



UDDI
nDDI



Barry d'Hoine | Shirin Ravash | Zahra Ahmadi

INFO-H-511 Web Services 2013



Outline

Online

- Story so far
- Registry and discovery
- UDDI in general
- Structure of UDDI
- Registration
- Categorization
- Importing WSDL documents
- Queries in UDDI
- Issues and alternatives



Story So Far

Web Services Technology Stack



- HTTP protocol and URI's
- REST architecture
- RESTful web services
- SOAP protocol
- WSDL format



Service Discovery

- The process of finding a suitable web service for given task
- Service Discovery
 - Publish descriptions
 - Search for resource descriptions based on semantic criteria
 - By agents and end-user
- Dynamic discovery
 - Interact directly with the discovery service
- Static discovery
 - Interact with the discovery service through an appropriate software agent, such as a browser



Service Registry

- Central part of service oriented architecture (SOA)
- Allow developers or end-user to register and locate useful web services
- Storing web service descriptions such as businesses, services and technical information persistently
- Contains metadata about web services, information about service providers, consumers and their relationships



Why Registration?

- Global, platform-independent, open framework
- Reuse of existing services
- Organize information about services
- Provide facilities to publish and discover services
- Define how to interact over the Internet



Private/Corporate Registry

- Expose web services to developers on your own server
- Internal registry, behind a firewall, is isolated from the public network
- A business may set up multiple private registries
- Data is not shared with other registries
- Access to both administrative features and registry data is restricted
- Enterprise Web Service Registry



Public Registry

- Global directory for public web services access
- Used after developing and deploying a web service
- Developers and clients can discover it
- Access to registry data is open and public
- Universal Business Registry (UBR) maintained by IBM, Microsoft SAP and NTT
 - Similar to the Domain Name Service (DNS) that the Internet infrastructure has



Document & Meta Based Registry

- Document based registry
 - Publish information by storing XML-based service documents
 - Technical specifications (WSDL descriptions)
- Meta based registry
 - Meaningful information in descriptive document by provider
 - Stored in service registry
 - Describe the structure of information like XML schema, interface definitions or endpoints, etc.



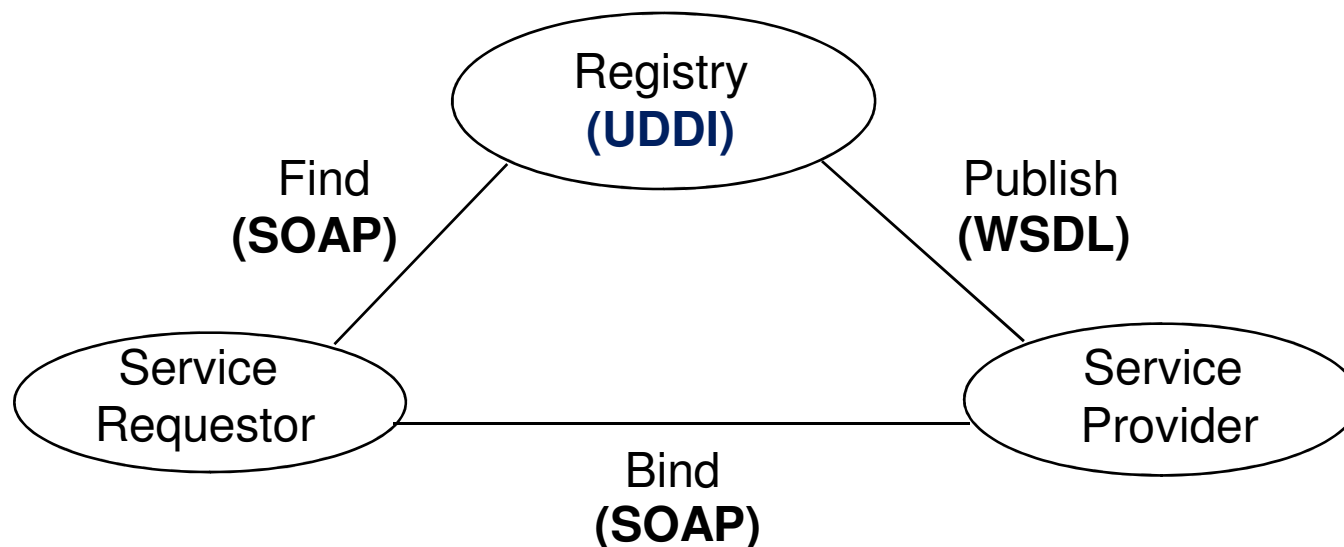
What Problems Do We Solve?

- Broader B2B (describe services)
 - easier integration into and across marketplaces
- Smarter search (discover services)
 - programmatic discovery of web services
- Easier aggregation (integrate services)
 - programmatically combine services from multiple businesses



What is UDDI

- Stands for **U**niversal **D**escription, **D**iscovery and **I**ntegration
- UDDI is used to
 - **Describe** a business and its services
 - **Discover** other businesses offering desired services
 - **Integrate** with these other businesses





UDDI History

2000	UDDI.org Formed by Ariba, IBM, and Microsoft
2002	UDDI.org Transitions Work to OASIS
2003	UDDI 2.0 Ratified as OASIS Standard
2005	UDDI 3.0.2 Ratified as OASIS Standard
2006	UDDI XML.org Focus Area Launched



UDDI Registry Data

White
Pages

Yellow
Pages

Green
Pages

Interrelated components



UDDI Registry Data

White
Pages

Yellow
Pages

Green
Pages

Who the business entity represents & What they do

Where to find the service they provide

How to access the service they provide



UDDI Registry Data

White
Pages

Yellow
Pages

Green
Pages

- Business name
- Human readable descriptions and identifiers
- Address and contact info
 - names, phone numbers, fax numbers, web sites, etc.



