Web Fundamentals

Phong Tran

January 5, 2023

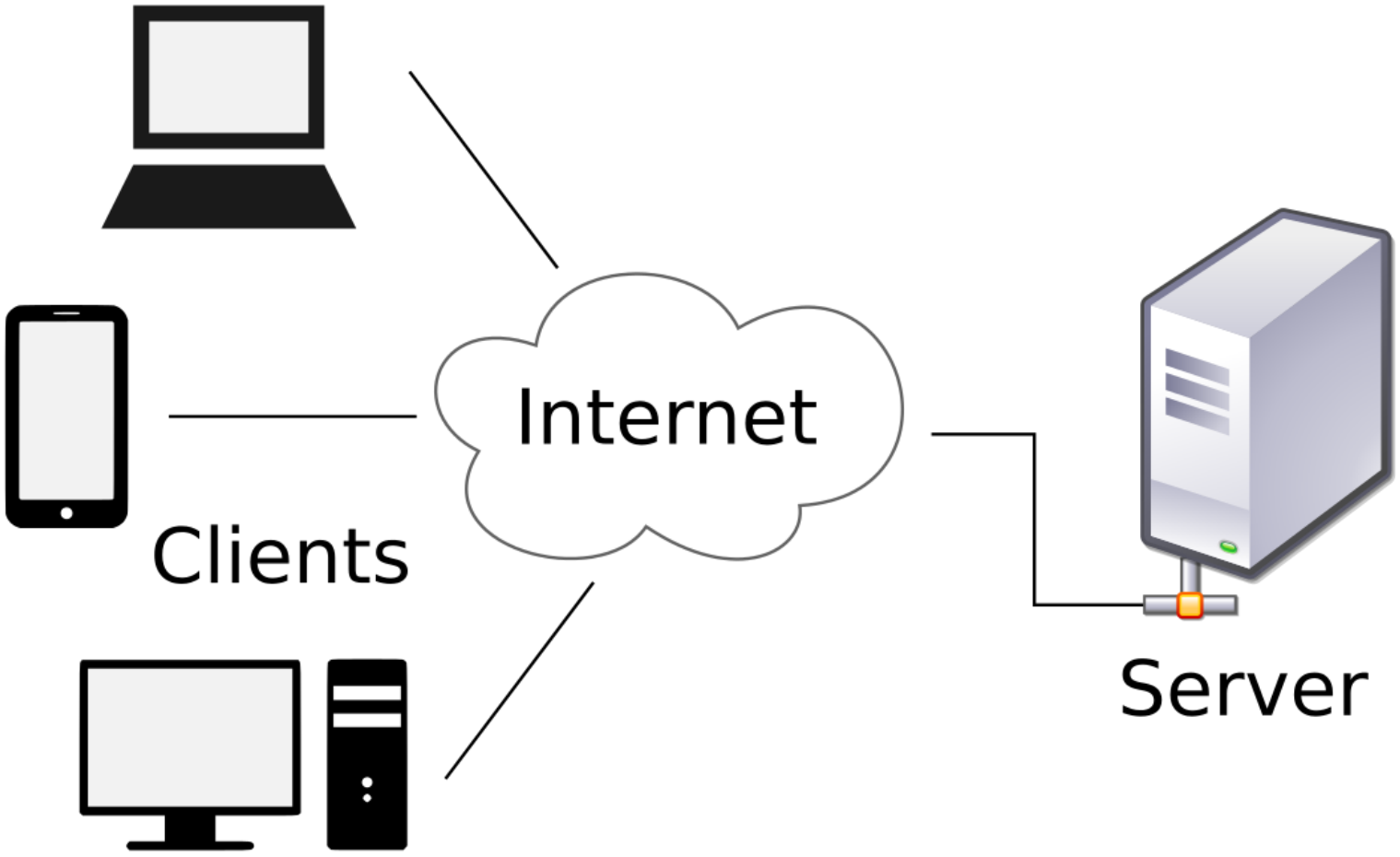
Table of Contents

- Client - Server Model
- Session/Cookie
- Webservice

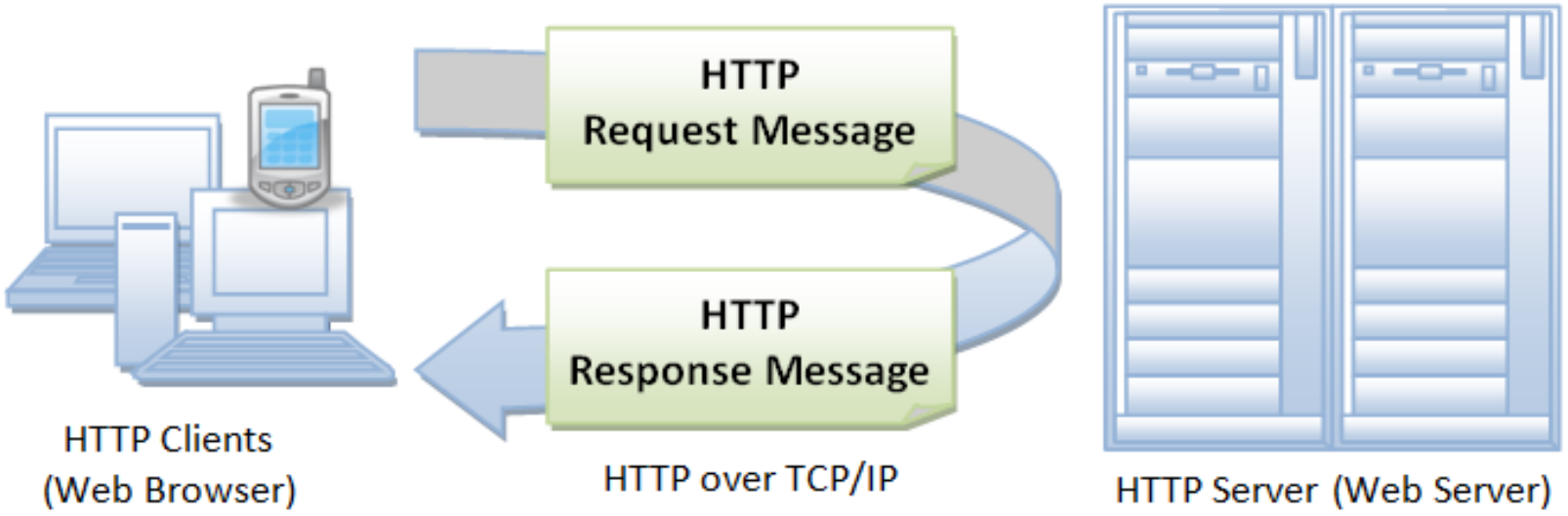
Client - Server Model

- What does Client-Server Model mean?
- Hypertext Transfer Protocol (HTTP)
- Domain Name System (DNS)
- Simple Mail Transfer Protocol (SMTP)
- Telnet

What does Client-Server Model mean?



Hypertext Transfer Protocol (HTTP)

