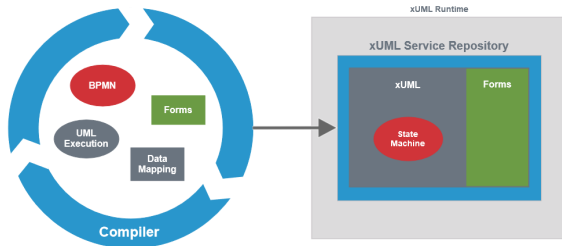


Technical Concepts

Scheer PAS Designer is a low-code modeling tool to create applications and integration services. Using BPMN and UML, you can create microservices that are directly executed without source code generation. We call them **xUML services**.

With the Designer, you can model your business processes in BPMN, design your forms, add some execution, and compile all to an xUML service. The validation and compilation is done continuously in the background while you are working on your model.

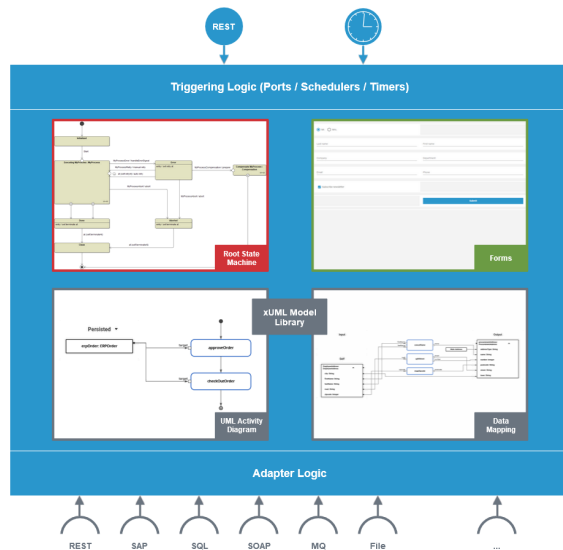


Compiled xUML services can be deployed to and executed by the xUML Runtime. Using the **Scheer PAS ANALYZER**, you can directly debug and analyze your model, not some generated source code.

xUML Service

The Designer allows to create BPMN models in combination with UML execution, Data Mapping and Forms in a way that serves as process documentation, and in the same time as production service.

xUML services are platform independent. Their logic is modeled as flows that are triggered by modeled endpoints, such as REST, Schedulers and Timers.



The (asynchronous) process logic is modeled as BPMN, completed by (synchronous) UML activities that implement the execution.

- [Technical adapters](#) to access backend systems (e.g. REST, SOAP, OData, SAP, SQL, and more) can be implemented via libraries.
- [Data mapping](#) can easily be defined in a mapping diagram.
- UIs can be added by an [integrated form editor](#).

All that is compiled into a platform independent xUML service repository that translates the modeled logic into an UML state machine. This state machine handles the actual process.

Refer to [xUML Service State Machines](#) for more details on the state machine and related options.

xUML service repositories can be deployed to the [Scheer PAS BRIDGE](#), or as a standalone xUML service Docker container. The contained state machine of an up and running service is accessible via a REST interface. You can use this interface to get information on the service's/process' states, and to trigger events within the service.

Refer to [xUML Service Interface](#) for more information on the REST interface.

On this Page:

- [xUML Service](#)

Related Pages:

- [xUML Service State Machines](#)
- [xUML Service Interface](#)
- [Form Communication](#)
- [Distributed Tracing](#)
- [xUML Runtime Architecture and Transaction Concept](#)

Related Documentation:

- [Bridge Integration Platform](#)
- [xUML Services Reference Guide](#)
 - [xUML Service Adapters](#)