No. 15-777

In the Supreme Court of the United States

SAMSUNG ELECTRONICS CO., LTD.,
SAMSUNG ELECTRONICS AMERICA, INC., AND
SAMSUNG TELECOMMUNICATIONS AMERICA, LLC,
Petitioner,

v.

APPLE, INC.,
Respondent.

On Writ of Certiorari to the United States Court of Appeals for the Federal Circuit

BRIEF OF BSA | THE SOFTWARE ALLIANCE AS *AMICUS CURIAE* IN SUPPORT OF NEITHER PARTY

JEFFREY J. VANDAM
Mayer Brown LLP
71 S. Wacker Drive
Chicago, IL 60606
(312) 782-0600

ANDREW PINCUS
Counsel of Record
PAUL W. HUGHES
Mayer Brown LLP
1999 K Street, NW
Washington, DC 20006
(202) 263-3000
apincus@mayerbrown.com

Counsel for Amicus Curiae

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Page(s)

Cases

Statutes and Regulations

TABLE OF AUTHORITIES—continued

Page(s)

Other Authorities

TABLE OF AUTHORITIES—continued

Page(s)

TABLE OF AUTHORITIES—continued

Page(s)

TABLE OF AUTHORITIES—continued

Page(s)

INTEREST OF THE *AMICUS CURIAE*

BSA | The Software Alliance is an association of the world's leading software and hardware technology companies. On behalf of its members, BSA promotes policies that foster innovation, growth, and a competitive marketplace for commercial software and related technologies. Because patent policy is vitally important to promoting the innovation that has kept the United States at the forefront of software and hardware development, BSA members have a strong stake in the proper functioning of the U.S. patent system.[1]

BSA's members advocate a balanced approach to patent enforcement litigation. BSA members are among the Nation's leading technology companies, producing much of the hardware and software that power computer and telecommunication networks. They thus pursue patent protection for their intellectual property and, as a group, hold a large number of patents. But, in light of their innovative products, they are frequently subject to unjustified patent infringement claims. BSA members therefore seek proper calibration of intellectual property rights.

The members of the BSA include Adobe, ANSYS, Apple, Autodesk, Bentley Systems, CA Technologies, CNC/Mastercam, DataStax, Dell, IBM, Intuit, Microsoft, Minitab, Oracle, salesforce.com, SAS Insti-

---

[1] Pursuant to Rule 37.6, amicus affirms that no counsel for a party authored this brief in whole or in part and that no person other than amicus and its counsel made a monetary contribution to its preparation or submission. The parties' letters consenting to the filing of this brief have been filed with the Clerk's office.

tute, Siemens PLM Software, Splunk, Symantec, Tekla, The MathWorks, Trend Micro and Workday.

SUMMARY OF ARGUMENT

Design patents provide important legal protection for software innovations, and effective remedies for infringement of design patents are therefore essential to provide the incentives needed to maintain investment in this critical sector of our Nation's economy.

Software companies account for a significant—and a quickly growing—amount of spending on research and development. In addition, software innovations significantly improve productivity and efficiency for the broader economy. Developers rely on appropriate legal protections to safeguard their innovations; absent such intellectual property rights, developers would have little incentive to create the remarkable array of software systems that now touch on virtually every aspect of the modern world.

Utility patents address innovative software functions; design patents are essential to protect innovative design. In the software context, that protection encompasses "graphical user interfaces"—the user-friendly visual displays that allow virtually any user to interact with the complex technology.

Take, for example, the personal computer. Early computers required operators with significant technical skill: a user typically had to type in commands via an appropriate string of letters and numbers. The graphical operating systems developed by Apple (for its Macintosh computers) and Microsoft (for the Windows operating system) enabled personal computers to reach a mainstream audience. Now, graphical user interfaces are found everywhere—on smart

phones, automobile dashboards, and home thermostats.

