

Flow and Subflow

A subflow provide a common sequence of activities to be used by several flows. You can include subflows in your flows in the same way as you include activity and tool. You can also connect subflows to other nodes in the same way.

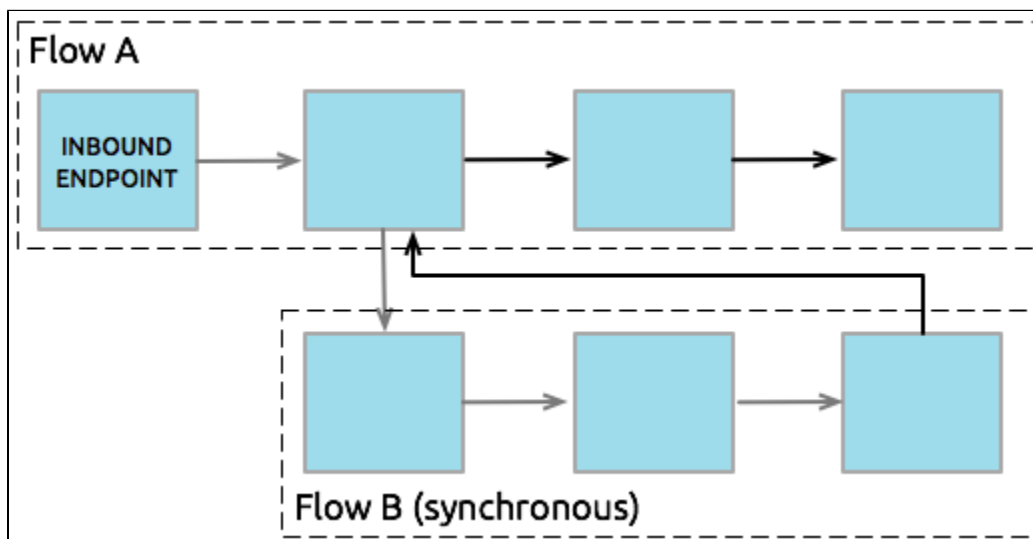
A subflow provides the following benefits:

- Reusability and reduced development time.
- Consistency and increased maintainability of your similar flows (consider a subflow as analogous to a programming macro, or to inline code that is written once but used in many places).
- Flexibility to tailor a subflow to a specific context (for example, by updating the output queue or data source information).
- Asynchronous subflow can perform time-consuming tasks, such as writing data to an external database or emailing a message, without stalling main flow, the flow that triggered its execution.

