

Portals, Portlets and Web Services for Remote Portlets

Web Services for the Integrated Enterprise

OMG's Second Workshop on Web Services Modeling, Architectures, Infrastructures and Standards

10. Feb 2003

Dr. Carsten Leue

Overview

- Portal Architecture
 - Portal
 - Portlet
- Standards
 - Portlet API
 - WSRP
- Web Services for Remote Portlets

What is a Portal?

- Provides unified access to all your internal applications and content
- Delivers a collaborative working environment
- Unified workplace – the new standard user interface
- Gives you a personalized user interface
- Access through multiple devices – convenience and personalization
- All with very little coding



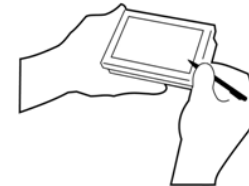
Content

Applications

Collaboration

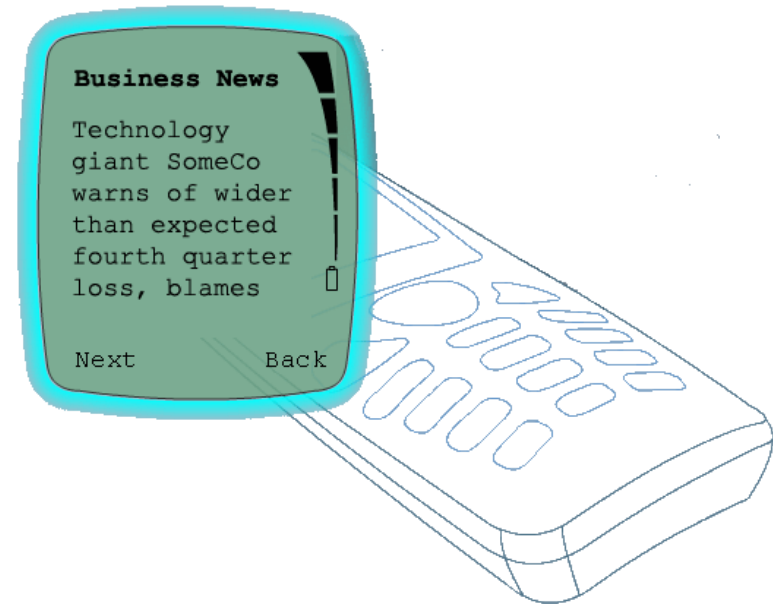
Presentation

Personalization

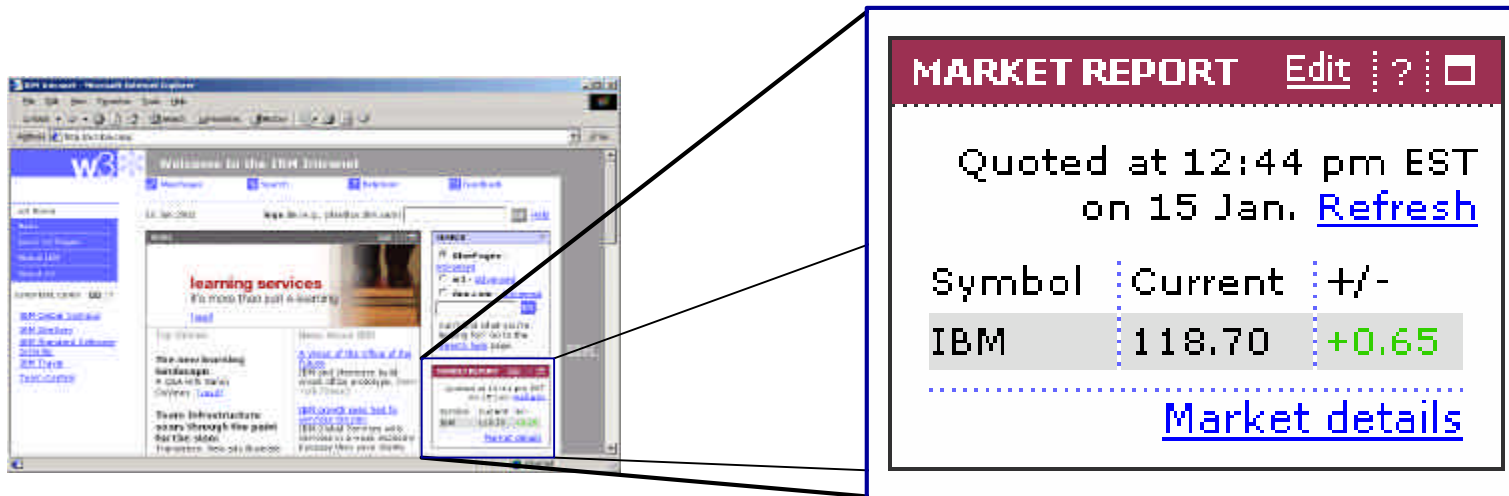


Mobile Access

- Access from anywhere, anytime
- Desktop and mobile browsers
 - Internet Explorer
 - Netscape Navigator
 - WAP/WML phones
 - iMode/cHTML phones
- Services, content and user interface are adapted to the device and user's context
- More devices in the future
 - personal digital assistants
 - Voice



