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. 5:14-CV-05344-BLF, Judge Beth Labson Freeman

NON-CONFIDENTIAL BRIEF FOR DEFENDANT-APPELLEE ARISTA NETWORKS, INC.

ROBERT A. VAN NEST
BRIAN L. FERRALL
DAVID J. SILBERT
STEVEN A. HIRSCH
MICHAEL KWUN
RYAN WONG
ELIZABETH K. MCCLOSKEY
KEKER & VAN NEST LLP
633 Battery Street
San Francisco, CA 94111-1809
Telephone: 415 391 5400
Facsimile: 415 397 7188

Attorneys for Defendant-Appellee ARISTA NETWORKS, INC.

DECEMBER 22, 2017

COUNSEL PRESS, LLC (888) 277-3259

UNITED STATES COURT OF APPEALS FOR THE FEDERAL CIRCUIT

CISCO SYSTEMS, INC. **v.** ARISTA NETWORKS, INC.

Case No. 2017-2145

CERTIFICATE OF INTEREST

Counsel for the:
☐ (petitioner) ☐ (appellant) ☐ (respondent) ☒ (appellee) ☐ (amicus) ☐ (name of party)

Arista Networks Inc.

certifies the following (use "None" if applicable; use extra sheets if necessary):

1. Full Name of Party Represented by me	2. Name of Real Party in interest (Please only include any real party in interest NOT identified in Question 3) represented by me is:	3. Parent corporations and publicly held Companies that own 10 % or more of stock in the party
Arista Networks Inc.	N/A	None

4. The names of all law firms and the partners or associates that appeared for the party or amicus now represented by me in the trial court or agency or are expected to appear in this court (**and who have not or will not enter an appearance in this case**) are:

Keker, Van Nest & Peters: Robert Van Nest, Brian Ferrall, David Silbeti, Steven Hirsch, Michael Kwun, Ajay Krishnan, Ryan Wong, Elizabeth McCloskey, David Rosen, Eduardo Santacana, Andrea Nill-Sanchez, Ashok Ramani, Katherine Lloyd-Lovett, Nick Marais

Fish & Richardson: Juanita Brooks, Kelly Hunsaker, Lauren Degnan, Michael McKeon, Ruffin Cordell

Wilson Sonsini Goorich & Rosati: Scott A. Sher, Susan Creighton, Bradley Tennis, Jonathan Jacobson, Chul Pak, David Reichenberg

Tensegrity Law Group: Matthew Powers, William Nelson, Paul Elulich, Alex Chan, Wanli Chen,

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. See Fed. Cir. R. 47. 4(a)(5) and 47.5(b).

None

December 22, 2017	/s/ Brian Ferrall
Date	Signature of counsel
Please Note: All questions must be answered	Brian Ferrall
	Printed name of counsel

cc:

TABLE OF CONTENTS

Page

Confidential Material Omitted

Confidential material on page 11 relates to settlement agreement details subject to the protective order filed below.

TABLE OF AUTHORITIES

Pages

Cases

Statutes

Rules

Other Authorities

STATEMENT OF RELATED CASES

Appellee agrees with Appellant's Statement of Related Cases.

I. INTRODUCTION

This appeal concerns the alleged copying of compilations of multiword textual commands included in the "command-line interfaces" (CLIs) found in four operating systems that are used in Cisco's[1] Ethernet switches.[2]

These commands—known as "nerd knobs" in the trade—are like the controls on the front of an old-style stereo receiver. They enable a technician to control the features built into the switch—just as the volume, balance, and tone controls of a stereo receiver enabled a user to control the features of the receiver.

Arista is accused of using 506 industry-standard commands from Cisco's CLIs—a tiny fraction of the roughly 16,000 commands found in just *one* of those CLIs.[3] Cisco doesn't allege that Arista copied any Cisco

---

[1] Throughout this brief, "Cisco" refers to plaintiff-appellant Cisco Systems, Inc.; "Arista" refers to defendant-appellee Arista Networks, Inc.; "CB" refers notionally to "Cisco Brief" or literally to the "Brief of Plaintiff-Appellant"; "Appx" refers to the appellate Appendix; emphases in quotations were added unless otherwise stated; and internal punctuation and citations were omitted from quotations.

[2] Other asserted compilations are not discussed here for reasons explained below at Part III.F.

[3] Appx12128-12129,Appx12189–12190(Black);Appx62649 (TX7543,slide14).

source code to implement Arista's interfaces or any other aspect of Arista's switches; and Arista didn't do so. Instead, Arista wrote 100% of its operating-system source code from scratch or based on open-source software.

After a two-week trial featuring 32 witnesses and some 256 admitted exhibits, the jury—applying instructions that Cisco does *not* challenge on appeal—found *some compilation* to be original to Cisco. And it found that Arista copied *some portion* of that original compilation. But the jury also found that whatever portion of whatever compilation it had found to have been copied was scènes à faire. Throughout this process, the jury knew that many elements of Cisco's user interfaces weren't protected by copyright and couldn't form the basis for an infringement finding—because the instructions said so.

On appeal, the question is whether substantial evidence supported the jury's determination. But because the verdict form doesn't precisely delineate what the jury determined along the way—*what* compilation it found original, *what portion* of that compilation it found to have been copied, or *how* it determined that the copied material was scènes à

faire—the question really is *whether substantial evidence supports* ***any*** *reasonable path that the jury may have taken to reach its verdict.*

As discussed below, the jury reasonably could have found that once it followed the district court's instructions, there was almost nothing left that possibly could be original. The jury nonetheless must have found *something* original, but perhaps only a small sliver of a something—a tiny kernel of originality within Cisco's selection of multiword commands. And although Arista was accused of copying only 506 out of many thousands of commands, perhaps the jury also found that Arista had copied enough of Cisco's selection of commands to incur liability (i.e., a "non-de minimis" amount).

But because the jury reasonably could have found that any copying of original material was very minor—perhaps just barely enough to rise to the level of infringement—it also reasonably could have found that this small amount of copying was excused by the scènes à faire doctrine. Under the instructions, Arista prevailed on that defense if it proved that the copying was dictated by concerns external to any creativity that the jury had found in the copied material. If the jury found minimal originality—as appears likely—then finding the

copied material to be scènes à faire would have been correspondingly easy. For example, the jury reasonably could have found that Cisco's selection of commands was dictated by its choices of which features to include in its switches. Or the jury reasonably could have found that Cisco's selection of commands was dictated by the industry standards that governed the networking protocols supported by the switches.

In sum: If the jury found that very little was original and copied, it would have been a simple matter for it also to find that Arista had proved its defense of scènes à faire. Because the verdict form tells us so little about the jury's thought process—and also because Cisco failed to ever define, submit adequate evidence of, or even separately register its compilations—Cisco carries an especially heavy burden when trying to demonstrate that no reasonable jury could have found as this one did. Cisco has not met that burden. *See* Part V.B., below. Accordingly, the scènes à faire verdict should be affirmed. Alternatively, the judgment may be affirmed on four alternative grounds discussed below at Part V.D.

II. ISSUES PRESENTED

1. The jury found that Arista had infringed Cisco's copyrighted operating-system user interfaces by copying the selection and arrangement of at least one compilation of elements from those interfaces. The jury also found that Arista had established a complete scènes à faire defense to that infringement. Was the jury's scènes à faire verdict supported by substantial evidence?

2. Did Cisco waive its argument that there can be no scènes à faire defense where the copying standard is "virtual identity"; and if not waived, is the argument irrelevant given that the jury likely didn't apply the virtual-identity standard?

3. As alternative grounds of affirmance, should the district court have granted Arista JMOL of non-infringement because:

- Cisco's CLI is a method of operation excluded from copyright protection by 17 U.S.C. § 102(b);
- Cisco failed to place the allegedly infringed and infringing works in evidence so the jury could compare them;

- Cisco failed to prove that its "user interfaces" were separable from its operating systems, as required for them to be independent copyrighted works; or
- Cisco failed to prove infringement at the compilation level?

III. STATEMENT OF THE CASE

A. Cisco established dominance in network hardware while encouraging both customers and competitors to use its "industry-standard CLI."

Founded in 1984, Cisco has long been the dominant company in the network-switch field.[4] Cisco once held 99% of the market[5] but its market share has slipped to around 80% in recent years.[6]

Every Cisco switch comes with one of Cisco's four operating systems. The original one was called the Internetwork Operating System ("IOS"); but Cisco has since introduced its IOS XE, IOS XR, and

[4] A switch is a hardware device that allows computers in a local network to communicate with each other. Appx10455(Bakan); Appx11047(Kathail). A router connects multiple networks. Appx10456(Bakan). We use "switch" to refer to both types of device.

[5] Appx11863–11864(Li).

[6] Appx10468(Bakan).

NX-OS operating systems. Each operating system includes some version of Cisco's command-line interface ("CLI").[7]

A CLI is a method of operating the features of an Ethernet switch by typing in a textual command and then hitting enter or return.[8] Because they're entered by a user, CLI commands generally are not displayed in its user interface. When Cisco says that Arista copied its CLI commands, it means that Arista wrote original or in-licensed software that responds to some of the same commands that Cisco's software does when a user types those commands.[9]

CLI commands have three basic purposes: to configure the switch (i.e., instruct it on how it should behave in the network); to display

---

[7] Appx10866–10867(Duda);Appx10905(Sadana);Appx10460(Bakan).

[8] Appx12090–12091(Black). Throughout this brief, "product features" and "features" refer to switch functionality that lies outside of, and is controlled by, commands within the asserted interfaces—e.g., the ability to support a particular function of a networking protocol. The jury instructions, by contrast, twice refer to allegedly copied "features" *within* the interfaces—a different reference altogether. *Cf.* Appx1396–1397,Appx1406. We will call these "CLI features."

[9] Appx12090–12091(Black). Although they are generally not contained in a switch's software or displayed in its user interface, CLI commands sometimes are set forth in manuals and other documentation. But the jury below found no illegal copying of Cisco's documentation—a finding that Cisco has not challenged on appeal. Appx1427–1431.

device status; and, through so-called "exec commands," to take an immediate action such as rebooting or shutting down the switch or upgrading its software.[10]

As Arista's expert explained at trial, CLI commands are sometimes called "nerd knobs" because they're analogous to "an old style stereo where you've got an on-off button and volume and tuner knobs and so forth. In essence, when you are using CLI, you are pushing buttons and turning knobs."[11] Each CLI command relates to one of the functions in the switch.[12] Just as a stereo's features dictated the kinds of knobs on its control panel, the features of a network switch dictate the kinds of commands in its CLI.[13]

Between 1993 and 2000, as Cisco achieved market dominance, its CLI became the industry standard.[14] Cisco saw a clear business advantage in the widespread use of its CLI. Cisco's marketing pitches

---

[10] Appx10874–10875(Duda);Appx11249(Almeroth).

[11] Appx12091(Black).

