

# ESB BEST PRACTICES



## *3 INTEGRATION SCENARIOS*

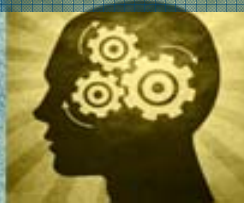
**DATA CONSISTENCY:** INDEPENDENT APPLICATIONS “GET THE FACTS STRAIGHT”.

**MULTI-STEP PROCESS:** INDEPENDENT APPLICATIONS IMPLEMENT A BUSINESS PROCESS

**COMPOSITE APPLICATION:** NEW FUNCTIONALITY BY TYING TOGETHER EXISTING APPLICATIONS.



# AGENDA



- INTRODUCTION TO SOA
- DEMYSTIFYING ESB
- IMPLEMENTATION METHODOLOGY OF ESB
  - STEP I – REQUIREMENTS GATHERING
  - STEP II – IDENTIFYING “MUST HAVE” FEATURES
  - STEP III – COMPONENTIZING BUSINESS PROCESS
  - STEP IV – DEFINING DISTRIBUTED ESB PROCESS
  - STEP V – DEPLOYING ESB PROCESS
  - STEP VI – LAUNCHING AND MONITORING ESB PROCESS
  - STEP VII – CHANGE MANAGEMENT, VERSION ROLLOVER
- INTEGRATION SCENARIOS – PUTTING THEORY TO PRACTICE

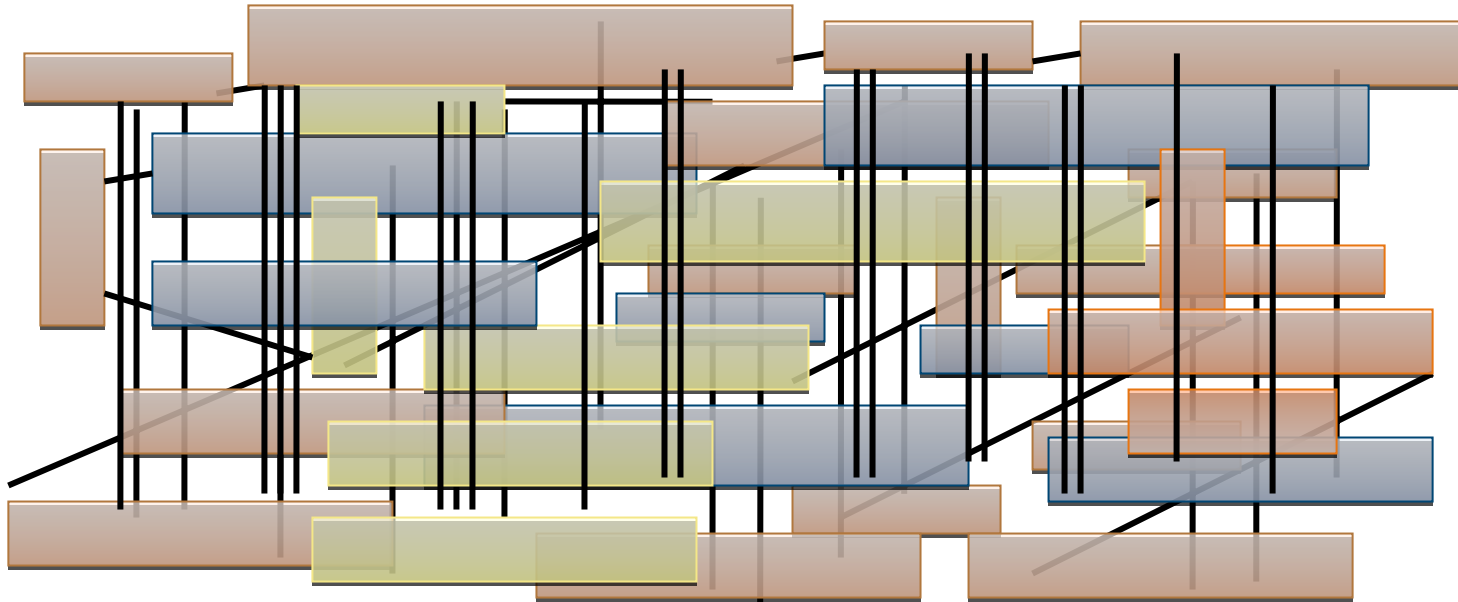


# INTRODUCTION TO SOA



# INTRODUCTION TO SOA

## POINT TO POINT INTEGRATION



- “Accidental” Architecture (Synchronous, Fine-grained, Not scalable, Many connections and data formats, Extremely difficult to manage, Expensive to maintain and extend)
- Hinders business change: new products, channels, suppliers, etc.
- Slow and expensive maintenance, No Reusability
- Expensive to Manage, Monitor and Extend
- => Need realistic way to evolve new infrastructure and add value

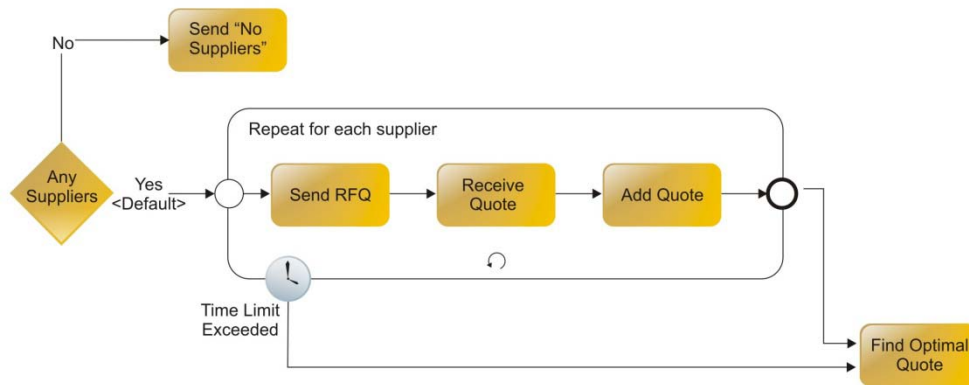


# BUSINESS PROCESSES BECOME AGILE WHEN IMPLEMENTED AS SERVICES



## Business Perspectives

- Simplification
- Elimination
- People
- Process Performance Measurement
- Simulation modeling
- Agility
- etc.

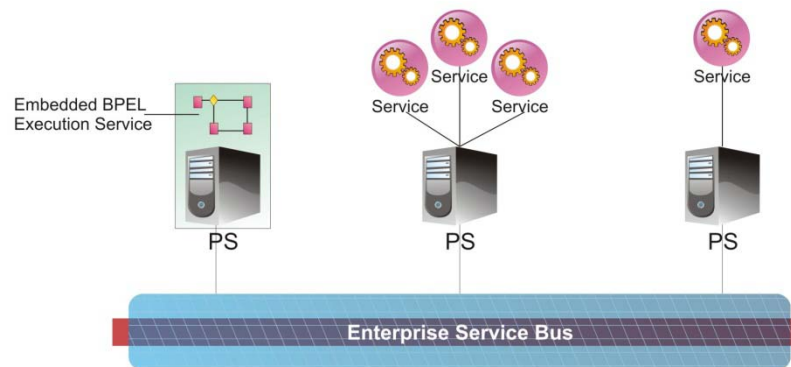


BPMN Notation representing a business process with an expanded Sub-process



## Systems Perspectives

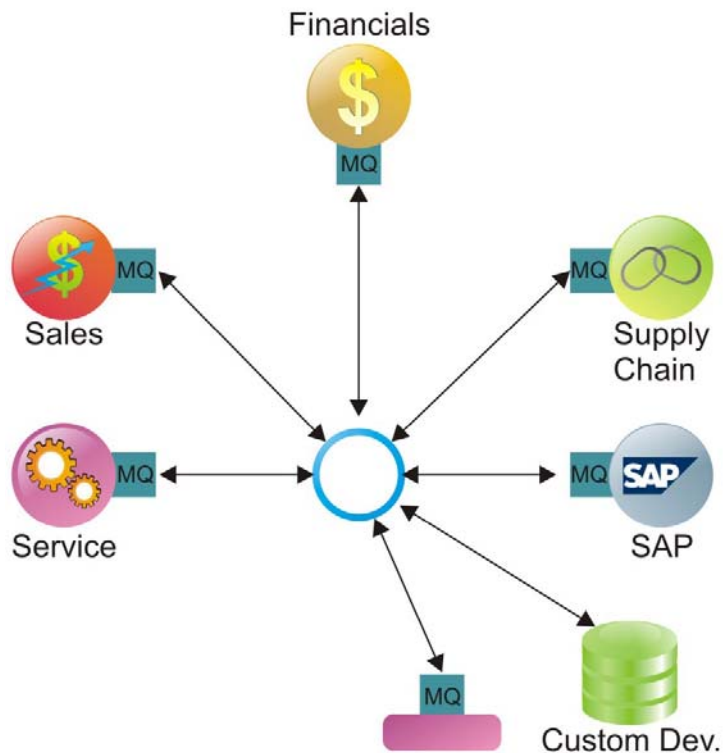
- Working Software
- Reliability
- Configuration Management
- Scalability / Performance
- Reusability
- Speed of Delivery
- etc.



Example: Processes implemented as Services

With SOA – Processes Can Be Layered on Top of Existing Infrastructure

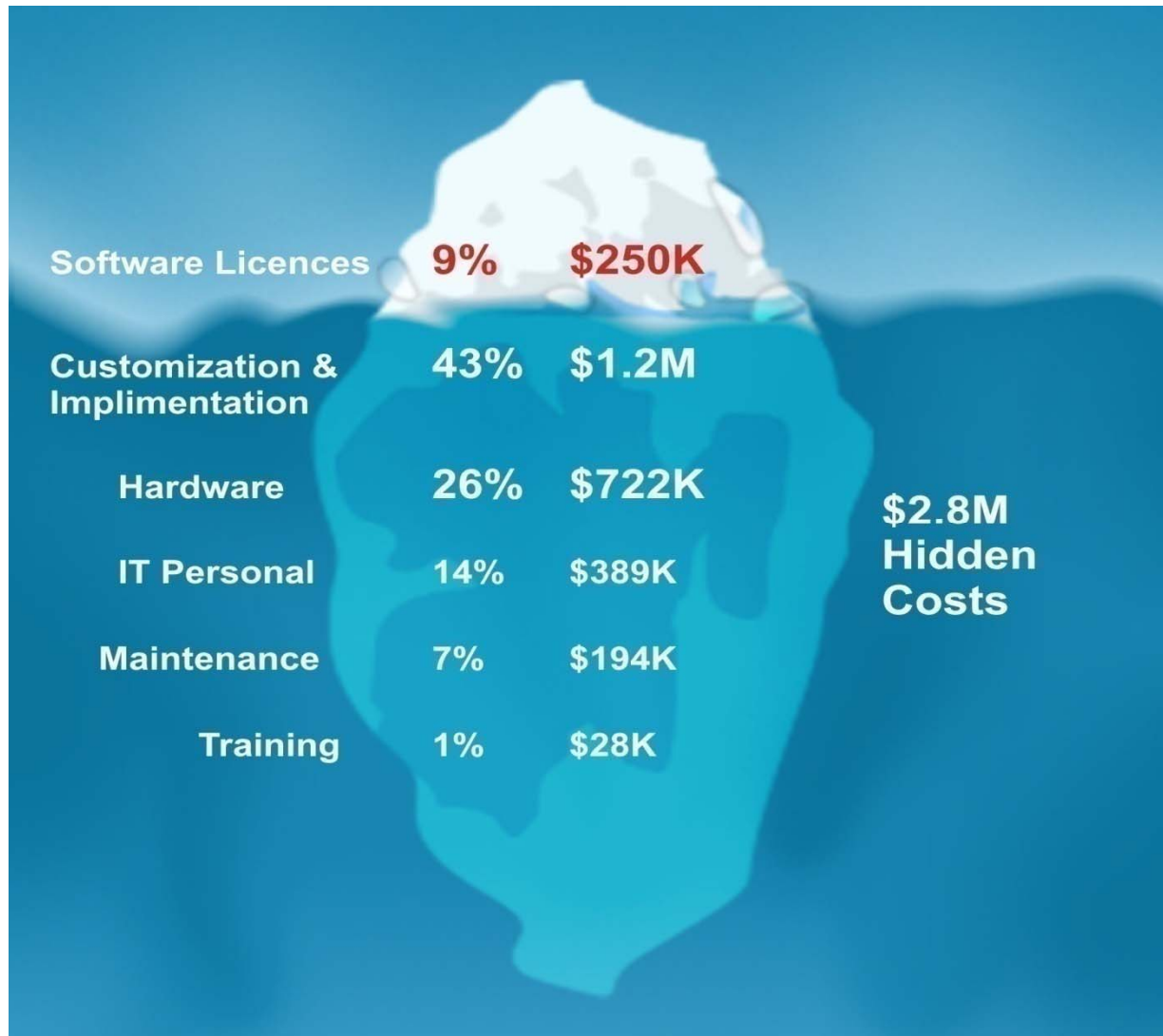
# INTRODUCTION TO SOA – MOM CHALLENGES



- Easy Inter-Operability
- Easy Common Data Model for integration
- Robust Process Management
- Scalability and High-Availability
- Distributed services management – Real time configuration changes
- Reusability of Components & Services in an effective IDE
- Portable Adapters
- Cost, Speed, Quality & Control



# THE INTEGRATION BROKER ICEBERG



# INTRODUCTION TO SOA - WEB SERVICES CHALLENGES



- Suffer from incomplete and competing standards definition that might not ensure easy interoperability across different implementation
- Interoperability
  - ✓ SOAP
  - ✓ Only binding for HTTP; no bindings for SMTP, JMS
  - ✓ “SOAP encoding”: XML  $\leftrightarrow$  Objects
  - ✓ RosettaNet, EDIINT: no SOAP
  - ✓ SOAP 1.1  $\rightarrow$  1.2 (W3C)
- WSDL: RPC/Encoded vs. Document/Literal
- Do not offer a complete package but provide only the functionality of access and invocation leaving majority of the development work to the user
- Business Semantics - Orchestration & Choreography
- Security
  - ✓ HTTPS is not enough
  - ✓ No agreement regarding attachments
  - ✓ S/MIME in RosettaNet and EDIINT
  - ✓ Single Signon
- Transactional Integrity
- Reliable Asynchronous Message Handling
  - ✓ WS-Reliability (IBM, MS) vs. WS-ReliableMessaging (OASIS)
- Transformation Services



