Java EE

Lecture 1

Main Presentation

Master ADEO M2 course

Course Overview

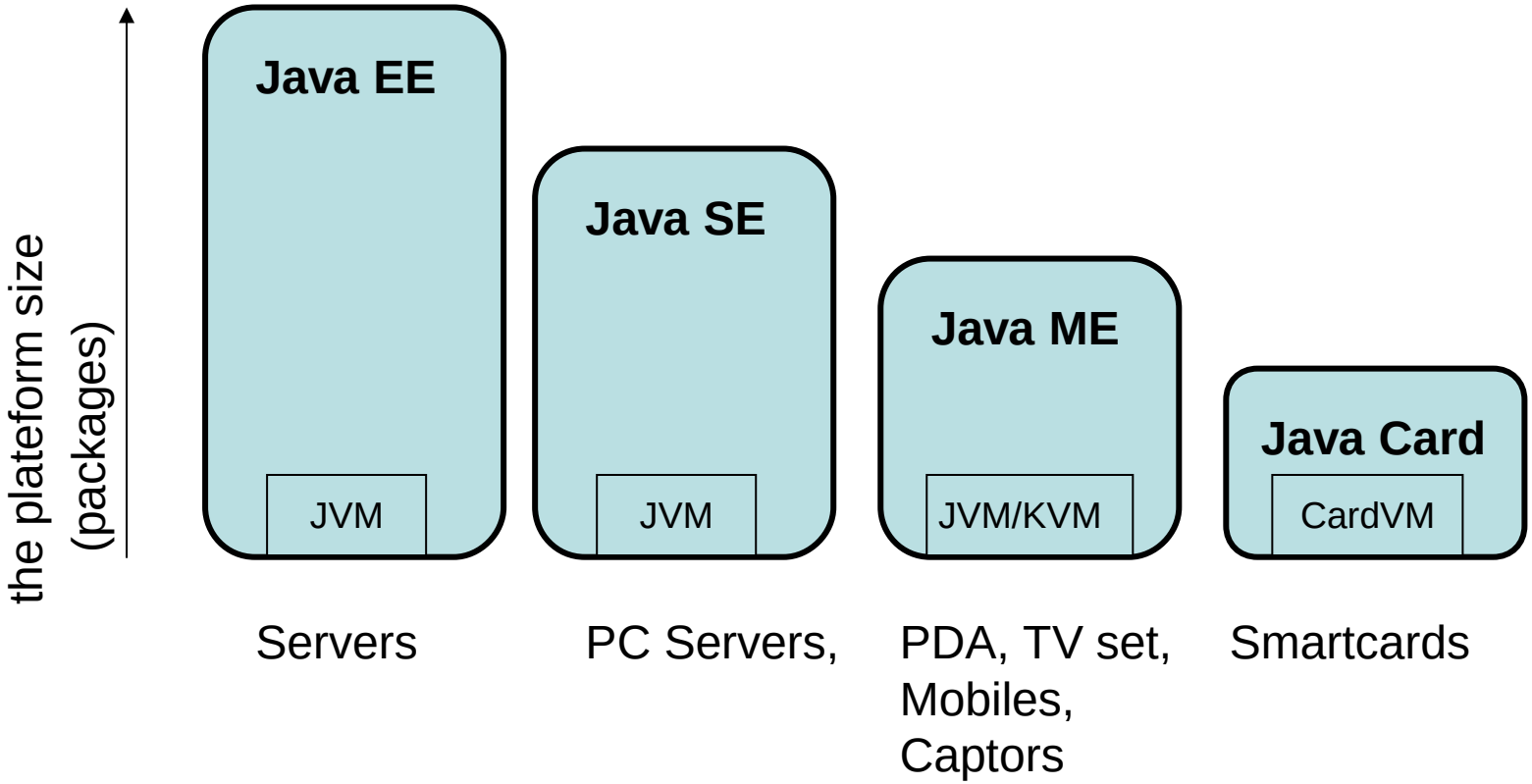
- Objectives
  - Development of robust web applications
    - «Do not reinvent the wheel» → using a standard framework [1]
  - Learning a part of Java EE
    - Servlet, JSP et EL/JSTL
- Prerequisites
  - Masters of Java language (Java SE)
  - Mastery of web development Client
    - XHTML (at least structural tags and Forms)
    - CSS and XML are more
  - Bases Network (Client /Server Architecture)

[1] In OOD using Design Patterns

Java EE?

