No. 21-12835

IN THE

United States Court of Appeals for the Eleventh Circuit

APPLE INC.,

Plaintiff-Appellant,

v.

CORELLIUM, LLC,

Defendant-Appellee.

On Appeal from the
United States District Court for the Southern District of Florida,
Case No. 9:19-cv-81160-RS, Hon. Rodney Smith

BRIEF OF 15 COMPUTER SCIENTISTS IN SUPPORT OF APPELLEE AND AFFIRMANCE

Jef Pearlman
INTELLECTUAL PROPERTY &
TECHNOLOGY LAW CLINIC
UNIVERSITY OF SOUTHERN
CALIFORNIA GOULD
SCHOOL OF LAW
699 Exposition Blvd.
Los Angeles, CA 90089-0071
(213) 740-7613
jef@law.usc.edu

Counsel for Amici Curiae

CERTIFICATE OF INTERESTED PERSONS AND CORPORATE DISCLOSURE STATEMENT

I hereby certify that the following listed persons as described in Eleventh Circuit Rule 26.1-2(a) have an interest in the outcome of this case and were not listed in the Certificates of Interested Persons in briefs previously filed in this case:

1. Adida, Ben
2. Bloch, Joshua
3. Burger, Eric
4. Checkoway, Stephen
5. Durumeric, Zakir
6. Enck, William
7. Evans, David
8. Felten, Edward W.
9. Green, Matthew
10. Jones, Douglas W.
11. Kirda, Engin
12. McDaniel, Patrick
13. Pearlman, Jef: USC Gould School of Law IP & Technology Law Clinic
14. Sherr, Micah
15. Spafford, Eugene H.

16. Wagner, David

Jef Pearlman
Counsel for Amici Curiae

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Cases

Other Authorities

How to make right-click on Mac and in Parallels Desktop virtual machine, Parallels (Sep. 4, 2019), https://kb.parallels.com/fr/9151 [https://perma.cc/9NF4-YBFF] 9

Jamie Mercer, *Java Turns 22 Today*, I Programmer (May 23, 2017), https://www.i-programmer.info/news/80-java/10791-java-turns-22-today.html [https://perma.cc/R5QN-RWUR] 6

Matthew Tyson, *What is the JVM? Introducing the Java Virtual Machine*, InfoWorld (Jan. 17, 2020), https://www.infoworld.com/article/3272244/what-is-the-jvm-introducing-the-java-virtual-machine.html [https://perma.cc/5D2D-KV7X] 6

Oracle VM VirtualBox Overview, Oracle. (June 2021), https://www.oracle.com/assets/oracle-vm-virtualbox-overview-2981353.pdf [https://perma.cc/86FY-UGZA] 18

Refurbished iPhone, Apple (last visited Feb. 14, 2022), https://www.apple.com/shop/refurbished/iphone [https://perma.cc/L9PD-97XQ] 12

Sean Conroy, *History of Virtualization*, I Don’t Know, Read the Manual (Jan. 25, 2018), https://www.idkrtm.com/history-of-virtualization/ [https://perma.cc/4Z9J-TDV3] 5, 6, 7

Shyam Sundar Ramaswami, *Picture perfect: How JPG EXIF data hides malware*, Cisco Umbrella (October 28, 2021), https://umbrella.cisco.com/blog/picture-perfect-how-jpg-exif-data-hides-malware [https://perma.cc/7E5K-5FWE] 14

Steve Baca, *Virtualization for Newbies: Five Types of Virtualization*, Global Knowledge (Nov. 29, 2021), https://www.globalknowledge.com/us-en/resources/resource-library/articles/virtualization-for-newbies-five-types-of-virtualization/ [https://perma.cc/9QUP-6RYR] 4

Thiago Alves et al., *Virtualization of SCADA testbeds for cybersecurity research: A modular approach*, ScienceDirect (Aug. 2018), https://www.sciencedirect.com/science/article/pii/S0167404818304905 [https://perma.cc/GW48-US6T] 15

Thomas J. Trappler, *Software licensing in the cloud*, Computerworld (Apr. 18, 2013), https://www.computerworld.com/article/2496855/software-licensing-in-the-cloud.html [https://perma.cc/JW3Y-V7TZ] 19

INTEREST OF AMICI CURIAE[1]

Amici are 15 computer scientists, academics, and technologists with experience in virtualization, including its many research and commercial uses. We have included a full list of amici along with titles and affiliations, for identification purposes only, in the Appendix. Amici file this brief to help the Court understand the history and broader context of virtualization technology. Amici also seek to highlight the risks of finding that developing and offering tools based on that technology is infringing.[2]

STATEMENT OF ISSUES

1. Whether the district court properly found that making and selling a virtualization tool qualifies as fair use under 17 U.S.C. § 107.

SUMMARY OF ARGUMENT

Plaintiff Apple Inc. (Apple) alleges that Corellium, LLC (Corellium) has infringed copyrights in iOS and some of the images embedded in iOS[3] by selling a

[1] No party or party's counsel authored this brief in whole or in part or contributed money that was intended to fund preparing or submitting this brief. No one other than amicus and its counsel contributed money that was intended to fund preparing or submitting this brief. Pursuant to Fed. R. App. P. 29(a)(2), all parties have consented to the filing of this brief.

