1918
TALLINNA
TEHNIKAÜLIKOOL

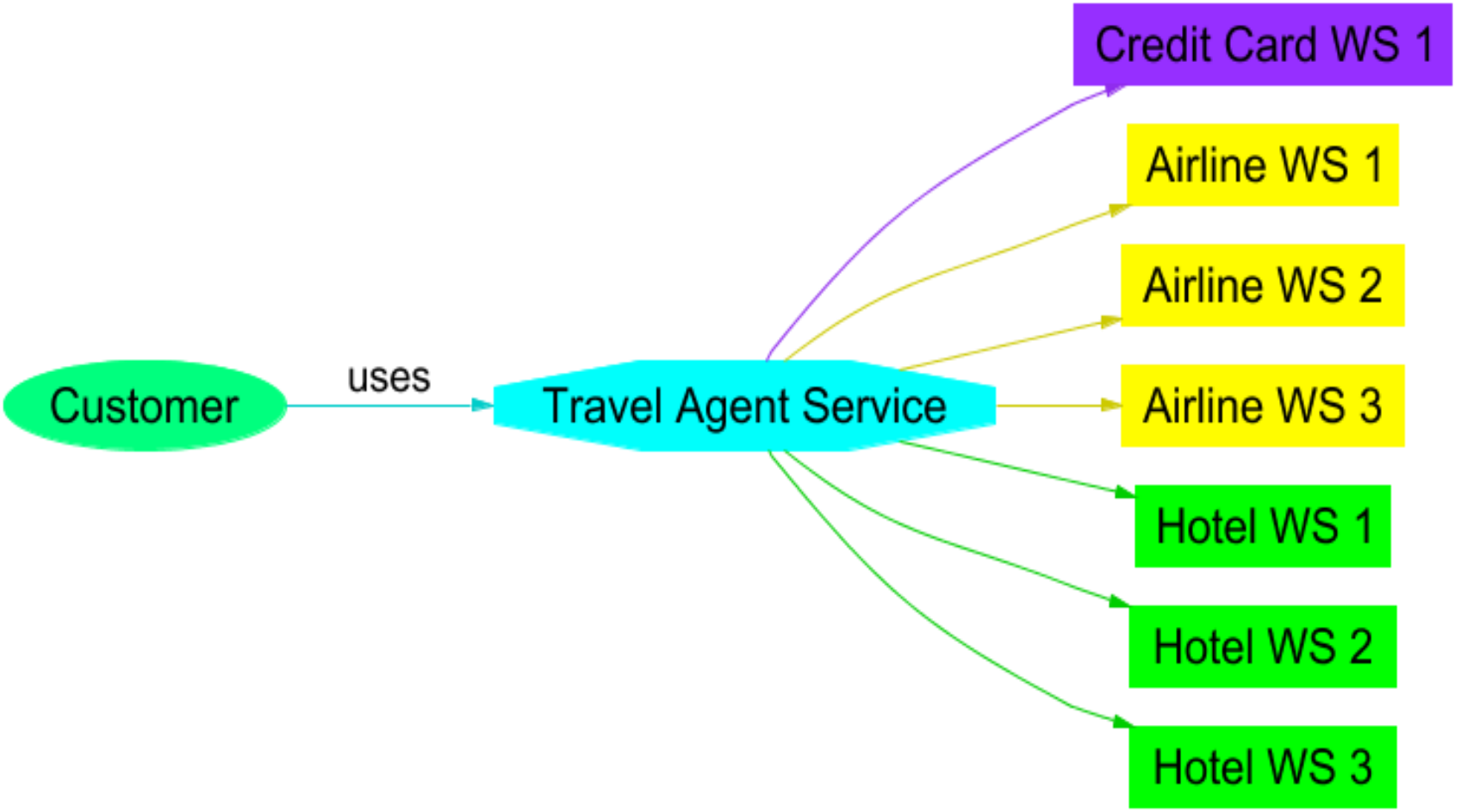
Organisatsiooni digitaalstrateegia ja äriprotsesside modelleerimine

Loeng 15

WSDL

Enn Õunapuu

enn@cc.ttu.ee


Credit Card WS 1
Airline WS 1
Airline WS 2
Airline WS 3
Customer
uses
Travel Agent Service
Hotel WS 1
Hotel WS 2
Hotel WS 3

Mis on XML?

- **XML** on meetod tekstilise informatsiooni struktureerimiseks
- **XML** näeb välja nagu HTML kuid ei ole HTML
- **XML** on tekst, kuid ta ei ole lugemiseks
- **XML** on tehnoloogiate perekond
- **XML** standard on liiga mahukas, kuid see ei ole probleem
- **XML** on uus tehnoloogia, kuid mitte liiga uus
- **XML** on litsentsivaba, platvormist sõltumatu ja hästi toetatud standard

