

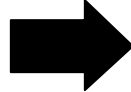
SESSION	<b>UDDI Overview</b>
NUMBER	<b>E418011</b>
SPEAKER	<b>David Ehnebuske</b>
TITLE	<b>Senior Technical Staff Member</b>
AFFILIATION	<b>IBM Corporation</b>

# What is UDDI?

- An architecture for web services
  - Standards-based specifications for web service description and discovery
- A set of internet-based implementations
  - Interoperating to share registrations
- Partnership among industry leaders
  - Initiated by Ariba, IBM, and Microsoft
  - Now 15 working group members
  - More than 160 advisors

# What Problems Does It Solve?

**Broader  
B2B**



A mid-sized manufacturer needs to create 400 online relationships with customers, each with their own set of standard and protocols.

**Smarter  
Search**



A flower shop in Australia wants to be "plugged in" to every marketplace in the world, but doesn't know how

**Easier  
Aggregation**



A B2B marketplace cannot get catalog data for relevant suppliers in its industry, along with connections to shippers, insurers, etc.

*Describe  
Services*

*Discover  
Services*

*Integrate  
Them  
Together*

# Vision and Process

## 1. Start with existing and emerging

- XML/SOAP --> XP
- Industry-specific schemas
- Shared vision of open protocols

## 2. Augment and implement via a Web Service

- Common web services "stack"
- Shared implementation to avoid confusing customers
- Public specs, open service, inclusive process

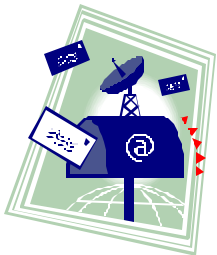
## 3. Transition to a Standards Body

- Manage design process for 3 revs
- License control and IP to a 3<sup>rd</sup> party

