Loosely Coupled e-Business Solutions

406.306 Management Information Systems

Jonghun Park

jonghun@snu.ac.kr

Dept. of Industrial Engineering

Seoul National University

9/20/2007

A Quick Overview of Web Services

406.622 Industrial Information Technology

Jonghun Park

jonghun@snu.ac.kr

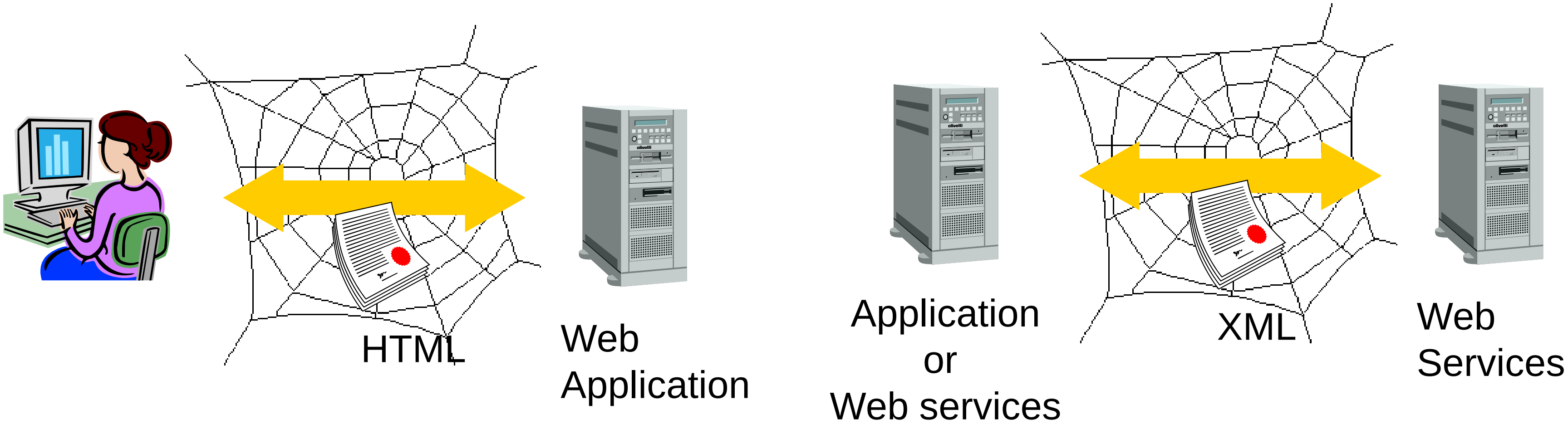
Dept. of Industrial Engineering

Seoul National University

9/20/2007

Web services: Towards “programmable” web

- **Web services**: A **software application** identified by a **URI**, whose interfaces and bindings are capable of being **defined**, **described**, and **discovered** as XML artifacts
- A Web service supports direct interactions with other software agents using **XML-based messages** exchanged via **Internet-based protocols** (W3C)
  - *Web based*
  - *Big thrust from major IT vendors*
  - *Interoperability supported by international standards*



HTML

Web Application

Application or Web services

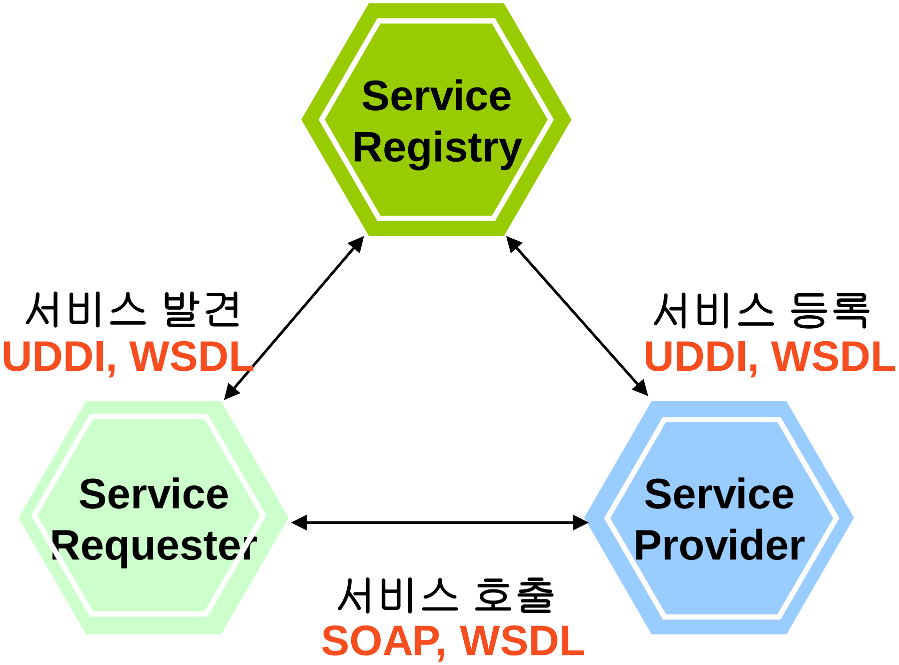
XML

Web Services

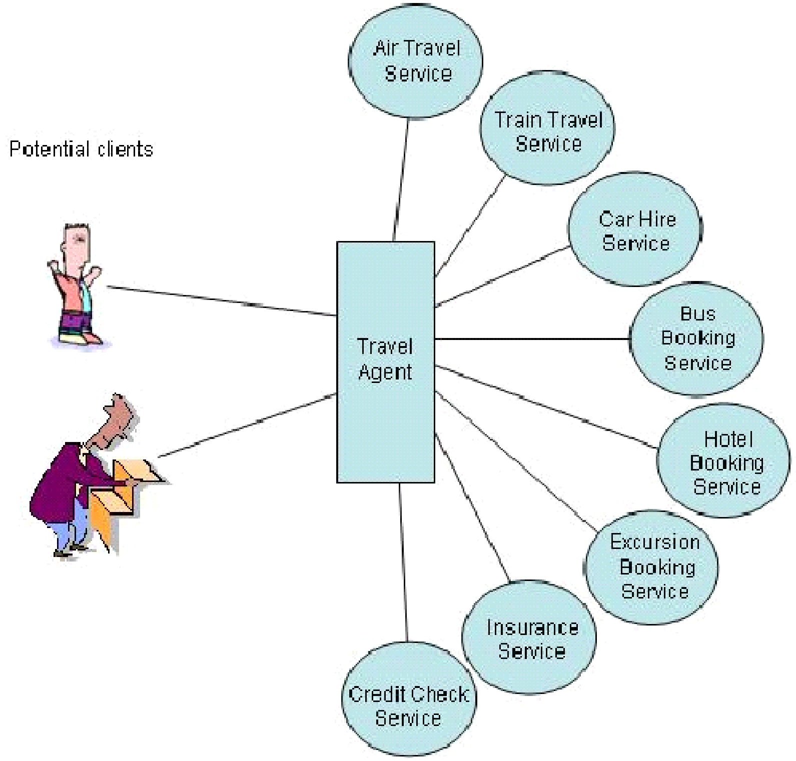
SOA, SOC, SODA, & SOBA, …

Everything is abstracted as a service!

Service: A procedure, method, or object with a stable, published interface that can be invoked by clients



Applications of web services



YAHOO! SEARCH

Google™

amazon.com.

ebaY®

xignite Powering On-Demand Financial Applications

TerraServer USA

Web Service List

WebserviceX.NET

SearchWebServices.com

BindingPoint

salcentral web services brokerage

X METHODS

GET THE FEELING TOYOTA

MSNBC HOME

Sabre Holdings™

JPMorgan

NASDAQ®

우리은행

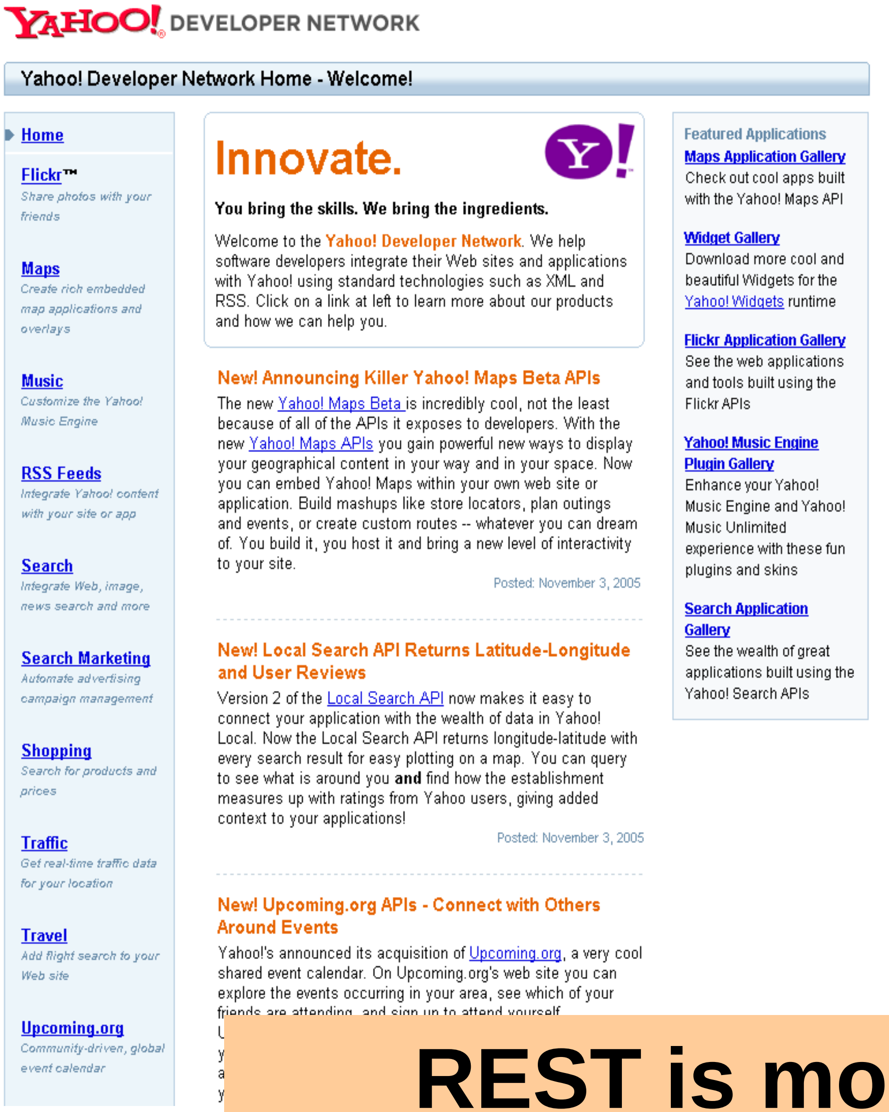
Daum

LG Electronics

SAMSUNG 삼성생명

SAMSUNG 삼성전자

대표적 웹 서비스 제공 업체



YAHOO! DEVELOPER NETWORK

Yahoo! Developer Network Home - Welcome!

- Home
- Flickr™ Share photos with your friends
- Maps Create rich embedded map applications and overlays
- Music Customize the Yahoo! Music Engine
- RSS Feeds Integrate Yahoo! content with your site or app
- Search Integrate Web, image, news search and more
- Search Marketing Automate advertising campaign management
- Shopping Search for products and prices
- Traffic Get real-time traffic data for your location
- Travel Add flight search to your Web site
- Upcoming.org Community-driven, global event calendar

Innovate.

You bring the skills. We bring the ingredients.

Welcome to the Yahoo! Developer Network. We help software developers integrate their Web sites and applications with Yahoo! using standard technologies such as XML and RSS. Click on a link at left to learn more about our products and how we can help you.

New! Announcing Killer Yahoo! Maps Beta APIs

The new Yahoo! Maps Beta is incredibly cool, not the least because of all of the APIs it exposes to developers. With the new Yahoo! Maps APIs you gain powerful new ways to display your geographical content in your way and in your space. Now you can embed Yahoo! Maps within your own web site or application. Build mashups like store locators, plan outings and events, or create custom routes -- whatever you can dream of. You build it, you host it and bring a new level of interactivity to your site.

Posted: November 3, 2005

New! Local Search API Returns Latitude-Longitude and User Reviews

Version 2 of the Local Search API now makes it easy to connect your application with the wealth of data in Yahoo! Local. Now the Local Search API returns longitude-latitude with every search result for easy plotting on a map. You can query to see what is around you **and** find how the establishment measures up with ratings from Yahoo users, giving added context to your applications!

Posted: November 3, 2005

New! Upcoming.org APIs - Connect with Others Around Events

Yahoo!'s announced its acquisition of Upcoming.org, a very cool shared event calendar. On Upcoming.org's web site you can explore the events occurring in your area, see which of your friends are attending, and sign up to attend yourself.

U
y
a
y

Featured Applications

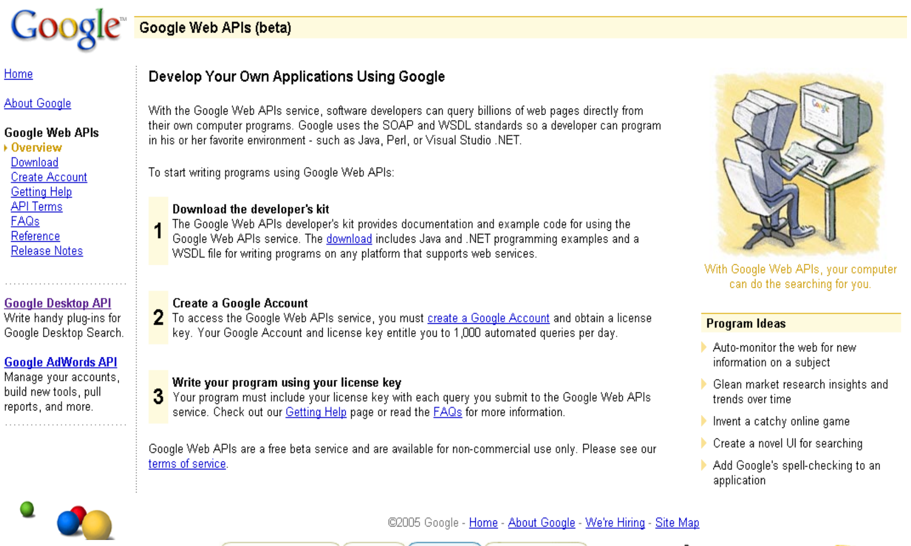
Maps Application Gallery
Check out cool apps built with the Yahoo! Maps API

Widget Gallery
Download more cool and beautiful Widgets for the Yahoo! Widgets runtime

Flickr Application Gallery
See the web applications and tools built using the Flickr APIs

Yahoo! Music Engine Plugin Gallery
Enhance your Yahoo! Music Engine and Yahoo! Music Unlimited experience with these fun plugins and skins

Search Application Gallery
See the wealth of great applications built using the Yahoo! Search APIs



Google Web APIs (beta)

- Home
- About Google

Google Web APIs
- Overview
- Download
- Create Account
- Getting Help
- API Terms
- FAQs
- Reference
- Release Notes

Google Desktop API
Write handy plug-ins for Google Desktop Search.

Google AdWords API
Manage your accounts, build new tools, pull reports, and more.

Develop Your Own Applications Using Google

With the Google Web APIs service, software developers can query billions of web pages directly from their own computer programs. Google uses the SOAP and WSDL standards so a developer can program in his or her favorite environment - such as Java, Perl, or Visual Studio .NET.

To start writing programs using Google Web APIs:

1 Download the developer's kit
The Google Web APIs developer's kit provides documentation and example code for using the Google Web APIs service. The download includes Java and .NET programming examples and a WSDL file for writing programs on any platform that supports web services.

2 Create a Google Account
To access the Google Web APIs service, you must create a Google Account and obtain a license key. Your Google Account and license key entitle you to 1,000 automated queries per day.

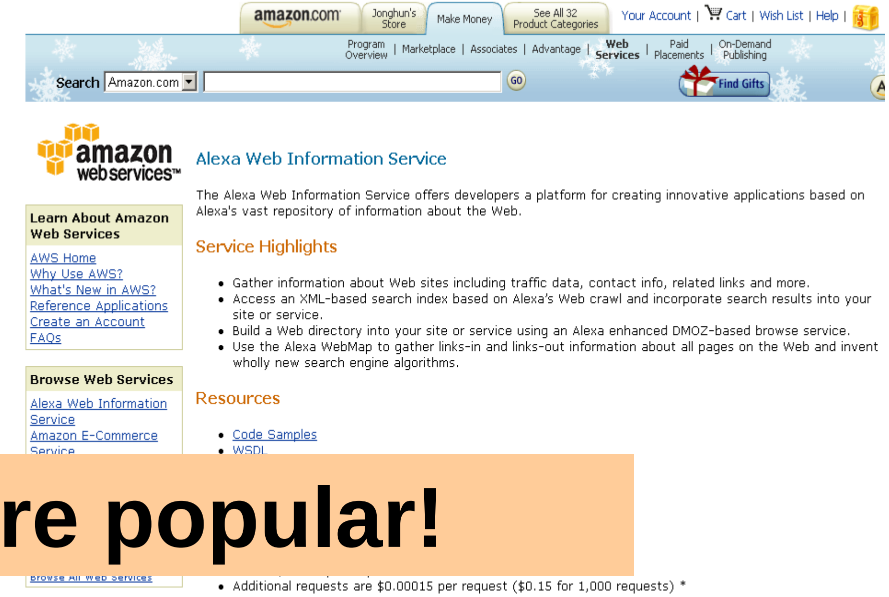
3 Write your program using your license key
Your program must include your license key with each query you submit to the Google Web APIs service. Check out our Getting Help page or read the FAQs for more information.

Google Web APIs are a free beta service and are available for non-commercial use only. Please see our terms of service.

With Google Web APIs, your computer can do the searching for you.

Program Ideas
- Auto-monitor the web for new information on a subject
- Glean market research insights and trends over time
- Invent a catchy online game
- Create a novel UI for searching
- Add Google's spell-checking to an application

©2005 Google - Home - About Google - We're Hiring - Site Map



amazon.com
Jonghun's Store
Make Money
See All 32 Product Categories
Your Account | Cart | Wish List | Help |
Program Overview | Marketplace | Associates | Advantage | Web Services | Paid Placements | On-Demand Publishing
Search Amazon.com
GO
Find Gifts

amazon webservices™

Learn About Amazon Web Services
- AWS Home
- Why Use AWS?
- What's New in AWS?
- Reference Applications
- Create an Account
- FAQs

Browse Web Services
- Alexa Web Information Service
- Amazon E-Commerce Service
- Browse All Web Services

Alexa Web Information Service

The Alexa Web Information Service offers developers a platform for creating innovative applications based on Alexa's vast repository of information about the Web.

Service Highlights

- Gather information about Web sites including traffic data, contact info, related links and more.
- Access an XML-based search index based on Alexa's Web crawl and incorporate search results into your site or service.
- Build a Web directory into your site or service using an Alexa enhanced DMOZ-based browse service.
- Use the Alexa WebMap to gather links-in and links-out information about all pages on the Web and invent wholly new search engine algorithms.

Resources

- Code Samples
- WSDL
- Additional requests are $0.00015 per request ($0.15 for 1,000 requests) *

REST is more popular!

Some “real” web services

- Google: http://www.google.com/apis/
- Amazon:
  - http://www.amazon.com/gp/browse.html/104-1656612-4225519?%5Fencoding=UTF8&node=3435361
  - http://pages.alexa.com/prod_serv/WebInfoService.html
- eBay: http://developer.ebay.com/DevProgram/preview.asp
- OpenDBLP: http://opendblp.psu.edu/
- Swoogle: http://swoogle.umbc.edu/modules.php?name=News&file=article&sid=13&mode=&order=0&thold=0
- Xignite: http://www.xignite.com/
- TerraServer: http://terraservice.net/webservices.aspx

Web services directories

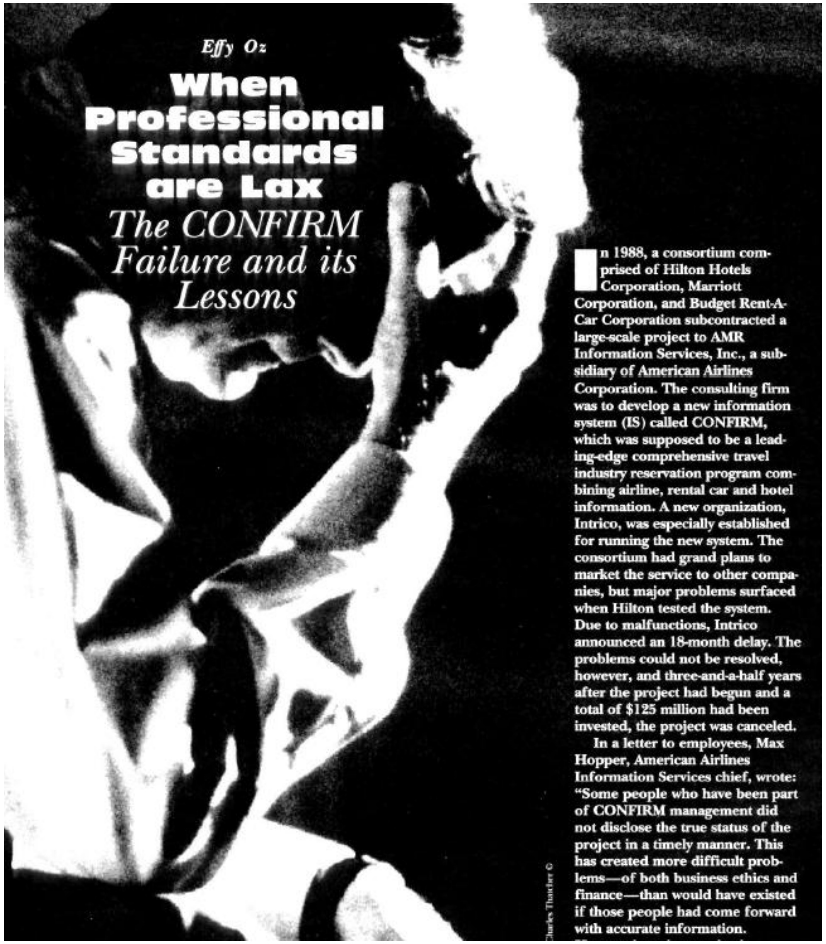
- Web Service List: http://www.webservicelist.com/
- Remote Methods: http://www.remotemethods.com/
- WebserviceX.NET: http://www.webservicex.net/WS/default.aspx
- SearchWebServices.com: http://searchwebservices.techtarget.com/bestWebLinks/0,289521,sid26_tax292848,00.html
- Binding Point: http://www.bindingpoint.com/
- X Methods: http://www.xmethods.com/
- SalCentral: http://www.salcentral.com/Search.aspx
- Google directory: http://directory.google.com/Top/Computers/Programming/Internet/Web_Services/

X METHODS

salcentral
web services brokerage

Lessons learned from past: CONFIRM

- An ambitious software development initiative that sought to **integrate airline reservations, car rentals, and hotel reservations**, along with their respective decision support mechanisms into a single system
- The project development firm: AMRIS (AMR Information Services, Inc., a subsidiary of AA)
- The project lasted **3.5 years**, spending **$125 million** and producing an unusable system (CACM, 37(10), 1994)



Effy Oz

When Professional Standards are Lax

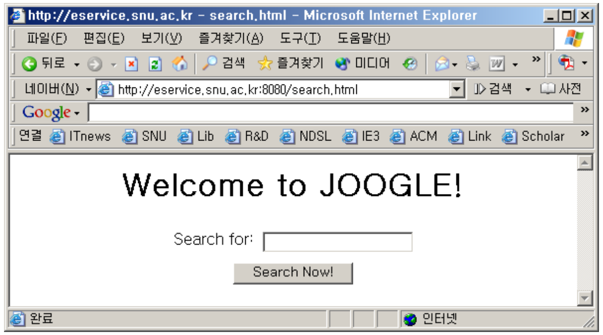
The CONFIRM Failure and its Lessons

In 1988, a consortium comprised of Hilton Hotels Corporation, Marriott Corporation, and Budget Rent-A-Car Corporation subcontracted a large-scale project to AMR Information Services, Inc., a subsidiary of American Airlines Corporation. The consulting firm was to develop a new information system (IS) called CONFIRM, which was supposed to be a leading-edge comprehensive travel industry reservation program combining airline, rental car and hotel information. A new organization, Intrico, was especially established for running the new system. The consortium had grand plans to market the service to other companies, but major problems surfaced when Hilton tested the system. Due to malfunctions, Intrico announced an 18-month delay. The problems could not be resolved, however, and three-and-a-half years after the project had begun and a total of $125 million had been invested, the project was canceled.

In a letter to employees, Max Hopper, American Airlines Information Services chief, wrote: "Some people who have been part of CONFIRM management did not disclose the true status of the project in a timely manner. This has created more difficult problems—of both business ethics and finance—than would have existed if those people had come forward with accurate information.

Charles Thatcher ©

In search of killer applications of web services… ☺

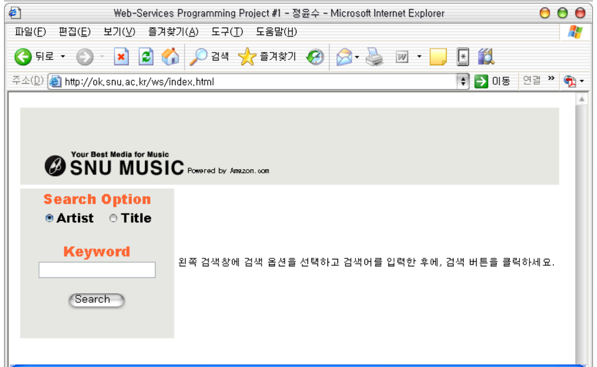

http://eservice.snu.ac.kr - search.html - Microsoft Internet Explorer

http://eservice.snu.ac.kr:8080/search.html

Welcome to JOOGLE!

Search for:

Search Now!
파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)
ITnews SNU Lib R&D NDSL IE3 ACM Link Scholar
완료
인터넷


Web-Services Programming Project #1 - 정윤수 - Microsoft Internet Explorer

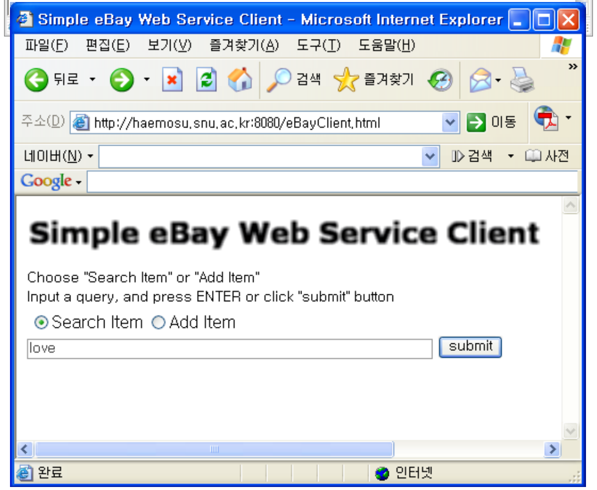
http://ok.snu.ac.kr/ws/index.html

Your Best Media for Music
SNU MUSIC Powered by Amazon.com

Search Option
● Artist ○ Title

Keyword

왼쪽 검색창에 검색 옵션을 선택하고 검색어를 입력한 후에, 검색 버튼을 클릭하세요.


Simple eBay Web Service Client - Microsoft Internet Explorer

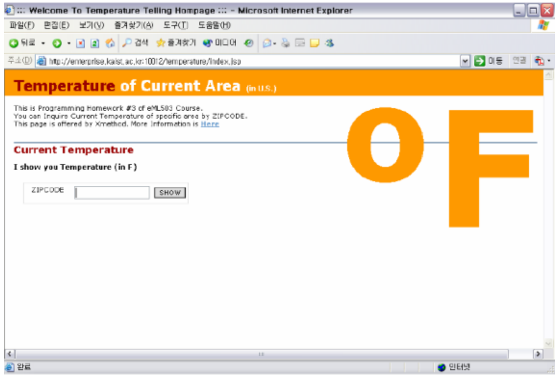
http://haemosu.snu.ac.kr:8080/eBayClient.html

Simple eBay Web Service Client

Choose "Search Item" or "Add Item"
Input a query, and press ENTER or click "submit" button

● Search Item ○ Add Item

love


::: Welcome To Temperature Telling Hompage ::: - Microsoft Internet Explorer

http://enterprise.kaist.ac.kr:10012/temperature/index.jsp

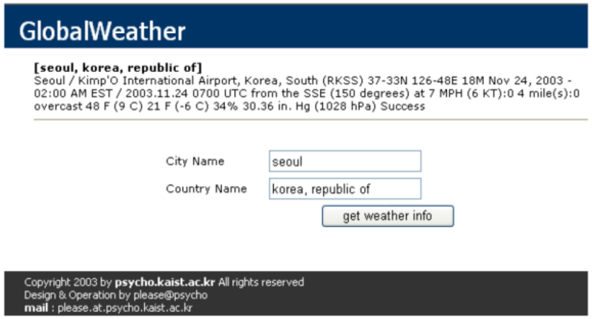
Temperature of Current Area (in U.S.)

This is Programming Homework #3 of eML503 Course.
You can Inquire Current Temperature of specific area by ZIPCODE.
This page is offered by Xmethod. More Information is Here

Current Temperature

I show you Temperature (in F)

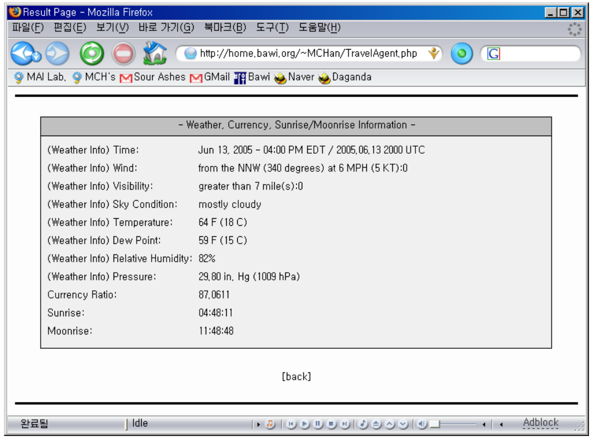
ZIPCODE


GlobalWeather

[seoul, korea, republic of]
Seoul / Kimp'O International Airport, Korea, South (RKSS) 37-33N 126-48E 18M Nov 24, 2003 - 02:00 AM EST / 2003.11.24 0700 UTC from the SSE (150 degrees) at 7 MPH (6 KT):0 4 mile(s):0 overcast 48 F (9 C) 21 F (-6 C) 34% 30.36 in. Hg (1028 hPa) Success

City Name: seoul
Country Name: korea, republic of

Copyright 2003 by **psycho.kaist.ac.kr** All rights reserved
Design & Operation by please@psycho
mail : please.at.psycho.kaist.ac.kr

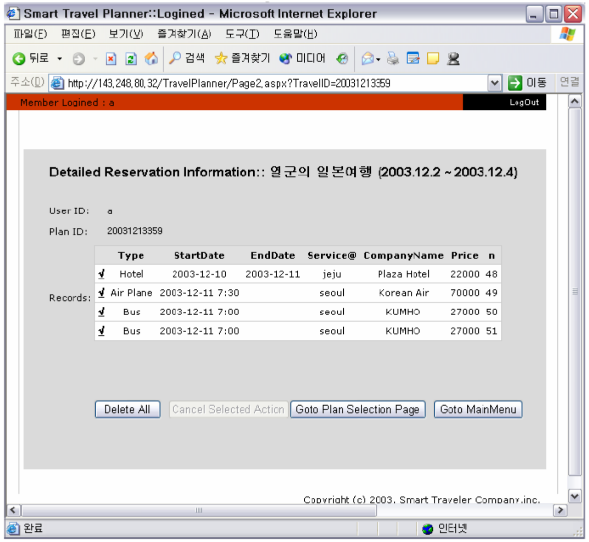

Result Page - Mozilla Firefox

http://home.bawi.org/~MCHan/TravelAgent.php

- Weather, Currency, Sunrise/Moonrise Information -

(Weather Info) Time:	Jun 13, 2005 - 04:00 PM EDT / 2005.06.13 2000 UTC
(Weather Info) Wind:	from the NNW (340 degrees) at 6 MPH (5 KT):0
(Weather Info) Visibility:	greater than 7 mile(s):0
(Weather Info) Sky Condition:	mostly cloudy
(Weather Info) Temperature:	64 F (18 C)
(Weather Info) Dew Point:	59 F (15 C)
(Weather Info) Relative Humidity:	82%
(Weather Info) Pressure:	29.80 in. Hg (1009 hPa)
Currency Ratio:	87.0611
Sunrise:	04:48:11
Moonrise:	11:48:48

[back]


Smart Travel Planner::Logined - Microsoft Internet Explorer

http://143.248.80.32/TravelPlanner/Page2.aspx?TravelID=20031213359

Member Logined : a

Detailed Reservation Information:: 열군의 일본여행 (2003.12.2 ~ 2003.12.4)

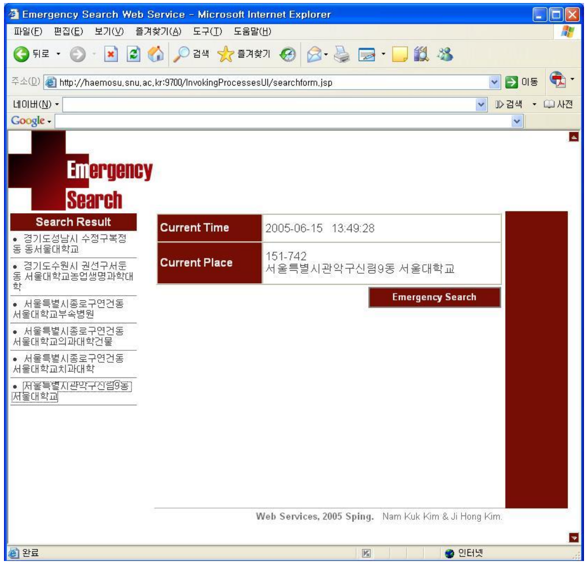
User ID: a

Plan ID: 20031213359

Records:

	Type	StartDate	EndDate	Service@	CompanyName	Price	n
✓	Hotel	2003-12-10	2003-12-11	jeju	Plaza Hotel	22000	48
✓	Air Plane	2003-12-11 7:30		seoul	Korean Air	70000	49
✓	Bus	2003-12-11 7:00		seoul	KUMHO	27000	50
✓	Bus	2003-12-11 7:00		seoul	KUMHO	27000	51

Copyright (c) 2003. Smart Traveler Company.inc.


Emergency Search Web Service - Microsoft Internet Explorer

http://haemosu.snu.ac.kr:9700/InvokingProcessesUI/searchform.jsp

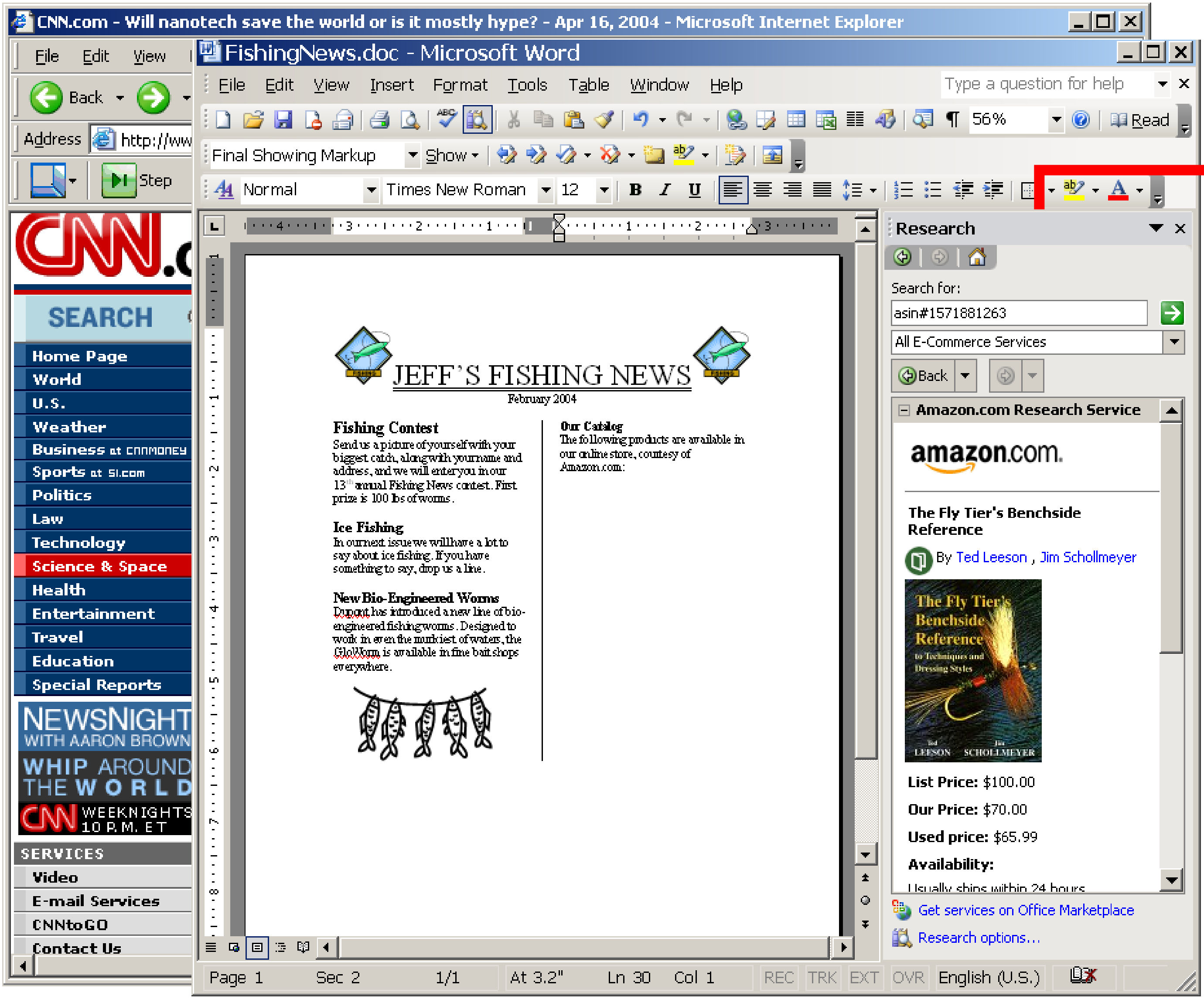
Emergency Search

Search Result
- 경기도성남시 수정구복정동 동서울대학교
- 경기도수원시 권선구서둔동 서울대학교농업생명과학대학
- 서울특별시종로구연건동 서울대학교부속병원
- 서울특별시종로구연건동 서울대학교의과대학건물
- 서울특별시종로구연건동 서울대학교치과대학
- 서울특별시관악구신림9동 서울대학교

Current Time	2005-06-15 13:49:28
Current Place	151-742 서울특별시관악구신림9동 서울대학교

Web Services, 2005 Sping. Nam Kuk Kim & Ji Hong Kim.

Amazon Web Services
(출처: ETRI PEC, 2005)



JEFF'S FISHING NEWS

February 2004

Fishing Contest

Send us a picture of yourself with your biggest catch, along with your name and address, and we will enter you in our 13th annual Fishing News contest. First prize is 100 lbs of worms.

Ice Fishing

In our next issue we will have a lot to say about ice fishing. If you have something to say, drop us a line.

New Bio-Engineered Worms

Dupont has introduced a new line of bio-engineered fishing worms. Designed to work in even the murkiest of waters, the GloWorm is available in fine bait shops everywhere.

Our Catalog

The following products are available in our online store, courtesy of Amazon.com:

Search for: asin#1571881263

All E-Commerce Services

Back

Amazon.com Research Service

amazon.com

The Fly Tier's Benchside Reference

By Ted Leeson , Jim Schollmeyer

List Price: $100.00

Our Price: $70.00

Used price: $65.99

Availability:

Usually ships within 24 hours

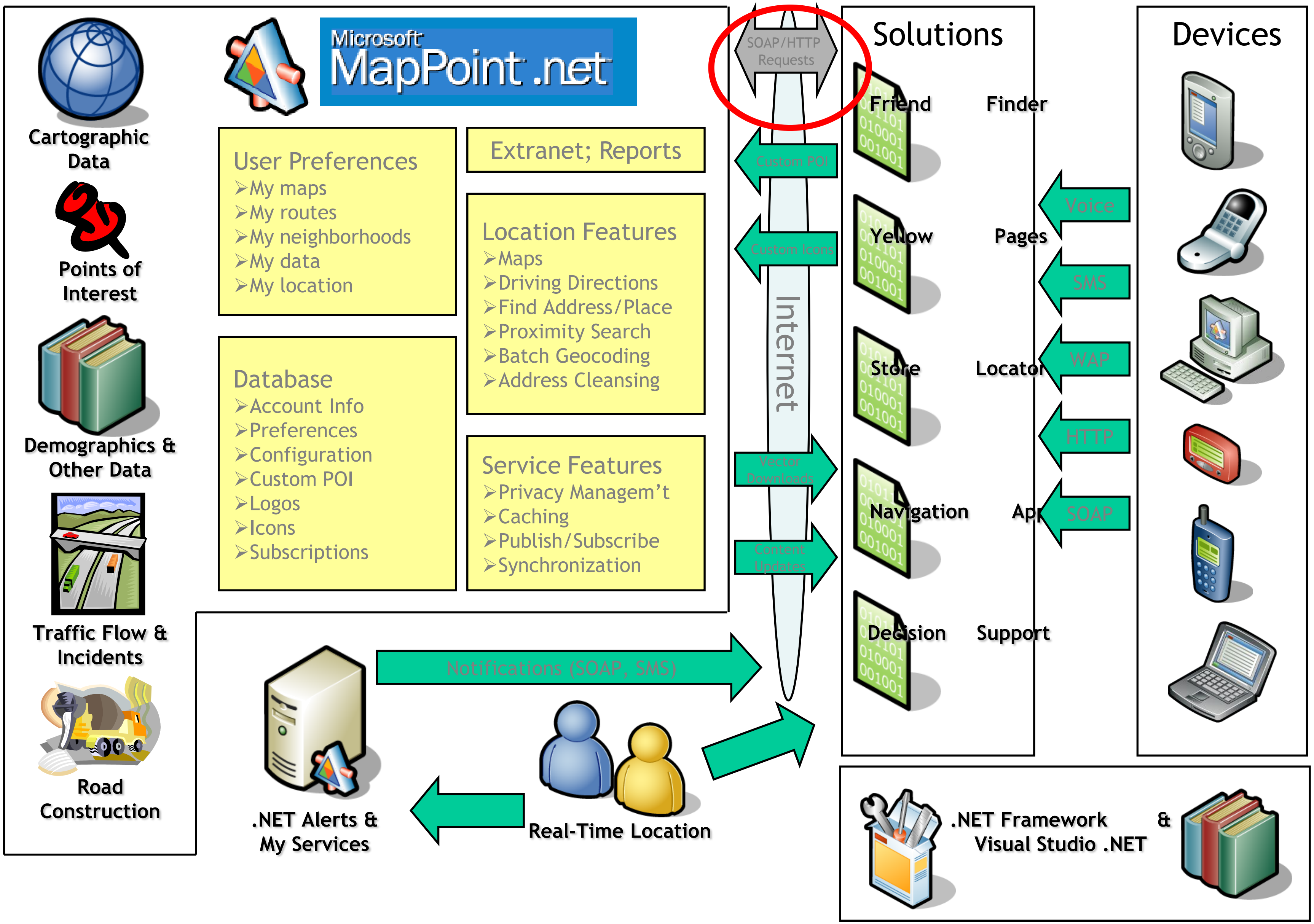
Get services on Office Marketplace

Research options...

MS Word
Can support
AWS Features

MapPoint

The Power of Location
MapPoint mapping and location intelligence solutions have the latest mapping technology built-in.



Cartographic Data

Points of Interest

Demographics & Other Data

Traffic Flow & Incidents

Road Construction

Microsoft MapPoint .net

User Preferences
- My maps
- My routes
- My neighborhoods
- My data
- My location

Extranet; Reports

Location Features
- Maps
- Driving Directions
- Find Address/Place
- Proximity Search
- Batch Geocoding
- Address Cleansing

Database
- Account Info
- Preferences
- Configuration
- Custom POI
- Logos
- Icons
- Subscriptions

Service Features
- Privacy Managem't
- Caching
- Publish/Subscribe
- Synchronization

SOAP/HTTP Requests

Custom POI

Custom Icons

Internet

Vector Downloads

Content Updates

Notifications (SOAP, SMS)

.NET Alerts & My Services

Real-Time Location

Solutions

Friend Finder

Yellow Pages

Store Locator

Navigation App

Decision Support

Voice

SMS

WAP

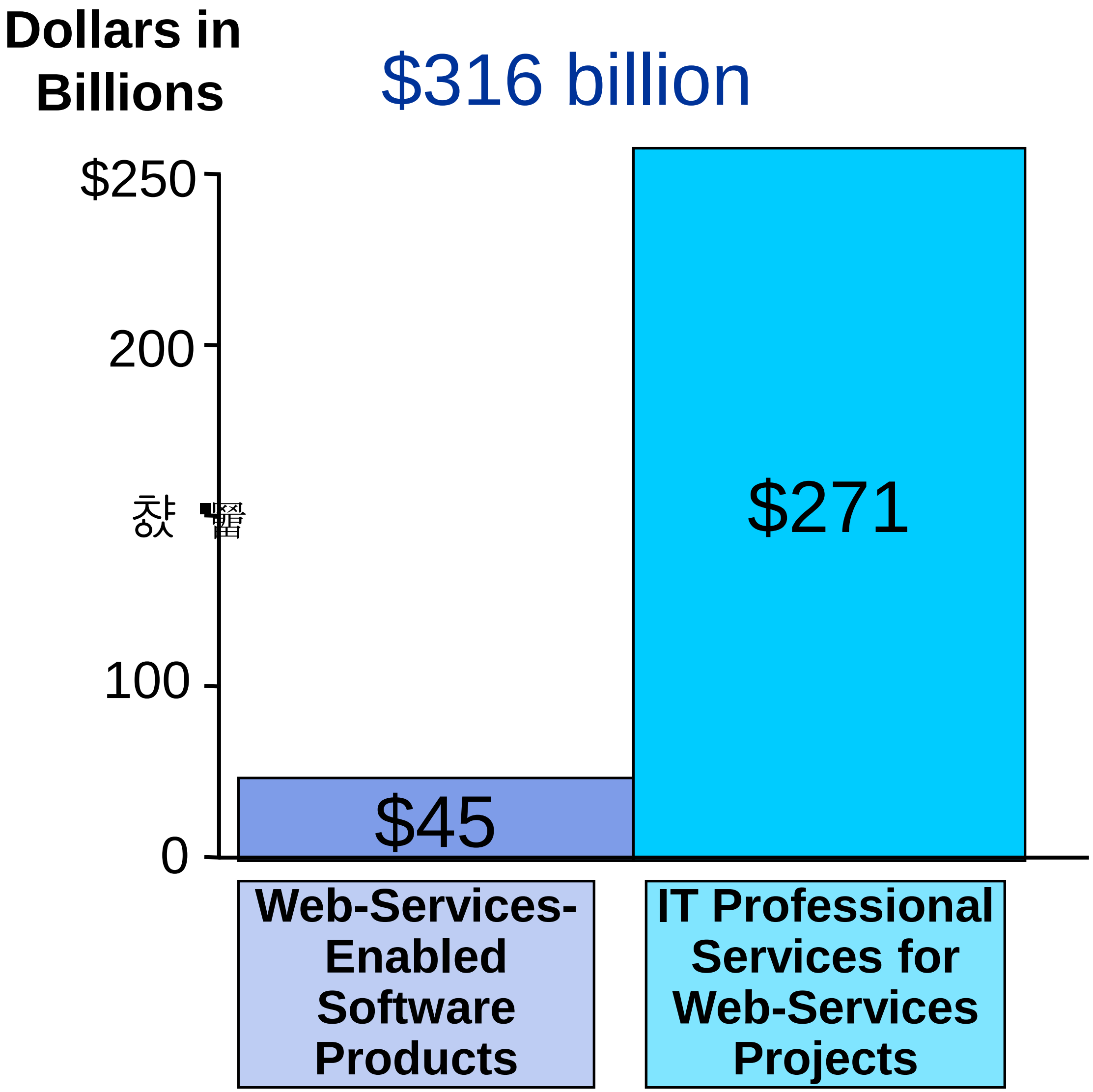
HTTP

SOAP

Devices

.NET Framework & Visual Studio .NET

2007 년도의 웹 서비스 시장 (Gartner, 2004)



- Revenue for Web-services-enabled software and professional services will grow from $61 billion (2003) to $316 billion (0.7 probability).
- 41 percent of enterprise software purchased in 2007 will be Web-services-enabled (0.8 probability).
- Services CAGR will be more than 61 percent in the next four years (0.8 probability).

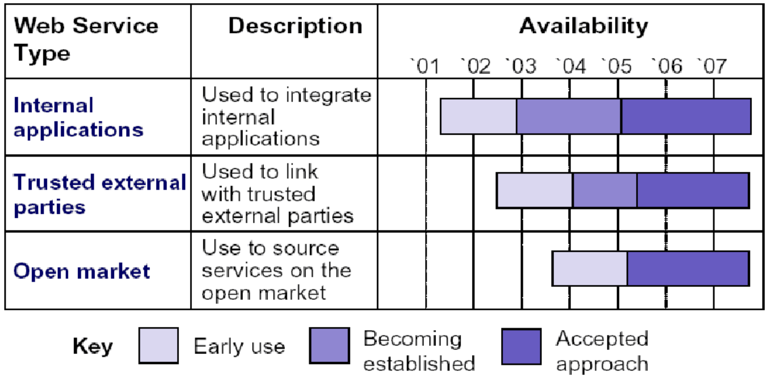
SOA에 대한 전망

- By 2007, SOA will be the **mainstream software engineering practice**, ending the 40-year domination of monolithic architecture (0.7 probability; Gartner, 2003)
- No later than year-end 2006, 90 % of application development staffs in the Global 2000 will be developing secure, marketable services for external use (0.8 probability; Gartner, 2003)
- Web services vendors with greater than $10 million annual revenue will add non-WS products or collapse by year-end 2006 (0.7 probability; Gartner, 2003)

Gartner predicts that **75% of companies with more than $100 million in annual revenue** will use Web service by the middle of 2003, and that the technology will reach mainstream users by 2004

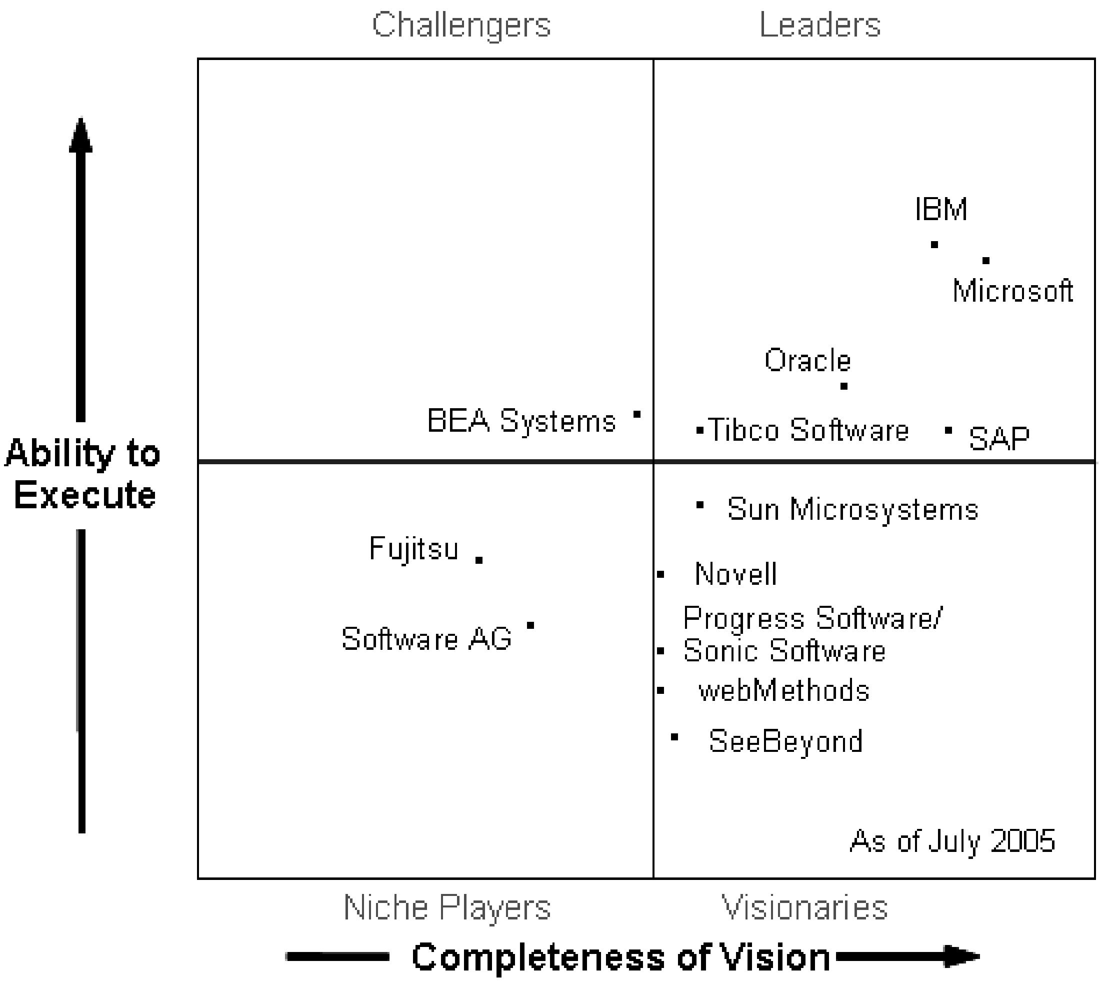
Web services market will be **$21 billion by 2007** and will peak at $27 billion in 2010 (IDC, 2003)

80% of US enterprises will have some type of Web services project under way by 2008 (IDC, 2003)



Web Service Type	Description
Internal applications	Used to integrate internal applications
Trusted external parties	Used to link with trusted external parties
Open market	Use to source services on the open market

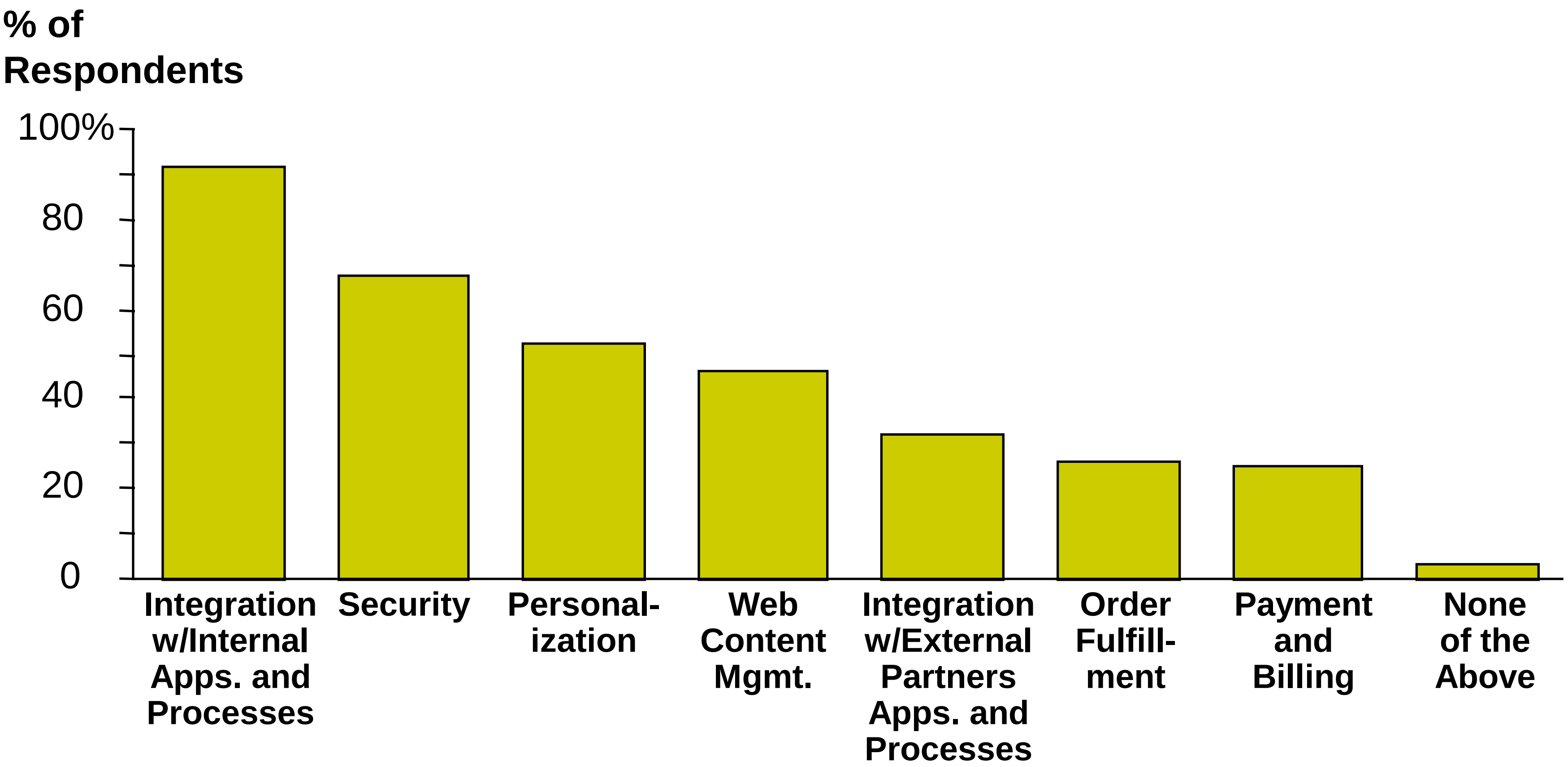
Web services platform vendors
(Gartner, 2005)



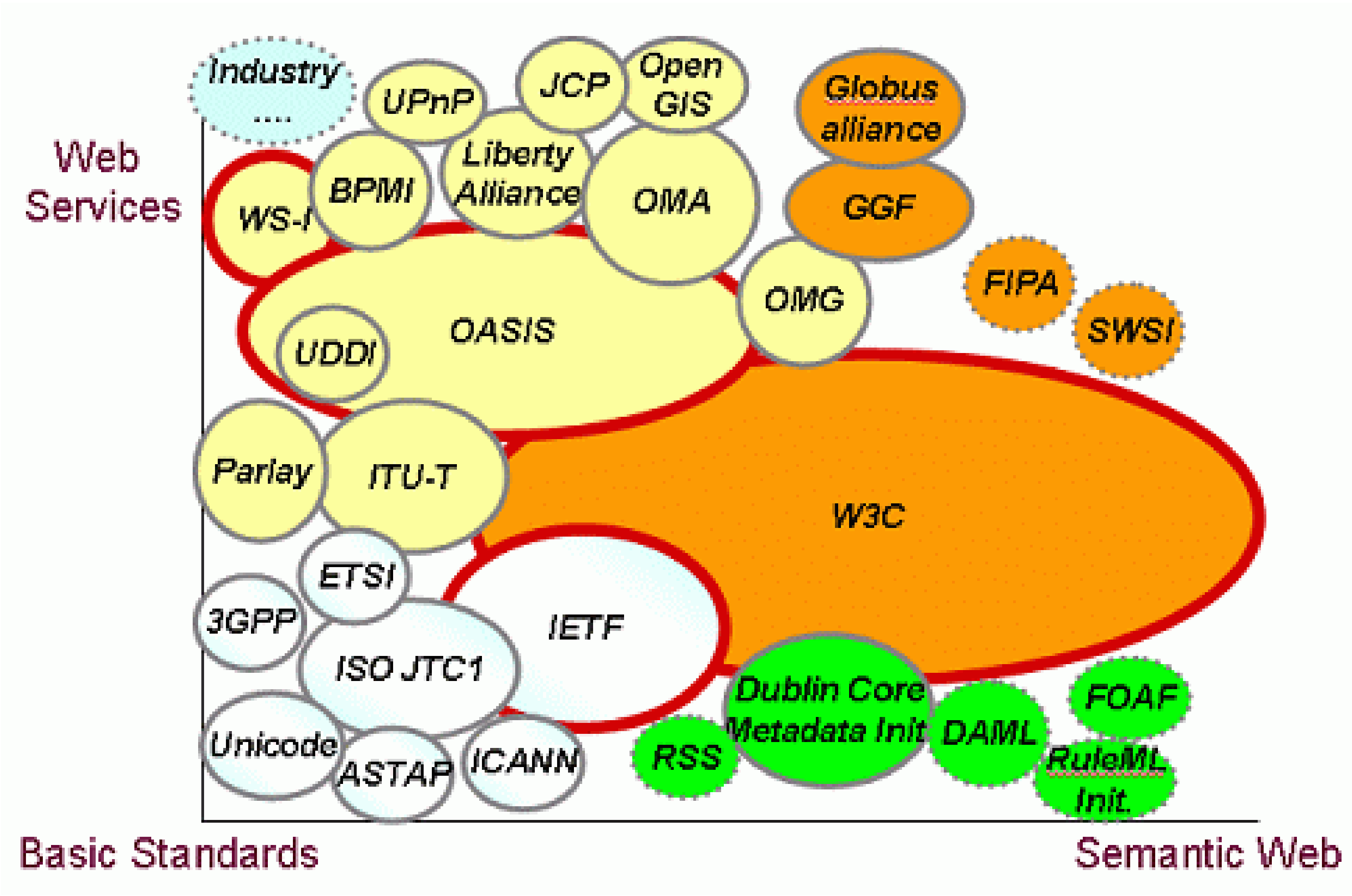
웹 서비스의 적용 현황

(Gartner, 2004)

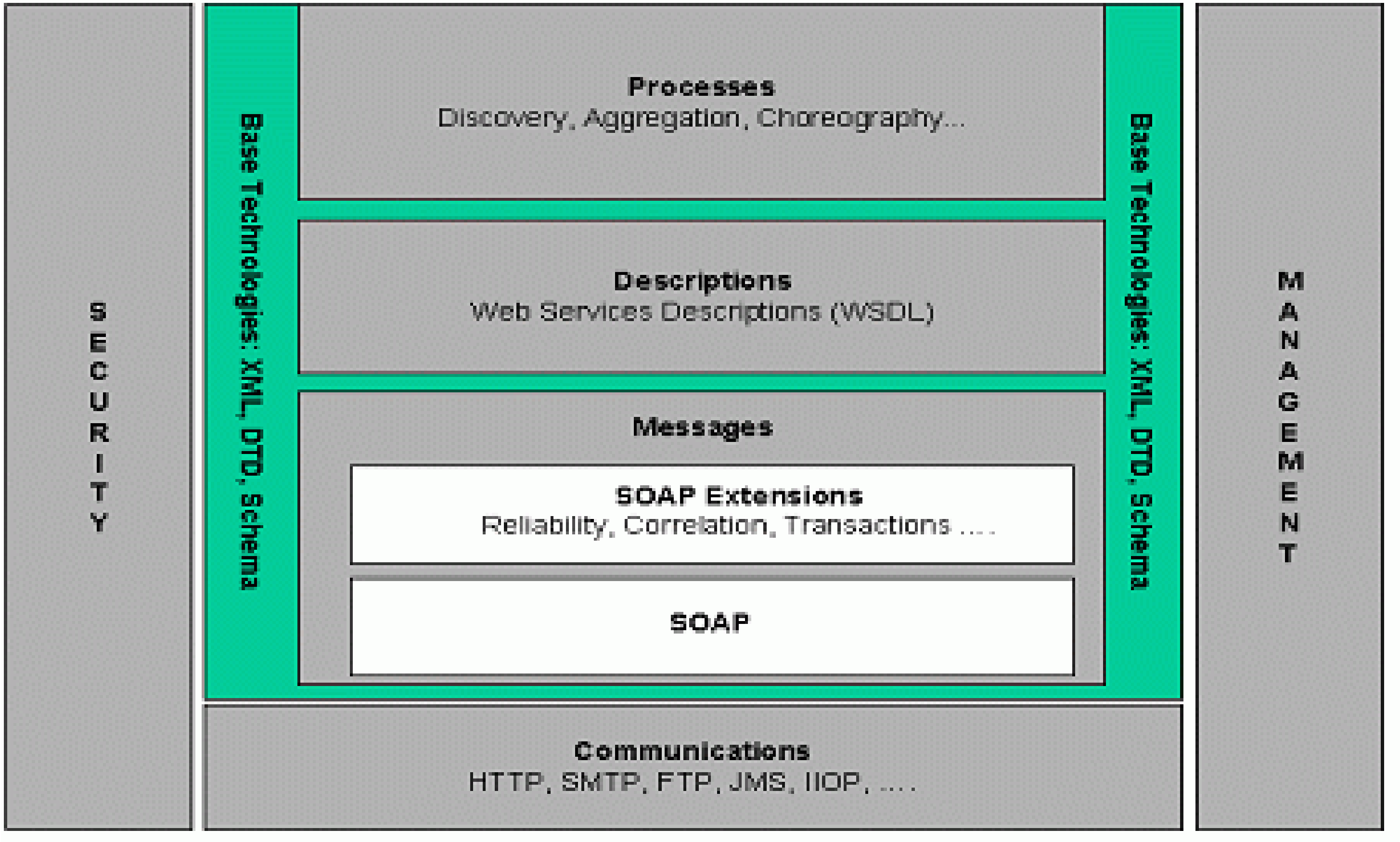
What Functions or Activities Are Common to Many Web Services Projects? Which Apply to This Project?



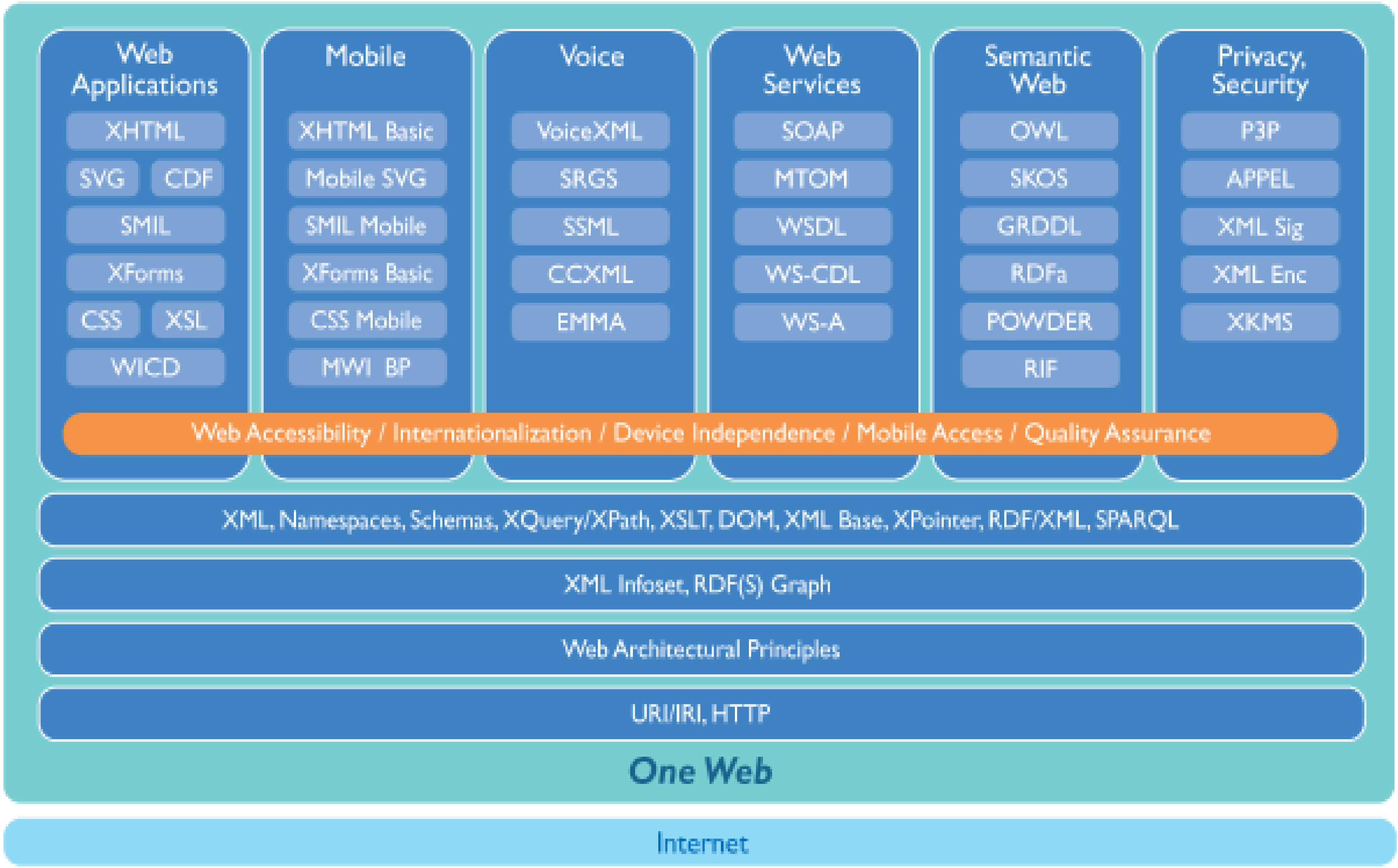
웹 기술 관련 표준화 기구


Industry
....
UPnP
JCP
Open
GIS
Globus
alliance
Web
Services
WS-I
BPMI
Liberty
Alliance
OMA
GGF
OMG
FIPA
SWSI
OASIS
UDDI
Parlay
ITU-T
W3C
ETSI
3GPP
IETF
ISO JTC1
Unicode
ASTAP
ICANN
RSS
Dublin Core
Metadata Init
DAML
FOAF
RuleML
Init.
Basic Standards
Semantic Web

Web services standards stack from W3C


SECURITY
Base Technologies: XML, DTD, Schema
Processes
Discovery, Aggregation, Choreography...
Descriptions
Web Services Descriptions (WSDL)
Messages
SOAP Extensions
Reliability, Correlation, Transactions ....
SOAP
Base Technologies: XML, DTD, Schema
MANAGEMENT
Communications
HTTP, SMTP, FTP, JMS, IIOP, ...

Another stack from W3C



Web Applications
XHTML
SVG
CDF
SMIL
XForms
CSS
XSL
WICD

Mobile
XHTML Basic
Mobile SVG
SMIL Mobile
XForms Basic
CSS Mobile
MWI BP

Voice
VoiceXML
SRGS
SSML
CCXML
EMMA

Web Services
SOAP
MTOM
WSDL
WS-CDL
WS-A

Semantic Web
OWL
SKOS
GRDDL
RDFa
POWDER
RIF

Privacy, Security
P3P
APPEL
XML Sig
XML Enc
XKMS

Web Accessibility / Internationalization / Device Independence / Mobile Access / Quality Assurance

XML, Namespaces, Schemas, XQuery/XPath, XSLT, DOM, XML Base, XPointer, RDF/XML, SPARQL

XML Infoset, RDF(S) Graph

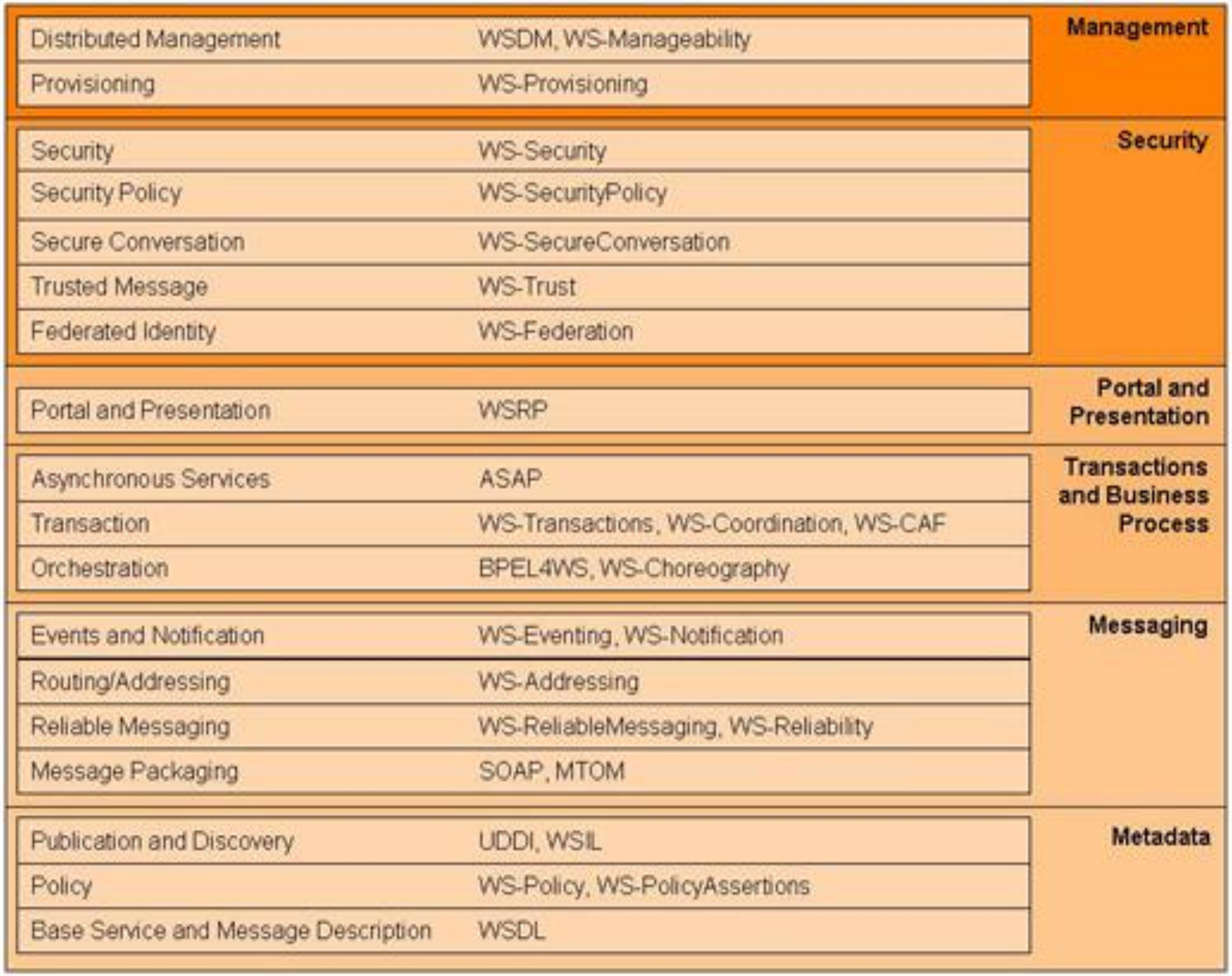
Web Architectural Principles

URI/IRI, HTTP

One Web

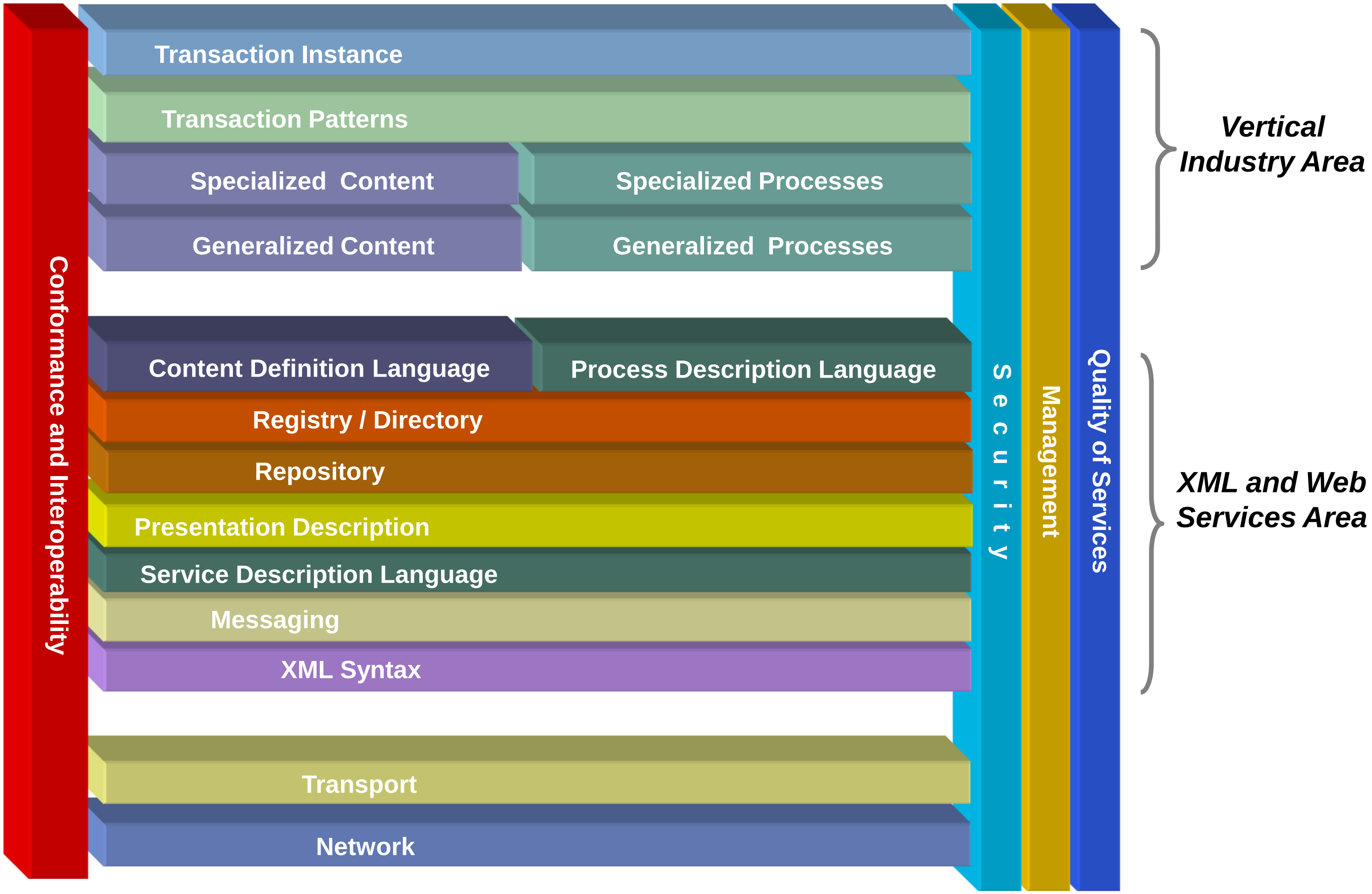
Internet

CBDi



Distributed Management	WSDM, WS-Manageability	Management
Provisioning	WS-Provisioning	
Security	WS-Security	Security
Security Policy	WS-SecurityPolicy	
Secure Conversation	WS-SecureConversation	
Trusted Message	WS-Trust	
Federated Identity	WS-Federation	
Portal and Presentation	WSRP	Portal and Presentation
Asynchronous Services	ASAP	Transactions and Business Process
Transaction	WS-Transactions, WS-Coordination, WS-CAF	
Orchestration	BPEL4WS, WS-Choreography	
Events and Notification	WS-Eventing, WS-Notification	Messaging
Routing/Addressing	WS-Addressing	
Reliable Messaging	WS-ReliableMessaging, WS-Reliability	
Message Packaging	SOAP, MTOM	
Publication and Discovery	UDDI, WSIL	Metadata
Policy	WS-Policy, WS-PolicyAssertions	
Base Service and Message Description	WSDL	

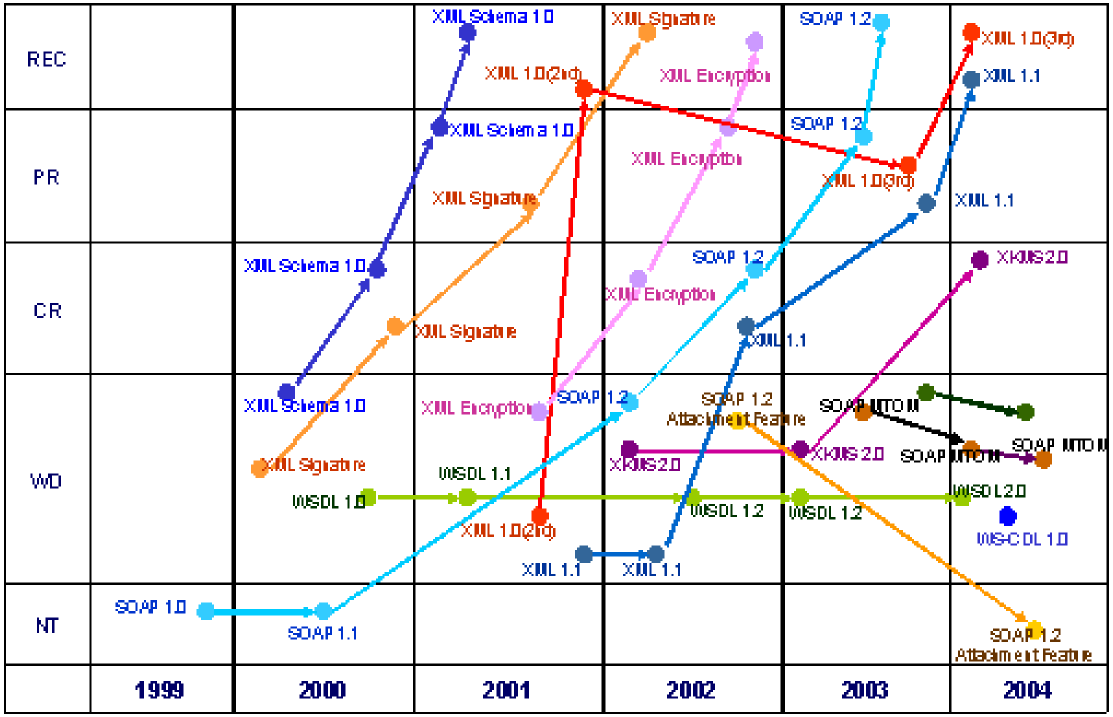
OASIS's e-Business stack

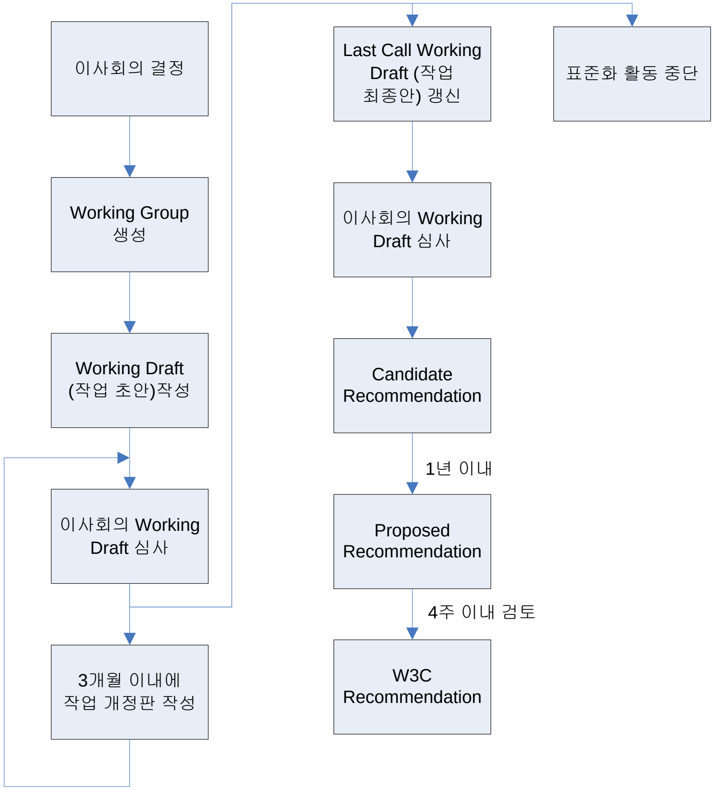


웹 서비스 관련 OASIS TCs

분류	기술위원회 이름
웹 서비스 관련	TWS; Translation Web Services TC
	UDDI; UDDI Specification TC
	WSBPEL; Web Services Business Process Execution Language TC
	BTP; Business Transaction Protocol TC
	WSDM; Web Services Distributed Management TC
	WSIA; Web Services Interactive Application TC
	WSRM; Web Services Reliable Messaging TC
	WSRP; Web Services for Remote Portlet TC
	WSS; Web Services Security TC
ebXML 관련	Business-Centric Methodology TC
	ebXML CPPA TC
	ebXML Implementation TC
	ebXML Messaging TC
	ebXML Registry TC
	Universal Business Language TC

