Case No. 2017-2145

UNITED STATES COURT OF APPEALS

FOR THE FEDERAL CIRCUIT

CISCO SYSTEMS, INC.,

Plaintiff-Appellant,

v.

ARISTA NETWORKS, INC.,

Defendant-Appellee.

Appeal from the United States District Court for the Northern District of California in Case No. 14-cv-5344, Judge Beth Labson Freeman

---

BRIEF OF AMICUS CURIAE ELECTRONIC FRONTIER FOUNDATION IN SUPPORT OF DEFENDANT-APPELLEE AND AFFIRMANCE

Michael Barclay
(*Principal Attorney of Record*)
Vera Ranieri
ELECTRONIC FRONTIER FOUNDATION
815 Eddy Street
San Francisco, CA 94109-7701
Tel: (415) 436-9333
Fax: (415) 436-9993
michael@eff.org

December 26, 2017

Attorneys for Amicus Curiae Electronic Frontier Foundation

CERTIFICATE OF INTEREST

Pursuant to Federal Circuit Rules 29(a) and 47.4, counsel for Amicus Curiae certifies that:

1. The full name of the amicus I represent is:

Electronic Frontier Foundation

2. The name of the real party in interest (if the party named in the caption is not the real party in interest) I represent is:

N/A

3. All parent corporations and any publicly held companies that own 10 percent or more of the stock of the amici curiae I represent are:

None.

4. The names of all law firms and the partners or associates that appeared for the amicus I represent or are expected to appear in this Court are: Michael Barclay and Vera Ranieri, Electronic Frontier Foundation, San Francisco, California.

5. The title and number of any case known to counsel to be pending in this or any other court or agency that will directly affect or be directly affected by this court's decision in the pending appeal are: None.

December 26, 2017

/s/ Michael Barclay
Michael Barclay
Attorneys for Amicus Curiae
Electronic Frontier Foundation

TABLE OF CONTENTS

Page

TABLE OF AUTHORITIES

Page(s)

Cases

Statutes

Other Authorities

TABLE OF CONVENTIONS

Cisco	Plaintiff-Appellant Cisco Systems, Inc.
Arista	Defendant- Appellee Arista Networks, Inc.
Cisco Br.	Cisco's Opening Brief, dated September 13, 2017
Arista Br.	Arista's Response Brief, dated December 22, 2017
Mathworks Br.	Amicus Brief of The Mathworks, Inc. et al., dated September 20, 2017
Appx	The appellate Appendix in this appeal

STATEMENT OF IDENTITY AND INTEREST OF AMICUS CURIAE[1]

The Electronic Frontier Foundation ("EFF") is a non-profit civil liberties organization that has worked for 26 years to protect consumer interests, innovation, and free expression in the digital world. EFF and its more than 40,000 dues-paying members have a strong interest in helping the courts and policymakers strike the appropriate balance between intellectual property and the public interest, and ensuring that copyright law serves the interests of creators, innovators, and the general public.

Pursuant to Fed. R. App. P. 29(a), all parties have consented to the filing of this brief.

INTRODUCTION AND SUMMARY OF ARGUMENT

The Court should affirm the judgment below.

First, the works in question (to the extent they can be adequately discerned) are not copyrightable under Ninth Circuit law. Cisco and its amici rely on this Court's holding, in *Oracle*,[2] that the Ninth Circuit would treat a work as copyrightable if there were multiple ways to express the underlying

[1] No party's counsel authored this brief in whole or in part. Neither any party nor any party's counsel contributed money that was intended to fund preparing or submitting this brief. No person other than amicus, its members, or its counsel contributed money that was intended to fund preparing or submitting this brief. Web sites cited in this brief were last visited on December 18 & 19, 2017.

[2] *Oracle Am., Inc. v. Google Inc.*, 750 F.3d 1339, 1367 (Fed. Cir. 2014) ("*Oracle*").

idea. But a subsequent Ninth Circuit opinion rejected that approach. Now that the Ninth Circuit has clarified its law, this Court should revisit its holding in *Oracle*, and find the works at issue uncopyrightable.

And the flaws in Cisco's copyrightability analysis do not end there. The overwhelming body of precedent weighs against granting copyright protection for computer menu commands similar to those at issue, under 17 U.S.C. § 102(b) ("§ 102(b)"). What's more, Cisco appears to be claiming that Arista infringed a copyright in a so-called compilation that was selected and arranged years later, solely for purposes of this litigation.

Second, even if the Court affirms the copyrightability of the "compilations" at issue, the facts, the law, and strong public policy support the jury's scènes à faire verdict. Indeed, if courts are going to grant copyright protection to functional works, robust defenses such as scènes à faire help ensure that protection does not impede competition and innovation. The jury did its job, and this Court should leave its conclusion undisturbed.

ARGUMENT

I. THE NINTH CIRCUIT HAS REJECTED *ORACLE*'S APPROACH TO COPYRIGHTABILITY

Cisco and its amici rely on *Oracle* for the proposition that the works at issue are copyrightable because there were multiple ways to express the same idea. *Oracle*, 750 F.3d at 1367. *See* Cisco Br. at 38, 40; Mathworks Br. at 9.

Oracle is no longer good law for that proposition.

A. *Bikram's Yoga* Resolved an Issue this Court Believed the Ninth Circuit Had Not Decided

Between the time of the 2014 *Oracle* opinion and this appeal, the Ninth Circuit issued its opinion in *Bikram's Yoga College of India, L.P. v. Evolation Yoga, LLC*, 803 F.3d 1032 (9th Cir. 2015). That opinion clarified the Ninth Circuit's view of copyrightability, a question this Court believed was unresolved. In light of this new authority, the works at issue in this case and in *Oracle* are not copyrightable under Ninth Circuit law.