# UDDI v1



Manufacturers



Flower Shops



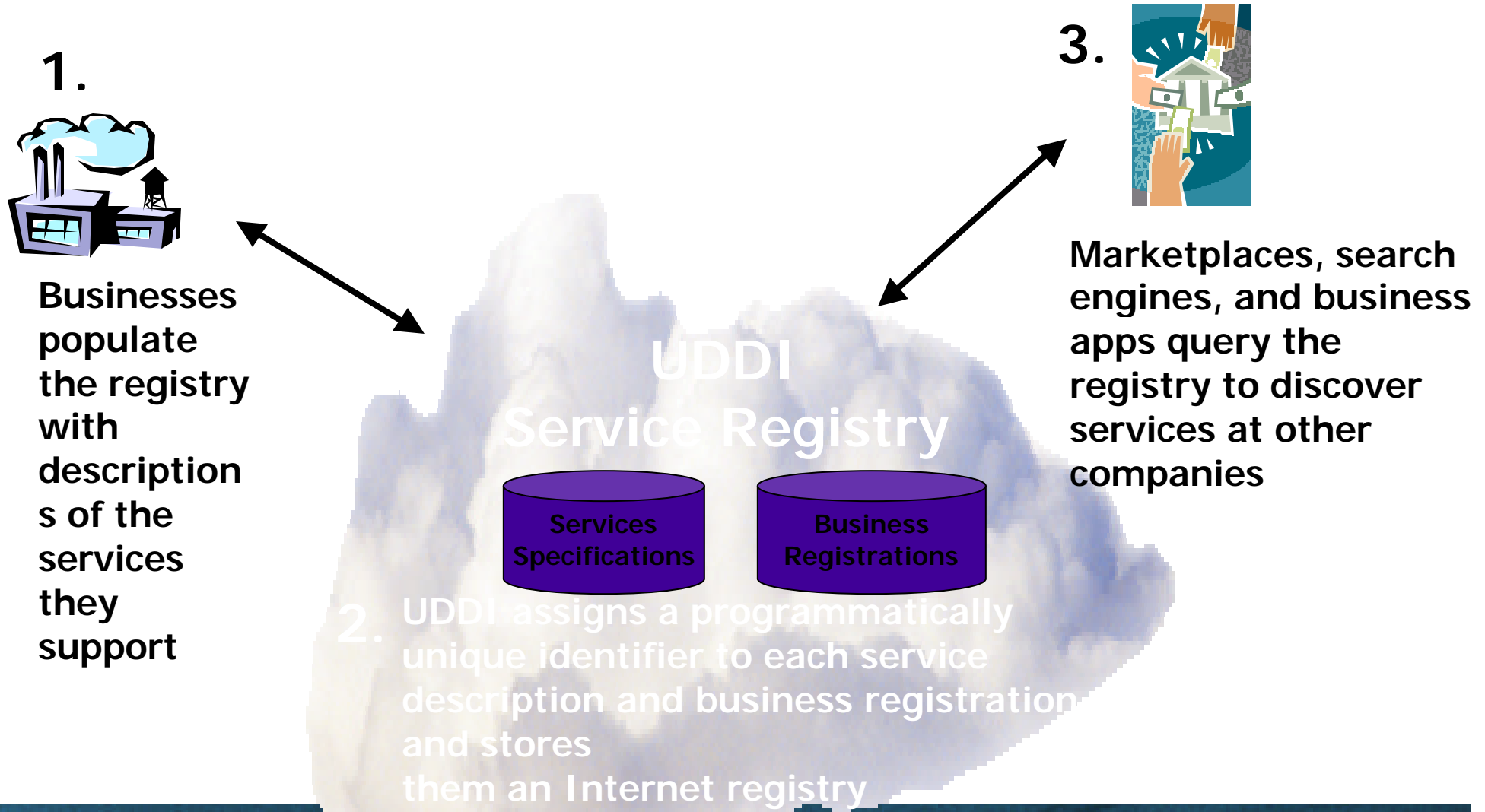
Marketplaces

**websphere**2001

## UDDI Service Registry

- ◆ Programmatic descriptions of web services
- ◆ Programmatic descriptions of businesses and their web services
- ◆ Programming model, schema, and platform agnostic
- ◆ Uses XML, HTTP, and SOAP
- ◆ Free on the Internet

