UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF WISCONSIN

APPLE INC. and NeXT SOFTWARE INC. (f/k/a NeXT COMPUTER, INC.), Plaintiffs, v. MOTOROLA, INC. and MOTOROLA MOBILITY, INC. Defendants.	)	Case No. 10-cv-662-bbc **JURY TRIAL DEMANDED**

MEMORANDUM IN SUPPORT OF MOTOROLA'S MOTION FOR CLAIM CONSTRUCTION AND A HEARING

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Page

Cases

Statutes

Defendants Motorola Solutions, Inc. (f/k/a Motorola, Inc.) and Motorola Mobility, Inc. (collectively, "Motorola") respectfully submit this brief in support of their Motion For Claim Construction And A Hearing. This brief addresses the proper construction of disputed terms in the patents asserted by Motorola and by Plaintiffs Apple Inc. and NeXT Software Inc. (collectively, "Plaintiffs" or "Apple").

I. INTRODUCTION

The modern wireless communications industry owes its existence in large part to decades of research and development efforts undertaken by Motorola. Founded in Chicago in 1928, Motorola has been, and remains, a global leader in radio communications. It was a Motorola radio that Neil Armstrong used in 1969 to transmit from the Moon the famous words: "That's one small step for man, one giant leap for mankind." Subsequently, in 1973, Motorola demonstrated the first-ever prototype of a radiotelephone cellular handset, and 10 years and more than $100 million in research and development later, Motorola introduced the world's first portable cellular phone, the Motorola DynaTAC, in 1983.

Motorola's wireless communications innovation continues to this day. In 1999, Motorola introduced the first cellular phone featuring an Internet browser, email, and two-way messaging. In 2002, Motorola deployed its first 3G nationwide network in Japan. In 2006, it introduced the MING smartphone to the Asian market – a touch screen phone capable of recognizing more than 10,000 handwritten Chinese characters. And, in 2007, Motorola demonstrated in Chicago the world's first WiMAX mobile network, which enabled mobile devices to have high-speed Internet access even as they move through a city, seamlessly switching connections from one cellular tower to another. Later that year, Motorola became a founding member of the Open Handset Alliance, sponsor of the Android open source mobile phone platform.

In June, 2007 – more than 30 years after Motorola introduced the world's first cellular handset – Apple released its first cellular phone product, the iPhone. Apple's cellular phone products have been successful and profitable. Nonetheless, those products would not exist without technology developed and patented by Motorola; technology which Apple has yet to license.

In this case, both Apple and Motorola assert patents against the other's portable communications products. But, as the claim construction disputes addressed below reveal, Apple's patents generally relate to software, not wireless communications. To the extent they are found valid, Apple's patents generally cover aspects of computer software that were relevant to the Macintosh computers that were Apple's principal products for decades. In order to assert those patents against modern cellular phones, Apple forces contorted claim construction positions divorced from the intrinsic record. In some cases, Apple's positions directly contradict statements made by applicants in the specification or prosecution history, and positions previously advanced by Apple in an ITC investigation involving some of the same patents. As demonstrated below, each of Apple's attempts to retroactively rewrite its claims to target Motorola's products are unsupported by the intrinsic record and should be rejected.

Apple's approach to the construction of terms in Motorola's patents is similarly flawed. The terms identified by Apple simply do not require construction. Motorola asserts straightforward patent claims covering aspects of wireless technology that Apple undeniably practices. By contrast, Apple's claim construction proposals represent efforts to manufacture non-infringement arguments out of whole cloth. Thus, they should be rejected.

II. LEGAL STANDARDS

Patent claims are construed from the perspective of a person of ordinary skill in the art. Accordingly, "the ordinary and customary meaning of a claim term is the meaning that the term

would have to a person of ordinary skill in the art in question at the time of the invention . . . ." *Phillips v. AWH Corp.*, 415 F.3d 1303, 1313 (Fed. Cir. 2005). The patent specification and prosecution history provide "intrinsic" evidence of the meaning of disputed claim terms. The specification is considered "the single best guide to the meaning of a disputed term." *Id.* at 1315 (quoting *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1582 (Fed. Cir. 1996)). Intrinsic evidence is particularly important because "the person of ordinary skill in the art is deemed to read the claim term not only in the context of the particular claim in which the disputed term appears, but in the context of the entire patent, including the specification." *Id.* at 1313. Accordingly, "[t]he construction that stays true to the claim language and most naturally aligns with the patent's description of the invention will be, in the end, the correct construction." *Id.* at 1316 (quoting *Renishaw PLC v. Marposs Societa' per Azioni*, 158 F.3d 1243, 1250 (Fed. Cir. 1998)). For example, the specification may reveal that the patentee defined a claim term, in which case that definition governs. *See Phillips*, 415 F.3d at 1316; *CCS Fitness, Inc. v. Brunswick Corp.*, 288 F.3d 1359, 1366 (Fed. Cir. 2002). Further, a claim term may be redefined by implication and "the written description of the preferred embodiments 'can provide guidance as to the meaning of the claims, thereby dictating the manner in which the claims are to be construed, even if the guidance is not provided in explicit definitional format.'" *Bell Atl. Network Servs., Inc. v. Covad Commc'ns Group, Inc.*, 262 F.3d 1258, 1268 (Fed. Cir. 2001) (quoting *SciMed Life Sys., Inc. v. Advanced Cardiovascular Sys., Inc.*, 242 F.3d 1337, 1344 (Fed. Cir. 2001)).

The Court may consider certain extrinsic evidence, such as dictionaries or expert testimony, to provide background on the technology at issue, to explain how an invention works, or to explain the meaning of a term as it would be understood by a person of ordinary skill in the

art at the time of the invention. *See Phillips*, 415 F.3d at 1317-18. However, such extrinsic evidence cannot be used to contradict the intrinsic evidence. *See Finisar Corp. v. DirecTV Grp., Inc.*, 523 F.3d 1323, 1328 (Fed. Cir. 2008). "[C]onclusory, unsupported assertions by experts as to the definition of a claim term are not useful to a court." *Phillips*, 415 F.3d at 1318. "Similarly, 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 (quoting *Key Pharms. v. Hercon Labs. Corp.*, 161 F.3d 709, 716 (Fed. Cir. 1998)).

Claim limitations using "means for" language are presumed to be means-plus-function claims within the scope of 35 U.S.C. 112, ¶ 6. Such means-plus-function claims must be construed to include the corresponding structure described in the specification and equivalents thereof. *See* 35 U.S.C. § 112, ¶ 6; *see also B. Braun Med. Inc. v. Abbott Labs.*, 124 F.3d 1419, 1424 (Fed. Cir. 1997). Accordingly, when faced with means-plus-function limitations, courts "must turn to the written description of the patent to find the structure that corresponds to the means recited in the ... limitation." *B. Braun Med.*, 124 F.3d at 1424. "[S]tructure disclosed in the specification is 'corresponding' structure only if the specification or prosecution history clearly links or associates that structure to the function recited in the claim." *Id*. Moreover, the focus of the "corresponding structure" inquiry is not merely whether a structure is capable of performing the recited function, but rather whether the corresponding structure is "clearly link[ed] or associat[ed] . . . to the function recited in the claims." *Id*. "If one fails to set forth an adequate disclosure with respect to the corresponding structure of a means-plus-function limitation, then one has failed to 'particularly point[ ] out and distinctly claim[ ] the subject matter which the applicant regards as his invention,' as required by § 112, P2." *Freeman v.*

Gerber Prods. Co., 120 F. App'x 322, 325 (Fed. Cir. 2005) (quoting *In re Donaldson Co.*, 16 F.3d 1189, 1195 (Fed. Cir. 1994) (en banc)).

III. CONSTRUCTION OF TERMS IN APPLE'S PATENTS

A. United States Patent Number 6,275,983

1. Introduction Of The Disputed Phrase "During Runtime"

The parties dispute the construction of the phrase "during runtime" found in U.S. Patent 6,275,983 (the "'983 patent"). The parties' proposed constructions for that phrase are set forth below:

Disputed Claim Phrase	Motorola's Proposed Construction	Apple Proposed Construction
during runtime	during the execution of the object-oriented application	while running or executing

Apple asserts claims 1, 7, 16, and 22 from the '983 Patent against certain of Motorola's Android-based products. The phrase "during runtime" appears in each of the asserted claims. Claim 1, reproduced below, is representative of the asserted claims. This claim includes three elements and associated limitations that must be performed "during runtime":

> 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 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.

(Ex. 1, '983 Patent, at 37:51-38:7 (emphasis added).)

Motorola's proposed construction of "during runtime" is consistent with the intrinsic evidence and, in particular, the specification's description of the invention. Apple's construction is incomplete in that it fails to identify what is running or executing. Furthermore, Apple's proposed construction should be rejected because Apple made the same proposal in a prior case in which it also asserted the '983 Patent against products based on Android, but later conceded that the proper construction of the term is virtually the same as the one proposed here by Motorola.[1] In Investigation No. 337-TA-710 of the International Trade Commission ("ITC"), Apple asserted the '983 Patent against Android based smartphones manufactured by HTC.[2] In that Investigation, Apple made the same proposal it does here, but Respondent HTC and the Commission Staff both determined that the proper construction of "during runtime" is virtually identical to the one proposed by Motorola. Subsequently, Apple withdrew its proposed claim construction of this term, and adopted the construction proposed by HTC and the Commission Staff before the ALJ reached a determination. (*See* Ex. 31, April 4, 2011 Joint Motion To Amend The Joint List of Undisputed Claim Terms With Agreed Constructions, Apple's Corrected Proposed Claim Construction Chart, and HTC's Proposed Claim Constructions, at

[1] The parties in *Certain Personal Data And Mobile Communications Devices and Related Software*, USITC Inv. No. 337-TA-710 agreed that the construction for "during runtime" was "during the running or execution of the object oriented application." (*See* Ex. 31, Apr. 4, 2011 Joint Motion To Amend The Joint List of Undisputed Claim Terms With Agreed Constructions, Apple's Corrected Proposed Claim Construction Chart, and HTC's Proposed Claim Constructions, at Appx. A)

[2] In an ITC investigation there are three parties, the Complainant (plaintiff), the Respondent (defendant), and the ITC Commission Staff: an impartial group of attorneys who reach their own conclusions regarding the disputed issues and make recommendations to the Administrative Law Judge ("ALJ") that decides the matter.