Example of a Stock Quote Portlet



The screenshot shows a web browser window with a portlet titled "MARKET REPORT". The portlet displays the following information:

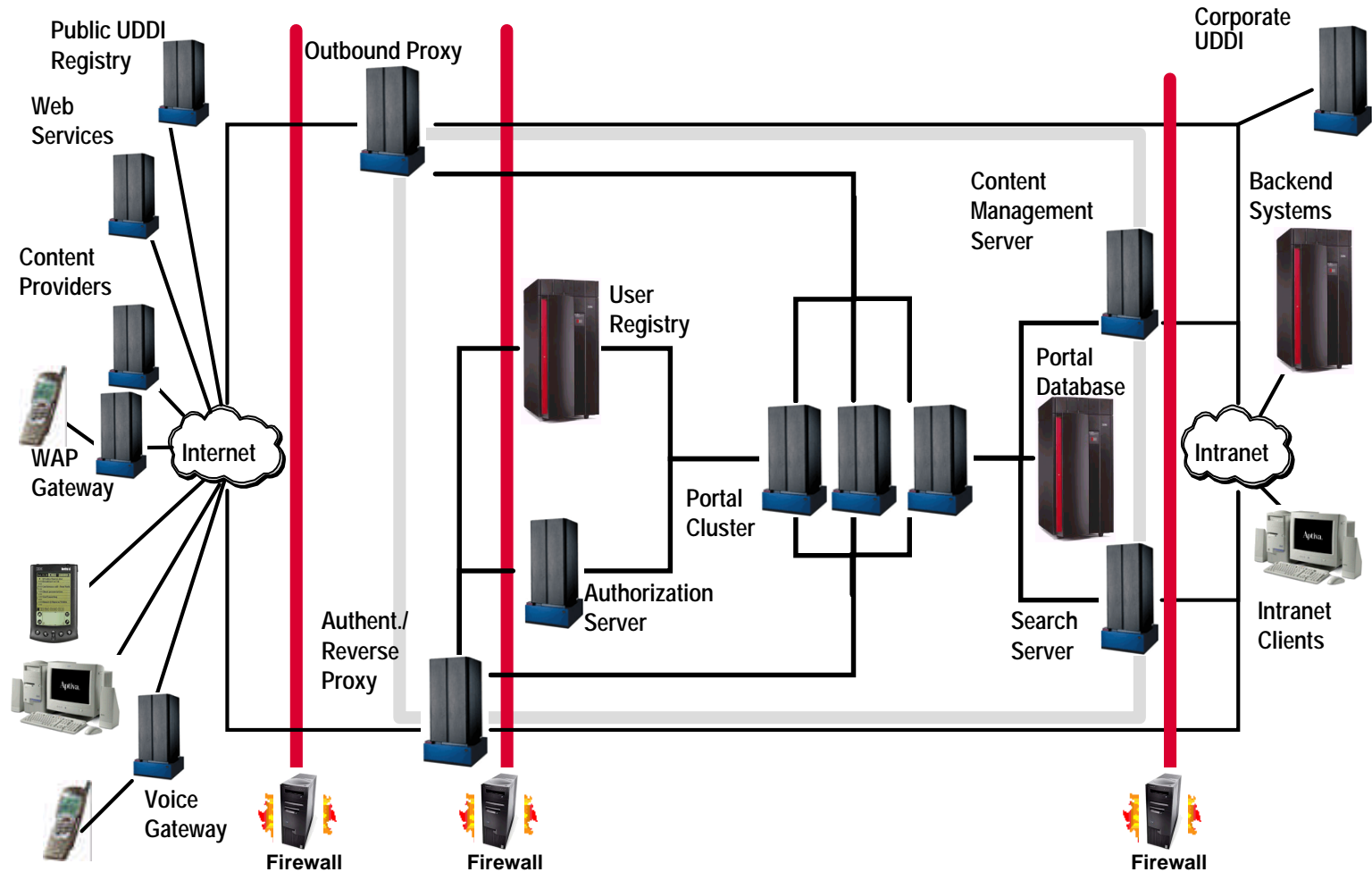
Quoted at 12:44 pm EST
on 15 Jan. [Refresh](#)