- Java Enterprise Edition is a framework
  - rich (Java SE + severals API)
  - open (specs. of Java Community Process)
  - dedicated to the development, deployment and implementation of modern Internet applications (necessary for companies)
- Favors the separation of concerns
  - business code vs. Non-functional properties
    - QoS, persistance (JPA), administration (JMX), security, transaction (JTS/JTA),…

The Java galaxy



- Editing does not necessarily include the whole of a smaller edition

Web development

- Le World Wide Web
  - Universal and public SI deployed on Internet
  - A language : HTML (or XHTML)
  - A communication model : client-server
  - A protocol :HTTP
- Static pages
  - HTML pages prepared in advance
  - The server returns pages without making any special treatment
- Dynamic pages
  - Please do not confuse this with an animated page (ex. Flash)
  - HTML pages generated by the **server**
  - The server builds the response according to the request of the user

Server

• Server : a computer with a number of resources that it makes available other computers (clients) via the network.

• Kinds of servers :
    • Web server
    • Application Server
    • ...

Web server

- Program running on a machine connected to the Internet

- HTTP protocol: responds to the client (customer) requests (web browser) and and processes it

- Returns HTML pages to the **Customer**

Java and the web development

- Several Java technologies allow make the web development at different levels

	Development	
	Client side	**Server side**
Autonomous .class	applet	servlet
Java source code mixed with html	JavaScript	JSP

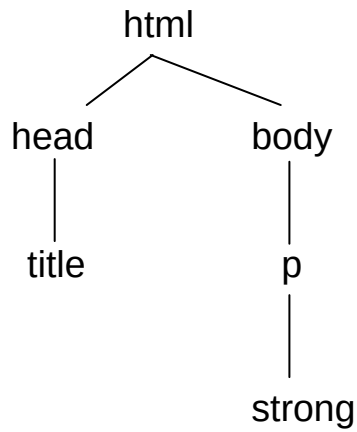
Static pages

Dynamic pages

HTML (reminder)