[2] Amici thank USC Gould School of Law IP & Technology Law Clinic students Emmanuel Hamidi, J.D. Candidate '23 and Martin Yi, J.D. Candidate '23, for their valuable contributions to this brief.

[3] For simplicity, we will refer to this group of works simply as "iOS."

virtualization tool that works with that software. The district court correctly found that to the extent Corellium's software requires copying iOS, such copying is fair use.

The implications of this case, however, reach far beyond a single security testing tool. Virtualization is a powerful, flexible technological tool that is reliant on use of copyrighted software, is inherently transformative, and is non-substituting. It has been used since the 1960s and has become a critical tool in modern, commercial computing. Should the court hold that this tool is not fair use, the legal risk and uncertainty created could encompass many common, broadly used technologies that have long been understood to be lawful, and that are necessary infrastructure for constructing a secure internet.

Virtualization encompasses a broad set of technologies with complex capabilities and many different underlying approaches. Amici's goal is to assist the Court by highlighting some key features of virtualization relating to this case in understandable language. This means that our descriptions, while lacking some of the internal complexity in some cases, will still be technically accurate throughout. Though there are other kinds of virtualization, we will focus on software virtualization as that is the most relevant inquiry in this case.

Amici respectfully request that this court affirm the decision below and not open the door to unproductive copyright litigation that would undermine such a critical technology.

ARGUMENT

Corellium's product (the Product) implements a technology known as virtualization. This flexible technology has many well-established uses, many of which rely on use of copyrighted software. Much like general purpose computers themselves, virtualization tools enable the use of third-party software, but they generally should not and do not require the permission of software developers to sell the tools themselves. Should the court find there is no fair use here, it would create legal uncertainty over the entire field of virtualization and threaten multiple areas of research and industry.

I. VIRTUALIZATION IS AN IMPORTANT, FLEXIBLE, AND LAWFUL TECHNOLOGY WHICH HAS BEEN INTEGRAL TO COMPUTING FOR OVER HALF A CENTURY.

Virtualization, also known as "virtual machine emulation," is a widespread technology that is not just used to create virtual phones, but also to enable cloud computing, enhance hardware efficiency, allow cross-platform software usage, and test the security of the systems we rely on every day. These virtualization tools are lawfully built without the permission of the owners of the copyrights in the software that can be virtualized.

A. *Virtualization is a tool for simulating the hardware or software necessary to enable existing software to function in new contexts and new ways.*

Virtualization is an essential tool in computing industries. It allows software to simulate an existing computer system, including much of that system's hardware and software, within a new, "virtualized" environment. This simulation may have many different applications, from allowing an operating system to run on new hardware for which it was not designed to allowing a single physical machine to run many "virtual machines" at one time. *What is virtualization?*, IBM (Dec. 15, 2021, 4:47pm), https://www.ibm.com/topics/virtualization, [https://perma.cc/9PWS-PYQN]; Steve Baca, *Virtualization for Newbies: Five Types of Virtualization*, Global Knowledge (Nov. 29, 2021), https://www.globalknowledge.com/us-en/resources/resource-library/articles/virtualization-for-newbies-five-types-of-virtualization/ [https://perma.cc/9QUP-6RYR]. *See also* Dkt. 784, pg. 8 ("Virtualization is the ability to run software on hardware it is not ordinarily meant to run on.").

This feat is accomplished by simulating a computer's hardware and other systems that are present in the original system but not in the new one. This allows software designed for one environment to execute on another environment, without the need to modify the software. Critically, a virtualization tool does not recreate the original software—it simply provides an environment in which that software may function correctly. Here, Corellium's Product (the virtualization software) uses virtualization to simulate, on non-Apple hardware, an environment that can run Apple's iOS operating system software (the software being virtualized), which was originally designed to run on Apple hardware. *See* Dkt. 784, pg. 12 (describing loading Apple-provided copies of iOS into Corellium's Product).

B. Virtualization is a widely accepted technology that has been in use since the 60s.

Virtualization has long been a crucial technology in computing. After intensive research through the 1960s, IBM publicly introduced the first commercial mainframe to support a virtual machine environment in the form of the System/360 Model 67. Sean Conroy, *History of Virtualization*, I Don't Know, Read the Manual (Jan. 25, 2018), https://www.idkrtm.com/history-of-virtualization/ [https://perma.cc/4Z9J-TDV3]. This machine had virtualization software that allowed multiple users to each run their own virtualized, interactive operating system. *Id.* While today we assume we will be able to interact with a computer directly, at the time this was a technological leap forward, adding significant new capabilities:

> The User interaction portion is important. Before this system, IBM focused on systems where there was no user interaction. You would feed your program into the computer, it would do its thing; then spit out the output to a printer or a screen. An Interactive Operating System meant you actually had a way of interacting with the programs while they ran.

Id. The virtualization also enabled multiple users to efficiently share resources and made sure no user could crash the entire system or someone else's operating system running on the same system.

