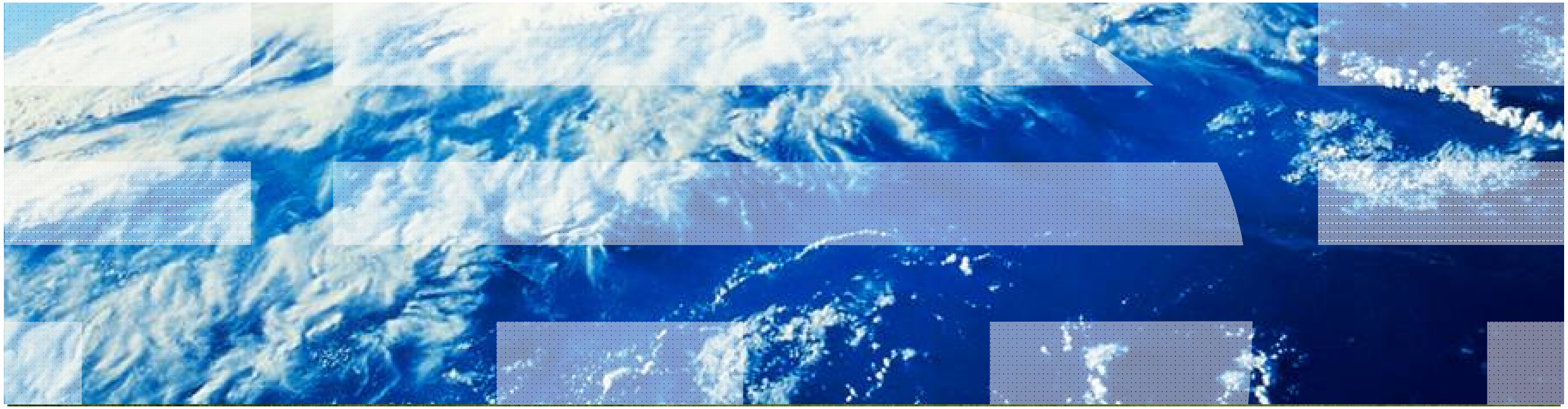
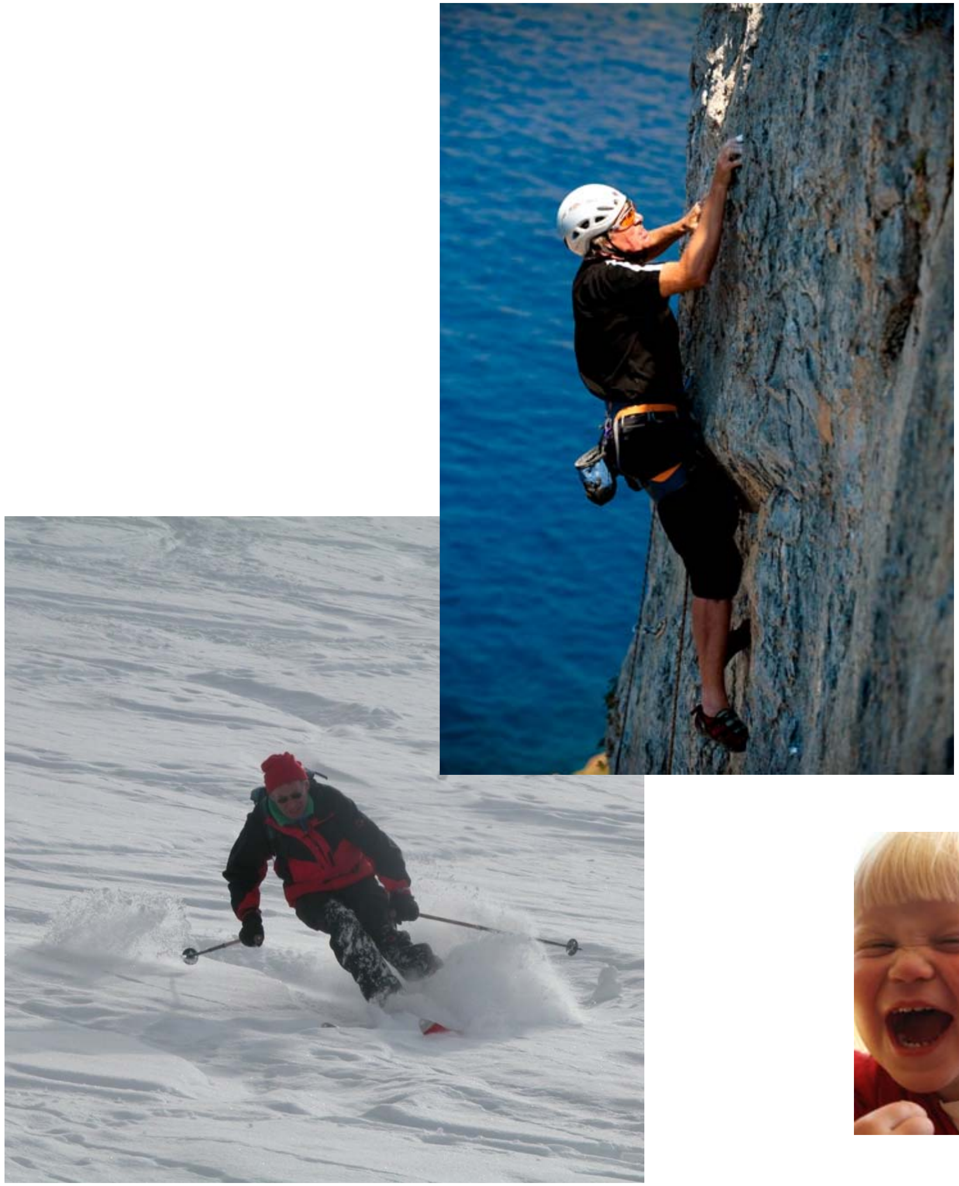
Dr. Hans-Peter Hoidn
Executive IT Architect, IBM Software Group
Global Business Integration "Tiger" Team

Enterprise IT Architectures

SOA Part 1



Hans-Peter Hoidn

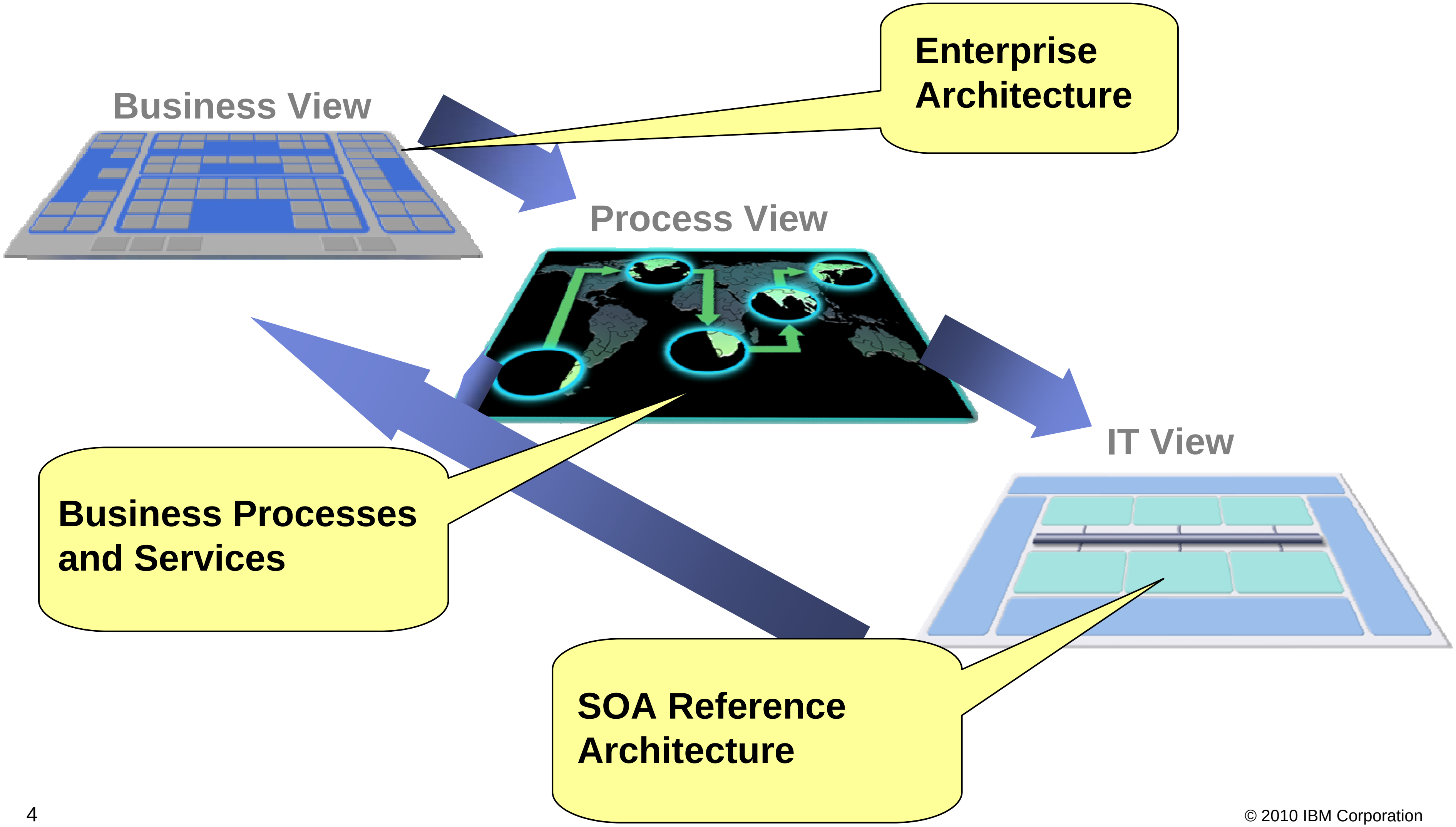


- Executive IT Architect in Global Business Integration “Tiger” Team
- Almost 10 years IBM (BCS and SWG) and PwC
- Previously with UBS, Digital Equipment Corp, etc.
- Architect since about 18 years
- Starting programming 1971
- Dr. sc. math ETH

three daughters (35, 33, 30 years old); two granddaughters (born January 2008, August 2009) and one grandson (born June 2010)

Positioning

Different views for a Holistic Approach Aligning Strategy with Business and IT Execution



Enterprise Architecture

Business Processes and Services

SOA Reference Architecture

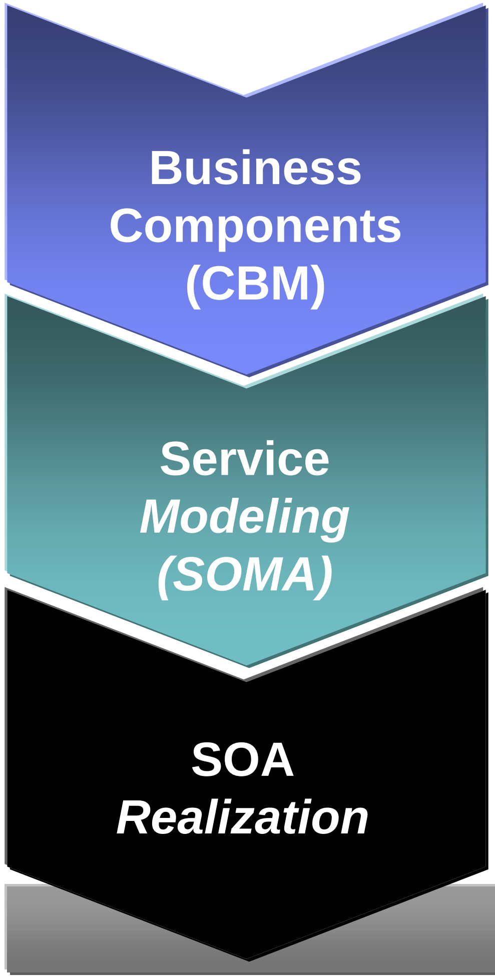
Different Views influencing the Implementation

- **IT View**
  - **Where we usually work on**

- **Process View**
  - **The view of the Business Actions**
  - **Essential for linking Business and IT**
  - **Business Process Management as new discipline**

- **Business View**
  - **The view of your stakeholder and sponsor**
  - **Addressing the needs of the company**
  - **Essential to link to this level**

Top-Down (Ideal) Approach for SOA Start with Business Design



Step 1: Break down your business into components

- Decide what is strategically important, and what is just operations in the value chain domains
- Analyze the different KPIs attached to these components
- Prioritize and scope your transformation projects

Step 2: Define a Service Model

- Identify your services based on your business components
- Specify the services and components accordingly
- *Make SOA realization decisions based on architectural decisions*

Step 3: Implement a Service Model

- Develop a service-oriented architecture to support the Componentized Business
- Implement service based scoping policy for projects
- Implement appropriate governance mechanism

Business-Aligned IT Architecture

Agenda for SOA (Service Oriented Architecture)

- **Day 1 (19 November 2010)**
  - **Introduction - Key Models and Methods for SOA**
  - **Methodology for Identification and Specification of Services**
  - **SOA Layered View**
- **Day 2 (26 November 2010)**
  - **IT View**
  - **SOA Reference Architecture**
  - **BPM (Business Process Management)**
- **Later: Student Presentations, Enterprise Architecture including CBM, Architecture Management including SOA Governance**

