



# ICT336 Internet Systems Programming

---

## SOAP, WSDL and UDDI

(Week 10 Lecture 2)



# Learning Objectives

---

- Learn more details of the three components of Web Services.



# Lecture Outline

---

- Components of Web Services:
  - SOAP
  - WSDL
  - UDDI



# Components of Web Services

---

- In Web Services, we
  - describe a service using the ***Web Services Description Language*** (WSDL).
  - define a way to publish and discover information about Web services using ***Universal Description, Discovery and Integration*** (UDDI), and
  - invoke the service using ***Simple Object Access Protocol*** (SOAP).



# Web Service Component: SOAP

---

- Simple Object Access Protocol (SOAP) is an XML-based messaging and remote procedure call specification that enables the exchange of information among distributed systems.
- SOAP is a public standard defined by W3C.



# SOAP Communication

---

- In SOAP communication, the parties involved are called **SOAP nodes**.
- Nodes send **SOAP messages** to each other, in a one-way transmission from a **sender** to a **receiver**.
- There may be **intermediaries** between the sender and receiver.



# SOAP Communication

---

- SOAP communication combines this underlying message transmission to implement complex interactions, from
  - Single message with no response, to ...
  - Single request-response exchange, to ...
  - Multiple back-and-forth "conversations".
- SOAP 1.1 defined 4 explicit client/server request/response-type interactions.
- SOAP 1.2 (current version) tries to move away from this terminology, to more general message exchange models of any type.



# Single Request-Response

---

- Because a lot of SOAP services today depend on HTTP, we see a lot of these services following the simple single request-response exchange.
- Our lab exercises involve only services of this type.
  - Client ask for something - server sends something back - end of exchange.
  - The version of SOAP supported by SOAP::Lite we use in our script is version 1.1.



# Example SOAP Service


---

- Let's look at a hypothetical service that allows us to query information about a University unit.
- What would requests and responses from a client to a server in this service look like?



# Example SOAP Request

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://www.w3.org/2001/09/soap-envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:getUnitInfo
      xmlns:ns="urn:examples:unitinfoservice"
      SOAP-ENV:encodingStyle=" http://www.w3.org/2001/09/soap-encoding">
      <unit_code xsi:type="xsd:string">B336</unit_code>
    </ns1:getUnitInfo>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



# Example SOAP Request (in essence)

---

```
<Envelope>  
  <Body>  
    <getUnitInfo>  
      <unit_code>B336</unit_code>  
    </getUnitInfo>  
  </Body>  
</Envelope>
```



# Example SOAP Response

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://www.w3.org/2001/09/soap-envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:getUnitInfoResponse
      xmlns:ns1="urn:examples:weatherservice"
      SOAP-ENV:encodingStyle="http://www.w3.org/2001/09/soap-encoding">
      <unit_name xsi:type="xsd:string">Internet Systems Programming</unit_name>
    </ns1:getUnitInfoResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



# Example SOAP Response (in essence)

---

```
<Envelope>  
  <Body>  
    <getUnitInfoResponse>  
      <unit_name>Internet Systems Programming</unit_name>  
    </getUnitInfoResponse>  
  </Body>  
</Envelope>
```

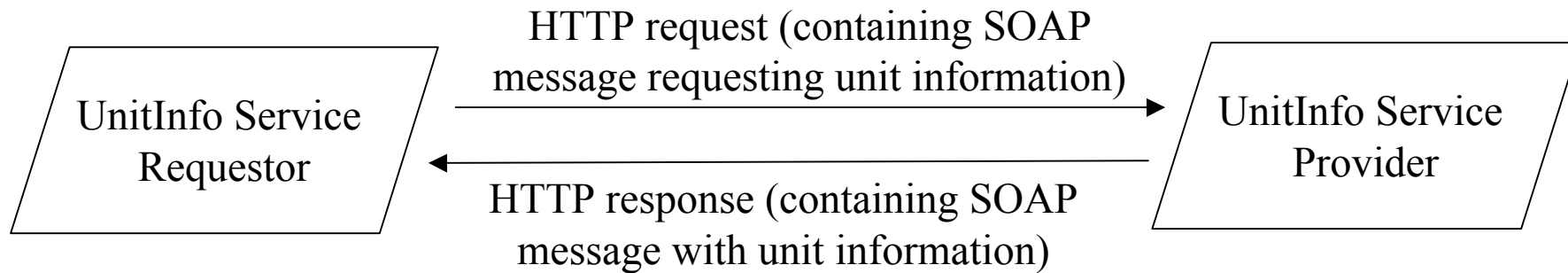


# Binding to Different Transport Protocols

---

- SOAP does not define how the messages are to be transported.
- SOAP must **bind** to certain messaging protocols (eg. HTTP, SMTP) to transport the messages.
- It is up to the implementor of the SOAP service to decide which protocol can support their service interaction best.

