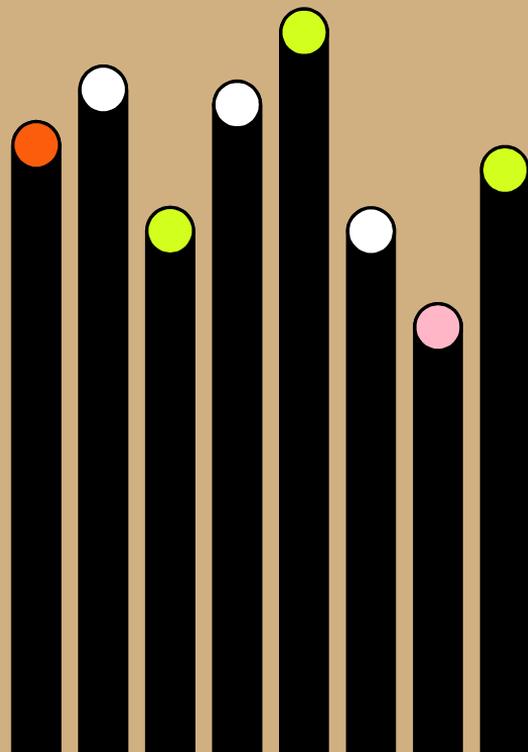


Choose Your Own Adventure

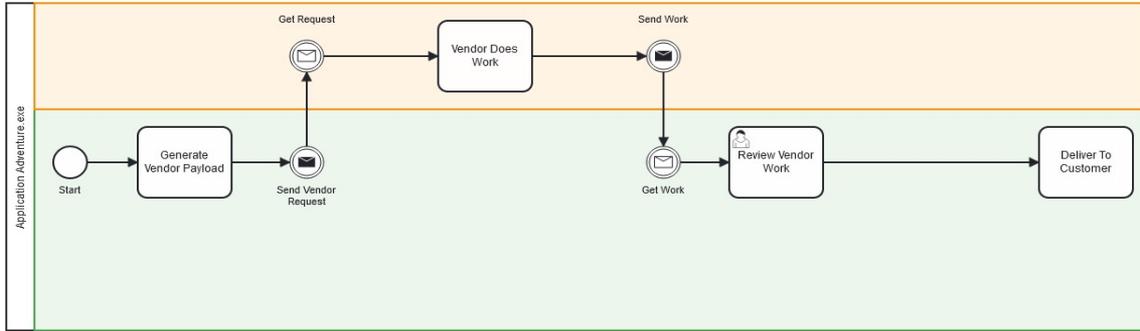
How First American implemented dynamic
User Task outcomes

Brian Perry and Bill Powell

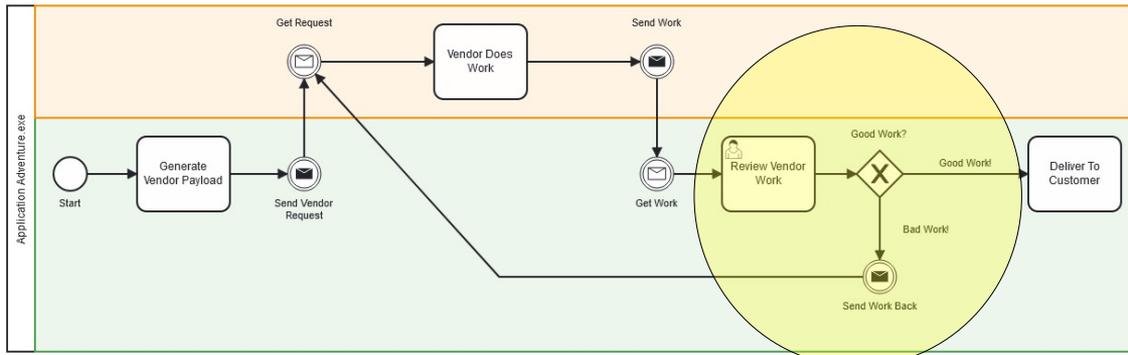


The Problem With Tasks

Camunda tasks work great when your path is a straight line. Did the work get completed, or did we put it off off?



But what happens when you need to do something different what happens at the fork in the road?



