



Veer Narmad South Gujarat University

M.Sc. (IT) Programme

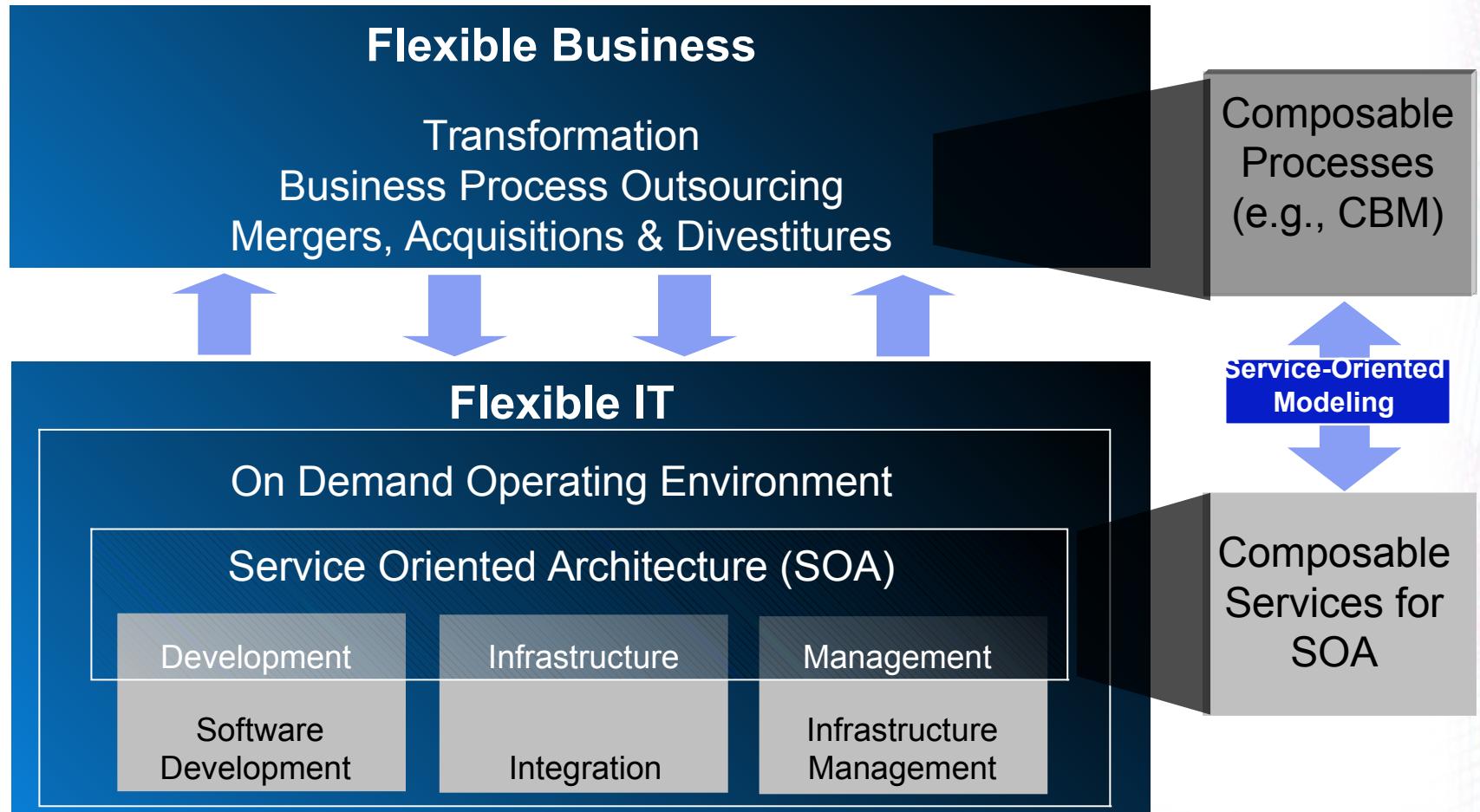
Service Oriented Architecture

* Basics of SOA Architecture *

Presented By : Kamlendu Kumar Pandey
Asst. Prof., M.Sc(IT) Programme
J.P. Dawar Institute of Info Science and Tech.
Surat



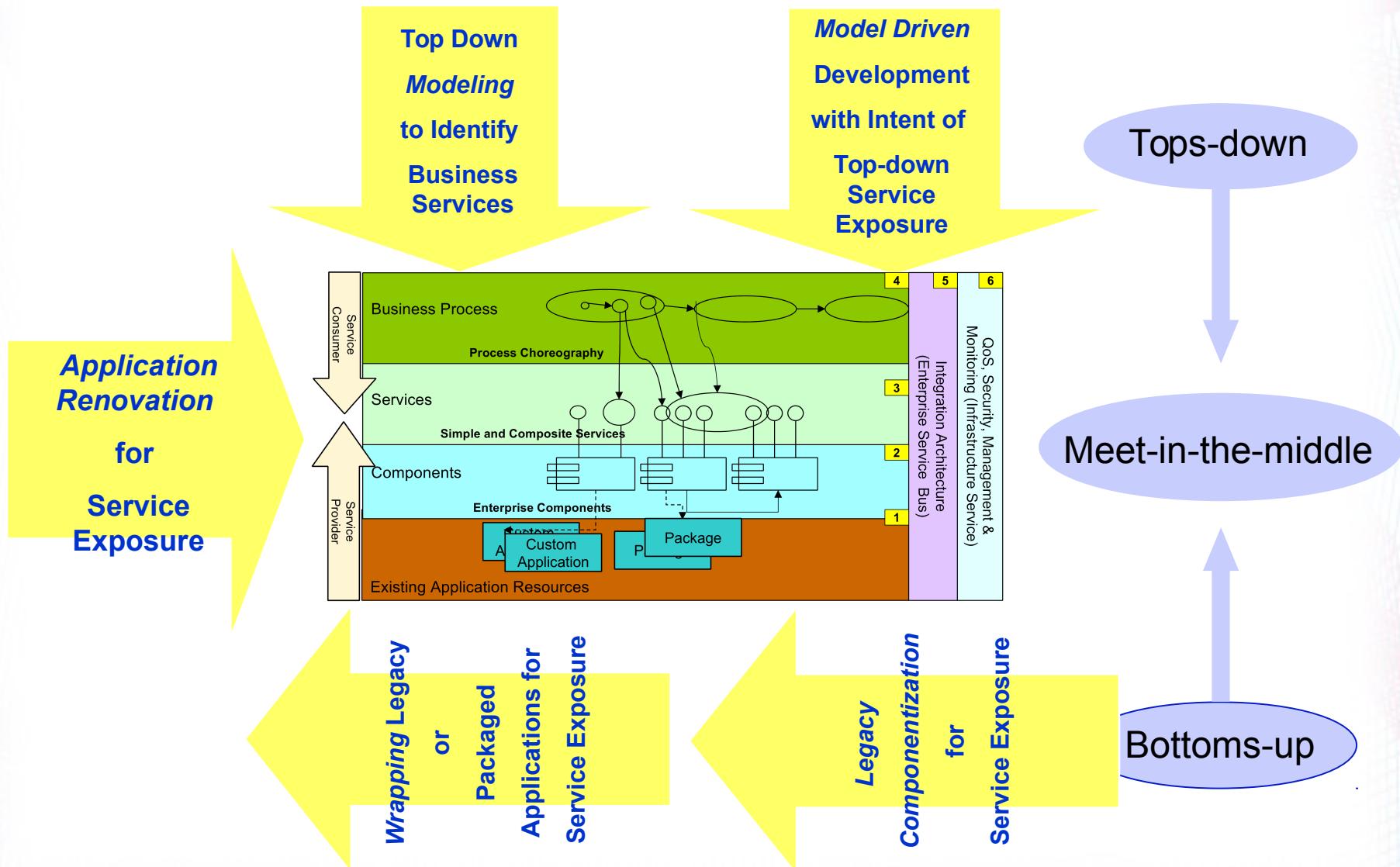
Greater Flexibility Required From Business Models And The Supporting IT Architecture



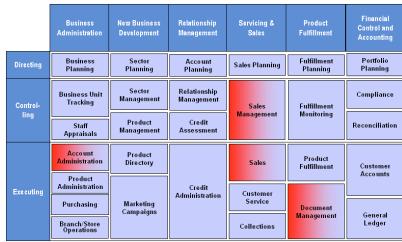
The Goals of SOA Transformation

- Development by assembly
- Rapid response to new opportunities and threats
- Reuse of components
- Application Portfolio consolidation
- Process automation across business units and business partners
- On-demand, real-time business processes
- Support multiple access channels

Approaches to Service Oriented Architecture



Overview of Service Analysis and Design



- **Business Analysis and Design**
Capture the Business Architecture

Identification
of Candidate Services and Flows

- **Service Analysis**

Specification
of Service Components and Flows

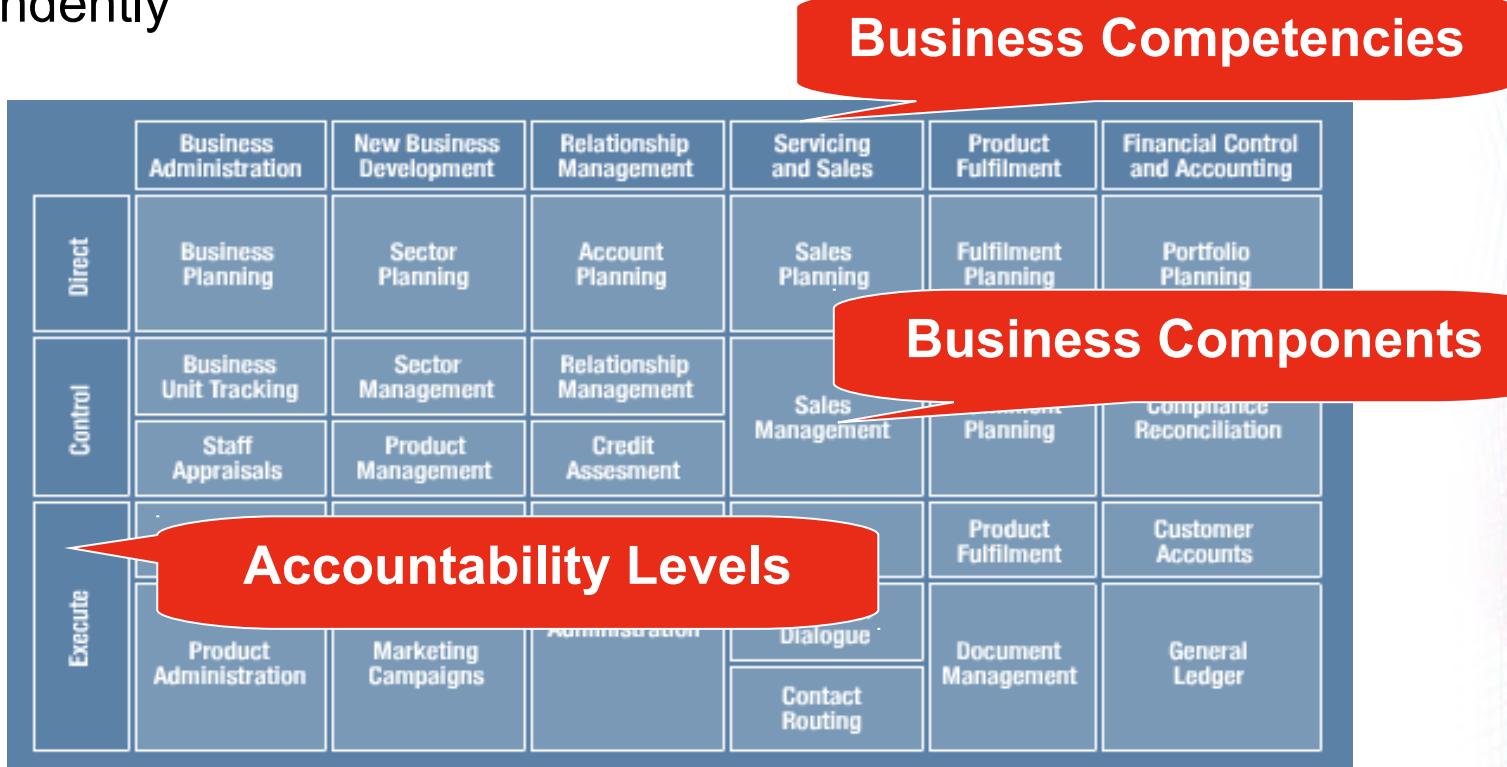
- **Service Design**

Realization
decisions

- **Service Implementation**

Component Business Model

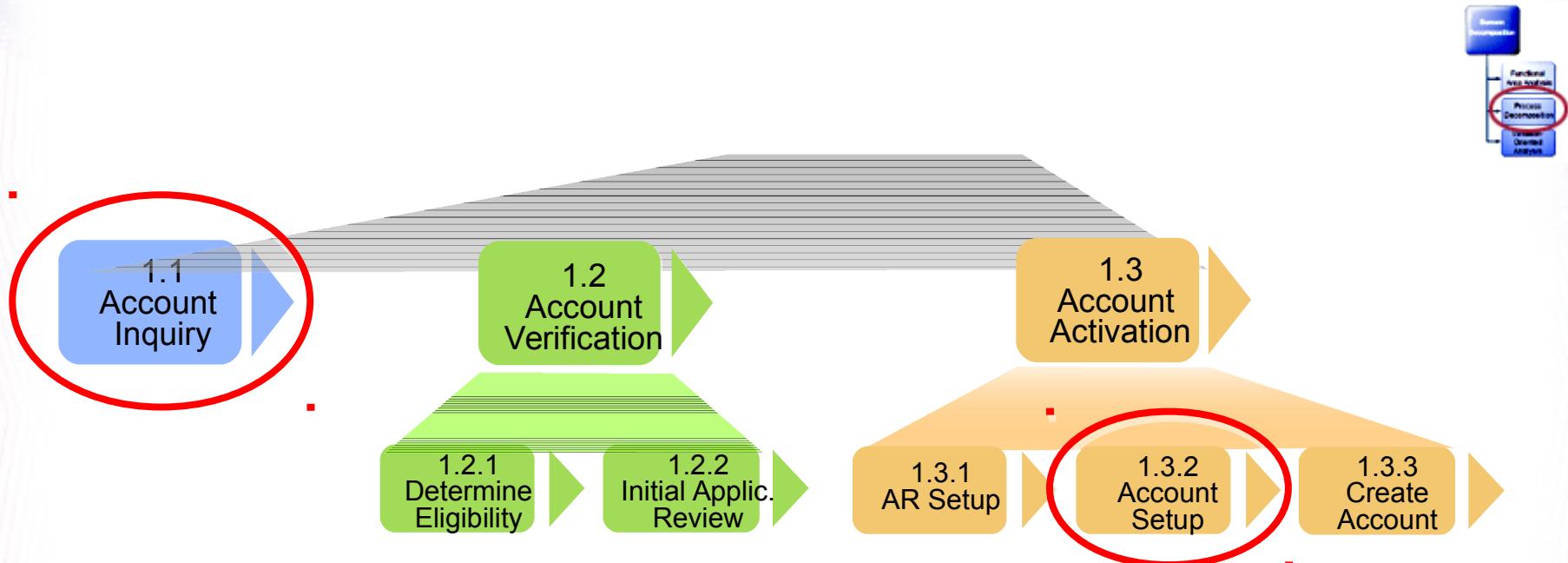
- CBM is a technique to decompose an Enterprise into its constituent building blocks (business components)
- A component is a logical grouping of people, technology, and resources that delivers specific business value, and can operate independently



Heat Map

		Business Administration	New Business Development	Relationship Management	Servicing and Sales	Product Fulfilment	Financial Control and Accounting
		Business Planning	Sector Planning	Account Planning	Sales Planning	Fulfilment Planning	Portfolio Planning
		Business Unit Tracking	Sector Management	Relationship Management	Sales Management	Fulfilment Planning	Compliance Reconciliation
Direct		Business Planning	Sector Planning	Account Planning	Sales Planning	Fulfilment Planning	Portfolio Planning
Control		Business Unit Tracking	Sector Management	Relationship Management	Sales Management	Fulfilment Planning	Compliance Reconciliation
Execute		Staff Appraisals	Product Management	Credit Assesment	Sales	Product Fulfilment	Customer Accounts
		Staff Administration	Product Delivery	Credit Administration	Customer Dialogue	Document Management	General Ledger
		Product Administration	Marketing Campaigns		Contact Routing		

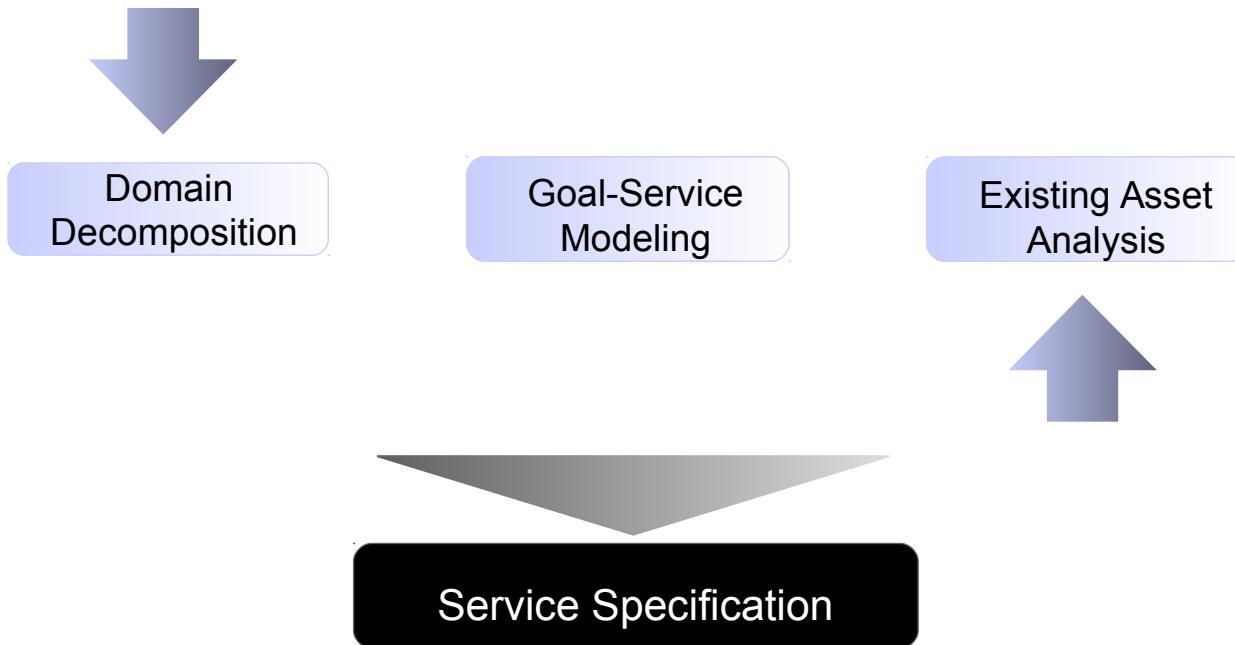
Process Decomposition



- Analyze business processes and sub-processes to a level of detail where candidate services can be identified

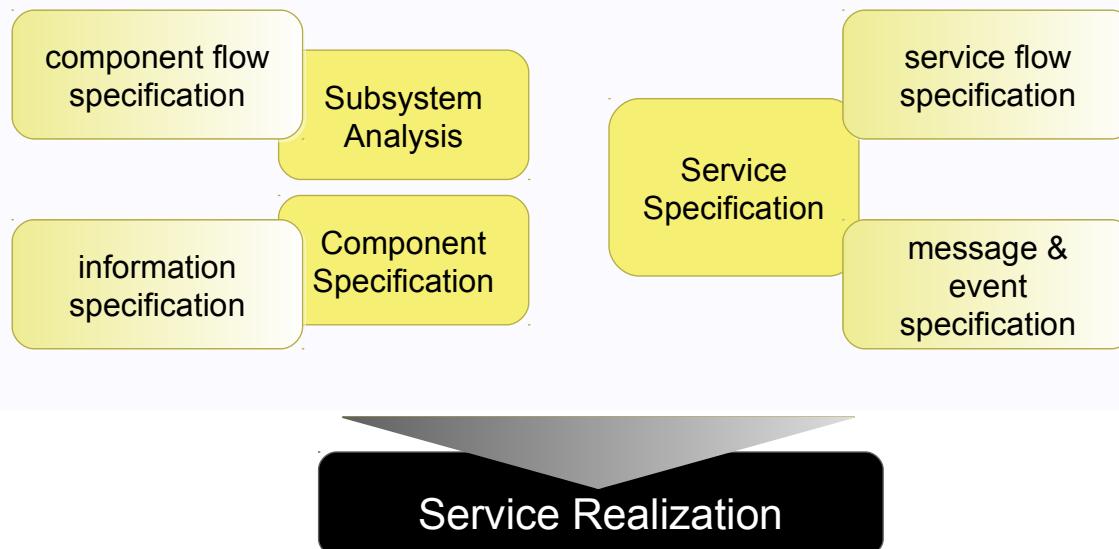
Service Identification

- A combination of three strategies is used
 - Goal-Service Modeling
 - Domain Decomposition (Top down Analysis)
 - Existing Asset Analysis (Bottom-up Analysis)