Novel design in these elements often provides an important component of a software system's value. A highly functional software program that lacks attractive, intuitive design will have significantly-reduced value in the marketplace. And, in circumstances where competing software systems offer similar functionality, innovative design is often the most important basis on which companies compete.

Design patents provide unique protection for this area of innovation. No other form of intellectual property—including utility patents, copyrights, and trade dress—serves as a substitute. Design patents, with effective remedies for infringement, are therefore important to prevent appropriation of novel software design.

ARGUMENT

Design Patents Are Essential To Maintain An Innovative Software Industry—And Therefore Should Receive Appropriate Legal Protection.

A. The Software Industry Is A Key Engine Of America's Economic Growth.

Software is a key driver of the modern economy—"many breathtaking software-implemented innovations power our modern world, at levels of efficiency and performance unthinkable even just a few years ago." David Kappos, An Examination of Software Patents (Nov. 20, 2012), http://goo.gl/UZGdvw. PricewaterhouseCoopers' recent survey of the world's top software firms found that, as software becomes integrated into an ever-expanding array of consumer

goods, "[s]oftware companies are going to be in a prime position to meet the needs of nontech companies that want valuable capabilities embedded into their products." PwC Technology Institute, PwC Global 100 Software Leaders, at 8 (March 2014), http://tiny.cc/ltalby.

For the industry to continue to innovate, appropriate legal protection for software innovations is essential. Otherwise, some companies will quickly pilfer novel software elements, preventing developers from recouping on their investments—ultimately diminishing investment and depressing innovation.

1. Software is a critical contributor to the U.S. economy.

The software industry contributes to the economy by investing substantial amounts in research and development. This R&D spending provides a significant number of well-paying jobs to the American workforce, and it creates new technologies that spur innovation and growth throughout the economy.

Software investment is significant—and rapidly growing. In 2015, the computing, electronics, and software industries accounted for roughly 37.5% of the Nation's entire research and development spending—for a total of more than $50 billion dollars annually. Barry Jaruzelski *et al.*, Global Innovation 1000: Innovation's New World Order, Strategy and Business (Winter 2015), http://tiny.cc/5i1kby. BSA member Microsoft alone spent more than $11.4 billion on R&D last year. *Ibid.*

Software is the fastest growing area of R&D spending in the entire economy. Between 2014 and 2015, the software and Internet sector increased R&D spending by 27.4%. *Ibid.* Over the past decade,

the sector's 13.2% average annual growth in R&D spending was the largest increase of any industry segment. *Ibid*.

The software industry garners more venture capital seed investment than any other sector, and it attracts the second-highest share of total investment. Nat'l Sci. Bd., Science & Engineering Indicators, at 6-90 (2016), http://tiny.cc/a25kby. The year-over-year increases in all stages of venture capital investment have jumped from $3.79 billion in 2010 to $5.34 billion in 2012 to $7.38 billion in 2014. *Id*. at 6-90 (Fig. 6-35).

U.S. software R&D spending now outstrips *industrial* R&D spending. The National Science Board recently confirmed that the information sector of the economy—two-thirds of which is driven by software—was responsible for 18% of business R&D in 2013, up from a 13% share in 2008. *Id*., at 4-55 (2016), http://tiny.cc/wx5kby.

This heavy investment in research has resulted in unprecedented innovation. Companies producing software are among the nation's leading innovators. And they have made the United States the world's leader in creating new software. Jaruzelski, *supra*. Software companies achieved an astonishing rate of innovation: 69% of software firms reported introducing a new product or service in 2013, versus a 9% average among all nonmanufacturing companies. Nat'l Sci. Bd., supra, at 6-5, http://tiny.cc/a25kby.