Choose Your Own Adventure

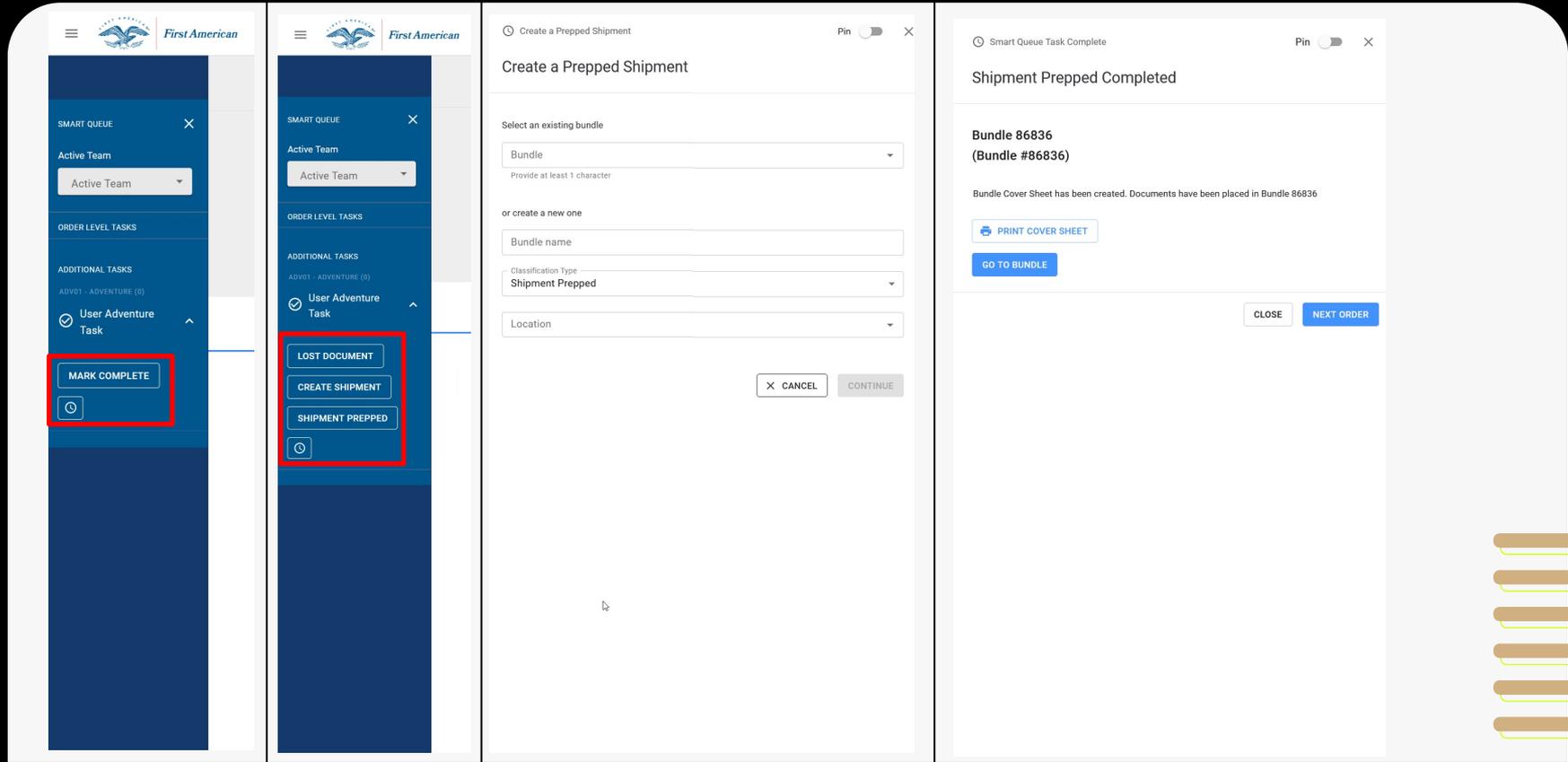
Based on the feature workflows presented By Howard Chang and Kenny Dang in CamundaCon 2022



- Workflows refined to micro workflows
- Workflows also act to close original task
- Task Resolutions bubbled up to original workflow, allowing for use in future evaluations.
- This allows branching paths.



What it Looks Like



Refactor Time!

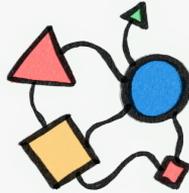
We had over **80** models, some with multiple User Tasks, that must support this feature.

