BPMN 2.0 cheat sheet

Events and activities

Events: something that happens during the process

Start single border · **Intermediate** double border · **End** thick border · **Non-interrupting** dashed border · **Catching** hollow icon: waits · **Throwing** filled icon: triggers

	Start			Intermediate				End
	Process	Event sub-process	Event sub-process, non-interr.	Catching	Boundary, interrupting	Boundary, non-interr.	Throwing	End
None no specific trigger	○	—	—	—	—	—	◉	●
Message something arrives from / goes to another participant	✉	✉	✉	✉	✉	✉	✉	✉
Timer date, deadline or cycle	🕒	🕒	🕒	🕒	🕒	🕒	—	—
Error failure that aborts the activity	—	↯	—	—	↯	—	—	↯
Escalation notifies a higher level without aborting	—	▲	▲	—	▲	▲	▲	▲
Cancel cancels a transaction	—	—	—	—	✕	—	—	✕
Compensation undoes work already completed	—	◀◀	—	—	◀◀	—	◀◀	◀◀
Conditional a data condition becomes true	≡	≡	≡	≡	≡	≡	—	—
Signal broadcast to anyone listening	△	△	△	△	△	△	▲	▲

Empty cell = combination not allowed by the specification. Cancel is only used in transactions.



How to read boundary events. An event attached to the boundary of an activity only acts while that activity is running. **Solid border (interrupting):** cancels the activity and diverts the flow. **Dashed border (non-interrupting):** the activity continues and a parallel path starts, e.g. a deadline reminder. Error and cancel always interrupt; a compensation boundary event fires after the activity has completed and needs to be undone.

Activities: the work



Task unit of work · **Sub-process** collapsed (+) · **Call activity** thick border, reusable · **Event sub-process** dotted border · **Transaction** double border, all or nothing

Task types (icon in the top-left corner)



User: a person performs it with system support (task inbox)



Service: performed by a system, an API or an automated service



Script: executed by the engine itself with a script



Send: sends a message to another participant



Receive: waits for a message to arrive



Manual: done without any system (e.g. signing on paper)



Business rule: evaluates a rule or decision table (DMN)

Markers (bottom centre)



Loop repeats while the condition holds · **Parallel multi-instance** several at once · **Sequential multi-instance** several, one after another · **Compensation** undoes the work · **Ad hoc** steps in no fixed order

Quick rule: name every task with verb + object (“Approve invoice”), not with a noun (“Invoice”).

BPMN 2.0 cheat sheet

Gateways, flows, data, pools and example

Gateways: where the flow splits or joins



Exclusive (XOR)
Split: takes a single path, the first whose condition is true; define a default flow. **Join:** passes each arrival through without waiting.



Parallel (AND)
Split: activates all paths at once, without conditions. **Join:** waits until all of them arrive.



Inclusive (OR)
Split: activates one or more paths, every one whose condition is true. **Join:** waits for all paths that were activated.



Event-based
Waits and follows the path of the first event that occurs (message, timer, signal…); the others are discarded. Only catching events or receive tasks may follow it.



Complex
Its behaviour is defined by its own expression, e.g. continue when 2 of 3 approvals arrive. Use it only if no other gateway fits.

Golden rule: close every split with a gateway of the same type.

Flows: how elements connect



Sequence: order of execution within one pool



Default: slash at the source: taken if no condition is true



Conditional: diamond at the source: taken if its condition is true



Message: communication between different pools

Association: links data and annotations to elements

Data and artifacts



Data object: information created or used in the instance



Collection: list of objects (||| marker)



Input / output: hollow arrow = input; filled = output

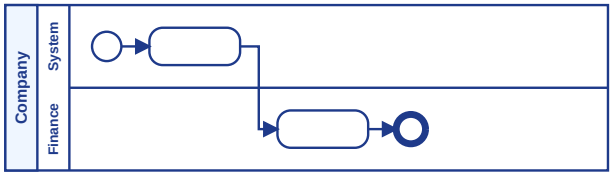


Data store: information that persists outside the process (DB, ERP)



Annotation: text note for the reader

Pools and lanes

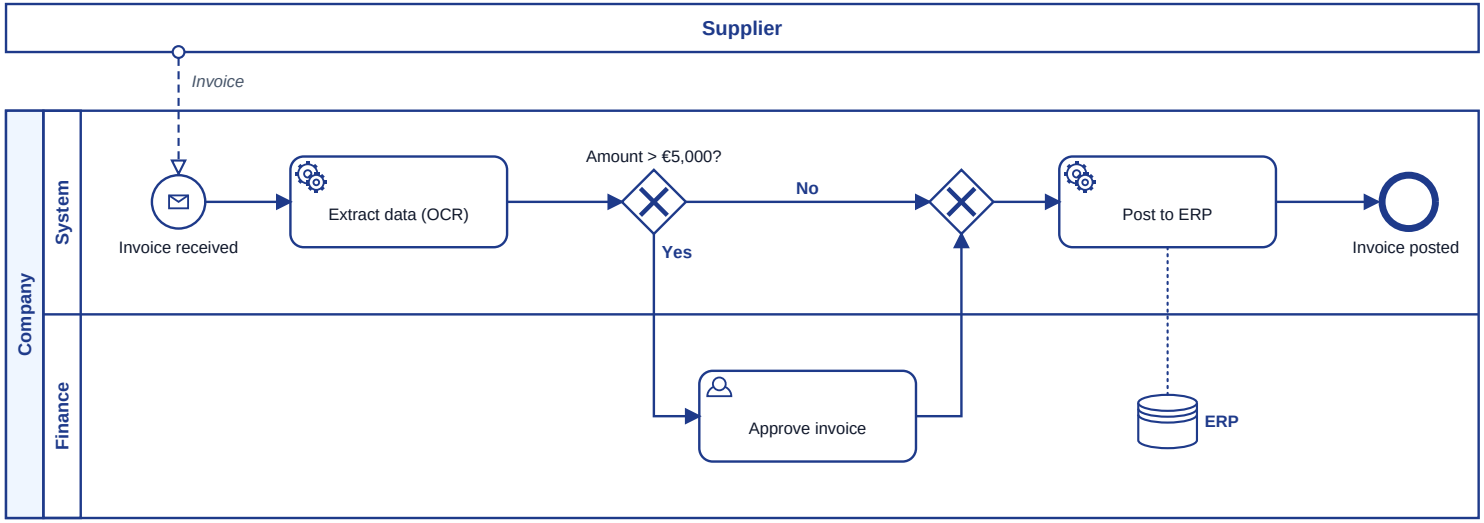


Pool: a participant (your company, a customer, an external system). **Lane:** a role or department inside the pool. A sequence flow never leaves its pool: between pools there are only message flows. An empty (collapsed) pool is a participant whose process is not detailed.

8 common mistakes

1. Closing a parallel gateway with an exclusive one (or vice versa): the flow deadlocks or work is duplicated.
2. Connecting two pools with a sequence flow: between participants use a message flow.
3. Using a message flow inside the same pool: between lanes use a sequence flow.
4. Gateways without a question or labelled branches, or without a default flow.
5. Tasks named with a noun (“Invoice”) instead of verb + object (“Approve invoice”).
6. Paths without an explicit start or end event: nobody knows where they begin or how they end.
7. Connecting data objects with sequence flows: data is linked with associations.
8. More than ~30 elements on one level: group steps into sub-processes.

Example: invoice approval



The exclusive gateway decides by amount: only invoices above €5,000 go through a person. The second exclusive gateway joins both paths before posting. The supplier is another pool, so the invoice arrives through a message flow.