Symbol	Current	+/-
IBM	118.70	+0.65

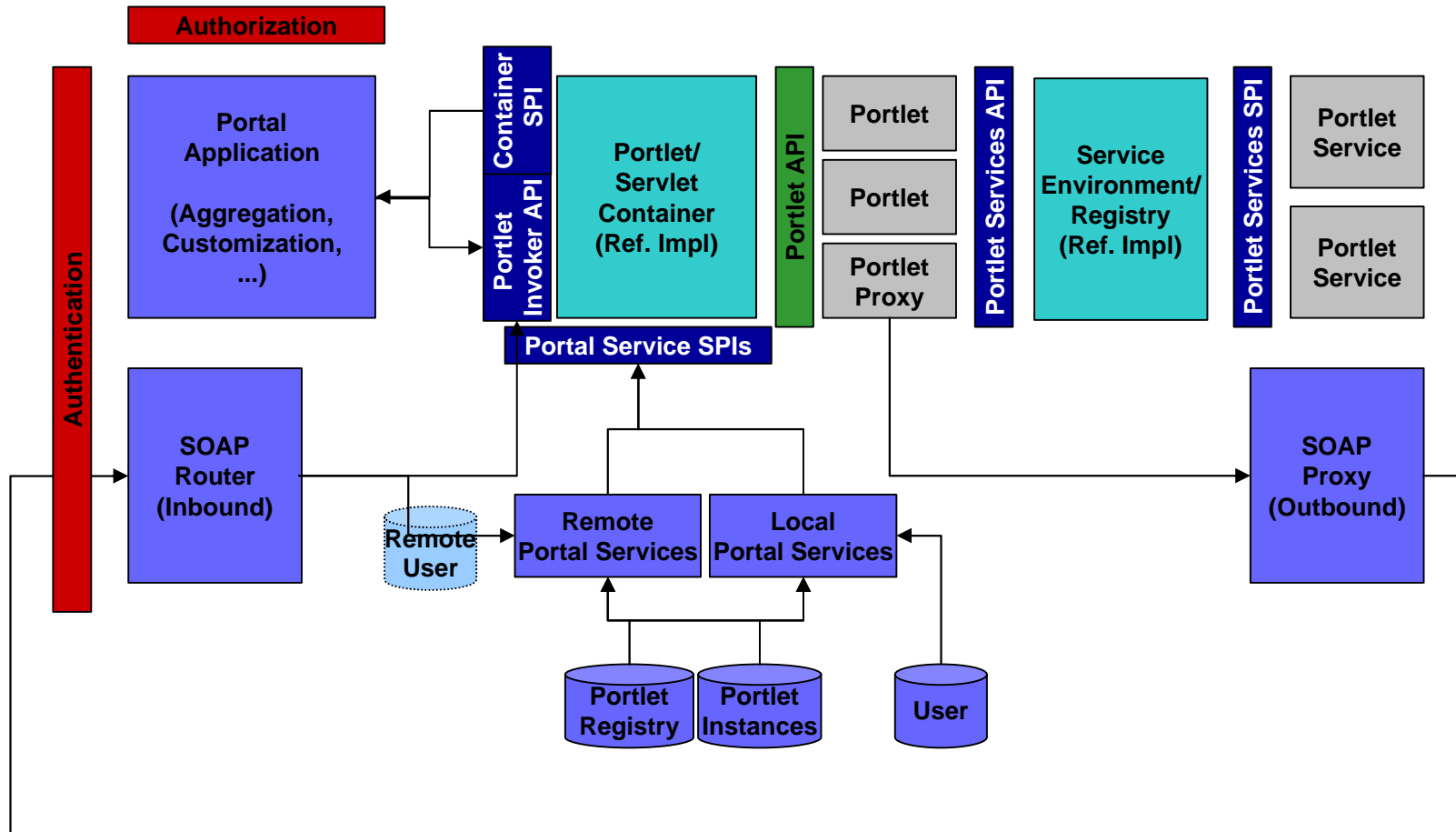
[Market details](#)

- Stock prices for user-selected list of stock symbols:
 - VIEW mode shows stock prices: **doView**
 - EDIT mode lets user change stocks
 - HELP mode explains the portlet
 - CONFIG mode lets administrator select stock quote source to use

Example of a Portal Topology



Portal Architecture



 will be standardized in the JSR 168

What to remember?

- Portals are highly modular and distributed web applications
- Portals provide multi-modal, personalized access to applications
- Applications are represented as **portlets** and are aggregated on a portal page

Current Situation

- Different Portal Vendors have defined different APIs
 - Different APIs for local „components“ (simple JSPs, Portlets, Modules, ...)
 - Different interfaces/protocols for invocation of „remote components“ (Gadgets, Remote Portlets, ...)
- No interoperability between local/remote portlets and portal servers
- Application and Content Providers must implement different portlets for different portal servers
- Customers developing significant numbers of portlets are quickly locked into a particular portal solution
- No standardized, easy way to plug-n-play content and applications into portals exists

