



Software Service Engineering

Prof. Dr.-Ing. Martin Gaedke

Technische Universität Chemnitz

Fakultät für Informatik

Professur Verteilte und selbstorganisierende
Rechnersysteme

<http://vsr.informatik.tu-chemnitz.de>



Chapter 2

WEB SERVICES DESCRIPTION LANGUAGE (WSDL)



WSDL – Introduction

- Web Services Description Language (WSDL)
 - Use of XML to describe functionality as a set of endpoints
 - Similarly to an IDL: methods, arguments and return values are described
 - Unlike an IDL, message encoding and exchange methods are described as well
- History:
 - WSDL 1.1: W3C Note 15 March 2001
 - <http://www.w3.org/TR/wsdl>
 - Independent development efforts of IBM and Microsoft
 - Current: Web Services Description Language (WSDL) 2.0 – W3C Recommendation 26 June 2007



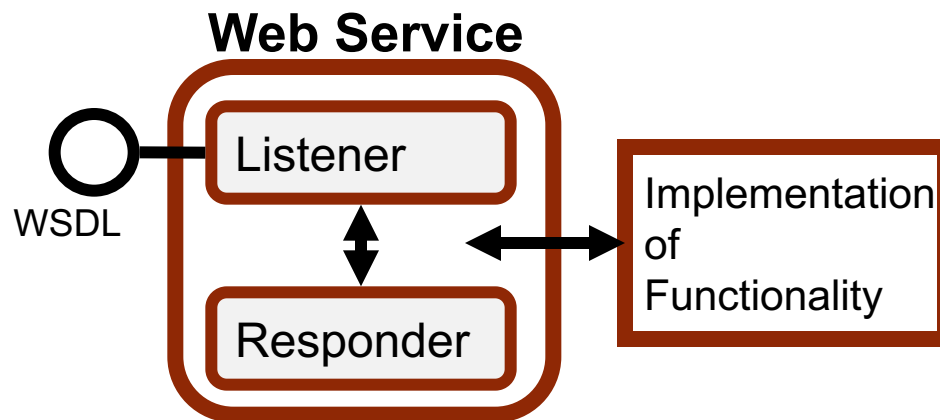
Describing Web Services

- Web Services Description Language (WSDL)
1.1
 - Status: W3C Note 15 March 2001
 - <http://www.w3.org/TR/wsdl>
 - Independent efforts from IBM and Microsoft
- WSDL is for describing Web Services
 - Defines XML-based grammar for describing network services as a set of endpoints
 - Describes their methods, arguments, return values and how to use



Web Services Architecture

- WSDL: Core element of the Web Service Architecture stack (Endpoint definition language)



Simplified Web Service Stack (WS-I Basic Profile 1.0 compliant)

