# UDDI and Its Role in Web Services

George Zagelow

Chair, OASIS UDDI Steering Committee

Managing Director, UDDI Operators Council

IBM Web Services Strategy

# Agenda

- **The Web Services business drivers**
  - What problem are we trying to solve?
  - What are Web Services
  - How do Web Services help?
- **Enabling Technologies**
  - XML
  - SOAP - ~~Simple Object Access Protocol~~
  - WSDL - Web Services Description Language
  - UDDI - Publishing and Finding Web Services
  - New Hot Items
- **IBM and Web Services**
- **Customer Examples**
- **Summary**

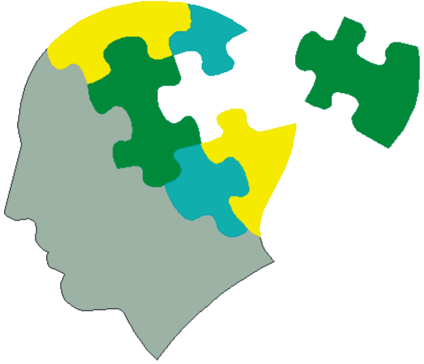
# A New Web Model

- Until now, the Web has provided for
  - browsing of linked documents
  - manually-initiated purchases and transactions
  - downloading files
    - all of this is manual, by way of a browser
- Web Services is a new model for using the Web
  - transactions initiated <u>automatically</u> by a program, not necessarily using a browser
  - can be described, published, discovered, and invoked dynamically in a distributed computing environment
  - new ways of using the web: intelligent agents, marketplaces, auctions
    - **all built on XML and other internet standards!**

# Why Web services?

- We want and need:
  - to integrate systems regardless of their implementation
  - to move from monolithic, custom-coded apps to choreographed, scripted components.
  - agility and flexibility to reconfigure business functions

    to try new process models.
  - to move from tightly coupled systems to loosely coupled ones to deal with inevitable change.
  - a well-understood programming model for connecting businesses via the Internet.

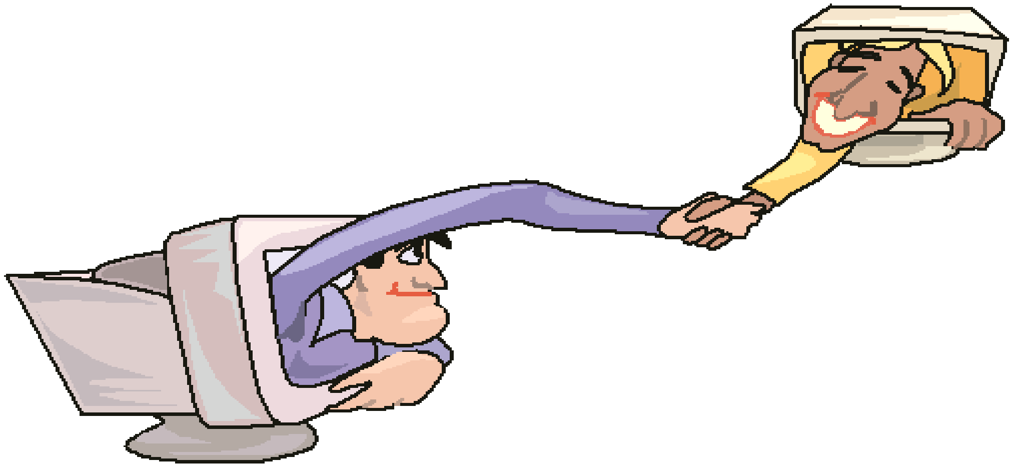


# Enterprise Application Integration: Putting together the pieces

- **Legacy: heterogenous application systems**
  - Difficult to tie them together
- **Traditional integration (compiled-in API's and file formats)**
  - Leads to "brittle" systems that crash with the smallest of changes
  - Different systems, different techniques: no standards
- **Need fast integration for Mergers and Acquisitions**
  - An "integration-ready" company has much greater value
- **Need agility and flexibility in business processes**
  - Respond to business changes
  - Try out new business process ideas quickly and cheaply

# B2B: better faster cheaper

- **Rapid and deep integration with business partners**
  - Harder than EAI: you don't control the software you need to integrate with!
- **Reduced cost of doing business through more efficient communication**
  - eliminate manual processes, paper communication
- **Find new business partners, integrate quickly**
  - new supplier when the old one can't deliver in time
  - respond to emerging business opportunities while they're hot
- **Participate in emerging business models**
  - industry-specific marketplaces
  - auctions



# Advantages of Web Services Architecture

## Flexibility

- universal interfaces can cope with inevitable changes in software caused by changing business needs

## Agility and Productivity

- rapid application assembly tools allow integration for new business opportunities or trying new business ideas

## Cost Savings

- allow automatic transactions
- replace manual methods
- reduce staffing requirements
- replace paper processing
- reduce errors

## Leverage Existing Investments

- old software can be used in new ways by building a Web services layer for universal access

## Leverage Developer Skillsets

- the plumbing code (the harder part) is generated automatically
- can be integrated and tested with traditional methods

## Broad Applicability

- EAI, B2B, handhelds, browsers, grid computing, …

"By 2005, the aggressive use of Web services will drive a 30% increase in the efficiency of IT development projects"

Gartner Inc, "The Hype Is Right: Web Services Will Deliver Immediate Benefits", October 2001

# Web Services: Emerging standards

## Standardized specifications are required to make software from different vendors work together well

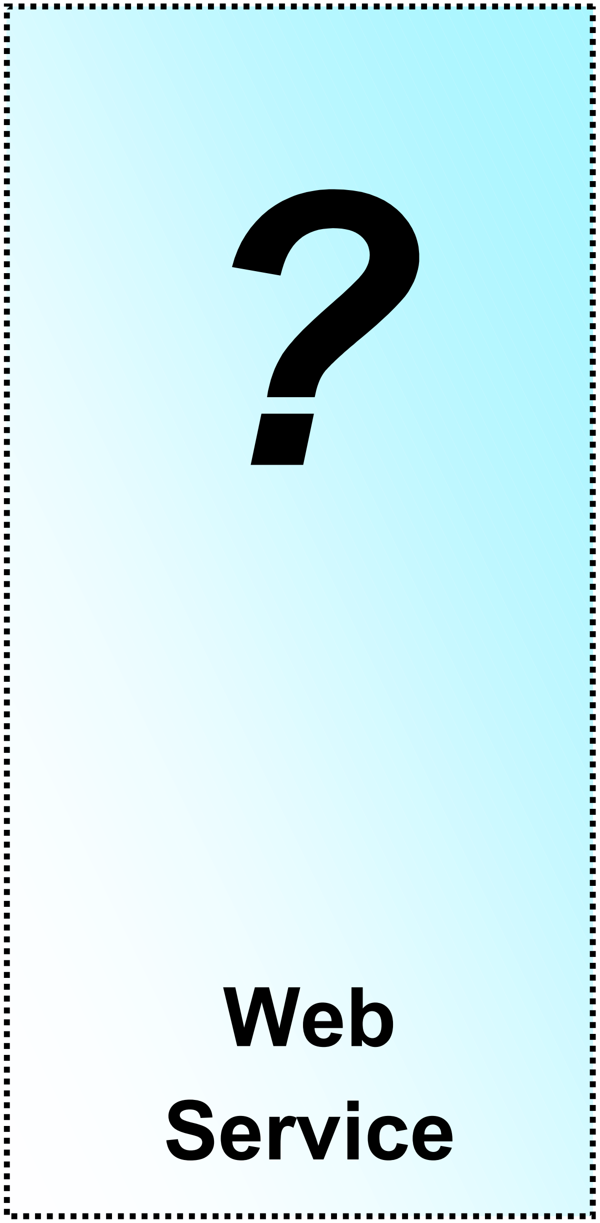
- **XML** defines a universal way of representing any data, making data integration simpler
- **SOAP** uses XML as messages to define a universal Web service requests, making process integration simpler
- **WSDL** specifies all information needed for integration, making application assembly tools possible
- **UDDI** is a special Web service which allows users and applications to locate required Web services

## This year we have had three important advances:

- **WS-I.org** formed to achieve seamless interoperability
- **WS-Security** defines message-level security for SOAP
- **Business Processes:** BPEL4WS, WS-Transaction, WS-Coordination

# What is a Web Service?

"Web services are software components described via WSDL which are capable of being accessed via standard network protocols such as SOAP over HTTP."



?

Web Service

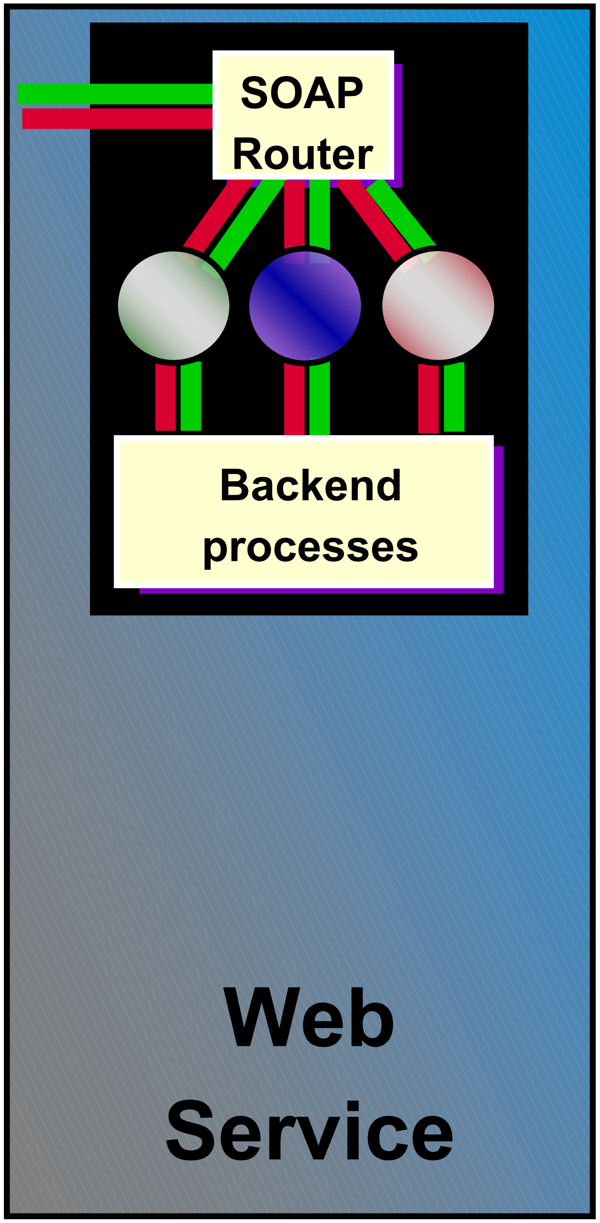
# What is a Web Service?

**"Web services are software components described via WSDL which are capable of being accessed via standard network protocols such as SOAP over HTTP."**

Today, SOAP over HTTP is the common protocol for Web services.

For now, a SOAP interface connected to application processes can be thought of as a minimum...