Appx. A.) Now, however, Apple seeks to resurrect its flawed claim construction proposal. The Court should reject Apple's flip-flopping and adopt Motorola's construction, which is supported by the intrinsic evidence.

2. Description Of The '983 Patent

The '983 Patent purportedly addresses the problem of enabling object oriented applications to access the services of a procedural operating system in an object oriented manner.[3] When a programmer creates a program to run on a personal computer, the programmer can take advantage of services provided by the operating system. Operating system services typically include tasks such as reading and writing data or processing input from a keyboard. (*See* Ex. 1 at 3:5-10.) The '983 Patent attempts to deal with the problem encountered by programmers of object-oriented applications that need to invoke operating system services, where the operating system does not provide a native object-oriented interface. (*See id*. at 3:10-30.) The first part of the '983 Patent's solution is to create objects and methods for the operating system services. (*See id*. at 3:31-42.)

The second part of the '983 Patent's solution is to efficiently use memory. The instructions in a program or application are executed by a central processing unit ("CPU"). The instructions need to be stored in memory, where the CPU can access them. The act of copying program instructions to memory, where the CPU can access them, is sometimes called "loading." Ordinarily, delaying the decision of which function to call until the application is running results in wasted memory resources, because every function that the application could call would be pre-loaded into memory. The '983 Patent's stated invention avoids the potential waste of

[3] In the early 1990s, the operating systems available for personal computers were mostly "procedural." A "procedural" operating system does not provide a native interface for object-oriented programming.

memory by waiting until the application is running and needs a function before loading the appropriate code into memory. In other words, the code is copied into executable program memory during runtime. (*See id.*, at 6:53-7:1, 8:61-9:39.)

3. The Court Should Construe The Term "During Runtime" Because It Will Confirm That Motorola's Accused Products Do Not Infringe The Asserted Claims of the '983 Patent

In its infringement contentions, Apple points to Android's Dalvik Virtual Machine as the "means for making determinations during runtime execution if object-oriented methods to be invoked are present in executable program memory." (Ex. 19, Apple's '983 Patent Infringement Contentions Chart, at p. 4.) According to Apple, a method called "ClassLoader.loadClass()" is used in Motorola's products to determine whether object-oriented methods requiring native system services are in memory and if not, to load them. But Motorola's products do not load methods requiring native system services during runtime, *i.e.*, while the application invoking the methods is executing. Instead, Motorola's Android-based products load all classes containing methods requiring native system services when the system starts up, prior to executing any application. These methods are loaded when the system starts up because the system services provided by Motorola's Android products are likely to be used frequently by applications so, unlike in the '983 Patent, there is no advantage to postponing the loading of the corresponding code.

Accordingly, if "during runtime" is properly construed to mean "during the execution of the object-oriented application," this construction will confirm that the accused Motorola Android products do not infringe the asserted claims of the '983 Patent.

4. The Court Should Adopt Motorola's Proposed Construction

Motorola's proposed construction is consistent with the intrinsic evidence, including the specification and the prosecution history. The specification is "the single best guide to the

meaning of a disputed term." *Phillips*, 415 F.3d at 1315. In the '983 specification the phrase "during runtime" always refers to "run-time execution ***of the application*** in a computer." (*See, e.g.*, Ex. 1 at 4:10-14, 5:4-9, 6:26-31, 6:54-60, 7:32-35, 7:67-8:5 and 37:26-33 (emphasis added).) Nothing else is disclosed, and nothing else would make sense. The purpose of the invention is to allow an object-oriented application to access services of a procedural operating system. (*See id*. at 5:1-10.) Accordingly, the invention is only needed when an application is actually running and in need of a native system service. This is confirmed by the '983 Patent's description of "the present invention":

> The operation of the ***present invention*** shall now be generally described with reference to FIG. 2, which illustrates a high-level operational flow chart 202 of the ***present invention***. The ***present invention*** is described ***in the context of executing the object-oriented application*** 130A on the computer platform 102. In step 206, which is the first substantive step of the flow chart 202, an object-oriented statement which accesses a service provided by the operating system 114 is located in the application 130A ***during the execution of the application*** 130A on the computer platform 102. The object-oriented statement is defined by the object-oriented class library 402. For example, the object-oriented statement may reference a method defined by one of the classes of the class library 402.

(*Id.* at 7:61-8:8 (emphasis added).) The present invention described above occurs when, during execution of an application, an object-oriented statement is encountered which requires a service of the operating system. *See Edwards Lifesciences LLC v. Cook Inc.*, 582 F.3d 1322, 1330 (Fed. Cir. 2009) (explaining that specification's frequent description of an "intraluminal graft" as "the present invention" or "this invention" indicates an intent to limit invention to intraluminal devices). The steps of "determining" and "loading" must occur during runtime, *i.e.*, after the application has started to execute. Only then does it make sense to determine whether the code required by the application is already loaded and, if not, loading it into the task address space (memory):

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

(Ex. 1 at 8:55-59 (emphasis added).)

> 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 completion of step 314, the library server has copied the requested program logic to the task address space.

(*Id.* at 9:34-39 (emphasis added).) Once the required method code is loaded into executable program memory (task address space) the code is executed "on behalf of the application 130A." (*Id.* at 9:60-61.) Accordingly, the steps of the claimed invention can only occur while the application is executing.

In fact, during prosecution of the '983 Patent the Applicant highlighted this aspect of the invention in distinguishing prior art. In responding to a PTO rejection, the Applicant distinguished a prior art reference (Schmidt) by stating that in "the claimed invention" the actual code to be executed is not even determined until the application is actually running:

> 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.
>
> 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 [sic] library to use at development time, ***not run time***.

(Ex. 15, Prosecution History of the '983 Patent, Amendment C dated June 28, 2000 at p. 5 (emphasis added).)

As the Federal Circuit has stated, the prosecution history helps "show what a person of skill in the art would have understood disputed claim language to mean." *Phillips*, 415 F.3d at 1314 (quoting *Innova/Pure Water, Inc. v. Safari Water Filtration Sys., Inc.*, 381 F.3d 1111, 1116

(Fed. Cir. 2004). Here, it confirms that "during runtime" means "during the execution of the object-oriented application," as proposed by Motorola.

Apple's proposed construction of "during runtime" is: "while running or executing," a flawed proposal because it fails to tell the trier of fact what is "running or executing." Moreover, Apple's proposed construction is redundant to language already in the claims. Claim 1, for example, recites "means for making determinations during runtime execution…." The claim's use of the word "execution" already conveys that something is "running or executing." Furthermore, Apple's position here contradicts the position it took with respect to this claim term before the ITC, where it conceded that Motorola's proposed construction is correct. (*See* Ex. 33 at Appx. A.) In sum, Apple's proposal is an attempt to broaden the scope of the claims beyond the invention disclosed in the specification and prosecution history; thus, it should be rejected.

B. United States Patent Number 5,969,705

1. Introduction Of The Disputed Phrase "Events For Controlling Said User Interface Display"

The parties dispute the construction of the phrase "events for controlling said user interface display" found in U.S. Patent 5,969,705 (the "'705 Patent").

Disputed Claim Phrase	Motorola's Proposed Construction	Apple's Proposed Construction
events for controlling said user interface display	"events that direct the foreground process to change the user interface display in a specified way"	"information regarding the status of the operations performed by the background process for the foreground process that is used to control the user interface display"

Apple asserts claim 1 of the '705 Patent against certain of Motorola's Android-based products. Claim 1 – the patent's only claim – provides:

> 1. In a computer system comprising a processor, a display, a memory, a user input device, a ***first process*** operative in the computer system, a ***second process*** operative in the computer system as a foreground process and a user interface on said computer system display under the control of the second process, a ***method***

for the first process to perform operations for the second process and control a content of the user interface on said computer system display, said content under control of the foreground second process operative in said computer system, said first process controlling the content to display information regarding the operations performed by the first process for the second process, said method comprising the following steps:

a. installing an event handling process as part of said second process, said event handling process when said second process is operative in said computer system, servicing ***events generated by the first process for controlling said user interface display*** under control of said second process;

b. said second process initiating said first process to perform operations for said second process, said second process operative in the foreground and said first process operative in the background;

d. said first process generating ***events for controlling said user interface display*** while the second process remains as a foreground process and the first process is a background process, said events providing information regarding the operations performed by said first process for the second process; and

e. said event handling process receiving events generated by said first process, said event handling process updating said user interface on said computer system display according to said events generated by said first process, while said first process remains in the background, and received by said event handling process.

(Ex. 2, '705 Patent, at 12:30-65 (emphasis added).)

Motorola's proposed construction for the phrase "events for controlling said user interface display" is consistent with the intrinsic evidence and in particular, the relevant claim language. Apple's construction is overly broad and contradicted by the '705 specification. Accordingly, the Court should adopt Motorola's proposed construction.

2. Description Of The '705 Patent

The '705 Patent describes a method to enable a background or inactive application program to control the user interface display which is otherwise under control of a foreground or active program.

