

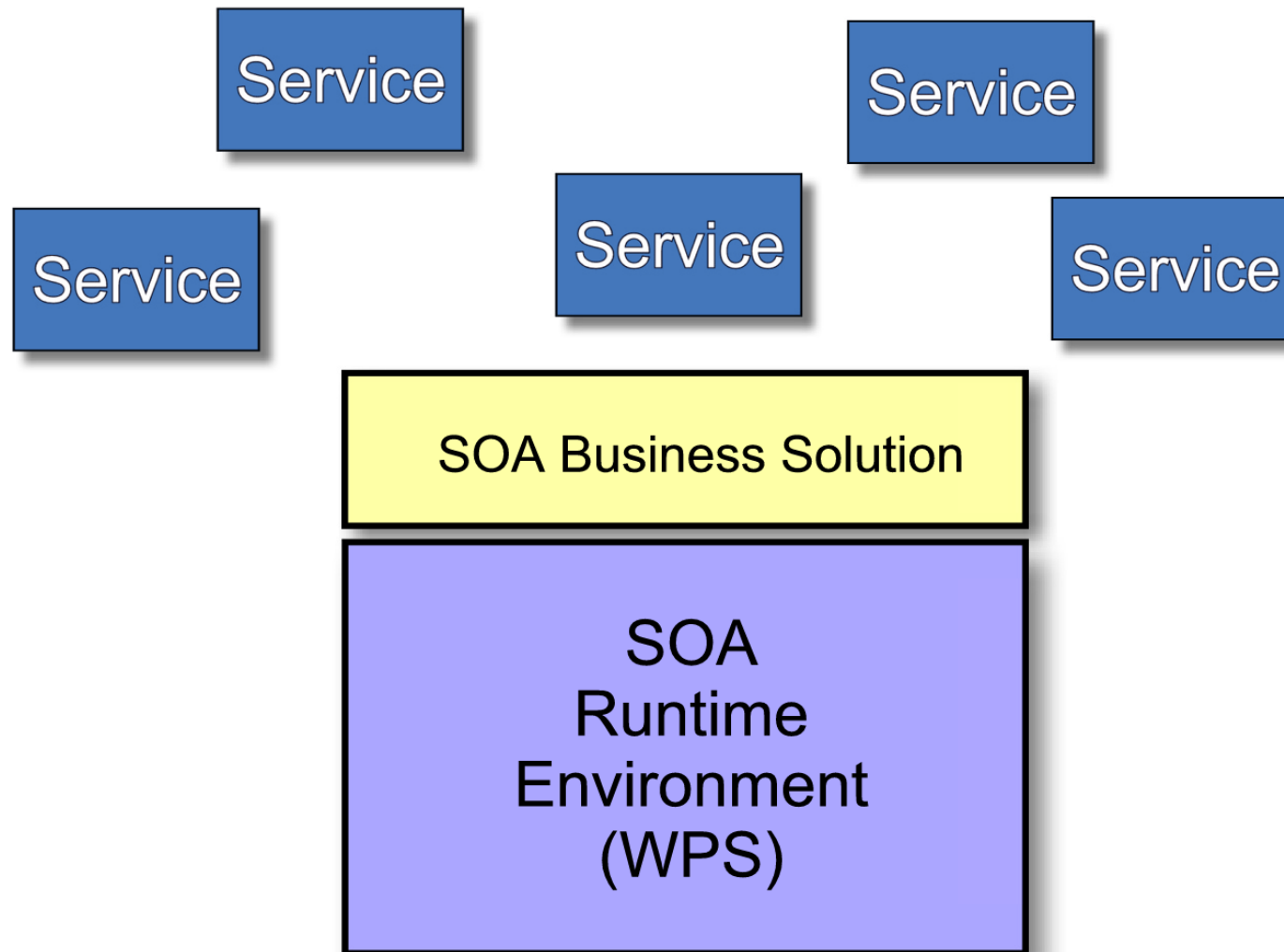


| IBM ISV and Developer Relations

Develop a new web service in an SOA environment

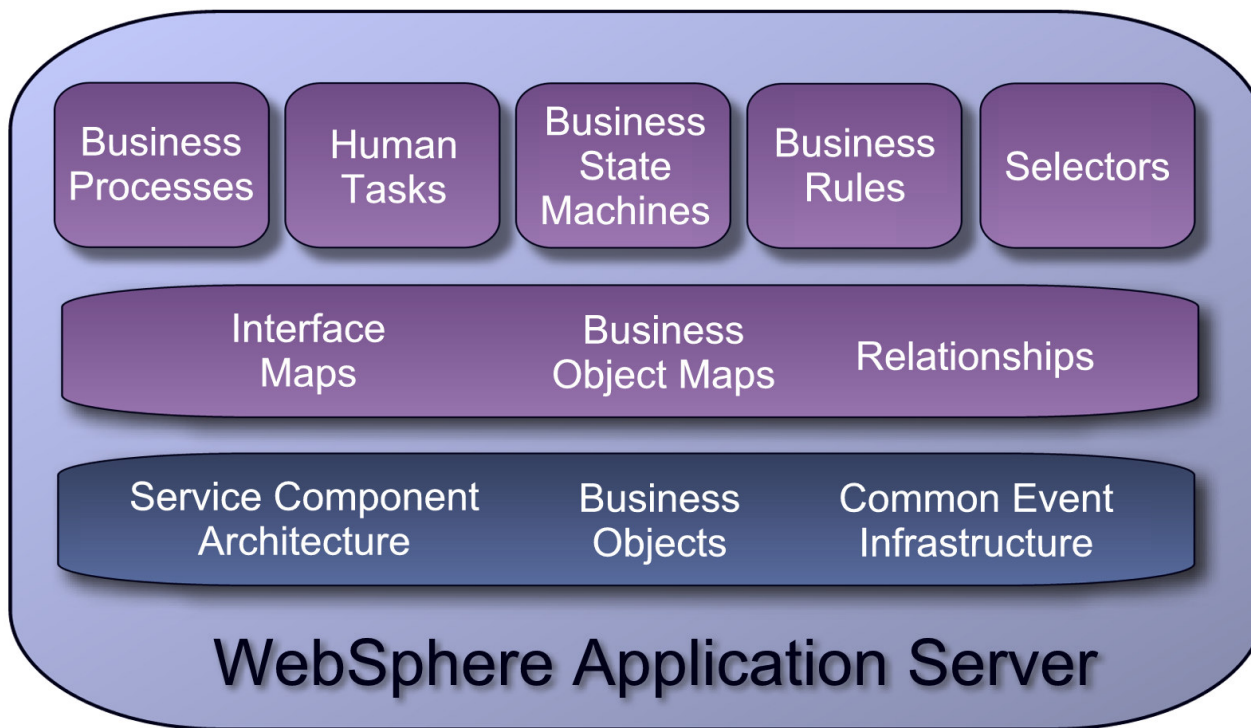
Francis Geysermans
eArchitect
La Gaude

What is WebSphere Process Server? ...



WebSphere Process Server ...

