



Web Services NOW!

why NOW is the time to get started with Web Services

@ John Lord

Consulting IT Architect

IBM Software Group Strategy

jglord@us.ibm.com

Enterprise Application Integration: 1990

Not much had changed...

- ▶ A variety of network types are in use
- ▶ File formats still compiled into programs

Enterprise Application Integration is difficult, time-consuming, expensive!

- ▶ the legacy of rigid and proprietary file formats, platform, programming language, and operating system dependence are the culprits
- ▶ ERP systems rise to solve problems with integrating business process software applications
- ▶ Companies revise business processes to fit the software

But isn't this backwards?

- ▶ Software should model the business process, not the other way around!
- ▶ EDI based on rigid message formats, private networking

B2B Integration: 1990

EDI is the only standard...

- ▶ Based on private networking (not the internet)
- ▶ Expensive, time-consuming to build and integrate
 - Rigid message formats (XML doesn't exist yet) is one culprit
- ▶ Is EDI a success?
 - Used primarily by the largest companies in the most industrialized nations
 - Wide-spread use? 5% of businesses
 - what about small- and mid-sized companies?
 - what about developing nations?

Early B2B integration

- ▶ months or years of development
- ▶ once you have integrated with a supplier, you're dependent on them
- ▶ what do you do if they take advantage of you?

B2B and EAI with XML

Flexible file formats

- ▶ systems less likely to break as software evolves
- ▶ easier integration

Information: messages and documents

Interoperability: sharing data across applications and platforms

Integration: bringing together data from multiple sources

XML data is Independent of:

- ▶ hardware platform
- ▶ operating system
- ▶ programming language

XML is in UNICODE, so it is international

XML defines the data format for content

- ▶ but what about the exchange standards?
- ▶ attend "Technical Overview of Web Services" to find out

XML Standards



Core Technology Standards

IBM participation includes XML, Schema, DOM, XSL, Namespaces, Linking, XHTML, RDF, XHTML, XML Protocol (SOAP), and XML Query.



IBM alphaWorks

Providing early access to emerging technologies to developers.



Accelerating the adoption of industry standards

oasis-open.org

100+ member companies including IBM, Sun, Microsoft, Corel, Software AG, and Oracle.



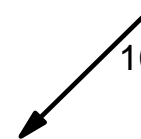
The XML Industry Portal

Sponsored by IBM, Sun, Oracle, SAP, ...
A vendor-neutral XML schema clearinghouse.
Info on how to apply XML in industrial and commercial settings.



UN/CEFACT

United Nations Centre for the Facilitation of Procedures and Practices for Administration, Commerce and Transport
www.unece.org/cefact/



Enabling a Global Electronic Market
ebxml.org



XML: where the industry is now

Many important base technical standards are now W3C recommendations (<http://w3.org>)

- ▶ XML Schema is a W3C Recommendation (May 2001)
- ▶ XSL Formatting Objects is now a W3C Recommendation
- ▶ XML Signature is a W3C Proposed Recommendation

OASIS has established XML.ORG as the registry and repository for industry-specific vocabulary standards

xml.apache.org has robust, mature implementations of XML parser, XSL processor, and SOAP4J as open-source implementations, and much more
countless products using XML from all vendors

- ▶ IBM products: XML features in WebSphere (etc) are mature; XML Features available for 3+ releases

Web Services: A Simple View

“Web services” is how

- ▶ businesses describe functionality (services) they want to externalize
- ▶ businesses publish that information
- ▶ businesses discover services
- ▶ businesses connect to each other and invoke services with appropriate security, reliability, and confidentiality

If XML defines a platform-independent way of representing data,

- ▶ making data integration easy and standard

...then Web services defines a platform-independent way of exchanging that data.

- ▶ process-level integration becomes easy

This is all moving very quickly because, basically, **it is a really good idea.**

A Universal Internet Programming Model

Share functionality and information on the Web, regardless of

- ▶ Operating system
- ▶ Hardware or delivery device
- ▶ Programming language
- ▶ Distributed object system
- ▶ Database or other back-end system

Direct program-to-program integration for

- ▶ Business-to-business applications
- ▶ Enterprise Application Integration
- ▶ Reusable components for interactive applications
- ▶ Mobile applications
- ▶ Grid computing
- ▶ ...and it's general enough to handle anything else that requires integration across a network

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.

