

Modulo II WebServices

Prof. Ismael H F Santos

Bibliografia

Ementa

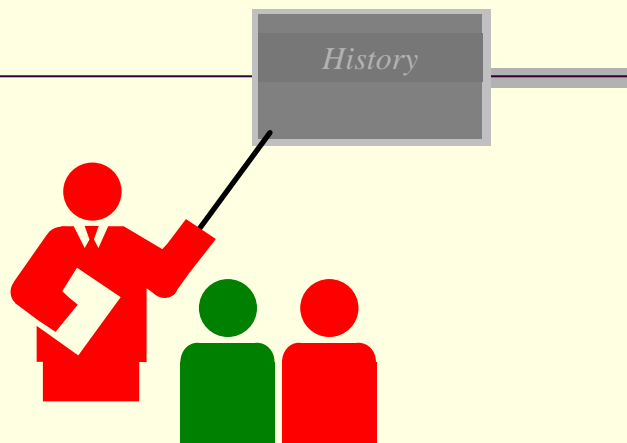
- History
- Introduction to WebServices
- XML for WebSrvices
 - XML
 - XML Schema
- Architecture
- SOAP
- WSDL
- UDDI
- JEE & WebServices
- Why to use Web Services

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

3

SOA

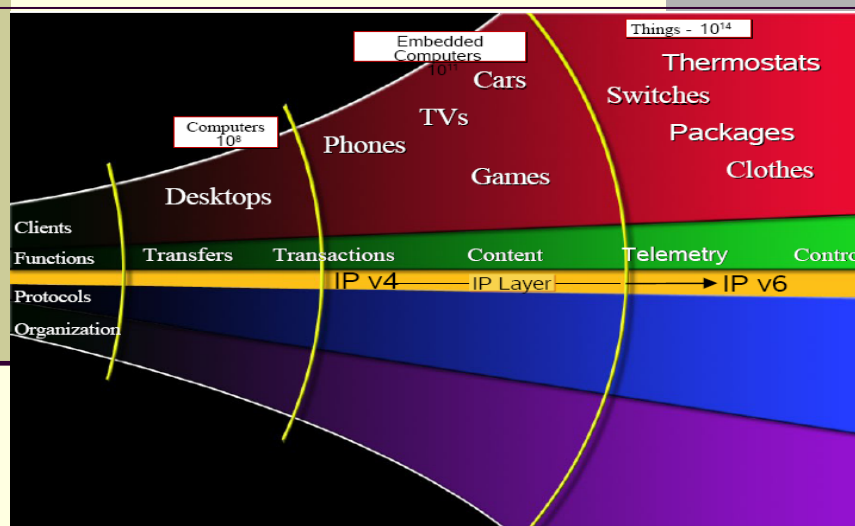


April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

4

Evolution of Network Computing (1)

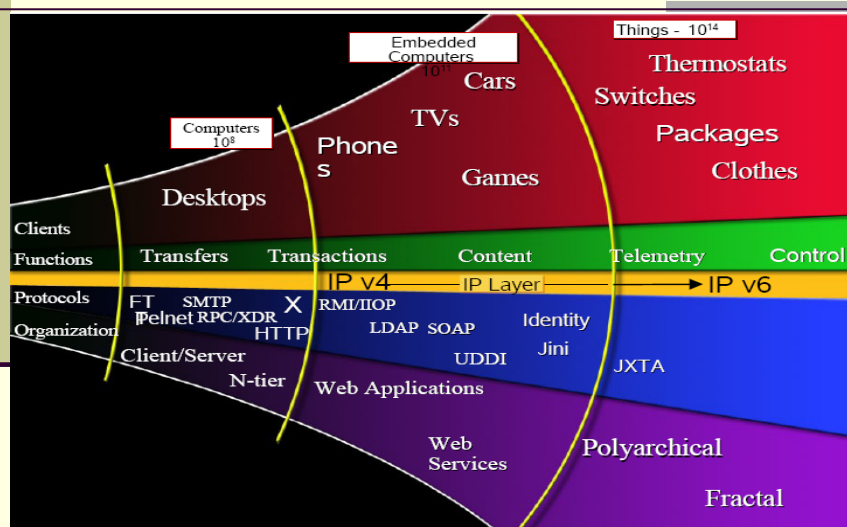


April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

5

Evolution of Network Computing (2)

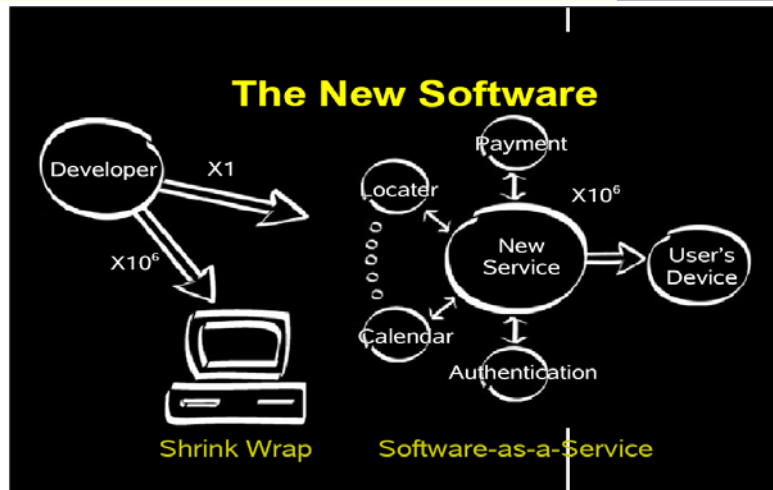


April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

6

The New software



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

7

Platform Evolution

Platform Evolution

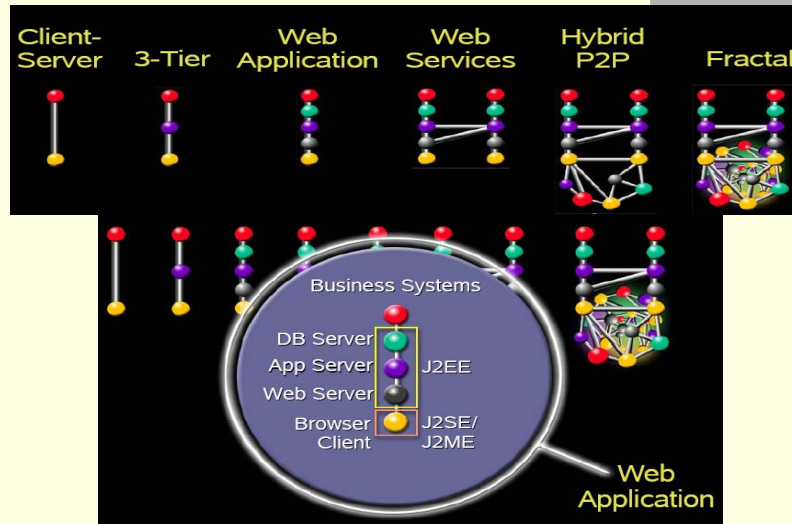
Catch Phrase	The Network Is the Computer	Objects	Legacy to the Web	The Computer Is the Network	Network of Embedded Things	Network of Things
Scale	100s	1,000s	1,000,000s	10,000,000s	100,000,000s	100,000,000s
When/Peak	1984/1987	1990/1993	1996/1999	2001/2003	1998/2004	2004/2007
Leaf Protocol(s)	X	X	+HTTP (+JVM)	+XML Portal	+RM	Unknown
Directory(s)	NS, NS+	+CDS	+LDAP(*)	+UDDI	+Jini	+?
Session	RPC, XDR	+CORBA	+CORBA, RM	+SOAP, XML	+RM/Jini	+?
Schematic						

