



Enterprise IT Architectures

Enterprise IT Architectures SOA Part 1

Hans-Peter Hoidn

hans-peter.hoidn@ch.ibm.com

November 12, 2007

Agenda

- I. SOA Introduction
- II. Move from C/S to SOA
- III. SOA Reference Architecture
- IV. Identification and Specification of Services
- V. Example

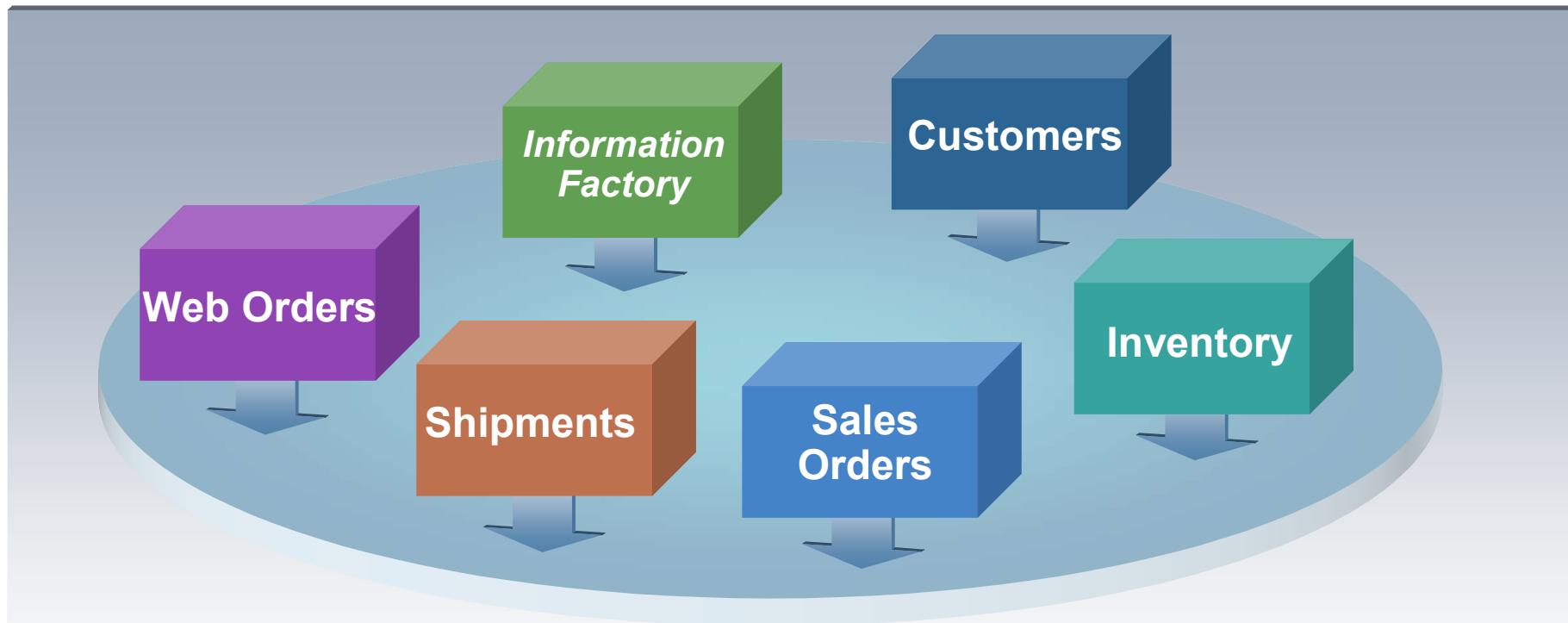
I. SOA Introduction

Why SOA (Service Oriented Architecture)

- Business is constantly changing (no longer stable)
- IT is required to be more flexible
- Monolithic applications can't be reused (historical limitations of current IT)
- SOA ties together changing Business Models and supporting IT Architecture
- SOA separates concerns, locations
- SOA follows essential principles: loosely coupled, federated, contract based
- SOA provides integration and supports business processes

Service Oriented Architecture

Moves IT Logic Out of Services



Services defined as units of business logic separated from...

- Flow of control and routing
- Data transformation and protocol transformation

SOA addressing IT as well as Business – common shift

Shift to a Service-Oriented Architecture

From **To**

- Function oriented
- Build to last
- Prolonged development cycles

- Process oriented
- Build to change
- Incrementally built and deployed



- Application silos
- Tightly coupled
- Object oriented
- Known implementation

- Orchestrated solutions
- Loosely coupled
- Message oriented
- Abstraction

What is SOA

- SOA is an architectural style or approach whose goal is to achieve loose coupling among interacting software agents
- All functions (that need to be used by more than one system) are defined as "services"
- Service providers agree to a defined, implementation-independent interface with service clients
- Services oriented architecture is the policies, practices and frameworks
 - that enable application functionality and IT services to be
 - provided and requested as a set of services
 - using a standards based form of interface.

SOA Principles

Componentized

**Standardized services
interfaces for applications
and resources**

Interoperable

**Easy information exchange
between applications and/or
resources**

Modular

**Mix and match, add or
remove, business processes
and infrastructure**

Scaleable

**Start with what you have and
add additional resources as
needed**

SOA is different things to different people

A set of services that a business wants to expose to customers and clients

an architectural style which requires a service provider, requestor and a service description.

a set of architectural principles and patterns which address characteristics such as *modularity, encapsulation, loose coupling, separation of concerns, reuse, composable and single implementation.*

A programming model complete with standards, tools, methods and technologies such as web services.

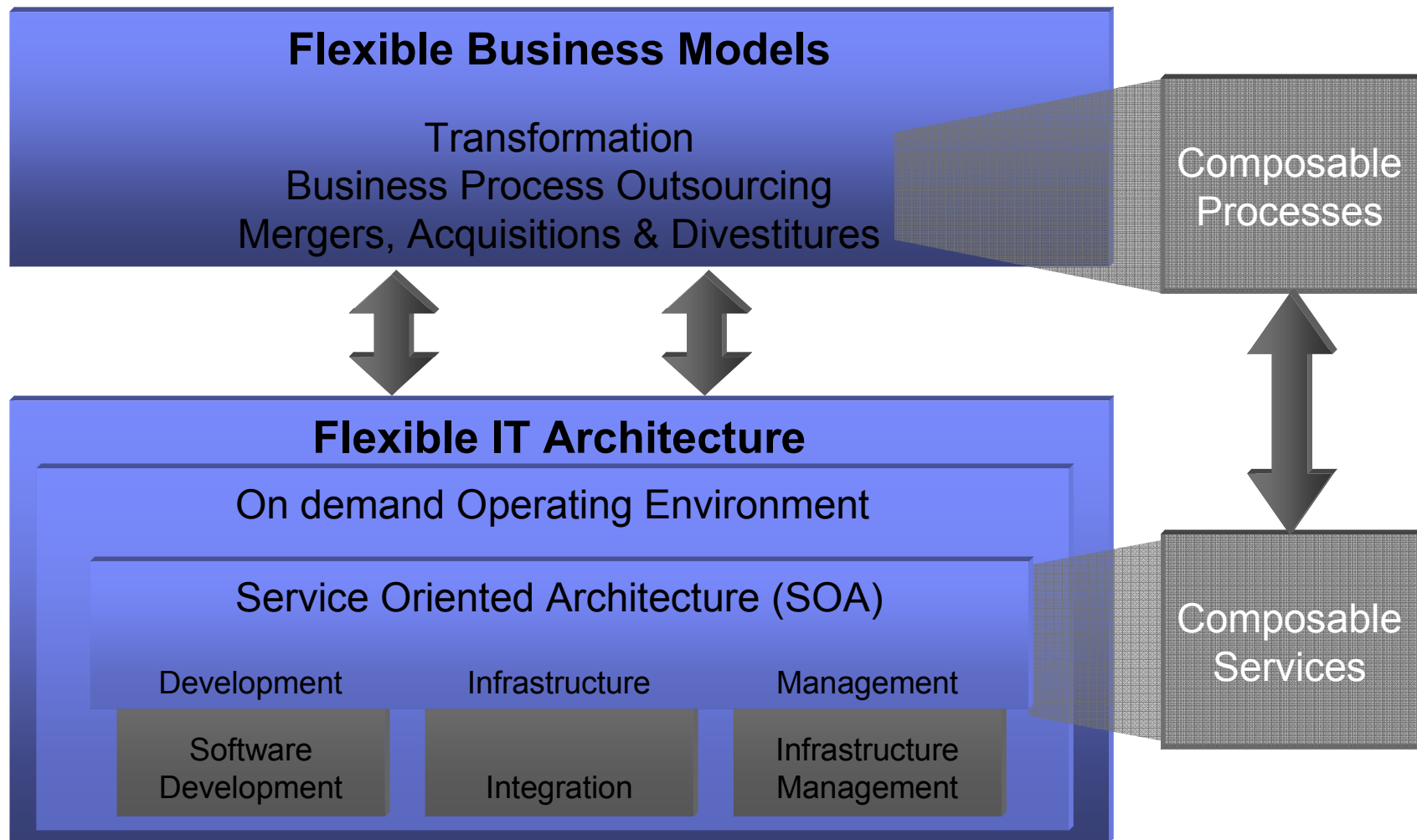
Roles

Business

Architecture

Implementation

Greater flexibility from business models and supporting IT architecture



II. From C/S to SOA

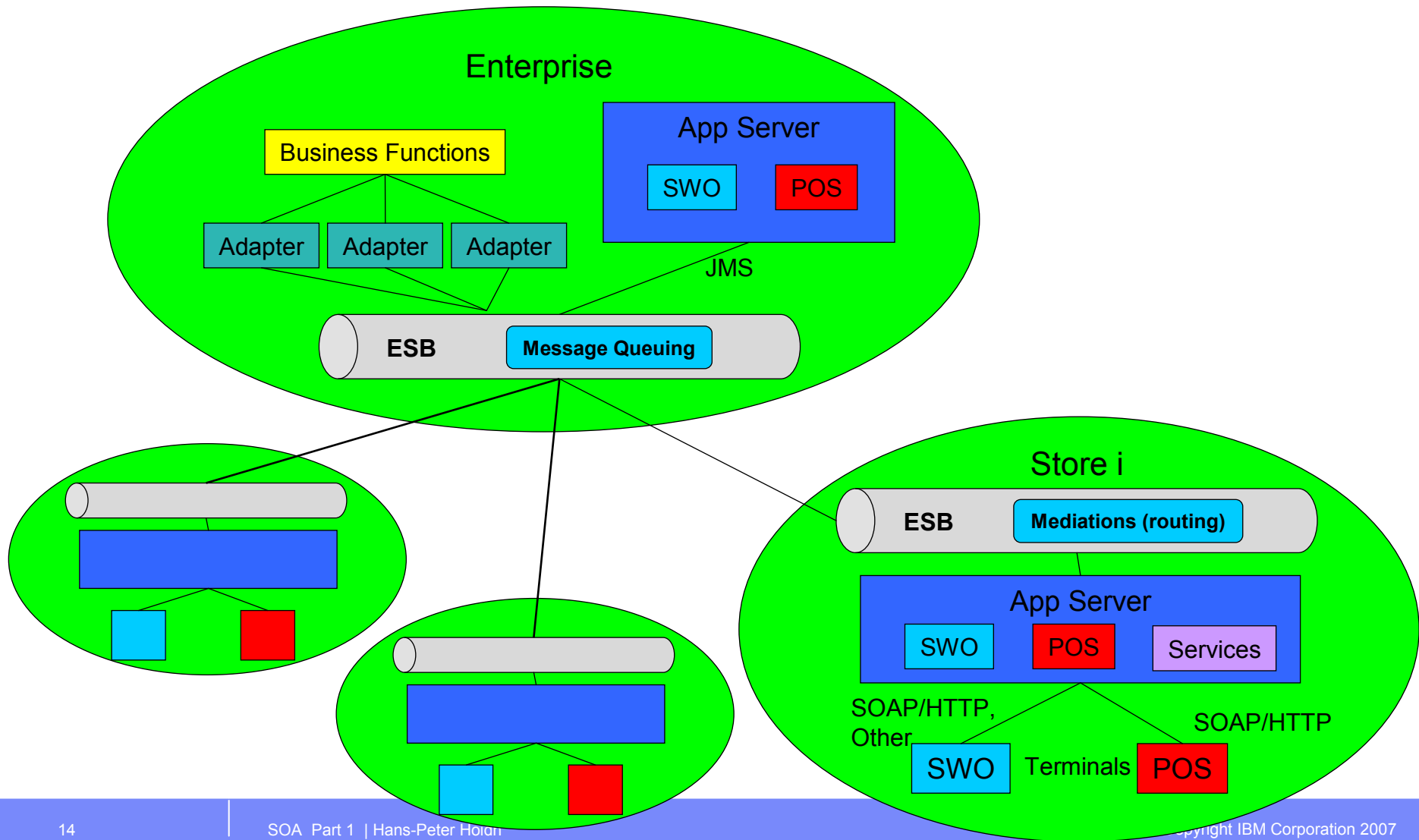
Elements for a successful End-to-End Implementation

- Success with SOA Requires a Core Infrastructure
- The SOA reference architecture is a way of looking at the set of services that go into building an SOA.
- The backbone of the reference architecture is the ESB (Enterprise Service Bus), which facilitates communication between services.
- The reference architecture is a great tool for laying out roadmaps for pursuing SOA.