The screenshot shows the Camunda Cockpit interface. At the top, there is a navigation bar with 'Camunda Cockpit' and a user profile 'Bill Powell'. Below this is a breadcrumb 'Dashboard » Processes'. The main content area is titled 'Search process instances' and includes a search input field with the placeholder 'Add criteria'. Below the search field, a summary states '160 process definitions deployed'. A large exclamation mark is placed over this text. To the right of the summary are tabs for 'List' (selected) and 'Previews'. Below the summary is a table with the following columns: State, Incidents, Running Instances, Name, Tenant ID, History view, Report, and Action. The table contains four rows of data.

State	Incidents	Running Instances	Name	Tenant ID	History view	Report	Action
✓	0	1	Add Document To Order		History view	📊	🗑️
✗	16	16	Add Event		History view	📊	🗑️
✓	0	0	Add Payment		History view	📊	🗑️
✓	0	0	Add Shipping Label		History view	📊	🗑️

Design Objectives

Extensible



Allow for an indefinite number of new resolution paths

Low-impact



Minimize changes to existing business models

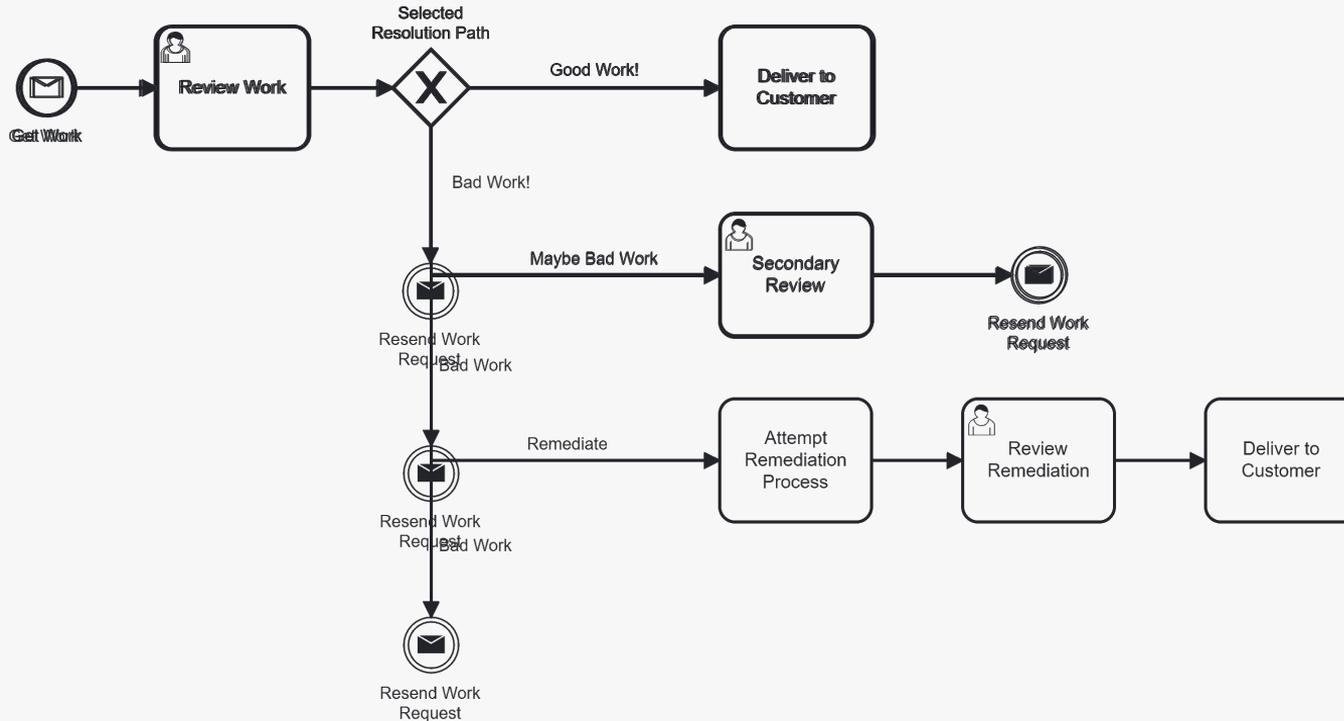
Traceable



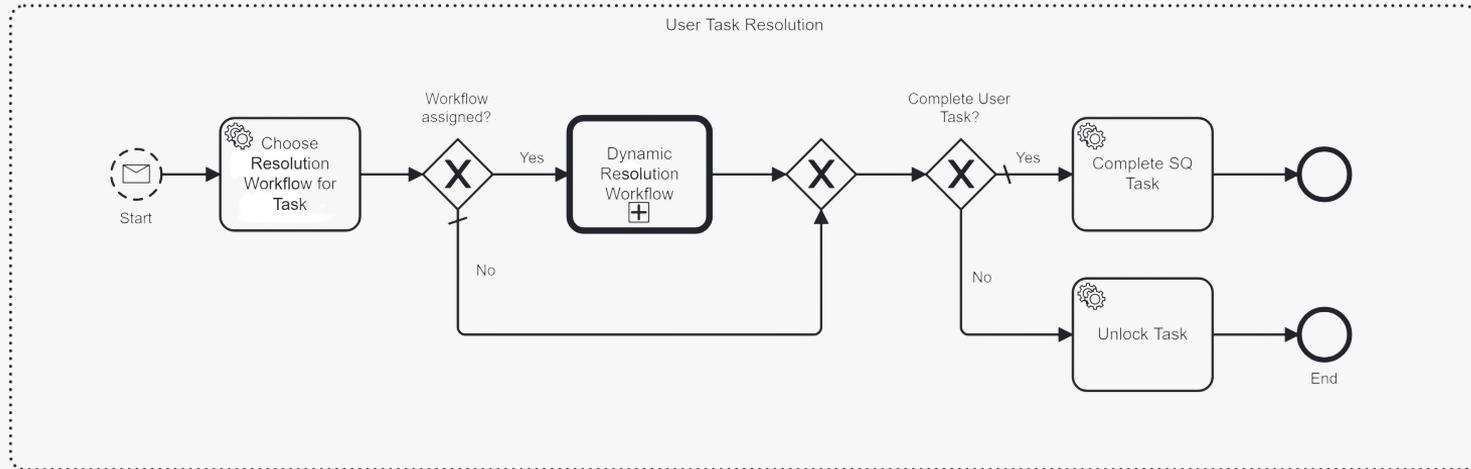
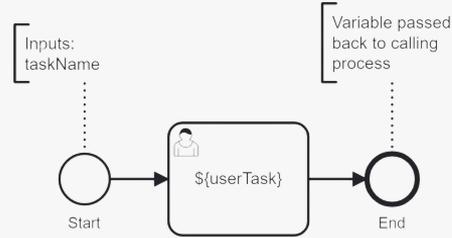
Easy to design, troubleshoot and support



Monolithic model won't work



User Task Process



Leverage Camunda support for expressions



taskNameOverride

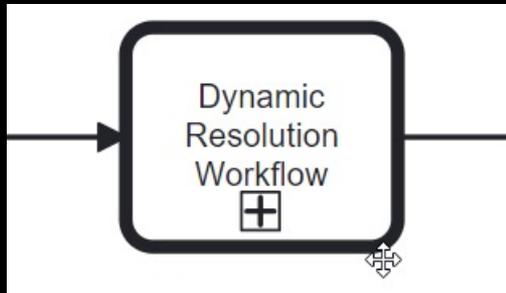
Local variable name

Assignment type

String or expression

Value

Start typing "\${}" to create an expression.



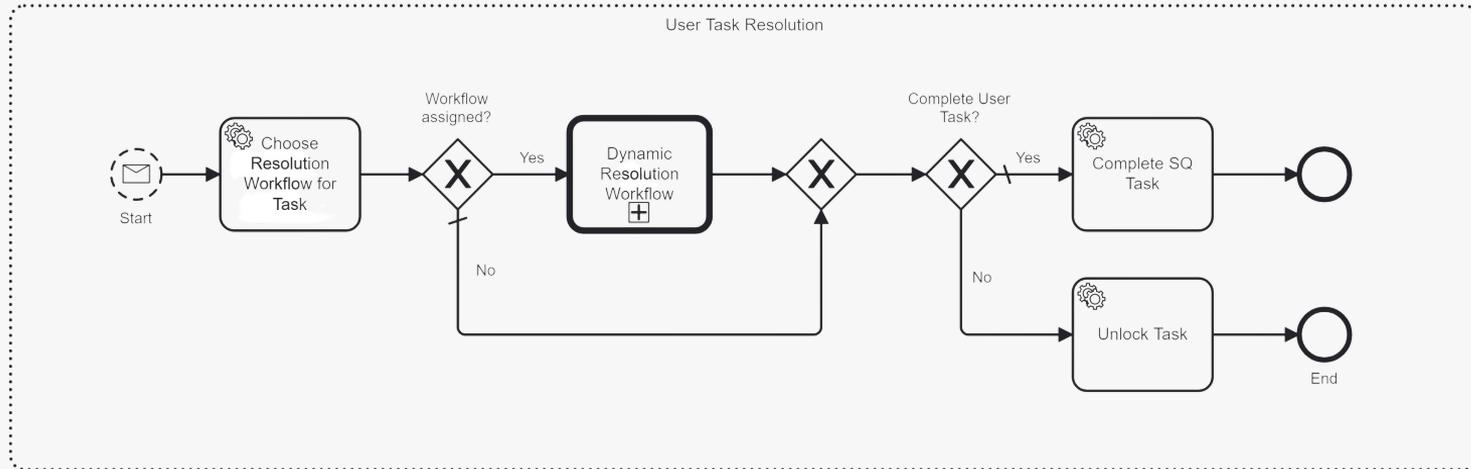
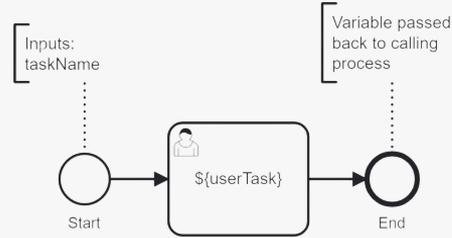
Called element

