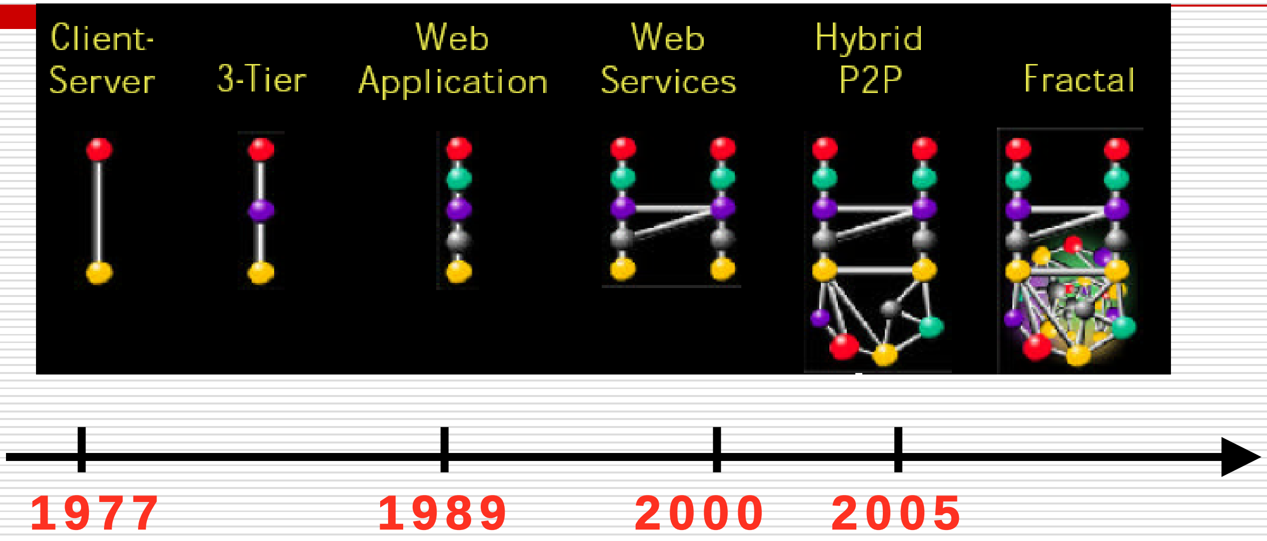
Web Services Introduction

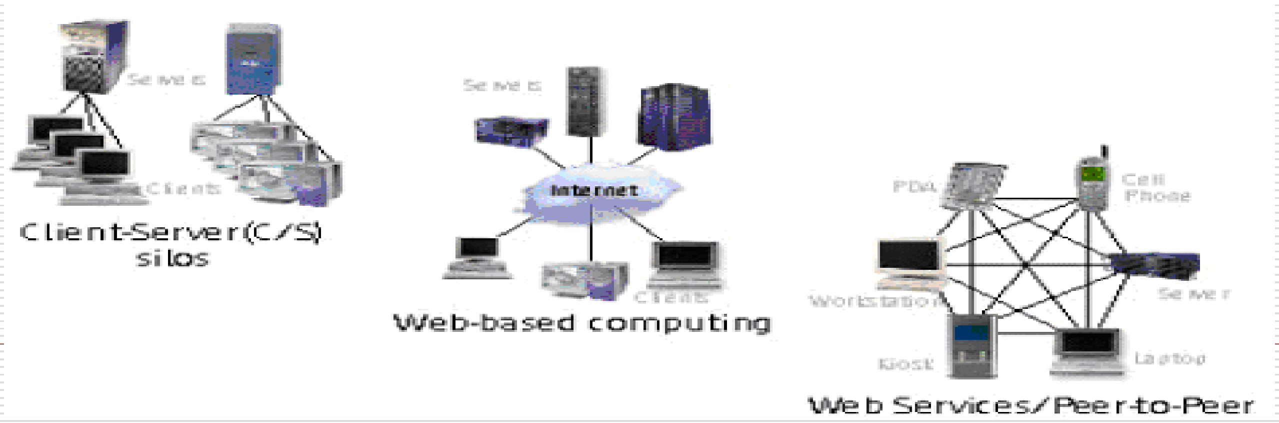
申丽萍 Shen Li-ping

Shen-lp@cs.sjtu.edu.cn

