Apple, Inc. v. Motorola, Inc. et al Doc. 239 Att. 8

Exhibit 12

Dockets.Justia.com

PUBLIC VERSION

UNITED STATES INTERNATIONAL TRADE COMMISSION
Washington, D.C.

In the Matter of **CERTAIN PERSONAL DATA AND MOBILE COMMUNICATIONS DEVICES AND RELATED SOFTWARE**	**Inv. No. 337-TA-710**

INITIAL DETERMINATION

Administrative Law Judge Carl C. Charneski

Pursuant to a notice of investigation, 75 Fed. Reg. 17434 (2010), this is the Initial Determination in Investigation No. 337-TA-710. It is held that complainants Apple Inc. and NeXT Software, Inc. have established that respondents HTC Corp., HTC America, Inc., and Exedea, Inc. infringed asserted claims 1, 2, 24, and 29 of U.S. Patent No. 6,343,263 (the '263 patent) and asserted claims 1, 8, 15, and 19 of U.S. Patent No. 5,946,647 (the '647 patent) in violation of section 337(b) of the Tariff Act of 1930, as amended. 19 U.S.C. § 1337(b). Complainants have not established that respondents infringed asserted claim 3 of the '647 patent or the asserted claims of U.S. Patent Nos. 6,275,983 (the '983 patent) and 5,481,721 (the '721 patent). It is further held that the asserted patents are not invalid.

TABLE OF CONTENTS

WI-Apple1604800

WI-Apple1604801

I. Background

A. Institution of Investigation

The Commission instituted this investigation by publication of a notice in the *Federal Register* on April 6, 2010, pursuant to subsection (b) of section 337 of the Tariff Act of 1930, as amended. 19 U.S.C. § 1337(b). This investigation was instituted:

> to determine whether there is a violation of subsection (a)(1)(B) of section 337 in the importation into the United States, the sale for importation, or the sale within the United States after importation of certain personal data or mobile communications devices or related software that infringe one or more of claims 1-3, 7, 12, and 32 of U.S. Patent No. 5,519,867; claims 1, 3, 7, 8, and 22 of U.S. Patent No. 6,275,983; claims 1, 3, 8-10, 12, 18, 19, 23, and 24 of U.S. Patent No. 5,566,337; claims 1-3 and 7-13 of U.S. Patent No. 5,929,852; claims 1, 3, 6, 8, 10, 13-16, 19, 20, and 22 of U.S. Patent No. 5,946,647; claim 1 of U.S. Patent No. 5,969,705; claims 1-6, 24, 25, 29, and 30 of U.S. Patent No. 6,343,263; claims 1, 3, 4, 6, 7, 9, 10, 15, and 17 of U.S. Patent No. 5,915,131; claims 1-3, 6, 8, 9, 12, and 14-17 of U.S. Patent No. RE39,486; and claims 1-6 and 19-22 of U.S. Patent No. 5,481,721, and whether an industry in the United States exists as required by subsection (a)(2) of section 337.

75 Fed. Reg. 17434 (2010).

The notice of investigation names Apple Inc., f/k/a Apple Computer, Inc. of Cupertino, California; and NeXT Software, Inc., f/k/a NeXT Computer, Inc. of Cupertino, California as complainants. The named respondents are: High Tech Computer Corp. a/k/a/ HTC Corp. of Taoyuan, Taiwan; HTC America, Inc. of Bellevue, Washington; and Exedea, Inc. of Houston, Texas. The Commission Investigative Staff also is named as a party to this investigation. *Id.*

WI-Apple1604802

B. Procedural History

On April 19, 2010, an 18-month target date of October 6, 2011, was set in this investigation. Order No. 6; Comm'n Notice Not To Review Initial Determination (May 7, 2010).

On April 26, 2010, Chief Judge Luckern consolidated a portion of Inv. No. 337-TA-704 entitled, *Certain Mobile Communications And Computer Devices And Components Thereof,* with the current investigation. This resulted in the addition of two related respondents to the 710 investigation, *i.e.*, Nokia Corporation of Finland and Nokia Inc. of White Plains, New York (collectively, "Nokia"), with respect to the '867, '131, '705, '263, and '486 patents. *See* Inv. No. 337-TA-704, Order No. 5.

On November 10, 2010, the investigation was terminated as to the '867, '131, '852, and '486 patents. Order No. 41; Comm'n Notice Not To Review Initial Determination (Nov. 29, 2010). Shortly thereafter, on November 16, 2010, the investigation was terminated as to (1) claims 6, 10, 13, 14, 16, and 20 of the '647 patent; (2) claims 3 and 8 of the '983 patent; (3) claims 8, 23, and 24 of the '337 patent; (4) claims 4, 5, 25, and 30 of the '263 patent; and (5) claims 2, 3, 4, and 22 of the '721 patent. Order No. 46; Comm'n Notice Not To Review Initial Determination (Dec. 3, 2010).

On January 3, 2011, the target date was extended by two months to a 20-month target date of December 6, 2011. Order No. 73; Comm'n Notice Not To Review Initial Determination (Jan. 27, 2011).[1]

[1] As noted in Order No. 73, the hearing was postponed by approximately six weeks due to Apple's late discovery production.

WI-Apple1604803

Thereafter, on March 1, 2011, the investigation was terminated as to the '705 patent. Order No. 92; Comm'n Notice Not To Review Initial Determination (Mar. 24, 2011). Thereafter, on April 13, 2011, the investigation was terminated as to the following: claims 17, 20, and 22 of the '647 patent; claim 22 of the '983 patent; claims 3 and 6 of the '263 patent; claims 19, 20, and 21 of the '721 patent; and claims 10 and 12 of the '337 patent. Order No. 109; Comm'n Notice Not To Review Initial Determination (Apr. 27, 2011).

On April 7, 2011, an initial determination issued finding that complainants have satisfied the economic prong of the domestic industry requirement. Order No. 102. The Commission reviewed this initial determination and ultimately agreed that "Apple has satisfied the economic prong of the domestic industry requirement" with respect to each of the asserted patents. Comm'n Notice To Review Initial Determination at 3 (May 9, 2011).[2]

A tutorial was presented on November 23, 2010, and the evidentiary hearing was held April 18 – May 6, 2011. During the evidentiary hearing, complainants moved to terminate the investigation as to the '337 patent. On May 9, 2011, the investigation was terminated as to this patent. Order No. 117; Comm'n Notice Not To Review Initial Determination (May 27, 2011). Thus, four patents and fourteen claims remain in issue in this investigation: claims 1, 2, 24, and 29 of the '263 patent; claims 1 and 7 of the '983

---

[2] Inasmuch as the Commission has determined that Apple has satisfied the economic prong, Apple's arguments relating to its licensing activities need not be addressed as the issue is moot.

WI-Apple1604804

patent; claims 1, 3, 8, 15, and 19 of the '647 patent; and claims 1, 5, and 6 of the '721 patent.

On July 5, 2011, the investigation was terminated as to Nokia respondents based on a settlement agreement. Order No. 118 (Initial Determination).

C. The Parties

The complainants are Apple Inc. and NeXT Software, Inc. (collectively, "Apple").[3] The respondents are High Tech Computer Corp. a/k/a/ HTC Corp., HTC America, and Exedea, Inc. (collectively, "HTC").[4] The Commission Investigative Staff ("Staff") is also a party in the investigation.

II. Jurisdiction

The Commission has subject matter, personal, and *in rem* jurisdiction in this

---

[3] Apple Inc. designs, develops, markets, and sells (i) the Mac line of desktop and notebook computers, such as MacBook products, including the MacBook Pro and the ultra-light MacBook Air; (ii) a portfolio of software, such as the Mac OS X operating system that comes pre-installed on every Macintosh computer; (iii) the iPod line of mobile digital devices; (iv) the iPhone (including the iPhone 3G and the iPhone 3GS) and related accessories and services, including a complete software development kit providing tools for programmers to create their own iPhone applications; and (v) a variety of related products, accessories, peripherals, and services, including warranty and customer support. Complaint, ¶ 9. NeXT Software, Inc., a wholly-owned subsidiary of Apple Inc., was an early developer of software for object-oriented programming. Software originally developed by NeXT forms the basis for portions of Mac OS X. *Id.*, ¶ 10.

[4] High Tech Computer Corp. changed its name to HTC Corporation. HTC Corp.'s business includes developing, manufacturing, and selling wireless communication devices. HTC Response, ¶ 13. HTC (BVI) Corp., a non-party, is a wholly-owned subsidiary of HTC Corp. *Id.*, ¶ 14. HTC America is a wholly-owned subsidiary of HTC (BVI) Corp. and provides after-sale support services for HTC's wireless communication devices. *Id.*, ¶ 15. Exedea, Inc. is a wholly-owned subsidiary of HTC (BVI) Corp. *Id.*, ¶ 16.

WI-Apple1604805

investigation. 19 U.S.C. § 1337. All of the respondents have responded to the complaint and notice of investigation and have participated fully in the hearing conducted in this investigation.

III. Importation

Respondents have stipulated to the fact that they have imported into the United States, sold for importation into the United States, and/or sold within the United States after importation the accused personal data and mobile communications devices and related software. Stipulation Relating to Importation of Respondents HTC Corp., HTC America, Inc., and Exedea, Inc. (Nov. 12, 2010); Supplement to the Stipulation Relating to Importation of Respondents HTC Corp., HTC America, Inc., and Exedea (Apr. 15, 2011).

IV. Products at Issue

A. Apple's Domestic Industry Products

Apple's domestic industry products include the MacBook Pro running Mac OS X v10.6 Snow Leopard (CPX-10) and the iPhone 3GS (CPX-11).

B. Accused HTC Products

The Notice of Investigation identified HTC's personal data and mobile communications devices and related software as within the scope of this Investigation. Apple and HTC stipulated that certain HTC handsets, including HTC Evo 4G (CPX-1), HTC Aria (CPX-2), HTC Incredible (running Android 2.1) (CPX-3), HTC Incredible (running Android 2.2) (CPX-4), HTC T-Mobile G2 (CPX-5), running various versions of the Android operative system (including Android 1.5 ("Cupcake"), Android 1.6

WI-Apple1604806

("Donut"), Android 2.1 ("Éclair"), and Android 2.2 ("Froyo")), the Browser application, the HTC Messages application, and the Android Messaging application are representative of all HTC handsets running various versions of the Android operating systems and the additional applications. Stipulation Relating to HTC Representative Handsets (Mar. 29, 2011) ("HTC Rep. Prod. Stip.").

Apple and HTC stipulated that certain HTC handsets are representative of certain versions of the Android operating system and certain applications.

Specifically, for the '263 patent, the source code for the HTC Evo 4G (running Android 2.2) (CPX-1) is representative of certain source code for all HTC products running Android 2.2, and the source code for the HTC Aria (running Android 2.1) (CPX-2) is representative of certain source code for all HTC products running Android 2.1, 1.6 and 1.5. HTC Rep. Prod. Stip. at 2-3.

For the '647 patent, HTC Incredible (running Android 2.1) (CPX-3) is representative of all HTC products with the HTC Messages application and running Android 2.1, 1.6, or 1.5; HTC Incredible (running Android 2.2) (CPX-4) is representative of all HTC products with the Browser application and HTC products running Android 2.2 with the HTC Messages application; and HTC T-Mobile G2 (CPX-5) is representative of all HTC products with Android Messaging application. HTC Rep. Prod. Stip. at 6-11.

For the '721 patent, HTC Incredible (running Android 2.1) (CPX-3) is representative of all HTC products running Android 2.2, 2.1, 1.6, or 1.5. HTC Rep. Prod. Stip. at 4-5.

For the '983 patent, the source code entered into evidence for the HTC Aria (running Android 2.1) is representative of the source code for all HTC products running

Android 1.5, 1.6, 2.1, and 2.2; and the existence of a processor, RAM, and/or ROM in the HTC Aria is representative of the existence of a processor, RAM, and/or ROM in all HTC handsets running Android 1.5, 1.6, 2.1, and 2.2. HTC Rep. Prod. Stip. at 5-6.

Apple and HTC have also stipulated that HTC and its customers have powered on the HTC accused products, such that the devices have become functional for use in the United States (by testing or otherwise), as well as other uses discussed further in the patent sections below. Stipulation Relating to Use of Respondents HTC Corp., HTC America, Inc., and Exedea, Inc.'s Products in the United States (Apr. 17, 2011) ("HTC Use Stip.").

Apple and HTC also stipulated that HTC's Windows products are not subject to the Investigation. Stipulation Regarding Products With Windows Mobile or Windows Phone Operating Systems between Complainants and HTC (Nov. 18, 2010) ("Windows Stip.").

V. Overview of the Asserted Patents

A. The '263 Patent

United States Patent No. 6,343,263 ("the '263 patent") is entitled, "Real-Time Signal Processing System For Serially Transmitted Data." JX-6. The '263 patent "is directed to the transmission of data to and from a computer, and more particularly to a system for performing real-time signal processing of data that is serially transmitted to and from a computer." *Id.*, col. 1, lns. 4-7 (Field of the Invention). The invention of the '263 patent "enables any arbitrary type of data, such as voice, facsimile, multimedia and the like, which is transmitted over any type of communication network, to be handled

WI-Apple1604808

with any type of real-time engine, by abstracting the functions of each of the elements of the systems from one another." *Id.,* col. 2, ln. 66 – col. 3, ln. 6 (Brief Statement of the Invention).

The '263 patent "generally relates to systems for realtime processing of signals and creating greater flexibility in how those systems are managed to allow the signal processing components to be abstracted from the overall system." Polish Tr. 310. Specifically, the '263 patent is directed toward data, such as video, audio, voice, business records, and word files, that is sent over a network and then processed. Realtime signal processing is relevant to video, audio, and voice data that are associated with time frames, *i.e.,* it is time sensitive data. Polish, Tr. 311-12.

B. The '721 Patent

U.S. Patent No. 5,481,721 ("the '721 patent") is entitled, "Method For Providing Automatic And Dynamic Translation Of Object Oriented Programming Language-Based Message Passing Into Operation System Message Passing Using Proxy Objects." JX-1. The invention of the '721 patent provides a method and apparatus for the distribution of objects and the sending of messages between objects that are in different processes. *Id.,* (Abstract). *See* Spielman, Tr. 2703 ("Generally, the '721 patent is about an object-oriented message that is used in an interprocess communication making use of an operating system-based message.").

The invention relates to the field of object-oriented programming and distributed computing. JX-1 at col. 1, lns. 15-16 (Field of the Invention). The invention of the '721 patent is described as follows:

> The present invention permits the distribution of objects and sending of messages between objects that are located in different processes. Initially, a "proxy" object is created in the same process as a sender object. This proxy acts as a local receiver for all the objects in the local program. When the proxy receives a message, the message is encoded and transmitted between programs as a stream of bytes. In the remote process, the message is decoded and executed as if the sender was remote. The result follows the same path, encoded, transmitted, and then decoded back in the local process. The result is then provided to the sending object.

JX-1 at col. 6, lns. 52-63.

C. The '647 Patent

U.S. Patent No. 5,946,647 ("the '647 patent") is entitled, "System And Method For Performing An Action On A Structure In Computer-Generated Data." JX-3. The invention of the '647 patent relates to a system and method for performing computer-based actions on structures identified in computer data. JX-3 at col. 1, lns. 9-11 (Background of the Invention.) The '647 patent "brought together ideas from very different areas of computer science that are not typically combined, such as pattern recognition, user interfaces, launching applications by way of the operating system." Mowry Tr. 2443.

D. The '983 Patent

U.S. Patent No. 6,275,983 ("the '983 patent") is entitled, "Object-Oriented Operating System." JX-4. The '983 patent concerns object-oriented applications that make object-oriented method calls to a procedural operating system. Spielman Tr. 1892; JX-4 at col. 1, lns. 19-22 (Field of the Invention) ("The present invention relates generally to object-oriented computing environments, and more particularly to a system

WI-Apple1604810

and method for providing an object-oriented interface for a procedural operating system.").

VI. General Principles of Patent Law

Pursuant to the Commission's notice of investigation, this is a patent-based investigation. *See* 75 Fed. Reg. 17434 (2010). All of the unfair acts alleged by complainants are instances of alleged infringement of the asserted patents. Any finding of patent infringement or non-infringement requires a two-step analytical approach. First, the asserted patent claims must be construed as a matter of law to determine their proper scope.[5] Second, a factual determination must be made as to whether the properly construed claims read on the accused devices. *Markman v. Westview Instruments, Inc.*, 52 F.3d 967, 976 (Fed. Cir. 1995) (*en banc*), *aff'd*, 517 U.S. 370 (1996).

A. Claim Construction

Claim construction begins with the language of the claims themselves. Claims should be given their ordinary and customary meaning as understood by a person of ordinary skill in the art, viewing the claim terms in the context of the entire patent. *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312-13 (Fed. Cir. 2005), *cert. denied*, 546 U.S. 1170 (2006).[6]

---

[5] Only those claim terms that are in controversy need to be construed, and only to the extent necessary to resolve the controversy. *Vanderlande Indus. Nederland BV v. Int'l Trade Comm.*, 366 F.3d 1311, 1323 (Fed. Cir. 2004); *Vivid Tech., Inc. v. American Sci. & Eng'g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999).

[6] Factors that may be considered when determining the level of ordinary skill in the art include: "(1) the educational level of the inventor; (2) type of problems encountered in the art; (3) prior art solutions to those problems; (4) rapidity with which innovations are made; (5) sophistication of the technology; and (6) educational level of active workers in

WI-Apple1604811

In some instances, claim terms do not have particular meaning in a field of art, and claim construction involves little more than the application of the widely accepted meaning of commonly understood words. *Phillips*, 415 F.3d at 1314. "In such circumstances, general purpose dictionaries may be helpful." *Id.*

In many cases, claim terms have a specialized meaning and it is necessary to determine what a person of skill in the art would have understood the disputed claim language to mean. "Because the meaning of a claim term as understood by persons of skill in the art is often not immediately apparent, and because patentees frequently use terms idiosyncratically, the court looks to 'those sources available to the public that show what a person of skill in the art would have understood disputed claim language to mean.'" *Id.* (quoting *Innova/Pure Water, Inc. v. Safari Water Filtration Sys., Inc.*, 381 F.3d 1111, 1116 (Fed. Cir. 2004)). The "sources" identified by the *Phillips* Court include "the words of the claims themselves, the remainder of the specification, the prosecution history, and extrinsic evidence concerning relevant scientific principles, the meaning of technical terms, and the state of the art." *Id.*

In cases in which the meaning of a claim term is uncertain, the specification usually is the best guide to the meaning of the term. *Id.* at 1315. As a general rule, the particular examples or embodiments discussed in the specification are not to be read into the claims as limitations. *Markman*, 52 F.3d at 979. However, the specification is always highly relevant to the claim construction analysis, and is usually dispositive. *Id.* Moreover, "[t]he construction that stays true to the claim language and most naturally

---

the field." *Environmental Designs, Ltd. v. Union Oil Co.*, 713 F.2d 693, 696 (Fed. Cir. 1983), *cert. denied*, 464 U.S. 1043 (1984).

aligns with the patent's description of the invention will be, in the end, the correct construction." *Id.* at 1316.

Claims are not necessarily, and are not usually, limited in scope to the preferred embodiment. *RF Delaware, Inc. v. Pacific Keystone Techs., Inc.*, 326 F.3d 1255, 1263 (Fed. Cir. 2003); *Decisioning.com, Inc. v. Federated Dep't Stores, Inc.*, 527 F.3d 1300, 1314 (Fed. Cir. 2008) ("[The] description of a preferred embodiment, in the absence of a clear intention to limit claim scope, is an insufficient basis on which to narrow the claims.").

Furthermore, claim interpretations that exclude the preferred embodiment are "rarely, if ever, correct and require highly persuasive evidentiary support." *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1583 (Fed. Cir. 1996). Such a conclusion can be mandated in rare instances by clear intrinsic evidence, such as unambiguous claim language or a clear disclaimer by the patentees during patent prosecution. *Elekta Instrument v. O.U.R. Sci. Int'l*, 214 F.3d 1302, 1308 (Fed. Cir. 2000); *Rheox, Inc. v. Entact, Inc.*, 276 F.3d 1319 (Fed. Cir. 2002).

If the intrinsic evidence does not establish the meaning of a claim, then extrinsic evidence may be considered. Extrinsic evidence consists of all evidence external to the patent and the prosecution history, and includes inventor testimony, expert testimony, and learned treatises. *Phillips*, 415 F.3d at 1317. Inventor testimony can be useful to shed light on the relevant art. In evaluating expert testimony, a court should discount any expert testimony that is clearly at odds with the claim construction mandated by the claims themselves, the written description, and the prosecution history, in other words, with the written record of the patent. *Id.* at 1318. Extrinsic evidence may be considered

if a court deems it helpful in determining the true meaning of language used in the patent claims. *Id.*

This investigation involves means-plus-function claim limitations. When a claim uses the term "means" to describe a limitation, a presumption arises that the inventor used the term to invoke the means-plus-function format authorized by 35 U.S.C. § 112, ¶ 6.[7] *Altiris, Inc. v. Symantec Corp.*, 318 F.3d 1363, 1375 (Fed. Cir. 2003). "This presumption can be rebutted when the claim, in addition to the functional language, recites structure sufficient to perform the claimed function in its entirety." *Id.*

Once a court concludes that a claim limitation is a means-plus-function limitation, two steps of claim construction remain: (1) the court must first identify the function of the limitation; and (2) the court must then look to the specification and identify the corresponding structure for that function. *Biomedino LLC v. Waters Technologies Corp.*, 490 F.3d 946, 950 (Fed. Cir. 2007). If there is no structure in the specification corresponding to the means-plus-function limitation, the claim will be found invalid as indefinite. *Id.*

While the specification must contain structure linked to claimed means, "[a]ll one needs to do in order to obtain the benefit of [§ 112, ¶ 6] is to recite some structure

---

[7] The relevant portion of section 112 provides:

> An element in a claim for a combination may be expressed as a means or step for performing a specified function without the recital of structure, material, or acts in support thereof, and such claim shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof.

35 U.S.C. § 112, ¶ 6.

corresponding to the means in the specification, as the statute states, so that one can readily ascertain what the claim means and comply with the particularity requirement of [§ 112,] ¶ 2." *Id.* (citing *Atmel Corp. v. Info. Storage Devices, Inc.*, 198 F.3d 1374, 1382 (Fed. Cir. 1999)). Additionally, interpretation of what is disclosed in the specification must be made in light of the knowledge of one skilled in the art. *Id.* at 1380.

Thus, in order for a means-plus-function claim to be valid under section 112, the corresponding structure of the limitation "must be disclosed in the written description in such a manner that one skilled in the art will know and understand what structure corresponds to the means limitation. Otherwise, one does not know what the claim means." *Id.* at 1382. However, "the testimony of one of ordinary skill in the art cannot supplant the total absence of structure from the specification." *Id.* (quoting *Default Proof Credit Card Sys., Inc. v. Home Depot U.S.A., Inc.*, 412 F.3d 1291, 1302 (Fed. Cir. 2005)).