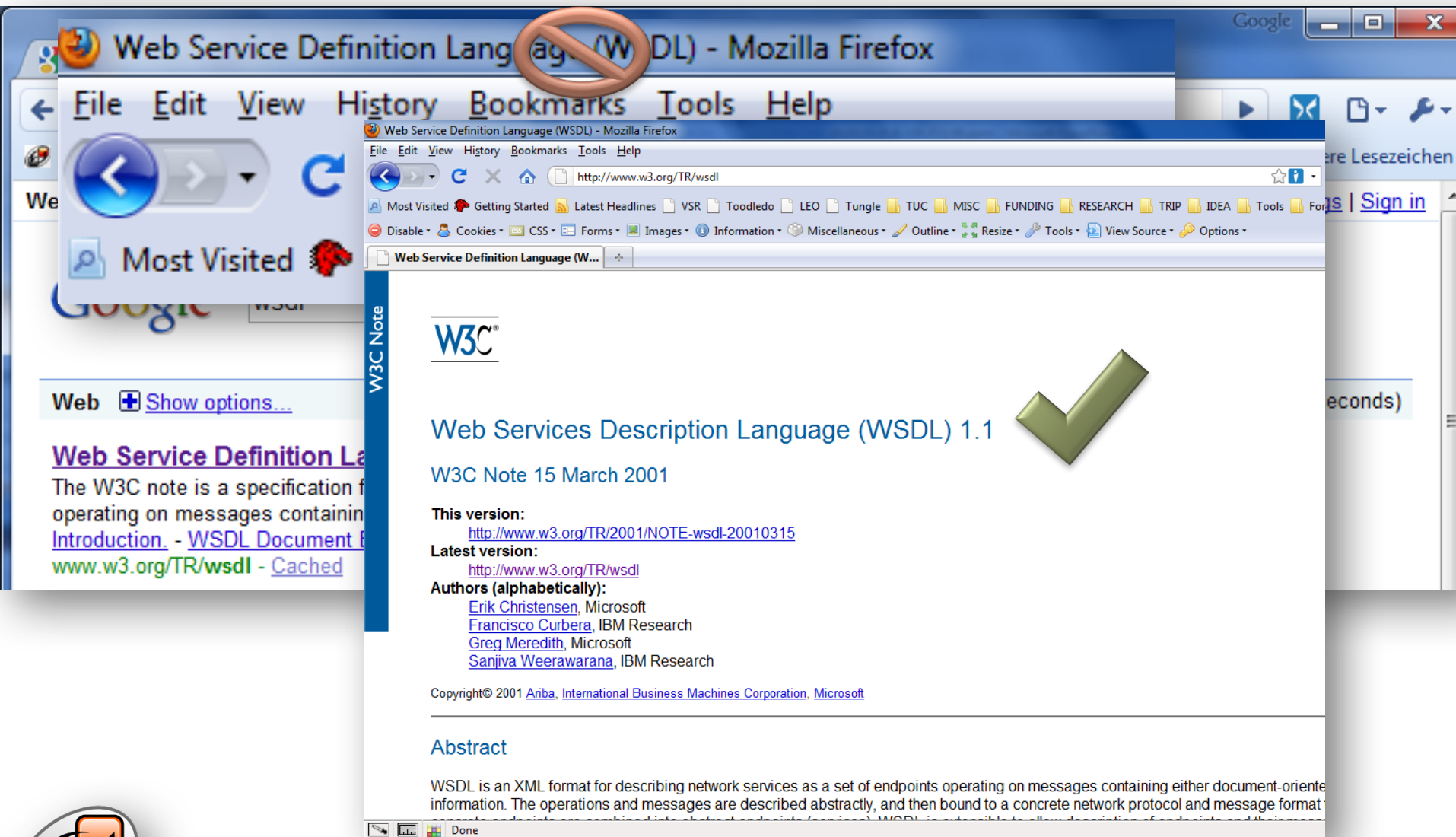
UDDI (service discovery)
WSDL (service description)
XSD (service description)
SOAP (messaging)
XML 1.0 + Namespaces (messaging)

WSDL Goals

- Extensibility wrt.
 - New Transport protocols
 - New Encoding rules
- Abstraction wrt.
 - Endpoints and Messages
 - THEN mapped onto **n** concrete transports and encodings
- Reuse wrt.
 - Definitions – reusable to create new definitions



WSDL stands for...



The screenshot shows a Mozilla Firefox browser window with the title "Web Service Definition Language (WSDL) - Mozilla Firefox". The address bar shows the URL "http://www.w3.org/TR/wsdl". The page content includes the W3C logo, the title "Web Services Description Language (WSDL) 1.1", the date "W3C Note 15 March 2001", and a list of authors: Erik Christensen, Francisco Curbera, Greg Meredith, and Sanjiva Weerawarana. A green checkmark is placed next to the title "Web Services Description Language (WSDL) 1.1".

Web Service Definition Language (WSDL) 1.1

W3C Note 15 March 2001

This version:
<http://www.w3.org/TR/2001/NOTE-wsdl-20010315>

Latest version:
<http://www.w3.org/TR/wsdl>

Authors (alphabetically):
[Erik Christensen](#), Microsoft
[Francisco Curbera](#), IBM Research
[Greg Meredith](#), Microsoft
[Sanjiva Weerawarana](#), IBM Research

