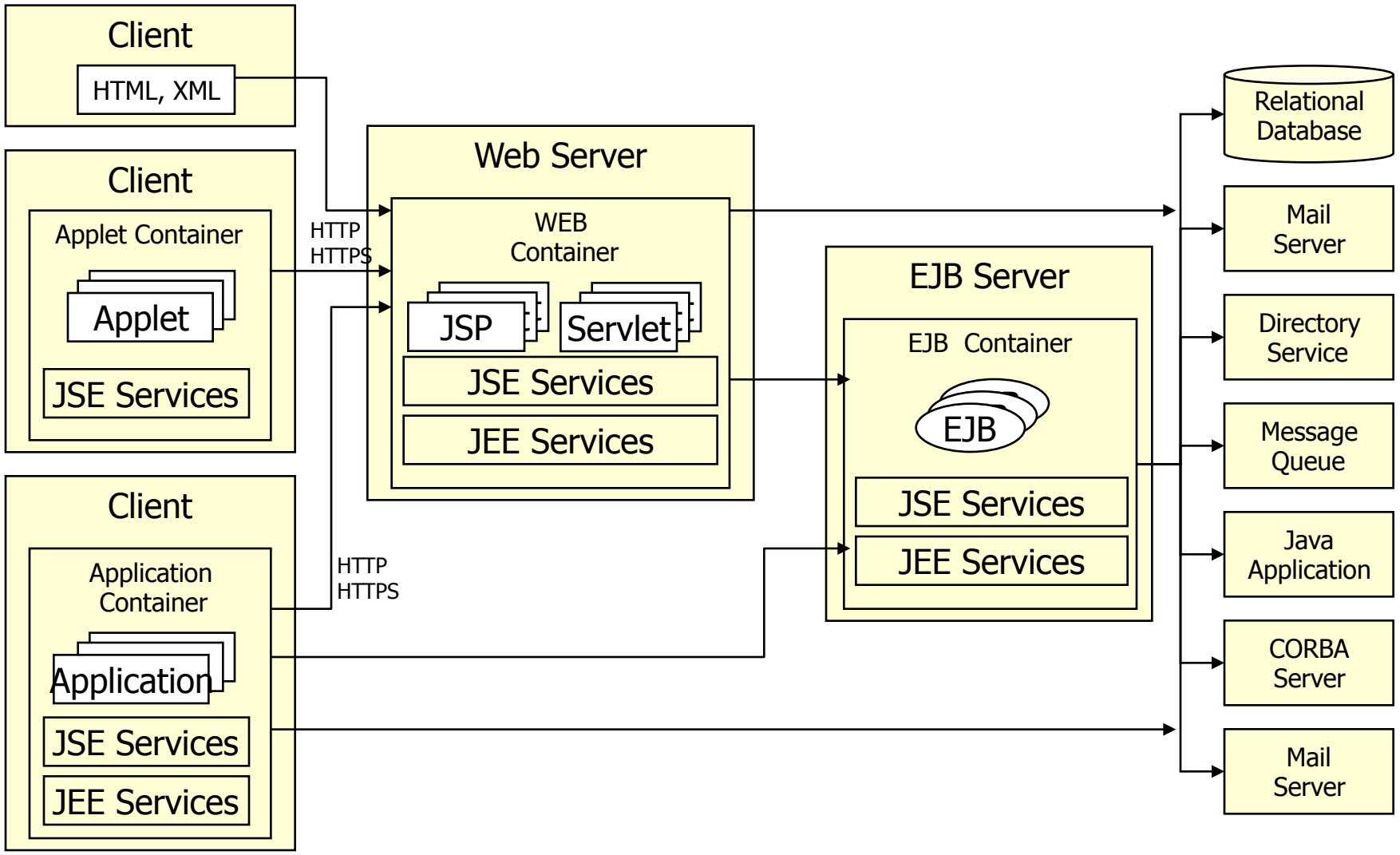
Architecture des Systèmes d' Information

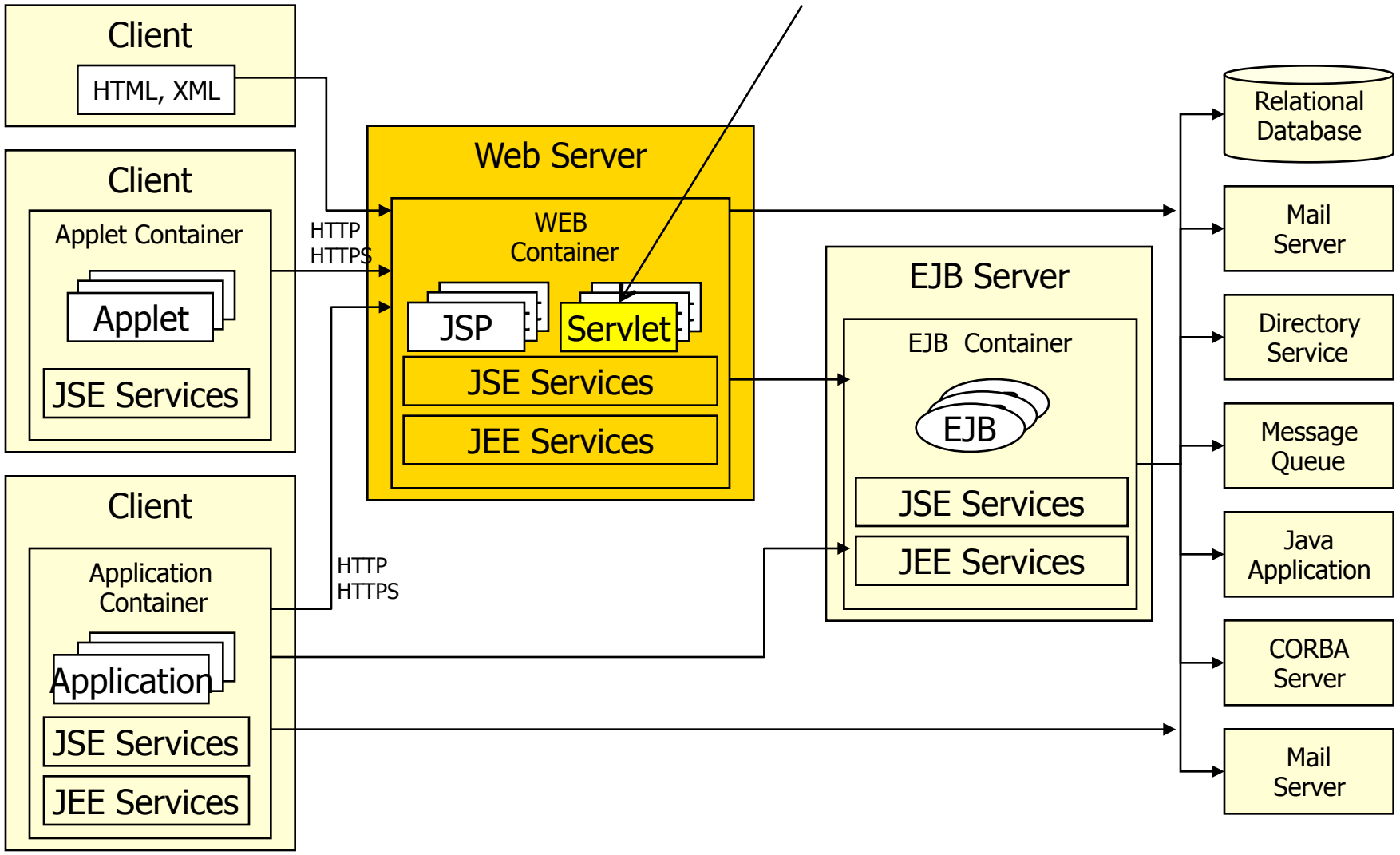
Servlet

Traduction en cours

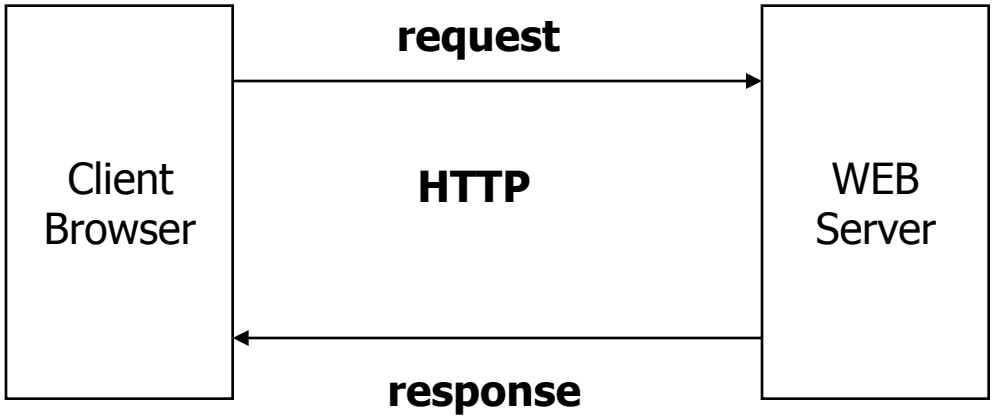
Web Server Enterprise Components



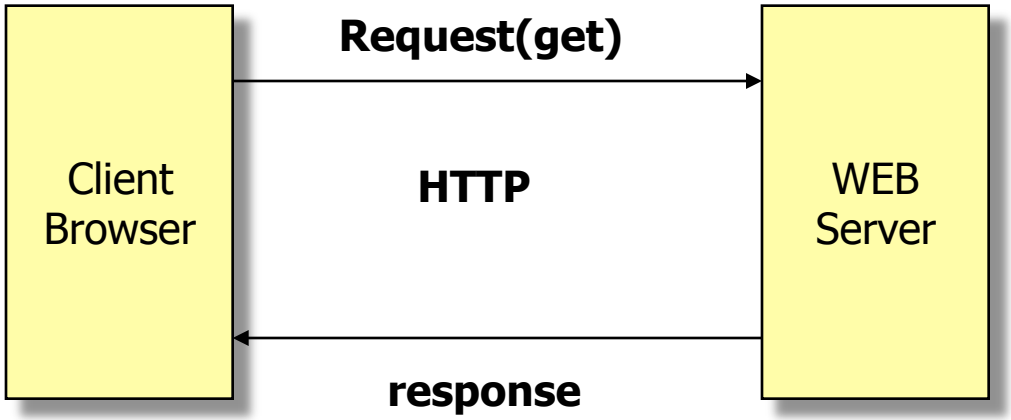
Servlet


Client
HTML, XML
Client
Applet Container
Applet
JSE Services
Client
Application Container
Application
JSE Services
JEE Services
HTTP
HTTPS
HTTP
HTTPS
Web Server
WEB Container
JSP
Servlet
JSE Services
JEE Services
EJB Server
EJB Container
EJB
JSE Services
JEE Services
Relational Database
Mail Server
Directory Service
Message Queue
Java Application
CORBA Server
Mail Server

