

# Universal Description, Discovery and Integration (UDDI)

---

Asst. Prof. Dr. Kanda Runapongsa  
(krunapon@kku.ac.th)  
Department of Computer Engineering  
Khon Kaen University

1

---

---

---


---

---

---

---

---



## Agenda

- What is UDDI and Why UDDI?
- UDDI Data Types and their structural relationship
- UDDI Programming Interface
- UDDI over SOAP
- Authoring Steps of WSDL, UDDI
- Discovery, Binding and Invocation of a Service
- Java API for UDDI

2

---

---

---


---

---

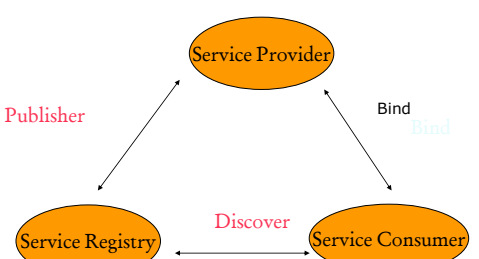
---

---

---



## Service Architecture



The diagram illustrates the UDDI service architecture. It features three main components: Service Registry, Service Provider, and Service Consumer, all represented by orange ovals. Service Registry is positioned at the bottom left, Service Provider at the top center, and Service Consumer at the bottom right. A red arrow labeled 'Publish' points from Service Registry to Service Provider. A red arrow labeled 'Discover' points from Service Consumer to Service Registry. A blue arrow labeled 'Bind' points from Service Consumer to Service Provider. A light blue arrow labeled 'bind' points from Service Provider to Service Consumer.

UDDI defines a scheme to **publish** and **discover** information about Web services

---

---

---

---

---

---

---

---

### What is UDDI?

- Programmatic registration and discovery of business entities and their Web services
- Based on SOAP, HTTP, XML
- Registry data
  - Business registrations
  - Service type definitions

4

---

---

---

---

---


---

---

---

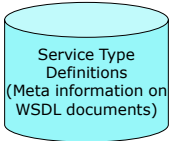
### Registry Data

Created by businesses



Business Registrations

Created by standard Organizations, industry consortium,



Service Type Definitions  
(Meta information on WSDL documents)

5

---

---

---

---

---

---

---

---

### Business Registration Data

- “White pages”
  - Basic contact information and identifiers about a company
    - Business name, address, contact information, and unique identifiers, such as tax IDs
- “Yellow pages”
  - Information that describes a web service using different categorizations
- “Green pages”
  - Technical information that points to the grouping information of web services and where the web services are located

6

---

---

---


---

---

---

---

---

 **Why UDDI?**

- Businesses need to reach their customers and partners with information about their products and web services
- Need a uniform method that detailed how to integrate the systems and processes that are already in place at and between business partners
- Platform independent service publication and discovery
- Enables dynamic service discovery 7

---

---

---


---

---

---

---

---

 **How UDDI is Used (1/3)**

- UDDI has several different uses, based on the perspective of who is using it
- From a business analyst's perspective, UDDI is similar to an Internet search engine for business processes
- A business analyst can browse one or more UDDI registries to view the different businesses that expose web services and specifications of those services 8

---

---

---


---

---

---

---

---

 **How UDDI is Used (2/3)**

- However, business users probably won't browse a UDDI registry directly
- A series of marketplaces and business search portals could crop up to provide business analysts with a more user-oriented approach
  - To browse the services and businesses hosted in a UDDI registry 9

---

---

---


---

---

---

---

---

 **How UDDI is Used (3/3)**

- Software developers use the UDDI programmer's API
  - To publish services (i.e., put information about them in the registry)
  - To query the registry to discover services matching various criteria
- It is convenient that software will eventually discover a service dynamically and use it without requiring human interaction

10

---

---

---


---

---

---

---

---

 **What Uses UDDI?**

- Tool building client (Service Consumer)
  - Browse or search registry
  - Create a service proxy
- Tool publishing the service
  - Generate WSDL
  - Construct UDDI entities
- Application that needs dynamic binding
  - Directly access UDDI
  - Query can be pre-generated