Over time, new forms of virtualization emerged. In 1990, Sun Microsystems began a project that eventually, in 1995, became what we now call "Java." Jamie Mercer, *Java Turns 22 Today*, I Programmer (May 23, 2017), https://www.i-programmer.info/news/80-java/10791-java-turns-22-today.html [https://perma.cc/R5QN-RWUR]. Java was designed to use a common virtual machine that would run the same, virtualized software on many kinds of host machines, enabling new forms of cross-platform development. Matthew Tyson, *What is the JVM? Introducing the Java Virtual Machine*, InfoWorld (Jan. 17, 2020), https://www.infoworld.com/article/3272244/what-is-the-jvm-introducing-the-java-virtual-machine.html [https://perma.cc/5D2D-KV7X]. This removed the need to adapt software to new environments, saving a significant amount of development effort—historically, switching environments might require software to be rewritten from scratch.

Throughout the late 90s and 2000s, hardware virtualization capabilities were added to personal computers, and desktop virtualization for consumers, discussed below in Section II, became a reality. Sean Conroy, *History of Virtualization*, I Don't Know, Read the Manual (Jan. 25, 2018), https://www.idkrtm.com/history-of-virtualization/ [https://perma.cc/4Z9J-TDV3]. This enabled enterprise users to launch multiple virtualized machines in place of physical hardware, greatly reducing enterprise IT expenditure and simplifying the deployment of new infrastructure.

Over time, virtualization has developed into a mature industry in which the developer of the virtualization tool, the developer of the virtualized software, and the end-user are generally three different parties. Virtualization companies build and offer these tools for use with software from other developers, and the end-user—not the creator of the tool—decides which software to install and run.

Today, continuing over 50 years of research and commercial use, virtualization remains a powerful, valuable tool for purposes including server consolidation, cloud computing, and security research.

C. Partial copying is technically infeasible because virtualization requires access to code in its entirety.

Virtualization functions by providing a simulated environment to create a virtual computer system. *What is Virtualization?*, vmware, https://www.vmware.com/solutions/virtualization.html [https://perma.cc/9P7J-8N7G] ("Virtualization relies on software to simulate hardware functionality and create a virtual computer system."). When the goal is to virtualize an entire system, like an iPhone, the environment is designed to behave identically to the original target hardware. Once this environment has been provided, the original operating system software can execute within the environment in an entirely unmodified form.

Utilizing a virtualization technology comprises two separate actions, only one of which involves the software itself. First, a "virtual machine" is designed. This machine does not contain the original software and is simply designed to provide a correct environment for that software to run. Second, a copy of the original software itself must be installed within the virtual machine. It is our understanding that for the Product, consistent with industry practice, Corellium provides only the virtual environment and does not provide copies of Apple's software. Instead, that software is "loaded by the user." Dkt. 784, pg. 1; *see also* Dkt. 784, 11-12.[4] The end user knows the specifics of the software they intend to run, including the version, hardware model, etc., and that user provides a copy of the software to be virtualized in the way such a system typically expects—as a whole.

Furthermore, since the goal of a virtualized environment is to enable the execution of the original software without loss of functionality, modification of the virtualized software more than necessary, including removing images or attempting to use just a small subset of features, *see* Apple Brief at 16, is undesirable. Indeed, such modifications would be technically challenging and likely to cause the virtualized software to behave differently—and often incorrectly. Because the goal

[4] Because amici are focused on virtualization in general and not solely the Product, we explain general industry practice, wherein the end user provides the software to be virtualized. However, because building, offering, and using these tools is fair use, who performs the act of copying should be irrelevant.

is for that software to behave as if it is running in its original context, this undermines the core purpose of virtualization.

D. Virtualization and physical devices are not substitutes for one another.

Virtual devices and physical devices each have inherent limitations and advantages that mean they do not effectively substitute for each other.

1. Virtual devices cannot replace physical devices.

A virtual device cannot simply replace a physical device because virtual devices typically lack some of the physical components necessary for real-world functionality. Having a virtual smartphone on a server is not like having a real one in your pocket. Similarly, a virtualized server might have a different amount of memory and lack a keyboard and mouse when compared to a physical server. A virtualized desktop might have different video capabilities, memory, or disk space, or even lack a second mouse button. *See, e.g.*, *How to make right-click on Mac and in Parallels Desktop virtual machine*, Parallels (Sep. 4, 2019), https://kb.parallels.com/fr/9151 [https://perma.cc/9NF4-YBFF] (explaining how to simulate a right mouse button on a host system that lacks one).

As the district court recognized, Corellium's Product has such limitations, lacking the physical camera and phone features. Dkt. 784, pg. 1 ("users cannot make phone calls or use camera [functions]"). Importantly, this means that the Product is incapable of simulating these features. While the Product may enable execution of iOS code that provides for camera and phone call functions, the executed code does not actually perform the original functions because the Product does not include a physical camera or speaker that can take real world signals and convert them into usable information.

Again, this is because the Product virtualizes the entirety of Apple's iOS code as a necessity but does not require every function of a physical iPhone to be fully realized because the Corellium Product serves a different purpose: research. Fundamentally, virtual devices like those offered by Corellium do not offer the qualities of physical ownership embodied in a physical device. Much as a picture of a diamond ring is not nearly as valuable as a real one, so too is a physical iPhone more valuable to a consumer than a virtualization of one. One cannot keep the product in their pocket, take it out to make a phone call, or snap a vacation photo with the Product.

