
Service Oriented Computing: the Challenges and Opportunities in e- Business Development



Jian Yang
Department of Computing
Macquarie University

Agenda

- The world of e-Business: definitions and characteristics
- Service oriented computing (SOC): a new paradigm?
- When SOC meets e-business
- Looking forward

Types of e-Business

- ◆ **Business-to-Consumer**
 - **E-commerce between organizations and individual consumers**

- ◆ **Business-to-business (e-Business)**
 - **E-commerce between businesses**
 - **Accounts for a much larger portion of e-commerce than business-to-consumer.**

- ◆ **Business-to-administration**

- ◆ **Consumer-to-administration**

◆ *Electronic-Business:*

The conduct of transactions by means of electronic communications networks (e.g., via the Internet and/or possibly private networks) end-to-end.

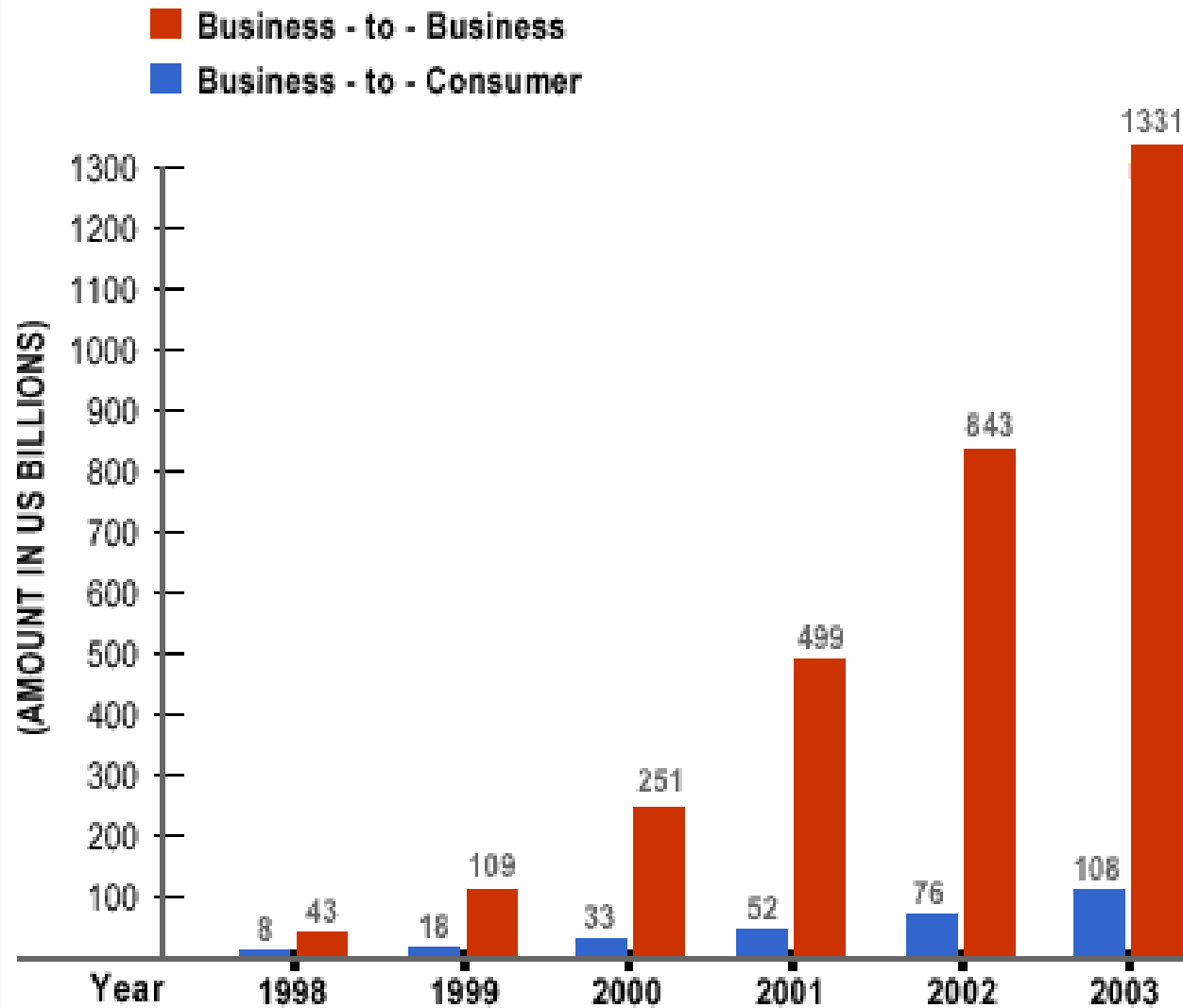
■

e-Business vs. e-Commerce

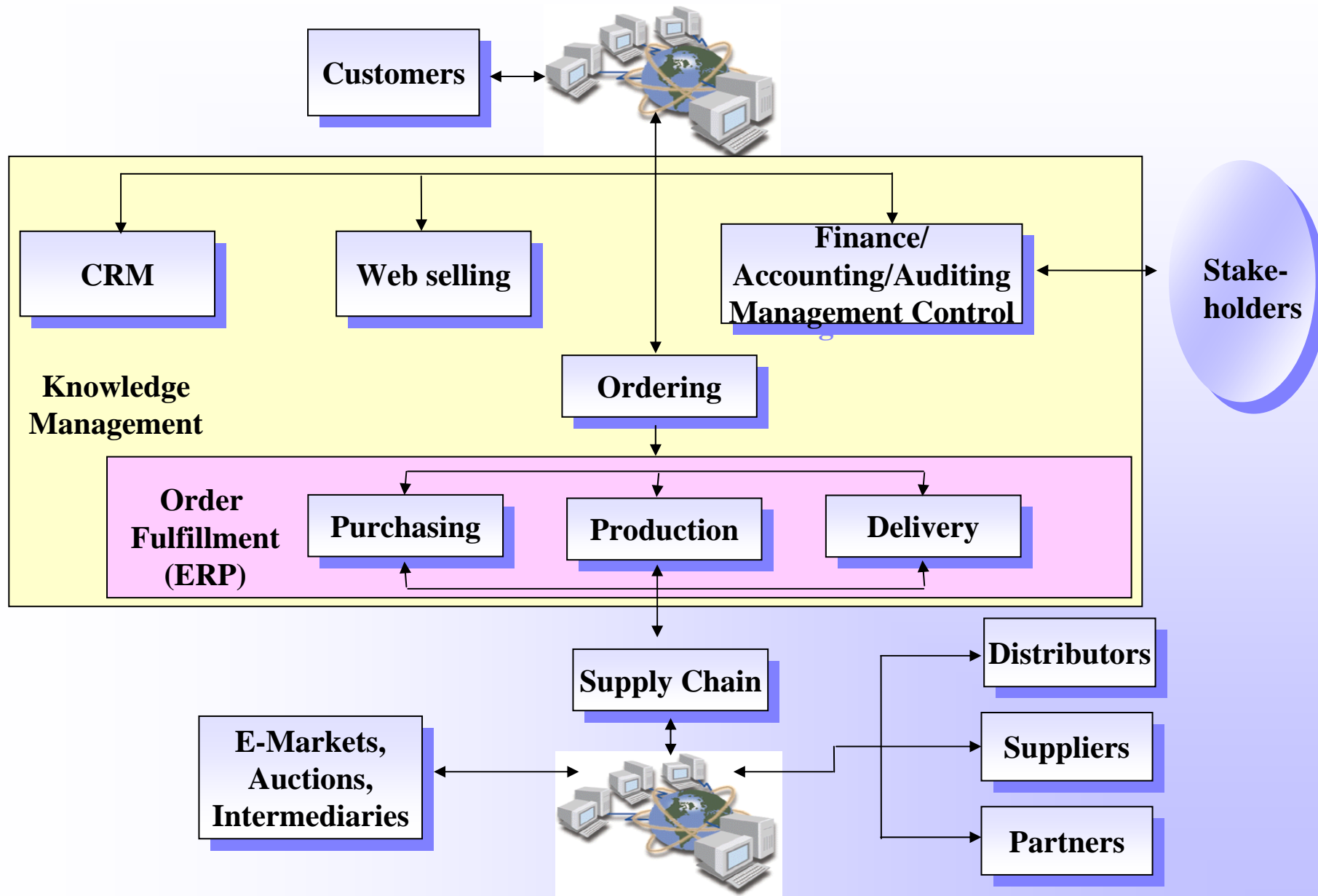
- Compared with e-Commerce, e-Business is a more generic term, it refers not only to information exchanges related to buying and selling but also to servicing customers & collaborating with business partners, distributors & suppliers.
- e-Business encompasses sophisticated business-to-business interactions and collaboration activities at a level of enterprise applications and business processes.

- e-Business processes are integrated *end-to-end* across the company and with key partners, suppliers and customers and can respond with flexibility and speed to customer demands and market opportunities.
 - companies link their internal and external processes more efficiently & flexibly, work more closely with suppliers to satisfy the needs & expectations of their customers.
 - Internal or *back-office* processes include distribution, manufacturing and accounting while external or *front-office* processes include these processes that connect an organization to its customers and suppliers.

