Lab Introduction

Getting Started with SwitchYard

Lab Objective

This session provides a developer's introduction to SwitchYard. By the end of the session, you will have experience with:

1. The structure and layout of a SwitchYard application.
2. Implementing service logic and configuring service bindings.
3. Configuring declarative transformation.
4. Unit testing services.
5. Deploying to a SwitchYard runtime.
6. Monitoring and debugging SwitchYard apps.

Lab Structure

The training consists of four labs (lab1 - lab4), which contain SwitchYard applications in various stages of completion. Lab 1 provides a detailed guided tour of a SwitchYard application and covers the basic building blocks you need to complete the other labs.

Lab Key

TODO Lists

TODO

This is a TODO list, which defines tasks which you need to perform during the lab. If you see one of these on a lab slide, make sure you follow each step in the TODO list.

Lab Key

FYI Notes

FYI

This is a note which provides background on a given step in the lab or a particular configuration or code snippet.

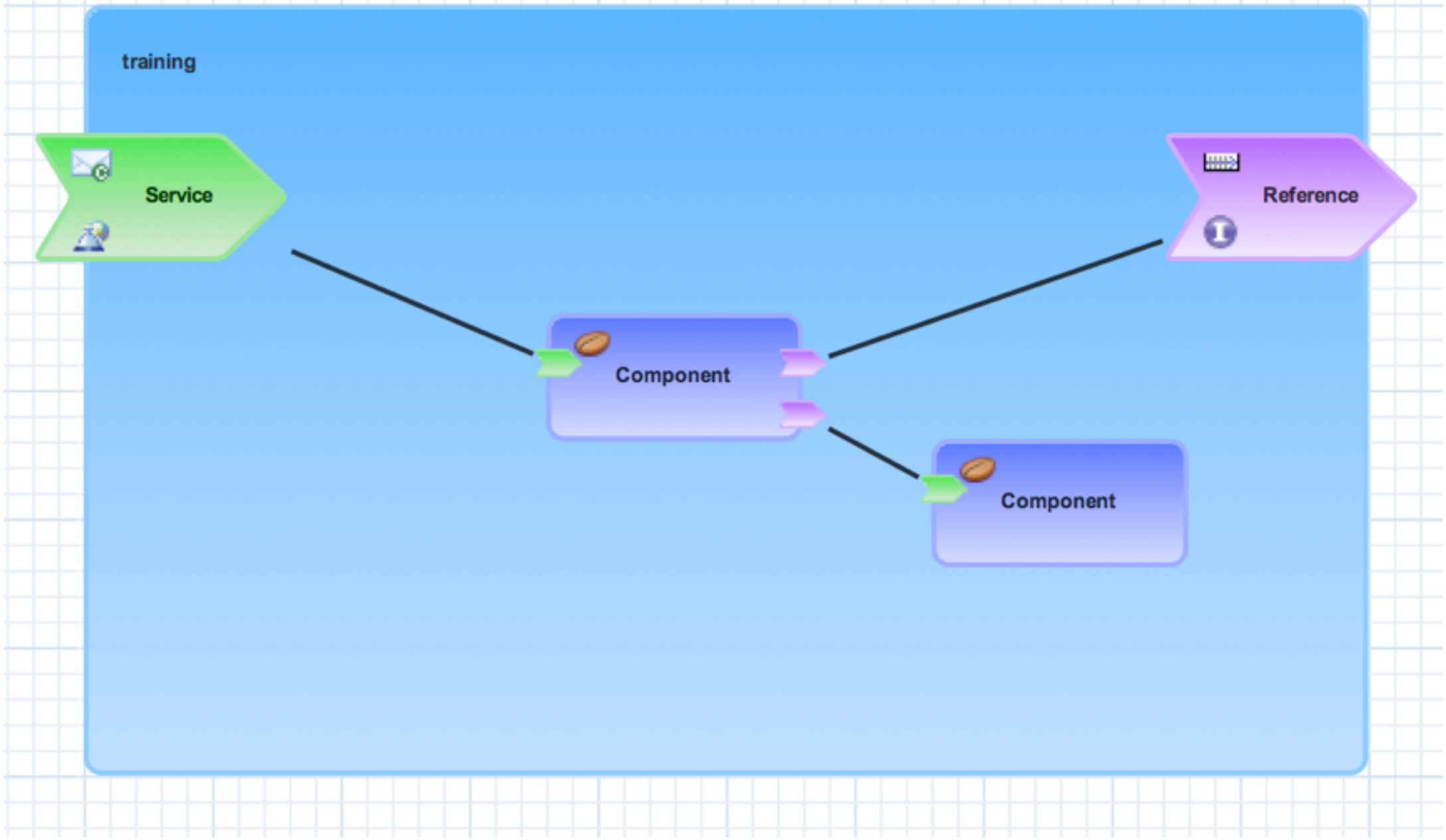
Prerequisites

- Eclipse Kepler or JBoss Developer Studio 7 with SwitchYard tooling installed
- EAP 6.1
- SwitchYard Installer (used in lab #1)
- Important links:
  - Install Guide : https://docs.jboss.org/author/display/SWITCHYARD/Installation+Guide
  - Downloads : http://www.jboss.org/switchyard/downloads