Key Web Services Technologies

SOAP: Simple Object Access Protocol

- ▶ XML as a message protocol

WSDL: Web Services Description Language

- ▶ describe integration requirements

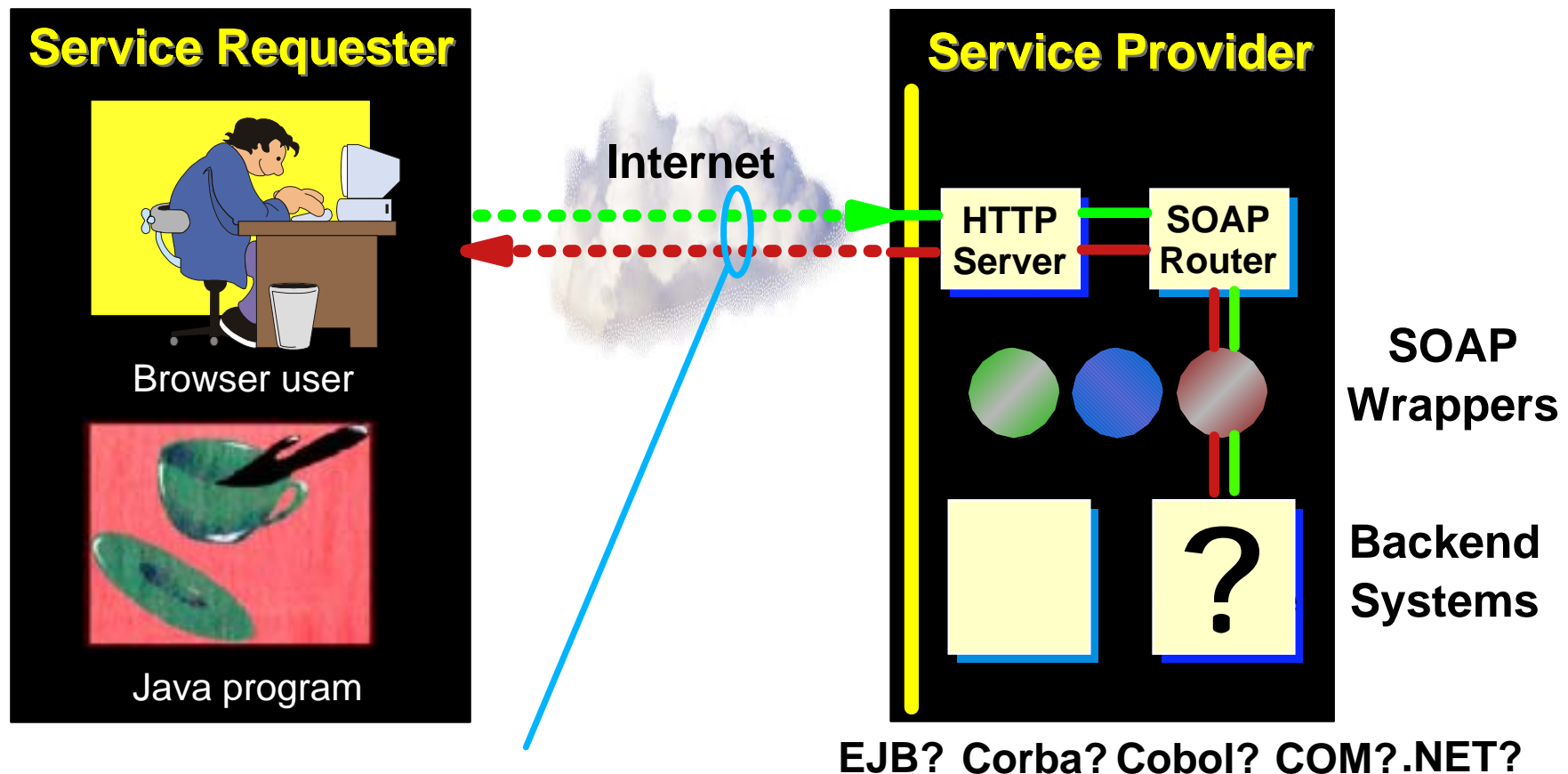
UDDI: Universal Description, Discovery, Integration

- ▶ publish and find available services

Security and Reliability

- ▶ Protocol-level vs message-level security
- ▶ Reliable transactions with HTTPR

SOAP hides the technology choices and implementation details from both parties



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 and therefore could not become pervasive:

- × **Unix RPC** - requires binary-compatible Unix implementations at each endpoint
- × **CORBA** - requires compatible ORBs
- × **RMI** - requires Java 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

UDDI Roles and Operations

Service Registry

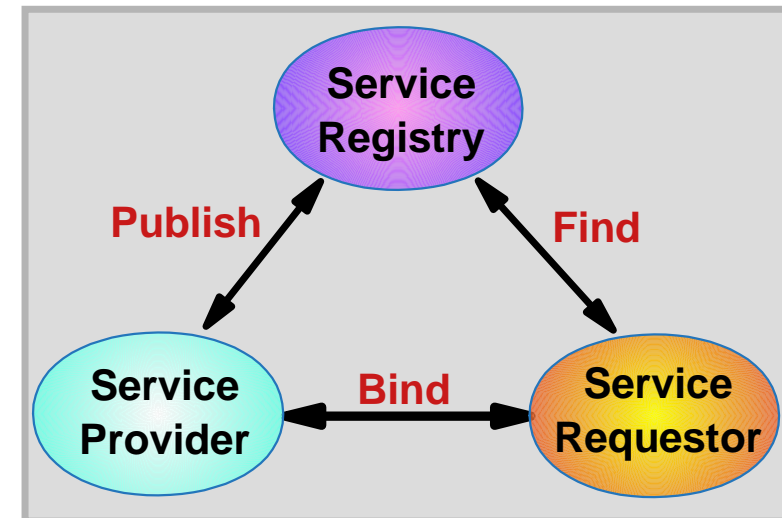
- ▶ provides support for publishing and locating services
- ▶ like telephone yellow pages

