

IT ARCHITECTURE

WEB SERVICE

Willy Sudiarto Raharjo

Teknik Informatika
Universitas Kristen Duta Wacana



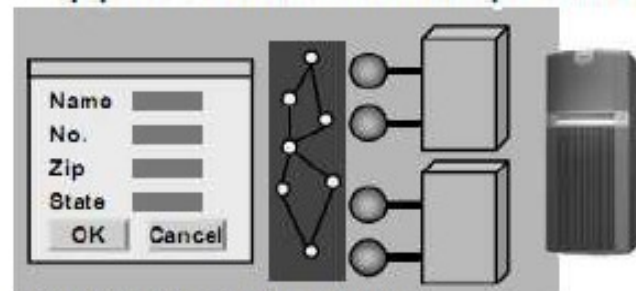
Customer



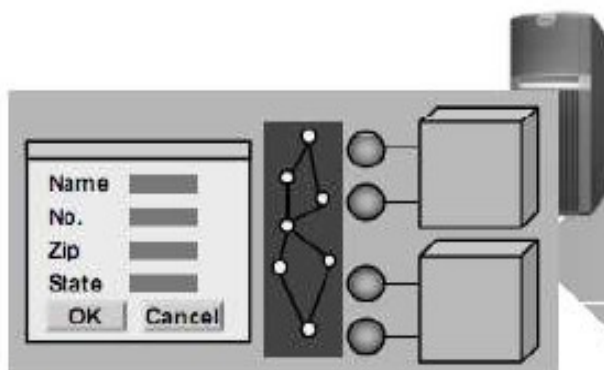
View in browser

Typical ASP

Application solution provider



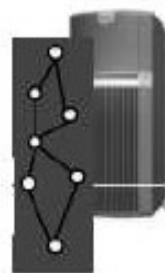
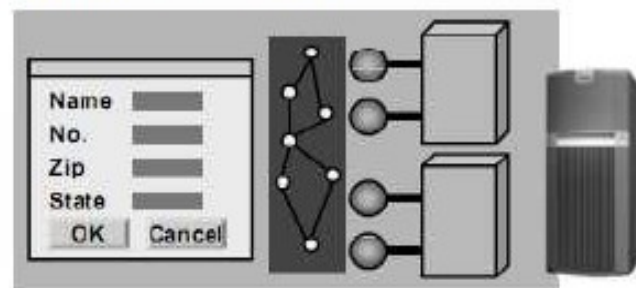
Application runs here



Application runs here

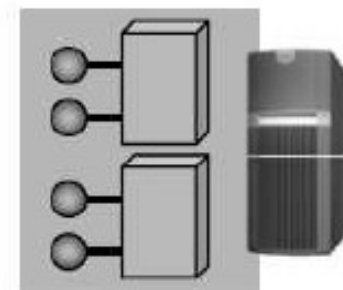
Download on demand

Software as a Service



Access service

Web services

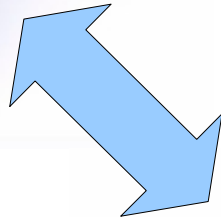


Application runs partly here
(composition, repackaging, differentiation,
value-added services)

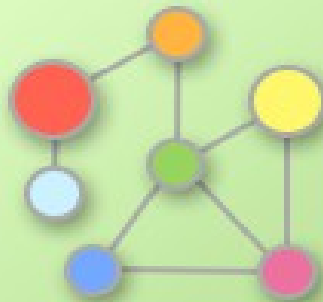
Application runs partly
here

Definition

- **Generic:** any application accessible to other applications over the Web
 - Very open definition, almost any URL becomes a “Web service” under this definition
- **UDDI consortium:** web services are self-contained, modular business applications that have open, Internet-oriented, standards-based interfaces
 - Better, but still not specific enough
- **W3C:** A Web service is a software application identified by a URI, whose interfaces and bindings are capable of being defined, described, and discovered as XML.
 - A Web service supports direct interactions with other software agents using XML-based messages exchanged via Internet-based protocols
 - Slightly better



Web-Oriented Architecture



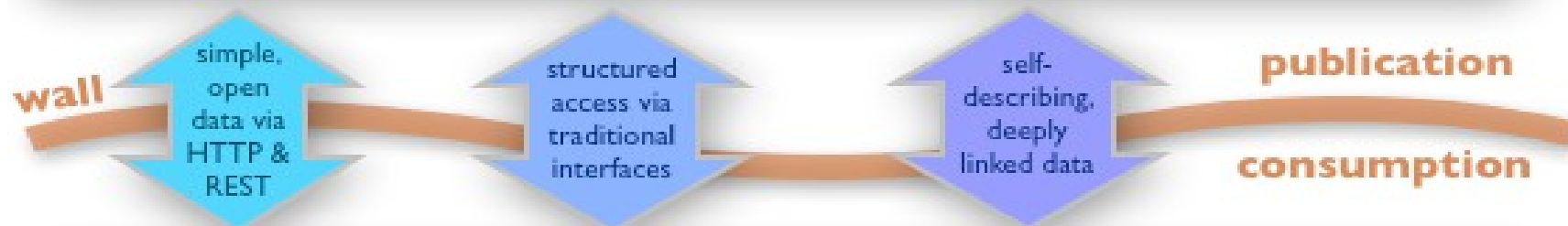
Open APIs incl. SOA



Semantic Web & Linked Data



Data
enterprise or Web



Search

Composite Apps & Mashups

SOA

BPM

Usage
internal, VPN, or Web access

Example: Google Search

My Desktop
Windows

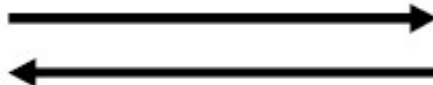
```
MyService s =  
  new  
  com.google.api.GoogleSearchService();
```



Implementation via
proxy class and
HTTP transport

The Internet
TCP/IP

**SOAP
Request**



**SOAP
Response**

Vendor-neutral
XML-encoding
over HTTP

Google.com/apis
Unix/Linux?