Evolution of WS-related standards at W3C





이사회의 결정

Working Group 생성

Working Draft (작업 초안)작성

이사회의 Working Draft 심사

3개월 이내에 작업 개정판 작성

Last Call Working Draft (작업 최종안) 갱신

표준화 활동 중단

이사회의 Working Draft 심사

Candidate Recommendation

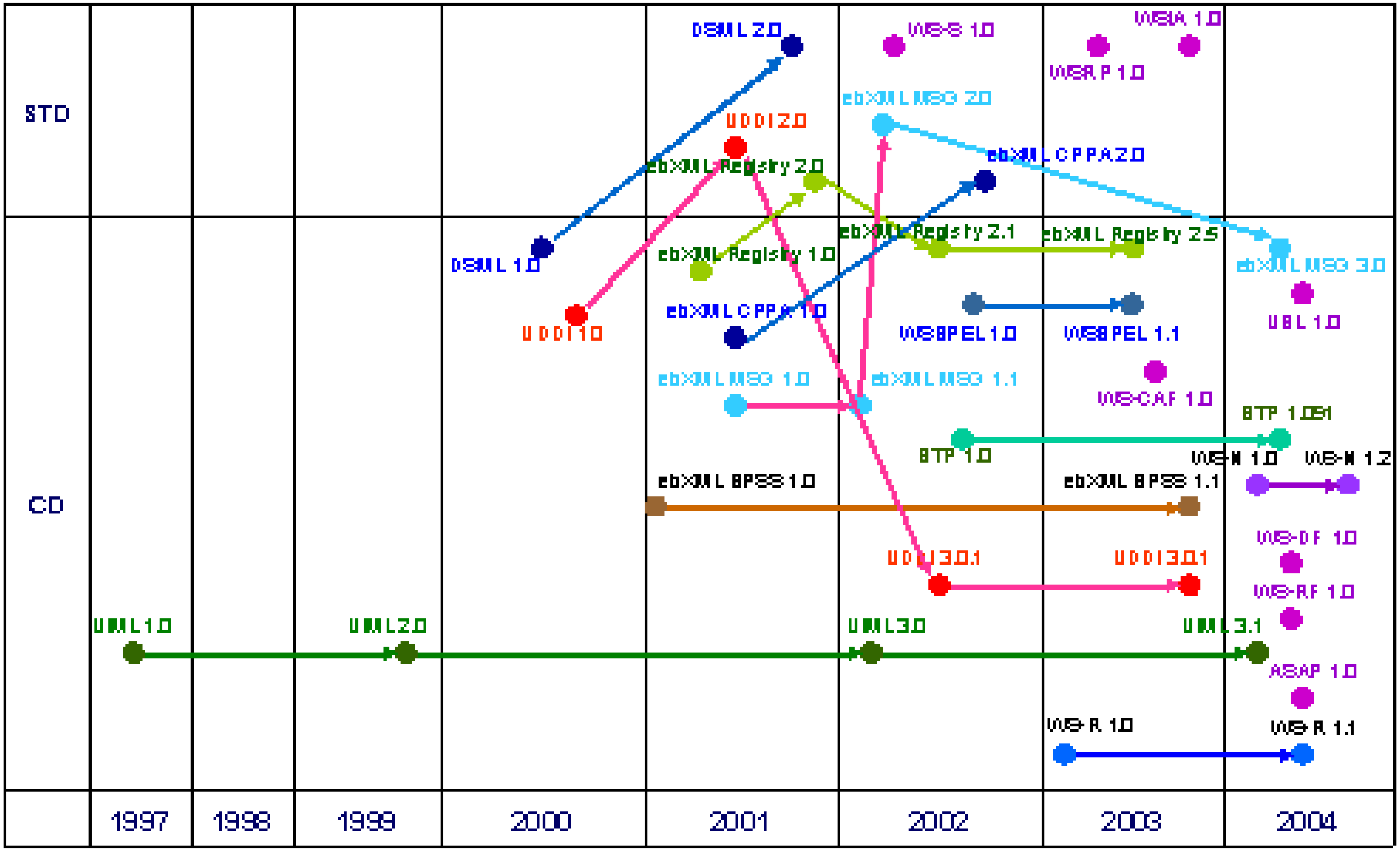
1년 이내

Proposed Recommendation

4주 이내 검토

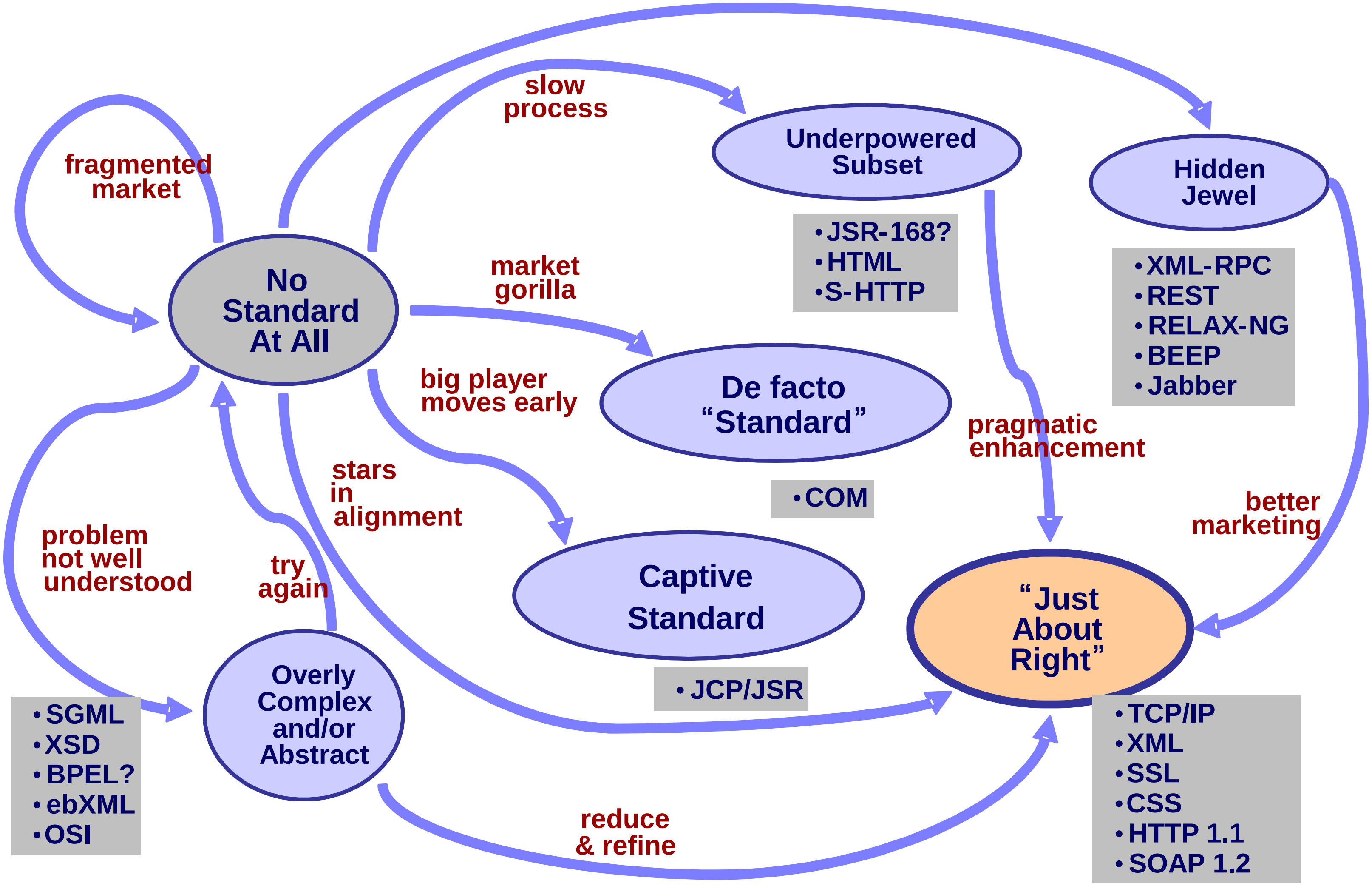
W3C Recommendation

Evolution of WS-related standards at OASIS



A state diagram for standards

(Gartner, 2003)



No Standard At All

fragmented market

slow process

market gorilla

big player moves early

stars in alignment

problem not well understood

try again

Underpowered Subset

- JSR-168?
- HTML
- S-HTTP

Hidden Jewel

- XML-RPC
- REST
- RELAX-NG
- BEEP
- Jabber

De facto “Standard”

- COM

Captive Standard

- JCP/JSR

pragmatic enhancement

better marketing

“Just About Right”

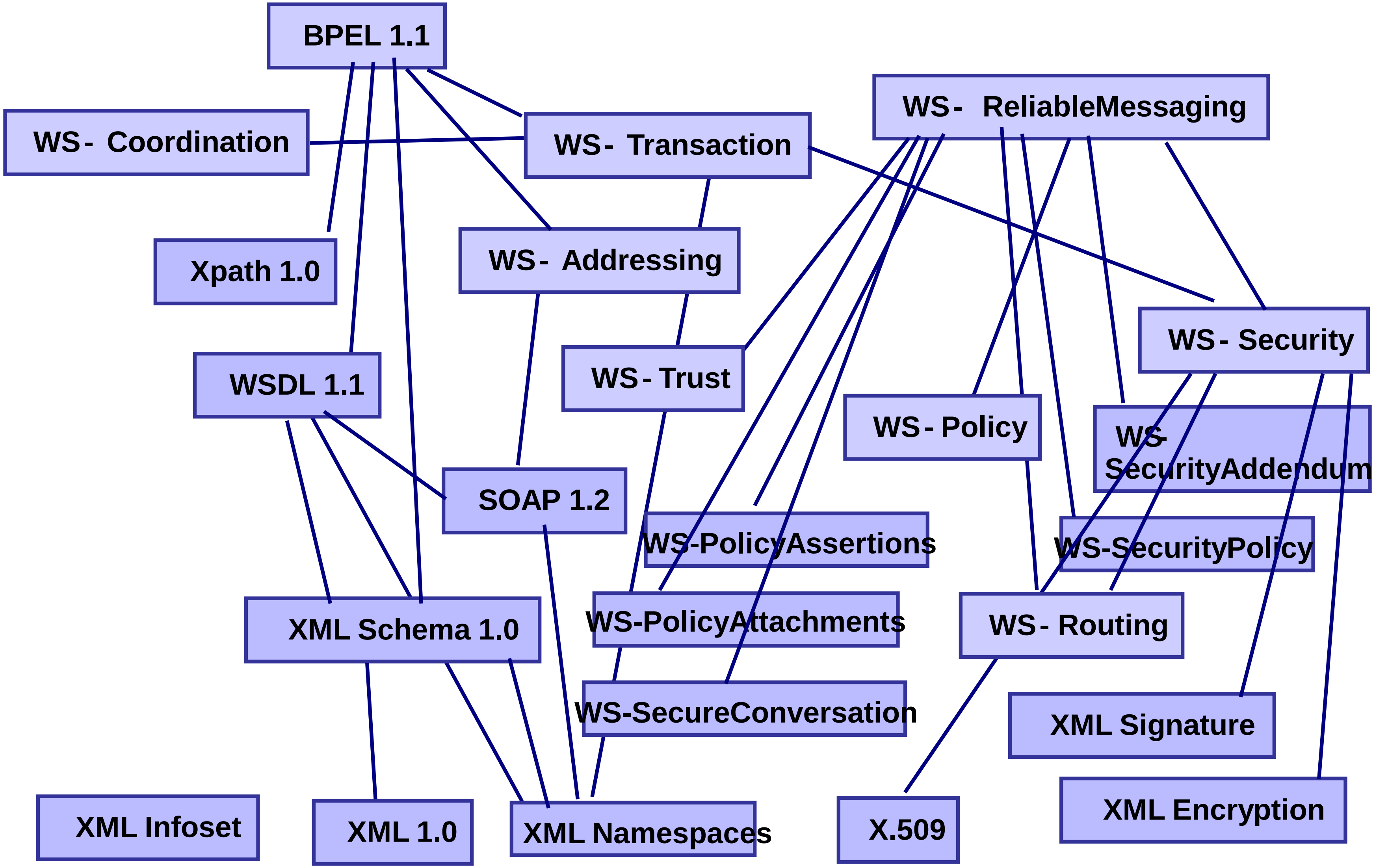
- TCP/IP
- XML
- SSL
- CSS
- HTTP 1.1
- SOAP 1.2

Overly Complex and/or Abstract

- SGML
- XSD
- BPEL?
- ebXML
- OSI

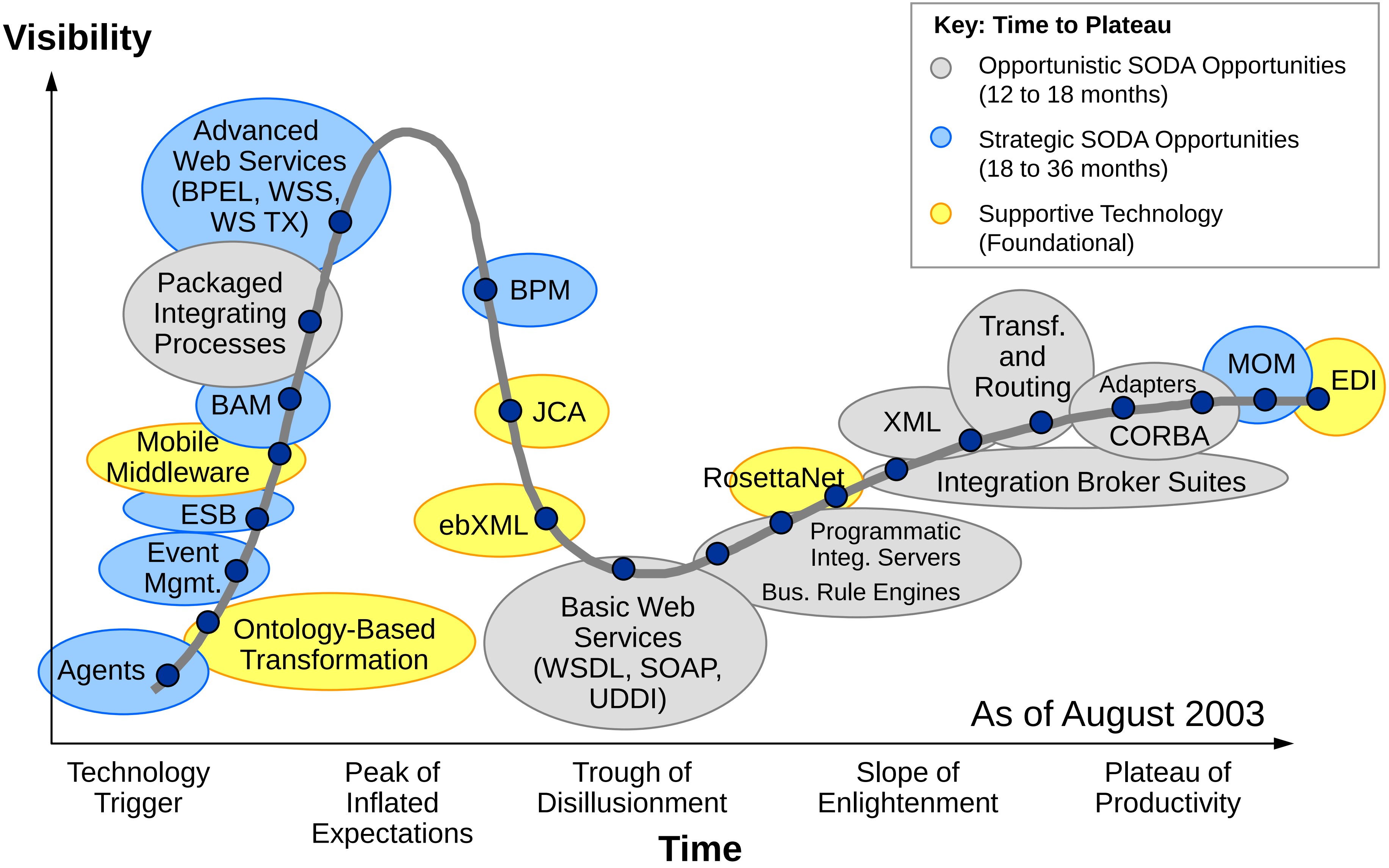
reduce & refine

웹 서비스 표준 간의 상호의존성



웹 서비스의 전략적 응용 분야

(Gartner, 2004)

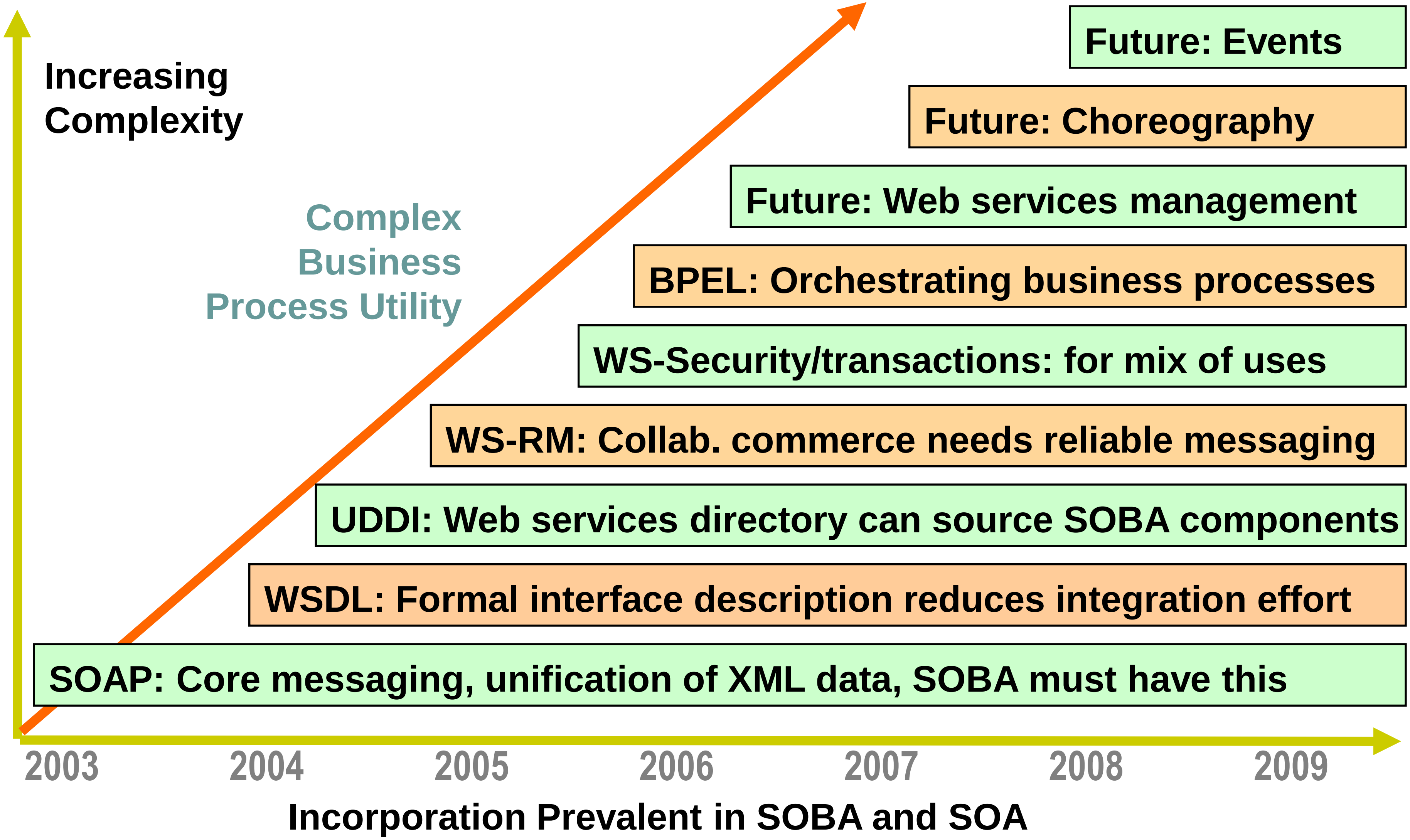

Visibility
Advanced Web Services (BPEL, WSS, WS TX)
Packaged Integrating Processes
BAM
Mobile Middleware
ESB
Event Mgmt.
Ontology-Based Transformation
Agents
BPM
JCA
ebXML
Basic Web Services (WSDL, SOAP, UDDI)
Programmatic Integ. Servers
Bus. Rule Engines
RosettaNet
Integration Broker Suites
XML
Transf. and Routing
Adapters
CORBA
MOM
EDI
As of August 2003
Technology Trigger
Peak of Inflated Expectations
Trough of Disillusionment
Slope of Enlightenment
Plateau of Productivity
Time

Key: Time to Plateau

- ○ Opportunistic SODA Opportunities (12 to 18 months)
- ● Strategic SODA Opportunities (18 to 36 months)
- ● Supportive Technology (Foundational)

웹 서비스 표준의 발전과 SOBA

(Gartner, 2004)



Future: Events

Future: Choreography

Future: Web services management

BPEL: Orchestrating business processes

WS-Security/transactions: for mix of uses

WS-RM: Collab. commerce needs reliable messaging

UDDI: Web services directory can source SOBA components

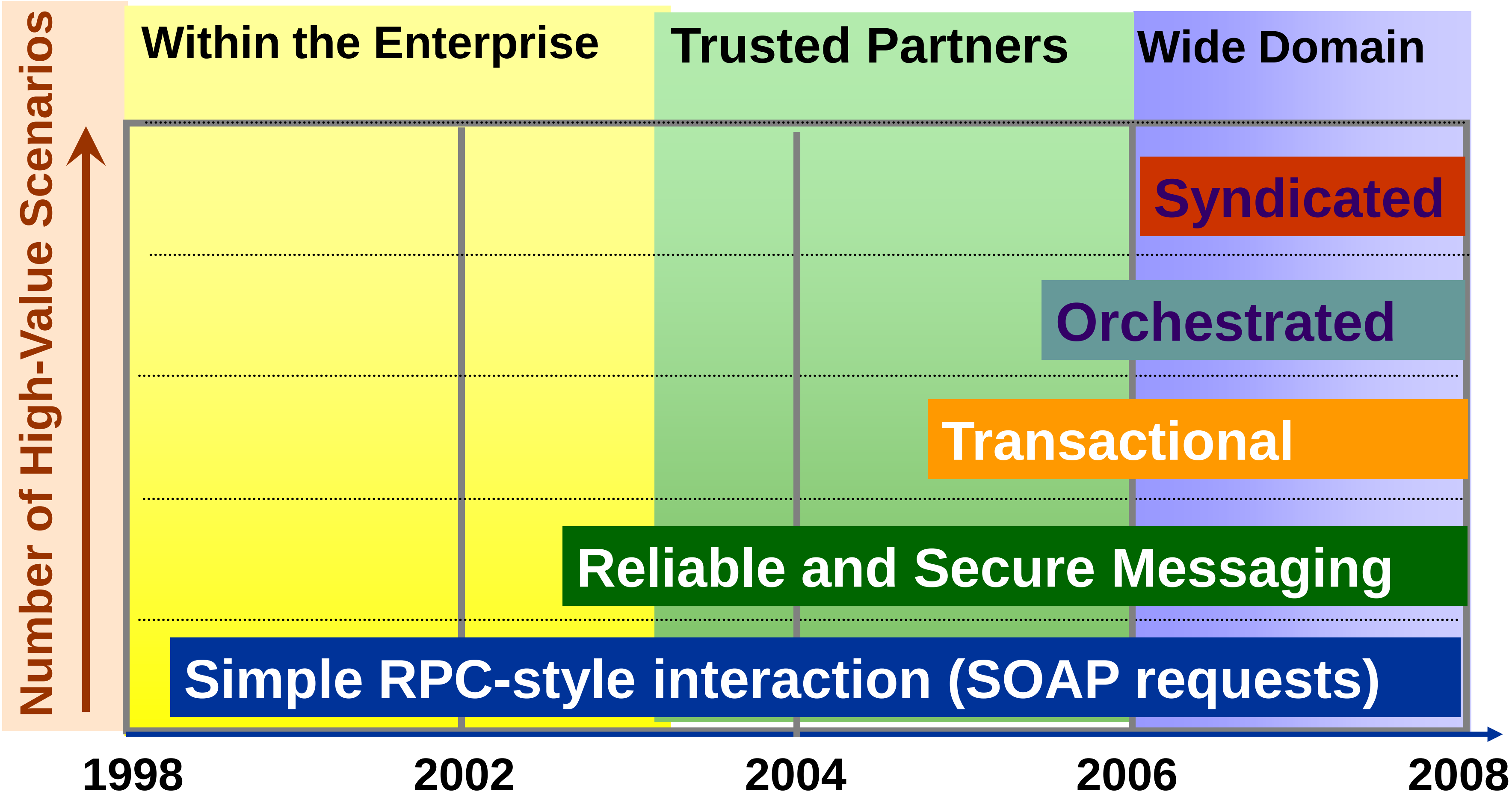
WSDL: Formal interface description reduces integration effort

SOAP: Core messaging, unification of XML data, SOBA must have this

Increasing Complexity
Complex Business Process Utility
2003
2004
2005
2006
2007
2008
2009
Incorporation Prevalent in SOBA and SOA

향후 웹 서비스 프로젝트의 진행 방향

(Gartner, 2004)


Number of High-Value Scenarios
Within the Enterprise
Trusted Partners
Wide Domain
Syndicated
Orchestrated
Transactional
Reliable and Secure Messaging
Simple RPC-style interaction (SOAP requests)
1998
2002
2004
2006
2008

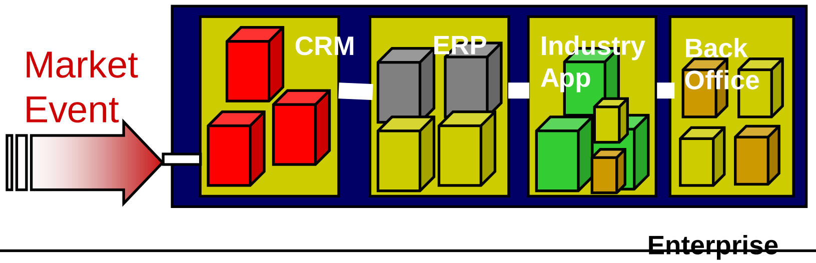
Adoption of Web Services Protocols

웹 서비스를 통한 비즈니스 환경의 변화

(Gartner, 2004)

Business process fusion aims for transformational change in business capabilities by coherent IT support for dynamic, time-sensitive, end-to-end business processes

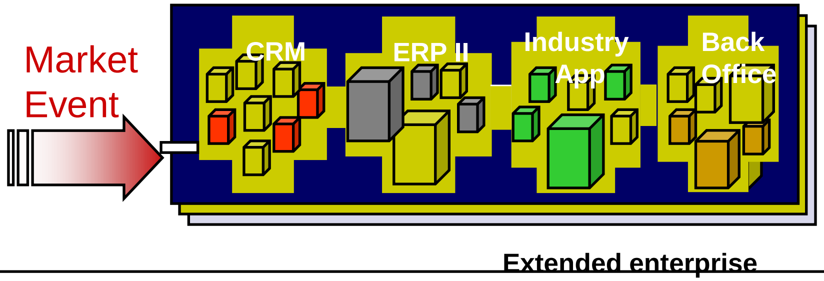
1998: Rigid but Simple
Proprietary, Monolithic Application Suites & Modules



Business Response
Enterprise Centric

Business Advantage
Reactive Enterprise

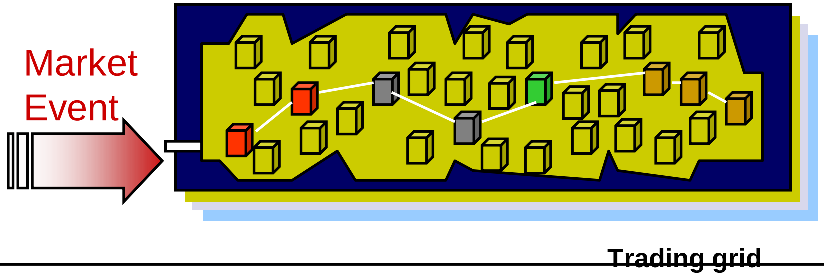
2003: Flexible but Complex
Net-Enabled, Services Wrapped Application



Business Response
Loosely Coupled Value Chain

Business Advantage
Value Chain Visibility
Proactive Enterprise

2008: Fusion & Consolidation
SOA- and BPM-enabled, Business Process Fusion



Business Response
Process-Driven Value Network

Business Advantage
Agility, Adaptability

web services

406.424 Internet applications

Jonghun Park

jonghun@snu.ac.kr

Dept. of Industrial Engineering

Seoul National University

9/20/2007

SOC

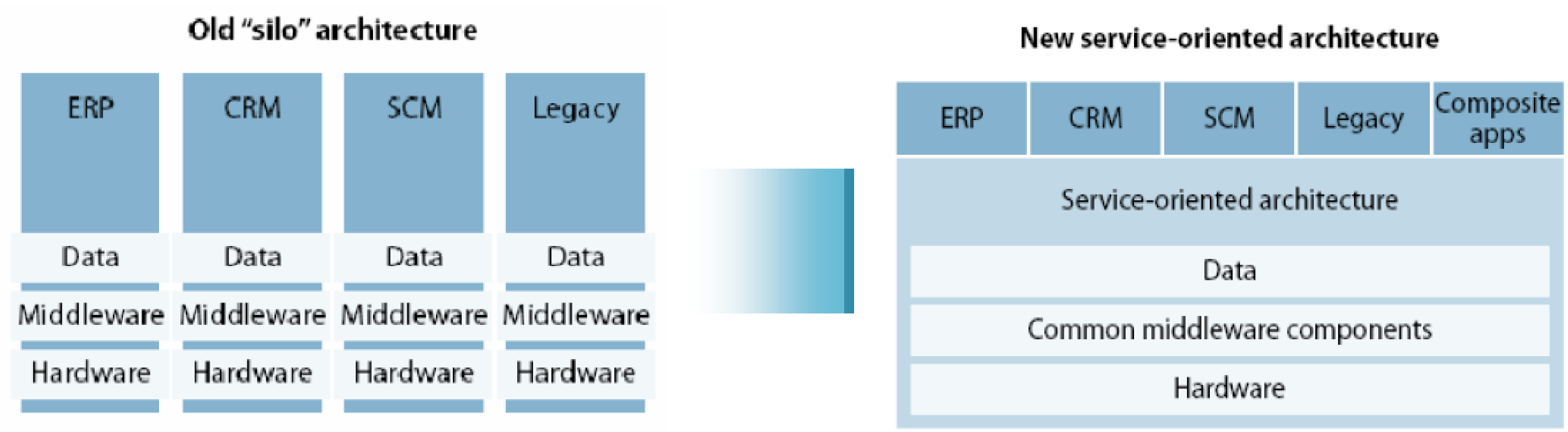
기업의 내 /외부 시스템의 통합 및 연계를 용이하게 하며, 기 개발된 서비스를 재사용하는 시스템 아키텍쳐로 진화됨

레거시 애플리케이션을 재사용 가능한 서비스로 활용

표준 기반이므로 IT 시스템의 통합을 쉽고 빠르게 지원

서비스 합성으로 신규 서비스를 생성

Standard & Component based architecture



Old "silo" architecture

ERP	CRM	SCM	Legacy
Data	Data	Data	Data
Middleware	Middleware	Middleware	Middleware
Hardware	Hardware	Hardware	Hardware

New service-oriented architecture

ERP | CRM | SCM | Legacy | Composite apps

Service-oriented architecture

Data

Common middleware components

Hardware

(출처: Forrester Research)

SAP ORACLE IBM Microsoft

주요 웹 서비스 적용 분야의 예

	CRM	E-Commerce	ERP	E-Procurement	Supply Chain Management
Challenge No. 1	Coordination among departments	Security	BPR	BPR	Culture
Challenge No. 2	Integration	Integration	Inter-enterprise integration	Implementation	Coordination among partners, suppliers, customers
Challenge No. 3	BPR	BPR	Continuous improvement	High cost	Coordination among business units
Goals (large enterprise)	Drive revenue growth	More revenue, more customers, better loyalty	Operational efficiency	Cost savings	Improve production costs; advanced fulfillment
Goals (midsize enterprise)	Improve productivity, Single view of customer	Better customer loyalty	Operational efficiency	Cost savings, shorten transaction time	Enable collaborative production
Goals (small enterprise)	Improve sales productivity	More revenue, More customers	Operational efficiency	Cost savings	Improve logistics/ production/ warehouse costs

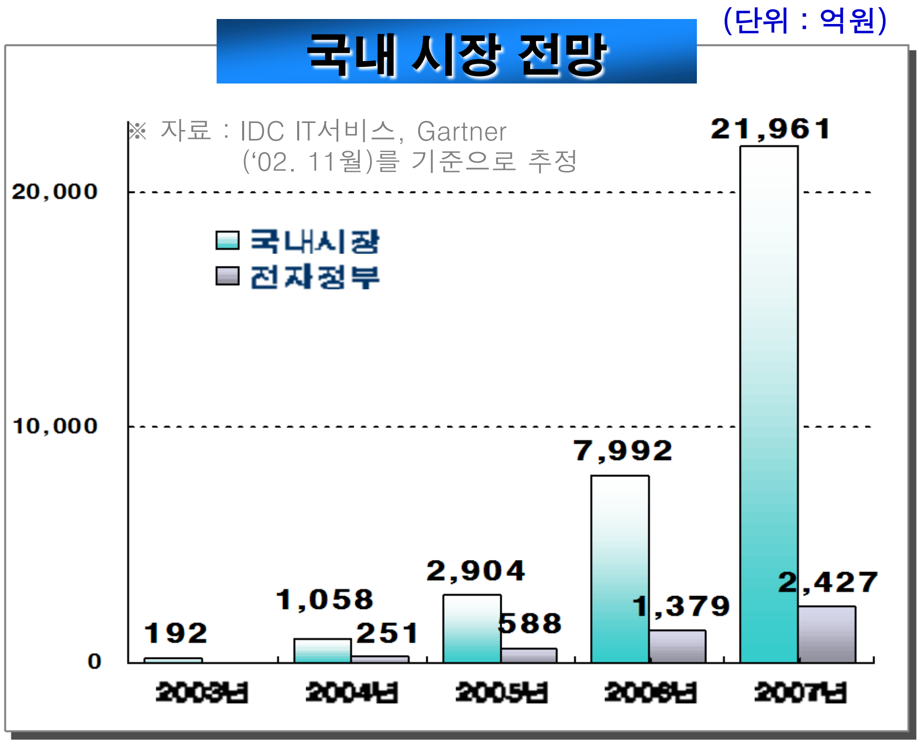
Yellow = Web services target

해외 공공 부문 웹 서비스 적용 사례

- 미국
  - 전자정부에 웹 서비스를 도입하기 위해 웹 서비스 워킹그룹을 중심으로 파일럿 프로젝트를 진행한 데 이어, 현재 공공부문 적용을 본격 추진 중
  - 법무부는 범죄자 지문기록, 범죄이력 등을 웹 서비스로 제공, 교통경찰관이 단말기를 통해 법무성의 데이터를 조회할 수 있도록 하고 있음
- 영국
  - 게이트웨이 프로젝트 (www.gateway.gov.uk)에 SOAP 프로토콜을 이용한 데이터 교환 수준의 웹서비스를 적용, 전자정부 통합을 위한 표준 모델을 제시
  - e-GIF (e-Government Interoperability Framework)를 통해 정부기관과 국민, 정부기관과 기업체, 정부기관과 공공기관 등의 원활한 정보교환을 위한 가이드라인을 제시
- 호주
  - 통계청이 XML 스키마를 이용한 데이터 통합과 웹 서비스 기반의 시스템 통합 아키텍처를 도입, 통계청 내・외부 시스템과의 상호운영성을 확보

국내 웹 서비스 시장 및 도입 사례

(Source: 정통부, 2004)



※ 자료 : IDC IT서비스, Gartner ('02. 11월)를 기준으로 추정

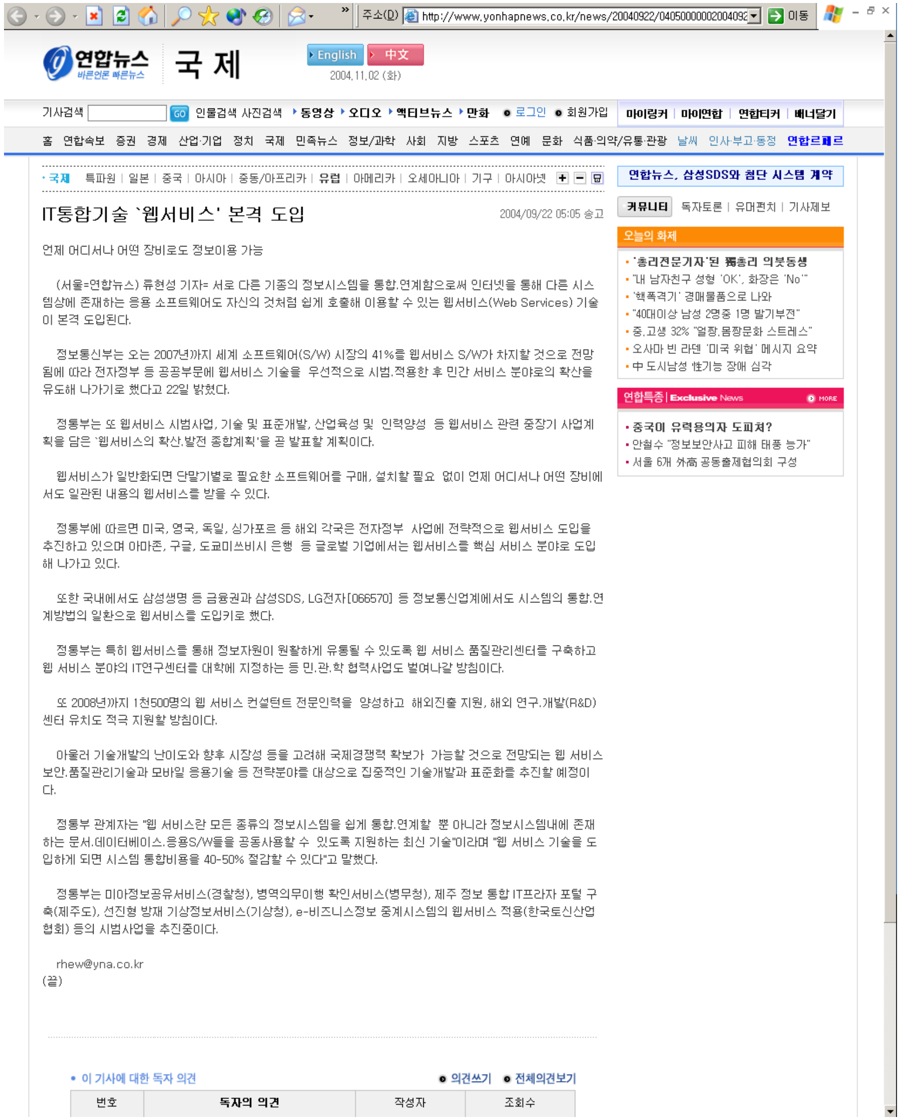
공공

사 업	도 입 사 례
G4C	주민등록, 토지대장, 지방세 및 자동차납세증명 등 15개 공공 문서 교환에 웹 서비스 초기모델 도입
디지털 청와대 사업	다부처 시스템간 연계와 내부시스템 통합에 웹 서비스 도입 (청와대, 인사위 : 업무관리시스템, 정통부 : GPLC)

민간

업 종	도 입 사 례
금융권 (삼성생명 등)	은행 및 보험사 등에서 내부 시스템 연계 및 파트너 연계에 웹 서비스 초기 모델 시범사업 실시
전자업계 (LG, 삼성 등)	삼성전자는 해외 비즈니스 파트너와 시스템 연계을 웹 서비스로 연계하여 수발주 업무 처리

국내 웹 서비스 현황


주소(D) http://www.yonhapnews.co.kr/news/20040922/04050000002004092
이동
연합뉴스
바른언론 빠른뉴스
국 제
English
中文
2004.11.02 (화)
기사검색
GO
인물검색 사진검색
동영상
오디오
액티브뉴스
만화
로그인
회원가입
마이링커 | 마이연합 | 연합티커 | 배너달기
홈 연합속보 증권 경제 산업·기업 정치 국제 민족뉴스 정보/과학 사회 지방 스포츠 연예 문화 식품·의약/유통·관광 날씨 인사·부고·동정 연합르페르
국제 특파원 | 일본 | 중국 | 아시아 | 중동/아프리카 | 유럽 | 아메리카 | 오세아니아 | 기구 | 아시아넷
연합뉴스, 삼성SDS와 첨단 시스템 계약
커뮤니티 독자토론 | 유머펀치 | 기사제보
오늘의 화제
연합특종 | Exclusive News
MORE
의견쓰기
전체의견보기

IT통합기술 `웹서비스' 본격 도입

2004/09/22 05:05 송고

언제 어디서나 어떤 장비로도 정보이용 가능

(서울=연합뉴스) 류현성 기자= 서로 다른 기종의 정보시스템을 통합.연계함으로써 인터넷을 통해 다른 시스템상에 존재하는 응용 소프트웨어도 자신의 것처럼 쉽게 호출해 이용할 수 있는 웹서비스(Web Services) 기술이 본격 도입된다.

정보통신부는 오는 2007년까지 세계 소프트웨어(S/W) 시장의 41%를 웹서비스 S/W가 차지할 것으로 전망됨에 따라 전자정부 등 공공부문에 웹서비스 기술을 우선적으로 시범.적용한 후 민간 서비스 분야로의 확산을 유도해 나가기로 했다고 22일 밝혔다.

정통부는 또 웹서비스 시범사업, 기술 및 표준개발, 산업육성 및 인력양성 등 웹서비스 관련 중장기 사업계획을 담은 `웹서비스의 확산.발전 종합계획'을 곧 발표할 계획이다.

웹서비스가 일반화되면 단말기별로 필요한 소프트웨어를 구매, 설치할 필요 없이 언제 어디서나 어떤 장비에서도 일관된 내용의 웹서비스를 받을 수 있다.

정통부에 따르면 미국, 영국, 독일, 싱가포르 등 해외 각국은 전자정부 사업에 전략적으로 웹서비스 도입을 추진하고 있으며 아마존, 구글, 도쿄미쓰비시 은행 등 글로벌 기업에서는 웹서비스를 핵심 서비스 분야로 도입해 나가고 있다.

또한 국내에서도 삼성생명 등 금융권과 삼성SDS, LG전자[066570] 등 정보통신업계에서도 시스템의 통합.연계방법의 일환으로 웹서비스를 도입키로 했다.

정통부는 특히 웹서비스를 통해 정보자원이 원활하게 유통될 수 있도록 웹 서비스 품질관리센터를 구축하고 웹 서비스 분야의 IT연구센터를 대학에 지정하는 등 민.관.학 협력사업도 벌여나갈 방침이다.

또 2008년까지 1천500명의 웹 서비스 컨설턴트 전문인력을 양성하고 해외진출 지원, 해외 연구.개발(R&D) 센터 유치도 적극 지원할 방침이다.

아울러 기술개발의 난이도와 향후 시장성 등을 고려해 국제경쟁력 확보가 가능할 것으로 전망되는 웹 서비스 보안.품질관리기술과 모바일 응용기술 등 전략분야를 대상으로 집중적인 기술개발과 표준화를 추진할 예정이다.

정통부 관계자는 "웹 서비스란 모든 종류의 정보시스템을 쉽게 통합.연계할 뿐 아니라 정보시스템내에 존재하는 문서.데이터베이스.응용S/W들을 공동사용할 수 있도록 지원하는 최신 기술"이라며 "웹 서비스 기술을 도입하게 되면 시스템 통합비용을 40-50% 절감할 수 있다"고 말했다.

정통부는 미아정보공유서비스(경찰청), 병역의무이행 확인서비스(병무청), 제주 정보 통합 IT프라자 포털 구축(제주도), 선진형 방재 기상정보서비스(기상청), e-비즈니스정보 중계시스템의 웹서비스 적용(한국토신산업협회) 등의 시범사업을 추진중이다.

rhew@yna.co.kr
(끝)

- '총리전문기자'된 獨총리 의붓동생
- "내 남자친구 성형 'OK', 화장은 'No'"
- '핵폭격기' 경매물품으로 나와
- "40대이상 남성 2명중 1명 발기부전"
- 중.고생 32% "얼짱.몸짱문화 스트레스"
- 오사마 빈 라덴 '미국 위협' 메시지 요약
- 中 도시남성 性기능 장애 심각

- 중국이 유력용의자 도피처?
- 안철수 "정보보안사고 피해 태풍 능가"
- 서울 6개 外高 공동출제협의회 구성

• 이 기사에 대한 독자 의견

번호	독자의 의견	작성자	조회수

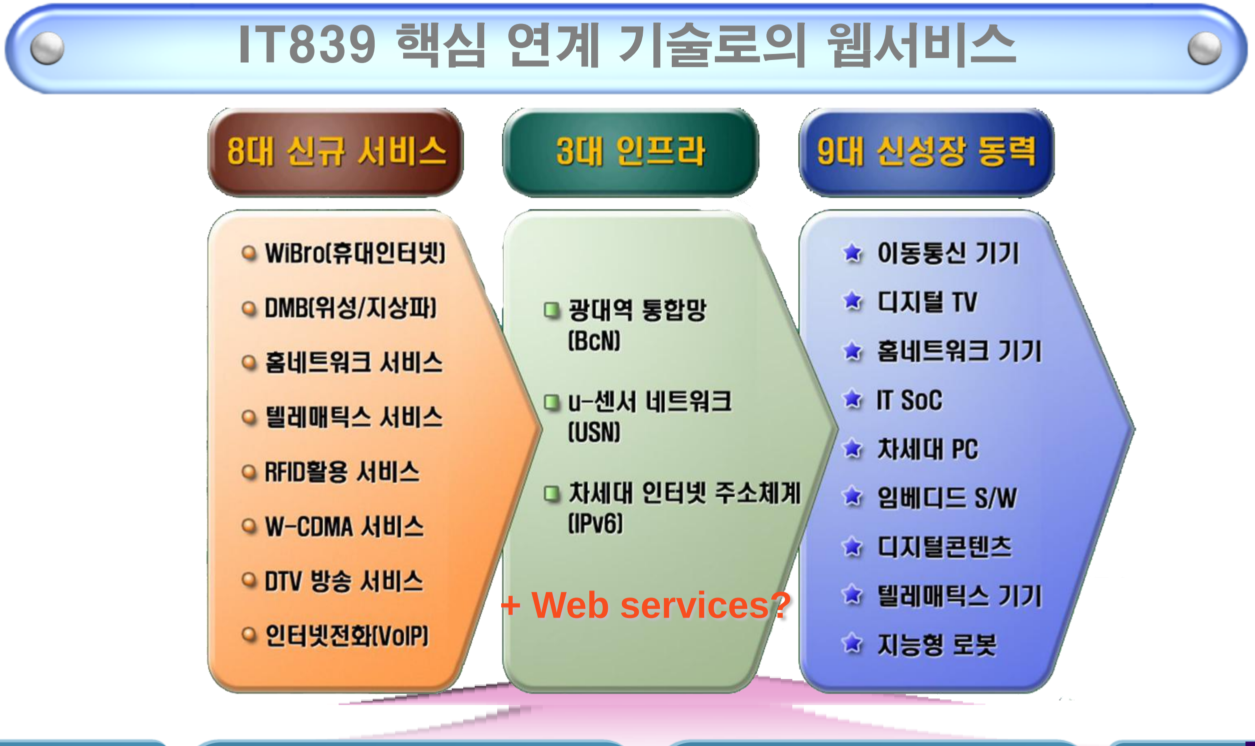
국내 WS 시범 사업:

- 미아정보 공유 서비스 (경찰청)
- 병역의무이행 확인 (병무청)
- 제주 정보 통합 IT 프라자 포털 구축 (제주도)
- 방재 기상정보서비스 (기상청)
- e-비즈니스 정보 중계 시스템 (한국통신산업협회)
- 중소기업 ASP 사업
- 통합 국적관리 시스템 (법무부)
- 영유아 보육교사 포털 (대전시)

전자정부의 기관간 시스템 연계표준으로 지정

IT839에서의 웹 서비스의 위상

(Source: 정통부, 2004)



IT839 핵심 연계 기술로의 웹서비스

8대 신규 서비스

- WiBro(휴대인터넷)
- DMB(위성/지상파)
- 홈네트워크 서비스
- 텔레매틱스 서비스
- RFID활용 서비스
- W-CDMA 서비스
- DTV 방송 서비스
- 인터넷전화(VoIP)

3대 인프라

- 광대역 통합망 (BcN)
- u-센서 네트워크 (USN)
- 차세대 인터넷 주소체계 (IPv6)

+ Web services?

9대 신성장 동력

- 이동통신 기기
- 디지털 TV
- 홈네트워크 기기
- IT SoC
- 차세대 PC
- 임베디드 S/W
- 디지털콘텐츠
- 텔레매틱스 기기
- 지능형 로봇

개발 도구 기술

- 통합 개발 도구 기술
- 다중 플랫폼 기술

미들웨어 기술

- 경량형 미들웨어 기술
- 협업적 미들웨어 기술

유비쿼터스 WS 기술

- 임베디드 웹서비스 기술
- 웹서비스 디바이스 기술
- 차세대 인터넷 연동 기술
- 웹서비스 네트워크 기술
- 유비쿼터스 미들웨어 통합
- 시맨틱 웹서비스 기술

응용 및 서비스 기술

- 융합 서비스 응용
- 그리드 웹서비스 응용
- 모바일 웹서비스 응용
- P2P 웹서비스 응용
- 협업형 웹서비스 응용
- 비즈니스 웹서비스 응용
- 차세대 웹서비스 응용
- 웹서비스 신디케이션 응용

관리 기술

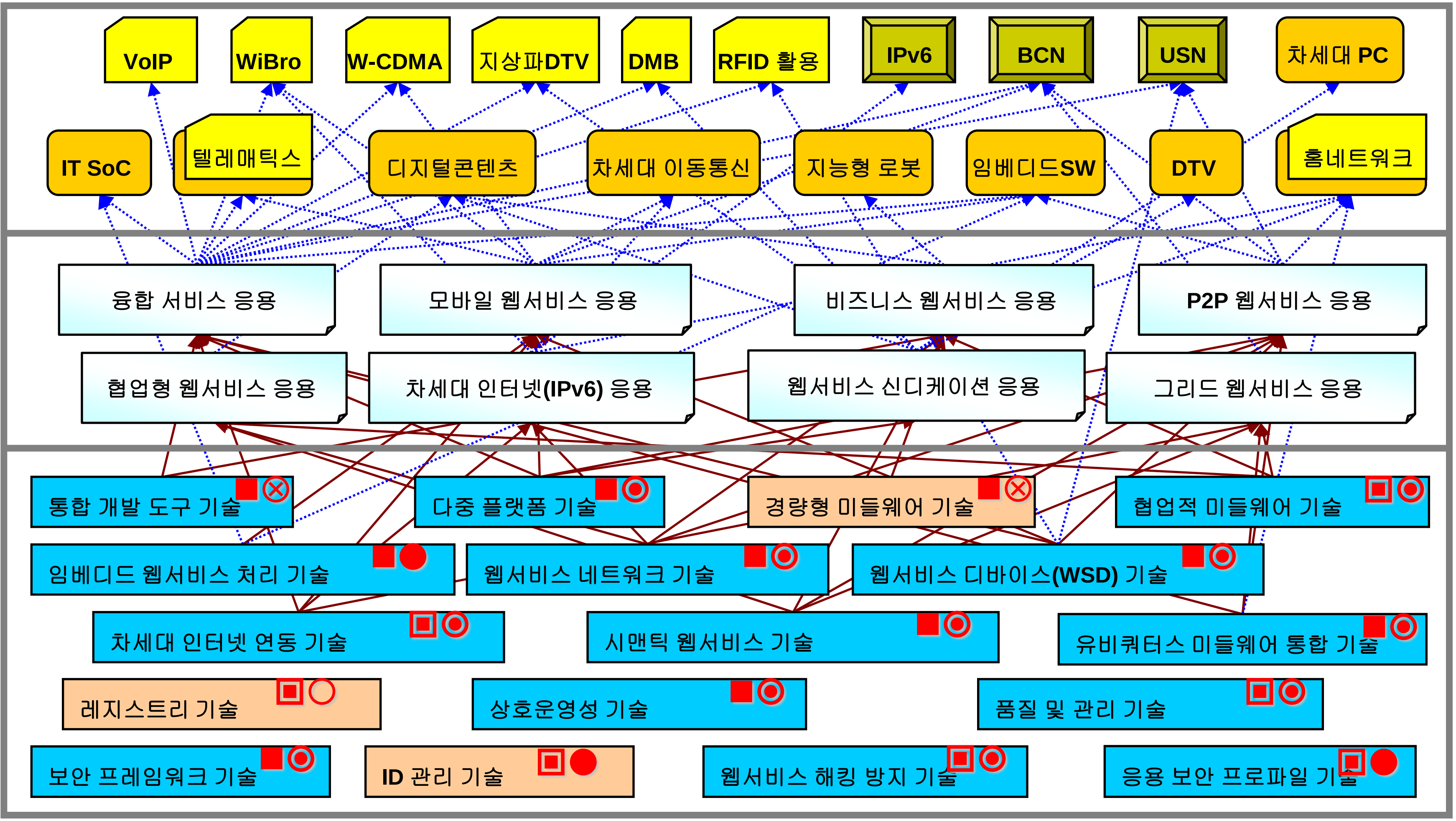
- 레지스트리 기술
- 상호운용성 기술
- 품질 및 관리 기술

보안 기술

- 보안 프레임워크 기술
- 웹서비스 해킹 방지 기술
- ID 관리 기술
- 응용 보안 프로파일 기술

국내 웹 서비스 사업 추진 계획 (안)

(Source: 정통부, 2004)



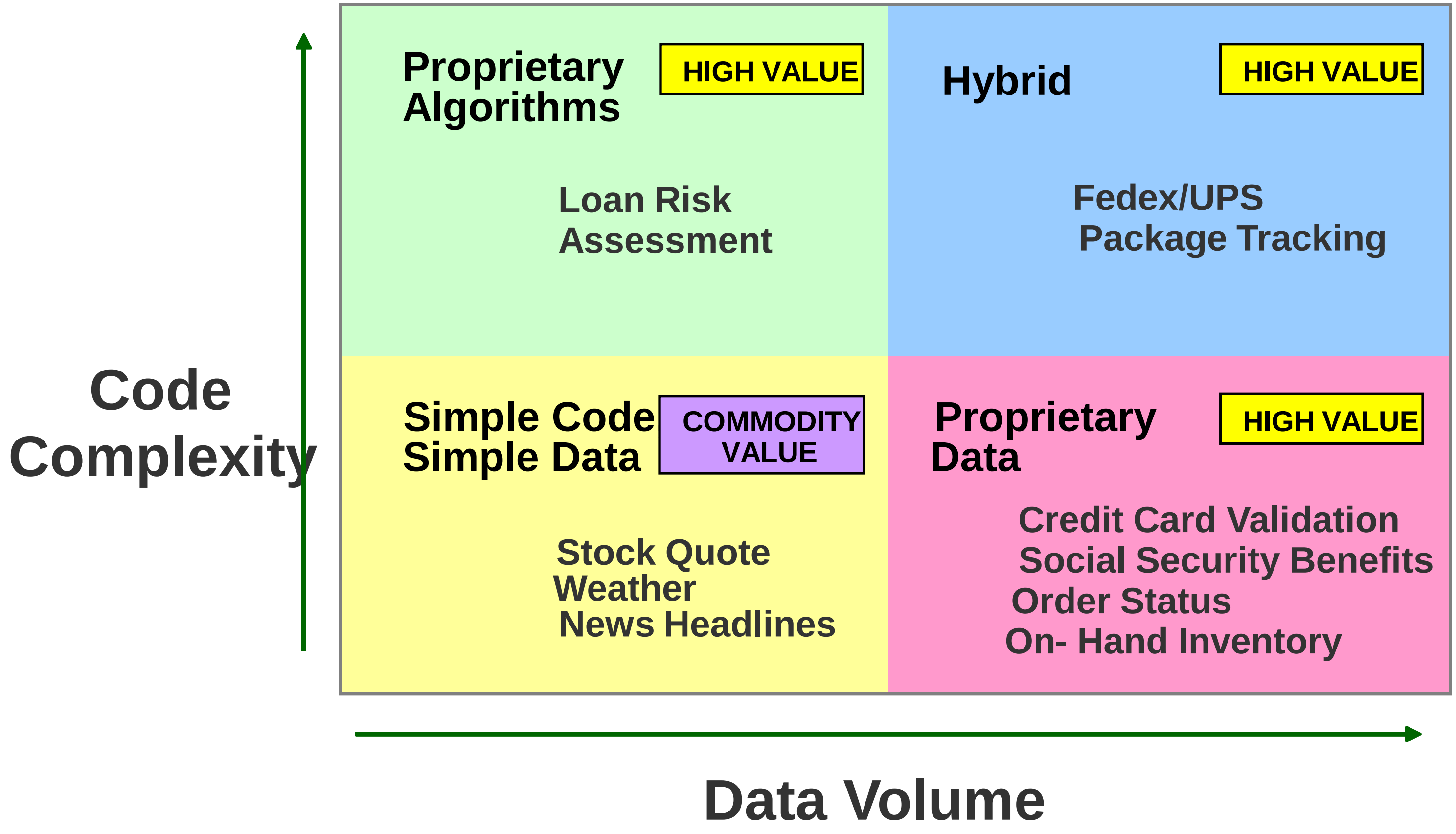
기술 수준
- 🟨 국내성숙기술
- 🟧 국내개발기술 (초기)
- 🟦 국내미비기술

중요도 고 ↔ 저
■ ▣ □

연구개발전략
- ● 기초연구
- ⊗ 실용화 개발
- ⦿ 국제협력분야
- ○ 기술도입

High-value scenarios in Web services

(Gartner, 2003)



Proprietary Algorithms — HIGH VALUE

Loan Risk Assessment

Hybrid — HIGH VALUE

Fedex/UPS Package Tracking

Simple Code Simple Data — COMMODITY VALUE

Stock Quote
Weather
News Headlines

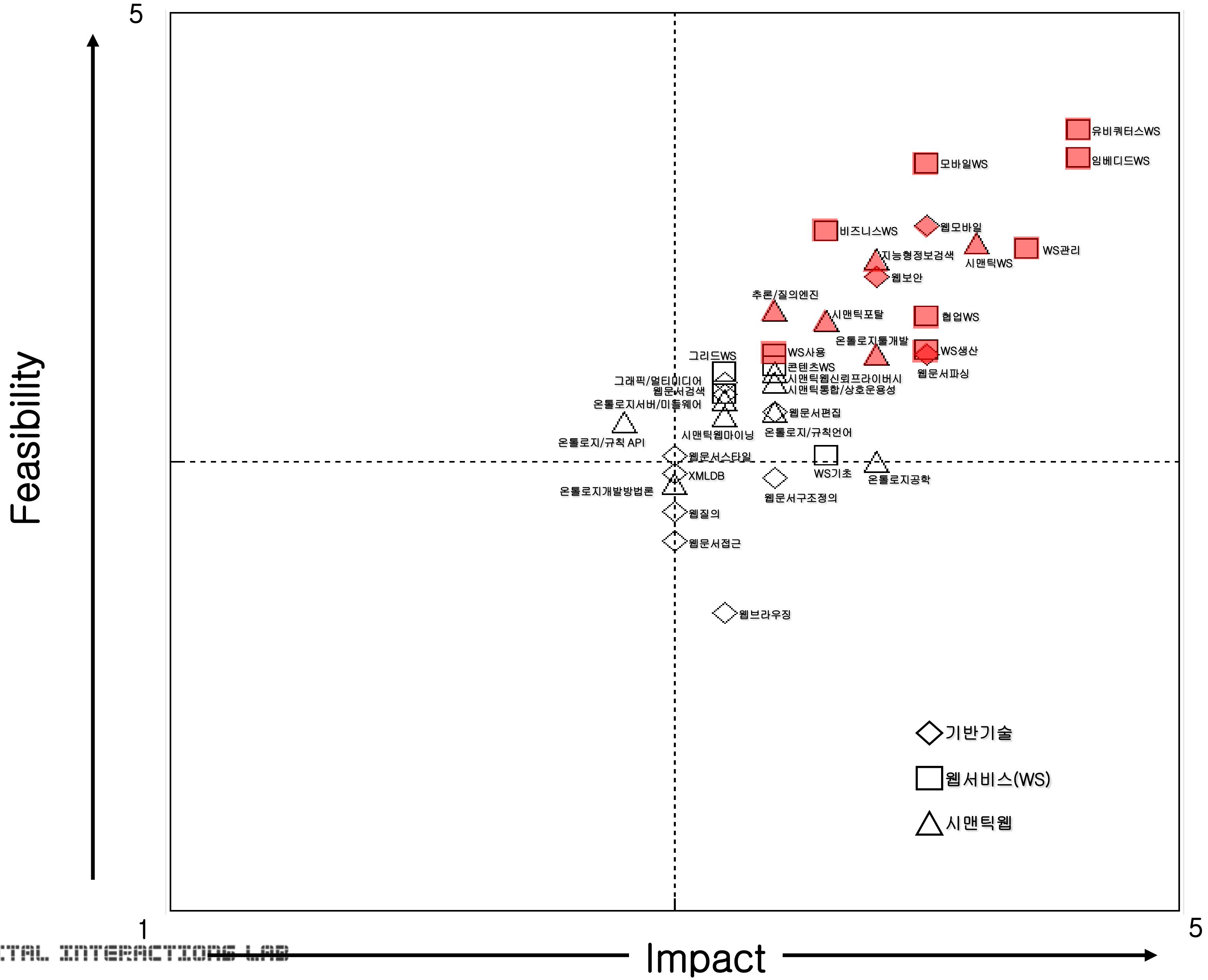
Proprietary Data — HIGH VALUE

Credit Card Validation
Social Security Benefits
Order Status
On- Hand Inventory

Web vs. Web Services

- Increased visibility of web via web services
- Business mind + Technical mind

웹 기술 평가 결과
(출처: 한국 전산원)



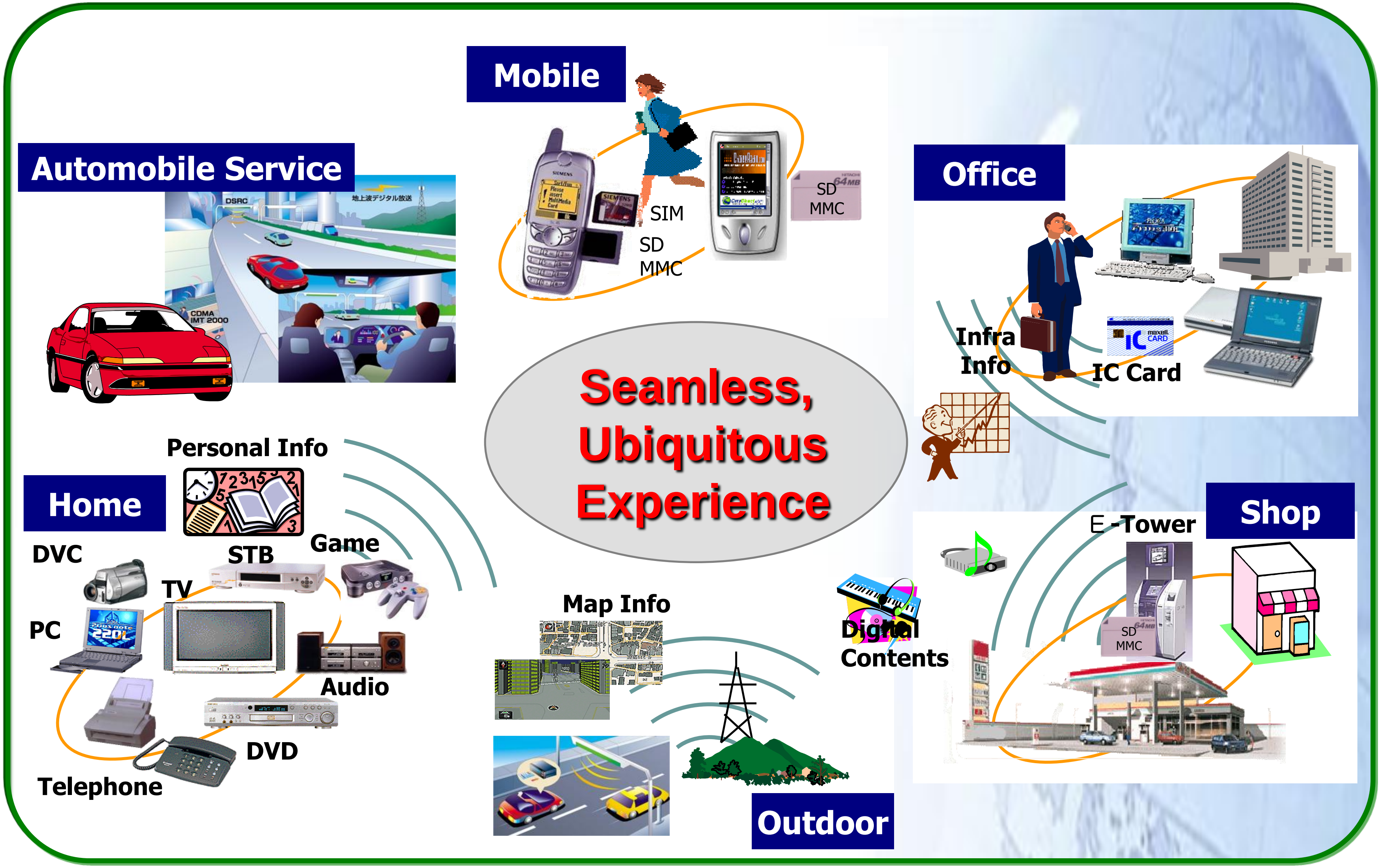
웹 서비스의 교훈 (2001 – 2005)

- 현상
  - .COM bubble 때와 같은 business analogy가 성립하지 않았음
  - 표준의 난립과 기관 간의 주도권 다툼: WS-??
  - 세상은 생각보다 더 closed 되어 있음
    - 예: 신라 호텔이 웹 서비스를 제공 해야하는 이유는?
  - 초기 예상대로 EAI와 B2Bi 관련한 적용 사례가 가장 많았음
    - BPMS boom-up으로 인해 한 때 관심이 고조
  - 막상 전사적으로 도입하기에는 꺼려지지만 무시할 수도 없는 기술
    - IT integration 측면의 잠재적 가치
- 교훈
  - "자동화"적인 특성이 강함
  - Increased visibility of web via web services: e.g., product search
  - 고부가 가치의 웹 서비스 발굴 필요
    - 사고 싶은 정보: 구하기 힘든 정보 (contents) and / or 처리하기 힘든 정보 (information processing) and / or 실시간 정보 (real-time)
  - 일반 대중보다는 기업 대상의 웹 서비스 비즈니스 발굴이 수익성이 있음
  - 기업 (또는 국가)의 IT 자산으로서의 인프라적 성격 및 공공성이 강함
    - 예: Increased product visibility under WS-enabled e-Business
  - 웹 서비스를 "적용하면 좋은" 영역 이외에, "반드시 적용 해야 하는" 영역은 어디인가?

And then comes the ubiquitous computing…

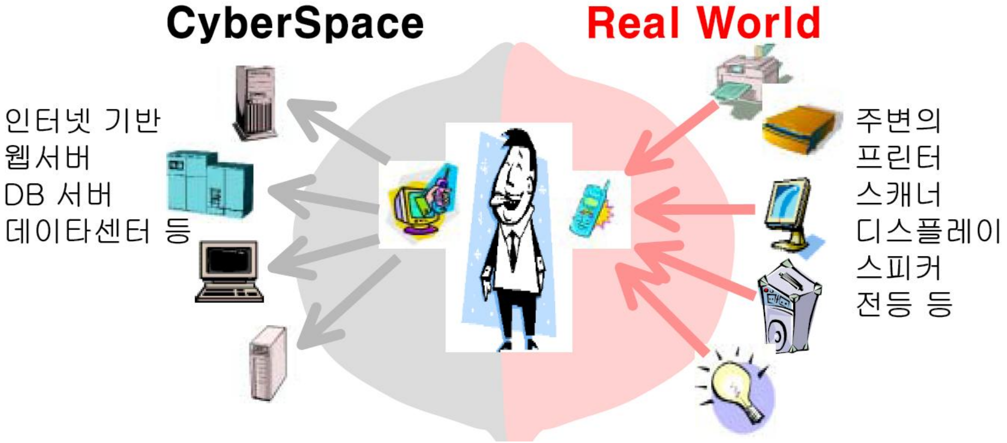
- The grand objective: enhance computer use by making **many computers available throughout the physical environment**, but making them **effectively invisible** to the user (Weiser, CACM, 1993)
- Pushing **computational services** out of conventional desktop interfaces **into environments** characterized by transparent forms of interactivity
- Recently has been accelerated by improved wireless telecommunication capabilities, open networks, continued increases in computing power, improved battery technology, and the emergence of flexible software architectures

유비쿼터스 환경
(출처: ETRI PEC, 2005)



Ubiquitous IT

- 사방 어디에나 있고, 보이지 않는 곳에 숨어 있다는 의미
- 시간과 장소의 구애를 받지 않고 눈에 보이지 않아도 컴퓨팅을 할 수 있게 함
- 물리적 공간과 가상 공간이 합쳐져 새로운 통합 공간을 구현해 냄
- 주변 환경 속에 노출된 모든 사람과 사물을 네트워크로 연결, 사용자가 필요로 하는 정보와 서비스를 제공할 수 있는 기반 기술
- e-비즈니스 -> m-비즈니스 -> u-비즈니스


CyberSpace

Real World

인터넷 기반
웹서버
DB 서버
데이터센터 등

주변의
프린터
스캐너
디스플레이
스피커
전등 등

Ubiquitous computing의 특징

Pervasive / Embedded Computing	Ubiquitous Computing
Traditional Business Computing	Mobile / Wearable Computing

Embeddedness

Computer-embedded objects and devices

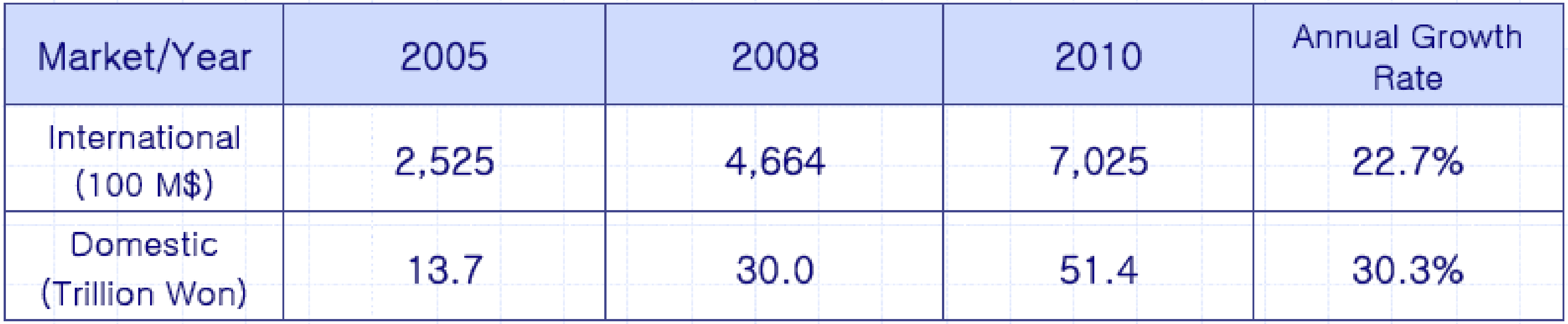
Mobility

5A: Any time, Any where, Any device, Any service, Any network

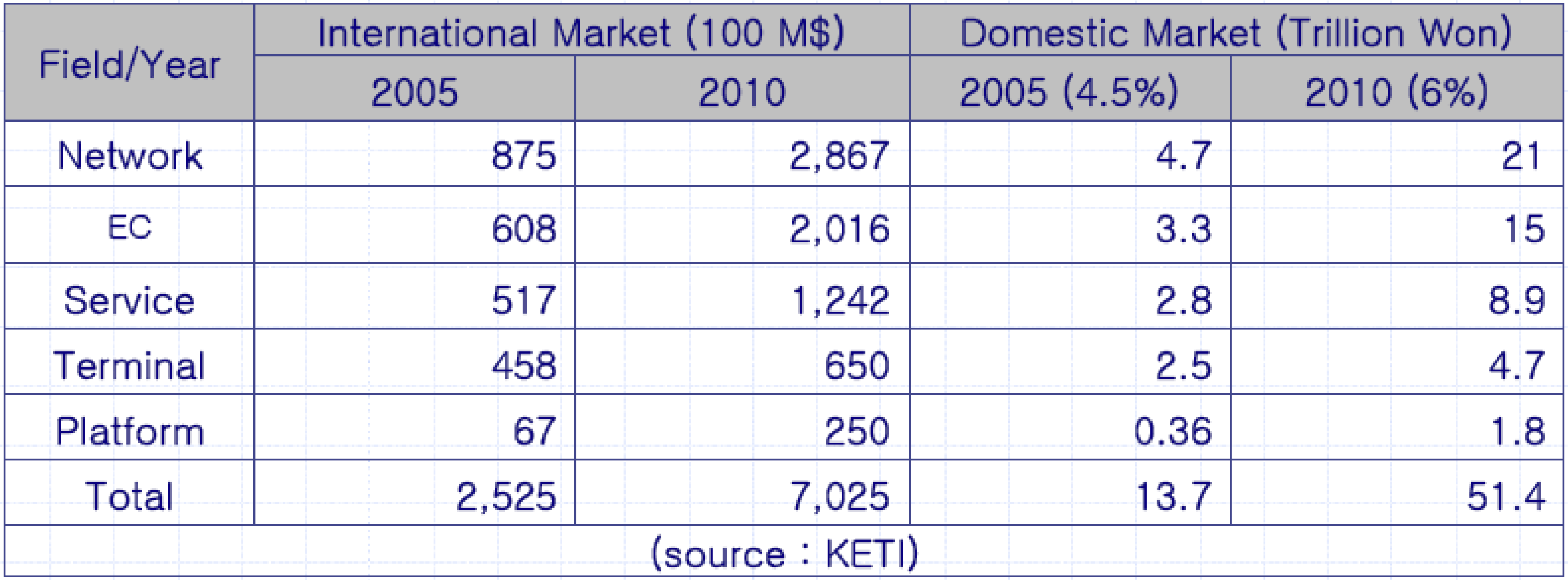
Characteristics of ubiquitous environments

- In the near future, an **enormous number of RFID tags, sensors, and other heterogeneous small devices** will be **embedded** in the real world
  - Events are provided, or often triggered, based on **physical conditions** -> Real-time processing of large amount of data
  - Services need to know the **real-world status** and **users situations** -> Context awareness
  - Services are provided when a user is not expecting them -> **Intrusive** or **invisible**. Attention focus
  - Devices will constitute a **global, open, dynamic networking infrastructure** -> The devices need to be coordinated for better interactions

유비쿼터스 산업 시장



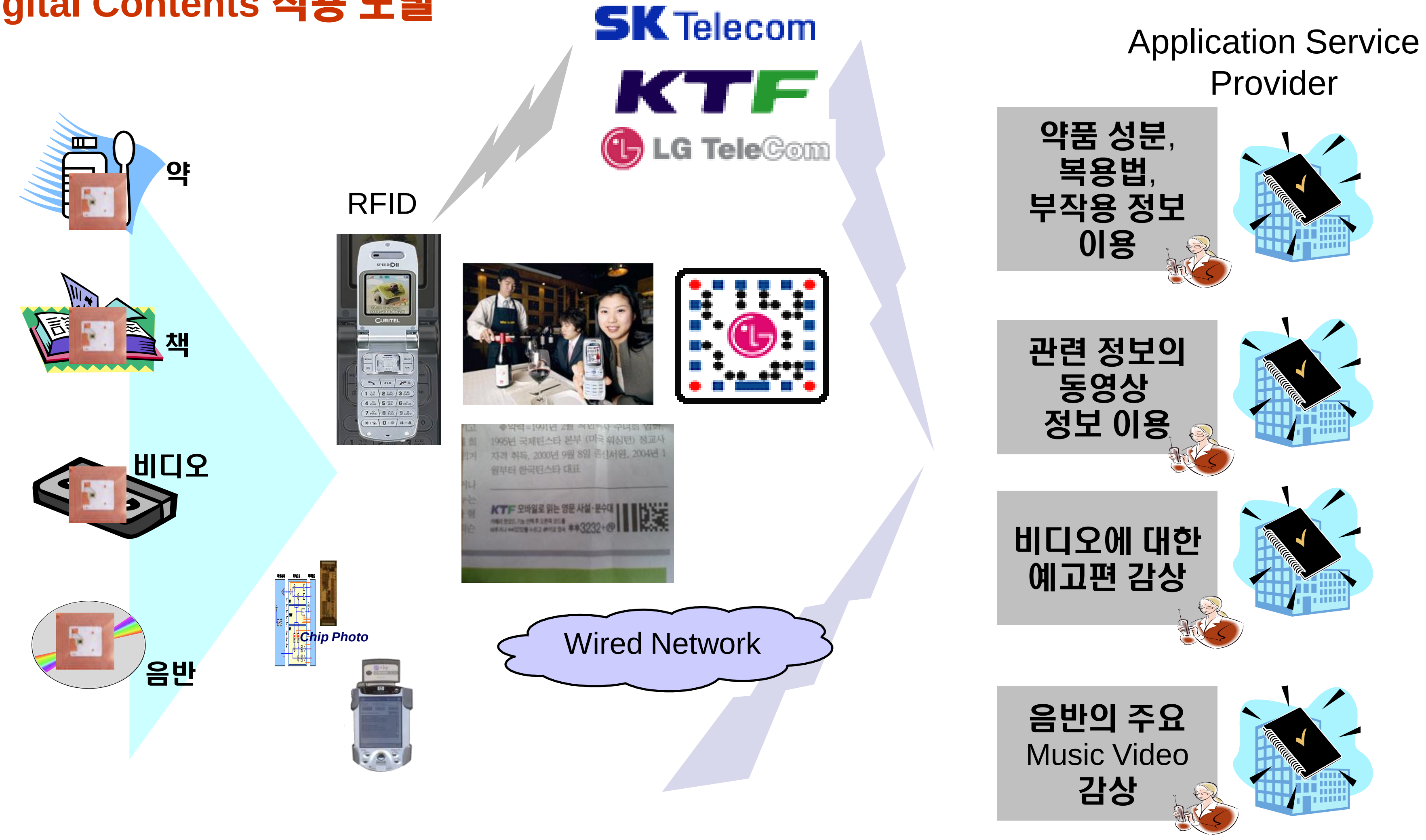
Market/Year	2005	2008	2010	Annual Growth Rate
International (100 M$)	2,525	4,664	7,025	22.7%
Domestic (Trillion Won)	13.7	30.0	51.4	30.3%



Field/Year	International Market (100 M$)		Domestic Market (Trillion Won)	
	2005	2010	2005 (4.5%)	2010 (6%)
Network	875	2,867	4.7	21
EC	608	2,016	3.3	15
Service	517	1,242	2.8	8.9
Terminal	458	650	2.5	4.7
Platform	67	250	0.36	1.8
Total	2,525	7,025	13.7	51.4
(source : KETI)				

Mobile services

Digital Contents 적용 모델



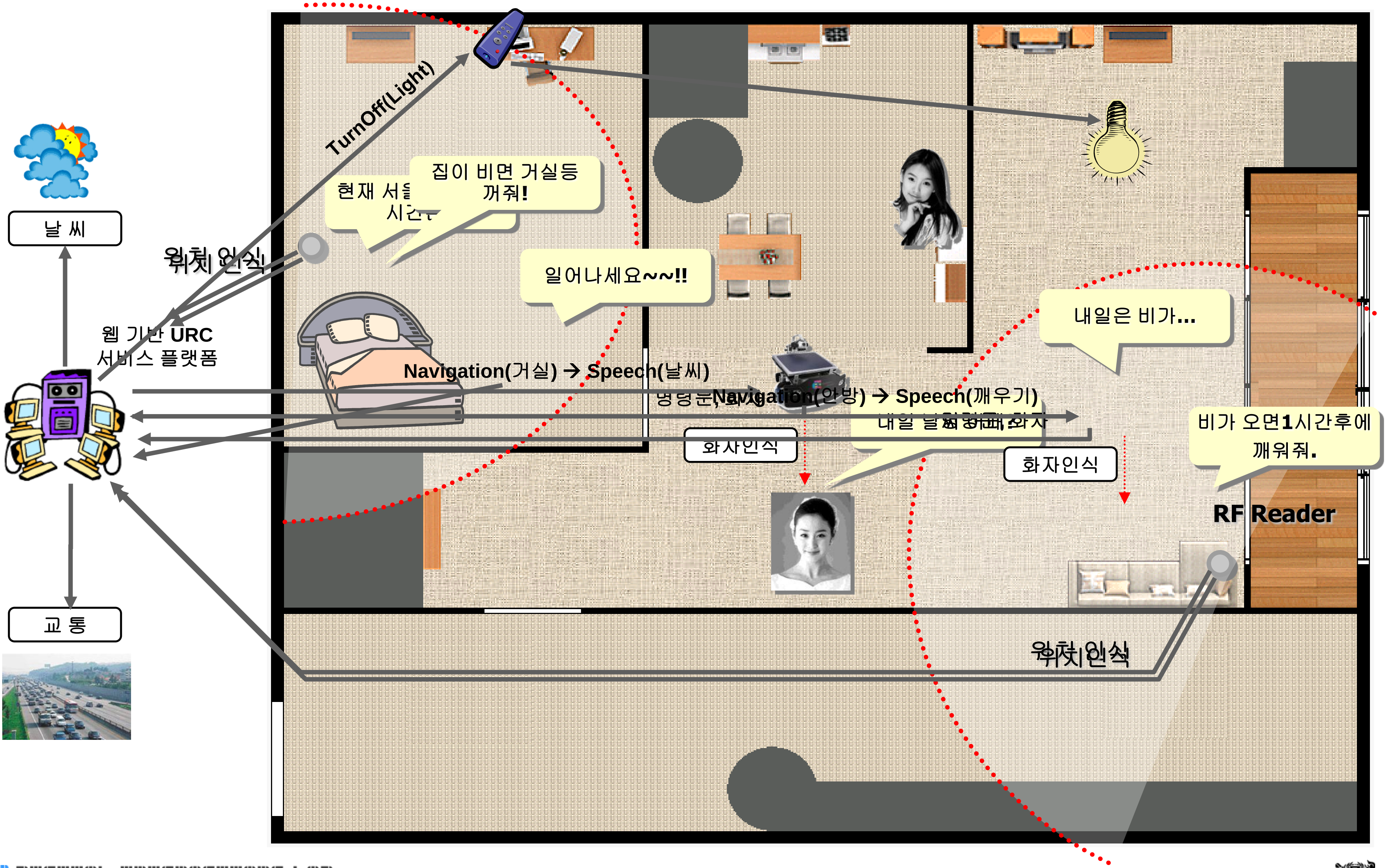
약품 성분, 복용법, 부작용 정보 이용

관련 정보의 동영상 정보 이용

비디오에 대한 예고편 감상

음반의 주요 Music Video 감상

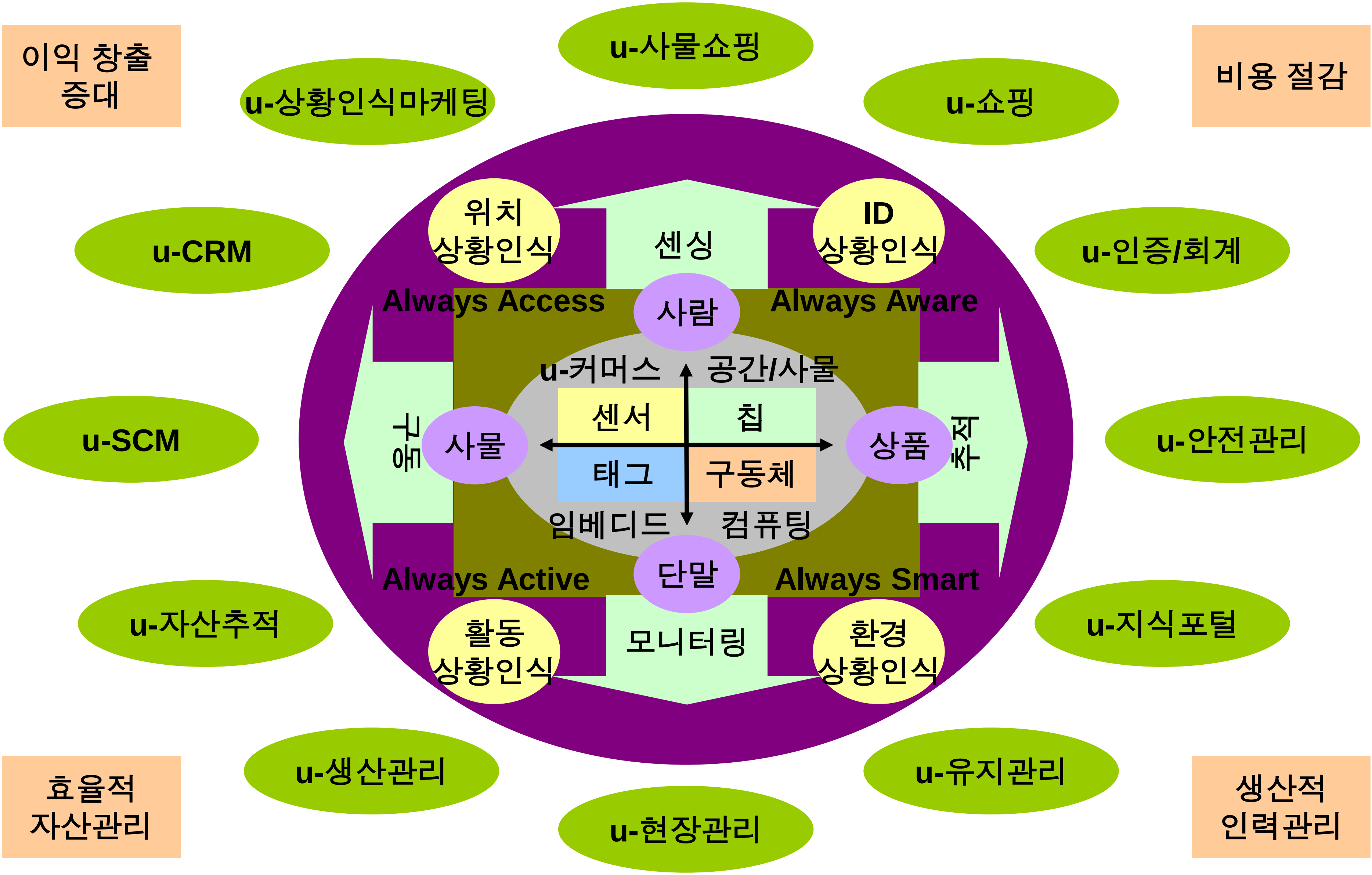
유비쿼터스 홈의 예
(출처: ETRI PEC, 2005)



uIT 기반의 기업 정보화 전망

- 새로운 기술은 경쟁에서 우위를 가지기 위해 사용되며 새로운 제품과 서비스에 적용됨
- uIT 기반의 기업 정보화는 현실 세계에서 사실 기반의 실시간 정보를 제공함으로써, 오류를 방지하고, 중요한 정보의 가시화, 정보 수집 및 전달의 실시간화가 가능
- 실시간 데이터에 기반한 효과적 의사 결정 및 기업 자원 관리, 가치 사슬 관리, 프로세스 최적화가 가능
- 일상생활의 사물, 어플라이언스, 상품, 기업의 생산, 물류, 판매, 고객관리 등의 비즈니스 프로세스를 구성하는 기기나 시스템들이 모두 지능화되고 네트워크로 연결되어 센스와 반응의 실시간화되는 RTE의 구현이 가능
- M2M2 장치들은 사람과 사물 사이에서 상호작용을 하며 부가가치를 창출해 낼 수 있음
- 새로운 개념의 "스마트 서비스" 및 새로운 개념의 상거래가 도래
- Ubiquitous connectivity를 통한 고객 서비스의 향상, 새로운 사업 기회의 창출이 가능

uIT 기반 기업정보화 시스템



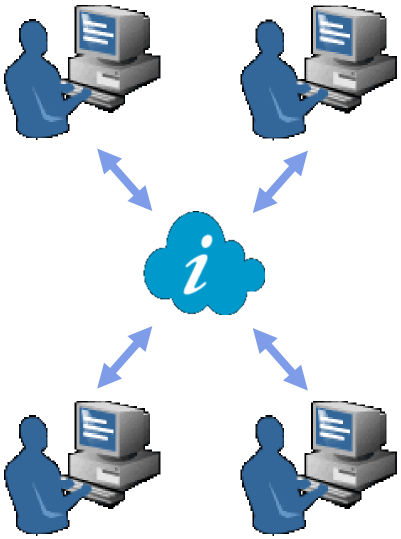
Transactions in ubiquitous environment

Data Processing

- Weeks
- Batch
- Megabytes
- Punch Cards
- Few People

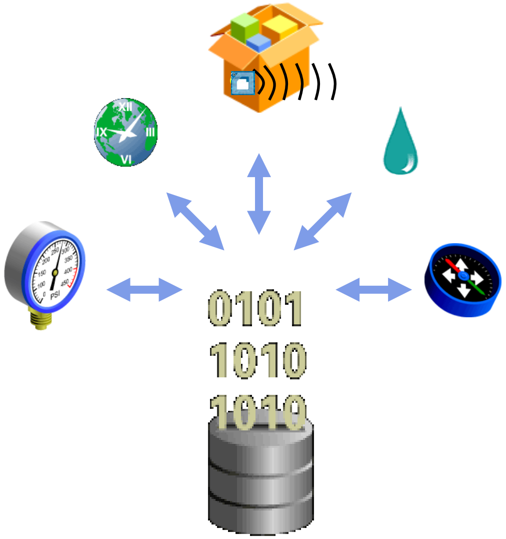
Internet

(Still Happening)



- Days
- Request/Reply
- Terabytes
- Human
- Many People

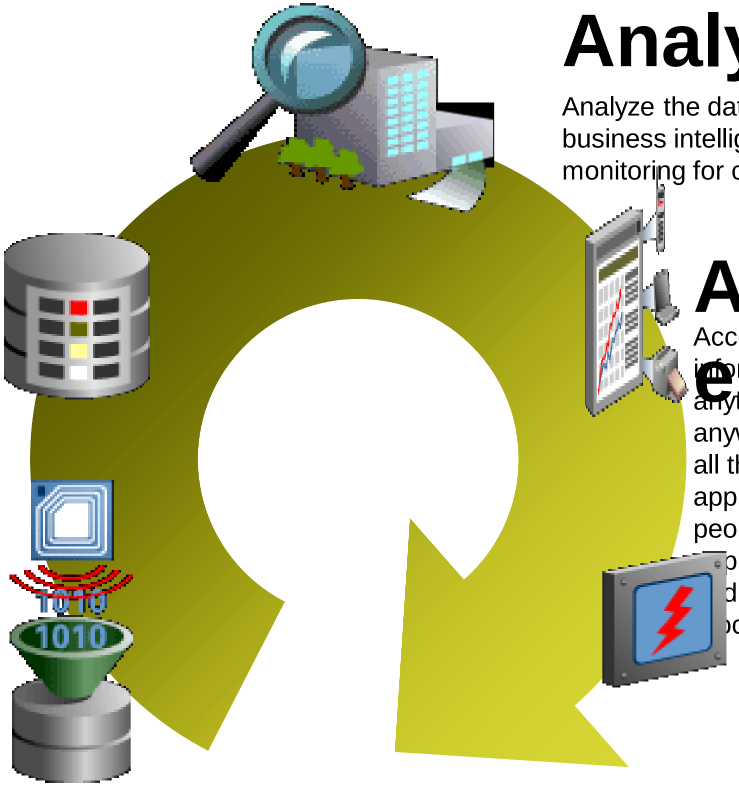
Real Time



- Minutes
- Automated
- Exabytes
- Event Driven
- Beyond People

RFID Middleware Challenges

(Source: Oracle, 2004)



Analyze

Analyze the data and events in real-time to provide business intelligence and business activity monitoring for continuous process improvement.