In the early 1990s, first-generation multitasking operating systems for desktop personal computers, such as Windows 3.0 and Apple System 7.0, gained popularity. "Multitasking" refers to the ability of a computer to execute more than one program at the same time. A program can be thought of as a list of instructions which are executed by a processor. Normally, a computer with a single processor is only able to execute one program at a time. Computer multitasking became possible by employing techniques to make programs share a single processor by taking turns executing their respective instructions. The multitasking technique employed by Windows 3.0 and the Apple System 7.0 operating system is called "cooperative" multitasking. In cooperative multitasking, a single program has control of the processor at any one time, but the program yields control back to the operating system whenever it can – voluntarily – so that other programs can use it. The program in control of the processor at any one time is said to be running in the "foreground," while programs being executed but not in control of the processor at the time are said to be in the "background."

The purported invention of the '705 Patent was intended to solve a problem arising from Apple's implementation of multitasking in its System 7.0 operating system for Macintosh computers. In Apple's system, only the foreground application could control the display. (*See* Ex. 2 at 1:20-32.) The problem arose when a program running in the background wanted to update the display to provide an indication to the user. The program was unable to do so because only the foreground program could update the display. (*See id*. at 1:37-58.)

This problem is illustrated in the specification using the example of an email application. (*See id*. at 4:66-5:11.) The email application is running in the foreground and has exclusive control of the computer display. (*See id*. at 4:65-5:30, Fig. 2.) A user may wish to attach a file to an email message, and adds the file to the list of "enclosures" to the message. As a result, the

email program must copy the file from the location where it is currently stored to a new location designated for outgoing email attachments. The email program, however, does not have the necessary code to copy files. Instead, it makes a request from a program provided by the operating system (Apple System 7.0) called "Finder." Finder runs in the background to copy the file from one location to another. The problem at the time was that the Finder program may have to alert the user of errors, such as the file being unavailable to copy, or the destination having insufficient space. Even if the copy operation can proceed normally, the background program needs to provide the user with updates regarding the status of the copy operation, so that the user knows that the requested operation is being performed.

The '705 Patent purports to solve this problem using the event management system of the Apple System 7.0 operating system. (*See id*. at 5:65-6:30.) Events are occurrences, such as a keystroke or a mouse click, of which certain applications must be informed. For example, the email application needs to be informed when the user presses a key corresponding to a letter, so that it can respond by displaying the letter on the screen as the user drafts an email message. A different application may be in charge of monitoring keyboard inputs. Both applications can communicate through events. The application monitoring the keyboard acts as an event producer; it generates an event when the user presses a key. The word processing application acts as an event consumer; it wants to be notified of key-pressing events. In the '705 Patent, the Finder application becomes an event producer. Whenever the Finder application wants to update the display it generates an event indicating the specific change it wants to make – for example, moving the progress bar from 20% to 50% complete – and reports that event value. The email application, which has control of the display, acts as an event consumer. It receives the display-

control events from the Finder and follows the directions to update the display in the specified way. (*See, e.g., id*. at 7:16-8:39.)

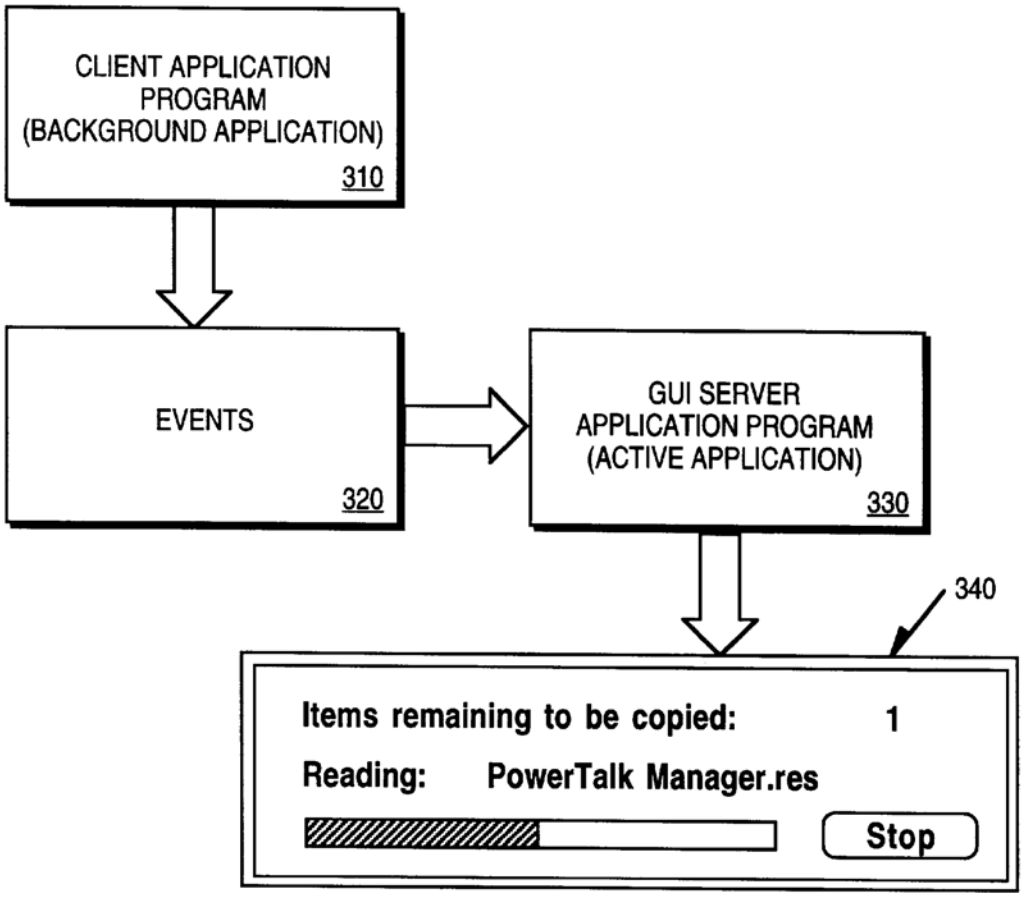


FIG. 3

In the example discussed above, item 310 in Fig. 3 represents the Finder application in charge of the task of copying the file. Item 330 is the email application, which has control of the display (item 340). The Finder application sends events (item 320) to the email application, instructing it to update the display in a specified way. (*See id*. at 7:25-31.) The finder application follows the instructions to update the display accordingly.

3. The Court Should Construe The Phrase "Events For Controlling Said User Interface Display" Because It Will Confirm That Motorola's

Accused Products Do Not Infringe Asserted Claim 1 Of The '705 Patent

In its infringement contentions, Apple accuses Motorola's Android-based products of infringing the asserted claim by running the Google Talk application ("Talk"). (Ex. 20, Apple's '705 Patent Infringement Contentions Chart, at p. 4.) According to Apple, the Talk application operates in the foreground and is in control of the display. Moreover, according to Apple's infringement contentions, the Talk application purportedly uses a second process called the GTalk service to monitor the online status of the user's contacts. Whenever a contact changes status from online to offline, or vice-versa, according to Apple the GTalk service generates an event – the accused "events for controlling said user interface display" – that inform the foreground Talk application of the change. In turn, Apple contends the Talk application updates the display to reflect the new online status of the contact.

Even assuming, *arguendo*, that the Talk application works as described by Apple, Motorola's products do not infringe for multiple reasons, including because they do not generate "events for controlling said user interface display" as specified by the '705 Patent. Accordingly, as properly construed, Motorola's accused products cannot infringe the asserted claim of the '705 Patent.

4. The Court Should Adopt Motorola's Proposed Construction Because It Is In Accord With The Intrinsic Evidence

In construing a disputed term, the Court must consider the surrounding words in the claim. *See ACTV, Inc., v. Walt Disney Co.*, 346 F.3d 1082, 1088 (Fed. Cir. 2003) ("[T]he context of the surrounding words of the claim also must be considered in determining the ordinary and customary meaning of those terms."). Here, the disputed term appears in limitation (d) of the claim, but the preamble states that the claimed method is, among other things, "for the first process to… control a content of the user interface on said computer system display."

Accordingly, one purpose of the method is to ***control***, rather than merely influence the contents of the display. The term itself, "events for controlling…" has a plain and ordinary meaning that the events must include instructions or commands that exert control over the display. This understanding is confirmed by the specification.

The '705 Patent specification provides examples of "events for controlling" the user interface display in a specified way. (*See, e.g.*, Ex. 2 at 7:16-24.) Each example comprises a specific instruction to modify the display. Nothing else is disclosed, and nothing broader can be supported by the specification. For example, the NewCopyWindow event signals that a new window for displaying the status of a copy operation should be displayed. (*See id*. at 7:46-8:38.) Similarly, the DisposeCopyWindow event causes a copy window to be removed from the display. (*See id*. at 8:39-45.) Each event contains a series of attributes directing exactly what will be shown on the display. For example, a NewCopyWindow event includes an attribute setting the maximum value of the progress bar to be displayed on the window, an attribute for the window title, and one for displaying the action currently being performed (*e.g.*, "reading," "writing" or "verifying"). (*See id*. at 7:45-66.) Similarly, the progress bar is changed by a ChangeBar event which specifies the current progress value to be displayed. (*See id*. at 9:1-13.) Accordingly, the "events for controlling" do not merely convey information, they specify changes to the display. In fact, the phrase "in a specified way" in Motorola's proposal comes directly from the specification, which provides:

> Another of the objects of the present invention is to provide a protocol wherein a background application program may direct a foreground application program to ***control its user interface in a specified way***.

(*Id.* at 1:65-2:2 (emphasis added).)

> All of the above events are shown for illustrative purposes only and are for the client application process 310 (*e.g.*, the "Finder" performing the copy operation) alerting server process 330 (*e.g.*, an electronic mail application program) ***to***

> ***change the user interface display in a specified manner***. However other events may also be defined to ***specify other changes to the user interface display***[.]

