Assessing Tying Claims in the Context of Software Integration: A Suggested Framework for Applying the Rule of Reason Analysis

David A. Heinert†

Over the last six years, Microsoft’s releases of new versions of its Windows operating system products have been met with regulatory investigations and litigation in the United States, Europe, and Asia.1 Regulators and private plaintiffs have focused on Microsoft’s inclusion of functionality in Windows where competitors offer arguably comparable functionality on a standalone basis. Major government antitrust actions have challenged the design of Windows as unlawful tying. In resolving these claims, tribunals here and in Europe have applied varying and conflicting tests. As a result, the application of tying law to product design decisions remains an unsettled area of antitrust law—and an important one given that the adoption of an overly restrictive test could have a chilling effect on product innovation in computer software and hardware, consumer electronics, and high-technology products generally.

This Essay describes the various tests that have been applied in the proceedings involving Microsoft, and in an earlier round of cases involving IBM, and proposes a framework for analyzing such tying claims—sometimes called “technological tying” claims—in the future. Drawing primarily upon the recent cases involving Microsoft, this Essay focuses upon tying claims concerning software integration (particularly platform software integration), although the suggestions here may well have broader applicability to technological tying claims generally.

In the United States, the D.C. Circuit has held that the rule of reason governs the legality of alleged tying arrangements involving platform software.2 This Essay contends that the rule of reason is the appropriate test for judging tying claims involving software integration. In assessing potential anticompetitive effects, courts should be

† Deputy General Counsel (Antitrust), Microsoft Corporation. The views expressed here are my own. I would like to thank Richard C. Pepperman II of Sullivan & Cromwell for his assistance in preparation of this Essay.

1 The United States and European proceedings are described in this Essay. In addition, the Korean Fair Trade Commission is assessing the inclusion of certain functionality in recent versions of Windows.

2 See United States v Microsoft Corp, 253 F3d 34 (DC Cir 2001) (Microsoft III).
alert to the possibility that an integrated product design need not foreclose sales of competitors’ products and, indeed, may create opportunities for competitors by providing new platform capabilities that they (and others) may draw upon in developing their own products. These platform benefits and other procompetitive efficiencies should be judged under a deferential standard by courts because product improvement through the addition of functionality is commonplace in information technologies industries, product improvement is presumptively efficient, and the cost of suppressing procompetitive product improvement is high. Indeed, it will often be possible to design and license high-tech products such as software so that they work well with third-party products, which may in turn thrive by offering customers superior or differentiated capabilities on a standalone basis. In this way consumers will have the benefit of both an integrated approach to software design as well as standalone offerings. The remedies approved by the D.C. Circuit in United States v Microsoft Corp (Microsoft III) are premised on such an approach.

Part I briefly discusses the IBM plug-compatible peripheral decisions from the 1970s. These decisions established the general proposition that judicial review of high-tech product design decisions should be narrow and deferential. Part II describes the unique characteristics of “platform software” (such as the Windows operating system), a product category that became increasingly important in the 1990s. In general, technological tying claims do not fit easily within an off-the-shelf per se test developed to assess tying claims in other factual contexts; the characteristics of platform software make it particularly ill-suited for per se treatment.

Part III discusses the recent cases, both in the United States and in Europe, concerning Microsoft’s integration of various capabilities into new versions of Windows. In the United States, the D.C. Circuit found per se treatment inappropriate in view of the unique characteristics of platform software described in Part II. In a proceeding now under review by the European courts, the European Commission applied a test that is similar to the per se approach rejected by the D.C. Circuit. Part IV presents a proposed framework for applying the rule of reason test to software integration claims.

---

3 253 F3d 34 (DC Cir 2001).
I. IBM PLUG-COMPATIBLE PERIPHERAL CASES

In the 1970s, manufacturers of plug-compatible peripheral products brought a series of antitrust lawsuits against IBM under § 1 and § 2 of the Sherman Act, challenging various IBM product design decisions that combined functions previously provided by separate products. These manufacturers competed against IBM by selling plug-compatible products such as disk drives and memory for use with IBM computers. The IBM product design decisions at issue generally made its computers incompatible with existing plug-compatible products manufactured by competitors.

In analyzing tying claims brought under § 1 of the Sherman Act, these courts generally did not address whether such claims should be judged under the rule of reason or the per se rule. They instead disposed of the tying claims on one of two grounds: either (i) that the integration of previously separate functions involved a single product rather than two products as required for tying law purposes, or (ii) that the combination resulted in facially plausible benefits judged under a highly deferential standard. Courts addressing monopolization

modified per se approach in its case against Microsoft despite its claim that it used a rule of reason approach).  

§ 1 of the Sherman Act forbids “contract[s] ... in restraint of trade or commerce,” while § 2 forbids “monopoliz[ing], or attempt[ing] to monopolize, or combin[ing] or conspir[ing] ... to monopolize any part of the trade or commerce among the several States.” 15 USC §§ 1-2 (2000).

See, for example, In re IBM Peripheral EDP Devices Antitrust Litigation, 481 F Supp 965, 1003-04 (ND Cal 1979) (holding that IBM's design change was a permissible technological advancement, even though it precluded the use of competitors' replacement parts), aff'd as Transamerica Computer Co, Inc v IBM, 698 F2d 1377 (9th Cir 1983); ILC Peripherals Leasing Corp v IBM Corp, 448 F Supp 228, 230-31 (ND Cal 1978) (applying a “function of the aggregation” test), aff'd as Memorex Corp v IBM Corp, 636 F2d 1188 (9th Cir 1980) (per curiam; Telex Corp v IBM Corp, 367 F Supp 258, 341-42 (ND Okla 1973) (holding that IBM's tying of “peripheral products” was appropriate because it “represented technological advancements” and “a desire to make available in the market improved devices” as soon as possible), rev'd on other grounds, 510 F2d 894 (10th Cir 1975). Three courts have held that so-called “technological ties” do not fall within the per se rule, but rather should be judged under the rule of reason. See Foremost Pro Color, Inc v Eastman Kodak Co, 703 F2d 534, 542 (9th Cir 1983); Condesa Del Mar, Inc v White Way Sign & Maintenance Co, 1987 WL 17474, *2 (ND Ill); Innovation Data Processing, Inc v IBM Corp, 585 F Supp 1470, 1475 (D NJ 1984). In those cases, the defendant did not condition the purchase of one product on the purchase of another, but rather designed its products in such a way that as a practical matter, one could not be used without the other.

