

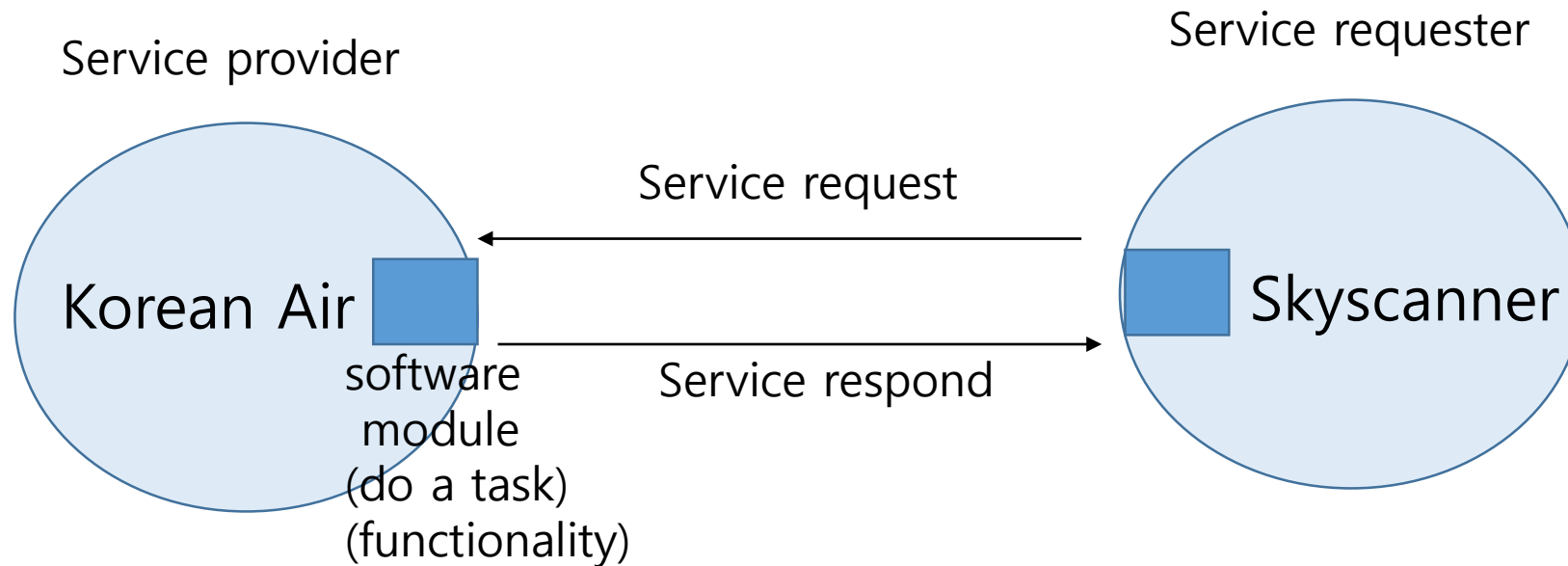
Intro to Web Services

2019. 6. 18

User Case 3

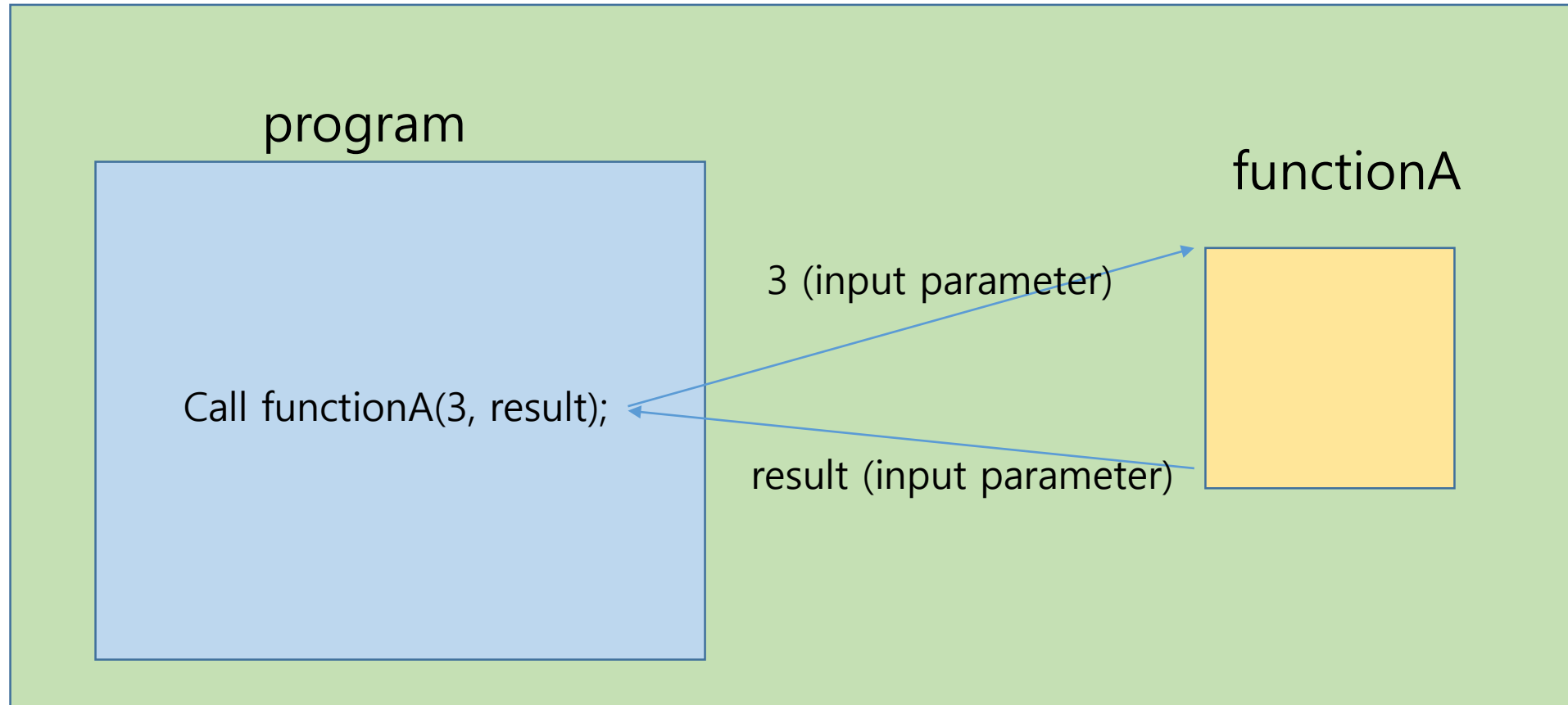
- Suppose that I want to buy an air ticket, so I accessed one travel fare aggregator website, eg, Skyscanner.
- Among the fares offered to me, I chose the fare of MyTrip.com and clicked the link.
- So, I moved to MyTrip.com site. But it didn't request any authentication information to me. Does it already know who I am?
- Later, I wanted to rent a car. So, I clicked Herz on the Skyscanner page.
- Now I moved to the Herz site. Surprisingly, it seemed that it already knew who I am, furthermore my purchase of the air ticket, my departure date, and my destination city.
- How can it be possible?

Web services



Simply put, the web service is a standard way of providing services between a service provider and a service requester on the Web.

Functions in the program



In an application operating on the same programming platforms, a function call is often used to implement a specific task.

Remote Procedure Calls(RPC)

- **RPC** is a common tool for applications on heterogeneous developing platforms to communicate each other before the Web services.
- **CORBA** (Common Object Request Broker Architecture)
 - Based on an object-oriented architecture
 - Developed in a separate language called the Interface Definition Language
- **DOM** (Distributed Component Object Model)
 - Microsoft technology for clients to access remote components

What the Web services should do?

