Module Three: Service Discovery UDDI and Service Composition BPEL

Task

- Each group should keep discussing on Blackboard and finalize what topic they want to work on by Wednesday

Hot topic study - possible topics

- IoT and Services
- Fog/Edge Computing and Services
- Cloud Computing
- Big Data Services
- Digital Health – services that support healthcare
- Services in Smart cities
- …..

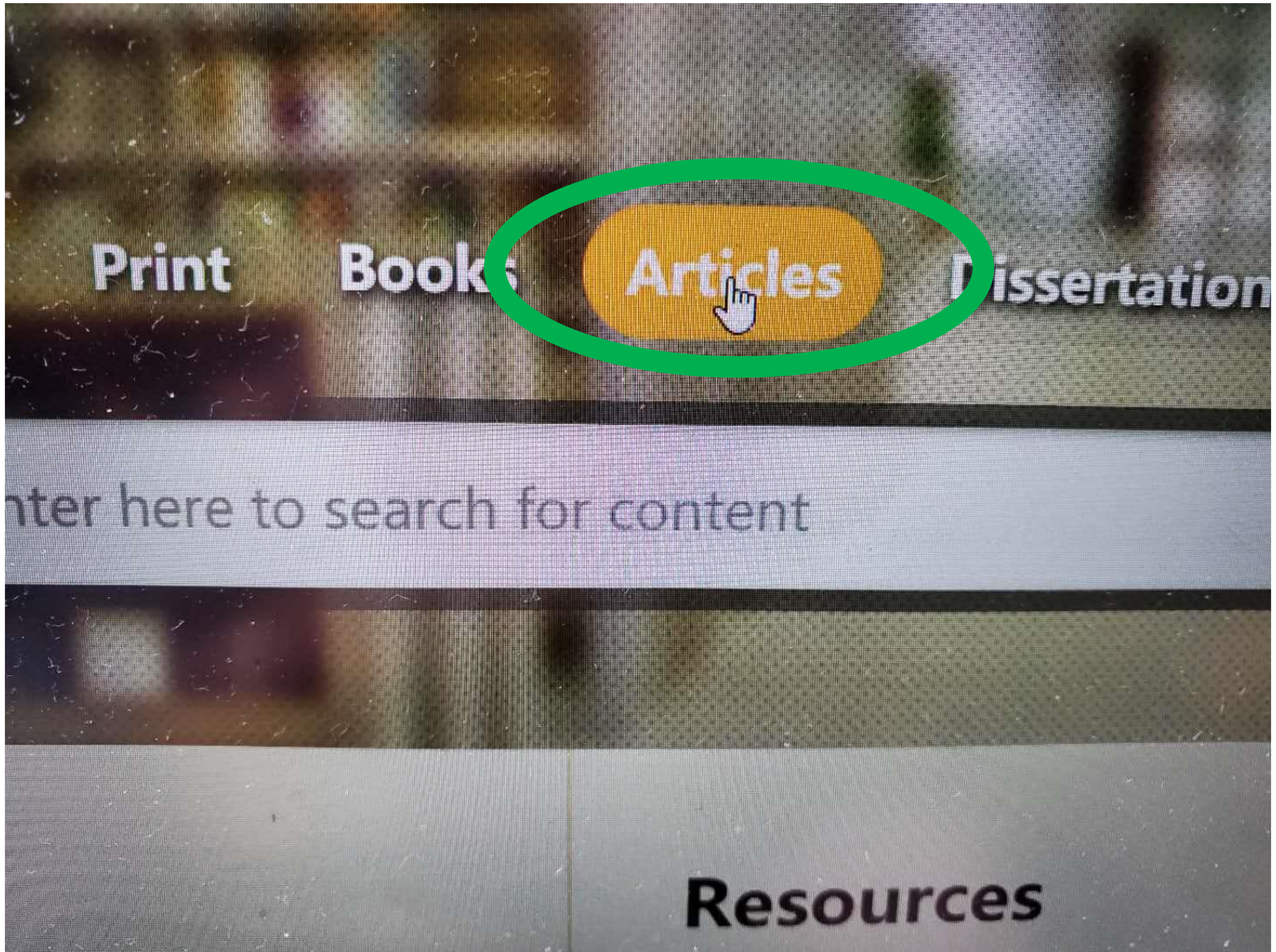
Transactions on Services Computing – free access

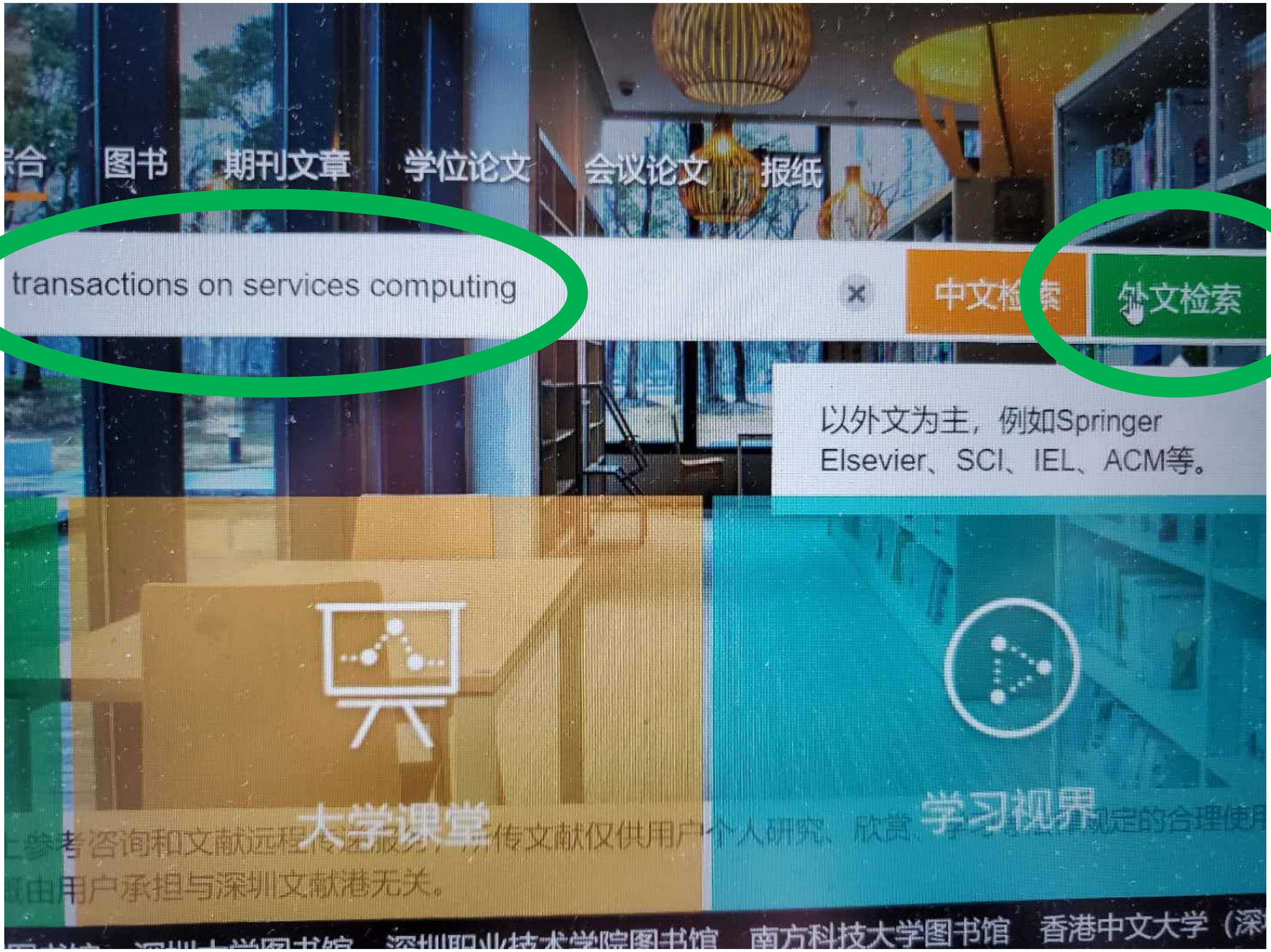
To access the articles for free, login through our library (a short guide in the next slides)

- http://ieeexplore-ieee-org-s.vpn.hitsz.edu.cn:8118/xpl/RecentIssue.jsp?punumber=4629386

Guest editorials offer an overview of some topics:

- http://www-computer-org-s.vpn.hitsz.edu.cn:8118/digital-library/journals/sc/tsc-editorials-and-guest-editorials