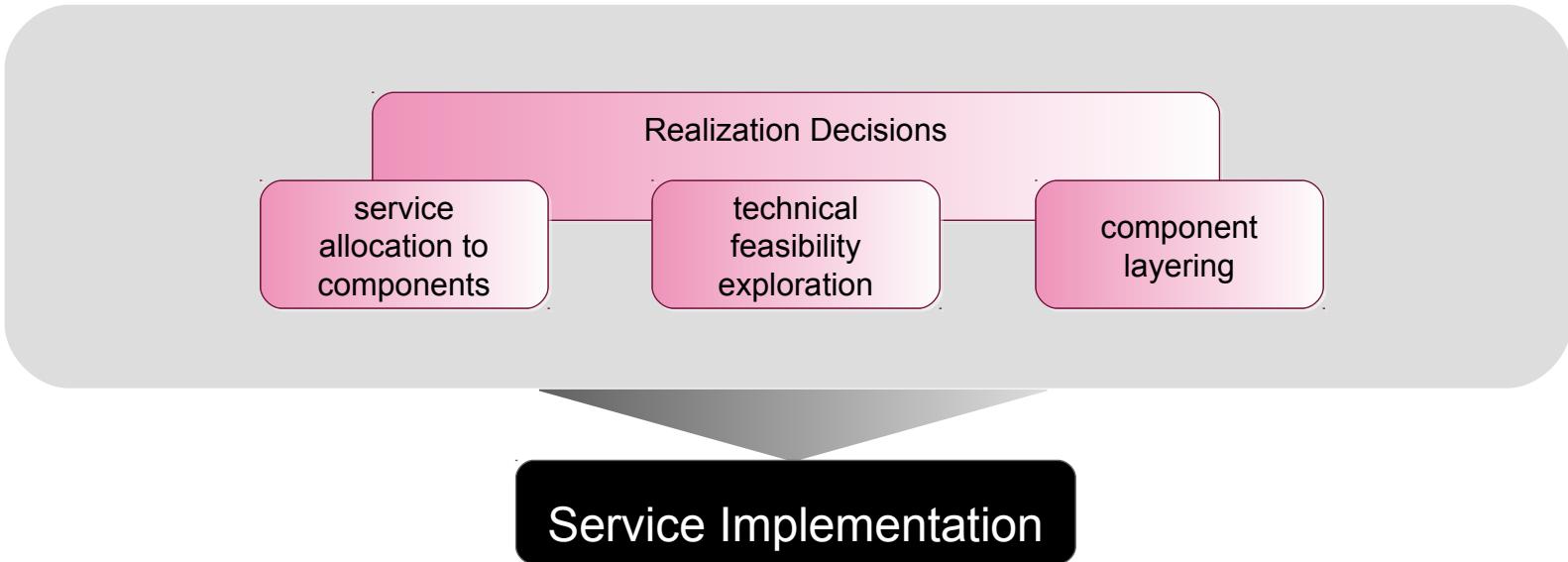
Service Specification

- Service Specification
 - Elaborates the Service Model. For example: service dependencies, composition, non-functional requirements, service message specifications, design decisions, etc.
 - 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, and messages



Service Realization

- Review the Service Model to determine how each Service is realized
 - Decide how Services are Realized:
buy, build, integrate, transform, subscribe, outsource
 - Document the Service Realization Decisions

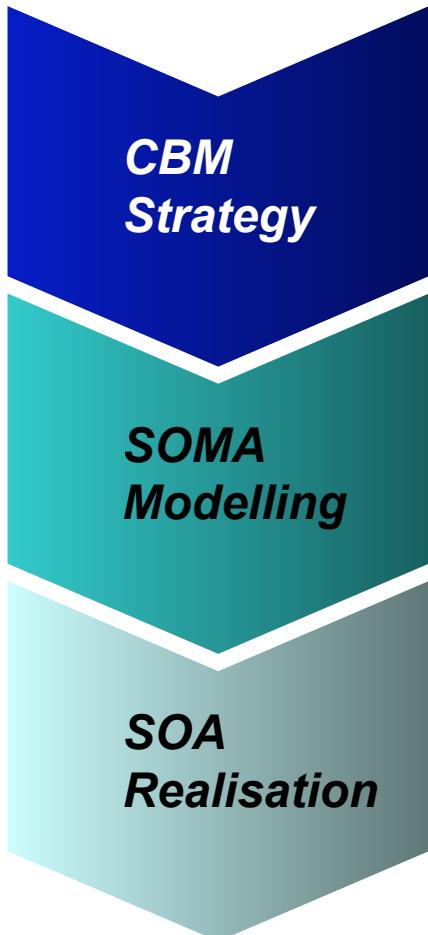


What Problems May Arise?

Building a composite application

- **Service Granularity**
 - Advice is to create services that are business relevant
 - What's that mean?
- **Interface Definition**
 - Objective: a single, stable interface
 - Problem: Service requestors want “easy to use” interface
- **Service Reuse**
 - Difficult to determine reusability at outset
- **Change Control**
 - Changing the implementation of the service is not supposed to affect requestor, but ...
- **How to Build the entire protocol stack**
 - Connectivity, Security, transactions, recovery...
 - See ws-i.org

Top Down Approach to SOA



Step 1: Breakdown your business into components

Decide what is strategically important and what is just operations in the value chain domain

*Analyse the different KPIs attached to these components
Prioritise and scope your transformation projects*

Step 2: Define a Service Model

*Identify your business service based on business components
Identify your business processes*

*Specify the services processes and components accordingly
Make SOA realisation decisions based on architectural decisions*

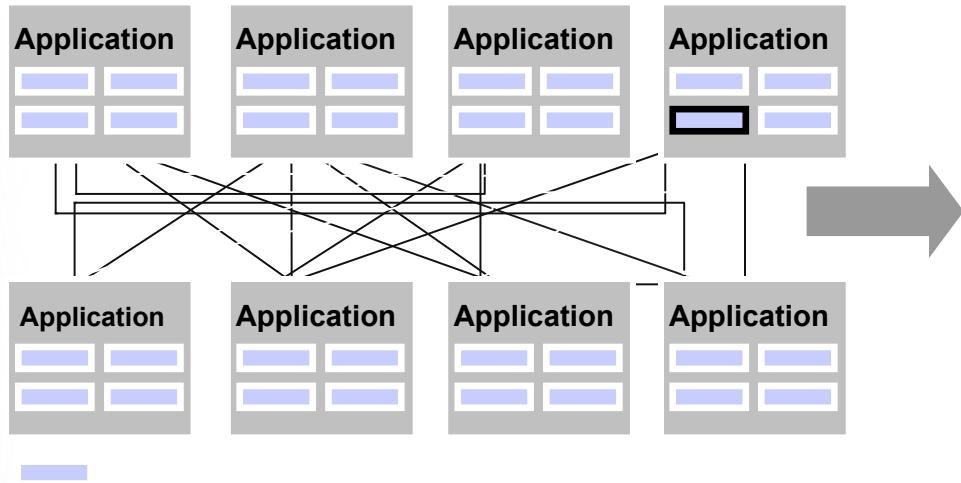
Step 3: Implement a Service Model

Develop a service-oriented architecture to support the Componentized Business

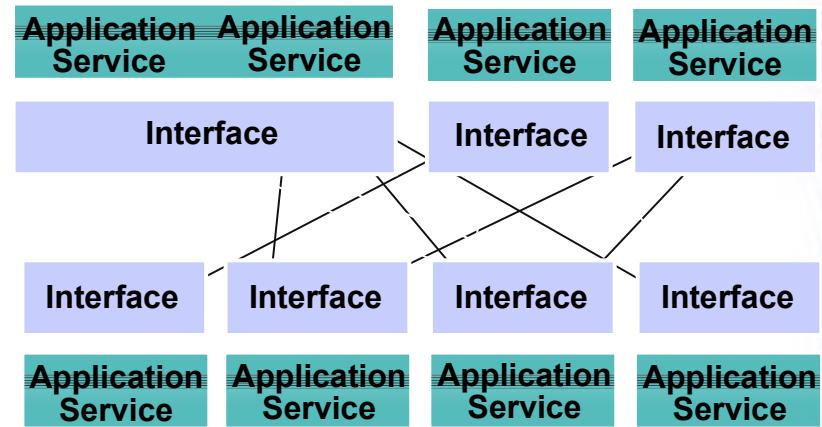
*Implement service based scoping policy for projects
Implement appropriate governance mechanism*

Web Services decouples interfaces from their applications...

Turn this ...



...into this (web services).



- ✓ Rich business abstractions describe the application interface
- ✓ Decouples the interfaces from the business applications
- ✓ The number and complexity of the interfaces is reduced
- ✓ Business applications and their interfaces become reusable

But separate connection points still leaving bloated interfaces

Service Orchestration

Orchestration vs. Choreography

Orchestration

A Single Director In Control



Choreography

Defines Interaction.
WS-Choreography
describes Publicly
Visible Message
Exchange



The Perfect Match: SOA + Orchestration

SOA + Orchestration =

Externalized Processes & Process Visibility

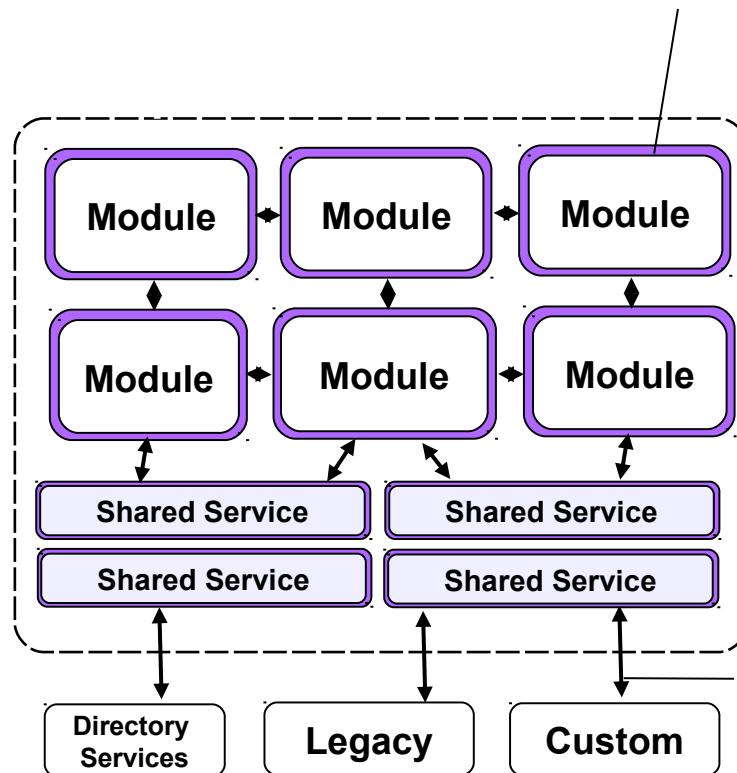
High Degree Of Flexibility

Low Cost Of Change

Business Agility

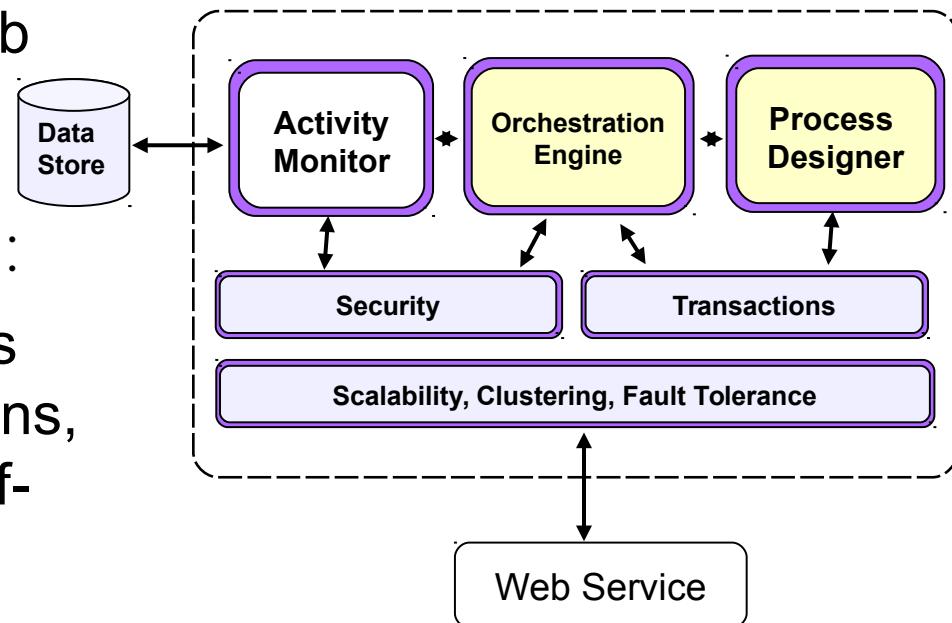
Product Suites are Migrating to SOA

- Cumbersome monolithic suites with self-sufficient architectures are **out**
- Replaced by decoupled application modules with web service interoperability wrappers
 - ✓ Lower acquisition and maintenance costs
 - ✓ Enables best-of-breed module assembly
 - ✓ Reduces vendor lock-in



Service Orchestration Solution

- Integration and process automation between web services only
- Standardized BPEL process orchestrations :
- Ideal fit for organizations with SOA implementations, and multi-vendor best-of-breed BPM approach

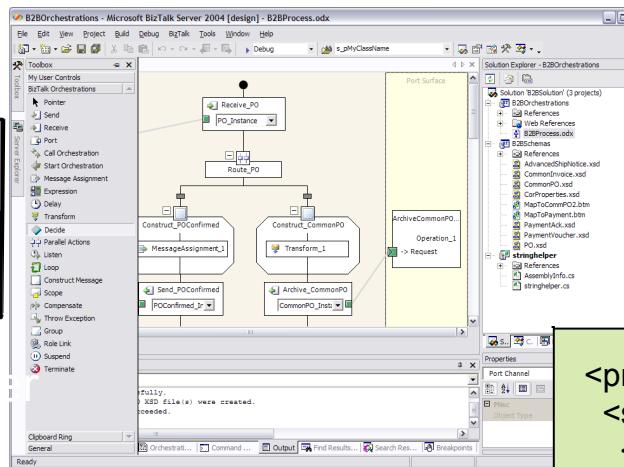


Orchestration Components

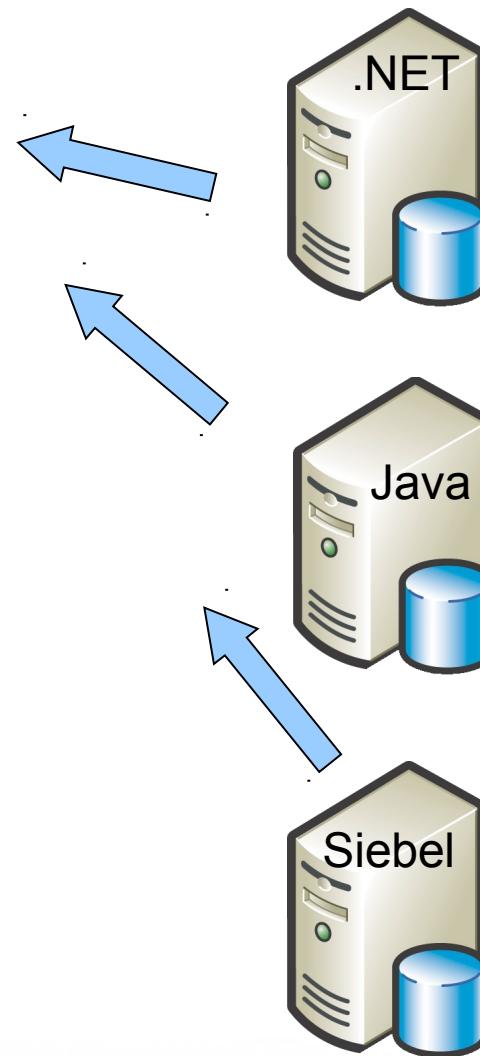
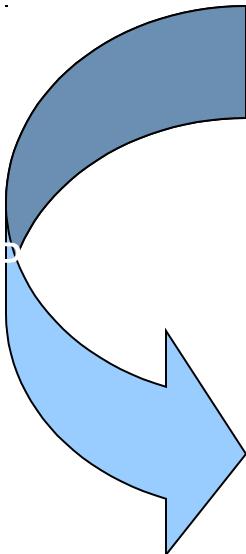
Design Time Tools

- Service Management (WSDL, endpoints)
- Define Process Flow
 - Data Mapping
 - Debugging
 - Deployment
- Expose Process As A Web Service
- Execute Processes
- (De-)Hydrate Process State
 - Process State Monitoring
 - Exception Management

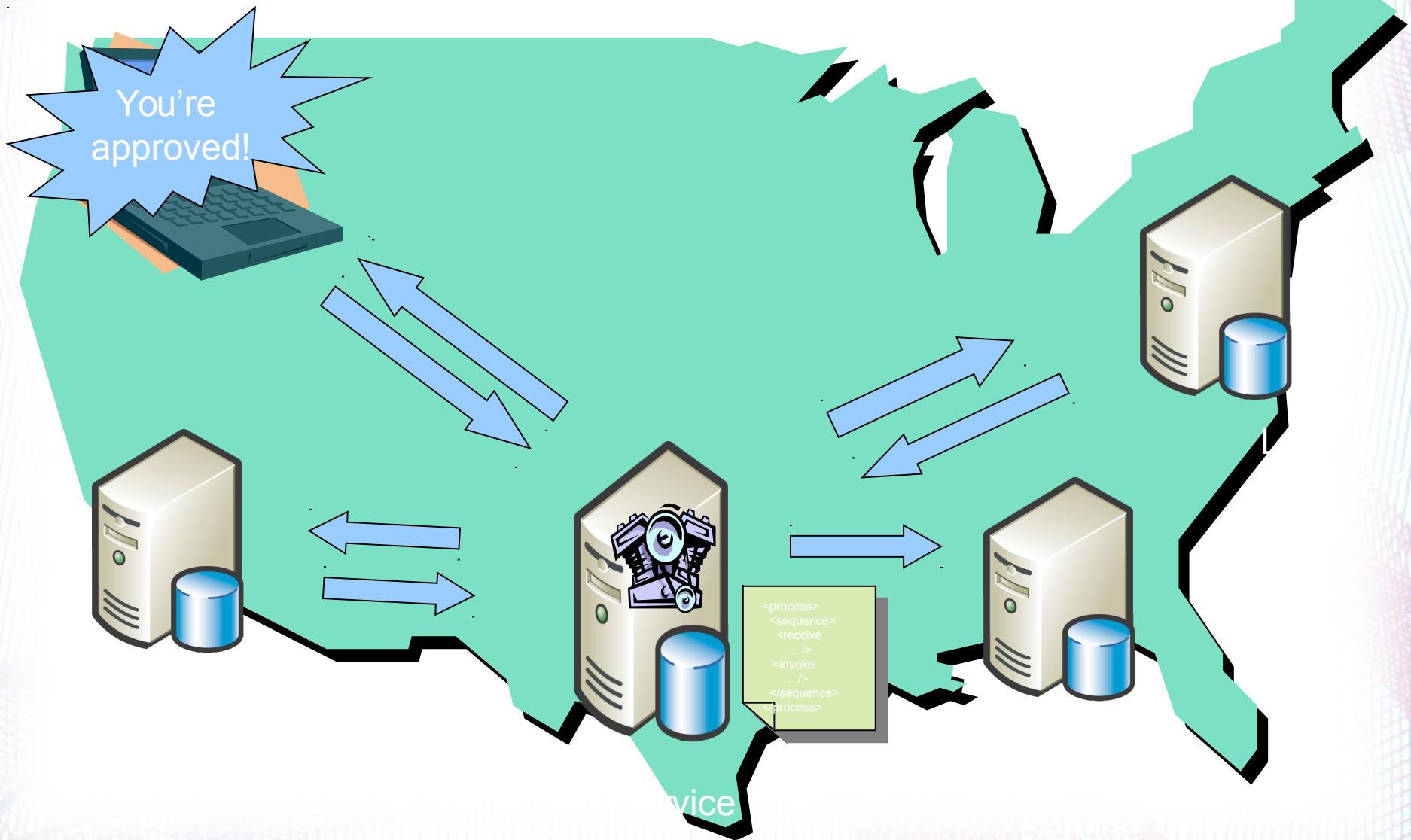
Developing Orchestrations



```
<process>
  <sequence>
    <receive
      ... />
    <invoke
      ... />
  </sequence>
</process>
```

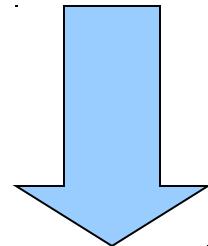


Executing Orchestrations



Orchestration Environments

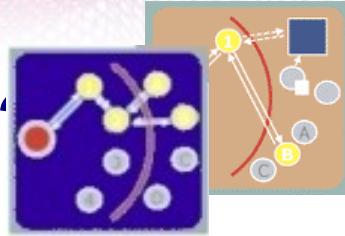
- Limited Support Orchestration Standards
 - webMethods
 - BizTalk 2000 / 2002 (2004)
 - Intalio
 - ...
- Industry Rallies around BPEL
 - OpenStorm Orchestration Suite
 - Collaxa
 - Microsoft BizTalk 2004
 - IBM Web Sphere Process Choreographer
 - Many more...



Service Orchestration With BPEL

What can you do with BPEL?

What Is BPEL?