1. *Oracle* Assumed that the Ninth Circuit Had Not Addressed "the Precise Issue" in that Case

In order to decide the copyrightability issue, the *Oracle* court had to determine whether the structure, sequence, and organization of the Java API packages was copyrightable under § 102(b). *Oracle*, 750 F.3d at 1364-65. The Court declined to follow *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 49 F.3d 807 (1st Cir. 1995) ("*Lotus*"), *aff'd by an equally divided court*, 516 U.S. 233 (1996). The Court believed that *Lotus* was distinguishable and that the Ninth Circuit would not follow the *Lotus* court's reasoning. 750 F.3d at 1365-66. Given a belief that the Ninth Circuit had not "addressed the precise issue" in that case, this Court came to its own conclusion about what the Ninth Circuit would do:

> We agree with Oracle that, under Ninth Circuit law, an original work—even one that serves a function—is entitled to copyright protection as long as the author had multiple ways to express the underlying idea.

Id. at 1367.

Even assuming this was an accurate statement of Ninth Circuit law in 2014, it is not an accurate statement now.

2. *Bikram's Yoga* Shows that this Court's "Multiple Ways to Express" Reasoning Is Not Good Law in the Ninth Circuit

In October 2015, the Ninth Circuit considered a copyright claim in a sequence of yoga poses. *Bikram's Yoga*, 803 F.3d 1032. Bikram Choudhury and his company, Bikram's Yoga, had sued a competitor who used the same sequence. Simply put, Choudhury was claiming copyright in a process for improving one's health by practicing certain yoga poses in a specific order.

Relying on § 102(b), the court concluded that a "Sequence" of 26 yoga poses and two breathing exercises, performed in a particular order, was not subject to copyright protection. Writing for the court, Judge Wardlaw stressed that copyright recognizes a "vital distinction" between ideas and expression, so "the copyright for a work describing how to perform a process does not extend to the process itself." *Id.* at 1037-38. Choudhury himself described his Sequence as a "system" or "method" to use yoga to optimize the body's health and function. *Id*. at 1038-39. Given that, Judge Wardlaw had little difficulty

concluding that the idea/expression dichotomy, codified in § 102(b), precluded copyright protection of the Sequence. *Id*. at 1039-40.

Bikram's Yoga also considered, and rejected, the argument that a system or method can be copyrightable if there were different ways to "express" that system. The Ninth Circuit held:

> It makes no difference that similar results could be achieved through a different organization of yoga poses and breathing exercises. . . . the possibility of attaining a particular end through multiple different methods does not render the uncopyrightable a proper subject of copyright. Though it may be one of many possible yoga sequences capable of attaining similar results, the Sequence is nevertheless a process and is therefore ineligible for copyright protection.

803 F.3d at 1042 (citations, quotations, and footnote omitted).

A finding of copyrightability based on the fact that "the author had multiple ways to express the underlying idea," *Oracle,* 750 F.3d at 1367, cannot be reconciled with the Ninth Circuit's holding above. This Court's "multiple expressions" theory is not the law in the Ninth Circuit, and therefore should not apply in this case.

B. The Court Should Take this Opportunity to Correct Its Interpretation of the Ninth Circuit's Approach to Copyrightability

It is important that the Court recognize the change of law outlined above.

First, *Oracle* has been criticized as "deeply flawed and at odds with more than two decades of copyright precedents applying copyright law to computer

programs." Pamela Samuelson, *Three Fundamental Flaws in CAFC's Oracle v. Google Decision*, 37 Eur. Intell. Prop. Rev. 702, 708 (2015).[3] In particular, it created an unnecessary split with the First Circuit holding in *Lotus,* 49 F.3d 807. Using language quite similar to *Bikram's Yoga*, *Lotus* also rejected Cisco's "multiple ways to express" argument:

> The fact that Lotus developers could have designed the Lotus menu command hierarchy differently is immaterial to the question of whether it is a "method of operation." . . . The "expressive" choices of what to name the command terms and how to arrange them do not magically change the uncopyrightable menu command hierarchy into copyrightable subject matter.

49 F.3d at 816. The *Oracle* decision was based in part on the mistaken premise that the Ninth Circuit would not follow *Lotus*, *see Oracle,* 750 F.3d at 1365-66. *Bikram's Yoga* dispels that perception and creates an opportunity to reconcile the circuits.

Second, correcting *Oracle* avoids the serious practical problems that arise if courts treat methods of operating a computer as copyrightable. Indeed, as outlined in detail in amicus briefs filed by numerous prominent computer scientists in both *Oracle* appeals, the pervasive belief and expectation that APIs were uncopyrightable was essential to the development of modern computers and the Internet. *See* Corrected Brief of Amici Curiae Computer Scientists, filed May 30, 2013 in the first appeal in *Oracle* (Fed. Cir. Case No. 2013-1021,

[3] Available at https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2643840.

Docket No. 118);[4] Brief of Computer Scientists as Amici Curiae, filed May 30, 2017 in the second appeal in *Oracle* (Fed. Cir. Case No. 2017-1118, Docket No. 175).[5]

Reversing that expectation has already sparked "a new wave of litigation concerning copyright and interoperability"—including this very case. Jonathan Band, *Software Copyright Litigation After Oracle v. Google*, Disco (January 9, 2017).[6] *Oracle* has fostered a form of "appellate forum shopping," where a plaintiff bringing an API-related case that properly sounds in copyright nonetheless tacks on a patent claim—perhaps even knowing that the claim wouldn't survive trial—to ensure that any appeal will go to this Court. Such plaintiffs want to avoid regional circuits, such as the Ninth and First Circuits, that would make short shrift of API copyright claims. *See* Peter Menell, *API Copyrightability Bleak House: Unraveling and Repairing the* Oracle v. Google *Jurisdictional Mess*, 31 Berkeley Tech. L.J. 1515, 1581 (2016) (*Oracle* "motivates software intellectual property owners to bundle patent and copyright claims in order to take advantage of the Federal Circuit's expansive interpretation of software copyright protection.").[7] Notably here, the jury ruled

[4] Available at https://www.eff.org/document/amicus-brief-computer-scientists.
[5] Available at https://www.eff.org/document/computer-scientists-amicus-brief.
[6] Available at http://www.project-disco.org/intellectual-property/010917-software-copyright-litigation-oracle-v-google/.
[7] Available at https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2859740.

against Cisco on its patent claim, Appx1430-1431, but Cisco's copyright appeal has gone to this Court. Cisco's brief barely mentions (if at all) the adverse jury verdict on patent infringement, which Cisco did not appeal.

