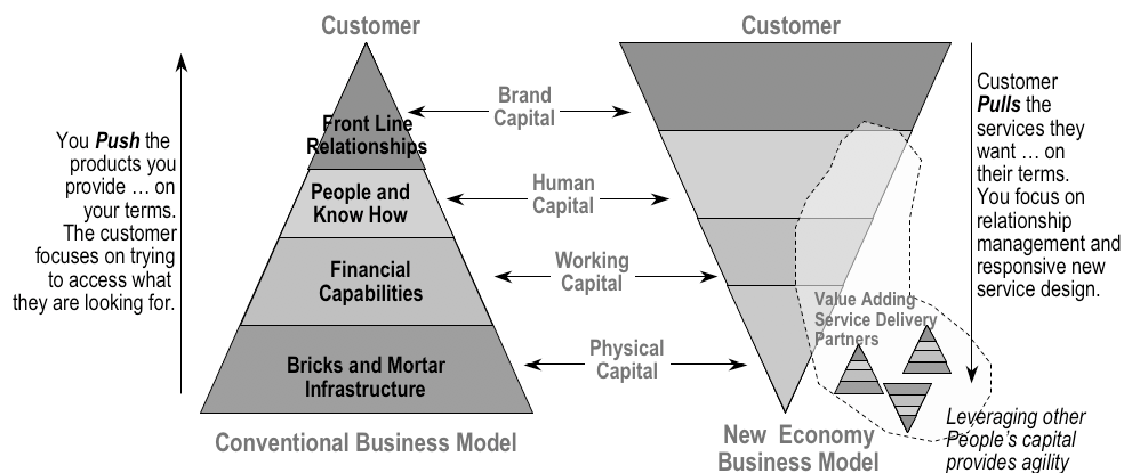


# The Web as an SOA platform

- “Start Here to learn about Web services”
  - <http://www-106.ibm.com/developerworks/webservices/library/ws-starthere.html>
- “The Web Services Conceptual Architecture”
  - <http://www-3.ibm.com/software/solutions/webservices/pdf/WSCA.pdf>
- and a presentation by F. Curbera
  - <http://www.ece.rutgers.edu/~parashar/Classes/01-02/ece566/slides/WebServices.pdf>



It's a services world, no longer just a products world



# **The role of Information Systems in Service Delivery**

- The system as an archive
  - The system stores records
- The system as a support infrastructure to employees delivering service to customers
  - Employees may use the system in several steps of their tasks, to access information and to record more details of the process
- The system as a means of service delivery
  - The system is exposed (at least partially) to customers over the web
- The service-delivery system is the product
  - Software companies develop information systems for other companies to deliver their services, usually based on general application frameworks and standards-based Service-Oriented Architectures
- The service-delivery system motivates partnerships
  - Businesses may engage in partnerships for delivering added-value services by integrating their service-delivery systems

## **Some specific scenarios**

- Intra-organization software consolidation
  - A company needs
    - to have all its divisions use the same software
    - to develop on the same infrastructure
    - to customize fast for special problems
- Inter-organization integration
  - A company wants to enable access to some of its software to customers and partners, to automate interactions
- Value-added service provision
  - A company wants to consolidate its purchase orders with other companies

# The Web

- The Web was designed for humans - web sites for information dissemination
  - Few, tolerant standards (http + html)
  - Various computing platforms
- It has been extremely successful
  - It is ubiquitous
  - It has enabled information sharing and B2C e-commerce
- From web sites to web applications
  - HTML + Javascript + CGI + Server-side scripting
  - Java (applets, servlets, J2EE)
  - New languages and frameworks - RubyOnRails
- Web Services standards

## Objectives for Web Services

- “Web services” is a set of open standards for **web-based middleware**, designed to
- Enable “universal” interoperability
    - Across programming languages and middleware frameworks
    - With minimal amount of required infrastructure
  - Support a service oriented architecture (SOA)
    - Focusing on analysis in terms of tasks and communication through messages and documents, not on object APIs.
    - For loosely-coupled systems

**Widespread adoption?**

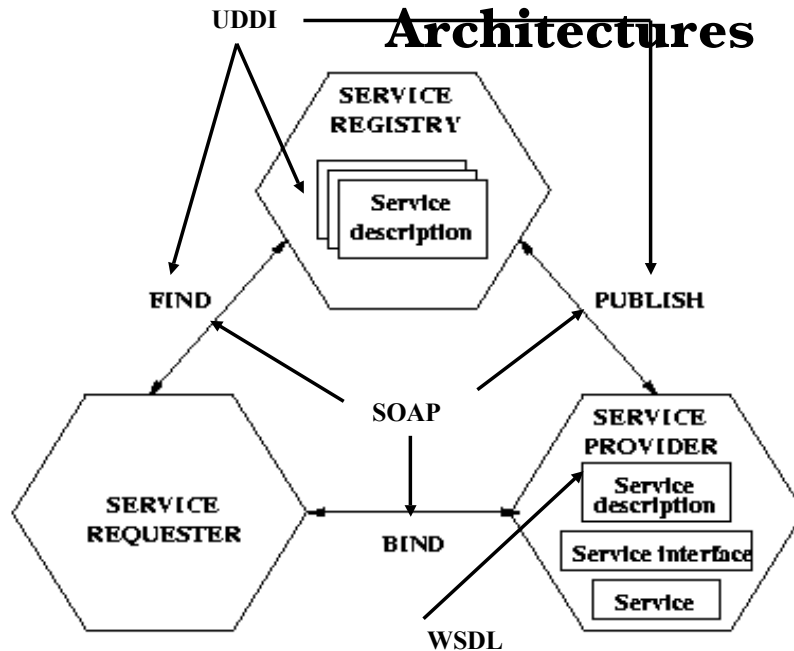
## **Technically a “Web Service” is**

- A platform- and implementation- independent software component that can be
  - **Described** using a service-description language
  - **Published** to a registry of services
  - **Discovered** through a standard mechanism
  - **Invoked** through a declared API, through a network
  - **Composed** with other services