Type

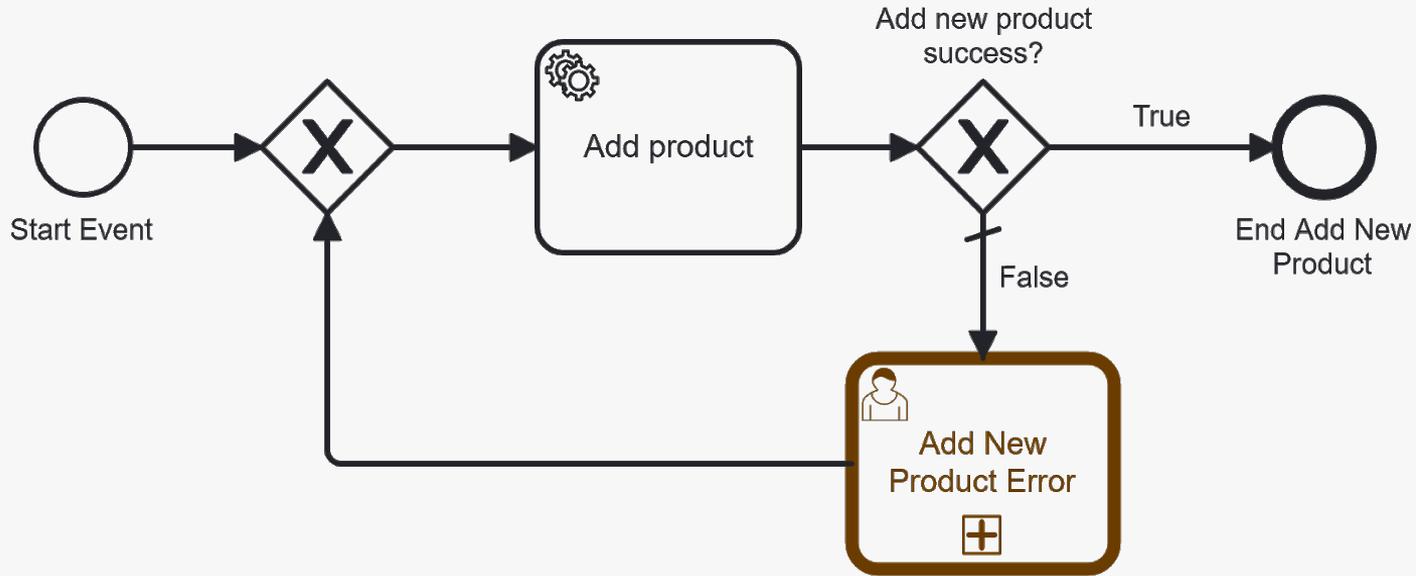
BPMN

Called element

User Task Process

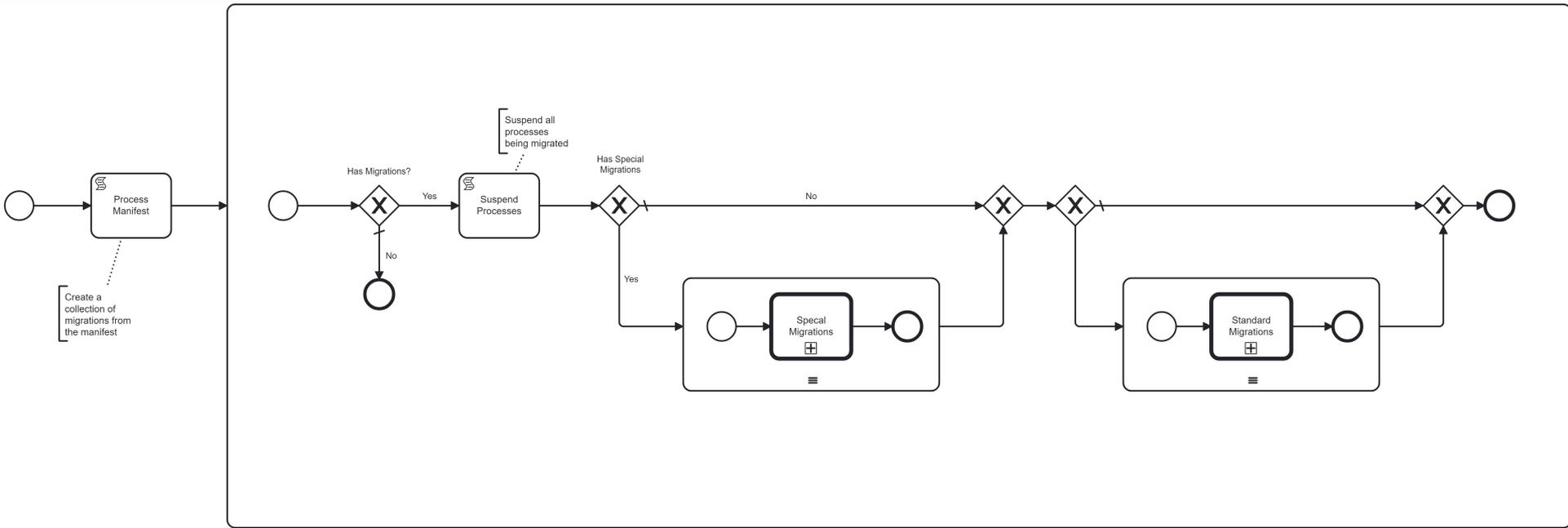


Processes changed to Call Activity