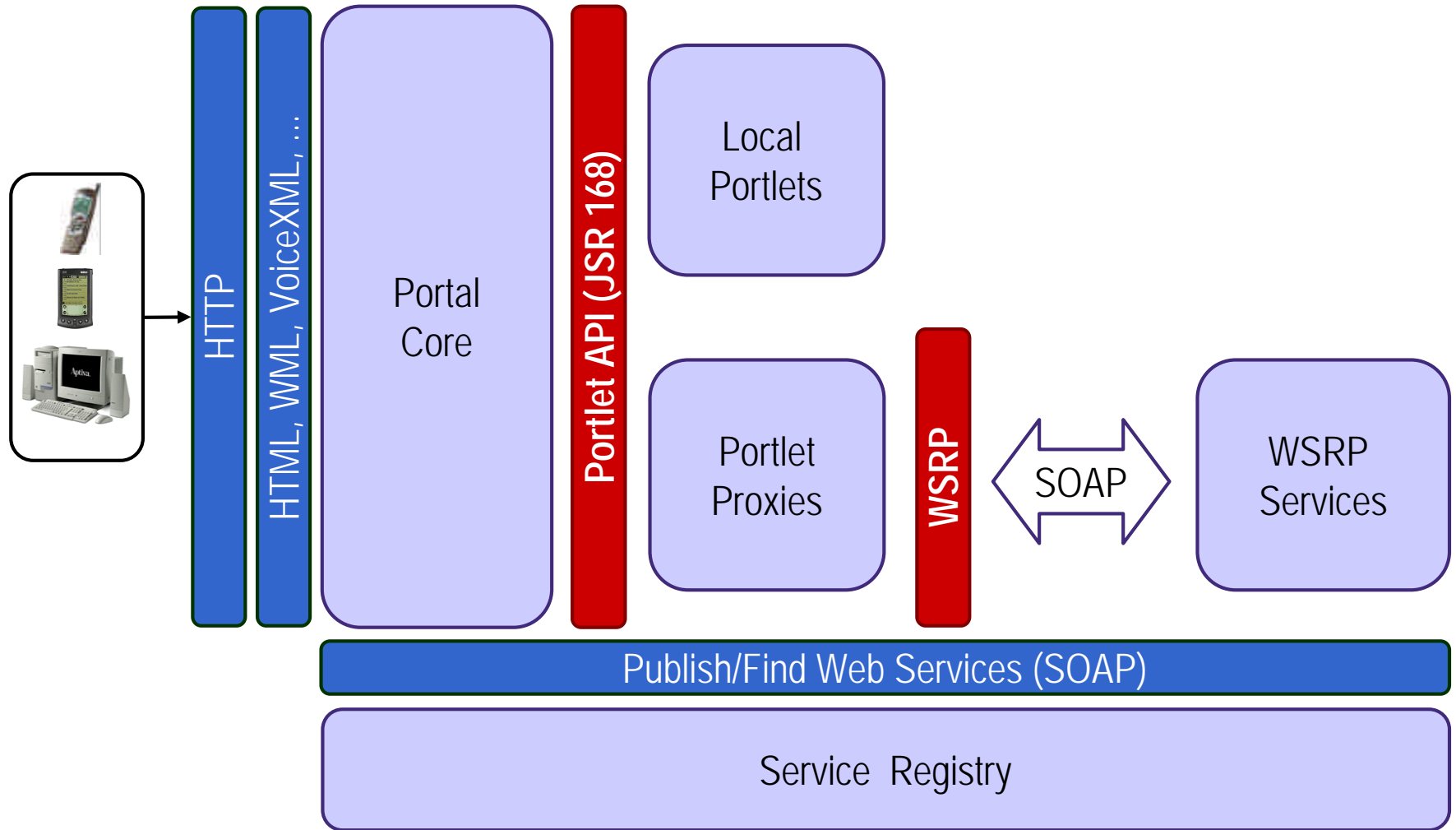
Portlet API

- API defining interaction between portals and portlets
- leverages the Java™ Servlet API
- items covered in V1.0
 - window states / portlet modes
 - action/render phases
 - user specific preference data
 - session concept
 - include to Servlets/JSPs
 - user profile information
 - caching
 - portal context
 - packaging & deployment
 - portlet JSP tag lib

Standards for Portals

- Java Portlet API
 - Achieves Interoperability between local Portlets and Portals
 - Standardization in JSR in the Java Community Process
 - JSR is co-lead by IBM and Sun
 - <http://jcp.org/jsr/detail/168.jsp>
- Web Services for Remote Portlets (WSRP)
 - Achieves Interoperability between Visual, User-Facing Web Services and Portals
 - Standardization in OASIS WSRP Workgroup
 - IBM leads workgroup
 - <http://oasis-open.org/committees/wsrp/>
- Portlet API and WSRP provide particular value when used together:
 - Portlets written to the Java Portlet API may be wrapped in WSRP services and published to directories
 - WSRP services can be found in a directory and bound to by wrapping them into Portlets written to the Java Portlet API to aggregate them in portals

