SOA Technology Summit

SOA Summit – Paris 13th June

Building an ESB
Services Infrastructure Deployment

"Collaborate to Innovate"

Arnauld Desprets, IT Architect
IBM ISSW – Pan EMEA team (IBM Software Services for WebSphere)
arnauld_desprets@uk.ibm.com

SOA on your terms and our expertise

Agenda

- Why do we need an Enterprise Service Bus?
  - Achieving loose coupling
  - Capabilities
- Making a Bus
  - Patterns
  - Technologies and products
- Real-life samples
- Managing an ESB

Objectives

At the conclusion of this presentation, you should be able to

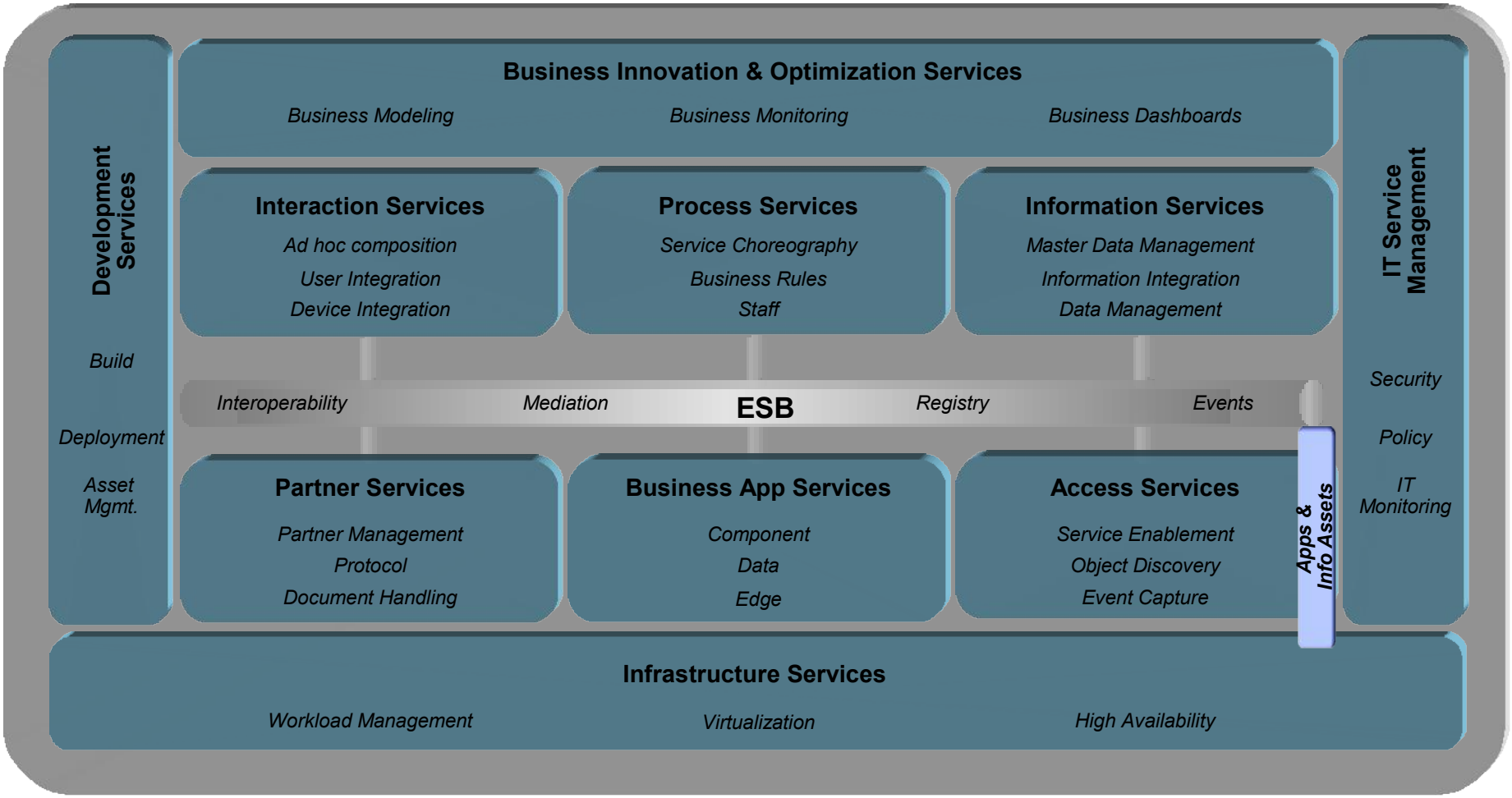
- **Have a common understanding of what is an ESB**
- **The benefits achieved by implementing an ESB**
- **How to implement an ESB using IBM technologies**

We assume a common understanding of Web Services and SOA principles

Agenda

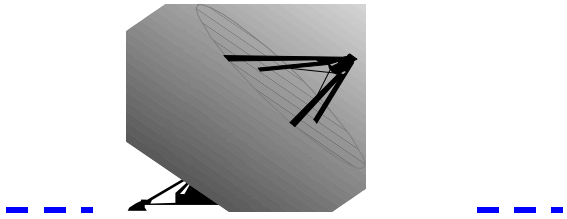
- Why do we need an Enterprise Service Bus?
  - Achieving loose coupling
  - Capabilities
- Making a Bus
  - Patterns
  - Technologies and products
- Real-life samples
- Managing an ESB

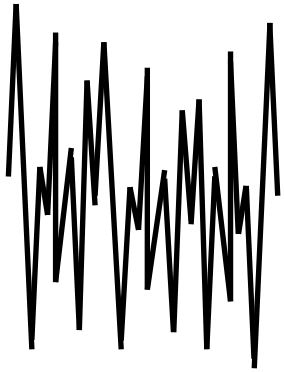
SOA Reference Architecture



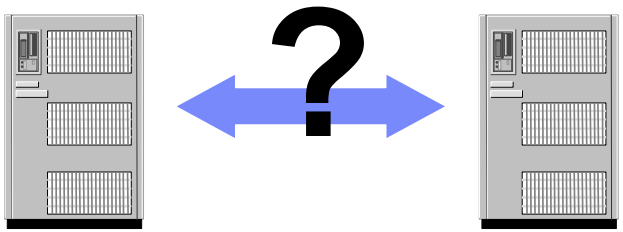
The "loose coupling" required by Service Oriented Architecture is not something I know how to implement for IT systems …

- *"Services are loosely coupled and invoked through communication protocols that stress location transparency and interoperability."*
- Coupling is a precisely defined term in the transmission of electric signals, but not in IT.
- Moving the receiver away from the transmitter reduces the coupling … and weakens the signal.



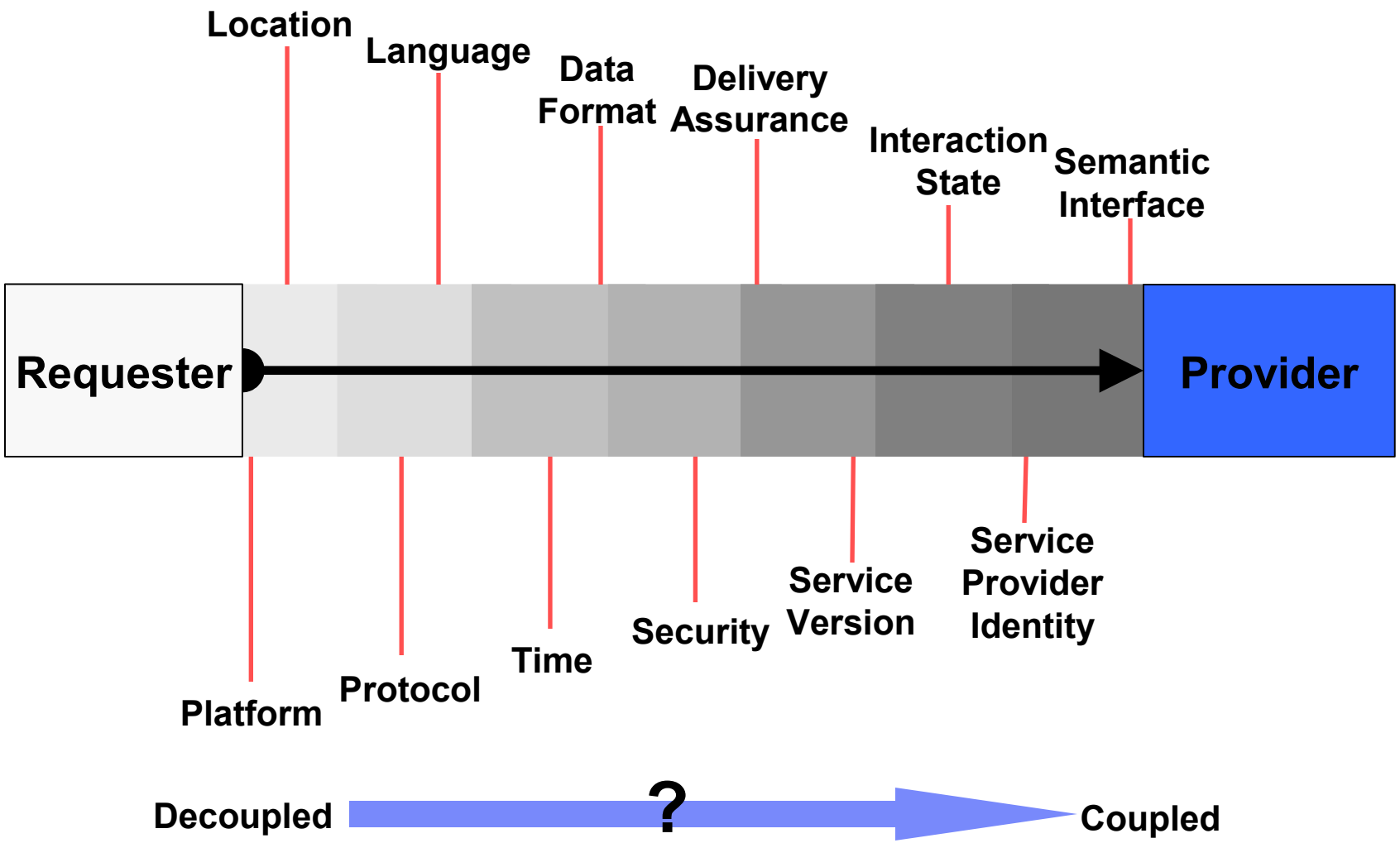


- *Question: how do I apply that idea to IT systems?*




```
<Book>
    <Title>War and Peace</Title>
    <Text>Some people lived in Russia for
        a while whilst international and
        domestic events took
        place.</Text>
</Book>
```

… but I can consider whether I would like to couple various *aspects* of service interactions …