Service Provider

- ▶ provides e-business services
- ▶ **PUBLISH**es availability of these services through a registry

Service Requestor

- ▶ **FIND**s required services in the Service Registry
- ▶ **BIND**s to services from Service Provider



WSDL: Web Services Description Language

an XML Vocabulary

- ▶ similar in purpose to IDL, but in XML form

operational information about the service

- ▶ everything an IDE needs to know to help you integrate the service into your application

WSDL is the convergence of IBM's NASSL and Microsoft's SDL

- ▶ Submitted to W3C by 25 companies

We'll see the important role WSDL plays when we look at Web Services tooling later....

Prolog

Types

Messages

Port Types

Operations

Bindings

What HTTPS/SSL offers (at the protocol level)

- ✓ **identification:** who are you?
- ✓ **authentication:** how do I know your identity is true?
- ✗ **authorization:** are you allowed to perform this transaction?
- ✓ **integrity:** is the data you sent the same as the data I received?
- ✓ **confidentiality:** are we sure that nobody read the data you sent me?
- ✗ **auditing:** record of all transactions so we can look for security problems after the fact
- ✗ **non-repudiation:** both sender and receiver can provide legal proof to a third party that
 - the sender did send the transaction, and
 - the receiver received the identical transaction

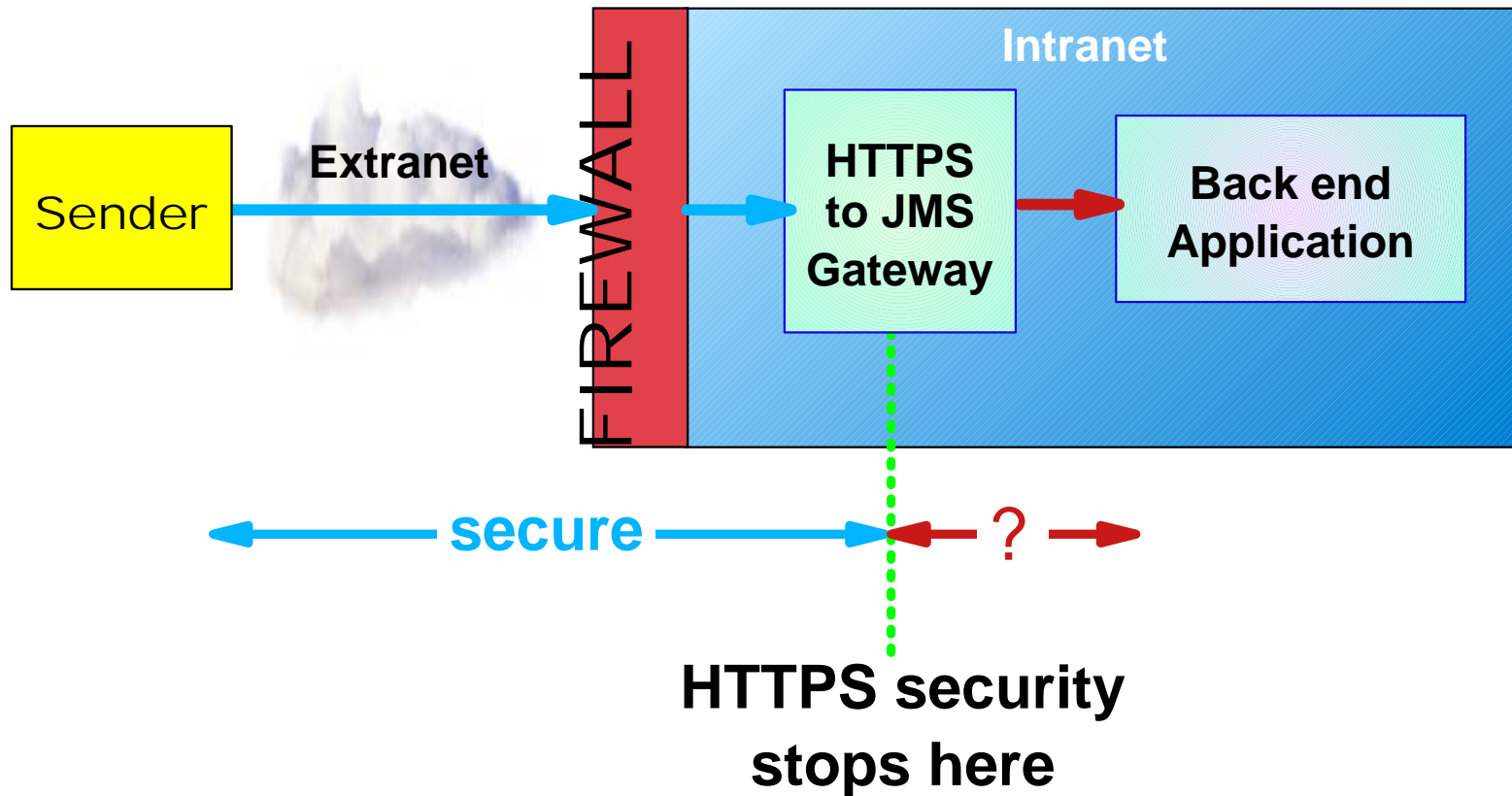
Why isn't HTTPS enough?