Manage

Manage the explosion of data and events in a scalable, reliable and secure single source of truth.

Access

Access the information anytime, anywhere by all the appropriate people, applications and business processes.

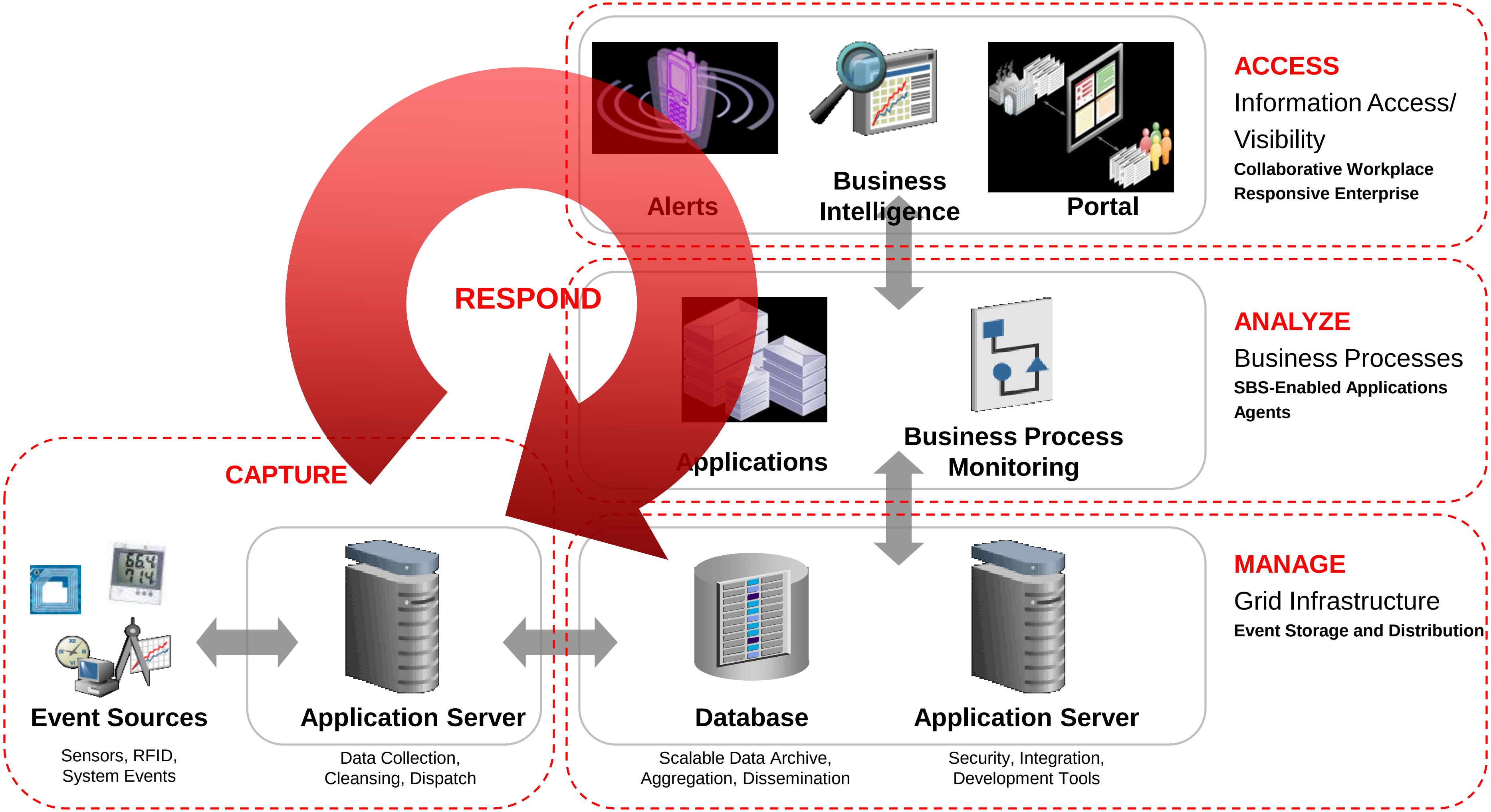
Capture

Capture appropriate, filtered information from a variety of different readers and sensors.

Respond

Respond to events and information automatically and allow for people to manage by exception.

Oracle's sensor-based services



ACCESS

Information Access/ Visibility

Collaborative Workplace

Responsive Enterprise

Alerts

Business Intelligence

Portal

RESPOND

ANALYZE

Business Processes

SBS-Enabled Applications

Agents

Applications

Business Process Monitoring

CAPTURE

MANAGE

Grid Infrastructure

Event Storage and Distribution

Event Sources

Sensors, RFID, System Events

Application Server

Data Collection, Cleansing, Dispatch

Database

Scalable Data Archive, Aggregation, Dissemination

Application Server

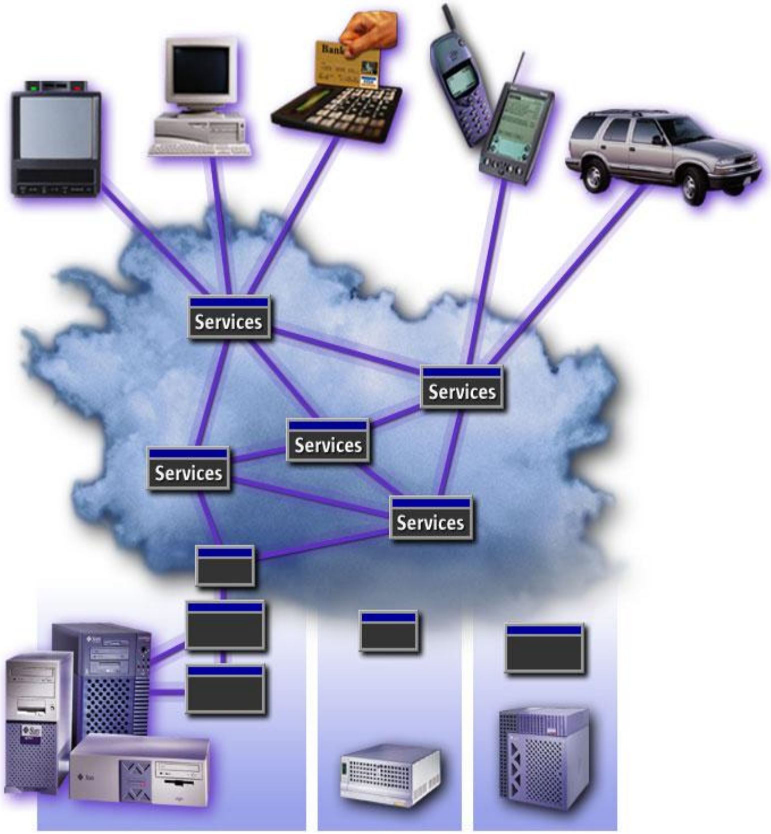
Security, Integration, Development Tools

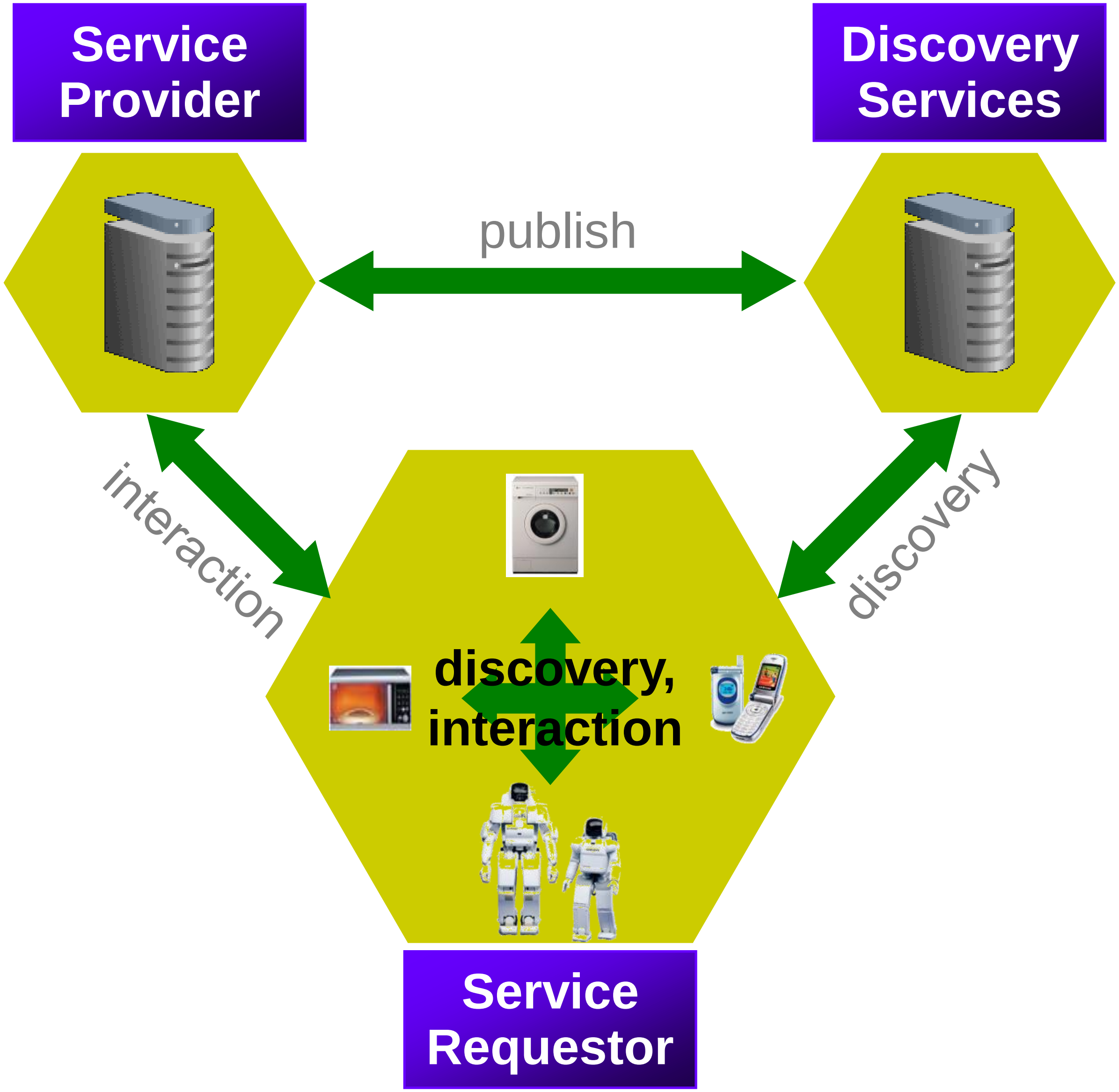
유비쿼터스 서비스 (u-Services)

- Ubiquitous computing
  - 사물들에 칩, 센서, RFID 태그 등을 심거나 부착함으로써 컴퓨팅 능력과 통신 능력을 부여함
- Ubiquitous networks
  - 유비쿼터스 컴퓨팅 능력을 갖는 개체들을 브로드밴드 네트워크, 위성, 모바일 네트워크, 무선랜 등을 통해 연결함
- Ubiquitous services
  - 유비쿼터스 컴퓨팅과 유비쿼터스 네트워크 기술을 기반으로 서비스와 콘텐츠가 이음새 없이 연계되고 통합됨으로써 새로운 가치를 창출
  - 네트워크에 연결된 모든 가치있는 것으로부터 서비스 창출

유비쿼터스 웹 서비스 (UWS)

- 어떠한 단말 / 네트워크 환경에서도 다양한 응용 서비스를 연계 / 융합 / 이용할 수 있도록 하는 웹 서비스 기술
- 웹 서비스 기술은 비즈니스 분야 뿐만 아니라 점차 광대역통합망(BcN)의 개방형 API 기반 유무선 통합 응용, 방송 / 통신 융합, 정보 가전, 텔레매틱스, 지능형 로봇, 임베디드 환경 등 다양한 분야에서 핵심 기술로 활용되어가고 있음
- Ubiquitous availability of services: Any time, Any where, Any devices, Any networks, Any services





Service Provider

Discovery Services

publish

interaction

discovery

discovery, interaction

Service Requestor

Interoperability
Device embedding
Performance
Service provision
Service consumption

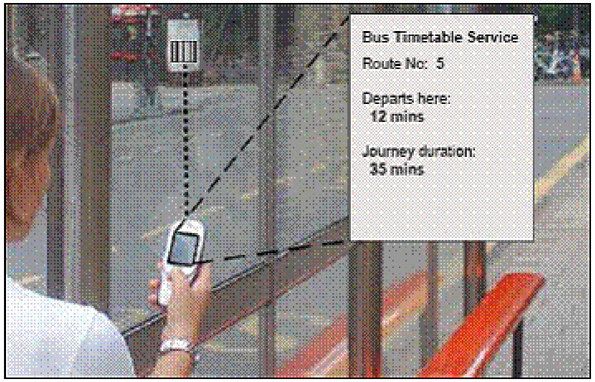
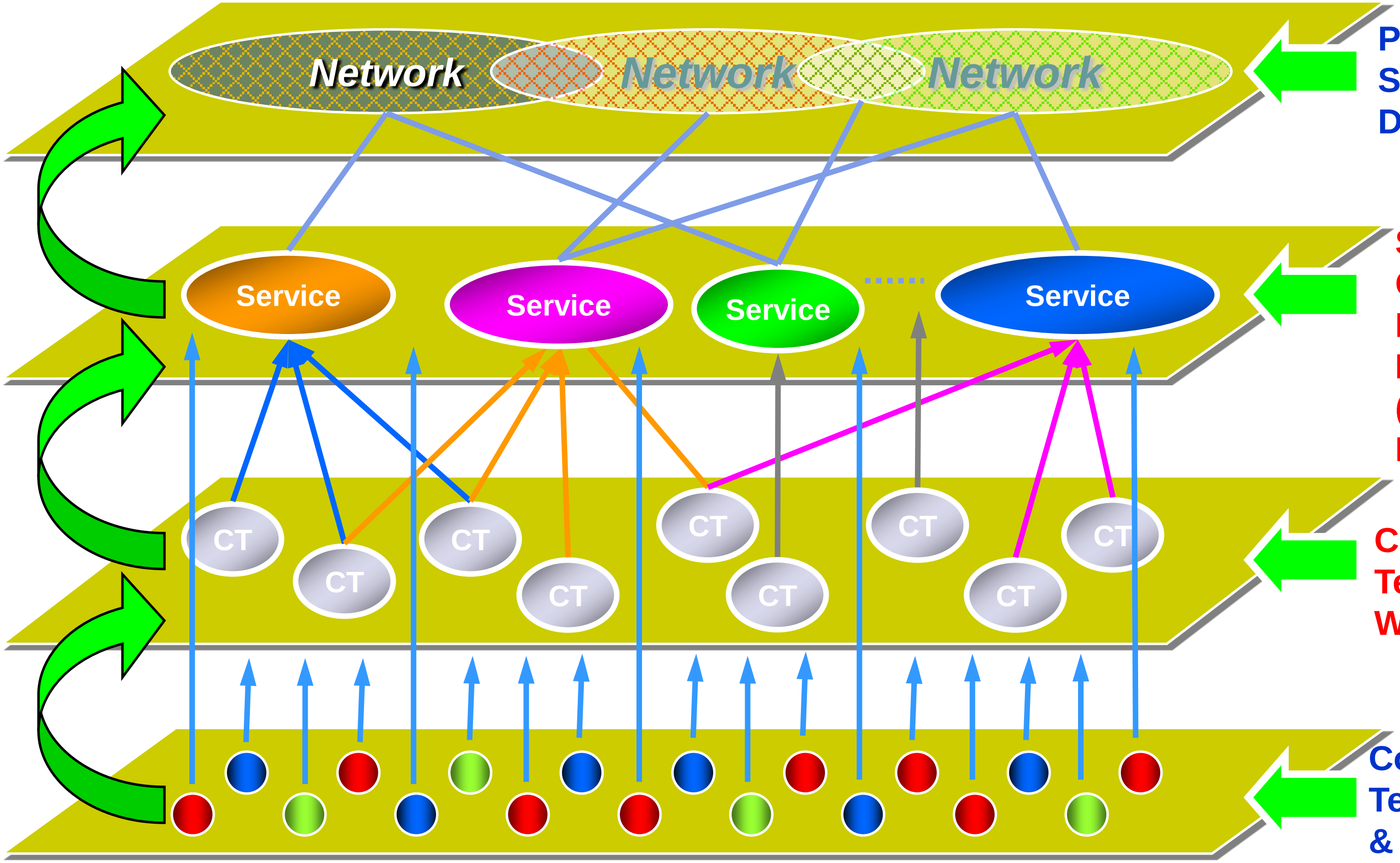


Figure 1 – A traveller accessing ubiquitous services at a bus stop

u-Services: a big picture

(출처: ETRI PEC, 2005)



Productive Service Networks Deployment

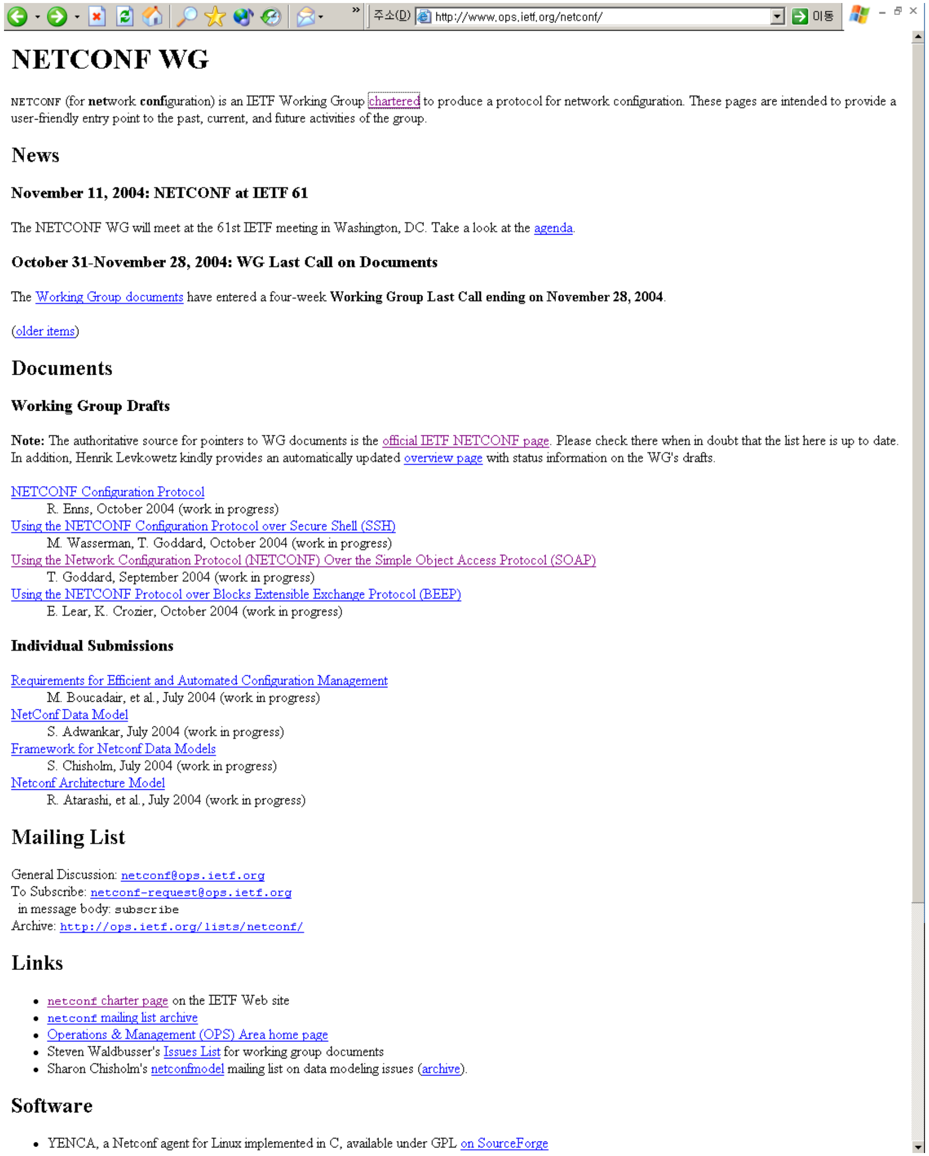
Service Composition & Differentiate Internet Services (New Business Model)

Convergence Technology with Web Services

Core Component Technology & Other Technology

NETCONF

- WG in IETF
- Chartered to produce a **protocol suitable for network configuration**
- draft-ietf-netconf-soap-03 (Sep, 2004): Using the Network Configuration Protocol (NETCONF) Over the Simple Object Access Protocol (SOAP)
  - implementing NETCONF protocol as a SOAP-based web service



NETCONF WG

NETCONF (for **net**work **conf**iguration) is an IETF Working Group chartered to produce a protocol for network configuration. These pages are intended to provide a user-friendly entry point to the past, current, and future activities of the group.

News

November 11, 2004: NETCONF at IETF 61

The NETCONF WG will meet at the 61st IETF meeting in Washington, DC. Take a look at the agenda.

October 31-November 28, 2004: WG Last Call on Documents

The Working Group documents have entered a four-week **Working Group Last Call ending on November 28, 2004**.

(older items)

Documents

Working Group Drafts

Note: The authoritative source for pointers to WG documents is the official IETF NETCONF page. Please check there when in doubt that the list here is up to date. In addition, Henrik Levkowetz kindly provides an automatically updated overview page with status information on the WG's drafts.

NETCONF Configuration Protocol
R. Enns, October 2004 (work in progress)
Using the NETCONF Configuration Protocol over Secure Shell (SSH)
M. Wasserman, T. Goddard, October 2004 (work in progress)
Using the Network Configuration Protocol (NETCONF) Over the Simple Object Access Protocol (SOAP)
T. Goddard, September 2004 (work in progress)
Using the NETCONF Protocol over Blocks Extensible Exchange Protocol (BEEP)
E. Lear, K. Crozier, October 2004 (work in progress)

Individual Submissions

Requirements for Efficient and Automated Configuration Management
M. Boucadair, et al., July 2004 (work in progress)
NetConf Data Model
S. Adwankar, July 2004 (work in progress)
Framework for Netconf Data Models
S. Chisholm, July 2004 (work in progress)
Netconf Architecture Model
R. Atarashi, et al., July 2004 (work in progress)

Mailing List

General Discussion: netconf@ops.ietf.org
To Subscribe: netconf-request@ops.ietf.org
in message body: subscribe
Archive: http://ops.ietf.org/lists/netconf/

Links

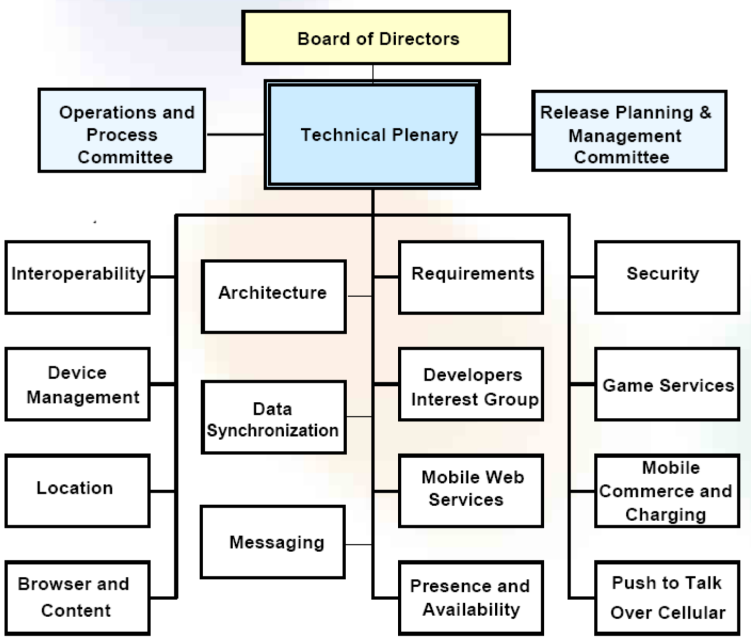
- netconf charter page on the IETF Web site
- netconf mailing list archive
- Operations & Management (OPS) Area home page
- Steven Waldbusser's Issues List for working group documents
- Sharon Chisholm's netconfmodel mailing list on data modeling issues (archive).

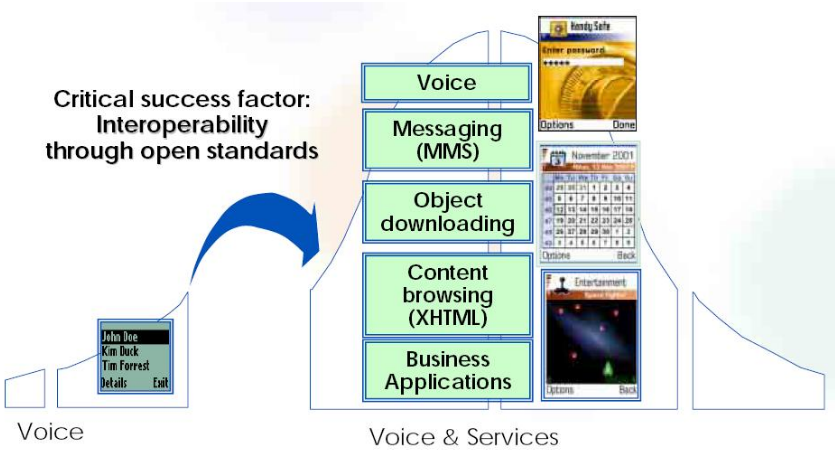
Software

- YENCA, a Netconf agent for Linux implemented in C, available under GPL on SourceForge

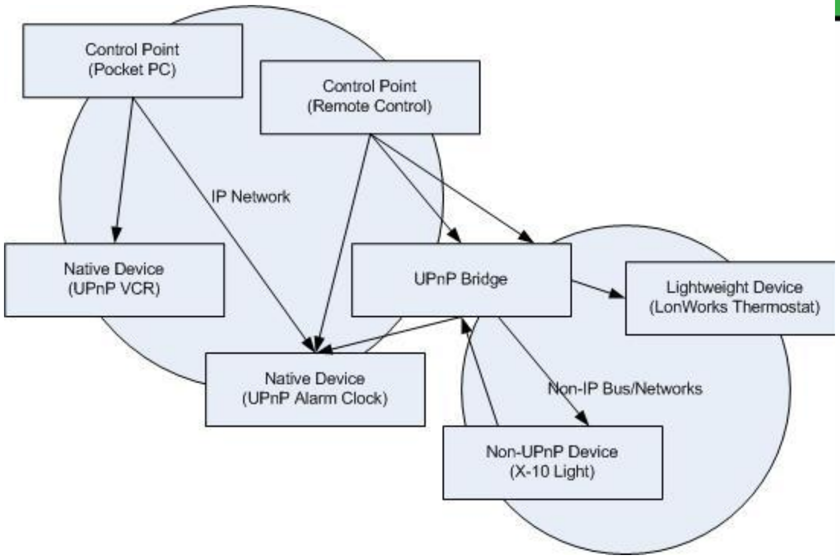
OMA's Mobile Web Services WG

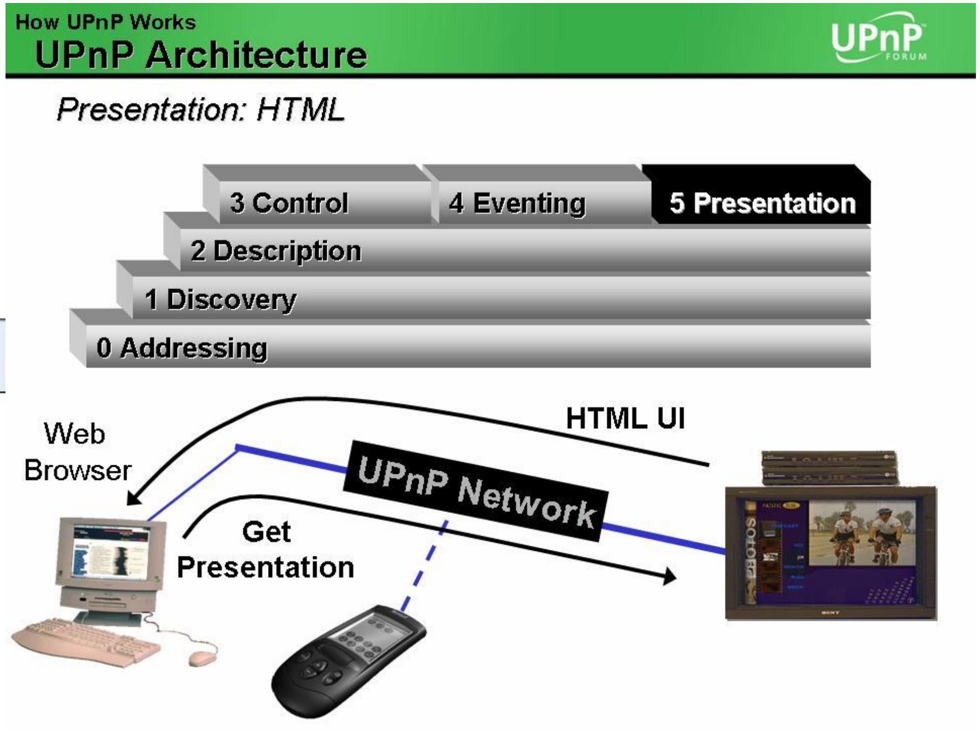
- To provide consistent, standard, federated access to service enablers that exist within or connected to the wireless network and devices
- 2004년 모바일 서비스 호환을 위한 OWSER (OMA Web Service Enabler)를 발표
- 현재 general analysis, network identity의 WS 응용 등에 관해서 작업 중





UPnP 2.0 (http://www.upnp.org/)





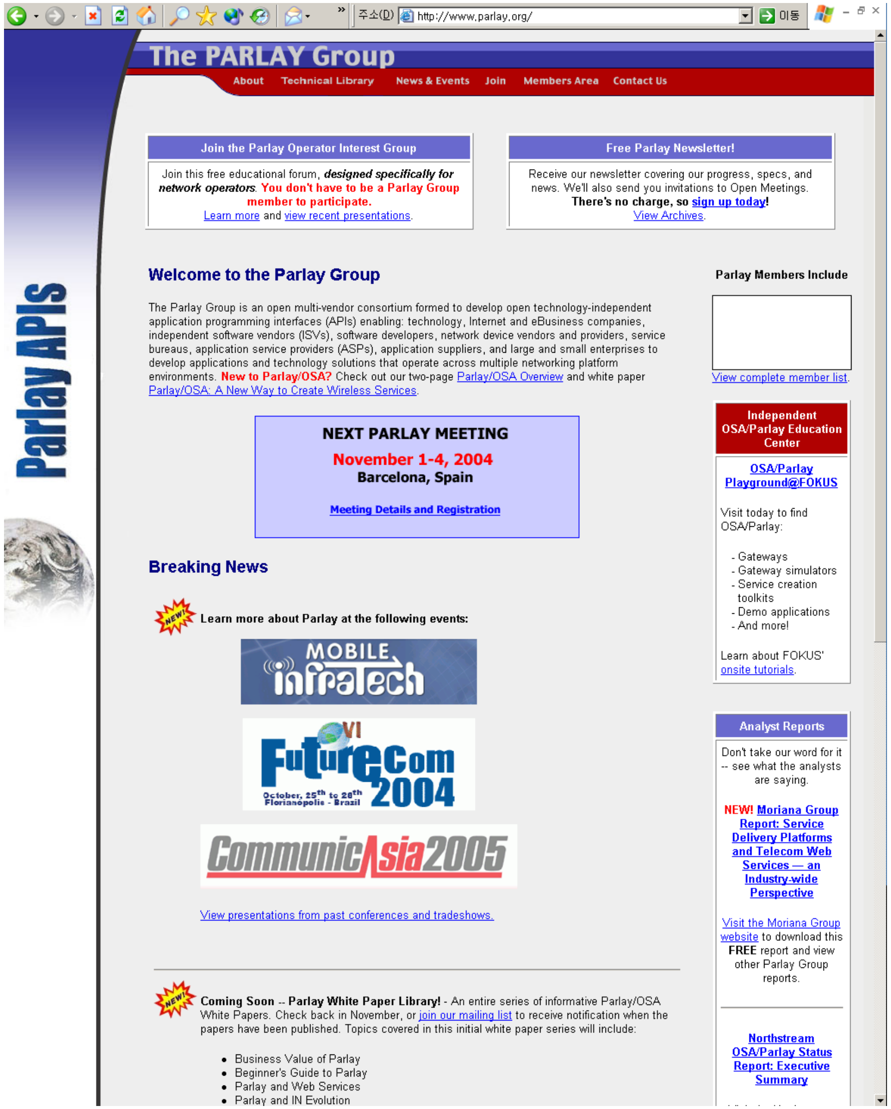
How UPnP Works

UPnP Architecture

Presentation: HTML

Parlay group: Parlay X web services

- 통신망 사업자나 서비스 사업자들이 유/무선망 등의 네트워크 하부구조에 독립적으로 통신서비스를 정의하고, 구현하기 위한 웹 서비스 기반 표준
- Intended to stimulate the development of next generation **network applications** by IT developers who are not necessarily experts in telecommunications



The PARLAY Group

About | Technical Library | News & Events | Join | Members Area | Contact Us

Join the Parlay Operator Interest Group

Join this free educational forum, ***designed specifically for network operators***. **You don't have to be a Parlay Group member to participate.** Learn more and view recent presentations.

Free Parlay Newsletter!

Receive our newsletter covering our progress, specs, and news. We'll also send you invitations to Open Meetings. **There's no charge, so sign up today!** View Archives.

Welcome to the Parlay Group

The Parlay Group is an open multi-vendor consortium formed to develop open technology-independent application programming interfaces (APIs) enabling: technology, Internet and eBusiness companies, independent software vendors (ISVs), software developers, network device vendors and providers, service bureaus, application service providers (ASPs), application suppliers, and large and small enterprises to develop applications and technology solutions that operate across multiple networking platform environments. **New to Parlay/OSA?** Check out our two-page Parlay/OSA Overview and white paper Parlay/OSA: A New Way to Create Wireless Services.

NEXT PARLAY MEETING

November 1-4, 2004

Barcelona, Spain

Meeting Details and Registration

Breaking News

Learn more about Parlay at the following events:

View presentations from past conferences and tradeshows.

Coming Soon -- Parlay White Paper Library! - An entire series of informative Parlay/OSA White Papers. Check back in November, or join our mailing list to receive notification when the papers have been published. Topics covered in this initial white paper series will include:

- Business Value of Parlay
- Beginner's Guide to Parlay
- Parlay and Web Services
- Parlay and IN Evolution

Parlay Members Include

View complete member list.

Independent OSA/Parlay Education Center

OSA/Parlay Playground@FOKUS

Visit today to find OSA/Parlay:

- Gateways
- Gateway simulators
- Service creation toolkits
- Demo applications
- And more!

Learn about FOKUS' onsite tutorials.

Analyst Reports

Don't take our word for it -- see what the analysts are saying.

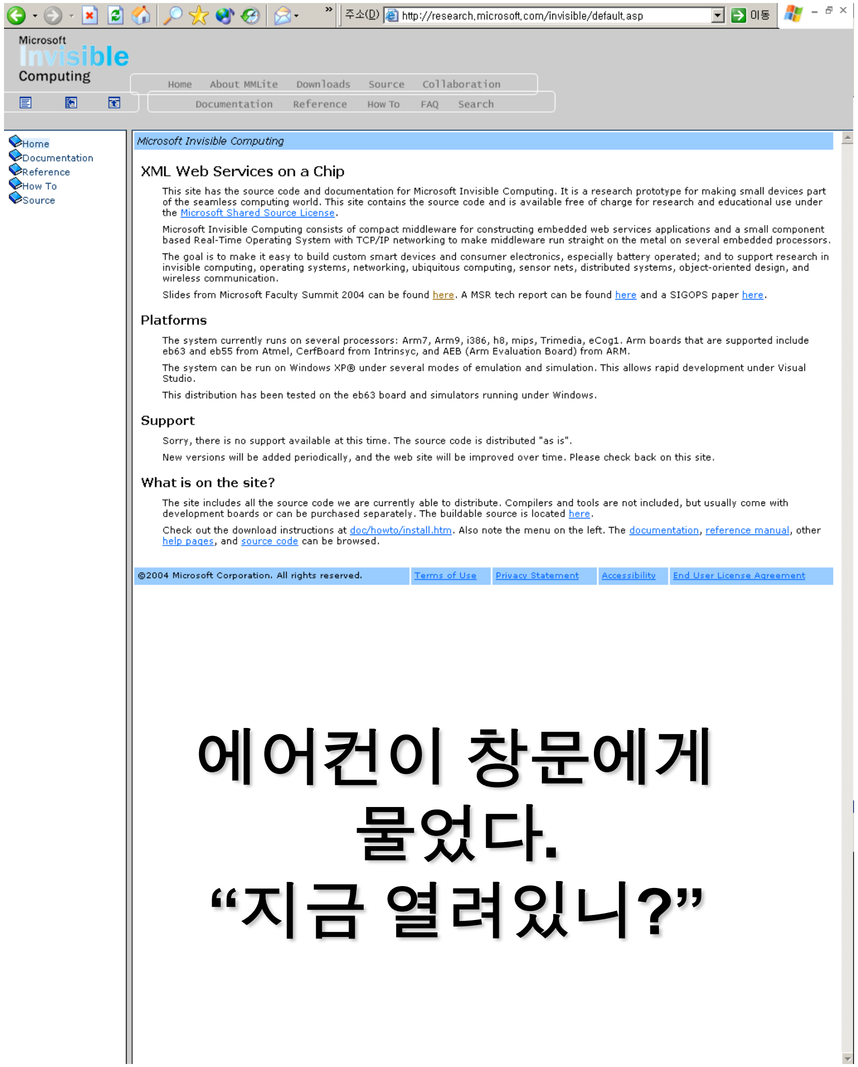
NEW! Moriana Group Report: Service Delivery Platforms and Telecom Web Services — an Industry-wide Perspective

Visit the Moriana Group website to download this **FREE** report and view other Parlay Group reports.

Northstream OSA/Parlay Status Report: Executive Summary

Microsoft's invisible computing

- A software platform for **low cost embedded systems** that communicate with each other and with big computers
  - XML **Web services**
  - Flexible development for **multiple platforms**
  - Interoperation with small and big computers
  - Security and privacy
  - Real-Time & Energy aware
  - Low parts cost (targeted for <= $5 computer)



XML Web Services on a Chip

This site has the source code and documentation for Microsoft Invisible Computing. It is a research prototype for making small devices part of the seamless computing world. This site contains the source code and is available free of charge for research and educational use under the Microsoft Shared Source License.

Microsoft Invisible Computing consists of compact middleware for constructing embedded web services applications and a small component based Real-Time Operating System with TCP/IP networking to make middleware run straight on the metal on several embedded processors.

The goal is to make it easy to build custom smart devices and consumer electronics, especially battery operated; and to support research in invisible computing, operating systems, networking, ubiquitous computing, sensor nets, distributed systems, object-oriented design, and wireless communication.

Slides from Microsoft Faculty Summit 2004 can be found here. A MSR tech report can be found here and a SIGOPS paper here.

Platforms

The system currently runs on several processors: Arm7, Arm9, i386, h8, mips, Trimedia, eCog1. Arm boards that are supported include eb63 and eb55 from Atmel, CerfBoard from Intrinsyc, and AEB (Arm Evaluation Board) from ARM.

The system can be run on Windows XP® under several modes of emulation and simulation. This allows rapid development under Visual Studio.

This distribution has been tested on the eb63 board and simulators running under Windows.

Support

Sorry, there is no support available at this time. The source code is distributed "as is".

New versions will be added periodically, and the web site will be improved over time. Please check back on this site.

What is on the site?

The site includes all the source code we are currently able to distribute. Compilers and tools are not included, but usually come with development boards or can be purchased separately. The buildable source is located here.

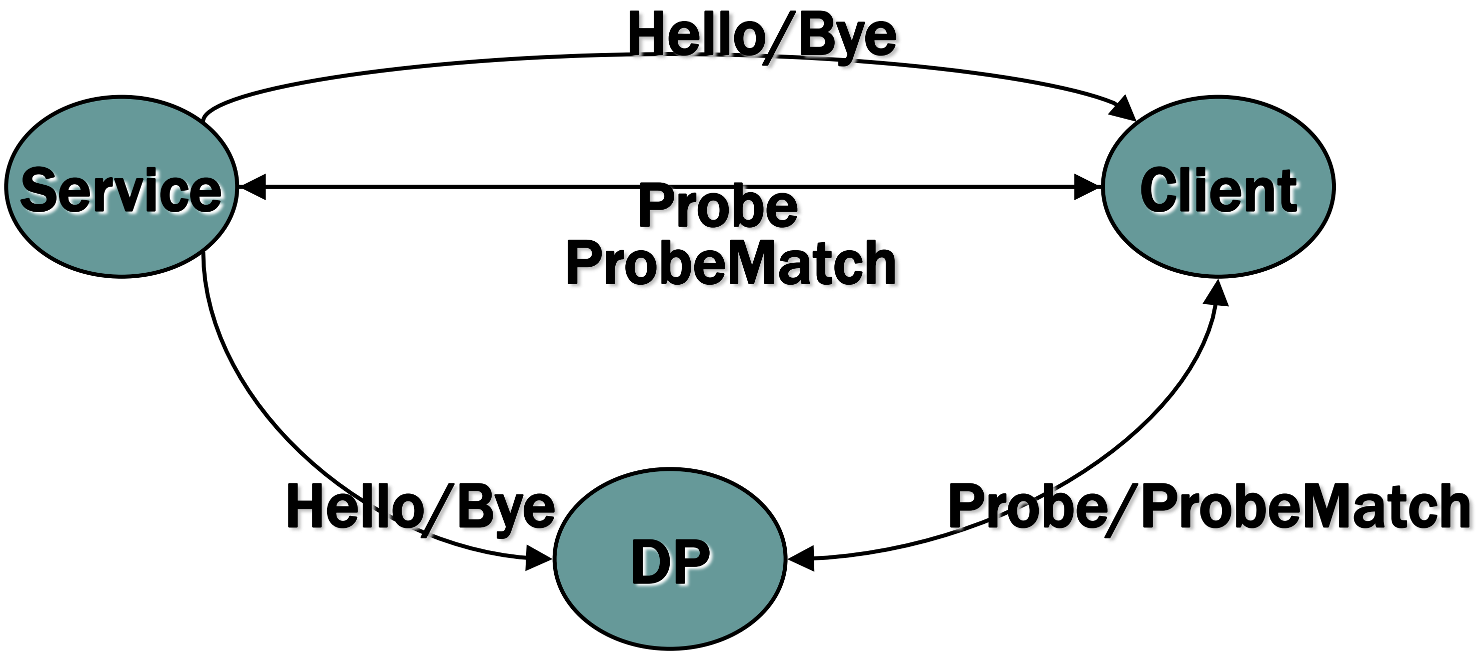
Check out the download instructions at doc/howto/install.htm. Also note the menu on the left. The documentation, reference manual, other help pages, and source code can be browsed.

©2004 Microsoft Corporation. All rights reserved.
Terms of Use
Privacy Statement
Accessibility
End User License Agreement

에어컨이 창문에게
물었다.
"지금 열려있니?"

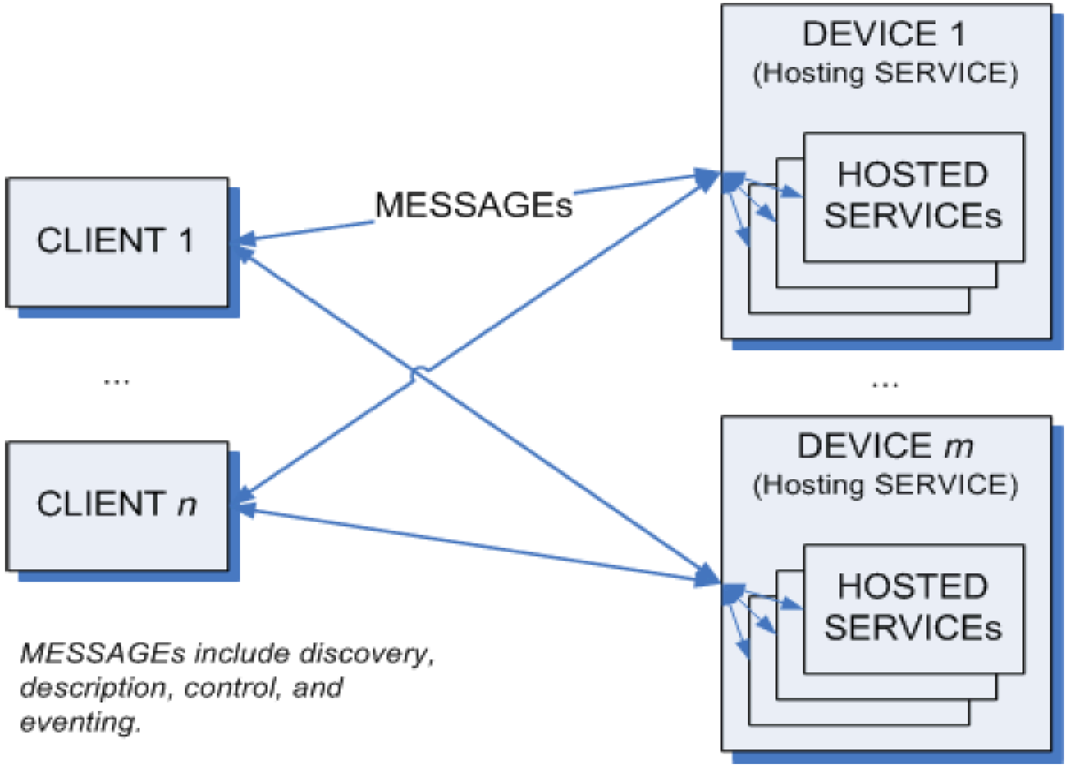
WS Dynamic Discovery

- 웹서비스 레지스트리가 없이 필요한 서비스 검색
- MS에서 WS-Discovery 스펙 개발 중
- WS-Discovery
  - Hello, Bye, Probe, Probe Match(PM), Resolve, Resolve Match(RM) 총 6개의 메시지 형태 사용



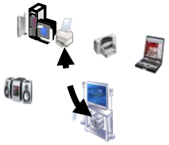
Device Profile for Web Services

- defines a minimal set of implementation constraints to enable secure Web service messaging, discovery, description, and eventing on resource-constrained endpoints

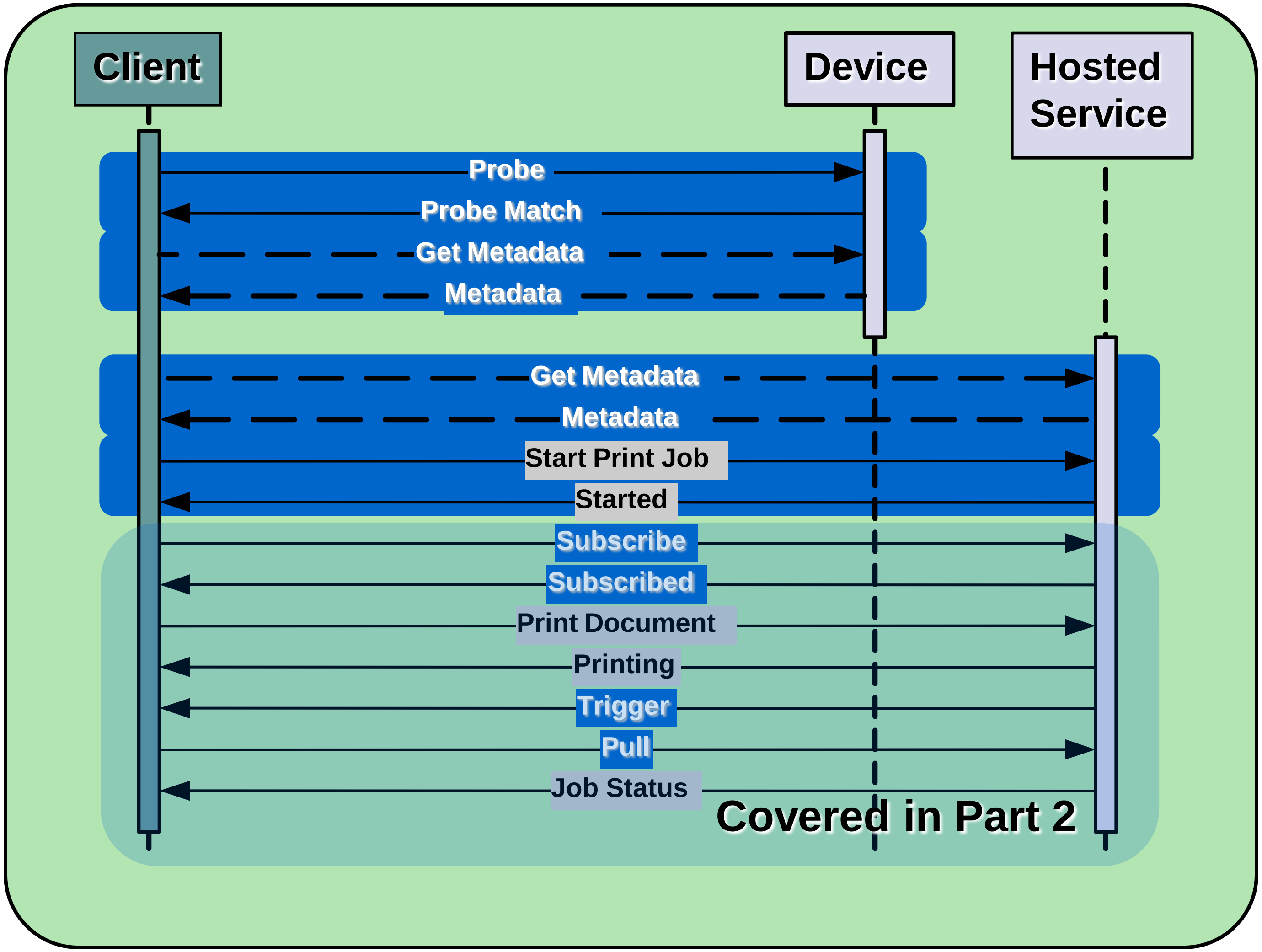


MESSAGEs include discovery, description, control, and eventing.

Example

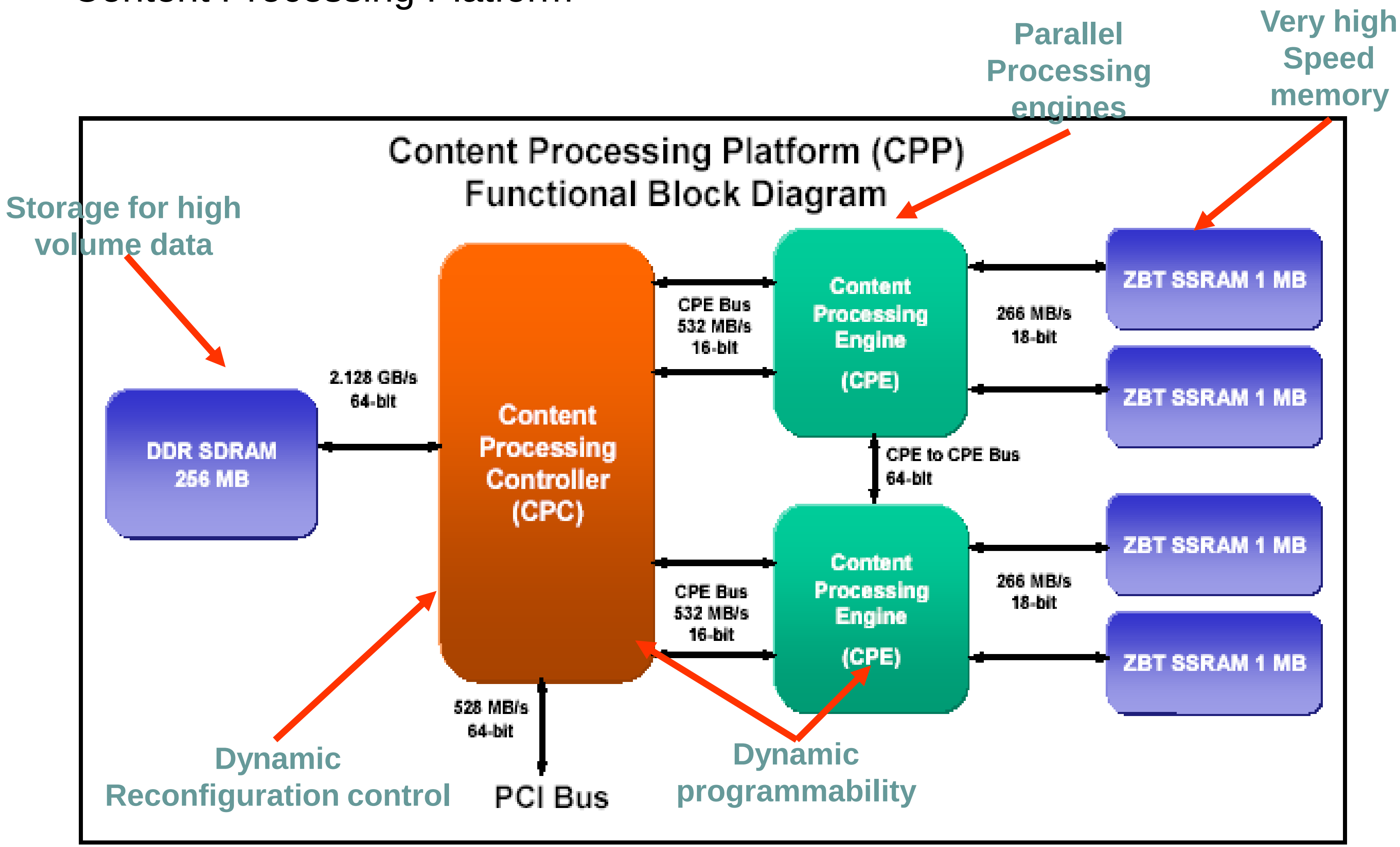


Message Flow, A.K.A., Talk Agenda



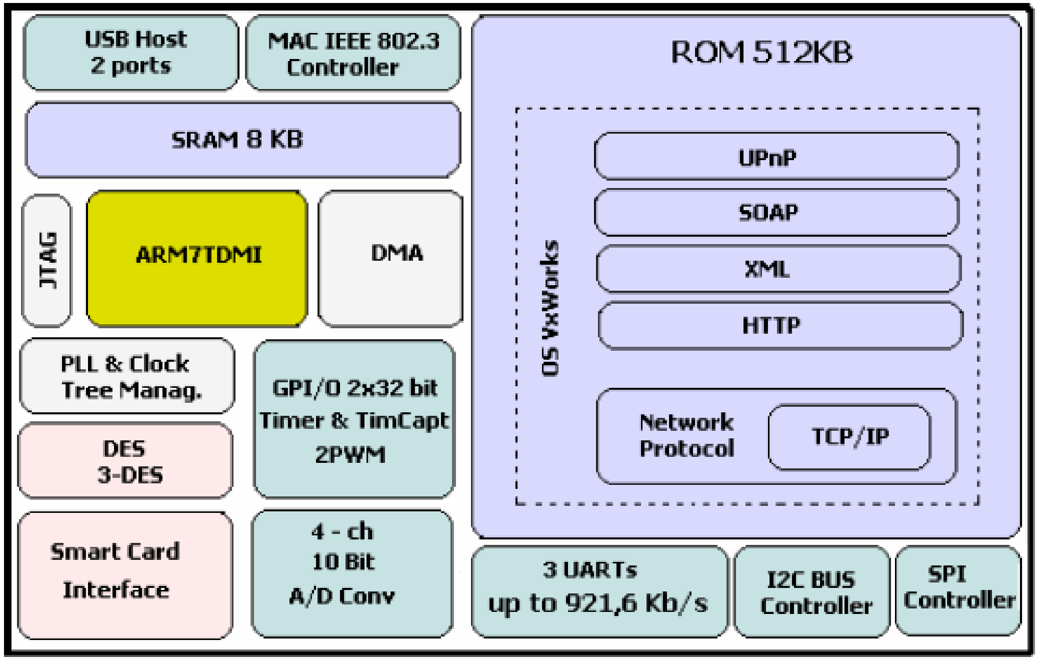
XML Processor

XML Content Processor (Tarari)
Content Processing Platform



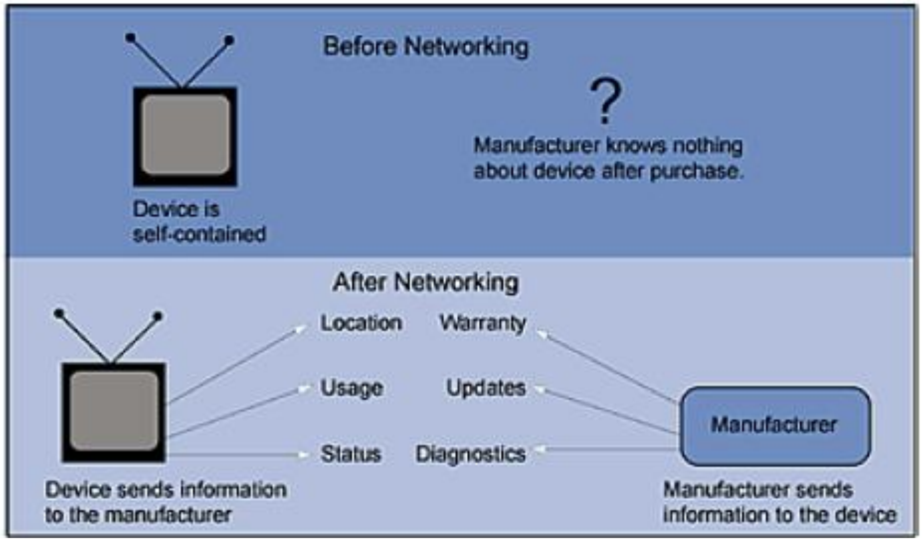
Web Services for Ubiquitous Devices

- is called “iPC”
- Developed by Samsung and Thinkware
- Goal
    - How can a device be connected to the internet in a simple, fast and standard way ?
- All in the box

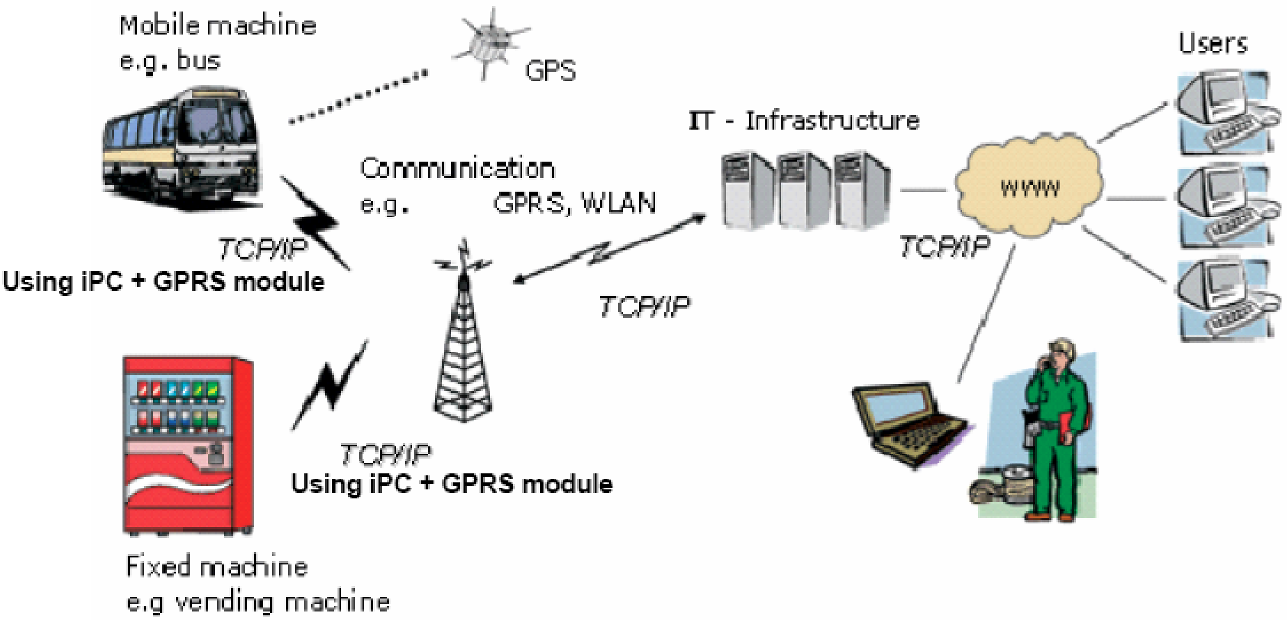


Web Services on a Single Chip

- Internet Connectivity

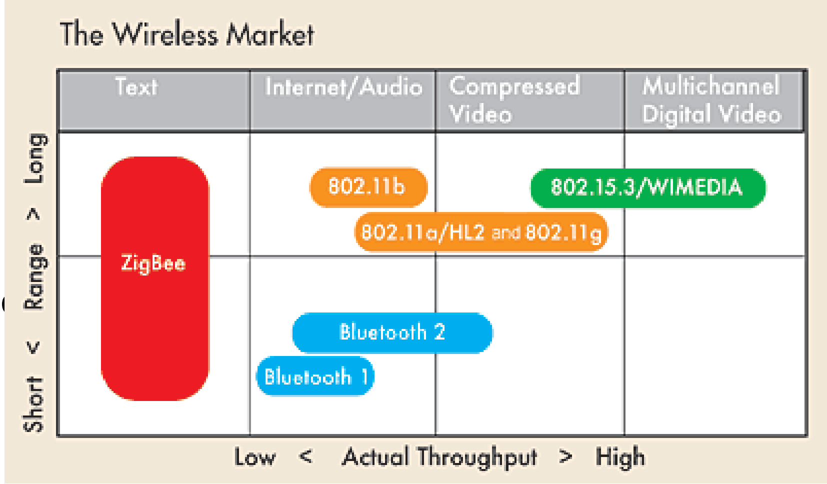

Before Networking
Device is self-contained
?
Manufacturer knows nothing about device after purchase.
After Networking
Location
Warranty
Usage
Updates
Status
Diagnostics
Manufacturer
Device sends information to the manufacturer
Manufacturer sends information to the device

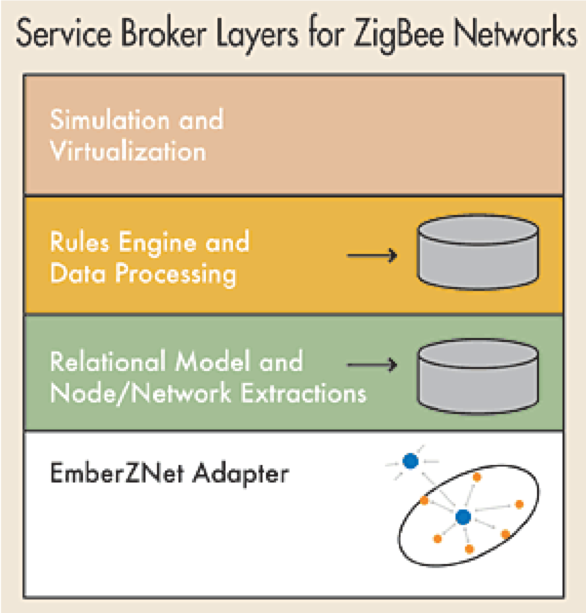
- ‘iPC’ Applications



ZigBee

- Developers of large sensor networks face investing dozens of man-years into developing common foundational software services that are unrelated to the application they seek to build
- Developers can tap Web service "brokers" to provide the **network discovery, extraction, commissioning, configuration, management, security, event/rule logi** and **data management** functions for large, diverse ZigBee systems
- Tendril Networks has developed service brokers that work with Ember's ZigBee-based wireless nodes



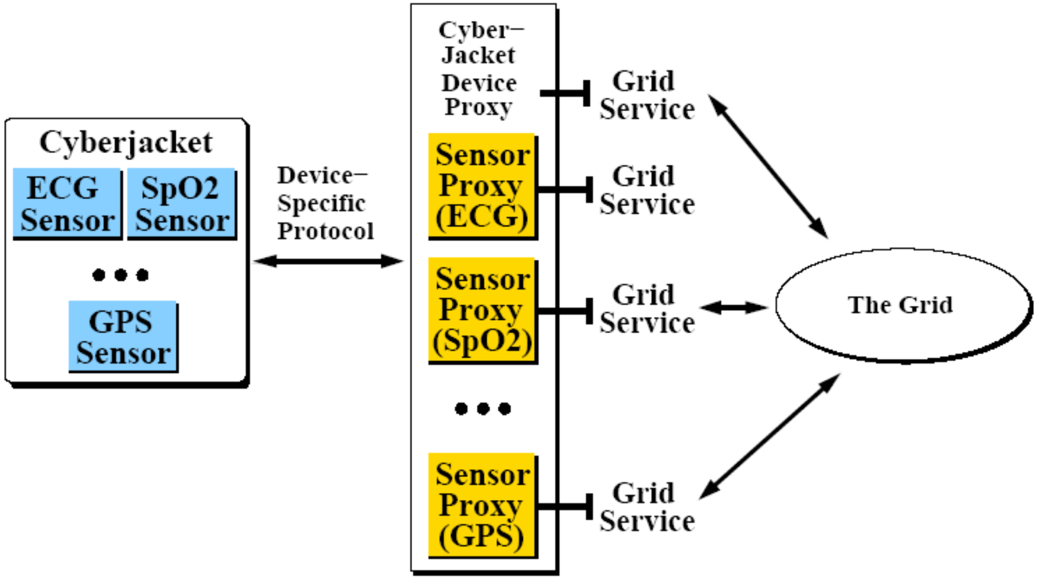

Service Broker Layers for ZigBee Networks
Simulation and Virtualization
Rules Engine and Data Processing
Relational Model and Node/Network Extractions
EmberZNet Adapter

RFID middleware

	EPC Network 미들웨어(Savant)	Microsoft 미들웨어 (2005년 출시 예정)	ETRI 자동식별 미들웨어
태그 데이터	❖ 식별 코드(EPC)	❖ 식별 코드, 이력 정보	❖ 식별 코드, 이력 정보
지원 기기	❖ 수동형 RFID 리더	❖ 수동/능동형 RFID 리더 ❖ 바코드 리더	❖ 수동/능동형 RFID 리더 ❖ 바코드 리더
표준 준수	❖ EPCglobal 표준	❖ EPC, 웹서비스 표준	❖ EPC, ISO, 웹서비스 표준
데이타 모니터링,관리	❖ 데이타 필터링, 수집, 요약 ❖ 테스크 관리(스케줄링)	❖ 데이타 필터링, 수집, 요약 ❖ 테스크 관리(스케줄링)	❖ 데이타 필터링, 수집, 요약 ❖ 테스크 관리(스케줄링)
Legacy 시스템 통합	❖ 웹서비스	❖ 웹서비스 ❖ 프로세스 자동화(BPEL)	❖ 웹서비스(내부시스템 통합) ❖ ebXML(외부시스템 통합)
검색 서비스	❖ EPC 기반 ONS 연동 ❖ IPV4 연동	❖UDDI, Active Directory	❖ MDS 연동(멀티 코드 지원) ❖ IPV4/IPV6 연동
전자태그 객체 정보 관리	❖ EPC IS ❖ 분산된 정적, 이력 데이타	❖ MS SQL Server ❖ 분산된 정적, 이력 데이타	❖ EPC IS 확장(센서 데이터 관리) ❖ 분산된 정적, 이력, 센서 데이타
개발 환경	❖ JAVA 플랫폼	❖ MS 윈도우 플랫폼	❖ JAVA 플랫폼, Open Source

Grid computing: The server side

- Ubicomp-RG (at GGF):
  focus on using Grid technologies both as a means to interconnect existing and emerging ubiquitous computing environments and as a core underlying technology for developing and deploying new ubiquitous computing systems
- OGSI and WS-RF



W3C’s Ubiquitous Web WG

- seeks to broaden the capabilities of browsers to enable **new kinds of web applications**, particularly those involving **coordination with other devices**
- Some examples include connecting a camera phone to a nearby printer, using a cell phone to give a business presentation with a wireless projector, and viewing your mailbox while listening to your messages
- These applications involve **identifying resources** and **managing** them within the context of an application session
- The resources can be **remote** as in a network printer and projector, or **local**, as in the estimated battery life, network signal strength, and audio volume level
- Ubiquitous Web will provide a framework for exposing **device coordination** capabilities to Web applications

Requirements & enabling techs for ubiquitous web

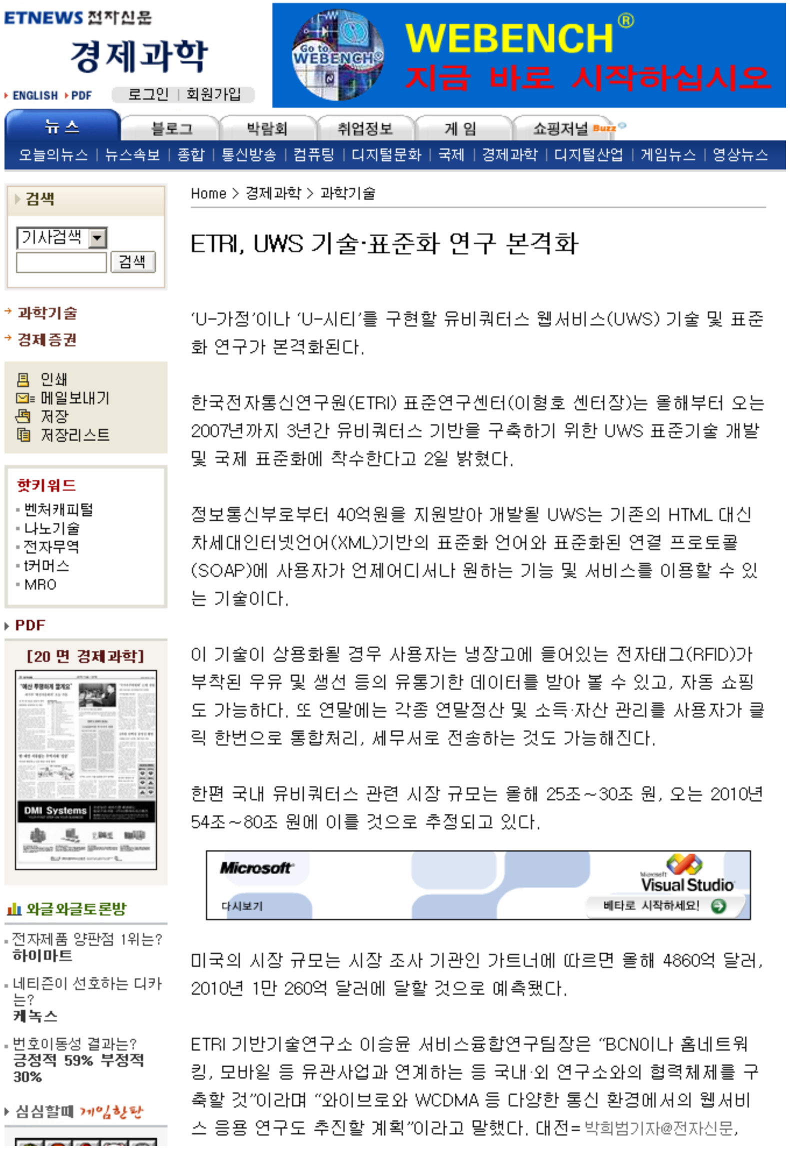
- Requirements
  - Dynamically adapt to user preferences, device capabilities and environmental conditions
  - Extend device capabilities through access to resources available via the network
  - Respond to events over the network from servers and other devices
  - Enable applications involving multiple devices
  - Use events to coordinate voice and data to augment human to human conversations
  - Manage resources in terms of temporary and persistent sessions

- Enabling technologies
  - IDL for describing interfaces for distributed systems and as used for the W3C DOM
  - URIs for naming resources, sessions and interfaces
  - Semantic Web for ontologies describing device capabilities
  - **Web Services** for passing commands and events
  - Existing device coordination mechanisms

기타 관련 사례

- Nokia
  - Nokia has vowed to build support for web services into all its smart phones by the end of the year
  - By 2006 all Nokia smartphones will be web services enabled
  - While Nokia is not planning to offer web services directly, it will support them and offer tools to help developers design software that could be used on smart phone
- Cisco
  - Cisco will launch products for handling XML traffic in June 2005 that will bring advanced XML security and management capabilities to large enterprise networks.
  - Cisco is using Tarari's programmable chip to perform low-level tasks such as checking XML signatures and verifying XML schemas

국내 동향



Home > 경제과학 > 과학기술

ETRI, UWS 기술·표준화 연구 본격화

'U-가정'이나 'U-시티'를 구현할 유비쿼터스 웹서비스(UWS) 기술 및 표준화 연구가 본격화된다.

한국전자통신연구원(ETRI) 표준연구센터(이형호 센터장)는 올해부터 오는 2007년까지 3년간 유비쿼터스 기반을 구축하기 위한 UWS 표준기술 개발 및 국제 표준화에 착수한다고 2일 밝혔다.

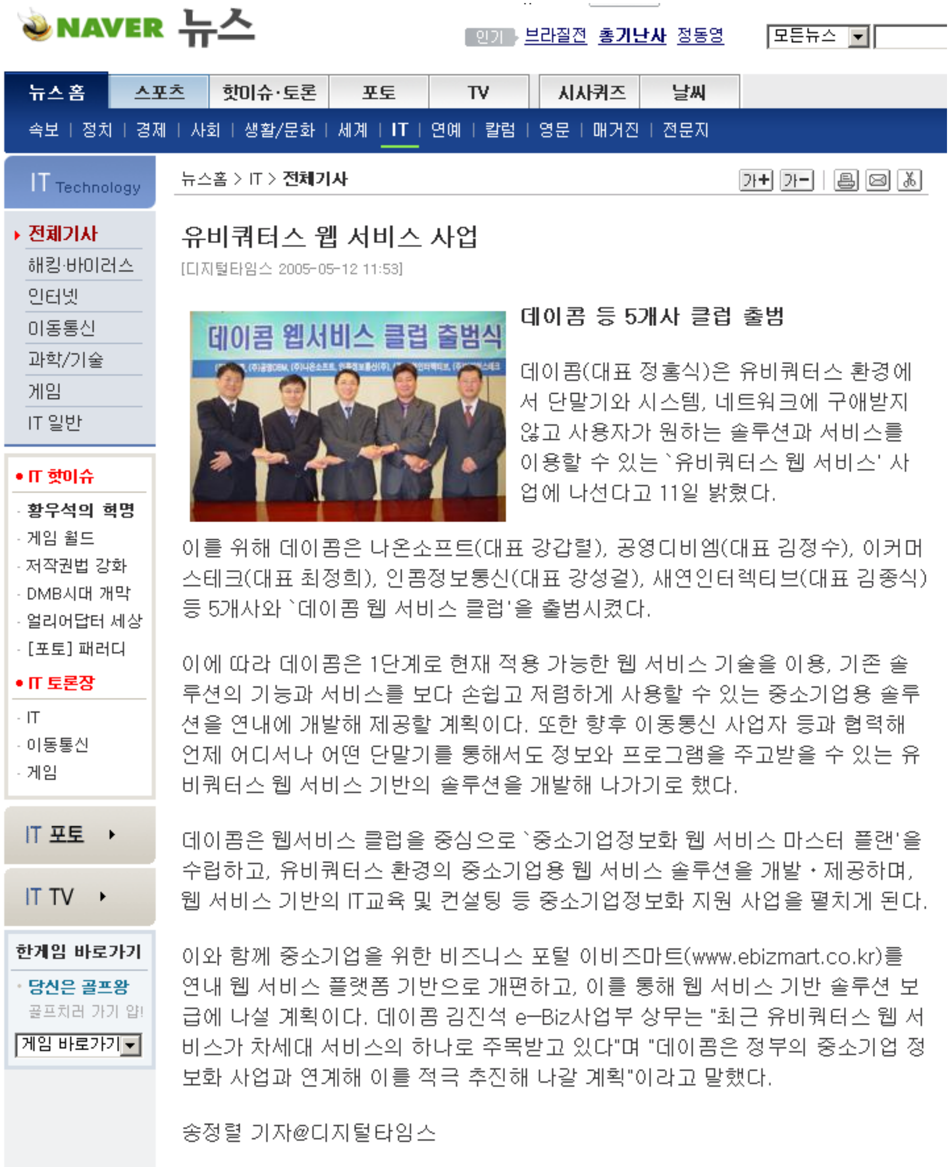
정보통신부로부터 40억원을 지원받아 개발될 UWS는 기존의 HTML 대신 차세대인터넷언어(XML)기반의 표준화 언어와 표준화된 연결 프로토콜(SOAP)에 사용자가 언제어디서나 원하는 기능 및 서비스를 이용할 수 있는 기술이다.

