أكاديمية الحكومة الإلكترونية الفلسطينية

The Palestinian eGovernment Academy

www.egovacademy.ps

Tutorial III:

Process Integration and Service Oriented Architectures

Session 1
Introduction to SOA

Prepared By

Mohammed Aldasht

Reviewed by
Prof. Marco Ronchetti and Prof. Paolo Bouquet, Trento University, Italy

European Commission
TEMPUS

About

This tutorial is part of the PalGov project, funded by the TEMPUS IV program of the Commission of the European Communities, grant agreement 511159-TEMPUS-1-2010-1-PS-TEMPUS-JPHES. The project website: www.egovacademy.ps

Project Consortium:

Birzeit University, Palestine
(**Coordinator**)

Palestine Polytechnic University, Palestine

Palestine Technical University, Palestine

Ministry of Telecom and IT, Palestine

Ministry of Interior, Palestine

Ministry of Local Government, Palestine

University of Trento, Italy

Vrije Universiteit Brussel, Belgium

Université de Savoie, France

University of Namur, Belgium

TrueTrust, UK

Coordinator:

Dr. Mustafa Jarrar
Birzeit University, P.O.Box 14- Birzeit, Palestine
Telfax:+972 2 2982935 mjarrar@birzeit.edu

© Copyright Notes

Tutorial Map

Intended Learning Objectives

A: Knowledge and Understanding

- 3a1: Demonstrate knowledge of the fundamentals of middleware.
- 3a2: Describe the concept behind web service protocols.
- 3a3: Explain the concept of service oriented architecture.
- 3a4: Explain the concept of enterprise service bus.
- 3a5: Understanding WSDL service interfaces in UDDI.

B: Intellectual Skills

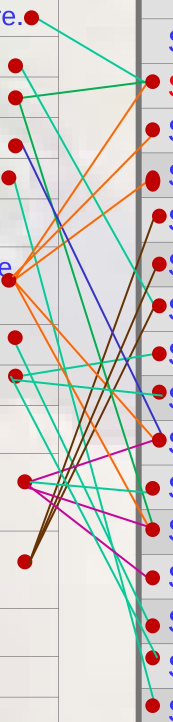
- 3b1: Design, develop, and deploy applications based on Service Oriented Architecture (SOA).
- 3b2: use Business Process Execution Language (BPEL).
- 3b3: using WSDL to describe web services.

C: Professional and Practical Skills

- 3c1: setup, Invoke, and deploy web services using integrated development environment.
- 3c2: construct and use REST and SOAP messages for web services communication.

D: General and Transferable Skills

- d1: Working with team.
- d2: Presenting and defending ideas.
- d3: Use of creativity and innovation in problem solving.
- d4: Develop communication skills and logical reasoning abilities.



Title	T	Name
Session0: Syllabus and overview	0	Aldasht
Sesson1: Introduction to SOA	2	Aldasht
Session2: XML namespaces & XML schema	2	Aldasht
Session 3: Xpath & Xquery	4	Romi
Session4: REST web services	3	M. Melhem
Session5: Lab2: Practice on REST	3	M. Melhem
Session 6: SOAP	2	Aldasht
Session 7: WSDL	3	Aldasht
Session8: Lab 3: WSDL practice	3	Aldasht
Session9: ESB	4	Aldasht
Session10: Lab4: Practice on ESB	4	Aldasht
Session11: integration patterns	4	M. Melhem
Session12: Lab5: integration patterns	4	M. Melhem
Session13: BPEL	3	Aldasht
Session14: Lab6: Practice on BPEL	3	Aldasht
Session15: UDDI	2	Aldasht

Session 1: Introduction to SOA

Session ILOs

After completing this module students will be able to:

1. Demonstrate knowledge of the fundamentals of middleware
2. Explain the concept of service oriented architecture
3. Describe some characteristics of web services and applications based on SOA

Session Outlines

- **Introduction**
  - SOA concepts and terms.
  - SOA key principles and basic role.
- Middleware
- Web services
- Palestinian e-Government Architecture
- Summary

What is SOA?

- SOA: an architectural style based on loosely coupled interacting software components that provide services [3].
  - A **service** is a piece of functionality provided by a service provider and delivers final results to the service consumer.
- SOA based application packages functionality as a suite of interoperable services to be used within multiple systems from several business domains [5].
- The core concept of SOA is **loose coupling** of software services.
  - Means organizing different services to minimize their interdependencies
- This is much more adaptable than expensive and tightly coupled technologies, like CORBA and DCOM

SOA key terms

- A service: is a repeatable business task; e.g. renew driving license or register a student.
- Service oriented: a way of integrating a business as a linked services and the outcomes that they bring.
- Service oriented architecture: an IT architectural style that supports service orientation [6].
- A composition application: a set of related and integrated services that support a business process built on an SOA.

