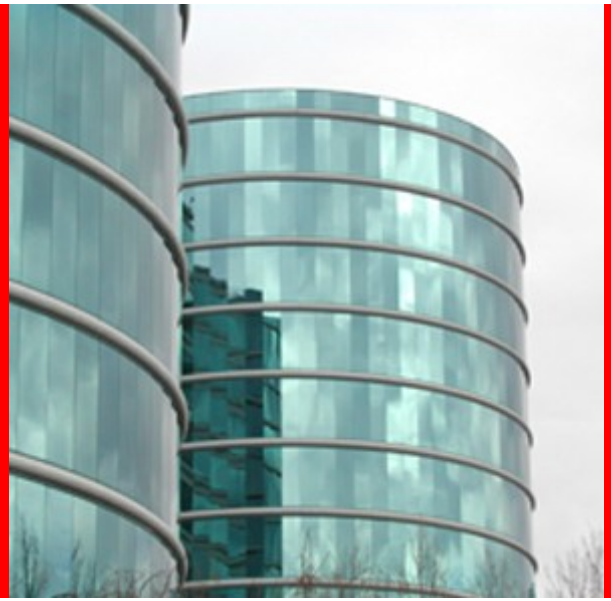

ORACLE®

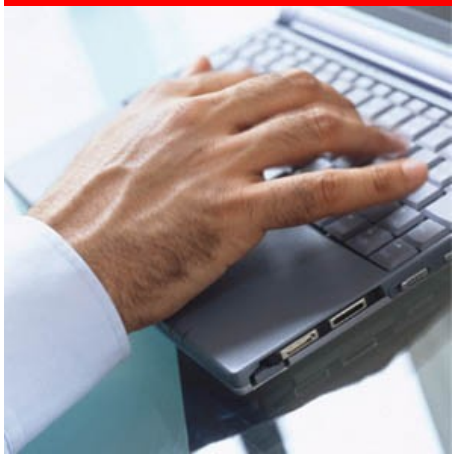


ORACLE®

Implementierung einer SOA mit der Oracle SOA-Suite 11g

Markus Lohn
Principal SOA Consultant

Agenda

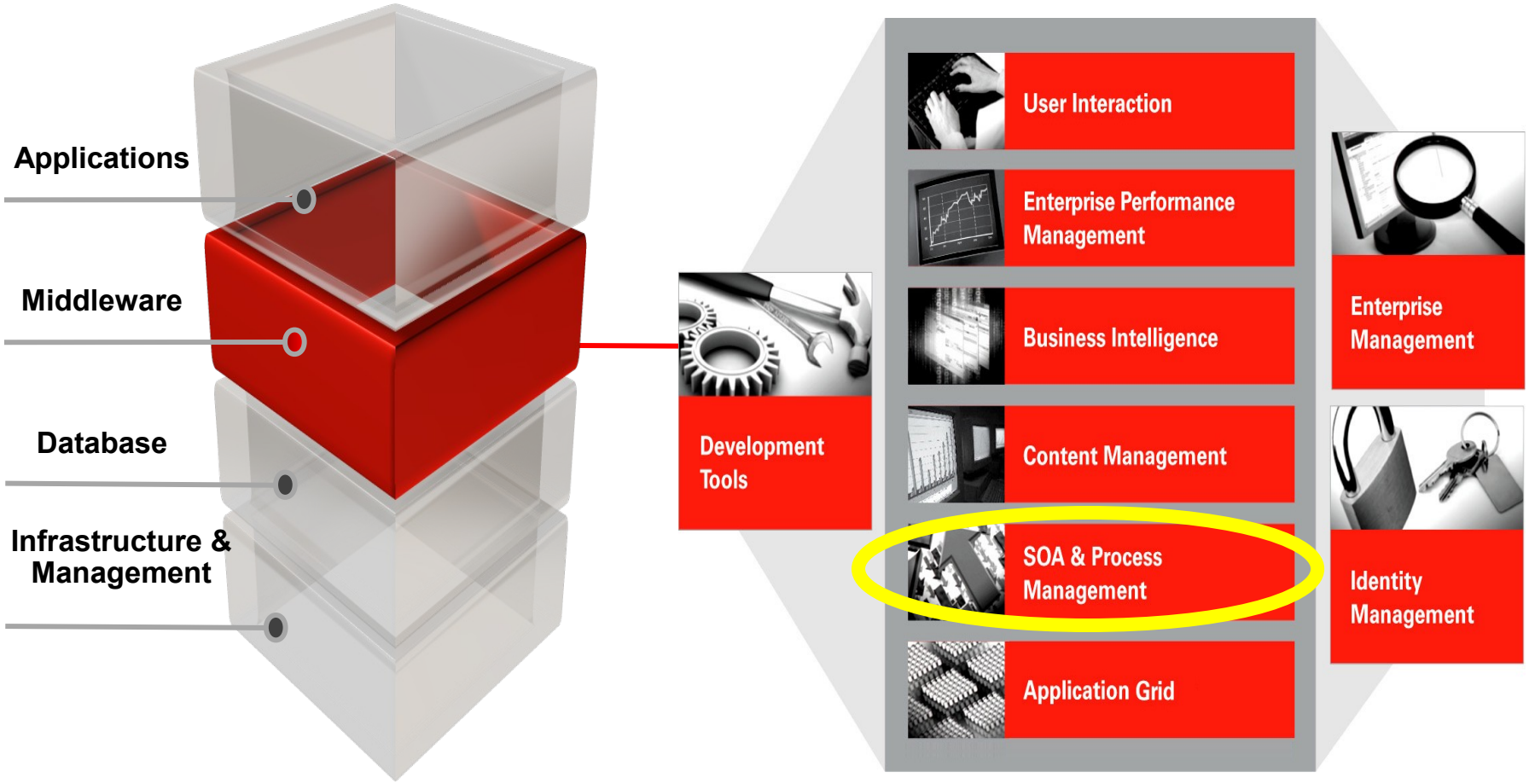


- Architektur Oracle SOA-Suite 11g
- Service Component Architecture (SCA)
- Demo: Produktbestellung
- Entwicklung eines HelloWorld-SCA