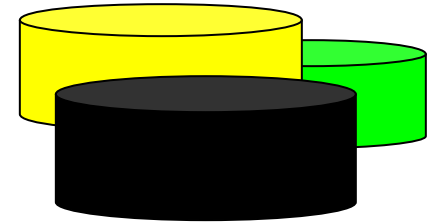
# How UDDI v1 Works



# The Specification

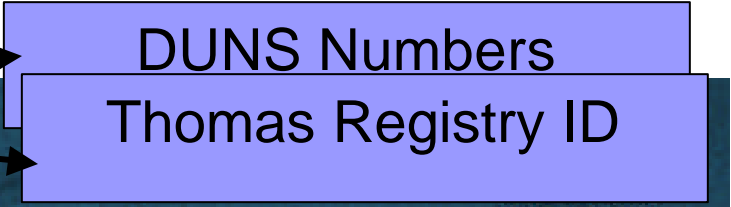
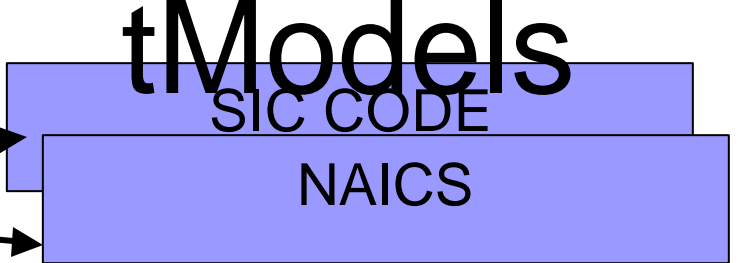
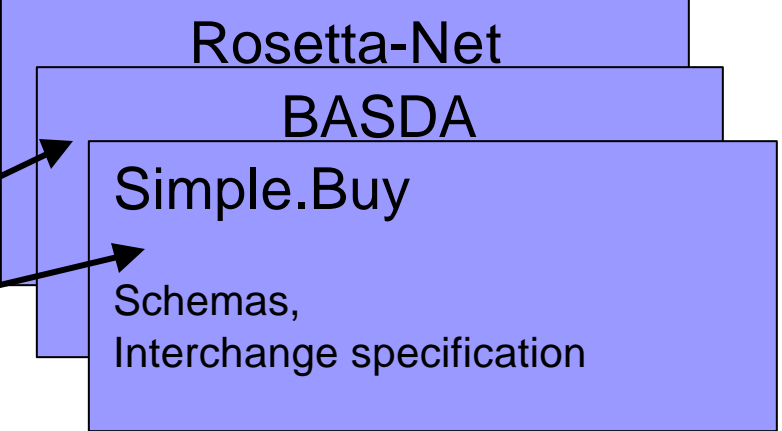
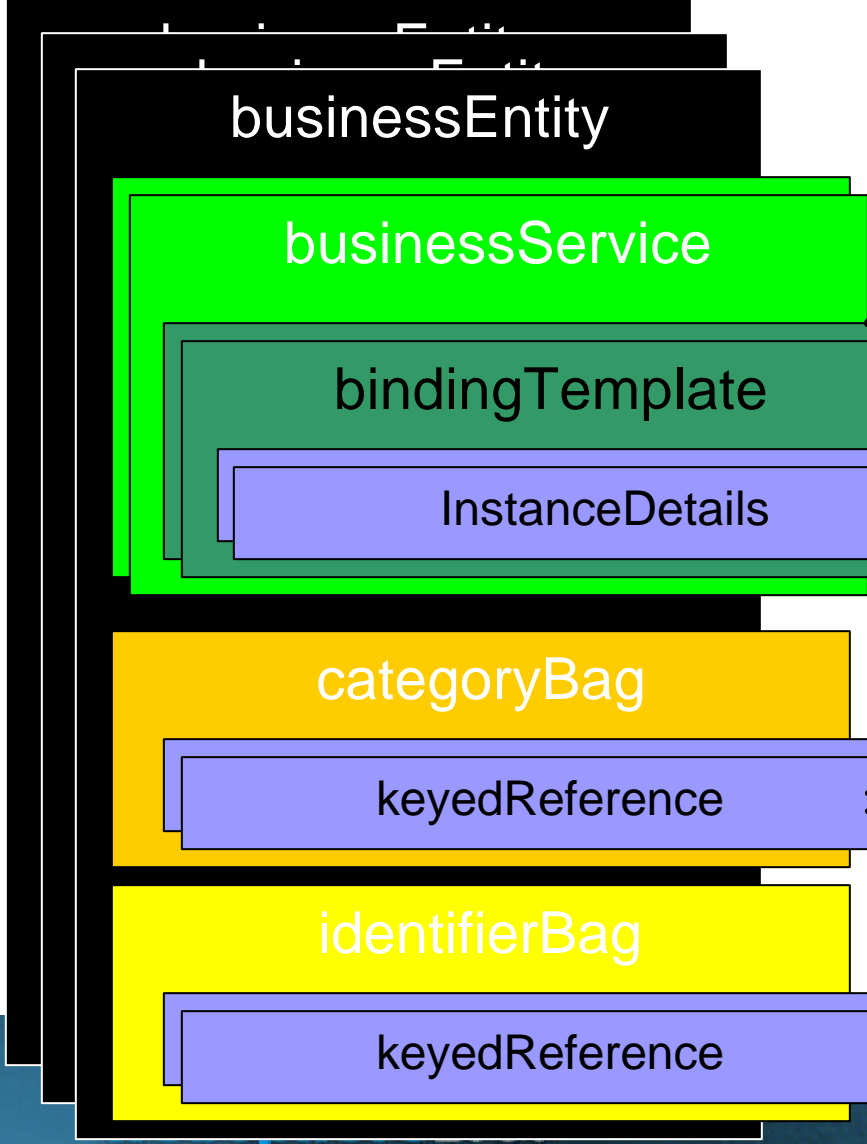
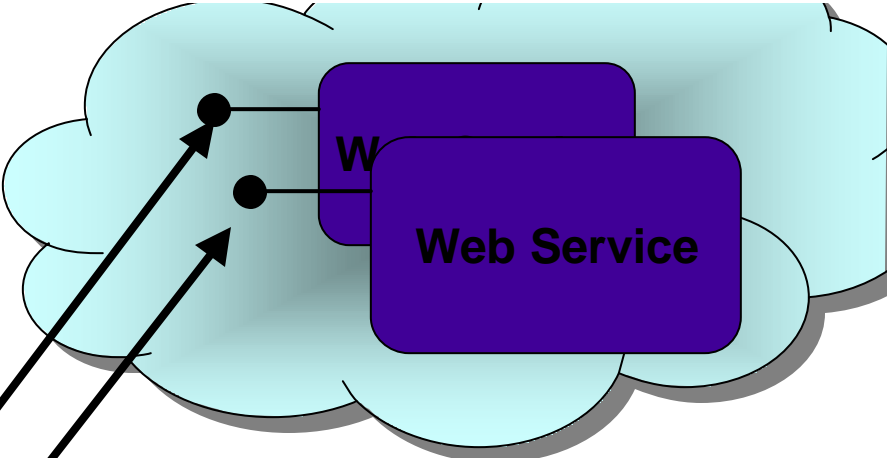
- Programmer's API defining
  - Publication of information for discovery
    - I am RosettaNet, here's the definition for my service types
    - "I'm company X, I have these services..."
  - General Inquiry
    - Get me the service type definitions that are published
    - "Get me the stuff that company X saved"
  - Web Service call specifics
    - Call request/response structure semantics
    - SOAP, XML, and error handling details

