

# UDDI

---

A Web Services Seminar

Ahmed ABDEEN  
Géraud TALLA FOTSING

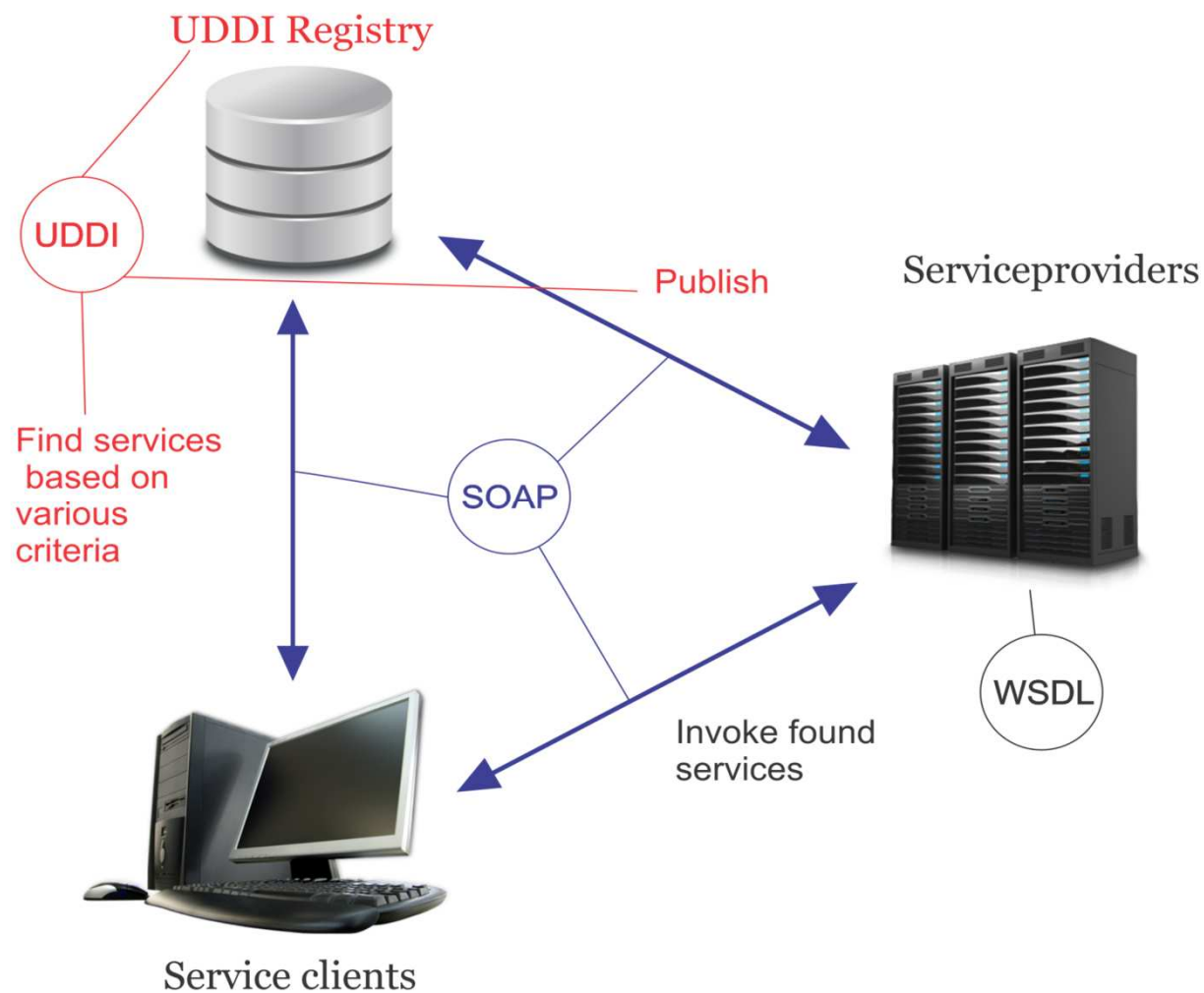
# Content

- Introduction
- UDDI Semantics & Structure
- WSDL to UDDI mapping model
- UDDI Categorization
- UDDI API: Publishing and Enquiry
- UDDI & RESTful web services
- UDDI Alternatives