Limitation 1:

- ▶ no authorization, auditing, non-repudiation

Limitation 2: Protocol translation

- ▶ identification, authentication, confidentiality stop at HTTPS end point



Why isn't HTTPS enough?

Limitation 3: Signature and non-repudiation

- ▶ we want an integrity signature to persist...
all the way to a database used for audit trail
- ▶ prove message has not been modified
- ▶ HTTPS has no signature (that can be used for non-repudiation)

Limitation 4: Element-wise encryption

- ▶ decryption is necessary to route the message
 - HTTPS encrypts everything...
 - so you have to decrypt everything to route it
- ▶ we may need certain data (credit card #) to remain encrypted all the way to endpoint

Message-level Security

How message-level security helps

- ▶ W3C XML Signatures in the message carry through to the ultimate endpoint for processing
- ▶ W3C XML Encryption allows element-wise encryption
 - some parts exposed for routing purposes
 - other parts hidden up to final endpoint
- ▶ OASIS SAML proposes an architecture for authorization
 - message carries "receipt" of authorization
 - credentials carry through with the message
- ▶ OASIS XACML proposes an access control mechanism

XKMS: a Web service to manage public keys

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

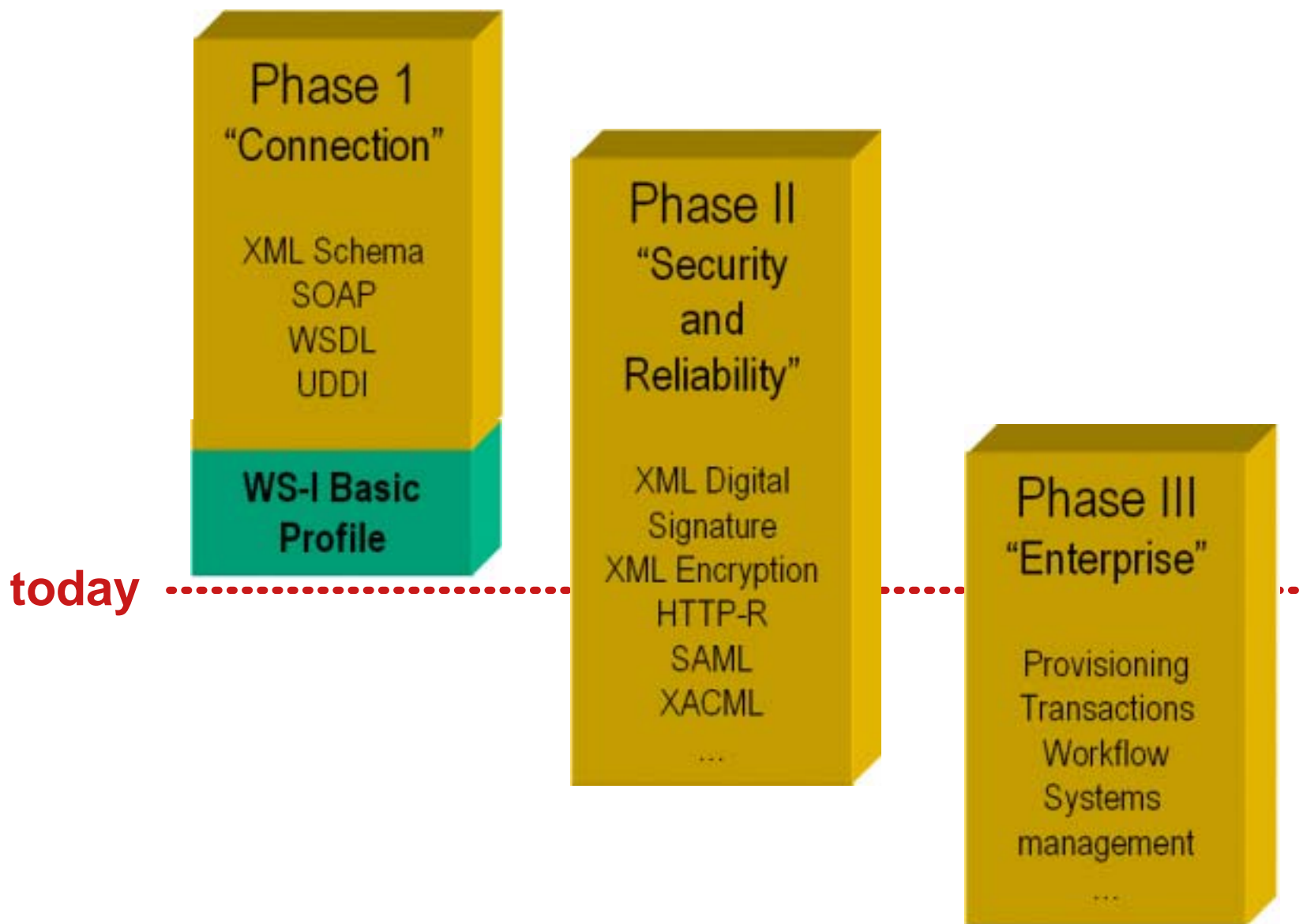
Focused on promoting Web service interoperability