이 기술이 상용화될 경우 사용자는 냉장고에 들어있는 전자태그(RFID)가 부착된 우유 및 생선 등의 유통기한 데이터를 받아 볼 수 있고, 자동 쇼핑도 가능하다. 또 연말에는 각종 연말정산 및 소득·자산 관리를 사용자가 클릭 한번으로 통합처리, 세무서로 전송하는 것도 가능해진다.

한편 국내 유비쿼터스 관련 시장 규모는 올해 25조~30조 원, 오는 2010년 54조~80조 원에 이를 것으로 추정되고 있다.

미국의 시장 규모는 시장 조사 기관인 가트너에 따르면 올해 4860억 달러, 2010년 1만 260억 달러에 달할 것으로 예측됐다.

ETRI 기반기술연구소 이승윤 서비스융합연구팀장은 "BCN이나 홈네트워킹, 모바일 등 유관사업과 연계하는 등 국내·외 연구소와의 협력체제를 구축할 것"이라며 "와이브로와 WCDMA 등 다양한 통신 환경에서의 웹서비스 응용 연구도 추진할 계획"이라고 말했다. 대전=박희범기자@전자신문,



뉴스홈 > IT > 전체기사

유비쿼터스 웹 서비스 사업

[디지털타임스 2005-05-12 11:53]

데이콤 등 5개사 클럽 출범

데이콤(대표 정홍식)은 유비쿼터스 환경에서 단말기와 시스템, 네트워크에 구애받지 않고 사용자가 원하는 솔루션과 서비스를 이용할 수 있는 `유비쿼터스 웹 서비스' 사업에 나선다고 11일 밝혔다.

이를 위해 데이콤은 나온소프트(대표 강갑렬), 공영디비엠(대표 김정수), 이커머스테크(대표 최정희), 인콤정보통신(대표 강성걸), 새연인터렉티브(대표 김종식) 등 5개사와 `데이콤 웹 서비스 클럽'을 출범시켰다.

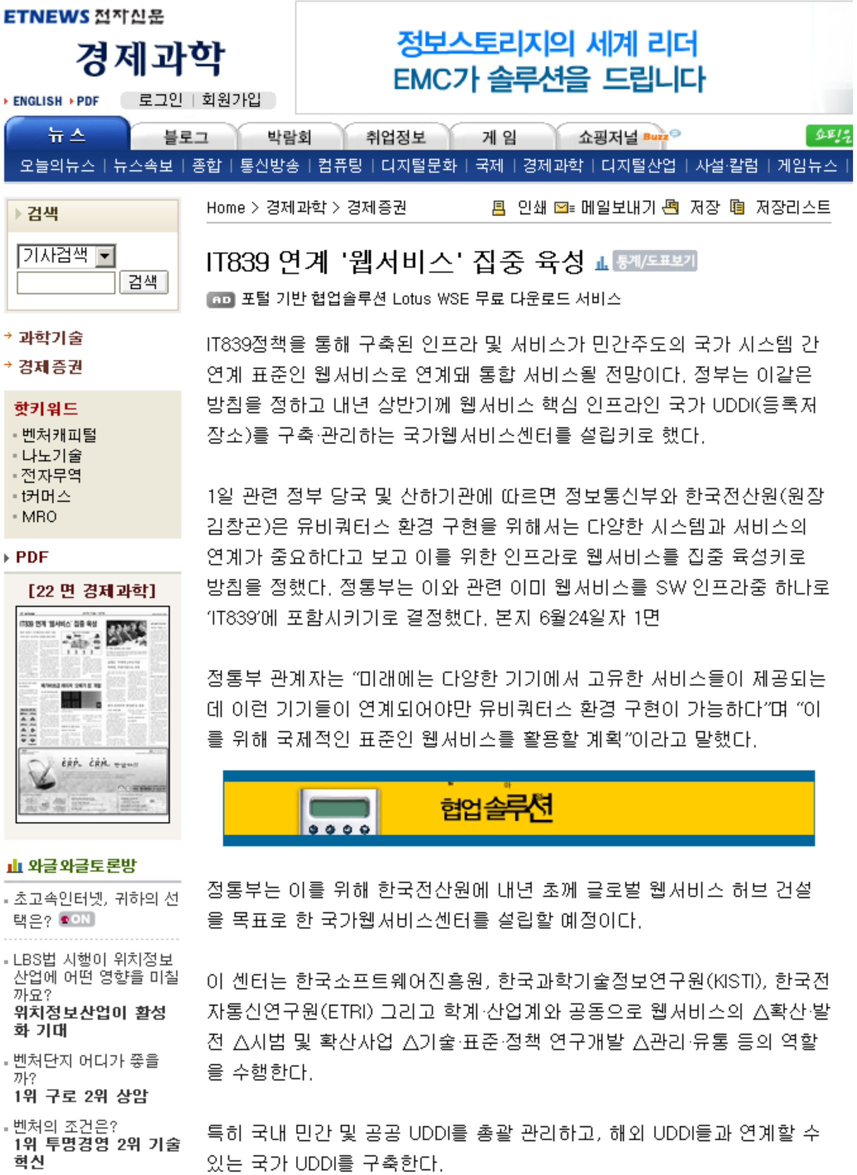
이에 따라 데이콤은 1단계로 현재 적용 가능한 웹 서비스 기술을 이용, 기존 솔루션의 기능과 서비스를 보다 손쉽고 저렴하게 사용할 수 있는 중소기업용 솔루션을 연내에 개발해 제공할 계획이다. 또한 향후 이동통신 사업자 등과 협력해 언제 어디서나 어떤 단말기를 통해서도 정보와 프로그램을 주고받을 수 있는 유비쿼터스 웹 서비스 기반의 솔루션을 개발해 나가기로 했다.

데이콤은 웹서비스 클럽을 중심으로 `중소기업정보화 웹 서비스 마스터 플랜'을 수립하고, 유비쿼터스 환경의 중소기업용 웹 서비스 솔루션을 개발·제공하며, 웹 서비스 기반의 IT교육 및 컨설팅 등 중소기업정보화 지원 사업을 펼치게 된다.

이와 함께 중소기업을 위한 비즈니스 포털 이비즈마트(www.ebizmart.co.kr)를 연내 웹 서비스 플랫폼 기반으로 개편하고, 이를 통해 웹 서비스 기반 솔루션 보급에 나설 계획이다. 데이콤 김진석 e-Biz사업부 상무는 "최근 유비쿼터스 웹 서비스가 차세대 서비스의 하나로 주목받고 있다"며 "데이콤은 정부의 중소기업 정보화 사업과 연계해 이를 적극 추진해 나갈 계획"이라고 말했다.

송정렬 기자@디지털타임스

국내 동향



ETNEWS 전자신문

경제과학

▸ENGLISH ▸PDF 로그인 | 회원가입

- 초고속인터넷, 귀하의 선택은? ON
- LBS법 시행이 위치정보 산업에 어떤 영향을 미칠까요?
  위치정보산업이 활성화 기대
- 벤처단지 어디가 좋을까?
  1위 구로 2위 상암
- 벤처의 조건은?
  1위 투명경영 2위 기술혁신

Home > 경제과학 > 경제증권 인쇄 메일보내기 저장 저장리스트

IT839 연계 '웹서비스' 집중 육성

IT839정책을 통해 구축된 인프라 및 서비스가 민간주도의 국가 시스템 간 연계 표준인 웹서비스로 연계돼 통합 서비스될 전망이다. 정부는 이같은 방침을 정하고 내년 상반기께 웹서비스 핵심 인프라인 국가 UDDI(등록저장소)를 구축·관리하는 국가웹서비스센터를 설립키로 했다.

1일 관련 정부 당국 및 산하기관에 따르면 정보통신부와 한국전산원(원장 김창곤)은 유비쿼터스 환경 구현을 위해서는 다양한 시스템과 서비스의 연계가 중요하다고 보고 이를 위한 인프라로 웹서비스를 집중 육성키로 방침을 정했다. 정통부는 이와 관련 이미 웹서비스를 SW 인프라중 하나로 'IT839'에 포함시키기로 결정했다. 본지 6월24일자 1면

정통부 관계자는 "미래에는 다양한 기기에서 고유한 서비스들이 제공되는데 이런 기기들이 연계되어야만 유비쿼터스 환경 구현이 가능하다"며 "이를 위해 국제적인 표준인 웹서비스를 활용할 계획"이라고 말했다.

정통부는 이를 위해 한국전산원에 내년 초께 글로벌 웹서비스 허브 건설을 목표로 한 국가웹서비스센터를 설립할 예정이다.

이 센터는 한국소프트웨어진흥원, 한국과학기술정보연구원(KISTI), 한국전자통신연구원(ETRI) 그리고 학계·산업계와 공동으로 웹서비스의 △확산·발전 △시범 및 확산사업 △기술·표준·정책 연구개발 △관리·유통 등의 역할을 수행한다.

특히 국내 민간 및 공공 UDDI를 총괄 관리하고, 해외 UDDI들과 연계할 수 있는 국가 UDDI를 구축한다.

The future of business services

Source: Fano and Gershman, CACM, 2002

- The location of your customer becomes the location of your business
- A physical point of presence wherever your products and services are used will become a competitive necessity
- Mobile devices and appliances become the eyes and ears of remote service providers
- Services we associate with locations become attached to people
- Services will use the customer’s location resources to provide the best possible service
- Service providers must pay continuous attention to their customers
- Service providers will have to be very selective and precise in their interactions with their customers
- If we value privacy, someone will sell it to us
- Customers will not necessarily be human

General issues in u-Services

- **Enhanced middleware**
  - scalable processing of environmental data, disconnected operation, efficient resource management and real-time autonomous reaction
- **Understanding of the real world**
  - RFID mitigates the object recognition, but how to **understand** it?
  - Need for ontology
- **Understanding the human**
  - human attention, intention, behavior, preference
- And finally, **understanding the services**!
  - Again, what are services?

Challenges in ubiquitous web services

- Interoperability
- Embedding
- Performance
- Service provision
- Service consumption

Issues in ubiquitous web services

- *Dynamic discovery*
  - Can the directory based discovery mechanism such as UDDI still work?
  - Do we need something like PnP or P2P mechanisms?
- *Dynamic binding, composition, and coordination*
  - How to address the problem of service interface changes?
  - How to dynamically compose the ubiquitous services on-the-fly?
  - How to efficiently (in a distributed manner) coordinate / broker UWSs?
  - What are the effective means to support the collaboration between UWSs at runtime?

Issues in ubiquitous web services (cont.)

- *Location-based services*
  - Support of MIPs
  - Fusion of multi-dimensional information
- *Context awareness*
  - Need for ontology on the real-world
  - Interoperable semantics between UWSs
- *Performance*
  - Emergence of real-time WS
  - What to embed into small devices?
  - And how? Web services on chip? (Parser + SOAP processor + Security)

Issues in ubiquitous web services (cont.)

- *Interoperation with old and new systems*
  - How to interoperate with the existing (i.e., traditional) web services and other middleware systems?
  - How can we make them work on IPv6?
  - Interoperation with portals and portlets
- *Ubiquitous web services middleware*
  - Real-time, event-driven, stream-like messages
  - Dealing with heterogeneity between the middleware
- *Domain specialization*
  - USN, telematics, home networking, …
- *Leadership in standardization*
- *Business models and killer services*

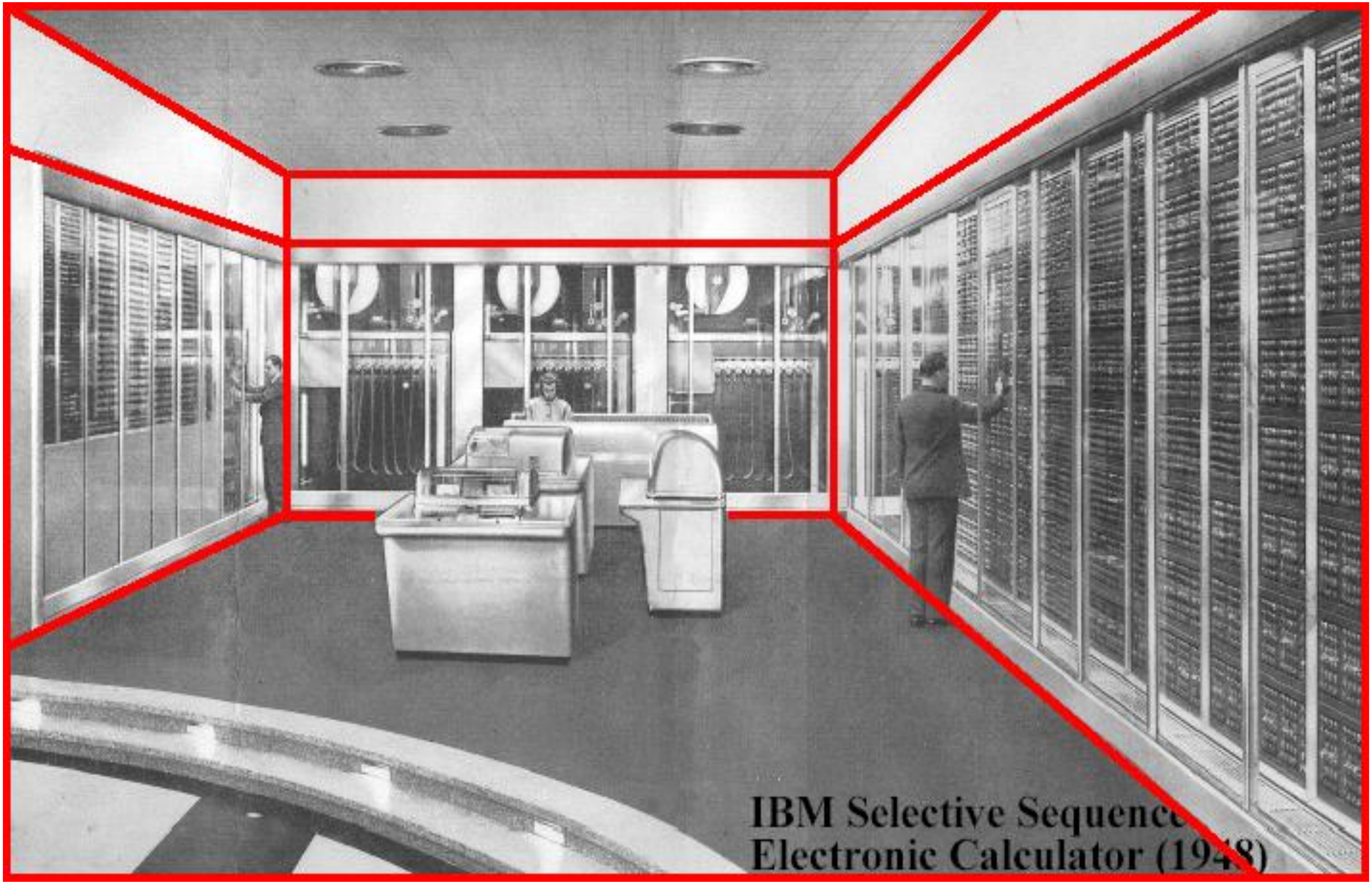
Pros and cons of UWS

- Pros
  - The **interoperability** problem is the crux of ubiquitous computing
- Cons
  - **Resource constraints**: computing power, battery, bandwidth, …
  - **Security**
- Observations
  - Web services are getting faster
  - Hardware performance and network bandwidth are getting increasing and cheaper
  - Better security mechanisms are emerging
- Remember: "**Web services are meant to be consumed by PROGRAMS**"

In conclusion

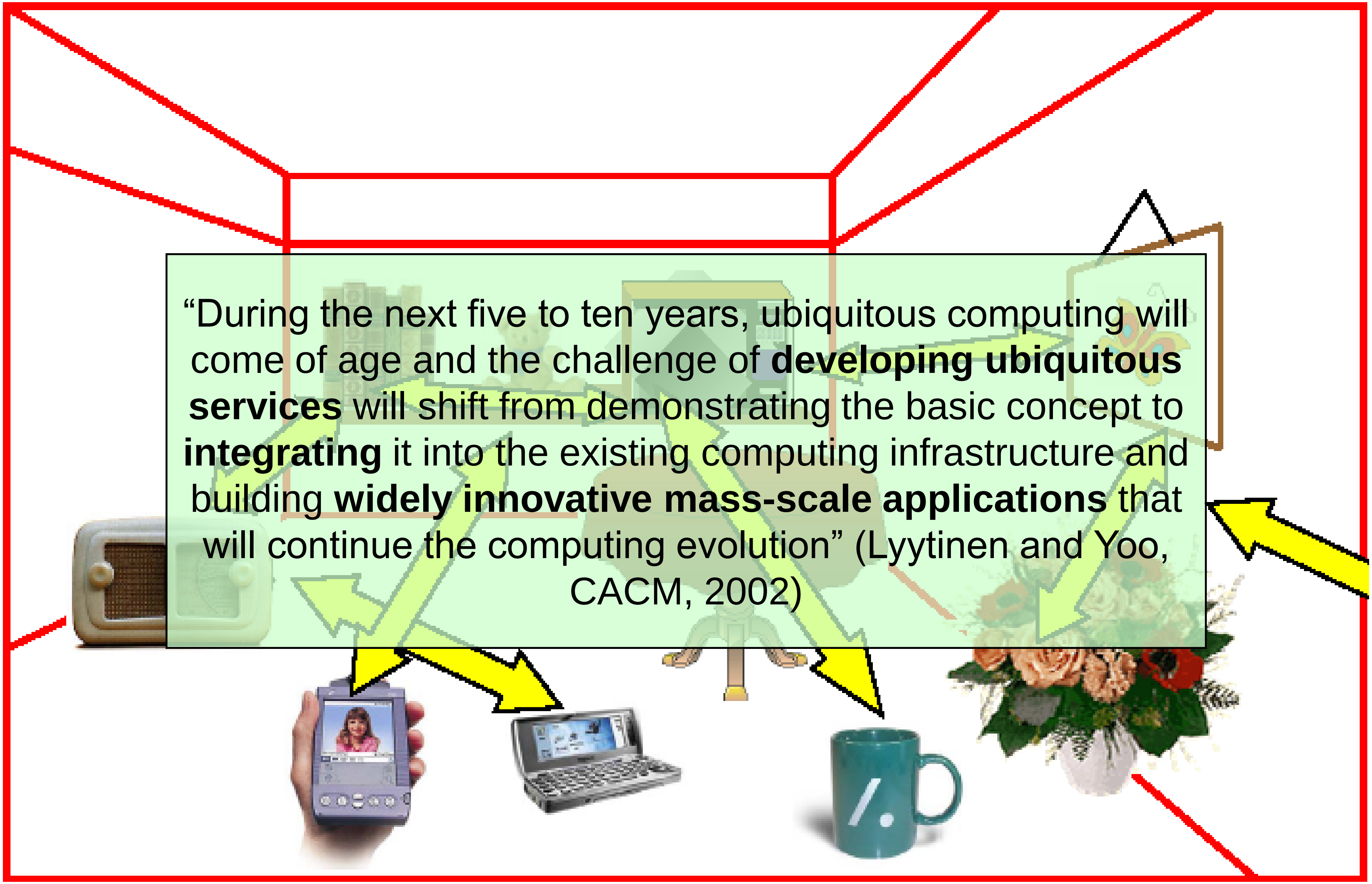
- Internet이 e-Business의 인프라 이었던 것처럼, e-Business는 u-Service를 위한 필수적인 인프라 역할을 할 것으로 판단됨
- 다양한 디바이스와 네트워크, 수 많은 종류의 기능들에 의해 야기되는 복잡성을 감출 수 있도록 서비스가 제공되어야 하며, 이러한 서비스들은 표준방식을 통해 연계되고 융합될 수 있어야 함
- 각종 국제 표준화 단체 및 유수 기업들은 u-Service에 대한 준비를 이미 시작하였으며, 그 인프라로 Web services를 채택하고 있음
- 국내에서는 모바일 웹 서비스, 웹 서비스 chip 및 device, UWS 모니터링, 제어, 이벤팅 기술, UWS 미들웨어 등에 대한 관련 기술 개발 및 국제 표준 주도가 시급
- 아울러, 비즈니스 가치가 있는 주요 서비스 모델의 발굴이 필요

Yesterday's Computers Filled Rooms …



IBM Selective Sequence Electronic Calculator (1948)

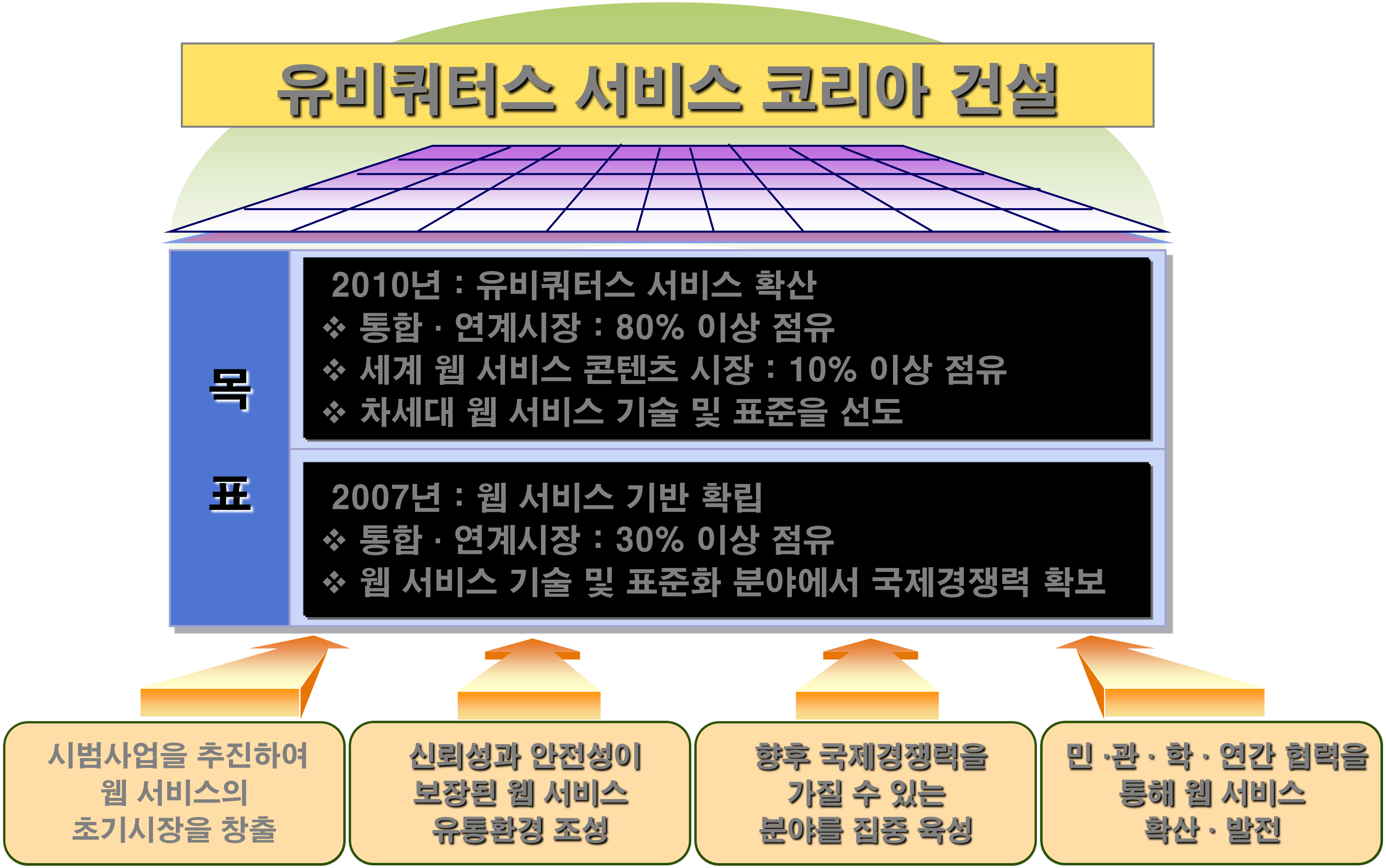
… So Will Tomorrow's



"During the next five to ten years, ubiquitous computing will come of age and the challenge of **developing ubiquitous services** will shift from demonstrating the basic concept to **integrating** it into the existing computing infrastructure and building **widely innovative mass-scale applications** that will continue the computing evolution" (Lyytinen and Yoo, CACM, 2002)

참고: 웹 서비스 비전과 추진전략

(출처: 정통부, 2004)



유비쿼터스 서비스 코리아 건설

목표

2010년 : 유비쿼터스 서비스 확산

❖ 통합 · 연계시장 : 80% 이상 점유

❖ 세계 웹 서비스 콘텐츠 시장 : 10% 이상 점유

❖ 차세대 웹 서비스 기술 및 표준을 선도

2007년 : 웹 서비스 기반 확립

❖ 통합 · 연계시장 : 30% 이상 점유

❖ 웹 서비스 기술 및 표준화 분야에서 국제경쟁력 확보

시범사업을 추진하여 웹 서비스의 초기시장을 창출

신뢰성과 안전성이 보장된 웹 서비스 유통환경 조성

향후 국제경쟁력을 가질 수 있는 분야를 집중 육성

민 ·관 · 학 · 연간 협력을 통해 웹 서비스 확산 · 발전

Web services resources

- Microsoft
  - Got dot net: http://gotdotnet.com/
  - Developer center: http://msdn.microsoft.com/webservices/
- IBM
  - AlphaWorks: http://www.alphaworks.ibm.com/
  - Developerworks: http://www-130.ibm.com/developerworks/webservices/
- Sun: http://java.sun.com/webservices/
- Apache: http://ws.apache.org/
- Coverpages: http://xml.coverpages.org/
- WebServices.Org: http://www.mywebservices.org/index.php/article/archive/61
- WebServices.XML.Com: http://webservices.xml.com/

Principles of Service-Oriented Computing

406.622 Industrial Information Technology

Jonghun Park

jonghun@snu.ac.kr

Dept. of Industrial Engineering

Seoul National University

9/20/2007

Benefits of SOC

- SOC enables new kinds of **flexible business applications** of open systems that simply would not be possible otherwise
- SOC improves the productivity of programming and administering applications in open systems

Use cases

- Intraenterprise interoperation
  - XML as a data format of choice
  - SOC provides the tools to model the information and relate the models, construct processes over the systems, assert and guarantee transactional properties, add in flexible decision support, and relate the functioning of the component software systems to the organizations that they represent
- Interenterprise interoperation
  - adhoc -> rigid EDI -> XML as a data format of choice
  - SOC provides the same benefits as for intraenterprise interoperation. In addition, it provides the ability for the interacting parties to choreograph their behaviors so that each may apply its local policies autonomously and yet achieve effective and coherent cross-enterprise processes

Use cases (cont.)

- Application configuration
  - Introduction of a new enterprise IT system requires that the **right interface be exposed** by the new system and by the existing systems
  - Messaging and semantics problem
  - SOC enables the customization of new applications by providing a web service interface that eliminates messaging problems and by providing a semantic basis to customize the functioning of the application
- Dynamic selection
  - business partner can be chosen on the fly
  - SOC enables dynamic selection of business partners based on QoS criteria that each party can customize for itself

Use cases (cont.)

- Software fault tolerance
  - SOC provides support for dynamic selection of partners as well as abstractions through which the state of a business transaction can be captured and flexibly manipulated; in this way, **dynamic selection** is exploited to yield application-level fault tolerance
- Grid
  - Several resources are made available over a network, and are combined into large applications on demand
  - SOC enables the efficient usage of Grid resources
- Utility computing
  - Computing resources are modeled as a utility analogous to electric power or telecommunications
  - Enterprise would concentrate on their core business and outsource their computing infrastructure to a specialist company (e.g., data centers)
  - SOC facilitates utility computing, especially where redundant services can be used to achieve fault tolerance

Use cases (cont.)

- Software development
  - SOC provides a semantically rich and flexible computational model that simplifies software development

Requirements of SOAs

- Loose coupling
  - **High-level contractual relationships** through which the interactions among the components are specified
- Implementation neutrality
  - Not to be specific to a set of programming languages
- Flexible configurability
  - The system is configured late and flexibly
  - The configuration can change dynamically
- Long lifetime
  - The components must exist long enough to be able to detect any relevant exceptions, to take corrective action, and to respond to the corrective actions taken by others
- Granularity
  - should be understood at a **coarse granularity** to **reduce dependencies** among the participants and to **reduce communications** to a few messages of greater significance
- Teams
  - Instead of a participant commanding its partners, computation in open systems is more a matter of business partners working as a team

RPC vs. document orientation

- RPC-centric view
  - treats services as offering a set of methods to be invoked remotely
  - more natural for the use case of making independently developed applications interoperate
- Document-centric view
  - treats services as exchanging documents with one another
  - coheres better with the use cases of applying service in open environments

Major benefits of SOC

- Services provide **higher-level abstractions** for organizing applications in large-scale, open environments
- The abstractions are standardized
- **Standards** make it possible to develop general-purpose tools to manage the entire system lifecycle, including design, development, debugging, monitoring, and so on
- The standards feed other standards

Composing services

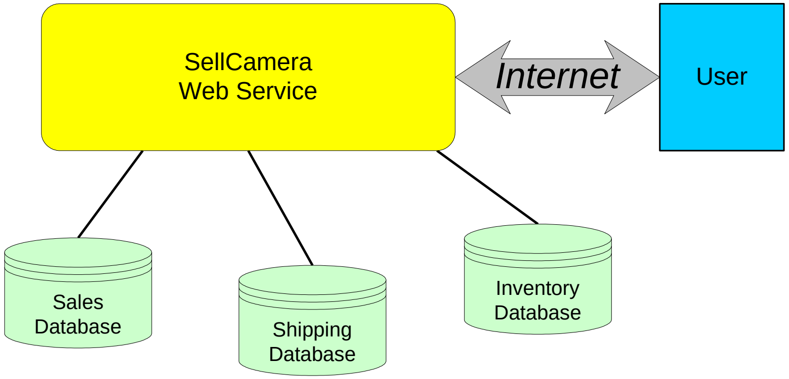
- Composition
  - Any form of putting services together to achieve some desired functionality
  - Recursive
- Composition leads to the **creation of new services** from old ones and can potentially add much value beyond merely a nicer interface to a single preexisting service
- The fruitful, challenging applications require services to be combined in ways that yield more powerful and novel uses
- The problem of **creating workflows** over the existing services
- Web sites can be thought of as not only offering content, but also providing services

Composition examples

- A travel agency web site could combine the services of Southwest Airlines, Dollar Rent-A-Car, and Sheraton to construct custom travel packages
- The airline's flight schedule might enable a fuel supplier to anticipate fuel purchases by the airline and to alert its refineries to adjust their production rates
- Yahoo provides a news service and Amazon provides a book selection service
  - Find the latest news headlines and search for books that match those headlines
  - Take the news from one service, filter it through a service that selects news based on a given user's interests, and pass the selected news items through a transcoding service to create a personalized web page that a user could review through a handheld device

Simple B2C Web Service Example

- Suppose you want to sell cameras over the Web, debit a credit card, and guarantee next-day delivery
- Your application must
  - update sales database
  - debit the credit card
  - send an order to the shipping department
  - receive an OK from the shipping department for next-day delivery
  - update an inventory database
- Problems: Some steps complete but not all
  - What if the order is shipped, but the debit fails?
  - What if the debit succeeds, but the order was never entered or shipped?



Database Approach

- A traditional database approach works only for a closed environment:
- Transaction processing (TP) monitors (such as IBM’s CICS, Transarc’s Encina, BEA System’s Tuxedo) can ensure that all or none of the steps are completed, and that systems eventually reach a consistent state
- But what if the user’s modem is disconnected right after he clicks on OK? Did the order succeed? What if the line went dead before the acknowledgement arrives? Will the user order again?
- The TP monitor cannot get the user into a consistent state!

Approach for Open Environment

- Server application could send email about credit problems, or detect duplicate transactions
- Downloaded applet could synchronize with server after broken connection was restored, and recover transaction; applet could communicate using http, or directly with server objects via CORBA/IIOP or RMI
- If there are too many orders to process synchronously, they could be put in a message queue, managed by a Message Oriented Middleware server (which guarantees message delivery or failure notification), and customers would be notified by email when the transaction is complete
- The server behaves like an agent!

Other challenges for composition

- Security will be more difficult
- **Incompatibilities** in vocabularies, semantics, and pragmatics among the service providers, service brokers, and service requesters
- As services are composed dynamically, performance problems might arise that were not anticipated
- Dynamic service composition will make it difficult to guarantee the QoS that applications require

Sprit of the approach

- To publish effectively, we must be able to **specify services with precision and with greater structure**
- From the perspective of the registry, it must be able to **certify the given providers** so that it can endorse the providers to the users of the registry
- Requesters of services should be able to find a registry that they can trust
- The requestor and the provider must develop a finer-grained sharing of representations
  - must be able to participate in conversations to conduct long-lived, flexible transactions
  - how a SLA can be established and monitored
- The key to the next-generation web are **cooperative services**, **systemic trust**, and **understanding based on semantics**, coupled with a declarative agent-based infrastructure

SOC Overview

- Revolutionary camp
  - The web services will facilitate the development of infrastructures that support programmatic application integration, dynamic B2B marketplaces, and the seamless integration of IT infrastructures from different corporations
- Evolutionary camp
  - Web services are just an additional layer on top of existing middleware and EAI platforms that provides a set of simple, lowest common denominator interfaces for interactions across the Internet
- The truth
  - Currently web services are mostly used today for conventional EAI
  - In the future, Web services may lead to a new computing paradigms and to dynamic B2B integrations
  - But, it remains to be seen whether this “revolutionary” power can be unleashed

Web services technology: Current status

- Most of the proposals, except SOAP and WSDL, are still at very early stages in the standardization process
- UDDI has changed its goals in a significant manner from version to version
  - UBR -> Private repositories
  - Currently most of the systems that provide support for SOAP and WSDL either ignore UDDI or provide only minimal support
- Very strong client/server flavor -> P2P interactions are likely to be the accepted interoperability paradigm in the future
  - A service is offered by a service provider and invoked by a service requester
  - Mainly due to the way SOAP works and is being used today

Web services technology: Current status

- SOAP is mainly being used for synchronous interactions -> Not well suited for B2B
  - Asynchronous interaction is not the way the vast majority of middleware platforms work today
- Adaptation of web services mainly involves transforming the procedural interfaces into message interfaces
- Need for much more infrastructure to be properly supported (e.g., security, transactions, coordination)

EAI as a natural fit for today's web services

- SOAP and WSDL are so close to conventional middleware that they seem to have been designed with mainly that purpose in mind
- Web services are indeed a natural solution to the standardization problems that have plagued EAI in the past
  - One of the biggest limitations in EAI is the **manual and ad hoc implementation of the wrappers** that allow different platforms and applications to interact with the EAI system
- Web services enable a company to open its IT infrastructure to external partners
  - There's a clear lack of support for the more advanced aspects of web services involved in realistic B2B settings (e.g., conversations, enforcement of business protocols, delivery guarantees)

The holy grail of web services

- Dynamic interaction in a completely open community
  - Automated clients browse UDDI registries, find adequate services and service providers, automatically discover how to interact with the service, and finally invoke the service, all programmatically without manual intervention
  - Automated interactions among applications that support different interfaces and protocols previously unknown to the clients
- The layers of extensions and enhancements of web services are being designed and it will take a quite a while before they are sufficiently established
- Full dynamism and complete automation will require a real revolution in technology

Complexity of B2B interactions

- Two business partners that decide to engage in a business transaction with each other according to a well-known, thoroughly designed, and well documented set of specifications
  - No exchange between the partners will be possible until they agree on the semantics of the documents involved
- Imprecise specifications: A “price” field within an XML document
  - Different currency units: Assume a given currency?
  - How should the currency unit be specified
  - Conventions on the decimal symbol
  - Inclusive or exclusive of sales tax? VATs?
  - Taxation rules?
- All practical situations have their own peculiarities
  - Despite the adoption of a specific standard, meetings and discussion among the interested parties are required before two companies can operate, to agree not only on the exact meaning of each data item, but also on how to exchange additional information for which there is no room in the standard specification
- Doing business with previously unknown partners is something that many companies, both clients and service providers, tend to avoid for many reasons: Trust, Quality, Legal agreements

Bypassing complexity in closed communities

- Many of the limitations of the current technology can be overcome “by hand”
- Most B2B implementations with web services will be of this kind, at least in the short and medium terms
- 2 approaches
    - Joint development teams that cover issues such as business protocols, semantics of the operations and their parameters, as well as contracts and legal issues
    - Existence of a dominant entity that simply prescribes what to do to invoke its Web services to anybody wanting to do business with it

Toward open communities

- Introducing an **intermediary**
  - The potential gains by acting as an intermediary in web services-based exchanges are very large!
  - Trust and QoS problems could be solved by introducing rating services
  - Can also verify and certify compliances to the specifications involved in the exchange
- Introducing **market markers** (aka hubs)
  - Create and control an e-marketplace, where customers and service providers meet to conduct business online
  - May define rules, constraints, legal bindings and offer a “protected” environment that makes e-business more reliable
  - Can mediate among possible differences in the implementation and deployment of the standard, and provide robustness to changes

Semantic web

- Aims at fully automating all the stages of the web services lifecycle
- Standardize the representation and handling of the semantic metadata used to describe web services and all aspects of using them (e.g., searching for services that are semantically equivalent to each other)
- Specifications
  - RDF, RDFS
  - DAML+OIL
  - OWL
  - DAML-S

RDF, RDFS

- RDF: Resource Description Framework
  - Designed for the representation and processing of metadata about resources on the web
  - Defines a model for describing relationships among resources in terms of uniquely identified properties and values
- RDFS: RDF Schema
  - Extends RDF by defining a class and property system similar to an object-oriented system
- It is possible to describe complex properties of resources (such as web services), and complex relations between these resources, using a notation that resembles that used in an OOS

DAML+OIL

- Ontology
  - A formal definition of a common set of terms used to describe and represent a domain of knowledge
  - Makes knowledge **reusable** by encoding formal definitions of basic concepts and the relationships among them
  - Usually expressed in a logic-based language so as to support automated reasoning upon them
  - Provide a means for representing the semantics of documents and for structuring and defining the meaning of standardized metadata terms
- DAML: DARPA Agent Markup Language
  - Extends XML, RDF, and RDFS to enable the creation and instantiation of ontologies that describe web services
- OIL: Ontology Inference Layer
  - A proposal for a web-based representation and an inference layer for ontologies
- DAML+OIL
  - A semantic markup language for web resources that extends RDF and RDFS with richer modeling primitives

OWL

- OWL: Ontology Web Language
- A revision of the DAML+OIL web ontology language currently being designed by W3C Web Ontology Working Group as a successor to DAML+OIL
- To facilitate greater machine readability of web content than XML, RDF, and RDFS by providing an additional vocabulary for term descriptions
- Allows untyped literals and provide a better data typecasting solution that is compatible with XML Schema and RDFS
- Provides 3 increasingly expressive sublanguages
    - OWL-Lite, OWL-DL (Description Logic), OWL-Full

DAML-S

- A DAML+OIL language for web services
- A collaborative effort by BBN technologies, CMU, Nokia, Stanford, SRI International
- To facilitate the description of the semantics of services, their interfaces, and their behavior
- Provides a means to create descriptions of web services that can be interpreted programmatically
- Would enable applications such as search engines to utilize the WSDL descriptions automatically

Enterprise application management

- The task of **monitoring and controlling applications** in an enterprise so that they can be made resilient to failures, configurable to changing needs of the business, accountable for billing and auditing, capable of performing under varying workloads, and secure to intended or unintended attacks
- Efforts for defining standards for interfaces between the management system and managed applications
  - Lifecycle interfaces for configuration management in CORBA
  - Java Management eXtensions (JMX) from Sun
  - Management Information Bases (MIBs) in SNMP
  - Common Information Model (CIM) from DMTF (Desktop Management Task Force)
  - Application Response Measurement (ARM)
- EAMS
  - HP OpenView, CA Unicenter, Tivoli

Web services management

- Management of applications within an enterprise
- Management of relationships with other web services across enterprises
- Web services simplify certain aspects of application management through their standardized abstractions
- In case of web services, additional tier is usually implemented using SOAP routers, conversation controllers, horizontal protocol handlers, and composition engines

Web services management architecture

- Operation invocations on a web service happen through a **SOAP router**
  - SOAP management can monitor and control the performance of a web service (e.g., response time thresholds)
- **Conversation controller** dispatches incoming messages to the right conversation instance
  - Conversation management can analyze conversation bottlenecks, manage their performance, and perform lifecycle operations, such as suspend and resume, on ongoing conversations
- **Composition engine** implements the business logic of web services operations by invoking them in a particular order
  - Composition management can correlate the response times of a web service's operations with the response times of its component web services
  - Also, it can analyze the composition model for bottlenecks, compare two alternative service providers for executing a step in the composition, or automatically select one of the service providers depending on historical data analysis

Business management

- Web services help bridge the gap between business management and application management
- Business management involves managing business metrics (e.g., revenue, # of completed orders) and business objectives (e.g., revenue targets, order targets)
- Application management involves managing the data collected from instrumenting the software and hardware that constitute the application
- SLM (Service Level Management): provides a gauge of how clients perceive the performance or availability of an application
    - SLM measurements: Operation response times, SLA violations, comparison between 2 supplier web services...

Cross enterprise management

- Need to manage relationships with web services across enterprise boundaries
- Requires additional web services protocols to be standardized
- Challenges: **limited visibility and control** over portions of the application that are not owned by the same enterprise, **lack of trust**
- When a web service composes other web services, it is not only the functional but also the non-functional attributes such as performance, availability, and security that get composed
- Need to support multi-party conversations

Support for multi-party conversations

- Measurements or states
  - WSDL is not being used to expose state information or measurements to clients
  - States: e.g., anticipated response time, the state of a conversation
  - Clients may need to query the attributes before composition and during execution
- Events
  - When two or more web services interact, a problem in one can affect the other
  - Need for decisions on how to isolate problems without propagating its effects to other components
  - It should be possible for one web service to generate events that are intended for another web service

Support for multi-party conversations

- Policies
  - A convenient mechanism to change the behavior of a system
  - To indicate what a management system may or may not do and must and must not do (e.g., no more than 5 unsuccessful login attempts are permitted)
- SLA (Service Level Agreements)
  - A mechanism for expressing constraints between web services
  - Similar to the CPA in ebXML
- Trust
  - May use trusted 3rd party
  - May be resolved by manual mechanisms (e.g., auditing)

Management through web services

- Use web services as a mechanism for managing infrastructure within or across enterprises -> Management is becoming an application of web services!
- OGSI (Open Grid Services Infrastructure) from GGF
  - Resources and clusters of resources are being wrapped with web services interfaces defined in WSDL
- Open Management Interface (OMI) from OASIS, Common Information Model (CIM) from DMTF
  - Based on web services standards or web services-like standards and are intended for managing infrastructure components such as devices, systems, applications, and business platforms in an enterprise
- Biz opp: Management outsourcing
  - MSP (Management Service Provider)

Web services management standards

- Just beginning to emerge
- Enable interoperability between management systems and managed resources wrapped as web services
- XMLCIM from DMTF
  - Enable interoperability between applications, repositories that store the CIM data model, and management systems that manage applications using the data model
  - An encoding of CIM data models in XML and a way of transporting that information using HTTP
  - Lagged behind the developments in the web services world

Web services management standards

- OMI from OASIS
  - Emerging standard that relies on SOAP and HTTP to exchange management information
  - Proposes standardized SOAP messages through which management data and controls can be exchanged between applications and management systems
- OGSI from GGF
  - A distributed computing infrastructure that builds on web services standards and protocols
  - Services created using OGSI are known as grid services
  - Proposes extensions to attach attributes (called service data) to port types in WSDL
    - (Note) Currently the only way in which the state of a web service can be exposed to clients through WSDL is by defining a new set of operations. However, # of attributes that comprise the state of a web service can be potentially very large

In summary

- More is needed before web services technology can be used to support dynamic business interactions among casual business partners
- Web services are already being used, and the applicability of this technology is likely to grow as extensions are added and as standardization efforts in areas such as reliability, security, and management finally converge and become generally accepted
- Once completed, this process will generate **a robust framework for taking the Internet to the next level**, from a web of unstructured information to a web of services through which customers and providers can conduct business in a seamless manner

Computing with Services

406.622 Industrial Information Technology

Jonghun Park

jonghun@snu.ac.kr

Dept. of Industrial Engineering

Seoul National University

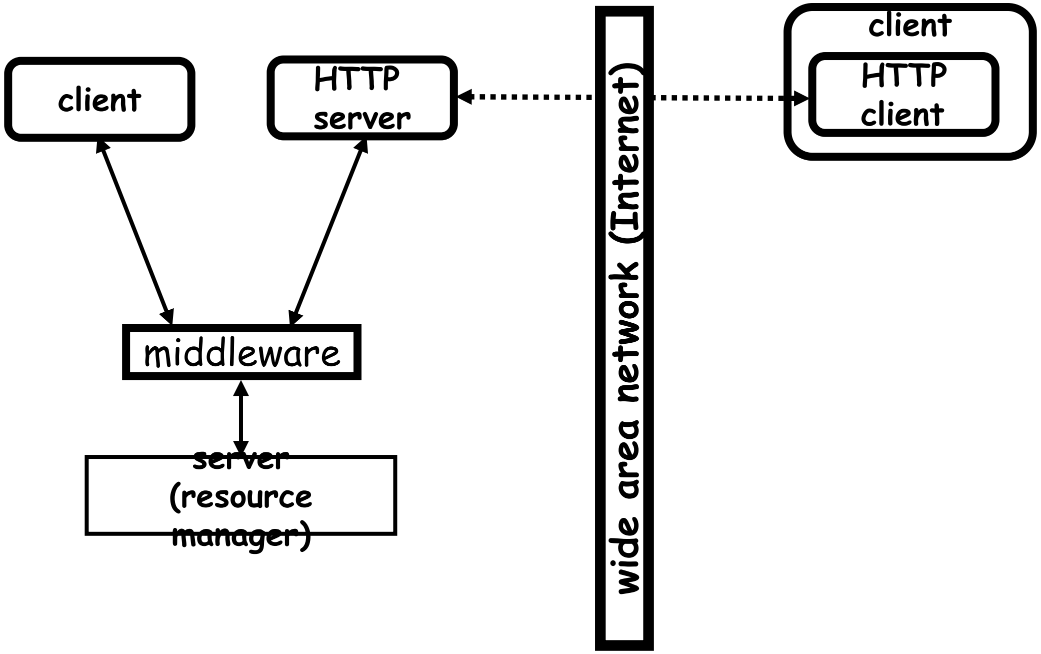
9/20/2007

Exchanging information over the Internet

- Internet: a global system of computer networks
- ARPANET
  - Stanford Research Institute, UCLA, UCSB, Univ. of Utah in 1969
  - TCP, IP
- Before the web
  - Telnet, SMTP (& MIME), FTP, Archie, Gopher

Web technologies for remote clients

- Motivation: ATM
- B2C: the business allows consumers to access their information services directly
- Web's contribution: provides a universal client



The web

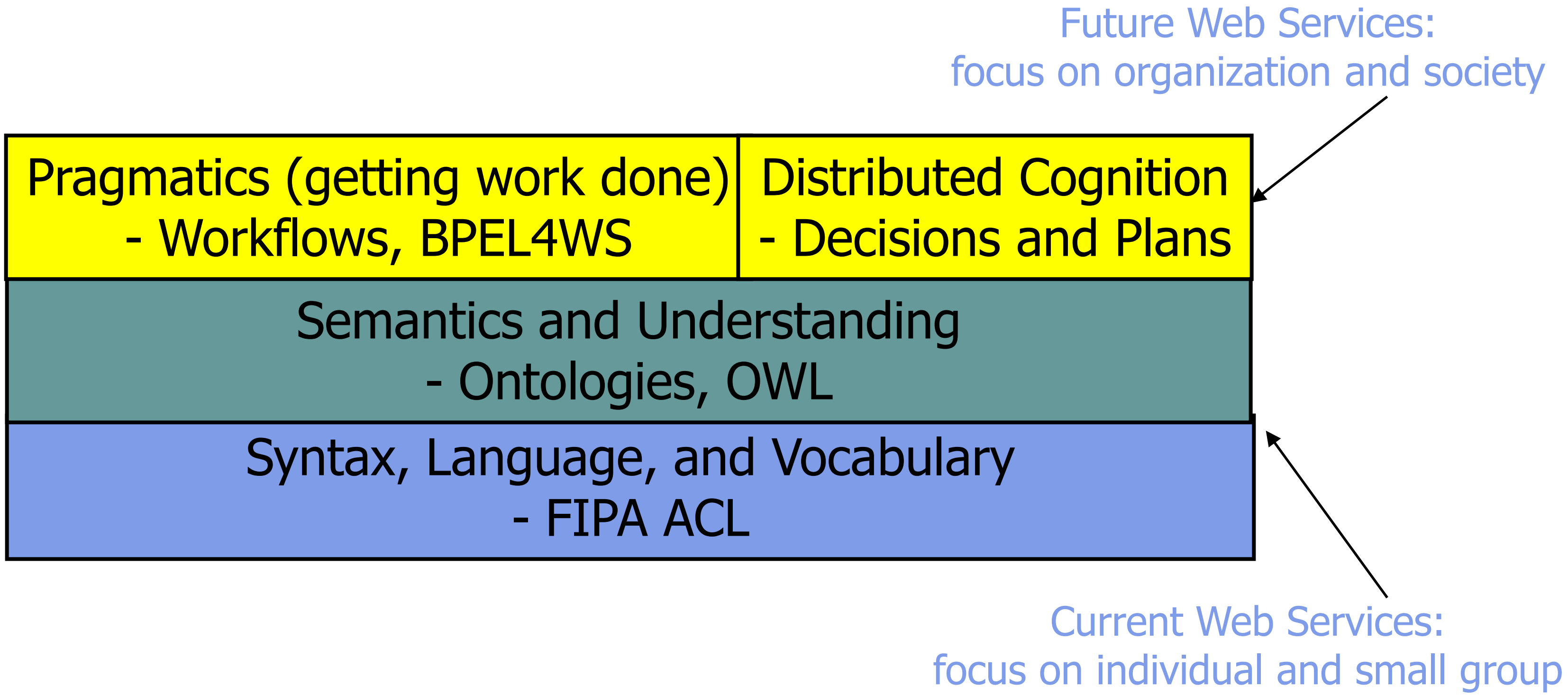
- HTTP, HTML, Web servers, Web browsers are an evolution of early technologies
- HTTP
  - A generic, stateless protocol that governs the transfer of files across a network
  - A client/server model using TCP/IP
  - GET, POST, PUT, DELETE, OPTIONS
  - Originally developed at CERN
- HTML
  - defines a standard set of special textual indicators (markups) that specify how a Web page’s words and images should be displayed by the web browser
- URI, URL
- W3C

Today’s web

- Designed for people to get information
- HTML describes how things appear
- HTTP is stateless
- Sources are independent and heterogeneous
- Processing is asynchronous client-server
- No support for integrating information
- No support for meaning and understanding
  - e.g., screen-scraping program that extracts the price of a book from a search results page on amazon.com
- ...

Pragmatic web

- Automation: Human -> Machine
- Richer markup: HTML -> XML -> ?
- Richer activities: Passive -> Active; Data -> Services -> Processes
- Greater interaction: Client-Server -> P2P -> Cooperative
- Accommodating context: Syntax -> Semantics -> Mutual Understanding -> Pragmatics and Cognition



Future Web Services:
focus on organization and society

Pragmatics (getting work done)
- Workflows, BPEL4WS

Distributed Cognition
- Decisions and Plans

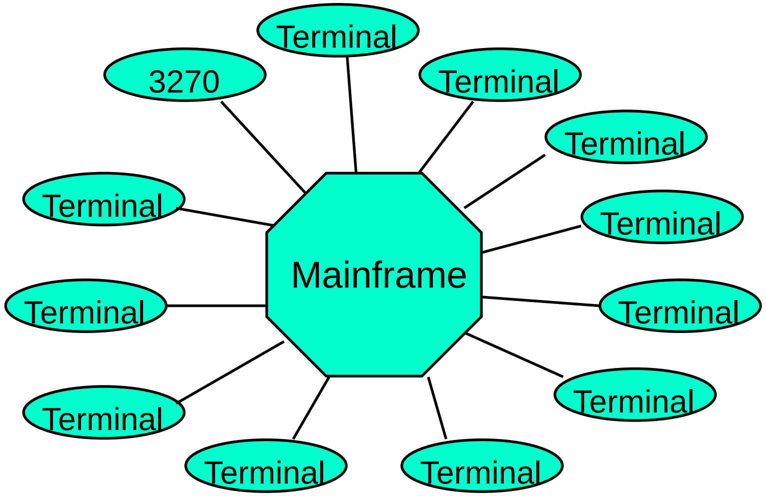
Semantics and Understanding
- Ontologies, OWL

Syntax, Language, and Vocabulary
- FIPA ACL

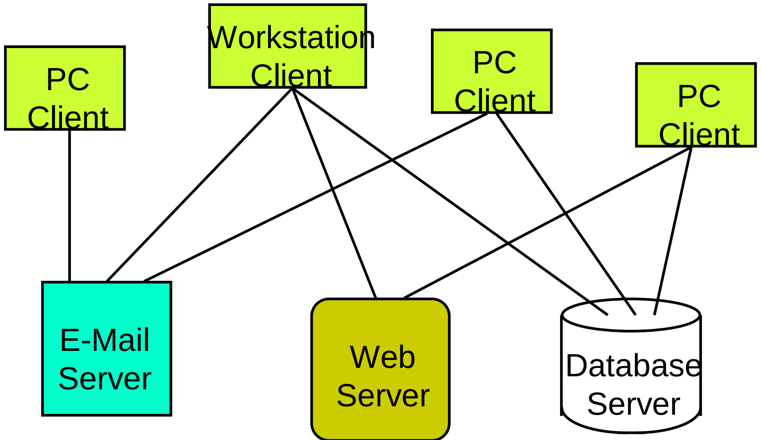
Current Web Services:
focus on individual and small group

Precursors

Centralized

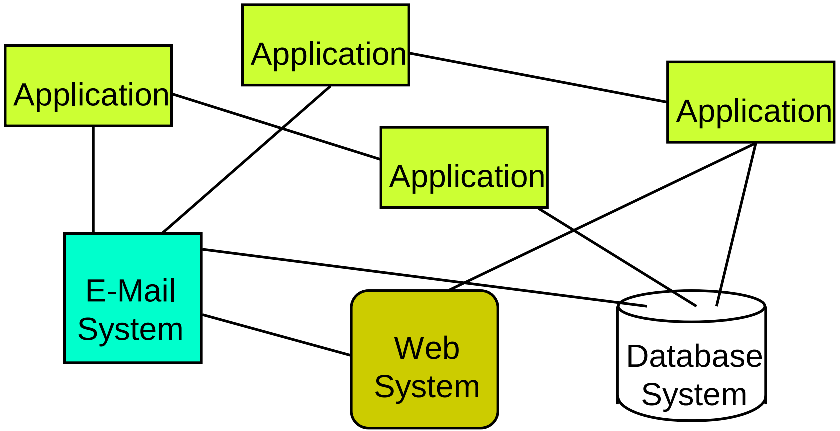


Client-Server

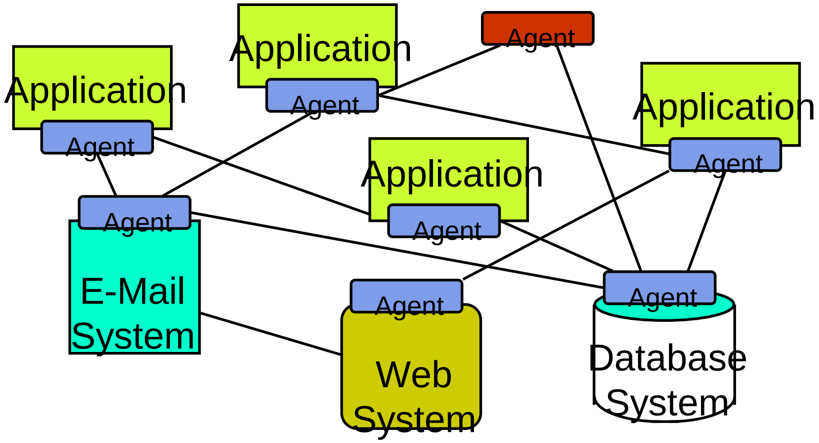


Master-Slave

P2P



Cooperative



Open Environments: Characteristics

- “Open”: implies that the components involved are **autonomous** and **heterogeneous**, and system configurations can change **dynamically**
- Cross enterprise boundaries
- Comprise autonomous resources that
  - Involve loosely structured addition and removal
  - Range from weak to subtle consistency requirements
  - Involve updates only under local control
  - Frequently involve nonstandard data
- Have intricate interdependencies

Autonomy

- The components in an environment function solely under their own control -> Independence of business partners (users)
- Political reasons
  - Ownership of resources
  - Control, especially of access privileges
  - Payments
- Technical reasons
  - Opacity of systems with respect to key features, e.g., precommit

Heterogeneity

- The various components of a given system are different in their design and construction -> Independence of component designers and system architects
- Political reasons
  - Ownership of resources
- Technical reasons
  - Conceptual problems in integration
  - Fragility of integration
  - Difficulty to guarantee behavior of integrated systems

Dynamism

- Autonomy -> Participants can behave arbitrarily and may join or leave an open environment at whim
- Needed because the parties change
    - Architecture and implementation
    - Behavior
    - Interactions
- Make configurations dynamic to improve service quality and maintain flexibility

Locality

- Global information (data, schemas, constraints) causes
  - Inconsistencies
  - Anomalies
  - Difficulties in maintenance
- Global information is essential for coherence
  - Locations of services or agents
  - Applicable business rules
- Relaxation of constraints works often
  - Obtain other global knowledge only when needed
  - **Correct rather than prevent** violations of constraints: often feasible
  - When, where, and how of corrections must be specified, but it is easier to make it local

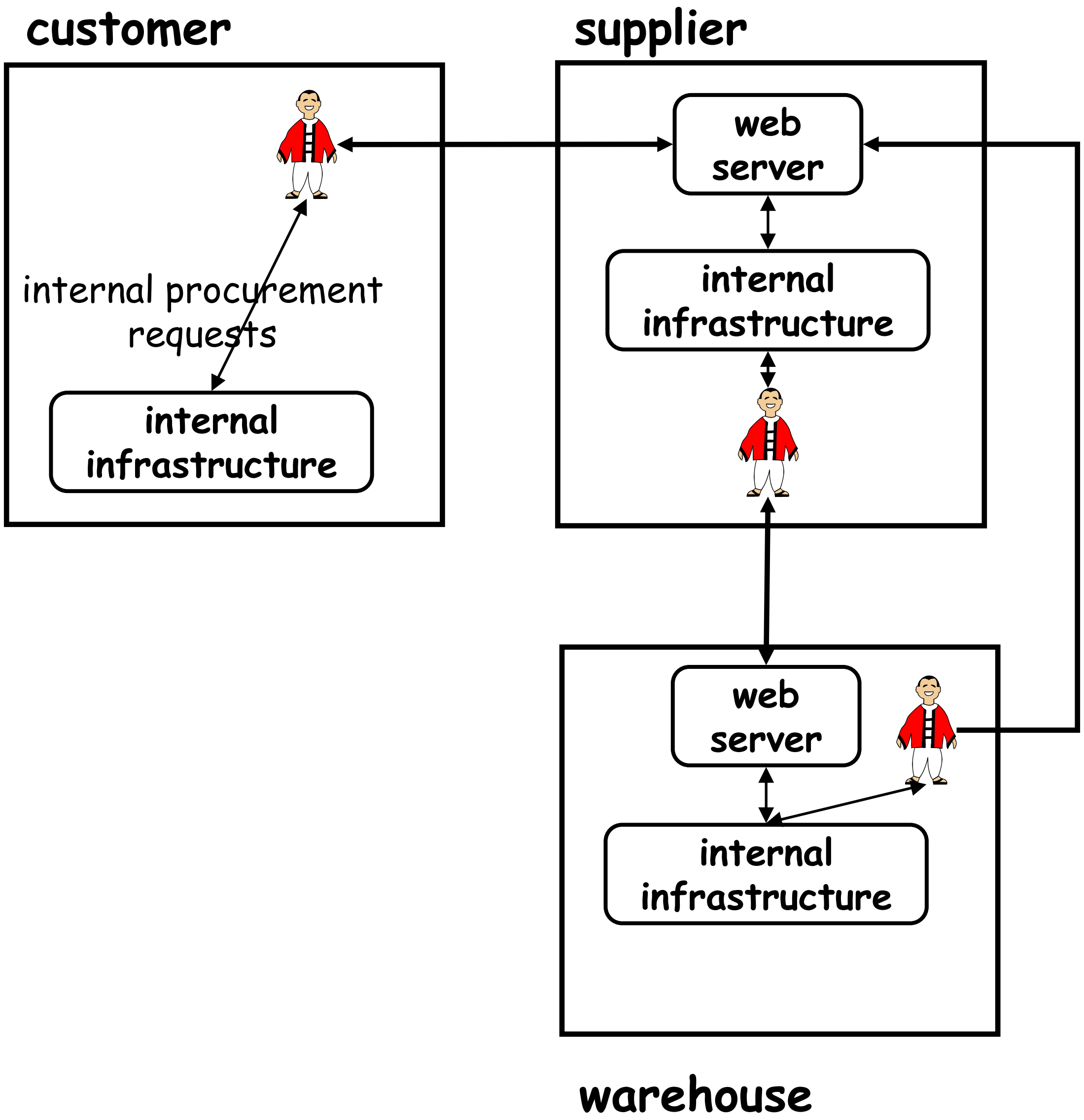
Definitions of web service

- "… a piece of business logic accessible via the Internet using open standards…" (Microsoft)
- Encapsulated, loosely coupled, contracted software functions, offered via standard protocols over the web (DestiCorp)
- Loosely coupled software components that interact with one another dynamically via standard Internet technologies (Gartner)
- Self-contained, modular business applications that have open, Internet-oriented, standards-based interfaces (UDDI consortium)
- A software application identified by a URI, whose interfaces and bindings are capable of being defined, described, and discovered as XML artifacts and supports direct interactions with other software applications using XML-based messages exchanged via Internet-based protocols (W3C)
- Our working definition: A WS is **functionality that can be engaged over the Web**

And still more: the recent definition from W3C

- Source: Web Services Glossary, Feb. 2004 (http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/)
- “A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL).
  Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.”

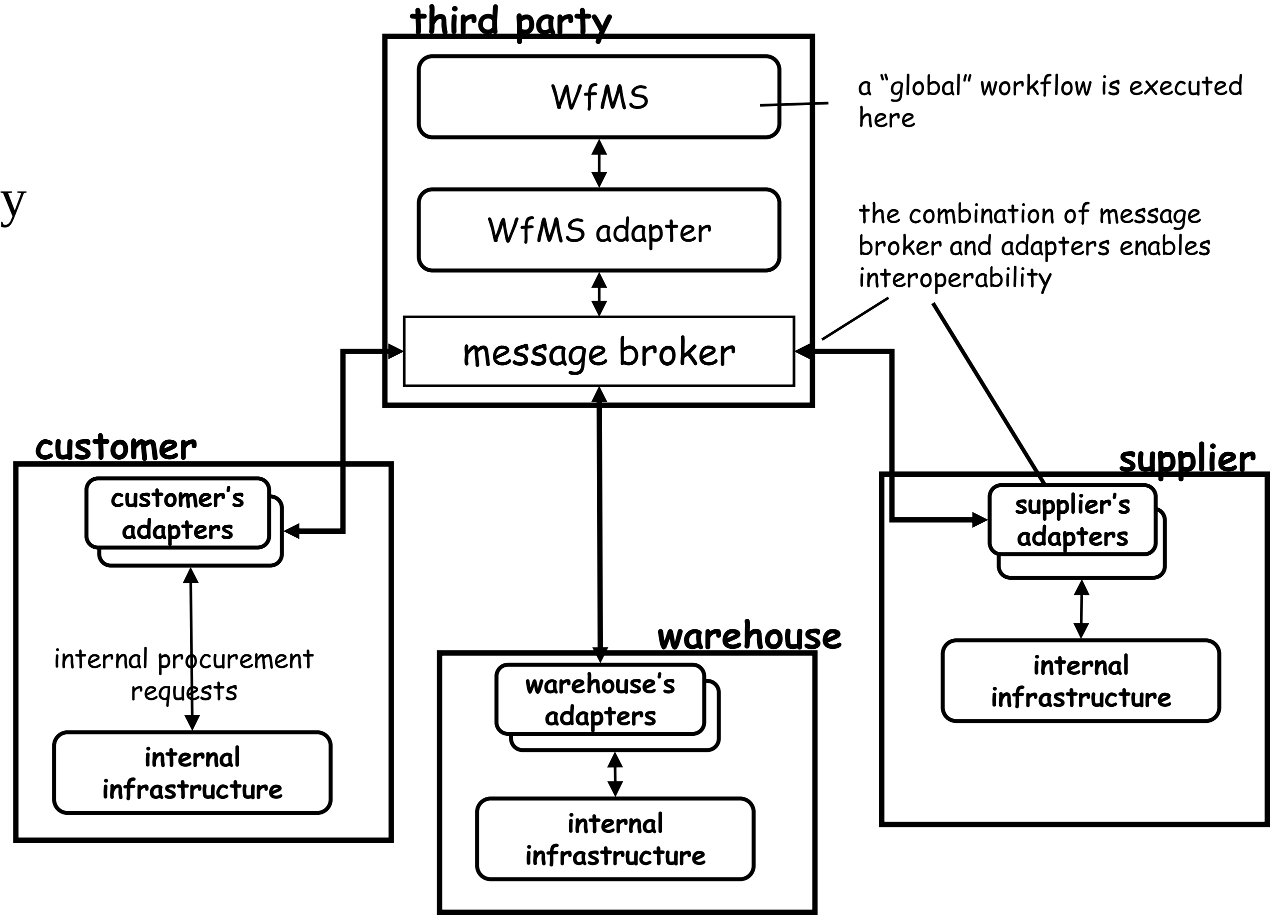
Motivating the need for Web services: B2Bi



B2B interactions occur by accessing Web pages, filling Web forms, or via email.

Limitations of conventional middleware for “global workflow”

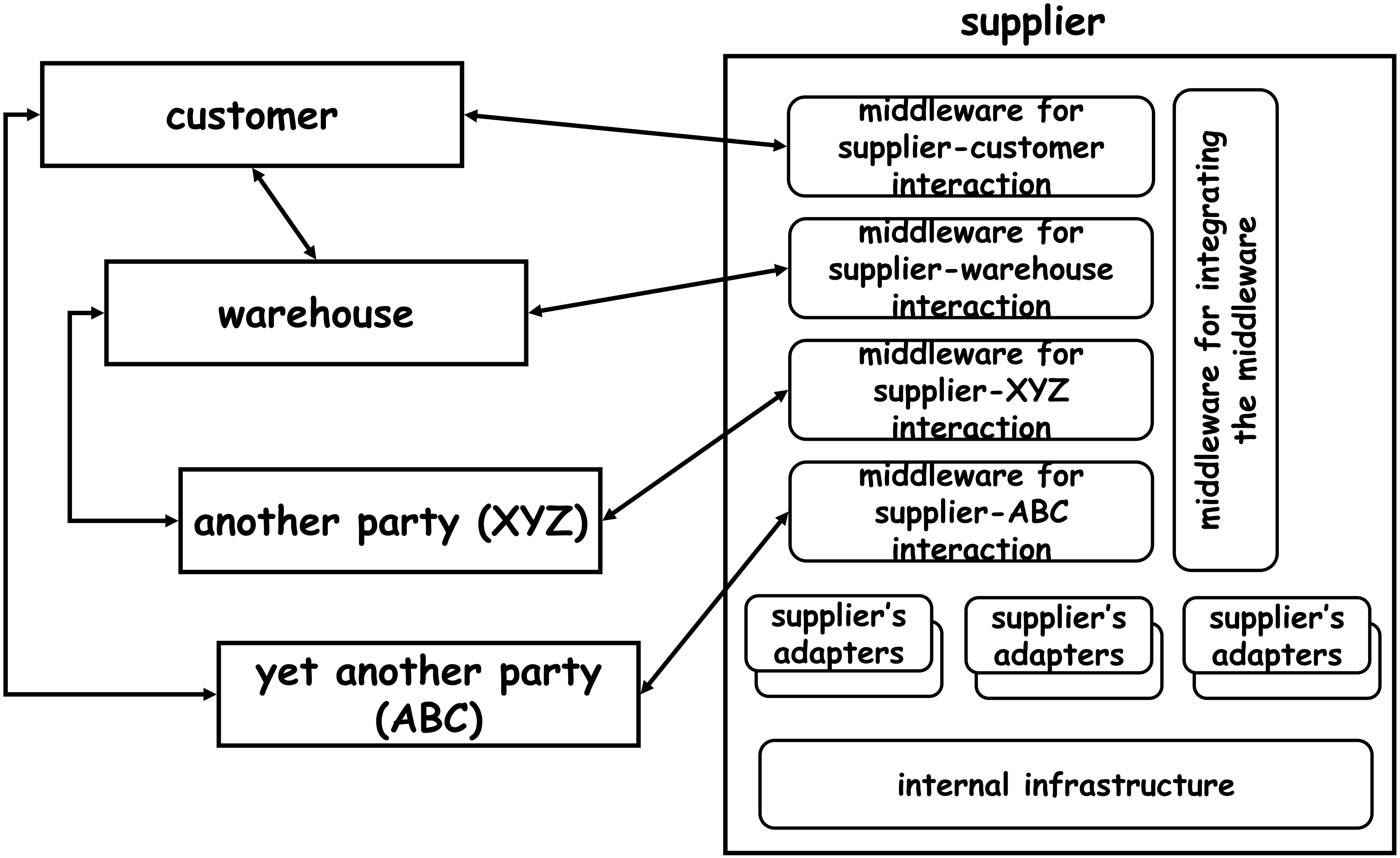
- No obvious place where to put the middleware
- Lack of trust
- Autonomy
- Confidentiality



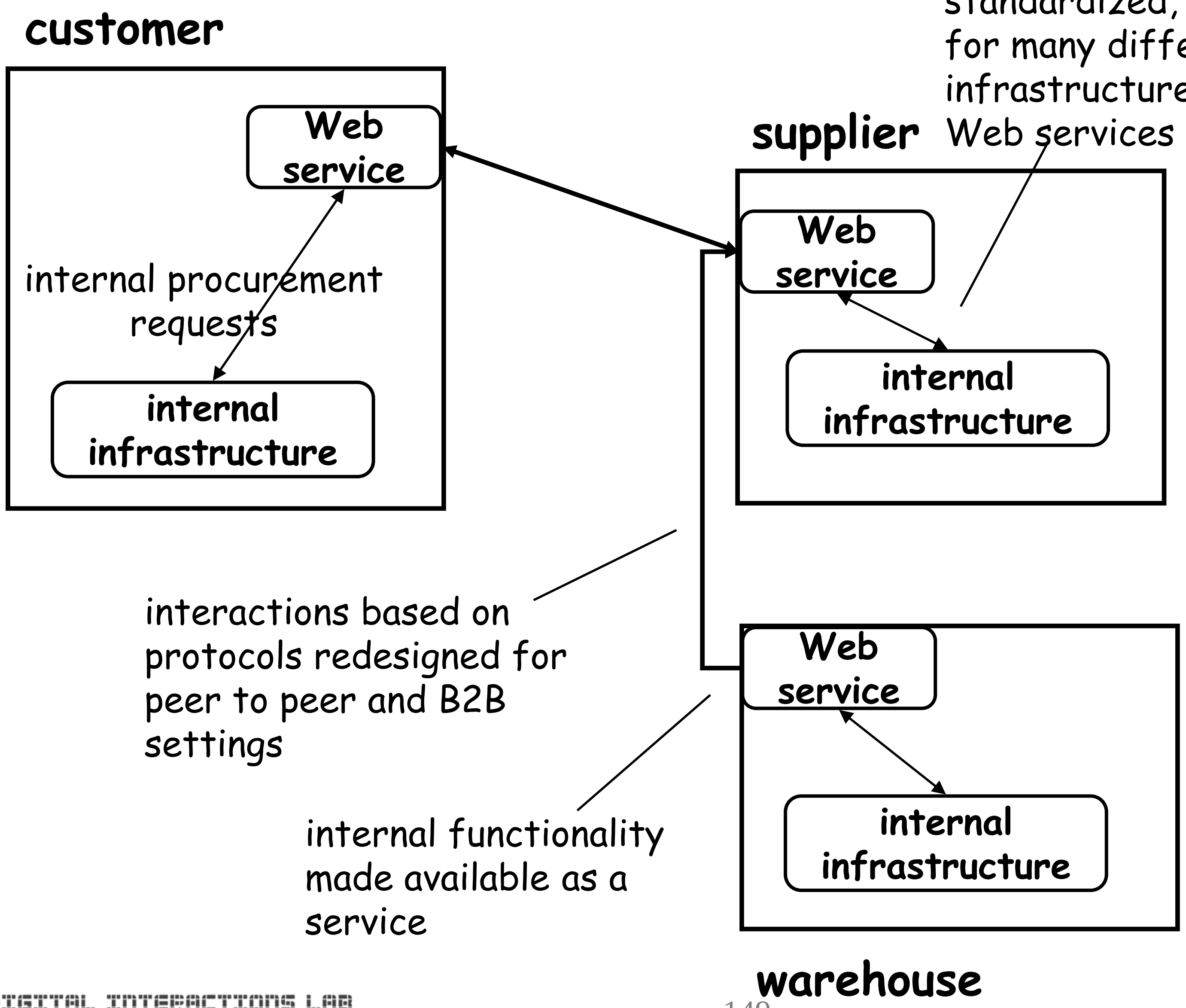
a “global” workflow is executed here

the combination of message broker and adapters enables interoperability

An alternative approach: P2P


supplier
customer
warehouse
another party (XYZ)
yet another party (ABC)
middleware for supplier-customer interaction
middleware for supplier-warehouse interaction
middleware for supplier-XYZ interaction
middleware for supplier-ABC interaction
middleware for integrating the middleware
supplier's adapters
supplier's adapters
supplier's adapters
internal infrastructure

SOA approach



internal procurement requests

languages and protocols standardized, eliminating need for many different middleware infrastructures (need only the Web services middleware)

interactions based on protocols redesigned for peer to peer and B2B settings

internal functionality made available as a service

Historical View of Services over the Web

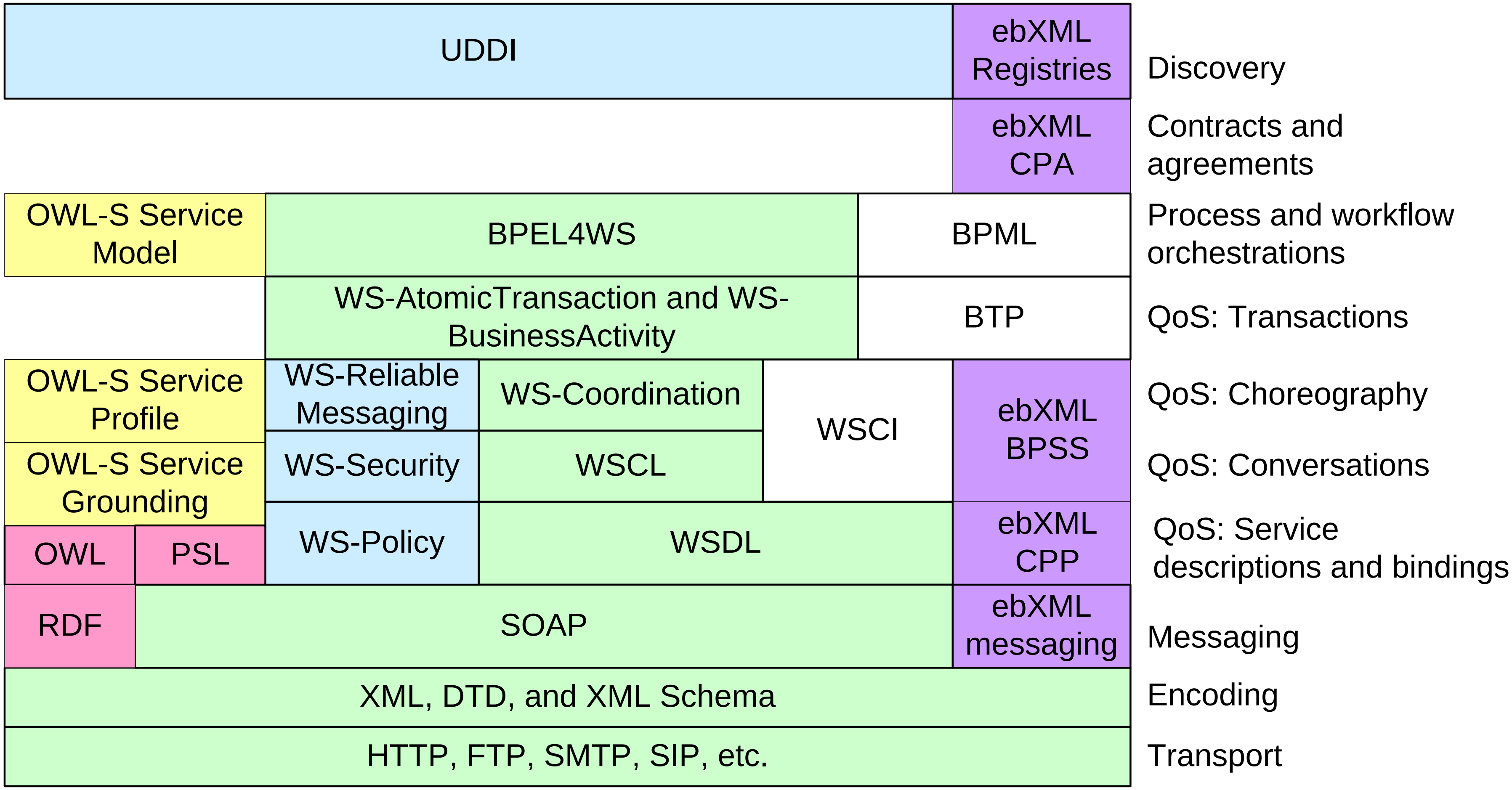
Generation	Scope	Technology	Example
First	All	Browser	Any HTML page
Second	Programmatic	Screen scraper	Systematically generated HTML content
Third	Standardized	Web services	Formally described service
Fourth	Semantic	Semantic Web services	Semantically described service

The Evolving Web

(B. Joy, former chief scientist of Sun, 2000)

- **Near** Web: conventional mouse-keyboard-monitor interaction with a personal computer, typically for purposes such as surfing the Web
- **Far** Web: interaction with a computer from across a room as with a TV remote control, typically for entertainment, such as listening to music or viewing a movie
- **Here** Web: interaction with a mobile device, with narrow bandwidths for input and output
- **Weird** Web: interaction through emerging interface technologies, such as voice and wearable computing
- Two additional webs without user interactions
  - B2B web
  - Pervasive web: device-to-device interactions

Standards for Web Services



UDDI | ebXML Registries | Discovery

ebXML CPA | Contracts and agreements

OWL-S Service Model | BPEL4WS | BPML | Process and workflow orchestrations

WS-AtomicTransaction and WS-BusinessActivity | BTP | QoS: Transactions

OWL-S Service Profile | WS-Reliable Messaging | WS-Coordination | WSCI | ebXML BPSS | QoS: Choreography

OWL-S Service Grounding | WS-Security | WSCL | QoS: Conversations

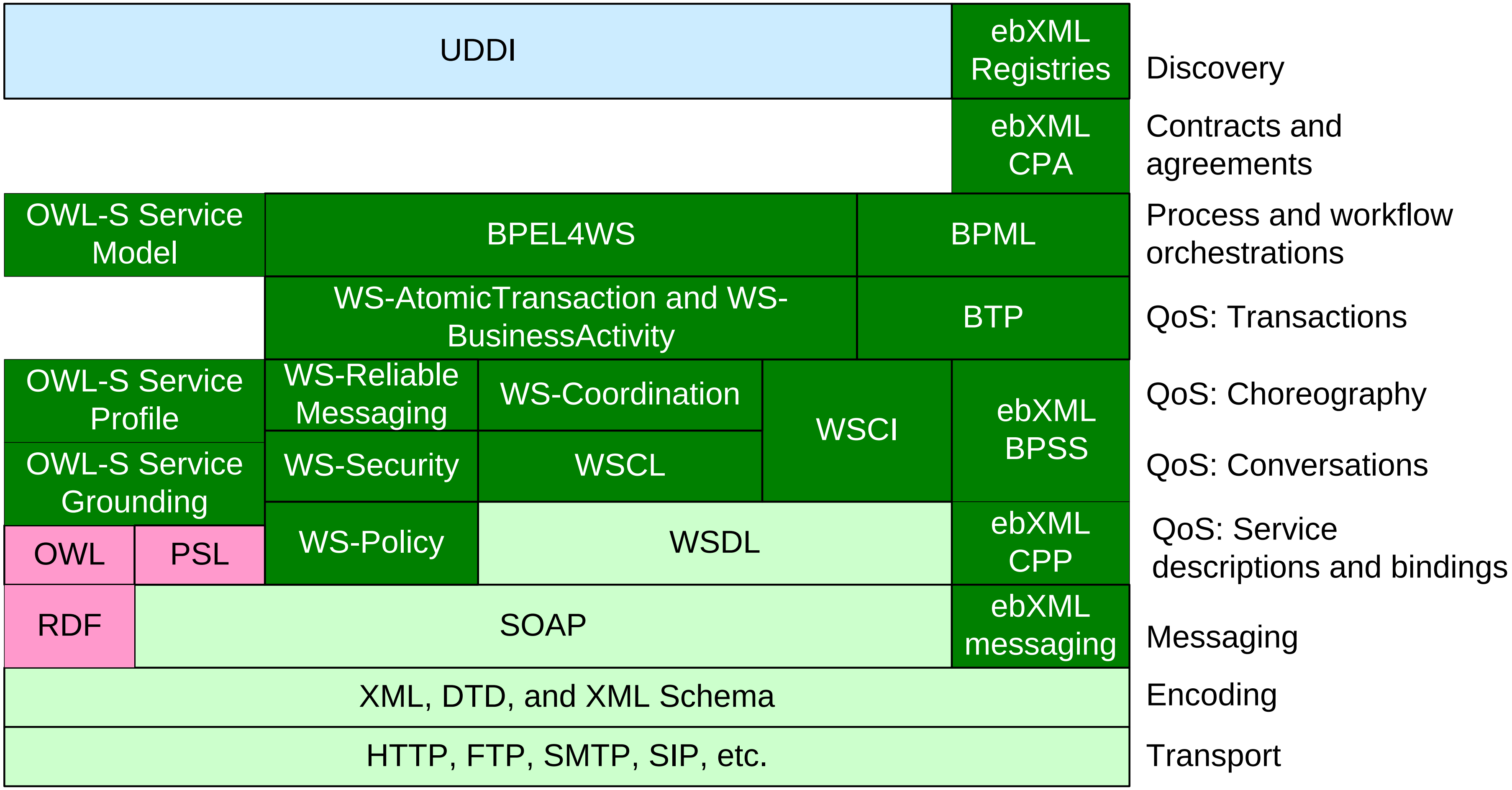
OWL | PSL | WS-Policy | WSDL | ebXML CPP | QoS: Service descriptions and bindings

RDF | SOAP | ebXML messaging | Messaging

XML, DTD, and XML Schema | Encoding

HTTP, FTP, SMTP, SIP, etc. | Transport

Process specifications in the web services stack



UDDI | ebXML Registries | Discovery

ebXML CPA | Contracts and agreements

OWL-S Service Model | BPEL4WS | BPML | Process and workflow orchestrations

WS-AtomicTransaction and WS-BusinessActivity | BTP | QoS: Transactions

OWL-S Service Profile | WS-Reliable Messaging | WS-Coordination | WSCI | ebXML BPSS | QoS: Choreography

OWL-S Service Grounding | WS-Security | WSCL | QoS: Conversations

OWL | PSL | WS-Policy | WSDL | ebXML CPP | QoS: Service descriptions and bindings

RDF | SOAP | ebXML messaging | Messaging

XML, DTD, and XML Schema | Encoding

HTTP, FTP, SMTP, SIP, etc. | Transport

Standards bodies

- IETF
- OMG
- W3C
- OASIS
- OMA
- 3GPP/3GPP2
- UPnP
- GGF
- UN/CEFACT
- WS-I
- BPMI.org
- WfMC
- EDIFACT
- FIPA
- …

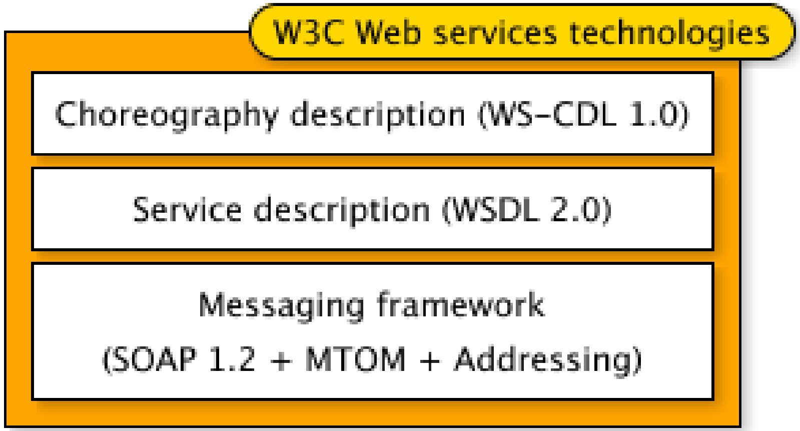
W3C's activities

- **Architecture** Domain
  - develops the underlying technologies of the Web.
- **Interaction** Domain
  - seeks to improve user interaction with the Web, and to facilitate single Web authoring to benefit users and content providers alike. It also works on formats and languages that will present information to users with accuracy, beauty, and a higher level of control.
- **Technology** and **Society** Domain
  - seeks to develop Web infrastructure to address social, legal, and public policy concerns.
- Web **Accessibility** Initiative (WAI)
  - is pursuing accessibility of the Web through five primary areas of work: technology, guidelines, tools, education and outreach, and research and development.