[12] Appx11086(Kathail).

[13] *See* Parts V.B.2. & 3., below, for the evidence on this point.

[14] Appx12038–12039(Volpi);Appx52862(TX5134,p.3).

boasted that its switches used the "industry-standard" CLI.[15] Cisco knew that its customers didn't want to be locked into one vendor and would feel more comfortable buying Cisco products if those products interoperated with those of other vendors.[16] Cisco also knew that its customers favored industry-standard commands because their technicians knew them and preferred not to learn new ones each time they bought a new switch.[17] And Cisco executives and engineers didn't believe that the CLI was proprietary or protectable.[18] Neither did the vendors who copied them.[19]

---

[15] Appx62888–62889(TX8110);Appx10471(Bakan);Appx53312(TX5299, p.2);Appx10473–10474(Bakan);Appx11970(Ullal); Appx12038–12044 (Volpi);Appx53749;Appx53653–53800 (TX5457);Appx53645(TX5451); Appx62905(TX8237,p.2).

[16] Appx10994–10995(Giancarlo);Appx11969–11970(Ullal);Appx12265; Appx63484,Appx63490(TX9079,pp.82–84,206(Gourlay)).

[17]Appx10608–10610(Lougheed);Appx53802(TX5464);Appx62795 (TX7996);Appx10668–10670(Remaker);Appx12265;Appx63483, Appx63487(TX9079,pp.73–74,124(Gourlay)).

[18] Appx10990,Appx10094–10095,10998–11000(Giancarlo); Appx11035(Dale); Appx12265;Appx63484(TX9079,pp.82–84 (Gourlay));Appx12326; Appx63499(TX9081,p.71(Cato)). We agree that they aren't. *See* Part V.D.1., below.

[19] Appx11863–11864(Li);Appx12326;Appx63498–63502(TX9081, pp.64,69,71,116–122(Cato)).

Cisco therefore declined to go after a long list of competitors, including IBM, BLADE Network Technologies, Foundry Networks, Xtreme Networks, Hewlett-Packard, Procket Networks, Dell, Juniper Networks, and Nortel,[20] that openly marketed their products as having a "Cisco-like" or "industry-standard" CLI.[21] Dell, for example, uses over 1,400 CLI commands that are identical to Cisco's—nearly three times as many as Arista is alleged to use—without any objection from Cisco.[22] Indeed, before this lawsuit, Cisco never threatened or took legal action against any vendor that copied its CLI[23]—with one exception.

---

[20] Appx11093–11094(Kathail);Appx53611–53633 (TX5441;Appx11183–11184(Lang);Appx11860–11864(Li);Appx11970–11971(Ullal)(re: Junos-E);Appx12073–12074(Schafer)(re:Junos-E);Appx12088–12089,12134–12146(Black);Appx63400–63409(TX9041);Appx53866–53872(TX5630);Appx53884–53887(TX5635);Appx53888–53890(TX5637);Appx12265;Appx63484(TX9079,p.75(Gourlay)).

[21] Appx10811–10815(Duda);Appx62707 (TX7748);Appx12088–12089, 12134–12146(Black);Appx63400–63409(TX9041);Appx53866–53872 (TX5630);Appx53884–53887(TX5635);Appx53888–53890(TX5637).

[22] Appx12141–12142(Black).

[23] Appx10990–10091,Appx10994–10995(Giancarlo);Appx11091–11094 (Kathail);Appx11183–11184(Lang);Appx11863–11864(Li); Appx12326,Appx63498–63499(TX9081,p.69–71(Cato)). The jury nevertheless rejected Arista's copyright-abandonment defense. Appx1429.

In 2003, Ci sco learned that a Chinese competitor, Huawei, was selling knockoffs of its switches that featured bug-compatible[24] copies of the IOS CLI and most of its implementing source code.[25] After extensive negotiations,[26] Cisco and Huawei entered into a confidential settlement agreement in which Cisco [redacted]—a degree of overlap at least *three times* that alleged here.[27]

B. Arista invented a better switch for cloud computing.

In 2004, Arista was founded by a small group of engineers led by Silicon Valley legend Andy Bechtolsheim.[28] They soon focused on developing a new type of network switch designed from the ground up to serve the cloud computing data-center market.[29]

---

[24] A "bug-compatible" copy is one that even includes the same bugs found in the copied software. Appx10990–10991(Giancarlo).

[25] Appx10992–10993(Giancarlo);Appx11169,Appx11177–11178(Lang); Appx53497(TX5345).

[26] Appx10992–10993(Giancarlo).

[27] Appx11189–11192(Lang);Appx50785–50786(TX4672,pp.10–11); Appx12130(Black);Appx12128–12129(Black).

[28] Appx107909–10710(Remaker);Appx52990–52991(TX5157); Appx11940(Ullal).

[29] Appx11783–11788(Duda).

A modern data center is vastly different from the "network closet" where Cisco's switches dominated. The size of four football fields,[30] it contains rack after rack and row after row of thousands of servers. Atop each rack sits a switch connected to each server in the rack; and each switch is linked to an aggregation switch that is linked in turn to another rack or data center.[31] No human being can manually configure all those devices using an ordinary CLI—the task must be automated.[32]

These data centers needed a new ultra-high-performance switch that could restore a failed server to operation in sub-seconds, not minutes; move data packets at unprecedented speeds; let users modify the operating system to obtain better control over network traffic; provide high "East-West bandwidth" from one server to another; and automate switch controls.[33]

---

[30] Appx12500(Sadana).

[31] Appx11790–11791(Duda).

[32] Appx12500(Sadana);Appx11789–11790(Duda).

[33] Appx11926–119277(Ullal);Appx11788–11789(Duda).

Developing that switch took Arista nearly five years.[34] Besides their blazing speed,[35] Arista products embody a slew of innovations that make them especially attractive to cloud-computing companies.

- **State-sharing architecture** can shut the switch down when it malfunctions and then rapidly bring it back online after recovering its state from the system database.[36]
- A **Linux-based Extensible Operating System ("EOS")** enables cloud customers to add their own software to the product so they can better control how the switch routs network traffic.[37]
- **Zero Touch Provisioning ("ZTP")** simplifies switch installation in large data centers.[38]

---

[34] Appx11787–11788(Duda).

[35] Appx12265;Appx63485–63486,Appx63488–63489(TX9079,pp.94–96,200(Gourlay));Appx11928,Appx11932–11933(Ullal).

[36] Appx11791–11795,Appx11809–11810(Duda);Appx12504–12505(Sadana).

[37] Appx11790,11795–11800,Appx11828–11833(Holbrook);Appx11929–11931(Ullal).

[38] Appx11835–11836(Duda);Appx12121–12122,Appx12124(Black); Appx62629(TX7408,p.5213);Appx.57120–57126(TX6743); Appx12265;Appx63487–63488(TX9079,p.137(Gourlay)).

But there was one domain in which Arista did *not* seek to innovate—the human-facing aspects of the user interface, or CLI.[39] Like so many network-switch vendors, Arista openly marketed its machines as having the "industry-standard CLI." Arista knew that some customers felt locked into Cisco's products—not because of the performance of those products, but because the customers' technicians were familiar with Cisco's CLI commands.[40]

At the same time, Arista maintained strict rules against taking technology from other companies, including Cisco,[41] and took pains to write 100% of its operating-system source code from scratch or based on open-source software.[42]

---

[39] Appx10811(Duda);Appx11243–11244(Almeroth);Appx45604–45610 (TX295).

[40] Appx12265;Appx63483,Appx63488–63489(TX9079,pp.73, 200(Gourlay)).

[41] Appx10808–10810(Duda);Appx46038(TX536);Appx50693(TX4667); Appx10944–10945(Sadana);Appx11940–11942(Ullal);Appx63447–63465 (TX9069).

[42] Appx10865–10866(Duda). "Under the hood," Arista innovated by writing the CLI's source code in Python and providing direct access to Unix pipes. Appx11807(Duda).

C. Arista overtook Cisco in the cloud-computing market.

Fueled by massive R&D spending,[43] Arista's innovations led swiftly to success as cloud titans like Microsoft, Facebook, eBay, Yahoo, Google, Amazon, and Apple became Arista customers, along with service providers like Comcast and Netflix, and large banks and traders.[44]

Although it remains a small player compared to Cisco,[45] Arista started beating out Cisco and other competitors for contracts in the data-center market—and Cisco noticed.[46] But Cisco never fully closed the gap. Arista's switches continue to beat Cisco's in terms of port density, power efficiency, and programmability.[47] An industry journal awarded Arista's DCS-7124 switch top marks, outscoring Cisco's

---

[43] Appx10932–10933(Sadana);Appx11931–11932(Ullal).

[44] Appx11928,Appx11936–11937(Ullal).

[45] Arista has around 12% of the high-speed Ethernet data-center switching market; Cisco had around 66%. Appx10775–10776(Duda).

[46] Appx10481–10484(Bakan);Appx57116–57119(TX6736);Appx107909–10710(Remaker);Appx52990–52991(TX5157);Appx11098–11102 (Kathail);Appx62752–62762(TX7956);Appx11730–11734,Appx11744–11748 (Chambers); Appx53831–53834(TX5495);Appx53605–53610(TX5423).

[47] Appx10947,Appx12494–12498(Sadana).

competing product;[48] and in a rare feat, Arista's 7500 switch twice won the annual Interop ITX Conference "Best of Show" award.[49] Arista's EOS now leads the industry in cloud computing.[50]

D. Cisco sued Arista for using "industry standard" CLI commands used by most switch vendors.

In December 2014, Cisco suddenly and without discussion or warning filed this suit in the Northern District of California alleging causes of action for both copyright and patent infringement.[51]

Late in the trial, after extensive briefing, the district court issued an analytic-dissection order finding that numerous aspects of the claimed works were not protected by copyright.[52] On appeal, Cisco does

---

[48] Appx11726–11729(Chambers);Appx53582–53604(TX5416).

[49] Appx11937–11938(Ullal).

[50] Appx10952–10953(Sadana).

[51] Appx11950–11951(Ullal).

[52] Appx1328–1347. Analytic dissection requires the plaintiff to submit a list of particular features in its works that that are allegedly similar to those in defendant's works so that the court may determine whether any of the allegedly similar features are protected by copyright. *See Apple Computer, Inc. v. Microsoft Corp.*, 35 F.3d 1435, 1443 (9th Cir. 1994).

not challenge that order or any of the jury instructions that implemented its findings.[53]

E. The jury instructions filtered out many unprotectable elements of Cisco's works, while the general verdict form that Cisco requested didn't require the jury to disclose its thought processes.