Print
Books
Articles
Dissertation
nter here to search for content
Resources



宗合 图书 期刊文章 学位论文 会议论文 报纸

transactions on services computing

中文检索 外文检索

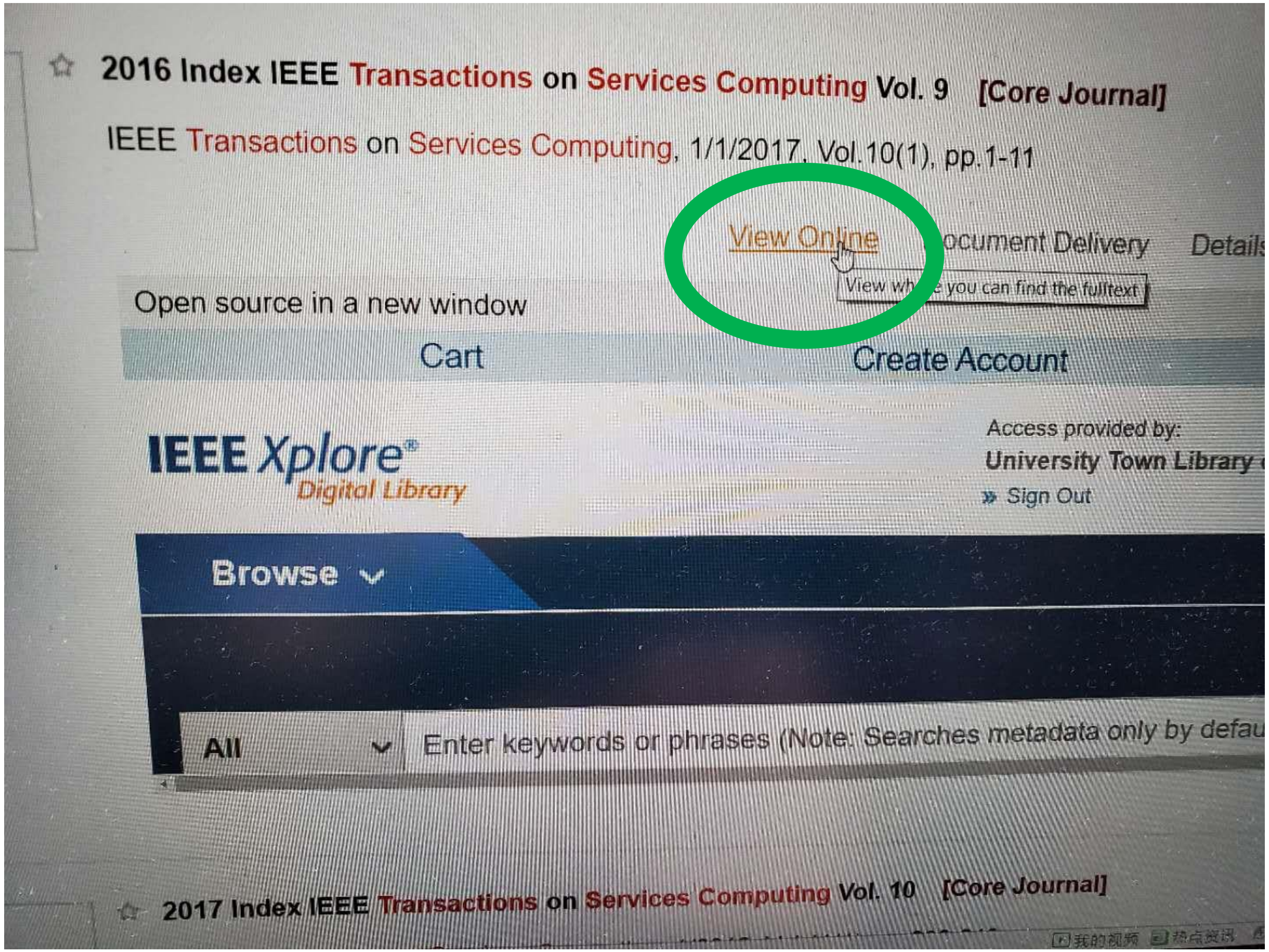
以外文为主，例如Springer Elsevier、SCI、IEL、ACM等。

大学课堂 学习视界

上参考咨询和文献远程传送服务，所传文献仅供用户个人研究、欣赏、 规定的合理使用

证由用户承担与深圳文献港无关。

图书馆 深圳大学图书馆 深圳职业技术学院图书馆 南方科技大学图书馆 香港中文大学（深



2016 Index IEEE Transactions on Services Computing Vol. 9 [Core Journal]

IEEE Transactions on Services Computing, 1/1/2017, Vol.10(1), pp.1-11

View Online ocument Delivery Details

View wh e you can find the fulltext

Open source in a new window

Cart Create Account

IEEE Xplore® Digital Library

Access provided by:
University Town Library

» Sign Out

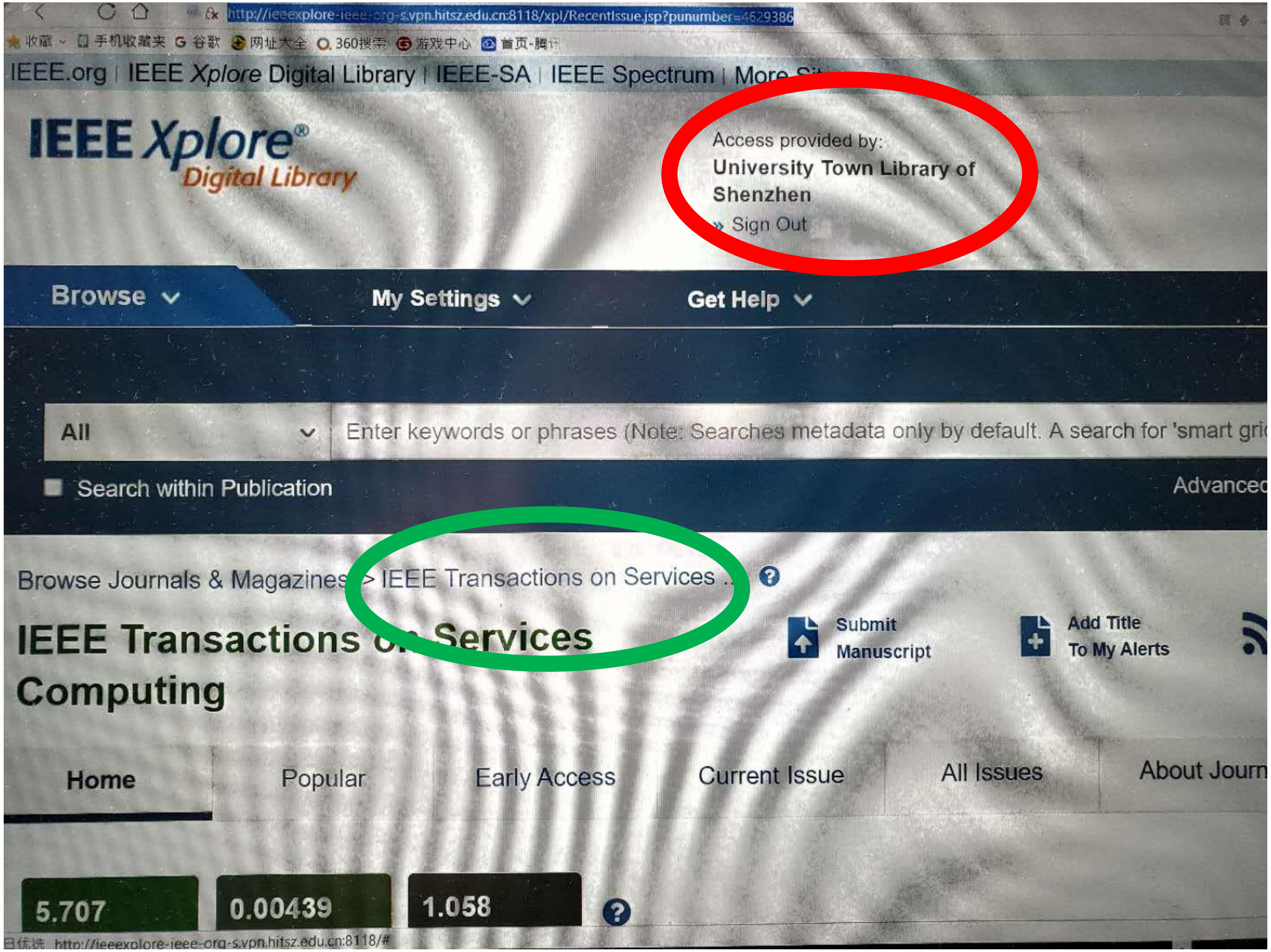
Browse

All

Enter keywords or phrases (Note: Searches metadata only by defau

2017 Index IEEE Transactions on Services Computing Vol. 10 [Core Journal]

我的视频 热点资讯



http://ieeexplore-ieee-org-s.vpn.hitsz.edu.cn:8118/xpl/RecentIssue.jsp?punumber=4629386

收藏 手机收藏夹 谷歌 网址大全 360搜索 游戏中心 首页-腾讯

IEEE.org | IEEE *Xplore* Digital Library | IEEE-SA | IEEE Spectrum | More Si

Access provided by:
University Town Library of Shenzhen
» Sign Out

Browse | My Settings | Get Help

All | Enter keywords or phrases (Note: Searches metadata only by default. A search for 'smart gri

Search within Publication | Advanced

Browse Journals & Magazines > IEEE Transactions on Services ...

IEEE Transactions on Services Computing

Submit Manuscript | Add Title To My Alerts

Home | Popular | Early Access | Current Issue | All Issues | About Journ

5.707 | 0.00439 | 1.058

http://ieeexplore-ieee-org-s.vpn.hitsz.edu.cn:8118/#

Transactions on Services Computing

To access the articles for free, login through our library (a short guide in the next slides)

- http://ieeexplore-ieee-org-s.vpn.hitsz.edu.cn:8118/xpl/RecentIssue.jsp?punumber=4629386

Guest editorials offer an overview of some topics:

- http://www-computer-org-s.vpn.hitsz.edu.cn:8118/digital-library/journals/sc/tsc-editorials-and-guest-editorials

Module Three: Service Discovery UDDI and Service Composition BPEL

XML-RPC

SOAP

WSDL

UDDI

BPEL

JEE

SOA

Textbooks

- Web Services Essentials
  - Chapter 7
- Services Computing
  - Chapters 3.3-3.6

You can find an online version of this book **for free** through our library webpage

Module 3 Learning Outcomes

- Understand the main concepts of UDDI
- Understand main uses of UDDI,
- Understand the technical aspects of UDDI
- Understand basic concepts of BPEL
- Understand BPEL basic structure
- Be able to create business process

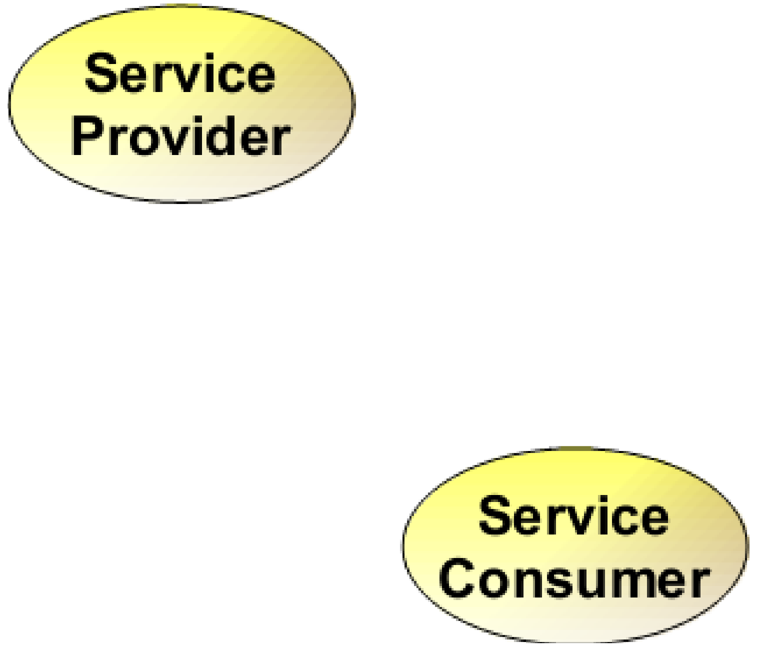
Module 3 Guiding Questions

- What is service discovery?
- What is UDDI?
- What is the relationship between XML, SOAP and UDDI?
- What are the technical aspects of UDDI?
- What are the main uses of UDDI?
- Can you explain the UDDI data model in details?
- How to search UDDI via web based interface?
- How to use the UDDI programmatic API?
- How to publish new companies and services to UDDI?

Module 3 Guiding Questions

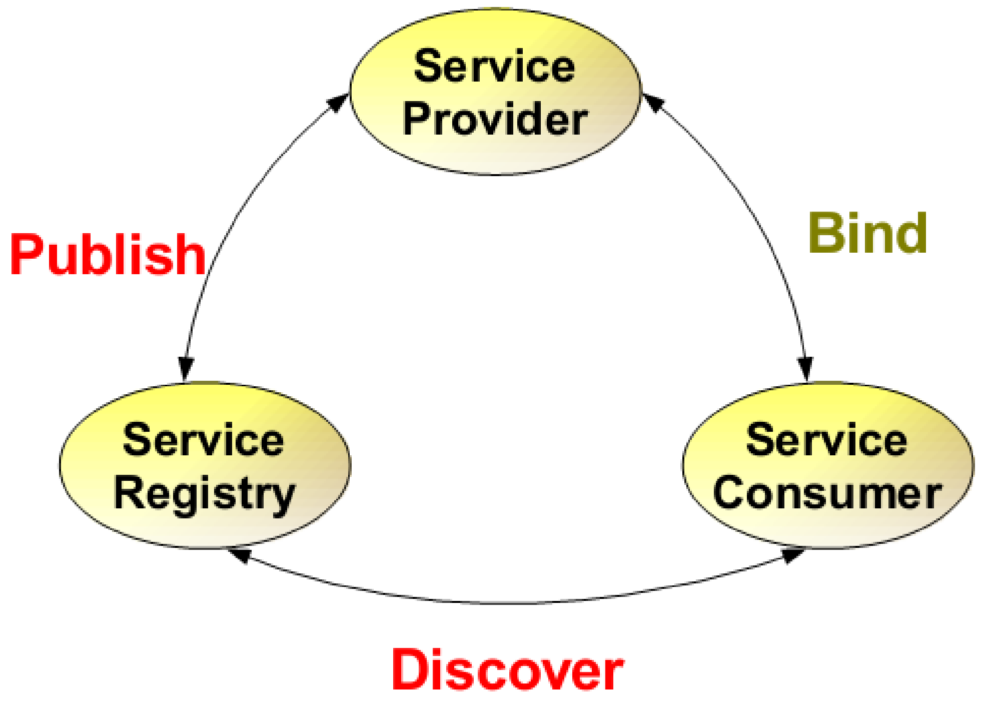
- What is service composition?
- what is business process?
- What is BPEL?
- How to create the business process in BPEL?
- What is the basic structure of BPEL document?

