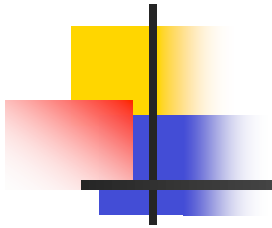
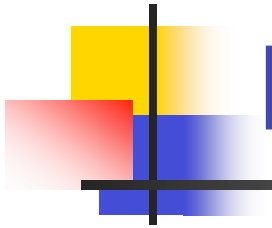
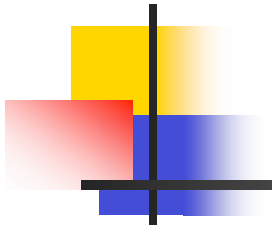


SOAP
and
Web Services
(Week 10 Lectures)



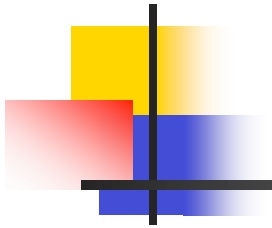
Learning Objectives

- Learn the concept of Web Services and why it is important
- Look at the preliminary details of Web Service components.



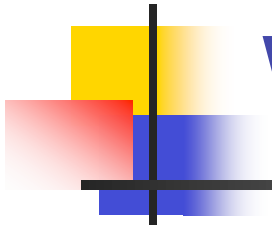
Learning Objectives

- In the scheme of what we are doing in this unit:
  - We are studying how to use XML as an important set of Internet technologies to use as solutions in different areas.
  - It is likely that any XML work in the industry will involve the use of Web Services. This lecture is to start you in the direction of learning about the concepts.



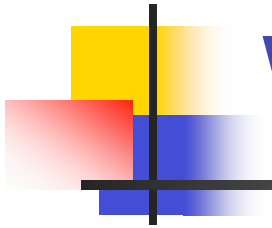
Lecture Outline

- Web Services and the Semantic Web
- Components of Web Services:
  - SOAP
  - WSDL
  - UDDI



Web Services

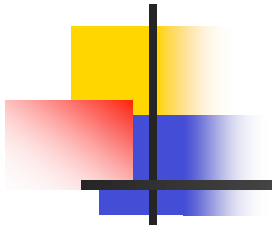
- The term ***Web Services*** refer to our current generation of technologies to allow application-to-application (program-to-program) communication by using XML.
- It covers how to:
  - define modular, self-**describing** applications
  - **publish** that the applications are available
  - **locate** these available applications from anywhere
  - **invoke** these applications remotely
- All these using open Internet standards like XML, HTTP and SMTP.



Web Services

- In other words***, Web Services*** are self-contained applications that can be described, published, located, and invoked over a network (especially the Internet).

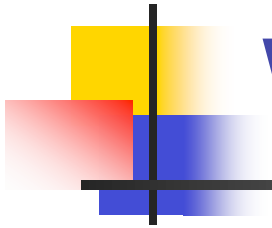
- Please note the difference between this technical use of the word “Web Services” and the non-technical use of “web services”, which refers to any program available through HTML over the web.



Distributed Object Technologies

- There have been past efforts in providing program-level object exchange:
  - OMG's CORBA/IIOP (www.corba.org/)
  - Microsoft's DCOM (www.microsoft.com/com/tech/dcom.asp)
  - Java RMI (java.sun.com/rmi)