Software companies also are significant generators of high-paying jobs. Currently, software companies and related services employ millions of U.S. workers, paying salaries that are far above the national average. Victoria Espinel, BSA President and

CEO, Testimony before the United States Senate Committee on the Judiciary, at 3 (Sept. 16, 2015), http://tiny.cc/bgsrby. BSA has projected that jobs in the software sector will grow at a rate of 3.1% through 2020. BSA, Powering the Digital Economy: A Trade Agenda to Drive Growth, at 4 (2014), http://tiny.cc/risrby. Additionally, data "strongly suggest that software is a significant source of expanded employment" and acts as a stronger "employment multiplier" than in other industries, given that every ten jobs in software support an additional five jobs in other industries. Robert J. Shapiro, The U.S. Software Industry: An Engine for Economic Growth and Development, SIIA White Paper at 6-7 (2014), http://tiny.cc/sc3kby.

Most importantly, software today provides the backbone for virtually all businesses, because "software supports" critical business functions including "finance, human resources, operations and logistic, sale and market." Santanu Kumar Misra & Amitava Ray, Integrated AHP-TOPSIS Model for Software Selection Under Multi-criteria Perspective, in Driving the Economy Through Innovation and Entrepreneurship 879, 879 (2013).

Innovations in information technology provide significant efficiencies in other industries, thus improving economic growth. Software helps companies "collaborate more effectively internally and externally, scale operations faster, operate more efficiently, and innovate and experiment more strategically." The Boston Consulting Group ("BCG"), The Great Software Transformation, at 11 (2013), http://goo.gl/ftyEz0. For example, software is now responsible for 80% of innovation in the automobile industry, just one of many sectors where products are increasingly

relying on sophisticated software solutions to improve efficiency, safety, and functionality. Peter Andén et al., *The Perils of Ignoring Software Development*, McKinsey Quarterly (February 2015), http://tiny.cc/qn2kby.

Today, "all companies are effectively tech companies." BCG, *supra*, at 11. It is no surprise, therefore, that the "key factor responsible for reversing the 20-year productivity slowdown from the mid-1970s to the mid-1990s" was information technology innovation. Robert D. Atkinson & Andrew S. McKay, Info. Tech. & Innovation Found. (ITIF), Digital Prosperity: Understanding the Economic Benefits of the Information Technology Revolution, at 10 (2007), http://tiny.cc/xbclby.

All of this investment and innovation has made the U.S. software industry the world's leader. Eight of the top ten corporations on the Pricewaterhouse-Coopers Global 100 Software Leaders list are based here. PwC Global 100 Software Leaders, *supra*, at 4. Other countries are attempting to catch up quickly through aggressive education programs and strategic investment. Scott Andes & Mark Muro, *Software: America's Hidden Manufacturing Advantage*, Brookings Institution: The Avenue (Feb. 25, 2014), http://tiny.cc/ahalby.

2. Protection of software inventions against infringement is critical to continued economic growth.

Software innovators rely upon legal protection to secure their investments. Absent adequate protections, companies will invest less, innovate less, and contribute less to economic growth and job creation.

As then-PTO Director David Kappos explained, "patent protection is every bit as well-deserved for software-implemented innovation" as for earlier innovations "that enabled man to fly, and before that for the innovations that enabled man to light the dark with electricity, and before that for the innovations that enabled the industrial revolution." Kappos, *supra*. Given the importance of software in driving increased productivity and new products and services throughout the economy, as well as the vital contribution by the software industry itself, appropriate legal protection for software innovations is critical to maintaining and promoting economic growth.

Precluding theft of software design is necessary to ensure that those who invest in research and development reap proper rewards—and thus have a sufficient incentive to make the investments in the first place. Software companies rely on patent protection to protect and promote fair returns on their massive investments in research and development. Martin Goetz, *Misconceptions About Software Patents*, Forbes (Mar. 12, 2013), http://tiny.cc/l4clby. See also Julie E. Cohen & Mark A. Lemley, *Patent Scope and Innovation in the Software Industry*, 89 Cal. L. Rev. 1, 5 (2001) ("[B]oth economic theory and practical experience suggest that the availability of patents for software promotes innovation by supplying (additional) incentives to inventors.").