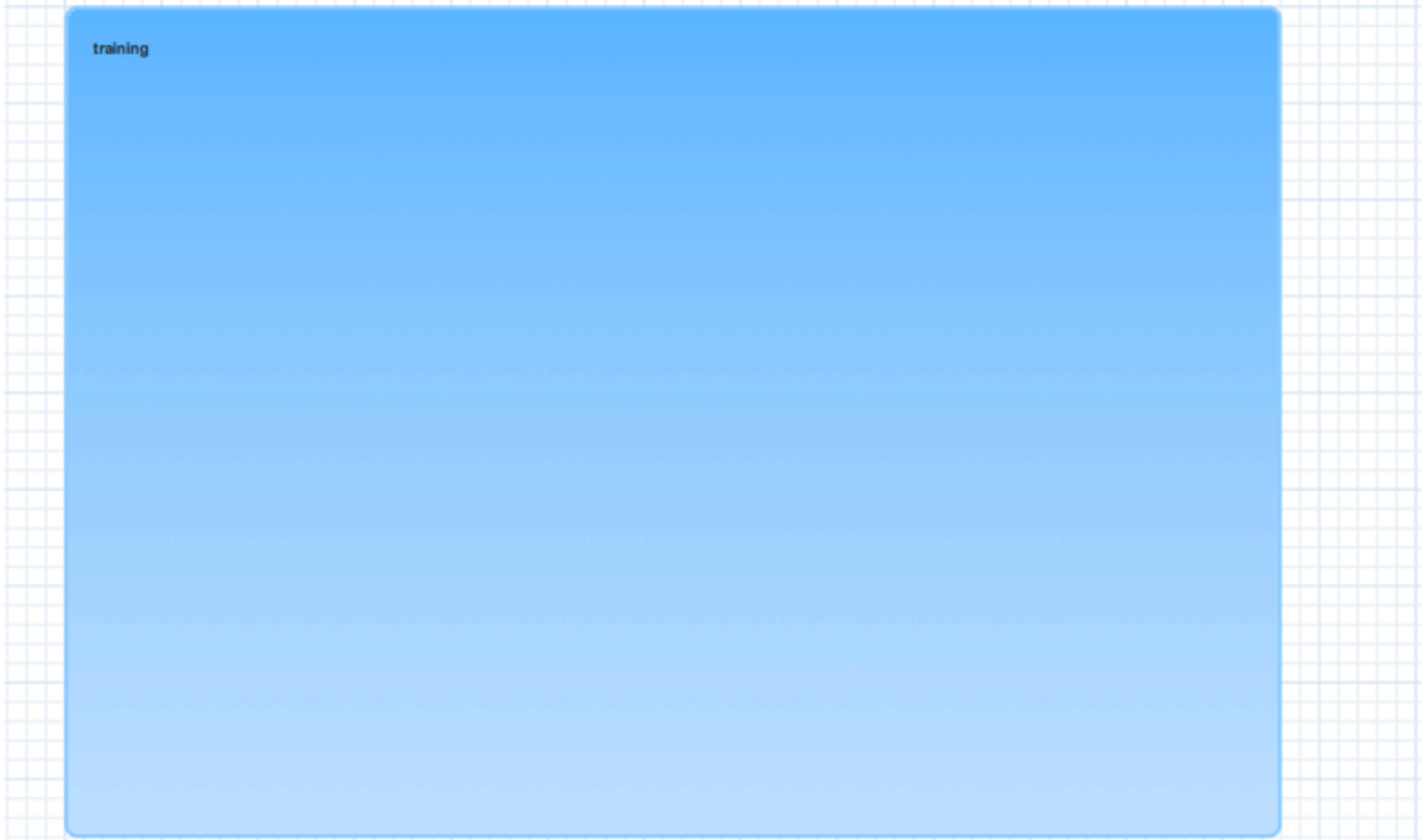
Before we get started

- A little background ...