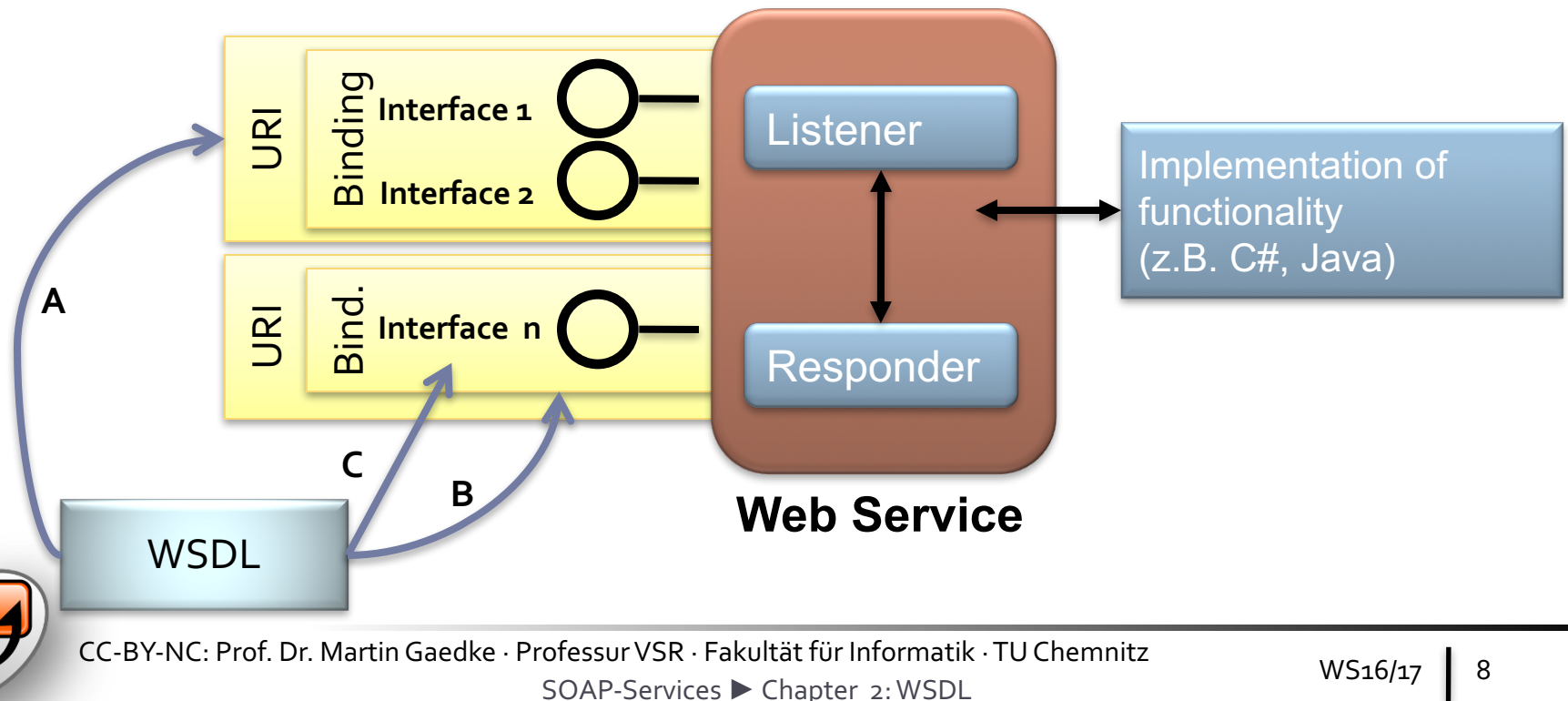
Copyright© 2001 [Aniba](#), [International Business Machines Corporation](#), [Microsoft](#)

Abstract

WSDL is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented information. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format.