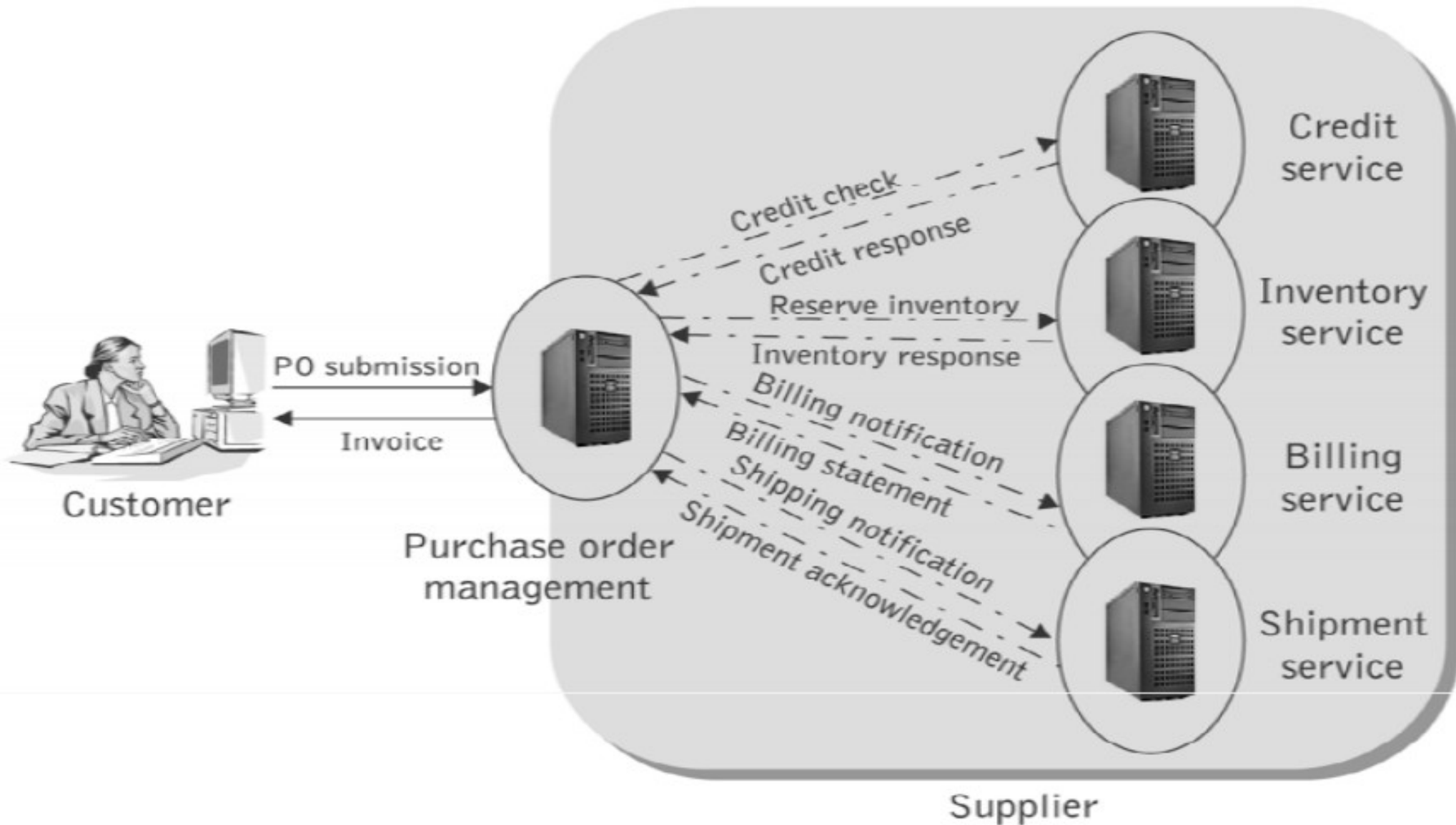
```
[WebMethod]  
... doGoogleSearch(myKey, q) ...
```



