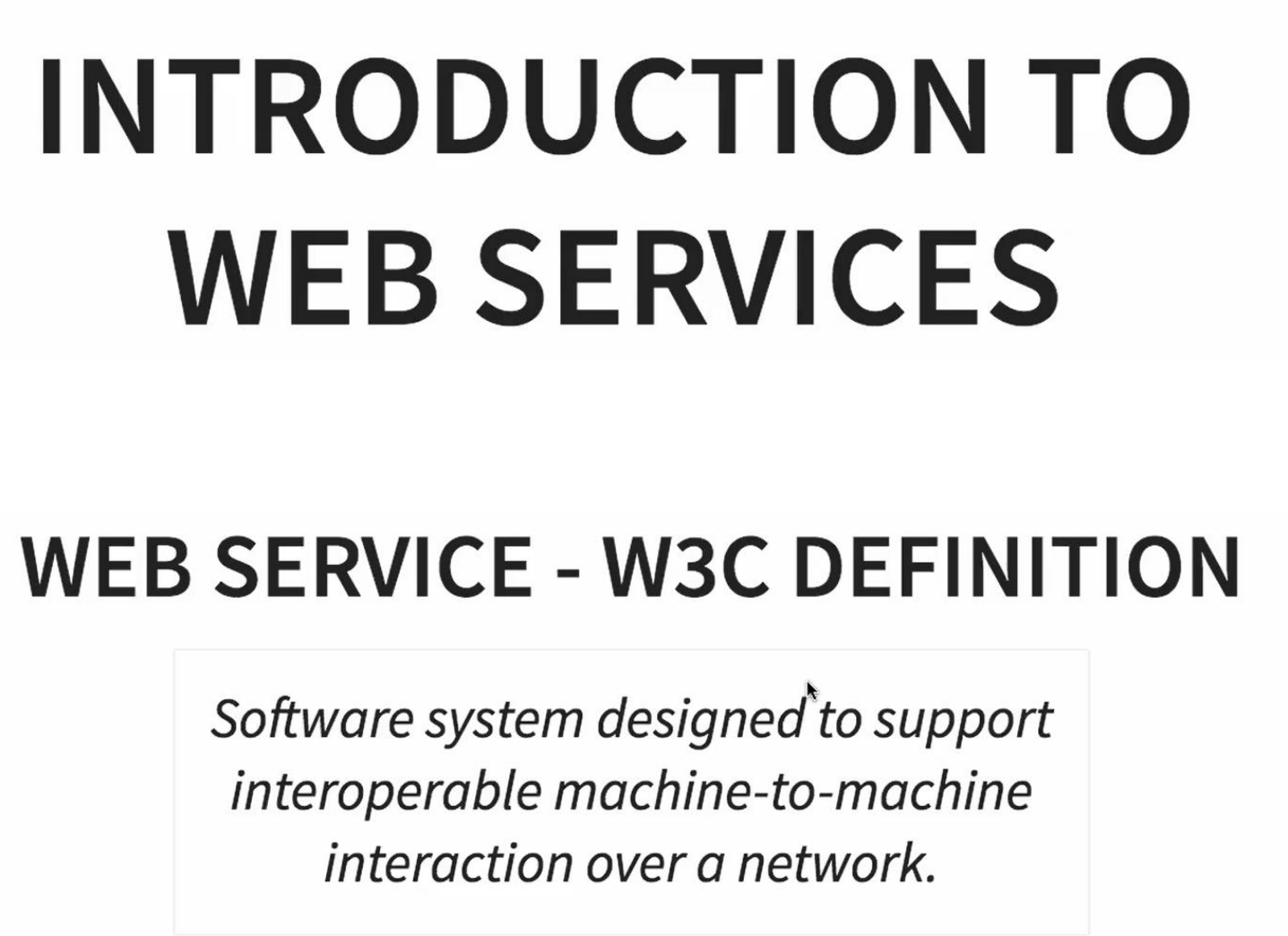