Portlet API & WSRP

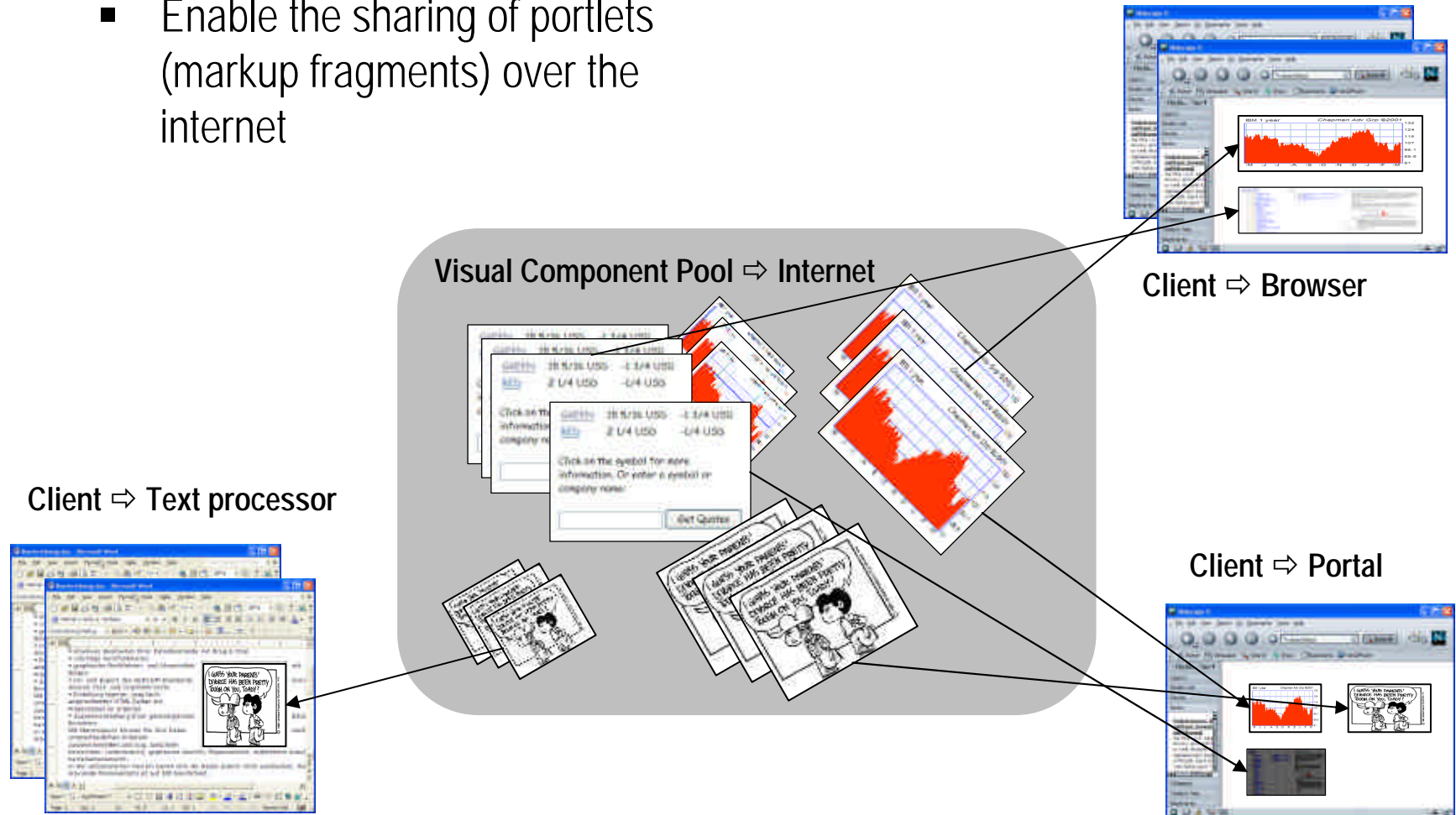


What to remember?

- JSR168 defines a Java API that allows portlets to run on any compliant portal
- WSRP allows Web Services to be integrated as portlets in a plug&play fashion

WSRP Motivation

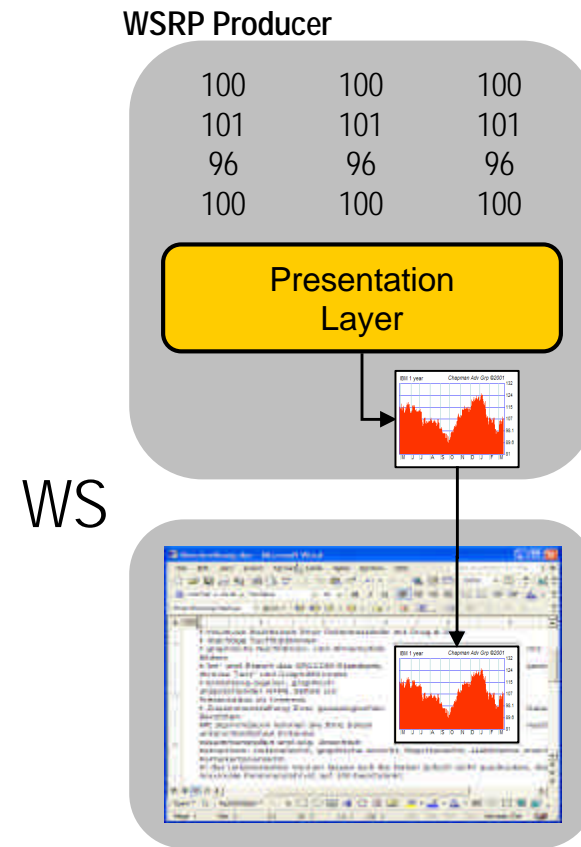
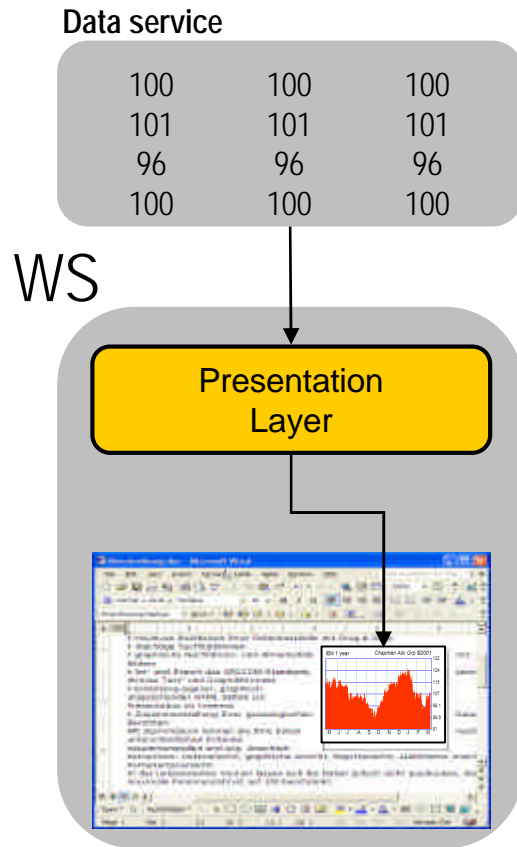
- Enable the sharing of portlets (markup fragments) over the internet