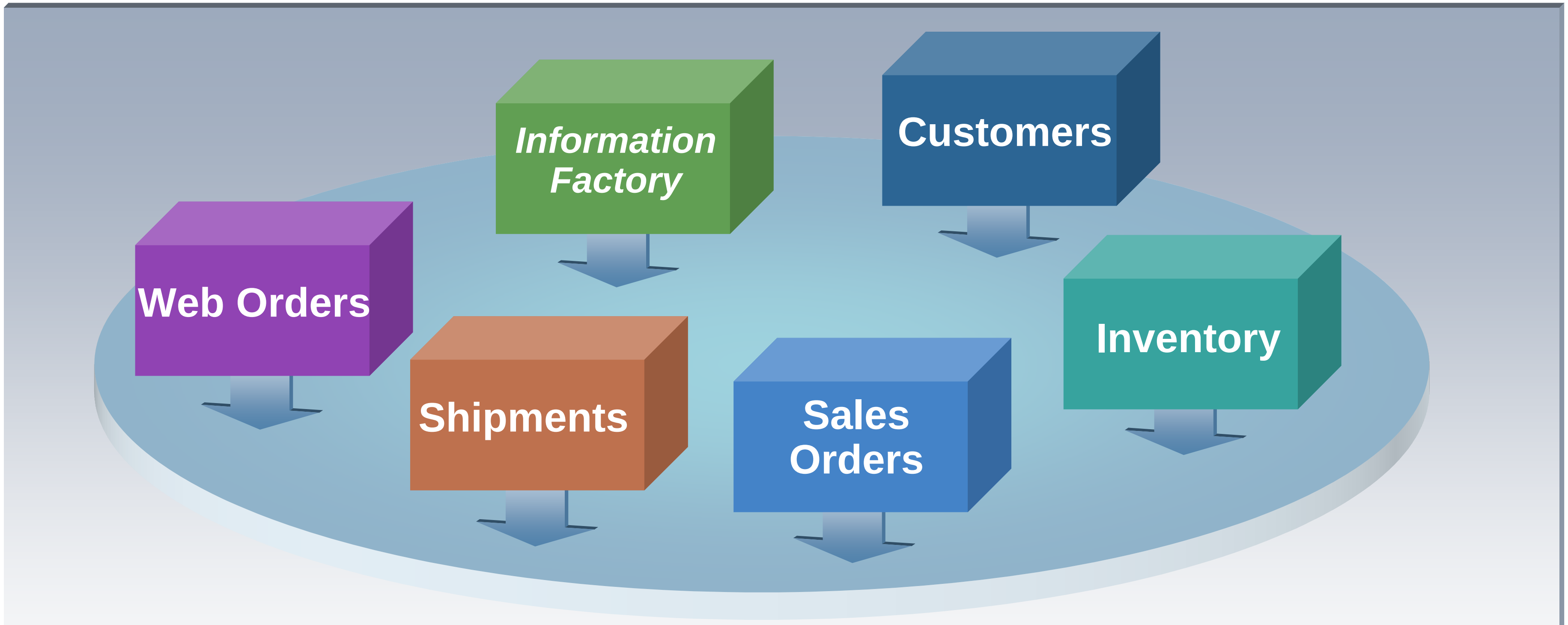
SOA Introduction

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

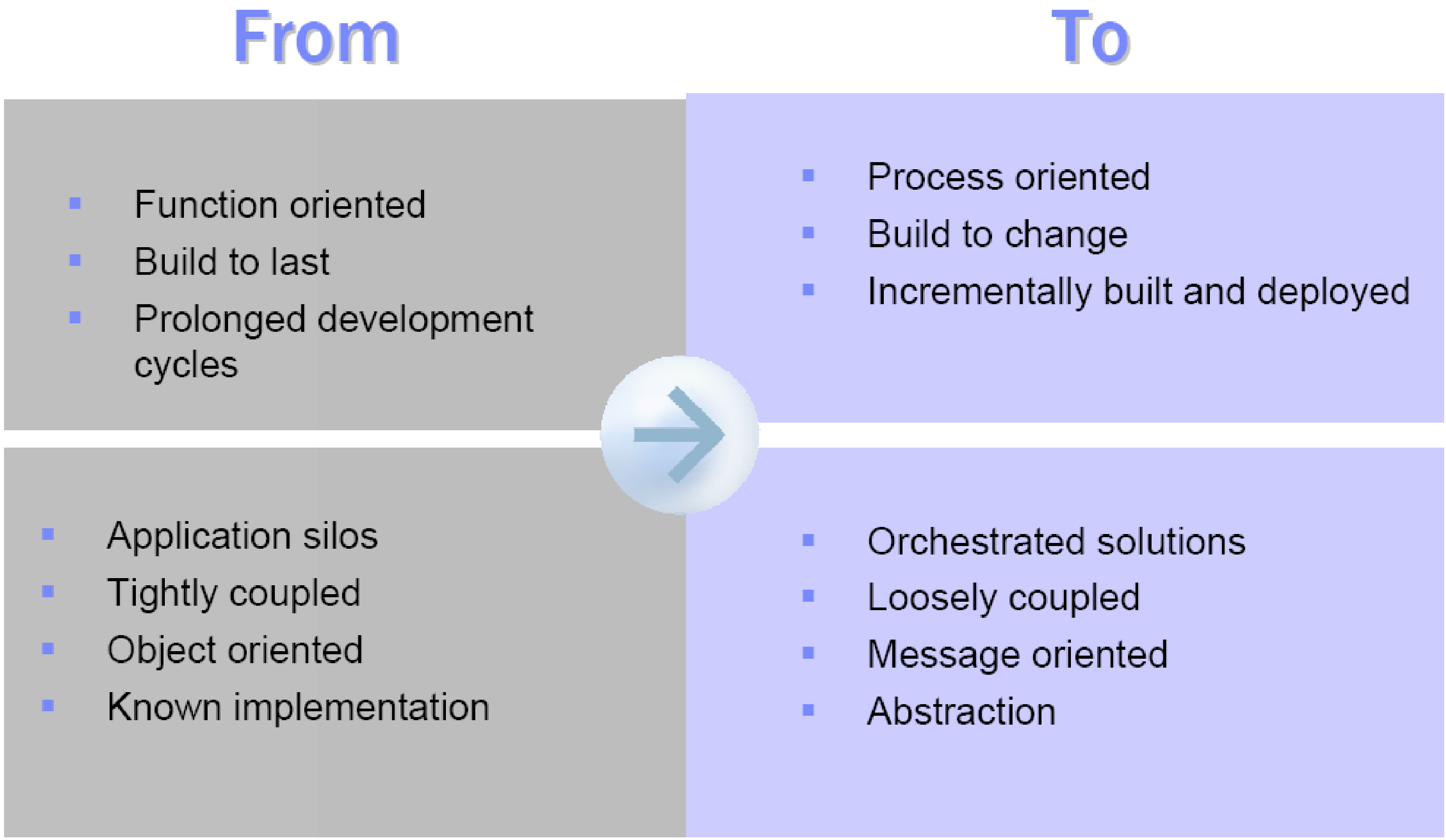


Services defined as units of business logic separated from…

- 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

- Function oriented
- Build to last
- Prolonged development cycles

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

To

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

- Orchestrated solutions
- Loosely coupled
- Message oriented
- Abstraction

What is SOA

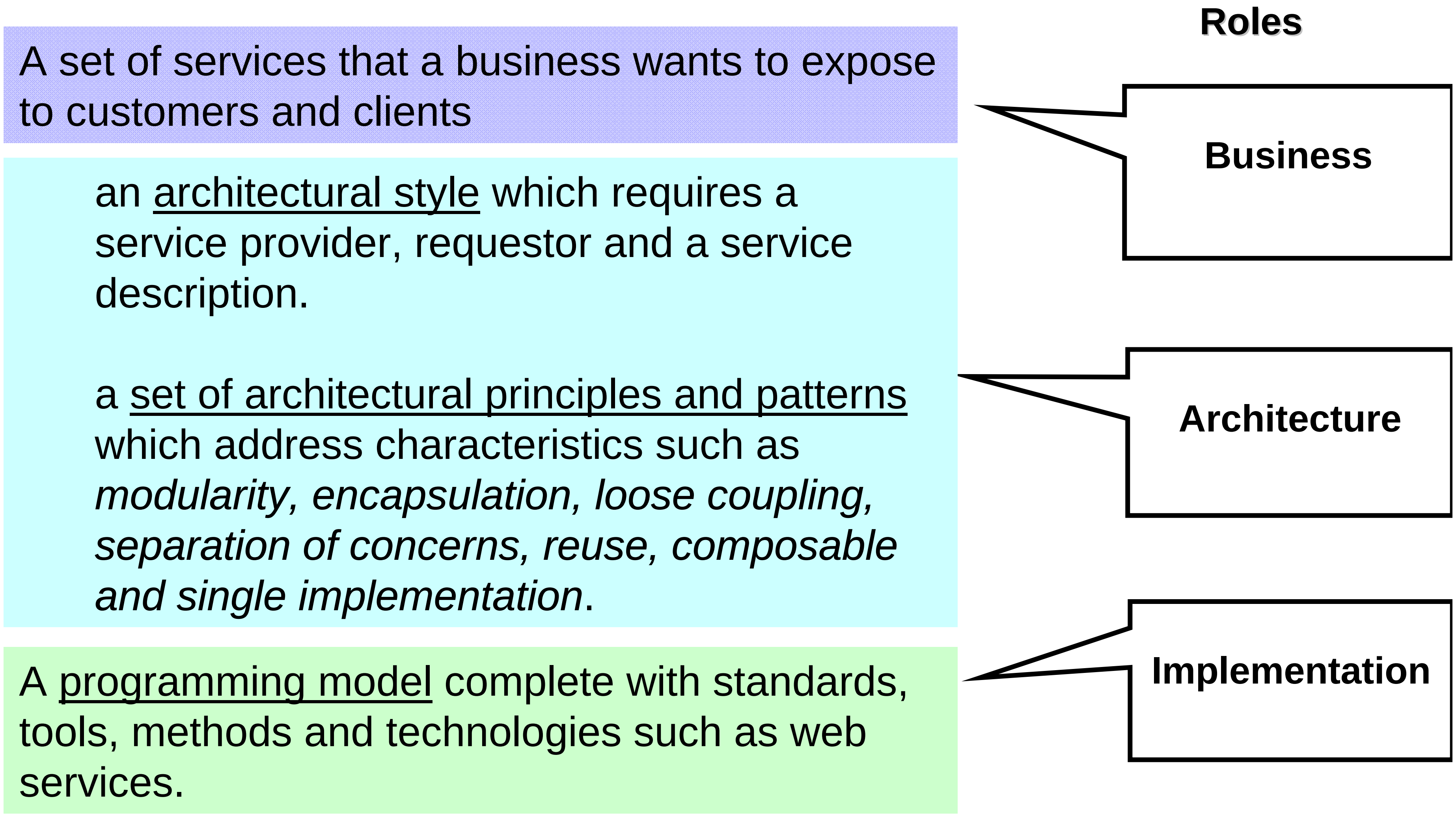
- **SOA is an *architectural style* or approach whose goal is to achieve loose coupling among interacting software agents**

- **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 is different things to different people



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

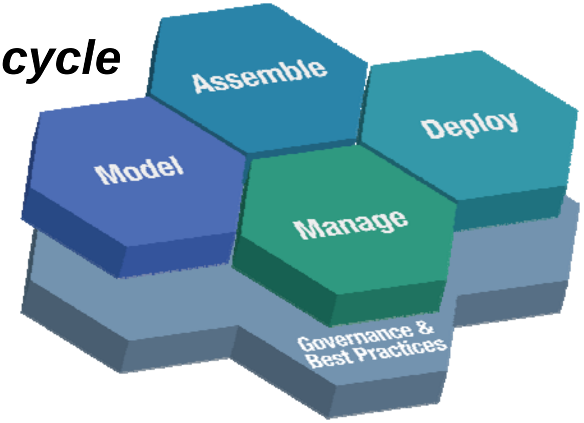
an architectural style which requires a service provider, requestor and a service description.

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

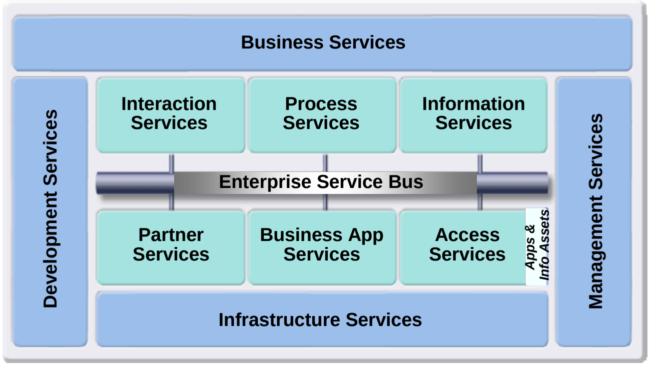
A programming model complete with standards, tools, methods and technologies such as web services.

Key Models and Methods for SOA – Enabling greater flexibility in Enterprise IT Architectures

SOA Lifecycle

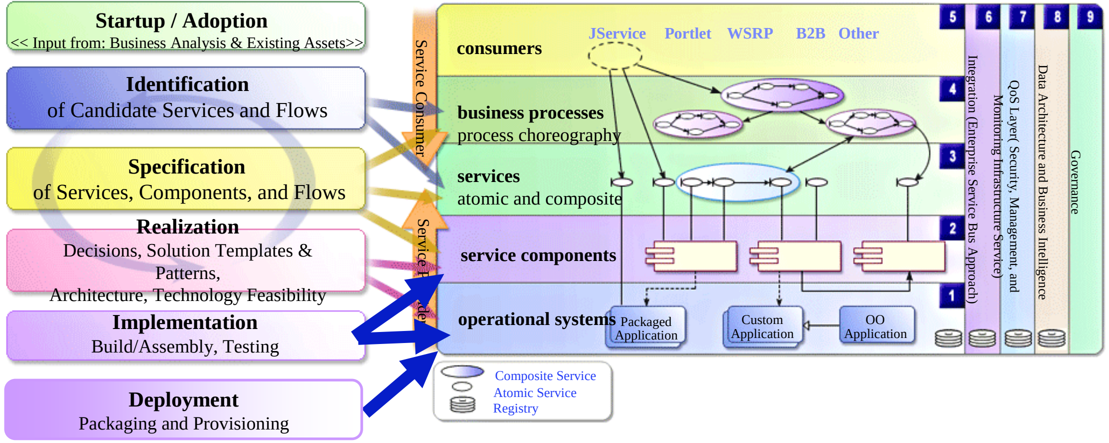


SOA Reference Architecture



The SOMA Method: Service-Oriented Modeling and Architecture

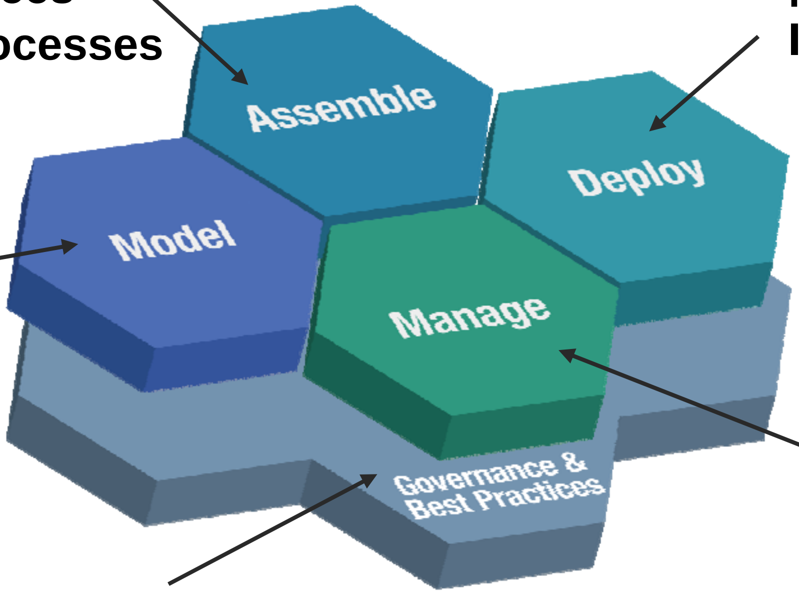
The SOA Solution Stack: Layered solution view



The SOA Lifecycle

- Discover Services
- Construct & Test Services / UIs
- Compose Services
- Orchestrate Processes