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

8

Communication Patterns

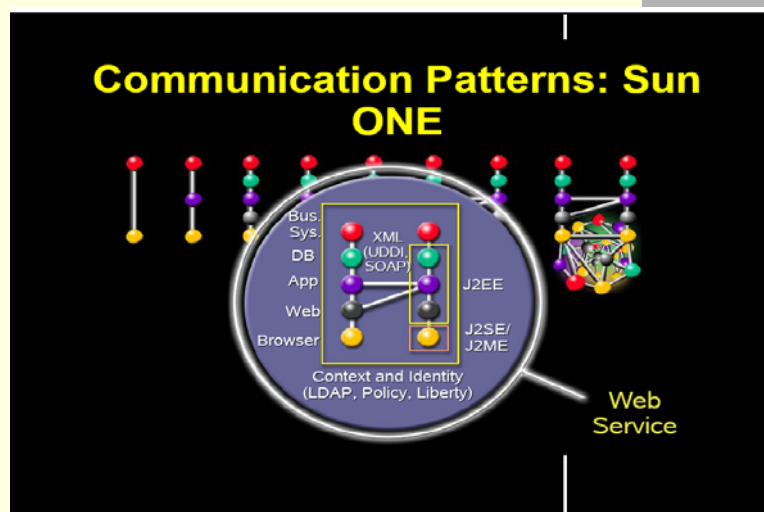


April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

9

Communication Patterns - WebServices



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

10

SOA

Introduction



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

11

Why to use more than one computer?

- Distributed resources
 - access to shared data
 - access to shared programs
 - access to CPU (e.g. many desktop PCs together), to memory, to special devices (e.g. printer)
- Complete independence on the internal implementation

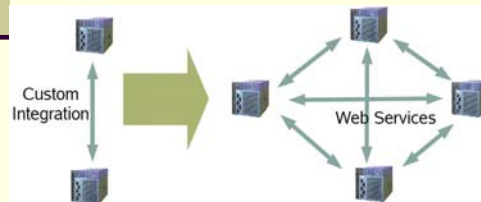
April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

12

Three Laws of Computing

- **Moore's Law**
 - Computing power doubles every 18 months
- **Gilder's Law**
 - Network bandwidth capacity doubles every 12 months
- **Metcalfe's Law (Net Effect)**
 - Value of network increases exponentially as number of participants increases



**Impact on Integration:
Trigger the Network
Effect**

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

13

Distributed architecture

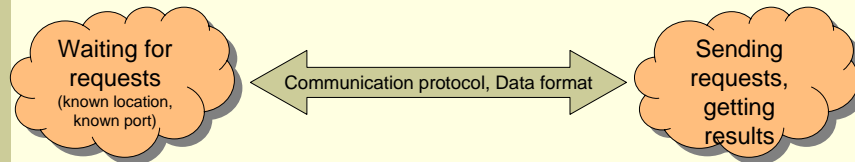
- gives
 - access to distributed resources
 - development encapsulation
 - maintainability, re-usability, legacy-awareness
 - implementation independence
- requires
 - adding a communication layer between parts
 - synchronization of efforts
 - including such nasty things as distributed garbage collection

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

14

Distributed architecture



- **Basic questions are:**

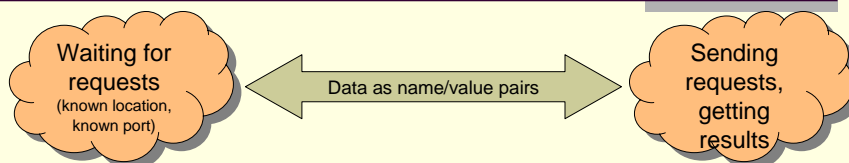
- What kind of protocol to use, and what data to transmit
- What to do with requests on the server side

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

15

Traditional CGI-based approach



- **cgi-bin scripts:**

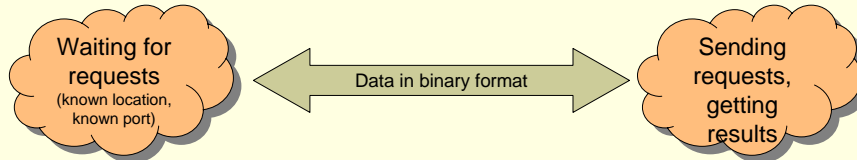
- Data transmitted as name-value pairs (HTML forms)
- Transport over (state-less) HTTP protocol
- no standards for keeping user sessions (state-fullness)
- server side: a script is called

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

16

CORBA-based approach



■ RMI/Java & CORBA:

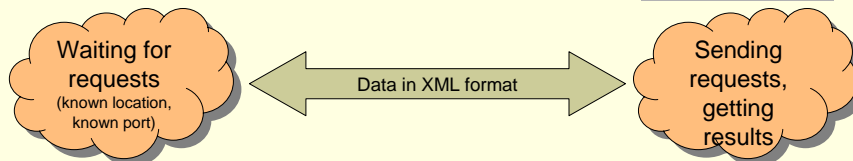
- Data transmitted as objects (at least it looks like that)
- Transport (usually) over well standardised IIOP protocol
- user sessions (state-fullness) very inter-operable
- server side: an RPC call is made

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

17

SOAP-based communication



■ SOAP:

- Data in a well-defined XML format
- Transport over various protocols
 - HTTP, SMTP are the most used, perhaps because they are firewall-friendly
- server side: either an RPC call or a message delivered

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

18

Serialization

```
class PurchaseOrder
{
String item =
"socks";
int amount = 1;
}
```

Serializer



```
<PurchaseOrder>
<item
type="xsd:string">
    socks
</item>
<amount
type="xsd:int">
    1
</amount>
</PurchaseOrder>
```

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

19

Web services

♦ Web Service Definition:

- A **web service** is simply a remote procedure call (RPC) over the web that uses XML as the calling format



- Not unlike some of the other RPC mechanisms...



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

20

SOA

Introduction



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

21

Abstract

- **Web Services** is a technology applicable for computationally distributed problems, including access to large databases
 - What other technologies were/are available and how they compare with Web Services?
- **The main buzzwords:**
 - Enterprise Integration Application & Standardization & Access by programs

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

22

Web services

- A collection of XML-based technologies developed by the e-business community to address issues of:
 - service discovery
 - interoperable data exchange and/or application invocation
 - service compositions (workflow, business processes)
- Major developers include:
 - Apache, IBM, HP, SUN & Microsoft (.NET)
 - <http://www.webservices.org/>

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

23

Web Services Definition by W3C

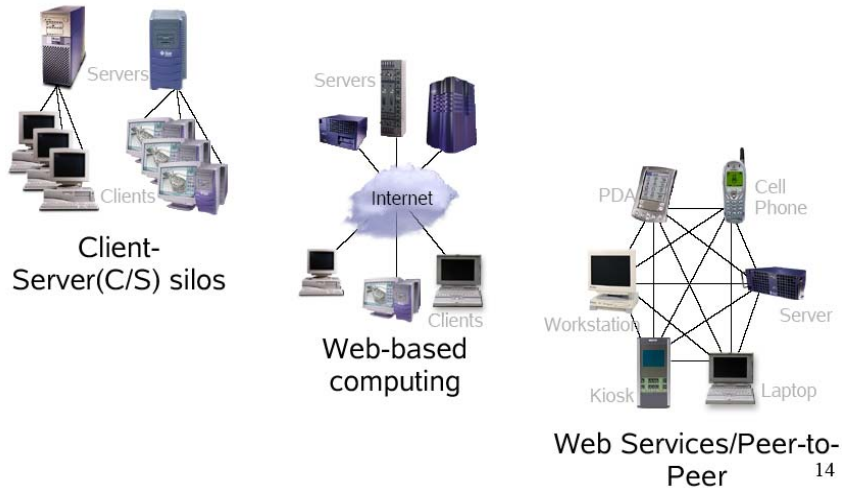
- A **Web Service** is a software application
 - identified by a **URI**,
 - whose interfaces and binding are capable of being **defined**, **described** and **discovered** by XML artifacts and
 - supports direct **interactions** with other software applications
 - using **XML based messages** via **internet-based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

