Web

Master 1 IFI

Andrea G. B. Tettamanzi

Université de Nice Sophia Antipolis

Département Informatique

andrea.tettamanzi@unice.fr

Unit 6

An introduction to Web Services, UDDI, and SOAP

Agenda

- Introduction
- Web Services
- WSDL
- UDDI
- SOAP
- SOAP in J2EE

Introduction

- The term “Web service” can be used in two senses
  - Generic: any service offered through the HTTP protocol
  - Specific: W3C Web Services
- HTTP is used for machine-to-machine communication
  - machine-readable formats such as XML and JSON
- Typical use case: OO Web-based interface to a DB server
- Many organizations with Web pages also provide the same data through a Web service to allow **syndication**
  - e.g., Wikipedia's Export.
- Another application is **mashup**, where a Web server calls several Web services and compiles the content into one user interface

Web Syndication

- A form of syndication
  - Originated in media such as print, radio, and television
  - Allows content creators to reach a wider audience
- Content is made available from one Web site to other sites
  - Summaries or full renditions of recently added content
  - On a subscription basis, for free, or barter
- Popular Web feed syndication formats:
  - **RSS** (Rich Site Summary / Really Simple Syndication)
  - **Atom**, born to overcome limitations of RSS
- Content creators: (+) amortize costs; (–) may lose control
- Content distributors: (+) cheap content; (–) not exclusive

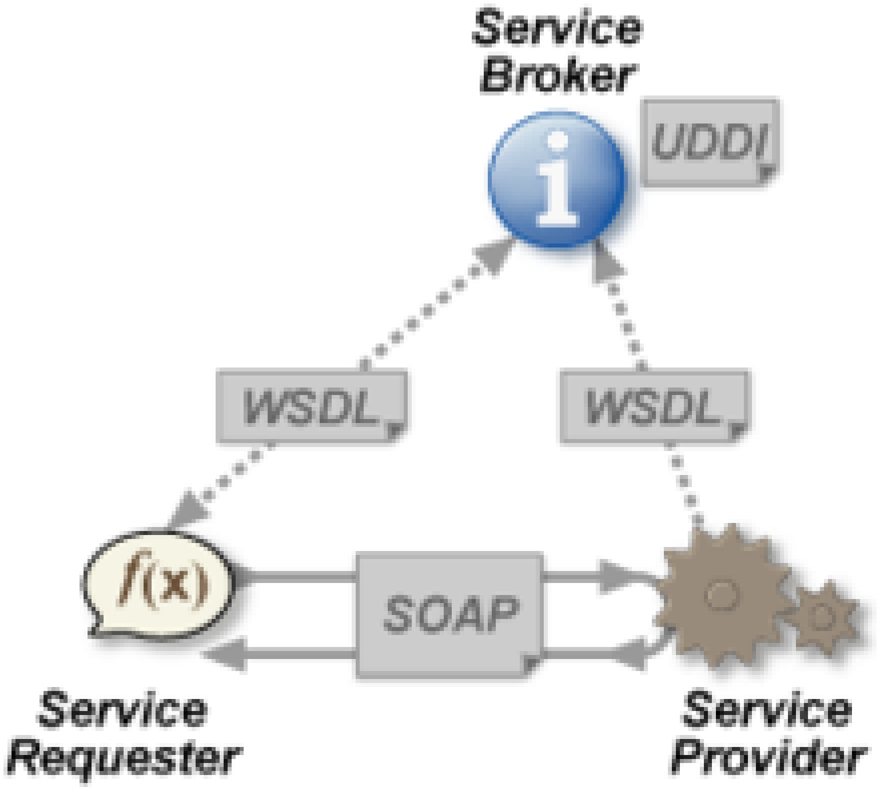
Generic Web Services

- Good news: you know a few of them already!
  - AJAX
  - REST-compliant Web services
  - RESTful APIs
- Can use markup languages
  - Web-Service Description Language (WSDL)
  - Web-Service Conversation Language (WSCL)
  - Web-Services Flow Language (WSFL), superseded by BPEL (Business Process Execution Language)
- Or other formats:
  - JSON-WSP, a protocol based on JSON

