

A Review of Microsoft's Internet Architecture

Kimble Consultancy Services Ltd

4th February 1997

<http://www.kimble.co.uk/>

Copyright Notice

This document is copyright © DB Harvey-George/Kimble Consultancy Services Ltd. No reproduction in whole or part without permission. Please Email info@kimble.co.uk for use.

Created on 05/03/99 4:18 PM

A Review of Microsoft's Internet Architecture

A REVIEW OF MICROSOFT'S INTERNET ARCHITECTURE	1
Scope	3
Introduction	3
The Client	5
The Server	7
Qu'est que c'est OLÉ?	9
Microsoft, ActiveX and Java	10
Microsoft and WebTV	13
Conclusion	13
References	13

Scope

This document reviews Microsoft's Internet architecture known as the Active Platform and contrasts it with other tools.

Introduction

Microsoft has learnt from Netscape - use accepted protocols, give the browser away for free then add extensions to make it attractive to buy their server products.

Late in 1995, Bill Gates made a keynote speech in which he highlighted the importance of the Internet to Microsoft. It would be fair to say that up to that date Microsoft had taken an ambivalent attitude to the Internet, preferring instead to concentrate resources on its alternative Microsoft network (MSN). However the day of the proprietary online service provider is past, this is witnessed by CompuServe's \$27 million dollar loss recorded for last year.

In the desktop arena Microsoft has taken advantage of its intimate knowledge of its proprietary APIs in order to extend its software business beyond operating systems to include applications software and developer products. The sheer economics of the Windows installed base convinces many developers to write for Windows alone. Gates saw the danger presented by the upstart Netscape Corporation aided and abetted by Sun Microsystems. Netscape gained huge market share by essentially giving its Web browser away for free, it then realised that it could practically build a desktop of its own within the browser framework thus breaking the Windows hegemony.

Since the date of Gates' speech Microsoft has refocused a considerable portion of its annual \$2 billion R&D program on Internet technologies. The Platform group alone has a research budget of over \$1 billion, of which \$160 million has been spent on Microsoft's Web browser (Internet Explorer). MSN has gone through a number of reincarnations, now finding a niche as an provider of Internet services with some extra local content. A sort of gated community of the Internet.

Microsoft couldn't have developed a complete Internet architecture in so short a time. Much of the technology is a repackaging of existing products, other components have been bought from third parties. For example, Internet Explorer is a development of the seminal Mosaic Web browser by way of Spyglass Inc.. Microsoft and Spyglass have just settled a dispute over royalty payments in Spyglass's favour to the tune of several million dollars, it would seem that Microsoft's boasts about Internet Explorer's popularity, a claimed 28% of the browser market, hadn't reached their own accounts department.

During the MSN era Microsoft had not been ignoring the Internet entirely, although anyone who bought Windows NT 3.5 hoping to get the power and utility of a Unix TCP/IP server may have thought so. The European Microsoft Windows Academic Centre (EMWAC), based in Edinburgh, were sponsored to provide various kinds of Internet server for NT. Thus Microsoft already had some experience developing Internet applications when Bill Gates experienced his conversion on the road to Damascus.

The components of Microsoft's architecture are in place although a full implementation won't be available

A Review of Microsoft's Internet Architecture

until **Internet Explorer 4.0** (aka Nashville) hits the super-highway. On the server side things are pretty much centred around the Windows 32-bit environment, and that means the premium priced **NT 4.0 server**. Microsoft's recent service packs have fixed the loop-hole discovered by O'Reilly and Associates whereby an owner of NT 4.0 workstation could 'upgrade' to the server version merely by changing an entry in the Window's registry.