# Introduction

- Service Registry
  - A global, platform-independent, open framework
  - Goal: To enable organizations to
    - Discover each other
    - Define how they interact over the Internet
    - Share information
- UDDI = Universal Description, Discovery and Integration
  - XML-based registry for companies and individuals to list Web Services
  - Sponsored by the Organization for the Advancement of Structured Information Standards (OASIS)
  - Founded by Microsoft, Ariba and IBM (2000)

# Introduction



# UDDI Structure

- A UDDI Business registration consists of 3 components

## White Pages

Name  
Address  
Contacts  
Identifiers  
...

## Yellow Pages

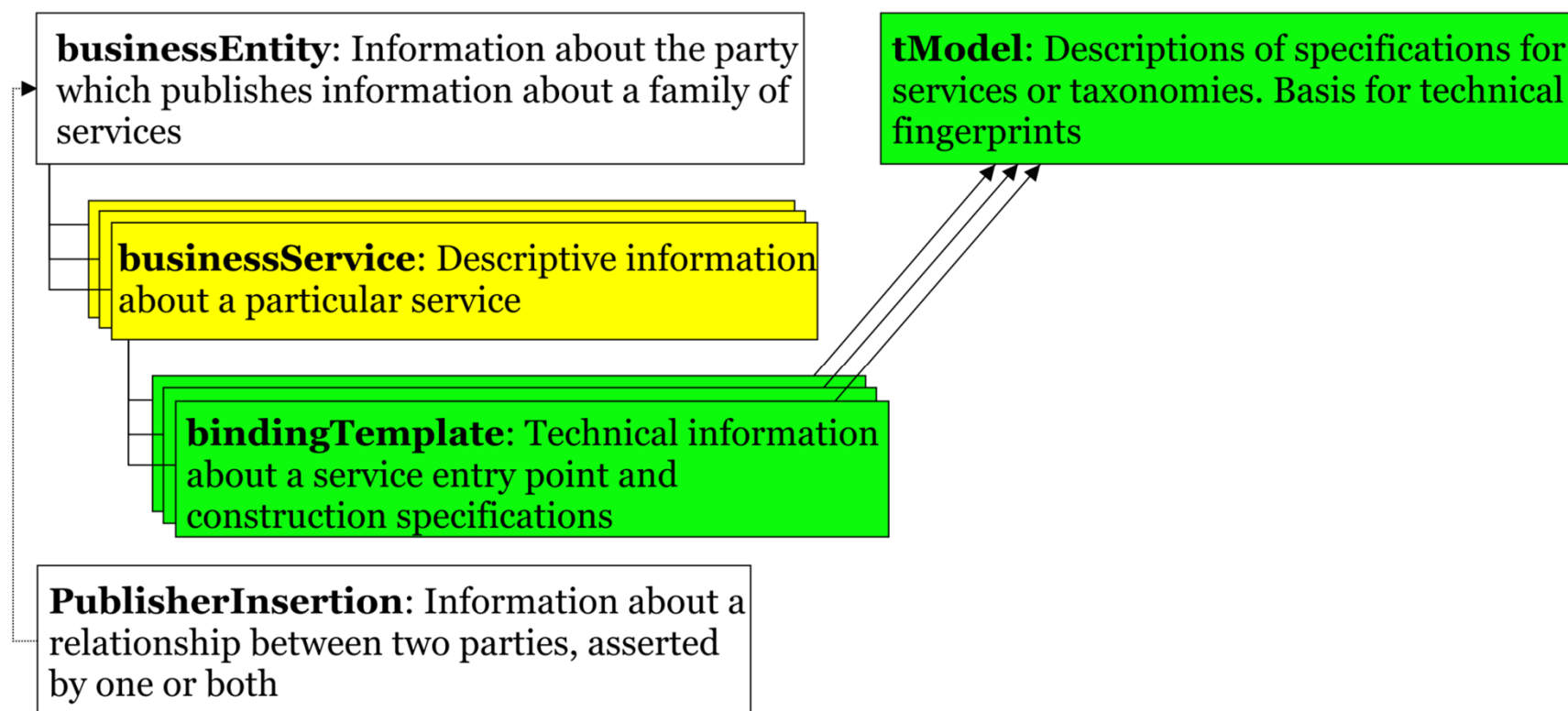
Industrial  
categorizations  
based on industry  
taxonomies

