A.   The HTML Model vs. the Java Model

A.1. Objectives

This section provides an overview of Java, and its role as a technology for reducing browser bloat. This section is divided into N section. Section A.2 summarizes what Java is. Section A.3 describes the 7 layer OSI model and provides for a mapping to current Internet protocols. Section A.4 shows the HTML model and compares it with the Java model, placing it in the OSI model of the Internet.

A.2. What is Java?

The connotative meaning of *Java* is technically equivalent to the term; "*Java Technology*." Java Technology includes the Java programming language, supporting class libraries and the Java Virtual machine (the JVM).

Java Technology enables the running of Java programs using the *Java Model*. The Java model typically makes use of several layers (also called *substrates*). A Java Program is isolated from the hardware by virtue of a substrate called the *Java Virtual Machine*. Fig A.2-1 shows a diagram of the Java model.

```
┌─────────────────────────────────────┐
│                                      │
│         Java Program      ◄──────────── can be
│                                      │   Applet/
│                                      │   Application
├─────────────────────────────────────┤
│                                      │
│              JVM                     │
│                                      │
├──────────────┬──────────────────────┤
│              │         API          │
│              ├──────────────────────┤
│    O.S.      │                      │◄── map uniform
│              │    Peer methods      │    method
├──────────────┴──────────────────────┤    names
│                                      │    to specific
│             CPU                      │    functions
│                                      │
└─────────────────────────────────────┘
```
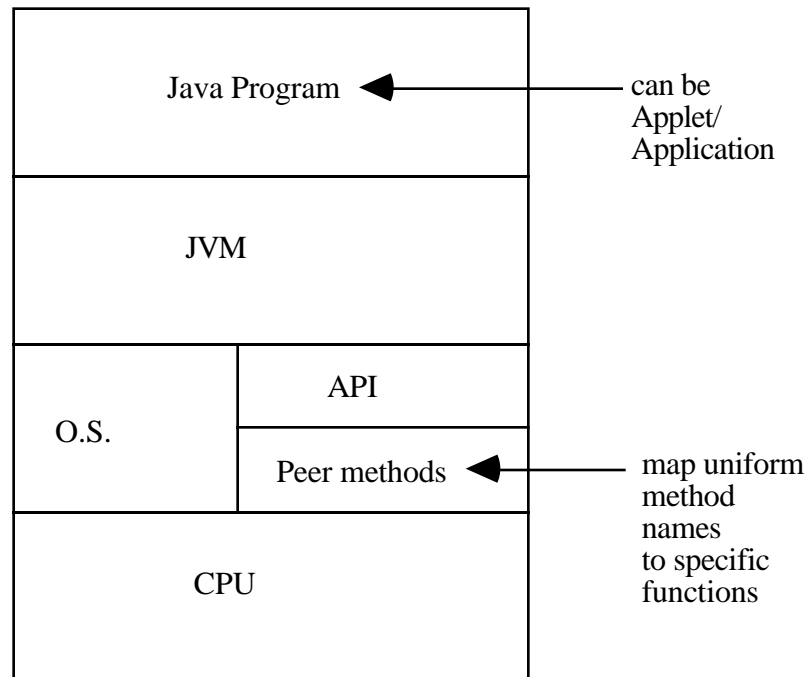
Fig. A.2-1 The Java model.

Java is popular for several reasons. A promise of Java is that it is multi-platform. Sorry to say, Java does not live up to it promise in this area. For example, Apple's management has underestimated the importance of Java to the Macintosh community. As a result, there are no Macs, as of this writing, that are able to run Java 2.

Fig. A.2-1 shows a substrate called *peer methods*. Peer methods provide an API whose goal is to provide a mapping from the high-level Java API to the low-level operating system subroutines. Peer methods provide the key to a portable operating system interface. They enable the creation of programs that have a look and feel of the host computer. For example, the *java.awt.FileDialog* on a Mac will bring up the *Standard File-Open dialog box*. This is the same dialog box that is shown in response to the C-language subroutine call. Thus the Mac look and feel can be supported within the Java programming environment.