Contents

Distributed Computing Evolution

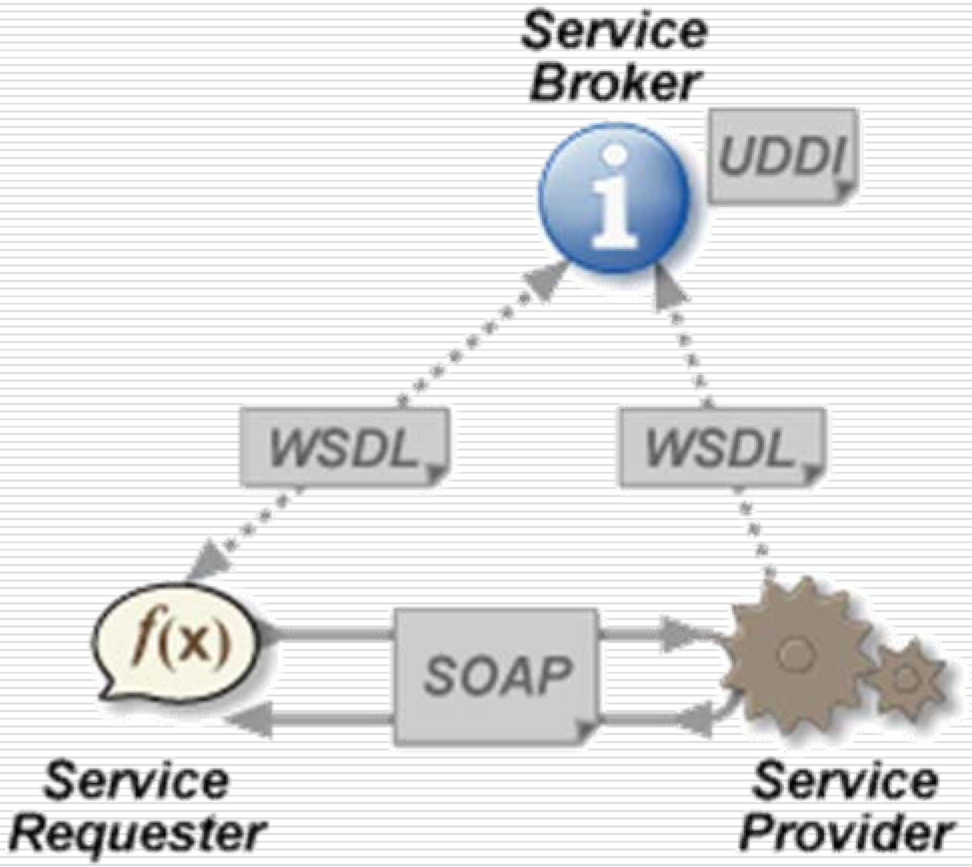




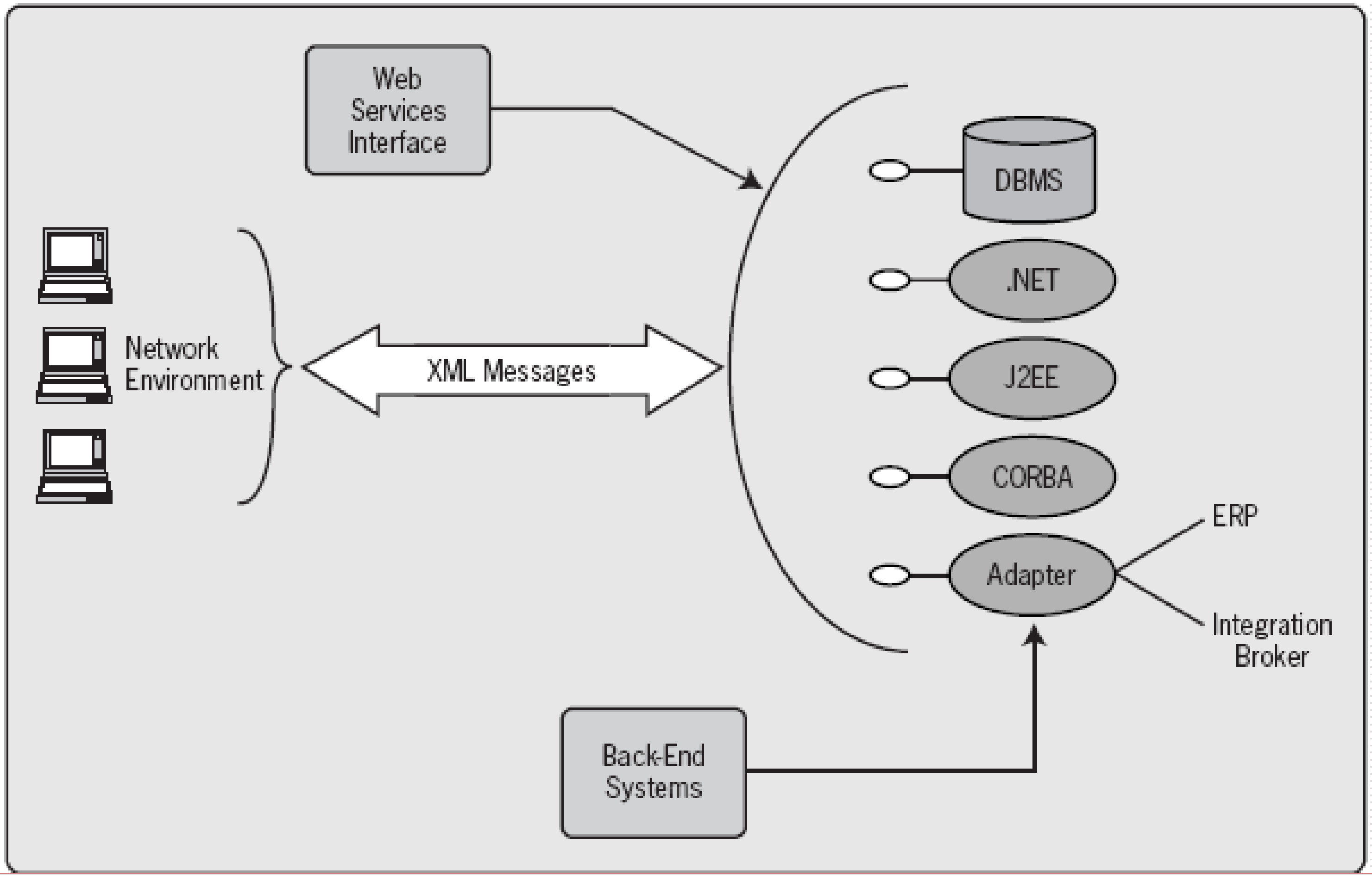
What is Web Services?