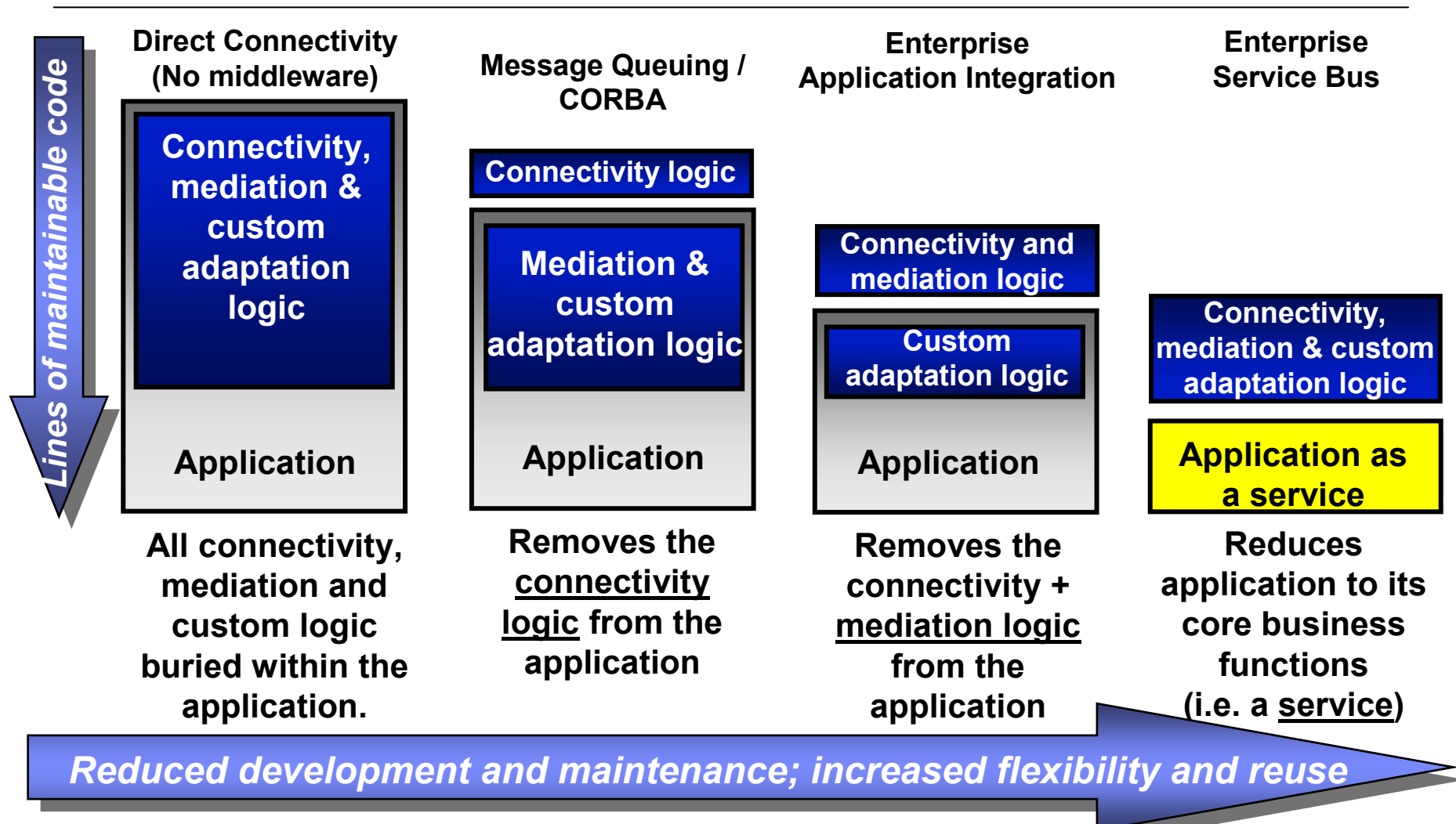
ESB (Enterprise Service Bus)

- An Enterprise Service Bus (ESB) is an architectural pattern defining a flexible connectivity infrastructure for integrating applications and services.
- The architecture pattern is a guiding principle to enable the integration and federation of multiple service bus instantiations.
- An ESB performs:
 - Routing messages between services
 - Converting transport protocols between requestor and service – managing multiple protocols
 - Transforming message content between requestor and service
 - Handling business events from disparate sources

ESB Pattern in Action – Retail Scenario

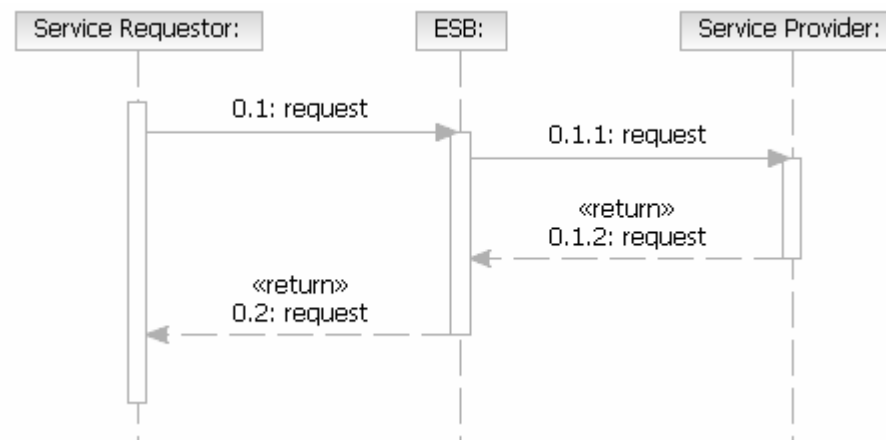


ESB is the next stage in the technology evolution



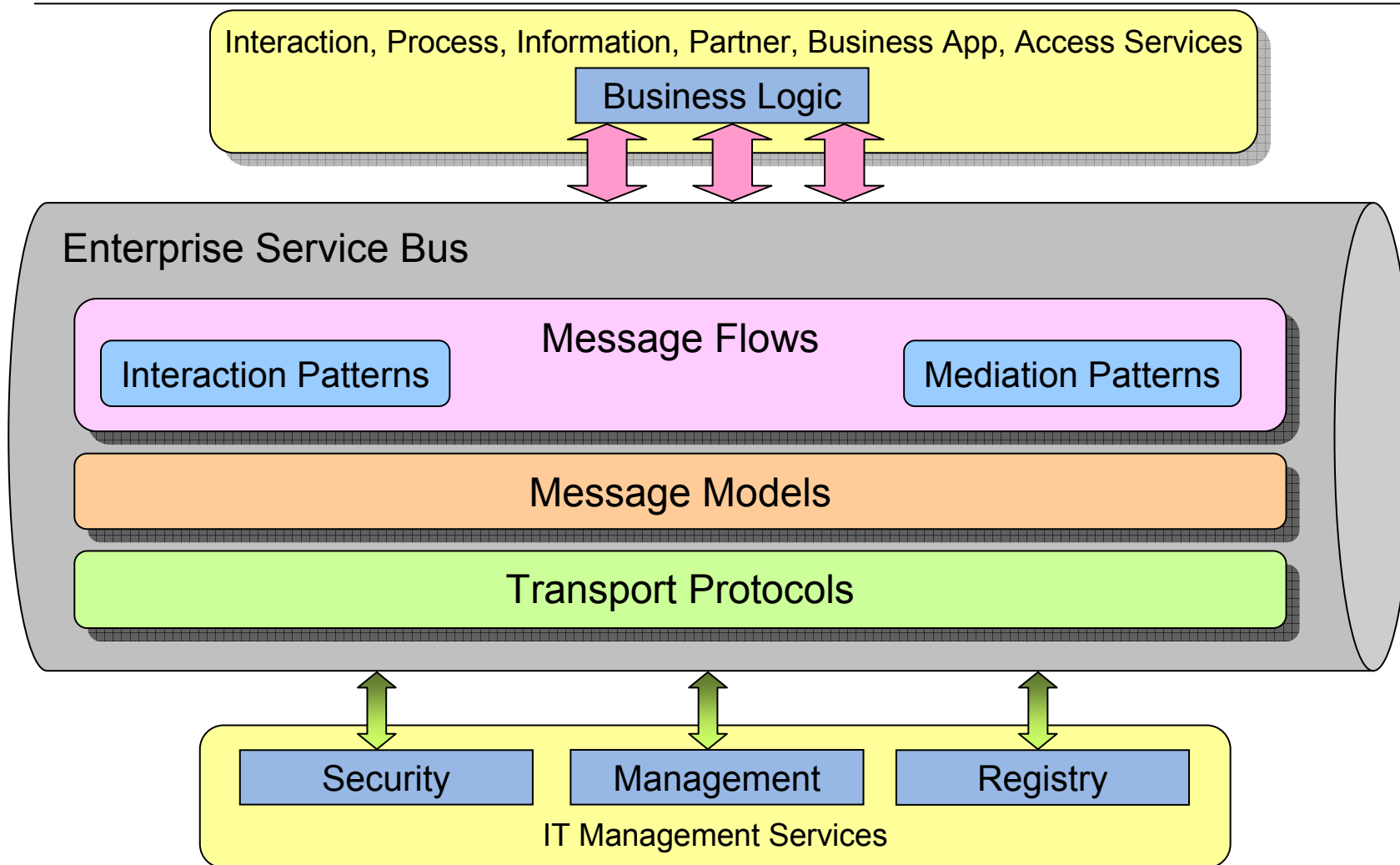
ESB Core Principle – Service Virtualization

- ESB acts as an intermediary (proxy) between requestor and provider

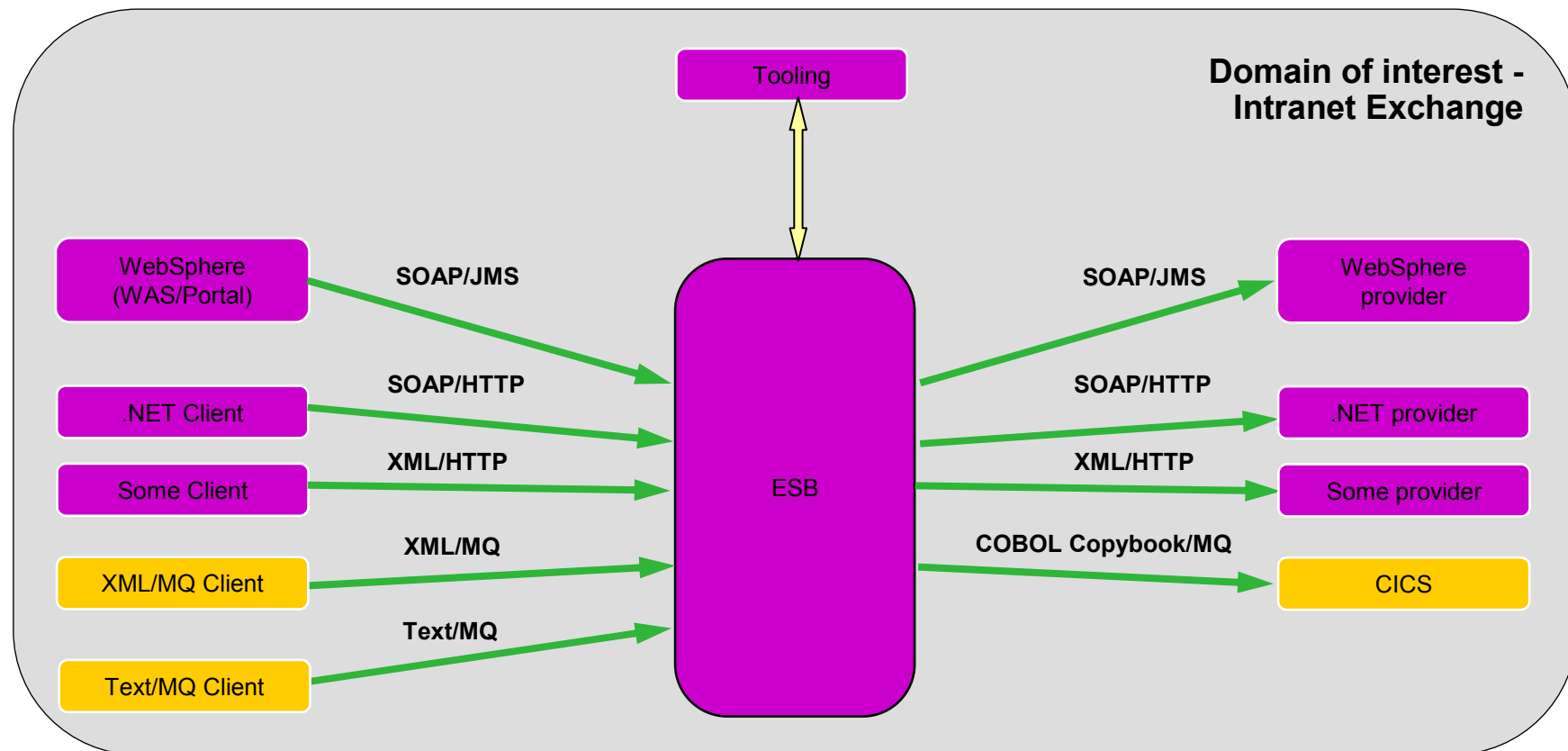


- ESB provides *service virtualization* of
 - Location and identity*
 - Interaction protocol*
 - Interface*
- Interactions are *decoupled*, supporting *separation of concerns*

Expanded View of the Enterprise Service Bus

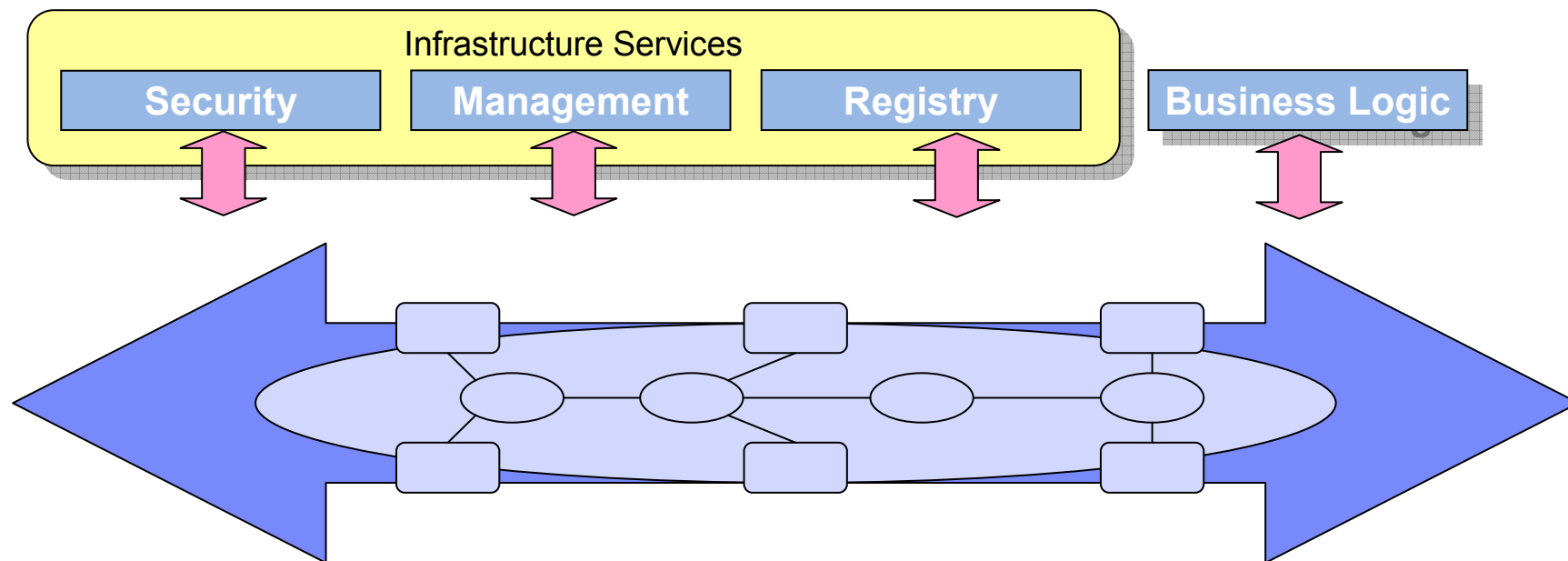


Case: Multi-protocol Exchange – Intermediary decoupling heterogeneous consumers and suppliers