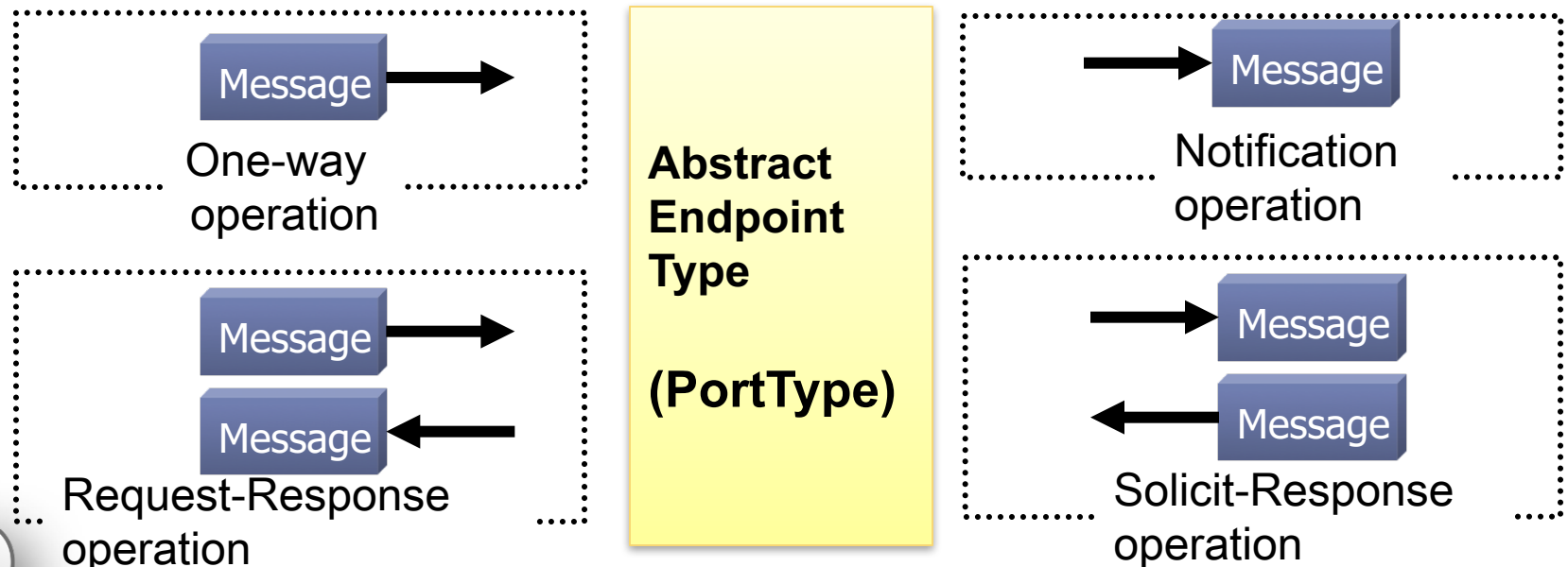
Web Service - Description

- WSDL is the central element of Web Services Technologies in WSA
 - Technology-independent XML description of endpoints (ABC)



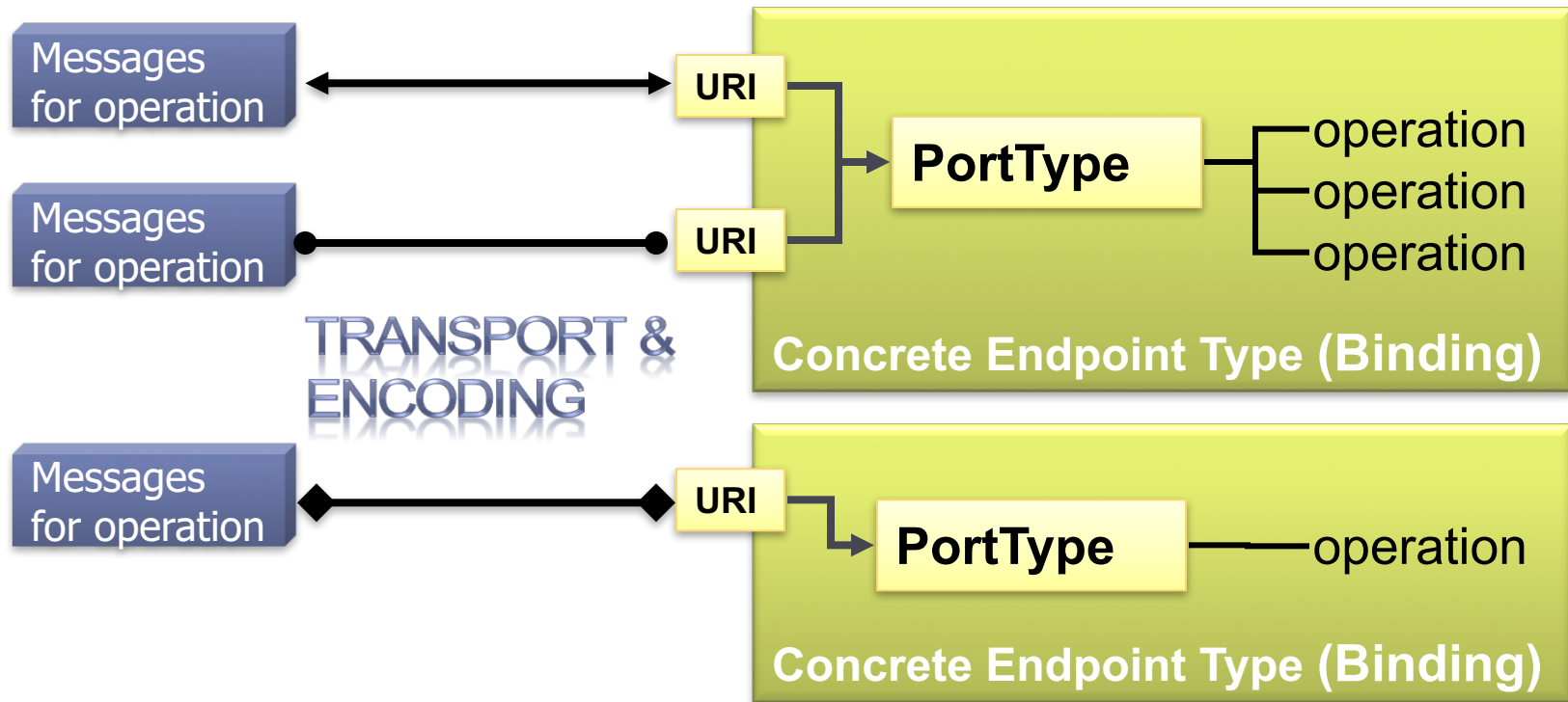
WSDL: Abstract Endpoint Type

- **Abstract Endpoint Type** (PortType / Contract) – (Possible) Part of WSDL specification
 - Describes: Message types, operations and parameters
 - Set of message flows (operations), which are expected from a dedicated endpoint type (independent of technology in use)