## Green Pages

Technical  
information about  
services exposed by  
the business

# UDDI Structure

- Four core data structure types
- Defined in an XML Schema



# UDDI Structure

- Service provider information

## A simple <businessEntity> example

```
<?xml version="1.0" encoding="utf-8"?>
<businessEntity businessKey="A687FG...">
  <name>Acme Travel Incorporated</name>
  <description xml:lang="en">Acme is a world leader in online travel services</description>
  <contacts>
    <contact useType="US general">
      <personName>Acme Inc.</personName>
      <phone>1 800 CALL ACME</phone>
      <email useType="">acme@acme-travel.com</email>
      <address> ... </address>
    </contact>
  </contacts>
  <businessServices> ... </businessServices>
  <identifierBag> ... </identifierBag>
  <categoryBag>
    <keyedReference tModelKey="UUID:DB..." keyName="E-check-in" keyValue="84121801"/>
  </categoryBag>
</businessEntity>
```

# UDDI Structure

- Web service description information

## A simple <businessService> example

```
<?xml version="1.0" encoding="utf-8"?>
<businessService serviceKey="894B5100..." businessKey="D203311...">
  <name>ElectronicTravelService</name>
  <description xml:lang="en">Electronic Travel Service</description>
  <bindingTemplates>
    <bindingTemplate bindingKey="6D665B..." serviceKey="89470B...">
      <description>SOAP-based e-checkin and flight info </description>
      <accessPoint URLType="http">
        http://www.acme-travel.com/travelservice
      </accessPoint>
      <tModelInstanceDetails>
        <tModelInstanceInfo tModelKey="D20...">...</tModelInstanceInfo>
      </tModelInstanceDetails>
    </bindingTemplate>
  </bindingTemplates>
  <categoryBag> ... </categoryBag>
</businessService>
```



# UDDI Structure

- Web service access and technical information

## A simple <bindingTemplate> example

```
<?xml version="1.0" encoding="utf-8"?>
<bindingTemplate bindingKey="...">
  <description lang="en">Use your web browser to search the parts registry</description>
  <accessPoint URLType='http'>
    http://www.automeq.com/b2b/actions/search.jsp
  </accessPoint>
  <tModelInstaceDetails>
    <tModelInstanceInfo tModelKey='uddi:...' />
  </tModelInstaceDetails>
</bindingTemplate>
```

# UDDI Structure

- Web service access and technical information

## A simple <tModel> example

```
<?xml version="1.0" encoding="utf-8"?>
<tModel tModelKey="">
  <name>http://www.travel.org/e-checkin-interface</name>
  <description xml:lang="en">
    Standard service interface definition for travel services
  </description>
  <overviewDoc>
    <description xml:lang="en">WSDL Service Interface Document</description>
    <overviewURL>http://www.travel.org/services/e-checkin.wsdl</overviewURL>
  </overviewDoc>
  <categoryBag> ... </categoryBag>
</tModel>
```