SwitchYard Application



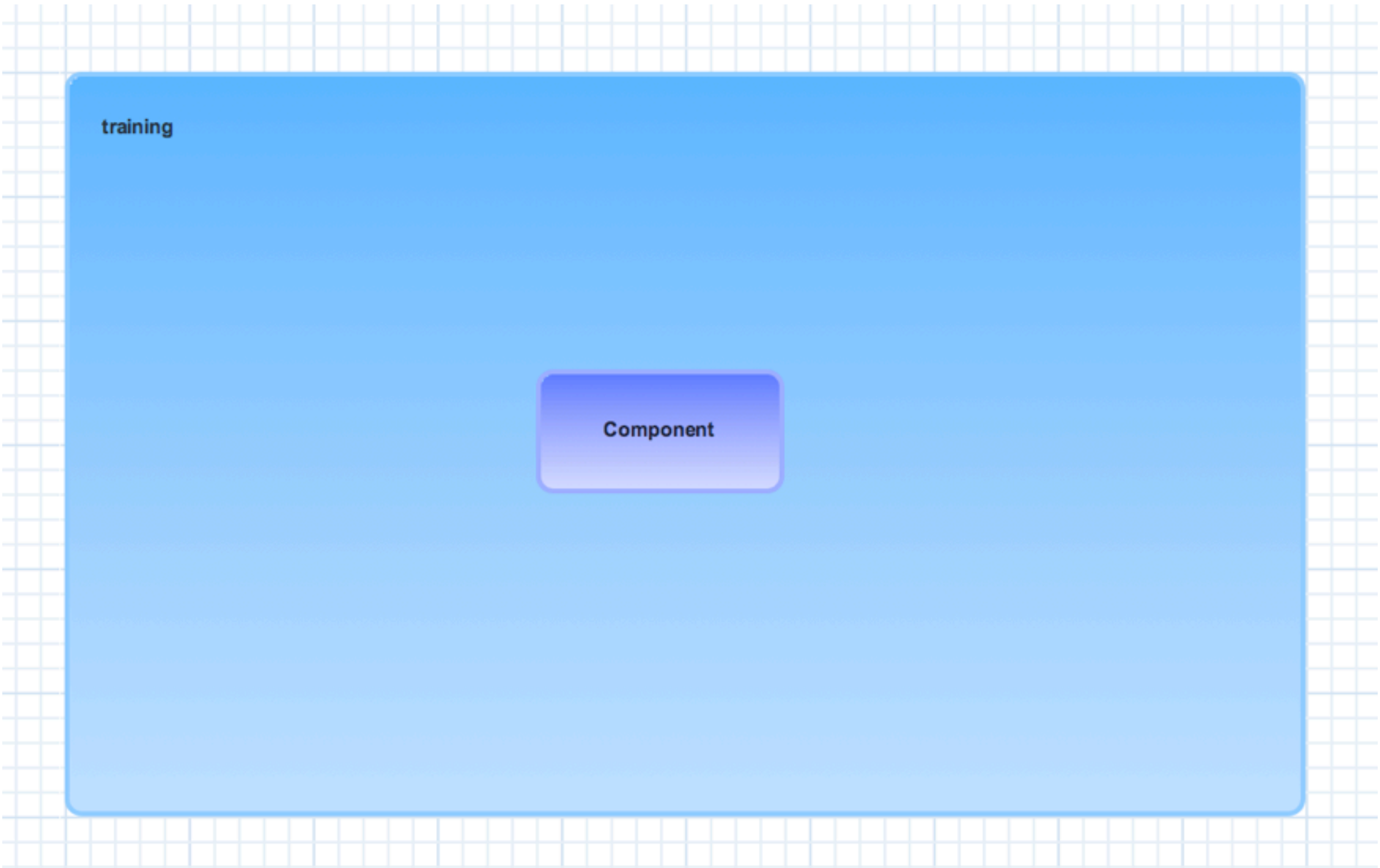
Composite



Composite

- Defines the application boundary
- Application name (documentation)
- Application namespace (important)
- One per application

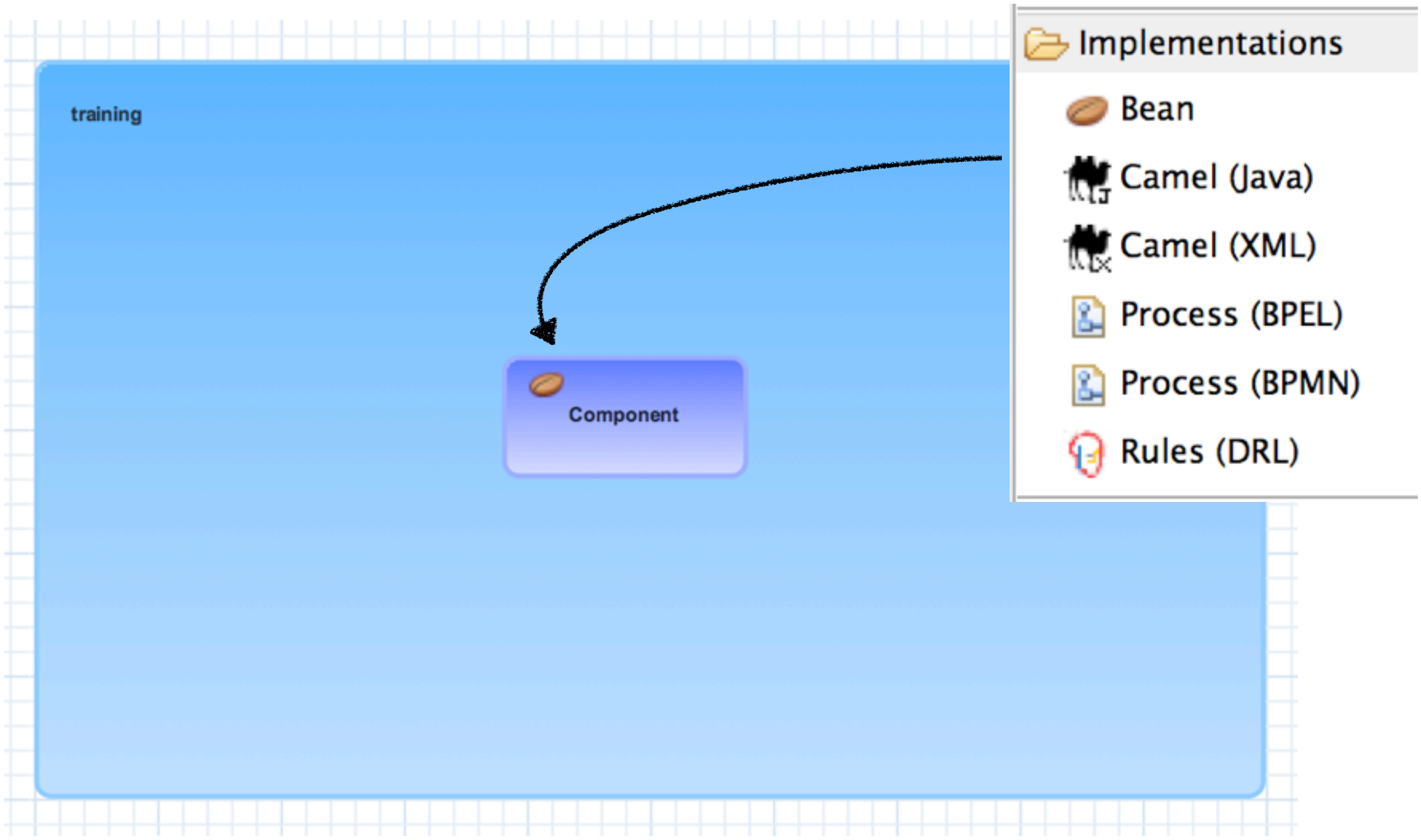
Component



Component

- Container for application logic
- Worthless without an implementation
- 0 ... 1 services
- 0 ... * references

