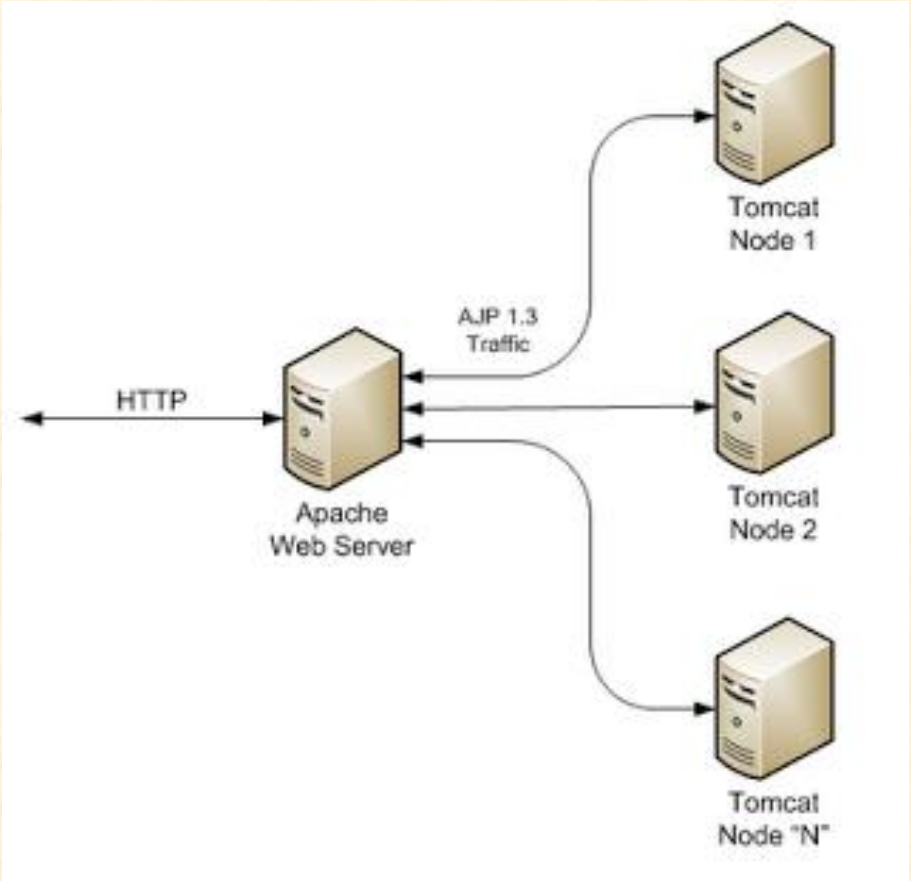
INTRODUCTION TO SERVLETS AND WEB CONTAINERS

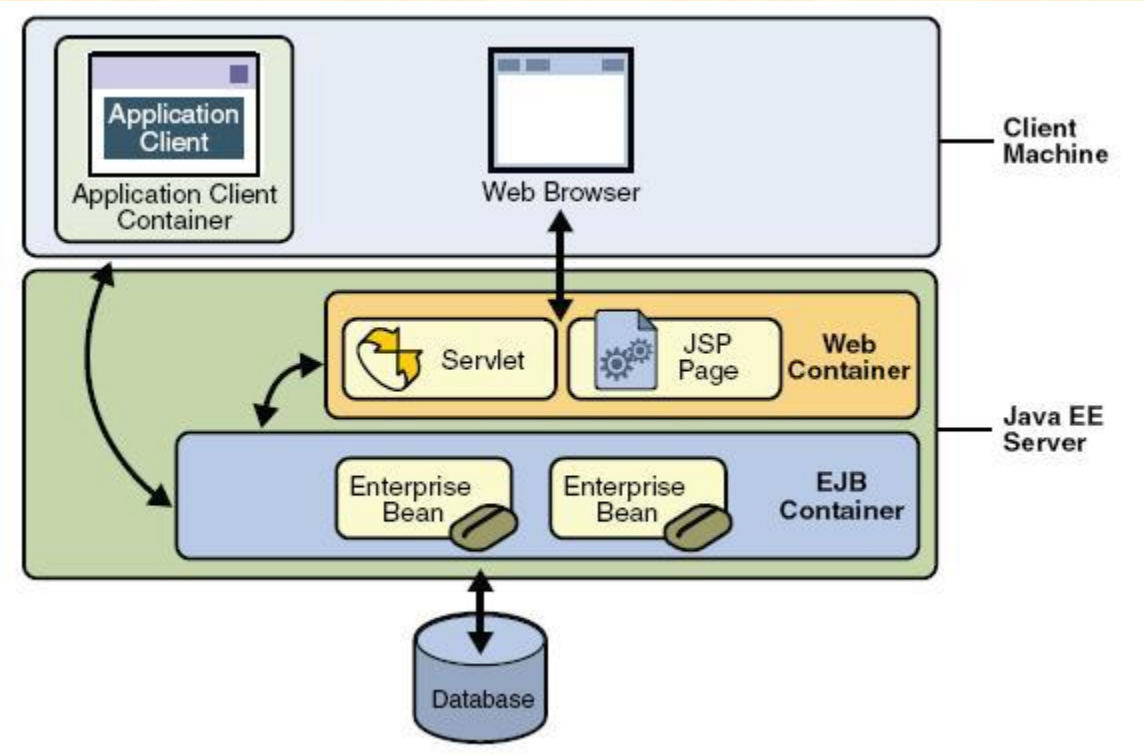
Actions in Accord with All the Laws of Nature