Languages other than Java may exploit Java technology. For example, it is possible to implement a non-Java compiler that creates Java byte-codes, suitable for running on a Java virtual machine.

A.3. The 7 layer model & equivalent in real world

The OSI (Open Systems Interconnection) Reference Model is a 7-layer model adopted by the ISO (International Standards Organization). Each layer provides a level of abstraction that provides a well-defined function. The data that flows between the layers is identified by encapsulation with headers that increases the overhead of the protocol.

Application Layer
(Web browser)

Presentation Layer
(HTTP)

Physical Layer
(bits)

Session Layer

Data Link Layer
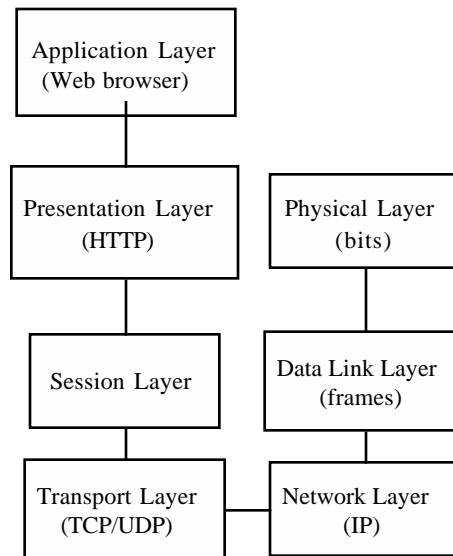(frames)

Transport Layer
(TCP/UDP)

Network Layer
(IP)

Fig A.3-1. The OSI Reference Model

Figure A.3-1 depicts the seven layers of abstraction in the OSI reference model. HTTP stands for hyper text transfer protocol and is considered to be a presentation layer protocol. TCP/UDP stands for Transport Control Protocol/User Datagram Protocol and is used in the Transport Layer. IP stands for Internet Protocol and is used at the Network Layer.

A.4. HTTP model v/s Java model

On the Internet, we find a variety of files. Data structures are saved in files and *REQUIRE DECODING*. The number of different data files that can be created is unbound. At any given time the number of file formats is countable, yet generally remains uncounted and increase in number at an unknown rate. There are few tools available to count the number of different file formats and even fewer that can decode many of them.

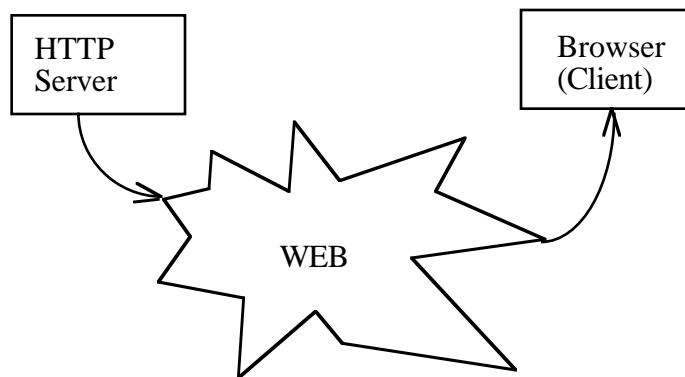Figure A.4-1 depicts the web model of data distribution on the Web.

Fig A.4-1 The Internet

With a browser, using "plug-ins" or format specific code is a static decoding. Different file formats (e.g. GIF, MPEG, QT, etc.) require the respective plug-ins (of "helper applications) for display. Thus, plug-ins extend a browsers capability or built-in algorithms.

A browser with Java is able to perform decoding using dynamically downloaded algorithms. This requires a means of running a decoder program on demand. To provide for security, Java provides feature limitations for the programs that are downloaded on demand. Such Java programs are said to be *untrusted*. A class

called the *security manager* enables precise control over the feature limits on an untrusted Java program.