Key principles of SOA

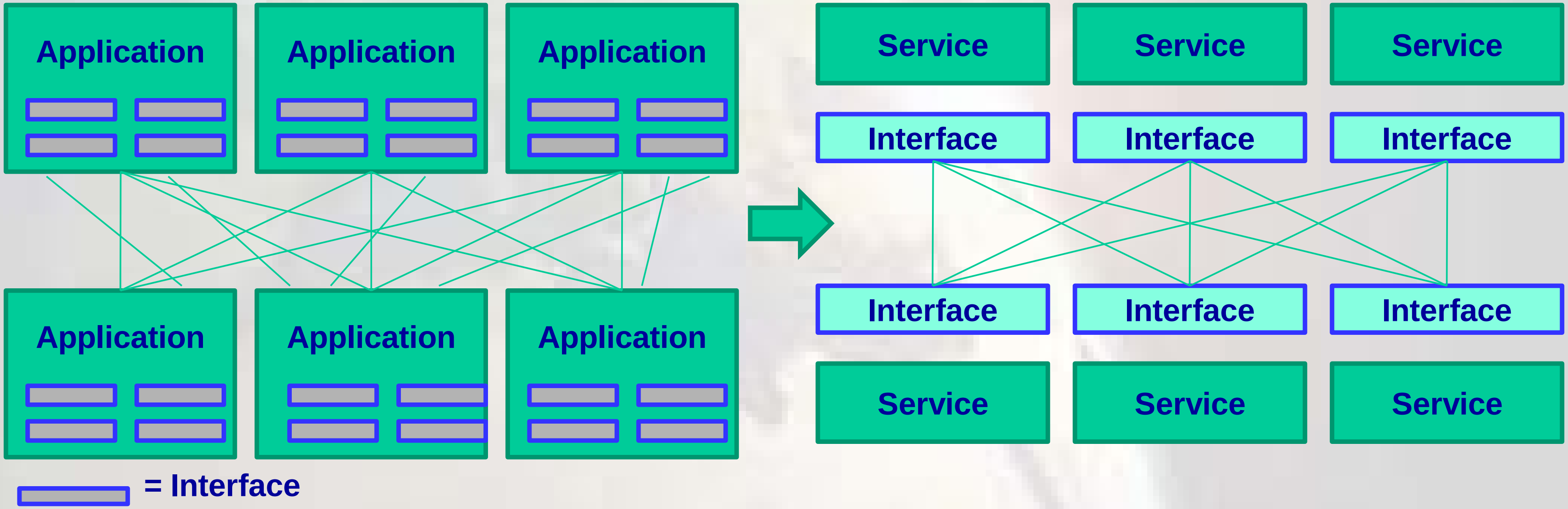
- Some key principles are [7]:

1. With *loose coupling*, a service logic can change with minimal impact on other services within the same SOA.
2. *Service contract:* descriptions and other documents describing how a service can be accessed.
3. *Abstraction* of underlying logic means that services interact with each other only via their public interfaces, hiding the service implementation details.
4. *Autonomy* means that services control only the logic they encapsulate
5. *Interoperability.*

SOA: Basic Role

- SOA decouples interfaces from applications.
- The number and complexity of interfaces is reduced.
- Business applications and their interfaces become reusable.
- Interfaces tightly coupled with point to point connections

SOA: Basic Role



Source, IBM education assistant:
http://publib.boulder.ibm.com/infocenter/ieduasst

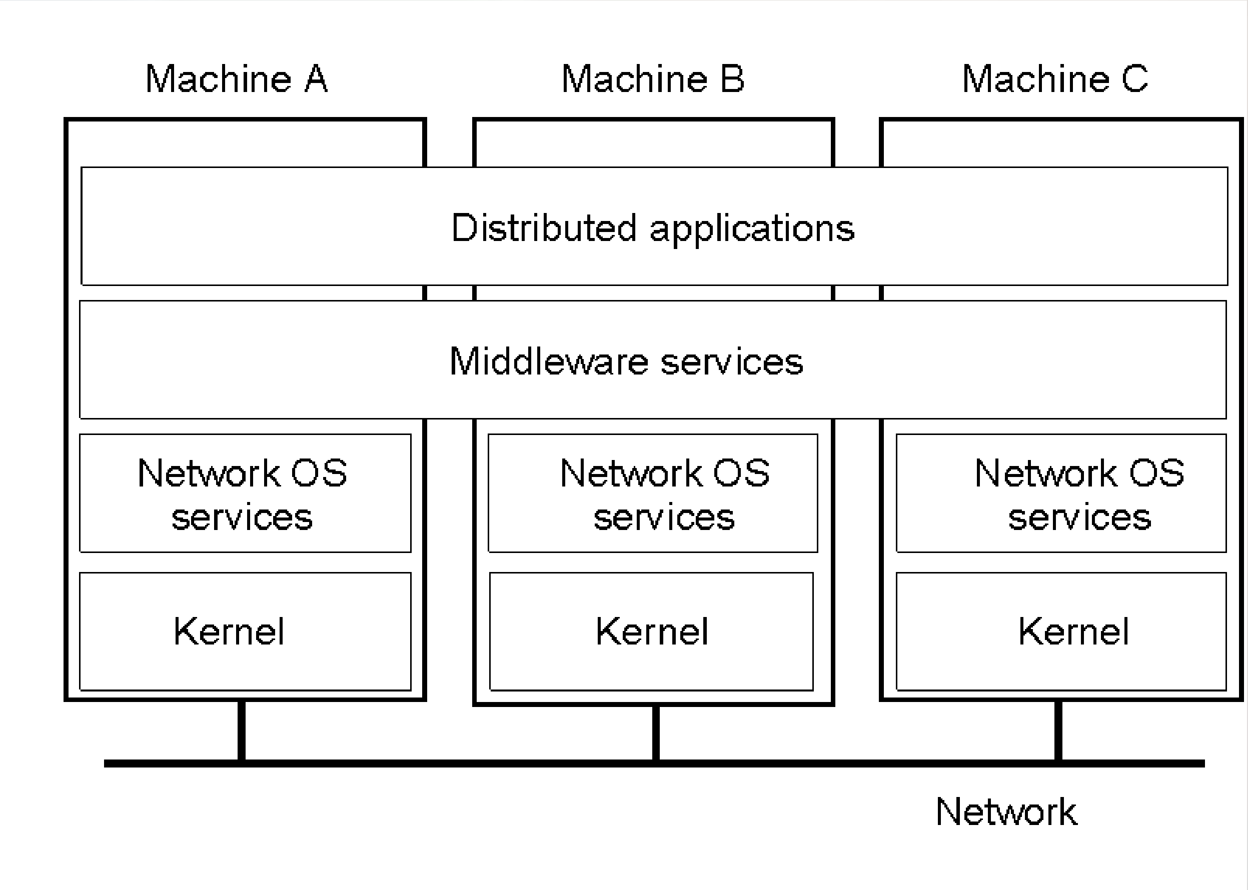
Session Outlines

- Introduction
  - SOA concepts and terms.
  - SOA key principles and basic role.
- **Middleware**
- Web services
- Palestinian e-Government Architecture
- Summary