Components



Service Components

are the added-value components

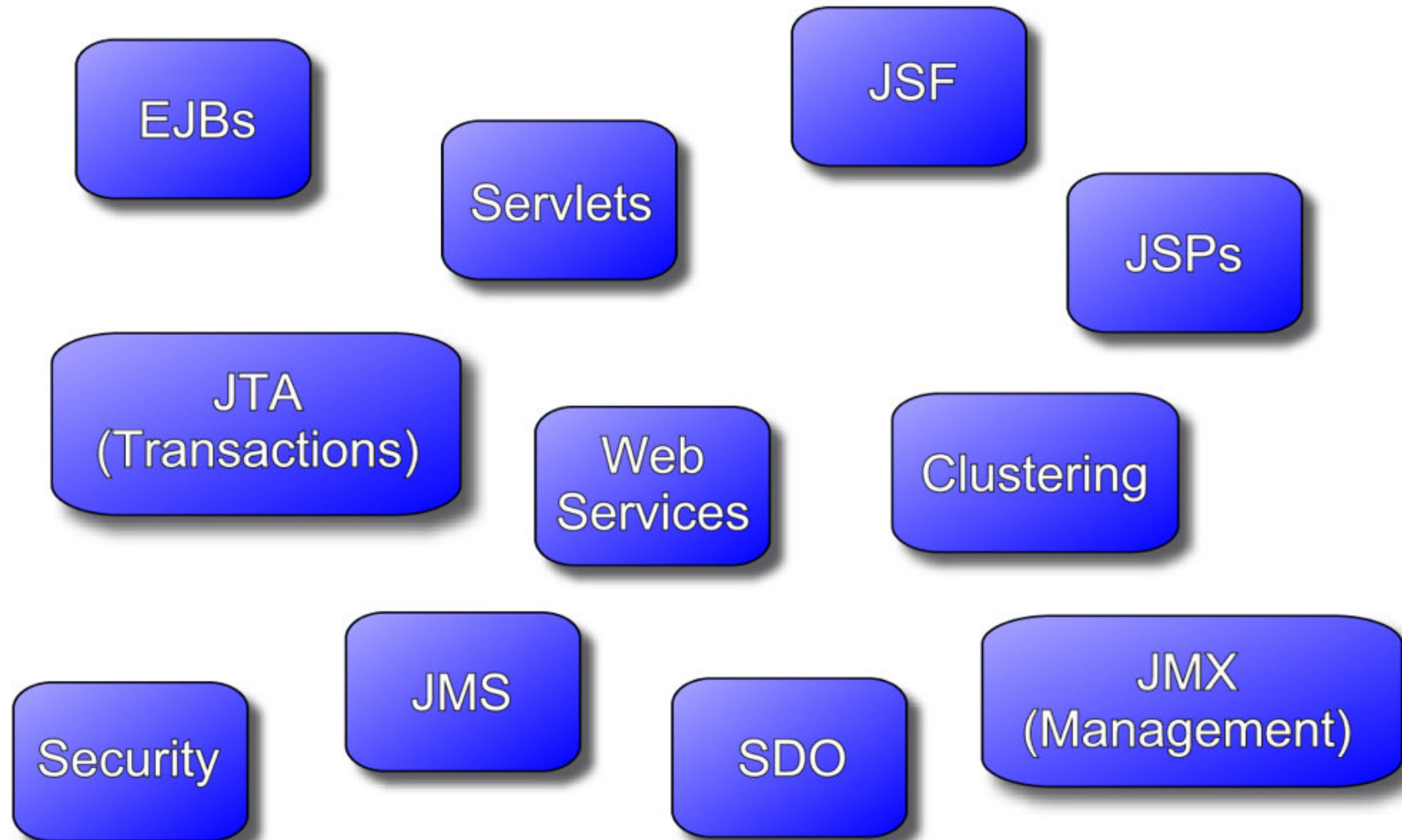
Supporting Services

simplify common integration tasks

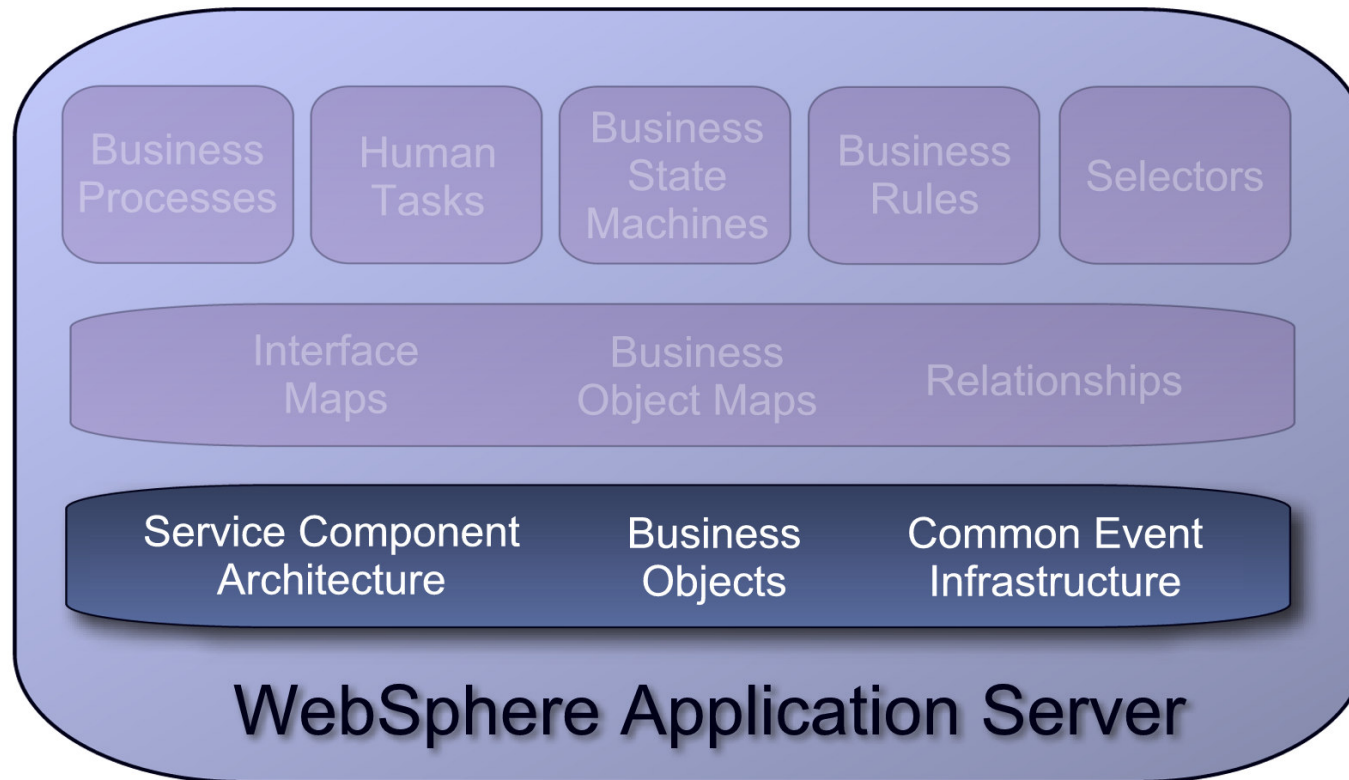
SOA Core is the foundation technology

WebSphere Application Server V6 foundation

Based on WebSphere Application Server ...

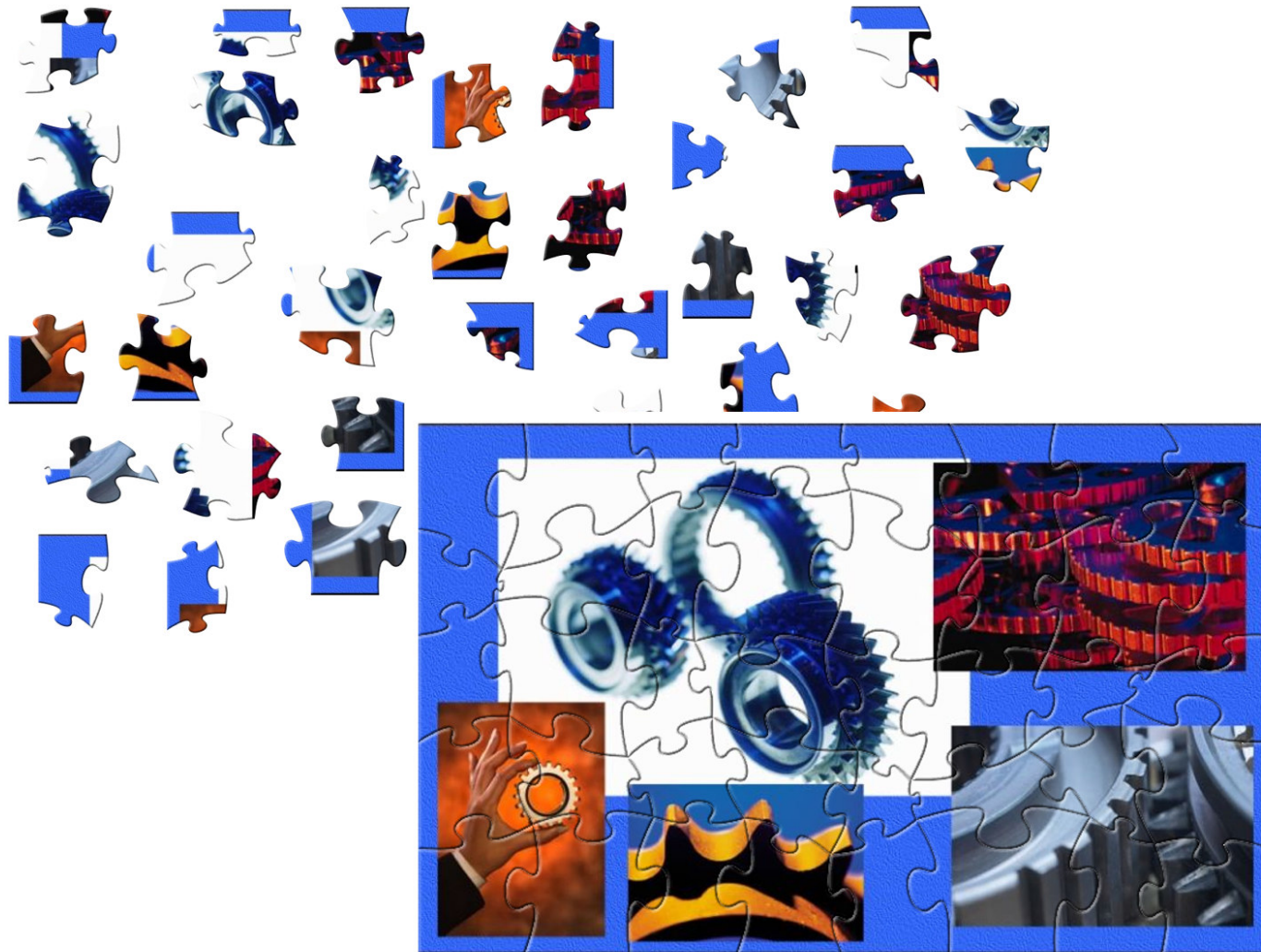


WebSphere Process Server ... *SOA Core*



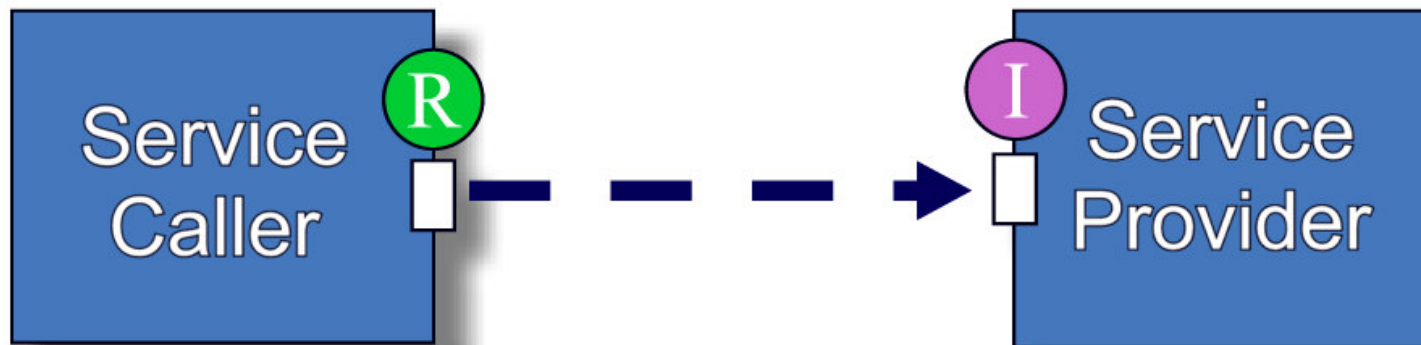
SOA Core is the foundation technology

Assembling the Services ...