- Markup Language, nonproprietary (W3C)
- Designed to display documents on the Web
- Possible links between documents
- XHTML now provides compatibility with XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
 <head>
  <title>XHTML 1.0 valid   !</title>
 </head>
 <body>
  <p>The XHTML 1.0 page <strong>valid</strong>.</p>
 </body>
</html>
```



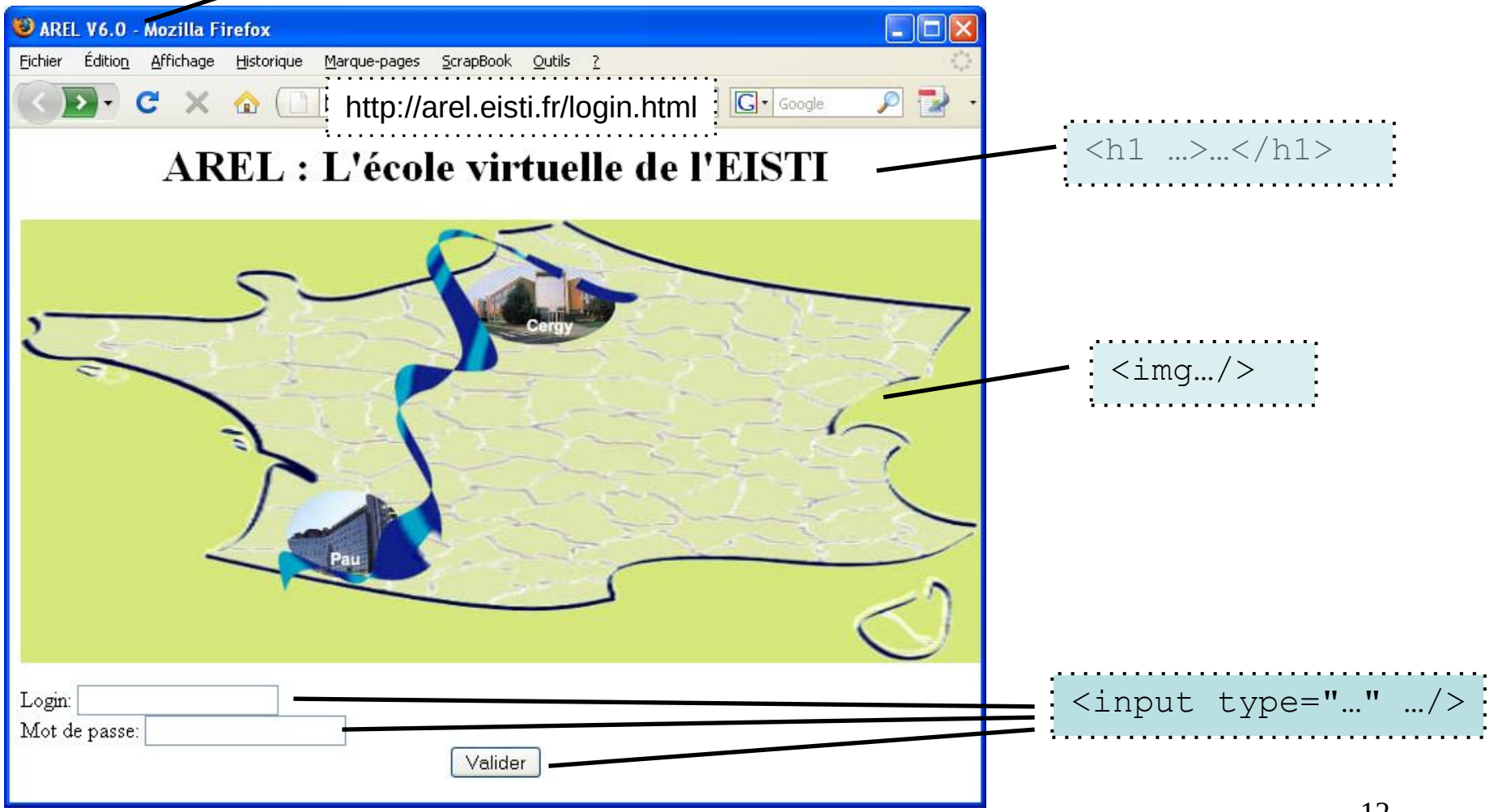
some tags (markers)

- `<!--  -->` comments
- `<a>` ancre (hyperlink href)
- `<body>` body of the document
- `<br>` *line break*
- `<form>` input form
- `<h1>` Large font as a title1
- `<head>` header
- `<html>` limits the document
- `<input type>` buttons and input fields
- `<p>` paragraph
- `<title>` title

Ex : AREL V6 (HTML code)