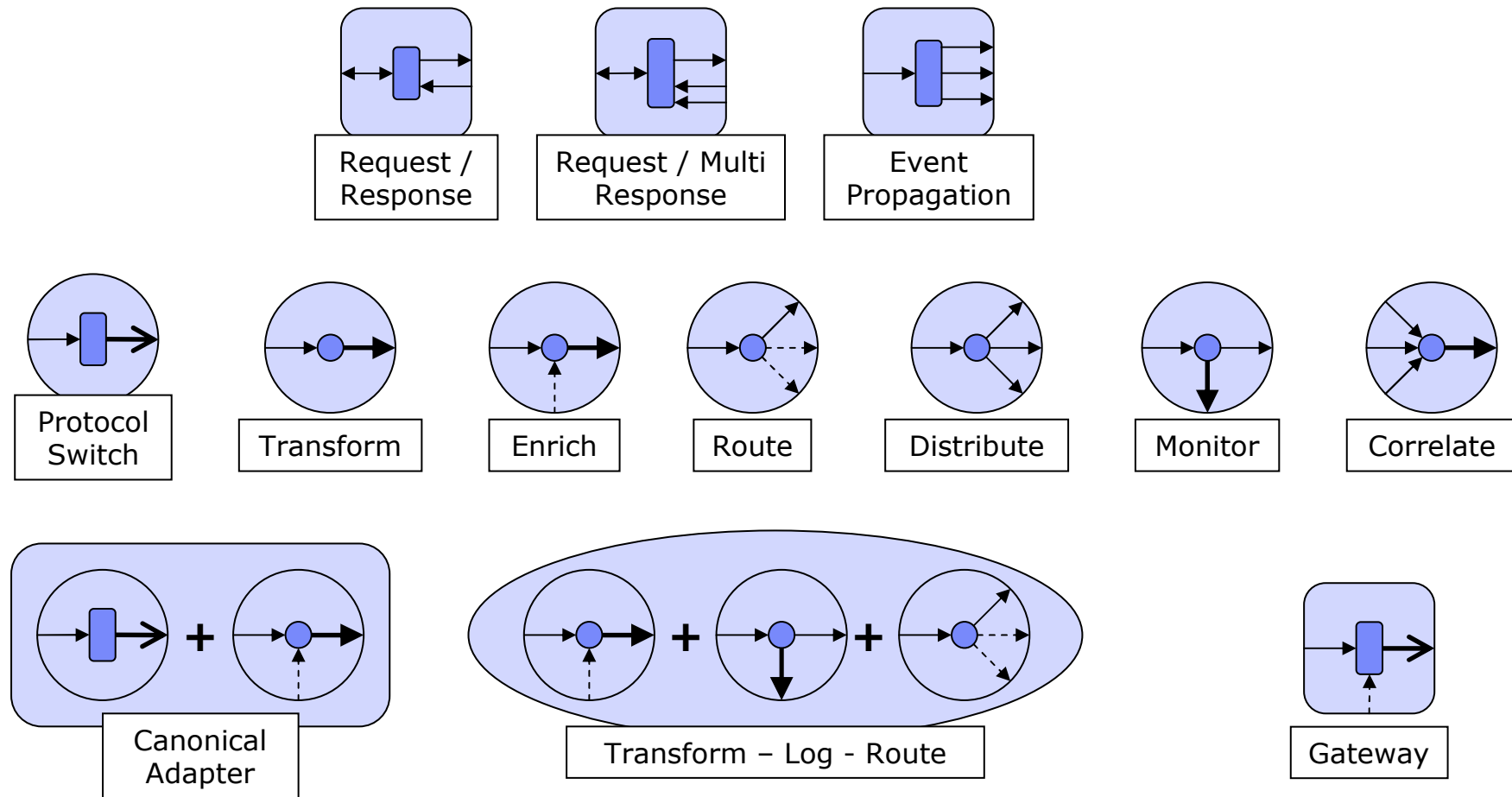


What is NOT in the Enterprise Service Bus?

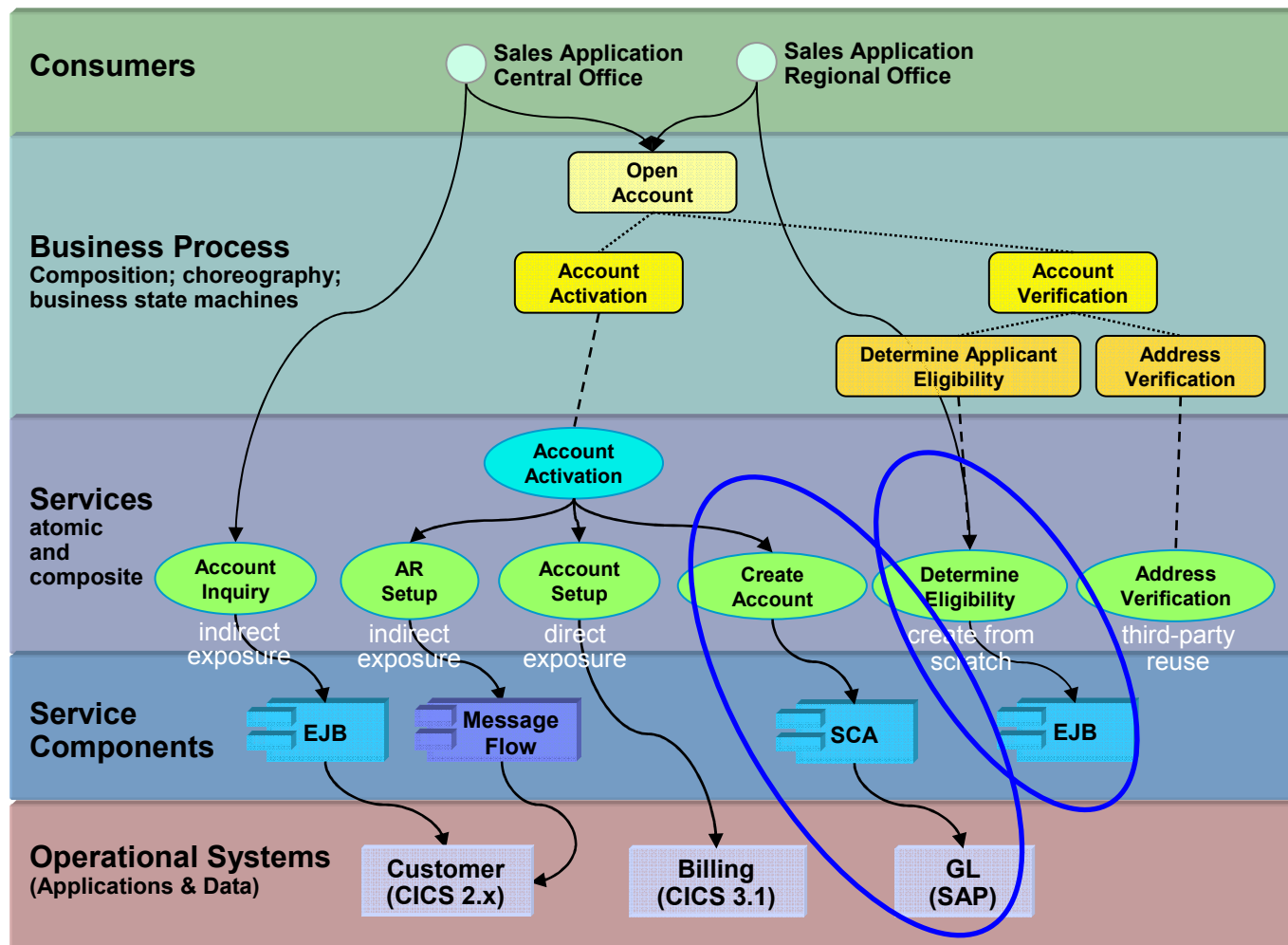
- In the ESB – capability necessary to support service virtualization
- Not in the ESB – everything else!
 - But things not in the ESB may use or be used by the ESB



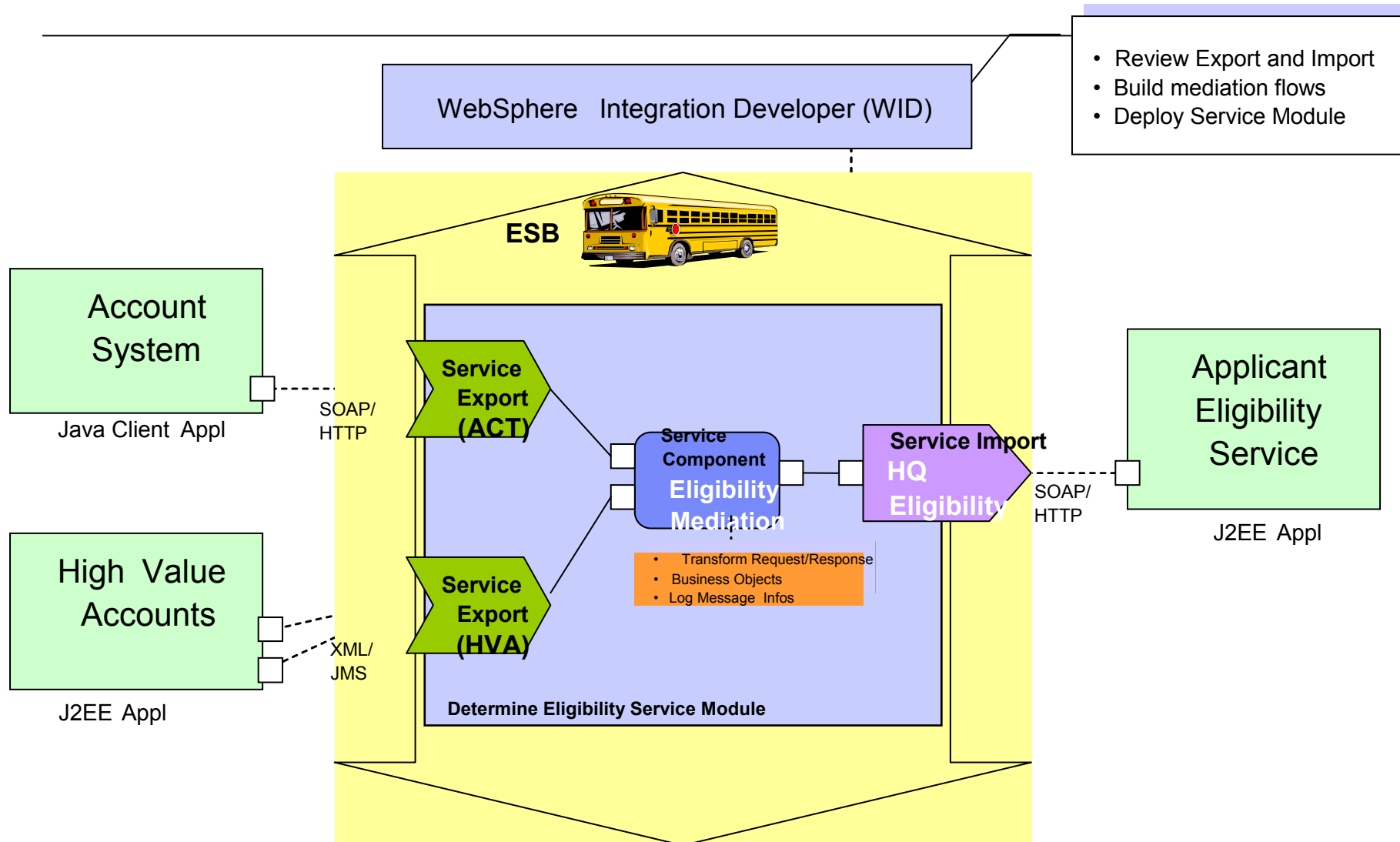
Mediation Patterns



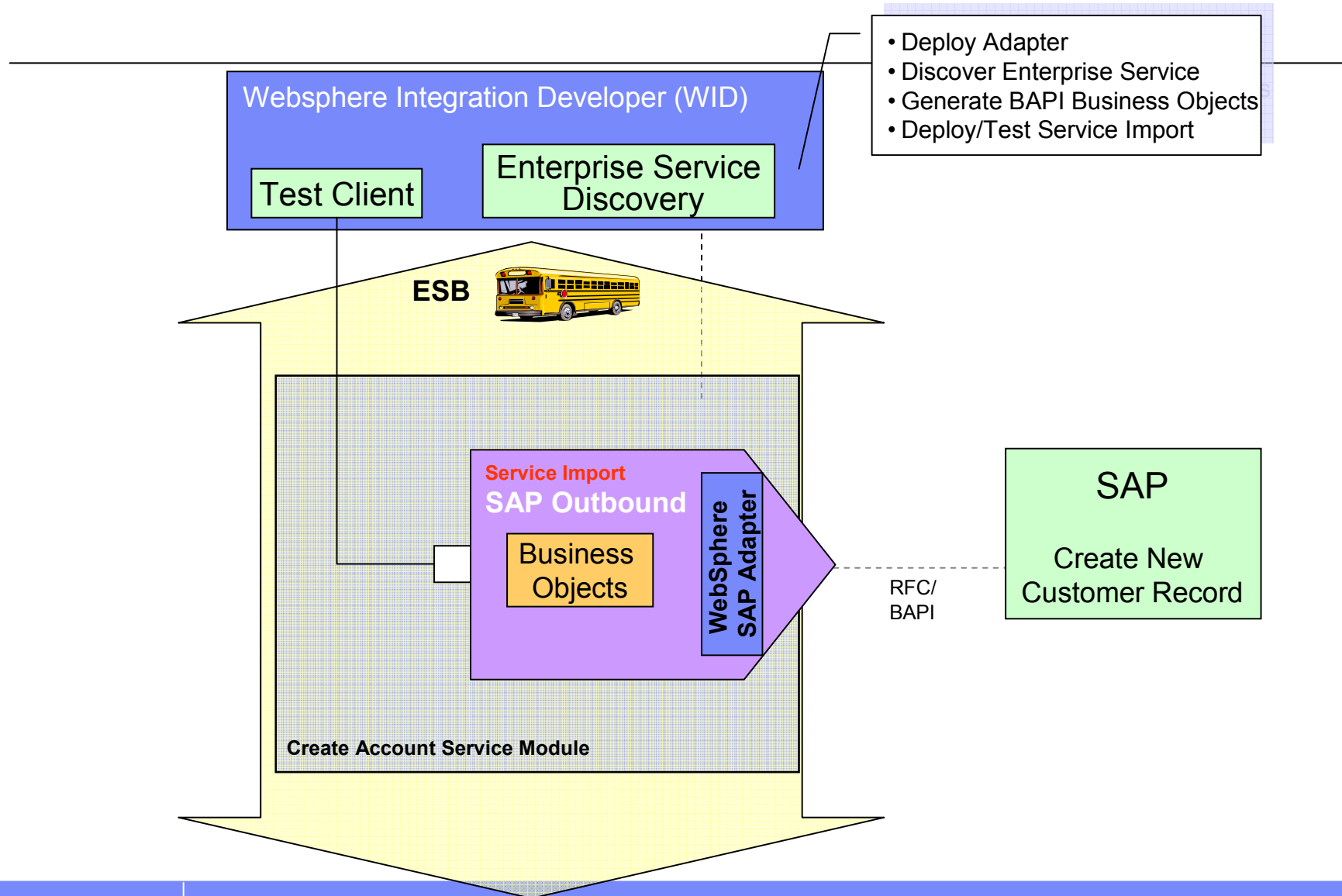
Example JK Enterprise – a virtual company with an „Open Account Process“



