Lecture 5.d

Web Services I
Protocols and Models

Dr. Gabriele Pierantoni
17.02.2021

Web Service Protocols and Models

Web Service Protocols (1)

XML

- eXtensible Markup Language
- Uniform data representation and exchange mechanism.

WSDL

- Web Services Description Language
- Standard meta language to described the services offered

Web Service Protocols (2)

UDDI

- Universal Description, Discovery and Integration specification
- Mechanism to register and locate WS based application

SOAP

- Simple Object Access Protocol
- Standard way for communication

Web Service Model (1)

Interactions between three roles:

- **Service Provider**
- **Service Registry**
- **Service Consumer (or Requestor)**

Web Service Model (2)

Interactions involve three operations:

Publish Operation

- **The service provider *publish*es its service(s) to a service registry such as UDDI in the form of a WSDL document.**

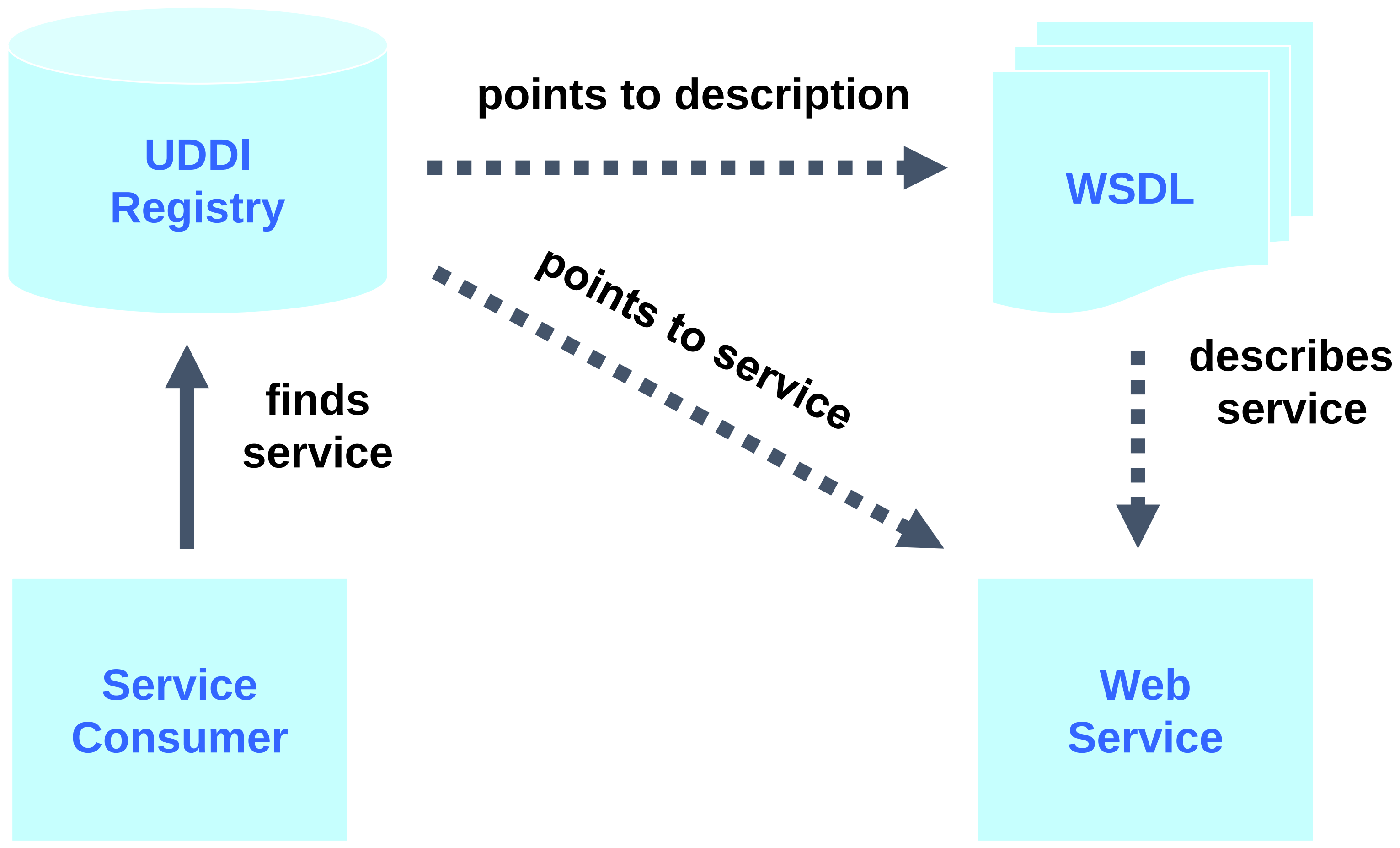
Find Operation

- **The service consumer *find*s services for consumption via service registries and this process is also called "service discovery"**