# DEMYSTIFYING ESB



# DEMYSTIFYING ESB

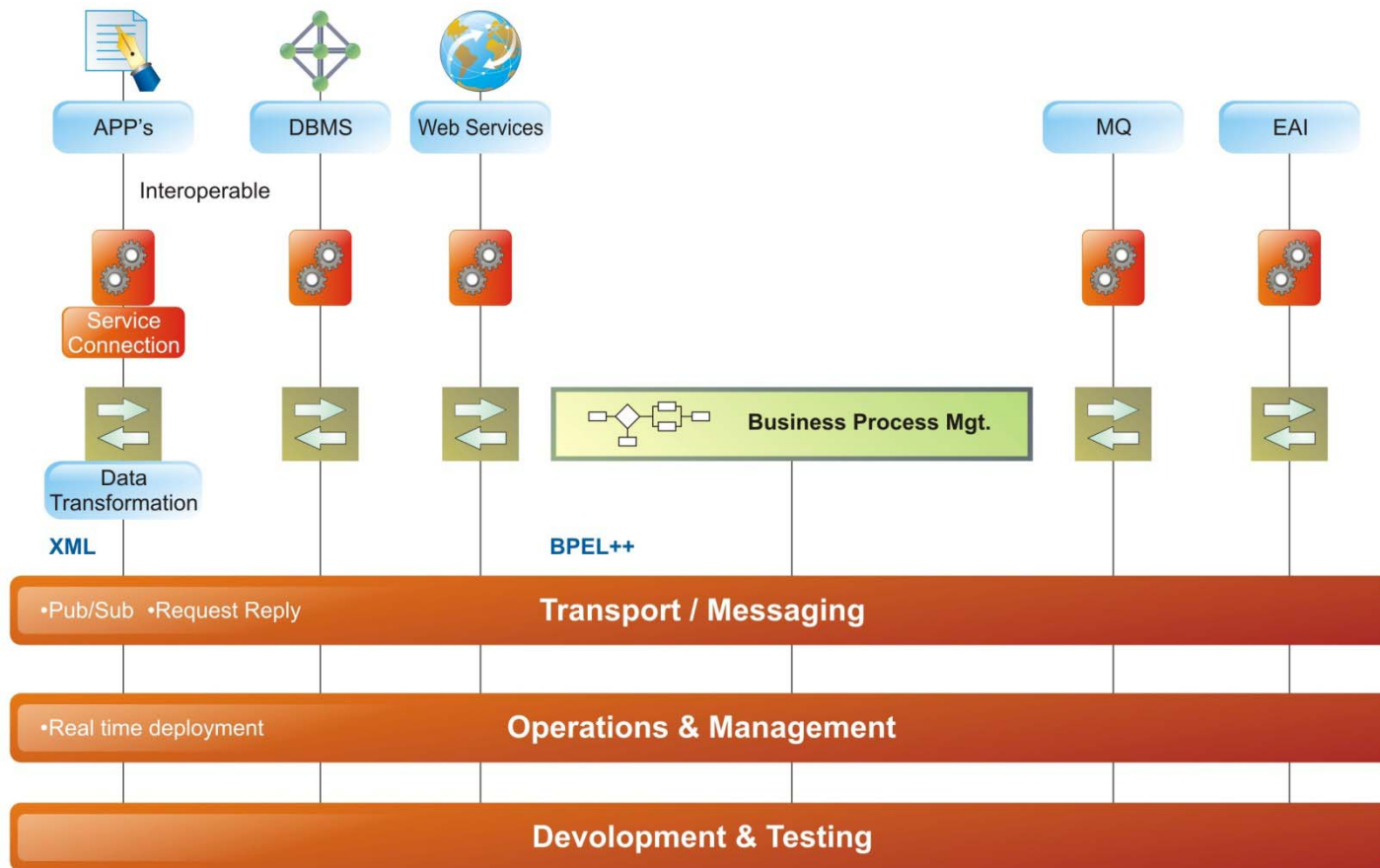


## Enterprise Service Bus evolved out of SOA

- IT Components can be accessed as services
- Defined form of invocation and entry points to service
- Business process - Event-driven/Asynchronous Invocation of Services
- Composite Application – mix of existing and new components



# FUNDAMENTALS OF SOA FOR INTEGRATION



Graphical tools to visualize integration



# DEMYSTIFYING ESB



## Definition (An Affordable EAI) ?

An ESB is a standards-based, service-oriented backbone capable of connecting hundreds of application endpoints. ESBs combine messaging, Web Services, XML, data transformation and management to reliably connect and coordinate application interaction. The ESB deployment model is an integrated network of collaborating service nodes, deployed in service containers.



# DEMYSTIFYING ESB



## Enterprise Service Bus – Standards based Integration

- Communication and data routing (JMS)
- Data protocols (XML)
- Transformation (XSLT)
- Content Based Routing (CBR)
- Connectivity (JCA)
- WebServices
- Security
- *Business Process Management (BPM)*
- *Pre-built Business Components*
- *Business Process Modelling (BPEL)*
- *B2B – trading partner management*



# WHY STANDARDS FOR INTEGRATION?

## BUSINESS AND TECHNOLOGY DRIVERS



- Increases ability to integrate
  - ✓ No platform or vendor technology dependencies
- Lowers cost of integration
  - ✓ Minimizes need for expensive proprietary adapters
  - ✓ Larger talent pool, broader education offerings
- Integration projects become more predictable



# STANDARDS-BASED INTEGRATION

## BACKBONE OF STANDARDS BASED INTEGRATION



- Deploying a Service-Oriented Architecture (SOA)
- Use of XML
- Rise of JMS as the de-facto for underlying communication mechanism
- BPEL for business process
- Connections to Packaged Apps, Legacy Systems using J2EE Connector Architecture (JCA)
- Web Services



# SERVICE-ORIENTED ARCHITECTURE (SOA)

## FLEXIBLE, EXTENSIBLE INTEGRATION



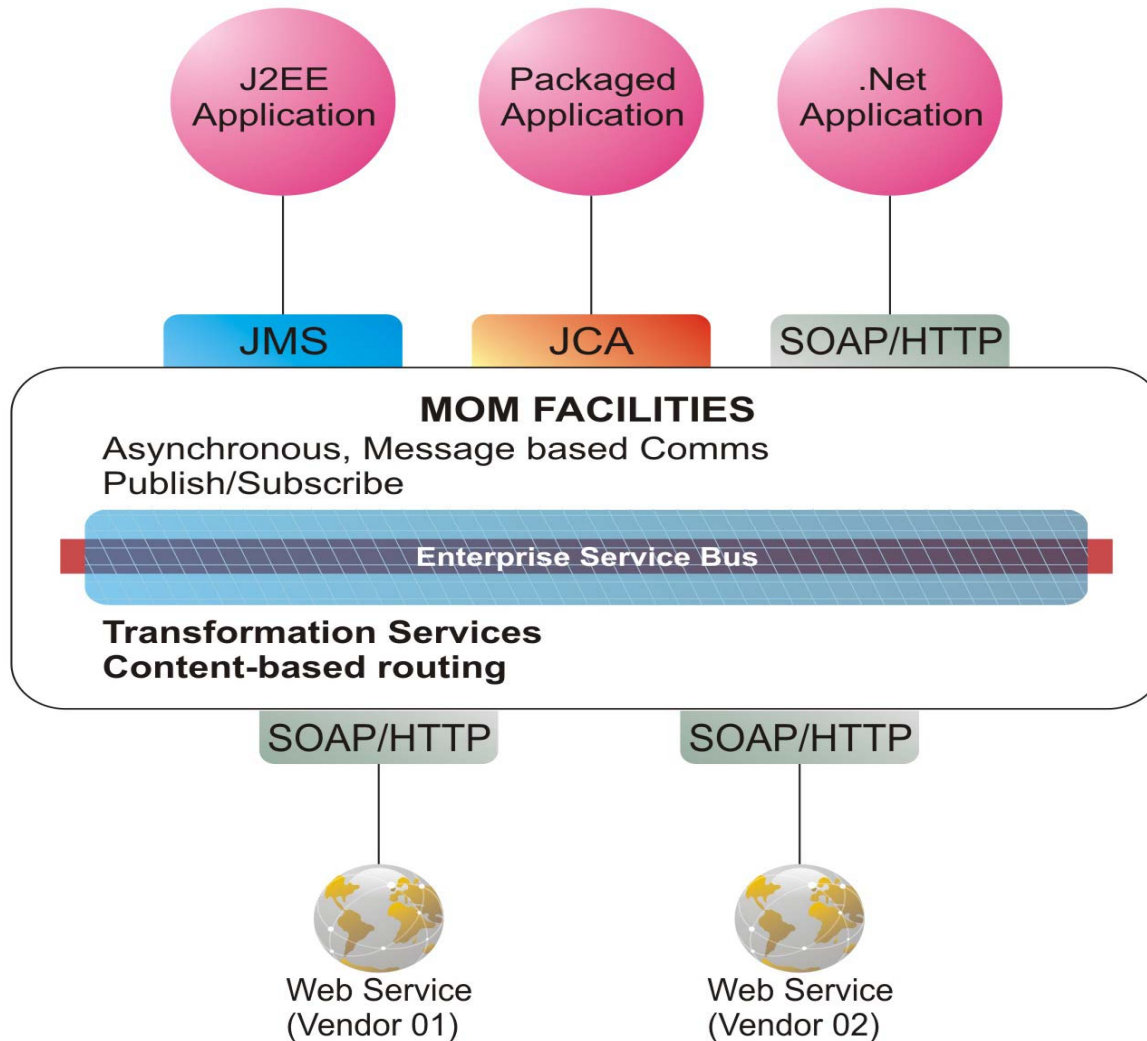
- Design methodology for distributed systems
- Applications expose functionality through service interfaces
- Loosely-Coupled
- Platform and language neutral
- Impervious to implementation changes
- Coarse-Grained- business level Interfaces
- Asynchronous
- No single points of failure



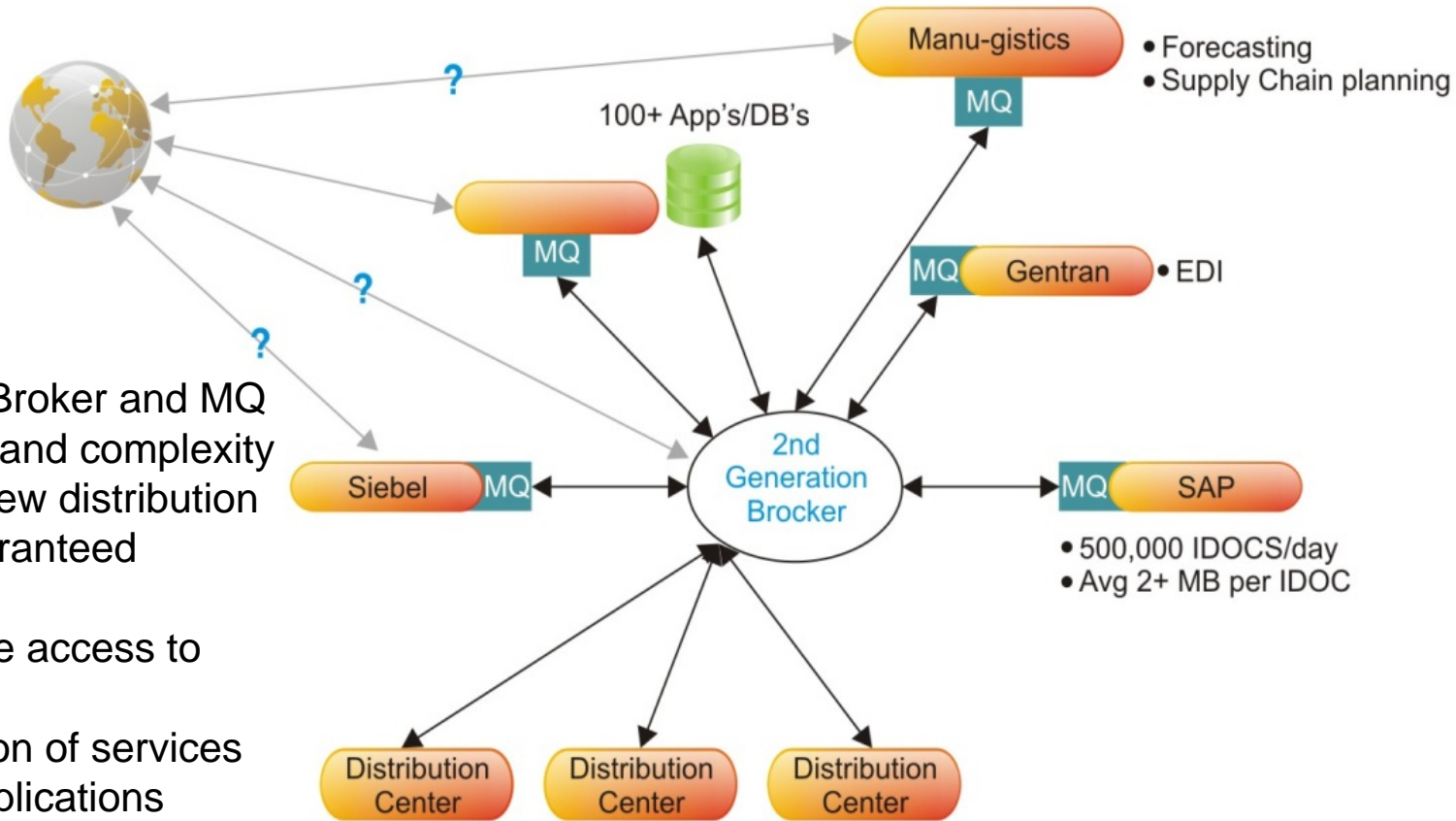
# DEMYSTIFYING ESB



# DEMYSTIFYING ESB



# EXAMPLE: BUILD ON EXISTING INFRASTRUCTURE

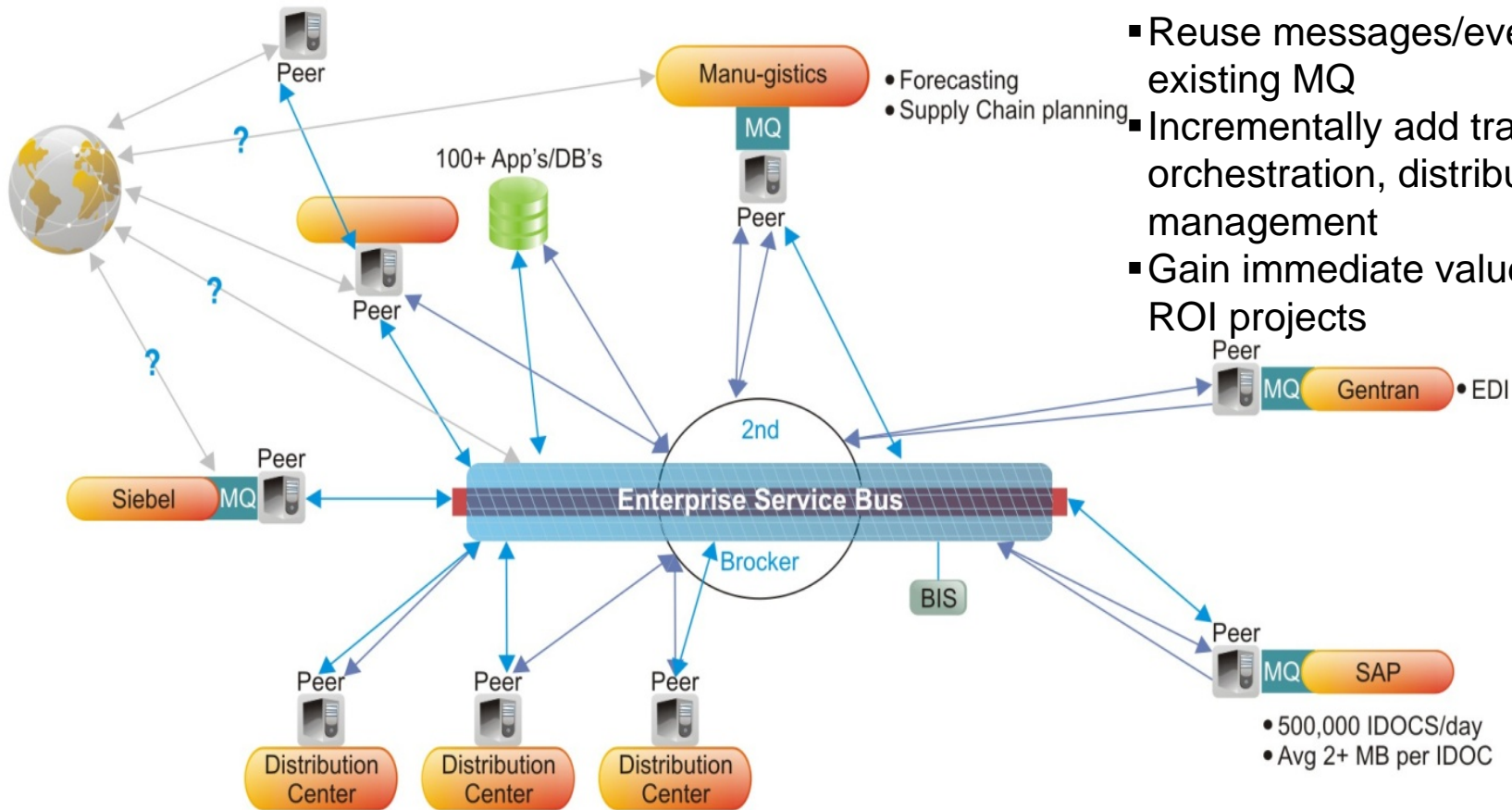


## Business Needs

- Want to replace Broker and MQ due to high TCO and complexity
- Reduce cost of new distribution centers, with guaranteed messaging
- Need web-service access to data
- Need orchestration of services
- Integrate new applications faster and cheaper