Questions

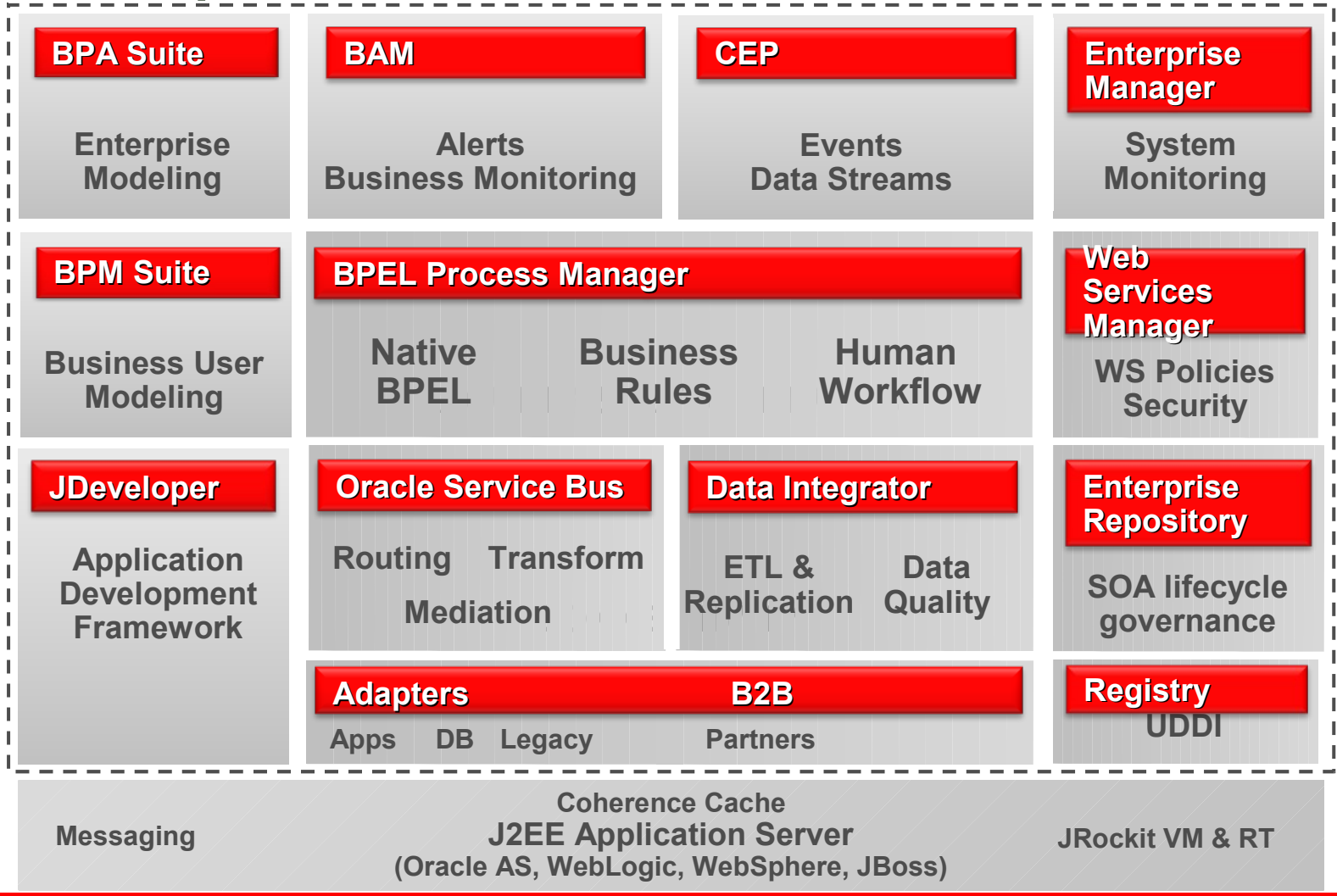
Architektur
Oracle SOA-Suite 11g

Oracle Fusion Middleware

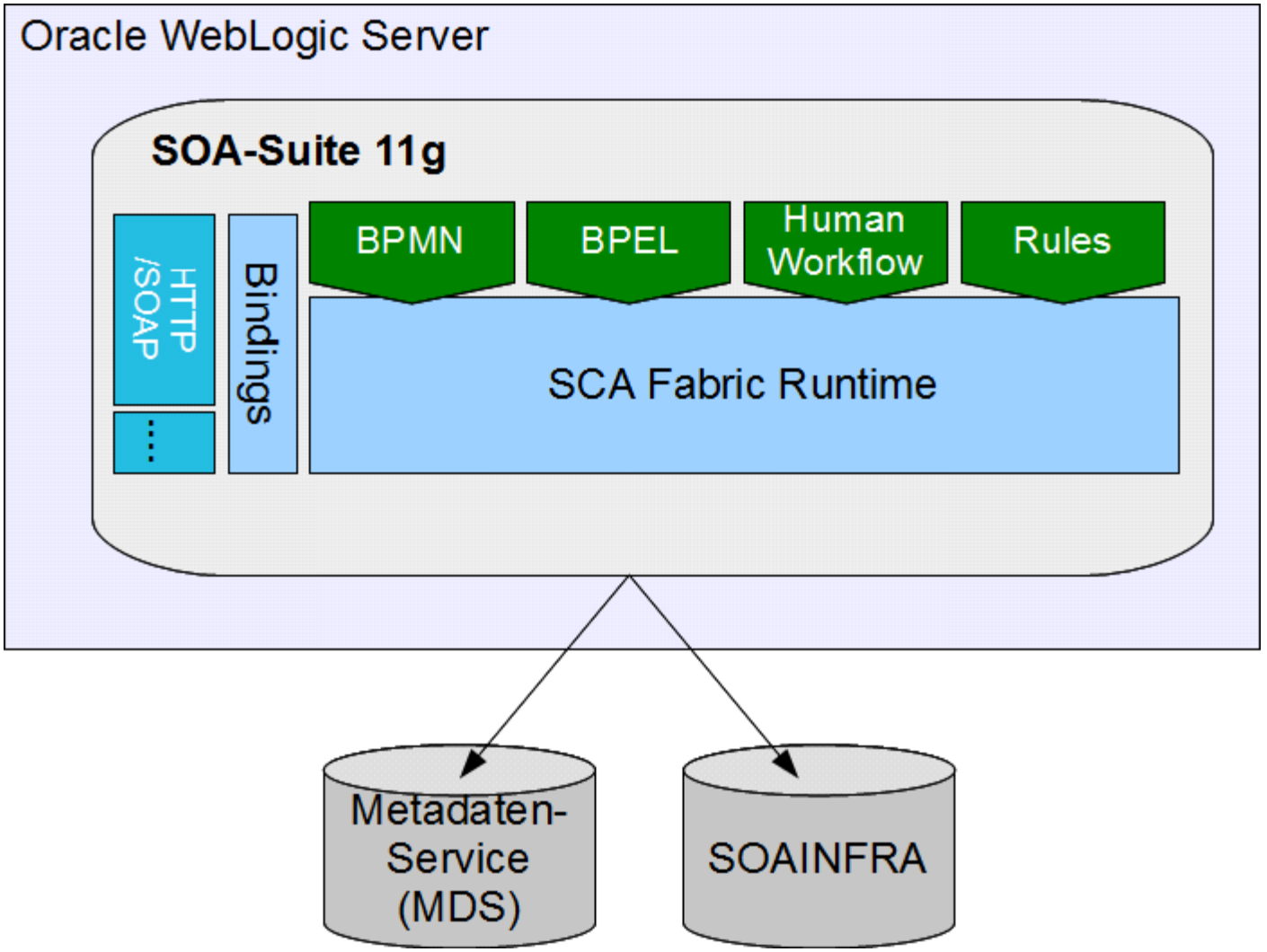


Oracle Fusion Middleware

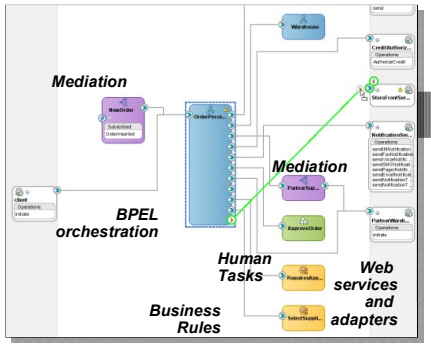
A comprehensive solution for SOA/BPM



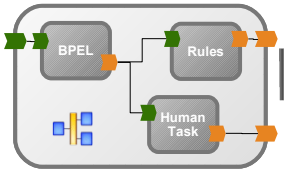
Oracle SOA-Suite 11g – Architektur (1)



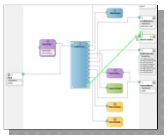
Oracle SOA-Suite 11g – Architektur (2)



SOA Composite Editor



SCA Composite



IDE

Application composers

SOA-Suite 11g

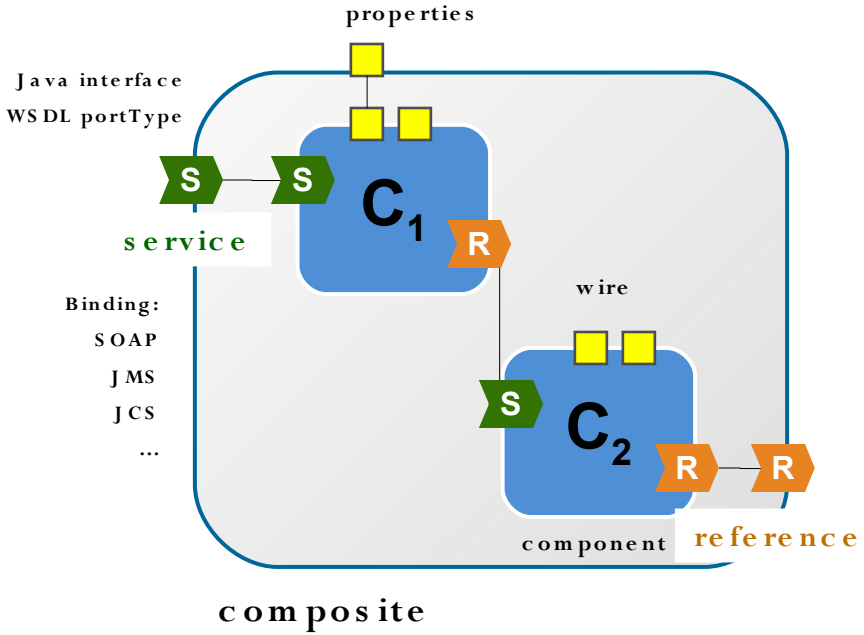
Web-based console

SOA Operations

Service Component Architecture

SCA: Service Component Architecture

Key enabler of the Service Platform



SCA terminology:

- **Composite: deployment unit**
- **Service: entry-point into composite**
- **Component: provides logic**
- **Reference: refers to external services**
- **Wire: connects services, components and references – no special semantic.**

SOA Requirements:

- ***Implement services***
- ***Assemble composite applications***
- ***Lifecycle management***
- ***Metadata management***
- ***Versioning and testing***

→ Service Component Architecture specifications:

- **language-neutral**
- **component model**
- **assembly model**

→ Specifications backed by all major players:
http://www.OSOA.org
Clear leadership from Oracle

→ Being standardized at OASIS, the international open standards consortium

SCA Design Principles

- Declarative model for service assemblies
  - Dependency resolution and configuration
  - Declarative policies
- Loose coupling between components
- Allows one to combine new and existing components into a unified system
- Vendor-, language-, and technology-neutral
- Deployment flexibility
- Separation of concern
  - SCA Components & Composites: define business logic and describe structure of business solution
  - Components and composition separated from infrastructure
    - Communication mechanisms = SCA Bindings
    - Infrastructure facilities = SCA Policy

SOA Programming Model

- **SOA Programming Model derives from the basic concept of a *service*:**
  - **A service is an abstraction that encapsulates a software function**
  - ***Developers* build services, use services and develop solutions that aggregate services.**
  - **Loose coupling**
  - **Composition of services into integrated *solutions* is a key activity**
- **SCA provides simplified programming model for SOA**