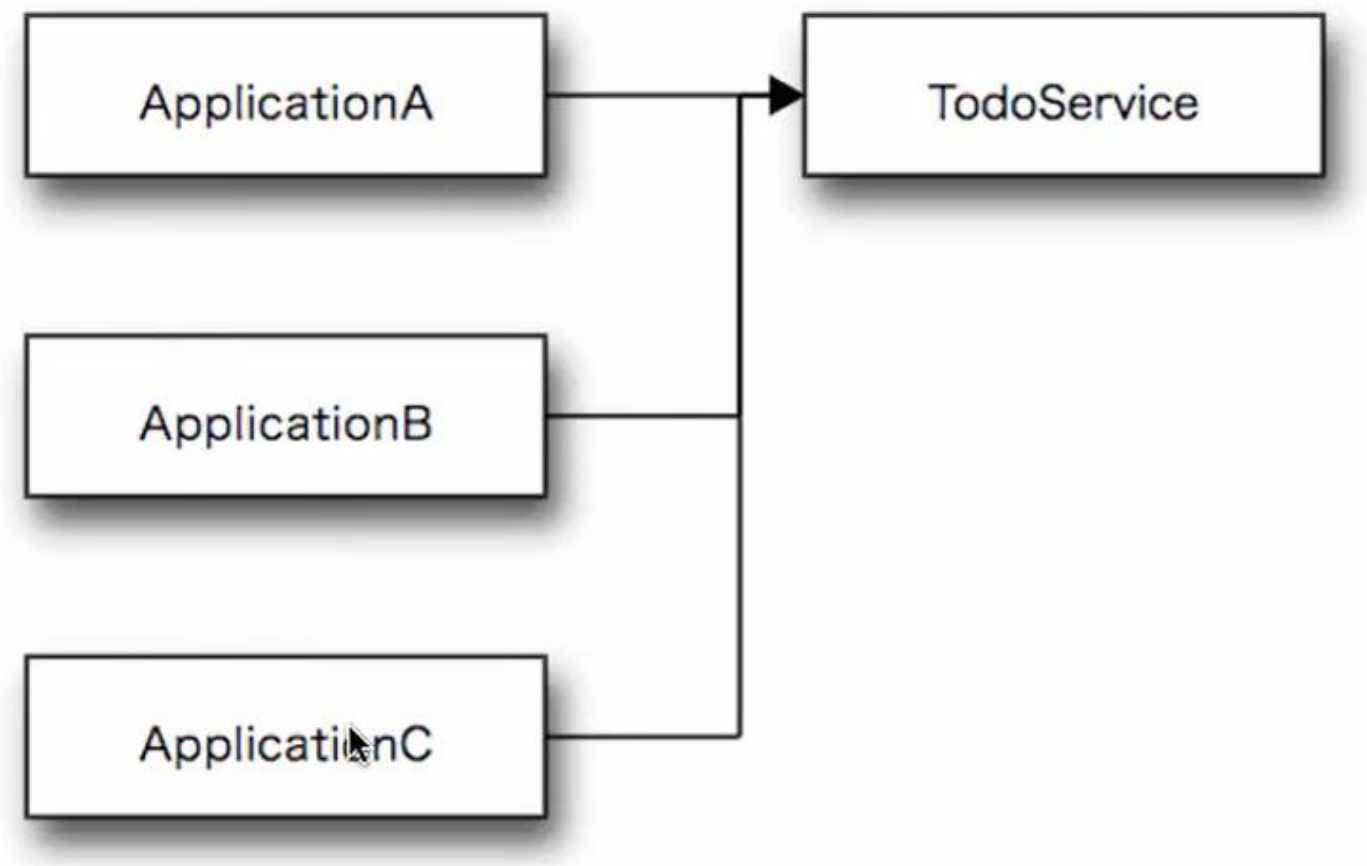
2. Introduction to Web Services

