UNITED STATES INTERNATIONAL TRADE COMMISSION
WASHINGTON, D.C.

In the Matter of CERTAIN NETWORK PERSONAL COMPUTERS AND MOBILE DEVICES	Investigation No. 337-TA-_____

Complaint of Aqua Connect Inc. and Strategic Technology Partners, LLC under Section 337 of the Tariff Act of 1930, as amended

Complainants

Aqua Connect, Inc.
Strategic Technology Partners, LLC
1815 E. Heim Ave., Suite 100
Orange, CA 92865
Telephone: (310) 694-5043

Proposed Respondent

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
Telephone: 408-996-1010

Counsel for Complainants

Gregory S. Dovel
Rick Lyon
Simon Franzini
DOVEL & LUNER, LLP
201 Santa Monica Blvd.
Suite 600
Santa Monica, CA 90401
Telephone: (310) 656-7066
Facsimile: (310) 656-7069

V. James Adduci, II
Jonathan J. Engler
Daniel F. Smith
ADDUCI MASTRIANI & SCHAUMBERG L.L.P
1133 Connecticut Avenue, N.W.
Washington, DC 20036
Telephone: (202) 467-6300
Facsimile: (202) 466-2006

Table of Contents

I. Introduction. 1

II. Background. 1

III. Complainants. 5

IV. Proposed Respondent. 6

V. Asserted Patents. 6

A. Overview of the Invention. 6

1. Technological Background. 6

2. Challenges to developing macOS remote desktop and terminal server applications. 10

3. The old way of doing things. 12

4. The patented invention. 15

B. The '386 Patent. 16

1. Ownership of the '386 Patent. 17

2. Foreign Counterparts to the '386 Patent. 17

3. Non-Technical Description of the '386 Patent. 18

4. Apple's Knowledge of the '386 Patent. 19

C. The '502 Patent. 23

1. Ownership of the '502 Patent. 23

2. Foreign Counterparts to the '502 Patent. 24

3. Non-Technical Description of the '502 Patent. 24

4. Apple's Knowledge of the '502 Patent. 25

D. Licensees. 26

VI. Products at Issue. 26

A. Aqua Connect's Remote Desktop and Terminal Server Product. 26

B. Apple's Accused Products. 27

VII. Apple's Unlawful Acts. 28

A. Infringement of the '386 Patent. 28

B. Infringement of the '502 Patent. 32

VIII. Specific Acts of Importation and Sale. 35

IX. Classification of the Infringing Product under the Harmonized Tariff Schedule of the United States. 36

X. Related Litigation. 37

XI. Domestic Industry. 37
A. Technical Prong. 37
1. The '386 Patent. 37
2. The '502 Patent. 37
B. Economic Prong. 38
1. Significant Investment in Plant and Equipment 38
2. Significant Employment of Labor and Capital. 38
3. Substantial Investment in Exploiting the Asserted Patents, Including Engineering, Research, and Development 38
XII. Request for Relief. 39

Exhibit list

Exhibit[1]	Description
1	U.S. Patent RE46,386 (Certified Copy)
2	U.S. Patent 8,924,502 (Certified Copy)
3	macOS "Lion" End User License Agreement
4	Apple Inc. Form 10K
5	Assignment from Joseph Cohen to Aqua Connect (Certified Copy)
6	Assignment from Aqua Connect to STP (Certified Copy)
7	Confirmatory Exclusive License between Aqua Connect and STP
8	Aqua Connect Home Page
9	Aqua Connect Website – Ignision Server
10	Ignision Server User Manual
11	Aqua Connect Website – Remote Desktop Services
12	ACRDS User Manual
13	Declaration of Charlotte Morris in support of the Complaint
14*	Representative Claim Chart comparing Apple Mac Computer to '386 Patent
15*	Representative Claim Chart comparing Apple Mac Computer to '502 Patent
16*	Representative Claim Chart comparing Ignision Server to '386 Patent
17*	Representative Claim Chart comparing Ignision Server to '502 Patent
18*	Declaration of Renee Mehrian on behalf of Aqua Connect, Inc.
19	Apple website – macOS Sierra: Set up and use screen sharing
20	Apple website – Compare Mac Models
21	Apple Developer website – Mach Overview
22	Singh, Amit, Mac OS X internals (Pearson Education, 2007) (excerpts)
23	Bootstrap.h source code (macOS 10.9)
24	Launchd configuration file for ScreensharingAgent (com.apple.screensharing.agent)
25	launchd.plist man page
26	Apple Developer website – Creating Launch Daemons and Launch Agents
27	Mach context tree (macOS 10.9)
28	Launchd configuration file for screensharingd (com.apple.screensharing.plist)
29	Apple Developer website - CGDisplayStreamUpdateGetRects
30*	decompiled pseudocode for ScreensharingAgent - sub_10001f171.c
31	Apple Developer - IOSurface
32*	decompiled pseudocode for ScreensharingAgent - EntryPoint.c
33*	decompiled pseudocode for ScreensharingAgent - sub_100009862.c
34	Ion client user manual
35	Aqua Connect Support Center - Ignision Information
36	Launchd configuration file for Ignision KvmAgentServer (net.aquaconnect.ignision.se.kvmagentserver.plist)
37	Launchd configuration file for IonServer-KvmAgentClient (net.aquaconnect.ignision.se.ionserver.plist)
38	Email re Apple's knowledge of Aqua Connect

[1] Exhibits marked with an asterisk are confidential.

39*	List of Apple leads
40	Emails to Aaron Stanley
41	Emails to Trea Grillo
42	Emails to Fred Licht
43	Emails to Tristan Barthe
44	Emails to Atif Alam
45	Emails to Damon Whitney
46	Emails to Rune Hansen
47	Emails to Matt Peterman
48	Mass email printout
49	05-02-17 Ignision Announce Mass E-Mail
50	Apple Developer website – Bootstrap contexts
51	ZDNet – Aqua Connect Terminal Server 3.1
52	Screenshots of Accused Products
53	Email chain with Aaron Stanley re trial installation
54	Email chain with Trea Grillo
55	Email between Aqua Connect and Apple
56	Email between Aqua Connect and Apple
57	Email between Aqua Connect and Apple
58	Email between Aqua Connect and Apple
59	Email between Aqua Connect and Apple
60	Email between Aqua Connect and Apple
61	Attachment to email (Exhibit 60) between Aqua Connect and Apple
62	macOS "Snow Leopard" End User License Agreement
63	macOS "Sierra" End User License Agreement
64	U.S. Patent 8,549,093
65	Strategic Technology Partners, LLC operating agreement
66*	Declaration of Joseph Cohen
67*	Ignision source code excerpts
68	Cooperation through bug reporter

Appendix list

Appendix Item	Description
A	Prosecution history of U.S. Patent RE46,386 (Certified Copy)
B	Prosecution history of U.S. Patent 8,549,093 (Certified Copy)
C	Prosecution history of U.S. Patent 8,924,502 (Certified Copy)
D	Technical References Cited in Prosecution Histories of Asserted Patents

I. Introduction.

1. Complainant Aqua Connect develops and sells remote desktop and terminal server products for macOS. Aqua Connect's products are protected by two United States patents, U.S. Patent RE46,386 and U.S. Patent 8,924,502.

2. Proposed Respondent Apple unlawfully imports, sells for importation, and sells after importation Mac computers, iPhones, iPads, iPods, and Apple TVs that, on information and belief, infringe one or more claims of Aqua Connect's patents:

- '386 patent: claims 1-4, 8-19, 21-29, and 31-35.
- '502 patent: claims 1-4, 8-19, 21-29, 31-36, and 38.

