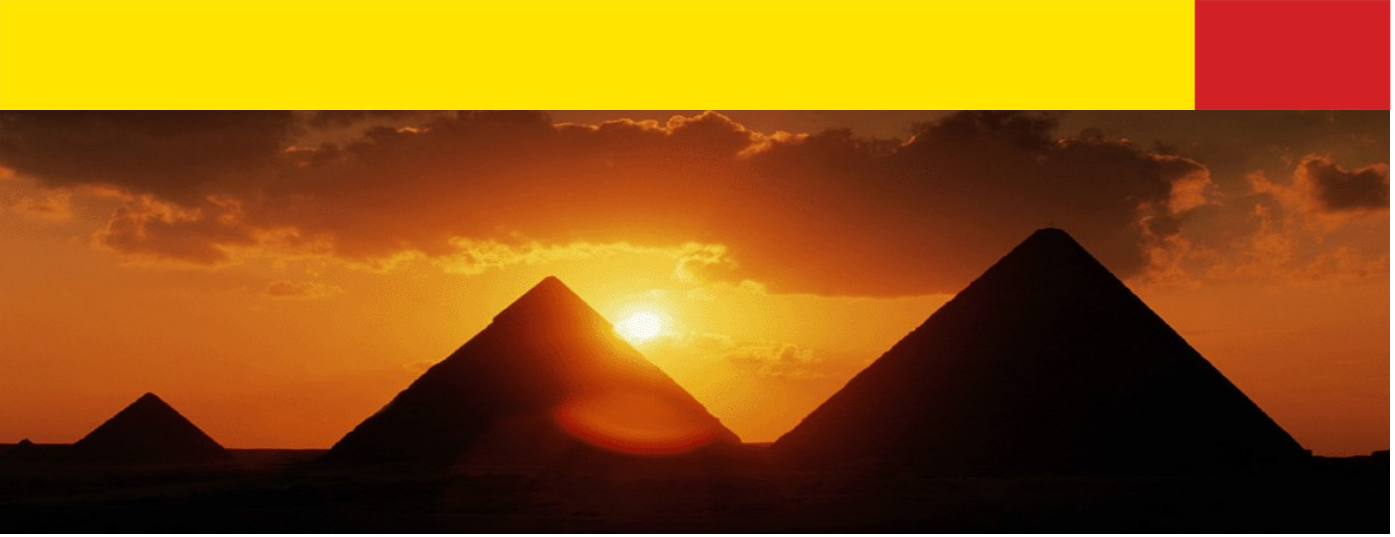
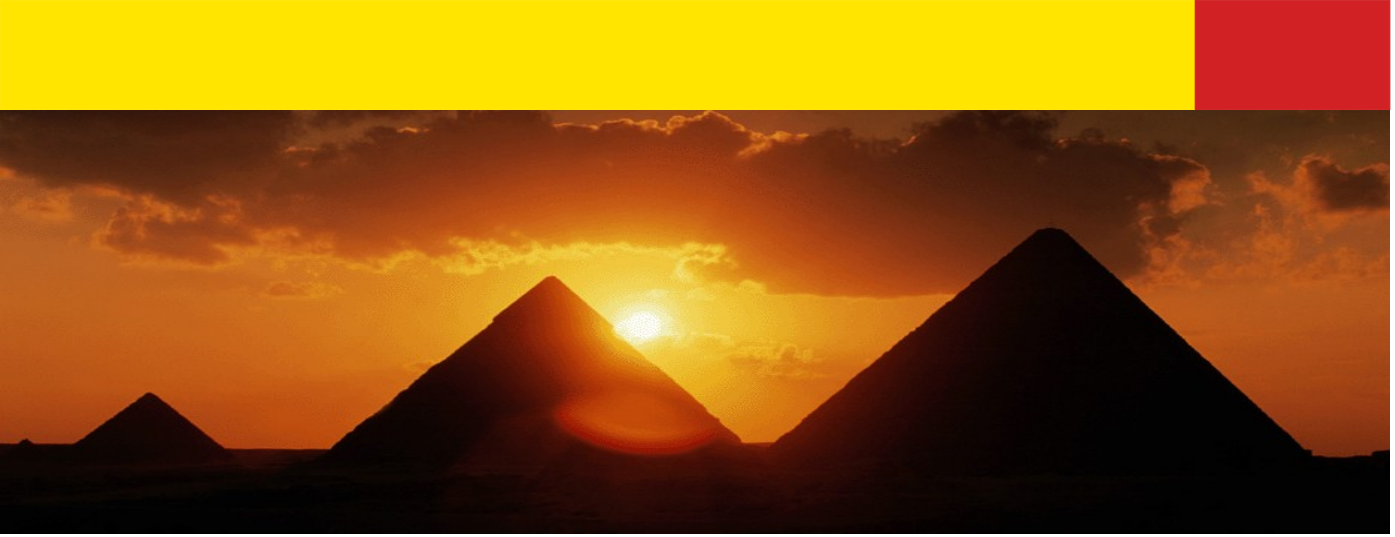


UDDI

(Universal Description, Discovery and Integration)

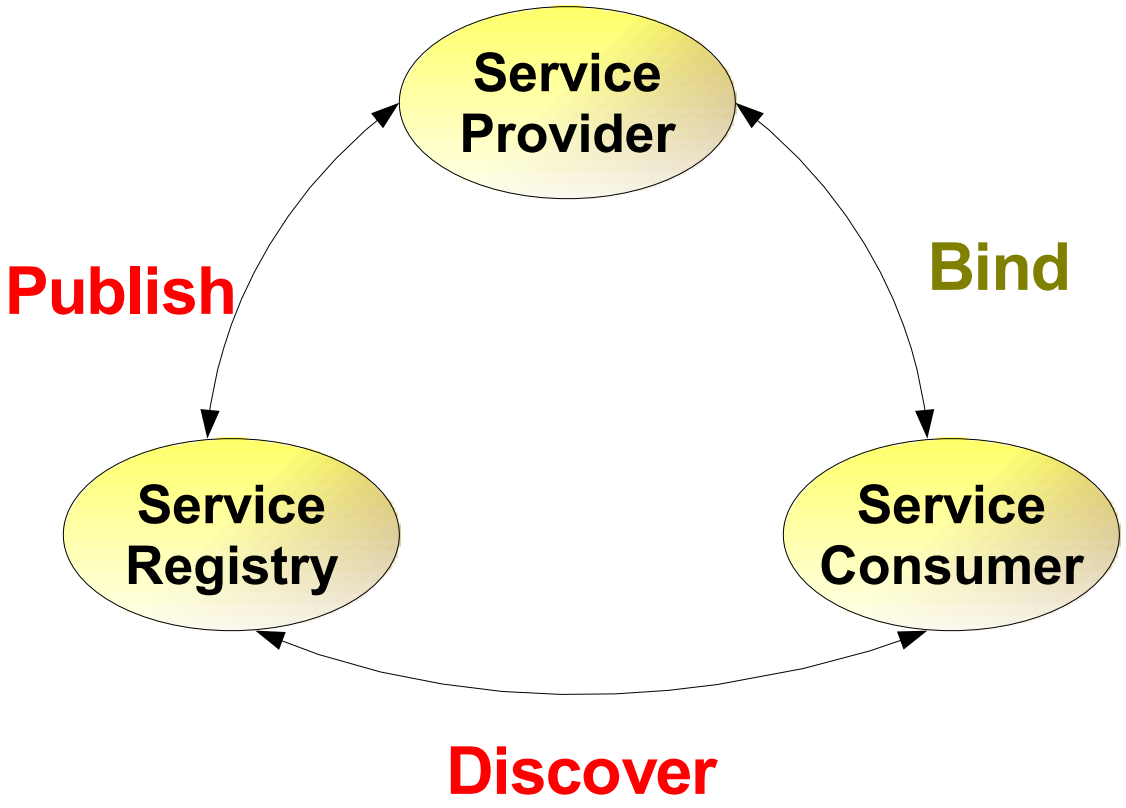
Agenda

- What is and Why UDDI?
- UDDI Data Types and their structural relationship
- UDDI Categorization
- UDDI Programming model
- UDDI over SOAP
- Authoring Steps of WSDL, UDDI
- Discovery, Binding and Invocation of a Service
- Issues of UDDI
- Java API for UDDI
- UDDI 3.0 & Future of UDDI



What is and Why UDDI?