"A means-plus-function claim encompasses all structure in the specification corresponding to that element and equivalent structures." However, "[t]he statute does not permit limitation of a means-plus-function claim by adopting a function different from that explicitly recited in the claim. Nor does the statute permit incorporation of structure from the written description beyond that necessary to perform the claimed function." *Micro Chem. Inc. v. Great Plains Chem. Co., Inc.*, 194 F.3d 1250, 1258 (Fed. Cir. 1999).

B. Infringement

Under 35 U.S.C. §271(a), direct infringement consists of making, using, offering to sell, or selling a patented invention without consent of the patent owner. The complainant in a section 337 investigation bears the burden of proving infringement of

the asserted patent claims by a "preponderance of the evidence." *Certain Flooring Products*, Inv. No. 337-TA-443, Comm'n Notice of Final Determination of No Violation of Section 337, 2002 WL 448690 at *59, (Mar. 22, 2002); *Enercon GmbH v. Int'l Trade Comm'n,* 151 F.3d 1376 (Fed. Cir. 1998).

Each patent claim element or limitation is considered material and essential. *London v. Carson Pirie Scott & Co.*, 946 F.2d 1534, 1538 (Fed. Cir. 1991).[8] Literal infringement of a claim occurs when every limitation recited in the claim appears in the accused device, *i.e.,* when the properly construed claim reads on the accused device exactly. *Amhil Enters., Ltd. v. Wawa, Inc.*, 81 F.3d 1554, 1562 (Fed. Cir. 1996); *Southwall Tech. v. Cardinal IG Co.*, 54 F.3d 1570, 1575 (Fed Cir. 1995).

If the accused product does not literally infringe the patent claim, infringement might be found under the doctrine of equivalents. The Supreme Court has described the essential inquiry of the doctrine of equivalents analysis in terms of whether the accused product or process contains elements identical or equivalent to each claimed element of the patented invention. *Warner-Jenkinson Co., Inc. v. Hilton Davis Chemical Co.*, 520 U.S. 17, 40 (1997). Thus, infringement may be found when the accused product performs substantially the same function in substantially the same way to obtain substantially the same result. *See Eagle Comtronics, Inc. v. Arrow Comm. Labs.*, 305 F.3d 1303, 1315 (Fed. Cir. 2002).

As noted, certain of the claim elements at issue in this investigation are written in

[8] Thus, if an accused device lacks a limitation of an independent claim, the device cannot infringe a dependent claim. *See Wahpeton Canvas Co. v. Frontier, Inc.,* 870 F.2d 1546, 1552 n.9 (Fed. Cir. 1989).

means-plus-function format. "Literal infringement of a § 112, ¶ 6 limitation requires that the relevant structure in the accused device perform the identical function recited in the claim and be identical or equivalent" to the structure identified in the written description as corresponding to the recited function. *JVW Enter. v. Interact Accessories, Inc.*, 424 F.3d 1324, 1333 (Fed. Cir. 2005) (citing *Odetics, Inc. v. Storage Tech. Corp.*, 185 F.3d 1259, 1267 (Fed. Cir.1999)). For the relevant structure in the accused device to be equivalent to the structure in the written description, differences between the two must be insubstantial. For example, the structure in the accused device must perform the claimed function in substantially the same way to achieve substantially the same result as the structure in the written description. *JVW*, 424 F.3d at 1333.[9]

Under 35 U.S.C. § 271(b), "[w]hoever actively induces infringement of a patent shall be liable as an infringer." To establish liability, a patentee must prove direct infringement for each instance of indirect infringement. *DSU Medical Corp. v. JMS Co.*, 471 F.3d 1293, 1303 (Fed. Cir. 2006). "In order to succeed on a claim of inducement, the patentee must show, first that there has been direct infringement, and second, that the alleged infringer knowingly induced infringement and possessed specific intent to encourage another's infringement." *Cross Medical Products, Inc. v. Medtronic Sofamor*

[9] "The primary difference between structural equivalents under section 112, paragraph 6 and the doctrine of equivalents is a question of timing." *Frank's Casing, Crew & Rental Tools, Inc. v. Weatherford Int'l, Inc.*, 389 F.3d 1370, 1379 (Fed. Cir. 2004) (citing *Al-Site Corp. v. VSI Int'l, Inc.*, 174 F.3d 1308, 1321 n.2 (Fed. Cir. 1999)). As the Federal Circuit has explained, "[a] proposed equivalent must have arisen at a definite period in time, i.e., either before or after [patent filing]. If before, a § 112, ¶ 6 structural equivalents analysis applies and any analysis for equivalent structure under the doctrine of equivalents collapses into the § 112, ¶ 6 analysis. If after, a non-textual infringement analysis proceeds under the doctrine of equivalents." *Id.*

Danek, Inc., 424 F.3d 1293, 1312 (Fed. Cir. 2005). Mere knowledge of possible infringement by others does not amount to inducement. Specific intent and action to induce infringement must be proven. *Warner-Lambert Co. v. Apotex Corp.*, 316 F.3d 1348, 1363 (Fed. Cir. 2003).

C. Validity

One cannot be held liable for practicing an invalid patent claim. *Pandrol USA, LP v. AirBoss Railway Prods., Inc.*, 320 F.3d 1354, 1365 (Fed. Cir. 2003). However, the claims of a patent are presumed to be valid. 35 U.S.C. § 282; *DMI Inc. v. Deere & Co.*, 802 F.2d 421 (Fed. Cir. 1986). A respondent that has raised patent invalidity as an affirmative defense must overcome the presumption by "clear and convincing" evidence of invalidity. *Checkpoint Systems, Inc. v. United States Int'l Trade Comm'n*, 54 F.3d 756, 761 (Fed. Cir. 1995).

1. Anticipation

Pursuant to 35 U.S.C. § 102, prior art anticipates a patent claim when a single piece of art discloses each and every limitation of the claimed invention. *See Schering Corp. v. Geneva Pharms.*, 339 F.3d 1373, 1377 (Fed. Cir. 2003); *C.R. Bard v. M3 Sys.*, 157 F.3d 1340, 1349 (Fed. Cir. 2000). The disclosure by an invalidating reference need not be express, but may anticipate by inherency where such inherency would be appreciated by one of ordinary skill in the art. *EMI Group North America, Inc. v. Cypress Semiconductor Corp.*, 268 F.3d 1342, 1350 (Fed. Cir. 2001). Anticipation does not require that the reference "teach" the subject matter of the patent. It is necessary only that the claims being challenged "read on" something that is disclosed in the reference.

Celeritas Techs., Ltd. v. Rockwell Int'l, 150 F.3d 1354, 1361 (Fed. Cir. 1998).

Section 102 provides that, depending on the circumstances, a claimed invention may be anticipated by variety of prior art, including publications, earlier-sold products, and patents. *See* 35 U.S.C. § 102. Anticipation, like all forms of patent invalidity, must be established by clear and convincing evidence. *Glaxo Inc. v. Novopharm Ltd.*, 52 F.3d 1043, 1047 (Fed. Cir. 1995). Whether a patent claim is anticipated is a question of fact. *See Smith Kline Beecham Corp. v. Apotex Corp.* 403 F.3d 1331, 1343 (Fed. Cir. 2005).

2. Obviousness

Obviousness is grounded in 35 U.S.C. § 103, which provides, *inter alia*, that:

> A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

35 U.S.C. § 103(a).

An allegation of obviousness is evaluated under the so-called *Graham* factors: (1) the scope and content of the prior art; (2) the differences between the prior art and the claims at issue; (3) the level of ordinary skill in the art; and (4) objective evidence of nonobviousness, the so-called "secondary considerations," *e.g.*, commercial success, long felt need, and failure of others. *Graham v. John Deere Co.*, 383 U.S. 1, 13-17 (1966);

Dystar Textilfarben GmbH v. C.H. Patrick Co., 464 F.3d 1356, 1361 (Fed. Cir. 2006).[10]

"[E]vidence arising out of the so-called 'secondary considerations' must always when present be considered en route to a determination of obviousness." *Stratoflex, Inc. v. Aeroquip Corp.*, 713 F.2d 1530, 1538 (Fed. Cir. 1983). Secondary considerations, such as commercial success, will not always dislodge a determination of obviousness based on analysis of the prior art. *See KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 426 (2007) (commercial success did not alter conclusion of obviousness).

"One of the ways in which a patent's subject matter can be proved obvious is by noting that there existed at the time of invention a known problem for which there was an obvious solution encompassed by the patent's claims." *KSR*, 550 U.S. at 419-20. "[A]ny need or problem known in the field of endeavor at the time of invention and addressed by the patent can provide a reason for combining the elements in the manner claimed." *Id.*

Specific teachings, suggestions, or motivations to combine prior art may provide helpful insights into the state of the art at the time of the alleged invention. *Id.* at 420. Nevertheless, "an obviousness analysis cannot be confined by a formalistic conception of the words teaching, suggestion, and motivation, or by overemphasis on the importance of published articles and the explicit content of issued patents. The diversity of inventive pursuits and of modern technology counsels against limiting the analysis in this way." *Id.* "Under the correct analysis, any need or problem known in the field of endeavor at the time of invention and addressed by the patent can provide a reason for combining the

[10] "Before answering *Graham's* 'content' inquiry, it must be known whether a patent or publication is in the prior art under 35 U.S.C. § 102 – a legal question." *Panduit Corp. v. Dennison Mfg. Co.*, 810 F.2d 1561, 1568 (Fed. Cir. 1987).

elements in the manner claimed." *Id.* A "person of ordinary skill is also a person of ordinary creativity." *Id.* at 421.

The Federal Circuit has harmonized the *KSR* opinion with many prior circuit court opinions by holding that when a patent challenger contends that a patent is invalid for obviousness based on a combination of prior art references, "the burden falls on the patent challenger to show by clear and convincing evidence that a person of ordinary skill in the art would have had reason to attempt to make the composition or device, or carry out the claimed process, and would have had a reasonable expectation of success in doing so." *PharmaStem Therapeutics, Inc. v. ViaCell, Inc.*, 491 F.3d 1342, 1360 (Fed. Cir. 2007); *see KSR*, 550 U.S. at 416 (a combination of elements must do more than yield a predictable result; combining elements that work together in an unexpected and fruitful manner would not have been obvious).[11]

The ultimate determination of whether an invention would have been obvious is a legal conclusion based on underlying findings of fact. *In re Dembiczak*, 175 F.3d 994, 998 (Fed. Cir. 1999).

3. Indefiniteness

The definiteness requirement of 35 U.S.C. § 112 ensures that the patent claims particularly point out and distinctly claim the subject matter that the patentee regards to be the invention. *See* 35 U.S.C. § 112, ¶ 2; *Metabolite Labs., Inc. v. Laboratory Corp. of America Holdings*, 370 F.3d 1354, 1366 (Fed. Cir. 2004). If a claim's legal scope is not

[11] Further, "when the prior art teaches away from combining certain known elements, discovery of a successful means of combining them is more likely to be nonobvious." *KSR*, 550 U.S. at 416 (citing *United States v. Adams*, 383 U.S. 39, 52 (1966)).

clear enough so that a person of ordinary skill in the art could determine whether or not a particular product infringes, the claim is indefinite, and is, therefore, invalid. *Geneva Pharm., Inc. v. GlaxoSmithKline PLC*, 349 F.3d 1373, 1384 (Fed. Cir. 2003). Thus, it has been found that:

> When a proposed construction requires that an artisan make a separate infringement determination for every set of circumstances in which the composition may be used, and when such determinations are likely to result in differing outcomes (sometimes infringing and sometimes not), that construction is likely to be indefinite.

Halliburton Energy Servs. v. M-I LLC, 514 F.3d 1244, 1255 (Fed. Cir. 2008).

D. Domestic Industry

Complainants must demonstrate that an industry, relating to the articles protected by the asserted patent, exists or is in the process of being established within the United States. 19 U.S.C. § 1337(a)(2). The domestic industry requirement consists of an economic prong (an actual industry in the United States) and a technical prong (that industry must relate to articles protected by the intellectual property being asserted). *See Certain Ammonium Octamolybdate Isomers*, Inv. No. 337-TA-477, Comm'n Op. at 55, USITC Pub. 3668 (Jan. 2004).

With respect to the technical prong of the domestic industry requirement, complainant must demonstrate that it or its licensee is exploiting or practicing the patents at issue. *Certain Microlithographic Machines and Components Thereof*, Inv. No. 337-TA-468, ID (unreviewed in relevant part) at 63 (Public Version, Apr. 1, 2003) ("*Microlithographic Machines*"). The test for claim coverage for the purposes of the domestic industry requirement of Section 337 is the same as that for infringement: "Only

WI-Apple1604822

if the complainant's article or process is covered by the claims of the patent at issue, is the patent exploited for purposes of the domestic industry requirement of § 337." *Certain Doxorubicin and Preparations Containing Same*, Inv. No. 337-TA-300, ID at 109 (May 21, 1990), *aff'd*, Views of the Commission at 22 (Oct. 31, 1990). "It is sufficient to show that the domestic industry practices any [one] claim of [each of the asserted patents]." *Microlithographic Machines* at 64; *see Certain Video Graphics Display Controllers and Products Containing Same*, USITC Pub. 3224, Inv. No. 337-TA-412, Initial Determination (Unreviewed Portion) at 13-14 (Aug. 1999).

VII. U.S. Patent No. 6,343,263

The '263 patent is entitled, "Real-Time Signal Processing System For Serially Transmitted Data." JX-6. The '263 patent "is directed to the transmission of data to and from a computer, and more particularly to a system for performing real-time signal processing of data that is serially transmitted to and from a computer." *Id.* at col. 1, lns. 4-7 (Field of the Invention). The invention of the '263 patent "enables any arbitrary type of data, such as voice, facsimile, multimedia and the like, which is transmitted over any type of communication network, to be handled with any type of real-time engine, by abstracting the functions of each of the elements of the systems from one another." *Id.* at col. 2, ln. 66 – col. 3, ln. 6 (Brief Statement of the Invention).

Apple asserts independent claim 1 and dependent claims 2, 24, and 29. The asserted claims read as follow:

> **1**. A signal processing system for providing a plurality of realtime services to and from a number of independent client applications and devices, said system comprising:

a subsystem comprising a host central processing unit (CPU) operating in accordance with at least one application program and a device handler program, said subsystem further comprising an adapter subsystem interoperating with said host CPU and said device;

a realtime signal processing subsystem for performing a plurality of data transforms comprising a plurality of realtime signal processing operations; and

at least one realtime application program interface (API) coupled between the subsystem and the realtime signal processing subsystem to allow the subsystem to interoperate with said realtime services.

2. The signal processing system as set forth in claim **1**, wherein said signal processing system receives and transmits a plurality of datatypes over a plurality of different wide area networks (WANs).

24. The signal processing system of claim **1**, wherein said realtime signal processing subsystem comprises:

a realtime processor including an operating system for executing a plurality of realtime functions;

a realtime communications module which is independent of said realtime processor and is coupled to receive a plurality of communications commands from said application programs via said device handler program and said realtime API, said realtime communications module operating in response to said communications commands to issue a plurality of requests for realtime services to said realtime processor; and

a translation interface program which is specific to said realtime processor and is coupled to receive said requests for realtime services from said communications module and provide said requests to said realtime processor.

29. The signal processing system of claim **24**, wherein said realtime processor is embodied in a hardware device and includes realtime function libraries that are embodied in programmable software.

WI-Apple1604824

JX-6 at col. 11, lns. 28-47; col. 13, ln. 59 – col. 14, ln. 8; col. 14, lns. 22-25.

A. Claim Construction[12]

1. "realtime"

Claim Term	Apple Construction	Staff Construction	HTC Construction
"realtime" (claims 1, 24, 29)	within a defined upper bounded time limit	within a fixed upper bounded time limit	must be processed within a fixed time limit and without data handling delays

Joint Claim Construction, App'x A at 14.

Complainants argue that the proper construction of the claim term "realtime" is "within a defined upper bounded time limit." Apple Br. at 14; Motion No. 710-114, App'x A at 14 (Joint Motion of All Relevant Parties to Amend the Joint List of Undisputed Claim Terms With Agreed Constructions, Apple's Corrected Proposed Claim Construction Chart, HTC's Proposed Claim Constructions, and the Staff's Proposed Claim Constructions) (Feb. 24, 2011) ("Joint Claim Construction"); *see* Order 93 (granting Motion No. 710-114) (Mar. 1, 2011).

HTC argues that claim term "realtime" should be construed to mean that data "must be processed within a fixed time limit and without data handling delays." HTC Br. at 13; Joint Claim Construction, App'x A at 14. The Staff submits that the claim term "realtime" should be construed to mean that data should be processed "within a fixed upper bounded time limit." Staff Br. at 24; Joint Claim Construction, App'x A at 14.

Thus, the parties at least agree that processing in "realtime" involves processing

[12] A person of ordinary skill in the art would have a B.S. degree in computer science, or the equivalent, and at least two to three years of experience in signal processing systems. Polish Tr. 337; Brandt Tr. 1338; Staff Br. at 23 n.6.

WI-Apple1604825

time-sensitive data in a manner consistent with its upper-bounded time limit. The dispute is whether these time limits must be "defined" (Apple's proposal) or "fixed" (respondents' and Staff's proposal), and whether these time limits "must" be met without any processing delays (respondents' proposal).

As proposed by Apple, the claim term "realtime" is construed to mean "within a defined upper bounded time limit."[13]

Asserted independent claim 1 contains several claim limitations that include the disputed term "realtime" (*i.e.*, "realtime services," "realtime signal processing

---

[13] HTC notes that "[t]hree claimed realtime components require construction: 'realtime,' 'realtime signal processing subsystem,' and 'realtime application programming interface.'" HTC Br. at 13. HTC and the Staff argue that "[b]ecause the parties have agreed that 'realtime' in isolation requires time limits, it necessarily follows that components modified by 'realtime' — the signal processing system and API — must also be realtime, that is, have time limits." HTC Br. at 15; Staff Br. at 25-26.

Apple characterizes this argument as "grammatical gamesmanship." Apple Reply at 6. Complainants' submit: "The word 'realtime' is used to modify eight different terms in the asserted claims that describe widely-varying components and also 'services.' No basis exists to import separate fixed time limits into each component regardless of context, much less into 'services,' especially when the patent and preferred embodiment never mention such component-based limits." *Id.*

HTC's and the Staff's contention is rejected as being contrary to the plain reading of the claim language. Moreover, as argued by Apple (Reply at 7) this theory would lead to illogical results. For example, claim 24 recites "a realtime processor including an operating system." Thus, if HTC and the Staff were correct, the "realtime processor" would have fixed time limits, but its *non*-realtime *operating system* would not. JX-6 at col. 13, ln. 61. Further, the preamble ("signal processing system for providing a plurality of realtime services") would not understand time limits, but the realtime subsystem that comprises the signal processing system to provide realtime services would. JX-6 at col. 11, ln. 29. Finally (although not exhaustive), if HTC and the Staff were correct, claim 1's unmodified "plurality of data transforms" would not understand time limits, but the "plurality of realtime signal processing operations" it is comprised of must.

In sum, because HTC's argument is a formalistic word play that defeats the purpose of the invention and leads to nonsensical results, it is rejected. *See Howmedica Osteonics Corp v. Tranquil Prospects, Ltd.*, 401 F.3d 1367, 1372 (Fed. Cir. 2005) (rejecting claim interpretation that would defeat the "overriding purpose of the invention").

WI-Apple1604826

subsystem," "realtime signal processing operations," and "realtime application program interface (API)"). The specification of the '263 patent explains that "[t]he present invention is directed to the transmission of data to and from a computer, and more particularly to a system for performing real-time signal processing of data that is serially transmitted to and from a computer." JX-6 at col. 1, lns. 4-7 (Field of the Invention) (emphasis added). This statement that the particular invention of the '263 patent is directed to "a system for performing real-time signal processing of data" is entirely consistent with the preamble of asserted claim 1, *i.e.*, "[a] signal processing system for providing a plurality of realtime services." JX-6 at col. 11, lns. 28-30. In other words, a signal processing system that provides "realtime services" is a system that performs "real-time signal processing of data."

While the words "realtime," "real-time," and "real time" are disclosed throughout the specification of the '263 patent, including the claims, most of the disclosures are associated with the phrase "real-time engine" of the exemplary embodiments of the invention. The exemplary embodiments, as the patent explains, are "described with reference to the specific example of a telephone-based telecommunication subsystem that provides basic fax/data modem services, plus call management and audio stream handling." JX-6 at col. 3, lns. 43-47. The patent further explains that "[o]ther implementations of the invention, for example in the context of transmitting sounds and video data, will become apparent from an understanding of the principles of the invention explained with respect to this particular example." *Id.* at col. 3, lns. 54-58.

Significantly, the specification explains the broad and flexible applicability of the claimed invention using a "real-time engine." The specification states:

> The invention enables any arbitrary type of data, such as voice, facsimile, multimedia and the like, which is transmitted over any type of communication network, to be handled with any type of real-time engine, by abstracting the functions of each of the elements of the system from one another. This abstraction is provided through suitable interfaces that isolate the transmission medium, the data managers and the real-time engine from one another.

JX-6 at col. 2, ln. 66 – col. 3, ln. 6 (emphasis added).

The specification also explains the usefulness of the claimed invention in any system that transmits and processes data at "real-time rates" including "any type of data acquisition system." It provides:

> In essence, the real-time engine allows any type of transform to be performed on any type of data delivered over any type of transmission medium.
>
> *****
>
> .... [T]he data is delivered at a real-time rate, where it is handled by the computer's CPU.
>
> *****
>
> .... [T]he invention will find utility in any environment in which it is desirable to transmit and process data at real-time rates. Thus, while the invention has been described in the context of communications over a wide-area network, it can be used in any type of data acquisition system.

JX-6 at col. 10, ln. 61 – col. 11, ln. 21.

Therefore, as disclosed by the specification, the invention of the '263 patent is a system in which "the data is delivered at a real-time rate, where it is handled by the computer's CPU" and it "will find utility in any environment in which it is desirable to transmit and process data at real-time rates."