II. CISCO'S MULTIWORD COMMANDS AND COMPILATIONS ARE UNCOPYRIGHTABLE UNDER § 102(B)

As an alternate ground for affirmance, the Court should hold that the compilation of the unprotected multiword commands at issue here are uncopyrightable under § 102(b).

A. Courts in Several Circuits Have Rejected Similar Copyright Claims

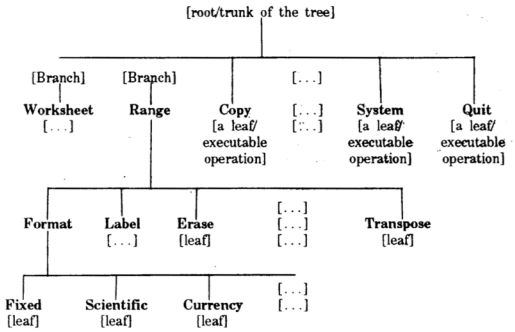
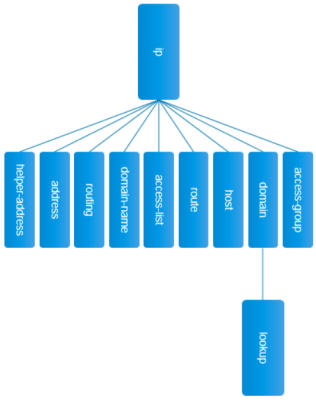
Courts around the nation have held that names of commands used to control a computer program are not properly the subject of copyright, despite the fact that there are "multiple ways to express" the commands. Cisco's CLI commands and their arrangements are not meaningfully different from those rejected methods of operation.

In particular, Cisco's CLI commands are directly comparable to the spreadsheet commands and hierarchy at issue in *Lotus*. The First Circuit opinion describes them briefly, and several predecessor district court opinions offer more detail. *See Lotus Dev. Corp. v. Borland Int'l, Inc.*, 831 F. Supp. 202 (D. Mass. 1993) ("*Borland III*"), and *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37 (D. Mass. 1990) ("*Paperback*"). There are 469 commands

arranged in a hierarchy of more than 50 menus and submenus, called a menu "tree." *Lotus*, 49 F.3d at 809-10. The menus contain a serious of words that are command choices such as "Copy," "Print," "Quit," and "File." *Id.* at 809; *Borland III*, 831 F. Supp. at 210; *Paperback*, 740 F. Supp. at 64. The top line of the main menu presents choices of the commands "Worksheet Range Copy Move File Graph Data Quit." Choosing certain of those commands brings up a list of subchoices. For example, to select the "Currency" operation, the user would navigate the menu tree through "Range," "Format," and then "Currency." *Borland III*, 831 F. Supp. at 210. Using these hierarchical commands, the user operates the electronic spreadsheet.

Cisco's arrangement does the same thing here; the only difference is that its commands operate network switches instead of a spreadsheet. Cisco Br. at 6. Like Lotus, Cisco "organized and arranged the multiword commands (and related responses) into different, particular hierarchies." *Id*. at 9. The examples of Cisco menu trees shown on pages 9-10 of the Cisco Br. operate the same way as the Lotus 1-2-3 menu tree, except that Cisco uses commands suitable for a network switch such as "ip" or "show." *Id*.

The following comparison shows how the two menu trees are arranged in a similar manner.

Portion of Lotus 1-2-3 Menu Tree *Borland III*, 831 F. Supp. at 210	**Cisco Menu Tree, "ip" hierarchy Cisco Br. at 9**
	

Lotus argued that the words it chose for its commands, and their hierarchy, were copyrightable because there were supposedly expressive choices to use for those words. For example, "Quit" could be renamed "Exit," and "Copy" could be renamed "Clone," "Duplicate," or others. *Lotus*, 49 F.3d at 810-11. Similarly, Cisco argues here that it had numerous choices to select its network command names and organization, such as renaming "ip" either "ipv4" or "internet-protocol" or others. Cisco Br. at 11-14.

The *Lotus* district court held that these so-called expressive choices meant that the menu command hierarchy was copyrightable. *Lotus*, 49 F.3d at 811. The First Circuit reversed, relying on § 102(b). It held that the selection of names of menu items in the Lotus 1-2-3 computer program "is not copyrightable because it is part of Lotus 1-2-3's 'method of operation.'" *Id.* at 816. The Lotus 1-2-3 menu commands were like buttons on a VCR; in both cases the user

operates the spreadsheet program or the videotaping machine by pushing buttons. *Id*. at 817. Or, as Arista's expert explained to the jury, the Cisco CLI commands were like "knobs" on an old-style stereo. Arista Br. at 8, 42.

Given the similarity of function between the Lotus 1-2-3 menu commands and the Cisco CLI commands, the same result should occur here. That is particularly the case given that the Ninth Circuit has now agreed with the First Circuit that "expressive" choices of what to name commands does not "magically" render otherwise uncopyrightable subject matter copyrightable. *Lotus*, 49 F.3d at 816.

