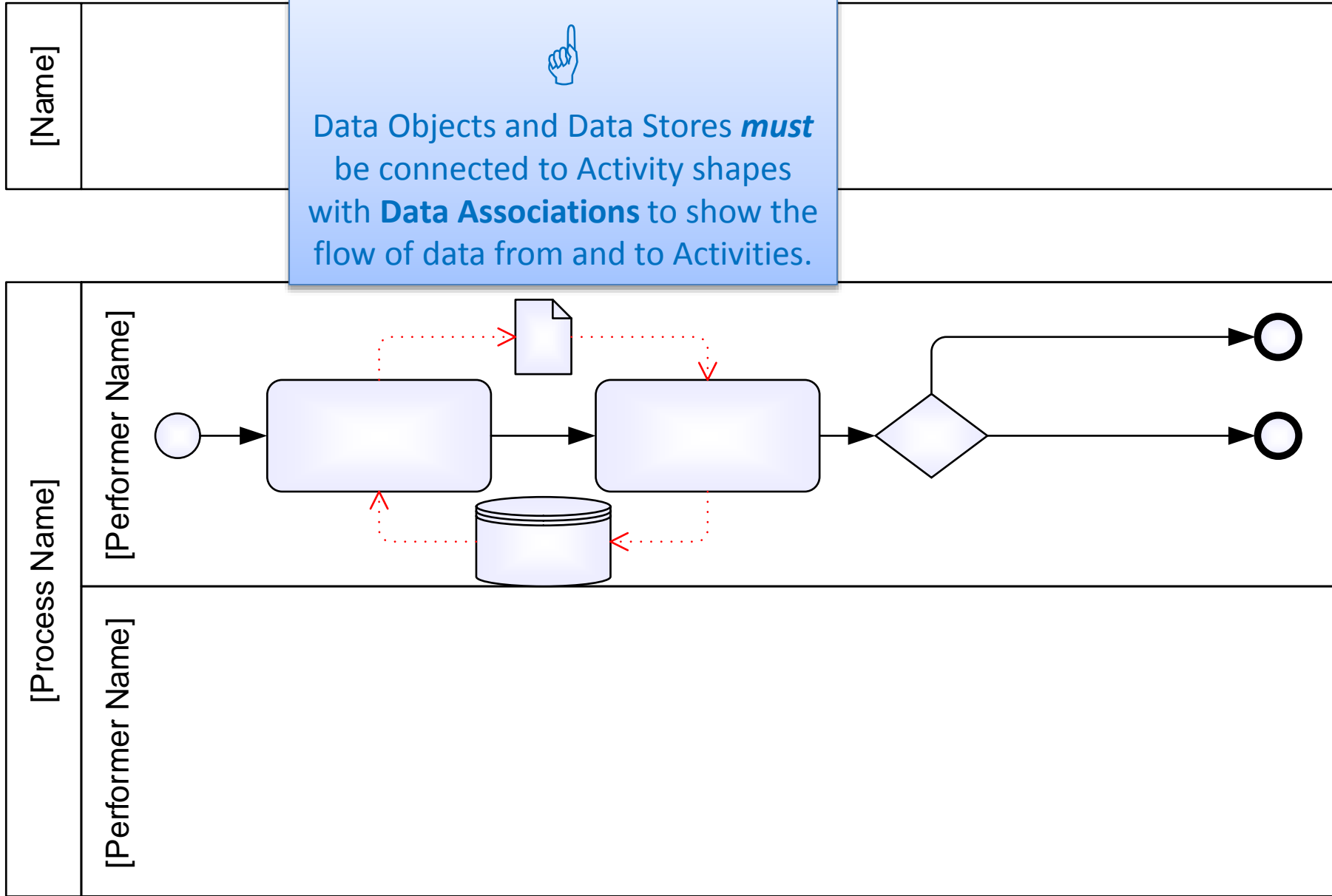
A Data Store is ***not necessarily a database!*** It is simply a logical representation of persisted data.