# EXAMPLE: BUILD ON EXISTING INFRASTRUCTURE



## Solution

- Services Oriented Integration
- Reuse messages/events with existing MQ
- Incrementally add transformation, orchestration, distributed management
- Gain immediate value from higher ROI projects



# DEMYSTIFYING ESB



## ESB's best suited for

- Projects that will mix heterogeneous application servers (for example, Microsoft or Java portals with disparate Java or Microsoft server back ends)
- Enterprises who want to start with a basic SOA and add other features later.
- Enterprises that want to assemble their own best-of-breed comprehensive integration suites
- Mix and match off-the-shelf adapters, BPM, B2B and BAM tools from other vendors.
- Distributed services (different programming languages) running on disparate nodes

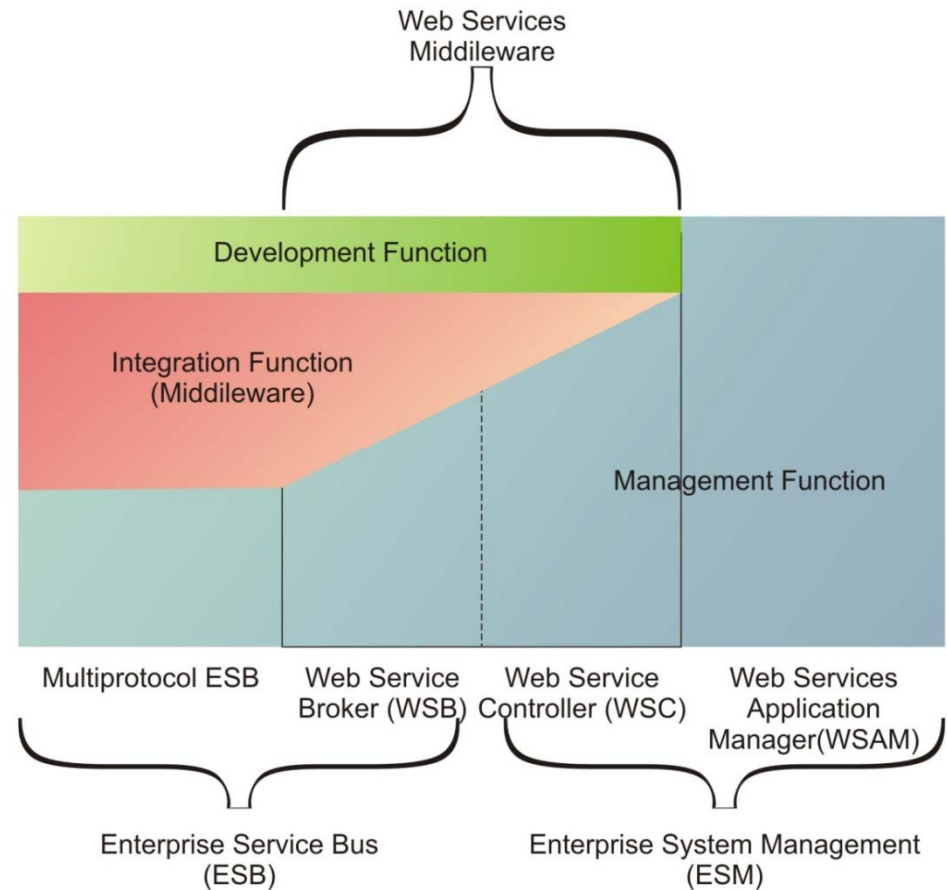


# DEMYSTIFYING ESB



## Types of ESB

1. ESBs based solely on SOAP -Web services brokers (WSBs)
2. Multiprotocol ESBs that support Web services and other communication mechanisms



Source: Gartner, Who's who in middleware, 1Q04



# ESB - VENDORS



Support SOAP/HTTP and additional protocols guaranteed delivery, publish-and-subscribe often following the JMS standard

- IBM
- Oracle
- Microsoft
- Software AG
- Progress Software
- TIBCO



# IMPLEMENTATION METHODOLOGY OF ESB

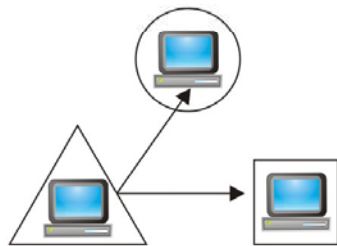


# 3 BASIC INTEGRATION PATTERNS



## DB SYNCHRONIZATION

### Data Consistency



- Multiple Processes
- Parallel unrelated steps
- One-Way, asynchronous
- Batch or Immediate
- Physically Independent
- Logically dependent

Source: Gartner Group

## ORDER FULFILLMENT APPLICATION

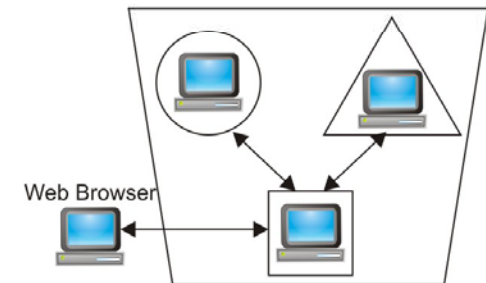
### Multistep Process



- One Business Process
- Multi Steps
- One-Way, asynchronous
- Batch or Immediate
- Physically Independent
- Logically dependent

## QUOTE TO BIND APPLICATION

### Composite Application

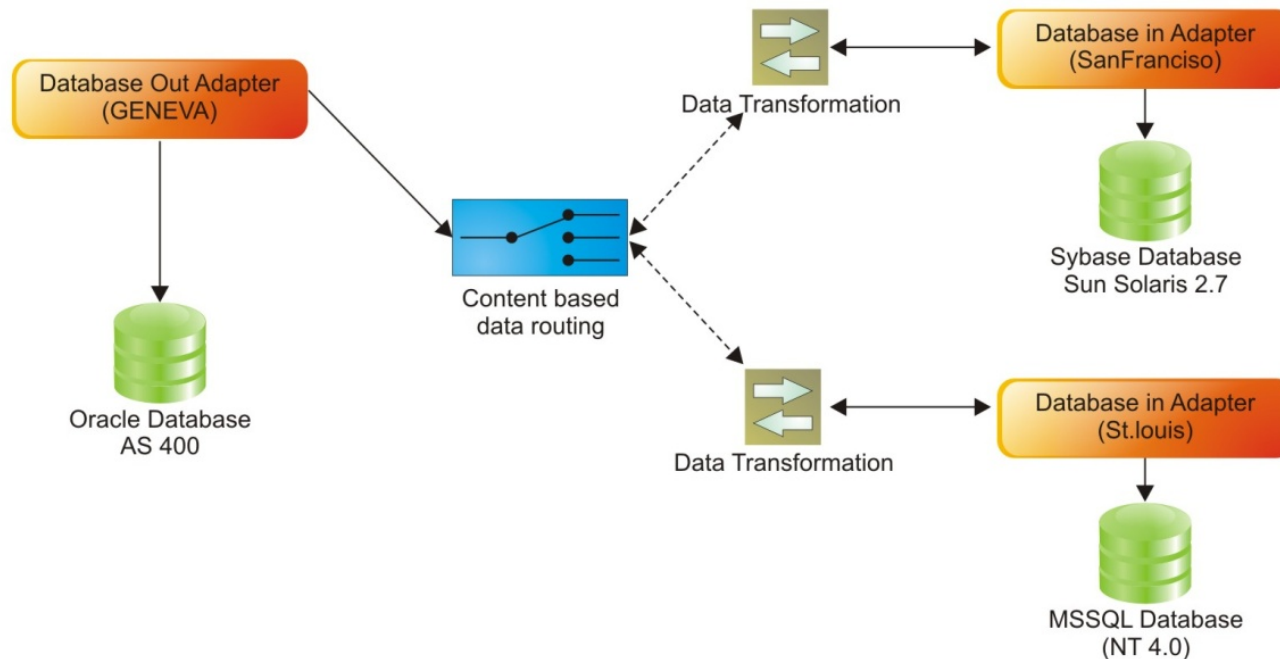


- One Business Process
- One Step
- Two-Way, synchronous
- Physically Independent
- Logically dependent



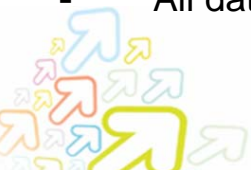
# STEP 1 – REQUIREMENTS GATHERING

## DATA CONSISTENCY – DATABASE SYNCHRONIZATION



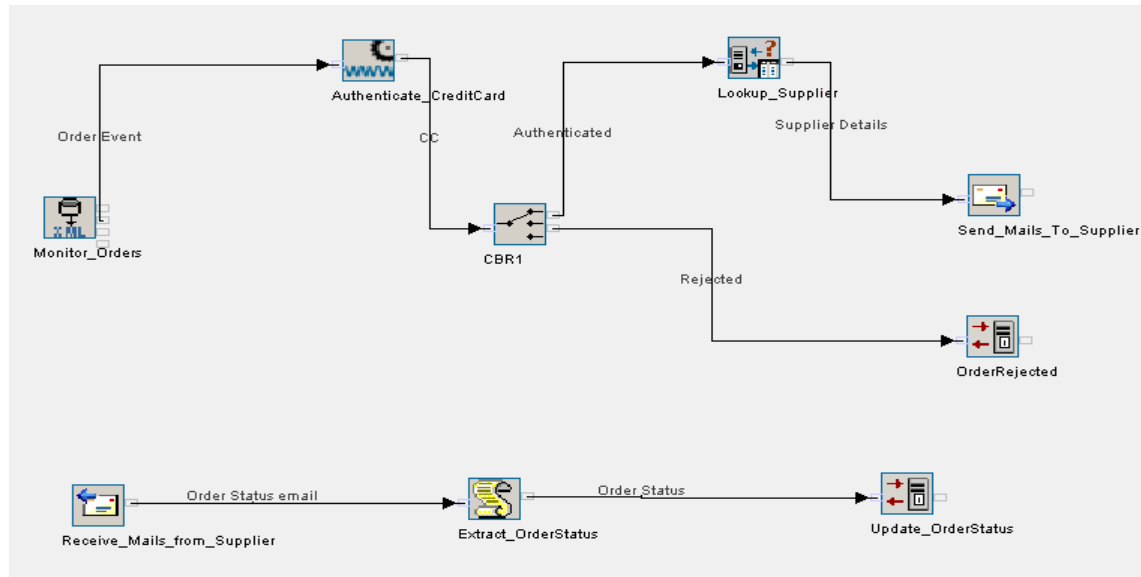
### The technical requirements of ABC Corporation can be summarized as

- Data needs to be replicated across multiple data centers in asynchronous fashion using a Message Bus.
- Any changes made to the Oracle Database instance in Geneva needs to be reflected (based on a selection criteria) in either the Sybase database instance or the MSSQL database instance located in San Francisco and St.Louis respectively.
- The data needs to be transformed from the source table data format to target table data format(s).
- All data transfers need to be secure



# STEP 1 – REQUIREMENTS GATHERING

## MULTI STEP PROCESS – ORDER FULFILLMENT

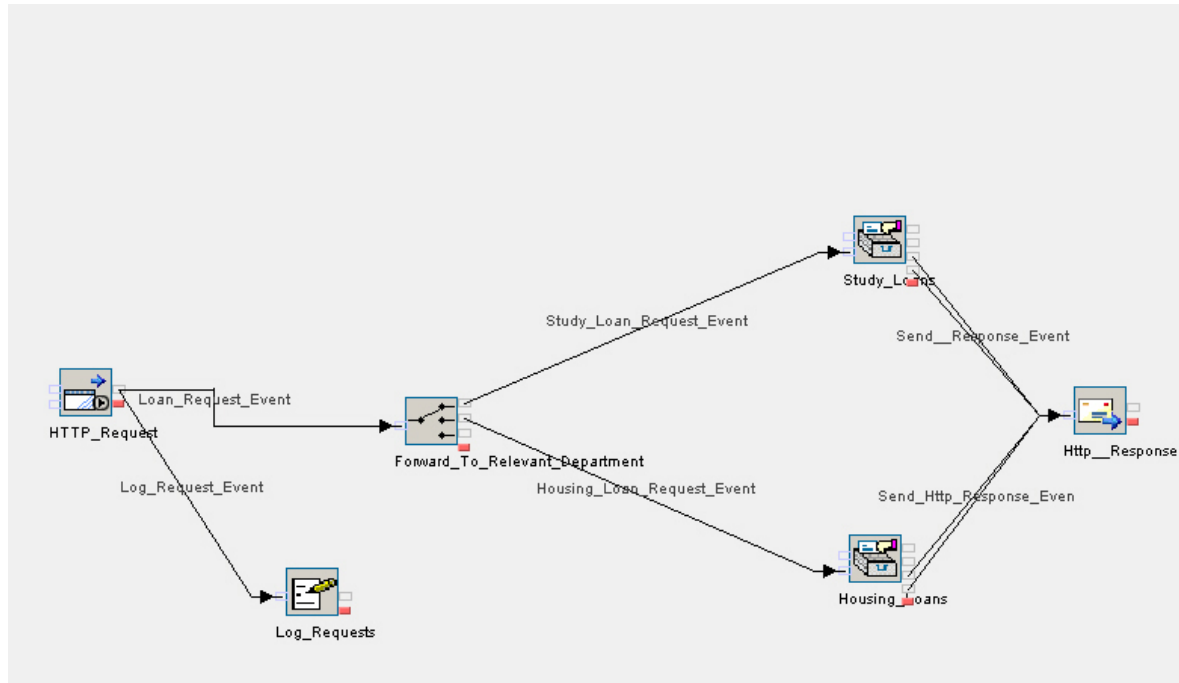


Company X runs an online market place for electronics product. Orders are accepted over web and are saved in Oracle DB. For processing an order, the company needs to perform credit-card verification using a 3rd party hosted credit card gateway and then send out orders to suppliers over email. The supplier sends back acceptance or rejection of the order (along with expected delivery schedule) by a return email. This information needs to be updated in the Oracle DB, so that customer can track the status of his order using online means.