## Four building blocks



- ◆ tModel is a keyed reference to an API or namespace
- ◆ Business Entity – description of a business and their contacts
- ◆ Business Service – a named reference to a service
- ◆ Binding Template – a description of a service including its location and tModels it supports

# UDDI Relationships



# UDDI API

## ■ Inquiry API

- Find things
  - find\_business
  - find\_service
  - find\_binding
  - find\_tModel
- Get Details about things
  - get\_businessDetail
  - get\_serviceDetail
  - get\_bindingDetail
  - get\_tModelDetail

## ■ Publishing API

- 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

SESSION	<b>Using WSDL to Define Services in UDDI</b>
NUMBER	<b>E418011</b>
SPEAKER	<b>Francisco Curbera</b>
TITLE	<b>Research Staff Member</b>
AFFILIATION	<b>IBM Research</b>

# What is WSDL

- An XML language for describing Web Services
  - Provides a functional description of a Web service
    - IDL description
    - Access protocol
    - Deployment details
- A platform independent description
- An extensible language

# Structure of WSDL Service Descriptions

- WSDL separates four elements of a service description:
    - Type/message definitions
    - Abstract interface descriptions
    - Reusable protocol bindings
  - Endpoints of deployed services
- } Reusable description

# WSDL Description Structure

```
<message name="Msg1" /> ...
```

```
<portType name="PType1"> ...
```

```
<binding name="Bnd1" type="PType1"> ...
```

} Reusable

Deployment

```
<service name="svc1">
```

```
  <port binding="Bnd1" >
```

```
    <soap:address location="http://..."/>
```

```
  </port>
```

```
</service>
```

# Reusable WSDL Specs as tModels

- UDDI tModels provide base for technical service descriptions:
  - Registered technical specification
  - UDDI does not define how to author tModels.
  - A stack of tModels defines how to access a service.
- WSDL provides an XML schema to define reusable technical specifications.
  - Registered as “wsdlSpec” type tModels
  - Referenced by compliant registered services

# UDDI v1 and WSDL: A Developer's View

2.



Service developers create and register new services compliant with specs.

3.



Business users find service specs and develop code capable of accessing the service.

1.



Business groups create and register WSDL specs.

4.



Business apps discover and bind to compliant services.

UDDI  
Service Registry