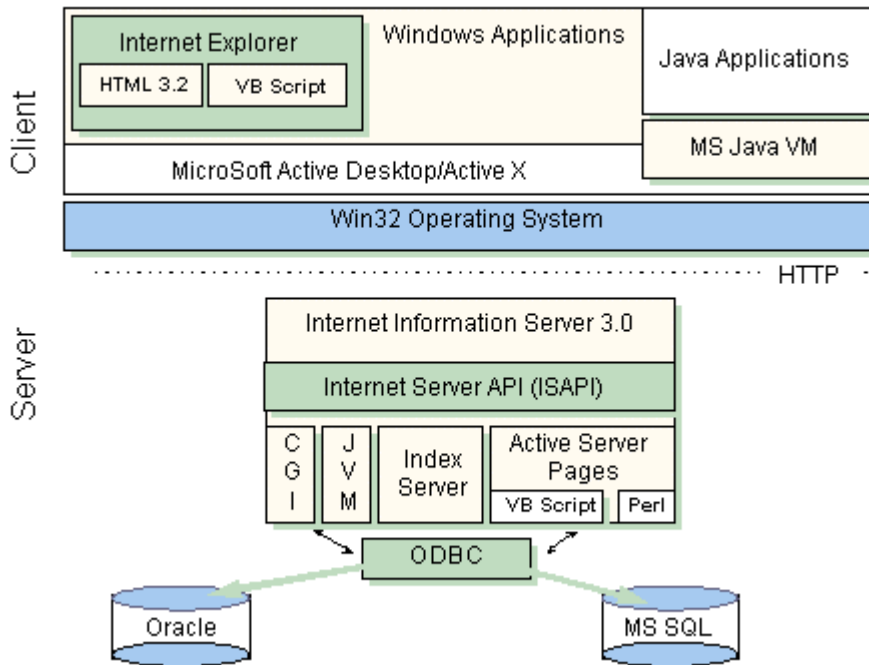


Figure 1: Microsoft's Active Platform Architecture

On the client side Microsoft has recognised that although Windows 95 may be an important operating system, it is not the only operating system on the Internet. Ports of ActiveX and Internet Explorer have either been completed or are underway for MacOS and Unix. Normally Microsoft out sources these jobs (as was the case for OLE on Unix) but they couldn't find any takers for the Unix port of IE and have been forced to do it in-house. A marked contrast to the last time Bill Gates was asked to write code for Unix.

Thus signing up to the Microsoft strategy means NT on the Server but a reasonable degree of freedom on the client, the jargon is **multi-platform** as opposed to Java's **cross-platform** approach. The Internet itself is all about openness and interoperability, it is quite possible to use any operating system on the server, but by building some proprietary extensions Microsoft has made it compelling to use their technology.

If things work out the way Microsoft wants this should all be academic to the client. Microsoft have learnt well from Netscape. Build on the accepted protocols, give the client side away free and add extensions to make it attractive to buy into the server side.

The Client

Microsoft's philosophy is to make the operating system the browser, Netscape's is to make the browser the operating system.

At the heart of Microsoft's client strategy is the Internet Explorer Web browser and the Active Desktop. Internet Explorer is derived from NSCA Mosaic's browser, the Web browser developed by Netscape's current Technical Director, Marc Andreessen. It's now in version 3.0 and presents a serious challenge to the dominant Netscape Navigator.

Active Desktop forms part of Microsoft's Active Platform and is a range of services that will be integrated into the 32 bit Windows operating systems as well as MacOS and popular variants of Unix such as Solaris, HP-UX, and AIX. The combination of these services takes place in Internet Explorer. In release 4.0 Explorer will see much tighter integration with the Win32 Operating Systems and this will be extended with subsequent releases of Windows 95 and NT.

Active Desktop offers the following services to developers:

- Support for the WWW consortiums (<http://www.w3.org>) latest HTML standard.
- Support for browser scripting based on Microsoft's own VB Script (derived from Visual Basic) and Netscape's JavaScript. The Microsoft implementation of JavaScript is about 90% compliant with Netscape's.
- Support for Java applets through the Microsoft Java Virtual Machine and JIT compiler technology.
- Support for ActiveX controls.
- Support above and beyond the lowest common denominator. Developers can access platform specific features such as DirectX APIs on Windows or Quicktime on the Macintosh.
- Leverage existing OLE component libraries available for Windows

Support for HTML 3.2 means that Web sites created with Internet Explorer in mind should also be viewable with other Web browsers ranging from the text-only Lynx through to Netscape Navigator. Internet Explorer probably features the best support for the current standard but other browsers will eventually implement missing features that are within the display capacity of client platform.

Browser scripting was introduced with Netscape's LiveScript (now JavaScript). It basically is a way to break the client/server bottleneck introduced by Common Gateway Interface programs. These programs are a way to link the Web into existing information sources such as a relational databases. However they result in a significant dumbing down of the client. A £1,000 PC is essentially turned into a block mode terminal. Browser scripts are included directly within Web pages, browsers that don't support scripting simply ignore them. Otherwise the browser can interpret the script directly to perform such actions as idiot checks and cross-referencing of forms data. Scripts can even access databases directly via ODBC

A Review of Microsoft's Internet Architecture

drivers, although this is of most relevance to **intranet** applications. Because Scripting languages are interpreted by the browser there is significant scope for security precautions against malicious or buggy code.