Implementation via
WebService classes
in Web Server



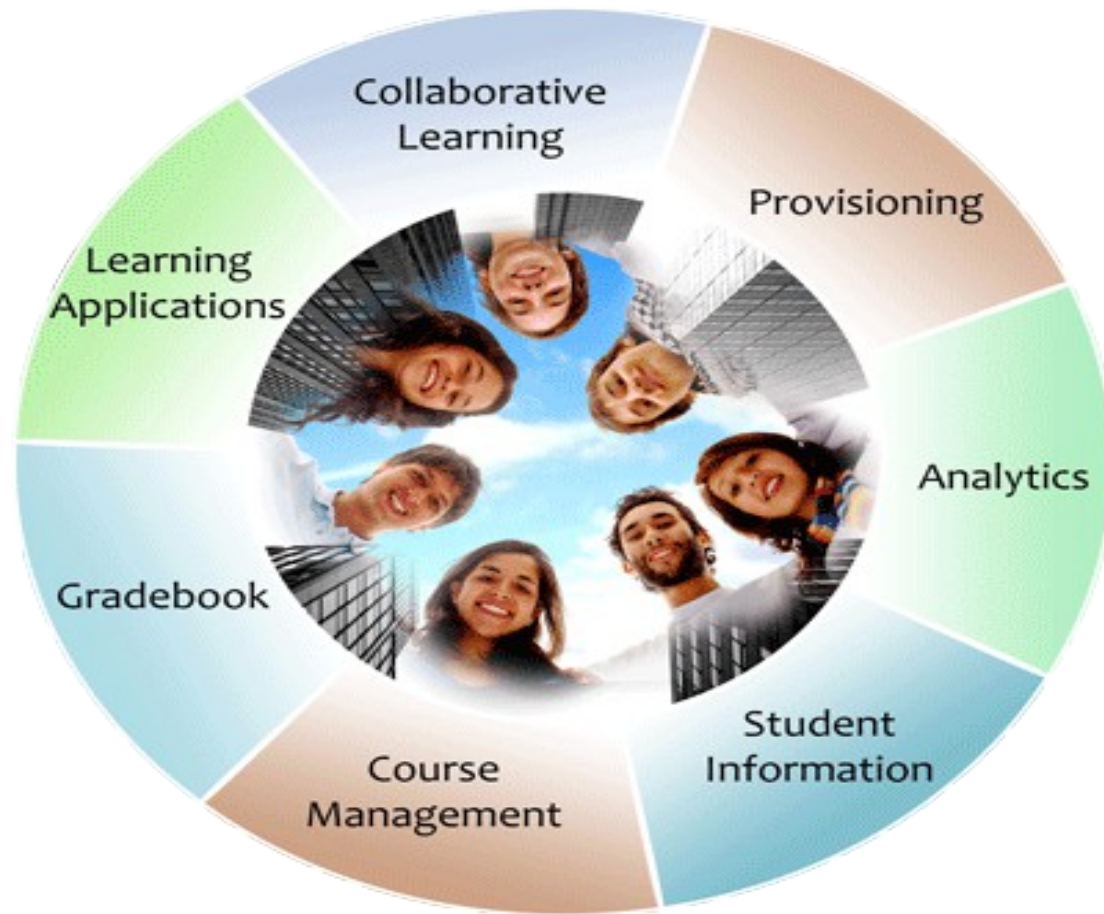
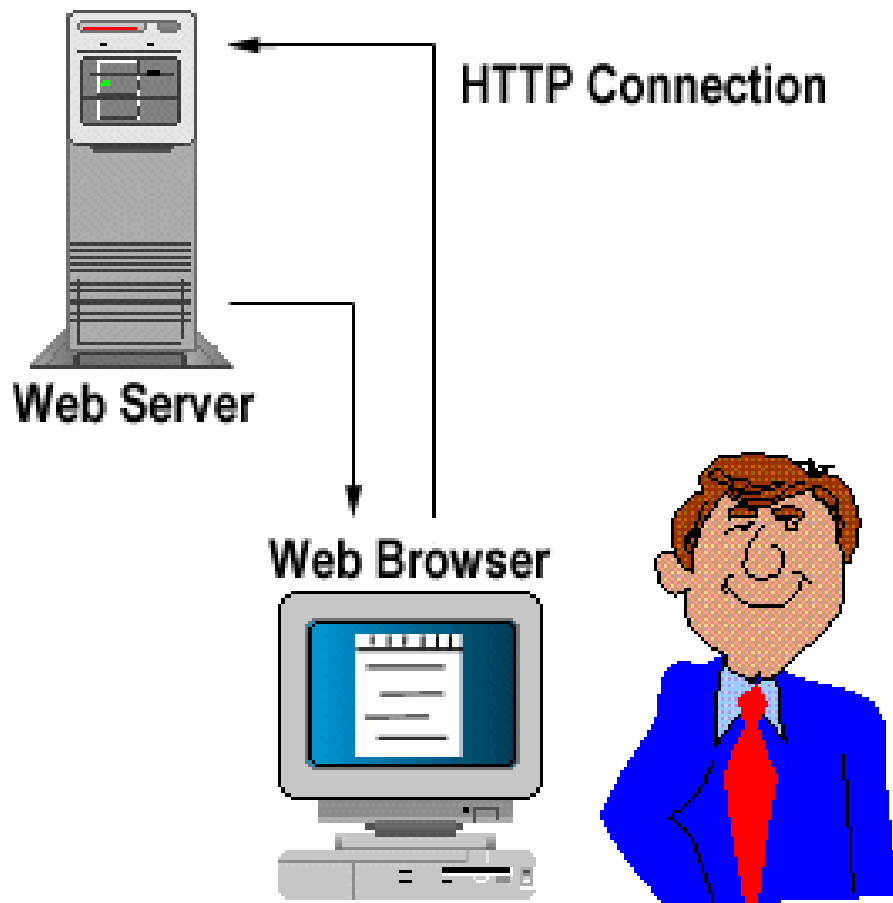
Example in Enterprise