While repeatedly emphasizing the flexibility of the invention, the patent

WI-Apple1604828

distinguishes the "hardwired" prior systems that did not provide such flexibility. *Id.* at col. 1, lns. 30-32; col. 11, lns. 7-10. Requiring only a "defined" upper bounded limit conforming to different data types or communication networks, as proposed by Apple, is consistent with the patent's description and emphasis on flexibility, whereas HTC's and the Staff's restrictive construction requiring "fixed" limits that "must" be met is not. Moreover, the '263 patent does not mention "fixed" time limits or explicit clocking requirements tied to system components.[14]

Furthermore, there is no support in the prosecution history for the additional limitation that respondents seek to add (*i.e.*, "without additional handling delays"). Even though the prior art described processing delays as part of their realtime implementations (*see, e.g.*, RX-1103 at col. 17, lns. 19-37), the '263 applicants never distinguished this art on the basis that their use of "realtime" was limited to "fixed" time limits that must always be met without processing delays. Polish Tr. 1677.[15]

HTC's expert, Dr. Brandt, presented opinions on "realtime" that were inconsistent with both the intrinsic evidence and his own writings. Under cross-examination,

---

[14] The only reference to time management in the patent is in the specific context of the "native" DSP (digital signal processor) implementation where the DSP functions are carried out by the host CPU, noting that adequate CPU processing time should be ensured for DSP operations. JX-6 at col. 7, lns. 5-14; Brandt Tr. 1447-48.

[15] The extrinsic evidence further confirms Apple's proposed construction. Dr. Polish (Apple's expert) explained the various forms of realtime processing (Tr. 328-31), the proper construction of the term encompassing all such forms (Tr. 337-38), and why respondents' proposed construction is inconsistent with the intrinsic evidence and how persons of ordinary skill in the art use this term and would interpret the patent (Tr. 338-45). Nokia's expert, Dr. Gottesman, testified that he did not have "a problem with Apple's construction." Gottesman Tr. 1555. Dr. Gottesman further agreed that the time constraints in a real-time system are set by the input, *i.e.*, the data, and not something explicitly hard-coded within the computer itself. Tr.1552.

WI-Apple1604829

Dr. Brandt effectively admitted that he was proffering a construction of realtime that was historically limited to applications like flight control systems – where failure to meet a processing deadline generally means application or system failure – not at all applicable to desktop realtime applications that are the subject of the patented invention. Brandt Tr. 1455-58; CX-7294 at 4. Although respondents' proposed construction seeks to impose fixed deadlines that "must" be met, Dr. Brandt's own writings confirm that desktop realtime applications (such as streaming video or audio) ***do not*** require such strict performance guarantees. Brandt Tr. 1459-60 ("Q. You also said, at CDX-4002, page 4 of your thesis, that desktop realtime applications are not critical. A. Yes. Q. 'These processes do not need such strict performance guarantees, only a reasonable assurance that their resource needs will be largely met by the operating system.' You said that, right? A. Yes, exactly."); CX-7294 at 2, 4.

Also on cross-examination, Dr. Brandt conceded that "soft realtime processing" is a "type of realtime processing" that specifically permits missing deadlines and pertains to the types of applications specifically described in the '263 patent. Brandt Tr. 1460-63; CX-7294 at 4.

Furthermore, Dr. Brandt's reliance on the Laplante book as support for the HTC's construction of "realtime" is misplaced. Laplante makes clear that the term "realtime" evolved from the 1950s, when it was limited to time-critical hard realtime applications such as flight control and missile defense systems, to the relevant time period of the 1980s and 1990s, when it was commonly understood to apply to a much broader array of applications. RX-270 (Phillip Laplante, *It Isn't Your Father's Realtime Anymore* (2006)) ("For the next 30 years or so, the term realtime was applied only to industrial control,

weapons systems, and other exotic applications, all of which were essentially characterized as those where inability to meet deadlines led to failure —usually a spectacularly catastrophic one."); *see* Brandt Tr. 1466-67.

In that regard, Laplante explains:

> In short, a system does not have to process data in microseconds (millionths of a second) to be considered real-time: it must simply have response times that are constrained and thus predictable.
>
> It can be argued that all practical systems are real-time systems. It should be evident that even a batch-oriented system – the kind many insurance companies now use to process automobile insurance punch cards – is real-time. Although the system may have response times of days or weeks (the time between when you mail your card and are sent your insurance certificate), it must respond within a certain time or your insurance will lapse – a disaster. Even a word-processing program should respond to your commands within a reasonable amount of time (e.g., 1 second) or it will become torture to use. Most of the literature refers to such systems as *soft real-time systems*; that is, systems where performance is degraded but not destroyed by failure to meet response time constraints. Furthermore, systems where failure to meet response time constraints leads to system failure are called *hard real-time systems*. Recently, the term *firm real-time systems* has been defined to include those systems with hard deadlines where some low probability of missing a deadline can be tolerated. As noted, all practical systems minimally represent soft real-time systems. Since we are most interested in the so-called hard real-time systems, we will use the term "real-time system" to mean hard real-time system without loss of generality.

RX-777 at HTC007769262 (Phillip A. Laplante, *Real-Time Systems Design and Analysis* (1993) at 11) (italics in original, underlines added).

Thus, a person of ordinary skill at the time of the '263 patent, *i.e.,* 1994, would apply the evolved definition of "realtime"described by Laplante, and not the earlier,

outdated "hard realtime" definition. Thus, the Laplante references confirm that the proper construction of "realtime" is "within a defined, upper bounded time limit."

a. "a realtime signal processing subsystem"

Claim Term	Apple Construction	Staff Construction	HTC Construction
"a realtime signal processing subsystem" (claims 1, 24, 29)	Plain meaning	Plain meaning	A processor and optionally software that processes and handles data in a realtime manner

Joint Claim Construction, App'x A at 14.

As proposed by Apple and the Staff, the claim term "realtime signal processing subsystem" carries a plain and ordinary meaning.[16] "Realtime signal processing subsystem" is construed to mean "a subsystem capable of processing signals in realtime." Claim 1 recites two subsystems. HTC has never contended that the word "subsystem" is unclear. The terms "signal" and "processing" are standard computer science/engineering terms understood by ordinary artisans and laypeople alike and are similarly clear. Accordingly, this term does not require construction because it conveys that it is a subsystem that is capable of processing signals in realtime. Polish Tr. 349-50.

In fact, neither of respondents' experts (Dr. Brandt for HTC and Dr. Gottesman for Nokia, which had settled out) offered any explanation of the need for construction of the term. Moreover, respondents' proposed construction apparently requires an

[16] Inasmuch as the Staff's proposed construction for this claim term is simply "realtime signal processing subsystem," it is understood that the Staff is proposing a plain and ordinary meaning for the term.

WI-Apple1604832

additional processor, which is expressly inconsistent with the native implementation of a DSP described in the specification and within the scope of claims 1 and 24.

Thus, the claim term "realtime signal processing subsystem" is self-descriptive and needs no further construction. *CCS Fitness, Inc. v. Brunswick Corp.*, 288 F.3d 1359, 1366 (Fed. Cir. 2002) ("[Courts] indulge a 'heavy presumption' that a claim term carries its ordinary and customary meaning.").

b. "realtime API"

Claim Term	Apple Construction	Staff Construction	HTC Construction
"realtime application programming interface (API)" (claims 1, 24)	API that allows realtime interaction between two or more subsystems	Same as Apple	A realtime software module that receives commands from applications and generates commands

Joint Claim Construction, App'x A at 14.

As proposed by Apple and the Staff, the claim term "realtime API" is construed to mean an "API that allows realtime interaction between two or more subsystems."

The '263 patent shows that the "realtime API" of the asserted claims is an API (1) positioned at the interface of the realtime signal processing subsystem, and (2) allowing for the provision of realtime services. The specification does not explicitly mention any time constraints or clock for the realtime API. Rather, all references in the preferred embodiment are to an API that is positioned as the interface to the realtime processing subsystem that allows for realtime services to be provided by that subsystem. *See* JX-6 at col. 6, lns. 26-38 ("As illustrated in FIG. 2, there can be a number of interfaces 48 situated between the handler 44 and the real-time engine 46."); col. 7, lns. 52-67

("interface 48").

The claim phrase "realtime application program interface (API)" or "realtime API" is not disclosed anywhere else in the specification of the '263 patent. Rather, the phrase "API" or "application program interface (API)" without the modifier "realtime," is used to describe the "realtime API" in the specific embodiments. While the Inventor does not use the word "realtime" to modify the "API" that is disclosed in the detailed description portion of the specification, it is clear that one of ordinary skill would find the description of "API" as applying to "realtime API."

For example, in describing the specific embodiment as shown in FIG. 2, the specification of the '263 patent discloses:

> A real-time engine 46 can perform transforms on data streams provided to and received from the adapter 36. The particular transforms to be performed are sent as commands to the real-time engine from the adapter handler 44 via suitable application programming interfaces 48. For communicating with the real-time engine, each interface includes shared command/control mailboxes in the computer's RAM, as well as bi-directional first-in, first-out (FIFO) buffers for transferring data.
>
> *****
>
> As illustrated in FIG. 2, there can be a number of interfaces 48 situated between the handler 44 and the real-time engine 46. Each interface represents services for a particular class of functionality. For example, one interface may relate to the operation of the engine as a virtual telephone, another interface can be associated with a virtual sound device, e.g. stereo, and a third interface can pertain to a virtual video device. Each interface receives commands from an application program, through the handler 44, and instructs the real-time engine to carry out the necessary transforms which relate to the function of the virtual device being implemented, e.g. text-to-speech conversion, video image processing, etc.

JX-6 at col. 5, lns. 21-29; col. 6, lns 25-38 (emphasis added).

Thus, the above specification portion makes clear that in the context of the overall communication system that uses a "real-time engine 46," "the application programming interfaces 48" are indeed able to communicate in a realtime manner. The specification states that in "communicating with the real-time engine," each API 48 is used "for transferring data."

Similarly, the '263 patent discloses the following with respect to the application programming interfaces (APIs) in describing a specific embodiment of the invention:

> The ability to communicate over different types of transmission mediums in this single system is made possible by the fact that each of the various components is isolated from the particular features of the other through suitable levels of abstraction implemented via the application programming interfaces. For example, to change the transmission medium from the telephone lines to an ISDN line, the telecom adapter 36 is disconnected from the serial port 37, and a new adapter appropriate for ISDN is plugged into the serial port. The associated adapter handler 44 is also loaded into the system. Thereafter, whenever the adapter handler issues a command to the real-time engine to perform a transform, it identifies the fact that the transformed data must be suitable for ISDN format. In response thereto, the API 48 which receives these commands supplies the real-time engine with the appropriate parameters for performing the transforms in the required format, e.g. the proper number of bits per word, etc.
>
> Similarly, if the computer is transported from one country to another, the only change that needs to be implemented to carry out telephone communications in the new country is to switch the adapter and its handler. Upon initialization, the adapter identifies the fact that it is designed for a specific country. Whenever commands are to be sent to the real-time engine, the handler instructs the API 48 of the country as well as the command itself. For example, the

WI-Apple1604835

> command might be to generate a dial tone for country X. In response, the API 48 instructs the real-time engine to generate the dial tone, and provides it with the parameters pertinent to dial tones in country X.

JX-6 at col. 10, lns. 27-56 (emphasis added).

Again, the specification shows that APIs are positioned as the interface to the realtime processing subsystem that allows for realtime services to be provided by that subsystem. Further, even though the APIs are disclosed without the modifier "realtime," one of skill in the art would understand the above description is describing "realtime" APIs.

Moreover, all of HTC's witnesses admitted that an API serves as an interface to access lower-level functionality while abstracting the details of how that functionality is implemented. *See* Apple Br. at 20, 37-40. These witnesses also admitted that an API can be defined by a set of software functions defined in a ".h" header file. *See, e.g.*, Sherwood Tr. 858-59; Sparks Tr. 1067-68; Gottesman Tr. 1562.

In addition, HTC argues that the "'realtime application program interface' must also be a software module that generate commands." HTC Br. at 18. In support of this argument, respondents cite to the '263 patent (JX-6 at col. 6, lns. 33-38 & col. 10, lns. 53-56), as well as to the testimony of Inventor John Lynch (Tr. 244-46, 252, 277-78, 301-02) and the testimony of Apple's expert, Dr. Polish (Tr. 354-55). HTC Br. at 19. HTC also argues that complainants are estopped from arguing otherwise because "Apple distinguished the purported invention from the Chen patent on the grounds that the claimed API generated commands rather than merely routing commands received from elsewhere." *Id.*, citing JX-11 at APPHTC_00013716 & 000135773; Brandt Tr. 1321-22.

WI-Apple1604836

For the reasons that follow, HTC's arguments must fail. First, the prosecution history statements cited by respondents confirm that **all** APIs generate commands, and that "[a]n element which routes API commands to other elements is ***not the same as an API itself***." HTC Br. at 19 (citing JX-11 at APPHTC_00013716 (emphasis added)). As Inventor John Lynch explained, "[a]n API is a set of software functions, and executing those functions is issuing the commands." Lynch Tr. 302-303. In other words, generating commands means executing the functions in an API.

Second, HTC veers away from the plain meaning of the words "realtime API" when it argues, via its proposed claim construction requiring a separate "software module," that a typical header-file-defined API is not coupled between the subsystems for which it provides an interface. The API is indeed functionally connected to the object(s) for which it provides an interface, and HTC does not contend otherwise. But this API is also "distinct from" the objects for which it provides an interface in the sense that it is the only aspect of the object exposed to the higher-level components (Sherwood Tr. 857-59; Sparks Tr. 1091-92), is defined separately in a header file (Sherwood Tr. 858-59; Sparks Tr. 1067; Gottesman Tr. 1562), and can provide a generic interface for multiple different objects of a similar type (Sparks Tr. 1091 (explaining that header file defined interface for multiple nodes)).

Additionally, the fact that the defined functions become part of the objects for which they provide an interface upon compilation does not transform the functions such that they are no longer APIs. Sherwood Tr. 853, 857-59. HTC's own documents confirm this. *See* CX-6947C at ln. 316 ("This API is to allow for extensibility of the PVMF Node interface.").

Finally, HTC's reliance on the prosecution history to estop Apple from arguing that APIs route commands is equally unavailing. Nothing in the prosecution history approaches the required "clear and unmistakable" surrender by Apple. *Purdue Pharma L.P. v. Endo Pharms. Inc.*, 438 F.3d 1123, 1136 (Fed. Cir. 2006) (disavowal of claim scope must be "clear and unmistakable" during prosecution).

Contrary to HTC's argument, the Chen reference was not overcome during prosecution based on an argument that it disclosed an API, just the wrong type. Instead, the Chen reference was overcome because the DSP Manager 71, fundamentally, "***is not an API.***" JX-11 at APPHTC_00013529-30, 3544, 3713-16, 3773-74.

The basis for Apple's distinction of the Chen reference was that Chen's DSP Manager 71 was "not an API" and "does not include an API." JX-11 at APPHTC_00013716, 73. While the Board of Patent Appeals and Interferences ("BPAI") of the USPTO also found that Chen's DSP Manager 71 "did not generate API commands," this is because a non-API does not generate API commands. *See* Apple Br. at 20; Apple Reply at 15-17. Indeed, respondents themselves rely on a statement in the prosecution history explaining that "[a]n element which routes API commands to other elements is not the same as an API itself." HTC Br. 19. Chen's DSP Manager 71 was not an API, could not generate API commands, and could only perform the function of "routing" commands. JX-11 at APPHTC_00013773.

The DSP Manager 71 in Chen did not include the claimed API because Chen solved a different problem. JX-11 at APPHTC_00013713. Chen explicitly sought to improve upon prior, single soft DSP systems, such as VCOS, by managing multiple soft DSPs. RX-1103 at col. 1, ln. 58 – col. 2, ln. 11. Specifically, Chen only teaches that the

DSP Manager 71 routed requests from an application to load a particular task onto the corresponding soft DSP, *e.g.*, load task H onto DSP #3. JX-11 at APPHTC_00013713; Polish Tr. 1668-69. Under this architectural arrangement, the application knows that the system includes several DSPs, specifically only soft DSPs, knows which tasks each DSP supports, and knows that it is loading a chosen task onto a specific DSP. *See id.* Chen's DSP Manager 71 abstracts none of these details. Thus Apple argued, and the BPAI agreed, that DSP Manager 71 is not an API, does not generate API commands, and does not include an API. JX-11 at APPHTC_00013716, 73.

HTC has not identified any statements made during prosecution that rise to the level of establishing a clear and unmistakable surrender of claim scope.[17] Apple did not argue that the Chen reference lacked a realtime API by clearly and unmistakably disavowing its plain and ordinary meaning. Rather, Apple prevailed in prosecution because Chen's DSP Manager 71 was not an API at all. The '263 patent overcame Chen during prosecution because Chen's DSP Manager was not an API providing the requisite layer of abstraction. The distinction ***cannot*** have been that Chen's DSP Manager was only an ordinary header file API because it ***wasn't a header file at all***—it was a software module. RX-1103 at FIG. 22 (showing active internal functions of DSP Manager 71 software module).

[17] *Purdue Pharma L.P. v. Endo Pharms. Inc.*, 438 F.3d 1123, 1136 (Fed. Cir. 2006) (disavowal of claim scope must be "clear and unmistakable" during prosecution); *Rexnord Corp. v. Laitram Corp.*, 274 F.3d 1336, 1347 (Fed.Cir.2001) (refusing to limit the ordinary meaning of the claim because the alleged disclaimer in the file wrapper was at best "inconclusive"); *Northern Telecom Ltd. v. Samsung Electronics Company*, 215 F.3d 1281, 1293-95 (Fed. Cir. 2000) (same).

Like the "realtime" limitation itself, HTC also veers from the plain meaning of the "realtime API" limitation. But HTC does not identify a clear and unmistakable disclaimer of the plain meaning of "realtime API" during prosecution or elsewhere, and documents and testimony show that the products have realtime APIs providing an interface to, and abstracting the details of, their realtime processing subsystems.

Respondents propose a construction that the "realtime API" must be a stand-alone software module, ostensibly excluding typical APIs such as functional interfaces defined by object-oriented header files. This is inconsistent with the intrinsic evidence discussed above and the plain and ordinary meaning of the term.

The evidence established that the plain and ordinary meaning of the term "application program interface" includes functional interfaces defined by objected-oriented (*e.g.* C++) header files. [

] [18] Google's manager for the Android multimedia framework, David Sparks, confirmed that "APIs are designed to abstract the implementation details" and that ".H" header files defined the APIs for the nodes in the Android products. Sparks Tr. 1091-92.

---

[18] [

]

WI-Apple1604840

Even Nokia's expert, Dr. Gottesman, agreed that in "the general usage of the term 'API,' a person of skill in the art would consider that to include this concept of header files." Gottesman Tr. 1562. Further, contrary to respondents' contention that APIs must be a separate "software module," Dr. Gottesman was impeached with his unequivocal deposition testimony that "absolutely, for decades" persons of ordinary skill in the art in 1994 understood that "an API itself would not be a running piece of software." *Id.*, Tr. 1563-64.

Based on the reasons set forth above, the claim term "realtime API" is construed to mean an "API that allows realtime interaction between two or more subsystems."

2. "device handler"

Claim Term	Apple Construction	Staff Construction	HTC Construction
"device handler" (claims 1, 24)	software associated with an interface device that sets up dataflow paths, and also presents data and commands to a realtime signal processing subsystem	same as Apple	a software module specific to a device that sets up dataflow paths, and presents data and commands to the realtime signal processing subsystem

Joint Claim Construction, App'x A at 14.

Apple and the Staff construe the claim term "device handler" to mean "software associated with an interface device that sets up dataflow paths, and also presents data and commands to a realtime signal processing subsystem." Joint Claim Construction, App'x A at 14.

Respondents differ with this construction insofar as they insert the term "specific to" in place of "associated with" for the software's relation to the interface device. *Id.*

WI-Apple1604841

Respondents argue that this narrower language requires a unique relationship between a device handler and a particular device. HTC also interprets the agreed language "presents data and commands" to require direct handling or processing of data by the device handler.

The claim term "device handler" is construed to mean "software associated with an interface device that sets up data flow paths, and also presents data and commands to a realtime signal processing subsystem."

As for the "associated with" (Apple/Staff) versus "specific to" (HTC) dispute, Apple's and the Staff's construction is consistent with the '263 patent's Abstract and Brief Statement of the Invention. JX-6 Abstract ("A device handler ***associated with*** the interface device sets up dataflow paths, and also presents data and commands from the data managers to a real-time data processing engine.") (Emphasis added); *Id.* at col. 2, lns. 57-59 ("a device handler ***associated with*** the interface device sets up data paths and issues service requests.") (Emphasis added).

Non-asserted independent claim 31 provides additional support for Apple's and the Staff's position. Claim 31 recites "a device handler program ***associated with*** said input/output device." (Emphasis added). The term "specific to" cannot be narrower than "associated with" (as HTC suggests) because the unmodified term "device handler program" in claim 1 cannot be more restricted than the modified "device handler program" in claim 31. *Renishaw PLC v. Marposs Societa' per Azioni*, 158 F.3d 1243, 1249-50 (Fed. Cir. 1998) ("Nor may we, in the broader situation, add a narrowing modifier before an otherwise general term that stands unmodified in a claim.").

HTC derives its proposed language from a misreading of the description of the preferred embodiment: "An adapter handler 44 is specific to the particular adapter 36 and carries out features associated with that adapter." JX-6 at col. 5, lns. 9-10. Unlike the Brief Statement of the Invention, this language is anchored to the preferred embodiment and thus an inappropriate source for limiting the claims. *Innova/Pure Water, Inc. v. Safari Water Filtration Sys., Inc.*, 381 F.3d 1111, 1117 (Fed. Cir. 2004) ("particular embodiments appearing in the written description will not be used to limit claim language that has a broader effect"); *Seachange Intern, Inc. v. C-COR, Inc.*, 413 F.3d 1361, 1377 (Fed. Cir. 2005) ("[W]e do not import limitations from a preferred embodiment.").

Moreover, even the language respondents cite does not mean that the device handler must be "specific to" in the sense that it has special knowledge of a particular device. As the context of the Detailed Description reveals, there can be a "number of interfaces" connected to a handler, and the interfaces can correspond with different types of devices and/or networks. JX-6 at col. 6, lns. 26-38; col. 4, lns. 49-54 (adapter for "one or more communications networks").

Apple's expert, Dr. Polish, confirmed that the intrinsic evidence would lead a person of ordinary skill to understand the disputed term as Apple's and the Staff's proposed construction does. Polish Tr. 347-49. He also explained that it would not make sense to have the device handler be "specific to" a particular device to the exclusion of others (Polish Tr. 349), and noted that the device handler program was abstracted from the device by two layers (including a hardware abstraction layer) in the preferred embodiment (Polish Tr. 799-800). In contrast, HTC's expert failed to offer any opinion supporting its proposed construction.

As to the dispute about the meaning of the agreed language "presents data and commands," HTC's expert, Dr. Brandt, testified that the "presents data and commands to a realtime signal processing subsystem" language requires the device handler program to itself process and handle data and then send that data to the realtime signal processing subsystem. Brandt Tr. 1299-1300, 1304; *see* HTC Br. at 22. This new "interpretation" is, however, improper because it would exclude the preferred embodiment of the invention. *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1583 (Fed. Cir. 1996) ("Such an interpretation is rarely, if ever, correct and would require highly persuasive evidentiary support.").