Feature List	Internet Explorer 3.0	Netscape Navigator 3.0
HTML Compliance	near HTML 3.2 with extensions	HTML 2.0 with many 3.2 features + Netscapisms
VBScript	Yes	No
JavaScript	90% compliant	Multi-platform incl. Win16
Java	Win32 only	Multi-platform incl. Win16
ActiveX	Win32/MacOS only	Win32 via plug-in
Admin Kit	free	\$1995
Cost	free	\$49/seat
Script Migration	VB Script/ActiveX to Win32	JavaScript/Java to all supported platforms
General	Slower than Netscape, now suffering from code-bloat. Poor Mail Client. Buggy.	Complicated setup, News interface hard to use, Excellent Web support.

Table 1: Battle of the Browsers

Microsoft supports Netscape's Javascript in the client although their version is based on an earlier standard is only about 90% compliant with Netscape Navigator 3.0. They also offer existing developers a way into the Internet/intranet through VB Script, a cut down version of Visual Basic.

Microsoft has licensed Sun Microsystems's Java technology and can now run Java Applets on the Windows 32 versions of their browser. This support will be extended to the MacOS, Unix and 16 bit Windows platforms. Microsoft's continuing interest in Windows 3.1 reflects the lack of success of Windows 95 on the corporate desktop and the need to capture a large slice of the intranet marketplace.

Java offers more than the scripting languages. It is a full application development language and can be used to build stand-alone products as well as Internet oriented applets. To date, most interest has centered on applet development. Stand alone Java coded products such as Java Workshop (Javasoft's development environment) have been too slow, lacking the advantage of direct operating system support and native compilation.

Like the scripting languages applets are interpreted directly by the browser. Once again this gives a good, but not totally foolproof, degree of security from viruses and rogue programs. However applets are compiled by the developer into a lightweight, platform-independent byte-code. This resembles a computer's native machine code. Consequently they are faster to load and run than scripts.

A Review of Microsoft's Internet Architecture

The range of applications that can be built with Java is much greater than with scripting languages. Through a class-library Java applets can access a spectrum of services including database connectivity, restricted portions of the native operating system APIs, the GUI and other Internet services. Java applets are built as a set of objects, called classes. To improve download times over slow networks classes are only loaded as required. Frameworks such as JavaBeans are expected to create an industry for classes just as OLE has done for Visual Basic components.

The final plank in the client strategy are ActiveX components. ActiveX is a subject in itself (see below) but basically these are compiled modules that give 'applet' developers access to a much richer set of resources than would be possible through Java's security restricted and cross-platform model. Again whole applications can be written using ActiveX technology.

Both Java and ActiveX have brought dynamic (or active) content to the Web and have unleashed the power of client systems. Applications can range from intelligent forms through to mortgage calculators and from direct database access up to spreadsheets and word processors (Corel have rewritten their office suite in Java). Given security and bandwidth considerations ActiveX is probably more of an intranet as opposed to an Extranet/Internet technology when compared with Java. The Halifax Home Arranger sits between an intranet and extranet, that is, it will be used over the organisation's high speed ATM network but may also be fully accessible to customers with a PC connected to the Internet.

For organisations interested in corporate image Microsoft supply a browser administration kit. This allows key features of Internet Explorer (itself a set of ActiveX components) to be modified. For example the IE logo can be replaced with the corporate logo. This is of interest where the browser will be distributed on CD-ROM, perhaps as part of a hybrid Web site. This could include bandwidth hungry components such as ActiveX mortgage calculators or Windows movies on the CD-ROM, linking back to the Web pages for the latest information. Microsoft actively encourage organisations to modify and distribute their browser and give the administration kit away.

The Server

Microsoft gives its browsers and servers away for free, because this adds value to Windows platform and other Microsoft products at the same time as undermining Netscape's core business.

On the server side Microsoft's efforts are based around its Internet Information Server (IIS). First released with Windows NT 4.0 beta in early 1996, it is now on version 3.0. IIS is a Web, FTP and Gopher server rolled into one. The Web side of things is of most interest. As well as supporting the standard Hypertext Transfer Protocol (HTTP) for serving static documents the IIS implements the Common Gateway Interface (CGI), a widespread method for Web servers to communicate with backend programs such as databases. A large number of CGI scripts have been written and are freely available. Perl is a popular language for implementing CGI scripts as it:

- free
- widely available

A Review of Microsoft's Internet Architecture

- has powerful text processing features (and the Web is text, albeit hypertext)
- is interpreted, which suits the RAD nature of many Web developments

Microsoft have recognised the need for a fully featured version of Perl for the Win32 platforms and have sponsored HIP communications to port the latest version, 5.0. HIP have added OLE (see below) support to Perl, this allows it to use facilities from Windows programs such as Excel.