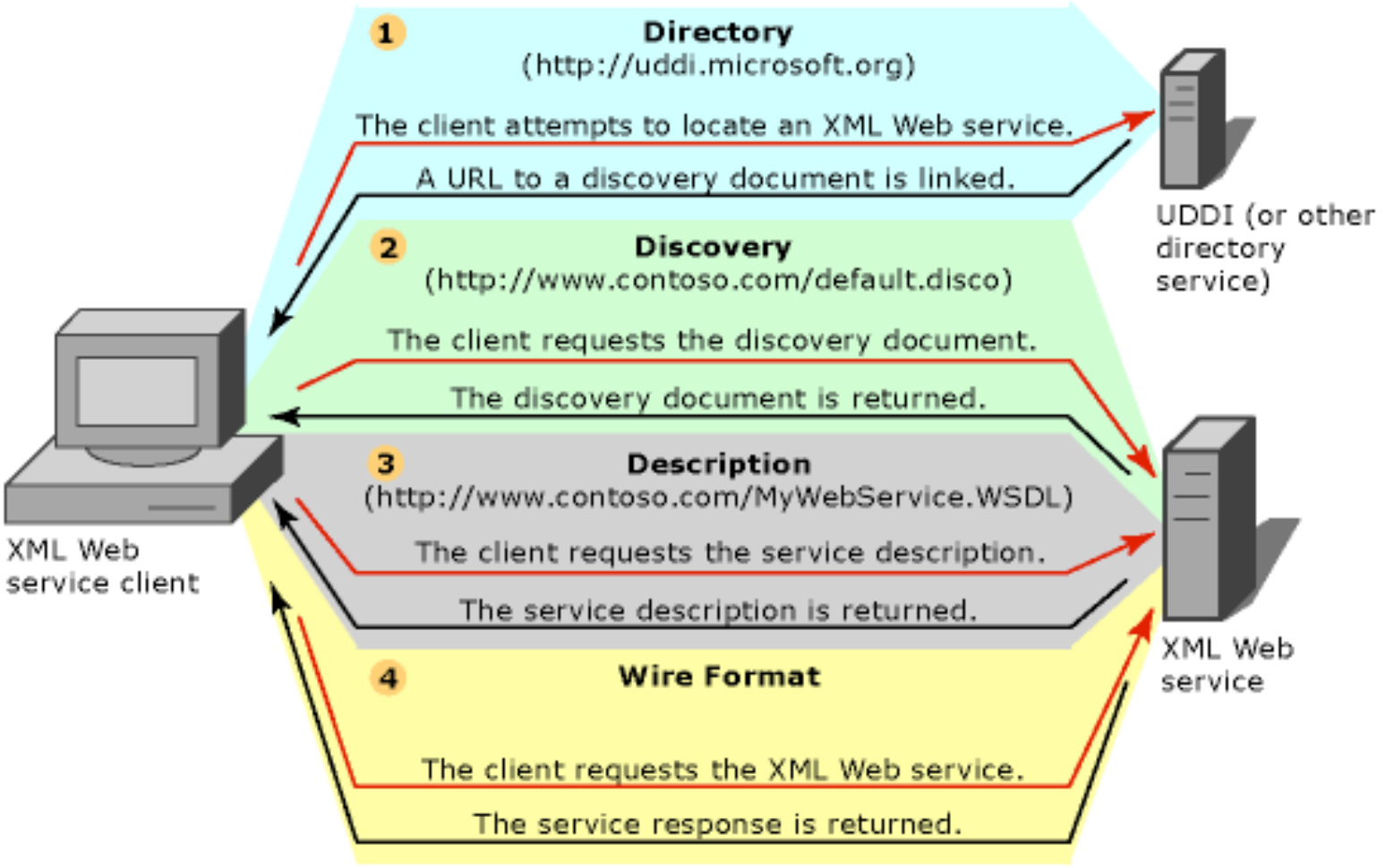
- But all of these technologies suffer from:
  - Complex set-up and object invocation.
  - Platform and language dependence.
  - lack of universal acceptance.
  - Lack of extensibility to different problem areas.



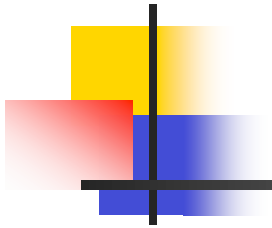
What is missing today?

- What is missing to today to allow a service (program, application) communication with another:
  - There is no consistent framework for describing how a service can be invoked. Eg. how do I use a particular PHP application?
  - There is no consistent framework for listing available services, and how clients can discover them. Eg. how do I find out what other PHP applications are available?
  - Location of services on the web and how to use them have to manually determined, and then manually invoked - they can't be done by programs automatically.