Example A: Multiple Channel Access to Backend Service

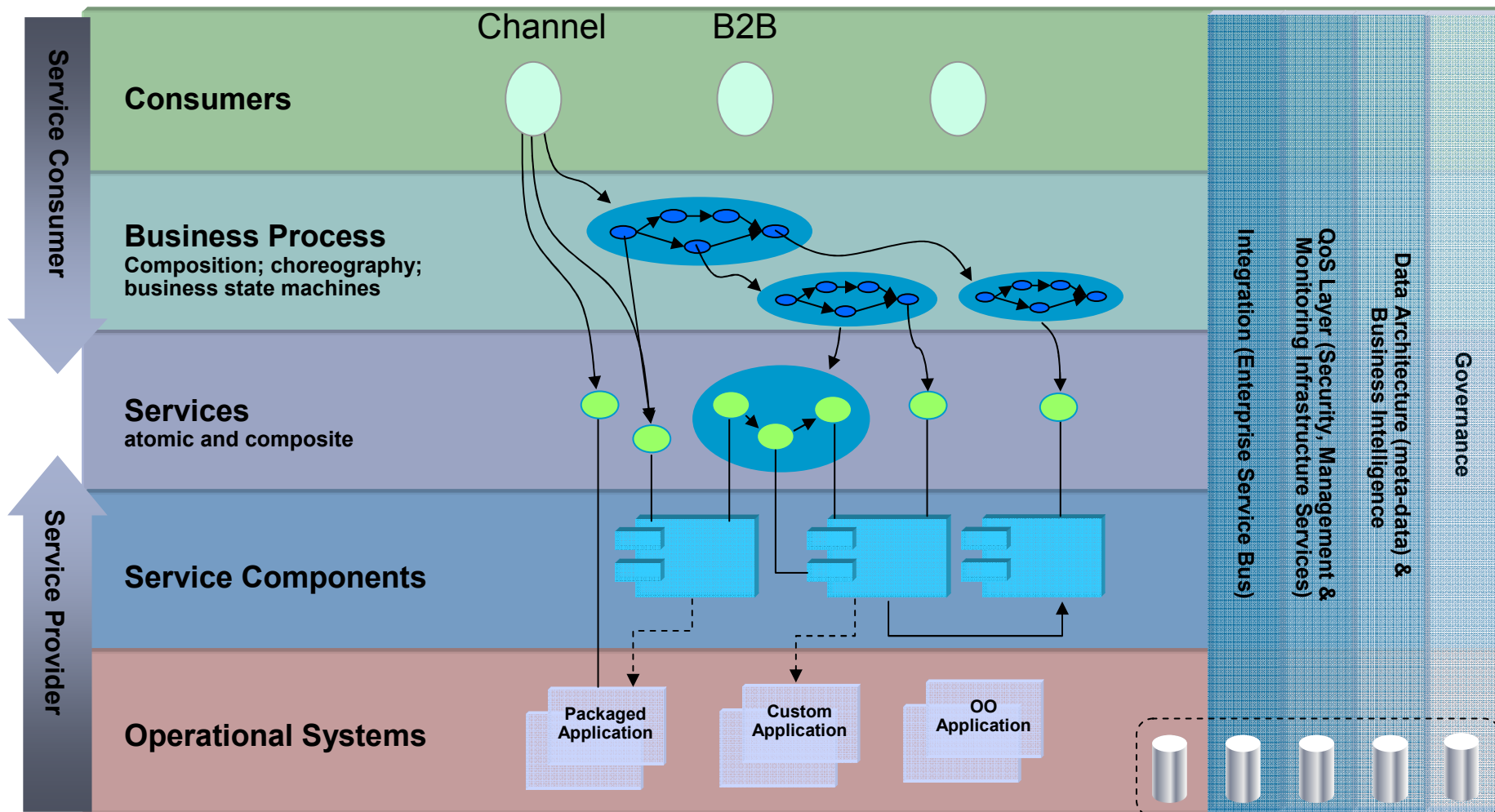


Example B: Create SAP Service



III. SOA Reference Architecture

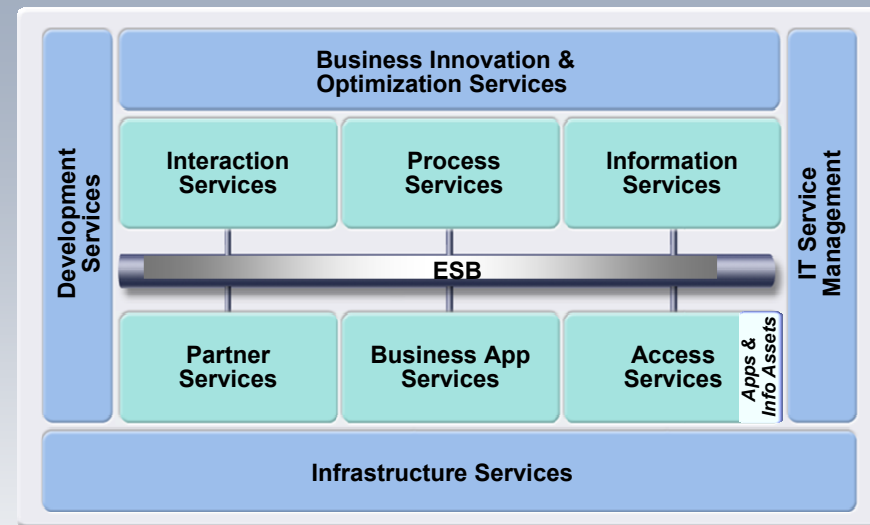
Moving to Services-Oriented Solutions – Layered Architecture View



The SOA Reference Architecture and its Key Principles

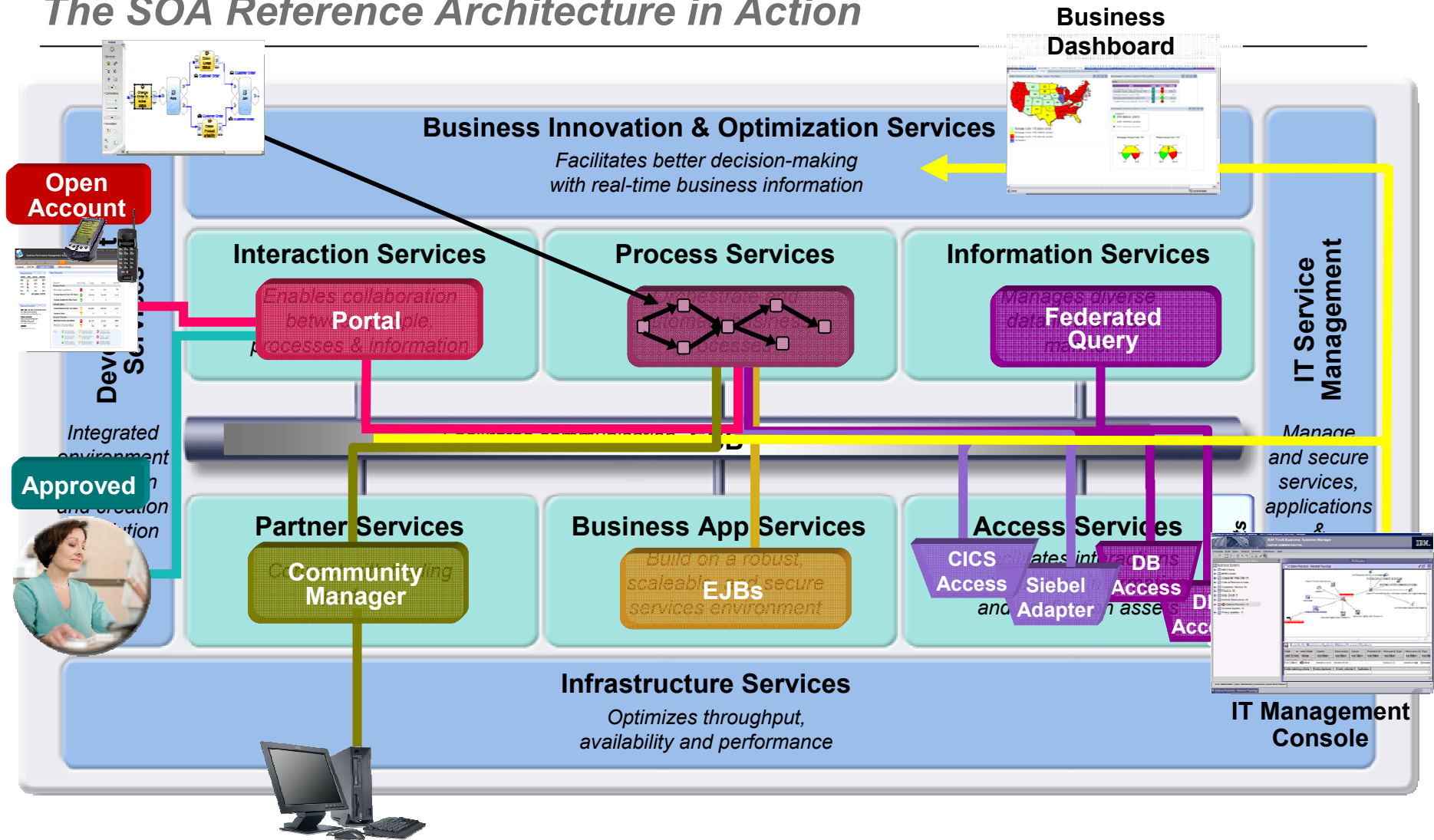
Providing IT Flexibility to Meet the Demands of Business

- Linkage between business and IT through support of the entire SOA Lifecycle
- Connectivity and Service Isolation through the Enterprise Service Bus
- Separation of Concerns/Modularity for incremental adoption
- Component-based Programming and Solution Development
- Business and IT Monitoring and Management
- Open Standards