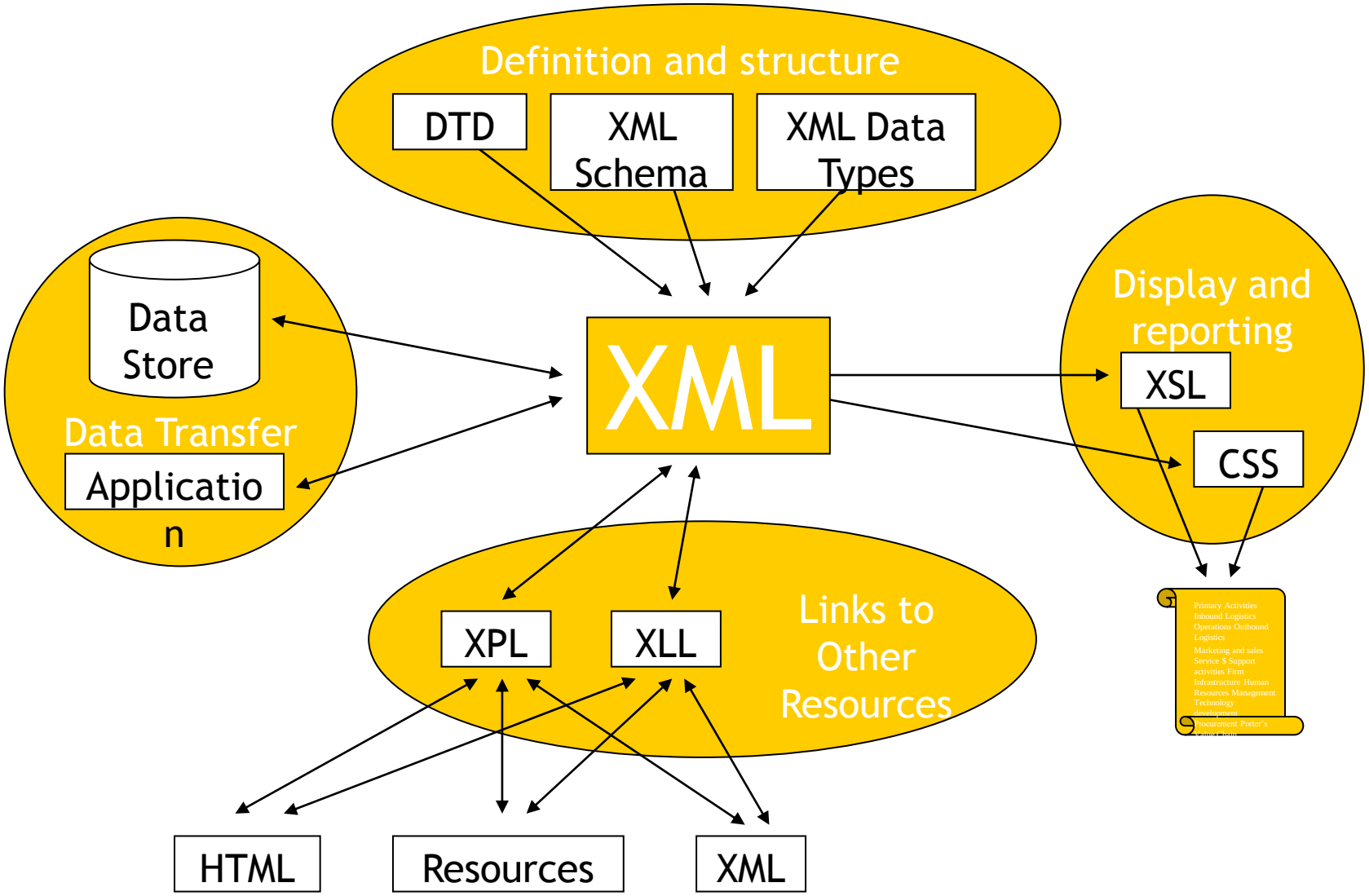
XML näide

- **Tavaline tekst:**

  Te võiksite vaadata Mark Wilson raamatut “XML and the Internet for Visual Basic 6” Manningsi kodulehel. Vaadake autori publikatsioonide jaotist *author online section*. Et saada kontakti teiste Visual Basic huvilistega minge http://www.vbxml.com

- **XML tekst**

  <review subject-type="book"> Te võiksite vaadata <AUTHOR>Mark Wilson</AUTHOR> raamatut <BOOK_NAME>“XML and the Internet for Visual Basic 6”</BOOK_NAME> <PUBLISHER>Manningsi</PUBLISHER> kodulehel. Vaadake autori publikatsioonide jaotist <LINK_AREA> *author online section*</LINK_AREA>. Et saada kontakti teiste <USER_TOPIC>Visual Basic </USER_TOPIC> huvilistega<BOOK_TOPIC>XML</BOOK_TOPIC> , minge <URL>http://www.vbxml.com</URL>.


Definition and structure
DTD
XML Schema
XML Data Types
Data Store
Data Transfer
Application
XML
Display and reporting
XSL
CSS
XPL
XLL
Links to Other Resources
HTML
Resources
XML

XML ülevaade

www.w3.org/xml SGML näide

```
<EMail>
    <sender>
      <person>
          <firstname> Karen </firstname>
          <lastname> Lemone </lastname>
      </person>
   <receiver>
      <person>
         <distributionList> cs525@cs </distributionList>
       </person>
    </receiver>
         <contents> Don't you agree this is really ugly?
</contents>

</EMail>
```

DTD

<!ELEMENT note (to,from,heading,body)>

<!ELEMENT to (#PCDATA)>

<!ELEMENT from (#PCDATA)>

<!ELEMENT heading (#PCDATA)>

<!ELEMENT body (#PCDATA)>

XML Schema

- Defines elements that can appear in a document
- Defines attributes that can appear in a document
- Defines which elements are child elements
- Defines the order of child elements
- Defines the number of child elements
- Defines whether an element is empty or can include text
- Defines data types for elements and attributes
- Defines default and fixed values for elements and attributes

XML Schema vs DTD

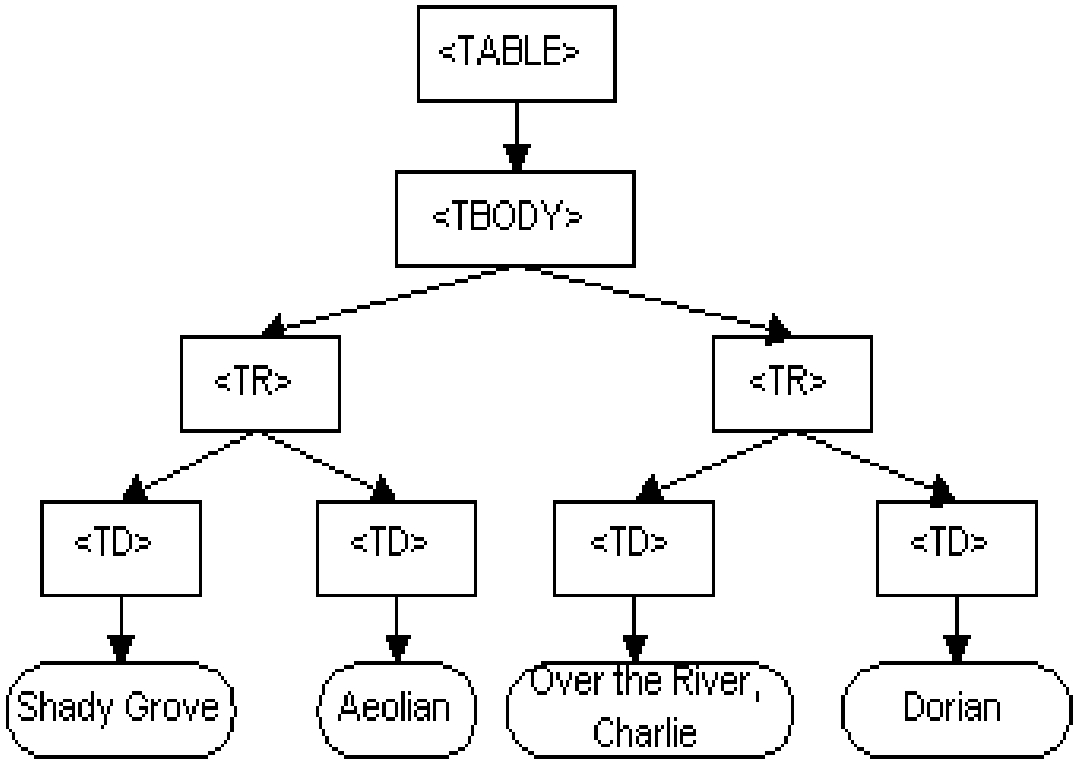
- XML Schemas are extensible to future additions
- XML Schemas are richer and more useful than DTDs
- XML Schemas are written in XML
- XML Schemas support data types
- XML Schemas support namespaces