Introduction to Middleware

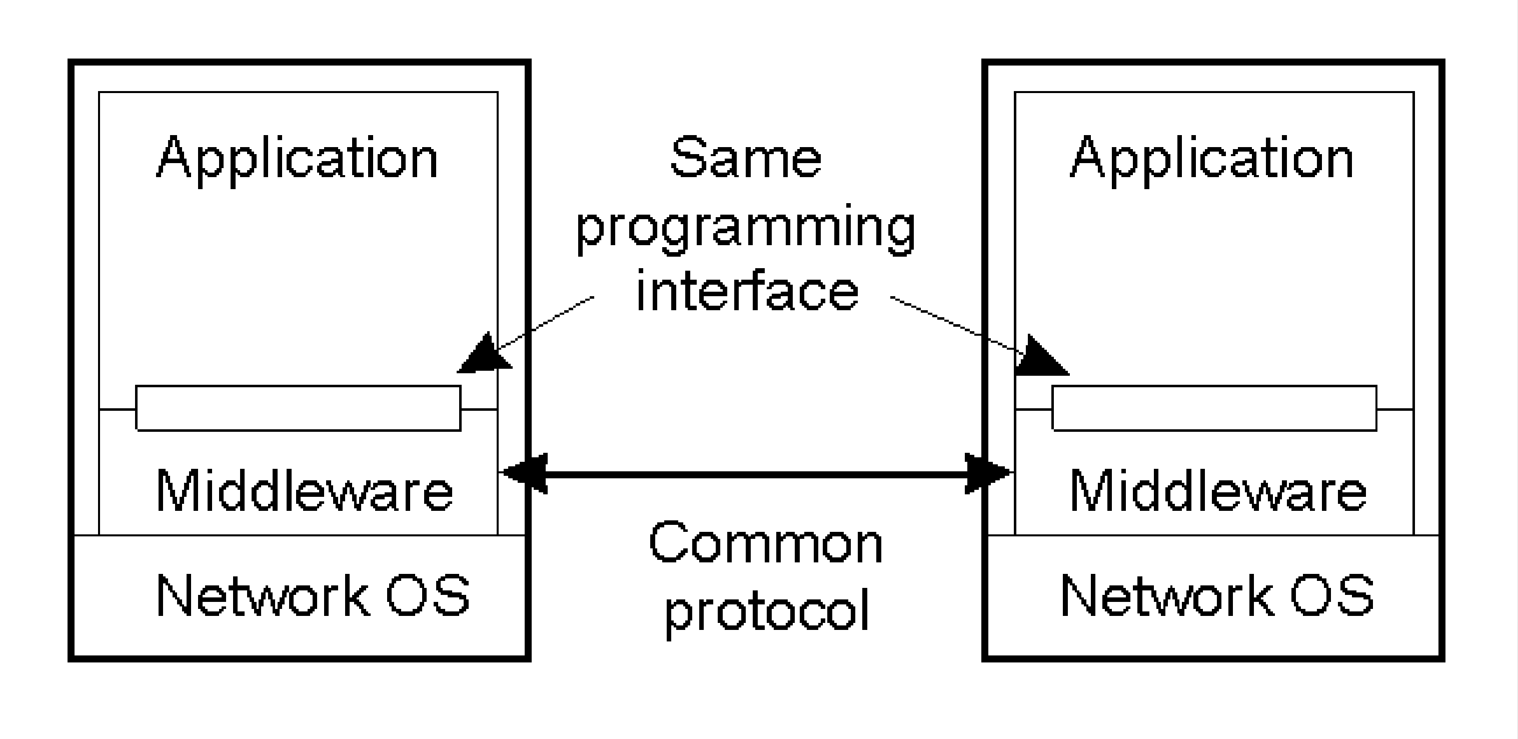
- A layer of software that lies between the application code and the run-time infrastructure [1].
  - Connects software components or people and their applications
  - Provide interoperability in support of the move to coherent distributed architectures
  - Allows data contained in one database to be accessed through another.
  - Fits enterprise application integration and data integration software.

Middleware-based distributed system



General structure of a distributed system as middleware, source [2]

Middleware protocols and interfaces



In an open middleware-based distributed system, the protocols used by each middleware layer should be the same, as well as the interfaces they offer to applications, source [2]

Communication models in Middleware

- 3 widely used communication models [2]:

1. RPC: a client calls procedures running on remote systems, can be *asynchronous* or *synchronous*.
2. Message Oriented Middleware (MOM)
3. Data streaming: communication in which timing plays a crucial role “Audio and video”.

- Generally, there are two ways to communicate:
  - **Enterprise messaging system:** message passing in standard formats, using XML, SOAP or web services.
  - **Enterprise Service Bus:** some type of integration middleware that supports both MOM and Web services.

Session Outlines

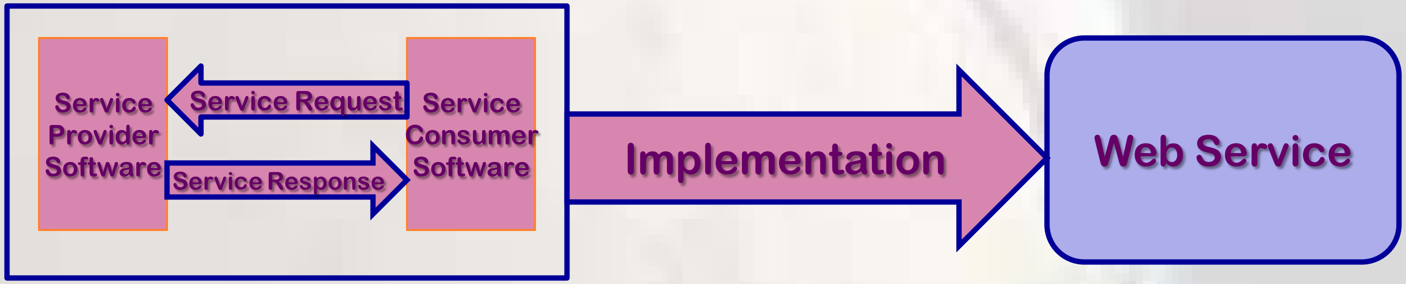
- Introduction
  - SOA concepts and terms.
  - SOA key principles and basic role.
- Middleware
- **Web services**
- Palestinian e-Government Architecture
- Summary

What is a web service?

