Today’s lesson has 2 sets of slides

- The PREP slides already submitted to the blackboard
  - they contain more detailed information
- The discussion slides you will see in the online session today
  - These slides were prepared to help us make online session – discuss and practice the knowledge learnt from PREP slides

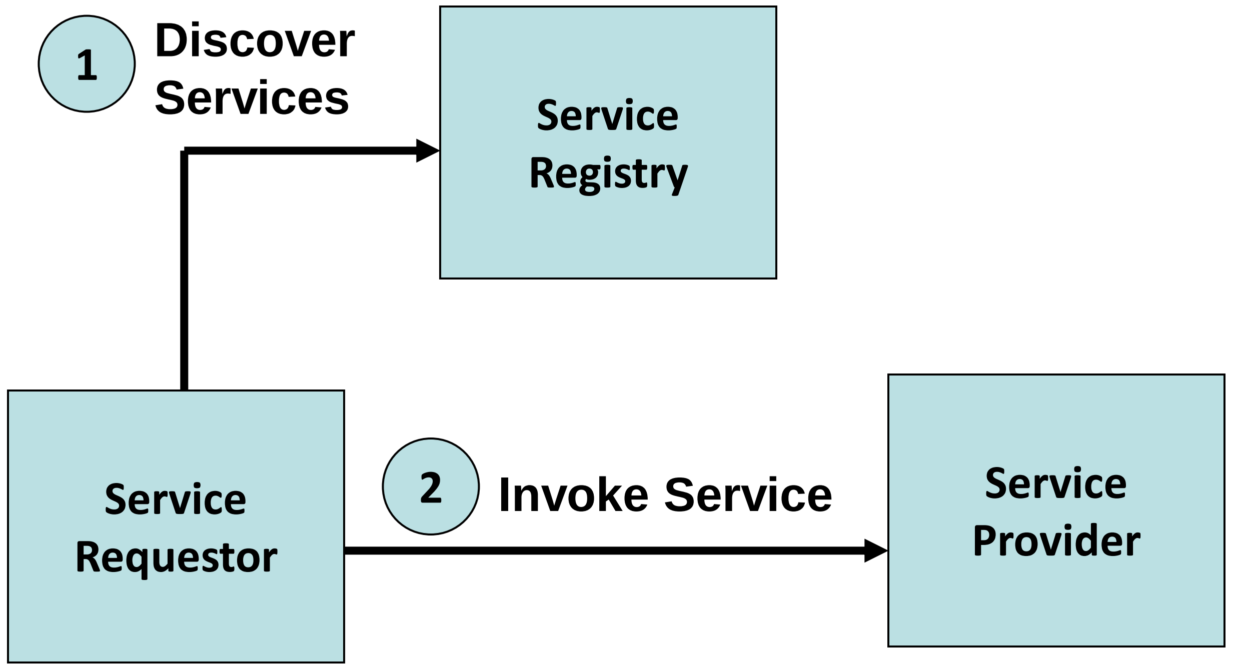
Module One: Introduction to Service Computing and XML-RPC

Web Services Architecture

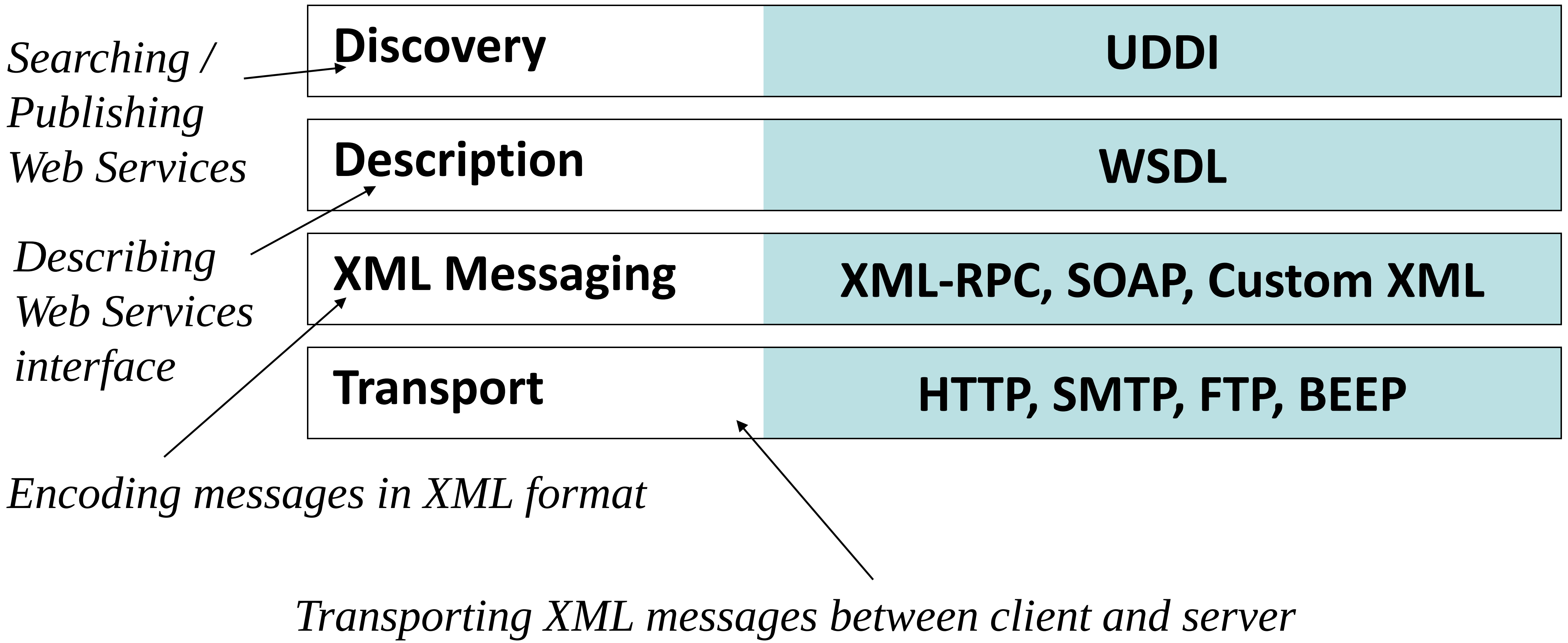
Web Service Architecture

- There are two ways to view the web service architectural framework:
  1) Examine individual roles of each web service actor
  2) Examine the emerging web service protocol stack.

Web Service Roles



Web Service Protocol Stack


Discovery
UDDI
Description
WSDL
XML Messaging
XML-RPC, SOAP, Custom 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

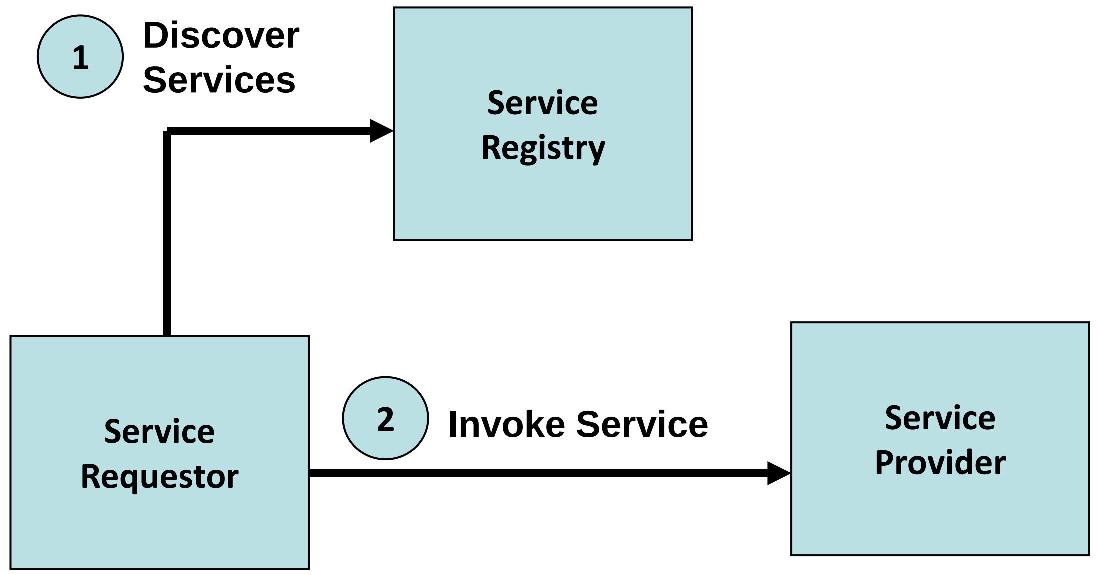
Part II:

Web Service Protocols

XML Messaging

Discovery	UDDI
Description	WSDL
XML Messaging	XML-RPC, SOAP, Custom XML
Transport	HTTP, SMTP, FTP, BEEP

Encoding messages in XML format



Option 1: XML-RPC

XML-RPC Example

- Here is a sample XML-RPC request to a weather service:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<methodCall>
  <methodName>weather.getWeather</methodName>
  <params>
    <param><value>10016</value></param>
  </params>
</methodCall>
```

"Give me the current weather conditions in zip code: 10016."

XML-RPC Example

- Here is a sample XML-RPC request to a weather service:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<methodCall>
  <methodName>weather.getWeather</methodName>
  <params>
    <param><value>10016</value></param>
  </params>
</methodCall>
```

"Give me the current weather conditions in zip code: 10016."

XML-RPC Example

- Here is a sample XML-RPC request to a weather service:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<methodCall>
  <methodName>weather.getWeather</methodName>
  <params>
    <param><value>10016</value></param>
  </params>
</methodCall>
```

"Give me the current weather conditions in zip code: 10016."

XML-RPC Example

- Here is a sample Weather response:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<methodResponse>
   <params>
      <param>
         <value><int>65</int></value>
      </param>
   </params>
</methodResponse>
```

“Current temperature is 65 degrees”

XML-RPC Example

- Here is a sample Weather response:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<methodResponse>
  <params>
    <param>
      <value><int>65</int></value>
    </param>
  </params>
</methodResponse>
```

"Current temperature is 65 degrees"

Option 2: SOAP

SOAP 1.1 Example

- Here is a sample SOAP request to a weather service:

```xml
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://www.w3.org/2001/09/soap-envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:getWeather
      xmlns:ns1="urn:examples:weatherservice"
      SOAP-ENV:encodingStyle="http://www.w3.org/2001/09/soap-encoding/">
      <zipcode xsi:type="xsd:string">10016</zipcode>
    </ns1:getWeather>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP 1.1 Example

"Give me the current weather conditions in zip code: 10016."

- Here is a sample SOAP request to a weather service:

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://www.w3.org/2001/09/soap-envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:getWeather
      xmlns:ns1="urn:examples:weatherservice"
      SOAP-ENV:encodingStyle="http://www.w3.org/2001/09/soap-encoding/">
      <zipcode xsi:type="xsd:string">10016</zipcode>
    </ns1:getWeather>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP 1.1 Example:

- Here is a sample SOAP response:

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://www.w3.org/2001/09/soap-envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:getWeatherResponse
      xmlns:ns1="urn:examples:weatherservice"
      SOAP-ENV:encodingStyle="http://www.w3.org/2001/09/soap-encoding/">
      <return xsi:type="xsd:int">65</return>
    </ns1:getWeatherResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP 1.1 Example:

“Current temperature is 65 degrees”

- Here is a sample SOAP response:

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://www.w3.org/2001/09/soap-envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:getWeatherResponse
      xmlns:ns1="urn:examples:weatherservice"
      SOAP-ENV:encodingStyle="http://www.w3.org/2001/09/soap-encoding/">
      <return xsi:type="xsd:int">65</return>
    </ns1:getWeatherResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Open Question Points: 10

Setting

What is the main difference between XML-RPC and SOAP?

Open Question is only supported on Version 2.0 or newer.

Answer

Weather request

XML-RPC

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<methodCall>
  <methodName>weather.getWeather</methodName>
  <params>
    <param><value>10016</value></param>
  </params>
</methodCall>
```

SOAP

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://www.w3.org/2001/09/soap-envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:getWeather
      xmlns:ns1="urn:examples:weatherservice"
      SOAP-ENV:encodingStyle="http://www.w3.org/2001/09/soap-encoding/">
      <zipcode xsi:type="xsd:string">10016</zipcode>
    </ns1:getWeather>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Weather response

XML-RPC

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<methodResponse>
  <params>
    <param>
      <value><int>65</int></value>
    </param>
  </params>
</methodResponse>
```

SOAP

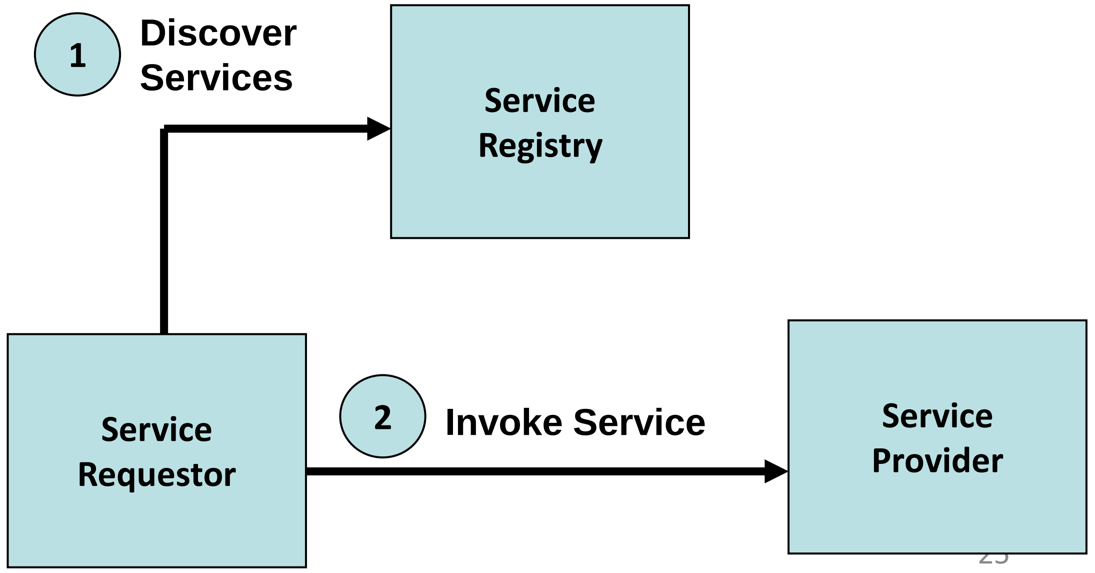
```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://www.w3.org/2001/09/soap-envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:getWeatherResponse
      xmlns:ns1="urn:examples:weatherservice"
      SOAP-ENV:encodingStyle="http://www.w3.org/2001/09/soap-encoding/">
      <return xsi:type="xsd:int">65</return>
    </ns1:getWeatherResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

XML-RPC vs SOAP

- XML-RPC is the easiest way to get started with web services.
  - Simpler than SOAP
  - Simpler data structures for transmitting data.

Discovery	**UDDI**
Description	**WSDL**
XML Messaging	**XML-RPC, SOAP, Custom XML**
Transport	**HTTP, SMTP, FTP, BEEP**

Describing Web Services interface



WSDL

- WSDL: Web Service Description Language.
- WSDL is an XML grammar for specifying an interface for a web service.
- Specifies
  - location of web service
  - methods that are available by the web service
  - data type information for all XML messages
- WSDL is commonly used to describe SOAP services.

WSDL In a Nutshell

<definitions>: Root WSDL Element

- **<types>: What data types will be transmitted?**
- **<message>: What messages will be transmitted?**
- **<portType>: What operations (functions) will be supported?**
- **<binding>: What SOAP specific details are there?**
- **<service>: Where is the service located?**

WSDL Excerpt: Weather Service

```
<message name="getWeatherRequest">
  <part name="zipcode" type="xsd:string"/>
</message>
<message name="getWeatherResponse">
  <part name="temperature" type="xsd:int"/>
</message>

<portType name="Weather_PortType">
  <operation name="getWeather">
    <input message="tns:getWeatherRequest"/>
    <output message="tns:getWeatherResponse"/>
  </operation>
</portType>
```

```
…
 <service name="Weather_Service">
     <documentation>WSDL File for
    Weather Service</documentation>
     <port binding="tns:Weather_Binding"
               name="Weather_Port">
               <soap:address
          location="http://ecerami.com/soap/servlet/rpcrouter"/>
     </port>
   </service>
</definitions>
```

WSDL Excerpt: Weather Service

```
<message name="getWeatherRequest">
  <part name="zipcode" type="xsd:string"/>
</message>
<message name="getWeatherResponse">
  <part name="temperature" type="xsd:int"/>
</message>

<portType name="Weather_PortType">
  <operation name="getWeather">
    <input message="tns:getWeatherRequest"/>
    <output message="tns:getWeatherResponse"/>
  </operation>
</portType>
```

<types>: What data types will be transmitted?

WSDL Excerpt: Weather Service

```
<message name="getWeatherRequest">
  <part name="zipcode" type="xsd:string"/>
</message>
<message name="getWeatherResponse">
  <part name="temperature" type="xsd:int"/>
</message>

<portType name="Weather_PortType">
  <operation name="getWeather">
    <input message="tns:getWeatherRequest"/>
    <output message="tns:getWeatherResponse"/>
  </operation>
</portType>
```

<types>: What data types will be transmitted?

WSDL Excerpt: Weather Service