HTTP request and response

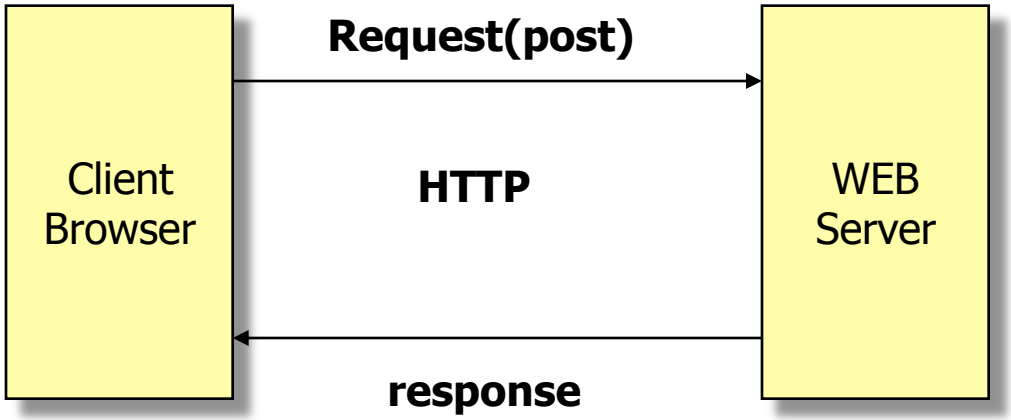


HTTP request and response



HTTP 0.9 uniquement GET

HTTP request and response

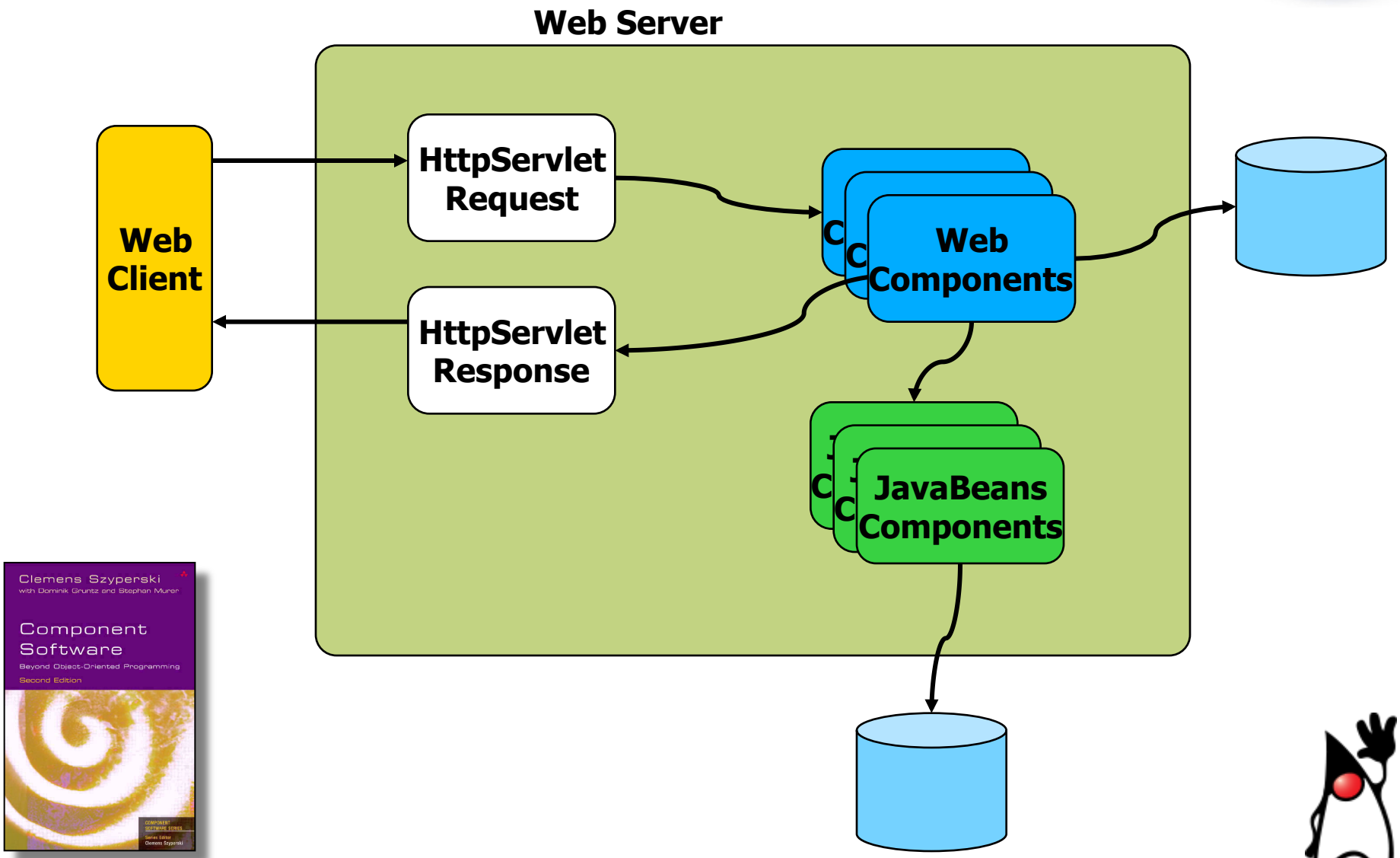


HTTP 1.0 (DATE) :

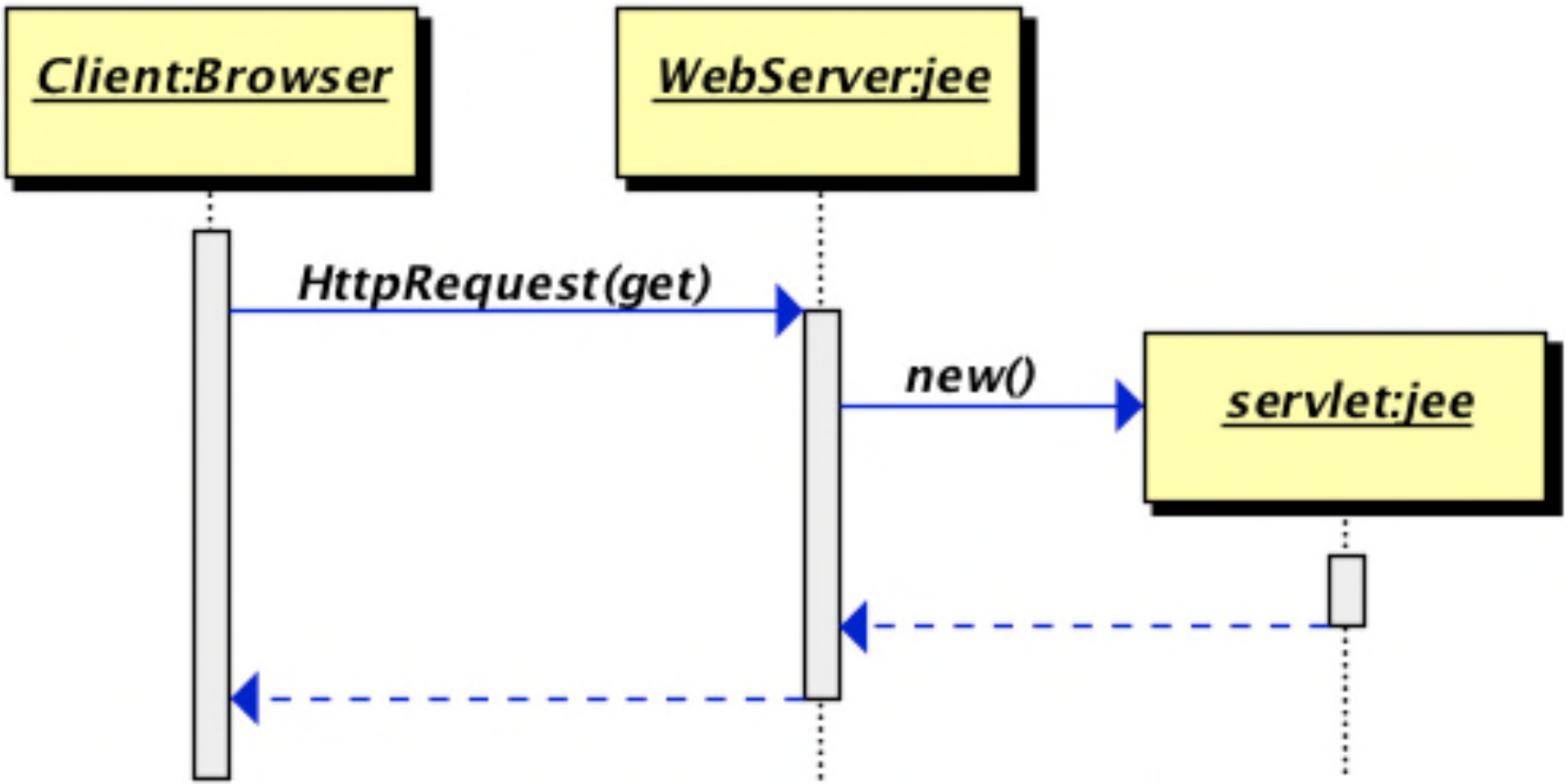
Multipurpose Internet Mail Extensions (MIME)

POST : formulaire.