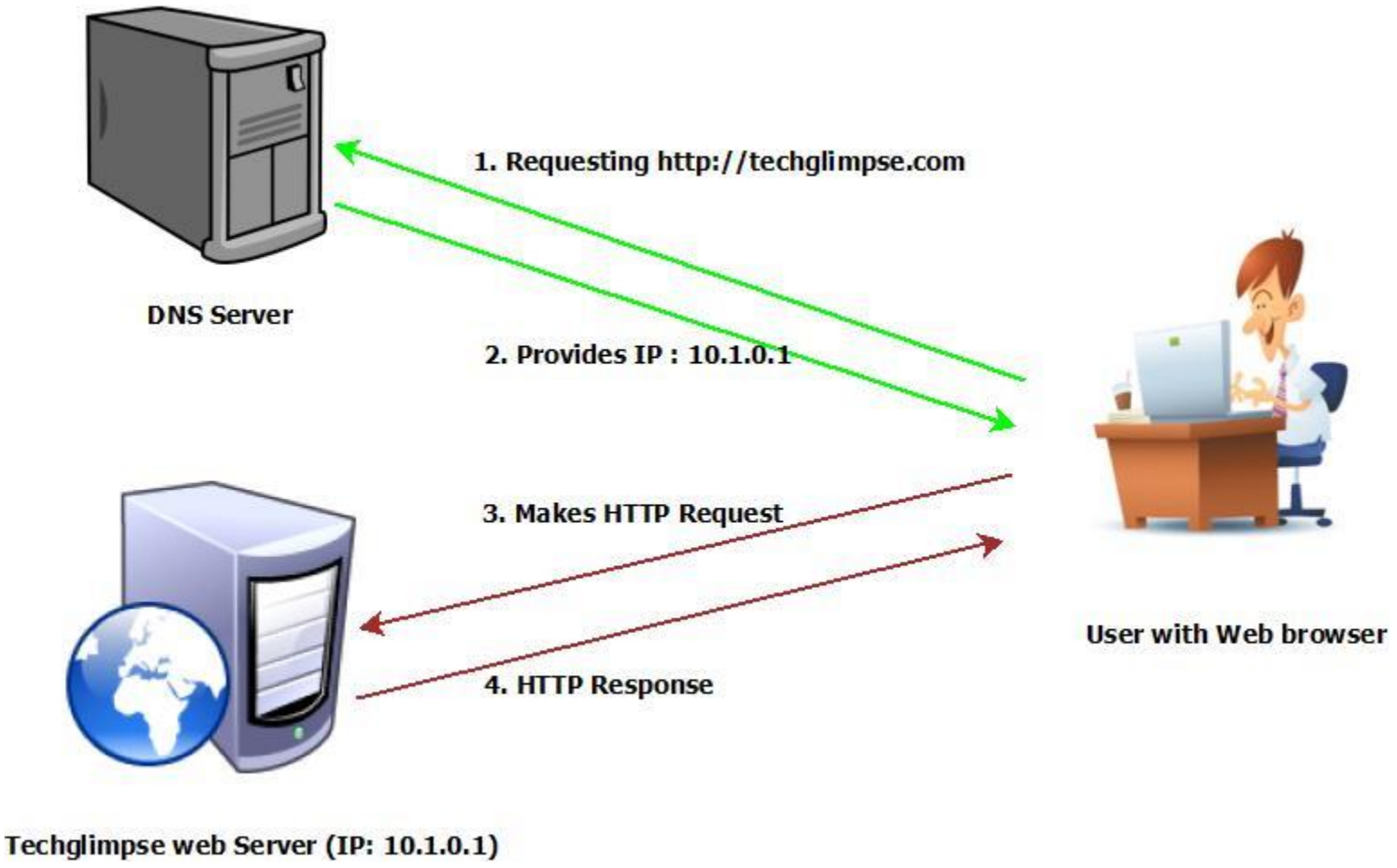


HTTP - Methods

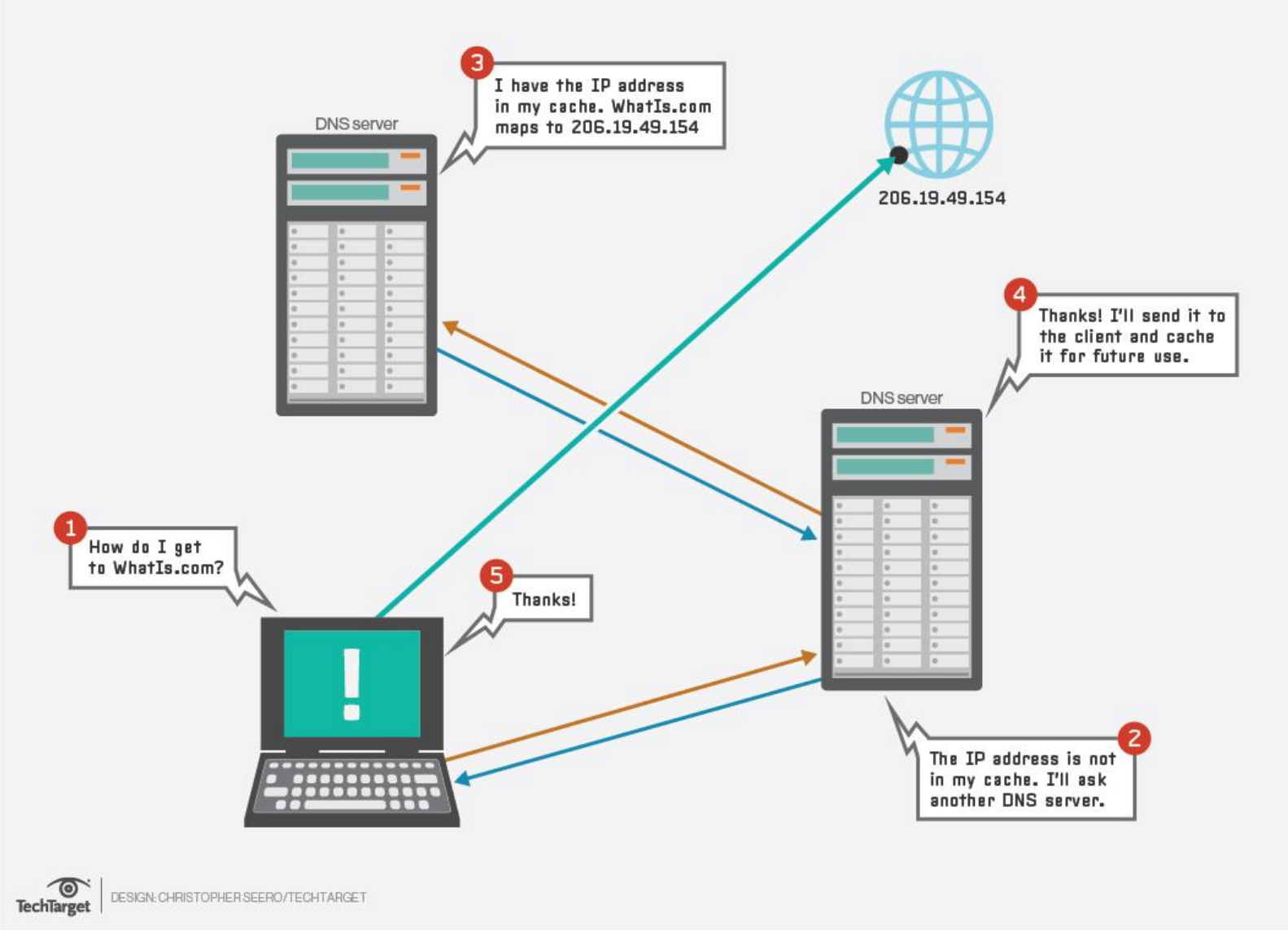
smartosc

HTTP Method	RFC	Request Has Body	Response Has Body	Safe	Idempotent	Cacheable
GET	RFC 7231	No	Yes	Yes	Yes	Yes
HEAD	RFC 7231	No	No	Yes	Yes	Yes
POST	RFC 7231	Yes	Yes	No	No	Yes
PUT	RFC 7231	Yes	Yes	No	Yes	No
DELETE	RFC 7231	No	Yes	No	Yes	No
CONNECT	RFC 7231	Yes	Yes	No	No	No
OPTIONS	RFC 7231	Optional	Yes	Yes	Yes	No
TRACE	RFC 7231	No	Yes	Yes	Yes	No
PATCH	RFC 5789	Yes	Yes	No	No	No

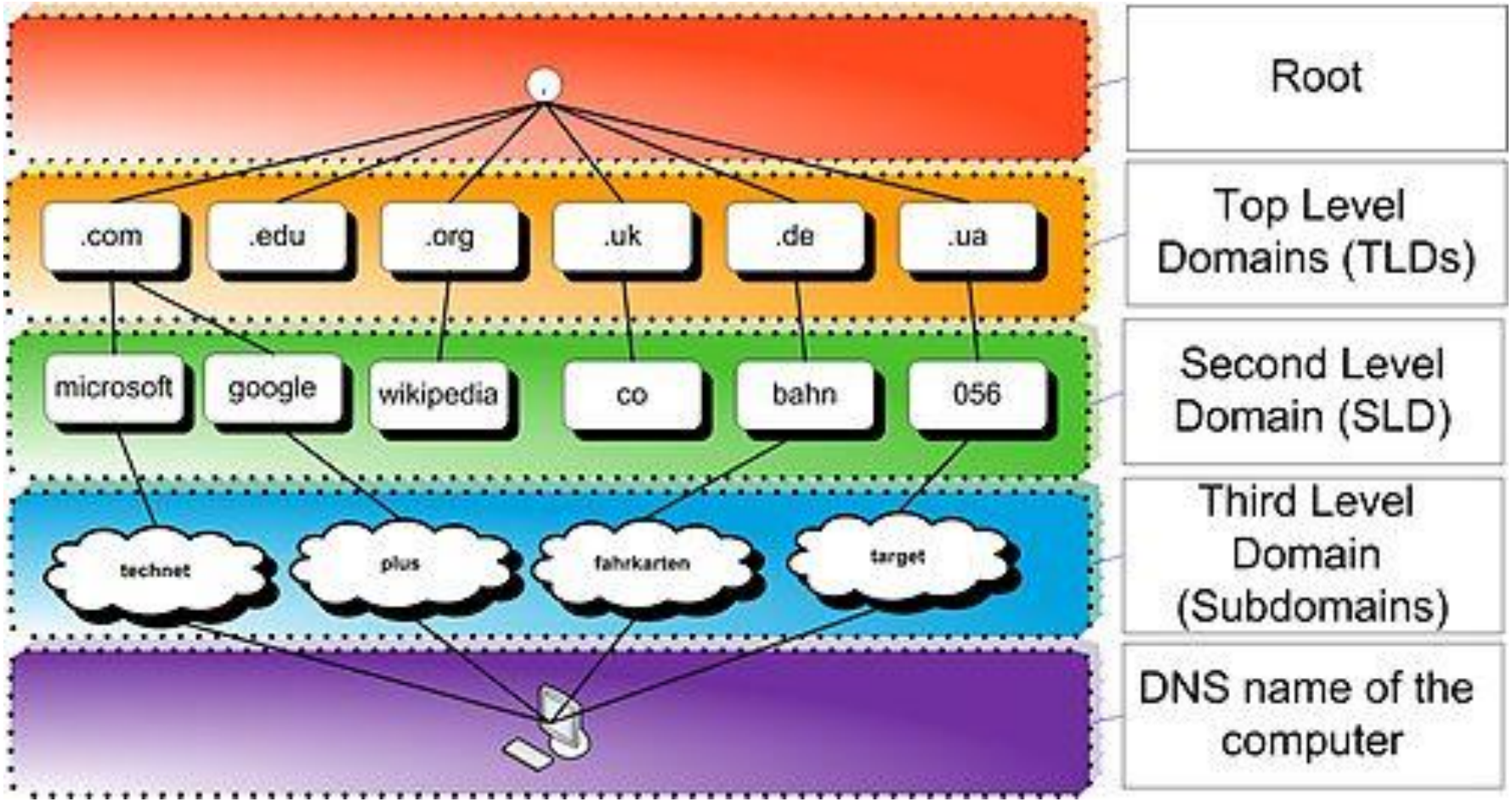
Domain Name System (DNS)