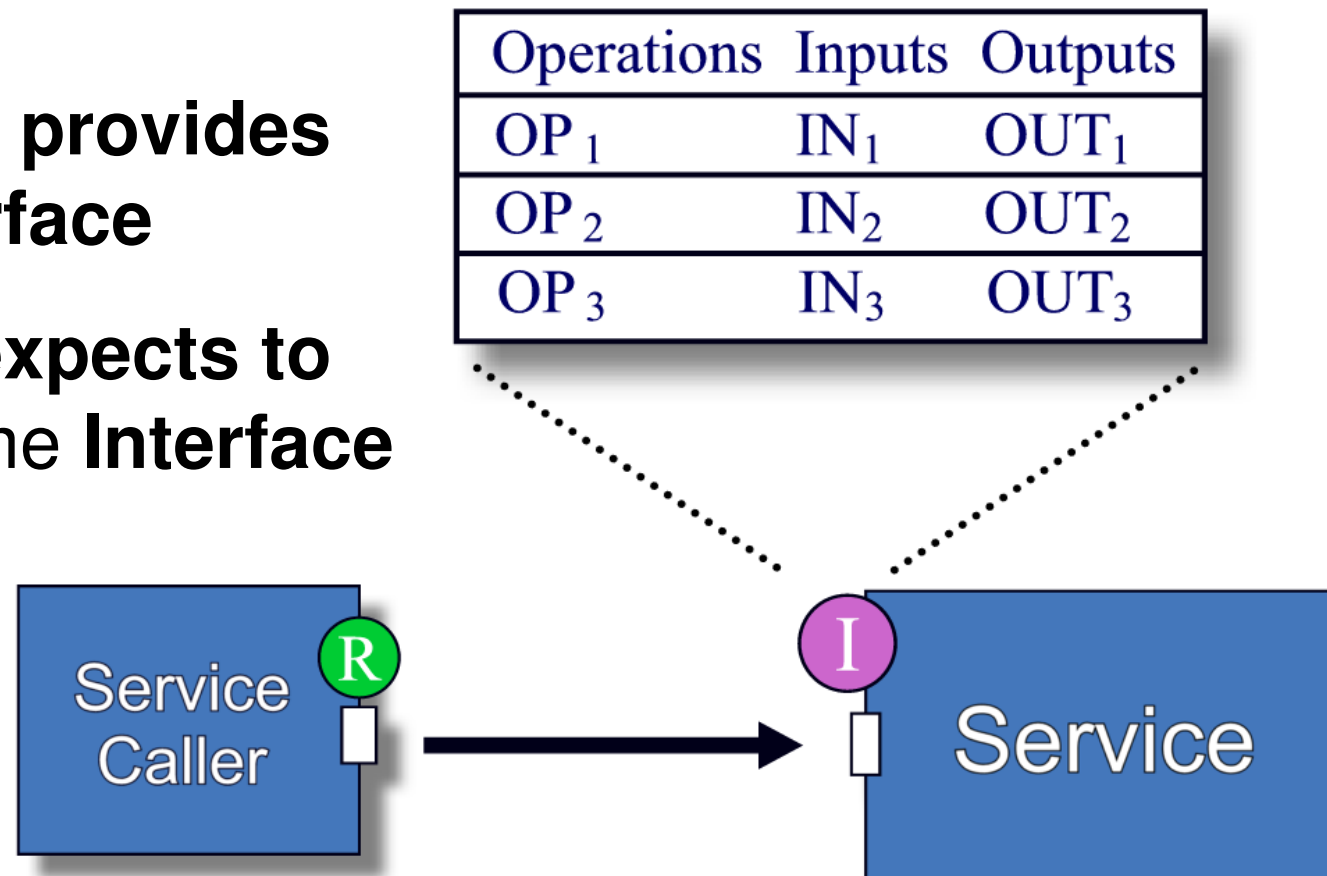
A Basic Service ...

- **Service Provider provides an Interface**
- **Service Caller invokes the Service Provider's Interface**
- **Logical components:**
 - Interface – Contract for the Service
 - Reference – Service needed in order to execute



The Logical Interface ...

- **Service provides an Interface**
- **Caller expects to call same Interface**



The Problems ...

- **Coupling of Callers to Services is too tight**
 - How do I replace one Service with another?
 - What if the provider's Interface changes?
 - Protocol and other implementation details exposed!

- **How do I specify extra qualities of Service?**
 - Transactionality
 - Security
 - Others...

- **How do I manage which Services invoke which other Services?**

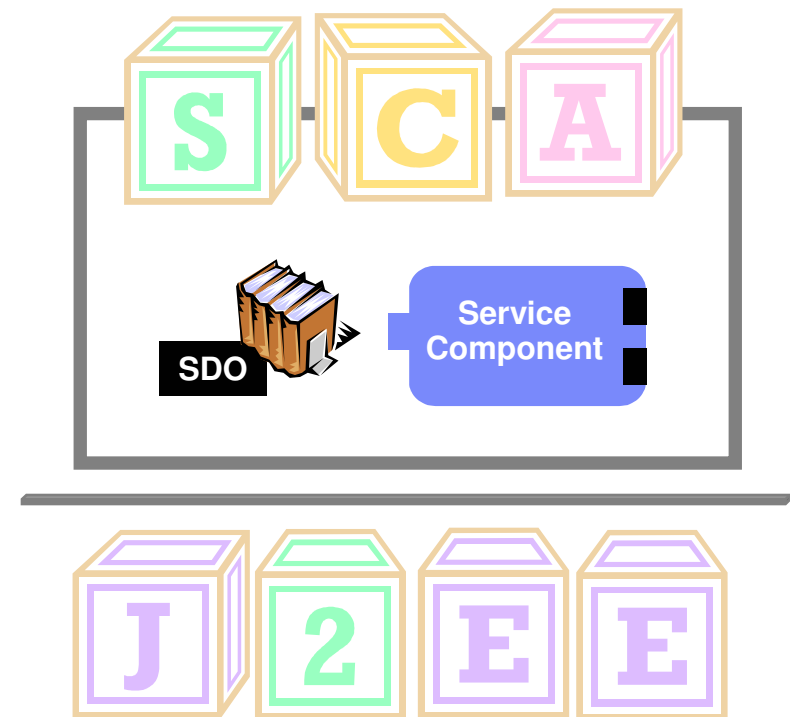
WPS Programming Model

▪ SCA - **Service Component Architecture**

- Based on J2EE
- Service Components
- Service Data Objects
- Programming model for building business processes in a service oriented way

▪ Value

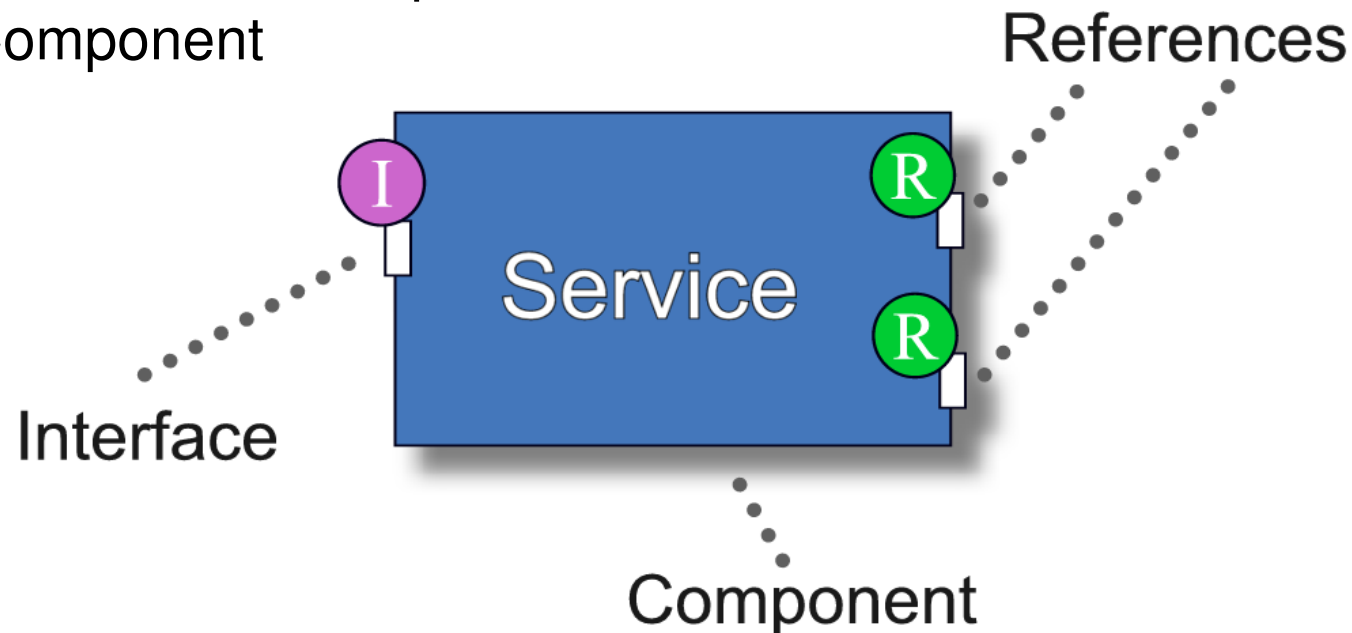
- Lowers the barriers for a developer to enter the world of SOA and become productive
- J2EE and Web Service skills not required to create sophisticated business applications
- Makes the experienced J2EE developers more productive.



Service Component Architecture ...

■ Core Concepts ...

- Services are called Components
- Each Component has an Interface
- A caller of a Component has a Reference to that Component



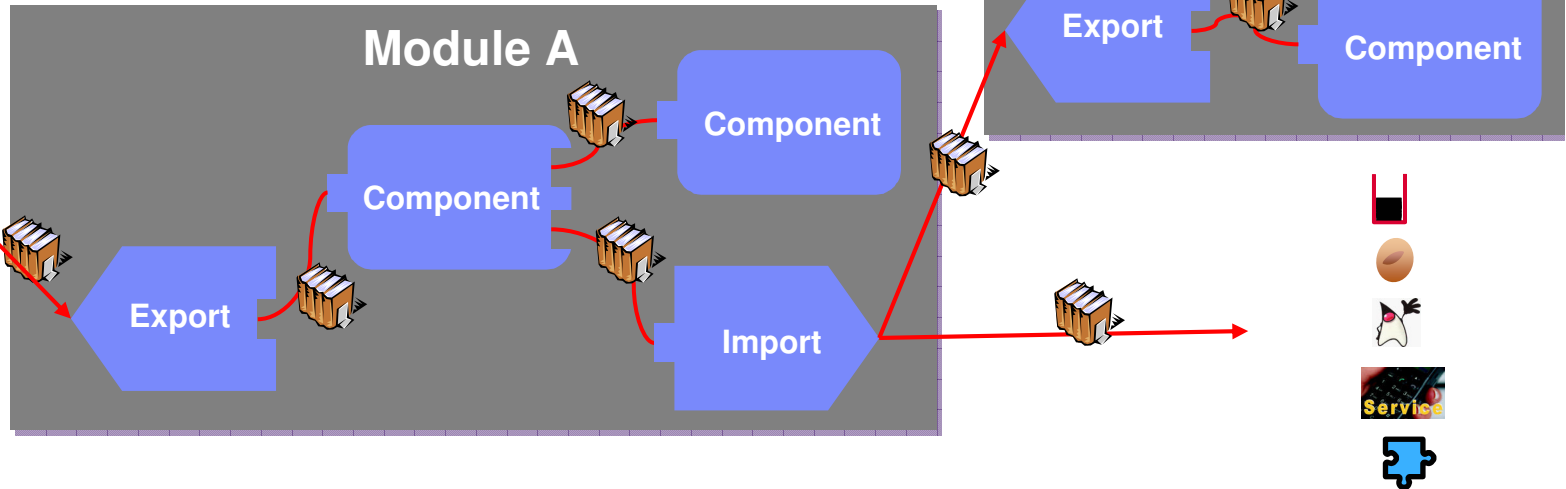
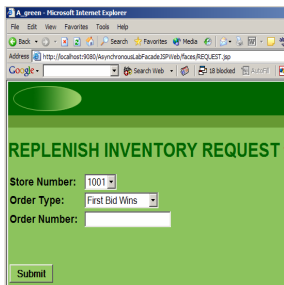
SCA In General