# STEP 1 – REQUIREMENTS GATHERING

## COMPOSITE APPLICATION – QUOTE TO BIND



End to End business process for opening a new insurance policy that includes Capture of customer information, Evaluation of risk of requestor, Process the Application, Develop a Price Quote, Send Response (Quote)



# STEP 1 – REQUIREMENTS GATHERING

## BUSINESS – DATA SYNCHRONIZATION



- Data synchronization for Disaster recovery planning and business continuance, Content distribution, Backup consolidation, Server migration.
- Completely eliminate all manual data synchronization without having to stop the system to accommodate new outlets or changes to the tables in the source or target systems
- Business process monitoring Dash Board
- Add new retail outlets
- Business users should be able to orchestrate business processes
- Timelines – ASAP
- Reduce development, maintenance cost



# STEP 1 – REQUIREMENTS GATHERING

## TECHNICAL/SYSTEMS – DATA SYNCHRONIZATION



- **TECHNICAL**
  - High Level Business process representation – see figure shown earlier
  - Databases (SQL, Oracle, DB2 and files) across disparate OS.
  - Data flow representation
  - Intended users –
    - Business Analyst (Orchestration and Execution)
    - System Administrator (setup, configuration, data transformation)
  
- **SYSTEM**
  - Platforms and Operating Systems
  - Overall network topology



# STEP 1 – REQUIREMENTS GATHERING

## OUTPUT DOCUMENTS

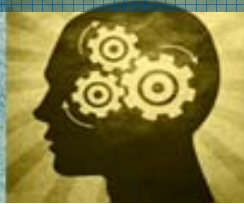


- HL Design documents that addresses the integration requirements (business & technical)
- Document all the identified data and process flows.
- List of different networks, and the servers available in each of this network
- Recommended hardware specifications for the specific ESB product
- Any 3<sup>rd</sup> party software that is essential to the functioning of ESB



# STEP 2 – IDENTIFYING “MUST HAVE” FEATURES

## DATA CONSISTENCY



- **Architecture**
  - SOA ? (Hub & Spoke, Peer to Peer, Brokered Peer to Peer)
    - ✓ Brokered P2P with store and forward at end points of the network
    - ✓ Automatic reconnects
- **Supported Standards**
  - ✓ XML, JCA, CBR
- **Message Bus (Asynchronous, Transformation etc)**
  - ✓ Asynchronous, Transformation (at source or destination)
- **Support for Distributed Applications (Compose, Execute and Monitor distributed Apps)**
  - ✓ Location and Technology transparency, Intelligent routing, Single point of control, Deployment support
- **Connectivity services ( Web services, J2EE Connectors, JMS, WebSphere MQ)**
  - ✓ JMS, Database Connectivity (JDBC)



# STEP 2 – IDENTIFYING “MUST HAVE” FEATURES

## DATA CONSISTENCY

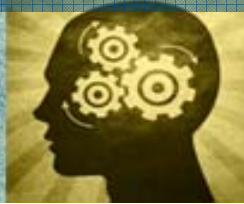


- **Administration / Deployment**
  - ✓ Single point of control
  - ✓ Dynamic changes
  - ✓ Single point of control
  - ✓ Remote access capability
  - ✓ Start / stop facilities
  - ✓ Manual routing support
  - ✓ Tracing
  - ✓ Message editing
- **Monitoring**
  - ✓ Problem determination
  - ✓ Problem prediction
  - ✓ Internal and external support
  - ✓ Support for enterprise management frameworks



# STEP 2 – IDENTIFYING “MUST HAVE” FEATURES

## DATA CONSISTENCY



- **Robustness (Service and the Infrastructure level)**
  - ✓ Fault avoidance
  - ✓ Fault tolerance
- **Scalability and Performance (Service and the Infrastructure level)**
  - ✓ Asynchronous messaging
  - ✓ Multi-threading
  - ✓ Load balancing
  - ✓ Large data handling
- **Security (Infrastructure and Application level)**
  - ✓ Access control
  - ✓ Information security
  - ✓ Tools usage



# STEP 2 – IDENTIFYING “MUST HAVE” FEATURES

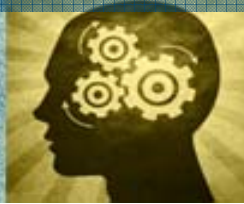
## DATA CONSISTENCY



- **Breadth of Connectivity (Configure, Modify, Connect with other Adapters)**
  - ✓ DBMS access (Oracle, SQL, DB2 etc)
  - ✓ Legacy systems (Mainframe Applications)
  - ✓ Application servers
  - ✓ .NET
  - ✓ COM / CORBA - Multi-Language Adapters
  - ✓ WebServices (Publisher and Consumer)
- **Tools**
  - ✓ Configuration (Business Processes, Services, Infrastructure)
  - ✓ Incremental deployment
  - ✓ Life cycle support



# STEP 3 – COMPONENTIZING BUSINESS PROCESSES



- Component/Service/Business Service – definition
- Fine Grained Components - Issues
- Coarse Grained Components – Advantages
- ESBs and Components
- Pre-built Components
- User Defined Components – Best practices
- Reusing existing resources



# STEP 3 – COMPONENTIZING BUSINESS PROCESSES

## FINE GRAINED COMPONENTS - ISSUES



- Typically represent function call (static input and outputs)
- Low reusability (impossible to build a general purpose Database Adapter)
- Development overheads (tight coupling between client and component)
- Change of service functionality needs re-coding ( Static data formats for input and output)
- Does not lend well to business process composition
- Synchronous invocation of function calls
- Skilled developers needed to use these components



# STEP 3 – COMPONENTIZING BUSINESS PROCESSES

## COARSE GRAINED COMPONENTS - ADVANTAGES



- Represents a high level business component
- Orchestrate business processes using these components
- Reusable across business process (changes only in the design time properties)
- Dynamic data formats for input and output
- Synchronous & Asynchronous invocation
- Low development cost (middleware is hidden)
- Business Analyst can orchestrate business processes



# STEP 3 – COMPONENTIZING BUSINESS PROCESSES

## ENTERPRISE SERVICES - REUSABLE, BUSINESS LEVEL BUILDING BLOCKS

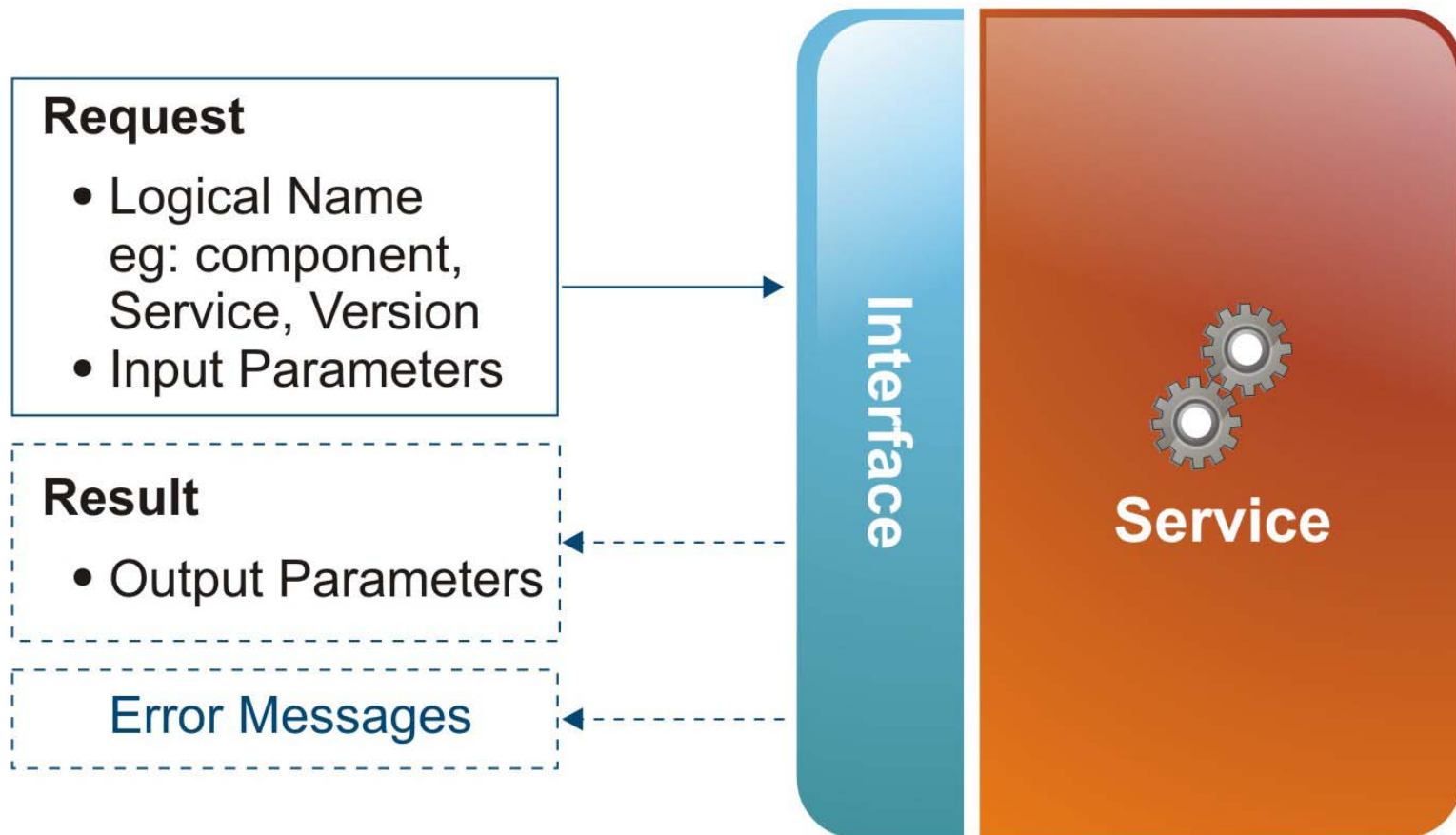


- Coarse-Grained
  - ✓ Level of abstraction easily understood by business people
- Event-Enabled Interfaces
  - ✓ Easily composed into Event Processes
- Multi-Language
  - ✓ No development restrictions
- Generalization of Web-Services
  - ✓ Standards-based WSDL interfaces



# STEP 3 – COMPONENTIZING BUSINESS PROCESSES

## SERVICE → INTERFACE



# STEP 3 – COMPONENTIZING BUSINESS PROCESSES

## COARSE GRAINED COMPONENTS



*“A component is a non-trivial, nearly independent, and replaceable part of a system that fulfills a clear function in the context of a well-defined architecture. A component conforms to and provides the physical realization of a set of interfaces.”*

- Philippe Krutchen, Rational Software

- Well defined interfaces & contracts
- Provides implementation of interfaces
- Coarse grained functionality



# STEP 3 – COMPONENTIZING BUSINESS PROCESSES

## COARSE GRAINED COMPONENTS



*"A software component is a unit of composition with contextually specified interfaces and explicit context dependencies only. A software component can be deployed independently and is subject to third-party composition."*

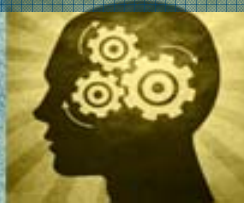
- Clemens Szyperski, Component Software

- Developed, tested and deployed in complete isolation
- Can be configured for different environment during deployment
- Supports contextual dependencies.



# STEP 3 – COMPONENTIZING BUSINESS PROCESSES

## COARSE GRAINED COMPONENTS



*"A component is a physical and replaceable part of a system that conforms to and provides the realization of a set of interfaces...typically represents the physical packaging of otherwise logical elements, such as classes, interfaces, and collaborations."*

- Grady Booch, Jim Rumbaugh, Ivar Jacobson

- Includes physical packaging of files, executables, jars, dlls etc.
- Aggregation of classes & functions into 1 complete functionality.
- Higher level of reusability than Classes



# STEP 3 – COMPONENTIZING BUSINESS PROCESSES

## PRE-BUILT SERVICES

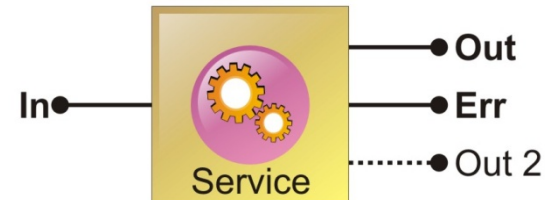
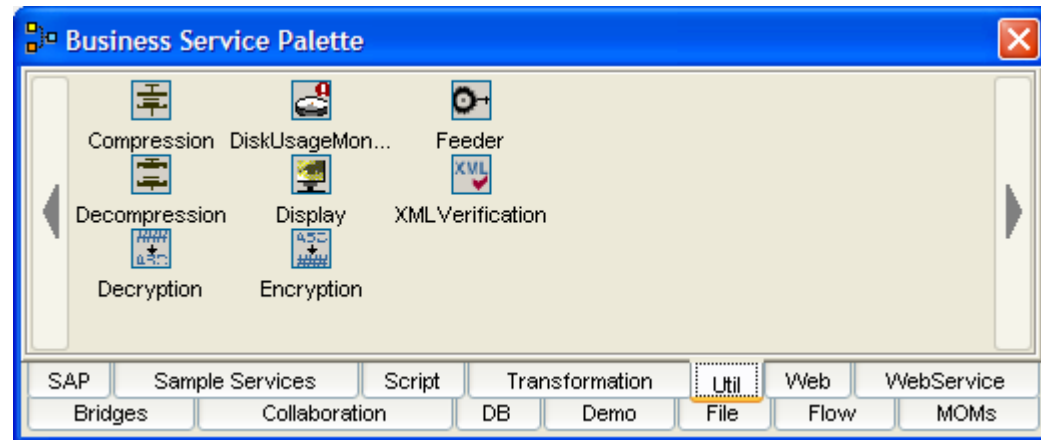


### Services

- ✓ Bridges
- ✓ Database/File Adapters
- ✓ Web/WebService Adapters
- ✓ Routing (flow services)
- ✓ Transformation
- ✓ Signing, encryption, encoding Services

### Adapters

- ✓ Middleware
- ✓ Application



# STEP 3 – COMPONENTIZING BUSINESS PROCESSES

## USER DEFINED – BEST PRACTICES



- Identifying set of user defined services
- Choosing Appropriate programming language  
(Java, C, C++, C#, VB, ActiveX, Perl, Python, JavaScript etc)
- Identifying input and output ports  
(dynamic or static)
- Custom Property Sheet  
(Design & runtime properties)
- Exception and Error Handling mechanisms
- Tracing and Logging support
  - ❖ Trace Modules and trace levels
  - ❖ Logging details
- Transaction support
  - ❖ Local
  - ❖ Global



# STEP 3 – COMPONENTIZING BUSINESS PROCESSES

## USER DEFINED – BEST PRACTICES



- Custom Property Sheet (Design & runtime properties)

**RDBMS SQL Adapter Custom Properties**

Steps in wizard

1. Database Configuration
2. SQL Configuration
3. DateTime Format
4. Advanced Properties
5. Error Handling

**Database Configuration**  
Information required to connect to database

Database: Oracle

JDBC Driver: oracle.jdbc.driver.OracleDriver

JDBC URL: jdbc:oracle:thin:@qalab06:1521:ORCL

User Name: tinfosiq

Password: \*\*\*\*\*

Test Connection

Help Previous Next Finish Cancel



# STEP 3 – COMPONENTIZING BUSINESS PROCESSES

## USER DEFINED – BEST PRACTICES



- Identifying input and output ports (dynamic or static)
- Data format definitions (XSD, DTD, XML)

The screenshot displays the Fiorano Event Process Orchestrator interface. On the left, the Event Process Explorer shows a tree view of the Fiorano Enterprise, including various event processes like EAI Demo, Batch Processing, and Database Replication. The Properties window for DBAdapter1 is visible, showing attributes like Instance Name, Node Name, and Version. The main area shows the Event Port Properties dialog for the XSD 'SQL\_CFG\_1\_OUT\_PORT'. The dialog has tabs for Tree View, Text View, and Sample XML. The Tree View shows the XSD structure:

```
<E> SQL_CFG_1  
  <E> [ZERO MANY]  
    <E> countryName_Update  
      <E> UpdateCount
```

Below the tree view is a table with columns 'Parameter' and 'Value'.