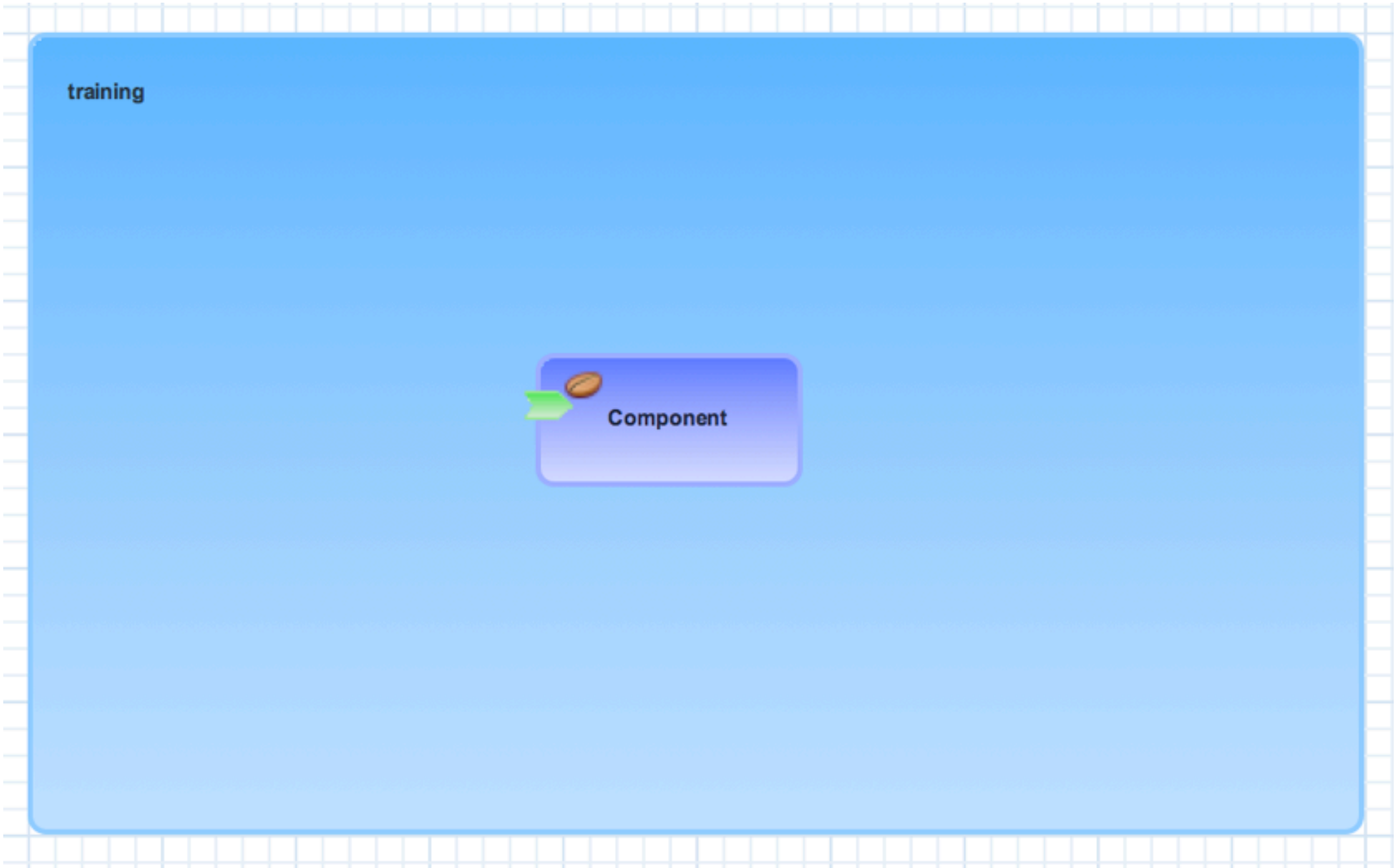
Implementation



Implementation

- Adds intelligence/action to a component
- Provides and/or consumes
- Private to an application
- First step in bottom-up approach
- Implementations are fungible