Web server vs web container

- Most commercial web applications use Apache
  - proven architecture and free license.
- Tomcat can act as simple web server
  - for production environments it may lack features like load-balancing, redundancy, etc.
- Glassfish is like Tomcat, but is also JEE container
  - TomEE ?



Web container and servlet architecture



A servlet is a Java class that extends the capabilities of servers that host applications access by means of a request-response programming model.

World's simplest possible servlet demo

- Read the steps for demo1
- Note the following:
  - The html page (why is it called index.jsp?)
  - What will happen when you click on the hyperlink?

    ```
    <a href='SimplestServlet'>SimplestServletDemo</a>
    ```
  - When will doGet be called and what will it do?
  - When will doPost be called and what will it do?
- Try implementing it (follow the steps carefully, especially 13)
- Now you know how to implement a web app in Netbeans!

SimplestServlet

```
public class SimplestServlet extends HttpServlet {

  @Override
  protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException
  {
    PrintWriter out = response.getWriter();
    out.print("<html><head><title>Test</title></head><body>");
    out.print("<form method='post'>");
    out.print("<p>Please click the button</p>");
    out.print("<input type='submit' value='Click me'/>");
    out.print("</form>");
    out.print("</body></html>");
  }

  @Override
  protected void doPost(HttpServletRequest request, HttpServletResponse response)
  throws ServletException, IOException
  {
    PrintWriter out = response.getWriter();
    out.print("<html><head><title>Test</title></head><body>");
    out.print("<p>Postback received</p>");
    out.print("</body></html>");
  }
}
```

web.xml

```
<web-app …>
    <servlet>
        <servlet-name>SimplestServlet</servlet-name>
        <servlet-class>mum.cs545.SimplestServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>SimplestServlet</servlet-name>
        <url-pattern>/SimplestServlet</url-pattern>
    </servlet-mapping>
    <session-config>
        …
    </session-config>
</web-app>
```