W3C's architecture domain

Architecture domain

- DOM
- Internationalization
- URI
- XML
- Web services
  - builds a set of technologies that allow **application-to-application interactions on the Web**
  - was formed by expanding the former XML Protocol Activity in January 2002


W3C Web services technologies
Choreography description (WS-CDL 1.0)
Service description (WSDL 2.0)
Messaging framework
(SOAP 1.2 + MTOM + Addressing)

Web services activity

- Objectives
  - W3C는 Web Services에 대한 infrastructure와 architecture, core technologies를 정의한다.
  - 2000년 9월 XML Protocol Activity가 시작되었고, 2002년 1월 Web Services Activity가 활동을 개시하였다.
- Groups
  - Web Services Architecture Working Group (now closed)
  - XML Protocol Working Group.
  - Web Services Addressing Working Group.
  - Web Services Description Working Group.
  - Web Services Choreography Working Group.
  - Semantic Web Services Interest Group.

Semantic Web Services Interest Group

- to provide an open forum to discuss Web Services topics essentially oriented towards integration of Semantic Web technology into the ongoing Web Services work at W3C
- Current open topics
  - W3C Web services technologies
  - Implementation
  - Deployment
  - Application design
  - Semantic Web technologies in Web services discovery, composition, ...
  - Mapping WS technologies to SW (RDF)
  - RESTful Web services
  - What do you use Web services for? Do you need/use Semantics?
  - Web services and agent technologies
  - ...

Other notable work

- Many interesting submissions related to web services can be found from Acknowledged Member Submissions (http://www.w3.org/Submission/)
- 2004
  - OWL Web Ontology Language for Services (OWL-S)
  - Semantic Web Rule Language (SWRL)
  - WS-MessageDelivery
- 2002
  - Web Service Choreography Interface (WSCI)
  - Web Services Conversation Languages (WSCL)
- 2001
  - Tentative Hold Protocol (THP)
  - …

Visions for the web

- Today, the components are primarily web pages, but increasingly they will be **programs** in general
- A dilemma: More distributed and independently managed that resources on the web become, the greater is their potential value, but the harder it is to extract that value
- Web as a provider for content and services

Process Specifications

406.622 Industrial Information Technology

Jonghun Park

jonghun@snu.ac.kr

Dept. of Industrial Engineering

Seoul National University

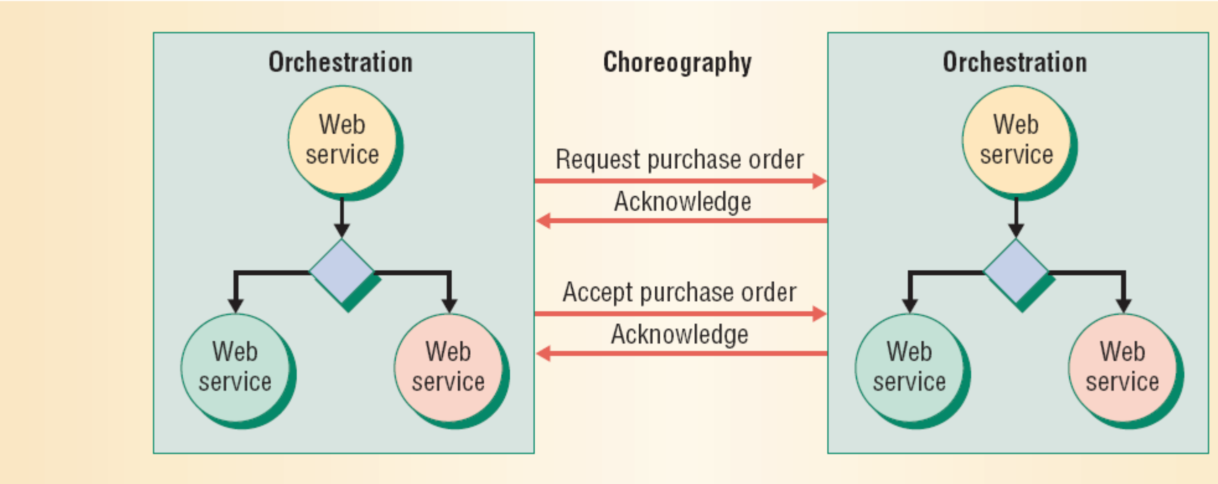
9/20/2007

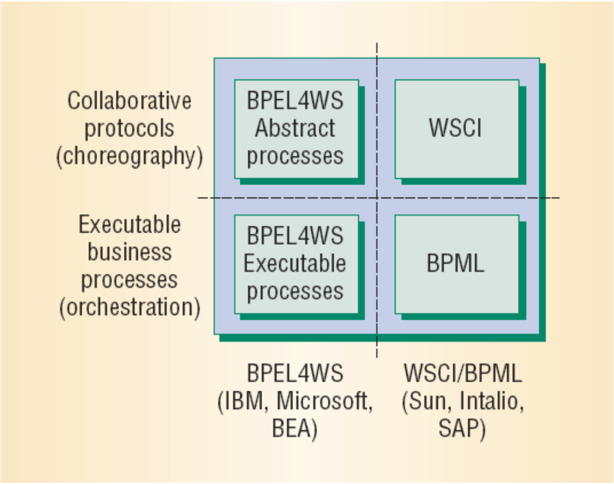
Processes

- A composite activity geared to serve some purpose
- Some combination of services that correspond to queries, transactions, applications, and administrative activities
- may be distributed within (intra-enterprise) or across enterprises (inter-enterprise)
- Technical challenges
  - Exceptions and revisions in process modeling
    - Long running -> the information they take as input may be subject to revision and thereby causing their own results to be invalidated
  - Interfacing a process to underlying functionalities

Process Abstractions

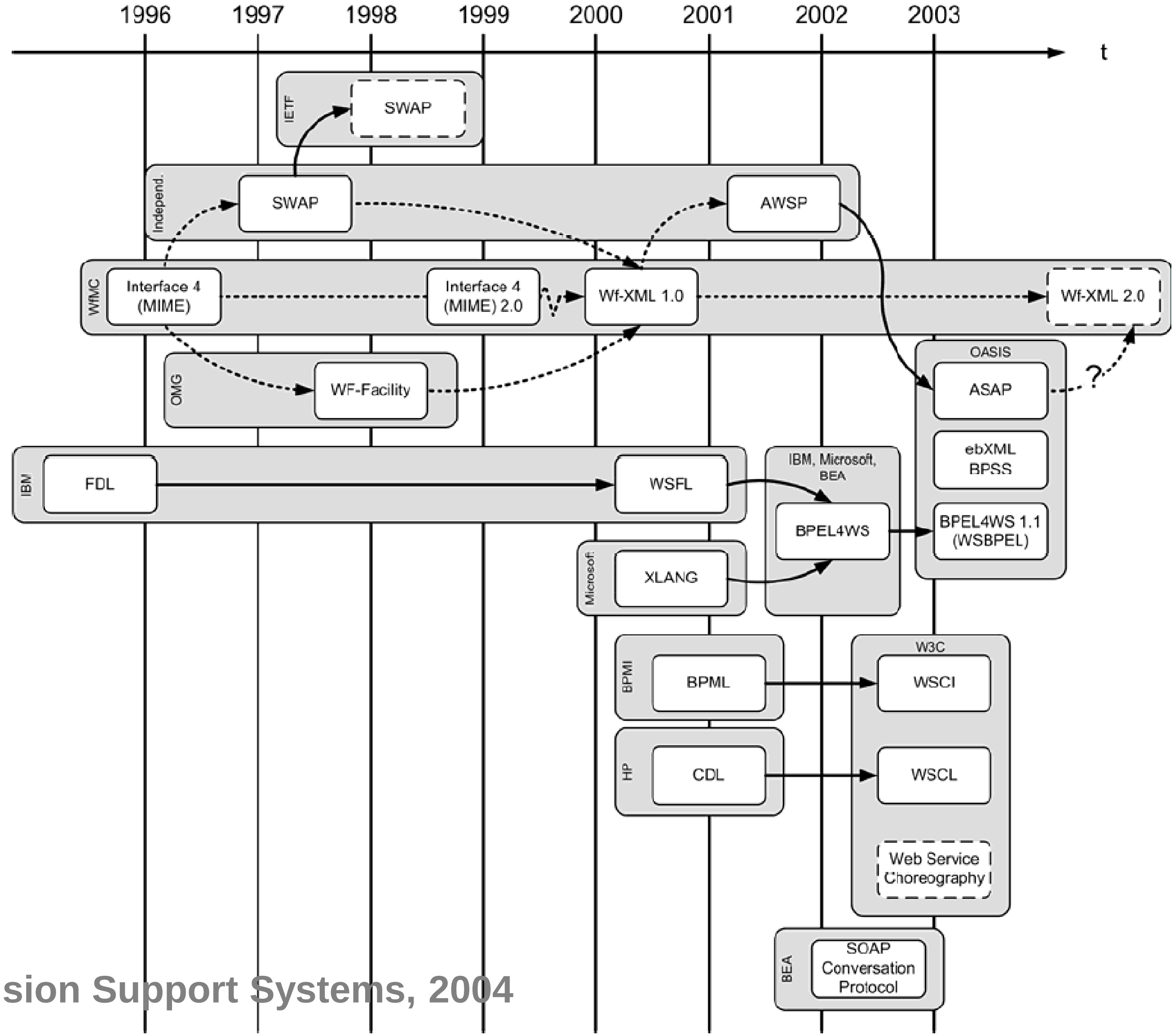
- Orchestration
  - views a process as a partial order of operations that need to be executed
  - views a process from the perspective of one orchestrating engine
  - corresponds best to the workflow representations
  - example: BPEL4WS, OWL-S
- Choreography
  - views a process as a set of message exchanges between participants
  - message exchanges are constrained to occur in various sequences and may be required to be grouped into various transactions
  - example: WSCL, WSCI, WS-CDL
- Collaboration
  - views a process as a collaboration among business partners
  - The business partners not only send messages to one another, but also enter into business relationships such as contracts and obligations





Collaborative protocols (choreography)
Executable business processes (orchestration)
BPEL4WS Abstract processes
WSCI
BPEL4WS Executable processes
BPML
BPEL4WS (IBM, Microsoft, BEA)
WSCI/BPML (Sun, Intalio, SAP)

Evolution of orchestration and choreography standards



Source: Decision Support Systems, 2004

Service composition

- Implementation of web services whose **business logic involves the invocation of operations offered by other web services**
  - Composite service: a service implemented by combining the functionality provided by other web services
  - Service composition: the process of developing a composite web service
- Web services composition middleware
  - provides abstractions and infrastructures facilitating the definition and execution of a composite service

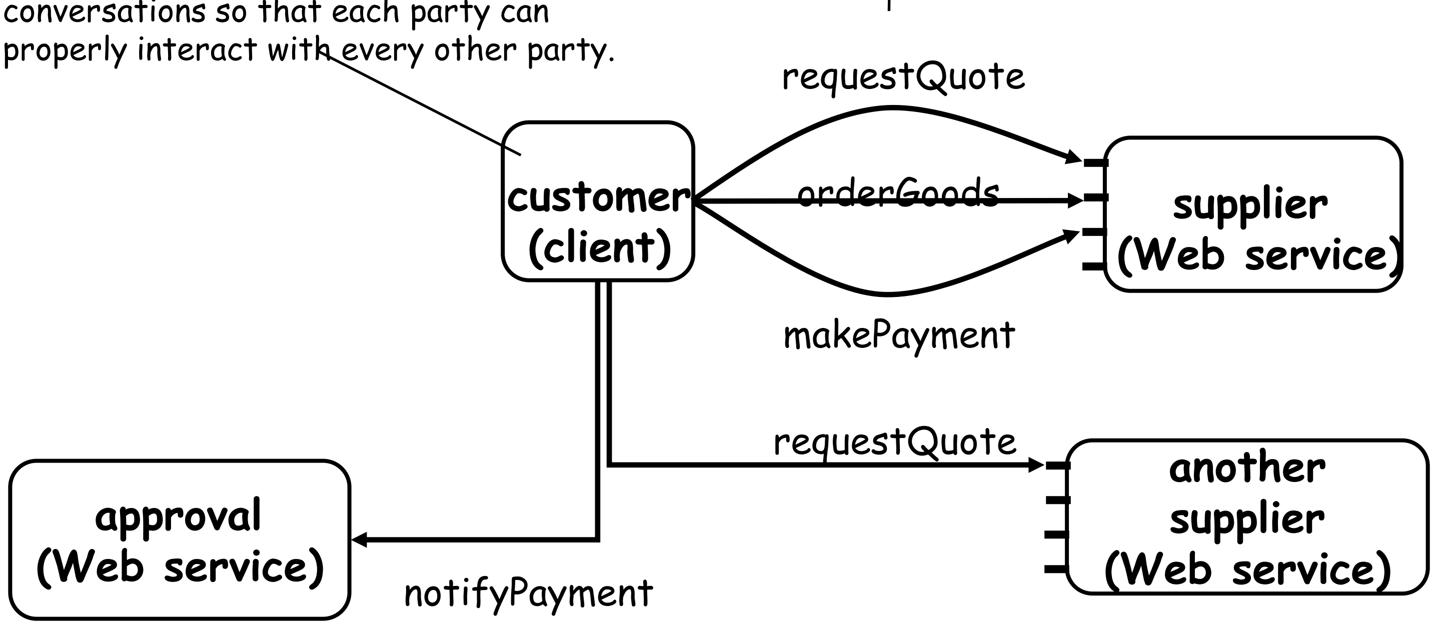
Basics of service composition

- The business logic of a client can be realized by **composing multiple services**, and by executing a different conversation with each of them
- **A client can itself be exposed as a web service**
    - Service composition can be iterated, allowing the definition of increasingly complex applications by progressively aggregating components, at increasingly higher levels of abstraction
- It is the responsibility of a client to **implement all the protocols needed to communicate with the invoked services**
- Composition of web services is not based on the physical integration of all components -> Based on interfaces
- Composition of web services equates to specifying **which services need to be invoked, in what order, and how to handle exceptional situations**

Composition example

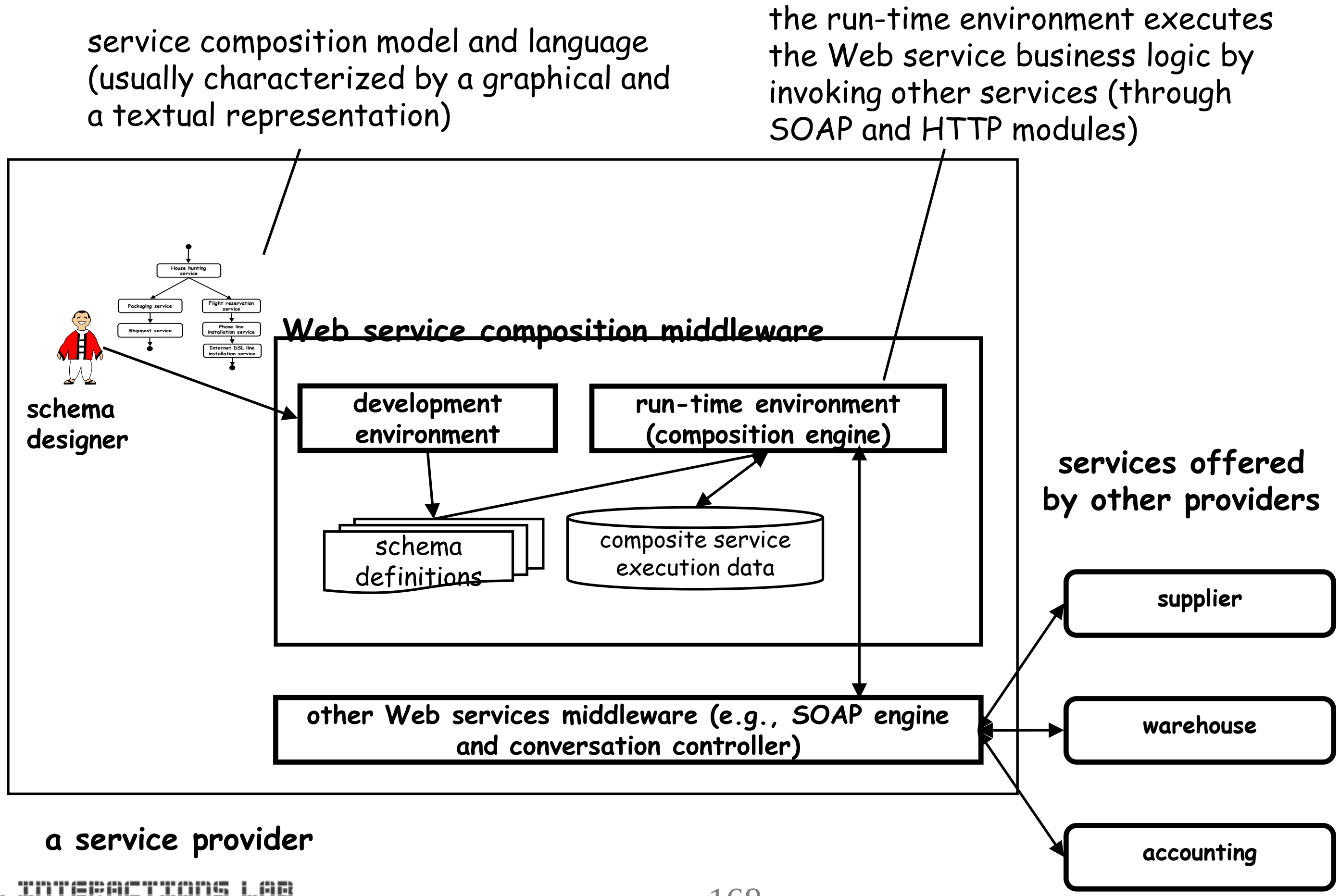
A client engages in different conversations with several Web services. In general, these conversations may be regulated by different protocols, and each invoked Web service may not be aware that the client is invoking other Web services.

The internal business logic of clients and Web services is quite sophisticated, as it must support the execution of different conversations so that each party can properly interact with every other party.



Need for service composition middleware

- Conventional programming languages were not designed with web service composition in mind
- In the absence of web service composition middleware, the business logic and these low-level details are intermingled in the code



service composition model and language (usually characterized by a graphical and a textual representation)

the run-time environment executes the Web service business logic by invoking other services (through SOAP and HTTP modules)

House hunting service
Packaging service
Shipment service
Flight reservation service
Phone line installation service
Internet DSL line installation service
schema designer
Web service composition middleware
development environment
run-time environment (composition engine)
schema definitions
composite service execution data
other Web services middleware (e.g., SOAP engine and conversation controller)
a service provider
services offered by other providers
supplier
warehouse
accounting

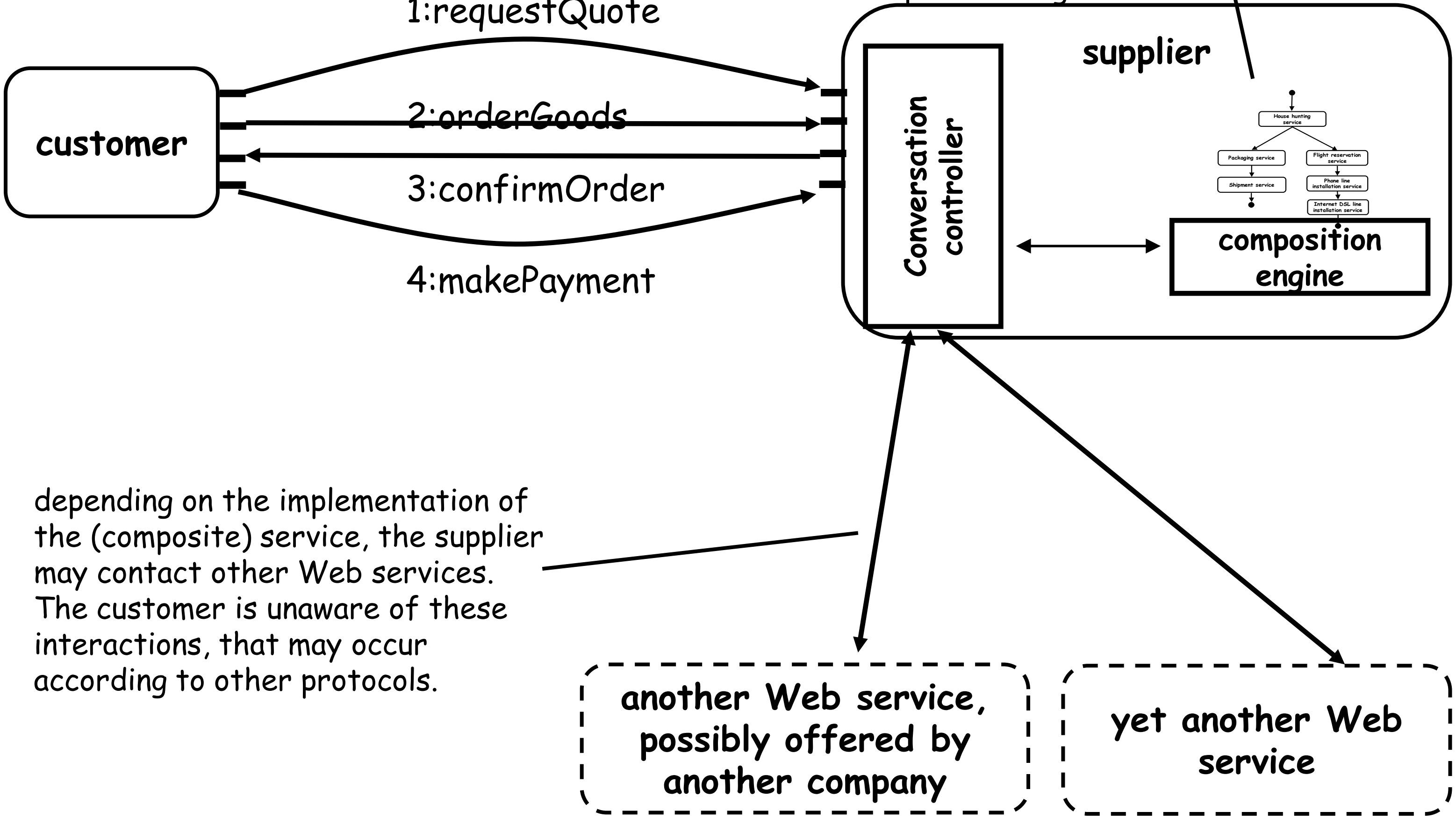
Composition vs. coordination middleware

- Composition: Internal
  - The specification of a composite service is done by a company and is kept private
  - The specification of the composition is for the consumption of the WS middleware
  - Whether a service is basic or composite is irrelevant from the client’s perspective
  - Determines the conversations that a composite service is able to execute
- Coordination protocols: External
  - Are public documents
  - Meant to be advertised in WS registries
  - Support design-time discovery and run-time binding
  - Impose requirements on how the composition is to take place, since the order in which operations are invoked has to be compliant with the protocol definition

Scopes of composition and coordination

the *procurement* business protocol executed among Web services

if the supplier is implemented by means of composition technologies, then its business logic is defined by a composition schema and its execution is driven by a composition engine.



depending on the implementation of the (composite) service, the supplier may contact other Web services. The customer is unaware of these interactions, that may occur according to other protocols.

Dimensions of a WS composition model

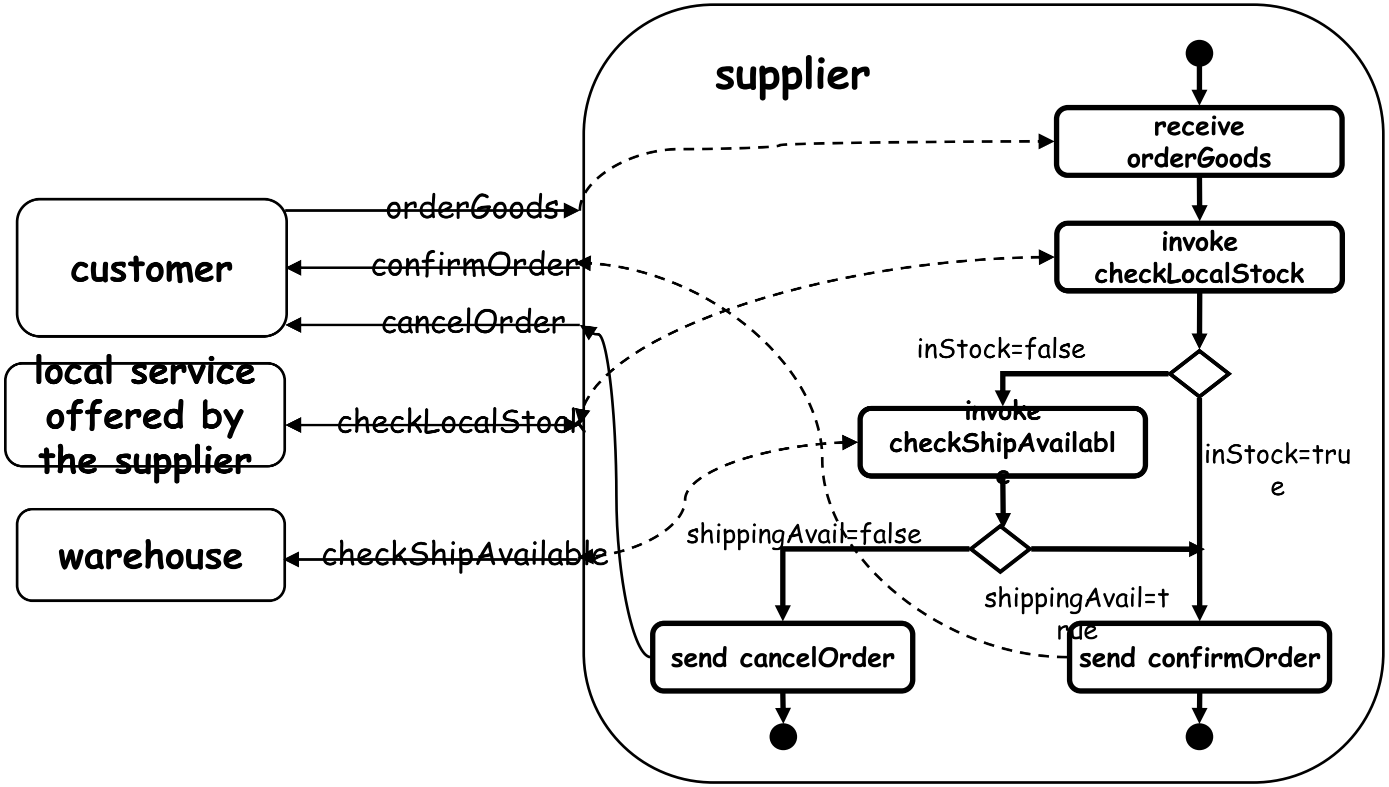
- Component model
  - Defines the **nature of the elements** to be composed in terms of the assumptions that the model makes on such components
- Orchestration model
  - Defines abstractions and languages used to define the order in which services are to be invoked
- Data and data access model
  - Defines how data is specified and how it is exchanged between components
- Service selection model
  - Defines how static or dynamic binding takes place
- Transactions
  - Defines which transactional semantics can be associated with the composition, and how this is done
- Exception handling
  - Defines how exceptional situations occurring during the execution of the composite service can be handled

Component model

- The components implement a specific set of web service standards such as HTTP, SOAP, WSDL, and WS-Transaction
  - Limits heterogeneity -> Makes composition easier
- The components interact by exchanging XML messages in either a synchronous or asynchronous fashion
  - The model is more general
  - Increased heterogeneity of the components

Orchestration model

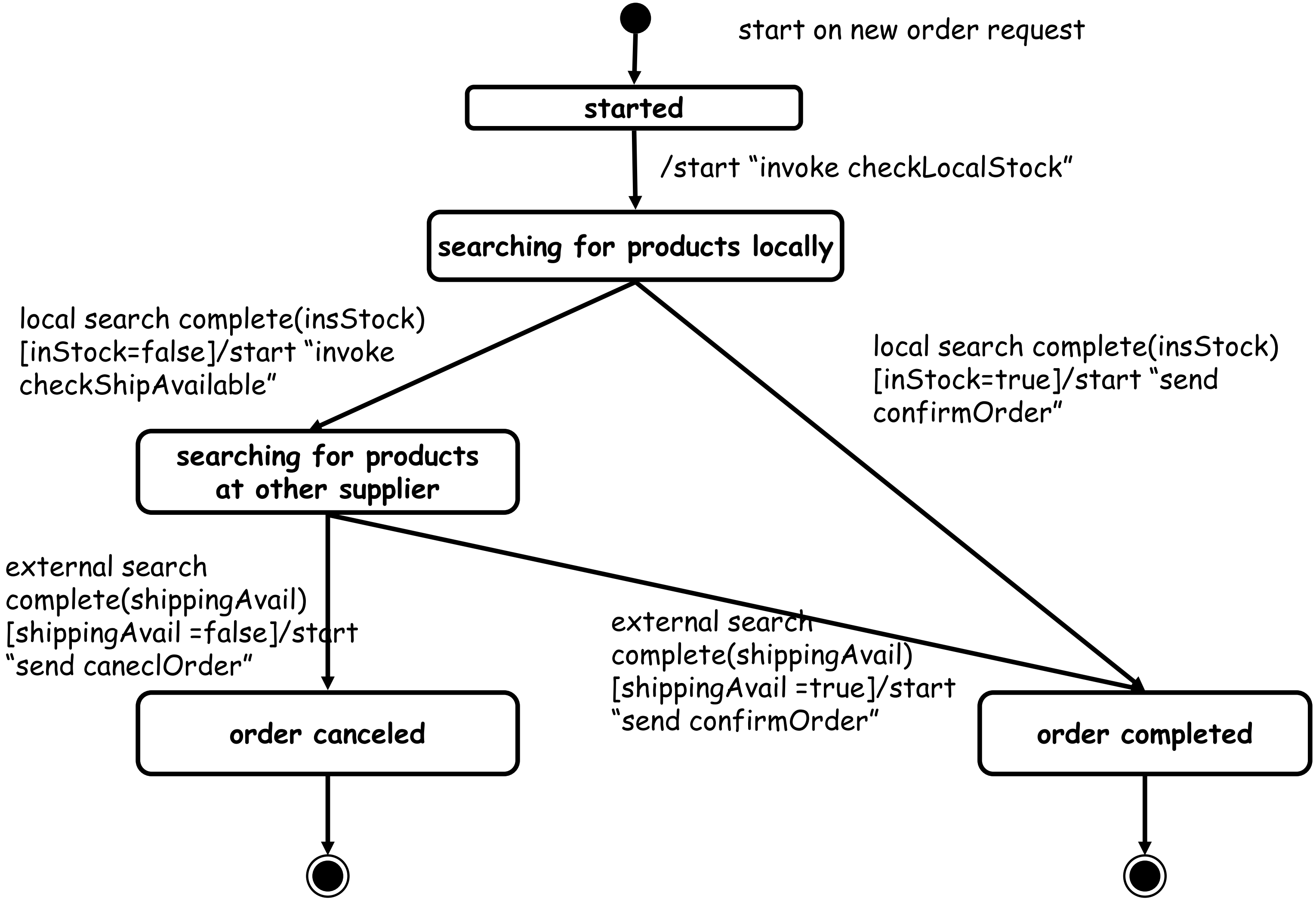
- Deals with how different services are composed into a coherent whole
- Alternative representations: UML, Statecharts, Petri nets, π-calculus, Activity hierarchies, Rule-based orchestration



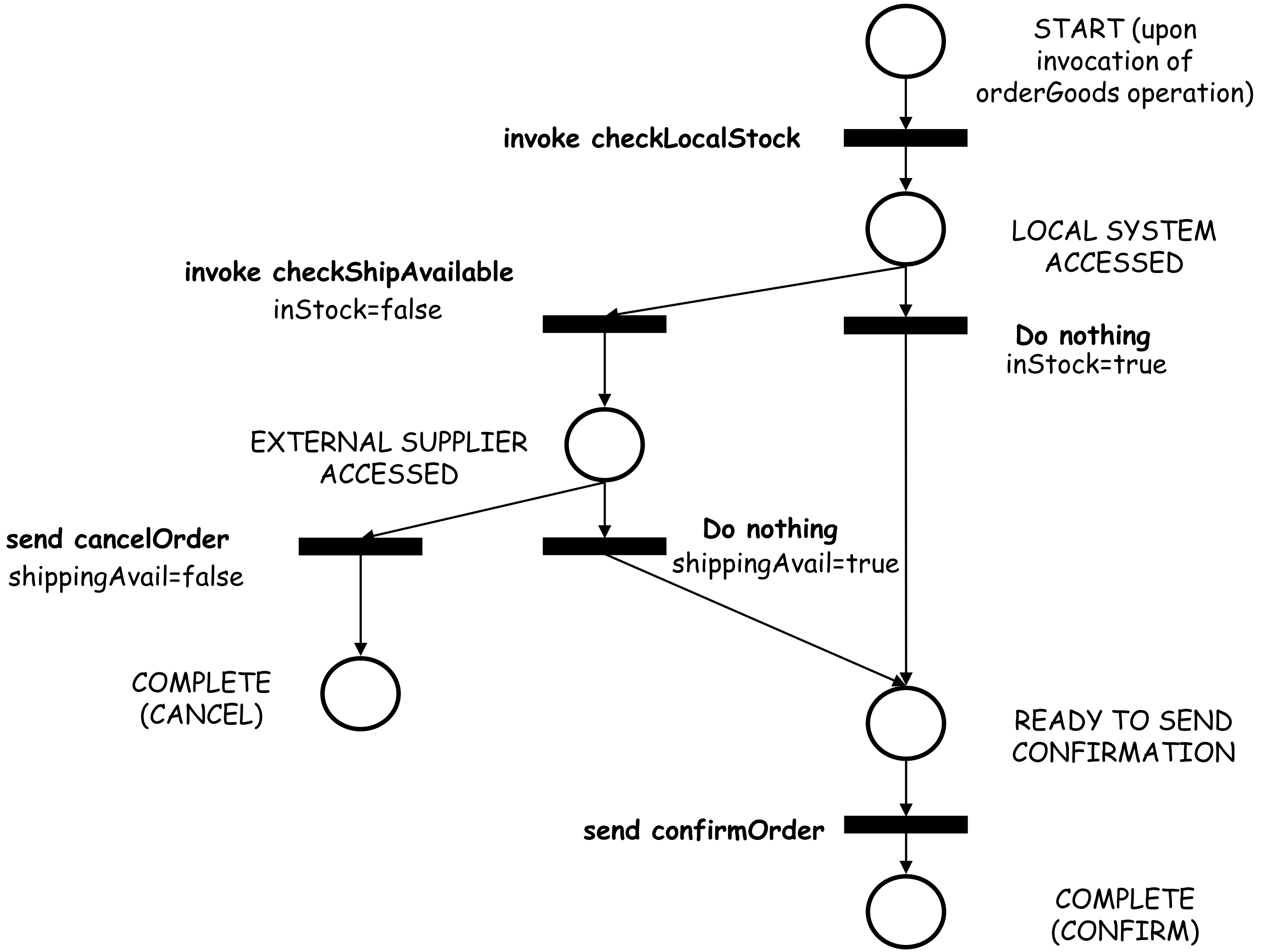
Orchestration model

- Activities in the orchestration model
  - Send activity (e.g., send cancelOrder)
    - Notification of messages to other web services
    - Nonblocking
  - Invoke activity (e.g., invoke checkLocalStock)
    - Invocations of synchronous (request/reply) operations offered by another web service
    - Blocking
  - Receive activity (e.g., receive orderGoods)
    - Receipts of messages, corresponding to component services invoking one-way or request/reply operations offered by the composite service
    - Blocking
  - Reply activity
    - If the received message is invoking a request/reply operation, then the composition schema will also include a reply activity, that will send a response to the invoking client

Statecharts


start on new order request
started
/start "invoke checkLocalStock"
searching for products locally
local search complete(insStock) [inStock=false]/start "invoke checkShipAvailable"
local search complete(insStock) [inStock=true]/start "send confirmOrder"
searching for products at other supplier
external search complete(shippingAvail) [shippingAvail =false]/start "send caneclOrder"
external search complete(shippingAvail) [shippingAvail =true]/start "send confirmOrder"
order canceled
order completed

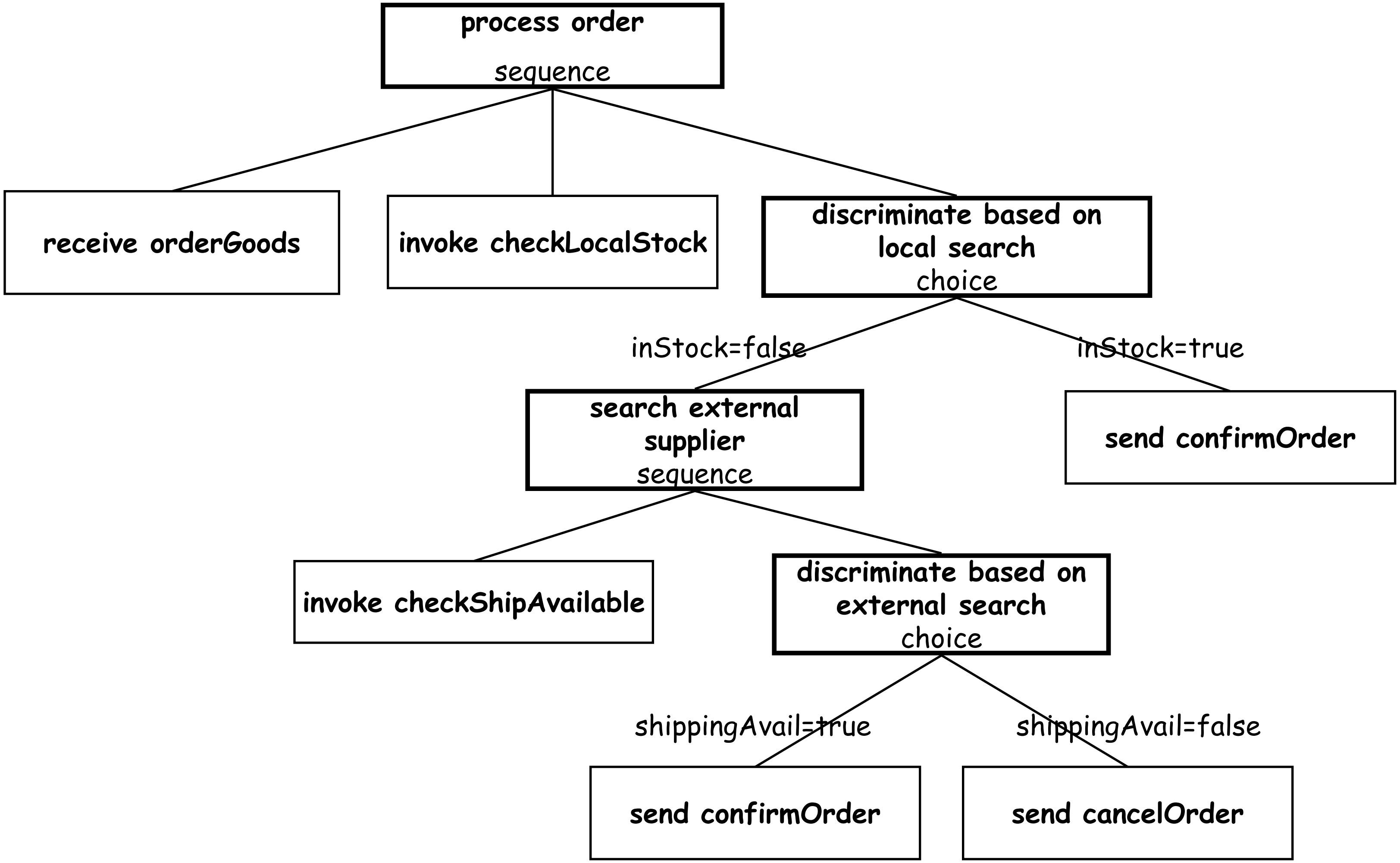
Petri nets


START (upon invocation of orderGoods operation)
invoke checkLocalStock
LOCAL SYSTEM ACCESSED
invoke checkShipAvailable
inStock=false
Do nothing
inStock=true
EXTERNAL SUPPLIER ACCESSED
send cancelOrder
shippingAvail=false
Do nothing
shippingAvail=true
COMPLETE (CANCEL)
READY TO SEND CONFIRMATION
send confirmOrder
COMPLETE (CONFIRM)

π-Calculus

- A=receiveOrderGoods.invokeCheckLocalStock
  B=[shippingAvail=false]sendCancelOrder+
    [shippingAvail=true]sendConfirmOrder
  C=invokeCheckShipAvailable.B
  Procurement=A.(([inStock=false]C) +
                  ([inStock=true]sendConfirmOrder)
)

Activity hierarchies



Rule-based orchestration

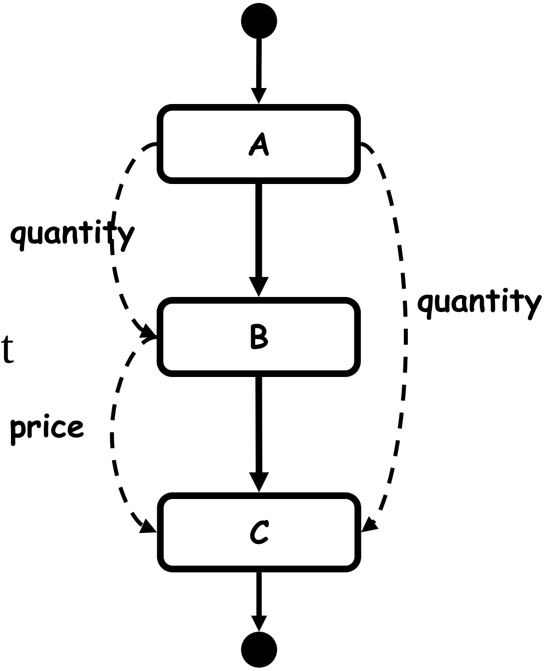
```
ON receive orderGoods
IF true
THEN invoke checkLocalStock;

ON complete(checkLocalStock)
IF (inStock==true)
THEN send confirmOrder;

ON complete(checkLocalStock)
IF (inStock==false)
THEN invoke checkShipAvailable;

ON complete(checkShipAvailable)
IF (shippingAvail ==true)
THEN send confirmOrder;

ON complete(checkShipAvailable)
IF (shippingAvail ==true)
THEN send cancelOrder;
```

Data types and data transfer model

- Service composition models need explicit ways to define and access data
- Data types
  - **Control flow data** (process variables)
    - Used to evaluate branching conditions, and in general those accessed by the composition engine to determine how the execution should proceed
    - Usually restricted to a few basic types such as string, integer, or real
  - **Application-specific data**
    - The parameters sent or received as part of message exchanges
    - Application data are more complex and involve more sophisticated data types
    - Can be treated as a black box (e.g., exchanging only URLs) or can be explicit by including appropriate data definitions as part of the composition schema

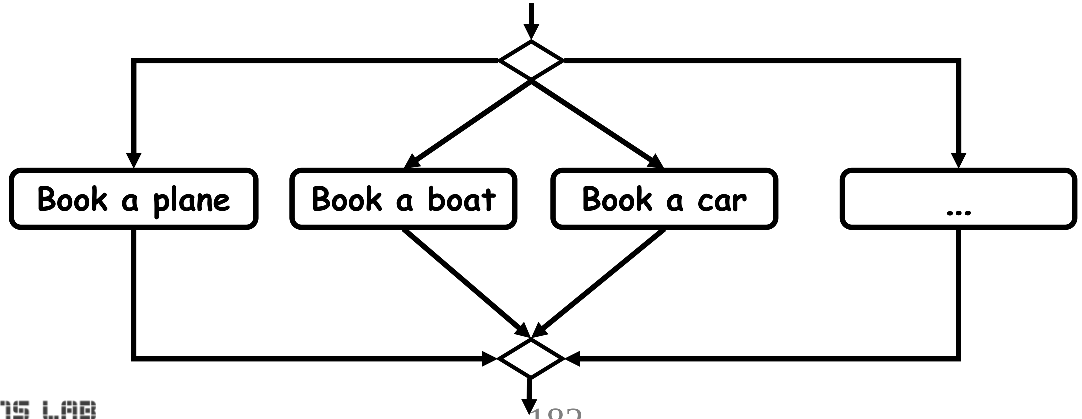
Data types and data transfer model

- **Data transfer**
  - Refers to how data is passed from one operation invocation to the next
  - Blackboard approach
    - Analogous to conventional programming language
    - All data involved in the execution of the composite service is explicitly named and listed
    - Each composition instance has its own blackboard
  - Explicit data flow approach
    - Makes the data flow between activities an explicit part of the composition
    - Used in several workflow engines like MQSeries Workflow and BioOpera
    - More flexible and richer than the blackboard approach, but more complex



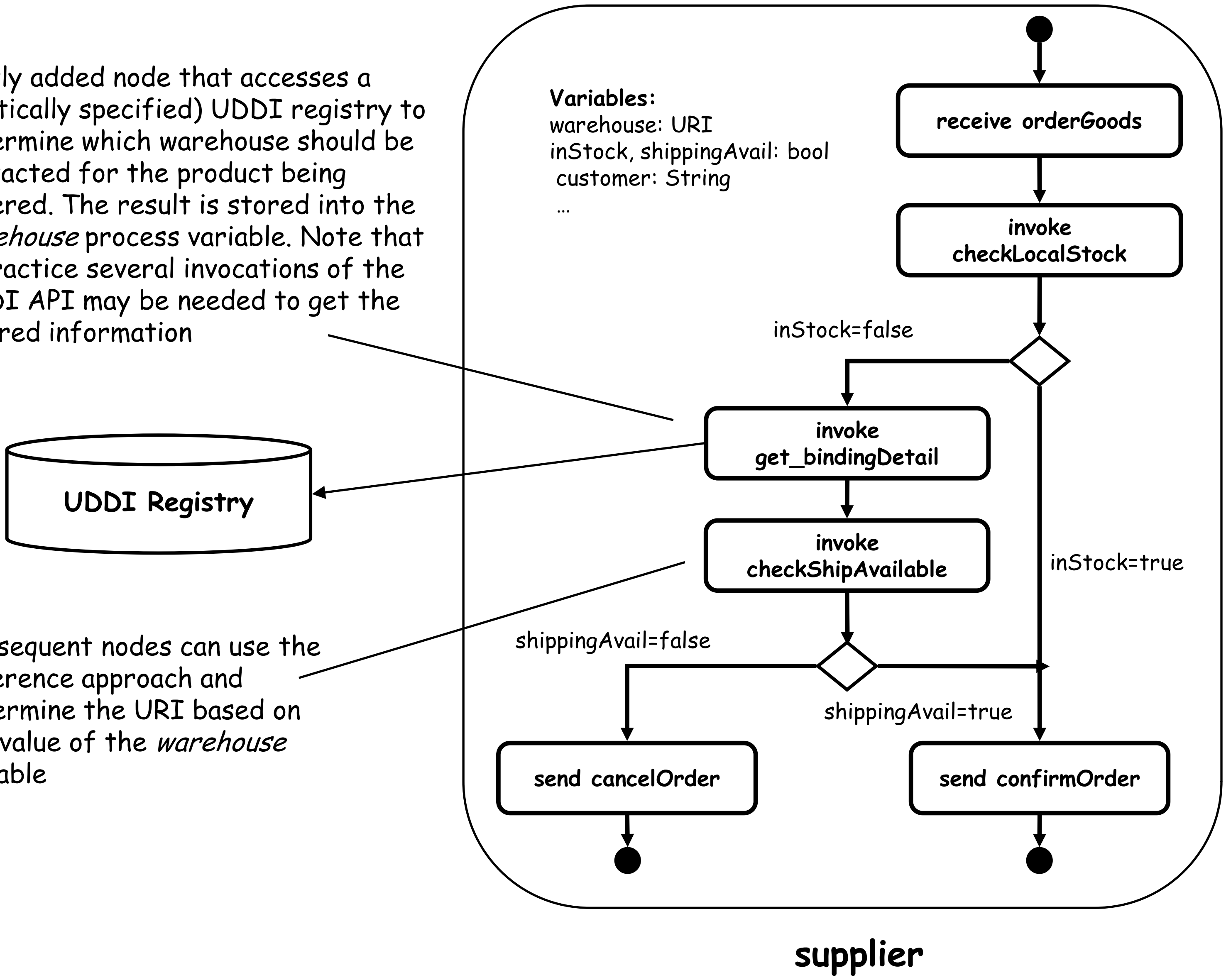
Service selection

- To execute the composition logic, the engine must be informed which specific service (e.g., which URL) is to be the target of a message
- Static binding
  - Hardcode the URL as part of the composite service specification
  - Not robust to changes of the service URI
- Dynamic binding by reference
  - Activities determine the URIs of the services to be invoked from the value of specified process variables
- Dynamic binding by lookup
  - Allows the definition, for each activity, of a query whose result will be used to determine the service to be invoked, to be executed on some directory
  - May cause multiple URIs to be returned
- Dynamic operation selection
  - Does not explicitly specify the operation to be invoked. Instead, the operation is selected at run-time, along with the serviced
  - The signature of the operation to be invoked may vary with the selected service
  - Very difficult to implement



Service selection by reference

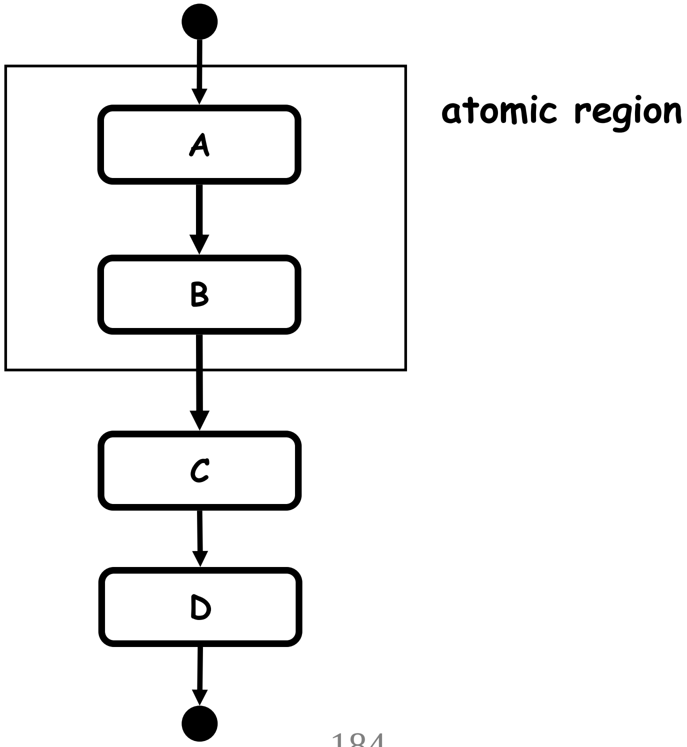
Newly added node that accesses a (statically specified) UDDI registry to determine which warehouse should be contacted for the product being ordered. The result is stored into the *warehouse* process variable. Note that in practice several invocations of the UDDI API may be needed to get the desired information



Subsequent nodes can use the reference approach and determine the URI based on the value of the *warehouse* variable

Transactions

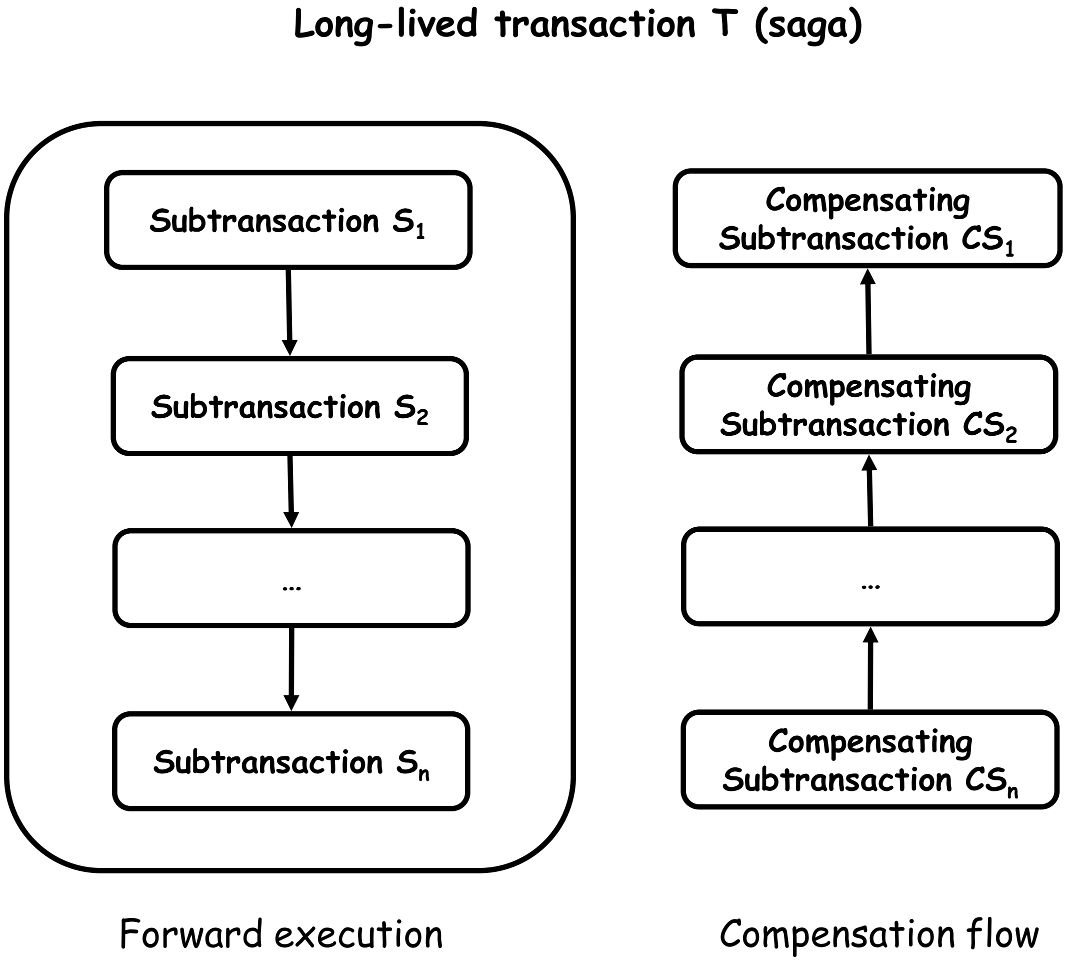
- Enables the definition of atomic regions within an orchestration schema
- The atomic region typically surrounds a set of activities that should exhibit the all-or-nothing property
- Used when the composite service wants to explicitly define the business logic executed to perform compensation



Compensation

- Allows long-lived transactions to be broken down into **a set of sub-transactions to be executed in some predefined order**
  - The sub-transactions have the ACID properties
  - Each **subtransaction S_k is associated with a compensating transaction CS_k,** whose execution semantically undoes the effect of the sub-transaction
  - A rollback is performed by aborting all active sub-transactions, and by then compensating for committed sub-transactions in the reverse order of execution
  - The development of the compensation logic is left entirely to the designer of the composition
- The better way
  - Placing the burden for implementing the compensation on the web service developer, i.e., on the component side instead of the composition side
  - A component includes the definition of the transaction and compensation logic (e.g., operation o can be compensated for by operation c)
  - Composition designers can simply invoke operation c whenever they need to compensate for o

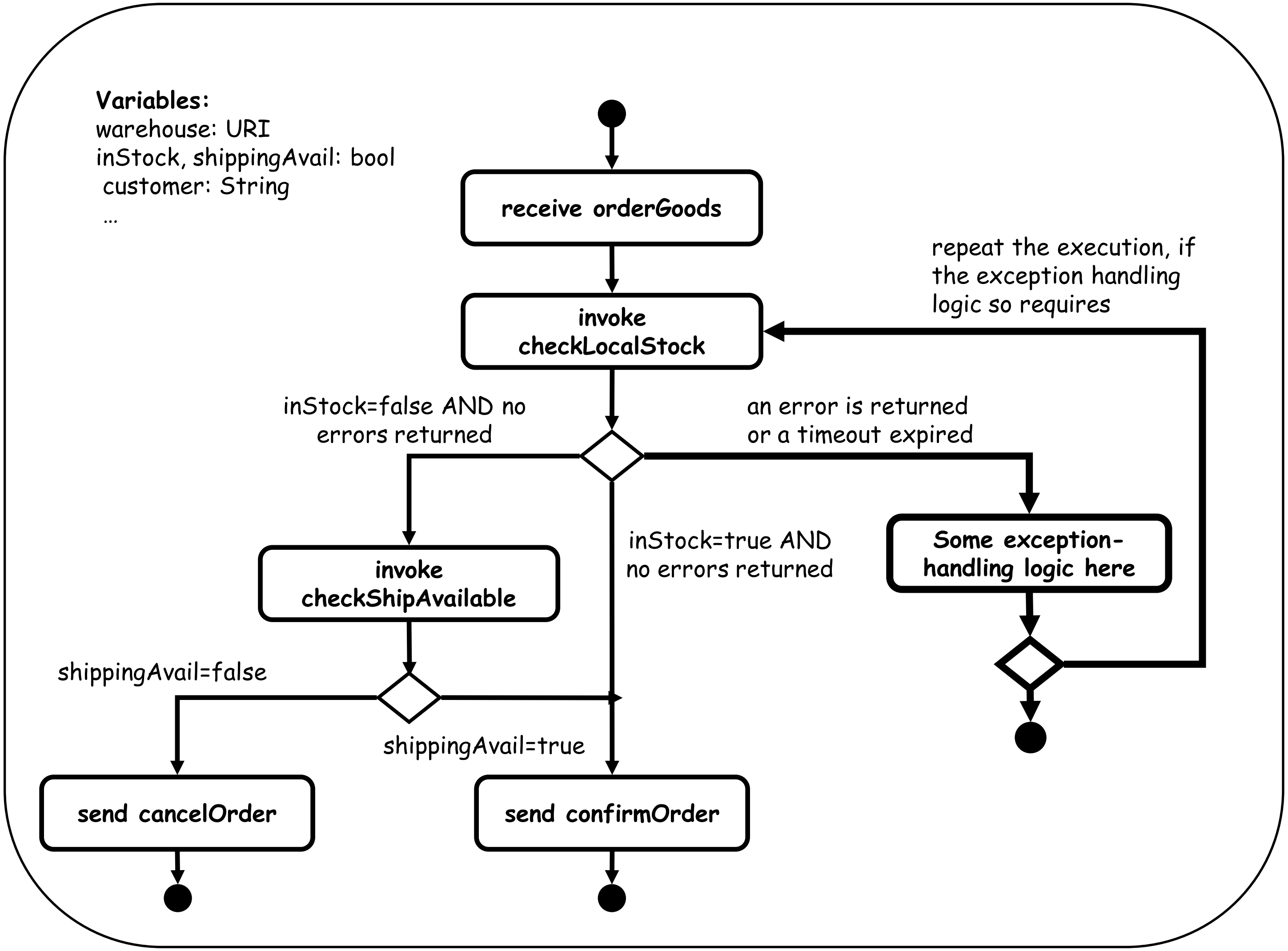
Compensation


Long-lived transaction T (saga)
Subtransaction S_1
Subtransaction S_2
...
Subtransaction S_n
Compensating Subtransaction CS_1
Compensating Subtransaction CS_2
...
Compensating Subtransaction CS_n
Forward execution
Compensation flow

Exception handling

- Exception refers to a deviation from the expected or desired execution of the composition
- Flow-based approaches
  - At the end of each operation invocation, the result is tested for errors, and appropriate actions are taken if an error has been detected
  - May require the definition of timeouts associated with activities
- Try-catch-throw approaches
  - Similar to what is done in Java
  - Associate exception-handling logic to an activity or to a group of activities
  - Enables the clear separation of the normal and exceptional logic
  - Useful if the orchestration model is hierarchically structured
- Rule-based approaches
  - The exception handling logic is specified by means of ECA rules, where the event defines the exceptional event
  - Applicable only if the number of rules is very small

Example of an flow-based approach

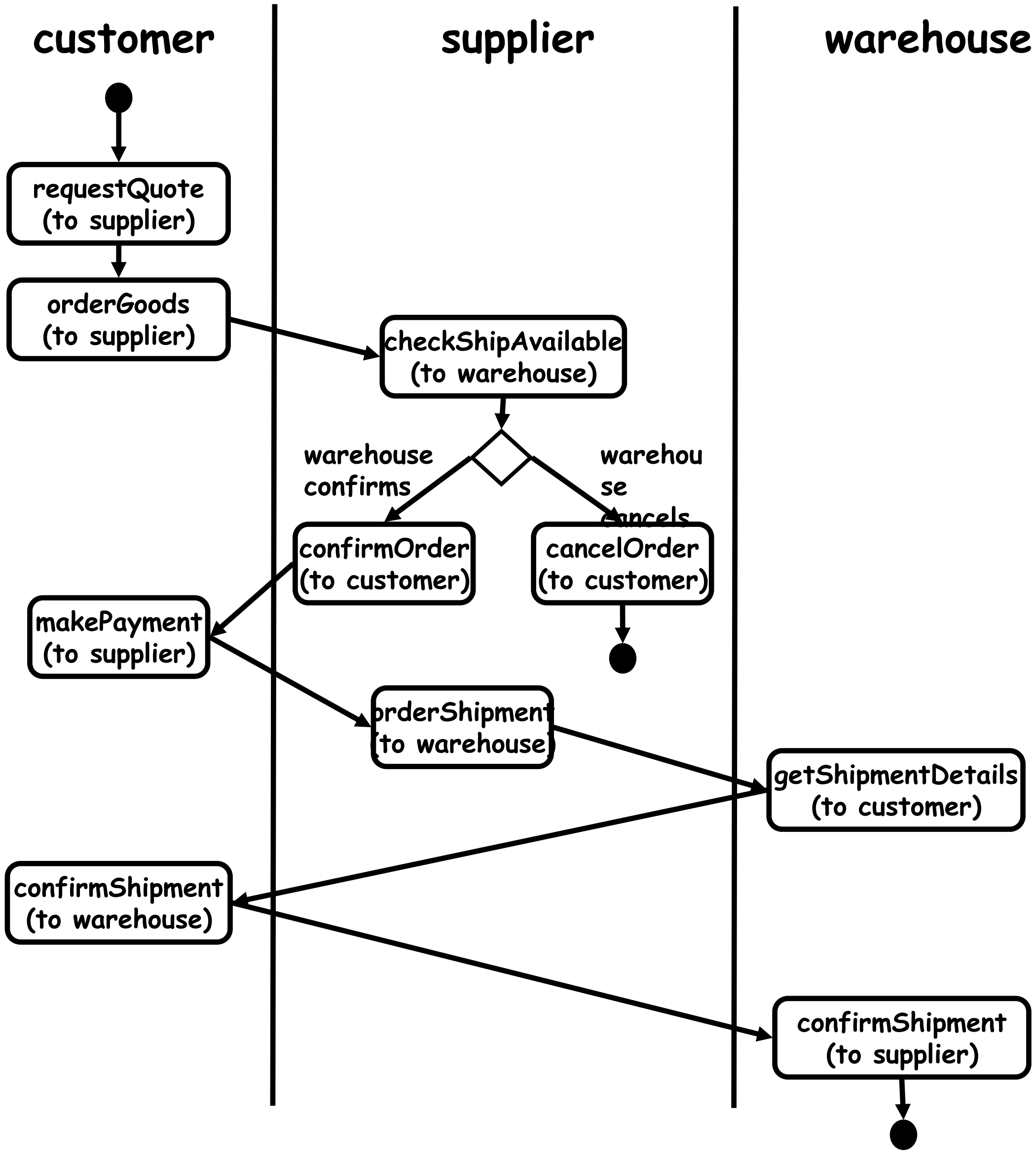

Variables:
warehouse: URI
inStock, shippingAvail: bool
customer: String
...
receive orderGoods
invoke checkLocalStock
repeat the execution, if the exception handling logic so requires
inStock=false AND no errors returned
an error is returned or a timeout expired
invoke checkShipAvailable
inStock=true AND no errors returned
Some exception-handling logic here
shippingAvail=false
shippingAvail=true
send cancelOrder
send confirmOrder

supplier

Coordination protocols and composition schemes

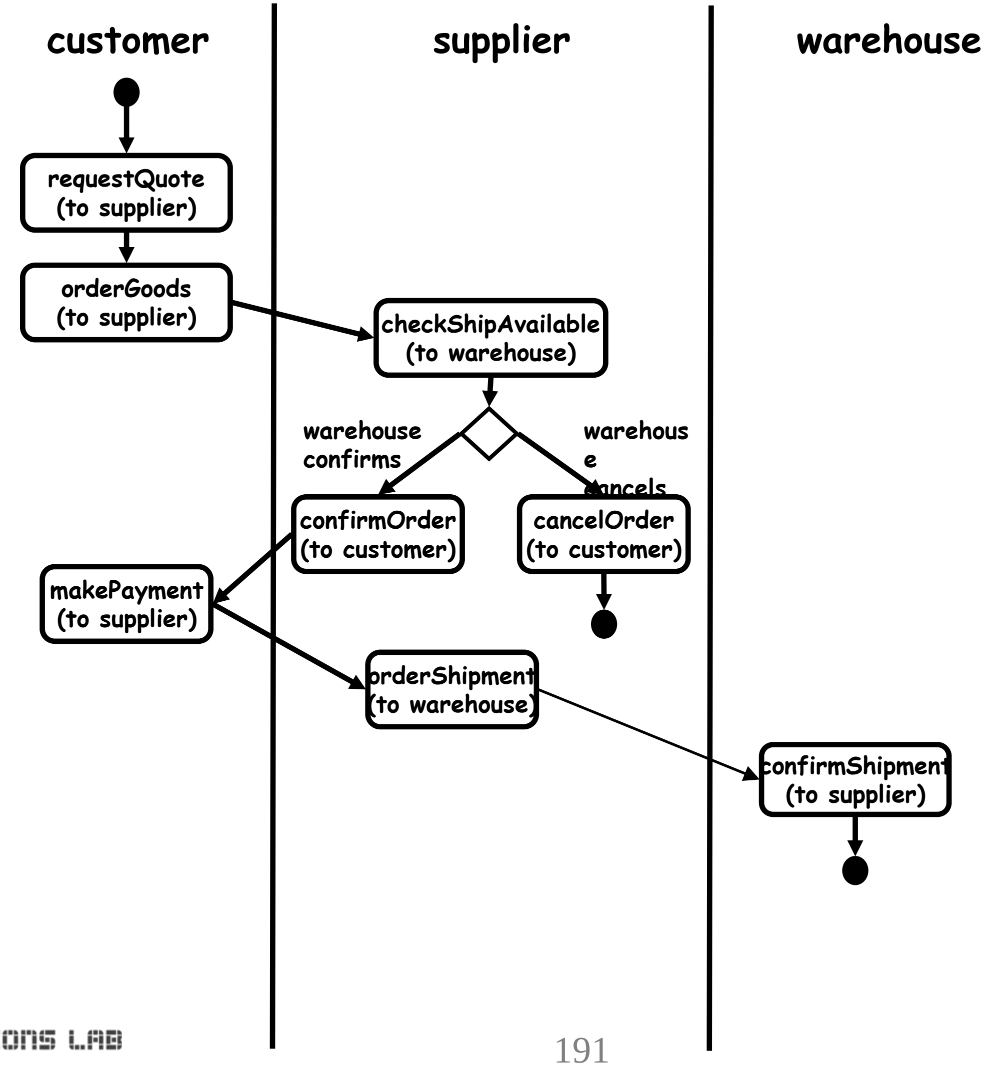
- The definition of a coordination protocol **imposes constraints on the composition schema** of web services implementing the protocol logic
- The composition schema must include activities that receive and send messages as prescribed by the protocol, and in the appropriate order
- How can protocol definitions be used to **drive the design of composition schemes** that send and receive the messages in a way that is compliant with the protocol?

Example


customer
supplier
warehouse
requestQuote (to supplier)
orderGoods (to supplier)
checkShipAvailable (to warehouse)
warehouse confirms
warehou se cancels
confirmOrder (to customer)
cancelOrder (to customer)
makePayment (to supplier)
orderShipment (to warehouse)
getShipmentDetails (to customer)
confirmShipment (to warehouse)
confirmShipment (to supplier)

Developing a WS for the supplier role

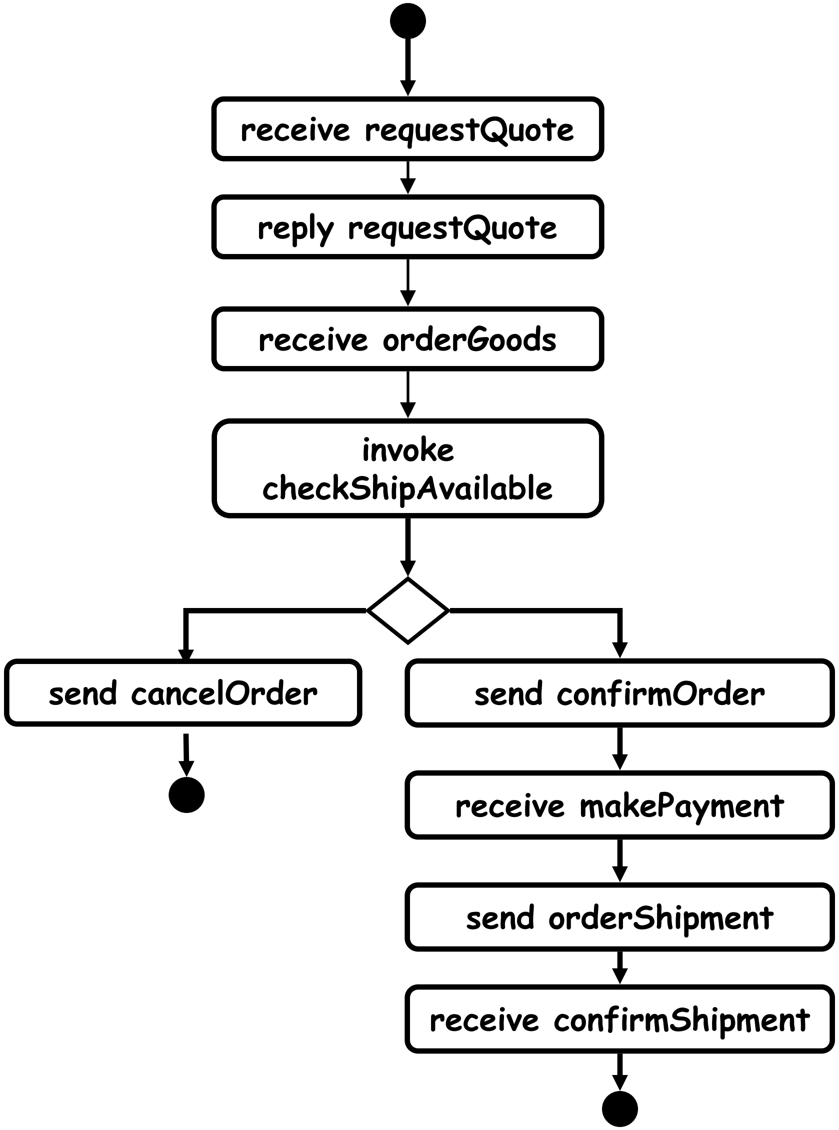
- Create the role-specific view of the protocol
  - Includes all the message exchanges that involve a certain role


customer
supplier
warehouse
requestQuote (to supplier)
orderGoods (to supplier)
checkShipAvailable (to warehouse)
warehouse confirms
warehous e cancels
confirmOrder (to customer)
cancelOrder (to customer)
makePayment (to supplier)
orderShipment (to warehouse)
confirmShipment (to supplier)

Developing a WS for the supplier role

- Switch **from the role-specific view** of the protocol **to the definition of a process** that exchanges messages as prescribed by role-specific view
  - A skeleton of a process that includes all the activities that send and receive messages as prescribed by the protocol
- Steps
  - Request/reply operations invoked by a role R on the supplier are mapped into a receive and a reply activity
    - requestQuote -> receive requestQuote and reply requestQuote
  - One-way operations invoked by another role on the supplier are modeled as receive activities
    - orderGoods -> receive orderGoods
  - One-way operations invoked by the supplier are modeled as send activities
    - confirmOrder -> send confirmOrder
  - Request/reply operations initiated by the supplier are modelled as invoke activities
    - checkShipAvailable -> invoke checkShipAvailable
  - The other constructs are mapped as is into the process skeleton

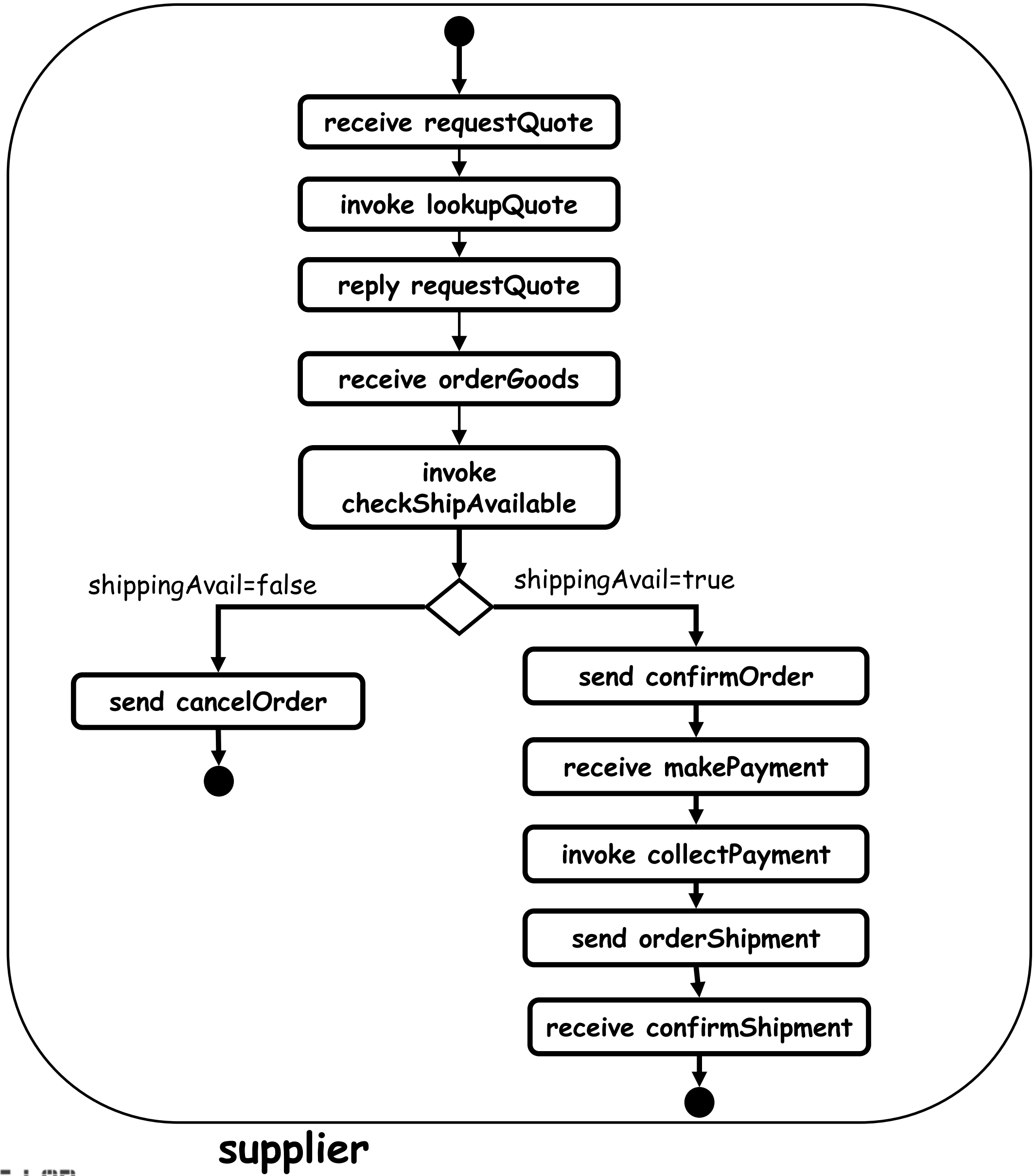
Developing a WS for the supplier role



Abstract process

- The process skeleton (also called abstract process or public process) is essentially a dual representation of the role-specific protocol view
- describes a **protocol** -> Does not include any internal and confidential business logic
- Not executable
- Make it easy to understand how composition is constrained by the protocol and to define a composition schema that implements a protocol
- **Executable process**
  - Can be obtained by specifying other activities to invoke other services as well as defining anything left undetermined in the abstract process, such as **branching conditions**, **data assignments**, and **data transfer rules**
  - Private and can be enacted by a composition engine

Sample of an executable process



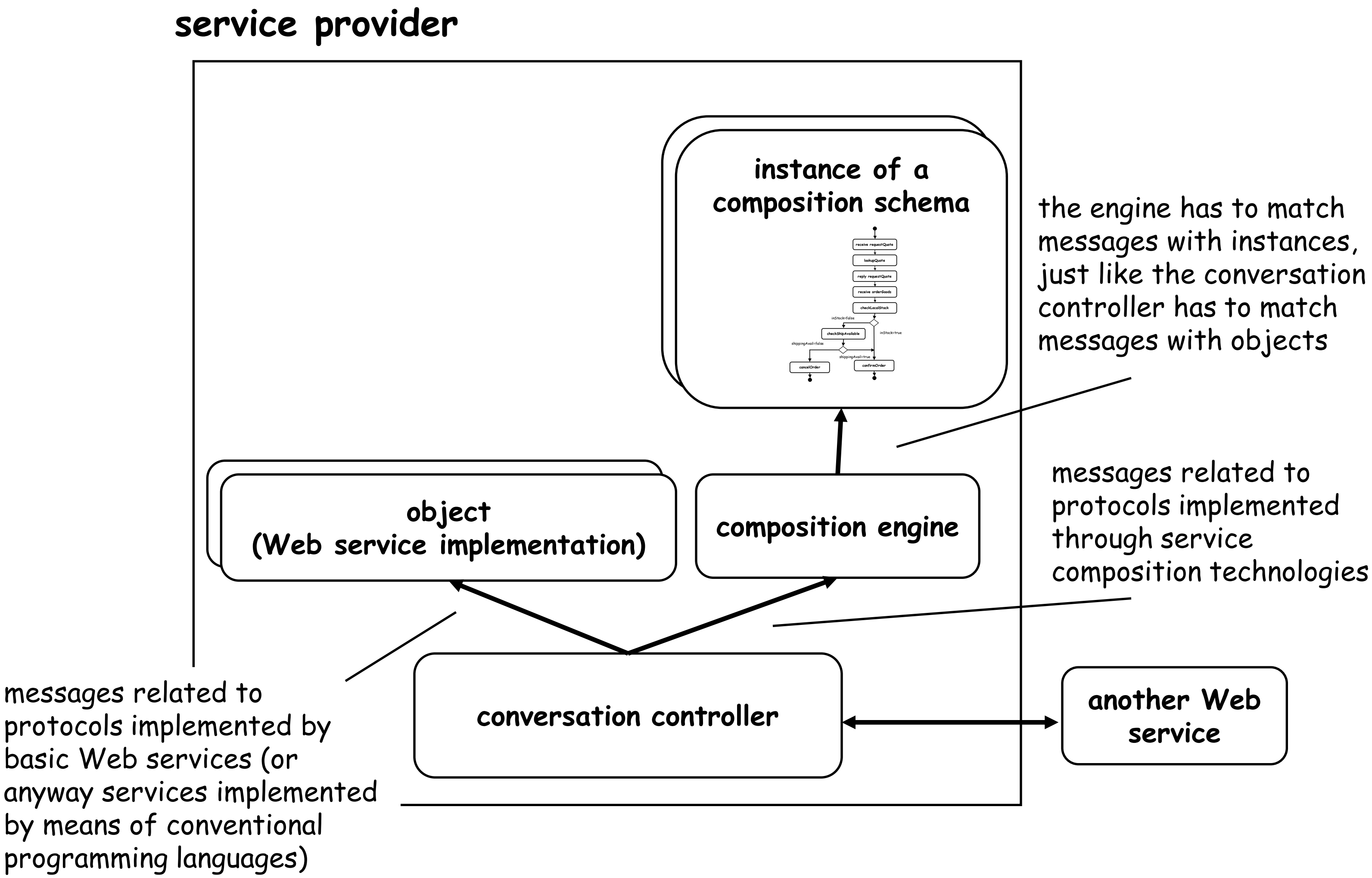
On the composition and coordination

- A web service may need to support several protocols and carry on many concurrent conversations
- How to establish dependencies between the different protocols that are independently defined?
  - e.g., to define that a manufacturer is to be contacted whenever the order comes from premium customers and the warehouse does not have goods in stock
  - A hot research topic in the very near future
- Yet another approach
  - One can first design the internal process and then generate the corresponding abstract process and role-specific protocol