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

24

Distributed Computing Evolution



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

25

Traditional C/S vs. Web Services

Traditional C/S

- Within enterprise
- Tied to a set of programming languages
- Procedural
- Usually bound to a particular transport
- Tightly-coupled
- Efficient processing (space/time)

Web Service

- Between enterprises
- Program language independent
- Message-driven
- Easily bound to different transports
- Loosely-coupled
- Relatively not efficient processing

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

26

Web Application vs. Web Services

Web Application

- User-to-program interaction
- Static integration of components
- Monolithic service

Web Service

- Program-to-program interaction
- Possibility of dynamic integration of components (in the future)
- Possibility of service aggregation (in the future)

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

27

Characteristics of Web Services

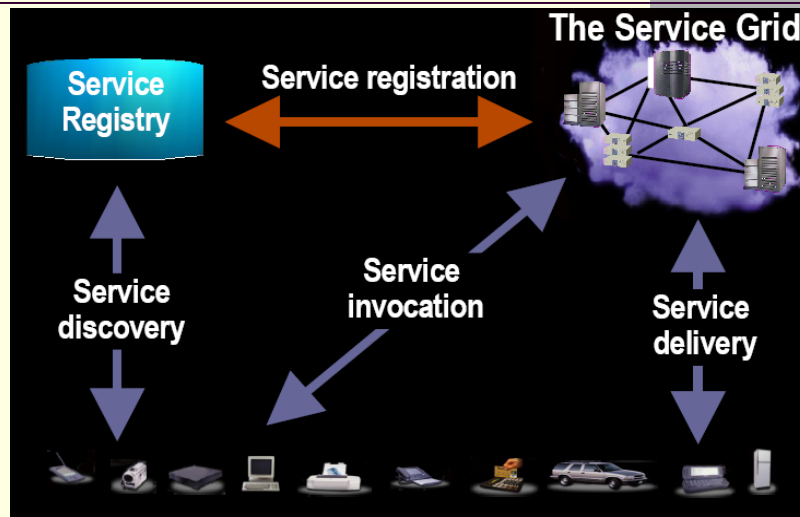
- XML based everywhere
- **Message-based**
- **Programming language independent**
- Could be dynamically located
- Could be dynamically assembled or aggregated
- Accessed over the internet
- Loosely coupled
- Based on industry standards

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

28

Web Services

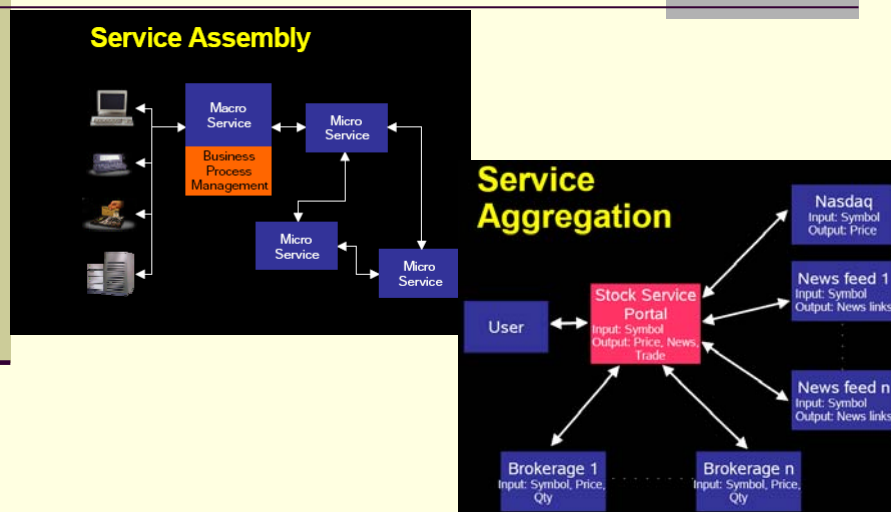


April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

29

Service Assembly & Agreggation

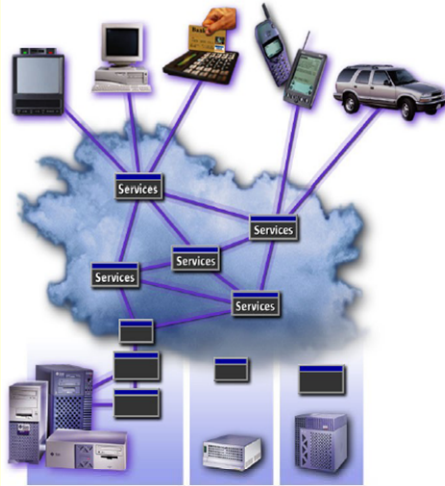


April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

30

Why Web Services?



Web Services:

- Are platform neutral
- Are accessible in a standard way
- Are accessible in an **interoperable** way
- Use simple and ubiquitous plumbing
- Are relatively cheap
- Simplify **enterprise integration**

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

31

Why Web Services?

- ◆ Web services are old technologies wearing a funny new hat
 - You can get XML/HTTP today with sockets
 - You can describe services with IDL or interfaces
 - You can register services via JNDI or Active Directory
- ◆ So why will web services succeed?
 - It is simple
 - People agree on it
 - All major vendors are pushing for web services
 - People are excited about them
 - People are beginning to use them in projects

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

32

Why Web Services?

- ◆ Paradigm benefits
 - Loose application coupling
 - Independent application evolution
- ◆ Interoperability
 - 'Component wars' and 'Languages wars' don't matter
 - Standardization of integration technologies
- ◆ B2B
 - Leverages Internet - Cheaper than VANs
 - Extensible
- ◆ EAI
 - Non-intrusive integration

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

33

Why Web Services?

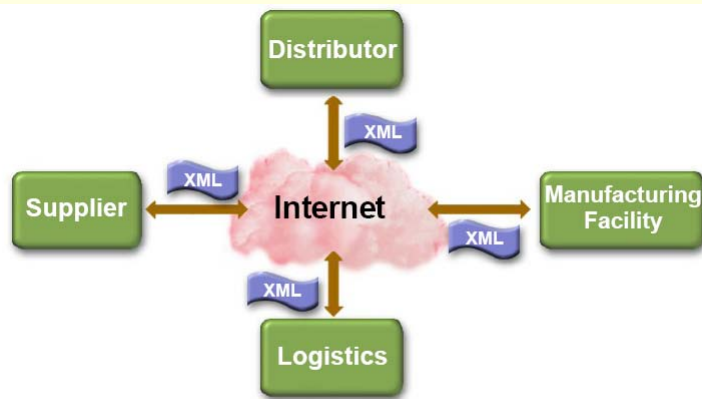
- **Interoperable** – Connect across heterogeneous networks using ubiquitous web-based standards
- **Economical** – Recycle components, no installation and tight integration of software
- **Automatic** – No human intervention required even for highly complex transactions
- **Accessible** – Legacy assets & internal apps are exposed and accessible on the web
- **Available** – Services on any device, anywhere, anytime
- **Scalable** – No limits on scope of applications and amount of heterogeneous applications

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

34

Web Services Usage Example



“Growing need for a **standard lightweight infrastructure** for data exchange in e-business applications.”