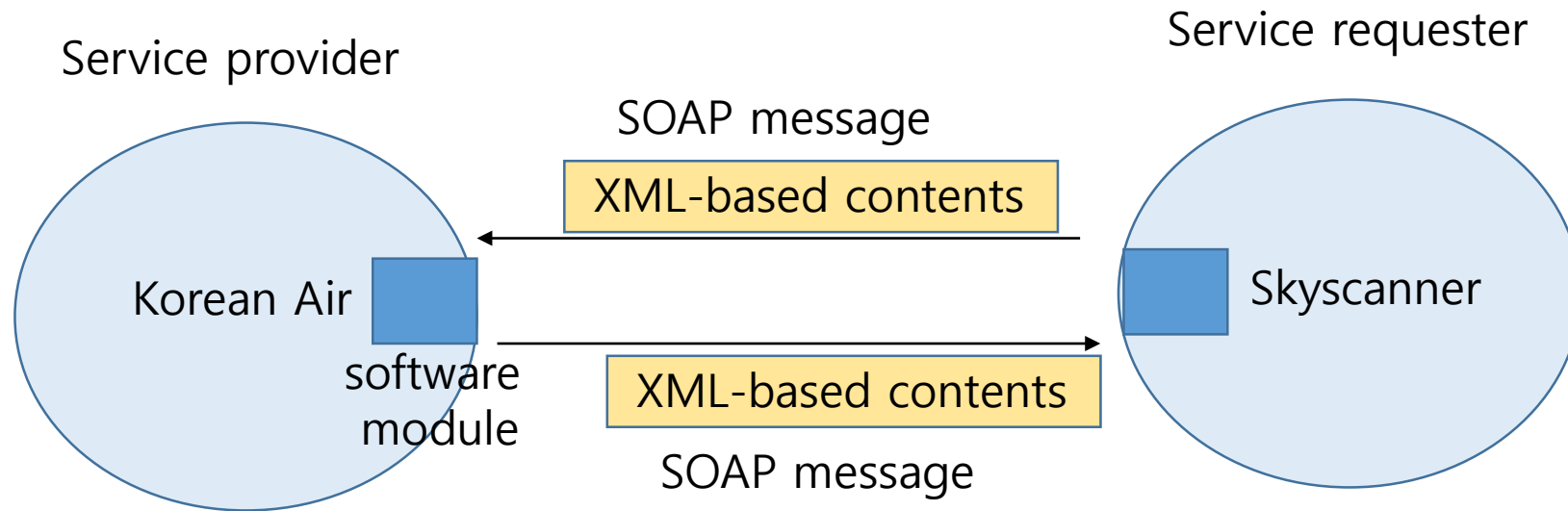
- Modern business applications are web-based applications operating on top of a variety of programming platforms.
- The Web services should provide a standard method, whereby web applications can communicate each other regardless of application developing platforms or languages.
- So, an application is requested only to specify the following:
 - What the service is (service identifier)
 - Input parameters
 - Output parameters
- In this sense, the Web service can be called the RPC over http.

Two types of Web service

- SOAP web services
- REST(RESTful) web services

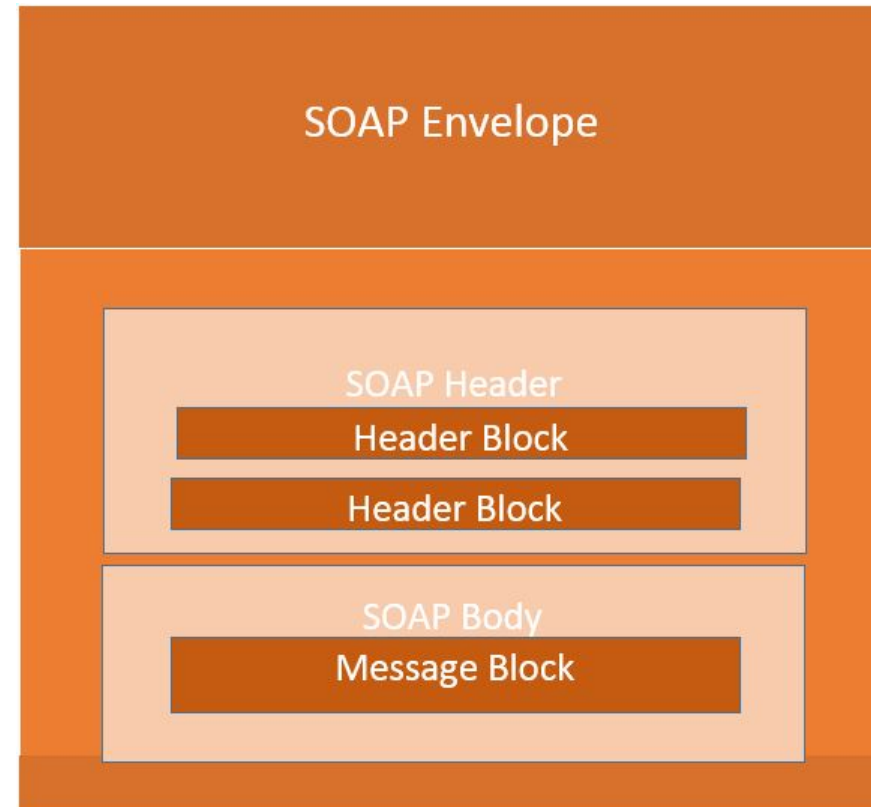
SOAP Web service

- **XML** (Extensible Markup Language) is used as the intermediate language for exchanging data between applications.
- **SOAP** (Simple Object Access Protocol) is a standard protocol to define the messages which convey the data contents described in XML.