DOM

The **D**ocument **O**bject **M**odel is a platform- and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure and style of documents.

The document can be further processed and the results of that processing can be incorporated back into the presented page.

This is an overview of DOM-related materials here at W3C and around the web.


<TABLE>
<TBODY>
<TR>
<TR>
<TD>
<TD>
<TD>
<TD>
Shady Grove
Aeolian
Over the River, Charlie
Dorian

XPATH

<?xml version="1.0" encoding="ISO-8859-1"?>
<catalog> <cd country="USA"> <title>Empire Burlesque</title> <artist>Bob Dylan</artist> <price>10.90</price> </cd>

<cd country="UK"> <title>Hide your heart</title> <artist>Bonnie Tyler</artist> <price>9.90</price> </cd>
<cd country="USA"> <title>Greatest Hits</title> <artist>Dolly Parton</artist> <price>9.90</price> </cd>
</catalog>

/catalog

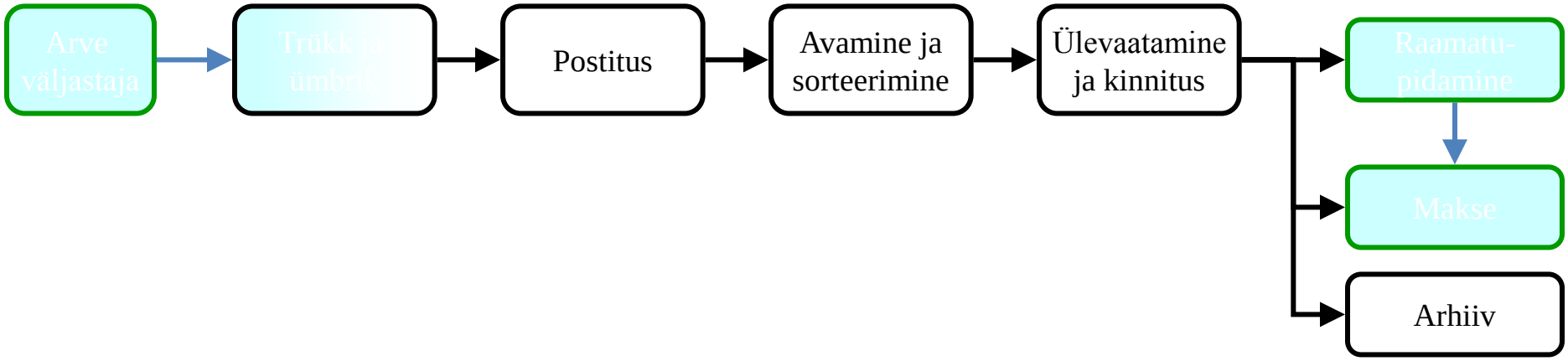
/catalog/cd[price>10.80]