- Have an application!
- Want to convert your application into a web application?
- Web services is the answer!
- Web application then, can publish its function to the rest of the world!
- Web Services are *published*, *found*, and *used* through the Web
- XML is the basis for Web services

SOA and Web services

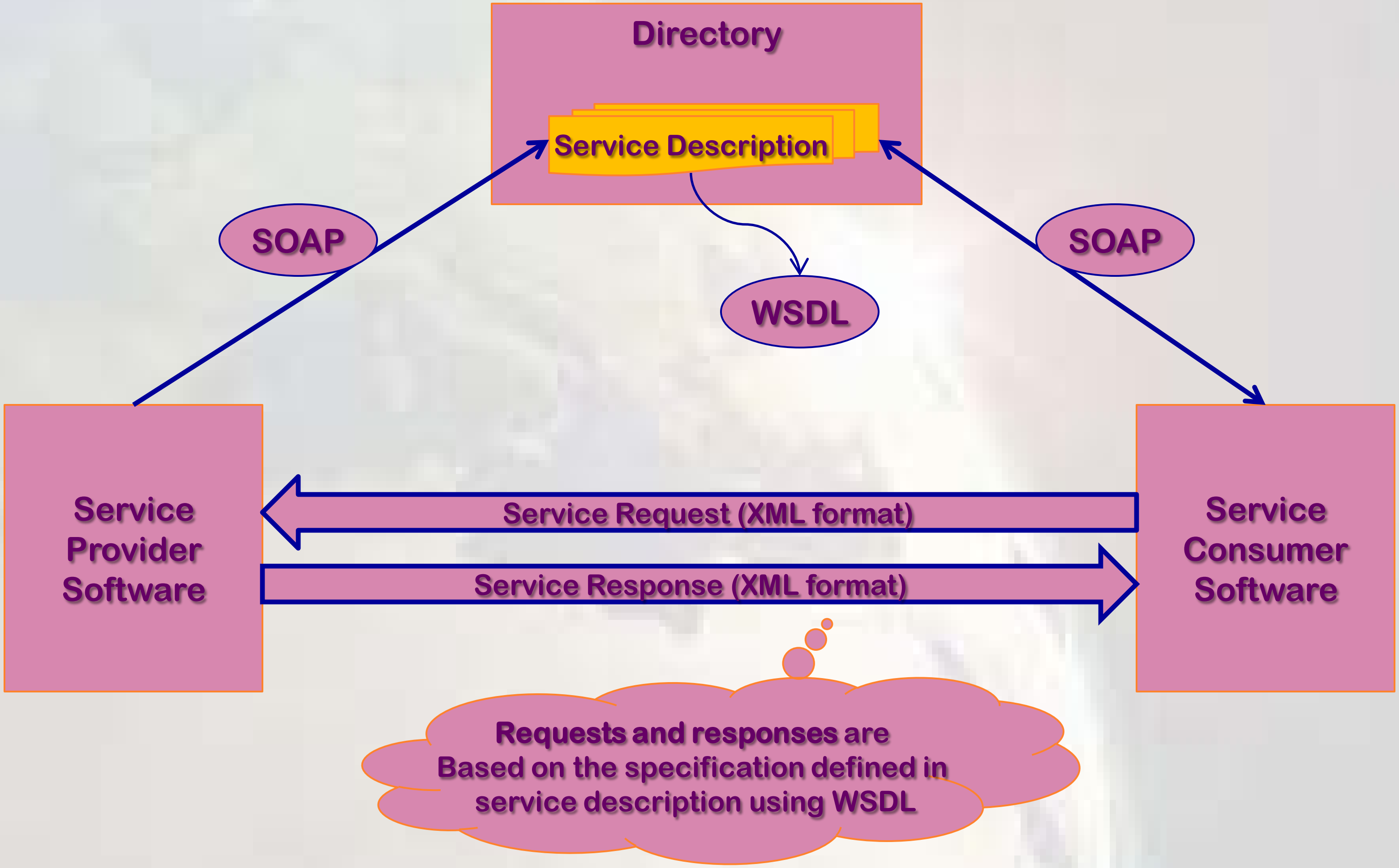
- The term Web service describes a standardized way of integrating Web-based applications [4];
  - This can be done using the XML, SOAP, WSDL and UDDI open standards over an Internet protocol backbone.
  - SOAP (Simple Object Access Protocol)
  - UDDI (Universal Description, Discovery and Integration)
  - WSDL (Web Services Description Language)
- Web service is an implementation of service oriented architecture



SOA and Web services, cont.

- SOAP, WSDL and UDDI support the interaction between the WS requester and the WS provider
- Provider can publish a WSDL description of its Web service, and the requester accesses the description using a UDDI
- Requester can request the execution of the provider's service by sending a SOAP message to it
- WS standards add a layer of abstraction which is open to integrate with new and existing environments.

Basic Web services architecture


Directory
Service Description
SOAP
WSDL
SOAP
Service Provider Software
Service Request (XML format)
Service Response (XML format)
Service Consumer Software
Requests and responses are Based on the specification defined in service description using WSDL

Web services architecture consists of specifications, source [4].

Web Services Technology

- XML: enabling the definition, transmission, validation, and interpretation of data between applications and between organizations.
- SOAP: lightweight XML-based (W3C standard) messaging protocol used to encode the information in Web service request and response messages before sending them over the Internet.
  - SOAP is platform independent
  - SOAP is language independent
  - SOAP allows you to get around firewalls

Web Services Technology, cont.