WSDL: Concrete Endpoint Type

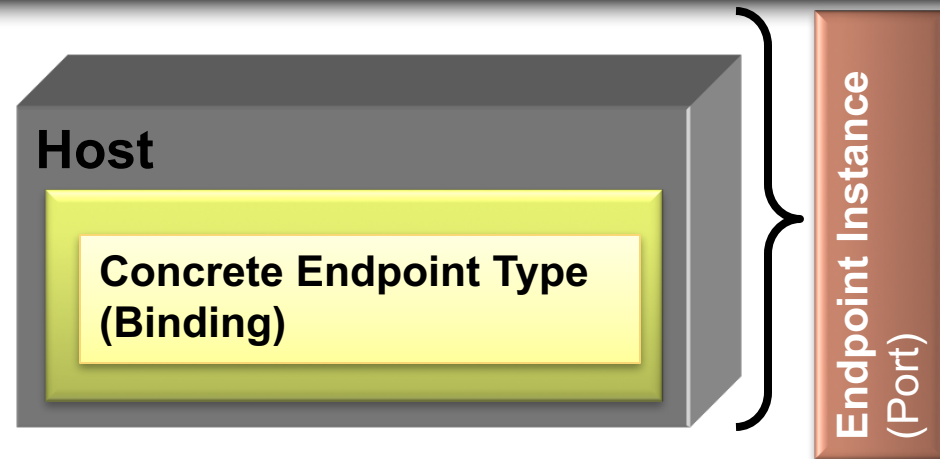
- **Concrete Endpoint Type (Binding)** – Defines transport and encoding particulars for a portType



WSDL: Service Definition

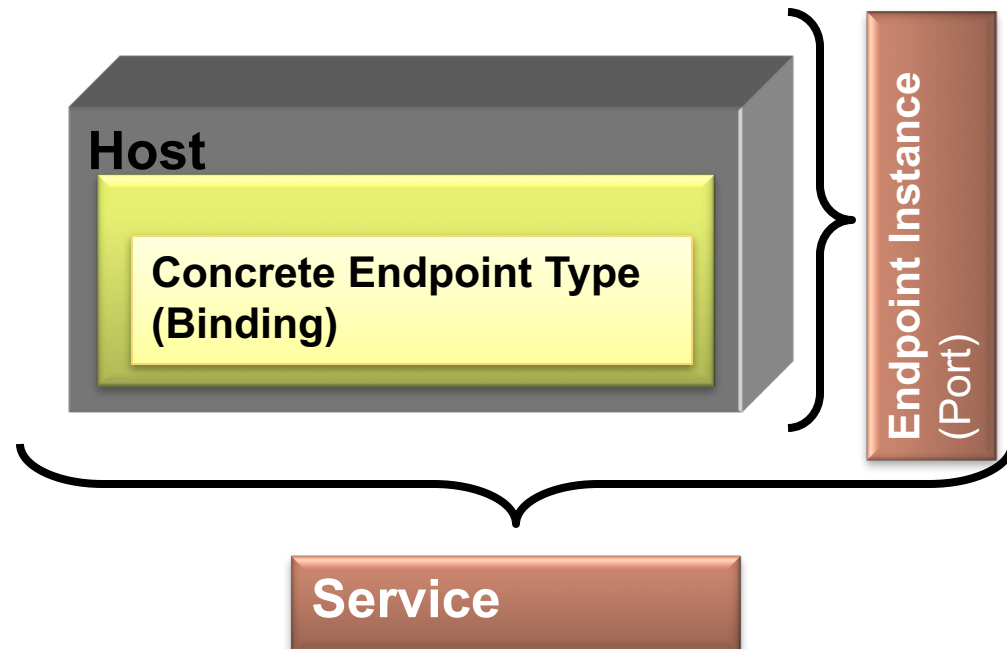
- **Port (Endpoint Instance)**

- Addresses of an endpoint and the according binding (address is a URI)
- Attention: Do not confuse Port with a TCP port!



