Web Services Using SOAP, WSDL, and UDDI

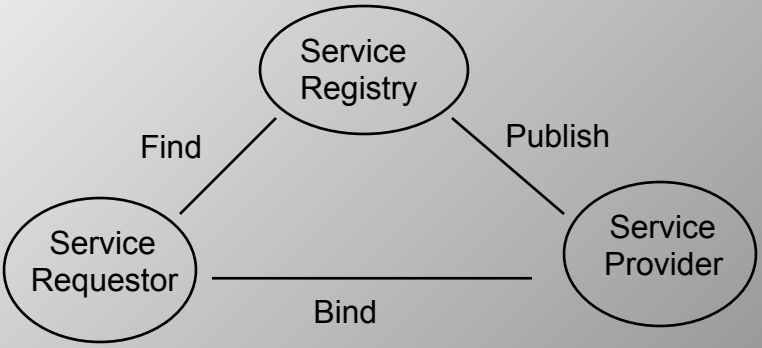
What are “Web Services”?

- IBM
  - “A Web service is an interface that describes a collection of operations that are network accessible through standardized XML messaging”
- Microsoft: XML Web Services
  - “.. expose useful functionality to Web users through a standard Web protocol”
  - “.. provide a way to describe their interfaces in enough detail to allow a user to build a client application to talk to them”
  - “.. are registered so that potential users can find them easily”

Why Web Services?

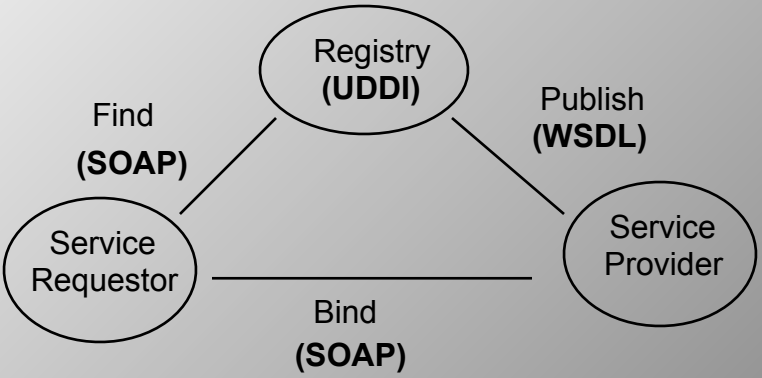
- From business standpoint
  - Integration
    - Within an organization
    - Between companies
    - Allows time/cost efficiencies
      - Purchase orders
      - Answering inquiries
      - Processing shipment requests
    - Do this without locking in to a single partner

Web Service Architecture



- Service-Oriented Architecture

Architecture II



- All the technologies are XML based …

XML Leveraging Features

- XML Namespaces
  - Collision
    - Common XML element names
      - Application specific or embedded in message?
  - Allows composition of multiple XML documents
    - Identifies elements belonging to the same document type

XML Leveraging Features II

- XML Schemas
  - Alternative to DTDs for describing document structure
  - Written in XML
    - Simple types
    - Complex types
  - Reusable
    - Intended to be used with namespaces

SOAP

- **S**imple **O**bject **A**ccess **P**rotocol
- Web service messaging and invocation
- 2nd Generation XML Protocol
  - Takes advantage of
    - XML Namespaces
    - XML Schema

First Generation XML Protocol

- Based on XML 1.0
- Example: XML-RPC
  - Introduced by Userland in 1998
  - Uses HTTP as underlying transport

Call

```
<methodCall>
  <methodName>NumberToText</
methodName>
   <params>
       <param>
          <value><i4>28</i4></value>
       </param>
   </params>
</methodCall>
```

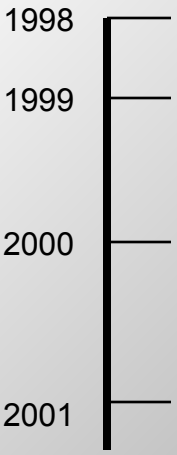
Response

```
<methodResponse>
  <params>
    <param>
      <value>
        <string>twenty-eight</string>
      </value>
    </param>
  </params>
</methodResponse>
```

