

Basic Architectural Modeling

Besides defining behavior in activity diagrams, and structure in class diagrams, you also need to model the architecture of an xUML service. Architectural information describes all components required to build a service, and is modeled in a **Component Diagram**.

Component Diagrams define the types of all used entities like components, classes, interfaces, and artifacts.

We distinguish between components defined by the modeler and implemented as xUML services, and components accessed and used by the Bridge.

- The first category is called Bridge components. Their stereotypes have the prefix E2E, such as [<E2ESOAPService>](#).
- The latter category is named **backend** components. The notion backend derives from the logical location of these components.

In front of the service, we have a service client, then we have the service itself (sometimes called **frontend** service), and in the back of this service there are utilized services – i.e. the backend services. The term frontend is also used to explain the building blocks of the Runtime in section [xUML Runtime Architecture](#). Besides frontend and backend components, it is also possible to configure Proxy components. These components are instances of an Apache reverse proxy shielding SOAP and HTTP services. However, proxy concepts are no basic architectural concepts but are described in [Advanced Behavioral Modeling > Security Model - Proxy](#).

Examples of frontend and backend components are:

- Frontend components: [SOAP](#), [HTTP](#), [Java](#), [Timer](#), [Scheduler](#), [SAPRFC](#), etc.
- Backend components: [SQL database](#), [SOAP services](#), [SAP systems](#), etc.

Typically, these diagrams are created using the xUML Components Wizard. Please refer to [The Components Wizard](#) to learn more about this wizard.

Related Pages:

- [Frontend Components](#)
- [Backend Components](#)