For example, the court in Response of Carolina, Inc v Leasco Response, Inc, 537 F2d 1307 (5th Cir 1976), did not address the single product question, but rather concluded that computer software and hardware that were compatible only with each other do not constitute an unlawful tie if the combination “achieve[d] some technologically beneficial result.” The court reasoned that “[a]ny other conclusion would enmesh the courts in a technical inquiry into the justifiability of product innovations.” Id at 1330.
claims under § 2 generally rejected such claims based on the technical superiority of the combined product.\footnote{See, for example, California Computer Products, Inc v IBM Corp, 613 F2d 727, 744 (9th Cir 1979) (holding that a monopolist has “the right to redesign its products to make them more attractive to buyers—whether by reasons of lower manufacturing cost or improved performance”). See generally Phillip E. Areeda and Herbert Hovenkamp, 3A Antitrust Law: An Analysis of Antitrust Principles and Their Application § 781a at 268 (Aspen 2d ed 1996) (“[P]roduct superiority is one of the objects of competition and cannot be wrongful, even for a monopolist.”).}

In all of these cases, the courts assessed efficiencies associated with integrated product designs under a highly deferential standard of review—even where the resulting design rendered competing products incompatible with IBM’s high share products. In Telex Corp v IBM Corp,\footnote{367 F Supp 258 (ND Okla 1973).} for example, the court observed that where “a court is dealing with what is physically and in fact a single product,” the antitrust laws do “not contemplate judicial dissection of that product into parts and reconstitution of these parts into a tying arrangement.”\footnote{Id at 347.} The court cautioned against an approach that “would enmesh the courts with technical and uncertain inquiry into the technological justifiability of functional integration and cast unfortunate doubt on the legality of product innovations in serious detriment to the industry and without a legitimate antitrust purpose.”\footnote{Id.} Similarly, in ILC Peripherals Leasing Corp v IBM Corp,\footnote{448 F Supp 228 (ND Cal 1978).} the court reasoned that “what Memorex is really attempting to do here is to hold the line against progress in disk storage technology,” something the antitrust laws do not countenance.\footnote{Id at 233.} The court observed that IBM’s “new disk storage technology will have a serious impact on the market for customer-removable disk products,” but that “the antitrust laws were not designed to insure the maintenance of the status quo for any competitor.”\footnote{Id.}

II. THE NATURE OF PLATFORM SOFTWARE

The IBM cases of the 1970s all concerned computer hardware. In those days, there was no separate software industry; software was offered to customers primarily as a component of mainframe computers. The development in the early 1980s by Microsoft and others of “platform software” that ran across computers manufactured by various hardware manufacturers changed all that,\footnote{For a discussion of the key economic characteristics of software platforms and case studies concerning personal computers, cell phones, video games, and the like, see David S. Evans, Andrei Hagiu, and Richard Schmalensee, A Survey of the Economic Role of Software Pla-} leading to the creation of a
distinct software industry that today consists of thousands of "independent software vendors." Software, in turn, has become subject to tying claims.

A number of aspects of software combine to present greater challenges for analyzing tying claims than in computer hardware and other contexts. To begin with, software programs providing discrete functions can easily be combined and offered to consumers as a single product. Although combining functionality is common in hardware as well—witness today's "four-in-one" printers that also copy, fax, and scan—combining functions is even easier in software because there are no physical constraints (although other constraints apply). Software functions may be packaged together in various ways for all the reasons identified in the economic and legal literature explaining why bundling is common throughout the economy, such as savings in production and distribution costs and pricing efficiencies. Given the zero or near-zero marginal cost of adding functions to a software product, it will often be economically efficient to do so, just as newspapers include news, sports, arts, and style sections. Consequently, software developers routinely seek to spur sales of new versions of their products by touting the availability of new features—features which they or others may have offered as separate products in the past.

Combining software (or hardware) functions can enable the development of entirely new capabilities, a powerful form of innovation. Software is highly abstract, however, and for those of us who are not computer programmers, understanding how software programs can fit together to create a whole greater than the sum of the parts is often quite challenging. We understand easily that an engine should be regarded as part of a single product—a car—because the car will not run without it. It is not so easy to understand the comparably important role that a particular set of software code may play as part of a larger software product.

Software products that serve as "platforms" for developing and running other software products exhibit additional considerations that ought to be examined in connection with a proper tying analysis. A


17 For a discussion of the value that a platform can provide to a network of firms, see Marco Iansiti and Roy Levien, The Keystone Advantage: What the New Dynamics of Business Ecosystems Mean for Strategy, Innovation, and Sustainability (Harvard Business 2004).


“platform” is software that provides functionality upon which other software developers can call in creating their own software programs. Platform functionality is made available to developers via software interfaces developed for that purpose, commonly referred to as application programming interfaces (APIs). Many microprocessor-based products, such as personal computers, game consoles, and, increasingly, cell phones, run on platform software.

Platform software is unique for tying law purposes in that the addition of new functionality to such software often provides substantial benefits to the developers of complementary products, namely, third-party software developers. Software developers call upon underlying platform software to obtain basic system services utilized by applications generally rather than having to recreate such functionality themselves in their own applications. For example, if a software developer wants to create an application with a graphical user interface—that is, an application with icons on the screen, toolbars, drop-down menus, and such—the developer can call upon the underlying operating system platform to provide that functionality. Thus, software developers can focus their efforts and resources on adding unique features and capabilities to their applications without the need to “reinvent the wheel.” This approach greatly reduces the cost and time required to write applications because there is less software to design, develop, and test.

The existence of a common software platform enables software developers easily to create applications that run on hardware offered by any manufacturer on which the software platform is installed and promotes interoperability among applications and between applications and various peripheral devices. Without such a development layer, software developers would need to create multiple versions of their applications to run on myriad hardware configurations (or to run on multiple versions of the platform, if the multiple versions of the platform were created with various functions missing). As it is, any application written to Windows, for example, can run on any of the thousands of different brands of personal computer that are sold with Windows preinstalled. Third-party developers and their customers benefit from the compatibility provided by such a platform.

Software and hardware developers continually demand the addition of new functions to platform software to facilitate the development of additional capabilities in their own products. The demand for integration of new functionality is driven in large part by technological advances in other parts of the computer ecosystem, such as the development of vastly more powerful microprocessors, falling prices for computer memory and storage, and the advent of the internet. As microprocessors become more powerful, for example, operating sys-
tems can offer new capabilities, such as richer graphics or the ability to display full-motion video, and those capabilities can in turn be made available as platform services to developers of complementary products.