- WSDL: an XML-based language (W3C standard)
  - Used to locate and describe a Web service's capabilities as collections of communication endpoints capable of exchanging messages.
- UDDI: a Web-based distributed directory
  - UDDI enables businesses to list themselves on the Internet and discover each other.
  - UDDI is a directory for storing information about web services
  - UDDI is a directory of web service interfaces described by WSDL
  - UDDI communicates via SOAP

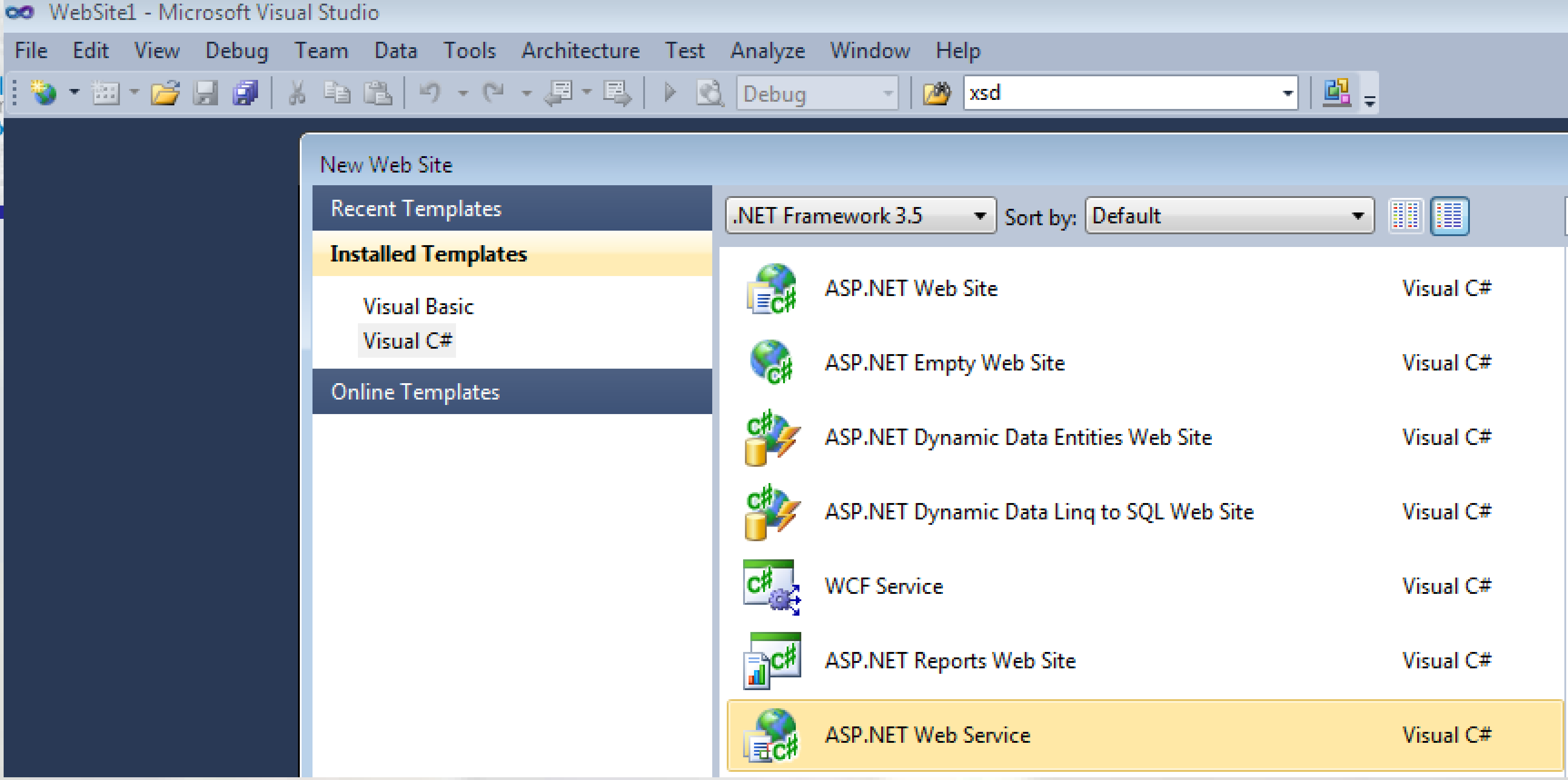
Why using Web services?

- **Reusable application-components**
  - Application-components like language translation, currency conversion, etc... can be reused.
  - A widely used approach to simplify software reuse is the component-based software engineering.
- **Connect existing software**
  - Interoperability problem is solved!
  - Existing applications are developed in different technologies like: ms.net, PHP, Java, etc…
  - With Web services different applications and different platforms can exchange data (interoperate).

Web Service example:

- A simple ASP.NET Web Service that converts the temperature from Fahrenheit to Celsius, and vice versa.
- Go to the following link: http://www.w3schools.com/webservices/ws_example.asp
- To use the Web Service Example, Go to the following link: http://www.w3schools.com/webservices/ws_use.asp

- Now we can Implement our first web service example using MS Visual Studio 2010!