1. **What is Webservices**

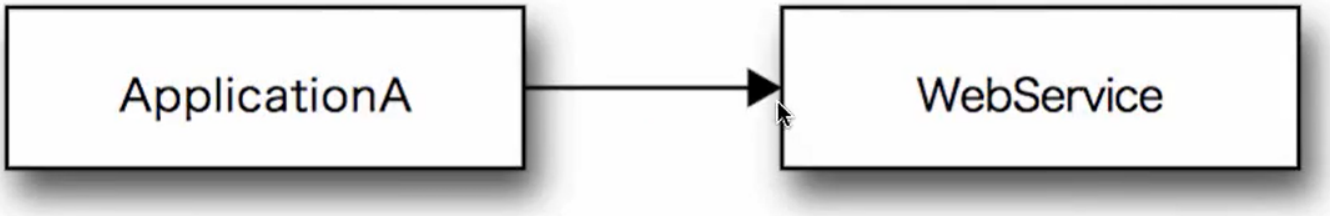


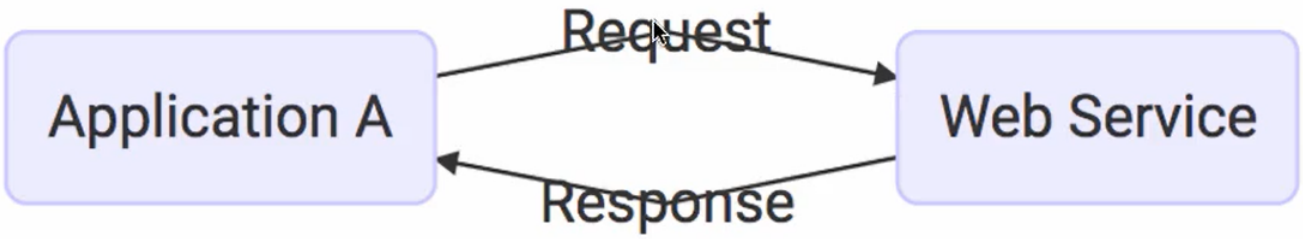
3 KEYS

- Designed for machine-to-machine (or application-to-application) interaction
- Should be interoperable - Not platform dependent
- Should allow communication over a network