Leading software companies obtain patents to protect software innovations. See Maulin Shah, *Software Patents Are Resilient in the Wake of Alice Corp. v. CLS Bank*, PatentVue (Sept. 9, 2015), http://tiny.cc/5qdlby. Companies' continued reliance

on such patents demonstrate their importance in protecting innovation.

These legal protections are particularly crucial for small and startup companies. Effective legal protection, principally through patents, "is significantly correlated with any of several variables that are indicators of the firm's progress through the venture capital cycle (including number of rounds, total investment, and longevity)." Ronald J. Mann & Thomas W. Sager, *Patents, Venture Capital, and Software Start-Ups*, 36 Res. Pol'y 193, 194 (2007), http://tiny.cc/pydlby. Indeed, "[f]or small-scale developers, a software patent may be the only effective way to develop and market an innovative new form of software without having key features of the software scooped up by large competitors with no compensation to the small innovator." Richard S. Gruner, *Better Living Through Software: Promoting Information Processing Advances Through Patent Incentives*, 74 St. John's L. Rev. 977, 1065-1066 (2000) (noting that software patents also benefit larger companies by allowing them to play defense and ensure their control over their new designs).

B. Design Patents Provide Important Legal Protection For Software.

Utility patents protect software's functional aspects. Another just-as-critical element of software—particularly software powering the myriad devices that use screens to convey information and receive a user's commands—is the "graphical user interface" through which user interaction occurs. Design patents play an essential role in protecting this critical component of software's value.

Early computing systems lacked user-friendly interfaces; they were text-only systems, such as the classic DOS interface made up of letters, numbers, and a blinking cursor. This approach had significant built-in limitations—only an individual with significant technological knowledge could make use of it, greatly reducing the popular adoption of such software and the devices that it powered.

Graphical user interfaces substitute novel, user-friendly images for technical jargon, enabling the user to access the software's varied functions through common-sense icons and words. These intuitive, user-friendly graphic interfaces enable the integration of software into the myriad of devices and products in which it is used today.

By making software functions accessible without technological expertise, these interfaces are largely responsible for the rise of personal computing. Email systems, word processing, and an array of other productivity-enhancing technology became accessible to any user. The interfaces in Apple's Macintosh operating system and Microsoft's Windows program—which combined easily-understood interfaces with these functions—were therefore instrumental developments in computer technology, significantly contributing to the computer and Internet revolution that has touched virtually every aspect of modern life.

Now, the software that powers our smart phones, our automobile dashboards, and even our thermostats similarly relies on graphical interfaces that permit individuals—with no technological training—to interact with sophisticated software systems. These design elements, combined with the software's

underlying functionality enable users to realize the true value of the product

Interface design is also a key means by which software companies compete: while the underlying software functions may be similar—indeed, perhaps indistinguishable to an average consumer—the *design* of the user interface is at the forefront of a product. How software looks and feels to the consumer is a key driver in the marketplace.

User interfaces thus provide a substantial portion of the value of a particular piece of software, making the functional features of software accessible, intuitive, and aesthetically pleasing for users. Protection of such software features, through design patents that receive appropriate legal protection, is thus crucial to continued development and innovation in this important industry. That is particularly true because design patents provide legal protections that are not available through utility patents, copyrights, or trade dress.

1. Design patents play an important role in protecting software innovations.

Design patents are not at all a new feature on the U.S. intellectual property landscape, having arrived by statute in 1842. See generally Thomas B. Hudson, *A Brief History of the Development of Design Patent Protection in the United States*, 30 J. Pat. Off. Soc'y 380 (1948). Design patents have protected the appearance of a great number of iconic American designs and images. The Eames chair was protected by a design patent (U.S. Patent No. D155,272 (issued Sept. 20, 1949)) as was the Gibson "Flying V" guitar (U.S. Patent No. D181,867 (issued Jan. 7, 1958)),

and, perhaps most famously, the Coca-Cola bottle (U.S. Patent No. D48,160 (issued Nov. 16, 1915)).

