UDDI Overview

- Realised as UDDI registries
- Enables businesses providing services to register information to enable their discovery
  - “white pages” – address, contact & known identifiers
  - “yellow pages” – industrial categorizations using standard taxonomies
  - “green pages” – technical information about exposed services
- Enables web service clients to discover potential services
- Enables businesses to discover
  - Potential business partners
  - Potential services to make use of

UDDI in the Web Service Architecture

History/Standards

- Started by IBM, Microsoft & Ariba
- OASIS (Organization for the Advancement of Structured Information Standards) open standard
  - UDDI version 2 – April 2003
  - UDDI version 3 – Feb. 2005
  - See also http://uddi.xml.org/ community site
- See also Java API for XML Registries (JAX-R)

UDDI registries

- May be
  - Public – available to anyone
  - Private – operated for one particular group of participants, often with more security
- Each is independent by default
  - different UDDI key namespaces
- Can be accessed as a web service
  - possibly among other methods
- May be physically distributed across multiple nodes
  - Which logically provide interfaces to the same information (the same registry)
### UDDI APIs

- A UDDI registry node implements one or more UDDI APIs
  - Inquiry
  - Publication
- And for UDDI 3:
  - Security
  - Custody transfer
  - Subscription
    - With client Subscription Listener API
  - Replication
- Specified in terms of XML schema types
  - For use with SOAP document/literal style

### UUIDs

- All structures (except publisherAssertion) are identified by 128-bit globally unique identifier also known as UUID.
  - E.g. uuid:23453aeaf-af35-6a3f-c34a-bf798dab965a
- These UUIDs can later be used as keys to access the specific data within the registry.

### <businessEntity>

- Primary information about a business
  - Contact information
  - Categorization of business in Specific taxonomy or classification scheme(s)
  - Identifiers
  - Relationships to other business entities
    - E.g. partners, business-to-customer
    - Must be asserted by BOTH business entities through their own <publisherAssertion>s before the relationship is visible
    - So that a business cannot unilaterally claim a relationship that does not actually exist

### <publisherAssertion>

- Used to represent a relationship between <businessEntity>s
- Relationship may be of any kind
  - E.g. partners, business-to-customer
- Must be asserted by BOTH business entities through their own <publisherAssertion>s before the relationship is visible
  - Represents a service of a business
  - Abstract representation
  - Contained by a <businessEntity>
  - Can be referenced by other <businessEntity>s
<bindingTemplate>
• Contained by a <businessService>
• Describes the service in terms of
  – Pointers to technical descriptions
    • i.e. <tModel>s
  – Access URLs
    • E.g. SOAP endpoint URLs

Additional data structures
• <identifierBag>
  – for <businessEntity> or <tModel>
  – E.g. D-U-N-S numbers (9 digit business identifier), tax IDs
• <categoryBag>
  – for <businessEntity>, <businessService> or <tModel>
  – Categorisation in some system/taxonomy
    • (see next slides)
  – Support enquiry

Standard classification schemes
  – NAICS (North American Industry Classification System)
    • e.g. 424920=Book, periodical and newspaper merchant wholesalers
    • http://www.census.gov/epcd/www/naics.html
  – UNSPSC (United Nations Standard Products and Services Classification)
    • E.g. 55101523=Exercise books
    • http://www.unspsc.org/
  – ISO 3166 geographic coding standard
    • E.g. 826/GBR/GB=United Kingdom, GB-NGM=Nottingham (unitary authority)
    • http://en.wikipedia.org/wiki/ISO_3166

Categorization in UDDI registries
• (aka “classification”)
• Esp. useful for searching/discovery
• Extensible framework, including
  – Standard schemes - see next slide
  – Arbitrary additional schemes
    • Identified by <tModel>s
• Categorisation scheme may be:
  – “Checked” – checked with categorisation organisation during publishing
  – “Unchecked” – no checking performed

Inquiry API
• Used to find matching entities:
  – <find_business>, <find_relatedBusinesses>,
  <find_service>, <find_binding>, <find_tModel>
• Or to get detailed information:
  – <get_businessDetail>,
  <get_businessDetailExt>, <get_serviceDetail>, <get_bindingDetail>,
  <get_tModelDetail>
• Specified as XML schema elements
Find_xx Inquiry API functions

- Support browse, search and narrow
- Returns abbreviated information
  - Including UDDI key
  - Used to then get detailed information

Example request

```xml
<uddi:find_business generic="2.0" maxRows="10">
  <uddi:name>
  ACM%
  </uddi:name>
</uddi:find_business>
```

Example response

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV=
  "http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <businessList xmlns="urn:uddi-org:api_v2"
      generic="2.0" operator="SYSTINET">
      <businessInfos>
        <businessInfo businessKey=
          "uuid:23453aef-af35-6a3f-c34a-
          bf798dab965a">
          <name xml:lang="en">
            ACME Computer Services
          </name>
          <description xml:lang="en">
            Provides professional services ...
          </description>
        </businessInfo>
      </businessInfos>
    </businessList>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example response (cont.)