Other circuits considering the copyrightability of menu commands have reached the same result. *Gates Rubber Co. v. Bando Chem. Industries, Ltd.,* 9 F.3d 823 (10th Cir. 1993), involved a computer program used to select industrial machine belts. The Tenth Circuit vacated a district court decision holding that the program's menu commands were copyrightable, and gave the district court instructions for further consideration of the issue. *Id*. at 843-44. On remand, the district court held that the menu commands were uncopyrightable, citing *Lotus* approvingly. *Gates Rubber Co. v. Bando Chem. Indus., Ltd.*, No. 92–S–136, slip opn. at 6-7 (D. Colo. filed June 12, 1995) (not appealed).[8] Similarly, *MiTek Holdings, Inc. v. Arce Engineering Co., Inc.*, 89 F.3d 1548 (11th Cir. 1996) dealt

[8] Available at https://www.eff.org/document/gates-rubber-v-bando-district-court-remand-order

with a wood truss layout computer program. Relying on *Lotus* and § 102(b), the Eleventh Circuit held that the program's menu and submenu command tree structure was not copyrightable. *Id.* at 1556-57.

B. Cisco Cannot Claim Infringement of a Compilation That Did Not Exist Prior to Litigation

Even if this Court finds that compilations of functional commands are copyrightable, there can be no infringement if the particular "compilations" at issue in this case do not actually embody any expression in existence prior to Arista's alleged infringement.

A compilation is protectable to the extent it embodies some creative expression in the selection and arrangement of materials. 17 U.S.C. § 101; *see also Feist Publ'ns, Inc. v. Rural Tel. Serv. Co., Inc.*, 499 U.S. 340, 348 (1991). For example, a publisher may possess a thin copyright in a compilation of particular excerpts from public domain Jane Austen novels used to show a theme, assuming there was creativity and expression embodied in the selection of the excerpts.

But the copyright in a compilation only exists with respect to the compilation itself, not to the underlying work, and only to the extent the compilation embodies an author's original expression. *Id.* at 348-49. Thus, if a scholar were to select a few of the excerpts that make up the publisher's

copyrighted compilation of Jane Austen quotes to expound on a different theme, there would be no infringement of any copyright. *See id.* at 349 ("Notwithstanding a valid copyright, a subsequent compiler remains free to use the facts contained in another's publication to aid in preparing a competing work, so long as the competing work does not feature the same selection and arrangement."). The second scholar did not copy the publisher's *compilation*, i.e. the publisher's protected expression, but merely selected a few passages from it based on a *different* expression of creativity.

Here, there is no debate that individual words or multiword commands are not protected, *see* Appx1347, and thus they comprise unprotected 'facts' free for all to use. To the extent this Court finds Cisco has a protected compilation, for Cisco to prevail this Court must also find that it is the *compilation* that was infringed, not the underlying facts. *See Feist*, 499 U.S. at 350 ("only the compiler's selection and arrangement may be protected; the raw facts may be copied at will.").

Moreover, if Cisco's "compilation" is a litigation-driven selection that does not represent Cisco's expressive selection and arrangement, Arista could not have infringed it. *See Harper House, Inc. v. Thomas Nelson, Inc.*, 889 F.2d 197, 205 (9th Cir. 1989) ("[a]s with factual compilations, copyright infringement of compilations consisting of largely uncopyrightable elements

should not be found in the absence of 'bodily appropriation of expression'") (citation omitted). Here, Cisco claims as its compilation pieces from 26 separate copyright registrations, and as the district court noted, "it is not clear whether the Cisco CLI, as one compilation, existed prior to litigation and therefore, Cisco has not shown the Cisco CLI as a whole is entitled to copyright protection." *Cisco Sys. Inc. v. Arista Networks, Inc.*, No. 14-cv-05344, 2016 WL 4440239, Order re Summ. J., at *3 (N.D. Cal. Aug. 23, 2016).

Thus, to the extent Cisco is attempting to express its "creativity" through a compilation created only ex post, and the law demands that the expressed creativity exist ex ante, there can be no infringement. *See Apple Computer, Inc. v. Microsoft Corp*., 35 F.3d 1435, 1446 (9th Cir. 1994) (noting that, in the case of alleged infringement of a work as a whole (i.e., a compilation), "there can be no infringement unless the works are virtually identical"); *MiTek Holdings, Inc.*, 89 F.3d at 1558 ("[Compilation] protection is limited, however, and extends only to the work *as a whole*[.]"(emphasis added)).

III. IF THE COURT CONFIRMS COPYRIGHTABILITY, IT SHOULD AFFIRM THE JURY'S FINDING THAT ARISTA'S ACTIVITIES WERE LAWFUL

If the Court continues to treat the works in question as copyrightable, it should nonetheless uphold the jury's verdict. Arista has laid out the numerous sensible bases for the jury's finding (Arista Br. at 32-69), and we will not repeat them here. Rather, EFF will focus on a key underlying issue: the role of

extrinsic factors.

A. The Guiding Question for the Scènes à Faire Analysis Is Whether Extrinsic Factors Dictated the Creators' Choices, Not Whether Alternative Choices Existed

When determining whether copied elements comprise scènes à faire, a key question is whether the choices made by Cisco were a product of creativity as opposed to other factors, not whether Cisco could have used different words or structures. *See Apple Computer, Inc.,* 35 F.3d at 1444 (finding the use of overlapping windows to be scènes à faire despite existence of alternative method of using tiled windows); *see also Bikram's Yoga*, 803 F.3d at 1042. For example, a photographer could take a photo of the Manhattan Bridge as opposed to the Brooklyn Bridge, making the choice to use one bridge over another. The fact that two bridges exist, by itself, does not make the choice to photograph one bridge over the other anything other than pedestrian. *Cf. Feist*, 499 U.S. at 363 (finding "there is nothing remotely creative about arranging names alphabetically in a white pages directory," despite possibility of ordering entries in other ways).