First Gen. XML Protocol Issues

- Extensibility
  - All protocol architects had to agree for changes
  - Avoid with namespaces
- Datatyping
  - Single DTDs
    - Limited in use of XML elements
    - Descriptive nature of XML sacrificed
  - XML schema is a better solution

SOAP History



1998
- Term *SOAP* coined at Microsoft

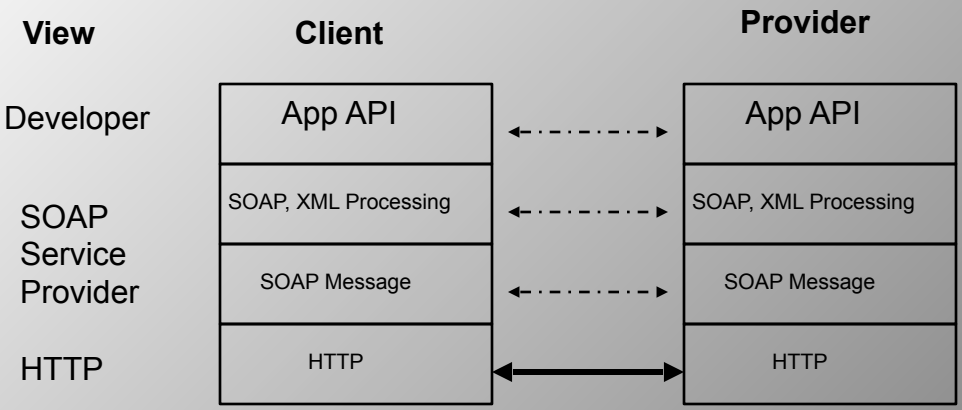
1999
- Microsoft works with BizTalk to release SOAP 0.9
- Submitted to IETF
- SOAP 1.0 released in December

2000
- SOAP 1.1 submitted to W3C with IBM
- IBM releases a Java SOAP implementation
- Sun starts work on Web services in J2EE

2001
- SOAP 1.2 released by XML Protocol working group at W3C

Currently, about 80+ SOAP implementations available including Apple…

SOAP Messaging Layers



SOAP Message

```
<?xml version="1.0" encoding="UTF-8"?>
<Envelope>
  <Header>
  </Header>
  <Body>
  </Body>
</Envelope>
```

SOAP Envelope

- Root element
- Mandatory
- Does not expose any protocol versions
  - Protocol version is the URI of SOAP envelope namespace
  - encodingStyle attribute for complex types

```
<SOAP-ENV:Envelope
   SOAP-ENV:encodingStyle=http://schemas.xmlsoap.org/soap/encoding/
   xmlns="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
   xmlns:xsd="http://www.w3.org/1999/XMLSchema">
```

SOAP Header

- Optional
- Allows packaging of information orthogonal to message
  - Transactions
    - Authentication information
    - Account information
- SOAP-ENV:mustUnderstand

SOAP-ENV:mustUnderstand

- Attribute for Header element
- Value 0 – skip this element if it doesn't make sense
- Value 1 – must fail if it doesn't make sense
  - Ensures recipients be aware of important protocol extensions

```
<SOAP-ENV:Header>
   <t:client xmlns:t="Some-URI"
      SOAP-ENV:mustUnderstand="0">
    sacharya@inktomi.com
   </t:client>
</SOAP-ENV:Header>
```

SOAP Body

- Can contain arbitrary XML
- Conventions for
  - RPCs
  - Faults
    - Faultcode – lookup string
    - Faultstring – human readable string
    - Faultactor – where in the message path
    - Detail – optional
  - Data encoding