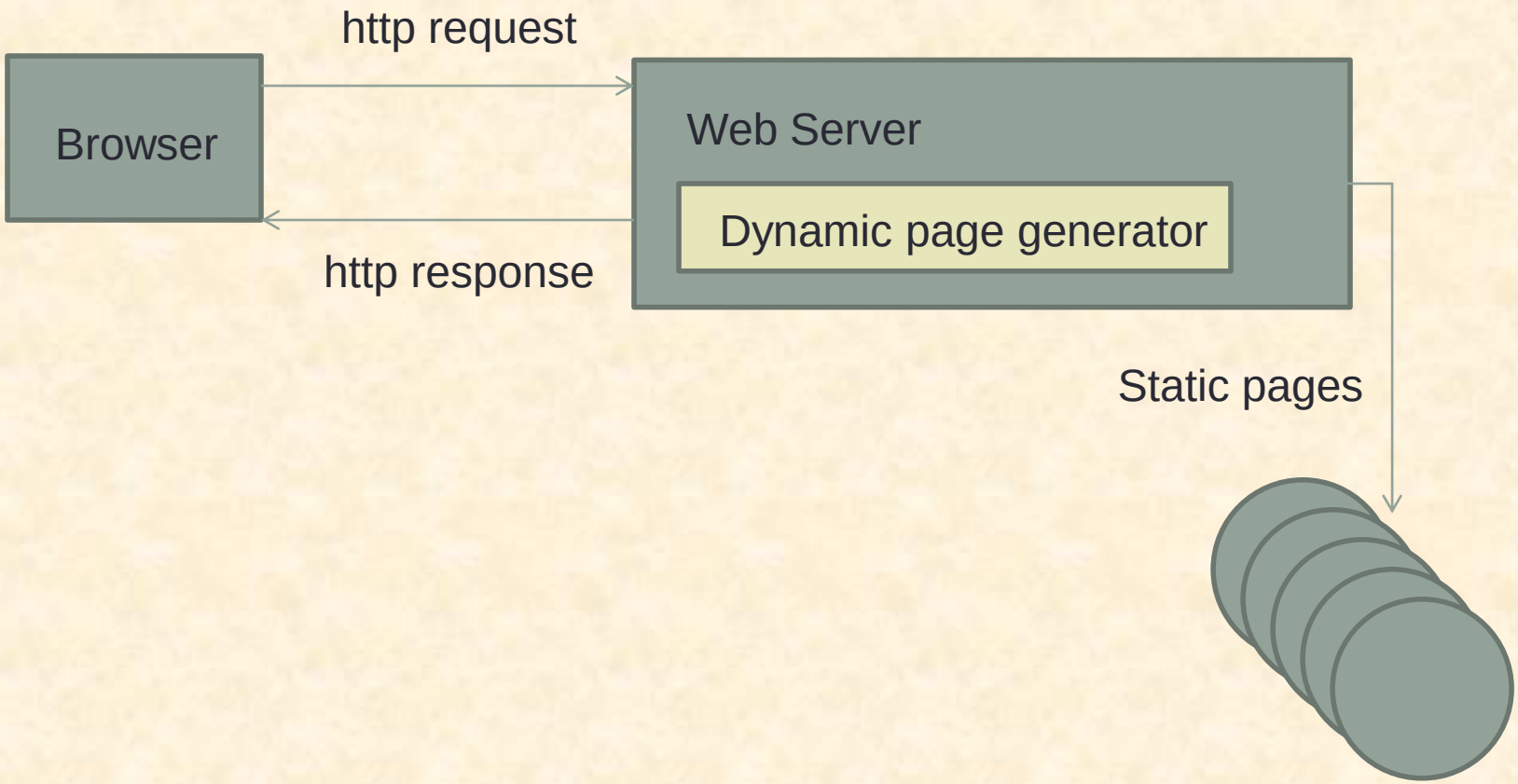
Main Point 1

Servlets are the basis of dynamic web applications. Servlets process information from a request object and generate new information in a response object.

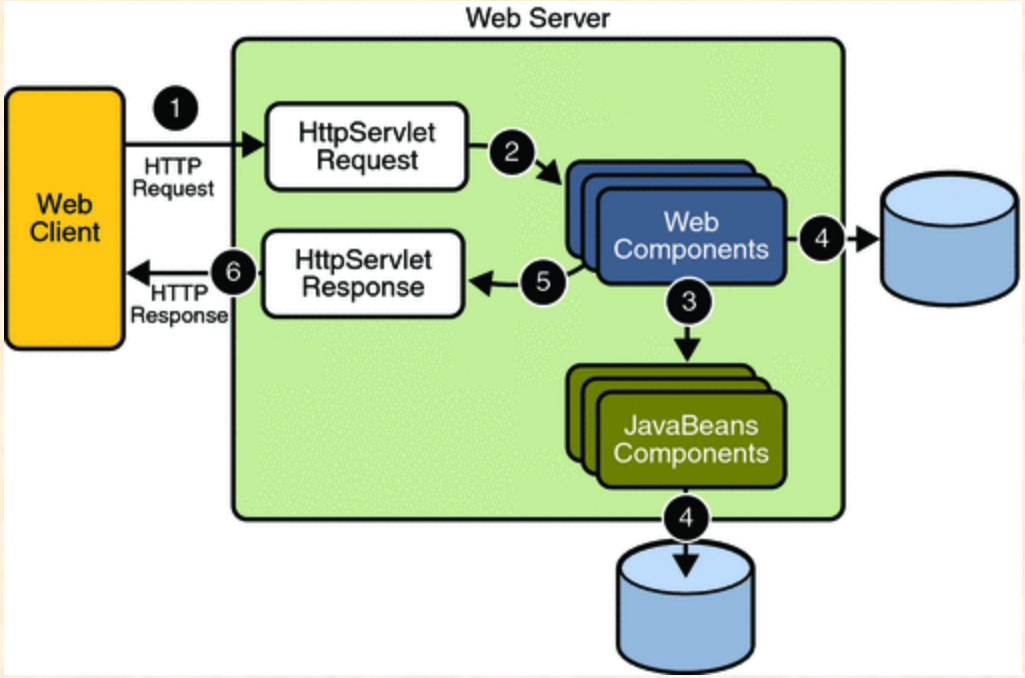
Science of Consciousness:

Analogously, humans also operate by giving responses to requests. Response are more likely to be correct and appropriate to the extent that one has broad awareness to consider all relevant information and fine focus to understand the request.