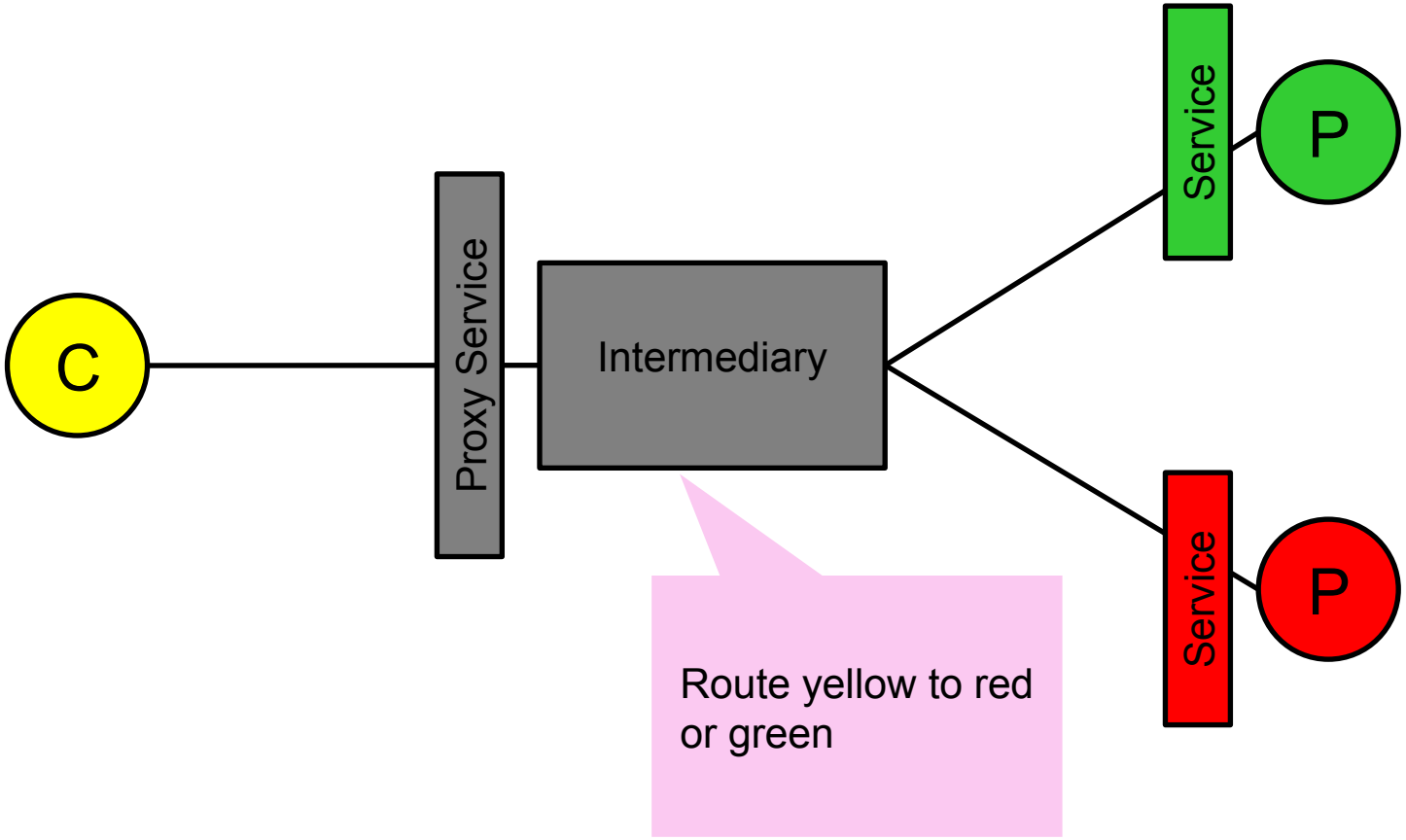
… and I can use more sophisticated *coupling styles* than "coupled" or "decoupled".

- **Coupled**
  - Directly manipulated by service requester and provider application code.
  - e.g. business data model

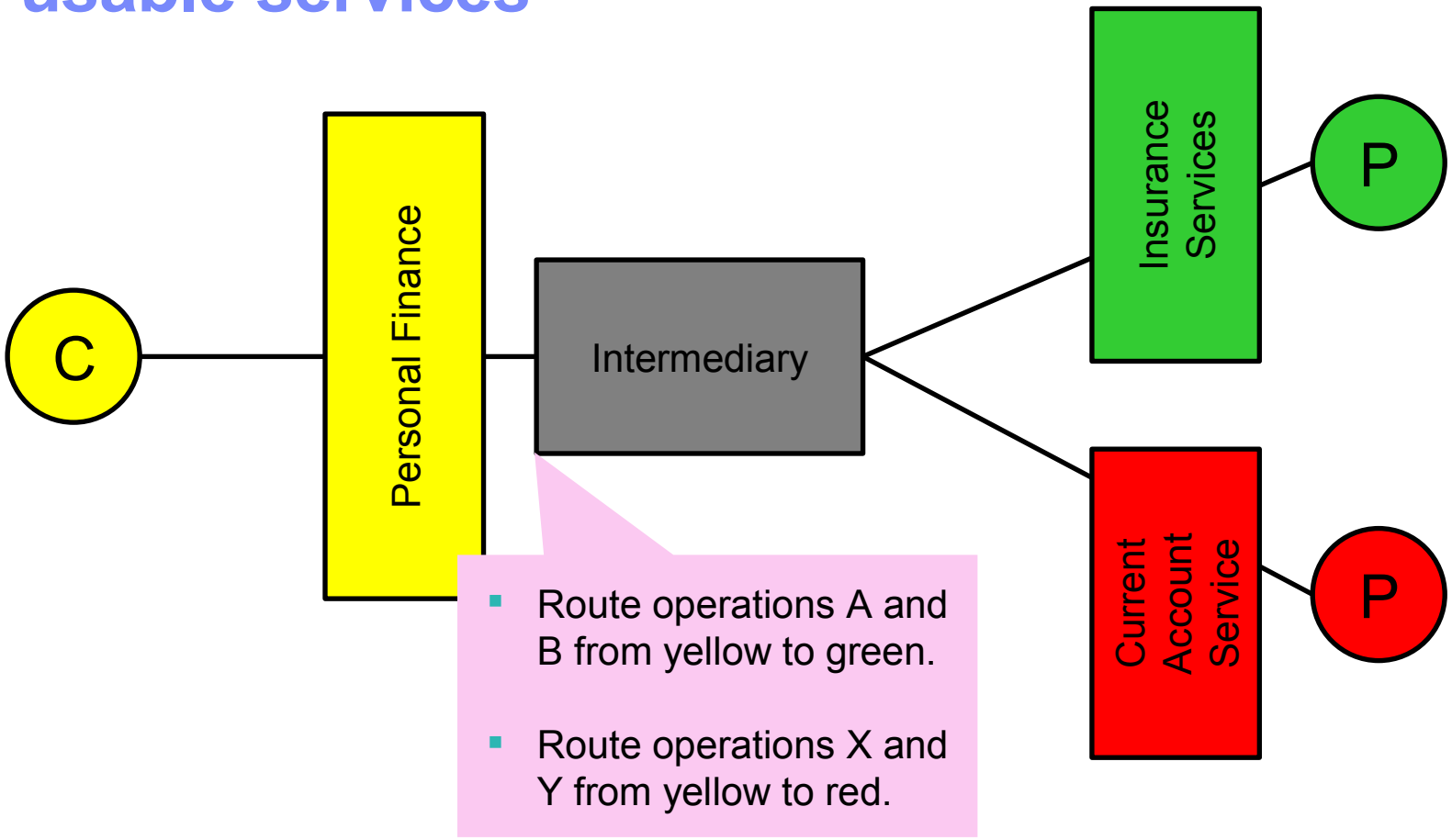
- **Declared**
  - Clients and providers declare matching behaviour in interfaces.
  - e.g. WS-Security
  - **Transformed**
    - Specified in the service interface, but not manipulated by application code. Services declare different characteristics but the service infrastructure can mediate between requester and provider.
    - e.g. data format
  - **Negotiated**
  - Requester and provider interfaces declare a spectrum of behaviours, infrastructure negotiates an agreed behaviour for each interaction.
  - e.g. proposed WS-Policy, service provider identified dynamically through UDDI

- **Decoupled**
  - Entirely independent between client, provider and infrastructure.
  - e.g. platform independence through use of XML and HTTP

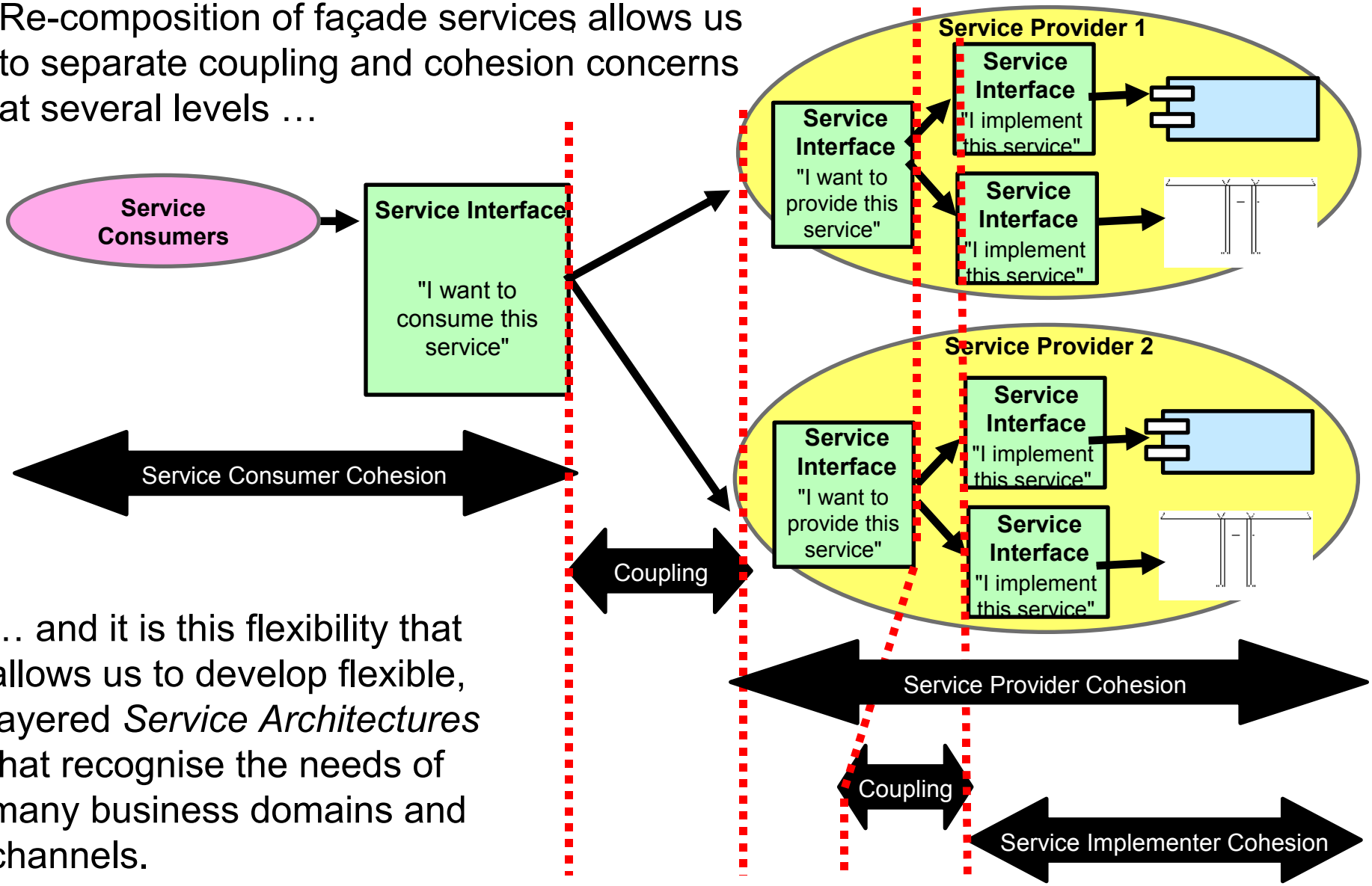
Proxy Services de-couple the Consumer from the identity of the Service Provider



Façade Services allow service consumers and service providers to have different ideas about re-usable services



Re-composition of façade services allows us to separate coupling and cohesion concerns at several levels …



… and it is this flexibility that allows us to develop flexible, layered *Service Architectures* that recognise the needs of many business domains and channels.