As the U.S. economy increasingly shifts to the digital world, innovators seek this same form of protection for software's design features, such as graphical images and user interfaces that developers employ in making their products both unique and attractive to consumers—the images that consumers see on the screen of the device (smartphone, computer, etc.) that is using the software.

There is, therefore, a thirty-year history of design patents in the software context. Xerox Corporation was an early applicant for such protection; it received patents for a menu bar and several icons it had designed—small images of a telephone, wastebasket, and dictionary, among others—in May 1988. Daniel J. Kluth & Steven W. Lundberg, *Design Patents: A New Form of Intellectual Property Protection for Computer Software*, 70 J. Pat. & Trademark Off. Soc'y 847, 847, 853 (1988). Patent commentators at the time treated this as revolutionary, announcing that "Xerox has discovered a new form of protection for the 'look and feel' of computer software." *Id.* at 847.

In 1996, the PTO specifically recognized the importance of design to software innovation. It released interim guidelines for issuing design patents for computer-generated icons, and sought notice and comment on the interim standards. See 60 Fed. Reg. 52,170-01 (1995).

PTO's final guidance included a statement of "[g]eneral [p]rinciple" that "[t]he PTO considers designs for computer-generated icons embodied in articles of manufacture to be statutory subject matter el-

igible for design patent protection under section 171." 61 Fed. Reg. 11,380-03, 11,381 (1996). Therefore, "if an application claims a computer-generated icon shown on a computer screen, monitor, other display panel, or a portion thereof, the claim complies with the 'article of manufacture' requirement of section 171." *Ibid.*

Importantly, the PTO's guidelines provided broad protection for such icons and images: the agency stated that it would "simply require a depiction of an article of manufacture in either solid or broken lines." *Ibid.*

The guidelines' flexibility with respect to graphical user interfaces, is particularly important in ensuring appropriate protection. For example, a "design patent applicant need not embody the entire design in a single set of figures," because the designer may "use a combination of phantom and solid lines to selectively protect certain aspects of the design in a particular design patent, and then use the combination of multiple design patents to protect the entire ornamental look of the device." Timothy W. Menasco, *Electronics Cos. Shouldn't Shy Away from Design Patents*, Law360 (Sept. 18, 2015), http://tiny.cc/f45oby (describing how protection for various design elements of Apple's iPhone constitutes protection of the design of the device as a whole).[2]

---

[2] These guidelines were later amended to encompass animated designs, providing an important additional form of protection for software design and thus "a powerful tool to obtain meaningful protection for innovators." *See* David Leason, *Design Patent Protection for Animated Computer-Generated Icons*, 91 J. Pat. & Trademark Off. Soc'y 580, 592-593 (2009).

The industry responded quickly to PTO's 1996 guidelines. Within a year, the number of design patents granted for graphical computer designs was greater than the number for the prior ten years combined. Jason J. Du Mont & Mark D. Janis, *Virtual Designs*, 17 Stan. Tech. L. Rev. 107, 134 (2013).

The PTO's endorsement of design patents for software elements directly preceded the explosion of mobile device technology—particularly smart phones and the "apps" that populate them. See Standing Committee on the Law of Trademarks, Industrial Design, and Geographical Indications, World Intellectual Property Organization (WIPO), *Proposal by the Delegations of the United States of America and Japan*, at 2 (Apr. 11, 2016), http://tiny.cc/1gmoby. Companies' ability to protect their software designs provided an important impetus for substantial investment in innovative design.[3]

Indeed, grants of graphical design patents increased dramatically in the mid-2000s, and the number continues to rise. Du Mont & Janis, *supra*, at 133-138. In the past twelve months, the PTO has issued more than 2,000 design patents related to software and graphical user interfaces.[4] Design pa-