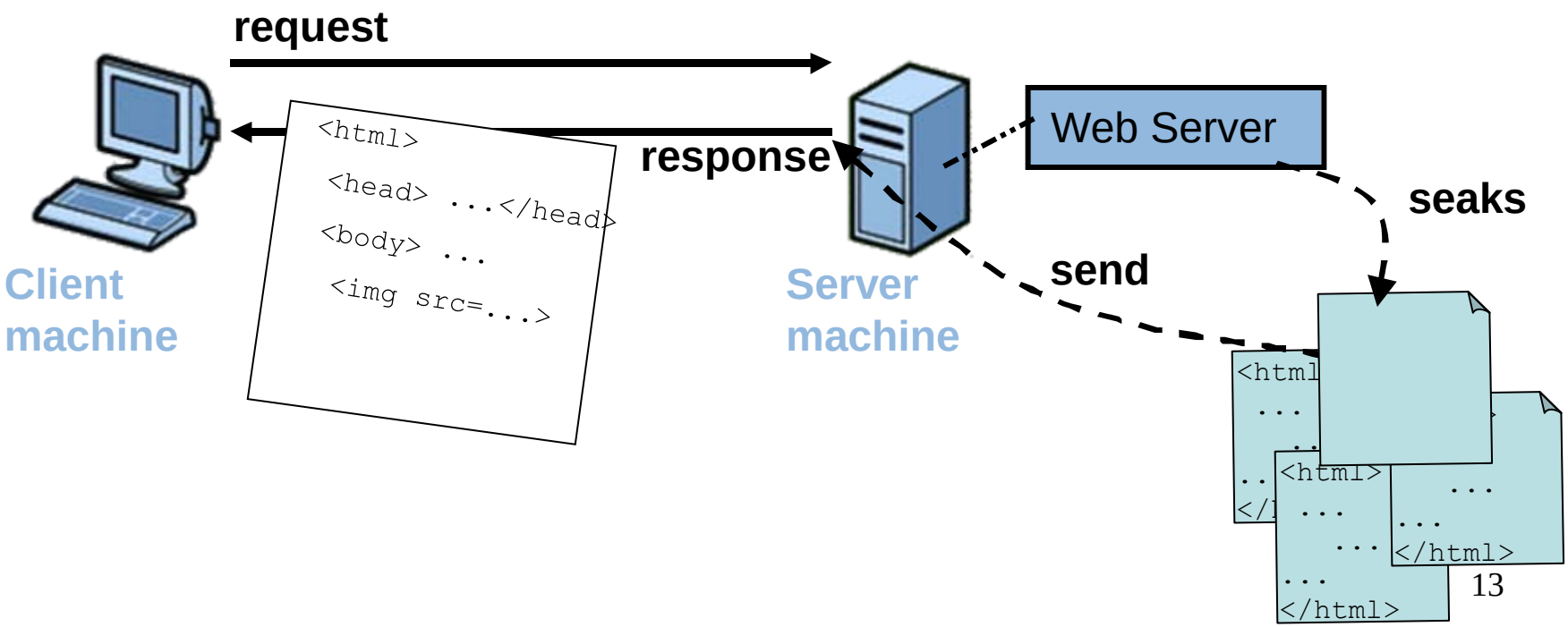
```
<html>
<!-- AREL V6.0 -->
<head>
  <title>AREL V6.0</title>
</head>
<body>
  <h1 align="center">AREL:L'école   virtuelle de l'EISTI</h1>
  <p><center>
    <img src="fondcergypau.jpg"/>
  </center></p>
  <form action="date2">
       Login: <input type="text" name="param1"/><br/>
Mot de passe: <input type="password" name="param2"/><br/>
   <center>
     <input type="submit" value="Valider"/>
   </center>
  </form>
</body>
</html>
```

Ex : AREL V6 (displaying)



Static Pages

- Normal operation of a single Web server
  - The Server seeks the page in the file system
  - The page is returned to the client as it is



Limits of a single Web server

- No dynamic content