Parameter	Value

A 'Close' button is located at the bottom right of the dialog.

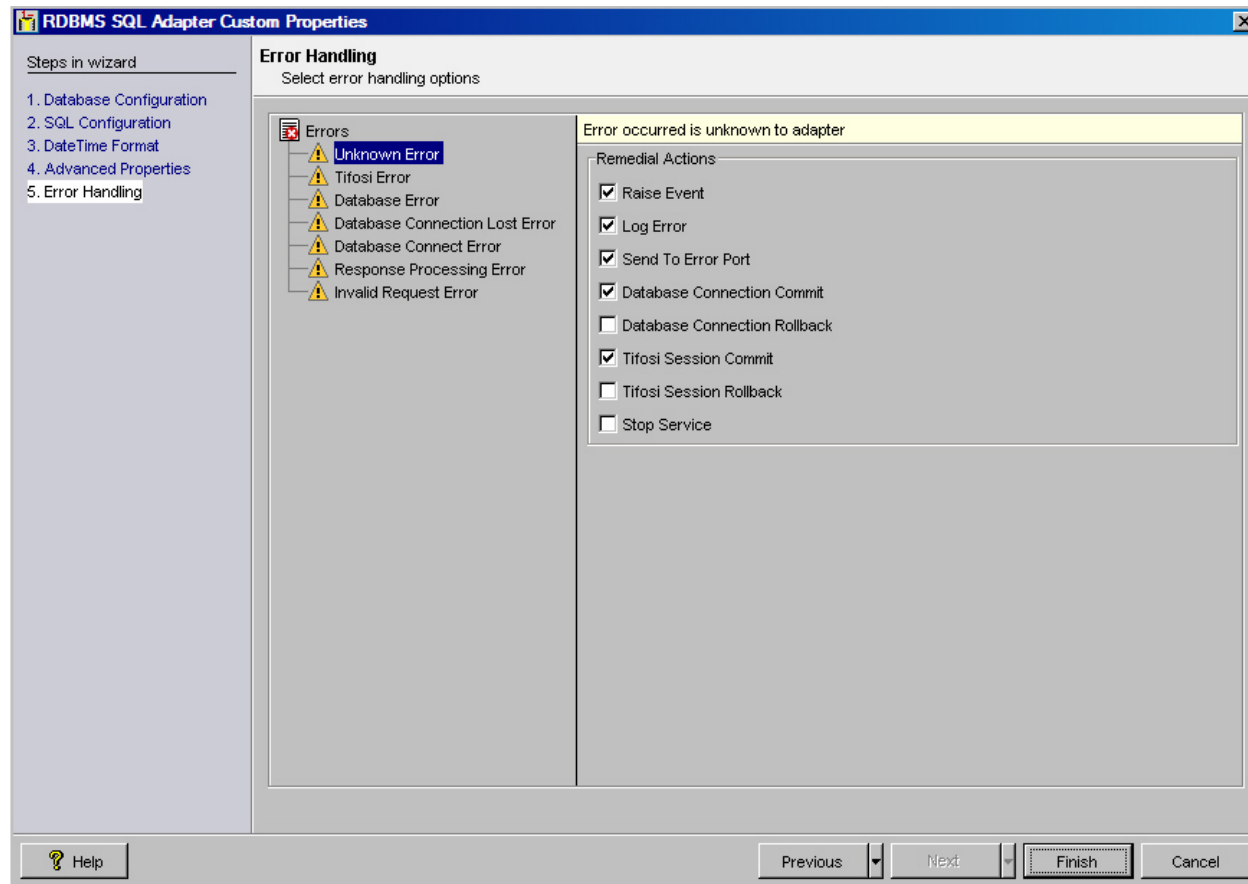


# STEP 3 – COMPONENTIZING BUSINESS PROCESSES

## USER DEFINED – BEST PRACTICES



- Exception and Error Handling mechanisms



# STEP 3 – COMPONENTIZING BUSINESS PROCESSES

## USER DEFINED – BEST PRACTICES



- Tracing and Logging support
  - ❖ Trace Modules and trace levels
  - ❖ Logging details

The screenshot displays the Fiorano Event Process Orchestrator interface. The main window shows the Event Process Explorer on the left, listing various event processes under 'Fiorano Enterprise'. The Properties window for 'DBAdapter1' is open, showing attributes like Instance Name, Node Name, Service GUID, Version, Version Locked, and Is Manual Launch. A dialog box titled 'Event Category Configuration - DBAdapter1' is also open, showing a table for Business Service Event Categories and trace level configuration.

Category	Event Trace Level	Alert
DBService	Quiet	<input type="checkbox"/>



# STEP 3 – COMPONENTIZING BUSINESS PROCESSES

## USER DEFINED – BEST PRACTICES



- Transaction support
  - ✓ Local (Client level transactions)
  - ✓ Global (XA)
  - ✓ Compensation transactions
- Performance Requirements
  - ✓ Expected transaction throughput
  - ✓ Multi-threading support
- Synchronous/Asynchronous

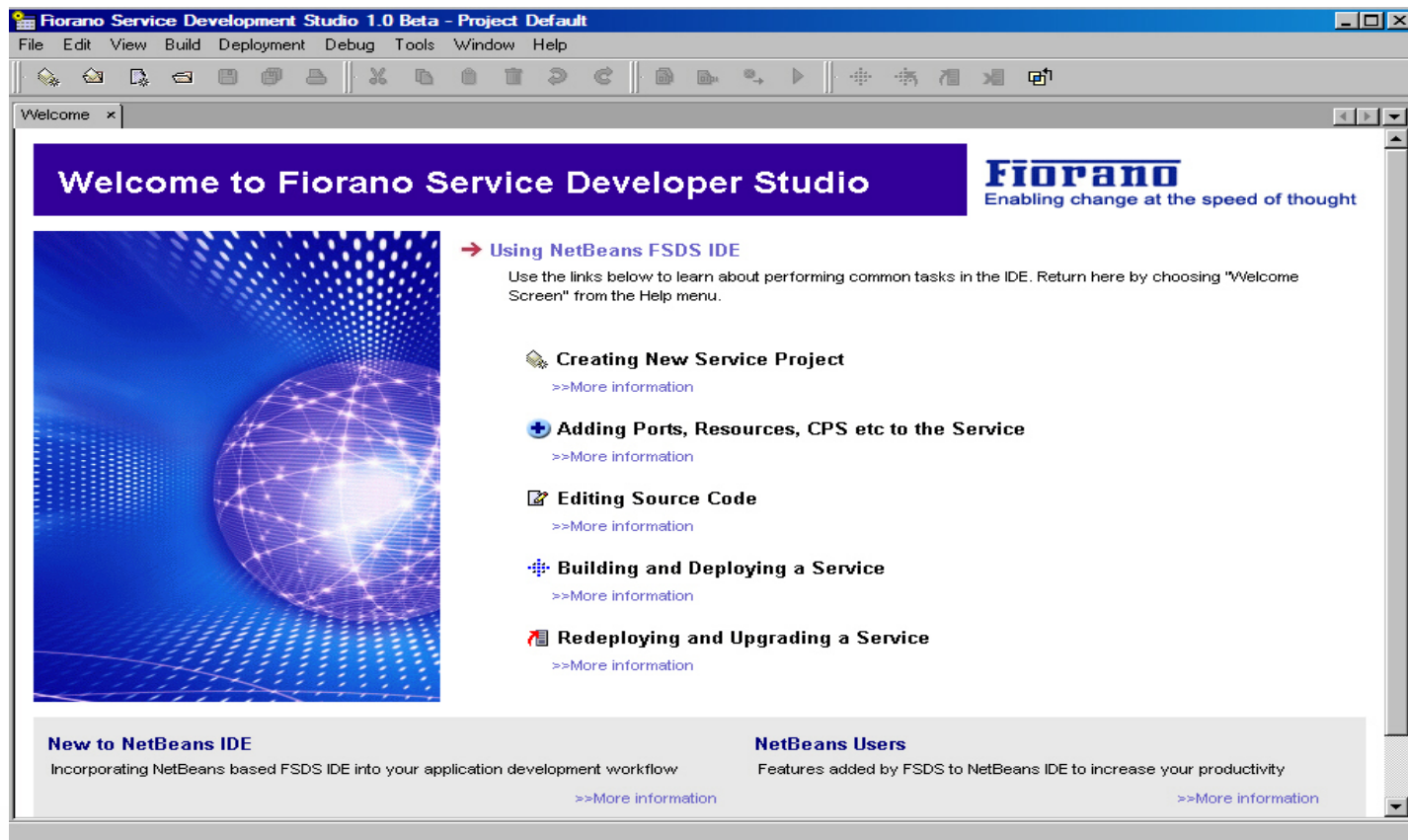


# STEP 3 – COMPONENTIZING BUSINESS PROCESSES

## USER DEFINED – BEST PRACTICES

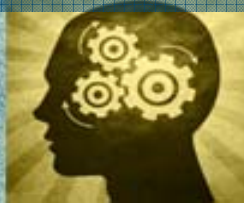


- Tools for service development
- IDE plug-ins



# STEP 3 – COMPONENTIZING BUSINESS PROCESSES

## REUSING EXISTING SERVICES – BEST PRACTICES



- Wrap existing code (Java, C, C++, VB etc)
- Adapters to COM, DCOM, EJB, CORBA, RMI etc
- Reusing business processes



# STEP 4 – DEFINING DISTRIBUTED ESB PROCESS



- Control vs. Data Flow
- Business Process Orchestration
- Event Warehousing
- Security
- Logging, Tracing & Alerts
- Data format impedance mismatch
- Configuring Business process for failover
- Configuring Business process for performance



# STEP 4 – DEFINING DISTRIBUTED ESB PROCESS

## CONTROL VS DATA FLOW



- Control Flow

The screenshot displays the Fiorano Business Process Composer interface. The left sidebar contains a palette of activities and invoke targets. The main canvas shows a sequence flow diagram with the following steps: Start, Assign, WebService, Assign, WebService, Assign, and Stop. The bottom of the interface has tabs for 'Flow Chart' and 'Algorithm'.

Name	Value
Process Name	unknownProcess
Target Namespace	http://www.fiorano.com
WSDLFiles	{2.wsdl=Definition: name={h...
Namespaces	{soapenc=http://schemas.x...
Partners	{tifosiSLT={http://www.fior...
Variables	{getQuoteRequest1={http://...
suppressJoinFailure	no
enableInstanceCompensation	no
abstractProcess	no
expressionLanguage	http://www.w3.org/TR/199...
queryLanguage	http://www.w3.org/TR/199...

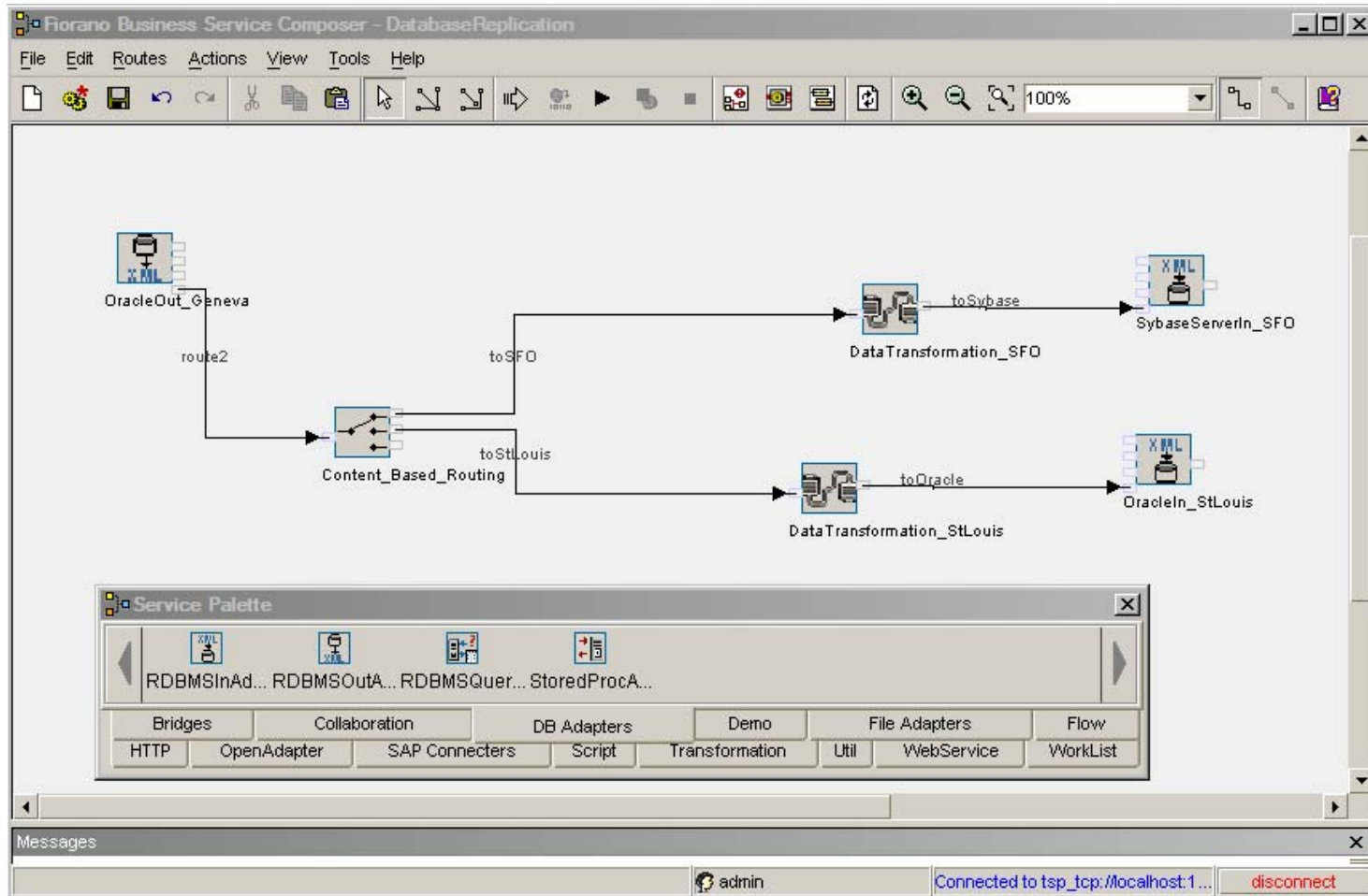


# STEP 4 – DEFINING DISTRIBUTED ESB PROCESS

## CONTROL VS DATA FLOW

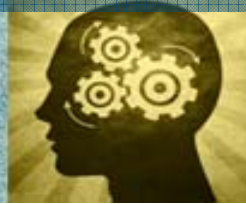


- Data Flow



# STEP 4 – DEFINING DISTRIBUTED ESB PROCESS

## BUSINESS PROCESS ORCHESTRATION



- Composition using pre-built services
- Data routing
- Control information
- Data transformation
- Identifying Node Names
- Configuring Service Design time properties