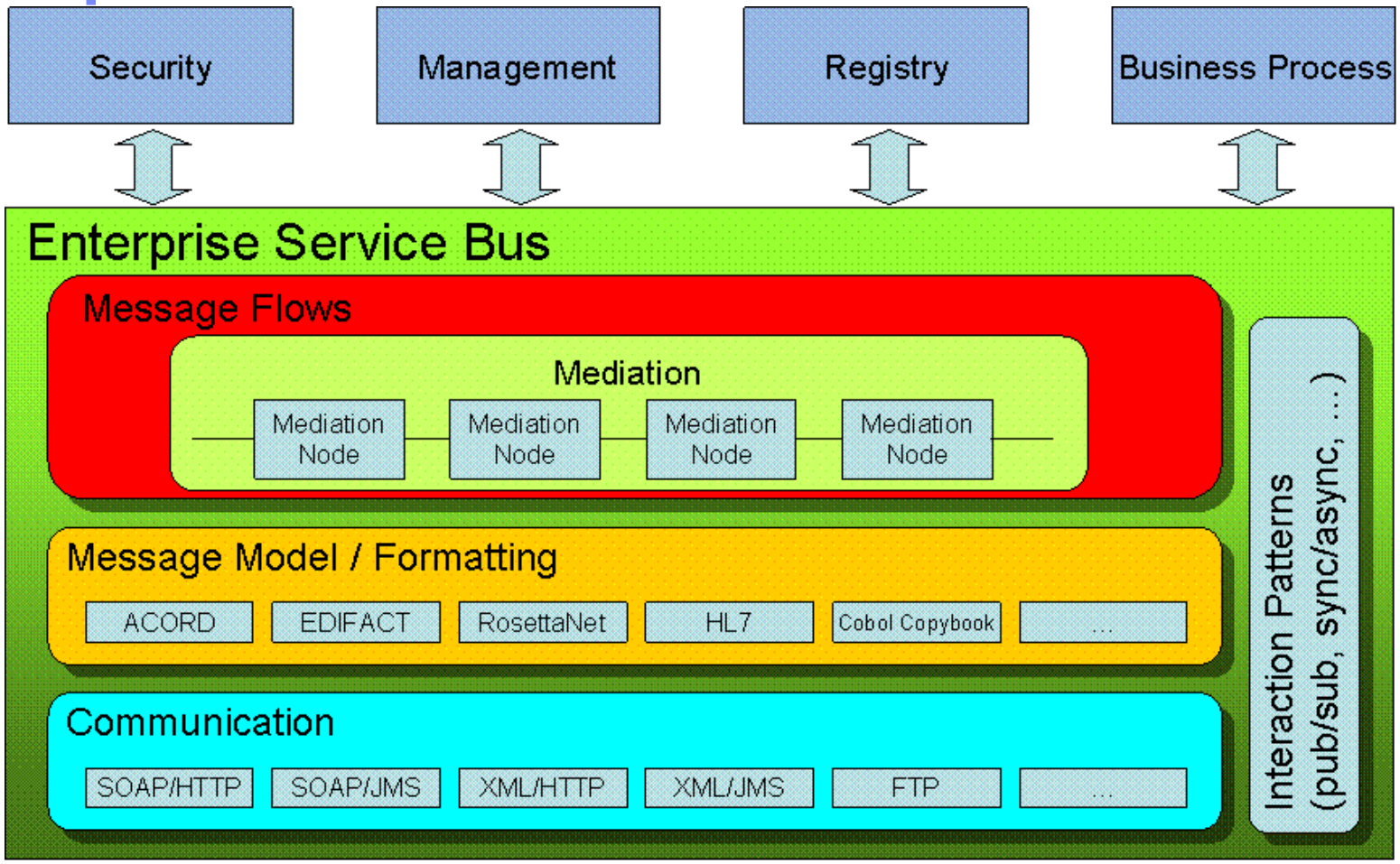
Agenda

- Why do we need an Enterprise Service Bus?
  - Achieving loose coupling
  - Capabilities
- Making a Bus
  - Patterns
  - Technologies and products
- Real-life samples
- Managing an ESB

The Patterns for e-business SOA Profile applies a common model of ESB capabilities to several uses and implementations of the ESB

Communications, e.g. ▪ Routing, addressing, protocols, pub/sub, async.	Service Interaction, e.g. ▪ Interface definition, service substitution, messaging model, SOAP, WSDL, directories
Integration, e.g. ▪ Database, legacy, middleware connectivity, service aggregation, app server connectivity, protocol transformation.	Quality of Service, e.g. ▪ Transactions, delivery assurance
Security, e.g. ▪ Authentication, authorisation, non-repudiation, confidentiality, standards support (WS-Security, Kerberos etc.)	Service Level, e.g. ▪ Performance, throughput, availability, scalability.
Message Processing, e.g. ▪ Encoded logic, content-based logic, message and data transformations, intermediaries etc..	Management and Autonomic, e.g. ▪ Service provisioning and registration, logging, metering, monitoring, systems management etc..
Modelling, e.g. ▪ Object modelling, common formats and libraries, public vs. private etc..	Infrastructure Intelligence, e.g. ▪ Business rules, policy driven behaviour, pattern recognition, etc..

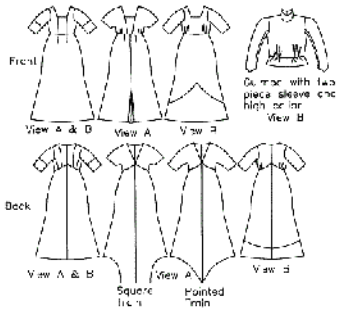
ESB Capability Model – an Architectural Viewpoint



Agenda

- Why do we need an Enterprise Service Bus?
  - Achieving loose coupling
  - Capabilities
- Making a Bus
  - Patterns
  - Technologies and products
- Real-life samples
- Managing an ESB

What is a pattern?



pat·tern (păt'ərn) n.

A model or original used as an archetype.
A person or thing considered worthy of imitation.
A plan, diagram, or model to be followed in making things: a dress pattern

Pattern Content

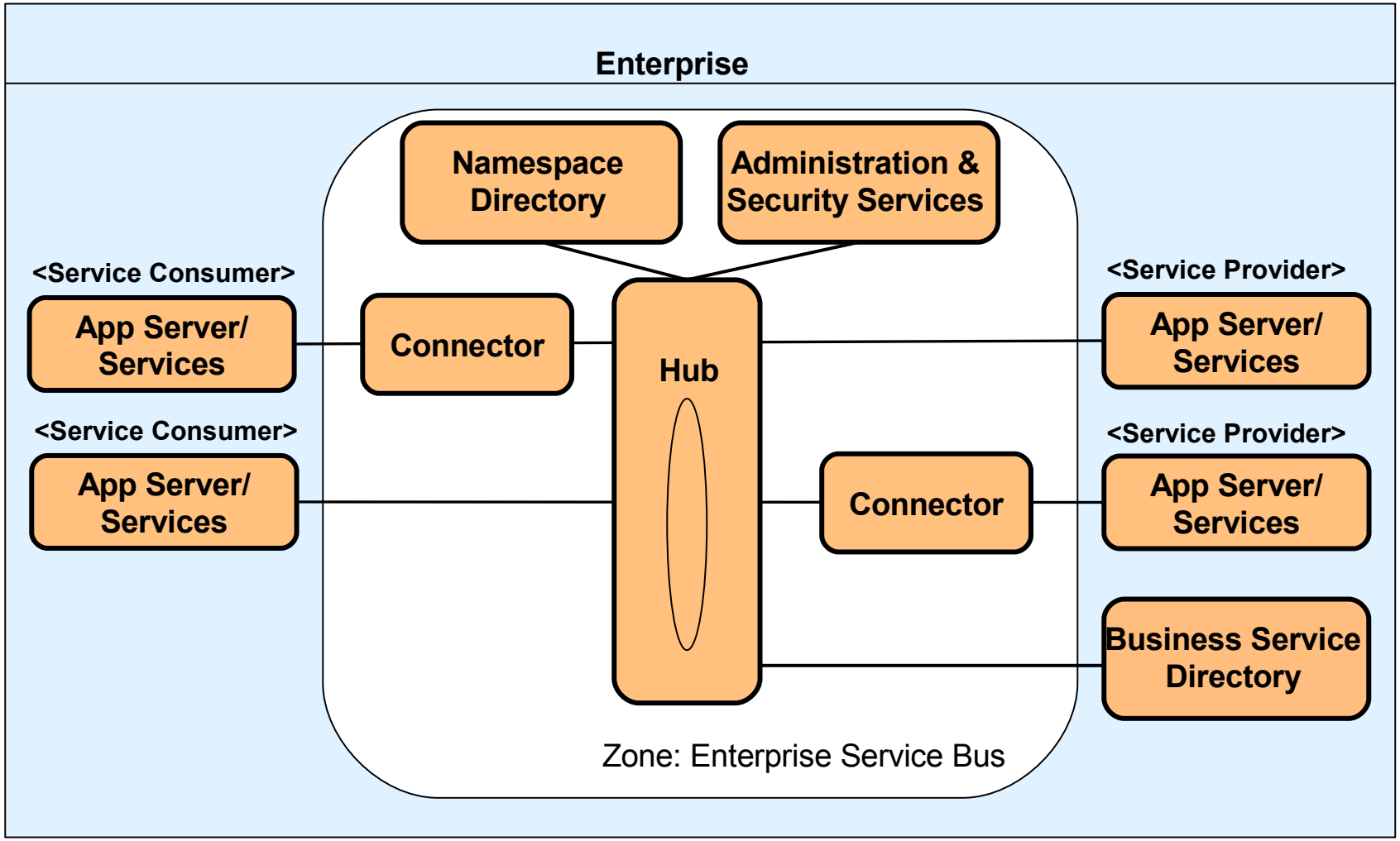
Pattern Authoring

Repository & Search

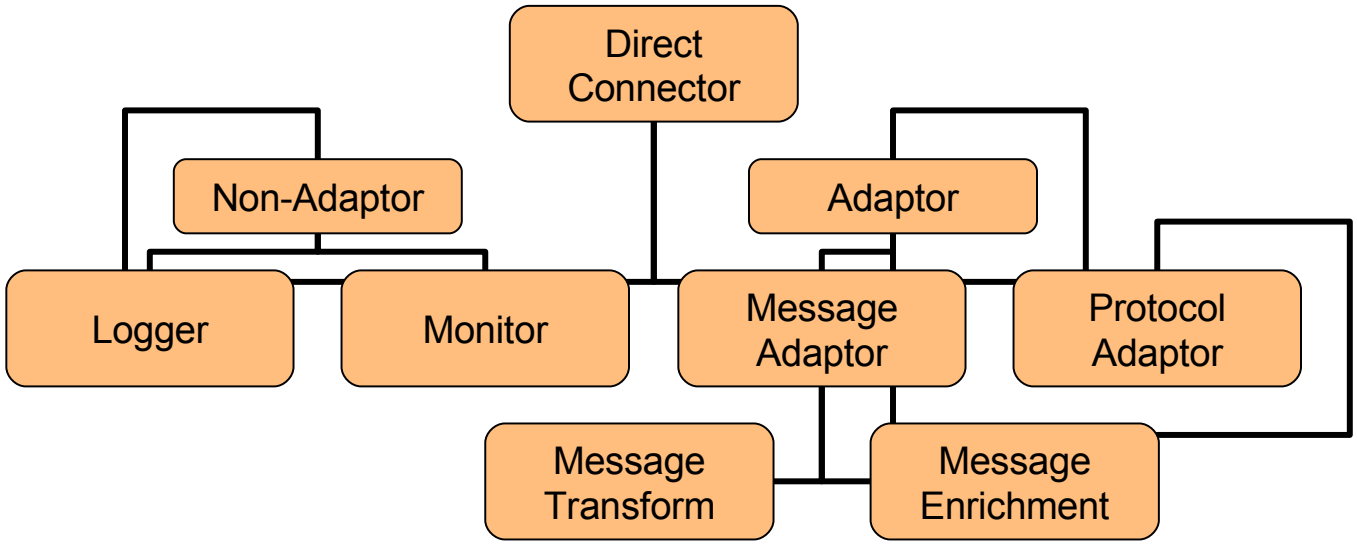
Applying Pattern

Patterns are reusable assets that describe solutions to a recurring problem
Patterns are discovered
- ✓ Someone identifies the recurrence of the problem and the solution
- ✓ Then formally documents the participants, their interactions and context