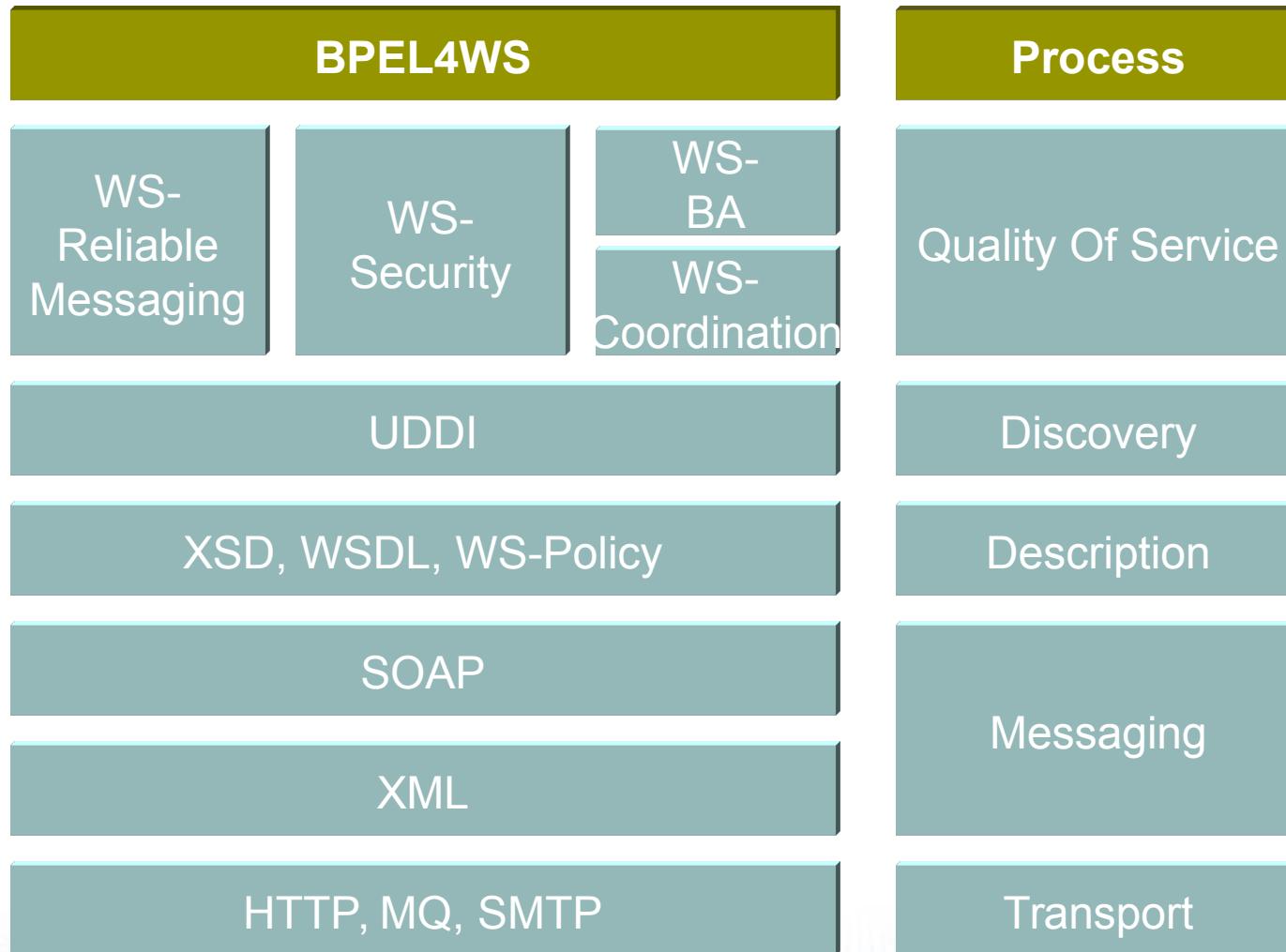


- **Business Process Execution Language**, platform independent, XML-based
- A language to specify the behavior of business processes *between Web services and as Web services*
- Contains process flow constructs for conditional branching, parallel processes, nested sub-processes, process joins, etc.
- Uses WSDL to describe process interfaces so that they can be more easily integrated into other processes or applications.
- Provided as an open standard under royalty free terms.

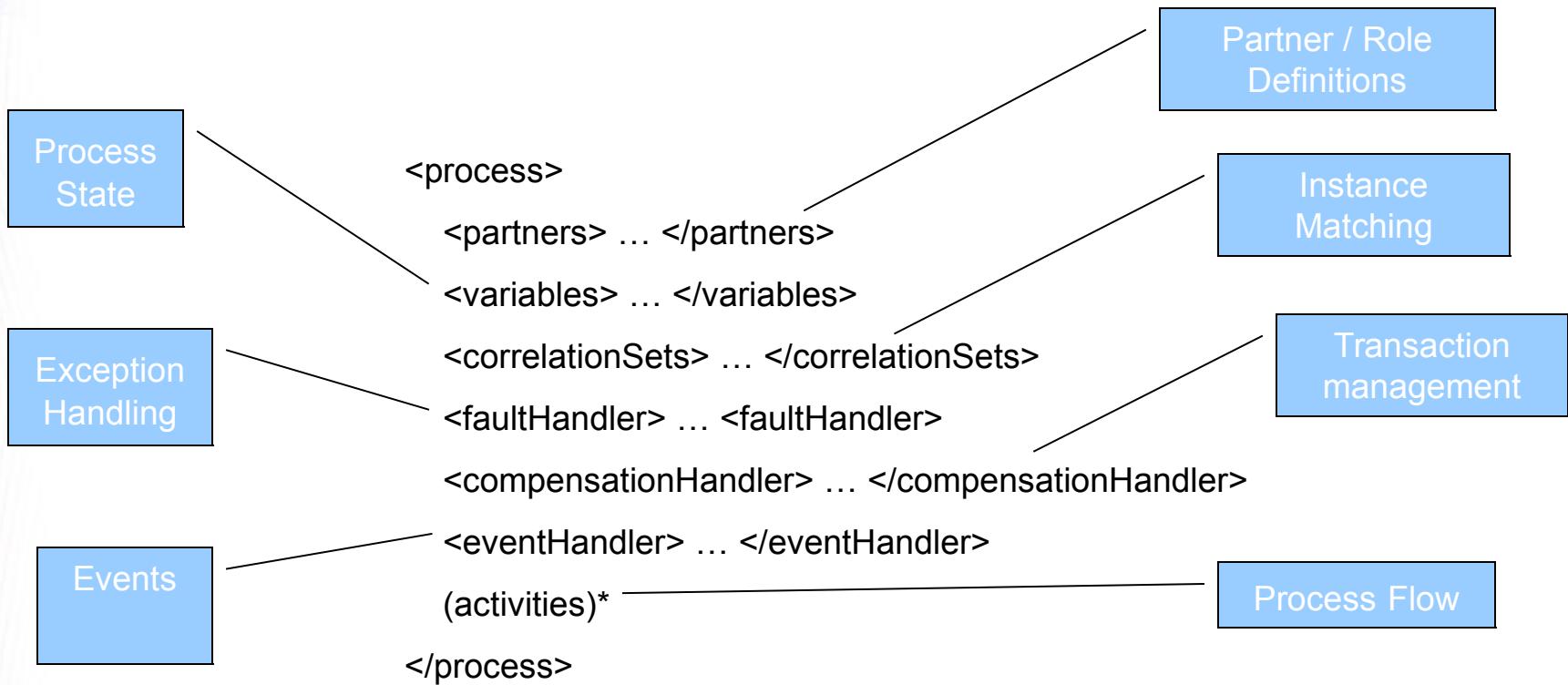
BPEL Goals

- Platform independence – XML based language (Java, .Net implementations available)
- Services and Messages are first order citizens
- Provide an asynchronous programming model
- Leverage runtime metadata based interface like WSDL
- ‘enterprise ready’ (security, transactional, reliable, scalable, etc.)

BPEL In The Web Service Stack



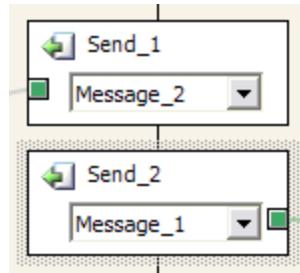
BPEL Structure Overview



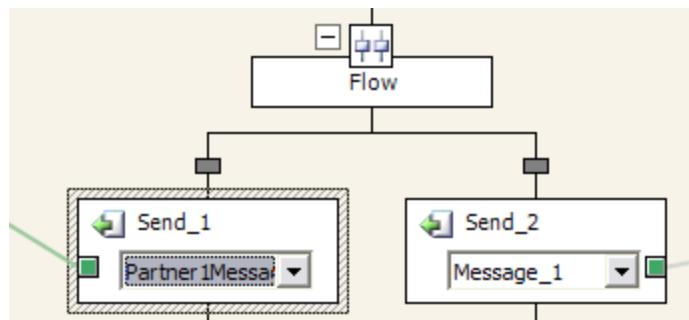
activities = `<receive>`, `<reply>`, `<invoke>`, `<assign>`, `<throw>`,
`<terminate>`, `<wait>`, `<empty>`, `<sequence>`, `<switch>`,
`<while>`, `<pick>`, `<flow>`, `<scope>`, `<compensation>`

BPEL Activities

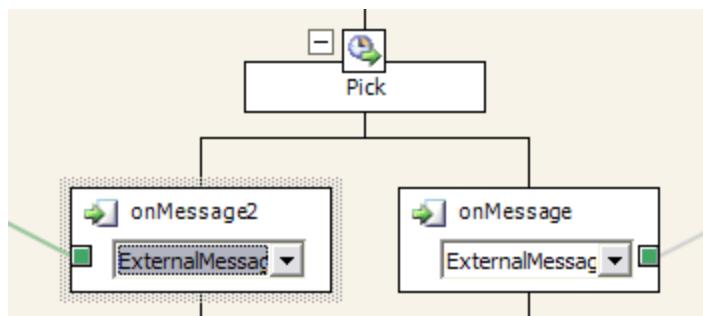
- <sequence>



- <flow>

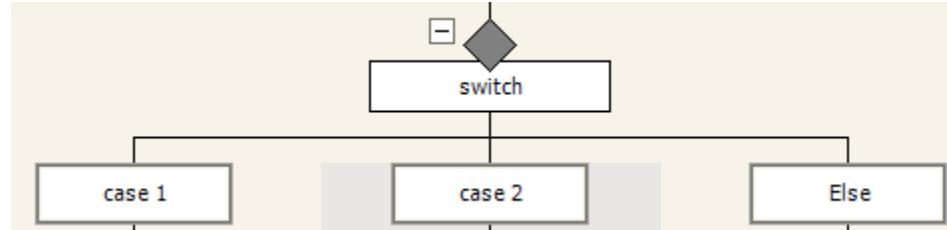


- <pick>

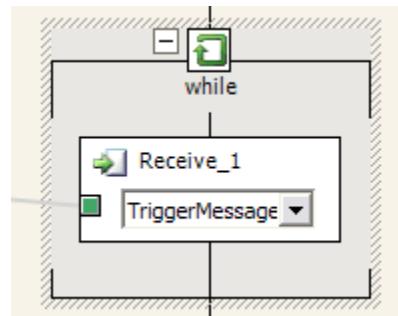


BPEL Flow Control

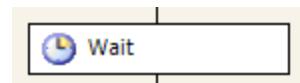
- <switch> <case /> <otherwise /> </switch>



- While



- Wait



BPEL Data Manipulation

- Scoped via <scope>
- Assignments between Variables using XPath 1.0

```
<assign>
  <copy>
    <from variable="AutoLoanRequest"
           part="creditRating"
           query="/creditRating/text()"/>
    <to
      variable="InterstateCarLoanRequest"
      part="credit"
      query="/credit/text()"/>
  </copy>
</assign>
```

BPEL Message Correlation

```
<propertyAlias propertyName="orderNumber"
  messageType="POMessage" part="PO"
  query="/PO/Order"/>
<correlationSets>
  <correlationSet name="PurchaseOrder"
    properties="orderNumber"/>
</correlationSets>
...
<receive partnerLink="Buyer" portType="PurchasingPT"
  operation="AsyncPurchase"
  variable="PO">
  <correlations>
    <correlation set="PurchaseOrder" initiate="yes">
  </correlations>
</receive>
```

The diagram illustrates the BPEL code with two oval shapes representing correlation sets. One oval encloses the entire section from the first `<propertyAlias>` to the end of the `<correlationSets>` block. Another oval encloses the `<correlationSet name="PurchaseOrder">` block. Two arrows point from the bottom of these ovals down to the `<correlation>` element in the `<correlations>` block of the `<receive>` activity.

BPEL Transactions

- Support for transactional integrity

- **<compensationHandler>**

```
<invoke partnerLink="Seller"  
       operation="SyncPurchase"  
       ...>  
  
...
```


<compensationHandler>

```
<invoke partnerLink="Seller"  
       operation="CancelPurchase" .../>  
  </compensationHandler>  
  </invoke>
```
 - Invoke via **<compensate>**

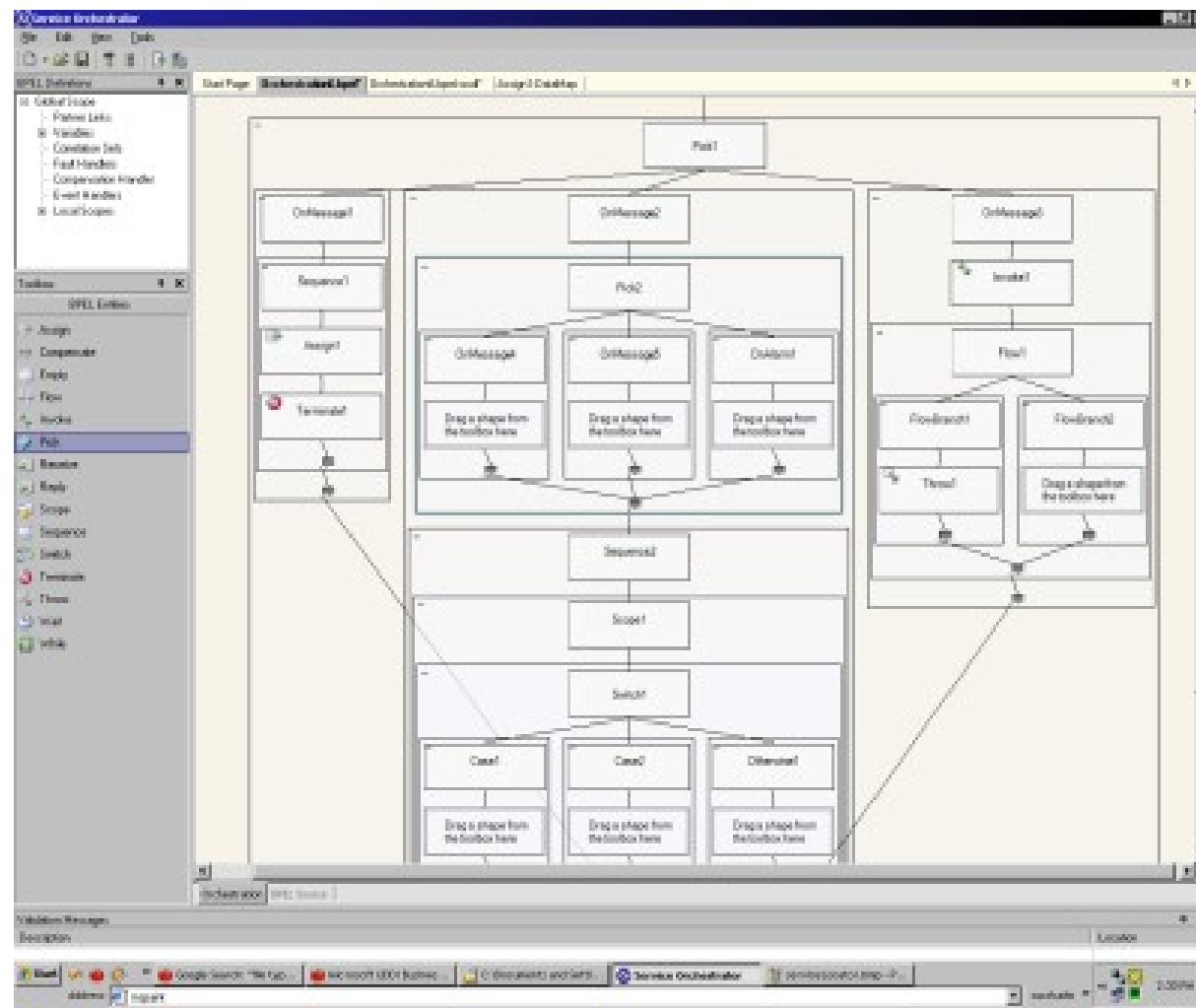
```
<switch>  
  <case ...>  
    <compensate/>  
  </case>
```

- Ongoing discussions in the use of WS-Coordination

BPEL Error Handling

```
<faultHandlers>
    <catchAll>
        ...
    </catchAll>
</faultHandlers>
...
<switch>
<case condition=
    "bpws:getVariableProperty(stockResult
    ,level) > 100">
    <throw faultName="OutOfStock"/>
</case>
```

Demo



Orchestration Products

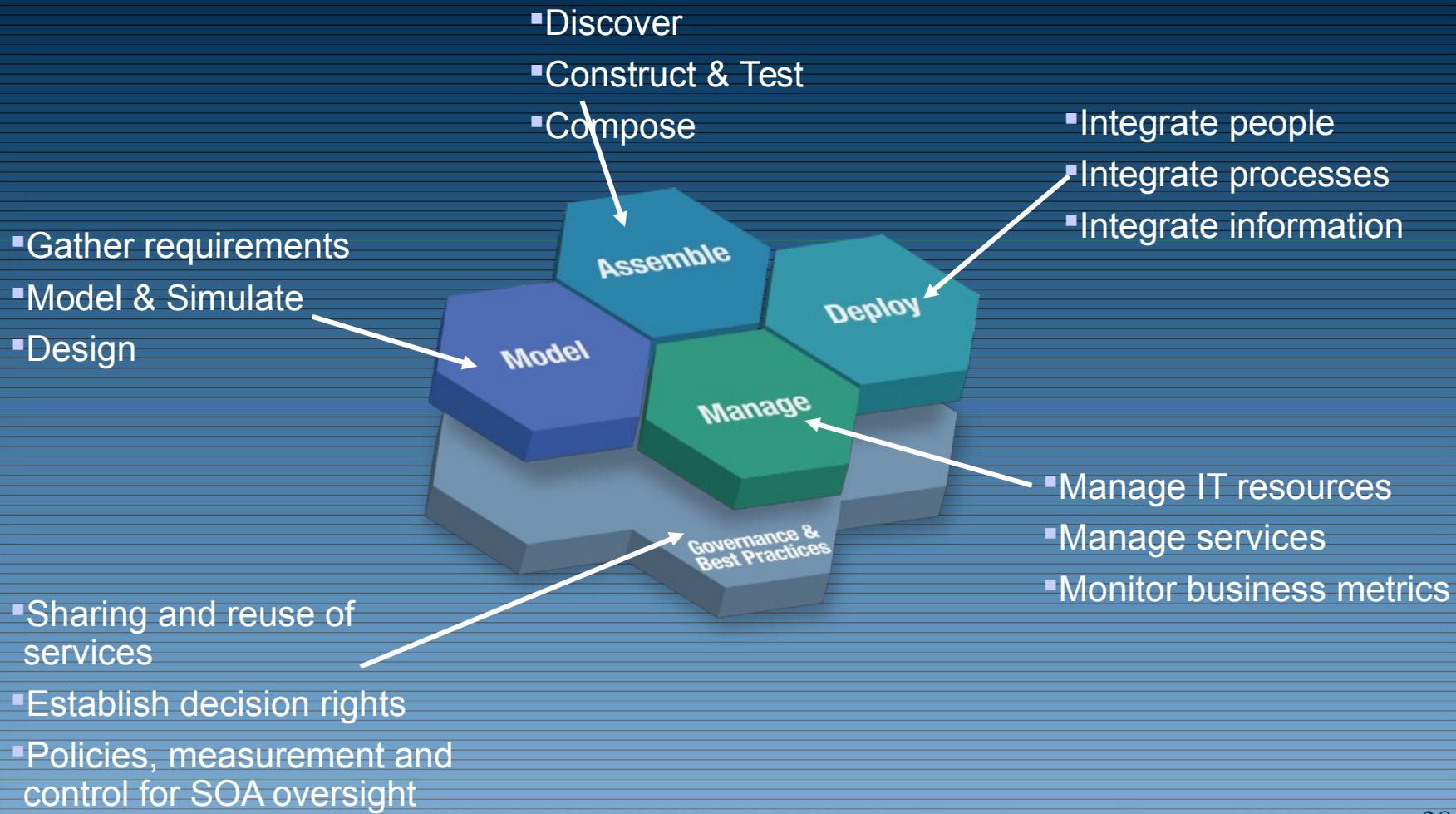
- OpenStorm Service Orchestrator
- Microsoft: BizTalk Server 2004, Visio
- Collaxa
- IBM WebSphere Process Choreographer
- Oracle SOA Suite, Jdeveloper
- Java Business Integration

...

Why BPEL?