Key benefits of SCA

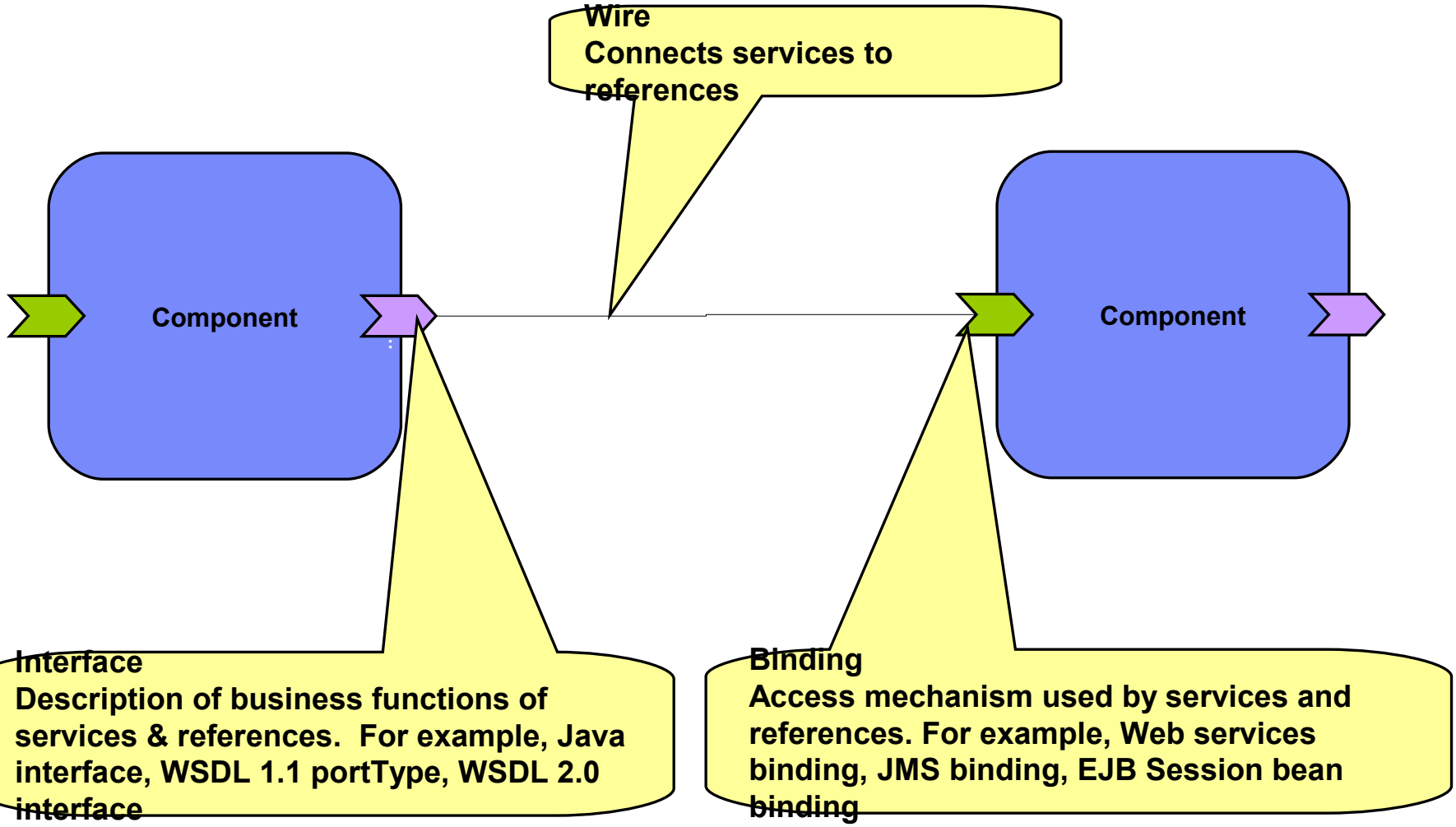
- Loose Coupling: components integrate without need to know how others are implemented
- Flexibility: components can easily be replaced by other components
- Services can be *easily* invoked either synchronously or asynchronously
- Composition of solutions: clearly described
- Productivity: easier to integrate components to form composite application
- Heterogeneity: multiple implementation languages, communication mechanisms
- Declarative application of infrastructure services

Simplification for all developers, integrators and application deployers

SCA: What it is not

- Does not model individual workflows
  - Use BPEL or other workflow technologies
- Is not Web services
  - SCA can use Web services, but can also build solutions without it
- Is not tied to a specific runtime
  - Can be distributed, heterogeneous, large or small
- Does not force use of specific programming languages and technologies
  - Encompasses many languages and technologies