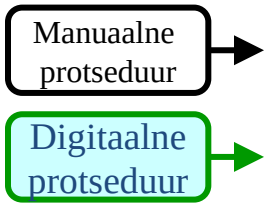
Eesti e-arve

e-invoice_ver1.1_eng.pdf

e-invoice_ver1.1.xsd

Arvete vastuvõtmise protseduurid täna

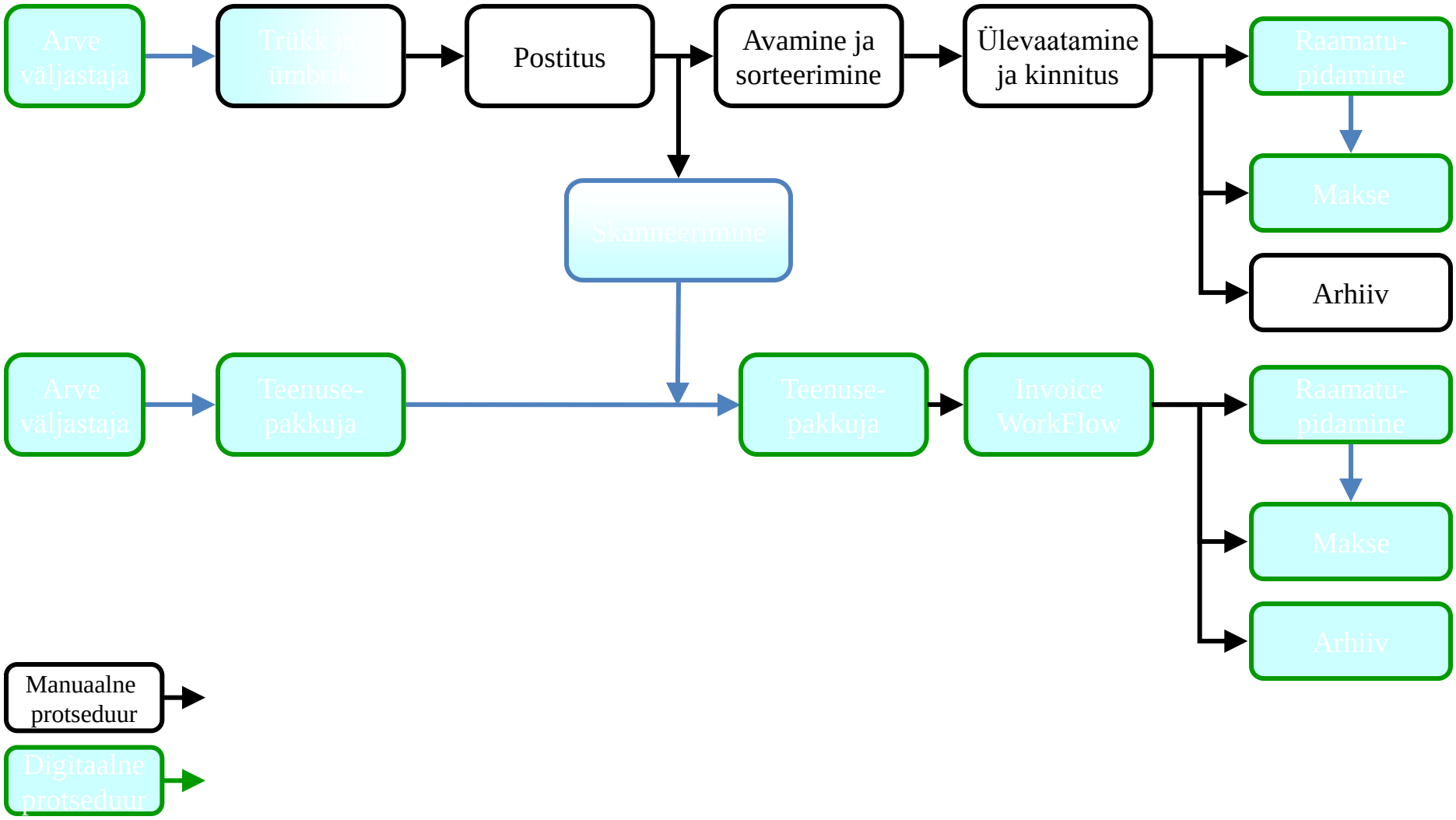




Kui palju mingi protsess aega võtab?

Protseduur	Minutit
Ümbriku avamine	1
Eeltöötlus (koopia, tempel, eelkontroll)	4
Eelkonteerimine	2
Kontroll	1
Aktsepteerimine ja kinnitus	2
Lõplik konteerimine	2
Arhiveerimine	2
Majasisene ringlus	10
Vigade käsitlus	2
Kokku	26

Arvete vastuvõtmise protseduurid homme



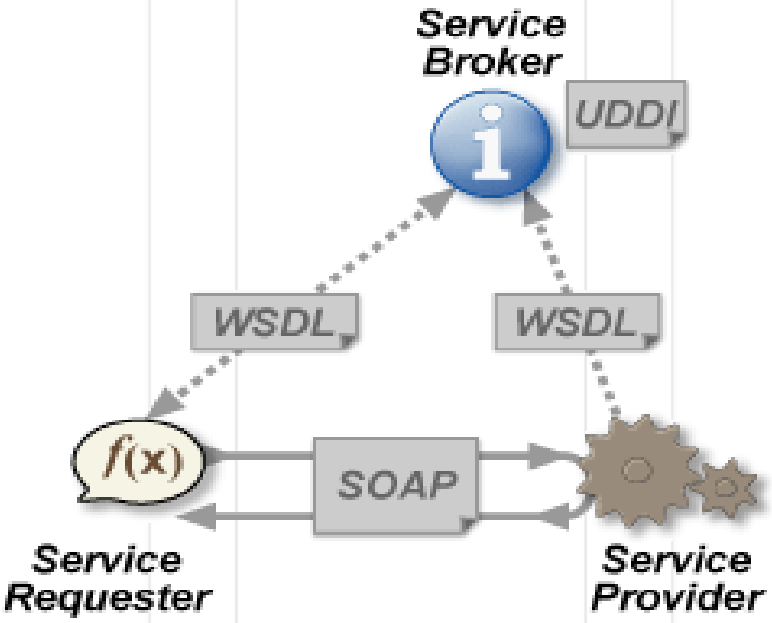
Kui palju mingi protsess aega võtab

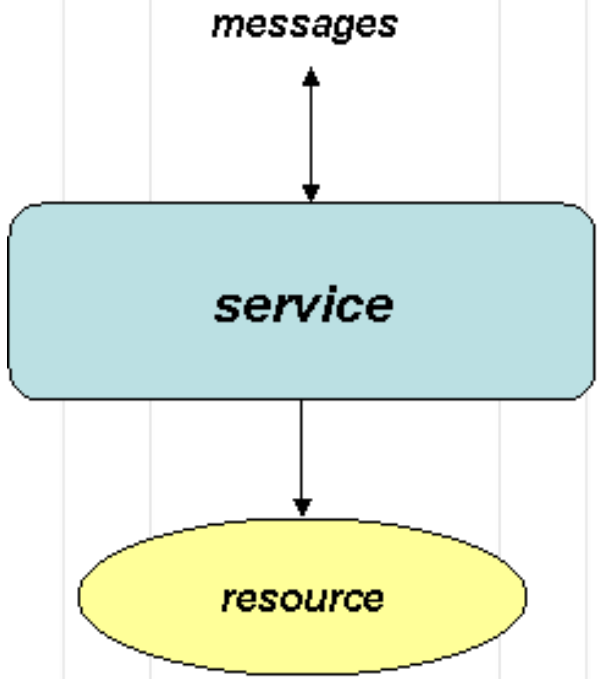
Protseduur	Manuaalselt	Digitaalselt	e-arve
Ümbriku avamine	1		
Eeltöötlus (koopia, tempel, eelkontroll)	4		
Eelkonteerimine	2		
Kontroll	1	1	1
Aktsepteerimine ja kinnitus	2	1	1
Lõplik konteerimine	2	2	
Arhiveerimine	2		
Majasisene ringlus	10		
Vigade käsitlus	2	1	1
Kokku	26	5	3

