C.   Cross-Platform Java also presented a middleware threat to Microsoft's operating system monopoly

57. Cross-platform Java is another middleware technology that has the potential to erode the applications barrier to entry by gaining widespread usage of APIs without competing directly against Windows as an operating system.

1.   The nature of the Java threat

58. James Gosling and others at Sun Microsystems developed Java in significant part to provide developers a choice between writing cross-platform applications and writing applications that depend on a particular operating system.

58.1. Java consists of a series of interlocking elements designed to facilitate the creation of cross-platform applications, *i.e.* applications that can run on multiple operating systems.

i.   Gosling testified: "The Java technology is intended to make it possible to develop software applications that are not dependent on a particular operating system or particular computer hardware . . . . A principal goal of the Java technology is to assure that a Java-based program -- unlike a traditional software application -- is no longer tied to a particular operating system and hardware platform, and does not require the developer to undertake the time-consuming and expensive effort to port the program to different platforms. As we said in the Preface to *The Java Programming Language*, 'software developers creating applications in Java benefit by developing code only once, with no need to 'port' their applications to every software and hardware platform.' . . . Because the Java technology allows developers to make software applications that can run on various JVMs on multiple platforms, it holds the promise of giving consumers greater choice in applications, operating systems, and hardware. The Java technology has the potential not only to free individual consumers from concern about whether the software they want to run is supported by a given operating system, but also to permit corporations and Internet users more easily to mix different types of computing systems across a network." Gosling Dir. ¶¶ 20-29(b); <u>see</u> <u>also</u> Gosling, 12/3/98am, at 6:3-6.

ii.   Gosling stated this theme in internal documents as early as August 1995: "The

issue of making developers CPU and OS independent is that they can port to Sun or to Windows. . . . Sun's or any alternate CPU company (MIPS and SGI) key to success is apps. Apps are the key to volume. Java allows developers to decrease their dependence on Intel and Microsoft." DX 1285; see also DX 2012, at SUN 87 001685 ("Sun is attempting to establish Java as a viable computing platform which is hardware and operating system independent.").

iii.   In his June 1999 rebuttal testimony, Dean Schmalensee recognized "Java is used for a wide range of things. It's how my son first learned computer programming. It's used to run on a wide array of platforms. That, of course, is one of its most important selling features from Sun." Schmalensee, 6/23/99pm, at 50:5-11.

iv.   As the District Court for the Northern District of California found: "Sun's JAVA Technology comprises a standardized application programming environment that affords software developers the ability to create and distribute a single version of programming code which is capable of operating on many different, otherwise incompatible, system platforms and browsers. Most computer systems implement platform-dependent programming environments, such as Microsoft's Win32 programming environment. Programs created to run on a particular platform will not function on a different platform. Thus, a software developer must choose the platforms for which it will develop and support different versions of the same program. Sun's platform-independent JAVA Technology, which can be implemented on many different system platforms and browsers, obviates the need for creating and supporting different versions of the same program." Sun Microsystems, Inc. v. Microsoft Corp., 999 F. Supp. 1301, 1302 (N.D. Cal. 1998).

58.1.1. Java provides ISVs a programming language with which to write applications. Java also includes a set of "class libraries," a collection of programs written in Java, that offer APIs that ISVs can use to develop software applications.

i.   Microsoft's Paul Maritz summarized the different aspects of Java: "Java, the programming language; Java the virtual machine, which you need to execute Java programs; and then there's this collection of other programs written in Java, which I call the Java classes. And it's that collection of software that is being put forward by Sun, and that Netscape has announced their intent to cooperate with Sun, that forms

139

another body of middleware that I am concerned about." Maritz, 1/26/99am, at 18:22 - 19:23.

ii.   As the District Court for the Northern District of California summarized: "The Java programming environment allows software developers to create a single version of program code that is capable of running on any platform which possesses a compatible implementation of the Java runtime environment. The Java programming environment comprises (1) Sun's specification for the Java language, (2) Sun's specification for the Java class libraries and (3) the Java compiler." Sun Microsystems, Inc. v.
Microsoft Corp., 21 F. Supp. 2d 1109, 1112 (N.D. Cal. 1998).