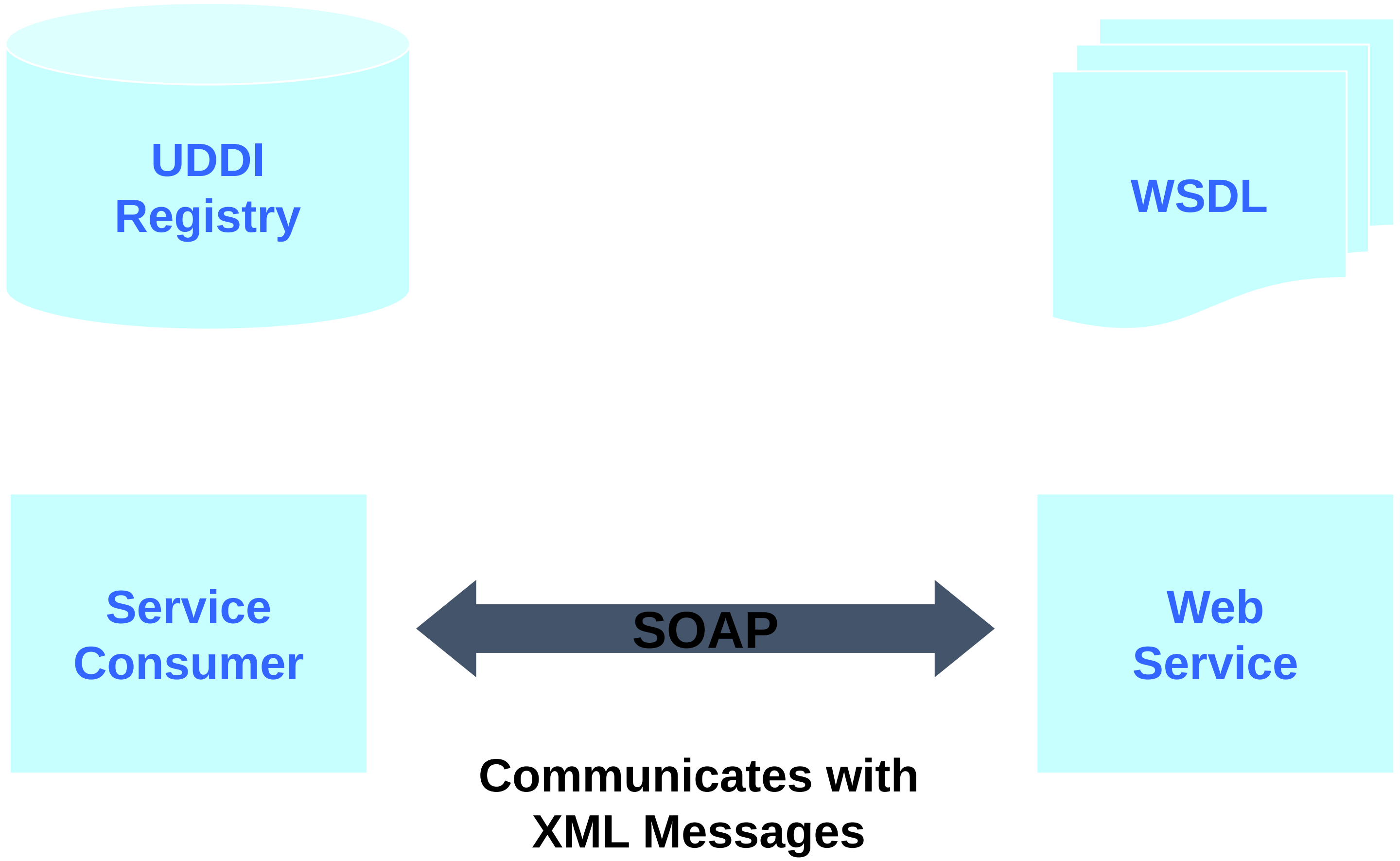
Bind Operation

- **Once the service consumer has acquired the service information, it can attempt to *bind* to the service and use it**

Web Service Model: Publish and Find