- **SCA is a universal model for *business services* that publish or operate on *business data*. Service Data Objects (SDO) provides the universal model for *business data*.**
- **An *SCA service is a service component* that runs on a SCA run-time.**
 - An *SCA interface* is defined by a Java interface or WSDL portType. Arguments and return are described using Java classes, simple Java types or XML schemas. Arguments described in XML schema are exposed to programmers as SDOs.
 - A *Service can be implemented* in a range of languages (e.g. Java, BPEL). When implementing a service component, the focus is on the business purpose, less on implementation technology.
- **An *SCA module* is a set of components. Modules have imports (things they need) and exports (interfaces they expose).**
- **SCA and non-SCA services can use other service components in their implementation. They do not hard-code which other services they use. They declare “soft-links” called *service references*.**
- ***Service wires* resolve service references. The SCA wiring allows you to create SCA applications by component assembly.**

Service Modules


- Modules are contain “wired” Service Components
- Service Components use SDOs for data
- Solutions are a collections of Modules

Web Client

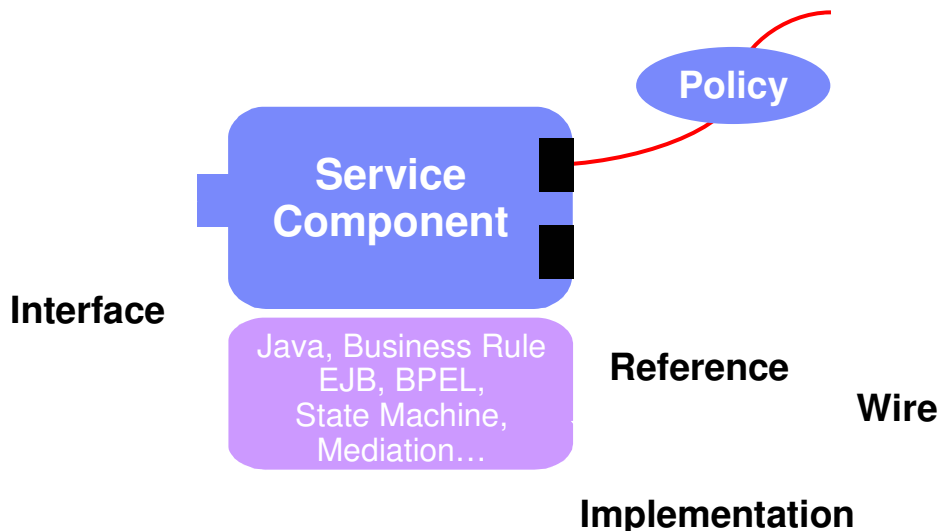


 **Wires**
 **Interfaces**
 **References**
 **SDO**

Service Components


 Service Component

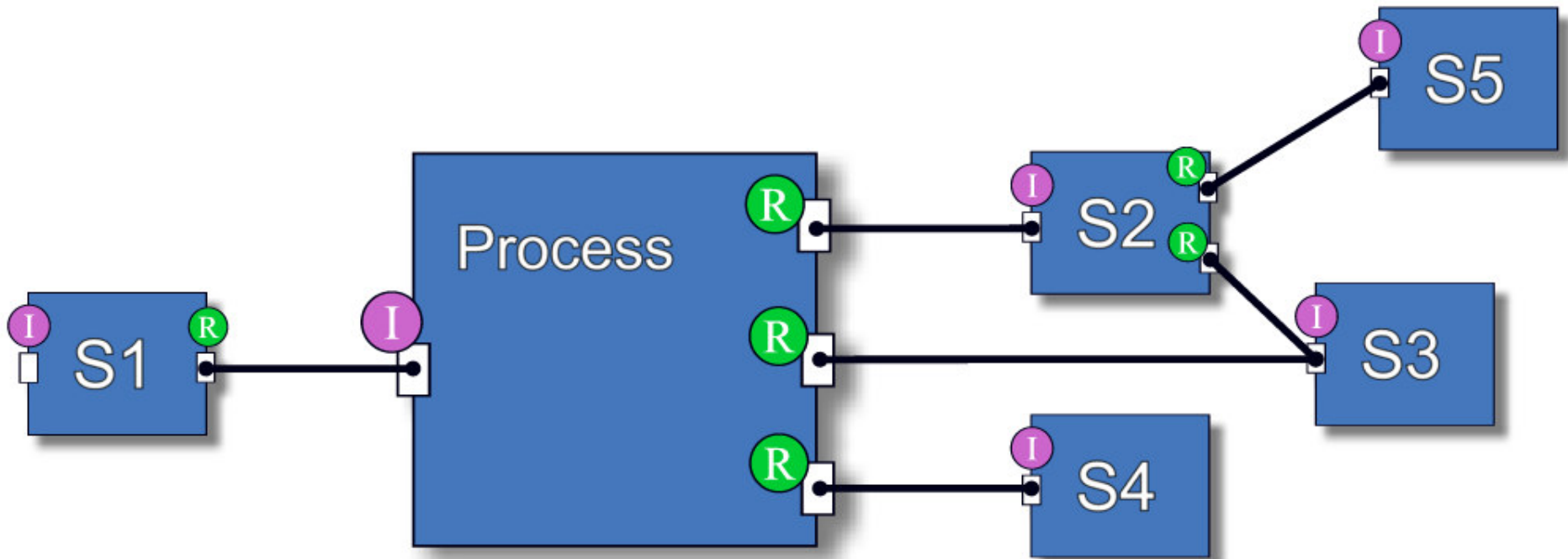
- **The invocation model for “business services”**
- **Are assembled into Modules**
- **Invocation can be**
 - Static or dynamic
 - Synchronous, asynchronous



Service Components have:

- **Interfaces** – define how to invoke Service Components (can be WSDL or Java Interfaces)
- **References** – specify how other Component are called from “my” Component
- **Wires** – define how a Reference connect to an Interfaces
- **Policies** – set on Wires, define quality of service
- **Implementations** – can be BPEL, Java, Mediation, etc...
- **Imports** – make external Interfaces visible inside the Module
- **Exports** – make internal (to the Module) Interfaces visible outside the Module

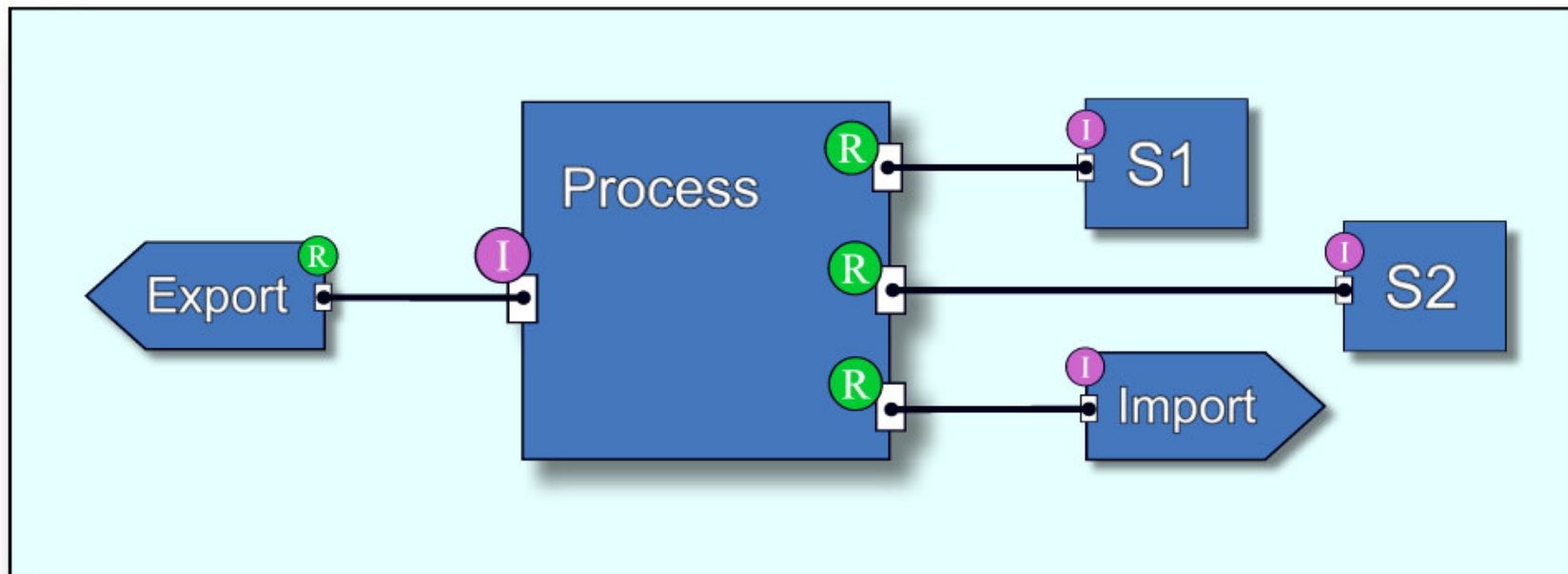
Assembling Components ...



Imports and Exports ...