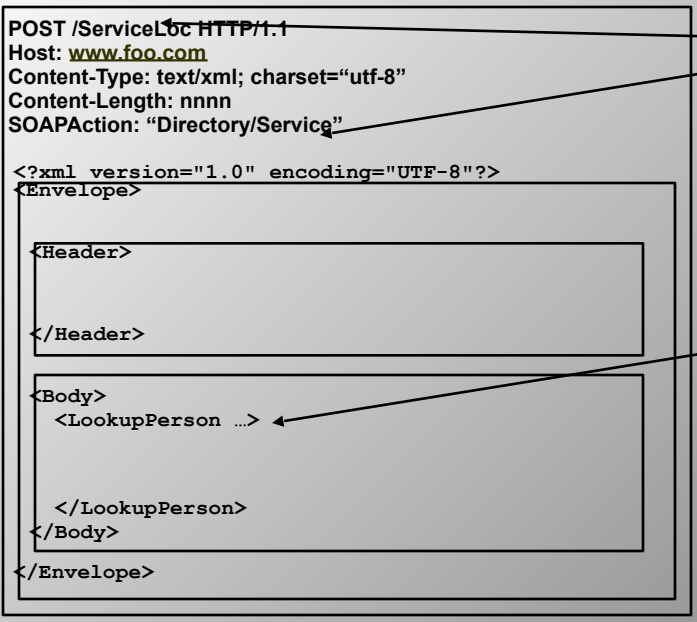
Data encoding in SOAP

- SOAP provides default encoding schema
  - Why reinvent the wheel?
- Simple data types
  - Use “xsi:type”
  - String, floats etc
- Complex data types
  - SOAP arrays
  - Structs: compound types
- Data referencing
  - Href and id attributes

Data encoding in SOAP

- Binary data
  - Base64 encoding
- Can roll your own schema
  - encodingStyle
  - Interoperability issues

SOAP Protocol Binding: HTTP



Sample RPC Call

Other SOAP Protocol Bindings

- HTTPS
  - Similar to HTTP
    - Use POST
    - Return 200 for success
    - 500 for failure + SOAP fault
    - SOAPAction HTTP header for hint
    - MIME media type: text/html
- SMTP
- SOAP messages with Attachments

SOAP RPC Example: getQuote

```
import SOAP
server = SOAP.SOAPProxy("http://services.xmethods.com:80/
    soap",
    namespace = 'urn:xmethods-delayed-quotes')

print "IBM>>", server.getQuote(symbol = 'IBM')
```

RPC Invocation Message

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
     SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
     xmlns:xsd="http://www.w3.org/1999/XMLSchema"
     xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
     xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
     xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/">

 <SOAP-ENV:Body>
    <ns1:getQuote
     xmlns:ns1="urn:xmethods-delayed-quotes" SOAP-ENC:root="1">

        <symbol xsi:type="xsd:string">IBM</symbol>

    </ns1:getQuote>
 </SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```

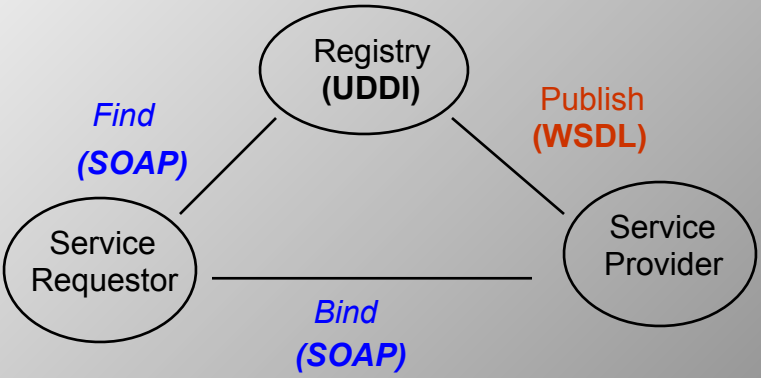
RPC Reply Message

```
<?xml version='1.0' encoding='UTF-8'?>
   <soap:Envelope
      xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'
      xmlns:xsi='http://www.w3.org/1999/XMLSchema-instance'
      xmlns:xsd='http://www.w3.org/1999/XMLSchema'
      xmlns:soapenc='http://schemas.xmlsoap.org/soap/encoding/'
   soap:encodingStyle='http://schemas.xmlsoap.org/soap/encoding/'>

 <soap:Body>
   <n:getQuoteResponse xmlns:n='urn:xmethods-delayed-quotes'>
     <Result xsi:type='xsd:float'>107.89</Result>
   </n:getQuoteResponse>
 </soap:Body>

</soap:Envelope>
```

Returns:

IBM>>107.89

Roadmap


Registry
(UDDI)
Find
(SOAP)
Publish
(WSDL)
Service
Requestor
Service
Provider
Bind
(SOAP)