```
<html>
  <head>
    <title>Clock</title>
  </head>
  <body>
    It is always 12:12.
  </body>
</html>
```

static

VS.

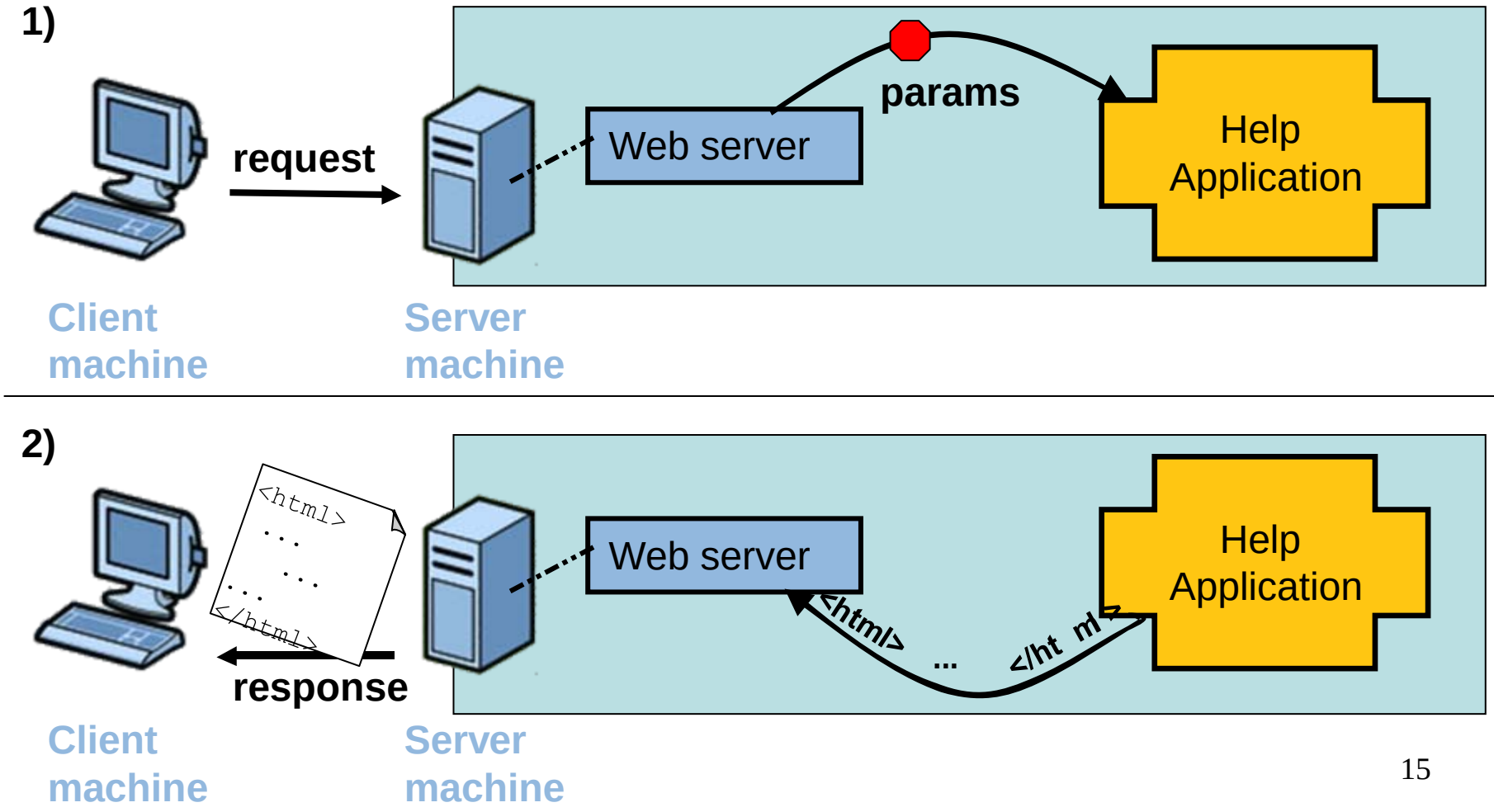
```
<html>
  <head>
    <title>Clock</title>
  </head>
  <body>
    It is[getTimeOnServer].
  </body>
</html>
```

dynamic

- No save data on the server
  - Forms processing:
    the Web server needs a help application
    - to evaluate the received parameters
    - to genereate an appropriate response

Dynamic pages

- The Web server needs help to make dynamic
  - In the past, the CGI (Common Gateway Interface)
  - Now, a Java EE container with servlets (ex: Tomcat)



What is Java EE

- Java EE is a platform a highly oriented server to the development and execution of distributed applications. It is composed of two essential parts :

  - A set of specifications for infrastructure in which runs components written in Java : this environment is called **application server**

  - A set of APIs that can be obtained and used separately. To be used, some require implementation by a third party provider.

  - A set of APIs that can be obtained and used separately. To be used, some require implementation an alternate provider

Java EE APIs

• API (*Application Programming Interface*)
is a programming interface. It is a set of functions, procedures or classes made available to computer programs by a software library, operating system or service.

• Les components **:** allows the cutting application and thus a separation of roles during development:

- The web components : **Servlet** and **JSP**(Java Server Pages).