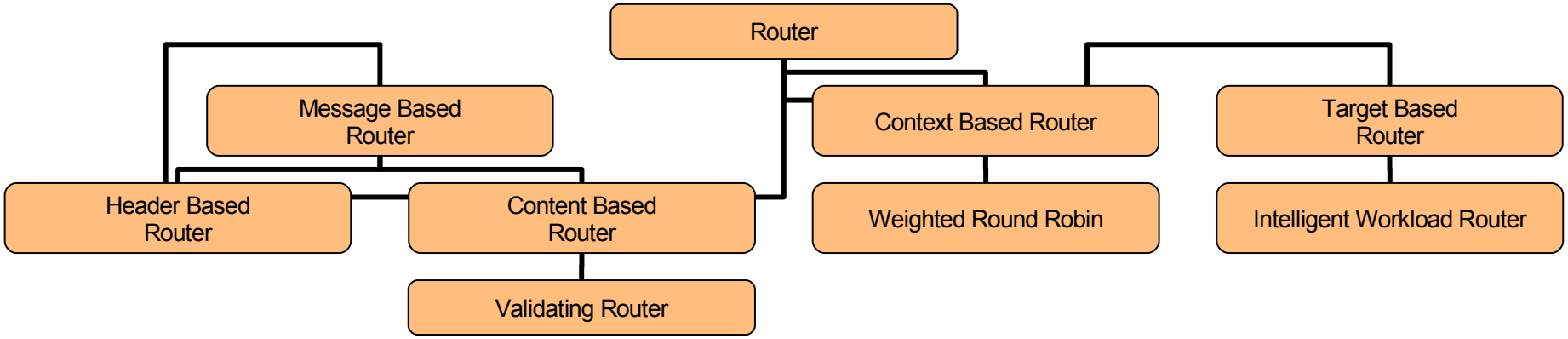
Enterprise Services Bus Runtime Pattern



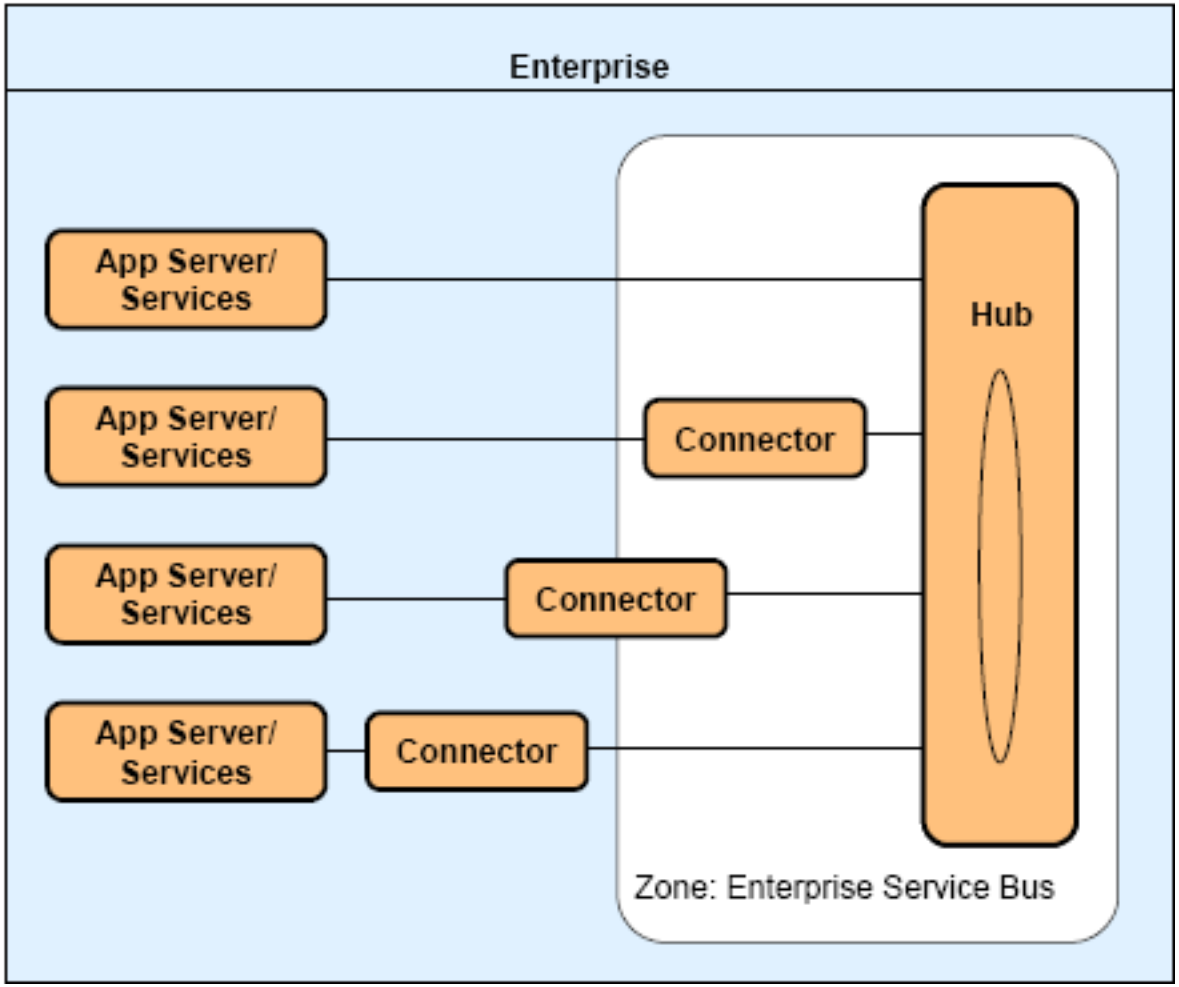
Specialising the Direct Connector Pattern



Specialising the Router Pattern



Connectors may generally be on or off the bus – don't waste time worrying about it if it is not obvious!



Agenda

- Why do we need an Enterprise Service Bus?
  - Achieving loose coupling
  - Capabilities
- Making a Bus
  - Patterns
  - Technologies and products
- Real-life samples
- Managing an ESB

First some product positioning

ESB:
WebSphere ESB

Advanced ESB:
WebSphere Message Broker

Web Services connectivity and data transformation

Universal connectivity and data transformation

HTTP **JMS**

WebSphere MQ

Web Services **XML**

WebSphere Adapters

HTTP **JMS** **WebSphere MQ**

Web Services **XML** **WebSphere Adapters**

Plus the following:

Weblogic JMS® Biztalk® TIBCO Rendezvous®

MQe Multicast Tuxedo® FTP TIBCO EMS JMS®

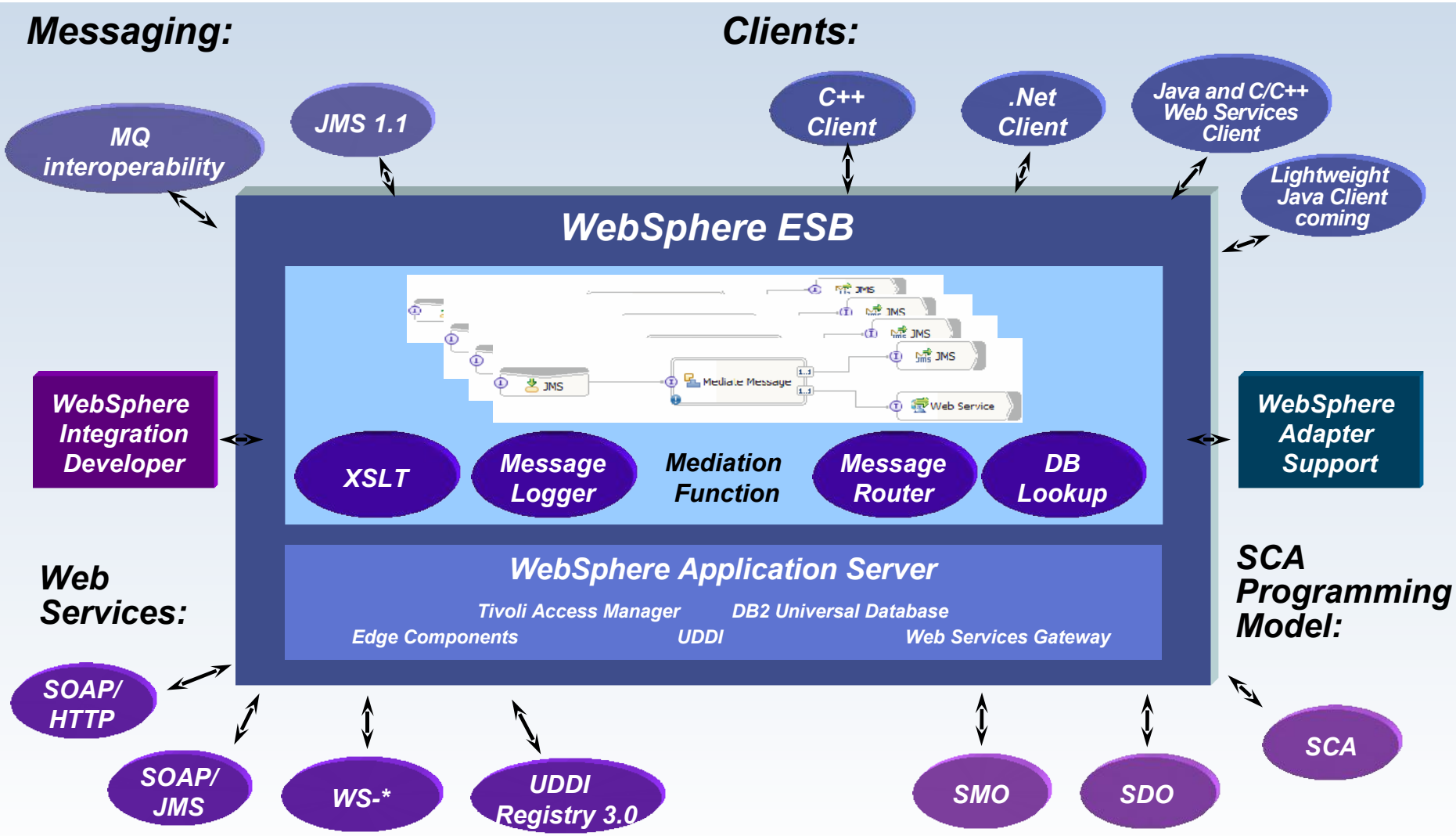
COBOL Copybook HIPAA EDI-FACT HL7 SonicMQ JMS®

ACORD Real-time IP AL3 Word/Excel/PDF

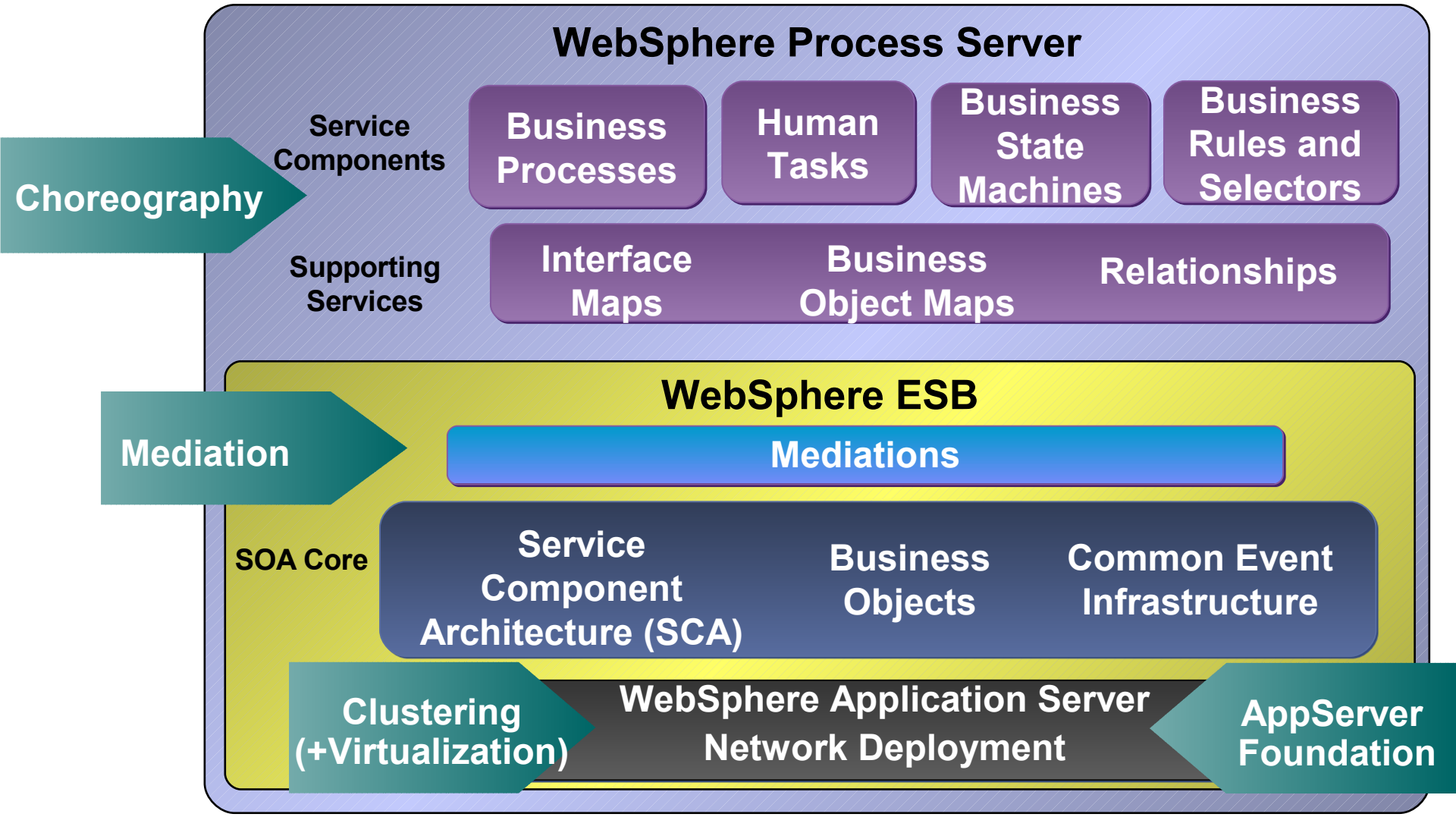
SWIFT FIX ebXML EDI-X.12 MQTT Custom Formats

Customers face a range of ESB requirements.
As a result, any given project might require an ESB or an Advanced ESB… OR BOTH.