- **Service**

- A set of related endpoint instances



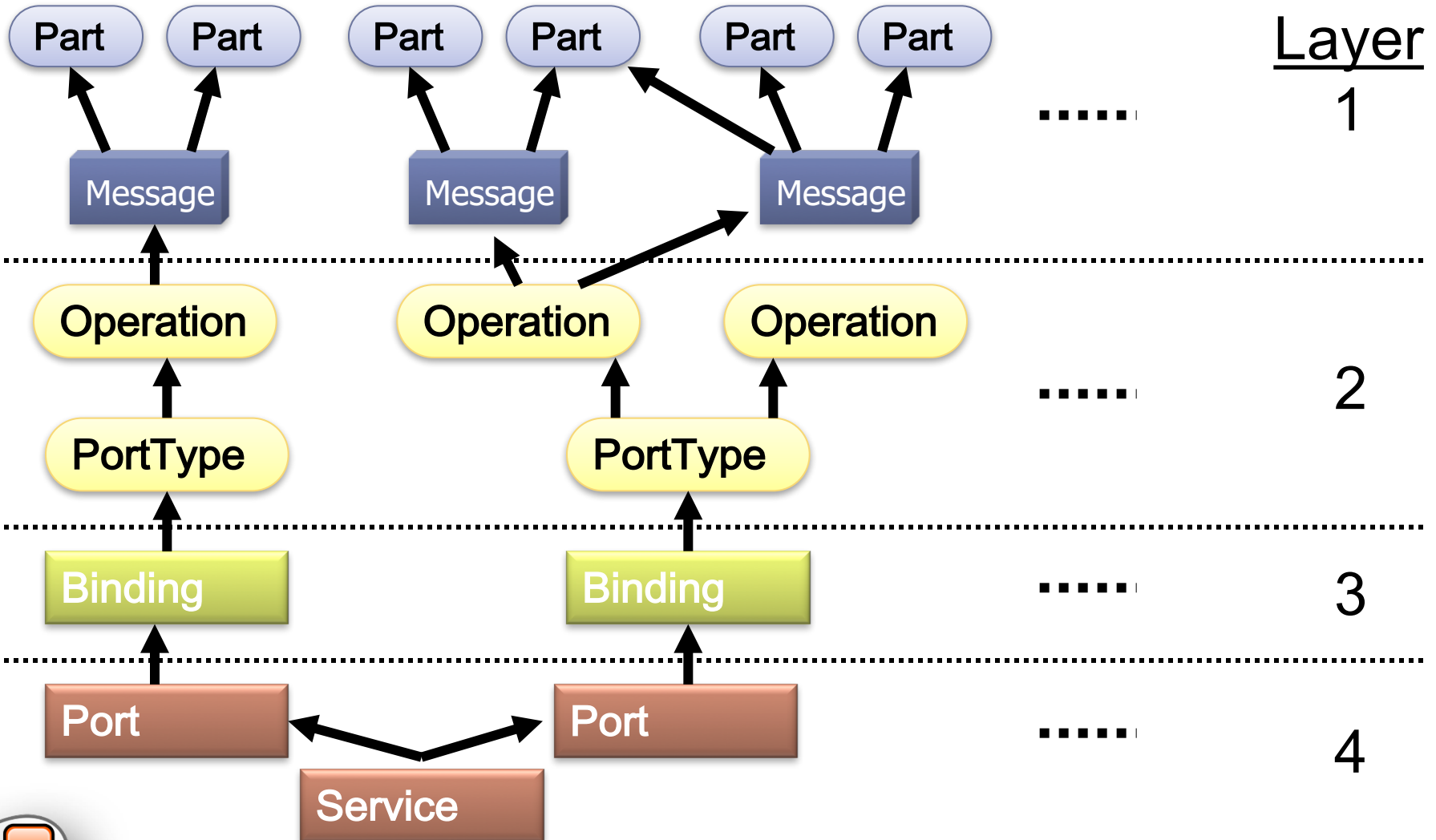
Vocabulary (1)

- All WSDL elements belong to a WSDL Namespace:
<http://schemas.xmlsoap.org/wsdl/>
- Namespaces for WSDL Binding
 - SOAP Binding:
<http://schemas.xmlsoap.org/wsdl/soap/>
 - HTTP GET and POST Binding:
<http://schemas.xmlsoap.org/wsdl/http/>
 - WSDL MIME binding:
<http://schemas.xmlsoap.org/wsdl/mime/>
 - Various other...
- Development of WSDL specification
- Problem: WSDL-first vs. Code-first
 - **Attention! Follow the rules of WS-I**
(<http://www.oasis-ws-i.org>)

The following examples and definitions are based on WSDL 1.1



Vocabulary (2)



WSDL – Example (1)

Part

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:tns="http://example.org/beispielws"
  xmlns:s="http://www.w3.org/2001/XMLSchema"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  targetNamespace="http://example.org/beispielws"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://example.org/beispielws">
      <s:element name="HelloWorld">
        <s:complexType />
      </s:element>
      <s:element name="HelloWorldResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="HelloWorldResult" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:schema>
  </wsdl:types>
</wsdl:definitions>
```