Developer demand for new platform functions points up a final—and very important—aspect of software: in most cases, particular functionality can be developed and offered to the marketplace either as platform software or as applications software, and consumers can have the benefit of both approaches. In the typical pattern, new functionality is first developed as an application and offered on a standalone basis separately from operating system software. Over time, as more and more software developers demand that capability for their applications, operating system vendors undertake the engineering work to redevelop the functionality as platform software. That is the pattern followed by web browsing software. In late 1994, Netscape released Navigator, which was an instant hit. Netscape thereafter continued to offer its web browsing software as an application, devoting engineering resources to developing versions of Navigator to run on a variety of computer operating systems. Microsoft's initial release of Internet Explorer in 1995 was, for the most part, similarly designed as an application. Over the next two years, however, Microsoft devoted enormous engineering resources to breaking down the functions performed by Internet Explorer into a series of software components, and then exposing the functions performed by those components to the software developer community through Microsoft's development of approximately five hundred new APIs. These Internet Explorer components, in turn, became an integral part of successive releases of Windows. How the courts assessed the integration of web browsing software into Windows is discussed in the next Part.

III. THE MICROSOFT CASES

In the late 1990s, the Department of Justice (DOJ) commenced two separate proceedings against Microsoft that challenged Microsoft's integration of web browsing functionality into its Windows operating system. These lawsuits were the first to address tying claims in the context of platform software. The resulting decisions articulated various divergent tests for analyzing claims that the addition of a new feature to platform software constituted an unlawful tie. These tests ranged from a highly deferential standard similar to that applied in the IBM plug-compatible peripheral cases, to the traditional per se

---

20 *United States v Microsoft Corp*, 147 F3d 935, 950 (DC Cir 1998) (*Microsoft II*).
standard applied by courts in tying cases,\textsuperscript{21} to a rule of reason standard that is sufficiently open ended to take into account the unique characteristics of platform software.\textsuperscript{22} In March 2004, the European Commission articulated a test for assessing tying claims that resembles the U.S. per se test and applied this with some novel twists to the case of platform integration.\textsuperscript{23}

\textbf{A. Microsoft II}

In 1997, the DOJ contended that Microsoft's inclusion of Internet Explorer in Windows 95 violated an anti-tying clause of an earlier consent decree between Microsoft and the United States. The clause included a proviso stating that "this provision in and of itself shall not be construed to prohibit Microsoft from developing integrated products."\textsuperscript{24} The issue before the D.C. Circuit in \textit{United States v Microsoft Corp} (Microsoft I) was whether Windows 95 constituted a single "integrated product" within the meaning of the proviso.

The D.C. Circuit adopted a construction of the relevant clause that it believed to be "consistent with the antitrust laws."\textsuperscript{25} Echoing the IBM plug-compatible peripheral cases discussed in Part I, the D.C. Circuit observed that "[a]ntitrust scholars have long recognized the undesirability of having courts oversee product design" and that "any dampening of technological innovation would be at cross-purposes with antitrust law."\textsuperscript{26}

The D.C. Circuit held that "an 'integrated product' is most reasonably understood as a product that combines functionalities (which may also be marketed separately and operated together) in a way that offers advantages unavailable if the functionalities are brought separately and combined by the purchaser."\textsuperscript{27} The court explained that "[t]he question is not whether the integration is a net plus but merely whether there is a plausible claim that it brings some advantage."\textsuperscript{28}

\begin{footnotesize}
\textsuperscript{21} \textit{United States v Microsoft Corp}, 87 F Supp 2d 30, 47–51 (D DC 2000).
\textsuperscript{22} Microsoft III, 253 F3d at 84.
\textsuperscript{24} Final Judgment, \textit{United States v Microsoft Corp}, Civil Action No 94-1564 § IV(E), online at http://www.usdoj.gov/atr/cases/f0000/0047.htm (visited Dec 10, 2004).
\textsuperscript{25} 147 F3d 935 (DC Cir 1998).
\textsuperscript{26} Id at 948.
\textsuperscript{27} Id.
\textsuperscript{28} Id.
\textsuperscript{29} Id at 950.
\end{footnotesize}
The court concluded that “this understanding is consistent with tying law.” In so doing, it rejected the DOJ’s reliance on Jefferson Parish Hospital District No 2 v Hyde “for the proposition that products are distinct for tying purposes”—and thus not “integrated” within the meaning of the proviso—“if consumer demand exists for each separately.” The court stated that Jefferson Parish’s focus on consumer demand is not a reliable test of separateness when a new product integrates functionalities previously provided by separate products.

The D.C. Circuit ruled that Microsoft had “clearly met the burden of ascribing facially plausible benefits to its integrated design of [Windows 95] as compared to an operating system combined with a stand-alone browser.” Foreshadowing its reasoning in later decisions, the court emphasized the unique efficiencies associated with improvements to platform software. The court noted that “[i]ncorporating browsing functionality into the operating system allows applications to avail themselves of that functionality without starting up a separate browser application.” It also stated that the Internet Explorer components of Windows “provide system services [to software developers] not directly related to Web browsing, enhancing the functionality of a wide variety of applications.” Finally, the court pointed out that Internet Explorer technologies “are used to upgrade aspects of the operating system unrelated to Web browsing,” such as the operating system’s user interface.

B. Microsoft III

In 1998, the DOJ (together with a number of state attorneys general) commenced a separate antitrust action against Microsoft, alleging violations of § 1 and § 2 of the Sherman Act. The DOJ contended, among other things, that Microsoft’s inclusion of Internet Explorer in Windows 98 constituted a per se unlawful tie under § 1.

---
30 Id.
31 466 US 2, 19 & n 30 (1984) (inquiring into the “character of the demand” for two products and not their “functional relation” when conducting a tying analysis).
32 Microsoft II, 147 F3d at 946.
33 Id at 947 & n 9.
34 Id at 950.
35 See Microsoft III, 253 F3d at 84–95; Massachusetts v Microsoft Corp, 373 F3d 1199, 1210–11 (DC Cir 2004) (en banc) (Microsoft IV) (discussing the potential harm to Microsoft and its customers if the company were forced to refrain from integration).
36 Microsoft II, 147 F3d at 950–51.
37 Id at 951.
38 Id.
39 Microsoft, 87 F Supp 2d at 47 (Jackson). The case was assigned to a different district court judge on remand due to the apparent partiality of the original trial judge. Microsoft III, 253 F3d at 107–17.
The district court held that "Microsoft is liable for illegal tying under § 1," even though it recognized that "this conclusion is arguably at variance" with the D.C. Circuit's decision in *Microsoft II.* Determining that the D.C. Circuit's "undemanding test" is "inconsistent with the pertinent Supreme Court precedents," the district court applied the per se test set out by the Supreme Court in *Jefferson Parish.*

Applying *Jefferson Parish*'s consumer demand test, the district court found that "the commercial reality is that consumers today perceive operating systems and browsers as separate 'products,' for which there is separate demand." Based on this finding, the district court concluded that "Windows and Internet Explorer [are] deemed 'separate products,'" without regard to the extent to which web browsing functionality is integrated into the operating system and results in technical benefits.