```
<message name="getWeatherRequest">
  <part name="zipcode" type="xsd:string"/>
</message>
<message name="getWeatherResponse">
  <part name="temperature" type="xsd:int"/>
</message>

<portType name="Weather_PortType">
  <operation name="getWeather">
    <input message="tns:getWeatherRequest"/>
    <output message="tns:getWeatherResponse"/>
  </operation>
</portType>
```

<message>: What messages will be transmitted?

WSDL Excerpt: Weather Service

```
<message name="getWeatherRequest">
  <part name="zipcode" type="xsd:string"/>
</message>
<message name="getWeatherResponse">
  <part name="temperature" type="xsd:int"/>
</message>

<portType name="Weather_PortType">
  <operation name="getWeather">
    <input message="tns:getWeatherRequest"/>
    <output message="tns:getWeatherResponse"/>
  </operation>
</portType>
```

<message>: What messages will be transmitted?

WSDL Excerpt: Weather Service

```
<message name="getWeatherRequest">
  <part name="zipcode" type="xsd:string"/>
</message>
<message name="getWeatherResponse">
  <part name="temperature" type="xsd:int"/>
</message>

<portType name="Weather_PortType">
  <operation name="getWeather">
    <input message="tns:getWeatherRequest"/>
    <output message="tns:getWeatherResponse"/>
  </operation>
</portType>
```

<portType>: What operations (functions) will be supported?

WSDL Excerpt: Weather Service

```
<message name="getWeatherRequest">
  <part name="zipcode" type="xsd:string"/>
</message>
<message name="getWeatherResponse">
  <part name="temperature" type="xsd:int"/>
</message>

<portType name="Weather_PortType">
  <operation name="getWeather">
    <input message="tns:getWeatherRequest"/>
    <output message="tns:getWeatherResponse"/>
  </operation>
</portType>
```

<portType>: What operations (functions) will be supported?

WSDL Excerpt: Weather Service

```
…
 <service name="Weather_Service">
     <documentation>WSDL File for
    Weather Service</documentation>
     <port binding="tns:Weather_Binding"
                name="Weather_Port">
                <soap:address
          location="http://ecerami.com/soap/servlet/rpcrouter"/>
     </port>
   </service>
</definitions>
```

<binding>: What SOAP specific details are there?

WSDL Excerpt: Weather Service