Goals of Web Services for Remote Portlets (WSRP)

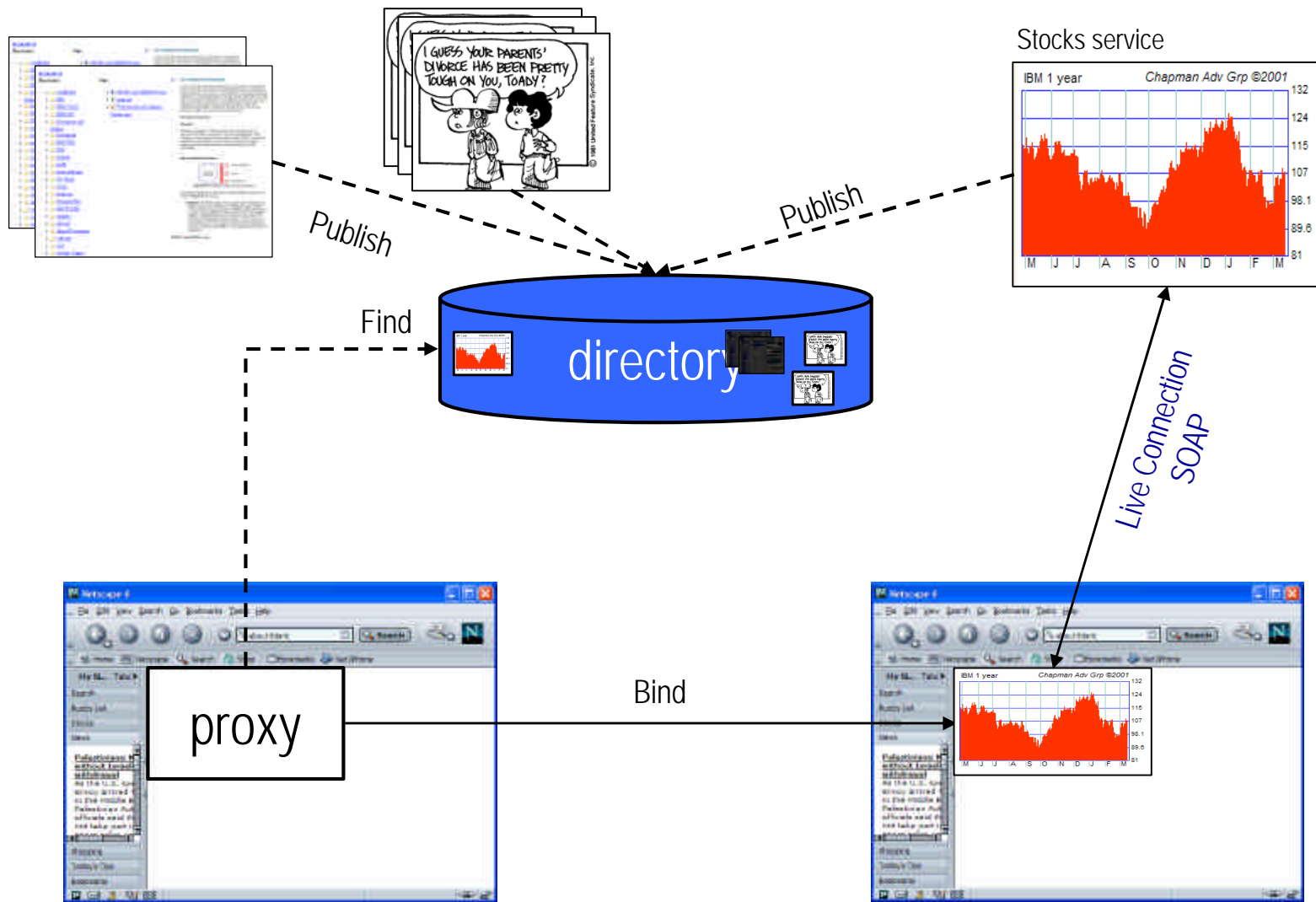
- Allow **visual, interactive**, user-facing web services to be **easily plugged into** all standards-compliant portals
- Let anybody **create and publish** their content and applications as user-facing web services
- Portal administrators can browse public or private directories for user-facing web services to plug into their portals as new portlets, **without any programming effort**
- Let portals **interact** and publish portlets so that they can be consumed by other portals
- Make the internet a **pool** of visual web services, waiting to be integrated