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

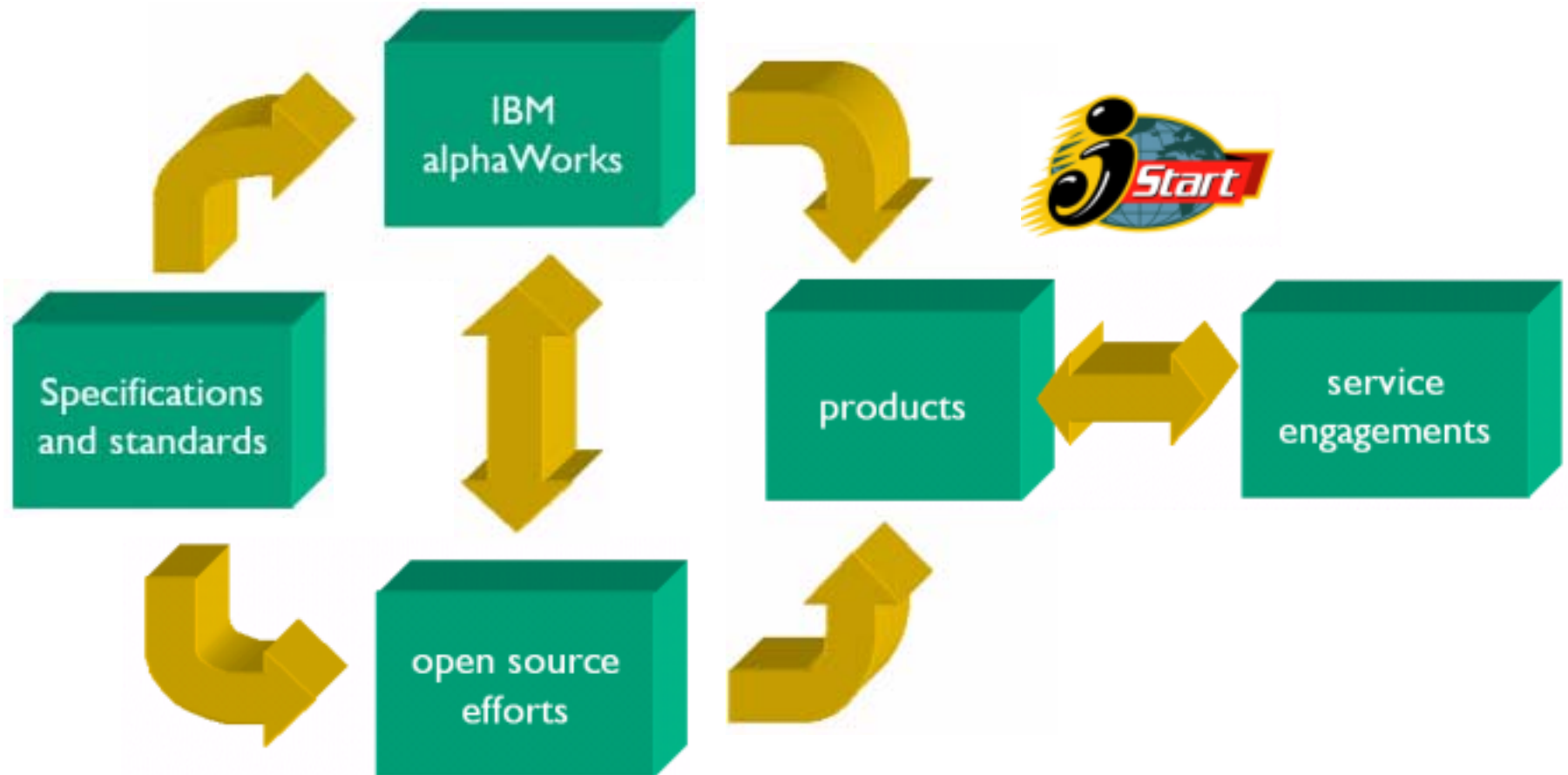
Promote customer adoption & deployment

- ▶ Integrate specifications from standards bodies
- ▶ Implementation guidance & tools for customers building and deploying Web services

Specifications and Standards



The IBM emerging technologies lifecycle



We have evolved this over the last 6 years with Java, XML, and Web services.