- Integrate People
- Integrate Processes
- Manage and integrate Information

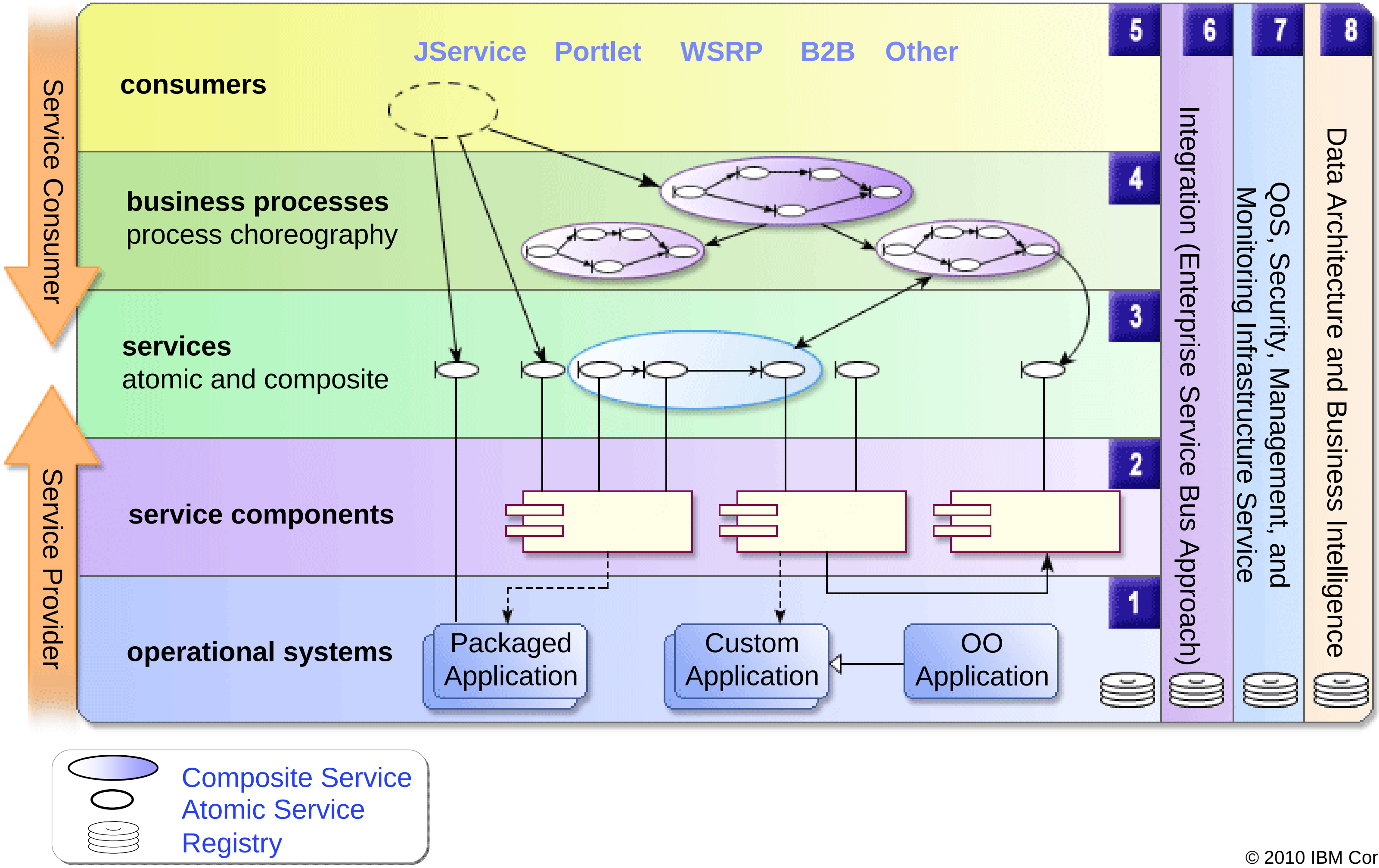


- Gather & Model Requirements
- Model & Simulate
- Model SW Architecture

- Financial Transparency
- Business/IT Alignment
- Process Control

- Manage Applications & Services
- Manage Identity & Compliance
- Monitor Business Metrics

SOA Solution Stack

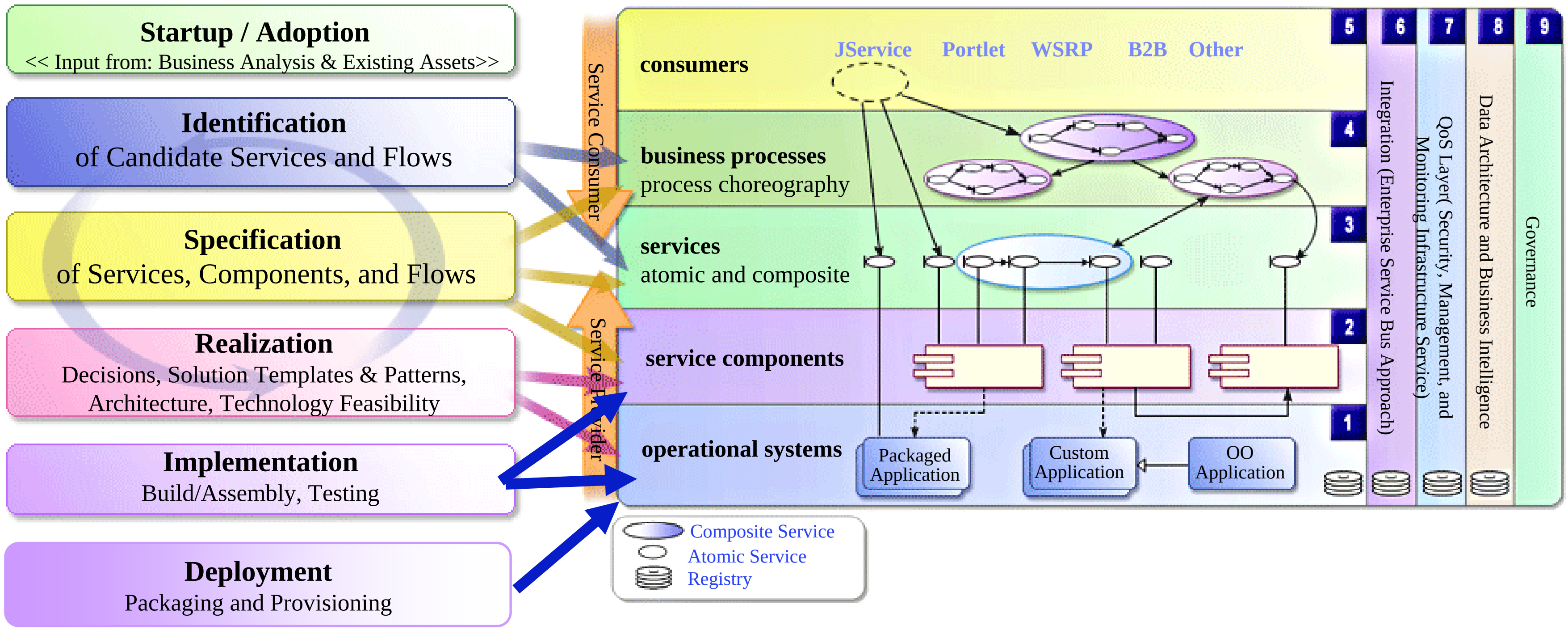


SOMA (Service Oriented Modeling and Architecture) provides SOA Methodology

SOMA is about identification, specification, realization, implementation, and deployment of services, components and flows

SOMA Method

SOA Solution Stack



Startup / Adoption
<< Input from: Business Analysis & Existing Assets>>

Identification
of Candidate Services and Flows

Specification
of Services, Components, and Flows

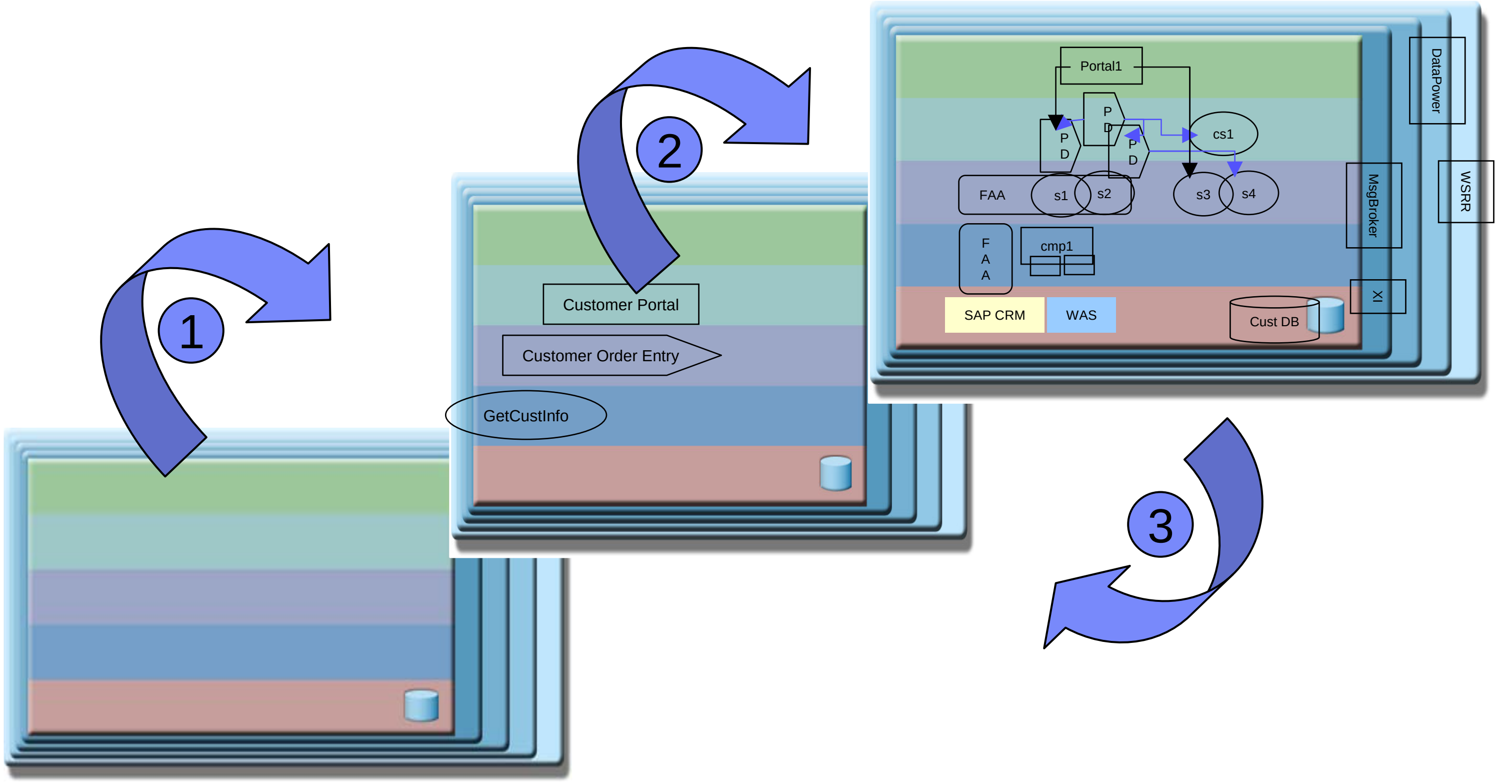
Realization
Decisions, Solution Templates & Patterns, Architecture, Technology Feasibility

Implementation
Build/Assembly, Testing

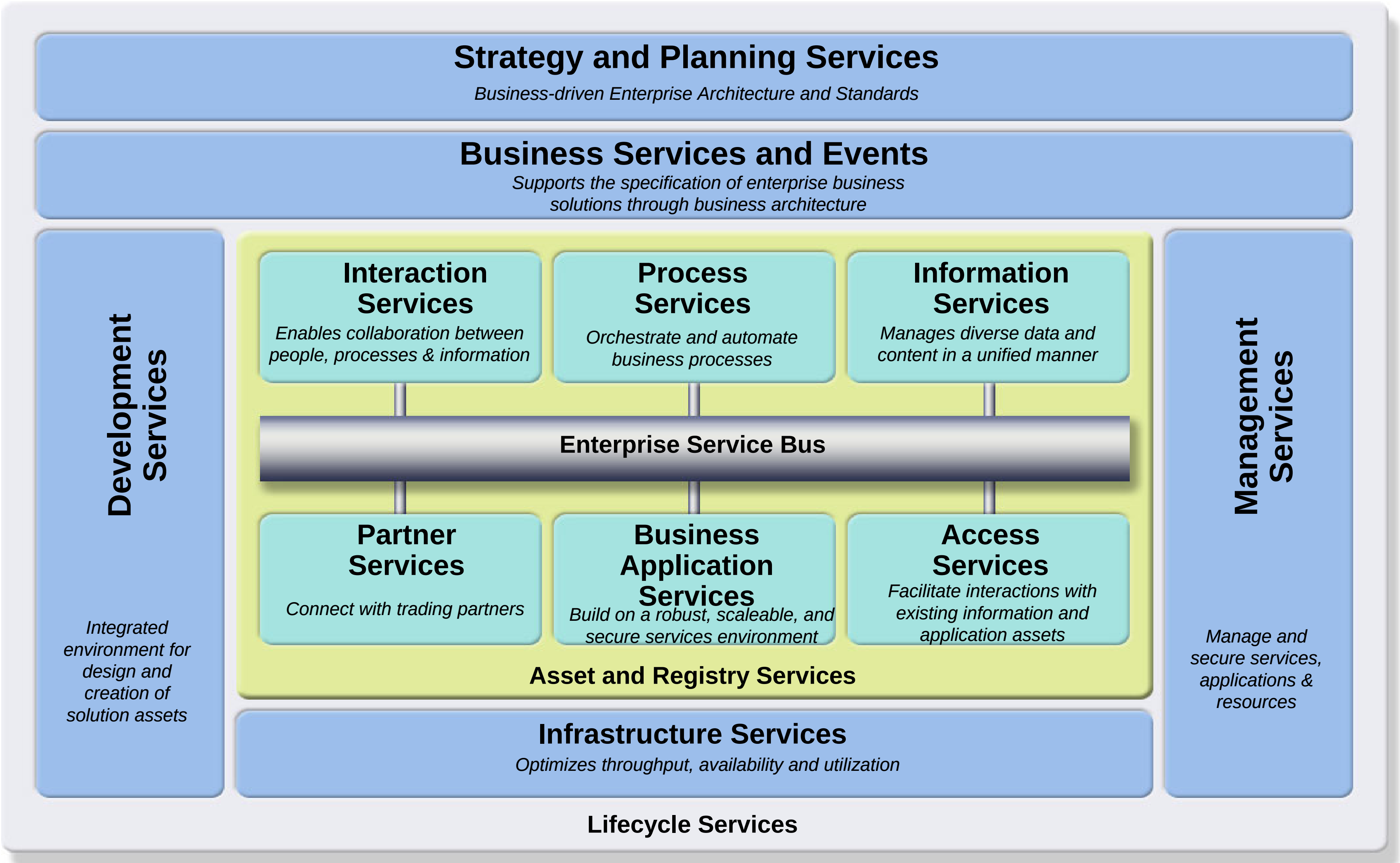
Deployment
Packaging and Provisioning

Iterative SOA Solution Design Process

As SOMA is applied during an engagement, we incrementally populate an architectural overview (“dashboard view”) of the SOA Solution



IBM SOA Foundation Reference Model



Strategy and Planning Services

Business-driven Enterprise Architecture and Standards

Business Services and Events

Supports the specification of enterprise business solutions through business architecture