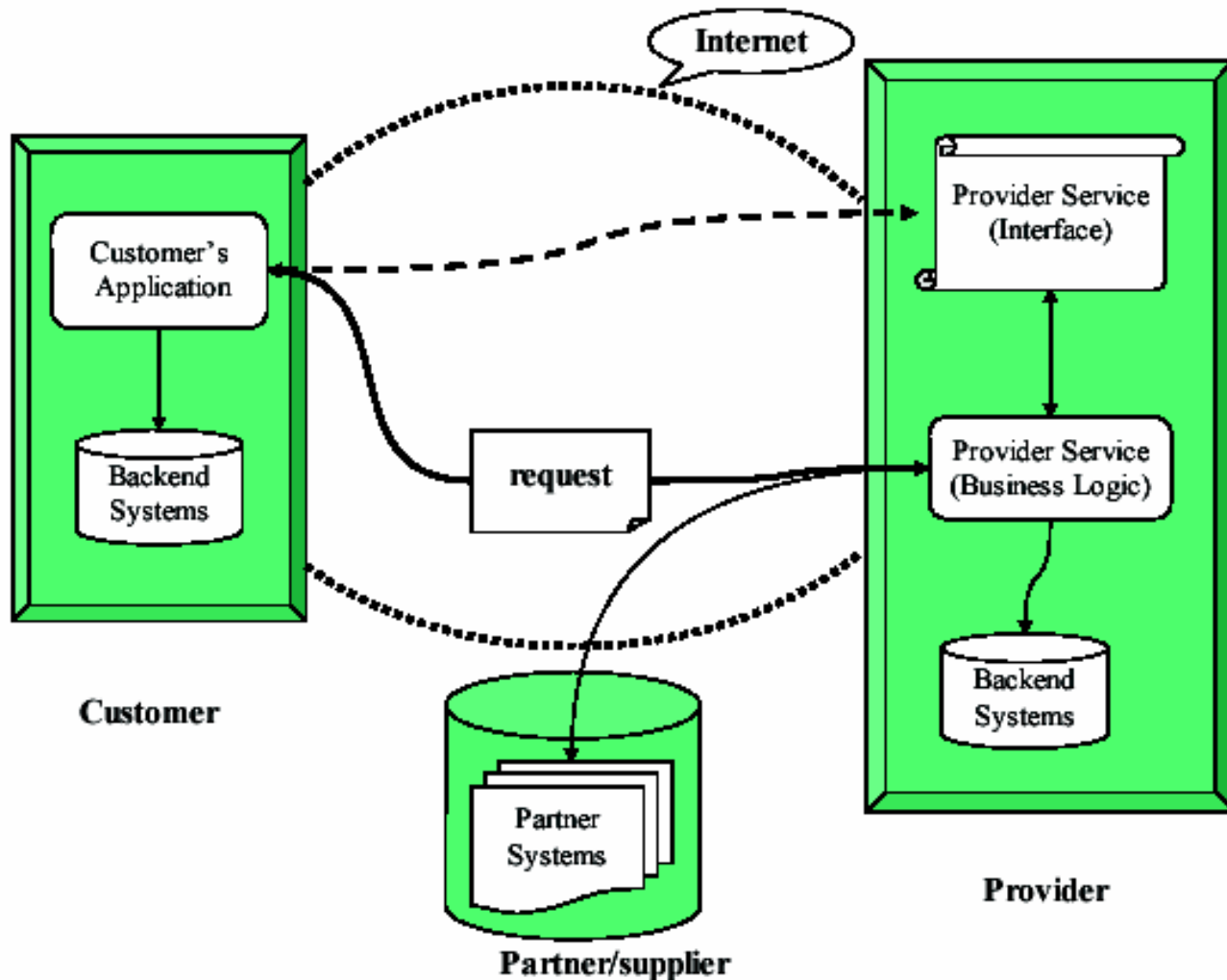
US ECOMMERCE 1998 - 2003



Ingredients of e-Business Solutions



B2B Transactions



From Boualem Benatallah's tutorial at ICSSOC03

□ Front-end

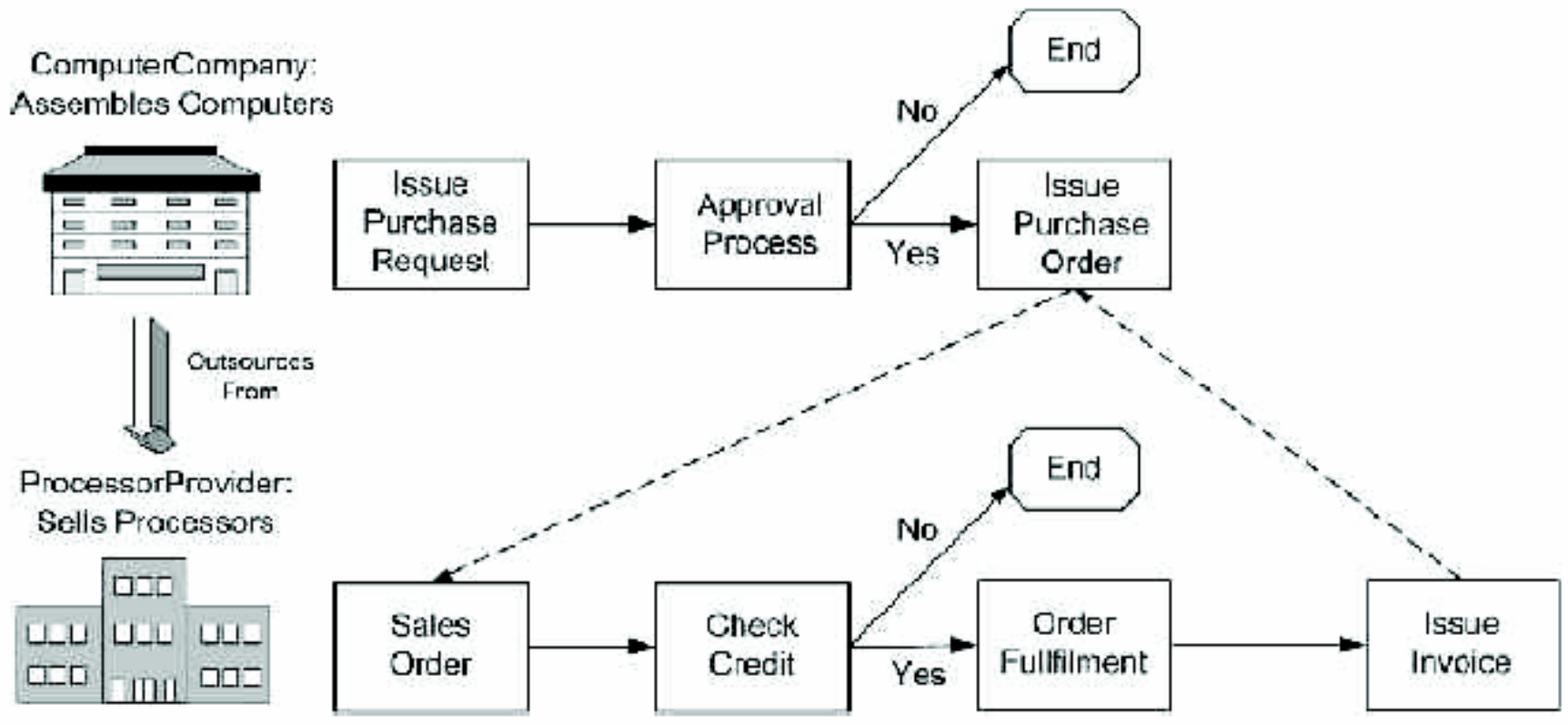
- Pricing
- Estimate lead times
- Configure order

□ Back-end

- Integration with front-end order capture
- Internal operations of order fulfilment
- Integration with roughly 100 manufactures/suppliers

(from Boualem Benatallah's tutorial at ICSSOC03)

B2B Application: outsourcing

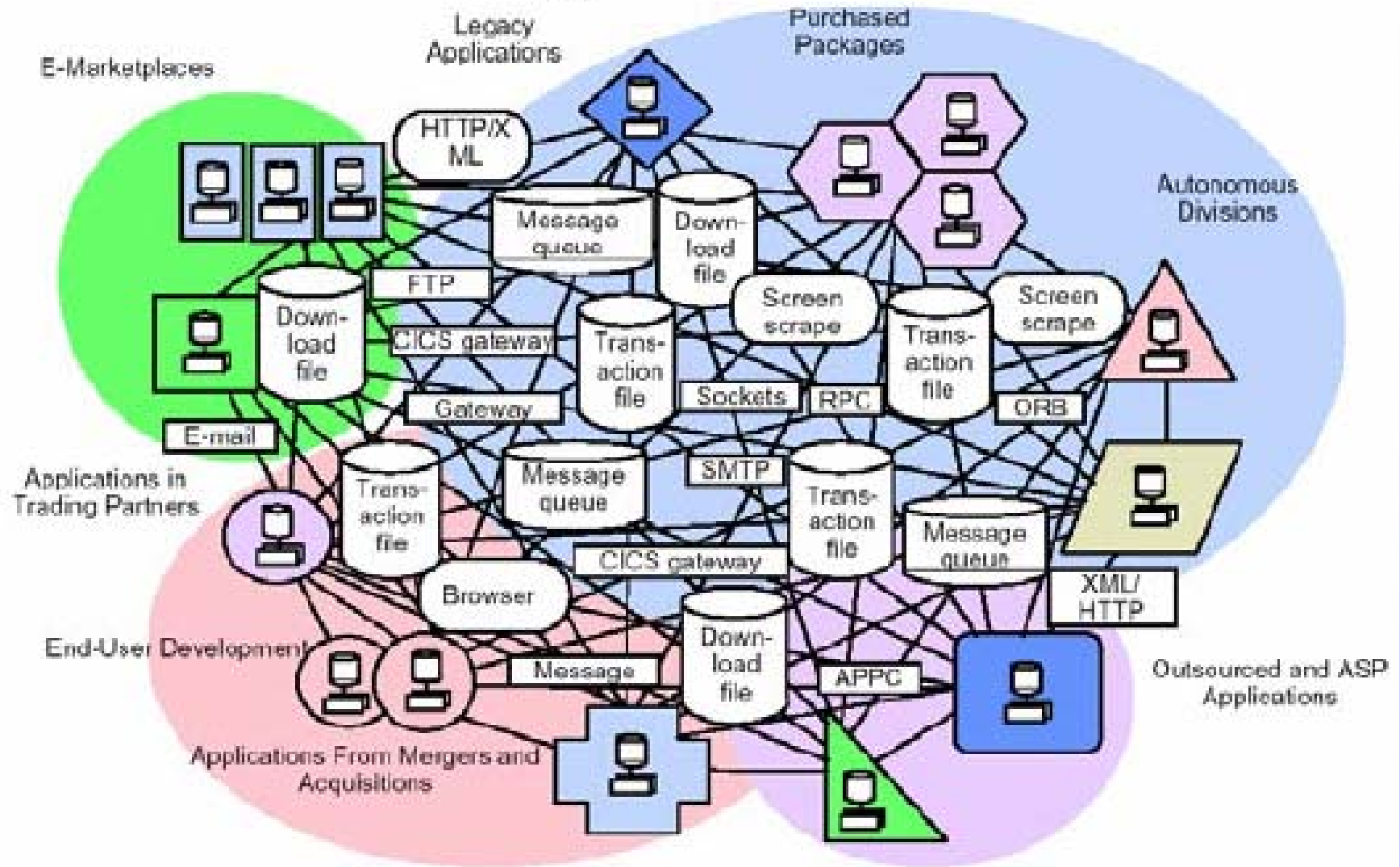


(from Boualem Benatallah's tutorial at ICSOC03)