58.1.2. The Java programming environment also provides software developers

a Java "virtual machine" (JVM) that, when ported to different operating systems, serves as the "host" or

"adaptor" to which programs written in the Java language can be run, independent of the underlying

operating system. Together, the Java class libraries and virtual machine are often referred to as the

"Java runtime environment" (JRE).

i.   As Gosling explained, Java makes it much easier for ISVs to develop platform-independent software because Java programs "need not run by interacting with a particular operating system's APIs. Instead, they typically interact with a Java virtual machine ('JVM'), which is an intermediate software layer that translates the Java-based program for the particular operating system and hardware platform that the Java virtual machine runs on. In essence, the Java-based program views the JVM as an operating system, and the operating system views the JVM as a traditional application . . . once a JVM is developed for a software platform, if the JVM is fully compliant with the Java specifications, . . . it should run most Java-based programs without the need to recompile or otherwise modify the programs. . . . Such programs thus have the potential to run on any PC, other type of computer, or even devices not traditionally thought of as computers (such as cellular telephones), provided that the machines have compatible JVMs installed on them."; see also Gosling Dir. ¶¶ 24, 25, 28; Gosling, 12/2/98am, at 55:5-11 (any Java program, if written properly and properly compiled into bytecodes, should run equivalently on any properly-designed and

140

implemented Java virtual machine, regardless of the underlying platform is); Gosling, 12/10/98pm, at 21:8-14 (describing the JVM as an adaptor).

ii.   Sean Sanders, an executive with Novell Corporation, testified: "The Java virtual machine is essentially yet another software layer that allows people to run Java-based applications and to help them -- provide them the tools that they could use to yet build and develop and to -- for optimization of any other Java applications that they might want to develop." Sanders Dep., 1/13/99, at 188:18 - 189:15.

iii.   The District Court for the Northern District of California described the Java architecture as follows: "Sun's JAVA Technology is a so-called 'class-based' language in that its functionality is determined by the Java classes available to the programmer. Therefore, new functionality requires developing new Java classes. Programs written in the Java programming language are compiled into intermediate instructions called bytecodes or Applets. These bytecodes or Applets are then 'interpreted' by another computer program which emulates a hypothetical CPU called the Java Virtual Machine. The Java Virtual Machine translates the Applets into instructions understood by the specific computer CPU on which the Java Virtual Machine is running. Therefore, a specific interpreter or virtual machine is needed for each computer CPU on which the Java program is run." <u>Sun Microsystems, Inc.</u> v. <u>Microsoft Corp.</u>, 999 F. Supp. 1301, 1302-03 (N.D. Cal. 1998) (citations omitted.).

58.2. Because Java offers alternative APIs, applications written using standard Java programming tools and class libraries can run on any operating system for which there is a Java virtual machine. The widespread adoption of a cross-platform Java programming environment could reduce computer users' dependency on the Windows operating system.

i.   As Gosling explained: "As more new Java-based programs are developed, distributed and used, new operating systems may be developed to take advantage of the existing body of Java-based software. In other words, potential developers of new operating systems and hardware platforms need not be deterred by the absence of platform-specific programs for their new systems, so long as there is a JVM available to enable existing Java programs

141

to run on the systems. This may give new operating systems and hardware platforms a chance to compete in markets previously dominated by a particular vendor." Gosling Dir. ¶ 29; Gosling, 12/10/98pm, at 28:20 - 29:2 ("Once the APIs that developers develop to are ones that are realized on many different operating systems, then those operating systems can compete with Windows. And that would lead to sort of a lesser -- lesser role for Windows in that they would have to compete with these operating systems on the merits of the operating system rather than on the lock that tends to be inherent in the APIs and the binary compatibility.").

ii.   Soyring explained IBM's rationale for supporting Java: "Primarily because the value that it provides to IBM's customers and the value it provides to IBM. As you probably know, IBM has a variety of operating systems, primarily four different ones. Many of our customers have many of these different -- several -- one or more of these operating systems installed. It's less expensive for them and less time-consuming for them to be able to buy one application or one software product that they can buy, maintain, and support but run it on different operating system platforms." The success of cross-platform Java would enhance the ability of other operating systems "to compete." . . . What drives demand for the sales of operating systems is the availability of applications. And if there is a large install base of Java that's consistently implemented, what it does is create an economic opportunity for commercial software developers to be able to develop a commercial software application using Java and then make it available to sell and be run on many different operating systems rather than just on one." Soyring, 11/18/98pm, at 54:8 - 55:10; see also Soyring Dir. ¶ 28 ("The Java technology from Sun is designed to allow Java-compatible application programs to run on a wide variety of different hardware and operating systems. This would provide users with the benefits of increased number of applications and would reduce the cost of ISVs of developing applications for multiple operating systems. This characteristic of Java also has the potential to undermine the Windows application advantage . . . .").

iii.   Barksdale testified that "the cross-platform benefits of Java, allowed for the development of software applications that were directed more to the Internet than to the desktop, and thus had the potential to serve as a partial substitute for the Windows OS as a development platform." Barksdale Dir. ¶ 15; see also Sasaki Dep. (played 12/16/98pm), at 31:24 - 32:7 (explaining that Java has the potential to level the playing field among operating systems).

iv.   Dean Schmalensee readily acknowledged that cross-platform Java technology poses a competitive threat to Windows by potentially rendering underlying

operating systems less significant: "Sun's Java poses potentially serious competitive implications for Windows. . . . If Java achieves its advocates ambitions, operating systems would become less important in the marketplace, and the important standards would come to be determined by Sun, which vigorously defends its control over the Java language." Schmalensee Dir. ¶¶ 141-142; see also Warren Boulton, 11/19/98pm, 31:8-12 (concluding "an increasing number of users may be able to simply do without Windows entirely").