UDDI Registry Data

White
Pages

Yellow
Pages

Green
Pages

- Categorization of businesses using standard taxonomies
 - Industrial classification
 - Geographical indexes



UDDI Registry Data

White
Pages

Yellow
Pages

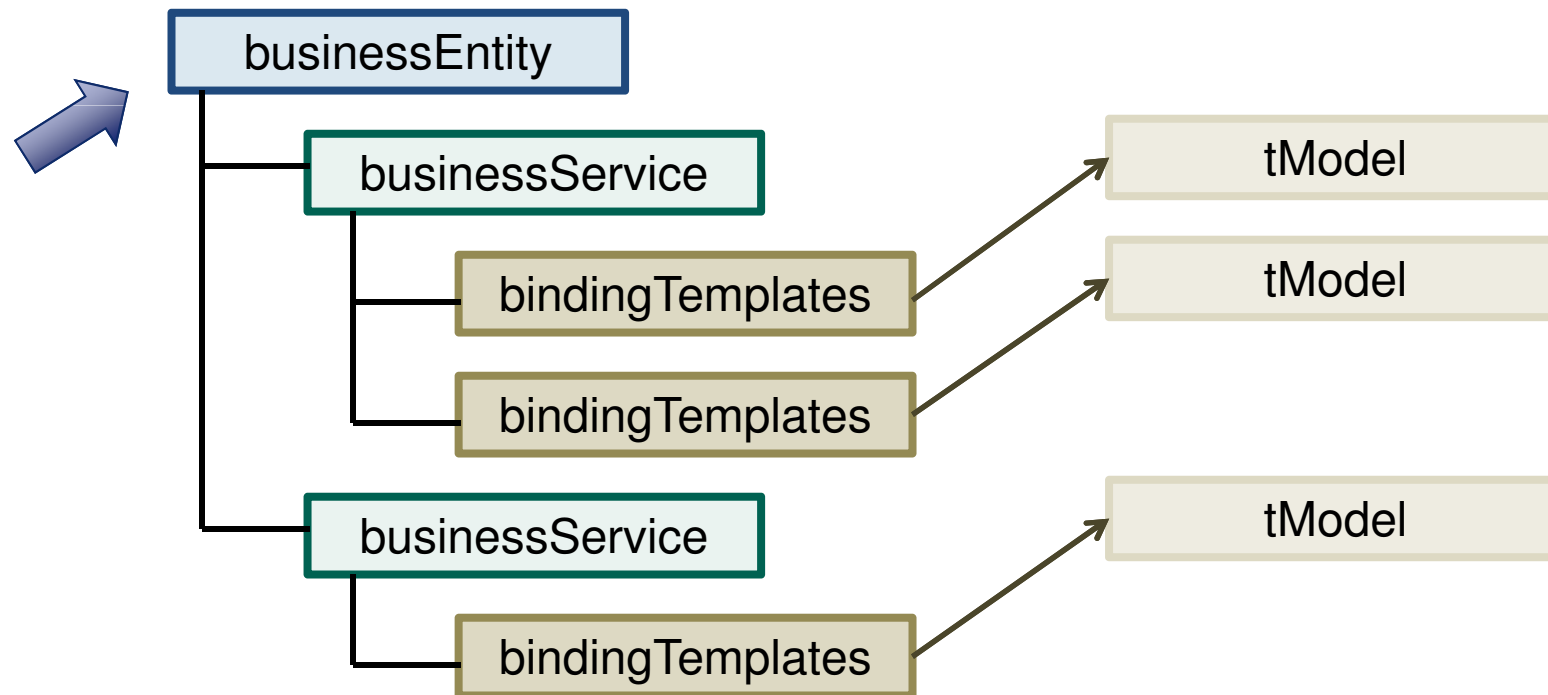
Green
Pages

- Technical information about the published services
 - Service description
 - Business rules
 - Application invocations