Integration Aspects in B2B Application

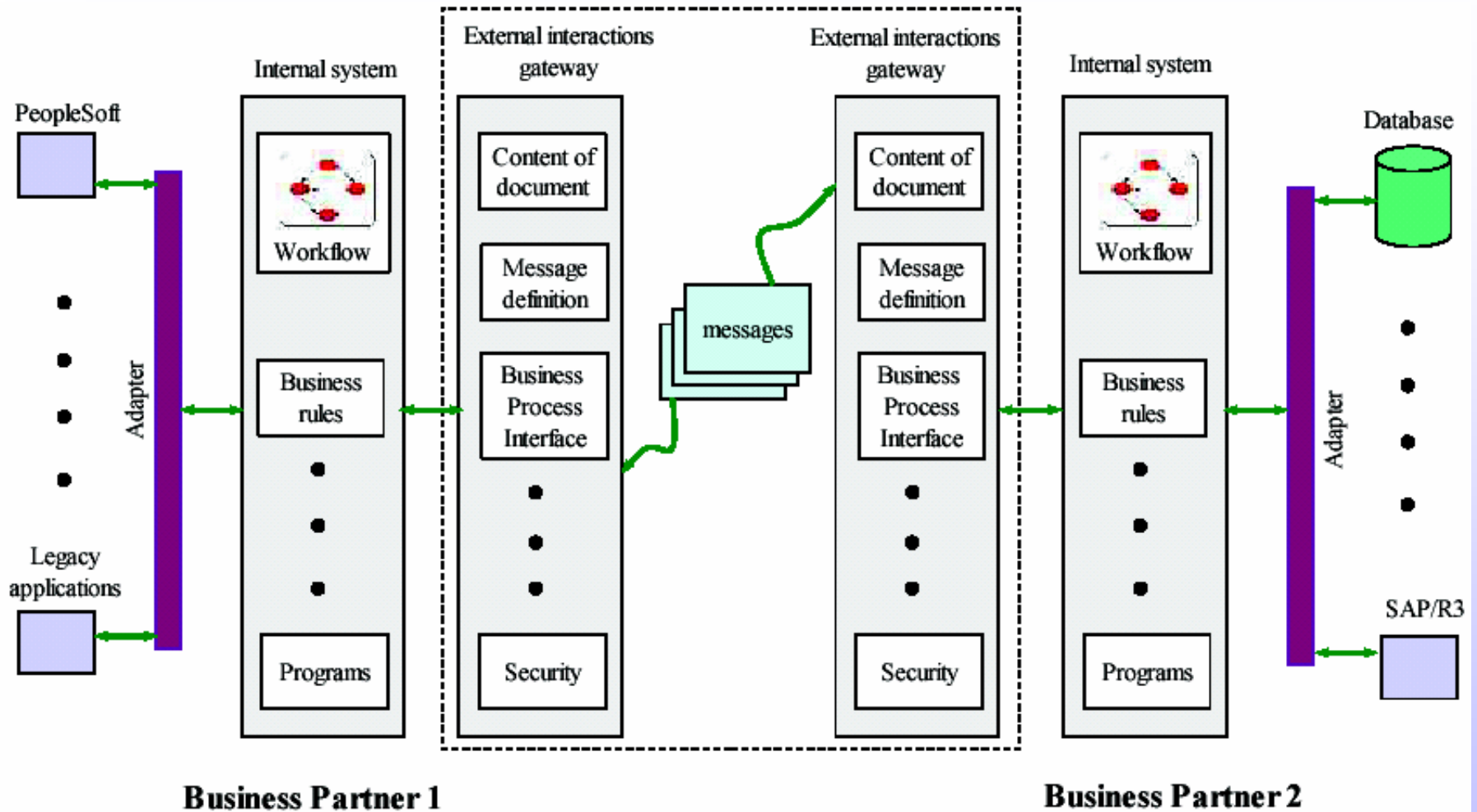
- Both data and processes (e.g., applications, workflows)
- Within enterprise
 - Connect front-end and back-end systems
 - Connect legacy data sources and applications to the front-end system
 - Connect to department systems
- Across enterprise
 - Integration with partner systems

Reality: Constellations of System Clusters



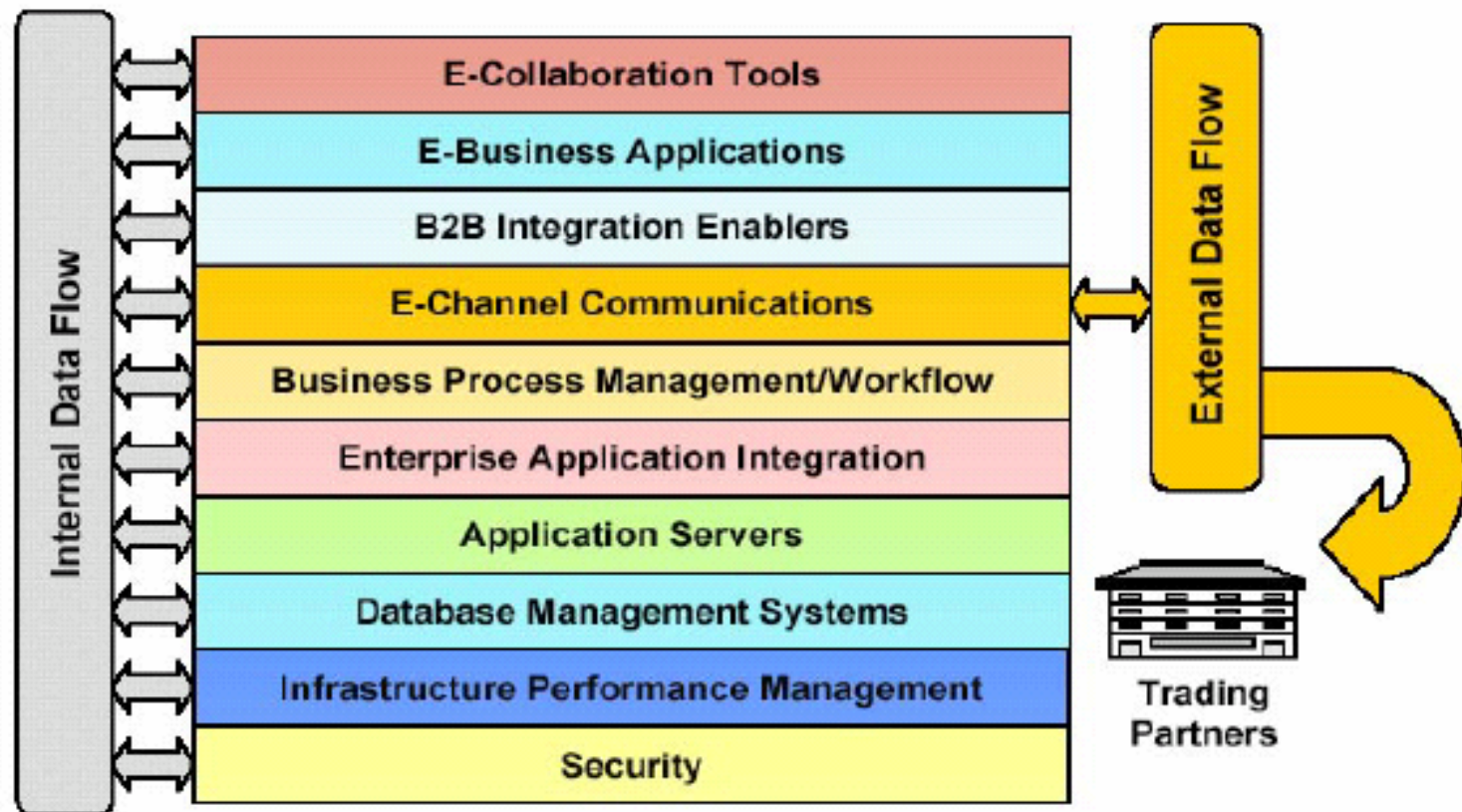
Source: Gartner Research

B2B Integration Framework



From Boualem Benatallah's ICSSOC03 tutorial

The E-Business Integration Technology Stack



Source: Giga Information Group

Figure 1

Practical Solution

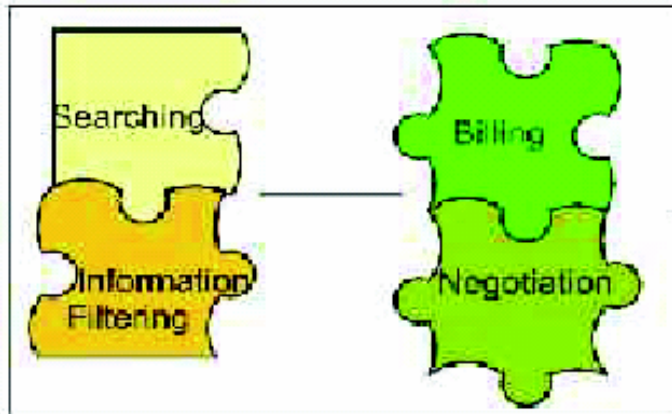
- Spectrum of Solutions
 - Infrastructure / platform ←----- Vendor Hard
 - Automation: language/ modeling / design-----
Turing Hard
 - How far can automation take you?
 - Semantics ←-----
 - Agreements ←----- Politics Hard
 - Formal Nobel Hard
- Community Agreement / Standards
 - Local
 - Enterprise
 - Powerful vendors, associations, ...
 - National / international
- Automation
 - Tools for specific problems
 - Automate
 - Reduce human error
 - Let's look at the current Chaos

Business Integration Paradigm

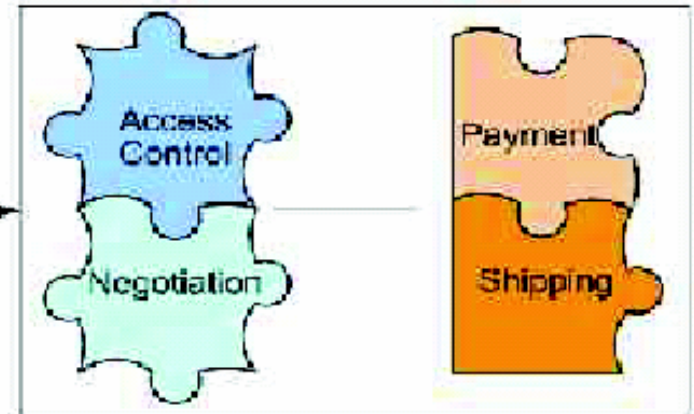
- Component-based middleware
- Process based integration
- Service Oriented computing



Business Partner 1



Business Partner 2



Interactions



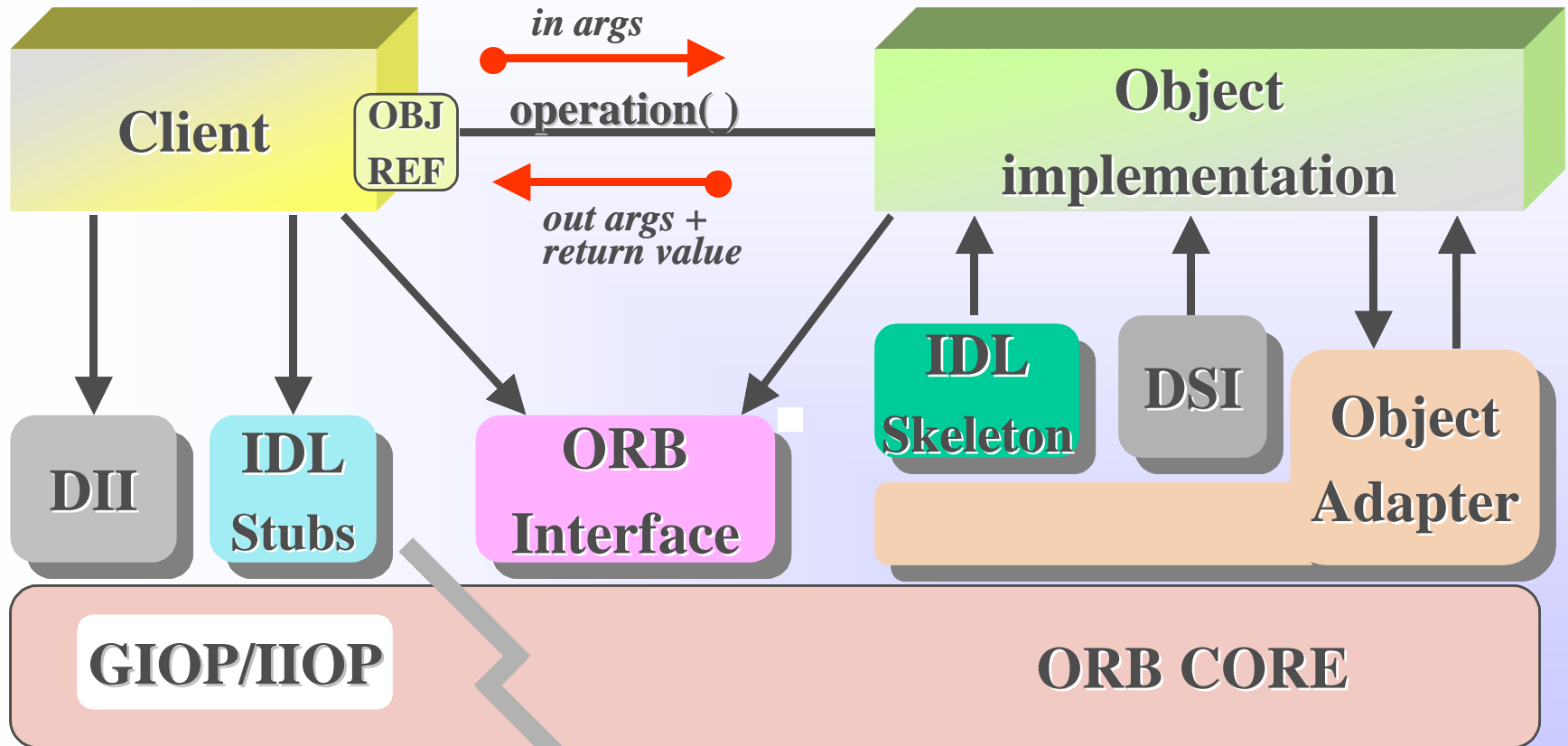
Conform To



Component Middleware
(CORBA, DCOM, EJB, etc.)