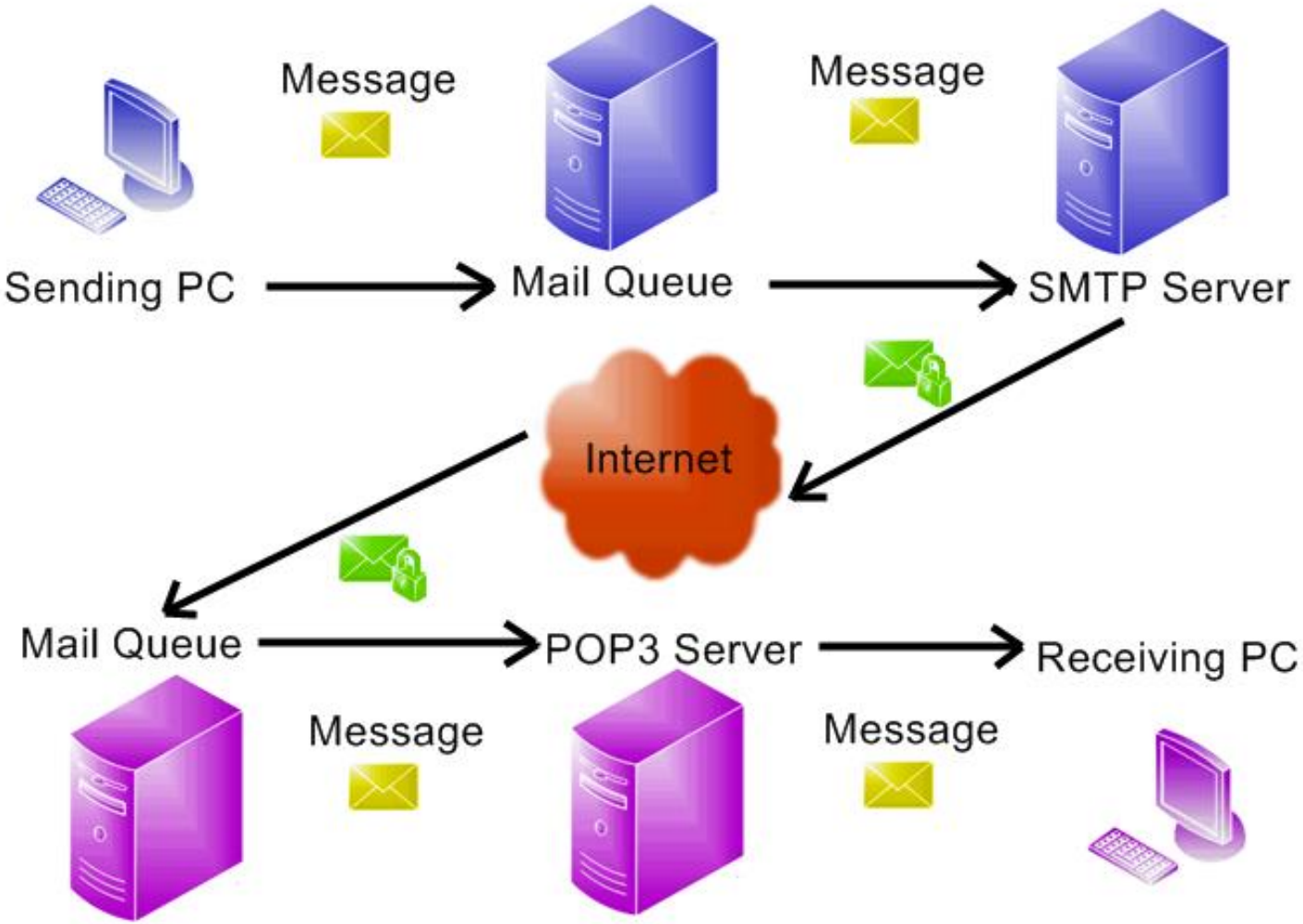
DNS - How does DNS work?



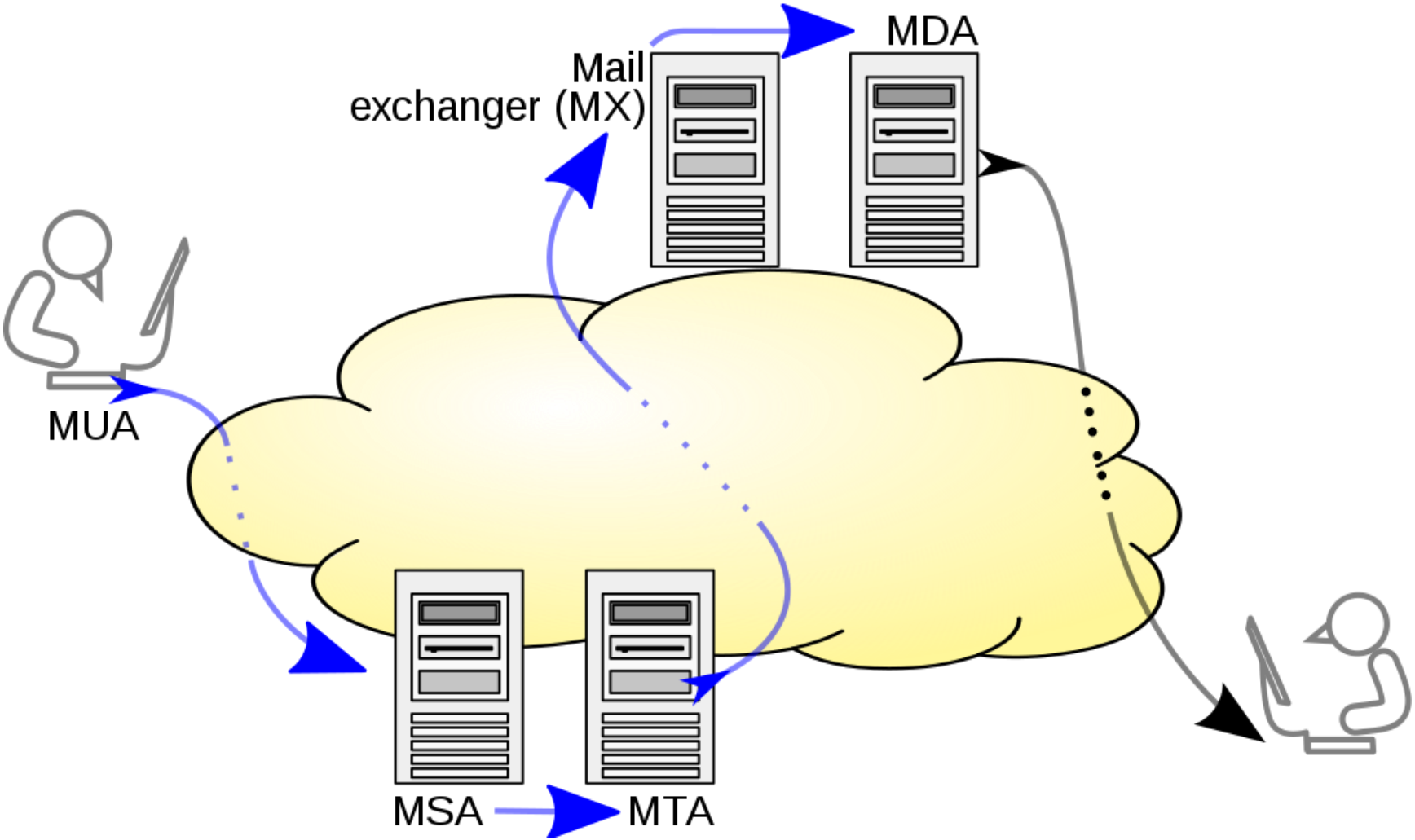
DNS - Hierarchy



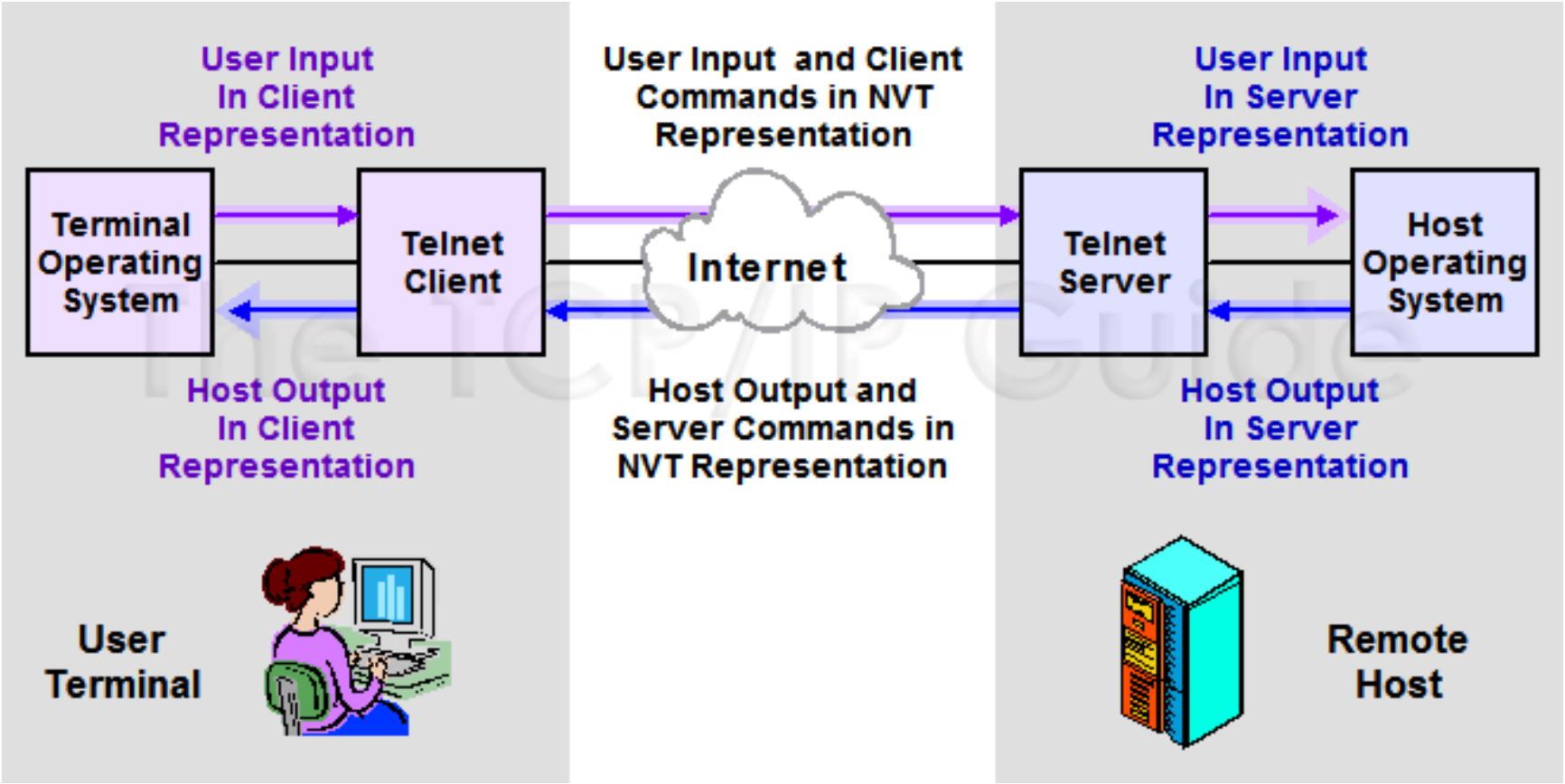
Simple Mail Transfer Protocol (SMTP)