According to W3C, “A Web service is a **software application** identified by a **URI**, whose **interfaces** and binding are capable of being defined, **described** and discovered by **XML** artifacts and supports direct interactions with other software applications using **XML based messages** via **internet-based protocols**.”

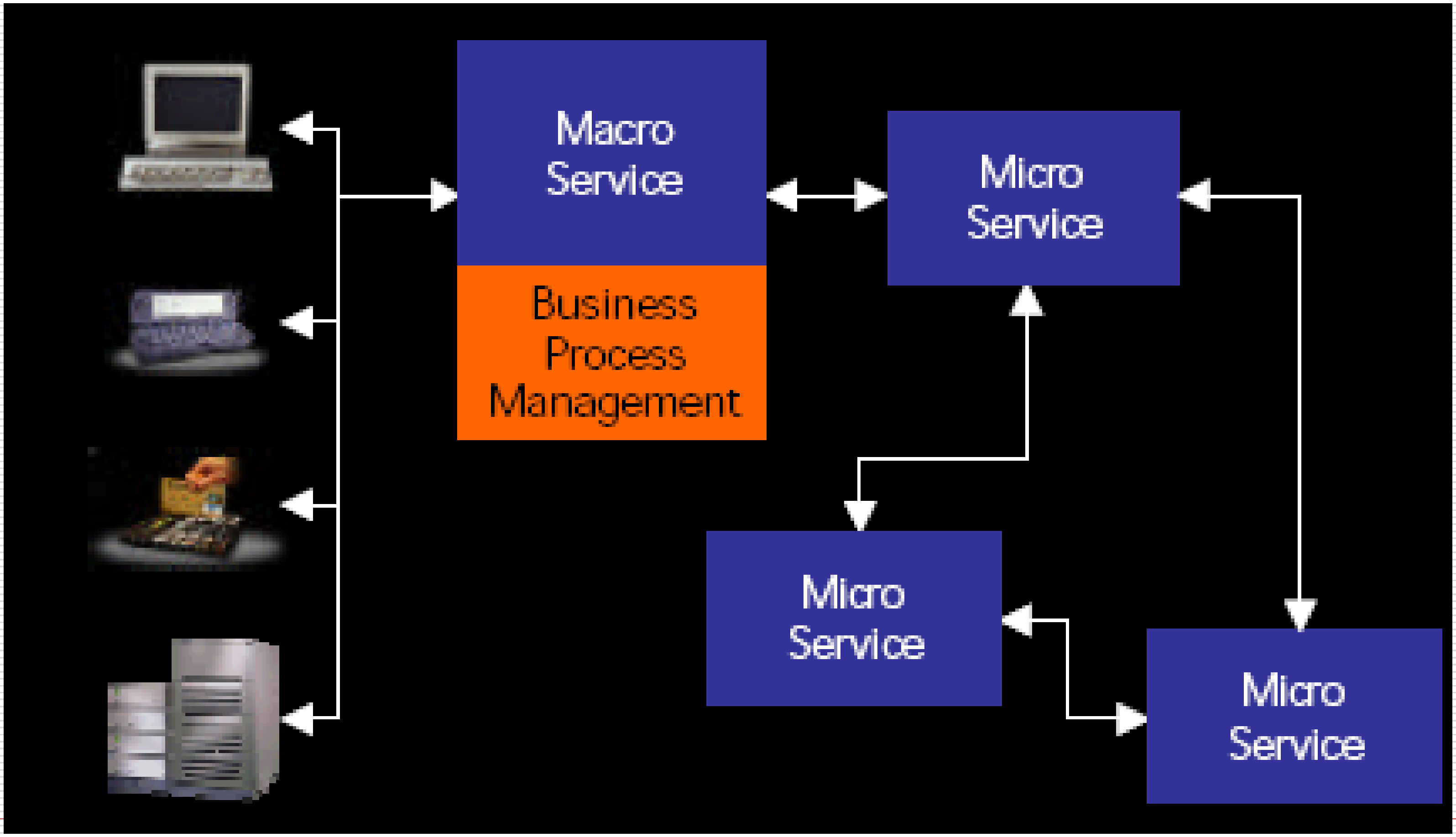
Web Service basic protocol stack consists of XML artifacts: WSDL, SOAP and UDDI.