(from Boualem Benatallah's tutorial at ICSOC03)

CORBA



integration

- Tight coupling between partner systems (operation invocation)
- Interfaces do not capture business process semantics beyond enumerating interface types
- Because of tight coupling, changes to back-end systems, mediator framework, and business applications must be coordinated across all the components
- Appropriate to integrate small number of tightly coupled systems

Process-based integration

- Automation of business processes is an important enabler for applications integration both : within an enterprise and across partner systems
- Workflow technology is already a mature technology for automating intra-enterprise processes
 -
- Traditional workflows: intra-enterprise, homogeneous and centrally controlled environments
- Inter-enterprise business processes requires:
 - support the collaboration among of diverse users, applications, and systems
 - automate business processes that integrate disparate applications systems

Is distributed workflow a solution?

- Partitioning global workflows into sub-workflows
- Sub-workflow = activities that are to be executed by a unit (organization)
- Impose that each participant deploys a full-fledged execution engine capable of interpreting the workflow definition
- Same process model must be adopted by all participants
- Assume a tight coupling among sub-workflows
- Quite restrictive for B2B collaboration :
 - Partners may use disparate data and process representation models
 - Modifications of back-end applications, sub-workflows, and global workflow need to be coordinated.
 - The cost of establishing a new relationship may be significant (business processes must be modelled and deployed in concert)

Recurring Theme:

Business integration requires:

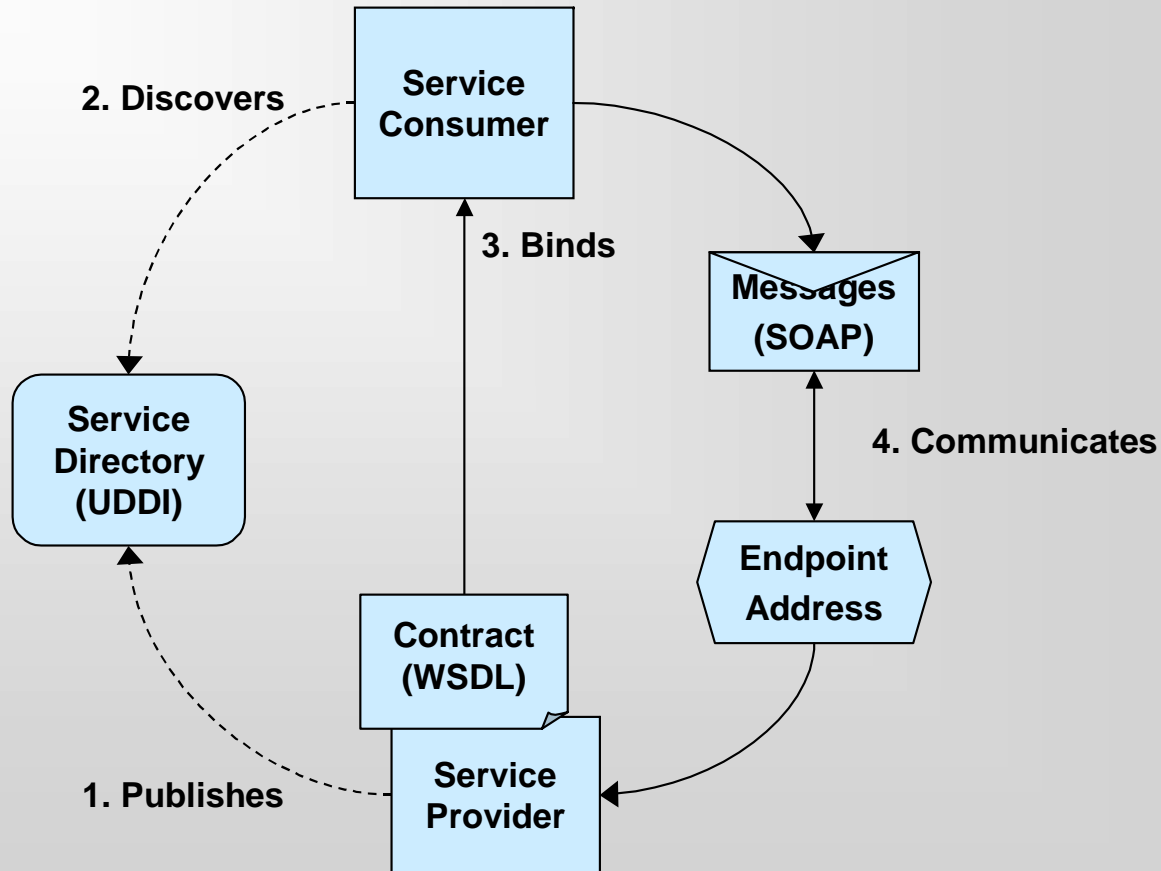
- Open
- Flexible
- On demand
- Distributed
- Adaptable
- Plug and play
- Re-usable
- Integrated

Service Oriented Computing: Definitions

- *Service Oriented Computing (SOC)* is the computing paradigm that utilizes services as fundamental elements for developing applications/solutions. SOC involves the service layers, functionality, and roles described by a Service-Oriented Architecture.
- *Services* are autonomous platform-independent *computational entities* that can be defined, published, discovered, selected, composed, managed and programmed using XML artefacts for the purpose of developing distributed interoperable applications.
- A *web service* is a specific kind of service that is identified by a URI and exhibits the following characteristics:
 - exposes its features programmatically over the Internet using standard Internet languages and protocols, &
 - can be implemented via a self-describing interface based on open Internet standards.

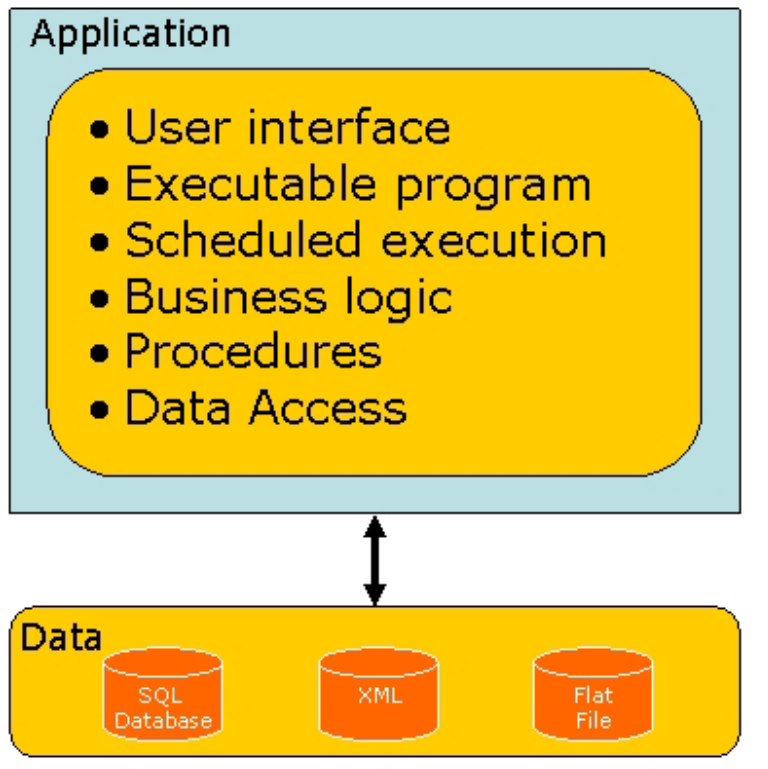
Service Oriented Architecture (SOA)

Basic Service Concept



© 2005 CBDI Forum Ltd

What is New?

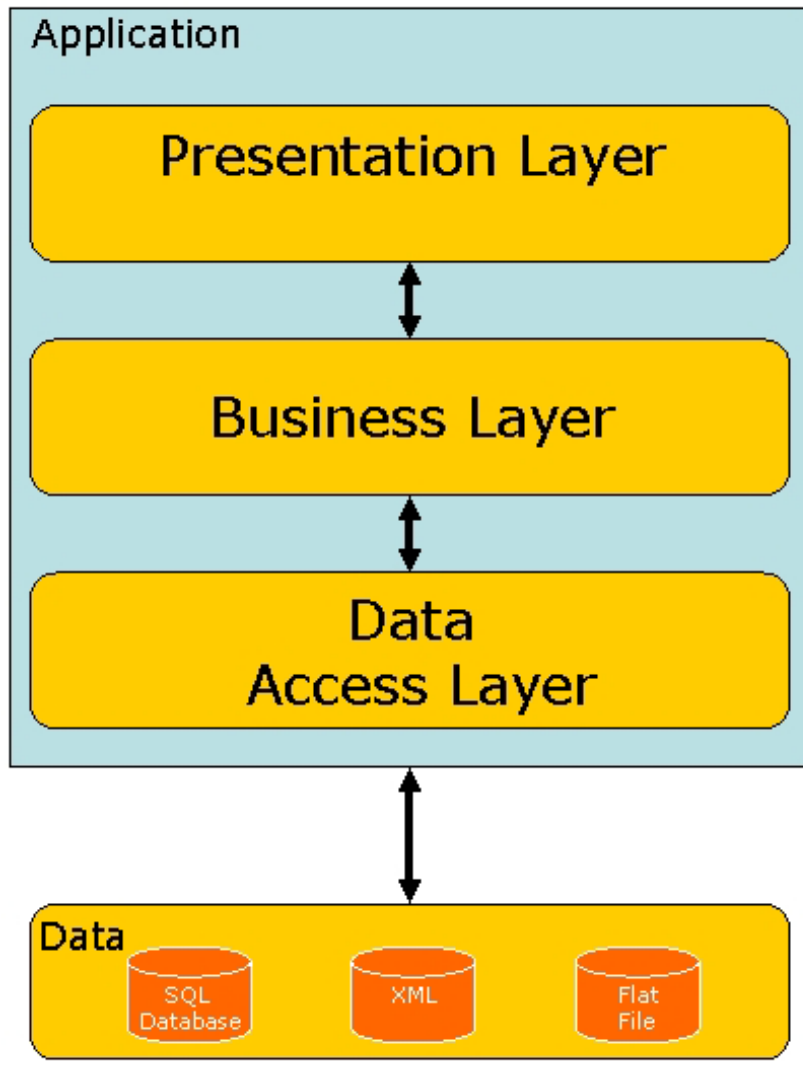


Problems with this approach:

- Application's functions cannot be reused.
- It is difficult to debug as the program grows.
- It's difficult to integrate other applications.
- Security is a problem.