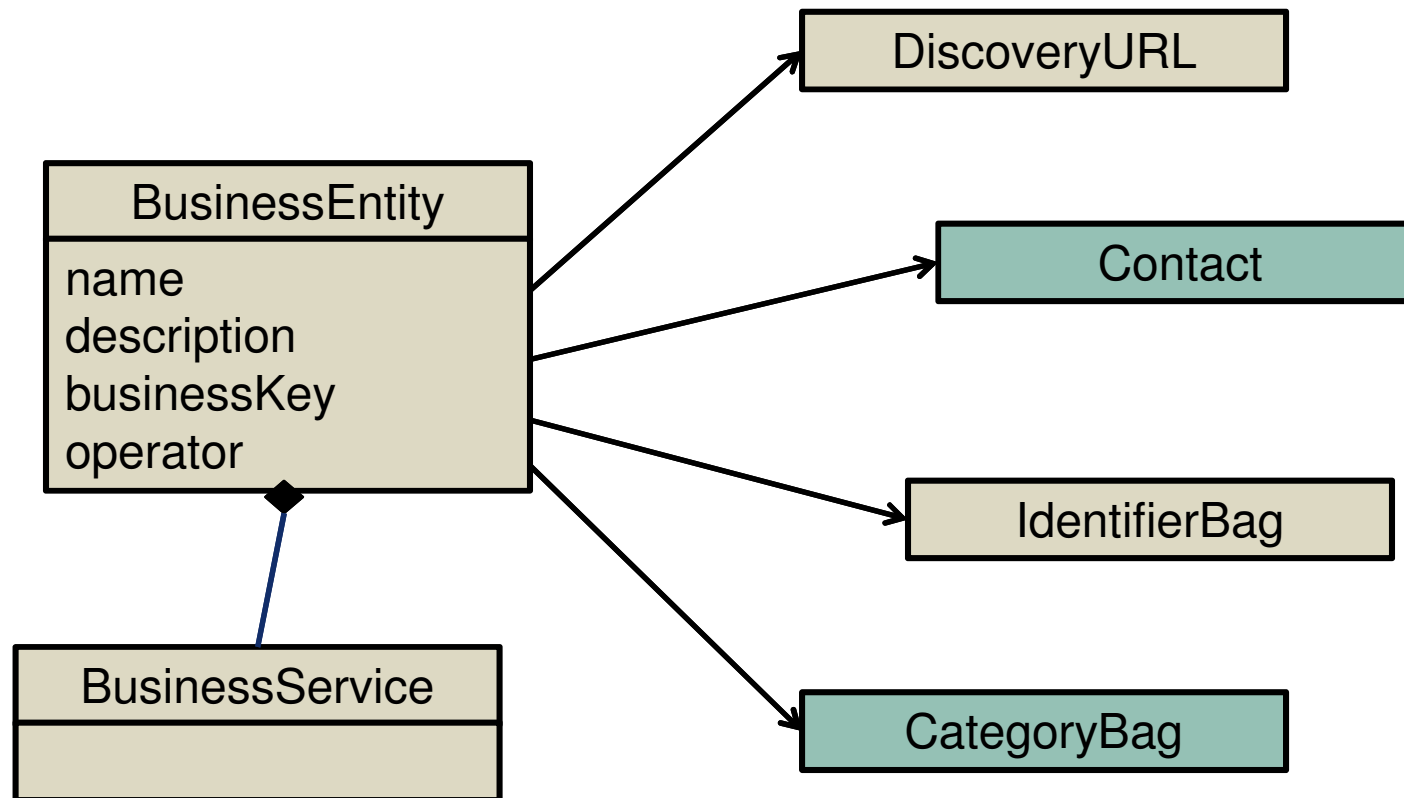
UDDI Data Structures

- Described by XML schema
- Universal Unique Identifier (UUID) as key





BusinessEntity





businessEntity Example

```
<businessEntity
  businessKey="uuid:c0e6d5a8-c446-4f01-99da-70e212685a40"
  operator="WS"
  authorizedName="John Doe">
  <name>Web Services Ltd</name>
  <description>
    We create cool Web services
  </description>
  <contacts>
    <contact useType="general info">
      <description>General Information</description>
      <personName>John Doe</personName>
      <phone>+32 (0)2 6501234</phone>
      <email>jdoe@vub.ac.be</email>
    </contact>
  </contacts>
</businessEntity>
```

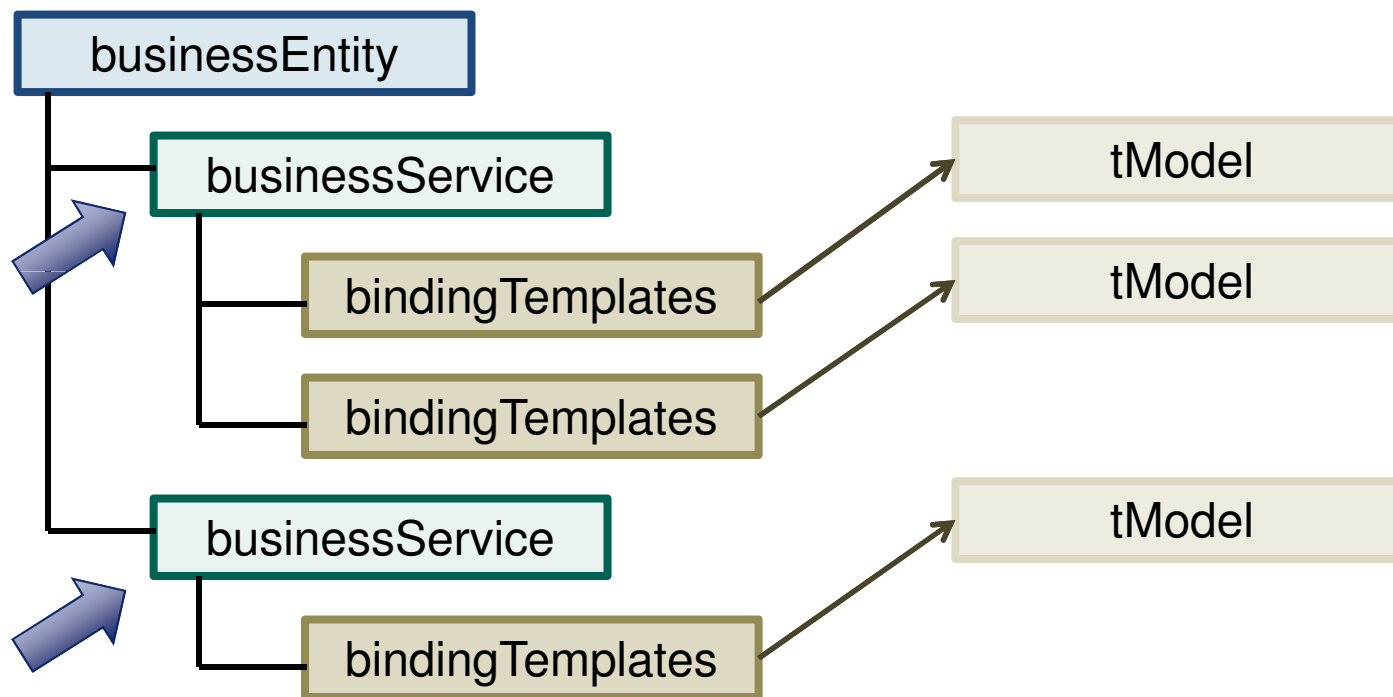


businessEntity Example

```
<businessEntity>
  <businessServices>
    ...
  </businessServices>
  <identifierBag>
    <keyedReference
      tModelKey="uuid:c0e6d5a8-c446-4f01-99da-70e212685a40"
      name="D-U-N-S"
      value="123456789" />
    </identifierBag>
    <categoryBag>
      <keyedReference
        tModelKey="uuid:c9e6d5a8-1234-4f01-ce46-685a470e2120"
        name="NAICS"
        value="111336" />
      </categoryBag>
    </businessEntity>
```