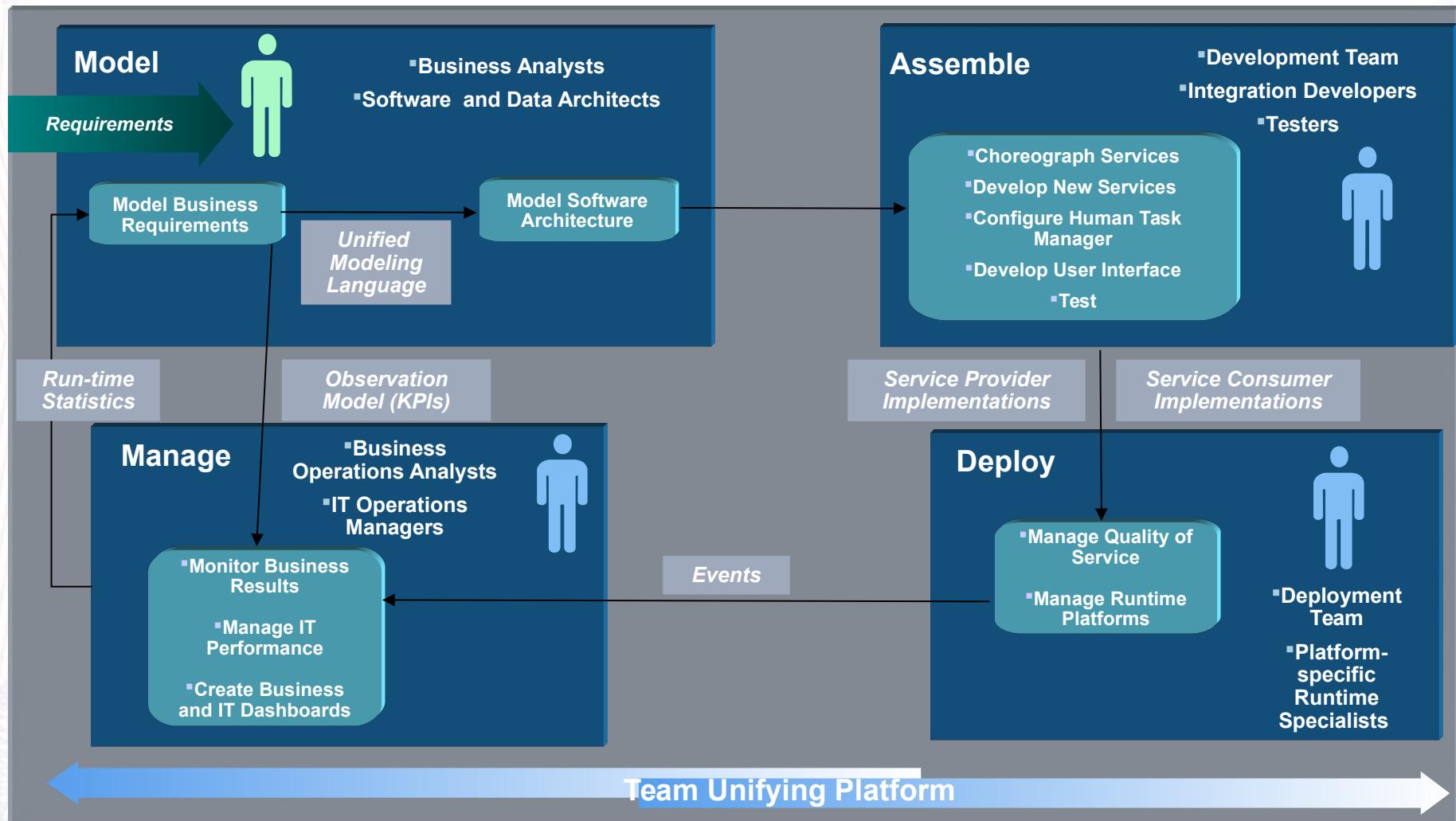
- Cost Pressure Through Choice
- Scale Up
- Mix-and-Match
- Open Standard
- Supported By Oracle And IBM and Microsoft

Follow the SOA Lifecycle for successful projects



A Key Principle: Business Driven Development

An Iterative, Business-focused Development Process

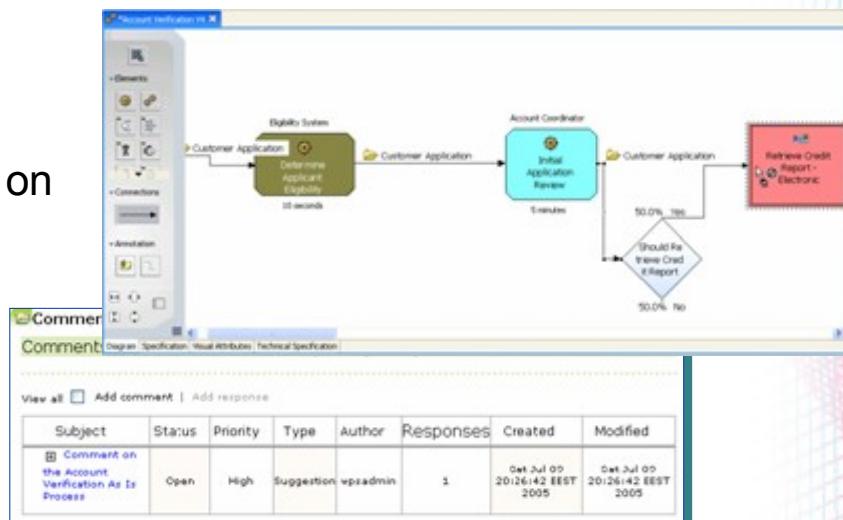


Model – For the Business Analyst



Business Modeler

- **Expanded user experience and analytical capabilities**
 - More granular and precise modeling of activities
- **Collaborative modeling to enable team work**
 - Ability to publish, share and comment on models through the web
- **Business performance modeling**
 - Ability to identify, generate and receive feedback on Key Performance Indicators
- **Microsoft Visio importing**



Assemble

Integration Developer

Streamlining process design hand-off between business and IT

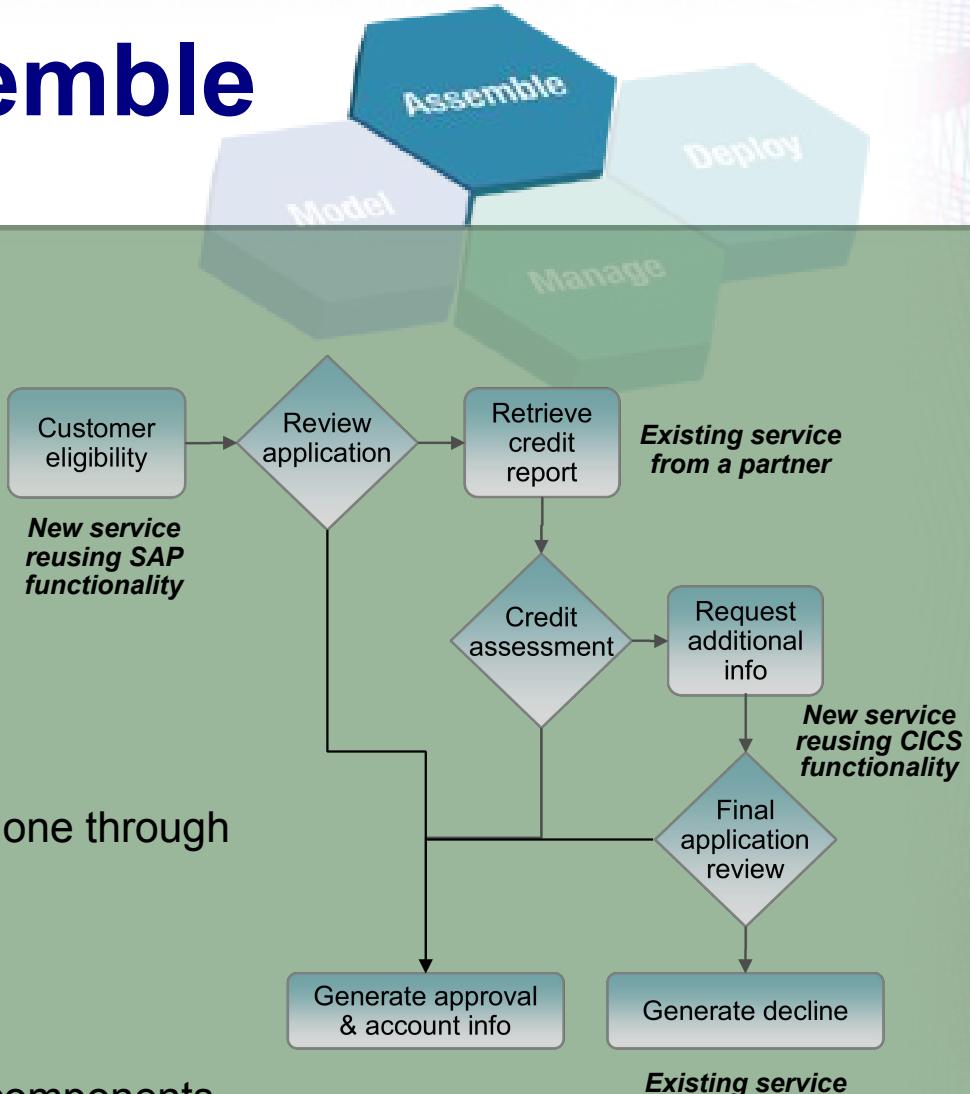
- Import and work with business process models directly from the business analyst (WebSphere Modeler)

Simplifying and speeding development

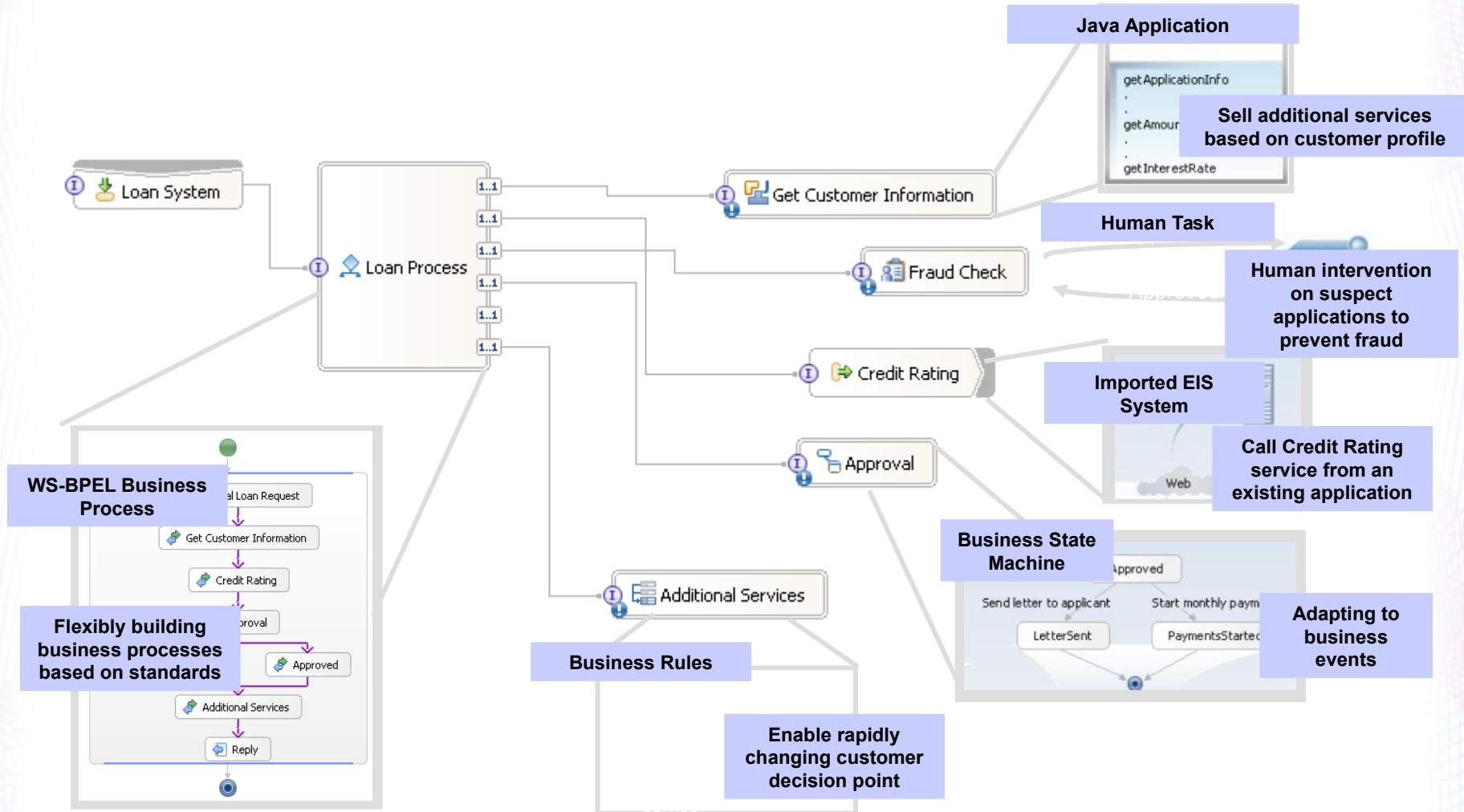
- Easy to use tools where everything can be done through the GUI

Maximizing re-use

- Ability to leverage existing services & save components for future service reuse



Example Composite Application



Deploy: Integrating the applications in SOA



Flexible connectivity infrastructure for integrating applications and services to power SOA

ESB: a new product delivering an Enterprise Service Bus

- Standards based connectivity including SOAP, XML, JMS, etc.

Message Broker: a new version delivering an advanced Enterprise Service Bus

- Universal connectivity including SOAP, XML, JMS, COBOL copybook, SCADA, etc.
- Advanced message transformation, enrichment and routing

Manage - For Business



Real-time visibility into process execution

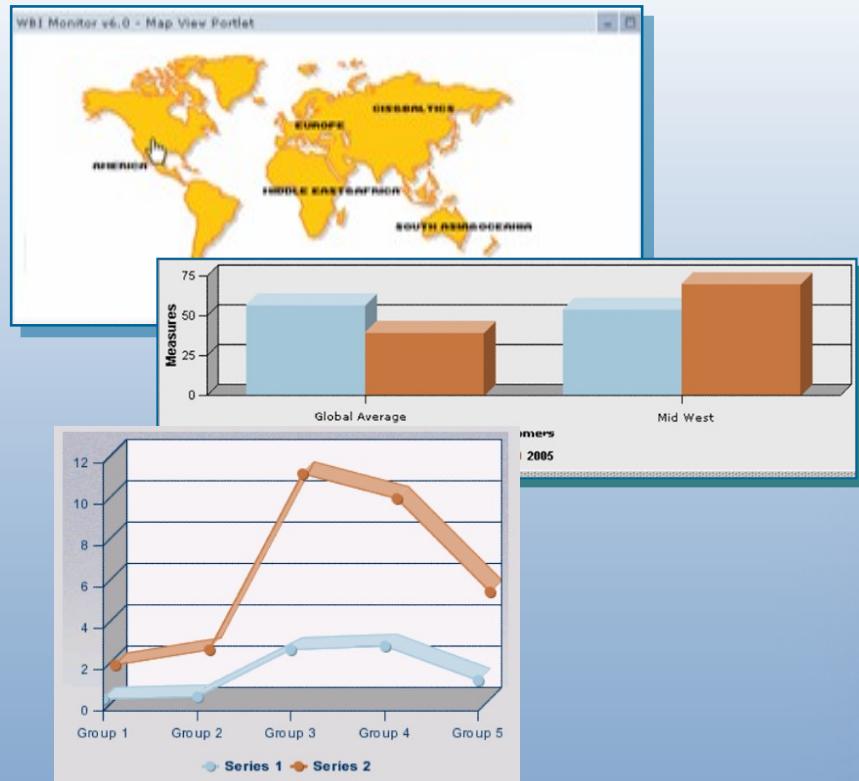
- Management dashboards and reporting capabilities, including trending information
- Event alerts tailored to various types of users

Active intervention in process execution

- Action Manager – supporting real-time response and action as performance data is received

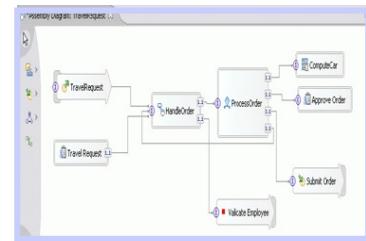
Supporting continuous process improvement

- Ability to export data to WebSphere Business Modeler for analysis and process improvement

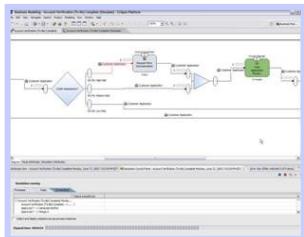


The Need for Business Process Monitoring

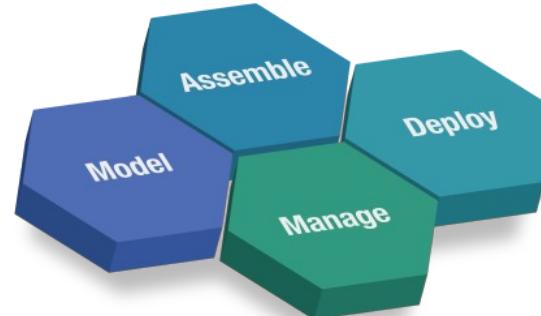
BPEL, WSDL,
SCA, and
Monitor Model
Artifacts



BPEL, ESB, SCA EARs
Monitor EARs



XML
(Monitoring
Results)



Process Server
ESB



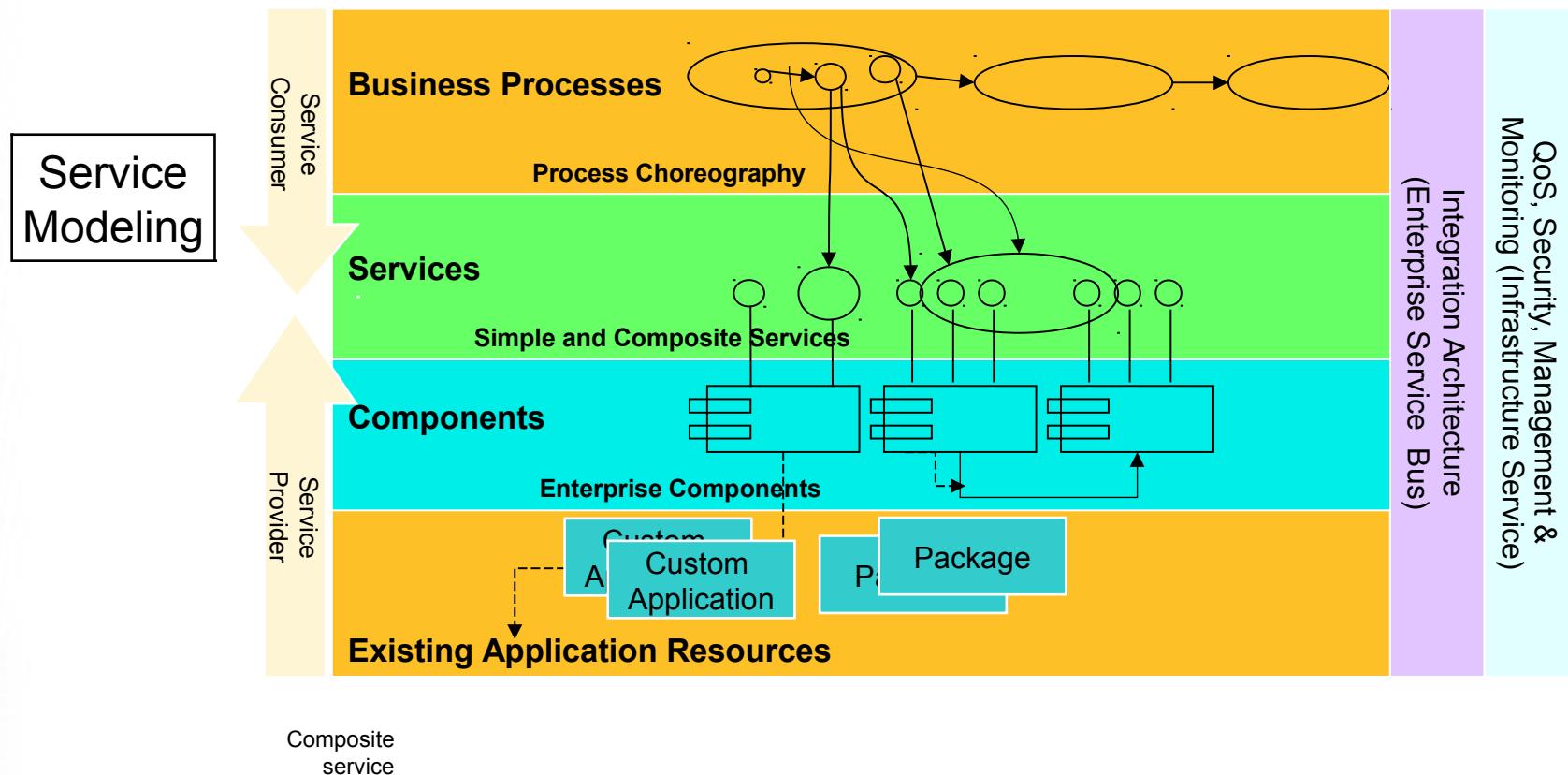
CBEs

Let us Develop SOA Apps with BPEL

Service-Oriented Solutions – Architecture View

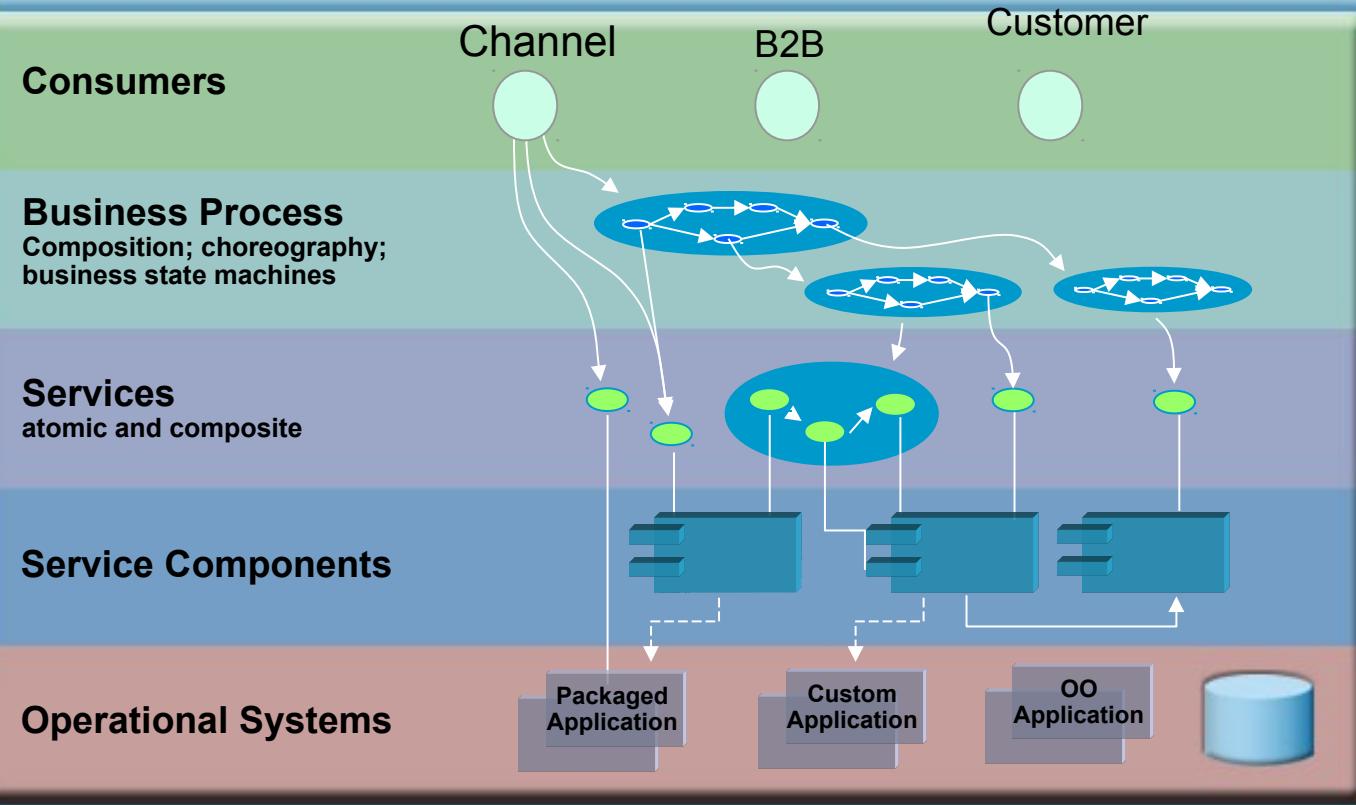
An SOA is composed of multiple layers.