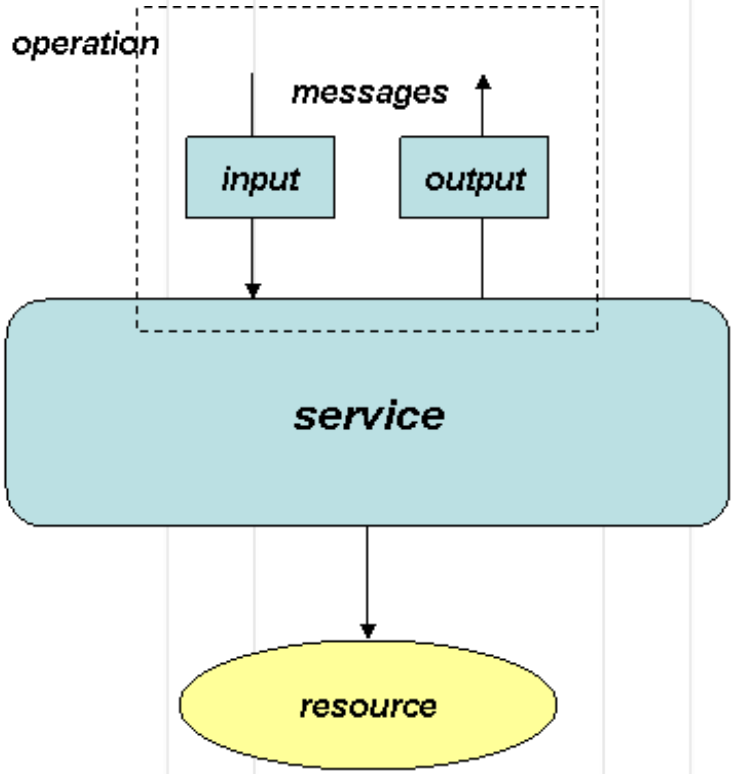
Veebiteenus

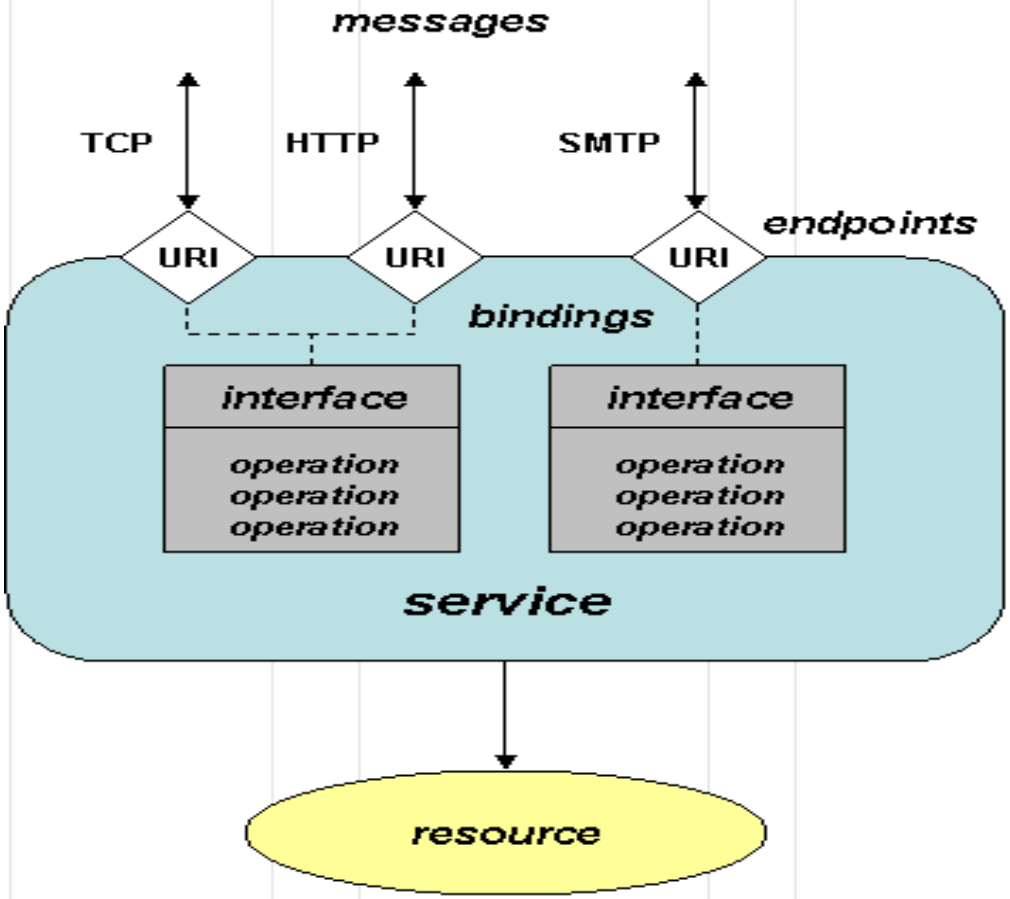
The W3C defines a "web service" as "a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format – specifically Web Services Description Language WSDL.

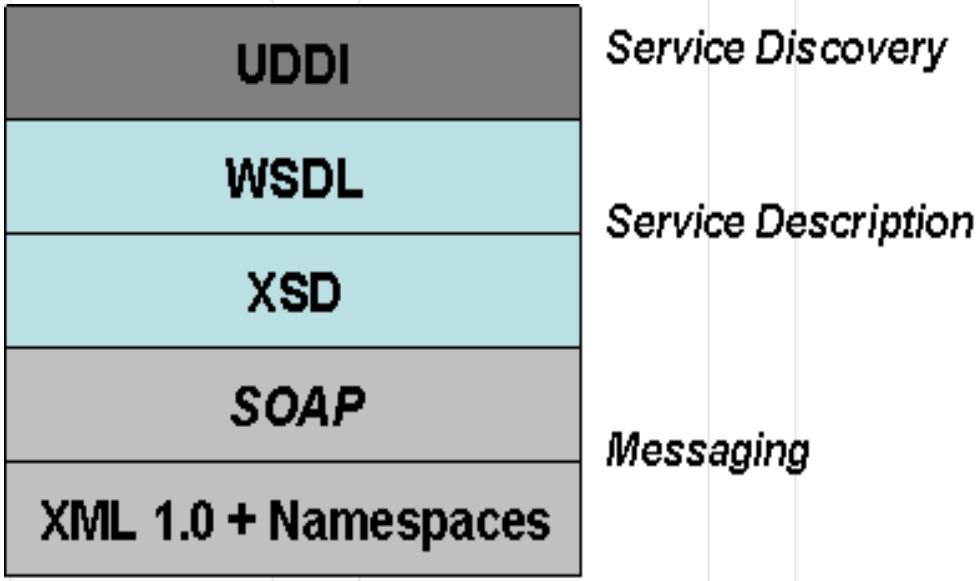
Other systems interact with the web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards"