Web Service Model: Bind



Web Service Operations (1)

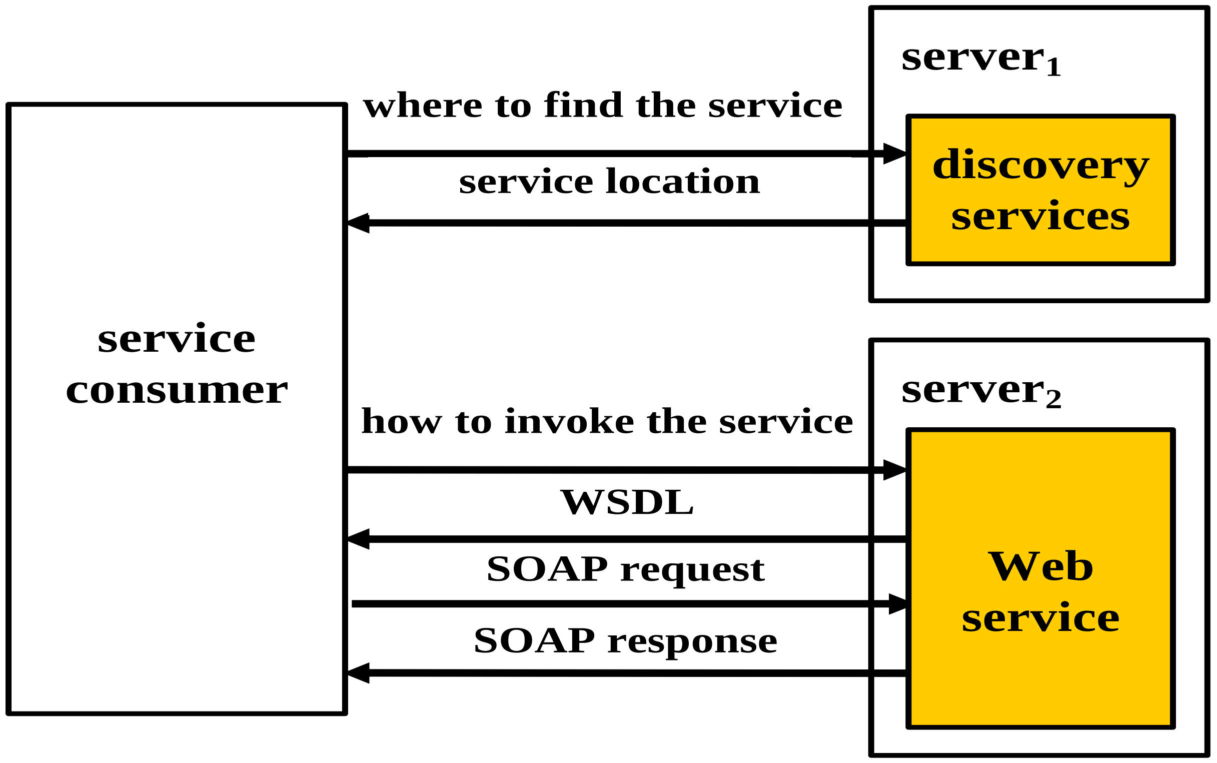
Operations of Web services Discovery

- **Service consumer may have no knowledge of what web service it is going to invoke**