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV=
  "http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <businessList xmlns="urn:uddi-org:api_v2"
      generic="2.0" operator="SYSTINET">
      <businessInfos>
        <businessInfo businessKey=
          "uuid:523f3aef-af35-6a3f-c34a-
          bf798dab965a">
          <name xml:lang="en">
            Billing Services
          </name>
          <description xml:lang="en">
            Provides professional services ...
          </description>
        </businessInfo>
      </businessInfos>
    </businessList>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

<find_business> operation

- Request syntax (UDDI 2.0):
  ```
  <find_business [maxRows="nn"]
              generic="2.0"
xmlns="urn:uddi-org:api_v2">
  <findQualifiers/>
  <name/>
  <discoveryURLs/>
  <identifierBag/>
  <categoryBag/>
  <tModelBag/>
</find_business>
  ```

- Supports query syntax (% wildcard, OR)
- See earlier notes (NR)
- Applies to binding as fingerprint (AND)

<findQualifiers> element

- List of <findQualifier> enumeration, e.g.
  ```
  <findQualifiers> 
  <findQualifier caseSensitiveMatch
  </findQualifier> 
  </findQualifiers>
  ```
- Includes: exactNameMatch, caseSensitiveMatch, sortByNameAsc, sortByNameDesc, sortByDateAsc (default), sortByNameDesc
<find_relatedBusinesses> operation

- Return UUIDs of related businesses (!)
  - Given a reference businessKey (UUID)
  - Optional findQualifiers
  - Optional keyedReferences

Example response

```xml
<serviceList generic="2.0" operator="SYSTINET" xmlns="urn:uddi-org:api_v2">
<serviceInfos>
  <serviceInfo serviceKey= "uuid:1245sdef-af35-6a3f-c34a-bf798da965a" businessKey= "uuid:23453aef-af35-6a3f-c34a-bf798da965a">
    <name>
      Billing Services
    </name>
  </serviceInfo>
</serviceInfos>
</serviceList>
```

Example Response

```
<bindingDetail generic="2.0" operator="SYSTINET" xmlns="urn:uddi-org:api_v2">
  <bindingTemplate bindingKey="uuid:acd5sdef-1235-6a3f-c34a-bf798dab124a" serviceKey="uuid:1245sdef-af35-6a3f-c34a-bf798da965a">
    <accessPoint URLType="http">
      http://www.acmecomputerservices.com/billingservices_entry/
    </accessPoint>
    <tModelInstanceDetails>
      <tModelInstanceInfo tModelKey= "uuid:acd5sdef-1235-6a3f-c34a-bf798dab124b">
        <description>
          Provides SOAP Interface. Described by BillingServices_WSDL.wsdl.
        </description>
      </tModelInstanceInfo>
    </tModelInstanceDetails>
  </bindingTemplate>
</bindingDetail>
```

<find_service> operation

- Request syntax (UDDI 2.0):
  ```xml
  <find_service businessKey="uuid_key" [maxRows="nn"] generic="2.0" xmlns="urn:uddi-org:api_v2">
    [findQualifiers/]
  [name/][<name/>]...
  [categoryBag/]
  [tModelBag/]
  </find_service>
  ```

Example response

```
<serviceList generic="2.0" operator="SYSTINET" xmlns="urn:uddi-org:api_v2">
<serviceInfos>
  <serviceInfo serviceKey= "uuid:1245sdef-af35-6a3f-c34a-bf798da965a" businessKey= "uuid:23453aef-af35-6a3f-c34a-bf798da965a">
    <name>
      Billing Services
    </name>
  </serviceInfo>
</serviceInfos>
</serviceList>
```

<find_binding> operation

- Syntax:
  ```xml
  <find_binding serviceKey="uuid_key" [maxRows="nn"] generic="2.0" xmlns="urn:uddi-org:api_v2">
    [findQualifiers/]
  [tModelBag/]
  </find_binding>
  ```

Example Response

```
<bindingDetail generic="2.0" operator="SYSTINET" xmlns="urn:uddi-org:api_v2">
  <bindingTemplate bindingKey="uuid:acd5sdef-1235-6a3f-c34a-bf798dab124a" serviceKey="uuid:1245sdef-af35-6a3f-c34a-bf798da965a">
    <accessPoint URLType="http">
      http://www.acmecomputerservices.com/billingservices_entry/
    </accessPoint>
    <tModelInstanceDetails>
      <tModelInstanceInfo tModelKey= "uuid:acd5sdef-1235-6a3f-c34a-bf798dab124b">
        <description>
          Provides SOAP Interface. Described by BillingServices_WSDL.wsdl.
        </description>
      </tModelInstanceInfo>
    </tModelInstanceDetails>
  </bindingTemplate>
</bindingDetail>
```

<find_tModel> operation

- Request syntax:
  ```xml
  <find_tModel [maxRows="nn"] generic="2.0" xmlns="urn:uddi-org:api_v2">
    [findQualifiers/]
    [name/]
    [identifierBag/]
    [categoryBag/]
  </find_tModel>
  ```