Service
Broker
UDDI
WSDL
WSDL
f(x)
SOAP
Service
Requester
Service
Provider


messages
service
resource


operation
messages
input
output
service
resource


messages
TCP
HTTP
SMTP
endpoints
URI
URI
URI
bindings
interface
operation
operation
operation
interface
operation
operation
operation
service
resource


UDDI
WSDL
XSD
SOAP
XML 1.0 + Namespaces
Service Discovery
Service Description
Messaging

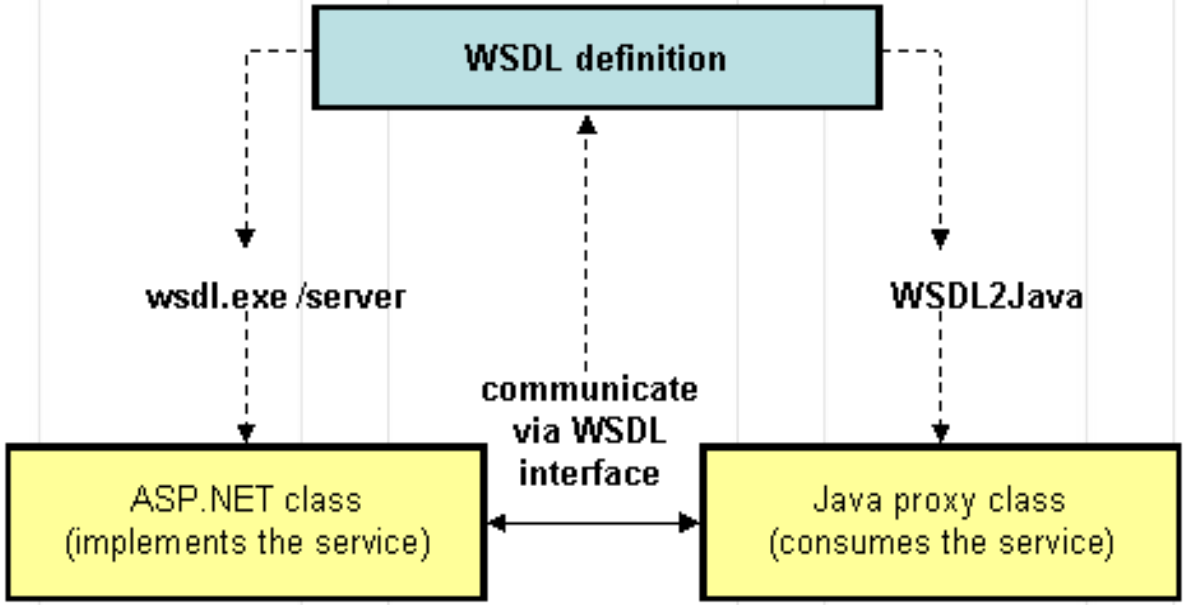
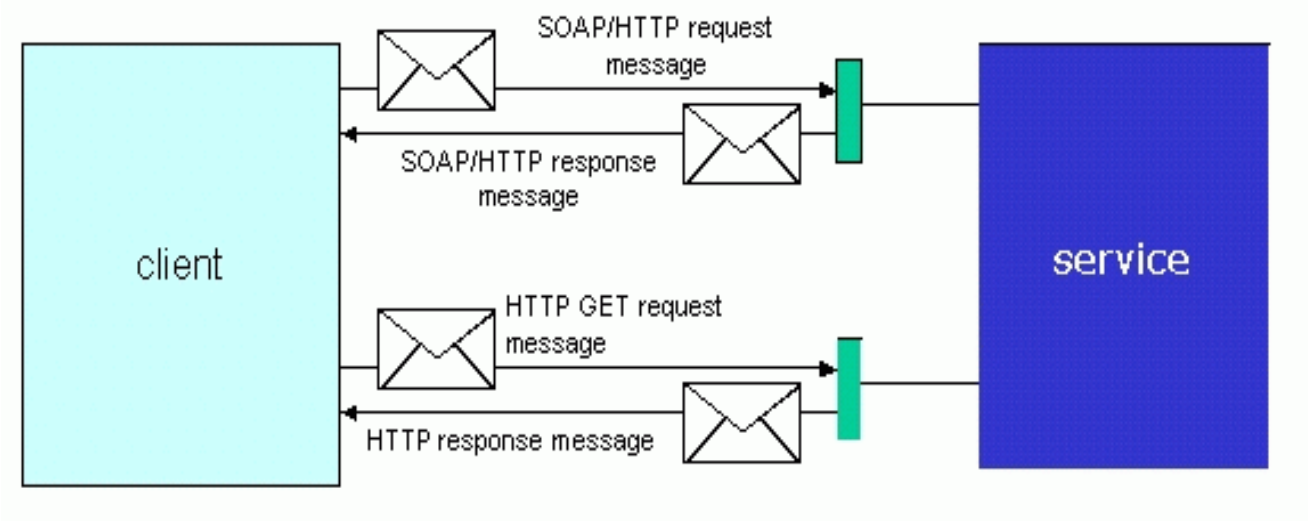

WSDL definition
wsdl.exe /server
WSDL2Java
communicate
via WSDL
interface
ASP.NET class
(implements the service)
Java proxy class
(consumes the service)

Figure 1. A client invoking a Web service



http://www.w3.org/TR/wsdl12/

WSDL Document Structure

Abstract Definitions

Types

Machine- and language-independent type definitions

Messages

Contains function parameters (inputs separate from outputs) or document descriptions

PortTypes

Refers to message definitions in Messages section to describe function signatures (operation name, input parameters, output pameters)