```
…
 <service name="Weather_Service">
     <documentation>WSDL File for
    Weather Service</documentation>
     <port binding="tns:Weather_Binding"
                name="Weather_Port">
                <soap:address
          location="http://ecerami.com/soap/servlet/rpcrouter"/>
     </port>
   </service>
</definitions>
```

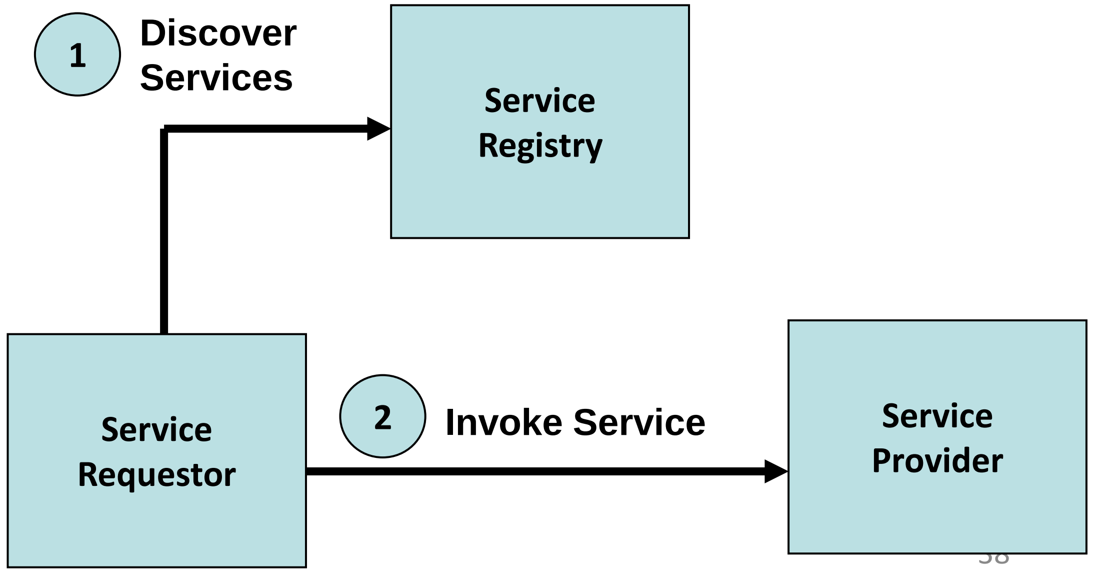
<service>: Where is the service located?

So What?

- Given a WSDL file, a developer can immediately figure out how to connect to the web service.
- Eases overall integration process.
- Better yet, with WSDL tools, you can *automate* the integration…

Searching / Publishing Web Services

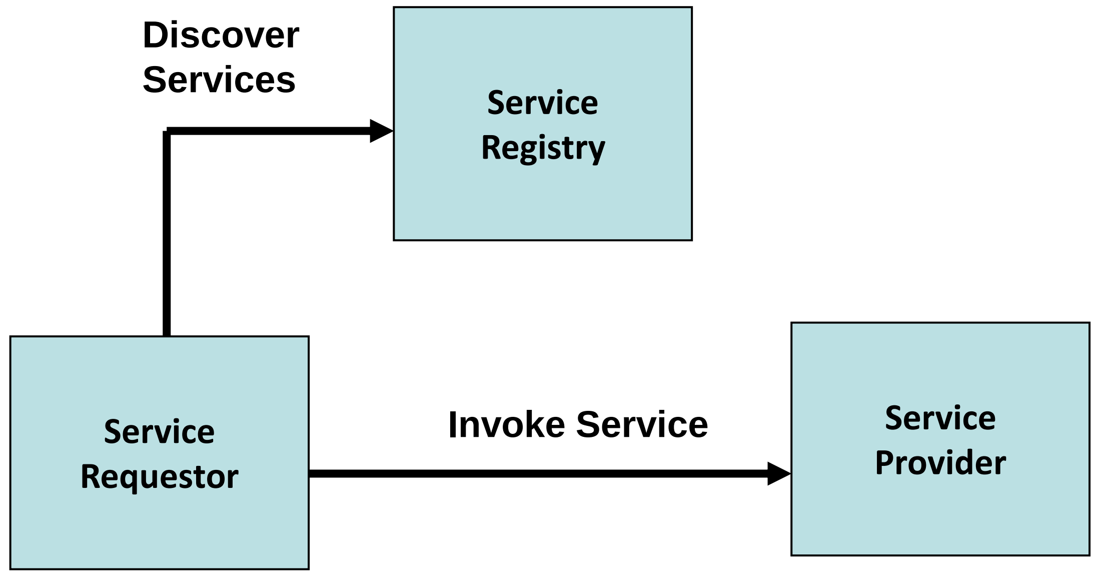
Discovery	UDDI
Description	WSDL
XML Messaging	XML-RPC, SOAP, Custom XML
Transport	HTTP, SMTP, FTP, BEEP



UDDI

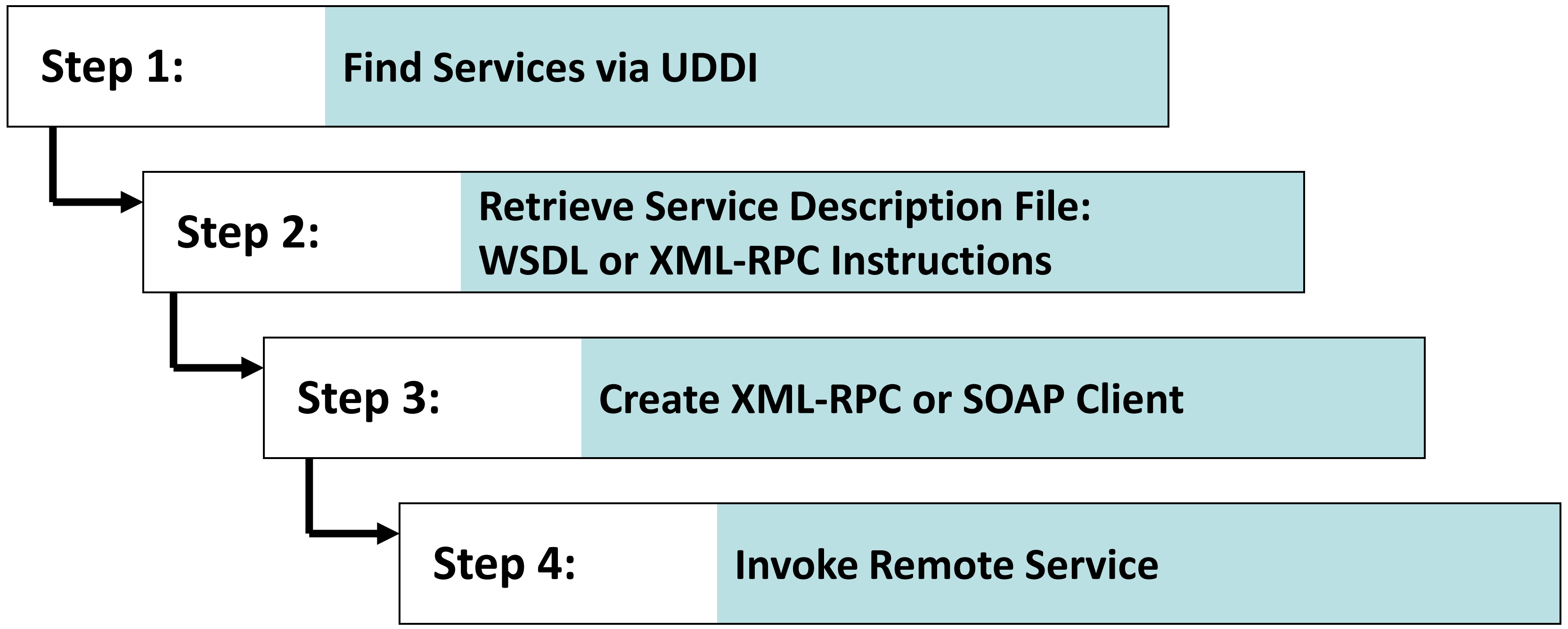
- UDDI: Universal Description, Discovery and Integration.
- Currently represents the *discovery* layer in the protocol stack.
- Originally created by Microsoft, IBM and Ariba.
- Technical specification for publishing and finding businesses and web services.

All Together Now!

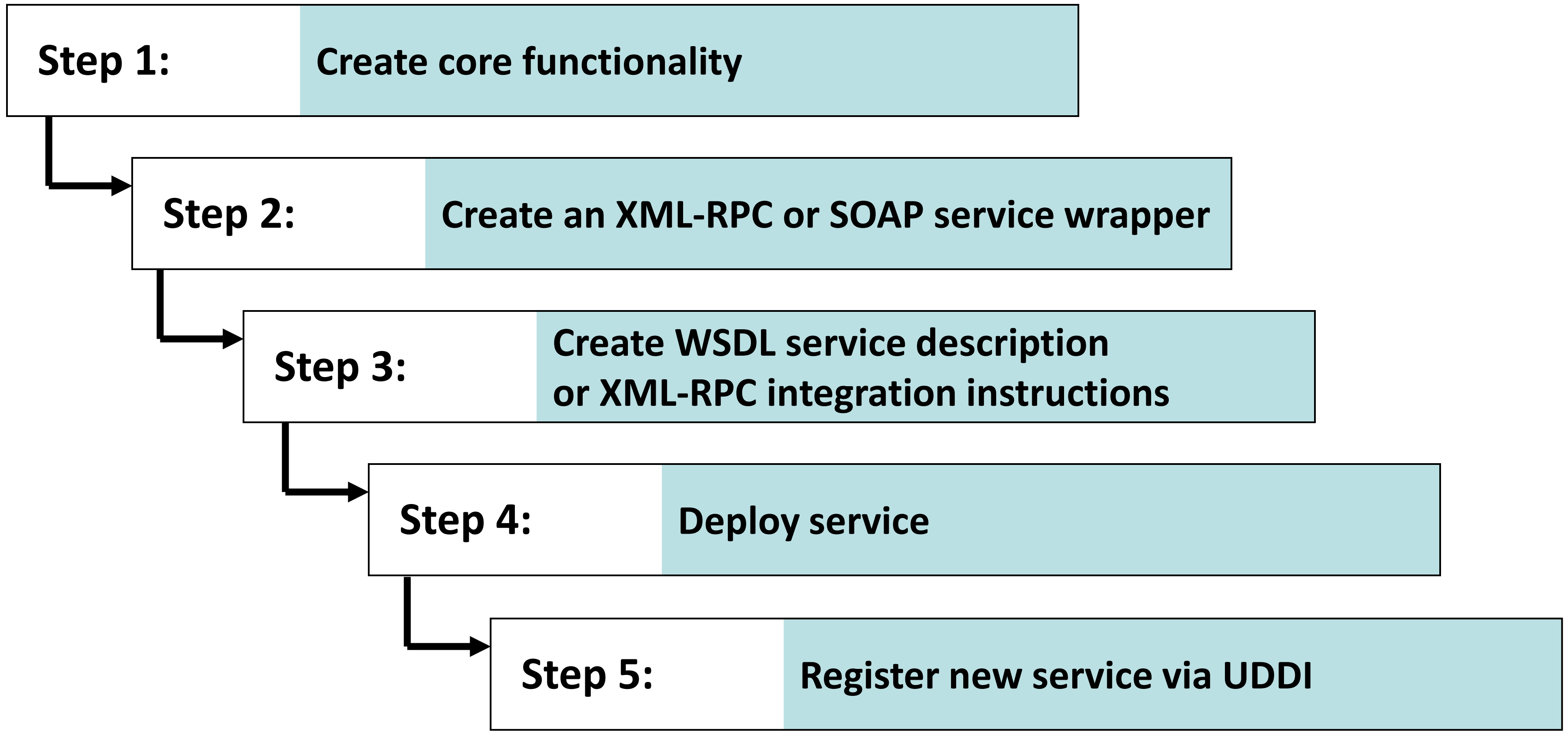


Discovery	UDDI
Description	WSDL
XML Messaging	XML-RPC, SOAP, Custom XML
Transport	HTTP, SMTP, FTP, BEEP

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

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

To review, please get ready to answer a multiple choice question

Only one answer will be correct, I will give you a moment, but try to answer as soon as you can.

Web Service Properties

- Available over the Internet or intranet
- Uses standardized XML messaging system
- Is not tied to any operating system or programming language
- Is self-describing via a common XML grammar
- Is discoverable via a simple find mechanism

Multiple Choice(single) Points: 1

Setting

In this module, we have learnt that a web service should be self-describing and discoverable. Which set of technologies can help to satisfy these 2 requirements? **(Only 1 answer is correct)**

A HTTP+XML

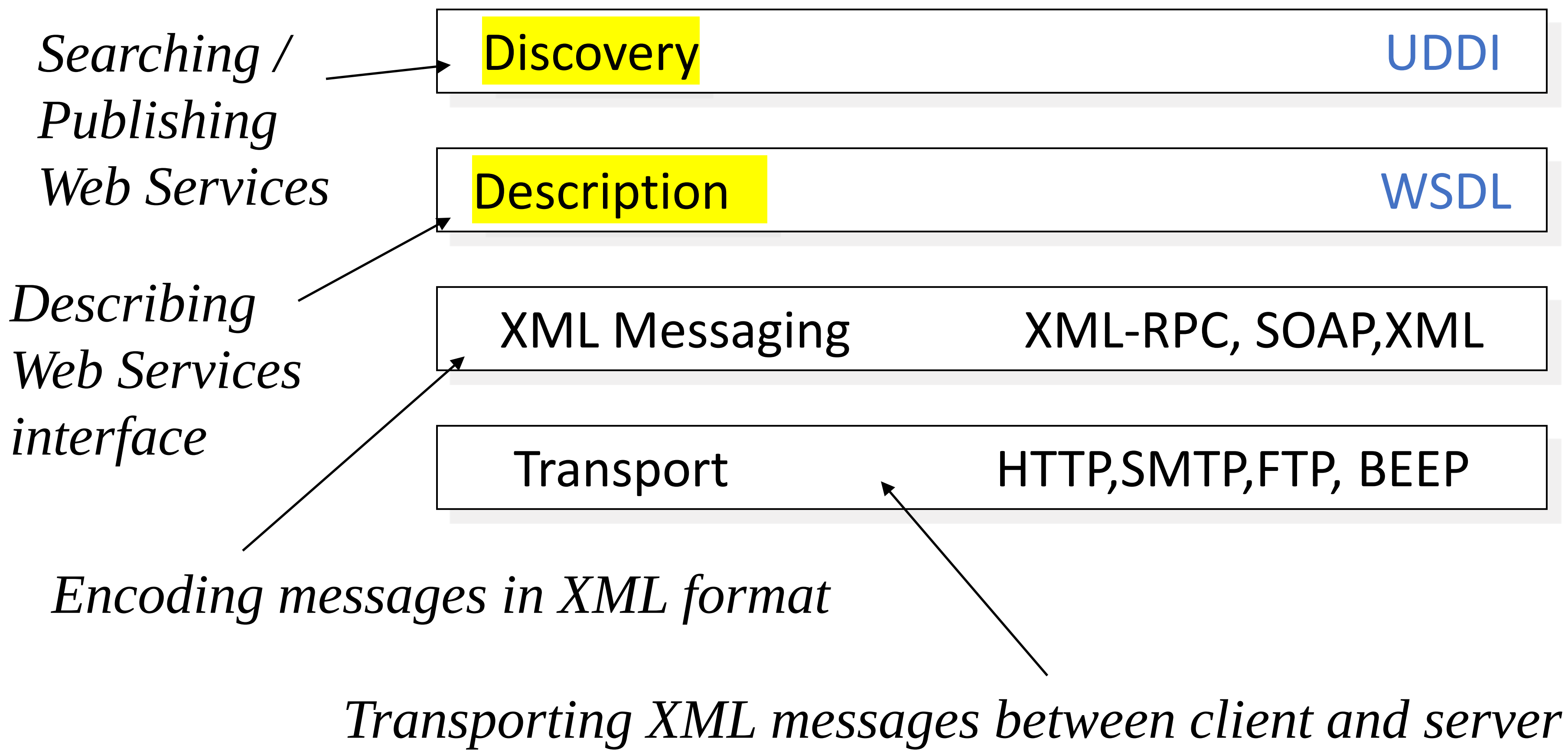