- **It contacts a discovery service, which is itself a Web service, to identify and find a web service that meets its requirements**

- **The discovery service replies telling which service providers can provide the required service**

Web Service Operations (1)



Web Service Operations (2)

Operations of Web services invocation:

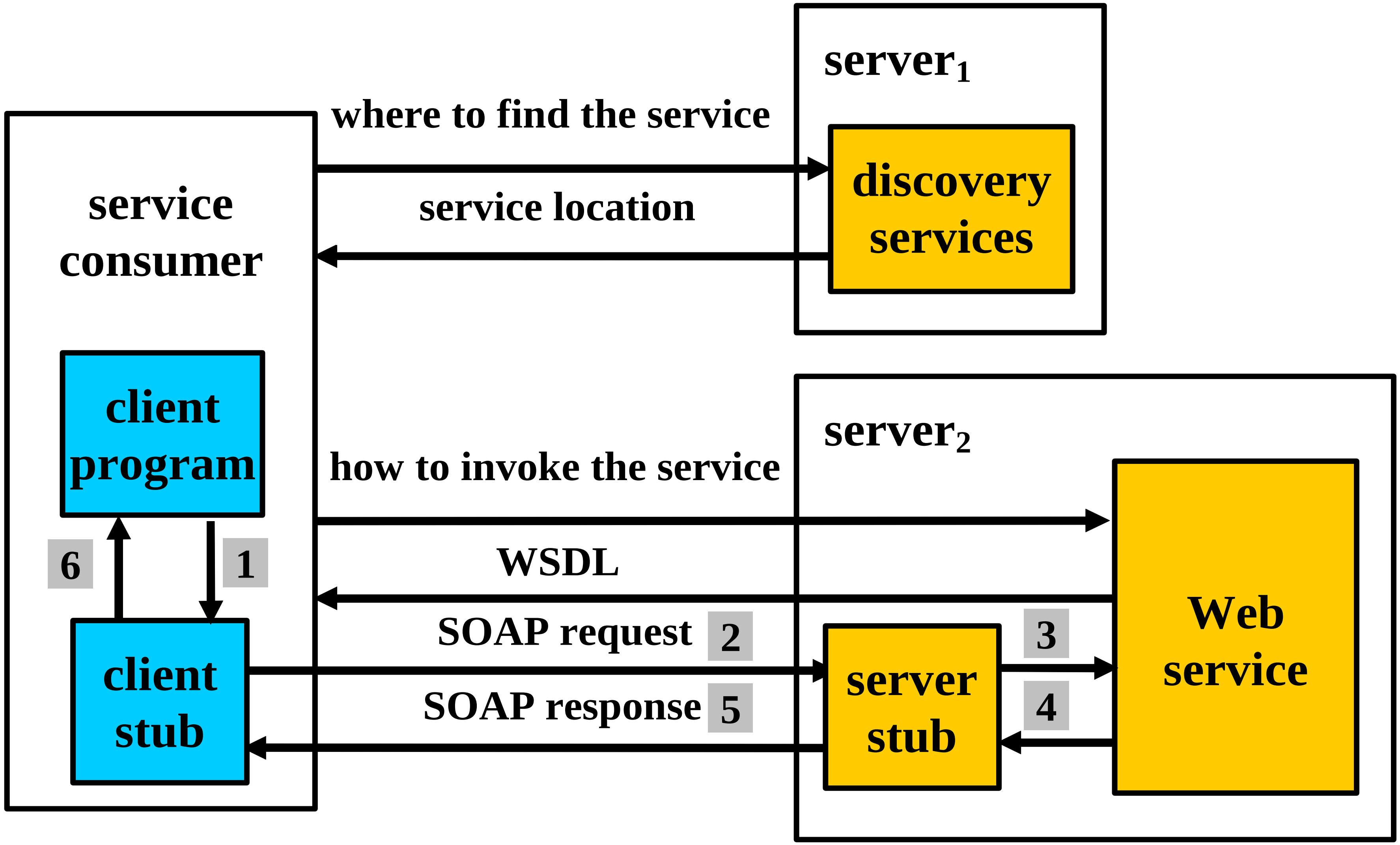
- **Knowing the location of the web service, the service consumer may have no idea of how to invoke it**
- **Service consumer has to ask the web service to describe itself, i.e. to tell how exactly to invoke it, and Web Service returns its description in WSDL format**
- **Service Consumer sends a SOAP request asking for the description of the web service**