- The Business components : EJB (Enterprise Java Beans).

- The **services :**
  - The **infrastructures services** : JDBC, JNDI, JTA, JCA, JMX
  - The **communication services** : RMI-IIOP, JavaMail, JAAS

Java EE Architecture

- JEE allows great flexibility in the choice of the architecture of the application by combining the various components.

- The application architecture is divided ideally into at least three third :
  - **The cliente part**: allows communication with the user. It can be composed by:
    - stand-alone application
    - web application
    - applets
  - The business part : encapsulates treatments(in EJB orJavaBeans)
  - **The datas part** : save the data

Servlets and JSP

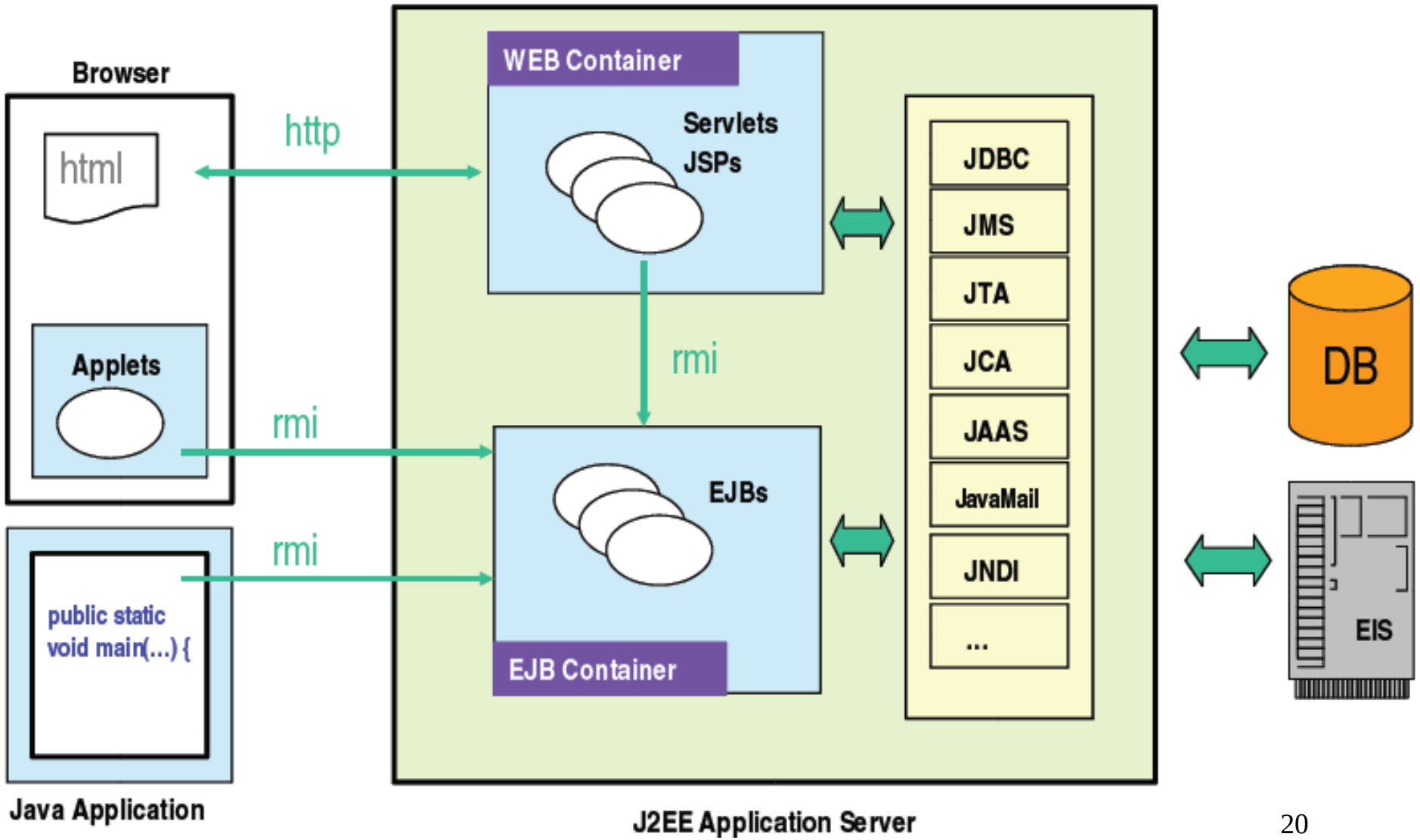
• In order to achieve dynamic Web applications we will achieve 2 large type " JEE pages" :

- The **Servlets** : which are specific Java classes that can be run on a JEE server. The main method of these classes is called at each client request and receive as parameter a submitted query. After treatment (in the body of the method), it will then return to the client the generated HTML page

The JSP : that have the same purpose as Servlets but with a syntax closer to the HTML (similar to PHP).

• These two kinds of programming can be used jointly or independently according to the application to realize

Java EE Architecture



Container

• Containers ensure the management of lifecycle of components that run inside them. Containers provide services that can be used by applications when they are running.

• The concept of container is found in many technologies :