Service Discovery UDDI

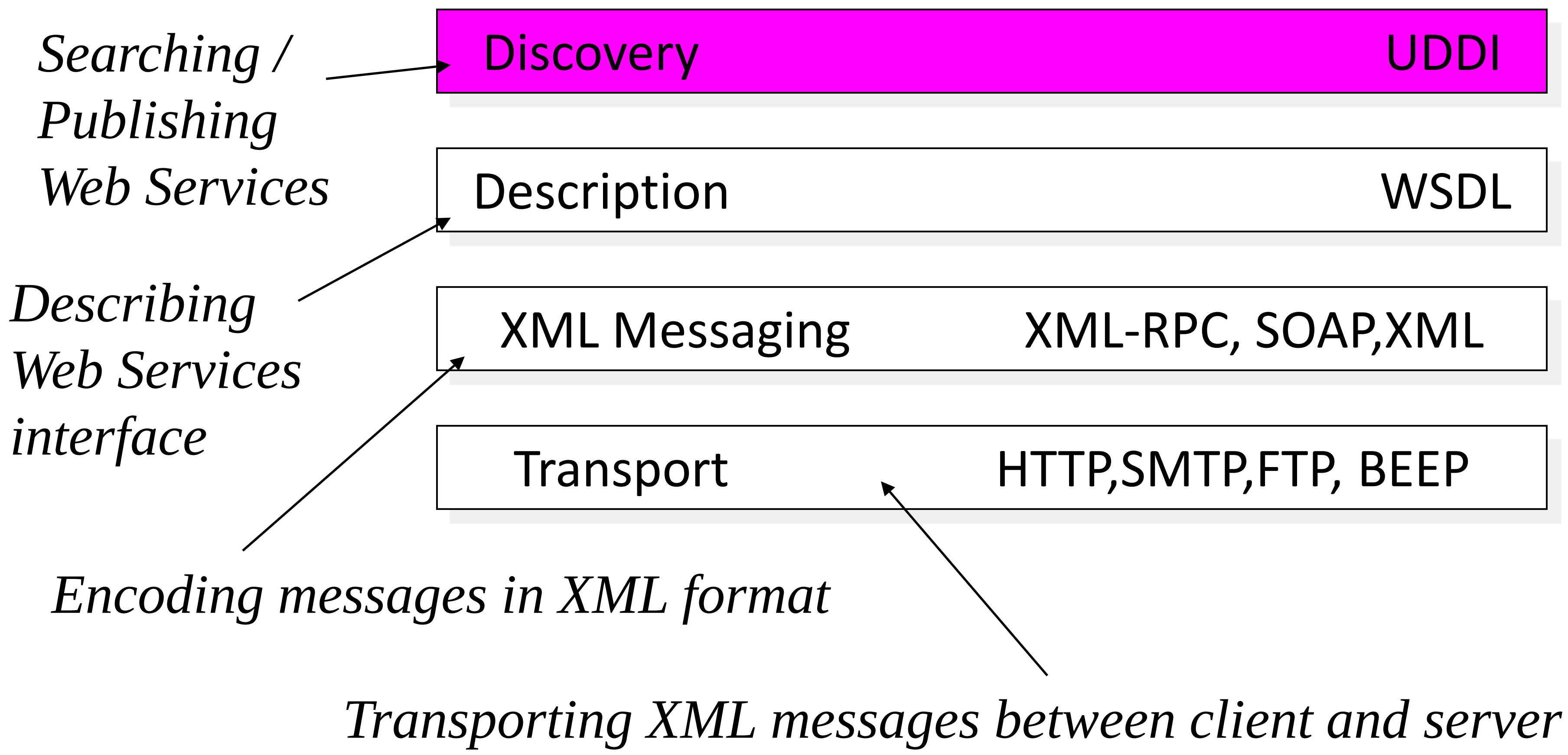

Service Provider
Service Consumer

Service Architecture

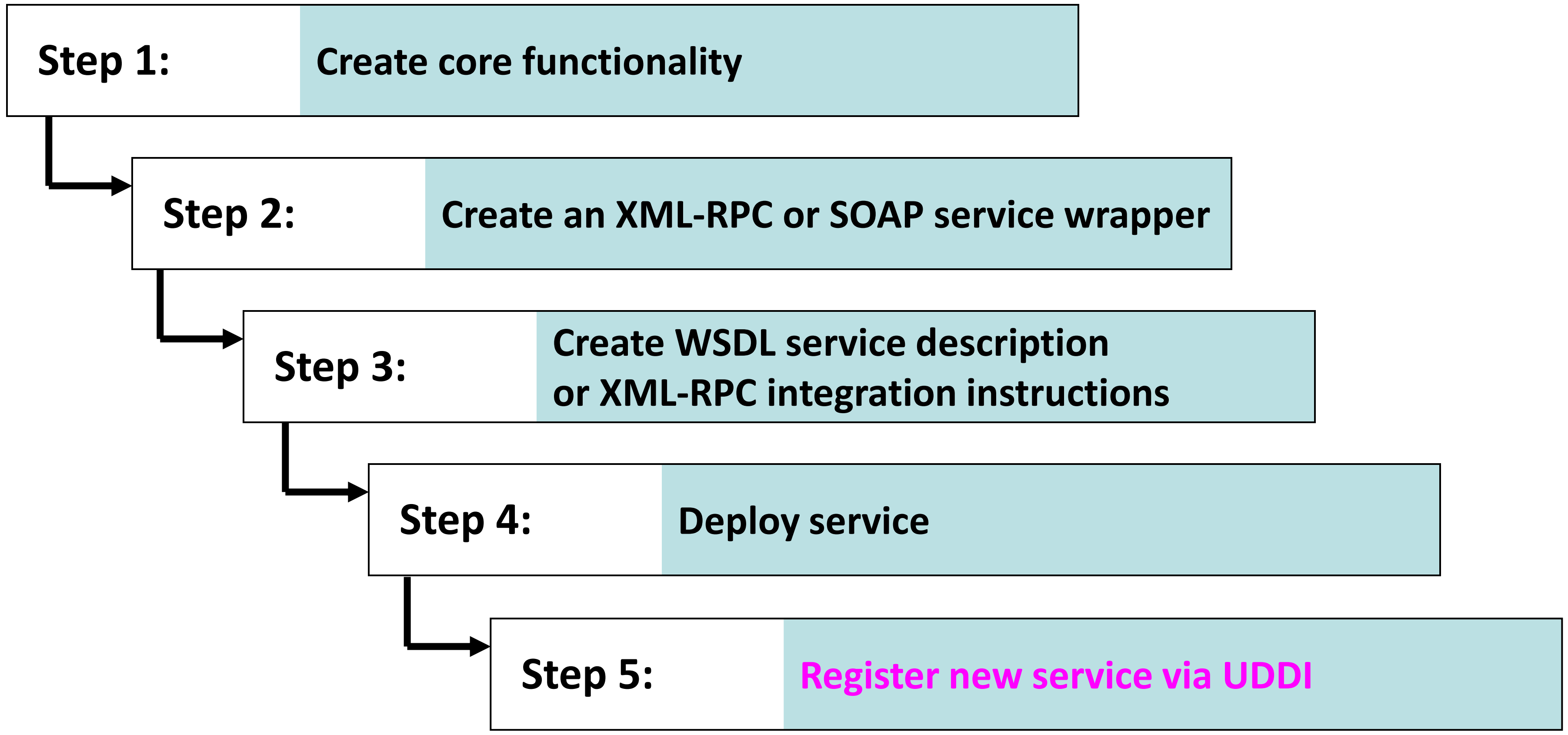
- UDDI(Universal Description, Discovery, and Integration) defines a scheme to publish and discover information about Web services



Web Service Protocol Stack


Discovery
UDDI
Description
WSDL
XML Messaging
XML-RPC, SOAP,XML
Transport
HTTP,SMTP,FTP, BEEP
Searching / Publishing Web Services
Describing Web Services interface
Encoding messages in XML format
Transporting XML messages between client and server

Using the Protocols Together – service provider perspective



Step 1: Create core functionality

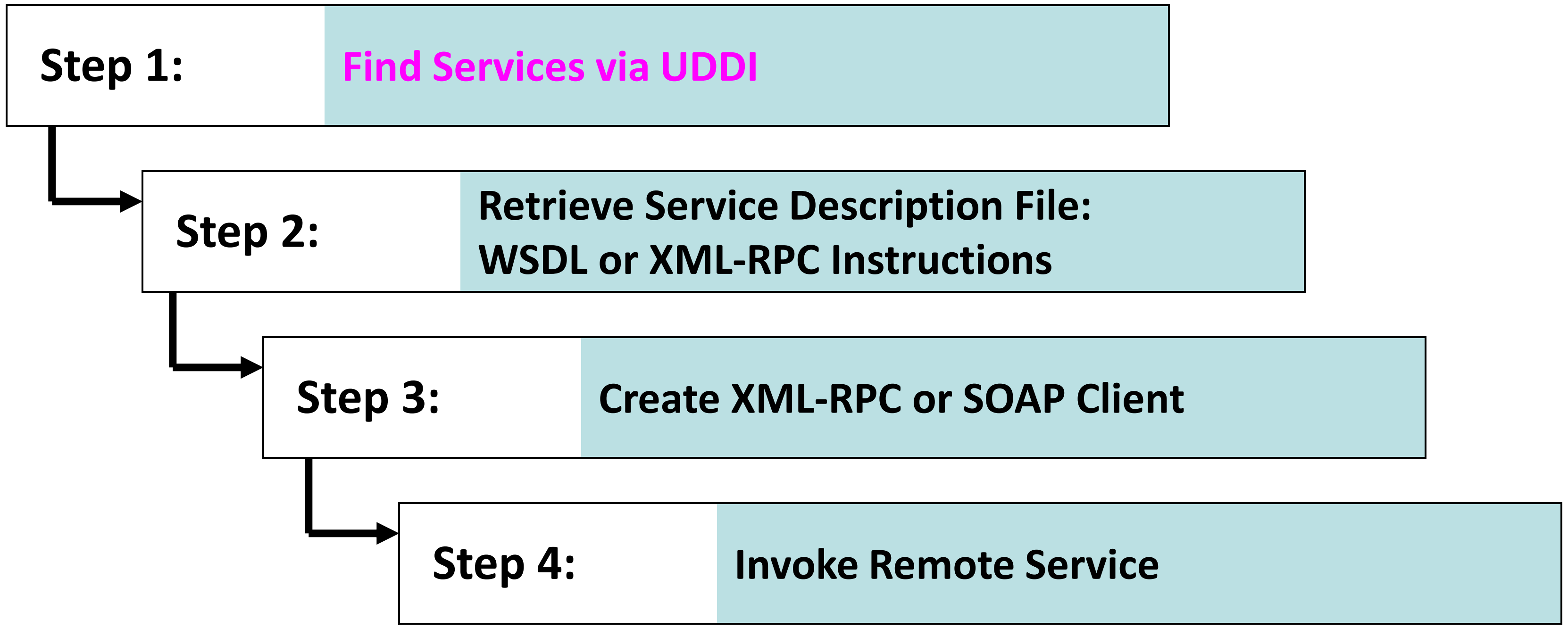
Step 2: Create an XML-RPC or SOAP service wrapper

Step 3: Create WSDL service description or XML-RPC integration instructions

Step 4: Deploy service

Step 5: Register new service via UDDI

Using the Protocols Together – service request perspective


Step 1:
Find Services via UDDI
Step 2:
Retrieve Service Description File:
WSDL or XML-RPC Instructions
Step 3:
Create XML-RPC or SOAP Client
Step 4:
Invoke Remote Service

Service discovery

- Automatic detection of devices and services offered by devices on a computer network

Service discovery

- Automatic detection of devices and services offered by these devices on a computer network
- Requires a common language to allow software agents to make use of one another’s services

Service discovery

- Automatic detection of devices and services offered by these devices on a computer network
- Requires a common language to allow software agents to make use of one another’s services
- Web Services Discovery