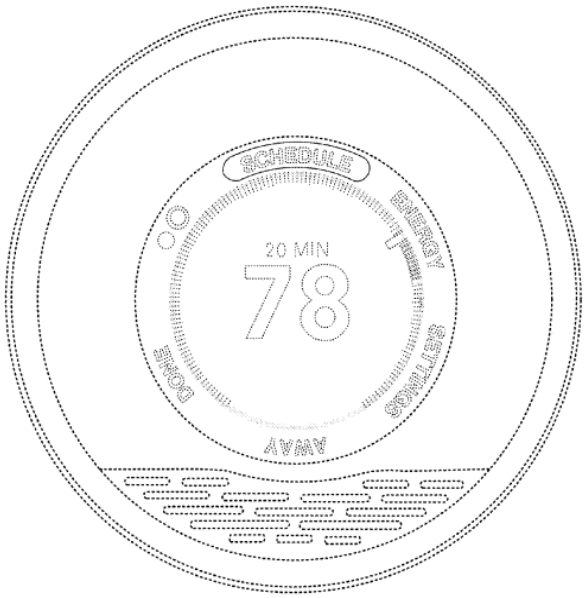
---

[3] Although Section 171 provides design patent protection for "any new, original and ornamental design for an article of manufacture," the patent owner may sue for infringement when "the patented design or any colorable imitation thereof" is applied to "*any article of manufacture* for the purpose of sale" or "*any article of manufacture*" to which the design or colorable imitation is applied, sold, or exposed for sale. 35 U.S.C. § 289(1) (emphasis added). The statute thus makes clear that the design patent is violated when the design is applied to an article of manufacture different from the one for which it was invented.

[4] We have examined the PTO's weekly gazette, which publishes statistics regarding granted design patents. We have totaled

tents in the software field are growing at a rate faster than any other industry and consumers have benefited from the increased ease of accessing software functions resulting from this investment in improved design. Rachel Stigler, *Ooey GUI: The Messy Protection of Graphical User Interfaces*, 12 Nw. J. Tech. & Intell. Prop. 215, 239 (2014).

Today, design patents protect a broad range of graphical user interfaces. Patents granted by the PTO run the gamut from the familiar to the cutting-edge. The animated graphical interface of the Internet-enabled Nest learning thermostat, for example, received design patent protection in 2013. U.S. Patent No. D687,047 (issued July 30, 2013).



---

the design patents granted in the D14 category, subclasses 485 to 495, which cover software and graphical user interfaces.

The ubiquity of software design patents demonstrates the importance to software innovators (and to consumer product companies that integrate software into their products) of protecting innovative designs. This is due in substantial part to the fact that "high-technology is not often clearly demonstrable except through the goods' outer appearance, which is frequently the target of copyists," meaning that "[d]esign patents have proved to be especially important" for such products. Barton Beebe, *Intellectual Property Law and the Sumptuary Code*, 123 Harv. L. Rev. 809, 863-864 (2010).

As Professor Beebe explained, "[d]esign patents enable the designers of such products to convert the absolute utility that they have created into clearly demonstrable (and protectable) forms of relative utility, which may be the primary form of utility that high-technology consumers ultimately desire." *Id.* at 864. Together, design patents and utility patents provide complementary protections that allow software creators to realize the full value of their innovations.

In addition, software design patents—like all design patents—promote competition that, in turn, produces more innovation. By enabling a software maker to secure intellectual property rights to graphical user interfaces and other ornamental features of software, design patents make the design elements a basis on which different software systems compete for market share. If the graphical user interface were not protectable, then all software developers would simply pilfer whatever *existing* interfaces are preferred in the marketplace—gutting any incentive for further innovation.

Finally, design patents enable software creators to continue to develop their innovative designs. After receiving an initial design patent, PTO guidelines permit holders of design patents to apply for further design patents "to add slight variations or a different focus of protection to an already patented design." Stigler, *supra*, 242. These continuation patents are retroactive to the date of the original design patent and allow a developer "to stay one step ahead of infringers," which incentivizes creative designs that benefit consumers. *Id.* at 242-243.

Design patents for software thus foster innovation by promoting competition among market participants based on novel designs. They enable companies to invest substantial sums in software development relating to the "look" and "feel" of software to consumers, because the design patent protects novel graphical user interfaces against infringement.