WebSphere ESB at a glance



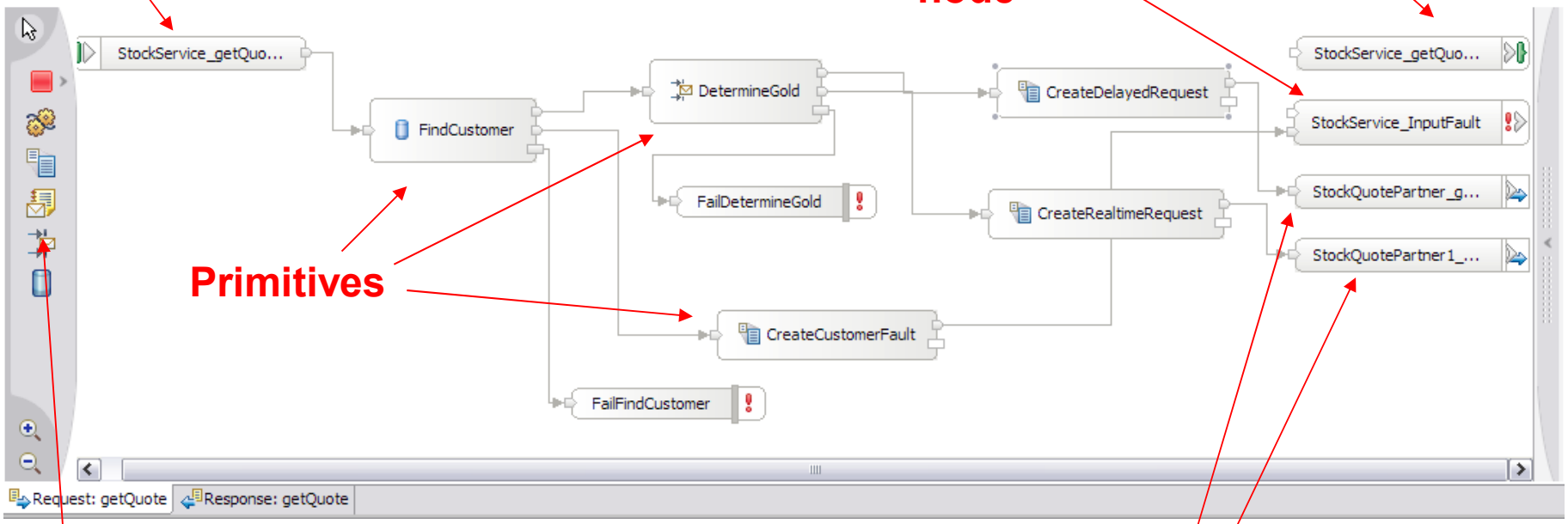
WebSphere Integration Family



WebSphere ESB - Mediation Primitives

Mediation Primitives	Symbol	Description
Message Logger		To log/store message information to a database
Message Filter		To filter messages selectively forwarding them on to output terminals, based on simple condition expression
Database Lookup		To access information in a database and store it in the message
XSLT		To manipulate or transform messages using XSL transformation
Stop		To stop a path in the flow, without generating an exception
Fail		To stop a path in the flow, and generate an exception
Custom		For custom processing of message. It uses a SCA Java component for custom message processing.

WebSphere Integration Developer- Mediation Flow Editor – Flow View -



Input operation

Input fault node

Input response node

Primitives

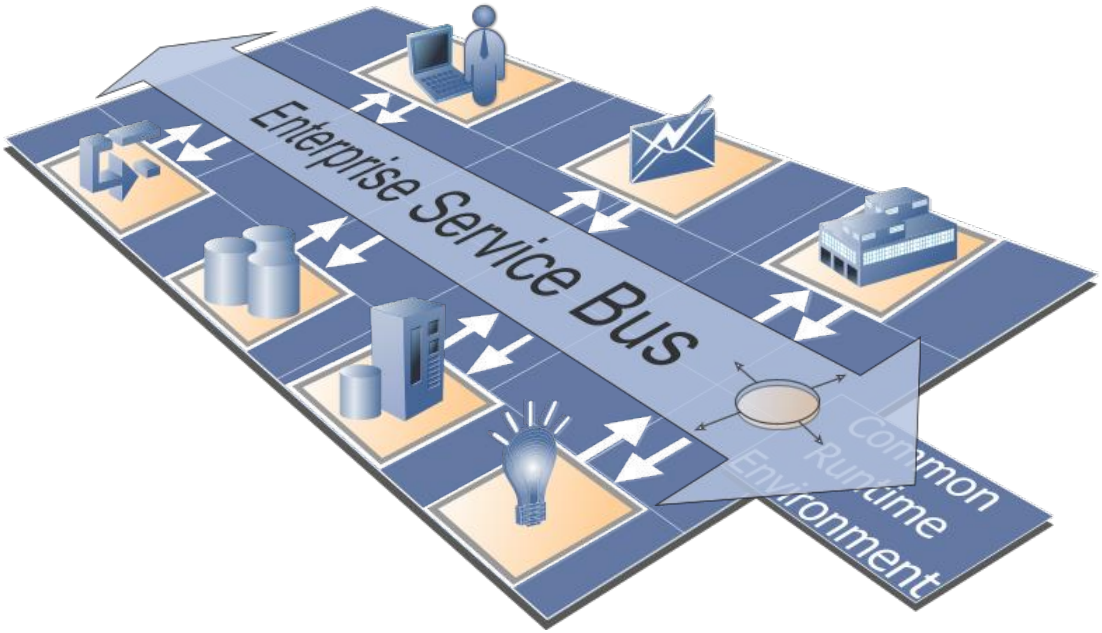
Callouts

Primitive palette

Response flow tab

WebSphere Message Broker (WMB) - Creating an Application Integrator

To create an application integrator requires the ability to join together all applications in your enterprise. Note that this is NOT the same application serving; it is business integration. WebSphere Brokers provide message processing integration hubs on the ESB.



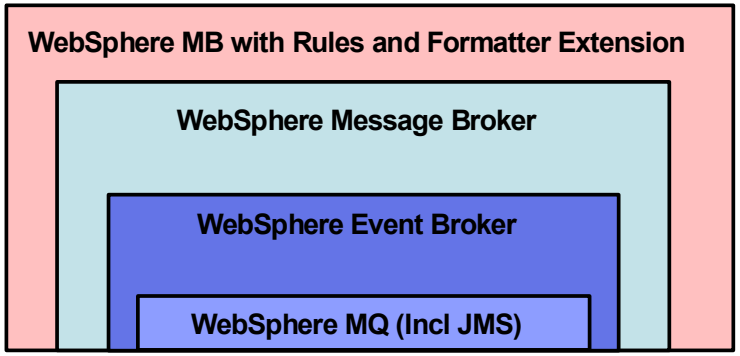
- ✓Join Application and Information sources
- ✓Heterogeneous and decoupled
- ✓Data transform
- ✓Data routing
- ✓DBMS Integration
- ✓Transactional
- ✓Advanced tooling

- ✓Simple
- ✓Extensible
- ✓Standards based

Transformation

Intelligent routing

Business Rules



WMB - Inherent Strength Example

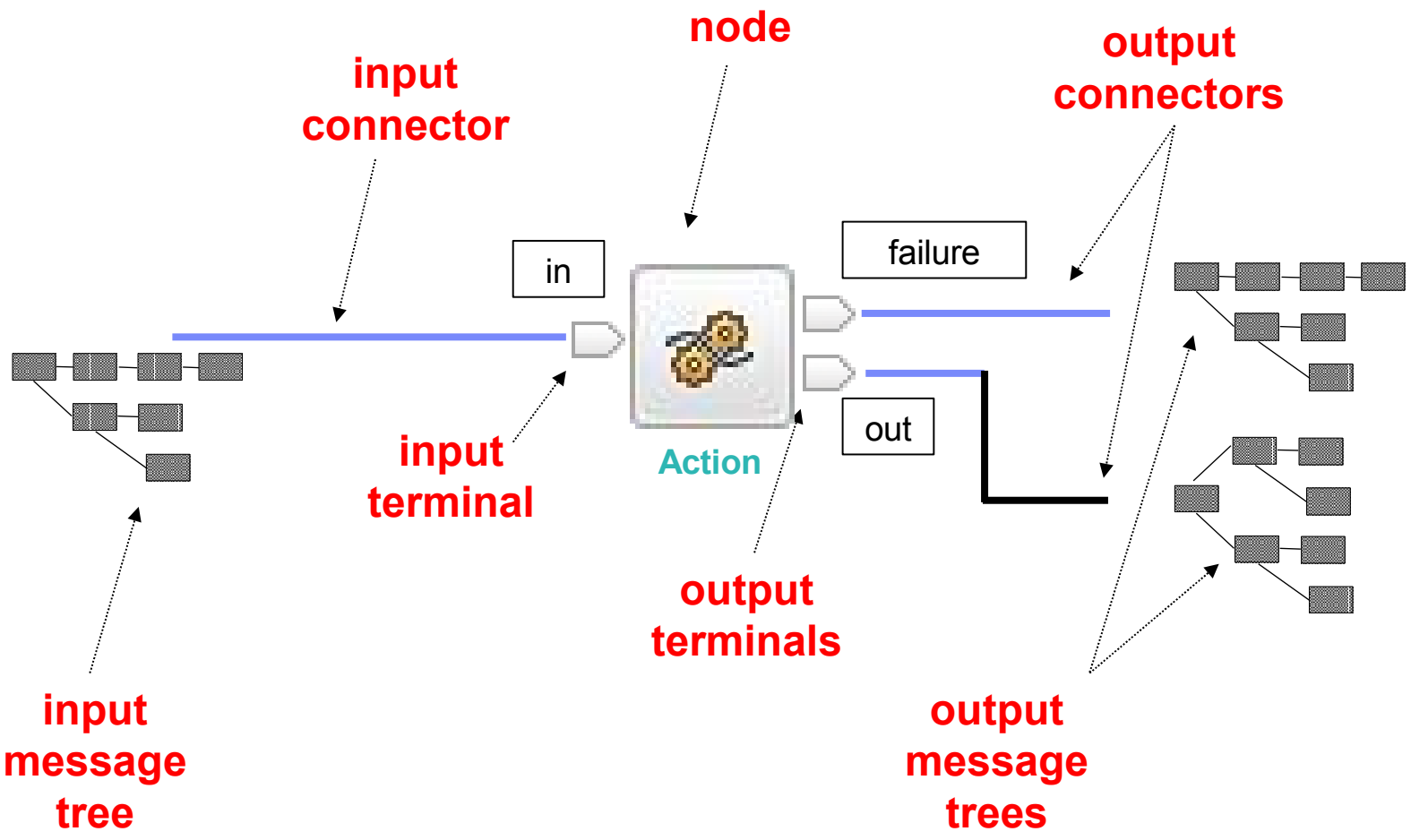
- WMB drives the overall solution – hands off to WPS when needed
- Implementation choices based on products' inherent capabilities
  - WMB handles pub/sub
  - WMB handles transformation
    - XML -> SOAP/HTTP (.NET)
    - XML -> COBOL fixed-format
  - WMB handles ASCII <-> EBCDIC conversion
  - WMB handles data warehousing
  - WMB handles routing