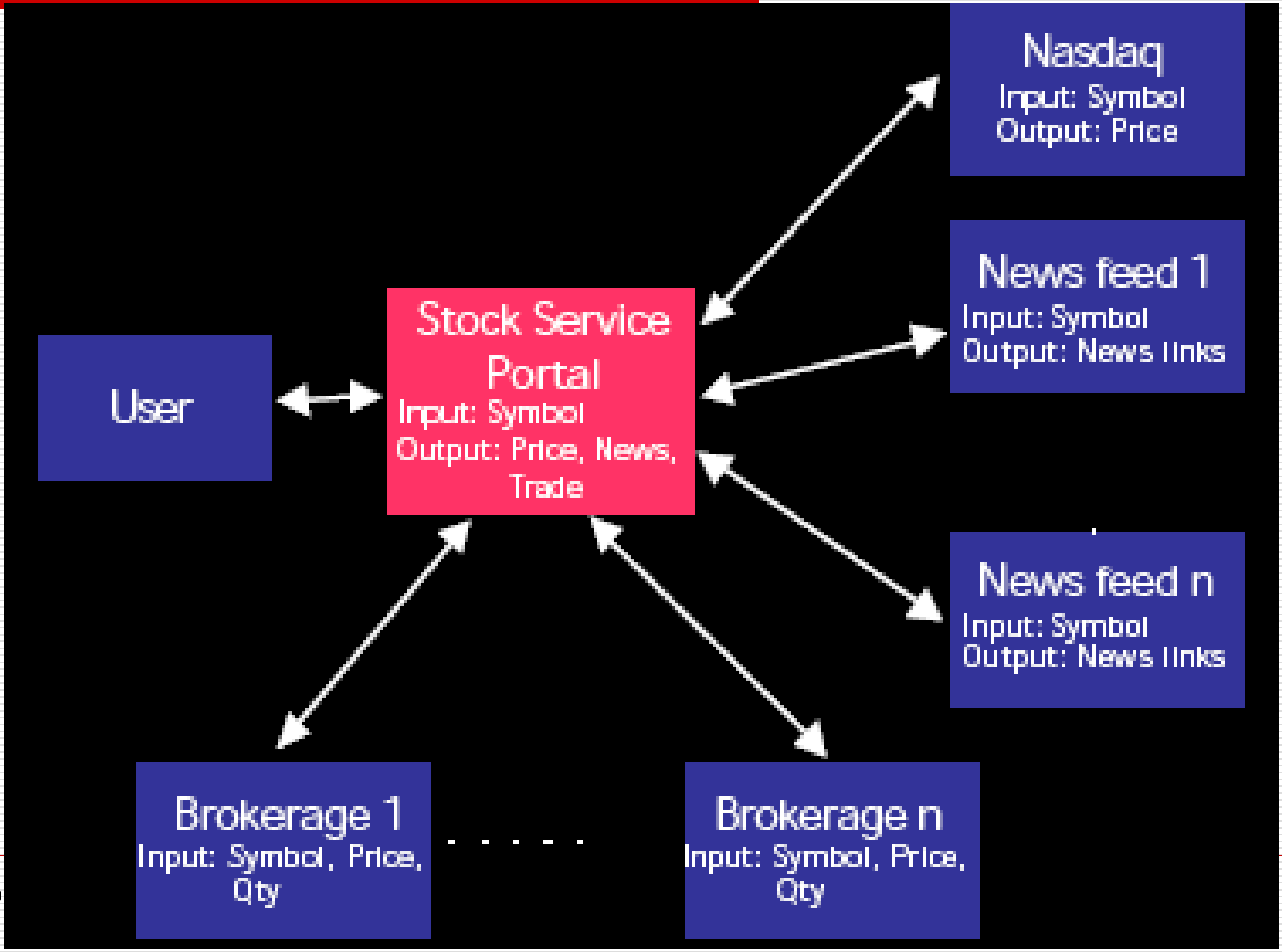
Web services provide a standard means of interoperability between different applications on a variety of platforms.