Service Architecture



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

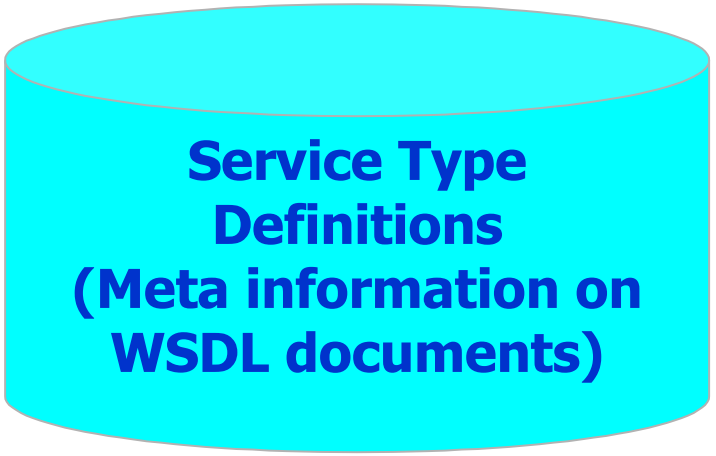
What is UDDI?

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

Registry Data

Created by businesses

Created by standard organizations, industry consortium



Business Registration Data

- "White pages"
  - address, contact, and known identifiers
- "Yellow pages"
  - industrial categorizations
    - Industry: NAICS (Industry codes - US Govt.)
    - Product/Services: UN/SPSC (ECMA)
    - Location: Geographical taxonomy
- "Green pages"
  - technical information about services

What uses UDDI?

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

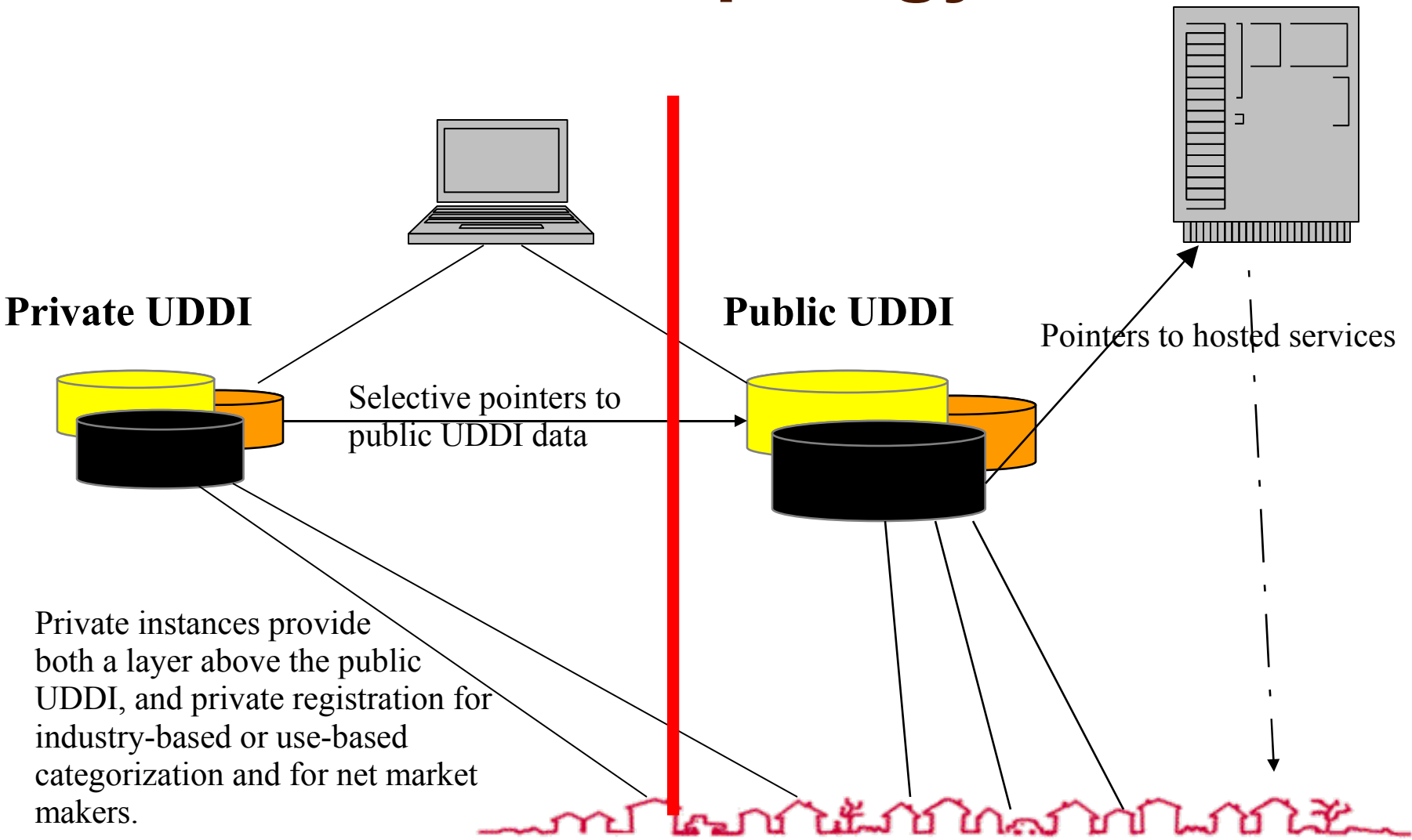
Why UDDI or something like UDDI?

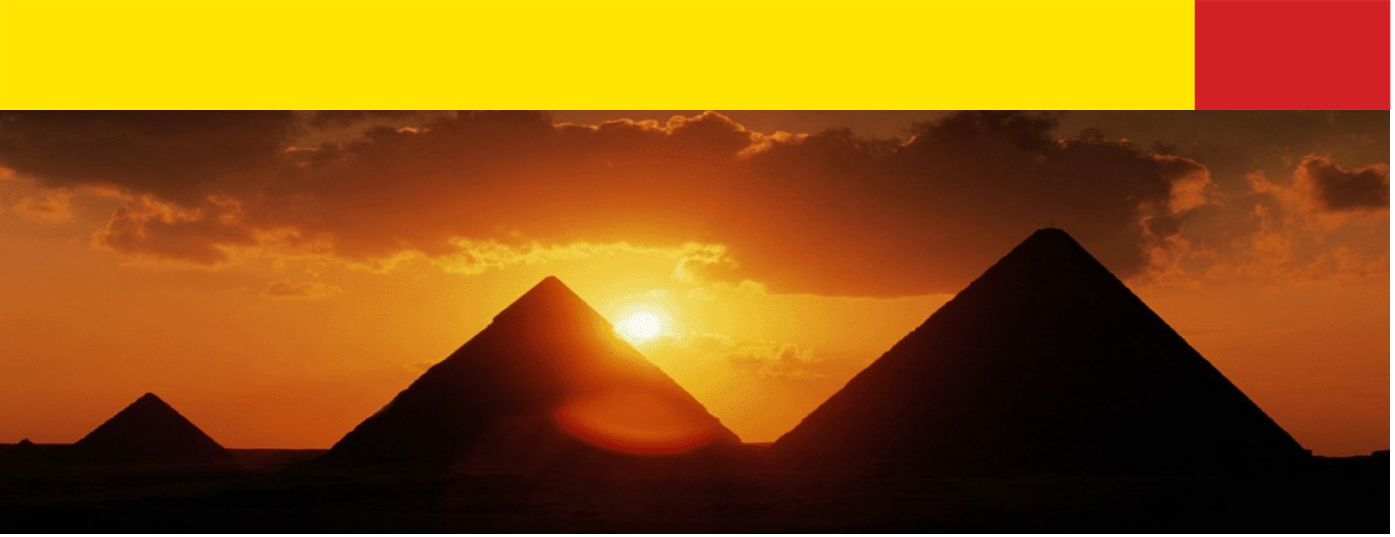
- Platform independent service publication and discovery
- Enables dynamic service discovery