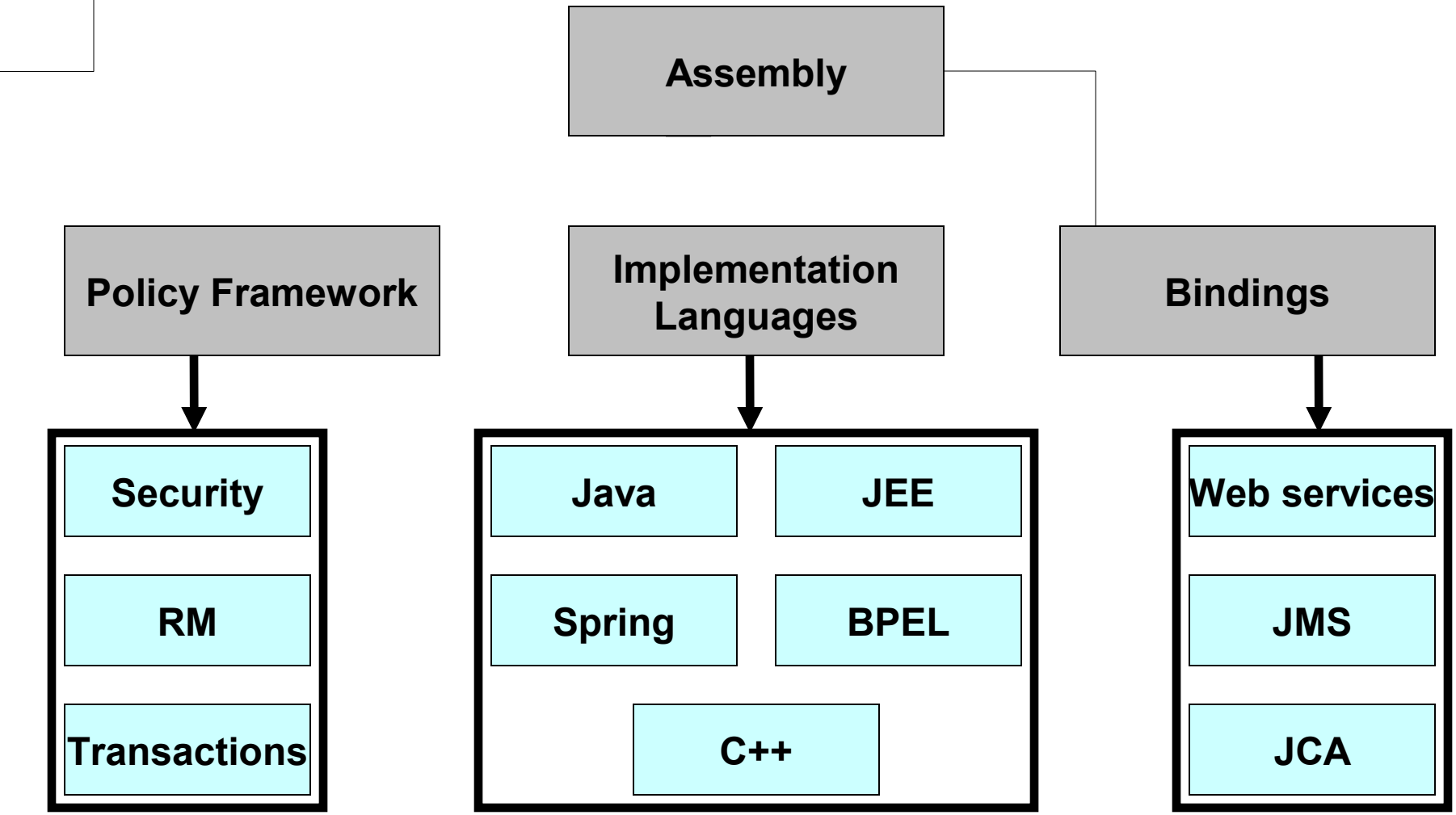
SCA Concepts: Wire, Interface, and Bindings



SCA: Standards Overview

- Started in Open SOA (OSOA) collaboration
  - Collaboration amongst 18 companies
  - Created in 2005
  - Released 0.9 version of the draft suite of specifications in November 2005
  - 1.0 final release in March 2007
- Work has been submitted to OASIS in July 2007
  - New Member Section called Open Composite Services Architecture (Open CSA)
  - Six Technical Committees created specifications for SCA
    - Assembly, Policy, Bindings, Java, BPEL, C/C++
  - Expected to release 1.1 version as a standard in 2009
  - Operating under royalty free mode

SCA Specifications



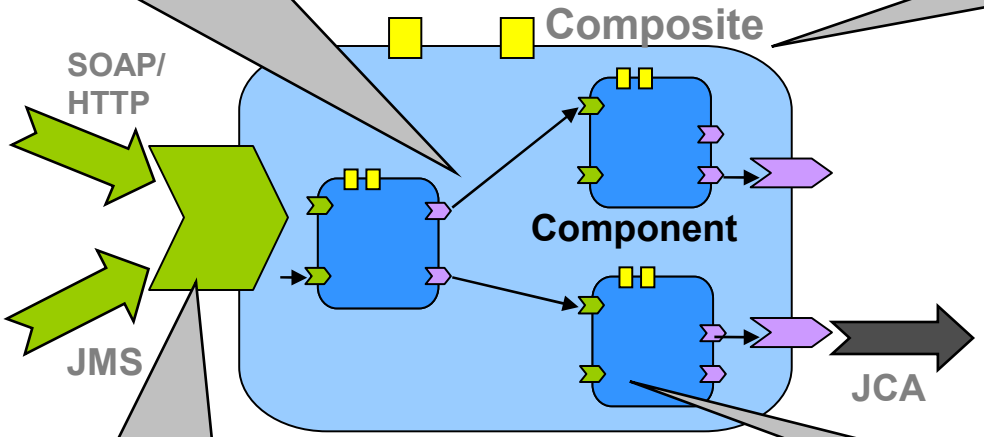
SCA Specifications (cont.)

- **Assembly Model specification**
  - Defines how to specify the structure of a composite application
- **Component Implementation specifications**
  - Defines how a service is actually written in a particular programming language
- **Binding specifications**
  - Defines possible methods of access
  - E.g. Web services, JMS, EJB
- **Policy Framework specification**
  - Defines how to add infrastructure to the service
  - E.g. Security, transactions, reliability

SCA Specifications (cont.)

How do I define, use and administer policies for non-functional aspects (QoS, etc)?
➔ SCA Policy Framework Spec