[✓ B] WSDL+UDDI

C SOAP+XML-RPC

D I do not know

Submit

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

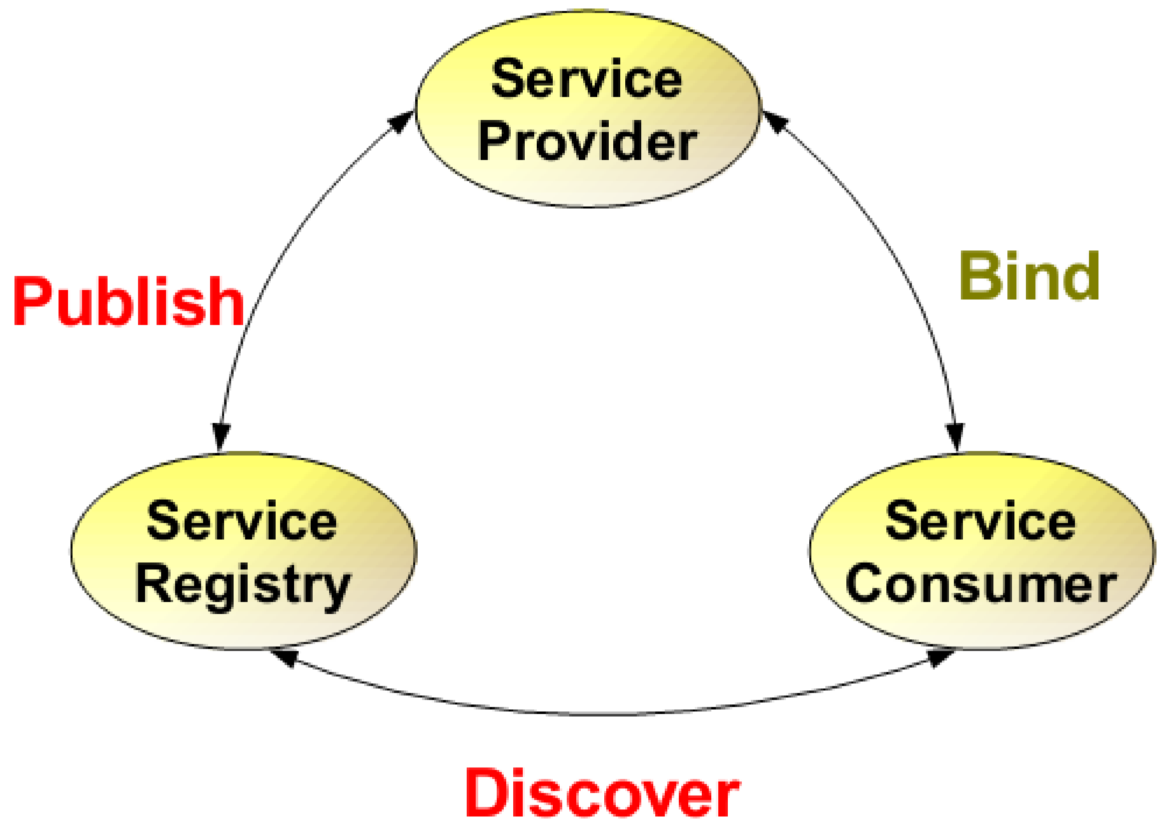
Web Service Architecture

- In this module, you have learnt about three major roles within the web service architecture:
  - Service provider
  - Service requestor / service consumer
  - Service registry
- Let’s examine them in more details

Let’s continue with our example

- Widget, Inc. sells parts through its website, enabling customers to submit purchase orders and check on order status.
- How do we call Widget Inc. in this scenario?
  - *Service provider*
- How do we call the customer of Widget Inc. in this scenario?
  - *Service consumer/requestor*

Web Service Architecture



How about service registry?

- The next activity will be an open answer question. Your answer should be very short, just a few sentences.

Open Question Points: 10

Setting

In our example, where

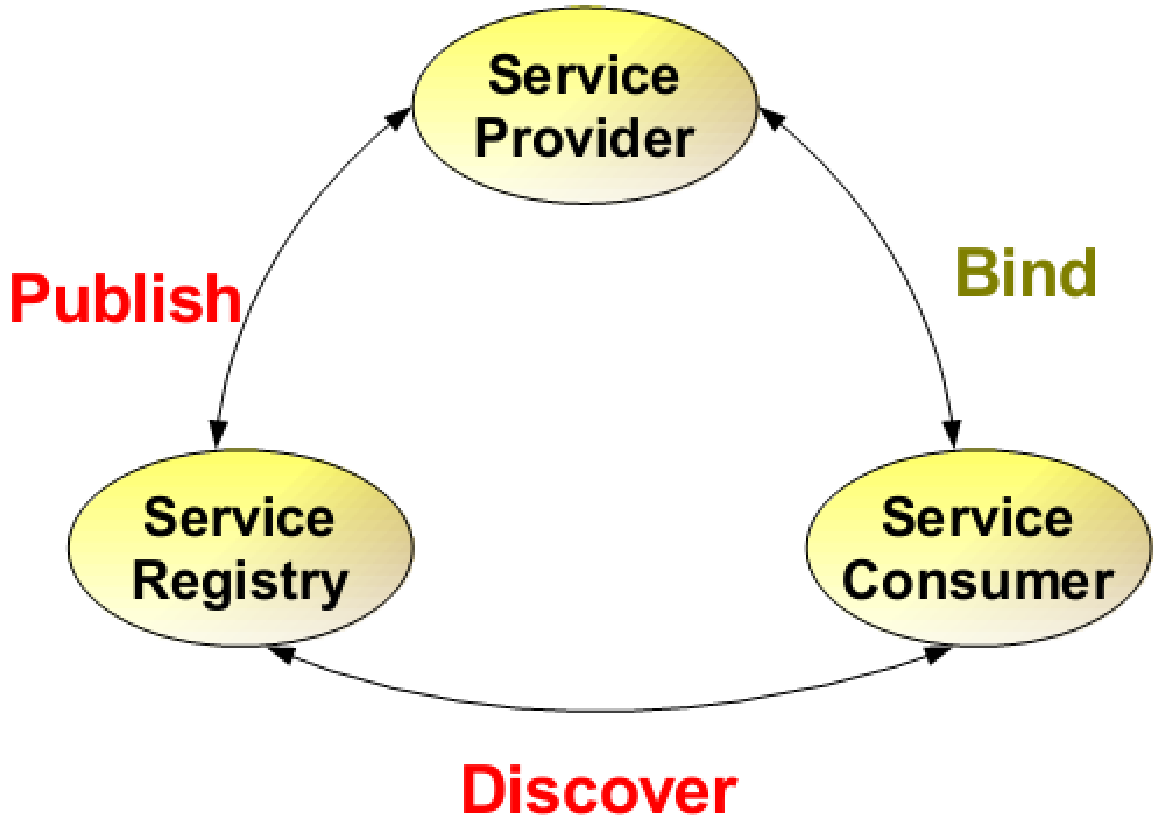
- Widget, Inc. is a service provider and sells parts through its website,
- customers of Widget, Inc. are service requestors and submit purchase orders and check on order status

How and when do you think would the registry be used by Widget, Inc. and their customers?

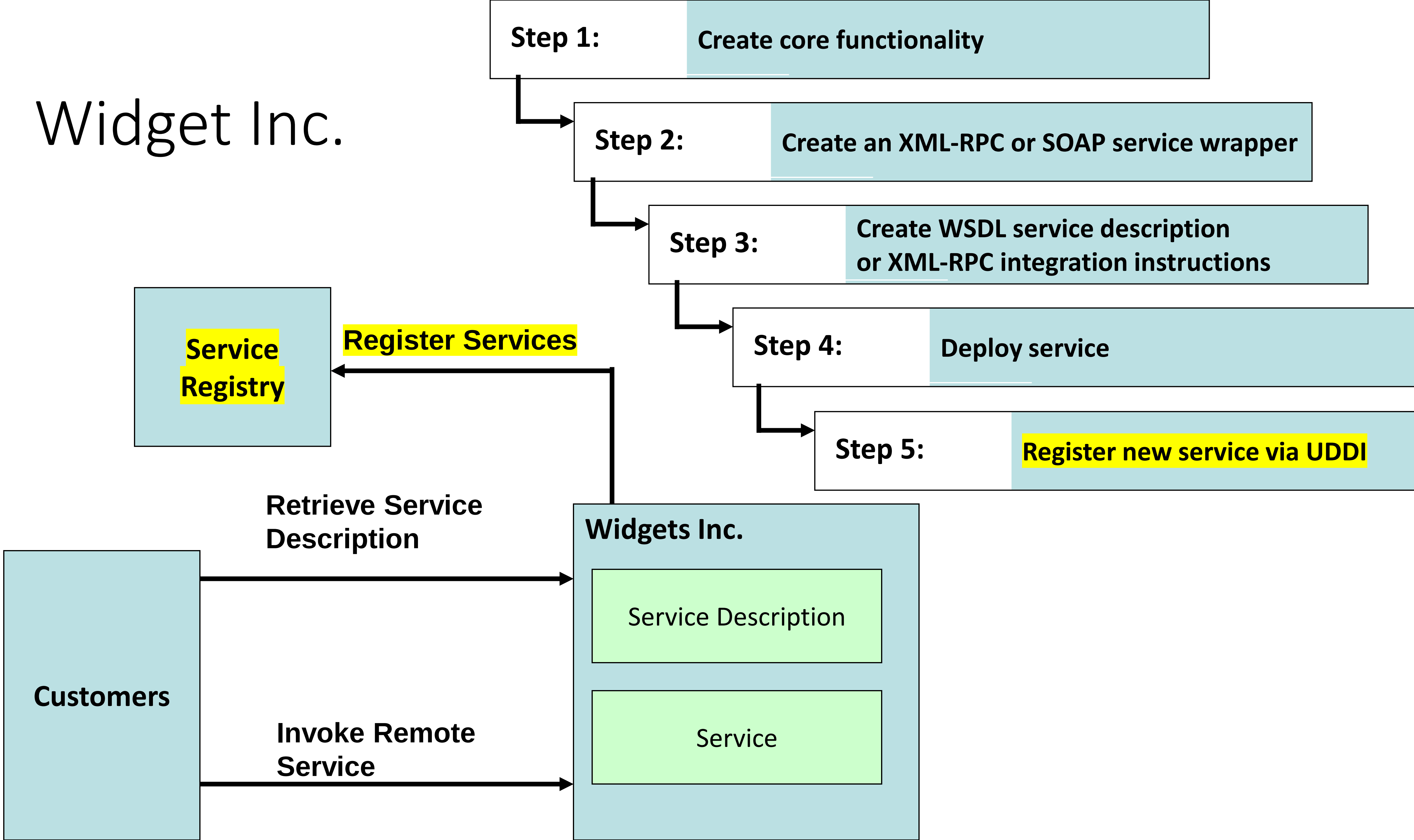
Open Question is only supported on Version 2.0 or newer.

Answer

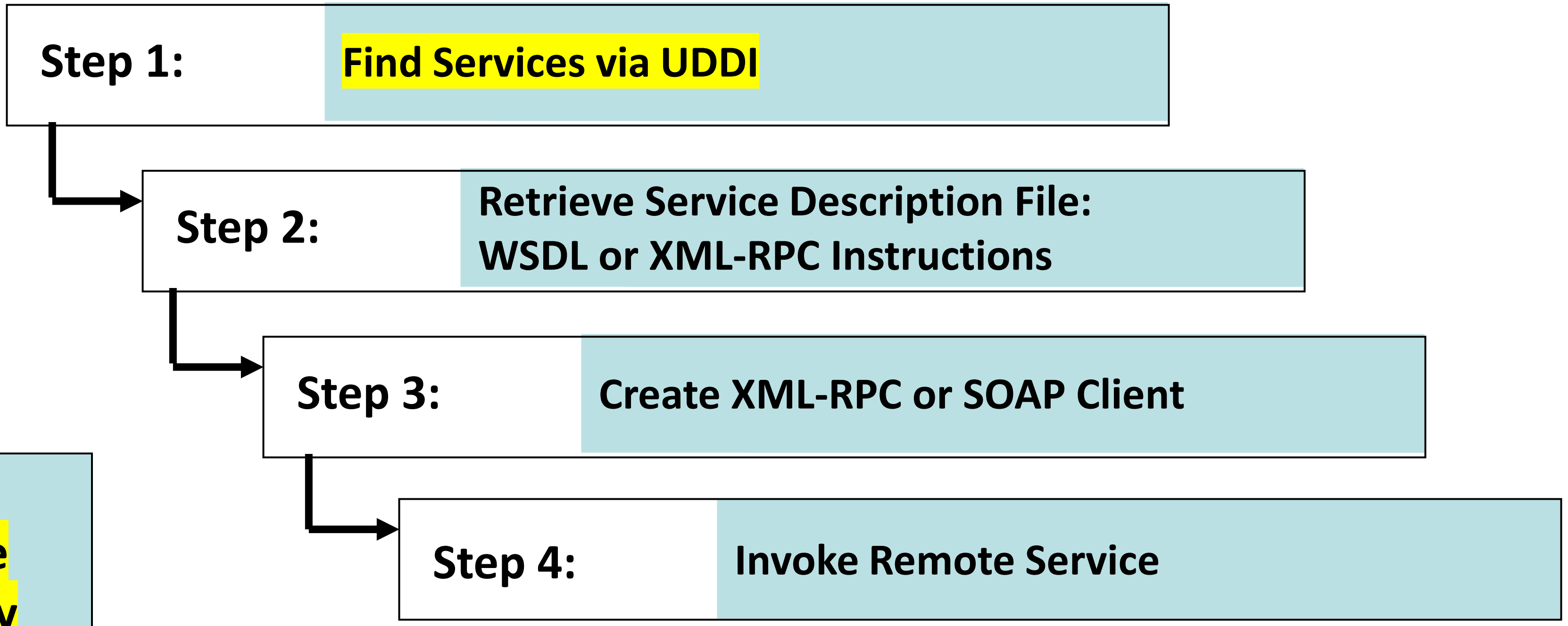
Web Service Architecture

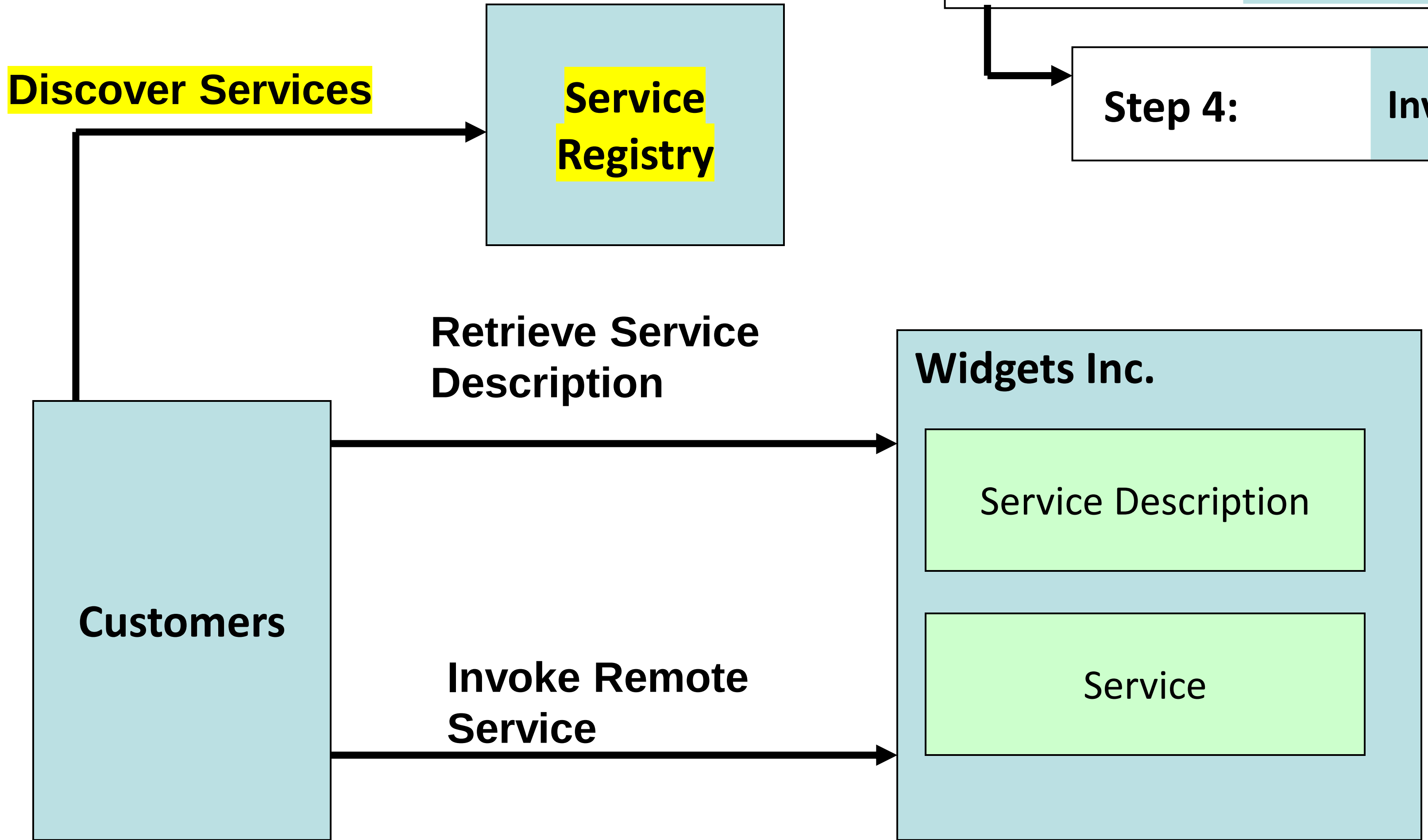


Widget Inc.


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
Service Registry
Register Services
Retrieve Service Description
Widgets Inc.
Service Description
Service
Customers
Invoke Remote Service

Customer


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


Discover Services
Service Registry
Retrieve Service Description
Widgets Inc.
Service Description
Service
Customers
Invoke Remote Service

Part III:
XML-RPC Essentials

XML-RPC

- XML-RPC provides an XML- and HTTP-based mechanism for making method or function calls across a network.

XML-RPC parts

- *XML-RPC data model*