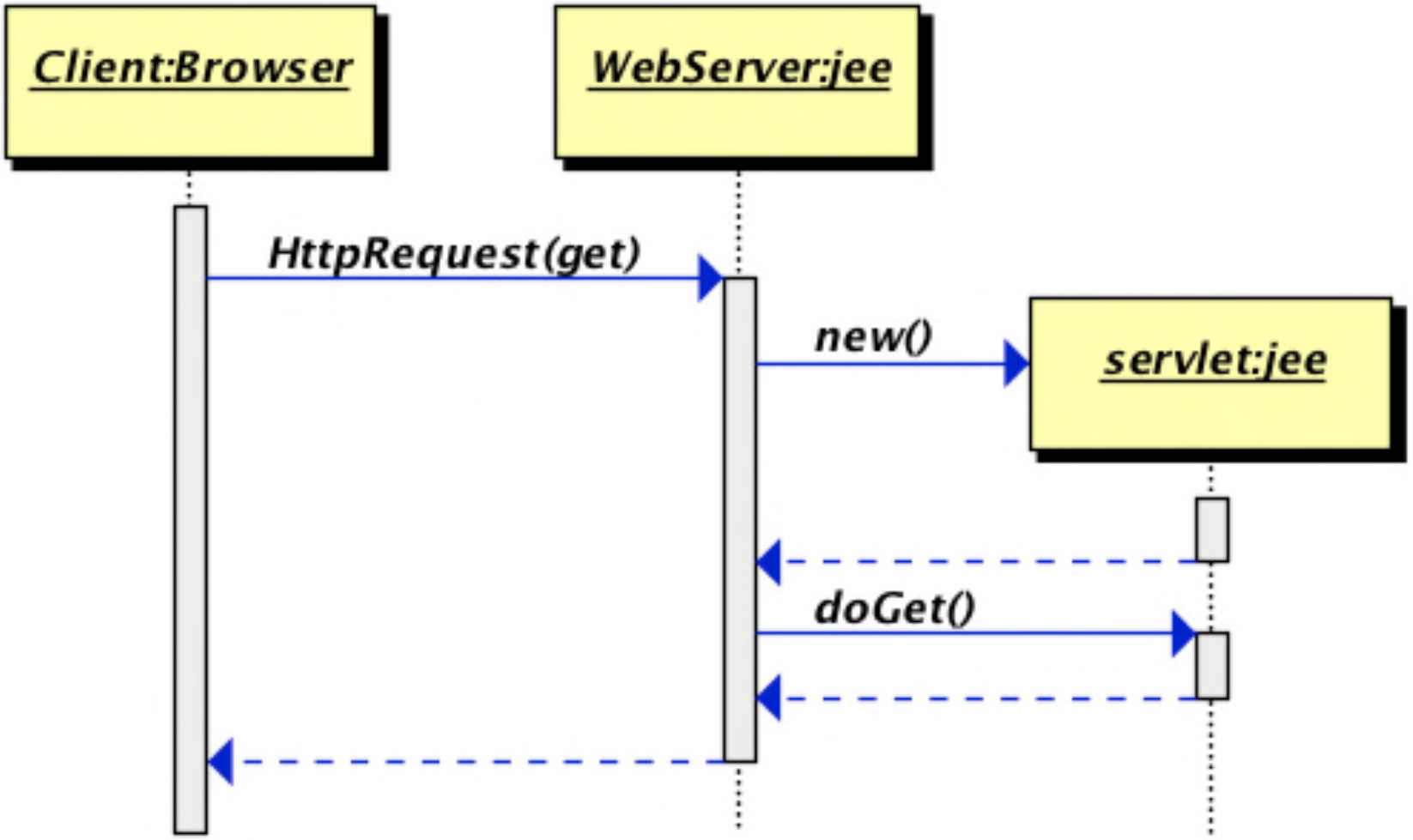
Web Server Component (SUN)



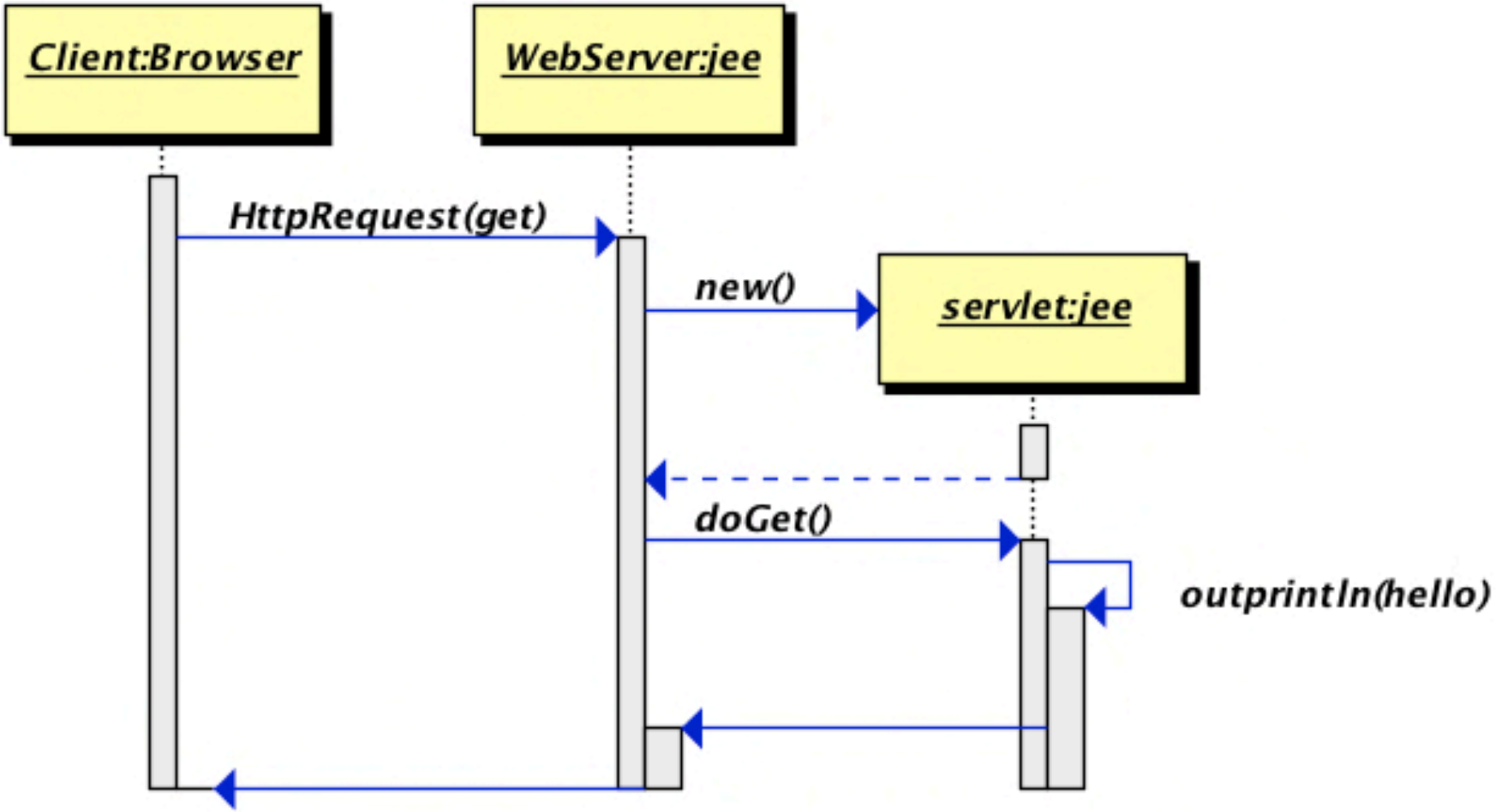
Web Server


Client:Browser
WebServer:jee
HttpRequest(get)
new()
servlet:jee

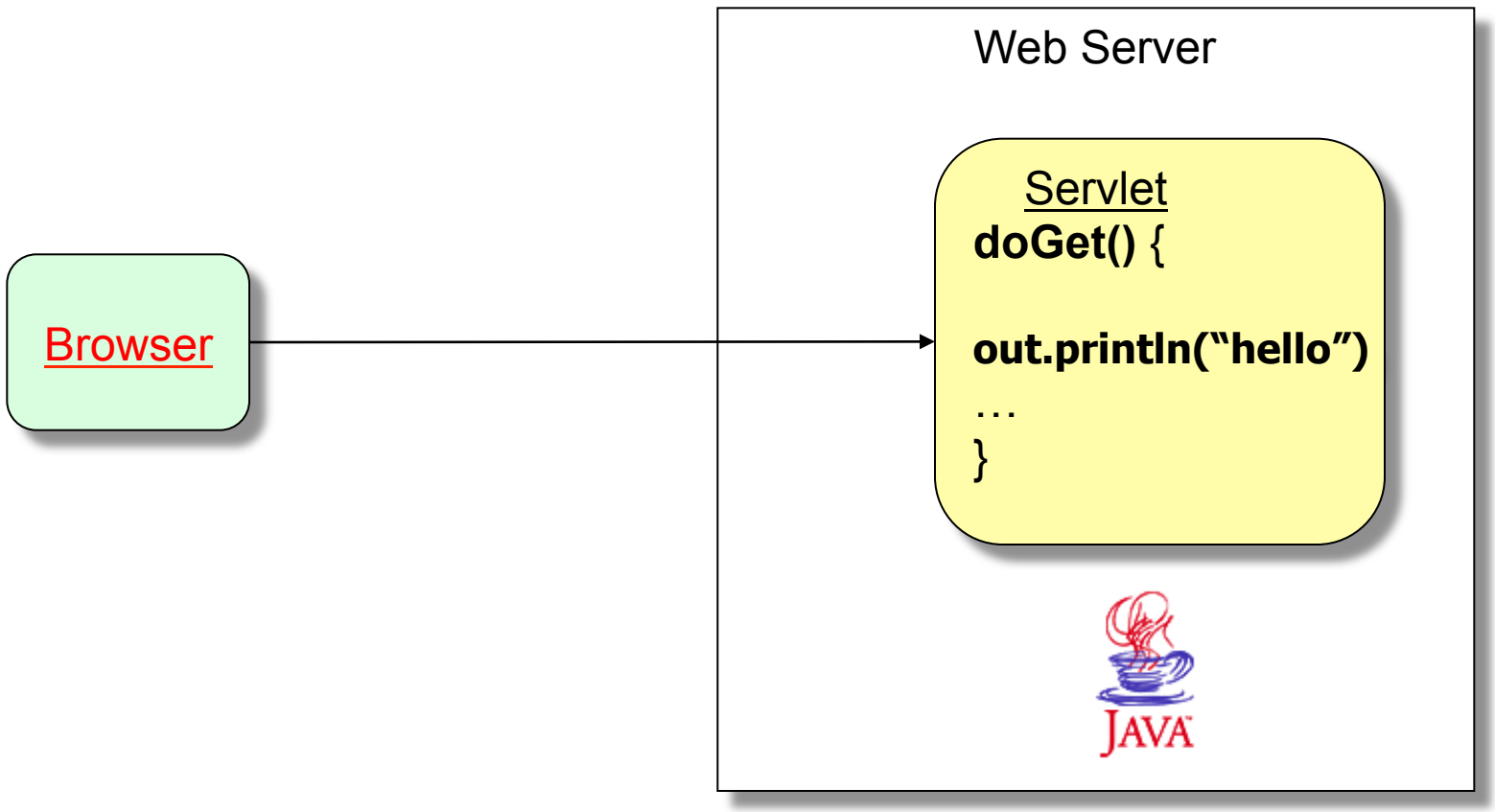
Web Server


Client:Browser
WebServer:jee
HttpRequest(get)
new()
servlet:jee
doGet()