Concrete Descriptions

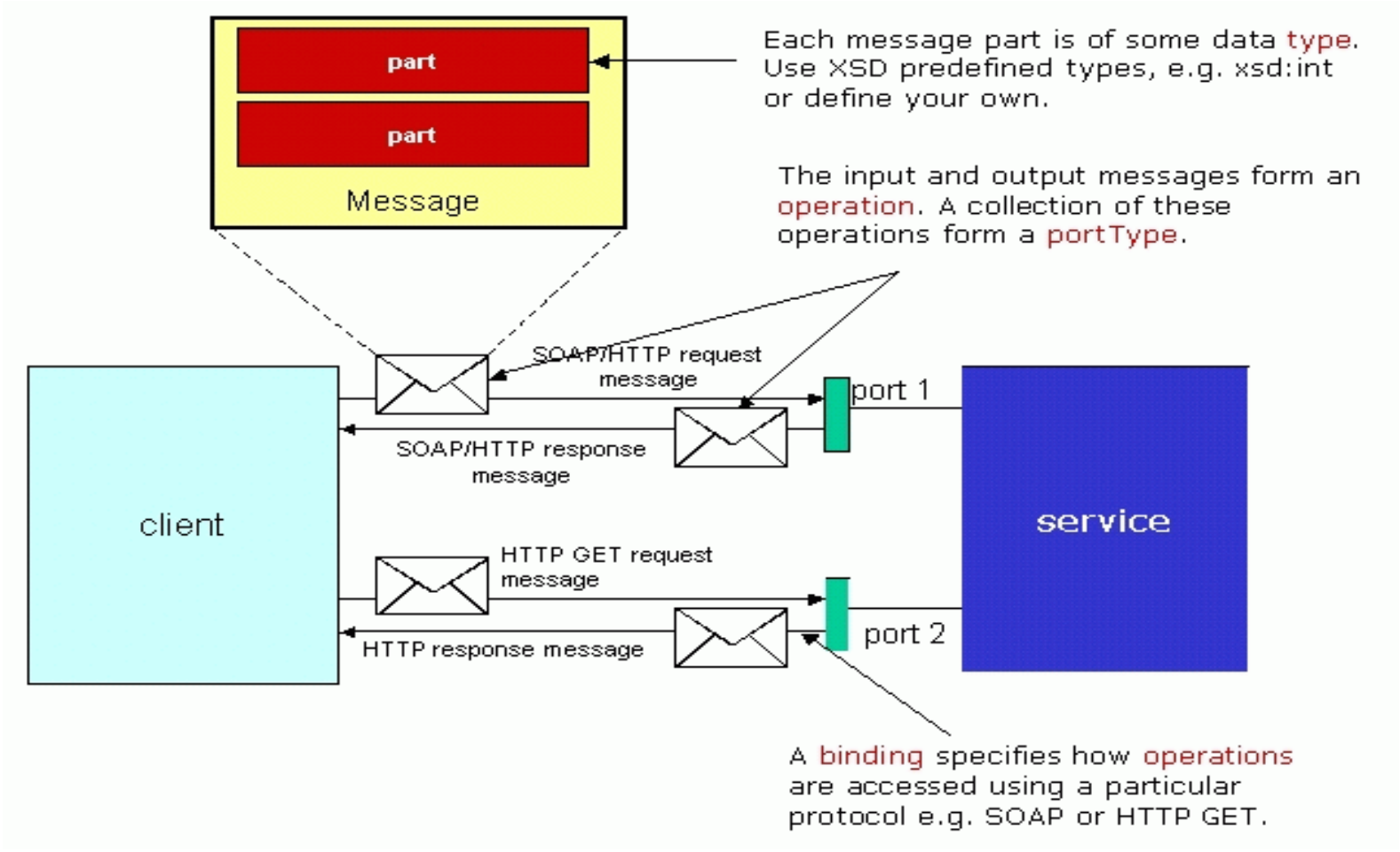
Bindings

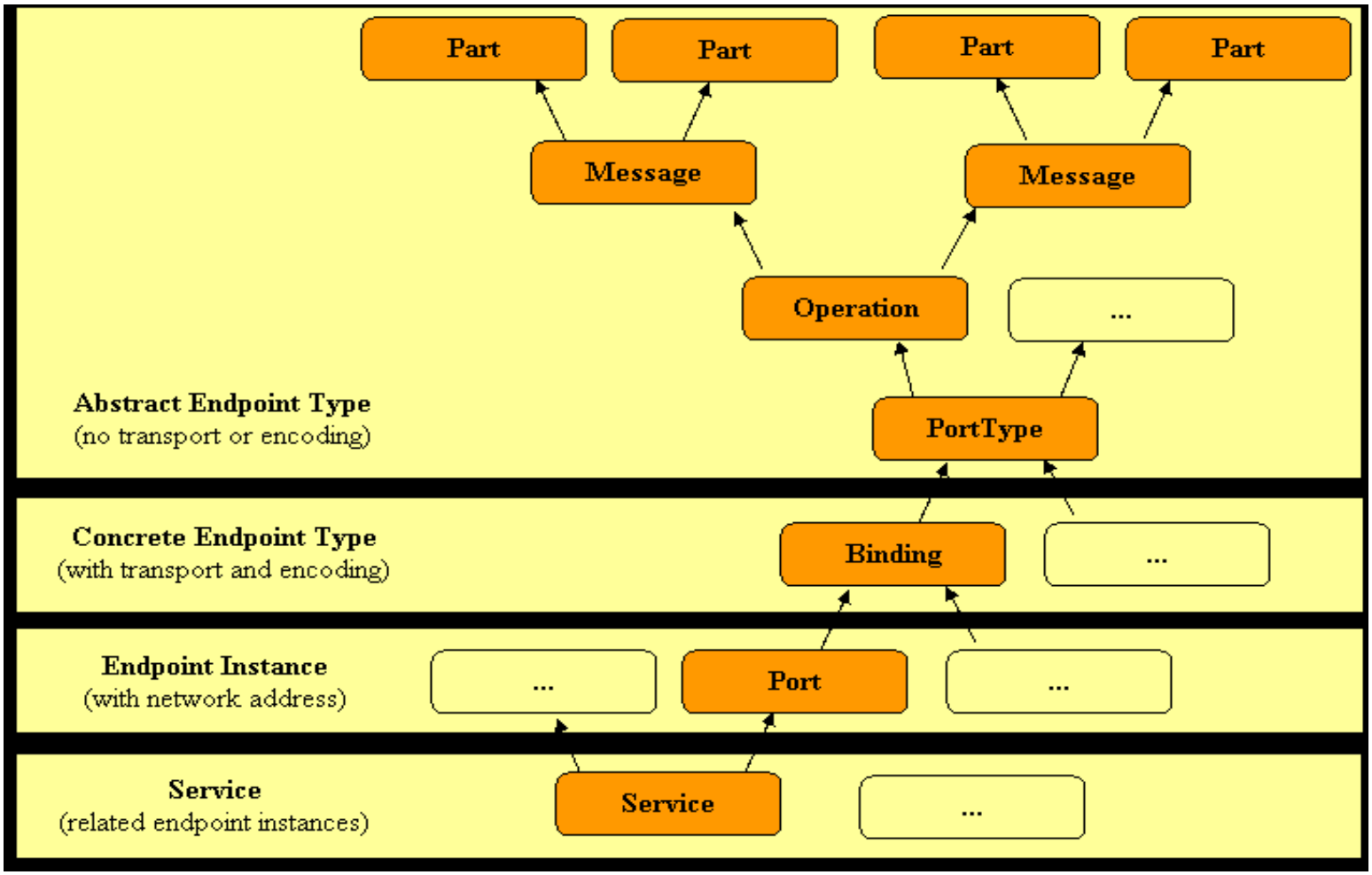
Specifies binding(s) of each operation in the PortTypes section