3. Aqua Connect seeks a limited exclusion order barring the importation, sale for importation, and sale after importation of Apple's infringing Mac computers, iPhones, iPads, iPods, and Apple TVs. Aqua Connect also seeks a permanent cease and desist order prohibiting Apple and its agents from selling, offering to sell, marketing, demonstrating, distributing, warehousing for distribution, or soliciting any sale of imported infringing Mac computers, iPhones, iPads, iPods, and Apple TVs in the United States.

II. Background.

4. A computer's "operating system" includes software that supports certain basic functions of the computer, such as executing software applications.

5. Apple's computer operating system is macOS.[2] The macOS architecture erects technological barriers to viewing and controlling the user interface (the visually displayed "desktop" for example) over a network.

---

[2] Earlier versions of macOS were marketed under the name "Mac OS X." For simplicity and consistency, this complaint refers to all versions of Apple's Mac operating system as "macOS," whether they were marketed as "macOS" or "Mac OS X" at the time.

6. For many years, no one could come up with a good way to overcome these barriers. As a result, there was no secure and efficient way to view and control macOS computers remotely over a network.

7. In 2008, inventor Joseph Cohen developed a novel technological solution that overcame these problems for computer systems running macOS and other "Mach-derived" systems (i.e., ones whose operating system core is based on an operating system core named "Mach"). Mr. Cohen's solution allowed the user interface information of a macOS computer to be securely and efficiently transferred out of the "user context" (the execution environment where macOS isolates user tasks) to a separate task in the "system context" (the execution environment for system tasks). His solution also allowed input information received over a network to be efficiently and securely transferred *into* the user context. This allowed a user to view and control a graphical desktop session on a macOS computer from a remote computer, over a network such as a Local Area Network or the Internet.

8. Mr. Cohen applied for and was awarded United States patents—including the asserted '386 patent and the '502 patent ("Asserted Patents")—to protect his inventions. Ex. 1 (U.S. Patent RE46,386); Ex. 2 (U.S. Patent 8,924,502).

9. Mr. Cohen also built his inventions into his company Aqua Connect's remote desktop and terminal services application, Aqua Connect Terminal Server (ACTS). The result was ACTS 3.0—the first fully functional, secure, and efficient remote desktop and terminal services solution for macOS.

10. ACTS 3.0 was released on September 24, 2008, and met with significant industry praise and commercial success. *See, e.g.*, Ex. 51 (ZDNet - Aqua Connect Terminal Server 3.1) (April 3, 2009) ("Organizations that have chosen to rely on Mac OS X-based servers had no way to enjoy this same advantage until Aqua Connect took up the challenge and created its terminal

services technology. This means that if organizations want to access Mac Server-based applications using thin clients designed to support Microsoft's remote graphics protocol (RDP), Aqua Connect is the only game in town... Aqua Connect keep doing what you're doing! Apple, Citrix are you listening? Aqua Connect would be a wonderful addition to your portfolio of technologies.") (emphasis added). Aqua Connect's customers included numerous universities, government agencies, and research facilities, including University of California, University of Texas, Harvard University, Stanford University, Florida State University, Johns Hopkins University, University of Chicago, University of Utah, University of Nebraska, Washington University, Wake Forest University, Los Alamos National Laboratory, Lawrence Livermore National Laboratory, NASA, Mayo Clinic, National Center for Biotechnology Information, National Library of Medicine, and National Oceanic and Atmospheric Administration, among others.

11. Initially, ACTS 3.0 had Apple's full support. Apple's enterprise and government customers (and potential customers) needed a fully-functional, secure, efficient remote desktop and terminal server solution for macOS to be able to use Apple computers for their enterprise or organization. (A "terminal server" is an application that "serves" user terminals to other computer systems—i.e., it allows multiple users to remotely control different user sessions on the same computer at the same time). ACTS 3.0 was the only application that could fill this need at the time. As a result, Apple worked closely with Aqua Connect on development and sales. *See, e.g.*, Ex. 55-61 (emails between Aqua Connect and Apple); Ex. 68 (communication between Aqua Connect and Apple through Apple's bug reporter tool).

12. In early 2011, however, Apple—abruptly and without explanation—stopped cooperating with Aqua Connect.

13. A few months later, in July of 2011, Apple launched a new major release of macOS, Mac OS X 10.7 (also known as "Lion"). Lion included a redesigned remote desktop and terminal server solution, "Screen Sharing." As explained below, on information and belief, Screen Sharing uses Aqua Connect's patented technology without Aqua Connect's permission. Every macOS release since Lion has included a Screen Sharing feature that, on information and belief, uses Aqua Connect's patented technology without Aqua Connect's permission.

14. In addition, around the same time, Apple launched a new major release of its mobile operating system, iOS 5. iOS 5 included a new feature, "Airplay Mirroring," which allowed the screen of an iOS device such as an iPhone to be shared with another computing device (such as an Apple TV) over a computer network. As explained below, on information and belief, AirPlay Mirroring uses Aqua Connect's patented technology without Aqua Connect's permission. Every iOS release since iOS 5 has included an AirPlay Mirroring feature that, on information and belief, uses Aqua Connect's patented technology without Aqua Connect's permission.

15. Furthermore, with Lion's release, Apple changed the end user license agreement for its operating system. The new end user license agreement required that any connection to control a separate graphical desktop session of macOS (i.e., one other than the one displayed on the screen attached to the computer running it) "may only be made through the Screen Sharing feature of the Apple Software"—and not through third-party remote desktop applications offered by Apple's competitors (such as Aqua Connect):

H. Remote Desktop Connections. Subject to the terms and conditions of this License, when remotely connecting from another computer or electronic device (each a "Device") to an Apple-branded computer that is running the Apple Software (for purposes of this Section, such Apple-branded computer is referred to as the "Home Mac"), whether through the Screen Sharing feature or through any other means:

(i) only one (1) Device may remotely connect at any one time, whether directly or indirectly, to control the graphical desktop session of the Apple Software that is running and being displayed on the Home Mac; and

(ii) a reasonable number of Devices may remotely connect at the same time for the sole purpose of simultaneously observing the same graphical desktop session of the Apple Software that is running and being displayed on the Home Mac, as long as they do not control the Apple Software in any way; but

(iii) only one (1) Apple-branded Device may remotely connect at any one time, whether directly or indirectly, to control a separate graphical desktop session of the Apple Software that is different from the one running and being displayed on the Home Mac, and such connection may only be made through the Screen Sharing feature of the Apple Software.

Ex. 3 (macOS 10.7 Lion End User License Agreement) at 2; *compare with* Ex. 62 (macOS 10.6 Snow Leopard End User License Agreement). This requirement has been included in the end user license agreements for each subsequent release of macOS. *See, e.g.,* Ex. 63 (macOS 10.12 Sierra End User License Agreement) at 3.

III. Complainants.

16. Aqua Connect, Inc. is a Los Angeles-based software company. Its principal place of business and headquarters is at 1815 E. Heim Ave, Suite 100, Orange, California 92865.

17. Aqua Connect was established in 2007 by the inventor of the Asserted Patents, Joseph Cohen. It is the exclusive licensee of the Asserted Patents and develops and sells a remote desktop and terminal server application for macOS that practices the Asserted Patents. Early versions of Aqua Connect's remote desktop and terminal server product was marketed under the name "Aqua Connect Terminal Server" ("ACTS"). The name of the product was subsequently changed to "Aqua Connect Remote Desktop Server" ("ACRDS") and then "Ignision Server."