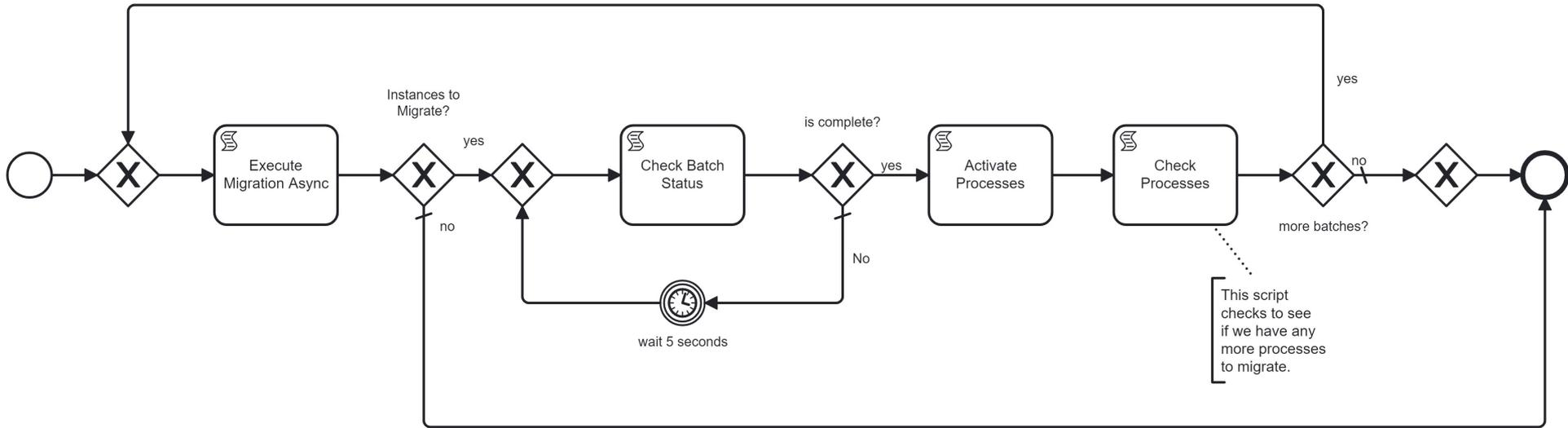
Migration Runner

Moving existing running processes from the old version (User Task) to the new version (Call Activity)