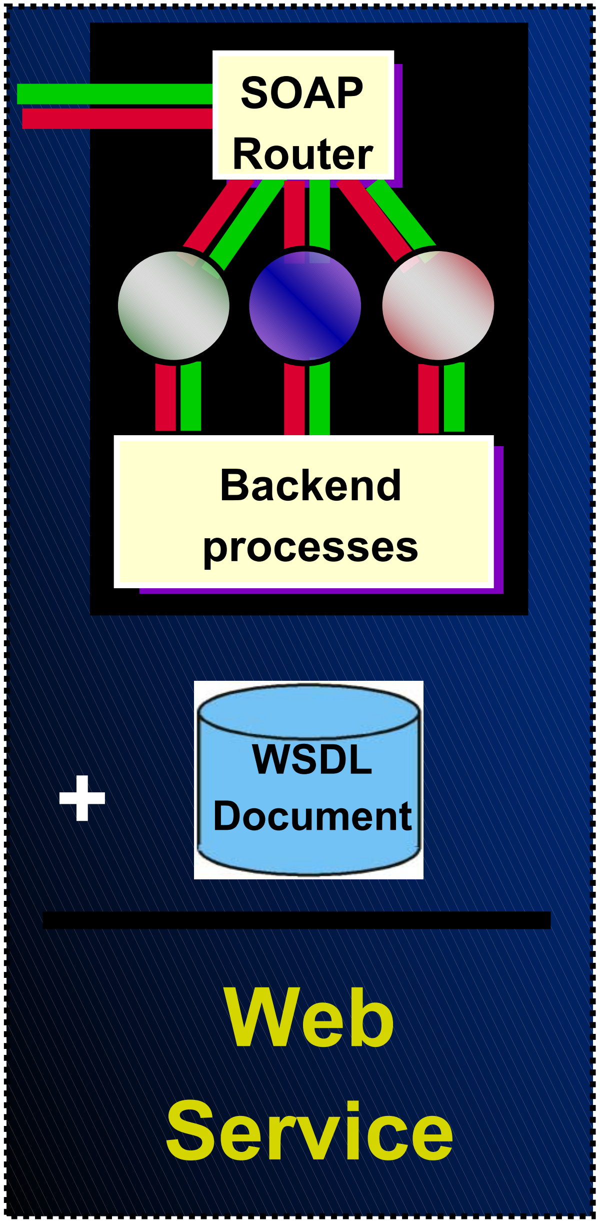
**...but by itself does not address rapid integration.**



# What is a Web Service?

""Web services are software components **described via WSDL** which are capable of being accessed via standard network protocols such as SOAP over HTTP."

- WSDL descriptions can be used to drive assembly tools, code generators, and other tools to speed integration.
- For now, SOAP+WSDL can be thought of as the base technologies for any Web service.
  - UDDI, other technologies can be considered optional, to add on as makes sense for the application



# XML (Extensible Markup Language)

**XML is the key to interoperability.**

**With XML we can exchange data between any applications, regardless of**

- operating system
- programming language
- hardware platform
- delivery device
- software vendors

**XML offers complete data-level integration.**

- SOAP defines a model of sending messages between applications, giving us **process-level integration**.

**XML is the basis for all Web services standards.**

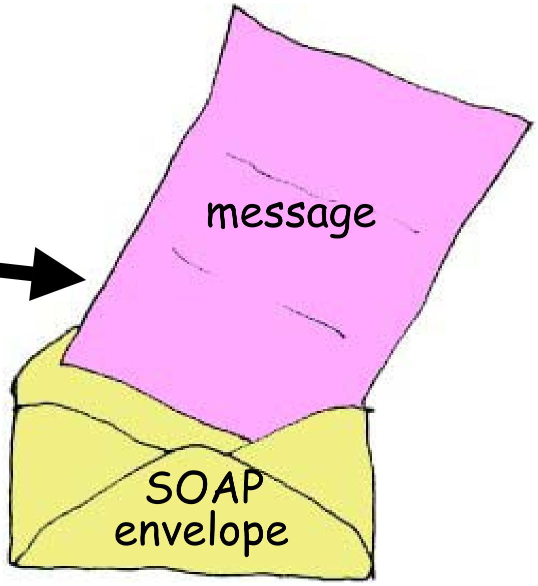
```
<?xml version="1.0"
  encoding="UTF-8"?>
<PurchaseOrder>
           <Customer name="IBM ISL"

           id="8204374">
                      <ShippingAddress>
                                 <street>17 Hawthorne Drive</street>
                                 <city>Hawthorne</city>
                                 <state>NY</state>
                                 <zip>10532</zip>
                      </ShippingAddress>
                      <Terms>30 days, cash</Terms>
           </Customer>
           <Order>
                      <Item id="194103-011" quantity="1000">
                      <Item id="923012-832" quantity="45">
           <Item id="452722-023" quantity="45">
           </Order>
</PurchaseOrder>
```

# SOAP Message structure

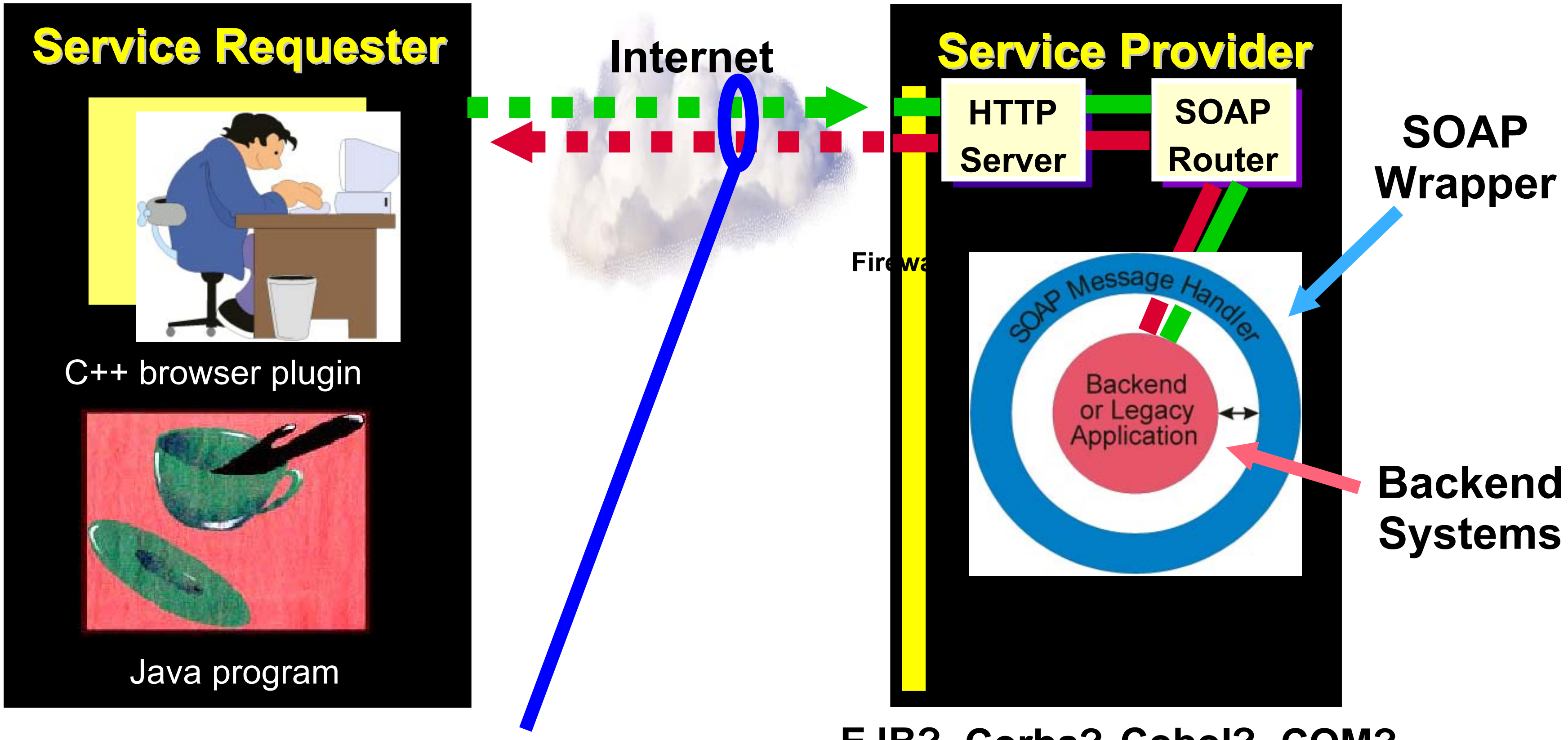
- Request and Response messages
  - Request invokes a method on a remote object
  - Response returns result of running the method
- SOAP specification defines an "envelope"
  - "envelope" wraps the message itself
  - message can be a different vocabulary

application-specific message vocabulary

SOAP envelope vocabulary



# SOAP hides the technology choices and implementation details from both parties



EJB? Corba? Cobol? COM?
SOAP can wrap any of these

**Simple, standard XML messages**

- we only care about message format and content
- the less we know about the implementation details, the less work for us!

# Why SOAP Will Succeed

Other distributed technologies failed on the Internet because they strongly coupled the endpoints:

- ✘ RMI - requires Java at each endpoint
- ✘ CORBA - requires compatible ORBs at each endpoint
- ✘ DCOM - requires Windows at each endpoint

✓SOAP is the platform-neutral choice

- ➔simply an XML wire format
- ➔places no restrictions on the endpoint implementation technology choices
- ➔implementations are free, some are open-source

# How do we define new web services?

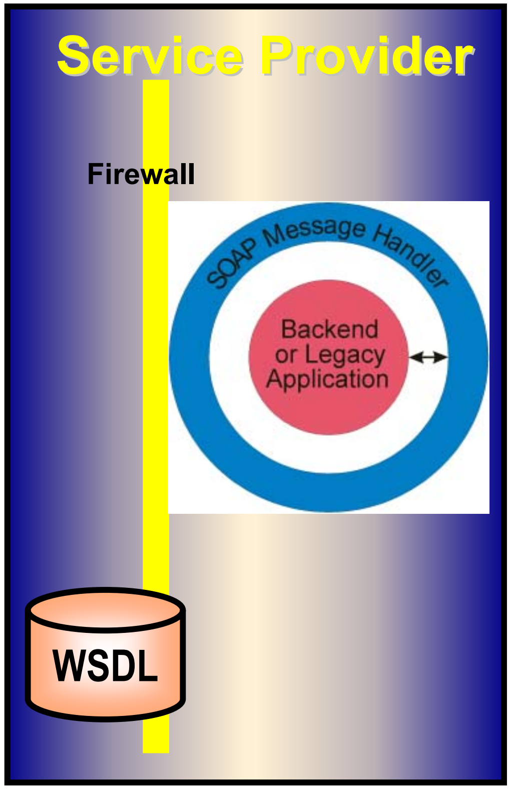
- **Refer to web services that others have defined and agreed upon (tModels).**
- **...but what if I want to publish my own web services**
  - e.g. simple functions useful to others
  - or complete business processes that I offer
- **To make it practical... and easy to integrate to many such services, we need a standard way of describing web services.**

**Web**
**Services**
**Description**
**Language**

# WSDL Simplifies and Speeds Integration

## Complete technical details required for integrating a Web service into your application

- ƒ available from service provider
- ƒ message format, URL, etc
- ƒ independent of programming language
- ƒ used by development tools for code-generation, programmer information
- ƒ key to rapid integration and productivity gains in Web Services programming model



# How does the Requestor get the WSDL?

- WSDL (or its URL) can be emailed to requestor
- find WSDL for available services at repository sites like xmethods.net or www.salcentral.com
- ...or use UDDI "find" methods to look it up in the UDDI Business Registry

# What is UDDI?

- Universal Description, Discovery, and Integration

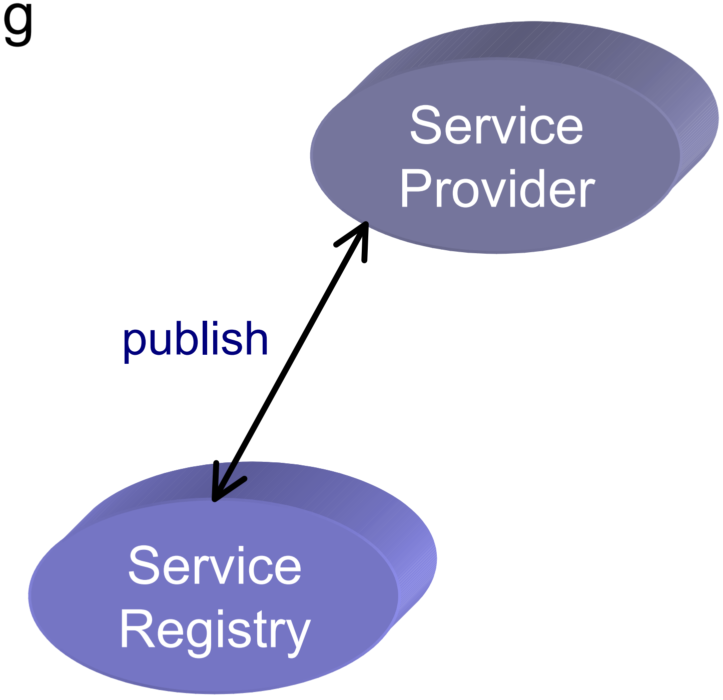
- A project to speed interoperability and adoption for web services by developing standards-based specifications for service description, publication and discovery.
- A set of Internet-based implementations, including the public UDDI Business Registries (UBRs) interoperating to share information.
- Private and semi-private registries implemented by companies and organizations.
- Partnership among industry and business leaders
  - Initiated by IBM, Microsoft, and Ariba with 12 other industry partners in September, 2000.
  - 300+ UDDI community members
- Specification work transferred to OASIS - July 2002
  - UDDI Member Section (IBM chair)
  - UDDI Spec TC (Microsoft, IBM co-chairs)