Similarly, just because Cisco chose to use the word "ip" over, say, "ipv4," Cisco Br. at 12, does not mean that using Cisco's compilation is not scènes à faire. The existence of synonyms is not sufficient to show creativity in choices or that the choice was not dictated by extrinsic factors. The photographer above

could not later claim infringement of her photo by a person who, after seeing the photo, similarly chose to photograph the Manhattan Bridge instead of the Brooklyn Bridge but otherwise did not copy any creative element. *See Landsberg v. Scrabble Crossword Game Players, Inc.*, 736 F.2d 485, 489 (9th Cir. 1984) ("a second author does not infringe even if he reproduces verbatim the first author's expression, if that expression constitutes stock scenes or scenes that flow necessarily from common unprotectable ideas"). Photographing a bridge (whether the Manhattan bridge or otherwise) is scènes à faire; the fact that the two photos contain the same bridge, even where the second photographer's choice to photograph that bridge is based on the choice by the first photographer, is not sufficient to find infringement.

B. A Jury Could Reasonably Find that Extrinsic Factors Dictated Arista's Choices

Cisco's design choices were constrained by industry standards, customer demands, and other functional considerations. The jury heard more than sufficient evidence to find that the scènes à faire doctrine protects Arista's use of whatever portion of the Cisco works the jury found to be copied. *See* Arista Br. at 37-51.

For example:

- Each multiword command-line expression corresponds to a specific

function implemented in the device. As the court noted, Arista's expert explained at trial that those functional features dictated the content of the compilation of commands. Appx9 (citing Appx12126 and Appx12255-12258).

- The arrangement of the commands followed an industry-standard format not original to Cisco, whereby each command took the form of "[verb] [object or entity] [additional parameters]." Appx1337. Cisco disavowed ownership of this syntax. *Id.*
- The supposedly creative "arrangement" of command hierarchies is identical to the command syntax that Cisco admitted to be unprotected. For instance, the hierarchy of the "show" command arranges multiword expressions with the verb "show" at the first level, the objects or entities "arp," "clock," and "environment" at the second level, and command parameters "all," "power," and "temperature" at the third level. Appx3. Within these levels, the commands are arranged in standard alphabetical order. *Id.*
- In several cases, even the exact selection and arrangement antedated Cisco's implementation in published standards documents, such as the IP protocol version 6 standards defined by the Internet Engineering Task Force (IETF); the "ip gimp"

commands selected and arranged by the Internet Group Management Protocol (IGMP); or the "Open Shortest Past First (OSPF)" Internet standard. Appx10.

- Customer demands reinforced Cisco's need to adopt standard, rather than creative, selections and arrangements of its command-line expressions. At trial, witnesses testified that the need for backward-compatibility—the ability to use the same command hierarchy as in pre-existing systems—dictated Cisco's selection and arrangement because customers desired this consistency. Appx12.
- Similarly, one of Cisco's engineers explained at trial that commands needed to be selected and arranged according to logical and predictable principles so that they would make sense to customers. *Id.* Selecting and arranging commands according to arbitrary aesthetics rather than standard practices would undermine this goal.

The selection and arrangement of Cisco's handful of "modes and prompts" are also standard and subject to the scènes à faire doctrine. The four "modes" are simply four escalating levels of authority that a user has to execute commands in a system, arranged from "least authority" to "most authority." Appx1329-1330, Appx1340-1341. Arranging four modes in ascending order is

hardly a creative, aesthetic choice, but rather a standard and practical one. The use of ">"and "#" have also been common since well before Cisco's creation of its interfaces, and the two other prompts are simply descriptive of the mode to which they correspond, with "(config)#" denoting "Global Configuration" mode using the common abbreviation of "config," and "(config-if)" adding the common abbreviation "if" in place of Interface for "Interface Configuration" mode. Appx1340-1341 (citing Black Rpt.'s discussion of pre-existing TOPS-20, UNIX, SUMEX, and MS-DOS modes and prompts).

Likewise, the selection and arrangement of help descriptions and command responses and outputs necessarily had to track the selection and arrangement of the commands themselves. Since the compilation of commands was standard and conventional, so too was the compilation of corresponding help descriptions.

The scènes à faire doctrine bars Cisco from controlling the use of standard and conventional compilations of multiword command-line expressions, "modes and prompts," or help descriptions. The jury had ample evidence to find that the doctrine applied to protect Arista's use of any or all of these elements.

IV. UPHOLDING THE JURY FINDING PROTECTS COMPETITION AND INNOVATION

A. Where Works at Issue Are Highly Functional, Courts Should Be Particularly Deferential to a Jury Finding of Scènes à Faire

Contrary to the protestations of Cisco's amici, Mathworks Br. at 14-16, 27-28, the jury's scènes à faire determination, and the district court's own judgment, reflect the crucial role the doctrine must play in protecting competition and innovation.

As the Ninth Circuit has explained, allowing a copyright owner to effectively control functionality without satisfying the requirements of patentability flies in the face of Congress's proscription against the control of ideas and facts via copyright. *Sony Computer Ent'mt, Inc. v. Connectix Corp.*, 203 F.3d 596, 605 (9th Cir. 2000). As Judge Boudin recognized in *Lotus*, that problem is especially acute where the work is a computer program; such program are fundamentally and necessarily useful and while "[u]tility does not bar copyright . . . it alters the calculus." 49 F.3d at 819 (Boudin, J., concurring). For one thing, the value and widespread adoption of a computer program is likely to derive not from how creative it is, but from how useful it is:

> A new menu may be a creative work, but over time its importance may come to reside more in the investment that has been made by *users* in learning the menu and in building their own mini-programs—macros—in reliance upon the menu. Better typewriter keyboard board layouts may exist, but the familiar QWERTY keyboard dominates the market because that is what everyone has

learned to use.

Id. at 819-20 (citation omitted; emphasis in original). Similarly, when a computer program has a selection of functional features (in *Lotus*, the spreadsheet commands; here, Cisco's CLI commands), the program's users who learn those features become "locked in" to the "choices" made by those programs. Simply put, the more useful the functionality is, the more the users invest their own time to learn the way the functionality is written and organized.