2. Physical devices cannot replace virtualized systems.

The reverse is true, too: physical devices cannot substitute for virtual ones because they lack crucial functions provided only by virtual devices. Typically, physical devices are immutably optimized for end users and, as such, cannot be customized in ways that change the underlying software. Virtual devices, on the other hand, offer customizability that allows for significant optimization for whatever purpose for which the virtual device is used—generally a very different use than that of the average consumer.

Because virtual devices are encapsulated in another system, they can be controlled, studied, and manipulated in ways that a physical device cannot. As the district court noted, the Product allows users to:

> (1) see and halt running processes;
> (2) modify the kernel;
> (3) use CoreTrace, a tool to view system calls;
> (4) use an app browser and a file browser; and
> (5) take live snapshots

Dkt. 784, pg. 21. This degree of customization is essential for functions such as security research and software debugging, but it is simply not available on a physical device. Hence these features allow researchers and other users to test functionality and security in new, more efficient, or even otherwise impossible ways.

In addition to technical infeasibility as outlined above, Apple's suggestion to use racks of iPhones instead of the Product is also economically and practically infeasible. In practice, many deployed Apple smartphones are older models, and understanding the behavior of these devices is essential to evaluating the impact of new security vulnerabilities. Unfortunately, Apple typically does not even sell significantly older models of its iPhones and tracking down older models via secondhand vendors would be prohibitively cumbersome. *Refurbished iPhone*, Apple (last visited Feb. 14, 2022), https://www.apple.com/shop/refurbished/iphone [https://perma.cc/L9PD-97XQ] (refurbished models only go as far back as the iPhone 8). As the district court noted, the Product permits the user to change the system to behave as different iterations of the iPhone e.g., an "iPhone 11 Max running iOS 13." Dkt. 784, pg. 12. This, in turn, allows researchers to compare different models and software versions under the *exact* same conditions—something one cannot do with a physical phone. Moreover, the Product is easily shared across multiple users of a team, whereas a rack would present portability issues—especially if the research team is global.

Virtual phones and physical phones are simply not the same thing. They do not serve the same purposes, and each offers capabilities the other does not.

II. TODAY, VIRTUALIZATION IS AN ESSENTIAL TOOL THAT MUST NOT BE HOBBLED BY MISUSE OF COPYRIGHT.

While this case is focused on the Corellium Product, the arguments that Corellium infringes copyright by offering a virtualization tool have implications for a broad variety of lawful products and services that currently benefit the public.

A. Virtualization is a critical technology to both the public interest and commercial spheres.

Virtualization is not merely a tool for testing phones. It enables critical cybersecurity research, powers cloud computing, and streamlines use of hardware computing resources.

1. Cybersecurity

Virtualization is a valuable tool for cybersecurity research specifically because it allows testers to test code in controlled, discrete environments with different capabilities than "in the wild." In malware detection, for instance, a researcher may create a "sandbox"—an isolated virtual environment—in which to execute suspicious code. This sandbox allows the researcher to run the code without fear of compromising the entire computing system because any malicious effects would be limited to the entirely virtual sandbox. Chris Greamo & Anup Ghosh, *Sandboxing and Virtualization: Modern Tools for Combating Malware* 79 (2014), https://ieeexplore.ieee.org./document/5739643 [https://perma.cc/2AGF-GPDV]. This is analogous to the way that medical researchers evaluate new, contagious organisms in a controlled environment, ensuring that disease does not spread into the population.

JPG image file malware and SCADA cybersecurity research provide two clear examples of the value of virtualization. JPG files can contain malware that otherwise wouldn't be detectable in a normal environment. *'Hackin with Pictures': Stegosploit and How to Stop It*, OPSWAT (Aug. 2, 2017), https://www.opswat.com/blog/hacking-pictures-stegosploit-and-how-stop-it [https://perma.cc/TEN2-XA8A]("The remarkable thing is that the malware is inserted into an image and the image still *looks* harmless. For that reason, it is difficult to detect and block this kind of attack."). One particular type of JPG malware, known as LokiBot, is particularly sinister because it "steal[s] information, act[s] as a keylogger, and [] establish[es] backdoors in Windows systems to both maintain persistence and send stolen data to the attacker." Charlie Osborne, *LokiBot malware now hides its source code in image files*, ZDNet (Aug. 7, 2019), https://www.zdnet.com/article/lokibot-information-stealer-now-hides-malware-in-image-files/ [https://perma.cc/D6JL-GC4Q]. The malware initially avoided detection because JPG images weren't typically considered as a malware risk. Shyam Sundar Ramaswami, *Picture perfect: How JPG EXIF data hides malware*, Cisco Umbrella (October 28, 2021), https://umbrella.cisco.com/blog/picture-perfect-how-jpg-exif-data-hides-malware [https://perma.cc/7E5K-5FWE] ("We ordinarily don't assume that the .JPG itself is the vector for malware."). However, in a specifically configured sandbox, researchers were able to identify a JPG image as an instance of LokiBot. *Id.* ("But when we analyzed the image file through a sandbox environment configured differently than the first, the service identified the image as a trojan.").