Development Services

Integrated environment for design and creation of solution assets

Interaction Services

Enables collaboration between people, processes & information

Process Services

Orchestrate and automate business processes

Information Services

Manages diverse data and content in a unified manner

Enterprise Service Bus

Partner Services

Connect with trading partners

Business Application Services

Build on a robust, scaleable, and secure services environment

Access Services

Facilitate interactions with existing information and application assets

Asset and Registry Services

Management Services

Manage and secure services, applications & resources

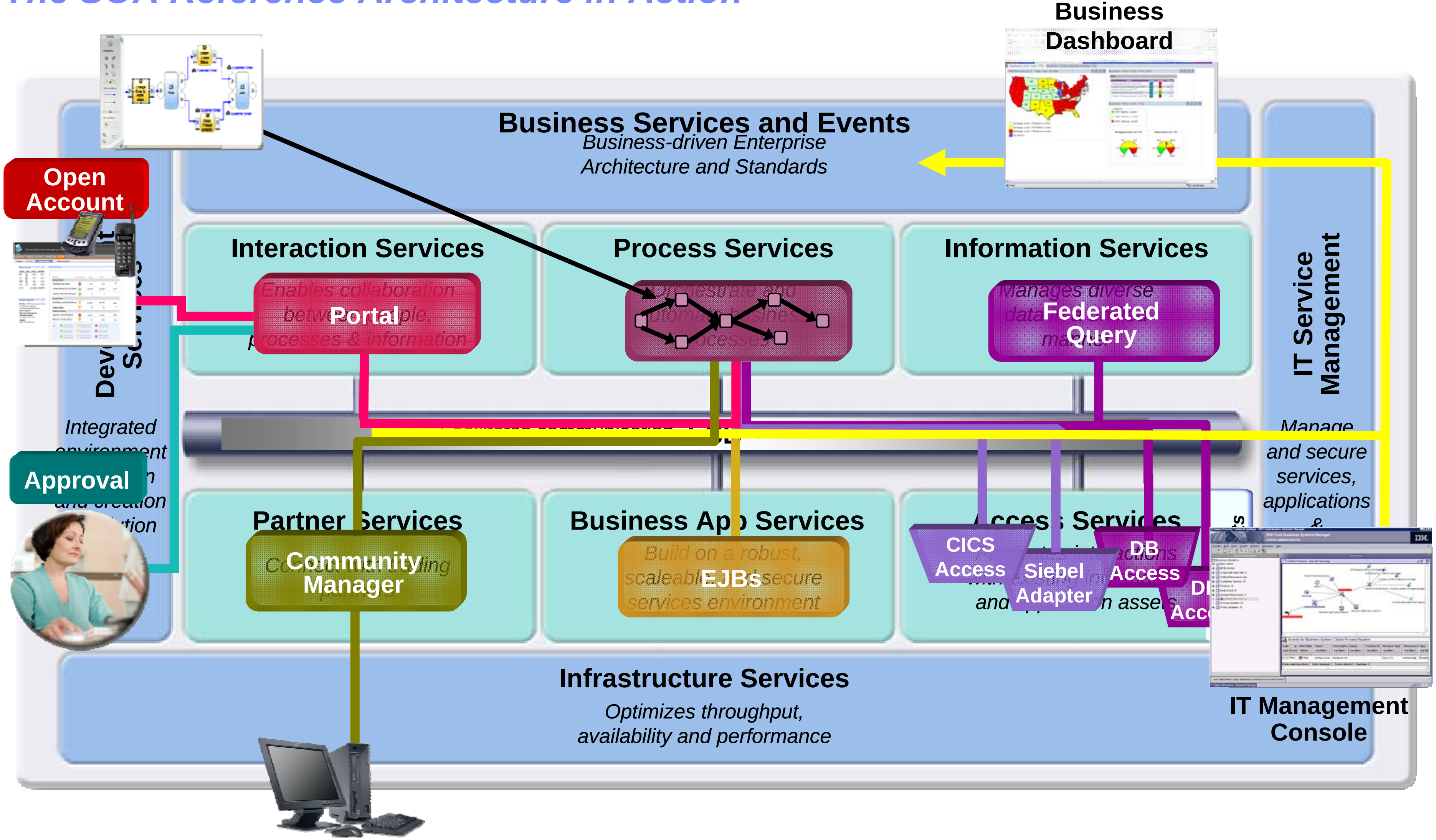
Infrastructure Services

Optimizes throughput, availability and utilization

Lifecycle Services

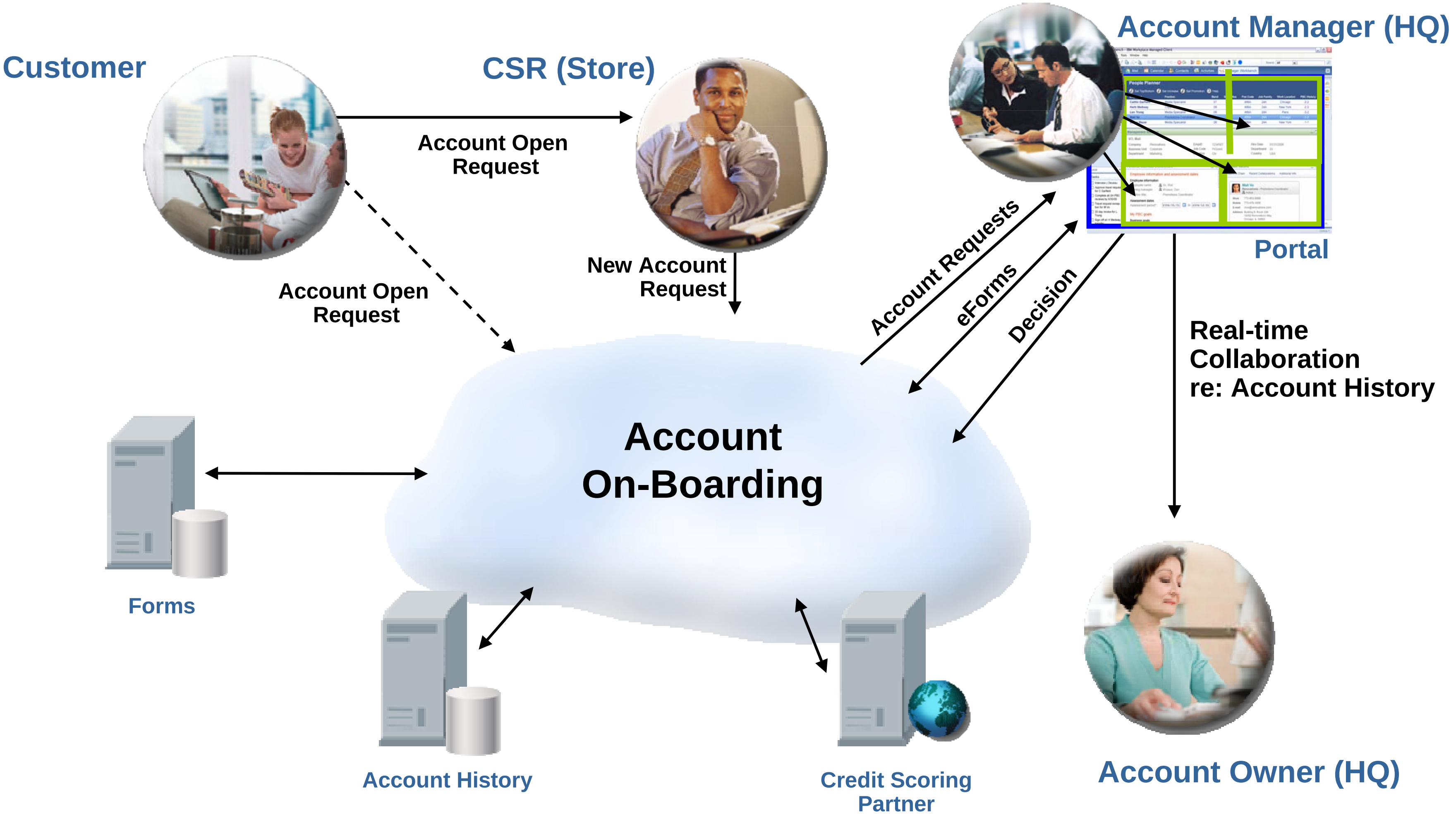
Separation of Concerns: Example “Open Account” Process

The SOA Reference Architecture in Action

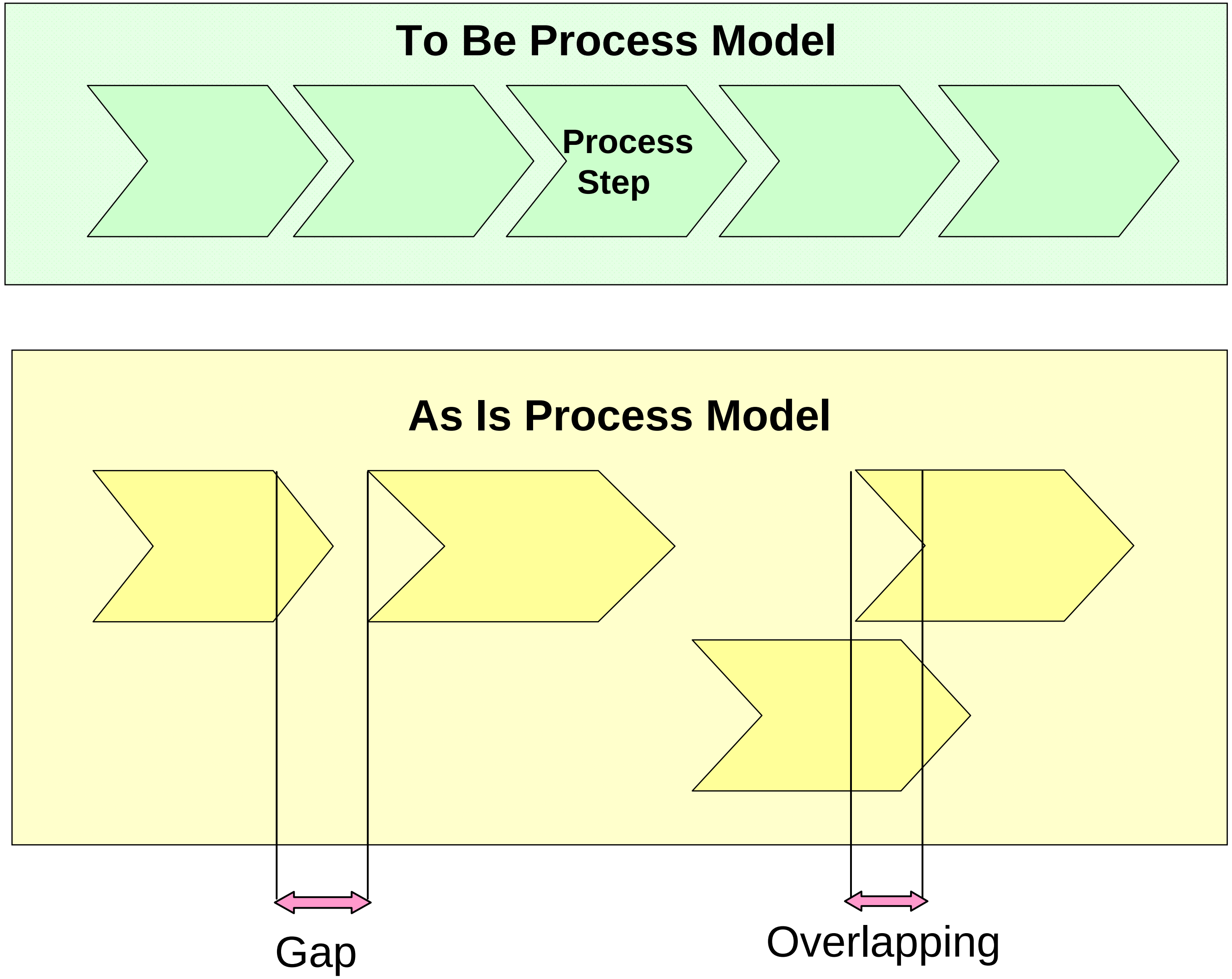


Identification and Specification of Services (SOMA)

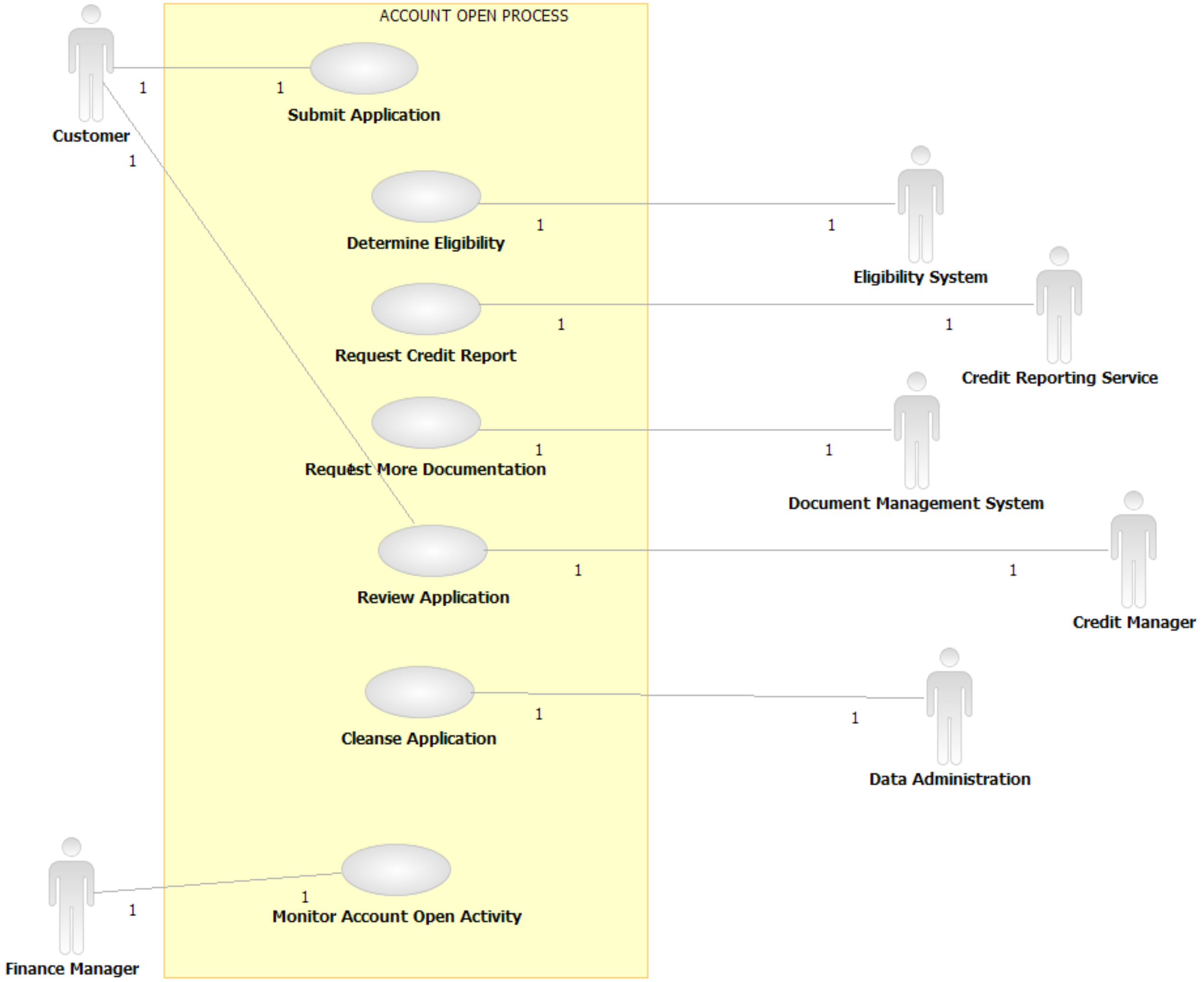
Example: Business Context Diagram for Business Process “Open Account” (Solution Viewpoint)



Business Process Reality and Plans – Streamline Business Process – Derive Requirements



Example: Use Case for JKE's "Open Account"

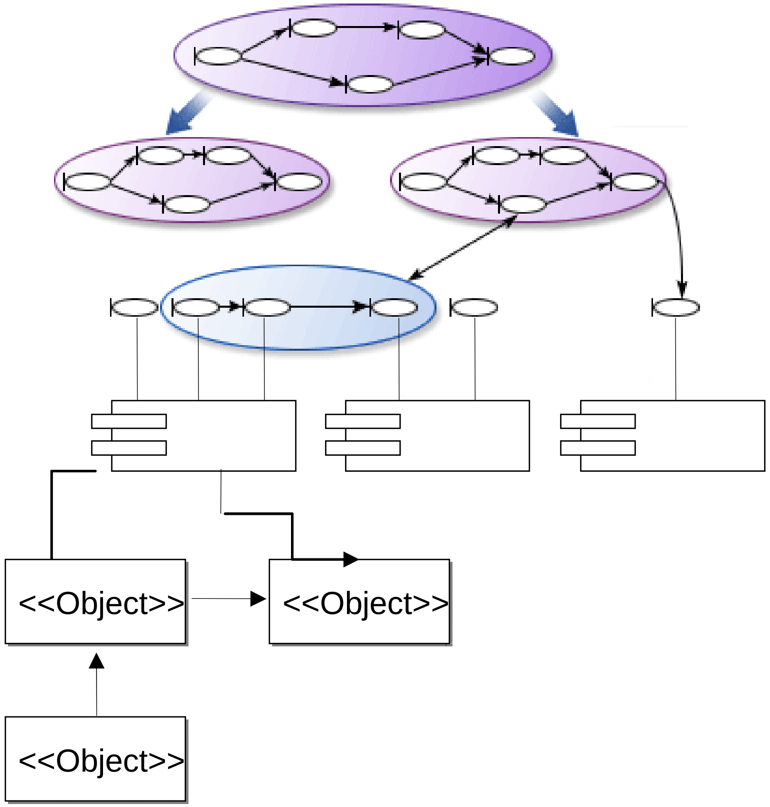


SOA Modeling Constructs

Business Processes
(Flows)