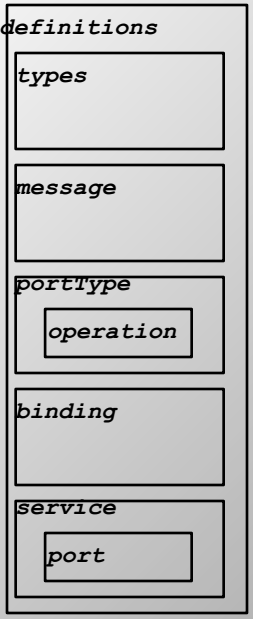
WSDL

- **W**eb **S**ervice **D**efinition **L**anguage
- Predecessors include
  - COM, CORBA IDLs
  - Network Accessible Service Specification Language (IBM)
  - SOAP Contract Language (Microsoft)
  - First submitted to W3C in Sep 2000
  - Current version is 1.1

WSDL

- Define a web service in WSDL by
  - Writing an XML document conforming to the WSDL specs
- Describes three fundamental properties
  - What a service does
    - Operations (methods) provided by the service
  - How a service is accessed
    - Data format and protocol details
  - Where a service is located
    - Address (URL) details

WSDL Components



All the data types used by the Web service

Parameters and messages used by method

Abstract interface definition – each *operation* element defines a method signature

Binds abstract methods to specific protocols

A service is a collection of ports.
A port is a specific method and its URI

Sample WSDL: getQuote

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

<definitions name="net.xmethods.services.stockquote.StockQuote"
targetNamespace="http://www.themindelectric.com/wsdl/
net.xmethods.services.stockquote.StockQuote/" xmlns:tns="http://
www.themindelectric.com/wsdl/net.xmethods.services.stockquote.StockQuote/"
   xmlns:electric="http://www.themindelectric.com/" xmlns:soap="http://
schemas.xmlsoap.org/wsdl/soap/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:wsdl="http://
schemas.xmlsoap.org/wsdl/" xmlns="http://schemas.xmlsoap.org/wsdl/">

<message name="getQuoteResponse1">
    <part name="Result" type="xsd:float" />
</message>

<message name="getQuoteRequest1">
    <part name="symbol" type="xsd:string" />
</message>
```

Sample WSDL: getQuote

```
<portType name="net.xmethods.services.stockquote.StockQuotePortType">
   <operation name="getQuote" parameterOrder="symbol">
     <input message="tns:getQuoteRequest1" />
     <output message="tns:getQuoteResponse1" />
   </operation>
</portType>

<binding name="net.xmethods.services.stockquote.StockQuoteBinding"
         type="tns:net.xmethods.services.stockquote.StockQuotePortType">
    <soap:binding style="rpc"
                  transport="http://schemas.xmlsoap.org/soap/http" />
    <operation name="getQuote">
       <soap:operation soapAction="urn:xmethods-delayed-quotes#getQuote" />
         <input>
           <soap:body use="encoded" namespace="urn:xmethods-delayed-quotes"
             encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
         </input>
         <output>
           <soap:body use="encoded" namespace="urn:xmethods-delayed-quotes"
             encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
         </output>
    </operation>
</binding>
```

Sample WSDL: getQuote

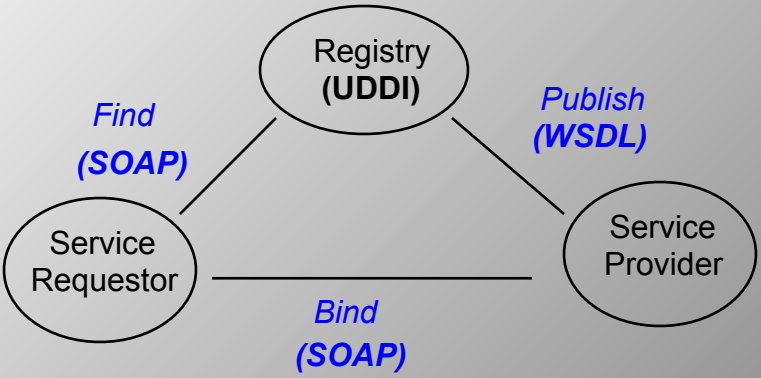
```
<service name="net.xmethods.services.stockquote.StockQuoteService">
  <documentation>net.xmethods.services.stockquote.StockQuote web service
  </documentation>
   <port name="net.xmethods.services.stockquote.StockQuotePort"
         binding="tns:net.xmethods.services.stockquote.StockQuoteBinding">
     <soap:address location="http://64.39.29.211:9090/soap" />
   </port>
</service>