Web Service Example

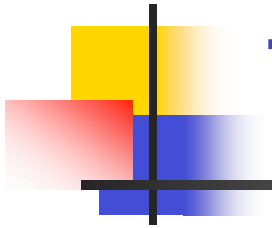


Source: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpguide/html/cpconwebservicesinfrastructure.asp



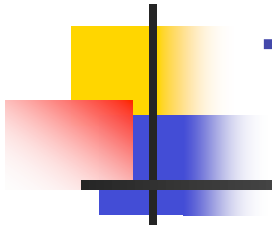
Components of Web Services

- 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).



The Semantic Web

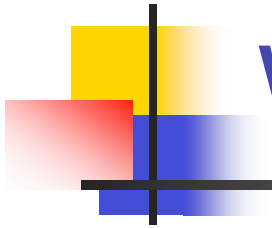
- The ***Semantic Web*** is W3C's vision of the web where:
  - "... it becomes a place where data can be shared and processed by automated tools as well as by people."
  - "... tomorrow's programs must be able to share and process data even when these programs have been designed totally independently."
  - "The ***Semantic Web*** is a vision: the idea of having data on the web defined and linked in a way that it can be used by machines not just for display purposes, but for automation, integration and reuse of data across various applications."
    - All Quotes from http://www.w3.org/2001/sw/