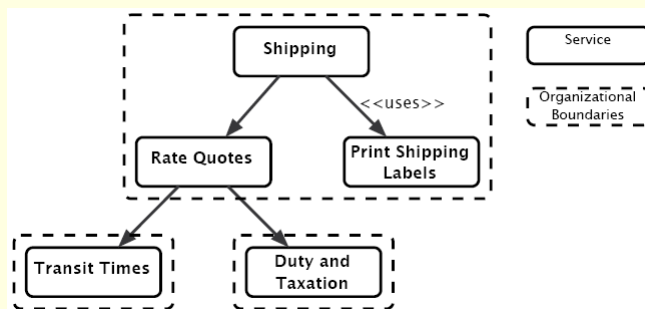
April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

35

Web Services generations

- The **first generation** of web services were “simple”, in the sense of non-composite, and closed (over existing, trusted relationships)
- The **second generation** is complex, and aggregated from web services provided by third parties (hence, open)

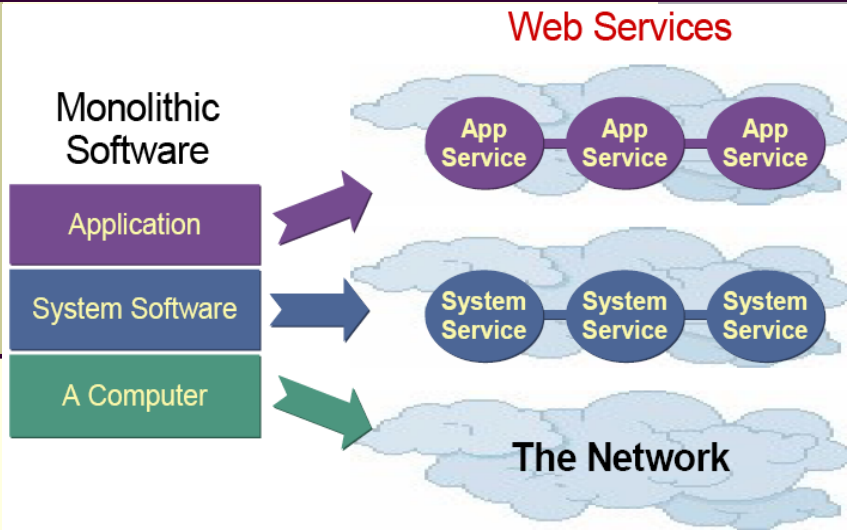


April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

36

Impact of Web Services on Software: "Application Dis-Integration"



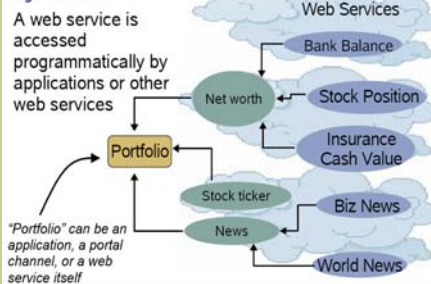
April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

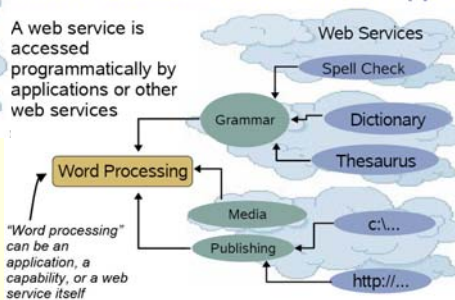
37

Macro & Micro web services

Macro web services – Virtual Systems



Micro web services – Virtual Apps



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

38

Myth: Web Services is a New Concept

- Web services is distributed computing all over again – only now it is based on the web

Concept	Distributed Computing ala CORBA / Java	Basic Web Services
Interface Description	CORBA IDL, Java interface	WSDL
RPC support	ORBs, Idl2java compilers, rmic	SOAP, compilers for WSDL
Service Registry	CORBA naming service, JNDI	UDDI
Messaging support	CORBA Event/Notification service, JMS	?
Transaction support	CORBA Transaction service, JTS	?
Security support	CORBA Security service, Java security	?

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

39

Other Popular Myths Surrounding Web Services

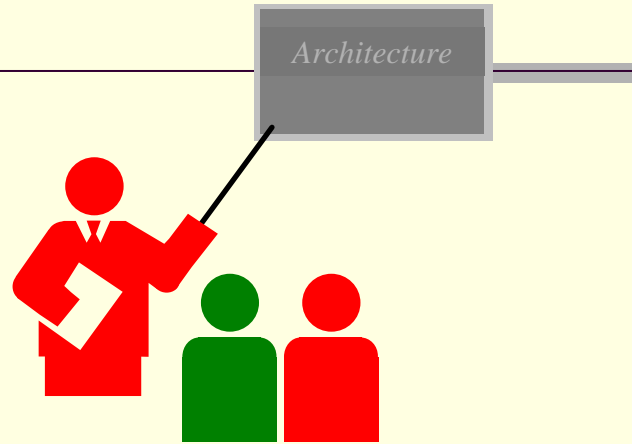
- Web services require only SOAP, WSDL, UDDI: **We need more high-level semantics**
- Web services are based on the RPC paradigm: **Document-driven model would be more popular communication model**
- Web services must be based on HTTP: **Other transports such as SMTP can be also used**

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

40

SOA



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

41

Web Services Architecture

- Web Services involve three major roles
 - Service Provider
 - Service Registry
 - Service Consumer

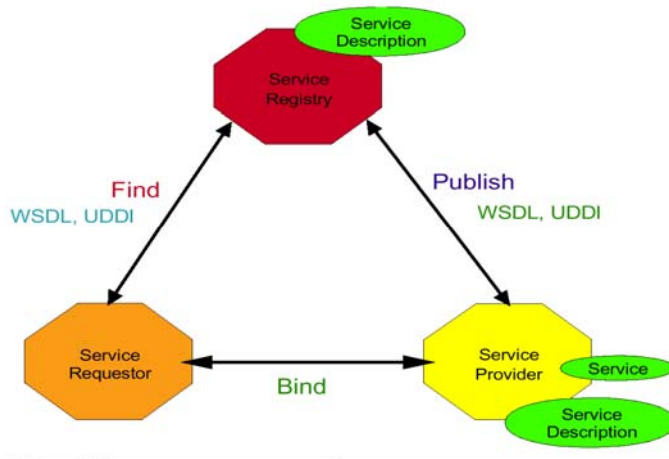
- Three major operations surround web services
 - Publishing – making a service available
 - Finding – locating web services
 - Binding – using web services

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

42

Web Services Architecture



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

43

Making a service available (1)

- In order for someone to use your service they have to know about it.
- To allow users to discover a service it is published to a registry (UDDI).
- To allow users to interact with a service you must publish a description of it's interface (methods & arguments). This is done using WSDL.

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

44

Making a service available (2)

- Once you have published a description of your service you must have a host set up to serve it.
- A **web server** is often used to deliver services (although custom application – application communication is also possible).
- This is functionality which has to be added to the web server. In the case of the apache web server a 'container' application (Tomcat) can be used to make the application (servlet) available to apache (deploying).

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

45

The old transfer protocols are still there.