# UDDI roles and operations

- Service Registry
  - provides support for publishing and locating services
  - like telephone yellow pages

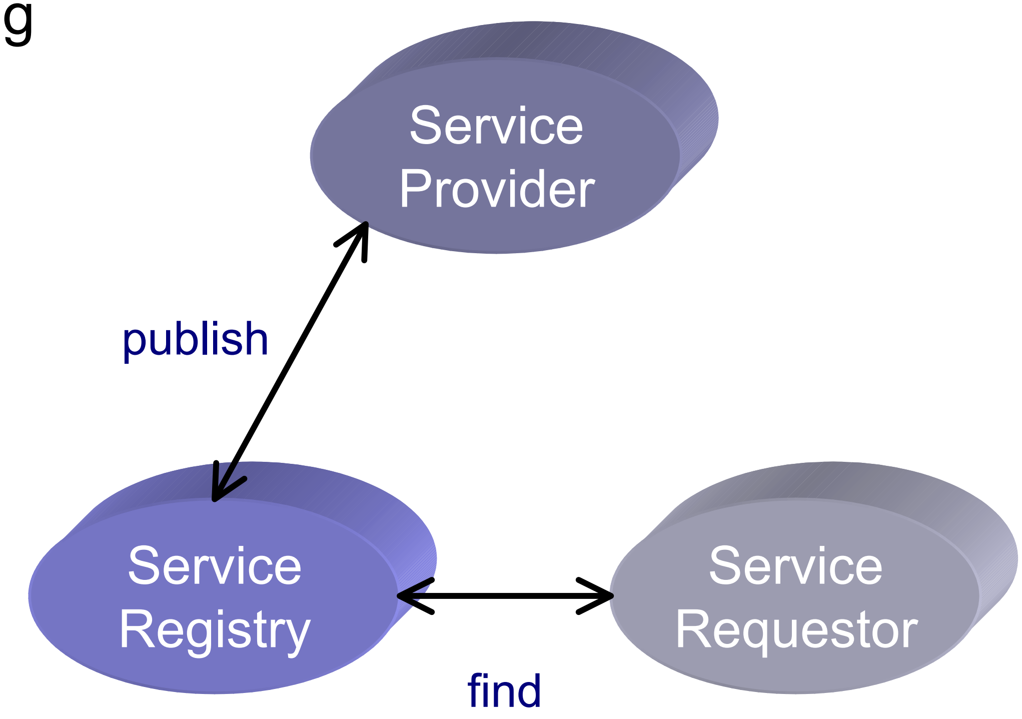
# UDDI roles and operations

- Service Registry
  - provides support for publishing and locating services
  - like telephone yellow pages
- Service Provider
  - provides e-business services
  - PUBLISHES availability of these services through a registry



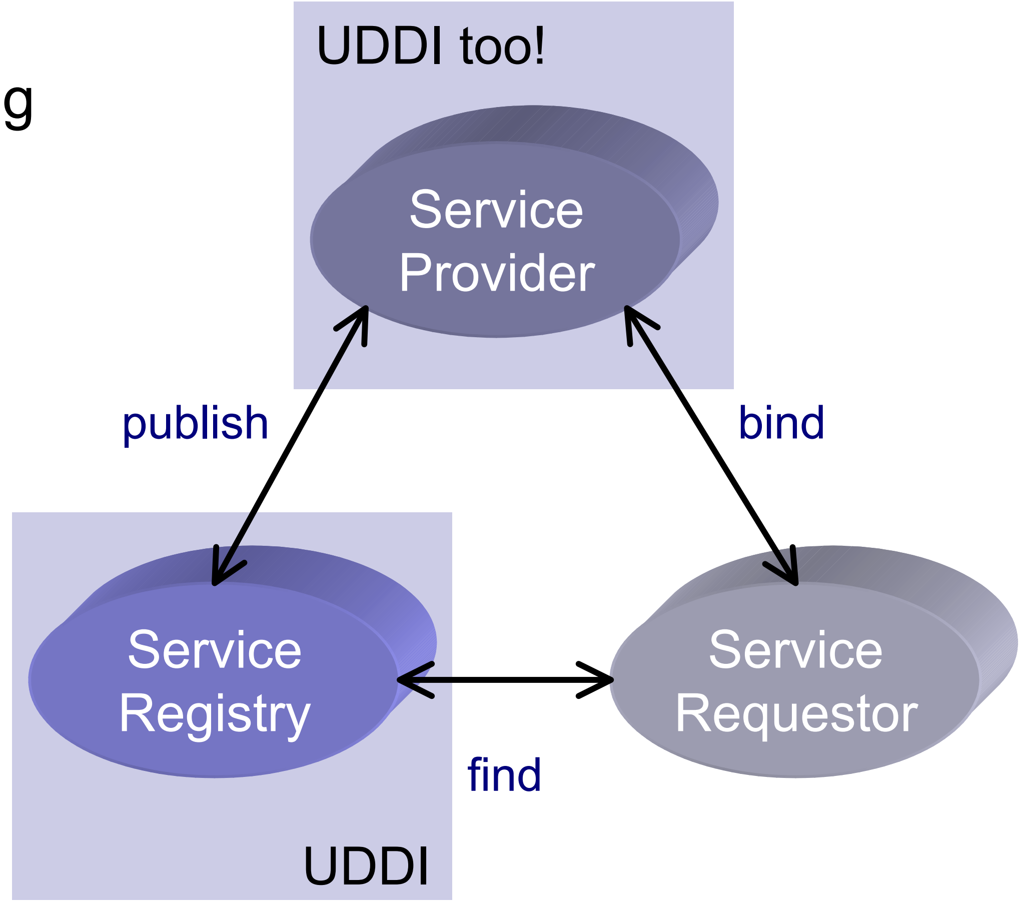
# UDDI roles and operations

- Service Registry
  - provides support for publishing and locating services
  - like telephone yellow pages
- Service Provider
  - provides e-business services
  - PUBLISHES availability of these services through a registry
- Service Requestor
  - FINDS required services via the Service Broker



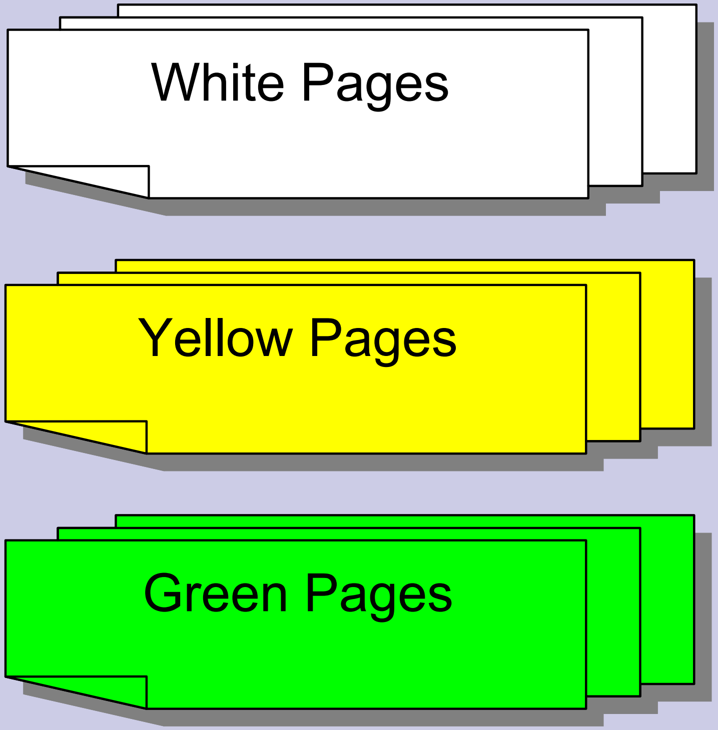
# UDDI roles and operations

- Service Registry
  - provides support for publishing and locating services
  - like telephone yellow pages
- Service Provider
  - provides e-business services
  - PUBLISHES availability of these services through a registry
- Service Requestor
  - FINDS required services via the Service Broker
  - BINDS to services via Service Provider

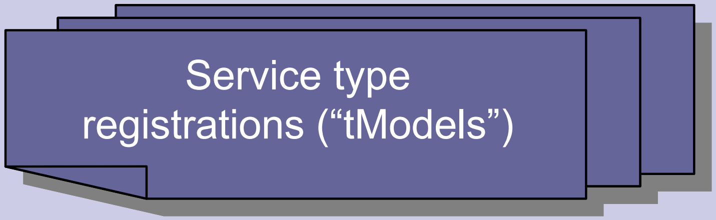


# UDDI registry data

Businesses register public information about themselves

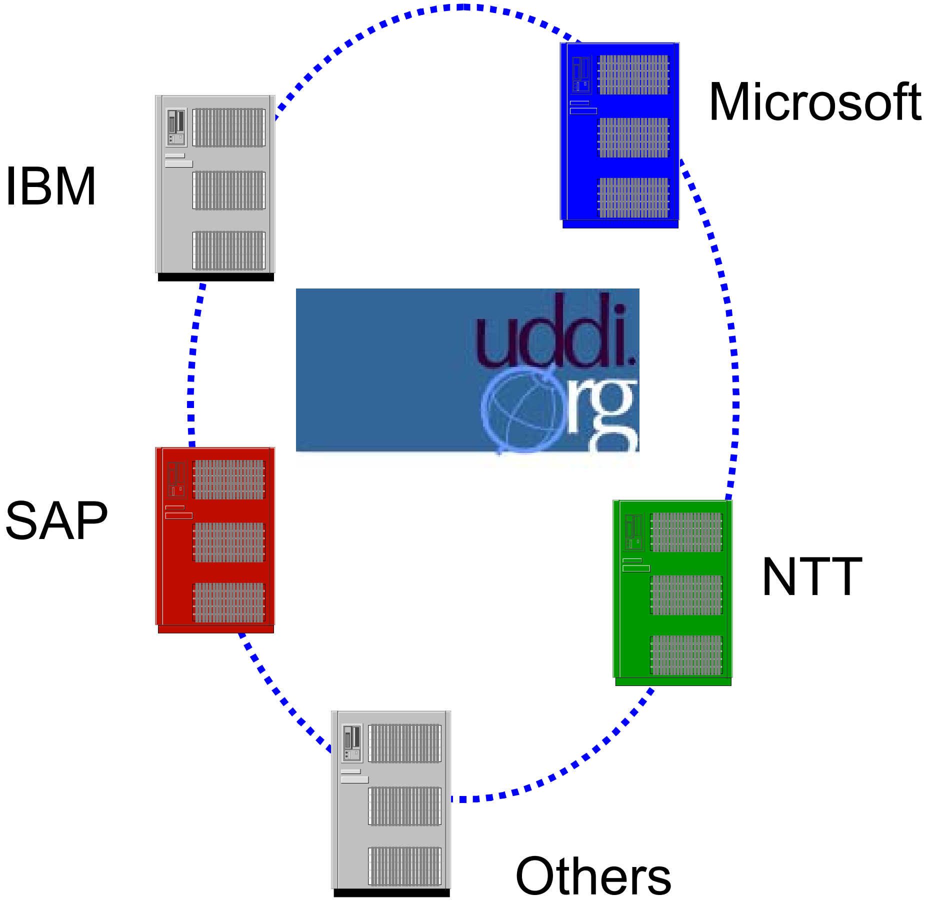


Standards bodies, programmers, businesses register information about their service types