As a result, overly broad copyright protections for such works may give rightsholders far greater power than Congress intended. Returning to Judge Boudin's concurrence in *Lotus*, "if a better spreadsheet comes along, it is hard to see why customers who have learned the Lotus menu and devised macros for it should remain captives of Lotus because of an investment in learning made by the users and not by Lotus." 49 F.3d at 821.

In this context, the best way to ensure that copyright does not impede innovation is to respect copyright's limits, including the scènes à faire doctrine. The doctrine helps implement the idea/expression dichotomy, one of the "traditional contours" of copyright. *See, e.g.*, *Lexmark Int'l, Inc. v. Static Control Components, Inc.*, 387 F.3d 522, 535 (6th Cir. 2004) (courts look to "two [] staples of copyright law—the doctrines of merger and scènes à faire" in ascertaining the boundary between idea and expression).

That role has been particularly useful with respect to software, where the line between process and function is particularly "elusive." *Id.*; *see also Sega Enters. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1524 (9th Cir. 1992) ("Computer programs pose unique problems for the application of the 'idea/expression' distinction[.]"). As the Ninth Circuit explained in *Sega*, courts should seek to avoid a result that "defeats the fundamental purpose of the Copyright Act—to encourage the production of original works by protecting the expressive elements of those works while leaving the ideas, fact, and functional concepts in the public domain for others to build on." *Id.* at 1527. Where, as here, the work is highly functional, and there are multiple bases for a jury finding that the scènes à faire defense applies, courts should hesitate to disturb that finding.

B. If Cisco's Approach Had Been the Industry Standard, Copyright Claims Could Have Stifled the Emergence of Key Technologies

Cisco's amici suggest that the software industries will collapse without strong copyright protection. Mathworks Br. at 11-13. To the contrary, the past three decades suggest that limits on copyright have helped foster competition and innovation.

1. The BIOS of the Original IBM-Compatible PC

In 1981, IBM released its first home computer, the PC. Charles H. Ferguson & Charles R. Morris, *Computer Wars: The Fall of IBM and the Future*

of Global Technology 27 (1994). Unlike prior offerings, the IBM PC had an open design. Thanks to that design, add-on innovation in PC software and hardware peripherals flourished. *Id*. at 28-29. To use IBM-exclusive software like the popular spreadsheet program Lotus 1-2-3, however, users initially had to buy IBM computers. *Id*. Although other manufacturers could run the same MS-DOS operating system that IBM used, many best-selling programs required complete hardware and Basic Input/Output System (BIOS) firmware[9] compatibility as well. Thus, the IBM model was the de facto standard. *Id*. at 51-53.

In order to create a computer that could compete with the IBM PC, other manufacturers needed to duplicate the functionality of IBM's BIOS firmware. *See id.* at 52-53. To avoid exposing themselves to copyright liability, Phoenix, Compaq, and other hardware manufacturers assembled two "teams." *Id.*; Van Lindberg, *Intellectual Property and Open Source: A Practical Guide to Protecting Code* 240-41 (2008). The first "team" analyzed the IBM BIOS and wrote functional specifications about the software's structure, sequence, and organization. Matthew Schwartz, *Reverse-Engineering*, Computerworld

[9] Firmware is software stored in read-only memory that stays intact even when a computer is switched off. Microsoft Press Computer Dictionary 167 (2d Ed. 1994). Firmware holds basic pieces of software in a computer, like startup routines and the interface that allows the operating system to interact with the computer hardware. *See generally* Jeff Tyson, How BIOS Works, HowStuffWorks, http://computer.howstuffworks.com/bios1.htm.

(November 12, 2001).[10] These functional specifications were passed to the "clean" teams of programmers who had never seen the BIOS source code. Van Lindberg, *supra*, at 240-41. The clean teams created new software from scratch using the interface specifications needed to interact successfully with the IBM PCs: the BIOS interface, including its structure, sequence, and organization. *Id.*

Once these firms developed their own BIOS firmware, they were able to produce cheaper, faster IBM-compatible computers, and market innovations like the first portable PC. Ferguson, *supra*, at 53-55; *see also* Mark Hall, *Compaq Computer Corporation: Portable Computer*, Encyclopædia Britannica.[11] With more computers and customers now available to them, software developers began to write and distribute more software than ever, innovating with new features and functionality and competing directly on price. The age of home computing began in earnest.

Key to that development was the fact that IBM owned the copyright on the BIOS source code, but could not claim a monopoly on the collection of commands used to communicate with that code. Thus, Compaq and Phoenix were able to reimplement the BIOS interface as long as they did not copy any of IBM's code. If the law had been otherwise, IBM's rights in the BIOS code

[10] Available at http://www.computerworld.com/article/2585652/app-development/reverse-engineering.html.
[11] Available at https://www.britannica.com/topic/Compaq-Computer-Corporation.

would have given it the ability to stifle competitive innovation, to the detriment of the public.

2. Major Modern Operating Systems' Reimplementation of the Groundbreaking UNIX Interface

Many popular operating systems today reimplement the command-line interface of one of the earliest operating systems, UNIX. Developed at AT&T Bell Labs and launched in 1969, UNIX is widely regarded as the first modern operating system. Heather J. Meeker, *The Open Source Alternative* 3-4 (2008). It ran on large mainframe and minicomputers owned by corporations, universities, and the government. *Id.*

When AT&T developed UNIX, however, the company was operating under a consent decree that forbade it from monetizing any project outside of telecommunications and special federal contracts. *Milestones in AT&T History*, ATT.com.[12] To comply with the decree, AT&T licensed UNIX source code to any interested party for a nominal fee. Meeker, *supra*, at 5. Thanks in part to that open license, computer scientists embraced UNIX, making it the dominant operating system of its day. *Id.* Programmers shared their source code and programming innovations freely, developing and releasing new versions of the operating system. *Id.*

The original versions of UNIX became obsolete as the computers that ran

[12] Available at http://www.corp.att.com/history/milestones.html.

them changed, but the UNIX platform could always return in new forms because AT&T's copyright in the UNIX code didn't bar new implementations of the same sets of commands. Software developers dissatisfied with available operating systems such as MS-DOS, Windows, and Apple's system, along with UNIX users, reimplemented the UNIX interface to run on a PC.[13] Meeker, *supra*, at 6.