- **Exports advertise capability out from a module**
- **Imports include capability from external services or modules**

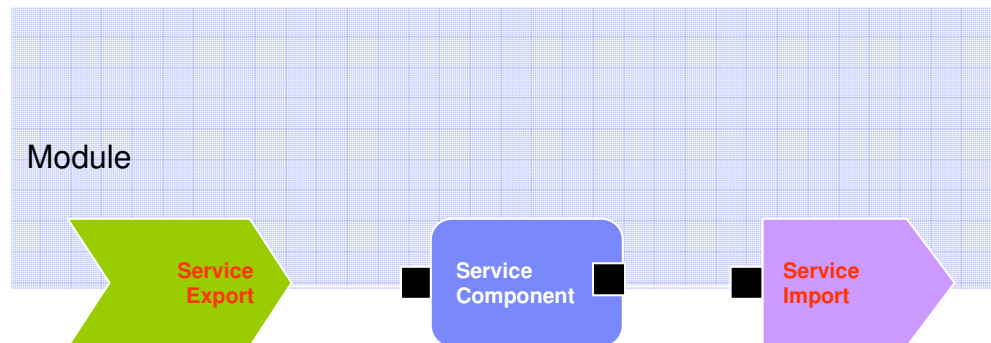
Module



SCA – Export / Import / Component Types

Export Bindings

- SCA
- Web services
- EIS (J2C, JMS)



Component Types

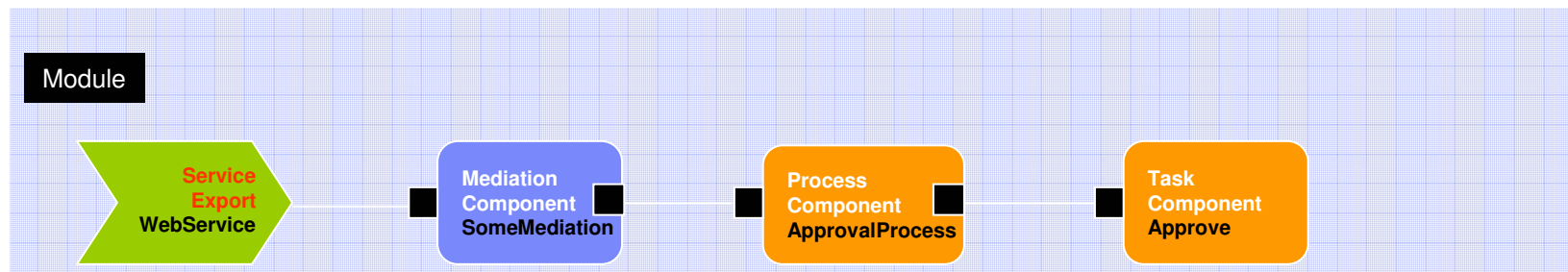
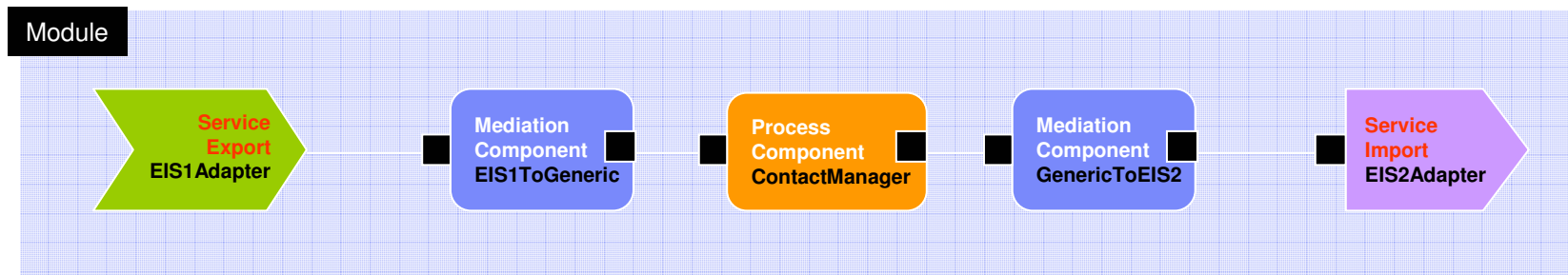
- Business Process
- Human Task
- Interface Map
- Business Rule
- Business State Machine
- Selector
- Java

Import Bindings

- SCA
- Web services
- EIS (J2C, JMS)
- EJB

SCA and Process

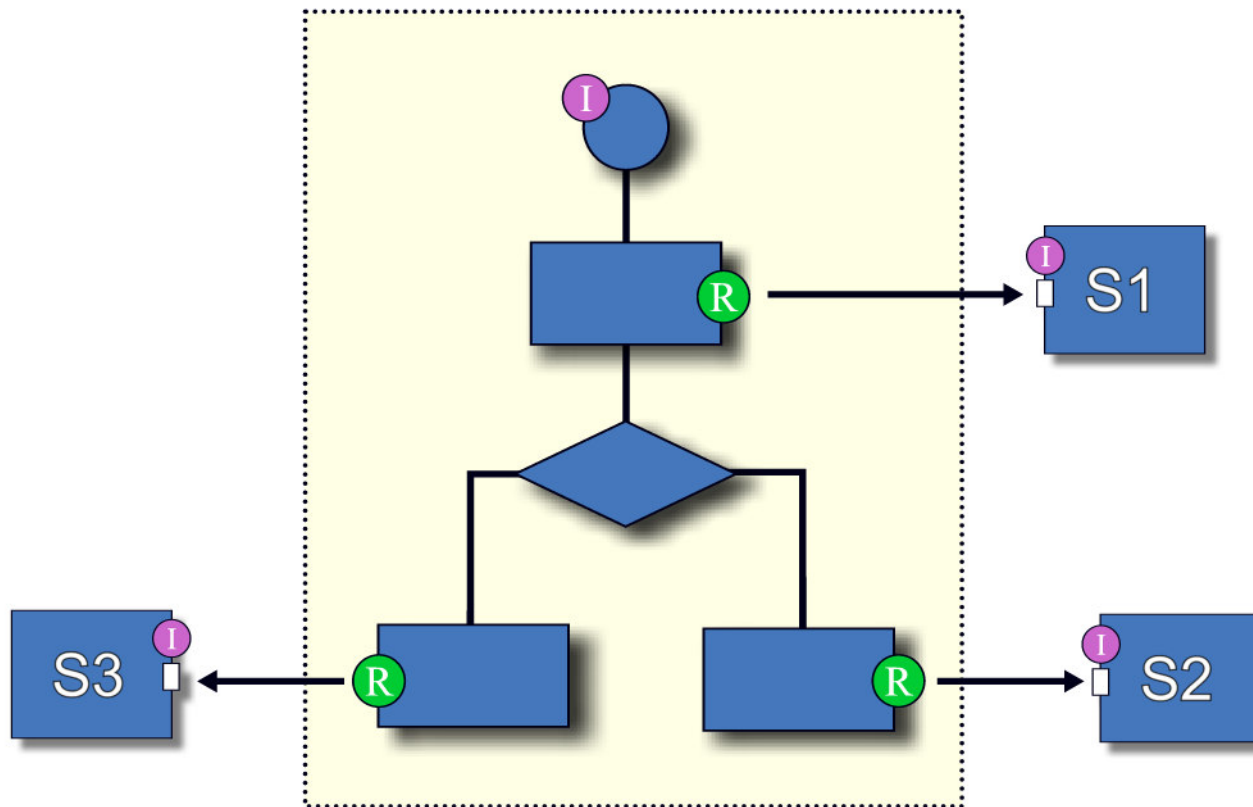
- Process are exposed as service components, and consume services.



- Human tasks are exposed as service component, and trigger services.

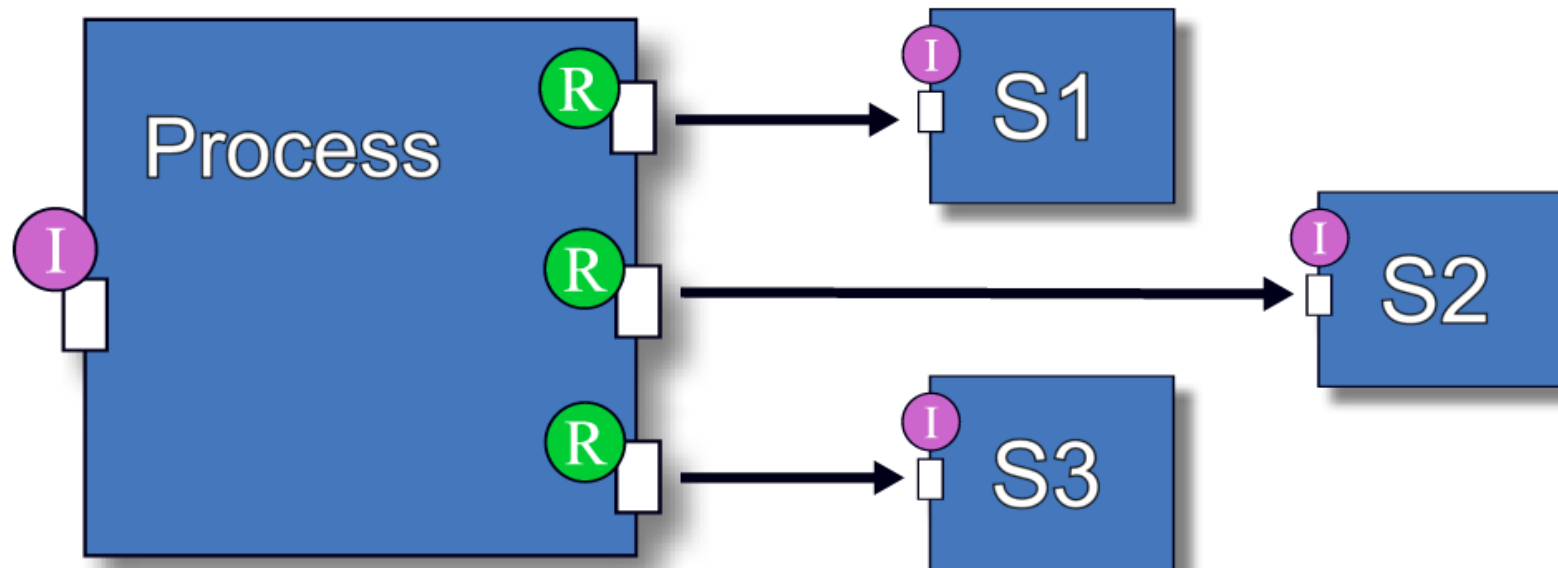
Processes and Services ...