Consistent with Ninth Circuit law, the jury instructions gave Cisco the burden of proving similarity between the allegedly infringed and infringing works under both an "extrinsic" and an "intrinsic" test. *See Benay v. Warner Bros. Entm't*, 607 F.3d 620, 624 (9th Cir. 2010). The extrinsic test "is an objective comparison of specific expressive elements" that are protected by copyright. *Id.* By comparison, the intrinsic test is "a subjective comparison that focuses on whether the ordinary, reasonable audience would find the works substantially similar in the total concept and feel of the works." *Cavalier v. Random House, Inc.*, 297 F.3d 815, 822 (9th Cir. 2002) (quotation marks and citation omitted).

---

[53] Cisco therefore has waived any instructional challenge, *see SmithKline Beecham Corp. v. Apotex Corp.*, 439 F.3d 1312, 1319 (Fed. Cir. 2006); *Brown v. Rawson-Neal Psychiatric Hosp.*, 840 F.3d 1146, 1148–49 (9th Cir. 2016), and on reply cannot invoke the rule that appellate courts aren't bound by incorrect jury instructions when reviewing JMOL rulings. *See Pincay v. Andrews*, 238 F.3d 1106, 1109 (9th Cir. 2001).

The subjective, holistic intrinsic test was set forth in the last paragraph of Instruction 39.[54] The objective extrinsic test was set forth in Instructions 36 and 39. Instruction 36 gave the jury two pathways—reliance upon either "direct" or "indirect" evidence—for finding that Arista had copied aspects of Cisco's works.[55] The jury almost certainly took the "direct"-evidence pathway because the parties stipulated that Arista uses 506 IOS CLI command expressions, making indirect evidence of copying superfluous.[56]

Either way, however, the jury was instructed to compare Arista's works only with the "original, *protected* elements" of Cisco's copyrighted works.[57] To determine which elements were protected, the jury had to look to Instruction 39.

1. Instruction 39 made the overall command compilations protectable while filtering out most of their attributes as unprotectable.

Instruction 39 listed the six elements of Cisco's works found to be protected in analytic dissection. But Instruction 39 also subjected those

---

[54] Appx1397[Instruction39,final paragraph].

[55] Appx1394[Instruction36].

[56] Appx11201–11202;Appx51349–51359(TX4821).

[57] Appx1394[Instruction36].

elements to two key limitations: each was protected *only* as "a compilation," and only if the jury found that element to be "original."[58]

First in the list of protected compilations was "1. The selection and arrangement of Cisco's multiword command-line expressions" (hereinafter "command compilations").[59] The instructions didn't expressly define what any of the command compilations was or which commands they encompassed, but did define Cisco's "works as a whole" as being "its four user interfaces associated with its four operating systems" (as well as related manuals later found not infringed).[60]

---

[58] Appx1396–1397[Instruction39]. Instruction 33 explained that "[a] compilation is a work formed by the collection and assembling of preexisting materials or of data that are selected, coordinated, or arranged in such a way that the resulting work as a whole constitutes an original work of authorship." Appx1391.

Instruction 32 explained that "[t]he original parts of the plaintiff's work are the parts created . . . 1. independently by the work's author, that is the author did not copy it from another work; and . . . 2. by use of at least some minimal creativity." Appx1390.

[59] The "command compilations" encompass the compilations of commands found in each of the interfaces of the four Cisco operating systems. Other compilations that Instruction 39 identified as protectable were "2. The selection and arrangement of Cisco's modes and prompts"; "3. The collection of Cisco's screen responses and outputs"; "4. The collection of Cisco's help descriptions"; "5. Cisco's user interfaces as a whole as compilations of elements 1 through 4"; and "6. Each of Cisco's technical manuals." Appx1396.

[60] Appx1397[Instruction39].

Accordingly, we assume here that the command compilations consisted of the selection and arrangement of all of the multiword command-line expressions used in the user interface of each of the Cisco IOS, IOS XE, IOS XR, and NX-OS operating systems. None of these works were separately registered. Instead, they were scattered across—and culled by litigation counsel from among—numerous registered copyrights covering Cisco operating systems.[61] And they never were offered into evidence. *See* Part V.D.2., below.

Instruction 39 also told the jury *not* to consider various aspects of the command compilations—aspects that the court's analytic dissection had identified as being unprotected by copyright law. Thus, the jury was told to disregard, among other things,

- the "[i]ndividual words used" in the multiword commands;
- "[a]ny single multiword command";
- "[t]he idea or method of grouping or clustering commands under common initial words, such as 'show' or 'ip'";
- "[a]ny command hierarchy";

[61] Appx11164–11168(Lang);Appx50945–51058(TX4791);Appx51173–51252(TX4803);Appx10866–10867(Duda);Appx11047–11048(Kathail). *See* Part V.D.3., below, concerning legal effect of failing to separately register the CLI.

- "[t]he idea of making certain commands available only in certain modes";[62]
- the "[u]se of command syntax such as '[verb] [object] [parameters]"; and
- "[t]he idea of using multiword command expressions to manage or configure a device[.]"[63]

Again, Cisco does not challenge these instructions on appeal.

2. Instruction 61 defined a scènes à faire defense directed to whatever compilation attributes survived filtration and were found to be infringed.

Instruction 61 laid out the elements of Arista's scènes à faire defense—the issue on which Arista ultimately prevailed. To establish that defense, Arista had to show by a preponderance of the evidence that, "at the time Cisco created the user interfaces—not at the time of any copying—external factors other than Cisco's creativity dictated that Cisco select, arrange, organize and design its *original features* in the manner it did."[64] By "original features," Instruction 61 meant whatever

---

[62] *See* Appx10506–10510(Lougheed discussing modes).

[63] Appx1396–1397[Instruction39].

[64] The language of Instruction 61 closely tracked the scènes à faire discussion in *Oracle Am., Inc. v. Google Inc.*, 750 F.3d 1339, 1363 (Fed. Cir. 2014).

CLI features the jury found original and infringed, as no other CLI features could form a basis for liability and thus no other CLI features mattered for purposes of establishing a scènes à faire defense to liability.

Instruction 61 added that the scènes à faire doctrine "depends on the circumstances presented to the creator at the time of creation, not the circumstances presented to the copier at the time it copied."

3. Cisco sought and obtained a verdict form that did not require the jury to identify which compilation(s) it had found both original and infringed.

The district court granted Cisco's request[65] for a general (or "black box") verdict form that didn't ask the jury to identify which of the protected compilations the jury had found to be both original and infringed by Arista. The form, in relevant part, simply asked: "Has Cisco proven that Arista infringed any of Cisco's user interfaces? Yes (For Cisco) ___ No (For Arista) ___." The form then asked whether, as to the infringed user interfaces, Arista had proved its defenses of fair use, scènes à faire, or merger.

[65] Appx12836[court].

F. The jury found for Arista on its scènes à faire defense.

On the third day of deliberations, the jury checked off boxes on the verdict form indicating that (1) Cisco had proved that Arista had infringed some unspecified, protectable aspect of one or more compilations contained in one or more of the four Cisco user interfaces, but that (2) Arista had proved its scènes à faire defense as to whatever the jury had found to be infringed.

The district court later denied Cisco's renewed JMOL motion attacking the scènes à faire verdict, and accordingly denied as moot Arista's countervailing motion seeking JMOL of non-infringement.[66] The court's JMOL order noted that Arista hadn't needed to prove scènes à faire as to all of the compilations listed in Instruction 39, but only as to whatever compilations the jury had found to be original and infringed.[67] But the black-box verdict form hadn't required the jury to identify which compilations those were.[68] The court therefore streamlined its JMOL analysis by assuming—as the verdict form

---

[66] Appx2–20.

[67] Appx7.

[68] Appx9.

entitled it to do—that the jury found only one of the command compilations to be both original and infringed.[69]

Adopting that assumption (as we likewise do here), the court ruled that substantial evidence supported a jury verdict that the command compilations were scènes à faire. Among other things, the court found that "the functional choice of features to be implemented in a system dictates the contents of the compilation of CLI commands."[70] The court cited Arista's expert's testimony that "commands are linked to and driven by device features, both at the level of individual commands or sub-groups of commands and as to the overall compilation of commands within the CLI."[71]

The court also found substantial evidence to support the verdict based on evidence concerning the way functionalities were defined by industry networking protocols.[72]

The court concluded that "constraints flowing from the overall industry context and the basic functional nature of the CLI dictated the

---

[69] Appx9.

[70] Appx9.

[71] Appx9.

[72] Appx11.

overall structure and arrangement of Cisco's asserted compilation of commands that the jury found was original and infringed."[73] Accordingly, substantial evidence supported the jury's scènes à faire verdict and JMOL was denied.

IV. SUMMARY OF ARGUMENT

In this case, the jury—following instructions that Cisco does not challenge on appeal—found that Arista had proved its affirmative defense that Cisco's infringed compilations were "scènes à faire." Under the applicable jury instruction, this meant that, "at the time Cisco created the user interfaces—not at the time of any copying—external factors other than Cisco's creativity dictated that Cisco select, arrange, organize and design its original features in the manner it did."[74] In a meticulous opinion backed by a wealth of record citations, the district court denied Cisco's renewed Rule 50(b) JMOL motion and upheld the jury's scènes à faire verdict.

The district court was right: More-than-substantial evidence supported that verdict. The jury heard multiple witnesses explain that the features built into Cisco's switches—including the features of

[73] Appx11.

[74] Appx1406[Instruction61].

industry-standard networking protocols—determined the types of commands that Cisco had to include in its CLI in order to enable users to control those features. *See* Parts V.B.2 & 3., below. Product features thus were "external factors"—factors unrelated to Cisco's creativity in developing the command compilations—that dictated the selection of commands within the compilations. *See* Parts V.B.4, below.

Insisting that no reasonable jury could have found scènes à faire, Cisco's brief touts the "aesthetic" and "creative" choices that it exercised when writing individual CLI commands and organizing them into hierarchies. But the jury instructions—which, again, Cisco does not challenge on appeal—advised the jury to disregard individual words, individual multiword commands, the grouping of commands under common initial words, any command hierarchy, and numerous other aspects of the command compilations, because those compilation elements were not protected by copyright law. So Cisco's arguments about its aesthetic and creative choices as to commands, groupings, or hierarchies are simply not relevant.

Cisco's focus on the creativity of individual commands and hierarchies contradicts its own core argument that the jury was

required to assess both infringement and scènes à faire "at the compilation level"—a level at which, under the jury instructions, the wording and hierarchical arrangement of individual commands became legally irrelevant. By holding Arista to proof of scènes à faire "at the compilation level" while proclaiming its own creativity at the individual-commands-and-hierarchies level, Cisco repeatedly crosses the line from inconsistency to hypocrisy. The same double standard pervades Cisco's argument that the district court erred in allowing Arista to prove that "subgroups" of commands were scènes à faire. That's all the jury needed to find in order to reach its scènes à faire verdict, because Cisco's entire infringement case was itself based on proving that a small and arbitrary subgroup of commands infringed Cisco's copyright in the user interfaces. *See* Part V.B.5., below.