The Semantic Web

- The idea of the Semantic Web is to build an environment where data and services are precisely defined so that programs can independently process without the need for human manual intervention at every step.

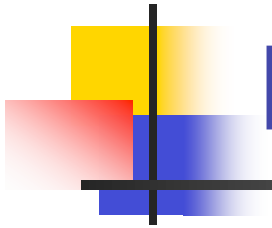
- Web Services is an integral part of that.



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.



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>
```

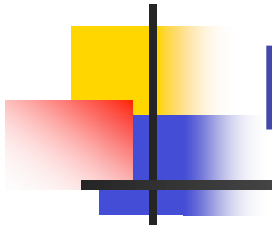
Example SOAP Response

```
<?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>
```

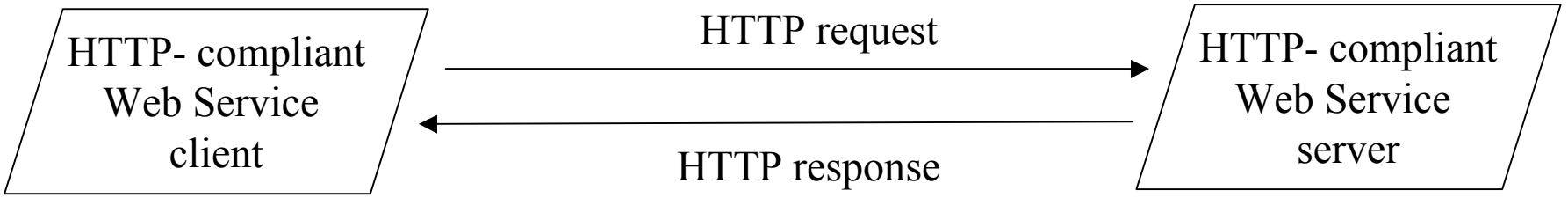
Binding to Different Transport Protocols



- 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.

Example: SOAP over HTTP



Example SOAP Request in HTTP Message

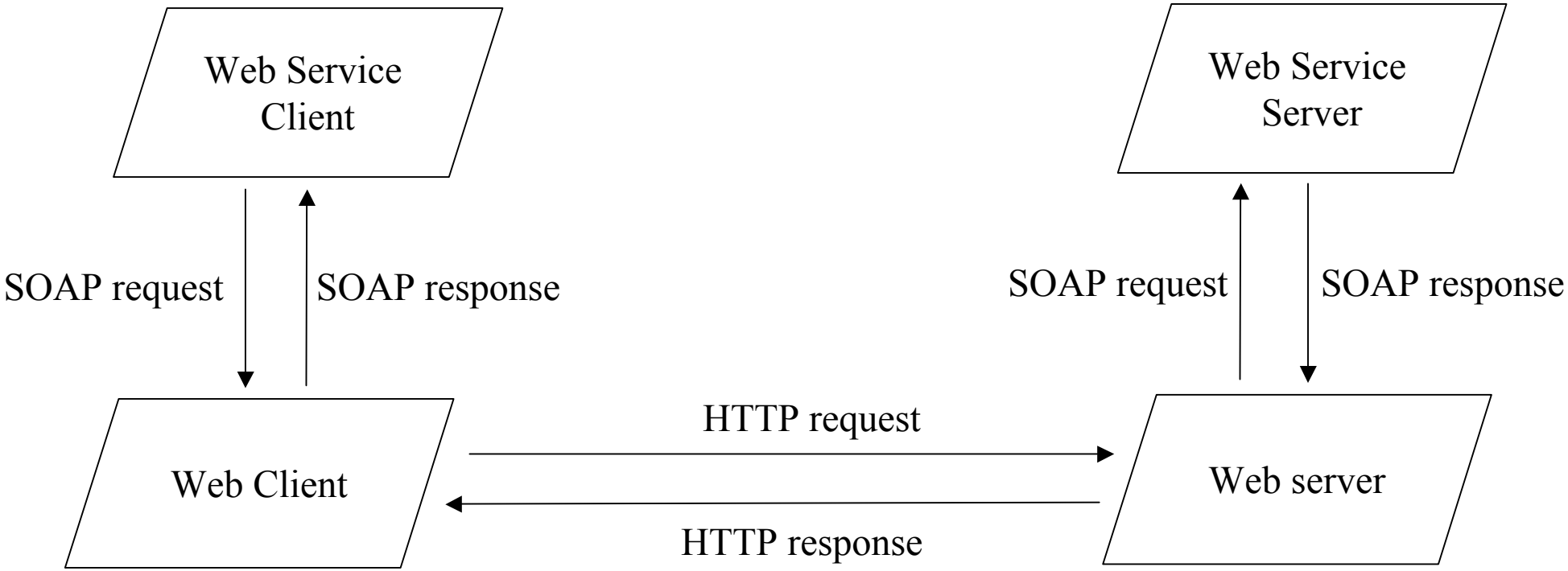
```
POST  /ws/UnitInfoService  HTTP/1.1
Host: 134.115.64.2
Content-type: text/xml
Content-length: 300
SOAPMethodName: urn:examples:unitinfoservice#getUnitInfo