On appeal, the D.C. Circuit vacated the district court's finding of a per se tying violation, holding that "the rule of reason, rather than per se analysis, should govern the legality of tying arrangements involving platform software products." In so holding, the court stated that this case "offer[ed] the first up-close look at the technological integration of added functionality into software that serves as a platform for third-party applications." Finding "no close parallel in prior antitrust cases," the D.C. Circuit concluded that "simplistic application of per se tying rules carries a serious risk of harm."

The D.C. Circuit's analysis focused largely on the efficiencies that may be associated with integration of new functionality into platform software. At the outset, the court emphasized "the poor fit between the separate-products test and the facts of this case." Indeed, the court accepted Microsoft's argument that "*Jefferson Parish*'s consumer demand test would 'chill innovation to the detriment of consumers by preventing firms from integrating into their products new functional-

---

40 *Microsoft,* 87 F Supp 2d at 47.
41 Id.
42 Id at 49.
43 Id at 50.
44 *Microsoft III,* 253 F3d at 84. The D.C. Circuit noted that the allegations underlying the DOJ's tying claim under § 1 overlapped with those underlying its monopoly maintenance claim under § 2, which the court affirmed in part and reversed in part. Id.
45 On remand, the plaintiffs declined to pursue the tying claim, in part due to the "significant legal hurdles" presented by the "rigorous proof requirements" under the rule of reason test articulated by the D.C. Circuit. See Memorandum of the United States in Support of the Entry of the Proposed Final Judgment, *United States v Microsoft Corp,* No 98-1232, *65 (D DC filed Feb 27, 2002), online at http://www.usdoj.gov/atr/cases/f10100/10143.wpd (visited Dec 10, 2004).
46 *Microsoft III,* 253 F3d at 84.
47 Id at 85.
ity previously provided by standalone products—and hence, by definition, subject to separate consumer demand.""48

More specifically, the D.C. Circuit concluded that Jefferson Parish's per se analysis is inappropriate for alleged tying arrangements involving "software that serves as a platform for third-party applications."48 The court explained that in none of the prior Supreme Court tying cases "was the tied good physically and technologically integrated with the tying good. Nor did the defendants ever argue that their tie improved the value of the tying product to users and to makers of complementary goods."49

The D.C. Circuit stressed that the "nature of the platform software market affirmatively suggests that per se rules might stunt valuable innovation."50 First, the court emphasized that the per se rule's separate-products test is unhelpful in cases involving newly integrated products "because at the moment of integration there will appear to be a robust 'distinct' market for the tied product."51 In contrast, rule of reason analysis "affords the first mover an opportunity to demonstrate that an efficiency gain from its 'tie' adequately offsets any distortion of consumer choice."52 The court expressed concern that the separate-products test failed to screen out productive integration, which is common in platform software markets, including "among firms without market power."53 Thus, the "ubiquity of bundling in competitive platform software markets should give courts reason to pause before condemning such behavior in less competitive markets."54

Second, the court stated that "because of the pervasively innovative character of platform software markets, tying in such markets may produce efficiencies that courts previously have not encountered and thus the Supreme Court had not factored into the per se rule as originally conceived."55 For example, the court noted, "the bundling of a browser with [operating systems] enables independent software developers to count on the presence of the browser's APIs, if any, on consumers' machines and thus to omit them from its own package."56 The court emphasized that such efficiencies "are common in technolo-

48 Id at 89.
49 Id at 84.
50 Id at 90.
51 Id at 92.
52 Id.
53 Id.
54 Id at 93.
55 Id.
56 Id.
57 Id.
logically dynamic markets where product development is especially unlikely to follow an easily foreseen linear pattern." The court concluded that integration of "new functionality into platform software is a common practice and that wooden application of per se rules in this litigation may cast a cloud over platform innovation in the market for PCs, network computers and information appliances."

C. Microsoft IV

While Microsoft entered into a consent decree with the DOJ and nine of the state attorneys general, the remaining states (and the District of Columbia) declined to settle and proposed a different set of remedies, leading to Massachusetts v Microsoft Corp (Microsoft IV). These remedies included a requirement that Microsoft develop multiple versions of Windows from which the software code that provides functionalities such as Internet Explorer and Windows Media Player could be removed by computer manufacturers (also known as "original equipment manufacturers" or OEMs) and end users. Following a thirty-two-day remedies trial that focused largely upon the effect of removing software code from Windows, the district court entered final judgments that were nearly identical to the negotiated consent decree. These final judgments do not require Microsoft to remove functionality from Windows—thereby preserving the benefits associated with Microsoft’s integrated design of the operating system—but rather impose a series of design and licensing obligations upon Microsoft that are intended to ensure that competitors’ software will run well on Windows and that competitors can distribute their software products to consumers through a variety of channels, including through installation on new PCs built on Windows.

In June 2004, the D.C. Circuit, sitting en banc, unanimously affirmed the remedies entered by the district court. The court observed—as it had in Microsoft III—that there was no apparent means "of distinguishing 'operating system' code from 'non-operating system' code, such as code that provides middleware functionality." Even more importantly, the court observed that "both [software developers] and consumers would be harmed if Microsoft were forced to

58 Id at 94.
59 Id at 95.
60 373 F3d 1199 (DC Cir 2004) (en banc).
62 Microsoft IV, 373 F3d at 1250.
63 Id at 1208. 
"[M]iddleware' refers to software products that expose their 'Applications Programming Interfaces,' upon which software developers rely in writing applications.” Id at 1208 n 3, citing Microsoft III, 253 F3d at 53.
redesign Windows by removing software code.

The court applauded the district court's entry of final judgments that remedy any anticompetitive effects while avoiding the drawbacks of "intruding itself into the design and engineering of the Windows operating system. We say, Well done!"