Services
Atomic and Composite

Service Components

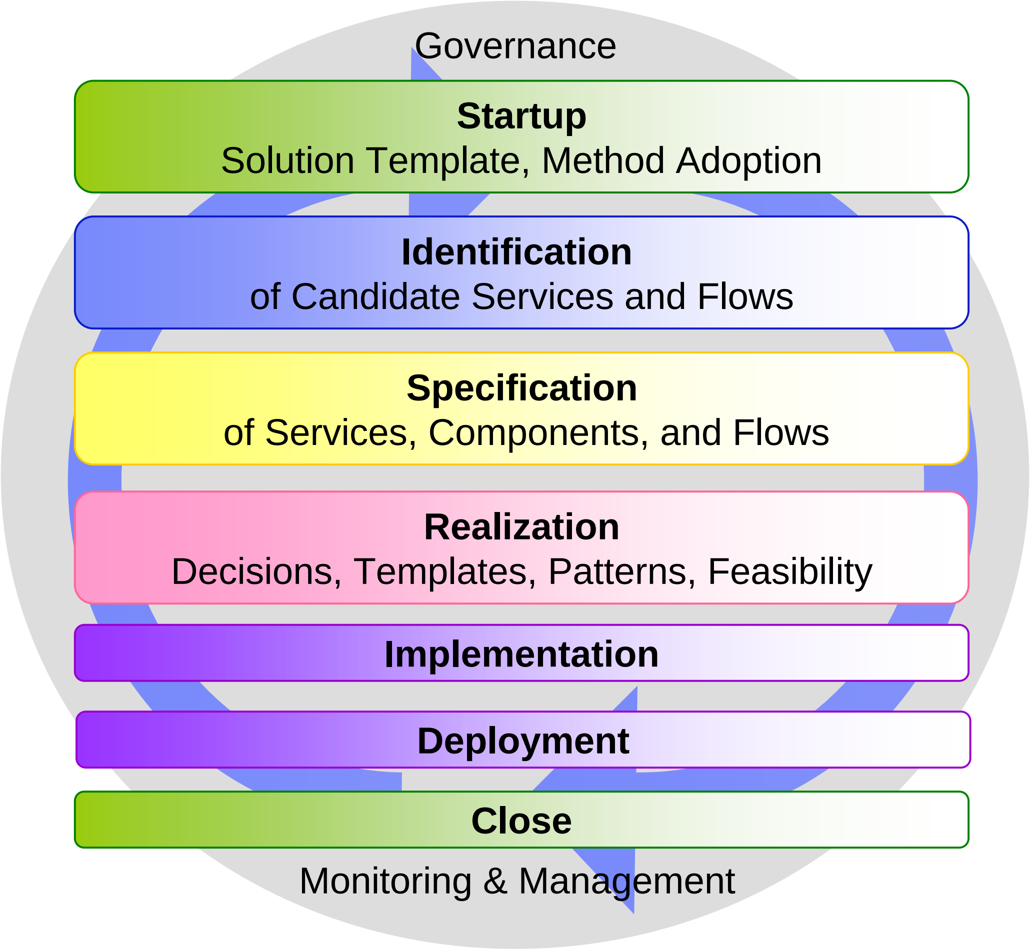


SOMA was created to specifically address modeling of all three constructs.

Introducing SOMA (Service Oriented Modeling and Architecture)

- **SOMA is a business-driven modeling and design method**
- **SOMA provides in-depth guidance on how to move from the business models to the IT models required by SOA**
- **SOMA adds new service-oriented aspects and techniques in intelligent ways to enable an SOA with services directly traceable to business goals and requirements**

At the heart of SOMA is identification, specification, realization and implementation of services, components and flows



Governance

Startup
Solution Template, Method Adoption

Identification
of Candidate Services and Flows

Specification
of Services, Components, and Flows

Realization
Decisions, Templates, Patterns, Feasibility

Implementation

Deployment

Close

Monitoring & Management

- **Design is separated in Identification and Specification**
- **Realization are mainly decisions on how to implement, buy, or use existing assets**
- **Implementation and Deployment as “classical” Software Engineering**

SOMA defines What we do and How we do it

What we do ?

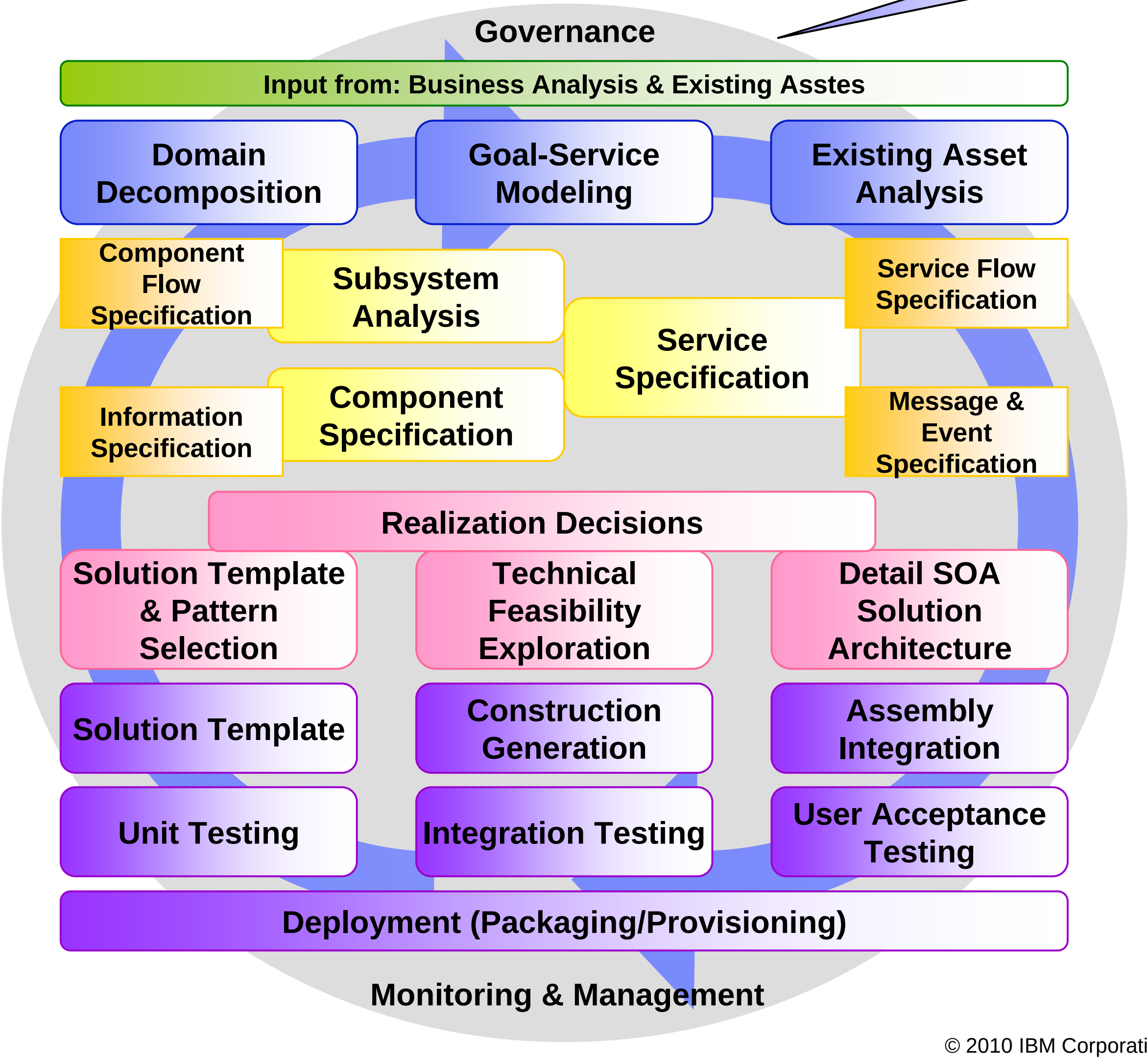
Identification
of candidate services and flows, leverageable existing assets

Specification
of services to be exposed, flows, and components (for realization of functionality)

Realization
captures realization decisions, selects solution templates, details SOA Solution Ref. Arch.

Implementation
incl. construction/ generation, assembly, testing, deployment, monitoring and management

How we do it?



Governance

Input from: Business Analysis & Existing Asstes

Domain Decomposition

Goal-Service Modeling

Existing Asset Analysis

Component Flow Specification

Subsystem Analysis

Service Flow Specification

Service Specification

Information Specification

Component Specification

Message & Event Specification

Realization Decisions

Solution Template & Pattern Selection

Technical Feasibility Exploration

Detail SOA Solution Architecture

Solution Template

Construction Generation

Assembly Integration

Unit Testing

Integration Testing

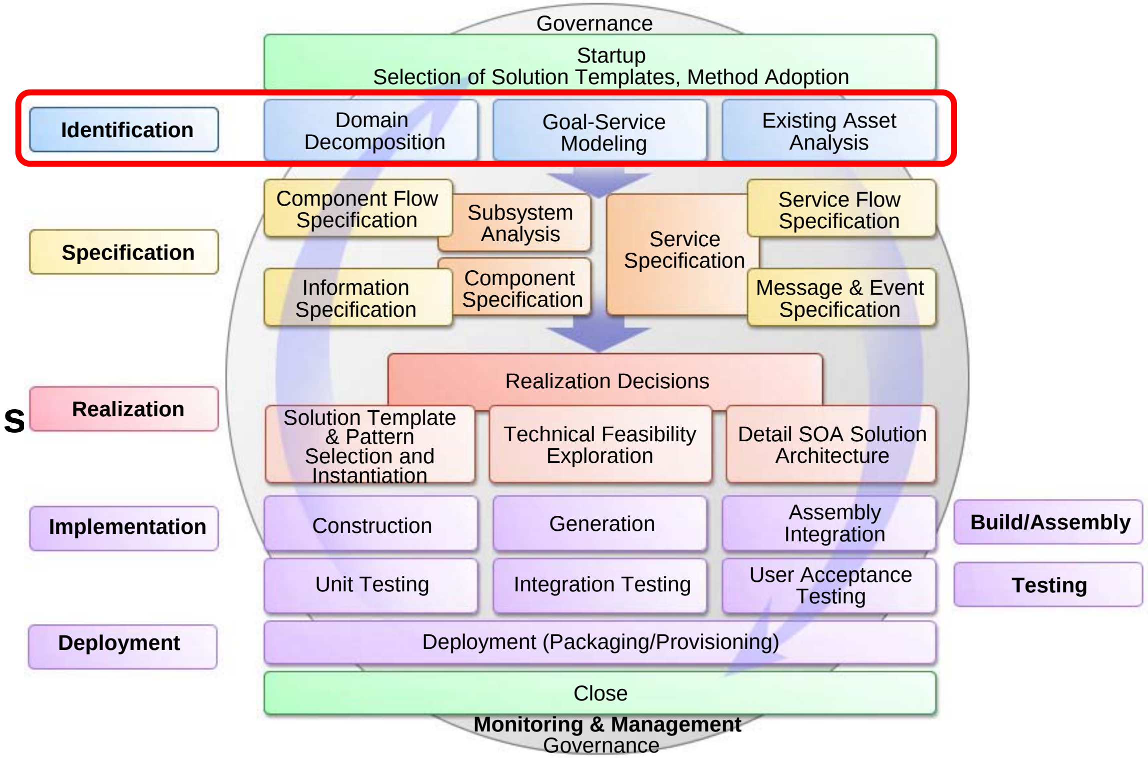
User Acceptance Testing

Deployment (Packaging/Provisioning)

Monitoring & Management

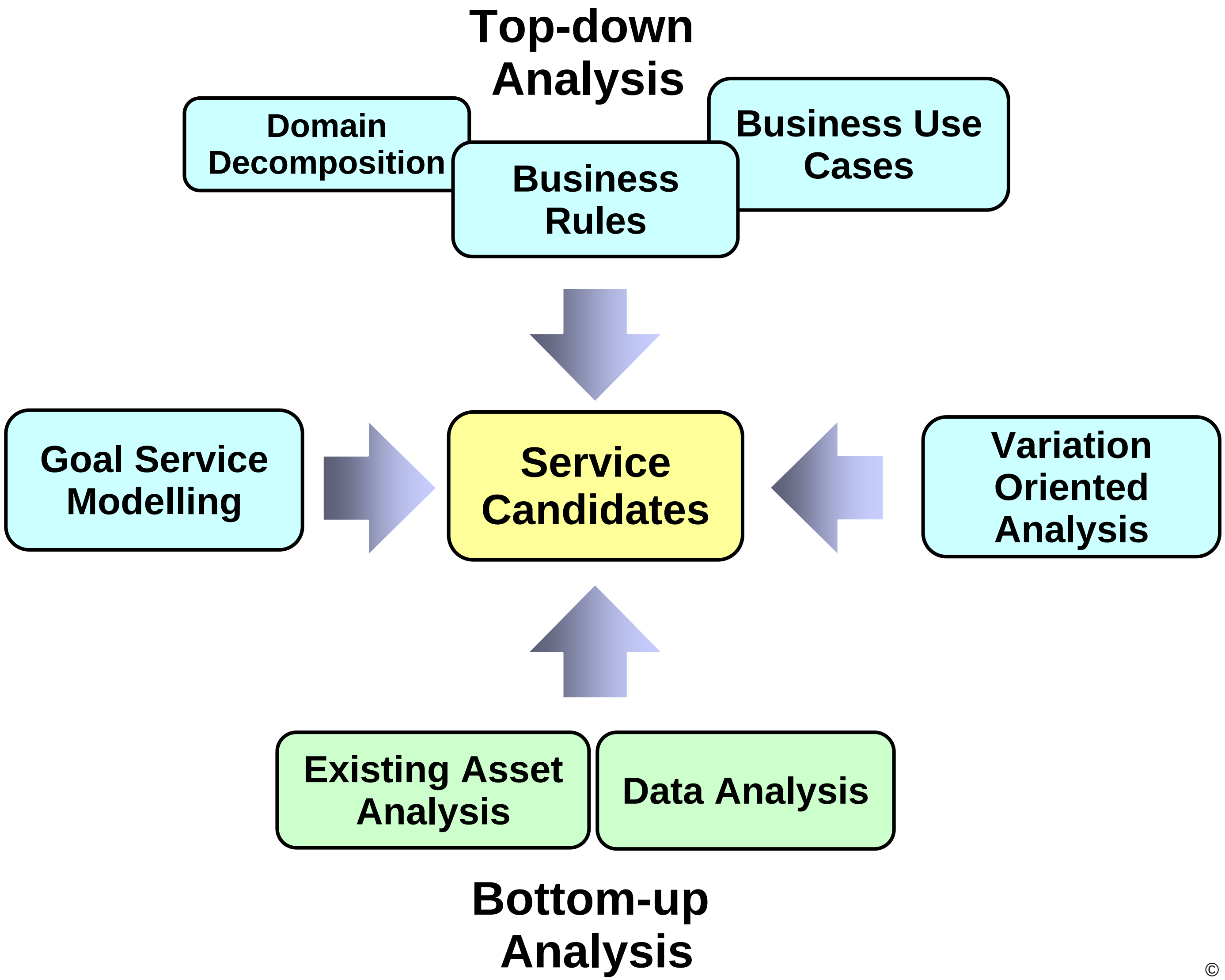
Identifies Services

- **Domain Decomposition** **(Top-down Analysis)**
  - **Process Decomposition**
  - **Functional Area Analysis**
  - **Information Analysis, Modeling, and Planning**
  - **Rule and Policy Analysis**
  - **Variation-Oriented Analysis**
- **Existing Asset Analysis** **(Bottom-up Analysis)**
- **Goal-Service Modeling**
- **Additionally, Service Refactoring and Rationalization**
  - **Service Litmus Tests**
  - **Exposure Decisions, including Exposure Scope**