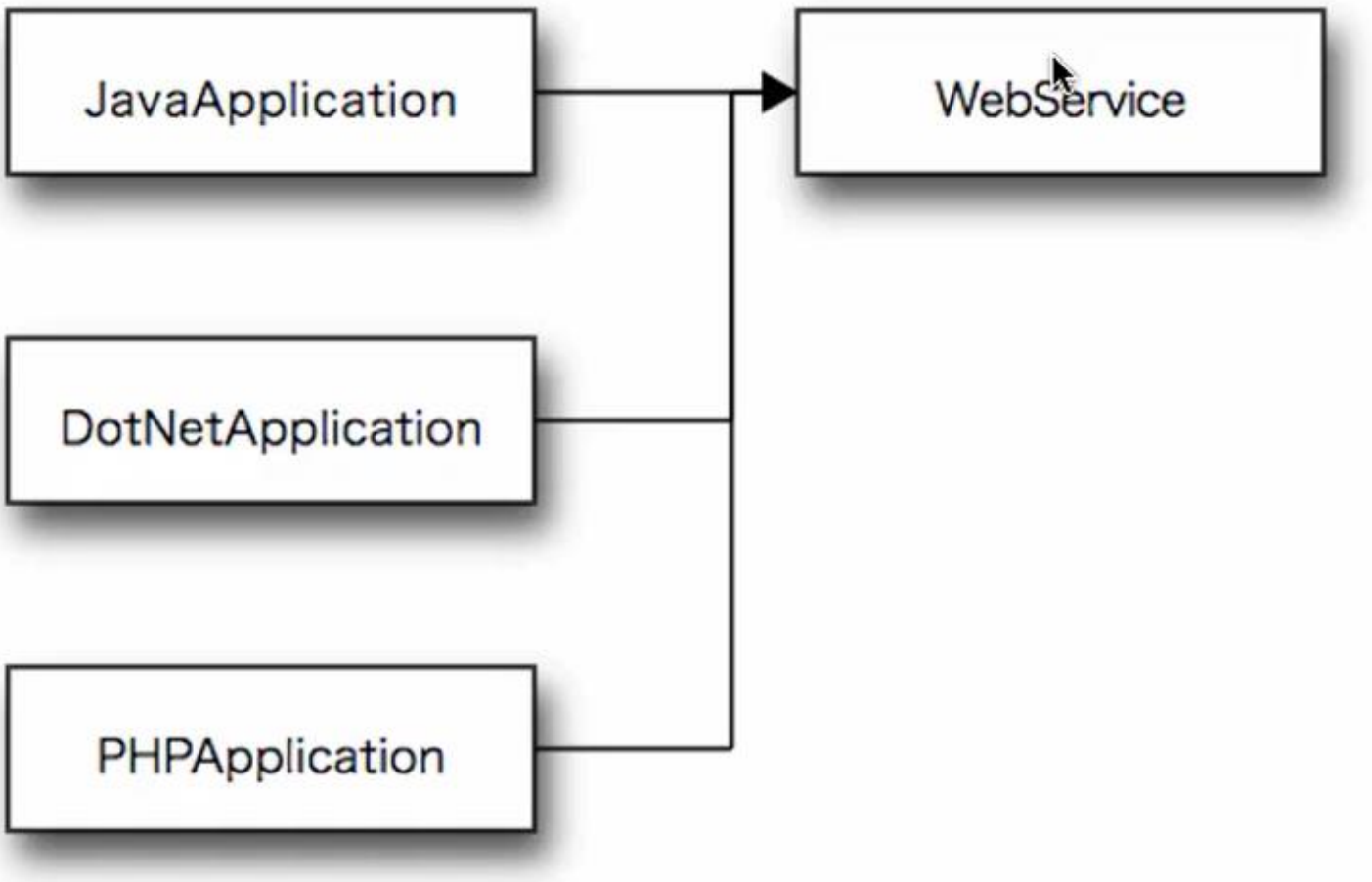


How does data exchange between applications take place?





How can we make web services platform independent?