Services

Specifies port address(es) of each binding

WSDL terminology used for describing Web services.




Part
Part
Part
Part
Message
Message
Operation
...
Abstract Endpoint Type
(no transport or encoding)
PortType
Concrete Endpoint Type
(with transport and encoding)
Binding
...
Endpoint Instance
(with network address)
...
Port
...
Service
(related endpoint instances)
Service
...

```
<message name="getTermRequest"> <part
name="term" type="xs:string"/> </message>

<message name="getTermResponse"> <part
name="value" type="xs:string"/> </message>

<portType name="glossaryTerms">
<operation name="getTerm">
<input message="getTermRequest"/>
<output message="getTermResponse"/>
</operation>
</portType>
```

<binding type="glossaryTerms" name="b1"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" /> <operation> <soap:operation soapAction="http://example.com/getTerm"/> <input> <soap:body use="literal"/> </input> <output> <soap:body use="literal"/> </output> </operation> </binding>

Myagent WSDL

```
<definitions name="MyAgent"
      targetNamespace="http://tempuri.org/wsdl/"
      xmlns:tns="http://tempuri.org/wsdl/"
      xmlns:xsd1="http://tempuri.org/xsd/"
      xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
      xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
      xmlns="http://schemas.xmlsoap.org/wsdl/">
```

```
<types>                    <schema version="1.0"
      xmlns="http://www.w3.org/2000/10/XMLSchema"
                  targetNamespace="http://tempuri.org/xsd/"
                  elementFormDefault="qualified">
      <element name="GetComputer">
      <complexType>

<all>
      <element name="SpeedProcessor" type="int"/>
      <element name="DiskSize" type="int"/>
      <element name="RAMSize" type="int"/>
      <element name="MonitorSize" type="int"/>
      <element name="Price" type="int"/>
      <element name="OperatingSystem" type="int"/>

 </all>

       </complexType>

      </element>
```

```
<element name="GetComputerResponse">
        <complexType>
        <sequence>
        <element name="result">
                <complexType>
                <sequence>
        <element name="SpeedProcessor" type="int"/>
        <element name="DiskSize" type="int"/>
        <element name="RAMSize" type="int"/>
        <element name="MonitorSize" type="int"/>
        <element name="Price" type="int"/>
        <element name="OperatingSystem" type="int"/>
        </sequence>
        </complexType>
        </element></sequence></complexType></element>

</schema>       </types>
```

```
<message name="GetComputerRequest">
        <part name="body" element="xsd1:GetComputer"/>
</message>
<message name="GetComputerResponse">
        <part name="body" element="xsd1:GetComputerResponse"/>
</message>
<portType name="MyAgentPortType“>
        <operation name="GetComputer">
                <input message="tns:GetComputerRequest"/>
                <output essage="tns:GetComputerResponse"/>
                </operation>
</portType>
```

```
<binding name="MyAgentBinding" type="tns:MyAgentPortType">
        <soap:binding style="document"
port="http://schemas.xmlsoap.org/soap/http"/>
        <operation name="GetComputer">
        <soap:operation soapAction="http://tempuri.org/GetComputer"/>
                        <input>  <soap:body use="literal"/>
                </input>
        <output><soap:body use="literal"/>                      </output>
        </operation>
</binding>
```

```
<service name="MyAgentService">
        <documentation>Find the good computer
        </documentation>
        <port name="MyAgentPort" binding="tns:MyAgentBinding">
                <soap:address
location='http://localhost/ludovic3/Listener.asp'/>
        </port>
        </service>
</definitions>
```

Veebiteenuse näide

193.40.244.84

Data contract

```
[DataContract(Name = "Vendor", Namespace =
    "http://Seroter.BizTalkSOA/Types")]
  public class VendorType
  {
      private string vendorId;
      private string vendorName;
      private string vendorContactName;

      [DataMember(IsRequired = true, Order = 0)]
      public string VendorId
      {
          get { return vendorId; }
          set { vendorId = value; }
      }

}
```

Service contract

```
[ServiceContract(Name = "VendorService", Namespace =
    "http://Seroter.BizTalkSOA/Contracts")]
  public interface IVendorContract
  {
        [OperationContract(Name = "InsertVendor")]
          [FaultContract(typeof(InsertFaultType))]
          void InsertVendor(VendorType newVendor);
        [OperationContract(Name = "DeleteVendor")]
   [TransactionFlow(TransactionFlowOption.Allowed)]
          bool DeleteVendor(string vendorId);
  }
```


Questions?