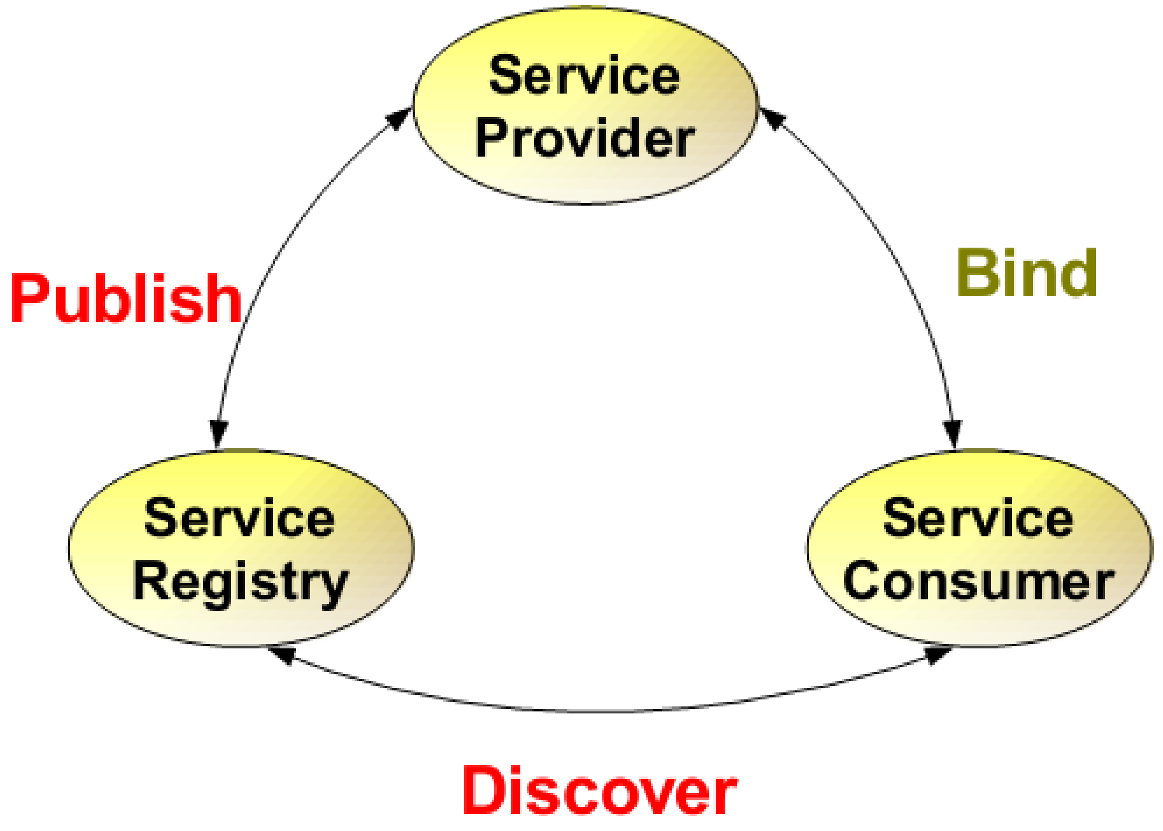
Service discovery

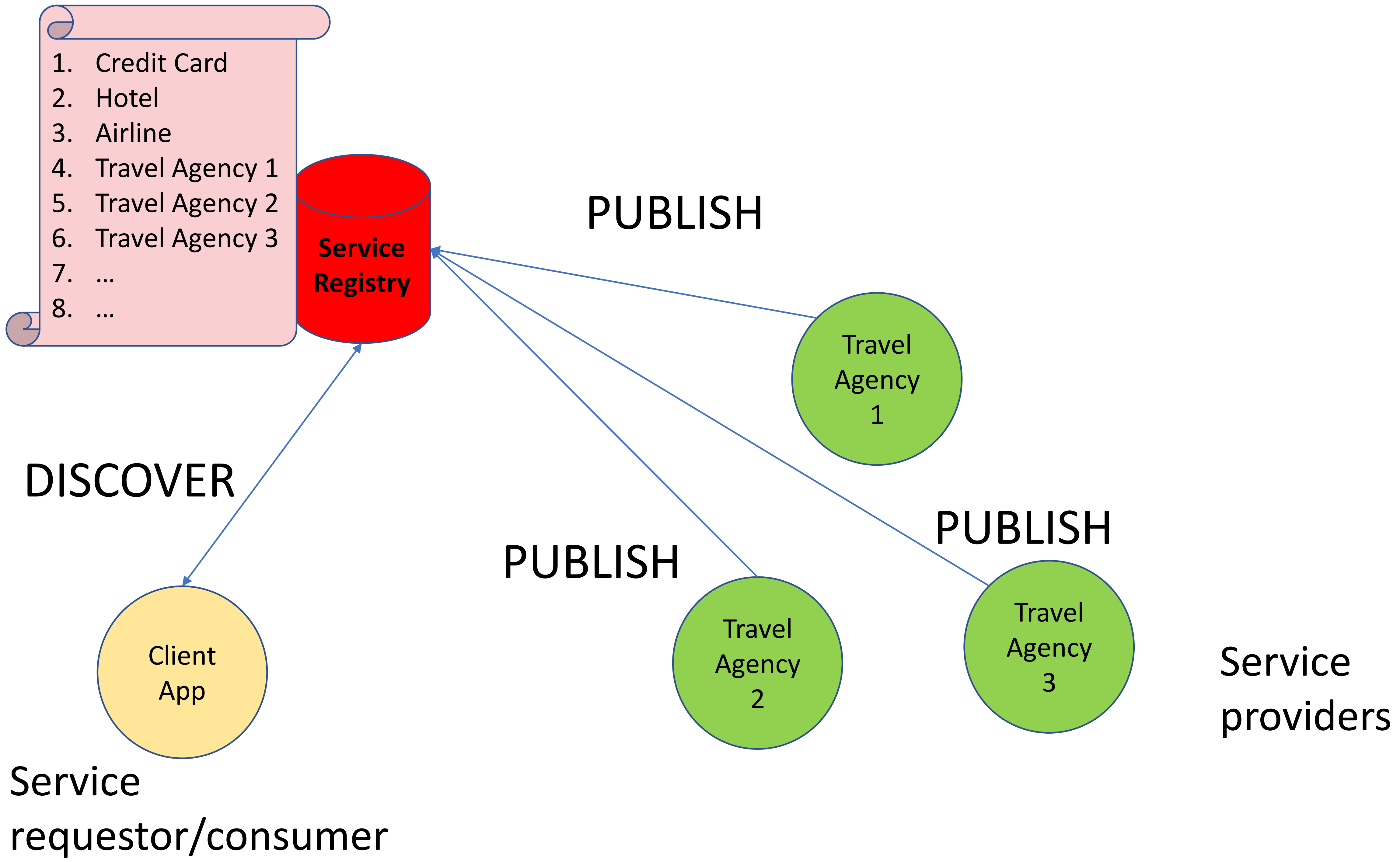
- Automatic detection of devices and services offered by these devices on a computer network
- Requires a common language to allow software agents to make use of one another’s services
- Web Services Discovery
  - provides access to software systems over the Internet using standard protocols

Service discovery

- Automatic detection of devices and services offered by these devices on a computer network
- Requires a common language to allow software agents to make use of one another’s services
- Web Services Discovery
  - provides access to software systems over the Internet using standard protocols
  - the process of finding suitable web services to a given task

Service discovery in Web Service Architecture




1. Credit Card
2. Hotel
3. Airline
4. Travel Agency 1
5. Travel Agency 2
6. Travel Agency 3
7. …
8. …
Service Registry
PUBLISH
DISCOVER
PUBLISH
PUBLISH
Travel Agency 1
Travel Agency 2
Travel Agency 3
Client App
Service requestor/consumer
Service providers

What is UDDI?

- A project to speed interoperability and adoption for web services

What is UDDI?

- A project to speed interoperability and adoption for web services
  - Standards-based <u>specifications</u> for service description and discovery

What is UDDI?

- A project to speed interoperability and adoption for web services
  - Standards-based <u>specifications</u> for service description and discovery
  - Shared <u>operation</u> of a business registry on the web

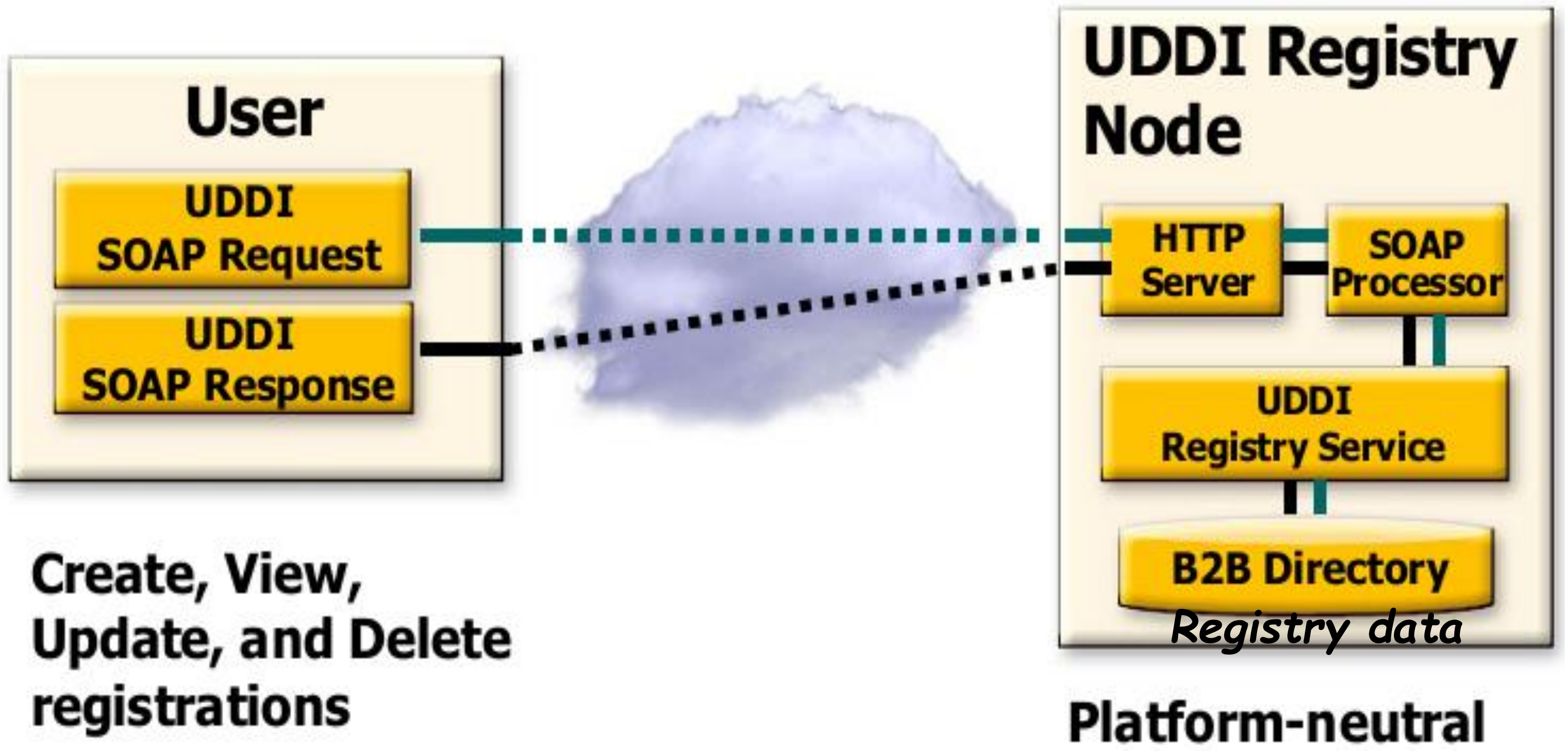
What is UDDI?

- A project to speed interoperability and adoption for web services
  - Standards-based <u>specifications</u> for service description and discovery
  - Shared <u>operation</u> of a business registry on the web
- Partnership among industry and business leaders

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

UDDI Runs “Over” SOAP



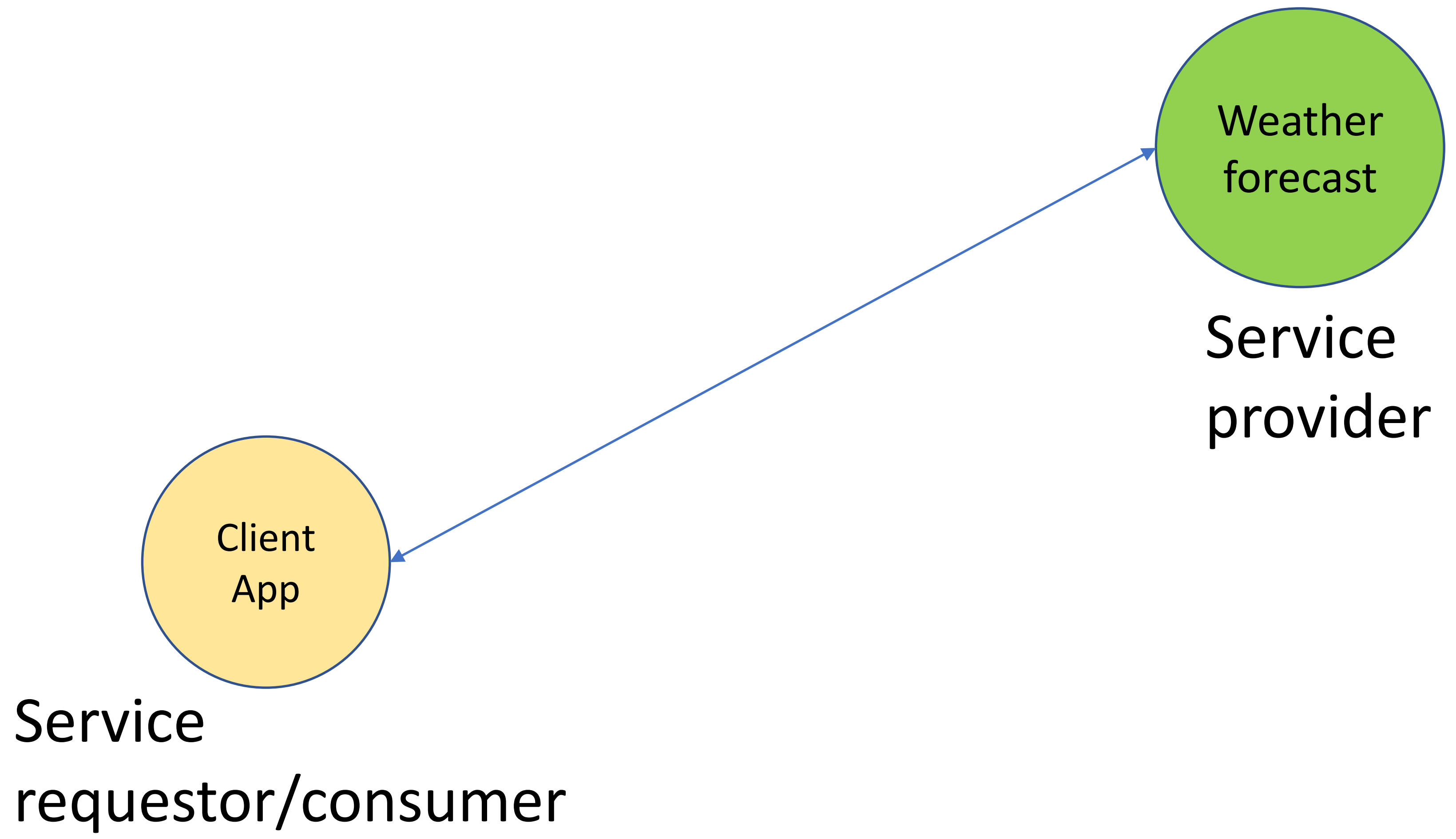
Why UDDI or something like UDDI?

- Platform independent service
  - publication and discovery

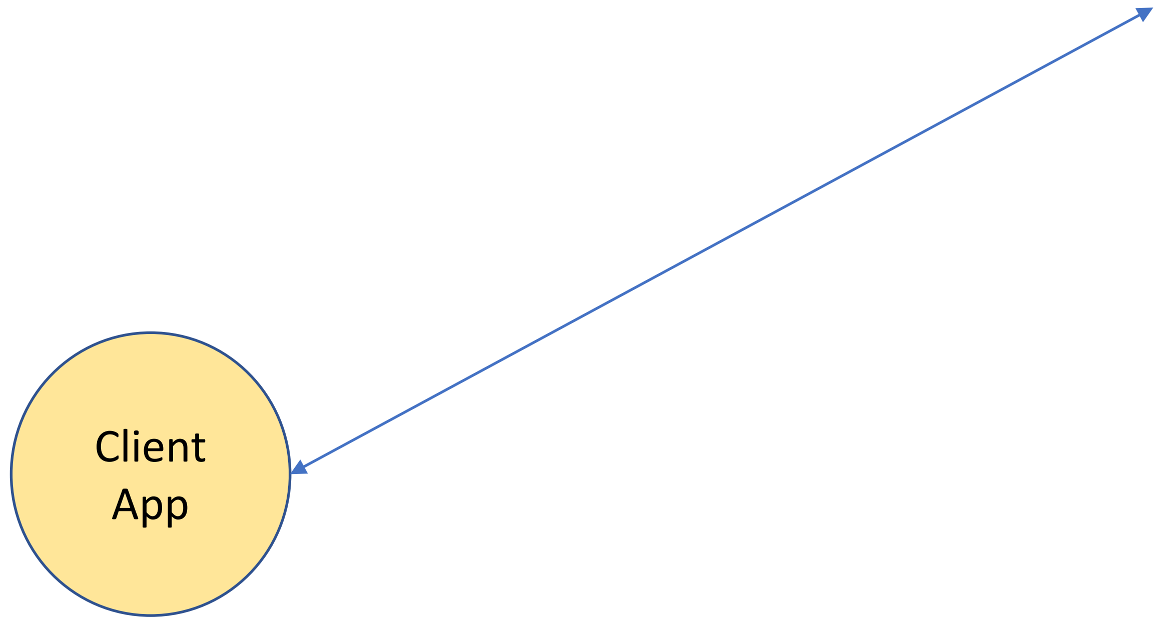
Why UDDI or something like UDDI?