[Process Name]	<p data-bbox="504 491 555 822">[Performer Name]</p> 
	<p data-bbox="504 902 555 1233">[Performer Name]</p> <div data-bbox="1406 848 2035 1243"> <p>Use a Data Store to represent data that persists <b><i>beyond the scope</i></b> of the process.</p> </div>

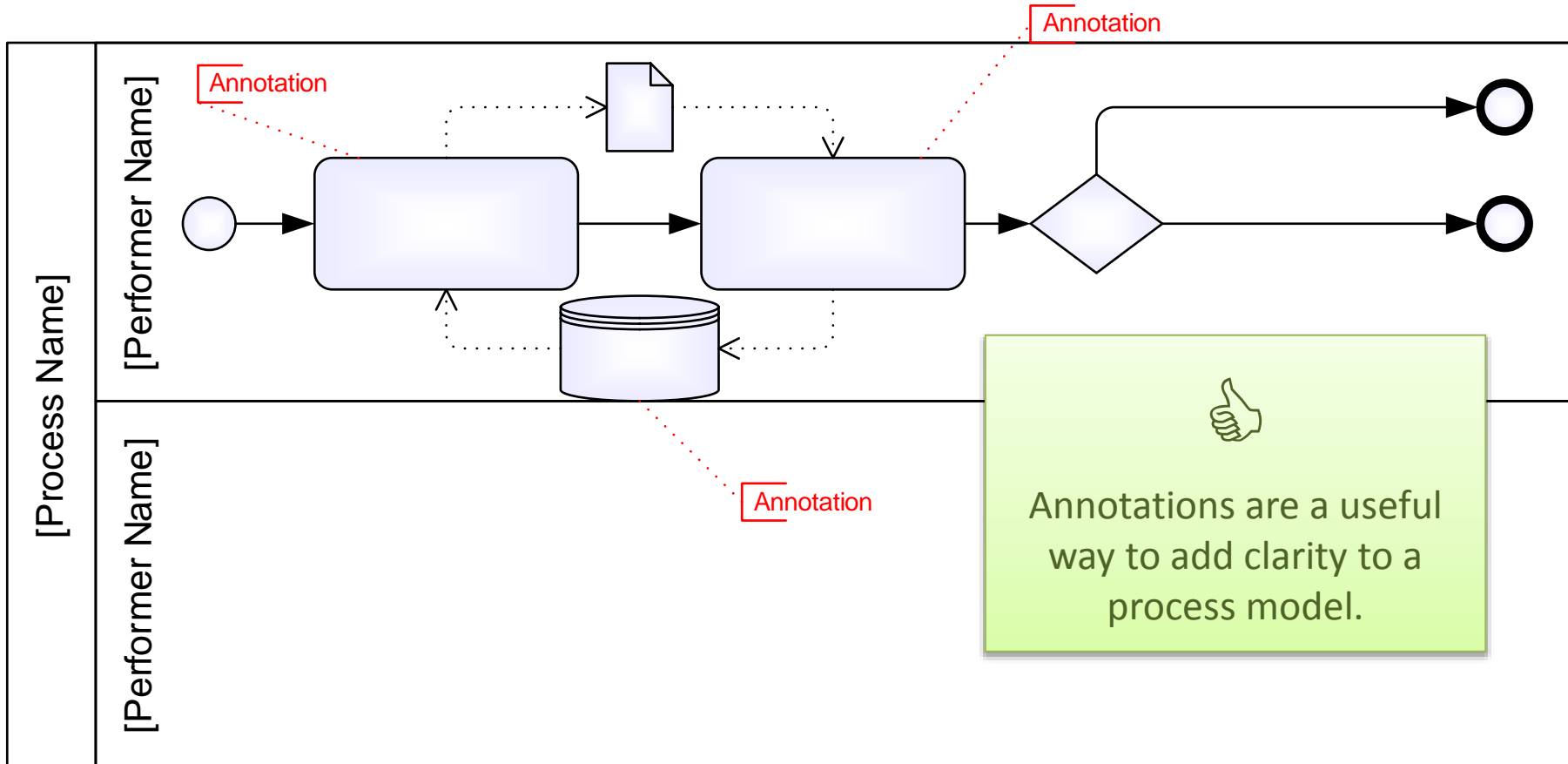


[Name]	
--------	--

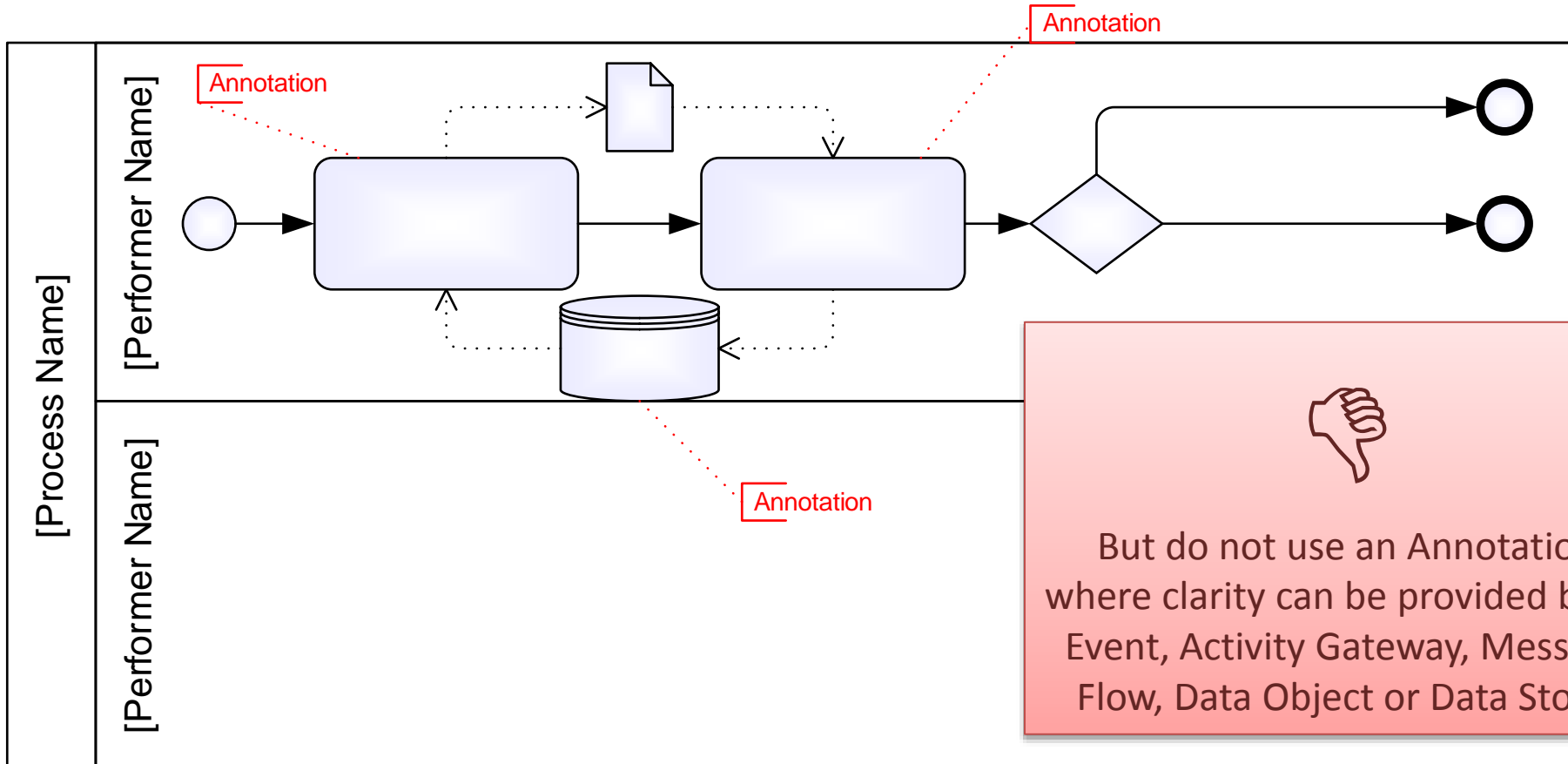




[Name]	
--------	--



[Name]	
--------	--



But do not use an Annotation where clarity can be provided by an Event, Activity Gateway, Message Flow, Data Object or Data Store.