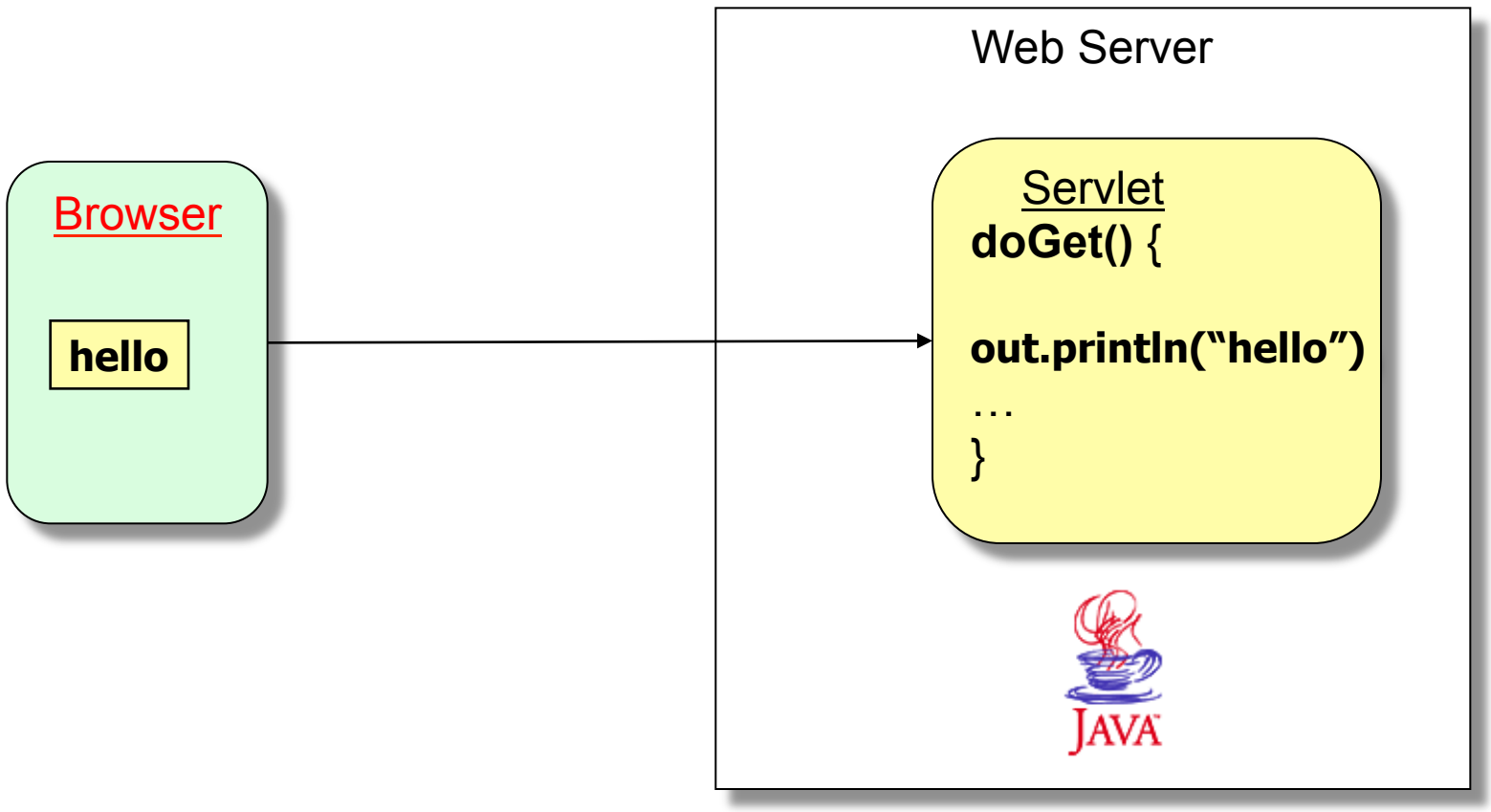
Web Server


Client:Browser
WebServer:jee
HttpRequest(get)
new()
servlet:jee
doGet()
outprintln(hello)

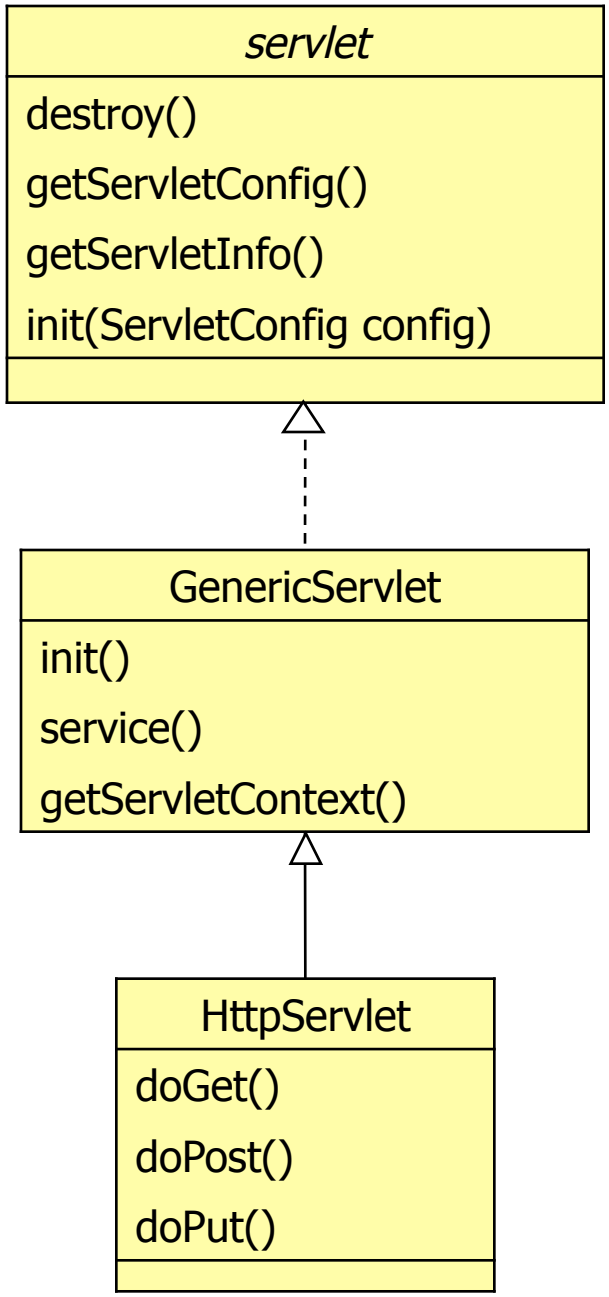
Web Server



Web Server

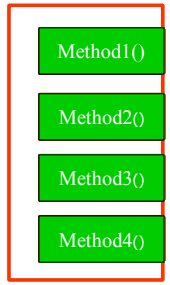

Web Server
Browser
hello
Servlet
doGet() {
out.println("hello")
…
}
JAVA

Servlet


servlet
destroy()
getServletConfig()
getServletInfo()
init(ServletConfig config)
GenericServlet
init()
service()
getServletContext()
HttpServlet
doGet()
doPost()
doPut()

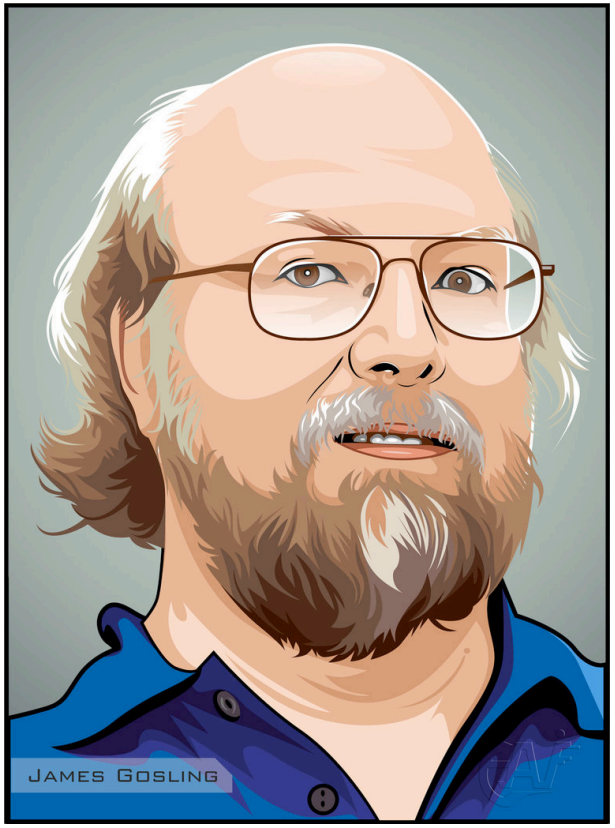
abstract class HttpServlet

- A subclass of HttpServlet must override at least one method, usually one of these:
  - *doGet()* : if the servlet supports HTTP GET requests
  - *doPost()* : for HTTP POST requests
  - *doPut() :* for HTTP PUT requests
  - *doDelete()* : for HTTP DELETE requests
  - *Init() and destroy()* : to manage resources that are held for the life of the servlet
  - *getServletInfo()* : which the servlet uses to provide information about itself



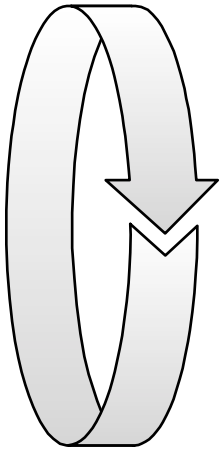
History

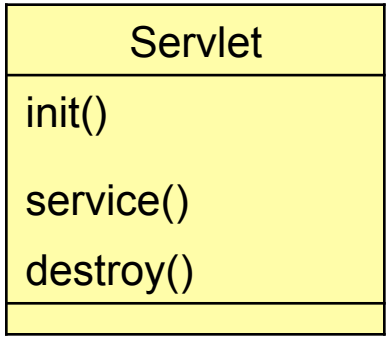
- Servlets were originally conceived of by James Gosling in 1995.
- Pavani Diwanji : Jeeves
  - Java Web Server
  - Java Server API
- Netscape

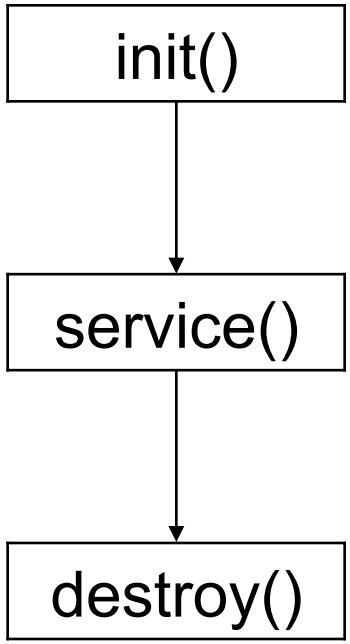


Cycle de vie d'un servlet

- Gérer par le conteneur.
  - Inversion de contrôle (Hollywood)
- Un servlet n'a pas de main()
- Un servlet n'a pas de constructeur (*)