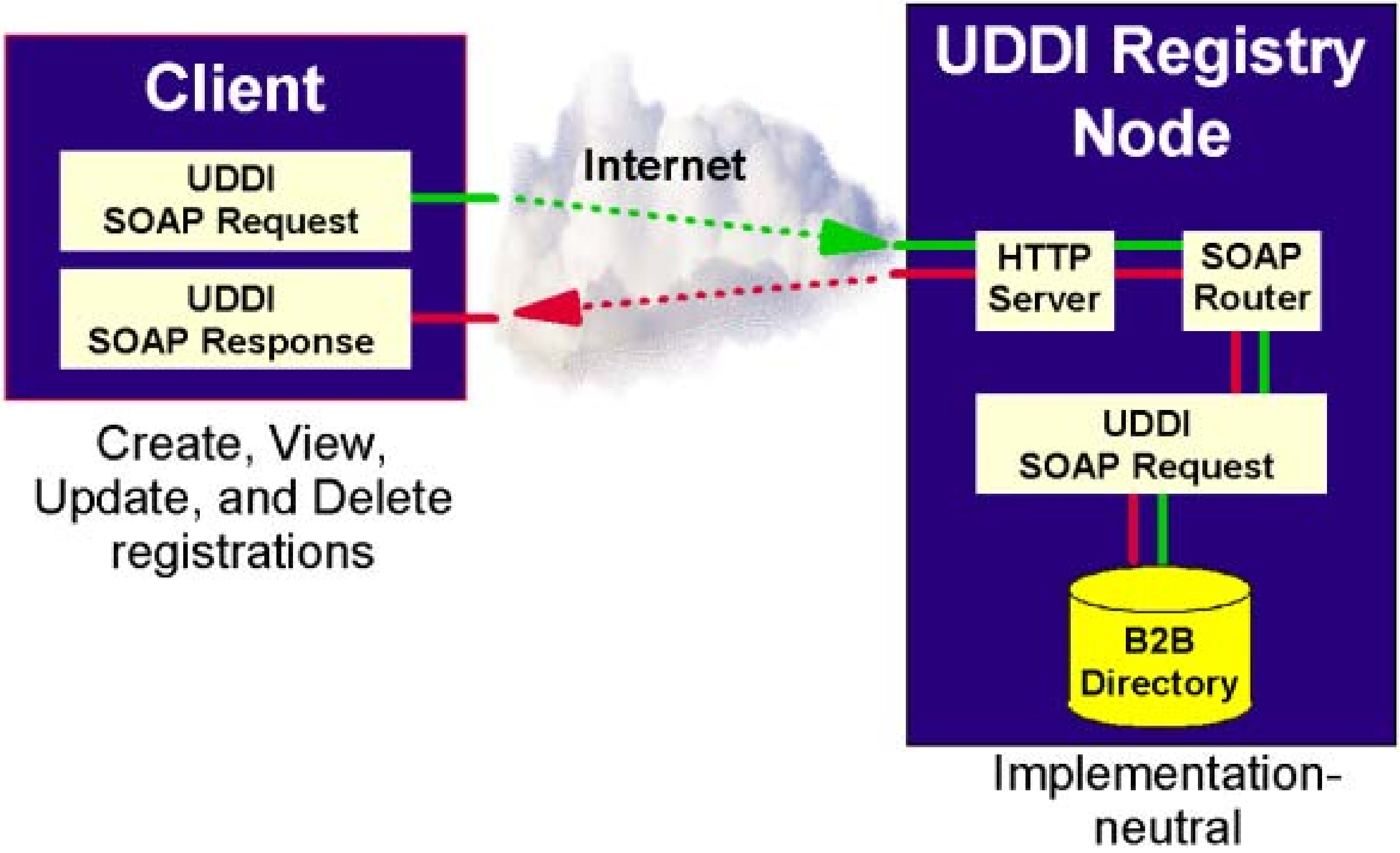


# UBR operation

- Peer nodes (websites)
  - Companies register with any node
  - Registrations replicated hourly
  - Complete set of “registered” records available at all nodes
- Common set of SOAP APIs supported by all nodes
- Compliance enforced by business contract
- UDDI v2 in "production" – July, 2002
- Operators Council now planning for v3



# UDDI and SOAP

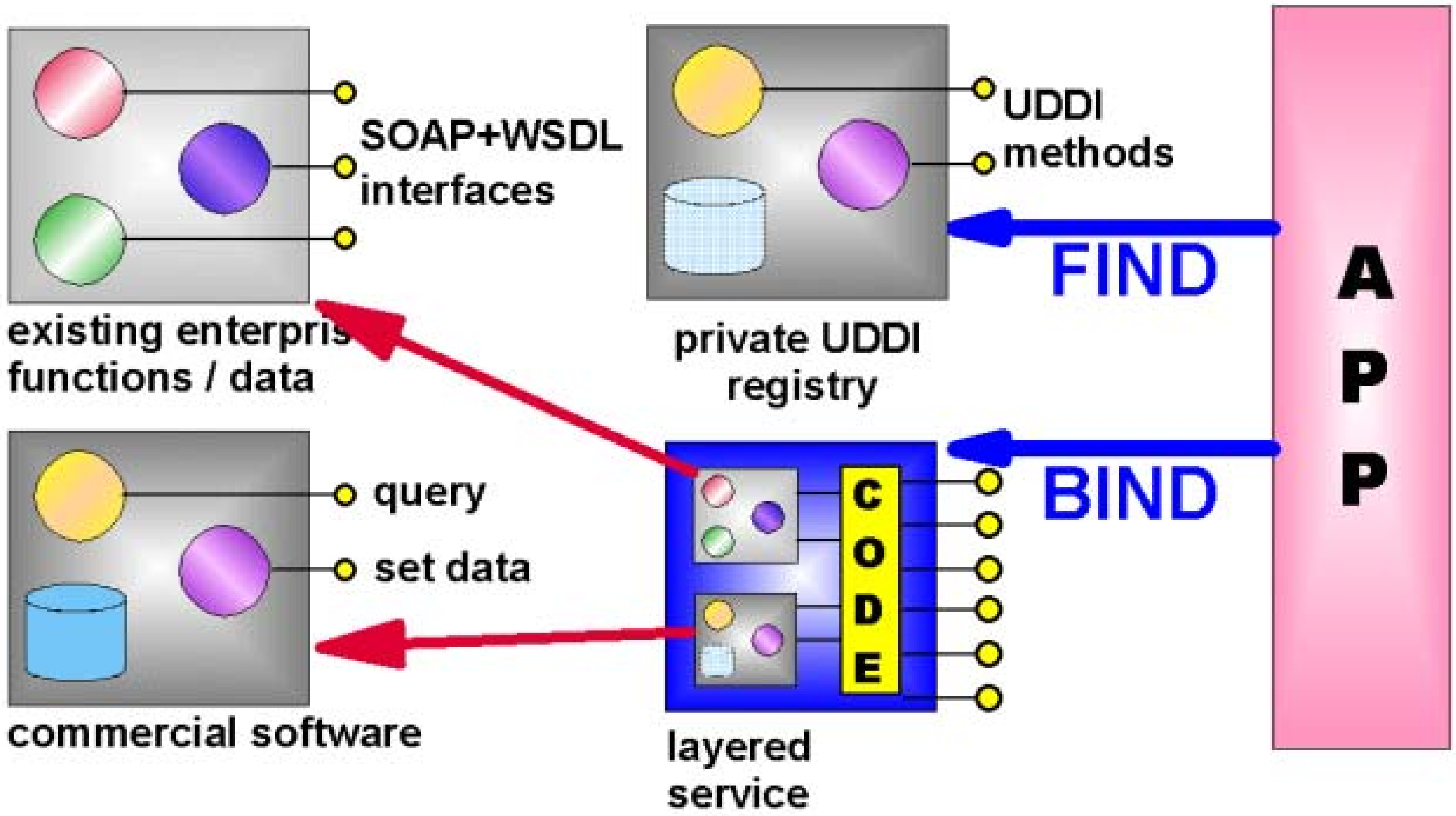


Create, View, Update, and Delete registrations

Implementation-neutral

# Web services inside the enterprise

SOAP+WSDL+UDDI is useful for an application or data integration strategy: offers loose coupling and late binding



# UDDI v. 3.0 – July, 2002

- Security
  - Support for Digital Signatures
- Multiple Registry Support
  - Topologies of public and private registries
- Advanced Data Management
  - Enhanced search capability
  - Better interpretation of query results
  - More meaningful descriptions of businesses and services
  - Easier management of existing data.
- Internationalization
  - Enhanced support for multinational corporations to describe their global operations across international business units
  - Addressing localization of UDDI data and services.

# UDDI transition process

- Transition Team formed by the UDDI Working Group
- RFP developed and distributed
  - Extensive list of relevant questions
  - List of potential receivers built through nomination
- Eight formal responses received
  - Almost all were viable "homes" for UDDI spec work
  - Several review sessions - detailed discussion of each response
  - Selection by closed ballot - OASIS selected
- Recommendation to Working Group accepted
- Some OASIS advantages
  - Member Section provides organizational flexibility
  - Affinity to other OASIS work on Web Services
  - Large percentage of UDDI members already OASIS members

# UDDI Summary

- V3 has established a solid design base
- OASIS is an ideal place for further work
  - TC will provide for broader participation
  - Focus on V2, V3 as OASIS Standards; V4 requirements collection
- UBR Operators Council continuing
  - V2 in production
  - V3 planning underwayC
  - Requirements feed to OASIS TC
- Customers already using/deploying UDDI
  - 10K entries in public UBR
  - Many private registries
- UDDI Tooling is readily available, from multiple vendors

# WebSphere Studio App Developer: speeding **deployment** of Web Services

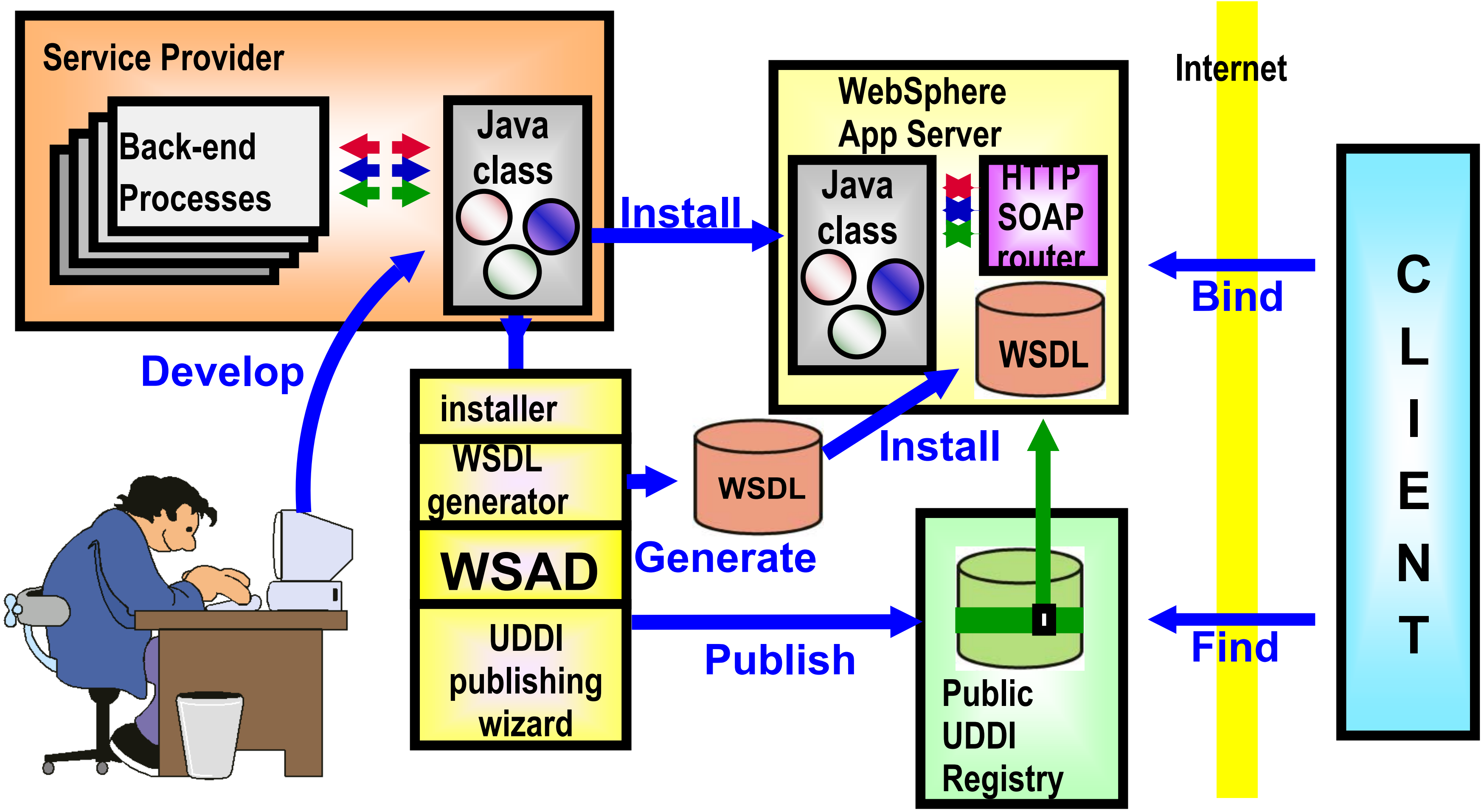
1 You **develop** a Java class for the service provider to be deployed