# UDDI Structure

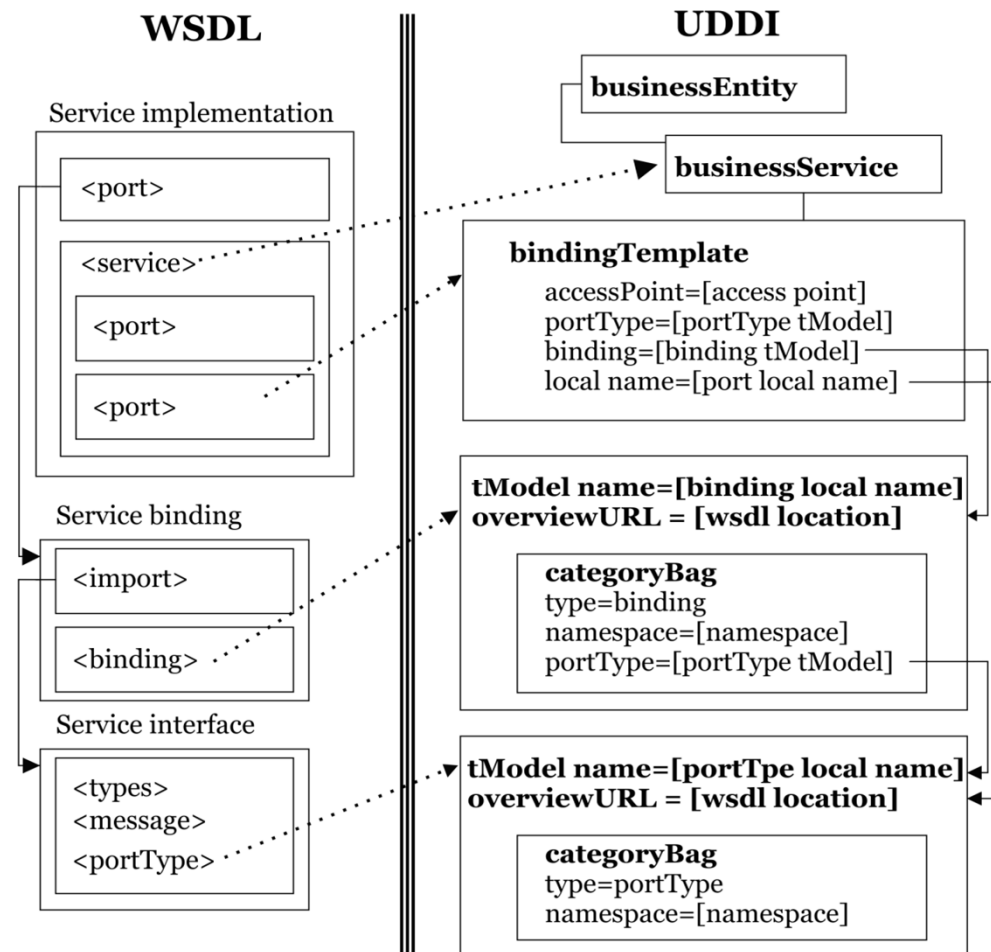
- The publisher assertion
  - An enterprise may have multiple <businessEntity>'s
    - Ex: Multinational organizations (divisions, departments, ...)
  - Defines relationships between pairs of <businessEntity> structures

## Example of a sample <publisherAssertion> entry

```
<publisherAssertion>  
  <fromKey> FR565 ... </fromKey>  
  <toKey> A237B ... </toKey>  
  <keyedReference tModelKey='uuid:80A..' keyName='subsidiary'  
    keyValue='parent-child' />  
</publisherAssertion>
```

# WSDL to UDDI mapping model

- WSDL and UDDI distinguish between interface and implementation



# UDDI Categorization

- Very important in UDDI
- Clients can search for entities based on multiple properties
- Concerns «providers», «services» and «tModels»
- 3 categorization taxonomies in the core part of UDDI
  - North American Industry Classification System (NAICS)
  - Universal Standard Products and Services Classification (UNSPSC)
  - International standard for geographical regions (ISO-3166)

# UDDI Categorization

## Example

```
<?xml version="1.0" encoding="utf-8"?>
<businessEntity businessKey="..." xmlns="urn:uddi-org:api_v2">
  <name>Company Name</name>
  <categoryBag>
    <keyedReference tModelKey="uuid:cob..." keyName="NAICS: Software Publisher"
    keyValue="51121" />
    <keyedReference tModelKey="uuid:4e4..." keyName="California" keyValue="US-CA" />
    ....
  </categoryBag>
</businessEntity>
```

```
<?xml version="1.0" encoding="utf-8"?>
<tModel tModelKey="uuid:cob...">
  <name>ntis-gov:naics:1997</name>
  ....
</tModel>
```

```
<?xml version="1.0" encoding="utf-8"?>
<tModel tModelKey="uuid:4e4...2">
  <name>uddi-org:iso-ch:3166:1999</name>
  ....
</tModel>
```