- **Process invokes Services**
- **Process exposes an Interface**

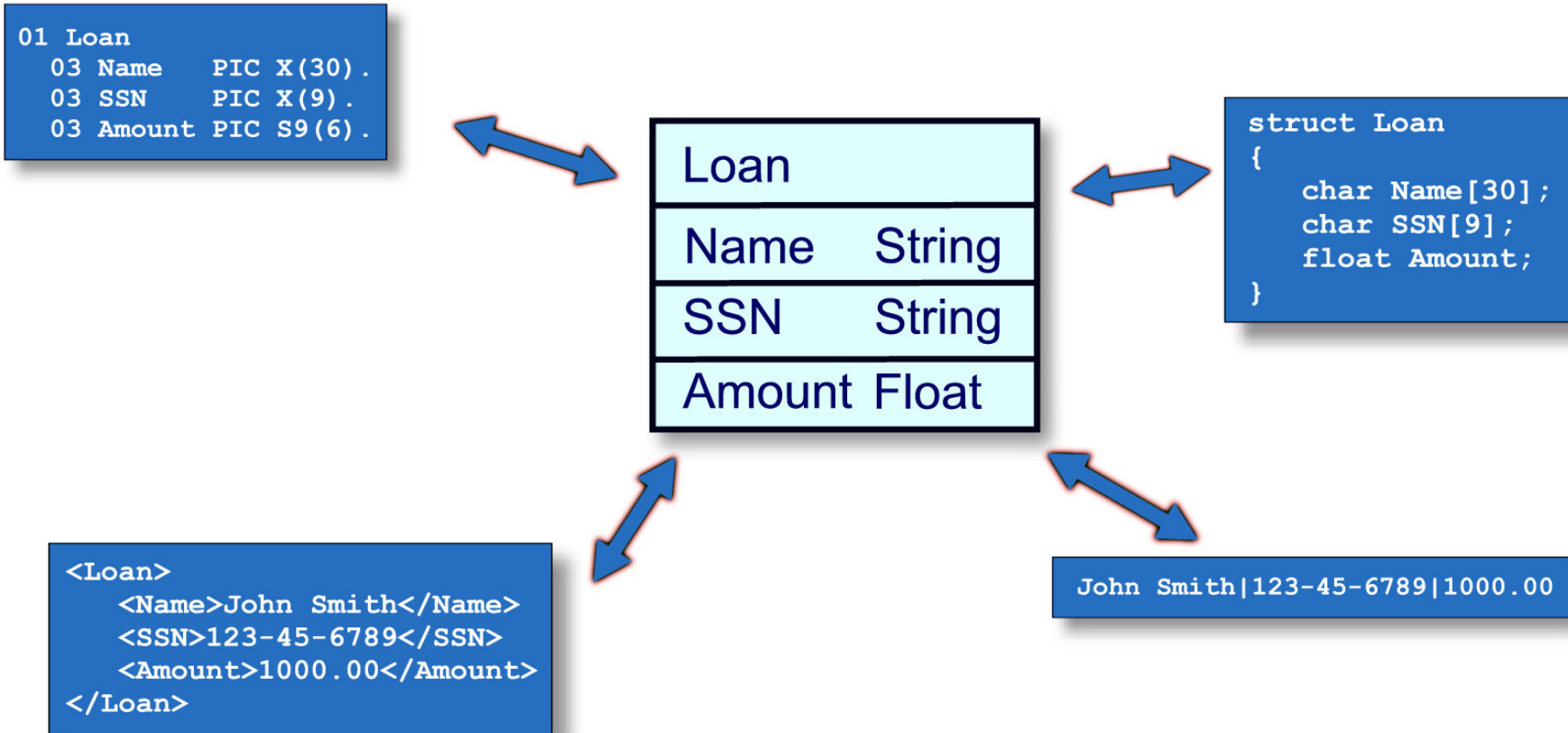


Logical View of Processes and Services ...

- **Process exposes an Interface**
- **Process invokes Services**
- **It looks just like any other Service**



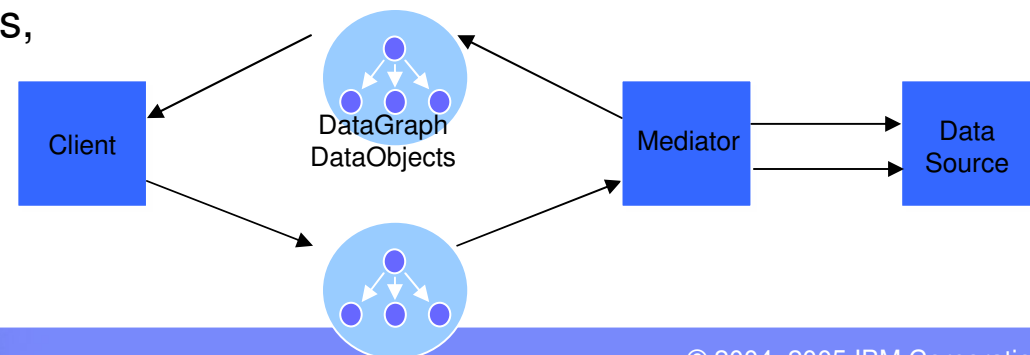
The Business Object ...



SDO – Service Data Object



- **Programming model for business data**
 - Used in Service Component implementation logic
- **Let programmers access data while unaware of the underlying data access technology**
 - JDBC, RMI/IIOP, JAX-RPC, JMS, etc.
 - Uses **Data Mediator** to implement different data sources
- **Can access data without knowing the message transport technology**
 - java.io.Serializable, DOM Objects, SOAP encoding, JMS message passing styles, etc.
- **Contains data and data graph**
- **Data graphs is an envelope for data objects**
 - Keeps change history
 - Allows for disconnected access to data objects
- **Data objects**
 - Representation is both XML and Java
 - Access can be static (strongly typed) or dynamic



JSR 235 - <http://jcp.org/en/jsr/detail?id=235>

Service Data Objects (SDO) / BusinessObjects

- **SCA support induces Service Data Objects.**
 - as service invocation parameters
 - as return value of service invocations
 - as exceptions thrown by service invocations

- **BPEL Java code snippets access variables represented by Data Objects.**

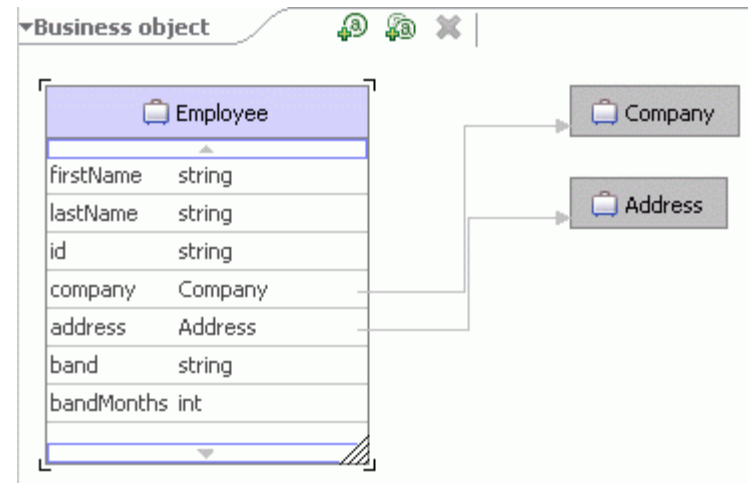
- **Data Objects are exposed via SCA client API.**