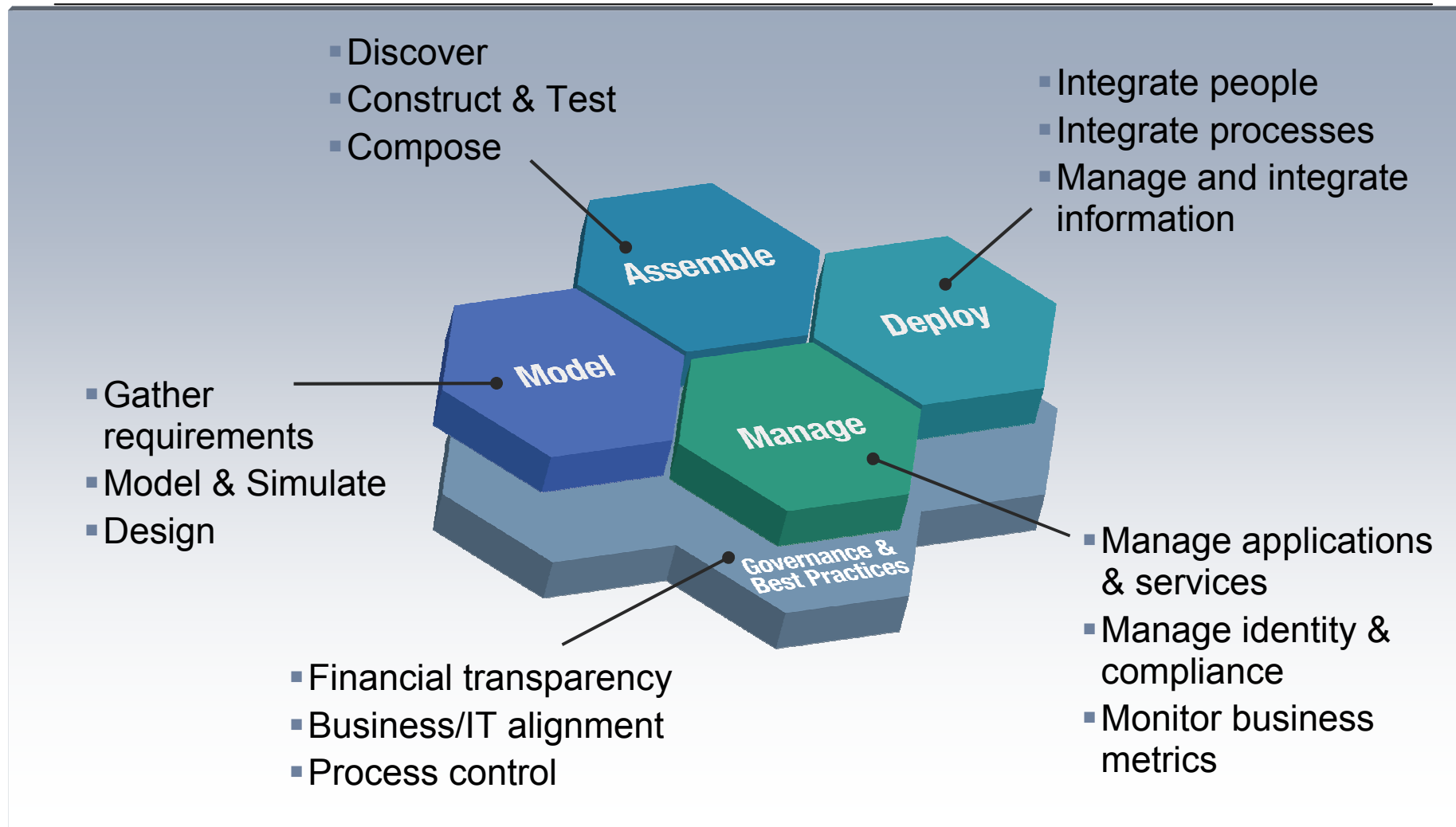
Separation of Concerns

The SOA Reference Architecture in Action

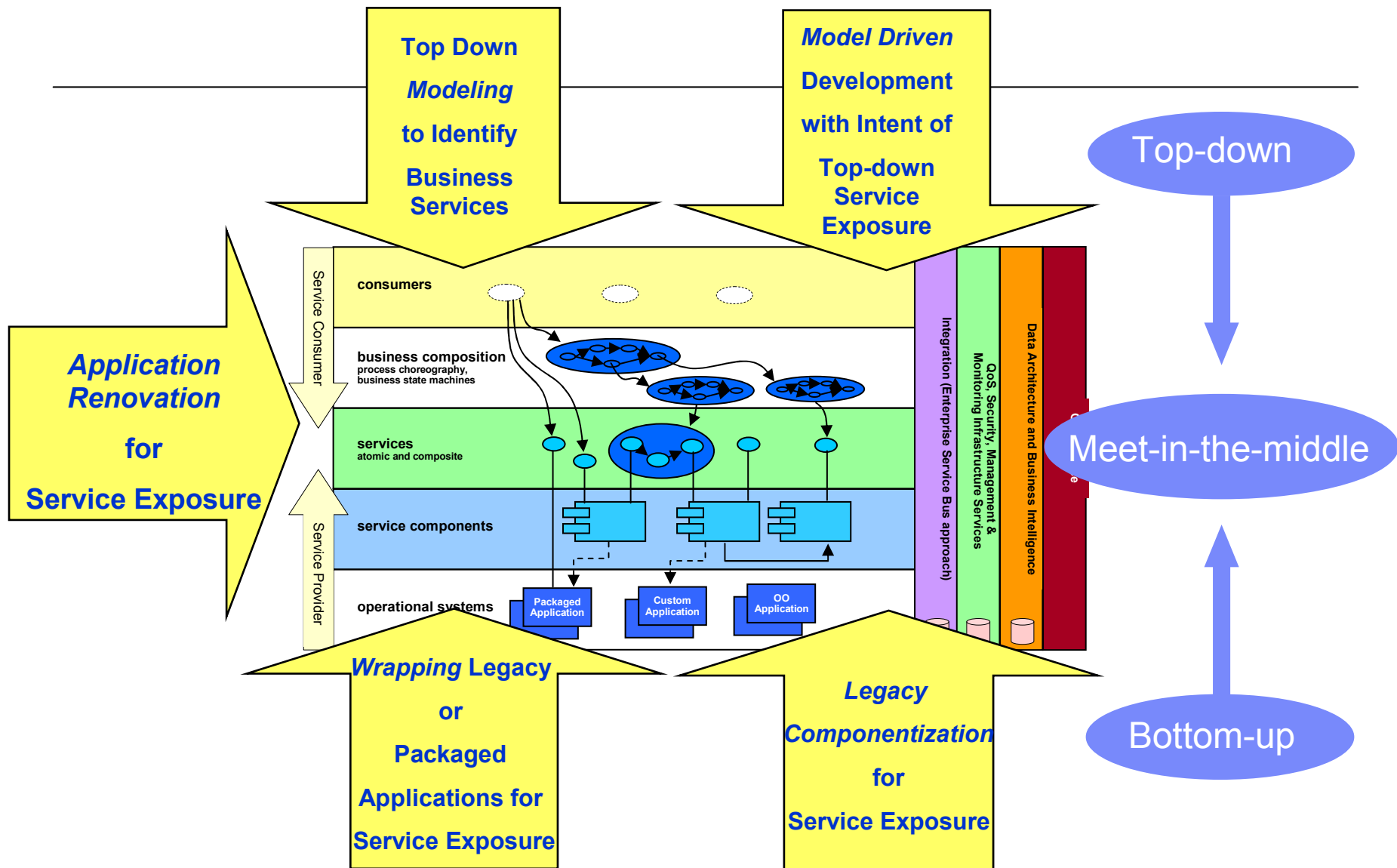


IV. Identification and Specification of Services

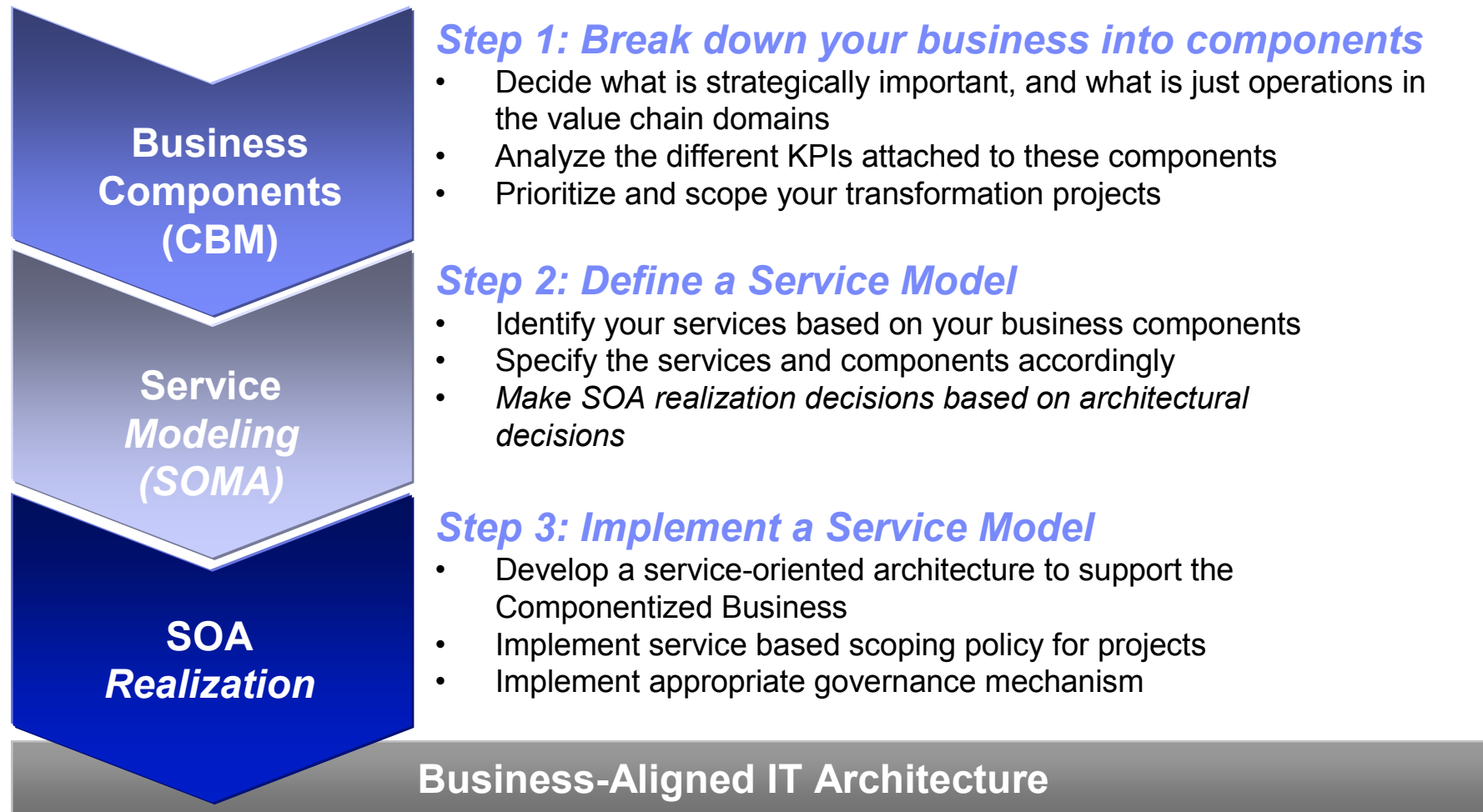
Method Background: The SOA Lifecycle



How: Approaches to SOA Solutions



Best implementations start with business design or a master plan



A CBM (Component Business Model) is a tabular overview of a business

A **Business Component** is a part of an enterprise that has the potential to operate independently, in the extreme as a separate company, or as part of another company.

Columns are **Business Competencies**, defined as large business areas with characteristic skills and capabilities, for example, product development or risk management.

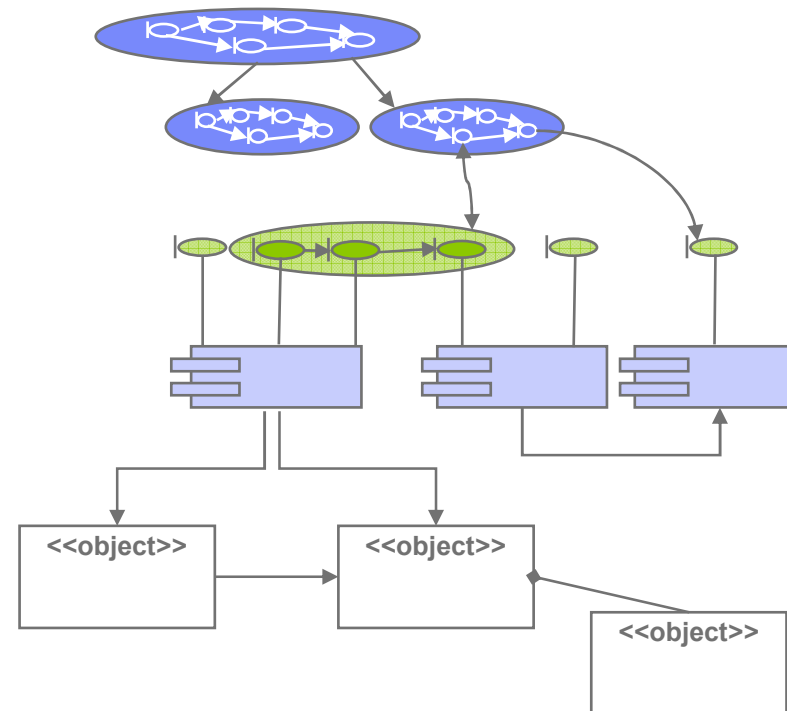
An **Accountability Level** characterizes the scope and intent of activity and decision-making. The three levels used in a component business model are Directing, Controlling and Executing.

- Directing is about strategy, overall direction and policy.
- Controlling is about monitoring, managing exceptions and tactical decision making.
- Executing is about doing the work.

Operational Levels	Competency	Business & Financial Management/ Infrastructure	Manufacturing/ Product Development	Relationship Management	Business Acquisition & Retention	Risk Management	Policy Administration & Services	Benefit Payments	Group Administration	
Direct	Business Strategy			Customer Relationship Strategy	Market and Promotion Planning	Reinsurance Strategy	Operations Management And Planning	Claims Management Strategy	Group Strategy	
	Investment Strategy					Risk, Compliance, Legal Management				
			Product Planning							
Control	Investment/Capital Management		Product Management Internal/External	Correspondence Management	Promotion Management	Risk Compliance, Legal Monitoring	Operations Management Controls	Claims Litigation		
	External Reporting					Underwriting Management			Account Management	
	Procurement/ Vendor Management			Customer Management	Producer Portfolio Analysis					
Execute	AR / AP		Product Development Internal/External	Customer Information Profile	Illustrations		Processing Policy Changes	Surrenders/ Withdrawals		
	Investment Operations									
	Financial Reporting and Controls (GL)					Underwriting Decisioning		Payment Processing (Disbursements)	Enrollment & Eligibility	
	Audit				Sales Support					
	Human Resources		Product Education	Correspondence Handling	New Business Processing (Application)		Billing		Process Claim	Plan Sponsor Services
	Learning/Knowledge Services					Contact Servicing/ Service Centers				
					Regulatory and Compliance Reporting			Fraud Investigation		

Services, service components, and processes are the SOA modeling artefacts to focus the business IT alignment

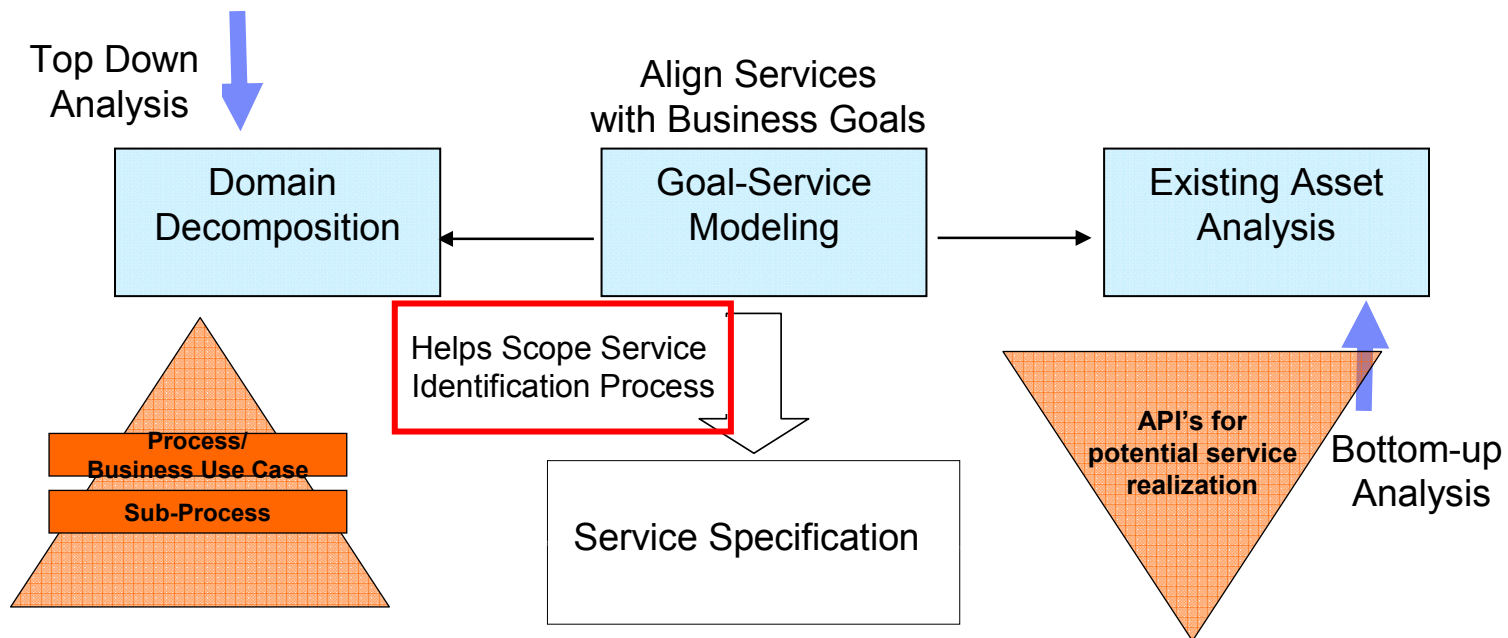
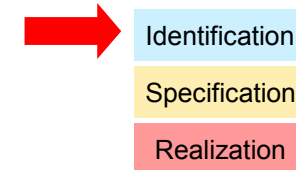
- **Business Processes**
(Consumers: Abstract Flows)
- **Services**
(Atomic and Composite)
- **Service Components**
(Providers: Systems)



- ■ ■ SOMA was created to specifically address modeling (analysis, identification and specification) of all three constructs.