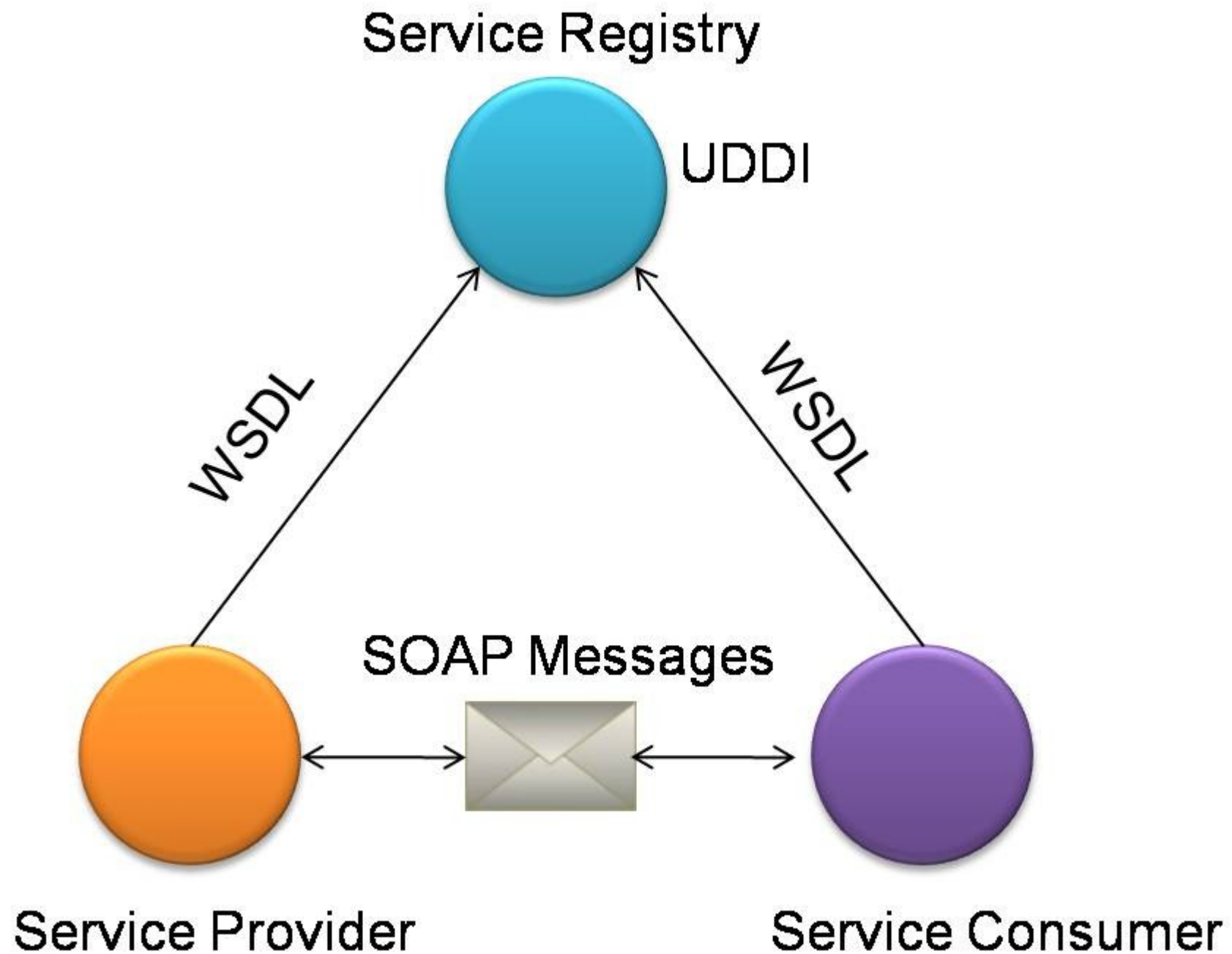


Web Service

- Exchange data using XML
- Available and communicate through Internet/Intranet
- Platform/programming language independent
- UI-less Web application
- Technologies:
 - XML-RPC
 - SOAP (Simple Object Access Protocol)
 - REST

Website vs Web Service





Web Service Properties

■ Self-describing

- Human-readable documentation
- Public interface

```
-<portType name="perpus-ukdwPortType">  
  -<operation name="getBuku">  
    <documentation>Ambil Informasi Buku Perpustakaan UKDW (C) 2012</documentation>  
    <input message="tns:getBukuRequest"/>  
    <output message="tns:getBukuResponse"/>  
  </operation>
```

■ Discoverable

```
-<service name="perpus-ukdw">  
  -<port name="perpus-ukdwPort" binding="tns:perpus-ukdwBinding">  
    <soap:address location="http://library.ukdw.ac.id/ws/index.php"/>  
  </port>  
</service>
```

Service Directory:

UDDI

Service Description:

WSDL

Service Interaction:

SOAP

Format Description:

XML Schema

Data Format:

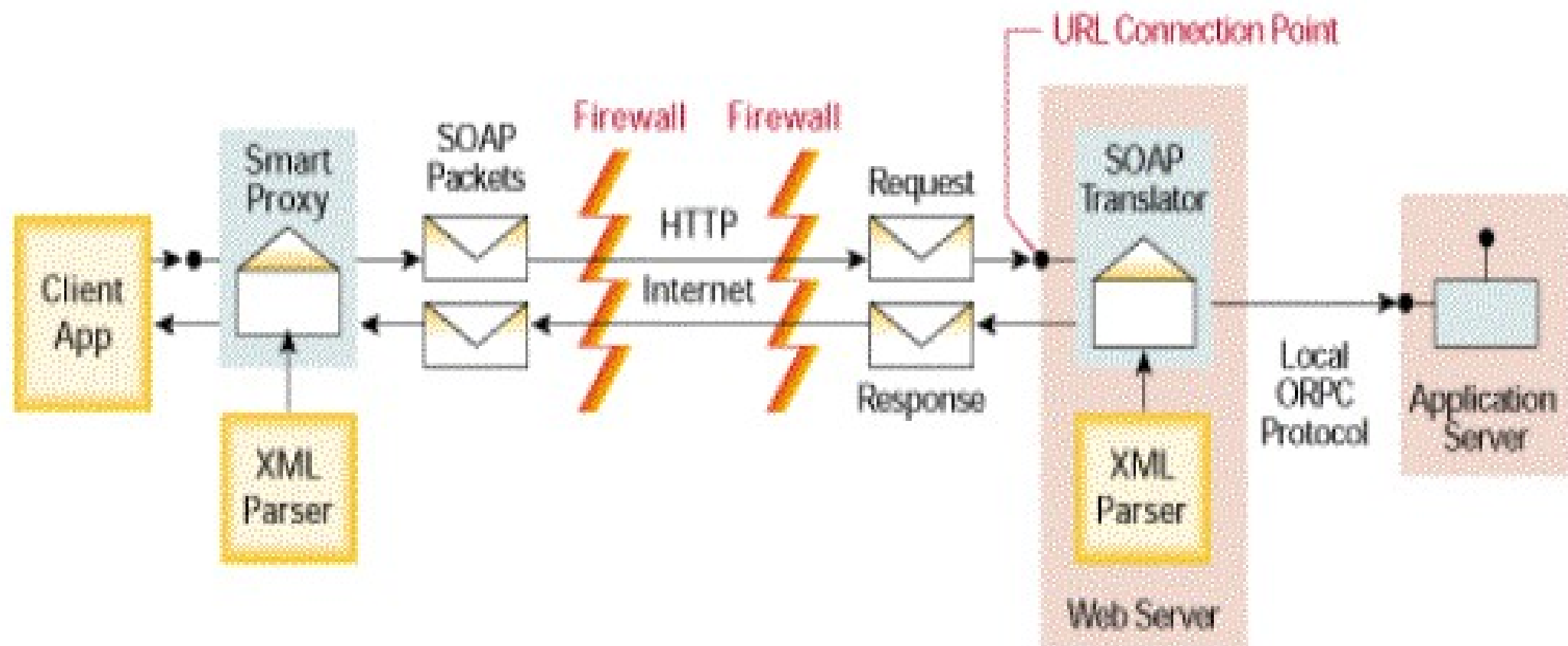
XML

Communication Protocol:

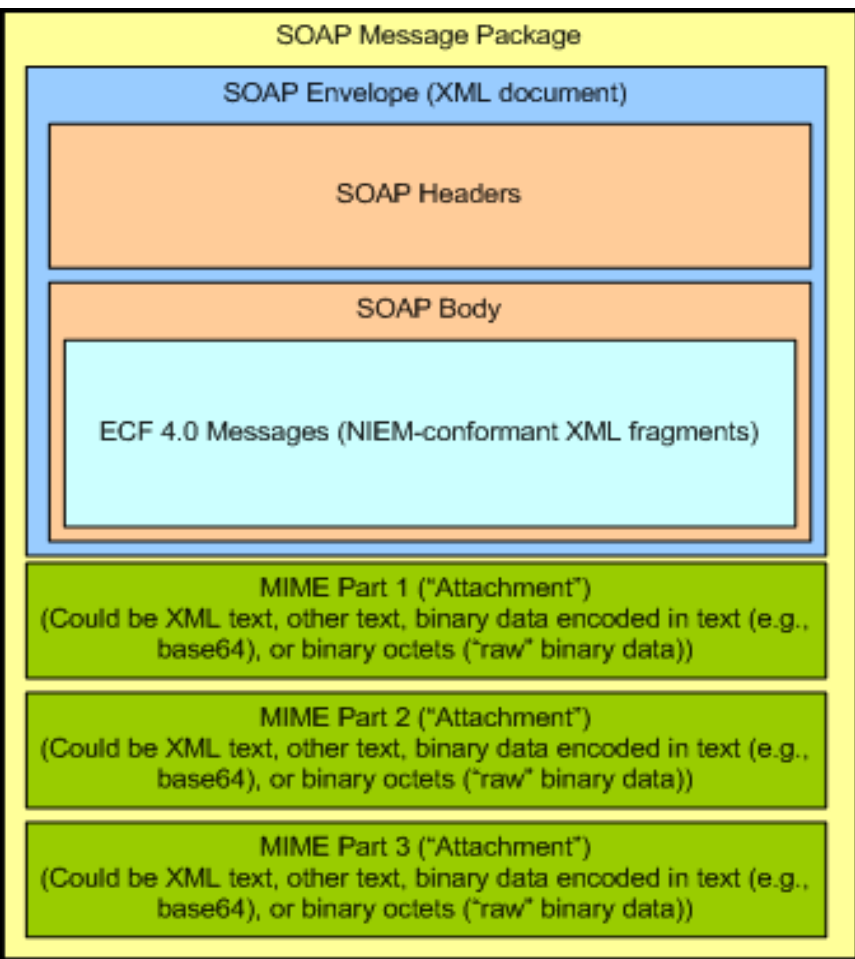
HTTP

Communication Network:

Internet



SOAP Specs



```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

  <soap:Header>
    ...
  </soap:Header>

  <soap:Body>
    ...
    <soap:Fault>
      ...
    </soap:Fault>
  </soap:Body>

</soap:Envelope>
```

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<SOAP-ENV:Envelope
```

```
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
```

```
  xmlns:ns1="urn:testns"
```

```
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

```
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
```

```
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
```

```
    <SOAP-ENV:Body>
```

```
      <ns1:getProfessor>
```

```
        <course>470</course>
```

```
      </ns1:getProf>
```

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

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

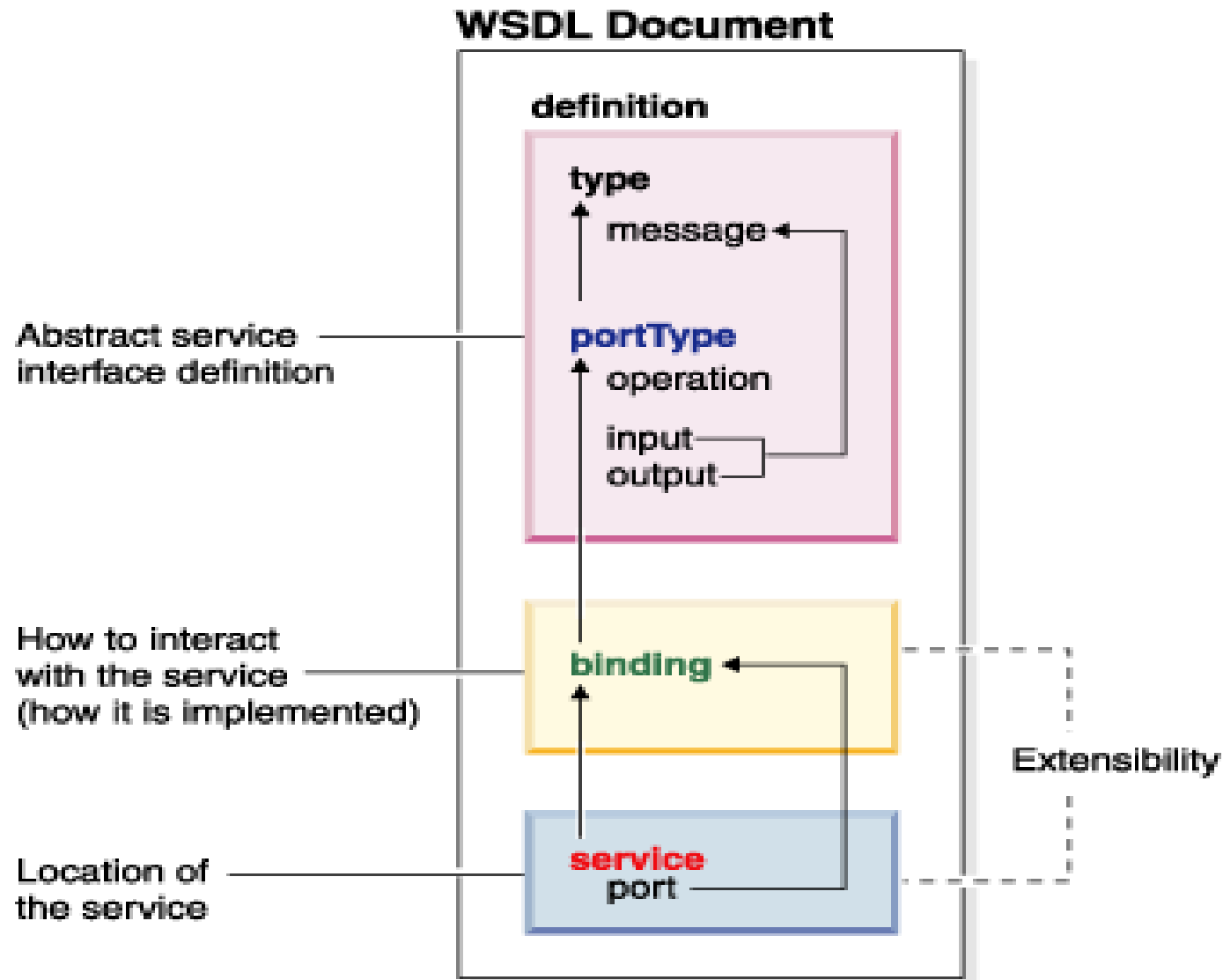
SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns1="urn:testns"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">