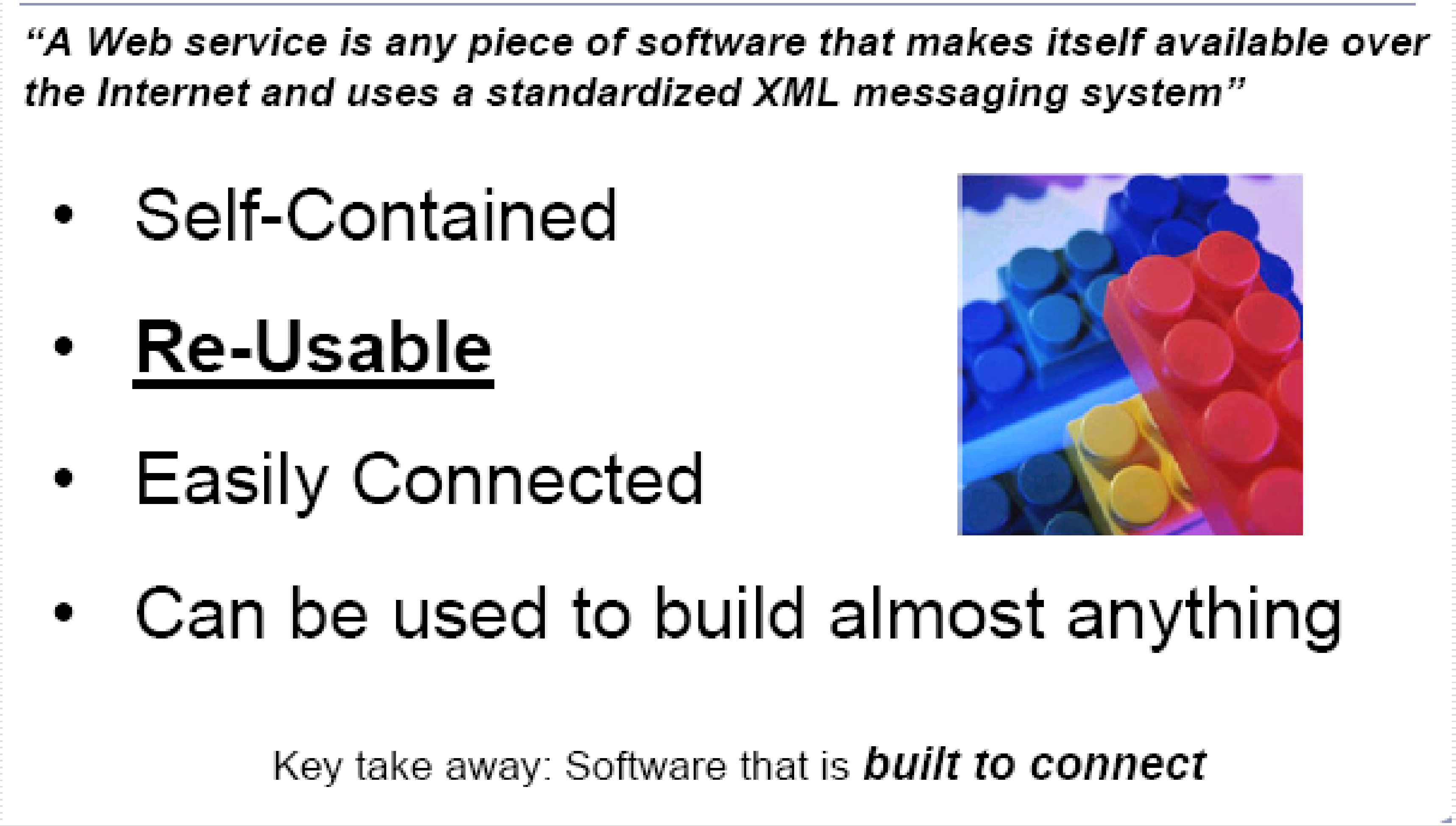
Mashup: Macro Services could be assembled from micro-services.



Mashup: service portal can aggregate information from different service sources



Like Legos, Web-Services are….



"A Web service is any piece of software that makes itself available over the Internet and uses a standardized XML messaging system"

- Self-Contained
- <u>**Re-Usable**</u>
- Easily Connected
- Can be used to build almost anything

Key take away: Software that is ***built to connect***

Web Services vs. Traditional C/S

Between enterprises	Within enterprise
Program language independent	Tied to a set of programming languages
Message-driven	Procedural
Easily bound to different transports	Bound to a proprietary transport
Loosely-coupled	Tightly-coupled
Relatively not efficient processing	Efficient processing

Web Services vs. Web Application

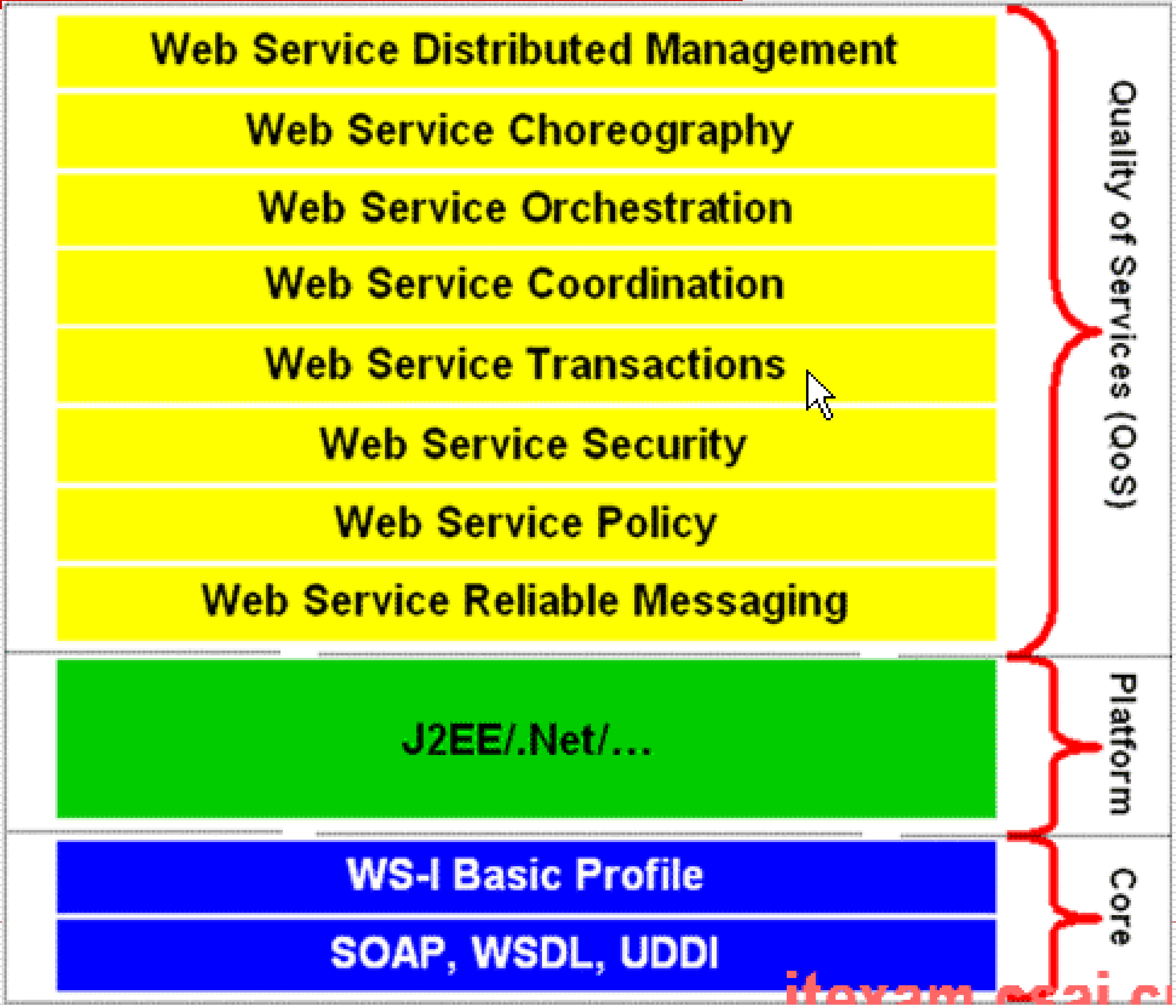
- Program-to-program Interaction
- Dynamic service integration
- Service aggregation