  - A set of types for use in passing parameters, return values, and faults (error messages)
- *XML-RPC request structures*
  - An HTTP POST request containing method and parameter information
- *XML-RPC response structures*
  - An HTTP response that contains return values or fault information
- You can find more detailed information in PREP slides (Blackboard)

XML-RPC

- XML-RPC offers a very simple, but frequently useful, set of tools for connecting disparate systems and for publishing machine-readable information.

Disparate system

- System that was designed to operate as a fundamentally distinct
- Without exchanging data
- Without interacting with other system

Connecting disparate systems - why

- Important – networking, internet, building distributed systems
- Always a challenge in computing

Connecting disparate systems – how (1)

- XML – data format, not a protocol
  - Flexibility
  - Cross-platform usability
- XML over HTTP POST request
  - Sender assembles XML document and sends it much like HTML from data
  - Recipient processes the XML and sends back the response, also in XML
  - Developers need to create custom vocabularies for these transactions

Connecting disparate systems – how (2)

- Use standardized vocabularies
  - XML-RPC
    - A very simple protocol, uses XML messages travelling on HTTP to represent client-server remote procedure call (RPC)
    - XML messages identify methods, parameters, and the results for calling the methods
    - XML documents use simple but effective set of data types to pass information between computers
  - SOAP
    - Details in Module 2

Connecting disparate systems – how (3)

- REST
  - HTTP-based alternative
- BEEP
  - Uses XML to build protocols on TCP sockets
  - Supports HTTP-style message-and-reply
  - SOAP messages can be transmitted over BEEP

Open Question Points: 10

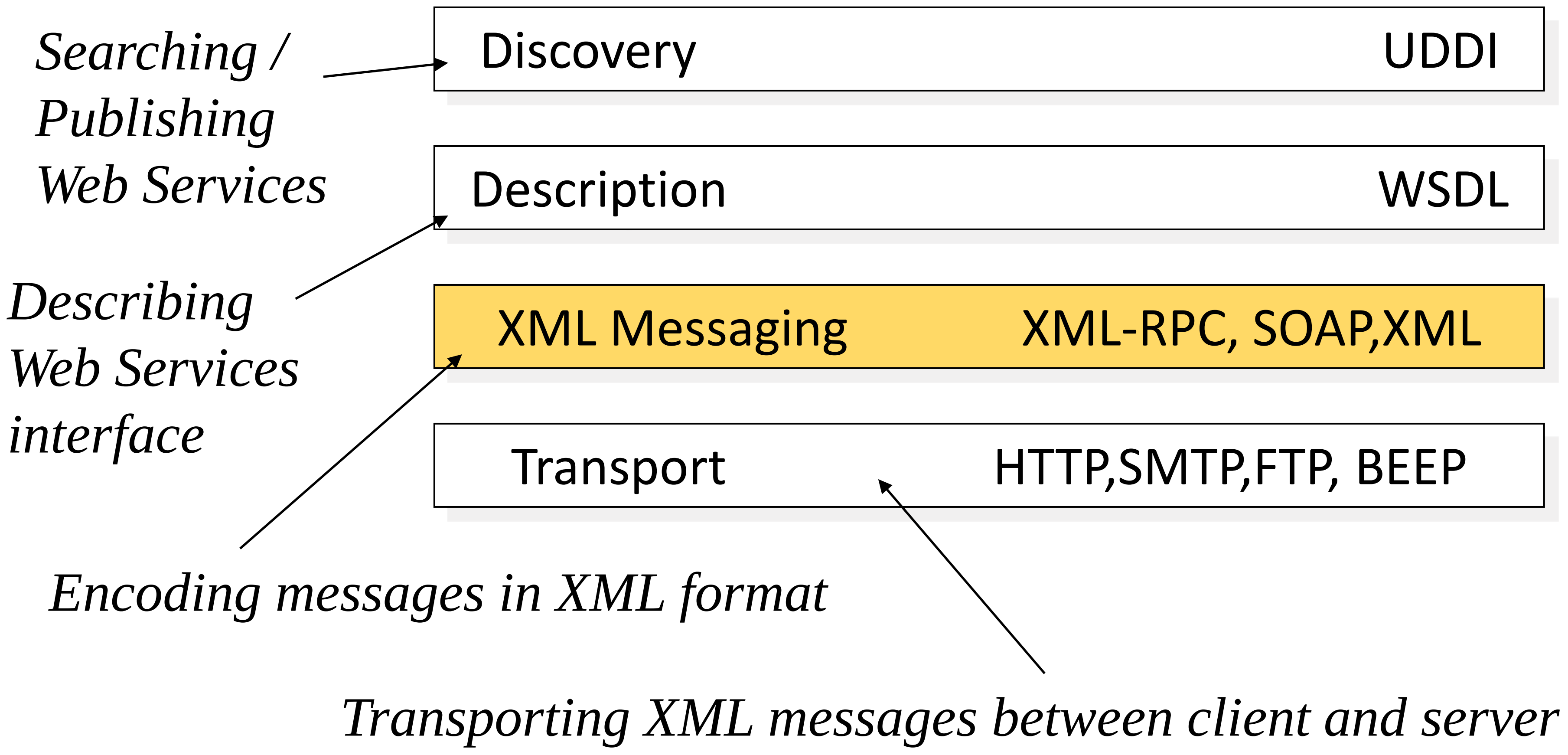
Setting

What is the relationship between XML, XML-RPC, and SOAP?

Open Question is only supported on Version 2.0 or newer.

Answer

What is the relationship between XML, XML-RPC, and SOAP?


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