SOMA (Service-Oriented Modeling and Architecture) identifies services through three complementary techniques

- Domain Decomposition (Top Down Analysis)
- Existing Asset Analysis (Bottom-up Analysis)
- Goal-Service Modeling



SOMA Specification uses multiple techniques to select services for exposure, to specify flows, services and services components that realize them

– Service Specification

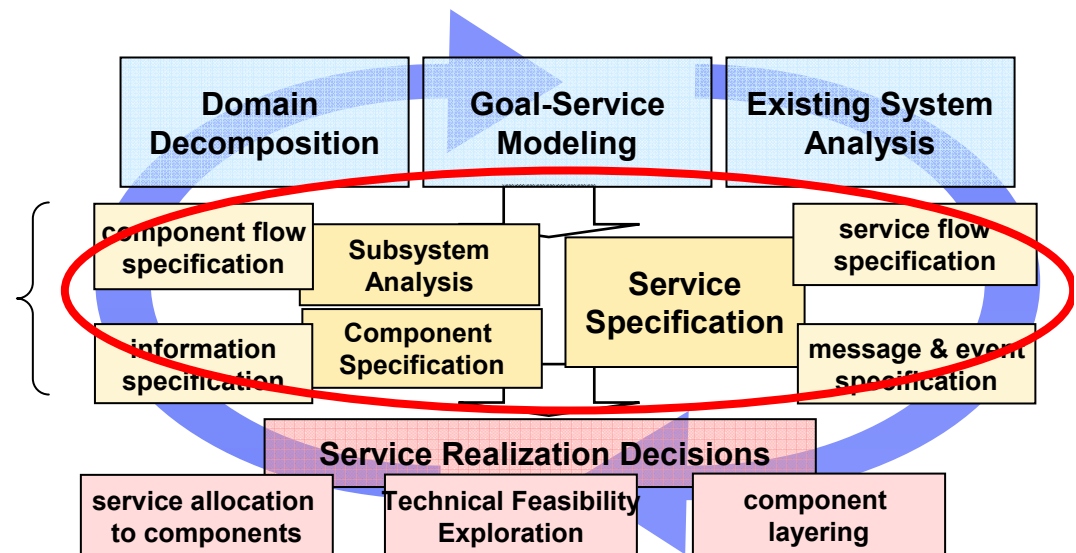
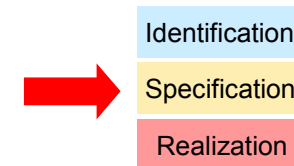
- Elaborates the *Service Model*, Includes **Service Litmus Test** that “gate” service exposure decisions

– Subsystem Analysis

- Partitioning into service components that will be responsible for service realization

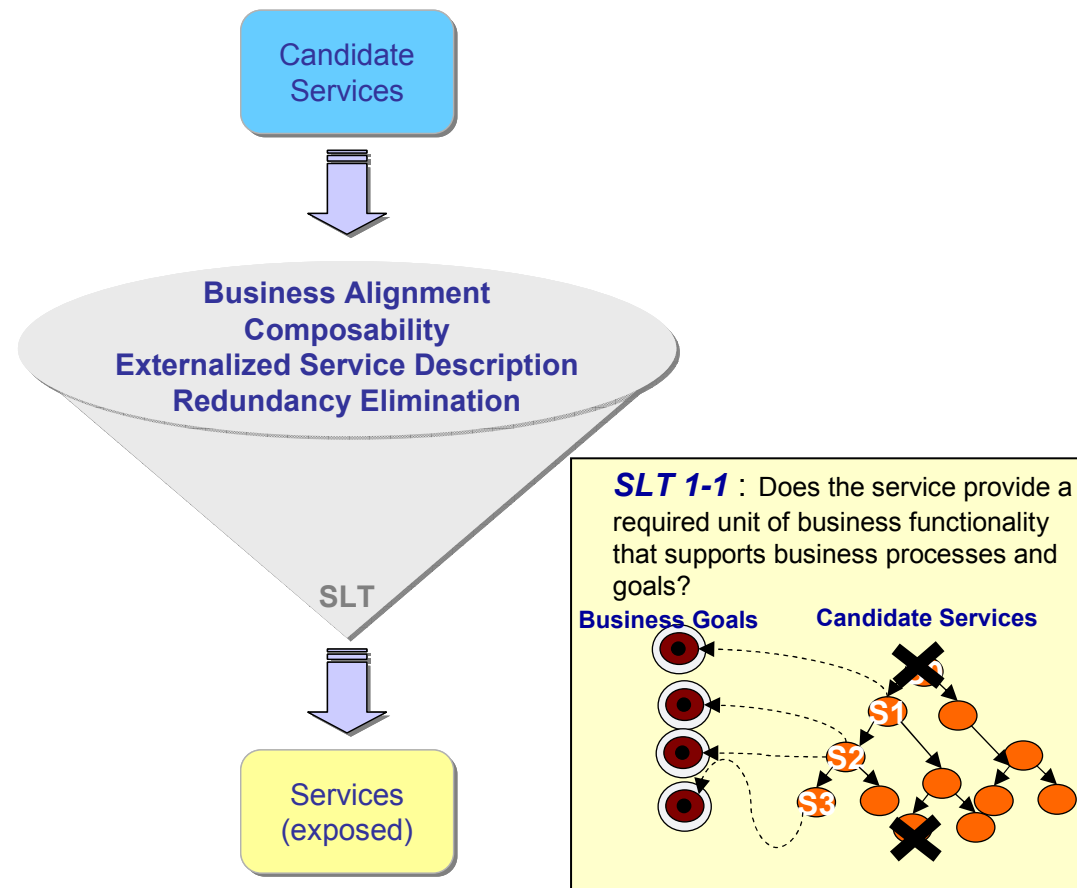
– Component Specification

- Detailed component modeling, flow, information architecture, messages

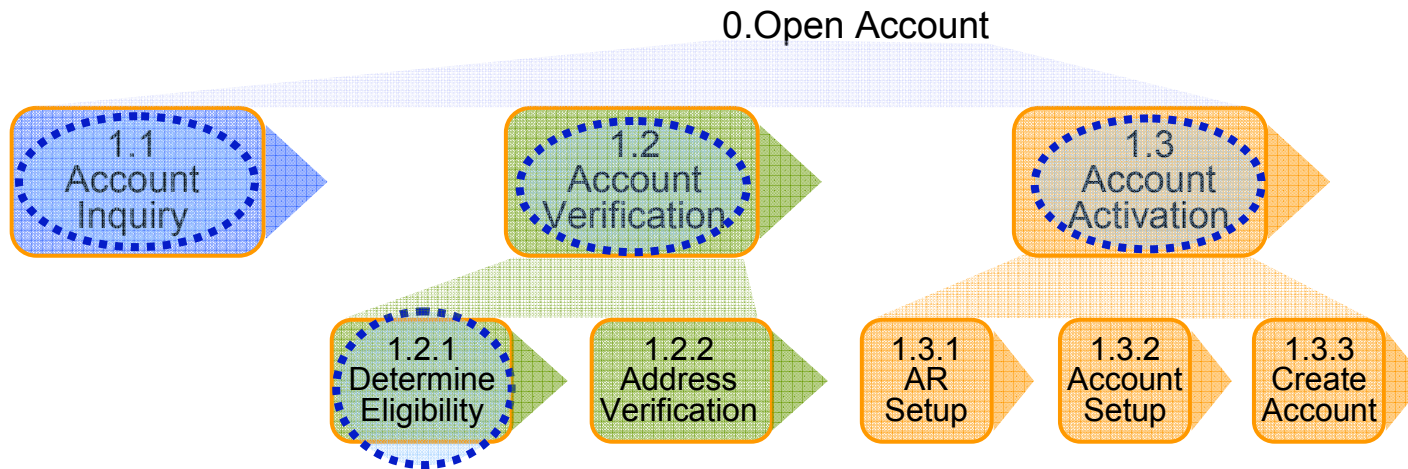


Service Litmus Test: During the Service Specification we make service exposure decisions

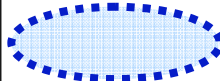
- “From all the candidate services, which ones should we expose?”
- Not all candidate services should be exposed
- Every implemented service has costs and risks
- “**Service Litmus Test**” helps make exposure decisions



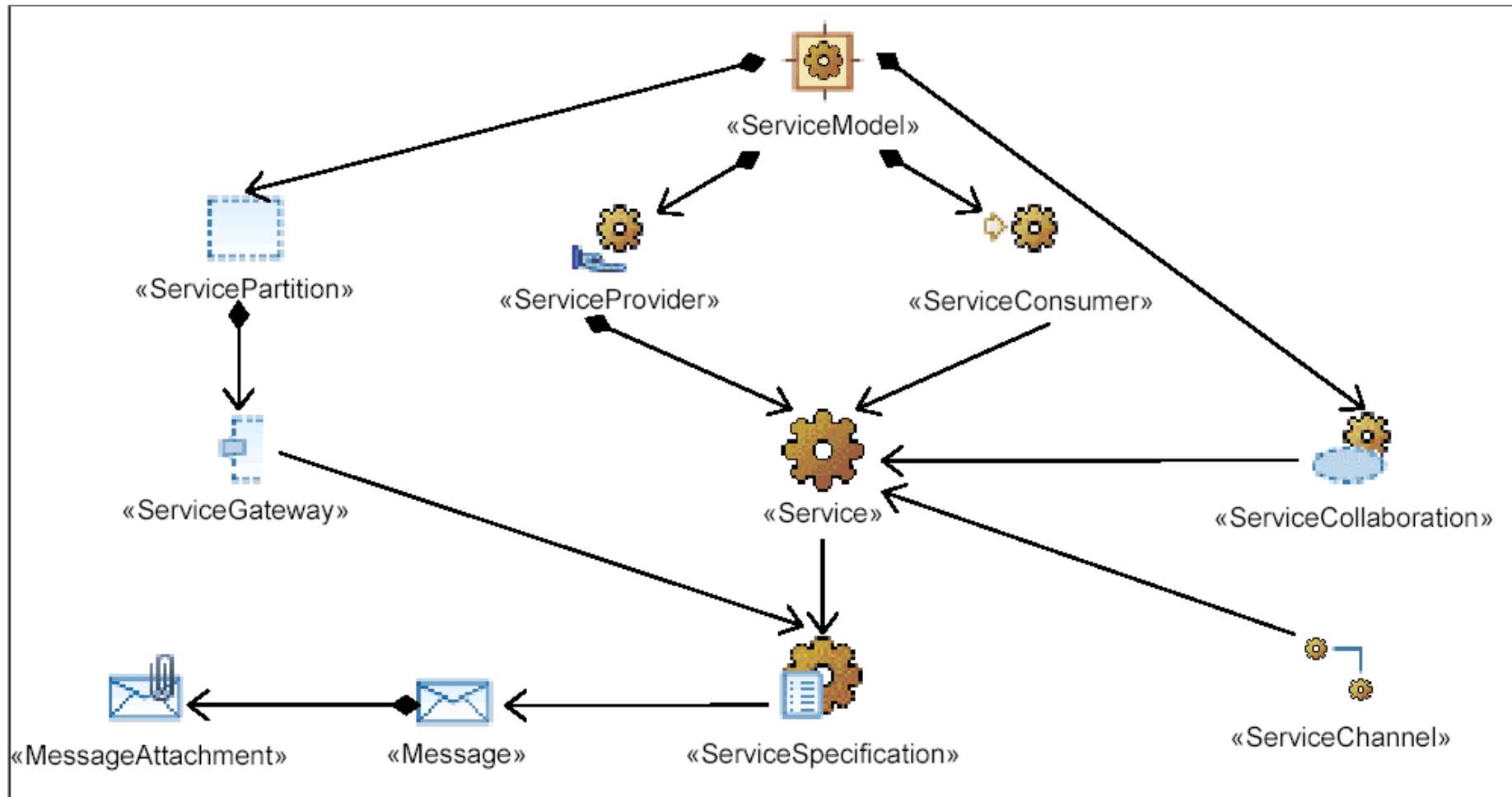
JK Enterprises Service Exposure Decisions