Because the collection of commands could be reimplemented, it took a minimal amount of work to make pre-existing software run on subsequent systems. For example, some developers wanted to create a new operating system that would run software made for UNIX, but was also free of AT&T's (or anyone's) intellectual property, specifically a system comprising only free software. *Id.* at 6-7. The GNU project, together with the Finnish programmer Linus Torvalds, produced the Linux operating system, which shares the UNIX command interface, but uses entirely original code. *Id.*

Today, Linux is widely used throughout the computer industry. Tens of millions of servers run Linux. Steven J. Vaughan-Nichols, *How Many Linux Users Are There (Really)?*, Linux Planet (Feb. 18, 2009).[14] 37% of Web servers

[13] MS-DOS itself reimplemented the programming interface of an earlier operating system, CP/M. *See Paterson v. Little, Brown & Co.*, 502 F. Supp. 2d 1124, 1128 (W.D. Wash. 2007).

[14] Available at http://www.linuxplanet.com/linuxplanet/reports/6671/1.

run on Linux.[15] As of November 1, 2017 Linux was used by all 500 of the 500 fastest supercomputers.[16] As of November 2017, Android (which uses the Linux kernel) had a 68.85% total share of mobile operating systems.[17] Countless Internet-based services from Facebook to ATMs rely on Linux-based high-speed networking systems. Vaughan-Nichols, *supra*.

The varied implementations of UNIX are textbook examples of the importance of limits on the scope of copyright to innovation and competition. If programmers had been forced to go to the expense and complexity of acquiring a license, much of this innovation would never have happened.

3. The C Programming Language

Thanks to standard collections of commands, software written in one programming language can be run on any operating system.

The evolution of "C" is a textbook example. Dennis Ritchie, one of the computer scientists who invented UNIX, also invented a new language, called

---

[15] *See Usage of operating systems for websites*, W3Techs, http://w3techs.com/technologies/overview/operating_system/all; *Usage statistics and market share of Unix for websites*, W3Techs, *available at* https://w3techs.com/technologies/details/os-linux/all/all.

[16] *See Operating System Development Over Time*, TOP500 Supercomputer List, http://www.top500.org/statistics/overtime/, which collects data on the 500 most powerful commercially available computer systems (select "Operating System Family" and "Systems Share," then click on "Submit.").

[17] Netmarketshare, Mobile Operating System Market Share (Nov. 2017), http://marketshare.hitslink.com/operating-system-market-share.aspx?qprid=8&qpcustomd=1&qptimeframe=M.

"C," in which to code it. Brian W. Kernighan & Dennis M. Richie, *The C Programming Language* ix (1978). Programs written in C use the C Standard Library to execute their functions and operate the computer on which they run—including tasks as basic as opening and closing files. Once programmers learn C, they can write code that will run on any operating system that can provide a reimplementation of the C standard library.

Today, those operating systems are legion. The C Standard Library has been reimplemented countless times to allow different operating systems to work with programs written in C. For example, Microsoft reimplemented the C Standard Library for Windows as part of the Microsoft C Run-Time Library. *CRT Library Features*, Microsoft Developer Network.[18] Google's reimplementation of the same for Android is called Bionic. *The Native Android API*, Mobile Pearls.[19] Another significant reimplementation was the GNU C Library, which was essential to the GNU Project's effort to create a free UNIX-compatible operating system. *The GNU C Library (glibc)*, The GNU Project.[20]

Limiting the ability to reimplement the C Standard Library would have severely limited the range of systems on which C programs could run. Each operating system would require a new, incompatible version of the language,

[18] Available at https://docs.microsoft.com/en-us/cpp/c-runtime-library/crt-library-features.
[19] Available at http://mobilepearls.com/labs/native-android-api/.
[20] Available at http://www.gnu.org/software/libc/.

potentially restricted to a different set of functionality. Many innovative software projects would be restricted to a single operating system, or simply never get off the ground. Old programs could become obsolete whenever a new operating system came into use, and new operating systems would be unable to take advantage of the thousands of existing C programs.

4. Industry Standards for Cloud Computing

Modern cloud computing providers, like Amazon Web Services, rely on a standard collection of commands and functionality derived from one of the oldest computer operating systems: the IBM PC BIOS. Cloud computing allows users to rent space and processing power on distant servers, accessible from anywhere in the world via the Internet. *What is Cloud Computing?*, Amazon Web Services.[21] At their core, cloud computing clusters act as "virtual machines"—imitations of small computers being run on huge servers. *See id.*; *see also* Margaret Rouse, *Definition: Virtual Machine (VM)*, SearchServerVirtualization (Oct. 2011).[22] Virtual machines send commands to invoke the functions of the BIOS just like physical computers, but they have no individual physical hardware. *See id.* Instead, a reimplementation of the library of commands from BIOS allows the server to execute the instructions of all the

[21] Available at http://aws.amazon.com/what-is-aws.
[22] Available at http://searchservervirtualization.techtarget.com/definition/virtual-machine.

virtual machines running on it. *See id.*

Cloud computing providers use a similar functional interface to govern how their users can interact with their services. *Amazon Web Services (AWS) and Eucalyptus Partner to Bring Additional Compatibility Between AWS and On-Premises IT Environments*, Eucalyptus (Mar. 22, 2012).[23] Because copyright does not restrict use of the set of command functions, companies like CloudStack and Eucalyptus can compete with Amazon to provide the best implementation of it. Businesses that employ cloud services can also write or commission their own proprietary software to perform operations on cloud servers. *See Business Applications*, Amazon Web Services.[24] In addition, since major cloud service providers like Amazon, Eucalyptus, and CloudStack use the same set of commands to invoke the same functions, their customers can easily switch from one cloud service to another. Steven J. Vaughan-Nichols, *OpenStack vs. CloudStack: The Beginning of the Open-Source Cloud Wars*, ZDNet (Apr. 12, 2012).[25] Software developers can write programs capable of interacting with the above three cloud services, creating new ways for users to access and manipulate information spread out across the Internet.