At the heart of the SOA are Services, Components that realize services and Service Flows.



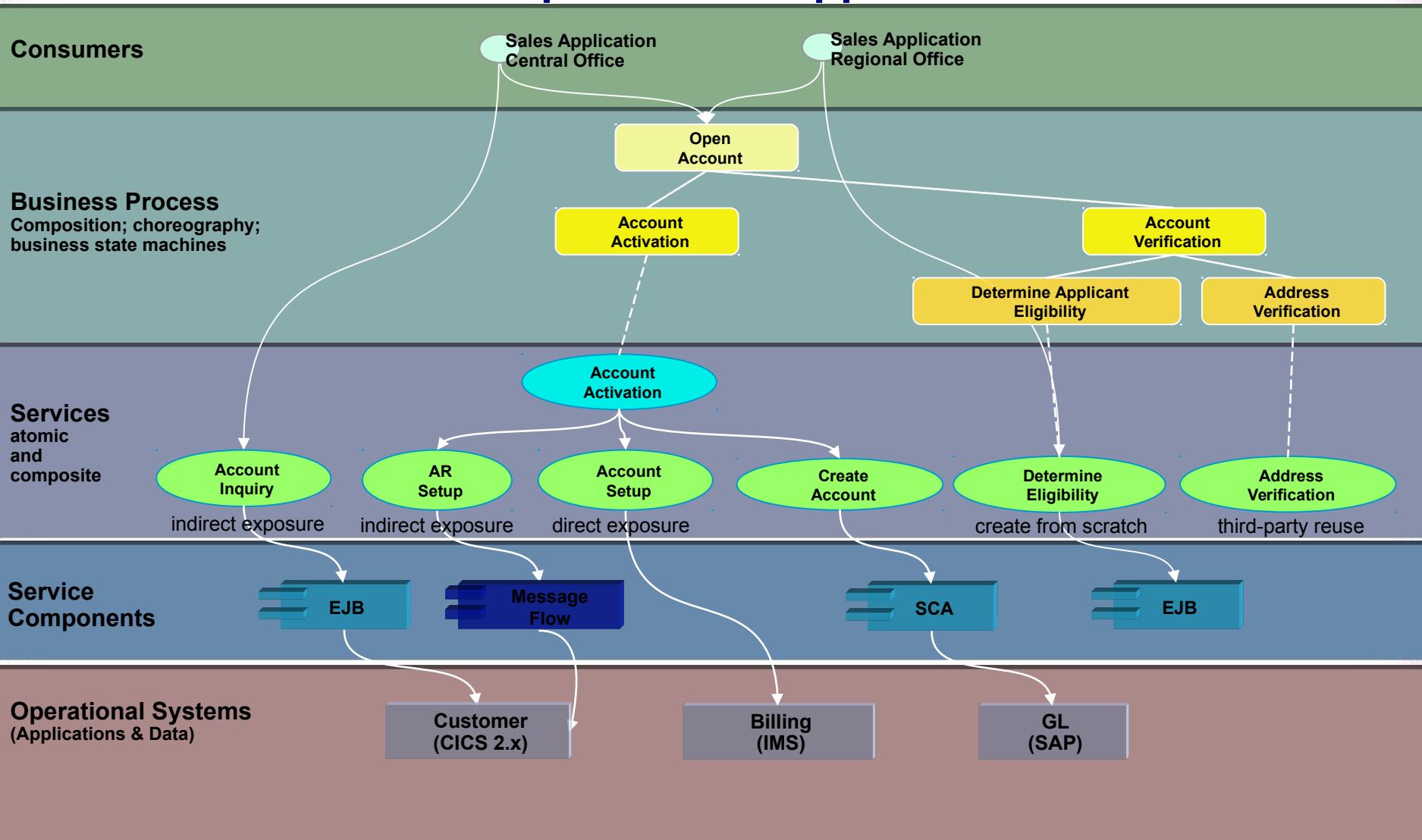
SOA Solution Layering

Leveraging the SOA Reference Architecture



Governance
Data Architecture (meta-data) &
Business Intelligence
QoS Layer (Security, Management &
Monitoring Infrastructure Services)
Integration (Enterprise Service Bus)

Example – Sales Application



Making Composite Application

How do you make it easier?

- Enabling applications to effectively communicate with each other
- Adding, removing and changing applications without disrupting the entire business
- Overcoming connectivity challenges:

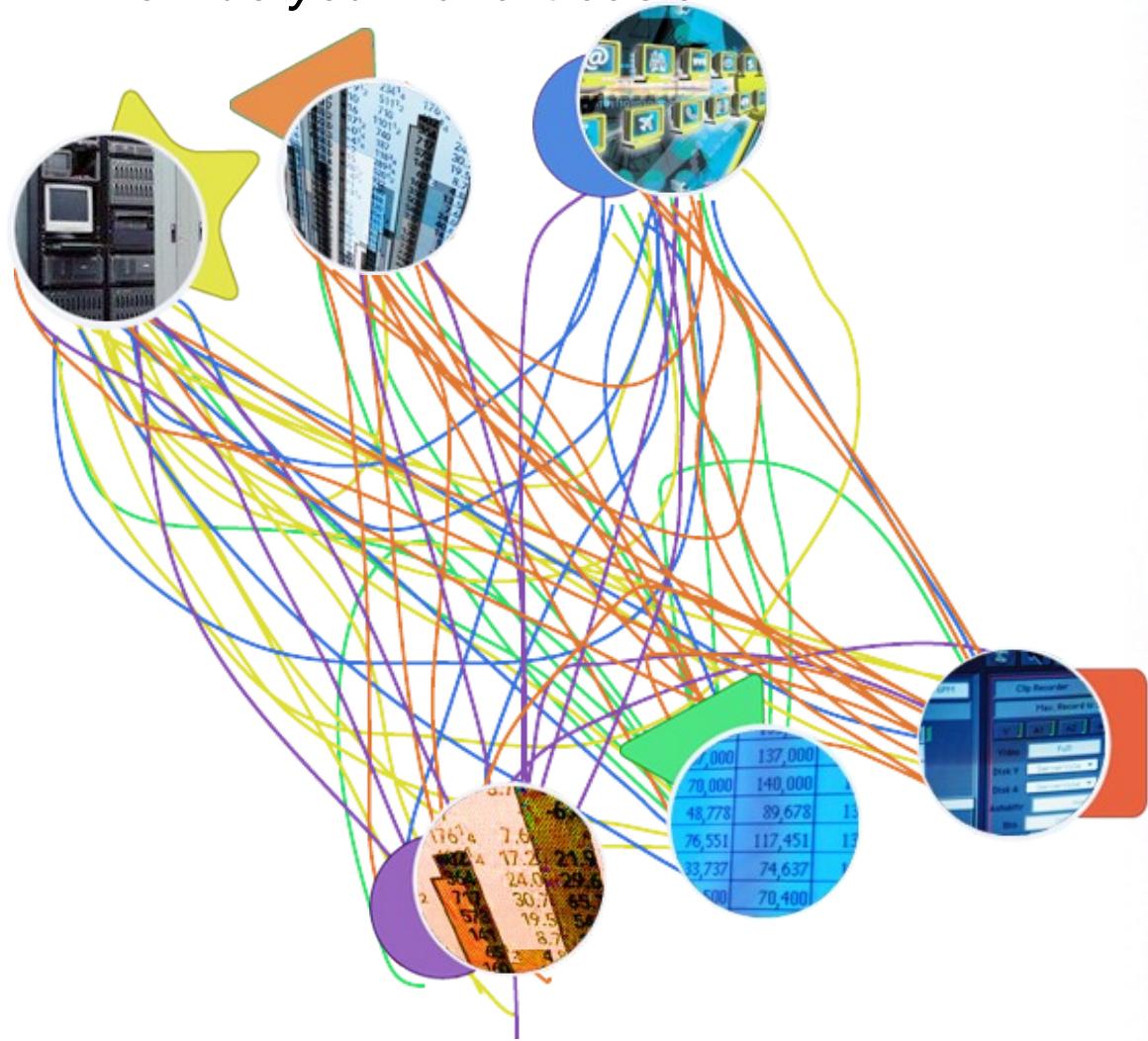
multiple programming models

multiple platforms

multiple message formats

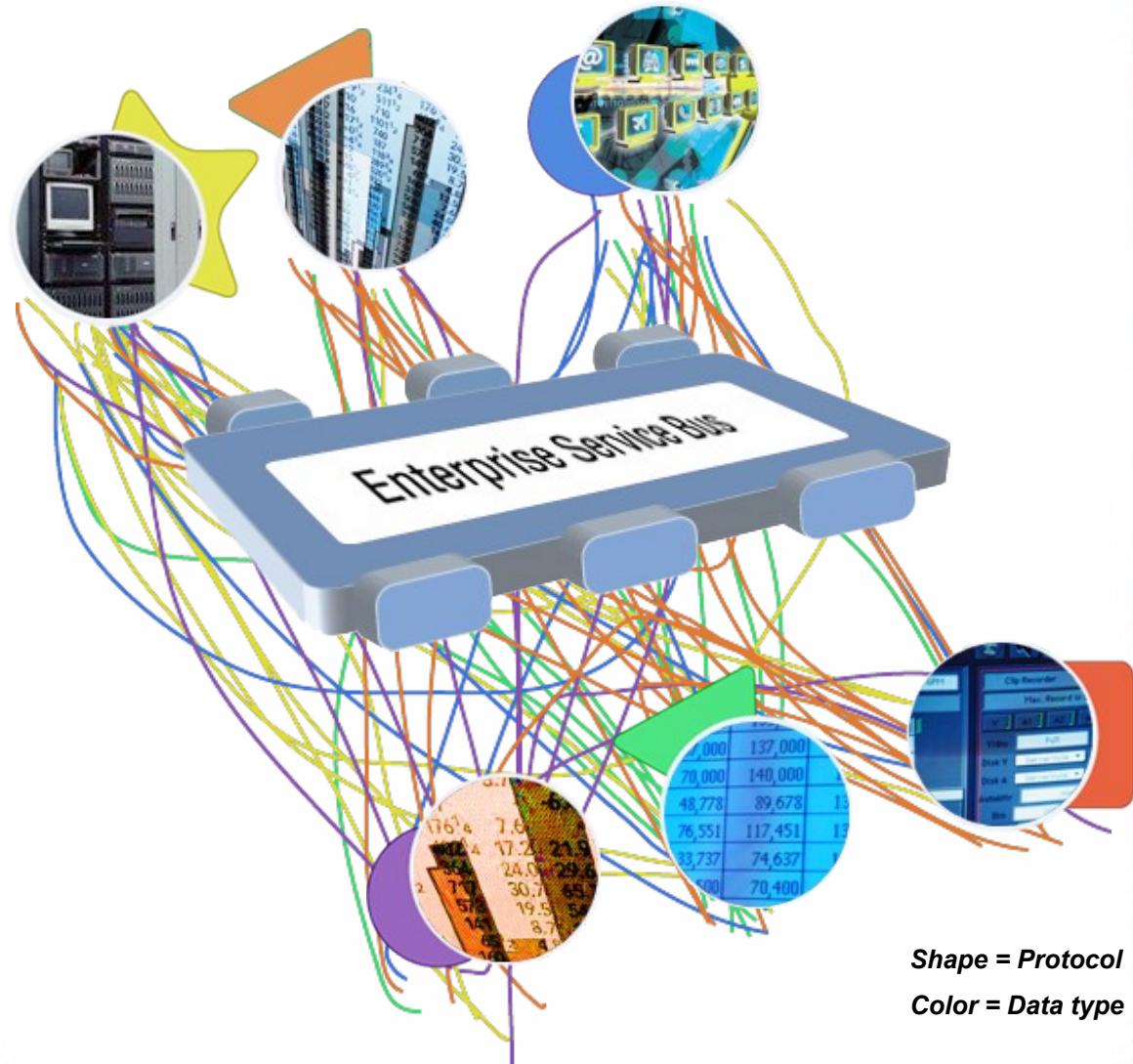
varieties of “standards”

multiple programming languages



Enterprise Service Bus (ESB)

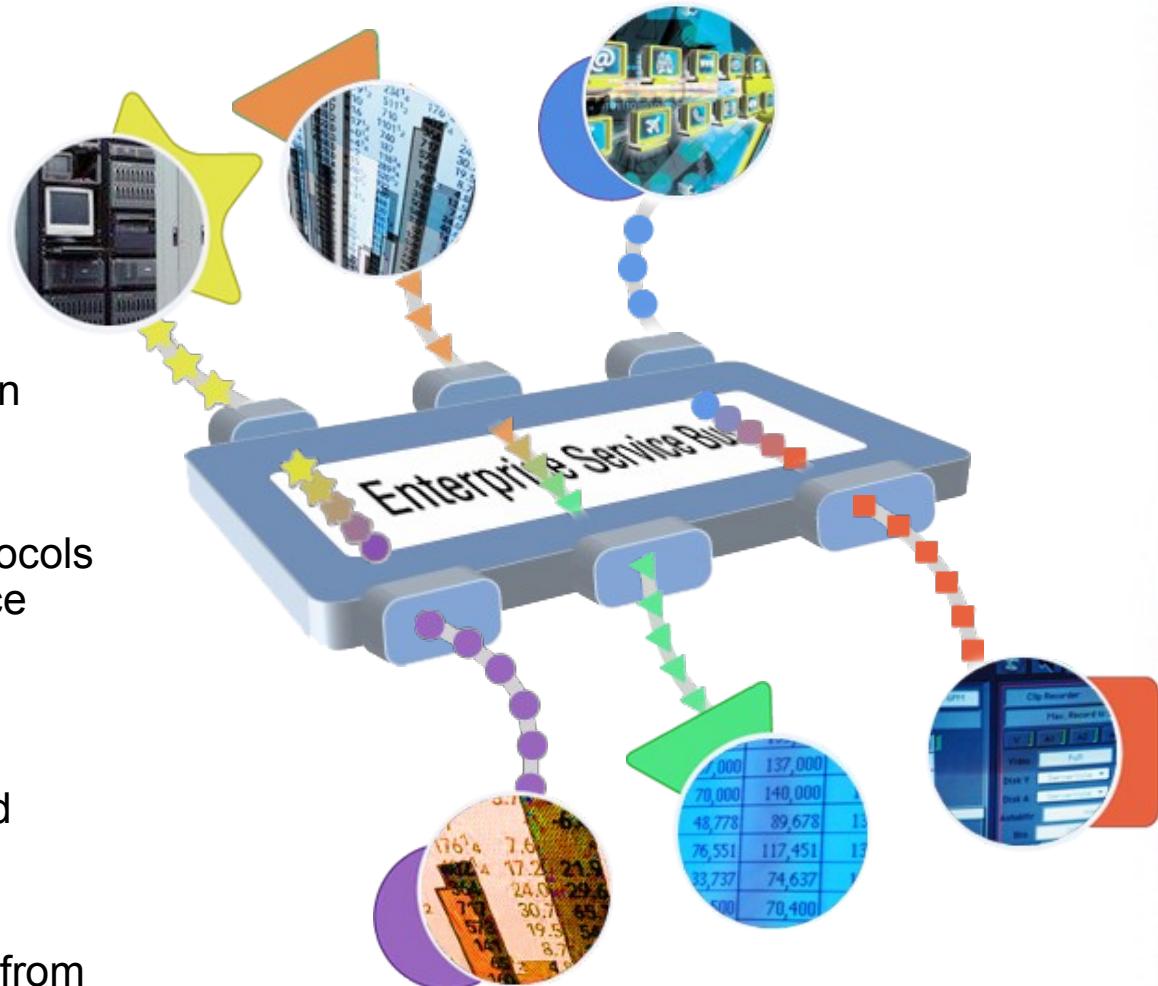
Flexible connectivity infrastructure for integrating applications and services to power your SOA



Enterprise Service Bus (ESB)

Flexible connectivity infrastructure for integrating applications and services to power your SOA

- ▶ **ROUTING** messages between services
- ▶ **CONVERTING** transport protocols between requestor and service
- ▶ **TRANSFORMING** message format between requestor and service
- ▶ **HANDLING** business events from disparate sources



Enterprise Service Bus

Central to the Workings of a Service Oriented Architecture

Mediation Services

Routing
Transformation

Event Services

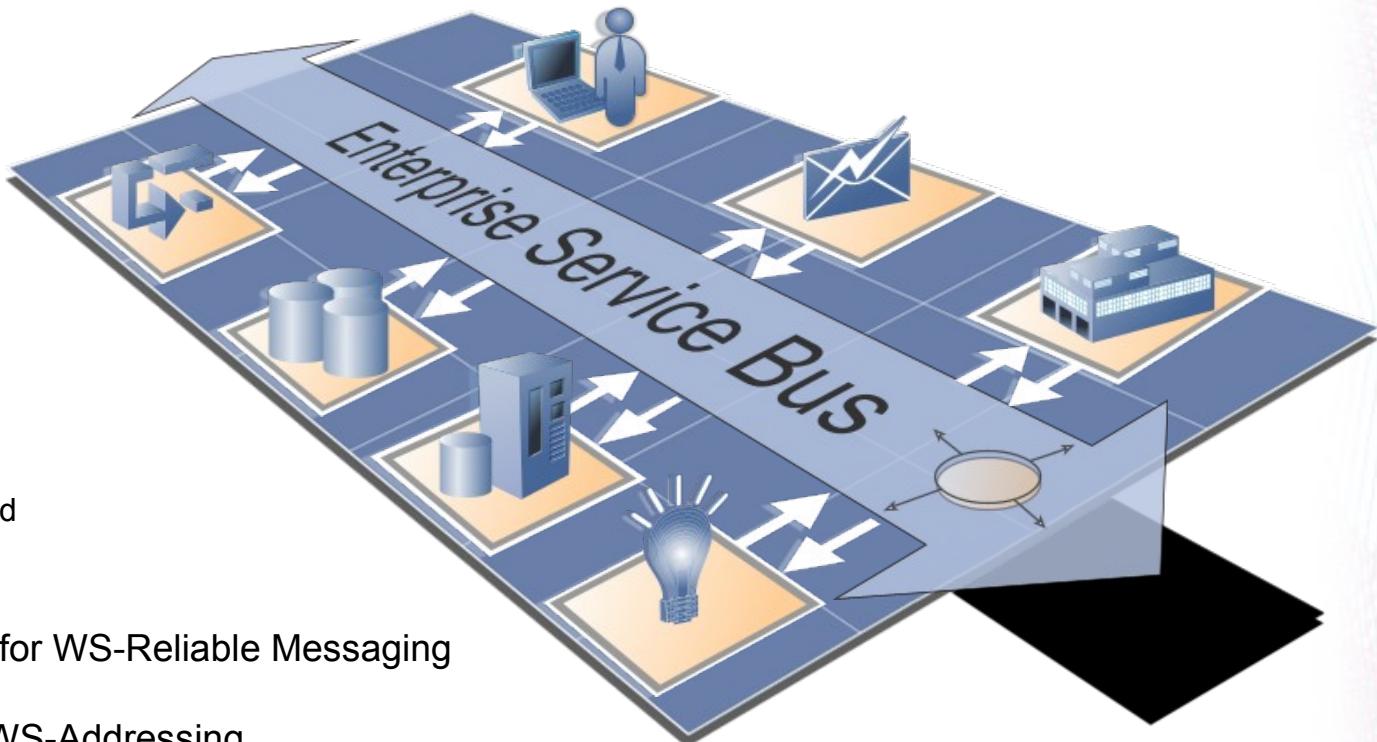
Publish and Subscribe

Transport Services

Synchronous/Asynchronous
Persistent/Non-persistent
Loosely-coupled/Tightly-coupled