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

Possible UDDI Topology

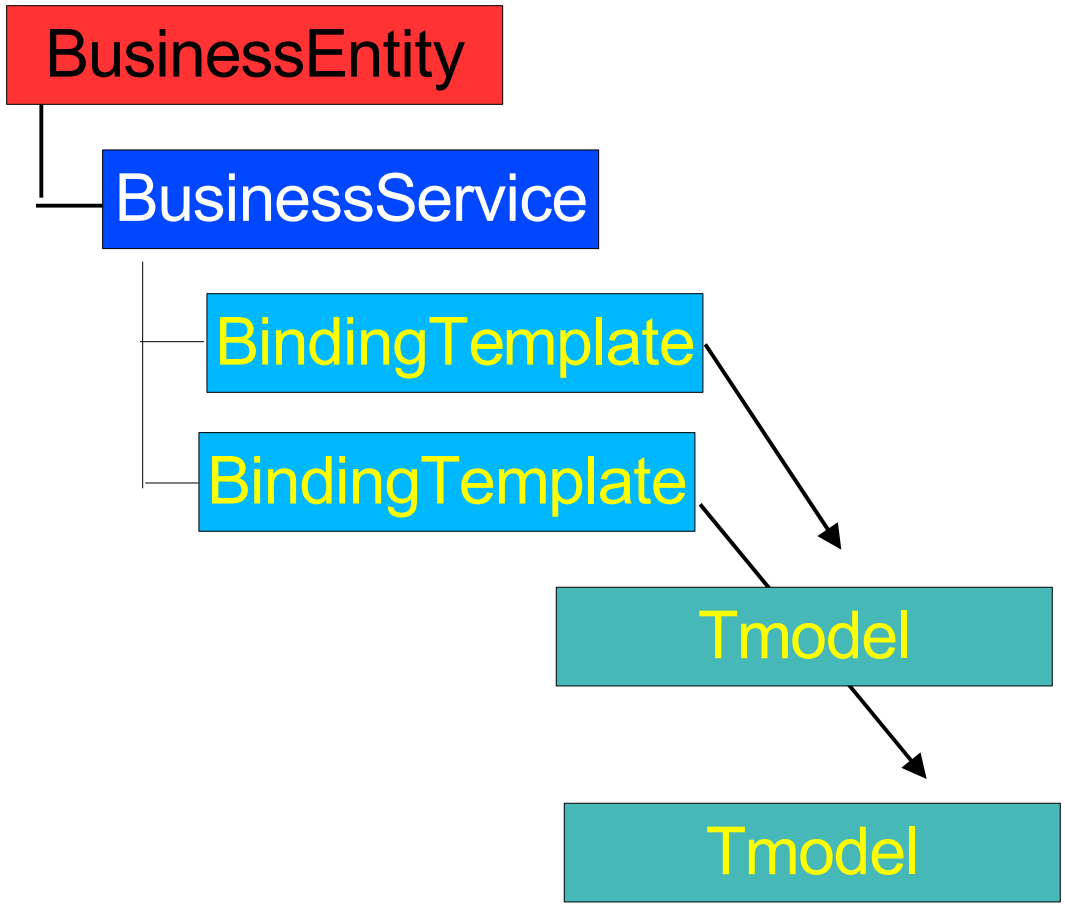




UDDI Data Types & Their Hierarchical Relationship

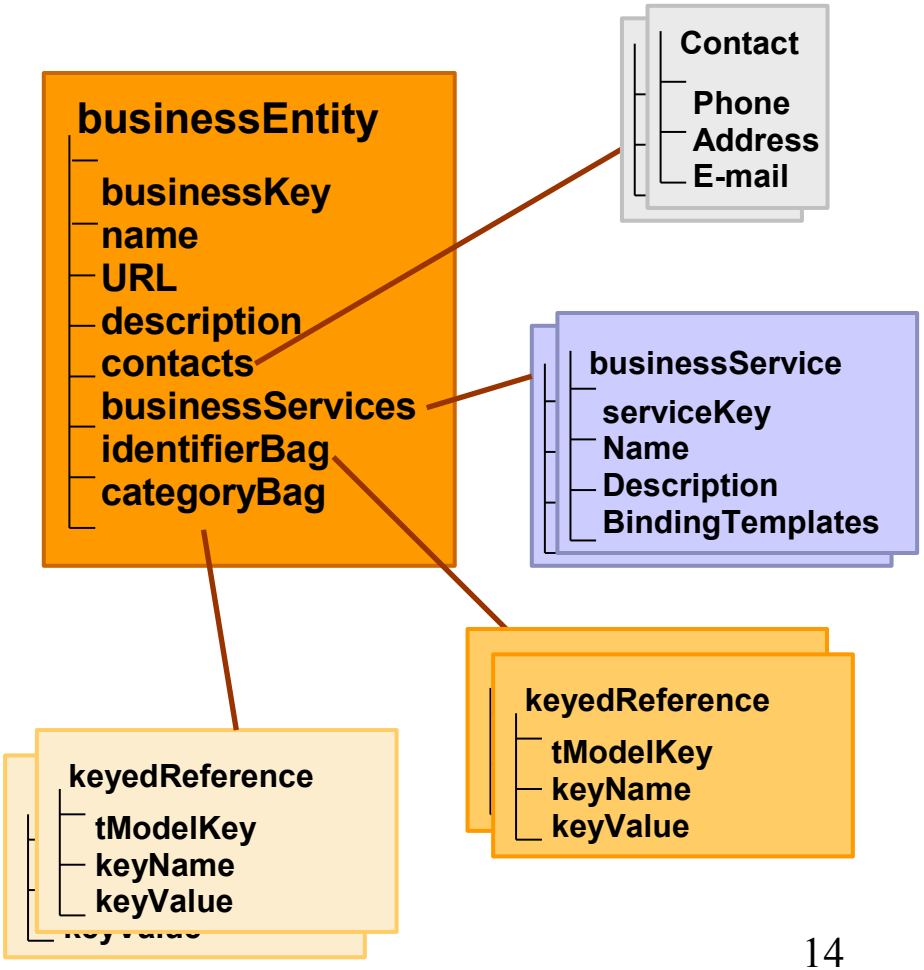
UDDI Data Types

- Business Entity
  - White Pages information
- Business Services
  - Yellow Pages information
- Binding Templates
  - Green Pages information
  - Contains references to tModels
- tModels
  - Service Type Definitions
  - Contains references to WSDL documents