D. The European Commission Decision

In March 2004, the European Commission ruled that Microsoft violated Article 82(d) of the Treaty of the European Communities by including Windows Media Player (WMP) in its Windows operating system. The Commission articulated a new four-part test for assessing tying in this case: (i) the tying and tied goods must be separate products; (ii) the defendant must be "dominant" in the tying product market; (iii) the defendant must not give customers a choice of obtaining the tying product without the tied product; and (iv) the tying arrangement must foreclose competition. A "foreclosure" requirement is normally not part of tying analysis under European competition law. As the Commission recognized, however, the case before it was not a "classical tying case[]" because consumers are free to, and routinely do, use third-party media players (which are available free of charge) with Windows. Although not articulated by the Commission as a part of its tying test, the Commission also discussed—but only to reject—the possibility that otherwise unlawful tying could be saved by an "objective justification" for the alleged tie. As applied, the analysis employed by the European Commission in finding an unlawful tie was more reminiscent of the per se approach of Jefferson Parish than of the rule of reason framework articulated in Microsoft III. In particu-

\[\text{64} \quad \text{Microsoft IV, 373 F3d at 1208.}\]

\[\text{65} \quad \text{Id at 1210.}\]

\[\text{66} \quad \text{European Commission Decision ¶ 792. Article 82(d) of the EC Treaty forbids "making the conclusion of contracts subject to acceptance by other parties of supplementary obligations which, by their nature or according to commercial usage, have no connection with the subject of such contracts." Treaty of Amsterdam Amending the Treaty on European Union, the Treaties Establishing the European Communities and Certain Related Acts, Art 82(d), 40 Off J Eur Communities (C 340) 147 (Oct 2, 1997). The Commission's decision "demanded that Microsoft share secret information about Windows to allow rivals to design products that can interoperate with its ubiquitous operating system; it also forced the company to sell a stripped-down version of its Windows operating system in Europe without its own media player." Kevin J. O'Brien, Microsoft Asks Appeals Court in Europe to Delay Punishment, NY Times W1 (Oct 1, 2004). Microsoft's appeal before the European Court of First Instance is currently pending; its request to delay implementation of the remedy was denied. See Steve Lohr and Paul Meller, Europe Rejects Microsoft's Bid to Preserve Bundling Plan, NY Times C1 (Dec 23, 2004).}\]

\[\text{67} \quad \text{European Commission Decision ¶ 794.}\]

\[\text{68} \quad \text{Id ¶ 841.}\]

\[\text{69} \quad \text{Id ¶ 961.}\]
lar, the Commission’s heavy focus on whether the technologies constitute “separate products” based on consumer demand was very similar to the district court’s tying ruling vacated by the D.C. Circuit in Microsoft III.70

The Commission devoted most of its attention to the first and fourth elements of the test it articulated (finding the others easily satisfied): whether Windows and WMP are separate products and whether the alleged tying arrangement forecloses competition. Like the district court in the liability phase of the U.S. proceedings, the Commission rejected this focus on the technical integration of the software functions at issue—and the efficiencies that result from improvements to platform software—holding that “[t]he distinctiveness of products for the purposes of an analysis under Article 82 . . . has to be assessed with a view to consumer demand.”71 The Commission concluded that there is “separate consumer demand for media players, distinguishable from the demand for client PC operating systems,” as evidenced by the fact that media players are provided separately in the marketplace by various firms.” The Commission stressed the fact that “non-insignificant consumer demand” for separate media players existed “some four years after Microsoft started tying its streaming media player with Windows.”72

On the fourth element of a tying claim, the Commission concluded that the “tying in this specific case has the potential to foreclose competition.”73 In other words, the Commission did not hold that competition had actually been foreclosed. Rather, it stated that “tying WMP with the dominant Windows makes WMP the platform of choice for complementary content and applications which in turn risks foreclosing competition in the market for media players.”74 The Commission contended that the inclusion of WMP in the ubiquitous Windows operating system leads content providers and software developers to rely upon Windows media formats and APIs, which has the effect of increasing usage of WMP.75 Over time, the Commission found, use of media player software would “tip” to WMP.76

70 Id ¶ 806; Microsoft III, 253 F3d at 84.
71 European Commission Decision ¶ 803. Compare Microsoft, 87 F Supp 2d at 47–49 (using a “character of the demand” test for determining whether two products are in reality a single product—as opposed to a “functional relationship” test).
72 European Commission Decision ¶¶ 804, 806, 808.
73 Id ¶ 808.
74 Id ¶ 842 (emphasis added).
75 Id (emphasis added).
76 Id ¶¶ 879–96.
77 Id ¶¶ 968, 1016, 1071. Before the Commission, and in its appeal from the Commission’s decision, Microsoft has argued that the economic conditions that might lead to “tipping” are not
The Commission rejected the justifications put forward by Microsoft for the integration. Although it acknowledged that an objective justification could serve to overcome tying liability,\(^7\) the Commission ruled that such justifications would not suffice unless the challenged "tying" was "indispensable" (that is, the only possible means) to attain the cited efficiencies—and, in this regard, the Commission said that efficiencies that relate from Microsoft's deliberate design choices do not count.\(^7\)

In June 2004, Microsoft filed an application to annul the decision with the European Court of First Instance. Microsoft's appeal is pending.

IV. A SUGGESTED FRAMEWORK FOR APPLYING THE RULE OF REASON ANALYSIS

For the reasons outlined by the D.C. Circuit in Microsoft III, tying claims arising from the integration of new functionality into software should be assessed under a rule of reason test.\(^8\) As the D.C. Circuit observed, none of the Supreme Court's prior cases involved a tied good that was "physically and technologically integrated with the tying good."\(^9\) The fact that a product has been designed as a single product—with, as the court put it in Microsoft II, an "interpenetrating design"—strongly suggests that the claimed "tie" entails substantial efficiencies that should be considered.\(^9\) Indeed, the per se test's focus on whether separate consumer demand exists for the tied product asks the wrong question. In the context of technological tying claims, the proper question to ask is whether there is substantial consumer demand for the "tying" product absent the "tied" product. Vendors develop and supply tires and spark plugs on a standalone basis separate from cars, but that does not mean that a fully functioning automobile constitutes multiple products.

Since 2001, no United States court has had occasion to apply the rule of reason test in the context of a technological tying claim involve-

---

\(^7\) European Commission Decision ¶ 961.

\(^8\) Although the D.C. Circuit made clear that its endorsement of a rule of reason approach was limited to platform software, much of the court's reasoning is applicable to technological tying claims generally. See note 6.