---

[23] Available at http://www.itbriefcase.net/amazon-web-services-aws-and-eucalyptus-partner.
[24] Available at http://aws.amazon.com/business-applications.
[25] Available at https://web.archive.org/web/20161116135027/http://www.zdnet.com:80/article/openstack-vs-cloudstack-the-beginning-of-the-open-source-cloud-wars/.

By contrast, if Amazon could use copyright to monopolize the functional aspects of its cloud storage interface, Amazon would be able to lock in its users and cripple new competitors. Because businesses use custom software built around the cloud service provider's standards for command inputs, switching to a cloud service provider with a different one would require rewriting their cloud software. Given the cost and disruption of doing so, few businesses would be willing to leave their cloud service provider, meaning late entrants in the cloud service market would be hard-pressed to build a customer base. The ultimate result: less choice, less innovation.

5. Creation of Software That Otherwise Would Not Be Written

When programmers can freely write their own code to implement the functionality of a library of commands, they can create compatible software that the interface's original creator might never have envisioned or had the resources to develop, and they can create competitive devices with the full range of functionality called for by the collection of device commands. Copyright in the library of functionality a product offers would discourage this innovation by creating potential liability for the mere act of creating a competitive or compatible product.

One straightforward and common reason to reimplement another programmer's command library is to make a program compatible with a

different application or platform. Small companies and volunteer groups often undertake such projects, but heavy licensing fees or the threat of copyright litigation would hinder this work.

For example, Wine is a service that interprets Windows command expressions so Windows programs can run on UNIX-based operating systems like Linux and Mac OS X. *About Wine*, WineHQ .[26] Millions of people use Wine to make their favorite Windows programs work on other operating systems. *Id.* Microsoft has no agreement and no contact with the Wine project. Scott Swigart & Sean Campbell, *Interview with Alexandre Julliard, Head of the Wine Project/CTO of CodeWeavers*, How Software is Built (Sept. 8, 2008).[27] In fact, Microsoft has interfered with Wine users' ability to update their software. Ingrid Marson, *Microsoft Admits Targeting Wine Users,* ZDnet (Feb. 25, 2005).[28] If Microsoft could monopolize the compilation of Windows input commands, Microsoft could demand licensing fees from Wine, or shut the project down, preventing its users from running software they have legally purchased or licensed on their own computers.

---

[26] Available at http://www.winehq.org/about/.
[27] Available at http://web.archive.org/web/20130719141118/http://howsoftwareisbuilt.com/2008/09/09/interview-with-Alexandre-Julliard-Head-of-the-Wine-Project-CTO-of-CodeWeavers/.
[28] Available at https://web.archive.org/web/20140313064317/http://www.zdnet.com/microsoft-admits-targeting-wine-users-3039189180/.

CONCLUSION

The Court should find the works in question uncopyrightable. If not, then it should still affirm the judgment below under the scènes à faire doctrine.

December 26, 2017 Respectfully submitted,

ELECTRONIC FRONTIER FOUNDATION

By: /s/ Michael Barclay
Michael Barclay
(*Principal Attorney of Record*)
Vera Ranieri
ELECTRONIC FRONTIER FOUNDATION
815 Eddy Street
San Francisco, CA 94109-7701
Tel: (415) 436-9333
Fax: (415) 436-9993
michael@eff.org

Attorneys for Amicus Curiae
Electronic Frontier Foundation

CERTIFICATE OF SERVICE

I hereby certify that on December 26, 2017, I caused the foregoing BRIEF OF AMICUS CURIAE ELECTRONIC FRONTIER FOUNDATION IN SUPPORT OF DEFENDANT-APPELLEE AND AFFIRMANCE to be served by electronic means via the Court's CM/ECF system on all counsel registered to receive electronic notices.

/s/ Michael Barclay
Michael Barclay

ELECTRONIC FRONTIER FOUNDATION
815 Eddy Street
San Francisco, CA 94109-7701
Tel: (415) 436-9333
Fax: (415) 436-9993
michael@eff.org

CERTIFICATE OF COMPLIANCE WITH TYPE-VOLUME LIMITATION, TYPEFACE REQUIREMENTS, AND TYPE STYLE REQUIREMENTS PURSUANT TO FED. R. APP. P. 32(A)(7)(C)

I hereby certify as follows:

1. The foregoing Brief of Amicus Curiae Electronic Frontier Foundation In Support of Defendant-Cross Appellant and Affirmance complies with the type-volume limitation of Fed. R. App. P. 32(a)(7)(B) and Fed. Cir. R. 32(a). The brief is printed in proportionally spaced 14-point type, and there are 6,888 words in the brief according to the word count of the word-processing system used to prepare the brief (excluding the parts of the brief exempted by Fed. R. App. P. 32(f) and by Fed. Cir. R. 32(b)).

2. The brief complies with the typeface requirements of Federal Rule of Appellate Procedure 32(a)(5), and with the type style requirements of Federal Rule of Appellate Procedure 32(a)(6). The brief has been prepared in a proportionally spaced typeface using Microsoft® Word for Mac 2011 in 14-point Times New Roman font.

December 26, 2017

/s/ Michael Barclay
Michael Barclay
Attorneys for Amicus Curiae
Electronic Frontier Foundation