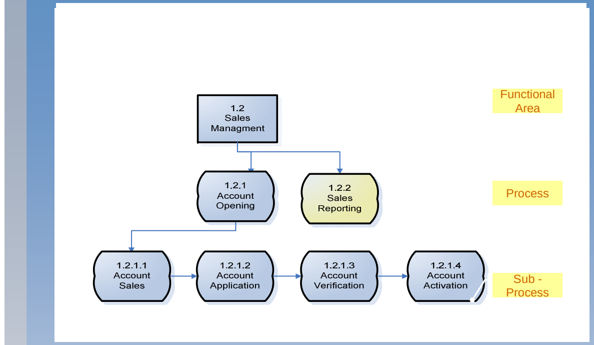


Id Services, Components, and Flows

Service Identification Through 3 main Complimentary Techniques

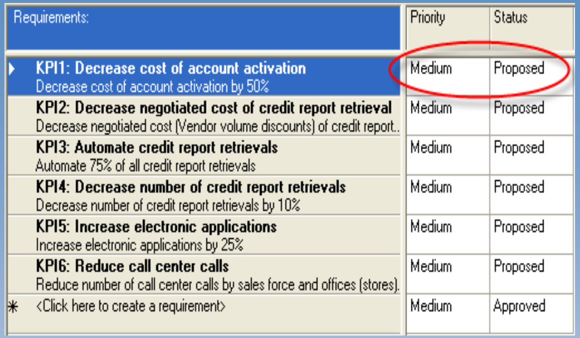


Service Design via SOMA – Service Identification



Domain Decomposition

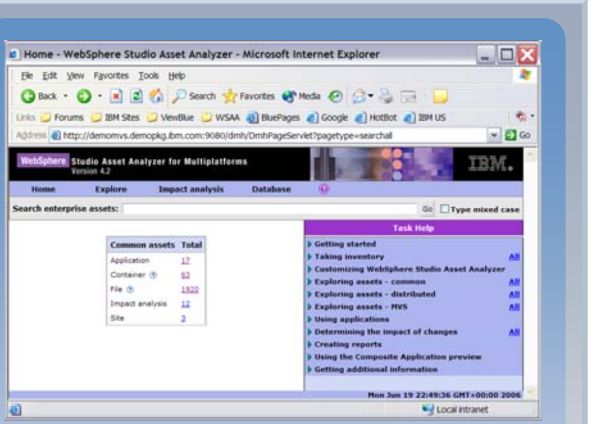
- Techniques:
  - Process Modeling Tools
  - Design of KPIs/Metrics
- Services Identified
  - Open Account
  - Account Activation
  - Account Verification



Requirements:	Priority	Status
KPI1: Decrease cost of account activation Decrease cost of account activation by 50%	Medium	Proposed
KPI2: Decrease negotiated cost of credit report retrieval Decrease negotiated cost (Vendor volume discounts) of credit report..	Medium	Proposed
KPI3: Automate credit report retrievals Automate 75% of all credit report retrievals	Medium	Proposed
KPI4: Decrease number of credit report retrievals Decrease number of credit report retrievals by 10%	Medium	Proposed
KPI5: Increase electronic applications Increase electronic applications by 25%	Medium	Proposed
KPI6: Reduce call center calls Reduce number of call center calls by sales force and offices (stores).	Medium	Proposed
<Click here to create a requirement>	Medium	Approved

Goal Service Modeling

- Techniques
  - Requirements Planning Tools
  - Design of KPIs/Metrics
- Services Identified
  - Determine Applicant Eligibility
  - Address Verification


Home - WebSphere Studio Asset Analyzer - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Links Forums IBM Sites ViewBlue WSAA BluePages Google HotBot IBM US
Address Go
WebSphere Studio Asset Analyzer for Multiplatforms
Version 4.2
IBM.
Home Explore Impact analysis Database
Search enterprise assets: Go □ Type mixed case
Task Help
Local intranet

http://demomvs.demopkg.ibm.com:9080/dmh/DmhPageServlet?pagetype=searchall

Common assets	Total
Application	17
Container	63
File	1920
Impact analysis	12
Site	3

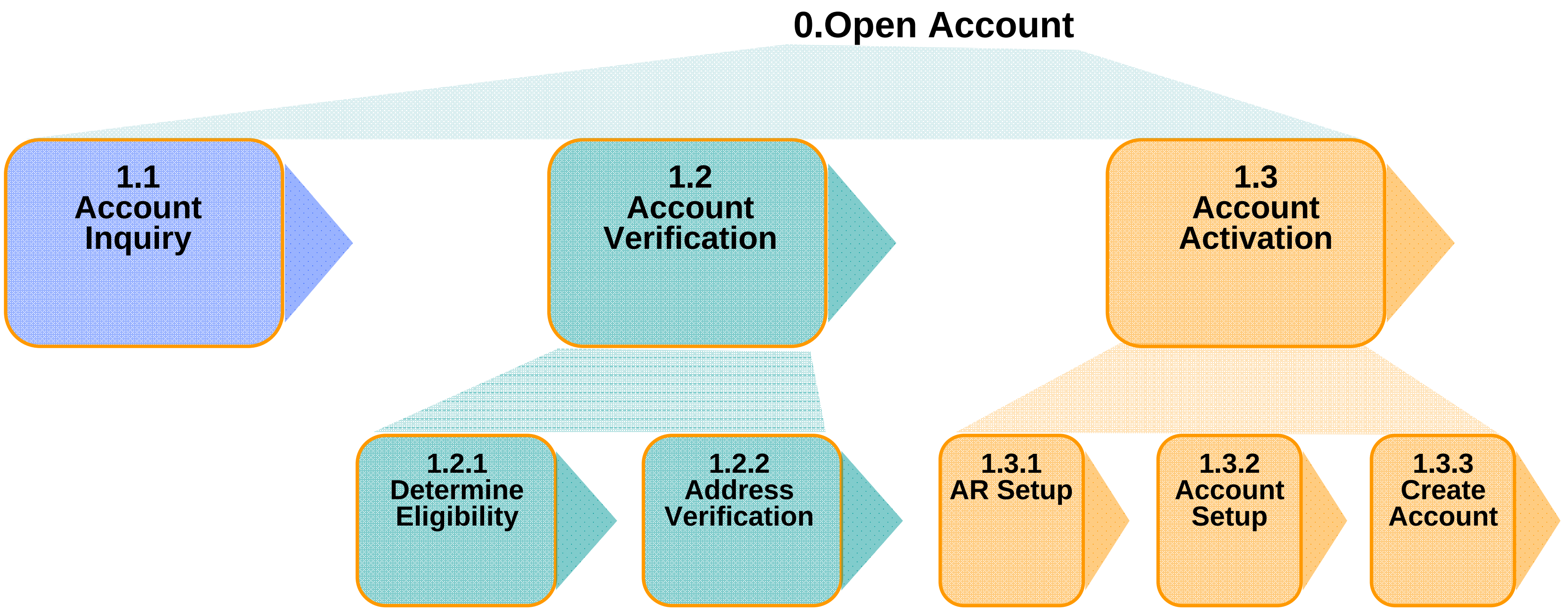
- Getting started
- Taking inventory All
- Customizing WebSphere Studio Asset Analyzer
- Exploring assets - common All
- Exploring assets - distributed All
- Exploring assets - MVS All
- Using applications
- Determining the impact of changes All
- Creating reports
- Using the Composite Application preview
- Getting additional information

Mon Jun 19 22:49:36 GMT+00:00 2006

Existing Asset Analysis

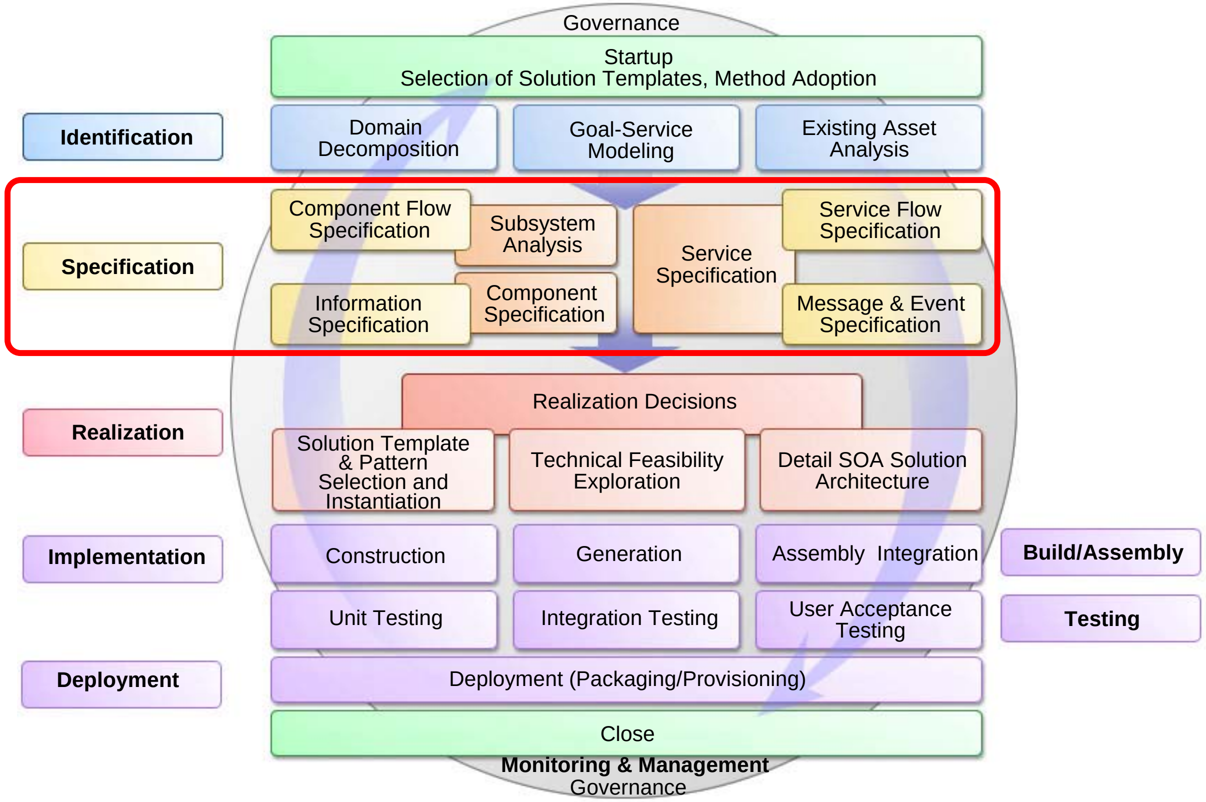
- Techniques
  - Asset Analysis Tools
  - Interviews/Documentation
- Services Identified
  - Account Inquiry (CICS 2.2)
  - AR Setup (CICS 2.2)
  - Account Setup (CICS 3.1)
  - Create Account (SAP)

Example: Domain Decomposition – Business Process Modeling for JKE's "Open Account"

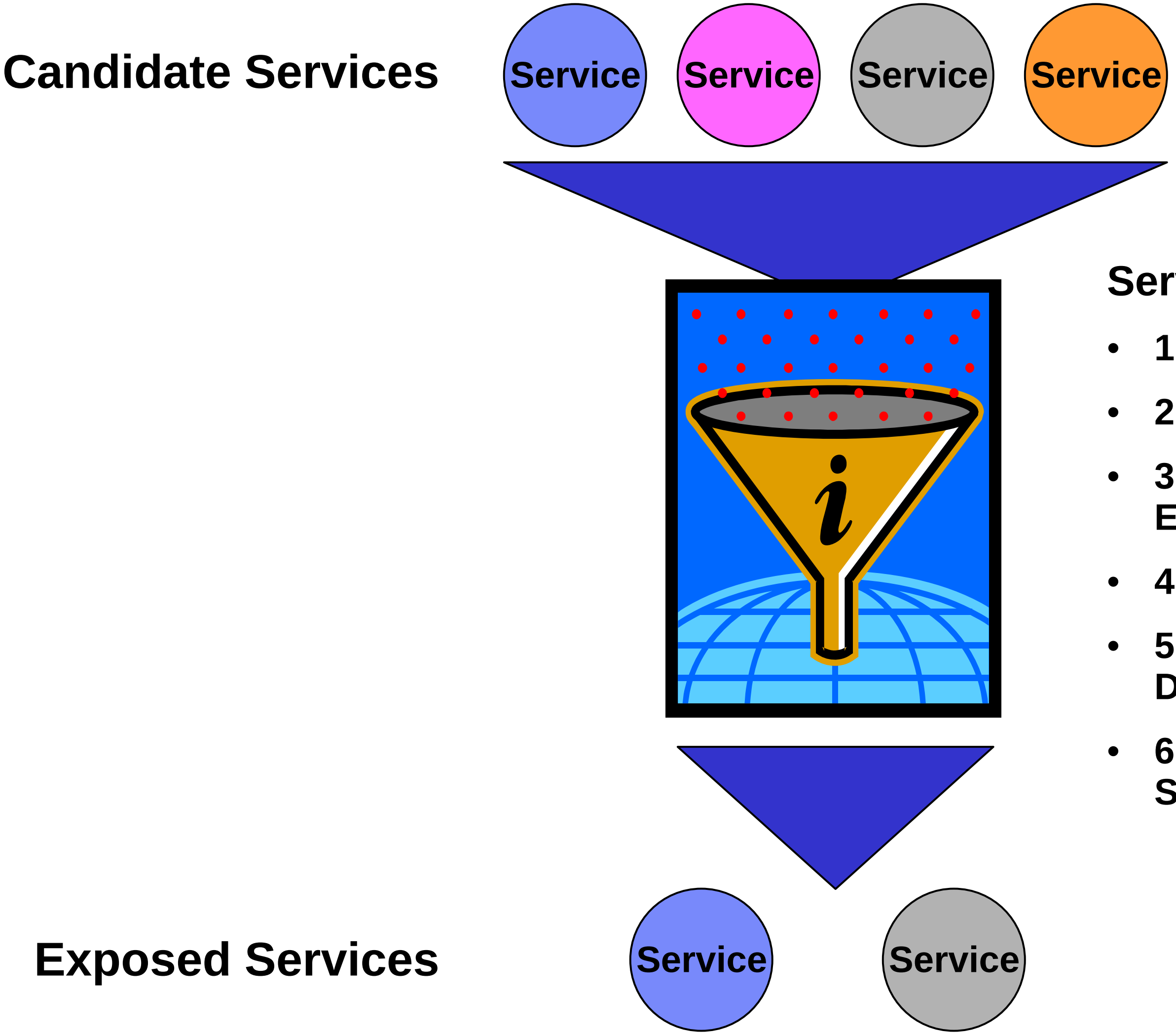


SOMA Specification uses comprehensive techniques to specify Services, Flows, and Service Components that Realize Services

- **Information Specification**
  - **Data Model, Message Model, Business Glossary**
- **Existing Asset Analysis – Fine Grained**
  - **Determine the technical viability of existing applications and approaches to realize services**
- **Service Specification**
  - **Elaborates the Service Model, for example, service dependencies, service composition and flow, rules and policies, event specification, service operation, service message specification, QoS requirements, design decisions, and so on**
- **Subsystem Analysis**
  - **Partitions subsystems into service components that will be responsible for service realization**
- **Component Specification**
  - **Details component modeling, flow, information architecture, messages**