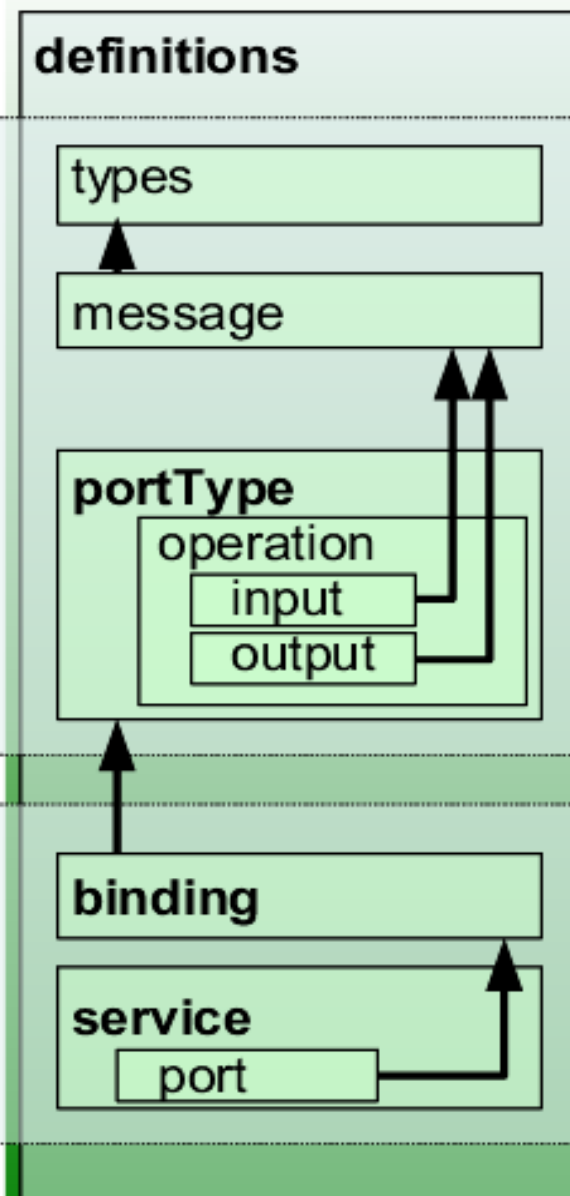
  <SOAP-ENV:Body>
    <ns1:getResponse>
      <Result> WILLY </Result>
    </ns1:getResponse>
  </SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```

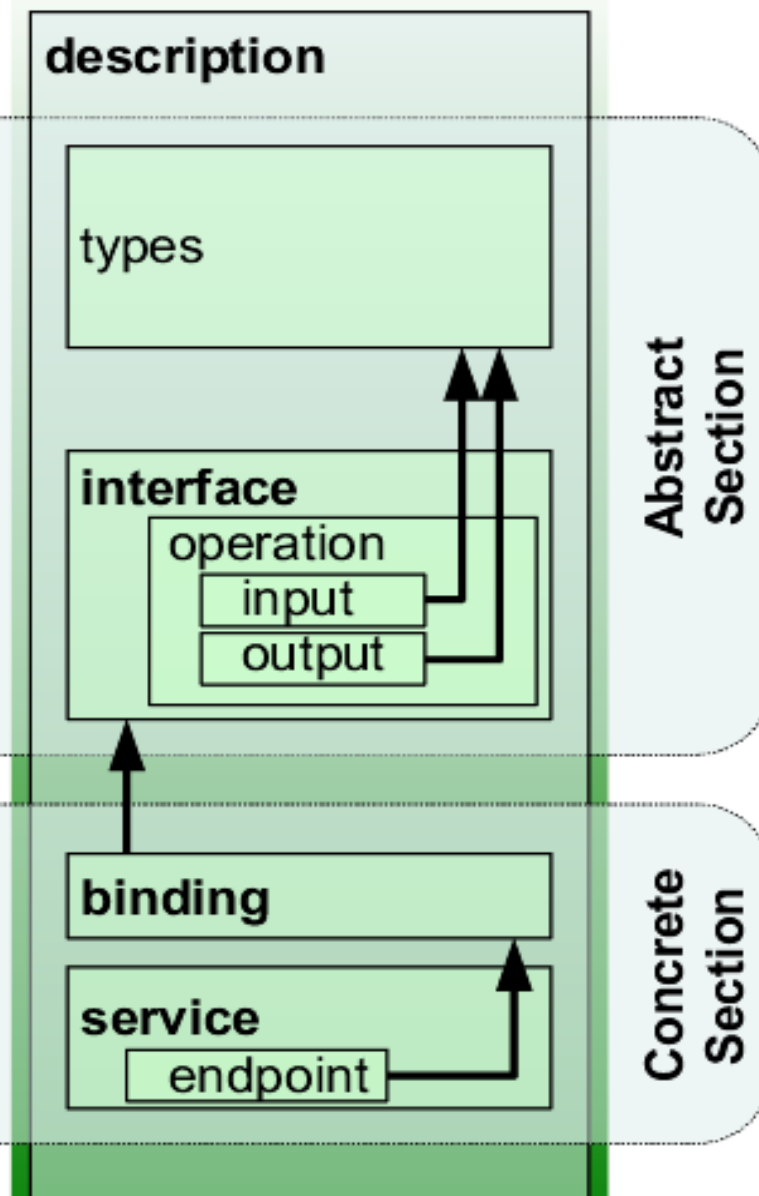
WSDL Structure



WSDL 1.1



WSDL 2.0



```

<!-- 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:complexType>
        <xs:sequence>
          <xs:element name="header" maxOccurs="unbounded">
            <xs:complexType>
              <xs:simpleContent>
                <xs:extension base="xs:string">
                  <xs:attribute name="name" type="xs:string" use="required"/>
                </xs:extension>
              </xs:simpleContent>
            </xs:complexType>
          </xs:element>
          <xs:element name="body" type="xs:anyType" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="method" type="xs:string" use="required"/>
        <xs:attribute name="uri" type="xs:anyURI" use="required"/>
      </xs:complexType>
    </xs:element>

    <xs:element name="response">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="header" maxOccurs="unbounded">
            <xs:complexType>
              <xs:simpleContent>
                <xs:extension base="xs:string">
                  <xs:attribute name="name" use="required"/>
                </xs:extension>
              </xs:simpleContent>
            </xs:complexType>
          </xs:element>
          <xs:element name="body" type="xs:anyType" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="status-code" type="xs:anySimpleType" use="required"/>
        <xs:attribute name="response-phrase" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:schema>
</types>