2. Microsoft recognized the Java threat

59. Microsoft understood the threat Java posed to its monopoly power. Java offered ISVs the ability to create a robust set of cross-platform applications that might reduce the applications barrier to entry.

59.1. Java provided software developers with a platform to create applications that could run on different operating systems and hardware platforms.

      i.   Eubanks testified: "One of the great things with Java is that when you create a Java application, it will run on any machine that has a Java virtual machine." Eubanks, 6/16/99am, at 68:11-20.

      ii.   Gosling, whose responsibilities include working with numerous application developers and who himself has a career of experience as a developer, made clear that Java's theme of "'write once, run anywhere' was terrifically attractive to developers. Developers want this more than just about anything you can imagine." Gosling, 12/3/98am, at 32:10-12.

      iii.   Soyring testified: "'Write once run everywhere' . . . has been the holy grail of programming for many years to be able to write an application once and then run it on many different operating systems or hardware platforms, and we find that Java is a technology that most closely approaches this by a long distance versus any other technology, and we have been able to successfully demonstrate with a set of our clients that it is possible using the Java technology to write an application once, compile it once and then run that exact same code on a variety of different operating systems, giving our customers the choice to choose different operating systems and different hardware platforms." Soyring, 11/18/98pm, at 51:18 - 52:6; see also Sanders Dep., 1/13/99, at 186:20 -

187:3 (stating that Java "provides the benefit of an application-running environment that would allow people to run applications independent of any kind of operating system or cpu type of restraints they may currently be facing").

    iv.   Barksdale testified that "Java allows software developers to write cross-platform applications that will run on any operating system, increasing consumer flexibility and ease of use, while reducing development costs associated with writing an application and then porting it to run on various different operating systems . . . . The Java programming technology solves the platform dependency problem that has so long plagued software development. Programs written in Java can be run on any platform that has a Java virtual machine and Java class libraries, which Navigator does." Barksdale Dir. ¶¶ 15, 83.

59.2. Microsoft recognized, and continues to recognize, the competitive threat that

Java poses to Windows by providing an attractive cross-platform programming environment that could

erode the applications programming barrier to entry.

    i.   Dr. Warren-Boulton summarized the evidence of Microsoft's perception of the competitive threat posed by Java: "Microsoft has, almost from the beginning, recognized that the clearest threat to that monopoly power is the emergence of Java technologies combined with an independent browser market. Their response to that threat has been to attempt to take that technology, and instead of making it multi -- cross-platform, has been to transform that technology into a technology that is Windows-specific so as to prevent the emergence of a large stock of applications that could be used on any operating system . . . ." Warren-Boulton, 12/1/98am, at 19:24 - 20:8.

    ii.   Dr. Fisher, similarly, summarized the evidence that Microsoft treated Java as a significant competitive threat to Windows. Fisher Dir. ¶¶ 204-207.

    59.2.1. Microsoft executives have throughout the past four years treated

cross-platform Java as a serious threat to Microsoft's operating system dominance.

    i.   In a June 1996 e-mail to Microsoft executive staff, Paul Maritz focused on the need to "fundamentally blunt Java/AWT momentum and to re-establish ActiveX and non-Java approaches as a viable strategy for

144

structuring software." The reason Mr. Maritz provided for this objective was to "protect our core asset Windows -- the thing that we get paid $'s for. While Java per se is not the problem, if everything & everybody moves to Java as a language, then it will be so much more easy for AWT to become the API, and Windows is damaged." GX 42, at MS6 6010347 (emphasis in original); see also GX 473, at MS6 6006237 ("Java. Gaining as scripting language . . . Class libraries define 'API.' Becoming the 'brand' for software components."); GX 504, MS98 0169096 (Maritz writing that:"Java. Sun's goal is: -- Java class library/runtime = new OS API -- leverage this new API to replace Windows by JavaOS.").

ii.   In August 1996, Bob Muglia wrote: "When a Java developer writes to AWT, they are writing to Sun APIs, and their application can be easily run on competitive platforms." GX 466, at MS6 5003781; see also Muglia, 2/26/99pm, 10:5 - 11:22 (by offering an alternative platform, Sun could get developers to write to the Java platform and not to Windows, and therefore the applications that they wrote would not be focused on Windows).