Cisco also tries to inject into this appeal an irrelevant argument that it waived below—namely, that scènes à faire can't be a defense to infringement findings made under the "virtual-identity" standard. Waiver aside, the argument lacks relevance because the jury wasn't required to—and likely didn't—apply the virtual-identity standard when it found infringement. *See* Part V.C., below.

Even if the Court were to find that the jury must have based its scènes à faire verdict on speculation—the applicable review standard[75]—it still should affirm the judgment on the grounds that (1) Cisco's CLI is a method of operation excluded from copyright protection under 17 U.S.C. § 102(b); (2) the jury lacked sufficient evidence to consider and compare the disputed works as a whole; (3) Cisco failed to prove that its "user interfaces" were copyrighted works apart from its registered operating systems; and (4) Cisco failed to prove infringement at the compilation level. *See* Part V.D., below.

For all these reasons, as explained more fully below, the Court should affirm the judgment.

V. ARGUMENT

A. Standards of review and governing law.

This Court applies regional (here, Ninth Circuit) law to resolve issues of substantive copyright law. *Chamberlain Grp., Inc. v. Skylink Techs., Inc.*, 381 F.3d 1178, 1191 (Fed. Cir. 2004).

Cisco appeals from the order denying its JMOL motion. In reviewing a district court's JMOL rulings, this Court applies regional

[75] *See* Part V.A., below.

circuit law to determine the review standards that bound the district court and that apply on appeal. *See Wi-Lan, Inc. v. Apple, Inc.*, 811 F.3d 455, 461 (Fed. Cir. 2016); *see also SynQor, Inc. v. Artesyn Techs., Inc.*, 709 F.3d 1365, 1373 (Fed. Cir. 2013). Five such standards apply here.

1. "[T]he standard of review for the denial of a motion for judgment as a matter of law is the same as the standard of review for reviewing a jury's verdict: both the verdict and the denial of the motion must be affirmed if there is substantial evidence to support the verdict." *Harper v. City of Los Angeles*, 533 F.3d 1010, 1021 n.9 (9th Cir. 2008).

2. Substantial evidence "is evidence adequate to support the jury's conclusion, even if it is also possible to draw a contrary conclusion." *Id.* at 1021. "[E]ven when the jury is given an essentially black box verdict form—that is, a form that merely asks the jury to answer 'yes' or 'no,'" this Court "presume[s] all factual disputes were resolved in favor of the verdict." *Agrizap, Inc. v. Woodstream Corp.*, 520 F.3d 1337, 1343 (Fed. Cir. 2008).

3. When applying the substantial-evidence standard, the district court must "review all of the evidence in the record" while "draw[ing] all reasonable inferences in favor of the nonmoving party"—

here, Arista—and the court cannot "make credibility determinations or weigh the evidence." *Reeves v. Sanderson Plumbing Prods., Inc.*, 530 U.S. 133, 150 (2000); *accord Johnson v. Paradise Valley Unified Sch. Dist.*, 251 F.3d 1222, 1227 (9th Cir. 2001). An inference is "reasonable" when it may be drawn from the evidence "'without resort to speculation.'" *Lakeside-Scott v. Multnomah Cty.*, 556 F.3d 797, 802–03 (9th Cir. 2009). JMOL is therefore appropriate only if the jury must have relied on speculation to reach its verdict. *Id.*

4. In addition, the district court "must disregard all evidence favorable to the moving party that the jury is not required to believe. That is, the court should give credence to the evidence favoring the nonmovant [i.e., Arista] as well as that 'evidence supporting the moving party that is uncontradicted and unimpeached, at least to the extent that that evidence comes from disinterested witnesses.'" *Reeves*, 530 U.S. at 151; *accord Johnson*, 251 F.3d at 1227.

5. On appeal, the Ninth Circuit reapplies the same JMOL standards de novo, thus affording "significant deference to the jury's verdict and to the nonmoving parties," *A.D. v. Cal. Highway Patrol*, 712 F.3d 446, 453 (9th Cir. 2013), and preserving the jury's role as "the

constitutional tribunal provided for trying facts in courts of law." *Harper*, 533 F.3d at 1021 (citation omitted).

B. This Court should affirm the scènes à faire verdict because substantial evidence showed that product features dictated Cisco's selection of the types of commands to include in the command compilations.

To prevail on its *scènes à faire* defense under Instruction 61, Arista had to prove that "at the time Cisco created the user interfaces . . . external factors other than Cisco's creativity dictated" the selection and arrangement of Cisco's multiword command-line expressions.[76]

The district court's JMOL order found that a reasonable jury could have found that "the functional choice of features to be implemented in a [switch] dictates the contents of the compilation of CLI commands."[77] In other words, once Cisco decided what features a switch would have, *that* choice—not Cisco's purported "creativity" in curating CLI commands—dictated the types of commands that went into the switch's interface.

The district court's product-features-constraint ruling was correct and deserves affirmance.

---

[76] Appx1406[Instruction61].

[77] Appx9; *see also* Appx12126(Black)(citing supporting evidence).

1. The jury reasonably could have found that the only originality in the command compilations was Cisco's selection of the types of commands necessary to control product features.

Construed together, as they must be,[78] the instructions in this case told the jury that it had to find a compilation (1) original before it could find that compilation to be (2) infringed and (3) scènes à faire. The instructions also told the jury that a compilation could be original if *either* its selection *or* its arrangement of commands was original. Applying those instructions, the jury reasonably could have found that the only originality in the command compilations was Cisco's *selection* of the types of commands necessary to control product features—and that the evidence demonstrated that this selection was scènes à faire.

---

[78] *See United States v. Park*, 421 U.S. 658, 674 (1975) ("[A] single instruction to a jury may not be judged in artificial isolation, but must be viewed in the context of the overall charge."); *Benatar v. United States*, 209 F.2d 734, 743 (9th Cir. 1954) ("[A]n instruction must be construed as a whole. A trial judge cannot be expected to cram all the limitations, qualifications, exceptions, and distinctions of a legal principle into one sentence or even into one paragraph. Judicial pronouncements, like every other type of human discourse, must be allowed some elbow room.").

More specifically:

1. Construed together, the instructions correctly informed the jury that it needed to find either the selection ***or*** the arrangement of a command compilation original, but need not find ***both*** original.

Instruction 33, drawn directly from the Copyright Act, explained that copyright can protect a compilation of preexisting material that is "selected, coordinated, *or* arranged in such a way that the resulting work as a whole constitutes a work of authorship."[79] This instruction built on Instruction 32, which explained that the original (and thus potentially protected) parts of a work are those that the author created "by use of at least some minimal creativity."[80]

Instruction 39 listed the compilations that the jury could find protected ***if*** the jury found them to be original.[81] The first in that list were the command compilations—"[t]he selection and arrangement of Cisco's multiword command-line expressions."[82] As Instruction 33 made clear, if Cisco "selected, coordinated, ***or*** arranged" the commands in a

---

[79] Appx1391; *see* 17 U.S.C. § 101 (defining "compilation").

[80] Appx1390; *see Feist,* 499 U.S. at 345–46.

[81] Appx1396–1397[Instruction39].

[82] Appx1396[Instruction39].

sufficiently original manner, then (in the words of Instruction 39) "[t]he selection and arrangement" of the commands was original. Not only is this exactly how the Copyright Act defines a compilation, it also is common sense. For example, a highly creative *selection* of poems for an anthology isn't rendered uncopyrightable merely because the author *arranges* the poems in an unoriginal order (alphabetically, for example).[83]

This reading not only harmonizes Instruction 39 with Instruction 33 but is consistent with the leading Supreme Court opinion on compilations, *Feist Publications, Inc. v. Rural Telephone Services Co.*, 499 U.S. 340 (1991). *Feist* held that "even a [telephone] directory that contains absolutely no protectable written expression, only facts, meets the constitutional minimum for copyright protection if it features an original selection ***or*** arrangement." *Id.* at 348 (emphasis added); *see also id.* at 350–51; *id.* at 356 (quoting 17 U.S.C. § 101). Although the court also used the phrase "selection ***and*** arrangement" several times, *see id.* at 348, 349, 350, 353, 362, its use of that phrase plainly wasn't intended to contradict its holding that a compilation can meet the constitutional

[83] *See Feist,* 499 U.S. at 362 (alphabetical arrangement is not creative).

minimum for copyright protection if it features "an original selection ***or*** arrangement." *Id.* at 348 (emphasis added).

Here, likewise, the jury could have found a command compilation protectable based on *command selection alone*. That conclusion is reinforced by Instruction 39, which removed from the jury's consideration nearly everything that could be considered an "arrangement," including "[a]ny command hierarchy," "[t]he idea or method of grouping or clustering commands under common initial words, such as 'show' or 'ip,'" and "[t]he idea of making certain commands available only in certain modes."

2. Next, the jury had to determine what aspect of Cisco's command selection was both original and infringed. As it set about that task, the jury could have worked its way up the chain from individual words, to particular strings of words (commands), to the principles for ordering the words within strings (syntax), to the grouping of commands under initial words, all the way up to the very idea of a CLI, without ever encountering *anything* that could result in liability if copied.

So what's left? A reasonable jury could have concluded: *Nothing*—except for Cisco's selection of the general types of commands it needed to include in the CLI to allow operators to control all of the features in Cisco's products.[84] For example, the jury could have based infringement upon the selection of the subgroup of commands pertaining to the Border Gateway Protocol (BGP) that are common to Cisco's IOS and Arista's EOS. And substantial evidence showed that this selection was dictated—as one might expect—by product features of the BGP protocol. *See* Parts V.B.2 & 3., below.

Of course, we'll never know the jury's actual reasoning because Cisco sought and obtained a black-box verdict form that entirely obscured the jury's thought processes. We therefore don't know which of the compilations the jury found original or what facts led the jury to that finding.

Accordingly, this Court must affirm so long as there is substantial evidence to support any set of facts that supports the verdict.

---

[84] Although Instruction 39 told the jury to disregard the "function" of any asserted interface feature (Appx1396), Instruction 32, reflecting the rule in *Feist*, set a low threshold of "minimal creativity" that enabled a reasonable jury to avoid characterizing the selection of types of commands to support product features as being merely functional and unprotectable. Appx1390.

2. Substantial evidence showed that product features dictated Cisco's selection of which types of commands to include in the compilations.

Cisco's one-paragraph discussion of the product-features constraint asserts that the district court had no evidentiary basis for concluding that product features dictated not just "particular commands" but the overall selection and arrangement of commands in the command compilations.[85]

But Cisco is wrong. Even granting Cisco's point that Arista had to prove scènes à faire "at the compilation level,"[86] substantial trial evidence supported the district court's conclusion that "commands are linked to and driven by device features, both at the level of individual

---

[85] CB39. Cisco faults the district court for discounting testimony by two supposedly "disinterested third-party witnesses," Phillip Schafer and Balaji Venkatraman, concerning the lack of constraints on programmers who formulate CLI commands. CB30. While it's true that a court ruling on JMOL should credit "uncontradicted and unimpeached" evidence from "disinterested" witnesses, *see Reeves*, 530 U.S. at 151, Schafer's and Venkatraman's testimony *was* contradicted by the testimony discussed in this section, and the district court correctly concluded that the jury needn't have credited it. Even if credited, their testimony didn't prove lack of constraints "at the compilation level," which Cisco insists is "the only relevant unit of analysis." CB23–24.