The function of the device handler program is to set up data flow paths and present data and commands – *i.e.*, to orchestrate the flow of data and commands necessary for the data to be directed to the right place in the architecture. In setting up and coordinating flow through those paths, however, the device handler need not, and in fact in the preferred embodiment does not, process and handle the data itself. JX-6 at FIG. 2; col. 5, ln. 67 – col. 6, ln. 3 (data flows directly through DMA 50 and not adapter handler 44). In support for this position, Inventor John Lynch testified that the device handler in the actual embodiment of the invention did not handle data. Lynch Tr. 221. HTC's argument conflicts with the intrinsic evidence and is rejected.

Accordingly, the claim term "device handler" is construed to mean "software associated with an interface device that sets up dataflow paths, and also presents data and commands to a realtime signal processing subsystem."

WI-Apple1604844

B. Infringement

1. Claim 1

Apple argues that the Android multimedia architecture, including PacketVideo's OpenCore PVPlayer[19] and/or Stagefright Player,[20] as implemented on all of the HTC products, infringe asserted claim 1 of the '263 patent. Apple Br. at 26.

HTC argues that the accused devices are missing several key limitations of claim 1 and thus do not infringe. According to respondents, the HTC phones lack the claimed "device handler," an API "coupled between" the host and realtime subsystems, and the claimed realtime components, *i.e.*, a realtime API, a realtime signal processing subsystem, and realtime services. HTC Br. at 20.

The Staff submits that under its constructions of the disputed claim terms, HTC's accused products do not satisfy the claim terms "realtime signal processing subsystem" and "realtime API" and, therefore, the accused products do not infringe claim 1. Staff Br. at 29-30.

For the reasons set forth below, Apple has shown that HTC's accused products infringe all asserted claims of the '263 patent.

The preamble of claim 1 recites:

> **A signal processing system for providing a plurality of realtime services to and from a number of independent client applications and devices, said system comprising:**

Apple has satisfied the preamble.

---

[19] [ ] is present in the accused HTC handsets based on Android versions 1.5, 1.6, 2.1, or 2.2. Polish Tr. 371-72; Sparks Tr. 1042-43.

[20] [ ] is present in the accused HTC handsets based on Android version 2.2. Polish Tr. 371-72; Sparks Tr. 1042-43.

WI-Apple1604845

The parties agreed that the claim term "realtime services" should be construed to mean "constant bit rate data handling in realtime."

Apple argues that HTC did not present any argument or evidence at trial that the preamble of claim 1 of the '263 patent should be found limiting and given this waiver, HTC cannot meet their burden of overcoming the "presumption against reading a statement of purpose in the preamble as a claim limitation." Apple Br. at 27, citing *Marrin v. Griffin*, 599 F.3d 1290, 1294-95 (Fed. Cir. 2010). Apple, nonetheless, asserts that the evidence adduced at the hearing proved that the HTC accused products include a signal processing system for providing a plurality of realtime services to and from a number of independent client applications and devices. *Id.*

The HTC accused products include client applications such as the YouTube video player and an audio player. CX-325 at 6; CX-453 at 1; Polish Tr. 368-69, 452-53; Sparks Tr. 1047. They include devices such as cellular and WiFi antennae. Polish Tr. 368-69. The HTC accused products provide, to and from the client applications and devices, realtime services such as audio and video playback, audio and video streaming from a network, audio and video encoding, and audio and video decoding. CPX-1; CPX-2; Polish Tr. 367-68.

HTC counters that "realtime services" as recited in the preamble is a claim limitation and that "Apple presented no evidence that the accused HTC phones provide 'realtime services'." HTC explains that the parties agreed that "realtime services" means "constant bit rate data handling in realtime," "[b]ut nothing in the record even hinted at

WI-Apple1604846

constant bit rate data handling." HTC Reply at 25-26.[21]

HTC's argument is without merit inasmuch as the testimony of Apple's expert, Dr. Polish, relating to the preamble of claim 1 is unrebutted. Polish Tr. 363 ("A. One of them is "realtime services," which has been an agreed construction of constant bit rate data handling in realtime... Q. And have you applied these agreed constructions in your analysis of the accused products [and] the domestic industry product in this case? A. Yes, I have."); Polish Tr. 369 ("Q. And do you understand the parties to be disputing in any way the satisfaction of these claim limitations [in the preamble of claim 1] with respect to the HTC accused products? A. No. It is my understanding that there is no dispute as to these elements.")

HTC did not challenge Dr. Polish on this point at the hearing. Thus, HTC cannot now argue that there is a dispute with respect to the preamble and its accused products.

The first element of claim 1 recites:

> **a subsystem comprising a host central processing unit (CPU) operating in accordance with at least one application program and a device handler program, said subsystem further comprising an adapter subsystem interoperating with said host CPU and said device;**

Apple has satisfied this claim element.

The claim term "device handler" has been construed to mean "software associated with an interface device that sets up dataflow paths, and also presents data and commands to a realtime signal processing subsystem."

[21] HTC did not address the preamble in its initial brief. The Staff does not dispute that the accused products satisfy the preamble.

WI-Apple1604847

HTC accused products include at least one [ ] central processing units (CPUs). Polish Tr. 368-69. The CPUs operate in accordance with application programs such as the YouTube video player or the music player. CX-325 at 6; CX-453 at 1; Polish Tr. 368-69, 452-53; Sparks Tr. 1047. HTC accused products include [

] Stipulation Regarding Issues Related to U.S. Patent No. 6,343,263 (Apr. 17, 2011) ("'263 Issues Stip."), ¶ 6; Polish Tr. 368-69.

The CPUs operate in accordance with a device handler program such as the [

] implementations. CX-325 at 6; CX-453 at ln. 16; CX-454 at ln. 50; Polish Tr. 371, 374-75, 460-61.

[

]

Both David Sparks, HTC's fact witness and Google's manager for the Android multimedia framework, and Drllip Sherwood, PacketVideo's chief software architect, confirmed that [ ] set up data flow paths and present data and commands to the realtime signal processing subsystem. CX-327 at 12; CX-307C at PV-92; Polish Tr. 372-73, 375, 383-85; Sherwood Tr. 822-23, 859-60; Sparks Tr. 1052-53, 1069-70; Sparks Tr. 1089 [

WI-Apple1604848

]

Moreover, the [ ] implementations are also associated with and specific to a device. CX-307C at PV-90, 92; Polish Tr. 387-88, 394-96. In particular, [

] The types of PVMF nodes used and the graph configuration would depend on the playback parameters such as source clip type and playback operation." CX-486C at 10. Dr. Sherwood confirmed that [

]

HTC argues that its accused devices do not infringe because they do not satisfy two of the four requirements for a device handler[22] – they do not present data to the accused realtime signal processing subsystem, and they are not "specific to" or "associated with" an interface device. HTC Reply at 12.

Regarding the [ ] HTC contends that Dr. Brandt confirmed that the [ ] implementation does not present data to the accused signal processing subsystem and data never flows through the [ ] implementation, but instead, the [ ] implementation sets up the[ ] and these [ ] receive and

[22] The claim term "device handler" has been construed to mean "software associated with an interface device that sets up dataflow paths, and also presents data and commands to a realtime signal processing subsystem."

WI-Apple1604849

pass data. HTC Br. at 22. Similarly, HTC claims that the accused device handler in [ ] do not present any data because they never receive data in the first place. *Id.*

HTC is incorrect in arguing that because the claimed "device handler program" allegedly "presents data . . . to the realtime signal processing subsystem," the media data must be "passed through" the device handler program. This additional "passed through" requirement ignores the remainder of the specification, and specifically the preferred embodiment, to improperly re-define the proposed construction to require that the media data be "passed through" the device handler program.

This argument represents an incorrect reading of "presenting" data that is inconsistent with the intrinsic evidence and in fact would exclude the preferred embodiment. The functions of the claimed device handler program are setting up and controlling the flow of data (*i.e.*, presenting data and commands) throughout the system. Dr. Polish explained that the device handler program's role was to "in a sense shepherd data to other aspects of the system" and that there "may not necessarily be direct communication" with the device. Polish Tr. 666.

The fact that the device handler program need not have data pass through it to control the flow of data is illustrated in FIG. 2 and its accompanying description in the specification. There, in describing an exemplary facsimile transmission, the '263 patent states that data sent to or from a network is "passed" by the preferred direct memory access components (DMA 50), not the device handler that "requests" (*i.e.*, controls) the transmission. JX-6 at col. 5, lns. 45-48; col. 5, ln. 64 – col. 6, ln. 3; FIG. 2.

Accordingly, the language "presents data" in each proposed construction does not

WI-Apple1604850

require that data "flows through" the device handler or else the construction would improperly read out the preferred embodiment.

Moreover, the proof of Apple's conception and reduction to practice further demonstrates that HTC is misconstruing this limitation and the invention. At the hearing, Inventor John Lynch testified regarding the embodiment of his invention that he had worked on and ultimately reduced to practice no later than [ ] Lynch Tr. 211-263. Mr. Lynch explained that [

] Lynch Tr. 221-22.

Lynch's testimony is consistent with the description of the preferred embodiment in the '263 patent in which the device handler program controls the flow of data but does not directly participate in "handling" or "processing" it. JX-6 at col. 5, lns. 45-48; col. 5, ln. 64 – col. 6, ln. 3.

In short, HTC's argument regarding the "device handler program" is in error because the claim does not cover the preferred embodiment that is disclosed in the '263 patent. *See, e.g.*, *Osram*, 505 F.3d at 1358 (finding claim construction erroneous because it "would exclude the . . . products that the patents were designed to cover").

Accordingly, HTC's argument that its device handlers do not "present data" is rejected.

HTC's contention that its accused devices do not infringe because they are not "specific to" or "associated with" an interface device is likewise rejected for the reasons that follow. In this vein, HTC argues that [ ] do not infringe

because "the accused device handlers do not meet the agreed requirement that the claimed device handler 'know something' about the device it is supposedly handling." HTC Br. at 23.

As an initial matter, the language of the claim does not require this "knowledge." Also, as noted, the claim term "device handler" has been construed to mean "software associated with an interface device that sets up dataflow paths, and also presents data and commands to a realtime signal processing subsystem." Thus, it is clear that under proper construction, the "device handler" does not have this "knowledge" requirement. Further, this "knowledge" requirement is not found elsewhere in the intrinsic evidence, but instead comes from HTC's misreading of certain extrinsic testimony.

HTC specifically contends that it cannot infringe because a "hardware abstraction layer" abstracts the antenna from the accused device handler program, and as a result the program allegedly cannot "know" details about this device. HTC Br. at 24. HTC cites to the testimony provided by Google's Mr. Sparks, who explained that the accused [

] doesn't have the allegedly necessary "knowledge" of the antenna because the accused products [

]

HTC's "knowledge" argument deviates from the intrinsic evidence and reads the preferred embodiment out of the claims. FIG 2's preferred embodiment shows a "hardware abstraction layer 40" (as well as an additional driver layer) between the preferred embodiment's device handler program ("adapter handler 44") and the device ("telecom adapter 36"). JX-6 at FIG. 2, *see* Polish Tr. 800. The hardware abstraction layer 40 is a "hardware dependent" driver that "isolates" the adapter from the "remainder

of the software" including the handler 44. JX-6 at col. 4, lns. 57-65. The "particular characteristics" that the hardware abstraction layer is configured to hide in the preferred embodiment are exactly the "details" that HTC improperly contends the device handler program must "know" about. *Id.* Thus, HTC's argument that a device handler program cannot be separated from a device by intervening modules like "hardware abstraction layers" is contrary to the description of the preferred embodiment.

In contrast to HTC's persistence to tie the device handler inflexibly to a single device, the '263 patent is specifically targeted at creating a system that has the flexibility to handle many different devices, data types, and communication networks. The specification makes clear that this flexibility is enabled "by abstracting the functions of ***each of the elements of the system*** from one another." JX-6 at col. 2, ln. 66 – col. 3, ln. 4 (emphasis added). Thus, despite HTC's assertion that one element – the device handler program – must have detailed knowledge about a particular device, the patent teaches the opposite. Consistent with John Lynch's testimony at the hearing regarding the commercial embodiment, the "association" specified in the Apple/Staff construction means that the handler [ ] Lynch Tr. 297; Polish Tr. 666. It does not need to know "hardware-dependent" characteristics like the resolution of a screen or features of a microphone. Adding onto the handler any further requirement that it needs to "know" or be "specific to" the "particular characteristics" of a device, as HTC suggests, would improperly read out the preferred embodiment. As a result, HTC's argument is rejected.

Based on the analysis set forth above, Apple has satisfied the first element of claim 1.

WI-Apple1604853

The second element of claim 1 recites:

> **a realtime signal processing subsystem for performing a plurality of data transforms comprising a plurality of realtime signal processing operations; and**

Apple has satisfied this claim element.

An example of a realtime signal processing subsystem comprises [

] Polish Tr. 446-47, 450-52, 454-55, 485-86. It performs a plurality of data transforms comprising a plurality of realtime signal processing operations, such as encoding and decoding of compressed audio and video data. CX-595C at 30-31; CX-453; Polish Tr. 448-49, 453-54, 459; Sparks Tr. 1057-60. [

] CX-453 at ln. 78; CX-7097C at ln. 72; CX-6872C at lns. 1131-1151; Chen Tr. 883-86.

Moreover, the [ ] engine must output the media data properly in time. This is accomplished by evaluating [

] More specifically, the [

WI-Apple1604854

]

Accordingly, Mr. Sparks and Dr. Sherwood confirmed that the HTC products perform signal processing operations within the upper bounded time limit defined by the data. Polish Tr. 477-83, 805-09; Sherwood Tr. 826, 830-32; Sparks Tr. 1081-87.

HTC argues that [

] But neither accused subsystems are 'realtime' because they both lack time limits." HTC Br. at 29 (citations omitted).

The claim term "realtime" has been construed to mean "within a defined upper bounded time limit." Documentation describing the accused HTC products confirms that its software expressly meets this requirement. [

]

HTC argues that [ ] the phones instead use [ ] (HTC Br. at 30, 32), but this ignores the fact that [

]

Thus, one can tell if processing is meeting the deadlines defined by the encoded rate if continuous playback is achieved. Polish Tr. 316; Sparks Tr. 1085 [

]

HTC contends that [

] But on cross-examination, Mr. Sparks clarified that [

] This is indeed Laplante's definition of hard realtime. RX-777.30 (Definition 1.5); Polish Tr. 329-30. The fact that [

]

Accordingly, Apple has satisfied the second element of claim 1.

The last element of claim 1 recites:

> **at least one realtime application program interface (API) coupled between the subsystem and the realtime signal processing subsystem to allow the subsystem to interoperate with said realtime services.**

Apple has satisfied this claim element.

As noted, the claim term "realtime API" has been construed to mean "API that allows realtime interaction between two or more subsystems."

Examples of realtime APIs include [

] These allow realtime interaction between two or more subsystems. [

WI-Apple1604856

]

Specifically, Mr. Sparks testified that [

]

Indeed, both Mr. Sparks and D Sherwood confirmed that [

] Polish Tr. 400-04; Sherwood Tr. 819-20, 857-59; Sparks Tr. 1067-68, 1091-93. In the Android architecture, [

] Sparks Tr. 1057-59.

[

] Sherwood Tr. 822 [

]

The Android architecture uses [

WI-Apple1604857

]

Sparks Tr. 1092 [

]

HTC contends that because "Apple failed to identify any time limits in the system," Apple cannot establish that the "realtime signal processing subsystem" limitation is satisfied. HTC Br. at 45. This argument is the same as the one for the first element of claim 1, *supra*. Thus, this argument is rejected for the same reasons that the first element of claim 1 was found to be infringed.

HTC further asserts that "Apple's allegations regarding the realtime API are also defective. In both OpenCORE and StageFright, Apple accuses 'header' files as the claimed realtime API." HTC Br. at 26. HTC explains that "Apple's exclusive reliance on header files results in a fatal flaw. Header files are not on the accused phones, and anything from them that is compiled onto the phones exists only within the accused signal processing system, preventing the accused API from being *coupled between or distinct from the subsystems*." *Id.* (emphasis added).

WI-Apple1604858

HTC's argument that accused APIs are not "coupled between" or "distinct from the subsystems" has already been rejected in connection with claim construction for "realtime API," *supra*.

Accordingly, Apple has satisfied the last element of claim 1.

2. Claim 2

Dependent claim 2 recites:

> **The signal processing system as set forth in claim 1, wherein said signal processing system receives and transmits a plurality of datatypes over a plurality of different wide area networks (WANs).**

Apple has satisfied claim 2.

HTC products receive and transmit datatypes such as MP3 audio, telephony audio and MPEG video. Stipulation Relating to Use of Respondents HTC Corp., HTC America, Inc., and Exedea, Inc.'s Products in the United States (Apr. 17, 2011), ¶ 5; Polish Tr. 486-87. The datatypes are received and transmitted over WANs such as various cellular networks and the internet via IEEE 802.11 WiFi. Polish Tr. 487.

3. Claim 24

The preamble and the first element of claim 24 recite:

> **The signal processing system of claim 1, wherein said realtime signal processing subsystem comprises:**
>
> **a realtime processor including an operating system for executing a plurality of realtime functions;**

Apple has satisfied the preamble and the first element.

Each HTC product includes a realtime processor, [

]

The second element of claim 24 recites:

> **a realtime communications module which is independent of said realtime processor and is coupled to receive a plurality of communications commands from said application programs via said device handler program and said realtime API, said realtime communications module operating in response to said communications commands to issue a plurality of requests for realtime services to said realtime processor; and**

Apple has satisfied this element. The accused products include a realtime communications module which is independent of the realtime processor and is coupled to receive a plurality of communications commands from the application programs via the device handler program and realtime API, the realtime communications module operating in response to said communications commands to issue a plurality of requests for realtime services to said realtime processor.

Each HTC product includes realtime communications modules such as the

[

] Indeed, [

WI-Apple1604860

]

These realtime communications modules are also [

]

The last element of claim 24 recites:

> **a translation interface program which is specific to said realtime processor and is coupled to receive said requests for realtime services from said communications module and provide said requests to said realtime processor.**

Apple has satisfied this element.

For HTC products, an example of a translation interface program is [

]

WI-Apple1604861

4. Claim 29

Dependent claim 29 recites:

> **The signal processing system of claim 24, wherein said realtime processor is embodied in a hardware device and includes realtime function libraries that are embodied in programmable software.**

Apple has satisfied claim 29.

Each HTC product includes a realtime processor, [

] CX-199C at 15, 21, 29; CX-202C at 17-18, 46-48, 100; CX-595C at 11, 30-31; Polish Tr. 465-66, 469-70, 487-88, 493. [

]

In summary, Apple has shown that HTC's accused products infringe all asserted claims of the '263 patent.

C. Technical Prong of the Domestic Industry Requirement

For the reasons set forth below, Apple has satisfied the technical prong of the domestic industry requirement with respect to the '263 patent.

1. Claim 1

The preamble of claim 1 recites:

> **A signal processing system for providing a plurality of realtime services to and from a number of independent client applications and devices, said system comprising:**

Apple has satisfied the preamble.

WI-Apple1604862

Apple's domestic industry product ("DI Product")[23] includes applications such as QuickTime X. CPX-10; Polish Tr. 616. It also includes devices such as the hardware network interface, speakers and a display. Polish Tr. 617. The DI Product also provides, to and from the client applications and devices, realtime services such as progressive download, audio and video playback, and audio and video streaming from a network. Polish Tr. 617, 619-20; CPX-10.

The first element of claim 1 recites:

> **a subsystem comprising a host central processing unit (CPU) operating in accordance with at least one application program and a device handler program, said subsystem further comprising an adapter subsystem interoperating with said host CPU and said device;**

Apple has satisfied this claim element.

The DI Product includes a CPU. Polish Tr. 616; CPX-10. The CPU operates in accordance with application programs such as QuickTime X. Polish Tr. 616. The CPU also operates in accordance with device handler programs [ ] CX-670 at 9; CX-5714C; CX-5717C; Polish Tr. 620-24. The DI Product includes an adapter subsystem, [ ] that interoperates with the host CPU and devices. '263 Issues Stip., ¶ 5; Polish Tr. 616-17.

HTC argues that Apple's domestic industry product, the MacBook Pro, does not practice the '263 patent because it does not include the claimed "device handler" program. HTC Br. at 45. Specifically, HTC contends that [

[23] Apple's domestic industry products include the MacBook Pro running Mac OS X v10.6 Snow Leopard (CPX-10) and the iPhone 3GS (CPX-11).

WI-Apple1604863

] (which Dr. Polish identifies as the device handler), is neither "specific to" nor "associated with" the claimed device, which Apple claims is the network interface because[

] *Id.*

HTC's contention is rejected for the same that its infringement argument was rejected above. As noted, the device handler program need not have detailed knowledge about a particular device. [Furthermore, HTC erroneously limits its domestic industry analysis to a discussion about the hardware network interface. Id. Rather than citing to any evidence, HTC claims that [ ] does not satisfy the device handler limitation based solely on conclusory statements by Dr. Brandt that [

] *Id.* (citing Brandt Tr. 1437-38).] Again, this argument incorrectly requires that the device handler program needs to know details about the device and specifically limits the device to the hardware network interface.

[

] CX-670 at 9. Dr. Polish explained how [

] Polish Tr. 620-21; CX-670 at 11, 18.] Satisfaction of this limitation is clear given the proper claim construction. Apple has satisfied the first element of claim 1 for the technical prong.

The second element of claim 1 recites:

> **a realtime signal processing subsystem for performing a plurality of data transforms comprising a plurality of realtime signal processing operations; and**

Apple has satisfied this claim element.

WI-Apple1604864

An example of a realtime signal processing subsystem comprises [

] CX-7453C; CX-7454C; Polish Tr. 627-28; *see* CX-5718C; CX-5719C; CX-5726C; CX-5727C; CX-5720C – CX-5725C. The realtime signal processing subsystem performs a plurality of data transforms comprising a plurality of realtime signal processing operations [

] CX-670 at 14; Polish Tr. 648-52.

HTC argues that Apple's domestic industry product, the MacBook Pro, does not practice the '263 patent because it does not perform realtime processing. HTC Br. at 44. Specifically, HTC asserts that all of the proposed constructions of the term "realtime" require the claimed "realtime" components, including the "realtime signal processing subsystem" to operate according to time limits and that the MacBook Pro does not operate under any such time limits, and as such, does not practice the '263 patent. *Id.* HTC also contends that "Dr. Polish did not even attempt to establish that the MacBook Pro uses time constraints [ ] *Id.*

HTC's arguments ignore Dr. Polish's explanation of how Apple's DI Product [ ] Polish Tr. 650. In particular, [

] Polish Tr. 650-51. Further, [

] CX-670 at 14; Polish Tr. 648-49. Thus, [

] Polish Tr. 650.