W3C Web Services

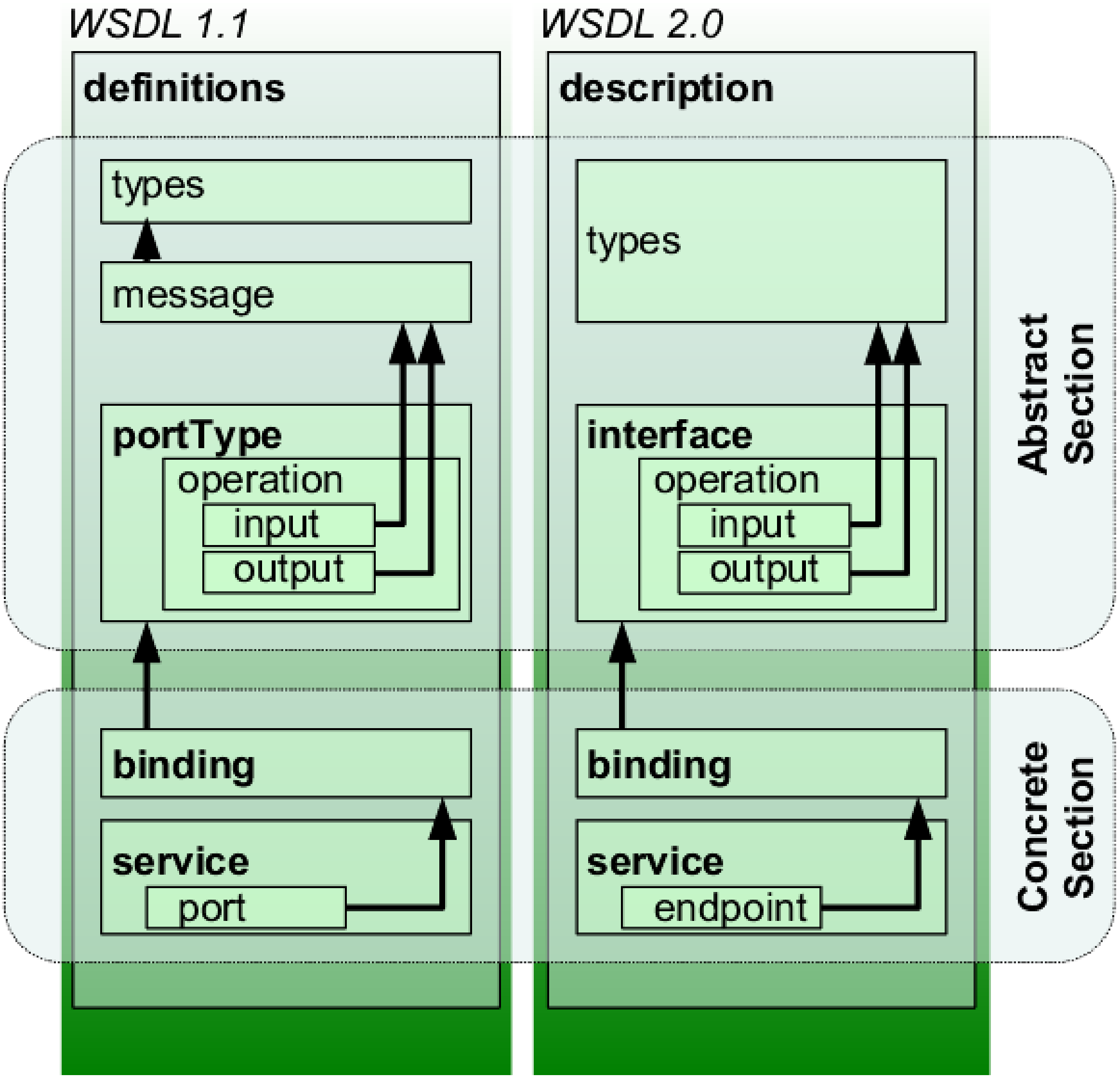
- A Web service is a software system designed to support interoperable machine-to-machine interaction over a network.
- It has an interface described in a machine-processable format (specifically, WSDL).
- Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages,
  - typically conveyed using HTTP with an XML serialization
  - in conjunction with other web-related standards
- SOAP over other protocols (e.g., FTP) is possible as well
- A service broker is needed to make services “discoverable”
  - UDDI is a standard for that purpose

Web Services Architecture



Web Service Description Language (WSDL)

- Rules for communication between different systems need to be defined, such as:
  - How one system can request data from another system
  - Which specific parameters are needed in the data request
  - What would be the structure of the data produced
  - What error messages to display when a certain rule for communication is not observed, to make troubleshooting easier.
- All of these rules for communication are defined in a WSDL file
- WSDL is based on XML


WSDL 1.1
definitions
types
message
portType
operation
input
output
binding
service
port
WSDL 2.0
description
types
interface
operation
input
output
binding
service
endpoint
Abstract Section
Concrete Section