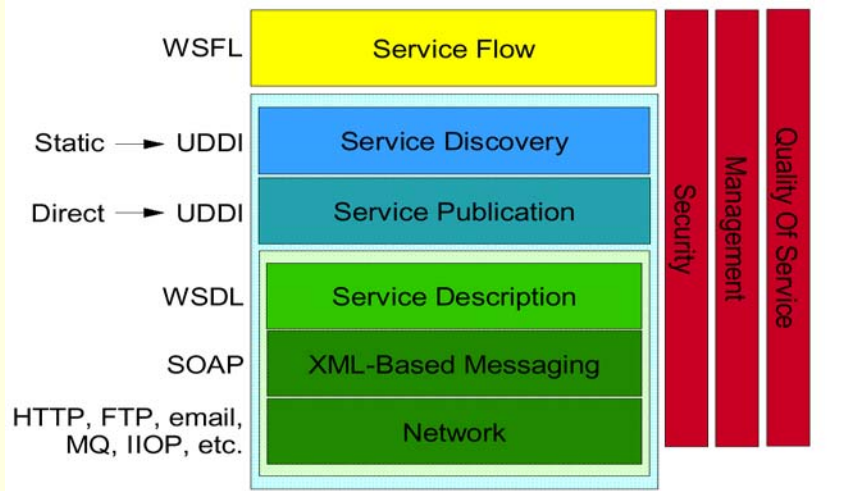
- Like the grid architecture web services is layered on top of existing, mature transfer protocols.
- HTTP, SMTP are still used over TCP/IP to pass the messages.
- Web services, like grids, can be seen as a functionality enhancement to the existing technologies.

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

46

Web Services Stack



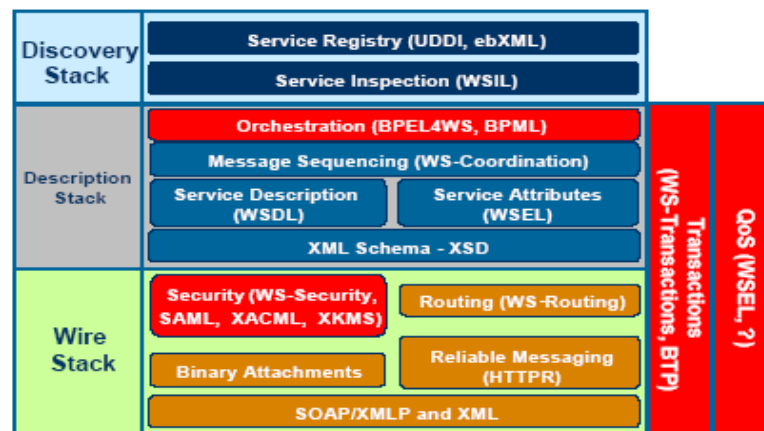
April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

47

Web Services Architecture (WSA, IBM)

Web Services – Mind the Gaps

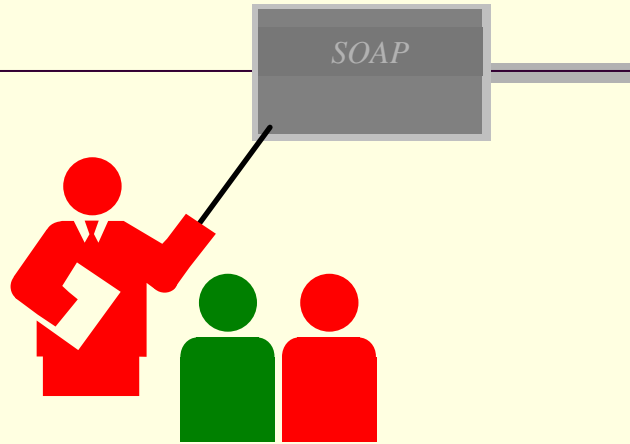


April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

48

SOA



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

49

SOAP

- **Simple Object Access Protocol**
 - <http://www.w3c.org/TR/SOAP/>
- A lightweight protocol for exchange of information in a decentralised, distributed environment
- Wire protocol similar to: **IOP for CORBA** and **JRMP for RMI**
- Two different styles to use:
 - to encapsulate RPC calls using the extensibility and flexibility of XML – **XML-based RPC**
 - ...or to deliver a whole document without any method calls encapsulated – **XML is used for data encoding**

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

50

SOAP messages

- SOAP provides a standard 'envelope' within which a message can be delivered.
- SOAP is mechanism (protocol) for transferring information (messages) between applications which may be widely distributed.
- SOAP says nothing about the content of the message – the sender and the receiver must understand the message for themselves.
- SOAP is part of a communication stack.

What SOAP is Not

- Not a component model
 - So it will not replace objects and components, i.e. EJB, JavaBeans
- Not a programming language
 - So it will not replace Java
- Not a solution for all
 - So it will not replace other distributed computing technologies such as RMI

What does SOAP define ?

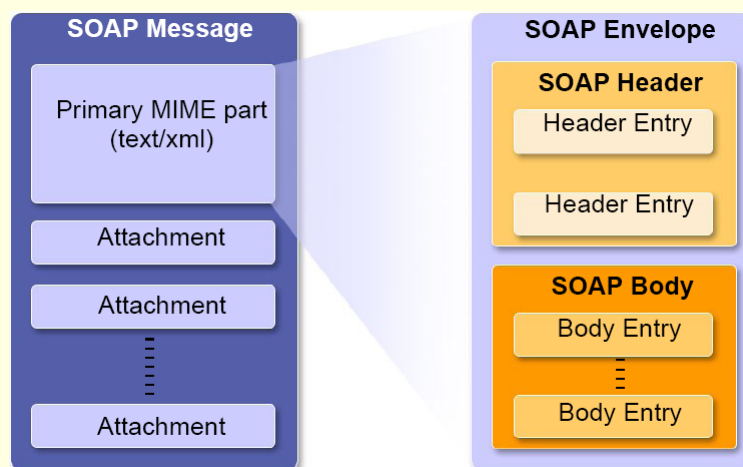
- Message Envelope
- Encoding Rules
- RPC Convention
- Binding with underlying protocols

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

53

SOAP Message Format



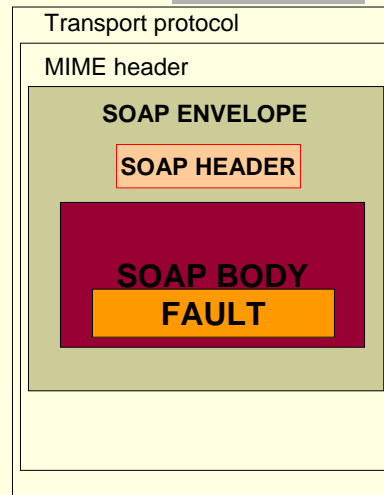
April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

54

SOAP Structure (1)

- Each SOAP message will have:
 - An **Envelope**
 - A **Header** (optional)
 - A **Body**
 - The **Body** may contain a **Fault** element



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

55

SOAP Structure (2)

- The **envelope** wraps the entire soap document
- The **header** contains allows additional information to be passed as well as the body of the document – e.g. authentication
- The **body element** contains the core of the SOAP document – this will contain either the RPC call or the XML message itself
- The **fault information** will contain any exception information

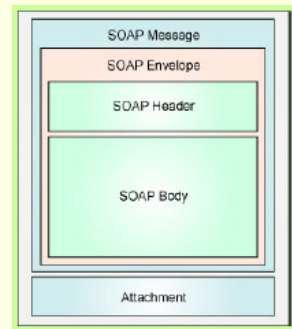
April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

56

SOAP Structure (3)

```
<s:Envelope xmlns:s="URN">
  <s:header>
    <s:transaction xmlns:m="soap-transaction">
      <m:transactionID>
        1234
      </m:transactionID >
    </s:transaction>
  </s:header>
```



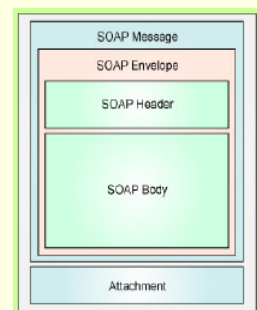
April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

57

SOAP Structure (4)

```
<s:Body>
  <n:purchaseOrder xmlns:n="URN">
    <n:item>socks</n:item>
    <n:amount>1</n:amount>
  </n:purchaseOrder>
</s:Body>
</s:Envelope>
```