System Control and Data Acquisition (SCADA) systems also rely on virtualization to enable threat detection. SCADA systems manage and monitor critical infrastructure such as power plants, water distribution systems, and nuclear energy systems. "The extremely high cost and critical nature of SCADA systems has made it nearly impossible for researchers to perform experiments with live cyber-attacks. Hence, replicating the behavior of these complicated systems [via the low cost and portable solution provided by virtualization] provides researchers with the necessary workspace to combat the threats currently haunting these legacy systems." 77 Thiago Alves et al., *Virtualization of SCADA testbeds for cybersecurity research: A modular approach*, ScienceDirect (Aug. 2018), https://www.sciencedirect.com/science/article/pii/S0167404818304905 [https://perma.cc/GW48-US6T]

2. Cloud Computing: Infrastructure as a Service

Virtualization is also one of the core technologies underlying the explosion of cloud computing, including general cloud computing services like Amazon Web Services (AWS) and the Software as a Service (SaaS) business model. AWS, like competing services such as Microsoft Azure and Google Cloud, uses virtualization to offer a wide variety of services.[5] *Cloud computing with AWS*, Amazon Web Services (Dec. 21, 2021, 10:35AM) https://aws.amazon.com/what-is-aws/ [https://perma.cc/EBW4-R3MF] ("Amazon Web Services (AWS) is the world's most comprehensive and broadly adopted cloud platform, offering over 200 fully featured services . . .").

[5] Although we focus on Amazon AWS, the descriptions below apply equally to competing services from Microsoft, Google, and other companies.

Amazon EC2, one of the AWS services, allows customers to create virtual servers in which users can run "instances" of their applications within AWS. This approach is known as "infrastructure as a service" (IaaS) because customers are paying for virtual server infrastructure but are responsible for providing the software to be virtualized. By virtualizing a server, AWS allows users to customize their virtual workspace in terms of CPU, storage, memory, and networking resources, creating "a wide selection of instance types optimized to fit different use cases." *Amazon EC2 Instance Types*, Amazon Web Services, https://aws.amazon.com/ec2/instance-types/ [https://perma.cc/TUW4-UWMT]. Critically, because these servers are virtualized, customers can adjust on the fly to suit their needs. For example, rather than building a massive physical server infrastructure to handle "spikes" where usage is extremely high for a short period, AWS customers can purchase temporary additional virtual servers from Amazon to meet the demands of their own customers. These configurations are crucial to users who may not have the computing capacities offered by AWS, and they allow for an ideal workflow for their needs, as well as letting Amazon allocate its hardware more efficiently.

3. Cloud Computing: Software as a Service

Software as a Service (SaaS) also uses virtualization, although here the advantage is providing users access to cutting edge software applications for their work. Chadni Babu, *Saas-Based Enterprise Virtualization and Applications*, ThinkPalm (April 9, 2020), https://thinkpalm.com/blogs/saas-based-enterprise-virtualization-applications/ [https://perma.cc/9FNX-CF6E] ("SaaS clearly is an application instance of virtualization."). Rather than install a copy of software on their laptop or desktop machine, a user opens applications that are hosted by software providers in the providers' own servers, using the providers' computing resources instead. Wesley Chai, *Software as a Service (SaaS)*, SearchCloudComputing (Feb. 2021), https://searchcloudcomputing.techtarget.com/definition/Software-as-a-Service [https://perma.cc/QS4P-CS2E]. In many cases, rather than owning servers, the providers are themselves using IaaS from *another* provider to adjust to customer demand on the fly.

For SaaS providers, key advantages include instantaneous rollout of application updates, defenses against unlawful copying, and alternative payment structures like subscriptions. *See, e.g.*, *What Is SaaS?*, CompTIA (Dec. 21, 2021), https://www.comptia.org/content/articles/what-is-saas [https://perma.cc/X2YB-4N99]. For users, SaaS provides the advantages of easy software setup, seamless support, no hardware setup or maintenance, accessibility on multiple devices, and customizable usage and payment. *Id.* Industry standard applications such as Google Workspace and Microsoft Office are often implemented partially or fully via the SaaS model.

4. Desktop Virtualization

Desktop virtualization tools such as VirtualBox, Parallels, and VMWare also rely on virtualization to provide IT solutions to businesses and consumers alike. VirtualBox is a "cross-platform virtualization platform [that] allows users to extend their existing computer to run multiple operating systems . . . at the same time." *Oracle VM VirtualBox Overview*, Oracle. (June 2021), https://www.oracle.com/assets/oracle-vm-virtualbox-overview-2981353.pdf [https://perma.cc/86FY-UGZA]. The platform is "ideal for testing, developing, demonstrating, and deploying solutions across multiple platforms from one machine." *Id.* One such solution is "allow[ing] software QA teams to control source code, share it within the company and execute software testing on multiple platforms on one unique device." *Id.* VirtualBox also is largely open source and available for free to end users, meaning users are free to modify the tool itself to offer whatever features the customer needs. *Welcome to VirtualBox.org!*, VirtualBox (Dec. 21, 2021), https://www.virtualbox.org/ [https://perma.cc/LYF8-EM9Y]("[VirtualBox] is also the only professional solution that is freely available as Open Source Software").