Dynamic versus static pages



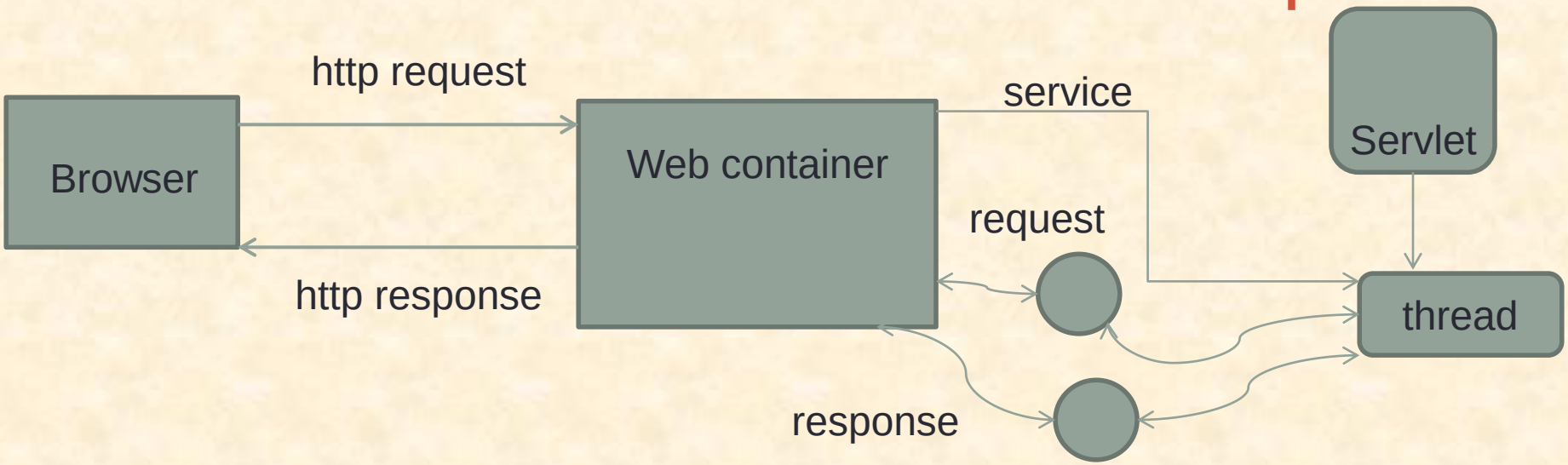
Web server with Servlet container



Containers provide fundamental support

- network communications
  - communicating with web server
- lifecycle management
  - no “main” method in a servlet, …
- concurrency
- state management
  - session and context attributes
- security
- Support for JSP (JSF, JPA, JTA, EJB, …)

How container handles an HTTP request



- Container receives new request for a servlet
- Creates HttpServletRequest and HttpServletResponse objects
- Calls service method on HttpServlet object in thread
- When thread completes, converts response object into HTTP response message

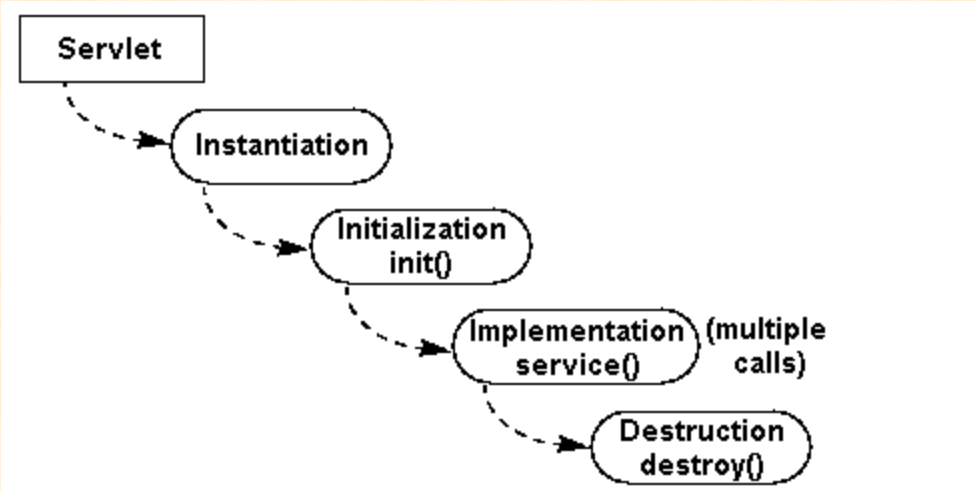
Main Point 2

Web containers provide essential support services for servlets. **Science of Consciousness**: Experience of the unified field of pure consciousness provides essential support services for broad awareness and fine focus in the individual.