Standards Based

~~Standards Based~~
HTTP/HTTPS with option for WS-Reliable Messaging
JMS, JAX-RPC, SOAP
WS-Security, WS-Policy, WS-Addressing

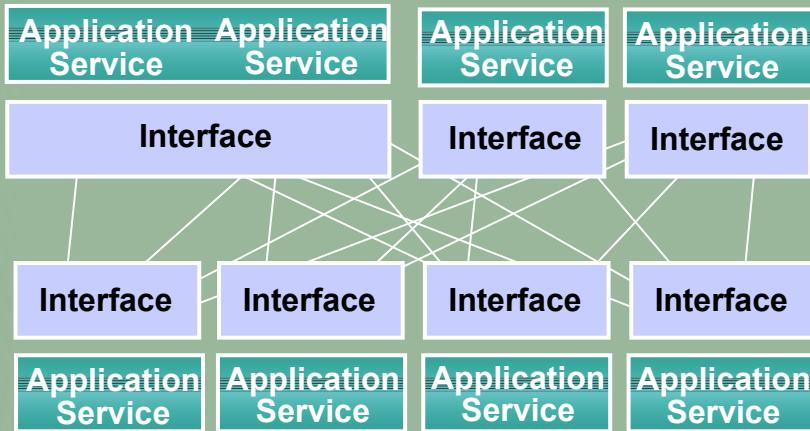


... with unsurpassed technical characteristics

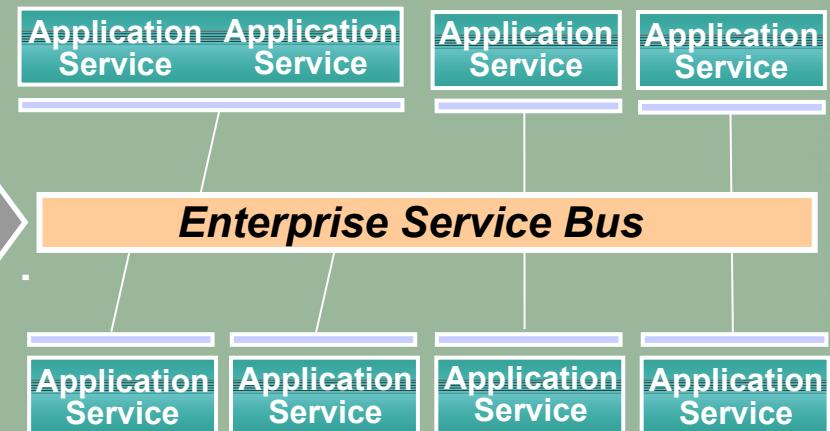
- Scale to match performance and throughput needs
- Security

The Enterprise Service Bus shrinks those interfaces further

Turn this (web services)...



...into this (SOA)



- ✓ Decouples the point-to-point connections from the interfaces
- ✓ Allows for dynamic selection, substitution, and matching
- ✓ Enables more flexible coupling and decoupling of the applications
- ✓ Enables you to find both the applications and the interfaces for re-use

RESULT → Greater Business Responsiveness

The Business Value of an Enterprise Service Bus



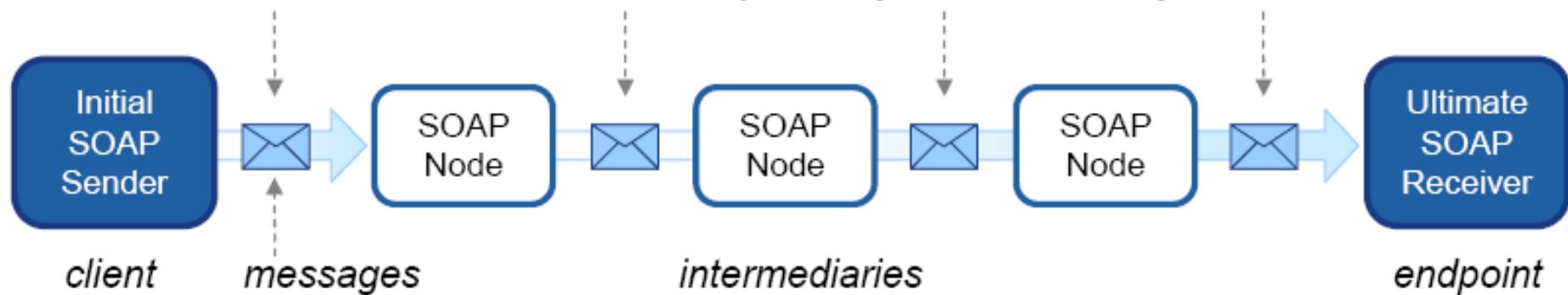
Add new services faster



Change services with no impact to existing services

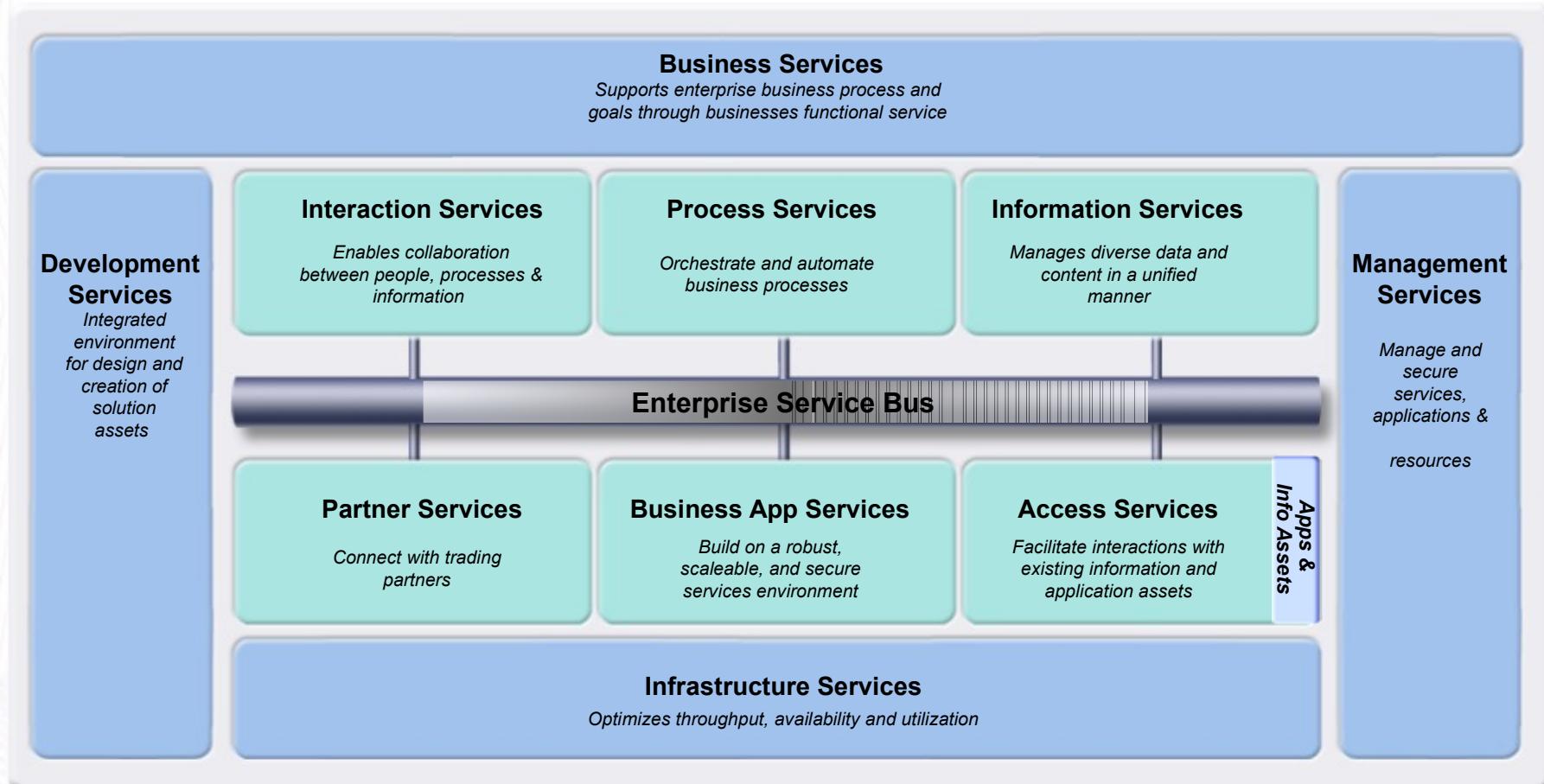
Mediation Capabilities

- Transformations: XML-to-XML translations
- Message Validation
- Content and quality service selections
- Content based routing
- Customised logging, metering & monitoring
- Policy management



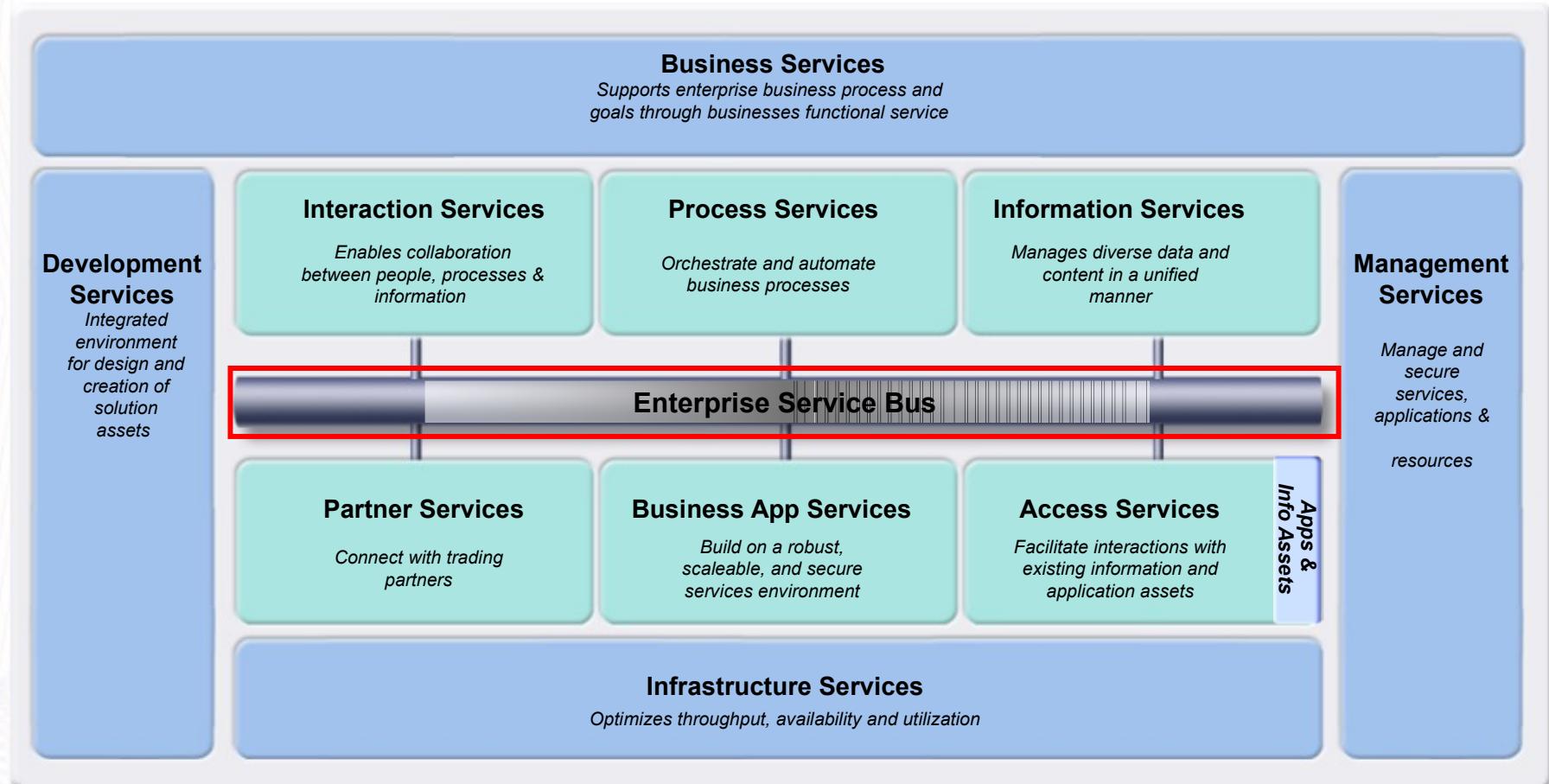
Defining the capabilities for your SOA environment

SOA Reference Architecture



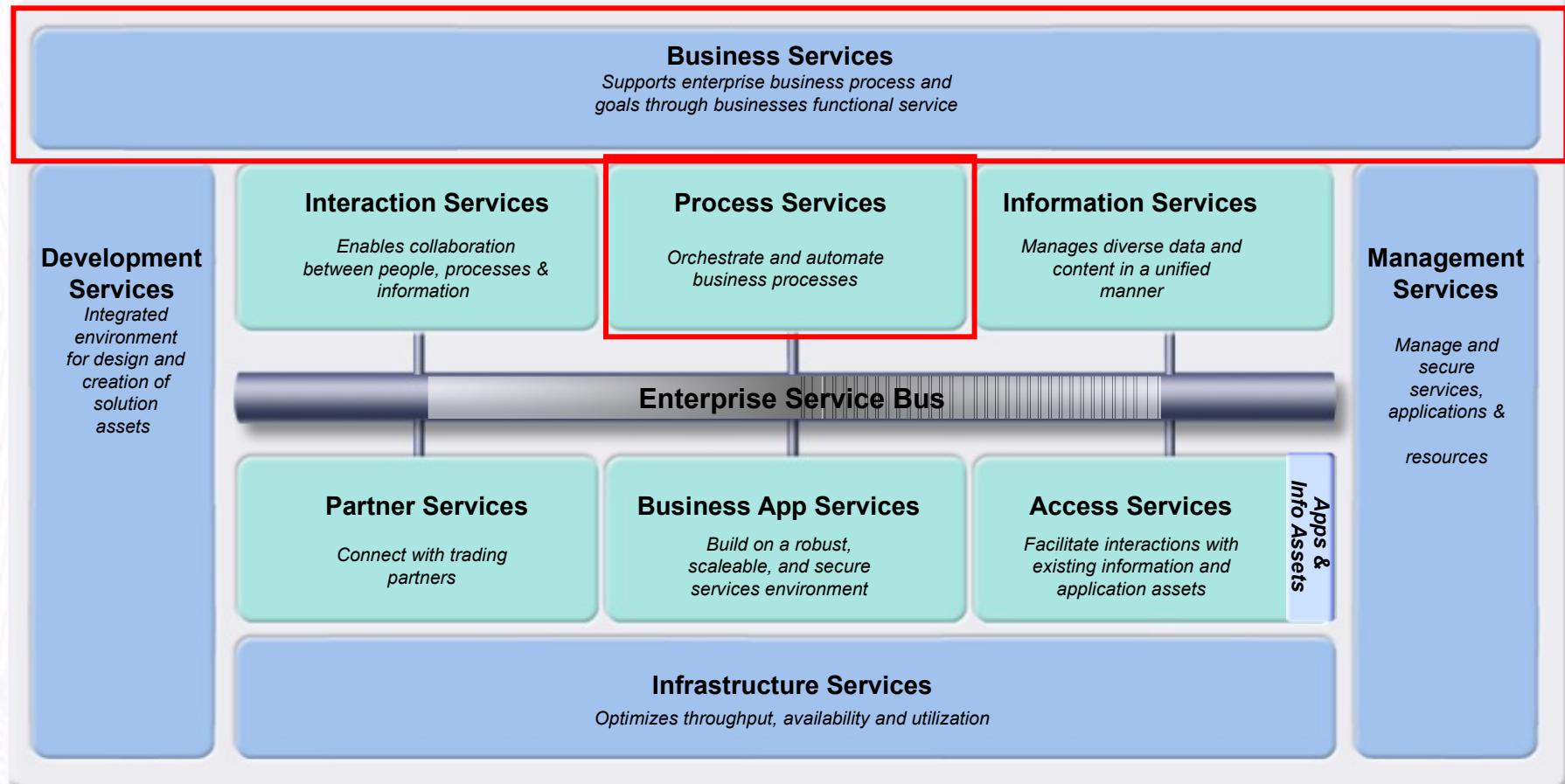
Defining the capabilities for your SOA environment

Template for designing at a service-level



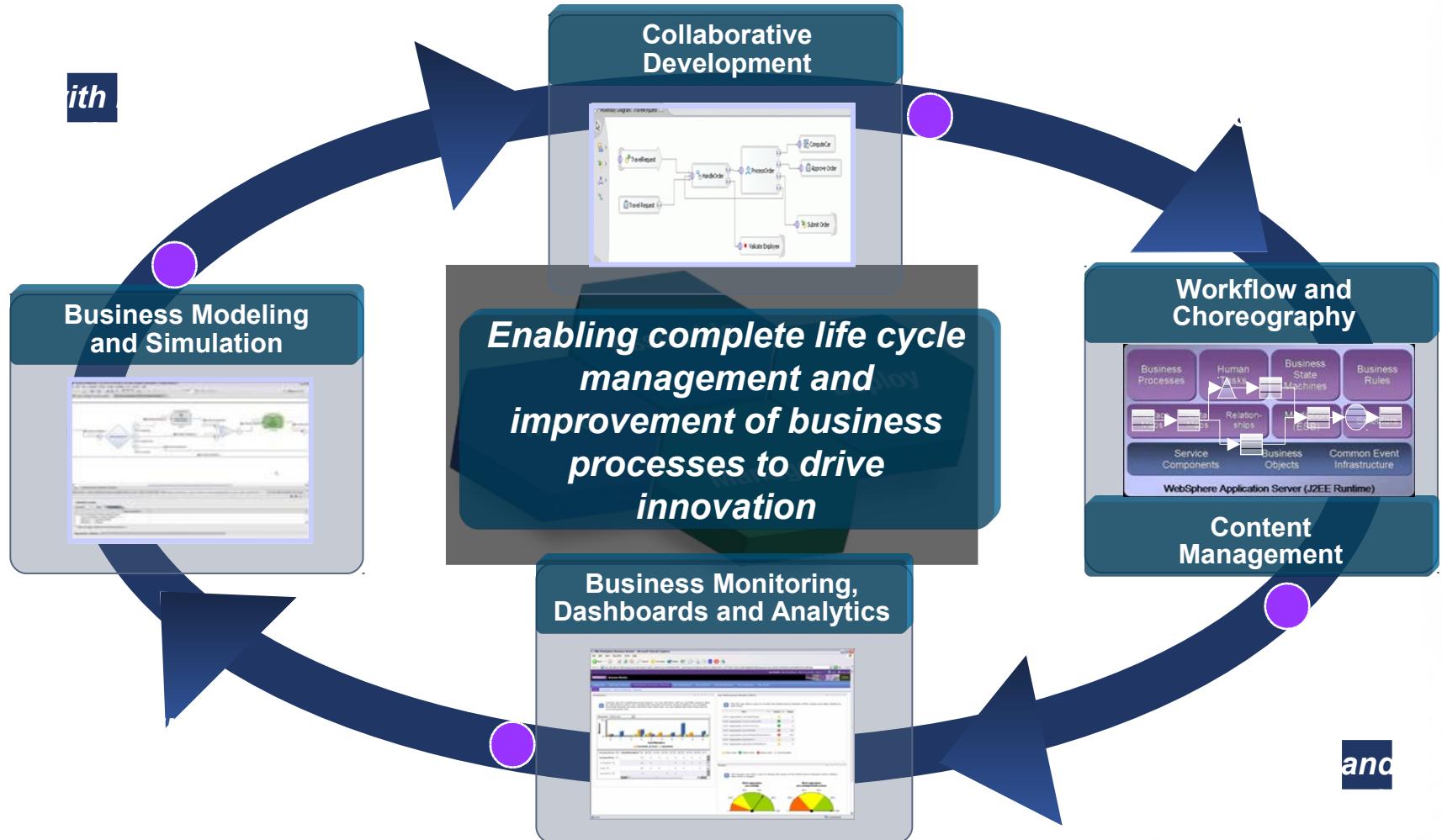
Defining the capabilities for your SOA environment

Template for designing at a service-level



Business Process Management capabilities in a SOA

Designed to Start Anywhere in the Cycle, Use Only What You Need



SOA delivers real customer value

by modeling and analyzing existing or new processes.