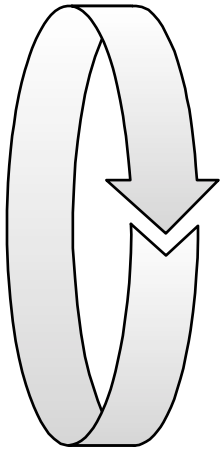






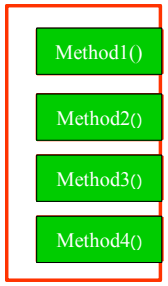
Un servlet n'a pas de main()

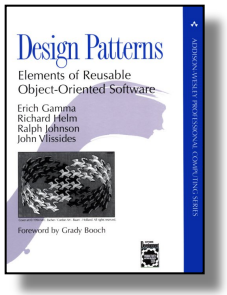
- La main loop est gérée par le « framework »

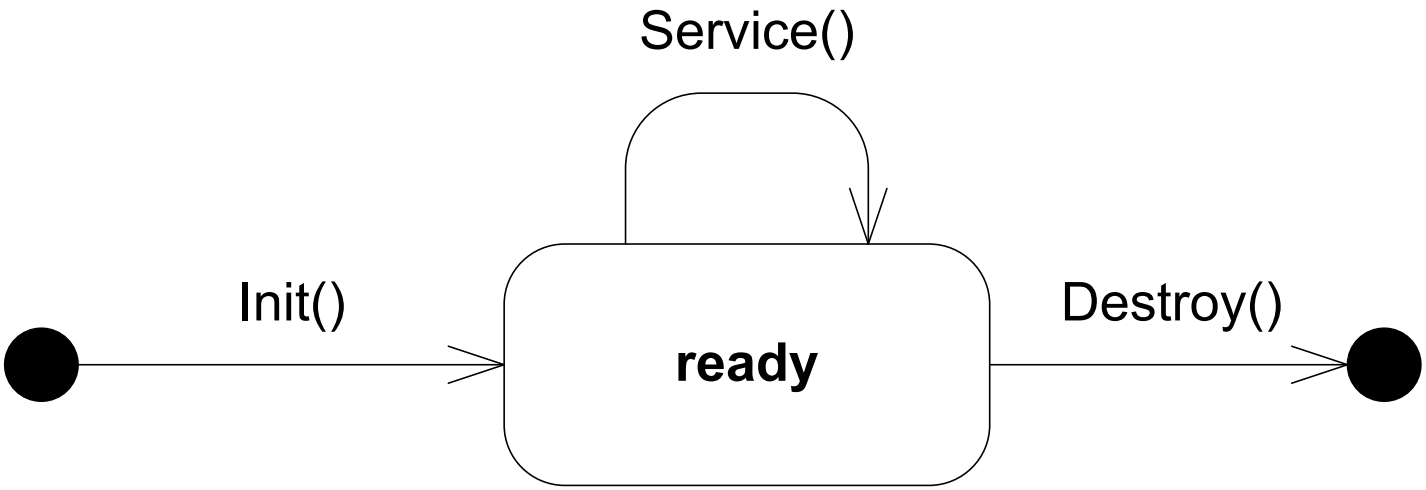


```
package noMain;

public class NoMain {

        public static void main(String[] args) {

        // run from here forever


        }

}
```