Competitors such as VMware offer similar virtualization-based solutions. For example, it is possible to run full Windows and Linux installations virtually within an OS X machine, allowing developers to test all three platforms without buying three physical machines. *VMware Fusion*, VMware (Dec. 21, 2021), https://www.vmware.com/products/fusion.html [https://perma.cc/4MNT-3JJE]. This technology simply would not be possible without virtualization.

Critically, both the desktop and cloud-based versions of these systems allow the end-user to install a copy of the software that they wish to virtualize, such as an operating system image. The systems allow the user to make the decision about which software to install—just as a real physical machine would—and do not check that the operating system is a licensed or even a particular version. Not only would this be a nearly impossible task given the complexity of operating system development and licensing, but it is well understood that the end user—not the provider of these powerful and flexible tools—bears responsibility for that task. Thomas J. Trappler, *Software licensing in the cloud*, Computerworld (Apr. 18, 2013), https://www.computerworld.com/article/2496855/software-licensing-in-the-cloud.html [https://perma.cc/JW3Y-V7TZ].

B. A finding against fair use could severely impair a broad array of tools used in crucial public and private systems.

If execution of copyrighted software on virtualization platforms is not considered fair use, then entire well-established services, tools, and industries may face the threat of crippling liability. Finding the developer of a virtualization tool liable for all possible uses of that tool may undermine development of virtualization systems that serve the public interest or serve as crucial building blocks in business and consumer computing.

The tension between copyright protection and fair use is at its core a tension between incentives for initial creators versus access for subsequent creators and users. 9 David W. Barnes, *The Incentives/Access Tradeoff* 96 (2010), https://scholarlycommons.law.northwestern.edu/cgi/viewcontent.cgi?article=1121 &context=njtip [https://perma.cc/WRY5-FL7Q] ("Courts, Congress, and commentators acknowledge that intellectual property law is a balance between providing incentives through exclusive rights and encouraging use of information through free access to creative works."). The fair use doctrine represents Congress's intent to ensure that while copyright does provide the incentive to create, it does not hobble socially beneficial, non-substituting uses of creative works by third parties. In other words, fair use doctrine has always been intended to allow access to copyrighted content for useful purposes that would otherwise be unnecessarily inhibited by copyright law.

With the relatively recent expansion of computing and networking, fair use often focuses on ensuring the ability of software developers to create tools that *use* copyrighted works, but do not supplant them. Software virtualization fits squarely within that intent because it serves the useful, transformative purpose of utilizing existing software and systems in new contexts, whether it be cybersecurity research or virtual workspaces. To find that fair use does not protect these developers would significantly impair the valuable, well-settled applications of virtualization illustrated above by throwing them into a sea of uncertainty.

III. DEVELOPING AND SELLING THIRD-PARTY VIRTUALIZATION TOOLS LIKE THE CORRELIUM PRODUCT IS FAIR USE.

To the extent that building and offering virtualization tools involves copying of copyrighted software, it is fair use. While we may touch on the details of the Product or the specifics of cybersecurity research, which have been addressed by Corellium and Amici Computer Security Research, EFF, and Public Knowledge (Security Researcher Amici), our focus is virtualization more broadly. While an individual could possibly infringe copyright in the process of using such a tool—just like it is possible to infringe copyright with a general-purpose computer—the building and distribution of tools that enable transformative, non-substituting, fair uses of copyrighted is itself fair use.

A. The development, sale, and use of virtualization software is generally transformative and only slightly commercial.

The purpose and character of use of virtualization software is inherently transformative, strongly favoring fair use. It is almost never the case that virtualization tools are used to allow someone to run the same software they already run in the same circumstances they already run it. The entire point is that they "provide[] a new collection of tasks operating in a distinct and different computing environment." *Cf. Google LLC v. Oracle Am.*, 141 S.Ct. 1183, 1203 (2021).

As explained by Security Researcher Amici, security research is one significant, transformative use made by users of virtualization tools like the Product. Security Researcher Amicus at 20-21. As Corellium points out, studying the functionality of consumer products, reverse engineering software, security research, and other forms of research are similarly transformative. Corellium Brief at 20-29. These are just examples of a broader point: virtualization technology in general offers transformative uses of existing software; it does not merely repackage existing technology to serve the same purpose.

The crux of virtualization is that it offers the ability to run old software in a new context. While virtualization tools do much more, even this basic act is transformative. For example, in *Sony Comput. Entm't, Inc. v. Connectix Corp.*, 203 F.3d 596, 599, 606 (9th Cir. 2000)*,* a similar technology, emulation, that allowed a video game to be played on a different hardware system was a transformative fair use as it created a "wholly new product" by altering the environment in which consumers could play games initially designed for an entirely other product. Virtualization achieves the same thing.

Many virtualization tools go beyond this, adding even more transformative capabilities. For instance, IaaS allows virtual devices to be created and decommissioned on the fly, saving energy and costs. Virtual desktops permit users to turn one type of computer into another, or even into more than one computer at once. Sandboxing tools allow potentially dangerous software to safely run and be tested without harming any computers. And tools like the Product enable research to verify and improve the security of systems we rely on daily. All of these uses are transformative.