```
        <s:element name="Add">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="a" type="s:int" />
      <s:element minOccurs="1" maxOccurs="1" name="b" type="s:int" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="AddResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="AddResult" type="s:int" />
    </s:sequence>
  </s:complexType>
</s:element>
</s:schema>
</wsdl:types>
```



```
<wsdl:message name="HelloWorldSoapIn">
  <wsdl:part name="parameters" element="tns:HelloWorld" />
</wsdl:message>
<wsdl:message name="HelloWorldSoapOut">
  <wsdl:part name="parameters" element="tns:HelloWorldResponse" />
</wsdl:message>

<wsdl:message name="AddSoapIn">
  <wsdl:part name="parameters" element="tns:Add" />
</wsdl:message>
<wsdl:message name="AddSoapOut">
  <wsdl:part name="parameters" element="tns:AddResponse" />
</wsdl:message>
```


WSDL – Example (4)

Operation

PortType

```
<wsdl:portType name="ServiceSoap">
  <wsdl:operation name="HelloWorld">
    <wsdl:input message="tns:HelloWorldSoapIn" />
    <wsdl:output message="tns:HelloWorldSoapOut" />
  </wsdl:operation>

  <wsdl:operation name="Add">
    <wsdl:input message="tns:AddSoapIn" />
    <wsdl:output message="tns:AddSoapOut" />
  </wsdl:operation>
</wsdl:portType>
```

WSDL – Example (5)

Binding

Operation

```
<wsdl:binding name="ServiceSoap" type="tns:ServiceSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="HelloWorld">
    <soap:operation soapAction="http://example.org/beispielws/HelloWorld" style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>

  <wsdl:operation name="Add">
    <soap:operation soapAction="http://example.org/beispielws/Add" style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
```

WSDL – Example (6)

Binding

Operation

```
<wsdl:binding name="ServiceSoap12" type="tns:ServiceSoap">
  <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="HelloWorld">
    <soap12:operation soapAction="http://example.org/beispielws/HelloWorld" style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>

  <wsdl:operation name="Add">
    <soap12:operation soapAction="http://example.org/beispielws/Add" style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
```



WSDL – Example (7)

Port

Service

```
<wsdl:service name="Service">
  <wsdl:port name="ServiceSoap" binding="tns:ServiceSoap">
    <soap:address location="http://.../WSExample/Service.asmx" />
  </wsdl:port>

  <wsdl:port name="ServiceSoap12" binding="tns:ServiceSoap12">
    <soap12:address location="http://.../WSExample/Service.asmx" />
  </wsdl:port>
</wsdl:service>