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

Dynamic service discovery



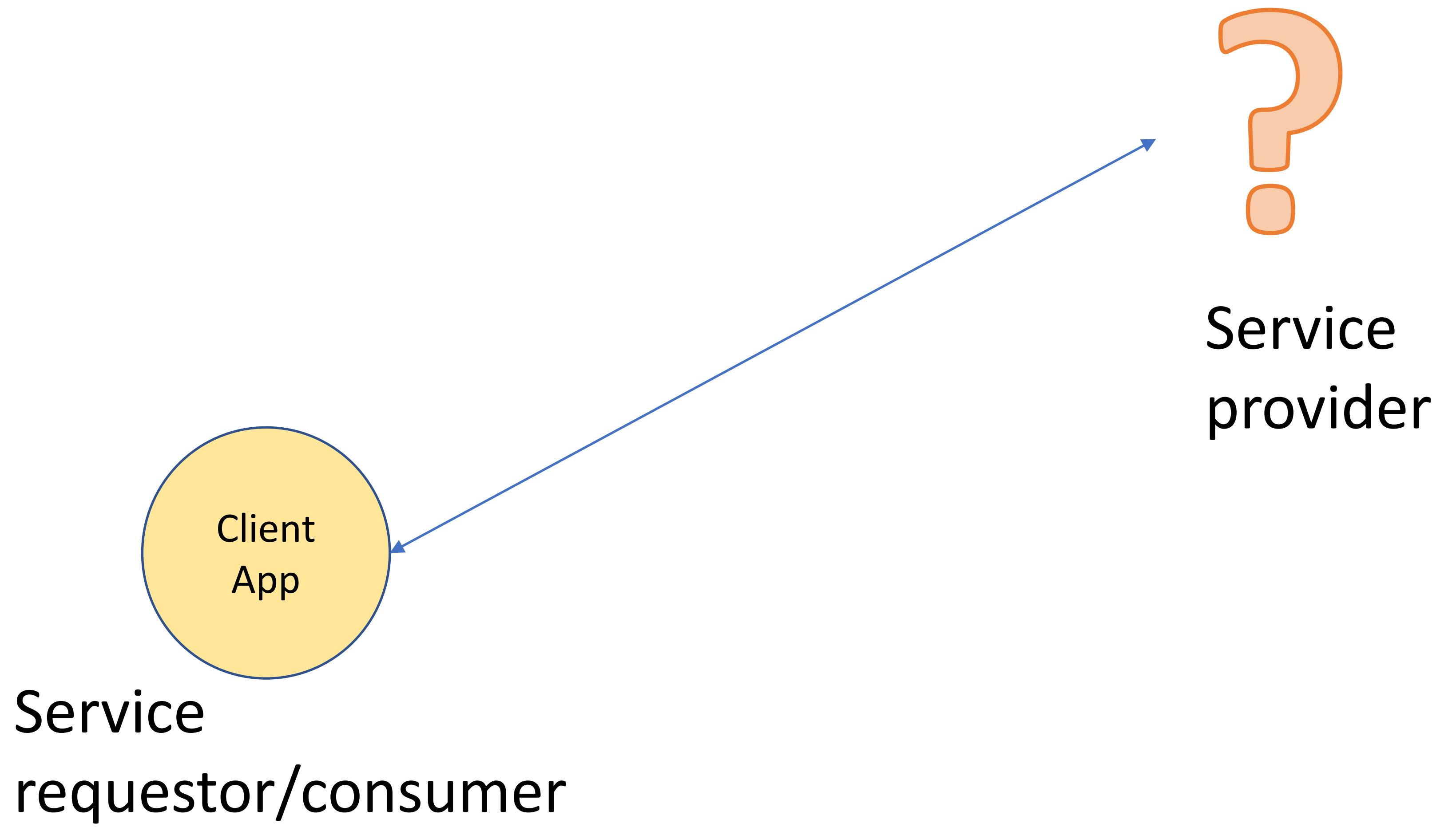
Dynamic service discovery



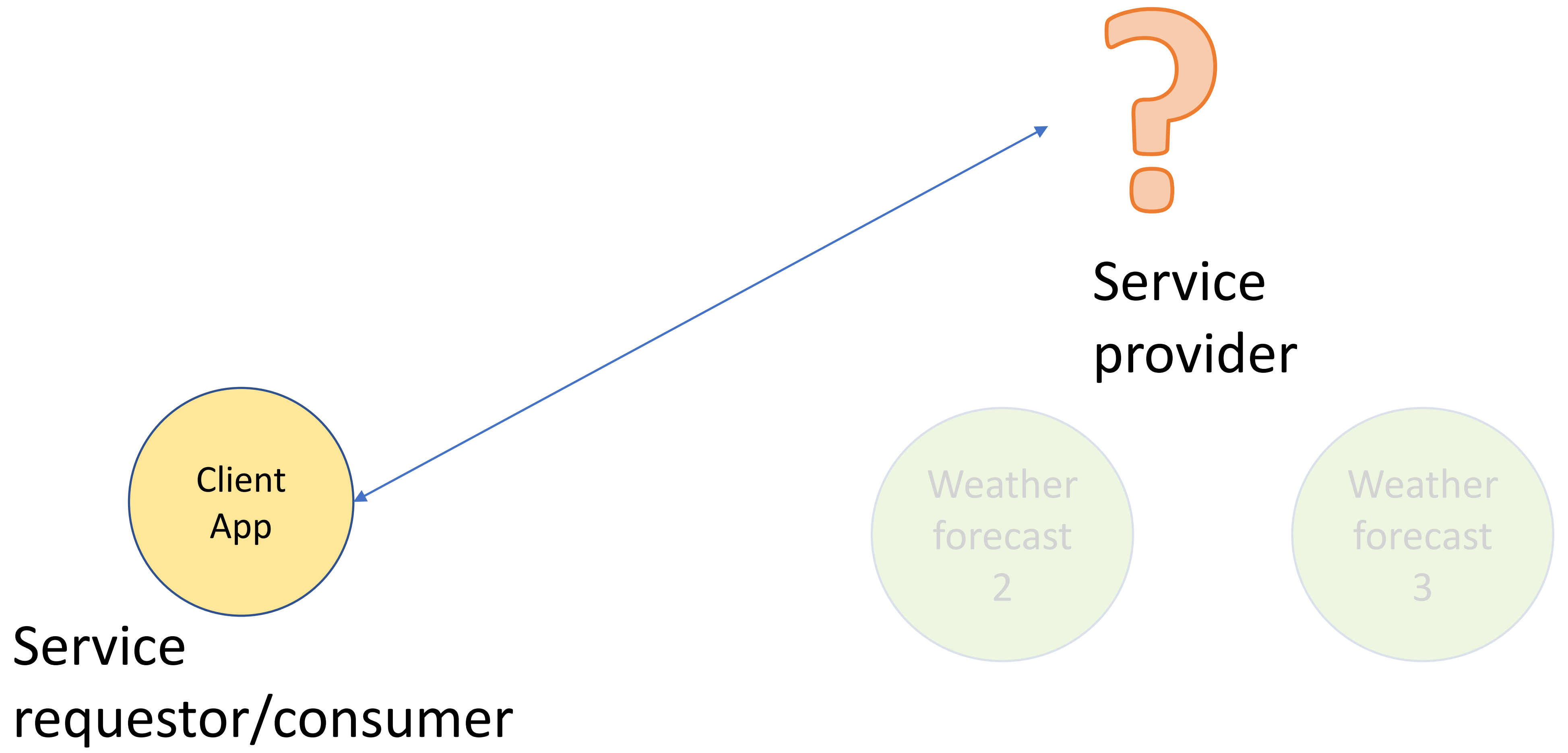
Service
provider

Service
requestor/consumer

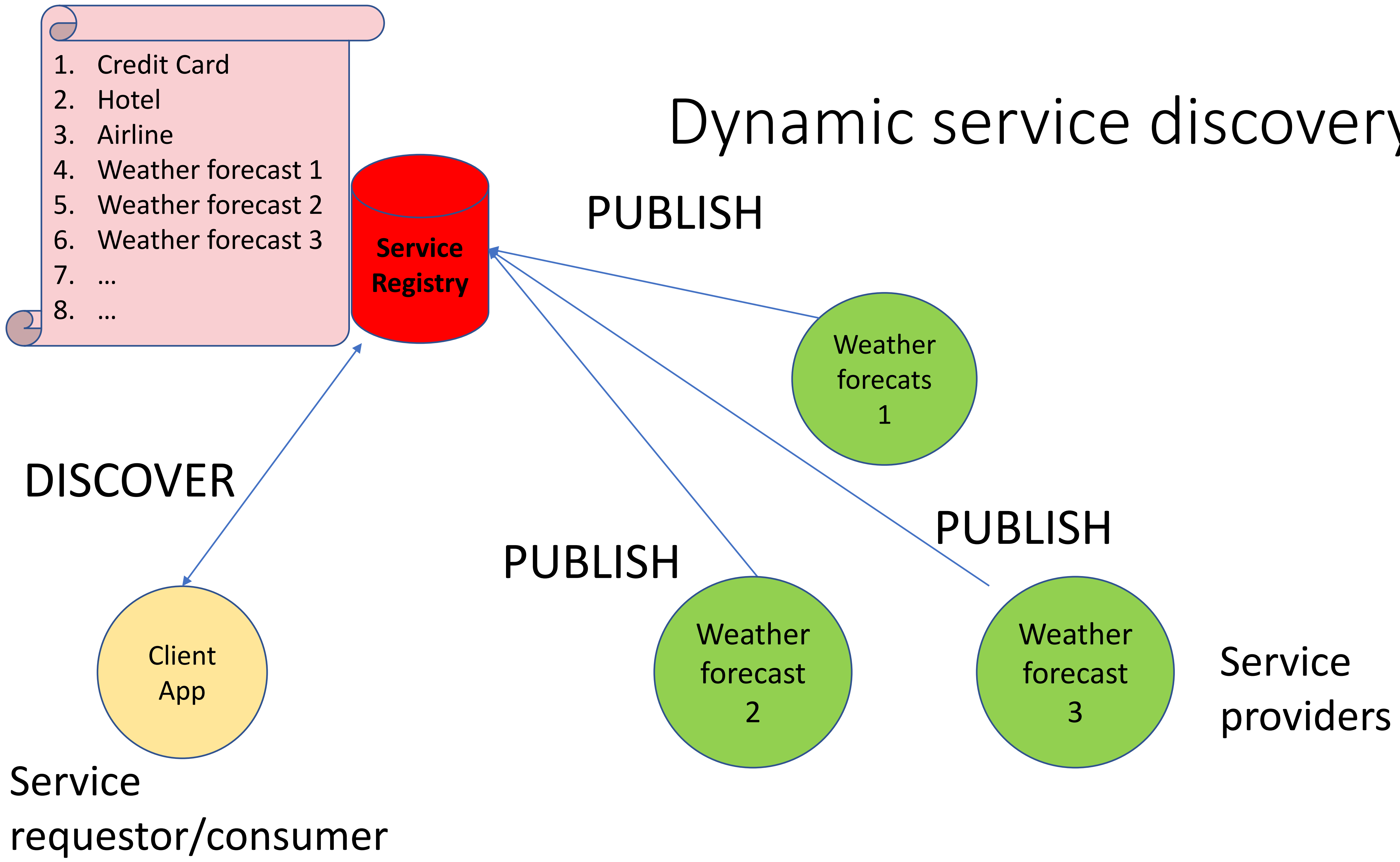
Dynamic service discovery



Dynamic service discovery



Dynamic service discovery



UDDI Vision - 2000

- Open industry initiative, enabling businesses to discover each other and define how they interact over the Internet

UDDI Vision - 2000

- Open industry initiative, enabling businesses to discover each other and define how they interact over the Internet
- Global e-commerce driven by dynamically emerging business relations

UDDI Vision - 2000

- Open industry initiative, enabling businesses to discover each other and define how they interact over the Internet
- Global e-commerce driven by dynamically emerging business relations
- Consumers of web services would be linked up with providers through a public or private dynamic brokerage system

UDDI Vision - 2000

- Open industry initiative, enabling businesses to discover each other and define how they interact over the Internet
- Global e-commerce driven by dynamically emerging business relations
- Consumers of web services would be linked up with providers through a public or private dynamic brokerage system
    - Anyone needing a service would go to their service broker and select a service supporting the desired SOAP service interface, and meeting other criteria

UDDI Vision - 2000

- Open industry initiative, enabling businesses to discover each other and define how they interact over the Internet
- Global e-commerce driven by dynamically emerging business relations
- Consumers of web services would be linked up with providers through a public or private dynamic brokerage system
    - Anyone needing a service would go to their service broker and select a service supporting the desired SOAP service interface, and meeting other criteria
    - The publicly operated UDDI node or broker would be critical for everyone