\(^9\) Microsoft III, 253 F3d at 90.

\(^{10}\) Microsoft II, 147 F3d at 951–52.
ing platform (or any other kind of) software. The D.C. Circuit's decision in Microsoft III, however, together with its conclusions in Microsoft IV, suggests a number of factors that should be considered when conducting a rule of reason analysis in the context of any technological tying claim. Most importantly, the outcome of the Microsoft cases suggests that the rule of reason test should be applied in a way that encourages firms to design and market integrated products so as to provide the benefits of integration while preserving strong competitive opportunities for other firms to provide functionality that competes with the integrated functionality on a standalone basis. In this way, consumers will have the best of both worlds: the benefits of integrated product design as well as the opportunity to choose standalone products, and the salutary benefits that result from competition between the two. Such a result can be achieved under a rule of reason rubric by focusing upon the manner in which the defendant integrated the challenged functionality—which will often directly affect the possibility of foreclosure—and not the mere fact that integration occurred, as well as the benefits that will often arise from an integrated design.

A. Foreclosure

As in any rule of reason case, courts must weigh whether any anticompetitive effects of the challenged product integration outweigh any procompetitive effects. In a tying case, the focus of judicial inquiry is whether the challenged practice "foreclose[s] competition on the merits in a product market distinct from the market for the tying item." Ties are unlawful only if they "deny competitors free access to the market for the tied product" and force buyers "to forego their free choice between competing products." Thus, courts should begin by assessing whether the challenged integration forecloses competitors' opportunities to develop or distribute products. As in any rule of reason analysis, courts should require the plaintiff to establish actual harm to competition in the tied product market, that is, proof that foreclosure has occurred, not merely the "risk" of foreclosure in the future.

---

83 Jefferson Parish, 466 US at 21 n 34.
84 Northern Pacific Railway Co v United States, 356 US 1, 6 (1958).
85 The D.C. Circuit explained that "[m]eeting that burden 'involves an inquiry into the actual effect' of [the challenged] conduct on competition in the tied good market." Microsoft III, 253 F3d at 95, quoting Jefferson Parish, 466 US at 29. The requirement of an "actual effect" on competition comes from Jefferson Parish, where the Supreme Court held that, in the absence of per se liability, "a showing of actual adverse effect on competition" is necessary to demonstrate an illegal tying arrangement. 466 US at 31. Following Jefferson Parish, courts have held that a
The extent to which an integrated product design may foreclose competitors' opportunities will vary widely depending upon the circumstances. In some cases, integration may essentially eliminate the possibility of offering a comparable product on a standalone basis. That may occur, for example, if an integrated product is designed in such a way as to be incompatible with competitors' offerings, or where it is simply impractical to offer a standalone product for use with an integrated product. In the 1960s and 1970s, various firms offered air conditioning systems as add-ons to automobiles. Later, automobile manufacturers designed automobiles with increasingly sophisticated air conditioning systems built in from the outset, which effectively eliminated any possibility of offering add-on air conditioning systems. As the automobile air conditioning example suggests, the fact that an integration effectively excludes competitors should not, in and of itself, serve to establish tying liability: the exclusion of competitors may simply reflect economic efficiencies arising from an integrated approach and consumer preference.

In the context of software (and some other high-tech products), it will often be possible to design and distribute integrated products in a manner that enables competitors to offer competing products that work well with the defendant's integrated product, avoiding exclusion. There are a range of possibilities, relating to matters such as product design, disclosure of interfaces, and intellectual property licensing terms, that may entail relatively greater or lesser opportunities for competitors that build complementary products. General purpose platform software such as modern computer operating systems are designed to enable third-party software developers to call upon their functionality, via APIs, to create any kind of software—including software that competes with functionality provided by the platform itself. In the case of media playback, for example, Windows includes WMP, while literally dozens of firms (including Apple) offer standalone media players designed to run on Windows. These media players call on the general purpose APIs exposed by Windows, including, in many
cases, APIs provided by the WMP functionality in Windows. In this case, the inclusion of integrated functionality tends to facilitate the development of new products, including products that compete with the integrated functionality.

In assessing the extent to which an integrated design could foreclose competitors’ opportunities, courts should also consider how easy or difficult it is for consumers to use standalone offerings with the integrated product. Microsoft designs Windows, for example, so that applications software, as well as features of Windows, can be accessed easily via icons on the Windows Start menu, desktop, or “Quick-Launch” bar along the bottom of the screen. Microsoft licenses Windows to OEMs on terms that enable manufacturers to install any software on PCs and put icons for such software in prominent locations within the Windows user interface.

In some cases, it may be possible to go further: designing an integrated product (and providing sufficient interfaces) such that a standalone offering not only works well with the product, but also can effectively replace the built-in functionality from the perspective of consumers. For example, although Windows is perfectly capable of opening and displaying a website when a user encounters a link to the Internet, the operating system can be configured to launch a third-party web browser “by default” to perform that function. From the consumers’ perspective, the third-party web browser has essentially replaced the web browsing functionality that is built into Windows.

In support of their tying and monopoly maintenance claims in the liability phase of the Microsoft case, the DOJ and state attorneys general went further still, arguing that the Internet Explorer software in Windows could be effectively “removed” from the operating system without negative consequences for third-party software developers by removing the means by which end users accessed Internet Explorer (such as an icon on the desktop), but leaving its underlying software (which provides useful APIs) intact. The district court agreed.89 The final judgment approved in Microsoft IV builds upon this approach to enhance opportunities for competitors.88 Under the final judgment, OEMs and consumers are provided with the right to configure Win-

88 United States v Microsoft Corp, 84 F Supp 2d 9, 165, 177 (D DC 1999); Brief for Appellees United States and the State Plaintiffs, United States v Microsoft Corp, Nos 00-5212 and 00-5213, *107 (DC Cir filed Jan 12, 2001), online at http://www.usdoj.gov/atr/cases/f7400/7425.wpd (visited Dec 10, 2004).
89 This remedy was addressed to the trial court's finding that OEMs would be less likely to install non-Microsoft web browsers if the icons for Internet Explorer appeared on the Windows desktop out of concern that computer users would be confused as to which icons to select to access the internet, which in turn would increase product support costs. Microsoft, 84 F Supp 2d at 49–50, 52–53.
dows so that the icons for key features that are built into the operating system, such as Internet Explorer and WMP, do not appear. OEMs can prominently feature third-party software that competes with features built into Windows, providing de facto exclusive promotion for such software on new PCs. In other high-tech product categories as well, it may be possible to design features that can essentially be "turned off" and removed from view in favor of a competing product that provides comparable functionality. Granting intermediaries in the distribution channel (such as OEMs) the right to "turn off" and remove features from view may facilitate both distribution and use of competing offerings.