WI-Apple1604865

Accordingly, Apple has satisfied the second element of claim 1 for the technical prong.

The last element of claim 1 recites:

> **at least one realtime application program interface (API) coupled between the subsystem and the realtime signal processing subsystem to allow the subsystem to interoperate with said realtime services.**

Apple has satisfied this claim element.

The DI Product includes a realtime API [ ] Polish Tr. 625. [ ] allows the subsystem to interoperate with realtime services [ ] CX-7453; Polish Tr. 625-28. Apart from the argument that Apple's DI Product does not contain realtime components, both respondents and the Staff do not dispute that [ ] qualifies as the claimed realtime API. Inasmuch as the proper claim construction of "realtime API" does not require that the API itself be realtime, it is undisputed that Apple's DI Product includes a realtime API [ ]

HTC contends that because "Apple failed to identify any time limits in the system," Apple cannot establish that the "realtime API" limitation is satisfied. HTC Br. at 45.

This argument is the same as the one for the first element of claim, *supra*. Thus, this argument is rejected for the same reasons that the first element of claim 1 was found to be infringed.

WI-Apple1604866

2. Claim 24

The preamble and the first element of claim 24 recite:

> **The signal processing system of claim 1, wherein said realtime signal processing subsystem comprises:**
>
> **a realtime processor including an operating system for executing a plurality of realtime functions;**

Apple has satisfied the preamble and the first element.

Apple's DI Product has a realtime processor such as an Intel core. Polish Tr. 656-57. The realtime processor includes an operating system such as Mac OS X, which is for executing a plurality of realtime functions. *Id.*

The second element of claim 24 recites:

> **a realtime communications module which is independent of said realtime processor and is coupled to receive a plurality of communications commands from said application programs via said device handler program and said realtime API, said realtime communications module operating in response to said communications commands to issue a plurality of requests for realtime services to said realtime processor; and**

Apple has satisfied this element.

Apple's DI products include realtime communications modules, [

] CX-7454; Polish Tr. 654-55. [ ] are independent of the realtime processor and coupled to receive a plurality of communication commands from applications [

] CX-7454; Polish Tr. 647-48, 654-55. [ ] issue requests for realtime services, [ ] to the

WI-Apple1604867

realtime processor. CX-670 at 17-18; CX-7454; Polish Tr. 627, 650-51, 654-55

The last element of claim 24 recites:

> **a translation interface program which is specific to said realtime processor and is coupled to receive said requests for realtime services from said communications module and provide said requests to said realtime processor.**

Apple has satisfied this element.

Apple's DI product includes a translation interface program [

] CX-7454; Polish Tr. 655-56. [

] CX-5726C; CX-5727C; Polish Tr. 657-58. [

] receives requests for realtime services [ ] and provides them to the realtime processor. CX-7454.

In summary, Apple has satisfied the technical prong of the domestic industry requirement with respect to the '263 patent.

D. Validity

HTC asserts that the AT&T VCOS system anticipates and renders obvious all asserted claims of the '263 patent. HTC Br. at 33-39. HTC further asserts that U.S. Patent No. 5,790,781 ("Cox") anticipates all asserted claims of the '263 patent. HTC Br. at 39-44.

The Staff submits that there is clear and convincing evidence that the asserted claims of the '263 patent are invalid as anticipated and obvious by AT&T's VCOS system and that claim 24 is invalid as indefinite because one skilled in the art would not have had "any idea as to the scope of the claim due to the irresolvably ambiguous term

WI-Apple1604868

'realtime communication module'." Staff Br. at 37, 40-41. The Staff also argues that "the other prior art asserted by respondents may also invalidate the '263 patent claims (such as U.S. Patent No. 5,790,781 to Cox), but that the VCOS system is the closest prior art." *Id.* at 37 n.15.

Apple argues that the VCOS system and the Cox patent do not invalidate the '263 patent because they are "fundamentally different from the '263 patent, and in the same way as the art that was cited and overcome during prosecution." Apple Br. at 48-49.

1. AT&T VCOS System

As noted, HTC and the Staff assert that the AT&T VCOS system anticipates and renders obvious all asserted claims of the '263 patent. HTC Br. at 33-39; Staff Br. at 37.

For the reasons set forth below, HTC and the Staff have not shown by clear and convincing evidence that the AT&T VCOS system anticipates and renders obvious all asserted claims of the '263 patent.

The VCOS system[24] does not anticipate any of the asserted claims because it fails to disclose at least the realtime API and device handler limitations. Polish Tr. 1671-76. The VCOS system was expressly discussed as similar but inferior prior art in column 1 of the Chen patent. RX-1103 at col. 1, ln. 58 – col. 2, ln. 2; JX-11 at APPHTC_00013767-76 (USPTO BPAI, Decision on Appeal (May 31, 2001)); Polish Tr. 1675-76. Thus, this art fails for at least the same reasons that the claims were found patentable over Chen during prosecution.

In order to determine whether the VCOS system anticipates the '263 patent, it is

[24] HTC and the Staff also failed to present clear and convincing evidence of any act sufficient to establish that a particular "VCOS system" was prior art to the '263 patent claims.

WI-Apple1604869

important to understand the prosecution history of the '263 patent as it relates to the Chen patent. During the prosecution of the '263 patent, the examiner focused on the Chen patent as the key prior art. JX-11 at APPHTC_00013307; *see* RX-1103. In relevant part, Chen discloses a DSP manager for loading tasks onto specific DSP hardware, and for managing DSP hardware resources (*e.g.* memory). RX-1103 at col. 2; Polish Tr. 1667-69.

Apple's response explained that Chen did not disclose an application programming interface "coupled between the subsystem [which includes the device handler program] and the realtime signal processing subsystem." JX-11 at APPHTC_00013529 (alteration in original). Chen did not disclose the invention's "architectural arrangement provid[ing] a layer of abstraction which eliminates the requirement that the handler have any knowledge of the particular implementation of the realtime engine." *Id.* at APPHTC_00013529-30.

Despite Apple's explanation, the examiner maintained his obviousness rejection and responded that Chen had "various APIs coupled between device drivers … and a DSP manager." *Id.* at APPHTC_00013544. Apple appealed, again explaining that "[t]he function of [Chen's DSP] manager is essentially to call and load various multi-media software tasks into the hardware for the digital signal processors" and that "does not constitute … an application programming interface provid[ing] a layer of abstraction between the particular implementation of a realtime engine and the [host CPU] subsystem." *Id.* at APPHTC_00013715. In other words, "[t]here is no disclosure in the Chen et al patent which supports a conclusion that the DSP manager 71 itself constitutes, or otherwise contains, an [API], nor that it generates API commands." *Id.* at

APPHTC_00013716.

The BPAI agreed with Apple and allowed the claims. *Id.* at APPHTC_00013772. The BPAI ruled that the "Examiner has failed to set forth a *prima facie* case," finding that Chen's DSP Manager "is not an API, does not include an API, and does not generate API commands." *Id.* at APPHTC_00013772-74. In other words, Chen's DSP Manager is not "an API interfaced between the CPU subsystem and the real-time processing subsystem." *Id.* at APPHTC_00013774.

"realtime API"

Functionally, VCOS is another "bottom-up" system, designed by AT&T to ease integration of AT&T's particular DSP3210 chips into a host system. Lynch Tr. 1186-87; Polish Tr. 1671-73; RX-960.006. More specifically, VCAS allows applications to load and run VCOS DSP tasks on a VCOS DSP. Lynch Tr. 1161-62; Polish Tr. 1673. It is undisputed that VCOS DSP tasks are "written for the VCOS system" and "would not run correctly" in a non-VCOS system, as confirmed by respondents' own paid consultant. Lynch Tr. 1164, 1187-88;[25] *see* RX-1038 at 17 ("The hardware that runs VCOS must include at least one AT&T DSP3210."). Applications using VCAS thus need to know that the DSP is a VCOS DSP in order to load VCOS DSP tasks. This need for knowledge about the particular implementation of the real-time engine is incompatible with the '263 invention and its realtime API, which "provid[es] a layer of abstraction

[25] Although HTC's paid consultant John Lynch, did mention that AT&T had "looked at" the possibility of building a different system that might be used with non-AT&T DSPs (Lynch Tr. 1164-65), this conclusory testimony falls far short of providing clear and convincing evidence of anything. He confirmed that they "didn't do it" and never explained any details about this hypothetical system or when the discussions even took place.

between the real-time engine and the remainder of the processing system, … eliminat[ing] the need for a device handler to have any knowledge about the particular implementation of the real-time engine." JX-11 at APPHTC_00013684; JX-6 at Abstract, col. 2, ln. 66 – col. 3, ln. 11; Polish Tr. 1671-73, 1675-76.

Similarly, if a non-VCOS DSP is used, the application would have to be redesigned to use different DSP tasks compatible with that non-VCOS DSP. This need to redesign the system is also inconsistent with the invention's realtime API, in which "any one of a hardware-implemented, software-implemented or native digital signal processor can be employed, without requiring any redesign of the system." JX-11 at APPHTC_00013684; Polish Tr. 1671-73, 1675-76.

As indicated above, the '263 patent is a fundamentally different dual subsystem architecture, with a realtime API that abstracts a separate realtime processing subsystem and so allows the host CPU subsystem to be reused—without redesign—with any different kind of realtime subsystem, even a "native" DSP engine running on the CPU. Lynch Tr. 216-18; Polish Tr. 310, 332-33, 352-53. Because VCOS lacks the requisite "layer of abstraction between the real-time engine and the remainder of the processing subsystem," the real-time engine is tightly integrated into the host CPU subsystem.

The VCOS documentation is clear in distinguishing its approach from the '263 invention: "***[r]ather than having a separate DSP subsystem***, the VCOS Operating System ***integrates*** the DSP32xx into [host] computer environments." RX-963 at 1; Polish Tr. 1672-73 (emphasis added). This tight integration is visually depicted in FIG. 1 (RX-963 at 1), which shows applications directly linked to DSP tasks without an API coupled between them. *See* RX-963 at 2 ("In the VCOS environment, applications

communicate directly with DSP tasks via standard buffers in system memory.") VCOS, like Chen, thus lacks a realtime API "coupled between" or "distinct from" the realtime subsystem and the host CPU subsystem.

"device handler program"

VCOS further fails to meet claim 1 because it does not disclose a device handler program. Polish Tr. 1676. VCOS device drivers and the VCOS Resource Manager (VRM) do not constitute the claimed device handler program as neither is involved in setting up data flow paths.[26] First, the VRM is not abstracted by VCAS (the alleged realtime API) because VCAS and VRM reside in the same software layer. RX-1038 at 14.

HTC attempts to correlate the function of the VRM to the device handler program by noting that it is involved in mixing streams of data into a single output stream. HTC Br. at 36. But this is not consistent with the patent's disclosure, as the '263 patent makes clear that "mixing audio streams into a single output stream" is a function in the preferred embodiment of serial driver 42, not the device handler program. JX-6 at col. 4, ln. 66 – col. 5, ln. 8.

Moreover, the only alleged support in the record that HTC points to for "VCOS device drivers" being the claimed device handler program is Dr. Brandt's testimony referencing a figure in non-prior-art RX-1037. Brandt Tr. 1389-90. But these "drivers" are akin to the "serial drivers" in the preferred embodiment, not the device handler

[26] Dr. Brandt relied on RX-1037 (published sometime in 1994) at the hearing as proof of the "device drivers" and the VRM. Dr. Brant, however, has not testified regarding these limitations based on any document that *actually* pre-dates the invention of the '263 patent (August 2, 1994).

WI-Apple1604873

program, and there is no evidence that they set up data flow paths as opposed to simply being a part of those paths.

VCOS also fails to meet claim 24 because it lacks at least a "realtime communications module." HTC's expert (Dr. Brandt) identified "VCOS communication code" without further explanation. Brandt Tr. 1396. Thus, HTC has insufficient proof with respect to at least the "realtime communications module" limitation of claim 24. HTC seeks in its brief to "fix" Dr. Brandt's conclusory testimony about "communications code" satisfying this limitation by redirecting its arguments to "FIFO and parameter functions" that were never referenced at the hearing.[27] HTC Br. at 38. Respondents' attorney argument is unsupported by record evidence.

Accordingly, HTC and the Staff have not shown by clear and convincing evidence that the AT&T VCOS system anticipates and renders obvious all asserted claims of the '263 patent.

2. U.S. Patent No. 5,790,781 ("Cox")

As noted, HTC asserts that U.S. Patent No. 5,790,781 ("Cox") (RX-1117) anticipates all asserted claims of the '263 patent. HTC Br. at 39-44. The Staff merely contends, without any relevant analysis, that the Cox patent "may also invalidate the '263 patent claims." Staff Br. at 37 n.15.

Apple argues that the Cox patent is not prior art against at least claim 1 because

[27] HTC relies on RX-1038 as showing that the FIFO and parameter functions are a "lower level" in VCAS. HTC Br. at 38-39. There is no evidence of multiple layers in VCAS. *E.g.*, RX-1038 at 13 ("Generic VCOS consists of ***only two host-side layers***. One of these is the HAL; the other is . . . VCAS.") (emphasis added). Even if the VCAS module were split into layers, respondents never once limit their identification of the VCAS API to anything less than the entire VCAS module.

WI-Apple1604874

its priority date is December 8, 1993 (RX-1117.02), well after the '263 patent's May 1992 conception date and August 1993 actual reduction to practice. Apple Br. at 53. Apple further submits that the Cox patent does not invalidate the '263 patent because it is "fundamentally different from the '263 patent, and in the same way as the art that was cited and overcome during prosecution." *Id.* at 48-59.

For the reasons stated below, respondents and the Staff have not shown by clear and convincing evidence that the Cox patent anticipates all asserted claims of the '263 patent.

Conception and Reduction to Practice

HTC did not challenge or rebut evidence proving the '263 invention was conceived by May 29, 1992. This date of conception is corroborated by [ ] CX-1490C; CX-1515C; CX-1518C; Polish Tr. 1687-90. [ ] describes a computer system architecture to support realtime services to and from a number of applications [ ] CX-1490C at 21, 29, 85, 89. The disclosed computer system includes a subsystem with a CPU, a [ ]device handler program, and adapter subsystem. CX-1490C at 85, 105. The disclosed computer system includes a realtime signal processing subsystem [ ] and realtime application programming interface (RALF API). CX-1490C at 89, 98. The realtime API is coupled between the subsystem and realtime signal processing subsystem. CX-1490C at 98, 105; CX-1515C; CX-1518C. [ ] that includes all of these elements of the '263 invention. CX-1490C at 105; Lynch Tr. 252-55. [ ] shows both the claimed elements and relative position of those

WI-Apple1604875

elements. CX-1490C at 105. [

] conception of the '263 invention was complete. Lynch Tr. 255 ("Q. Now, as of this date, May 29th, 1992, was [ ] A. Yes, it was.").

The '263 patent was reduced to practice by [

] the inventors worked to reduce their invention to practice. CX-1529C; CX-1537C; CX-1540C; Lynch Tr. 256. [

]

CX-1554C; Lynch Tr. 262-63; Polish Tr. 1690-91. HTC failed to challenge or rebut this evidence.

HTC counters that Apple cannot show that the "device handler" element of claim 1 was conceived and reduced to practice prior to the filing date of the '263 patent because Dr. Polish failed to provide any analysis for how the [ ] "presents data" to a realtime subsystem. HTC Br. at 43.

HTC's argument is without merit. The undersigned rejected HTC's earlier infringement argument that the agreed portion of the construction that "presents data" requires the device handler program to itself process and handle data, because it would exclude the preferred embodiment of the invention. Thus, [

] as HTC claims.

Accordingly, the Cox patent is not prior art against at least claim 1 because the record evidence shows that the invention was conceived in May 1992 and reduced to practice in August 1993.

WI-Apple1604876

"realtime API"

The Cox patent does not anticipate any of the asserted claims because it fails to disclose the "realtime API" element. Polish Tr. 1677-78. Cox was Intel's effort at an integrated DSP system and does not disclose a realtime API coupled between or distinct from a separate realtime signal processing subsystem—the applications need to have detailed knowledge of the DSP implementation in order to know what DSP task to run. Polish Tr. 1677-78; Brandt Tr. 1399; RX-1117 at col. 5, lns. 10-27.

Specifically, the alleged realtime API, DSP Interface 120, is not coupled between a first subsystem and a realtime signal processing subsystem. RX-1117 at FIG. 4. FIG. 4 depicts only two subsystems—a host subsystem and a multi-function I/O subsystem. RX-1117 at FIG. 4. DSP Interface 120 is located ***in the middle of the host subsystem***, and is thus a part of it, as opposed to coupled between to abstract the two subsystems. *Id.* In that regard, HTC confuses the record by stating that "Apple contends that the DSP Interface is not a 'realtime API' because it 'needs knowledge of what is running on that DSP.'" HTC Br. at 42. To be clear, Dr. Polish testified that the Cox patent is different from the '263 invention because in Cox "the ***application*** has to know what DSP task to run . . . ***it*** needs knowledge of what is running on that DSP." Polish Tr. 1678 (emphasis added).

HTC's argument that "the DSP interface has no knowledge" thus misses the point—it is the application (or alleged device handler program) in Cox that has knowledge of DSP implementation details, demonstrating that the DSP Interface 120 does not provide the necessary layer of abstraction between the application and DSP. Polish Tr. 1678. In fact, each device driver in the Cox patent needs to know the details of

WI-Apple1604877

an associated DSP-OS task. RX-1117 at col. 5, lns. 21-28 ("Each host device driver defines an appropriate set of message formats . . . for communications with the corresponding DSP-OS task."). The device driver controls the "functions of a DSP-OS task . . . by placing messages in a corresponding mailbox." RX-1117 at col. 6, lns. 20-26.

Moreover, like the DSP Manager in Chen, Cox's DSP interface 120 does not provide a layer of abstraction, but rather routes commands (messages) from the device driver to the DSP-OS task. RX-1117 at col. 6, lns. 33-42. "The DSP interface 120 enables command message transfer between the host device drivers 110-114 and the DSP-OS tasks." RX-1117 at col. 6, lns. 62-67. The communication is sufficiently direct that FIG. 5 does not even depict the DSP Interface. Indeed, the link between alleged device handler program and DSP is sufficiently tight that "[a] host device driver and a corresponding DSP-OS task have a client/server relationship." RX-1117 at col. 7, lns. 16-17; *see also id.* col. 7, lns. 29-30 ("The DSP-OS tasks send data to the host device drivers, and receive data from the host device drivers.").

Respondents' characterization of Cox as the "same mechanism used by the realtime API in the '263 Patent" (HTC Br. at 42) is unsupported and wrong. The '263 patent discloses command/control mailboxes, but the device handler program in the '263 patent places into those mailboxes abstract commands for "particular transforms to be performed"—not commands linked to a specific DSP task on a specific DSP. JX-6 at col. 5, lns. 21-25. If the device handler program of the '263 patent needed to know the "appropriate set of message formats" (RX-1117 at col. 5, lns. 29-37) for a directly associated DSP task, then the realtime API would not and in fact could not provide the requisite layer of abstraction that is required by the '263 patent invention.

WI-Apple1604878

"device handler program"

The Cox patent also does not anticipate any of the asserted claims because it fails to disclose the "device handler program" element. Polish Tr. 1678. Respondents' identification of the host device drivers in Cox as the device handler program is wrong. Cox fails to disclose that the host device drivers set up data flow paths. Polish Tr. 1678. In fact, host device drivers are not mentioned anywhere in the context of the data flow path: "DSP-OS tasks . . . read input data from a source device, process[] the input data, and then write the processed data to a sink device." RX-1117 at col. 7, lns. 49-51. Dr. Brandt's conclusory testimony on this matter (Brandt Tr. 1403-04) finds no support in the record, and falls far short of respondents' clear and convincing burden of proof.

Dependent Claims

Cox also fails to teach the limitations of the asserted dependent claims. Claim 2 requires "a plurality of different wide area networks (WANs)." HTC concedes that all of the disclosed data types are sent "via the telephone" and thus Cox does not disclose at least two WANs.

HTC's proof for the limitations of claim 24 also falls short. Dr. Brandt's conclusory analysis places the "realtime communications module" and the "translation interface" in two different subsystems, despite the fact that in the patent they are both part of the realtime processing subsystem. Brandt Tr. 1410-11. Moreover, HTC argues without support that "DSP device drivers" in Cox correspond with the translation interface program. However, as explicitly stated in Cox, the "DSP device drivers" are programmed directly into the DSP, *i.e.*, are part of the DSP. RX-1117 at col. 7, lns. 41-48, FIG. 4.

WI-Apple1604879

The Cox patent does not anticipate claims 24 or 29 at least because the alleged realtime communications module is not independent of the DSP (Brandt Tr. 1410; RX-1117 at col. 5, lns. 29-38); and because the alleged translation interface program resides entirely *on* the DSP (Brandt Tr. 1410). Thus, HTC and the Staff have failed to put forth clear and convincing proof that the Cox patent anticipates the asserted dependent claims of the '263 patent.

In sum, HTC and the Staff have not shown by clear and convincing evidence that the Cox patent anticipates all asserted claims of the '263 patent.

3. Claim 24 – Indefiniteness

As noted, the Staff asserts that dependent "claim 24 is indefinite because one skilled in the art would have not had any idea as to the scope of the claim due to the irresolvably ambiguous term 'realtime communication module'." Staff Br. at 40-41.

HTC no longer asserts that claim 24 is indefinite.

Apple argues that for claim 24, the claim language, the specification, the prosecution history, and the testimony of Dr. Polish, confirm that the "realtime communications module" is definite and should be afforded the construction offered by Apple. Apple Br. at 23-25.

As discussed below, the Staff has not shown by clear and convincing evidence that dependent claim 24 is indefinite because the term "realtime communications module" cannot be understood by a person of ordinary skill in the art. The claim term "realtime communications module" is not insolubly ambiguous—the high standard required to be met before classifying a claim issued by the USPTO as indefinite. *Exxon Research and Engineering Co. v. United States*, 265 F.3d 1371, 1375 (Fed. Cir. 2001)

("If a claim is insolubly ambiguous, and no narrowing construction can properly be adopted, we have held the claim indefinite. If the meaning of the claim is discernible, even though the task may be formidable and the conclusion may be one over which reasonable persons will disagree, we have held the claim sufficiently clear to avoid invalidity on indefiniteness grounds.").