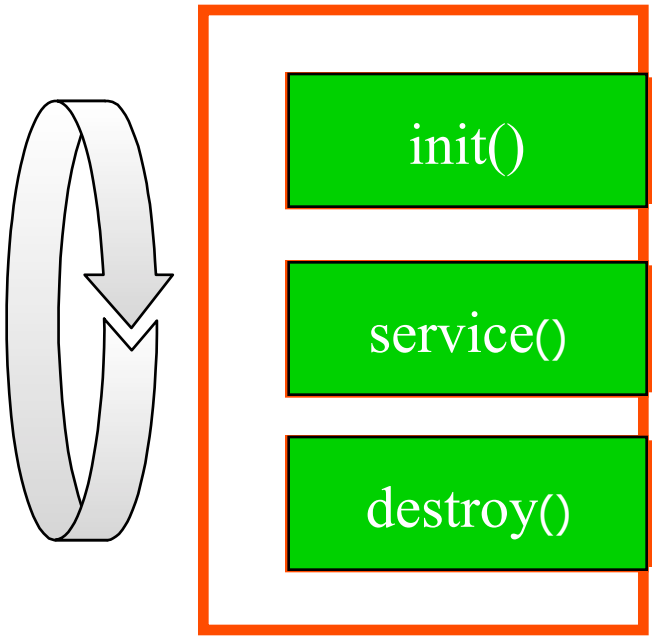
Servlet life cycle State Diagramm



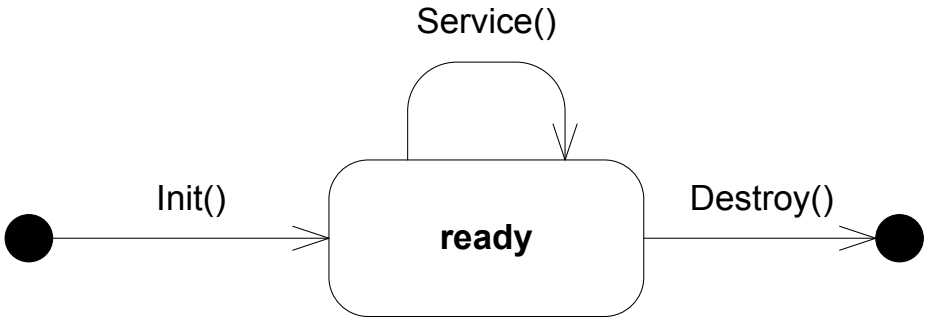


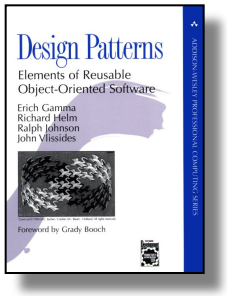


Servlet life cycle State Diagramm

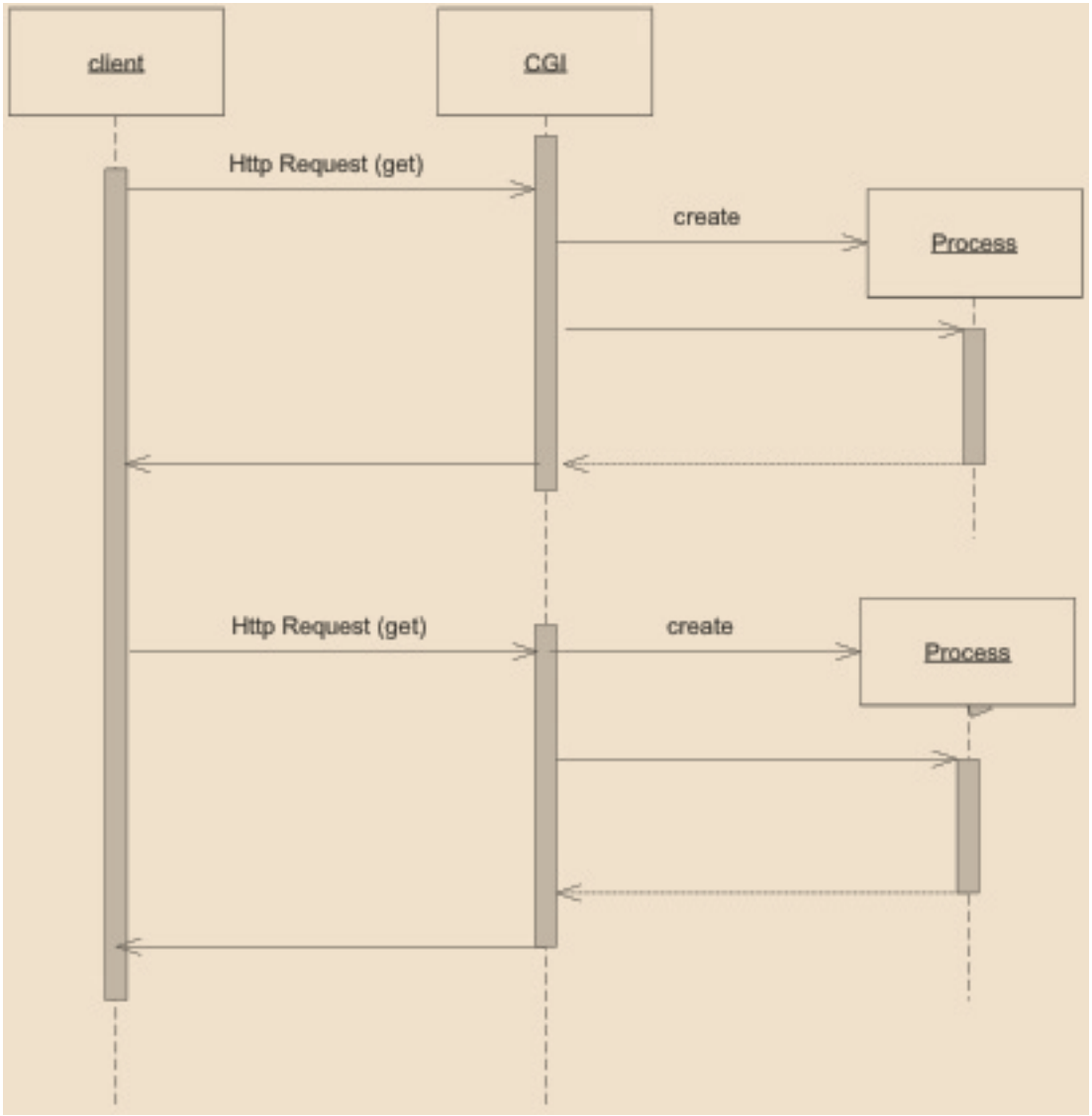


Container

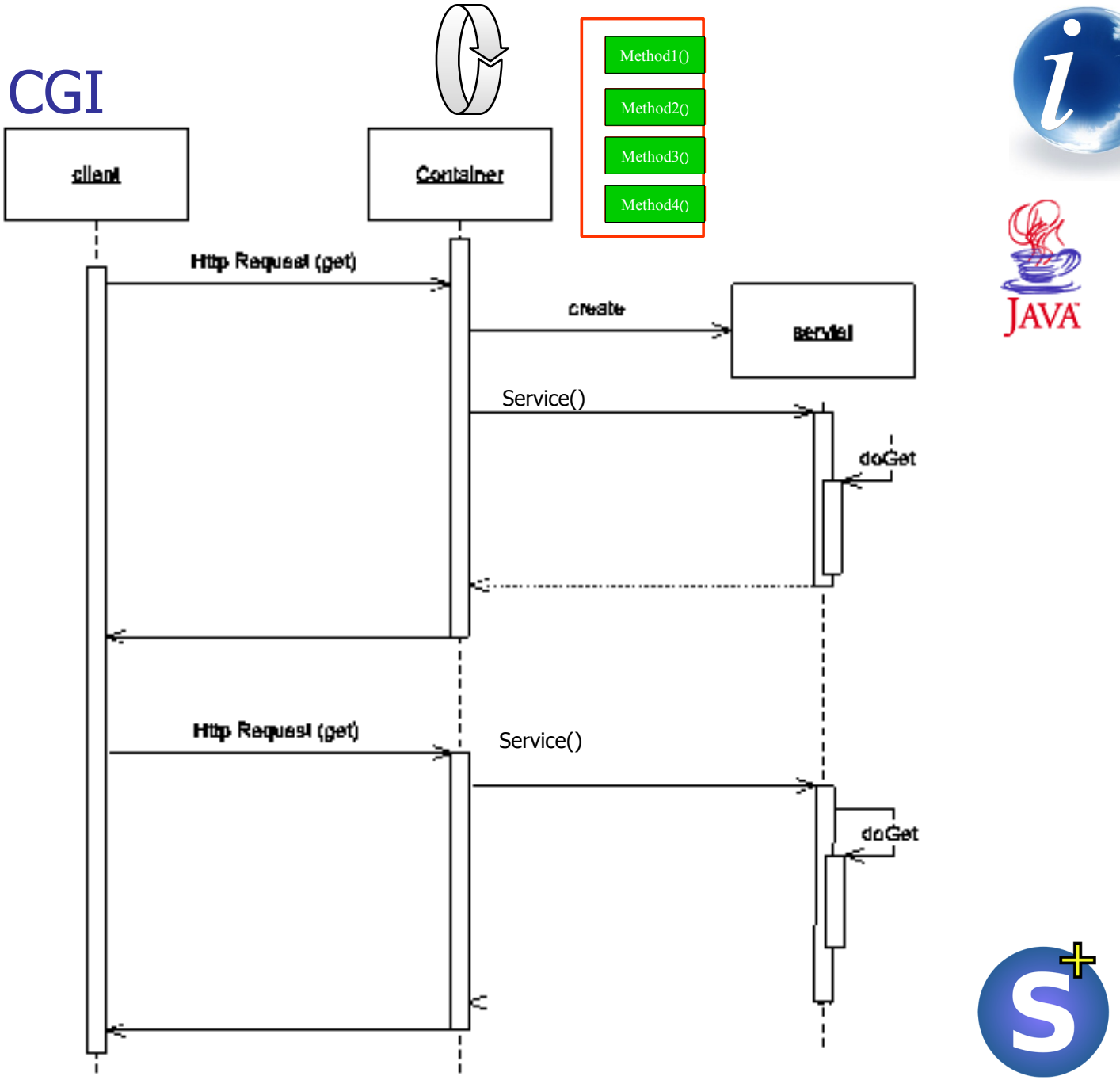




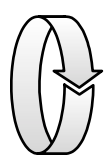
Servlet vs CGI (Common Gateway Interface)