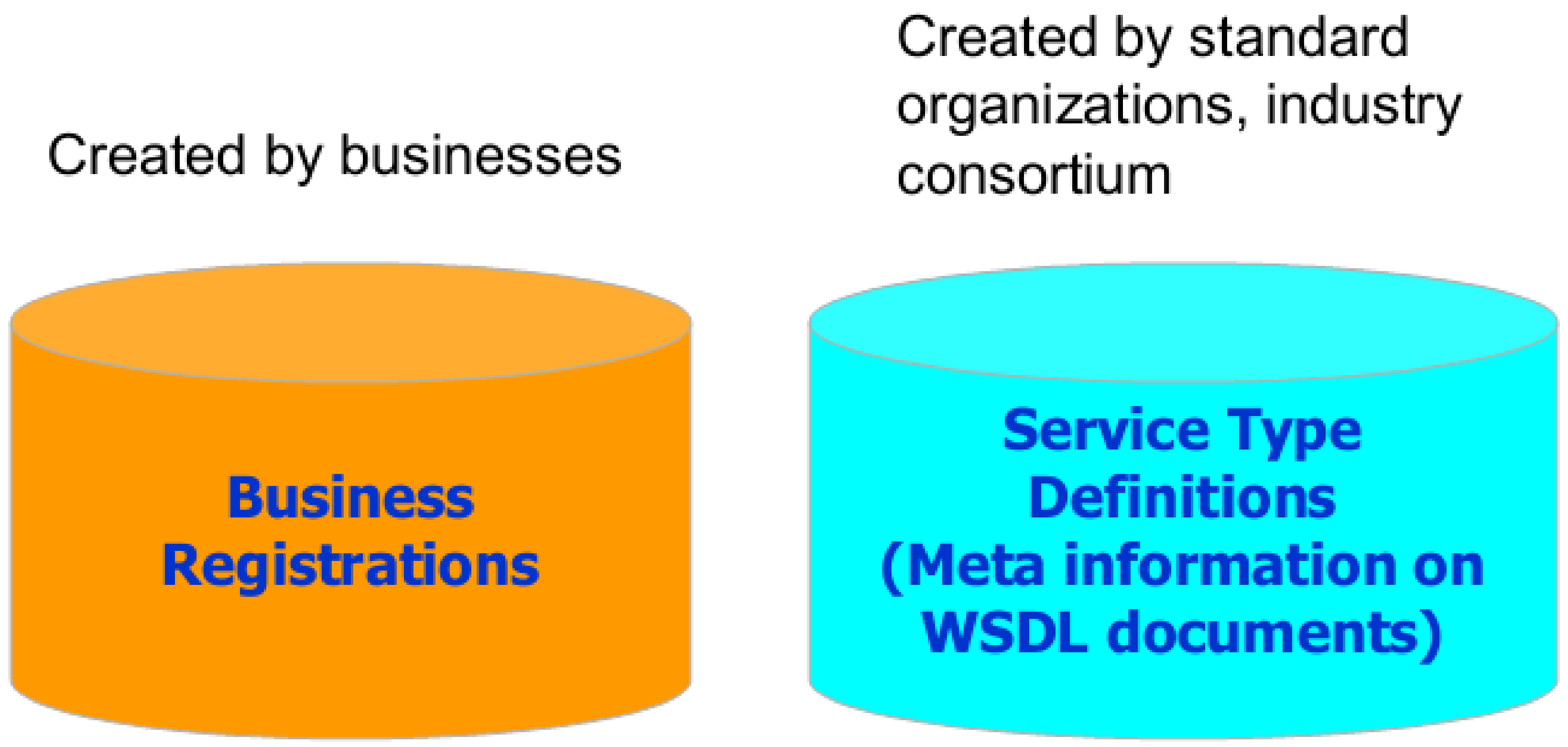
UDDI Vision - 2000

- Open industry initiative, enabling businesses to discover each other and define how they interact over the Internet
- Global e-commerce driven by dynamically emerging business relations
- Consumers of web services would be linked up with providers through a public or private dynamic brokerage system
  - Anyone needing a service would go to their service broker and select a service supporting the desired SOAP service interface, and meeting other criteria
  - The publicly operated UDDI node or broker would be critical for everyone
    - For the consumer – public or open brokers would only return services listed for public discovery by others

UDDI Vision - 2000

- Open industry initiative, enabling businesses to discover each other and define how they interact over the Internet
- Global e-commerce driven by dynamically emerging business relations
- Consumers of web services would be linked up with providers through a public or private dynamic brokerage system
  - Anyone needing a service would go to their service broker and select a service supporting the desired SOAP service interface, and meeting other criteria
  - The publicly operated UDDI node or broker would be critical for everyone
    - For the consumer – public or open brokers would only return services listed for public discovery by others
    - For service producer – matadata of index categories would be critical for effective placement

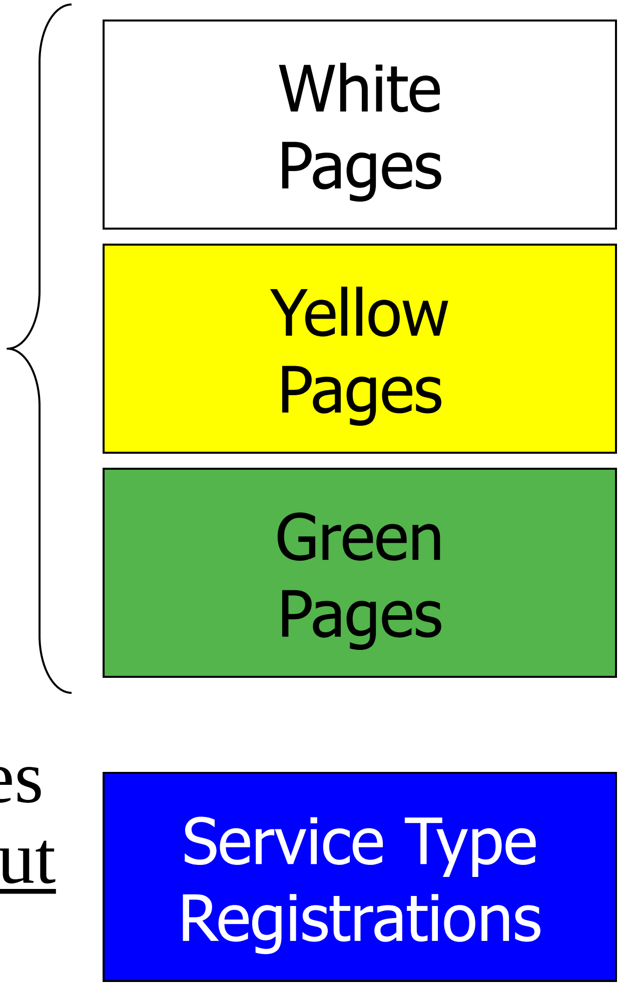
Registry Data


Created by businesses
Created by standard organizations, industry consortium
Business Registrations
Service Type Definitions (Meta information on WSDL documents)

Registry Data

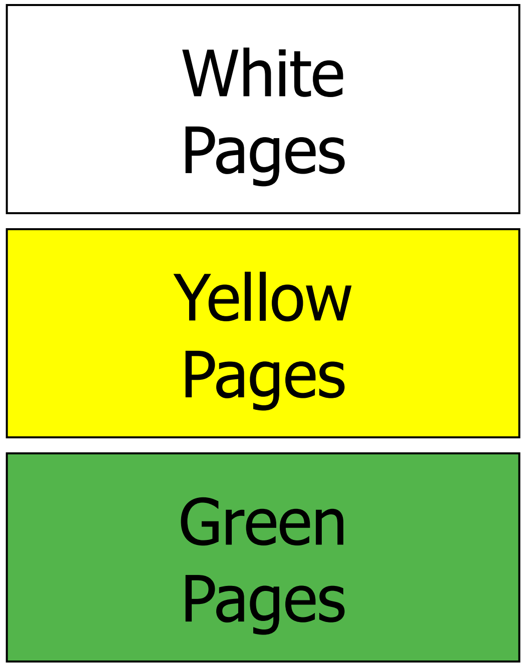
- Businesses register public information about themselves

- Standards bodies, Programmers, Businesses register <u>information about their Service Types</u>



Business Registration Data

- “White pages”
  - Business name, 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
  - a pointer to an external specification and an address for invoking the web service



What uses UDDI?

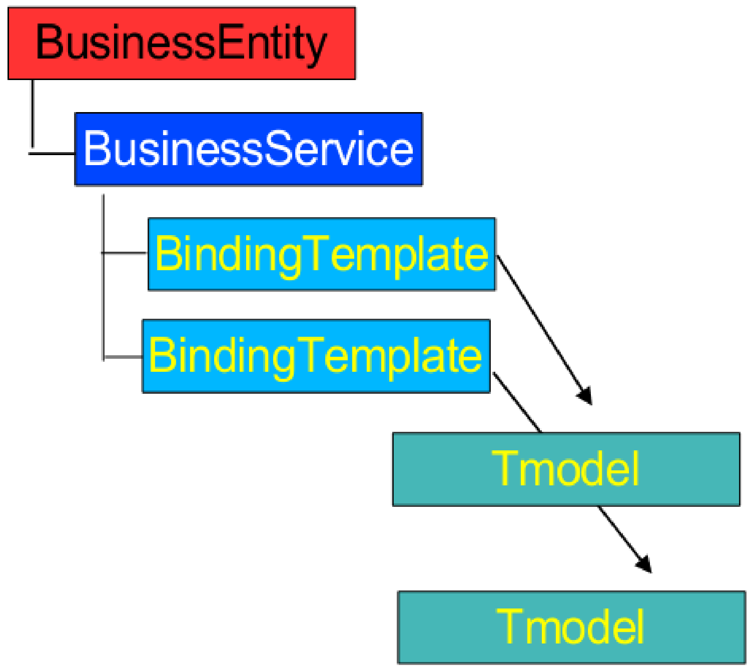
- Tool building client (Service Consumer)
  - Browse or search registry
  - Create a service proxy
- Tool publishing the service
  - Generates WSDL
  - Construct UDDI entries
- Application that needs dynamic binding
  - Directly access UDDI
  - Query can be pre-generated

UDDI Adoption Phases

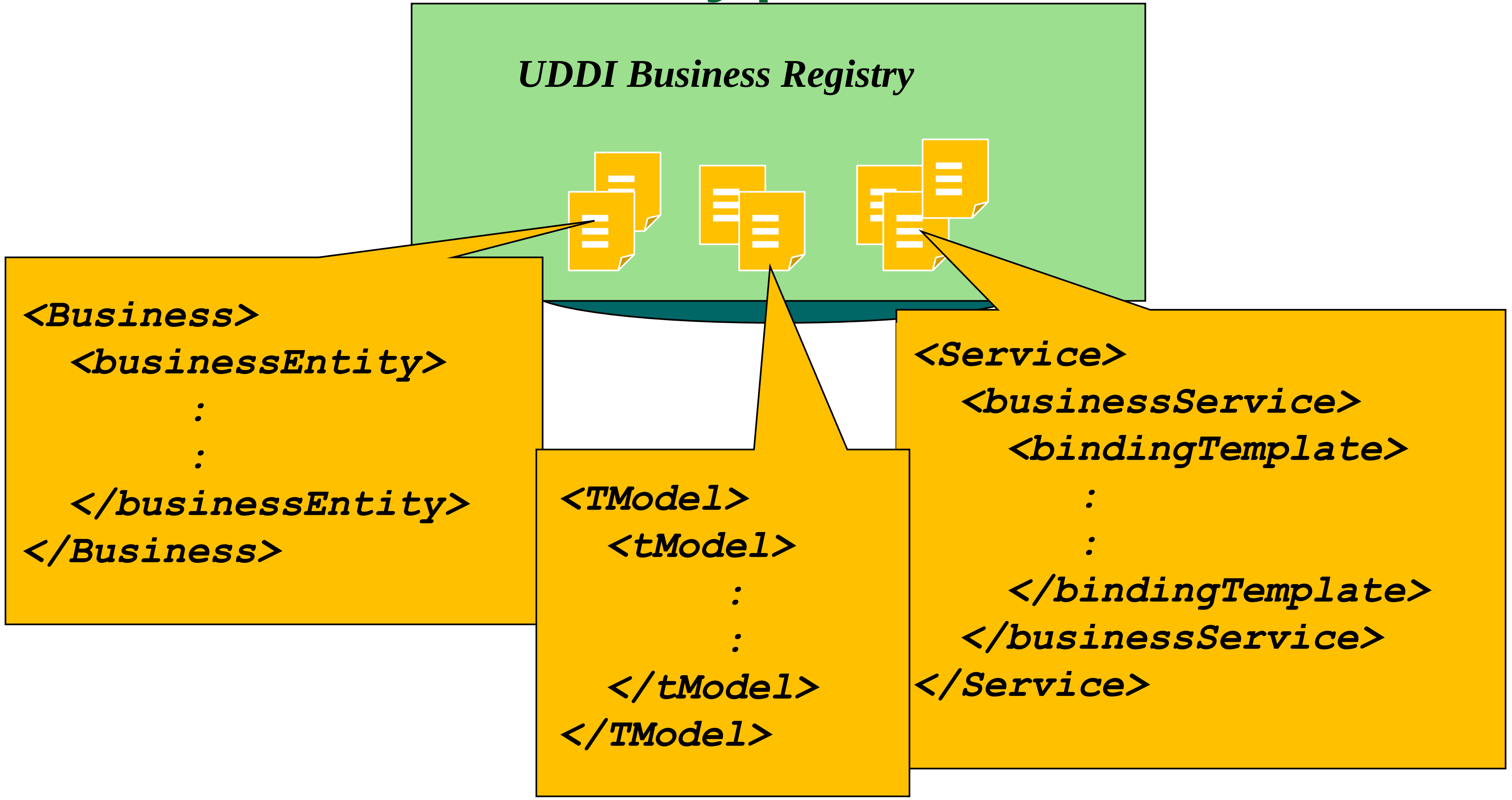
- Phase 1: Experimental stage
- Phase 2: Private UDDI registry within an intranet
- Phase 3: Public UDDI registries with no coordination among them
- Phase 4: Public UDDI registries with coordination (i.e. replication)
- Phase 5: Value added registry services

UDDI Data Model/Types

- UDDI includes an XML Schema that describes four core types of information:
  - businessEntity
    - About the actual business, e.g. business name, etc.
  - businessService
    - About the services provided by the business
  - bindingTemplate
    - About how and where to access a specific service
  - tModel (Technical Model)
    - Include descriptions and pointers to external technical specifications or taxonomies



UDDI Data Model/Types



```
<Business>
  <businessEntity>
        :
        :
  </businessEntity>
</Business>
```

```
<TModel>
  <tModel>
        :
        :
  </tModel>
</TModel>
```

```
<Service>
  <businessService>
    <bindingTemplate>
        :
        :
    </bindingTemplate>
  </businessService>
</Service>
```

XML Schema describes these four core types of information

A. businessEntity

```
<businessEntity
  businessKey=
     “ba744ed0-3aaf-11d5-80dc-002035229c64”>
  <name> XMethods </name>
  <description> … </description>
  <contacts>
    <contact> … </contact>
    <contact> … </contact>
  </contacts>
  <identifierBag> … </identifierBag>
  <categoryBag> … </categoryBag>
</businessEntity>
```

Typical contents of businessEntity element

A. businessEntity

- businessEntity element includes info about the actual business
  - Business name, description, contact info such as address, phone, contact person, etc.