# Example: SOAP over HTTP



# Example SOAP Request in HTTP Message

```
POST /ws/UnitInfoService HTTP/1.1
Host: 134.115.64.2
Content-type: text/xml
Content-length: 300
SOAPMethodName: urn:examples:unitinfoservice#getUnitInfo
```

HTTP Message Headers

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://www.w3.org/2001/09/soap-envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:getUnitInfo
      xmlns:ns="urn:examples:unitinfoservice"
      SOAP-ENV:encodingStyle=" http://www.w3.org/2001/09/soap-encoding">
      <unit_code xsi:type="xsd:string">B336</unit_code>
    </ns1:getUnitInfo>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

HTTP Message Body  
(our SOAP request from before)

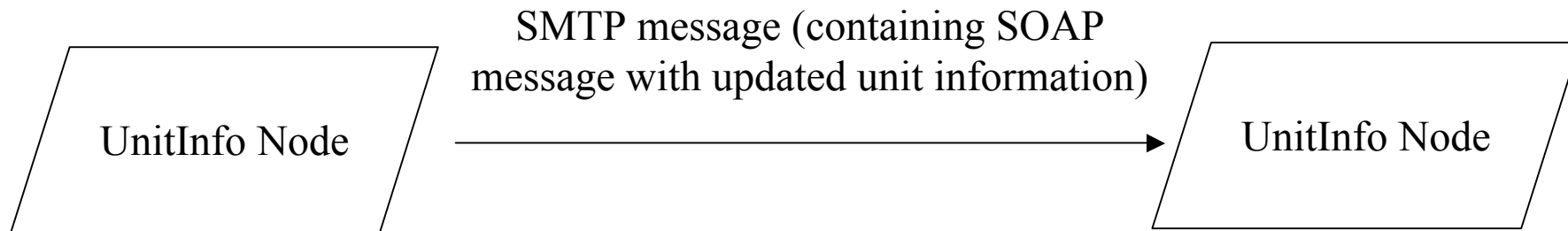


# Binding to other Protocols

---

- Although binding to HTTP is the most prevalent model, we can bind to other protocols as well.
  - Eg. there is currently a W3C Note describing SOAP binding over Email (such as the SMTP infrastructure).
- Sometimes it may be appropriate to bind to something else besides HTTP.
- Eg. we may not want any response to the message we send to the receiver.

# Example: SOAP over SMTP



- One node sending update information to another node.



# Web Service Component: WSDL

---

- Web Service Description Language (WSDL) is an XML-based language to describe the operations of a web service.
- Web Service servers can publish WSDL documents to enable clients to read and determine how to use the service.

# Example WSDL

Input/output  
description

Service location  
description

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="UnitInfoService"
  targetNamespace="http://www.pieinthesky.edu.au/UnitInfo.wsdl"
  xmlns:tns="http://www.pieinthesky.edu.au/UnitInfo.wsdl"
  ...>

  <message name="getUnitInfoRequest">
    <part name="unit_code" type="xsd:string"/>
  </message>
  <message name="getUnitInfoResponse">
    <part name="unit_name" type="xsd:string"/>
  </message>

  <portType name="UnitInfo_PortType">
    <operation name="getUnitInfo">
      <input message="tns:getUnitInfoRequest"/>
      <output message="tns:getUnitInfoResponse"/>
    </operation>
  </portType>

  <service name="UnitInfo_Service">
    <documentation>WSDL File for Unit Info Service</documentation>
    <port binding="tns:UnitInfo_Binding" name="UnitInfo_Port">
      <soap:address
        location="http://localhost:8080/soap/servlet/rpcrouter"/>
    </port>
  </service>