When to use WMB

- Routing
- Transformation
- Reusable message models
- High performance
- Stored procedures in COBOL, C
- Industry-standard formats
- Fixed-format binary data
- Formatted text
- Multiple-transports in same flow
- Complex pub/sub (e.g. content-based filtering, topic security)

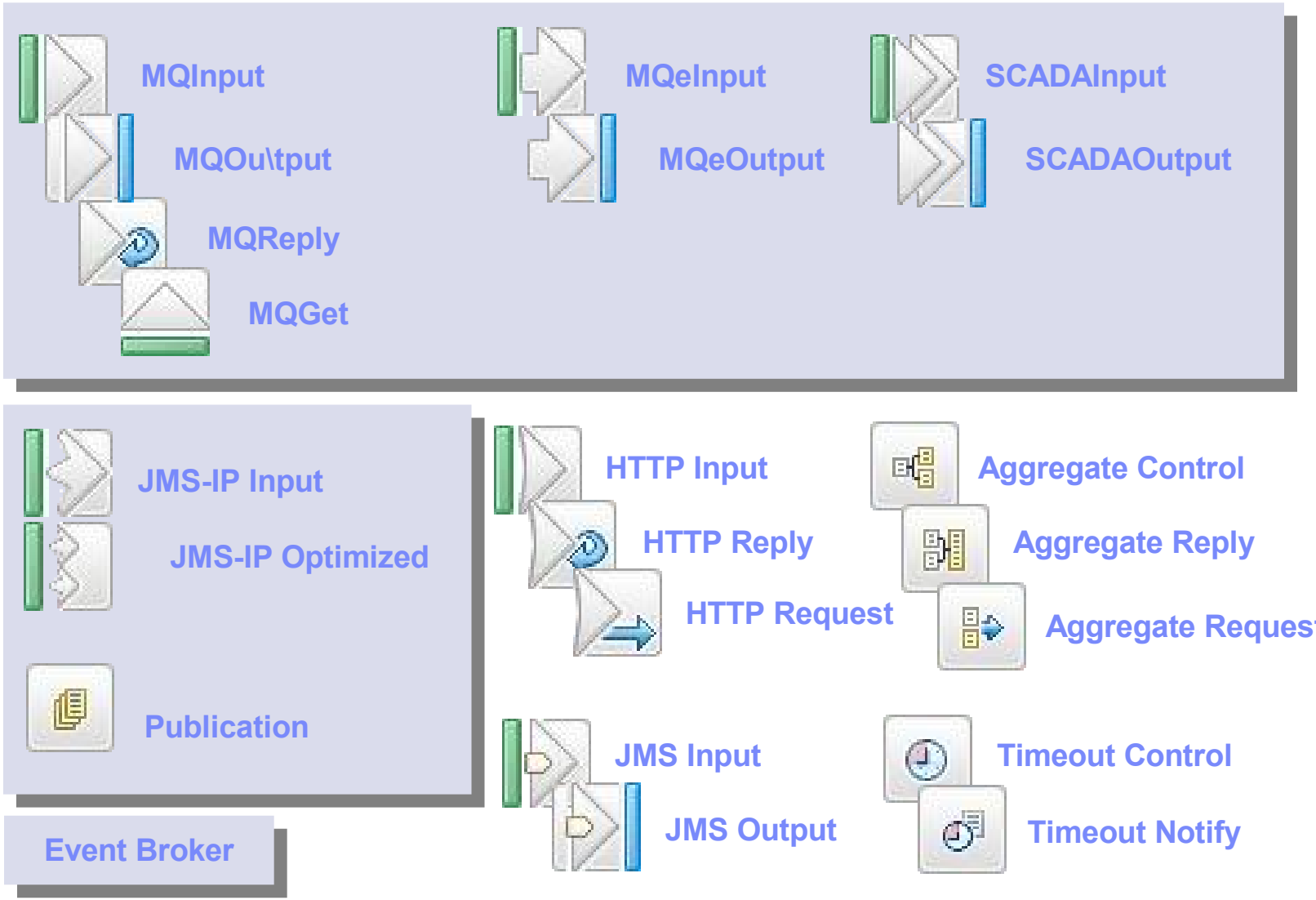
WMB – Message Processing Nodes

The Message processing node is the next object type to understand. There are many nodes, which both transform and route messages; they are the atoms of message integrations. They are combined to form message flows.

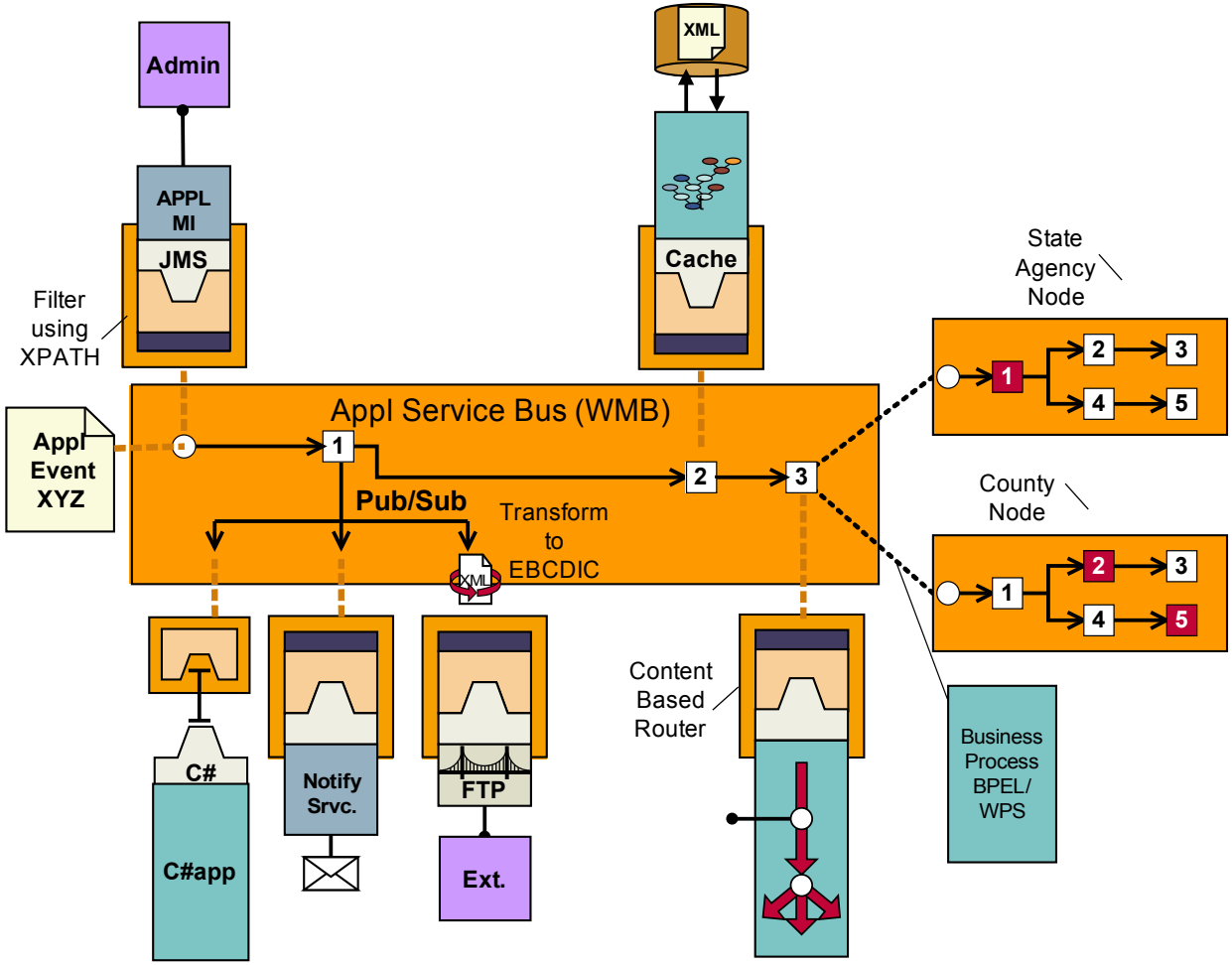


WMB - Built-in Nodes

The broker comes with a built-in set of nodes that form an integration "Starter Pack." You can see that many of the most common transport protocols are supported, enabling you to get 'out of the box' quickly.



WMB - Inherent Strength



Agenda

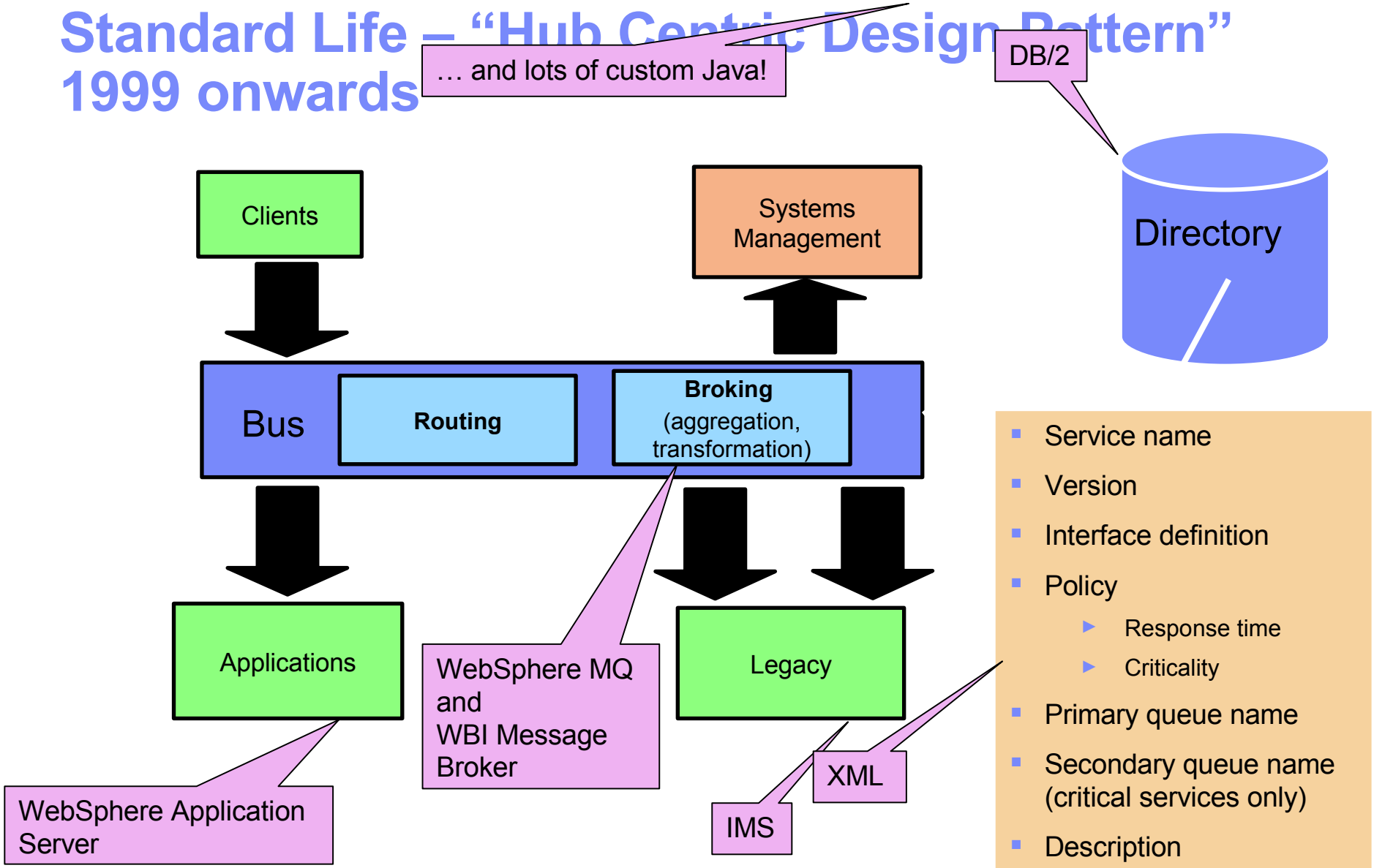
- Why do we need an Enterprise Service Bus?
  - Achieving loose coupling
  - Capabilities
- Making a Bus
  - Patterns
  - Technologies and products
- Real-life samples
- Managing an ESB