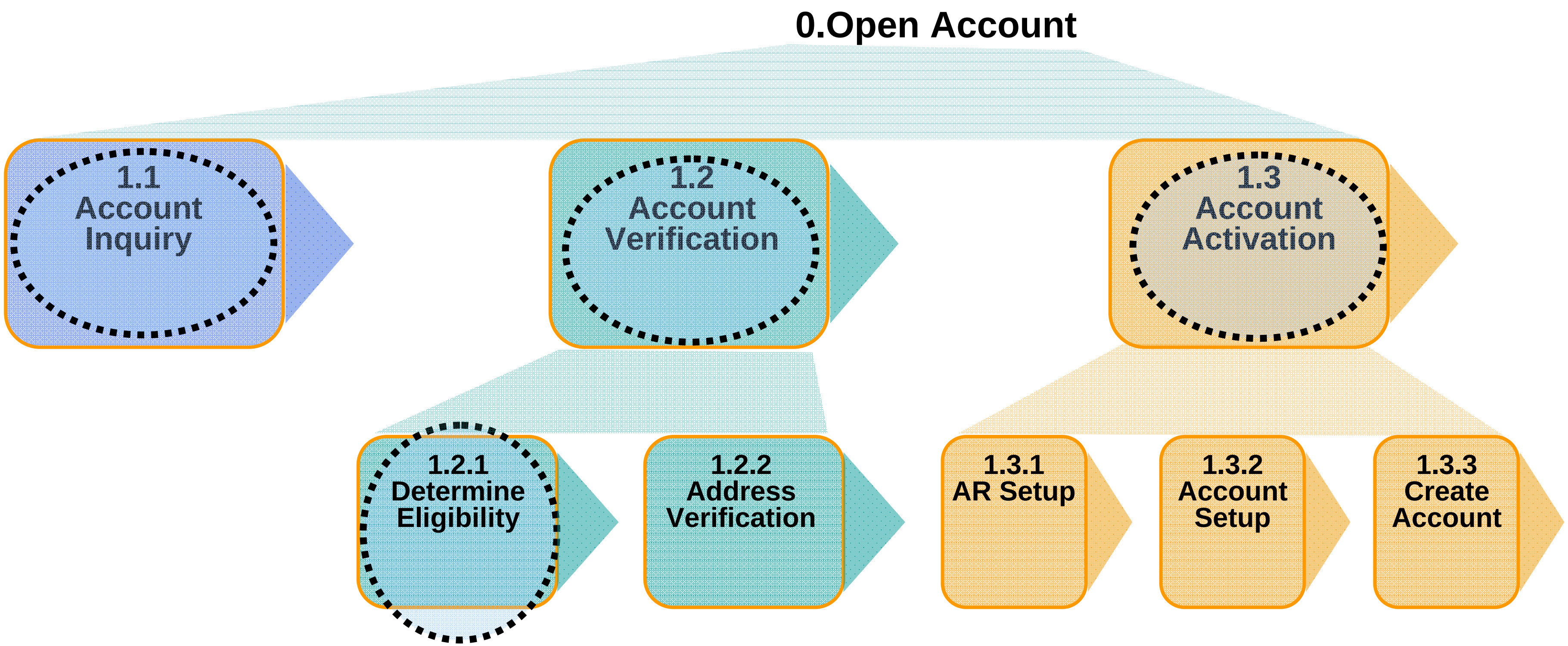
Service Litmus Tests Are Gating Criteria Used to Determine If a Candidate Service Should Be Exposed



Service Litmus Tests

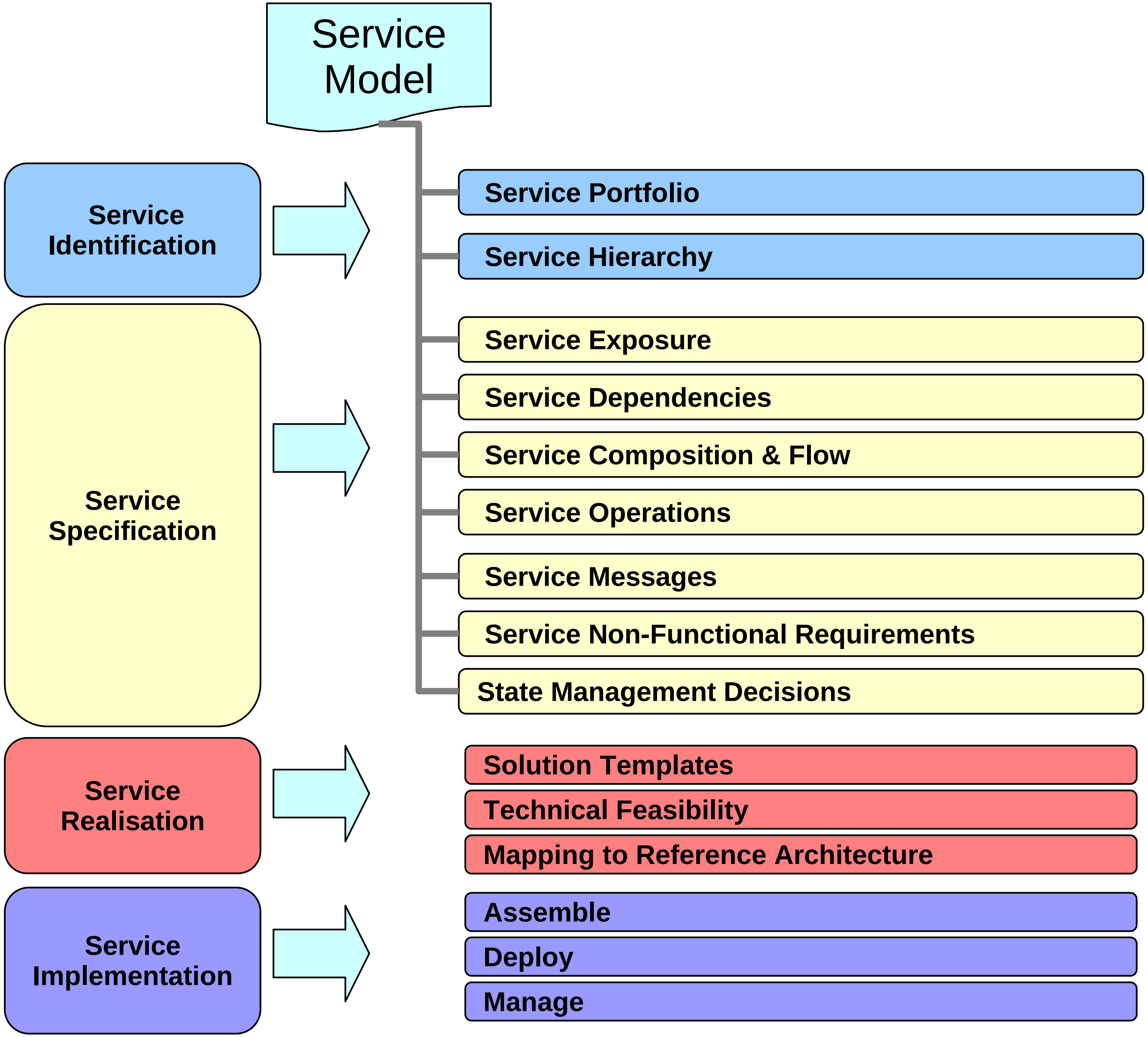
- 1. Business Alignment
- 2. Composability
- 3. Consolidation (Redundancy Elimination)
- 4. Technical Feasibility
- 5. [Externalized Service Description]
- 6. Project Defined/Customer Specific SLTs

Example: JK Enterprises Service Exposure Decisions



Legend

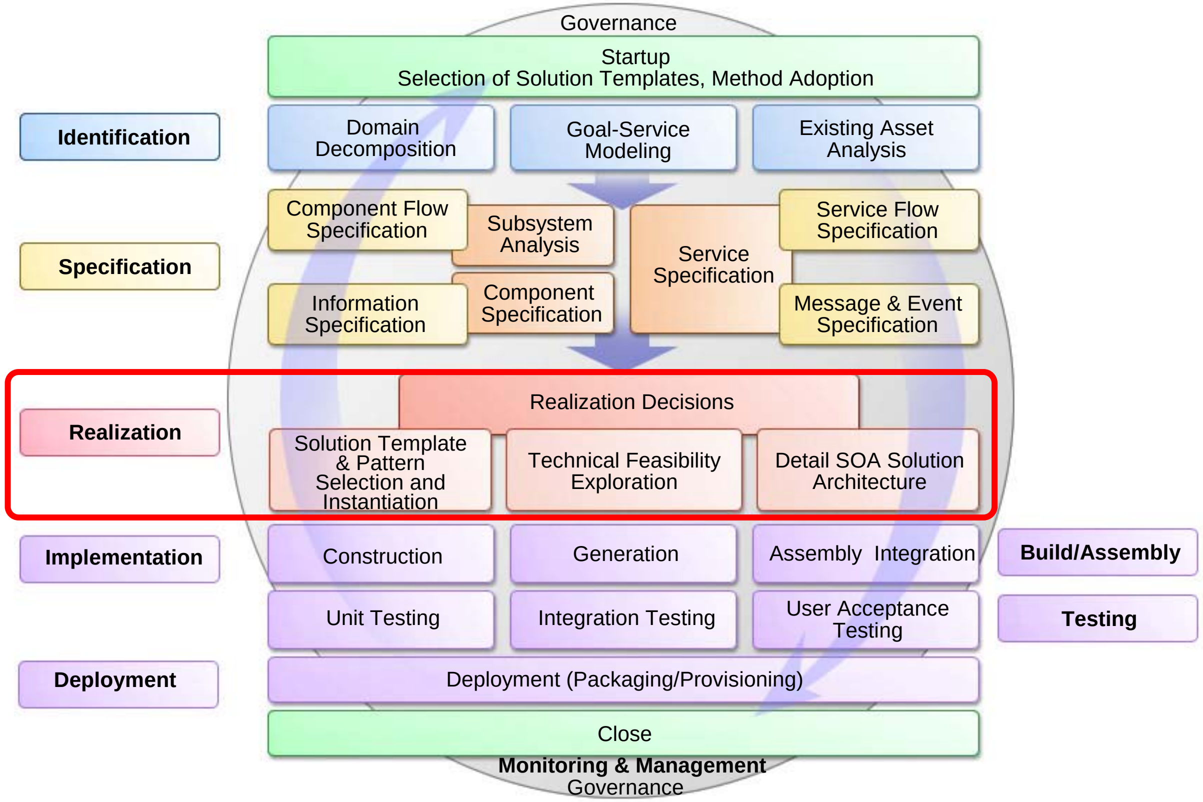
◌ = Service to be exposed



Service Model
Service Identification
Service Specification
Service Realisation
Service Implementation
Service Portfolio
Service Hierarchy
Service Exposure
Service Dependencies
Service Composition & Flow
Service Operations
Service Messages
Service Non-Functional Requirements
State Management Decisions
Solution Templates
Technical Feasibility
Mapping to Reference Architecture
Assemble
Deploy
Manage

SOMA Realization (Includes SOA Solution Stack Instantiation)

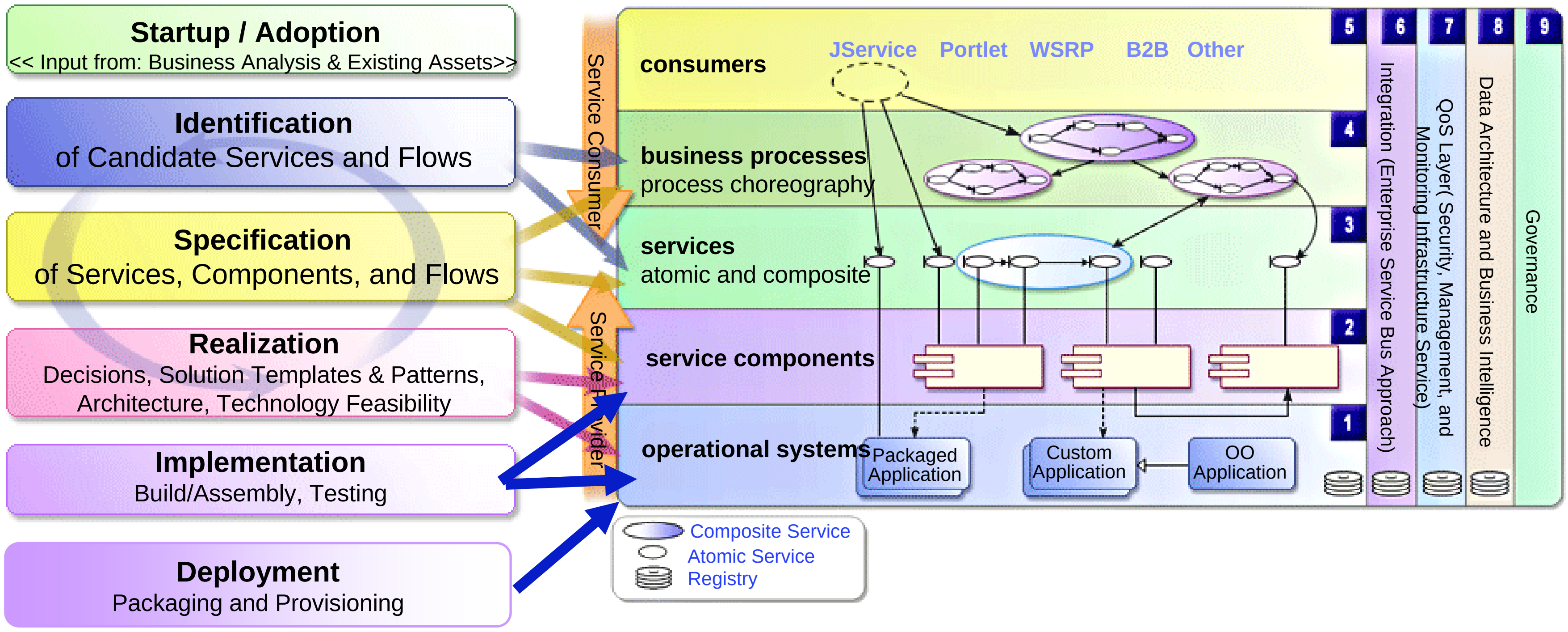
- Select and instantiate Solution Templates and Patterns
- Technical Feasibility Exploration
  - Examine approaches to handle client requirements
  - Examine legacy application specific considerations
- Detail SOA Solution Stack
- Realization Decisions
  - Consider alternatives
  - Select the alternative
  - Provide justification



SOA Layered View

SOMA for Service / Business Process Development

SOMA is an end-to-end development method aimed at enabling target business processes through the definition of business-aligned services that form the service-oriented architecture foundation



Startup / Adoption
<< Input from: Business Analysis & Existing Assets>>

Identification
of Candidate Services and Flows

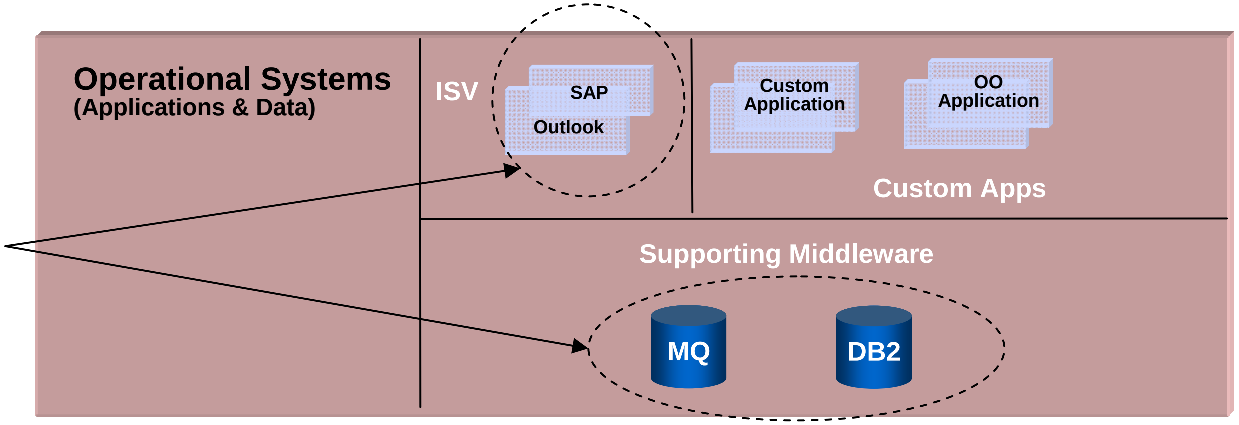
Specification
of Services, Components, and Flows