# STEP 4 – DEFINING DISTRIBUTED ESB PROCESS EVENT WAREHOUSING



- Enable at runtime

The screenshot displays the Fiorano Event Manager application. The main window shows a tree view on the left with 'Tracked Documents' selected. The main pane displays a table of running event processes:

Instance ID	Doc ID	Status	Last Business Service	Cycle Time
WF_INST_ID:tifosi1_1084262379674_8	654443	EXECUTING	xsitransform	
WF_INST_ID:tifosi1_1084262379674_9		EXECUTING	Display1	
WF_INST_ID:tifosi1_1084262379674_10	654444	EXECUTING	xsitransform	
WF_INST_ID:tifosi1_1084262379674_11		EXECUTING	Display1	

A 'Document States Traversed' dialog box is open, showing details for WorkflowInstance ID: WF\_INST\_ID:tifosi1\_1084371041472\_4. It contains a table of document states:

State	Business Service	Date	Doc ID	Processing Time (ms)	Comments
OUT_...	CRM	12 May 200...	DOC_ID:CRM_000BA_EAI_D...		Document sent fr...
PO_IN	ERP	12 May 200...	DOC_ID:ERP_000BA_EAI_D...		Document Reache...
PO_R...	ERP	12 May 200...	DOC_ID:ERP_000BA_EAI_D...		Document sent fr...
IN_PO...	Rejected Orders	12 May 200...	DOC_ID:Rejected Orders_00...		Document Reache...
OUT_...	Rejected Orders	12 May 200...	DOC_ID:Rejected Orders_00...	130	Document sent fr...

# STEP 4 – DEFINING DISTRIBUTED ESB PROCESS SECURITY



## Security (Role based Security)

The screenshot shows the Fiorano Services and Security Manager interface. The left pane displays a tree view of the Fiorano Enterprise structure, with 'Access Rights Assignment' selected under 'Security Policy'. The right pane displays a table of network rights and their assigned roles.

Network Rights	Assigned To
PERMISSION TO CLEAR USER EVENTS	ADMINISTRATORS
PERMISSION TO PUSH MESSAGES IN QUEUE	ADMINISTRATORS
PERMISSION TO CREATE OR EDIT AND DELETE A PRINCIPAL	ADMINISTRATORS
ALL PERMISSIONS	ADMINISTRATORS
PERMISSION TO DELETE MESSAGES IN QUEUE	ADMINISTRATORS
PERMISSION TO COMPOSE AN APPLICATION	WORKFLOW COMPOSER,ADMINISTRATORS
PERMISSION TO VIEW MESSAGES IN QUEUE	ADMINISTRATORS
PERMISSION TO CHANGE PROPERTIES OF AN APPLICATION	WORKFLOW COMPOSER,ADMINISTRATORS
PERMISSION TO CREATE OR EDIT AND REMOVE SERVICE ACL	ADMINISTRATORS
PERMISSION TO VIEW RUNNING AND SAVED APPLICATIONS	PRODUCTION ENGINEER,WORKFLOW COMPOSER,ADMINISTRAT
PERMISSION TO KILL AN APPLICATION	WORKFLOW COMPOSER,PRODUCTION ENGINEER,ADMINISTRAT
PERMISSION TO CONFIGURE A TPS	ADMINISTRATORS
PERMISSION TO CREATE AN ACL	ADMINISTRATORS
PERMISSION TO CREATE OR UPDATE AND DELETE A SERVICE	ADMINISTRATORS
PERMISSION TO LAUNCH AN APPLICATION	WORKFLOW COMPOSER,PRODUCTION ENGINEER,ADMINISTRAT



# STEP 4 – DEFINING DISTRIBUTED ESB PROCESS LOGGING, TRACING & ALERTS



Log Category Configuration - xsltransform

Business Service Log Categories and trace level configuration:

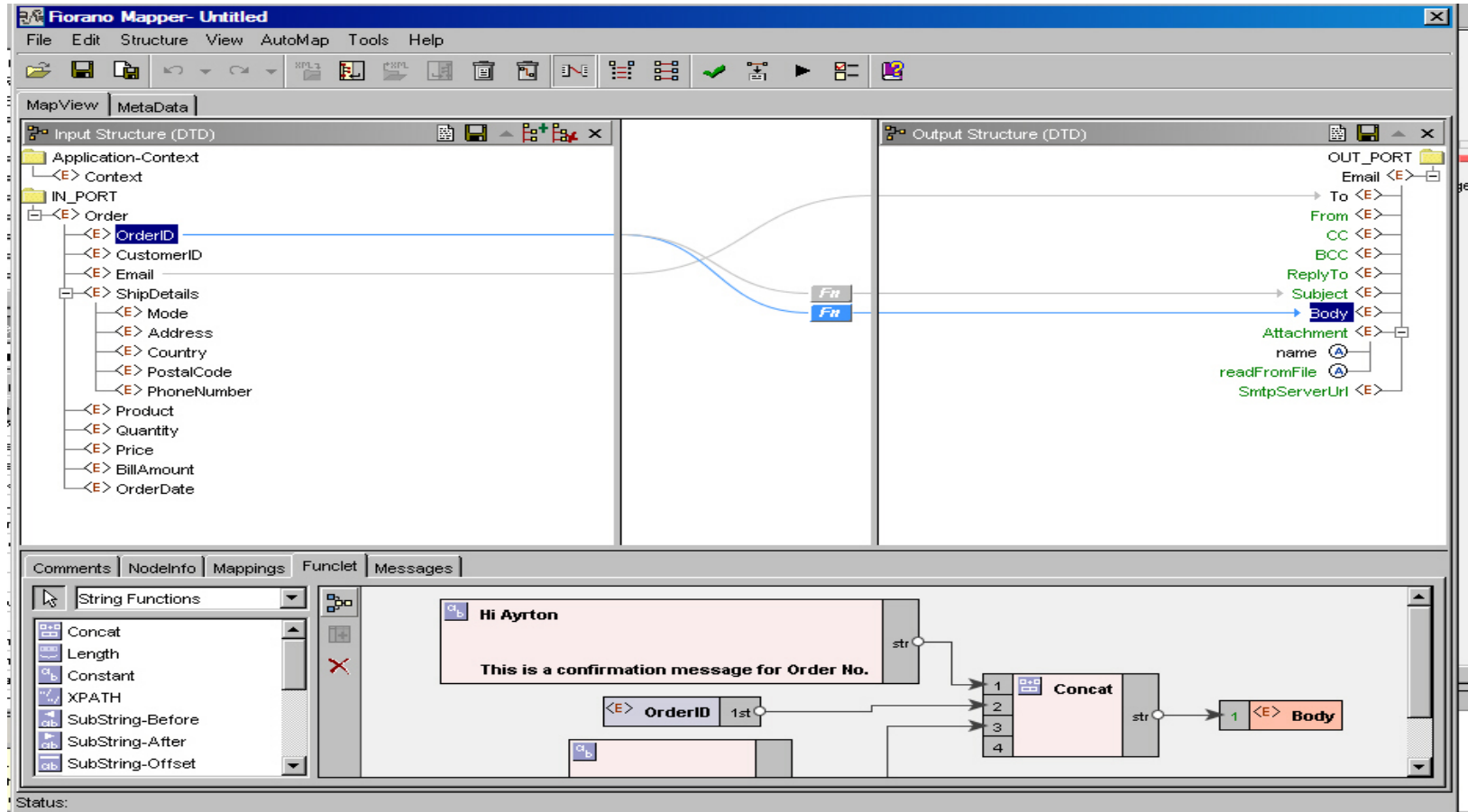
Category	Trace Level
XSLTServiceException	Quiet
XSLTServiceInfo	Quiet

Ok Cancel



# STEP 4 – DEFINING DISTRIBUTED ESB PROCESS

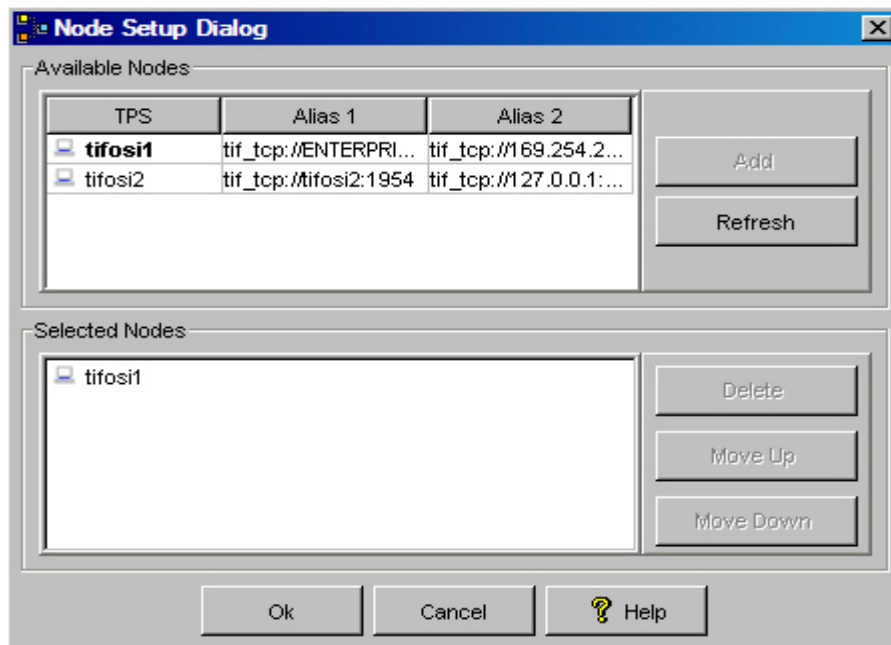
## DATA FORMAT IMPEDANCE MISMATCH



# STEP 4 – DEFINING DISTRIBUTED ESB PROCESS CONFIGURING BUSINESS PROCESS FOR FAILOVER



- Multiple node names for Services
- State failover for services
- Server level failover



# STEP 4 – DEFINING DISTRIBUTED ESB PROCESS

## CONFIGURING BUSINESS PROCESS FOR PERFORMANCE



- Identifying Parallel data flows
- Dynamic rerouting of data based on load
- Identifying Heavy weight services (80/20 Rule)
- Running multiple instances of “heavy weight” services on different nodes
- Sub-Flows and Sub-Processes for effective BP representations
- Log/Trace level optimization
- Event tracing optimization



# STEP 4 – DEFINING DISTRIBUTED ESB PROCESS CONFIGURING BUSINESS PROCESS FOR PERFORMANCE



- Service level Load Balancing (static/Dynamic)

**DistributionService Custom Property Sheet Wizard**

Steps in wizard

1. Output Port Details
2. Weight Distribution

**Weight Distribution**  
specify weight of each output port

Port Name	Weight
OUT_PORT_0	1
OUT_PORT_1	1
OUT_PORT_2	1
OUT_PORT_3	1
OUT_PORT_4	1

Help Previous Next Finish Cancel



# STEP 5 – DEPLOYING ESB PROCESS



- Deployment - what does it mean ?
- Identifying Network Domain/Topology
- Manual Services vs. Auto-Launched Services
- Security issues for Deployment
- Service Development Languages and Platforms



# STEP 5 – DEPLOYING ESB PROCESS

## WHAT DOES IT MEAN?



- Deployment Descriptor

**Service Properties**

Steps in wizard

1. Business Service Header
2. Resources & Dependencies
3. Business Service Description
4. Event Ports Information
5. Execution Information
6. Logging & Document Tracking
7. Advanced Configuration

### Resources & Dependencies

specify the resources & Dependencies for the Business Service

Resources:

Name	For Configuration	For Execution
DBAdapter.jar	<input type="checkbox"/>	<input checked="" type="checkbox"/>
DBAdapterui.jar	<input checked="" type="checkbox"/>	<input type="checkbox"/>
antlr.jar	<input checked="" type="checkbox"/>	<input type="checkbox"/>
DMLParser.jar	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Tifosi Libraries:

Library name	Version
TifosiJavaRTL	2.0
TifosiUIFramework	1.0
AdapterUtils	1.0
OracleDriver	1.2
XMLUtil	1.0
Base64	1.0

Applicable OS:  Win32  Solaris  Linux  MacOS  Others

Buttons: Add, Remove, Help, Previous, Next, Finish, Cancel

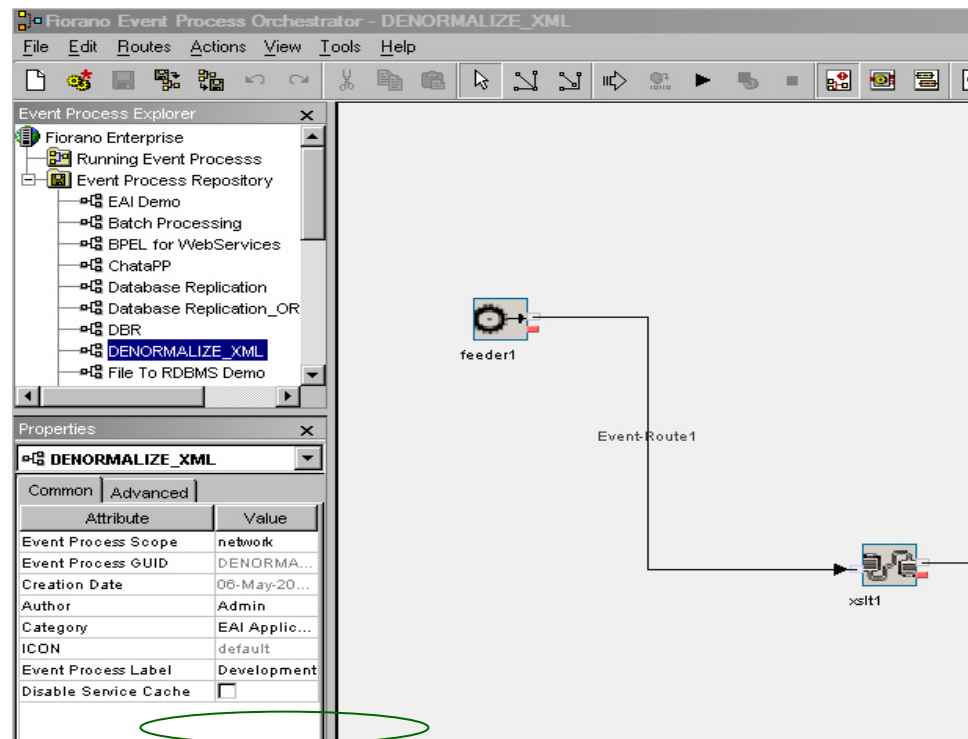


# STEP 5 – DEPLOYING ESB PROCESS

## WHAT DOES IT MEAN?



- Remote Deployment of Service and its dependencies
- Remote installation of service dependencies
- Caching



# STEP 5 – DEPLOYING ESB PROCESS

## WHAT DOES IT MEAN?



- Managing Dependencies / Order of installation /Order of Launch

**Business Service Upgradation Wizard**

Steps in wizard

1. Business Service Header
2. Resources & Dependencies
3. Business Service Description
4. Event Ports Information
5. Execution Information
6. Logging & Document Tracking
7. Advanced Configuration

Select the panel to view.