While virtualization tools are sometimes commercial, weighing against fair use, this first factor overall remains in favor. As the Supreme Court recently observed, commerciality "is not dispositive of the first factor," particularly with "inherently transformative" uses. *Google v. Oracle*, 141 S.Ct. at 1204. Virtualization is inherently transformative, as are the research and educational uses that virtualization helps to facilitate. And in many cases, commercialization is necessary to make such a transformative technology feasible, as virtualization systems are complex to design and develop.

Regardless of whether the technology is sold for profit, virtualization facilitates computer science education, research, efficiency, and other transformative uses. This ultimately supports the goals of copyright to promote the progress of science and the useful arts. *Id.* at 1203. As a result, this factor favors fair use.

B. Because software is primarily functional, the second factor does not weigh against fair use.

The second factor, while not usually having a determinative role in fair use determinations, favors virtualization technologies, which extend and transform the functionality of existing software rather simply allowing use of creativity embodied by that software. As EFF and Corellium have pointed out, computer software, while subject to copyright, is further from the core of copyright than other types of works, particularly when it is the utilitarian aspect of software and not its expressive aspect that is being reused. *See* Corellium Brief at 32-33; Security Researcher Amicus at 21.

This is the case with virtualization technologies in general, whose purpose is to build new functionality on top of existing functionality—not to provide the same consumptive experience that is available from the creator of the virtualized software. When viewed properly, even the inclusion of visual works like icons and wallpaper do not undermine the second factor's favoring fair use. Even where software includes visual displays, those displays are part of the functionality. *See, e.g.*, *Sega Enters. v. Accolade, Inc,* 977 F.2d 1510, 1524 (9th Cir. 1992). The appearance of the operating system—what is being transformed to a new context by virtualization—is a function of the operating system's code itself.

Virtualization is generally not used as a costly, indirect way to access the creative value of either software or visual works; it is used to create an accurate representation of the virtualized system while expanding its functionality. *Cf. Lexmark Int'l, Inc. v. Static Control Components, Inc.*, 387 F.3d 522, 544 (6th Cir. 2004). This factor therefore favors fair use.

C. The amount and substantiality used is consistent with fair use because it is necessary to use an entire work to virtualize software.

As discussed above, it is generally necessary to use the entire work when virtualizing software for any purpose, including reverse engineering, security testing and research, or offering a new environment. *See supra* Section I.C. Corellium and Security Amici have explained why it is necessary to use the entire work to do security research and offer the Product to researchers. Security Researcher Amicus at 21-22; Corellium Brief at 37-42. The principle applies more broadly to virtualization tools in general. Because using the full work is reasonable for the transformative purpose, this factor does not weigh against fair use.

The purpose of virtualization is to accurately simulate a device's behavior in a new context, whether that is for enabling that software to run in the cloud on EC2 or Azure, allow software to safely run in a sandbox to ensure it is free of malware, or allow observation, security testing, and reverse engineering of phone software as the Product does. Partial software does not act the same as full software, and these transformative purposes cannot be realized with partial copying. The copying necessary to enable virtualization is therefore "tethered to a valid, transformative purpose," *Google v. Oracle*, 141 S.Ct. at 1205 (citing *Campbell*), and the third factor therefore favors fair use.

D. Virtualization tools have no relevant effect on the market, favoring fair use.

For the fourth factor, the district court correctly focused on the core question: "whether the copy brings to the marketplace a competing substitute for the original, or its derivative, so as to deprive the rights holder of significant revenues because of the likelihood that potential purchasers may opt to acquire the copy in preference to the original." *Authors Guild v. Google Inc.*, 804 F.3d 202, 223 (2d Cir. 2015). As discussed above in Section I.D.1, virtualized software tools rarely, if ever, substitute for the original uses. That is a core purpose of virtualization: to allow something new, whether that is a careful probing of the properties of existing software or making more efficient use of hardware. In the cases where a virtualization tool might be flexible enough that an end-user could make a substituting use, a license may be necessary, but that does not affect the question of whether the tool's development and sale is fair use.

As Security Amici have explained, security research on iOS does not substitute for Apple's commercial use of iOS. Security Researcher Amicus at 22-24. Corellium further explains why this is true of the Product in general. Corellium Brief at 42-49. As with the first three factors, the same is true of virtualization technology in general, and a contrary finding would undermine the use of all those technologies.

Virtualization of mobile operating systems such iOS poses a particularly low risk of substitution because the virtualized system is not an alternative for consumers in the market for mobile phones. In *Google v. Oracle*, 141 S.Ct. at 1207, the court recognized that even markets for different *types* of phones would be an "important difference" which could lead a jury to find no market substitution. The Product is even further from the consumer market, as the Product does not even offer a phone and iOS itself is freely available directly from Apple.

More broadly, virtualization tools generally do not substitute the market for the copyrighted work itself. Consumers generally do not want to pay for an additional, likely more expensive tool to let them do a thing they can already do. The added capabilities of a virtualized system, whether they are security analysis features, sandboxing, or allowing multiple instances, are targeted at particular, new markets because, even when done well, they do not perfectly simulate the consumer experience. The value of these tools comes from offering additional capabilities to specific new markets like security researchers, not providing an alternative consumer product.

