Parallel Gateways

Gateways serve to split or join process flows. A split gateway is a point where a single inbound sequence flow diverges into two or more outbound branches. A join gateway is where multiple inbound branches of a process flow merge into a single outbound branch.

The parallel gateway indicates where activities begin to be performed in parallel. All active flows started at the AND-split gateway must finish before the process continues. In other words, the synchronization of parallel flows waits until the tokens from all incoming sequence flows have arrived before proceeding. For this reason, every AND-split gateway has a corresponding AND-join gateway.

Here's an example of a process flow that utilizes parallel gateways.

- The process is instantiated when the traveler receives a boarding pass.
- Next, the traveler proceeds to the security checkpoint.
- Upon arriving at the checkpoint, the sequence flow splits at a parallel gateway. Unlike other gateways, parallel gateways aren't typically labeled.
- One sequence flow calls for the traveler to submit her carry-on luggage to be screened.
- In parallel, the traveler herself must complete a screening process.
- The sequence flows are synchronized at another parallel gateway.
- Only after both parallel process flows are completed can the travel proceed to the departure gate.
- The process instance is finished when the traveler arrives at her gate.

The important thing to remember is that the AND-join gateway acts as a gate rather than a simple pass-through. That is, the gateway holds incoming tokens until tokens from all active flows in the parallel structure have arrived. Only then are the tokens synchronized, and the process continues.

Let's use token simulation to see this process in action. The tokens from the parallel flows arrive at the AND-join parallel gateway independently. However, the process flow doesn't continue to the final task until both tokens have arrived at the AND-join gateway and have been synchronized.