(*Id.* at 9:54-61 (emphasis added).)

The language describing "the present invention" cannot be dismissed by Apple as referring only to a particular embodiment. *See Trading Techs. Int'l, Inc. v. eSpeed, Inc.*, 595 F.3d 1340, 1353-54 (Fed. Cir. 2010) (describing limitation as part of "the present invention" or "the invention" is strong evidence that claims should be limited); *SanDisk Corp. v. Kingston Tech. Co.*, No. 10-cv-243-bbc, 2011 WL 972507, at *8 (W.D. Wis. Mar. 16, 2011) (applying *Trading Techs.*). Similarly, the language from column 9 confirms that the scope of the patent's sole claim is limited to events which, like those described in the specification, specify changes to the user interface display.

Apple's broad proposal, which would encompass events that do not specify changes to the user interface display, is not supported by the specification. Apple seeks a much broader construction of this term, which would allow any information to constitute an "event for controlling" the display as long as the information is at some point used to affect the contents of the display. Apple's overly broad proposal is unsupported by the intrinsic record and should be rejected.

C. United States Patent Number 5,566,337

1. Introduction Of The Disputed Phrase "Storing Means"

The parties dispute the construction of the means-plus-function element "storing means for storing a specific set of events of which said at least one event consumer is to be informed." This element is found in claims 1 and 6 of U.S. Patent No. 5,566,337 ("the '337 Patent"). The parties' proposed constructions are set forth below.

Disputed Claim Phrase	Motorola's Proposed Construction	Apple's Proposed Construction
storing means for storing a specific set of event of which said at least one event consumer needs to be informed	*Corresponding Function:* storing a specific set of events of which at least one event consumer is to be informed *Corresponding Structure:* Combination, as described in Col. 6:59-65, of: (1) sequential consumer database as shown in Fig. 2 as element 350, in Fig. 5b as part of element 3500, and as described in Col. 7:65-8:5 and 8:36-55; (2) subscription matrix as shown in Figs. 2 and 4 as element 330, in Fig. 5a as element 3300, and as described in Col. 7:61-65 and 8:6-35; (3) event queues as shown in Figs. 2 and 4 as element 320, and as described in Col. 6:59-65 and 7:26-60; and (4) event kind headers as shown in Figs. 2 and 4 as element 331 and as described in Col. 8:61-9:4.	Function: Storing a specific set of events of which said at least one event consumer is to be informed. Structure: Combination of: (1) subscription matrix as shown in Figs. 2 and 4 as element 330, and as described in the '337 patent specification col. 7:61-65 and 8:6-35; (2) event queues as shown in Figs. 2 and 4 as element 320, and as described in the '337 patent specification col. 6:59-65 and 7:26-60; and (3) event kind headers as shown in Figs. 2 and 4 as element 331, and as described in the '337 patent specification, col. 8:61-9:4. OR: (1), (2) and (3) above, and (4) a sequential consumer database as shown in Fig. 2 as element 350 and Fig. 5b, and as described in the '337 patent specification col. 7:65-8:5 and 8:36-55.

Apple asserts claims 1 and 6 of the '337 Patent against certain of Motorola's Android-based products. Claim 6 is dependent on claim 3, which in turn is dependent on claim 1. Accordingly, the construction of the disputed term affects both asserted claims.

The disputed "storing means" is one of three means-plus-function limitations recited in claim 1:

> 1. In a computer including at least one event producer for detecting that an event has occurred in the computer and generating an event and at least one event

consumer which needs to be informed when events occur in the computer, a system for distributing events comprising:

storing means for storing a specific set of events of which said at least one event consumer is to be informed;

event manager control means for receiving the event from the event producer, comparing the received event to the stored set of events, and distributing an appropriate event to an appropriate event consumer; and

distributor means for receiving the event from the control means and directing said control means to distribute an appropriate event to an appropriate event consumer.

(Ex. 3, '337 Patent, at 21:6-19 (emphasis added).)

The parties agree on the proposed function for this means-plus-function limitation, but disagree on the structure. Because Motorola's proposed structure corresponds with the structure set forth in the '337 specification, it should be adopted by the Court.

2. Description Of The '337 Patent

The '337 Patent describes a method and apparatus for distributing events to applications in an operating system. Events are occurrences, such as a keystroke or a mouse click, of which certain applications must be informed. (*See, e.g.,* Ex. 3 at 1:20-30.) For example, a word processing application needs to be informed when the user presses a key corresponding to a letter, so that it can respond by displaying the letter on the screen. A different application may be in charge of monitoring keyboard inputs. Both applications can communicate through events. For example, the application monitoring the keyboard acts as an "event producer"; it generates an event when the user presses a key. (*See id*. at 4:5-8.) The word processing application acts as an "event consumer"; it receives notification of key-pressing events. (*See id*. at 4:12-16.) The '337 Patent purports to solve alleged problems with existing systems for communicating events.

According to the patent, prior art systems communicated events directly between event producers and event consumers. (*See id*. at 1:53-67.) This was inefficient because, for example,

each producer had to be informed of each consumer interested in its events. In addition to monitoring the events of interest, each producer would keep its own list of interested consumers, modifying it as needed and consuming additional computing resources. Furthermore, if a new consumer was introduced, such as a new type of application, the producer may have required modifications to support it. (*See id*. at 2:1-7.) For example, a new consumer may have required events to be sent to it in a particular format or in a particular order that the producer was unable to deliver without modification. The '337 Patent's purported solution to this problem was a centralized system for distributing events to consumers. Instead of communicating directly with each other, producers and consumers communicate through a central Event Manager.

The '337 Patent discloses a specific structure for the Event Manager. The structure is designed to support new kinds of events and new ways to distribute events without requiring major modifications to the system. The Event Manager system supports different kinds of events, *e.g.*, keystrokes or mouse clicks. (*See id*. at 6:47-52.) For each kind of event that a consumer wishes to receive, it must register with the Event Manager and specify whether it will be a "broadcast" or "sequential" consumer for that kind of event. (*Id.* at 4:20-59; 7:1-13.) Broadcast consumers receive events simultaneously with other consumers. Sequential consumers receive events one at a time; the first consumer is notified of the event, then the second consumer is notified, and so on.

The Event Manager includes an Event Manager control unit and each of the following components (*id.* at 6:59-65):

- Storage data structures, including:
    - a sequential consumer database 350 for storing the events in which each sequential consumer is interested (*id.* at 8:2-5);

- a subscription matrix 330 or 3300, for storing the subscriptions for the events in which broadcast consumers are interested (*id* at 7:61-65);
- event queues 320, one for each broadcast consumer for storing the of events of interest to the corresponding broadcast consumer until they are consumed (*id.* at 7:25-40; 9:51-61);
- and event kind headers which specify each kind of event supported by the system (*id.* at 8:22-34).

- Event manager control unit. This consists of software to manage the storage of events and subscriptions. (*Id.* at 6:59-67.)
- Event distributors. These are programs that distribute appropriate events to appropriate broadcast consumers, according to their subscriptions. There is an event distributor for each kind of event. (*Id.* at 9:5-12.)

The Event Manager of the alleged invention, comprising the structure described above, is illustrated in Fig. 2, reproduced below with the portion identified by Motorola as corresponding to the structure of the disputed "storing means" highlighted in yellow.

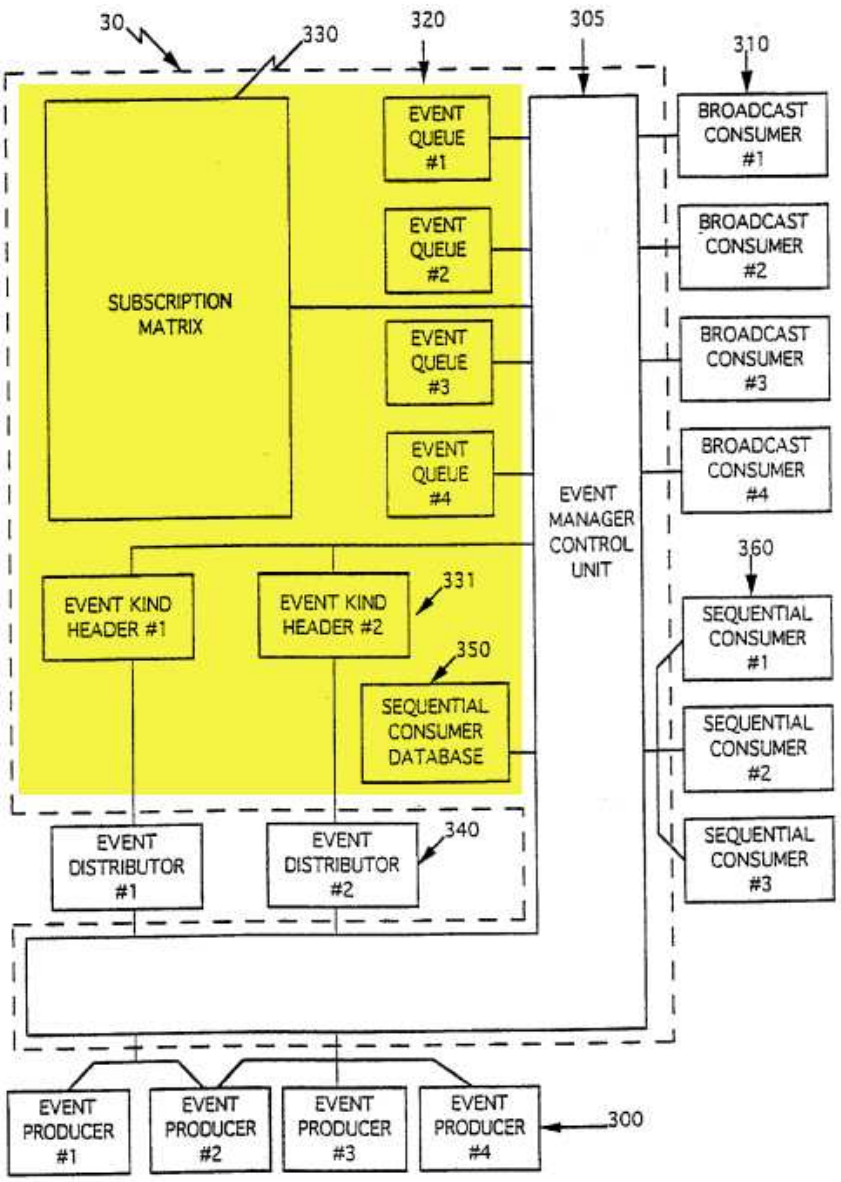


FIG. 2

3. The Court Should Construe The "Storing Means…" Limitation Because It Will Confirm That Motorola's Accused Products Do Not Infringe The Asserted Claims Of The '337 Patent