**Advanced Configuration**  
specify advanced properties to configure business service

- Is auto installable.
- Is auto launchable.
- Supports Error Handling.
- Supports Failover to another Peer Server.
- Is manually launchable.
- Select manual launch by default.
- Supports in-memory launch.
- Select in-memory launch by default.

Cleanup Service

GUID:  Version:

Help Previous Next Finish Cancel



# STEP 5 – DEPLOYING ESB PROCESS

## IDENTIFYING NETWORK DOMAIN/TOPOLOGY



- Deployment Domain/Nodes

The screenshot displays the Fiorano ESB Administrator interface. The title bar reads "Fiorano ESB Administrator" with a menu bar containing "Actions", "View", and "Help". Below the menu bar is a toolbar with various icons. The main area is split into two panes. The left pane shows a tree view of the "Fiorano Enterprise" configuration. The right pane shows a detailed view of the selected "Fiorano Enterprise" configuration.

**Fiorano ESB Administrator**  
Actions View Help

**Fiorano Enterprise**

- MessageBus Cluster
- PRIMARYSP**
  - Configuration
  - System Events and Logs
  - Status
- Peer Server Groups
  - TifosiDomain
    - tifosi1
    - tifosi2

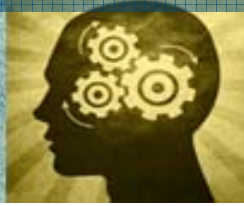
**Fiorano Enterprise**

- MessageBus Cluster
- PRIMARYSP**
- Peer Server Groups



# STEP 5 – DEPLOYING ESB PROCESS

## IDENTIFYING NETWORK DOMAIN/TOPOLOGY



- Domain/Node Name Alias (Hierarchical Domains)
- Configuration Alias



# STEP 5 – DEPLOYING ESB PROCESS

## MANUAL VS AUTO LAUNCH



### Manual Services

- Executed externally to the ESB (Servlets, EJBs etc)
- Combination of managed and unmanaged components
- Managed Components like:-
  - ✓ A Webservice deployed in a Webservice container
  - ✓ An EJB deployed in J2EE container
  - ✓ A COM Object deployed in COM+ server
  - ✓ A CORBA based server Object deployed in an ORB
  - ✓ Windows NT/2K Service
- Unmanaged Components like:-
  - ✓ Java executable archive
  - ✓ A C/C++ executable
  - ✓ Legacy Application running in a mainframe environment.
  - ✓ A Unix shell program (functioning within a pipe-and-filter style architecture).



# STEP 5 – DEPLOYING ESB PROCESS

## MANUAL VS AUTO LAUNCH



### Auto Launched Services

- Native ESB services- managed and launched by ESB containers
  - Auto start/stop and restart of these services
  - Connectivity management
  - Fine grained monitoring



# STEP 5 – DEPLOYING ESB PROCESS SECURITY ISSUES



- Deployment Manager to restrict service deployment

Fiorano Deployment Manager

Actions View Help

Available Rules (Default rule is A)

Rule1

Rule Description (click on a blue link to edit it)

Disallow execution of business service[s] where business service version matches 2.0 as part of event process[s] where event process version matches 1.0 on peer server(s) where peer server label contains QA

Name of the Rule

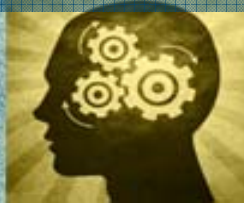
QA\_Rule

Ok Cancel

admin Connected to tsp\_tcp://enterprise:1947 disconnect



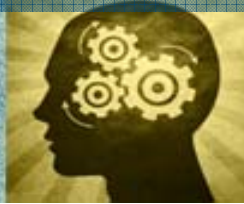
# STEP 6 – LAUNCHING & MONITORING ESB PROCESS



- Remote Launch of Business Process
- Monitoring state of Application and all its associated service instances
- Runtime hooks to determine state of service
- Real-time data debugging
- Business Process Monitoring
  - ✓ SNMP, JMX etc
  - ✓ Monitoring of Servers



# STEP 6 – LAUNCHING & MONITORING ESB PROCESS



## Launch of ESB process

- Remote Launch of ESB Services
- Service Instantiation
  - ✓ On-demand/On-Event Instantiation
  - ✓ Auto Instantiation
- Re-Launch of Business Process
  - ✓ Auto re-launch on different set of nodes
  - ✓ Rules based re-launch of business process

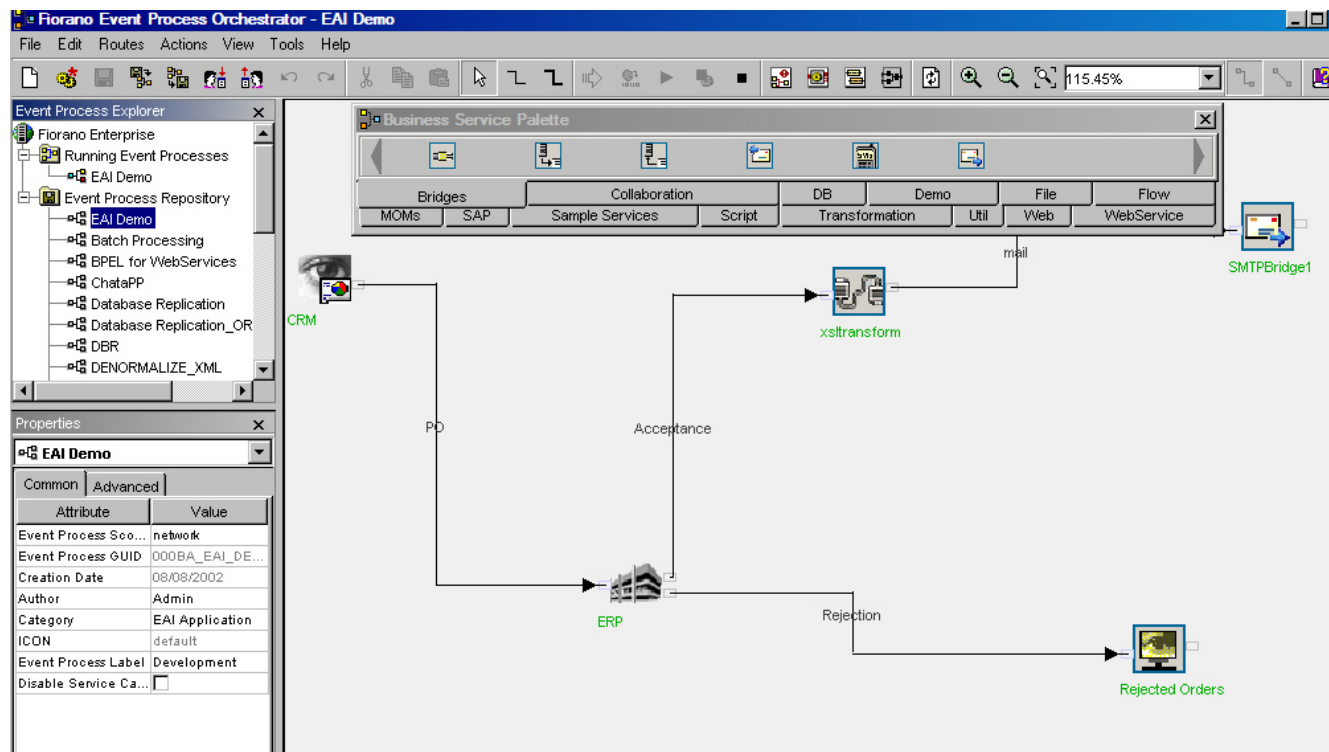


# STEP 6 – LAUNCHING & MONITORING ESB PROCESS



Monitoring state of Application and all its associated service instances

- ✓ Local Queue Monitoring and Management
- ✓ Identify performance/scalability bottlenecks in Application
- ✓ Identify parallel flows in Application

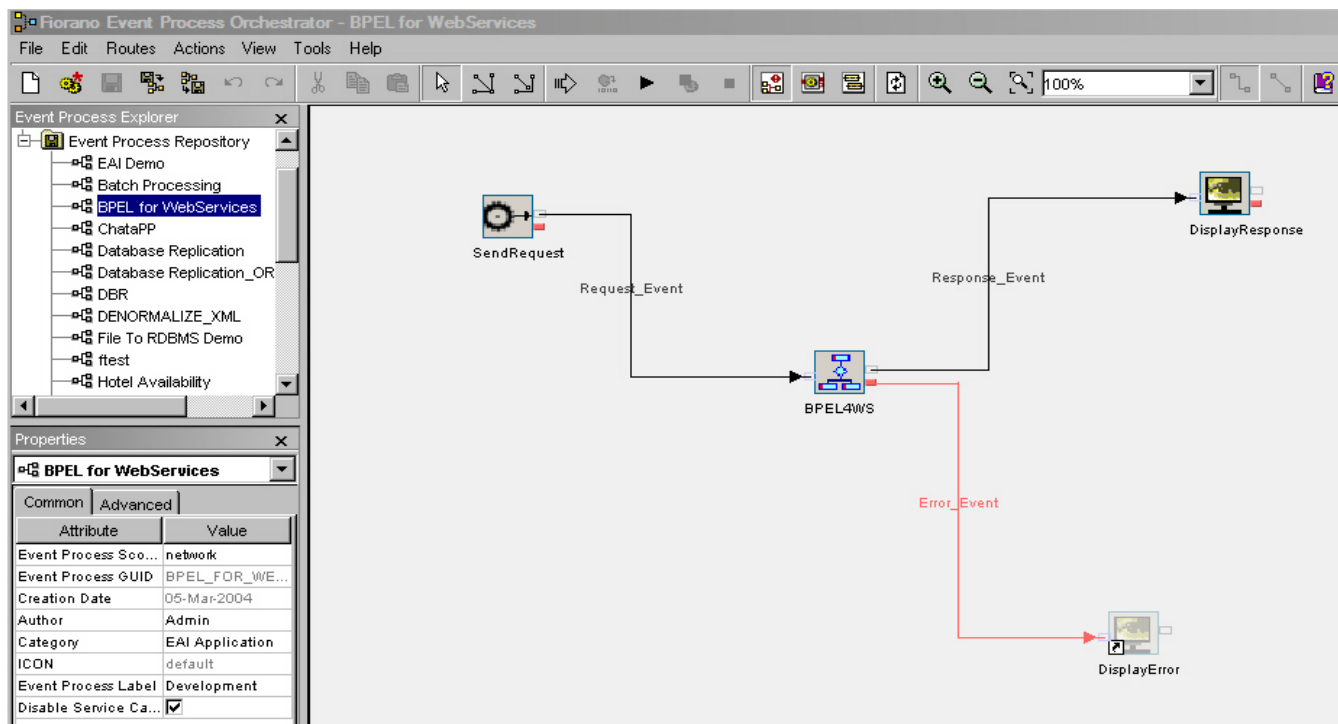


# STEP 6 – LAUNCHING & MONITORING ESB PROCESS



## Runtime hooks to determine state of service

- ✓ Runtime tracing and Logging
- ✓ Sub-Flows to handle Error conditions
- ✓ Alert Handlers



# STEP 6 – LAUNCHING & MONITORING ESB PROCESS



## Real-time data debugging

The screenshot shows an ESB process flow and its real-time debugging interface. The process flow at the top consists of a 'SendRequest' component connected to a 'Request\_Event' component, which is then connected to a 'Response\_Event' component, and finally to a 'DisplayResponse' component. Below this, the 'Debugger Window [Route=Request\_Event]' is open, displaying a table of received messages and a detailed view of the message body.

Index	Received	Message
1	Mon May 24 11:29:25 GMT-08:00 2004	<?xml version="1.0" encoding="UTF-8" ...