Criticism of XML-RPC

- Critics argue that RPC calls can be made with plain XML instead
  - Both XML-RPC and XML require an application level data model (such as which filed names are defined in the XML schema or the parameter names in XML-RPC)
  - XML-RPC uses about 4 times the number of bytes compared to plain XML to encode the same objects
- Does XML-RPC add any value?

Open Question Points: 10

Setting

Does XML-RPC add any value?

Answer

Module 1 Summary

- Concept of “service oriented” and the background of service computing
- Basics of XML-RPC technology

Module Two: Service Message Exchange SOAP and Service Description WSDL

Our textbook for this module:

- **Ethan Cerami, Web Services Essentials, Publisher：O’Reilly，ISBN：9780596002244，**
  - Chapter 3 - SOAP
  - Chapter 6 - WSDL
- **Liang-Jie Zhang, Services Computing, Publisher: Springer，ISBN：9783540382812 -** You can find an online version of this book for free through our library webpage.
  - Chapter 3.1
  - Chapter 3.2 (without 3.2.5)

Module 2 Learning Outcomes

- Understand the basics of the SOAP protocol
- Understand the details about the SOAP XML Message specification
- Understand the SOAP encoding rules
- Understand the basics of WSDL

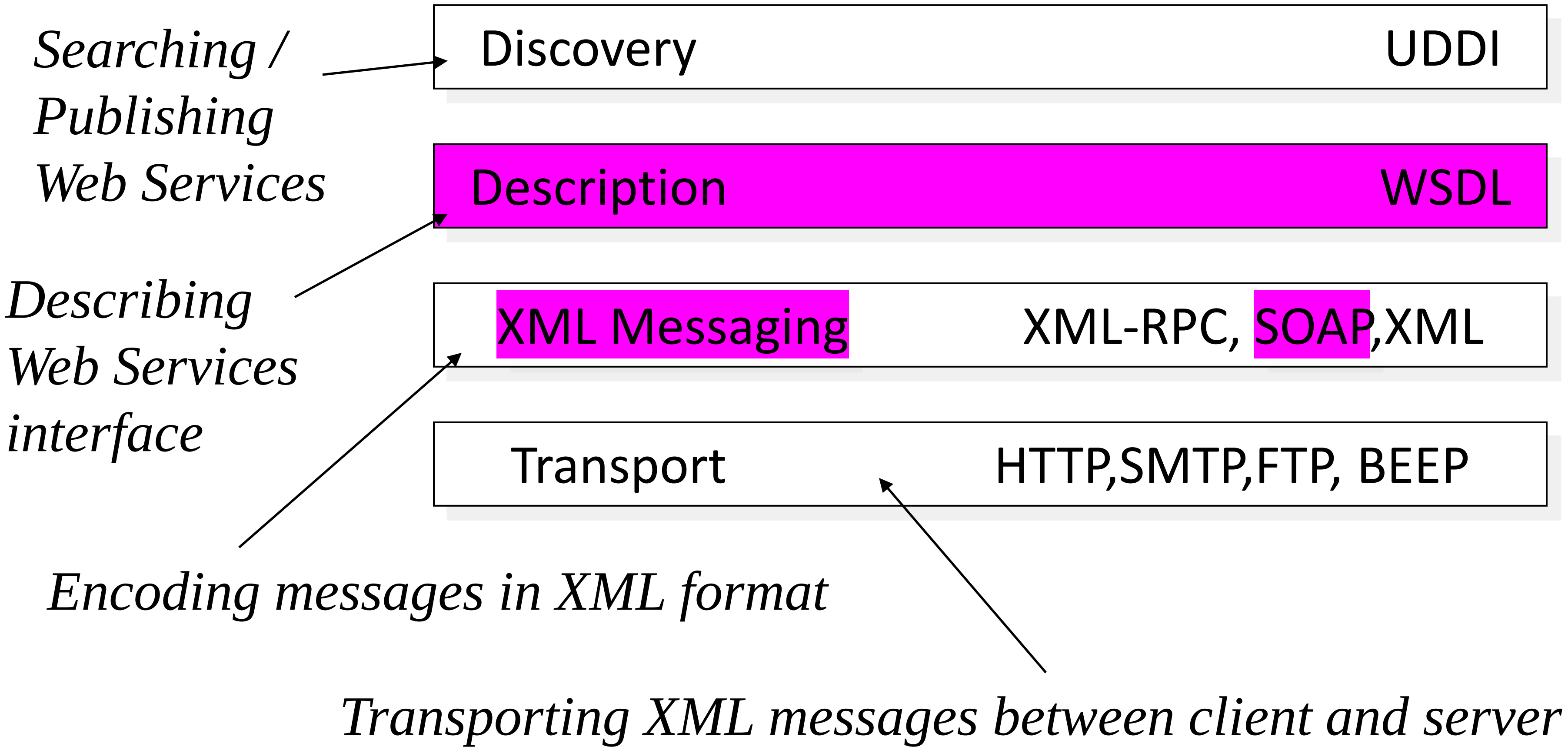
Guided questions for Module 2

- What is XML messaging?
- What is SOAP?
- What is the relationship between XML, XML-RPC, and SOAP?
- What is advantage of SOAP over CORBA, DCOM, and Java RMI?
- What platform and language do we need to use with SOAP? (tricky question)
- What are the major parts in SOAP specification?
- What are the main SOAP encoding rules?

Guided questions for Module 2

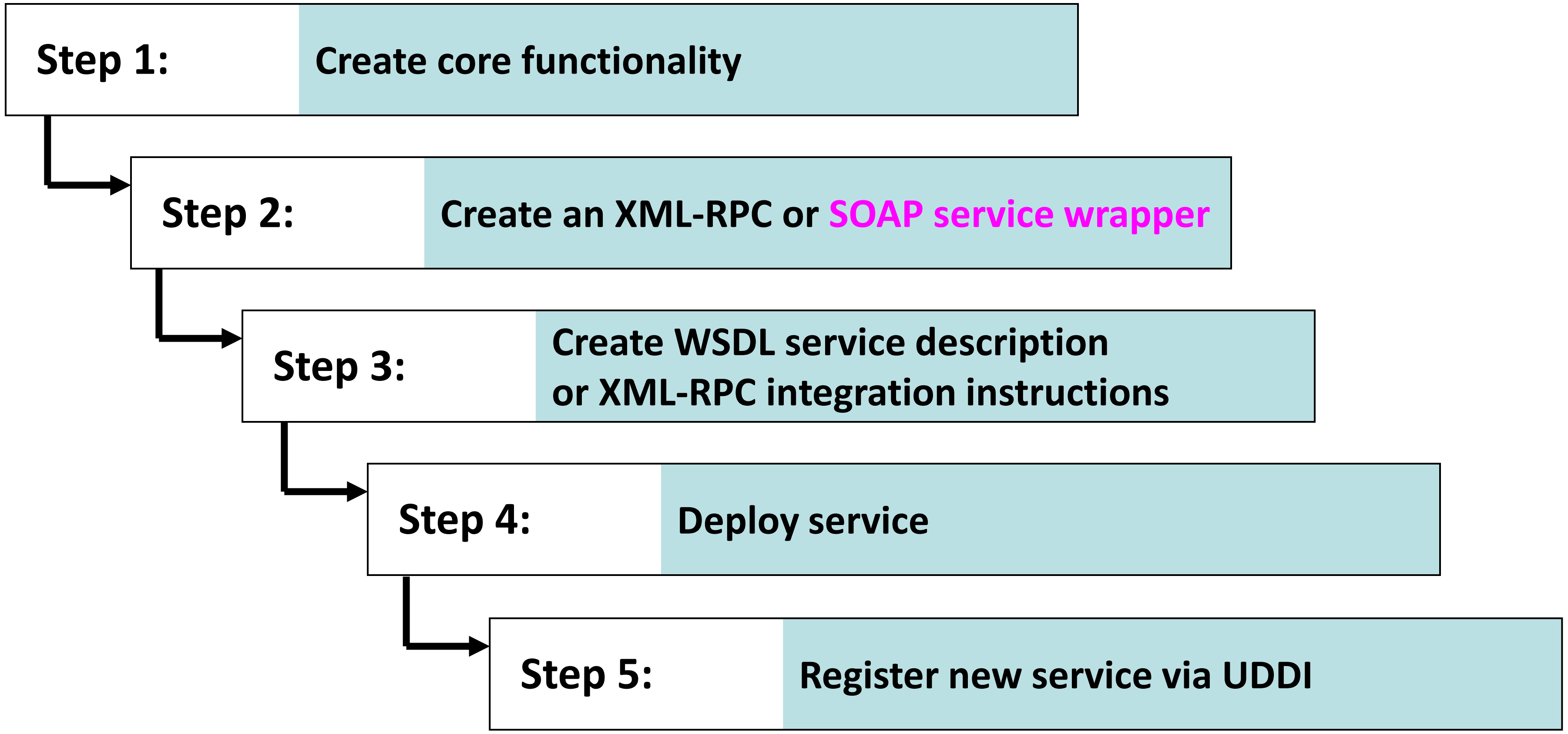
- What is WSDL?
- What is the relationship between WSDL and service description?
- What data does WSDL describe?
- What is WSDL used for?
- What platform and language do we need to use with WSDL? (tricky question)
- What are the major elements of WSDL?
- What is the relationship between SOAP, WSDL, and UDDI?

Web Service Protocol Stack


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