- User-to-program Interaction
- Static Integration of components
- Monolithic service

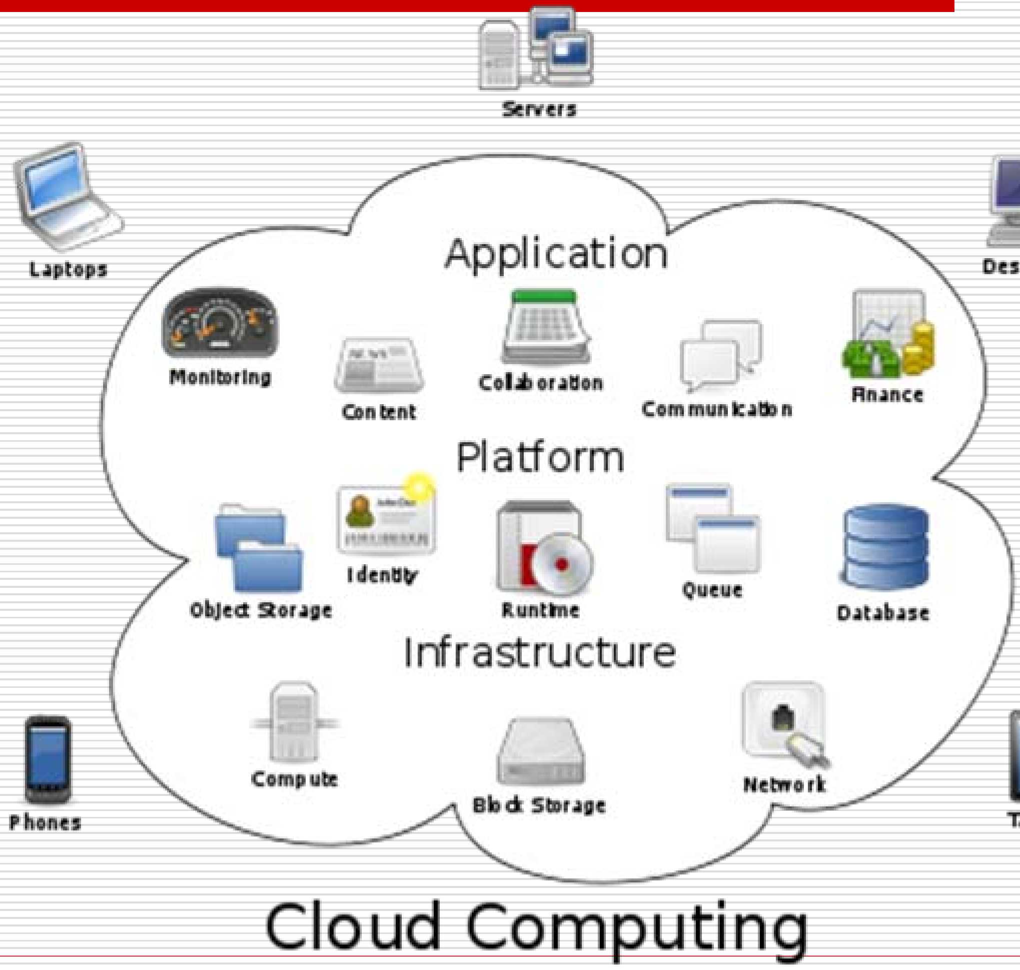
Web2.0 is a Web-as-participation-platform based on REST and SOAP interactions where we can all meet and read and write.