General Message Body Properties Attachments Application Context

Text Bytes

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <ns0:receivedMessage xmlns:ns0="http://fiorano.com/bpel/types" xmlns:
3   <wsdl1_getQuoteRequest1>MSFT</wsdl1_getQuoteRequest1>
4   <wsdl2_getRateRequest>
5     <country2>INDIA</country2>
6   </wsdl2_getRateRequest>
7 </ns0:receivedMessage>
```

plain text tree view

business Service EJBBridge



# STEP 6 – LAUNCHING & MONITORING ESB PROCESS



## Business Process Monitoring

- SNMP, JMX etc
- Monitoring of Servers

The screenshot shows the Fiorano ESB Administrator interface. On the left is a tree view of the configuration hierarchy, and on the right is a table of configuration parameters.

Name	Value	Description
Max. Messages per Queue	2000	Maximum no. of message
Peer Server Group Name	TifosiDomain	Peer Server Group Name
Email Address for BCC Field	tifosi@fiorano.com	The email address to wh
Drop messages on queue limit	false	Dispose/drop the messag
Inmemory queue limit	500	Inmemory queue limit.
Max. No. of Log Files	4	Maximum no. of log files
Queue Warning Percentage	80	It is the limit of message c
Is Security Enabled	false	Boolean to indicate whet
SMS Callback PIN/Mobile No.		The callback number of th
Time Stamp in Logging	true	Timestamp in LogManage
No. of Tries to Connect to MQ	15	The maximum number of
SMS Carrier / Service ID		The id of the carrier whic
SMS Connection Timeout (ms)	30000	The timeout for the SMS t
Event Handler No. 1	fiorano.tifosi.events.handler.jms.JMSEventHandler	Event handlers for gener
Queue alerts interval	10	Time interval(in min.) betw
Log Manager Module	fiorano.tifosi.log.def.LogManagerImpl	Log Manager is used for
Alert Handler 1	fiorano.tifosi.events.handler.smtp.SMTPEventHandler	Handlers required for gen
Email Address of the receiver	tifosi@fiorano.com	The email address to wh
Pager Identification / Mobile No.		The Pager Identification N
Ping Timeout (ms)	10000	The time in milliseconds a
Max. No. of Lines in Log File	1000	Number of lines permitte
Email Address of sender	tifosi@fiorano.com	The name/email address
Outgoing Mailserver	fiorano.com	The SMTP server to be u
Drop Previous Messages	false	Boolean indicating if the r
Log File Path Location	.	Set the directory where l
Email Address for CC Field	tifosi@fiorano.com	The email address to wh
Check for Host IP	true	Indicate whether the TPS
Peer Server Label	Development	Deployment stage label f
SMS Sender Name		Any information about the



# STEP 6 – CHANGE MANAGEMENT & VERSIONING



- **Dynamic Extensions to Business process**
  - ✓ Static vs. Dynamic
  - ✓ Impact Analysis
  - ✓ Extend Business process to include new services
  - ✓ Extend Business process to include data services with different data formats
  - ✓ Optimize Business process for performance, scalability
  - ✓ Extend Business processes to handle network configuration changes
- **Dynamic changes to Business process**
- **Service and Application Versioning**



# STEP 6 – CHANGE MANAGEMENT & VERSIONING



- **Configuration Management**
  - ✓ Moving from one stage to another (QA to Deployment)
  - ✓ Moving from one environment to another (customer to internal testing)
- **Extension to Business process**
  - ✓ Updating Data Consistency Application to include the data center in San Francisco
- **Change to Business Process**
  - ✓ Flexibility to adapt to new technologies (WebService instead of a C++ stand alone Application)



# STEP 6 – CHANGE MANAGEMENT & VERSIONING



## Service and Application Versioning

- Manage and maintain multiple versions of services along with Labels
- Quickly allow migration from one service version to another

The screenshot displays the Fiorano Event Process Orchestrator interface. On the left, the 'Event Process Explorer' shows a tree view of the 'Event Process Repository' with 'EAI Demo' selected. Below it, the 'Properties' window for the 'CRM' service is visible, showing attributes like Instance Name, Node Name, Service GUID, and Version (3.0). The main workspace shows a workflow diagram with a 'CRM' service icon connected to an 'xsfttransform' service icon. A 'Change Service Version : CRM' dialog box is open, prompting the user to 'Select a version from the available versions'. The dialog lists three versions: 1.0, 2.0, and 3.0, with 3.0 selected. The dialog has 'Ok' and 'Cancel' buttons.

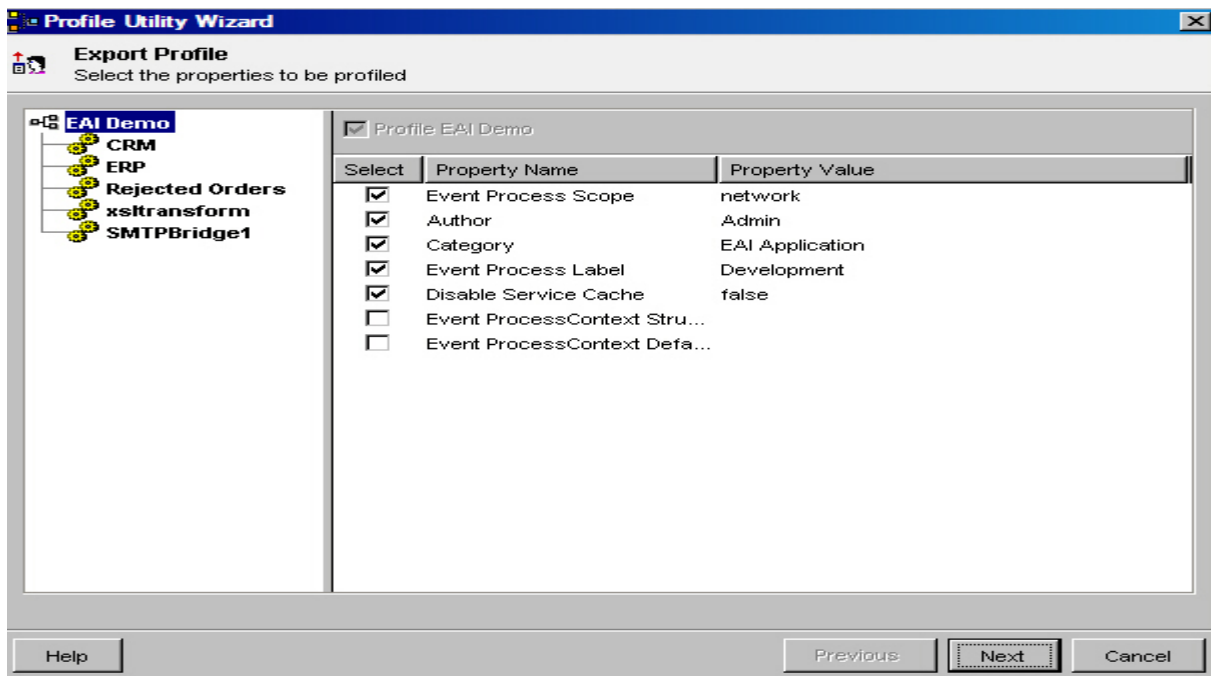


# STEP 6 – CHANGE MANAGEMENT & VERSIONING

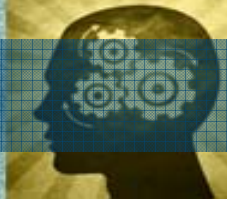


## Configuration Management

- Moving from one stage to another (QA to Deployment)
- Moving from one environment to another (internal to client deployment)



# PUTTING THEORY TO PRACTICE



# TECHNOLOGY HYPE IS CONFUSING THE ISSUES WITHOUT PROVIDING A CLEAR PATH FORWARD



Example: ZapThink view



ZapThink view

- Need to choose standards that work today and will evolve for future
- Solve the real integration problem – more than a Proof of Concepts
- Must build incrementally on top of existing systems



# STANDARDS REDUCE COSTS BUT... FUNDAMENTAL PROBLEMS REMAIN!



- **Business Person's View**
  - ✓ High-level model of business process flow
- **IT Level View**
  - ✓ Implementation flow differs substantially from business process view
- **Impedance mismatch creates fundamental problems**
  - ✓ Implementations have too many “moving parts”
  - ✓ Business-level change requirements difficult and time-consuming to implement



# THE FUNDAMENTAL PROBLEM – DIVERGENT BUSINESS TECHNOLOGY

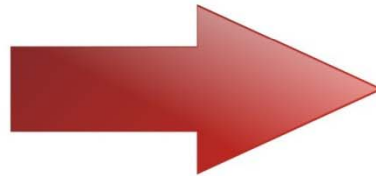


## BUSINESS

Check Inventory

Get Inventory from IMS

Get Inventory from  
Partner Apps



## TECHNOLOGY

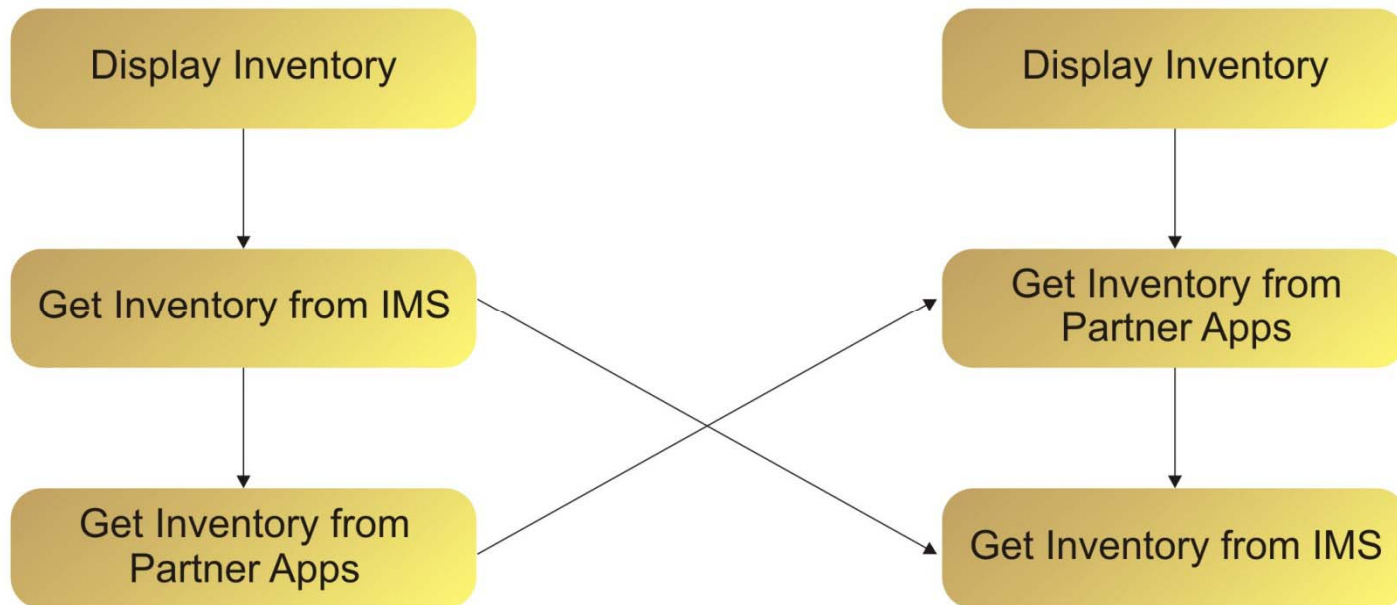
Web Apps

Inventory Management  
System

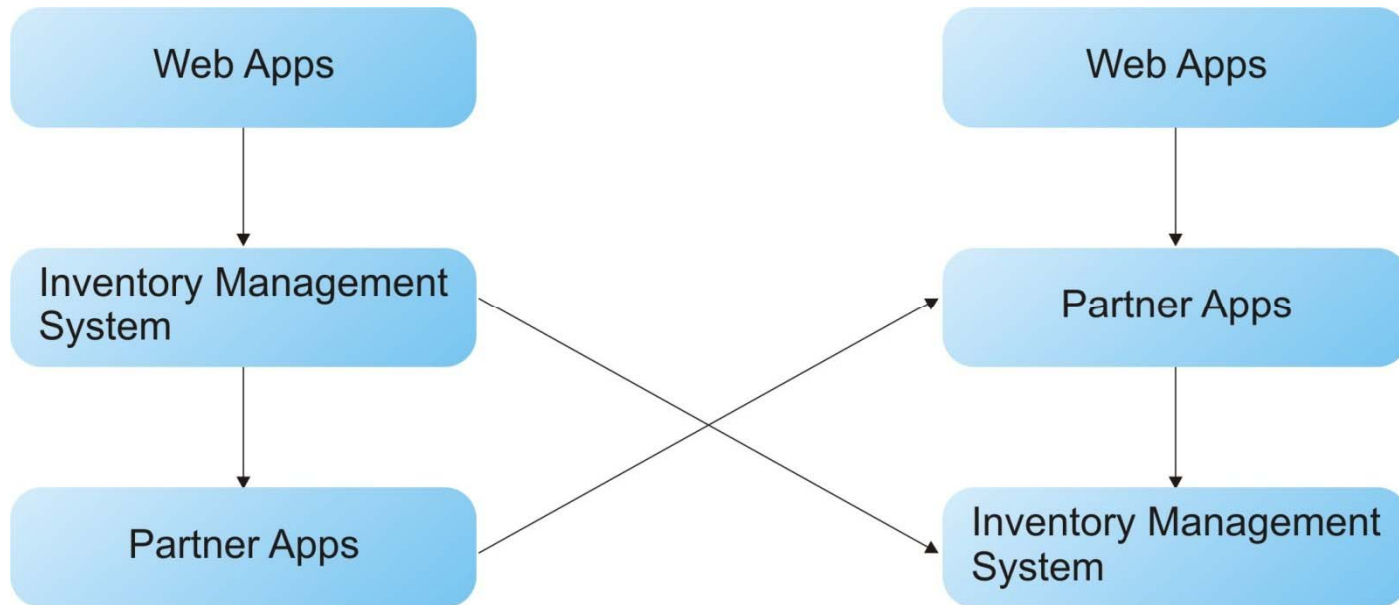
Partner Apps



# A CHANGE IN BUSINESS PROCESS

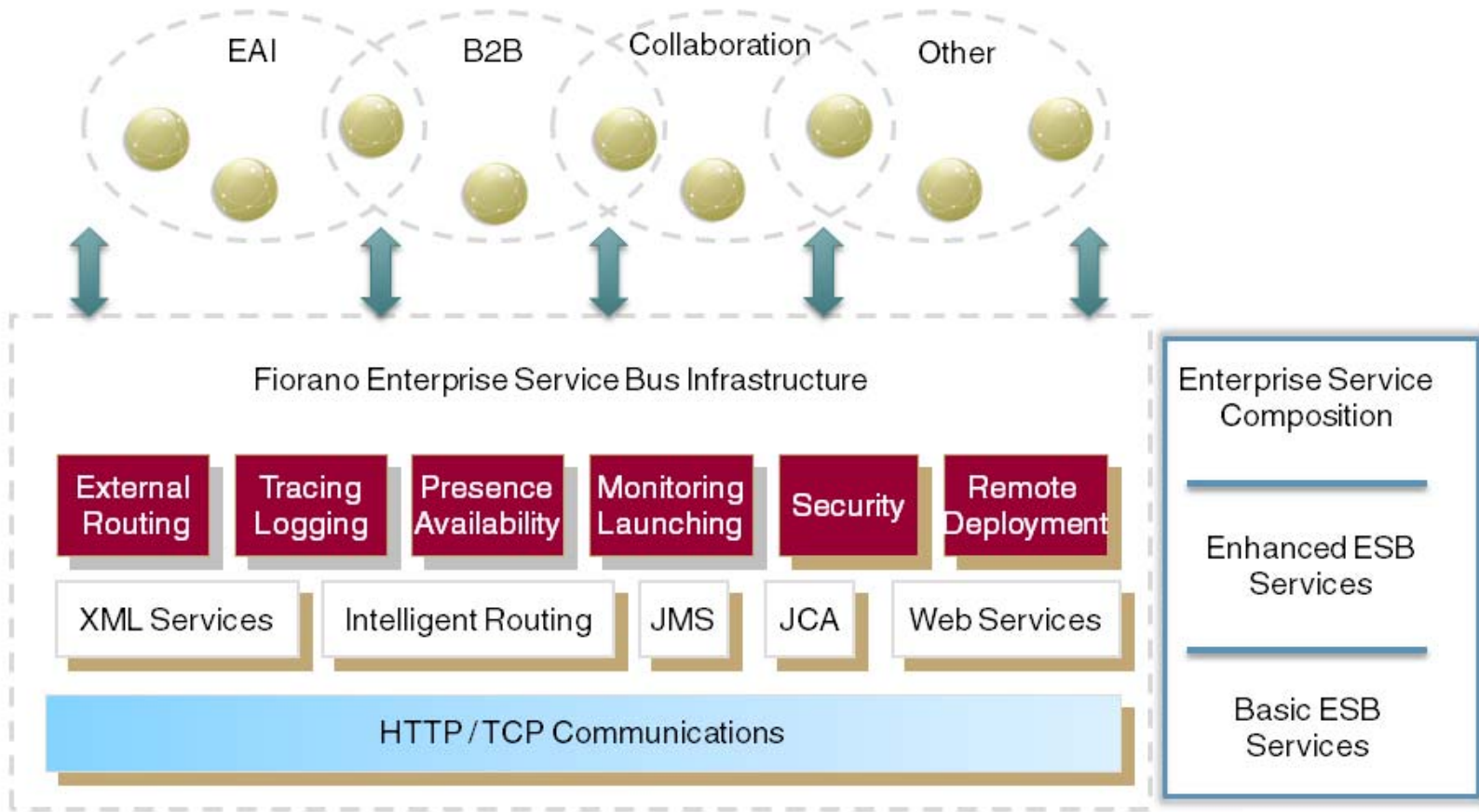


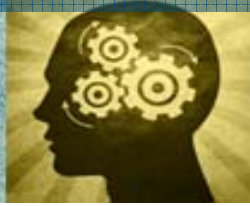
# IS NOT EASILY MAPPED TO IMPLEMENTATION LEVEL



# FIORANO ESB

## SECOND GENERATION ESB





THANK YOU!

FOR MORE DETAILS, VISIT [www.fiorano.com](http://www.fiorano.com)