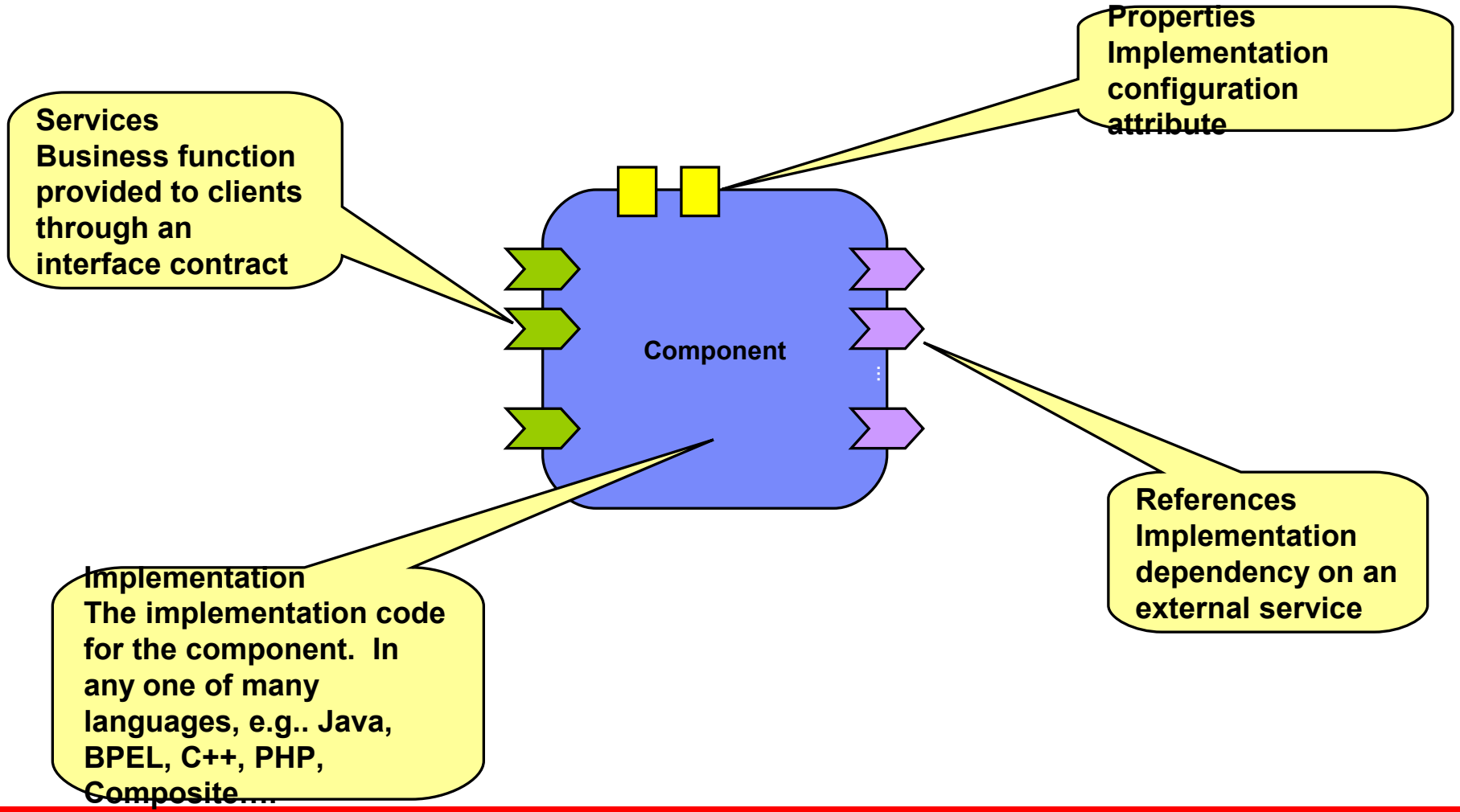
How do I define, configure and assemble components to create composites?
➔ SCA Assembly Spec



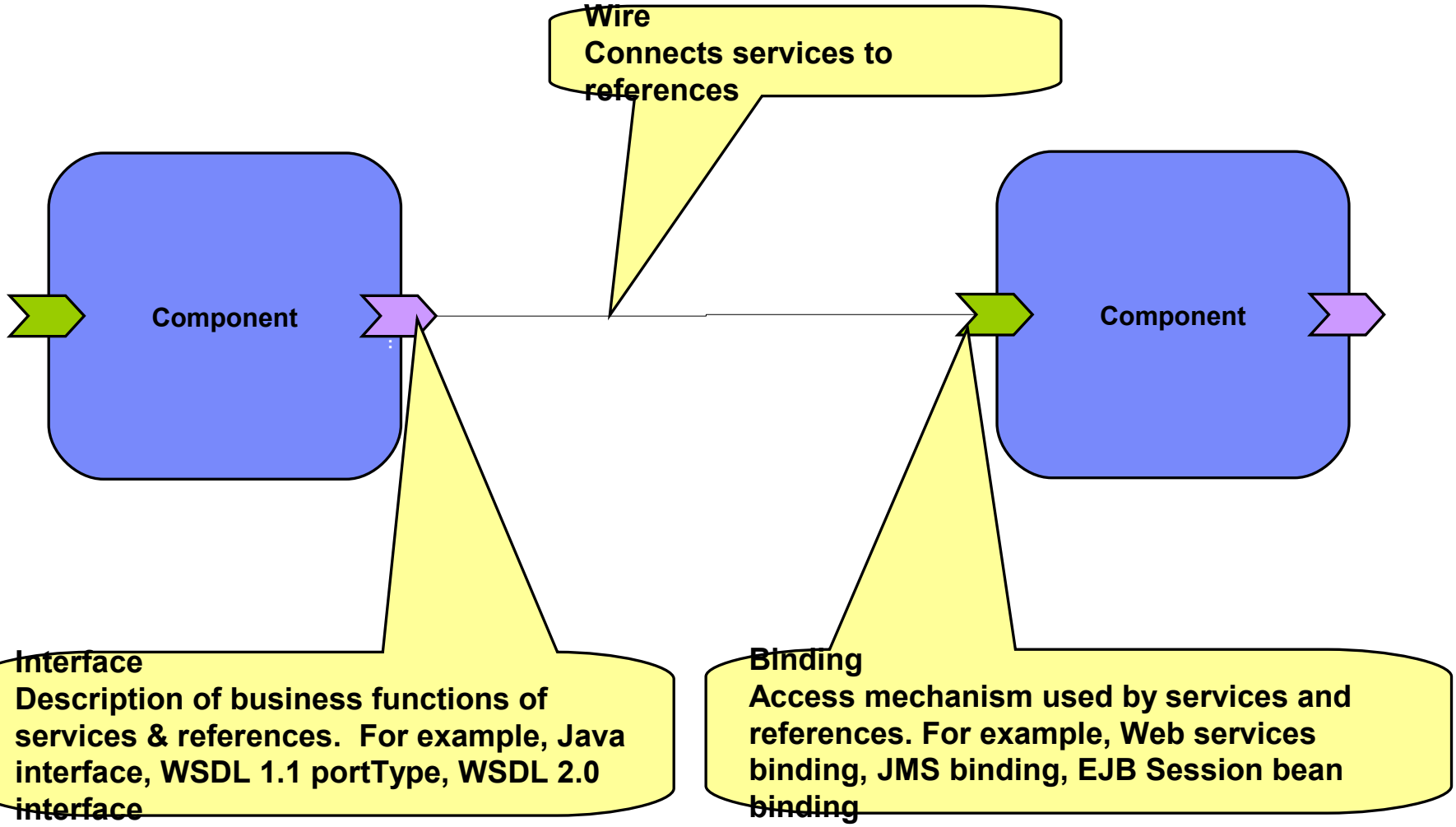
How do I configure SCA services/references to use SOAP/HTTP or JMS or JCA, …
➔ SCA WS Binding Spec, …