Conversation controllers and composition engines

- Service composition architectures following an engine-based approach are faced with a **conversation routing problem**
- The conversation controller verifies protocol compliance and routes messages to the composition engine
- The engine has to figure out the composition instance to which a message is directed
- Two cases
  - If the conversation controller and the SOAP router leave header information in the message when routing it to the engine, then the coordination context can be used to determine the target instance
  - If the conversation controller strips header information from the SOAP message and just delivers the payload, then the engine has to find some way to correlate messages with instances
    - Have the composition schema explicitly include correlation information by defining the logic by which messages can be associated with composition instances, based on the message parameters (e.g., orderID)

Conversation controllers and composition engines



service provider

instance of a composition schema

object (Web service implementation)

composition engine

conversation controller

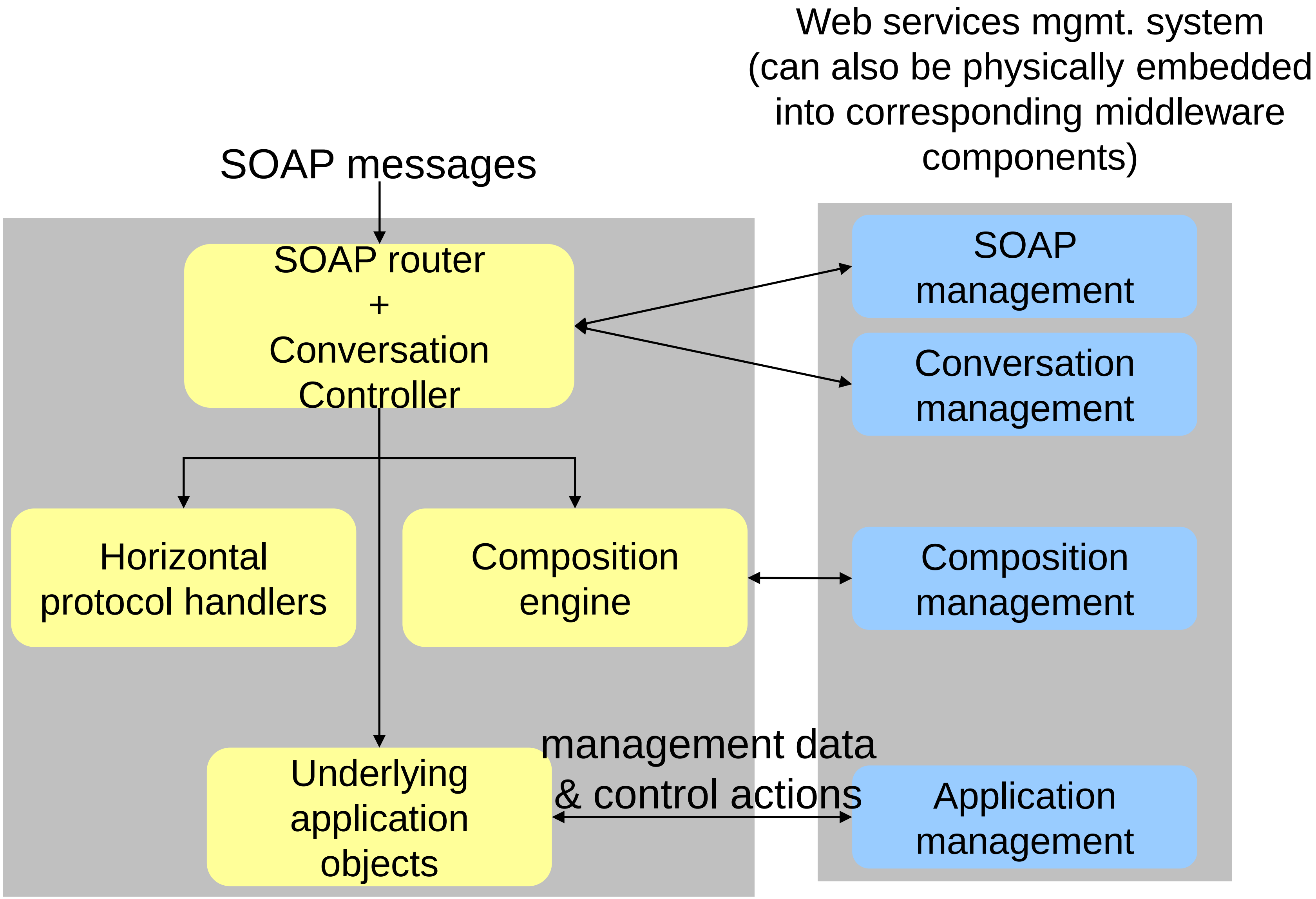
another Web service

the engine has to match messages with instances, just like the conversation controller has to match messages with objects

messages related to protocols implemented through service composition technologies

messages related to protocols implemented by basic Web services (or anyway services implemented by means of conventional programming languages)

Web services management architecture



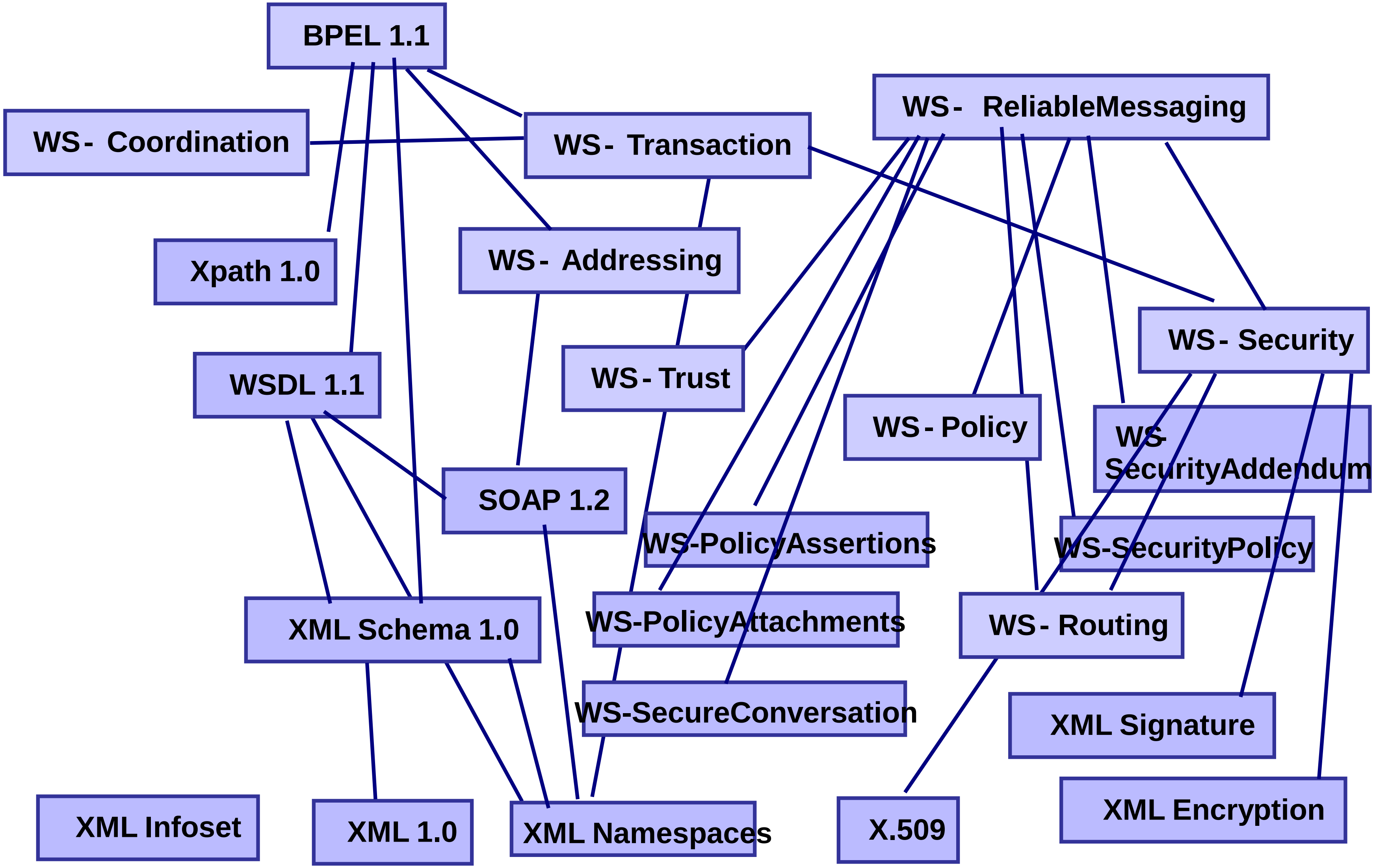
Business processes

- Business process
    - A collection of activities performed by human users or software applications that together constitute the different steps to be completed to achieve a particular business objective
    - e.g., travel expense reimbursement, hiring a new employee
    - Distinguished by being possibly long-running, involving multiple autonomous participants, and having correctness and completion guarantees
- Process model
    - describes the structure of a business process in the real world
    - defines all possible paths through the business process, including the rules that define which paths should be taken and all actions that need to be performed
- Process instance: the instances that are created from process models
- Can be described in two ways
    - Executable BP
        - models the actual behavior of a participant in a business interaction
    - Abstract BP
        - uses process descriptions that specify the mutually visible message exchange behavior of each of the parties involved in the protocol, without revealing their internal behavior
        - cannot be executed

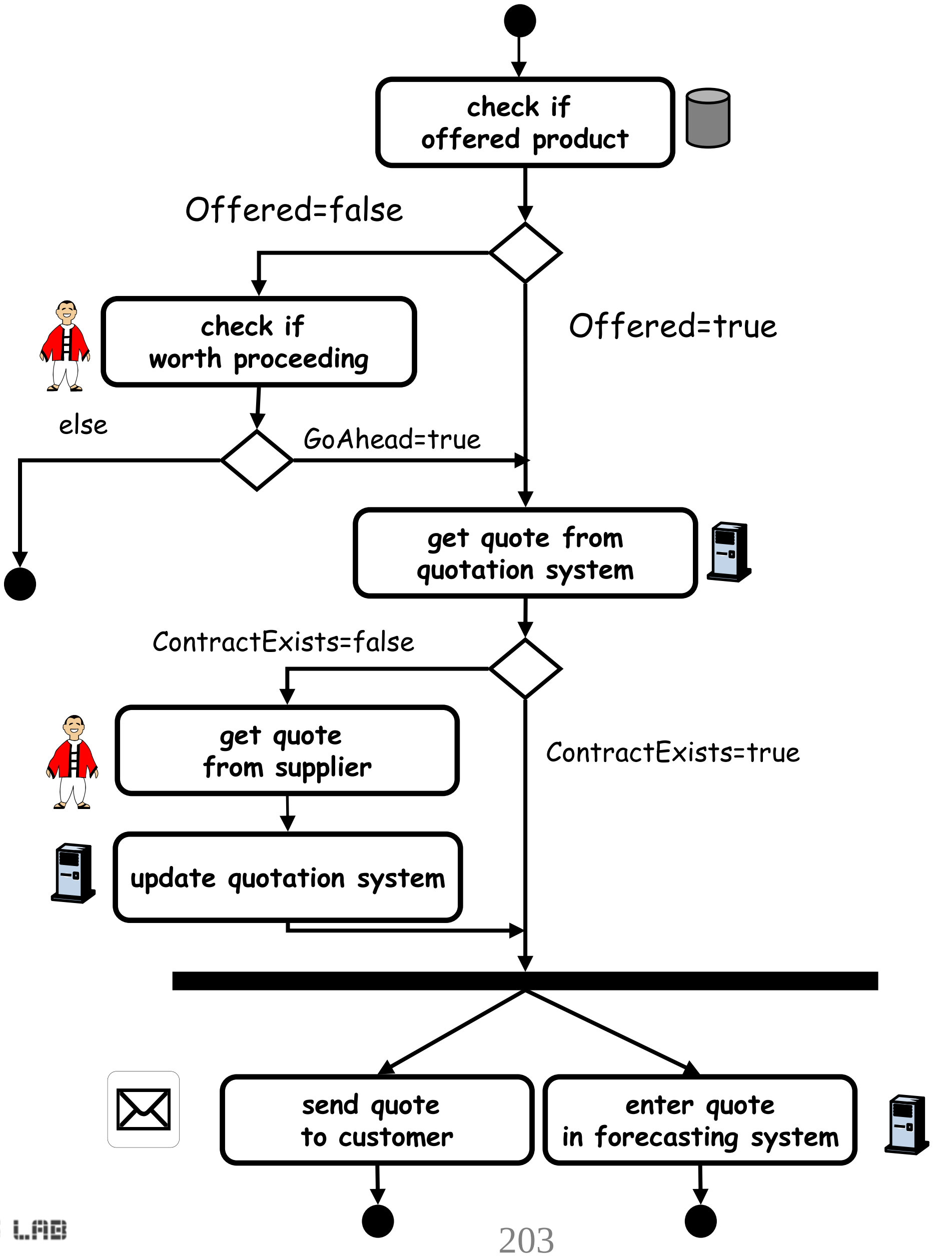
BPEL4WS

- Business Process Execution Language for Web Services
- serve as both an implementation language for executable processes and a description language for nonexecutable business protocols
- defines a model and a grammar for describing how multiple web service interactions among the process's participants, termed partners, are coordinated to achieve a business goal, as well as the state and the logic necessary for the coordination to occur
- Interactions with each partner occur through lower-level web service interfaces, as might be defined in WSDL
- can define mechanisms for dealing with exceptions and processing faults, including how individual or composite process components are to be compensated when exceptions and faults occur or when a partner requests an abort
- An executable BPEL4WS is a new web service composed of existing services
  - The interface of the composite service is a collection of WSDL portTypes
- can be translated into and from UML
- History
  - Initially proposed in July 2002 by BEA, Microsoft, and IBM
  - WSBPEL TC: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel
  - Version 1.1 released in May 2003: http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/
  - Now being revised for Version 1.2

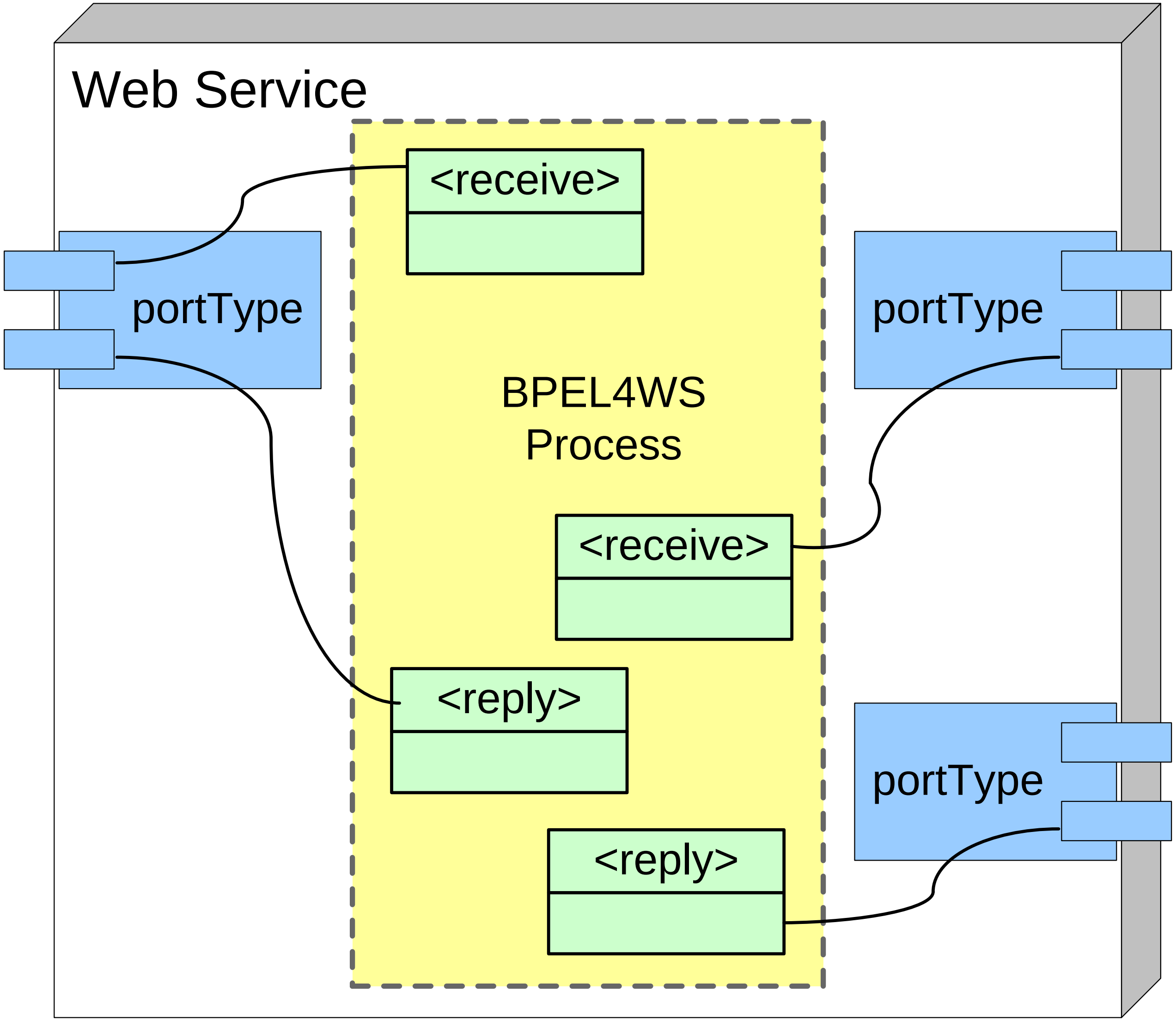
WSBPEL관련 표준 간의 상호의존성



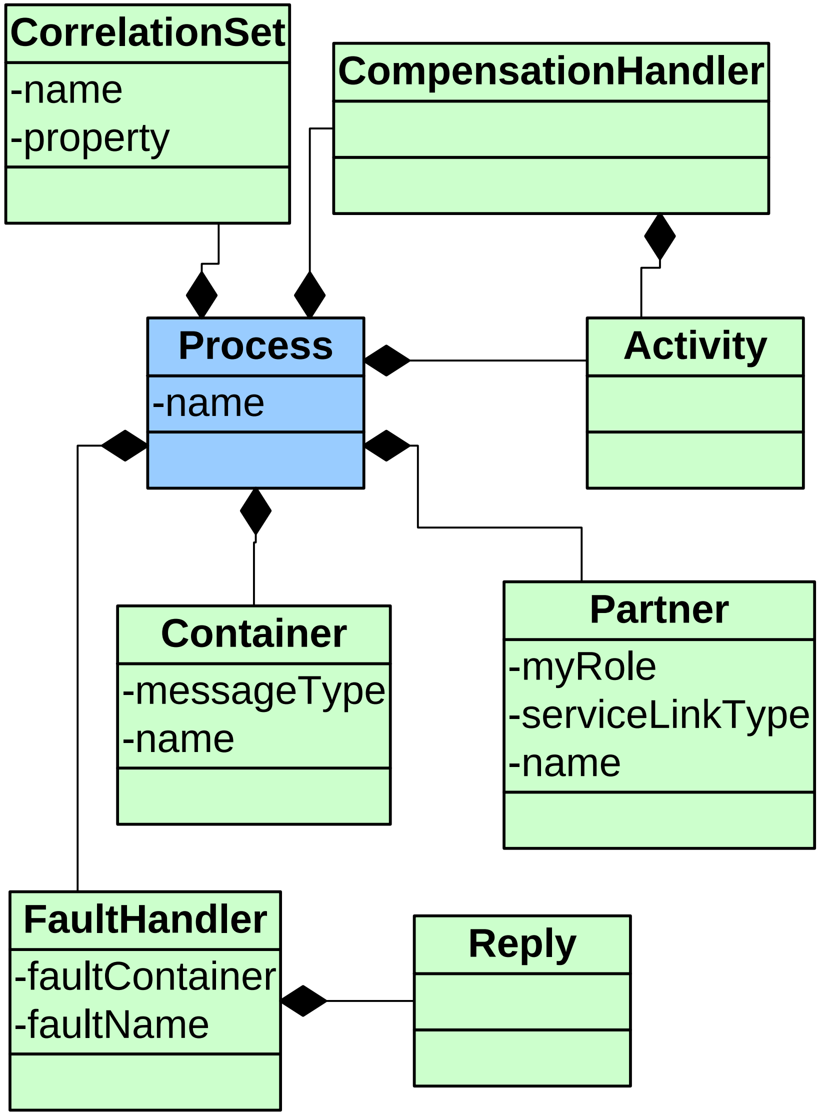
Sample BP specification in UML



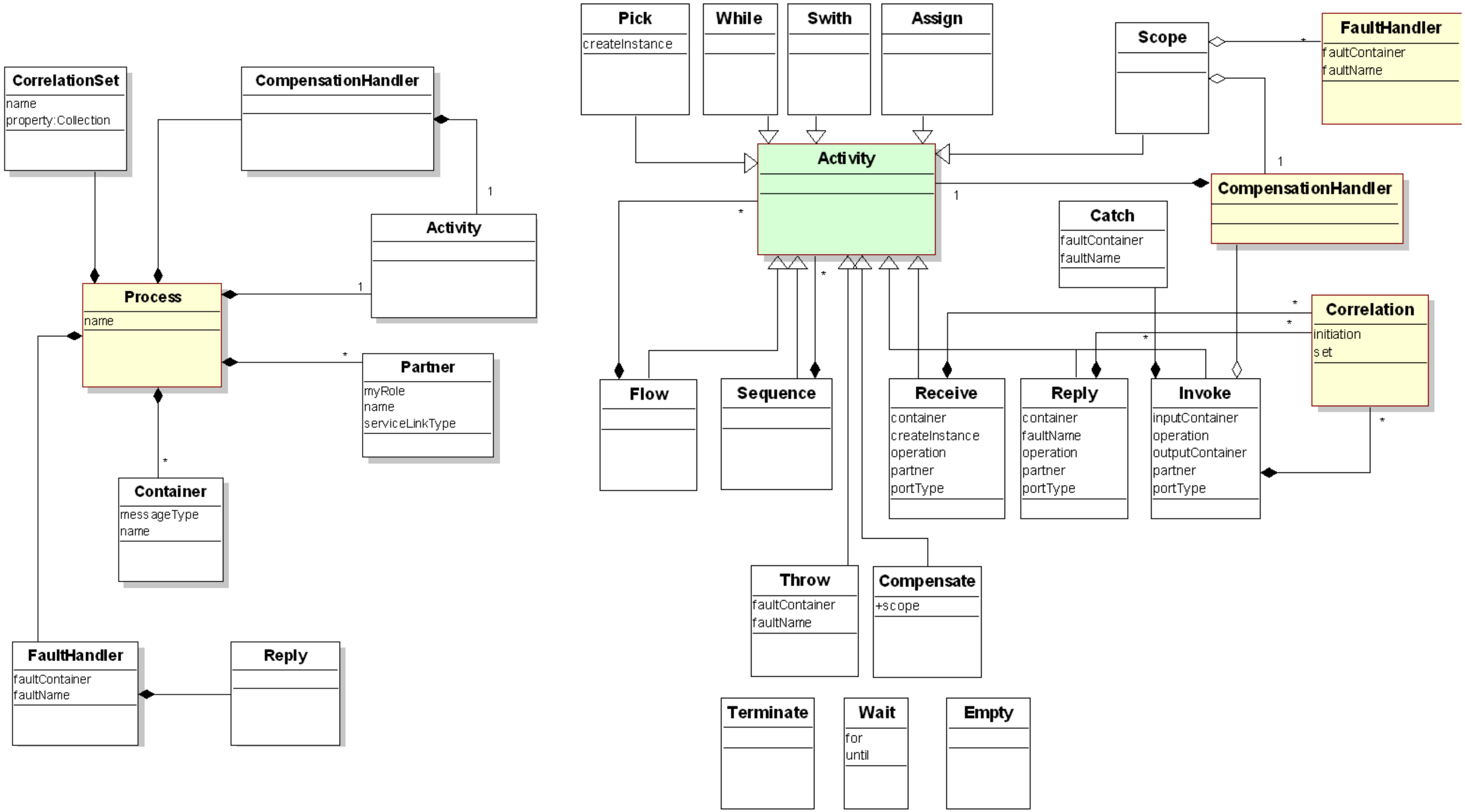
A BPEL4WS process as a composite web service



BPEL4WS Metamodel



BPEL4WS metamodel (full)



BPEL4WS structure

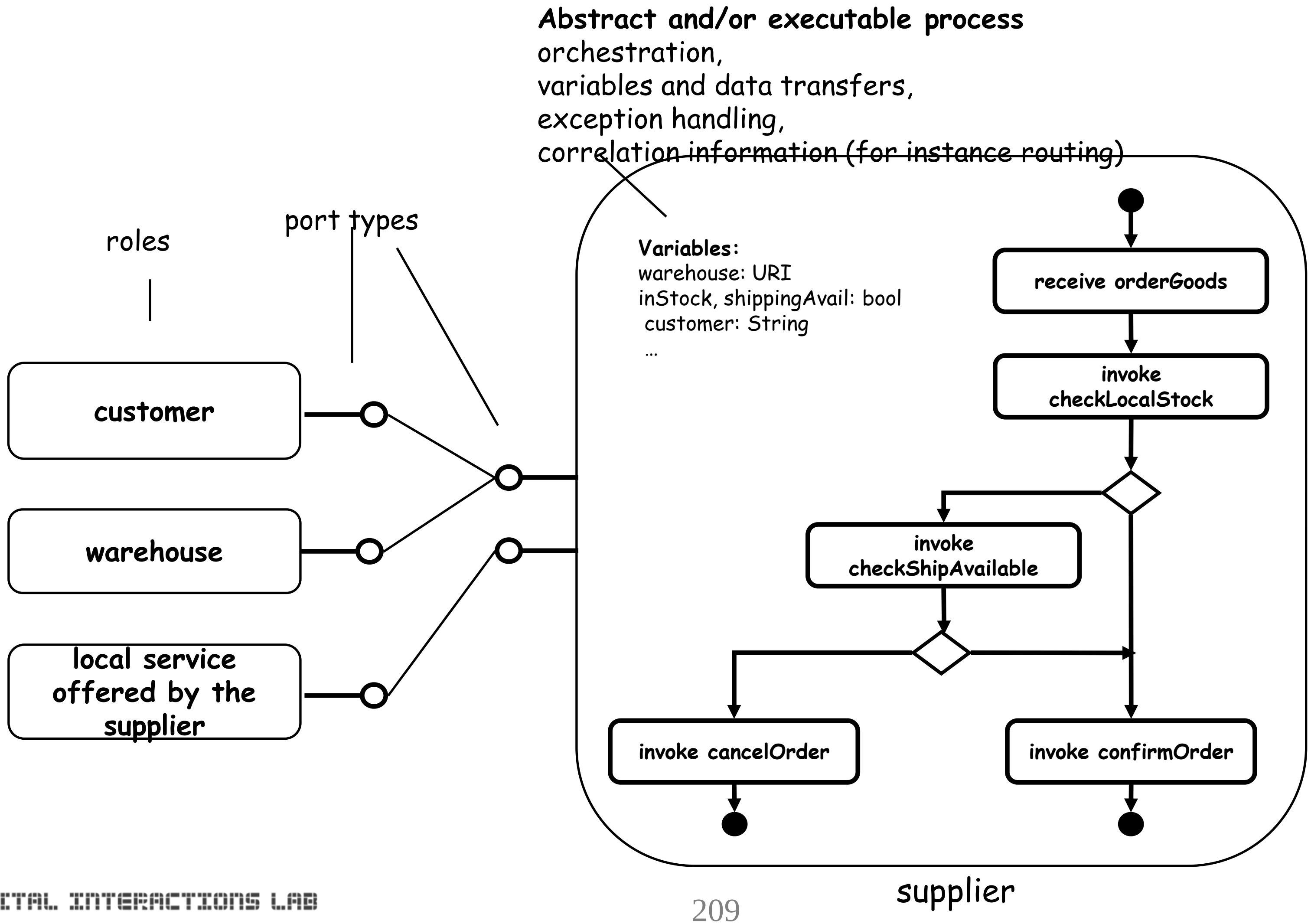
- partners: a list of the web services invoked as part of the process
- containers: the data containers used by the process, providing their definitions in terms of WSDL message types
- variables: the variables that are used and flow through the process
- faultHandlers: the exception handling routines
- compenstationHandlers: compensation to perform when a transaction rollback occurs
- eventHandlers: routines for handling external, asynchronous events
- correlationSets: precedences and correlations among web service invocations that cannot be expressed as part of the main process logic
- main process logic: a series of nested control flow structures that combine primitive activities into more complex algorithms
  - sequence, while, switch, pick, flow

Atomic actions in BPEL4WS

- invoke: invoking a specific web serivce
- receive: a server waiting to receive a message from a client, which would invoke the server's service
- reply: generating the response to an invocation request
- wait: waiting either for a deadline or some amount of time
- assign: assigning a value, which might have come from a received message, to a variable
- throw: indicating that something went wrong
- terminate: terminating an entire service instance
- empty: doing nothing

BPEL

- Assumes that both the process and the partners that interact with it are described as WSDL abstract services (port types and operations)


Abstract and/or executable process
orchestration,
variables and data transfers,
exception handling,
correlation information (for instance routing)
roles
port types
customer
warehouse
local service offered by the supplier
Variables:
warehouse: URI
inStock, shippingAvail: bool
customer: String
...
receive orderGoods
invoke checkLocalStock
invoke checkShipAvailable
invoke cancelOrder
invoke confirmOrder
supplier

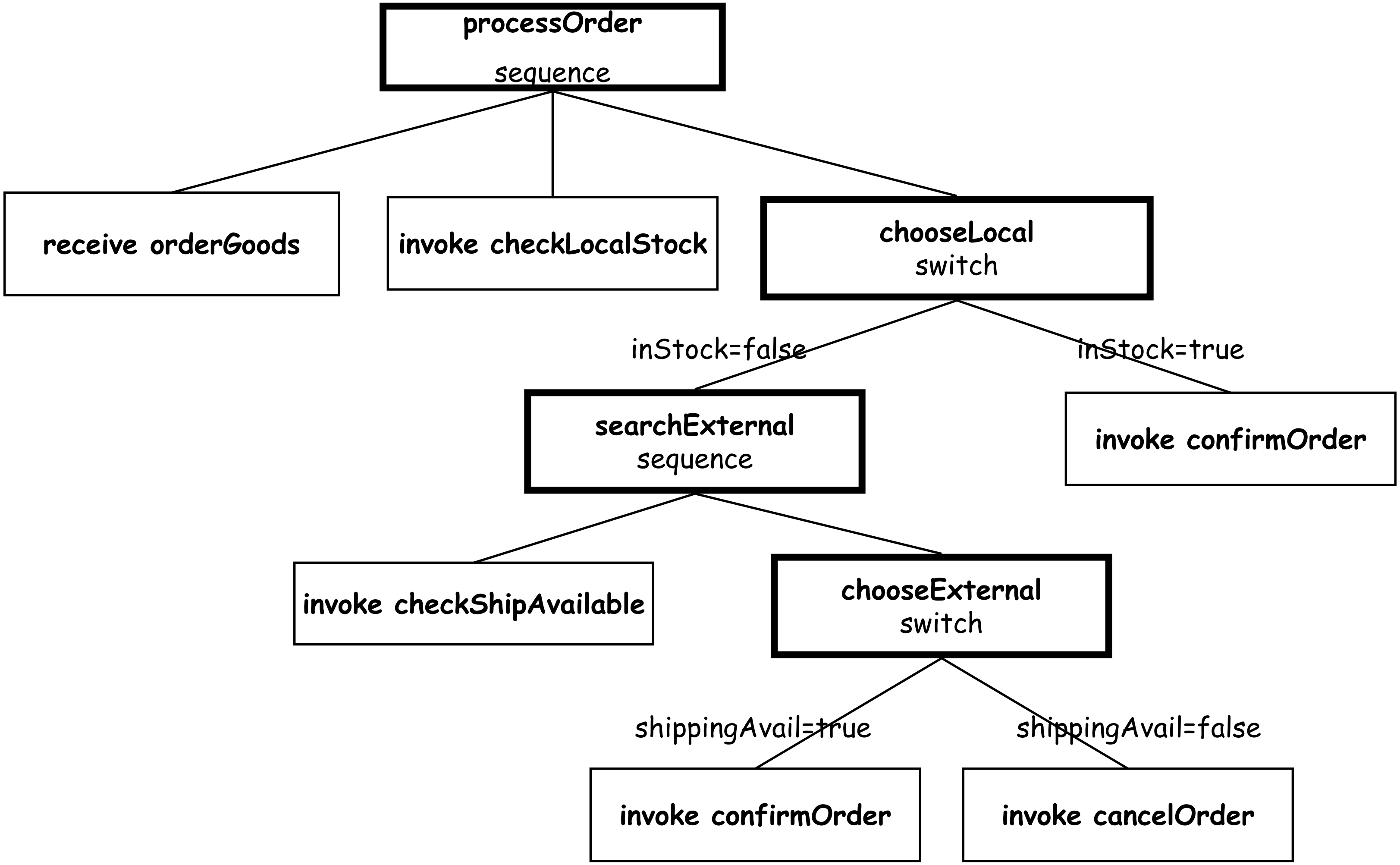
Component model

- BPEL has a fine-grained component model, consisting of **activities**, which can be **basic** or **structured**
  - Structured activities are used for defining the orchestration
  - Basic activities represent the actual “components”, and correspond to the **invocation of a WSDL operation** performed by a service playing a role onto a service playing a different role
- Offers
  - **invoke** activities: the invocation of a request/reply or a one-way operation offered by a service
  - **receive** activities: the receipt of a message from a client
  - **reply** activities: a message sent by the process in response to an operation invoked by a client
  - **assign** activities: assigning data to variables
  - **wait** activities: defining points in the process where the execution should block for a certain period of time

Orchestration model

- BPEL allows the definition of s**tructured activities**, which can group a set of other structured or basic activities to define **ordering** constraints among them
- Structured activities
  - **Sequence**: Contains a set of activities to be executed sequentially
  - **Switch**: Includes a set of activities, each associated with a condition
  - **Pick**: Includes a set of events, each associated with an activity
  - **While**: Includes exactly one (basic or structured) activity, which is executed repeatedly while a specified condition is true
  - **Flow**: Groups a set of activities to be started in parallel. Considered complete when all the included activities are completed

Example of BPEL orchestration model



Orchestration model

- A flow activity can include the specification of **links**
- A link l can be used to connect one and only one **source activity** S to one and only one **target activity** T
- The target activity T cannot be started until the source activity S has been completed
- An activity may have **multiple incoming and outgoing links**
- Links can also be associated with a **transition condition**, evaluated when the link's source activity completes
- The set of links can only generate **acyclic graphs**
- Links may require the definition of **join** conditions (to further define when an activity should be executed in case it has **multiple incoming links**) and of **attributes** denoting the behavior when a transition condition is false
- The possibility of specifying the same orchestration logic in two different ways
  - The sequence of invocation of activities, A, B, and C can be defined as a sequence or a flow and links between A and B and between B and C

Data types and data transfer

- BPEL maintains the state of the process and manipulates control data by means of **variables**
- Variables are characterized by a **name** and by a **type**, specified as a reference to a WSDL message type, XML schema simple types, or XML schema elements
- Variables can be used as **input** or **output** parameters in operation invocations, and are referred to by **conditions**
- **XPath 1.0** can be used to identify a particular value in the variable
- BPEL follows the **blackboard** approach
- **In the abstract process, the source of the data for variable assignment is left unspecified** -> Enable non-deterministic specifications both in the orchestration and in the data transfer between activities

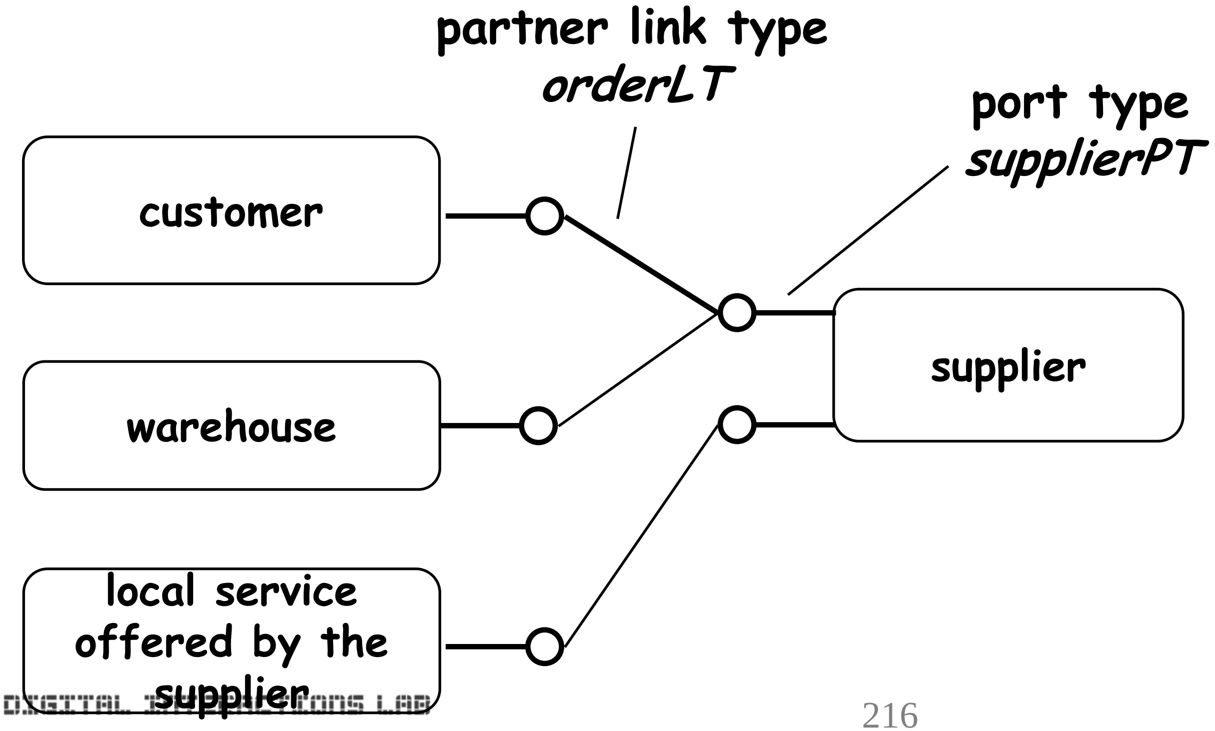
Service selection

- Revolves around the notion of **partner link types**, **partner links**, and **endpoint references**
- Partner link types identify a **pair of roles** that exchange messages during process execution and the WSDL port types that the services playing these roles are required to implement
- Once **roles**, **relationships**, and **port types** have been defined by means of partner link types, the next step consists of specifying partner links
- While partner link types identify roles, partner links identify **services invoked during the execution of a process**, and are therefore bound to specific endpoints
- The definition of a partner link **references a partner link type**, and then states the role played by the process and the one played by the partner with respect to that link -> A specific **endpoint reference** can be then associated to the partner link, identifying a specific customer
  - Example: invoking the web services from several warehouses
- **Activity definitions can then refer to partner links**

Service selection

partner link definition: it further qualifies the interactions occurring through a partner link type. Its definition refers to a partner link type and specifies the role played by the composite service as well as the one played by the other partner

```
<partnerLink name="customerP"
    partnerLinkType=“orderLT"
    myRole=“supplier”
    partnerRole=“customer”>
</partner>
```



partner link type ***orderLT***

port type ***supplierPT***

customer

warehouse

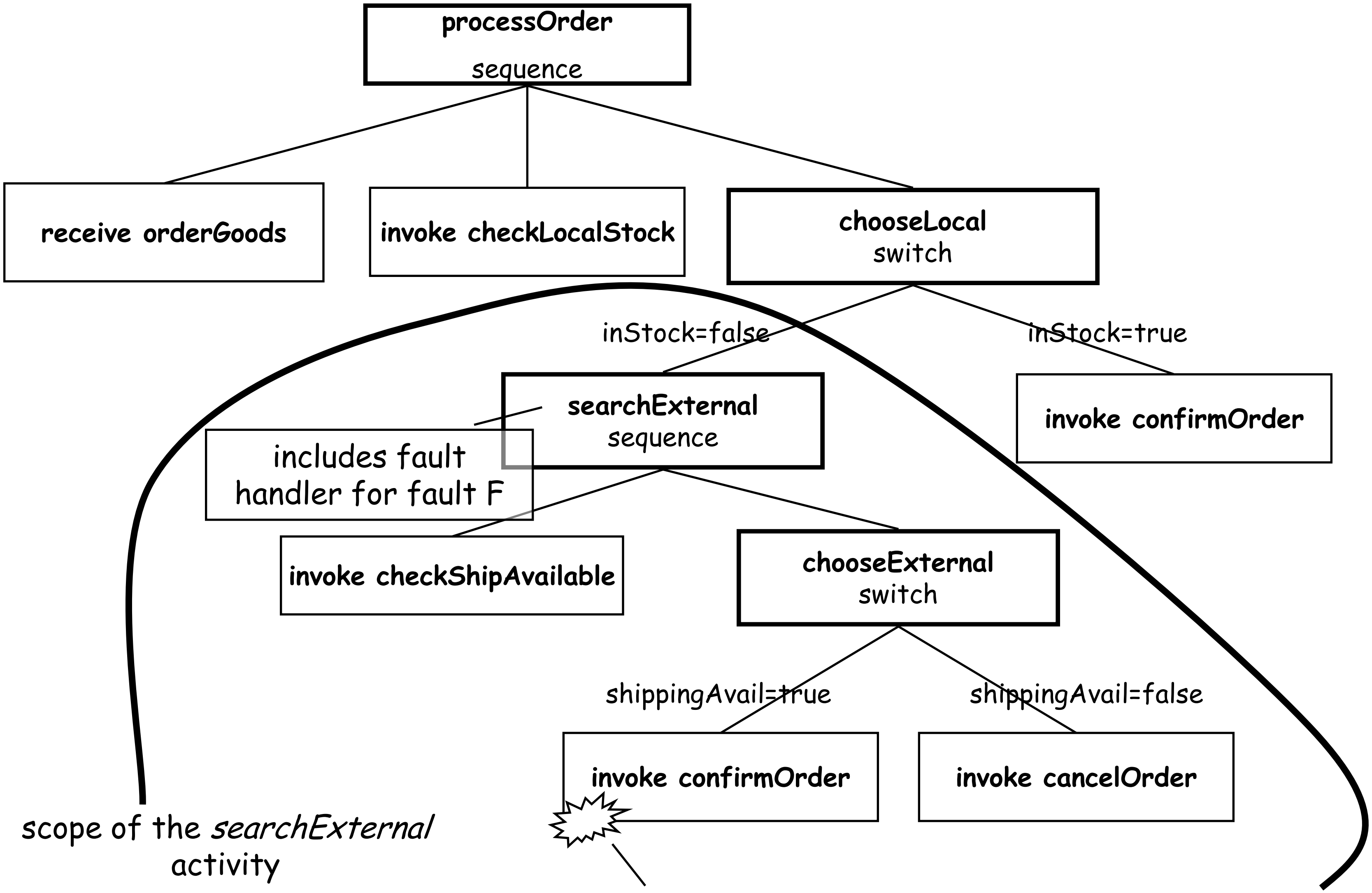
supplier

local service offered by the supplier

Exceptions

- BPEL follows a **try-catch-throw** approach
- Each activity implicitly defines a **scope** or scopes can be explicitly declared
- Any scope-defining element can include the specification of one or more **fault handlers**
- Faults can be generated during the execution of an activity within the scope, either by the invoked operation or by the execution engine, or within the orchestration schema
- In addition to the try-catch-throw, the **event handler**, which enables continuous monitoring for a certain event and executes an activity in response to the event can be used for handling exceptions

Exception handling



includes fault handler for fault F

scope of the *searchExternal* activity

due to the behavior of the default handler, implicitly associated with each activity, a fault F occurring in activity *send confirmOrder* would propagate up until activity searchExternal, where the handler resides

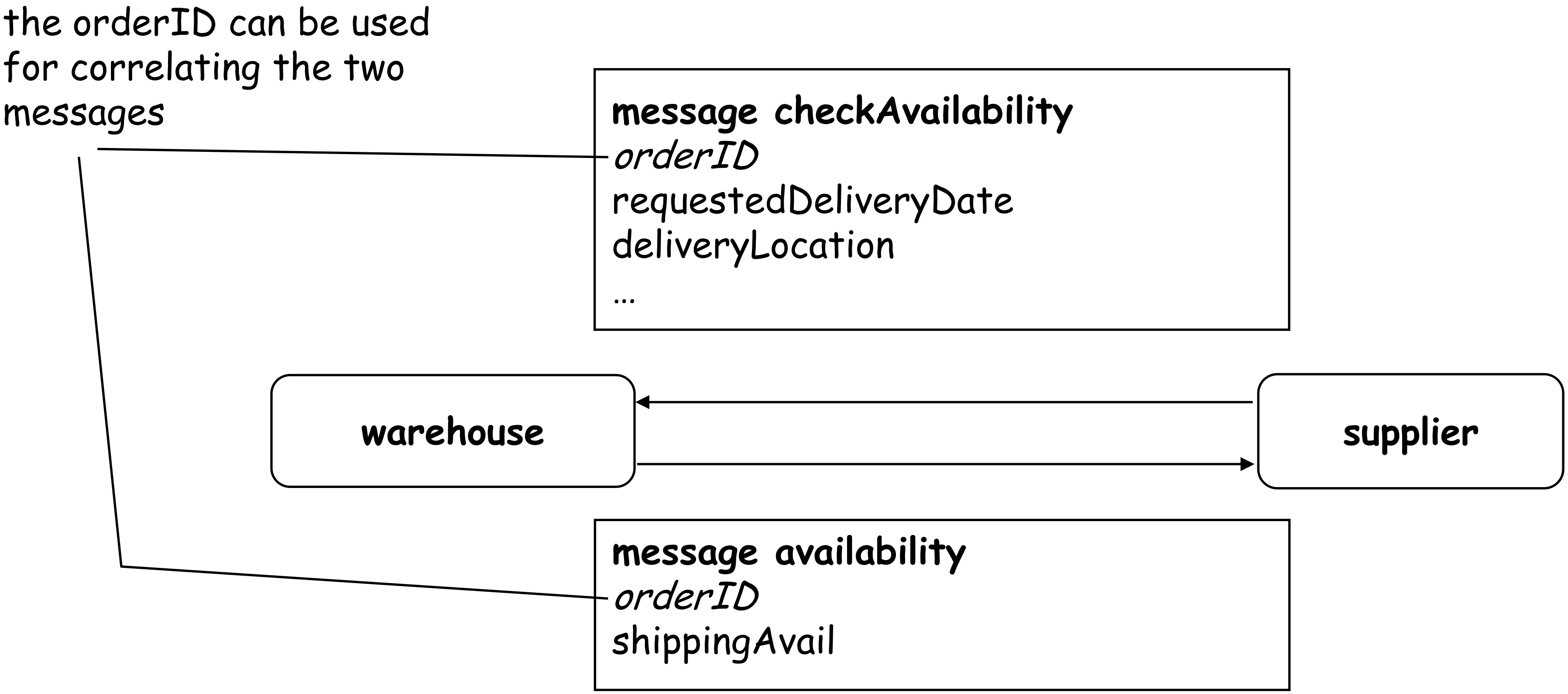
Transactions

- It is possible to define, **for each scope**, the logic required to semantically **undo** the execution of activities in that scope
- The compensation logic is specified by a **compensation handler,** consisting of a single (basic or structured) activity, that will take care of performing whatever actions are needed to compensate for the execution
- Every scope has a default compensation handler, whose behavior consists of invoking the compensation handler for each enclosed scope in the reverse order of execution

Instance routing

- BPEL includes instance routing support to cater to the cases in which routing is not transparently managed by the infrastructure
- Defines **how to correlate messages with instances**, based on the message data
- A composition schema may include the definition of **correlation sets**, a construct that essentially identifies a set of data items (e.g., a customer ID)
- Correlation sets can be **associated with messages** sent or received by the composite service within invoke, reply, or receive activities
  - If the messages have the same value for the correlation set, they belong to the same instance
  - Correlation set is used to uniquely identify a composition instance
  - Multiple correlation sets can be defined

Correlation set



the orderID can be used for correlating the two messages

message checkAvailability
orderID
requestedDeliveryDate
deliveryLocation
…

message availability
orderID
shippingAvail

Abstract BPEL4WS process

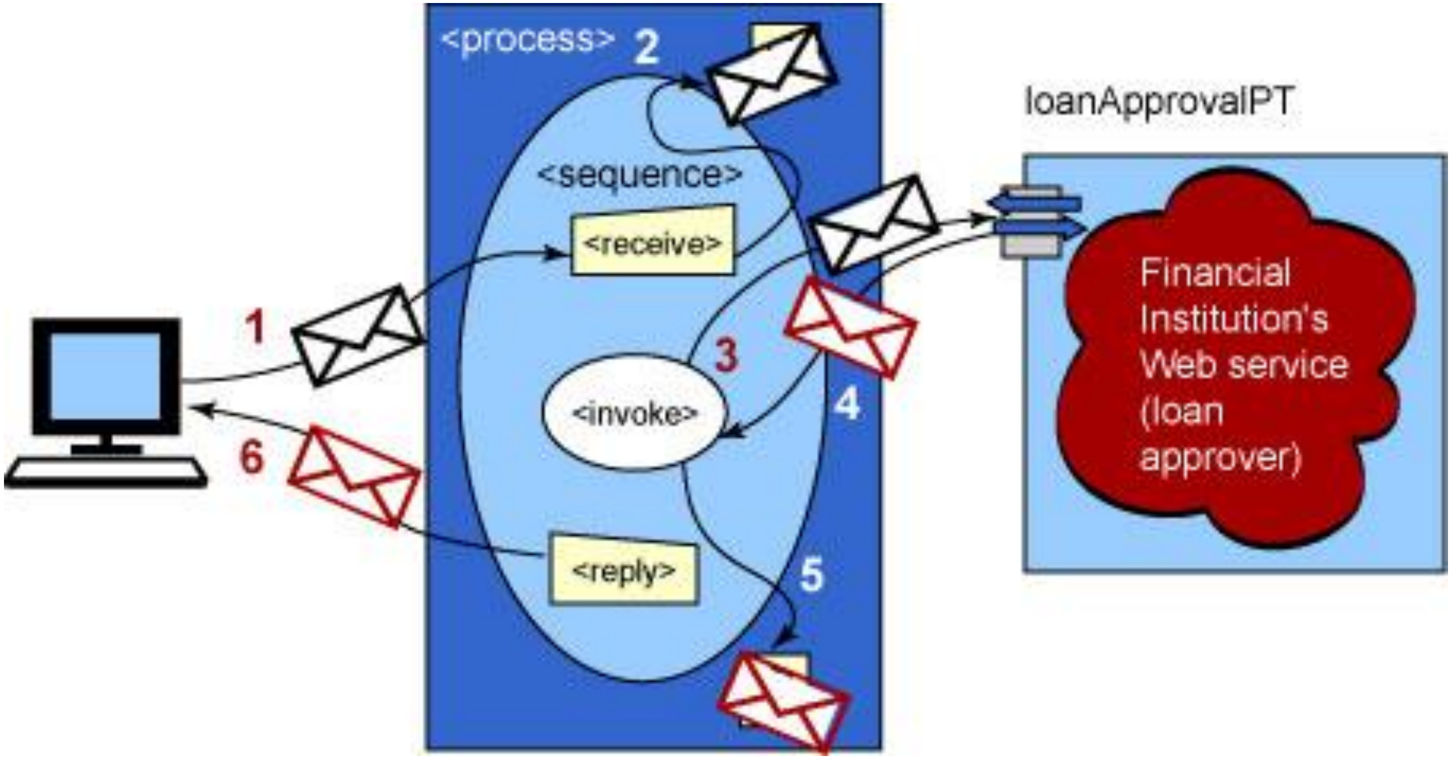
- describes just public aspects of the protocol
  - e.g., roles, service links
- restricted to manipulation of values contained in message properties, and use nondeterministic values to reflect the results of hidden private behavior

Example BPEL4WS Specification for a Stock Quotation Composite Service

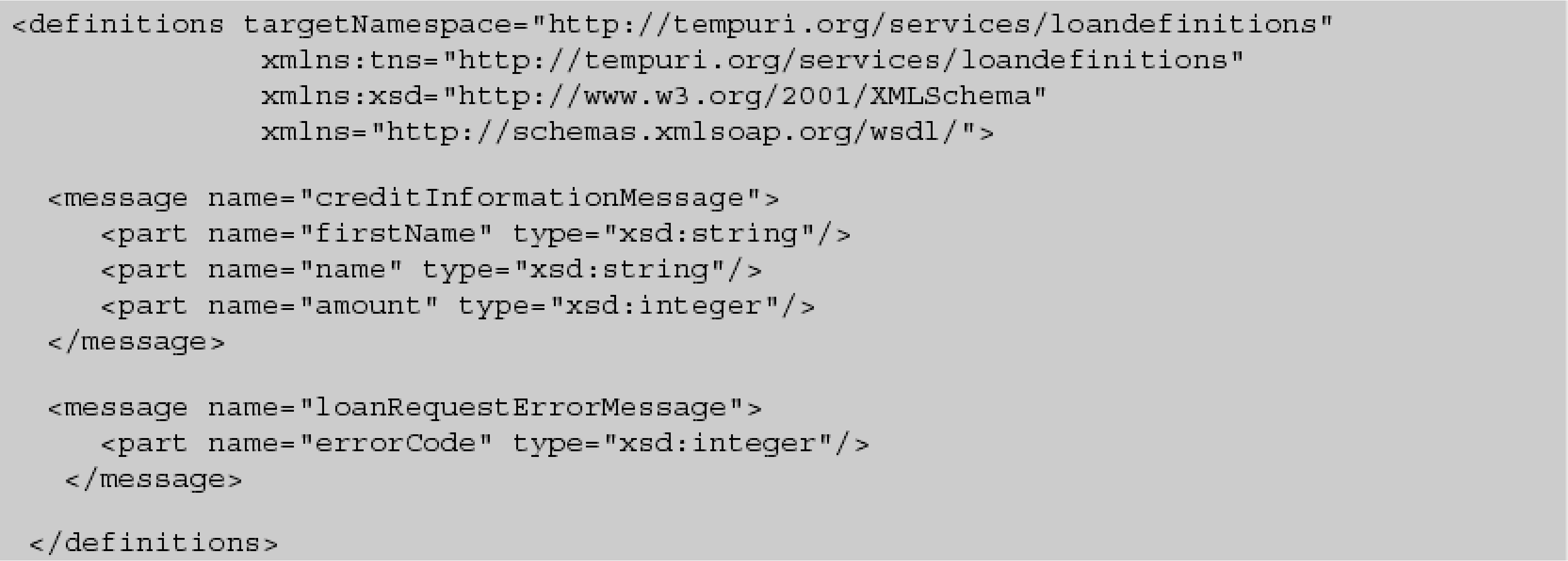
```
<process name="simple" targetNamespace="urn:stockQuoter" xmlns:tns="urn:stockQuoter"
         xmlns:sqp="http://tempuri.org/services/stockquote" xmlns=&BPEL;/>
  <containers>
    <container name="request" messageType="tns:request"/>
    <container name="response" messageType="tns:response"/>
    <container name="invocationRequest" messageType="sqp:GetQInput"/>
    <container name="invocationResponse" messageType="sqp:GetQOutput"/>
  </containers>
  <partners>
    <partner name="caller" serviceLinkType="tns:StockQuoteSLT"/>
    <partner name="provider" serviceLinkType="tns:StockQuoteSLT"/>
  </partners>

  <sequence name="sequence">
    <receive name="receive" partner="caller" portType="tns:StockQuotePT"
             operation="wantQuote" container="request" createInstance="yes"/>
    <assign>  <copy>
          <from container="request" part="symbol"/>
          <to container="invocationRequest" part="symbol"/>
      </copy> </assign>
    <invoke name="invoke" partner="provider" portType="sqp:StockQuotePT"
            operation="getQuote" inputContainer="invocationRequest" outputContainer="invocationResponse"/>
    <assign>  <copy>
          <from container="invocationResponse" part="quote"/>
          <to container="response" part="quote"/>
      </copy>  </assign>
    <reply name="reply" partner="caller" portType="tns:StockQuotePT" operation="wantQuote" container="response"/>
  </sequence>
</process>
```

BPEL4WS Example

(Source: IBM developerWorks)

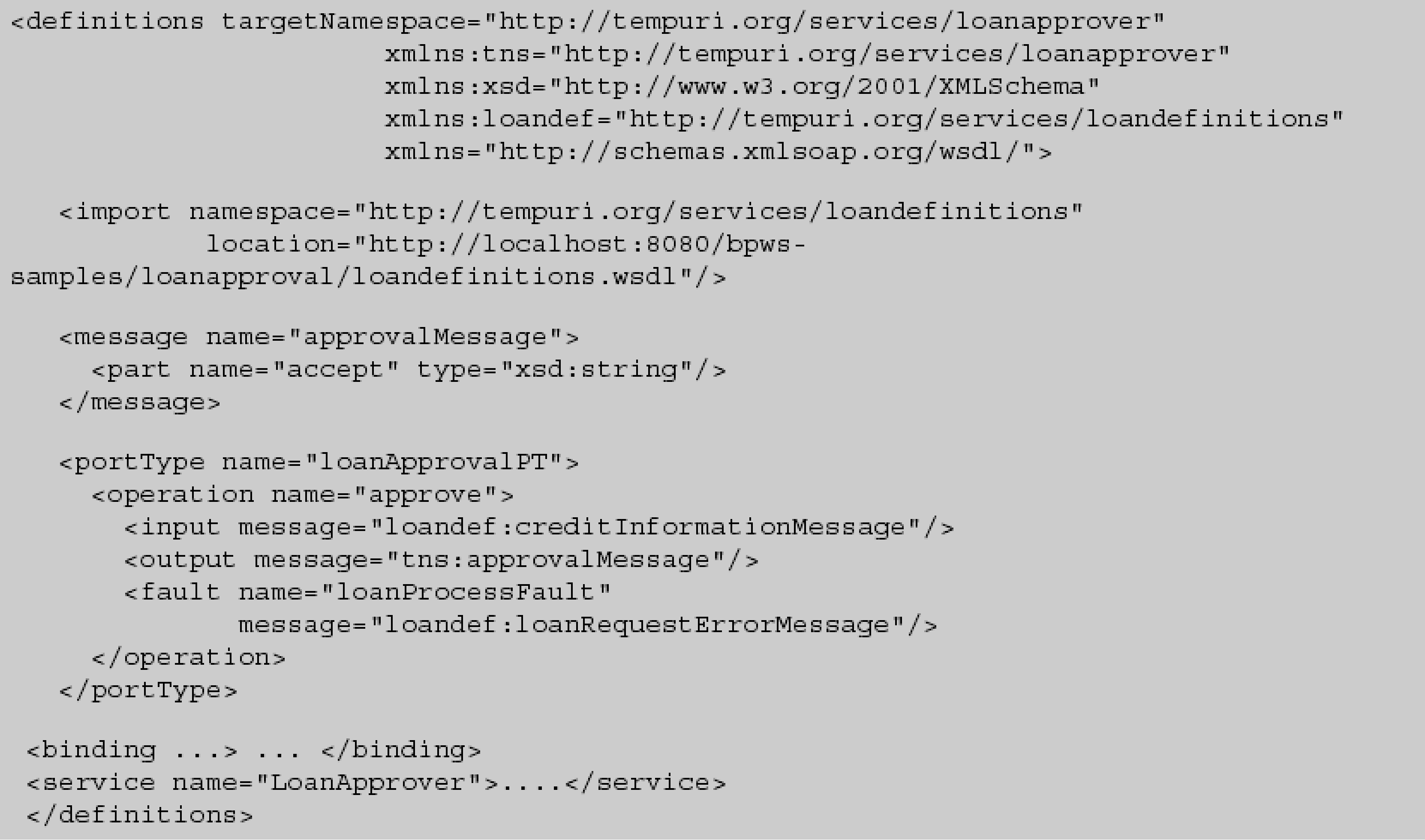


BPEL4WS Example: LoanDefinitions.wsdl



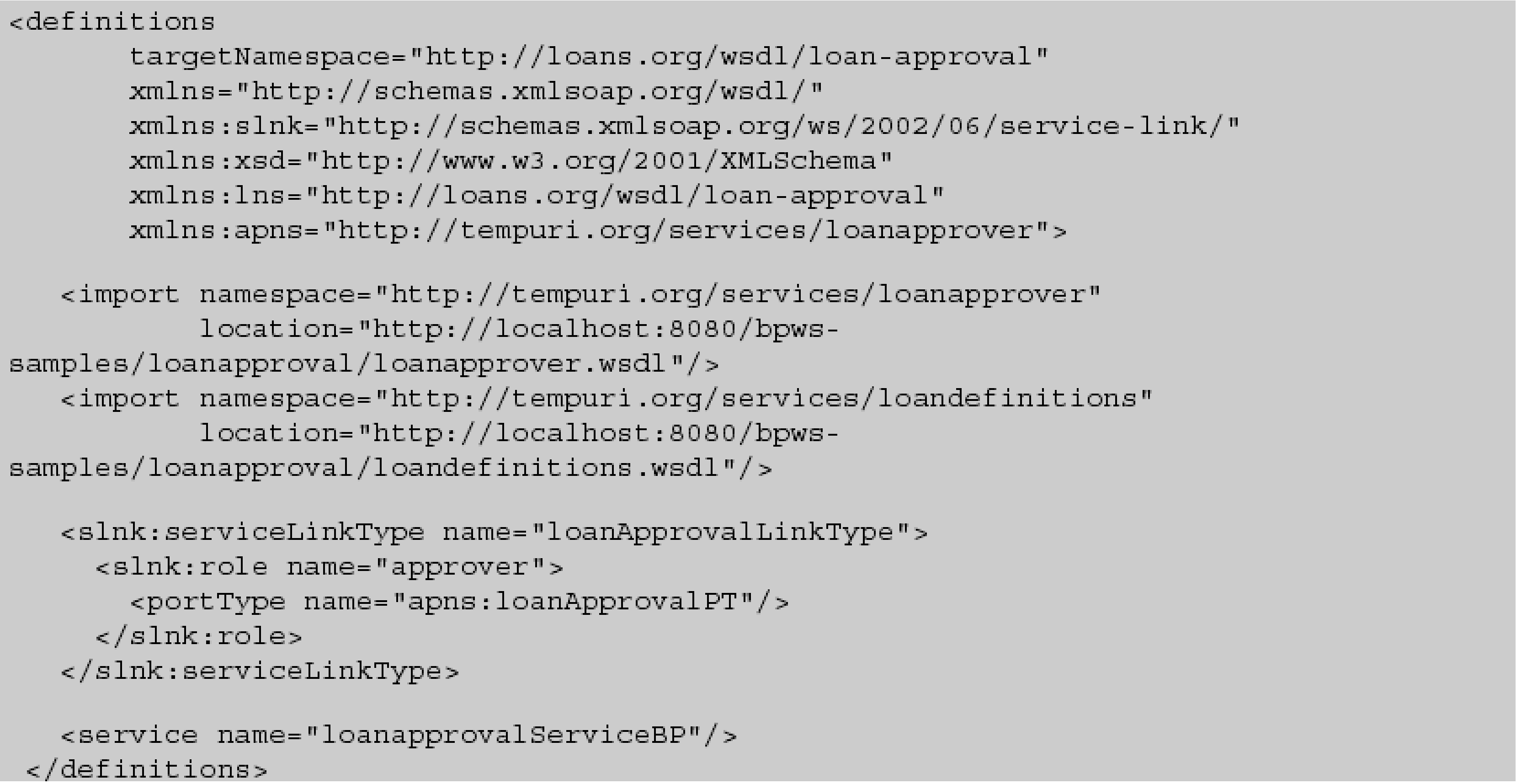
```
<definitions targetNamespace="http://tempuri.org/services/loandefinitions"
              xmlns:tns="http://tempuri.org/services/loandefinitions"
              xmlns:xsd="http://www.w3.org/2001/XMLSchema"
              xmlns="http://schemas.xmlsoap.org/wsdl/">

  <message name="creditInformationMessage">
     <part name="firstName" type="xsd:string"/>
     <part name="name" type="xsd:string"/>
     <part name="amount" type="xsd:integer"/>
  </message>

  <message name="loanRequestErrorMessage">
     <part name="errorCode" type="xsd:integer"/>
   </message>

 </definitions>
```

BPEL4WS Example: LoanApprover.wsdl



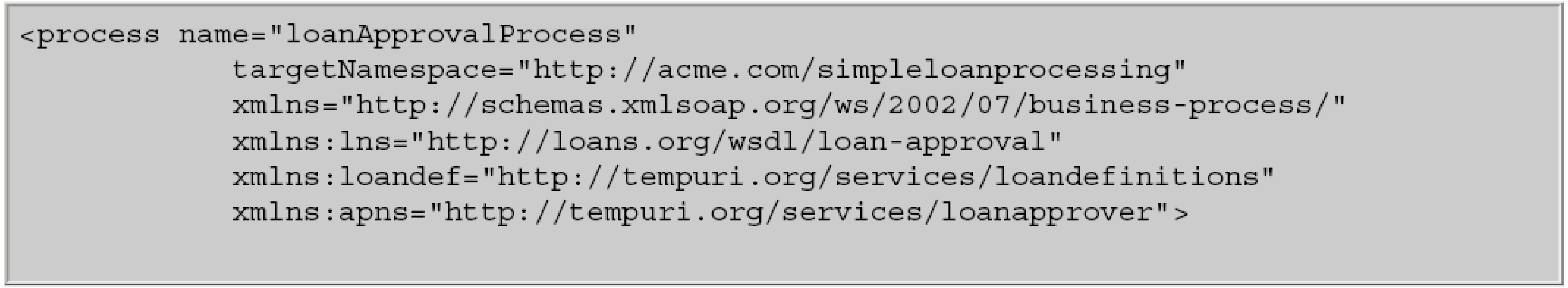
```
<definitions targetNamespace="http://tempuri.org/services/loanapprover"
                       xmlns:tns="http://tempuri.org/services/loanapprover"
                       xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                       xmlns:loandef="http://tempuri.org/services/loandefinitions"
                       xmlns="http://schemas.xmlsoap.org/wsdl/">

   <import namespace="http://tempuri.org/services/loandefinitions"
            location="http://localhost:8080/bpws-
samples/loanapproval/loandefinitions.wsdl"/>

   <message name="approvalMessage">
     <part name="accept" type="xsd:string"/>
   </message>

   <portType name="loanApprovalPT">
     <operation name="approve">
       <input message="loandef:creditInformationMessage"/>
       <output message="tns:approvalMessage"/>
       <fault name="loanProcessFault"
              message="loandef:loanRequestErrorMessage"/>
     </operation>
   </portType>

 <binding ...> ... </binding>
 <service name="LoanApprover">....</service>
 </definitions>
```

BPEL4WS Example: LoanApproval.wsdl

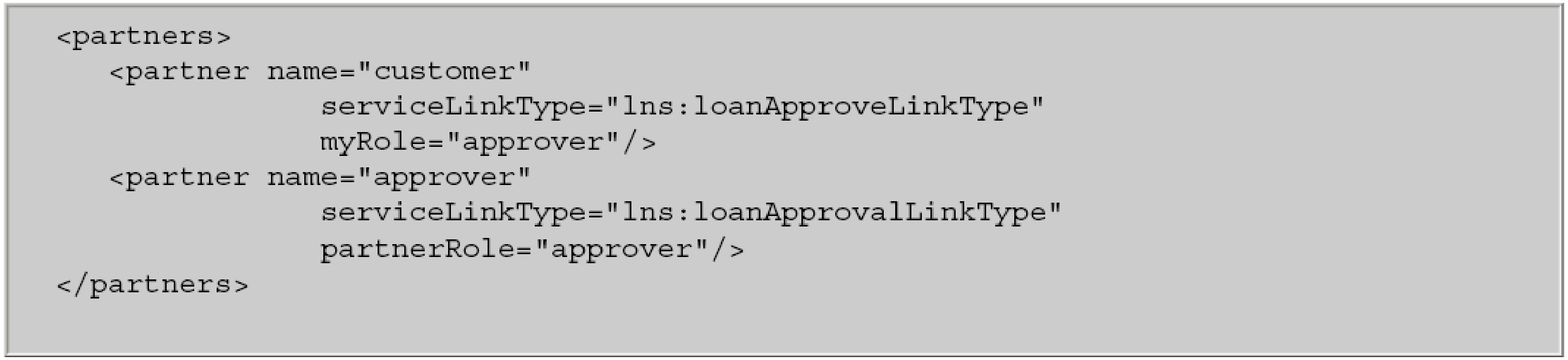


```
<definitions
       targetNamespace="http://loans.org/wsdl/loan-approval"
       xmlns="http://schemas.xmlsoap.org/wsdl/"
       xmlns:slnk="http://schemas.xmlsoap.org/ws/2002/06/service-link/"
       xmlns:xsd="http://www.w3.org/2001/XMLSchema"
       xmlns:lns="http://loans.org/wsdl/loan-approval"
       xmlns:apns="http://tempuri.org/services/loanapprover">

   <import namespace="http://tempuri.org/services/loanapprover"
           location="http://localhost:8080/bpws-
samples/loanapproval/loanapprover.wsdl"/>
   <import namespace="http://tempuri.org/services/loandefinitions"
           location="http://localhost:8080/bpws-
samples/loanapproval/loandefinitions.wsdl"/>

   <slnk:serviceLinkType name="loanApprovalLinkType">
     <slnk:role name="approver">
       <portType name="apns:loanApprovalPT"/>
     </slnk:role>
   </slnk:serviceLinkType>

   <service name="loanapprovalServiceBP"/>
 </definitions>
```

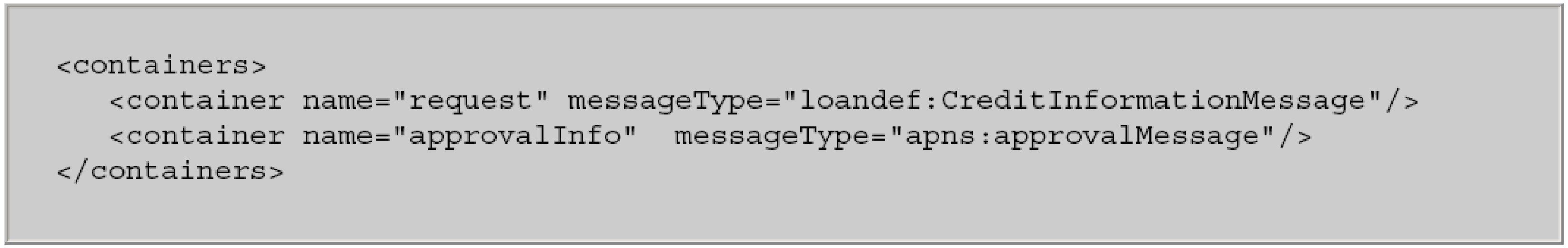
BPEL4WS Example: LoanApproval.bpel



```
<process name="loanApprovalProcess"
             targetNamespace="http://acme.com/simpleloanprocessing"
             xmlns="http://schemas.xmlsoap.org/ws/2002/07/business-process/"
             xmlns:lns="http://loans.org/wsdl/loan-approval"
             xmlns:loandef="http://tempuri.org/services/loandefinitions"
             xmlns:apns="http://tempuri.org/services/loanapprover">
```

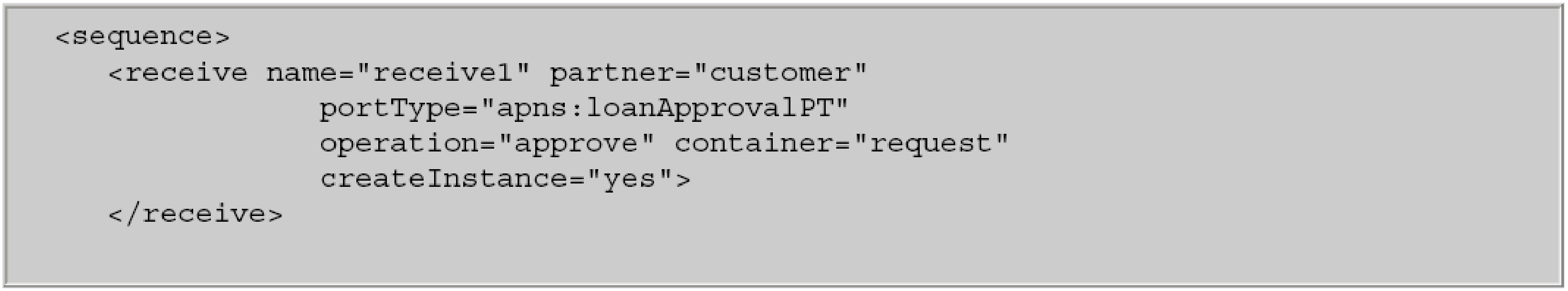


```
  <partners>
     <partner name="customer"
                 serviceLinkType="lns:loanApproveLinkType"
                 myRole="approver"/>
     <partner name="approver"
                 serviceLinkType="lns:loanApprovalLinkType"
                 partnerRole="approver"/>
  </partners>
```

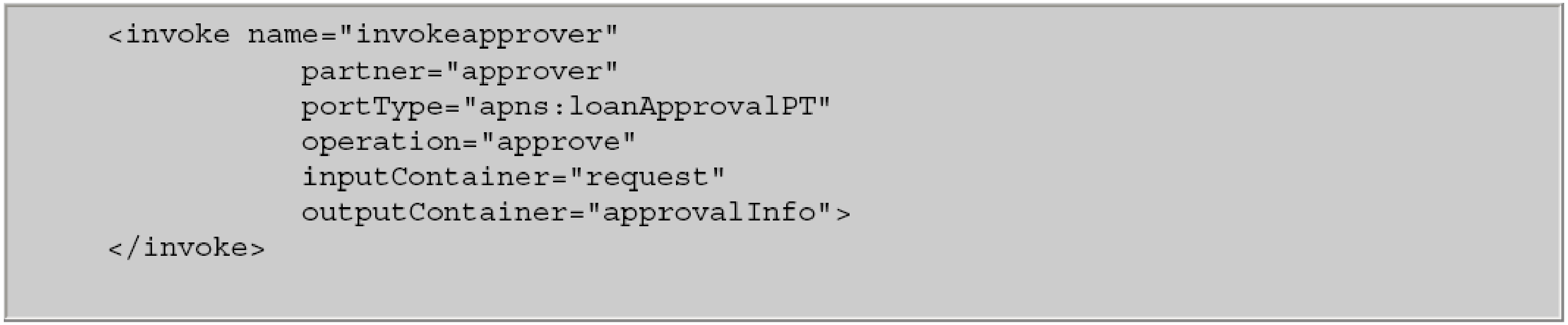


```
  <containers>
     <container name="request" messageType="loandef:CreditInformationMessage"/>
     <container name="approvalInfo"  messageType="apns:approvalMessage"/>
  </containers>
```

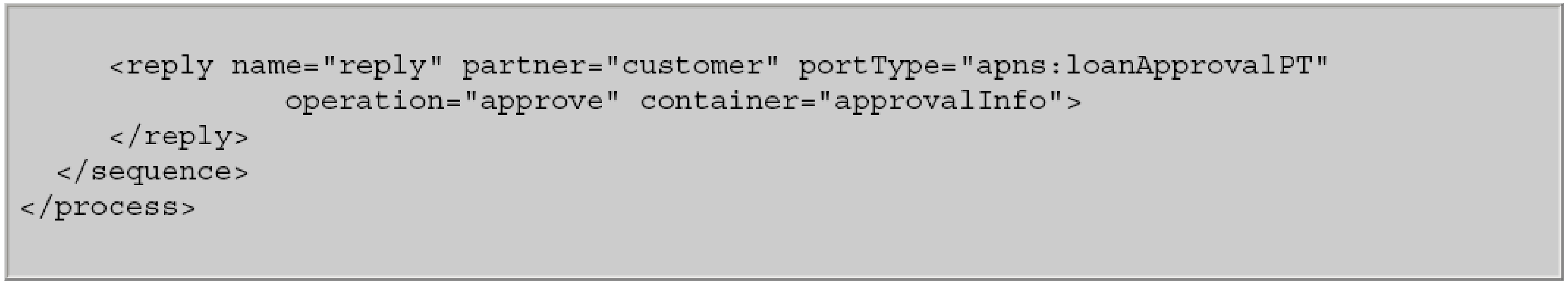
BPEL4WS Example: LoanApproval.bpel (cont.)



```
  <sequence>
     <receive name="receive1" partner="customer"
                 portType="apns:loanApprovalPT"
                 operation="approve" container="request"
                 createInstance="yes">
     </receive>
```



```
     <invoke name="invokeapprover"
                partner="approver"
                portType="apns:loanApprovalPT"
                operation="approve"
                inputContainer="request"
                outputContainer="approvalInfo">
     </invoke>
```



```
     <reply name="reply" partner="customer" portType="apns:loanApprovalPT"
               operation="approve" container="approvalInfo">
     </reply>
  </sequence>
</process>
```

상용 BPEL 엔진

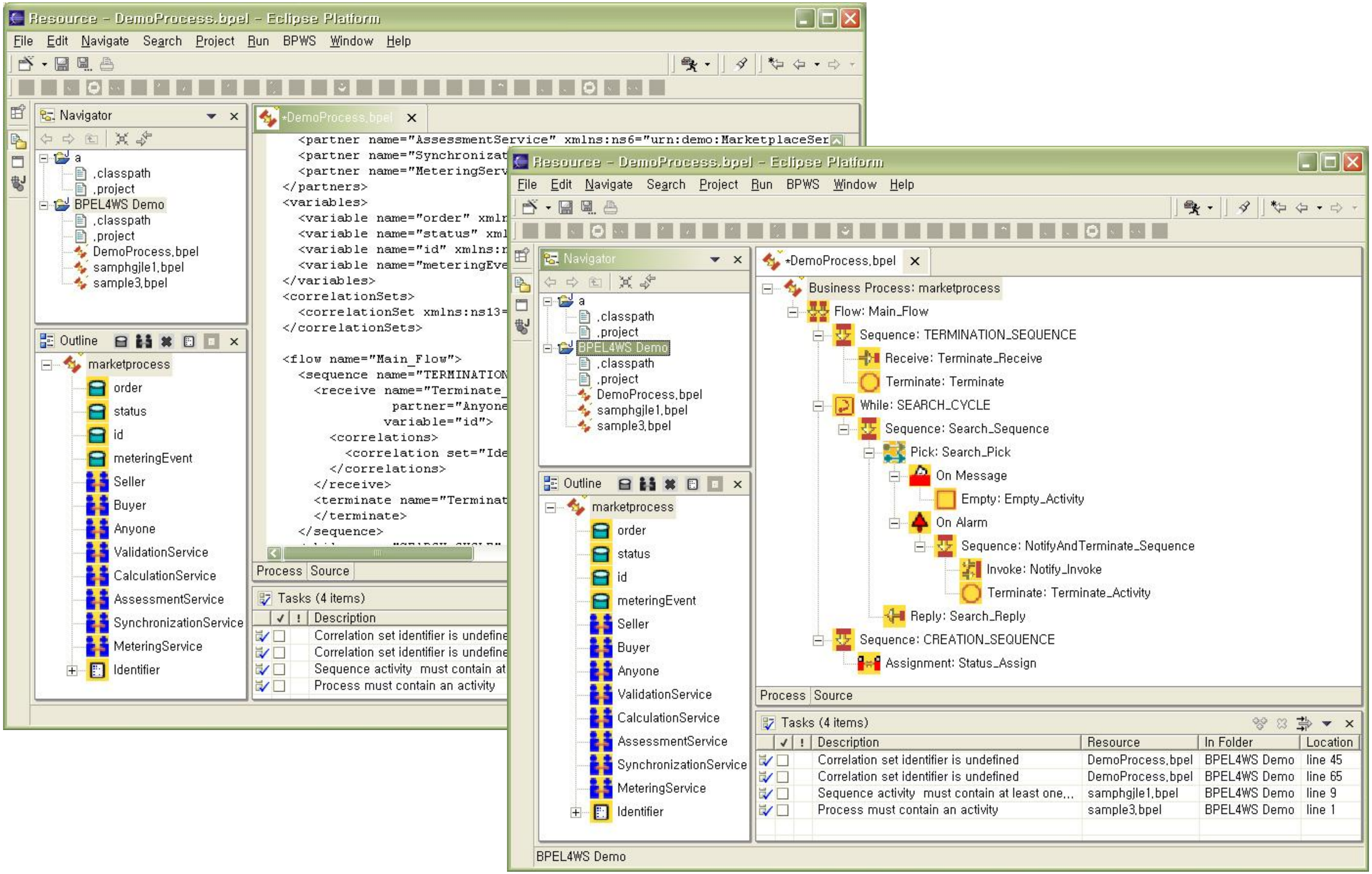
- **Collaxa BPEL server**
  - Java based
  - On top of JBoss or WebLogic
- **OpenStorm ChreoServer**
  - .NET & J2EE based
  - Transferred from Momentum software
  - Not yet released
- **IBM AlphaWorks BPWS4J**
  - Java based
  - Requires WebSphere AS and Eclipse
- **Microsoft BizTalk 2004**
- **Oracle:** Acquired Collaxa
- **WebMethods:** Soon to be included
- **FiveSight**’s PXE
- **ReadiMinds**

Open source BPEL engines

- ActiveBPEL
- Twister
- uEngine

- Note: BPELJ (BPEL for Java)
  - a combination of BPEL and the Java programming language allowing the two languages to be used together to build business process applications
  - http://www-106.ibm.com/developerworks/webservices/library/ws-bpelj/

BPEL4J


Resource - DemoProcess.bpel - Eclipse Platform
File Edit Navigate Search Project Run BPWS Window Help
Navigator
Outline
*DemoProcess.bpel
Process Source
BPEL4WS Demo

```
a
  .classpath
  .project
BPEL4WS Demo
  .classpath
  .project
  DemoProcess.bpel
  samphgjle1.bpel
  sample3.bpel
```

```
marketprocess
  order
  status
  id
  meteringEvent
  Seller
  Buyer
  Anyone
  ValidationService
  CalculationService
  AssessmentService
  SynchronizationService
  MeteringService
  Identifier
```

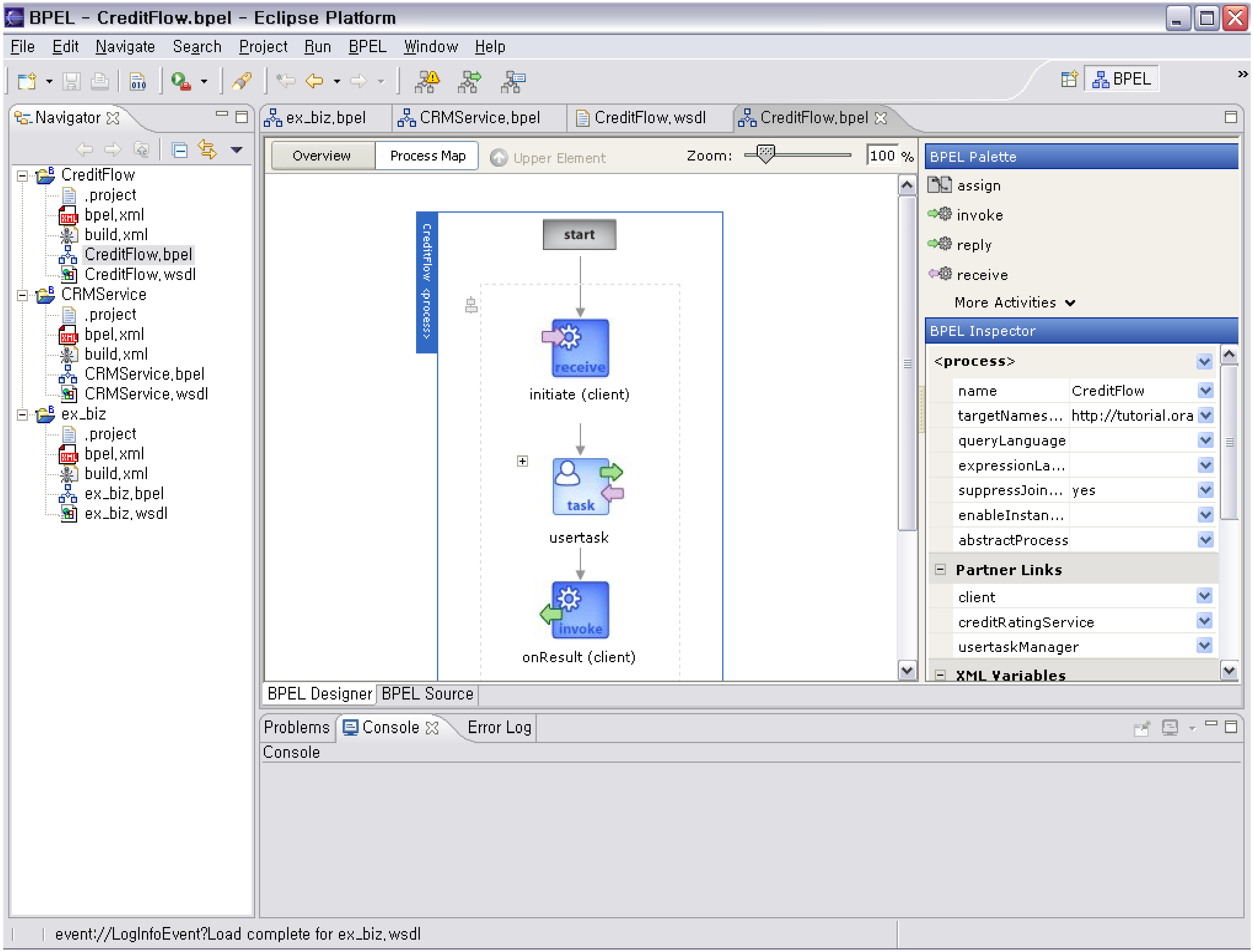
```
    <partner name="AssessmentService" xmlns:ns6="urn:demo:MarketplaceSer
    <partner name="Synchronizat
    <partner name="MeteringServ
  </partners>
  <variables>
    <variable name="order" xmln
    <variable name="status" xml
    <variable name="id" xmlns:n
    <variable name="meteringEve
  </variables>
  <correlationSets>
    <correlationSet xmlns:ns13=
  </correlationSets>

  <flow name="Main_Flow">
    <sequence name="TERMINATION
      <receive name="Terminate_
                 partner="Anyone
                variable="id">
        <correlations>
          <correlation set="Ide
        </correlations>
      </receive>
      <terminate name="Terminat
      </terminate>
    </sequence>
```

```
Business Process: marketprocess
  Flow: Main_Flow
    Sequence: TERMINATION_SEQUENCE
      Receive: Terminate_Receive
      Terminate: Terminate
    While: SEARCH_CYCLE
      Sequence: Search_Sequence
        Pick: Search_Pick
          On Message
            Empty: Empty_Activity
          On Alarm
            Sequence: NotifyAndTerminate_Sequence
              Invoke: Notify_Invoke
              Terminate: Terminate_Activity
        Reply: Search_Reply
    Sequence: CREATION_SEQUENCE
      Assignment: Status_Assign
```

Tasks (4 items)

✓	!	Description	Resource	In Folder	Location
□		Correlation set identifier is undefined	DemoProcess.bpel	BPEL4WS Demo	line 45
□		Correlation set identifier is undefined	DemoProcess.bpel	BPEL4WS Demo	line 65
□		Sequence activity must contain at least one...	samphgjle1.bpel	BPEL4WS Demo	line 9
□		Process must contain an activity	sample3.bpel	BPEL4WS Demo	line 1

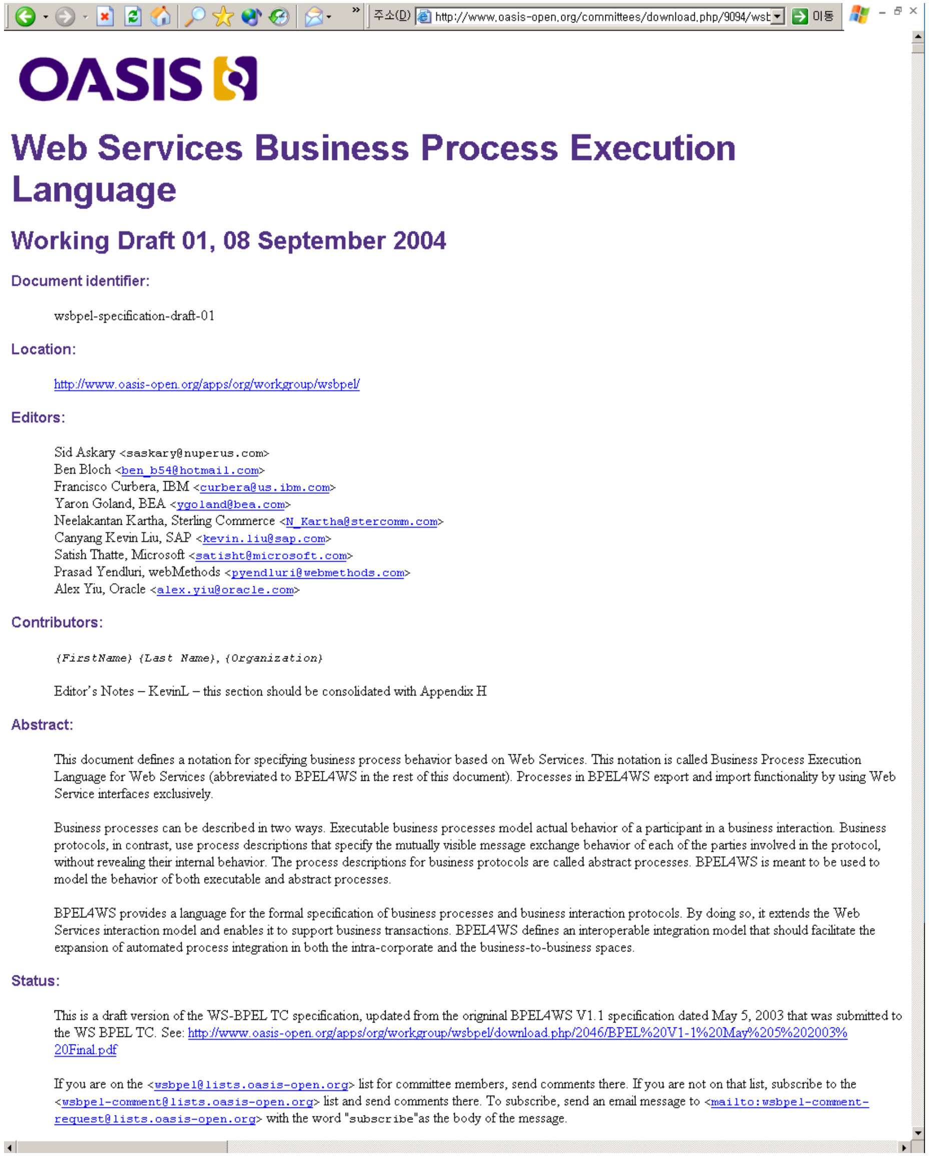
Oracle BPEL



name	CreditFlow
targetNames...	http://tutorial.ora
queryLanguage	
expressionLa...	
suppressJoin...	yes
enableInstan...	
abstractProcess	

event://LogInfoEvent?Load complete for ex_biz.wsdl

Current status of WSBPEL



Web Services Business Process Execution Language

Working Draft 01, 08 September 2004

Document identifier:

wsbpel-specification-draft-01

Location:

http://www.oasis-open.org/apps/org/workgroup/wsbpel/

Editors:

Sid Askary <saskary@nuperus.com>
Ben Bloch <ben_b54@hotmail.com>
Francisco Curbera, IBM <curbera@us.ibm.com>
Yaron Goland, BEA <ygoland@bea.com>
Neelakantan Kartha, Sterling Commerce <N_Kartha@stercomm.com>
Canyang Kevin Liu, SAP <kevin.liu@sap.com>
Satish Thatte, Microsoft <satisht@microsoft.com>
Prasad Yendluri, webMethods <pyendluri@webmethods.com>
Alex Yiu, Oracle <alex.yiu@oracle.com>

Contributors:

{FirstName} {Last Name}, {Organization}

Editor's Notes – KevinL – this section should be consolidated with Appendix H

Abstract:

This document defines a notation for specifying business process behavior based on Web Services. This notation is called Business Process Execution Language for Web Services (abbreviated to BPEL4WS in the rest of this document). Processes in BPEL4WS export and import functionality by using Web Service interfaces exclusively.