```

```

<!-- Abstract interfaces -->
<interface name="RESTfulInterface">
  <fault name="ClientError" element="tns:response"/>
  <fault name="ServerError" element="tns:response"/>
  <fault name="Redirection" element="tns:response"/>
  <operation name="Get" pattern="http://www.w3.org/ns/wsd/in-out">
    <input messageLabel="In" element="tns:request"/>
    <output messageLabel="Out" element="tns:response"/>
  </operation>
  <operation name="Post" pattern="http://www.w3.org/ns/wsd/in-out">
    <input messageLabel="In" element="tns:request"/>
    <output messageLabel="Out" element="tns:response"/>
  </operation>
  <operation name="Put" pattern="http://www.w3.org/ns/wsd/in-out">
    <input messageLabel="In" element="tns:request"/>
    <output messageLabel="Out" element="tns:response"/>
  </operation>
  <operation name="Delete" pattern="http://www.w3.org/ns/wsd/in-out">
    <input messageLabel="In" element="tns:request"/>
    <output messageLabel="Out" element="tns:response"/>
  </operation>
</interface>

```

```

<!-- Concrete Binding Over HTTP -->
<binding name="RESTfulInterfaceHttpBinding" interface="tns:RESTfulInterface"
  type="http://www.w3.org/ns/wsd/http">
  <operation ref="tns:Get" whttp:method="GET"/>
  <operation ref="tns:Post" whttp:method="POST"
    whttp:inputSerialization="application/x-www-form-urlencoded"/>
  <operation ref="tns:Put" whttp:method="PUT"
    whttp:inputSerialization="application/x-www-form-urlencoded"/>
  <operation ref="tns:Delete" whttp:method="DELETE"/>
</binding>

<!-- Concrete Binding with SOAP -->
<binding name="RESTfulInterfaceSoapBinding" interface="tns:RESTfulInterface"
  type="http://www.w3.org/ns/wsd/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" />
  <operation ref="tns:Post" />
  <operation ref="tns:Put" />
  <operation ref="tns:Delete" />
</binding>

```

WSDL OPENING

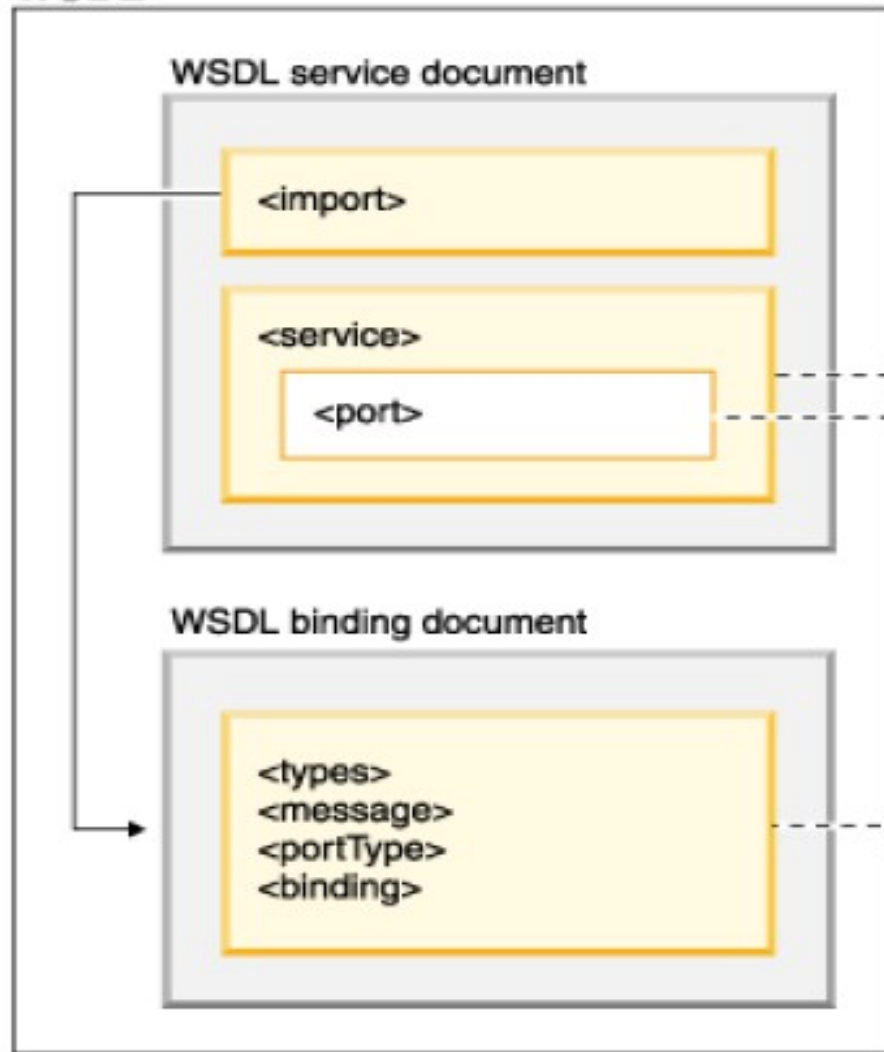
```
<?xml version="1.0" encoding="UTF-8"?>
<description xmlns="http://www.w3.org/ns/wSDL"
  xmlns:tns="http://www.tmsws.com/wSDL20sample"
  xmlns:http="http://schemas.xmlsoap.org/wSDL/http/"
  xmlns:soap="http://schemas.xmlsoap.org/wSDL/soap/"
  targetNamespace="http://www.tmsws.com/wSDL20sample">
```