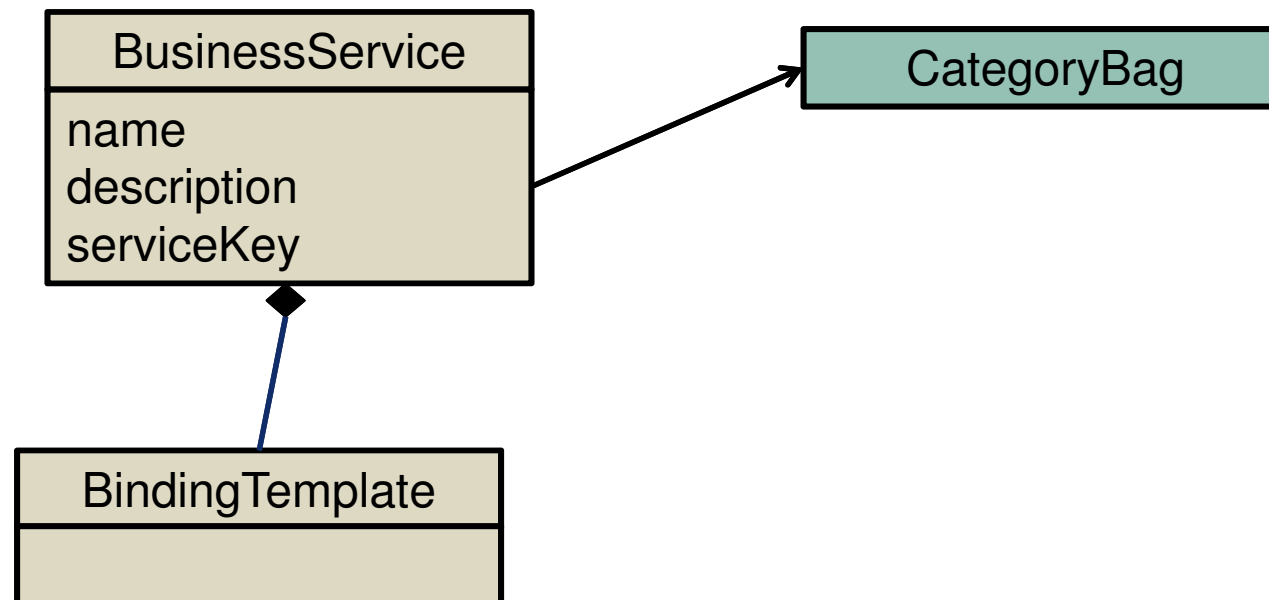


businessService





BusinessService



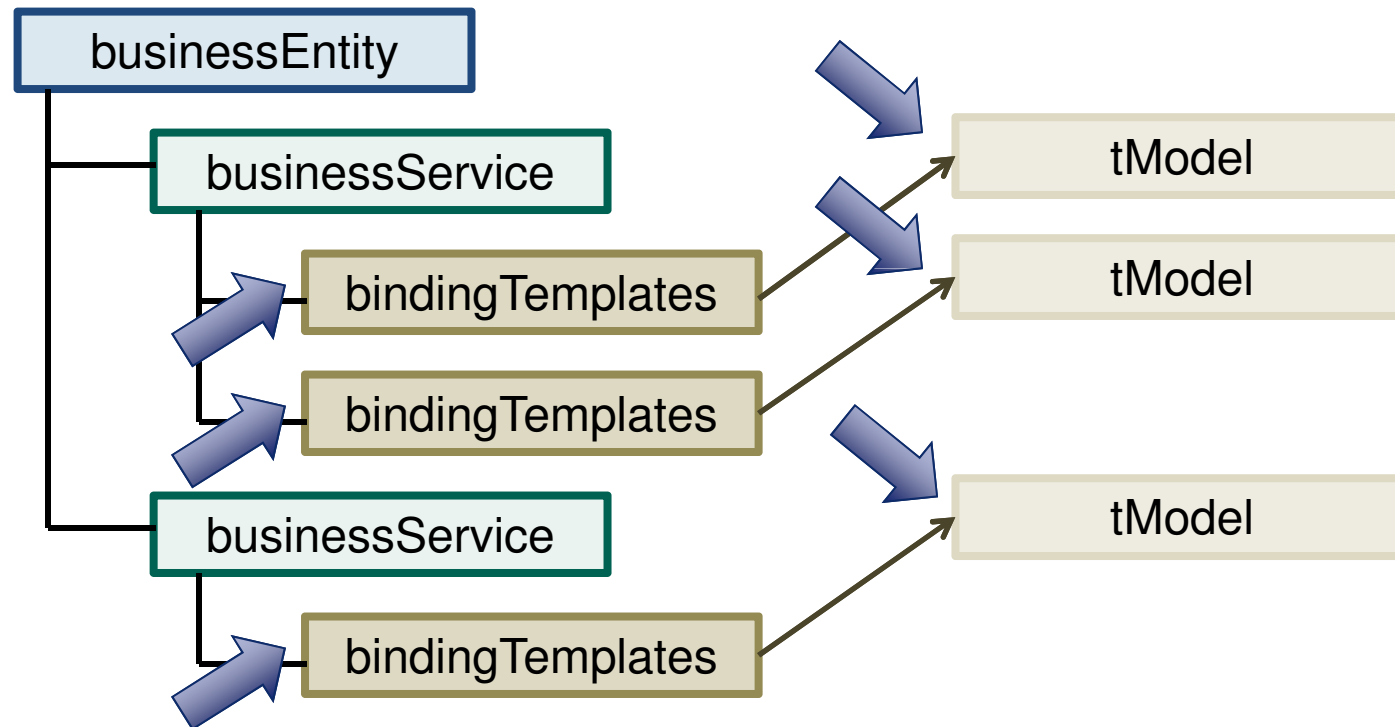


businessService Example

```
<businessService
  serviceKey="uuid:f26dc1ac-f6e2-9d70-ed04-b756e392ab46"
  businessKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4">
  <name>Hello World Web Service</name>
  <description>A friendly Web service</description>
  <bindingTemplates>
    ...
  </bindingTemplates>
  <categoryBag />
</businessService>
```