Its context in the claim language alone tells a person of ordinary skill in the art what is claimed by a "realtime communications module." Polish Tr. 356-59; *Phillips v. AWH Corp.*, 415 F.3d 1303, 1314 (Fed. Cir. 2005). The term uses common terminology, and the word "module" even appears in respondents' proposed constructions. Claim 24 describes that the realtime communication module: (1) is independent of the realtime processor; (2) is coupled to receive a plurality of commands from application programs via the device handler program and the realtime API; and (3) operates in responses to commands to issue a plurality of requests for realtime services to said realtime processor. Thus, a person of ordinary skill would be familiar with the constituent words in the limitation, would know where the software resides in the computer architecture, and would know what functions the software performs.[28]

The specification also confirms to a person of ordinary skill the proper meaning of the term "realtime communication module." In the preferred embodiment, the "generic service provider 62" performs exactly the same functions as recited in dependent claim 24 for the realtime communications module. JX-6 at col. 7, lns. 13-37, FIG. 3.

[28] Moreover, while not limiting the term, dependent claims 25 and 26 would provide additional guidance to a person of ordinary skill, explaining further what the realtime communications module does and further dispelling any alleged ambiguity. *Phillips*, 415 F.3d at 1314 ("Other claims of the patent ... can also be valuable sources of enlightenment as to the meaning of a claim term.").

Even though the exact claimed term may not be present in the specification, the fact that an embodiment of the realtime communication module is consistently described in the specification would confirm to a person of ordinary skill that the term is not insolubly ambiguous. *Bancorp Servs. LLC v. Hartford Life Ins. Co.*, 359 F.3d 1367, 1371-73 (Fed. Cir. 2004) (reversing district court finding of indefiniteness even though claim term was "not defined [or] used, anywhere in the specification" because its construction was discernible from context of a different term consistently described in the specification).

Moreover, the prosecution history shows that the examiner, presumptively a person of ordinary skill, had no trouble ascertaining what the "realtime communications module" was and in applying prior art against the claims that contained this limitation. *See* JX-11 at APPHTC_00013308 (Aug. 18, 1995 Office Action, at 5).

HTC offered essentially no expert testimony to support its construction. HTC's Dr. Brandt only made a passing reference to the term being indefinite when applying claim 24 against the prior art. Brandt Tr. at 1396. Such testimony falls well short of meeting the burden to prove the limitation is insolubly ambiguous by clear and convincing evidence. This contrasts with Dr. Polish's detailed explanation as to how a person of ordinary skill in the art would understand the limitation, and how the specification's description "match[es] up exactly" with the context in which the limitation appears in claim 24. Polish Tr. 335-337, 356-359.

The Staff argues that the claim term "realtime communications module" in claim 24 is indefinite because although "claim 24 specifies what the realtime communication *does*, the patent provides no description of what it *is*." Staff Br. at 41. But the claim *does* in part describe what the limitation is ("independent of said realtime processor"); the term

itself describes what it is (a "realtime communications module"); and the claims describe its architectural position in detail (*e.g.* "coupled to receive. . ."). A person of ordinary skill would know not only what the realtime communication module does, but also where it resides in the architecture and how it fits with other components.

Moreover, even if the limitation *were* purely functional, its meaning is discernible in context and thus it would not be indefinite. *Enzo Biochem, Inc. v. Applera Corp.*, 599 F.3d 1325, 1332-34 (Fed. Cir. 2010).

Accordingly, the Staff has not shown by clear and convincing evidence that dependent claim 24 is indefinite.

VIII. U.S. Patent No. 6,275,983

The '983 patent is entitled, "Object-Oriented Operating System." The '983 patent discloses the use of object-oriented methods requiring native system services that enable object-oriented applications to access a procedural operating system in an object-oriented manner. JX-4 at col. 5, lns. 11-15. The invention is described, in part, as follows:

> The present invention is directed to a system and method of enabling an object-oriented application to access in an object-oriented manner a procedural operating system having a native procedural interface. The system includes a computer and a memory component in the computer. A code library is stored in the memory component. The code library includes computer program logic implementing an object-oriented class library. The object-oriented class library comprises related object-oriented classes for enabling the application to access in an object-oriented manner services provided by the operating system. The object-oriented classes include methods for accessing the operating system services using procedural function calls compatible with the native procedural interface of the operating system. The system also includes means for processing object-oriented statements contained in the

> application and defined by the class library by executing methods from the class library corresponding to the object-oriented statements.

JX-4 (Summary of the Invention), at col. 3, lns. 45-62.[29]

Apple asserts independent apparatus claim 1 and independent method claim 7.

The asserted claims read as follow:

> **1.** A computer system, comprising:
>
> computer hardware for performing native system services;
>
> a procedural operating system, having a native interface, for controlling the computer hardware to perform the native system services;
>
> object oriented methods requiring native system services;
>
> procedural program logic code, responsive to invocations

---

[29] Under the heading "Computing Environment," the Detailed Description Of The Preferred Embodiments in part states:

> The present invention is directed to a system and method for providing an object-oriented interface to a procedural operating system having a native procedural interface. The present invention emulates an object-oriented software environment on a computer platform having a procedural operating system. More particularly, the present invention is directed to a system and method of enabling an object-oriented application to access in an object-oriented manner a procedural operating system having a native procedural interface during run-time execution of the application in a computer. The present invention is preferably a part of the run-time environment of the computer in which the application executes. In this patent application, the present invention is sometimes called an object-oriented wrapper since it operates to wrap a procedural operating system with an object-oriented software layer such that an object-oriented application can access the operating system in an object-oriented manner.

JX-4 at col. 4, ln. 66 – col. 5, ln. 15 (emphasis added).

WI-Apple1604884

of the object-oriented methods during runtime, for causing the procedural operating system to control the computer hardware to perform the required native system services;

executable program memory associated with the computer hardware for runtime execution of the procedural operating system, invocations of the object-oriented methods and related portions of the procedural program logic code;

means for making determinations during runtime execution if object-oriented methods to be invoked are present in the executable program memory; and

a runtime loader, responsive to the determinations, to selectively load required object-oriented methods into the executable program memory during runtime before invocation of the object-oriented methods.

7. A method for operating a computer system, comprising the steps of:

executing a procedural operating system on computer hardware, the procedural operating system including a native interface, responsive to procedural function calls, for providing native system services;

issuing calls during runtime, compatible with the native interface, to provide the native system services in response to invocations of object-oriented methods requiring such native system services;

determining during runtime if object-oriented methods to be invoked during runtime execution are present in executable program memory associated with the computer hardware; and

selectively loading the object-oriented methods into the executable program memory during runtime before invocation thereof, if not yet loaded.

JX-4 at col. 37, ln. 50 – col. 38, ln. 7; col. 38, lns. 40-56.

[redacted]

A. Claim Construction[30]

The parties have agreed to constructions for the terms "native systems services," "procedural logic code," and "during runtime" (claims 1 and 7). Joint Claim Construction, App'x A at 1 & April 4, 2011, Supplement. The disputed terms are discussed below.

1. "to selectively load required object-oriented methods into the executable program memory"

Claim Term	Apple Construction	HTC Construction	Staff Construction
"to selectively load required object-oriented methods into the executable program memory during runtime before invocation of the object oriented methods" (claim 1) "selectively loading the object-oriented methods into the executable program memory during runtime before invocation thereof" (claim 7)	loading required object-oriented methods into the executable memory during runtime before invocation of the object-oriented methods as needed	selecting the required object oriented method code for the system during runtime and copying the selected code into the executable program memory just before invocation thereof	copying (or transferring) required objected-oriented methods into the executable memory during runtime before invocation of the [object-]oriented methods as needed

Joint Claim Construction, App'x A at 2.

The parties dispute the meaning of "loading … into the executable program memory" and "selectively load required object-oriented methods." Apple argues that

[30] A person of ordinary skill in the art would have a B.S. degree in computer science or equivalent, and two to three years of industry experience. Jeffay Tr. 3306; Spielman Tr. 4681.

WI-Apple1604886

loading is not limited to physical copying, but includes virtual copying as well.[31] Apple further argues that with respect to "selectively loading," "[t]here is nothing in the '983 patent or its file history that imposes any limitation of 'selectively loading' in claims 1 and 7 to extracting only a single method from its class, as opposed to loading the class containing the desired method (but not loading other classes). JX-4; JX-10." Apple Br. at 59-63.

HTC argues that loading requires copying and that Apple disclaimed virtual copying during patent prosecution. HTC Br. at 64-69. HTC further contends that selective loading of methods is different from selective loading of classes. *Id.* at 69-72. The Staff agrees with HTC. Staff Br. at 46-51.

The claim term "to selectively load required object-oriented methods..." of claim 1 and "selectively loading the object-oriented methods..." of claim 7 are construed to mean "physically or virtually copying, or transferring, required object-oriented methods into the executable memory during runtime before invocation of the object-oriented methods as needed, where those methods do not include or cover classes."

Describing the operation of the preferred embodiment as shown in FIG. 3, the '983 patent specification states:

> After the library server associated with the code library 110 is identified, or if the library server was already known, then step 314 is processed. In step 314, a request is sent to the library server asking the library server to copy the computer program logic associated with the method reference in the statement to the task address space. Upon

[31] As noted by Apple, "[i]n a preferred embodiment, this loading is done by 'virtual copy' – setting a pointer that resides in the executable program memory part of the RAM to the portion of the code library (located in another part of RAM) containing the methods to be invoked." Apple Br. at 54-55, citing JX-4 at col. 9, lns. 42-50.

WI-Apple1604887

> completion of step 314, the library server has copied the requested computer program logic to the task address space. Preferably, the code library 110 is a shared library. That is, the code library 110 may be simultaneously accessed by multiple threads. However, preferably the computer program logic of the code library 110 is physically stored in only one physical memory area. The library server virtually copies computer program logic from the code library 110 to task address spaces. That is, instead of physically copying computer program logic from one part of physical memory to another, the library server places in the task address space a pointer to the physical memory area containing the relevant computer program logic. In step 316, the computer program logic associated with the object-oriented statement is executed on the computer platform 102. As noted above, in the case where the object-oriented statement accesses the operating system 114, the computer program logic associated with the method contains at least one procedural function call which is compatible with the native procedural interface of the operating system 114. Thus, by executing the method's computer program logic, the procedural function call is invoked and executed, thereby causing the operating system 114 to provide the service on behalf of the application 130A.

JX-4 at col. 9, lns. 32-61 (emphasis added).

The specification thus confirms that the '983 patent is not limited to physical copying. The description of the preferred embodiment encompasses virtual copying as complainants argue. HTC's and the Staff's attempt to limit copying to physical copying and to read out the preferred embodiment of virtual copying is contrary to the explicit disclosure of the '983 patent and must fail. *See Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1583–84 (Fed. Cir. 1996) (a construction that reads out a preferred embodiment "is rarely, if ever, correct and would require highly persuasive evidentiary support").

HTC and the Staff nonetheless argue that claims 1 and 7 cannot include virtual

WI-Apple1604888

copying, contending that the Applicants disclaimed virtual copying during prosecution of the '983 patent. This allegation is not supported by the record and falls short of the heavy burden required for a finding of a prosecution disclaimer. In order to find a prosecution disclaimer, there must be a showing that the Applicants made a "clear and unmistakable" disavowal of loading code by a means other than physical copying. *See Purdue Pharma L.P. v. Endo Pharms. Inc.*, 438 F.3d 1123, 1136 (Fed. Cir. 2006).

During the prosecution of the '983 patent, the Examiner found all claims invalid in light of U.S. Patent No. 5,247,681 to Janis et al. JX-10 at APPHTC_00012891-98. To distinguish Janis as prior art, the Applicants argued:

> In response, the Applicant asserts that the cited section at Column 3, lines 24-37 of the Janis reference teaches away from the Applicant's claimed invention. Janis is describing sharing previously loaded software modules. Janis says it twice in the same quotation used by the Examiner. The Applicant is claiming a runtime loader that selectively loads the required object-oriented methods into the executable program memory during runtime before invocation of the object-oriented methods. The Janis reference teaches away from the Applicant's claimed invention. A prior art reference must be considered in its entirety, i.e., as a whole, including portions that would lead away front the claimed invention. W.L. Gore & Associates, Inc. v. Garlock. Inc., 721 F.2d 1540, 220 USPQ 303 (Fed. Cir. 1983), cert. denied, 469 U.S. 851 (1984).

JX-10 at APPHTC_00012996 (Amendment at 12, Dec. 28, 2000) (emphasis in original). The Examiner subsequently granted the patent. JX-10 at APPHTC_00013000. HTC asserts that the above paragraph shows that to distinguish Janis as prior art, "the Applicants argued that sharing previously loaded software modules (*i.e.*, virtual copying) taught away from the claimed invention." HTC Br. at 67.

HTC and the Staff are conflating two different issues. Sharing previously loaded

WI-Apple1604889

software modules is not the same as virtual copying, and the Applicants never argued during prosecution that virtual copying is not covered by the claims. JX-10 at APPHTC_00012996. Furthermore, Dr. Jeffay (HTC's expert) conceded at the hearing that the portion of Janis actually cited by the examiner and responded to by Applicants has nothing whatsoever to do with how the code is copied, and instead focuses entirely on where the code is copied. Jeffay Tr. 3542-49.

In addition, Ms. Spielman (Apple's expert) made clear, and Dr. Jeffay did not contest, that the Applicants distinguished Janis on the basis of ***where*** that system loaded code from (loading software modules that were already in executable program memory) – not ***how*** the code was loaded (whether by sending, transferring, copying, or otherwise). Spielman Tr. 1951-52; Orton Tr. 1877-79; Jeffay Tr. 3542-43. Thus HTC ignores the distinction that the Applicants actually made over Janis. HTC urges that "[t]he Janis system kept track of software modules previously loaded into memory" (*id.*), but ignores that the functionality disclosed in Janis—loading the previously loaded modules ***from executable memory***—is fundamentally different from the '983 invention which loads code into executable memory ***from non-executable memory***. *Id.*; Spielman Tr. 1950-51.

Moreover, if Apple clearly disclaimed "virtual copying" or placing pointers within the "task address space" as argued by HTC and the Staff, then the Examiner would not have allowed non-asserted claim 16. Claim 16 includes multiple claim elements that cover the concept of "virtual copying" as disclosed in the specification, including "running an object-oriented program in a task address space of the memory" (element 5), "determining during runtime whether said … code is available in said task address space" (element 6), and "loading … code into said task address space during runtime" (element

7).

In addition, Apple argues that "the '983 patent makes explicit that class loading is covered by the asserted claims." Apple submits that "[t]here is nothing in the '983 patent or its file history that imposes any limitation of 'selectively loading' in claims 1 and 7 to extracting only a single method from its class, as opposed to loading the class containing the desired method (but not loading other classes)." Apple Br. at 62.

Apple is incorrect. The plain language of the claim requires selectively loading "object-oriented methods" rather than "object-oriented classes." Importantly, the second step of non-asserted independent method claim 12 provides guidance on this issue. The second step of claim 12 recites:

> providing an object-oriented interface, executing on the computer hardware environment, and responsive to object-oriented programming, for instantiating objects from object-oriented classes, encapsulating data for exclusive use with each object, and invoking object-oriented methods in the objects for operating on the encapsulated data;

JX-4 at col. 39, lns. 29-35 (emphasis added).

A person of ordinary skill reading the above step's use of both the "object-oriented classes" and the "object-oriented methods" in the same claim element would conclude that the "object-oriented classes" and the "object-oriented methods" are indeed not interchangeable as they are recited in claim 12. Similarly, unless the patentee explicitly uses the "object-oriented classes" language in asserted claim 1, a person of ordinary skill would not conclude that the "object-oriented methods" of claim 1 is broad enough to encompass "object-oriented classes." Since the patentee did not so claim, Apple's argument that "loading an entire class in order to load particular methods" would

be covered by asserted claim 1 is rejected.

Moreover, it is undisputed that "methods" are not "classes," and that classes can contain dozens, if not hundreds, of methods. Spielman Tr. 2138-39, 2209. In that regard, Inventor Debra Orton testified that there is a distinction between methods and classes, and a distinction between loading classes and loading individual methods:

> Q. When you refer to objects, what are you referring to?
>
> A. Objects are also somewhat used interchangeably with classes. That would be loading by classes. ***But it was perfectly possible to simply load a method as well.***

Orton Tr. 1792 (emphasis added).

Ms. Spielman (Apple's expert) also admitted that class loading differs from runtime loading of methods. She testified that class loading loads all the code for a class (without checking to see what will be executed), while runtime loading loads only the methods that will be executed. Spielman testified:

> Q. So runtime loading is a method of loading in which you are only loading the code that will actually be executed during runtime. Fair?
>
> A. Yes, that's the statement I make.
>
> Q. Okay. And that's what you state to distinguish what this patent is using, runtime loading, versus what you call static loading, right?
>
> A. Yes.
>
> *****
>
> Q. Would you agree with me that a class can contain method code that will not be executed in a particular program?
>
> A. Sure, if the method is never called, the code for the method will still be there. It might not be executed if no one invokes it, the method.

Spielman Tr. at 2147-48, 2157; *see* Bornstein Tr. 3088. Ms. Spielman further admitted that the asserted claims are directed towards a single method that corresponds to a single object-oriented statement. Spielman Tr. 4751-52, 4763.

Apple counters that "each reference to loading object-oriented methods in the '983 patent could entail loading a class containing the method(s) since the class is the basic unit of object-oriented programming and defines the methods it contains." Apple Br. at 62. To support its argument, Apple points solely to FIG. 4 and col. 10, lns. 14-17 (JX-4), asserting that the code library "stores those methods in classes arranged in a class library." *Id.*

Apple's argument is unpersuasive. The manner in which methods are stored does not determine how those methods are loaded from storage. Thus, a program can store methods in a "class library" but load them selectively, method-by-method, and Apple does not point to anything to the contrary in the '983 patent. Rather, Apple claims that the logical unit of selective loading is the class because methods are grouped into classes. *Id.* Yet, Apple admits that classes themselves are "arranged in a class library." Thus, by Apple's own logic, loading the entire class library could also be "selectively loading methods."

Further, Ms. Spielman testified that the claims revolve around the identification, determination, loading, and execution of methods – ***not*** classes. Spielman Tr. 4763. The claimed invention operates before the execution of a single object-oriented method (Spielman Tr. 4683-84) and, as detailed in FIG. 3, determines whether that method is in executable program memory and, if not, copies the method into executable program memory.

Accordingly, the claim terms "to selectively load required object-oriented methods into the executable program memory during runtime before invocation of the object-oriented methods" (claim 1) and "selectively loading the object-oriented methods into the executable program memory during runtime before invocation thereof" (claim 7) are construed to mean "physically or virtually copying, or transferring, required object-oriented methods into the executable memory during runtime before invocation of the object-oriented methods as needed, where those methods do not include or cover classes."

2. "a procedural operating system, having a native interface"

Claim Term	Apple Construction	HTC and Staff[32] Construction
"a procedural operating system, having a native interface" (claim 1) "procedural operating system including a native interface" (claim 7)	an operating system having a procedural interface that includes procedural functions which are called to access services	a procedural operating system: an operating system that provides a procedure-oriented environment in which to develop and execute software native interface: interface to an operating system in the same format as the operating system

Joint Claim Construction, App'x A at 1.

As proposed by Apple, the claim terms "a procedural operating system, having a native interface" (claim 1) and "procedural operating system including a native interface" (claim 7) are construed to mean "an operating system having a procedural interface that includes procedural functions which are called to access services."

[32] The Staff did not offer any arguments regarding this claim limitation in its post-hearing briefs.

WI-Apple1604894

As a threshold matter, this claim limitation is construed in a unified manner as proposed by Apple, rather than separately construing "procedural operating system" and "native interface" as proposed by HTC and the Staff. Both claims 1 and 7 require that the limitation be construed as a whole. Spielman Tr. 1927-28. Specifically, claim 1 requires "a procedural operating system, having a native interface, for controlling the computer hardware to perform the native system services." Similarly, claim 7 requires "the procedural operating system including a native interface, responsive to procedural function calls, for providing native system services." Because of the way in which "procedural operating system" and "native interface" are tied together in the claims, Apple's unified construction is correct.

The '983 patent specification supports this construction. It explicitly describes the claimed invention in terms of a procedural operating system having a native interface. Spielman Tr. 1928-31. For example, the Summary of the Invention states that "[t]he present invention is directed to a system and method of enabling an object-oriented application to access in an object-oriented manner a procedural operating system having a native procedural interface." JX-4 at col. 3, lns. 45-48 (emphasis added). Further, the Detailed Description of the Preferred Embodiments states that "[t]he computer platform 102 also includes a procedural operating system 114 having a native procedural interface (not shown)." JX-4 at col. 5, lns. 33-35 (emphasis added).

The specification of the '983 patent discloses the following regarding operating systems:

> It should be noted that the operating system 114 may represent a substantially full-function operating system, such as the Disk Operating System (DOS) and the UNIX

WI-Apple1604895

> operating system. However, the operating system 114 may represent other types of operating systems. For purposes of the present invention, the only requirement is that the operating system 114 be a procedural operating system having a native procedural interface. Preferably, the operating system 114 represents a limited functionality procedural operating system, such as the Mach micro-kernel developed by CMU, which is well-known to those skilled in the relevant art.

JX-4 at col. 5, lns. 45-56 (emphasis added).

Thus, as explained by the specification, "operating system 114 may represent a substantially full-function operating system" as one possible embodiment of the '983 patent. However, the specification also makes clear that "For purposes of the present invention, the only requirement is that the operating system 114 be a procedural operating system having a native procedural interface." Additionally, the specification explains that "limited functionality procedural operating system, such as the Mach micro-kernel" is a preferred embodiment of operating system 114.

The specification provides additional similar disclosure regarding operating systems:

> As noted above, the present invention shall be described herein with reference to the Mach micro-kernel, although the use of the present invention to wrap other operating systems falls within the scope of the present invention.

JX-4 at col. 7, lns. 6-10 (emphasis added).

Thus, the specification portion cited above makes clear that the operating system of the '983 patent can take many forms, including the preferred embodiment of the Mach micro-kernel.

In addition, HTC's construction is further contradicted by testimony from Ms. Orton, the first named Inventor on the '983 patent. Orton Tr. 1875-77 (testifying that

WI-Apple1604896

"we had envisioned a wide range of possible operating systems, from limited functionality all the way up to full blown bells and whistles"); *see* Spielman Tr. 2243-46.

HTC reads in an added limitation—an "environment in which to develop and execute software"—that is not required by the claims. Spielman Tr. 1934-35. Indeed, the only discussion of any development or execution environment in the '983 patent is found at column 3, lines 20-22 ("Thus, conventional operating systems provide procedure-oriented environments in which to develop and execute software."). HTC's conversion of one mention of a development and execution environment into a claim requirement in all instances is inconsistent with the disclosure provided at column 5, lines 45-56, cited above, which makes clear that while such functionality ***may*** be provided by an operating system that would meet the asserted claims, it is ***not required***. Spielman Tr. 1934-35; Orton Tr. 1875-76.