However when running, CGI scripts place a considerable load on the Server machine. In addition they, and HTTP, are not suited to operations that involve a number of Web screens where some knowledge about the client (state) must be retained between each operation. The interactive aspects of the Halifax home arranger are an obvious example.

Microsoft have borrowed an idea from Netscape with its Information Server API (**ISAPI**). Rather than construct their Web servers as monolithic entities. Netscape identified seven steps needed to satisfy any browser request. Each of these stages is implemented as a dynamic library with a clearly defined interface. If you want to change the way Netscape logs user requests simply write a new library based on the published API. The server will replace its own code with this library when it boots up. ISAPI is similar to NSAPI in functionality, both obviate the startup overhead of CGI programs and can maintain state about client connections. The downside is that ISAPI and NSAPI programs are not portable to other Web servers and if buggy can crash the server itself.

Active Server is part of the Active Platform released with IIS 3.0 and includes:

1. Active Server Pages
2. Microsoft Transaction Server
3. Message Queues.

Only Active Server Pages and Microsoft Transaction Server are currently available.

Active Server pages enable developers to create dynamic Web sites using VB Script or ActiveX components. ActiveX components can be written in a variety of programming languages and can communicate with other services running on the computer. For example an ActiveX component could embed an Excel chart into an HTML document.

Leveraging off existing Visual Basic and C++ development enables Microsoft to claim that there are already more than 2,500 ActiveX components available off-the-shelf. How relevant each one is to the Web remains to be demonstrated but it certainly holds out the prospect of being able to construct a dynamic Internet site by plugging together ready made modules.

Active Server uses the same component-based development model as the Active Desktop, this permits applications to be migrated from client to server. Netscape achieves the same portability with JavaScript and Java and the number of ready-made Java applets and classes is growing.

Qu'est que c'est OLE?

All ActiveX components are created equal but some are more equal than others.

ActiveX is a reworking of Microsoft's Object Linking and Embedding (OLE) and Common Object Model (COM) technologies. ActiveX provides a communications framework for bits of code referred to as components. A component implements some piece of functionality, for example, viewing a Word document within a browser. The browser simply becomes a frame in which the document and its application are displayed. ActiveX components can serve as building blocks for larger applications, by assembling components developers can create more complex features without writing any code.

Netscape **plugins** offer some of the functionality of ActiveX, they permit developers to extend the range of document types that can be viewed directly within the browser. Adobe Acrobat is such a plugin for viewing Portable Document Files (PDF). Plugins permit a good deal of integration with Web HTML documents. However installation is not transparent to the user as is the case with (small) ActiveX. ActiveX permits the browser to update itself without direct user intervention. Plugins must be downloaded and installed by the user. Plugins are also written as large, monolithic libraries and involve considerable development effort.

More obvious competitors are Apple's **OpenDoc** (although with its current financial troubles and halving of market share in a year who would bet on Apple?) and **JavaBeans**. JavaBeans are Java classes that implement features such as *introspection*. This enables developers to build applications from beans classes sourced from different vendors. Extra wrapper code can be added to customise functionality. Using wrappers Beans classes can even interoperate with ActiveX components. The Java Virtual Machine provides the environment in which beans can interoperate.

The main advantages of beans is that they are written in Java. They will run anywhere there is an implementation of the Java Virtual Machine. This includes all Windows platforms, most Unixes and even some large systems. With Just in Time compilers and integration of the JVM into some operation systems Java looks like an attractive route. However there are very few beans available (the much maligned Java AWT, a set of GUI components, being one of the few examples).

