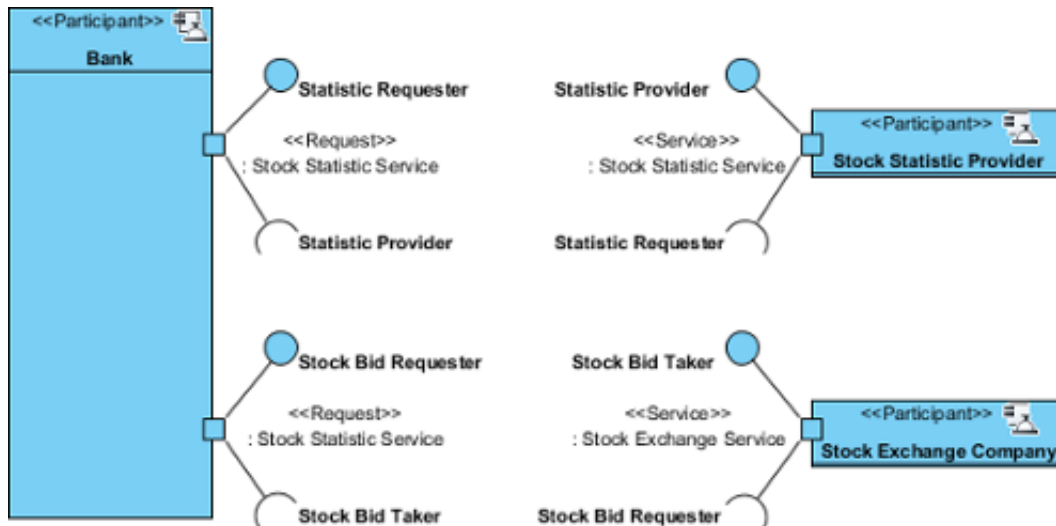


# How to draw a Service Participant Diagram in SoaML

[SoaML](#) service participant diagram focuses on the person, organization, system or anyone who take part in a services architecture. Modelers use service participant diagram to represent these participants as well as the interfaces they required or provided in accomplishing services. Note that the way how participants interact is not modeled in service participant diagram.






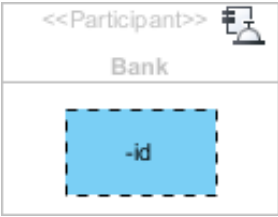
## Creating service participant diagram

1. Select **Diagram > New** from the application toolbar.
2. In the **New Diagram** window, select **Service Participant Diagram**.
3. Click **Next**.
4. Enter the diagram name and description. The **Location** field enables you to select a model to store the diagram.
5. Click **OK**.

## Notations

The description of notations is either extracted or derived from the OMG SoaML Specification v1.0.1.

Name	Representation	Description
Participant		<p>A Participant represents some (possibly concrete) party or component that provides and/or consumes services (participants may represent people, organizations, or systems that provide and/or use services). A Participant is a service provider if it offers a service. A Participant is a service consumer if it uses a service. A participant may</p>

Name	Representation	Description
Agent		<p>provide or consume any number of services. Service consumer and provider are roles Participants play: the role of providers in some services and consumers in others, depending on the capabilities they provide and the needs they have to carry out their capabilities. Since most consumers and providers have both services and requests, Participant is used to model both.</p> <p>In general, agents can be software agents, hardware agents, firmware agents, robotic agents, human agents, and so on. While software developers naturally think of IT systems as being constructed of only software agents, a combination of agent mechanisms might in fact be used from shop-floor manufacturing to warfare systems.</p> <p>Used as a composite component for the participant.</p>
Part		
Property		<p>A property is a structural feature. It relates an instance of the class to a value or collection of values of the type of the feature. A property may be designated as an identifier property, a property that can be used to distinguish or identify instances of the containing classifier in distributed systems.</p>
Service Port		<p>A service port is the feature that represents the point of interaction on a Participant where a service is actually provided. On a service provider, this can be thought</p>

Name	Representation	Description
Request Port		<p>as the "offer" of the service (based on the service interface). In other words, the service port is the point of interaction for engaging participants in a service via its service interfaces.</p> <p>A consumer of a service specifies the service interface they require by using a request port.</p>
Port		<p>Port is extended with a connectorRequired property to indicate whether a connector is required on this port or the containing classifier may be able to function without anything connected.</p>
Service Channel		<p>A ServiceChannel provides a communication path between consumer Requests and provider services.</p>
Connector		<p>Connect parts in a participant, if any.</p>
Capability		<p>A Capability models the ability to act and produce an outcome that achieves a result that may provide a service specified by a ServiceContract or ServiceInterface irrespective of the Participant that might provide that service. A ServiceContract alone, has no dependencies or expectation of how the capability is realized – thereby separating the concerns of "what" vs. "how." The</p>

Name	Representation	Description
Realization		<p>Capability may specify dependencies or internal process to detail how that capability is provided including dependencies on other Capabilities. Capabilities are shown in context using a service dependencies diagram.</p> <p>Connects service/request port and interface to represent the interface provided by the participant at specific port.</p>
Usage		<p>Connects service/request port and interface to represent the interface required by the participant, at specific port.</p>
Dependency		<p>Shows that a participant/interface/port depends on a participant/interface/port.</p>