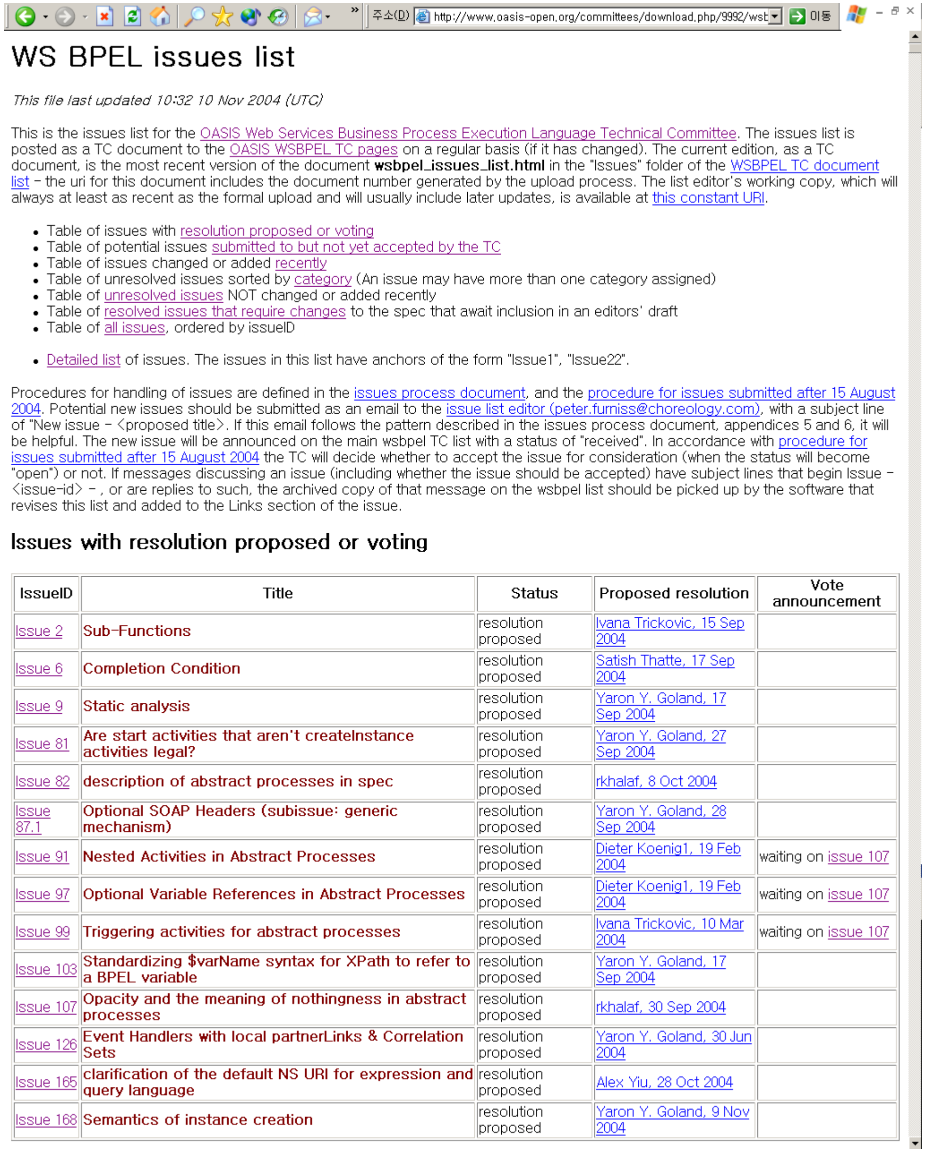
Business processes can be described in two ways. Executable business processes model actual behavior of a participant in a business interaction. Business protocols, in contrast, use process descriptions that specify the mutually visible message exchange behavior of each of the parties involved in the protocol, without revealing their internal behavior. The process descriptions for business protocols are called abstract processes. BPEL4WS is meant to be used to model the behavior of both executable and abstract processes.

BPEL4WS provides a language for the formal specification of business processes and business interaction protocols. By doing so, it extends the Web Services interaction model and enables it to support business transactions. BPEL4WS defines an interoperable integration model that should facilitate the expansion of automated process integration in both the intra-corporate and the business-to-business spaces.

Status:

This is a draft version of the WS-BPEL TC specification, updated from the origninal BPEL4WS V1.1 specification dated May 5, 2003 that was submitted to the WS BPEL TC. See: http://www.oasis-open.org/apps/org/workgroup/wsbpel/download.php/2046/BPEL%20V1-1%20May%205%202003%20Final.pdf

If you are on the <wsbpel@lists.oasis-open.org> list for committee members, send comments there. If you are not on that list, subscribe to the <wsbpel-comment@lists.oasis-open.org> list and send comments there. To subscribe, send an email message to <mailto:wsbpel-comment-request@lists.oasis-open.org> with the word "subscribe"as the body of the message.



WS BPEL issues list

This file last updated 10:32 10 Nov 2004 (UTC)

This is the issues list for the OASIS Web Services Business Process Execution Language Technical Committee. The issues list is posted as a TC document to the OASIS WSBPEL TC pages on a regular basis (if it has changed). The current edition, as a TC document, is the most recent version of the document **wsbpel_issues_list.html** in the "Issues" folder of the WSBPEL TC document list – the uri for this document includes the document number generated by the upload process. The list editor's working copy, which will always at least as recent as the formal upload and will usually include later updates, is available at this constant URI.

- Table of issues with resolution proposed or voting
- Table of potential issues submitted to but not yet accepted by the TC
- Table of issues changed or added recently
- Table of unresolved issues sorted by category (An issue may have more than one category assigned)
- Table of unresolved issues NOT changed or added recently
- Table of resolved issues that require changes to the spec that await inclusion in an editors' draft
- Table of all issues, ordered by issueID

- Detailed list of issues. The issues in this list have anchors of the form "Issue1", "Issue22".

Procedures for handling of issues are defined in the issues process document, and the procedure for issues submitted after 15 August 2004. Potential new issues should be submitted as an email to the issue list editor (peter.furniss@choreology.com), with a subject line of "New issue – <proposed title>. If this email follows the pattern described in the issues process document, appendices 5 and 6, it will be helpful. The new issue will be announced on the main wsbpel TC list with a status of "received". In accordance with procedure for issues submitted after 15 August 2004 the TC will decide whether to accept the issue for consideration (when the status will become "open") or not. If messages discussing an issue (including whether the issue should be accepted) have subject lines that begin Issue – <issue-id> – , or are replies to such, the archived copy of that message on the wsbpel list should be picked up by the software that revises this list and added to the Links section of the issue.

Issues with resolution proposed or voting

IssueID	Title	Status	Proposed resolution	Vote announcement
Issue 2	Sub-Functions	resolution proposed	Ivana Trickovic, 15 Sep 2004	
Issue 6	Completion Condition	resolution proposed	Satish Thatte, 17 Sep 2004	
Issue 9	Static analysis	resolution proposed	Yaron Y. Goland, 17 Sep 2004	
Issue 81	Are start activities that aren't createInstance activities legal?	resolution proposed	Yaron Y. Goland, 27 Sep 2004	
Issue 82	description of abstract processes in spec	resolution proposed	rkhalaf, 8 Oct 2004	
Issue 87.1	Optional SOAP Headers (subissue: generic mechanism)	resolution proposed	Yaron Y. Goland, 28 Sep 2004	
Issue 91	Nested Activities in Abstract Processes	resolution proposed	Dieter Koenig1, 19 Feb 2004	waiting on issue 107
Issue 97	Optional Variable References in Abstract Processes	resolution proposed	Dieter Koenig1, 19 Feb 2004	waiting on issue 107
Issue 99	Triggering activities for abstract processes	resolution proposed	Ivana Trickovic, 10 Mar 2004	waiting on issue 107
Issue 103	Standardizing $varName syntax for XPath to refer to a BPEL variable	resolution proposed	Yaron Y. Goland, 17 Sep 2004	
Issue 107	Opacity and the meaning of nothingness in abstract processes	resolution proposed	rkhalaf, 30 Sep 2004	
Issue 126	Event Handlers with local partnerLinks & Correlation Sets	resolution proposed	Yaron Y. Goland, 30 Jun 2004	
Issue 165	clarification of the default NS URI for expression and query language	resolution proposed	Alex Yiu, 28 Oct 2004	
Issue 168	Semantics of instance creation	resolution proposed	Yaron Y. Goland, 9 Nov 2004	

BPML

- Business Process Modeling Language
- has roots in web services (SOAP, WSDL, and UDDI)
- takes advantage of the XML technologies (XPath and XML Schema)
- designed to leverage other specifications (WS-Security, WS-Transaction)
- supports modeling of real-world business processes through its support for advanced semantics, such as nested processes and complex compensated transactions
- builds on the foundation of WSCI for expressing public interfaces and choreographies

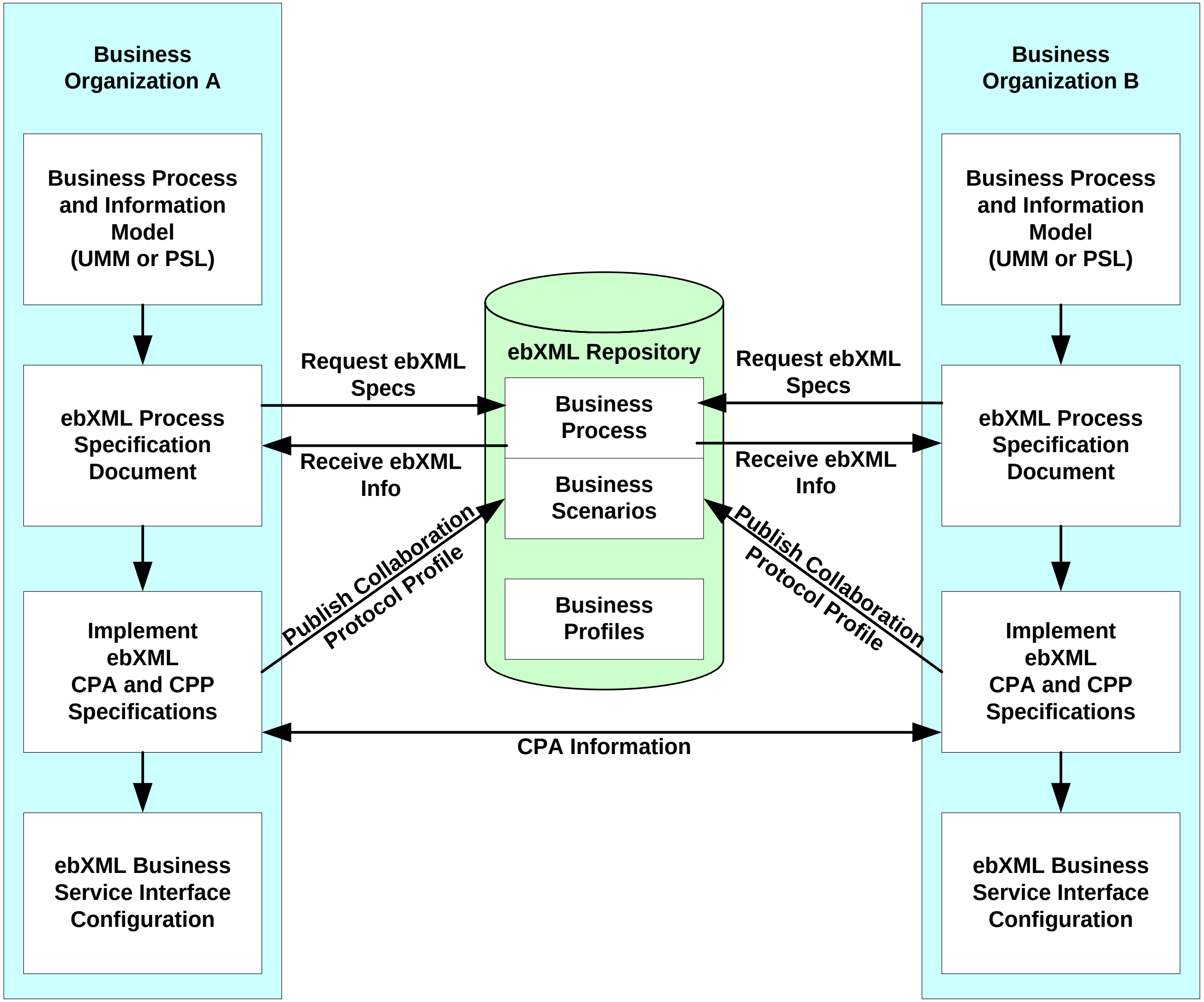
Electronic Business Extensible Markup Language (ebXML)

- Established by UN-CEFACT (United Nations Centre for Trade Facilitation and Electronic Business) and OASIS (Organization for the Advancement of Structured Information Standards)
- Provides specifications to define standard business processes and trading agreements among different organizations
- Also specifies the business messages that are exchanged as part of a business process
- The objective: to be a global standard for governmental and commercial organizations of all sizes to find business partners and interact with them

ebXML Vocabulary

- Unified Modeling Methodology (UMM)
  - Specialized UML for Business Processes
- Process specification document
  - describes the activities of the parties in an ebXML interaction
- Collaboration Protocol Profile (CPP)
  - describes an organization’s profile, i.e., which business processes it supports, its roles in those processes, the messages exchanged, and the transport mechanism for the messages (e.g., HTTPS)
- Collaborative Partner Agreement (CPA)
  - An intersection of two CPPs
  - represents a technical agreement between two or more partners
  - May be legally binding
- The CPP and CPA serve as configuration files (e.g., messaging headers) for ebXML business service interface software

Design of an ebXML System


Business Organization A
Business Process and Information Model (UMM or PSL)
ebXML Process Specification Document
Implement ebXML CPA and CPP Specifications
ebXML Business Service Interface Configuration
Request ebXML Specs
Receive ebXML Info
Publish Collaboration Protocol Profile
ebXML Repository
Business Process
Business Scenarios
Business Profiles
CPA Information
Request ebXML Specs
Receive ebXML Info
Publish Collaboration Protocol Profile
Business Organization B
Business Process and Information Model (UMM or PSL)
ebXML Process Specification Document
Implement ebXML CPA and CPP Specifications
ebXML Business Service Interface Configuration

An example ebXML BPSS document

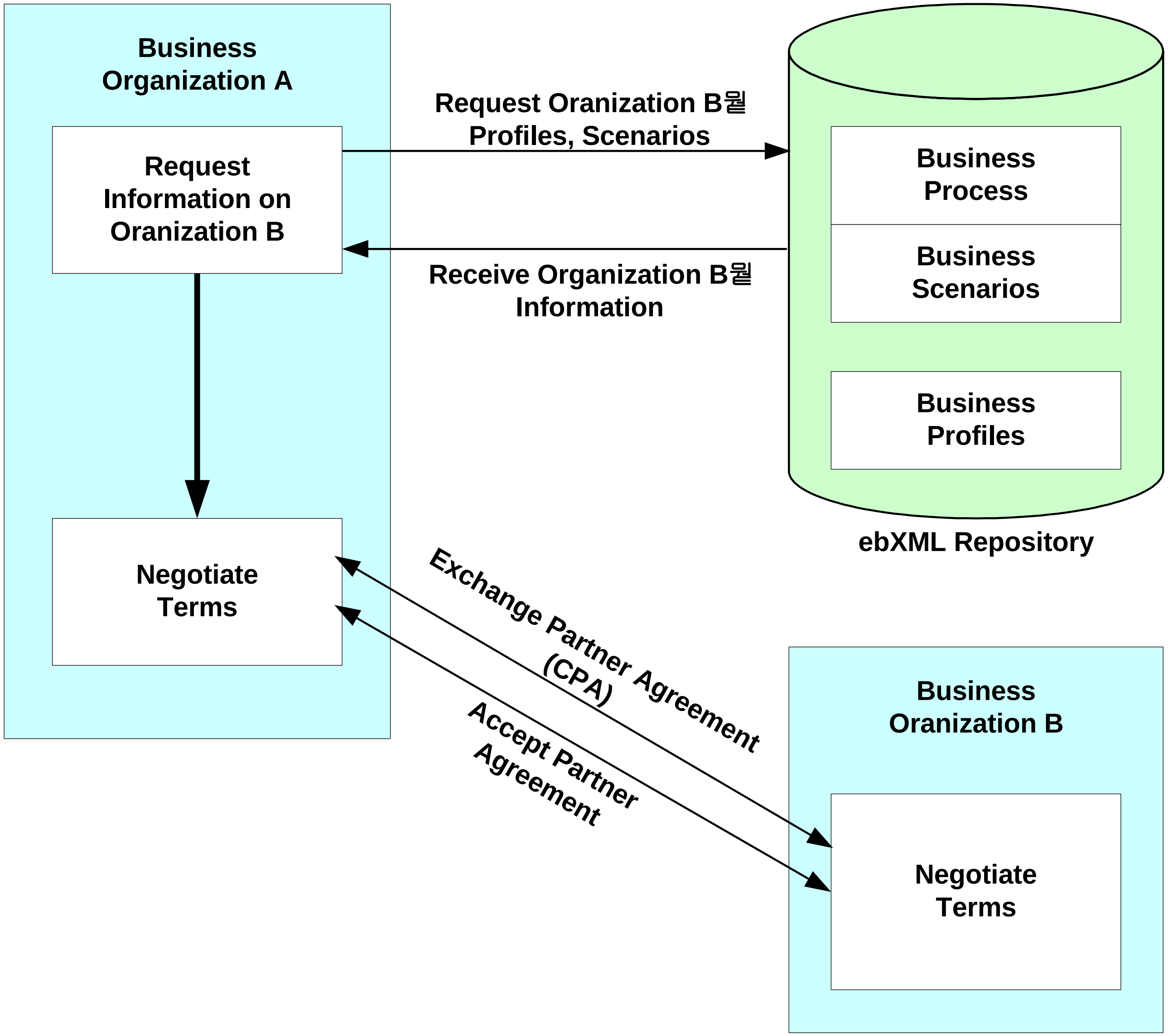
```
<ProcessSpecification
  xmlns="http://www.ebxml.org/BusinessProcess"
  name="PIP3A4RequestPurchaseOrder">
<!-- The request document and its XML Schema -->
 <BusinessDocument name="PO Request"
   nameID="Pip3A4PORequest"
   specificationLocation="PurchaseOrderRequest.xsd"/>
<!-- The confirmation document and its XML Schema -->
 <BusinessDocument name="PO Confirmation"
   nameID="Pip3A4POConfirmation"
   specificationLocation="PurchaseOrderConfirmation.xsd"/>
<!-- This process specification has one business -->
<!-- transaction consisting of a requesting and -->
<!-- a responding business activity-->
 <BusinessTransaction name="Request PO"
  nameID="RequestPO_BT">
   <RequestingBusinessActivity
    name="PO Request Action"
    nameID="PORequestAction"
    isAuthorizationRequired="true"
    isNonRepudiationRequired="true"
    timeToAcknowledgeReceipt="PT2H">
    <DocumentEnvelope
      businessDocument="PO Request"
      businessDocumentIDRef="Pip3A4PurchaseOrderRequest"/>
   </RequestingBusinessActivity>
   <RespondingBusinessActivity
    name="PO Confirmation Action"
    nameID="POConfirmationAction"
    isAuthorizationRequired="true"
    isNonRepudiationRequired="true"
    timeToAcknowledgeReceipt="PT2H">
    <DocumentEnvelope
     businessDocument="PO Confirmation"
     businessDocumentIDRef="Pip3A4PurchaseOrderConfirmation"/>
  </RespondingBusinessActivity>
 </BusinessTransaction>
```

```
<!-- The binary collaboration asserts that the buyer is -->
<!-- the initiator of the above business transaction and -->
<!-- the seller is the responder, and the process begins -->
<!-- in the Request PO state -->
 <BinaryCollaboration name="Request PO"
   nameID="RequestPO_BC">
  <InitiatingRole name="Buyer" nameID="BuyerId"/>
  <RespondingRole name="Seller" nameID="SellerId"/>
  <Start toBusinessState="Request PO"/>
  <BusinessTransactionActivity name="Request PO"
    nameID="RequestPO_BTA"
    businessTransaction="Request PO"
    businessTransactionIDRef="RequestPO_BT"
    fromAuthorizedRole="Buyer"
    fromAuthorizedRoleIDRef="BuyerId"
    toAuthorizedRole="Seller"
    toAuthorizedRoleIDRef="SellerId"
    timeToPerform="PT1D"/>
 </BinaryCollaboration>
</ProcessSpecification>
```

An example of an ebXML CPP

```
<cp:CollaborationProtocolProfile
  xmlns:cp="http://www.ebxml.org/specs/cpp-cpa-v2_0.xsd"
  xmlns:xlink="...">
  <cp:PartyInfo cp:partyName="PCInc"
    cp:defaultMshChannelId="asyncChannelA1">
    <cp:PartyId
      cp:type="urn:ebxml-cppa:partyid-type:duns">
        123456789
    </cp:PartyId>
    <cp:PartyRef
      xlink:href="http://PCInc.com/about.html"/>
    <cp:CollaborationRole cp:id="BuyerId">
      <cp:ProcessSpecification cp:version="2.0"
        cp:name="PIP3A4RequestPurchaseOrder“ xlink:type="simple“ xlink:href= "http://www.rosettanet.org/processes/3A4.xml"/>
     <cp:Role cp:name="Buyer"
        xlink:href=
         "http://www.rosettanet.org/processes/3A4.xml#Buyer"/>
      <cp:ServiceBinding>
        <cp:Service>
          bpid:icann:rosettanet.org:3A4v2.0
        </cp:Service>
        <cp:CanSend>
          <cp:ThisPartyActionBinding cp:id="PCInc_ABID1"
            cp:action="PO Request Action"
            cp:packageId="PCInc_RequestPackage">
            <cp:ChannelId>asyncChannelA1</cp:ChannelId>
            <cp:BusinessTransactionCharacteristics
              cp:isNonRepudiationRequired="true“ cp:isSecureTransportRequired="true"
              cp:isAuthorizationRequired="true“ cp:timeToAcknowledgeReceipt="PT2H“ cp:timeToPerform="PT1D"/>
          </cp:ThisPartyActionBinding>
        </cp:CanSend>
      </cp:ServiceBinding>
    </cp:CollaborationRole>
  </cp:PartyInfo>
</cp:CollaborationProtocolProfile>
```

Discover Partner Information and Negotiate

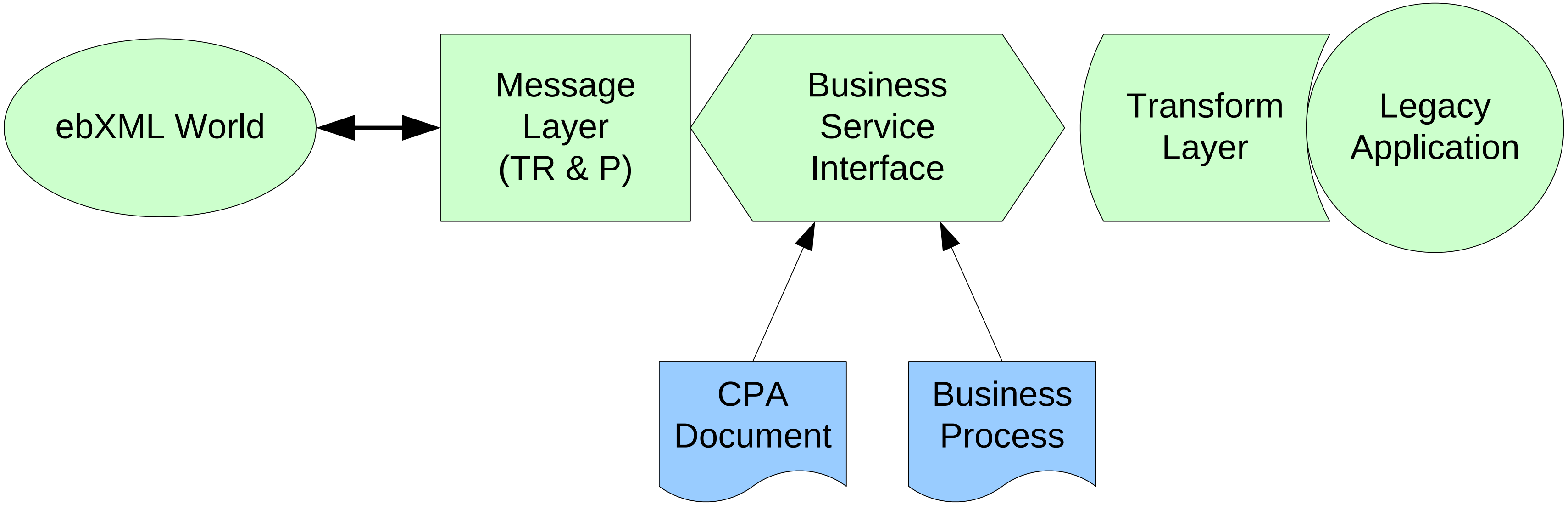


An example SOAP message header for sending a PO

```
<SOAP:Envelope
  xmlns:SOAP="http://schema.xmlsoap.org/soap/envelope/">
  <SOAP:Header
    xmlns:eb="http//www.ebxml.org/msg-header-2_0.xsd">
    <eb:MessageHeader id="123" eb:version="2.0"
      SOAP:mustUnderstand="1">
      <eb:From><eb:PartyId>123456</eb:PartyId></eb:From>
      <eb:To>
        <eb:PartyId eb:type="someType">987654</eb:PartyId>
        <eb:Role>
          http://rosettanet.org/processes/3A4.xml#seller
        </eb:Role>
      </eb:To>
      <eb:CPAId>uri:companyA-and-companyB-cpa</eb:CPAId>
      <eb:ConversationId>987654321</eb:ConversationId>
      <eb:Service eb:type="anyURI">
        bpid:icann:rosettanet.org:3A4v2.0
      </eb:Service>
      <eb:Action>Purchase Order Request Action</eb:Action>
      <eb:MessageData>
        <eb:MessageId>UUID-2</eb:MessageId>
        <eb:Timestamp>2000-07-25T12:19:05</eb:Timestamp>
        <eb:RefToMessageId>UUID-1</eb:RefToMessageId>
      </eb:MessageData>
      <eb:DuplicateElimination/>
    </eb:MessageHeader>
  </SOAP:Header>
  <SOAP:Body
    xmlns:eb="http//www.ebxml.org/msg-header-2_0.xsd">
    <eb:Manifest eb:version="2.0">
              ...
    </eb:Manifest>
  </SOAP:Body>
</SOAP:Envelope>
```

Implementing ebXML

- ebXML is just a set of specifications of collaborations and repositories for discovering business partners
  - An enterprise may build and deploy its own ebXML-compliant application to implement necessary roles in different collaborations
  - Use COTS ebXML compliant applications and components (from ERP vendors)
- Business Service Interface (BSI): a wrapper that enables a given party to participate properly in an ebXML exchange



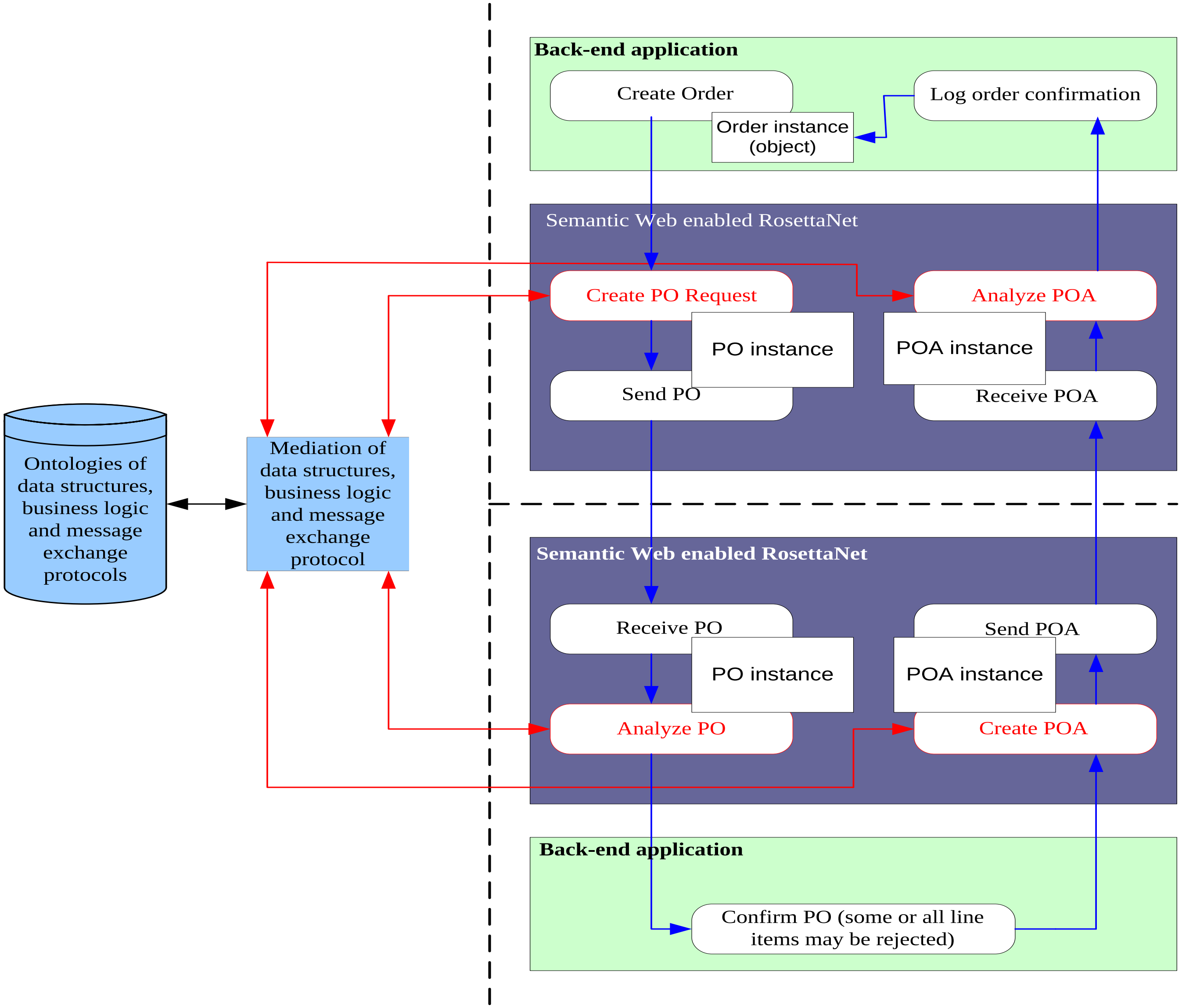
Characteristics of ebXML

- BPSS enables us to express collaboration protocols and agreements about protocols in a nice manner -> facilitates interoperation in cross-enterprise settings
- Limitations
  - Expressiveness: BPSS is limited to simple request-response protocols
  - Semantics: BPSS lacks a formal semantics, and it is not clear if specifications constructed by one party would have the same interpretations by another party

RosettaNet

- A consortium of information technology, semiconductor manufacturing, and telecommunications companies working to create and implement open e-business process standards
- The standards comprise a common e-business language that can align the interoperations among supply-chain partners on a global basis
- Each process is expressed as a PIP (Partner Interface Process) that defines the process of exchanging messages between two partners
- Distinction between PIP and BPSS
  - PIPs define specific processes (like a purchase-order process)
  - BPSS is a language for defining processes

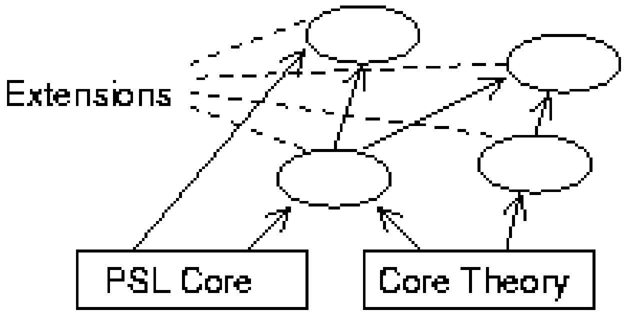
RosettaNet PIP for Creating a Purchase Order: The Content for ebXML


Back-end application
Create Order
Order instance (object)
Log order confirmation
Semantic Web enabled RosettaNet
Create PO Request
Analyze POA
PO instance
POA instance
Send PO
Receive POA
Ontologies of data structures, business logic and message exchange protocols
Mediation of data structures, business logic and message exchange protocol
Semantic Web enabled RosettaNet
Receive PO
Send POA
PO instance
POA instance
Analyze PO
Create POA
Back-end application
Confirm PO (some or all line items may be rejected)

PSL

- Process Specification Language
- http://www.mel.nist.gov/psl/
- Designed for describing or exchanging information among models of discrete processes, i.e., processes consisting of individually distinct events, tasks, or service invocations
- has a formally defined semantics in the language of first-order logic and represented using the Knowledge Interchange Format (KIF)

Fig 13.15 here



Coordination Frameworks for Web Services

406.622 Industrial Information Technology

Jonghun Park

jonghun@snu.ac.kr

Dept. of Industrial Engineering

Seoul National University

9/20/2007

Business Interchange Requirements

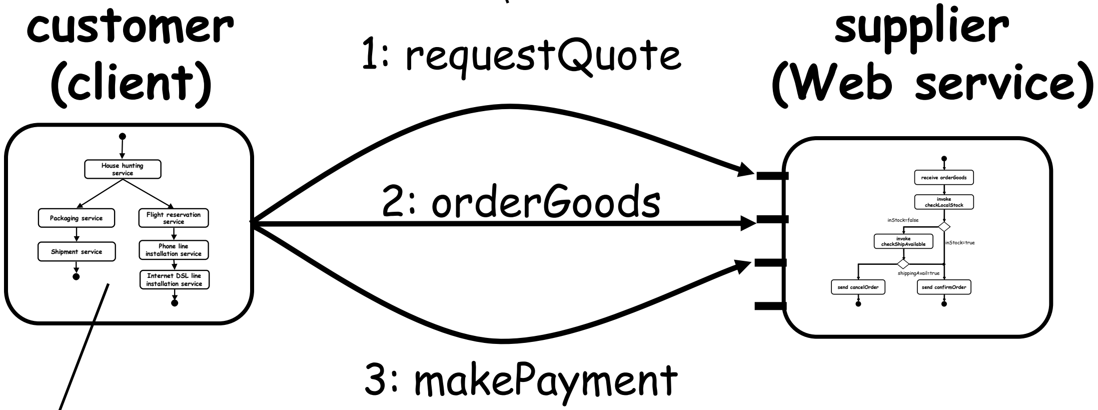
- The parties must agree on the **documents** to be exchanged, and the **semantics** of the documents (defined by XML, XML Schema, RDF, and OWL)
- The parties must agree on the **protocol** used to transmit a document (such as SOAP-RPC, asynchronous SOAP, or ebXML transport), the routing for the transmission, and the packaging of the document
- The parties must know each other’s **location**
- An **ordering** of the documents to be transmitted must be specified: a **conversation**

Need for service coordination protocols

- The basic web services infrastructure supports interactions where the client invokes a single operation on a Web service
- When the interaction involves **coordinated sequences of operations**, additional abstractions and tools are needed to ensure the **correctness** and **consistency**
- Moving from simple, independent invocations to sequences of operations where their order matters has important implications
  - Internal: The client must be able to execute relatively complex procedures to perform the different operations in the appropriate order. Furthermore, the client must be capable of maintaining context information
  - External: The interaction between the client and the server has to obey certain constraints. If these constraints are not followed, the Web service will be unable to process the messages and will return an error to the client

Example

The interaction between clients and services is often formed by a set of operation invocations (i.e., it is a *conversation*).
A service provider may support some conversations while disallowing others.



The internal business logic of clients and Web services must support the conversation, and maintain the state across different operation invocations belonging to the same conversation.

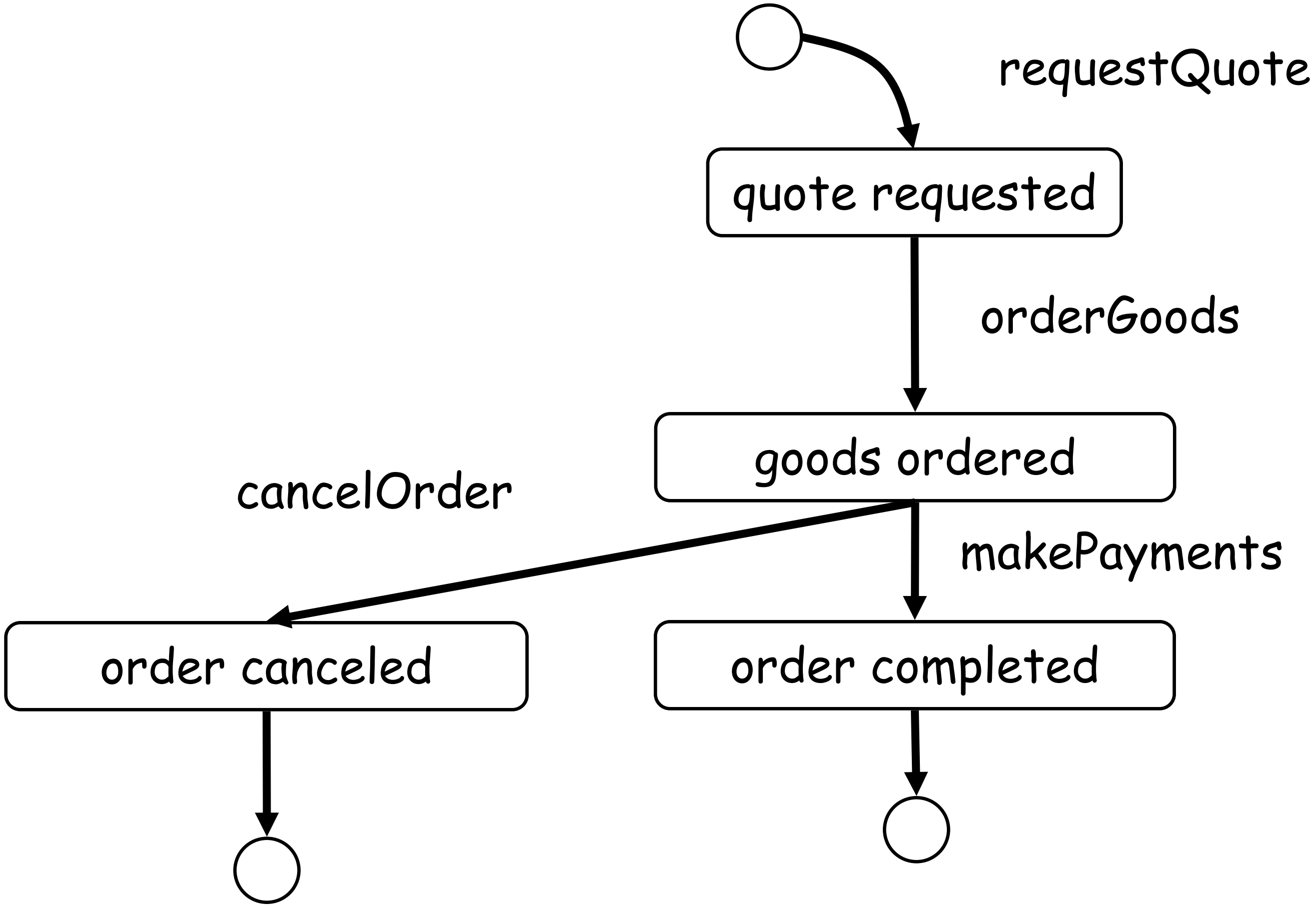
Issues

- **Conversation**: the sequences of operations (i.e., message exchanges) that could occur between a client and a service as part of the invocation of a Web service
- **Coordination protocol**: the specification of the set of correct and accepted conversations
- The problems
  - How to make it easy for developers to **specify complex procedures** that can implement the logic required to conduct a conversation
  - How a Web service can **describe the set of coordination protocols** it supports and **make the clients aware** of this information -> Needs to be advertised in registries

Conversation specification

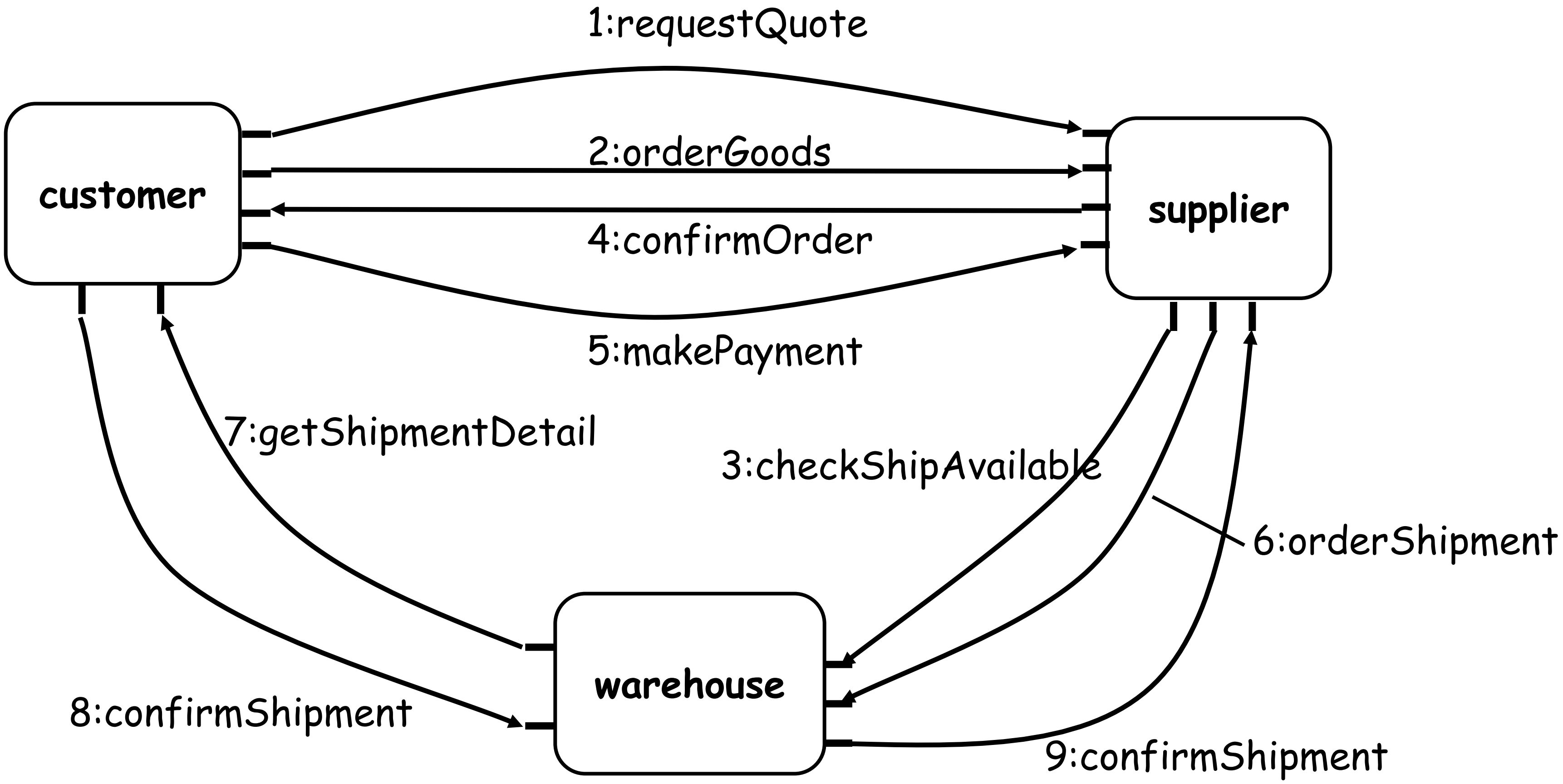
- A state machine
  - The states define each possible stage of a correct conversation (e.g., quote requested, goods ordered)
  - Each state can have one or more output transitions, each labeled with a different operation name corresponding to one of the operations offered by the Web service interface
  - A conversation is in one and only one state at any time
- Given the set of operations provided by a Web service interface, the state machine determines the set of correct conversations by defining which interface operation can be invoked, based on the conversation state

Conversation specification example



Conversations among multiple WSs

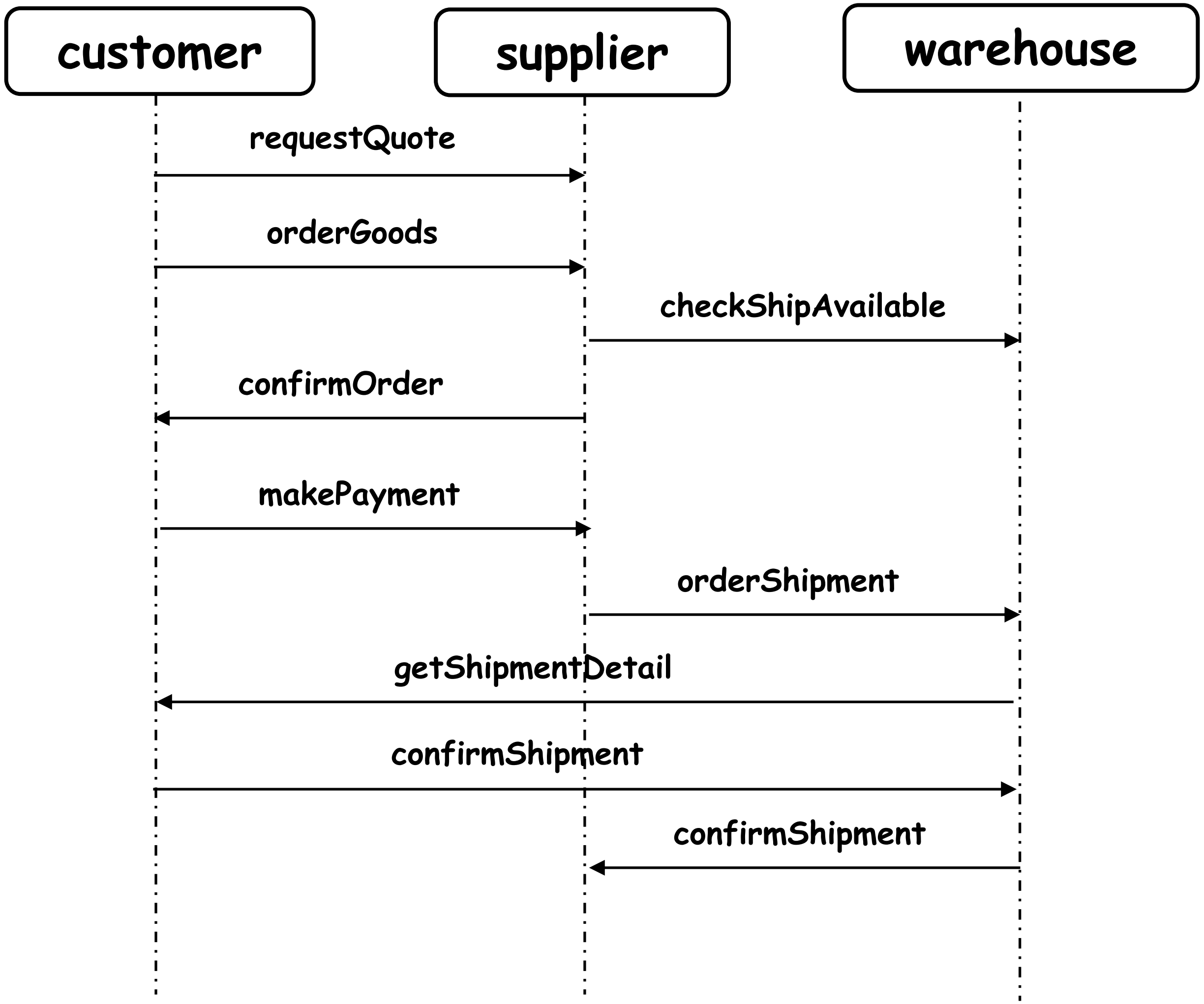
- Multi-party conversations
- Both sides impose constraints on how the conversation must proceed
- Independent of the number of Web services involved



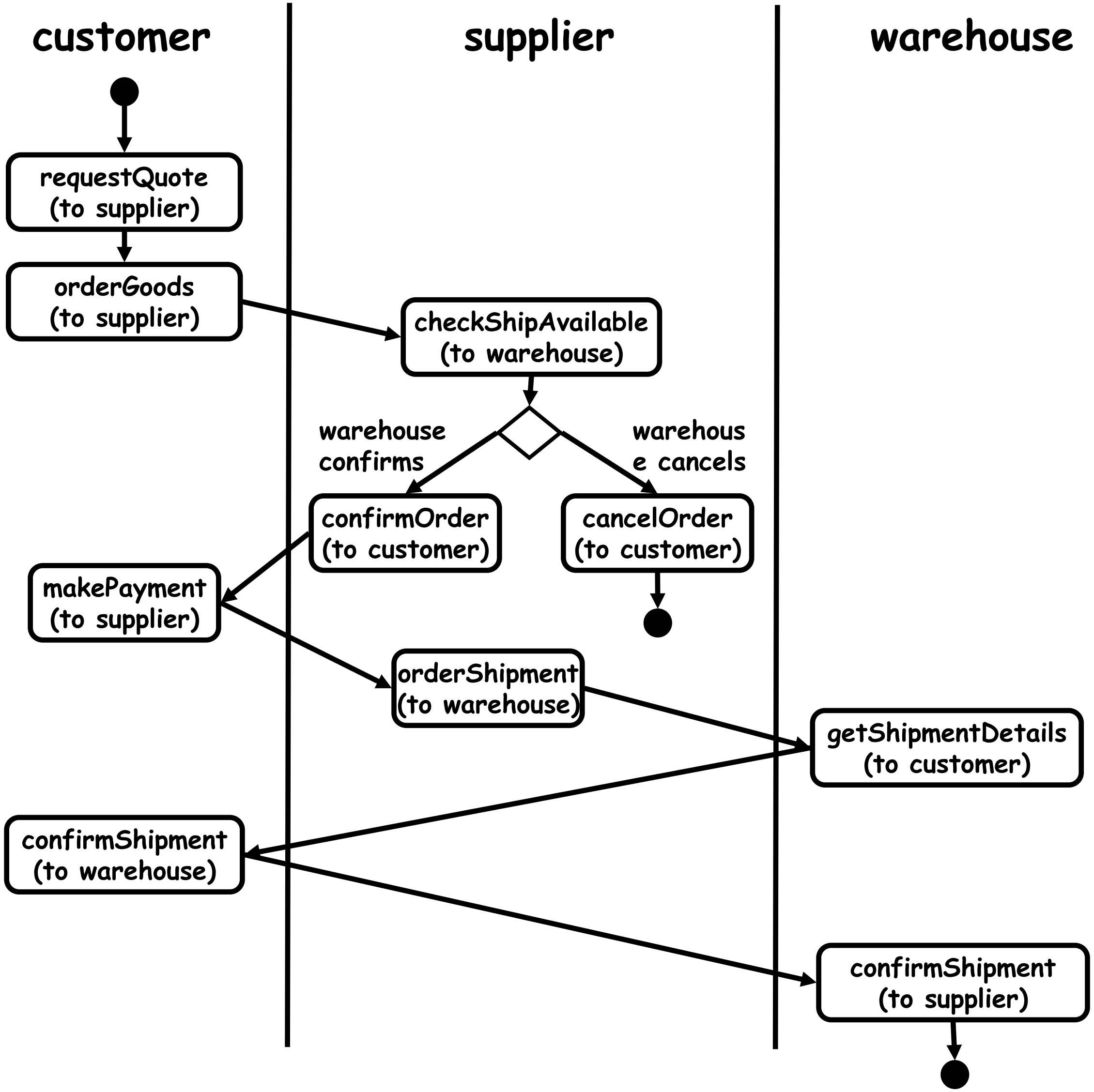
Specification of multi-party conversations

- Introduce the definition of a protocol in terms of **roles** and of **message exchanges** among entities playing those roles, along with **constraints** on the order in which such exchanges should occur
- Use state machine
    - Associate each transition of a state machine not just an operation name, but a triplet **<invoker, operation, provider>**, meaning that the state transition occurs when the invoker calls the specified operation offered by the provider
- Use sequence diagrams
    - Limitation: when the protocol becomes complex, one sequence diagram does not suffice -> Multiple sequence diagrams for alternative conversations
- Use activity diagrams
    - Can model alternative executions as well as parallel executions

A sequence diagram example



An activity diagram example

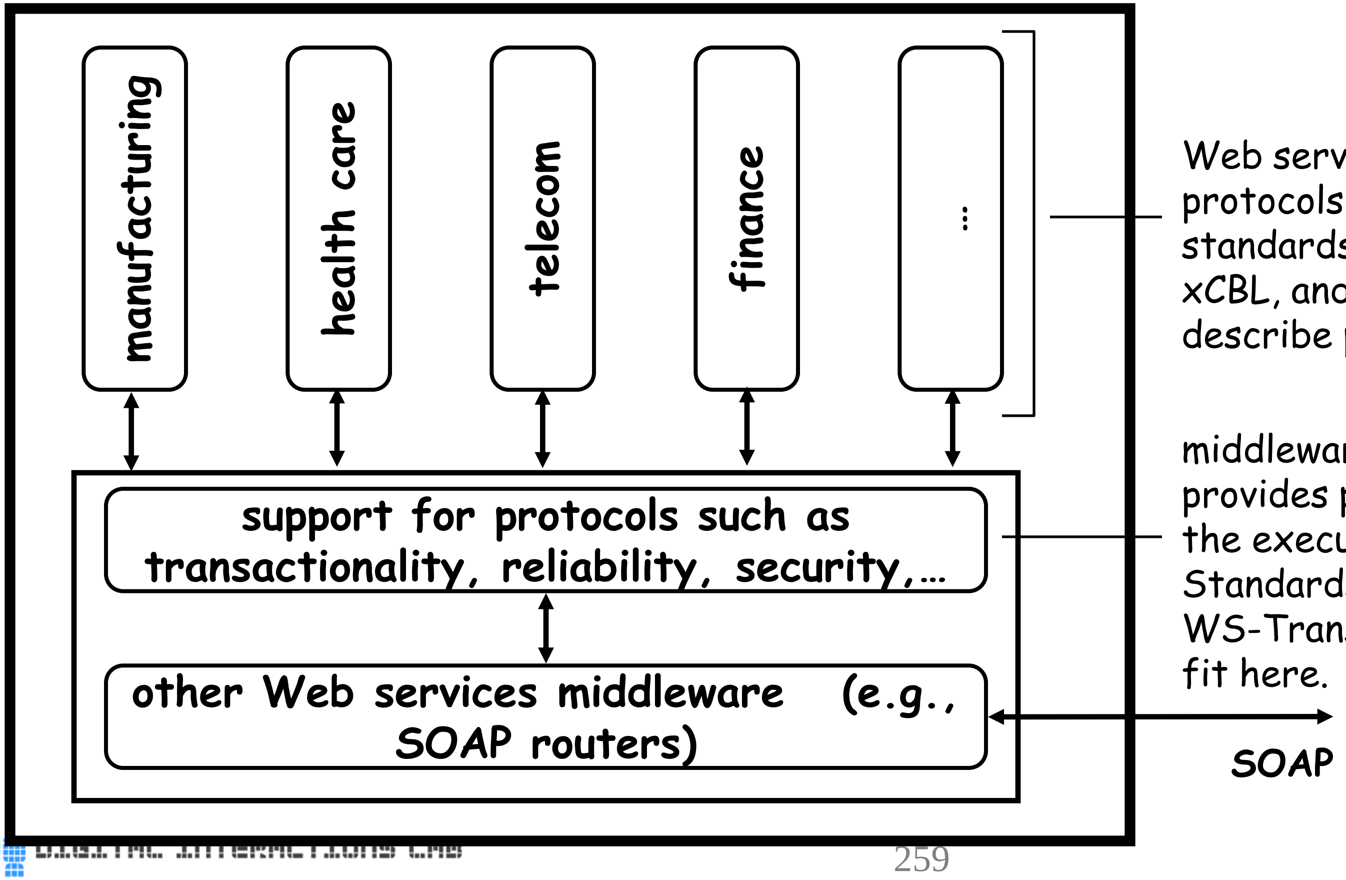

customer
supplier
warehouse
requestQuote (to supplier)
orderGoods (to supplier)
checkShipAvailable (to warehouse)
warehouse confirms
warehous e cancels
confirmOrder (to customer)
cancelOrder (to customer)
makePayment (to supplier)
orderShipment (to warehouse)
getShipmentDetails (to customer)
confirmShipment (to warehouse)
confirmShipment (to supplier)

Classification of Web service protocols

- Vertical protocols
  - Specific to business areas
  - xCBL, RosettaNet, ebXML, …

- Horizontal protocols
  - Define a common infrastructure independent of the application area
  - WS-Coordination, WS-Transaction, …

service provider



manufacturing

health care

telecom

finance

…

support for protocols such as transactionality, reliability, security,…

other Web services middleware (e.g., SOAP routers)

Web services executing vertical protocols. The main focus of standards such as RosettaNet, xCBL, and part of ebXML is to describe protocols at this level.

middleware for horizontal protocols provides properties and guarantees to the execution vertical protocols. Standards such as WS-Coordination, WS-Transaction, and part of ebXML fit here.

SOAP messages

Infrastructure for coordination protocols

- Conversation controllers
- Generic protocol handlers
- Standardization requirement for coordination protocols

Conversation controllers

- Tools facilitating the execution of conversations
- Provides
  - **conversation routing**: the problem of dispatching messages to the appropriate internal object
  - protocol compliance **verification**: if an operation invoked by a client in the context of a conversation is not allowed, then the conversation controller can return an error message

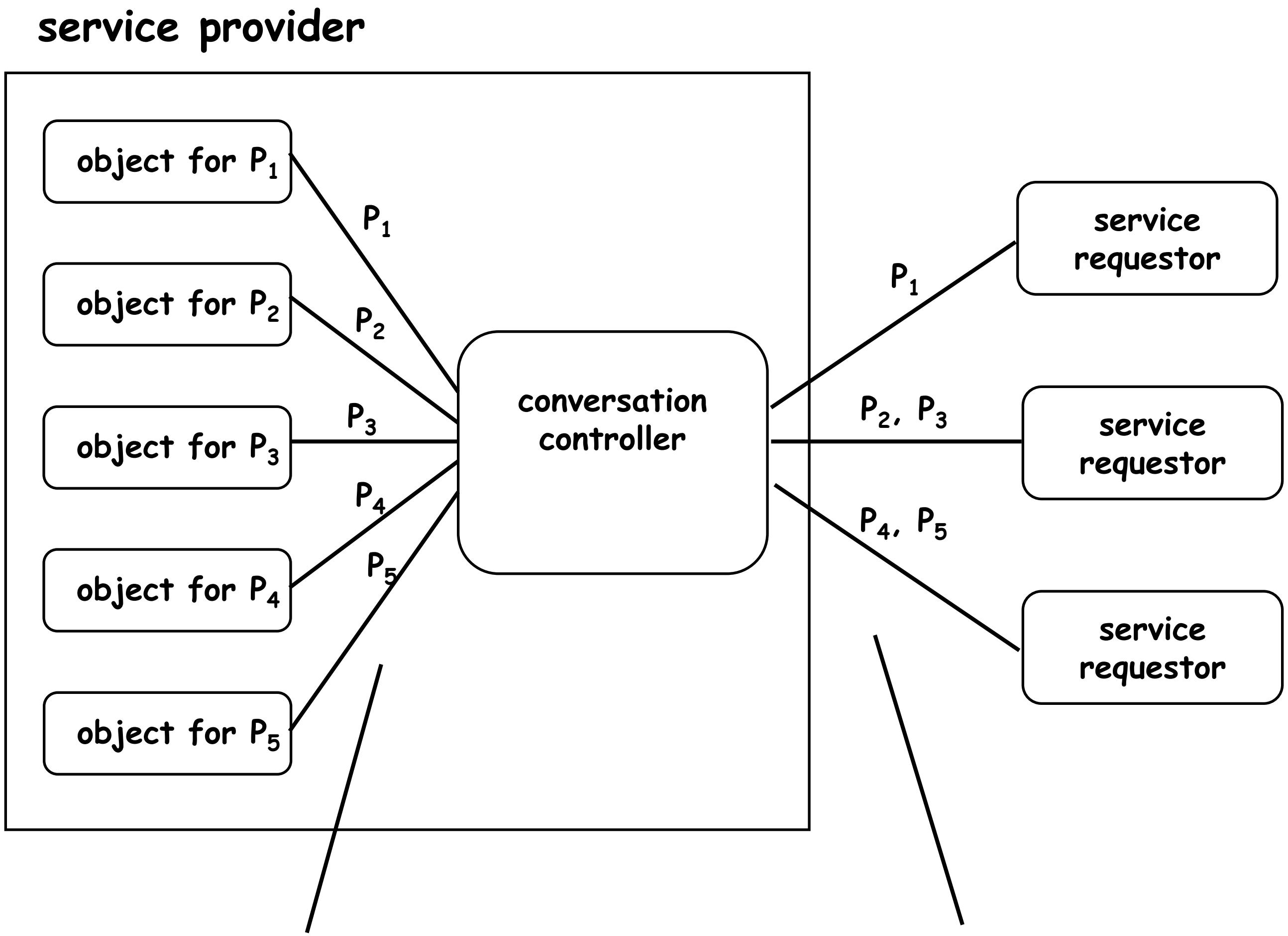
Conversation routing

- A web service that is engaged in several different executions of a coordination protocol P
- If all clients invoke operations provided by the same port, then they are sending messages to the same Web address
- When the Web service receives the message, it must determine **to which conversation the message belongs**, as each conversation has a state, and the way each message is processed depends on the conversation state
- Alternatives
  - A single object implementation
  - A conversation controller

Conversation controller

- Creating **one object for each conversation** and letting the infrastructure handle the routing of messages to the appropriate object
- The controller can handle the dispatching of messages pertaining to each conversation instance to the appropriate EJB
- The controller can accomplish this by
  - generating an **identifier** each time a message that starts a new conversation is received
  - including a **unique identifier** in the header of all messages exchanged within a conversation
  - requiring standardization: all interacting parties are expected to know how the conversation identifier is embedded into the messages and to insert it in every message they send

Conversation controller

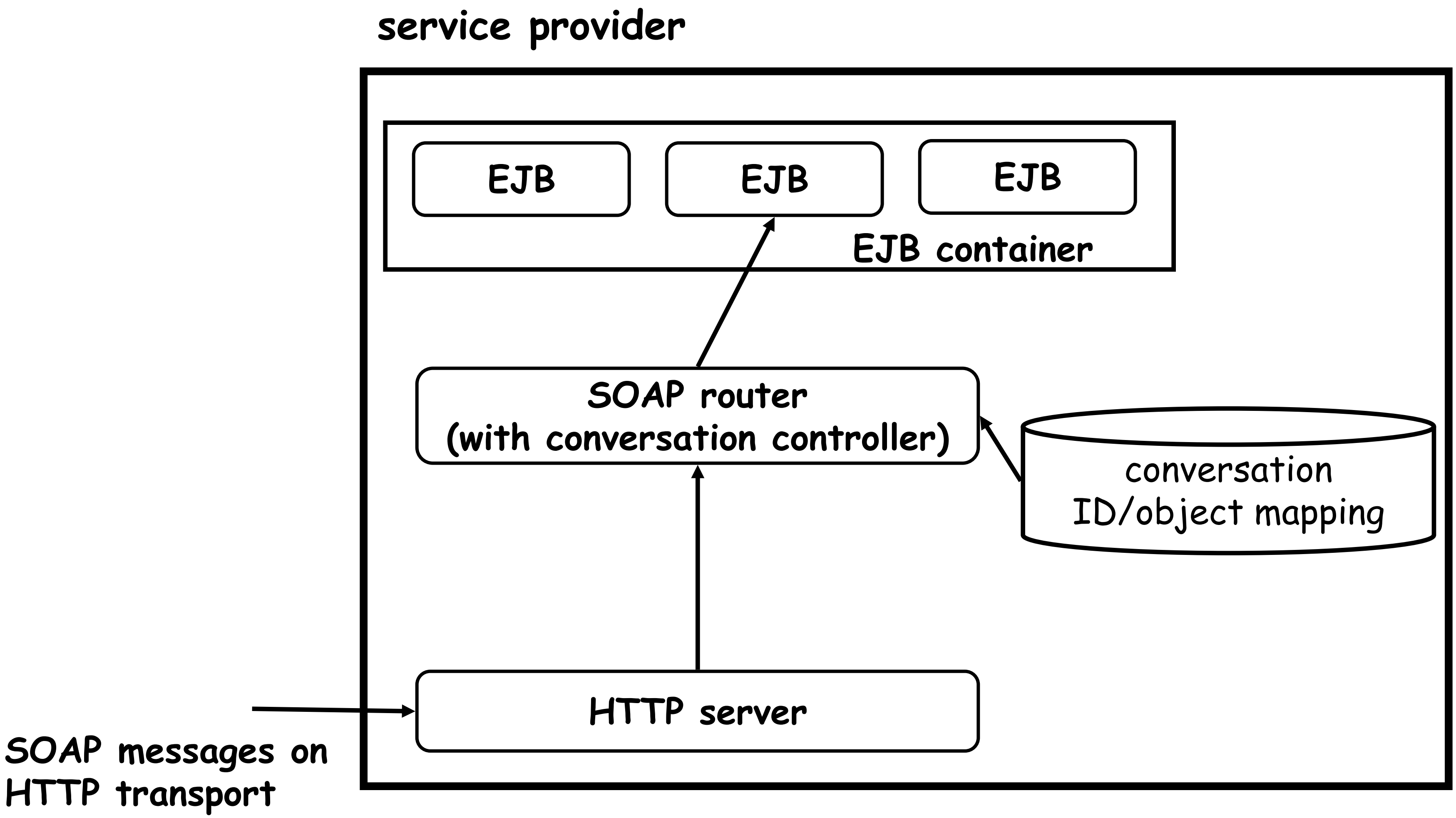


the controller dispatches messages to the appropriate implementation object

clients invoke operations at the same address

The role of conversation controller

- The controller, based on a <conversation identifier, object reference> mapping, determines the EJB to which the message should be delivered

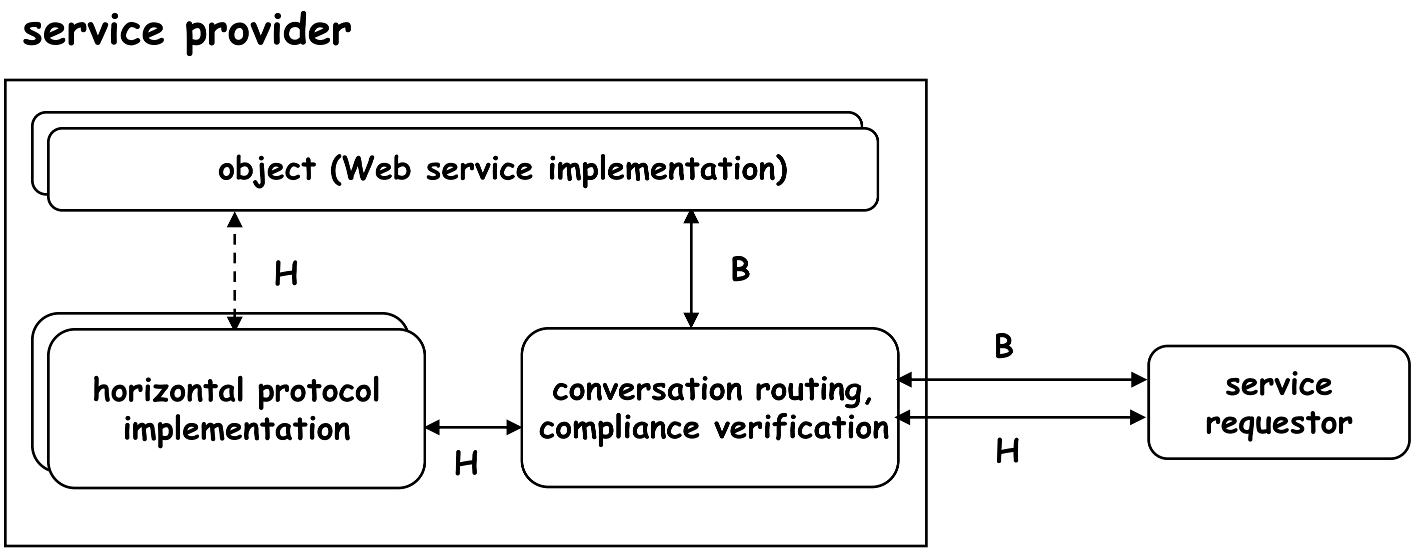


Generic protocol handlers

- A module that can include **protocol-specific logic** to act and generate messages in accordance with the rules defined by the protocol
- The protocol handler can support protocol execution in two forms
  - The handler receives, interprets, and sends protocols messages automatically, without intervention by the Web service (e.g., reliable message delivery)
  - The handler and the Web service share the burden of implementing the protocol (e.g., in 2PC, the logic for implementing the decision of whether to commit or abort is left to the participating Web services)

Interaction among conversation controllers, protocol handlers, and Web services

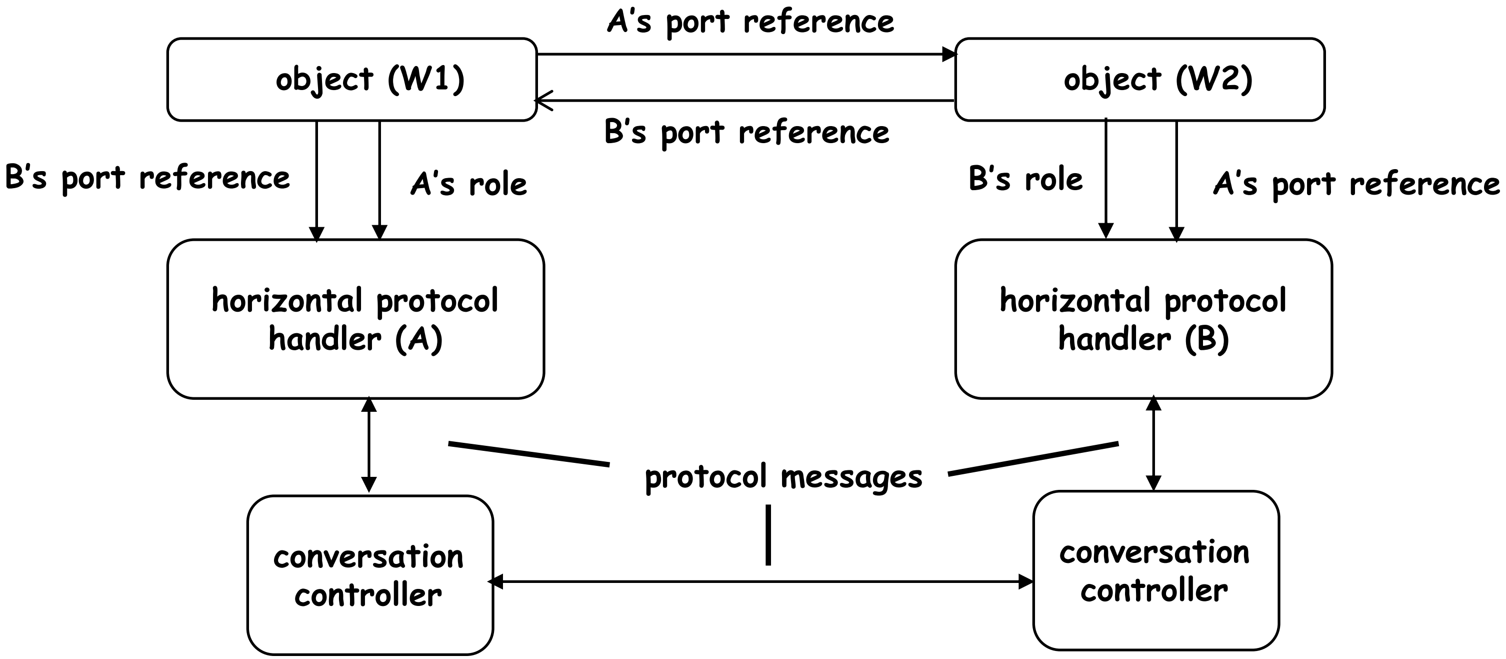
- The conversation controller routes business protocol messages to the appropriate Web services implementation, which contains the business logic implementing the protocol
- Messages related to horizontal protocols are instead routed to the protocol handlers


service provider
object (Web service implementation)
H
B
horizontal protocol implementation
H
conversation routing, compliance verification
B
H
service requestor

B: conversation compliant with a business protocol
H: conversation compliant with an horizontal protocol

Port references

- When the infrastructure handles the protocol implementation automatically, it should be told the role it has to take in implementing the protocol
- W1 and W2, both delegating the execution of a protocol to the protocol handlers A and B in their infrastructure
- Before A and B can exchange the protocol messages with each other, A needs to know the port reference of B and vice-versa



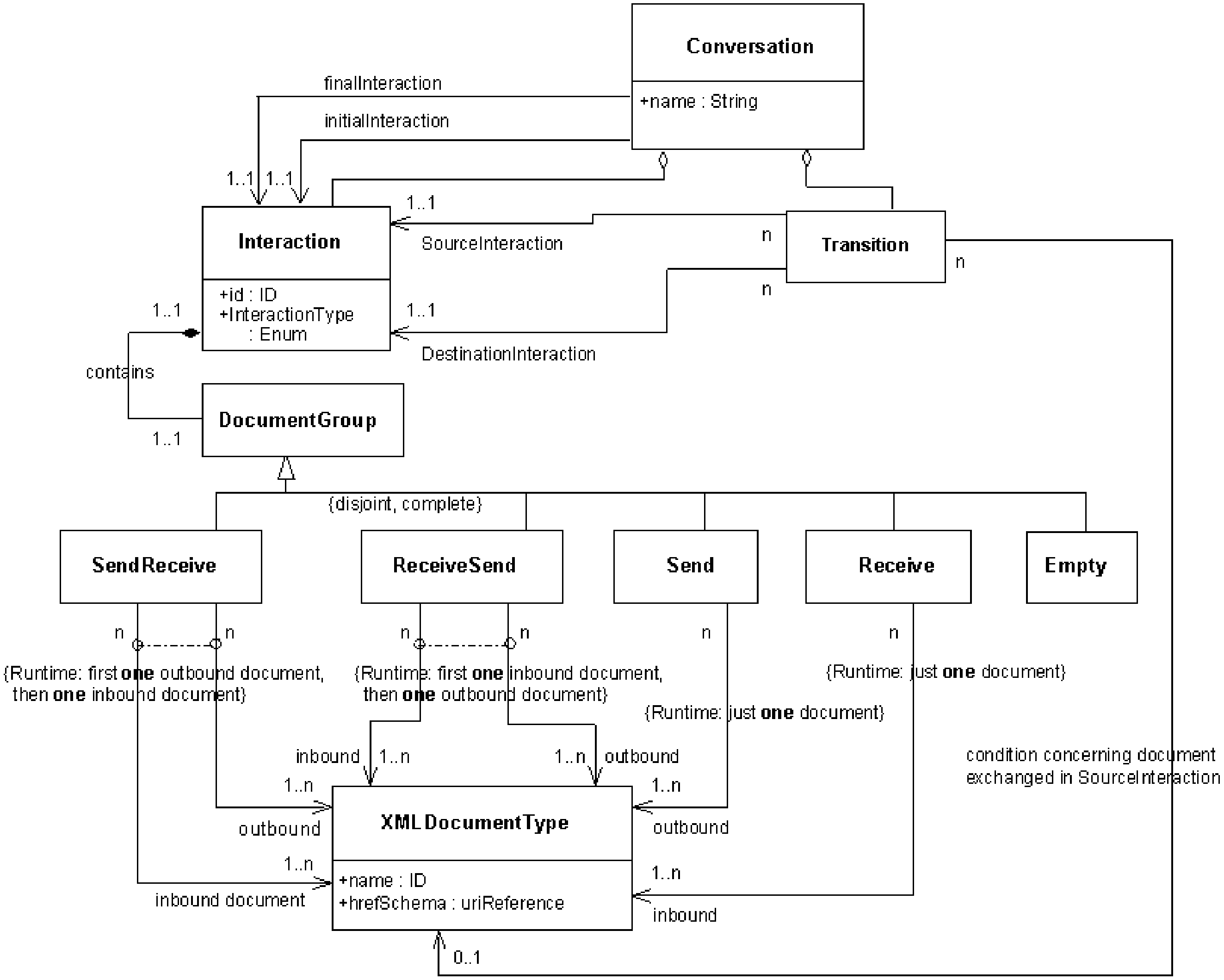
Standardization requirements

- A way to generate and transport unique conversation identifiers in the headers of SOAP messages (e.g., ebXML)
- A framework and a set of protocols (i.e., meta-protocols) whose purpose is to agree on such aspects as which protocol should be executed and how it is coordinated
- Horizontal protocols to be standardized, so that additional properties can be provided by the WS middleware
- A standard protocol languages, so that conversation controllers can interpret protocol specifications and verify protocol compliance