  ...
</definitions>
```

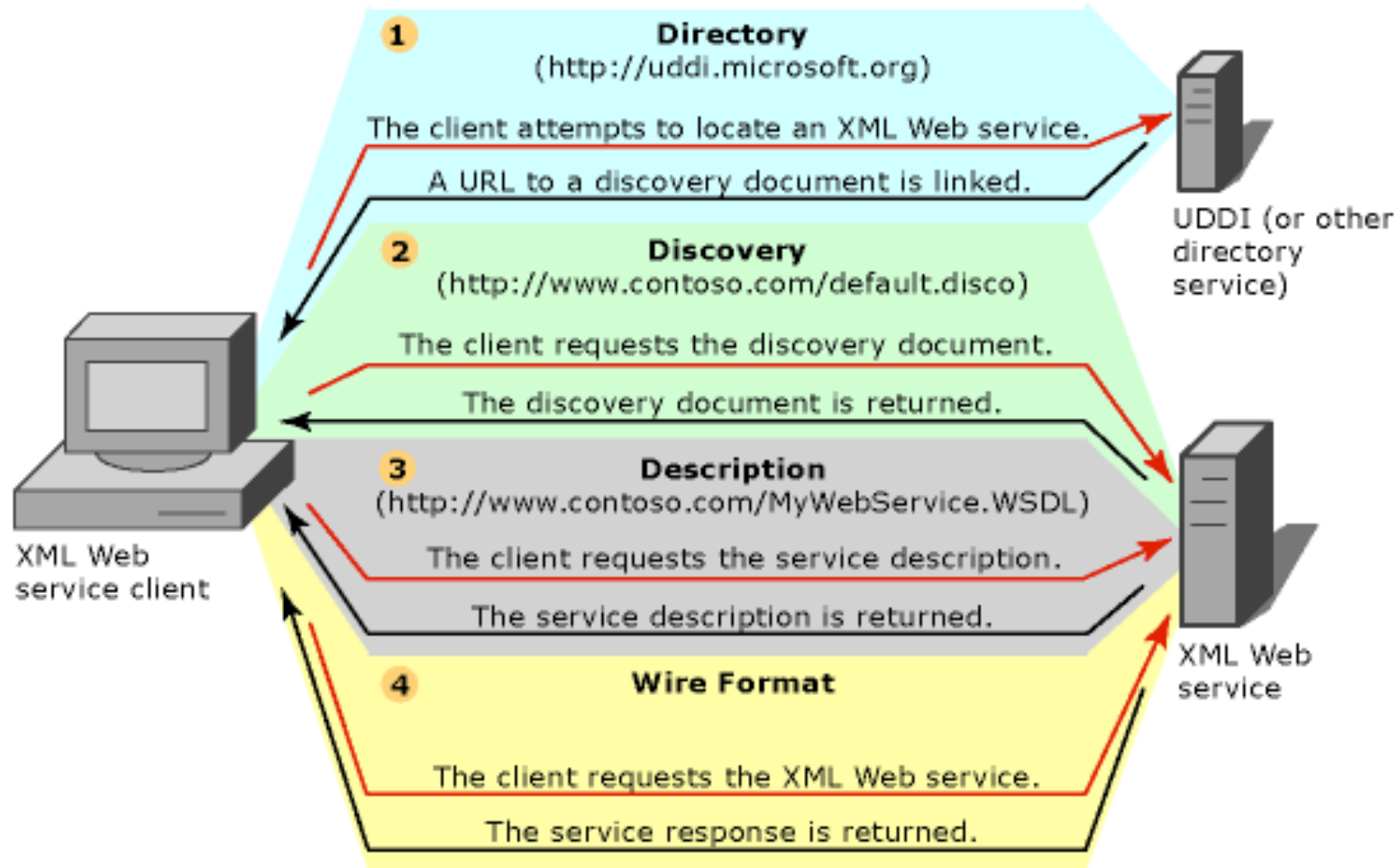


# Web Service Component: UDDI

---

- Universal Description, Discovery, and Integration (UDDI) is a platform-independent, open framework for describing, discovering businesses, and integrating business services.
- Basically,
  - Web Services companies publish themselves in UDDI **registries** (ie. announcing they have web services to be discovered).
  - Web Service clients queries the company to determine services.
  - Clients retrieves and reads the WSDL document for the appropriate service.
  - Clients then invokes the service according to how it is described in the WSDL document.

# Web Service Example



Source: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpguide/html/cpconwebservicesinfrastructure.asp>



# UDDI Registries

---

- There are currently UDDI registries maintained by companies like Microsoft and IBM.
- UDDI registries are loosely coupled after the DNS
  - Each registries should synchronize their data with other registries.
  - Like the DNS registries, UDDI registries do not contain the service information themselves. They are provide information on where and how to find and invoke the services.



# UDDI Registries

---

- **Unlike** the DNS, there are currently no contracts or restrictions that the different registries **MUST** synchronize with each other.
  - However, we expect such contracts to be in place as UDDI and Web Services develops.



# Information in UDDI Registries

---

- There are (potentially) 3 levels of information in UDDI registries:
  - White pages
    - Listings of companies that can be queried for services.
  - Yellow pages
    - Listing of companies searchable by kinds of service.
  - Green pages
    - Information on how to interact with companies Web Services.



# Example Web Service

---

- See description of Google Web API
  - <http://www.google.com.au/apis/index.html>
- This is a service without using UDDI, but will give you an idea about what is possible with Web Services.



# Reference

---

- Web Services at W3C
  - <http://www.w3.org/2002/ws/>



## Next week...

---

- Details of the SOAP message structure.
- Building a SOAP service using SOAP::Lite in Perl.