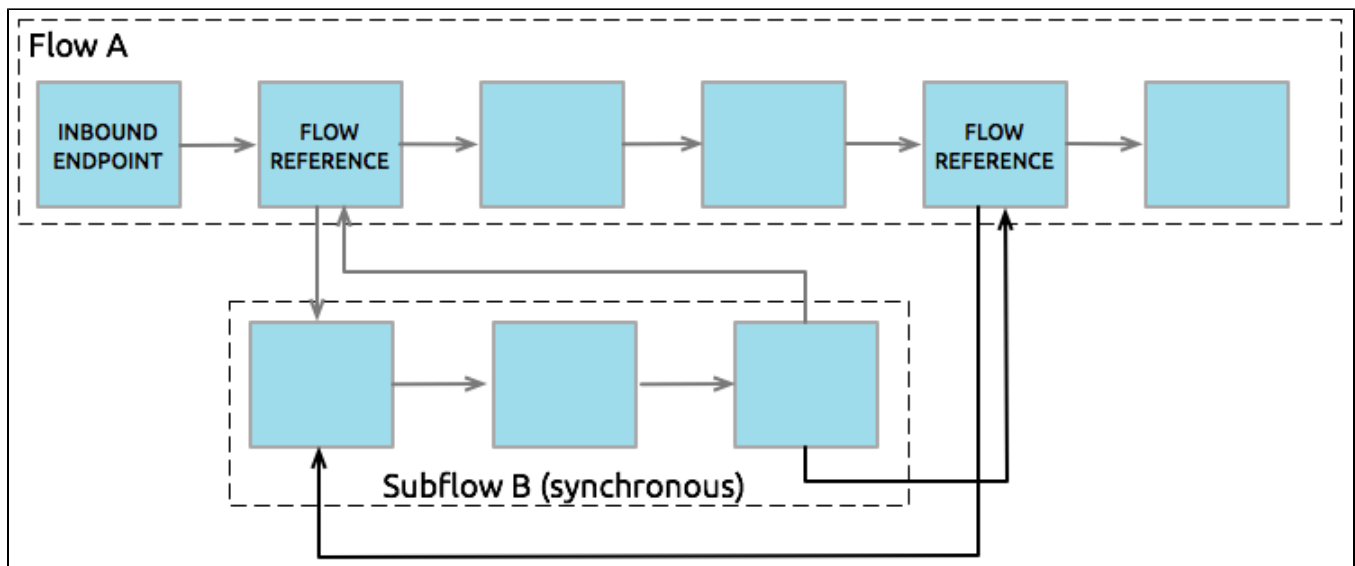
Types of Subflow

1. Synchronous

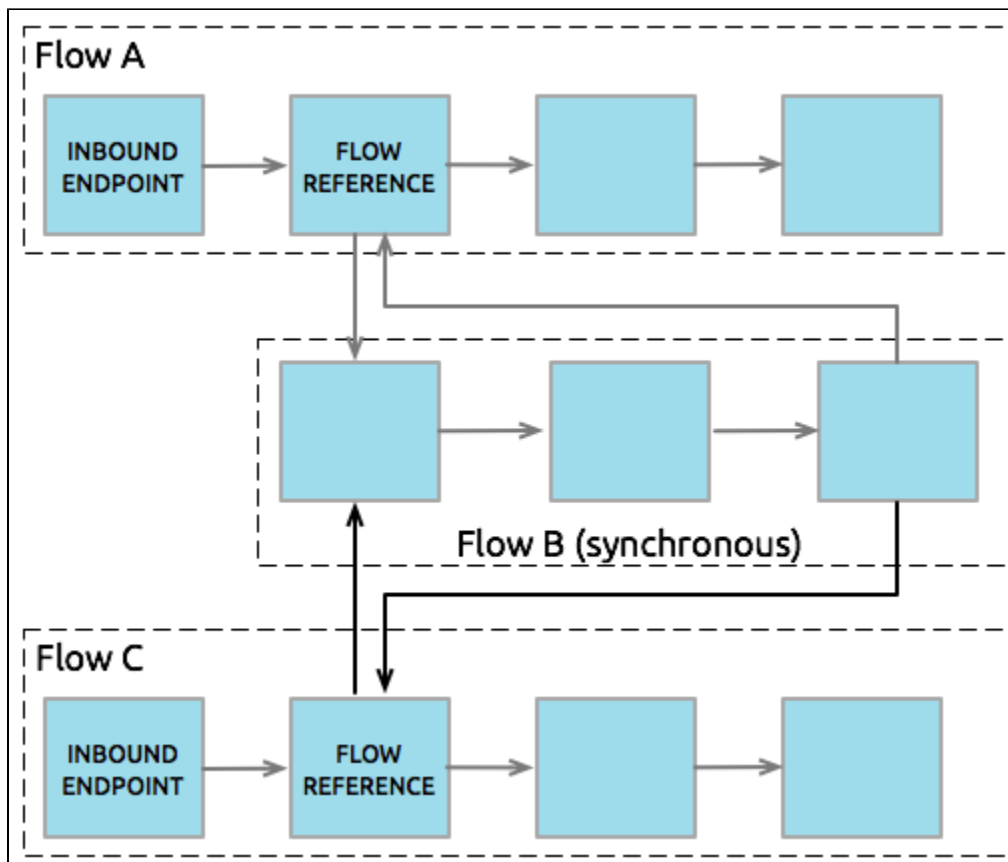
A subflow run synchronously (relative to the flow that triggered its execution) and always inherits both the processing strategy and exception strategy employed by the triggering flow. While a subflow is running, processing on the triggering flow pauses, then resumes only after the subflow completes its processing and hands the message back to the triggering flow.