- **Human-facing activities display/populate SDOs.**

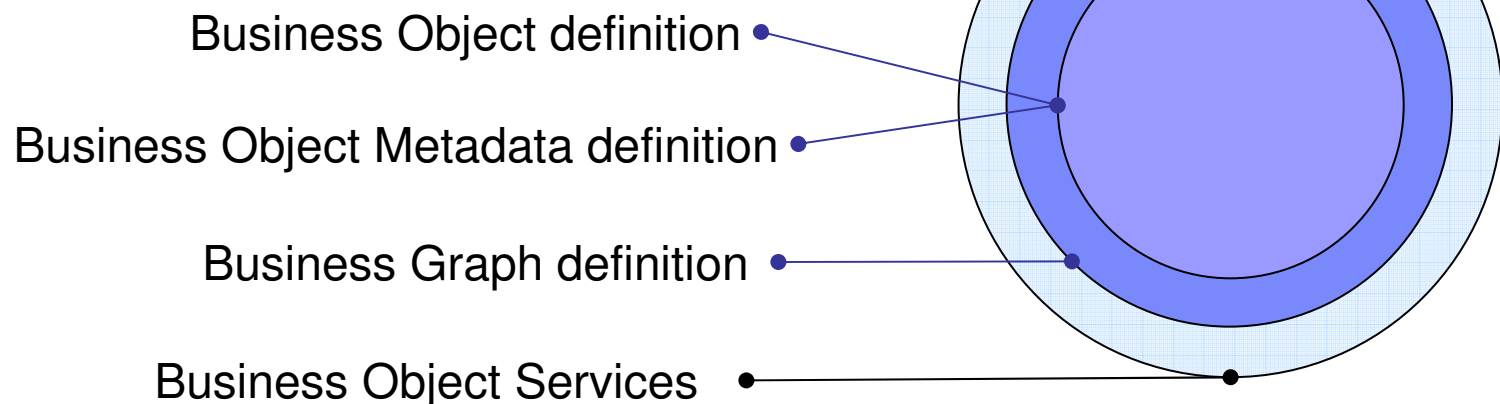
Common Data Model: Business Objects

- **Enhanced Service Data Object**

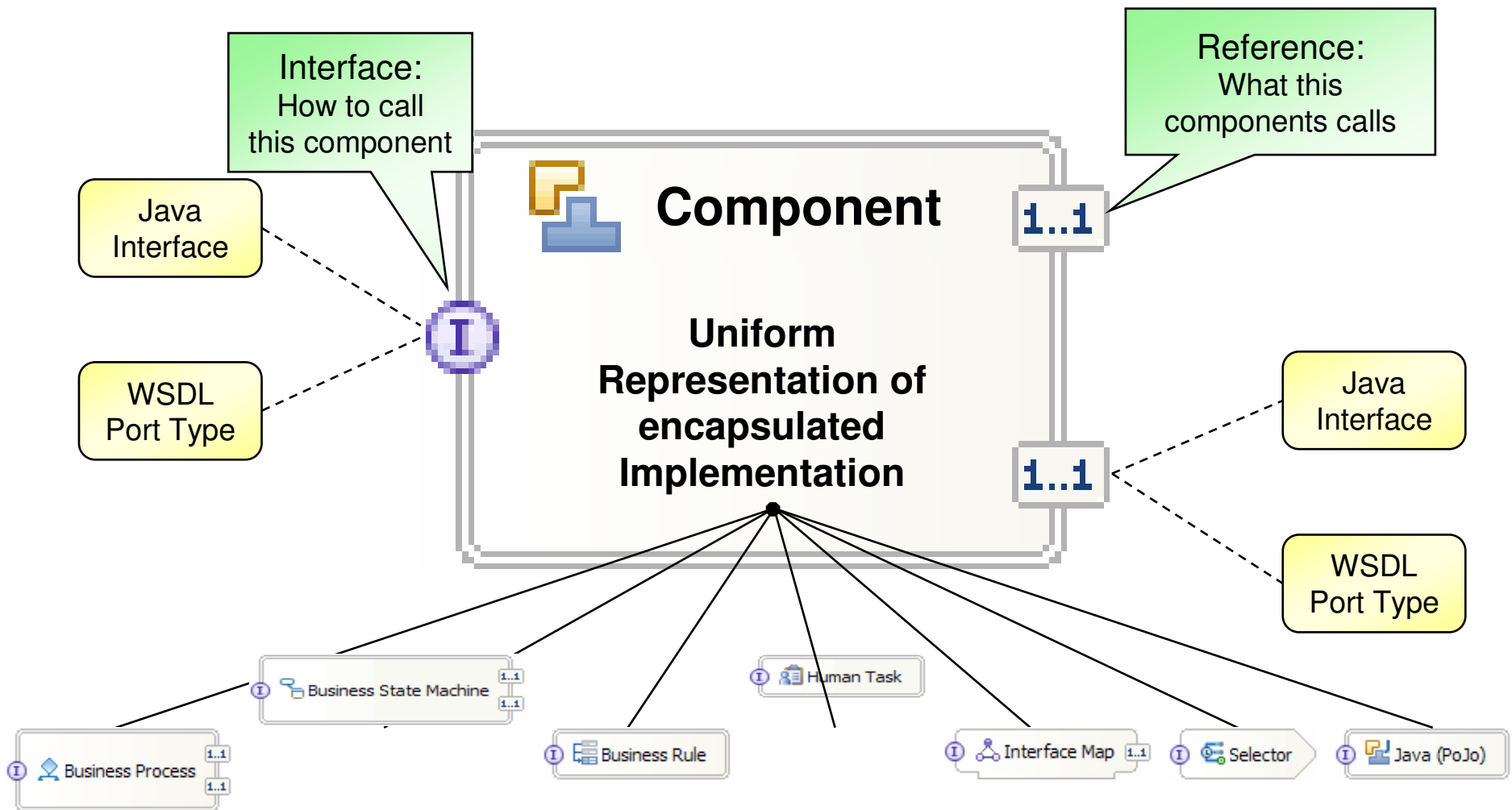
- Provides some function not available in base SDO specification (close to SDO 2.0)
- Supports Inheritance and Aggregation
- Enables import of 'standard' XSD



- **Business Object Framework consists of:**



Common Invocation Model: Service Components



Common Invocation Model: Imports / Exports

- **Adapters**
 - J2C 1.5
 - WBI Adapters

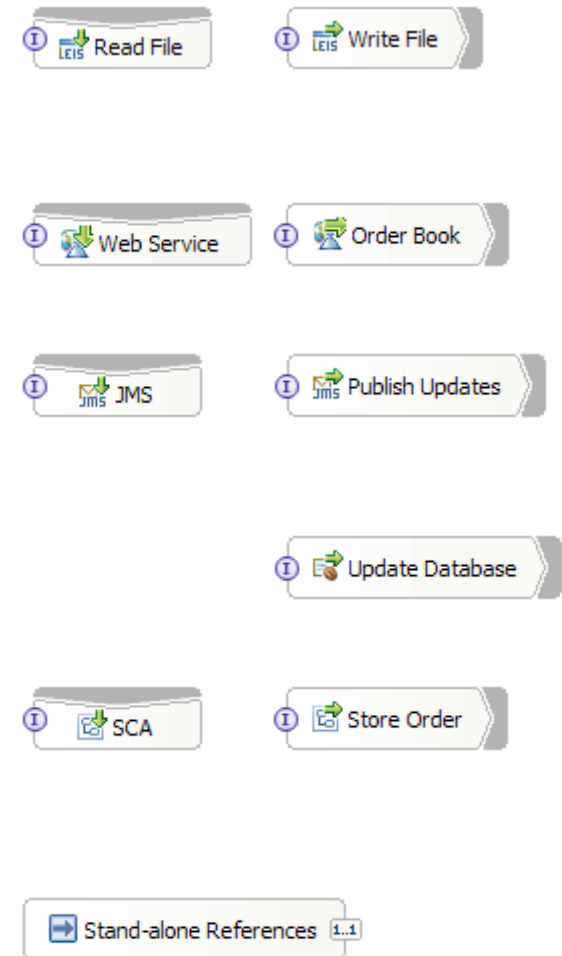
- **Web Services**
 - SOAP over HTTP, SOAP over JMS

- **JMS (WebSphere Messaging Resources)**
 - Point-to-Point and Publish/Subscribe
 - Integrate existing WebSphere MQ Solutions through MQ Link

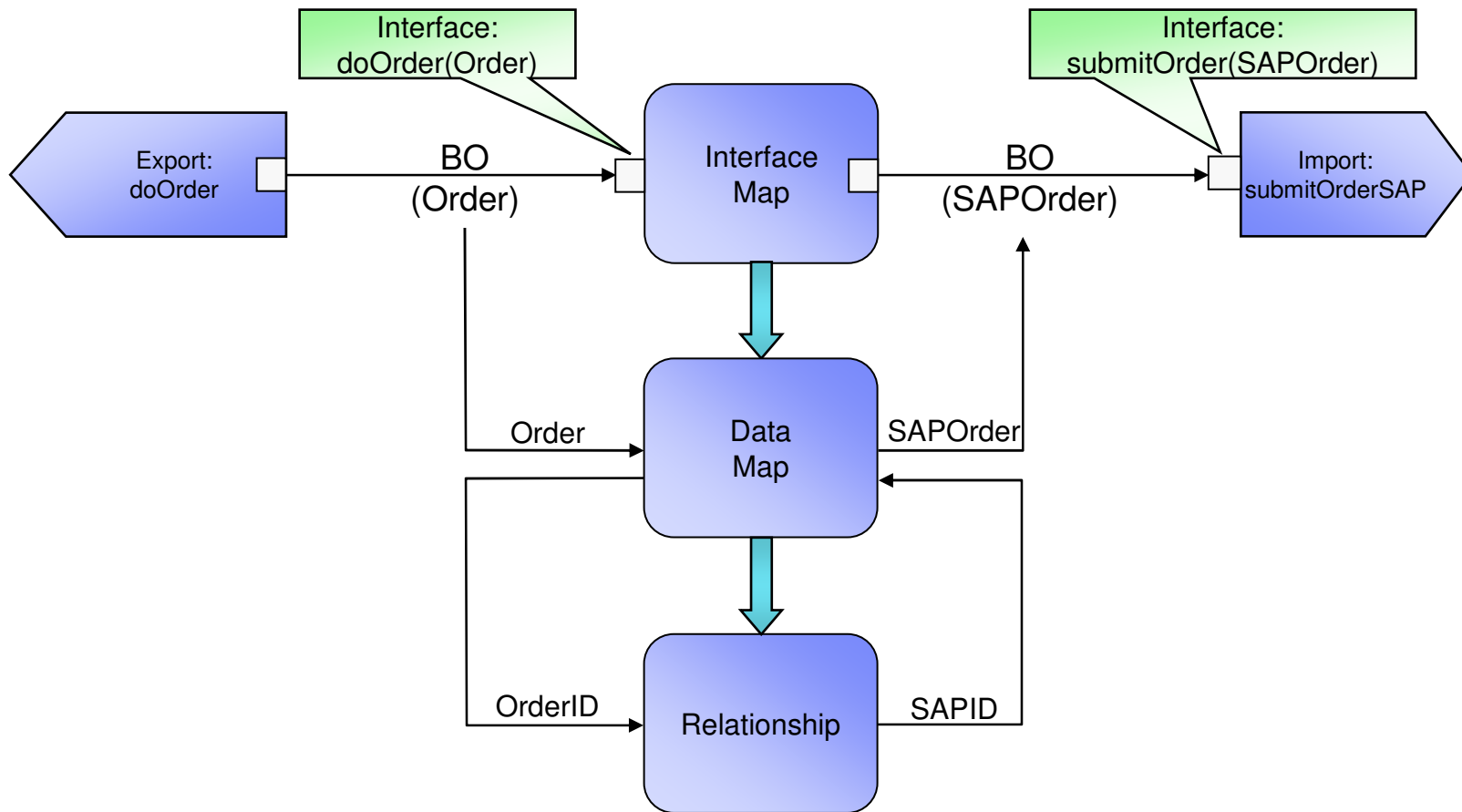
- **EJB (Session Beans)**

- **SCA**
 - Connect modules to each other without exposing the interface outside of WebSphere Process Server