Traditional application architecture combines Everything into a single program.

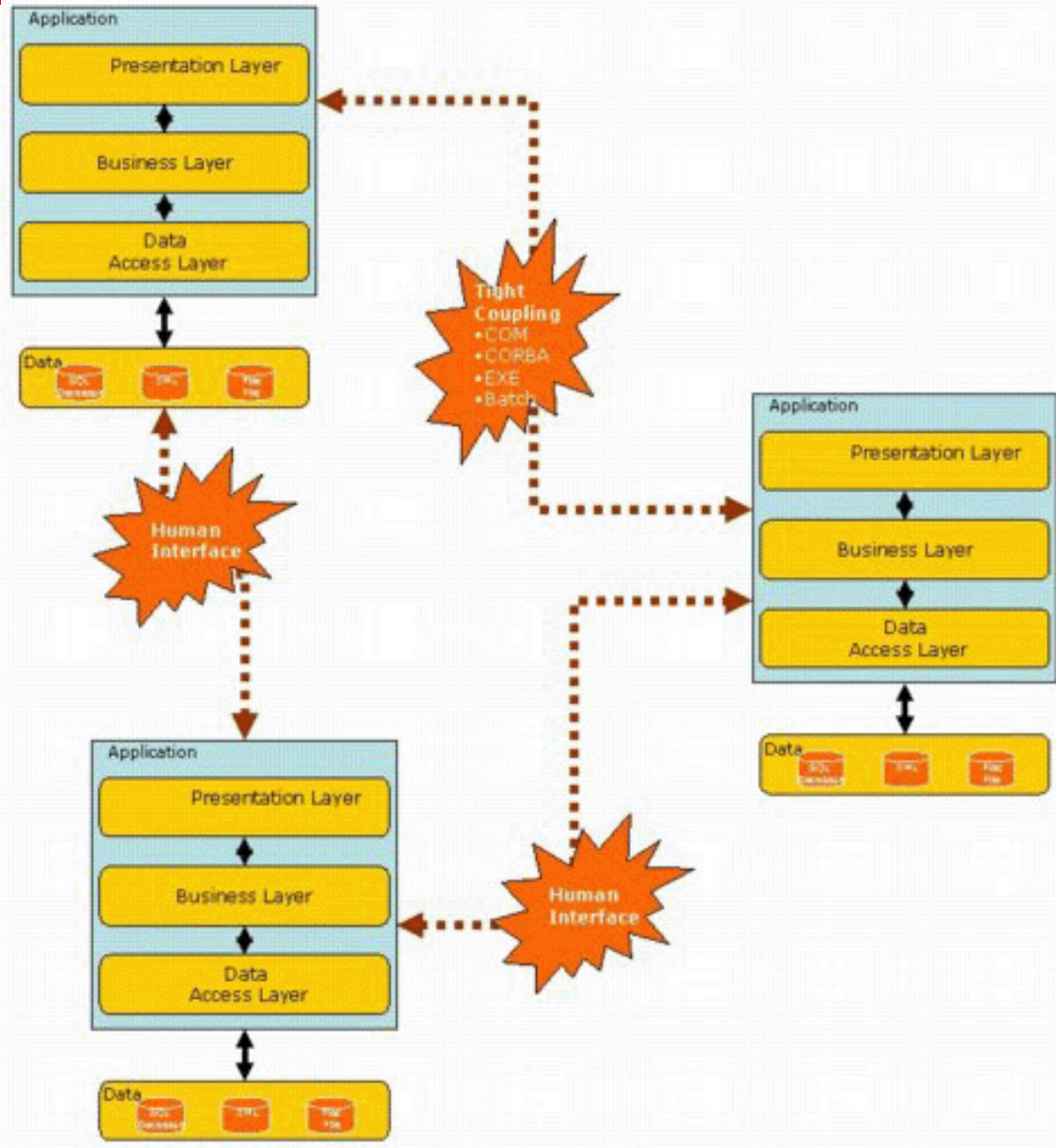
@IBM



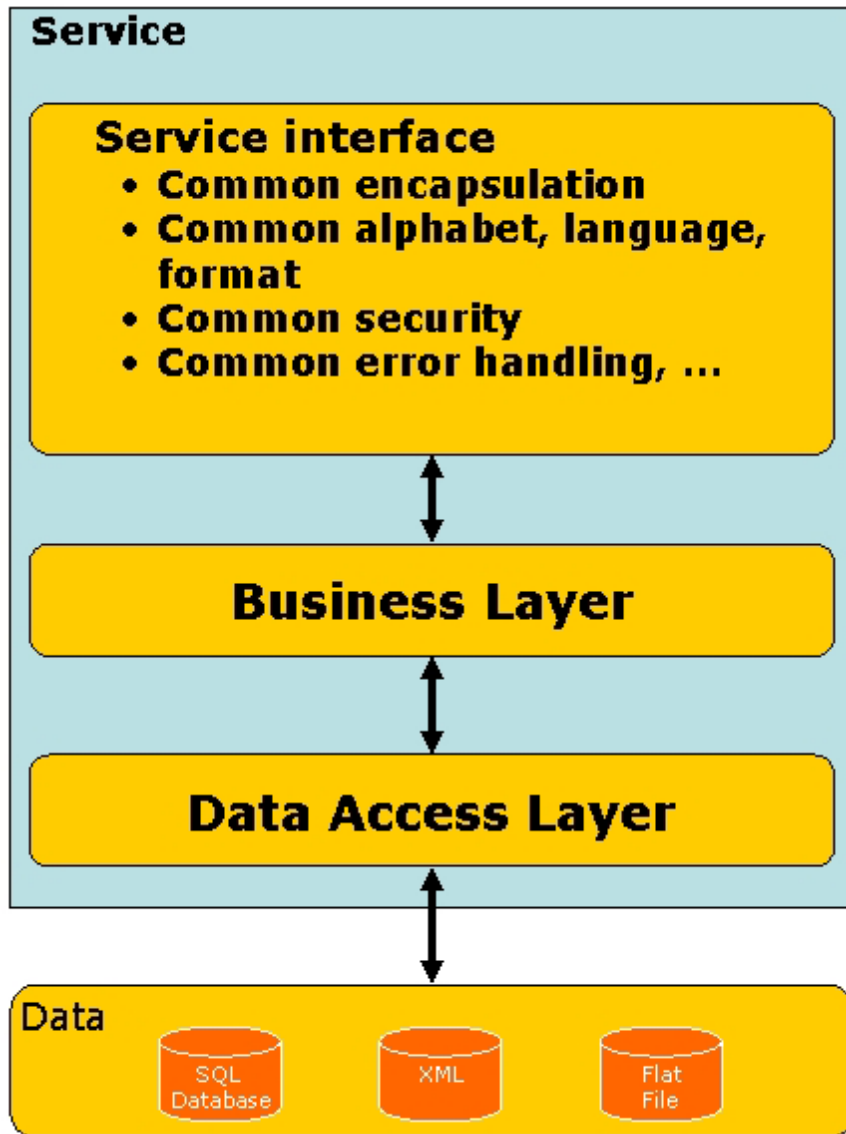
Component-based application architecture

@IBM

Problem in integration

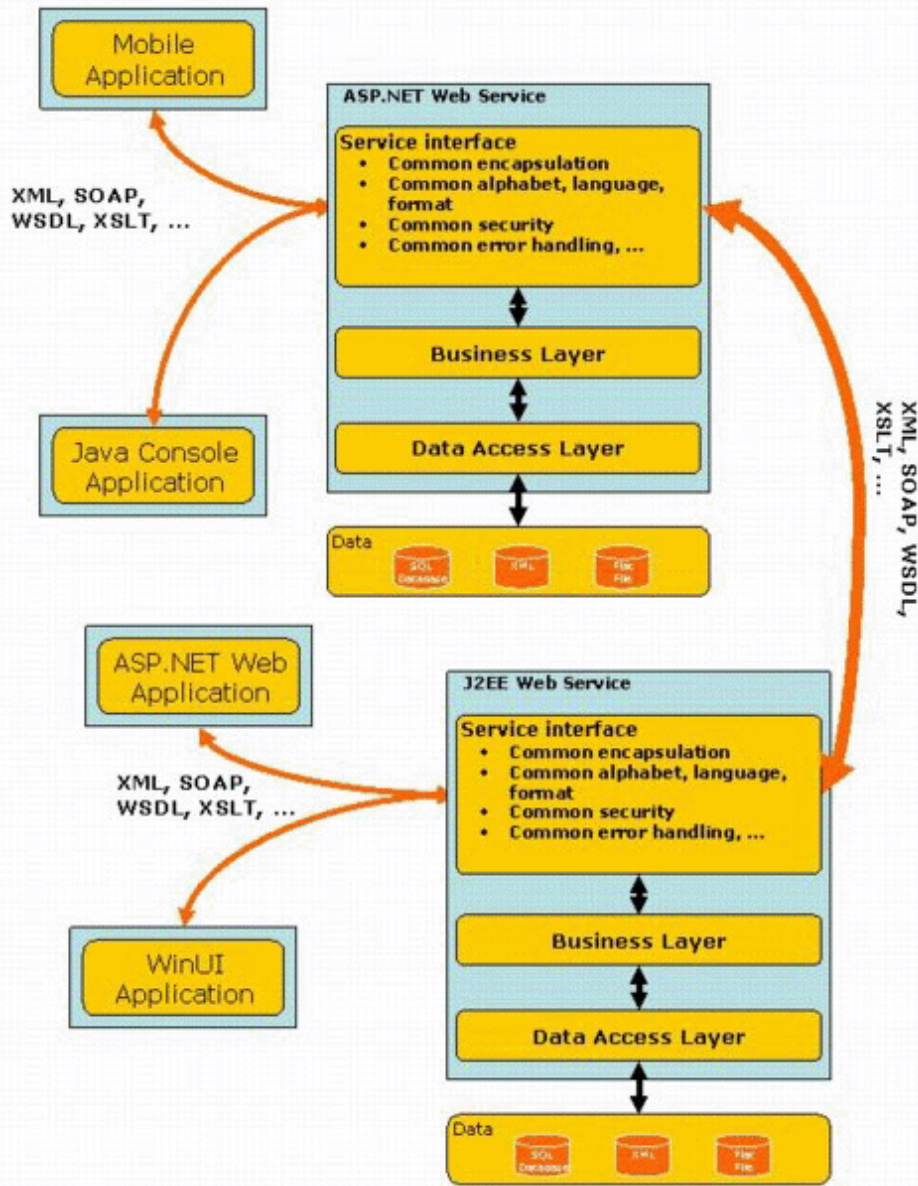


Solution: Service Oriented Architecture (SOA)



@IBM

SOA holds the promise of interoperability



The key to the interoperability is the concept of exposing services rather than application or data.