18. Strategic Technology Partners, LLC is a wholly-owned subsidiary of Aqua Connect. Aqua Connect formed Strategic Technology Partners on July 31, 2013, for the purpose of holding its patents in a dedicated subsidiary. Ex. 18 (Declaration of Renee Mehrian in support of the Complaint), ¶3; Ex. 65 (STP operating agreement). Strategic Technology Partner's principal place of business and headquarters is the same as that of its parent, Aqua Connect. *Id.*

19. Aqua Connect is and always has been the sole Member of Strategic Technology Partners, and has and always has had a 100% ownership stake in Strategic Technology Partners. Ex. 18, ¶3; Ex. 65 (STP operating agreement), ¶2. Strategic Technology Partners is and always has been managed solely by directors and officers of Aqua Connect. Ex. 18, ¶3; Ex. 65, ¶4(a).

20. On February 18, 2014, Aqua Connect assigned its patents to Strategic Technology Partners. Ex. 18, ¶3. Because Strategic Technology Partners is a wholly-owned subsidiary of Aqua Connect, Aqua Connect has always retained a 100% interest in the Asserted Patents through its ownership of Strategic Technology Partners.

21. This Complaint refers to the parent company Aqua Connect, Inc. and its wholly-owned subsidiary Strategic Technology Partners, LLC collectively as "Aqua Connect."

IV. Proposed Respondent.

22. On information and belief, Proposed Respondent Apple Inc. is a California corporation with its principle place of business at 1 Infinite Loop, Cupertino, California 95014. Ex. 4 (Apple Inc. Form 10K) at 2. Apple makes, imports, markets and sells Mac computers, iPhones, iPads, iPods, and Apple TVs. Apple also develops, imports, markets, sells, and licenses the macOS and iOS operating systems.

V. Asserted Patents.

A. Overview of the Invention.

1. Technological Background.

Mach inter-process communication.

23. Apple's macOS and iOS operating systems are "Mach-derived" operating systems. Their core or "kernel" is based on the Mach kernel, an operating system core developed at Carnegie Mellon University. *See* Ex. 21 (Apple Developer website – Mach Overview).

24. A process or "task" is a program or subprogram executing on a computer, for example a word processing task. Inter-process communication is communication between two computer processes or tasks. On Mach-derived operating systems, inter-process communication can make use of "messages" and "Mach ports." Ex. 22 (Amit Singh, *Mac OS X Internals: A Systems Approach – excerpts*) at § 9.2.

25. A message can be used to send information to a task. Ex. 22 (Singh), § 9.2. A Mach port is the endpoint of a unidirectional communication channel. A task can be the designated receiver of a Mach port. Other tasks can send messages to that task using that Mach port. Ex. 21 (Apple Developer website – Mach Overview); *see* Ex. 22 (Singh), § 9.2. To receive messages using a Mach port, the receiving task must have "receive rights" to that port. And to send messages using a Mach port, a sender must have "send rights" to that port. Ex. 22 (Singh), § 9.2.

Bootstrap server.

26. When a task creates a Mach port for communication, it gives itself "receive rights" to that Mach port. As a result, it can receive messages sent to that port.

27. To allow other tasks to acquire send rights to a Mach port, the designated receiver of a Mach port can give the port a name and register it with a "bootstrap server." *See* Ex. 22 (Singh), § 9.4.2.

28. In software, a "server" is a software component that provides services to other software components upon request. In macOS, a "bootstrap server" is a software component that stores and maintains information about Mach ports and provides that information to other tasks upon request. *See* Ex. 22 (Singh), § 9.4.2.

29. When a task registers a named Mach port with a bootstrap server, the bootstrap server saves the Mach port's name (as well as other information about that Mach port) in a data container in memory (for example, a linked list). Another task can look up a Mach port registered with a bootstrap server by its name and, in that manner, acquire send rights to that port.

30. A task can communicate with a bootstrap server using a "bootstrap port." *See* Ex. 22 (Singh), § 9.4.2. A "bootstrap port" is a special Mach port that a task can use to communicate with a bootstrap server. *Id.* Each task has send rights to at least one bootstrap port and can use it to access at least one bootstrap server or data container in memory for bootstrap ports. A task can create another task, in which case it is sometimes called the "parent" of the task it created. A task can inherit send rights to a bootstrap port from the task's "parent."*See* Ex. 22 (Singh), § 9.4.2.

Bootstrap contexts.

31. Tasks on Mach-derived operating systems run within "bootstrap contexts" (sometimes simply called "contexts"). *See* Ex. 22 (Singh), § 9.4.2; Ex. 50 (Apple Developer website – Bootstrap Contexts). A "context" is a task's execution environment, that determines what operating system services (Mach ports) it can access. Ex. 22 (Singh), § 9.4.2. There can be different contexts on the same computer.

32. In macOS, there is a top-level bootstrap context when the system starts up. This is sometimes called the "startup context" or "system context." The startup context can have a

bootstrap server (and associated data container for bootstrap ports) where tasks can register the named Mach ports that they create (and therefore have receive rights to). Ex. 22 (Singh), § 9.4.2.1; Ex. 50 (Apple Developer website – Bootstrap Contexts).

33. The system can create additional bootstrap contexts. For example, when a user logs in, a bootstrap context for that user can be created. This is sometimes called a "user context." Ex. 22 (Singh), § 9.4.2.1; Ex. 50 (Apple Developer website – Bootstrap Contexts). A task in the system context creates the user context; as a result, the user context is sometimes called a "child" of the system context (and the system context is called a "parent" of the user context). User tasks, for example, the graphical user interface that allows a user to interact with the computer, can be executed in the "user context." Ex. 22 (Singh), § 9.4.2.1; *see* Ex. 50 (Apple Developer website – Bootstrap Contexts).

34. Tasks in a user context can look up Mach ports registered with the bootstrap task (or stored in the associated data container) for the user context through their bootstrap port. Tasks in a user context also can generally look up Mach ports registered with the bootstrap task (or stored in the data container) for the parent context of the user context, i.e., the system context.[3]

35. However, tasks in a parent context generally cannot look up Mach ports registered with the bootstrap task (or in the data container) for their child contexts. Similarly, a task in one child context (for example, one user context) cannot look up Mach ports registered with the bootstrap task (or in the data container) for a different child context (for example, another user context). *See* Ex. 22 (Singh), § 9.4.2.

---

[3] However, in some implementations, the Mach-derived system can block tasks in the user context from looking up some or all of the Mach ports registered with the bootstrap task of (or the bootstrap database for) the system context.

36. One reason for this is that although a task in a child context such as a user context can access the bootstrap task of (or the data container for) the parent context through its bootstrap port, tasks in a parent context (such as a system context) cannot access the bootstrap task (or the data container for) any of its children. Similarly, a task in one child context (such as a user context) cannot access the bootstrap task (or data container) for other children (for example, other user contexts) for the same reasons. *Id.*

37. A system can have multiple user contexts (for example, one user context for a given user, and another user context for a different user who is also using the computer system), which can be children of the system context. Each user context can have its own bootstrap task for named Mach ports registered by tasks in that user context, and/or can have its own data container for named Mach ports registered by tasks in that user context. *See* Ex. 50 (Apple Developer website – Bootstrap Contexts).

38. Having a separate bootstrap context for each user has a number of benefits. It makes the system more secure. It prevents tasks in the system context or another user's context from interfering with a user's tasks. It allows the system to shield certain system tasks from user interference. And it segregates each user's data in his or her own context.

40. Having a separate bootstrap context also improves and facilitates software design. Because each user's tasks are segregated in that user's context, software programs do not need to design a system to keep track of different program instances and data for each user. Everything in a particular user's context belongs to that user.

2. Challenges to developing macOS remote desktop and terminal server applications.

41. Having a separate bootstrap context for each user also creates certain challenges. Because tasks in the system context cannot look up the Mach ports for tasks in the user context,

they cannot easily access user-related data, for example user interface data. This makes it difficult to develop an effective and secure remote desktop or terminal server application for macOS.