Cooperation and Competition

We cooperate with our competitors to create the standards that are essential to seamless connection of products created by different vendors, no matter

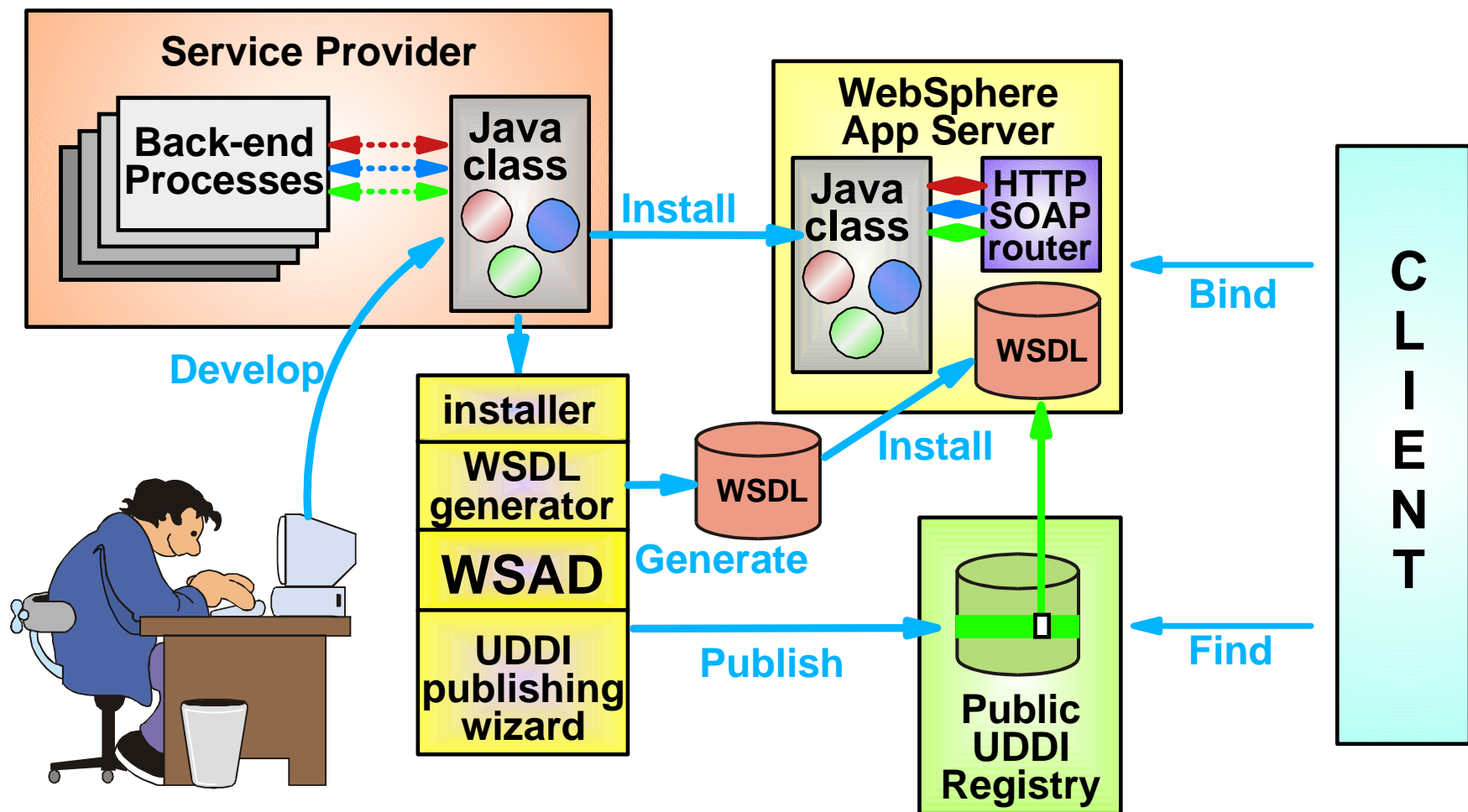
- ▶ how difficult the intra-industry politics become, and
- ▶ how skeptical some observers are of the attempt to cooperate

IBM will compete aggressively to produce and sell the best possible middleware across our entire product line (WebSphere, DB2, Lotus, Tivoli, ...) to build, invoke, and manage Web services.