</definitions>
```

WSDL to Code

- Translators available that can
  - Convert WSDL document to code
    - IBM's WSTK Toolkit
    - Apache AXIS WSDL2java program
    - Soapy.py in Python
    - Not perfect
  - Derive WSDL from Java classes
    - Apache WSDL program
    - Much work remains to be done


www.xmethods.net - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit
Links
Address http://www.xmethods.net/
XMETHODS
Services Implementations Manage Register Tutorials Mailing List About
Welcome to XMethods.
Emerging standards such as SOAP will enable a new generation of "web services" that allow systems to make remote procedure calls to other systems over the Internet. For example, a corporate inventory management system might publish a service that allows a customer system to check real-time inventory levels. This site lists publicly accessible web services.
Updates
2002-01-30 Configuring SOAP calls with the WSDL Analyzer [Read]
2002-01-15 SITE UPGRADED! [Read]
2002-01-14 All WSDL on the site has been validated. [Read]
2001-01-14 Sign up to be notified of new services. [Read]
SOAP Services
Owner
Service Name
Description
Implementation
erynaps
eDynagaFeed
Daily Articles. Coding Tips and .NET Code samples
MS .NET
mybubble.com
Company Profile
Provides company profile for a given stock ticker
Apache SOAP
mybubble.com
Current News for a Stock
Provides current news of a company for a given stock ticker.
Apache SOAP
mybubble.com
Stock Quotes
Provides current quotes and additional info. for a given stock ticker.
Apache SOAP
moyerconnect
Loan Term Worksheet
Loan Term Worksheet
GLUE
moyerconnect
Extra Payment Worksheet
Extra Payment Worksheet
GLUE
moyerconnect
Financial Calculator
Monthly Payment Worksheet
GLUE
geographynetwork.com
PlaceFinder
Returns the x,y location for a place name in any part of the world.
GLUE
daniel
Chess
Play Chess with a WebService
Delphi
erynaps
NFL Headline News
Get the NFL Headline News
MS .NET
erynaps
Who Is
The Web Service form of "Whois" Domain Registry service
MS .NET
erynaps
Daily Dilbert
Returns a binary stream of Today's Dilbert comic strip
MS .NET
elingra.com
Monthly Mortgage Payment
Calculates your monthly mortgage payment
EXADEL
jrosso
Location Information
Info about a location from zip code, area code, city, or state
MS .NET
caryjensen
Temperature Conversion Service
Converts Fahrenheit to Centigrade and vice versa
Delphi
alcyonl.com
SendEmail
Send a simple e-mail
Alcyonl-webDTP
simonfell
whois
A SOAP version of the standard whois service
dotlr
GIServer
GIServer Location Services
Country Location and Projections Transformation of Geographic Points using mapGIS objects