11

---

---

---


---

---

---

---

---

 **A UDDI registry**

- A UDDI registry is itself a Web service
- A Web service consumer queries the UDDI registry using the SOAP API defined by the UDDI specification
- The UDDI specification publishes a WSDL description of the UDDI registry service

12

---

---

---


---

---

---

---

---

 Public and Private UDDI Registries

- A UDDI registry can be operated in two modes: public mode and private mode
- A public UDDI registry is available for everyone to publish/query the business and service information on the Internet

13

---

---

---


---

---

---

---

---

 A Public UDDI Registry

- A public UDDI registry can be a logical single system built upon multiple UDDI registry nodes
  - Their data synchronized through replication
- Such global grouping of UDDI registry nodes is known as a **UDDI Business Registry (UBR)**

14

---

---

---


---

---

---

---

---

 UDDI Business Registry (UBR) (1/2)

- UBR is a conceptually single system built from multiple nodes that their data synchronized through replication
- A series of operator nodes each hosts a copy of the content
- The global grouping of operator nodes is jointly known as the UBR

15

---

---

---

---

---

---

---

---

### UDDI Business Registry (UBR) (2/2)

- Content inserted into the UBR is done at a single node, and that operator node becomes the master owner of that content
- Any subsequent updates or deletes of the data must occur at the operator node where the data was inserted

16

---

---

---

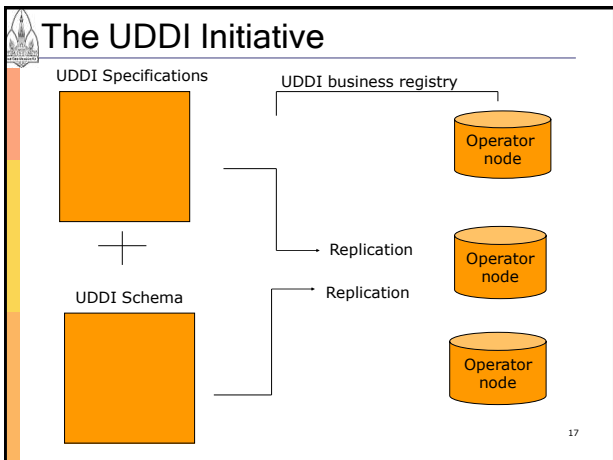
---

---

---

---

---




---

---

---

---

---

---

---

---

### A Private UDDI Registry

- A private UDDI registry is operated by a single organization or a group of collaborating organizations
  - Share the information that would be available only to the participating bodies
- Private UDDI registries can impose additional security controls to prevent access by unauthorized users

18

---

---

---


---

---

---

---

---

 A Private Operator Node

- A company can provide operator node that is not part of the UBR
- Private nodes do not have data synchronized with the UBR, so the information within is distinct
- A grouping of companies can also create a “private cloud” of nodes that have information replicated between their private nodes

19

---

---

---


---

---

---

---

---

 Interacting with a UDDI Registry

- Typically, vendors implementing a UDDI registry provide two ways of interacting with a UDDI Registry Service
  - A GUI, for interacting with a UDDI registry
  - A programming interface for communicating with the UDDI registry

20

---

---

---


---

---

---

---

---

 A List of Public UDDI Registries

- Microsoft:  
<http://uddi.microsoft.com>
- IBM: <http://www-3.ibm.com/services/uddi>
- SAP: <http://udditest.sap.com/>
- Systinet:  
<http://www.systinet.com/uddi/web>

21

---

---

---

---

---

---

---

---

### UDDI Adoption Phases

- Phase 1: Experimental stage
- Phase 2: Private UDDI registry within an intranet (where we are today)
- Phase 3: Public UDDI registries with no coordination among them
- Phase 4: Public UDDI registries with coordination (i.e. replication)
- Phase 5: Value added registry services

22

---

---

---

---

---

---

---

---

### Agenda

- What is UDDI and Why UDDI?
- **UDDI Data Types and their structural relationship**
- UDDI Programming Interface
- UDDI over SOAP
- Authoring Steps of WSDL, UDDI
- Discovery, Binding and Invocation of a Service
- Java API for UDDI