42. Indeed, a remote desktop or terminal server application running on a computer (the "host" computer) must be accessible from a remote device (the "remote" computer), to allow a user on a remote computer to connect to it. The application must also be able to access the user interface of the host computer, so that the application can (a) capture the user interface of the host computer and send the interface to the remote computer and (b) control the user interface of the host computer based on input provided on the remote computer. In addition, a terminal server application must allow users to log into their own sessions, and so must exist and be accessible *before* a user logs in. And a terminal server application must be able to support multiple concurrent user sessions that do not affect each other.

43. For operating systems without separate bootstrap contexts (such as Microsoft Windows), these requirements did not pose any particular challenges.

44. For macOS and other Mach-derived operating systems, however, these requirements gave rise to significant technological challenges.

45. To be able to access user tasks and the user interface (for example, to obtain display data to transmit to the remote device, or to control tasks based on input from the remote device), the remote desktop or terminal server application needed to be located in the user context (so it could look up the Mach ports for user tasks to be able to interface with them). But locating a remote desktop or terminal server application in the user context would lead to the following problems:

- To establish a network connection between the remote computer and the host computer to enable a user on the remote computer to view or control a particular user

session, a remote computer would need to know the specific connection information (e.g. TCP/IP address and port) associated with the remote desktop/terminal server application of that user on the host computer. And there would be no way for the remote computer to know this before the user logged into the host computer (which he or she could not do until after a connection was established).

- If a remote desktop/terminal server application runs only in the user context on the host computer, then for a remote computer to be able to connect, the user must already be logged into the host computer (otherwise there is no user context and therefore no remote desktop/terminal server application running). Accordingly, such a solution would require a user to already be logged into the host computer, and would not allow users to log into the host computer remotely. If, for example, a user wanted to access or control his or her work computer from home but forgot to log in before leaving the office, he or she would not be able to do so.
- Allowing multiple users to control their respective user sessions on the host computer at the same time (necessary for a fully functional terminal server application) would require separate connections from each remote computer to each application within each user context. Because these applications would be independent of one another (they would be multiple instances of a remote desktop/terminal server application), they would compete with each other for network resources, potentially in inefficient ways.

3. The old way of doing things.

46. The conventional solution to these problems, which Apple used for many years, was simply to use a single task in the user context, and use workarounds to address some of the shortcomings listed above.

47. For example, before the Screen Sharing application was released with macOS 10.7, Apple used a single task in the user context (AppleVNCserver) to provide rudimentary screen sharing functionality for macOS.

48. If no user was logged in, then macOS on the host computer would execute an instance of AppleVNCserver in the loginwindow context (the context for the login window). If a remote computer connected, the user operating that remote computer could log into the host computer using his or her credentials. This caused a new user context to be created. At that point, macOS killed the AppleVNCserver process in the loginwindow context and created a new AppleVNCserver process in the user context that had just been created. Because the new process was in the user context, it could access user tasks and therefore (1) access the graphical user interface and send it to the remote computer, and (2) control the user interface based on inputs received on the remote computer.

49. Apple's solution followed the teachings in the prior art at the time of the invention.

50. First, the conventional wisdom was that it was not feasible to access or control a task in the user context from the system context. If task 1 in the system context needed to access or control task 2 in the user context, the only way to allow it to do so was run task 1 in the user context. The "single task in user context" solution followed the conventional wisdom that for one task to send messages to and control another task in the user context, both tasks must be in that same user context.

51. Second, it was conventional wisdom that the steps for carrying out a particular functionality (such as sharing the screen) should not be split up into multiple tasks but should instead be executed by a single task if possible. This was because each task is considered a significant expenditure of system resources. As a result, the conventional wisdom taught to

minimize the number of tasks (i.e., not break up the steps needed to carry out a single functionality into multiple tasks). Moreover, it was thought especially important to avoid breaking up functions that needed to share large amounts of data with each other into separate tasks. This was because splitting up such functions would require transferring large amounts of data from task to task, which (a) drained system resources and (b) required significant programming attention to ensure that the transfer would be done efficiently. The "single task in user context" solution followed this conventional wisdom: it allowed a single task to perform a function and it avoided the need for data to be passed between tasks.

52. However—and despite its use by Apple for many years—Apple's conventional solution had a number of drawbacks. For example, running the AppleVNCserver process in the user context exposed the user context directly to the network and therefore to the possibility of malicious code accessing the user context. As a second example, because the AppleVNCserver process existed only in the user context, it did not allow a second user to connect to it and start up his or her own session with its own context. This prevented Apple from offering a terminal server solution (and, indeed, Apple did not offer one for many years, despite significant demand). As a third example, Apple's conventional solution also did not allow a user to connect to a computer using their own credentials if another user was logged in on the host computer—even if that other user was not using the computer at the time. As a fourth example, Apple's conventional solution did not allow a remote user session to run in the background on the host computer—if a remote user was controlling the user interface remotely, that user interface would also be displayed on the (locally connected) display of the host computer. As a fifth example, connecting to a macOS computer without any user logged on required establishing, tearing down, and then re-establishing a connection with the remote computer, which required additional system resources and which led to delays and connectivity issues.

4. The patented invention.

53. In 2008, inventor Joseph Cohen realized that instead of following the conventional wisdom to have a single remote desktop task in a single context like the solutions described above, you could split up the functions carried out by a remote desktop application into two separate but associated tasks in two separate Mach contexts: for example, one in the user context (which he called the "agent server"), and the other in the system context (which he called the "agent client").

54. Although a task in the system context cannot look up the Mach port for a task registered in the user context, a task in the user context *can* look up a Mach port for a task registered in the system context. Accordingly, if:

- an "agent client" task exists in the system context of the host computer; and
- a new "agent server" task is created in the user context of the host computer after the user logs in,

the "agent server" task can create an association with the "agent client" task, for example by registering a Mach port with a special data container or server maintained by or accessible to the agent client—as contrasted with registering a Mach port with the bootstrap server in the user context (which the "agent client" in the system context cannot access). Moreover, because a task in the user context can look up the Mach ports for tasks in the system context, the "agent server" (in the user context) would be able to look up and interact with (send a message to) the "agent client" (in the system context) in order to set up this association. Once the association was set up, the agent server and agent client could pass user-interface related data to each other, for example using Mach ports, shared memory, or sockets.

55. Mr. Cohen's invention went against conventional wisdom, which taught that tasks in a parent context cannot interact with tasks in a child context, and that functions should not be

split up across multiple tasks (which are expensive resources), especially when they need to pass large amounts of data among them.

56. But by doing so, Mr. Cohen's invention overcame the limitations of the Mach system architecture: the invention allowed a task in the system context (the agent client) to look up the Mach port for, and therefore be able to interact with, the task in the user context (the agent server), because the agent server task in the user context had registered its Mach port with a data container accessible to the agent client task in the system context. This enabled bidirectional communication between two associated processes, one in the user context and one in the system context. At the same time, Mr. Cohen's invention overcame the problems with Apple's conventional solution, did not rely on any unstable security loopholes, and preserved the benefits of separate user contexts in Mach.

57. Mr. Cohen's invention also had several additional benefits. It allowed a user session that was being controlled by a user on a remote computer to run in the background on the host computer (as opposed to being displayed on the local screen of the host computer). It enabled multi-user terminal server functionality: because the agent client remained in the system context and could communicate with an agent server in a user context, any number of users could connect to the agent client and log in (at which point a new agent server for just that user would be created and would interact with the agent client). In addition: (1) using the agent server as an intermediary allowed the terminal server application to efficiently manage the network connection (for example, by prioritizing connections that are time sensitive) and (2) because this solution allowed for true isolation of user tasks within a user context and did not require access to user tasks from the system context, the data of each terminal server user was more secure than regular multi-user functionality on macOS.