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

58

SOAP Encoding

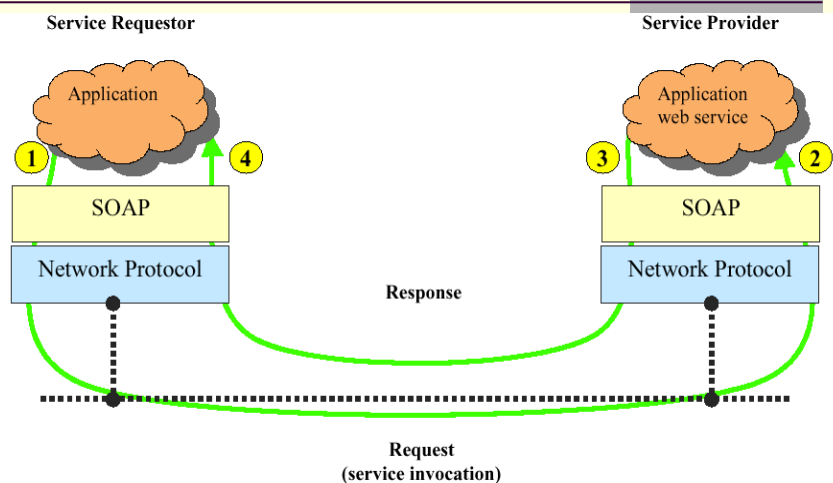
- Rules of expressing application-defined data types in XML
- Based on W3C XML Schema
- Simple values
 - Built-in types from XML Schema, Part 2 (simple types, enumerations, arrays of bytes)
- Compound values
 - Structs, arrays, complex types

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

59

XML Messaging Using SOAP



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

60

Request:
setHelloMessage

April 05 Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br 61

Anatomy of a SOAP message

```

<?xml version='1.0' encoding='UTF-8'?>

<SOAP-ENV:Envelope
xmlns:SOAP_ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3c.org/1999/XMLSchema-instance"
xmlns:xsd="http://www.w3c.org/1999/XMLSchema">

<SOAP-ENV:Header>

</SOAP-ENV:Header>

<SOAP_ENV:Body>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

```

April 05 Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br 62

SOAP protocol binding

```
SOAPAction = "urn:soaphttpclient-action-uri"
Host = localhost
Content-Type = text/xml; charset=utf-8
Content-Length = 701
```

```
<SOAP-ENV:Envelope
xmlns:SOAP_ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3c.org/1999/XMLSchema-instance"
  xmlns:xsd="http://www.w3c.org/1999/XMLSchema">
```

.....

```
</SOAP-ENV:Envelope>
```

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

63

SOAP RPC

- SOAP RPC messages contain XML that represents a method call or method response
- The SOAP XML will be converted into a method call on the server and the response will be encoded into SOAP XML to be returned to the client

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

64

SOAP Faults

- SOAP errors are handled using a specialised envelope known as a Fault Envelope
- A SOAP Fault is a special element which must appear as an immediate child of the body element
- `<faultcode>` and `<faultstring>` are required.

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

65

A SOAP fault

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
  xmlns:SOAP_ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3c.org/1999/XMLSchema-instance"
  xmlns:xsd="http://www.w3c.org/1999/XMLSchema">
  <SOAP_ENV:Body>

    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Server</faultcode>
      <faultstring>Test fault</faultstring>
      <faultactor>/soap/servlet/rpcrouter</faultactor>
      <detail>
        ..
      </detail>
    </SOAP-ENV:Fault>

  </SOAP_ENV:Body>
</SOAP-ENV:Envelope>
```

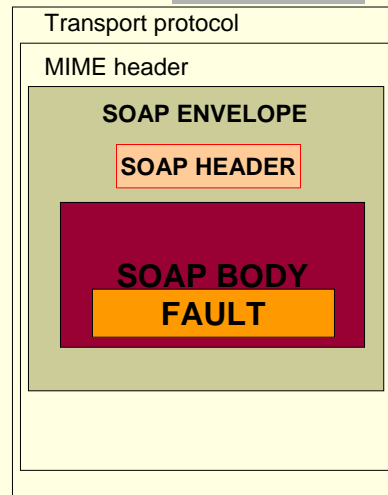
April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

66

SOAP Attachment

- Large quantities or binary data may not fit well into a XML SOAP message.
- In which case it can be sent 'out of band' by attaching it to a SOAP message
- *Analogy : email attachments.*



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

67

Attaching a file to a SOAP message

- To add a file to a SOAP message a tag is added within the body of the message.

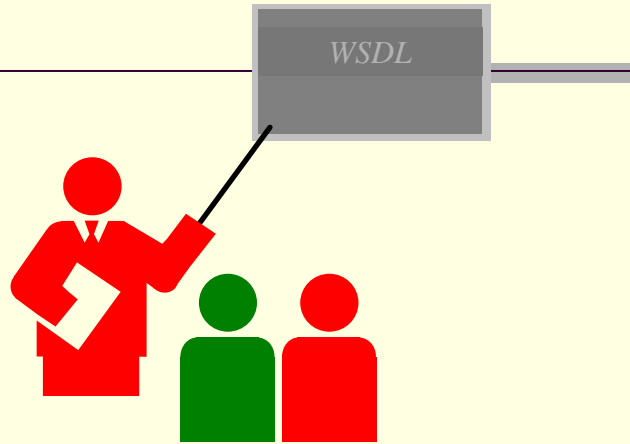
```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
xmlns:SOAP_ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3c.org/1999/XMLSchema-instance"
xmlns:xsd="http://www.w3c.org/1999/XMLSchema">
  <SOAP_ENV:Body>
    <attachment href="{URL}"/>
  </SOAP_ENV:Body>
</SOAP-ENV:Envelope>
```

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

68

SOA



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

69

Web Services Description Language - WSDL

- **Web Services Definition Language**
 - <http://www.w3.org/TR/wsdl/>
- An XML-based language for describing Web Services
 - what the service does (description)
 - how to use it (method signatures)
 - where to find the service
- It *does not* depend on the underlying protocol
- But: It is not much human-readable

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

70

Web Services Description Language - WSDL

- Provides functional description of network services:
 - IDL description
 - Protocol and deployment details
 - Platform independent description.
 - Extensible language.
- A short history:
 - WSDL v1.0, 9/2000
 - WSDL v1.1 submitted to W3C 3/2001.
 - A *de facto* industry standard.

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

71

Web Services Description Language - WSDL

- Describes the Web Service and defines the functions that are exposed in the Web Service
- Defines the XML grammar to be used in the messages
 - Uses the W3C Schema language
- Web service is described as
 - A set of **communication endpoints** (ports)
- Endpoint is made of two parts
 - **Abstract definitions of operations and messages**
 - **Concrete binding** to networking protocol (and corresponding endpoint address) and message format
- Why this separation?
 - Enhance **reusability** (as we will see in UDDI reference to WSDL document)

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

72

Why WSDL ?

- Enables **automation** of communication details between communicating partners
 - Machines can read WSDL
 - Machines can invoke a service defined in WSDL
- Discoverable through registry
- Arbitration
 - 3rd party can verify if communication conforms to WSDL

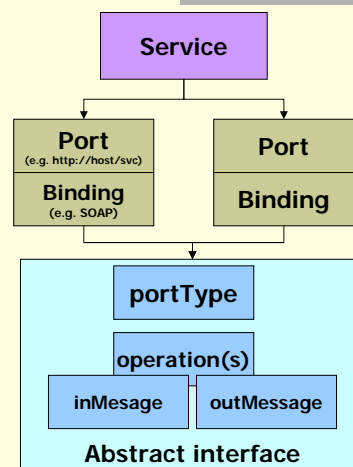
April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

73

WSDL Structure

- portType
 - Abstract definition of a service (set of operations)
- Multiple bindings per portType:
 - How to access it
 - SOAP, JMS, direct call
- Ports
 - Where to access it



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

74

Using WSDL

- As extended IDL: WSDL allows tools to generate compatible client and server stubs.
 - Tool support for top-down, bottom-up and “meet in the middle” development.
- Allows industries to define standardized service interfaces.
- Allows advertisement of service descriptions, enables dynamic discovery and binding of compatible services.
 - Used in conjunction with UDDI registry
- Provides a normalized description of heterogeneous applications.

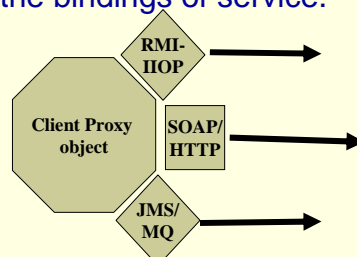
April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

75

Client invocation

- Single stub can invoke services over different bindings
 - Depends only on abstract interface.
- Are independent of binding (but pluggable).
 - Add new bindings without recompiling/redeploying stub
- Allows optimisations based on the bindings of service.
- Will support extended services models if described in WSDL

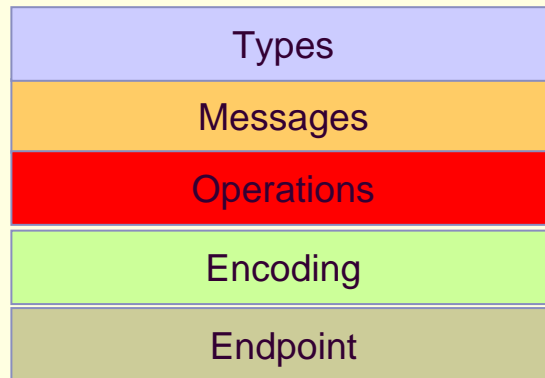


April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

76

Description – WSDL



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

77

Types, Messages