  – Servlet, Applet, MIDlet, Xlet, (*-*let*), EJB, …

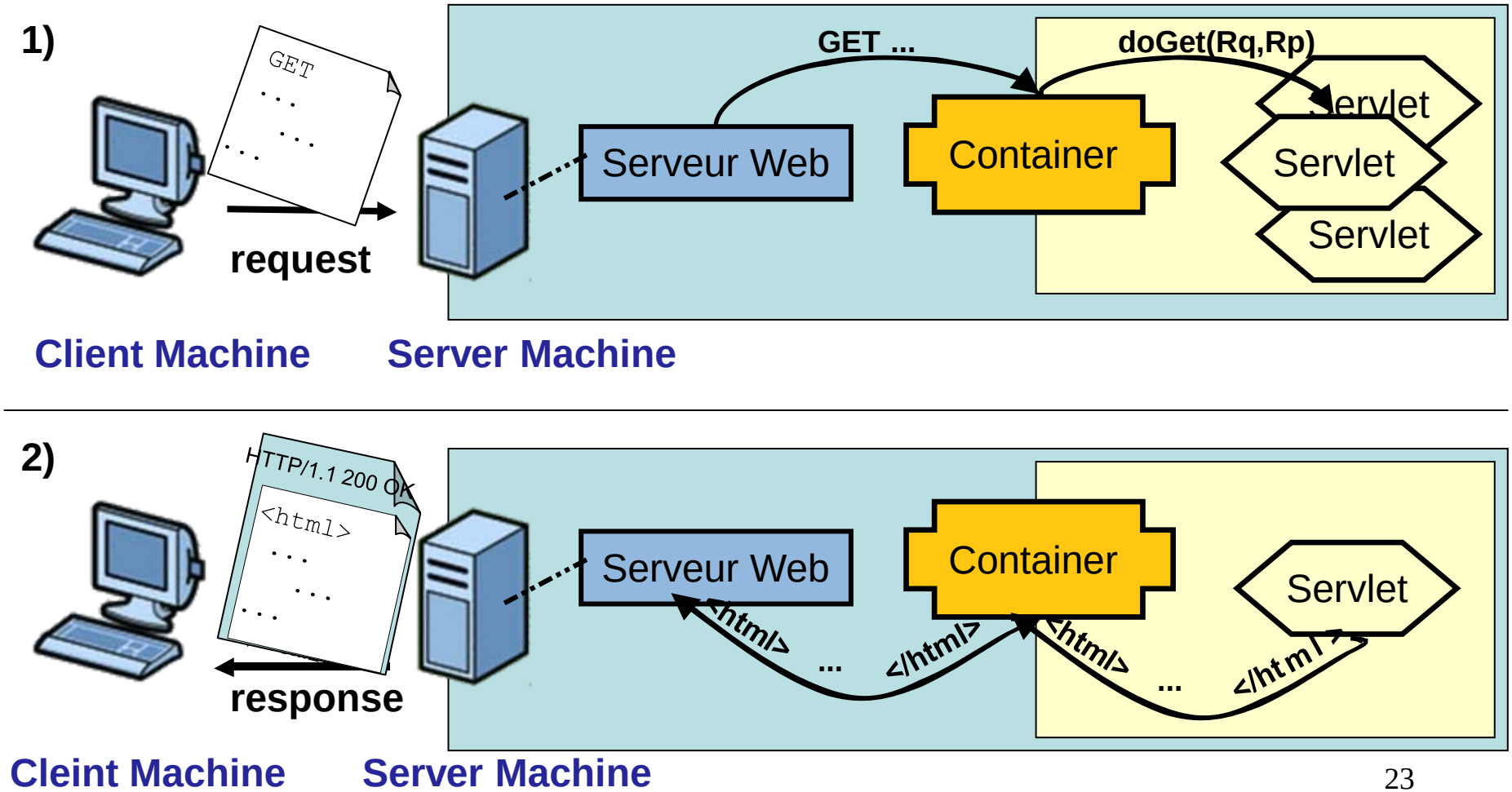
• There are several containers defined by JEE :
  – Web container: To run Servlets and JSPs
  – EJB container: To run the EJB
  – Client container: to run stand-alone applications on stations that use JEE components

Container

- A container is a *system* software component that controls other system components, called business
  - Tomcat is an example of container
  - Servlets have no main () method, they are controlled by the Tomcat container,
  - he requests are not sent to the servlet but the container in which they are deployed

Web application with a container

- he Web server needs help to make dynamic



Why a container?

- To forget the course of « Network » !
- A container provided for Servlets :
  - A supporting for communication
    - No need to ServerSocket, Socket, Stream, ...
  - The management lifecycle
  - A support for Multithreading
    - Automatic creation of Threads
  - A safety support
  - Support for JSP

Web module

- A servlet can not be directly deployed in a container, it must be part of a Web module.

- A Web module is a set of libraries, configuration files, Java code (bytecode of servlets …), …

- The Web module is the unit of deployment in container.

Web Module

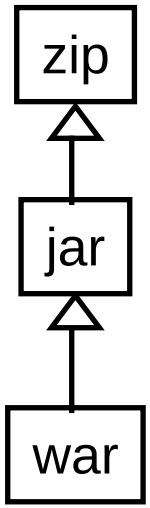
• To deploy an application in a container, it is necessary to provide two elements:

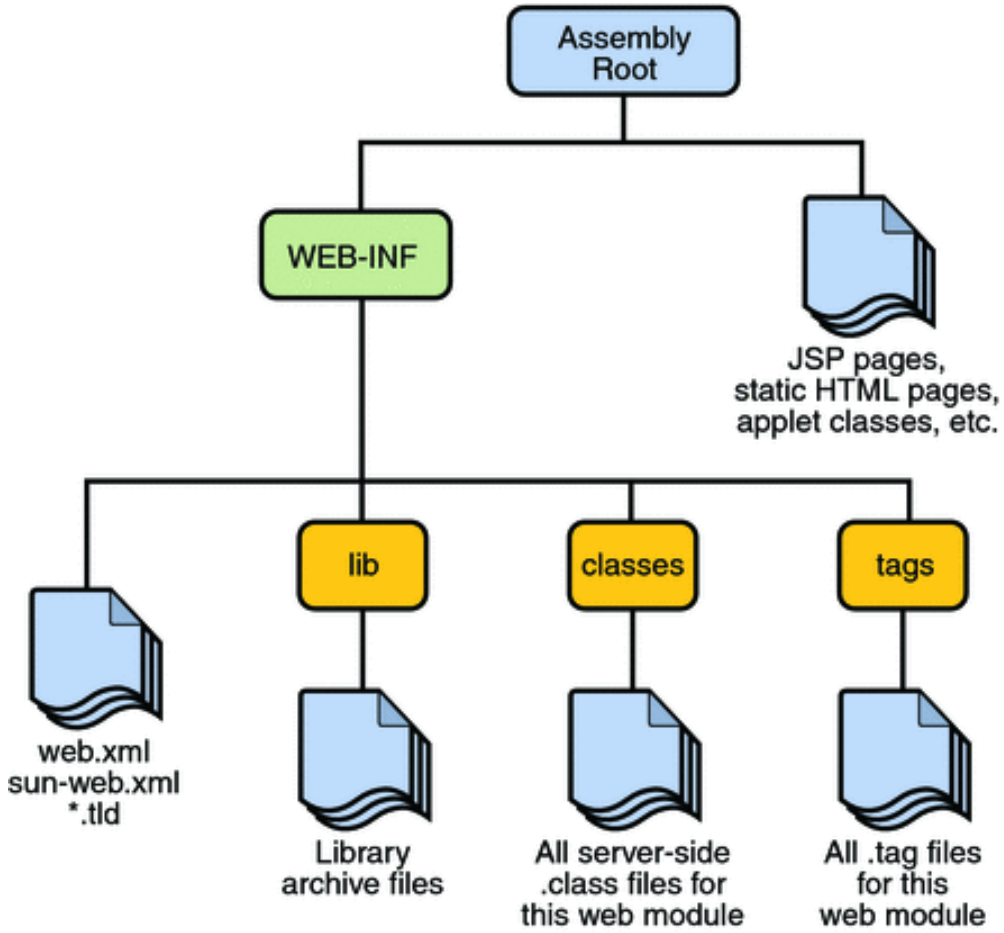
  • The application with all components (compiled classes, resources ...) grouped in an archive or module. Each container has its own archive format.