SOA encapsulating existing systems & functionality is a place where service-oriented applications really shine.

@IBM

Collaboration: ebXML

Compose, combine Services: BPEL

Publish, Find, Use Services: UDDI

Formal Service Descriptions: WSDL

Service Interactions: SOAP

Universal Data Format: XML

Ubiquitous Communications: Internet

Some integration technologies

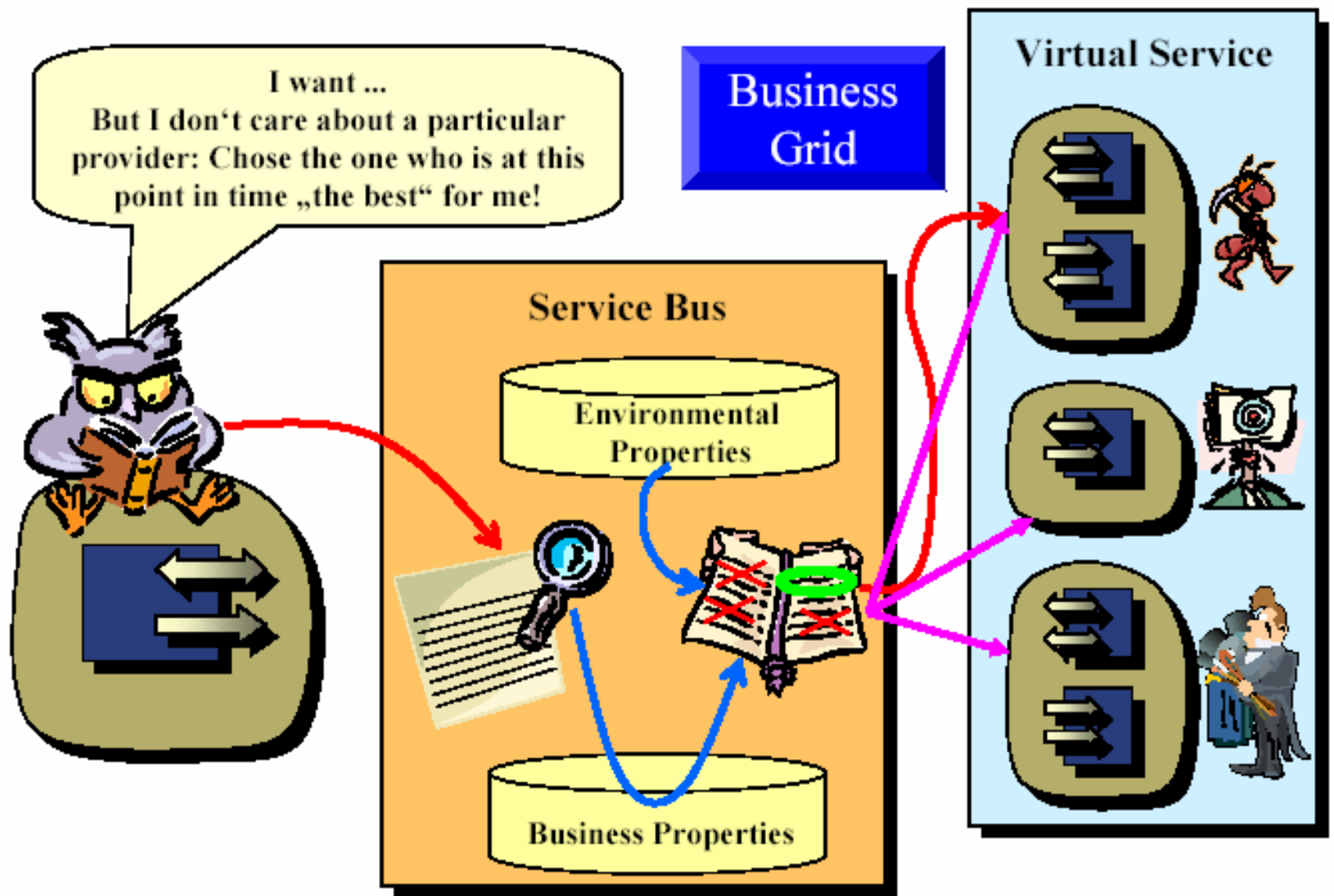
- All these technologies are confusing
 - Wrapper
 - Adaptor
 - Connector
 - Channel
 - ...
- Where does it stop? ■

Web Services!!!

Web Services Technology: the key thing:

- Web service technology provides a ‘virtual component model’ for using components (read: message processor)
- When exploiting a web service the supporting container hides ‘middleware idiosyncrasies’ like the underlying component model of the actual target component used, the invocation protocol, etc, as far as possible.
- Web service technology does not provide a new component model for implementing components.