"Cooperate on Standards...
Compete on Implementations"

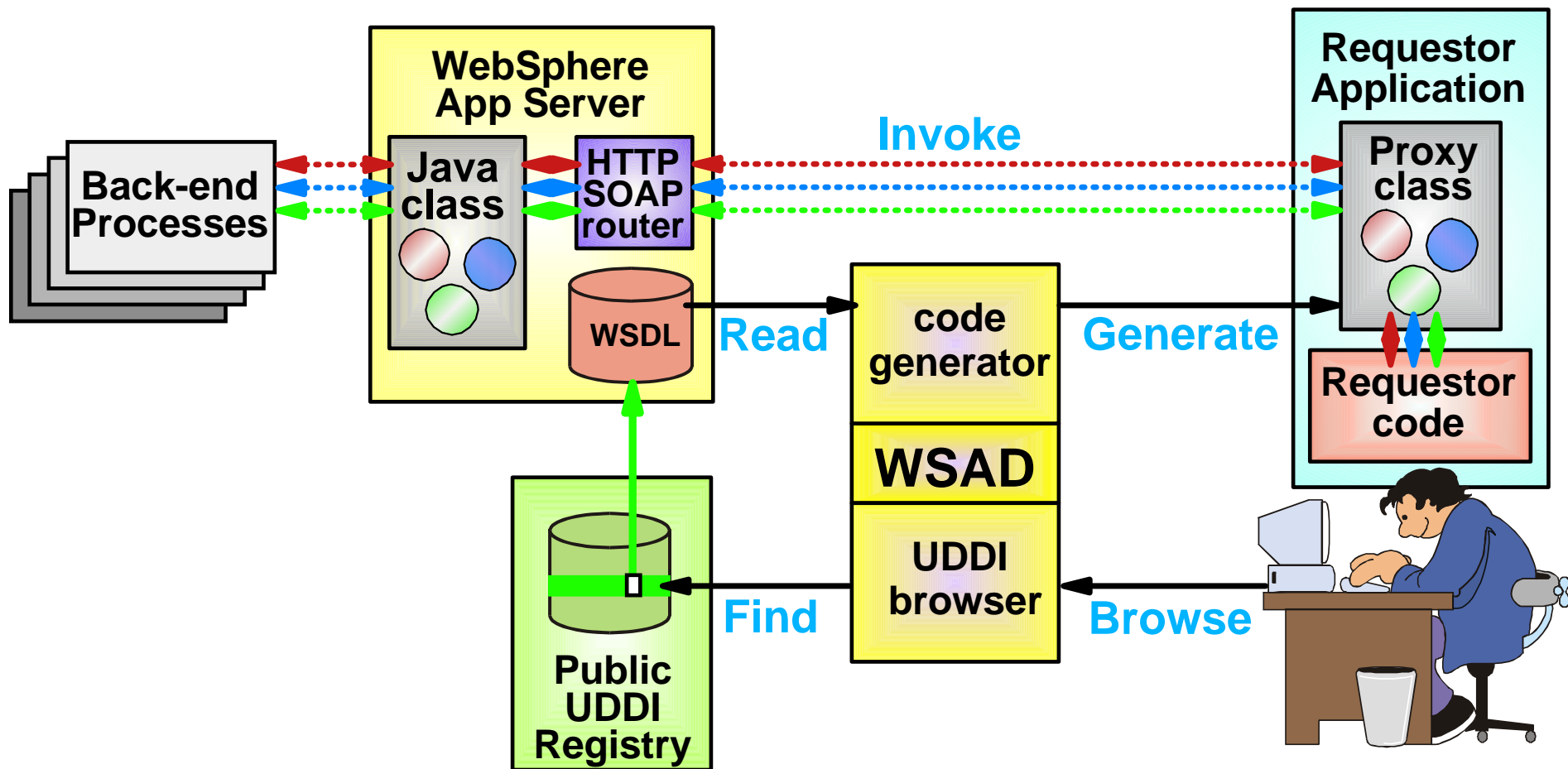
WebSphere Studio App Developer: speeding *deployment* of Web Services

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



IBM alphaWorks

<http://ibm.com/alphaWorks>

Hundreds of tools for Web Services, XML, Java

- ▶ early versions of features that may be in products
- ▶ some are solid production-code (XML4J, LotusXSL)
- ▶ some are experimental, prototypes
- ▶ free download and use

Some recent Web Services downloads:

- ▶ Web Services Toolkit 3.0 and demos
- ▶ Web Services Hosting Technology
- ▶ Web Services Process Management Toolkit
- ▶ Web Services Invocation Framework
- ▶ Web Services Gateway
- ▶ WSDL Toolkit