23

---

---

---

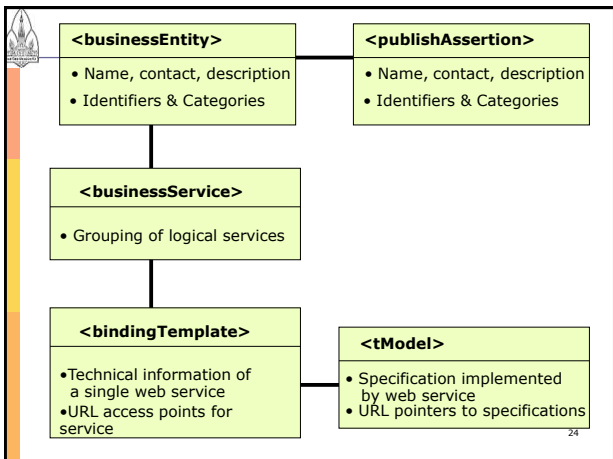
---

---

---

---

---




---

---

---

---

---

---

---

---

### UDDI Data Types

- Business Entity
  - White Pages information
- Business Services
  - Yellow Pages information
- Binding Templates
  - Green Pages information
  - Contains references to tModels
- tModels
  - Service Type Definitions
  - Interface specification for a service (WSDL)

25

---

---

---

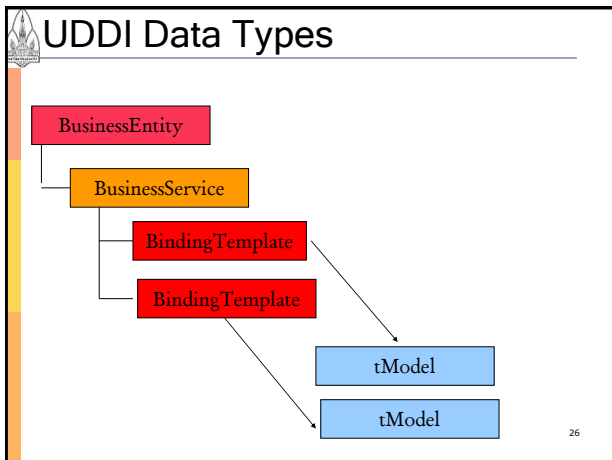
---

---

---

---

---




---

---

---

---

---

---

---

---

### Business Entity Data Type (1/2)

- Top-level data structure that holds descriptive information about a business entity
- Service descriptions and technical information are expressed within a businessEntity
- Contains categoryBag

- businessEntity
  - businessKey
  - name
  - URL
  - description
  - contacts
  - businessServices
  - identifierBag
  - categoryBag

27

---

---

---


---

---

---

---

---



### Business Entity Data Type (2/2)

- UDDI allows companies to establish relationships with one another
- Many different types of relationships are possible
- Each company must establish a unique <businessEntity> and separately establish its relationships to other companies that have their own <businessEntity> structures

28

---

---

---


---

---

---

---

---



### Publisher Assertion

- The <publisherAssertion> structure is used to establish public relationships between two <businessEntity> structures
- A relationship between two <businessEntity> structures is visible only to the “public” when both companies have created the same assertion with two separate <publisherAssertion> documents independently

29

---

---

---


---

---

---

---

---



### Business Service (1/2)

- A <businessEntity> contains one or more <businessService> structures
- A <businessService> represents a single, logical service classification
- A <businessService> element is used to describe a set of services provided by the business
- These services can be web services or manual services such as a nonelectronic service

30

---

---

---

---

---

---

---

---

### Business Service (2/2)

- A <businessService> document is reusable (i.e., a <businessService> element can be used by several <businessEntity> elements)
- A <businessService> contains
  - Name (service name)
  - Description (service description)
  - One or more <bindingTemplate> structures

31

---

---

---

---

---

---

---

---

### <businessService> Example

```

<businessService businessKey="..."
  serviceKey="..." <!-- Unique key -->
  <name>StockQuoteService</name>
  <description>...</description>
  <bindingTemplates>
    <bindingTemplate>
      ...
    </bindingTemplate>
  </bindingTemplates>
</businessService> <!-- Name of the service -->
    
```