- Each business will receive a unique businessKey value when registrating to a UDDI server
  - e.g. businessKey of Microsoft in its UDDI server: 0076b468-eb27-42e5-ac09-9955cff462a3

- The key is used to tie a business to its published services

A. businessEntity

```
<businessEntity
  businessKey=
      "ba744ed0-3aaf-11d5-80dc-002035229c64">
  <name> XMethods </name>
  <description> … </description>
  <contacts>
    <contact> … </contact>
    <contact> … </contact>
  </contacts>
  <identifierBag> … </identifierBag>
  <categoryBag> … </categoryBag>
</businessEntity>
```

Typical contents of businessEntity element

A. businessEntity

- Can also include other unique value(s) in identifierBag that identifies the company
  - UDDI supports Dun & Bradstreet D-U-N-S® Numbers and Thomas Registry Supplier IDs
  - e.g. Microsoft’s Dun & Bradstreet D-U-N-S® No:

    08-146-6849
- Businesses can also register multiple business categories in categoryBag based on standard taxonomies, e.g.
  - NAICS: The North American Industry Classification System provides industry classification
  - UNSPSC: Universal Standard Products and Service Classification provides product and service classification

A. businessEntity

Examples of identifierBag and categoryBag contents (Microsoft)

```
<identifierBag>
   <keyedReference
      tModelKey=
     “uuid:8609c81e-ee1f-4d5a-b202-3eb13ad01823”
      keyName=“D-U-N-S” keyValue=“08-146-6849” />
</identifierBag>
<categoryBag>
 <keyedReference
      tModelKey=
     “uuid:c0b9fe13-179f-413d-8a5b-5004db8e5bb2”
      keyName=“NAICS: Software Publisher”
      keyValue=“51121” />
 </categoryBag>
```

B. businessService

```
<businessService
   serviceKey=
     "d5921160-3e16-11d5-98bf-002035229c64"
   businessKey=
     "ba744ed0-3aaf-11d5-80dc-002035229c64">
   <name>XMethods Delayed Stock Quotes</name>
   <description> … </description>
   <bindingTemplates>
     <bindingTemplate>
       :
     </bindingTemplate>
   </bindingTemplates>
</businessService>
```

To tie the service with the business

Typical contents of businessService element

B. businessService

- businessService element includes info about a single web service or a group of related Web services

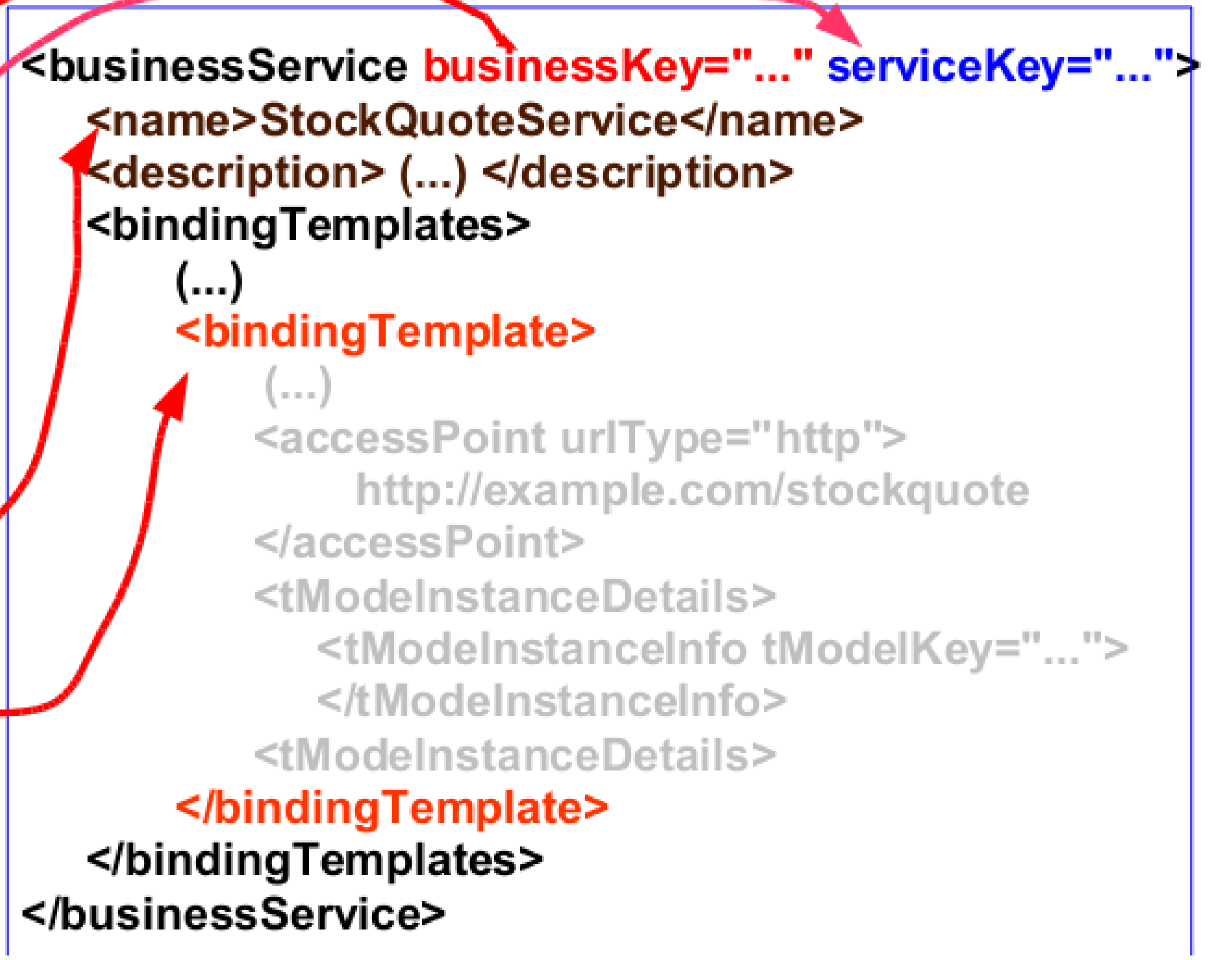
- Include the name, description and an optional list of bindingTemplates

- Like businessEnitity, each businessService has a unique service key

- Should specify the businessKey to relate with the business that provides that service

B. 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>
```

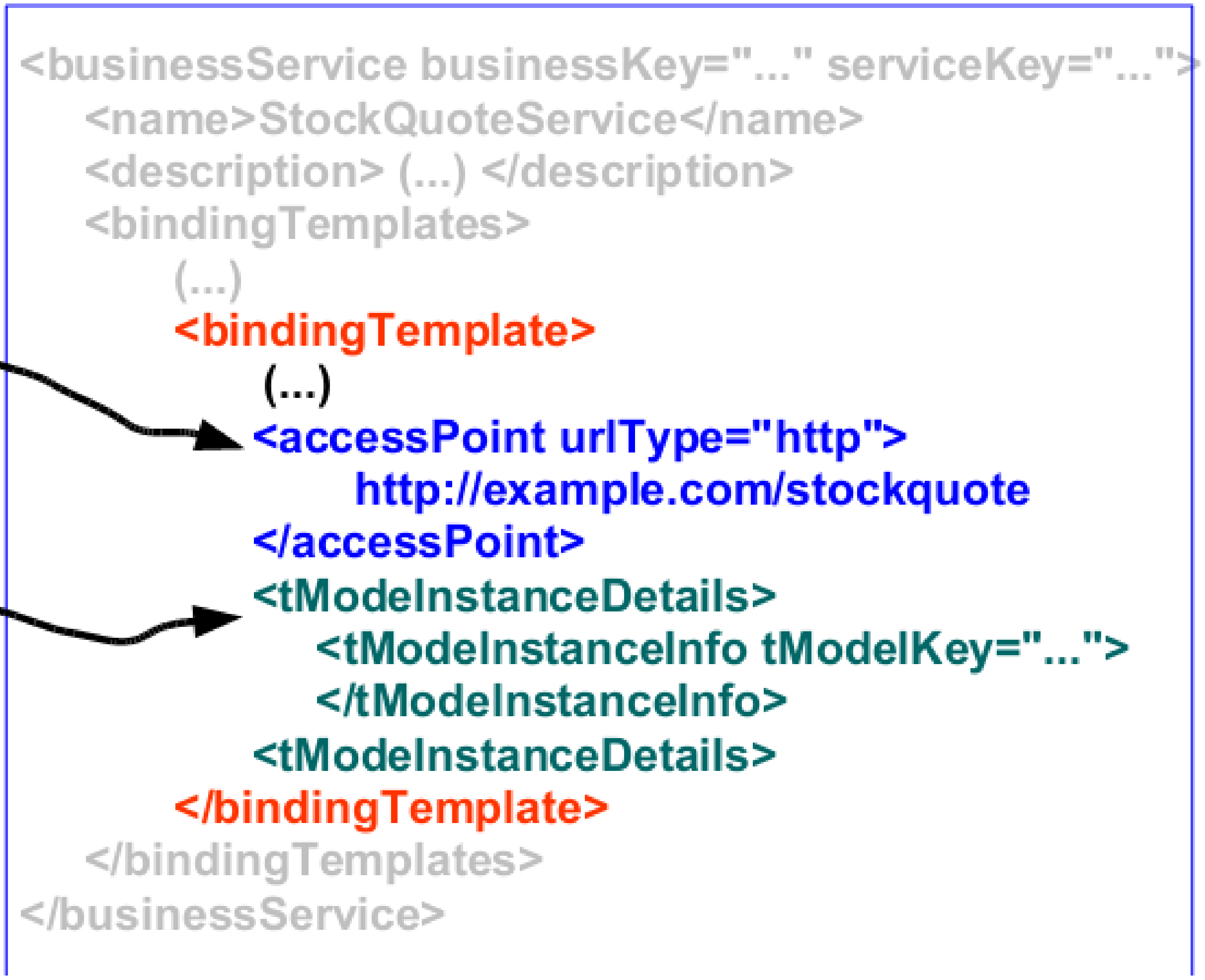
C. bindingTemplate

```
<bindingTemplate
  serviceKey=“d5921160-3e16-11d5-98bf-002035229c64”
  bindingKey=“…”>
  <description xml:lang=“en”>
     :
  </description>
  <accessPoint URLType=“http”>
    http://services.xmethods.net:80/soap
  </accessPoint>
  <tModelInstanceDetails>
     :
  </tModelInstanceDetails>
</bindingTemplate>
```

Typical contents of bindingTemplate element

C. bindingTemplate

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