Component Service



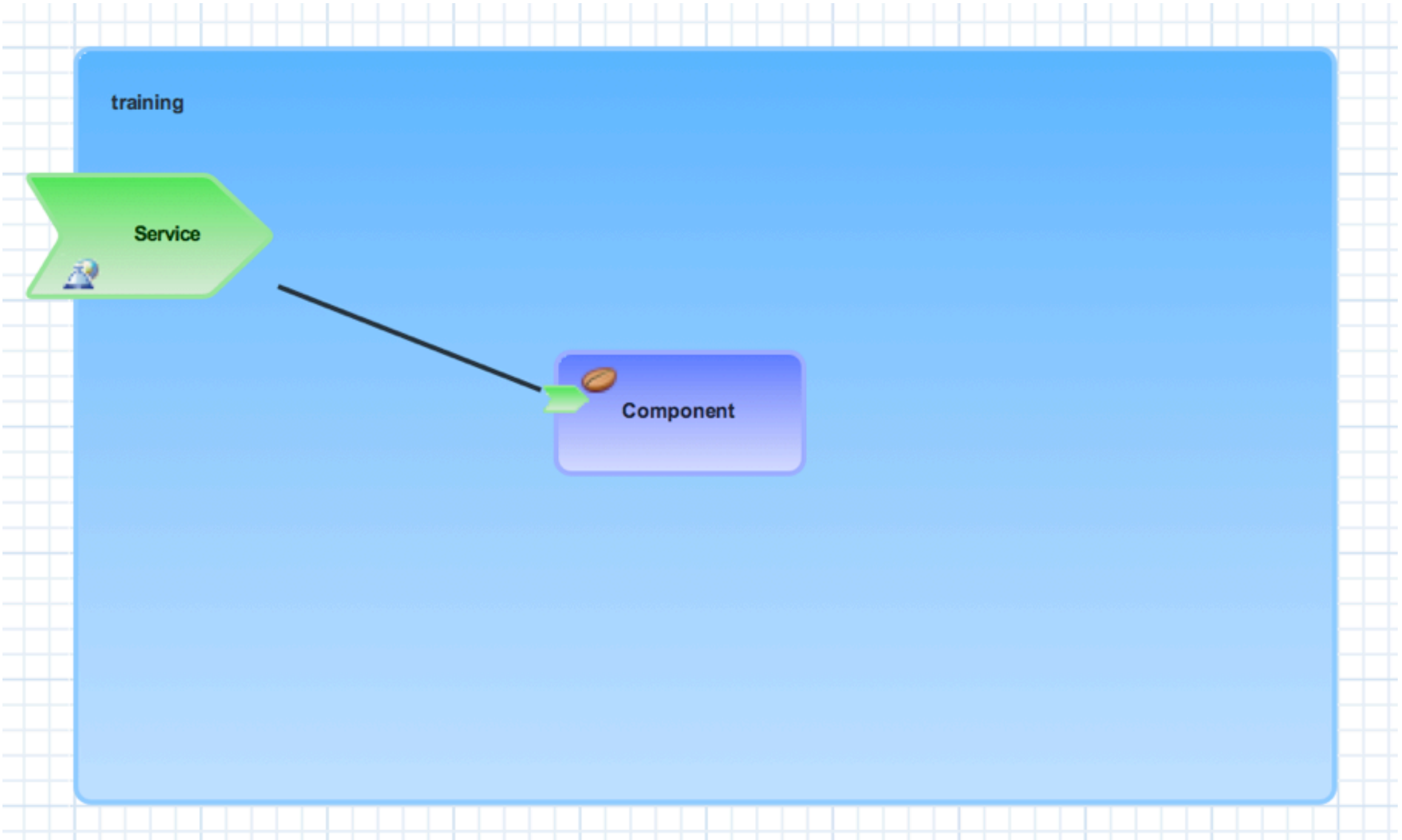
Component Service

- Exposes functionality of an implementation
- Name
- Interface
- Application private

Interface

- Service contract
- Collection of operations
- Each operation defines exchange pattern and message types
- Java, WSDL, ESB

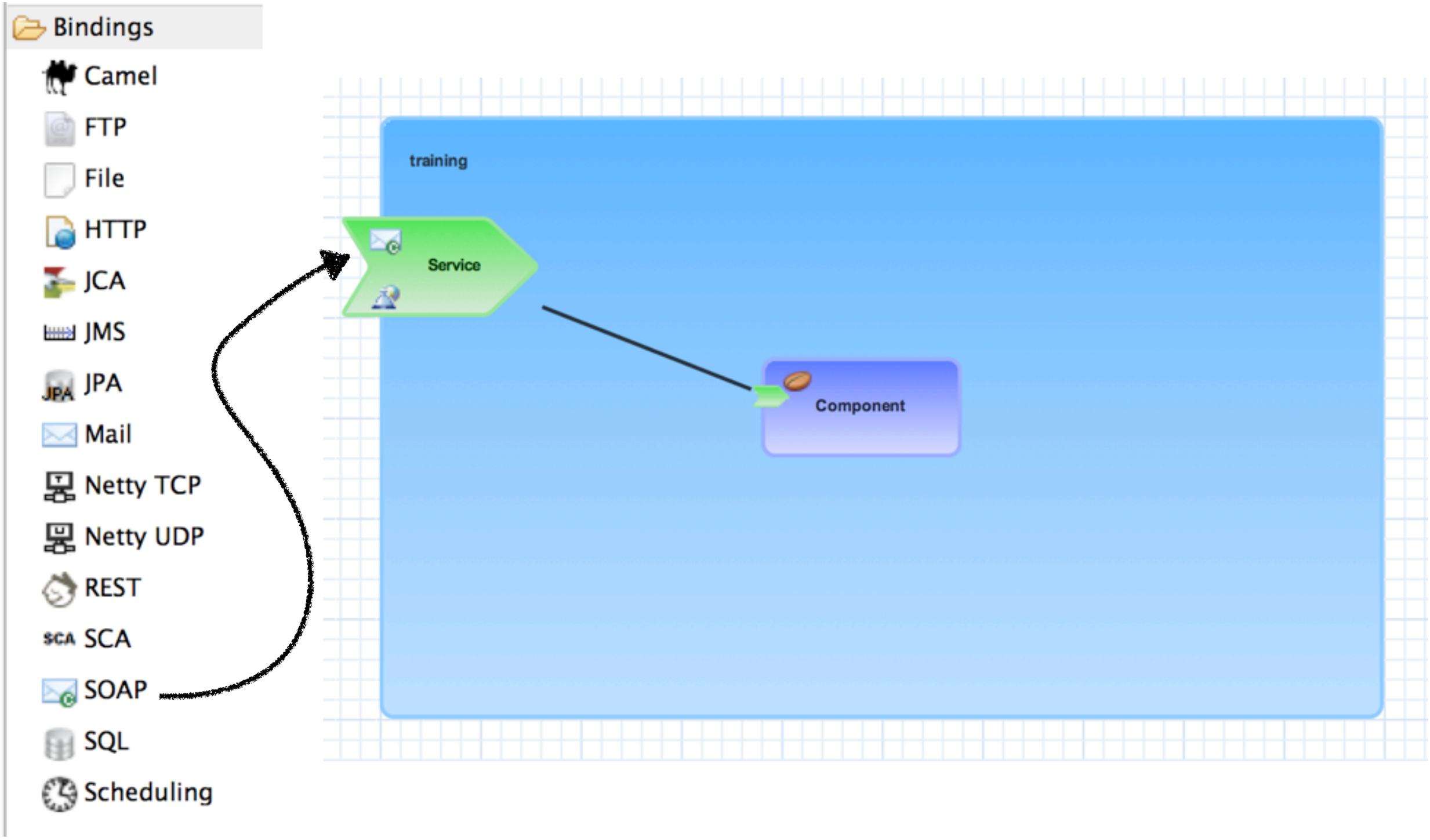
Composite Service



Composite Service

- Service available outside an application
- “Promotes” a component service
- Name
- Interface