How do I develop SCA components in BPEL? Or in Java? Or C++, PHP,…
➔ SCA BPEL Client & Impl Spec, …

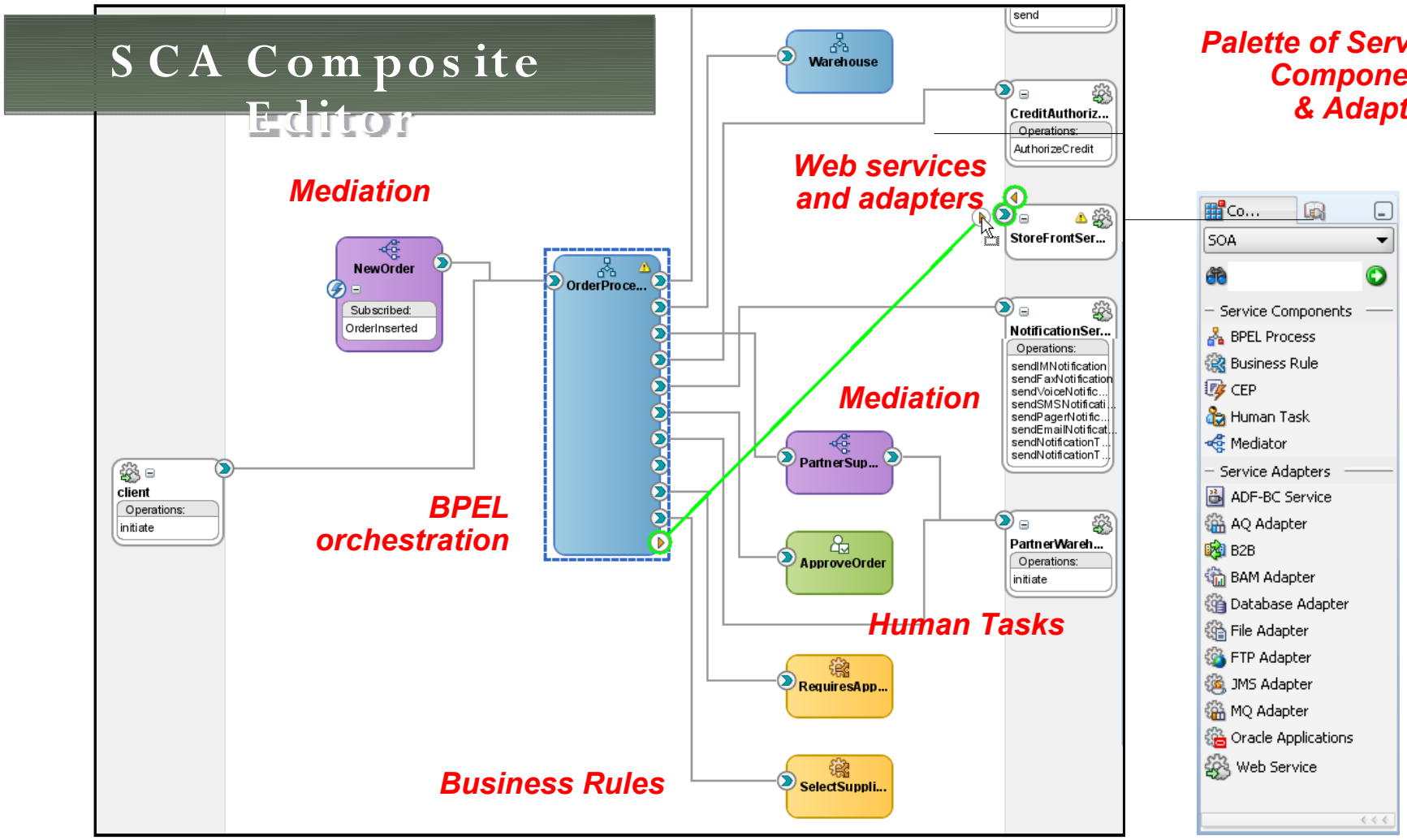
SCA Concepts: Components



SCA Concepts: Wire, Interface, and Bindings

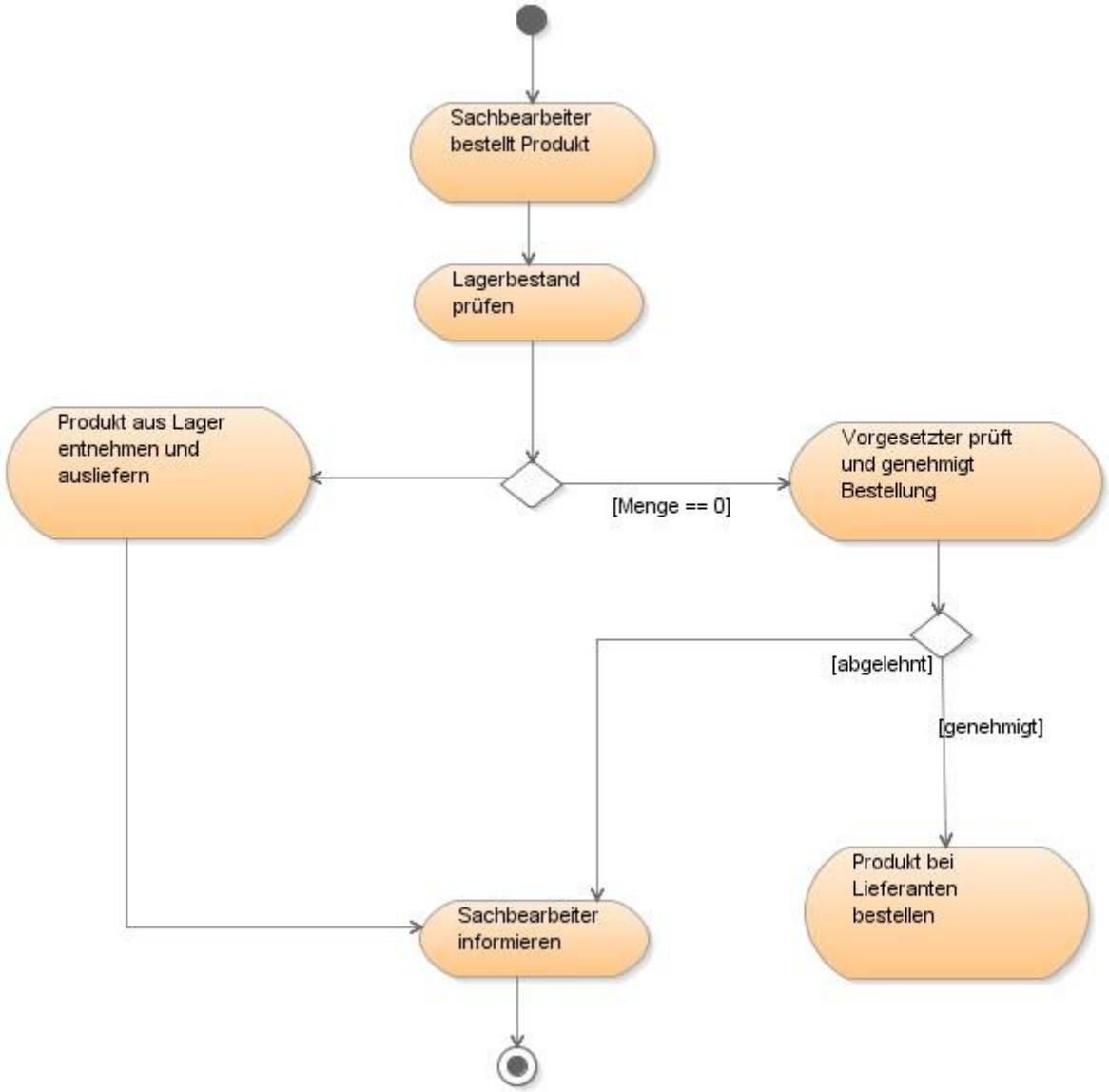


SCA Composite Editor

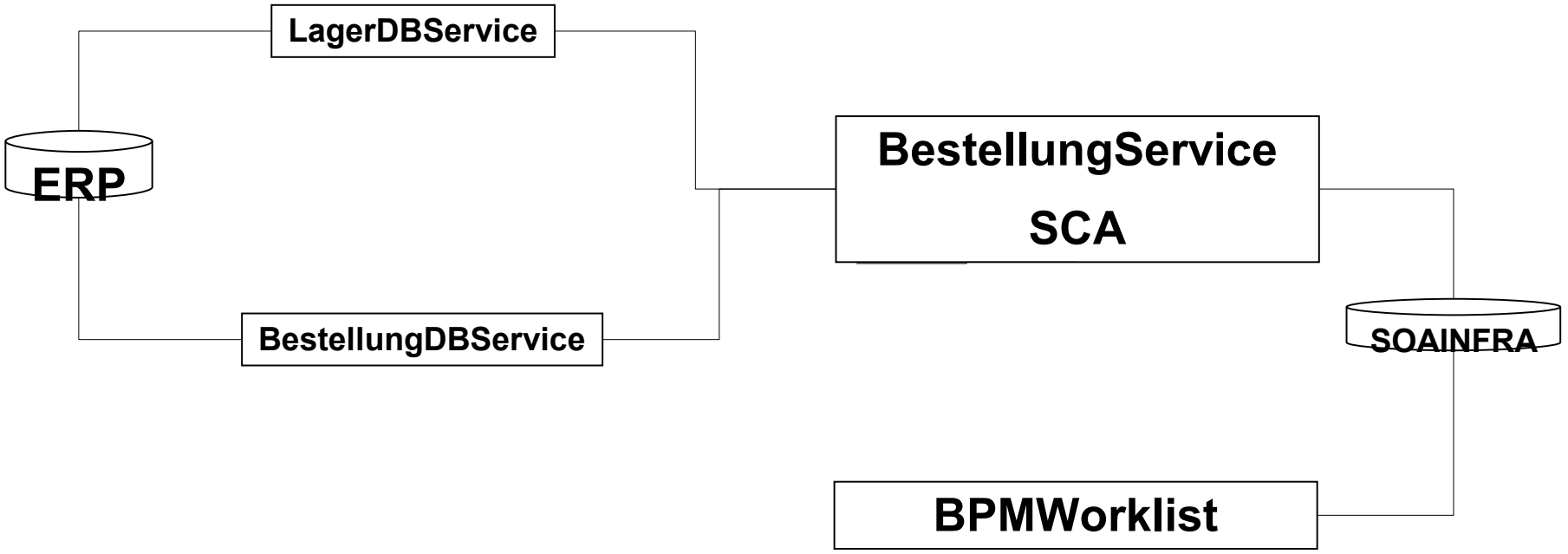


Demo: Produktbestellung

Produkt bestellen



Architektur der Demo-Anwendung



Entwicklung:

HelloWorld-SCA

HelloWorld-SCA

- Composite-Editor im JDeveloper
- Nutzung von BPEL/Mediator/Human Workflow/Rules
- Externen Service in BPEL-Prozess einbinden
- Deployment/Configuration Plan
- FMW Control zum Starten und Überwachen zeigen
- MDS


ORACLE®

ORACLE IS THE INFORMATION COMPANY