SMTP - Mail processing model



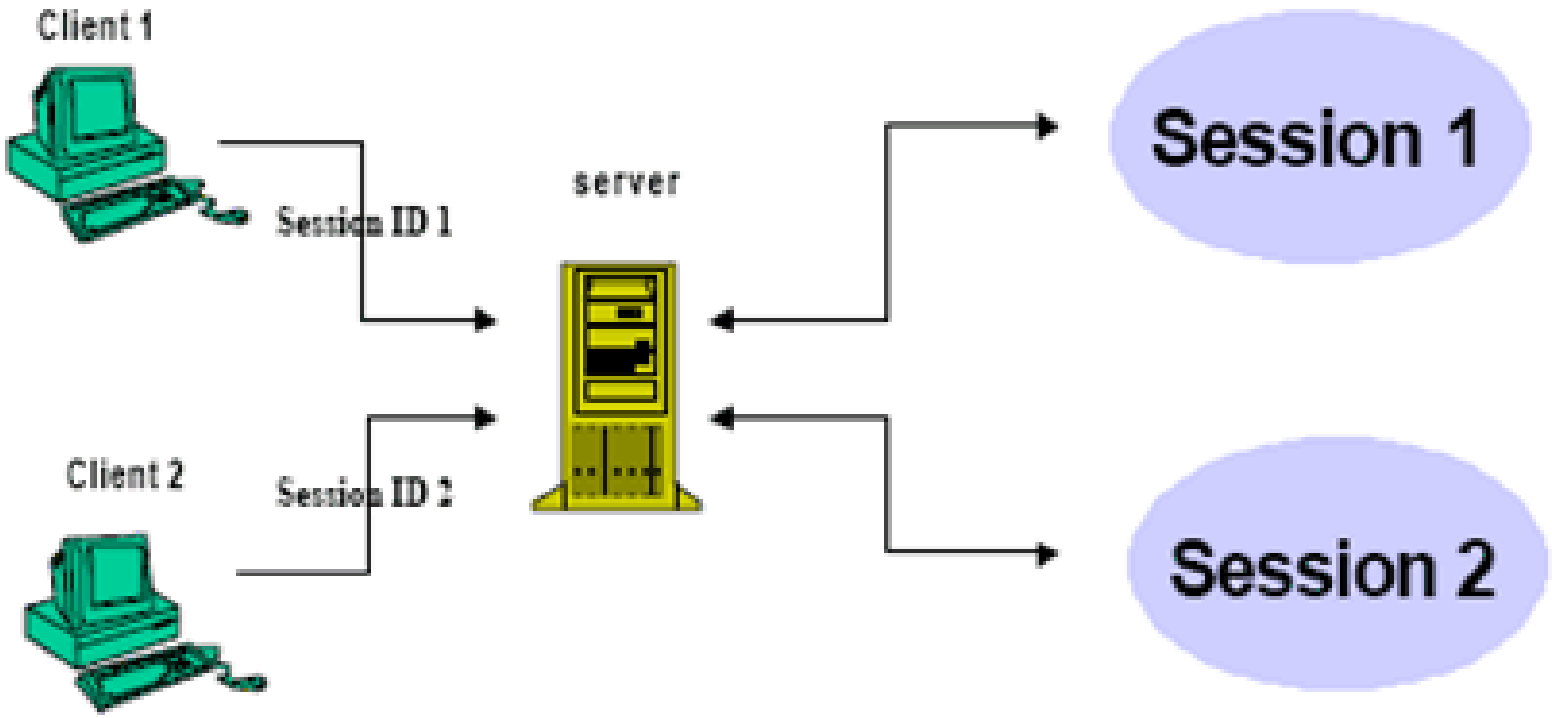
Telnet



Session/Cookie

- What is a "Session"?
- What is a "Cookie"?
- Setting and reading cookies
- Using Sessions

What is a "Session"?



HttpSession In Servlet

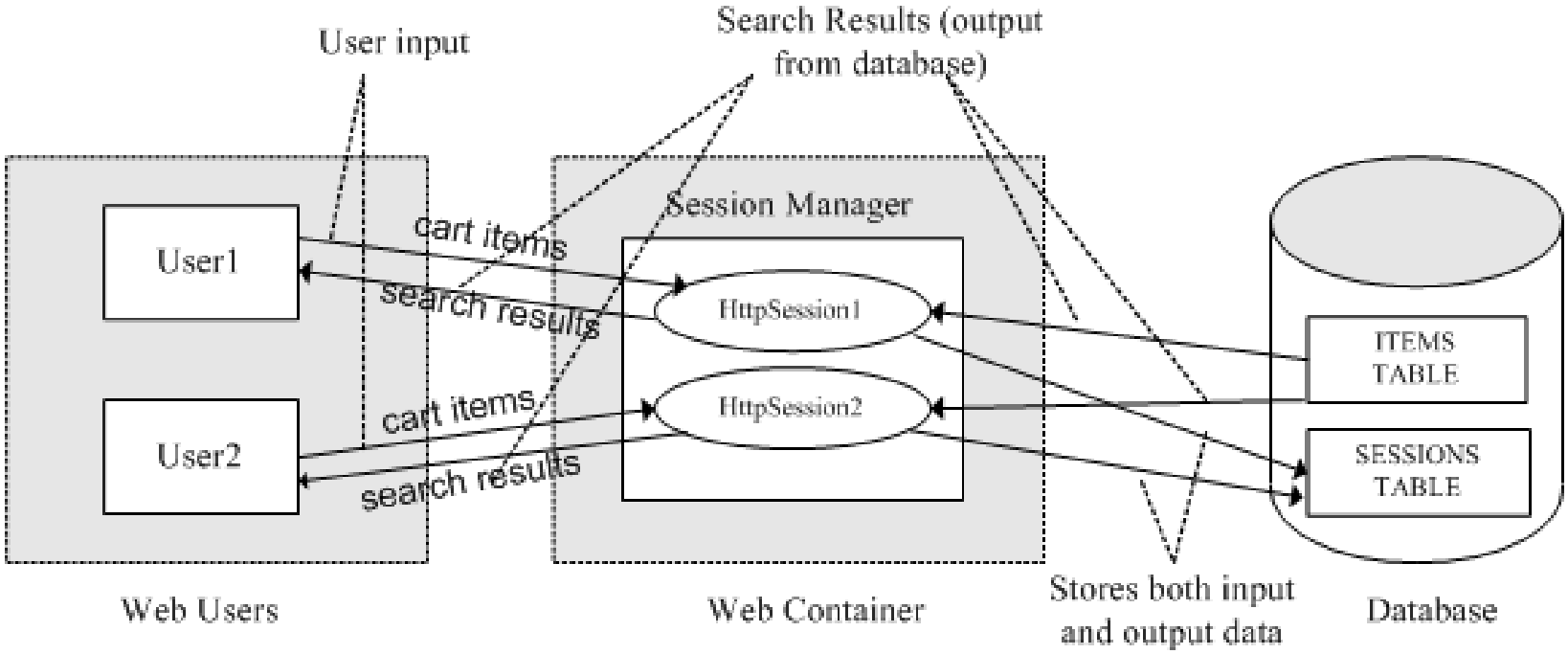
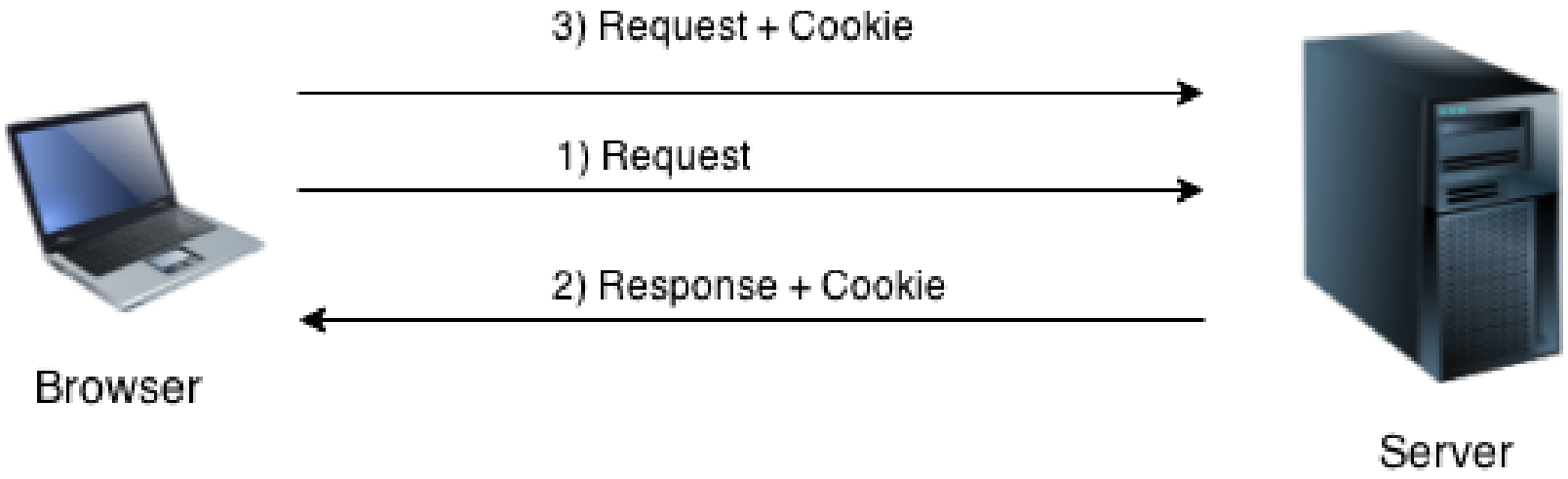


Figure 4.5 Session Manager stores HttpSession objects that contain both user input data and output data in its proprietary SESSIONS table

What is a "Cookie"?



Session vs Cookie

smartosc

Cookies	Sessions
Cookies are stored on client side	Sessions are stored on server side
Cookies can only store strings.	Sessions can store objects.
Cookies can be set to a long lifespan.	When users close their browser, they also lost the session.