Apple accuses Motorola's products of infringing claims 1 and 6 of the '337 Patent, even though those products do not include the required "storing means." In its infringement contentions, Apple points to the Activity Manager in Android as being the alleged storing

means.[4] (*See* Ex. 21, Apple's '337 Patent Infringement Contentions Chart.) According to Apple, the Activity Manager includes a data structure, called mRegisteredReceivers, which keeps track of all the receivers registered for broadcasts. In its infringement contentions, however, Apple fails to point to any data structure used for keeping track of sequential consumers. Indeed, no such structure exists in Motorola's Android products.

Accordingly, as properly construed, the accused Motorola Android products cannot infringe the asserted claims of the '337 Patent.

4. The Court Should Adopt Motorola's Proposed Structure For the "Storing Means" Limitation

The parties agree that "storing means" is a means-plus function limitation governed by 35 U.S.C. 112, ¶ 6. The parties also agree on the function of the storing means: "storing a specific set of events of which at least one event consumer is to be informed." The next step in construing a means-plus function limitation is to identify structure in the specification that is clearly linked or associated with the recited function. *See B. Braun Med.*, 124 F.3d at 1424. The structure that Motorola identifies as corresponding to the recited function consists of four elements. These are elements identified in the specification as storing the set of events of which consumers are to be informed, and related information. For example, the elements of the structure corresponding to the storing means are listed in the specification at column 6, lines 59-65, where they are referred to as data structures:

> The event manager 30 includes an event manager control unit 305 and data structures. The data structures include a subscription matrix 330, **a sequential consumer database 350**, a plurality of event queues 320 provided in one-to-one correspondence with the broadcast consumers 310, and a plurality of event kind headers 331 provided in one-to-one correspondence with the event distributors 340.

[4] Apple also accuses the Activity Manager of being the event manager control means and the distributor means recited in the claim.

(Ex. 3 at 6:59-65 (emphasis added).)[5]

Apple identifies two alternative structures. The first structure identified by Apple includes three out of the four elements in Motorola's proposal. The second Apple proposal includes all four elements. In essence, Apple's position is that the fourth element – the sequential consumer database – is optional.

Apple's proposed construction is not consistent with the language of 35 U.S.C. § 112, ¶ 6, which requires that "[a]n 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***." *Id.* (emphasis added). Apple's position that this structure is optional is the same as not including that structure at all. Moreover, Apple's position is contrary to the intrinsic evidence which, as described above, plainly links the structure identified by Motorola – including the sequential consumer database – to the function of storing a specific set of events of which consumers are to be informed.

Accordingly, the Court should adopt the structure proposed by Motorola.

D. United States Patent Number 5,455,599

1. Introduction of the Disputed Phrase "Means for Capturing"

The parties dispute the construction of the means-plus-function element "means for capturing state information and rendering information at the grafport object." This element appears in both of the asserted claims. The parties' proposed constructions are found below.

[5] Indeed, in *Certain Personal Data And Mobile Communications Devices and Related Software*, Inv. No. 337-710, *supra*, the Commission Staff agreed that all four elements identified here by Motorola are part of the structure of the storing means, rejecting the argument that Apple insists on making again in this case.

Surrounding Claim Language	Motorola's Proposed Construction	Apple's Proposed Construction
'599 Patent (claim 15): "means for capturing state information and rendering information at the grafport object"	*Corresponding Function:* capturing state information and rendering information at the grafport object *Corresponding Structure:* Indefinite, there is no structure disclosed in the patent to support the corresponding function	Function: capturing state information and rendering information at the grafport object Structure: graphic port in conjunction with the polymorphic cache. Fig. 1B, col. 2:63-5; col. 9: 56-59.

Apple asserts independent claim 15 and dependent claim 26 of the '599 Patent against certain of Motorola's products. Claim 26 is dependent on claim 15.

The disputed "means for capturing" limitation is one of three means-plus-function elements recited in claim 15:

> 15. An apparatus for graphic processing, comprising:
>
> (a) a processor,
>
> (b) a storage attached to and under the control of the processor;
>
> (c) a graphic device attached to and under the control of the processor;
>
> (d) a modeling layer object in the storage;
>
> (e) a grafport object in the storage;
>
> (f) means for generating calls from the modeling layer object to the grafport object using a predefined set of graphic primitives;
>
> (g) ***means for capturing state information and rendering information at the grafport object***; and
>
> (h) means for passing the state information and the rendering information to a graphic device object for output on the graphic device.

(Ex. 4, '599 Patent, at 19:4-19 (emphasis added).)

The parties agree on the proposed function for this means-plus-function limitation, but disagree on the structure. Indeed, Motorola submits that the '599 specification provides no corresponding structure for the "capturing" function and is consequently indefinite under 35 U.S.C. § 112, ¶¶ 2, 6. *See Encyclopedia Britannica, Inc. v. Alpine Elecs.*, 355 F. App'x 389, 395 (Fed. Cir. 2009).

2. Description Of The '599 Patent

The '599 Patent relates generally to an object-oriented graphic system and method that provides a specific framework for processing graphics in an object-oriented way. (*See* Ex. 4 at 2:40-59.) The '599 Patent teaches that traditional prior art graphic application systems suffered several shortcomings including, for example, incompatibility with other graphic application systems, non-extensibility, and the inability to access the graphic's state (such as color, transfer mode, clip area, etc.) at all times because the graphic's state was available only during a specific drawing operation. (*See id.* at 1:45-2:37.) The '599 Patent purports to solve this problem through the use of a "grafport object," which serves as a special interface for accessing the state of the graphic and facilitates access to the graphic's state by a variety of different graphic devices such as a display, printer or plotter. (*See id.* at 8:52-9:10.)

The '599 Patent identified the inaccessibility of the graphic's state information as the primary problem with traditional graphics applications. In past graphics architectures, a graphic typically stores its state (such as color, transfer mode, clip area, etc.) privately. When asked to draw, the graphic procedurally copies these state variables into a graphic port, where they are accessed by the rendering code. Thus, the graphic's state is available only during this explicit drawing operation. This is not object-oriented, and is a restriction a modern graphic system cannot afford to make. (*See id.* at 8:52-67.)

Although prior art graphics architectures allowed the storage of state variables in a graphic port, according to the '599 Patent, such systems stored state variables "privately," which made these systems non-extensible.

Such non-extensibility was "unacceptable in [the] rapidly changing environment of digital technology" of the time because "dedicated, single-purpose graphic systems" meant that there was no convenient, uniform way to manipulate and use graphics. (*Id.* at 1:48-60.) Therefore, the inventors of the '599 Patent saw a need in modern graphics systems to allow the developer to access state variables "publicly" (*i.e.*, outside the context of any particular draw operation so that the system could handle multiple different graphic applications). (*See id.* at 8:52-64.)

The '599 Patent purports to solve the nonextensibility problem identified in the prior art by providing a customizable, object-oriented framework for the graphic to store its state. It does so through a specially designed "graphic port class," which defines the interface for accessing the state of graphics. (*Id.* at 8:63-67.) According to the '599 Patent, this approach supports "a don't call us, we'll call you" architecture in which clients get access to the graphic state outside the context of any particular function." (*Id.* at 8:59-63.)

The summary of the invention further explains that in addition to the graphic object, the "processor [also] includes an object [(referred to as the 'grafport object' in the claims)] for connecting one or more graphic devices to various objects [(referred to as 'graphic device objects' in the claims)] responsible for tasks such as graphic accelerators, frame buffers, page description languages and vector engines." (*Id.* at 2:52-53.) According to the Patent, a graphic's "state" is "managed at the port level" and includes current transform (*e.g.*, rotation, scale, translation), bundle (*e.g.*, appearance such as frame, fill color, pen thickness, dash patterns and

shading), and clipping region attributes. (*Id.* at 8:26-43.) The state of the graphic is managed at the port level by capturing the transform, bundle, and clipping attributes of the graphic in what is referred to as a "polymorphic cache." (*Id.* at 9:56-59.) The '599 specification, however, provides no details concerning how this state information is captured.

3. The Court Should Construe The Means-Plus-Function Element "Means for Capturing…" Because the '599 Specification Fails to Describe Any Corresponding Structure Rendering The Asserted Claims Indefinite

Apple asserts independent claim 15 and dependent claim 26 against Motorola's accused products. The means-plus-function element "means for capturing state information and rendering information at the grafport object" appears in claims 15 and 26 (in claim 26 by reference to claim 15). If the Court finds that this means-plus-function element is indefinite because there is no structure disclosed in the specification to support the corresponding function, the asserted claims of the '599 Patent will be invalid under 35 U.S.C. § 112, ¶ 2. *See Honeywell Int'l, Inc. v. International Trade Comm'n*, 341 F.3d 1332, 1338 (Fed. Cir. 2003); *Encyclopedia Britannica, Inc.*, 355 F. App'x at 395.