2 WSAD **generates** a service description by introspecting your class

3 WSAD **installs** code and WSDL description on the server

4 WSAD wizard **publishes** the availability of the service to UDDI

5 Client **finds** your service via UDDI then **binds** to your code



# WebSphere Studio App Developer: speeding integration of Web Services

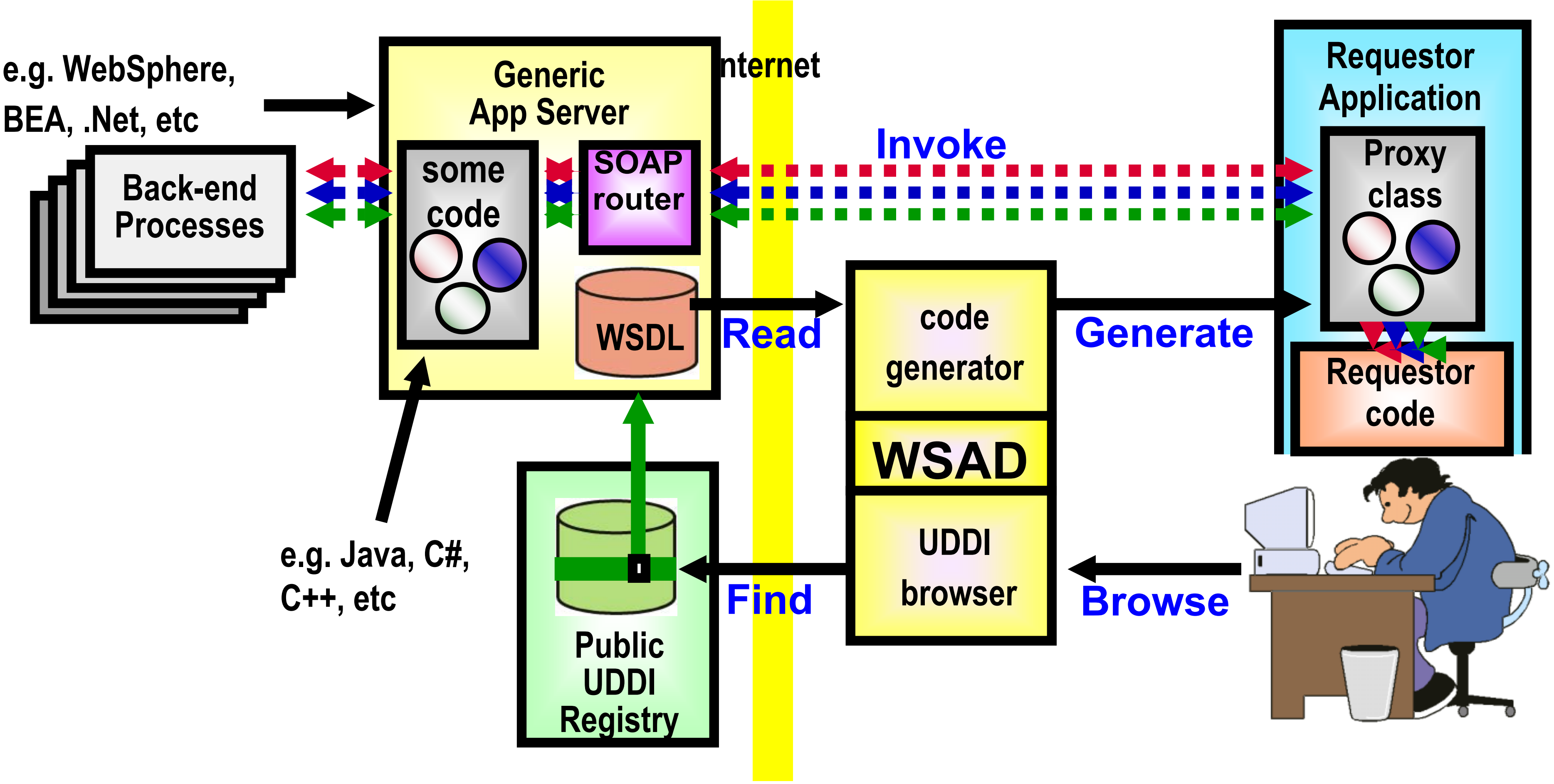
1 You use UDDI browser in WSAD to **find** the service you want

2 WSAD **reads** the service description and sets up environment

3 WSAD **generates** a Web service proxy class for local use

4 You **call methods** on the service proxy class just like local code

5 Service proxy class **invokes** the service for you via SOAP messages



# Web Services Interoperability

**WS-I.org announced Feb 6, 2002**

Industry initiative for Web services

- Open to any organization committed to Web services
- Promote and accelerate adoption, deployment

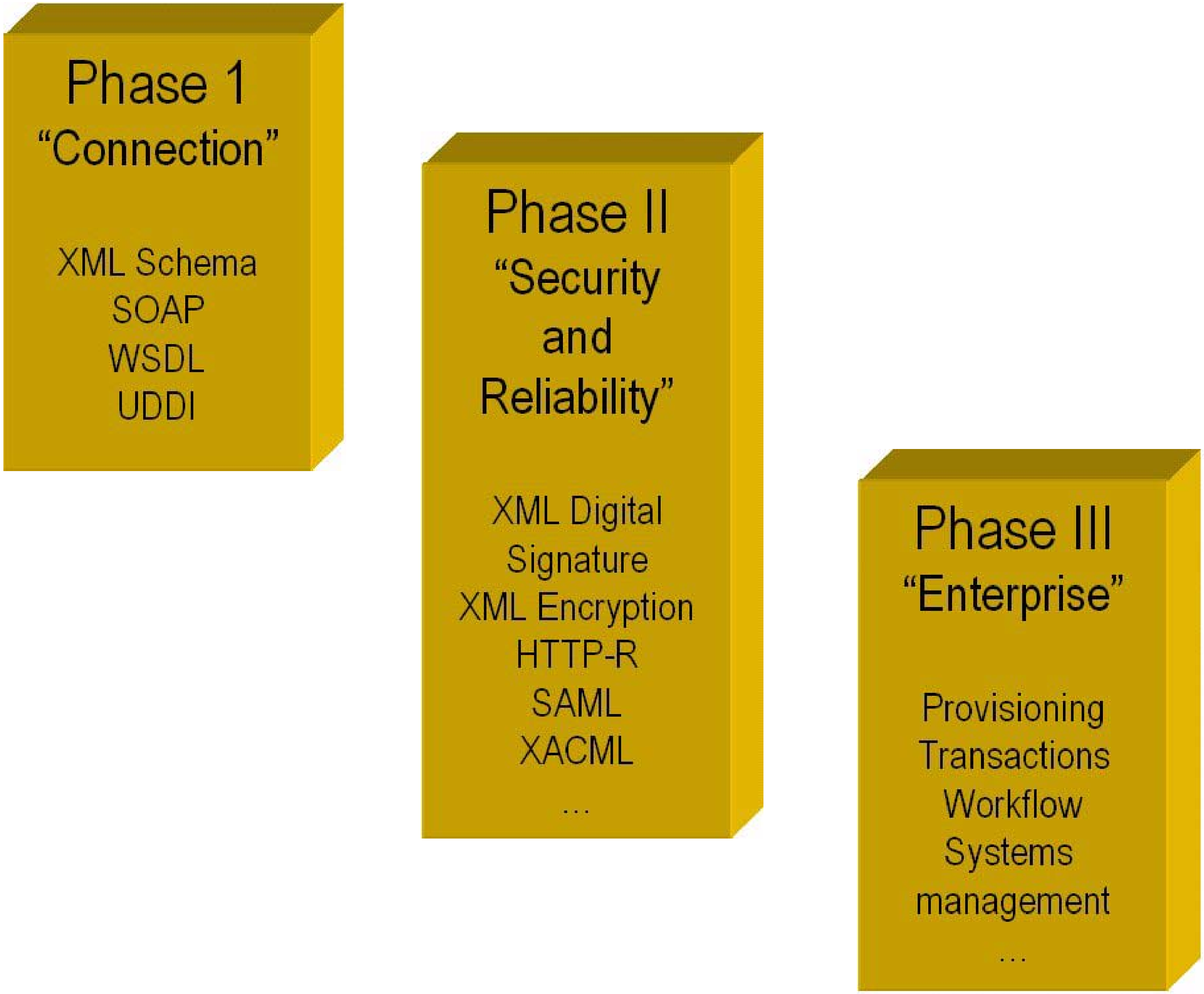
Drive seamless interoperability of Web service implementations

- Across platforms, applications, and programming languages
- Promote a common, clear definition for Web services

Promote customer adoption & deployment

- Integrate specifications from standards bodies by creating "profiles" based on specifications and standards
  - Implementation guidance & tools for customers building and deploying Web services

# Specifications and Standards



Phase 1
"Connection"