## **“Web Service” (IBM) definition**

- A Web service is an interface that describes a collection of operations that are network-accessible through standardized XML messaging.
- A Web service is described using a standard, formal XML notion, called its service description. It covers all the details necessary to interact with the service, including message formats (that detail the operations), transport protocols and location.
  - The interface hides the implementation details of the service, allowing it to be used independently of the hardware or software platform on which it is implemented and also independently of the programming language in which it is written.
- This allows and encourages Web Services-based applications to be loosely coupled, component-oriented, cross-technology implementations.
- Web Services fulfill a specific task or a set of tasks. They can be used alone or with other Web Services to carry out a complex aggregation or a business transaction.

# Service-Oriented Architectures



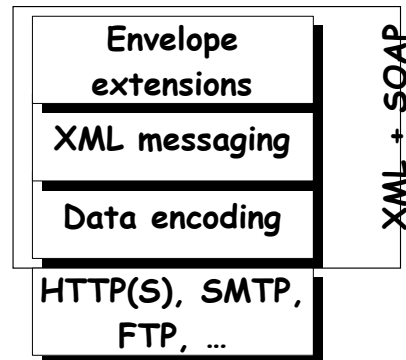
From the notes on a graduate course at the University of Lappeenranta, Finland. August 9-13, 2004. Gustavo Alonso, Cesare Pautasso].  
<http://www.inf.ethz.ch/personal/alonso/teaching.html>

## Publish - Find - Bind

- Service providers *publish* services by advertising service descriptions in the registry such as UDDI
  - WSDL (Web Services Description Language) is an XML-based syntax for describing the service IDL
- Service requestors use *find* operation to retrieve service descriptions from the service registry.
  - It is the requestor's responsibility to assess the appropriateness of the web services available
- Service requestors *bind* to service providers using binding information found in service descriptions to locate and invoke a service using SOAP.

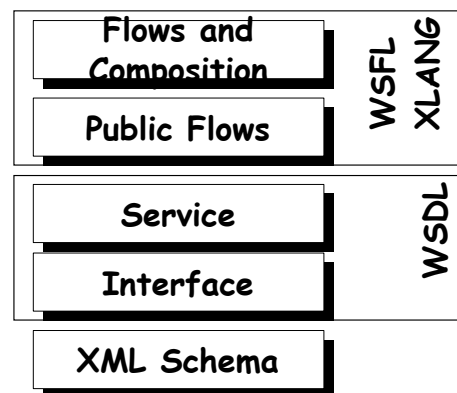
## **“what goes on the wire” the wire stack**

- SOAP defines:
  - An XML envelope for XML messaging,
    - Headers + body
  - An HTTP binding for SOAP messaging.
    - SOAP is “transport independent”.
  - A convention for doing RPC.



## **“what describes what goes on the wire” the description stack**

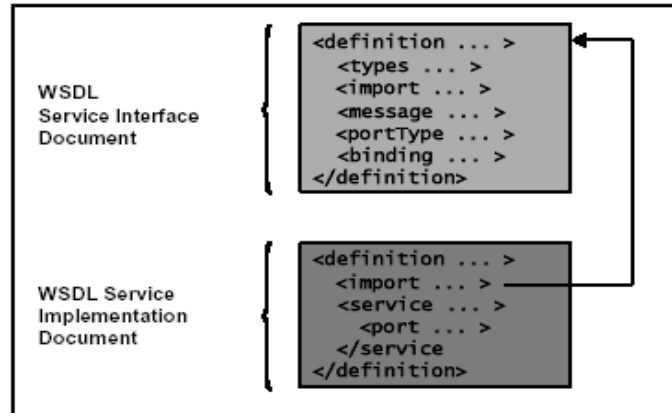
- Integration requires interoperable machine-understandable descriptions
- Enables dynamic, delayed binding of components.
- Language extensibility provides support for different levels of application integration.



# Web Services Description Language

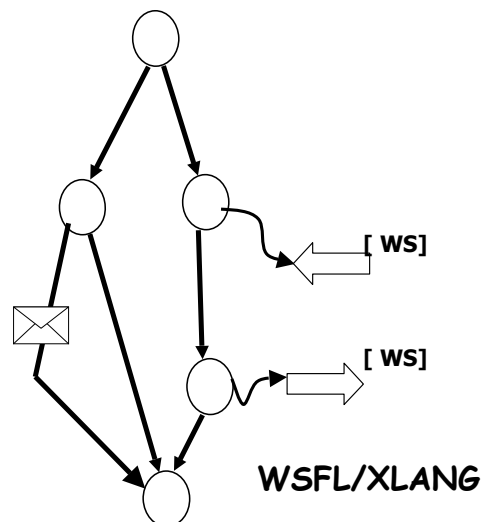
- Provides functional description of the network services:

- IDL description
  - I/O types
  - Operation signatures
- Network bindings



## Workflow Specification

- Composition/Orchestration: a description of how a collection of web services work together to produce an application



## **“how to find these descriptions” the discovery stack**

- Inspection: looking at the details of the web service, knowing its URI
- Directory: capability-based discovery of business partners