# UDDI Categorization

- Two types of categorization
  - Checked
    - ❖ UDDI registry will perform validation routine on «keyValue»
    - ❖ If the test fails, the registry will throw a SOAP fault
  - Unchecked
    - UDDI registry does not do any contextual check on «keyValue»

# The UDDI API : Publishing API

- Used by enterprises to manage information contained in a

## Example : Creating a New Classification Schemes

```
<?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">
      <authInfo>...</authInfo>
      <tModel tModelKey="">
        <name>New Classification Scheme</name>
        <overviewDoc>
          <overviewURL>http://moreinfo</overviewURL>
        </overviewDoc>
        <categoryBag>
          <keyedReference tModelKey="uuid:c1..." keyValue="categorization"/>
        </categoryBag>
      </tModel>
    </save_tModel>
  </Body>
</Envelope>
```

te

Service clients



# The UDDI API : Enquiry API

- Enquiries enable clients to find businesses, services or bindings meeting certain criteria
- Two usage patterns
  - Browse (search based on some criteria)
  - Drill-down (to get specific features using keys)

Browse	Drill-down
find_business	get_businessDetail
find_relatedBusiness	get_businessDetailExt
find_service	get_serviceDetail
find_binding	get_bindingDetail
find_tModel	get_tModelDetail

# The UDDI API : Enquiry API

## Example: Looking for a software publisher in the state of California

```
<?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 keyValue="51121" tModelKey="uuid:co... />
        <keyedReference keyValue="US-CA" tModelKey="uuid:4e..." />
      </categoryBag>
    </find_business>
  </Body>
</Envelope>
```

**AND**

# The UDDI API : Enquiry API

## Example: Looking for a software publisher in the state of California

```
<?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>
        <findQualifiers>
          <findQualifier>orAllKeys</findQualifier>
        </findQualifiers>
        <keyedReference keyValue="51121" tModelKey="uuid:co... />
        <keyedReference keyValue="US-CA" tModelKey="uuid:4e..." />
      </categoryBag>
    </find_business>
  </Body>
</Envelope>
```

OR

# UDDI & RESTful web services

- UDDI can be used for registration and discovery of RESTful web services in that it incorporates multiple protocols such as HTTP, SMTP, FTP, etc.
- DNS is a valid discovery mechanism available for RESTful web services

# Public registries in use today

- ebXML Registry
- Seekda

## Alternatives to UDDI : ebXML Registry

- ebXML : Electronic Business using eXtensible Markup Language
  - architecture similar to SOAP/WSDL/UDDI
  - adoption of SOAP in the transport specification
  - adding security, guaranteed messaging, and compliance with business process interaction specifications

# Alternatives to UDDI : seekda

- Search engine for public Web Services
- It is not UDDI based

# Sources

- ❖ T. Bellwood (2002). Understanding UDDI Tracking the evolving specification, available on <http://www.ibm.com/developerworks/webservices/library/ws-featuddi>
- ❖ P. Brittenham , F. Cubera, D. Ehnebuske, S. Graham (2001). Understanding WSDL in a UDDI registry, Part 1: How to publish and find WSDL service descriptions, available on <http://www.ibm.com/developerworks/webservices/library/ws-wsdl>
- ❖ F. Cubera, M. Duftler, R. Khalaf, W. Nagy, N. Mukhi, S. Weerawarana (2002). Unraveling the Web Services Web: Introduction to SOAP, WSDL and UDDI.
- ❖ K. Januszewski, The Importance of Metadata: Reication, Categorization, and UDDI, available on Microsoft MSDN <http://msdn.microsoft.com/en-us/library/ms953942.aspx>
- ❖ Oasis, UDDI Spec Technical Committee Draft (2004), available on [http://uddi.org/pubs/uddi\\_v3.htm](http://uddi.org/pubs/uddi_v3.htm)
- ❖ M. P. Papazoglou, Web Services: Principles and Technology, chapter 6, pages 174-210.