XML Schema
SOAP
WSDL
UDDI

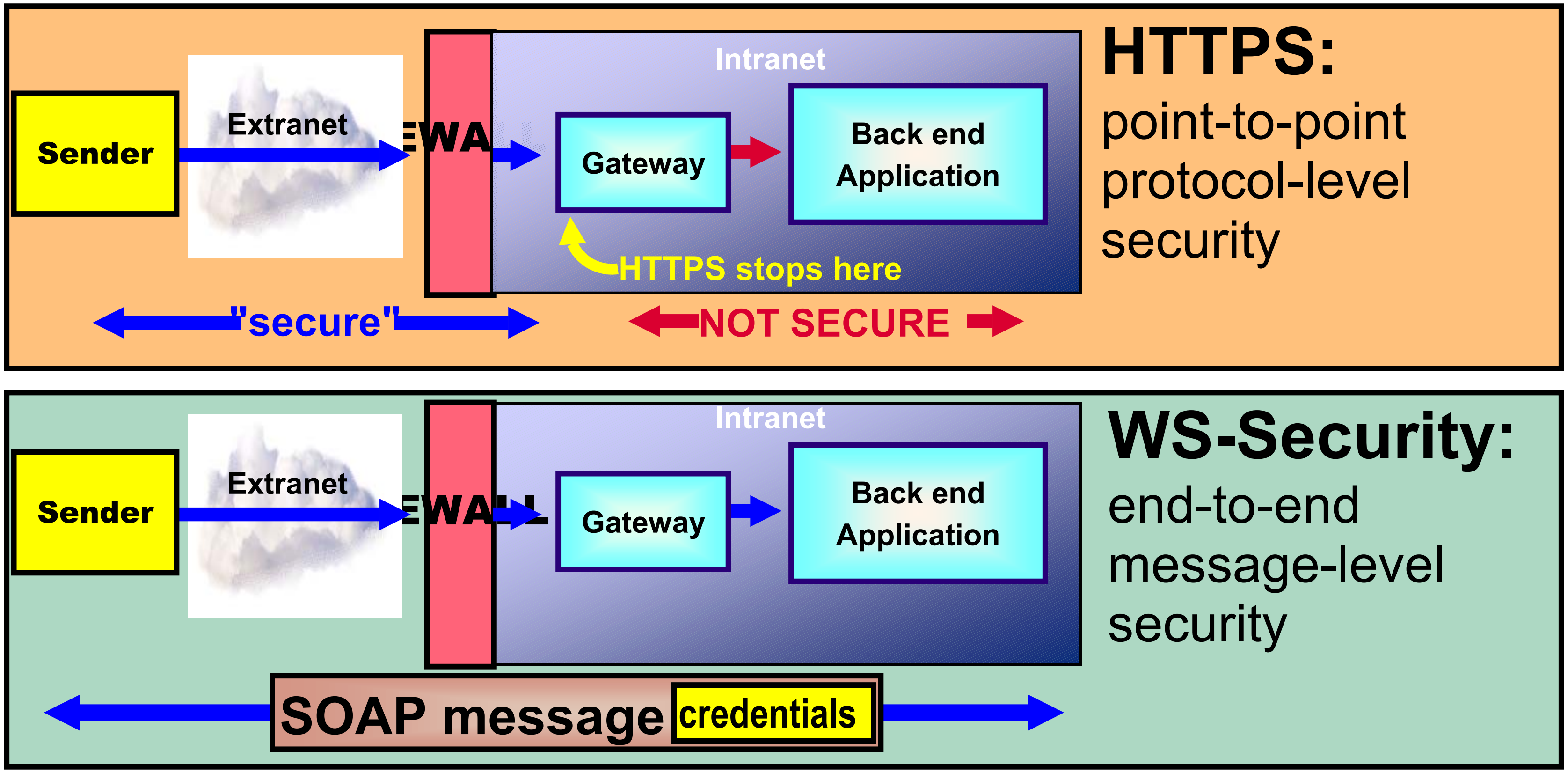
Phase II
"Security and Reliability"

XML Digital Signature
XML Encryption
HTTP-R
SAML
XACML
…

Phase III
"Enterprise"

Provisioning
Transactions
Workflow
Systems management
…

# WS-Security: message-level security



## Message-level security

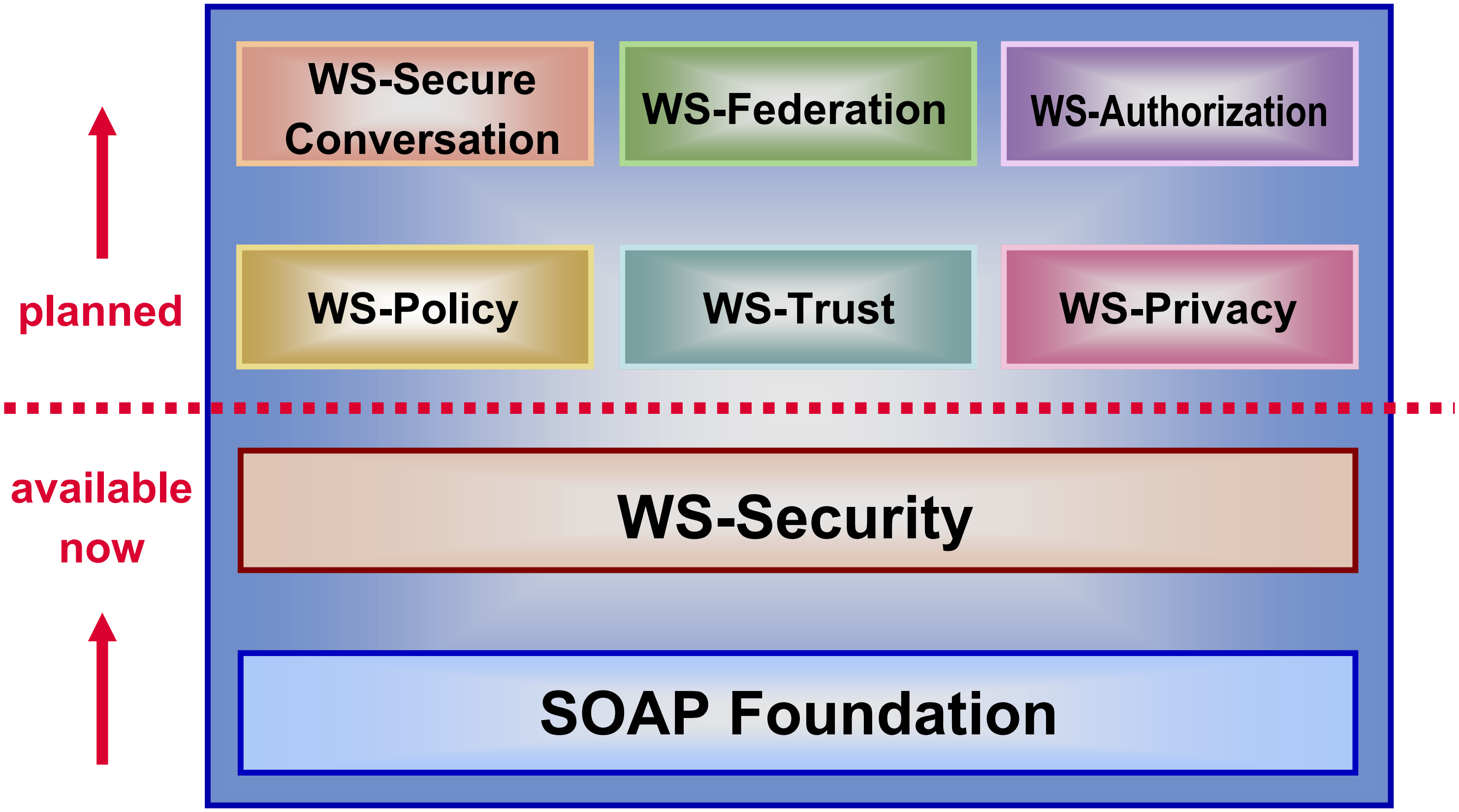
- credentials persist end-to-end
- allows non-repudiation
- element-wise encryption

## Now interoperable with Web services:

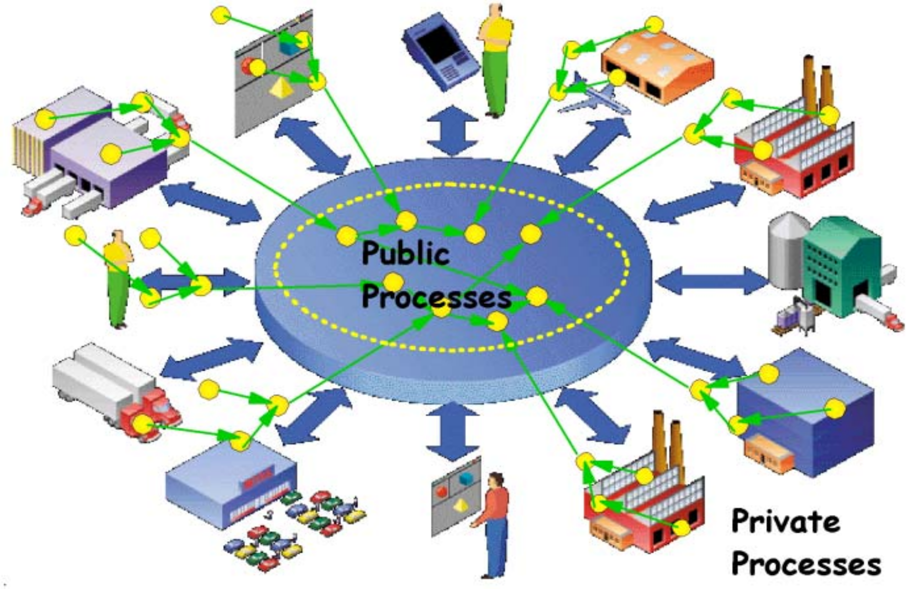
- Kerberos, PKI, X.509, HTTPS/SSL
- W3C XML Signature, XML Encryption, XKMS
- OASIS SAML, XACML

# WS-Security 1.0 Roadmap

Six more security technology proposals will be published in the coming months to address emerging security concerns



# Business Processes



## BPEL4WS

- Business Process and Execution Language for Web Services
- specifies operations, data, partners for a business model in a portable XML description

## WS-Transaction

- define transaction model based on SOAP messaging

## WS-Coordination

- standard mechanisms that coordinate the execution of distributed transactions in a Web services environment

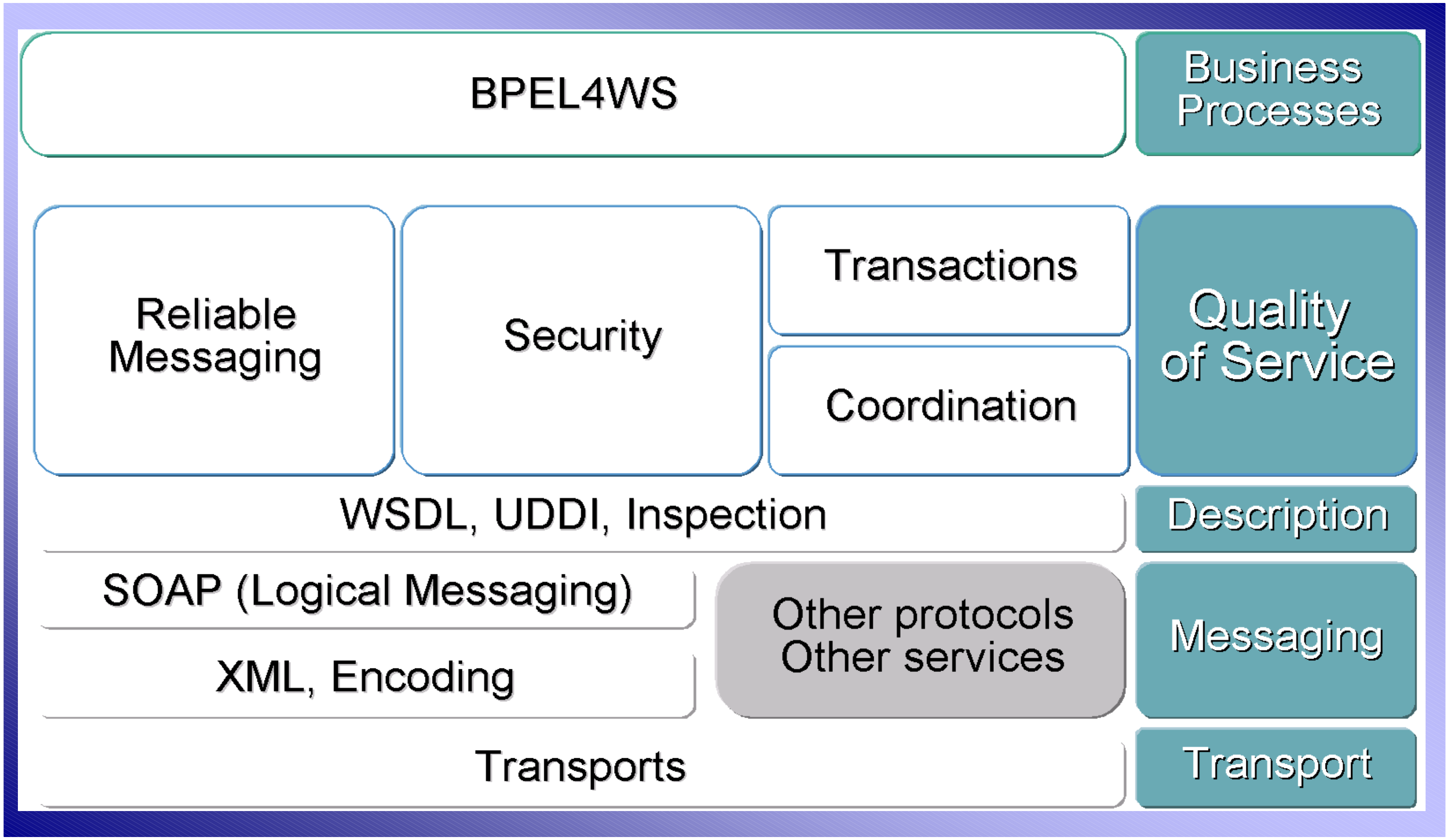
## Specifications announced August 8, 2002 by IBM, Microsoft, and BEA