Business Entity Data Type

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



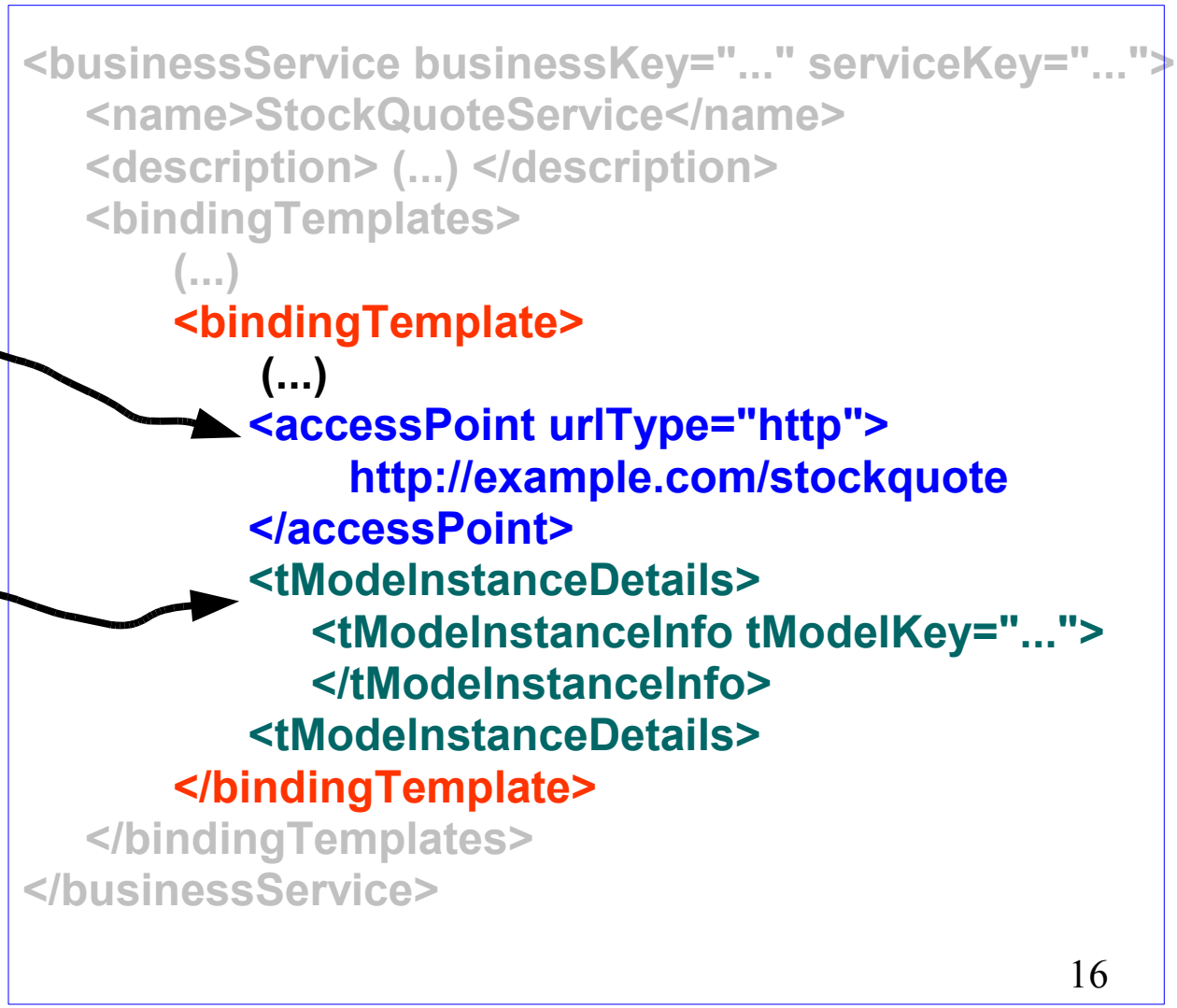
BusinessService

- Represents the business services provided by the *businessEntity*
- Unique key used to represent a service
- Name of the service
- Contains *BindingTemplate* structures

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

BindingTemplate

- **Specifies Network endpoint address**
- **Contains a reference to a tModel**



tModel

- Service type definition
- Is 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 supports a particular service
- Contains categoryBag