# Off-page Connectors



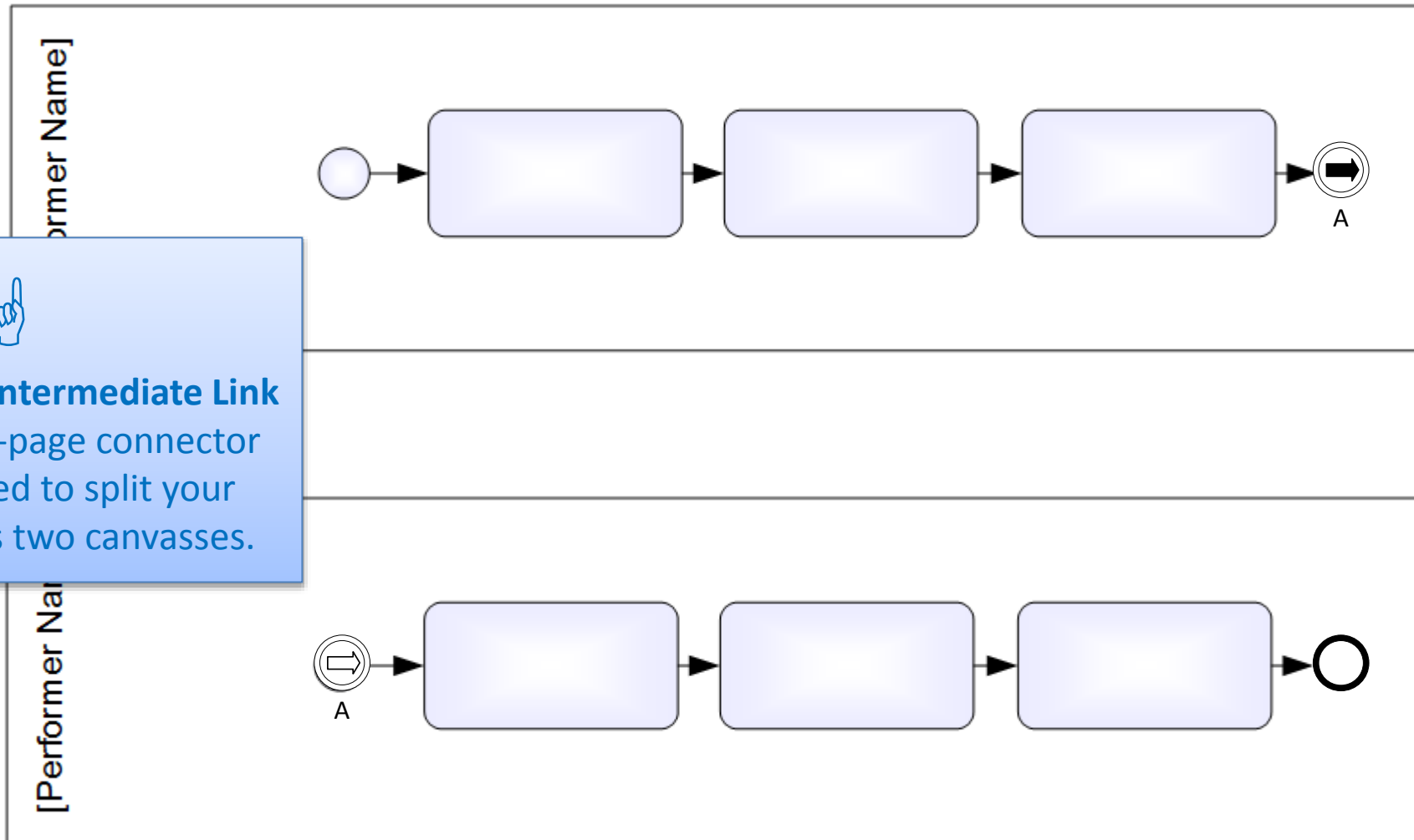
# Off-page Connectors