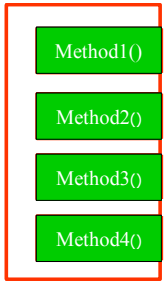


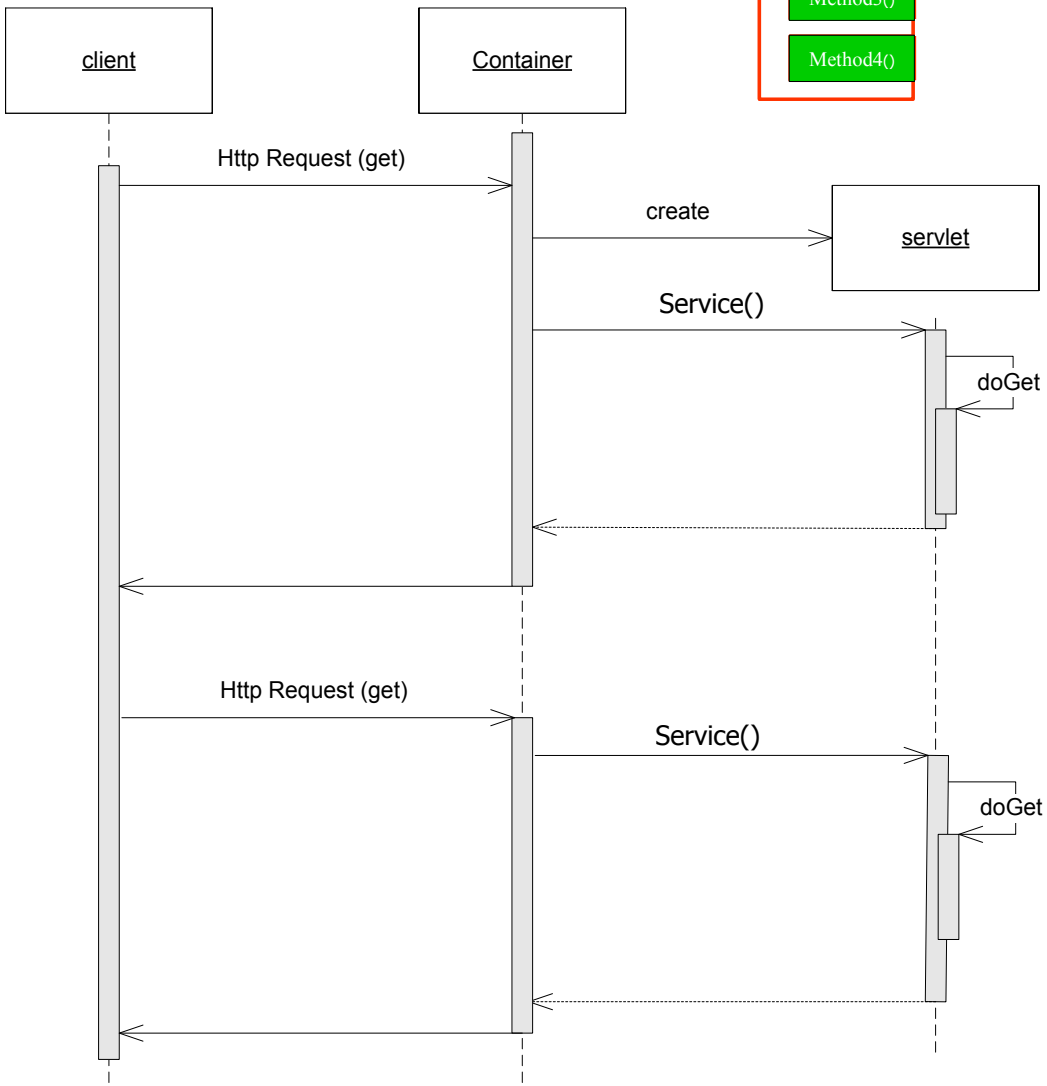
Servlet vs CGI



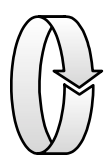
Servlet vs CGI

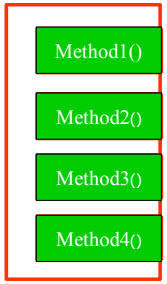


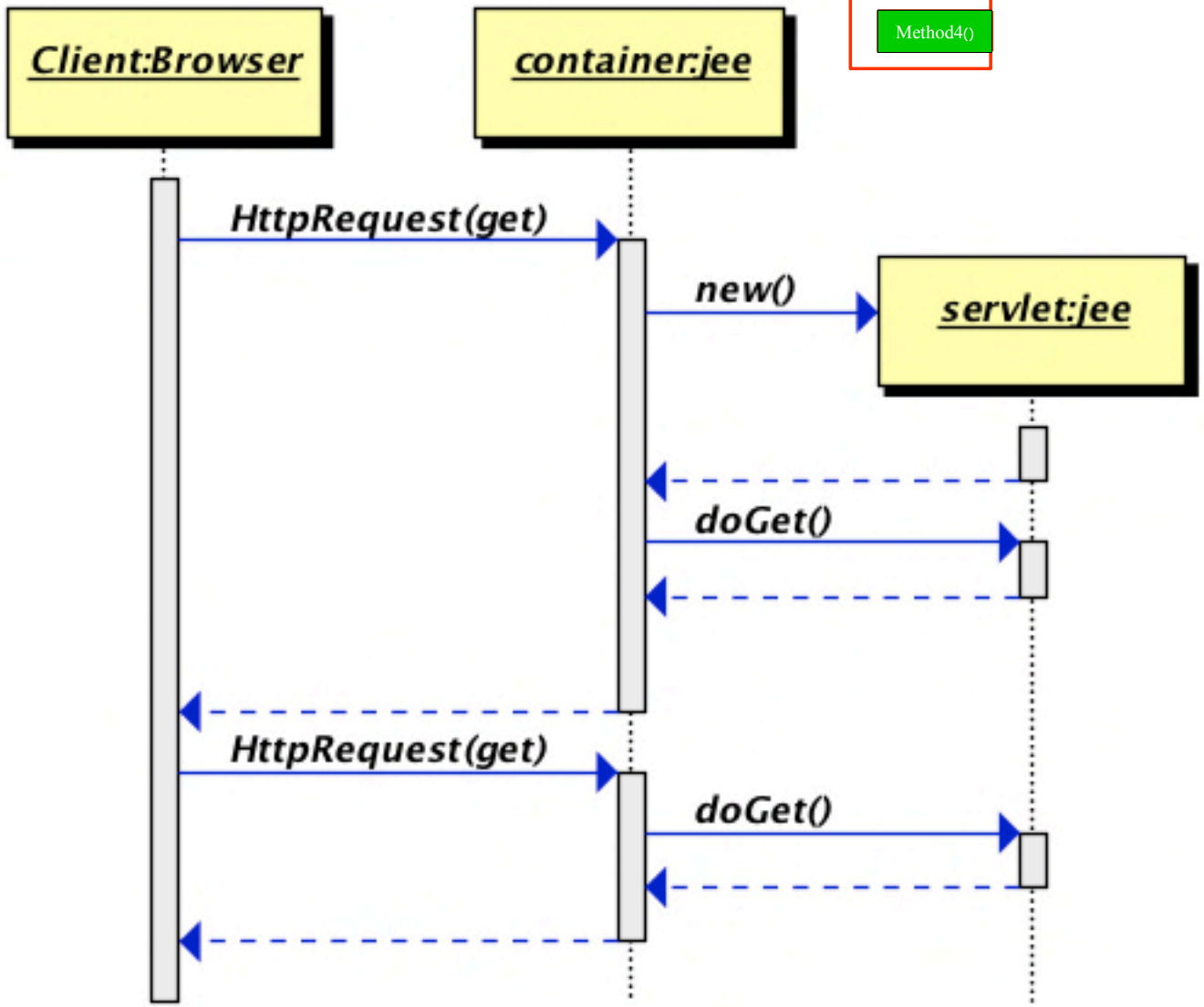




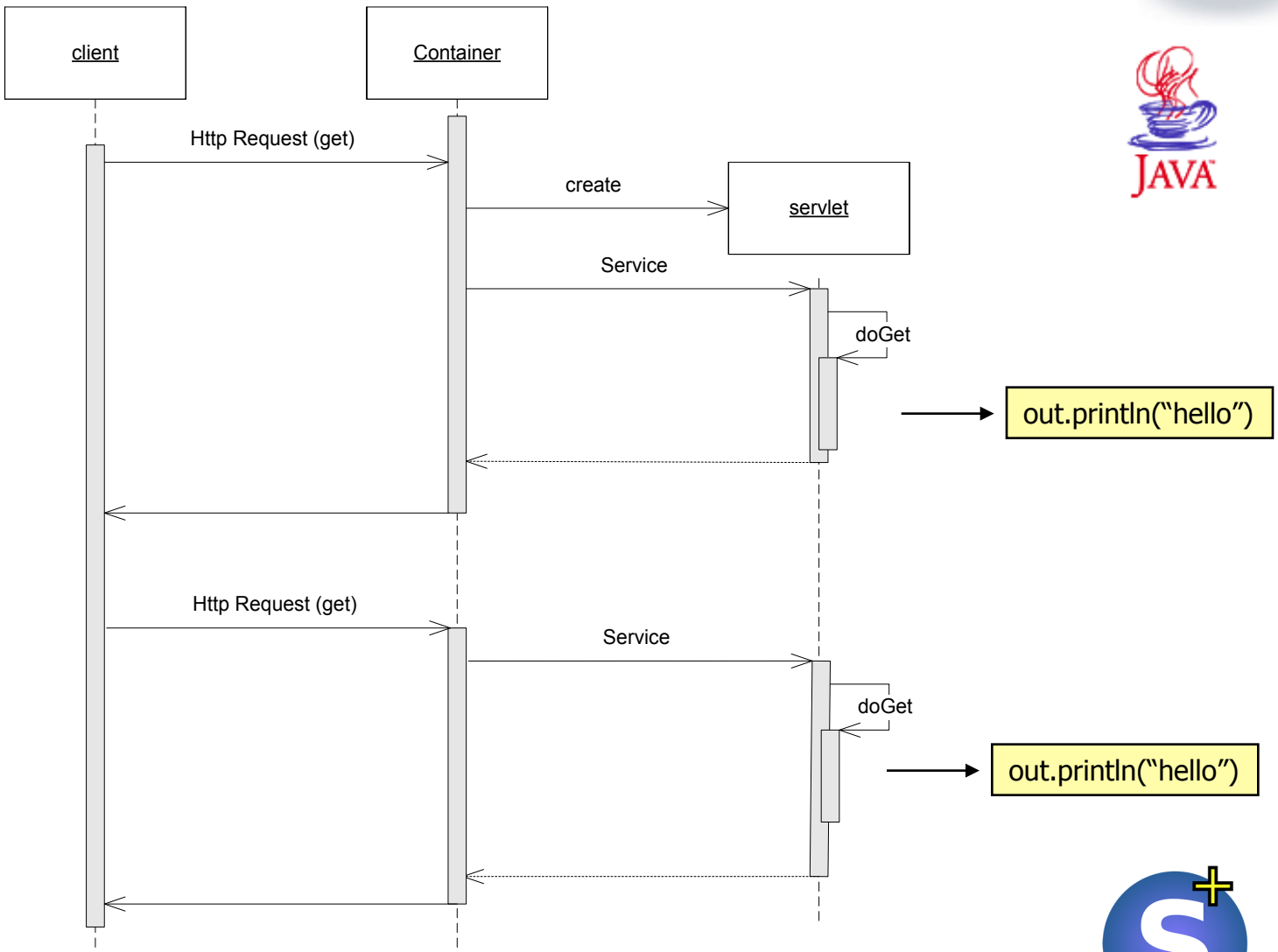
Servlet vs CGI



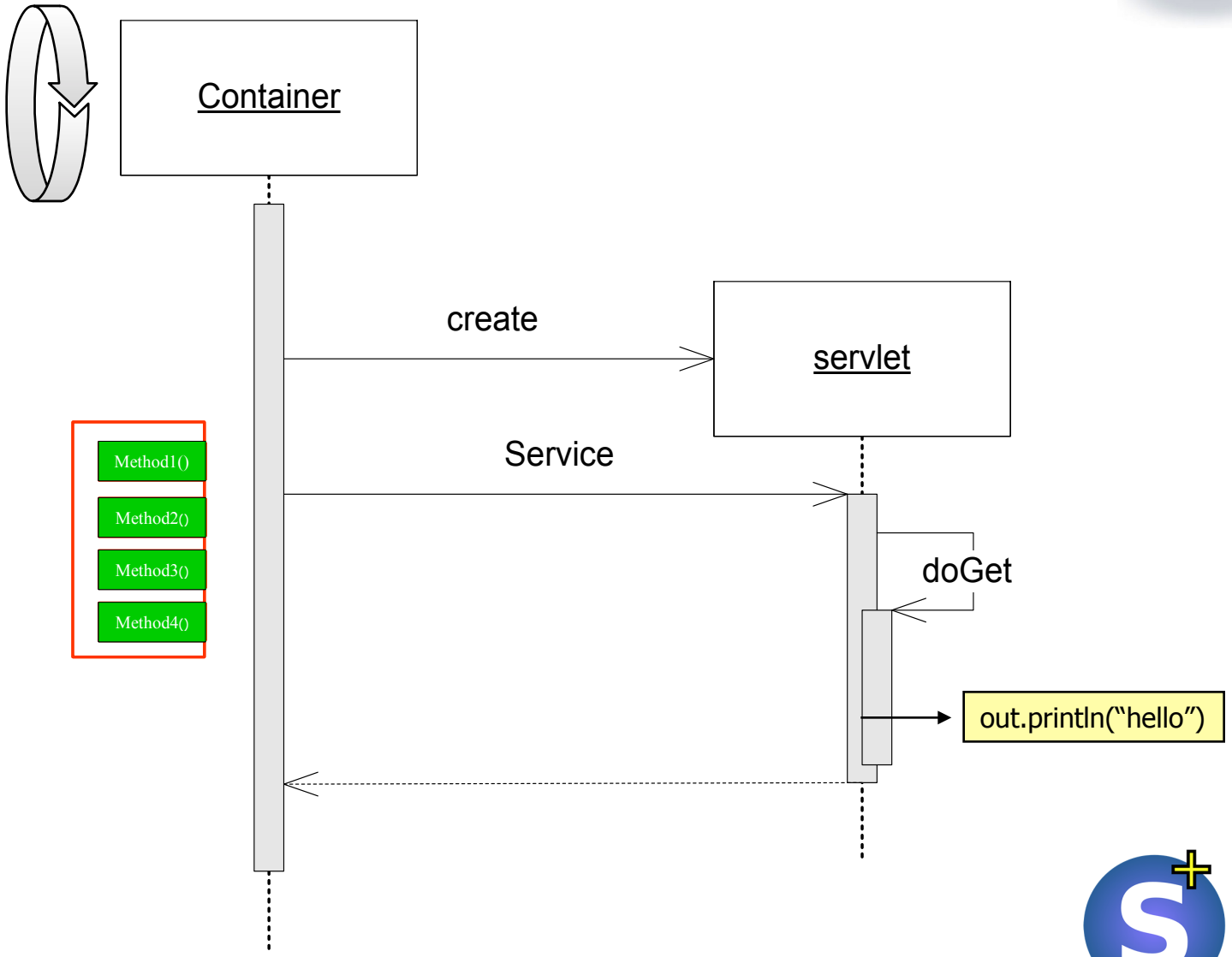




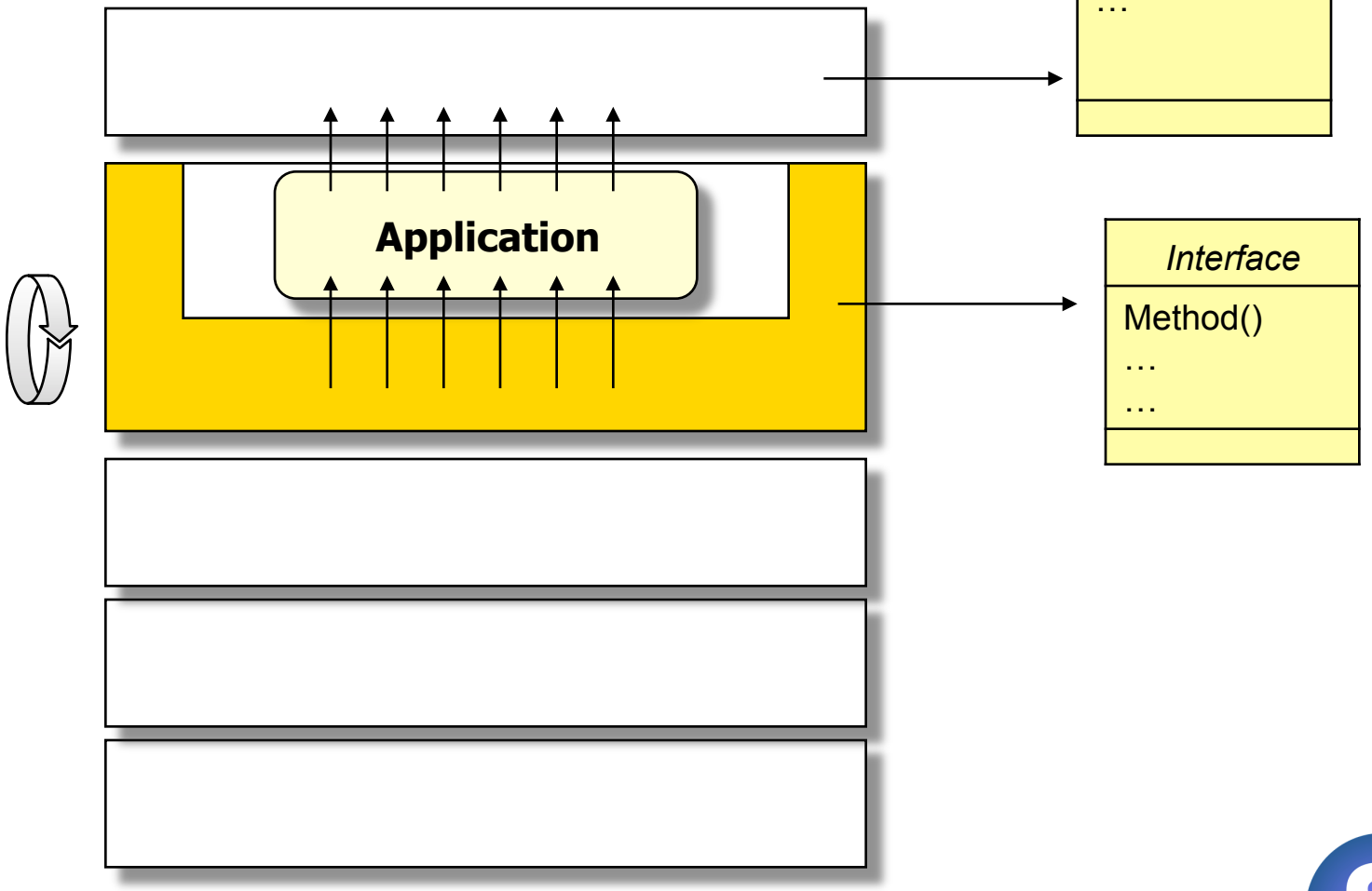
Servlet vs CGI



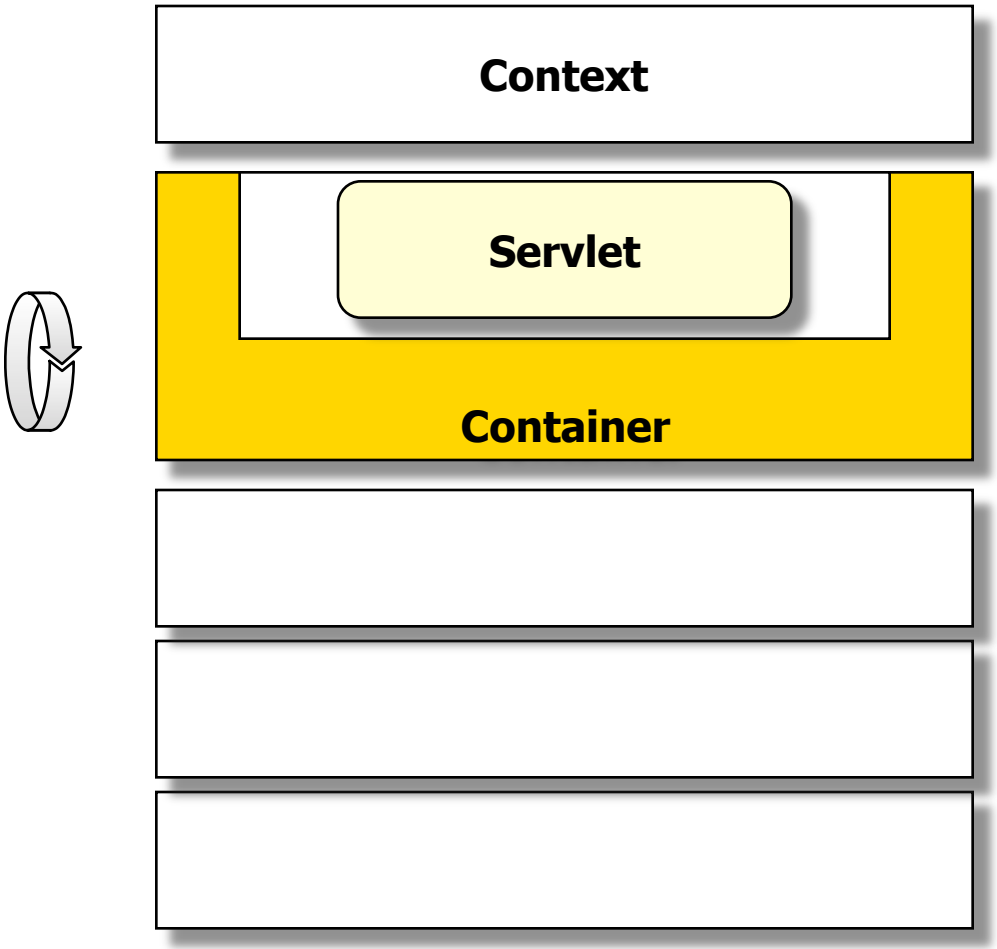
Service() invokes doGet

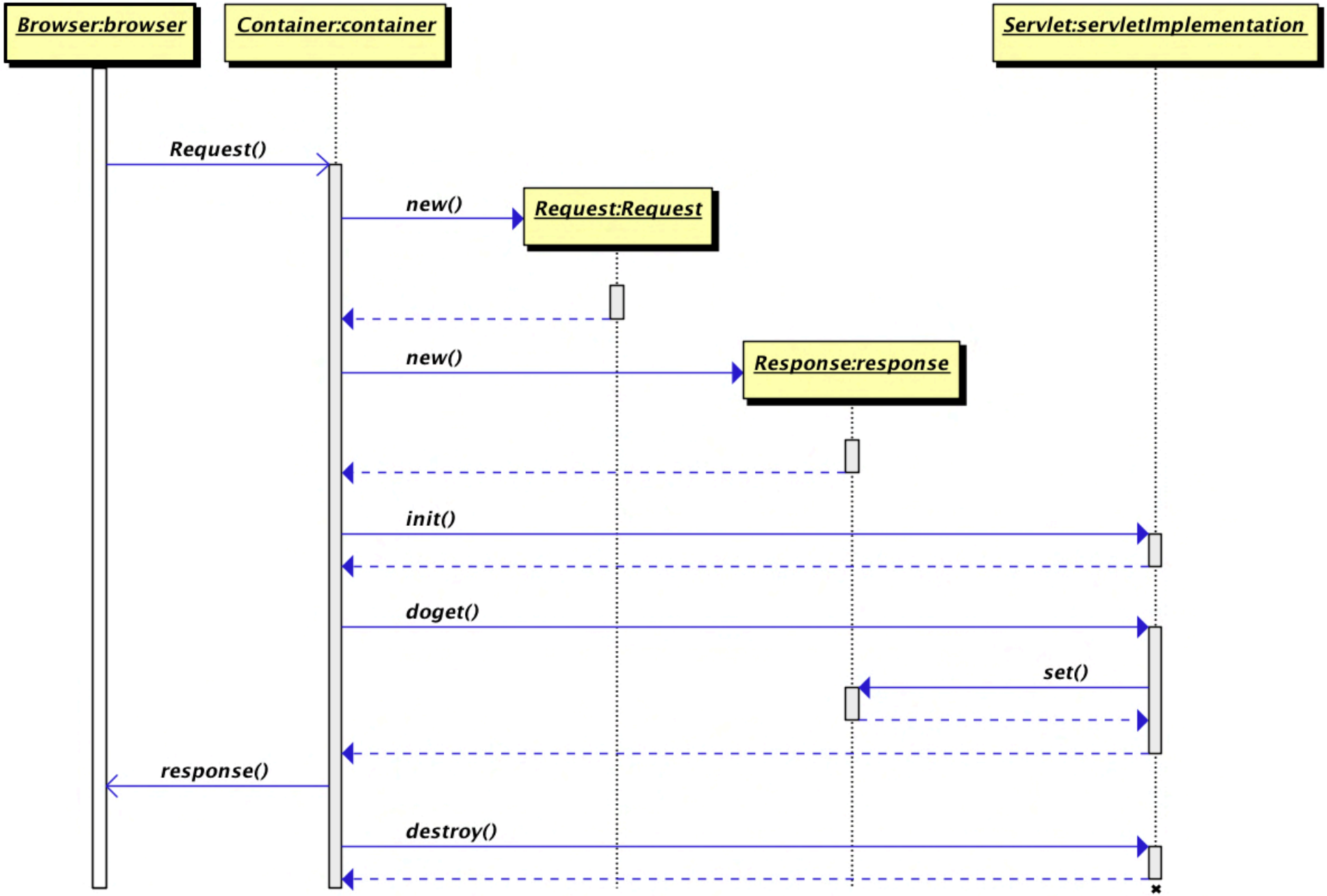


Container et Context



Servlet Container and Servlet Context

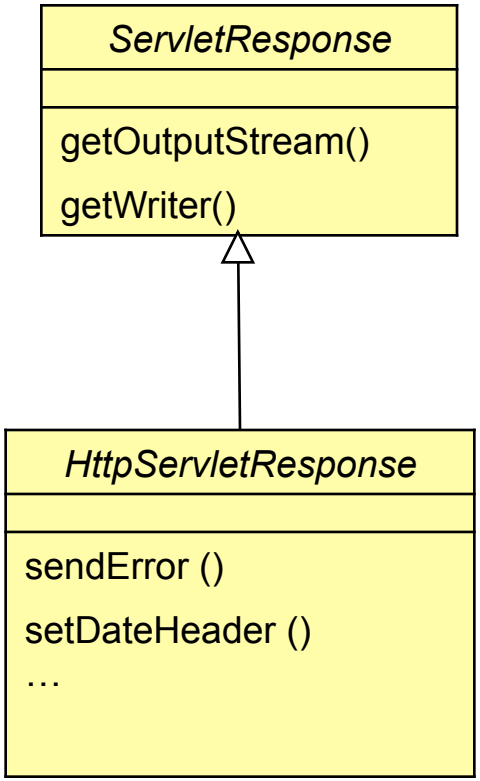



Browser:browser
Container:container
Servlet:servletImplementation
Request()
new()
Request:Request
new()
Response:response
init()
doget()
set()
response()
destroy()

Classe HttpServletRequest

- Un objet de cette classe permet d'accéder aux éléments du header HTTP de la requête, tels que:
  - les cookies envoyés avec la requête,
  - le type de méthode utilisée,
  - ainsi que les noms de paramètres et leur valeur passés avec la requète.
- La méthode *getHeader()* permet de récupérer tout ou partie du header de la requête HTTP.

ServletResponse



Hello world servlet

```
import java.io.*;
import javax.servlet.http.*;
import javax.servlet.*;
public class HelloServlet extends HttpServlet {


  public void doGet (                    HttpServletRequest request,
                                         HttpServletResponse response)

    throws ServletException, IOException {

                PrintWriter out = response.getWriter();
                out.println( "hello");

        }
}
```