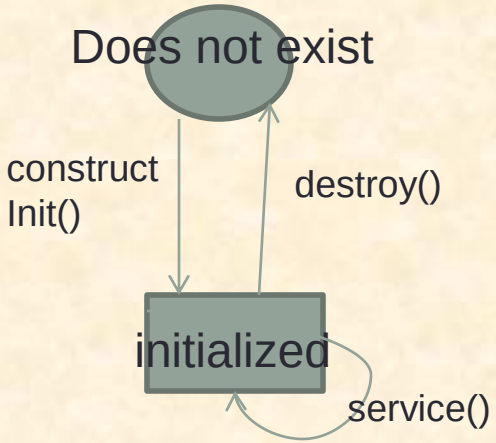
The simple web server becomes a (mylet) container

- Inspect the code for demo 2
- The red lines have been added to make the simple web server from lesson 1 into a mylet container
  - What's a mylet?

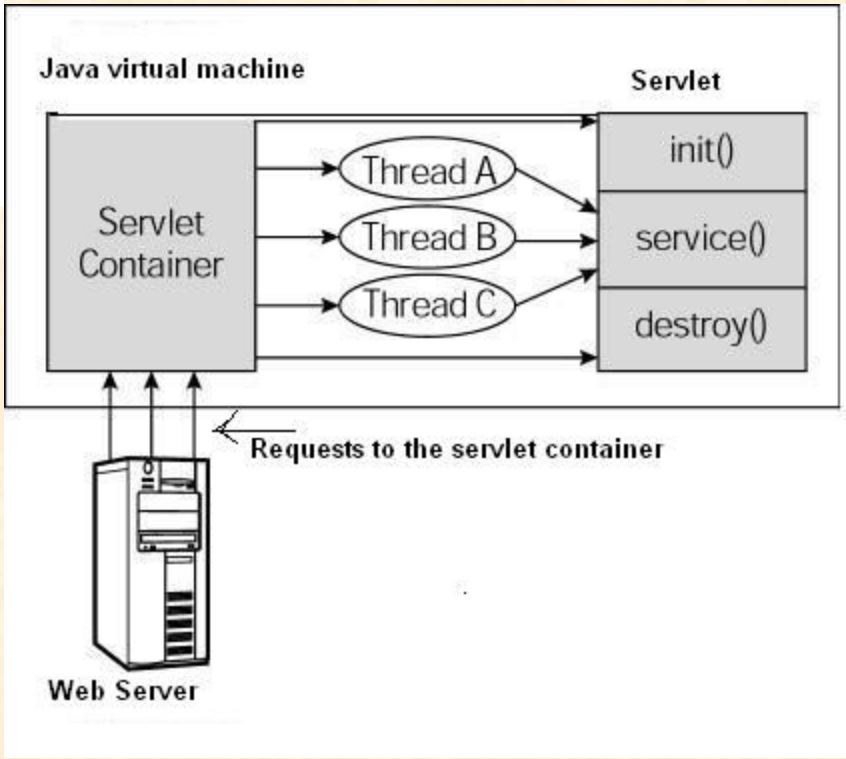
Servlet life cycle



. Load
. Create
. Init
. Service
. Destroy



- 1 instance of servlet
- new thread created for every request
  - Then service() called on the thread
- All threads share instance variables
- Each thread has own stack for local variables
- Compare with mylets



Servlet life cycle (related to mylet container)

1. Load servlet class (Class c = Class.forname(...))
2. Instantiate servlet (c.newInstance());
3. init() called only once in the servlet's life.
   Must complete before Container can call service().
4. service() (called for each request, each request runs in a separate thread)
5. destroy() (called only once)

Main Point 3

Web containers manage the life cycle of servlets. **Science of Consciousness**: The unified field manages the entire universe, and by experiencing this field of awareness on a regular basis we spontaneously act more and more in accord with all the laws of nature.