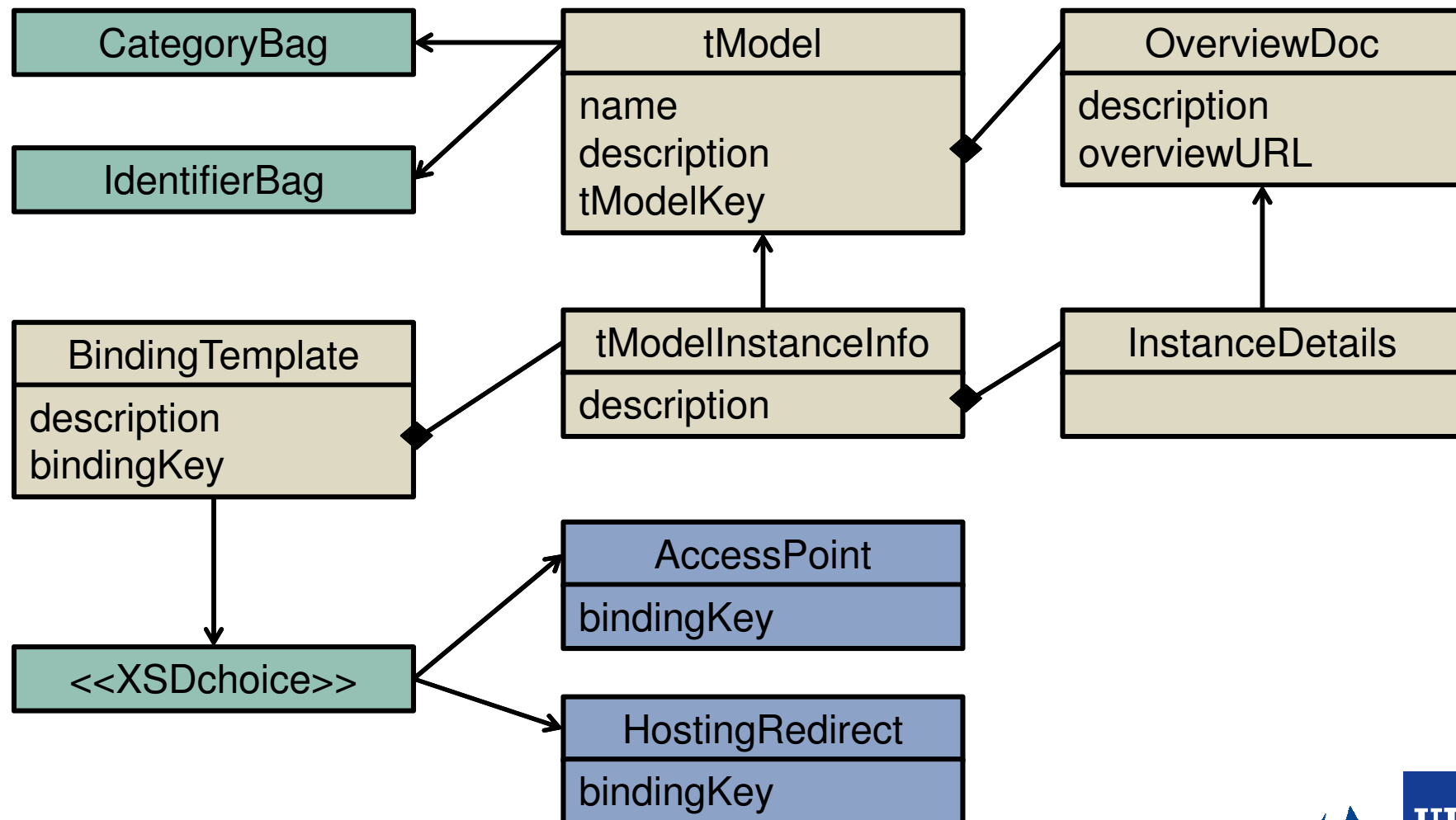



bindingTemplate & tModel





BindingTemplate & tModel





tModel Document

- Core data structure in the UDDI specification
- Represent the most detailed information
 - Technical fingerprint
 - Represent technical specifications
 - e.g. Wire protocols
 - Value sets
 - Specify organizational identity and various categories
 - Represents the system of values used to identify or categorize UDDI entities



tModel Example

```
<tModel tModelKey="uuid:aa254698-93de-3870-8df3-a5c075d64a0e">
  <name>uddi-org:protocol:soap</name>
  <description>A tModel for the SOAP 1.1 protocol</description>
  <overviewDoc>
    <overviewURL>
      http://.../uddi-spec-tc-tn-wsdl-v2.htm#soap
    </overviewURL>
  </overviewDoc>
  <categoryBag>
    <keyedReference
      tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4"
      keyName="uddi-org:types"
      keyValue="protocol" />
  </categoryBag>
</tModel>
```



Registration

- Use API calls in SOAP body to communicate with UDDI registry
 - Request authentication token
 - Get response with token
 - Get SOAP fault message
 - Register business to UDDI registry
 - Operator generates UUID
 - Get response with business info
 - Get SOAP fault message
- Registration complete
 - Update business info
 - Web service can be discovered and interacted with



Request authToken

```
<?xml version="1.0" encoding="UTF-8" ?>
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Body>
    <get_authToken
      generic="2.0"
      xmlns="urn:uddi-org:api_v2"
      userID="wscourse"
      cred="supersecretpassword" />
  </Body>
</Envelope>
```



Response authToken

```
<?xml version="1.0" encoding="UTF-8"?>
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Body>
    <authToken generic="2.0" xmlns="urn:uddi-org:api_v2">
      <authInfo>
        IHLMDYbBWFkbWluMB4XDTA×MTIzMDAwTIzMDA×NDYzNVoEDUFkb
        WaXN0cmF0b3IwDQYJKoZIhvcNAQEEBQADgYEA4Cci/CbDjiQFneRt7gVXwX/
        4TACZNUmTnXFJdVNFIDvp4WV+IW+/
      <authInfo>
    </authToken>
  </Body>
</Envelope>
```