B. The '386 Patent.

1. Ownership of the '386 Patent.

58. The '386 patent, entitled "Updating a User Session in a Mach-derived Computer System Environment," issued on May 2, 2017, from reissue Application No. 14/191,450, filed February 27, 2014. Ex. 1 (U.S. Patent RE46,386). The '386 patent is a re-issue of U.S. Patent No. 8,549,093, which issued on October 1, 2013, from Application No. 12/586,613, filed September 23, 2009. *Id.*; Ex. 64 (U.S. Patent 8,549,093).

59. Joseph Cohen, Aqua Connect's Founder and current Chief Technology Officer, is the sole named inventor of the '386 patent. Ex. 1 (U.S. Patent RE46,386). Mr. Cohen assigned all right, title, and interest to the application that later issued as the '093 patent and subsequently reissued as the '386 patent to Aqua Connect on April 29, 2009. Ex. 5 (Assignment from Joseph Cohen to Aqua Connect). Aqua Connect subsequently assigned all right, title, and interest to the '093 patent (which later reissued as the '386 patent) to its wholly-owned subsidiary Strategic Technology Partners on February 18, 2014. Ex. 6 (Assignment from Aqua Connect to STP). On or around that same date, Strategic Technology Partners granted a worldwide exclusive license to the '093 patent (and any reissues thereof) back to Aqua Connect. Ex. 7 (Confirmatory Exclusive License between Aqua Connect and STP). On September 6, 2017, Strategic Technology Partners and Aqua Connect executed a confirmatory exclusive license agreement confirming the existence and terms of this exclusive license. *Id.*

60. Both assignments described above were recorded with the U.S. Patent Office and are attached to this complaint as exhibits. Ex. 5 (Assignment from Joseph Cohen to Aqua Connect); Ex. 6 (Assignment from Aqua Connect to STP).

2. Foreign Counterparts to the '386 Patent.

61. No foreign counterpart to the '386 patent has been filed.

3. Non-Technical Description of the '386 Patent.[4]

62. The '386 patent relates generally to solutions for updating a user interface of a Mach-derived computing device that is being displayed to a user on a remote computer over a network.

63. The '386 patent explains that "[i]n certain terminal server environments" (where the server is a Mach-derived computer such as a Mac), satisfying the demands of remote clients "leads to issues in being able to securely and synchronously update the graphical display of the server output" that is being displayed on the remote client. Ex. 1 at 2:14-18. As explained in greater detail above, the reason for this is that the system architecture of Mach-derived computers—and specifically the way it segregates user tasks in dedicated "user contexts" that are inaccessible from other parts of the system—erects technological barriers to transferring data, including user interface data, in and out of user sessions. This makes it challenging to allow a remote user to view and control user sessions on a Mach-derived computer over a network.

64. At the time of the invention, no acceptable solution existed. As a result, there was a need "for an improved method for updating graphical display information securely and in a timely fashion" in a Mach-derived terminal server or remote desktop environment. *Id.* at 2:23-25. There was "also a need for an improved means to transport data from a user's session in a terminal server environment, allowing improved communications with a remote device." *Id.* at 2:26-28.

65. To overcome these problems, the '386 patent provides for new solutions to securely and efficiently transfer data about the user interface of a Mach-derived computer from a user session over a network, to update a user instance on a remote device. The '386 solutions

---

[4] This non-technical description is not intended to, and does not, limit, define, or otherwise affect the construction and/or application of the '386 patent's claims.

make use of a novel system architecture: an "agent server" within the context where the user interface is executing (the user context) and an "agent client" *outside* of that context, for example in a parent or "system" context. *See, e.g.,* Ex. 1 ('386 patent) abstract; Fig. 1; 2:31-40; 4:14-23; claim 1. Although they are in different contexts, the agent server and agent client are associated through Mach, which allows them to communicate over a system communication facility (such as a Mach inter-process communication facility). *Id.*, abstract; Fig. 1; 2:43-50; 4:23-63; claim 1. As a result, they can securely and efficiently pass information (such as user interface information) in and out of the user context, and then on to a remote computer over a network for update of the user interface. *Id.*; *see id.* 2:51-55.

66. Certain claims recite additional applications of this technology. For example, certain claims recite using *multiple* agent servers communicating with a single agent client to allow multiple user sessions executing on the same Mach-derived computing device at the same time to be viewed and controlled remotely by different users. *Id.*, claim 4.

4. Apple's Knowledge of the '386 Patent.

67. On information and belief, Apple knew of the '093 patent (which was re-issued as the '386 patent) since it issued on October 1, 2013, and was specifically informed of the '093 patent by Aqua Connect at least as early as February 7, 2014 (or Apple was willfully blind to these facts). *See* below.

68. On information and belief, Apple knew of the '386 patent since it issued on May 2, 2017, and was specifically informed of the '386 patent by Aqua Connect on June 29, 2017 (or Apple was willfully blind to these facts). *See* below.

69. Indeed, Apple was aware of Aqua Connect since at least 2007. Apple acknowledged this, for example, in the following email from Apple to Aqua Connect:

-------- Original Message --------
Subject: Cooperation with Apple
From: Joel Rennich <mactroll@apple.com>
Date: Thu, June 21, 2007 1:56 pm
To: autumnradtke@aquaconnect.net

To whom it may concern:

Apple is very aware and interested in Aqua Connect, Inc.'s products. We are currently working together on sharing both technical information through Apple's Developer Technical Services and sales leads generated from our interaction with our existing customers.

Apple Enterprise Sales has a number of customers that are quite interested in leveraging Aqua Connect's terminal server in a variety of large-scale environments, and I'm quite glad that we now have a product in this space on Mac OS X.

Thanks,

Joel Rennich

Consulting Engineering Manager - Apple Enterprise Sales
mactroll@apple.com - 217-721-3811

Changing the world, one server at a time.

Ex. 38 (Email re Apple's knowledge of Aqua Connect). In addition, Apple was aware of ACTS 3.0 and other Aqua Connect remote desktop/terminal server products that practice the '386 patent since the time they were released in 2008. *See* Ex. 55-61 (emails between Aqua Connect and Apple regarding Aqua Connect's products); Ex. 68 (communication between Aqua Connect and Apple through Apple's bug reporter tool). Indeed, Apple was the one to suggest that Mr. Cohen should build a terminal server application for macOS in the first place. Ex. 66 (Cohen Decl.), ¶7. Moreover, Aqua Connect shared the details of its terminal server application with Apple subject to a confidentiality agreement. *Id.*

70. Furthermore, software developer Dan Crosby was employed by Aqua Connect from 2008-2009 and then left to join Apple. Upon information and belief, based on his