<?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>
```

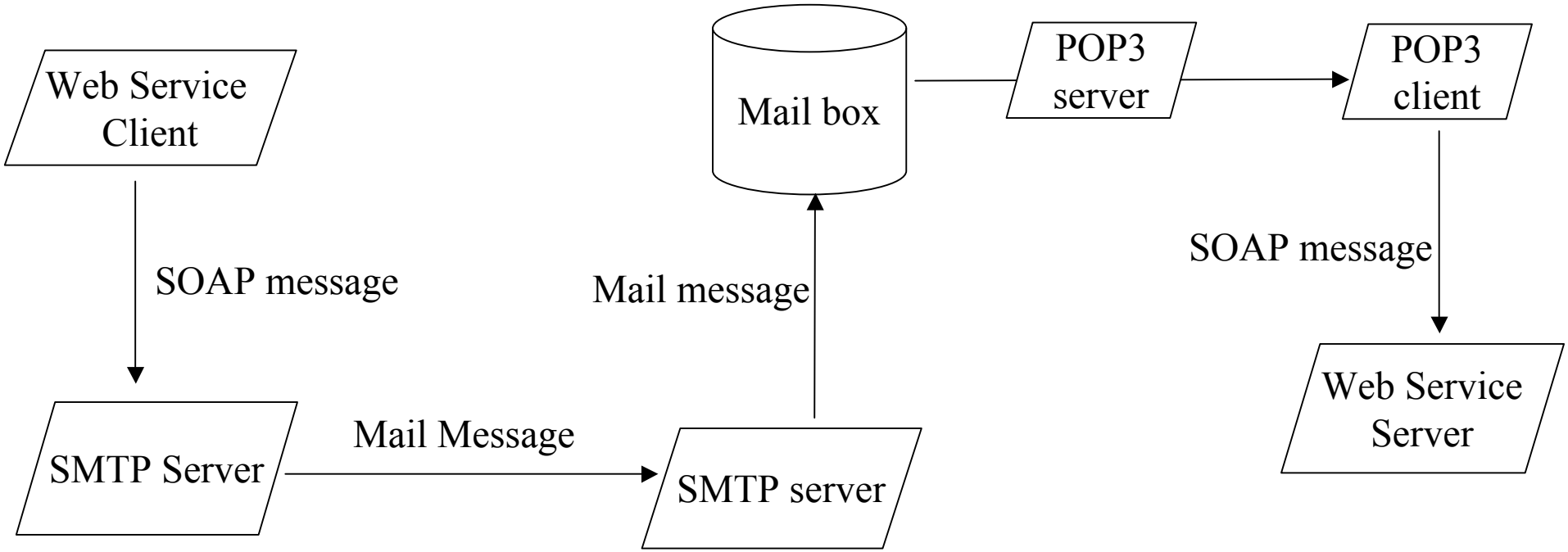
HTTP Message Headers

HTTP Message Body (our SOAP request from before)

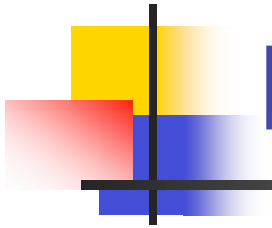
Another Example: SOAP over HTTP



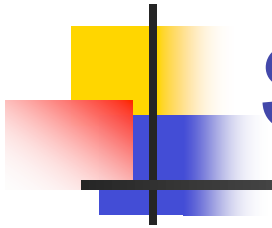
Example: SOAP over SMTP



Binding to Different Transport Protocols

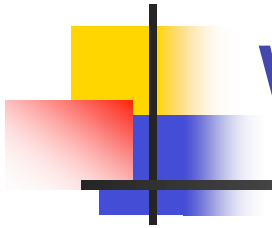


- The ability to bind to different transport protocols gives the parties involved the option of using the most appropriate set-up based on their infrastructure and requirements.



SOAP Messaging

- SOAP doesn't necessarily have to follow the request-response operations like HTTP.
- SOAP can support 4 types of operations:
  - Request-initiated response (the same as normal HTTP).
  - Server solicit request from client.
  - One way message from client to server (with no response)
  - Notification message from server to client.
- This gives the applications flexibility in communicating what they need to.



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

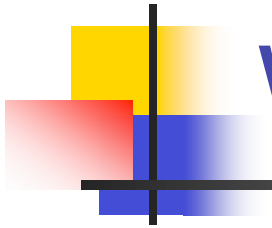
Service location description