Focus on Simplicity
Wikis
Joy of Use
Folksonomy
AJAX
Usability
Recommendation
The Long Tail
Social Software
Participation
Economy
Blogs
Affiliation
Web2.0 beta
Audio
Convergence
Design
Video
Mobility
Remixability
RSS
Standardization
OpenAPIs
CSS-Design
DataDriven
Web Standards
Microformats
MARKUS ANGERMEIER @ APERT

Web Services is enabling technology for SOA

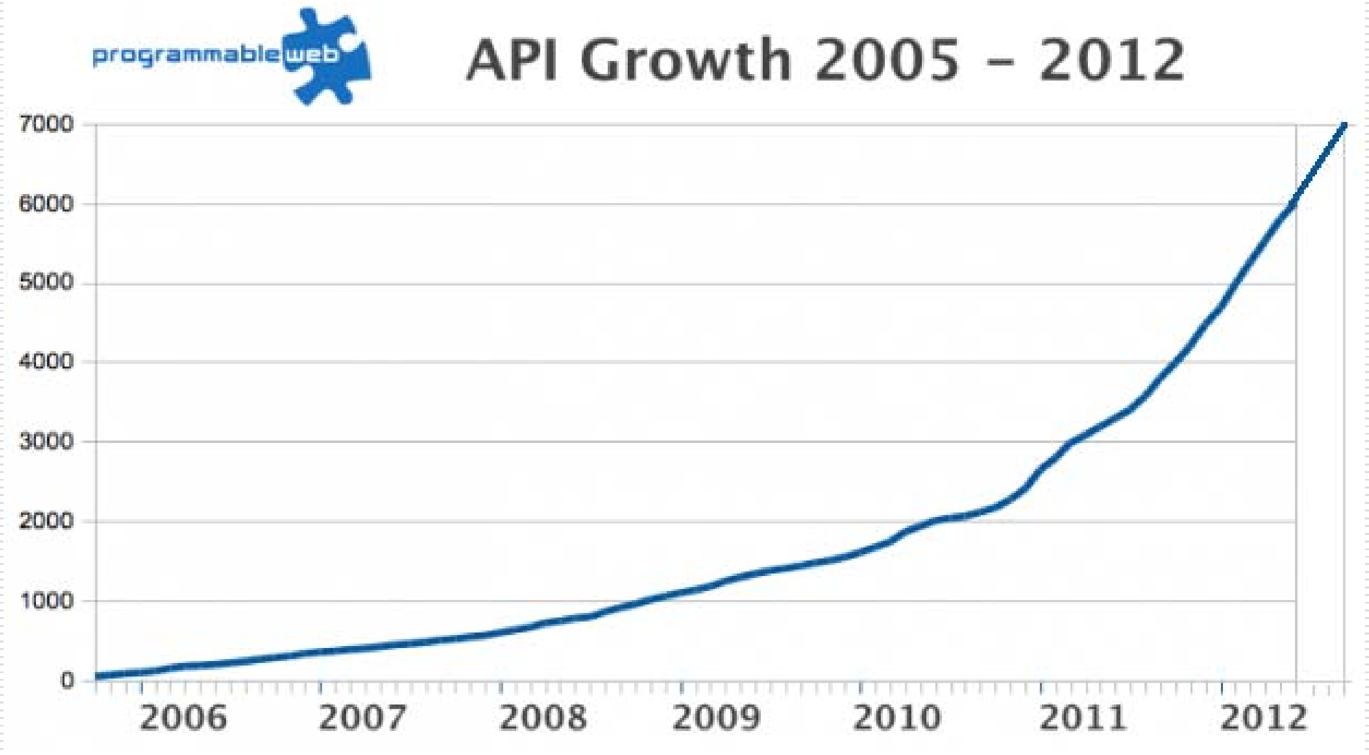


Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over Internet.



- **SaaS:** Software as a Service
- **STaaS:** STorage as a Service
- **DaaS:** Data as a Service
- **PaaS:** Platform as a Service
- **IaaS:** Infrastructure as a Service

Web Service (API) universe continues to expand rapidly



Source from programmableWeb:
http://blog.programmableweb.com/2012/08/23/7000-apis-twice-as-many-as-this-time-last-year/

Characteristics of Web Services

- XML based everywhere
- Message-based
- Programming language independent
- Could be dynamically located
- Could be dynamically assembled or aggregated
- Accessed over the internet
- Nicer encapsulation
- Loosely coupled
- Based on industry standards

Why Web Services

- **Interoperable** – Connect across heterogeneous networks using ubiquitous web-based standards
- **Economical** – Recycle components, no download ,installation, configuration and maintenance. developers can quickly create and deploy them using many tool-kits available on the Web
- **Automatic** – No human intervention required even for highly complex transactions, EAI
- **Accessible** – Legacy assets & internal apps are exposed and accessible on the web
- **Available** – Services on any device, anywhere, anytime. Web services communicate using HTTP and XML. Any connected device that supports these technologies can both host and access Web services.
- **Scalable** – No limits on scope of applications and amount of heterogeneous applications

4Y3N for Web Services

- Suitable for applications with distributed large amount of users
- Suitable for EAI & B2B Integration
- Suitable for submitting documents to long running business process flows.
- Suitable for software/component and data reuse
- Not suitable for close systems.
- Not suitable for standalone PC and LAN isomorphic systems.
- Not suitable for performance-vital and reliability-vital systems.

