



Core concepts

The section contains the following topics:

- [Core concepts, on page 1](#)

Core concepts

This section defines the main concepts and components used in the Crosswork Workflow Manager (CWM) that help understand how the platform works and its features. You can familiarize yourself with them to ease the very first steps with CWM.

- **Activity:** An activity is a function that executes a single, well-defined action against the target, outside system (whether it's an app or solution). Activities are defined in adapters and allow communication with an outside system.
- **Adapter:** Adapters are responsible for communication with external services, like other applications, systems or environments. The adapters define and expose activities that are consumed by workflow definitions. Every adapter can be associated with the worker that will execute the adapter activities.
- **Adapter SDK:** The adapter SDK automatically generates the structure for the required adapter's components. Developers can further define activities that are needed and then extend the integrations with the client environment.
- **Event:** Events are created during workflow execution based on the occurrences defined in the workflow definition. All events that happened during workflow execution are recorded in the **Job Event Log** table in CWM UI.
- **Execution engine:** CWM has an internal worker called the execution engine. It enables the execution of workflow definition. This worker is not visible in the CWM UI.
- **Job:** A job represents the single execution of a particular workflow definition. To be able to run a job in Crosswork Workflow Manager, you need first to add your workflow definition to the CWM. Running a new job instantiates a workflow definition stored in CWM. Before starting a job run, you enter the initial start data (Job variables). It means that your workflow executions are isolated and may use different data than other executions of the same workflow definition.
- **Workflow:** Workflows help you capture, organize and automate processes with repeatable actions performed in a specified order. In the context of CWM, documentation differentiates between:
 - **workflow definition:** piece of code written in JSON or YAML, based on the Serverless Workflow Specification and vendor-neutral, domain-specific language.

- **workflow execution (job):** single execution of a workflow definition.
- **Workflow engine:** The workflow engine manages the way how your workflow definitions are interpreted and conducted. It receives events, schedules tasks, and manages the execution of workflows.
- **Worker:** Workers carry out the workflow and are responsible for executing the workflow definition code, relevant adapter code and activities defined in the workflow definition. Depending on your needs and scale, you can have multiple workers for every workflow definition. Your worker can be associated with one adapter and its activities or with multiple ones.
- **CWM UI:** The CWM UI is a graphical user interface of CWM that allows users to interact with the system and gives access to its core functionalities.