3. "means for making determinations during runtime execution if object-oriented methods to be invoked are present in the executable program memory"

Claim Term	Apple Construction	HTC and Staff Construction
"means for making determinations during runtime execution if object-oriented methods to be invoked are present in the executable program memory" (claim 1)	*Function*: making determinations during runtime execution if object-oriented methods to be invoked are present in the executable program memory *Corresponding Structure*: a computer processor configured to perform the function of element 308 in Fig. 3, as described in the '983 patent specification at col. 8:55-	*Function* making determinations during runtime execution if object-oriented methods to be invoked are present in the executable program memory *Corresponding Structure*: a computer processor configured to perform the function of element 308 in Fig. 3, as described in the '983 patent specification at col. 8:55-59 and col. 9:62-65; '983 File History, June 28, 2000, Response at 5; '983

WI-Apple1604897

Claim Term	Apple Construction	HTC and Staff Construction
	59 and col. 9:62-65;	File History, Dec. 28, 2000, Response at 9 and 12

Joint Claim Construction, App'x A at 2.

The parties agree that this is a means-plus-function term, that the function is "making determinations during runtime execution if object-oriented methods to be invoked are present in the executable program memory," and that the '983 patent specification identifies corresponding structure.

Apple argues that the only remaining dispute is "whether the file history requires additional structure beyond that disclosed in the specification." Complainants propose that "the only required structure is the structure disclosed in the specification that is necessary to perform the claimed function" and additional citations to the file history are not necessary. Apple Br. at 65.

HTC and the Staff argue that the structure for the claim term "means for making determinations" should include "additional citations to the file wrapper that more fully describe the corresponding structure that performs the identified function." HTC Br. at 72-73; Staff Br. at 44.

As proposed by Apple, the structure of the claim term "means for making determinations" is construed to mean "a computer processor configured to perform the function of element 308 in Fig. 3, as described in the '983 patent specification at col. 8, lns. 55-59 and col. 9, lns. 62-65."

The specification describes element 308 as follows:

> Referring now to FIG. 3, in step 308, it is determined whether the computer program logic (also called computer

WI-Apple1604898

> code) from the code library 110 which implements the method referenced in the statement is present in the task address space associated with the application 130A.

JX-4 at col. 8, lns. 55-59.

> The above-described performance in the computer platform 102 of steps 306,[33] 308, 310, 312, and 314 is due, in large part, to the run-time environment established in the computer platform 102.

JX-4 at col. 9, lns. 62-65.

The file history for the '983 patent includes an excerpt from the Preliminary Amendment that is in issue. The excerpt reads:

> In response, the Applicant states that Schmidt fails to disclose or suggest loading the method during runtime before invocation thereof, as claimed by the Applicant. The examiner cites the Schmidt reference to show that the applications built using Schmidt's libraries ultimately make system calls at run-time. This is true of all applications, not merely Schmidt's. Ultimately, the actual calls on system functionality get made at runtime in all cases. What the Applicants are claiming is that the claimed invention can defer the decision about which system implementation to use until run time. The Applicant's claimed invention loads the method during runtime just before invocation thereof. Adding the claim element of "loading the method during runtime before invocation thereof" means that (unlike Schmidt) it is possible to wait until the program is running before the particular library is chosen and used by the program. There is nothing in Schmidt that even suggests this claimed feature.
>
> In the claimed invention, the application can be written and compiled, and only when it is actually running does the particular library get linked to it to specify which actual code (including the code with system calls specific to this platform) would be used. In the case of Schmidt, the developer makes the decision which library to use at development time, not run time. Schmidt then specifies a particular library with which to link (still at development time) and the resulting application is now hard-coded to work on only one particular system (and then, of course, the actual system calls eventually occur at run-time). Thus, in the claimed invention, the choice of which system implementation to use can be deferred to run-time, whereas in Schmidt's disclosed system it is determined prior to run-time and once determined can no longer be changed at run time".

JX-10 at APPHTC_00012889 (Preliminary Amendment at 5, June 28, 2000).

HTC's and the Staff's argument that the above Response requires additional structure (*i.e.*, structure for making a system implementation choice) is unconvincing. As

[33] It is noted that step "306" is not shown in FIG. 3, nor disclosed elsewhere in the specification.

WI-Apple1604899

explained by Ms. Spielman, "system implementation" as discussed in the Response appeared only in applied-for claim 27—which issued as claim 12 and is not an asserted claim. Spielman Tr. 1966-67. Indeed, the last element claim 12 recites: "whereby a choice of which system implementation to use can be deferred to run-time." There is no "system implementation" in asserted claims 1 and 7. *Id.* Thus, the portion of the file history that HTC and the Staff seek to include concerns a wholly different claim limitation from a non-asserted claim.

There is no connection between this portion of the file history and the "means for making determinations" limitation. In fact, the Applicants noted in the cited file history that Schmidt discloses system implementation determinations, so this would not even be a distinction between Schmidt and the '983 claims. JX-10 at HTC00012889. The point of the cited file history is that the '983 invention happens "during runtime," whereas Schmidt disclosed compile-time technology. *Id.* In other words, the Applicants informed the Examiner that the "during runtime" claim limitation was absent in Schmidt. Informing an Examiner that an entire limitation is missing from a prior art reference is *not* a disclaimer or additional structure and HTC's attempt to characterize it as one is misplaced. *See Uniroyal, Inc. v. Rudkin-Wiley Corp.*, 939 F.2d 1540, 1544 (Fed. Cir. 1991) (distinguishing a prior art reference in the file history as failing to disclose a claim limitation does not lead to file history estoppel.).

Additionally, HTC and the Staff further rely on Applicants' December 28, 2000 Response with regard to the Janis reference, and propose that this response should also be included in the construed structure. HTC and the Staff are wrong. As discussed above with respect to the "selectively loading" limitation, the Applicants' discussion of Janis in

the file history focused on distinguishing Janis' disclosure of "sharing previously loaded software modules" *within* executable memory space, as opposed to the claimed invention of the '983 patent that loads necessary methods *into* executable program memory space. Spielman Tr. 1972-73. Applicants' discussion of Janis in the file history adds no additional structure or disclaimers with regard to the "means for making determinations" limitation, since that discussion is focused on the "selective loading" limitation of the asserted claims as opposed to the "means for making determinations" limitation.

In short, there is no connection between the portions of the file history that HTC and the Staff seek to include as structure in the construction of the "means for making determinations" limitation. *See Telcordia Techs., Inc. v. Cisco Sys., Inc.*, 612 F.3d 1365, 1376 (Fed. Cir. 2010). There also is no "clear and unmistakable" disavowal of claim scope, as required to find a file history disclaimer. *Elbex Video, Ltd. v. Sensormatic Elec. Corp.*, 508 F.3d 1366, 1371 (Fed. Cir. 2007).

The only required structure is the structure disclosed in the specification that is necessary to perform the claimed function. *See John Mezzalingua Associates, Inc. v. International Trade Com'n*, 2010 WL 6561393, *4 (Fed. Cir. 2010) ("Claim interpretation under § 112, ¶ 6 does not 'permit incorporation of structure from the written description beyond that necessary to perform the claimed function.'") (*quoting Micro Chem., Inc. v. Great Plains Chem. Co.*, 194 F.3d 1250, 1258 (Fed. Cir. 1999)).

B. Infringement

For the reasons set forth below, Apple has not shown that HTC's accused products infringe the asserted claims of the '983 patent.

The preamble and the first and second elements of apparatus claim 1 recite:

A computer system, comprising:

computer hardware for performing native system services;

a procedural operating system, having a native interface, for controlling the computer hardware to perform the native system services;

The preamble and the first element of method claim 7 recite:

A method for operating a computer system, comprising the steps of:

executing a procedural operating system on computer hardware, the procedural operating system including a native interface, responsive to procedural function calls, for providing native system services;

Computer System Comprising Computer Hardware

The parties do not dispute that Apple has satisfied the preamble and the first element of claim 1, and the preamble and the "computer hardware" aspect of the first element of claim 7. The HTC accused products are computer systems that include computer hardware. For example, the HTC Aria (code name Liberty), which the parties stipulated to as representative of all HTC products, [ ] and RAM and ROM for memory. CX-1001C at HTC000013956; Spielman Tr. 1977; CX-7251C, R. Wu Dep. 32-34; CPX-2.

"procedural operating system"

The claim terms "a procedural operating system, having a native interface" (claim 1) and "procedural operating system including a native interface" (claim 7) have been construed to mean "an operating system having a procedural interface that includes procedural functions which are called to access services." The evidence establishes that

WI-Apple1604902

HTC accused products include [

]

It is undisputed that [

] Ms. Spielman and Dr. Jeffay agree that Linux is written in the procedural C programming language. Spielman Tr. 1980, 1982-83; Jeffay Tr. 3558. Both Ms. Spielman and Dr. Jeffay also agree that [

]

Moreover, HTC's own documents and witnesses establish that [

] The HTC presentation [

]

Accordingly, the evidence establishes that the HTC accused products have a procedural operating system, namely the Android Linux kernel.

WI-Apple1604903

"native interface"

The HTC accused products include [

]

"native system services"

The HTC accused products include [

]

[

] In fact, Ms. Spielman testified that she analyzed [

WI-Apple1604904

]

The HTC accused products literally infringe under proper construction of "procedural operating system, having a native interface." Spielman Tr. 1979. As described above, proper claim construction requires an operating system [

] having a procedural interface [ ] that includes procedural functions which are called to access services [

]

HTC asserts that [

] However, as discussed above, [

]

HTC also contends that Android has an object-oriented operating system because there is an object-oriented interface in Android. HTC Br. at 89-90. HTC misreads the '983 patent, which uses wrappers precisely for the purpose of providing an object-

WI-Apple1604905

oriented interface to an underlying procedural operating system, just like Android. JX-4 at col. 5, lns. 1-4; col. 6, lns. 25-40. Indeed, Android [

]

Finally, HTC urges that even if [

]

First, [

]

Second, Dr. Jeffay agreed that the native interface in the '983 patent is the procedural interface as shown in [

]

A comparison of the native interface (below) in Android as identified by Ms. Spielman, and in the '983 patent as identified by Dr. Jeffay, shows that the architecture of the '983 patent is the same as Android.

WI-Apple1604906

[

]

Accordingly, HTC's non-infringement position is rejected.

The third and fourth elements of apparatus claim 1 recite:

> **object oriented methods requiring native system services;**
>
> **procedural program logic code, responsive to invocations of the object-oriented methods during runtime, for causing the procedural operating system to control the computer hardware to perform the required native system services;**

The second element of method claim 7 recites:

> **issuing calls during runtime, compatible with the native interface, to provide the native system services in response to invocations of object-oriented methods requiring such native system services;**

"object oriented methods requiring native system services" (claim 1) and "object-oriented methods requiring such native system services" (claim 7)

The evidence establishes, and HTC does not dispute, that the HTC accused

WI-Apple1604907

products include object-oriented methods requiring native system services. For example, the Phone application source code includes [

]

"procedural program logic code" (claim 1) and "issuing calls during runtime, compatible with the native interface" (claim 7)

Apple's expert, Ms. Spielman, testified regarding two code traces to show how the HTC accused products satisfy these limitations of claims 1 and 7. HTC's expert does not dispute that the HTC accused products practice this limitation.

Phone Application code trace

The Phone application, an Android application (CX-7432C, A. Hsieh Dep. 95-96), [

WI-Apple1604908

]

Specifically, [

]

Zygote Application code trace

[

]

[

]

The fifth and sixth elements of apparatus claim 1 recite:

> **executable program memory associated with the computer hardware for runtime execution of the procedural operating system, invocations of the object-oriented methods and related portions of the procedural program logic code;**
>
> **means for making determinations during runtime execution if object-oriented methods to be invoked are present in the executable program memory; and**

The third element of method claim 7 recites:

> **determining during runtime if object-oriented methods to be invoked during runtime execution are present in executable program memory associated with the computer hardware; and**

"executable program memory"

Apple has not shown that HTC accused products include "executable program memory" as required by the fifth element of claim 1. This limitation requires "executable program memory . . . for runtime execution of the procedural operating system, invocations of the object-oriented methods and related portions of the procedural program logic code." That is, claim 1 requires a single "executable program memory" allowing both execution of the procedural operating system *and* invocation of the object-oriented methods and related procedural logic. Thus, the "executable program memory" must be the same memory for performing both claimed requirements. However, Apple identifies *two distinct and non-overlapping* areas of memory to satisfy this limitation: (1) the "kernel space" for the first requirement, and (2) the "task address space, which resides in an application's heap and is characterized as "dirty' memory" for the second requirement.

WI-Apple1604910

Apple Br. at 75-76.

In Android, [

] Apple Br. at 76. This assertion is incorrect. Daniel Bornstein, who designed Dalvik, confirmed that [

] Thus, Dalvik's class resolution functionality—accused by Apple of infringing the '983 patent—cannot even discern whether [

]

Moreover, the '983 patent does not mention clean or dirty memory. Also, named Inventor Debra Orton testified that this distinction "didn't matter" in her invention. Orton Tr. 1845-46. Indeed, Apple's expert, Susan Spielman, admitted that executable program memory includes clean as well as dirty memory:

> Q. Okay. So some clean memory is executable memory?
>
> A. Yes, ***if a program is executable, it can run in clean memory***. Clean memory is typically unwritten memory. So the attributes of the memory are defined by whatever is being—that memory is being used for. So in the Android presentation that was given to the Android developers, that's the term that they chose to use to define the attributes in their system.

Spielman Tr. 2215-16 (emphasis added).

Accordingly, Apple's contention that "dirty" memory is the only "executable program memory," or that preloaded .dex files are not loaded into "executable program memory" because they are allegedly mapped to clean rather than dirty memory, is rejected.

"means for making determinations during runtime execution if object-oriented methods to be invoked are present in the executable program memory"

Apple has not shown that the HTC accused products determine during runtime if object-oriented methods are present in executable program memory. The HTC accused devices do not include the claimed "means for making determinations" of the '983 patent for two reasons.

First, the accused products running Android [

]

Claims 1 and 7 require a determination during runtime if object-oriented methods to be invoked are present in executable program memory. The accused products, however, do not determine during runtime if object-oriented methods to be invoked are present in executable program memory. [

]

In that regard, Daniel Bornstein (Google tech leader and manager of the Dalvik team within the Android project) explained that [

]

[

]

Android's "load everything" approach is supported by ample evidence and is not disputed by Apple. Bornstein Tr. 3105-06. Apple's expert, Ms. Spielman, testified that [

]

This difference stems from the fact that Android operates in a fundamentally different environment than contemplated in the '983 patent. Jeffay Tr. 3278-79. Unlike users of early 1990s desktop computers, modern smartphone users generally demand much faster response times. Bornstein Tr. 3118-19. Since runtime loading is a complex operation which drains battery power and can cause glitches or delays in the user experience, [

]

WI-Apple1604913

Additionally, Apple accuses [ ] but this code determines [ ] not whether a method used by an existing object has been loaded into executable program memory. Apple specifically accuses [

]

The code itself confirms that [

] Thus, the opinion offered by Ms. Spielman is rejected.

[

] According to Dr. Jeffay, this simplicity in design

WI-Apple1604914

speeds up performance. *Id.*

Apple argues that because [

] Thus, Apple's infringement argument regarding classes is rejected.

Apple further argues that HTC accused products infringe under the doctrine of equivalents. Apple again misses the mark. Determining whether a class already in memory is resolved constitutes a substantially different function, performed in a substantially different way, to arrive at a substantially different result than determining whether a specific method code is in memory. Jeffay Tr. 3370-71.

The last element of apparatus claim 1 recites:

> **a runtime loader, responsive to the determinations, to selectively load required object-oriented methods into the executable program memory during runtime before invocation of the object-oriented methods.**

The last element of method claim 7 recites:

> **selectively loading the object-oriented methods into the executable program memory during runtime before invocation thereof, if not yet loaded.**

Apple has not satisfied these claim elements. The claim term "to selectively load required object-oriented methods…" of claim 1 and "selectively loading the object-oriented methods…" of claim 7 have been construed to mean "physically or virtually

copying, or transferring, required object-oriented methods into the executable memory during runtime before invocation of the object-oriented methods as needed, where those methods do not include or cover classes." Apple cannot prove infringement of these claim elements because it cannot show that the HTC accused devices "selectively load required object-oriented methods." It is undisputed that Android does not selectively load methods.

The Android platform includes [

]

Moreover, it is undisputed that [

WI-Apple1604916

] Accordingly, it cannot be the case that

[

]

Apple admits that [

]

Apple also argues that if this claim element is "interpreted so narrowly as to require method-by-method (as opposed to class) loading, then HTC still infringes under the doctrine of equivalents." Apple Br. at 85. Apple contends that [

] Apple further explains:

[

]

Apple's conclusory contention regarding doctrine of equivalents is unconvincing. As explained by HTC's expert, Dr. Jeffay, the accused HTC products perform a substantially different function in a substantially different way (class resolution as

opposed to method loading) to arrive at a substantially different result (resolving classes already in the task address space) than what is claimed in the '983 patent. Jeffay Tr. 3366-68. Indeed, Android is [

]

In summary, Apple has not shown that HTC's accused products infringe the asserted claims of the '983 patent.

C. Technical Prong of the Domestic Industry Requirement

Apple argues that "[t]he only dispute raised by HTC is whether the "selective loading" limitation of claims 1 and 7 is met by the MacBook Pro, which selectively loads the required object-oriented methods [

] Apple further contends that its "MacBook Pro meets the 'selective loading' limitations of claims 1 and 7 by [

] Apple Br. at 86.

HTC and the Staff agree with Apple that the only remaining dispute is with respect to "selectively loading" limitation, but argue that Apple's MacBook Pro does not satisfy this limitation. HTC Br. at 91; Staff Br. at 57-58.

For the reasons set forth below, Apple has not satisfied the technical prong of the domestic industry requirement with respect to the '983 patent.

The preamble and the first and second elements of apparatus claim 1 recite:

WI-Apple1604918

A computer system, comprising:

computer hardware for performing native system services;

a procedural operating system, having a native interface, for controlling the computer hardware to perform the native system services;

The preamble and the first element of method claim 7 recite:

A method for operating a computer system, comprising the steps of:

executing a procedural operating system on computer hardware, the procedural operating system including a native interface, responsive to procedural function calls, for providing native system services;

Computer System Comprising Computer Hardware

Apple has satisfied the preamble and the first element of claim 1, and the preamble and the "computer hardware" aspect of the first element of claim 7, which are not disputed by the parties. It is undisputed that Apple's MacBook Pro is a computer system that includes computer hardware for performing native system services, such as a processor, memory, and input/output devices. Spielman Tr. 2106-07; CX-802.

"procedural operating system"

It is undisputed that Apple's MacBook Pro includes the Darwin operating system, a procedural operating system. Spielman Tr. 2107-08; CX-803 at APPHTC_00001890, APPHTC_00002034. The Mach microkernel called out as the preferred embodiment in the '983 patent is part of the Darwin operating system. Spielman Tr. 2109; JX-4 at col. 5, lns. 53-56; CX-803 at APPHTC_00001894.

WI-Apple1604919

"native interface" and "native system services"

Darwin is accessible via a procedural native interface and [

] accessed via procedural function calls. Spielman Tr. 2109-11; CX-803 at APPHTC_00001901; CX-4379C at APPHTC-S_00001203-04; CX-4374C at APPHTC-S_00001090-91. Moreover, Darwin [

] Spielman Tr. 2112; CX-803 at APPHTC_00001901.

The third and fourth elements of apparatus claim 1 recite:

> **object oriented methods requiring native system services;**
>
> **procedural program logic code, responsive to invocations of the object-oriented methods during runtime, for causing the procedural operating system to control the computer hardware to perform the required native system services;**

The second element of method claim 7 recites:

> **issuing calls during runtime, compatible with the native interface, to provide the native system services in response to invocations of object-oriented methods requiring such native system services;**

Apple has satisfied this claim element.

"object oriented methods requiring native system services" (claim 1) and "object-oriented methods requiring such native system services" (claim 7)

It is undisputed that the MacBook Pro includes [

] which are object-oriented methods requiring native system services. At the hearing, Ms. Spielman detailed [

] Spielman

WI-Apple1604920

2112-17; CX-4363C; CX-4386C; CX-4387C; CX-4388C; CX-4473C; CX-4379C]

"procedural program logic code" (claim 1) and "issuing calls during runtime, compatible with the native interface" (claim 7)

It is undisputed that the MacBook Pro includes procedural program logic code, responsive to invocations of object-oriented methods during runtime, to control the computer hardware to provide native system services. Spielman Tr. 2115-17; CX-4473C at APPHTC-S_00013309; CX-4379C at APPHTC-S_00001203. Specifically, [

] *Id.*

The fifth and sixth elements of apparatus claim 1 recite:

> **executable program memory associated with the computer hardware for runtime execution of the procedural operating system, invocations of the object-oriented methods and related portions of the procedural program logic code;**
>
> **means for making determinations during runtime execution if object-oriented methods to be invoked are present in the executable program memory; and**

The third element of method claim 7 recites:

> **determining during runtime if object-oriented methods to be invoked during runtime execution are present in executable program memory associated with the computer hardware; and**

"executable program memory"

It is undisputed that the MacBook Pro includes executable program memory for runtime execution of the Darwin operating system, invocations of Java methods, and related portions of procedural program logic code. Spielman Tr. 2117-18. Specifically, the MacBook Pro provides executable application memory as well as operating system

WI-Apple1604921

memory. Spielman Tr. 2118; CX-803 at APPHTC_00001894.

> **"means for making determinations during runtime execution if object-oriented methods to be invoked are present in the executable program memory"**

It is not disputed that the MacBook Pro makes determinations whether object-oriented [ ] methods to be invoked are present in executable program memory. Spielman Tr. 2118-19. Specifically, the MacBook Pro includes the agreed structure of a CPU that will perform step 308 of FIG. 3 in the '983 patent. Spielman Tr. 2118-19; JX-4, FIGS. 1 and 3. Specifically, the MacBook Pro includes [ ] that checks to see if the [ ] methods are present in executable program memory. Spielman Tr. 2119-20; CX-802 at APPHTC_00001869; CX-4468C at APPHTC-S_00013399. At the hearing, Ms. Spielman described [

] Spielman Tr. 2119-22; CX-4383C; CX-4390C; CX-4391C; CX-4394C; CX-4467C; CX-4468C.

The last element of apparatus claim 1 recites:

> **a runtime loader, responsive to the determinations, to selectively load required object-oriented methods into the executable program memory during runtime before invocation of the object-oriented methods.**

The last element of method claim 7 recites:

> **selectively loading the object-oriented methods into the executable program memory during runtime before invocation thereof, if not yet loaded.**

Apple has not satisfied these claim elements which require selective loading. Apple has failed to prove that the MacBook Pro running Snow Leopard practices these

WI-Apple1604922

limitations for the same reason that Apple was unable to demonstrate that HTC infringes these elements. Apple contends that "MacBook Pro includes a runtime loader that selectively loads required object-oriented [ ] methods into executable program memory during runtime, before invocation thereof." Apple Br. at 88. Apple explains that [

] Id. at 88-89.