  •A deployment descriptor file contained in the module that specifies to the container options to run the application

Structure of a web module (.war)

- Automated in Eclipse
  - File/ Export…
    Web/WAR file





Different kinds of archives

Archive / module	Content	Extension	Deployment Descriptor
library	Regroups classes	jar	
Client application	Regroups the necessary resources to them running(classes, bibliothèques, images, ...)	jar	application-client.jar
web	Regroups servlets , JSP and the required resources to them running (classes, tag libraries, images , ...)	war	web.xml
EJB	Regroups the EJB and its components (classes)	jar	

An application is a combination of one or more modules in an EAR (Enterprise ARchive) file. The application is described in a application file .xml, itself is contained in the EAR file

Application Server

- Application servers can provide :
  - Only a web container (eg Tomcat) or
  - Only an EJB container (eg JBoss, Jonas, ...) or
  - Both containers (eg Websphere, Weblogic, ...).

The services offered by the JEE platform

A jee full executing platform, implemented in an application server, proposes the following services :

- naming service
- deployment service
- the service of transaction management
- security service)

These services are directly or indirectly used by the container but also by the components that run in the containers thanks to their respective API.

Development Environments

- The Development-Deployment-Execution cycle is too complex to your taste ?

- The IDE (*integrated development environment*) are here to assist you !

. . .