Register

```
<?xml version="1.0" encoding="UTF-8"?>
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Body>
    <save_business generic="2.0" xmlns="urn:uddi-org:api_v2">
      <authInfo>
        IHLMDYbBWFkbWluMB4XDTA×MTIzMDAwTIzMDA×NDYzNVoEDUFkb
        WaXN0cmF0b3IwDQYJKoZIhvcNAQEEBQADgYEA4Cci/CbDjiQFneRt7gVXwX/
        4TACZNUmTnXFJdVNFIDvp4WV+IW+/
      <authInfo>
      <name>Web Services Course Ltd</name>
      <description>Register to UDDI as example</description>
      <identifierBag> ... </identifierBag>
      ...
    </save_business>
  </Body>
</Envelope>
```




Fault Response

```
<?xml version="1.0" encoding="UTF-8"?>
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Body>
    <Fault>
      <faultcode>Client</faultcode>
      <faultstring>Client Error</faultstring>
      <detail>
        <dispositionReport xmlns="urn:uddi-org:api_v2">
          <result errno="20240">
            <errInfo errCode="E_requestTimeout">
              The request is unanswered or did not respond in time
            </errInfo>
          </result>
        </dispositionReport>
      </detail>
    </Fault>
  </Body>
</Envelope>
```



Categorization

- Use **<categoryBag>** and **<keyedReference>** for categorization
- Make use of the schema's defined by *tModels*
 - *tModelKey* as identifier for *tModel*
 - Lookup *tModel* by calling **get_tModelDetail()**
 - *keyValue* is the meta data associated with the entity
 - *keyName* to provide context for human user
 - Except when using the *general_keyword* taxonomy
- Create own classification scheme
 - UDDI registry generates new *tModelKey*
- Checked and unchecked categorization



Categorization example

```
<?xml version="1.0" encoding="utf-8"?>
<businessEntity businessKey="..." xmlns="urn:uddi-org:api_v2">
  <name>Web Services Ltd</name>
  <categoryBag>
    <keyedReference
      tModelKey="uuid:c0b9fe13-179f-413d-8a5b-5004db8e5bb2"
      keyName="NAICS: Software Publisher"
      keyValue="51121" />
    <keyedReference
      tModelKey="uuid:4e49a8d6-d5a2-4fc2-93a0-0411d8d19e88"
      keyName="Brussels"
      keyValue="BE-BXL" />
  </categoryBag>
</businessEntity>
```



Create Categorization example

```
<?xml version="1.0" encoding="utf-8"?>
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Body>
    <save_tModel generic="2.0" xmlns="urn:uddi-org:api_v2">
      <tModel tModelKey="">
        <name>New Classification Scheme</name>
        <overviewDoc>
          <overviewURL>http://moreinfo</overviewURL>
        </overviewDoc>
        <categoryBag>
          <keyedReference
            tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4"
            keyValue="categorization" />
        </categoryBag>
      </tModel>
    </save_tModel>
  </Body>
</Envelope>
```



Checked Categorization

- UDDI registry performs validation on *keyValue*
- Knowledge of all *keyValues* in UDDI registry
 - Display listing of a categorization scheme
- Prevent inserting “bad” data
 - Returns SOAP fault message
- Drawbacks
 - Difficult to create predefined list of values
 - Versioning
 - What to do with removed values?



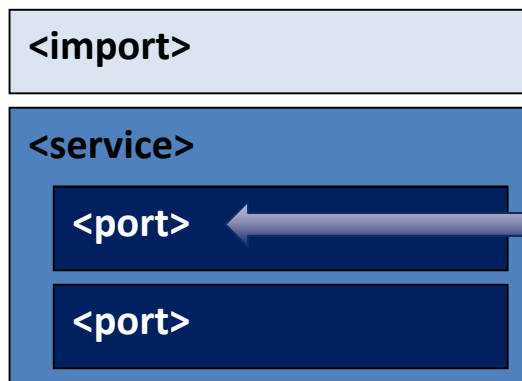
Unchecked Categorization

- UDDI registry cannot perform validation on *keyValue*
- Constrains for value
 - Valid XSD string
 - Less than 255 characters
- Meta data placed in *keyValue* can greatly vary
- Drawbacks
 - No listing of valid or suggested values
 - Insertion of “bad” data possible