As the district court notes, even if the operating system developer entered the security research market with their own device program, a copyright in the program should not confer a monopoly over the security research market. Dkt. 784, pg. 31. But this is exactly what Apple argues: that the Product substitutes for its "iOS

Simulator" product and for buying and using "racks of iPhones." Apple Brief at 46-48. Neither is correct, because the market for the tools is *separate* from the work allegedly infringed here. Those tools do not derive their value from an operating system license. They are not useful because they make a creative, user-friendly, or "aesthetically attractive" phone operating system available to a user, *see* Apple Brief at 6, but from the fact the fact that they can interact with iOS in a way that replicates real-world use *by others*.

And neither iOS Simulator nor any number of physical phones substitute for the Product. As discussed above, see Section I.D.2, virtualized phones can be used and tested in ways that physical ones cannot. Both Apple's "customized iPhone devices" and "iOS Simulator" may compete with the Product, but they inevitably offer different capabilities because they are offered by the party whose interests may align differently with researchers or the public. Competition in such tools is critical.

Fair use means permissionless use. It cannot turn on whether a similar tool making a similar use is available only "to legitimate security researchers" *in the eyes of the copyright holder*. *See* Apple Brief at 46. This is merely an attempt to do an end-run around fair use by creating different, competitive service and using copyright to limit it for business purposes. Much as a filmmaker cannot limit the use of short clips to whomever they view as "legitimate film critics," the fair use inherent in virtualization tools must be recognized independent of the existence of a competing tool offered by the copyright owner. The tool and the work are two different things.

Because virtualization technology is so broad and powerful, there are some situations in which the virtualized OS might substitute for the original. For instance, in virtualizing a desktop OS on a different desktop OS, in some cases a user may use that for a similar purpose to the original OS. In those rare situations, however, it is the user's responsibility to obtain a license; the developer of the virtualization tool cannot control or even realistically evaluate every possible end use. More importantly, those situations are irrelevant to the case at hand, as the Product simply is not a substitute for iOS on a phone. Because the Product, like most virtualization tools, does not create a substitute for the original work or supplant its value, this factor also favors fair use in creating virtualization tools.

CONCLUSION

As explained above, the implications of this case go far beyond the copyrighted works and research tools at issue here. Fair use protects not only the

Product, but a broad array of important, well-established virtualization tools. Amici therefore respectfully request that the Court affirm the decision below.

February 16, 2022

Respectfully Submitted,

Jef Pearlman
INTELLECTUAL PROPERTY &
TECHNOLOGY LAW CLINIC
UNIVERSITY OF SOUTHERN
CALIFORNIA GOULD
SCHOOL OF LAW
699 Exposition Blvd.
Los Angeles, CA 90089-0071
(213) 740-7613
jef@law.usc.edu

Counsel for Amici Curiae

February 16, 2021

APPENDIX — LIST OF AMICI

Amici sign this brief on their own behalf, not on behalf of the organizations with which they are affiliated; affiliations are listed for reference only.

1. Ben Adida
   Executive Director
   VotingWorks

2. Joshua Bloch
   Professor of the Practice in Computer Science
   Carnegie Mellon University

3. Eric Burger
   Research Professor of Computer Science
   Georgetown University

4. Stephen Checkoway
   Assistant Professor, Department of Computer Science
   Oberlin College

5. Zakir Durumeric
   Assistant Professor of Computer Science
   Stanford University

6. William Enck
   Professor, Department of Computer Science
   North Carolina State University

7. David Evans
   Professor of Computer Science
   University of Virginia

8. Edward W. Felten
   Robert E. Kahn Professor of Computer Science and Public Affairs, Emeritus
   Princeton Universityc

9. Matthew Green
   Associate Professor, Department of Computer Science
   Johns Hopkins University

10. Douglas W. Jones
    Emeritus Associate Professor of Computer Science
    University of Iowa

11. Engin Kirda
    Professor of Computer Science and Computer Engineering, Khoury College of Computer Sciences
    Northeastern University, Boston

12. Patrick McDaniel
    William L. Weiss Professor of Information and Communications Technology, School of Electrical Engineering and Computer Science
    The Pennsylvania State University

13. Micah Sherr
    Callahan Family Professor of Computer Science
    Georgetown University

14. Eugene H. Spafford
    Professor of Computer Science
    Executive Director Emeritus, CERIAS
    Purdue University

15. David Wagner
    Professor, Electrical Engineering and Computer Science Department
    UC Berkeley

CERTIFICATE OF COMPLIANCE

1. This document complies with the type-volume limit as set out in Fed. R. App. P. 32(a)(7), because it contains 6,492 words, excluding the parts of the document exempted by Fed. R. App. P. 32(f) and Circuit Rule 32-4.

2. This document complies with the typeface requirements of Fed. R. App. P. 32(a)(5) and the type-style requirements of Fed. R. App. P. 32(a)(6) because this document has been prepared in a proportionally spaced typeface using Microsoft Word for Microsoft 365 MSO (Version 2201) in 14-point Times New Roman font.

Jef Pearlman
Counsel for Amici Curiae

CERTIFICATE OF SERVICE

I hereby certify that on February 16, 2022, I electronically filed the foregoing brief with the Court of Appeals for the Eleventh Circuit via the Court's CM/ECF system. Parties represented by registered CM/ECF users will be served by the CM/ECF system.

Jef Pearlman
Counsel for Amici Curiae