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>
```

categoryBag Element

- Allows businessEntity, businessService and tModel structures to be categorized according to any of several available taxonomy based classification scheme

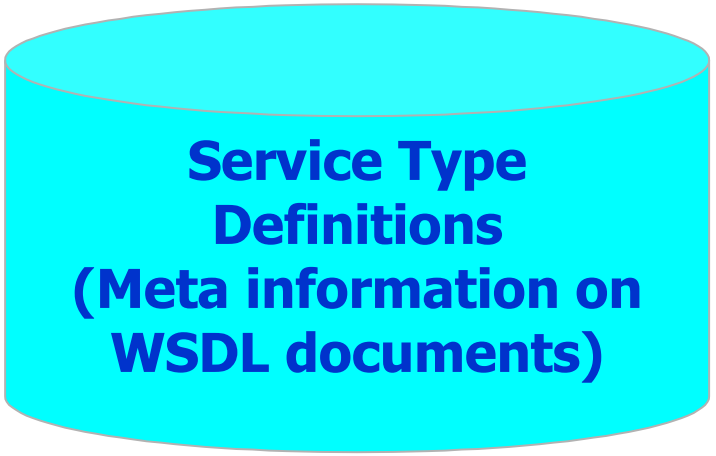
identifierBag Element

- Allows businessEntity or tModel structures to include information about common forms of identification such as D-U-N-S numbers, tax identifiers, etc.
- Can be used to signify the identity of the businessEntity, or can be used to signify the identity of the publishing party
- Enhances the search behaviors exposed via the find_xx

Registry Data

Created by businesses

Created by standard organizations, industry consortium



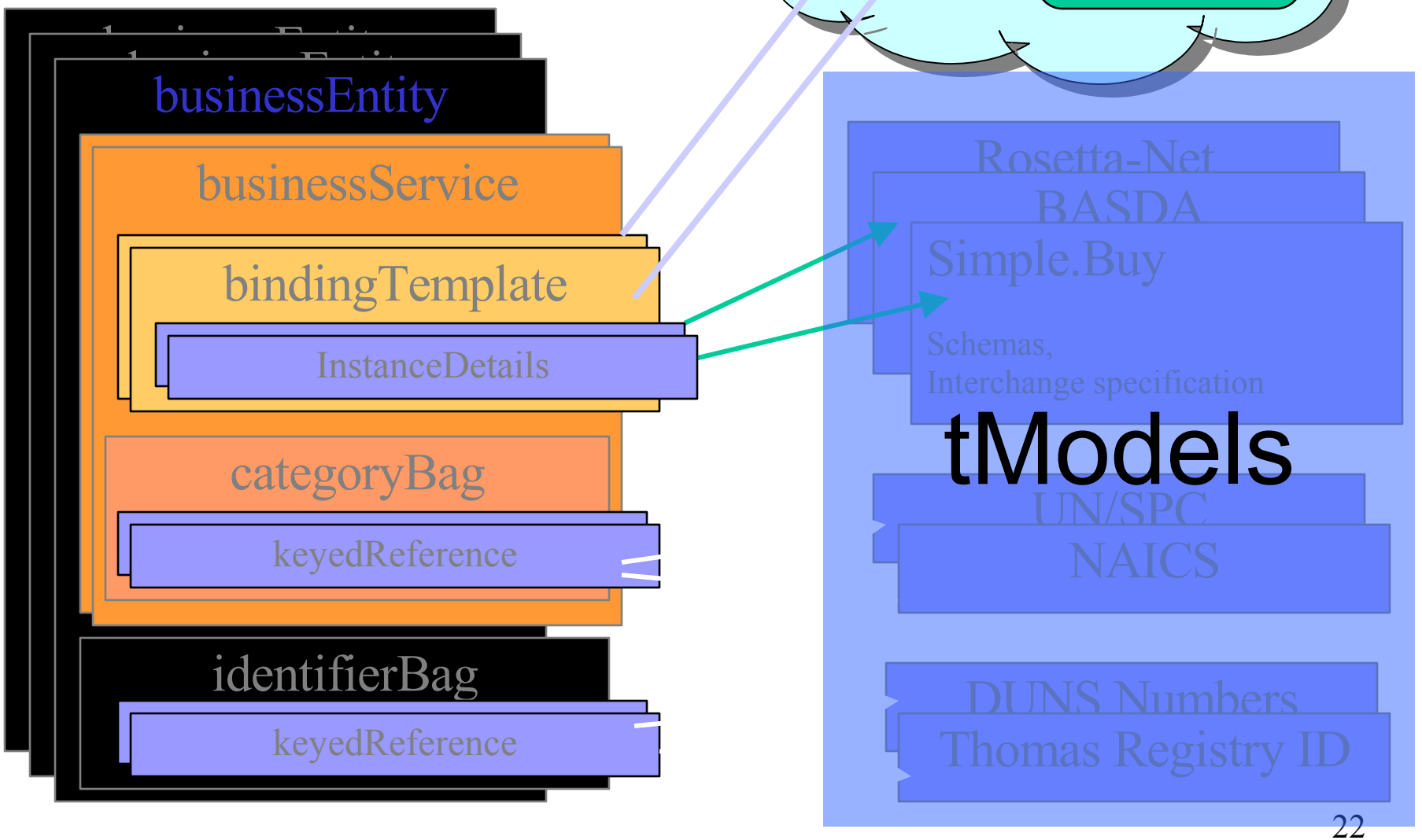
businessEntity's

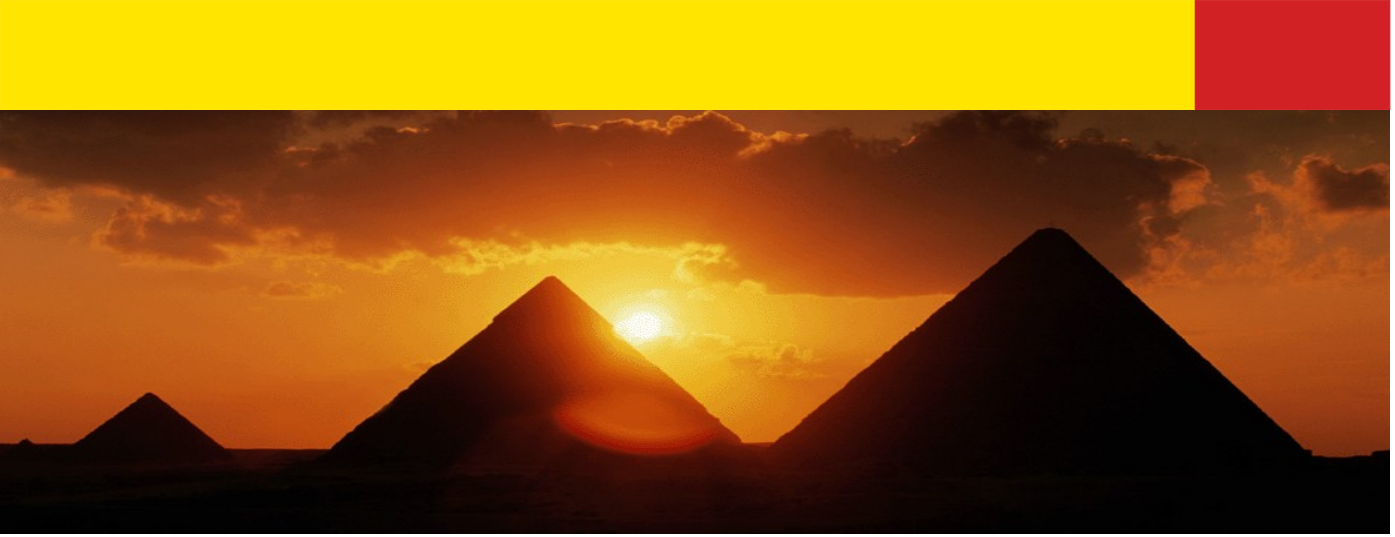
businessService's

bindingTemplate's

tModel's

Information Model

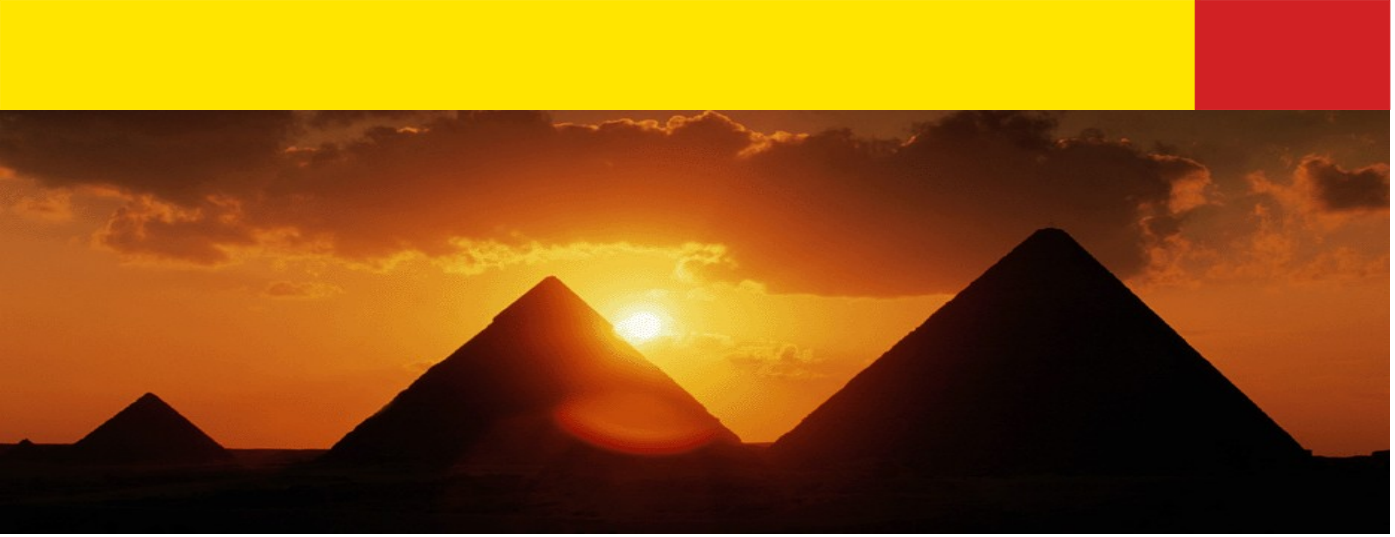




UDDI Categorization

Categorization

- NAICS (Industry code)
- UNSPAC
- D-U-N-S
- ISO 3166
- SIC



UDDI Programming Interface

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.

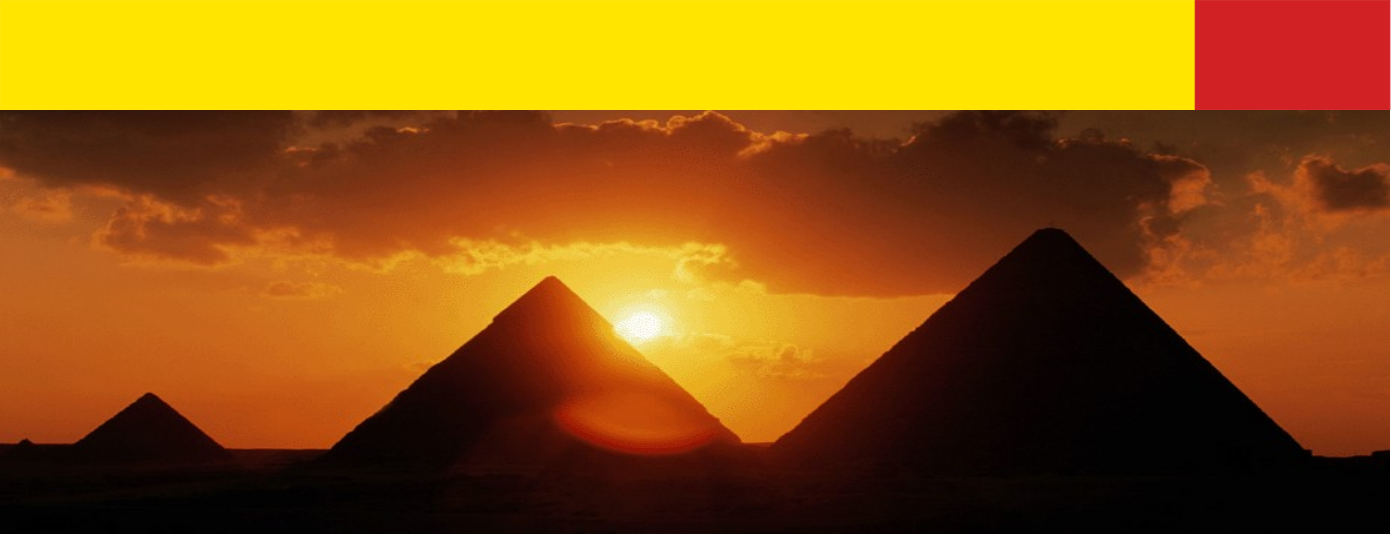
Security:

- request an authentication token
- inform registry that the authToken is no longer valid.

Programmer's API: Service Discovery

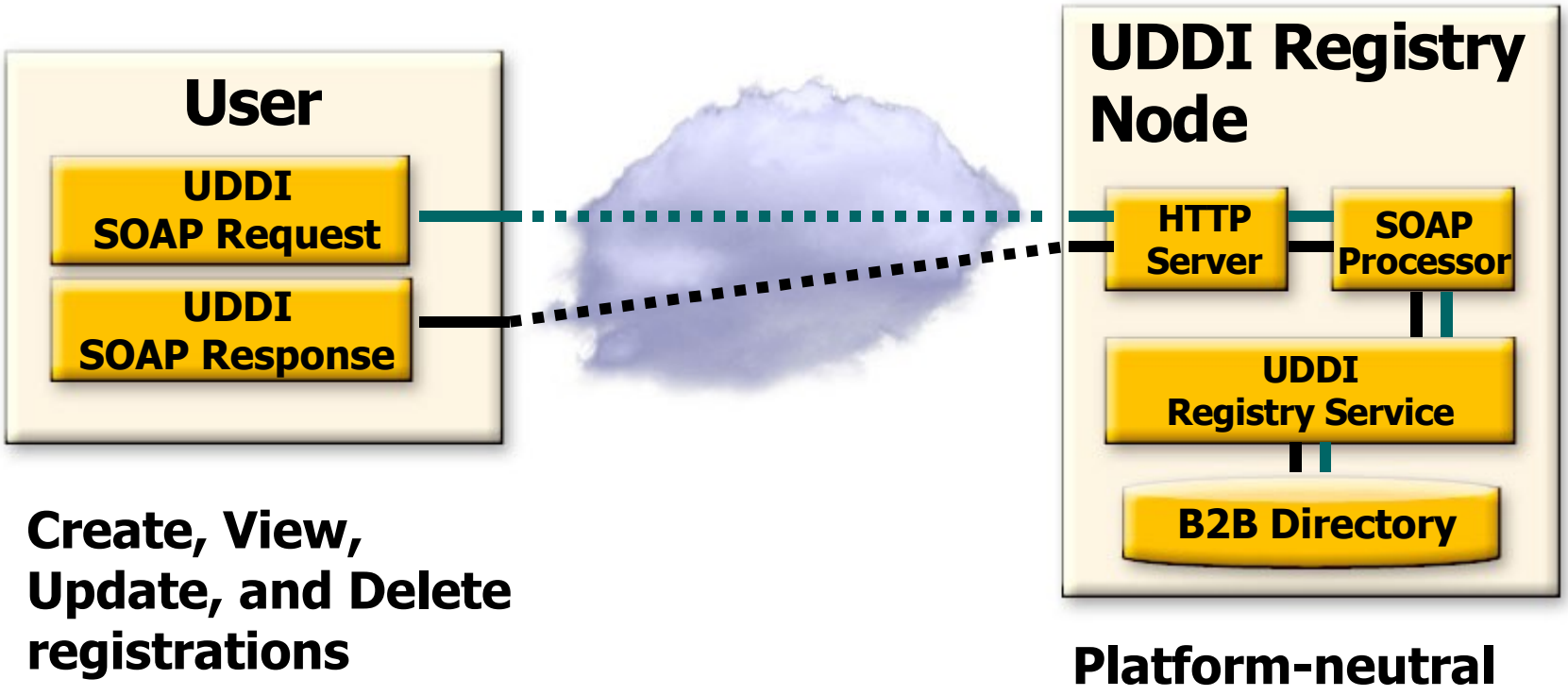
- Inquiry interface
  - Find things
    - **Find_business**
    - **Find_service**
    - **find_binding**
    - **find_tModel**
  - Get details
    - **Get_businessDetail**
    - **get_serviceDetail**
    - **get_bindingDetail**
    - **Get_tModelDetail**
- Taxonomy interface
    - **validate_categorization**

- Browse
  - 4 messages to **find** each of the 4 structures
- Drill-down
  - The get call can be used to get information regarding a specific instance of any of the 4 data types, given the key



UDDI over SOAP

UDDI Runs "Over" SOAP



Create, View, Update, and Delete registrations

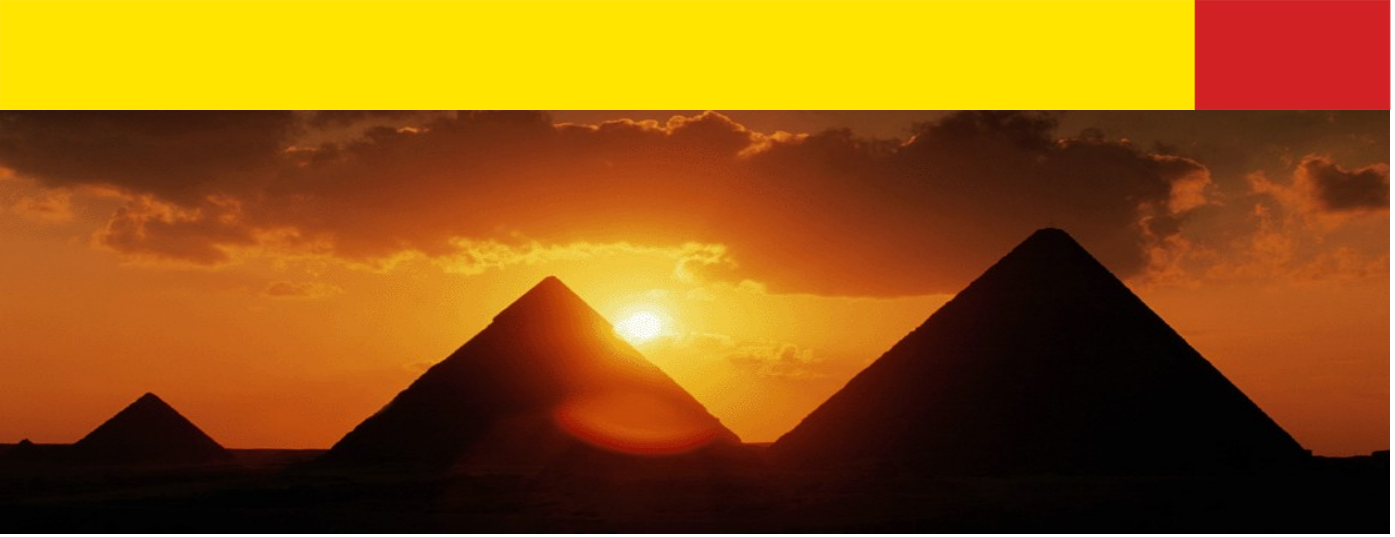
Platform-neutral

SOAP Message Example for get_serviceDetail request

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

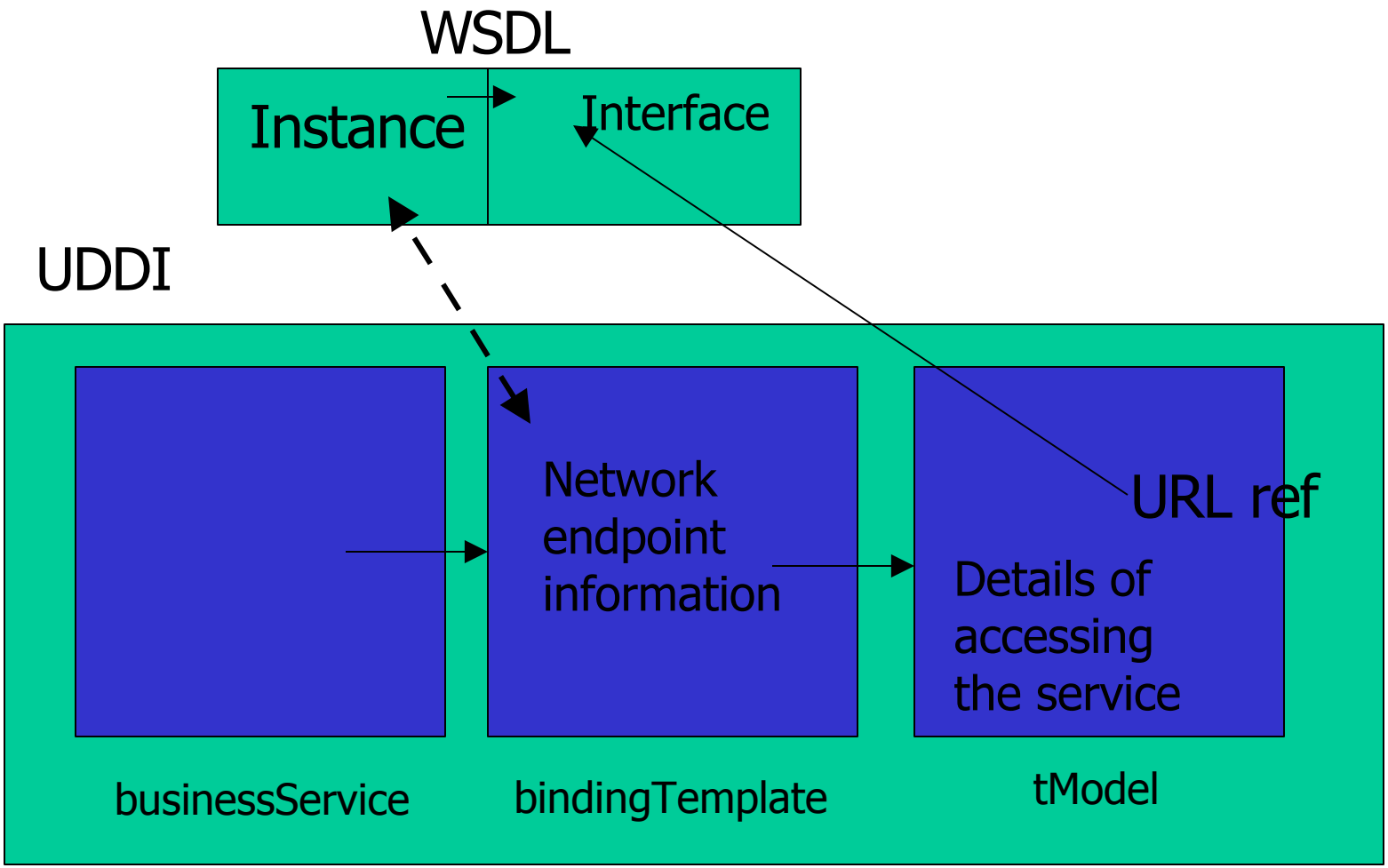
SOAP Message Example for get_serviceDetail response