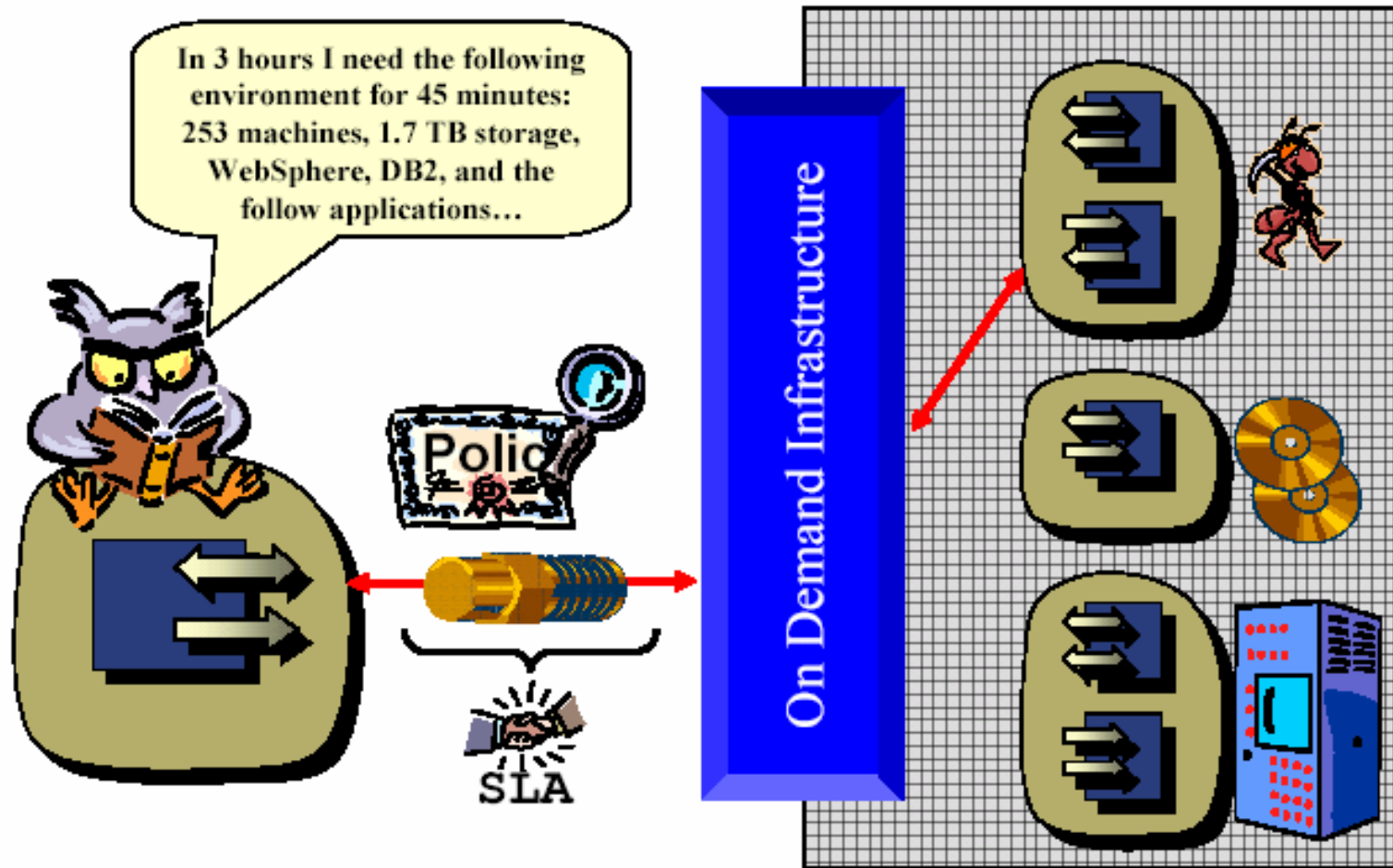
Web Service: virtual functions



© IBM

Frank Leymann

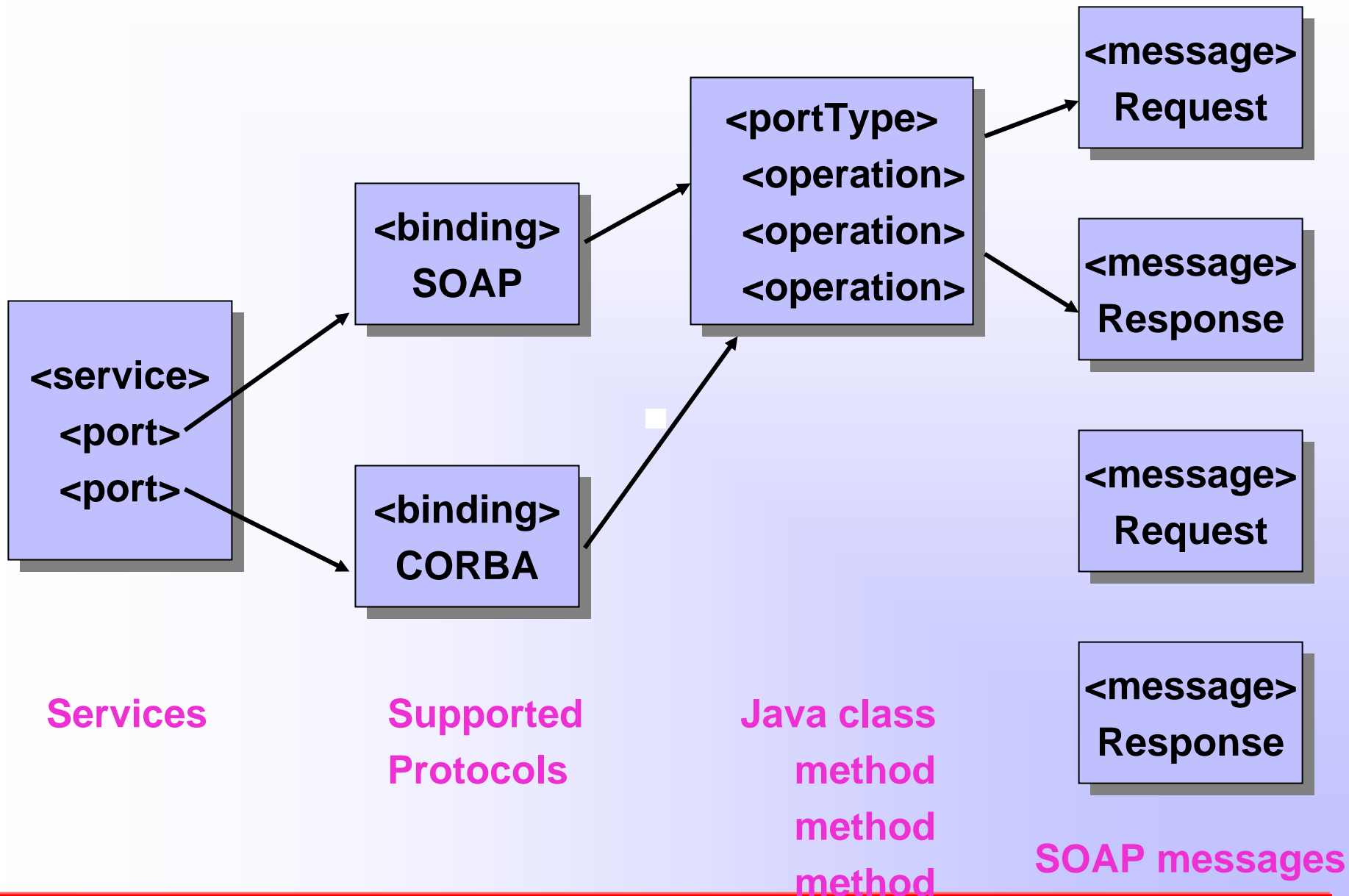
Service composition on demand



© IBM

Frank Leymann

WSDL element relationships



WSDL - Overview

<definitions>

<types> <!-- XML Schema --> </types>

<message name="getTradePrice_In" />

<message name="getTradePrice_Out" />

<portType name="StockQuoteServiceInterface">

<operation name="getTradePrice">

<input message="getTradePrice_In" />

<output message="getTradePrice_Out" />

</operation>

</portType>

<binding name="StockQuoteServiceBinding" type="StockQuoteServiceInterface">

<soap:binding transport="http://schemas.xmlsoap.org/soap/http" />

...

</binding>

<service name="StockQuoteService">

<port name="StockQuoteServicePort" binding="StockQuoteServiceBinding">

<soap:address location="http://www.stocks.com/services/stockquote" />

</port>

</service>

</definitions>

Definition of data types

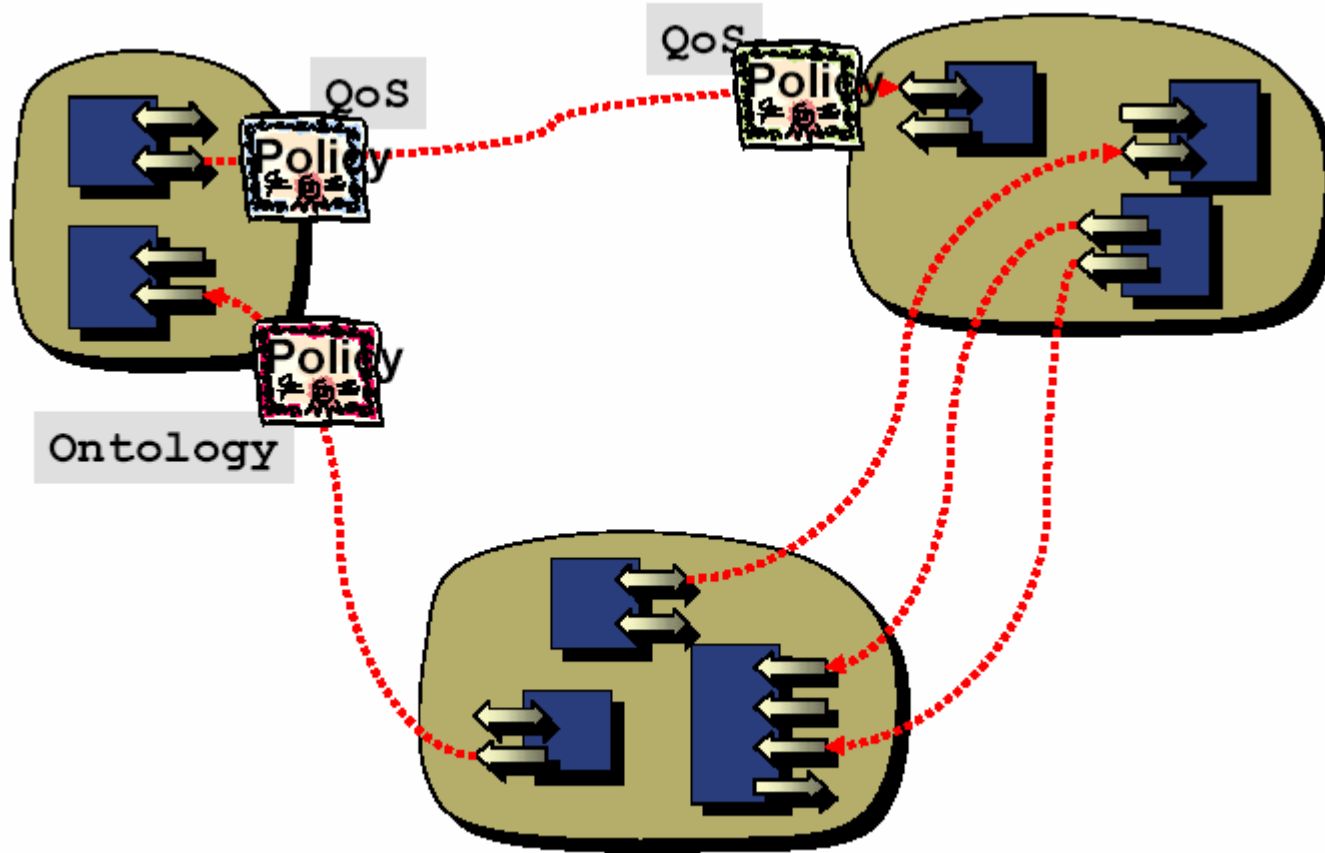
Definition of messages

Definition of port type

Definition of the bindings

Definition of the service

Quality of Services & Policies



© IBM

Frank Leymann

Policy: Example

```
01 <wsp:Policy>
02   <wsp:ExactlyOne>
03     <wsse:SecurityToken>
04       <wsse:TokenType>wsse:Kerberosv5TGT</wsse:TokenType>
05     </wsse:SecurityToken>
06     <wsse:SecurityToken>
07       <wsse:TokenType>wsse:X509v3</wsse:TokenType>
08     </wsse:SecurityToken>
09   </wsp:ExactlyOne>
10 </wsp:Policy>
```

Matchmaking Services:

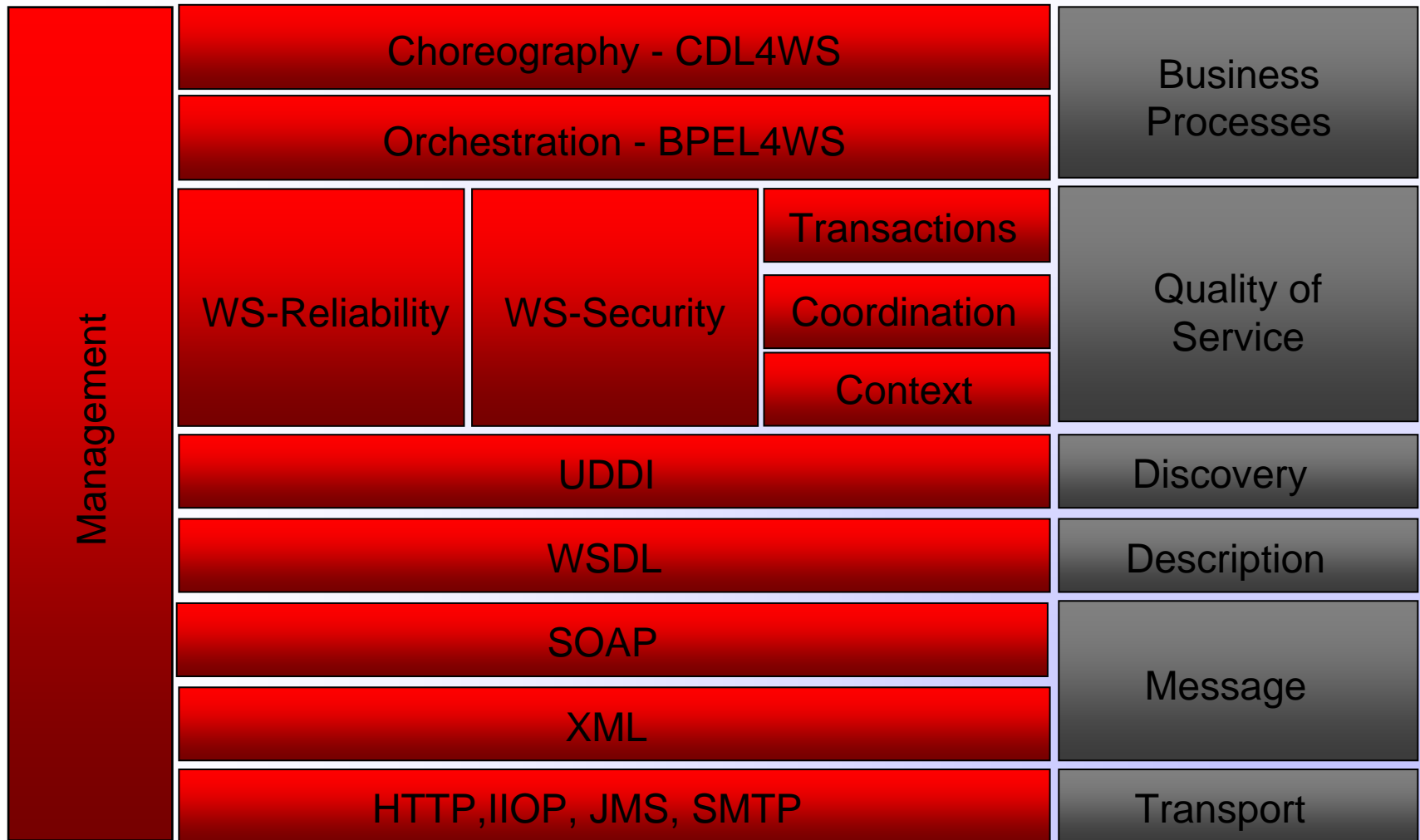
- WSDL description
- Explicit policies
- Implicit policies