Realization
Decisions, Solution Templates & Patterns, Architecture, Technology Feasibility

Implementation
Build/Assembly, Testing

Deployment
Packaging and Provisioning

Layer 1: Operational Systems (Leverage Existing Investment)

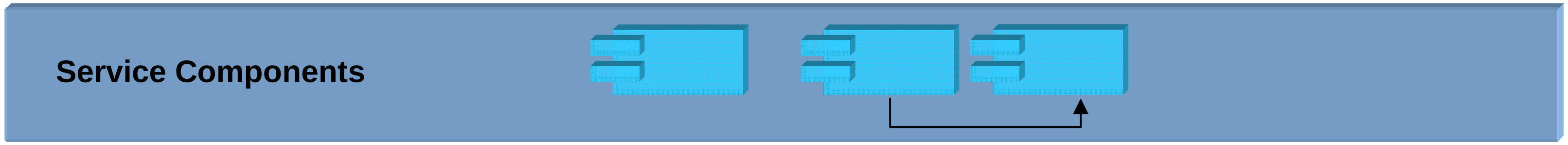


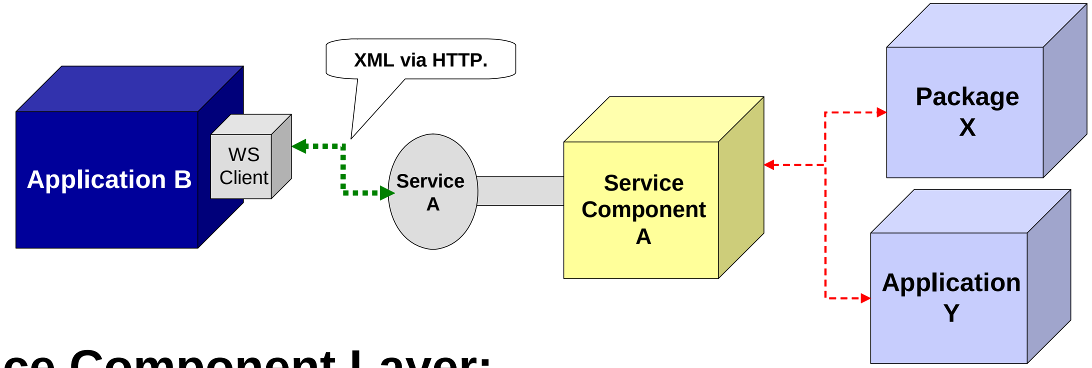
Examples for illustration: specifics are not in the scope of the reference model.

- Recognizes the value of existing IT investment
- Some SOA Related Activities:
  - Asset Inventory
  - Refactor existing applications to unlock business value

    May have a valuable asset hidden in side an application, e.g. a portfolio valuation algorithm buried inside a COBOL application.

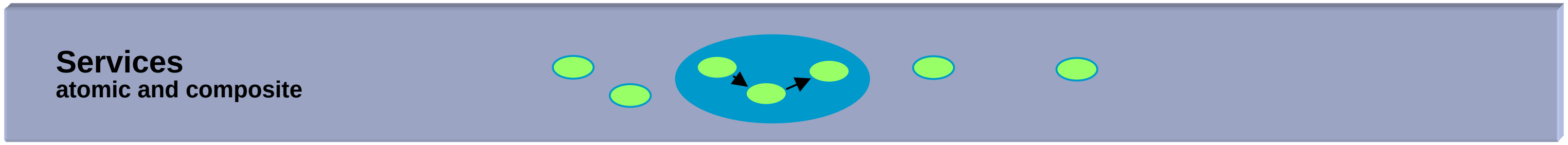
Layer 2: Service Components





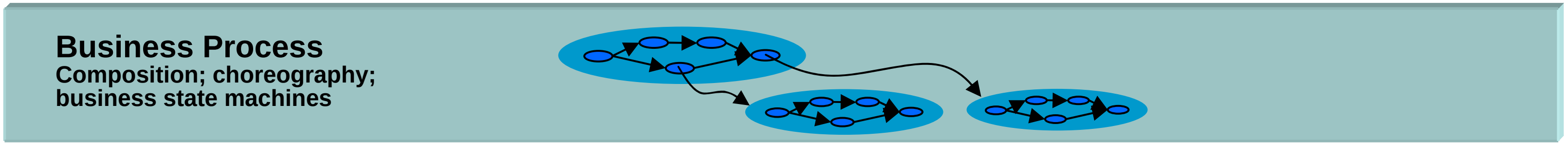
- **The Service Component Layer:**
  - **Enables IT flexibility by strengthening the decoupling in the system. Decoupling is achieved by hiding volatile implementation details from consumers.**
  - **Often employs container based technologies like EJBs**
- **Each Service Component:**
  - **Provides an enforcement point for service realization**
  - **Offers a facade behind which IT is free to do what they want/need to do**

Layer 3: Services (Decouple Business and IT)



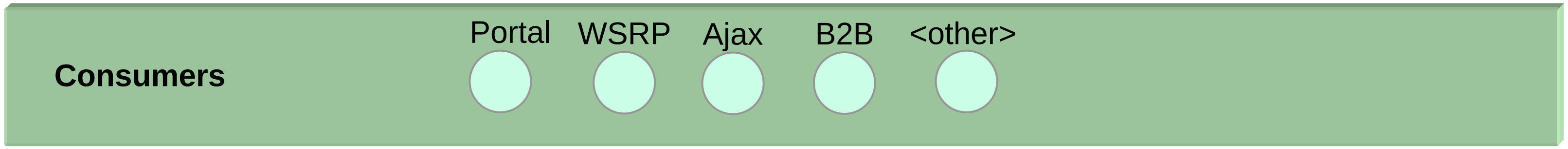
- **The Services Layer forms the basis for the decoupling of Business and IT.**
  - **Captures the functional contract (incl. QoS – Quality of Service) for each standalone** *business* **function or each task in a business process**
- **The assumption is that (within an SOA) IT responsibility is to realize/manage service implementations that faithfully conform to the set of services in the service model.**
- **This layer contains all the** *exposed services* **in the SOA**
- **Each service is a contract between the consumer(s) and the provider(s)**

Layer 4: Business Processes (Business process alignment of IT)



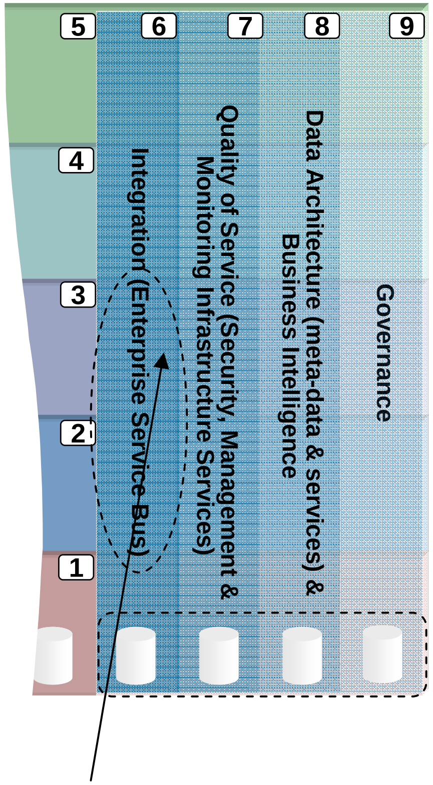
- This layer contains operational IT artifacts that implement business processes as a choreography of services
- The set of services that are choreographed/composed is restricted to those services that are defined in Layer 3
- While BPEL is often used in this layer it is not a requirement… e.g. a Java Bean could be used to choreograph a set of services.
  - The choice of technology depends on a set of realization decisions that must be made when establishing a physical Reference Model for a given SOA.
  - Those decisions are typically made based on requirements and the capabilities of the available alternatives.

Layer 5: The Consumer Layer (Channel independent access to business processes)



- **This layer exists to recognize that the technology chosen to expose Business Processes/Services must permit access from a wide set of interaction channels.**
- **When establishing a Operational Model for a given situation, it is important to populate this layer with the set of channels types that are required in a solution.**
- **Each channel type is typically accompanied limitation/capabilities that will shape the way the Operational Model supports communication with Business Processes and Services.**

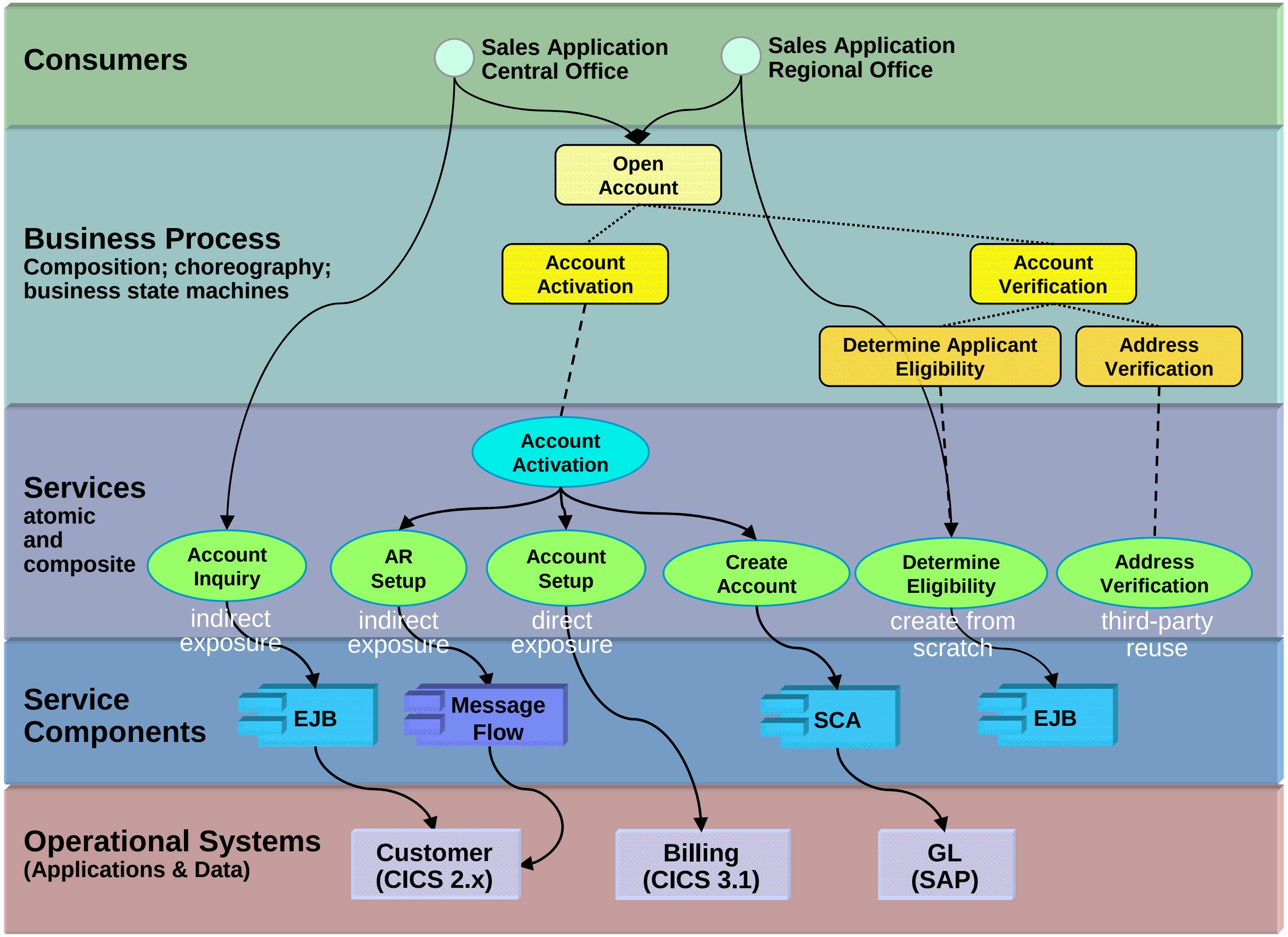
Cross-cutting concerns/capabilities


5
6
7
8
9
4
3
2
1
Integration (Enterprise Service Bus)
Quality of Service (Security, Management & Monitoring Infrastructure Services)
Data Architecture (meta-data & services) & Business Intelligence
Governance

for illustration: this is not saying that SOA requires an ESB.

- Several concerns are not restricted to a single layer in the Reference Model, these concerns are captured in 'Layers' 6-9
- These are not really layers but treating them as such gives us the ability focus discussions/decisions, for example "What is found where Governance intersects Services? i.e. what are the Governance concerns specific to Services?"
- Clearly there is interaction among these 'layers' also. For example, it is likely that most data architectures will be subject to governance

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



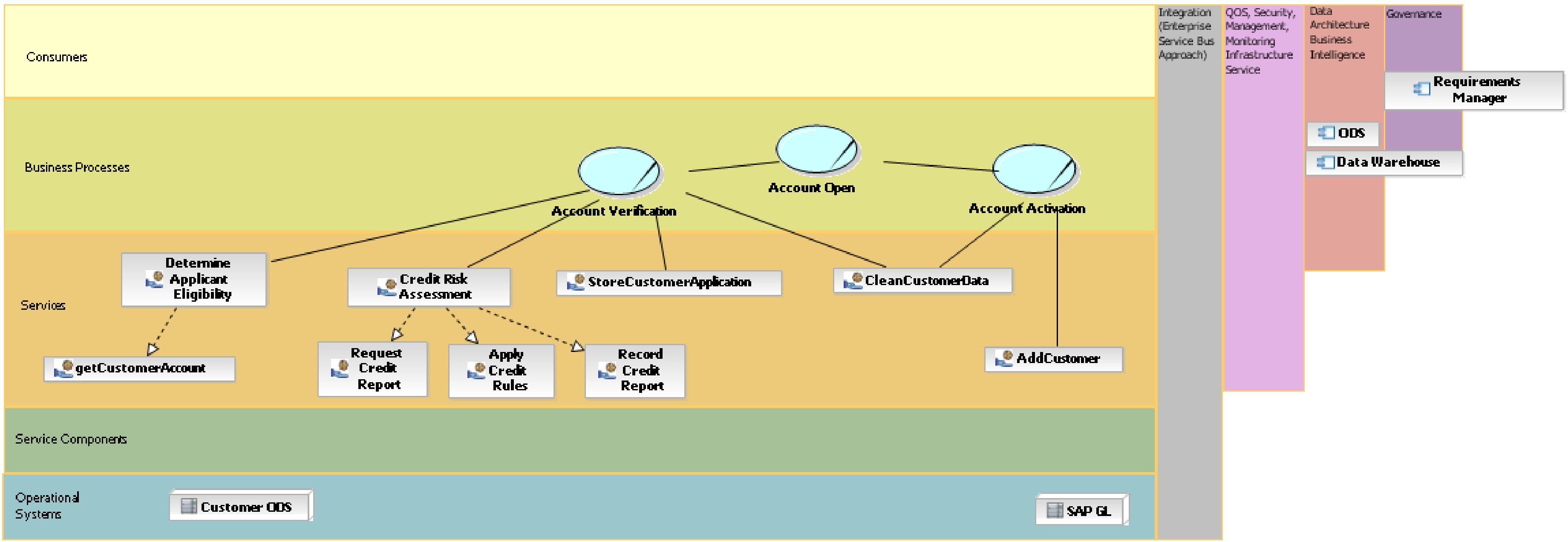
Exercise (Home Work)

Exercise Layered View

- **Usually a diagram (or set) which is used as a basis for discussion and explanation.**
- **Assume you will create many iterations of this document.**
- **Should contain processes, services, components, and operational systems**

Exercise – SOA Solution Layer Perspective – Add Missing Components

JKE SOA Solution Layer Perspective - AS IS



- Among the missing artifacts from this diagram, the Service Components (service realization)
- Also missing are To-Be supporting operational systems