Courts should be wary of any attempt to recast the efficiency benefits that independent software developers derive from broad (or "ubiquitous") distribution of platform software as a form of "foreclosure." The breadth of distribution of one firm's product says nothing about the ability of other firms to distribute their products, much less establish that they have been (or may be) foreclosed from such distribution. For example, literally hundreds of millions of copies of third-party media players have been distributed by firms such as RealNetworks, Apple, AOL, and others, even as Windows has consistently included media player functionality for more than ten years. Broad distribution of platform software is one of several attributes that developers value in considering whether to create products for a particular platform, but that is hardly a reason to cite "ubiquity" of distribution as unlawful foreclosure. "Tipping" to a ubiquitous platform technology is particularly unlikely where (as is often the case in software) vendors offer differentiated products.

Finally, to avoid penalizing product improvement, courts should scrutinize closely claims that foreclosure can result from improvements to integrated features, triggering tying liability. While it is a viable and oft-pursued strategy for firms to compete with integrated functionality by offering better or different features, the defendant's efforts to match or exceed such features should be encouraged.90

90 The European Commission charged Microsoft with unlawful tying from 1999 because that was the year that the media functionality built into Windows was improved to enable playing audio "streamed" over the internet (that is, without waiting for the audio file to be downloaded to the user's computer). The Commission met Microsoft's argument that it had included media functionality in Windows since 1992 by observing that "[w]hile Microsoft's bundled client-side media software was not capable of quality media streaming there were rival products which had a competitive edge over Microsoft's software." Sun Microsystems v Microsoft Corp, ECComm 1 (Comp/C-3/37.792) (Mar 24, 2004) ¶ 819 ("European Commission Decision"), online at http://europa.eu.int/commlcompetition/antitrust/cases/decisions/37792/en.pdf (visited Dec 10, 2004). In other words, improvements to the longstanding media functionality in Windows triggered tying liability.
B. Efficiencies

Under a rule of reason approach, the extent of any foreclosure will be balanced against any efficiencies arising from a challenged integrated design. As suggested by the IBM plug-compatible cases of the 1970s and Microsoft II (cited with approval in Microsoft IV), such efficiencies should be reviewed under a deferential standard.\(^91\) That is true because (i) integration of new functionality is common throughout the economy, which strongly suggests that the practice is usually efficient; (ii) courts are generally ill-equipped to second-guess engineering judgments; and, as the court emphasized in Microsoft III, (iii) welfare-enhancing innovation may be deterred if efficient integration is mistakenly condemned as unlawful tying.\(^92\) As the district court observed in Microsoft IV, the case law makes clear that "it is not a proper task for the Court to undertake to redesign products."\(^93\) In assessing efficiency, the courts should consider the following factors.

First, proof that competing firms also include the challenged functionality with their products should be seen as powerful evidence of efficiency. The D.C. Circuit observed in Microsoft III that "nearly all competitive OS vendors also bundle browsers."\(^94\) The court explained that "[f]irms without market power have no incentive to package different pieces of software together unless there are efficiency gains from doing so."\(^95\) At a minimum, proof that competing firms include challenged functionality in their products may reflect the commonsense notion that consumers generally prefer more capable products.

That there are independent suppliers of the "tied" product (web browsing software and media players)—presumptively reflecting consumer demand for such products on a standalone basis—should not matter. It is the actions of competing vendors of the integrated offering (here, operating systems) that are relevant to assessing whether including the challenged functionality leads to efficiency gains. Indeed, products such as shoes, computers, and automobiles that are efficiently...
sold as a single product often consist of components that are also available separately (such as shoelaces, video cards, and tires).”

As the D.C. Circuit observed in rejecting application of the Jefferson Parish test in the context of platform software, “[T]he separate-products test is a poor proxy for net efficiency from newly integrated products.” Even worse, reliance on consumer demand—which “focuses on historic consumer behavior, likely before integration”—threatens to chill innovation to the detriment of consumers by discouraging firms from integrating into platform software new functionality previously provided by standalone products.”

A wide variety of functions included in modern operating system products are also available separately, including alternative file systems, memory managers, and graphics engines. Rather than suggesting that a challenged integration is unlawful, the availability of standalone products that work well with the defendant’s product should properly be seen as evidence that the defendant’s design does not foreclose competitive opportunities.” Furthermore, an approach to assessing efficiencies that does not focus upon whether the “tying” product is typically offered with the “tied” product (as opposed to whether the “tied” product is also offered separately) threatens a perverse outcome whereby popular products such as Windows cannot be improved with new capabilities that are readily available in less widely used products, such as the Apple OS X operating system.

Second, courts should consider the extent to which the challenged functionality is integrated with the “tying” product physically or technically. A “bundle” of conceptually distinct products, without any physical or technical integration between the products, should not necessarily be condemned absent such integration. Bundling is often justified on the basis of savings in production, distribution and con-

---

96 See Jack Walters & Sons Corp v Morton Building, Inc, 737 F2d 698, 704 (7th Cir 1984) (“There are separate markets for sugar and sugarless breakfast cereals, but it would be surprising to find that a sugary cereal was a tie-in (sugar tied to cereal), assuming the seller refused to sell a sugar-free version.”). See also Phillip E. Areeda and Herbert Hovenkamp, 10 Antitrust Law: An Analysis of Antitrust Principles and Their Application § 1745d2 at 211 (Aspen 2d ed 1996) (“[A]n independent market for carburetors does not make a car with [a] carburetor installed two products because no significant market exists for cars stripped of their carburetors.”).
97 Microsoft III, 253 F3d at 92.
98 Id at 89.
99 The European Commission took a different tack, suggesting that no unlawful tie would be found if the effect of the tie were to eliminate demand for competing products altogether. European Commission Decision ¶ 808 & n 938 (contrasting the continued demand for non-Microsoft media players “some four years after Microsoft started tying its streaming media player with Windows” with the lack of appreciable demand for alternative graphical user interfaces “as an add-on to DOS”).
sumer transaction costs, and other efficiencies.\textsuperscript{100} Proof that allegedly "tied" and "tying" products are in fact integrated, however, is especially powerful evidence of efficiency.