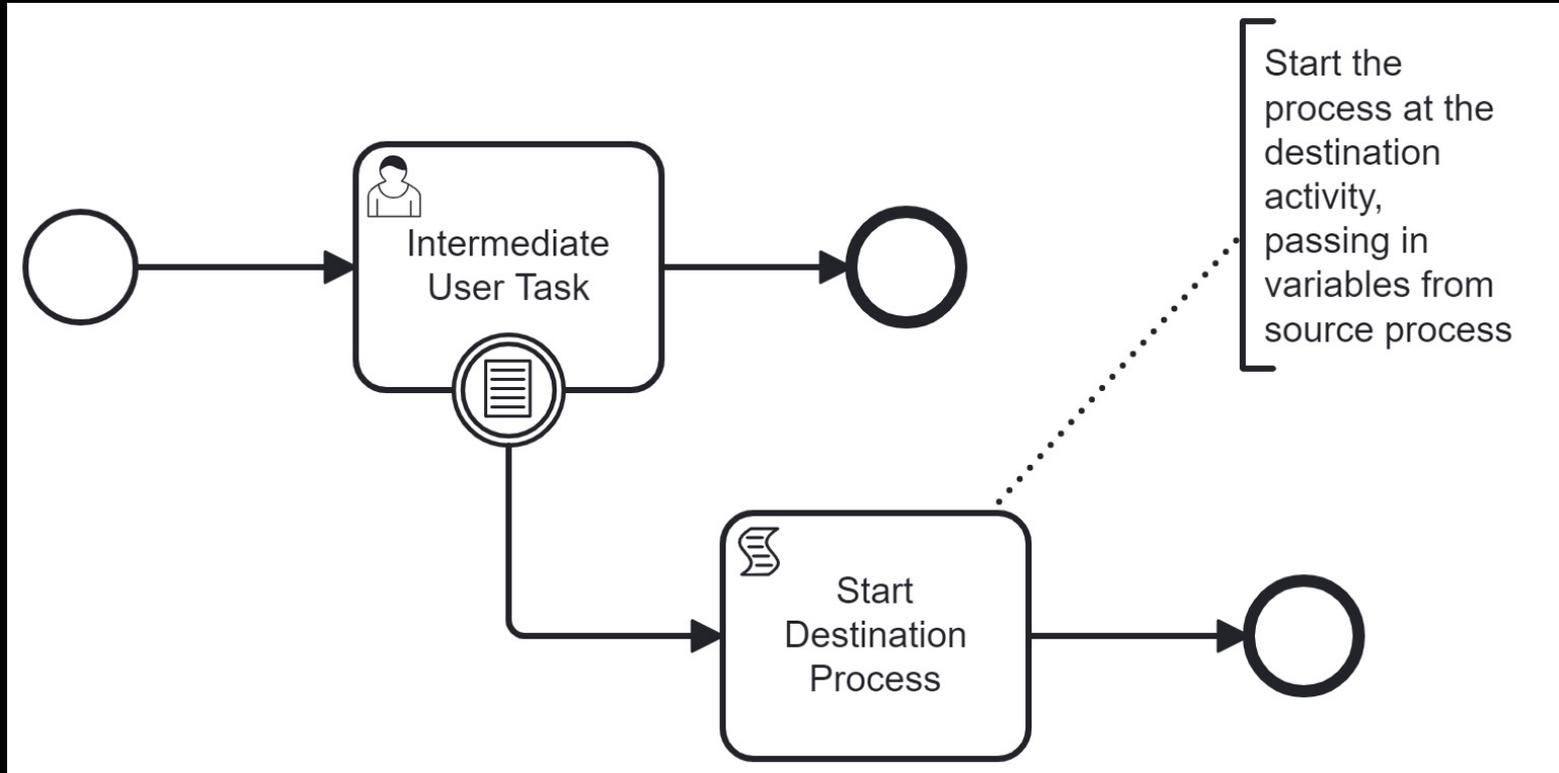
Migration Core

Run migrations asynchronously in batches



Intermediate Migration Process

A technique which allows migration to dissimilar task types



Cockpit – processes in action