You click on the form's "submit" button. The Web browser collects the data within the form, assembles it into a string of name/value pairs, and specifies a POST method, the URL of the target program in the "cgi-bin" directory. The HTTP server receives the request and parses the message to discover that it's a POST for the "cgi-bin" program. The server starts a socket connection to the Web browser. The HTTP server checks the environment variables (e.g., content_length) in a blackboard. The server parses the message body via the input pipe. The CGI program reads the environment variables. The CGI program starts a CGI interaction. The CGI program reads the environment variables (e.g., content_length), in a blackboard. The CGI program receives the message body via the input pipe. The CGI program does some work, typically by interacting with a DBMS, TP Monitor, or external program. The CGI program returns the results via the output pipe. The HTTP server returns the results to the Web browser. The browser displays the results.
A poor man’s compound document architecture

1. A Web browser requests a Java applet
2. The server retrieves the requested applet
3. The server sends the applet
4. The browser loads the applet into the client’s memory and executes it
5. The browser deletes the applet from memory when it exits the Web page

A Web client/Server

- The browser initiates a separate TCP/IP session to download each applet within a Web page.
- The applet’s region does NOT visually integrate with the rest of the page.
- The Java compiler compiles Java bytecode into native code.
- The Bytecode verifier verifies that the bytecode is valid.
- The Class loader loads the class file into memory.
- The Just-in-Time (JIT) compiler compiles bytecode into native code just before it is executed.
- The Java run-time interpreter executes the native code.
- The Java runtime engine manages memory and exceptions.
- The Applet region is displayed within the Web page.