Remote Portlets vs. Data Oriented WS



- WSRP ⇔ visual & user facing & interactive

WSRP Advertising



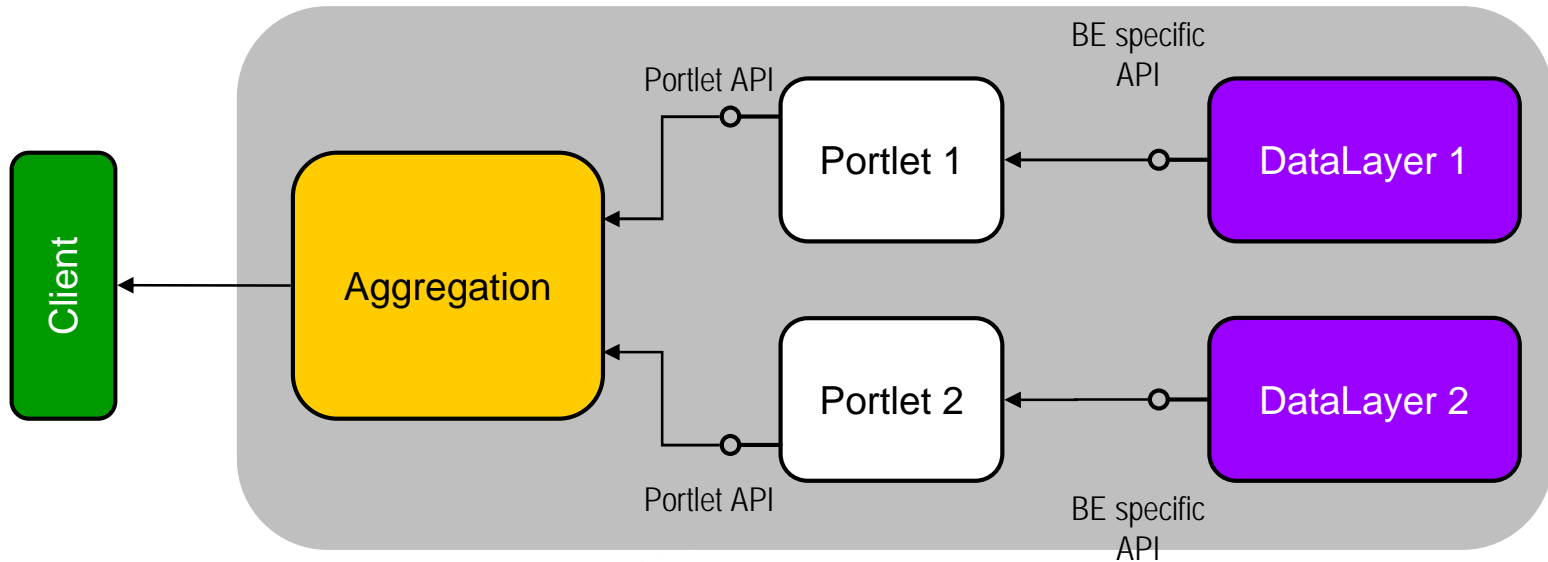
Relationship to other Standards

- **SOAP** (Simple Object Access Protocol)
 - Basis for the communication layer
 - Guarantees interoperability
- **WSDL** (WebService Definition Language)
 - Used as the interface definition
- **UDDI** (Universal Description & Discovery Interface)
- **ebXML Registry**
- **JSR168** (Definition of the Java Portlet API)
 - Defines the preferred JAVA API to implement WSRP entities
- **JSR109** (Integration of J2EE and WS)
- **WS-Security**
- **WS-Trust**