```
<Envelope>
  <Body>
    <serviceDetail generic="1.0" operator="XMethods">
      <businessService serviceKey="6FD77EF6-E7D6-6FF6-1E41-EBC80107D7B5"
                       businessKey="D1387DB1-CA06-24F8-46C4-86B5D895CA26">
        <name>Currency Exchange Rate</name>
        <description>Endpoint for service</description>
        <description>IMPLEMENTATION: glue</description>
        <description>CONTACT EMAIL: support@xmethods.net</description>
        <bindingTemplates>
          <bindingTemplate bindingKey="0036DEBC-2F1B-EB84-09E2-3A4332C3E8B4"
                           serviceKey="6FD77EF6-E7D6-6FF6-1E41-EBC80107D7B5">
            <description>SOAP binding</description>
            <accessPoint URLType="http">http://services.xmethods.net:80/soap</accessPoint>
            <tModelInstanceDetails>
              <tModelInstanceInfo tModelKey="uuid:D784C184-99B2-DA25-ED45-
    3665D11A12E5"/>
            </tModelInstanceDetails>
          </bindingTemplate>
        </bindingTemplates>
      </businessService>
    </serviceDetail>
  </Body>
</Envelope>
```



UDDI and WSDL How They Are Related

UDDI, WSDL Relationships



WSDL Service Implementation