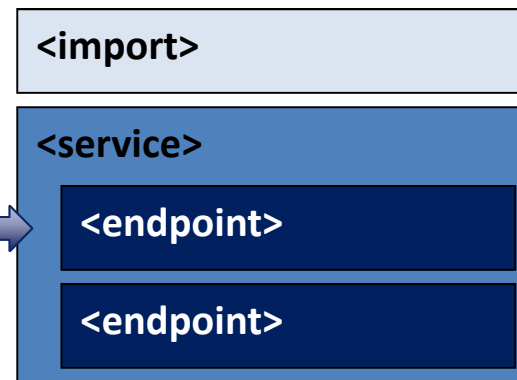


WSDL 1.1 vs 2.0

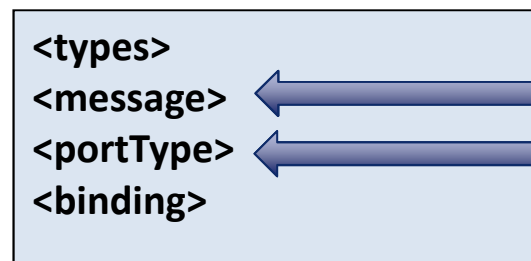
Concrete Section



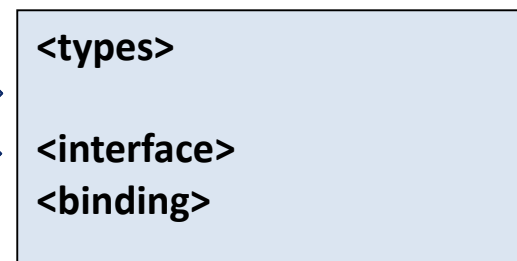
Concrete Section



Abstract Section



Abstract Section



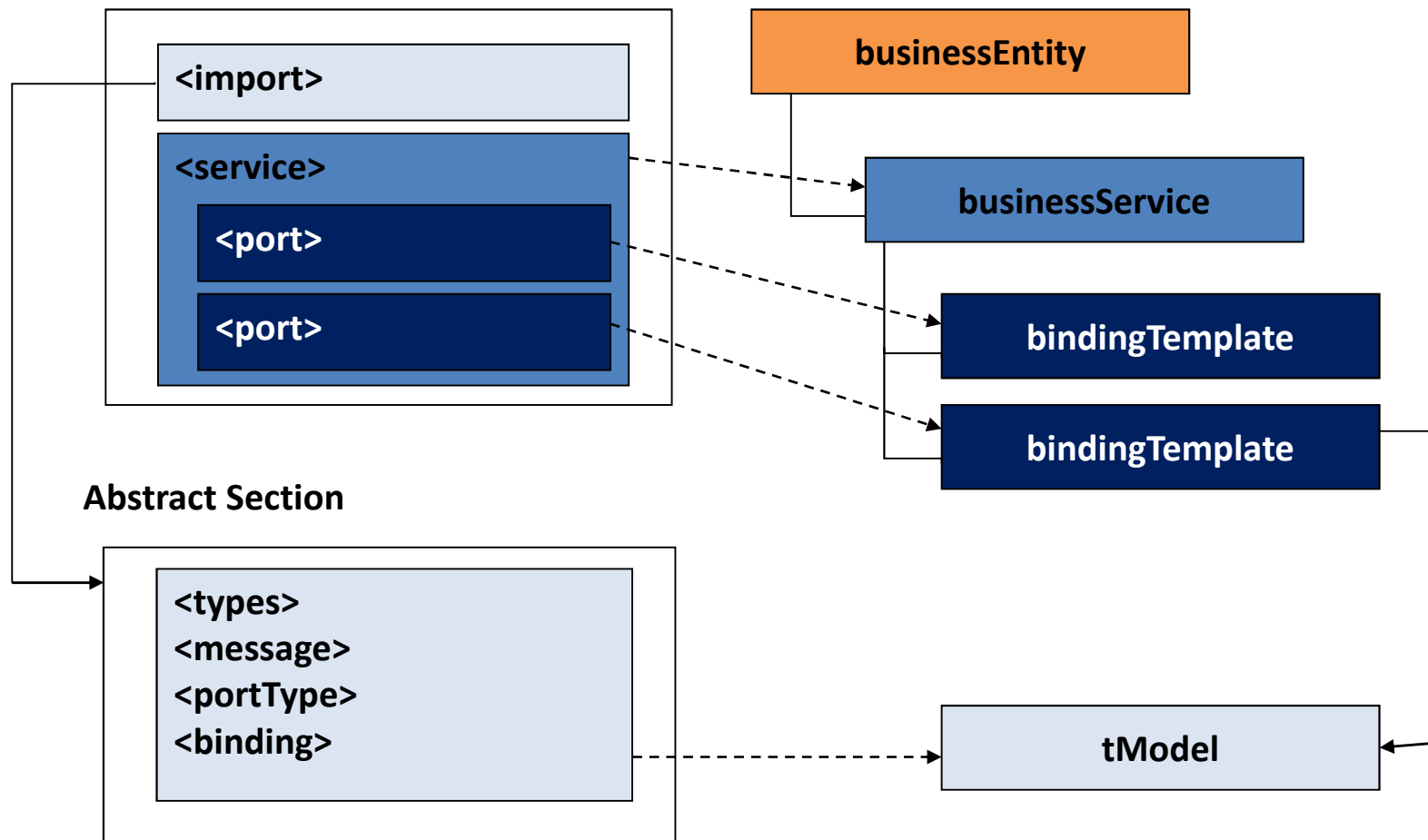
WSDL 1.1

WSDL 2.0



WSDL to UDDI mapping

Concrete Section



WSDL 1.1

UDDI



wsdl:portType to uddi:tModel

WSDL	UDDI
<i>portType</i>	<i>tModel</i> (categorized as <i>portType</i>)
Local name of <i>portType</i>	<i>tModel</i> name
Location of WSDL document	<i>overviewURL</i>
Namespace of <i>portType</i>	<i>keyedReference</i> in <i>categoryBag</i>



Example WSDL file to map

```
<?xml version="1.0" encoding="utf-8"?>
<definitions>
  <types /><message />
  <portType name="StockQuotePortType">
    <operation name="GetLastTradePrice">
      <input message="tns:GetLastTradePriceInput" />
      <output message="tns:GetLastTradePriceOutput" />
    </operation>
  </portType>
  <binding/>
  <service name="StockQuoteService">
    <port
      name="StockQuotePort"
      binding="tns:StockQuoteSoapBinding">
        <soap:address location="http://location/sample" />
      </port>
    </service>
  </binding>
</definitions>
```



Map portType

```
<?xml version="1.0" encoding="utf-8"?>
<definitions>
  <types /><message />
  <portType name="StockQuotePortType">
    <operation name="GetLastTradePrice">
      <input message="tns:GetLastTradePriceInput" />
      <output message="tns:GetLastTradePriceOutput" />
    </operation>
  </portType>
  <binding/>
  <binding name="tns:StockQuotesSoapBinding" type="tns:StockQuotePortType">
    <tModel tModelKey="uuid:e8cf1163-8234-4b35-865f-94a7322e40c3">
      ...
    </tModel>
    <soap:address location="http://location/sample" />
  </binding>
</definitions>
```





Map portType name

```
<?xml version="1.0" encoding="utf-8"?>
<definitions>
  <types /><message />
  <portType name="StockQuotePortType">
    <operation name="GetLastTradePrice">
      <input message="tns:GetLastTradePriceInput" />
      <output message="tns:GetLastTradePriceOutput" />
    </operation>
  </portType>
  <binding/>
  <service name="StockQuoteService">
    <binding name="tns:StockQuoteSoapBinding">
      <soap:address location="http://location/sample" />
    </binding>
  </service>
</definitions>
```

A blue arrow points from the `<portType name="StockQuotePortType">` line to a red-bordered box containing the following XML snippet:

```
<name>
  StockQuotePortType
</name>
```