Summary: Traditional Back-End Usage Scenario

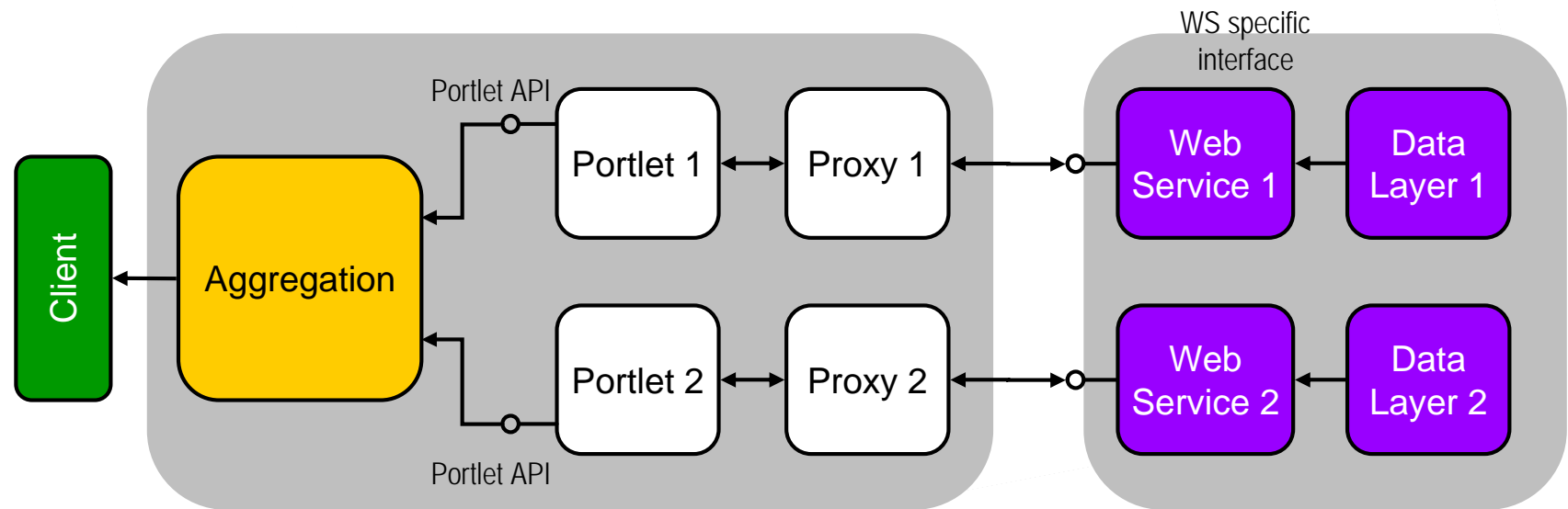
Local Portlets

- Efficient 😊
- Local deployment of code ☹️
- Specific UI for each deployed portlet ☹️
- Business layer and presentation layer both located on the portal server ☹️
- Portlets cannot be shared between portals!! ☹️



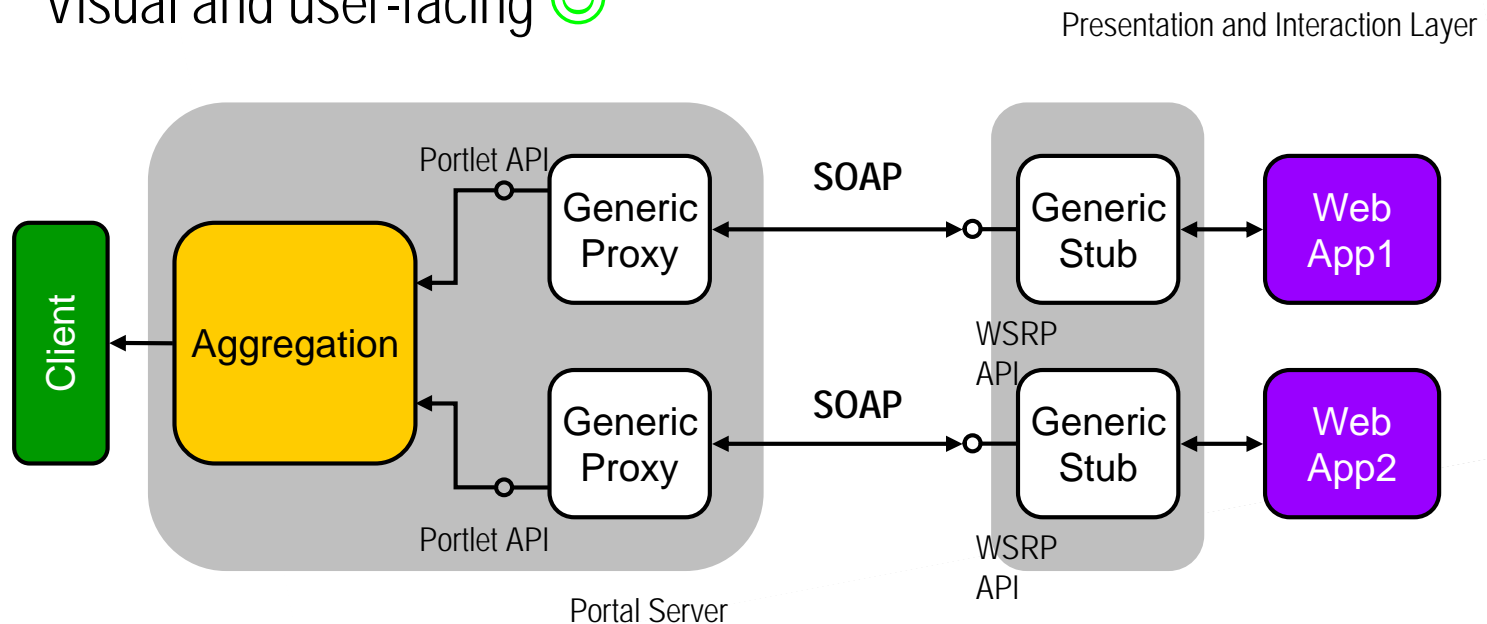
„Traditional“ Web Service Usage Scenario

- Portlets using Web Services
 - Different Web Services expose different interfaces ☹️
 - Specialized UI and proxy code required for each WS ☹️
 - Local deployment of code is still necessary ☹️
 - Data layer separated from presentation layer 😊



WebServices for Remote Portlets (WSRP)

- All remote connections share a unified API 😊
- No coding required, proxy and stub are coded **once** or generated automatically 😊
- Stable and standardized transport mechanism (e.g. SOAP) 😊
- Visual and user-facing 😊



WSRP Summary

- What's covered
 - Inter-portal communication via SOAP
 - Concept of transient and persistent state
 - User profile
 - Registration
 - Secure/non-secure transport
 - Simple Caching (enhanced by vendor extensions)
 - **Tight alignment with JSR168**
- Schedule
 - 1st public draft by end of Feb 2003
- What's **NOT** covered
 - Authorization
 - UDDI support
 - Eventing/messaging

Questions?
Questions?

Questions!!!