32

---

---

---

---

---

---

---

---

### Binding Template

- A <bindingTemplate> contains pointers to technical descriptions and the access point URL
  - But does not contain the details of the service's specifications
- A <bindingTemplate> contains
  - An optional text description of the web service
  - The URL of its access point
  - A reference to one or more <tModel> structures

33

---

---

---

---

---

---

---

---

### Binding Template Example

```

<businessService businessKey="..." serviceKey="...">
  <name>StockQuoteService</name>
  <description>...</description>
  <bindingTemplates>
    <bindingTemplate>
      ...
      <accessPoint
        urlType="http">http://example.com/stockquote</accessPoint
      >
        <tModelInstanceDetails>
          <tModelInstanceInfo tModelKey="..." />
        </tModelInstanceDetails>
      </bindingTemplate>
    </bindingTemplates>
</businessService>
    
```

Specify Network endpoint address

Contains references to tModel

34

---

---

---

---

---

---

---

---

---

---

### tModel

- Service type definition
- Expected to be created by industry consortium (as opposed to business entities)
  - Business entities create businessEntity's, businessService's, and bindingTemplate's
- Shared by business entities
- Has a reference to WSDL document
- Enables quick search of all "business entities" which support a particular service

35

---

---

---

---

---

---

---

---

---

---

### tModel Example

```

<tModel authorizedName="..." operator="..." tModelKey="...">
  <name>StockQuote Service</name>
  <description xml:lang="en">
    WSDL description of a standard stock quote service
    interface
  </description>
  <overviewDoc>
    <description xml:lang="en"> WSDL source document.
    </description>
    <overviewURL> http://stockquote-definitions/stq.wsdl
    </overviewURL>
  </overviewDoc>
  <categoryBag>
    <keyedReference tModelKey="UUID:..."
      keyName="uddi-org:types"
      keyValue="wsdlSpec"/>
  </categoryBag>
</tModel>
    
```

36

---

---

---

---

---


---

---

---

---

---



### Categorization

- <businessService> document contains a <categoryBag> structure
- <categoryBag> documents can appear with <businessEntity>, <businessService>, and <tModel> documents
- Each taxonomy categorization is registered as a <tModel> structure within UDDI
- This registration means that each categorization has a tModel name and UUID that can be used to reference it

37

---

---

---


---

---

---

---

---



### Agenda

- What is UDDI and Why UDDI?
- UDDI Data Types and their structural relationship
- **UDDI Programming Interface**
- SOAP over UDDI
- Authoring Steps of WSDL, UDDI
- Discovery, Binding and Invocation of a Service
- Java API for UDDI

38

---

---

---


---

---

---

---

---



### Publishing to a UDDI Registry

- Publishing to a UDDI registry involves any operation that would create, update, or destroy data in a UDDI registry
- Some key technical differences between publishing and inquiring
  - Authenticated access
  - Different access point
  - Space limits
  - Operator node binding

39

---

---

---


---

---

---

---

---



### Authenticated Access

- All publishing messages require authenticated access
- The process for authentication is not defined by the UDDI specification and is specific to the operator node
- Given authenticated credentials
  - Your program can access any publishing message

40

---

---

---


---

---

---

---

---



### Different Access Point

- Publishing message requests use a different access point than do inquiry messages
- The HTTP protocol was suitable for inquiry messages
- The HTTPS is required for all publishing messages

41

---

---

---


---

---

---

---

---



### Space Limits

- Operator nodes can impose space and registration restrictions on an individual or company
- A site may limit some users to one <businessEntity> structure and prevent them for inserting additional data without specific permissions

42

---

---

---


---

---

---

---

---



### Operator Node Binding

- When information inserted into an operator node, that site becomes the owner of that data's master copy
- Any subsequent updates or changes to the data must be performed at the same operator node
- UDDI does not have a mechanism for resolving conflicts if duplicate entries are made at another operator node