- **Standalone Reference**
 - Enables an SCA API Client to call a Module

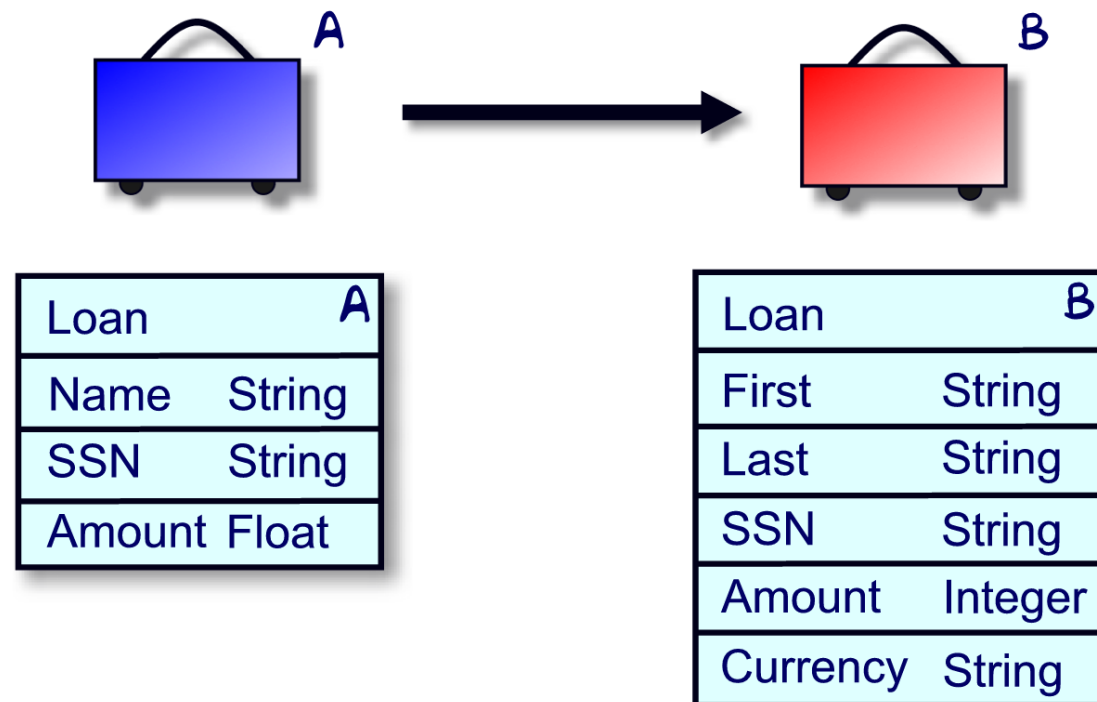


Transformation Components

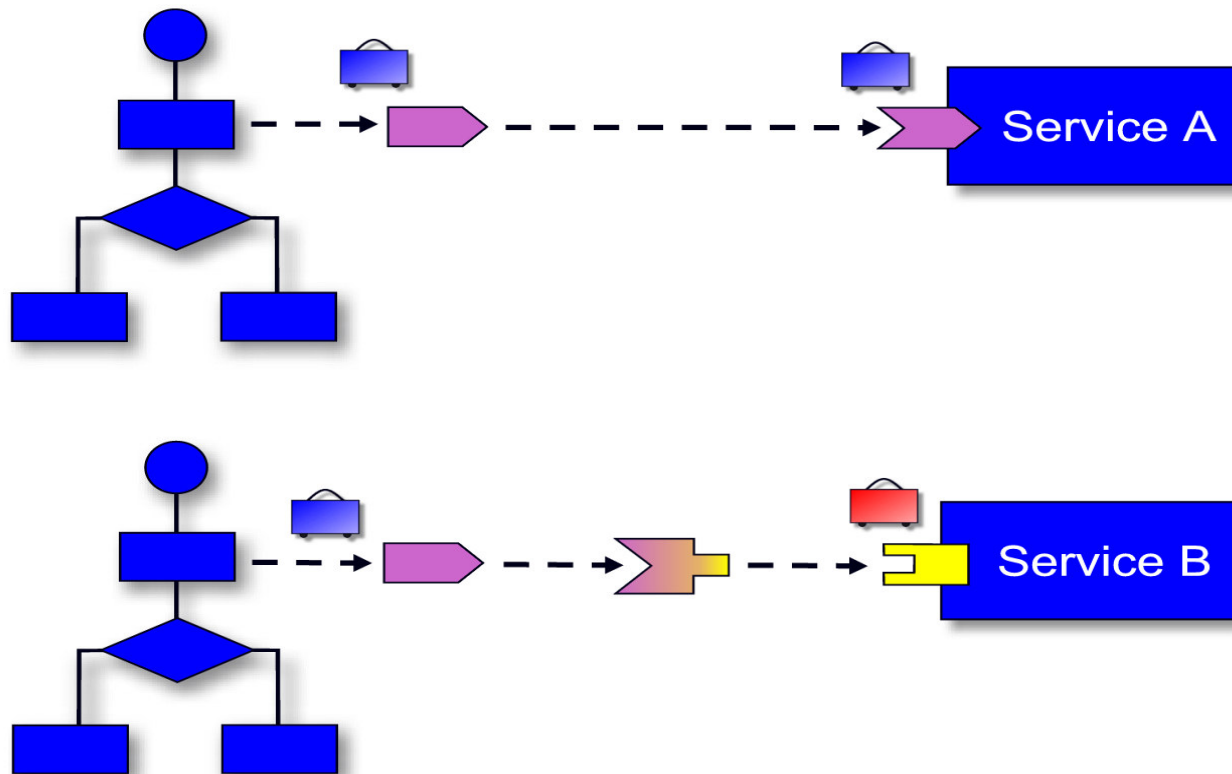


Business Object Maps ...

- Capability to map one Business Object to another
- Map attributes in one Business Object to attributes in another
- A variety of transformation rules available

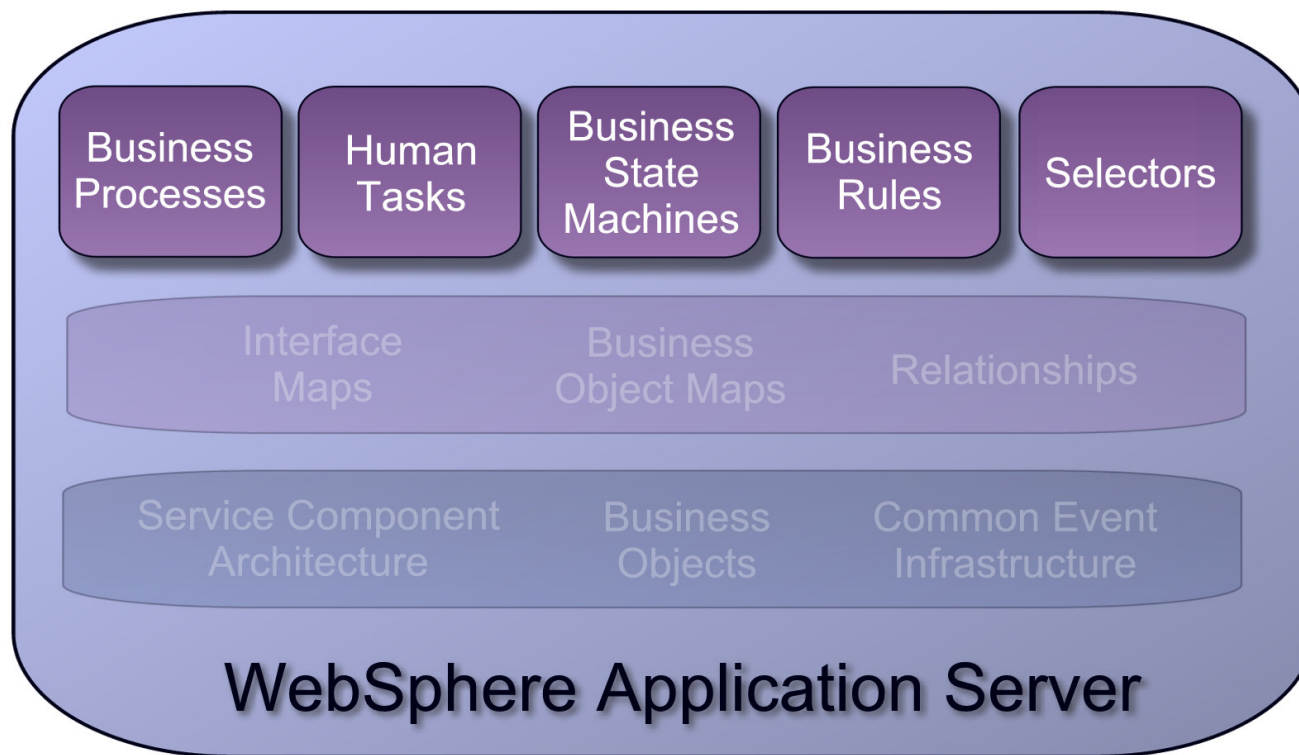


Interface Maps ...



WebSphere Process Server ...

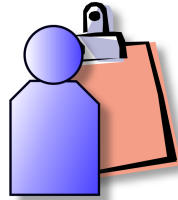
Service Components



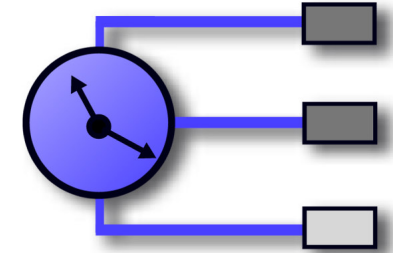
Service Components are the added-value components

Supplied Components ...

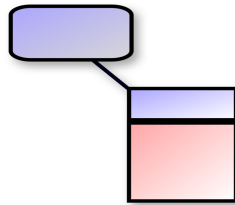
Human Task



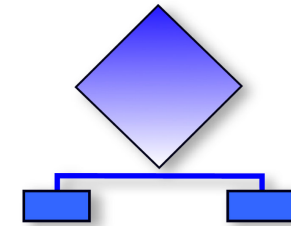
Selector



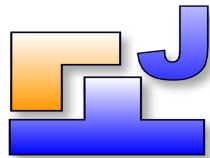
State Machine



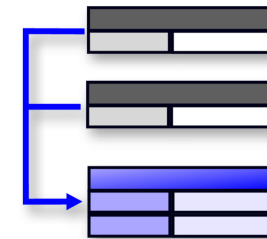
Process



Java



Business Rules



Resources

BPEL

- BPEL Specification
<http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/>
- OASIS Technical Committee
http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel

WBI Server Foundation

- WBI Server Foundation
<http://www-306.ibm.com/software/integration/wbisf/>
- WebSphere Studio Application Developer Integration Edition
<http://www-306.ibm.com/software/awdtools/studiointegration/>
- Various Redbooks: <http://www.redbooks.ibm.com/>

CEI

- Common Base Event Specification at
<http://www-106.ibm.com/developerworks/webservices/library/ws-cbe/>
- Further information can be found in the Infocenter at
<http://publib.boulder.ibm.com/infocenter/ws51help/index.jsp>

धन्यवाद

Hindi

多謝

Traditional Chinese

ขอบคุณ

Thai

Спасибо

Russian

Gracias

Spanish

English

Merc

i

French

Thank You

شكراً

Arabic

Obrigado

Brazilian Portuguese

Grazie

Italian

多谢

Simplified Chinese

Danke

German

நன்றி

Tamil

ありがとうございました

Japanese

감사합니다