4. The Court Should Adopt Motorola's Proposed Construction

The parties agree that the "means for capturing state information and rendering information at the grafport object" is written in means-plus-function form and should therefore be construed to encompass the structure disclosed in the specification for performing the recited function and its equivalents. Motorola and Apple disagree, however, as to the corresponding structure for the means-plus-function element. While Motorola finds no structure disclosed in the patent to support the corresponding function, Apple believes the corresponding structure to be a "graphic port in conjunction with the polymorphic cache."

For support, Apple cites to a single sentence in the '599 Patent: "The graphic port captures state information including transform, appearance ("bundle"), and clipping into a polymorphic cache 220 that is used across multiple types of devices." (*Id.* at 9:56-59.)[6] Although this description states, in the most superficial terms, that "[t]he graphic port captures state information … into a polymorphic cache," it provides no disclosure of the structure needed to implement the "capturing" function. At best, this description indicates where the state and rendering information are stored, but that language already appears in the claim ("at the grafport object") and cannot likewise serve as the structure for the "capturing" function. Indeed, this sentence does not describe how the state and rendering information is captured, much less the structure for performing the "capturing." This lack of structure renders the claim vague such that one of skill in the art could not perceive the bounds of the invention. *Encyclopedia Britannica, Inc.*, 355 F. App'x at 395.

Claim 15 (and by reference, Claim 26), consist of functional claiming, which well-settled Federal Circuit precedent precludes. *See Blackboard, Inc. v. Desire2Learn, Inc.*, 574 F.3d 1371, 1383 (Fed. Cir. 2009) ("The point of the requirement that the patentee disclose particular structure in the specification and that the scope of the patent claims be limited to that structure and its equivalents is to avoid pure functional claiming." (quoting *Aristocrat Techs. Austl. PTY Ltd. v. Int'l Game Tech.*, 521 F.3d 1328, 1333 (Fed. Cir. 2008)). Thus, claims 15 and 26 are invalid for indefiniteness.[7]

---

[6] Apple also cites to Fig. 1B and the corresponding description in the specification ("Fig. 1B is a hierarchical layout of a graphic port in accordance with a preferred embodiment;"). Fig. 1B, however, provides no disclosure of the structure for capturing state information and rendering information.

[7] Claim 1 of the '599 Patent has already been found invalid for having indefinite corresponding structure for a means-plus-function claim term. Administrative Law Judge Bullock in *Certain Mobile Commc'ns & Computer Devices & Components Thereof*, 337-TA-

E. United States Patent Number 6,424,354

1. Introduction Of The Disputed Term "Connection Information"

The parties dispute the construction of the term "connection information" found in U.S. Patent No. 6,424,354 ("the '354 Patent"). The parties' proposed constructions for "connection information" are set forth below.

Disputed Claim Term	Motorola's Proposed Construction	Apple's Proposed Construction
connection information	"data stored in a connection object that identifies a particular [source/second] object and a method of the [receiver/first] object"	"a representation of the [first/receiver] object's interest in, and an associated object method for, receiving notification of a change to a [second/source] object"

Apple asserts claims 1 and 41 from the '354 Patent against certain of Motorola's Android-based products. The parties dispute the meaning of the term "connection information," which is found in the two asserted claims. Claims 1 and 41 are set forth below:[8]

> 1. A method for operating a computer-implemented event notification system for propagating, among a plurality of objects, events representing changes in the objects, the operating method comprising the steps of:
>
> (a) creating, on behalf of a first object, ***connection information*** representing the first object's interest in, and an associated object method for, receiving notification of a change to a second object;
>
> (b) registering the ***connection information*** with a connection object;

704, in an action between Apple and Nokia, found the corresponding structure of a means-plus-function term indefinite (and thus, claim 1 invalid) for the '599 Patent. (*See* Ex. 30, Order No. 18 entered on July 30, 2010 in 337-TA-704, at 23 ("corresponding structure is found not to be sufficiently disclosed in the specification").)

[8] Claim 1 refers to a "first object" and a "second object," while claim 41 refers to a "receiver object" and a "source object." The parties do not dispute that the first object in claim 1 is equivalent to the receiver object in claim 41, and the second object in claim 1 is equivalent to the source object in claim 41, as reflected in the parties' proposed constructions.

(c) creating an event representing a change in the second object, responsive to the change in the second object; and

(d) notifying the first object of the event by invoking the associated object method for receiving notification registered with the connection object only if the event information corresponds to an interest registered on behalf of the first object.

41. A method for operating a computer-implemented event notification system for propagating, among a plurality of objects, events representing changes in the objects, the operating method comprising the steps of:

(a) creating, on behalf of a receiver object, ***connection information*** representing the receiver object's interest in, and an associated object method for, receiving notification of a change to a source object:

(b) registering the ***connection information*** using a connection object;

(c) creating an event representing a change in the source object, responsive to the change in the source object;

(d) notifying the receiver object of the event by invoking the associated object method for receiving notification registered using the connection object only if the event information corresponds to an interest registered on behalf of the receiver object; and

(e) using the ***connection information*** in the connection object to configure status information to enable the notifying step (d).

(Ex. 5, '354 Patent, at 35:32-50; 39:45-65 (emphasis added).)

Motorola's proposed construction for "connection information" is consistent with the '354 claims, specification, and prosecution history. By contrast, Apple's construction only repeats the claim language. Furthermore, Apple previously asserted claims 1 and 41 of the '354 Patent against Nokia in the International Trade Commission, Investigation No. 337-TA-704 ("the Apple-Nokia ITC Action"). There, the Administrative Law Judge ("ALJ") conducted a claim construction hearing and construed the phrase "***connection information*** representing the first/receiver object's interest in, and an associated object method for, receiving notification of a change to a second/source object" to mean "***information, stored in a connection object***, indicating specific source object events in which a receiver object is interested and the receiver

object method that should receive notification." (*See* Ex. 30, Order No. 18 entered on July 30, 2010 in 337-TA-704, at 53.) In particular, the ALJ addressed the "connection information" aspect of the disputed phrase and found that "connection information" is "stored in a connection object," just as suggested here by Motorola. (*Id.* at 52-53.) Based on the foregoing, Motorola requests that the Court adopt Motorola's proposed construction.

2. Description Of The '354 Patent

The '354 Patent, entitled "Object-Oriented Event Notification System With Listener Registration Of Both Interests And Methods," issued on July 23, 2002. The '354 Patent claims a specific implementation of an object-oriented software mechanism for communicating "change events." (Ex. 5 at 1:13-18.)

Communication of "events" is a well-understood concept in computer programming. Event notification allows different parts of a computer system to communicate with each other, such that one part is alerted of a change "event" in another part. (*See id.* at Abstract.) In the context of object-oriented applications, event notification can be used to alert one or more objects of a change in another object. (*See id.*)

One example of an event notification system is the copy/paste functionality in Microsoft Word. In a Microsoft Word document, the top menu bar contains the "Edit" menu, which in turn contains the document editing items "Cut" and "Copy." The operation of those items is implemented through an event notification system. When no text in the document is selected or highlighted, the "Cut" and "Copy" menu items are grayed out.

On the other hand, when the user highlights some portion of text, which is considered a particular "event," the program detects the event and propagates that change to all the parts of the program that have expressed their interest in such events, including the "Cut" and "Copy" menu

items. Upon receipt of the event notification that text has been highlighted, these two entities become selectable and are no longer grayed out. (*See generally id.* at 2:1-21.)

Thus, event notification allows the change – the highlighting of text – to be conveyed from the document to the menu items interested in events of a specific type. The entities that can trigger certain events can be called "event sources" – in this case the object representing the document – and the entities that receive notification of the event can be called "receivers" since they are intended to receive notification of events. (*See id.* at Abstract.) In the above example, the "receivers" are the objects representing the selectable "Cut" and "Copy" menu items.

The '354 Patent relates to event notification in the context of object-oriented programming techniques and addresses a specific issue – user interface consistency across applications. (*See id.* at 1:21-23, 41-45.) The inventors of the '354 Patent acknowledged that prior operating systems (*e.g.*, Macintosh, Windows) were already addressing user interface consistency issues. (*See id.* at 3:47-59.) Those prior art operating systems included applications that supported cutting and pasting between applications similar to the example discussed above. (*Id.* at 3:53-55.) The '354 Patent describes a specific event notification framework that provides a mechanism for propagating change information between objects. (*See id.* at 11:44-45.) This framework "allows objects to express interest in, and receive notification about objects on which they depend." (*Id.* at 11:46-48.)

3. The Court Should Construe The Term "Connection Information" Because It Will Confirm That Motorola's Accused Products Do Not Infringe The Asserted Claims Of The '354 Patent

Though Apple cites various portions of Android code in its infringement contentions for the '354 Patent, no specific detail is given as to what exactly Apple believes is the "connection information" in the accused Motorola products. (*See* Ex. 23, Apple's '354 Patent Infringement

Contentions Chart.) The reason for this omission is likely due to a misunderstanding by Apple of how Motorola's accused products operate.

First and foremost, the Motorola accused products do not store "connection information" inside a "connection object" (or elsewhere) because Android does not make an object representation of a connection. This is in direct contrast to the invention of the '354 Patent, in which the patentee expressly stated in the prosecution history that the "connection information" is stored in the connection object. (*See* Ex. 16, Prosecution History of the '354 Patent, Amendment B dated August 28, 2001.)

Second, the alleged "connection objects" in the Motorola accused products do not identify a particular source object because Android follows a fundamentally different model of communication.