Web services zone



Web service invocation sans SOAP





Web Services Invocation Framework creates an interface that is independent of the transport mechanism used by a service. It allows the developer to [invoke Web services by using the Web Services Description Language directly](#), thus completely hiding the transport layer interactions. (Articles)

- [Using WSDL in a UDDI registry, Part 2](#): We continue this series with an introspective on the various programming scenarios of using WSDL in a UDDI registry environment. (Articles)
 - [Using WSDL in a UDDI registry, Part 1](#): This paper discusses the specifics of dealing with WSDL in UDDI registry environments to allow services to search for each other. It expands the guidelines given by UDDI.org on how WSDL can work with UDDI with more concrete detail. (Articles)
 - [SOAP security extensions: digital signature](#): Satoshi Hada explains how Digital Signatures, SSL, and SOAP can work together in a cohesive, complementary, and standards-based system. (Articles)
 - [Web services and short messaging](#): This is a case study on the development of a Web services-enabled implementation of the Short messaging service used in cellphones, 2-way pagers, and wireless PDAs. (Articles)
 - [Web services and XML technologies CD](#): This recently updated CD offers articles, tutorials and tools to keep you up to date with the latest XML and Web services developments from developerWorks and alphaWorks. (Articles)
- **dW theme:** [Code reuse](#): Be the master of your code.

Discussion forums

- [Web services technical](#): Get answers to questions on designing, implementing, and managing vendor-independent Web services. (Forums)

Columns

-  **Web services architect, Part 3** by Dan Gisolfi
The Web services architect examines the structural differences between [Web services and CORBA](#).
-  **The Web services insider, Part 9** by James Snell and Maryann Hondo
The Insider defines the questions we should be asking about [Web services security](#). **New!**
-  **The Web services (r)evolution, Part 4** by Graham Glass
In this [installment](#), Graham explains WSDL, how to describe the core properties of a Web service, and introduces tools that leverage WSDL to accelerate your development process.
-  **The Python Web services developer, Part 4** by Uche Ogbuji and Mike Olson
This conclusion to the series on [Web services software repository](#) explains how WSDL plays its part in describing the packages.

September 26, 2001

- [Tutorials](#)
- [Submit content](#)
- [Find industry jobs](#)
- [Ask the experts](#)
- [IBM developers store](#)

Newsletters

Subscribe to the weekly dW newsletter:

 Text HTML→ [More newsletters](#)

WebSphere

NEW
solutions for
dynamic e-business

WEB SERVICES

briefing days

IBM delivers!
Download a trial copy of
VisualAge For Java 4.0

News [→ More news](#)

- [Firms wait for UDDI proof](#) (ZDNet UK)
- [IBM serves up Web services technology](#) (EarthWeb.com)
- [Flamenco Networks offers plug-and-play Web services](#) (InfoWorld)

How our customers are using Web services today

We have several customers who are already realizing the cost-savings and flexibility offered by Web services

- ▶ visit ibm.com/software/jstart for some case studies

There are two basic categories:

- ▶ EAI (Web services inside the firewall)
- ▶ Improving existing business partner integration (B2B)

Public UDDI is seen as a future growth

- ▶ focus is on current business partnerships
- ▶ advertising on UDDI for new business comes later
- ▶ Private UDDI is hot now

Web Services: Inside the Enterprise

Enterprise Application Integration

- ▶ Easy to integrate applications from different vendors, different hardware, different programming languages
- ▶ End-to-end integration leverages your systems and people for better efficiency
- ▶ Especially helpful for companies that formed from mergers and acquisitions

"Gartner believes that more than 40 percent of enterprises' first experience with Web services will be an internal deployment of a Web services-enabled architecture.

"In these implementations, enterprises will begin to realize immediate benefits even before the development of well-behaved [public] Web services."

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

Web Services: improving business partner integration

Common XML data formats allow different companies to integrate quickly for e-business

- ▶ Solves more difficult BP integration problem
- ▶ New model: find business partners dynamically, and begin doing business immediately
- ▶ Business directories, marketplaces, auctions

"Web services will offer some business-to-business benefits early on as well...by transforming the process through which enterprises make connections with one another.

"Established trading partners will seek to drive down the costs of interconnection by reducing the energy devoted to maintaining contacts."

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

Web Services: Summary

Software evolution, Business revolution

- ▶ leverage existing software as highly-integratable objects
- ▶ no need to learn a new programming language!
- ▶ integrate systems internally, or with business partners
- ▶ new business opportunities abound

Open standards is a requirement

- ▶ Web Services build on existing standards
- ▶ IBM leads the industry in development of new standards

Get started now with IBM

- ▶ WebSphere 4.0 fully supports Web Services applications
- ▶ WebSphere Studio Application Developer beta available now
- ▶ SOAP4J, UDDI4J, Web Services Toolkit on ibm.com/alphaworks
- ▶ jStart Web Services team helps get your dev team up to speed quickly with a limited-scope project

IBM



Thank You!