</wsdl:definitions>
```

Web Service Demo

- WSDL focus and discussion
- Problem: Multiple vs. single binding
- Approach: Code First vs. WSDL First
- Example: Complex data types
- WS-I Basic Profile



Chapter 3

WSDL 2.0

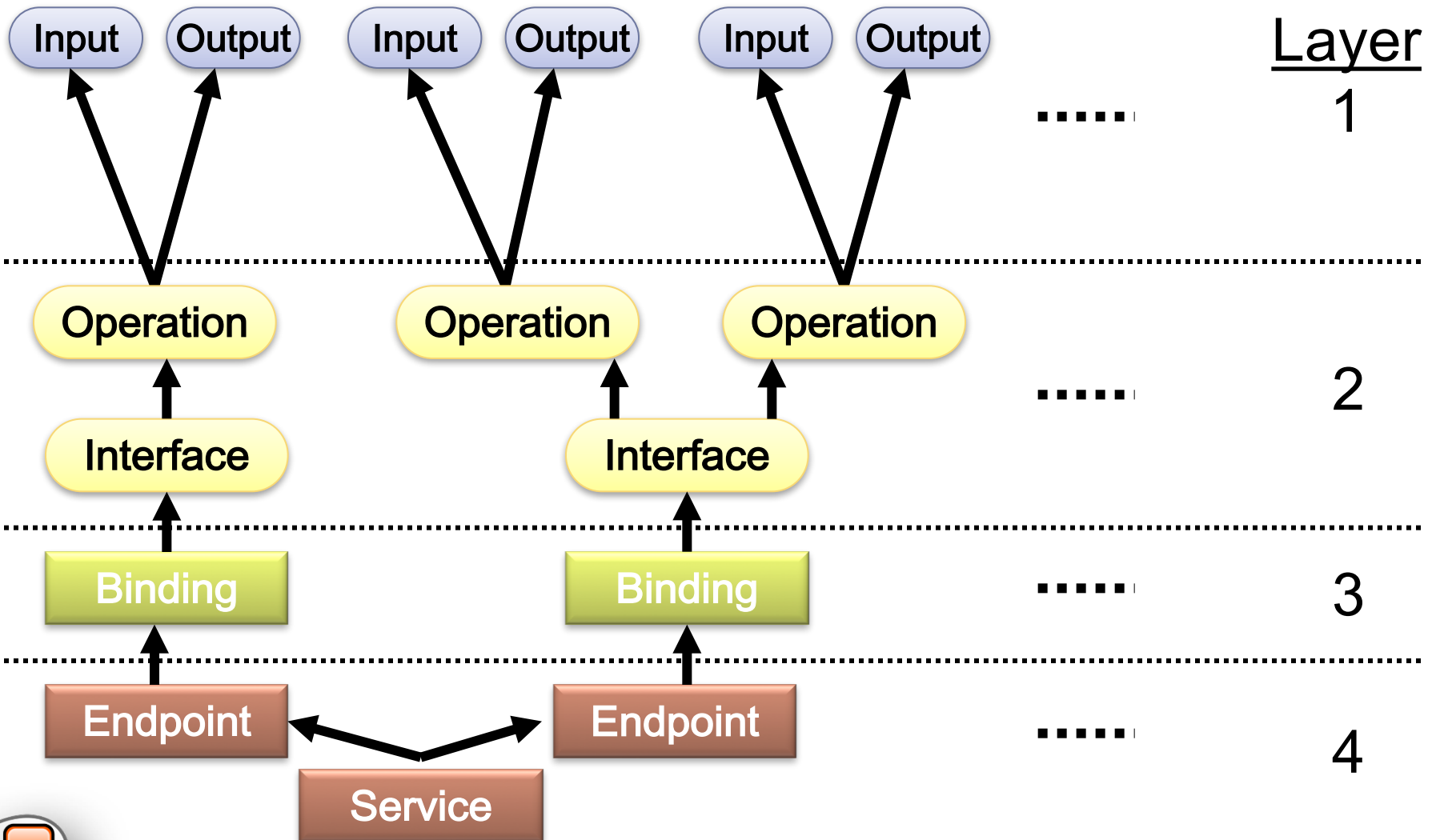


Changes

- Lower complexity than WSDL 1.1
- Port
 - Renamed to *Endpoint*
- PortType
 - Renamed to *Interface*
 - supporting inheritance by using *extends* attribute
- Message
 - Removed, instead inputs, outputs, faults etc. directly typed
- Improved support for non-SOAP Web services (cf. Part IV REST)



Revised Concepts



WSDL 2.0 Example

Interface

Operation

Input

Output

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:description
  targetNamespace="http://ticketAgent.example.com/"
  xmlns:tns="http://ticketAgent.example.com/"
  xmlns:wsdl="http://www.w3.org/ns/wsdl"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3.org/ns/wsdl
http://www.w3.org/2007/06/wsdl/wsdl20.xsd">
  <wsdl:types>
    <xs:import schemaLocation="TicketAgent.xsd"
namespace="http://ticketAgent.example.com/TicketAgent.xsd" />
  </wsdl:types>
  <wsdl:interface name="TicketAgent">
    <wsdl:operation name="listFlights" pattern="http://www.w3.org/ns/wsdl/in-out">
      <wsdl:input element="tns:listFlightsRequest"/>
      <wsdl:output element="tns:listFlightsResponse"/>
    </wsdl:operation>
    <wsdl:operation name="reserveFlight" pattern="http://www.w3.org/ns/wsdl/in-out">
      <wsdl:input element="tns:reserveFlightRequest"/>
      <wsdl:output element="tns:reserveFlightResponse"/>
    </wsdl:operation>
  </wsdl:interface>
```

Adapted from: <http://www.w3.org/TR/wsdl20/>



WSDL 2.0 Example

Service

Endpoint

Binding

```
<binding name="HttpBinding" interface="tns:TicketAgent"
        type="http://www.w3.org/ns/wsdl/http">
  <operation ref="tns:listFlights" whttp:method="GET" />
  <operation ref="tns:reserveFlight" whttp:method="POST"
    whttp:inputSerialization="application/x-www-form-urlencoded" />
</binding>

<service name="TicketAgentService" interface="tns:TicketAgent">
  <endpoint name="TicketAgentService Endpoint"
    binding="tns:HttpBinding"
    address="http://ticketAgent.example.com/" />
</service>

</wsdl:description>
```

Adapted from: <http://www.w3.org/TR/wsdl20/>