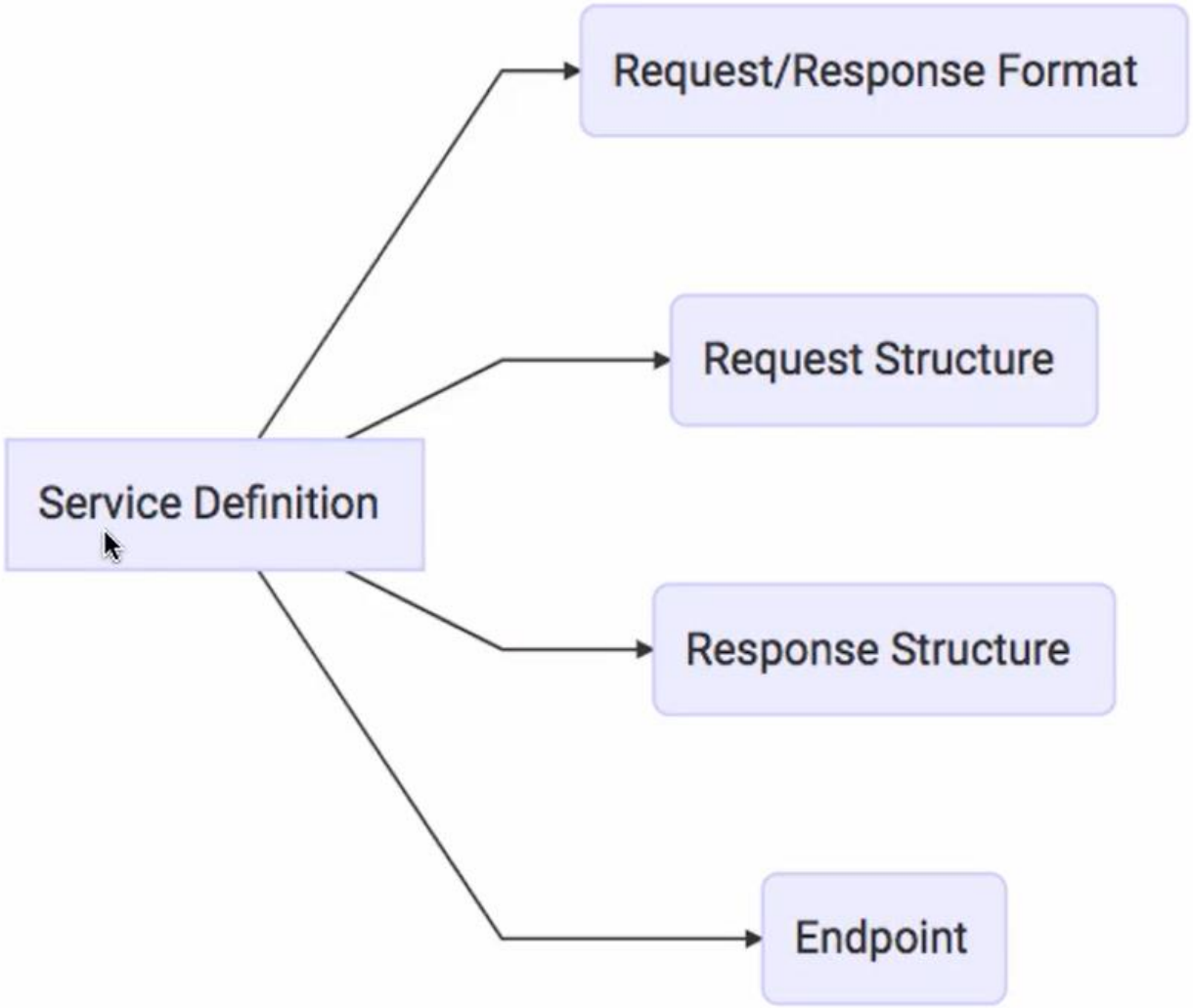
XML

```
<getCourseDetailsRequest>
    <id>Course1</id>
</getCourseDetailsRequest>
```

JSON


```
[
  {
    "id": 1,
    "name": "Even",
    "birthDate": "2017-07-10T07:52:48.270+0000"
  },
  {
    "id": 2,
    "name": "Abe",
    "birthDate": "2017-07-10T07:52:48.270+0000"
  }
]
```


How does the Application A know the format of Request and Response?