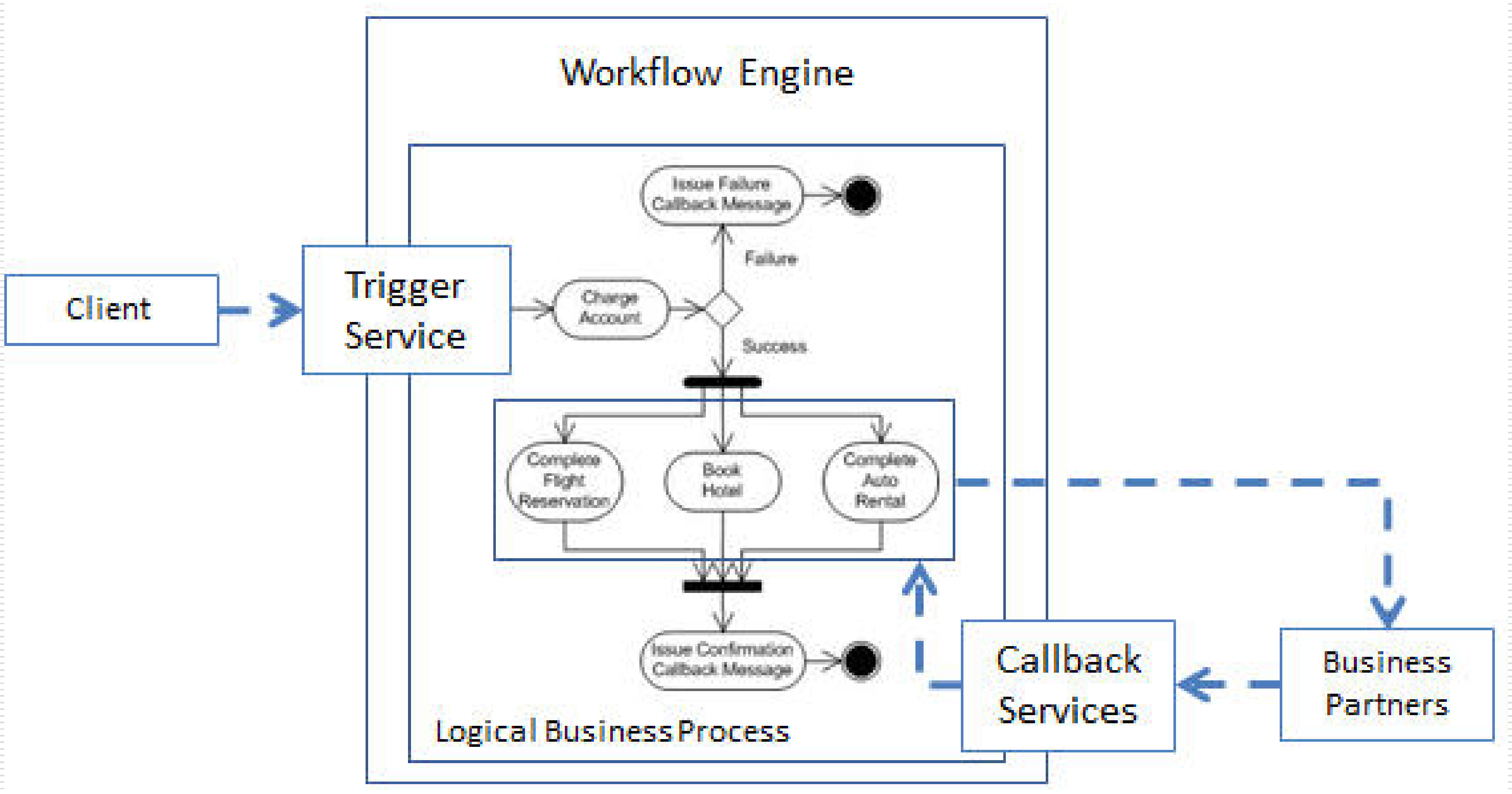
Types of Web Services (1)

- Remote Procedure Calls: RPC Web services present a distributed function (or method) call interface, the basic unit of RPC Web services is the WSDL operation.
- Message-oriented architecture: the basic unit of communication is a message, rather than an operation.
- Resource-oriented architecture: architectures that focus on interacting with stateful resources, rather than messages or operations.

Types of Web Services (2)

- Big Web services: SOAP based, have complex business logics.
- Web API: REST based, use HTTP or similar protocols by constraining the interface to a set of well-known, standard operations (like GET, POST, PUT, DELETE for HTTP), along with a definition of the structure of response messages, usually expressed in an XML or JSON (JavaScript Object Notation) format

Workflow engine manages the life cycle and execution of tasks within complex or long-running business processes.



Examples-HelloWorld Service

- step 1: Create Hello World Web Service
  - un-note the noted hello world example web method codes.
- step 2: Test Hello World Web Service with IE
  - http://localhost/.../HelloWorld/Service1.asmx
- step 3: Test Hello World Web Service with windows application
  - Add web reference http://localhost/.../HelloWorldService/Service1.asmx to create proxy class,then invoke it locally.

Examples-CurrencyConvertorClient

- CurrencyConvertor Web Services WSDL document address：http://www.webservicex.net/CurrencyConvertor.asmx?wsdl
- This is just a simple Windows Application to invoke one of the free accessible Web Services. Add a web reference to create a proxy class.

Assignment#1 (don’t submit)

- Find some Live Web Services:
  - Amazon.com Web Services
  - MapPoint.net Web Services
  - Webservicex.net Web Services
  - ……
- Sign up for free Amazon Web Services usage (http://aws.amazon.com/free/) and find out what services they provide.
- Find out what services the Openstack provide: http://www.openstack.org/software/
- Access one of the live web services in your application. Add a web reference to create a proxy class for the web services with VS.NET.