Customer scenario

Standard Life

The ESB enables re-use and flexibility in the enterprise

Standard Life – "Hub Centric Design Pattern" 1999 onwards



- Service name
- Version
- Interface definition
- Policy
  - Response time
  - Criticality
- Primary queue name
- Secondary queue name (critical services only)
- Description
- Error messages

Monitoring the infrastructure in context of the service model

Impact Analysis Browser

UID sl577h Password ******* Environment: PROD Connect Period: 200501

Application View | Component View | Development view

Application All

Business Services

Name [ID]		
MaintainRequestOutgoingDocumentation[1]	74356	1
MaintainRequestOutgoingDocumentation[]	74356	1
MaintainRequestOutgoingDocumentation[8]	74356	1
ProvideCustomerInfo[8]	58332	7
ProvideCustomerInfo[7]	58332	7
ProvideCustomerSearchResults[7]	36223	3
MaintainLpsProposal[1]	35374	3
ProvidePlanList[7]	28029	7
ProvidePlanList[8]	28029	7
ProvidePensionInvestmentInfo[8]	27211	7
ProvidePensionInvestmentInfo[7]	27211	7
ProvidePensionPolicyInfo[8]	22994	5
ProvidePensionPolicyInfo[7]	22994	5
ProvideEmploymentInfo[8]	20725	3
ProvideEmploymentInfo[7]	20725	3

Business Services

Data Services

Name [ID]		pp#
AIADRC[AI100]		1
BQOT0BS[BQ071]	3082	1
BQOT0BT[BQ072]	805	1
BQOT0BU[BQ073]	4818	1
CUST00A[CU005]	671	3
CUST0AL[CU043]	669	3
CUSTESU[CU604]	1834	3
CUSTGAR[CU602]	4051	2
CUSTGVR[CU603]	5272	3
CUSTMLU[CU601]	638	3

Data Services

Used By / Uses | Statistics | Impact View

Service Name MaintainRequestOutgoingDocumentation

Layer BSL ID 1

Total used by hitcount 74356

Directly Used by

Application	HitCount	Caller
TSPDD	4908	DrawdownQuotesACL
TSPDD	19084	MaintainRequestOutgoingDocumentation
TSPDD	50168	RodAcl
TSPDD	104	ServiceDrawdown
TSPDD	92	SetupDrawdown

Applications

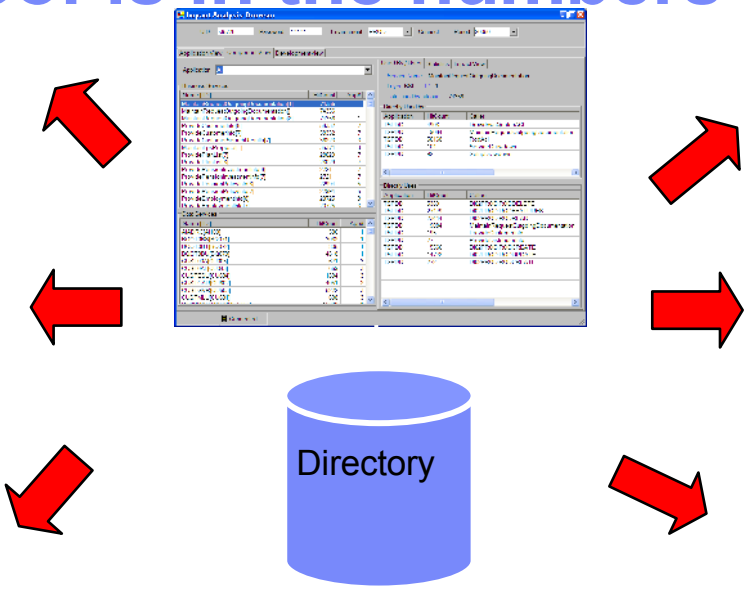
Directly Uses

Application	HitCount	Callee
TSPDD	5389	DI02PROD.RODDELETE
TSPDD	27479	DI02PROD.RODFEATURES
TSPDD	23444	DI02PROD.RODREAD
TSPDD	19084	MaintainRequestOutgoingDocumentation
TSPDD	195	ProvideCustomerInfo
TSPDD	77	ProvideDistributorInfo
TSPDD	15566	DI02PROD.RODCREATE
TSPDD	14778	DI02PROD.RODUPDATE
TSPDD	272	DI02PROD.RODCREATE

Transactions

Connected

The proof is in the numbers



- 400,000 Servlet invocations per day
- 800,000 EJB invocations per day
- 1,600,000 IMS Transactions per day
- 5 p690 CPUs in production
- 1.3 CPU p.a. expected growth

Standard life will tell almost anyone almost of all this …

… but they won't tell anyone about the development process and tools by which re-use is enforced and achieved

Customer scenario

Case Study: Financial Services Provider

Autonomic ESB Proof of Technology

Customer Background

- Sell services and information to finance and media organisations
- Sell approximately 2,500 products
  - e.g. Stock quotations and sales
  - e.g. Corporate reports
  - e.g. "Access to financial marketplace data in Europe with 20 minutes latency"
- Have grown rapidly through both in-house business development and acquisition
- Should be a good candidate to exploit SOA …
  - Information-based business
  - Technology business with a strong sense of IT strategy
  - Responding to a changing market requires not only new services, but the ability to quickly configure new combinations of services and service levels in competitive packages
- … but their technology is an inhibitor to the flexibility and performance their customers demand …

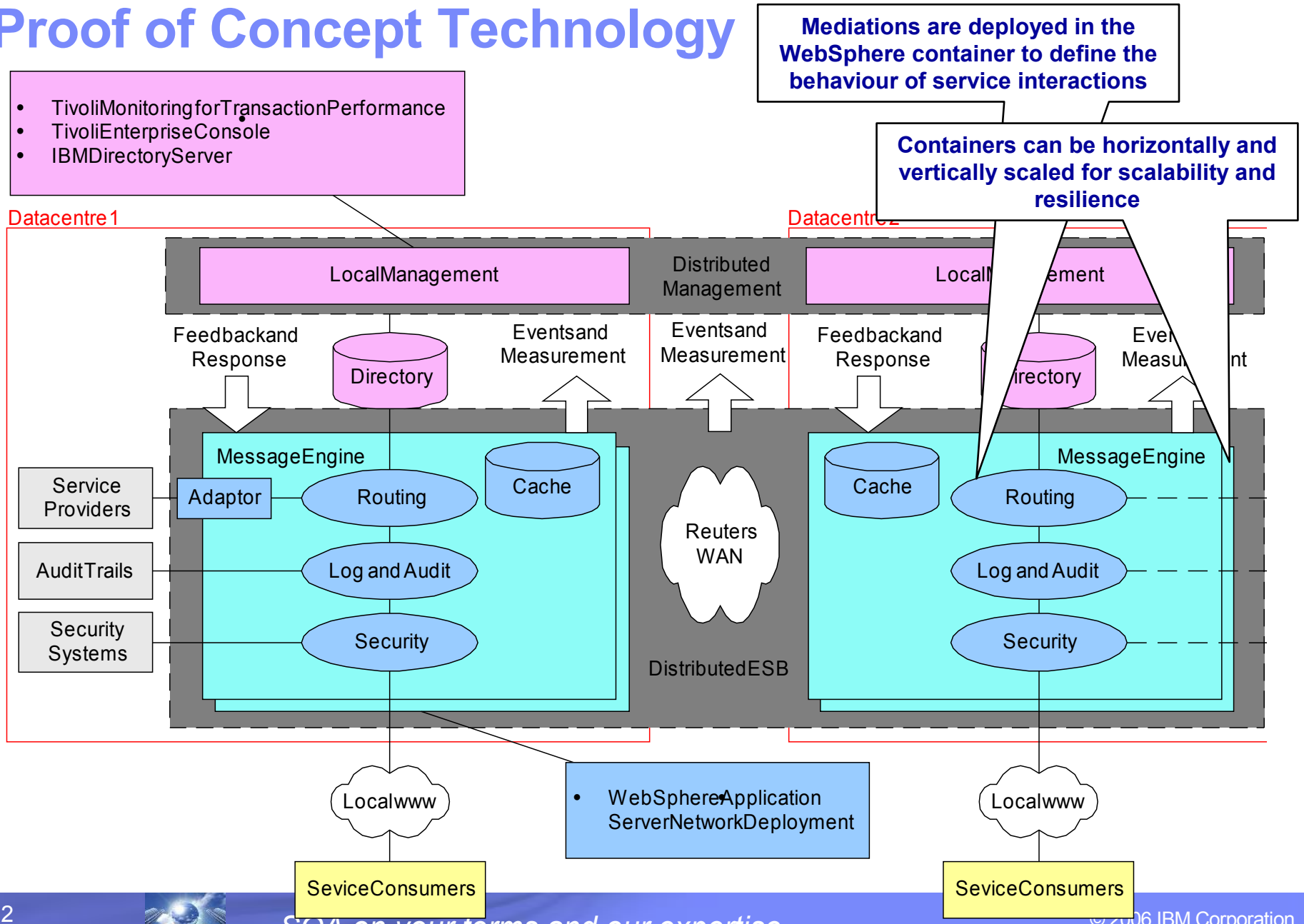
Management and Autonomic Requirements

- **Integrate with Tivoli deployment to provide "joined-up" management of *infrastructure, applications* and *services* (i.e. *products*).**
- **Support drill-down into underlying measurements to diagnose failures / deteriorations**
- **Support predictive alerting in context to provide product / service based alerts in response to lower level events**
- **Perform intelligent workload management to make efficient use of ESB, service and network capacity to support prioritised service requests.**
- **Define quality of service behaviour through the use of policies associated with services by configuration**
- **Control ESB routing, messaging and QoS behaviour (e.g. retries) through policy and with input from Systems Management**