Web Services – Key Terminology

KEY TERMINOLOGY

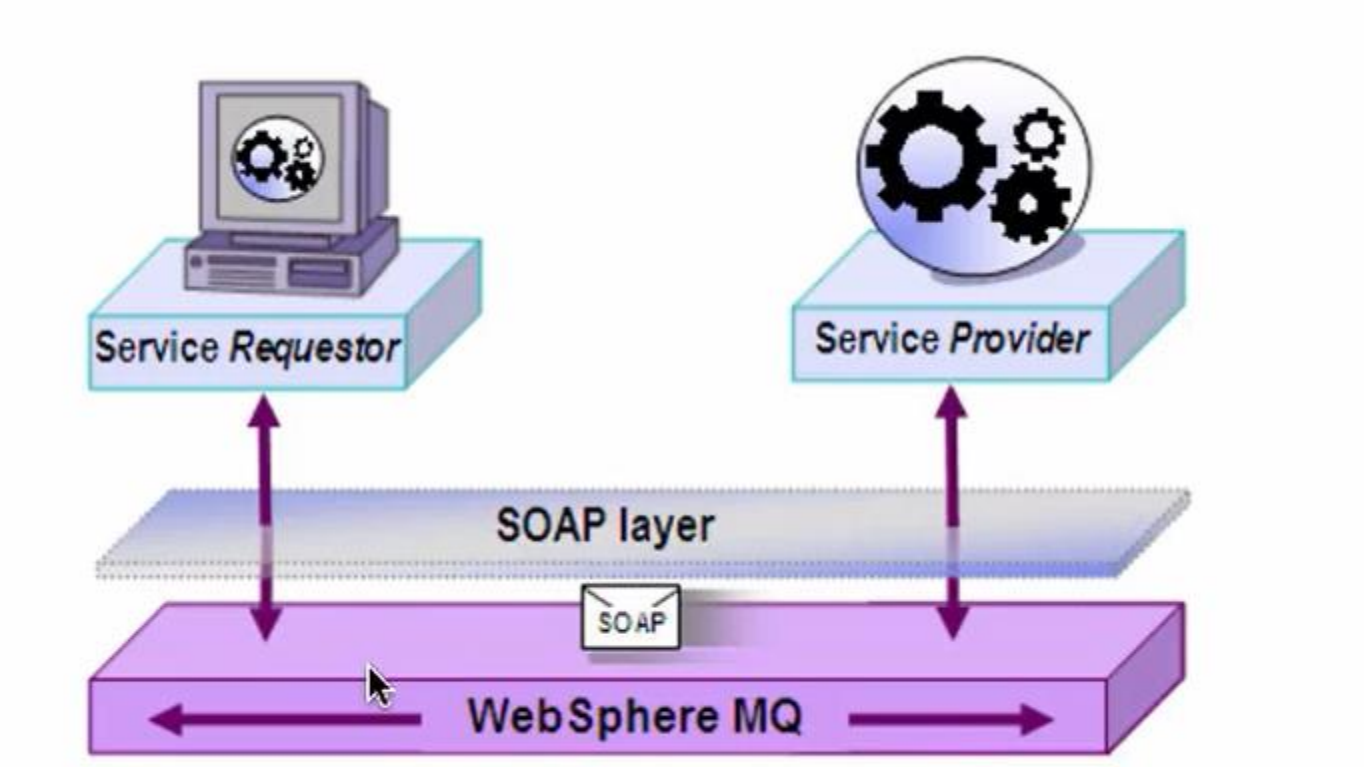
- Request and Response
- Message Exchange Format
  - XML and JSON