A Review of Microsoft's Internet Architecture

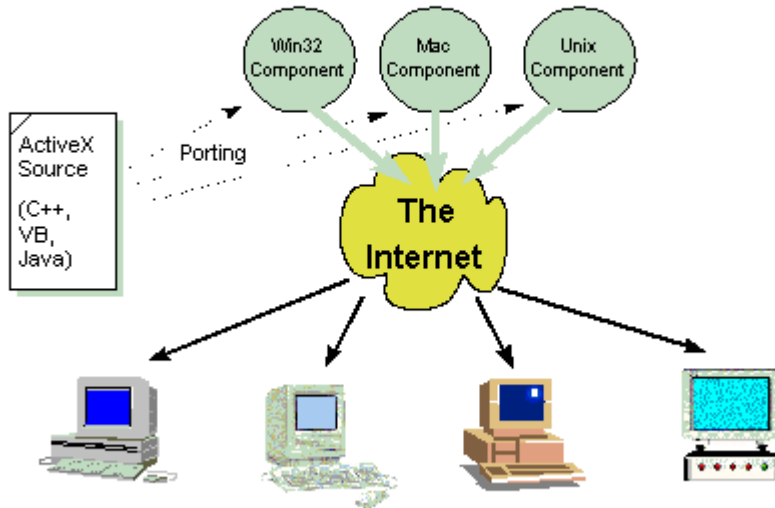


Figure 2: ActiveX Multitplatform Components

ActiveX components are native bits of code, they must be built for each and every operating system on which they will run. They can be written in a variety of languages including Visual Basic and Visual C⁺⁺. Microsoft is committed to providing an ActiveX framework on its own 32 bit Windows OSes, MacOS and popular Unixes. However there will still be difficulties in porting ActiveX components from one OS to another. The portability of components depends on how many of the thousands of Windows APIs Microsoft can, or cares to port to MacOS and Unix. Microsoft's openly admit that "*Windows is the best client on which to run Active Platform applications and has a current installed base of over 100 million users.*". ActiveX does leverage the exiting OLE code base and provides an easy migration path to the Internet for the large number of Visual developers.

Microsoft, ActiveX and Java

JavaSoft sell Java under the banner 'write once, run anywhere'. Microsoft's philosophy is to 'embrace and extend' Java by adding ActiveX support. This entrenches the norm of 'write once, run Windows'.

So if ActiveX is just an infrastructure, a way for components to communicate, where does that leave Java. Well at its core Java is a member of the C/C⁺⁺ programming language family. That means that ActiveX components can be written in Java and compiled into an executables for the target platform. Neat. So Java programmers can also be ActiveX programmers, everyone's happy. Well, maybe not.

A Review of Microsoft's Internet Architecture

Java is more than a programming language, its a new way of building programs. The idea is that programs can be written in Java, a language that requires coders to understand and use object oriented (OO) techniques. OO has been lauded as the productivity silver bullet but has failed to live up to its potential due to poor support in C++, the industry's language of choice.

Once written Java source code then get compiled into a platform independent byte code. This code cannot be run directly on a computer, or at least not until Java Actual Machines (JAMs) such as Sun Microsystem's picoJava chip become widely available. Instead the byte code is interpreted by another program running directly on the computer, this is the Java Virtual Machine (JVM). The only thing needed to run Java on a computer is a version of a JVM. This is the idea behind *write once, run anywhere*. Compare this to ActiveX where each component has to be specifically compiled for the target machine. Even with Unix and Mac support it is unlikely that ActiveX will ever be as widespread as Java has become.

Microsoft has adopted the 'embrace and extend' philosophy common to many legacy software companies faced with the challenge of the Internet. To Java, this means producing its own Java compilers and interpreters. Microsoft have even cocked-a-snook at Sun by labeling their own Java Virtual Machine the 'Microsoft Virtual Machine'. Microsoft claim to have fixed problems with Sun's original JVM implementation and have also optimised their JVM for Windows. They also implemented a technique called Just in Time compilation (JIT). The JIT turns Java bytecodes into native machine code that can be run directly by the CPU. This translation is performed by the JIT on-the-fly as the bytecodes are downloaded over the network.. Interpreted Java is estimated to be about 20 times slower than compiled C code, JIT can speed up execution by a factor of 4.

So where's the danger? Well Microsoft have also integrated Java and ActiveX. Programmers who use the ActiveX extensions will no longer be writing portable code. Microsoft would probably answer that Java is only portable as far as a JVM is implemented on the target platform. There is also a strong case that portability tends towards a lowest common denominator making some tasks difficult or impossible to achieve. It is probably fair to say that anyone using ActiveX extensions should realise what the implications are in the same way that the implications of using Java's native methods (a way of calling directly calling operating system services) are understood.