SOAP

Using the Protocols Together – service provider perspective



Step 1: Create core functionality

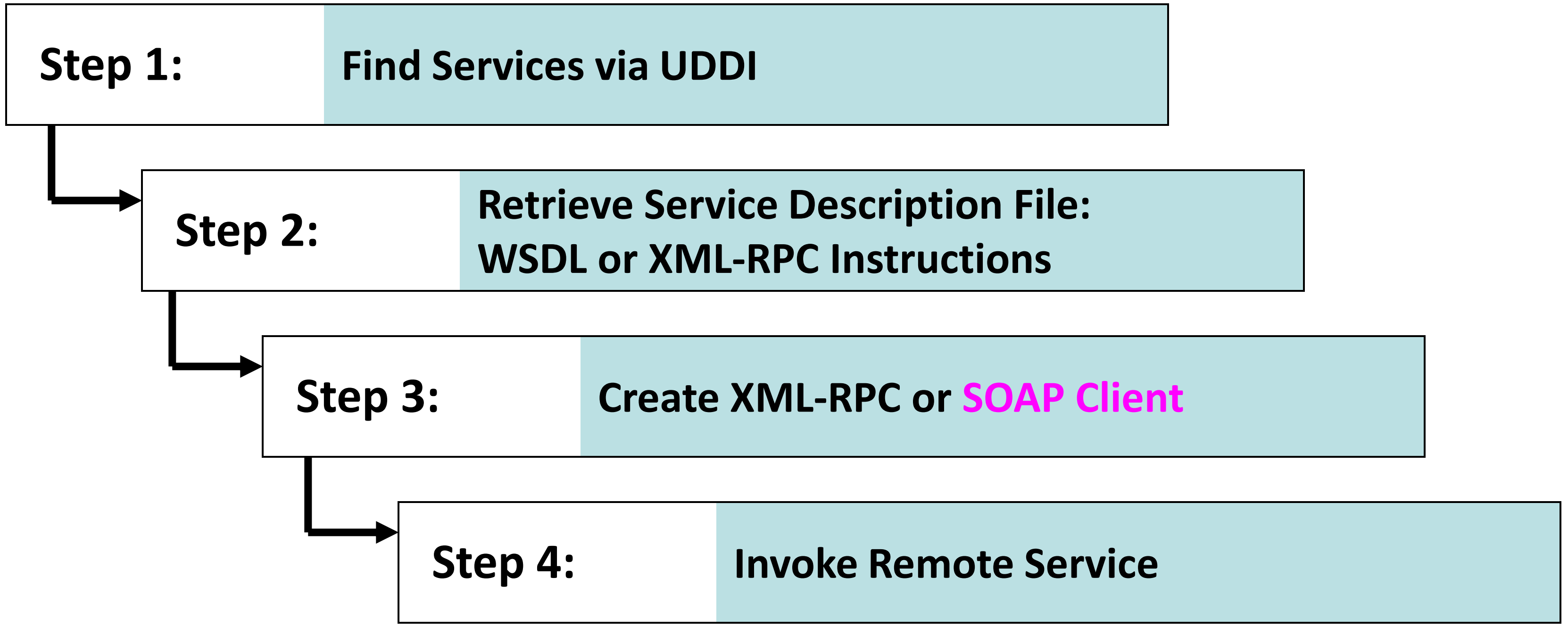
Step 2: Create an XML-RPC or SOAP service wrapper

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

A client program reads a WSDL document to understand what a Web service can do; then it uses SOAP to actually invoke the functions listed in the WSDL document.

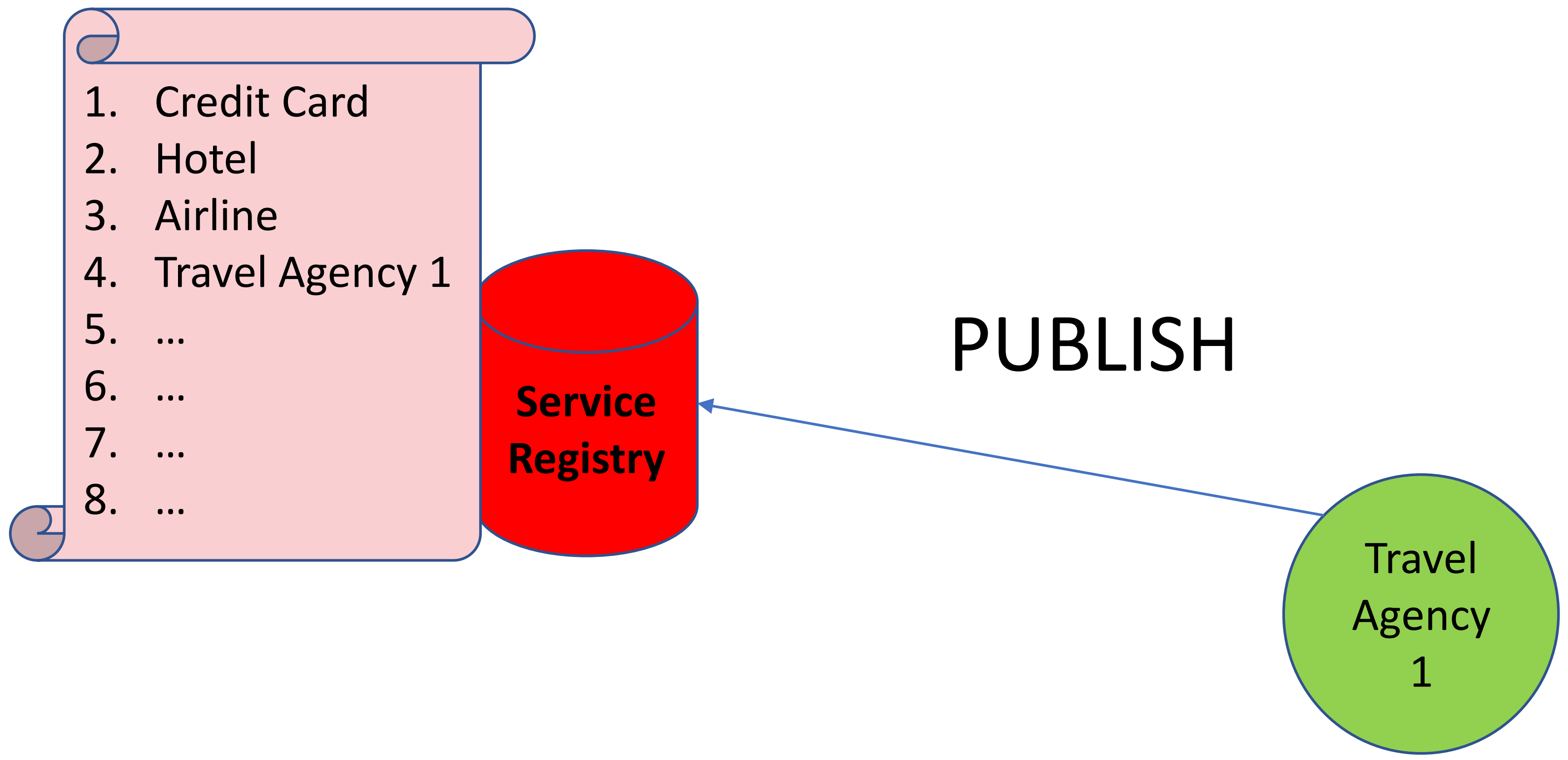
The SOAP specification

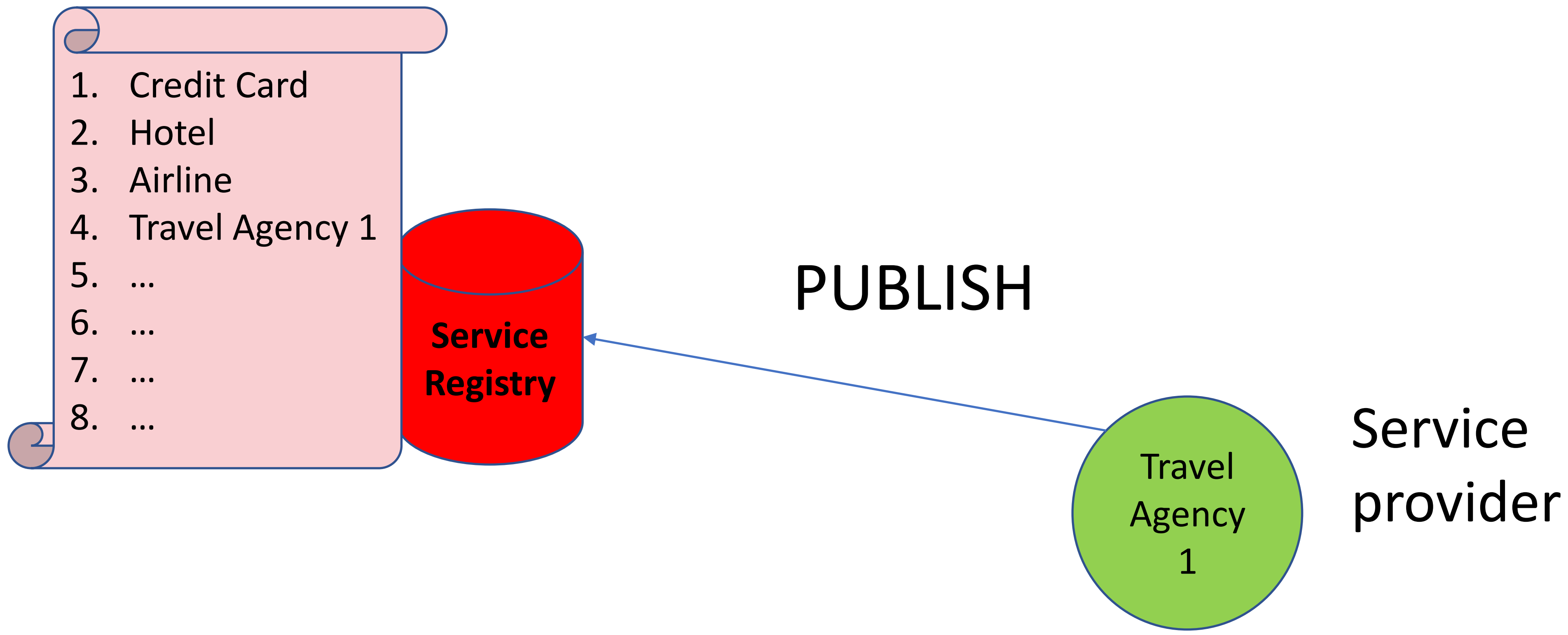
- The SOAP specification defines three major parts
  - SOAP envelope specification
  - Data encoding rules
  - RPC conventions

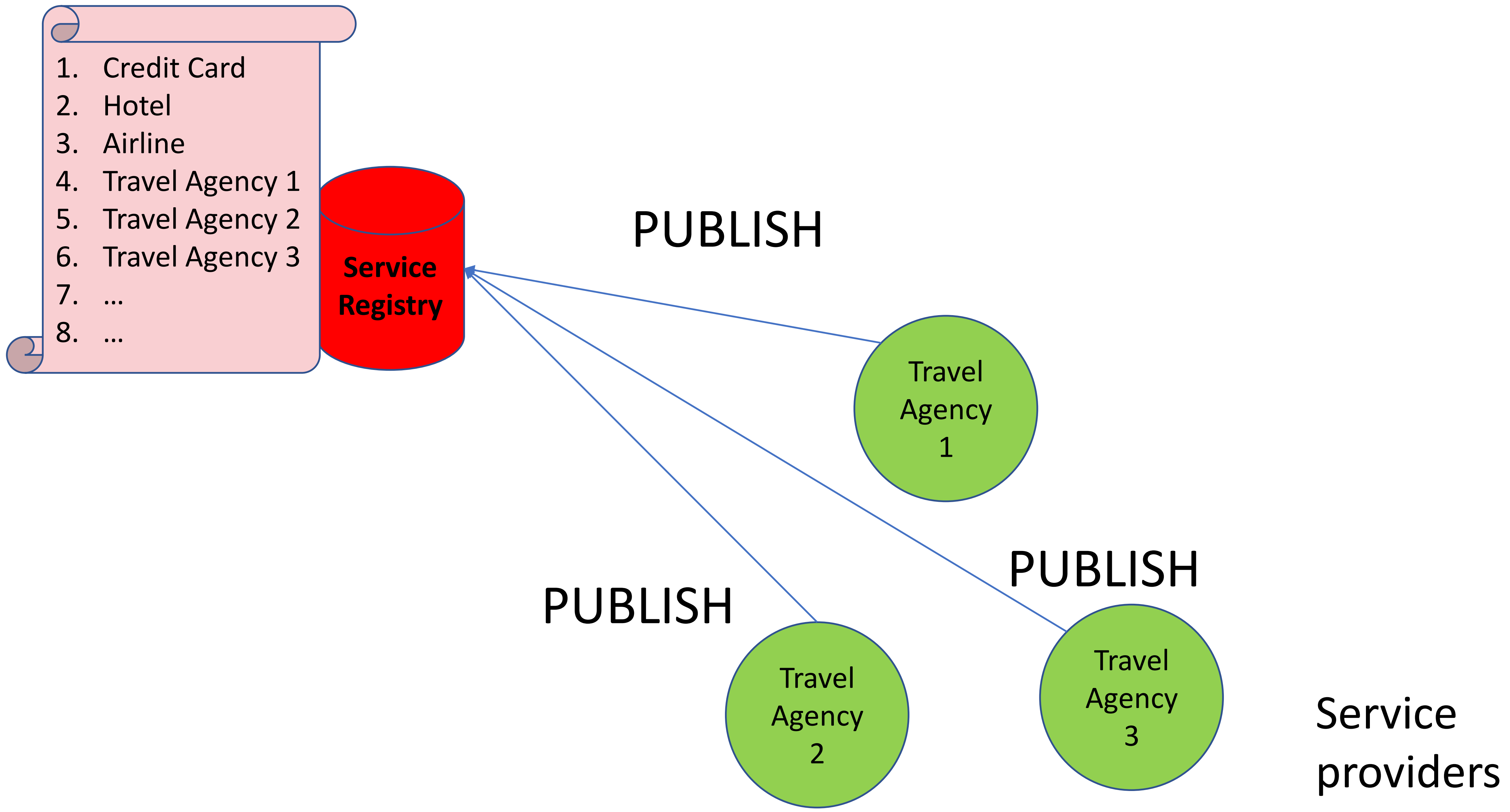
- For the details – check PREP slides

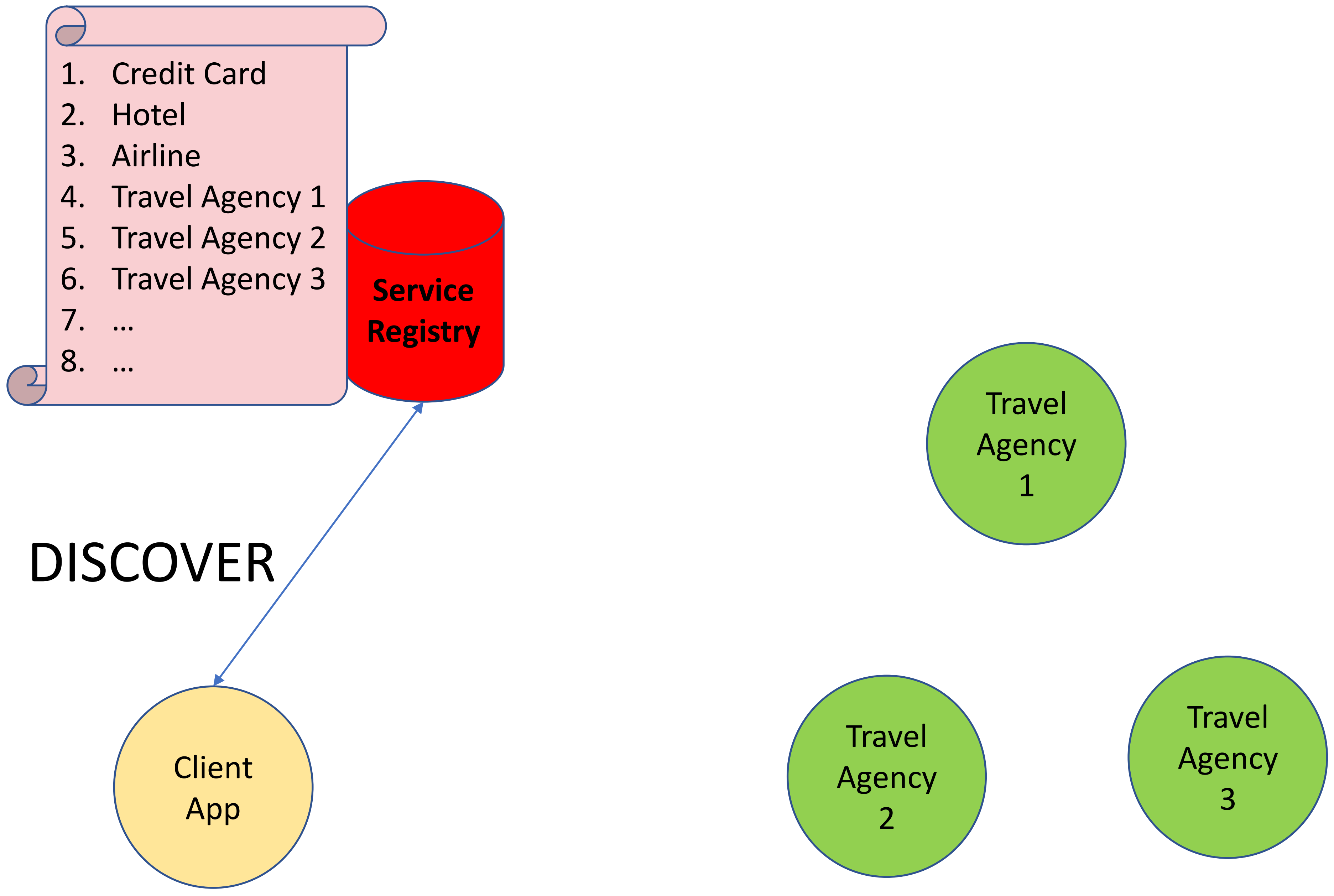
WSDL

Travel Agency 1


1. Credit Card
2. Hotel
3. Airline
4. Travel Agency 1
5. …
6. …
7. …
8. …
Service Registry
PUBLISH
Travel Agency 1


1. Credit Card
2. Hotel
3. Airline
4. Travel Agency 1
5. …
6. …
7. …
8. …
Service Registry
PUBLISH
Travel Agency 1
Service provider

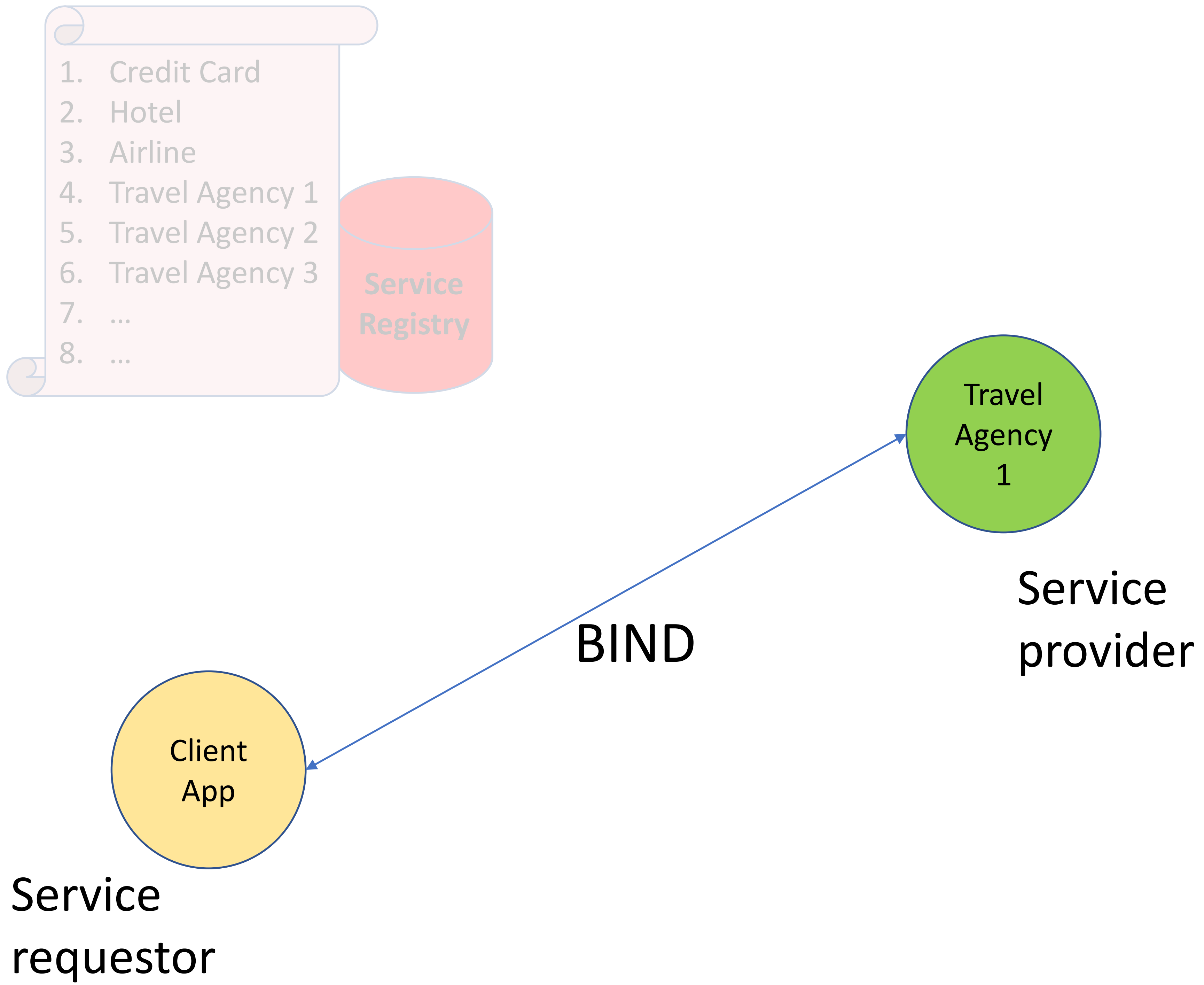

1. Credit Card
2. Hotel
3. Airline
4. Travel Agency 1
5. Travel Agency 2
6. Travel Agency 3
7. …
8. …
Service Registry
PUBLISH
Travel Agency 1
PUBLISH
PUBLISH
Travel Agency 2
Travel Agency 3
Service providers



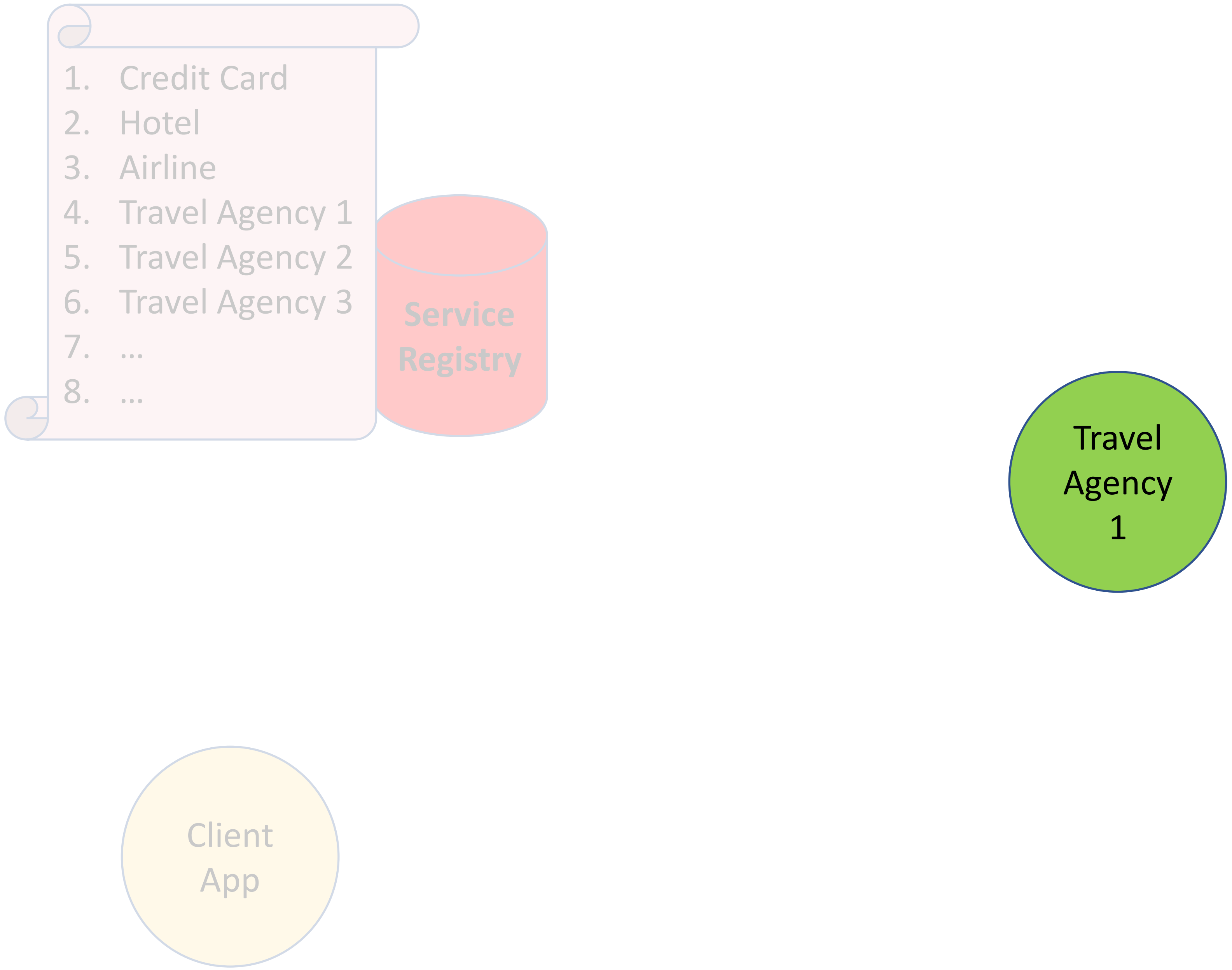
DISCOVER

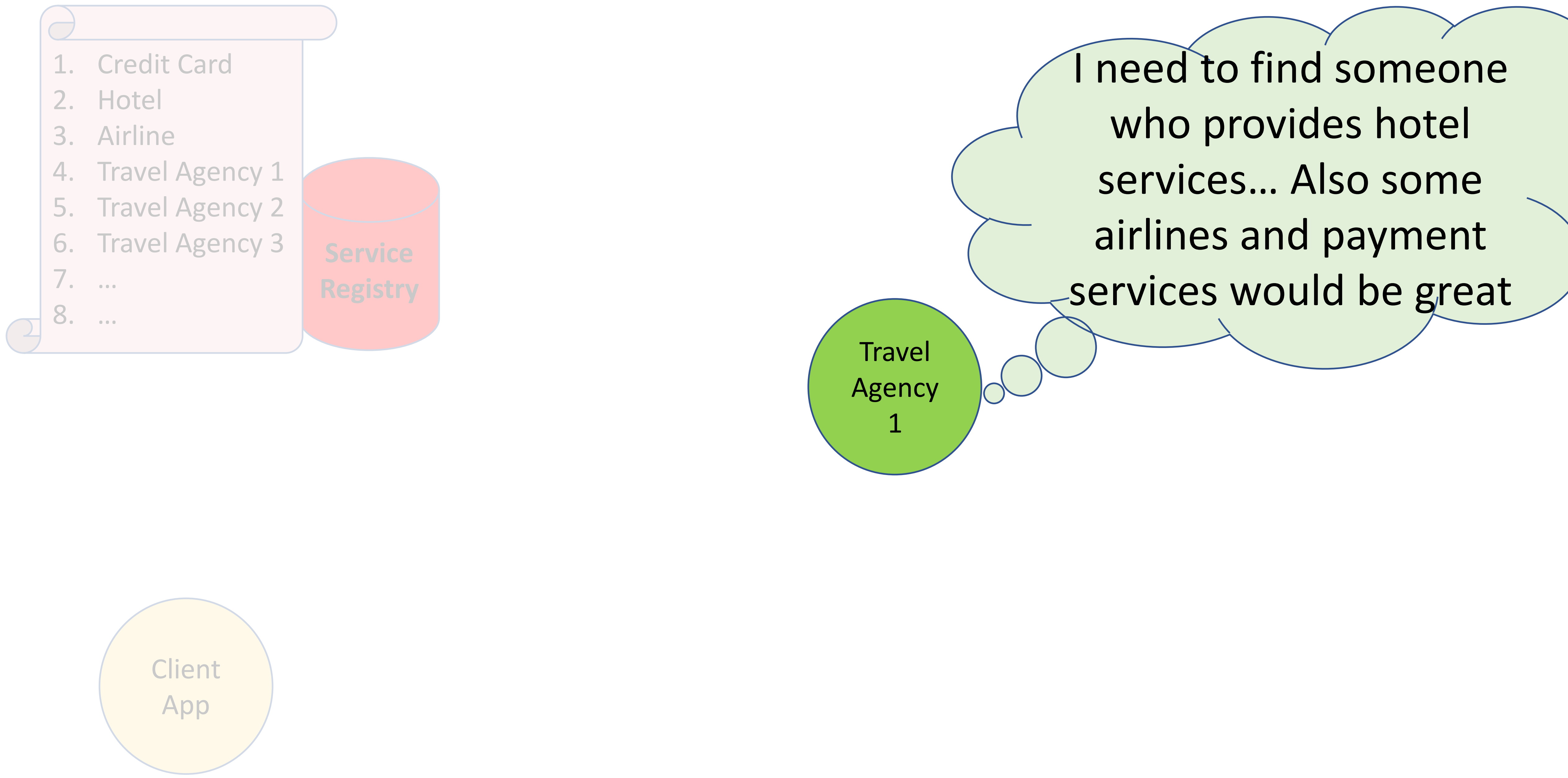
Service requestor

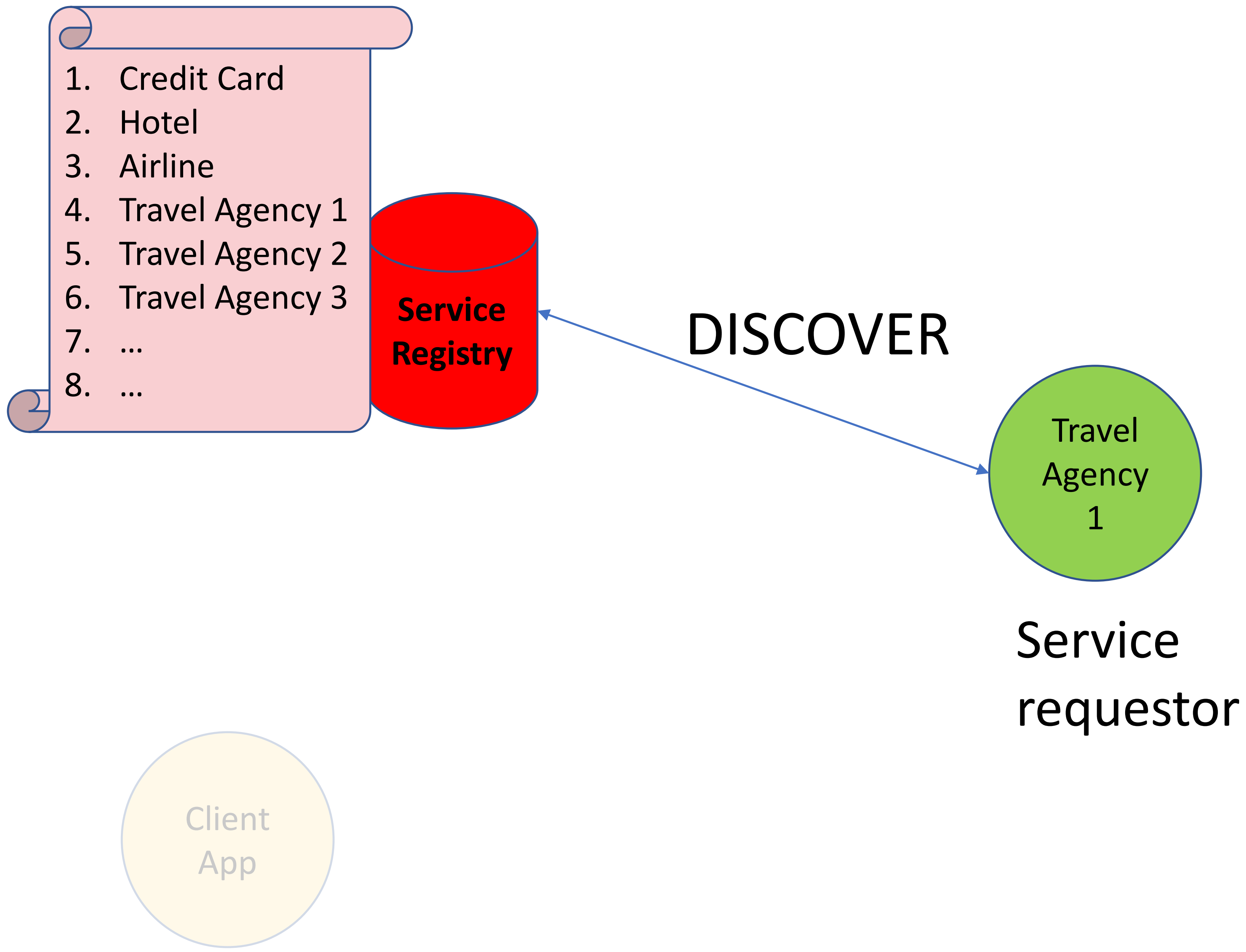
Service providers

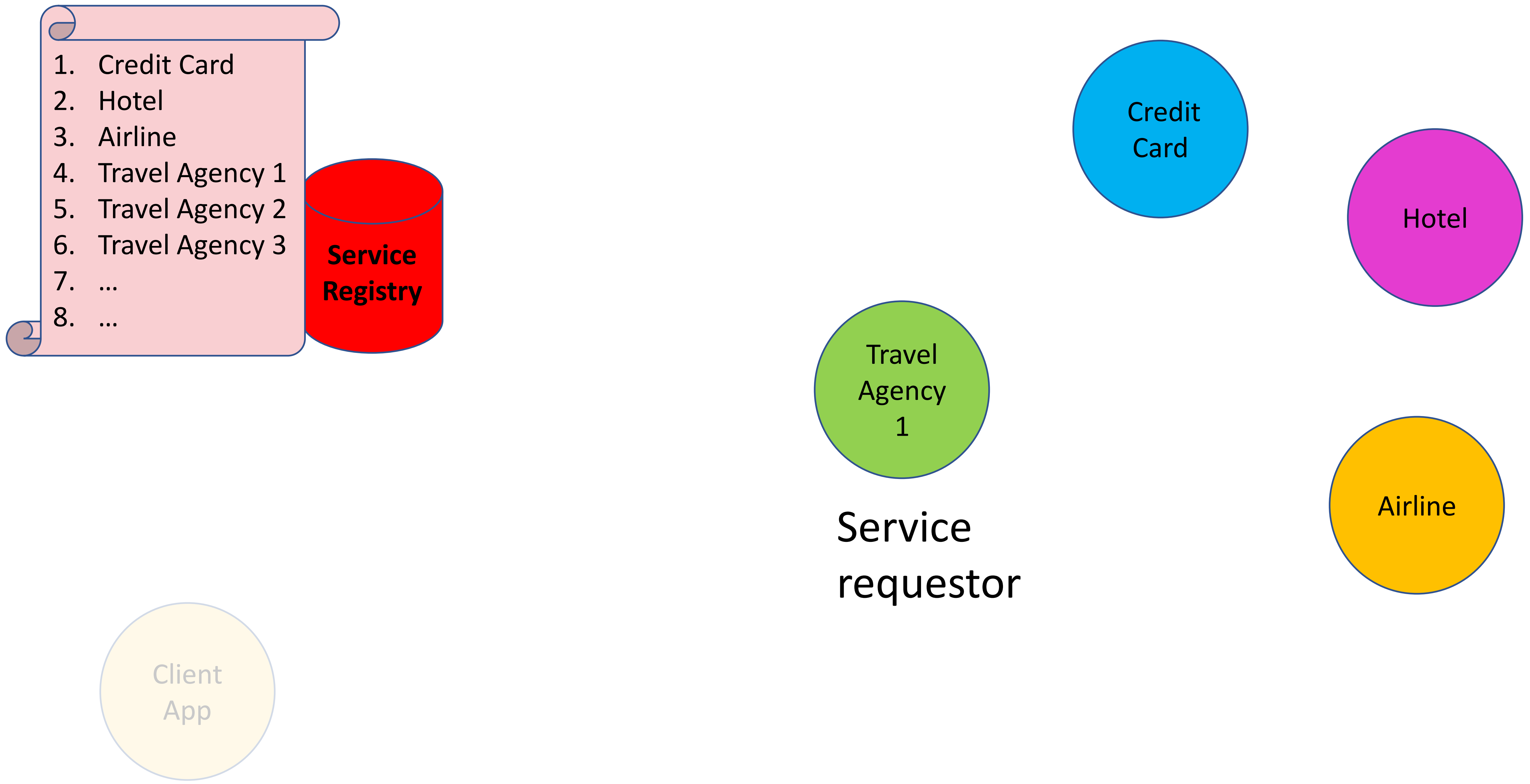


1. Credit Card
2. Hotel
3. Airline
4. Travel Agency 1
5. Travel Agency 2
6. Travel Agency 3
7. …
8. …
Service Registry
Travel Agency 1
Service provider
BIND
Client App
Service requestor

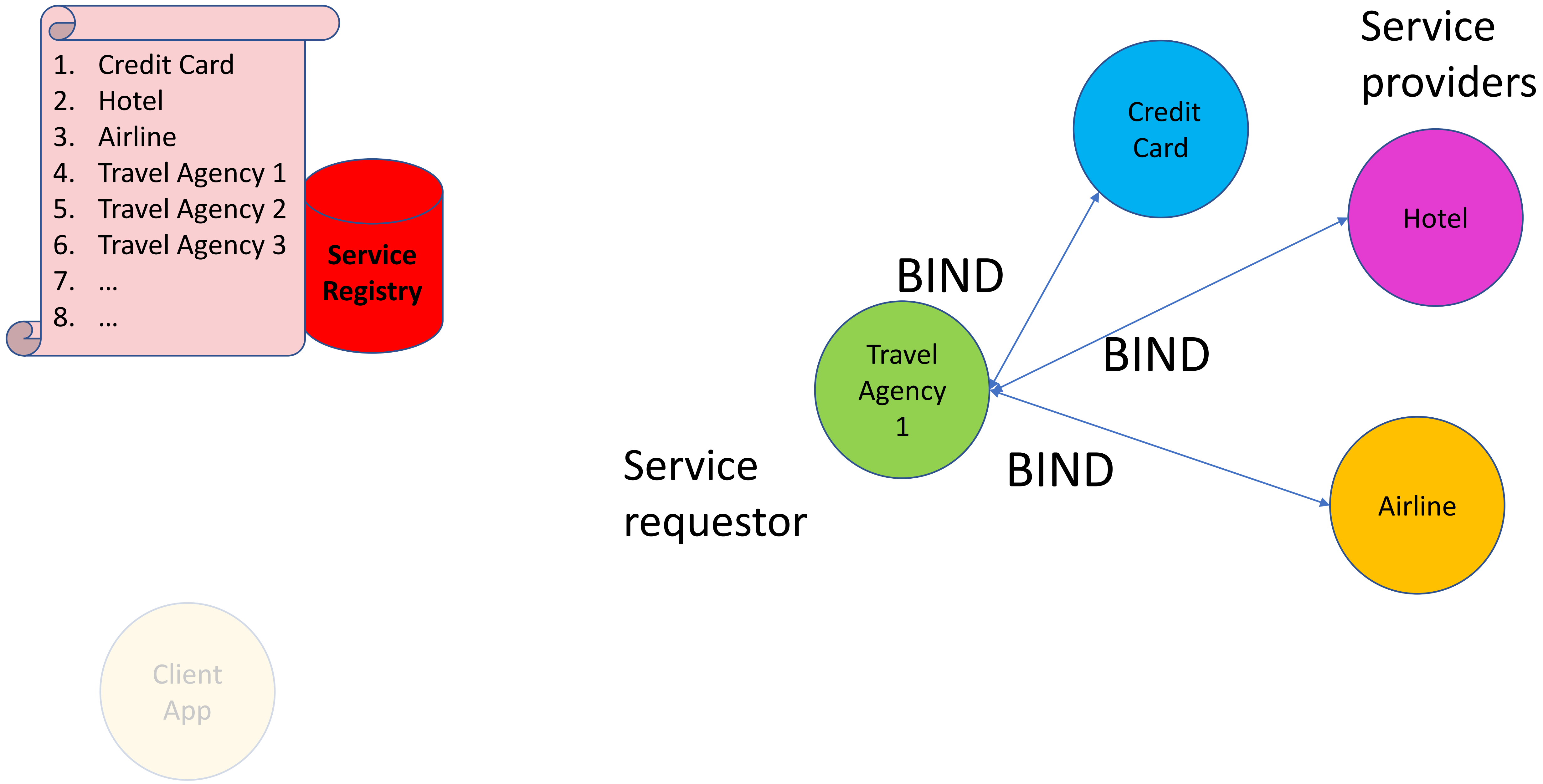

1. Credit Card
2. Hotel
3. Airline
4. Travel Agency 1
5. Travel Agency 2
6. Travel Agency 3
7. …
8. …
Service Registry
Travel Agency 1
Client App

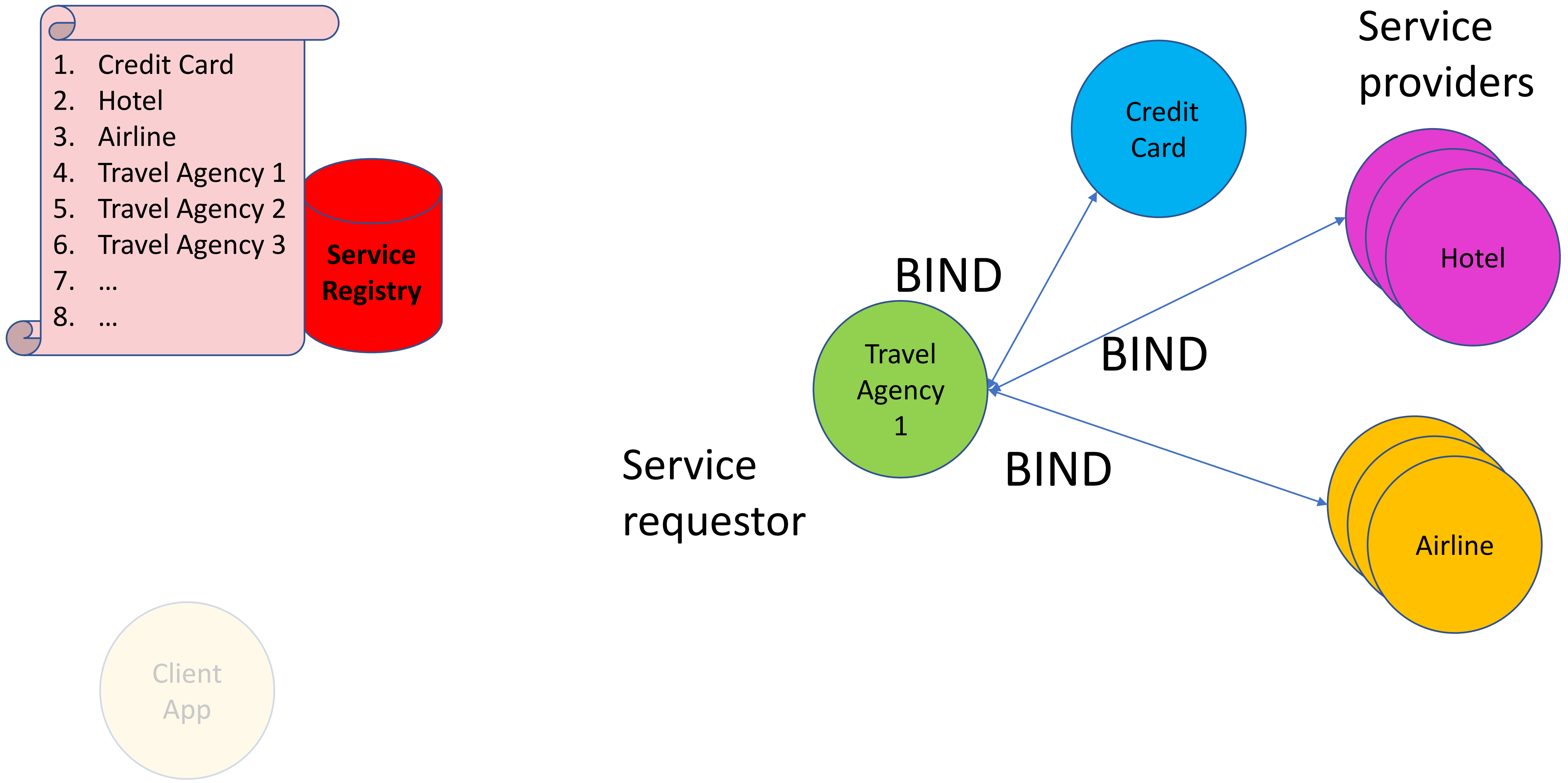

1. Credit Card
2. Hotel
3. Airline
4. Travel Agency 1
5. Travel Agency 2
6. Travel Agency 3
7. …
8. …
Service Registry
Client App
Travel Agency 1