```
<definitions name="StockQuoteService"
    targetNamespace="http://...">

  <import namespace="http://..."
      location="http://...">

  <service name="StockQuoteService">

    <port name="SingleSymbolService"
      binding="iface:SingleSymbolBinding">
    ...
  </service>
</definitions>
```

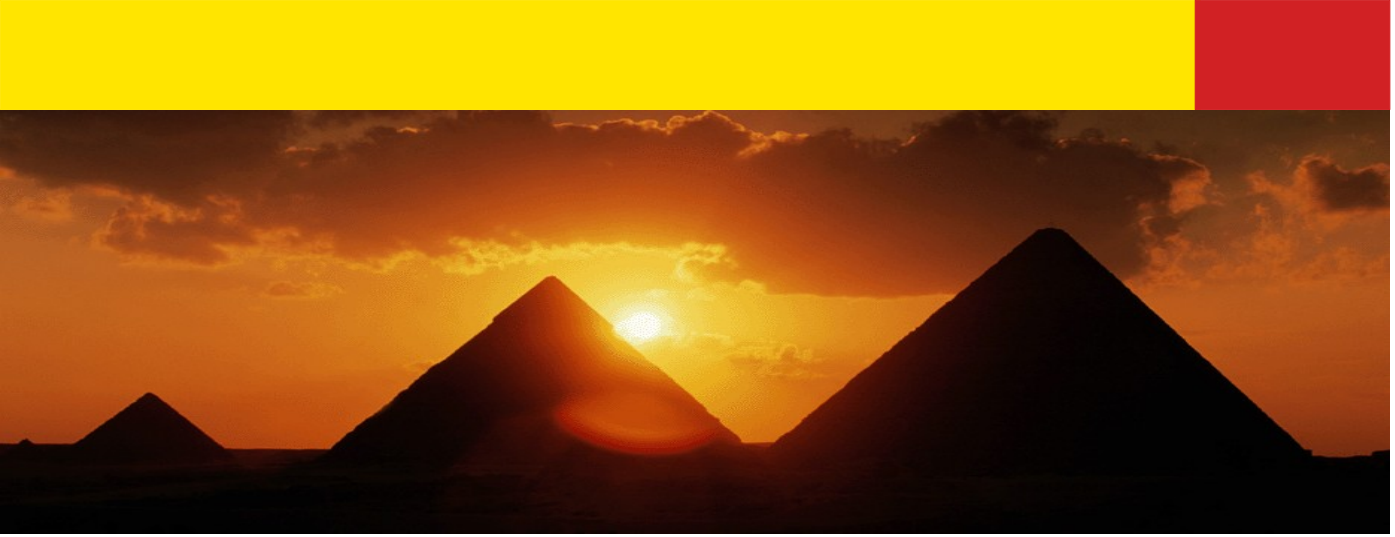
UDDI Registry

```
<businessEntity businessKey="...">
  <name>Stock Quote Service, Inc.</name>
  ...

  <businessService serviceKey="..."
    <name>StockQuoteService</name>
    ...
    <bindingTemplates>
      <bindingTemplate bindingKey="...">
        ...
      <tModelInstanceInfo tModelKey="...">
        ...
        <overviewDoc>
          <overviewURL>
            http://.../SQS.wsdl
          </overviewURL>
        ...
    </bindingTemplates>
  </businessService>
</businessEntity>
```

WSDL Service Interface

```
<definitions
    name="StockQuoteService-interface"
    targetNamespace="http://..." >

  <message name="SingleQuoteRequest">
  </message>
    ...
  <portType name="SingleSymbolService">
  </portType>
    ...
  <binding name="SingleSymbolBinding"
           type="tns:SingleSymbolService">
    ...
  </binding>
</definitions>
```