Debug

xsd

New Web Site

.NET Framework 3.5 Sort by: Default

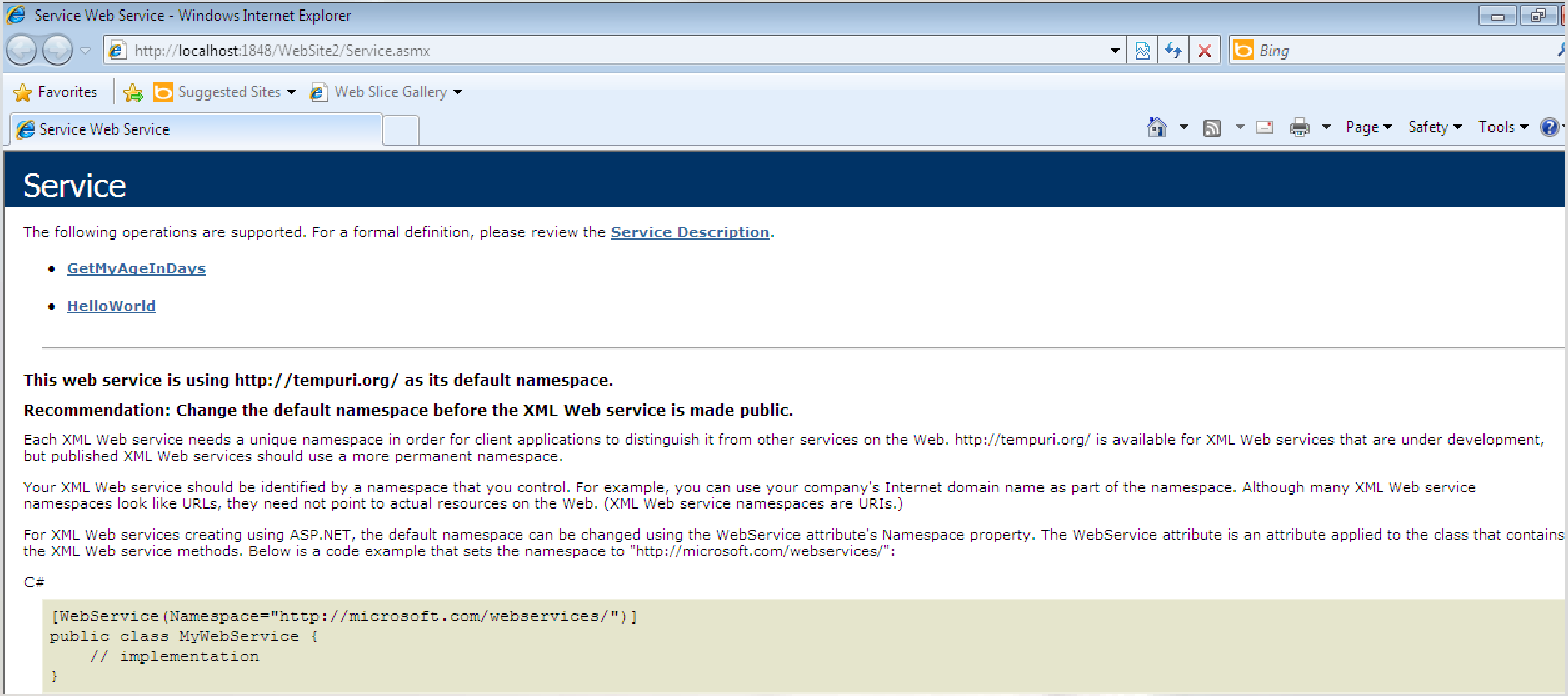
ASP.NET Web Site	Visual C#
ASP.NET Empty Web Site	Visual C#
ASP.NET Dynamic Data Entities Web Site	Visual C#
ASP.NET Dynamic Data Linq to SQL Web Site	Visual C#
WCF Service	Visual C#
ASP.NET Reports Web Site	Visual C#
ASP.NET Web Service	Visual C#

- Start new web site,
- Select from visual C# with .NET Framework 3.5, the template called ASP.NET Web Service.
- Put the name you want! (name your service)

- Edit the C# code “service.cs”; to add the WebMethod, which will achieve the required function:

```
[WebMethod]
public int GetMyAgeInDays(int day, int month, int year){
     DateTime dt = new DateTime(year, month, day);
     int days = DateTime.Now.Subtract(dt).Days;
     return days;
 }
```

- Save, build the Web site, and from the solution explorer, right click the “page”.asmx and select as a default page.
- Run the website! Press “Crtl + F5”
- You will get the output in the next slide!


Service Web Service - Windows Internet Explorer
http://localhost:1848/WebSite2/Service.asmx
Bing
Favorites
Suggested Sites
Web Slice Gallery
Service Web Service
Page
Safety
Tools

Service

The following operations are supported. For a formal definition, please review the **Service Description**.

- **GetMyAgeInDays**
- **HelloWorld**

This web service is using http://tempuri.org/ as its default namespace.

Recommendation: Change the default namespace before the XML Web service is made public.

Each XML Web service needs a unique namespace in order for client applications to distinguish it from other services on the Web. http://tempuri.org/ is available for XML Web services that are under development, but published XML Web services should use a more permanent namespace.

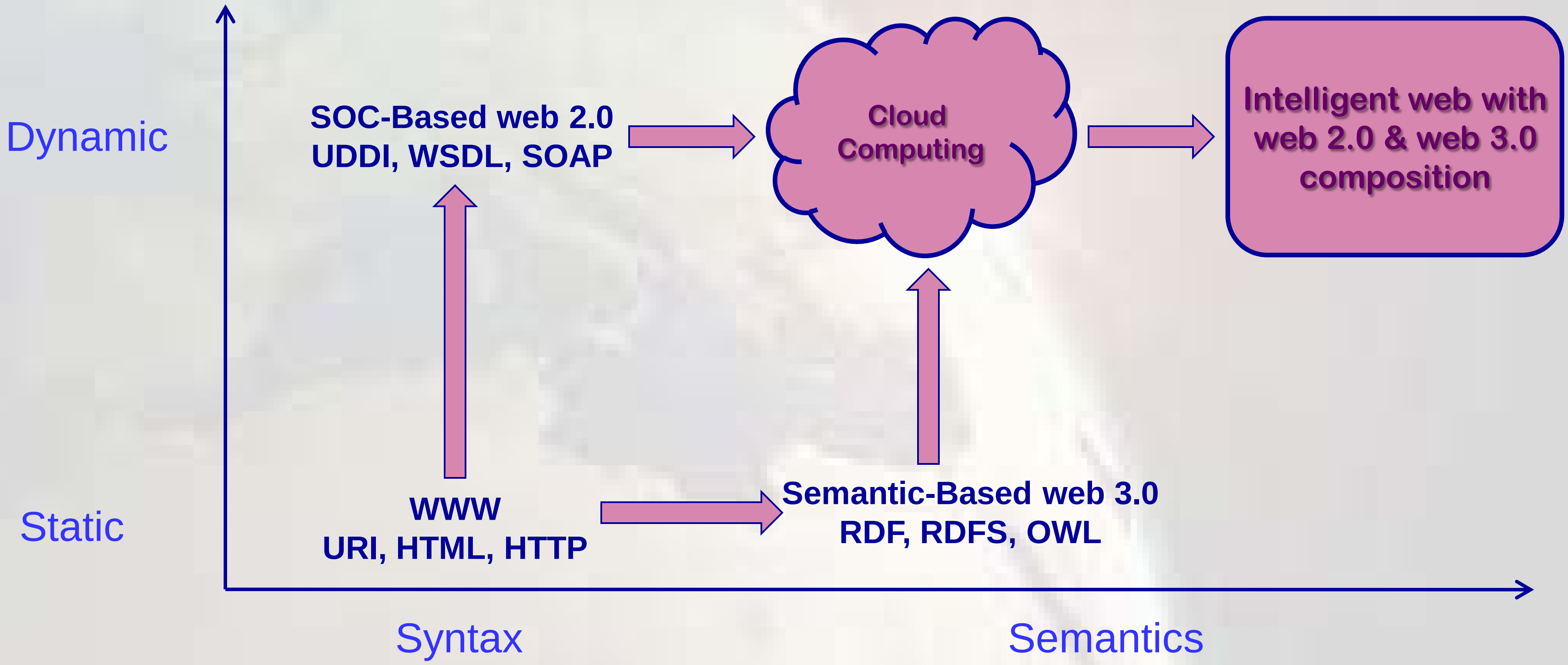
Your XML Web service should be identified by a namespace that you control. For example, you can use your company's Internet domain name as part of the namespace. Although many XML Web service namespaces look like URLs, they need not point to actual resources on the Web. (XML Web service namespaces are URIs.)