```
<types>  
<schema targetNamespace=" IMessageService.xsd"  
  xmlns=".../XMLSchema" xmlns:SOAPENC=".../soap/encoding"/>  
</types>
```

```
<message name="purchase">  
  <part name="item" type="xsd:string"/>  
  <part name="quantity" type="xsd:integer"/>  
</message>
```

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

78

Operations

```
<operation name="setMessage">
  <input name="setMessageRequest"
    message="tns:setMessageRequest"/>
  <output name="setMessageResponse"
    message="tns:setMessageResponse"/>
</operation>
```

Encoding

```
<soap:operation soapAction="" style="rpc"/>
  <input name="setMessage0Request">
    <soap:body use="encoded"
      namespace="MessageService"
      encodingStyle="../../../soap/encoding"/>
  </input>
```

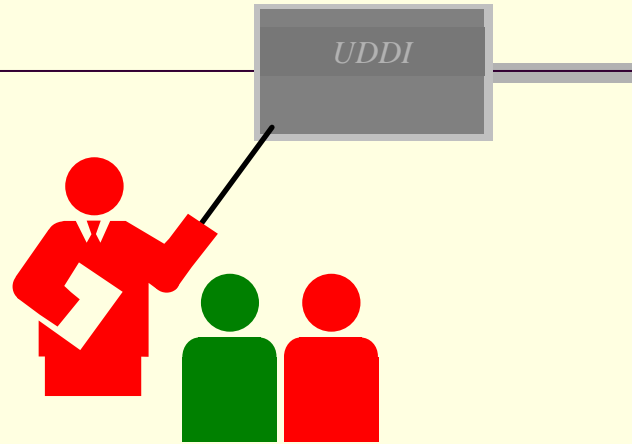

Endpoint