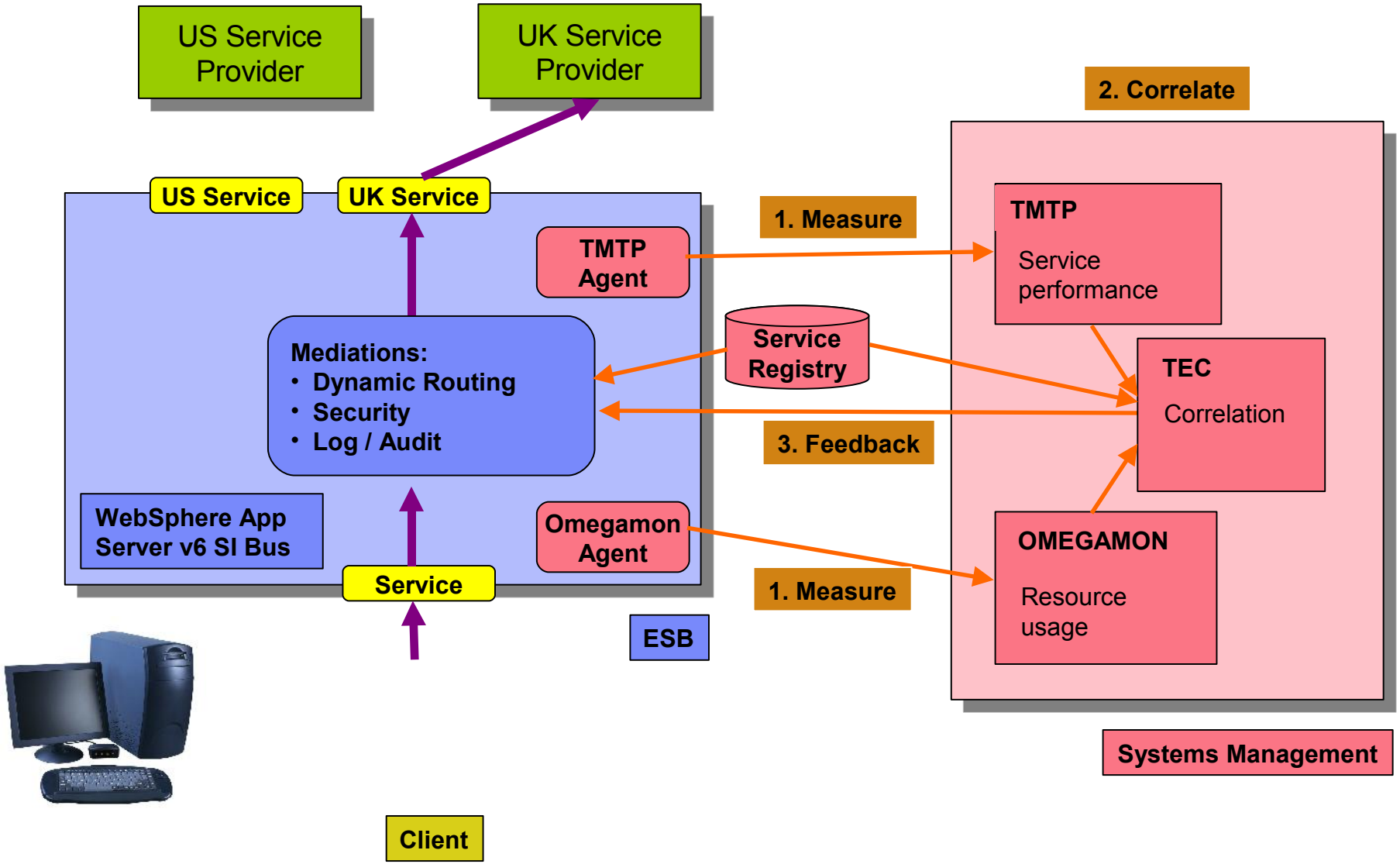
Example Proof of Concept Scenario

- The customer's service provider is available in UK and US data centres
- Two levels of service are sold to customers – "Gold" and "Silver"
- In normal circumstances, all UK customer requests are routed to the UK datacentre
- If the response time delivered by the UK provider falls below a certain threshold:
  - "Gold" requests are re-routed to the US provider in order to maintain service levels.
  - "Silver" requests remain routed to the UK provider and are delivered with a deteriorated service level.
- If the CPU threshold delivered by the UK provider also rises above a certain threshold:
  - "Silver" requests are now routed to the US provider to reduce the strain
- If the Wide Area Network load breaches a certain threshold
  - "Silver" requests are blocked from routing through the Wide Area Network to preserve remaining capacity for "Gold" customers

Proof of Concept Technology



Service management, event correlation and feedback



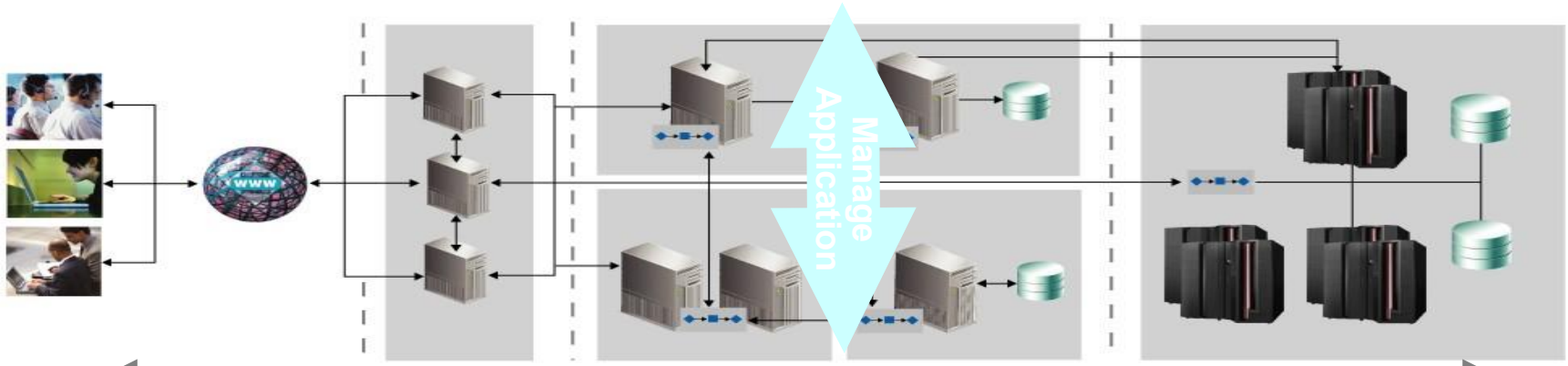
Agenda

- Why do we need an Enterprise Service Bus?
  - Achieving loose coupling
  - Capabilities
- Making a Bus
  - Patterns
  - Technologies and products
- Real-life samples
- Managing an ESB

The ITCAM Solution Portfolio

Three techniques tailored for different IT roles to effectively sense and respond to performance and availability events.

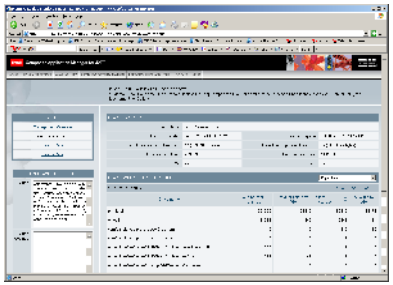
Analyze and Measure Transactions & Services



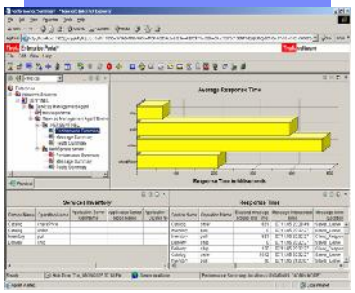
Monitor Infrastructure

ITCAM for RTT

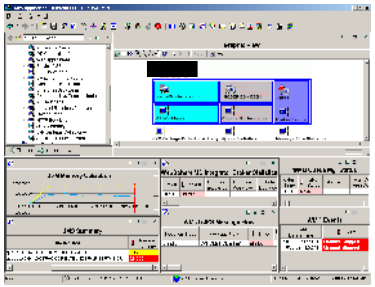
ITCAM for WebSphere



ITCAM for SOA



OMEGAMON XE for WBI



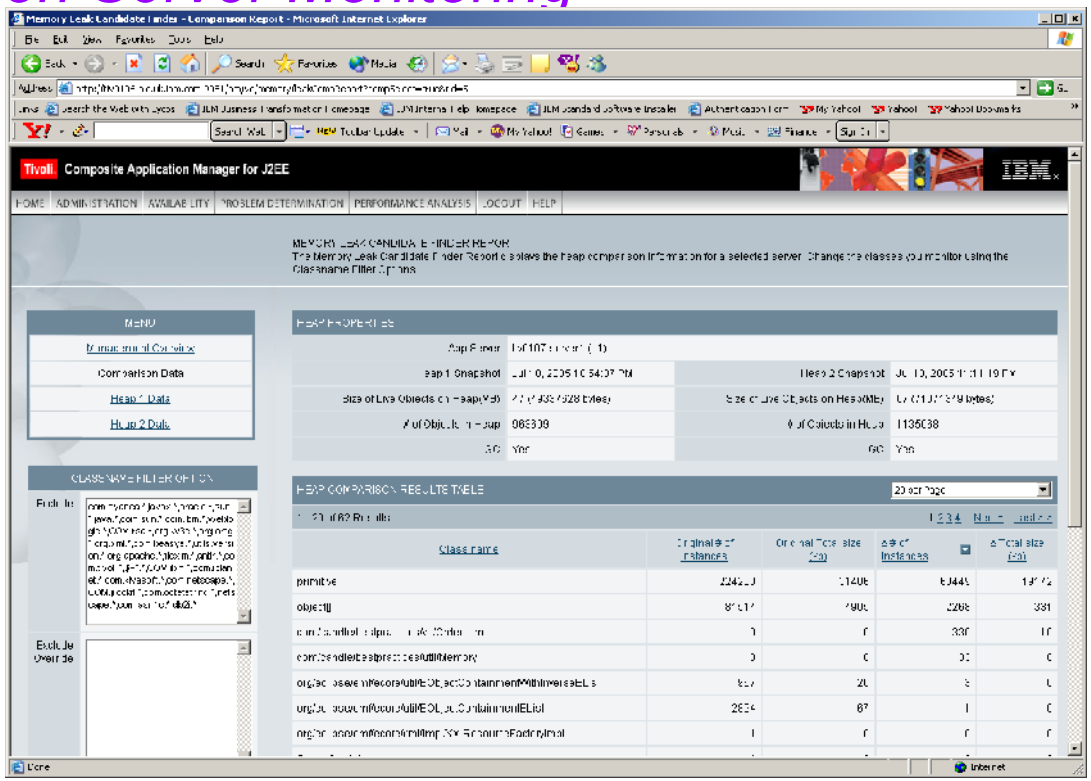
ITCAM for WebSphere v6.0 provides Comprehensive Deep-dive WebSphere Application Server Monitoring

Problem

- "The application slows down and then crashes after 3 days of uptime. I know I have a memory problem but I don't know where to begin to look to see what is causing the problem!"

Solution

- ITCAM for WebSphere provides comprehensive memory heap analysis that can tell you have a suspected memory leak in your application, as well as which class and source line is the cause of the leak!



Value

- ITCAM for WebSphere can significantly improve the performance and availability of your web application by reducing problem identification to resolution times

Conclusion / Wrap-up

- We have seen the role of the ESB in an SOA
- What makes SOA different from a standard EAI
- The capabilities of W-ESB and WMB
- ESB in action from customer projects
- How to monitor an SOA and its ESB

References: the SOA Profile in the Patterns for e-business

- **The IBM Patterns for e-business website:**
  http://www.ibm.com/developer/patterns

- **Patterns: Service-Oriented Architecture and Web Services, SG246303**
  http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/sg246303.html?Open
- **Patterns: Implementing a Service Oriented Architecture using an Enterprise Service Bus, SG246346**
  http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/sg246346.html?Open
- **Patterns: SOA with an Enterprise Service Bus in WebSphere Application Server V6, SG24-6494:**
  http://www.redbooks.ibm.com/abstracts/sg246494.html?Open
- **Patterns: Using Business Service Choreography In Conjunction With An Enterprise Service Bus, REDP3908-00**
  http://www.redbooks.ibm.com/abstracts/redp3908.html?Open
- **Using Message Sets in WebSphere Business Integration Message Broker to Implement an ESB in an SOA, REDP3978-00**
  http://www.redbooks.ibm.com/abstracts/redp3978.html?Open
- **Patterns: Serial and Parallel Processes for Process Choreography and Workflow, SG24-6306**
  http://www.redbooks.ibm.com/abstracts/sg246306.html?Open
- **Patterns: Implementing Self-Service in an SOA Environment, SG24-6680-00**
  http://www.redbooks.ibm.com/abstracts/sg246680.html?Open

End Of Presentation

Thank You
United Kingdom

Obrigado
Portugal

Dziekuje
Poland

Dankschen
Austria

Thanks
United States

Takk
Norway

Toda
Israel

Gracias
Spain

Danke
Germany

Bedankt
Netherlands

Tak
Denmark

Dekuju
Czech Republic

Merci
France

Engraziel
Switzerland

Tesekkür ederim
Turkey

Tack
Sweden

Dank u
Belgium

Thank You
United Kingdom

Grazie
Italy

Jag tackar
Finland

Dakujem
Slovakia

Спасибо
Russia