Put differently, in the current environment, a highly-functional software package has reduced market value absent innovative design; for this reason, design patents safeguard a very substantial aspect of the value of software innovation. Continued protection is therefore critical to encourage companies to invest substantial R&D capital in software development.

2. Design patents provide protections for software that complement other forms of intellectual property and, in combination, properly incentivize innovation.

Design patents are an essential aspect of the multifaceted intellectual property regime that protects software inventions. They secure important

rights not protected by a utility patent, copyright, or trade dress.

A household lamp, as Professor McCarthy explains, may be protected by multiple different rights: the lamp base (assuming it qualifies as a "work of art") is copyrightable; the electric circuit or bulb is subject to a utility patent; the label on the lamp can serve as a trademark; and, finally, the ornamental shade can receive a design patent. 1 McCarthy on Trademarks and Unfair Competition § 6.5 (4th ed. 1996). In the landmark decision *Mazer* v. *Stein*, 347 U.S. 201 (1954), the Court concluded that, while a lamp itself is functional, its base may nonetheless be subject to copyright protections if it qualifies as a "work of art." See also *Kohler Co. v. Moen Inc.*, 12 F.3d 632, 638 (7th Cir. 1993) ("[C]ourts have consistently held that a product's different qualities can be protected simultaneously, or successively, by more than one of the statutory means for protection of intellectual property.").

Design patents provide a critical complement to utility patents, because the latter do not protect designs. See 35 U.S.C. § 101. While utility patents provide important protection for a variety of innovative software functions (such as data compression or cryptography, to name just a few areas in which software utility patents are crucial), they cannot protect the "look" and "feel" of software.

The protection provided by copyrights also differs materially from design patents. "Unlike a patent, a copyright gives no exclusive right to the art disclosed; protection is given only to the expression of the idea—not the idea itself." *Mazer*, 347 U.S. at 217. Thus, a copyright protects "art" while a design pa-

tent protects "the invention of original and ornamental design." *Id*. at 218.

A software copyright, for example, typically covers software *code*, but not necessarily its designs. Design patents, by contrast, protect the intricate, ornamental individual elements of the graphical interface. See Michael Risch, *Functionality and Graphical User Interface Design Patents*, 17 Stan. Tech. L. Rev. 53, 90 (2013).

In some circumstances, specific images may also be subject to copyright. But the proof necessary to demonstrate infringement differs between copyright and design patents. Copyright infringement requires "substantial similarity not only of the general ideas but of the expressions of those ideas." *Sid & Marty Krofft Television Prods., Inc.* v. *McDonald's Corp.*, 562 F.2d 1157, 1164 (9th Cir. 1977). This "substantial similarity" analysis applies to software copyright claims. *Computer Assocs. Int'l, Inc.* v. *Altai, Inc.*, 982 F.2d 693, 706 (2d Cir. 1992).

To show infringement of a design patent, "[t]he patentee must establish that an ordinary observer, familiar with the prior art designs, would be deceived into believing that the accused product is the same as the patented design." *Richardson* v. *Stanley Works, Inc.*, 597 F.3d 1288, 1295 (Fed. Cir. 2010). The test considers the "overall effect on the designs" in determining whether there is "market confusion." *Ibid*. Thus, the "the 'colorable imitation' standard of the design patent statute involves the concept of equivalents." *Pac. Coast Marine Windshields Ltd.* v. *Malibu Boats, LLC*, 739 F.3d 694, 701 (Fed. Cir. 2014). In the context of software graphical user interfaces, this test provides protection that may not be

available under copyright law (but is limited by the shorter term of patent protection).

Trade dress principles also fall short of providing the necessary protection for software design. Trade dress is generally defined as "the total image, design, and appearance of a product and may include features such as size, shape, color, color combinations, texture or graphics." *Clicks Billiards, Inc.* v. *Sixshooters, Inc.*, 251 F.3d 1252, 1257 (9th Cir. 2001). This form of intellectual property protection is difficult to obtain, particularly for a graphical user interface, because the interface must "reach near-famous status before reaping any protection benefits." Stigler, *supra*, at 216.