```
<tModel tModelKey="...">
  <name>http://...</name>
  <overviewDoc>
    <overviewURL>
      http://.../SQS-interface.wsdl
    </overviewURL>
  </overviewDoc>
  <categoryBag>
    <keyedReference tModelKey="..."
        keyName="uddi-org:types"
        keyValuee="wsdlSpec"/>
  </categoryBag>
</tModel>
```



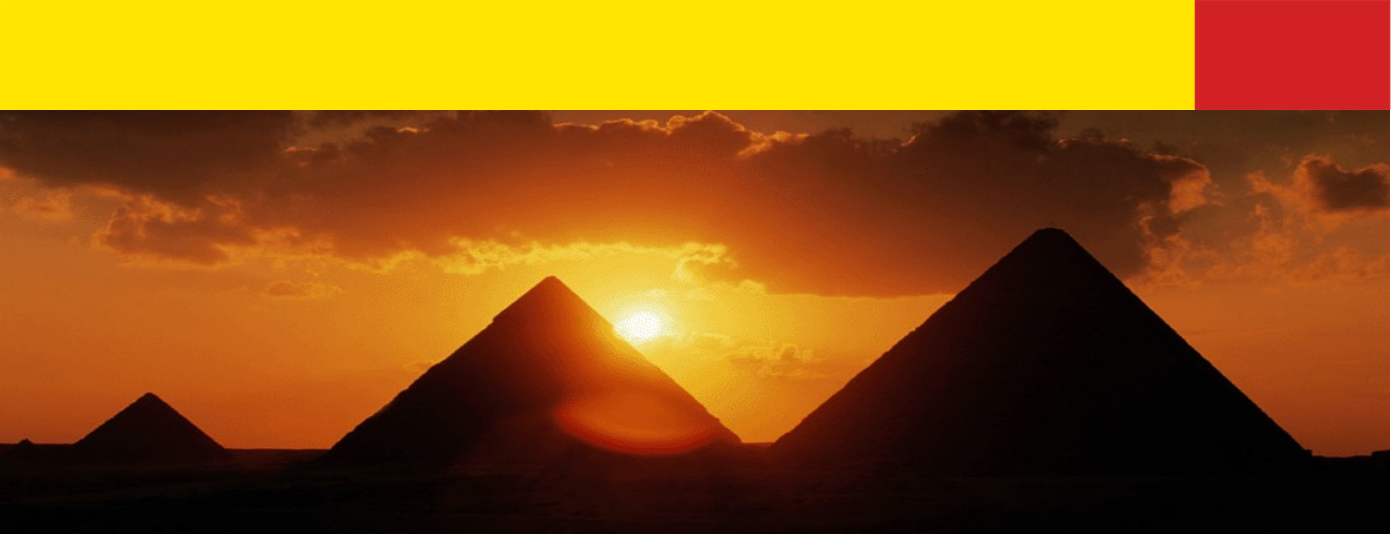
Authoring Steps of WSDL & UDDI

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

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



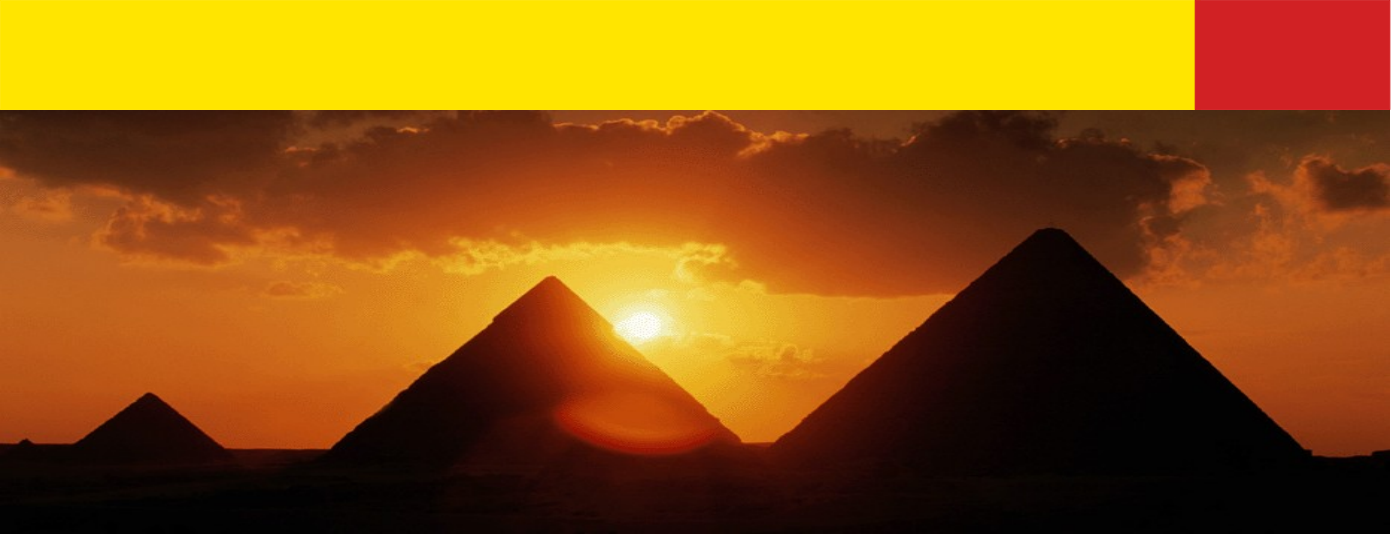
Discovery, Binding and Invocation of a Service

Discovery of a Service

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

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



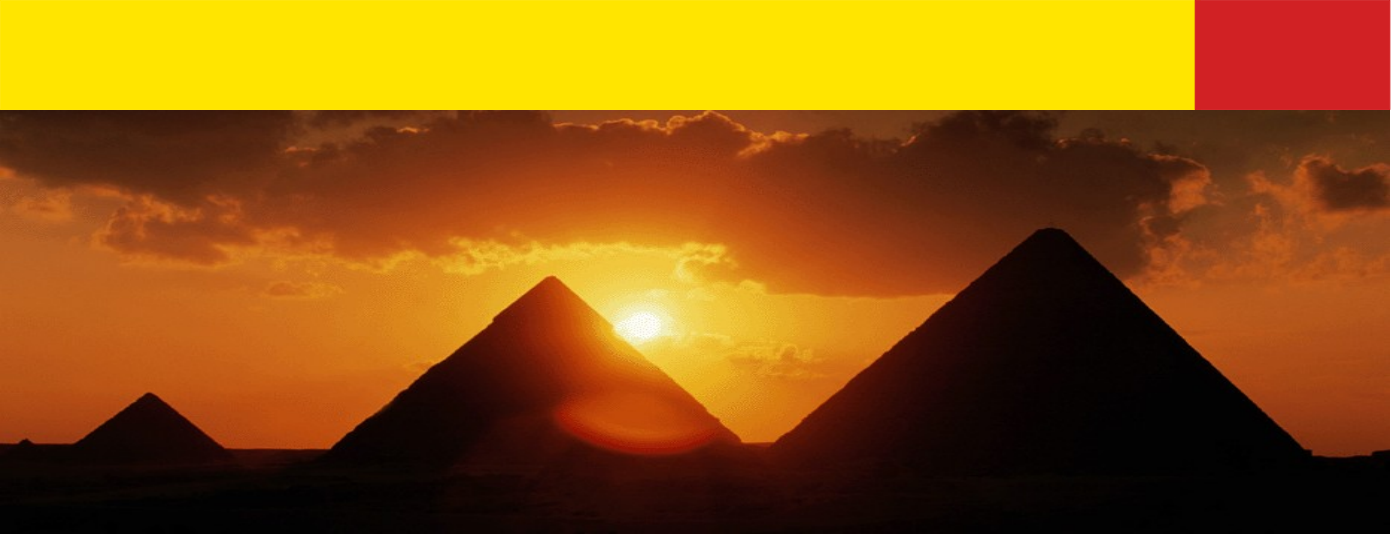
Issues and Current Status of UDDI

Issues of UDDI

- How do you know if the data you get is valid, legitimate, and up to date?
- How do you measure quality of data?
- How do you make sure only the qualified entities register their service information (authentication)?
- How do you provide access control to the data in the registry?
- How do you synchronize the data in multi-registry environment?

Current Status of UDDI

- No production-quality deployment and usage in a public domain
  - Only experimental UDDI servers are present
- Limited usage within an Intranet



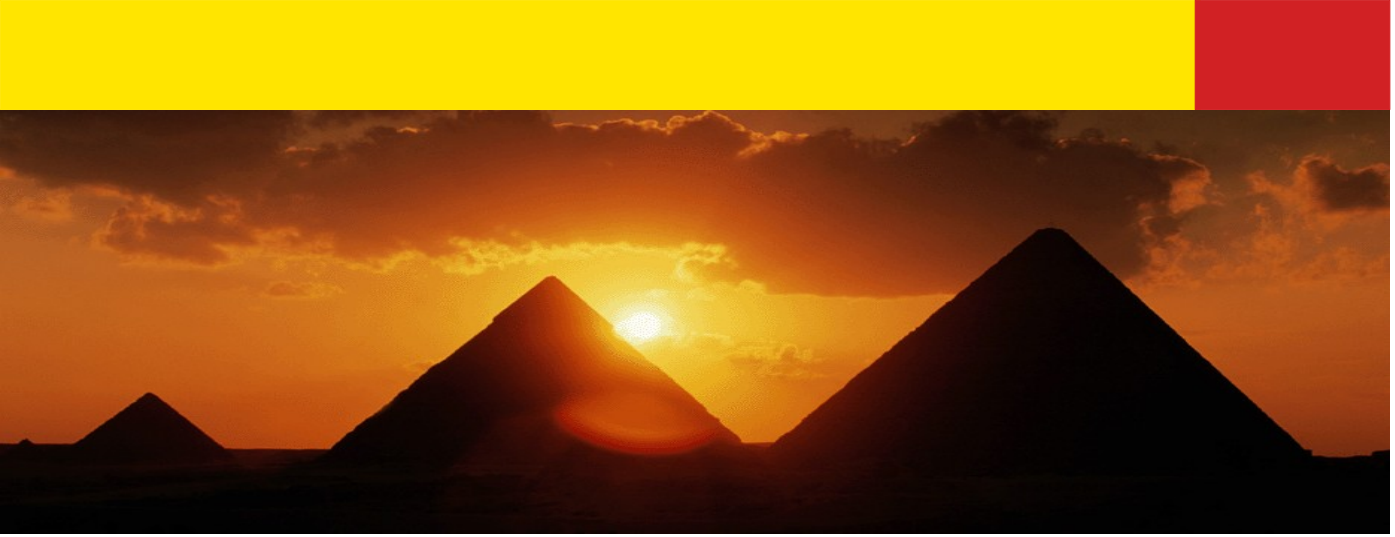
Java API for UDDI

JSR 93: Java™ API for XML Registries (JAXR)

- API to expose heterogeneous and distributed XML registries:
  - ebXML, UDDI, …
  - Publish, subscribe, query, associate…
    - Web service metadata
    - Schemas
    - Business processes
    - Documents

JUDDI

- Open source implementation of UDDI V2
- Runs over any Servlet 2.3 compliant container



UDDI 3.0 and Future of UDDI

UDDI 3.0 from OASIS

- 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

Resources

Resources

- OASIS UDDI TC
  - http://www.oasis-open.org/committees/uddi-spec/
- UDDI
  - www.uddi.org
- UDDI server from Java Web Services Development Pack
  - http://java.sun.com/webservices/downloads/webservicespack.html
- Example UDDI Browser from Java Web Services Development Pack
  - http://java.sun.com/webservices/downloads/webservicespack.html

Resources

- Using UDDI WSDL in a UDDI Registry (UDDI Working Draft Best Practice Document)
  - http://www.uddi.org/pubs/wsdlbestpractices-V1.05-Open-20010625.pdf