For XML Web services creating using ASP.NET, the default namespace can be changed using the WebService attribute's Namespace property. The WebService attribute is an attribute applied to the class that contains the XML Web service methods. Below is a code example that sets the namespace to "http://microsoft.com/webservices/":

C#

```
[WebService(Namespace="http://microsoft.com/webservices/")]
public class MyWebService {
    // implementation
}
```

Internet Based Computing



Dynamic

Static

SOC-Based web 2.0
UDDI, WSDL, SOAP

Cloud Computing

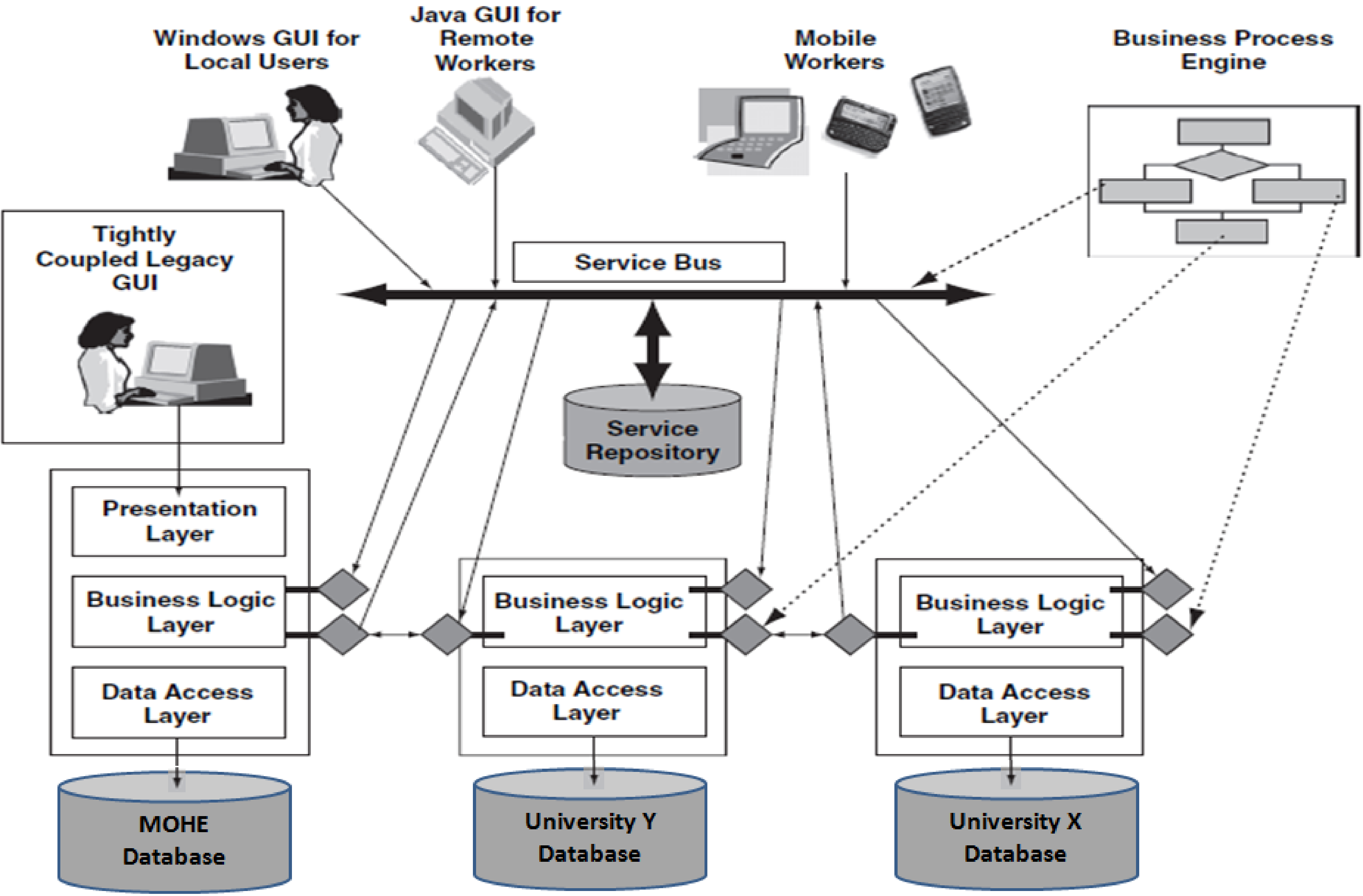
Intelligent web with web 2.0 & web 3.0 composition