[86] *See* CB23,41,45; *see also* CB24,27,29 & Part V.B.5., below.

commands . . . and as to the overall compilation of commands within the CLI."[87]

1. Arista's networking expert, Dr. John Black, cited "the large number of constraints to describe the [product] feature you are implementing" as a principal basis for his opinion that Arista possessed a valid *scènes à faire* defense.[88] As he explained: "If you . . . are going to go out and create a new CLI command, . . . the first thing you are going to do is think, well, *what does it do*, what's the [product] feature that I'm trying to describe[?]"[89]

Dr. Black added: "[W]hen you are designing a product you don't go, hey, I have this nifty CLI command, I wonder what kind of feature I can put in [the product] to correspond [with the command]. It's the other way [around], right? You decide what features are going into the switch [and] then that dictates what the computation [sic: compilation] of the CLI commands is."[90]

---

[87] Appx9.

[88] Appx12113–12114.

[89] Appx12110–12111.

[90] Appx12126.

Dr. Black thus explained how product features dictated not just the selection of particular commands but the contents of an overall command compilation. The jury was entitled to credit that testimony and other similar testimony by Dr. Black, which was entirely plausible and never contradicted. *See, e.g.,* Appx12222 ("[I]f there's not a hardware feature in the device, I don't think you want to take commands that describe a feature that's not present."); Appx12256–12257 ("[I]t wouldn't make sense" for a product that didn't implement a particular network protocol to have a command for that protocol); Appx12261–12262 ("Arista has a whole bunch of features that they've added to their product that require new CLI commands, thousands of those."). We could stop here—but there's more.

2. Dr. Black's testimony received the apparent endorsement of Cisco's counsel, who repeatedly pressed witnesses to agree that product features dictate the contents of *any* command compilation. For example, the jury heard this exchange between Cisco's counsel and former Cisco senior engineer Anthony Li[91] concerning the industry-standard CLI

[91] Appx11868–11869(Li).

commands that Li copied while designing switches at Procket Networks in 1999–2004:

> Q. So I think when you were talking about the Procket switch, . . . you said that you copied the commands that were relevant to the feature set you had, isn't that right?
>
> A. That's correct.
>
> Q. So in other words, you didn't copy commands that weren't relevant to features you didn't have in the product; right?
>
> A. Yes. For example, since we did not support Apple Talk, we did not implement the Apple Talk commands out of IOS.
>
> Q. Right. It wouldn't make much sense to copy commands that don't have anything to do with the features in your products; right?
>
> A. Yes.[92]

3. Cisco's counsel drove this point home again when cross-examining Arista President and CEO Jayshree Ullal:[93]

> Q. And you are generally aware that the initial set of products that Arista was making had limited features and limited commands, correct?
>
> A. Because they had limited features, they had limited commands, that's correct.
>
> Q. So when you introduced the Spine product in 2011 and [20]12, there were many, many more features and many,

[92] Appx11870(Li).

[93] Appx11915(Ullal).

> many more commands that were added to EOS; you don't deny that, do you?
>
> A. No. The more the features, the more the command line.[94]

4. Former Cisco Software Director Greg Satz[95] likewise testified that the list of commands in a switch's operating system expands "[d]epending on the feature set . . . [i]f it was an extension of existing feature set or if [the feature] was brand new."[96]

5. Dell Vice President Gavin Cato[97] testified that if a switch has a Virtual Local Area Network ("VLAN") feature, then "there's an expectation that a VLAN and the terminology around VLAN will

---

[94] Appx11971(Ullal). Of course, Li and Ullal weren't talking about constraints that existed when Cisco first created its IOS CLI—but a reasonable jury would have inferred that the same product-features constraint must have dictated Cisco's selection of CLI commands. Indeed, to conclude otherwise would have been illogical.

[95] Appx11872;Appx63472(TX9073,p.8(Satz)).

[96] Appx11872;Appx63474(TX9073,pp.45–46(Satz)). There was evidence that once a command went into a compilation it tended to remain there, even after it was no longer useful. Appx12190(Black). For scènes à faire purposes, however, what matters is why the command got selected in the first place, not why it languished in the compilation thereafter.

[97] Appx12326;Appx63494(TX9081,p.10(Cato)).

somewhere appear in the CLI along with the parameters necessary to structure VLAN so that it interoperates across multiple switches."[98]

6. Cisco Chief Network Architect Pradeep Kathail[99] testified that "[e]ach [CLI] command relates to one of the functions in the switch."[100]

In sum: Substantial evidence adduced by both sides demonstrated that the contents of the command compilations were dictated not by Cisco's purported "creativity" in curating commands for inclusion in the compilation but by the "external factor" of product features.[101] Just as the features of an old-style stereo dictated which types of knobs had to be included on its control panel—volume, bass, treble, balance—the features of a switch dictated which types of commands (or "nerd knobs") had to be included in its CLI.[102] Accordingly, substantial evidence

---

[98] Appx12326;Appx63495(TX9081,p.36(Cato)).

[99] Appx11045(Kathail).

[100] Appx11086(Kathail).

[101] *See* Appx1406[Instruction61].

[102] *See* Appx12091(Black).

supported the scènes à faire verdict and the judgment should be affirmed.[103]

3. Protocol support is one example of a product feature that imposed external constraints on command selection.

A huge variety of product features undoubtedly dictated command selection; but as an example, consider the feature of supporting a particular networking protocol.

Senior Cisco engineer Kirk Lougheed was present at the creation of Cisco's CLI. Beginning in 1986, he decided that Cisco would adopt a command-line interface instead of a graphical user interface or a menu-driven interface;[104] and he wrote many of the CLI's early commands and developed its basic hierarchies and modes.[105]

---

[103] The product-features constraint also answers Cisco's repeated assertion that no "mechanical" or "equipment-based" constraints dictated the contents of the protectable compilations. *See* CB2,7,30. Product features *are* "equipment-based." Anyway, nothing in the instructions or in any relevant authority limits the external factors to mechanical or equipment-based constraints. *See, e.g., Oracle Am.,* 750 F.3d at 1363 (citing examples of external constraints relevant to scènes à faire, including "'widely accepted programming practices within the computer industry'").

[104] Appx10503–10505(Lougheed).

[105] Appx10505–10523(Lougheed).

According to Lougheed, Cisco's first product supported only one protocol—the Internet Protocol ("IP"), which was formalized around 1980 by the Internet Engineering Task Force and has evolved through various versions to IP version 6 ("ipv6").[106] Borrowing from that protocol, 148 of the 506 asserted commands used the term "ip" and roughly 45 used "ipv6."[107]

Since then, the Internet community has developed hundreds of features and has published their standard functions and parameters in specifications and requests for comments.[108] Each decision by Cisco to support another network protocol or feature thus became an external constraint dictating the inclusion of a mini-compilation of commands referencing that protocol. For example, 24 of the allegedly copied commands refer to the Border Gateway Protocol ("BGP");[109] 35 refer to the Open Shortest Path First Protocol ("OSPF");[110] 27 refer to the

---

[106] Appx10513–10514(Lougheed); *see also* Appx11293–11294(Almeroth); Appx51910–51946(TX5040).

[107] Appx11291–11292(Almeroth);Appx11347(Almeroth).

[108] Appx12085,Appx12094–12095,Appx12101–12102,Appx12105–12107(Black).

[109] Appx11357(Almeroth);Appx10617–10620,Appx10636(Lougheed); Appx11061–11062(Kathail).

[110] Appx11347–11348(Almeroth).

Simple Network Management Protocol ("SNMP");[111] 21 refer to the Internet Group Management Protocol ("IGMP");[112] 20 refer to the Protocol Independent Multicast ("PIM");[113] 19 refer to the Multicast Source Discovery Protocol ("MSDP");[114] 10 refer to the Intermediate System to Intermediate System Protocol ("IS-IS");[115] and so on.[116]

Protocol support is therefore a prime example of a constraint that dictated the overall selection of the types of commands that went into the command compilations.

4. Cisco product features are "external factors" within the meaning of Instruction 61.

Cisco seems likely to argue on reply that the features included in Cisco products are "*internal*" to Cisco and therefore can't qualify as "*external* factors other than Cisco's creativity" that dictated the contents of the command compilations for scènes à faire purposes. As Cisco puts it (without citing to any authority): "The 'external' element [of scènes à

---

[111] Appx11354–11356(Almeroth).

[112] Appx11360(Almeroth).

[113] Appx11365–11366(Almeroth);Appx57461–57526(TX6870).

[114] Appx11366–11367(Almeroth);Appx57578–57596(TX6910).

[115] Appx11368(Almeroth);Appx57376–57460(TX6824).

[116] Appx12088–12089,Appx12093–12094(Black); *see generally* Appx12095–12110(Black).

faire] requires that the constraints originate outside the author"[117]—that is, entirely outside of Cisco as an organization.

That argument is specious and must be rejected if pursued on reply. Instruction 61 does state that a compilation is scènes à faire if its contents were dictated by "external factors other than Cisco's creativity." But that phrase can only mean "external factors other than Cisco's creativity *in selecting or arranging the commands in the command compilations*"—because that is the only creativity that the Copyright Act cares about in a compilation. "Creativity" in copyright

---

[117] CB28; *see also* CB3(asserting that Cisco's internal guidance to engineers isn't "evidence of any ***external*** constraint" (emphasis in original)),23(referring to factors "external to Cisco's creativity"),27(same),29(referring to constraints "external to Cisco");36("factors external to Cisco's creativity").

Although this Court's *Oracle America* decision refers once in passing to "considerations external to the author's creativity," 750 F.3d at 1364, the Court also granted external-factor status to "'the mechanical specifications of the computer on which a particular program is intended to run'" and to "'widely accepted programming practices within the computer industry.'" *Id.* at 1363. Cisco product features are and always have been "mechanical specifications" of switches that run the IOS CLI; and the testimony quoted in Parts IV.B.3. and 4. shows that selecting commands to support product features is a "widely"—indeed, universally—"accepted programming practice" in the network-switch industry.

law is tied to *originality*;[118] and the jury accordingly was instructed that the originality of a compilation depends on the manner in which its elements are "selected, coordinated, or arranged."[119] Thus, Cisco's creativity in selecting or arranging commands is the only creativity that logically matters for purposes of determining whether the command compilations are protected by copyright, have been infringed, or are scènes à faire.