Setting and reading cookies

```
Cookie ck=new Cookie("user","sonoo jaiswal");//creating cookie object
response.addCookie(ck);//adding cookie in the response
```

Setting and reading cookies - Constructor

smartosc

Constructor	Description
Cookie()	constructs a cookie.
Cookie(String name, String value)	constructs a cookie with a specified name and value.

Setting and reading cookies - Methods

smartosc

Method	Description
public void setMaxAge(int expiry)	Sets the maximum age of the cookie in seconds.
public String getName()	Returns the name of the cookie. The name cannot be changed after creation.
public String getValue()	Returns the value of the cookie.
public void setName(String name)	changes the name of the cookie.
public void setValue(String value)	changes the value of the cookie.

Using Sessions

- Creating or Accessing a Session
- Examining Session Properties
- Binding Data to a Session

Creating or Accessing a Session

smartosc

```
HttpSession mySession = request.getSession();
```

Examining Session Properties

smartosc

HttpSession Method	Description
`getCreationTime()`	Returns the session time in milliseconds since January 1, 1970, 00:00:00 GMT.
`getId()`	Returns the assigned session identifier. An HTTP session's identifier is a unique string that is created and maintained by the server.
`getLastAccessedTime()`	Returns the last time the client sent a request carrying the assigned session identifier (or `-1` if it's a new session) in milliseconds since January 1, 1970, 00:00:00 GMT.
`isNew()`	Returns a Boolean value indicating if the session is new. It's a new session if the server has created it and the client has not sent a request to it. This means the client has not acknowledged or joined the session and may not return the correct session identification information when making its next request.

Binding Data to a Session

smartosc

HttpSession Method	Description
`getAttribute()`	Returns the object bound to a given name in the session, or null if there is no such binding.
`getAttributeNames()`	Returns an array of names of all attributes bound to the session.
`setAttribute()`	Binds the specified object into the session with the given name. Any existing binding with the same name is overwritten. For an object bound into the session to be distributed it must implement the serializable interface.
`removeAttribute()`	Unbinds an object in the session with the given name. If there is no object bound to the given name, this method does nothing.

Overview of HTML

- HTML stands for Hyper Text Markup Language
- HTML describes the structure of Web pages using markup
- HTML elements are the building blocks of HTML pages
- HTML elements are represented by tags
- HTML tags label pieces of content such as "heading", "paragraph", "table", and so on
- Browsers do not display the HTML tags, but use them to render the content of the page