Example Response

```
<bindingDetail generic="2.0" operator="SYSTINET" xmlns="urn:uddi-org:api_v2">
  <bindingTemplate bindingKey="uuid:acd5sdef-1235-6a3f-c34a-bf798dab124a" serviceKey="uuid:1245sdef-af35-6a3f-c34a-bf798da965a">
    <accessPoint URLType="http">
      http://www.acmecomputerservices.com/billingservices_entry/
    </accessPoint>
    <tModelInstanceDetails>
      <tModelInstanceInfo tModelKey= "uuid:acd5sdef-1235-6a3f-c34a-bf798dab124b">
        <description>
          Provides SOAP Interface. Described by BillingServices_WSDL.wsdl.
        </description>
      </tModelInstanceInfo>
    </tModelInstanceDetails>
  </bindingTemplate>
</bindingDetail>
```
Example Response
“Developing Java Web Services”

```xml
<tModelList generic="2.0" operator="SYSTINET"
xmlns="urn:uddi-org:api_v2">
<tModelInfos>
<tModelInfo
tModelKey= "uuid:acd5sdef-1235-6a3f-c34a-bf798dad24b">
<name>
SOAP_WSDL_BillingServices
</name>
</tModelInfo>
</tModelInfos>
</tModelList>
```

Get_xx inquiry API functions

- **Support drill-down**
  - Typically getting more information for specific entities
  - Takes UUID as key
- **Operations:**
  - `<get_businessDetail>` → `<businessDetail>`
  - `<get_serviceDetail>` → `<serviceDetail>`
  - `<get_bindingDetail>` → `<bindingDetail>`
  - `<get_tModelDetail>` → `<tModelDetail>`

### <get_businessDetail> operation

- **Example request:**
  ```xml
  <uddi:get_businessDetail generic="2.0">
  <businessKey>
  23453aef-af35-6a3f-c34a-bf798dab965a
  </businessKey>
  </uddi:get_businessDetail>
  ```

Example response
“Developing Java Web Services”

```xml
<businessDetail generic="2.0" operator="SYSTINET"
xmlns="urn:uddi-org:api_v2">
<businessEntity authorizedName = "John Smith"
businessKey="uuid:23453aef-af35-6a3f-c34a-bf798dab965a"
operator="SYSTINET">
<discoveryURLs>
<discoverURL useType="businessEntity">
http://www.systinet.com/wasp/uddi/discovery?businessKey=
23453aef-af35-6a3f-c34a-bf798dab965a
</discoverURL>
</discoveryURLs>

<name> ACME Computer Services
</name>
<description xml:lang="en">
Provides professional services in the areas of computer software
</description>

<contacts>
<contact useType="information">
<description xml:lang="en">For sales related information
</description>
<personName> Joe Smith
</personName>
<address>
1, Computer Drive, Burlington, MA 01803 USA
</address>
</contact>
</contacts>
</businessEntity>
</businessDetail>
```

### Example response

```xml
<businessService businessKey="23453aef-af35-6a3f-c34a-bf798dab965a"
serviceKey="1245sdef-af35-6a3f-c34a-bf798dab965a">

<name xml:lang="en">Billing Services
</name>
<description xml:lang="en">Billing Services
</description>

<bindingTemplates> ...
</bindingTemplates>
</businessService>
```

See next slide
<bindingTemplate

bindingKey= "uuid:acd5sdef-1235-6a3f-c34a-bf798dab124a"

serviceKey="1245sdef-af35-6a3f-c34a-bf798dab965a">

<description xml:lang="en">
Here is where you should be visiting to get started with using billing services provided by us.
</description>

<accessPoint URLType="http">
http://www.acmecomputerservices.com/
billingservices_entry/
</accessPoint>

</bindingTemplate>

See next slide

See next slide

WSDL and UDDI

• See e.g.
http://www.oasis-open.org/committees/uddi-
spec/doc/tn/uddi-spec-tc-tn-wsdl-v2.htm

• tModels identify
  – WSDL portType
  – WSDL Binding

• businessService represents WSDL service
  • bindingTemplate represents WSDL port

Publisher API

• User to publish/change/delete the information in a UDDI registry
  – <save_business>, <delete_business>,
    <save_service>, <delete_service>,
    <save_binding>, <delete_binding>,
    <save_tModel>, <delete_tModel>

• To manage relationship assertions
  – <get/add/set_publisherAssertions>,
    <get_assertionStatusReport>

• <get_registeredInfo> - get businessEntity & tModels owned by this user
Publisher API Security

- All operations require authentication:
  - `<get_authToken>` - takes username/password, return `<authToken>` for subsequent use
  - `<discard_authToken>` - releases/invalidates `<authToken>`

Limitations

- Simple information model
  - e.g. little/no business oriented information such as business process models
  - cf. ebXML
- Registry only, i.e. Stores only metadata
  - cf. ebXML Registry/repository, also stores submitted objects