WSDL Example

```
<?xml version="1.0" encoding="UTF-8"?>
<description xmlns="http://www.w3.org/ns/wsdl"
         xmlns:tns="http://www.tmsws.com/wsdl20sample"
         xmlns:whttp="http://schemas.xmlsoap.org/wsdl/http/"
         xmlns:wsoap="http://schemas.xmlsoap.org/wsdl/soap/"
         targetNamespace="http://www.tmsws.com/wsdl20sample">
<documentation>
   This is a sample WSDL 2.0 document.
</documentation>
<!-- Abstract type -->
  <types>
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://www.tmsws.com/wsdl20sample"
           targetNamespace="http://www.example.com/wsdl20sample">
      <xs:element name="request"> ... </xs:element>
      <xs:element name="response"> ... </xs:element>
    </xs:schema>
  </types>
<!-- Abstract interfaces -->
  <interface name="Interface1">
    <fault name="Error1" element="tns:response"/>
    <operation name="Get" pattern="http://www.w3.org/ns/wsdl/in-out">
      <input messageLabel="In" element="tns:request"/>
      <output messageLabel="Out" element="tns:response"/>
    </operation>
  </interface>
<!-- Concrete Binding Over HTTP -->
  <binding name="HttpBinding" interface="tns:Interface1"
        type="http://www.w3.org/ns/wsdl/http">
    <operation ref="tns:Get" whttp:method="GET"/>
  </binding>
<!-- Concrete Binding with SOAP-->
  <binding name="SoapBinding" interface="tns:Interface1"
        type="http://www.w3.org/ns/wsdl/soap"
wsoap:protocol="http://www.w3.org/2003/05/soap/bindings/HTTP/"
        wsoap:mepDefault="
http://www.w3.org/2003/05/soap/mep/request-response">
    <operation ref="tns:Get" />
  </binding>
<!-- Web Service offering endpoints for both bindings-->
  <service name="Service1" interface="tns:Interface1">
    <endpoint name="HttpEndpoint"
          binding="tns:HttpBinding"
          address="http://www.example.com/rest/"/>
    <endpoint name="SoapEndpoint"
          binding="tns:SoapBinding"
          address="http://www.example.com/soap/"/>
  </service>
</description>
```

Universal Description, Discovery, and Integration (UDDI)

- A directory which defines which software system should be contacted for which type of data.
- When one software system needs one particular report/data, it would go to the UDDI and find out which other system it can contact for receiving that data.
- Once the software system finds out which other system it should contact, it would then contact that system using a special protocol called SOAP (Simple Object Access Protocol).
- The service provider system would first validate the data request by referring to the WSDL file, and then process the request and send the data under the SOAP protocol.

Structure of UDDI

A UDDI business registration consists of three components:

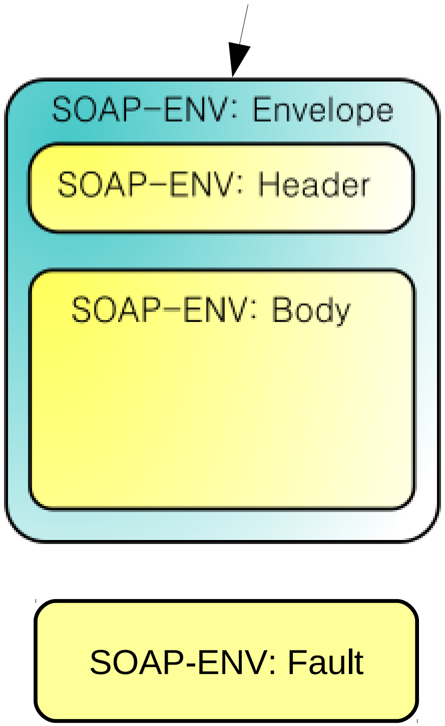
- White Pages
  - address, contact, and known identifiers
- Yellow Pages
  - industrial categorizations based on standard taxonomies, like SIC, NAICS, UNSPCSC, …
- Green Pages
  - technical information about services exposed
  - how to access each Web service
  - information on the service bindings

Simple Object Access Protocol (SOAP)

- Messaging protocol to exchange structured information
- Developed for Web services
- Goal: to provide extensibility, neutrality, and independence.
- Uses XML Information Set for its message format
- Relies on application-layer protocols, like HTTP, FTP, or SMTP
- SOAP allows clients to invoke Web services and receive responses independently of language and platforms.
- The verbosity of the protocol, slow parsing speed of XML, and lack of a standardized interaction model led to the domination in the field by RESTful services, using HTTP more directly