Web Service Operations (2)

Operations of Web services invocation:

- Web Service replies with a SOAP response, which includes the description or an error message if the SOAP request was incorrect
- Web Services programmers usually write code in one of the high-level programming languages or in WSDL, while the SOAP code is always generated and interpreted automatically
- Service Consumers and Web Services use proxies (or stubs), which are generated automatically usually based on the WSDL description of the web service, to invoke web services

Web Service Operations (3)



Web Service Operations (4)

Web service invocation

1. **Whenever the service consumer's application needs to invoke the Web Service, it will call the client stub, which turns this 'local invocation' into a proper SOAP request (*marshalling* or serializing *process*)**
2. **The SOAP request is sent over a network using the HTTP protocol, and the server receives the SOAP requests and hands it to the server stub**
3. **The server stub converts the SOAP request into something the service implementation can understand (*unmarshalling* or *de-serializing process*)**

Web Service Operations (4)

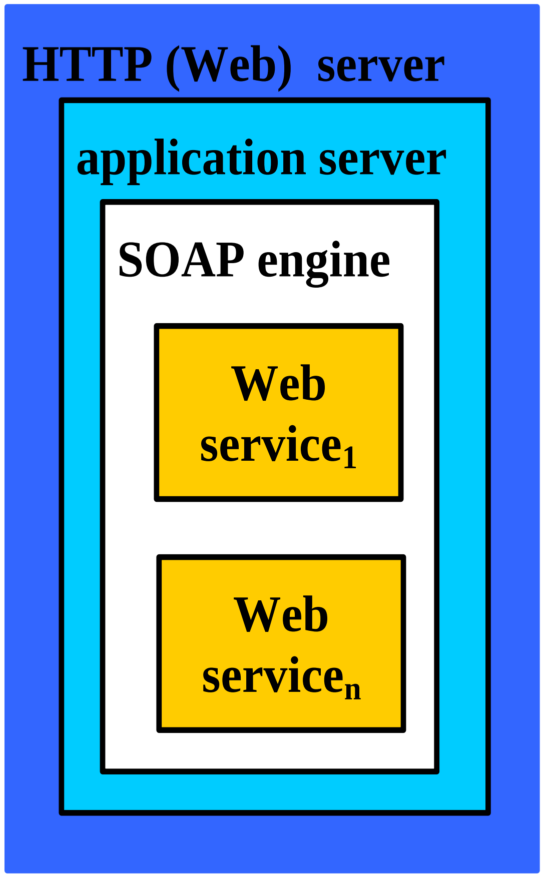
Web service invocation

1. **Once the SOAP request has been de-serialized, the server stub invokes the service implementation, which then carries out the work it has been asked to do, and the result of the requested operation is handed to the server stub, which turns it into a SOAP response**
2. **The SOAP response is sent over a network using the HTTP protocol and the client stub receives the SOAP response and turns it into something the service consumer's application can understand**
3. **Finally the application receives the result of the Web Service invocation and uses it**