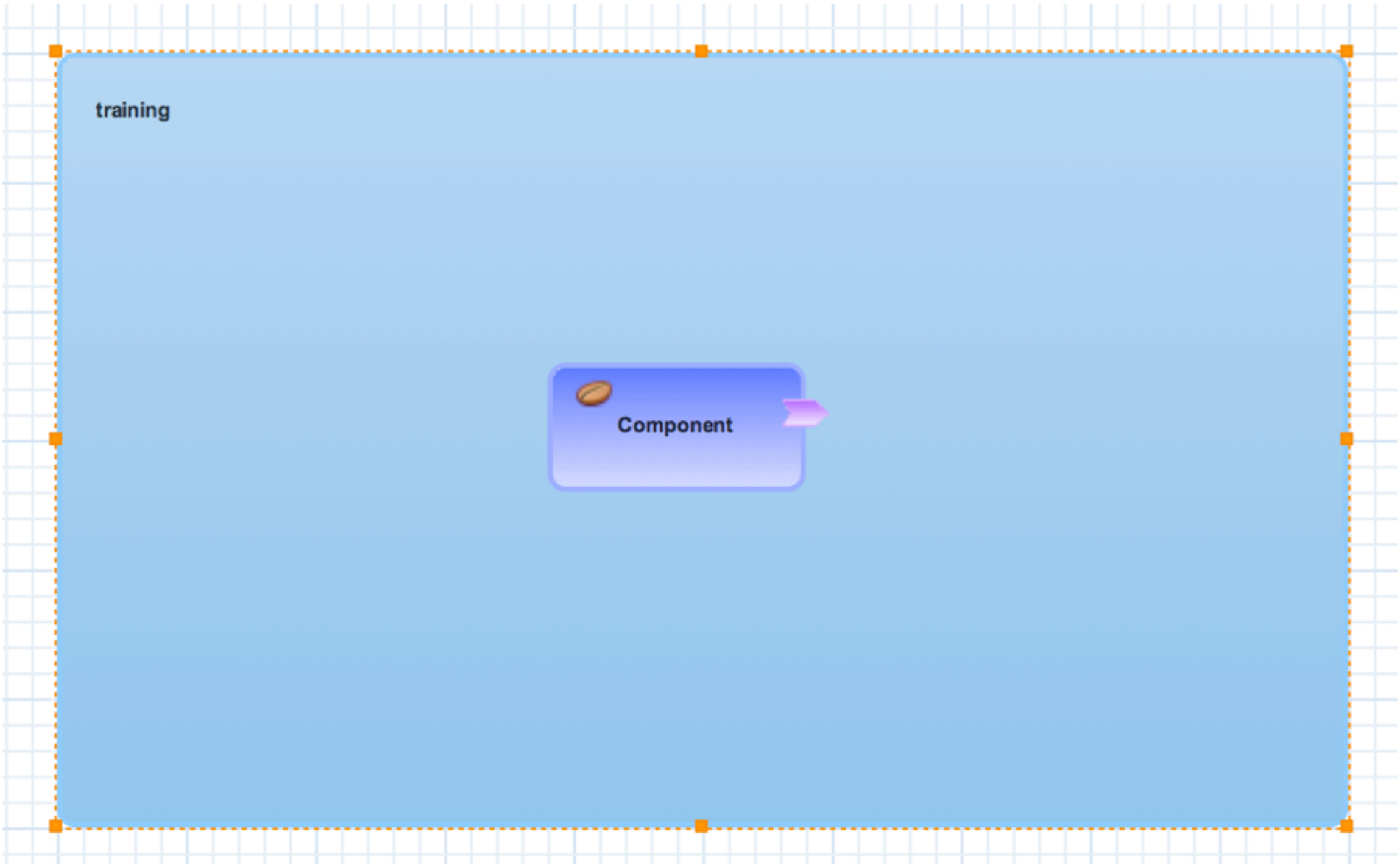
Service Bindings



Service Bindings

- How external consumers communicate with a service
- A *composite* service can have multiple bindings
- A *component* service never has bindings
- Conforms to service contract