WWW
URI, HTML, HTTP

Semantic-Based web 3.0
RDF, RDFS, OWL

Syntax

Semantics



Designing for service-oriented integration, source[4], with minor adjustment

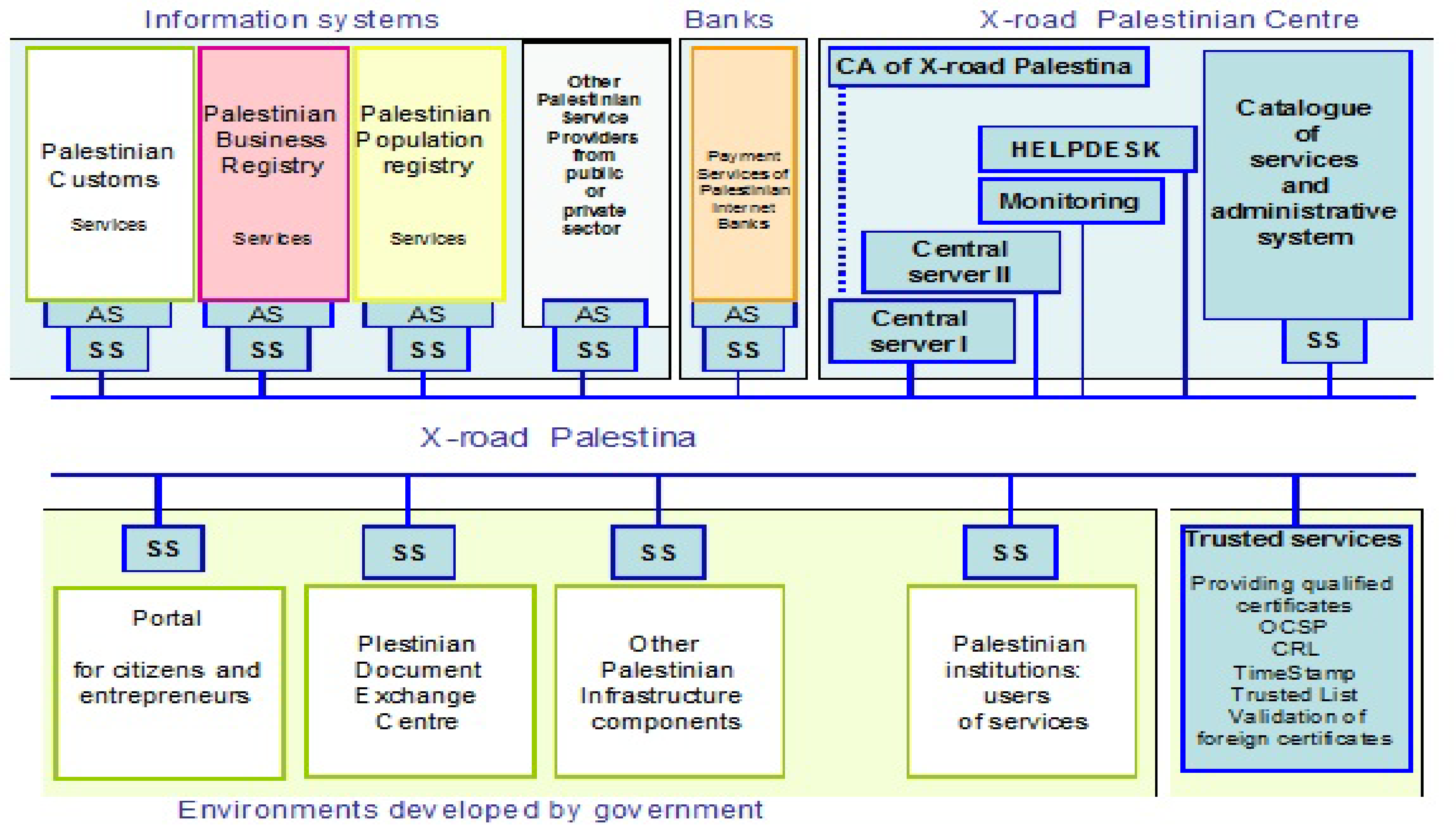
Session Outlines

- Introduction
  - SOA concepts and terms.
  - SOA key principles and basic role.
- Middleware
- Web services
- **Palestinian e-Government Architecture**
- Summary

The Palestinian e-Government Architecture

- The Palestinian e-Government architecture has been developed in cooperation with the Estonian government [8].
    - The architecture connects all ministries together through a *government service bus*, called “x-road Palestine”.
    - The service bus, with the other components, represents standard *service oriented architecture* with an emphasis on the provision of secure services.
    - Not yet implemented, but there is a general consensus about it among most governmental bodies

The Palestinian e-Government Architecture



Information systems

Banks

X-road Palestinian Centre

Palestinian Customs

Services

Palestinian Business Registry

Services

Palestinian Population registry

Services

Other Palestinian Service Providers from public or private sector

Payment Services of Palestinian Internet Banks

AS

SS

CA of X-road Palestina

HELPDESK

Monitoring

Central server II

Central server I

Catalogue of services and administrative system

SS

X-road Palestina

SS

Portal

for citizens and entrepreneurs

Plestinian Document Exchange Centre

Other Palestinian Infrastructure components

Palestinian institutions: users of services

Trusted services

Providing qualified certificates

OCSP

CRL

TimeStamp

Trusted List

Validation of foreign certificates

Environments developed by government

Source, [8]

Session Outlines

- Introduction
  - SOA concepts and terms.
  - SOA key principles and basic role.
- Middleware
- Web services
- Palestinian e-Government Architecture
- **Summary**

Summary

In this session we have introduced SOA where the concept of service oriented architecture has been explained, then the fundamentals of middleware were discussed.

Finally, we have described the most important characteristics of applications (web services) based on SOA.

References

1. Extracted from http://en.wikipedia.org/wiki/Middleware
2. Andrew S. Tanenbaum and Maarten van Steen Distributed Systems: Principles and Paradigms, 2nd Edition, Publisher: Pearson Education Inc., Copyrights: 2007
3. E.Newcomer and G.Lomow, Understanding SOA with Web Services, Addison Wesley Professional, ISBN: 0-321-18086-0, 2004.
4. Effective Performance Management for Service-Oriented Applications, WHITE PAPER, Copyright © 2009 Precise Software Solutions, Inc.
5. Extracted from: http://en.wikipedia.org/wiki/Service-oriented_architecture
6. Extracted from IBM education assistant: http://publib.boulder.ibm.com/infocenter/ieduasst, 9/2011.
7. Yuli Vasiliev , SOA and WS-BPEL, Copyright © 2007 Packt Publishing
8. Ministry of Telecommunication and Information Technology: e-Government Program, Palestine X-Road: Palestinian e-Government Architecture. Ramallah, Palestine. January 2011.

Thanks

Mohammed Aldasht