```
<service name="MessageService">
  <port name="MessageServicePort"
        binding="tns:MessageServiceBinding">
    <soap:address location="http://localhost:8080/setMessage/">
  </port>
</service>
```

Hello.wsdl

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="http://localhost:8080/axis/services/Hello"
  xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:impl="http://localhost:8080/axis/services/Hello-impl"
  xmlns:intf="http://localhost:8080/axis/services/Hello"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsdl:message
    name="setHelloMessageRequest">
    <wsdl:part name="in0" type="xsd:string"/>
  </wsdl:message>
  <wsdl:message
    name="getHelloMessageResponse">
    <wsdl:part name="return"
      type="xsd:string"/>
  </wsdl:message>
  <wsdl:message
    name="setHelloMessageResponse">
  </wsdl:message>
  <wsdl:message
    name="getHelloMessageRequest">
  </wsdl:message>
  <wsdl:portType name="HelloWorldService">
    <wsdl:operation name="getHelloMessage">
    <wsdl:input message="intf:getHelloMessageRequest"/>
    <wsdl:output
      message="intf:getHelloMessageResponse"/>
    </wsdl:operation>
    <wsdl:operation
      name="setHelloMessage" parameterOrder="in0">
    <wsdl:input
      message="intf:setHelloMessageRequest"/>
    <wsdl:output
      message="intf:setHelloMessageResponse"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="HelloSoapBinding" type="intf:HelloWorldService">
    <wsdlsoap:binding
      style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation
      name="getHelloMessage">
    <wsdlsoap:operation soapAction=""/>
    <wsdl:input
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        namespace="getHelloMessage" use="encoded"/>
    </wsdl:input>
    <wsdl:output
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        namespace="getHelloMessage" use="encoded"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation
    name="setHelloMessage">
    <wsdlsoap:operation
      soapAction=""/>
    <wsdl:input
      encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="setHelloMessage"
      use="encoded"/>
    </wsdl:input>
    <wsdl:output
      encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="setHelloMessage"
      use="encoded"/>
    </wsdl:output>
  </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="HelloWorldServiceService">
    <wsdl:port binding="intf:HelloSoapBinding" name="Hello">
    <wsdlsoap:address
      location="http://localhost:8080/axis/services/Hello/">
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

SOA



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

83

UDDI (and alternatives)

- **U**niversal **D**escription, **D**iscovery and **I**ntegration
 - <http://www.uddi.org>
- UDDI creates a platform-independent, open framework & registry for:
 - Describing services
 - Discovering businesses
 - Integrating business services
- The UDDI may be less used than predicted, especially on the Internet level
- A UDDI Server acts as a registry for Web Services and makes them searchable.

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

84

UDDI Overview

- UDDI defines the operation of a service registry:
 - Data structures for registering
 - **Businesses**
 - Technical specifications: tModel is a keyed reference to a technical specification.
 - Service and service endpoints: referencing the supported tModels
 - SOAP Access API
 - Rules for the operation of a global registry
 - “private” UDDI nodes are likely to appear, though.

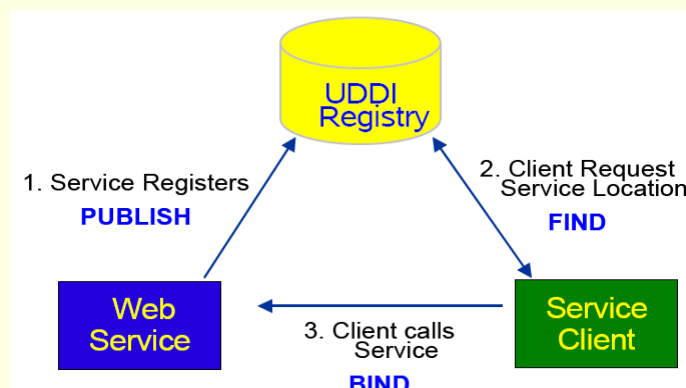
April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

85

Service Architecture

- UDDI defines a way to **publish** and **find** information about Web services.



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

86

UDDI

- “White pages”
 - address, contact, and known identifiers
- “Yellow pages”
 - industrial **categorizations**
- Industry: NAICS (Industry codes - US Govt.)
- Product/Services: UN/SPSC (ECMA)
- Location: Geographical taxonomy
- “Green pages”
 - **technical information** about services

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

87

UDDI

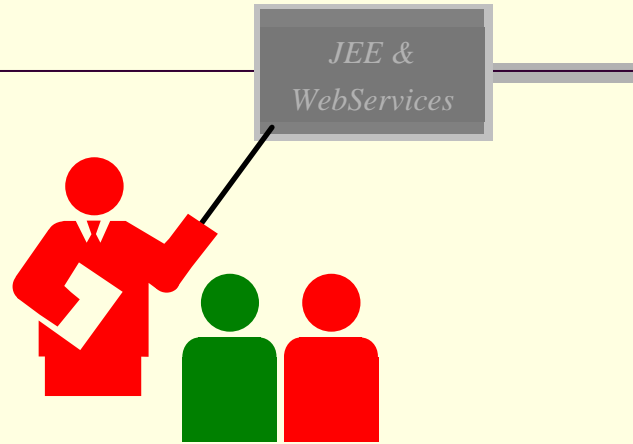
- UDDI is used to register and look up services with a central registry
- Service Providers can publish information about their business and the services that they offer
- Service consumers can look up services that are available by
 - **Business**
 - **Service category**
 - **Specific service**

April 05

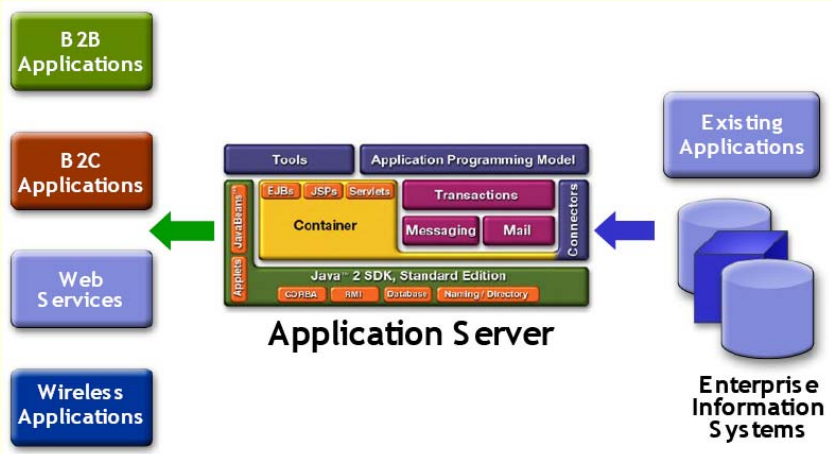
Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

88

SOA



J2EE Platform & Web Services



Why J2EE for Web Services?

- Web services is just one of many service delivery channels of J2EE
 - No architectural change is needed
 - Existing J2EE components can be easily exposed as Web services
- Many benefits of J2EE are preserved for Web services
 - Portability, Scalability, Reliability
 - No single-vendor lock-in

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

91

Where Are We Now?

- Java APIs for Web Services are being developed very rapidly
- Tools are available now for exposing existing J2EE components as Web services
- J2EE community has defined overall framework for Web Services (J2EE 1.4, JSR 109)

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

92

Design Goals J2EE Web Services Framework

- **Portability of Web services component**
 - Over different vendor platform
 - Over different operational environment
- **Leveraging existing J2EE programming models for service implementation**
- **Easy to program and deploy**
 - High-level Java APIs
 - Use existing deployment model

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

93

J2EE Web Services Framework

- **J2EE 1.4 and Java EE 5**
 - Umbrella framework for Web services
 - JSR 109, JAX-RPC, JAXR, EJB 2.1, Servlet 2.4,
- **JAX-RPC (JAX-WS)**
 - Defines client programming model
 - Defines Servlet-based Web services endpoint model
- **EJB 2.1**
 - Defines Stateless Session Bean-based Web services endpoint model
- **Servlet 2.4**
 - Will be aligned with JAX-RPC
- **JSR 109**
 - Defines standard Web services packaging and deployment model

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

94

Web Services Architecture over J2EE

- A Web Service is
 - A set of endpoints (ports) operating on messages
 - Ports are operating within a container
 - Container provides runtime environment
 - Contract for runtime environment are specified in JAX-RPC, EJB 2.1, JSR 109
 - Service is described in WSDL document and published to a registry
 - WSDL specifies a contract between service provider and client

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

95

Web Service Component and Container

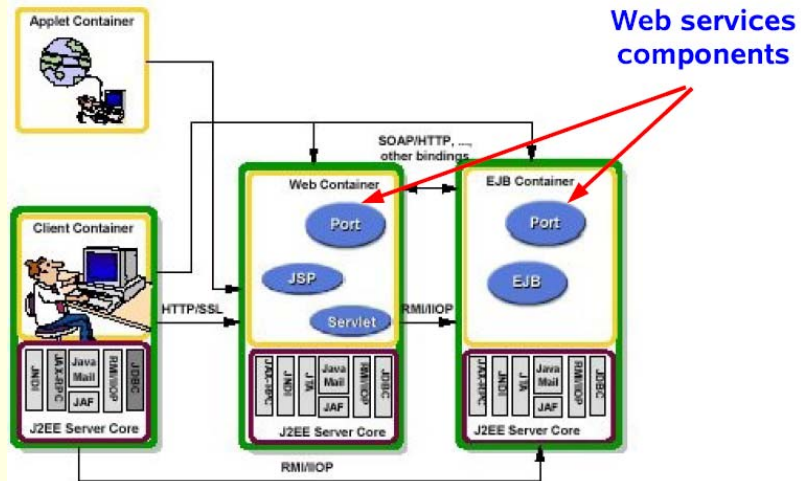
- Container vs. Component model
 - Web services components get executed within a container
 - Components are portable (under J2EE 1.4)
- Web service components
 - Web-tier (Servlet-based endpoint)
 - EJB-tier (Stateless session bean-based endpoint)

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

96

Web Service Components



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

97

SOA

Why to use
WebServices



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

98

Why to use Web Services...

(comparing to CORBA)

- WS are easier to deploy because of their firewall-friendliness
- WS are quite well marketed (both from IT companies and Open Source projects)
- However:
 - user sessions are less standardised
 - many parts yet-to-be-done (notification, transactions, security, etc.)
- The programming effort and maintainability is similar to other distributed technologies

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

99

1. What is similar

- The programming effort and maintainability is roughly the same both for Web Services and CORBA
 - For CORBA I need an ORB
 - ...but do you know anybody doing WS without a SOAP toolkit?
 - For CORBA I need an IDL compiler
 - ...not always (ask Perl folks)
 - ...for WS you use frequently stubs generated from WSDL
 - ...similar answers for valuetype/custom encoding, etc.

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

100

2. What is better

- WS are easier to deploy because of their firewall-friendliness
- WS are quite well marketed (both from IT companies and Open Source projects)
- Integration of WS into workflows seems to be very dynamic and very real topic
 - comparing with CORBA Components

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

101

3. What is worse

- Peer-to-peer access is problematic
 - notification by "server-push" is harder to achieve
- User sessions (server's state-fullness) are less standardised
 - ...and therefore less inter-operable
- Many parts yet-to-be-done, or they are quite complex (notification, transactions, security, etc.)

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

102

So what?

- Don't throw the baby out with the bathwater
 - combine the existing projects with a new Web Services layer; in most cases it is not so difficult
- Apply existing standards to new Web Services projects
 - think MDA – it may help, even without the whole OMG adoption process

Conclusions

- Distributed computing is inevitable
- More accesses by programs than by clicking on hyperlinks
- More technologies of distributed architecture will collaborate
- The better standards we have the better results we'll get Web Services is not a new hype but a trend to follow