- will be moved to a standards body

## Visit ibm.com/developerworks/webservices

- specs and whitepapers available now

# The Web Services "stack"



BPEL4WS

Business Processes

Reliable Messaging

Security

Transactions

Coordination

Quality of Service

WSDL, UDDI, Inspection

Description

SOAP (Logical Messaging)

XML, Encoding

Other protocols Other services

Messaging

Transports

Transport

- WS-Security was announced in April, 2002
- BPEL4WS, WS-Transaction, WS-Coordination were announced in August, 2002

# IBM's Web Services Strategy

**Mission: Deliver Web Services solutions that will help our customers and business partners build, deploy, and manage e-business applications.**

**We are doing this by**

- ƒ **Ensuring strong, open standards**
- ƒ **Enabling our entire product line for SOAP, UDDI, WSDL, and emerging Web Services technologies**
- ƒ **Building e-business solutions**

# Who is Using Web services today?

## Many customers are using Web services now

- to build a flexible enterprise infrastructure to solve today's problems while simplifying future development
- to reduce the cost of doing business with existing partners
- to prepare for more flexible e-business of the future

## Available today

- middleware products
- developer tools
- services

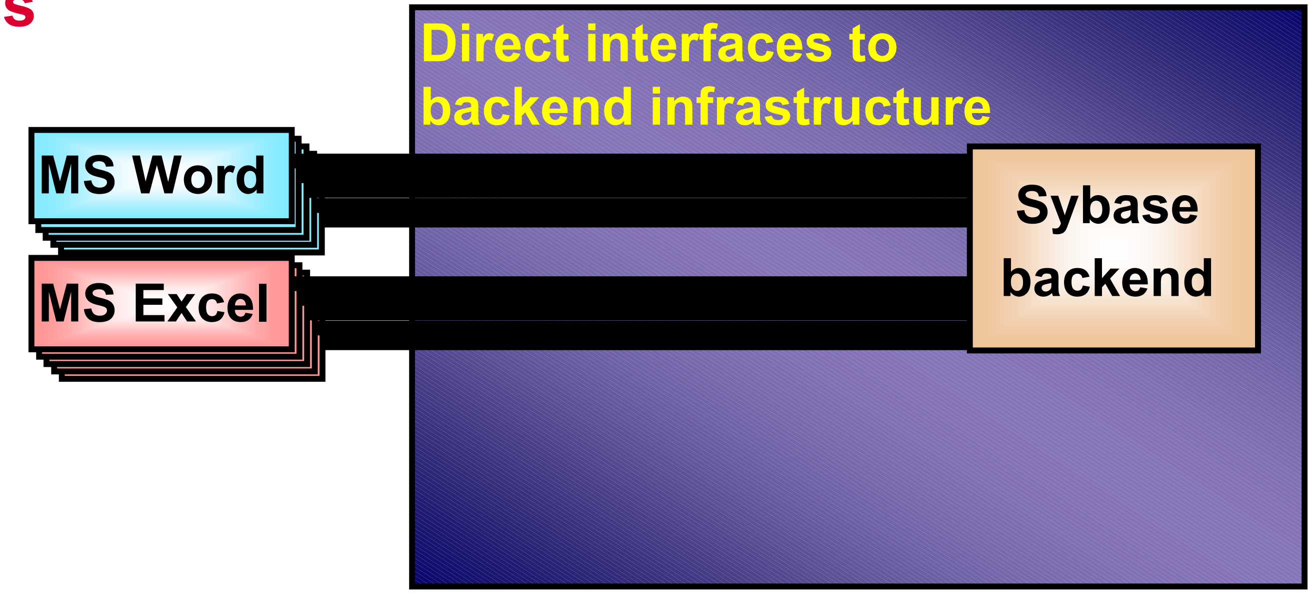
**visit http://ibm.com/software/jstart for case studies**

# Canadian Imperial Bank of Commerce

**Challenges:** Achieve interoperability and reusability for all financial data services

- fix income coupon schedules
- equity information
- moving markets
- other static descriptive instrument data that is not price dependant

Migrate to new backend systems with no disruption to existing applications



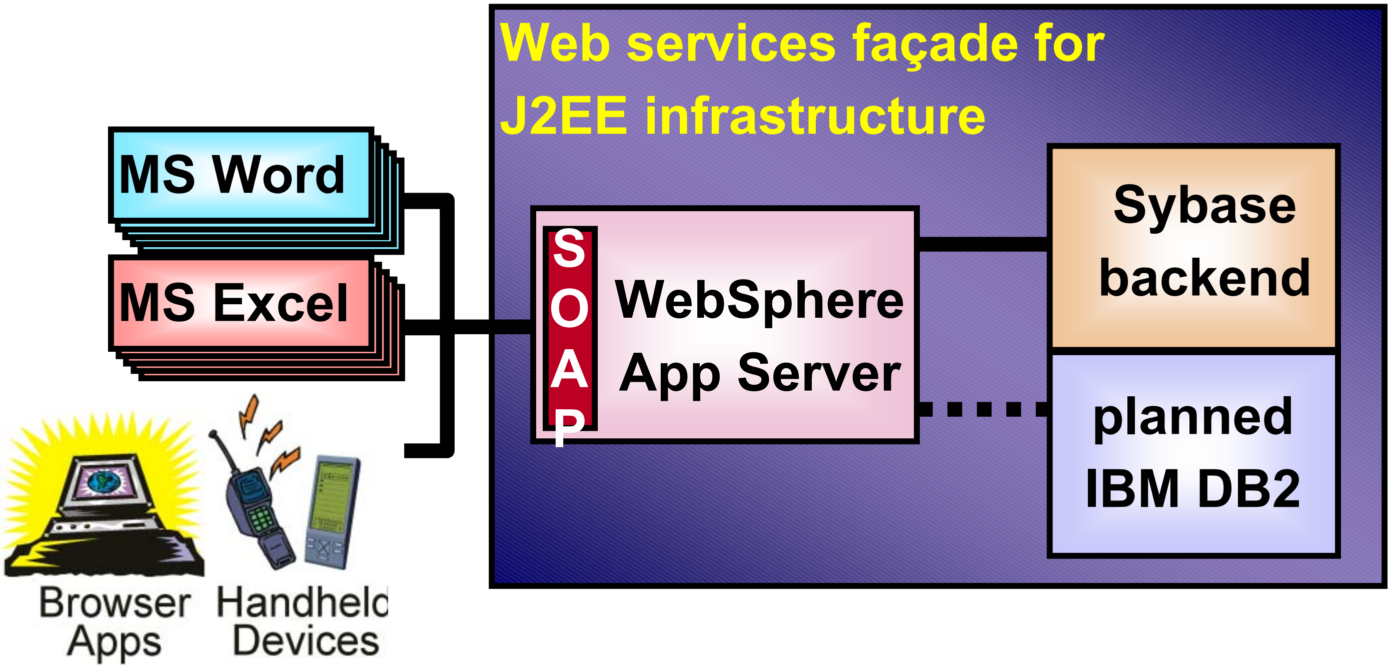
# Canadian Imperial Bank of Commerce

**Solution:** **Customer Account Data Service available across all lines-of-businesses**

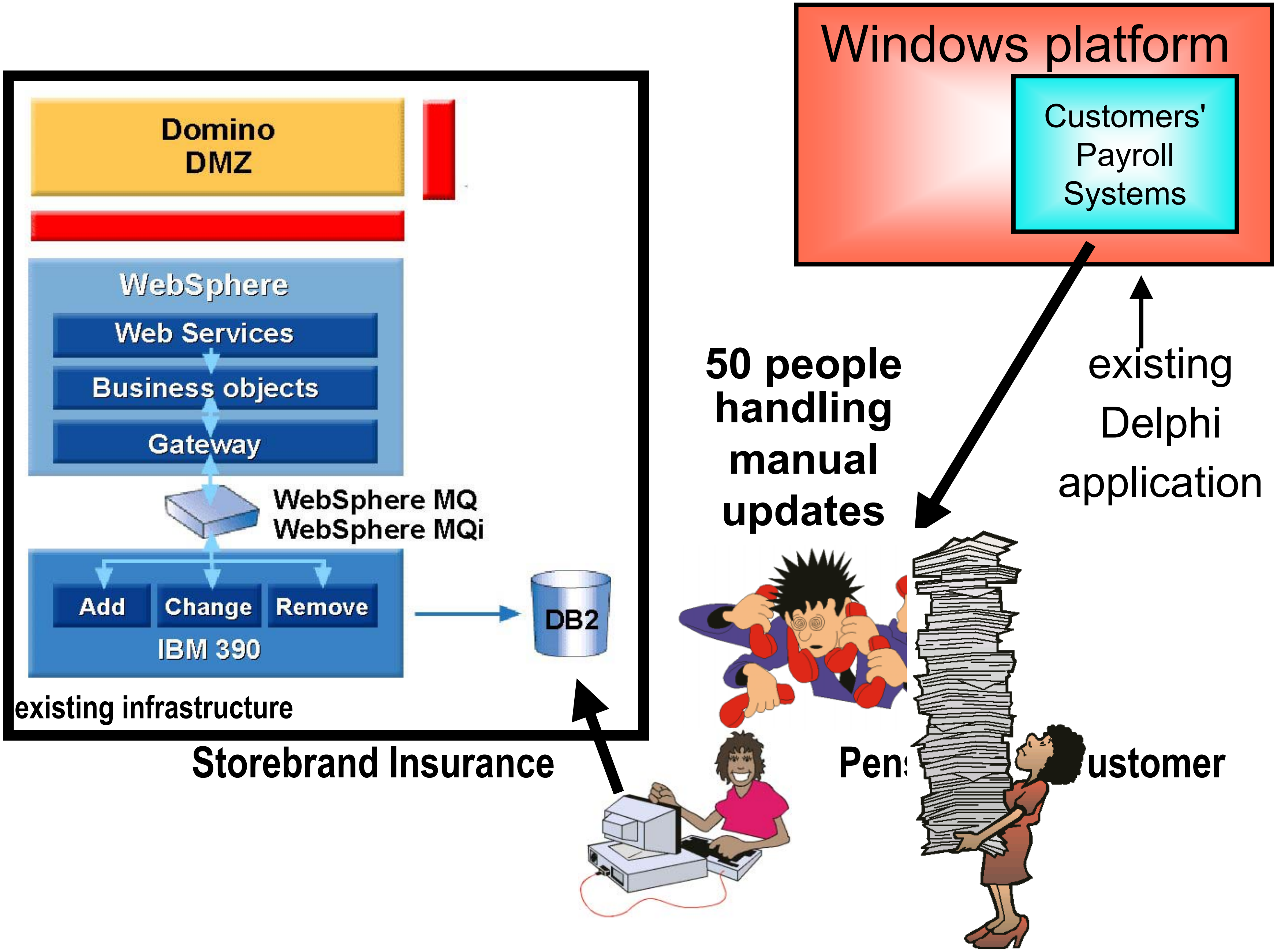
**The benefits:**

- “One interface, many clients."
- No change to client apps from database migration