Third, the alleged "connection objects" in the Motorola accused products do not identify a method of the receiver object. Once again, this is in direct contrast to the invention of the '354 Patent, in which the claims themselves confirm that "connection information" must identify "an associated object method for" the receiver object.[9]

Consequently, Motorola's accused products do not infringe claim 1 and 41 of the '354 patent under the proper construction of the term "connection information."

[9] The 354 prosecution history confirms that "associated object method" indicates "a method of the receiver object." (*See* Section (b)(iii), *infra*.)

4. The Court Should Adopt Motorola's Proposed Construction[10]

Motorola's proposed construction captures the essence of the invention as set forth in the specification and prosecution history. Apple's construction, on the other hand, merely repeats language already found in the claim.

Intrinsic evidence is particularly important because "the person of ordinary skill in the art is deemed to read the claim term not only in the context of the particular claim in which the disputed term appears, but in context of the entire patent, including the specification." *Phillips*, 415 F.3d at 1313. Accordingly, "the construction that stays true to the claim language and most naturally aligns with the patent's description of the invention will be, in the end, the correct construction." *Id.* at 1316 (quoting *Renishaw PLC*, 158 F.3d at 1250).

(a) "Connection Information" Is "Data Stored In A Connection Object"

As explained above, objects have a well-understood meaning in the art. The asserted claims dictate a particular implementation of "connection information," in the form of a connection object, using techniques of object-oriented programming. In the context of the specific object-oriented event-notification system claimed in the '354 Patent, one of ordinary skill in the art would have understood the connection information to be "data stored in a connection object that identifies a particular source object and a method of the receiver object." In other words, it is data that is encapsulated inside a connection object. (Ex. 5 at 35:36-39; Ex. 16 at 8.)

[10] Motorola's discussion of "connection information" refers to the "receiver object" and the "source object" as used in claim 41. As discussed above, those terms are interchangeable with the "first object" and second object" as used in claim 1 (and included in Motorola's proposed construction).

Indeed, the applicant affirmed this position during prosecution. In order to overcome the Examiner's rejection that the claims did not enable one to make and use the claimed invention, the applicant expressly stated that "connection information" was contained in "connection objects":

> The "connection information" is contained within "connection objects"… [which], in turn, have more specialized information about which of the events generated by a source object are of particular interest to each of the receiver objects for which it is responsible.

(Ex. 16 at 8.)

Thus, the applicant understood this term to refer to the data encapsulated inside a connection object representing the links between specific source object events and receiver objects, along with the receiver object method it needs to invoke to dispatch the change event. (*Id.*)

The specification also supports Motorola's construction. The specification describes the flow of information in the claimed event notification system and includes a description of the specific functionality of the connection object:

> For each connection [object] registered with the notifier [object] as interested in the notification [object] … the connection [object] is asked to dispatch the notification. In turn … the connection [object] dispatches the notification [object] to the appropriate method of the notification receiver.

(Ex. 5 at 12:14-18.) In other words, in the event notification system of the patent, the notification object propagates from a source object, to a notifier object, to a connection object, and finally to a receiver object. The connection object dispatches the notification object to the specific receiver object method interested in the change event encapsulated in the notification object. (*See id.* at 11:54-56, 12:17-19.) In order for the connection object to function in accordance with the invention as claimed and described, it must necessarily contain connection information that represents specific source object events in which a receiver object is interested

and the specific receiver object method to invoke in order to dispatch the notification object. (*See id.* at Abstract, 11:53-59.)

The claims further support Motorola's proposed construction. In particular, dependent claims 2-3 and 5-10, all of which depend from claim 1, all recite "the connection information in the connection object." (*Id.* at 35:51 – 36:36.) Dependent claim 2 is illustrated below:

> 2. The operating method of claim 1, wherein the connection object is associated with status information, the operating method further comprising the step of: (b. 1) using ***the connection information in the connection object*** to configure the status information to represent whether the notifying step (d) is activated or inactivated.

(*Id.* at 35:51-57 (emphasis added).)

The ALJ in the Apple-Nokia ITC Action also confirmed that the "connection information" is "stored in the connection object" for the same reasons set forth above. The ALJ in his decision set forth in detail why the intrinsic evidence shows that connection information is stored in a connection object:

> During prosecution, the applicant explicitly stated that the "'connection information' is contained within 'connection objects.'" (08/15/01 Resp. to Office Action at 8 (stating, "'Connection information'" is contained within 'connection objects' …. The connection objects, in turn, have more specialized information about which of the events generated by a source object are of particular interest to each of the receiver objects for which it is responsible.").) The specification provides additional support of Nokia's and Staff's construction for it establishes that the "connection information" resides in the connection object and that the receiver object method is part of the connection information. (*See, e.g.*, '354 patent at 11:14-18 ("For each connection registered with the notifier as interested in the notification, at function block 1860, the connection is asked to dispatch the notification. In turn, at function block 1870, the connection dispatches the notification to the appropriate method of the notification receiver.").) The dependent claims further confirm that the "connection information" is stored in the connection object because dependent claims 2, 3, 5-10, 44-47, and 50 all refer to the "connection information *in the* connection object," regardless of whether that connection information is registered "with" or "using" the connection object. (*Id.* at claims 2, 3, 501-, 44-47, and 50.) [FN 18]
>
> Accordingly, the undersigned construes the term "connection information representing the first/receiver object's interest in, and an associated object method

for, receiving notification of a change to a second/source object" to mean "***information, stored in a connection object, indicating specific source object events in which a receiver object is interested and the receiver object method that should receive notification***."

(Ex. 30 at 52-53.) Thus, an independent tribunal determined that "connection information is stored in a connection object," in accord with Motorola's proposed construction.

(b) "Connection Information" Identifies "A Particular Source Object"

Next, the '354 specification teaches that the connection information (which is stored in a connection object) must identify a particular source object of interest. For example, the Abstract states:

> A notification object is created … descriptive information about a change … [a] receiver object must register with a connection object its 'interest' in receiving notification of changes; ***specifying both the event type and the particular source object of interest*** … the receiver object receives only the events of the specified type for the source objects 'of interest' and no others

(Ex. 5 at Abstract (emphasis added).)

The body of the specification is consistent with the Abstract:

> Because there can be many presentations of a single model active at once, the data can be changed from many sources, including collaborators. Each presentation is responsible for keeping itself up to date with respect to the model data. This is accomplished by registering for notification when all or a portion of a model changes. When a change occurs to data in which the presentation is interested, the presentation receives notification and updates its view accordingly.

(*Id.* at 11:23-31.)

> The framework allows objects to express interest in, and receive notification about changes in objects on which they depend.

(*Id.* at 11:46-48.)

> Notification objects transport descriptive information about a change, and interests describe a specific notification from a notification source object.

(*Id.* at 11:58-59.)

(c) "Connection Information" Identifies "A Method Of The Receiver Object"

The asserted claims recite that the "connection information" represents "an associated object method for receiving[] notification of a change to a source object." For instance, the relevant limitation of claim 41 is set forth below:[11]

> (a) creating, on behalf of a receiver object, ***connection information representing*** the receiver object's interest in, and ***an associated object method for, receiving notification of a change to a source object***:

(Ex. 5 at 39:49-52.)

The applicants confirmed during prosecution that the term "associated object method" means a method of the receiver object. (*See* Ex. 16 at 8.) In response to a section 112 rejection, the applicants stated that:

> ***"an associated object method"*** is "the appropriate method of the notification receiver… at function block 1880, the notification receiver takes the appropriate action" disclosed at page 30, line 32 to page 31, line 1.

(*Id.* (emphasis in original).)

The ALJ in the Apple-Nokia ITC Action confirmed that the "associated object method" is "a method of the receiver object" based on the same prosecution history remarks set forth above. (*See* Ex. 30 at 47-49.)

By replacing the term "associated method object" in claims 1 and 41 with the applicants' understanding from the '354 prosecution history, it confirms that the last part of Motorola's construction is correct: the "connection information" represents a method of the receiver object.

[11] Claim 1 of the '354 Patent includes the same language except that the "first object" is used instead of the "receiver object" and the "second object" is used instead of the "source object."

Accordingly, the intrinsic evidence confirms that the "connection information" is "data stored in a connection object that identifies a particular source object and a method of the receiver object."

Apple's proposed construction, by contrast, merely recites the words within the claims in dispute. Thus, not only does Apple's proposed construction ignore the prosecution and file history of the '354 Patent, it is not a construction at all, but merely a recursive recitation of the same term. Apple's proposed construction will not assist the jury in understanding the asserted claims of the '354 Patent and should be rejected in favor of Motorola's proposed construction, which is supported by the intrinsic evidence.

F. United States Patent Numbers RE39,486 And 5,929,852

1. Introduction Of The Disputed Term "Software Computer Architecture"

The parties dispute the construction of the term "software computer architecture" found in Apple's U.S. Patent Nos. RE39,486 (the "RE '486 Patent") and 5,929,852 (the "'852 Patent") (collectively, the "Network Component Patents").[12] The parties' proposed constructions for this term are set forth below.

Disputed Claim Term	Motorola's Proposed Construction	Apple's Proposed Construction
software component architecture	"an architecture that provides a modular document-based computing arrangement"	"arrangement of software components"

Apple asserts claim 1 from the RE '486 Patent and claims 7 and 11 from the '852 Patent against certain of Motorola's Android-based products. The parties dispute the meaning of the

[12] The Network Component Patents share the same inventors and similar specifications. Indeed, the RE '486 Patent expressly identifies the application which led to the '852 Patent as a "related application." (Ex. 6, RE '486 Patent at 1:29-33.)

term "software component architecture," which is found in each of the asserted claims.[13] Claim 1 from the RE '486 Patent is representative of how the disputed term is used in the asserted claims:

> 1. An extensible and replaceable layered component computing arrangement residing on a computer coupled to a computer network, the layered arrangement comprising:
>
> a ***software component architecture*** layer interfacing with an operating system to control the operations of the computer, the ***software component architecture*** layer defining a plurality of computing components; and
>
> a network component layer for developing network navigation components that provide services directed to the computer network, the network component layer includes application programming interfaces; and
>
> a first class included in the application programming interfaces to construct a first network navigation object that represents different network resources available on the computer network, wherein the network component layer coupled to the ***software component architecture*** layer in integrating relation to facilitate communication among the computing and network navigation components.