Respond Quickly
with processes based on a flexible infrastructure.

Ensure Compliance
by identifying problems before they get out of control.

by analyzing activity to ensure processes meet objectives.

IBM's customer order and analysis process *reduced time and cost of new process improvement releases by 25%*

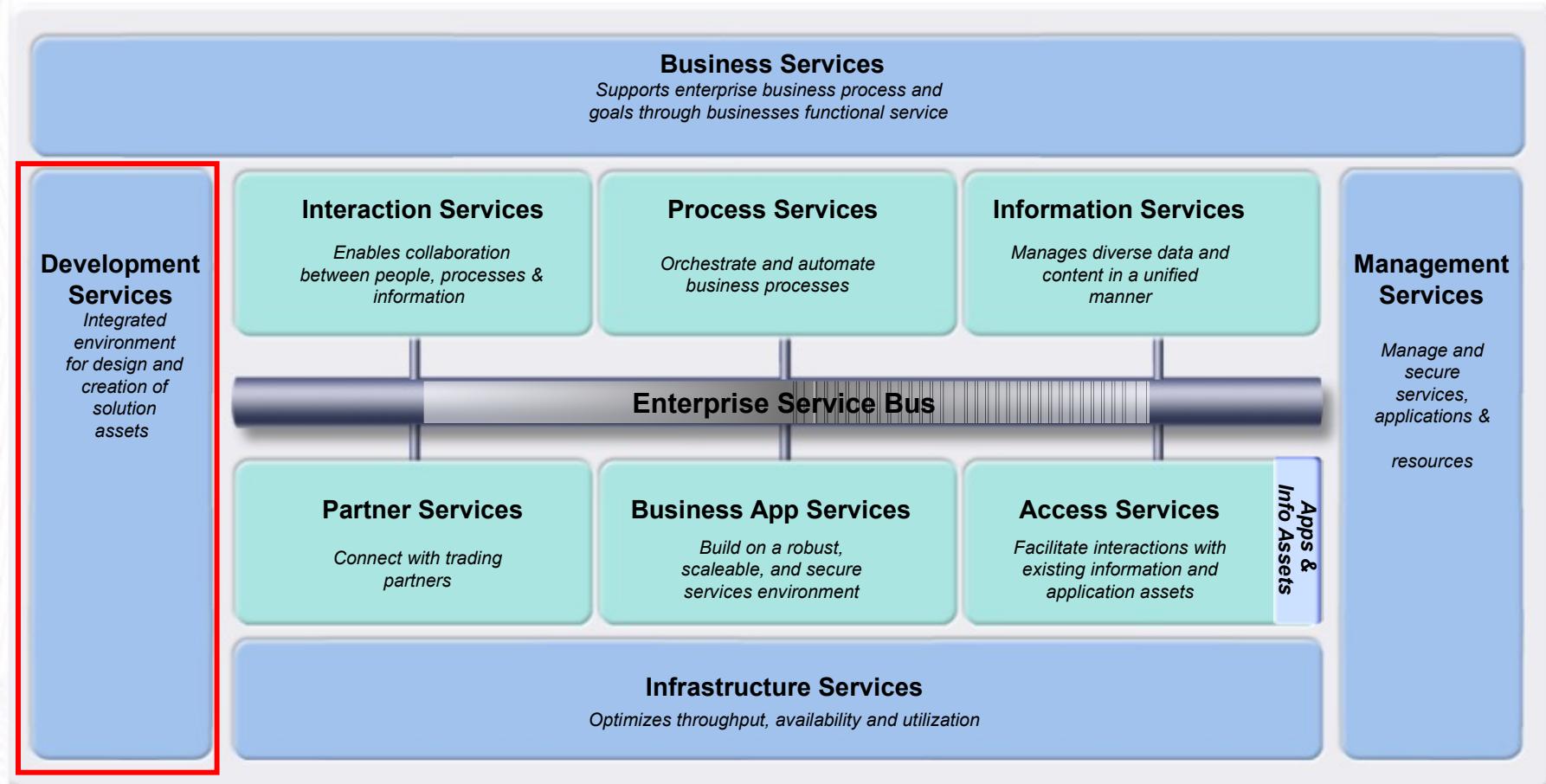
Danske bank realized *\$2M savings* in the first year through company wide increase in productivity

Telefonica to *triple its clients in two years* by being able to ensure changes made were executing correctly

Principal Financial reduced paper-intensive mortgage process time by *53%, achieved 34% gains in efficiency and estimated annual savings of \$4M*

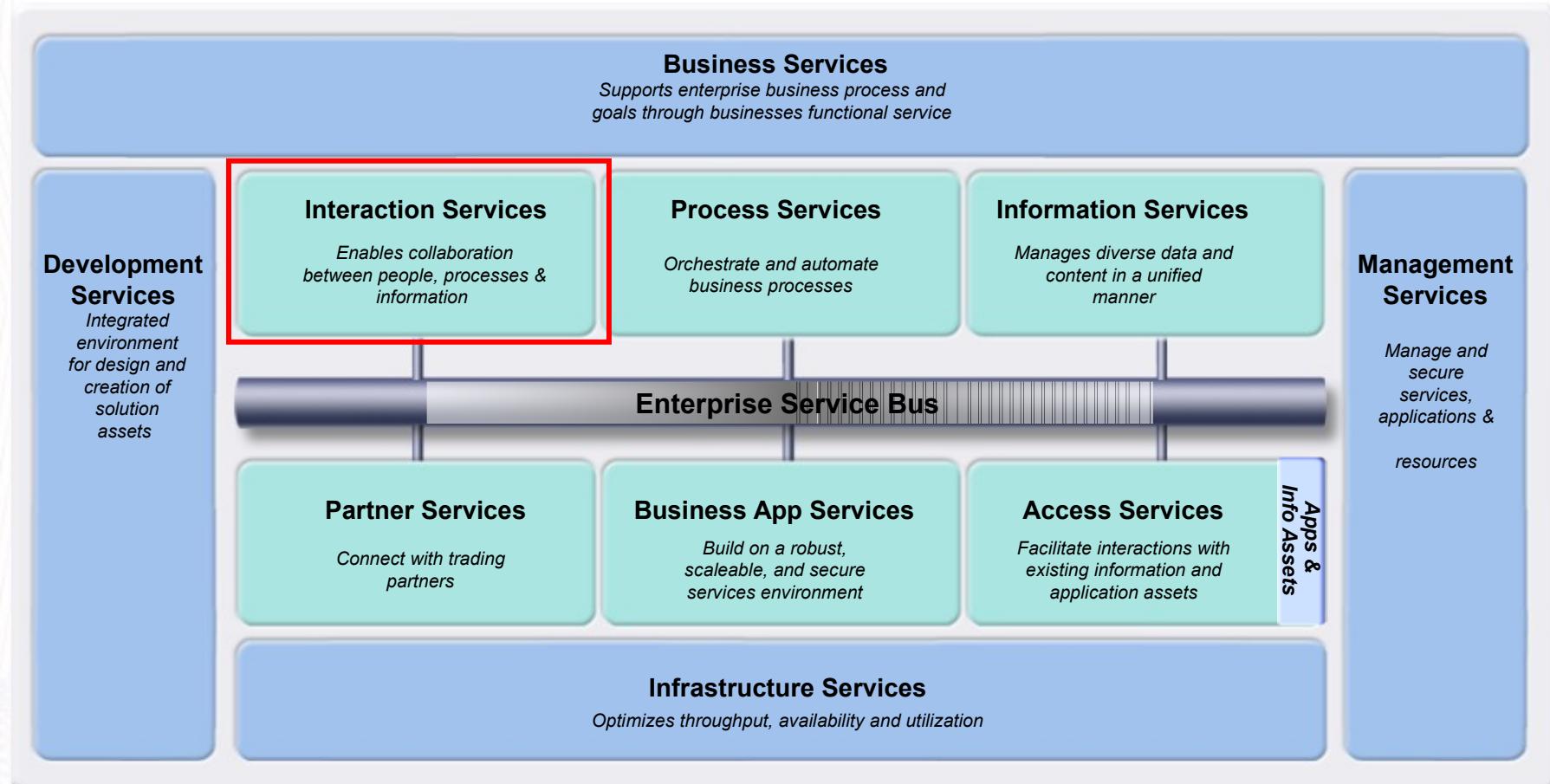
Defining the capabilities for your SOA environment

Template for designing at a service-level



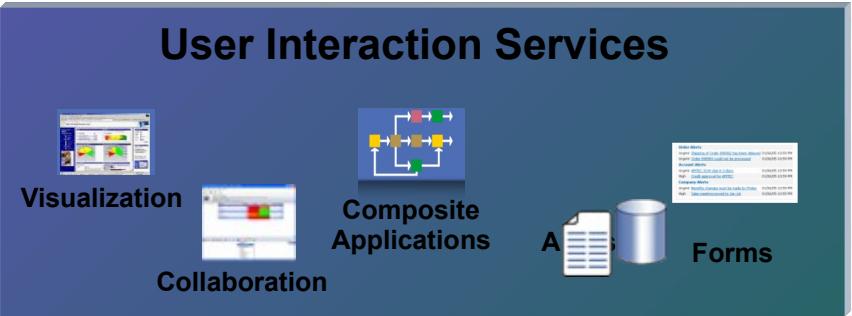
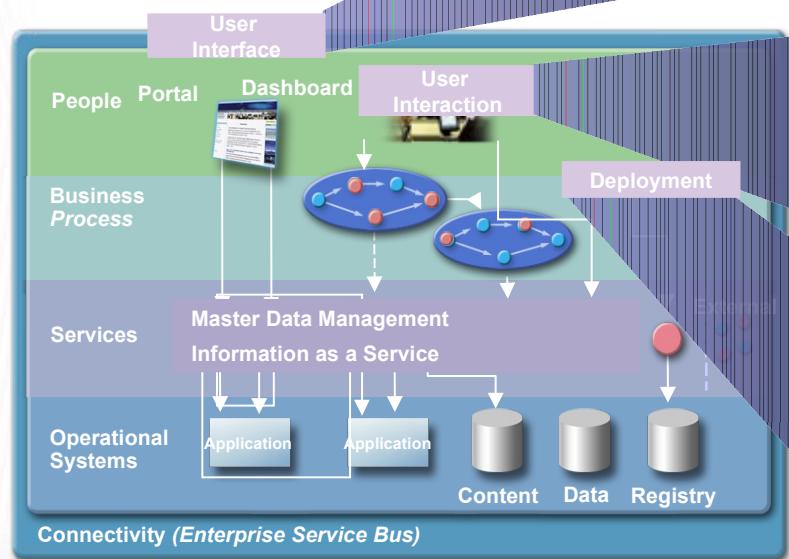
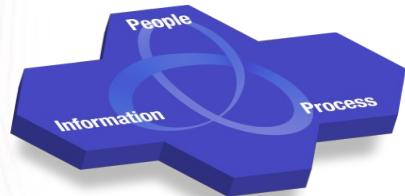
Defining the capabilities for your SOA environment

Template for designing at a service-level

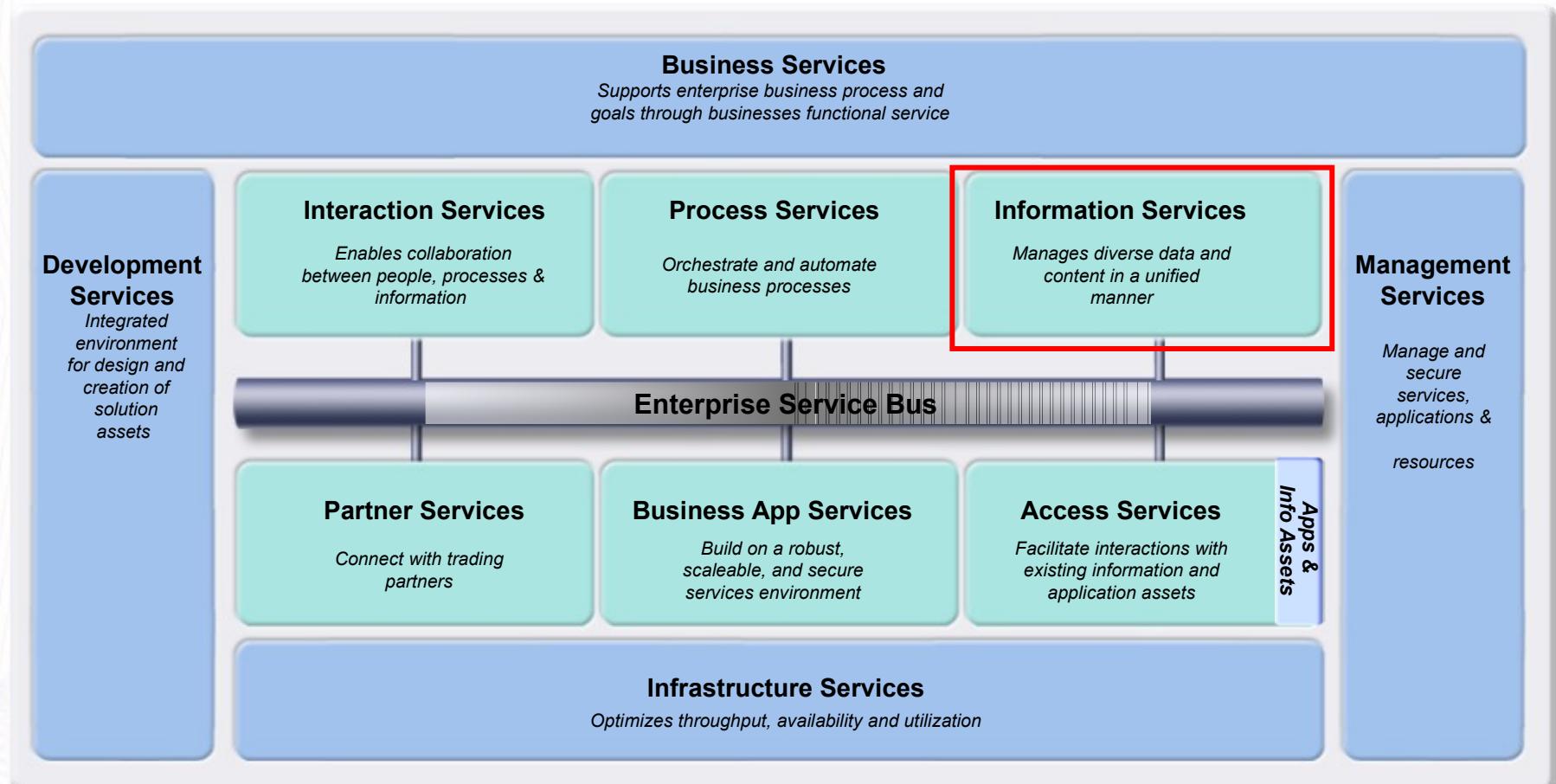


Interaction Services – Services for People

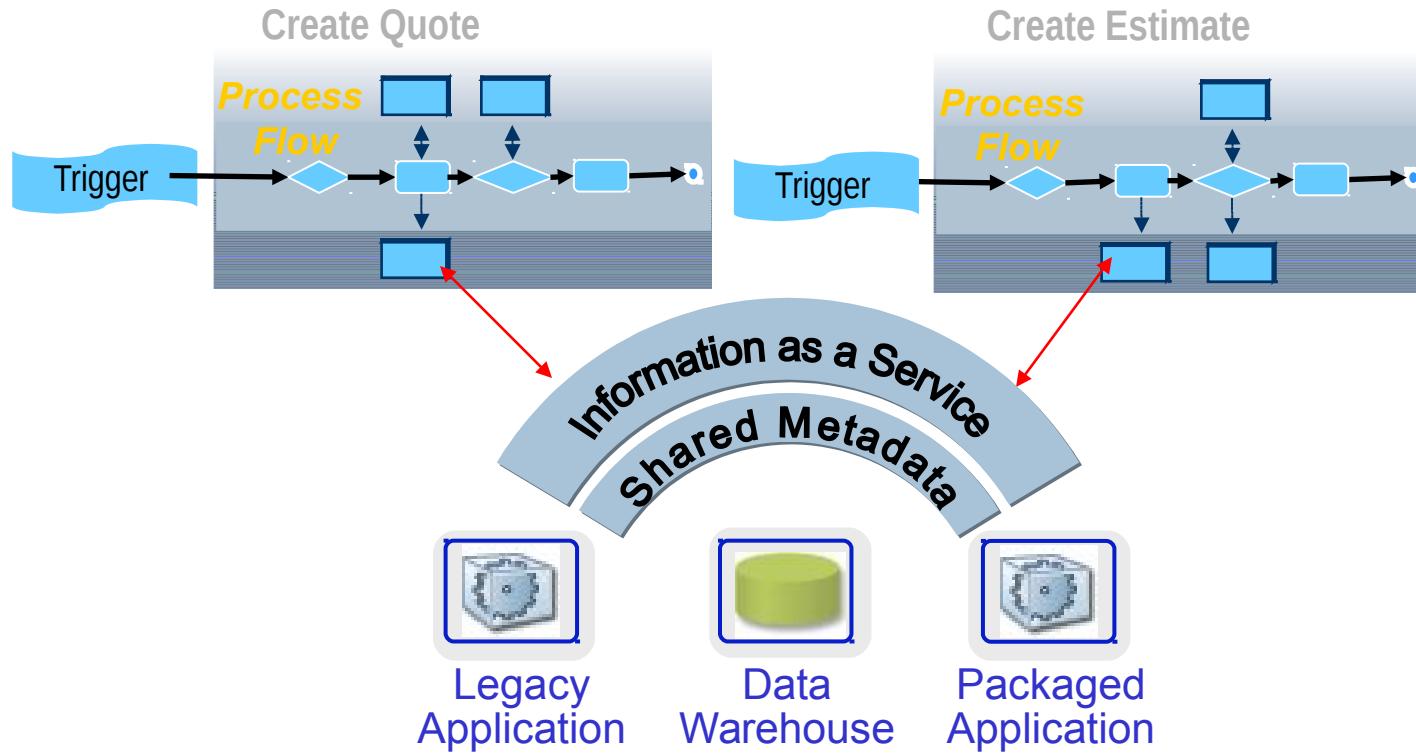
Making SOA Consumable



Defining the capabilities for your SOA environment



Information As A Service

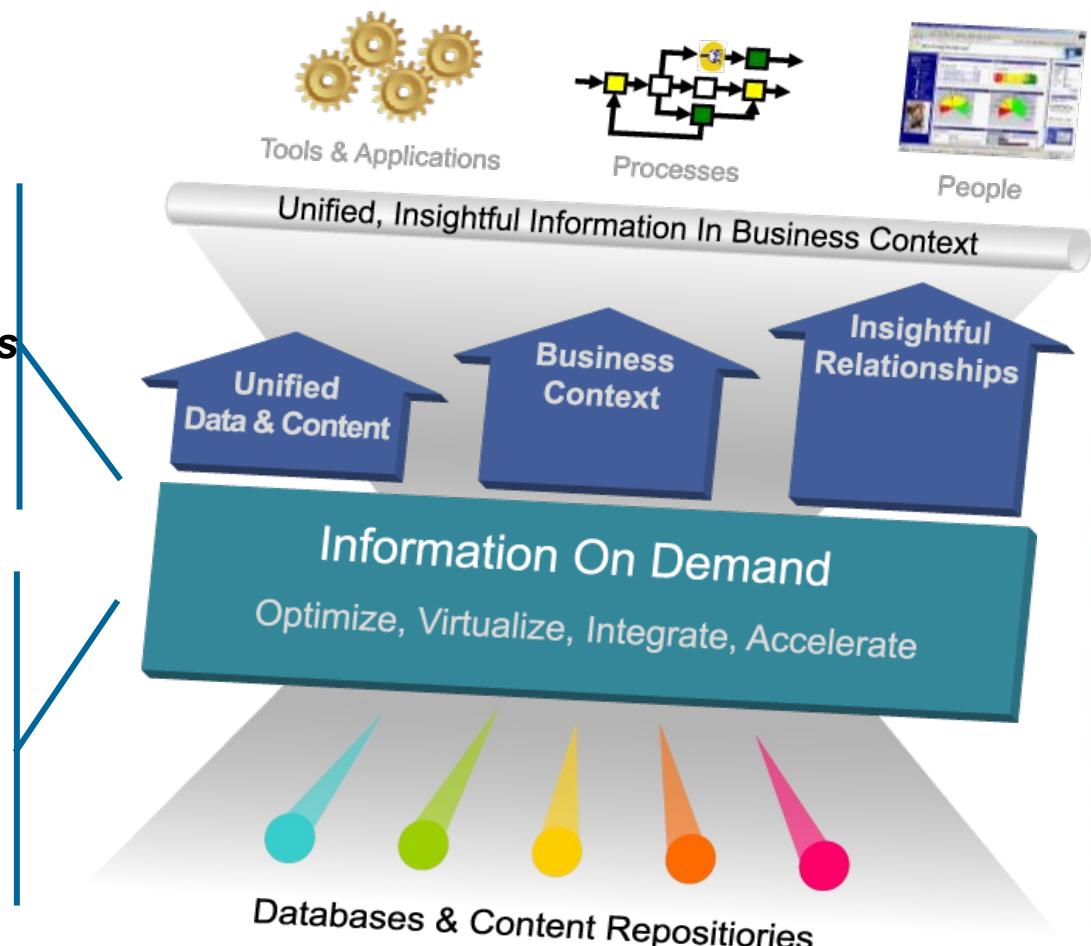


- *Consistent packaging of data*
- *Leverages understanding of metadata relationships*
- *Applies consistent rules to data*
- *Centralized control and maintenance*
- *Flexibility to change information sources and formats*