For example, support for a particular networking protocol is a product feature outside the command compilations; and that feature constrains the selection of what types of commands go into the compilation. *See* Parts V.B.2. & 3., above. But any creativity that Cisco exercised in choosing protocols to support—assuming that that was a creative choice at all—is irrelevant to the scènes à faire inquiry because Arista isn't charged with copying the selection of protocols, and the selection of protocols isn't what's accused of being scènes à faire. The same is true of all of the other many product features external to the command compilations. Moreover, a decision about which product

---

[118] *See Feist,* 499 U.S. at 345 (discussing the "creative spark" that is a prerequisite of originality); *see also* Instruction 32.

[119] Appx1391[Instruction33].

features to include is an unprotectable idea or system that falls outside the scope of copyright law. *See* 17 U.S.C. § 102(b); *Bikram's Yoga College of India v. Evolation Yoga,* 803 F. 3d 1032, 1036–41 (9th Cir. 2015) (sequence of yoga postures chosen to accomplish health objectives is not copyrightable, even if sequence is "beautiful" or "reflects [the author's] aesthetic preferences"); Part V.D.1., below. That type of "creativity" therefore has no bearing on the scènes à faire inquiry.

In the literary context in which scènes à faire originated, the relevant external constraint has *always* been a threshold choice made by the author herself as part of the infringed work itself. Two classic examples make the point. In a storm, two characters seek refuge in a church, and certain additional ideas naturally follow (e.g., playing a piano, a prayer, a "hunger motive," etc.). *Cain v. Universal Pictures Co.*, 47 F. Supp. 1013, 1017 (S.D. Cal. 1942) (Yankwich, J.)[120] Or a character burns her hand on a cigarette; later, "[s]omething had to be done with that burn, and the author use[d] it as a means of identification." *Schwarz v. Universal Pictures Co.*, 85 F. Supp. 270, 275 (S.D. Cal. 1945)

[120] Judge Leon Yankwich introduced the phrase scènes à faire to copyright law, and the major copyright treatises (Goldstein, Nimmer, and Patry) all cite his decisions.

(Yankwich, J.). In both examples, the initial plot choices that imposed the constraint—two people taking refuge in a church, a character burning her hand—were not "external" to the author's creativity or even to the specific infringed work. Rather, they were chosen by the author, yet were "external" to, and constrained, the subsequent story elements that the court found to be scènes à faire.

The same may be true in a software-infringement case. The fact that software *can* run on hardware platforms made by others and *can* interact with software authored by others creates opportunities for "external" constraints that don't exist in literature—but that shouldn't blind us to the fact that external constraints created by the same author are the norm, not a novelty, under this doctrine. For example, once Apple decided to include "windows" as a product feature of its GUI, that self-imposed constraint rendered Apple's "mere use of *overlapping*" windows scènes à faire. *Apple Computer, Inc. v. Microsoft Corp.*, 35 F.3d 1435, 1444 (9th Cir. 1994). (Emphasis added.)

Moreover, Cisco's initial, constraining choice to include a product feature—e.g., supporting a protocol—often *was* "external" to Cisco in the sense of being made "in response to customer demand or to ensure

compatibility with equipment already installed" by Cisco's customers. *Mitel, Inc. v. Iqtel, Inc.*, 124 F.3d 1366, 1375 (10th Cir. 1997); *see also Gates Rubber Co. v. Bando Chem. Indus., Ltd.*, 9 F.3d 823, 838 (10th Cir. 1993) (citing target-industry demands as external constraint).[121] *See* Part V.B.3., above. Cisco responds that anything that it has ever done to satisfy customer demands or requirements—including ensuring product "usability"—was "internal" rather than "external" because it was merely "aspiration[al]" and reflected "internal guidelines" for achieving authorial (i.e., market) success.[122] This argument guts the scènes à faire doctrine and must be rejected. *Any* device that Cisco developed for use in a network could have been the result of "internal guidelines" while still remaining subject to countless external constraints in order to function in an environment connected to many other devices. If those constraints can't be considered for scènes à faire purposes, then nothing can be scènes à faire.

---

[121] Cisco's distinction of *Mitel* therefore misses the mark. CB31.

[122] CB24.

Accordingly, Cisco's product features are an "external factor" under Instruction 61 and the Court should reject any argument to the contrary.

5. Cisco's arguments violate its own "symmetry principle" that infringement and scènes à faire both had to be proved "at the compilation level."

A pervasive theme of Cisco's brief—we'll call it "the symmetry principle"—is that Arista had to follow the same rules when proving scènes à faire that Cisco had to follow when proving infringement. Both had to be proved "at the compilation" level and without regard to the many elements identified as unprotectable in Instruction 39.[123] Cisco asserts (erroneously, *see* Parts V.B.2. & 3., above) that enforcing this symmetry rendered irrelevant all of Arista's trial evidence showing that major elements of Cisco's CLI were dictated by external factors such as product features, Internet protocols, and customer demands.

But Cisco abandons its own symmetry principle when defending the infringement verdict against Arista. Instead of sticking to "the compilation level," Cisco shifts gears and relies entirely on lower-level

[123] *See* CB23,24,27,29,41,45.

elements identified as unprotectable in Instruction 39. Thus, Cisco dwells at length on irrelevant evidence showing that:

- Cisco engineer Kirk Lougheed made aesthetic judgments reflecting his love of hyphens and of spelling out individual words when he "author[ed] and structur[ed] the [initial Cisco CLI] commands";[124]
- "Cisco engineers were free to use any number of different individual terms within the multiword commands";[125]
- "Cisco engineers were free to sequence the selected words in numerous ways within the multiword commands";[126]
- "different designers . . . can choose different words, different hierarchies, different syntax for the same functions";[127] and
- a CLI doesn't have to be "text-based."[128]

Individual words. Individual command expressions. Hierarchies. Syntax. The idea of text-based switch commands. Each of these concepts

---

[124] CB11.

[125] CB12.

[126] CB12.

[127] CB14.

[128] CB15.

is one that Instruction 39 told the jury to disregard as unprotectable and that accordingly could not form the basis for an infringement verdict.[129] Yet each is a concept that Cisco hypocritically rules out as a basis for the jury's scènes à faire verdict. We agree that Cisco and Arista had to play by the same rules when proving infringement and scènes à faire—but Cisco's arguments don't achieve that symmetry.[130]

Cisco again violates the symmetry principle when arguing that constraints on *subgroups* of commands were irrelevant for purposes of proving scènes à faire because the district court barred Cisco from using subgroups of commands to prove infringement.

But the argument's basic premise is wrong. Cisco seems to have forgotten that its whole infringement case was about using a small subgroup of 506 allegedly copied commands to *infer* that the selection

---

[129] *See Hutchins v. Zoll Med. Corp.*, 492 F.3d 1377, 1385 (Fed. Cir. 2007) ("Copyright does not protect individual words and 'fragmentary' phrases when removed from their form of presentation and compilation. Although the compilation of public information may be subject to copyright in the form in which it is presented, the copyright does not bar use by others of the information in the compilation.").

[130] Cisco's reliance on unprotected sub-compilation-level elements demonstrates (indeed, effectively concedes) that it failed to prove infringement at the compilation level—an alternative ground for affirmance. *See* Part V.D.4., below.

and arrangement of an entire compilation containing roughly 16,000 commands had been infringed to a non-de minimis extent.[131] Indeed, the "infringement subgroup" forming the basis for that inference may have been far *smaller* than 506, because no instruction told the jury that it must find all 506 commands both original and infringed before it could return an infringement verdict. (Cisco likely would have objected in the strongest possible terms to such an instruction.)

Under Cisco's symmetry principle, therefore, it should have been equally acceptable for the jury to use a subgroup of commands that it had found to be original, infringed, and scènes à faire to draw the inference that the selection and arrangement of an entire compilation also was scènes à faire.

For example, the jury could have used Arista's copying of the subgroup of commands that reference the IGMP protocol to draw an inference that Arista had committed a more-than-de minimis infringement of the selection or arrangement of an entire command compilation. Next, the jury could have found that this IGMP subgroup

---

[131] *See* Appx12204–12205(Black: only guiding principle to the subgroup of 506 commands was that they were shared by Cisco's IOS and Arista's EOS).

was scènes à faire; and then it could have inferred from that finding that the selection or arrangement of the entire surrounding compilation was likewise scènes à faire.[132]

That's what "subgroup symmetry" would have looked like in this case—but Cisco again prefers to rig the scales in its favor. Page after page of its brief is devoted to arguing that the same jury that it prevailed upon to infer infringement from a subgroup of 506 (or fewer) commands had no right to infer scènes à faire from subgroups that referred to networking protocols like "ivp6," "ip igmp," and "OSPF" or that employed other industry-standard terms.[133]

"Symmetry" aside, the jury instructions, on their face, allowed the jury to find scènes à faire if Arista proved that defense as to any aspect of Cisco's compilations that the jury found both original and infringed. Instruction 61 stated without further elaboration that Arista could prove scènes à faire as to "portions of Cisco's user interfaces," not entire compilations.[134] Instruction 39 likewise *didn't* say that the jury had to find that Arista had copied an "entire compilation" of commands in any

---

[132] *See* Appx11360–11365(Almeroth)(re:IGMP subgroup).

[133] CB11–16.

[134] Appx1406.

work as a whole before it could vote "yes" on infringement. Instead, Instruction 39 said that Arista could be held liable for copying the "*selection and arrangement*" of a command compilation—concepts that need not involve consideration of every command as long as the infringed subgroup possessed some originality in its creation that Arista could and did copy to a more-than-de-minimis extent.[135]

By analogy, you could copy the selection and arrangement of notes in a symphony to a more-than-de minimis extent even if you omitted one or more of the symphony's movements, or all of its woodwind parts. The symphony still would be the "relevant unit of analysis,"[136] to use Cisco's term. But if every musical phrase that you copied was scènes à faire, you couldn't be held liable for infringement even if other parts of the symphony were not scènes à faire. *See Hutchins v. Zoll Med. Corp.*, 492 F.3d 1377, 1384–85 (Fed. Cir. 2007) (holding that copying of two standard CPR-related phrases from list of 27 and paraphrasing of three others was scènes à faire) (applying First Circuit law). Here, likewise, if every aspect of the compilations that the jury found original and copied

---

[135] *See* Appx1398(Instruction 41, requiring "greater than *de minimis*" copying to find infringement).

[136] CB24,28.

was scènes à faire and therefore incapable of supporting a judgment for Cisco, then the inquiry is over and the judgment for Arista must be affirmed.

Of course, the jury's actual thinking process was forever obscured by Cisco's black-box verdict form. Thus, we will never know what aspect of Cisco's compilations the jury concluded was both original and copied. But as long as substantial evidence supported a scènes à faire verdict for *any* aspect of Cisco's compilations that the jury could have found infringed, the judgment must be affirmed. *See Agrizap,* 520 F.3d at 1343 (even with a black box verdict form, this Court "presume[s] all factual disputes were resolved in favor of the verdict."). That was the basis for the district court's careful JMOL opinion. And as previously discussed, *see* Parts V.B.2. and 3., more-than-substantial evidence supported that finding.[137]

Accordingly, the Court should reject Cisco's "entire compilation" argument and affirm the judgment.