SOAP Message Structure

Identifies the XML document as a SOAP message



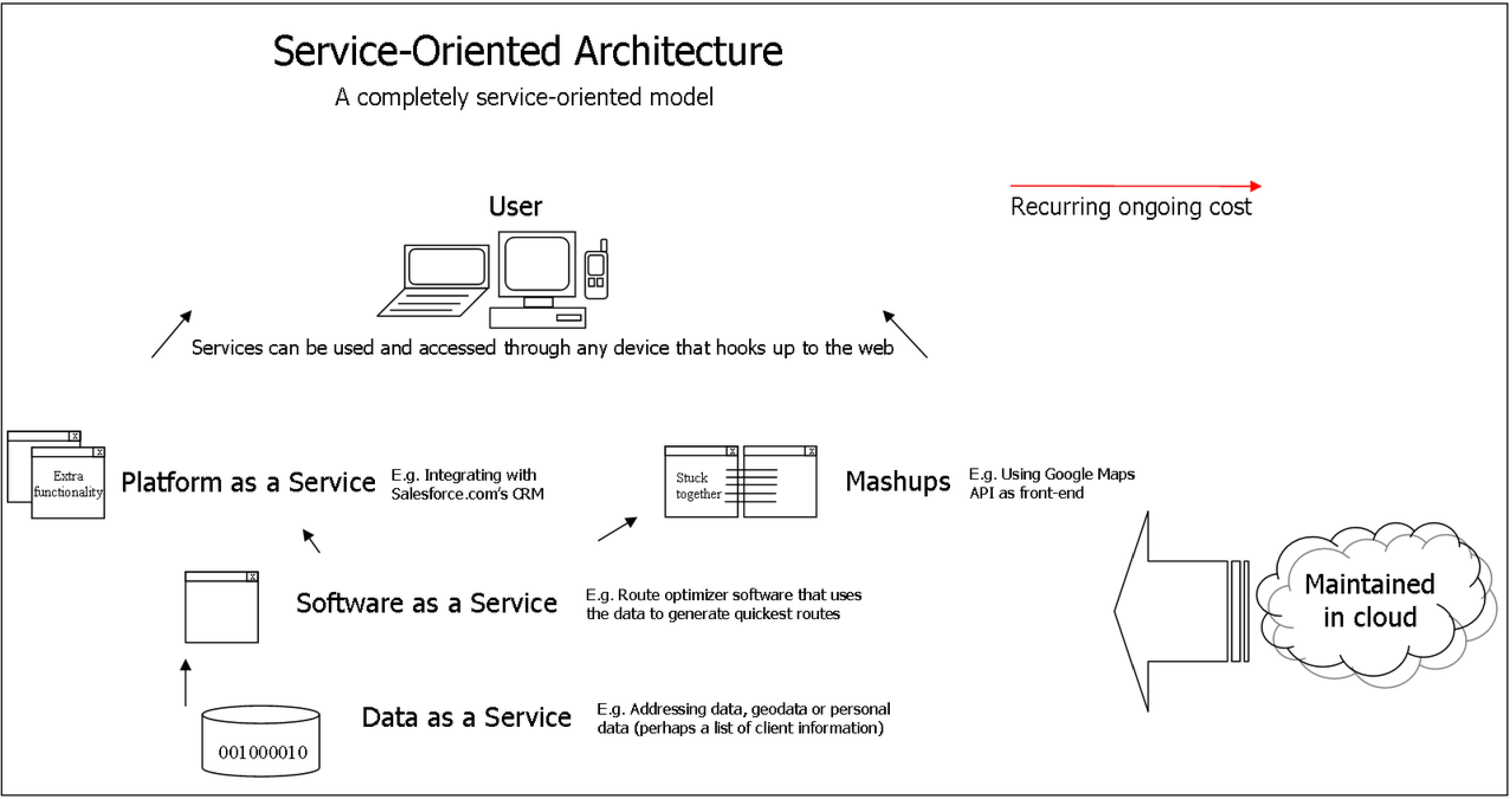
Contains call and response information

Provides information about errors that occurred while processing the message.

SOAP Example

```
POST /InStock HTTP/1.1
Host: www.example.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: 299
SOAPAction: "http://www.w3.org/2003/05/soap-envelope"


<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
               xmlns:m="http://www.example.org">
  <soap:Header>
  </soap:Header>
  <soap:Body>
    <m:GetStockPrice>
      <m:StockName>GOOG</m:StockName>
    </m:GetStockPrice>
  </soap:Body>
</soap:Envelope>
```



Service-Oriented Architecture

A completely service-oriented model

Recurring ongoing cost

Services can be used and accessed through any device that hooks up to the web

Platform as a Service — E.g. Integrating with Salesforce.com's CRM

Mashups — E.g. Using Google Maps API as front-end

Software as a Service — E.g. Route optimizer software that uses the data to generate quickest routes

Data as a Service — E.g. Addressing data, geodata or personal data (perhaps a list of client information)

J2EE Support of Web Services

- With the release of Java 6, the Java EE platform provides a comprehensive support of Web services (the Metro stack):
  - JAX-WS - provides a complete framework for the development of both SOAP-based and RESTful Java Web services
  - JAXB - assists JAX-WS in processing XML by allowing easy binding of XML schema to standard Java formats
  - WSIT - extends JAX-WS to allow Java-based web services to interact with .NET and other WCF components.
  - XWS-Security - configuration of security policies
  - JAX-RPC - legacy support for RPC-based services
- Glassfish application server