http://www-3.ibm.com/software/ebusiness/jstart/casestudies/cibc.html



# Storebrand ASA - before Web services



Domino DMZ

WebSphere

Web Services

Business objects

Gateway

WebSphere MQ
WebSphere MQi

Add

Change

Remove

IBM 390

DB2

existing infrastructure

Storebrand Insurance

Windows platform

Customers' Payroll Systems

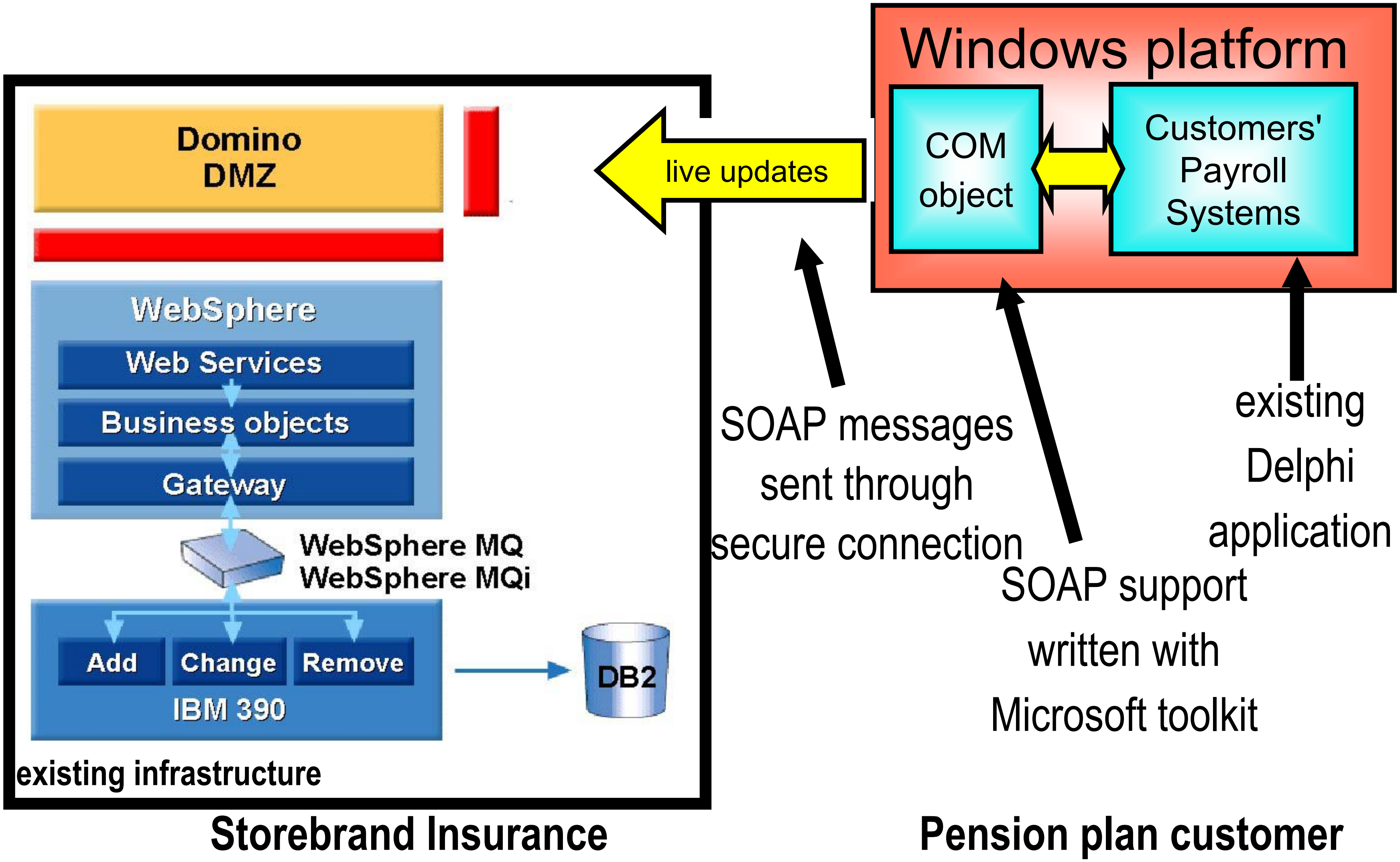
existing Delphi application

50 people handling manual updates

Pens

ustomer

# Storebrand ASA - after Web services



**Read all about it!**

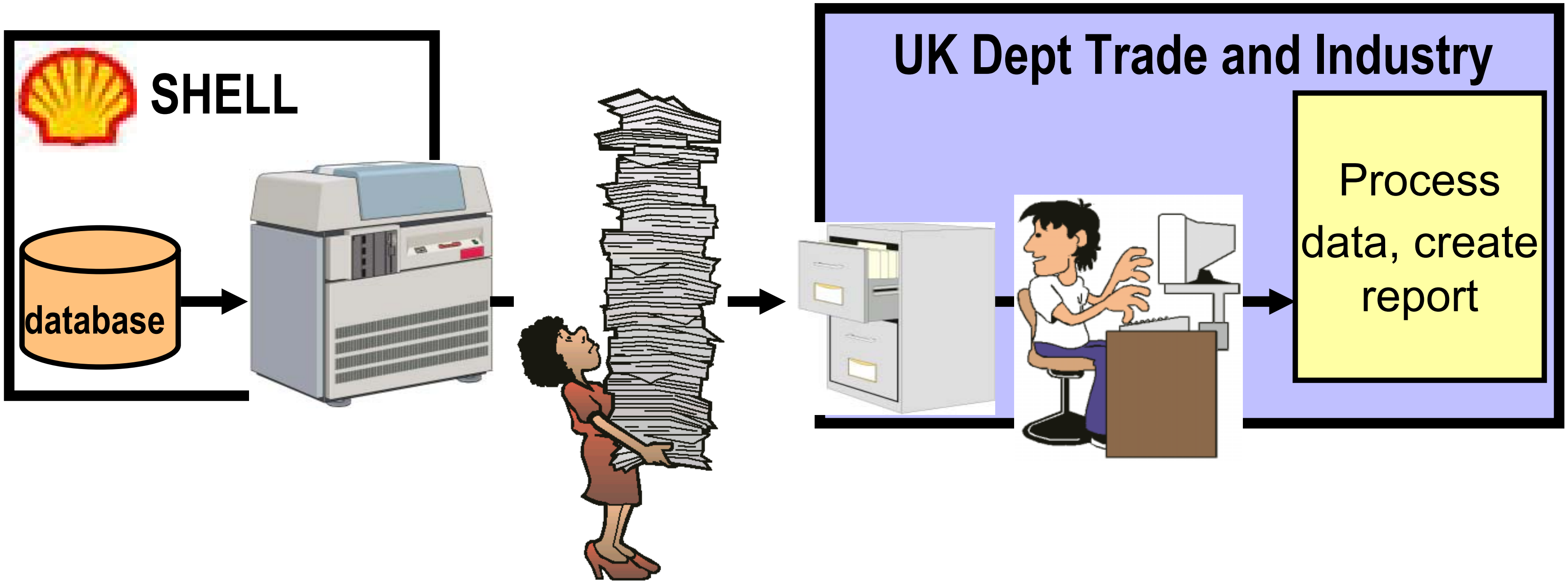
http://ibm.com/developerworks/webservices/library/ws-asa

# UK DTI - Royal Dutch/Shell

## UK Department of Trade and Industry requires reports for oil drilling sites it licenses

- well information: production, quality, hazards
- information is maintained on databases of oil companies, printed in full "**just in case**" it is needed
- when needed, DTI searches paper files and, if found,

manually enters information for reporting purposes



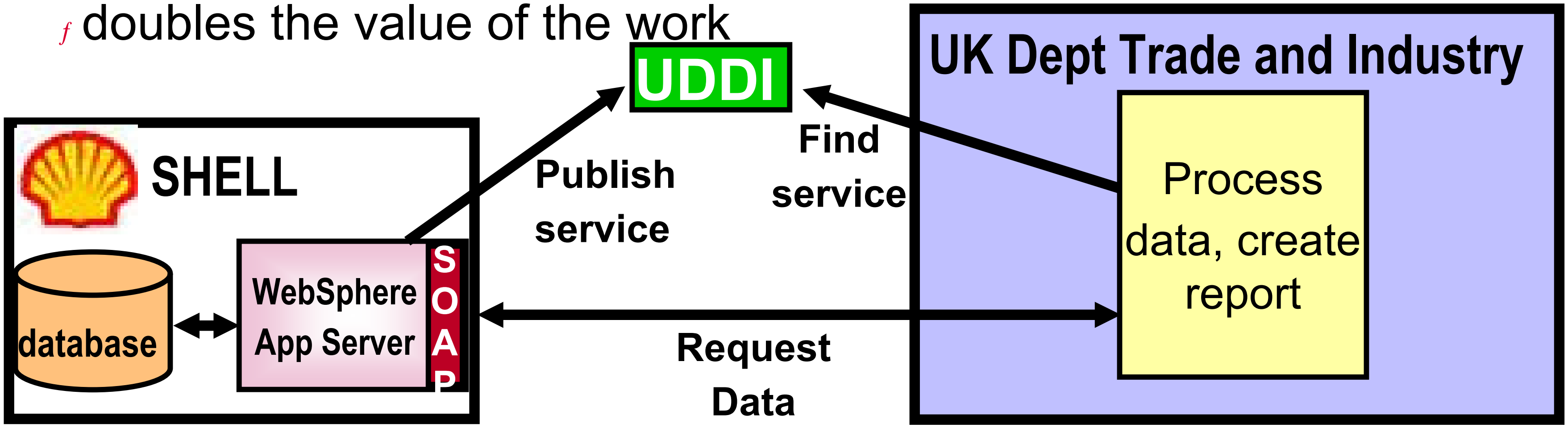
# UK DTI - POSC - Royal Dutch/Shell

## Shell and POSC will create direct Web services access to information needed by UK DTI

- "**just in time**" replaces "just in case"
- eliminate manual processes like searching and entering
- automate reporting; more accurate and timely information

## Same information is needed by drilling partners

- interfaces implemented for UK DTI can also be used by partners
- doubles the value of the work



**Read all about it!**

http://www-3.ibm.com/software/ebusiness/jstart/casestudies/shell.html

# Summary

## UDDI:

- Key enabler for Web Services
- Provides information necessary for dynamically integrating business processes.
- Useful both inside and outside your business.

## Web services:

- can speed development with a more flexible infrastructure
- can make you more profitable by reducing the cost of doing business with existing customers today and new customers later
- helps you find and quickly integrate with new business partners
- leverages your existing software investmentscan give you the agility to cope with changes in business requirements, or experiment with better business processes

# UDDI Resources at UDDI.org / OASIS

- Specifications
  - Version 2
  - Version 3
- White Papers:
  - Executive White Paper
  - Technical White Paper
- Best Practices papers:
  - Using WSDL in a UDDI Registry
  - Providing a Taxonomy for use in UDDI version 2.00

# More UDDI Resources

- **White papers, product offerings**
  - http://www.ibm.com/webservices
- **Software:**
  - UDDI4J - open-source Java API to access UDDI
    - code: http://oss.software.ibm.com
  - Private UDDI preview for developers edition
  - http://www7b.software.ibm.com/wsdd/downloads/UDDIregistry.html
  - Web Services ToolKit (WSTK)
    - http://www.alphaworks.ibm.com/tech/webservicestoolkit
- **Articles, tutorials: http://ibm.com/developerworks/webservices**
  - Steve Graham: Role of private UDDI nodes in Web services
    - Part 1: Six species of UDDI
    - Part 2: Private nodes and operator nodes
  - Doug Tidwell: Introduction to UDDI4J
    - ibm.com/developerWorks/library/ws-uddi4j.html

# Web Services Resources

- **Register for the Web services newsletter at:**
  - www.ibm.com/developerworks/newsletter/
- **Check out the Web services Zone at:**
  - www.ibm.com/developerworks/webservices/
- **Attend a local seminar or workshop:**
  - www.developer.ibm.com/spc/events
- **Need help getting started? Contact jStart at:**
  - www.ibm.com/software/ebusiness/jstart/
- **Free! Dev Tools and Resources from Apache and IBM**
  - download PDF from ibm.com/developerworks/speakers/colan
- **Web services on WebSphere (WoW) partner program:**
  - www.ibm.com/websphere/wow/
- **Get WebSphere Studio:**
  - www-3.ibm.com/software/info1/websphere/index.jsp