WSDL CLOSING

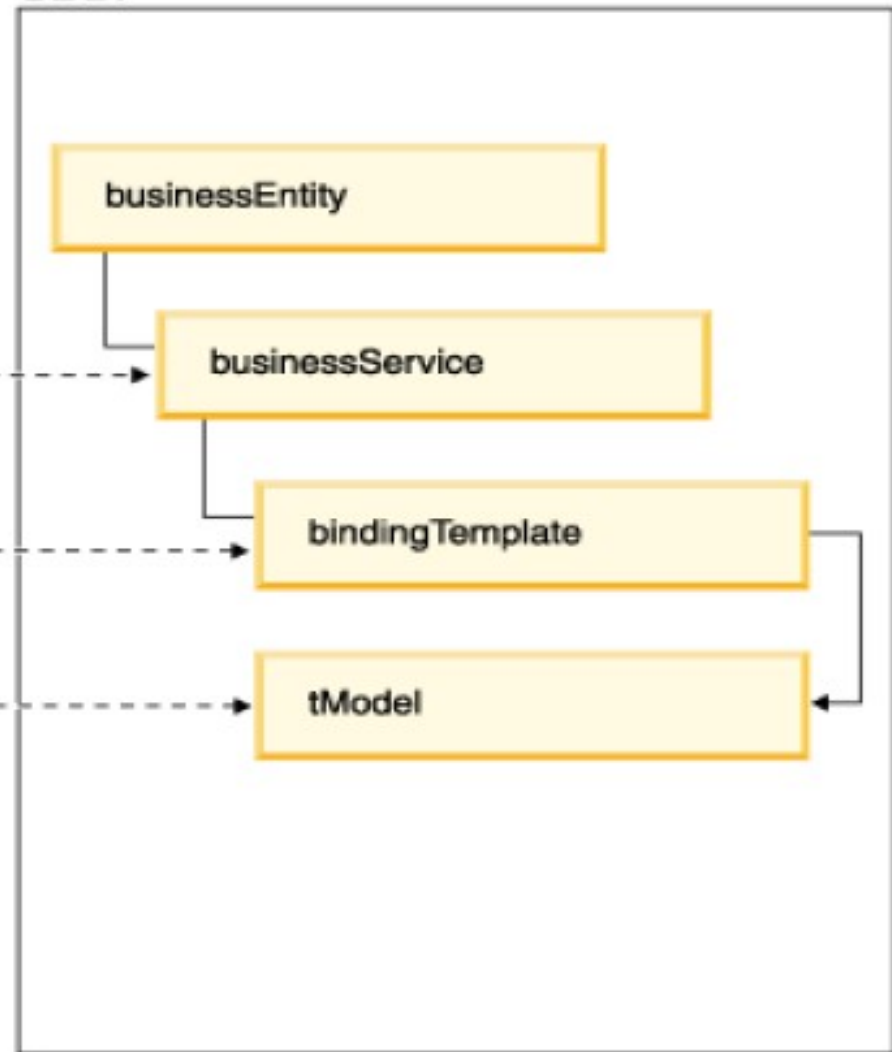
```
<!-- Web Service offering endpoints for both bindings-->
  <service name="RESTfulService" interface="tns:RESTfulInterface">
    <endpoint name="RESTfulServiceHttpEndpoint"
      binding="tns:RESTfulInterfaceHttpBinding"
      address="http://www.example.com/rest/" />
    <endpoint name="RESTfulServiceSoapEndpoint"
      binding="tns:RESTfulInterfaceSoapBinding"
      address="http://www.example.com/soap/" />
  </service>
</description>
```

WSDL – UDDI Relationship

WSDL

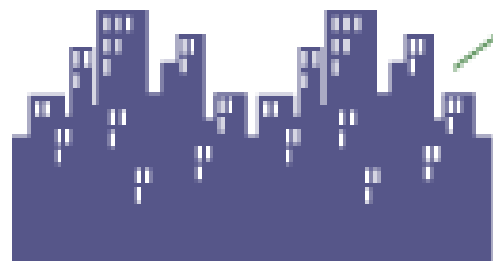


UDDI

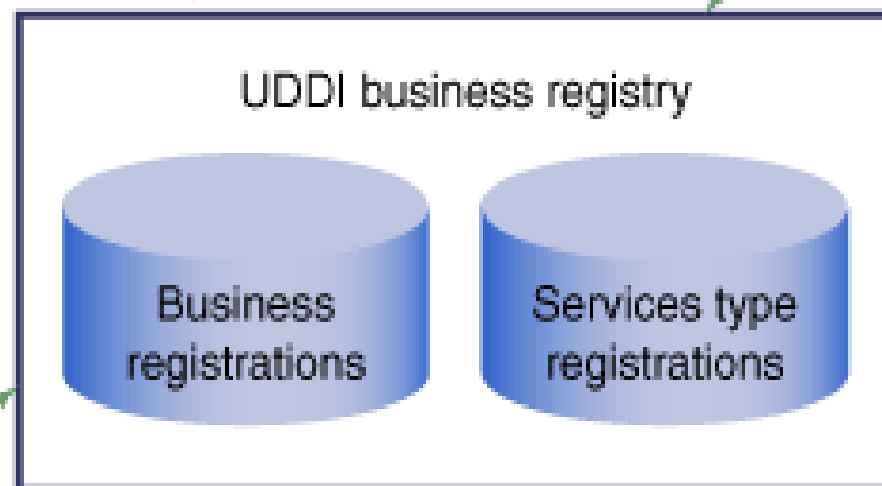




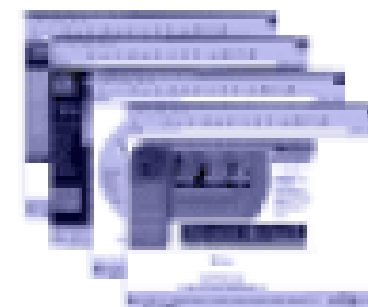
1. Software companies, standards bodies, and programmers populate the registry with descriptions of different tModels



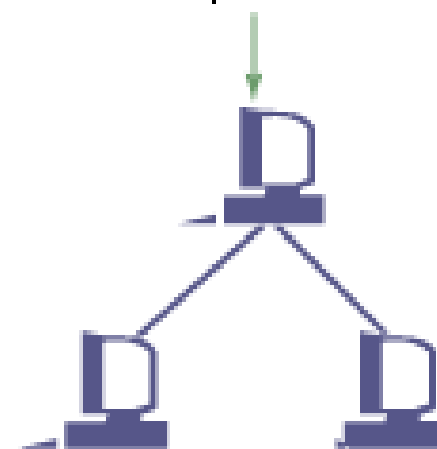
2. Businesses populate the registry with descriptions of the services they support



3. UDDI assigns a programmatically unique identifier (UUID) to each tModel and business registration and stores an Internet registry



4. Marketplaces, search engines, and business apps query the registry to discover services at other companies



5. Businesses use this data to facilitate easier integration with each other over the Web

SIMPLE DEMO

Service:

<http://library.ukdw.ac.id/ws/>

<http://lecturer.ukdw.ac.id/yuan/labs/quotes/>

Client:

<http://ti.ukdw.ac.id/~willysr/ati-ti/soap.php>

<http://ti.ukdw.ac.id/~willysr/ati-ti/rest.php>