---

[137] Moreover, as discussed above at Parts V.B.2. and 3., much of the evidence proving the product-features constraint ***did*** concern the CLI as a whole, not a subgroup. That evidence would refute Cisco's "entire compilation" argument even if Cisco weren't bound by its own symmetry principle.

C. Cisco waived its irrelevant argument that scènes à faire can't be a defense to infringement findings made under the virtual-identity standard.

Cisco argues that, as a matter of law, a reasonable jury could not find both that Cisco had proved copying under the "virtual identity" standard of similarity ***and*** that whatever Arista had copied was scènes à faire.

The district court, applying Ninth Circuit standards, determined that Cisco waived that argument.[138] That finding was correct. Cisco never objected that the instructions must bar the jury from returning a scènes à faire verdict if it found infringement under the virtual-identity standard. And the district court correctly found that Cisco's Rule 50(a) motion failed to preserve the argument, even under the Ninth Circuit's forgiving standard. *See E.E.O.C. v. Go Daddy Software, Inc.*, 581 F.3d 951, 961 (9th Cir. 2009). It's not that the argument was "ambiguous or inartfully made." *Id.* It's that the motion in no way, shape, or form articulated any argument linking the virtual-identity standard to the scènes à faire defense.[139] Moreover, the motion did not cite once to Cisco's principal authority for this argument, *Appl*e, 35 F.3d 1435; and

[138] Appx2–20.

[139] Appx1348–1373.

it cited Cisco's other case, *Ets-Hokin v. Skyy Spirits, Inc.*, 323 F.3d 763 (9th Cir. 2003), only in passing for the proposition that "[o]riginality requires only a minimal creative spark."[140] This is nothing like *Antonick v. Electronic Arts, Inc.*, 841 F.3d 1062 (9th Cir. 2016), where the defendant's pre- and post-verdict JMOL briefs both argued that "the failure to place the source code in evidence was fatal to [plaintiff's] claim that [defendant] had copied his work," *id.* at 1068—the very ground on which JMOL was granted. *See id.* at 1065–66.

Even if not waived, Cisco's argument is irrelevant because there is no reason to suppose that the jury applied the virtual-identity standard. Instruction 36 gave the jury two pathways—reliance upon either "direct" or "indirect" evidence—for finding that Arista had copied aspects of Cisco's works.[141] But only the indirect-evidence pathway led to application of Instruction 39's virtual-identity standard. And the jury almost certainly took the other (direct-evidence) pathway, since the parties stipulated that Arista uses 506 IOS CLI commands, making any

[140] Appx1359.

[141] Appx1394[Instruction36]; *see* Part III.E., above.

resort to indirect evidence unnecessary.[142] Indeed, Cisco urged the jury in closing to avoid the virtual-identity standard by relying on direct evidence in the form of admissions.[143]

Cisco responds that even if the jury took the direct-evidence pathway, "no reasonable jury could have found that Arista's copying was anything other than virtually identical."[144] Not so. It's undisputed that Arista copied only a miniscule subgroup of the commands contained in the "entire compilation"—which Cisco insists is "the only relevant unit of analysis."[145] Thus, a reasonable jury could and likely did find direct evidence of copying rather than indirect evidence of copying under the "virtual identity" standard.

---

[142] Appx11201–11202;Appx51349–51359(TX4821).

[143] Appx12705.

[144] CB57.

[145] CB24.

D. The judgment also may be affirmed on four alternative grounds.[146]

"[A]n appellate court may affirm a judgment of a district court on any ground the law and the record will support so long as that ground would not expand the relief granted." *Glaxo Grp. Ltd. v. TorPharm, Inc.*, 153 F.3d 1366, 1371 (Fed. Cir. 1998); *see also Gallegos v. Reinstein*, 560 F. App'x 669 (9th Cir. 2014). Four such grounds appear here.

1. Cisco's CLI is a method of operation excluded from copyright protection by § 102(b).

A fundamental rule of copyright is that ideas—including systems and methods of operation—cannot be protected by copyright; only creative *expression* of ideas is copyrightable. *Baker v. Selden*, 101 U.S. 99 (1879); *Hutchins*, 492 F.3d at 1383 ("[C]opyright protection does not extend to the methods that are performed with [computer] program guidance."); 17 U.S.C. § 102(b). Here, Cisco's witnesses described the CLI in purely functional terms as a "mechanism" for managing

---

[146] Arista presented each of the arguments in this Part IV.D. in its Rule 50(a) and 50(b) motions. Appx1348–1373,Appx1465–1496. JMOL review standards are presented at Part IV.A., above. The Ninth Circuit reviews a district court's Copyright Act interpretations *de novo*. *Perfect 10, Inc. v. Giganews, Inc.*, 847 F.3d 657, 665 (9th Cir.), *cert. denied*, No. 17–320, 2017 WL 3782333 (U.S. Dec. 4, 2017).

networking devices and as a method of operation within Cisco's operating systems.[147] Both the individual CLI elements, and Cisco's overall compilations of features, simply reflected the functions and features of the system. Indeed, Cisco's own expert testified that Cisco's arrangement of modes and prompts asserted here is an "idea."[148] Section 102(b) precludes protection for that idea, and for every other asserted part of Cisco's CLI.

No reasonable jury could have found that Cisco's CLI elements survive the application of § 102(b), because Cisco has not proven any original creative expression separable from the CLI systems or methods of operation that are unprotectable under § 102(b). Because of their "essentially utilitarian nature," and to protect fair competition, "many aspects" of computer programs are not entitled to copyright protection. *Sega Enters. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1524–25 (9th Cir. 1992). In some cases, "even the exact set of commands used . . . is deemed functional rather than creative." *Id.*;[149] *see also* at 1039–40

---

[147] Appx10460,10464,Appx10466–10468,Appx10486(Bakan).

[148] Appx11238(Almeroth).

[149] Courts routinely deny, under various theories, copyright protection for similarly functional sets of commands and menus (as distinct from

(citing *Sega*); *Feist*, 499 U.S. at 350 (rule "severely limits the scope of protection").

This rule bars Cisco's claims. The Cisco CLI is, quite literally, a method of operating a Cisco network switch or router. As Cisco concedes, the four Cisco operating systems whose interfaces are in issue "enable the operation of [Cisco's] network switches . . . and routers."[150] Indeed, the CLI commands on which Cisco based its claim of originality, generally are not contained in the switch's software or displayed in its user interface. It is the user who types those commands on a keyboard, and the software *responds* to them. Thus, the CLI is a "method of operation" excluded from copyright protection by § 102(b).

---

the specific code implementing them). *See Ashton-Tate Corp. v. Ross*, 916 F.2d 516, 521–22 (9th Cir. 1990) (list of commands); *Dream Games of Arizona, Inc. v. PC Onsite*, 561 F.3d 983, 989 (9th Cir. 2009) (video game menu); *see also Allen v. Academic Games League of America, Inc.*, 89 F.3d 614, 617–18 (9th Cir. 1996) ("abstract rules and play ideas" for games not copyrightable); *MiTek Holdings, Inc. v. Arce Eng'g Co., Inc.*, 89 F.3d 1548, 1556–57 & n.19 (11th Cir. 1996) (menu and "sub-menu command tree structure" implementing functional steps in a process not copyrightable); *Mitel,*124 F.3d at 1373 (no copyrightable expression in system of "command codes" used in telephone systems); *Hutchins*, 492 F.3d at 1384–85 (standard CPR commands not copyrightable); *Eng'g Dynamics, Inc. v. Structural Software, Inc.*, 26 F.3d 1335, 1347–48 (5th Cir. 1994) (remanding questions about input/output formats for technical information).

[150] CB6.

Cisco cannot avoid this result by relying on its claim that the CLI is a compilation. That the CLI "may possess many constituent parts does not transform it into a proper subject of copyright protection. Virtually any process or system could be dissected in a similar fashion." *Bikram's,* 803 F.3d at 1041. Likewise, a recipe could be called a "compilation" of steps; but even if those steps "reflect the personal preferences and tastes of the recipe's author . . . the recipe would remain, in most instances, a process that is not eligible for copyright protection." *Id.*[151]

The Court's *Oracle America* decision, 750 F.3d 1339, finding certain interfaces copyrightable, does not control here for three reasons.

First, in *Oracle* it was undisputed that the declaring code and the structure and organization of the Java API packages copied by the defendant were original and possessed the requisite "minimal degree of creativity" required under the *Feist* standard. *Id.* at 1354. Here, the district court's analytic-dissection order and Instruction 39 filtered out

[151] *Bikram's* speaks in terms of processes and systems, both of which are excluded from copyrightable subject matter by the same statutory provision that excludes methods of operation. 17 U.S.C. § 102(b).

so many attributes of the CLI that only their "utilitarian or functional purpose" remained. *Id*. at 1367. *See* Part V.B.1., above.

Second, the Court held in *Oracle* that interoperability is irrelevant to copyrightability under Ninth Circuit law, *id.* at 1368–70; yet the Ninth Circuit has subsequently clarified that in drawing the line between uncopyrightable ideas, systems, and methods of operation on the one hand and copyrightable expression on the other, courts must be mindful of the balance between preserving competition and granting copyright protection. *Bikram's,* 803 F.3d at 1040. Thwarting interoperability "serves to limit competition," *Sega,* 977 F.2d at 1530, while fostering interoperability can result in "legitimate competit[ion]," *Sony Computer Entm't, Inc. v. Connectix Corp.,* 203 F.3d 596, 607 (9th Cir. 2000). In light of *Bikram* as an intervening case, the *Oracle* panel's prediction that the Ninth Circuit would hold that interoperability is irrelevant to copyrightability should be revisited.

Third, the Court found in *Oracle* that Google had "designed Android so that it would ***not*** be compatible with the Java platform," *id*. at 1371 (emphasis in original); whereas here, Cisco never tired of mentioning that Arista's use of industry-standard CLI made its

products a "drop-in replacement" for Cisco's[152] and that Arista "took what they needed" to achieve that compatibility.[153]

However, to the extent that *Oracle* presents an impediment to this argument,[154] we ask that the Court revisit the decision by holding an initial hearing en banc under Circuit Rule 35(a)(1).[155]

---

[152] CB1,17;Appx10398,Appx10412,Appx10416(Cisco's opening); Appx12716,Appx12725,Appx12732 (Cisco's closing).

[153] Appx10416(Cisco's opening);Appx12696,Appx12714,Appx12717, Appx12724(Cisco's closing).

[154] *See, e.g., Oracle*, 750 F.3d at 1357 (holding that, under Ninth Circuit law, "although an element of a work may be characterized as a method of operation, that element may nevertheless contain expression that is eligible for copyright protection.").