A Review of Microsoft's Internet Architecture

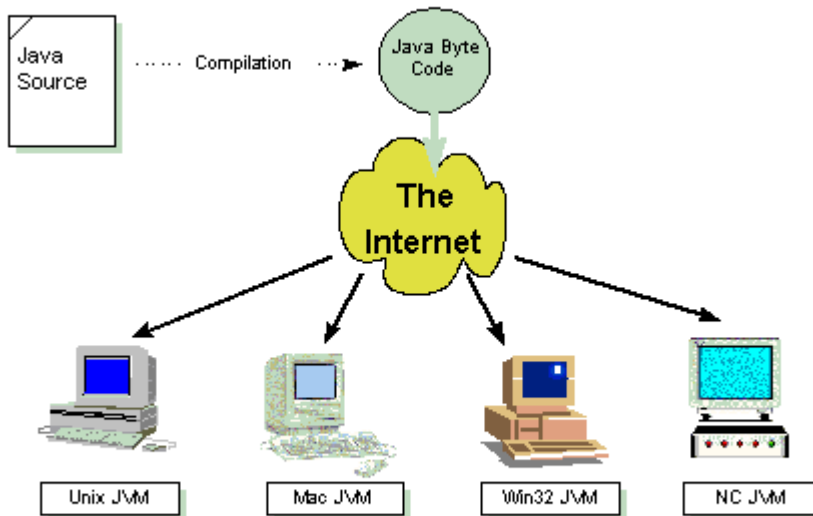


Figure 3: Java Cross-Platform Components

A more subtle danger are Microsoft's Java libraries. Microsoft has expressed its dissatisfaction with Sun's GUI classes, the Abstract Windowing Toolkit (AWT). Instead it has written its own toolkit, the Application Foundation Classes (AFC). The claim is that these provide a much richer framework for building real-world applications such as spreadsheets or word processors. Even the most die-hard Java fanatics would acknowledge the limitations of the AWT and even though the latest release (1.1) is much better for building larger systems the look and feel still leave something to be desired.

Microsoft's AFCs are written entirely in Java, so like any other Java program they should run anywhere. The danger is they will run better on a Windows platform and will run better on Microsoft's Virtual Machine. Microsoft has managed to garner some support from third parties, given sufficient market penetration it would be easy for Microsoft to modify the Java byte code so that Microsoft Java programs don't just run better on Microsoft's Virtual Machine, they **only** run on Microsoft's Virtual Machine. There is competition in the marketplace, Netscape has their own Internet Foundation Classes (IFC) which are slated for support in the leading Java integrated development environment (IDE): Visual Café Pro.

The Java standard is controlled by JavaSoft, a Sun Microsystems subsidiary. JavaSoft says that control is necessary to stop organisations adding proprietary extensions. They intend to place some of the virtual machine standard under the control of the Open Group (a collaboration between X/Open and OSF) this year. To allay fears of Microsoft gaining advantage over other developers through early knowledge of the ActiveX APIs they have placed control of the ActiveX standard with an independent group (<http://www.activex.org>). Crucially, however, Microsoft maintains control over many of the Windows APIs on which ActiveX depends.

Microsoft and WebTV

Microsoft have had a long term interest in the set-top box and appliance market place. Interestingly Java was also developed as an operating environment for STBs. Recent developments in the Internet with the browser and network computer markets have threatened to encroach on a market over which Microsoft had already staked a claim.

Microsoft has cast around for some strategic alliances and in the Autumn of last year announced a partnership with WebTV. WebTV has a well developed network of content providers aimed at their own set-top box and browser. Microsoft and WebTV have agreed to develop Internet Explorer for the next WebTV browser. Exact details of what the browser will support are currently unavailable

Microsoft also has a cut down version of the Windows 95 operating system that it would like to interest STB and Internet Access Device vendors in. This has been developed to provide the small memory footprint required by such devices. It is also portable across the range of CPUs found in such devices. Expect versions for Acorn's ARM processors. WindowsCE can run many Microsoft Applications including Internet Explorer. However the CE version understands only a subset of HTML and in its current incarnation cannot run JavaScript, VBScript, ActiveX or Java.

Conclusion

Microsoft provide a range of components that can help construct interactive Web sites. The tools leverage off an existing code base and pool of developers. However the suitability of this code and the developers for Internet oriented tasks is unproven. Adopting a Microsoft only strategy limits developments to Windows NT on the server (compare this with Netscape's support for NT and 12 flavours of Unix) and possibly limited functionality on non-Win32 platforms on the client. With Internet Explorer usage under NT and 95 around 25% Microsoft are not currently in the position of strength they enjoy in the PC environment.

References

Byte, January 1997

Microsoft Interactive Developer, Spring 1996

Fortune Magazine, 30th September 1996

JavaWorld, February 1997 edition (<http://www.javaworld.com>)

Microsoft Web Site (<http://www.microsoft.com>)

Building Secure Internet Servers Using Netscape, QA Training Publication, Harvey-George /Richardson.