KEY TERMINOLOGY

- Service Provider or Server
- Service Consumer or Client
- Service Definition

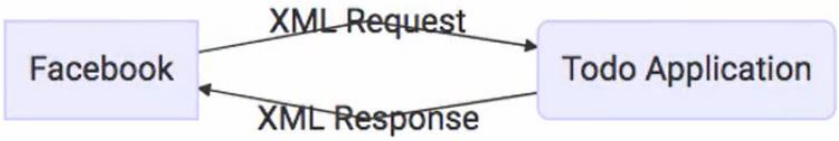
KEY TERMINOLOGY

- Transport
  - HTTP and MQ

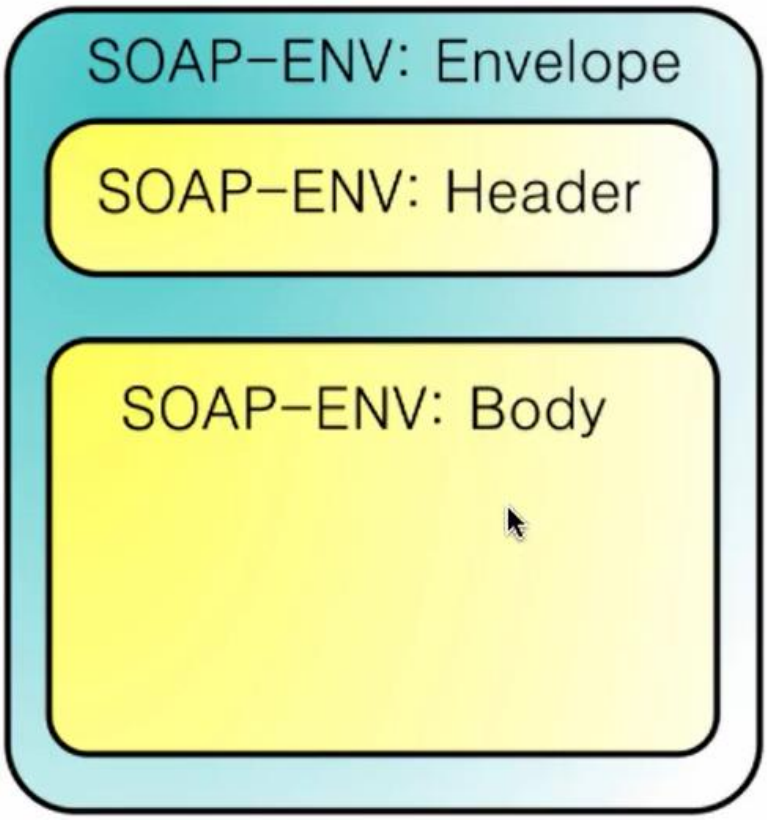


Introduction to SOAP Web Services

SOAP?




```
<getCourseDetailsRequest>
    <id>Course1</id>
</getCourseDetailsRequest>
```