BPEL – Service Ochestration

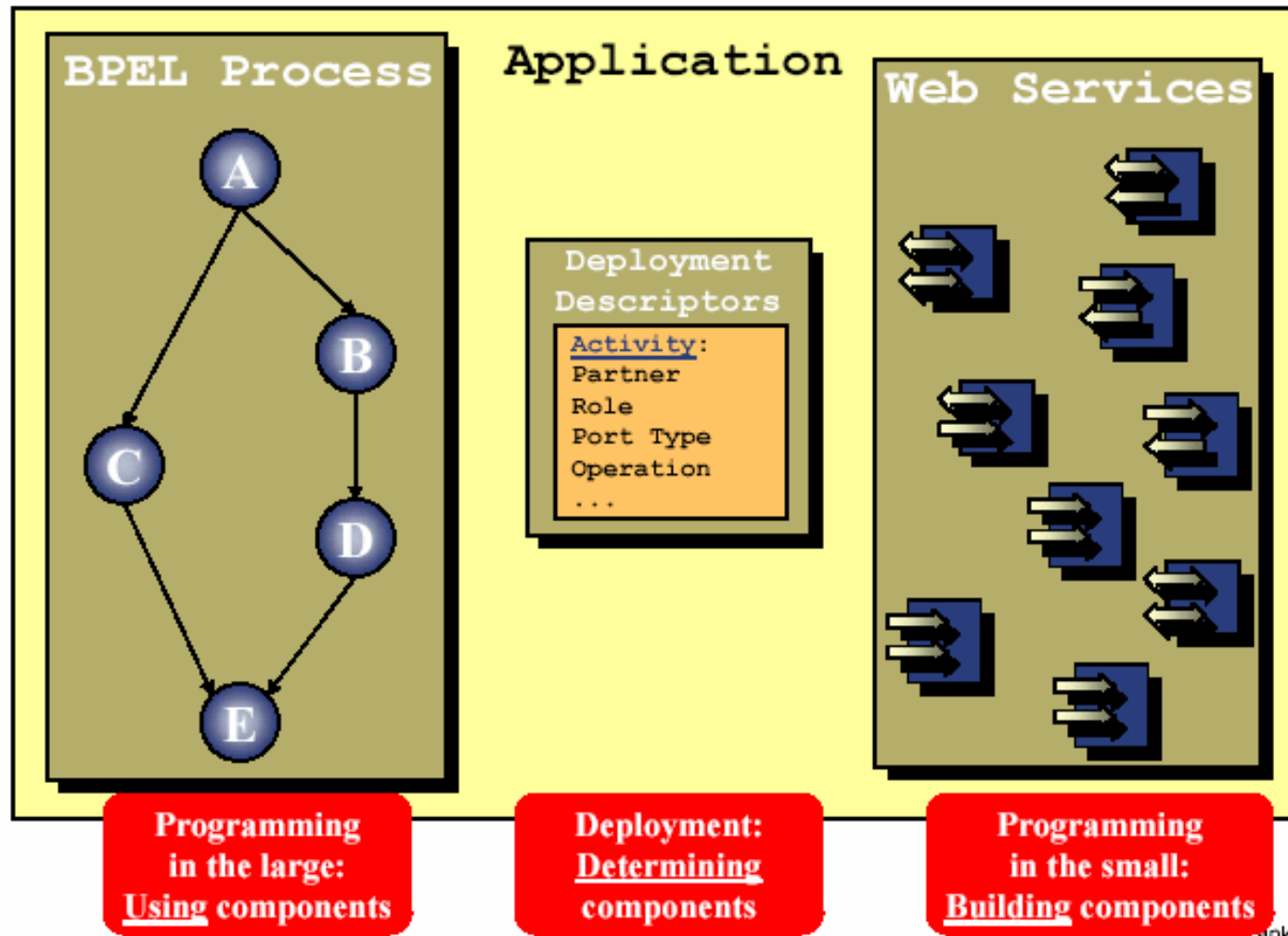
- A language to specify business processes...
 - ...between Web services...
 - ...and as Web services
- Same language to define executable processes and business protocols
- Executable processes
 - Well-defined operational semantics
 - Can be “enacted” in compliant environment
- Abstract processes
 - Specify constraints of message exchange
- Protocols between partners

Standards Building Blocks of BPEL



@Oracle

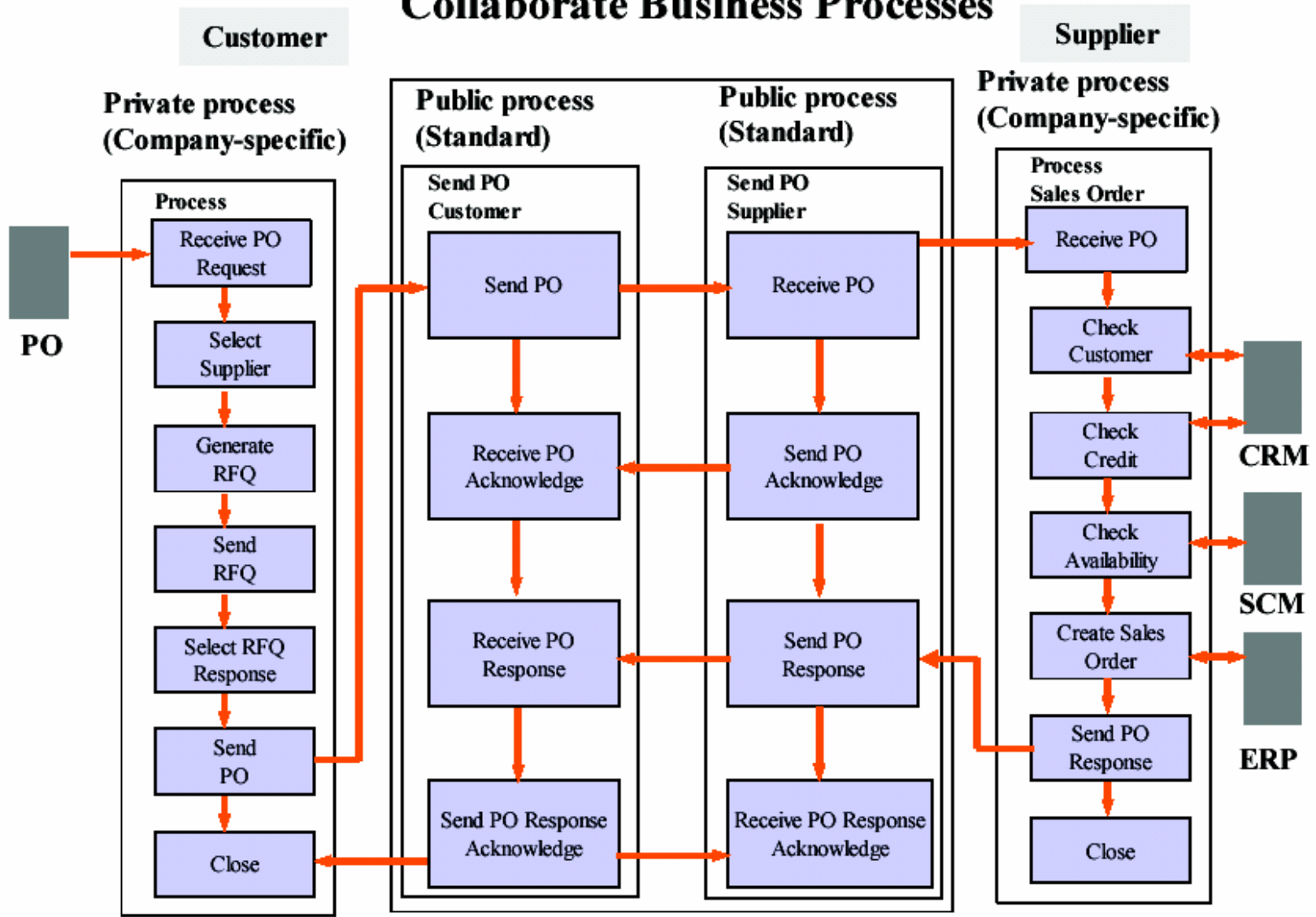
Two level programming



© IBM

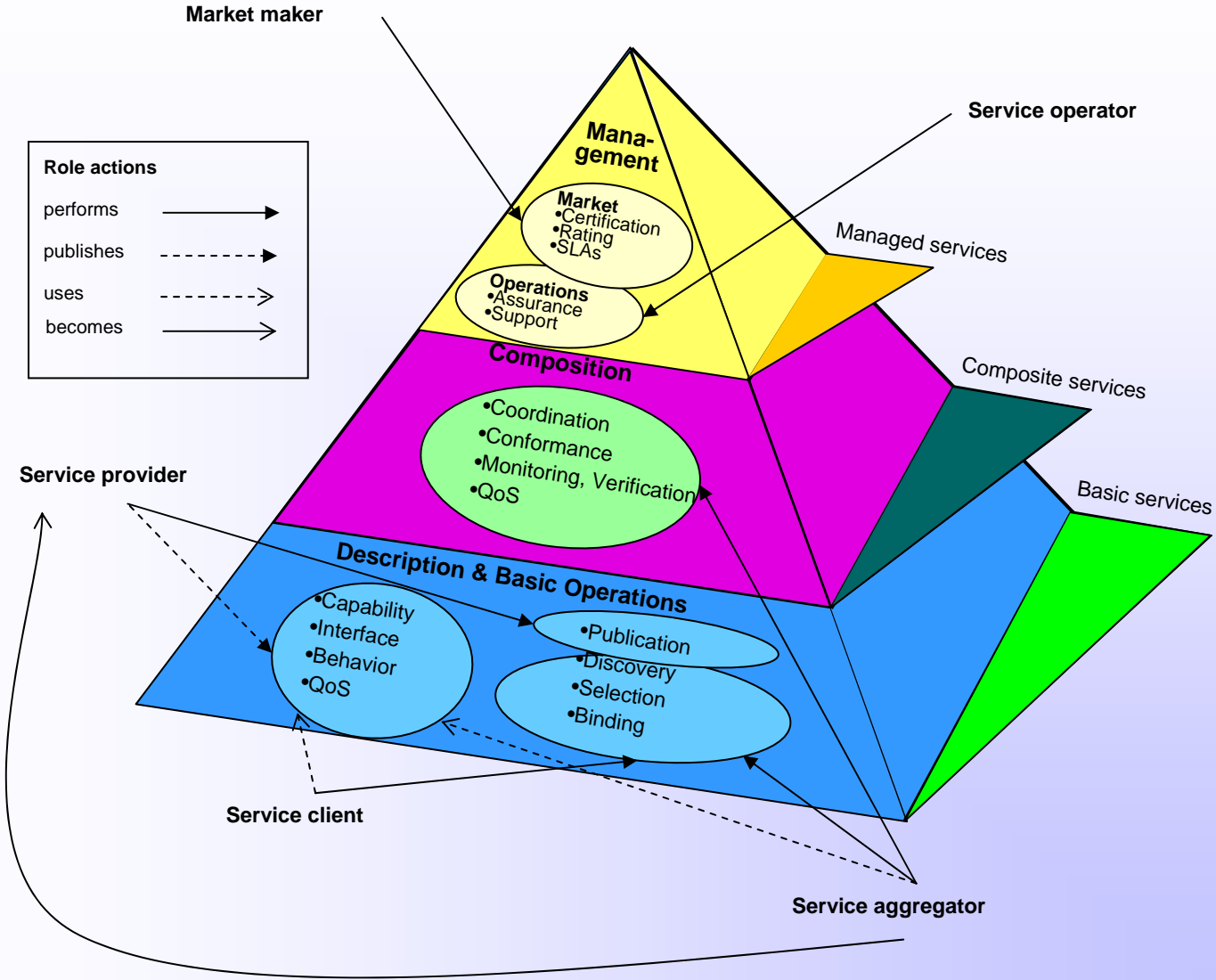
Frank Leymann

Collaborate Business Processes



(Source: e-business Architectures and Standards, Anil L. Nori, Tutorial, VLDB'2002, HongKong, China)

Extended Service-oriented Architecture



More ...



Issues to be researched

- Application model is centered around ...
 - SOA – policy driven dynamics
 - Choreography – two level programming
- Modeling and Design (dynamic, maintainable,...)
- Management issues (matrix, monitoring)
- Business transaction management
- Security and trust