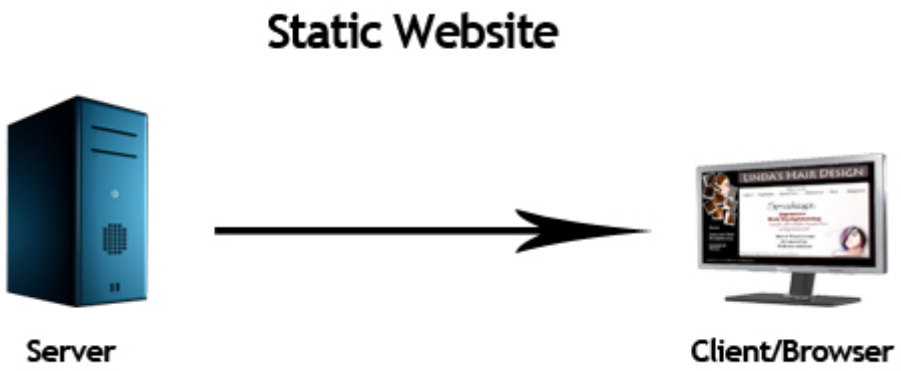
HTML - Example

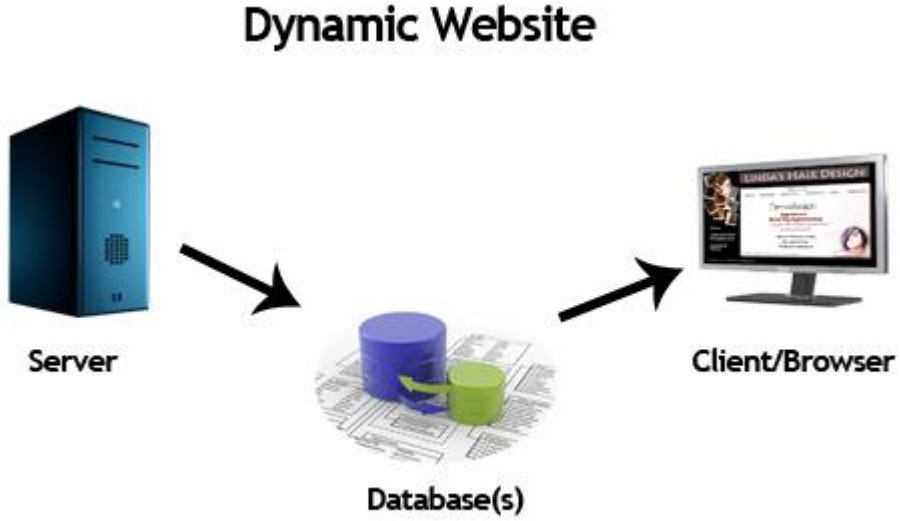
```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

Web Terminology

- Website: static vs dynamic.
- Get vs Post
- Container
- Server: Web vs Application
- Content Type

Website: static vs dynamic





Website: static vs dynamic

Static Website	Dynamic Website
Prebuilt content is same every time the page is loaded.	Content is generated quickly and changes regularly.
It uses the **HTML** code for developing a website.	It uses the server side languages such as **PHP,SERVLET, JSP, and ASP.NET** etc. for developing a website.
It sends exactly the same response for every request.	It may generate different HTML for each of the request.
The content is only changes when someone publishes and updates the file (sends it to the web server).	The page contains "server-side" code it allows the server to generate the unique content when the page is loaded.
Flexibility is the main advantage of static website.	Content Management System (CMS) is the main advantage of dynamic website.

Get vs Post

- Get
- Post

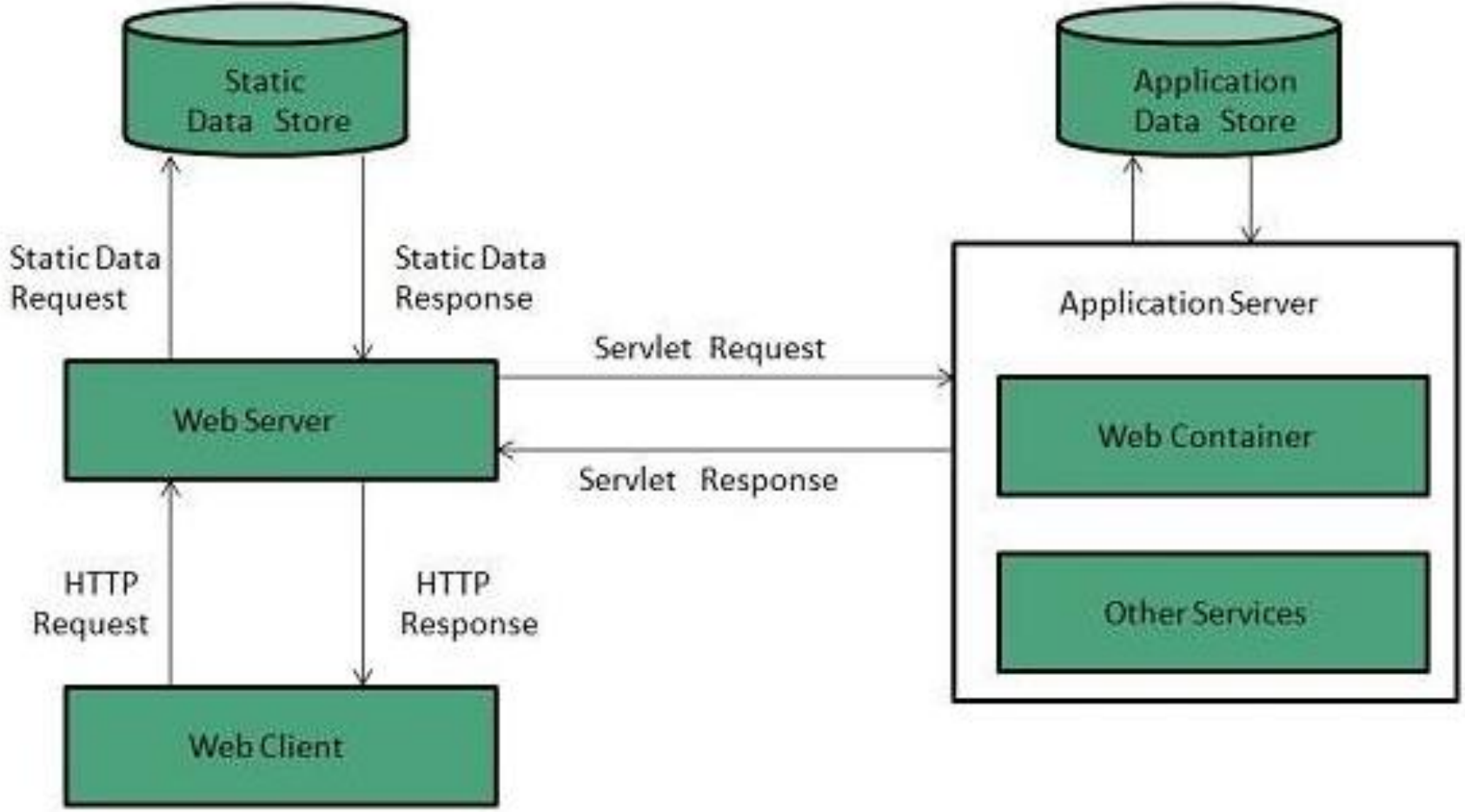
Get vs Post

GET	POST
1) In case of Get request, only **limited amount of data** can be sent because data is sent in header.	In case of post request, **large amount of data** can be sent because data is sent in body.