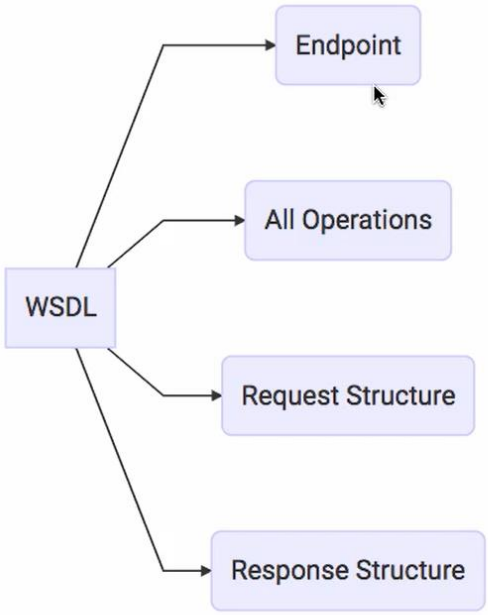


SOAP Header is optional

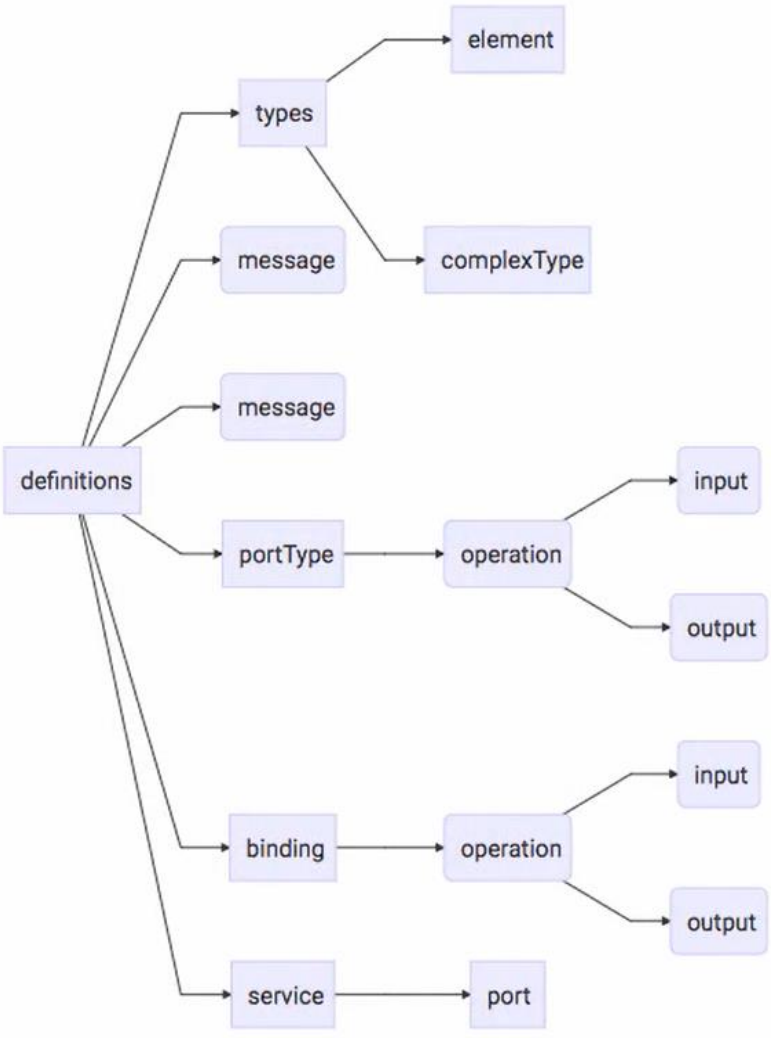
```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:getCourseDetailsResponse xmlns:ns2="http://in28mi
            <ns2:course>
                <ns2:id>Course1</ns2:id>
                <ns2:name>Spring</ns2:name>
                <ns2:description>10 Steps</ns2:description>
            </ns2:course>
        </ns2:getCourseDetailsResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP

- Format
  - SOAP XML Request
  - SOAP XML Response
- Transport
  - SOAP over MQ
  - SOAP over HTTP
- Service Definition
  - WSDL



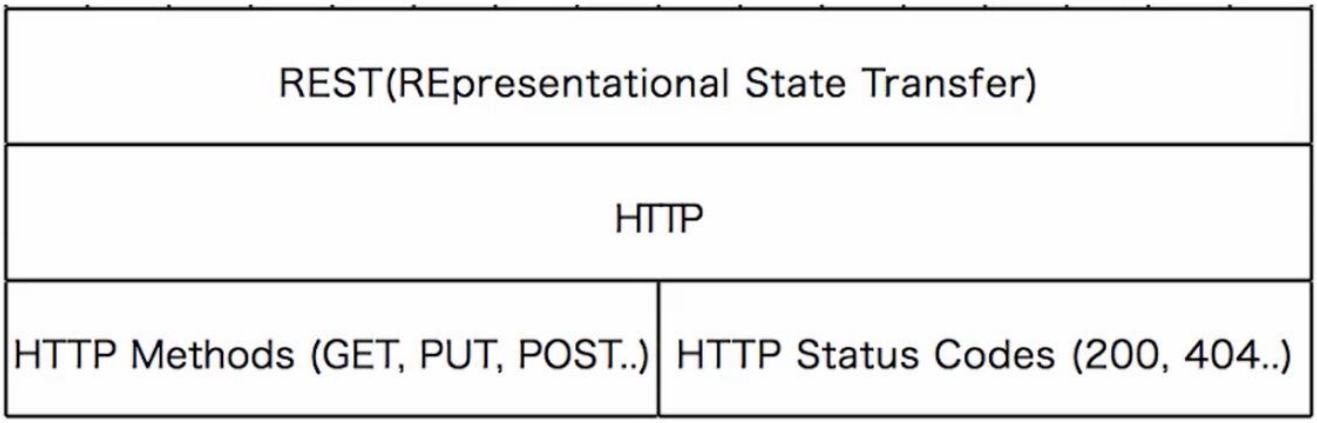
Web Service Definition Language



Introduction RESTful Web Services

REpresentational State Transfer

MAKE BEST USE OF HTTP



KEY ABSTRACTION - RESOURCE

- A resource has an URI (Uniform Resource Identifier)
  - /user/Ranga/todos/1
  - /user/Ranga/todos
  - /user/Ranga
- A resource can have different representations
  - XML
  - HTML
  - JSON

EXAMPLE

- Create a User - POST /users
- Delete a User - DELETE /users/1
- Get all Users - GET /users
- Get one Users - GET /users/1

REST

- Data Exchange Format
  - No Restriction. JSON is popular
- Transport
  - Only HTTP
- Service Definition
  - No Standard. WADL/Swagger/...

SOAP vs RESTful Web Services

REST VS SOAP

- Restrictions vs Architectural Approach
- Data Exchange Format
- Service Definition
- Transport
- Ease of implementation