WSCL

- Web services conversation language
- A submission by HP in 2002
  - W3C Note: http://www.w3.org/TR/wscl10/
- allows the business level conversations or public processes supported by a web service to be defined
- specifies the sequencing of XML documents (as well as specifications for the documents themselves) being exchanged between a web service and a user of that service
- can provide protocol binding information for abstract interfaces or can specify the abstract interfaces supported by a concrete service
- describes the interactions with just one service
- WSCL conversation definitions are themselves XML documents
- models only business-level interactions and not how an exchange of business-level documents is carried out by lower-level messaging protocols
  - the exchange of one business document might require several actual messages to be exchanged
- expected to be replaced by WS-CDL

Concepts in WSCL



Structure of a WSCL specification

- Document type definitions
  - specify what types of XML documents will be exchanged by parties
- Interactions
  - exchanges of documents between a service and a client
  - one of the 5 types: Empty, Send, Receive, SendReceive, ReceiveSend

```
<Interaction interactionType="SendReceive"
id="Payment">
 <OutboundXMLDocument id="Invoice"
    hrefSchema="http://sc.edu/InvoiceRS.xsd"/>
 <InboundXMLDocument id="Payment"
    hrefSchema="http://ncsu.edu/Payment.xsd">
 </InboundXMLDocument>
</Interaction>
```

- Transitions
  - the order of the interactions

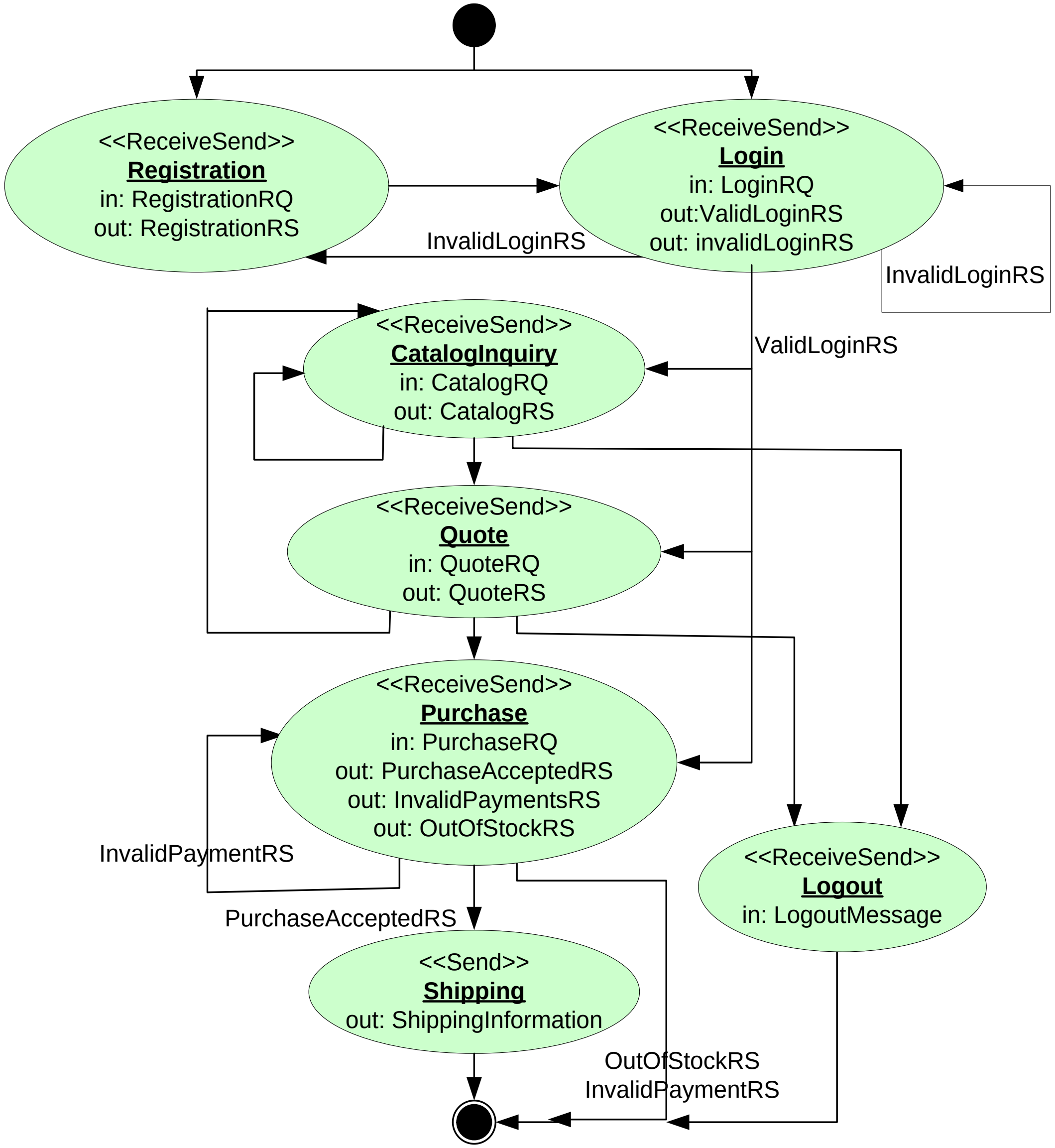
```
<Transition>
   <SourceInteraction href="Quote"/>
   <DestinationInteraction href="Purchase"/>
</Transition>
<Transition>
   <SourceInteraction href="Quote"/>
   <DestinationInteraction href="CatalogInquiry"/>
</Transition>
```

- Conversation: a list of all interactions and transitions in it, a starting interaction, and an ending interaction

Well-Formed Conversations

- All interactions are reachable from the initial interaction
- The final interaction is **reachable** from all interactions
- If a transition from interaction A to interaction B specifies a SourceInteractionCondition, then all transitions from A to B do so
- The final interaction and transitions to the final interaction unambiguously clarify for each participant when a conversation is finished

Example Conversation Definition



Example WSCL Specification

```
<Conversation name="StoreFrontServiceConversation"  xmlns="http://www.w3.org/2002/02/wscl10" initialInteraction="Start"
        finalInteraction="End" >
    <ConversationInteractions>
        <Interaction interactionType="ReceiveSend" id="Login">
            <InboundXMLDocument hrefSchema=“http://conv1.org/LoginRQ.xsd”
                id="LoginRQ"/>
            <OutboundXMLDocument hrefSchema=“http://conv1.org/ValidLoginRS.xsd”
                id="ValidLoginRS"/>
            <OutboundXMLDocument id="InvalidLoginRS"
        hrefSchema=“http://conv1.org/InvalidLoginRS.xsd”/>
        </Interaction>
        …
        <Interaction interactionType="Empty" id="Start" />
        <Interaction interactionType="Empty" id="End" />
    </ConversationInteractions>
 <ConversationTransitions>
        <Transition>
            <SourceInteraction href="Start"/>
            <DestinationInteraction href="Login"/>
        </Transition>
        …
        <Transition>
            <SourceInteraction href="Login"/>
            <DestinationInteraction href="Registration"/>
            <SourceInteractionCondition href="InvalidLoginRS"/>
        </Transition>
        <Transition>
            <SourceInteraction href="Logout"/>
            <DestinationInteraction href="End"/>
        </Transition>
    </ConversationTransitions>
</Conversation>
```

Limitations of WSCL

- Limited to two participants
- WSCL provides excellent graph primitives for describing control flows, but does not have specific constructs for **iteration** or **recursion**
- Conversations are modeled as scripted procedures (graphs), but the procedures are not flexible
- Cooperation is not supported
- Exception handling is done only at low level

WSCI: Web Service Choreography Interface

- An interface description language for business processes
- provides a **global message-oriented view** of the choreographed interactions among a collection of web services
- describes the flow of messages exchanged by a web service that is interacting with other services according to a choreographed pattern
- characterizes the **externally observable behavior** of the web service, not its internal operation
- offers interoperability across BPML, BPEL4WS, ebXML’s BPSS, and WfMC’s XPDL
- is an enhancement to WSDL, and a WSCI specification is intended to be part of a WSDL document describing a web service
- A service can participate in several different conversations at the same time, with the correlate element used to manage them and associate messages with the proper conversation
- supports both atomic transactions and open-nested transactions, the latter of which can be compensated when exceptions occur

Example WSCI Add-In to WSDL

```
<correlation name="quotationCorrelation“ property="tns:quotationID“/>
 <interface name="StockQuoteWS">
   <process name="ProvideStockQuote" instantiation="message">
     <sequence>
       <action name="ReceiveLogin“ role="tns:StockQuoteWS“
           operation="tns:QuoteToUser/LogIn"/>
       <action name="ReceiveStockQuoteRequest" role="tns:StockQuoteWS"
           operation="tns:QuoteToUser/ProvideQuote">
          <correlate correlation="tns:quotationCorrelation"/>
          <call process="tns:LookupPrice"/>
       </action>
       <action name="ReceiveLogout“ role="tns:StockQuoteWS“
           operation="tns:QuoteToUser/LogOut"/>
     </sequence>
   </process>

   <process name="LookupPrice" instantiation="other">
      <action name="QueryNYSE" role="tns:StockQuoteWS“
         operation="tns:QuoteToUser/QueryNYSE"/>
    </process>
 </interface>
```

WSCI Example for Transaction Compensation

```
<sequence>
  <context>
    <transaction name="buyStock" type="atomic">
      <compensation>
        <action name="NotifyUnavailable" role="NYSE“            operation="tns:NYSEtoBroker/NotifyUnavailable"/>
      </compensation>
    </transaction>
  </context>
  <action name="BuyShare" role ="Broker“ operation="tns:BrokerToNYSE/BuyShare"/>
  <while name="BuyShares">
    <condition>defs:fundsRemain</condition>
    <action name="BuyShare" role ="Broker“ operation="tns:BrokerToNYSE/BuyShare">
        <correlate correlation="defs:buyingCorrelation"/>
    </action>
  </while>
</sequence>
<!-- Compensating Behavior for the Above Transaction -->
<exception>
  <onTimeout property="tns:expiryTime" type="duration" reference="tns:BuyShares@end">
     <compensate transaction="tns:buyStock"/>
  </onTimeout>
</exception>
```

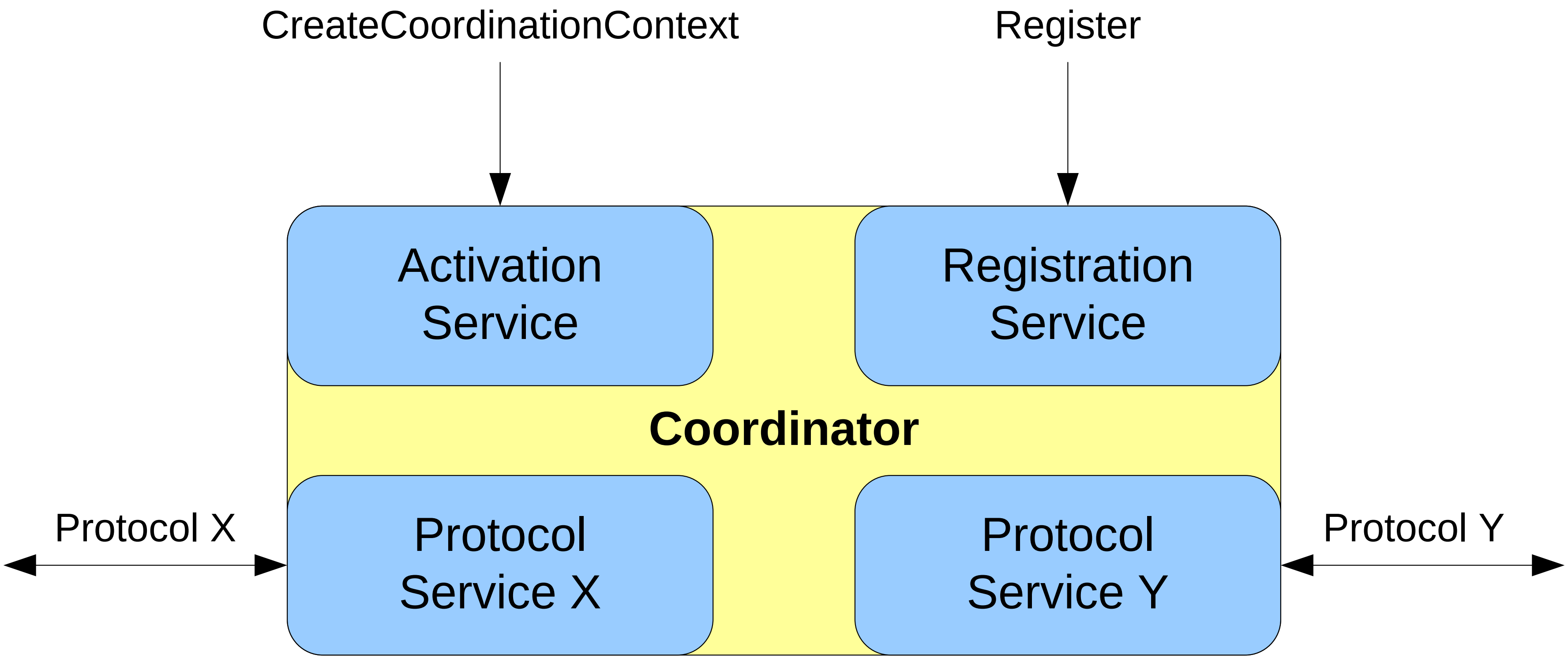
Web services coordination

- Coordination service: A service whose job is to coordinate the activities of the web services that are part of the business process
- Multiple participants can hide their proprietary protocols that reach agreement on the outcome of their activities
- Protocol
  - A set of well-defined messages that are exchanged between the web services participating in a process
- Context
  - A uniquely identified, conceptually coherent **activity** that includes the services being coordinated
- Application
  - An executing program instance at one site

WS-Coordination

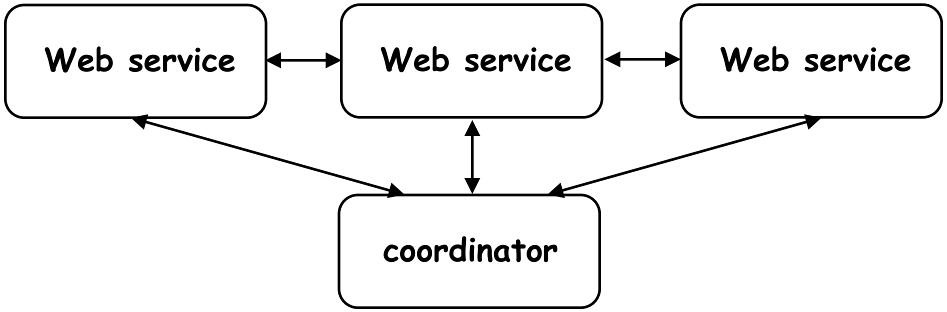
- Initially proposed by IBM, MS, and BEA in Aug. 2002
- a framework for supporting coordination protocols
- Defines **SOAP extensions** that are necessary to achieve coordination
- Defines meta-protocols for creating **coordination contexts** (Activation) and for **binding coordinators** and **participants** to each other (Registration)
- Defines a basic set of middleware components as well as their interfaces for implementing central or distributed coordination
- Standardizes
  - A method for **passing a unique identifier** between interacting Web services -> **coordination context**
  - A method for informing a protocol handler about the **port** of a Web service that participates in a conversation -> **registration** interface
  - A method for informing a protocol handler about the **role** it should assume in a conversation -> **activation** interface
- But, it is not a language for describing coordination protocols

WS-Coordination Service

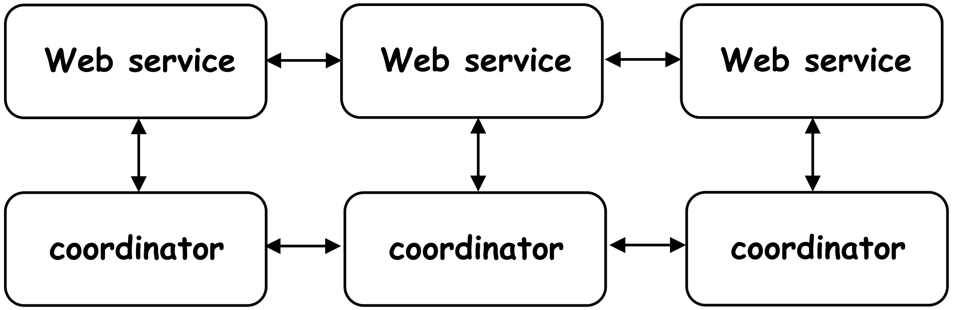


Components of WS-Coordination

- Basic entities: coordinator(s), participants



(a) central coordination



(b) distributed coordination

Abstractions in WS-Coordination

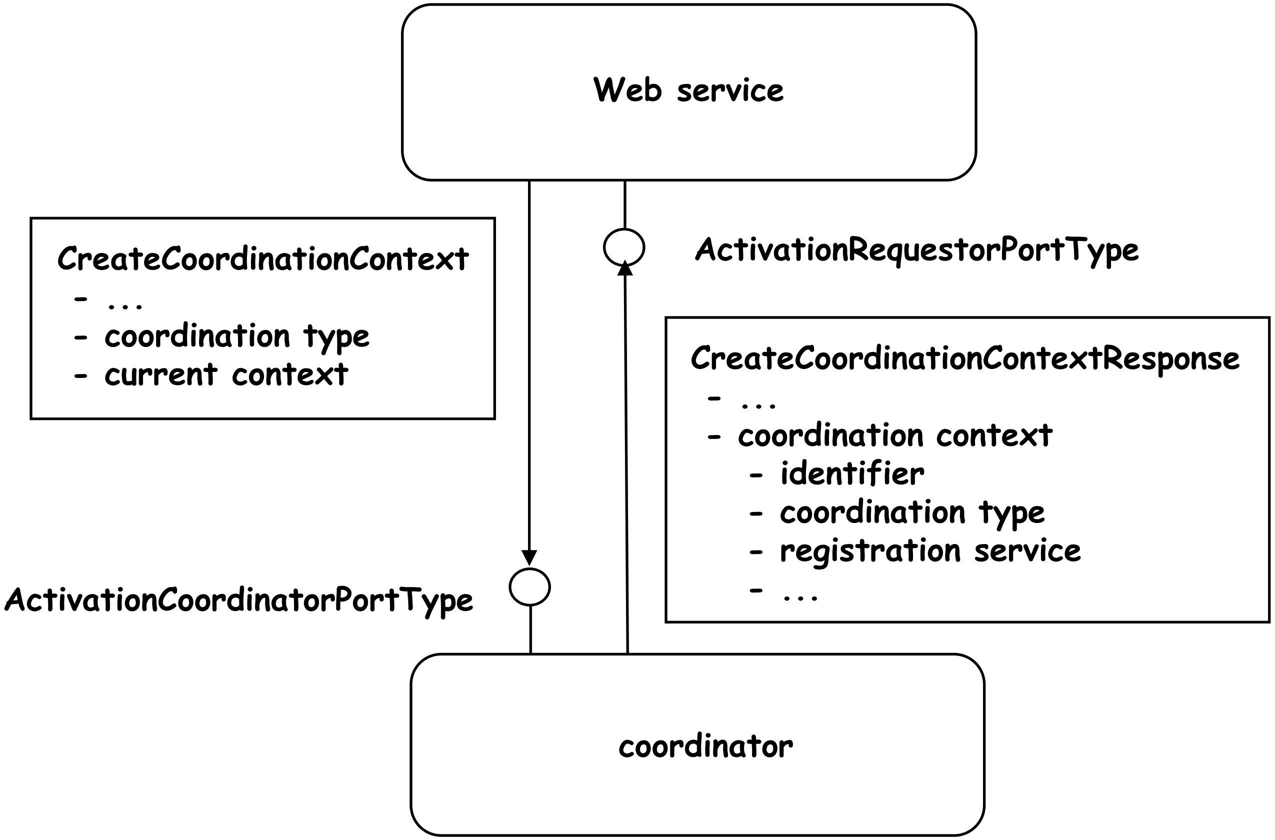
- Coordination protocol
  - A set of rules governing conversations between a coordinator and its participants (e.g., 2PC)
- Coordination type
  - **A set of coordination protocols**, logically related to each other
  - Example: Atomic transaction coordination type = 2PC + outcome notification protocol
- Coordination context
  - A data structure used to **mark messages belonging to the same conversation**
  - Included in the message headers
  - Contains a field that identifies the **coordination type** and a field that uniquely identifies the **instance** of that coordination type

Forms of interactions in WS-Coordination

- Activation
  - A participant requests a coordinator to **create a new coordination context**
- Registration
  - A participant **registers as a coordination protocol participant** with a coordinator
- Protocol-specific interactions
  - The coordinator and its participants **exchange messages** that are specific to a coordination protocol
- Note: Interactions for activation and registration are independent of the type of coordination

Port types in WS-Coordination: Activation

- Activation
  - ActivationCoordinatorPortType (by coordinator)
  - ActivationRequestorPortType (by participant)



Web service

ActivationRequestorPortType

CreateCoordinationContext
- ...
- coordination type
- current context

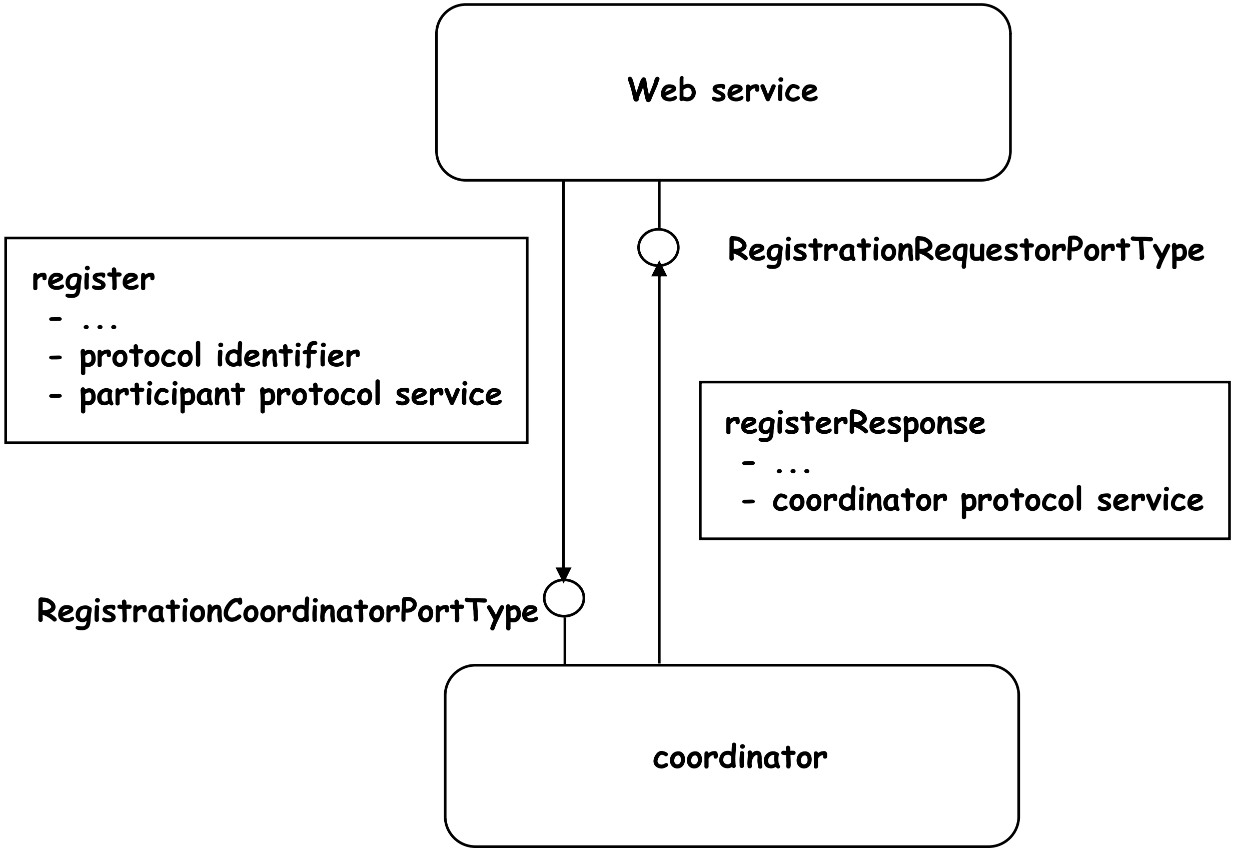
CreateCoordinationContextResponse
- ...
- coordination context
  - identifier
  - coordination type
  - registration service
  - ...

ActivationCoordinatorPortType

coordinator

Port types in WS-Coordination: Registration

- Registration
  - RegistrationCoordinatorPortType
  - RegistrationRequestorPortType



Web service

RegistrationRequestorPortType

register
- ...
- protocol identifier
- participant protocol service

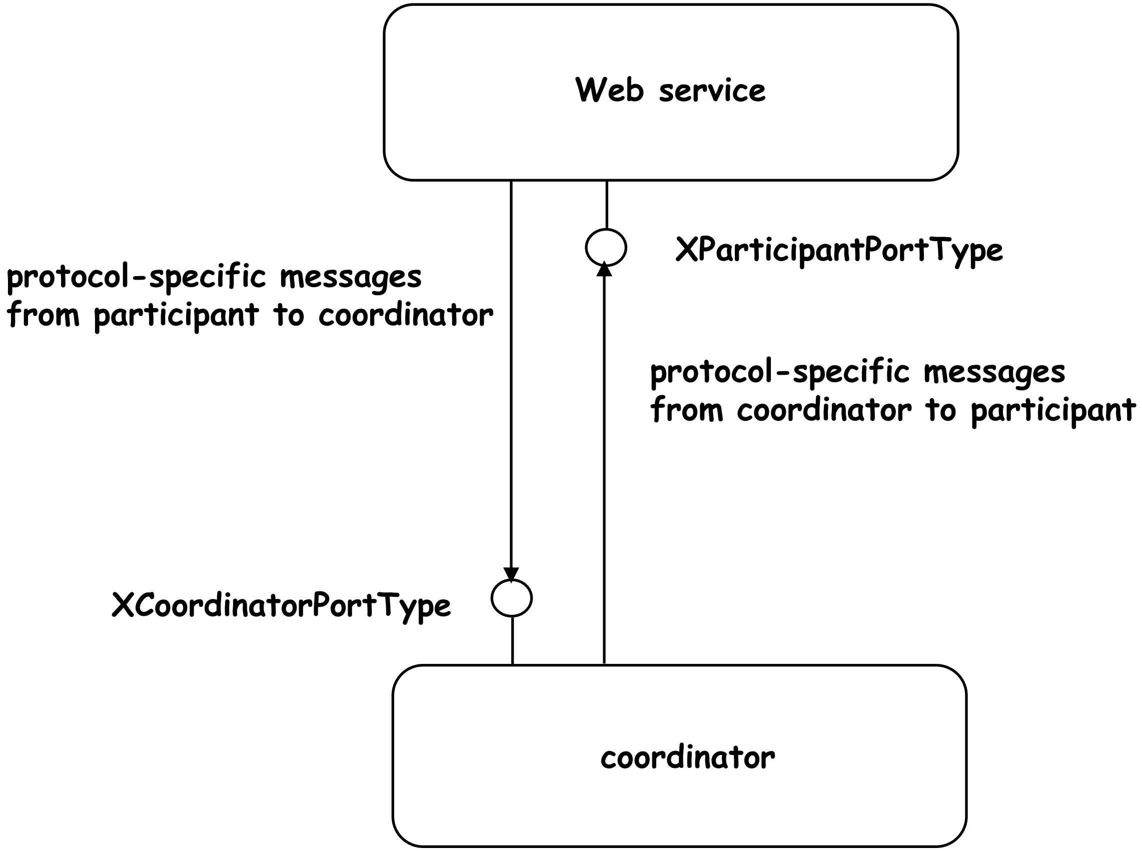
registerResponse
- ...
- coordinator protocol service

RegistrationCoordinatorPortType

coordinator

Port types in WS-Coordination: Protocol-Specific

- For each protocol X
  - XCoordinatorPortType
  - XParticipantPortType

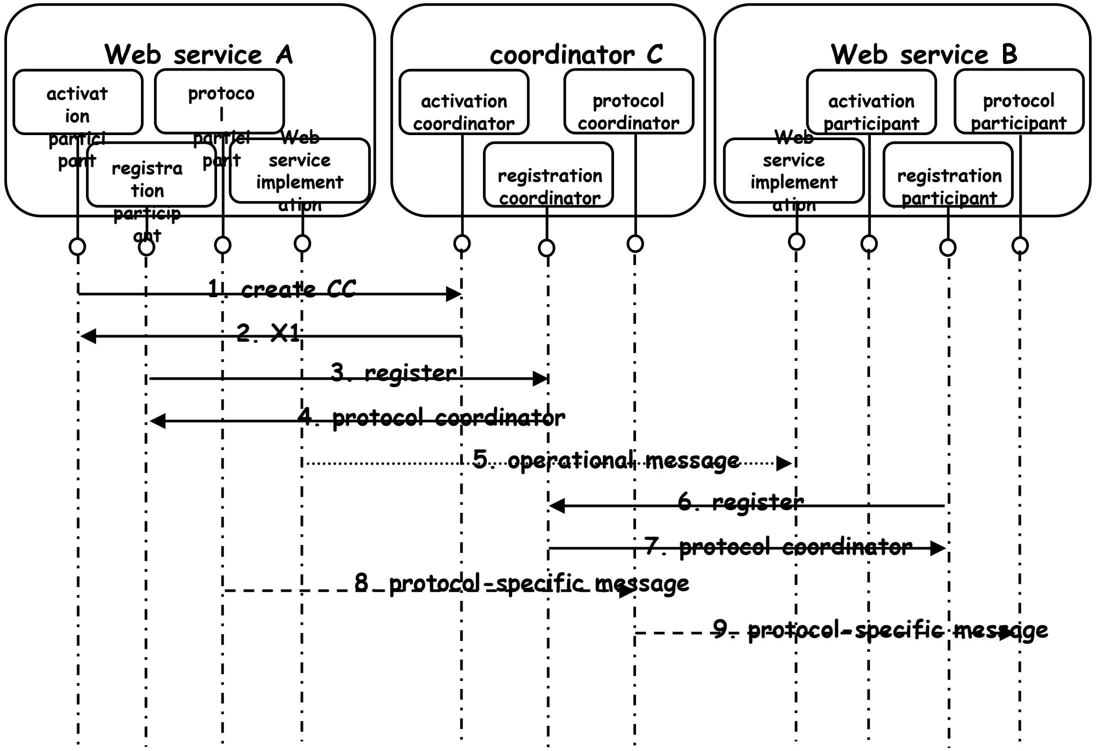


Web service
XParticipantPortType
protocol-specific messages from participant to coordinator
protocol-specific messages from coordinator to participant
XCoordinatorPortType
coordinator

Central coordination

- All the participating Web services need to **agree on who the coordinator** for a particular conversation is
- Consider Web services A and B that participate in a coordination protocol X
- Assume a single coordinator C

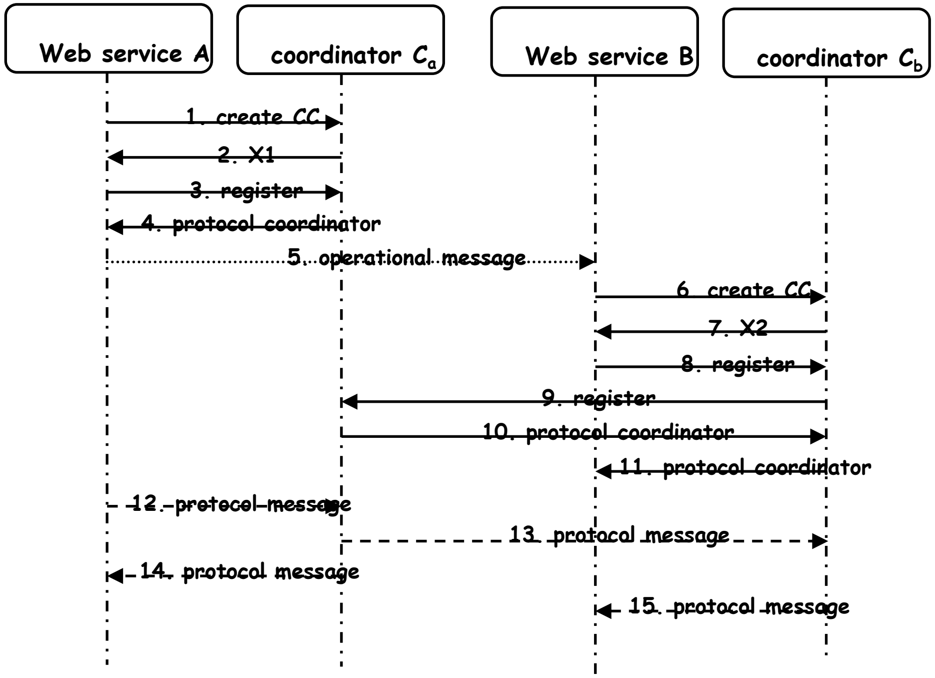
Example of central coordination



Operational messages: dotted lines
WS-Coordination messages: solid lines
Protocol-specific messages: dashed lines

1. A initiates a coordination instance by asking C to create a new coordination cotext
2. C returns a coordination context X1 to A. X1 contains a reference to C’s RCPT
3. A registers itself as a participant of protocol X with C by passing a reference to its XPPT
4. C returns a reference to its XCPT to A
5. A sends a message to B within the scope of the coordination. To indicate the scope, it includes X1 as part of the SOAP header
6. B retrieves C’s RCPT reference from X1 and uses it to register with C as a participant of X. B also passes its own XPPT to C as part of registration
7. C returns a reference to its XCPT to B

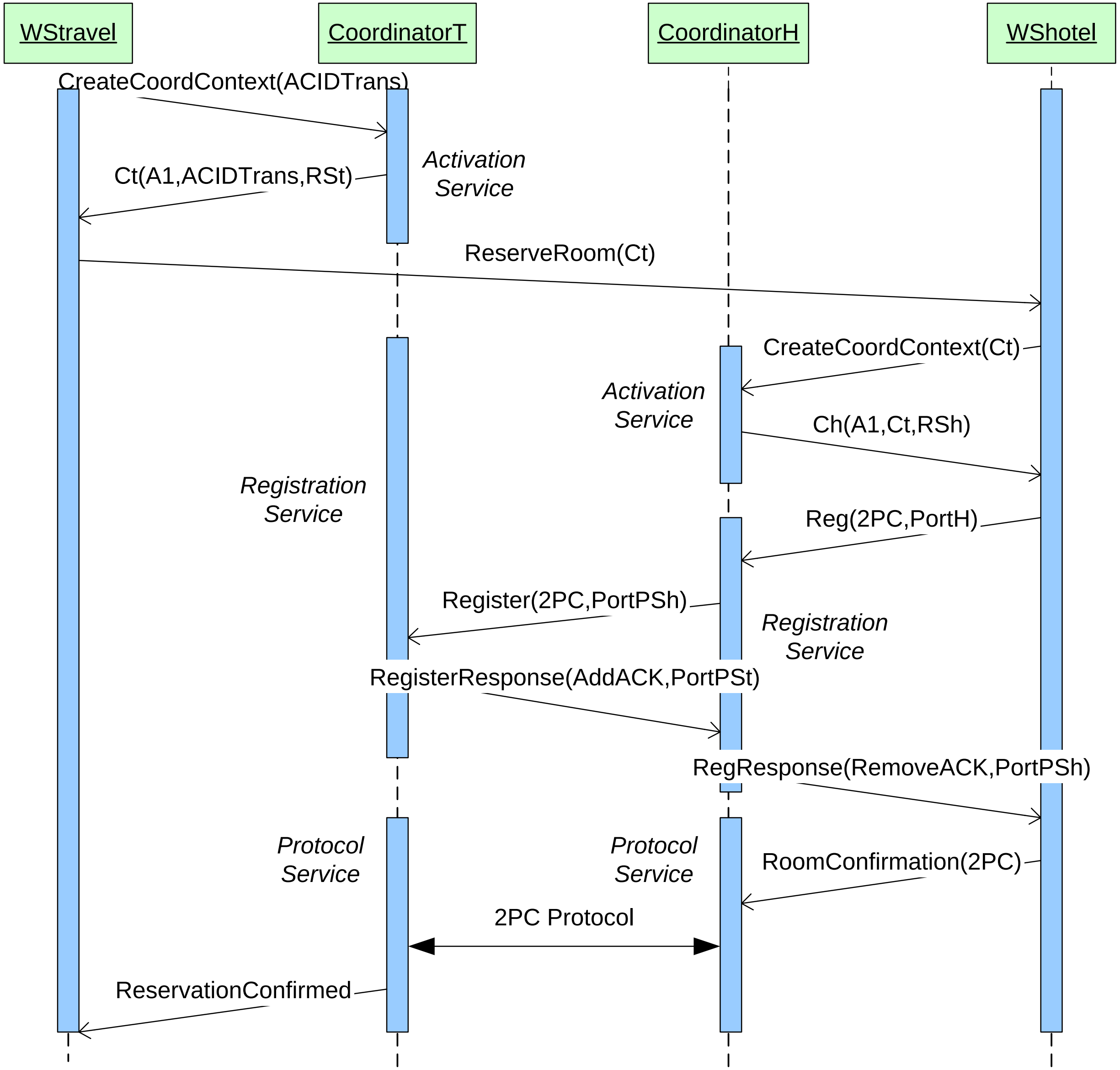
Distributed coordination



- Cb acted as a proxy to Ca: All the messages between Ca and B passed through Cb

1. A initiates a coordination instance by asking Ca to create a new coordination context
2. Ca returns a coordination context X1 to A. X1 contains a reference to Ca's RCPT
3. A registers itself as a participant of protocol X with Ca by passing a reference to its XPPT
4. Ca returns a reference to its XCPT to A
5. A sends a message to B within the scope of the coordination. To indicate the scope, it includes X1 as part of the SOAP header
6. B notices X1 in the SOAP message it receives. It asks Cb to create a new coordination context, which should be a sub-context of X1
7. Cb returns a coordination context X2 to B. X2 contains a reference to Cb's RCPT
8. B registers with Cb as a participant of X. B also passes X2 and its own XPPT to Cb as part of the registration
9. Cb registers as a participant of X with Ca. By doing so, Cb informs Ca about who its counterpart is with respect to X1, and can receive and forward all messages from Ca to B
10. Ca returns to Cb with a reference to its XCPT. Cb uses it to forward protocol messsages from B to Ca
11. Cb returns to B with a reference to its XCPT

Example


WStravel
CoordinatorT
CoordinatorH
WShotel
CreateCoordContext(ACIDTrans)
Activation Service
Ct(A1,ACIDTrans,RSt)
ReserveRoom(Ct)
CreateCoordContext(Ct)
Activation Service
Ch(A1,Ct,RSh)
Registration Service
Reg(2PC,PortH)
Register(2PC,PortPSh)
Registration Service
RegisterResponse(AddACK,PortPSt)
RegResponse(RemoveACK,PortPSh)
Protocol Service
Protocol Service
RoomConfirmation(2PC)
2PC Protocol
ReservationConfirmed

WS-Transaction

- A set of specifications built **on top of the WS-Coordination**
- Initially proposed in Aug. 2002 by IBM, MS, and BEA
- Specifies a standard protocol for **long-running transactions**, called **business activities**
- Also provides a set of specifications for **short-duration transactions**, called **atomic transactions**
- WS-AtomicTransaction and WS-BusinessActivity leverage WS-Coordination by defining two particular coordination types: a short-term atomic transaction and a long-duration business activity
- For a long-running transaction, WS-Transaction uses a compensation scheme where participants provide an undo operation that is used if the transaction does not complete

Transactions in Web services

- Transactions implemented through Web services are often long-running
  - Since Web services implement business applications, completing a transaction may involve manual intervention or executing lengthy business processes
  - e.g., Validating an order by checking that goods are in stock
- It may not always possible to preserve ACID properties using 2PC (too long lock duration)
- The lack of a fixed resource and operation model
  - WSDL can represent anything from database insertions to sending a letter to a customer
  - e.g., Rolling back the delivery of a letter to a customer

Transactions in web services

- To relax the rigidity of ACID properties and to leverage compensation mechanisms
  - Each of the participating Web services can update its persistent storage after each step of the transaction (i.e., atomicity and isolation constraints are relaxed)
  - If, for some reason, the transaction must be abort, then the Web services execute a compensation operation that semantically undoes the effects of the (partial) transaction execution
  - Each operation may have a different compensation logic
  - Providers may or may not impose a time limit for cancellation or impose cancellation fees

Relationship with WS-Coordination

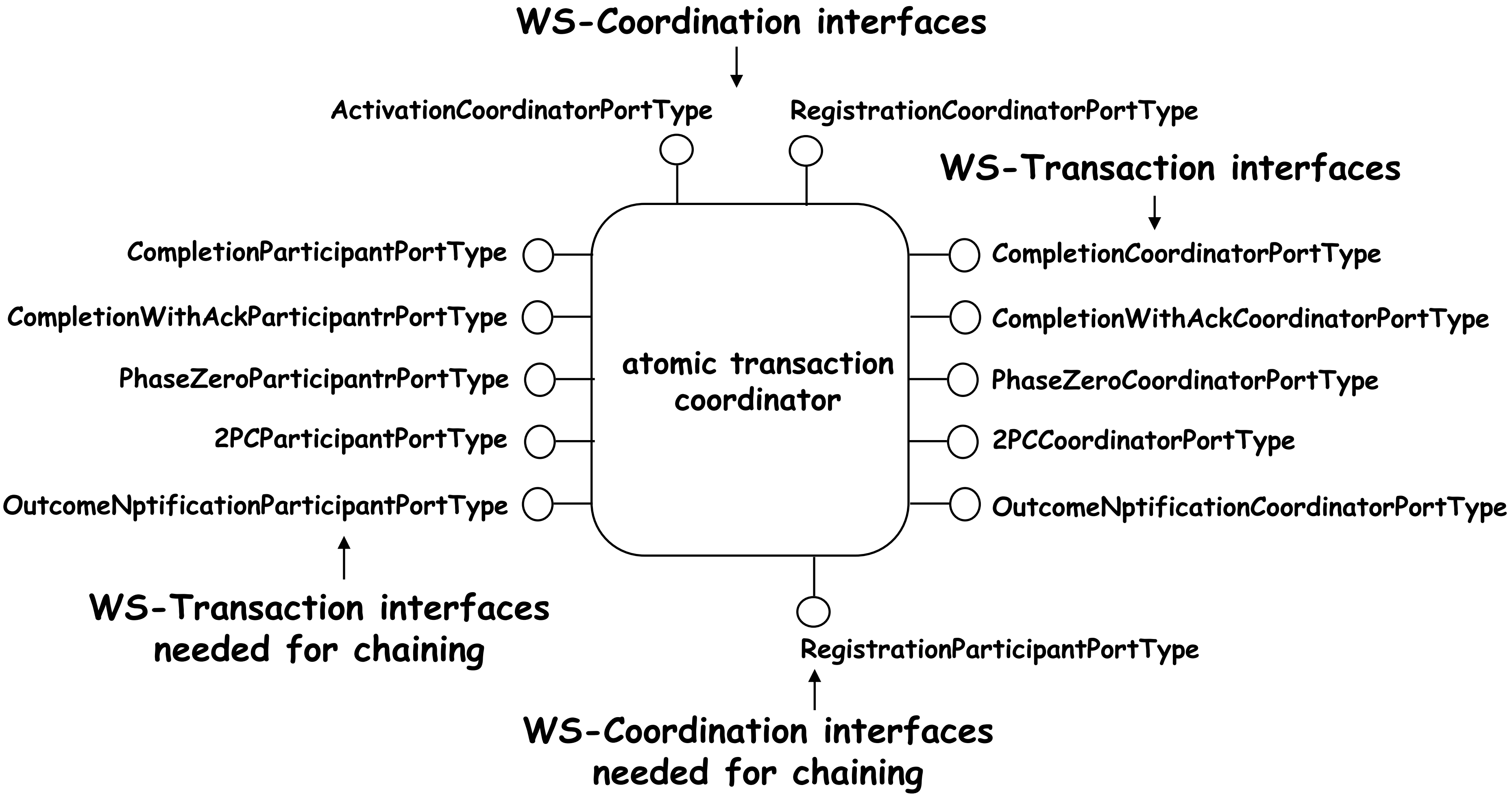
- WS-Transaction assumes the existence of a set of Web services that participate in a transaction and of one or more coordinators that coordinate the transaction
- WS-Coordination protocols are used by the following parties
  - By the initiating Web service, to create a new coordination context
  - By a participating Web service, to pass the coordination context to another Web service
  - By a participating Web service, to register for a transaction protocol with a central transaction coordinator or with its own coordinator
  - By a proxy coordinator, to chain with a primary coordinator

Atomic transactions

- Five protocols make up the coordination type:
  - Completion: To inform the coordinator that it should start a 2PC protocol to verify the outcome of the transaction and ask the participants to either commit or abort
  - 2PC: The standard two-phase commit protocol with a prepare phase and a commit or abort phase
  - PhaseZero: To let all participants know that a 2PC protocol is about to commence
  - OutcomeNotification: A participant can query the coordinator about the outcome of the transaction at any point during or after the execution of a 2PC protocol
  - CompletionWithAck: The coordinator has to remember the outcome of a transaction without discarding it, until the requesting Web service sends an acknowledgement that it has received the outcome
- Defines a structure for the transaction coordination context
- The value of the coordination type entry in the transaction context: http://schemas.xmlsoap.org/ws/2002/08/wstx

Port types in atomic transactions

- The reason for having coordinators implement the participant port types is that in this way coordinator chaining can be accomplished

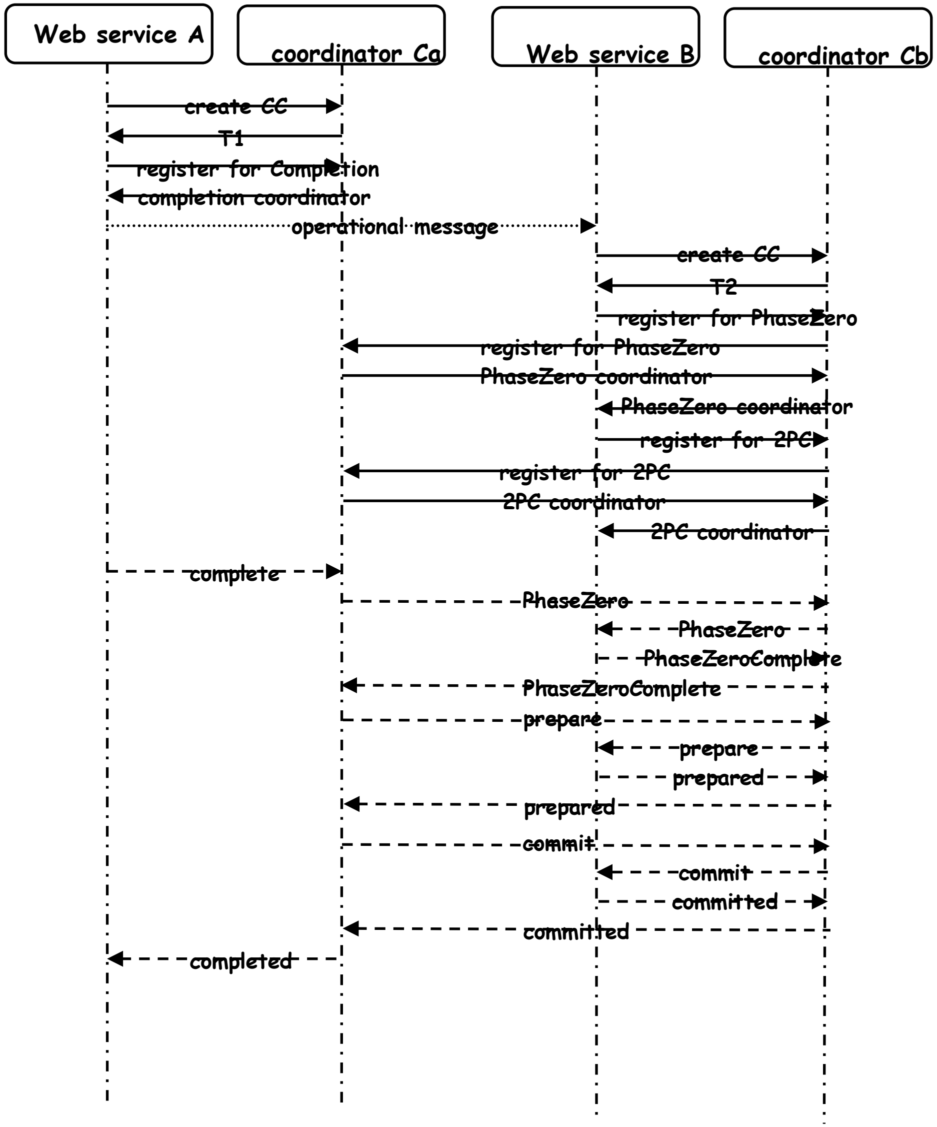


WS-Coordination interfaces
ActivationCoordinatorPortType
RegistrationCoordinatorPortType
WS-Transaction interfaces
CompletionParticipantPortType
CompletionWithAckParticipantrPortType
PhaseZeroParticipantrPortType
2PCParticipantPortType
OutcomeNptificationParticipantPortType
atomic transaction coordinator
CompletionCoordinatorPortType
CompletionWithAckCoordinatorPortType
PhaseZeroCoordinatorPortType
2PCCoordinatorPortType
OutcomeNptificationCoordinatorPortType
WS-Transaction interfaces needed for chaining
RegistrationParticipantPortType
WS-Coordination interfaces needed for chaining

Port types

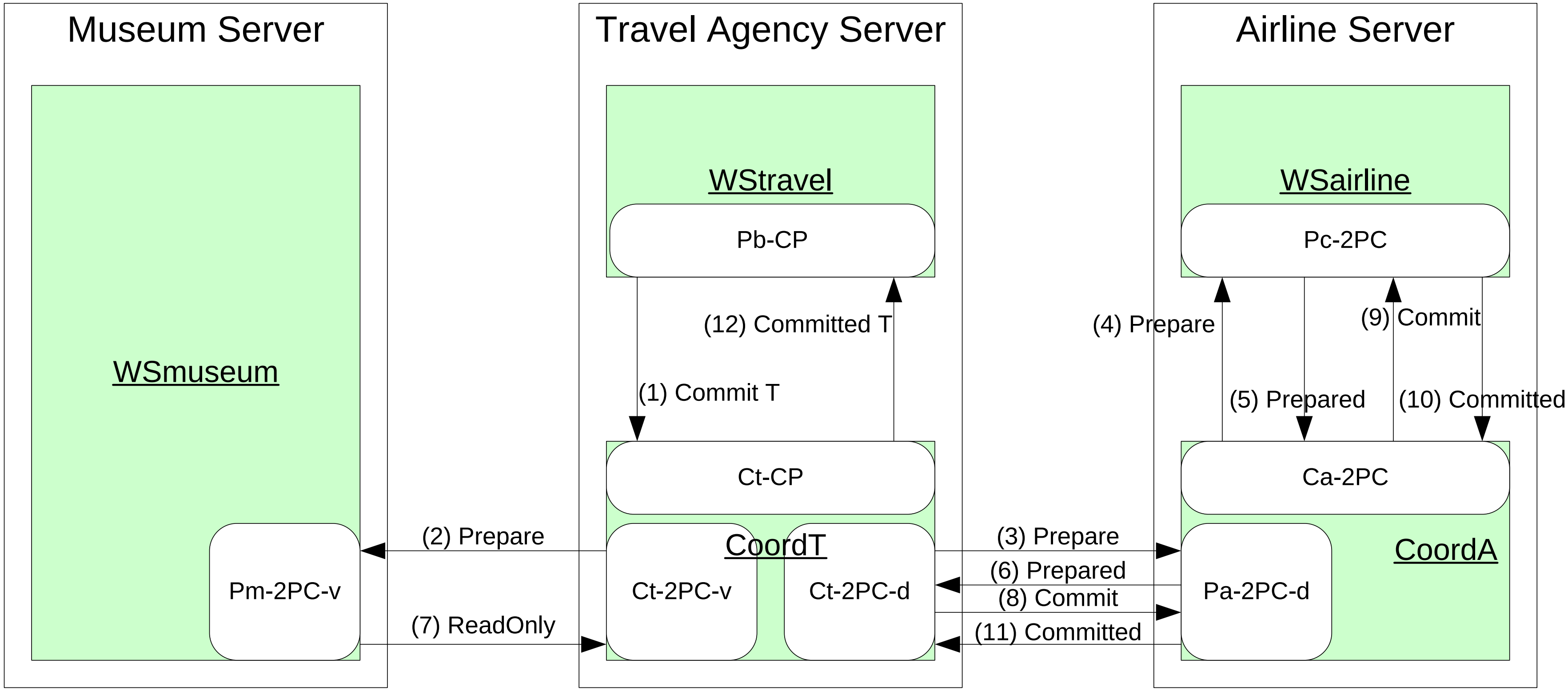
- A web service that initiates a transaction: CompletionParticipantPortType or CompletionWithAckParticipantPortType
- A web service executing database state changes that must be either committed or rolled back at the end of a transaction: 2PCParticipantPortType
- A web service that executes other forms of state changes (e.g., in-memory changes) and wishes to be notified before commencement of a 2PC: PhaseZeroParticipantPortType
- A Web service that would like to check the outcome of a transaction: OutcomeNotificationParticipantPortType

Example scenario for atomic transaction



Example

The travel agency and the airline can perform 2PC, but the museum can perform only a simple, zero-phase update

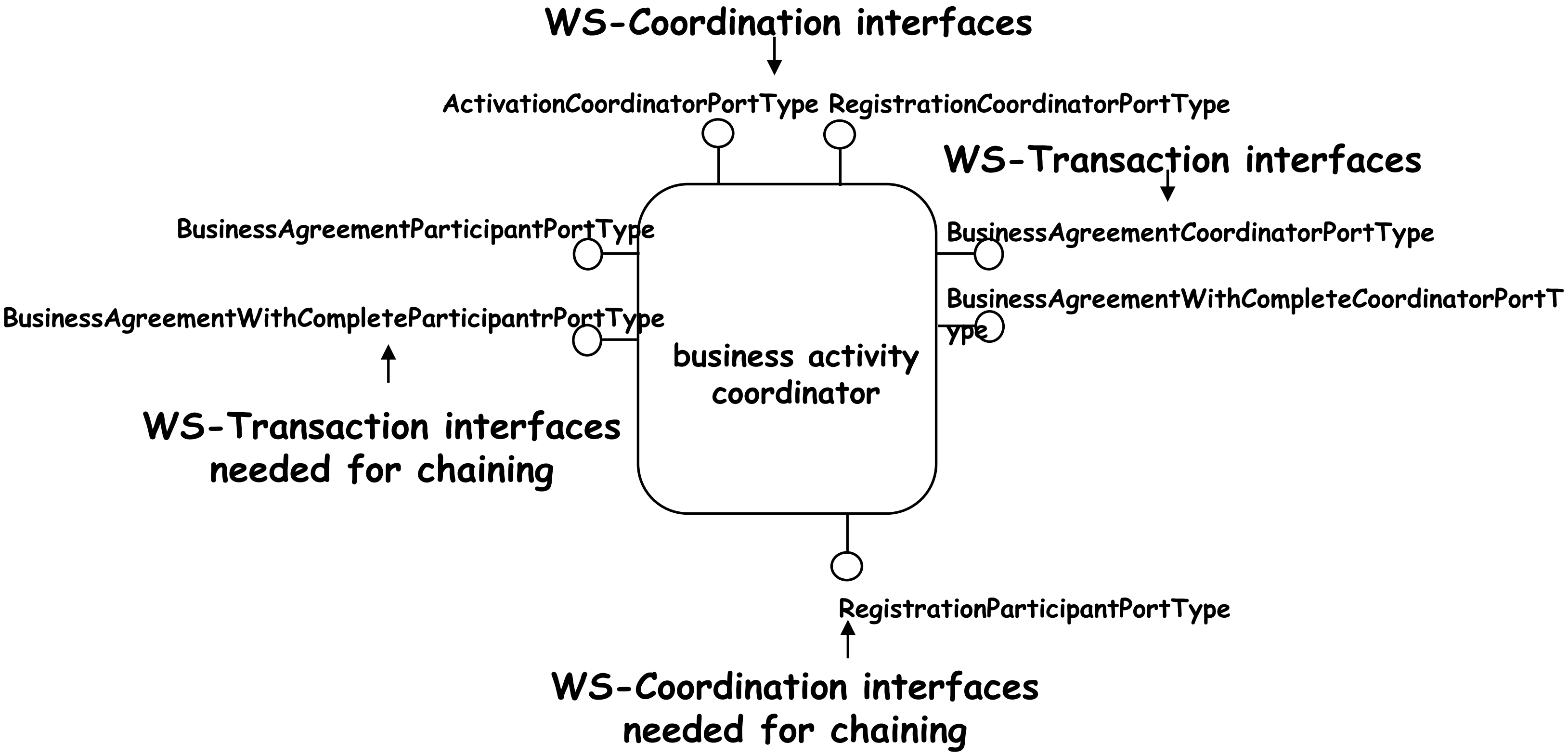


Business activities

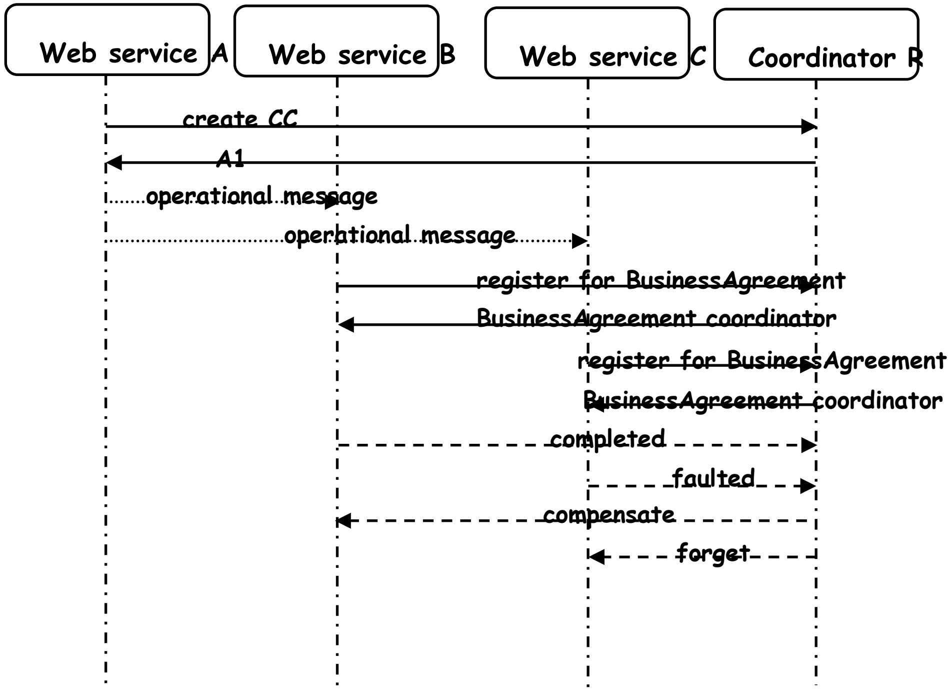
- A coordination type that does not require locking resources
- Two protocols make up the coordination type: BusinessAgreement and BusinessAgreementWithComplete
- The BusinessAgreement protocol is initiated by a participating web service to inform the coordinator about the status of its execution (Exited, Completed, or Faulted)
- After reaching a consensus on whether to go forward with the transaction or to abort it, the coordinator responds with a Close, Complete, Compensate, or Forget message to all the participants
- The BusinessAgreementWithComplete protocol is similar to the BusinessAgreement protocol with the addition that the coordinator has to tell a participant when all the tasks expected from the latter as part of the transaction have been requested

Port types

- BusinessAgreeementParticipantPortType and BusinessAgreementCoordinatorPortType
- BusinessAgreementWithCompleteParticipantPortType and BusinessAgreementWithCompleteCoordinatorPortType
- The coordinator should also implement the participant port types so that coordinator chaining can be accomplished
- The value of the coordination entry in the context for business activities should be set to http://schemas.xmlsoap.org/ws/2002/08/wsba


WS-Coordination interfaces
ActivationCoordinatorPortType
RegistrationCoordinatorPortType
WS-Transaction interfaces
BusinessAgreementParticipantPortType
BusinessAgreementCoordinatorPortType
BusinessAgreementWithCompleteCoordinatorPortType
BusinessAgreementWithCompleteParticipantrPortType
business activity coordinator
WS-Transaction interfaces needed for chaining
RegistrationParticipantPortType
WS-Coordination interfaces needed for chaining

An execution scenario of Business activity protocol

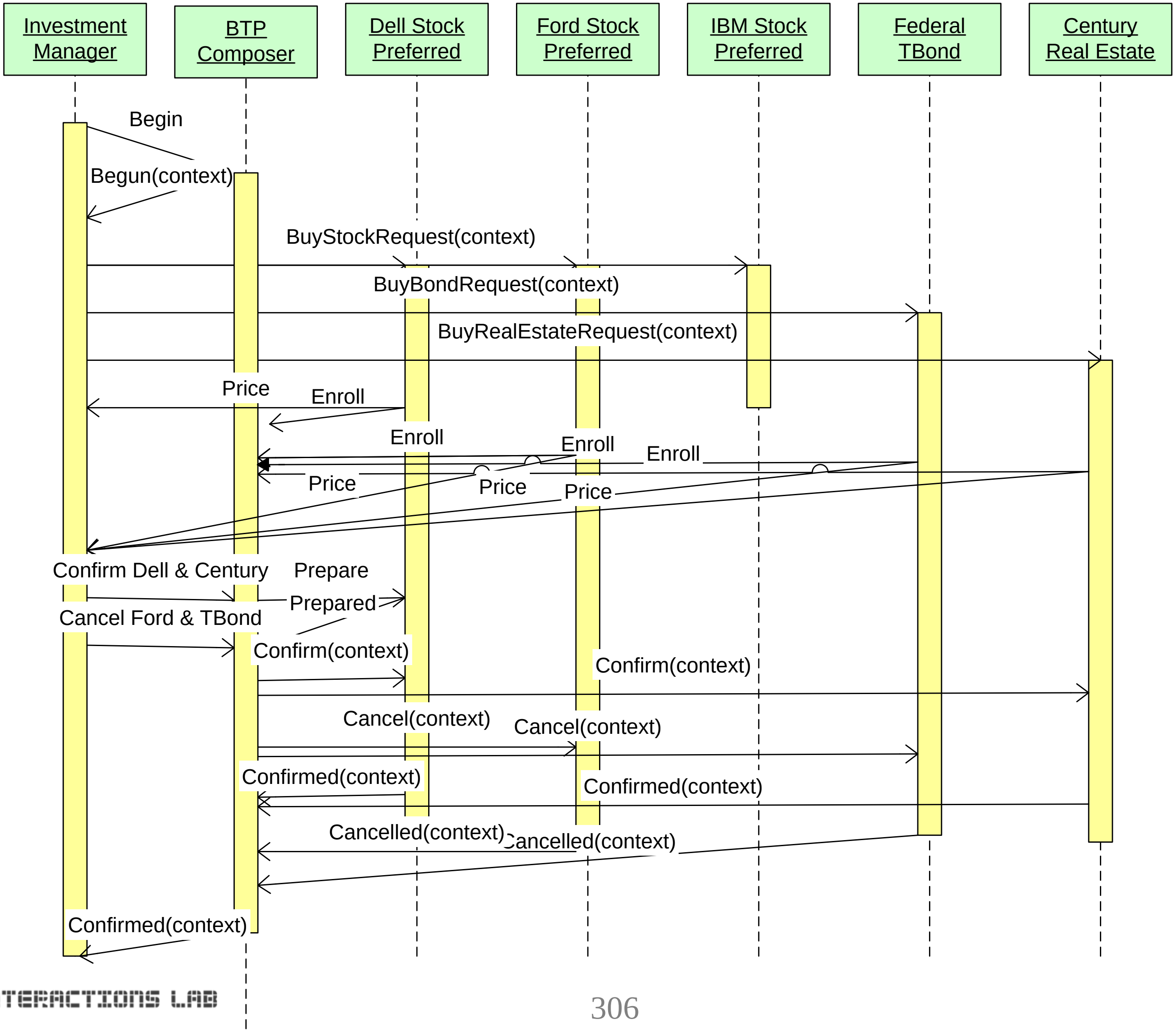


1. A initiates a business activity conversation and passes the context to B and C. B and C register for the BusinessAgreement protocol with R
2. B successfully finishes its tasks, but C encounters a failure. So, it sends a Faulted message to R
3. By the time R receives the Faulted message from C, it may or may not have received the Completed message from B. If it has received the Completed message from B, R sends a Compensate message to B. If not, it sends a Cancel message to B
4. R also sends a Forget message to C, denoting that the protocol is terminated and no further actions are required form C

BTP: Business Transaction Protocol

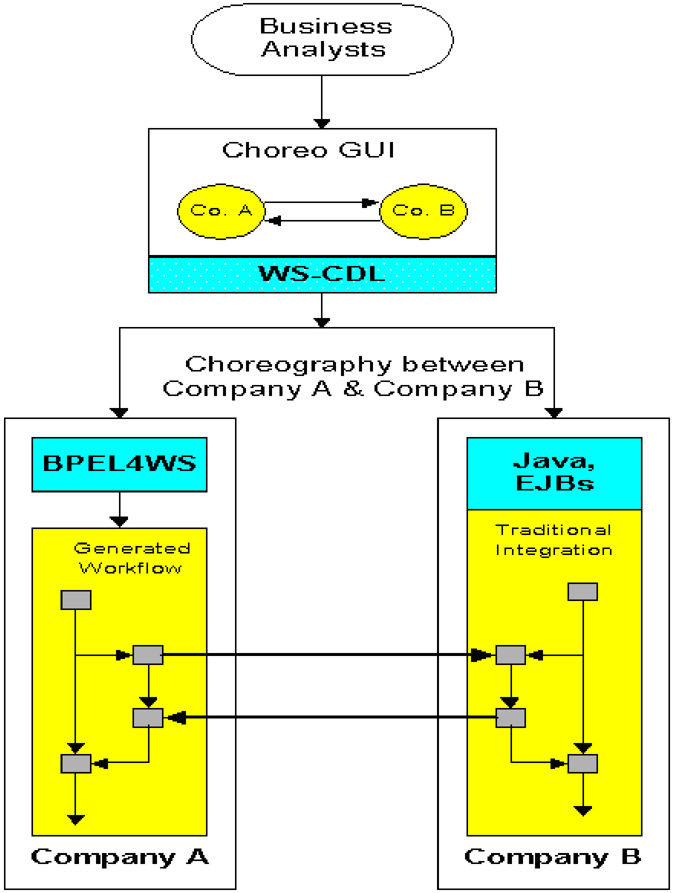
- Developed by OASIS to automate and manage long-running web-based collaborative business applications
- To support interactions that cross application and administrative boundaries, thus requiring extended transactional support beyond the classical ACID properties
- relaxes the ACID properties via two subprotocols:
  - atoms, where isolation is relaxed
  - cohesions, where both isolation and atomicity are relaxed
- More suitable for loosely coupled applications

BTP example for cohesion



Web Services Choreography Working Group

- 2003년 1월 활동 시작으로 2004년 12월까지 활동을 완료할 계획
- Web Services의 상호간의 메시지를 기술하는 WS-CDL 1.0 표준을 4월에 제안하고 10월에 수정
- OASIS의 WS-BPEL 등 타 choreography 기술 언어와 표준 선점에 경쟁 중임
- To specify **a declarative, and WSDL 1.2 based language** that describes **cross enterprise collaborations of Web Services participants** by defining their **common observable behavior**, where synchronized information exchanges through their shared contact points occur, when the commonly defined ordering rules are satisfied.

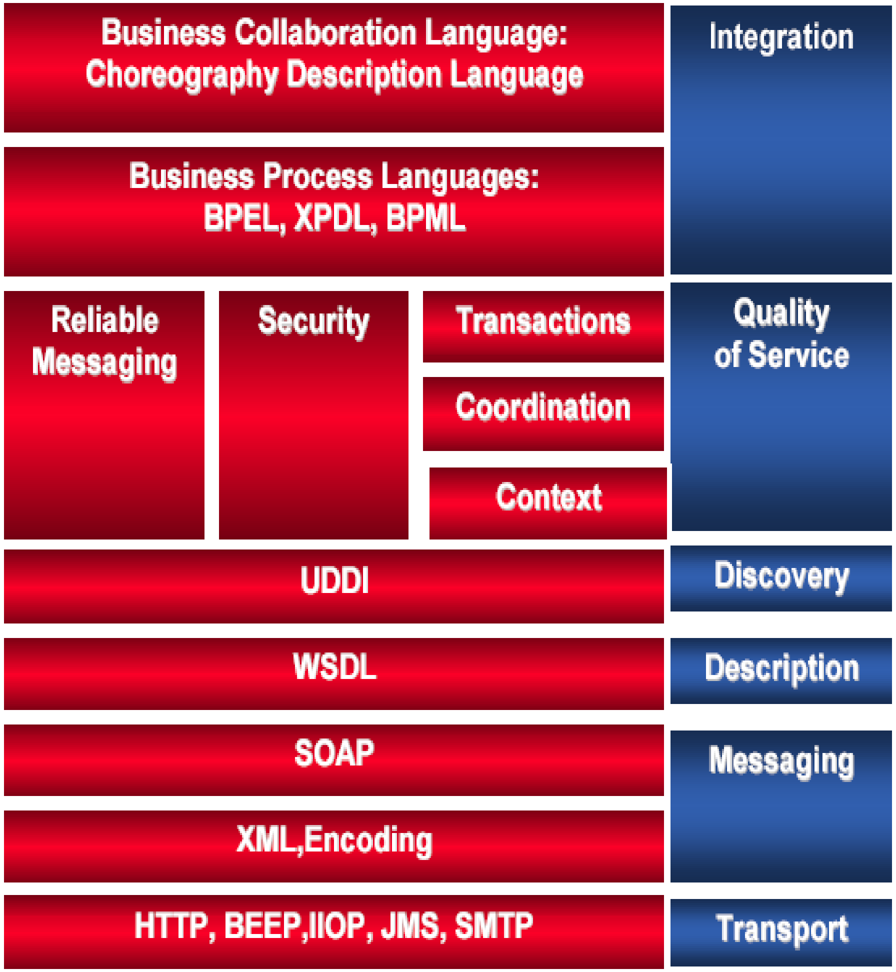


Documents by WS Choreography WG

- Working Drafts
  - Web Services Choreography Requirements 1.0 (Working Draft)
  - WS Choreography Model Overview (Working Draft)
  - Web Services Choreography Description Language Version 1.0

WS-CDL

- An XML-based language that describes peer-to-peer collaborations of Web Services participants by defining from a global viewpoint their common and complementary observable behavior where ordered message exchanges result in accomplishing a common business goal
- Supports SOAP Version 1.2, WSDL 2.0, and the Web's architectural layers
- The first working draft was released 27 April 2004
- New working draft of WS-CDL version 1.0 in October, 2004



Travel agent example

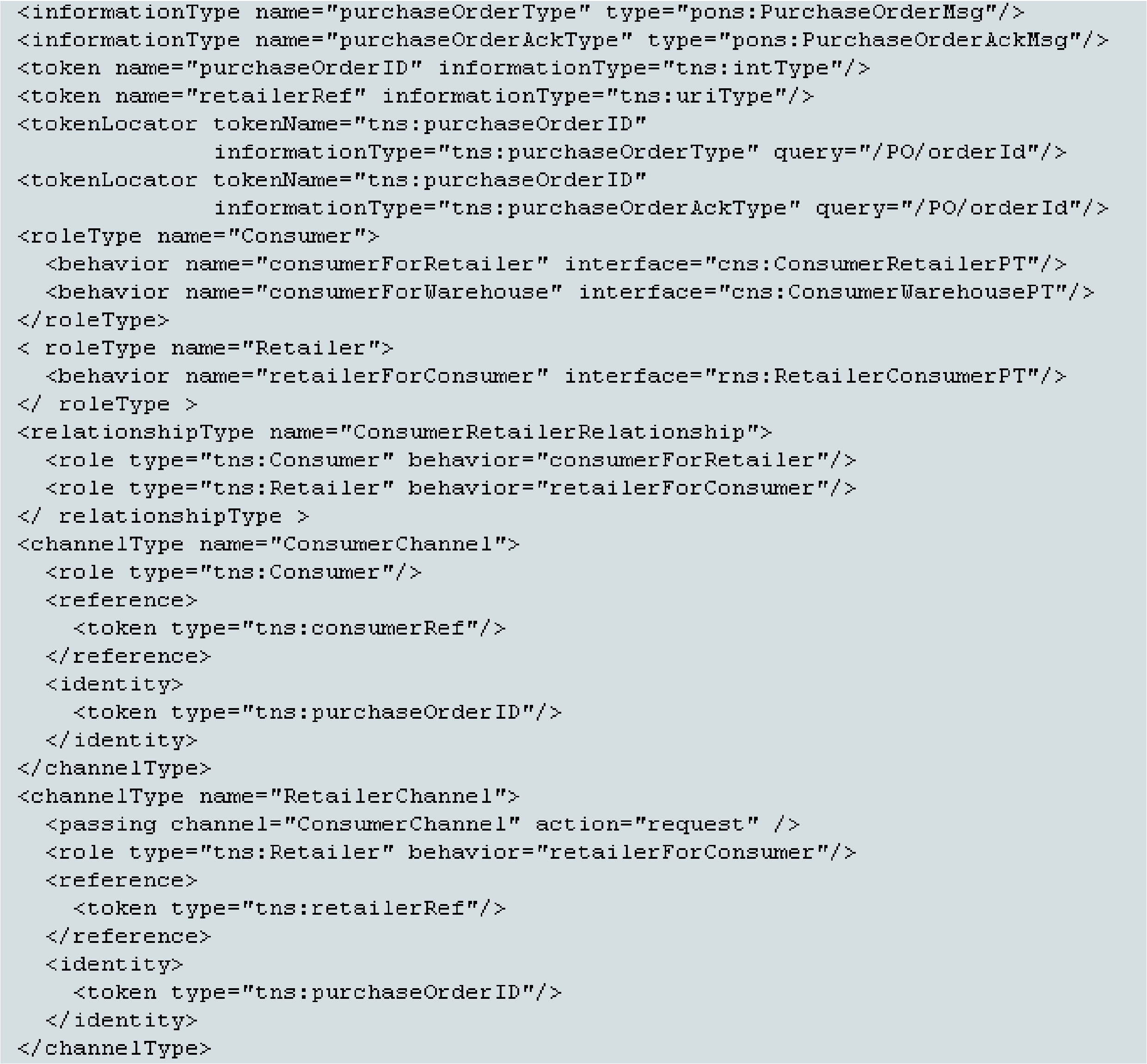
1. The client interacts with the travel agent to request information about various services.
2. Prices and availability matching the client requests are returned to the client. The client can then perform one of the following actions:
    1. The client can refine their request for information, possibly selecting more services from the provider (Repeat step 2). OR
    2. The client may reserve services based on the response,
       OR
    3. The client may quit the interaction with the travel agent.
3. When a customer makes a reservation, the travel agent then checks the availability of the requested services with each service provider.

Travel agent example (cont.)

4. Either
    1. All services are available, in which case they are reserved. OR
    2. For those services that are not available, the client is informed.
        1. Either
            1. Given alternative options for those services.
               OR
            2. Client is advised to restart the search by going back to step 1.
        2. Go back to step 3.
5. For every relevant reserved service the travel agent takes a deposit for the reservation. A credit card can be used as a form of deposit.
6. The client is then issued a reservation number to confirm the transaction.

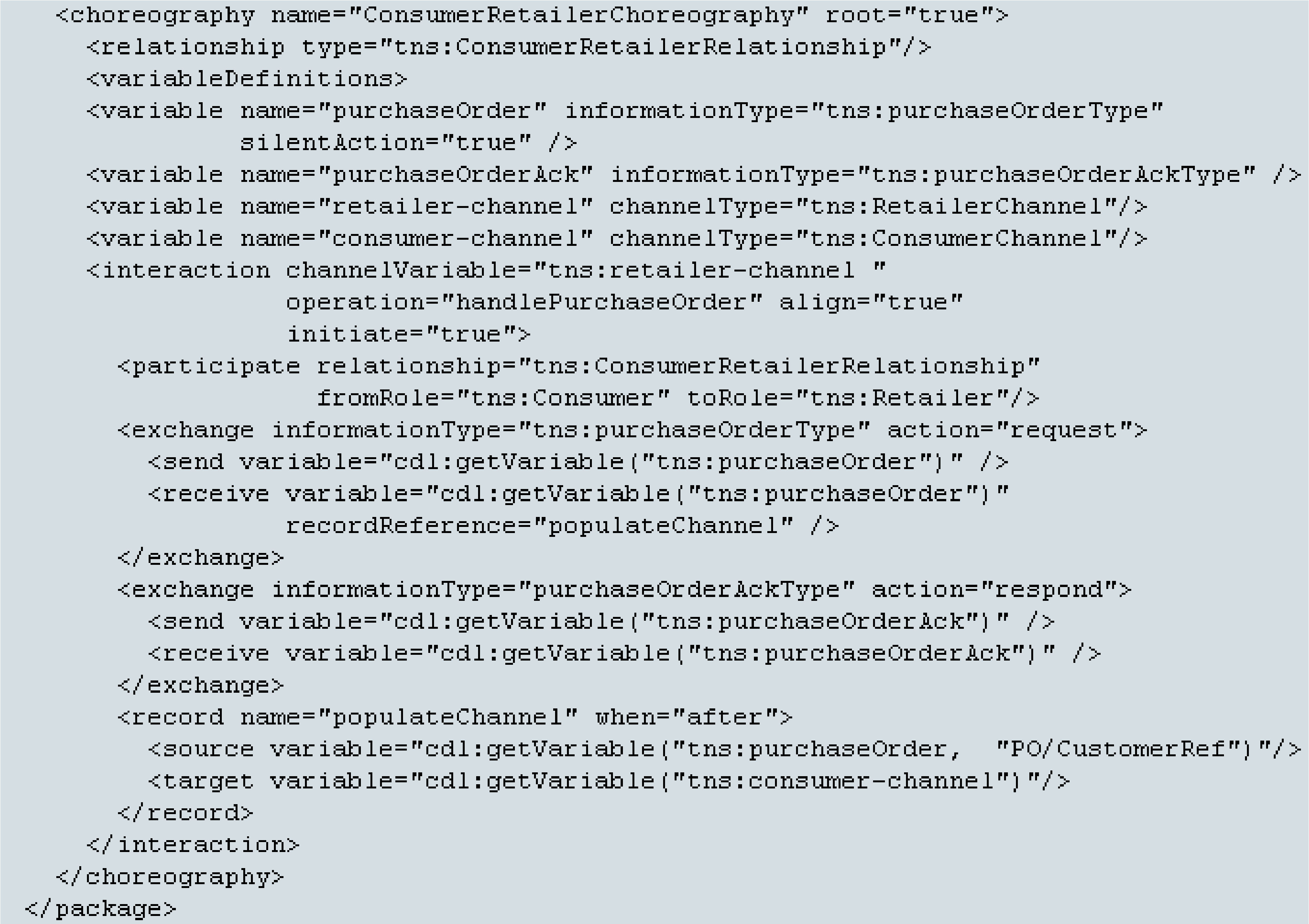
Example

The example right shows a Choreography that involves one Interactions. The Interaction happens from Role Type “Customer” to Role Type “Retailer” on the Channel “retailer-channel” as a request/response message exchange.



```
<informationType name="purchaseOrderType" type="pons:PurchaseOrderMsg"/>
<informationType name="purchaseOrderAckType" type="pons:PurchaseOrderAckMsg"/>
<token name="purchaseOrderID" informationType="tns:intType"/>
<token name="retailerRef" informationType="tns:uriType"/>
<tokenLocator tokenName="tns:purchaseOrderID"
              informationType="tns:purchaseOrderType" query="/PO/orderId"/>
<tokenLocator tokenName="tns:purchaseOrderID"
              informationType="tns:purchaseOrderAckType" query="/PO/orderId"/>
<roleType name="Consumer">
  <behavior name="consumerForRetailer" interface="cns:ConsumerRetailerPT"/>
  <behavior name="consumerForWarehouse" interface="cns:ConsumerWarehousePT"/>
</roleType>
< roleType name="Retailer">
  <behavior name="retailerForConsumer" interface="rns:RetailerConsumerPT"/>
</ roleType >
<relationshipType name="ConsumerRetailerRelationship">
  <role type="tns:Consumer" behavior="consumerForRetailer"/>
  <role type="tns:Retailer" behavior="retailerForConsumer"/>
</ relationshipType >
<channelType name="ConsumerChannel">
  <role type="tns:Consumer"/>
  <reference>
    <token type="tns:consumerRef"/>
  </reference>
  <identity>
    <token type="tns:purchaseOrderID"/>
  </identity>
</channelType>
<channelType name="RetailerChannel">
  <passing channel="ConsumerChannel" action="request" />
  <role type="tns:Retailer" behavior="retailerForConsumer"/>
  <reference>
    <token type="tns:retailerRef"/>
  </reference>
  <identity>
    <token type="tns:purchaseOrderID"/>
  </identity>
</channelType>
```

Example (cont.)



```
    <choreography name="ConsumerRetailerChoreography" root="true">
      <relationship type="tns:ConsumerRetailerRelationship"/>
      <variableDefinitions>
      <variable name="purchaseOrder" informationType="tns:purchaseOrderType"
                silentAction="true" />
      <variable name="purchaseOrderAck" informationType="tns:purchaseOrderAckType" />
      <variable name="retailer-channel" channelType="tns:RetailerChannel"/>
      <variable name="consumer-channel" channelType="tns:ConsumerChannel"/>
      <interaction channelVariable="tns:retailer-channel "
                   operation="handlePurchaseOrder" align="true"
                   initiate="true">
        <participate relationship="tns:ConsumerRetailerRelationship"
                     fromRole="tns:Consumer" toRole="tns:Retailer"/>
        <exchange informationType="tns:purchaseOrderType" action="request">
          <send variable="cdl:getVariable("tns:purchaseOrder")" />
          <receive variable="cdl:getVariable("tns:purchaseOrder")"
                   recordReference="populateChannel" />
        </exchange>
        <exchange informationType="purchaseOrderAckType" action="respond">
          <send variable="cdl:getVariable("tns:purchaseOrderAck")" />
          <receive variable="cdl:getVariable("tns:purchaseOrderAck")" />
        </exchange>
        <record name="populateChannel" when="after">
          <source variable="cdl:getVariable("tns:purchaseOrder,  "PO/CustomerRef")"/>
          <target variable="cdl:getVariable("tns:consumer-channel")"/>
        </record>
      </interaction>
    </choreography>
  </package>
```