2) Get request is **not secured** because data is exposed in URL bar.	Post request is **secured** because data is not exposed in URL bar.
3) Get request **can be bookmarked.**	Post request **cannot be bookmarked.**
4) Get request is **idempotent** . It means second request will be ignored until response of first request is delivered	Post request is **non-idempotent.**
5) Get request is **more efficient** and used more than Post.	Post request is **less efficient** and used less than get.

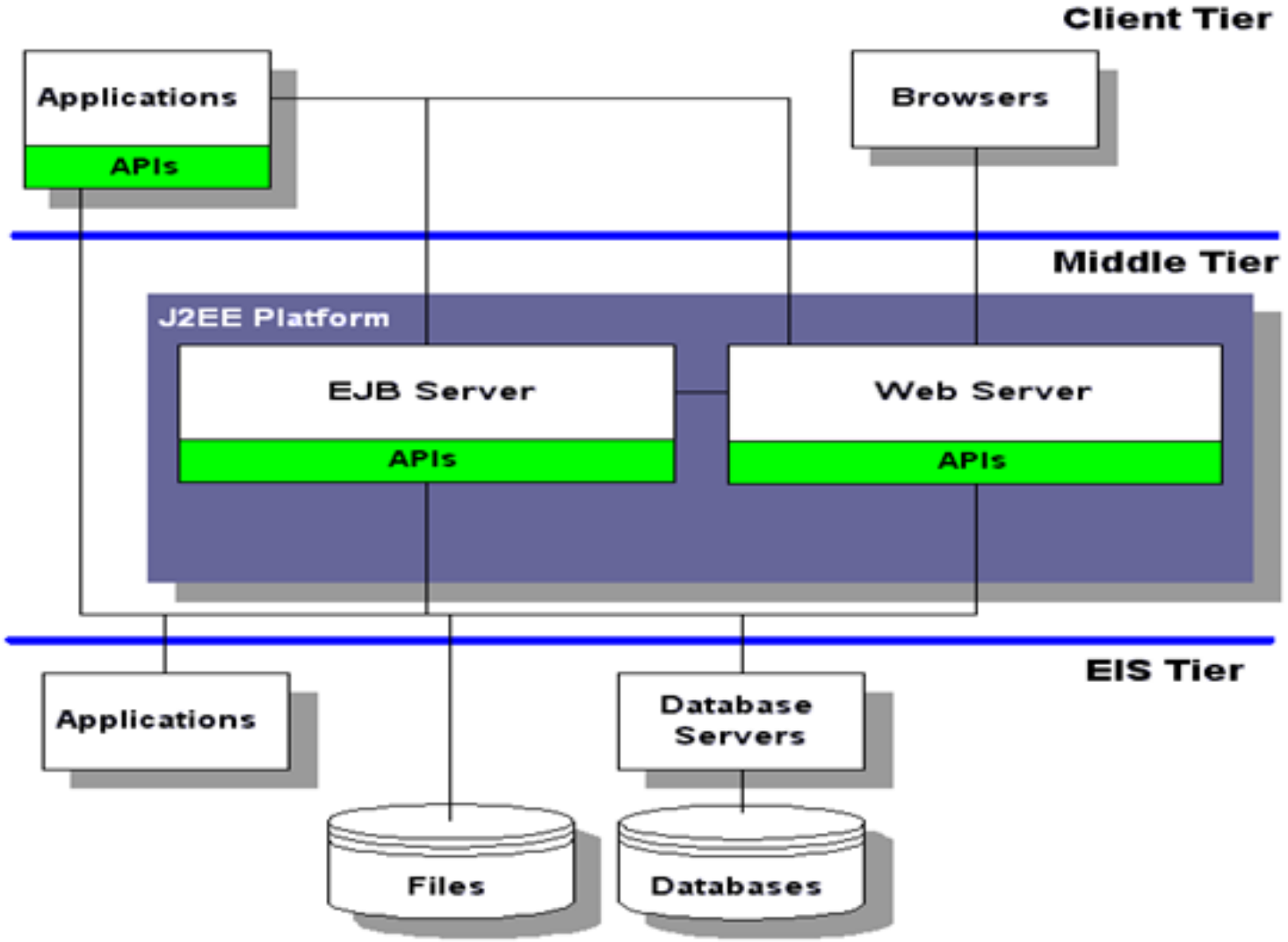
Server: Web vs Application

- Web Server
- Application Server

Web Server



Application Server



Content Type

- It supports the non-ASCII characters
- It supports the multiple attachments in a single message
- It supports the attachment which contains executable audio, images and video files etc.
- It supports the unlimited message length.

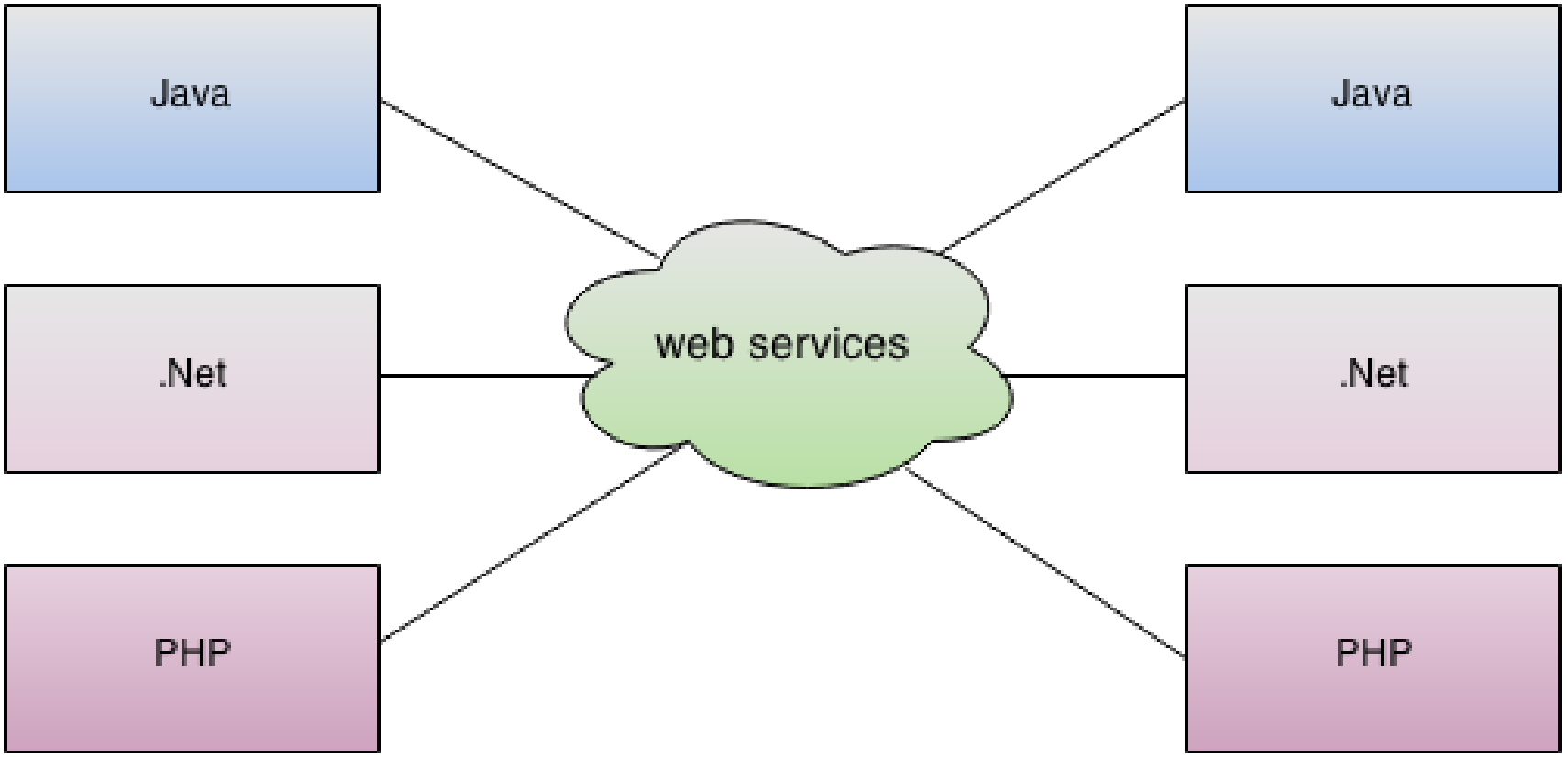
List of Content Types

- text/html
- text/plain
- application/msword
- application/vnd.ms-excel
- application/jar
- application/pdf
- application/octet-stream
- application/x-zip
- images/jpeg
- images/png
- images/gif
- audio/mp3
- video/mp4
- video/quicktime etc.

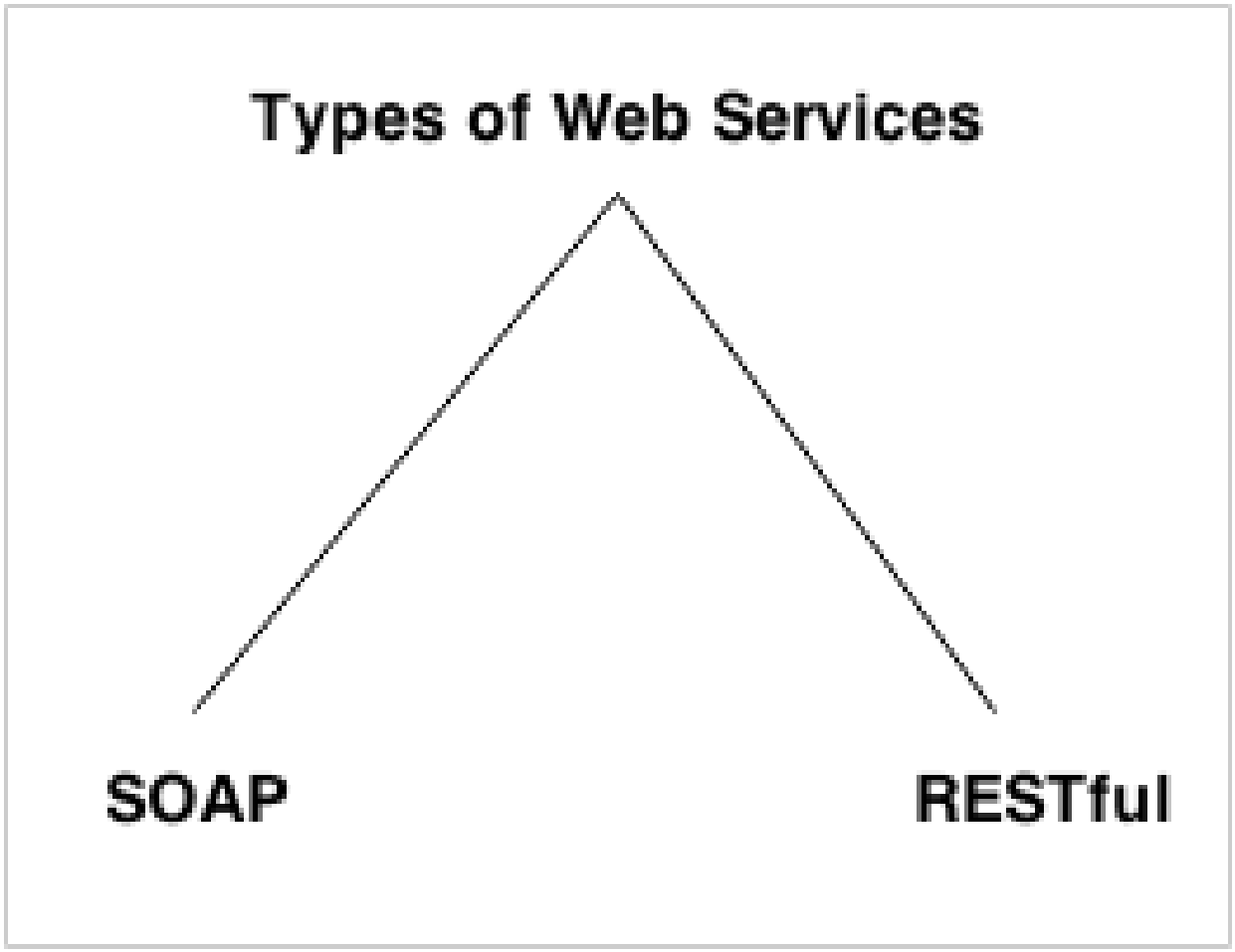
Webservice

- What are Web Services ?
- How does a web service work ?