Web Servers

SOAP engine:

- Knows how to handle SOAP requests and responses manipulating SOAP
- It is more common to use a generic SOAP engine, for example the *Apache Axis*, than to actually generate stubs for each individual Web service



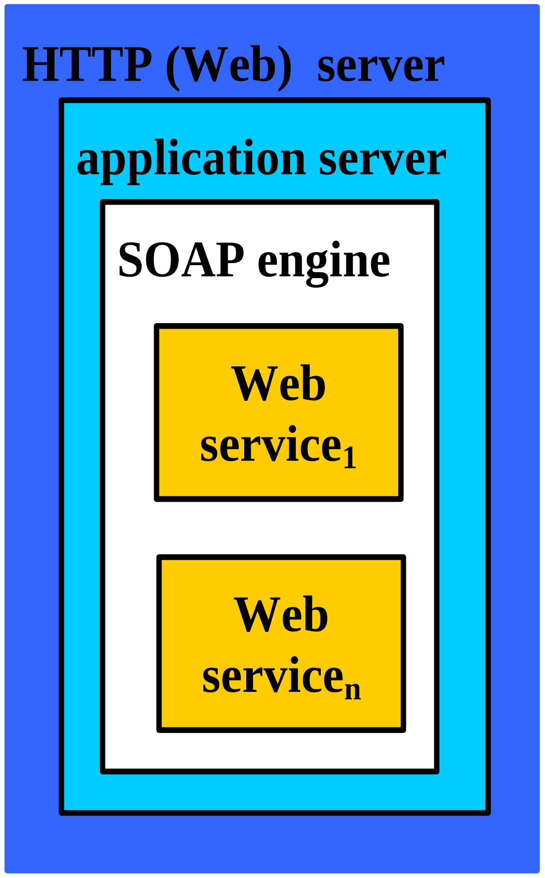
Web Servers

Application Server:

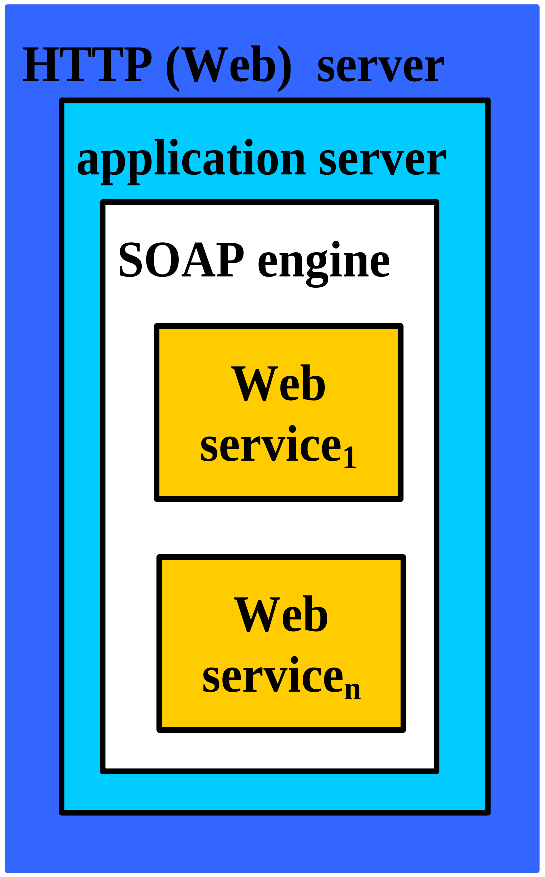
- Provides a 'living space' for applications that must be accessed by different clients

- The SOAP engine runs as an application inside the application server, for example inside the *Apache Tomcat*

- Many application servers already include some HTTP functionality and users can have Web services running by installing a SOAP engine and an application server



Web Servers



HTTP (or Web) Server

- **Handles HTTP messages for example the *Apache HTTP Server***