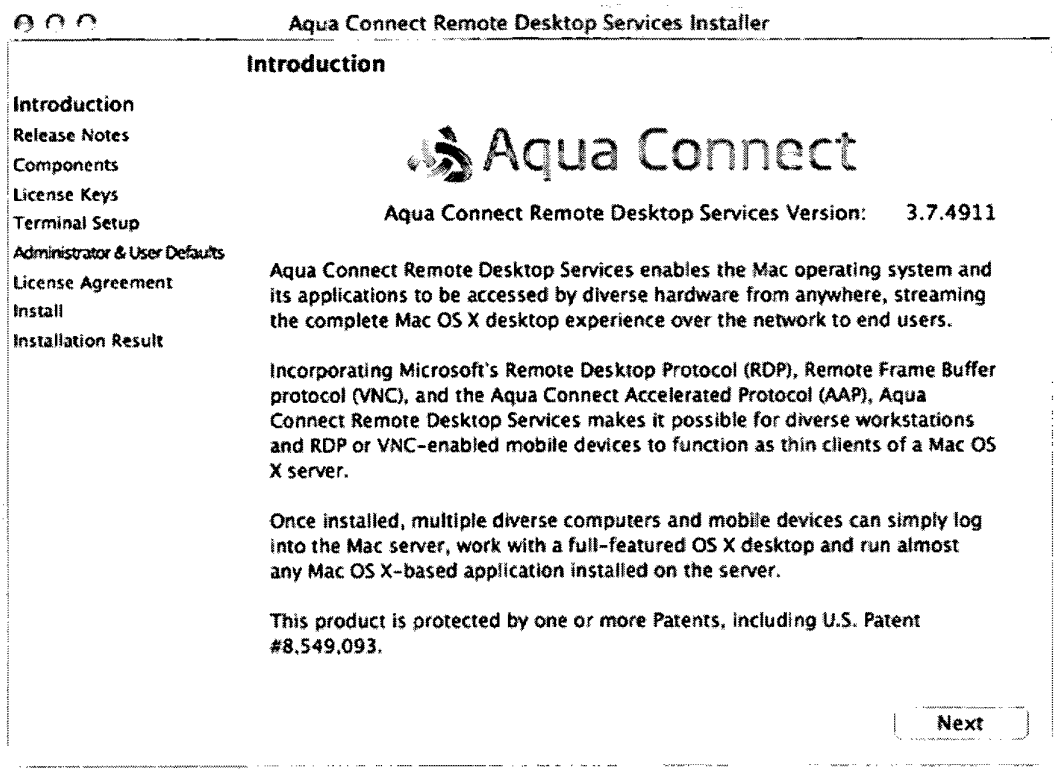
employment at Aqua Connect, Mr. Crosby was aware of ACTS 3.0 and other Aqua Connect products that practice the '386 patent, was aware of the patented solution that those products employed, and was aware that Aqua Connect had filed for patent protection of this solution. Ex. 66 (Cohen Decl.), ¶¶18-20.

71. In addition, at least seven Apple employees requested and were provided with a trial of Aqua Connect Remote Desktop Service after the '093 patent issued:

- Aaron Stanley (astanley@apple.com), February 10, 2014
- Trea Grillo (trea@apple.com), February 21, 2014
- Fred Licht (flicht@apple.com), June 11, 2014
- Tristan Barthe (tristan@apple.com), December 4, 2014
- Atif Alam (atif.alam@apple.com), February 4, 2015
- Damon Whitney (dwhitney@apple.com), February 16, 2015
- Rune Hansen (rune@apple.com), December 29, 2015

Ex. 39 (List of Apple leads); Ex. 40-47 (Emails to Apple employees); Ex. 53 (email chain with Aaron Stanley regarding trial installation); Ex. 54 (email chain with Trea Grillo regarding trial installation).

72. As of the time these trials were provided to Apple, the installer was marked with the '093 patent number and stated that the product "is protected by one or more Patents, including U.S. Patent #8,549,093." The text that was included in the installer is shown below:



73. In addition, on June 29, 2017, Aqua Connect sent an email to all of its "leads," including each of the Apple employees who had requested trials of Aqua Connect, informing them of a new release of Aqua Connect's Ignision product. Ex. 48 (Mass email printout). As shown below, the email was marked with the '386 patent number and informed its recipients (including Apple) that Ignision is "protected by one or more Patents, including U.S. Patent No. 8,924,502 and RE46,386":

Hi {!Contact.FirstName},

We wanted to let you know that our development team is finishing up work on the Enterprise Edition of Ignision Server- our latest line of terminal services and remote desktop products for macOS. The Enterprise Edition builds on the features of the Standard Edition by adding support for load balancing based on available server resources and automatic failover that considers bandwidth and network performance.

The Standard Edition of Ignision Server is available now. If you are interested in receiving a free two week trial or quote, simply respond to this e-mail and I'll be happy to assist. The first open customer BETA of the Enterprise Edition of Ignision Server will be available in July. We are currently enrolling customers in our BETA program at this time. If you are interested in being one of the first customers to try the Enterprise Edition, please let us know as soon as possible. The BETA program will be limited to the first 100 customers who respond.

To learn more about Ignision Server and the Enterprise Edition, please visit our product page: http://aquaconnect.net/ignision-server

To receive a trial immediately, fill out the trial request form for a license key: https://ilicensing.aquaconnect.net/trial/index.htm

Ignision Server is protected by one or more Patents, including U.S. Patent No. 8,924,502 and RE46,386.

Ronnie

--

Ronnie Exley
Executive Sales Consultant
Aqua Connect, Inc.
(310) 694-5043 ext. 513

Ex. 48 (Mass email printout).

74. In addition, at the time of this email, the Ignision installer stated that Ignision Server was protected by the Asserted Patents and identified them by their patent numbers, U.S. Patents RE46,386 and 8,924,502. *See* Ex. 52, Screenshot 10.

C. The '502 Patent.

1. Ownership of the '502 Patent.

75. The '502 patent, entitled "System, Method and Computer Program Product for Updating a User Session in a Mach-derived System Environment," issued on December 30,

2014, from Application No. 14/035,917, filed September 24, 2013. Ex. 2 (U.S. Patent 8,924,502). The '502 patent is a continuation of the '093 patent, and has the same named inventor and the same specification. *Id.*

76. As explained above, Mr. Cohen assigned all right, title, and interest (including all continuations) to the application that later issued as the parent of the '502 patent to Aqua Connect on April 29, 2009. Ex. 5 (Assignment from Joseph Cohen to Aqua Connect). *Id.* Like with the '386 patent, Aqua Connect subsequently assigned all right, title, and interest to the application that later issued as the '502 patent to Strategic Technology Partners on February 18, 2014. Ex. 6 (Assignment from Aqua Connect to STP). And like with the '386 patent, on or around that same date, Strategic Technology Partners granted to Aqua Connect a worldwide exclusive license to the application that later issued as the '502 patent. Ex. 7 (Confirmatory Exclusive License between Aqua Connect and STP). In addition, on September 6, 2017, Strategic Technology Partners and Aqua Connect executed a confirmatory exclusive license agreement confirming the existence and terms of the exclusive license. *Id.*

77. As noted above, both assignments were recorded with the U.S. Patent Office and are attached as exhibits to this complaint. Ex. 5 (Assignment from Joseph Cohen to Aqua Connect); Ex. 6 (Assignment from Aqua Connect to STP).

2. Foreign Counterparts to the '502 Patent.

78. No foreign counterpart to the '502 patent was filed.

3. Non-Technical Description of the '502 Patent.[5]

79. Like the '386 patent, the '502 patent relates generally to solutions for updating a user interface of a Mach-derived computing device that is being displayed on a remote computer

[5] This non-technical description is not intended to, and does not, limit, define, or otherwise affect the construction and/or application of the '502 patent's claims.

over a network. The '502 patent recites additional aspects and applications of the core agent server-agent client architecture that yield additional benefits. For example, certain claims recite using on a Mach-derived computing device the agent server-agent client architecture described above, not just to transmit and display user interface data on a remote computer, but also to receive input data from a remote computer and use the data to control the user interface of a Mach-derived computing device over a network. To do this, the claims recite receiving input data over a network from the remote computer at the agent client; transferring that input data to the agent server via the system communication facility; and then processing that input data at the agent server in the user context. This allows a user on a remote computer not only to view but also to control the user interface of a Mach-derived computer over a network. *See e.g.* Ex. 2 ('502 patent), claim 38.

4. Apple's Knowledge of the '502 Patent.

80. On information and belief, Apple knew of the '502's parent, the '093 patent, since it issued on October 1, 2013, and was specifically informed of the '093 patent by Aqua Connect at least as early as February 7, 2014 (or Apple was willfully blind to these facts). *See* above.