EXAMPLE
For illustration only

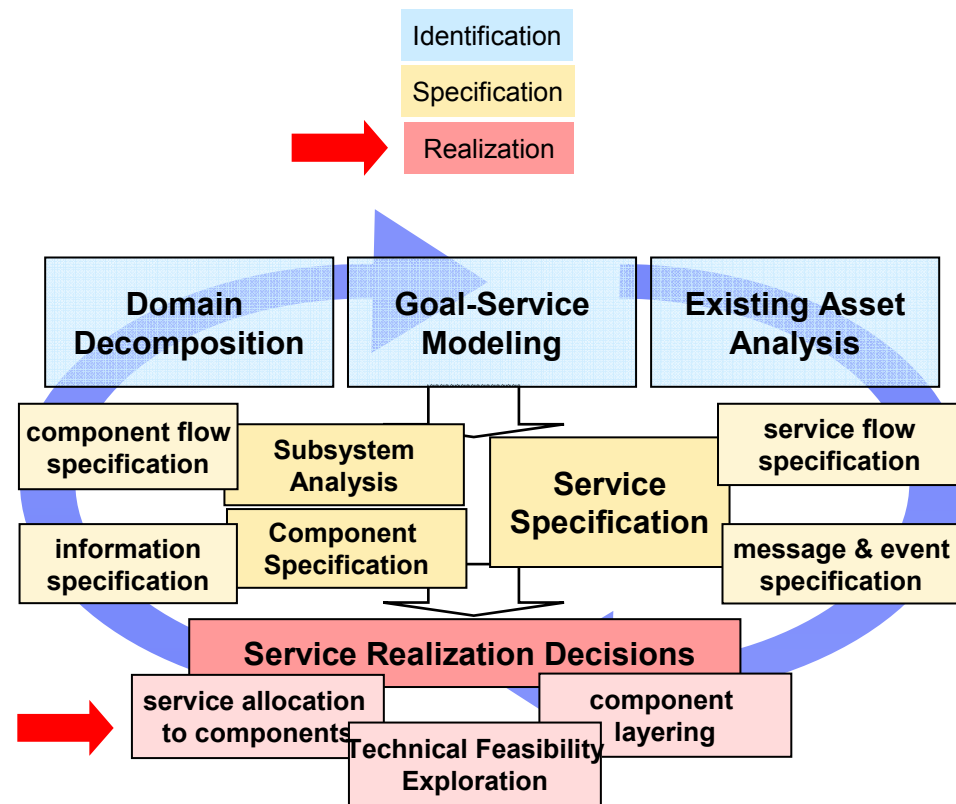
 **Legend**
= Service to be exposed

UML 2 Profile for Software Services Elements

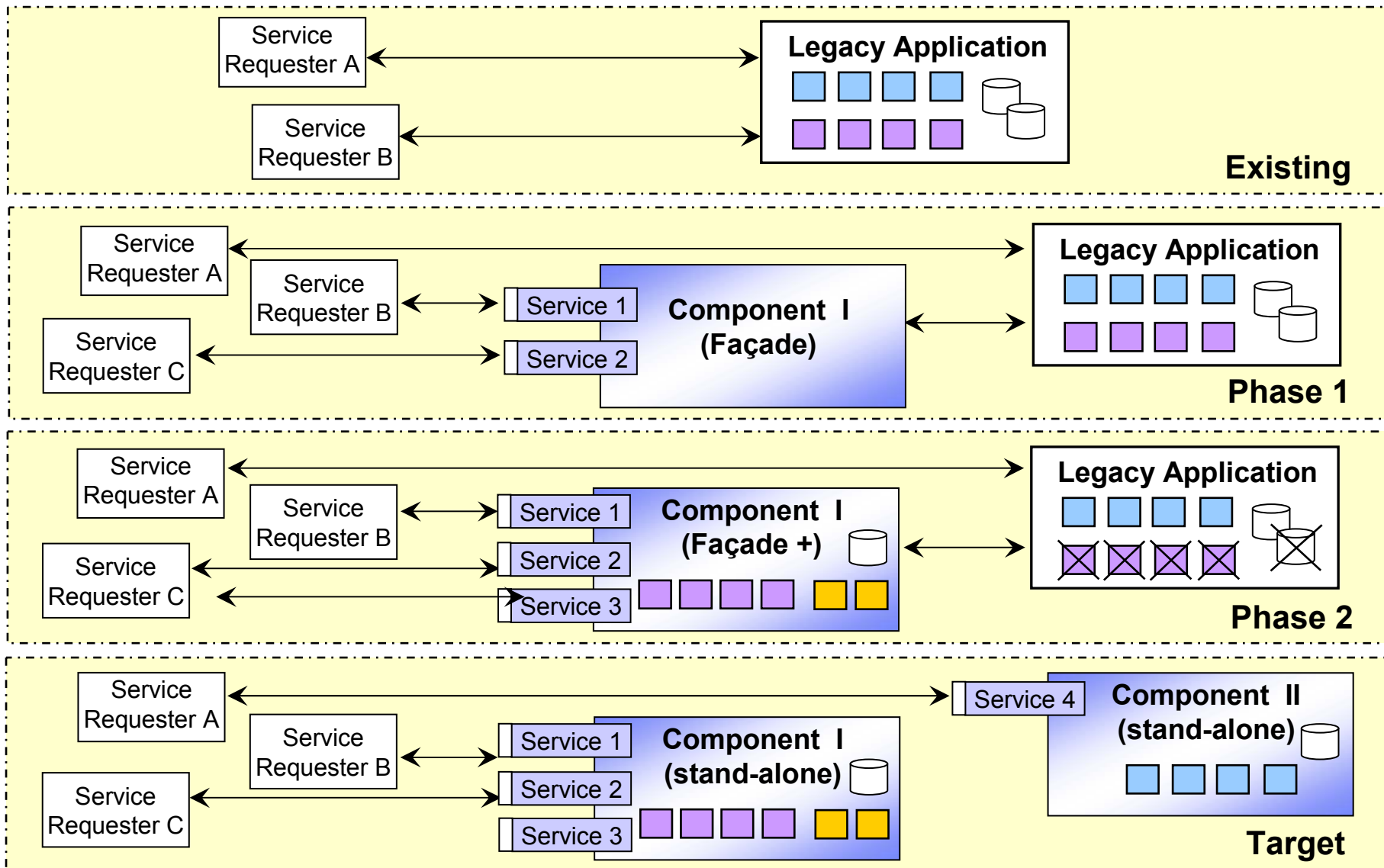


SOMA Service Realization Step

- **Component Layering**
 - **Allocation of component to the application architecture layers**
- **Allocation of services to service components**
- **Technical Feasibility Exploration**
- **Realization Decisions with Justifications**



Legacy Migration Example



Positioning of Portal as general User Interface

user perspective

- Personalization
- Customization
- Navigation
- Single Sign On
- People Awareness

IT perspective

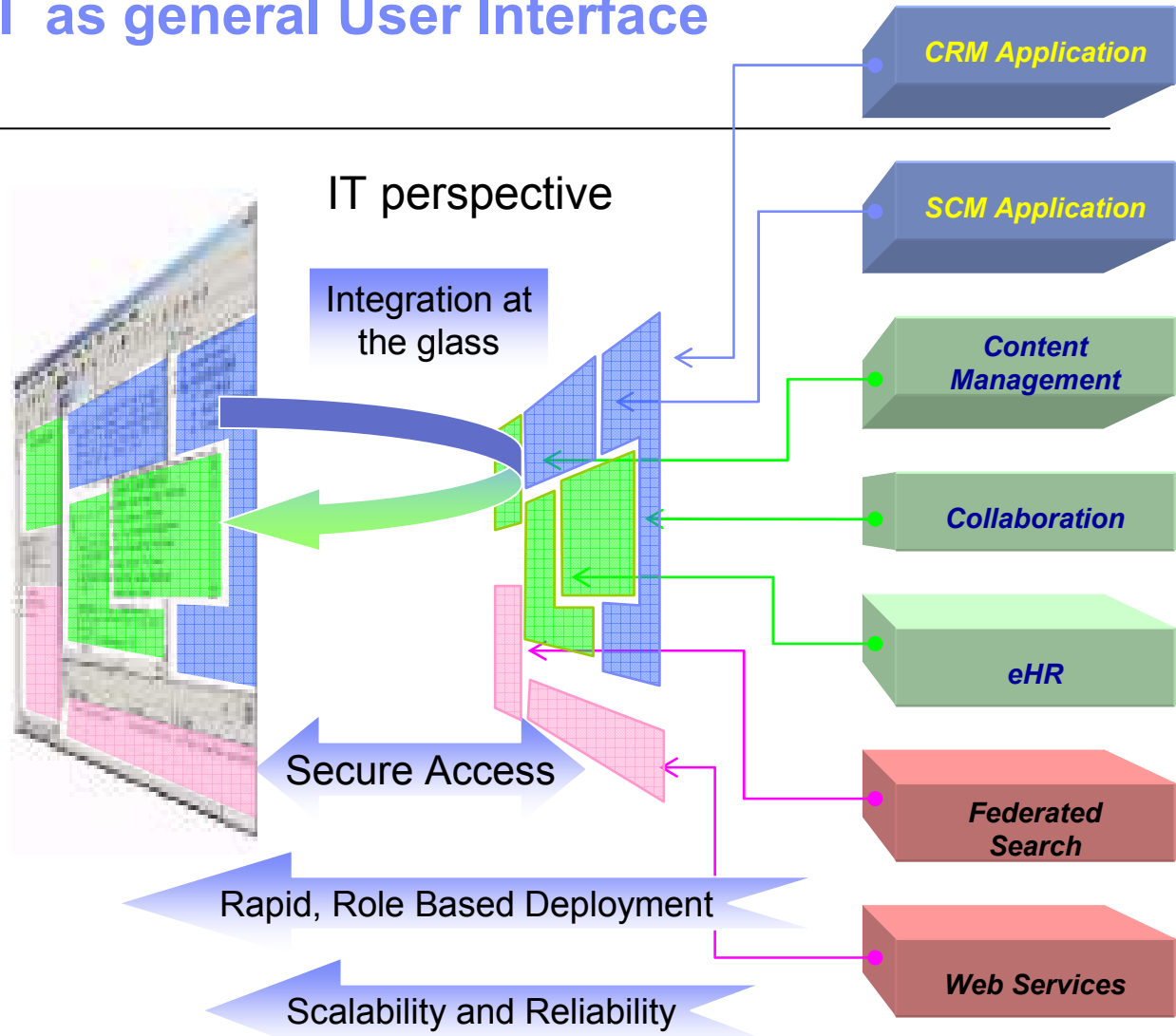
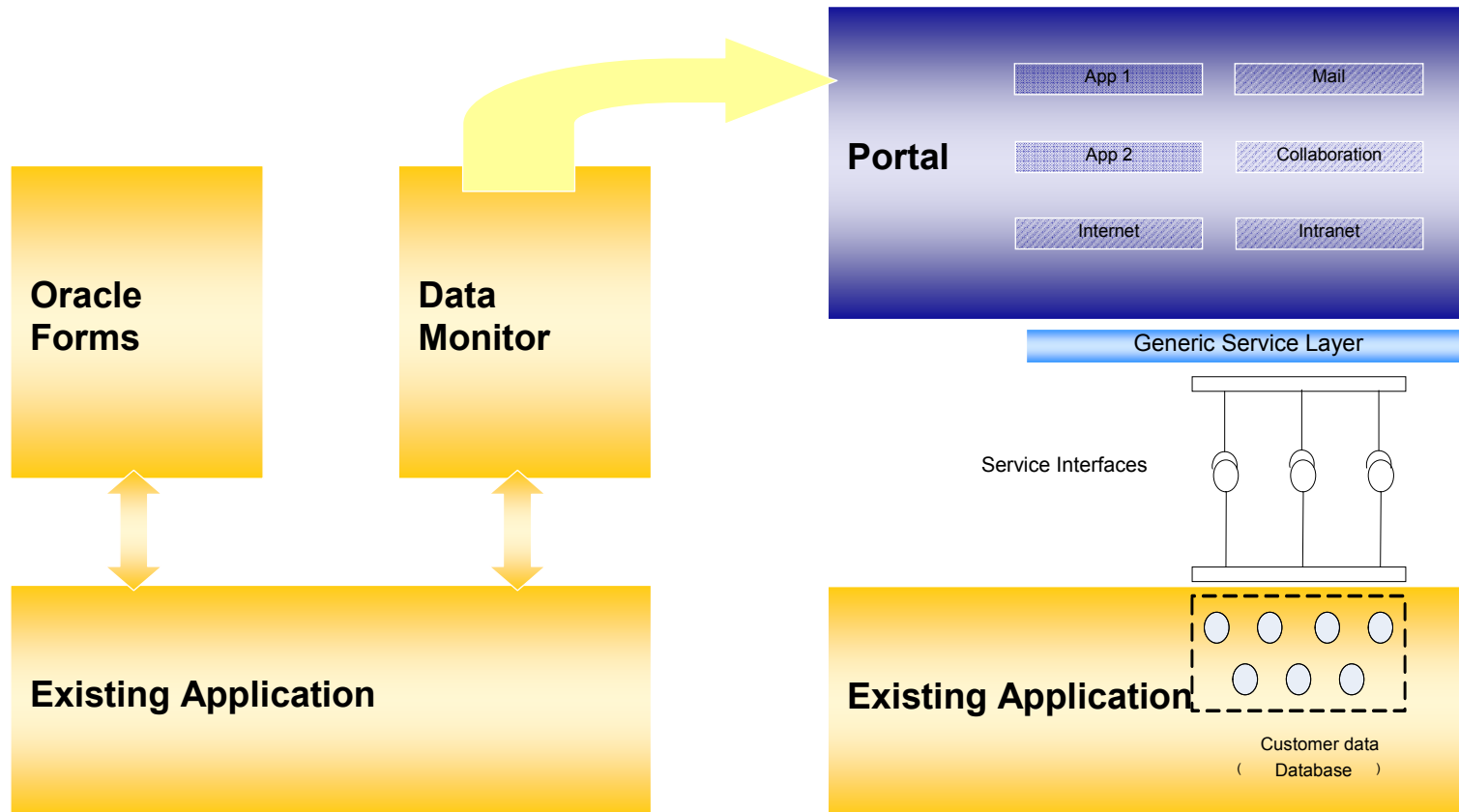