- Web Service Architecture
- Web Service Components
- Web Service Security

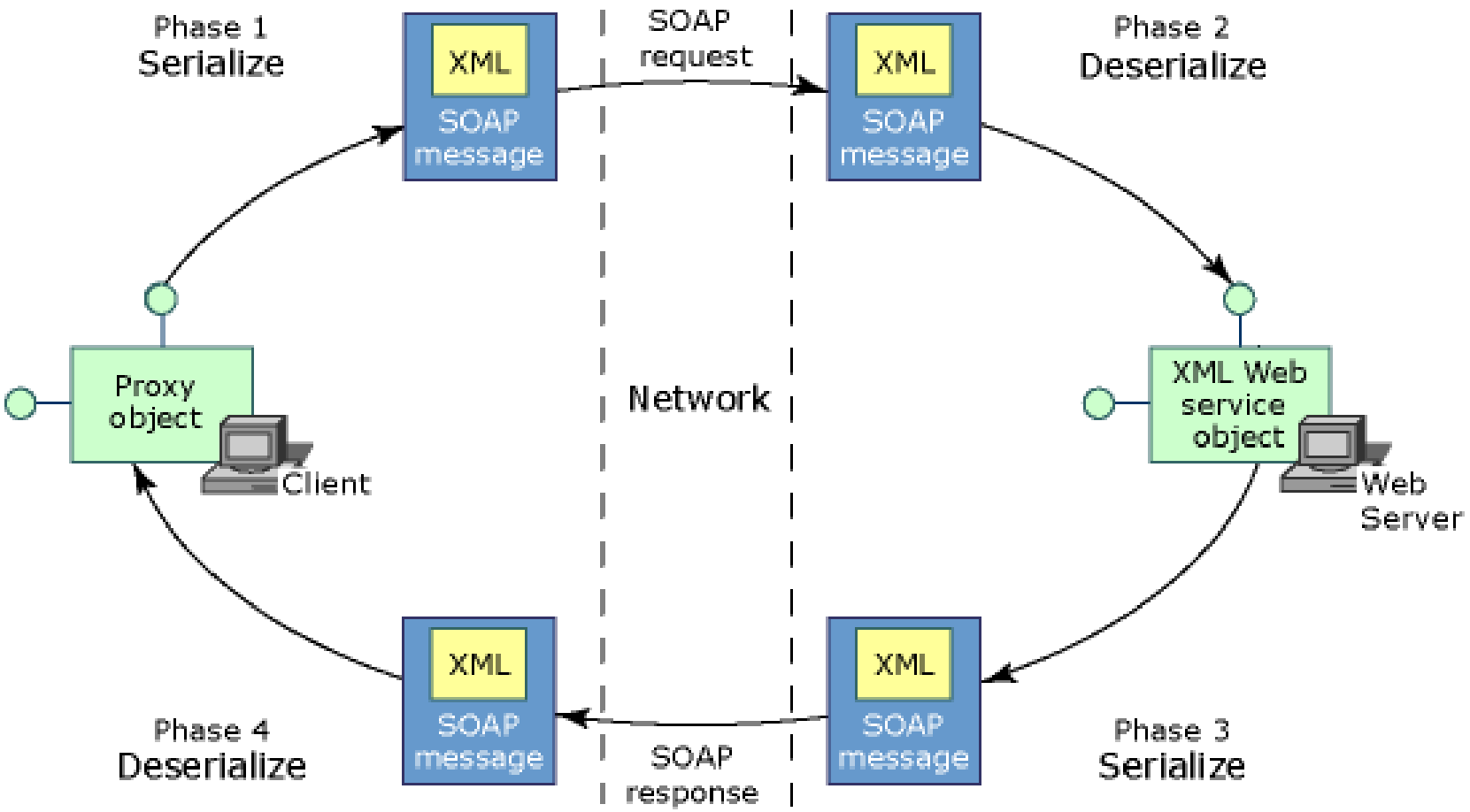
What are Web Services ?



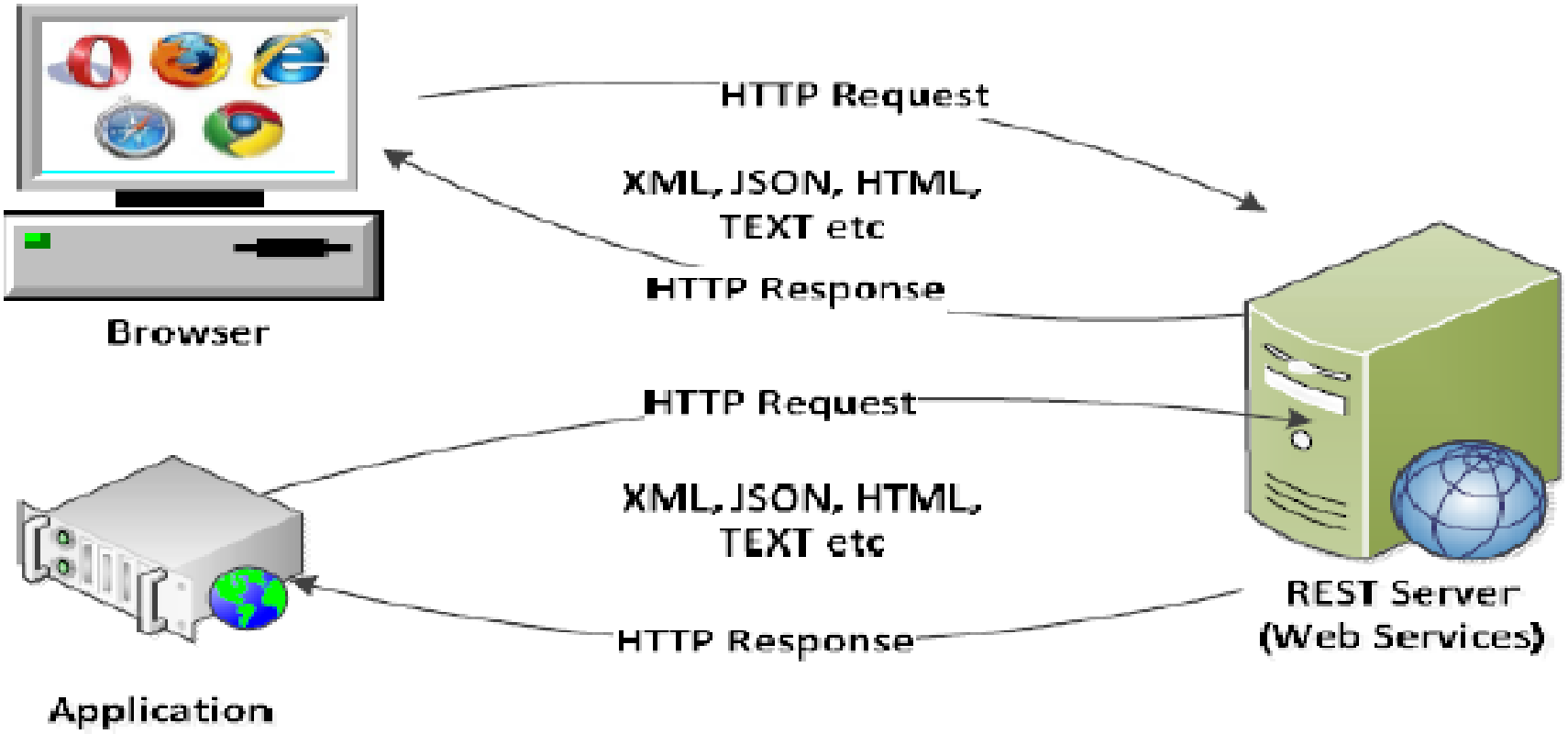
Types of Web Services



SOAP



RESTFul

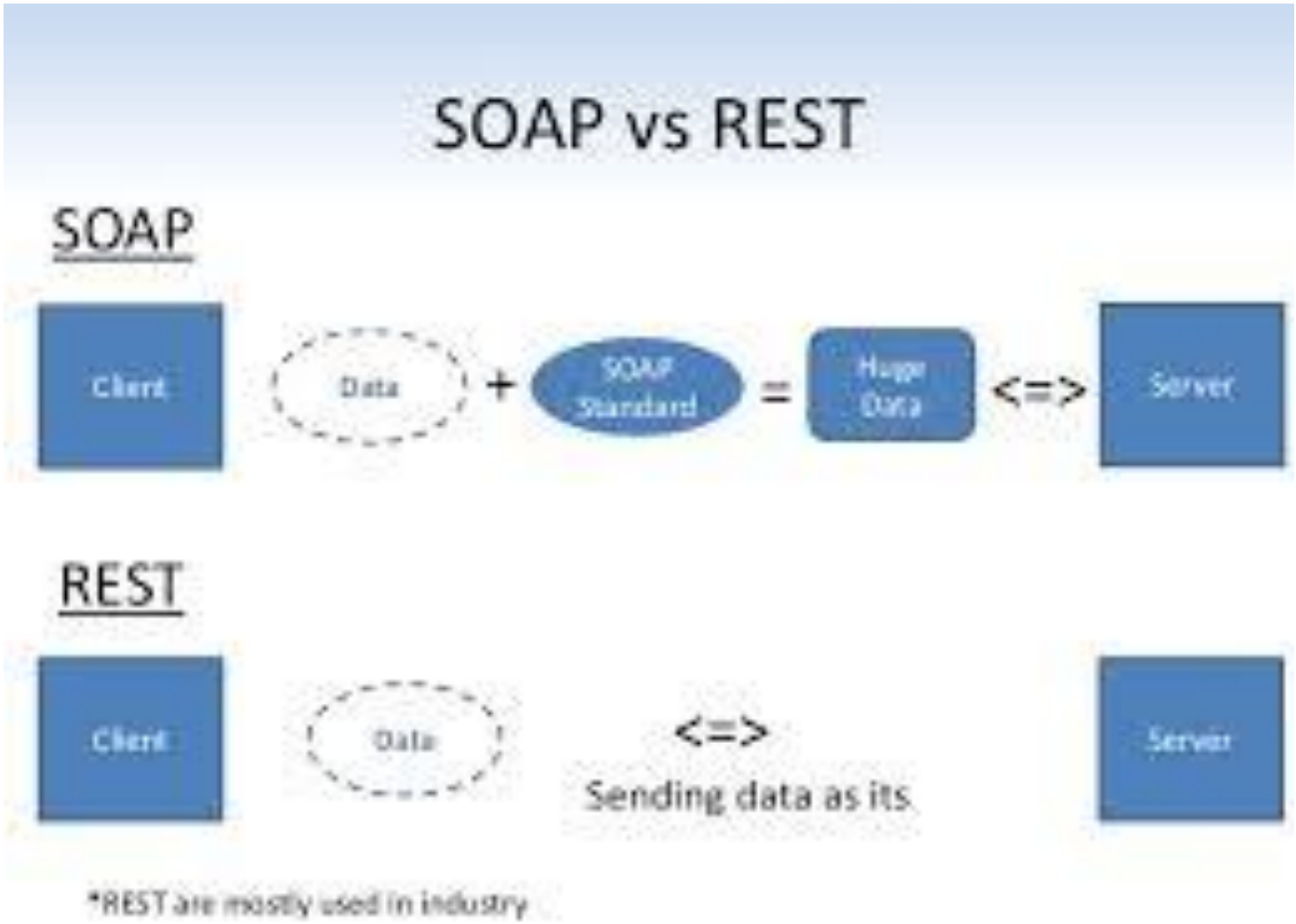


SOAP vs RESTFul

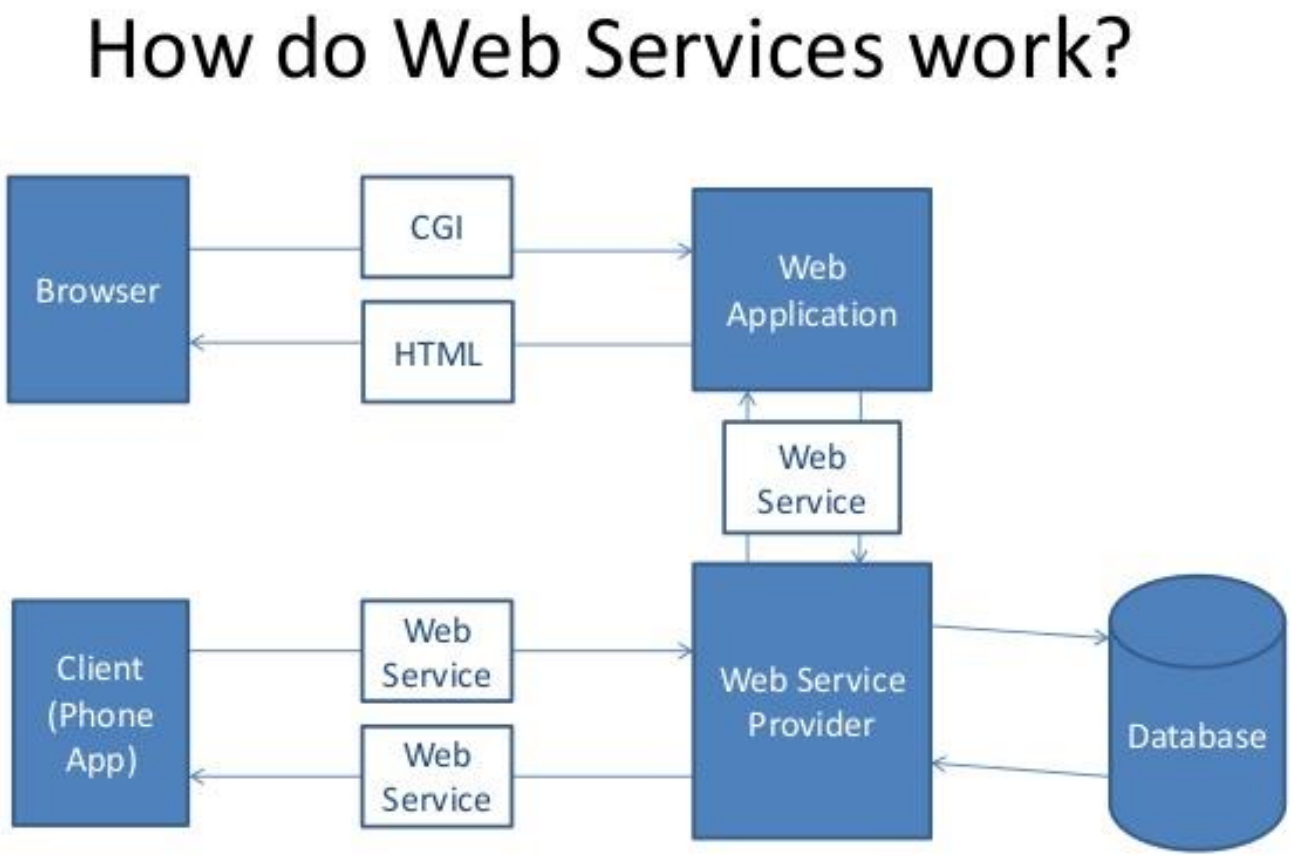
smartosc

#	SOAP	REST
1	A XML-based message protocol	An architectural style protocol
2	Uses WSDL for communication between consumer and provider	Uses XML or JSON to send and receive data
3	Invokes services by calling RPC method	Simply calls services via URL path
4	Does not return human readable result	Result is readable which is just plain XML or JSON
5	Transfer is over HTTP. Also uses other protocols such as SMTP, FTP, etc.	Transfer is over HTTP only
6	JavaScript can call SOAP, but it is difficult to implement	Easy to call from JavaScript
7	Performance is not great compared to REST	Performance is much better compared to SOAP - less CPU intensive, leaner code etc.

SOAP vs RESTFul



How does a web service work ?

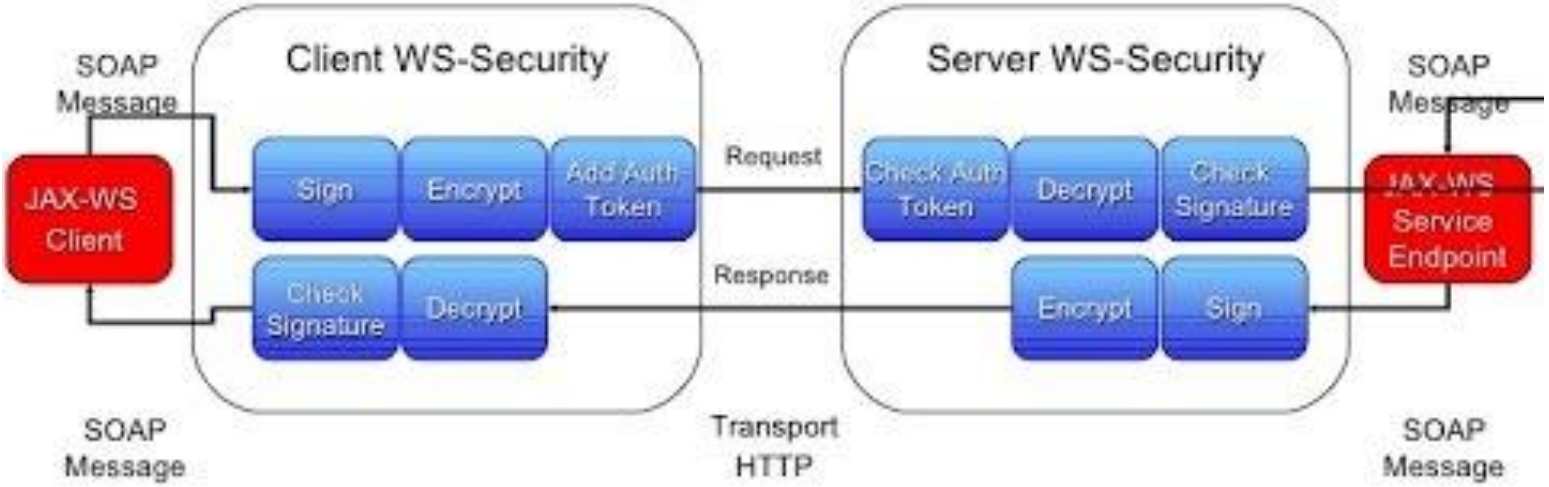


Web Service Components

Web Service Security

smartosc

Feature : Web Service Security



- Access Control(Aucn/Auzn)
- Message Confidentiality
- Message Integrity

- OOTB security policies
  - WLS policies
  - OWSM policies

ORACLE

Q & A