That limitation poses special problems when a smaller innovator launches a new product. To the extent that widespread commercial acceptance of a product factors into the trade dress analysis, newly-developed software will garner little protection. But products are often most vulnerable to imitation by large incumbents at this early juncture—when software is released and thus seeking widespread market adoption. Design patents are crucial to ensure that the creator of an innovative design is able to obtain a proper toehold in the marketplace.

Trade dress suffers from other limitations, too. Plaintiffs attempting to prove a trade dress violation must generally show "(1) that its claimed dress is nonfunctional; (2) that its claimed dress serves a source-identifying role either because it is inherently distinctive or has acquired secondary meaning; and (3) that the defendant's product or service creates a likelihood of consumer confusion." *Millennium Labs., Inc.* v. *Ameritox, Ltd.*, 817 F.3d 1123, 1126 n.1 (9th Cir. 2015).

In the context of software design, these elements may be difficult, if not impossible, to satisfy. For example, the design elements may not be inherently distinctive within the meaning of the trademark laws. Typically, this showing can be made only after the trade dress is used in commerce. Design patents, by contrast, face no such incongruous requirements; "[u]nlike trade dress protection, there is no requirement that a design be 'used in commerce' prior to applying for a design patent." See Burstein, *supra*, at 331.

Similarly, a competitor may avoid trade dress infringement through clear attribution of a product, which could negate the needed showing of a "likelihood of consumer confusion." For example, a software developer could closely mimic the "look" and "feel" of a certain successful graphical user interface, but it could include a label (or graphic) clearly identifying the software in question as the product of a different company, thus precluding consumer confusion. While this behavior could defeat a trade dress claim, the protections of a design patent are not so easily overcome. See Sarah Burstein, *Moving Beyond the Standard Criticisms of Design Patents*, 17 Stan. Tech. L. Rev. 305, 331 (2013).

Beyond that, trademarks can lose protection if they become a generic mark for a particular sort of product. See, *e.g.*, *King-Seeley Thermos Co.* v. *Aladdin Indus., Inc.*, 321 F.2d 577, 578 (2d Cir. 1963) (discussing genericness of "thermos"); *Bayer Co.* v. *United Drug Co.*, 272 F. 505, 512 (S.D.N.Y. 1921) (discussing genericness of "aspirin"). Design patents do not suffer from this risk of becoming an unprotected generic, but they are limited in duration to no more than 15 years. See 35 U.S.C. § 173.

Indeed, courts invalidating trade dress protection based on the presence of a functional element have observed that design patent protection is a comparatively superior method of safeguarding design innovations. Jason J. Du Mont & Mark D. Janis, *The Origins of American Design Patent Protection*, 88 Ind. L.J. 837, 844 & nn.38-39 (2013). It is no surprise, therefore, that design patents provide important protections for software innovations.

* * *

In order to properly incentivize the creation of new, innovative software, and to protect developers against unauthorized appropriation by competitors, software design patents are essential. They are often the only meaningful way to protect the look and feel of innovative graphical user interfaces, which are an integral part of consumers' experience of the software itself. Effective remedies for design patent infringement are therefore essential to protect this valuable form of intellectual property.

CONCLUSION

Design patents provide an essential element of legal protection for software innovations. This Court should ensure that design patents in the software context receive appropriate protection against infringement.

Respectfully submitted.

ANDREW PINCUS
Counsel of Record
PAUL W. HUGHES
Mayer Brown LLP
1999 K Street, NW
Washington, DC 20006
(202) 263-3000
apincus@mayerbrown.com

JEFFREY J. VANDAM
Mayer Brown LLP
71 S. Wacker Drive
Chicago, IL 60606
(312) 782-0600

Counsel for Amicus Curiae

JUNE 2016