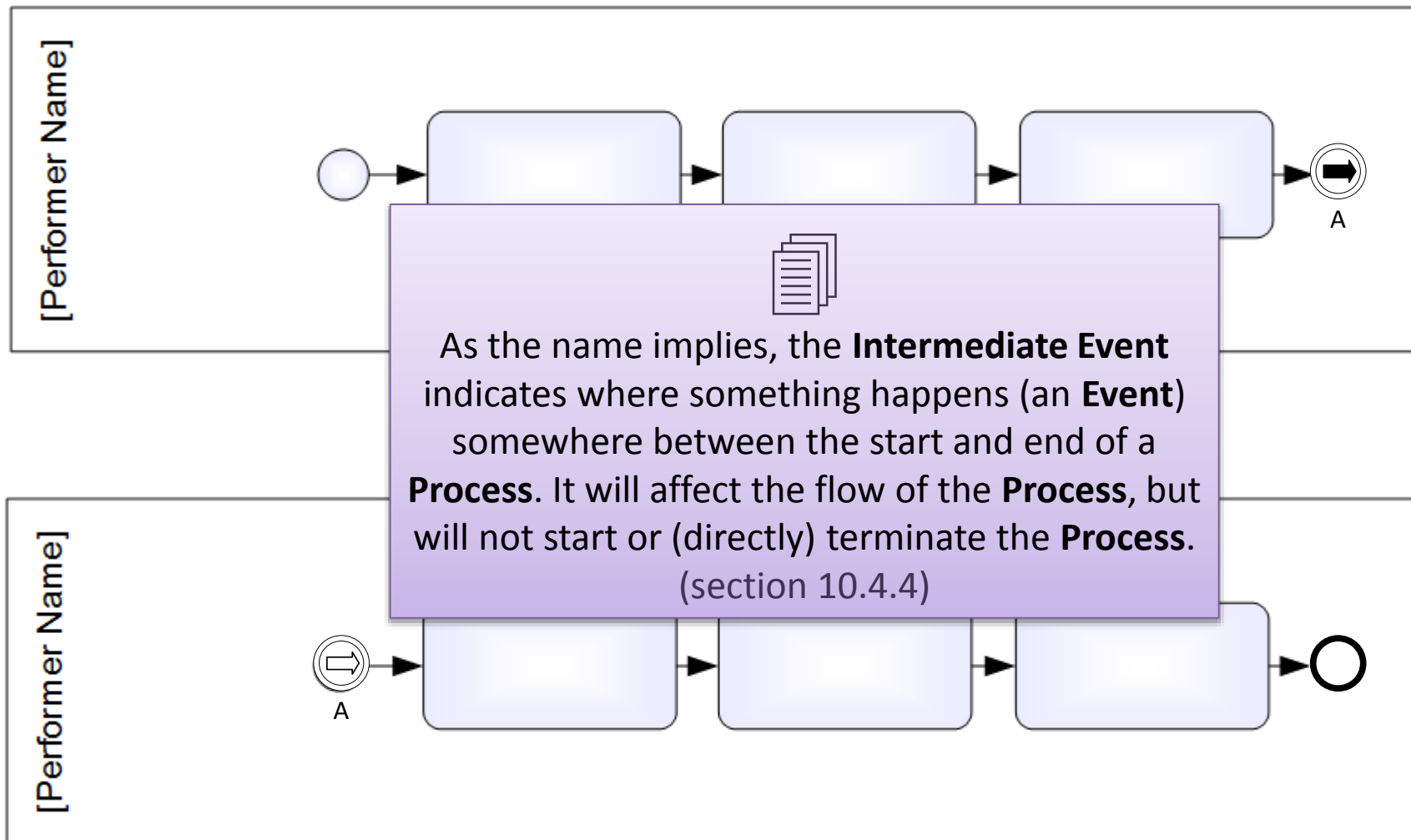


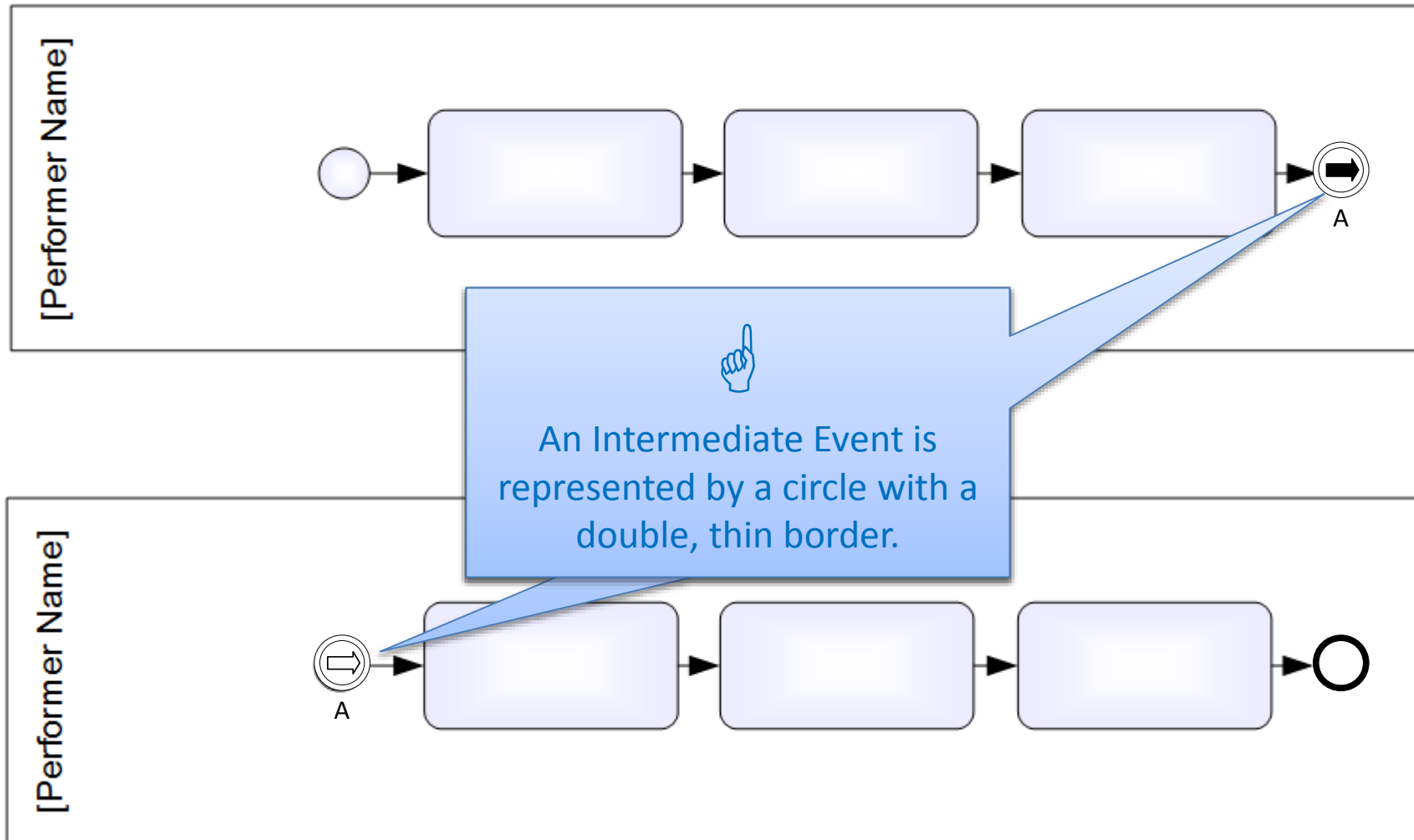
Generally used for printing, this object will show where a **Sequence Flow** leaves one page and then restarts on the next page.  
(table 7.2)