43

---

---

---


---

---

---

---

---



### Security and Authentication

- Authentication with an operator node is typically straightforward
- Most operator nodes implement a name/password scheme that allows you to retrieve an authentication token
- An operator node also has specific ways of registering new publishers and verifying their information

44

---

---

---


---

---

---

---

---



### Programmer's API: Publishing Services

- Publishers interface
  - Save things
    - save\_business
    - save\_service
    - save\_binding
    - save\_tModel
  - Delete things
    - delete\_business
    - delete\_service
    - delete\_binding
    - delete\_tModel
  - Security..
    - get\_authToken
    - discard\_authToken
- **4 messages to save each of the 4 structures**
  - Each save message accepts as input the authToken and one or more corresponding structures.
- **4 messages to delete each of the 4 core structures**
  - They all accept the corresponding **uuid** key as the parameter.
- **Security:**
  - request an authentication token
  - nform registry that the authToken is no longer valid.

45

---

---

---

---

---

---

---

---

**Browsing Basic Information**

- A series of messages allow a program to retrieve basic information about
  - A business
  - A web service
  - Metadata about a specification that a web service supports
- These messages all have SOAP messages whose XML body element begins with “find”

46

---

---

---

---

---

---

---

---

**Programmer’s API: Service Discovery**

<ul style="list-style-type: none"> <li>□ Inquiry interface                     <ul style="list-style-type: none"> <li>■ Find things                             <ul style="list-style-type: none"> <li>□ find_business</li> <li>□ find_service</li> <li>□ find_binding</li> <li>□ find_tModel</li> </ul> </li> <li>■ Get details                             <ul style="list-style-type: none"> <li>□ get_businessDetail</li> <li>□ get_serviceDetail</li> <li>□ get_bindingDetail</li> <li>□ get_tModelDetail</li> </ul> </li> <li>■ Taxonomy interface                             <ul style="list-style-type: none"> <li>□ validate_categorization</li> </ul> </li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>□ Browse                     <ul style="list-style-type: none"> <li>■ 4 messages to <b>find</b> each of the 4 structures</li> </ul> </li> <li>□ Drill-down                     <ul style="list-style-type: none"> <li>■ The get call can be used to get information regarding a specific instance of any of the 4 data types, given the key</li> </ul> </li> </ul>
--	---

47

---

---

---

---

---

---

---

---

**Agenda**

- What is UDDI and Why UDDI?
- UDDI Data Types and their structural relationship
- UDDI Messages
- SOAP over UDDI
- Authoring Steps of WSDL, UDDI
- Discovery, Binding and Invocation of a Service
- Java API for UDDI

48

---

---

---

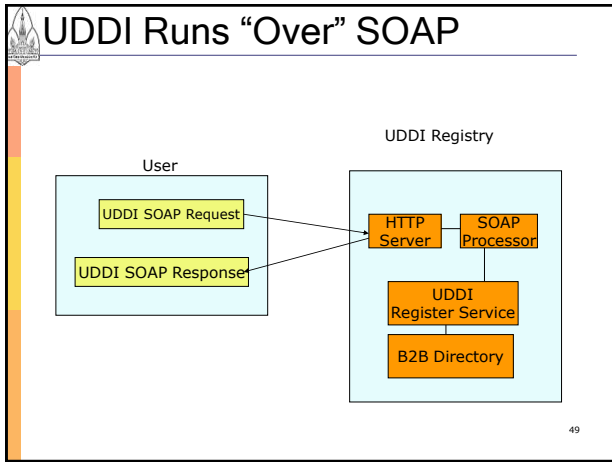
---

---

---

---

---




---

---

---

---

---

---

---

---

### SOAP Request Message Example

```

<Envelope>
  <Body>
    <get_serviceDetail generic="1.0">
      <serviceKey>6FD77EF6-E7D6-6FF6-1E41-
        EBC80107D7B5
      </serviceKey>
    </get_serviceDetail>
  </Body>
</Envelope>
    
```