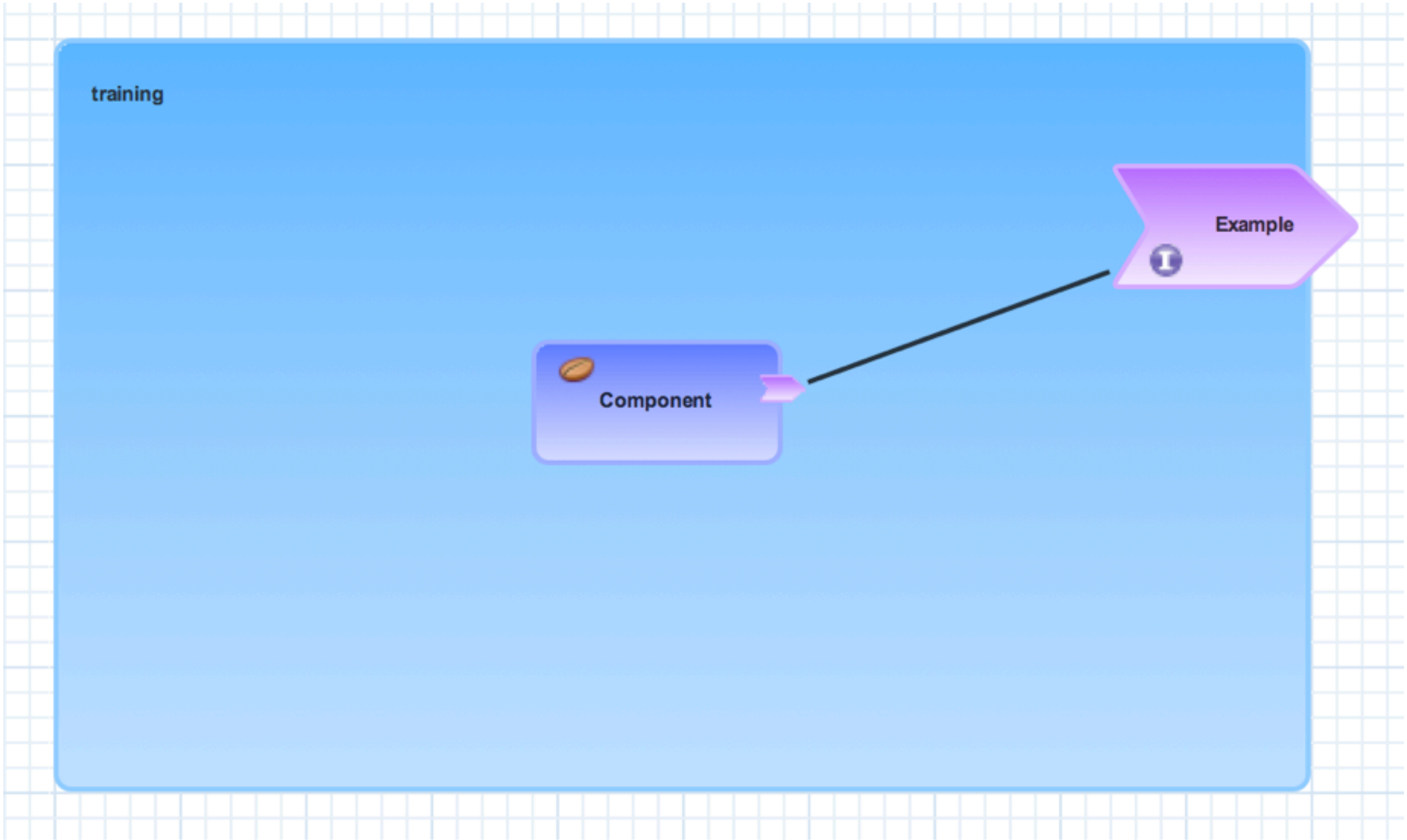
Component Reference



Component Reference

- How an implementation consumes other services
- Implementation dependency
- Name
- Interface
- Can be wired to services inside and outside an application

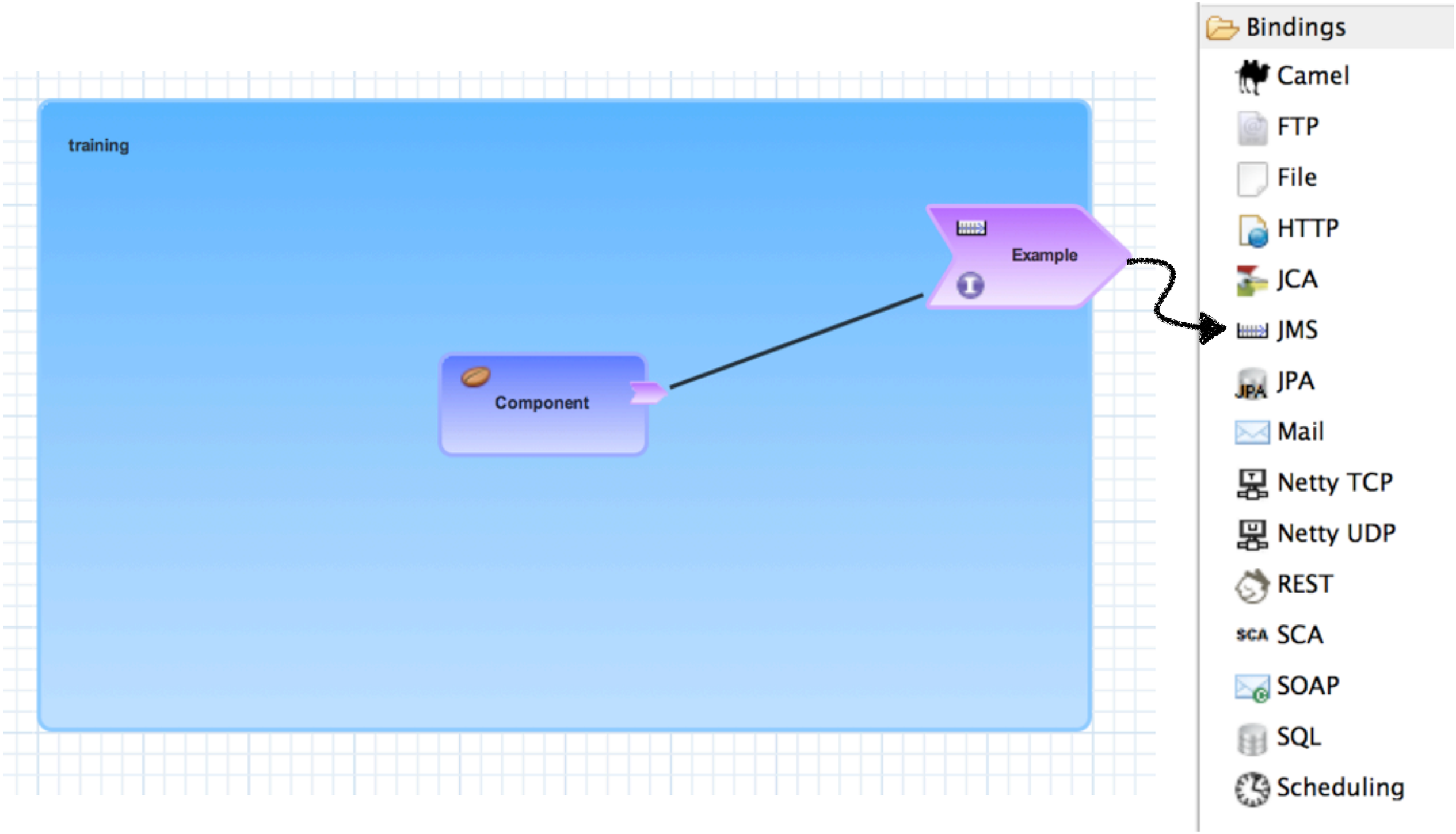
Composite Reference



Composite Reference

- Wires to services outside an application
- "Promotes" a component reference
- Application dependency
- Name
- Interface