I need to find someone who provides hotel services… Also some airlines and payment services would be great

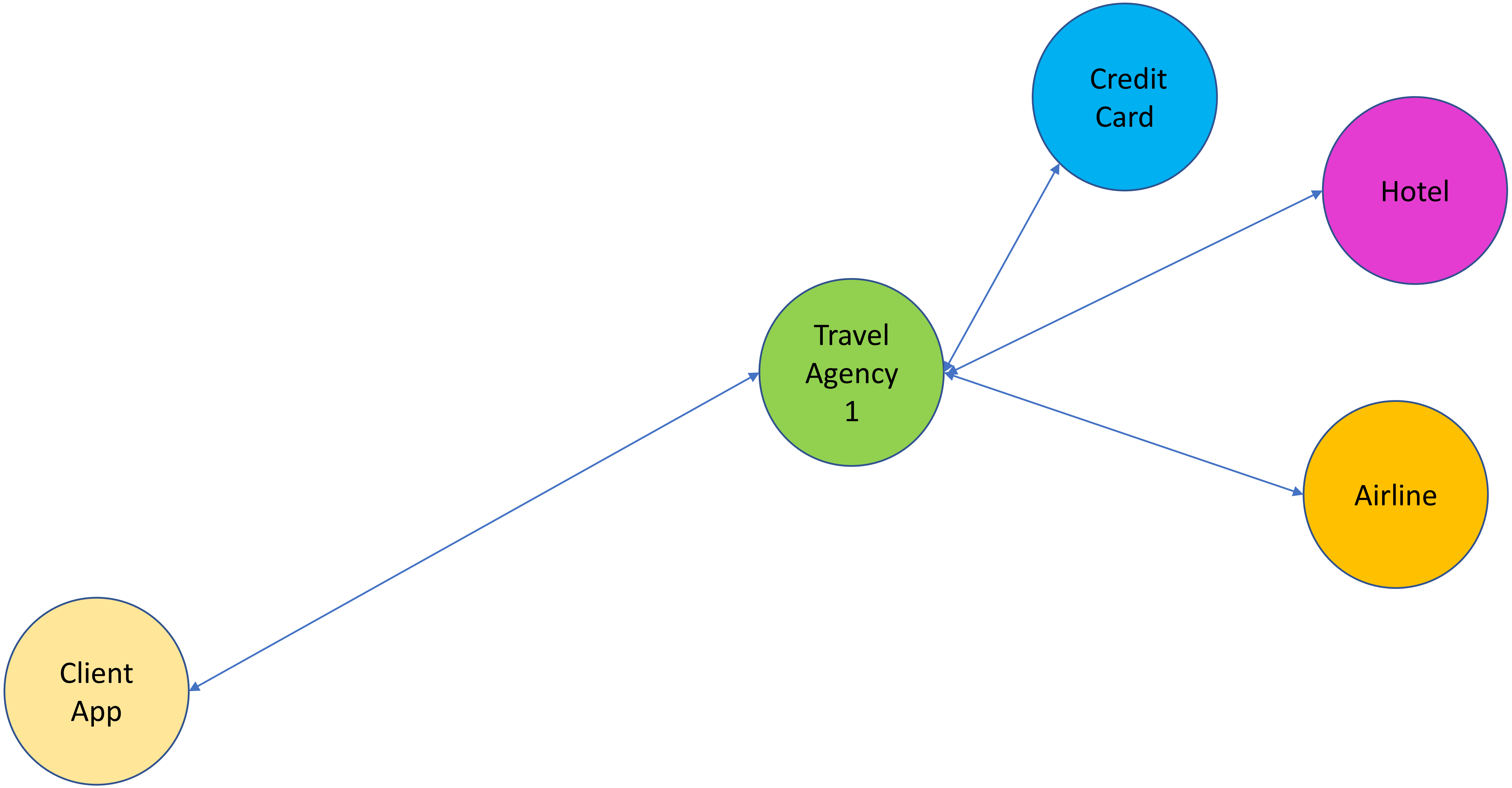

1. Credit Card
2. Hotel
3. Airline
4. Travel Agency 1
5. Travel Agency 2
6. Travel Agency 3
7. …
8. …
Service Registry
DISCOVER
Travel Agency 1
Service requestor
Client App



Service requestor

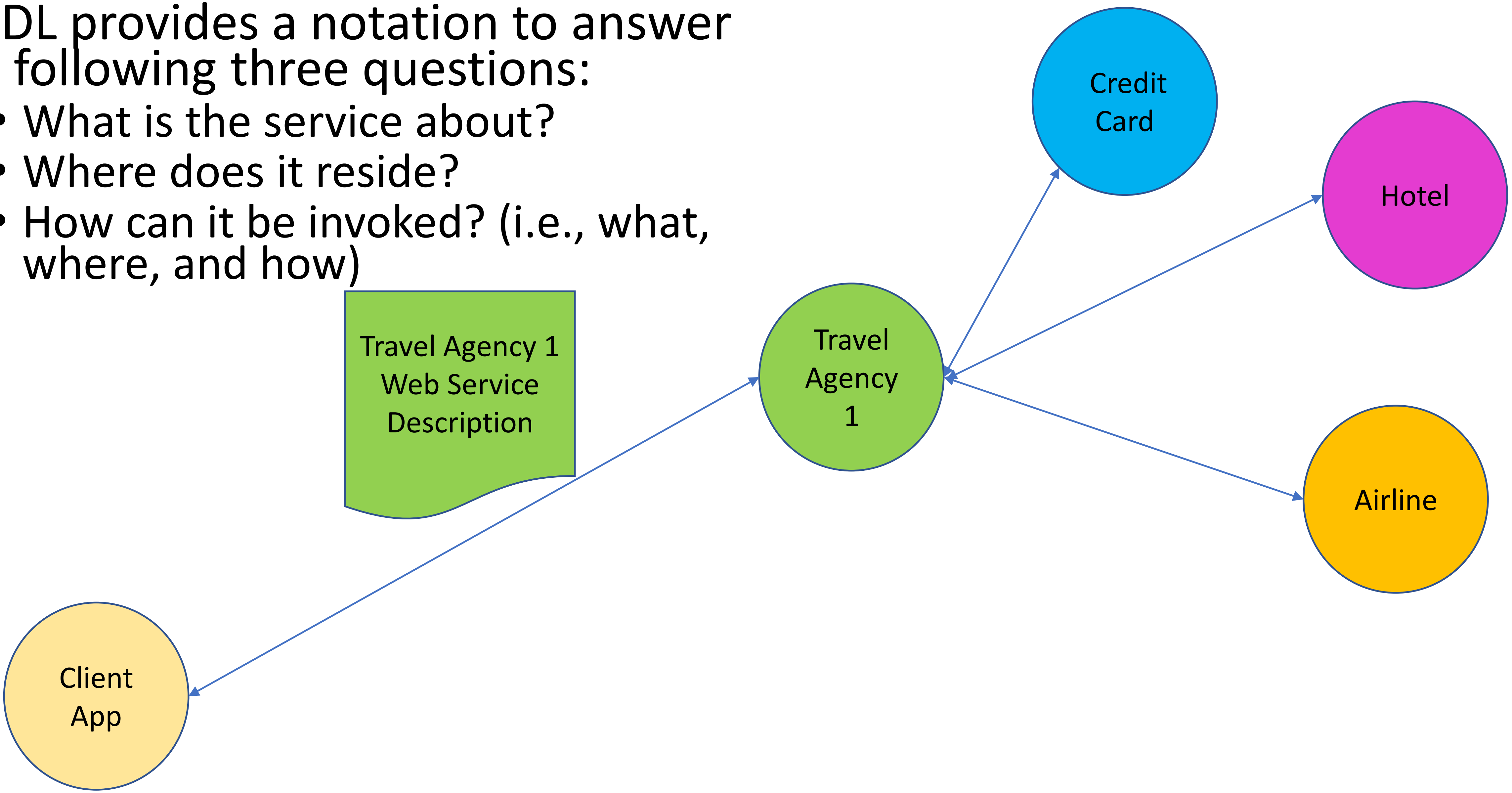

1. Credit Card
2. Hotel
3. Airline
4. Travel Agency 1
5. Travel Agency 2
6. Travel Agency 3
7. …
8. …
Service Registry
Service providers
Credit Card
Hotel
Airline
BIND
BIND
BIND
Travel Agency 1
Service requestor
Client App


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


Credit Card
Hotel
Travel Agency 1
Airline
Client App

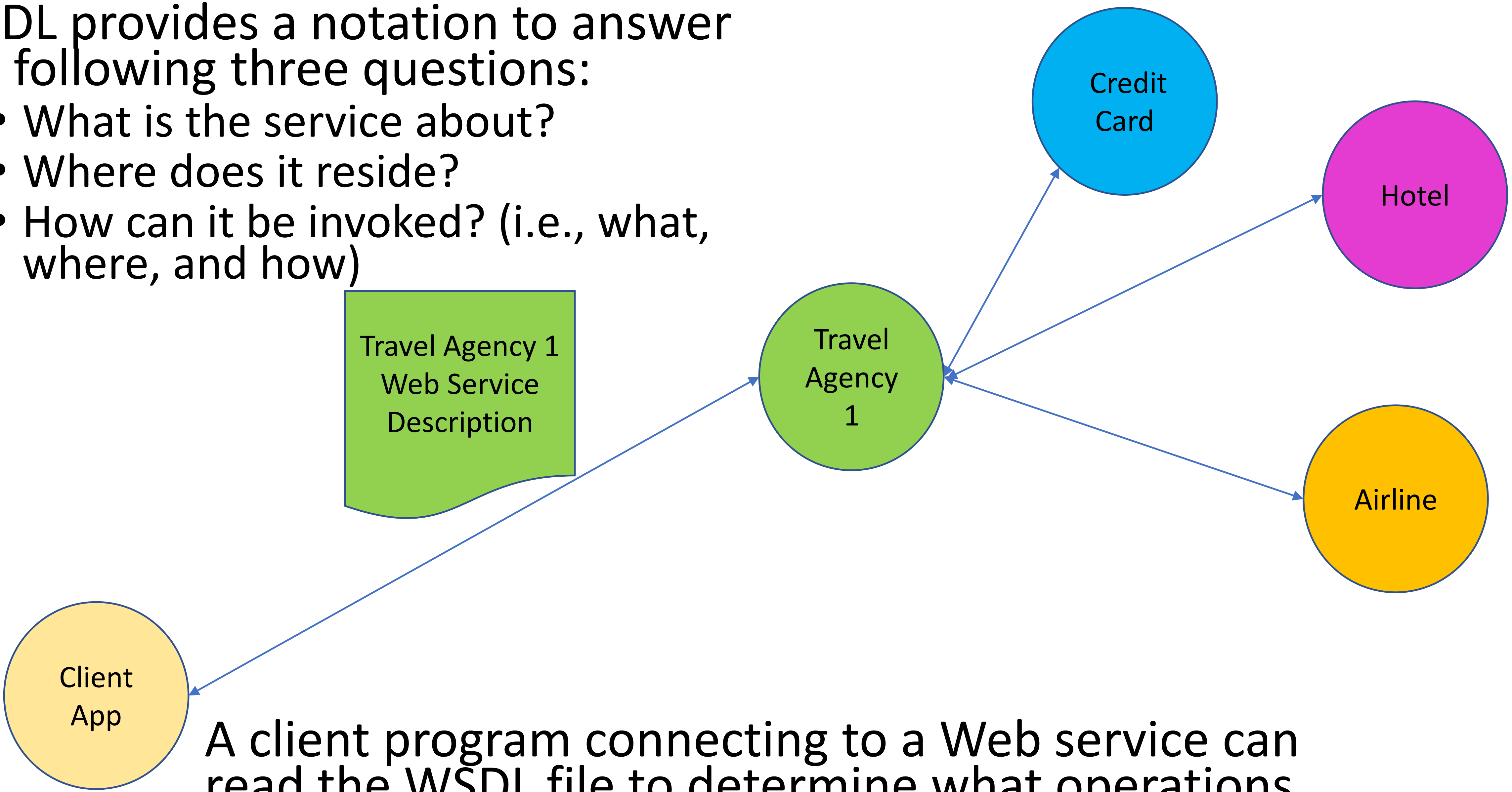
WSDL provides a notation to answer the following three questions:

- What is the service about?
- Where does it reside?
- How can it be invoked? (i.e., what, where, and how)

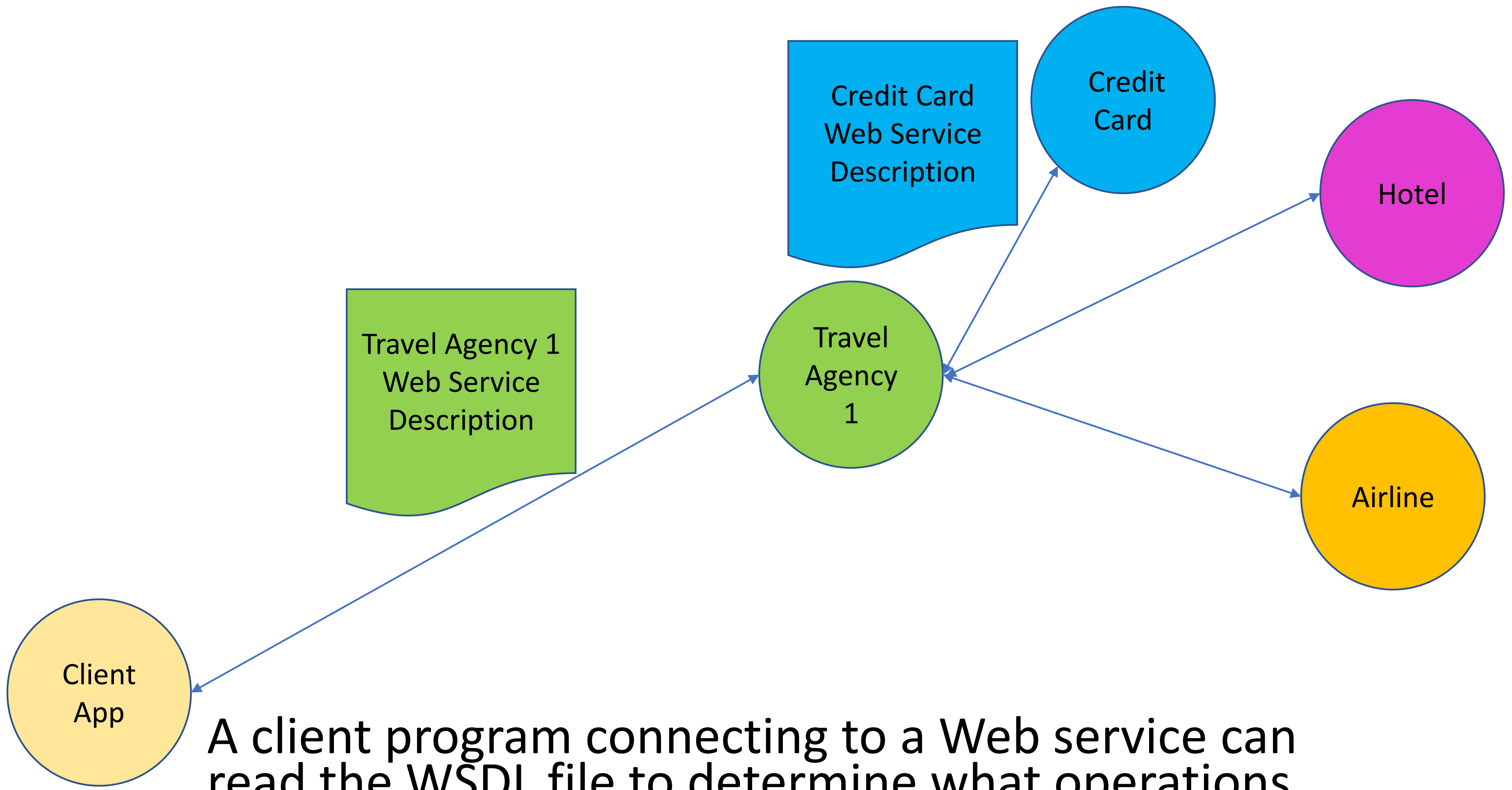


WSDL provides a notation to answer the following three questions:

- What is the service about?
- Where does it reside?
- How can it be invoked? (i.e., what, where, and how)

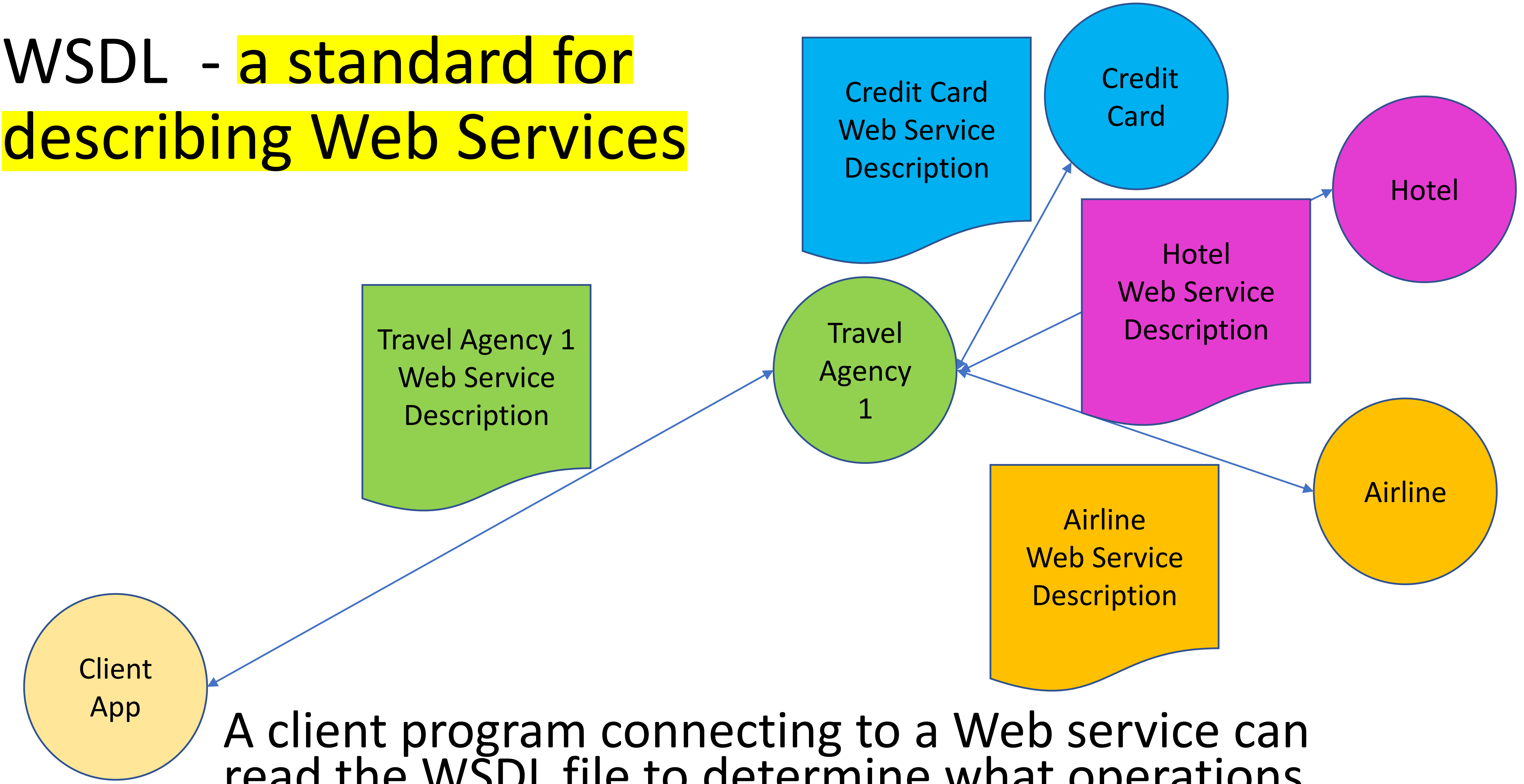


A client program connecting to a Web service can read the WSDL file to determine what operations are available on the server


Credit Card Web Service Description
Credit Card
Hotel
Travel Agency 1 Web Service Description
Travel Agency 1
Airline
Client App

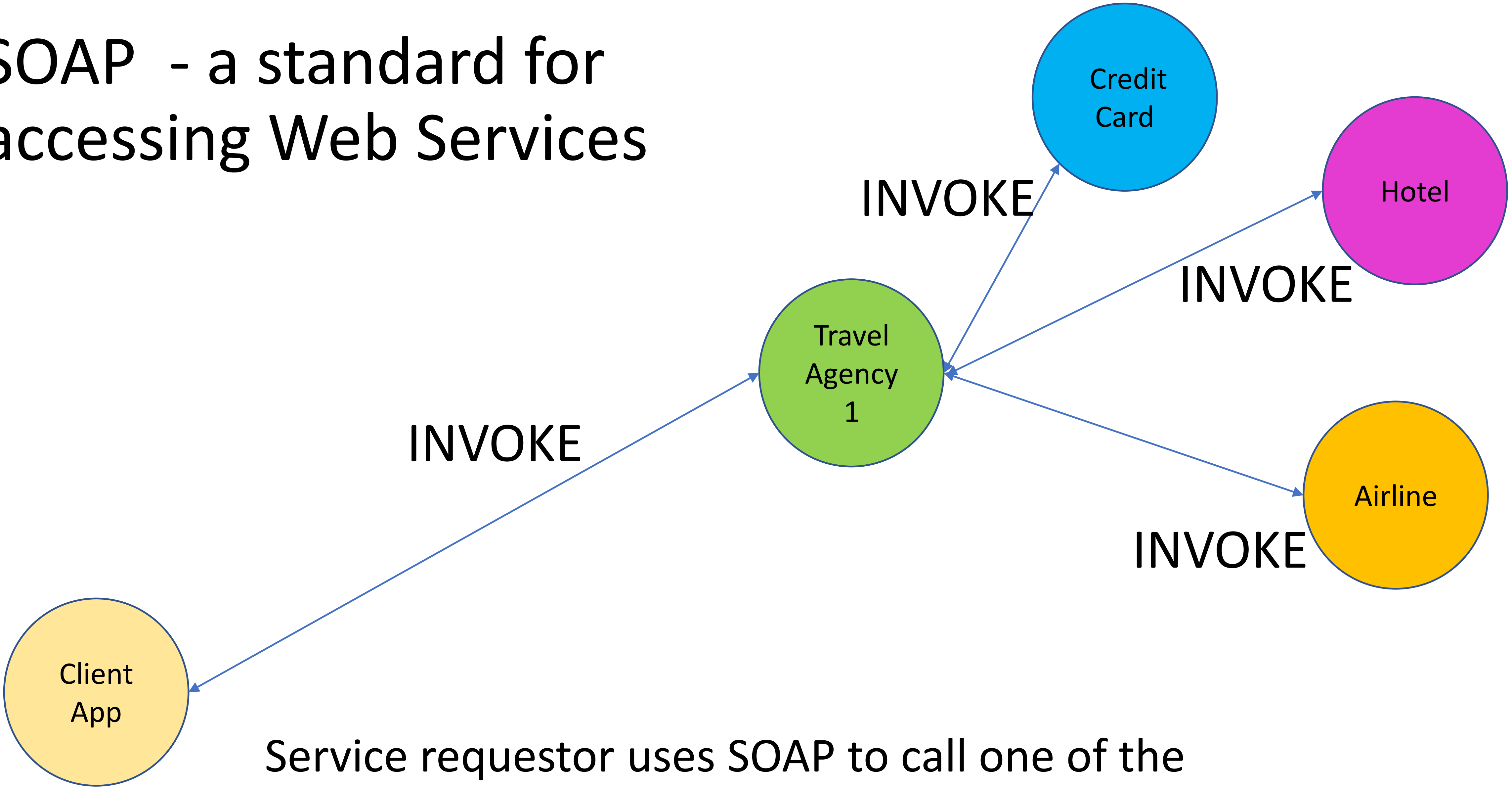
A client program connecting to a Web service can read the WSDL file to determine what operations are available on the server

WSDL - a standard for describing Web Services


Credit Card Web Service Description
Credit Card
Hotel
Hotel Web Service Description
Travel Agency 1 Web Service Description
Travel Agency 1
Airline
Airline Web Service Description
Client App

A client program connecting to a Web service can read the WSDL file to determine what operations are available on the server

SOAP - a standard for accessing Web Services



Service requestor uses SOAP to call one of the operations listed in the WSDL file