SOAP messages

- SOAP specification defines the SOAP message which is exchanged between applications.
- Building blocks of the SOAP message



In the example below, a *GetQuotation* request is sent to a SOAP Server over HTTP. The request has a *QuotationName* parameter, and a Quotation will be returned in the response.

The namespace for the function is defined in <http://www.xyz.org/quotation> address.

Here is the SOAP request –

```
POST /Quotation HTTP/1.0
Host: www.xyz.org
Content-Type: text/xml; charset = utf-8
Content-Length: nnn

<?xml version = "1.0"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV = "http://www.w3.org/2001/12/soap-envelope"
  SOAP-ENV:encodingStyle = "http://www.w3.org/2001/12/soap-encoding">

  <SOAP-ENV:Body xmlns:m = "http://www.xyz.org/quotations">
    <m:GetQuotation>
      <m:QuotationsName>MicroSoft</m:QuotationsName>
    </m:GetQuotation>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

A corresponding SOAP response looks like –

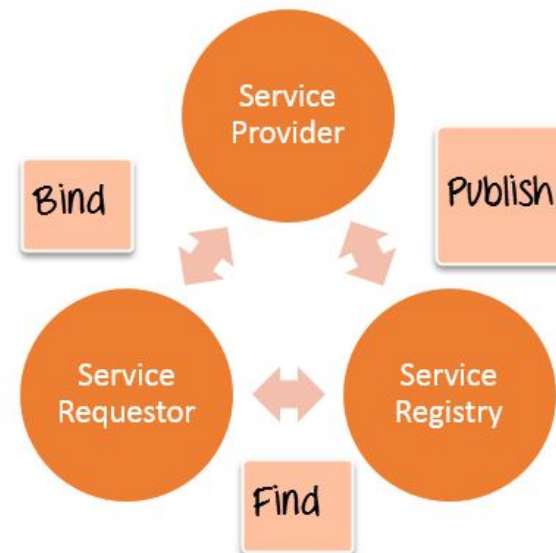
```
HTTP/1.0 200 OK
Content-Type: text/xml; charset = utf-8
Content-Length: nnn

<?xml version = "1.0"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV = "http://www.w3.org/2001/12/soap-envelope"
  SOAP-ENV:encodingStyle = "http://www.w3.org/2001/12/soap-encoding">

  <SOAP-ENV:Body xmlns:m = "http://www.xyz.org/quotation">
    <m:GetQuotationResponse>
      <m:Quotation>Here is the quotation</m:Quotation>
    </m:GetQuotationResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

WSDL

- WSDL (Web Service Description Language) is an XML-based file which tells the client application what the web service does.
- WSDL file contains
 - The location of the web service
 - The methods which are exposed by the web service
- Web service architecture



History of Web services

- In 2002, the Web consortium published the specification of **SOAP Web service**.
- In 2004, the Web consortium also released another standard called **REST (Representational State Transfer)**. Over the years this standard is so popular that many websites around world are using REST to access resources such as documents or pictures or videos.

REST

- REST is called an **architectural style**, not the protocol like the SOAP.
- **Basic REST implementation principles** are:
 - define resource as URI
 - <http://demo.mju.com/employee/1>
 - use HTTP methods as REST verbs to access resource
 - GET/POST/PUT/DELETE
 - Web service providers expose their service functionalities as **APIs**
- Exchanged data can be represented by any formatting languages, such as JSON or XML, etc.
 - Currently, JSON is widely used.

REST example

<https://openclassrooms.com/en/courses/3432056-build-your-web-projects-with-rest-apis/3496011-identify-examples-of-rest-apis>

SOAP vs. REST

- **Resources and bandwidth**
 - Since SOAP messages are heavier in contents, they consume greater bandwidth than REST.
- **Stateless**
 - REST is more convenient for the applications that have no need to maintain a state of information from one request to another.
- **Security**
 - REST has no security mechanism in itself. It relies on https.
- All in all, SOAP is more appropriate for enterprise-oriented applications which require a formal means of communication, while REST is more efficient for lightweight applications, especially running on laptops or mobile devices.

Web service security