The D.C. Circuit explained that "the very same lines of code" that provide key functionality for web browsing, such as the ability to display content in HTML, the format of the web, also provide key functionality within the operating system, such as the ability to display Help files.\textsuperscript{101} Using a single block of code to provide multiple functions is obviously efficient, saving not only on development costs, but testing as well (a process that can account for half or more of the cost of bringing a complex software product to market).

Even more fundamentally, such an "interpenetrating design"\textsuperscript{102} often enables entirely new capabilities to be provided to consumers. In the 1960s, consumer electronics companies combined a clock with a radio to build a new device—the clock radio—that enabled a new function: the ability to wake up to music. When Microsoft integrated HTML display software into Windows, the company was able to rewrite its extensive Help files for the operating system in the web format, thereby enabling consumers to use the familiar web browser paradigm (clicking on links, the "back" and "forward" buttons, and so forth) to navigate Help content, and to get updates to that content from the internet. Integration of web browsing software into Windows also enabled Microsoft to develop the "Windows Update" feature, which enables tens of millions of computer users worldwide to get the latest patches and updates to Windows automatically downloaded to their computers. As these examples suggest, where a challenged product design entails "productive integration,"\textsuperscript{103} it will often be somewhat artificial to speak of separate "tying" and "tied" products.

Third, the courts should consider the extent to which inclusion of challenged functionality in the defendant's product may benefit third parties. In rejecting per se analysis, the D.C. Circuit observed that in none of the Supreme Court tying cases "did the defendants ever argue that their tie improved the value of the tying product to users and to makers of complementary goods."\textsuperscript{104} In fact, the very purpose of platform software is to provide useful functionality to third parties. Plat-

\textsuperscript{100} See Microsoft III, 253 F3d at 87, citing Phillip E. Areeda and Herbert Hovenkamp, 9 Antitrust Law: An Analysis of Antitrust Principles and Their Application § 1703g2 at 51–52 (Aspen 2d ed 1996). Bundling may also be a first step in the software engineering process toward development of a technically integrated product. Integration of complex functionality is often a major undertaking that takes place over the course of several product releases.

\textsuperscript{101} Microsoft III, 253 F3d at 87. See also New York, 224 F Supp 2d at 246–47.

\textsuperscript{102} Microsoft II, 147 F3d at 952.

\textsuperscript{103} Microsoft III, 253 F3d at 92. See also New York, 224 F Supp 2d at 156, 157 n 64.

\textsuperscript{104} Microsoft III, 253 F3d at 90.
form developers continually update their platform products to provide new opportunities to developers of applications that run on the platform. At trial, Microsoft placed great weight on the fact that the very same lines of code that performed multiple functions throughout the operating system, such as the HTML display software, were also made available for use by third-party software developers via APIs. In this way, developers could build applications that rely upon HTML without having to recreate the HTML display software themselves. For example, the Windows platform provides APIs that simplify the process of creating Help files in software applications. Those Help APIs, in turn, build upon the HTML display software in Windows (which developers can also access directly). Microsoft also emphasized the importance of maintaining the same set of APIs in every version of Windows so that developers would know that if they wrote applications to rely on Windows APIs, their applications would run properly on any PC that ran Windows.

On remand following Microsoft III, the effect on developers was fully considered by the district court because the nonsettling states proposed remedies that would have required the development of multiple versions of Windows, with and without various features such as Internet Explorer and WMP (and thus their associated APIs). In rejecting that proposed remedy, the district court noted that the proposal “would hinder, or even destroy, Microsoft’s ability to provide a consistent API set,” thereby “fragment[ing]” the Windows platform.\textsuperscript{105} Such a result, the court found, “would be significantly harmful” to software developers, who “would not have any assurance that a particular functionality was present in any given configuration of . . . Windows.”\textsuperscript{106} In Microsoft IV, the D.C. Circuit quoted with approval the district court’s conclusion that “[t]he weight of the evidence indicates that fragmentation of the Windows platform would be significantly harmful to Microsoft, [independent software vendors], and consumers.”\textsuperscript{107}

Fourth, and relatedly, courts should carefully scrutinize claims that integrated functionality and standalone products are essentially fungible so that the integrated functionality could be removed from the platform and replaced by a standalone product.\textsuperscript{108} From the perspective of the functionality presented to end users, functionality built into platform software and that offered solely on a standalone basis

\textsuperscript{105} New York, 224 F Supp 2d at 252.
\textsuperscript{106} Id at 253.
\textsuperscript{107} Microsoft IV, 373 F3d at 1211, quoting New York, 224 F Supp 2d at 253.
\textsuperscript{108} European Commission Decision ¶ 967 (“[I]t is no more efficient for Microsoft to create client PC operating system-and-application bundles than for multiple OEMs to create those client PC operating system-and-application bundles that are desired by end users.”).
may be broadly comparable (although likely differentiated in various ways). From the platform perspective, however, that will seldom be the case. Software that has been designed as an application, to be offered apart from operating system software, typically will provide little or no platform capability. ¹⁰⁹

Finally, in the context of technological tying claims involving platform software, courts should be sensitive to the possibility that the only firms that may have a grievance with the inclusion of challenged functionality are those that wish to provide comparable functionality in other ways, while the industry as a whole and consumers benefit from improvements to the platform. ¹¹⁰

CONCLUSION

The D.C. Circuit’s rule of reason test from Microsoft III is the appropriate framework for analyzing antitrust challenges to the technological integration of new functionality into software. In applying the D.C. Circuit’s test, courts should focus on whether the integration results in an actual harm to competition in the “market” for the added functionality through incompatibilities or distribution foreclosure, or, conversely, whether the integrated product is designed and licensed in such a way as to preserve competitive opportunities. It will often be entirely feasible to use standalone products in conjunction with integrated offerings, leaving customers free to take advantage of both approaches to product design.

¹⁰⁹ Even in those cases where a standalone offering exhibits platform characteristics, the functions it provides, the manner in which it provides them, and the interfaces (if any) it exposes will be different from those provided by functionality integrated into the defendant’s product. Third-party applications written to take advantage of the integrated capability generally will not function properly, or at all, if that functionality is missing, even if standalone software, exposing different interfaces, is added. The same is true of any features of the defendant’s product that depend upon integrated capabilities that have been removed in favor of third-party software. See New York, 224 F Supp 2d at 204 (“[T]he Windows operating system is not designed like an ‘à la carte restaurant menu’ from which blocks of software may be included or excluded at the whim of the OEM.”).

¹¹⁰ Id at 193.