50

---

---

---

---

---

---

---

---

### SOAP Response Message Example

```

<Envelope>
  <Body>
    <serviceDetail generic="1.0" operator="XMethods">
      <businessService serviceKey="6FD77EF6-E7D6-6FF6-1E41-EBC80107D7B5">
        <businessKey>D1387DB1-CA06-24F8-46C4-86B5D895CA26</businessKey>
        <name>Currency Exchange Rate</name>
        <description>Endpoint for service</description>
        <description>IMPLEMENTATION: ghee</description>
        <description>CONTACT EMAIL: support@xmethods.net</description>
        <bindingTemplate>
          <bindingTemplate bindingKey="0036DEBC-2F1B-E8B4-09E2-3A4332C3E8B4">
            <serviceKey>6FD77EF6-E7D6-6FF6-1E41-EBC80107D7B5</serviceKey>
            <description>SOAP binding</description>
            <accessPoint URLType="http">http://services.xmethods.net:80/soap/accessPoint</accessPoint>
          </bindingTemplate>
          <ModelInstanceDetails>
            <ModelInstanceInfo ModelKey="unif:3D784C184-9902-DA25-ED45-3665D11A12E5">
              </ModelInstanceDetails>
            </bindingTemplate>
          </bindingTemplate>
          </businessService>
        </serviceDetail>
      </Body>
    </Envelope>
    
```

51

---

---

---


---

---

---

---

---

 Using WSDL Definitions with UDDI (1/3)

- WSDL is used to describe the interface of a web service
- <tModel> UDDI documents
  - Provide metadata descriptions of a web service
  - Pointers to specifications that describe their implementation
- A <tModel> document should be created for each WSDL document supported by a web service

52

---

---

---


---

---

---

---

---

 Using WSDL Definitions with UDDI (2/3)

- The WSDL document's URL should be listed as the value of the <overviewURL> element
- A <tModel> that references a WSDL document should have a categorization taxonomy of uddi-org:types
- A categorization value of wsdSpec should be applied to it by using a <categoryBag> element

53

---

---

---


---

---

---

---

---

 Using WSDL Definitions with UDDI (3/3)

- A <bindingTemplate> structure is created for each unique URL access point used by the web service
- The <bindingTemplate> document references one or more <tModel> documents containing the WSDL definitions supported at this access point
- A <businessService> document is created for each web service
- The document contains one <bindingTemplate> for each of the access points supported by the web service

54

---

---

---

---

---

---

---

---

### WSDL Service Implementation vs. UDDI Registry

<pre> &lt;definitions name="   StockQuoteService"   targetNamespace="http://..."&gt;   &lt;import namespace="http://..."   location="http://..."&gt;   &lt;service     name="StockQuoteService"&gt;   &lt;port name="     SingleSymbolService"     binding="iface:SingleSymbolBindi     ng"&gt;   ... &lt;/service&gt; &lt;/definitions&gt; </pre>	<pre> &lt;businessEntity businessKey="..."&gt;   &lt;name&gt;Stock Quote Service,   Inc.&lt;/name&gt;   ...   &lt;businessService serviceKey="..."   &lt;name&gt;StockQuoteService &lt;/name&gt;   ...   &lt;bindingTemplates&gt;   &lt;bindingTemplate bindingKey="..."&gt;   ...   &lt;tModelInstanceInfo tModelKey   ="..."&gt;   ...   &lt;overviewDoc&gt;   &lt;overviewURL&gt;   http://.../SQS.wsdl   &lt;/overviewURL&gt;   ...   &lt;/bindingTemplates&gt;   &lt;/businessService&gt; &lt;/businessEntity&gt; </pre>
--	---

55

---

---

---

---

---

---

---

---

---

---

---

---

### Agenda

- ▣ What is UDDI and Why UDDI?
- ▣ UDDI Data Types and their structural relationship
- ▣ UDDI Categorization
- ▣ UDDI Programming Interface
- ▣ SOAP over UDDI
- ▣ **Authoring Steps of WSDL, UDDI**
- ▣ Discovery, Binding and Invocation of a Service
- ▣ Java API for UDDI