```
<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>
```

UDDI binding options

Name	Description	UUID	Details
uddi-org:smtp	Email-based service	uuid:93335D49-3EFB-48A0-ACEA-EA102B60DDC6	Identifies a service that is invoked via SMTP email. For example, this could specify a person’s email address or an SMTP-based SOAP service.
uddi-org:fax	Fax-based service	uuid:1A2B00BE-6E2C-42F5-875B-56F32686E0E7	Identifies a service that is invoked via fax transmissions.
uddi-org:ftp	FTP-based service	uuid:1A2B00BE-6E2C-42F5-875B-56F32686E0E7	Identifies a service that is invoked via FTP.
uddi-org:telephone	Telephone-based service	uuid:38E12427-5536-4260-A6F9-B5B530E63A07	Identifies a service that is invoked via a telephone call. This could include interaction by voice and/or touch-tone.
uddi-org:http	HTTP-based service	uuid:68DE9E80-AD09-469D-8A37-088422BFBC36	Identifies a web service that is invoked via the HTTP protocol. This could reference a simple web page or a more complex HTTP-based SOAP application.

D. tModel

- tModels are primarily used to provide pointers to external technical specifications (e.g wsdl)

- bindingTemplate only provides info about where to access the SOAP binding, but not how to interface with it
- tModel element fills this gap by providing a pointer to an external specification, such as WSDL

- In fact, tModels are not reserved to Web services
- tModels are used whenever it is necessary to point to any external specification, such as the D-U-N-S® no.

Service Type Registration

- Pointer to the namespace where service type is described
  - What programmers read to understand how to use the service
- Identifier for who published the service
- Identifier for the service type registration
  - called a tModelKey
  - Used as a signature by web sites that implement those services

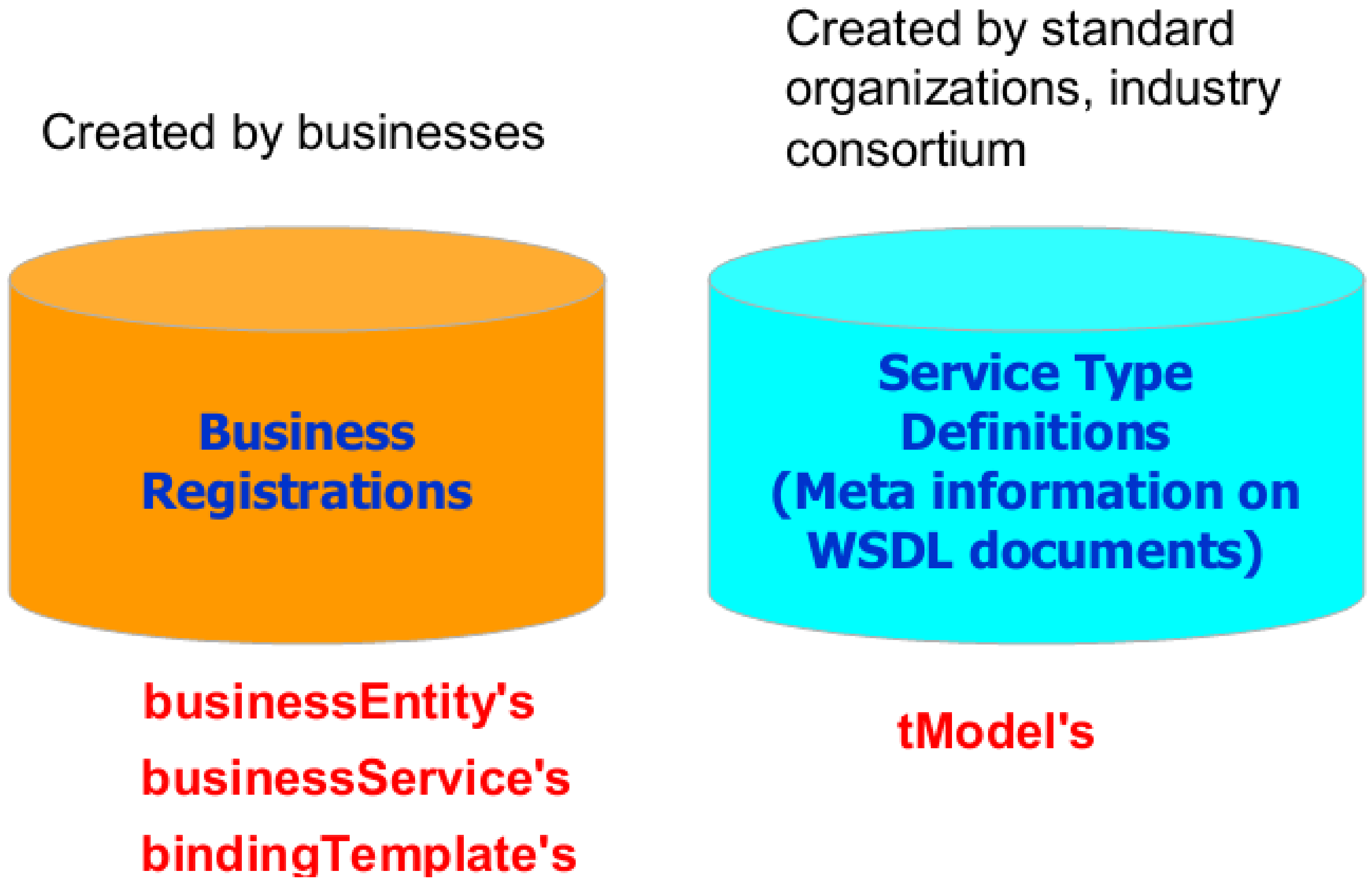
tModel Example

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

Registry Data


Created by businesses
Created by standard organizations, industry consortium
Business Registrations
Service Type Definitions (Meta information on WSDL documents)
businessEntity's
businessService's
bindingTemplate's
tModel's

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 inpu 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:

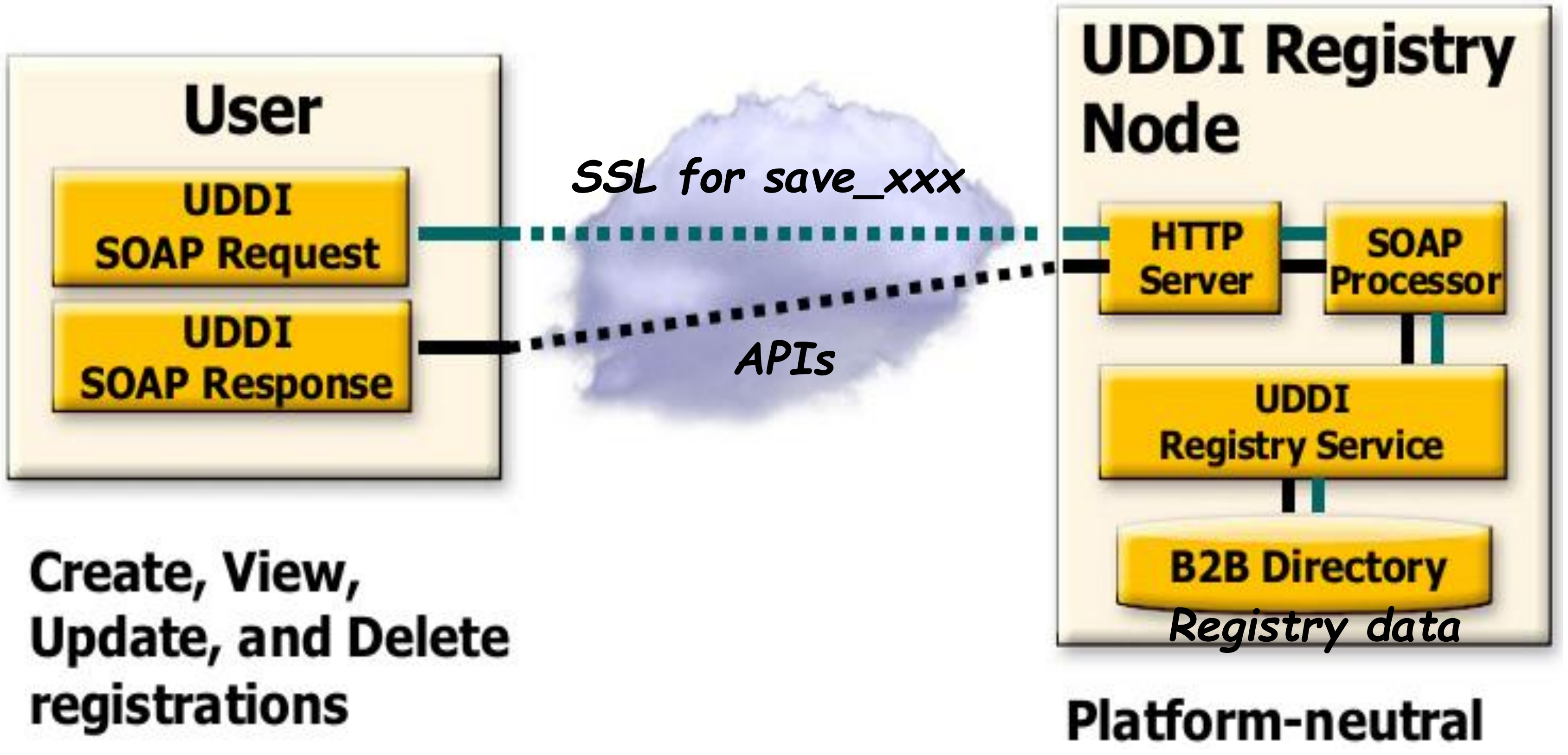
- request an authentication token
- inform registry that the authToken is no longer valid.

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 Runs “Over” SOAP



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>
```

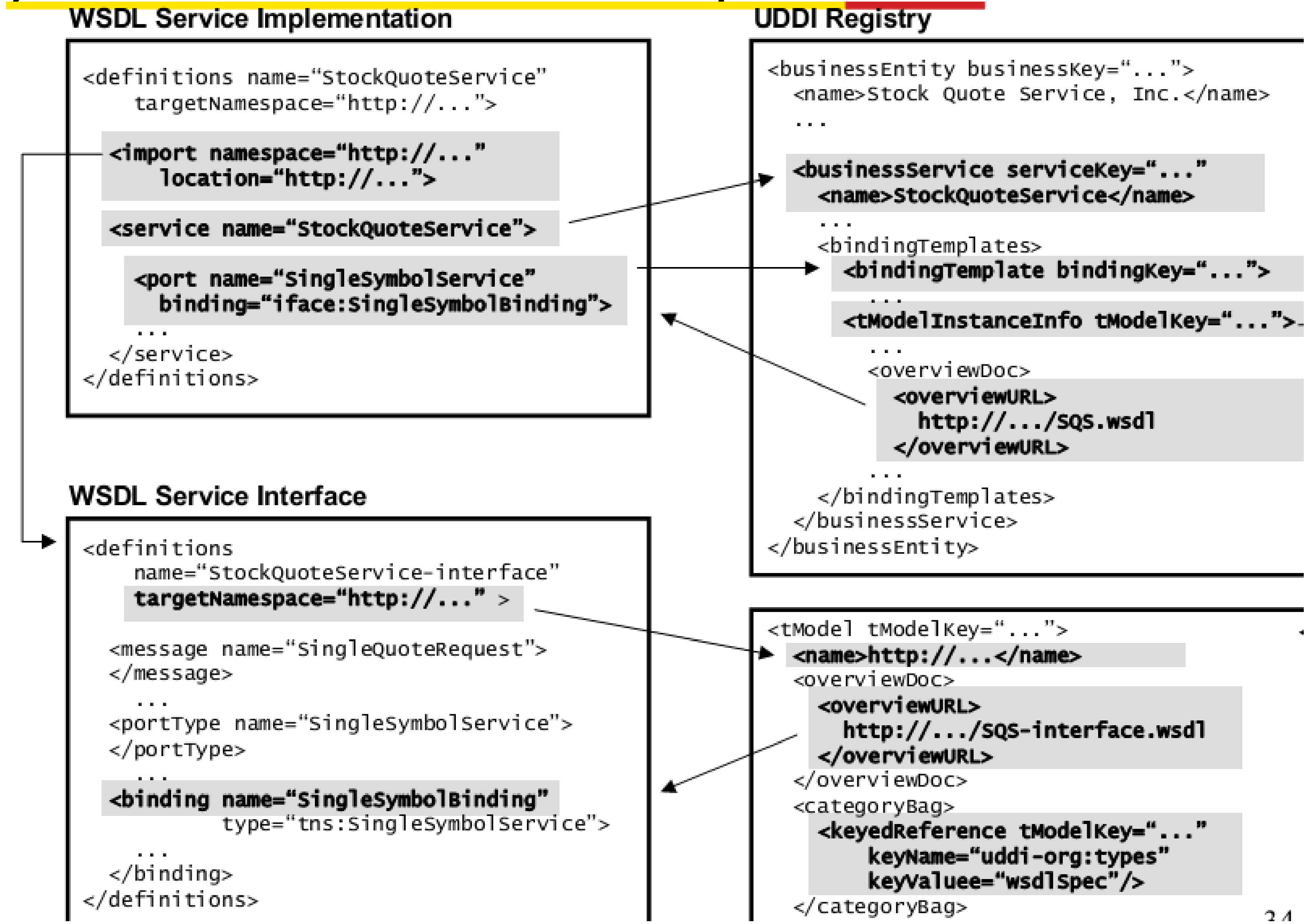
SOAP Message Example for get_serviceDetail response

```
<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>
```

http://www.ibm.com/developerworks/cn/webservices/ws-uwsdl/part1/

UDDI, WSDL Relationships



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

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

  <service name="StockQuoteService">

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

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

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

```
<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>
```

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

Steps that could be Performed by Industry Consortium (for tModel)

- 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
(for bindingTemplate)

- 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 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

UDDI discussion and review

Open Question Points: 10

Setting

What are your impressions after learning about UDDI technology? Is UDDI registry being used as intended? What are the problems with this approach to a Web services discovery?

Answer

Issues of UDDI

- How do you know if the data you get is valid, legitimate, and up to date?

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?

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)?

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?

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?

UDDI wide adoption failure

- 2006 - IBM, Microsoft and SAP closed their public UDDI nodes

UDDI wide adoption failure

- 2006 - IBM, Microsoft and SAP closed their public UDDI nodes
- 2007 – the group defining UDDI, the OASIS UDDI Specification Technical Committee has been closed

UDDI wide adoption failure

- 2006 - IBM, Microsoft and SAP closed their public UDDI nodes
- 2007 – the group defining UDDI, the OASIS UDDI Specification Technical Committee has been closed
- 2010 – Microsoft announce removing UDDI services from future versions of the Windows Server operating system
  - Moved this capability to BizTalk Server
  - 2016 – removed UDDI Services from BizTalk Server

Open Question Points: 10

Setting

In a 2000 vision, the **publicly operated** UDDI node or broker would return services listed for public discovery by others. We now know that this vision has failed. Do you know about alternative way to implement and use UDDI registries?

Answer

- We will continue this topic on Wednesday