Delphi

Roadmap


Registry
(UDDI)
Find
(SOAP)
Publish
(WSDL)
Service
Requestor
Service
Provider
Bind
(SOAP)

UDDI

- Universal Description, Discovery, and Integration
- API for a Web based registry
- Implemented by an *Operator* Site
  - Replicate each others' information
- Formally announced in Sept, 2000
  - Collaboration between IBM, Microsoft, Ariba
  - Community of 310 companies

A UDDI Registry

- Who?
  - Basic business information
    - Name, contact information
- What?
  - Get classification
    - Standard Industry Codes, NA Industry Code Std
- Where?
  - Service URI
- How?
  - Describes a how a given interface functions

UDDI Data Structures

- businessEntity:
  - Basic business information
  - Used by UDDI for “yellow” pages
- businessService:
  - Services provided by that business
  - Grouping of related businesses
- bindingTemplate:
  - What the service looks like (tModel element)
  - Where to access the service

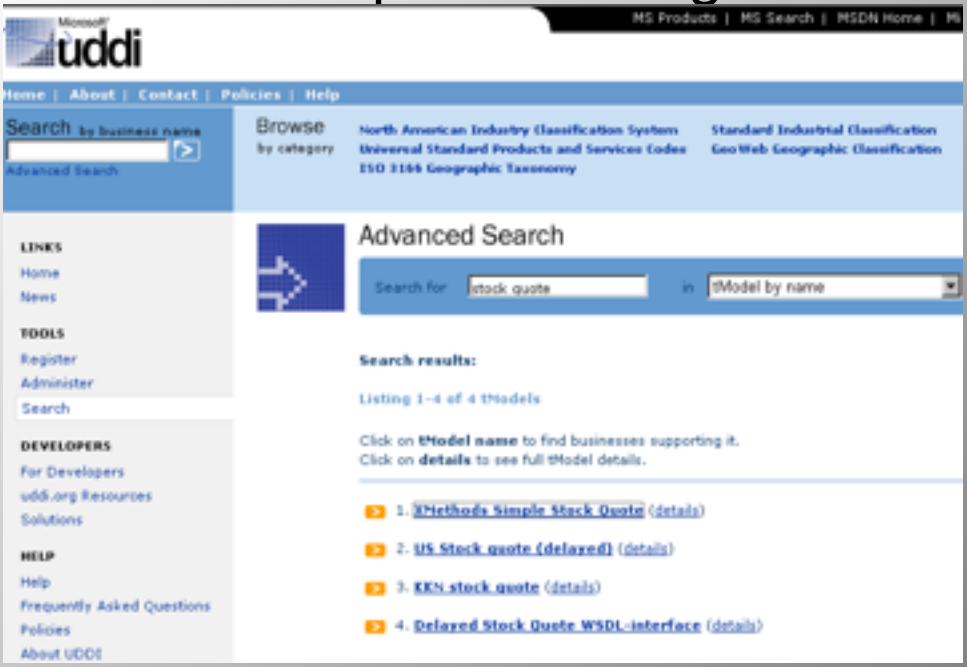
UDDI Data Structures

- tModel
  - Technology model
  - Could contain just about anything
  - Has service details
    - Abstract industry specs
    - Service specs
- Designed to be reusable
- Can contain pointer to WSDL document

Query Pattern

- Browse:
  - UDDI yellow page data has hierarchy
  - Search via Web/standalone client app
- Drill down:
  - Given a specific candidate, get all details
- Invocation

Sample Browsing



tModel Detail

Associated Businesses

Single Business

Inquiry API

- Generally accessible
  - Find_binding
  - Find_business, find_relatedBusiness
  - Find_service
  - Find_tmodel
  - Get_bindingDetail
  - Get_businessDetail
  - Get_serviceDetail
  - Get_tModelDetail
- Use SOAP to access

Publishing API

- Restricted access
  - Save_service, save_business, save_binding, save_tModel
  - Delete_service, delete_business, delete_binding, delete_tModel
  - Get_binding, get_registeredInfo, get_authToken
  - Add_publisherAssertions, get …, delete …

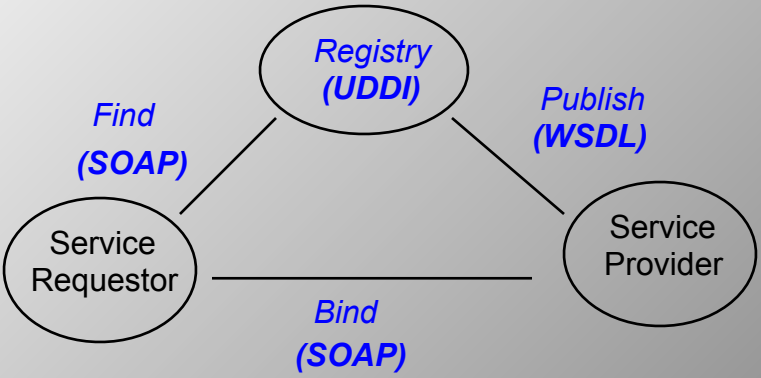
Other UDDI Issues

- Security
  - No global standard
  - Each operator site must select/implement an authentication protocol that still allows publishing
- Versioning
  - Numbers not used
  - *Generic* element used in function calls

Open UDDI Issues

- Effective search
  - Classification and Categorization
- Private UDDI registries
  - E-marketplace
  - Portal
  - Partner catalog
  - Internal Application Integration

Roadmap


Registry
(UDDI)
Find
(SOAP)
Publish
(WSDL)
Service
Requestor
Service
Provider
Bind
(SOAP)

Overall Issues

- Interoperability
- Web Services Everywhere
  - Peer to peer vs centralized