Illustration of an Portal Implementation

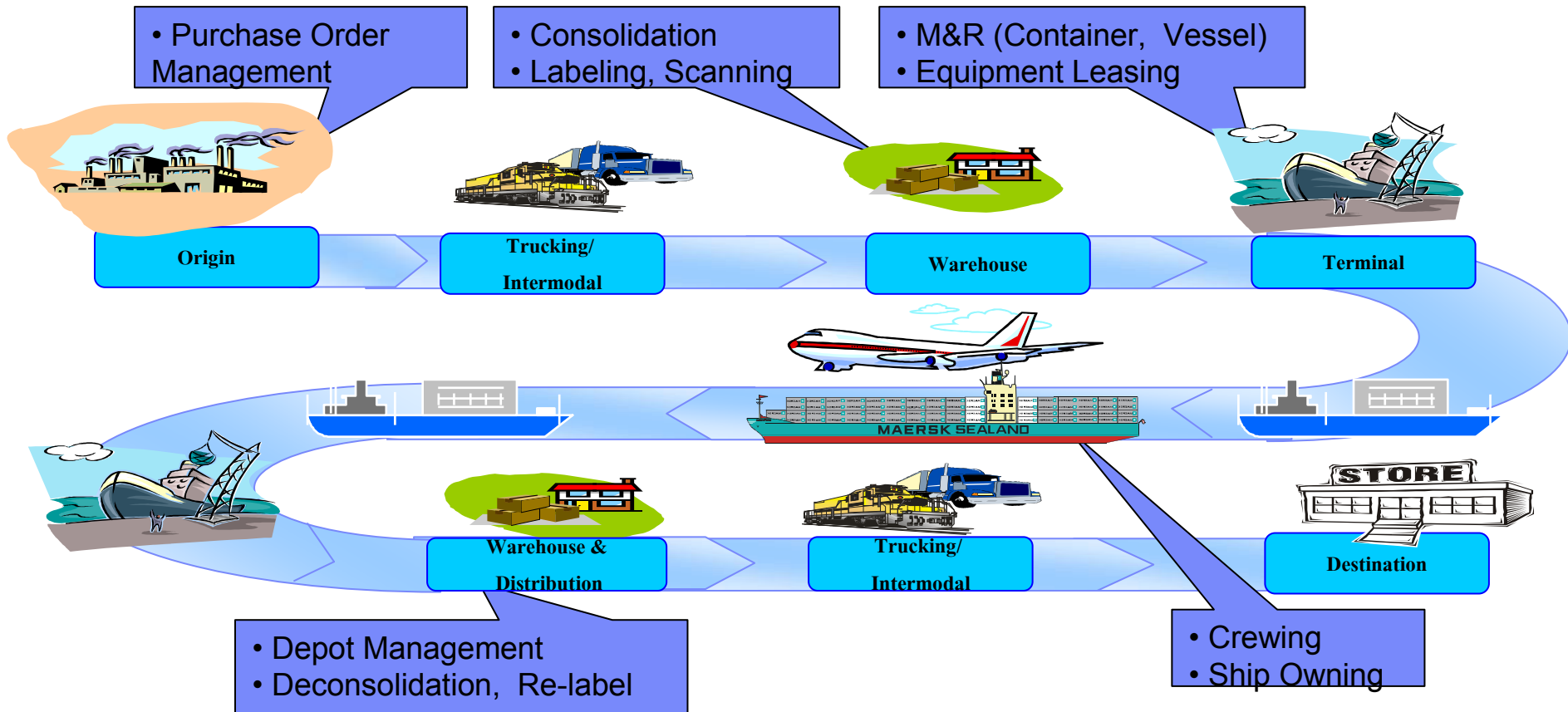


V. Case Study 2
Integration Architecture for FACT

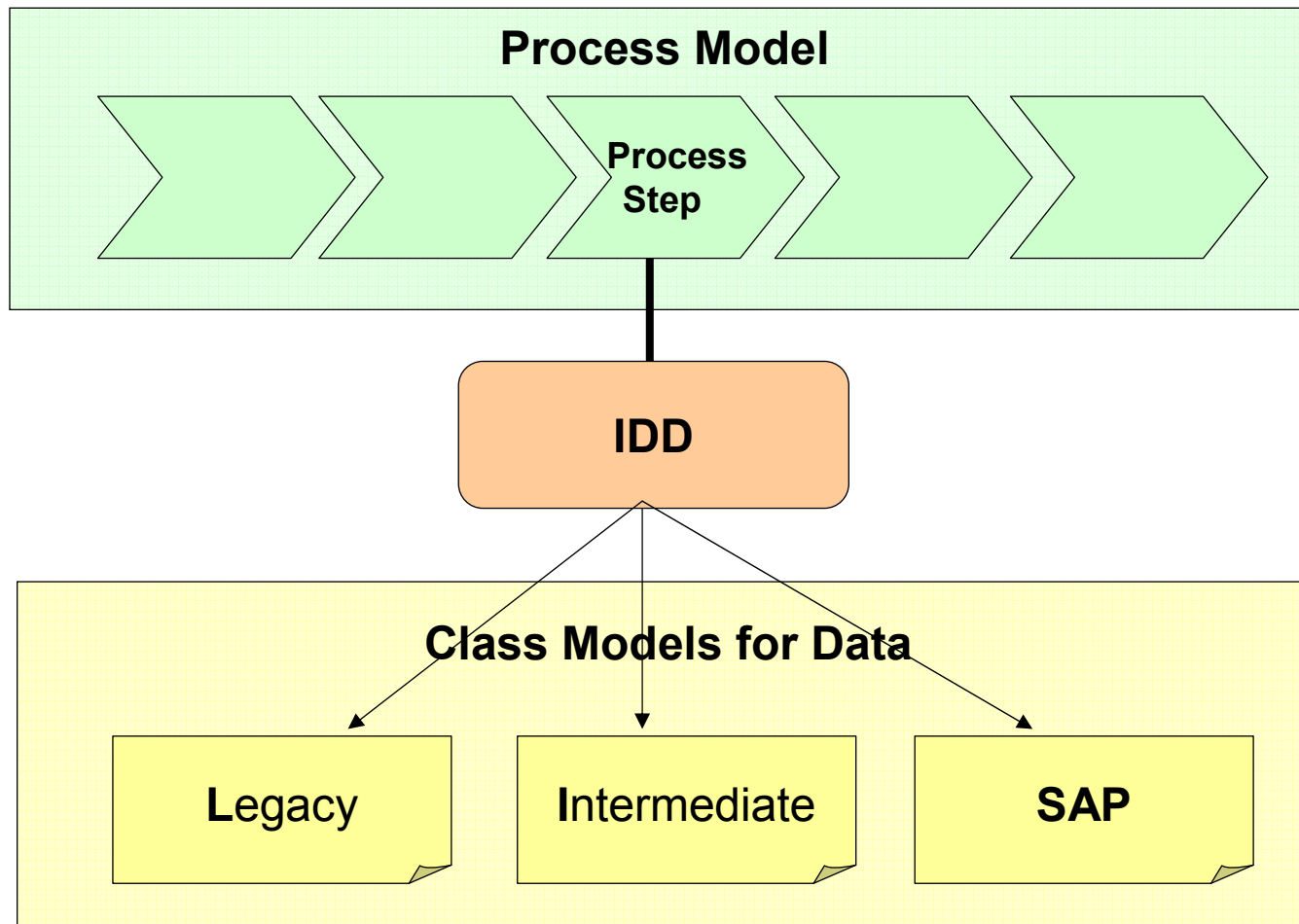
Financial Accounting for Container Business

- Financial Accounting performed by SAP
- Long lasting Business Processes – from taking the order to delivery at destination
- Global IT environment (140 countries)
- Integration of approx. 40 operational applications

Container Business Value Chain



Process Model describing IDD (Interface Detailed Definitions)



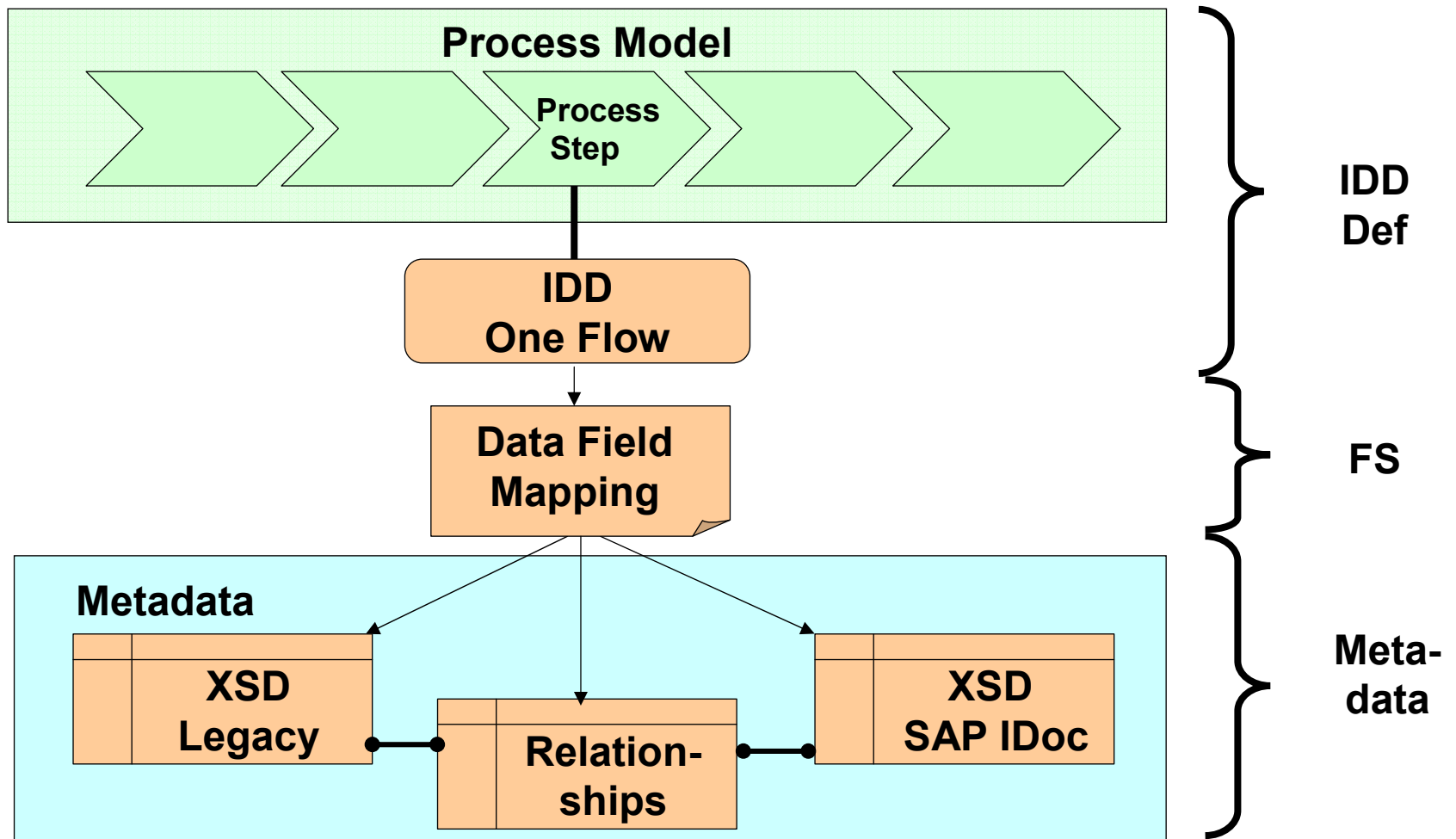
Overview

- IDD (Interface Detailed Definition) related to a step in the Business Process – may include multiple information flows

- FS (Functional Specification)
 - *Functional Specification* (FS) defines data (SAP and Legacy) from Business Perspective, including some transformations
 - Data Structure include IDoc / BAPI as well as Legacy interface data structure to be used and or customized

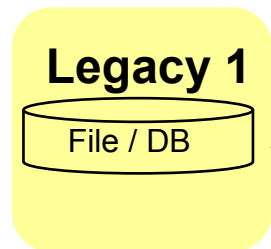
- TS (Technical Specification)
 - *Technical Specifications* (TS) (Overview, SAP, Legacy, Broker) include *AND data models* and XSD (XML Schema Definition) for XML of SAP IDocs, XML of Legacy Application Interfaces, Transformations

Deliverables of Data Structures for an IDD

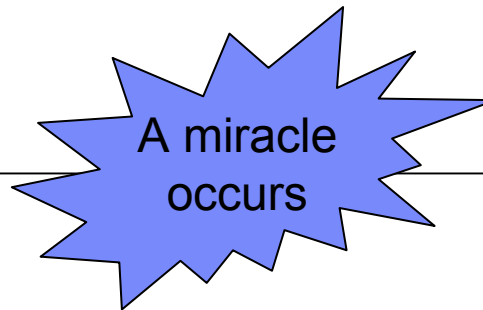


Integration Architecture is about breaking “Interfaces” into smaller chunks

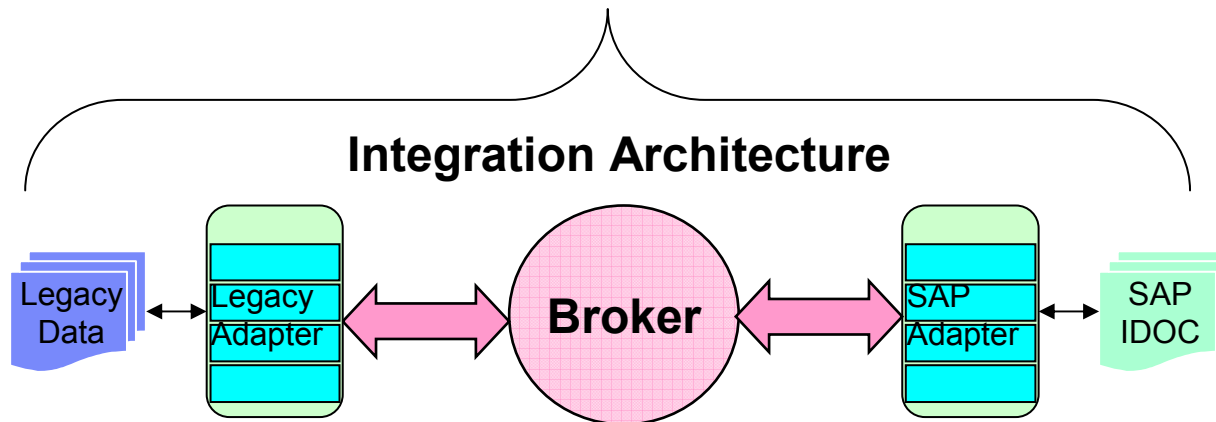
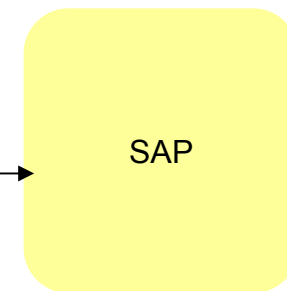
Legacy Systems



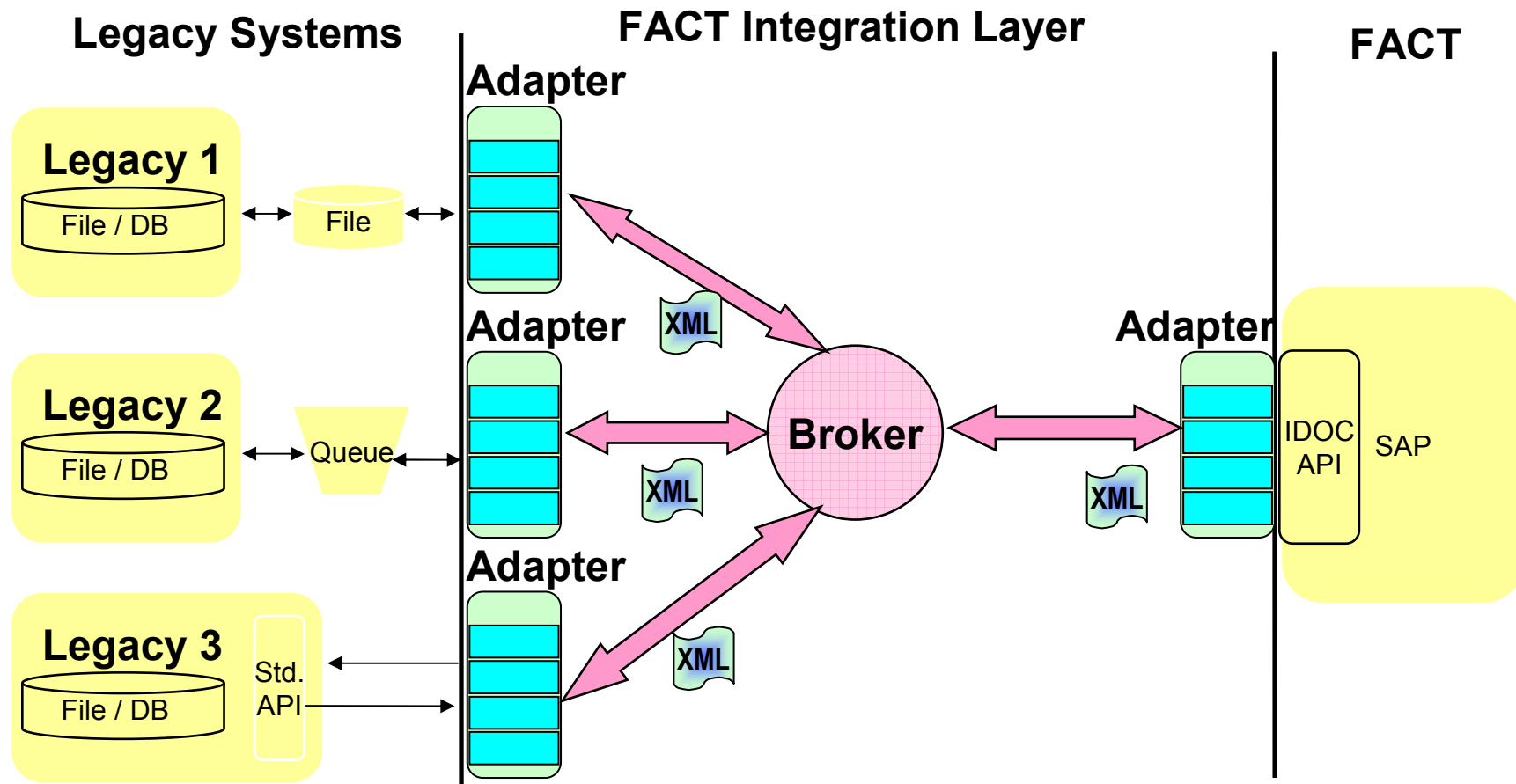
Interface



FACT



FACT Integration Layer – Technical View



Case Study 2

- Apply SOA Principles to the described environment
- Define SOA Infrastructure
- Explore the use of Business Process Management
- Present a SOA Solution for FACT