Thus, as Apple admits, [ ] Indeed, Apple's expert Ms. Spielman admits that [

]

Spielman Tr. 2121-22, 2124-25 [

] *see* Jeffay Tr. 3377-78 [

] CX-4383C at APPTHTC-S_00001281-82 (classLoader.cpp). Thus, this loading is not selective with regard to ***methods*** to be invoked as the claims require.

Accordingly, Apple has failed to show that it practices the "to selectively load required object-oriented methods into the executable program memory during runtime before invocation of the object-oriented methods" limitation of claim 1, or the "selectively loading the object-oriented methods into the executable program memory during runtime before invocation thereof, if not yet loaded" limitation of claim 7.

In summary, Apple has not satisfied the technical prong of the domestic industry

WI-Apple1604923

requirement with respect to the '983 patent.

D. Validity

HTC states that it "does not contend that the '983 patent is invalid under HTC's and the Staff's constructions, which require selectively loading ***methods***. HTC's invalidity theories are advanced solely under Apple's constructions, which remove the limitation of '***selectively*** loading the object-oriented ***methods***'." HTC Reply at 59 n.31. The Staff agrees that HTC only argued invalidity of the '983 patent only under Apple's proposed claim constructions. Staff Br. at 57-58.

The undersigned agreed with HTC and the Staff that the proper construction of the last elements of claims 1 and 7 require selective loading of *methods,* which is different from selective loading of *classes*. Thus, HTC's contentions regarding invalidity of the '983 patent is no longer in play.

Accordingly, HTC has not shown by clear and convincing evidence that the NeXTSTEP Release 3 System anticipates asserted claims 1 and 7 of the '983 patent. Further, HTC has not shown by clear and convincing evidence that the combination of Vernon and Gautron references renders obvious asserted claims 1 and 7 of the '983 patent.

IX. U.S. Patent No. 5,946,647

The '647 patent is entitled, "System and Method For Performing An Action On A Structure In Computer-Generated Data." The Summary of the Invention states, in part:

> The present invention overcomes the limitations and deficiencies of previous systems with a system that identifies structures in computer data, associates candidate actions with each detected structure, enables the selection

WI-Apple1604924

> of an action, and automatically performs the selected action on the identified structure. It will be appreciated that the system may operate on recognizable patterns for text, pictures, tables, graphs, voice, etc. So long as a pattern is recognizable, the system will operate on it. The present invention has significant advantages over previous systems, in that the present system may incorporate an open-ended number and type of recognizable patterns, an open-ended number and type of pattern analysis units, and further that the systems may enable an open-ended number and type (i.e. scripts, macros, code fragments, etc.) of candidate actions to associate with, and thus perform, on each identified structure.

JX-3 (Summary of the Invention) at col. 2, lns. 4-20.

Apple asserts apparatus claims 1, 3, and 8 and method claims 15 and 19. Claims 3 and 8 depend on independent claim 1 and claim 19 depends on independent claim 15. The asserted claims read as follow:

> **1**. A computer-based system for detecting structures in data and performing actions on detected structures, comprising:
>
> an input device for receiving data;
>
> an output device for presenting the data;
>
> a memory storing information including program routines including
>
> an analyzer server for detecting structures in the data, and for linking actions to the detected structures;
>
> a user interface enabling the selection of a detected structure and a linked action; and
>
> an action processor for performing the selected action linked to the selected structure; and
>
> a processing unit coupled to the input device, the output device, and the memory for controlling the execution of the program routines.

WI-Apple1604925

> **3**. The system recited in claim **1**, wherein the input device receives the data from an application running concurrently, and wherein the program routines stored in memory further comprise an application program interface for communicating with the application.
>
> **8**. The system recited in claim **1**, wherein the user interface highlights detected structures.
>
> **15**. In a computer having a memory storing actions, a method for causing the computer to perform an action on a structure identified in computer data, comprising the steps of:
>
> receiving computer data;
>
> detecting a structure in the data;
>
> linking at least one action to the detected structure;
>
> enabling selection of the structure and a linked action; and
>
> executing the selected action linked to the selected structure.
>
> **19**. The method recited in claim **15**, wherein the memory contains strings, and wherein the step of detecting a structure further comprises the steps of retrieving a string from the memory and scanning the data to identify the string.

JX-3 at col. 7, lns. 8-24, 27-32, 50-51; col. 8, lns. 22-33, 47-50.

A. Claim Construction[34]

The parties agree to the meaning of "detecting" / "detected," "structure," "analyzer server," and "application running concurrently."[35] The parties dispute the

[34] With respect to the '647 patent, a person of ordinary skill in the art has at least a B.S. degree in computer science (or equivalent coursework) and two to three years of academic or work experience in the field. Mowry Tr. 2459-62; Olsen Tr. 3839-3841; Staff Br. at 60 n.22.

[35] The parties agree that (1) detecting" / "detected" means "finding and identifying" /

WI-Apple1604926

meaning of "linking actions to the detected structures" / "linking at least one action to the detected structure" and "input device." These disputed terms are discussed below.

1. "linking actions to the detected structures" and "linking at least one action to the detected structure"

Claim Term	Apple Construction	HTC and Staff Construction
linking actions to the detected structures (claim 1)	linking detected structures to computer subroutines that cause the CPU to perform a sequence of operations on the particular structures to which they are linked	linking a detected structure to a computer subroutine that causes the CPU to perform a sequence of operations on that particular structure to which it is linked, rather than an informational structure
linking at least one action to the detected structure (claim 15)	linking a detected structure to at least one computer subroutine that cause the CPU to perform a sequence of operations on the particular structure to which it is linked	linking a detected structure to a computer subroutine that causes the CPU to perform a sequence of operations on that particular structure to which it is linked, rather than an informational structure

Joint Claim Construction, App'x A at 16.

The parties' constructions differ in two respects. First, HTC's and the Staff's proposals add the phrase "rather than an informational structure." Second, HTC and the Staff contend that claim 1's "linking actions" term does not require multiple linked actions.

"found and identified;" (2) "structure" means "an instance of a pattern, where a 'pattern' refers to data, such as grammar, regular expression, string, etc., used by a pattern analysis unit to recognize information in a document such as dates, addresses, phone numbers, etc.; (3) "analyzer server" means a program sub-routine that receives data from a document having recognizable structures, and uses patterns to detect the structures;" and (4) "application running concurrently" means "application running during the same run-time." Joint Claim Construction, App'x A at 16; *see* Mowry Tr. 2490; Olsen Tr. 3883-84.

WI-Apple1604927

As proposed by Apple, the claim term "linking actions to the detected structures" of apparatus claim 1 is construed to mean "linking detected structures to computer subroutines that cause the CPU to perform a sequence of operations on the particular structures to which they are linked." Similarly, the claim term "linking at least one action to the detected structure" of method claim 15 is construed to mean "linking a detected structure to at least one computer subroutine that causes the CPU to perform a sequence of operations on the particular structure to which it is linked."

During prosecution, the Examiner rejected the pending claims as obvious based on the Sobotka reference in in combination with U.S. Patent No. 5,247,437 ("Vale"). The Examiner contended that Vale disclosed "linking between DIN and HN structure." JX-9 at APPHTC_00338339-40 ('647 File History, Dec. 6, 1998 Office Action at 2-3). In response, the Applicants argued:

> The linked actions of the claimed invention are patentably distinguished from the heading node (HN) structure of *Vale*. The linked actions enable execution of an action, which is a computer subroutine causing a CPU to perform a sequence of operations. Additionally, in the claimed invention, "[a]n action may specify opening another application, loading the identified structure into an appropriate field, and closing the application. An action may further include internal actions... and external actions...." Thus the linked actions can cure deficiencies of prior systems employing laborious and disruptive processes.
>
> In contrast, the HN structure of *Vale* consists of heading nodes, each of which includes the title of its associated index entry and defines information for one of the headings listed in the heading column of a keyword list. Each heading node also stores heading string, a sort string, a see string, and a heading ID. The HN structure is thus used to delineate the structural relationship of the key words or headings for a given index, but cannot cause a CPU to

WI-Apple1604928

> perform an operation, or an action, as can the claimed invention.
>
> In brief, *Sobotka* does not teach or suggest *linking a structure to an action*. *Vale* discloses *linking to an informational structure*, but that does not cure the references' lack of *linking to an action*. Therefore, the claimed invention, which recites *linking to an action*, or as claimed "linking actions to the detected structures," is patentably distinguished from *Sobotka* and *Vale*, either alone or in combination.

JX-9 at APPHTC_00338619-20 ('647 File History, Mar. 15, 1999 Amendment at 8-9) (citations omitted) (emphasis in original).

Applicants distinguished Vale because it does not disclose linking to an action. Rather, Vale describes linking ***only*** to an informational structure, the "HN" or "heading node," which represents a type of index. Importantly, the heading node is not an action, does not contain an action, and does not link to an action. Mowry Tr. 2470-71 ("Vale does not link to an action"), Tr. 2472 (Heading nodes represent information in an index.), Tr. 2474-75 (Heading nodes do not lead to an action and, in Vale, there are no actions, *i.e.*, operations that are performed on a detected structure, because there are no detected structures.); JX-9 at APPHTC_00338620 ("Vale discloses linking to an informational structure, but that does not cure the references' lack of linking to an action.").

Accordingly, the Applicants did not disclaim use of informational structures. Rather, they emphasized that there must be a link to an action. Mowry Tr. 2471-72, 2661-62 (no disclaimer). Apple's proposed construction, which does not include the phrase "rather than an informational structure," more accurately defines the "linking

actions" claim terms.[36]

Moreover, the phrase "rather than an informational structure" is unnecessary. Indeed, in its opening statement at trial, HTC based its non-infringement case on a single contention – *i.e.,* that the HTC devices "link[] to an informational structure," and ***not*** to an action. In advancing this argument, HTC counsel stated, "Vale discloses linking to an informational structure. The claimed invention links to an action. Our devices work like Vale, not like the '647." Tr. 169. Later, HTC abandoned this distinction by agreeing that the phrase "rather than an informational structure" adds no meaning. Olsen Tr. 4030 (conceding that HTC accused products allow a user to select from a "list of possible actions"), Tr. 3836-37 & 4023-24 (admitting that Apple's and HTC's constructions "are essentially the same.").[37]

Additionally, Apple's constructions for "linking actions ..." (claim 1) and "linking at least one action..." (claim 15) differ in terms of the number of actions that must be linked to a detected structure. For example, Apple's construction for "linking actions..." in claim 1 requires that multiple actions be linked to a detected structure. HTC and the Staff propose that both phrases be construed in the exact same way such that there need only be one action linked to a detected structure. The intrinsic evidence

---

[36] The Staff argues that in the above Amendment, "the Applicant made clear that the required linking was to actions [ ] rather than to 'informational structures,' such as the indexes present in Vale." Staff Br. at 62. The Staff is incorrect for the reasons discussed above.

[37] In fact, as noted, HTC states that "only one of the remaining claim construction issues is outcome determinative," *i.e.*, "whether the term 'input device' can include software only or must include some form of hardware." HTC Br. at 104. (HTC did not even brief the present claim limitation.) *See* HTC Reply at 74 (briefing only the "input device" limitation for claim construction).

WI-Apple1604930

supports Apple's construction.

The language of claim 1 requires "'linking actions," the word "actions" being plural, to detected structures. *See* Mowry Tr. 2477. In contrast, claim 15 requires linking "at least one action" to a detected structure. *Id.* Apple's constructions reflect that clear distinction.[38] Further, the Summary of the Invention describes "candidate actions" (plural) available "[u]pon selection of a detected structure" (singular). JX-3 col. 2, lns. 42-62; *see* Mowry Tr. 2477-78. Dr. Olsen (HTC's expert) did not offer any opinion to the contrary. Olsen Tr. 3837 (referring to "some quibble about plurals" between the parties but offering no opinion in support of HTC's construction).

2. "input device"

Claim Term	Apple Construction	HTC and Staff Construction
input device	computer software or hardware	plain and ordinary meaning (hardware only)

Joint Claim Construction, App'x. A at 16.

As proposed by HTC and the Staff, the claim term "input device" is given its plain and ordinary meaning, *i.e.*, "computer hardware but not computer software."

The specification of the '647 patent discloses the following regarding "input device":

> Referring now to FIG. 1, a block diagram is shown of a computer system 100 including a CPU 120. Computer system 100 is preferably a microprocessor-based computer,

[38] *See Phillips v. AWH Corp.*, 415 F.3d 1303, 1314 (Fed.Cir.2005) ("Differences among claims can also be a useful guide in understanding the meaning of particular claim terms."); *see also Voda v. Cordis Corp.*, 536 F.3d 1311, 1319-1320 (Fed. Cir. 2008) (considering differences in independent claim language in determining scope of claims).

WI-Apple1604931

> such as a Power Macintosh manufactured by Apple Computer, Inc. of Cupertino, Calif. An input device 110, such as a keyboard and mouse, and an output device 105, such as a CRT or voice module, are coupled to CPU 120. ROM 155, RAM 170 and disk storage 175 are coupled to CPU 120 via signal bus 115. Computer system 100 optionally further comprises a printer 180, a communications interface 185, and a floppy disk drive 190, each coupled to CPU 120 via signal bus 115.

JX-3 at col. 3, lns. 22-33 (emphasis added); *see* FIG. 1 (clear denotation of hardware versus software elements).

Thus, the specification shows that "input device" includes hardware that receives input, such as a keyboard and mouse.[39] Indeed, there is nothing in the '647 specification that suggests that the Applicants were importing a special meaning into the term "input device." The disclosure of "input device" in the specification is entirely consistent with the plain and ordinary meaning of the term.

Apple seeks to expand the plain and ordinary meaning of "input device" to include any software that can receive data — regardless of whether hardware is present. Mowry Tr. 2465-66. Dr. Mowry does not dispute that an "input device" can be hardware that receives data. *Id.* at 2467 ("I agree that a hardware input device is an input device."). However, Dr. Mowry then expands such a meaning to conclude essentially that if something (whether it be a program or any software-based application) receives data it must be an "input device." *Id.* at 2466-67. Dr. Mowry reaches this conclusion by asserting that "programs pass information from one program to another. When this occurs, the mechanism that passes the information is software." *Id.* at 2467.

[39] Additionally, Dr. Olsen testified that "input device" includes other hardware such as an input buffer or a touchscreen. Olsen Tr. 4037-38 (describing various portions of the '647 patent and specification in which input devices are physical hardware).

This testimony is far from being convincing. As the Staff notes, simply because software routines pass information between them does not transform them into input devices. Moreover, there is no evidence that anyone in the art would refer to "software" as a device. Indeed, claim 1 of the '647 patent requires that the processing unit be "coupled to" the input device. This requirement would not make much sense if the input device was software running on the processing unit.

Apple's construction runs contrary to the plain meaning of the term "input device" and the intrinsic record and is, therefore, rejected.

B. Infringement

For the reasons set forth below, Apple has shown that HTC's accused products infringe the asserted claims 1, 8, 15, and 19 of the '647 patent. However, Apple failed to show infringement with respect to claim 3.

1. Independent claims 1 and 15

The preamble of independent apparatus claim 1 recites:

> **A computer-based system for detecting structures in data and performing actions on detected structures, comprising:**

The preamble of independent method claim 15 recites:

> **In a computer having a memory storing actions, a method for causing the computer to perform an action on a structure identified in computer data, comprising the steps of:**

Apple has satisfied the preambles of claims 1 and 15. The parties agree that the preambles of claims 1 and 15 are not limitations because they do not give "life, meaning and vitality to the claim[s]." *Altiris, Inc. v. Symantic Corp.*, 318 F.3d 1363, 1371 (Fed.

Cir. 2003); Mowry Tr. 2480. In any event, HTC accused products satisfy the preambles because they are computer-based systems for detecting structures in data and performing actions on detected structures (claim 1) and are computers having memory storing actions that perform an action on a structure identified in computer data (claim 15). Mowry Tr. 2480-81.

The first element of independent apparatus claim 1 recites:

an input device for receiving data;

The first element of independent method claim 15 recites:

receiving computer data;

Apple has satisfied these claim elements. There is no dispute that HTC accused products contain a hardware input device for receiving data and a method of receiving computer data. Mowry Tr. 2481-82; Olsen Tr. 4019-20. These products contain hardware input devices such as wireless-internet adapters for receiving internet data, radios for receiving text messages, a touchscreen, and memory. Mowry Tr. 2482; CPX-3; CPX-4; CPX-5.

The second element of claim 1 recites:

an output device for presenting the data;

Apple has satisfied this claim element. The HTC accused products contain an output device for presenting the data in the form of a touchscreen display. Mowry Tr. 2487; CPX-3; CPX-4; CPX-5.

The third element of claim 1 recites:

a memory storing information including program

routines including

Apple has satisfied this claim element. As confirmed by HTC's user guides and an analysis of the physical devices, the accused products contain memory storing information including program routines. Mowry Tr. 2488-90; CPX-3; CPX-4; CPX-5; CX-391; CX-3510 (Droid Incredible User Guide) at 301.

The fourth element of independent apparatus claim 1 recites:

an analyzer server for detecting structures in the data, and for linking actions to the detected structures;

The second and third elements of independent method claim 15 recite:

detecting a structure in the data;

linking at least one action to the detected structure;

Apple has satisfied these claim elements. As seen by the plain language of the claims, the fourth element of apparatus claim 1 coincides with the second and third elements of method claim 15. Thus, claim 1 has two requirements. First, the analyzer server must detect structures in the data. Second, the same analyzer server must link actions to the detected structures.

As an initial matter, the agreed-upon construction of "structure" requires, among other things, an "instance of a pattern," which is a "positive match of a pattern to something in a document." Mowry Tr. 2491. Examples of recognizable structures having semantic significance include dates, addresses, phone numbers and names. *Id.* 2492-93; JX-3 at col. 1, lns. 14-16.

WI-Apple1604935

"analyzer server for detecting structures in the data" (claim 1) and "detecting a structure in the data" (claim 15)

As to the first portion of this claim element, *i.e.*, "detecting structures in the data," HTC concedes that Browser, Android Messaging, and HTC Messages detect structures in data. Olsen Tr. 4020-21; Mowry Tr. 2490. The Browser detects e-mail addresses, phone numbers, and postal addresses, and Android Messaging and HTC Messages detect e-mail addresses and phone numbers. Mowry Tr. 2496, 2499, 2506.

[

]

Android Messaging and HTC Messages [

]

WI-Apple1604936

[

]

"analyzer server for ... linking actions to the detected structures" (claim 1) and "linking at least one action to the detected structure" (claim 15)

With respect to the second portion of this claim element, *i.e.*, "linking actions to the detected structures," Browser, Android Messaging, and HTC Messages infringe these limitations. Mowry Tr. 2515-16.

Generally, Browser and Android Messaging [

]

WI-Apple1604937

[

]

HTC Messages [

]

[

]

[

WI-Apple1604938

]

In other words, [

]

The Browser's analyzer server and method for linking actions includes [

WI-Apple1604939

]

[

]

[

40

]

[

]

[

40 [

]

]

HTC contends that Apple has failed to prove infringement because Apple has implicitly conceded that the pointers described in the '647 patent differ fundamentally from HTC's accused Android devices. HTC Br. at 101-104.

HTC states in its brief that it would apply "Apple's proposed claim constructions for purposes of HTC's non-infringement defense." HTC Br. at 104. But, contrary to this representation, HTC rejects Apple's construction in favor of a new one that implies that "linking" must occur through the use of pointers. *Id.* at 103-104. This new construction—improperly raised in the post-trial "background" section—is rejected as untimely under Ground Rule 4(c) because it was not in HTC's pre-hearing statement.

In any event, the unrebutted testimony shows that the plain meaning and proper construction of "linking" is "associating." Mowry Tr. 2475-77. HTC's new construction seeks to improperly limit the patent to the preferred embodiment. JX-3 at col. 3, lns. 65-67; Staff Br. at 66 ("the patent does not require any specific type of linking . . ."); *Altiris Inc. v. Symantec Corp.*, 318 F.3d 1363, 1369 (Fed. Cir. 2003) (district court wrongly imported limitation from preferred embodiment).[41]

[41] In an attempt to limit the claims to the use of pointers, HTC contends that the patent's reference to "automatically" performing actions is "inconsistent with" the operation of the HTC products. HTC Br. at 102. Per Ground Rule 4(c), HTC has waived this argument by failing to raise it in its pre-hearing statement. Moreover, the word "automatically" is not a part of any claim element and "pointers" are not necessary to enable the performance of a selected action. [

WI-Apple1604942

Moreover, HTC's comparison of its linked actions to pointers is incorrect because HTC misidentifies the infringing actions. As explained by Dr. Mowry, the infringing action subroutines [

] The detected structures are linked to these subroutines, which cause the CPU to perform a sequence of operations on the structures. Mowry Tr. 2520-22, 2533, 2535-36.

Likewise, HTC's focus on (1) the ability of users to install third-party applications like Skype and (2) [

]

HTC further submits that Apple has failed to prove infringement because Apple [

]

This new contention, that the HTC [

] Ground Rule 4(c) requires a party to "set[] forth with particularity" all of its contentions in its pre-hearing statement; "[a]ny contentions not set forth in detail as required herein shall be deemed abandoned or withdrawn." Order No. 2 at 4(c). HTC

]

WI-Apple1604943

has abandoned the argument that [ ] constitute a single action subroutine by failing to raise it in its pre-hearing statement.

HTC attempts to justify this failure by incorrectly claiming that Dr. Mowry changed his opinions at trial.[42] [

]

Substantively, HTC is incorrect that [

] As Dr. Mowry testified, these methods are

[42] HTC incorrectly contends that Dr. Mowry changed his opinion by identifying [

]

WI-Apple1604944

[

]

HTC incorrectly asserts that Dr. Mowry agreed that [

]

HTC contends still further that Apple has failed to prove infringement because

[

]

WI-Apple1604945

At the outset, HTC is precluded from making this new non-infringement argument under Ground Rule 4(c), and in any event, is wrong. Dr. Mowry's unrebutted testimony proved that the HTC products link actions before the user's selection of that action. *E.g.* Mowry Tr. 2522-23.

With respect to Browser and Android Messaging, [

] HTC's argument is based on the incorrect premise that Dr. Mowry [

] HTC fails to provide a supporting citation for this statement. In fact, Dr. Mowry, Google witness David Sparks, and Dr. Olsen all testified that [

]

HTC's citations to the transcript are improper because they all refer to the different linking mechanism of HTC Messages, not Browser or Android Messaging. HTC Br. at 112-113; Mowry Tr. 2680:3-7. In HTC Messages, [

WI-Apple1604946