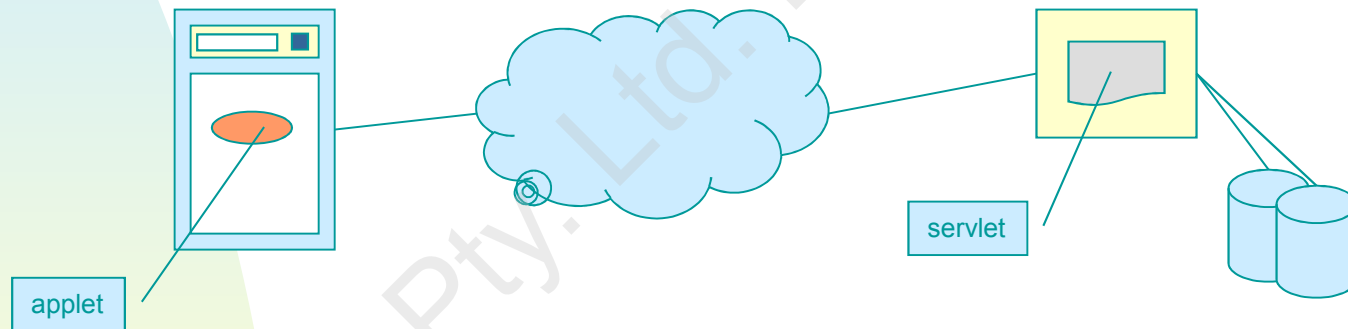


# Servlets

- Server-side Java applications
  - ◆ allow for dynamic extension of the functionality of a server
    - ☞ servlets execute in server's context
    - ☞ c.f. applets and browsers



- ◆ the new CGI (Common Gateway Interface)
  - ☞ more efficient
  - ☞ greater portability
- ◆ now being superseded by “application servers”...

# Servlets

## ◆ advantages:

- ☞ faster and 'cleaner' than CGI scripts
- ☞ can be called dynamically from within HTML page via server-side include tags
- ☞ easily configured using server's admin facility
  - typically a web-based GUI
- ☞ use a standard API and language (Java!)
  - gives all the inherent advantages of Java
- ☞ can be dynamically loaded & invoked from local disk or from another server
- ☞ can be chained and activated sequentially
- ☞ extensible

# Servlets

- ◆ standard extension to Java 1.1
  - ☞ javax.servlet.\* and javax.servlet.http.\*
- ◆ supported by several servers:
  - ☞ Lotus Domino Go
  - ☞ Netscape FastTrack & Enterprise
  - ☞ Microsoft IIS
  - ☞ Apache

# Servlets

- ◆ created by extending HttpServlet class
  - ☞ extension of GenericServlet
  - ☞ implements Servlet interface
  - ☞ class provides methods corresponding to HTTP protocol
    - doGet, doPost, doPut, etc.
    - getServletInfo
- ◆ undertakes a 'handshake' with its server context:
  - ☞ HttpServletRequest
    - HTTP request passed to servlet
  - ☞ HttpServletResponse
    - servlet's response to server

# Servlets

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;
public class MyServletClass extends HttpServlet
{
    // respond to POST form
    public void doPost (HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException
    {
        OutputStream respOS = resp.getOutputStream ();
        PrintStream out = new PrintStream (respOS);
        resp.setContentType ("text/html");
        out.println ("<BR>HTTP header info...<P>");
        out.println ("<BR>URI: " + req.getRequestURI ().toString ());
        out.println ("<BR>Query String: " + req.getQueryString ());
        out.println ("<BR>Method: " + req.getMethod ());
        out.println ("<BR>Header Fields:");
        Enumeration e = req.getHeaderNames ();
        while (e.hasMoreElements ())
        {
            String name = (String) e.nextElement (),
                value = (String) req.getHeader (name);
            out.println ("<BR>" + name + " = " + value);
        }
        out.println ("<P>FORM Fields...<P>");
        out.println ("<BR>FirstName: " + req.getParameter ("FirstName"));
        out.println ("<BR>LastName: " + req.getParameter ("LastName"));
        out.println ("<BR>DOBString: " + req.getParameter ("DOBString"));
        out.close ();
        respOS.close ();
    }
}
```

# Servlets

- ◆ servlet may be target of form's action specifier:

- ☞ HTML FORM fields specified as normal...

```
<FORM NAME="MyForm" METHOD = "POST"  
      ACTION = "http://my.machine.name/ServletDir/MyServlet">
```

- ◆ alternatively, can be specified in HTML with  
<SERVLET> tag

- ☞ similar to <APPLET>

- ☞ in this case, servlet is used to generate HTML that is then 'inlined' into the surrounding file

- ☞ file should be appended with ".shtml"

```
<SERVLET CODE="MyServletClass.class">  
  <PARAM NAME="Debug" VALUE="false">  
  <PARAM NAME="WelcomeString" VALE="Hi y'awl!">  
</SERVLET>
```