56

---

---

---

---

---

---

---

---

---

---

---

---

### Steps that could be Performed by Industry Consortium

- ▣ Create WSDL document that contains abstract part of service definition (WSDL interface definition)
- ▣ Create *tModel* that
  - makes a URL reference to WSDL interface definition
  - includes category information
  - can be shared by many business entities
- ▣ Register the *tModel* to UDDI registry

57

---

---

---

---

---

---

---


---

---

---

---

---

 Steps that are Performed by Business Entities

- Find *tModel* for a particular service to offer from the UDDI registry
- Determine the port address
- Create *bindingTemplate* that
  - Contains the port address
  - Makes a reference to the previously found *tModel*
- Create *businessService* that refers to the *bindingTemplate*
- Create *businessEntity* if necessary

58

---

---

---


---

---

---

---

---

 Agenda

- What is UDDI and Why UDDI?
- UDDI Data Types and their structural relationship
- UDDI Categorization
- UDDI Programming model
- Authoring Steps of WSDL, UDDI
- **Discovery, Binding and Invocation of a Service**
- Java API for UDDI

59

---

---

---


---

---

---

---

---

 Discovering of a Service

- Programmatically
  - via Categorization (Yellow paging)
  - via identity information (White paging)
  - via drill-down
  - Via name patterns
- Through UDDI Browser

60

---

---

---


---

---

---

---

---



### Binding to and Invocation of a Service

- Obtain WSDL interface information from the *tModel*
- Obtaining port address from *bindingTemplate*
- Construct WSDL instance definition (WSDL document with concrete binding and port address)
- Create service proxy from WSDL
- Invocation pattern
  - Cache the *bindingTemplate* info for a service
  - If call to web service fails, re-check info in UDDI

61

---

---

---


---

---

---

---

---



### Agenda

- What is UDDI and Why UDDI?
- UDDI Data Types and their structural relationship
- UDDI Categorization
- UDDI Programming Interface
- SOAP over UDDI
- Authoring Steps of WSDL, UDDI
- Discovery, Binding and Invocation of a Service
- **Java API for UDDI**

62

---

---

---


---

---

---

---

---



### Java-Based APIs

- Using a Java-based SOAP API
  - API that creates SOAP messages containing a UDDI XML document
- Using a custom Java-based UDDI client API
  - Clients API for accessing a UDDI registry
- Using JAXR
  - JAXR allows developers to write code that can access several different registries, including UDDI and the ebXML Registry/Repository

63

---

---

---


---

---

---

---

---



### Programming UDDI

- Two APIs are described by the UDDI specification
  - The inquiry API
  - The publishing API
- They are accessed using the same techniques
- But they use different XML documents, data structures, and access points

64

---

---

---


---

---

---

---

---



### The Inquiry API

- The inquiry API locates
  - Information about a business
  - The services a business offers
  - The specifications of those services
  - Information about what to do in a failure situation
- Any read operation from a UDDI registry uses one of the inquiry API's messages
- The inquiry API does not require authenticated access and is subsequently accessed using HTTP

65

---

---

---


---

---

---

---

---



### The Publishing API

- The publishing API is used to
  - Create, store, or update information located in a UDDI registry
- All functions in this API require authenticated access to a UDDI registry
- The UDDI registry must have a logon identity
- The security credentials for this identity must be passed as a parameter of the XML document for each UDDI invocation

66

---

---

---


---

---

---

---

---

 **UDDI 3.0 and Future of UDDI**

- Digital signature
- Publisher assigned key
  - Maintains the same key when copy a UDDI entry from one UDDI registry to another
- Multi-registry support
- DNS style key format (instead of UUID format)
- Policy

67

---

---

---


---

---

---

---

---

 **References**

- UDDI
  - <http://www.uddi.org>
- OASIS UDDI TC
  - <http://www.oasis-open.org/committee/uddi-spec/>
- Using UDDI WSDL in a UDDI Registry
  - <http://www.uddi.org/pubs/wsdlbestpractices-V1.05-Open-20010625.pdf>
- UDDI Browser from Java Web Services Development Pack (JWSDP)
  - <http://java.sun.com/webservices>
- Web Services Course Programming
  - <http://www.javapassion.com/webservices>

68

---

---

---

---

---

---

---

---