81. On information and belief, Apple knew of the '502 patent since it issued on December 30, 2014, and was specifically informed of the '502 patent by Aqua Connect on May 2, 2017, and again on June 29, 2017 (or Apple was willfully blind to these facts).

82. Indeed, as explained above, Apple was aware of Aqua Connect since at least 2007, and was aware of ACTS 3.0 and other Aqua Connect remote desktop/terminal server products that practice the '502 patent since at least 2008. *See* above.

83. In addition, as explained above, software developer Dan Crosby was employed by Aqua Connect from 2008-2009 and then left to join Apple. Mr. Crosby was aware of ACTS 3.0 and other Aqua Connect products that practice the '502 patent, was aware of the patented

solution that those products employed, and was aware that Aqua Connect had filed for patent protection of this solution. *See* above.

84. In addition, as explained above, numerous Apple employees requested and were sent trials of the Aqua Connect Patented Product that was marked with the '093 patent in the 2014-2016 timeframe, were sent trials of the Ignision product that was marked with the '502 patent no later than May 2, 2017, and were sent an email specifically stating that Aqua Connect's products were protected by the '052 patent no later than June 29, 2017. *See* Ex. 49; *see* exhibits cited above.

D. Licensees.

85. Aqua Connect is the only licensee of the Asserted Patents.

VI. Products at Issue.

A. Aqua Connect's Remote Desktop and Terminal Server Product.

86. Aqua Connect created the market for terminal server and remote desktop applications for macOS devices in 2008 when it launched Aqua Connect Terminal Server 3.0 or "ACTS 3.0."[6] ACTS 3.0 made use of Joseph Cohen's inventions (described in greater detail above) to provide the first secure, stable, and efficient solution for viewing and controlling a user session on a macOS computer over a network from a remote computer.

87. From the time ACTS 3.0 was launched in 2008, Aqua Connect has continued to develop, sell, and support its remote desktop and terminal server product that practices the Asserted Patents ("Aqua Connect Patented Product").

---

[6] Aqua Connect briefly sold two early versions of ACTS (versions 1 and 2) that did not practice Mr. Cohen's invention. These versions of ACTS made use of a security vulnerability to capture the screen of a Mac computer directly from the "system" context rather than making use of the agent server-agent client architecture described and claimed in the '386 and '502 patents. As a result, they were unstable and not secure.

88. The most recent version of the product is marketed under the name Ignision Server ("Ignision"). Ignision is a "[m]ulti-user remote access solution[] for macOS." Ex. 8 (Aqua Connect Home Page). Ignision "allows organizations to deliver macOS as a remote desktop experience to any number of users on any device, anywhere with Microsoft's Remote Desktop Protocol or [Aqua Connect's] own proprietary Ion Protocol." *Id.*
Ignision provides remote desktop and terminal server functionality for versions 10.10 (Yosemite) to 10.12 (Sierra) of macOS. *See* Ex. 9 (Aqua Connect Website – Ignision Server); Ex. 10 (Ignision Server User Manual). Earlier versions were marketed under the name "Aqua Connect Remote Desktop Services" ("ACRDS") and support versions 10.6 (Snow Leopard) to 10.10.2 (Yosemite) of macOS.

89. Each version of the product works in substantially the same manner, and practices at least one claim of the '386 patent as shown below.

B. Apple's Accused Products.

90. The Accused Products include, without limitation:

- Accused macOS Products: Apple's Mac computers running macOS version 10.7 or above (including macOS server installations), including MacBook, MacBook Air, MacBook Pro, iMac, iMac Pro, Mac Pro, and Mac mini models;
- Accused iOS Products: Apple iPhones, iPads (including iPad Pro, iPad, and iPad mini models), and iPods (iPod touch model only) running iOS 5 or above;
- Accused Apple TV Products: Apple's Apple TV products, second generation and above.

91. As explained in greater detail below, on information and belief, the Accused macOS Products infringe at least one claim of the '386 and '502 patents in connection with providing screen sharing and remote desktop functionality to users, including through (1)

"Screen Sharing," and/or "Remote Management" (collectively, "Screen Sharing") and (2) Finder, iChat, "Share my screen," "Back to Mac," and "Screen Sharing viewer" (collectively, "accused remote desktop clients"), and/or "Apple Remote Desktop."

92. In addition, on information and belief, the Accused iOS Products and Accused Apple TV Products infringe at least one claim of the '386 and '502 patents in connection with providing screen sharing functionality to users, including through the "Airplay Mirroring" functionality.

93. On information and belief, Apple makes the Accused Products containing the macOS and iOS software that infringes the Asserted Patents in China, and sells them for importation into the United States, imports them into the United States, and/or sells them within the United States after importation. Ex. 13 (Declaration of Charlotte Morris in support of the Complaint).

VII. Apple's Unlawful Acts.

A. Infringement of the '386 Patent.

94. On information and belief, Apple sells for importation into the United States, imports into the United States, and sells in the United States after importation Accused Products that infringe one or more claims of the '386 patent, either literally or under the doctrine of equivalents.

95. On information and belief, each Accused macOS Product includes the infringing Screen Sharing functionality and the infringing accused remote desktop clients. On information and belief, each Accused iOS Product and Accused Apple TV Product includes the infringing AirPlay Mirroring functionality. On information and belief, these infringing functionalities are loaded onto the Accused Products prior to importation. *See* Ex. 13 (Morris Decl.).

96. On information and belief, the Accused Products infringe claims 1-4, 8-18, 23, 24, 31, 33, and 35. In addition, on information and belief, the Accused macOS Products also infringe claims 19, 21-29, 32, and 34. A claim chart that applies asserted independent claims 1, 8, 25, and 27, and asserted dependent claim 9, to a representative Accused Product (a Mac computer running macOS Sierra) is attached as Exhibit 14.

The Accused Products' infringement of the medium claim:

97. On information and belief, each Accused Product infringes medium claim 9 at importation. Indeed, as set forth above, each Accused macOS Product includes the infringing Screen Sharing functionality and at least one infringing accused remote desktop client at importation. Likewise, each Accused iOS Product includes the infringing AirPlay Mirroring functionality on importation. As such, each Accused macOS Product and each Accused iOS Product includes non-transitory, tangible computer-readable storage media meeting each of the requirements of claim 9 at importation as set forth in greater detail in the attached claim charts. *See* Ex. 14.

The Accused Products' infringement of the method claims:

98. On information and belief, the Accused Products indirectly infringe the method claims at importation.

99. Indeed, on information and belief, the Accused Products are used by Apple's customers (users of Accused Products) after importation to directly infringe the method claims identified above when those customers use the infringing Screen Sharing or Airplay Mirroring functionality. A specific instance of such use is set forth in Exhibit 14. A second specific instance is set forth in a YouTube video available at https://www.youtube.com/watch?v=7B0GB0xIcLM. In addition, on information and belief, Apple was specifically informed of the '386 patent and that Aqua Connects products embodied

the '386 patent, and had copied the innovative aspects of Aqua Connect's products and incorporated them into the Screen Sharing feature. Accordingly, on information and belief, Apple knew that its customer's use of Screen Sharing would infringe the '386 patent, or alternatively was aware that there was a high probability that its customers use of Screen Sharing would infringe a patent and took deliberate acts to avoid confirming this. Furthermore, on the filing date below, Apple was served with this Complaint (including Exhibit 14, which maps the independent claims the '386 to a representative Accused Product), and therefore knew that its customers' use of Screen Sharing infringed the '386 patent for this additional reason. Accordingly, in such cases, Apple's customers directly infringe by performing each step of the patented method using the Accused Products. And Apple indirectly infringes those same method claims, by:

- inducing its customers to use its Accused Products to perform the patented methods, knowing or being willfully blind to the fact that such use would infringe the '386 patent; and
- offering to sell, selling, and/or importing the Accused Products with components or materials that constitute a material part of the claimed inventions, that are especially made or adapted for use in infringing the method claims, and that are not a staple article or commodity with a substantial non-infringing use, knowing or being willfully blind to the fact that using such components or materials would infringe the '386 patent.