```
<?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>
```

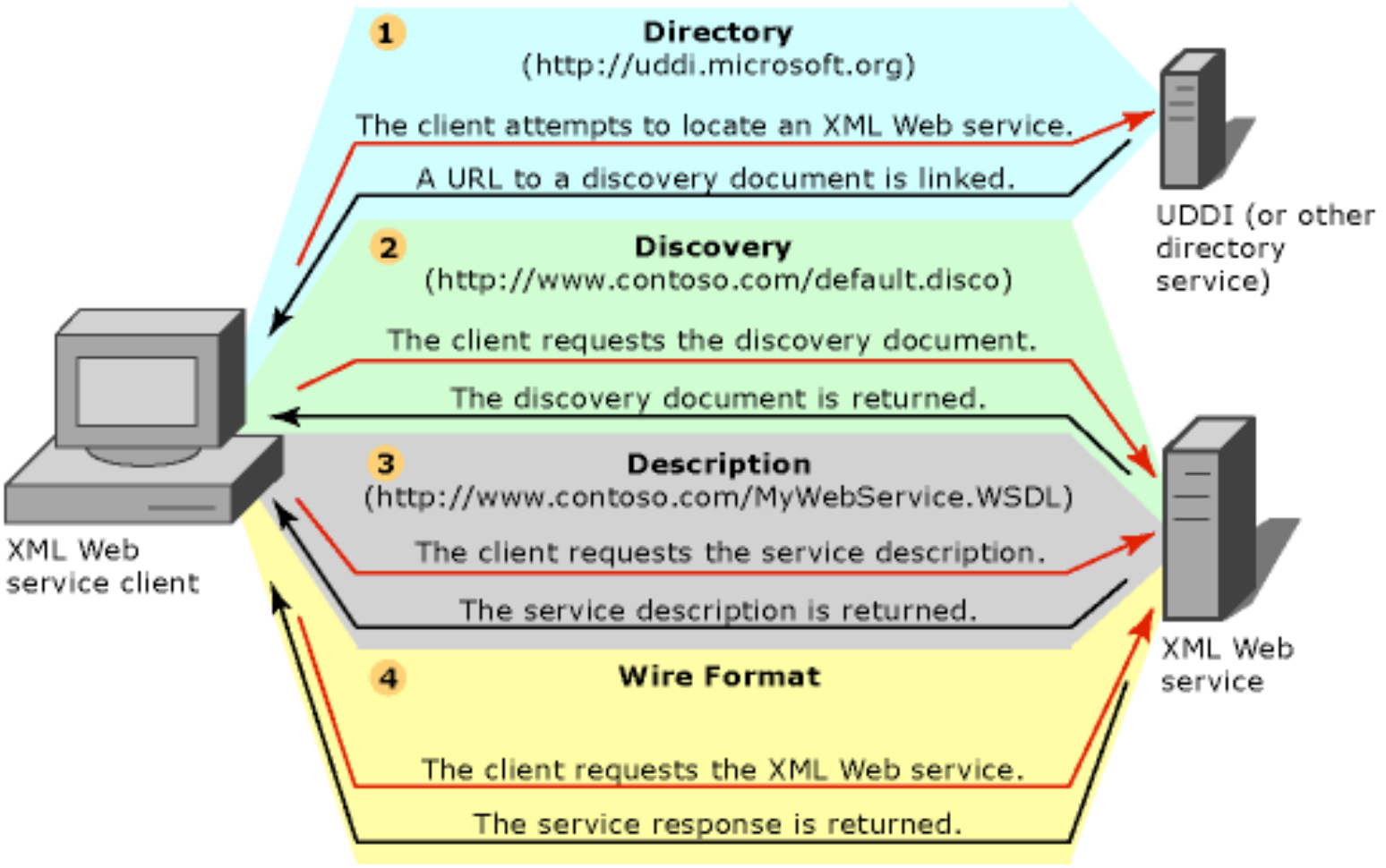


Web Service Component: UDDI

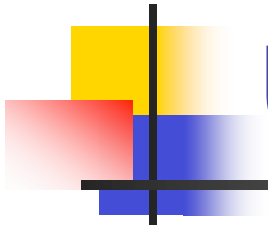
- Universal Description, Discovery, and Integration (UDDI) is a platform-independent, open framework for describing, discovering businesses, and integrating business services.

- 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 (again)

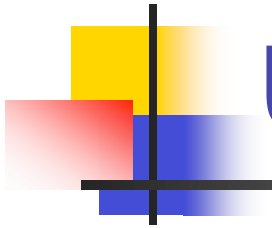


Source: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpguide/html/cpconwebservicesinfrastructure.asp



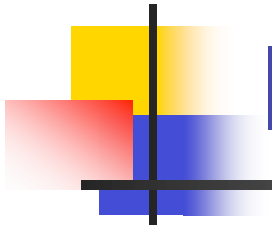
UDDI Registries

- 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.



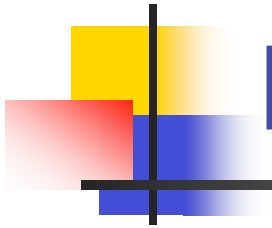
UDDI Registries

- **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

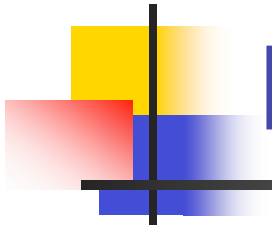
- There are (potentially) 3 levels of information in UDDI registries:
  - White pages
    - Listings of companies that can be queried for services.
  - Yellow pages
    - Listing of companies searchable by kinds of service.
  - Green pages
    - Information on how to interact with companies Web Services.



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/