Case 1: Standard synchronous subflow



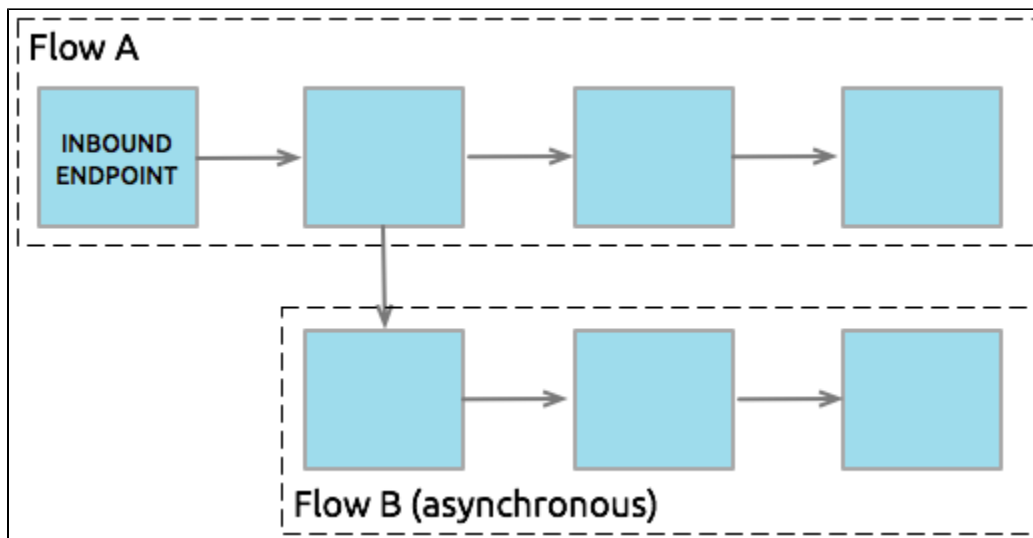
Case 2: Using Subflow multiple times in a flow.



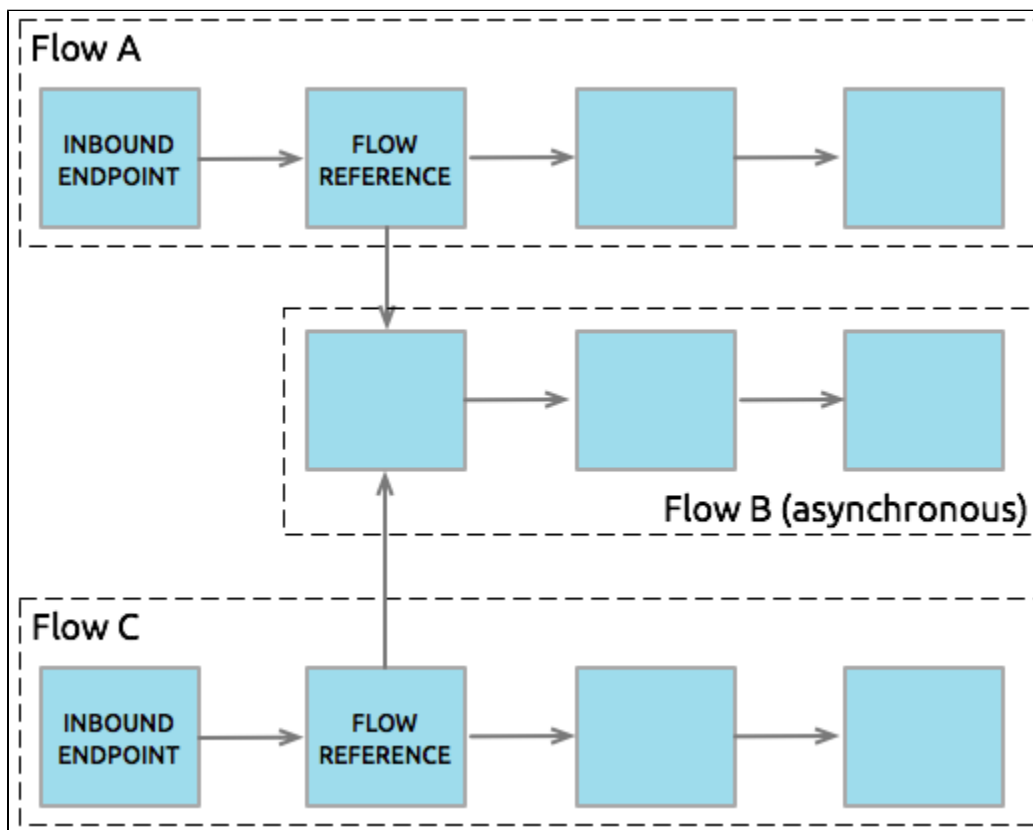
Case 3: Same subflow used by multiple flows

2. Asynchronous

An asynchronous flow runs simultaneously and asynchronously in parallel to the flow that triggered its execution. When a flow triggers an asynchronous flow, thus triggering its execution, it simultaneously flows to the next activity processor in its own flow. Thus, the two flows – triggering and triggered – execute simultaneously and independently, each finishing on its own. This type of flow does not inherit processing or exception strategies from the triggering flow.



Case 1: Standard asynchronous subflow



Case 2: Same subflow used by multiple flows