iii.   In September 1996, Adam Bosworth sent Bill Gates and others an e-mail discussing Java. Bosworth noted, "I think it is important to understand that Java is not just a language. If it were just a language, it would not be a threat to us. We would and could easily just build the best implementation of this language and be done. It is, however, much more. It is an alternative to COM. . . . Java is on Unix and requires no dealing with setup, install, de-install, or anything else. Thus it is really easy to understand how a system for dynamically authoring Web pages on the server that depended upon Java objects rather than COM ones would have wider appeal."Gates responds: "This scares the hell out of me. Its still very unclear to me what our OS will offer to Java client applications code that will make them unique enough to preserve our market position. Understanding this is so important that it deserves top priority." GX 983, MS7 032895.

iv.   In January 1997, an internal Microsoft analysis described the "platform challenge" posed by Java: "possible emergence of a set of APIs and underlying system software that lead to a lesser or no role for Windows." GX 51, MS7 005534.

145

v.   In a February 1997 e-mail to Jim Allchin, Mr. Gates again addressed the cross-platform threat posed by Java: "What will we have that the Java Runtime will not have? . . . The fact is that applications can be run on the server against an HTML client. . . . Most applications will have very little client code in the future. . . . The fact is there will be lots of machines where HTML/some level of Java is all they will have in common. Cheap devices and old PCs will be like this. It makes it very easy for people to think they should just program to this. . . . Lets work together to find the solution to this. I can say I am more scared than you are but that is not what will help us figure out where we should go." GX 475; see also GX 590 (Gates writing: "Java is the biggest threat to us and I certainly shouldn't be doing Apple events unless we are getting some help from us on this.").

59.2.2. Microsoft's witnesses in this litigation conceded that Java presented a

significant potential threat to Windows.

i.   Gates testified repeatedly that he perceived Java to be a threat to Windows: Gates Dep., (played 12/2/98am), at 22:19 - 23:1. Gates stated: "we did think of" Java APIs "as something that competed with us for the attention of ISV's in terms of whether or not they would take advantage of the advanced features of Windows." Gates Dep. (played 12/2/98am), at 24:15-22; see also Gates Dep., 8/27/98, at 90:12-19 (DX 2568).

ii.   Muglia also testified that Microsoft considered Java a serious cross-platform threat: ""Although Java was a new and unproven technology, Microsoft took Sun's claims seriously. . . . Sun has adopted a business strategy that seeks to transform the Java programming language into a full operating environment and software development platform. A key requirement of Sun's strategy is delivering on its WORA claim -- that programs written in Java, to the Java development platform, will run without modification on any underlying platform for which there is a JVM." Muglia Dir. ¶¶ 8, 10; see also Muglia, 2/26/99pm, at 4:8-18 (Muglia believed in 1995 and 1996 that Java represented a serious threat to Microsoft's operating system business); Muglia, 2/26/99pm, at 7:2-19 (explained that the cross-platform threat consisted of the JVM and Java class-libraries) ; Muglia, 2/26/99pm, at 9:3-21 (explained that: "what they were trying to do was get developers to write to that alternative platform. So, even if -- even if a

developer wrote a Java program and that program runs on Windows, even in the case where it runs on Windows, it's not written to Microsoft's programming interfaces. So when I said slide in their platform, what I meant is that they could, in essence, make what everything else that our platform did irrelevant, thus enabling to replace Windows and make it obsolete, so to speak.").

iii. Maritz also testified: "If successful, software developers could write programs to run on Sun's technology, and neither Windows nor any other operating system would provide significant value to customers." Maritz Dir. ¶ 243; Maritz, 1/26/99am, 20:23 - 21:3 (the Java foundation classes posed a potentially serious platform threat); Maritz, 1/28/99am, 59:10 - 60:17, 62:3 - 63:17 (Maritz explained Java is a form of middleware. Sun's goal was to provide most of the OS services through the Java runtime. The browser and Java have the potential to serve as a virtual operating system.).

iv. Dean Schmalensee also acknowledged the cross-platform threat Java poses to Windows: "Sun would like ISVs to write pure Java so that their applications can run anywhere, in principle. Microsoft would like ISVs to design applications that would run on Windows. It matters to those companies what choice the ISV makes, assuming it's a good application." Schmalensee, 6/22/99pm, at 23:23 - 24:7.

v. Slivka also testified regarding the Java threat: "my recollection was that this cross-OS Java platform stuff that we were attempting to do with AFC, he [Bill Gates] was very unhappy with that." "He thought that was a big threat to Windows." Slivka Dep., 9/4/98, at 367:13 - 369:3 (DX 2591); see also, Slivka, 1/13/99, at 735:13 - 736:4 ("All this comes back to Windows and the threat, you know, Sun's very direct threat to our Windows platform, and the success of Windows on the client. So, this seemed like if the library space fragmented, the 'write once, run anywhere,' I guess, actually is what Sun called it, that would be a lot less probable . . . I guess the end was to protect the Windows franchise, not to defeat the 'write once, run everywhere.'").