The Accused Products' infringement of the system claims:

100. On information and belief, the Accused macOS Products infringe claims 27-29 at importation. Indeed, each such product includes each of the hardware and software components

that meet the elements of the claims, including at least one accused remote desktop client application, at importation. *See* Ex. 14; Ex. 13 (Morris Decl.).

101. Furthermore, on information and belief, the Accused Products indirectly infringe each of the system claims identified above at importation. Indeed, the Accused Products are used by Apple's customers (users of Accused macOS Products or Accused iOS Products) after importation to directly infringe claims 8, 27-29, and 33-34 when they make and use computer systems that make use of the infringing Screen Sharing or Airplay Mirroring functionality. A specific instance of this is set forth in Exhibit 14. A second specific instance is set forth in a YouTube video available at https://www.youtube.com/watch?v=7B0GB0xIcLM. In addition, Apple knew or was willfully blind to the fact that use of Screen Sharing would infringe the '386 patent for the reasons set forth above. Accordingly, in such cases, Apple's customers directly infringe each of the system claims, after importation by making and using systems that include the Accused Products (for example, by turning on an accused Mac and connecting it to a network, or by using the resulting system for screen sharing and remote desktop). And Apple indirectly infringes those same system claims by:

- inducing its customers to use its Accused Products to make or use the patented systems, knowing or being willfully blind to the fact that this would infringe the '386 patent; and
- offering to sell, selling, and/or importing the Accused Products with components or materials that constitute a material part of the claimed inventions, that are especially made or adapted for use in infringing the system claims, and that are not a staple article or commodity with a substantial non-infringing use, knowing or being willfully blind to the fact that using such components or materials would infringe the '386 patent.

B. Infringement of the '502 Patent.

102. On information and belief, Apple sells for importation into the United States, imports into the United States, and sells in the United States after importation Accused Products that directly or indirectly infringe one or more claims of the '502 patent, either literally or under the doctrine of equivalents.

103. On information and belief, each Accused macOS Product includes the infringing Screen Sharing functionality and the infringing accused remote desktop clients. On information and belief, each Accused iOS Product includes the infringing AirPlay Mirroring functionality. On information and belief, these infringing functionalities are loaded onto the Accused Products prior to importation. *See* Ex. 13. On information and belief, these infringing functionalities are not staple articles or commodities of commerce suitable for substantial non-infringing use.

104. On information and belief, each of the Accused Products infringes claims 1-4, 8-18, 31, 33, 35, and 38. In addition, on information and belief, the Accused macOS Products also infringe claims 19, 21-29, 32, 34, and 36.

105. A claim chart that applies asserted independent claims 1, 8, 25, 27, 36, and 38, and asserted dependent claim 9, to a representative Accused Product (a Mac computer running macOS Sierra) is attached as Exhibit 15.

The Accused Products' infringement of the medium claim:

106. On information and belief, each Accused Product infringes medium claim 9 at importation. Indeed, as set forth above, each Accused macOS Product includes the infringing Screen Sharing functionality and at least one infringing accused remote desktop client at importation. Likewise, each Accused iOS Product includes the infringing AirPlay Mirroring functionality on importation. As such, each Accused Product includes non-transitory, tangible

computer-readable storage media meeting each of the requirements of claim 9 at importation as set forth in greater detail in the attached claim charts. *See* Ex. 15.

The Accused Products' infringement of the method claims:

107. On information and belief, the Accused Products indirectly infringe the method claims at importation.

108. Indeed, on information and belief, the Accused Products are used by Apple's customers (users of Accused macOS Products and Accused iOS Products) after importation to directly infringe the method claims identified above when those customers use the infringing Screen Sharing or Airplay Mirroring functionality. *See* Ex. 15. A specific instance of this is set forth in Exhibit 14. A second specific instance is set forth in a YouTube video available at https://www.youtube.com/watch?v=7B0GB0xIcLM. In addition, on information and belief, Apple was specifically informed of the '502 patent and that Aqua Connects products embodied the '502 patent, and had copied the innovative aspects of Aqua Connect's products and incorporated them into the Screen Sharing feature. Accordingly, on information and belief, Apple knew that its customer's use of Screen Sharing would infringe the '502 patent. Alternatively, on information and belief, Apple was aware that there was a high probability that its customers use of Screen Sharing would infringe a patent and took deliberate acts to avoid confirming this. Furthermore, on the filing date below, Apple was served with this Complaint (including Exhibit 15, which maps the independent claims the '502 to a representative Accused Product) on the filing date below, and therefore knew that its customers' use of Screen Sharing infringed the '386 patent for this additional reason. Accordingly, in such cases, Apple's customers directly infringe by performing each step of the patented method using the Accused Products. And Apple indirectly infringes those same method claims, by:

- inducing its customers to use its Accused Products to perform the patented methods, knowing or being willfully blind to the fact that such use would infringe the '502 patent; and
- offering to sell, selling, and/or importing the Accused Products with components or materials that constitute a material part of the claimed inventions, that are especially made or adapted for use in infringing the method claims, and that are not a staple article or commodity with a substantial non-infringing use, knowing or being willfully blind to the fact that using such components or materials would infringe the '502 patent.

The Accused Products' infringement of the system claims:

109. On information and belief, the Accused macOS Products infringe claims 27-29 at importation. Indeed, each such product includes each of the hardware and software components that meet the elements of the claims, including at least one accused remote desktop client, at importation. See Ex. 14; Ex. 13 (Morris Decl.).

110. Furthermore, on information and belief, the Accused Products indirectly infringe each of the system claims identified above at importation. Indeed, on information and belief, the Accused Products are used by Apple's customers (users of Accused macOS Products and Accused iOS Products) after importation to directly infringe claims 8, 27-29, and 33-34 when they make and use computer systems that make use of the infringing Screen Sharing or Airplay Mirroring functionality. A specific instance of this is set forth in Exhibit 14. A second specific instance is set forth in a YouTube video available at https://www.youtube.com/watch?v=7B0GB0xIcLM. In addition, Apple knew or was willfully blind to the fact that use of Screen Sharing would infringe the '502 patent for the reasons set forth above. In such cases, Apple's customers directly infringe after importation by making and

using systems that include the Accused Products (for example, by turning on an accused Mac and connecting it to a network, or by using the resulting system for screen sharing and remote desktop). And Apple indirectly infringes those same system claims by:

- inducing its customers to use its Accused Products to make or use the patented systems, knowing or being willfully blind to the fact that such use would infringe the '502 patent; and
- offering to sell, selling, and/or importing the Accused Products with components or materials that constitute a material part of the claimed inventions, that are especially made or adapted for use in infringing the system claims, and that are not a staple article or commodity with a substantial non-infringing use, knowing or being willfully blind to the fact that using such components or materials would infringe the '502 patent.

VIII. Specific Acts of Importation and Sale.

111. On information and belief, Apple is importing into the United States, selling for importation into the United States, and/or selling within the United States after importation Mac computers, iPhones, iPads, iPods, and Apple TVs that infringe one or more of the Asserted Patents in violation of Section 337, and will continue to do so absent an exclusion order and/or cease and desist order.

112. Aqua Connect obtained, in the United States, samples of the representative MacBook Air and MacBook Pro models, as well as a sample of an iPhone SE model. A detailed description of the steps that Aqua Connect took to procure these samples is set forth in Exhibit 13 (Declaration of Charlotte Morris in support of the Complaint). Specific instances of importation, sale for importation, and/or sale within the United States after importation are set forth below.