Information Services

Unparalleled Breadth and Depth

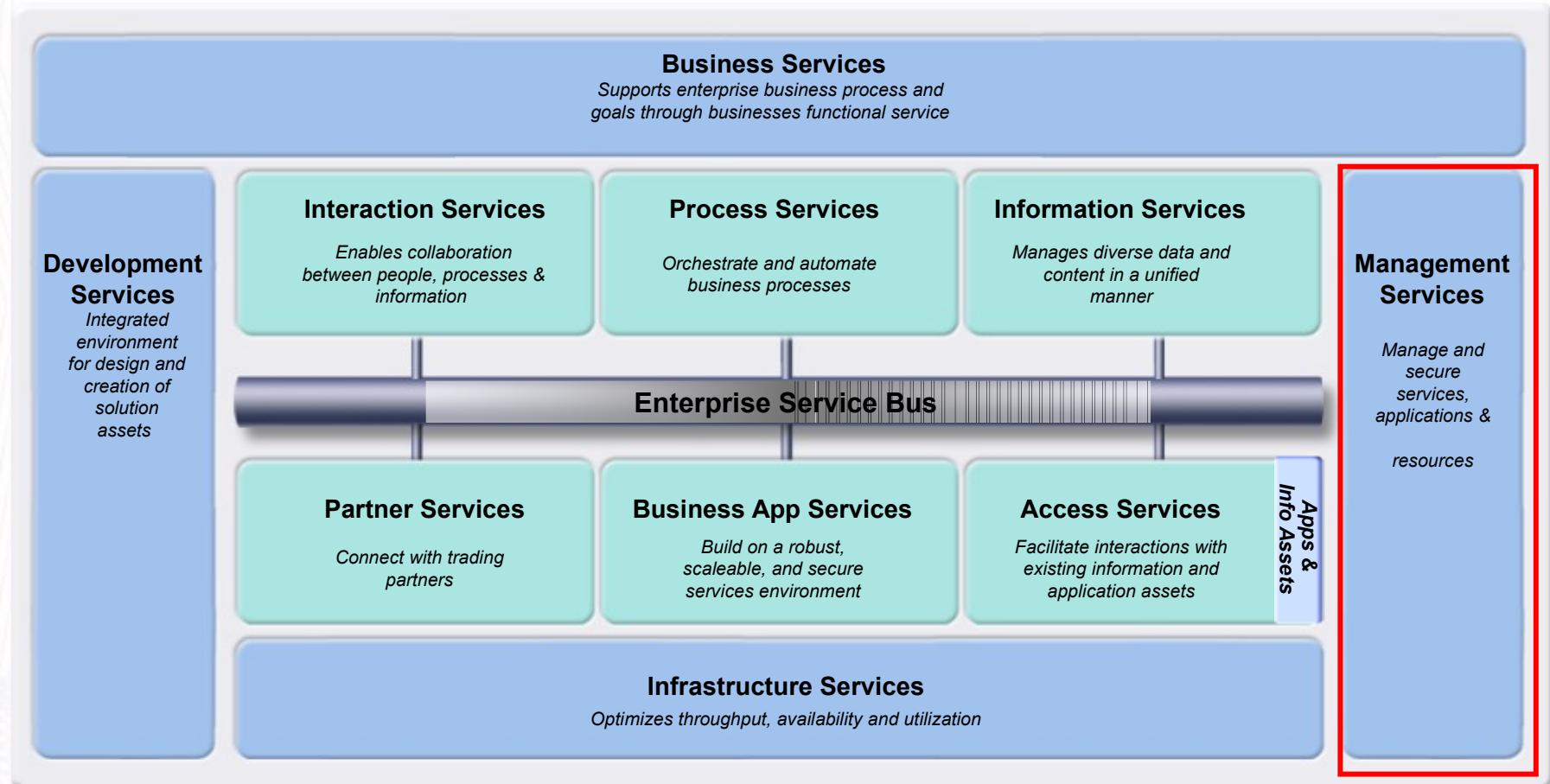
- - Master Data Management
 - Entity Analytics
 - Information Warehousing
 - Industry Models...
- **Information Integration Services**
 - Quality Services
 - Transformation Services
 - Federation Services
 - Metadata Services...
- **Content & Discovery Services**
 - Content Mgmt. & Integration
 - Discovery Services...
- **Data Services**
 - Databases, Warehouses, Tools...



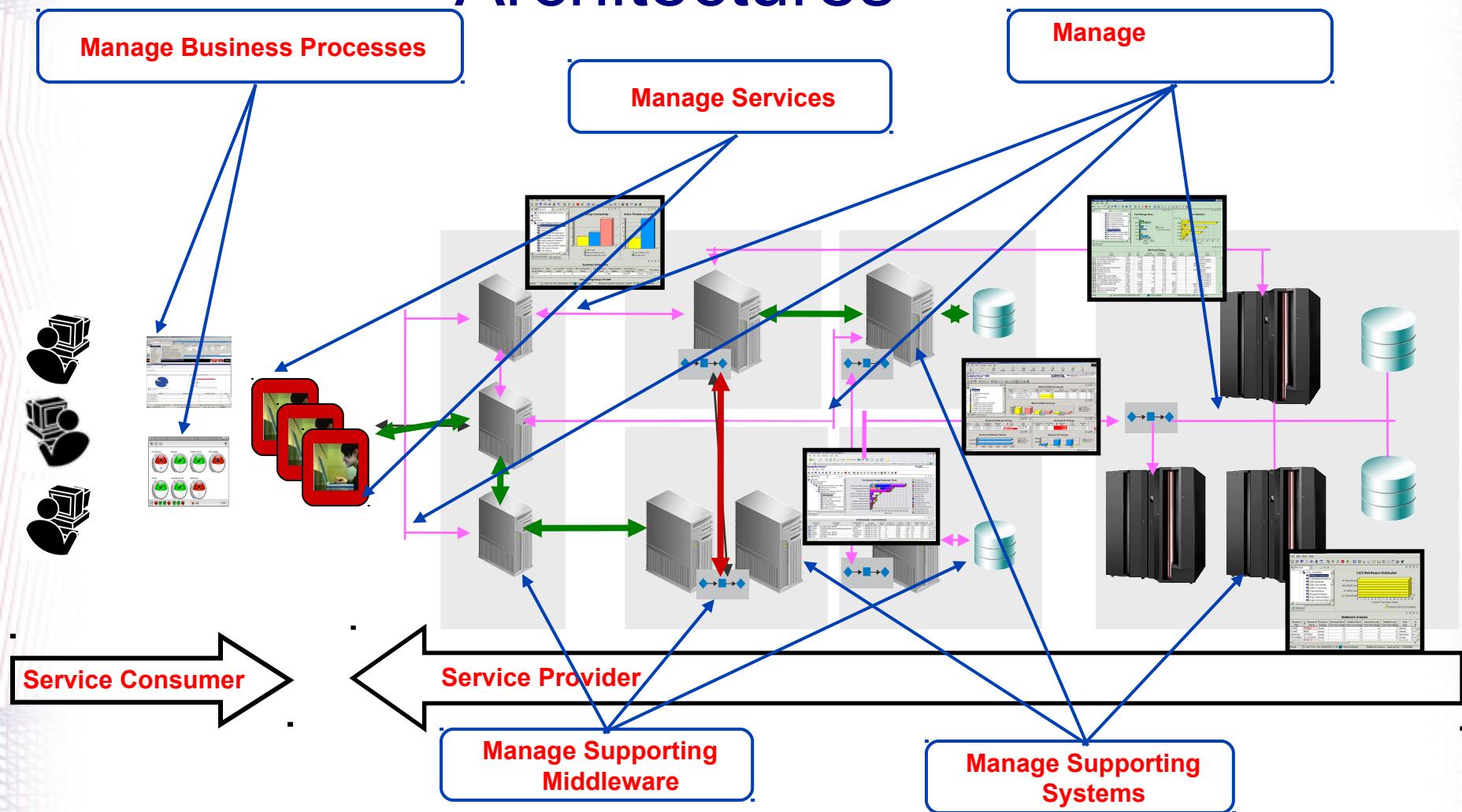
Source: Industry Publications & IBM Estimates

Defining the capabilities for your SOA environment

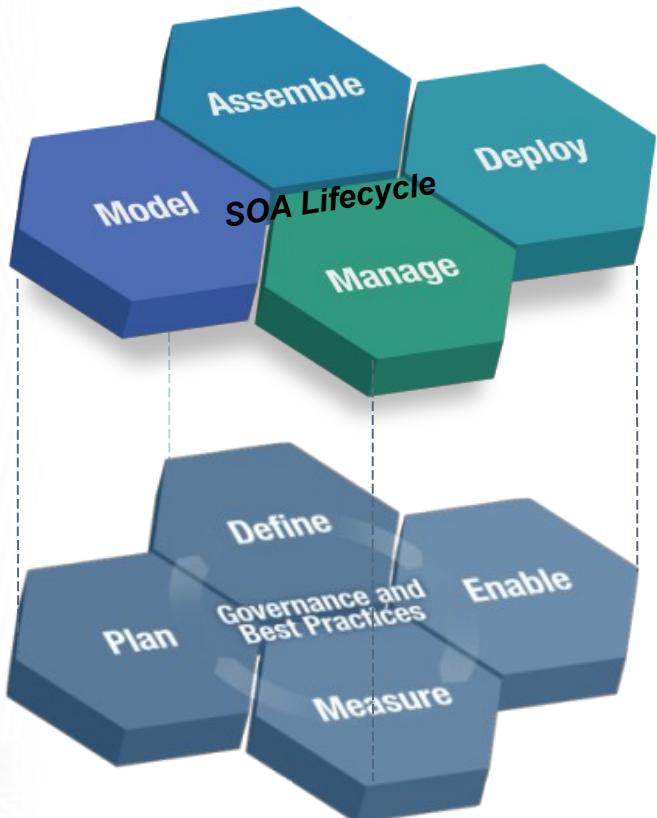
Template for designing at a service-level



Managing Service Oriented Architectures



SOA Governance is tightly associated with Service Lifecycle Management



Service Development and Delivery Management

Enforce, execute, automate process and policies

Infrastructure and Management In Support of SOA

Monitor and control operational policies

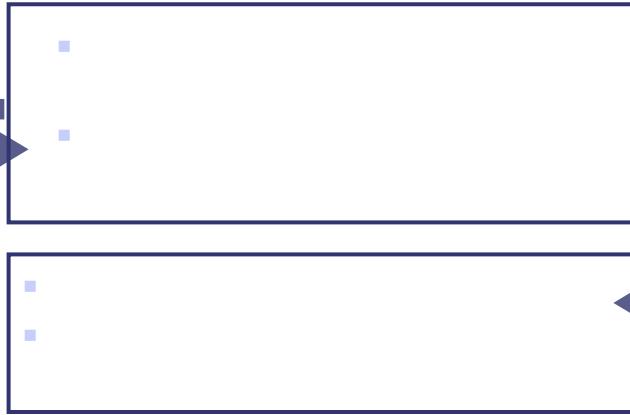
SOA Governance Method
Establish Process and Policies

Service Registry and Repository completes the Service Lifecycle Management

New!



Service Development and Delivery Management
Enforce Process and Policies

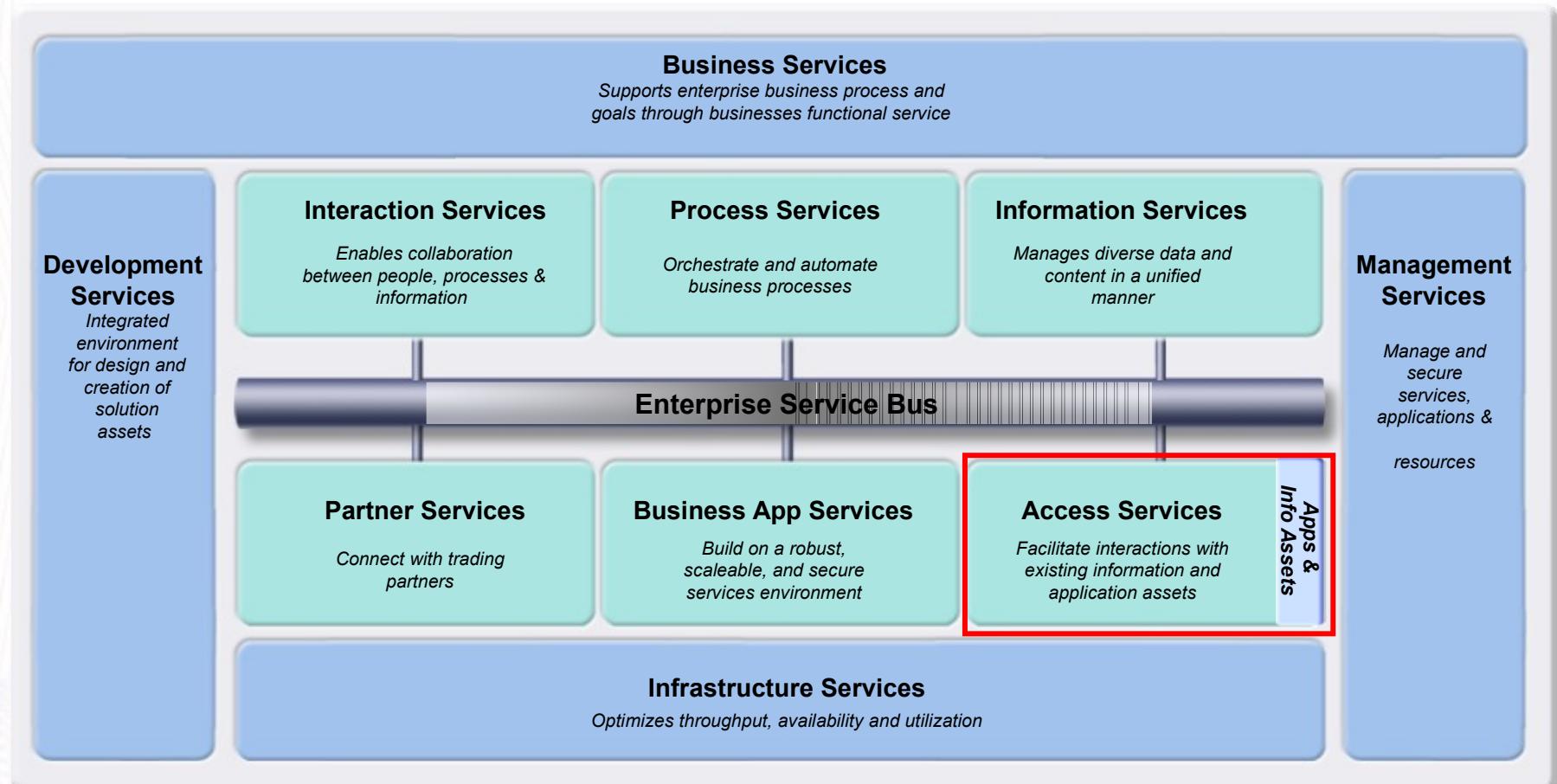


Infrastructure and Management In Support of SOA

Monitor Operational Policies

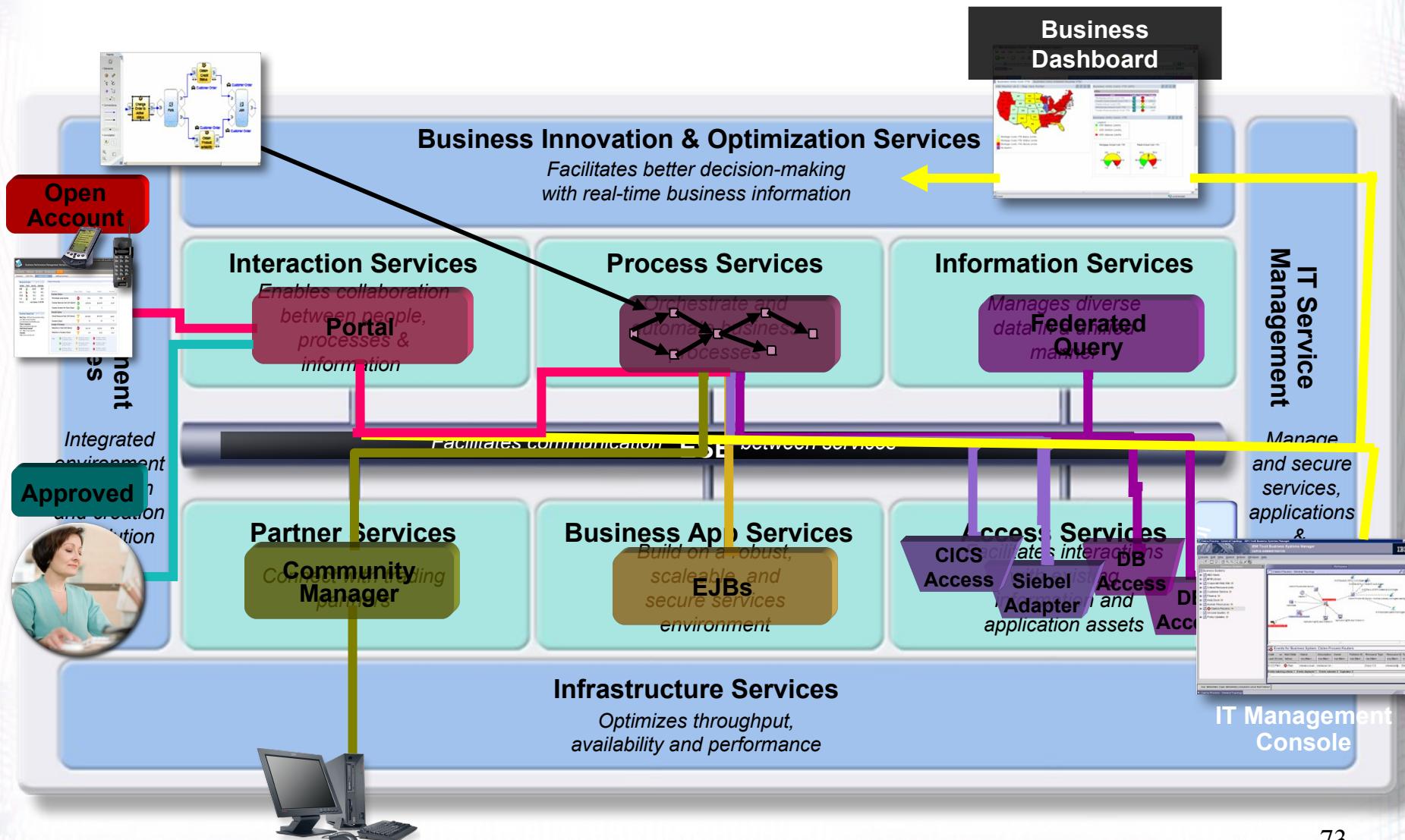
- Dynamic binding: ESB mediations query service metadata to retrieve candidate providers and policies
- System of record for service metadata by federating other registries

Defining the capabilities for your SOA environment



Separation of Concerns

The SOA Reference Architecture in Action



Deploying processes



Simple, flexible deployment of processes

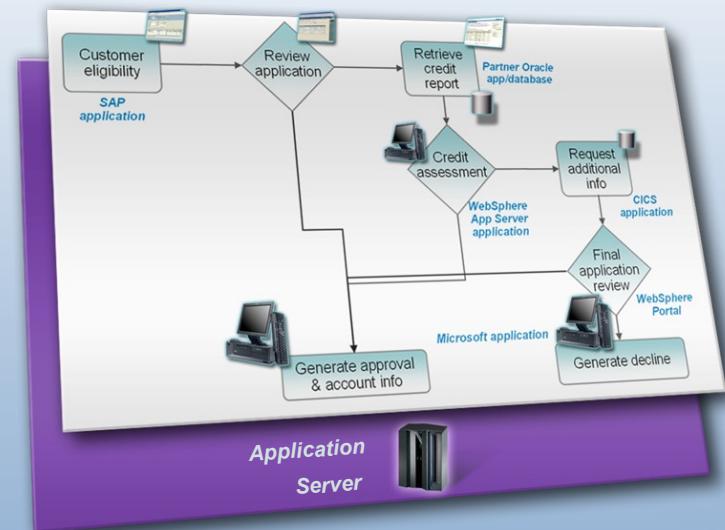
- Built and optimized on the market leading application server,

Powered by Enterprise Service Bus (ESB)

- Built on top of an open standards based ESB
- Flexible connectivity infrastructure for integrating applications, data, and services to power your SOA

Dynamically modify deployed processes

- Making plug-and-play of process components a reality
- Change business rules quickly and easily



Standards and open source support enhancing flexibility



Model

Modeling Essentials

- UML
- Eclipse



Assemble

Development Essentials

- SDO
- BPEL
- J2EE
- Eclipse



Deploy

SOA Runtime Essentials

- WS BPEL
- WS-I Basic Profile 1.1
- WS-I Simple Soap Binding Profile 1.0
- WS-I Basic Security Profile 1.0
- WS-Addressing
- WS-Reliable Messaging
- WS-Secure Conversations
- Other WS standards*
- J2EE

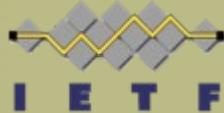


Manage

SOA Management Essentials

- Common Base Events

* includes a wide range of enterprise characteristics: security, transactions, reliable messaging



W3C



Java
Community
Process