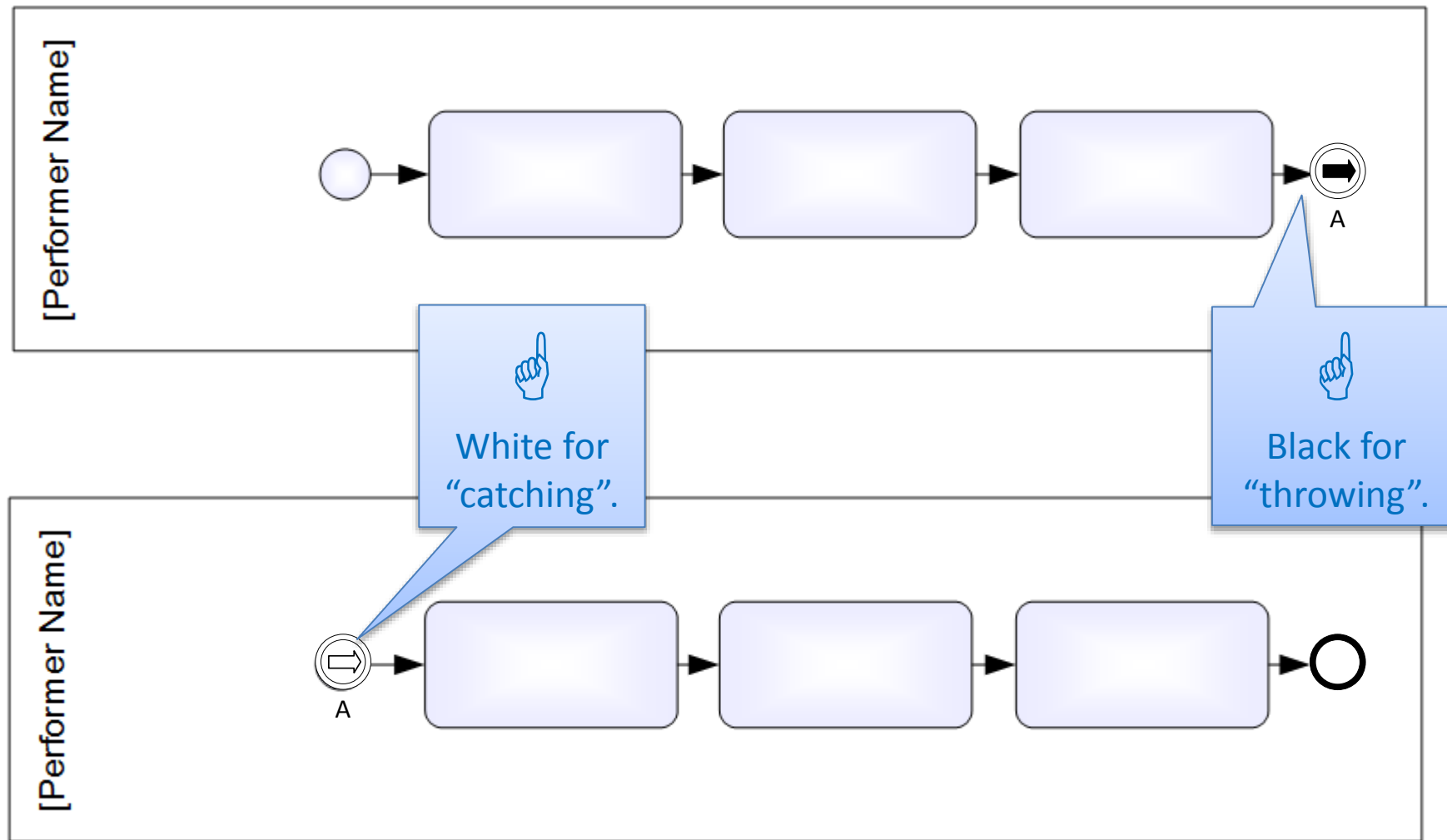


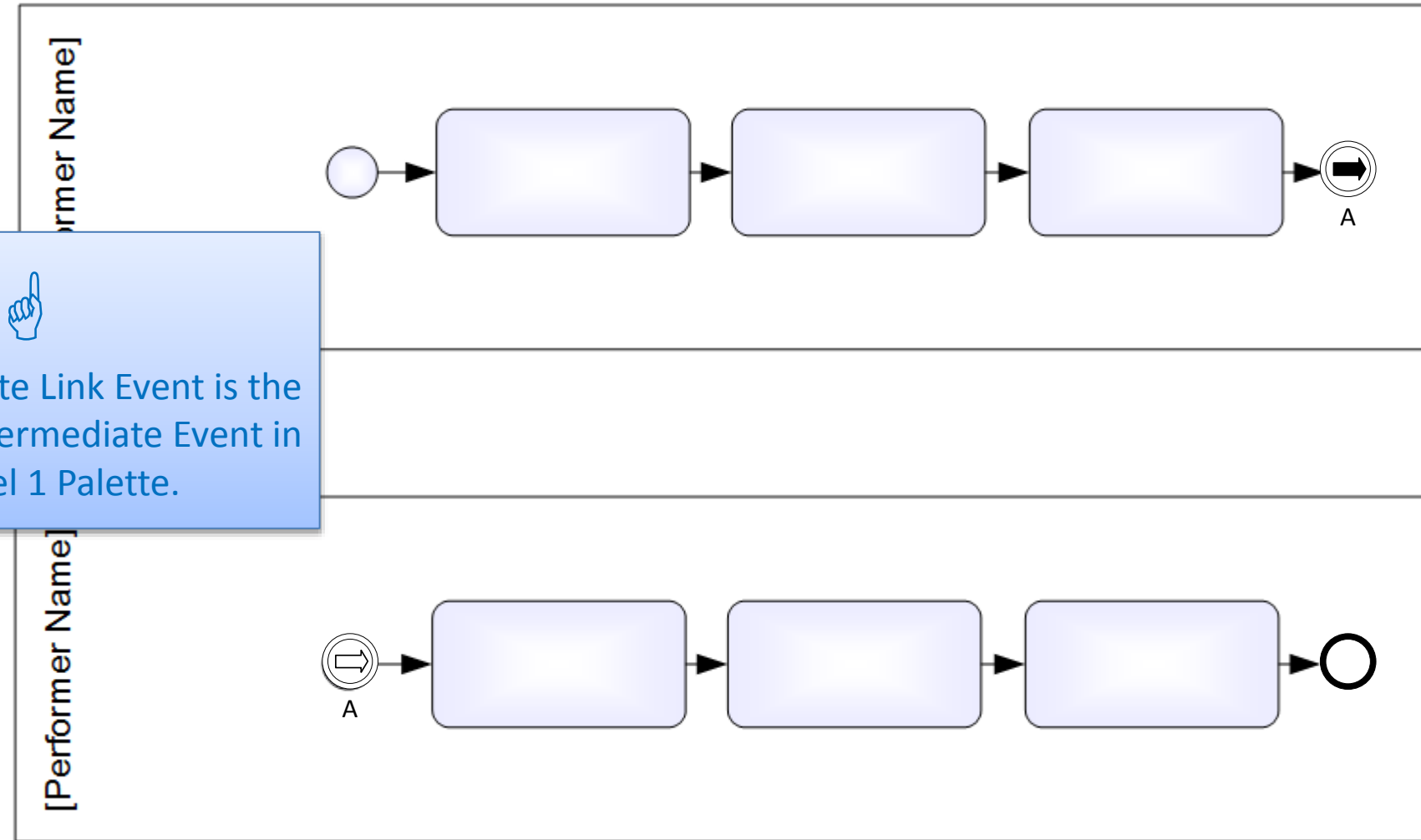
You can use an **Intermediate Link Event** as an off-page connector when you need to split your diagram across two canvasses.











[Former Name]



Use the Intermediate Link Event sparingly. Instead, try to use Sub Processes to keep each diagram of your model tidily on a single canvas.

[Performer Name]



**This has been a basic introduction to the Level 1 Palette of BPMN 2.0**

**ESSENTIAL REFERENCE WORKS:**

**“BPMN Method and Style” by Bruce Silver**

**“Process and Decision Modelling in BPMN/DMN” by Tom DeBevoise and James Taylor  
(both books available in paperback and for Kindle)**

**The BPMN Specification:  
<http://www.omg.org/spec/BPMN/2.0/>**



Care to discuss further?

[www.chellar.com/AnalysisFu/](http://www.chellar.com/AnalysisFu/)

LinkedIn: [es.linkedin.com/in/declanchellar/](https://es.linkedin.com/in/declanchellar/)

Twitter: @AnalysisFu