[155] Although rare, an initial hearing en banc "may be appropriate where (1) the court must answer a question of exceptional importance and (2) the answer will create a precedent." *George E. Warren Corp. v. United States*, 341 F.3d 1348, 1352 n.1 (Fed. Cir. 2003); *see also Lexmark Int'l, Inc. v. Impression Prods., Inc.*, 785 F.3d 565 (Fed. Cir. 2015) (ordering sua sponte that case be heard en banc under 28 U.S.C. § 46 and Fed. R. App. P. 35(a)). Scholarly reaction to *Oracle* demonstrates that both requirements are met here. *See, e.g.,* Clark D. Asay, *Software's Copyright Anticommons*, 66 EMORY L.J. 265, 303–07 (2017); Mark P. McKenna & Christopher Jon Sprigman, *What's in, and What's Out: How IP's Boundary Rules Shape Innovation*, 30 HARV. J.L. & TECH. 491, 534–35 (2017); Pamela Samuelson, *Functionality and Expression in Computer Programs: Refining the Tests for Software Copyright Infringement*, 31 BERKELEY TECH. L. J. 1215, 1252–67 (2016).

2. The jury lacked sufficient evidence to consider and compare the disputed works as a whole—or even to define their scope.

Under Ninth Circuit law, without adequate evidence of the works as a whole, the jury could not conclude that any alleged copying was actionable and its verdict cannot be sustained. "There can be no proof of 'substantial similarity' [or virtual identity] and thus of copyright infringement unless [plaintiff's] works are juxtaposed with [defendant's] and their contents compared." *Antonick*, 841 F.3d at 1066 (affirming defense JMOL after infringement verdict where complete works were not in evidence). The complete asserted works must be in evidence to support an infringement verdict. *Id.* Here, Cisco failed to put its complete works at issue (or Arista's accused works) into evidence, or even to *define* its works adequately. Thus, the jury lacked sufficient evidence to compare the works as a whole as required under *Antonick*. *Id.*[156]

Cisco's manuals did not evidence its works *as a whole*. There is no substantial evidence that the manuals contained *all* aspects of the

---

[156] During trial, the district court asked Cisco to provide a list of the exhibits that constituted Cisco's works. Appx11628–11629. Cisco never did so, even after Arista raised the issue in its Rule 50(a) motion.

operating-system user interface for any work, including but not limited to commands, modes and prompts, help descriptions, command responses (as well as all unasserted user-interface attributes). Cisco conceded in argument that the manuals did not prove "the totality of what's implemented in the product."[157] Likewise, Cisco's copyright deposits for its 26 registrations weren't evidence of the complete works because they didn't contain complete source code for any of the registered works. [158] And mere *excerpts* of source code are not evidence of the works as a whole.[159]

Notably, Cisco included on its trial-exhibit list the source code for its switch operating systems, including the source code responsible for the user interfaces that Cisco claims as the work. But Cisco never

---

[157] Appx11899.

[158] Appx11164–11166; Appx51173–51252(TX4803).

[159] Dan Lang, Cisco's sponsoring witness, testified that TX4803 is "the materials themselves that were sent to the Copyright Office, along with an index to them." Appx11168(Lang). In fact, for 18 of the 26 copyright registrations, the index in TX4803 did not refer to any linked materials at all. Appx51188-51252; Appx51173–51252(TX 4803). For seven of the eight registrations that refer to linked materials, those materials were described as documentation. And for the IOS 12.1, the sole registration that refers to "source code" as being included in the linked materials, none of those linked materials actually included any C source code, even though IOS is written in C.

offered that source code into evidence, even though the court commented before the close of evidence that, if the jury asked which exhibit contained the work, the court would be unable to answer.[160] As a result, the jury was forced to render its verdict without ever comparing the asserted work to the allegedly infringing one, and instead considered only excerpts of each. Under *Antonick*, that comparison cannot support a judgment of infringement.

3. Cisco failed to prove that its "user interfaces" were copyrighted works apart from its registered operating systems.

Cisco failed to submit substantial evidence to prove that its "user interfaces" were separable from its operating systems, as required for them to be independent copyrighted works. Cisco never separately registered its "user interface" apart from its 26 operating-system versions and offered no substantial evidence of any separate existence for its purported "user interface" works. Rather, the record confirms that the interfaces are merely non-literal elements of the operating systems. Cisco does not use, value, or even define them separately. For a work to be separately asserted, however, it must be one that "'can live

[160] Appx12636–12637(court).

[its] own copyright life' and 'has an independent economic value and is, in itself, viable.'" *Monge v. Maya Magazines, Inc.*, 688 F.3d 1164, 1180 (9th Cir. 2012) (photographs as works) (quoting *Columbia Pictures TV, Inc. v. Krypton Broad. of Birmingham, Inc.*, 259 F.3d 1186, 1193 (9th Cir. 2001) (TV-show episodes as works)). An amorphous abstraction of a software program like the "interfaces" that Cisco asserted here cannot be the "work" at issue. *See NXIVM Corp. v. Ross Inst.*, 364 F.3d 471, 480–81 (2d Cir. 2004) ("modules" of a manual not separate works); *see also Sony Computer Entm't Am., Inc. v Bleem, LLC*, 214 F.3d 1022, 1028 (9th Cir. 2000) (videogame screen shots "an insignificant portion of the complex copyrighted work as a whole"). Cisco's user interface thus has no value or "copyright life" separate from the operating systems and cannot be an independent "work."

Cisco also lacks substantial evidence that the "user interfaces" asserted here are fixed in a tangible medium of expression, as required by copyright law. 17 U.S.C. § 102(a). Cisco purports to assert an abstract work entirely separated from its operating system's source code (and consolidated across multiple separate versions of each operating system)—but the interface exists only as a function of the source code

(itself fixed in a tangible medium), as a non-literal element manifested by that code. *See Johnson Controls, Inc. v. Phoenix Control Sys.*, Inc., 886 F.2d 1173, 1175 (9th Cir. 1989), *overruled on other grounds in Perfect 10 Inc. v. Google Inc.*, 653 F.3d 976 (9th Cir. 2011). Without some grounding in source code, there is nothing fixed about Cisco's asserted works: The commands are entered by users, and the outputs and help strings are fixed in the code.[161] Moreover, Cisco presented no evidence that its source code or user interface contain the CLI commands that Arista allegedly copied. Cisco's software *responds to* those commands when a user types them, but the commands themselves are not fixed in either the code or the user interface that the code manifests.

4. Cisco failed to prove infringement at the compilation level.

As discussed above in Part V.B.5., Cisco's account of the infringement verdict relies on evidence of sub-compilation-level elements that the district court found unprotected by copyright and that Instruction 39 therefore advised the jury to disregard.[162] Cisco thus

---

[161] Appx10501–10502(Lougheed explaining CLI use).

[162] *See* CB11–12,14–15.

concedes that it failed to prove infringement at the compilation level—the very standard that Cisco espouses on appeal and that Instruction 39 required. Accordingly, the Court should affirm the judgment on that ground even if it rejects every other argument set forth above.

VI. CONCLUSION

More-than-substantial evidence supports the jury's verdict that product features, including support for existing networking protocols, dictated the selection and arrangement of Cisco's multiword command-line expressions. And multiple alternative grounds for affirmance exist as well.

Accordingly, for all the reasons stated above, the judgment should be affirmed.

Dated: December 22, 2017

Respectfully submitted,

s/Steven A. Hirsch

ROBERT A. VAN NEST
BRIAN L. FERRALL
DAVID J. SILBERT
STEVEN A. HIRSCH
MICHAEL S. KWUN
RYAN K. WONG
ELIZABETH K. MCCLOSKEY
KEKER & VAN NEST LLP
633 Battery Street
San Francisco, CA 94111-1809
Telephone: 415 391 5400
Facsimile: 415 397 7188

Attorneys for Defendant-Appellee
ARISTA NETWORKS, INC.

United States Court of Appeals
for the Federal Circuit

Cisco Systems, Inc. v. Arista Networks, Inc., 2017-2145

CERTIFICATE OF SERVICE

I, Robyn Cocho, being duly sworn according to law and being over the age of 18, upon my oath depose and say that:

Counsel Press was retained by KEKER & VAN NEST LLP, counsel for Appellee to print this document. I am an employee of Counsel Press.

On **December 22, 2017** counsel has authorized me to electronically file the foregoing **Brief for Defendant-Appellee (confidential and non-confidential versions** with the Clerk of Court using the CM/ECF System, which will serve via e-mail notice of such filing to all counsel registered as CM/ECF users, including the following principal counsel for the other parties:

Kathleen M. Sullivan
Quinn Emanuel Urquhart & Sullivan, LLP
51 Madison Avenue, 22nd Floor
New York, NY 10010
kathleensullivan@quinnemanuel.com
212-849-7000
Principal Counsel for Appellant

Additionally, the confidential version will be emailed and paper copies will also be mailed to the above principal counsel on this date. Any counsel for Amicus Curiae appearing at the time of filing will be served via CM/ECF notice of the public version only.

Upon acceptance by the Court of the e-filed document, six confidential paper copies will be filed with the Court within the time provided in the Court's rules.

December 22, 2017

/s/ Robyn Cocho
Counsel Press

CERTIFICATE OF COMPLIANCE MOTIONS OR BRIEFS CONTAINING MATERIAL SUBJECT TO A PROTECTIVE ORDER

Briefs Containing Material Subject to a Protective Order

[X] This brief complies with the limitations set forth in Fed. Cir. R. 28(d) and contains [**12**] words (including numbers) marked as confidential, or

[ ] This brief does not comply with the word count limitations set forth in Fed. Cir. R. 27(m) and a motion is requesting permission to exceed the maximum word count

Dated: December 22, 2017

s/Steven A. Hirsch
STEVEN A. HIRSCH
KEKER, VAN NEST & PETERS LLP

Attorneys for Defendant-Appellee ARISTA NETWORKS, INC.

CERTIFICATE OF COMPLIANCE WITH TYPE-VOLUME LIMITATION, TYPEFACE REQUIREMENTS AND TYPE STYLE REQUIREMENTS

1. This brief complies with the type-volume limitation of Federal Rule of Federal Circuit Rule 32(a). This brief contains 13,282, words, excluding the parts of the brief exempted by Federal Rule of Appellate Procedure 32(f).

2. This brief complies with the typeface requirements of Federal Rule of Appellate Procedure 32(a)(5) and the type style requirements of Federal Rule of Appellate Procedure 32(a)(6). This brief has been prepared in a proportionally spaced typeface using Microsoft Word 2010 in 14pt Century Schoolbook.

Dated: December 22, 2017

s/Steven A. Hirsch
STEVEN A. HIRSCH
KEKER, VAN NEST & PETERS LLP

Attorneys for Defendant-Appellee
ARISTA NETWORKS, INC.