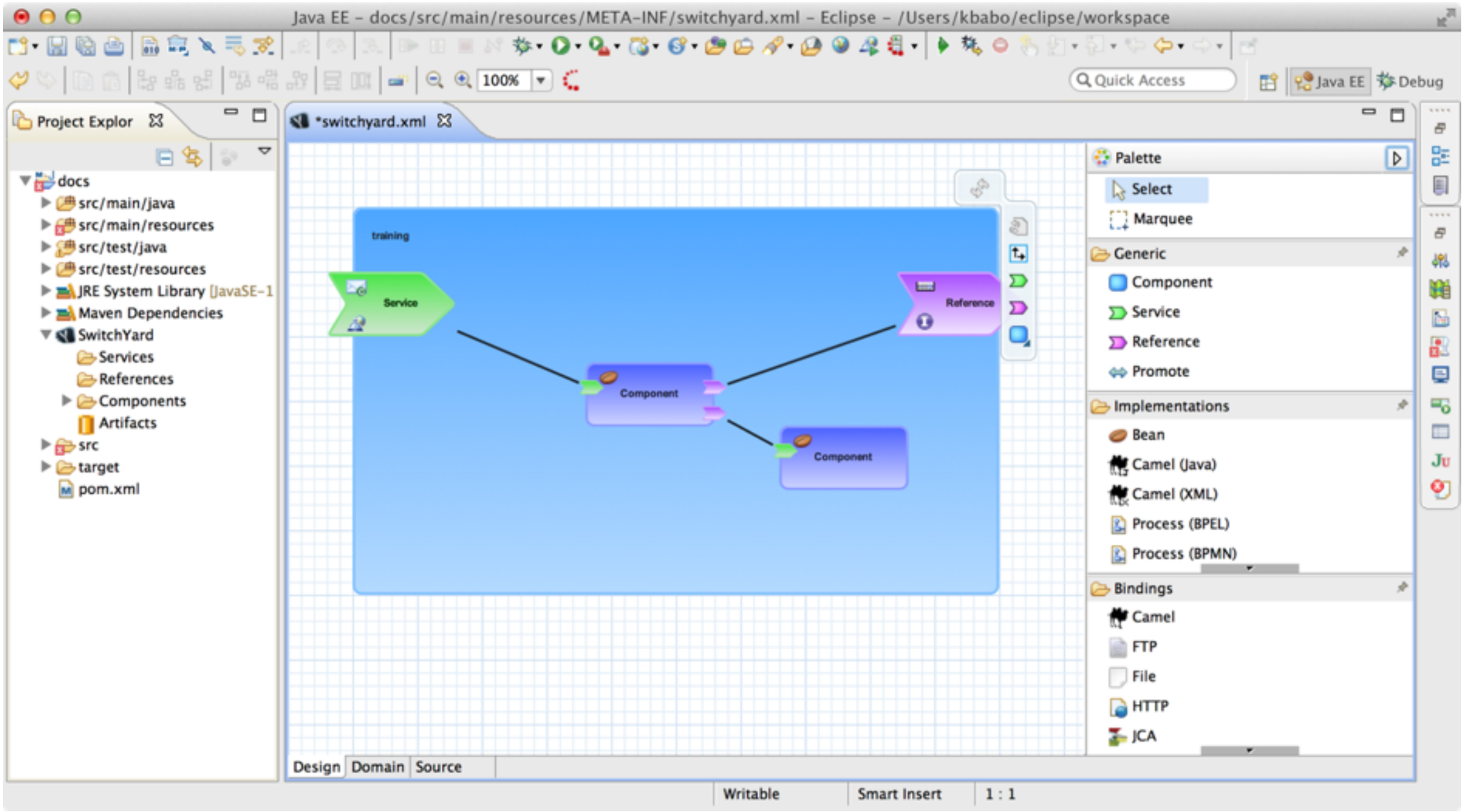
Reference Bindings



Reference Bindings

- Defines how external services are accessed
- A *composite* reference can have multiple bindings
- A *component* reference never has multiple bindings
- Conforms to reference contract

Development



Lab 1

Getting Started

- Walk through an existing SwitchYard application
- Get comfortable with the development environment
- Setup the SwitchYard runtime and deploy an application
- Poke around the admin console

Lab 2

Bean Services, Transformation, and JMS

- This lab begins with an incomplete application and walks you through the steps to complete it.
- Each step has an associated test so you can validate your application behavior as you proceed
- Lab steps:
  - Use a reference to invoke another service
  - Define transformations to/from XML and JSON
  - Add JMS bindings to services and references

Lab 3

Web Services, Camel Routing

- This lab begins with an incomplete application and walks you through the steps to complete it.
- Each step has an associated test so you can validate your application behavior as you proceed
- Lab steps:
  - Complete a proxy route between two web services
  - Add conditional routing to the proxy route
  - Configure SOAP binding for a service

Lab 4

Administration, Debugging, and General Hackery

- Cruise around the administration console and find some goodies
- Introduce message tracing to diagnose problems in poorly behaved applications
- Black belt debugging with auditors

It's Go Time

Open `lab1.pdf` to get started!