Map port location

```
<?xml version="1.0" encoding="utf-8"?>
<definitions>
  <types /><message />
  <portType
    <overviewDoc>
      <overviewURL>
        http://location/sample.wsdl
      </overviewURL>
    </overviewDoc>
    <operation name="GetStockQuote" output="true" />
  </portType>
  <binding />
  <service name="StockQuoteService">
    <port
      name="StockQuotePort"
      binding="tns:StockQuoteSoapBinding">
      <soap:address location="http://location/sample" />
    </port>
  </service>
</definitions>
```



Final UDDI tModel

```
<tModel tModelKey="uuid:e8cf1163-8234-4b35-865f-94a7322e40c3">
  <name>StockQuotePortType</name>
  <overviewDoc>
    <overviewURL>http://location/sample.wsdl</overviewURL>
  </overviewDoc>
  <categoryBag>
    <keyedReference
      tModelKey="uuid:d01987d1-ab2e-3013-9be2-2a66eb99d824"
      keyName="portType"
      keyValue="http://example.com/stockquote/" />
    <keyedReference
      tModelKey="uuid:6e090afa-33e5-36eb-81b7-1ca18373f457"
      keyName="WSDL type"
      keyValue="portType" />
  </categoryBag>
</tModel>
```



Register a Web service as a tModel

- Service interface and service binding represent a reusable definition of a service => Publish as <tModel>
- It provides flexible design patterns for applications using Web services
- If service becomes unavailable, then client's application can dynamically query for another company



wSDL:binding to uddi:tModel

WSDL	UDDI
binding	<i>tModel</i> (categorized as <i>binding</i> & <i>wSDLSpec</i>)
Namespace of binding	<i>keyedReference</i> in <i>categoryBag</i>
Local name of binding	<i>tModel</i> name
Location of WSDL document	<i>overviewURL</i>
<i>portType</i> binding relates to	<i>keyedReference</i> in <i>categoryBag</i>
Protocol from binding extension	<i>keyedReference</i> in <i>categoryBag</i>
Transport from binding extension (if there is one)	<i>keyedReference</i> in <i>categoryBag</i>



wsdl:service to uddi:businessService

WSDL	UDDI
Service	<i>businessService</i> (categorized as service)
Namespace of Service	<i>keyedReference</i> in <i>categoryBag</i>
Local Name of Service	<i>keyedReference</i> in <i>categoryBag</i> Optionally also the name of the service



wsdl:port to uddi:bindingTemplate

WSDL	UDDI
<i>port</i>	<i>bindingTemplate</i>
Namespace	Captured in <i>keyedReference</i> of the containing <i>businessService</i>
Local Name of port	<i>instanceParms</i> of the <i>tModelInstanceInfo</i> relating to the <i>tModel</i> for the <i>binding</i>
<i>Binding</i> implemented by port	<i>tModelInstanceInfo</i> with <i>tModelKey</i> of the <i>tModel</i> corresponding to the <i>binding</i>
<i>portType</i> implemented by port	<i>tModelInstanceInfo</i> with <i>tModelKey</i> of the <i>tModel</i> corresponding to the <i>portType</i>



UDDI Inquiry API (1)

Query for	By Using	UDDI API
business entity	business name, business key or business category	find_business()
A list of publisher assertions	business key	find_relatedBusiness()
A business service	business key of service key and service name	find_service()
Service key of a business entity	a binding template	find_binding()
A set of business entities and business services adopting same tModel	a tModel	find_tModel()

UDDI Search API allows users to query for service providers that provide a particular service.



UDDI Inquiry API (2)

Query	UDDI API
Retrieving the complete <i>businessEntity</i> for each specified <i>businessKey</i>	get_businessDetail()
Retrieving the <i>businessService</i> record for each specified <i>serviceKey</i>	get_serviceDetail()
Retrieving the complete <i>bindingTemplate</i> for each specified <i>bindingKey</i>	get_bindingDetail()
Retrieving the <i>tModel</i> record for each specified <i>tModelKey</i>	get_tModelDetail()

After finding the required UDDI entry, a set of API calls is used to get details of those entries



UDDI Query: Example 1

- Finding business entities by name
- Performs an “OR” search (default case)

```
<?xml version="1.0" encoding="UTF-8"?>
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Body>
    <find_business xmlns="urn:uddi-org:api_v2" generic="2.0">
      <name xml:lang="en">
        Automotive Equipment Manufacturing Inc.
      </name>
      <name xml:lang="en">
        Manufacturing Goods Co.
      </name>
    </find_business>
  </Body>
</Envelope>
```



UDDI Query: Example 2

- Finding business entities by category
- Returns a list of businesses matching **<keyedReference>**

```
<?xml version="1.0" encoding="UTF-8"?>
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Body>
    <find_business xmlns="urn:uddi-org:api_v2" generic="2.0">
      <categoryBag>
        <keyedReference
          tModelKey="uuid:6e090afa-33e5-36eb-81b7-1ca18373f457"
          keyValue="A92-D" />
      </categoryBag>
    </find_business>
  </Body>
</Envelope>
```



UDDI and RESTful

UDDI and RESTful

- WSDL 2.0 is not fully interoperable but can be used
 - Also not recommended and adopted by vendors
- Web Application Description Language (WADL)
 - REST answer to WSDL 1.1
- Create a tModel for one or both to register their taxonomy
- Register RESTful web service
 - Discovery now possible



Alternatives

- DNS for REST Web Service Discovery
 - Widely available for developers and administrators
- WSO2 Governance Registry
 - Open source, integrated SOA registry-repository,
 - Manage your organization's growing SOA efficiently
 - Easy-to-use metadata repository
 - Support for full versioning
 - Rich model for users
 - Social features such as tagging, rating and comments
- Enterprise Service Bus (ESB)
 - Acts as a directory service in the same spirit as UDDI



UDDI issues

- Relies on a standards-based mechanism to classify, catalog and manage web services
- Complex
- Not needed for most web services
- UDDI is not fully dead but it only lives in some private internal registries
- Multiple matches