(Ex. 6, RE '486 Patent, at 17:2-19 (emphasis added).)

Motorola's proposed construction for "software component architecture" is consistent with the purpose and intent of the Network Component Patents, namely, to provide a modular document-based computing arrangement. (*See id.* at 4:22-64; Ex. 7, '852 Patent, at 3:56 – 4:38.) Apple's construction is nothing more than a re-arrangement of the disputed term along with a substitution of the word "arrangement" for "architecture." Apple's construction is not consistent with the meaning of the term as defined by the intrinsic evidence, and will provide little guidance to the jury. Thus, the Court should adopt Motorola's proposed construction.

2. Description Of The Network Component Patents

The Network Component Patents are directed towards a purported simplification of a user's experience on computer networks by replacing the traditional application-based computing

[13] Asserted claim 11 from the '852 Patent is dependent on asserted claim 7.

environment with a modular document-based computing arrangement. (*See* Ex. 6 at 4:22-25, 4:40-49; Ex. 7 at 3:56-59, 4:6-31.)

A computer typically consists of an operating system and application software that, collectively, control the operations of the computer. (*See* Ex. 6 at 3:53-55; Ex. 7 at 3:20-22.) Each application is independent and task-specific, *e.g.*, a word processor application edits text, a video application plays videos, and a drawing application edits drawings. (*See* Ex. 6 at 3:55-59; Ex. 7 at 3:22-26.) A user must separately invoke different applications to view and manipulate different types of data. (*See* Ex. 6 at 3:59-62; Ex. 7 at 3:26-29.) For example, in an application-based computing environment, a user must identify and launch an application for viewing images, and separately locate and launch a different application to play video files. (*See* Ex. 6 at 2:11-15.)

An alternative to this "application-based approach" is a "software component architecture" that provides a modular "document-based computing" arrangement which eliminates the need to launch multiple separate applications. Document-based computing uses the "software component architecture" to create a "compound" document. (*See* Ex. 6 at 4:22-28; Ex. 7 at 3:56-62.) A "compound" document is a document composed of many different ***modules*** of data – text, drawings, videos, sound, etc. – that share the same file. (Ex. 6 at 4:25-30; Ex. 7 at 3:59-63.) Software component architectures initiate all the software components needed to view and edit a particular document. Thus, an advantage of a compound document-based architecture is that a user only needs to keep track of a ***document*** and not applications. The user no longer identifies and launches one or more applications to view or edit a document; the system will automatically locate and launch all of the appropriate software components once the user selects

a document. (*See* Ex. 6 at 12:29-38; Ex. 7 at 11:22-31.) Thus, in a software component architecture, the user does not deal with applications – only documents.

The Network Component Patents purport to extend document-based computing to network-oriented services. (*See* Ex. 6 at 4:40-44; Ex. 7 at 4:6-10.) A network-oriented system allows actions to take place that enhance the ability of a user to interact with the computer to search for and obtain information available over computer networks such as the Internet. (*See* Ex. 6 at 9:34-39; Ex. 7 at 8:23-28.) Network objects are created to represent particular resources available on a network, as well as common methods for interacting with those resources (*e.g.*, downloading, browsing, etc.). According to the Network Component Patents, the prior art forced users interacting with several Internet networks (such as FTP, HTTP, Telnet, or Gopher) to switch between multiple protocol-specific applications. (*See* Ex. 6 at 1:54-2:15; Ex. 7 at 1:54 – 2:15.) For example, a user would employ an FTP application to obtain information accessible via the FTP protocol; the user would separately use a Telnet client to obtain information accessible via the Telnet protocol. The specification alleges that this need to identify and launch different applications to view different network resources resulted in "a proliferation of applications directed to user activity on the Internet," and the need to separately identify and launch these different applications was "time consuming and disorienting to users." (Ex. 6 at 2:16-24.)

The Network Component Patents expressly acknowledge that the prior art allowed users to take advantage of applications such as the Netscape Navigator browser that were "configured to interact with many of the previously-described protocols, including HTTP, Gopher and FTP." (Ex. 6 at 3:26-29; Ex. 7 at 2:63-66.) But these applications were allegedly "monolithic," not component-based, such that "if a user does not like the way a monolithic application handles

certain protocols, the only recourse is to use another service because the user cannot modify the application to perform the protocol function in a different manner." (Ex. 6 at 5:57-61.)

The patents attempted to address this alleged problem in the prior art by extending the compound-document architecture to network resources. (*See* Ex. 6 at 4:40-44; Ex. 7 at 4:6-10.) This change enabled a non-"monolithic" approach using network components that implement networking functionality for multiple protocols.

The Network Component Patents acknowledge that software component architectures were already in use and well-known in the art at the time the patent was filed, including public offerings by Microsoft (OLE) and by Apple (OpenDoc):

> It should be noted that the network component layer 450 may operate with any existing system-wide component architecture, such as the Object Linking and Embedding (OLE) architecture developed by the Microsoft Corporation; however, in the illustrative embodiment, the component architecture is preferably OpenDoc, the vendor-neutral, open standard for compound documents developed by, among others, Apple Computer, Inc.

(Ex. 6 at 8:41-48; Ex. 7 at 7:49-56.) Thus, the alleged novelty was not a software component architecture itself, but the extension of an existing software component architecture to the networking arena.

3. The Court Should Construe The Term "Software Component Architecture" Because It Will Confirm That Motorola's Accused Products Do Not Infringe The Asserted Claims of the Network Component Patents

Asserted claim 1 of the RE '486 Patent and asserted claim 7 of the '852 Patent both require a "software component architecture layer." Apple provides very little detail in its infringement contentions regarding the type of software component architecture layer it alleges exists in the accused Motorola products. But it appears to conclude that Motorola's Accused Products satisfy the "software component architecture" limitation just because "the Android framework is object-oriented." (Ex. 25, Apple's '852 Patent Infringement Contentions Chart, at

2; *see also* Ex. 24, Apple's RE '486 Patent Infringement Contentions Chart, at 3.) Yet the Network Component Patents require that the "software component architecture" must provide a modular document based computing arrangement. (Ex. 6 at 4:22-25; Ex. 7 at 3:56-59.) That aspect of the invention is unambiguous and repeated multiple times in the patent. Motorola's Android-based Accused Products use the traditional ***application-based*** environment and not document-based arrangement. The Android application-based environment requires users to download and launch individual applications, including applications purchased in the Android Marketplace. (See, *e.g.*, https://market.android.com/apps.) Consequently, as properly construed, Motorola's Accused Products do not infringe the asserted claims of the RE '486 Patent or the '852 Patent.

4. The Court Should Adopt Motorola's Proposed Construction Because It Is In Accord With The Intrinsic Evidence

Motorola's proposed construction for "software component architecture" is consistent with the intrinsic evidence. The specification, which provides "the single best guide to the meaning of a disputed term," repeatedly and consistently explains the role of the "software component architecture" in the invention. *Phillips*, 415 F.3d at 1315 (quoting *Vitronics Corp.*, 90 F.3d at 1582). Here, the specification demonstrates that "software component architecture" refers to an architecture that "provides a modular document-based computing arrangement." (Ex. 6 at 4:22-25; Ex. 7 at 3:56-59.) The Applicants repeatedly emphasized the modular document-based nature of the software component architecture in the specification, as well as in their arguments during prosecution to distinguish their stated invention over prior art allegedly using an application-based approach. (*See, e.g.*, Ex. 17, Prosecution History of the RE '486 Patent, Response to Office Action dated July 7, 2004, at 8.)

The Network Component Patents teach that the "software component architecture" of the alleged invention is directed towards a "document-based computing" arrangement, not an application-based computing environment:

> In contrast to this typical application-based computing environment, ***a software component architecture provides a modular document-based computing arrangement*** using tools such as viewing editors.

(Ex. 6 at 4:22-26; Ex. 7 at 4:56-59 (emphasis added).)

The Network Component Patents claim that document-based computing offers advantages not available in prior art application-based systems:

> ***The key to document-based computing is the compound document*** … [s]everal editors, each designed to handle a particular data type or format, can work on the contents of the document at the same time, ***unlike the application-based computing environment***"

(Ex. 6 at 4:25-28; Ex. 7 at 3:59-67 (emphasis added).)

The Network Component Patents then explicitly state that the "present invention" is an effort to take a "software component architecture" that uses "documents of differing contents" (*i.e.*, compound documents) and extend this design paradigm to "network-oriented services":

> The ***software component architecture*** provides the foundation for assembling ***documents of differing contents*** and ***the present invention is directed*** to a system for ***extending this capability to network-oriented services.***

(Ex. 6 at 4:40-44; Ex. 7 at 4:6-10 (emphasis added).) The Patents further explain that the software component architecture creates a compound document, which is used for modular document-based computing:

> Using tools such as viewing editors, the ***component architecture layer creates a compound document*** composed of data having different types and formats

(Ex. 6 at 8:49-51; Ex. 7 at 7:57-59 (emphasis added).)

> ***[T]he present invention is based on a modular document computing arrangement as provided by an underlying software components architecture***, rather than the typical application-based environment of prior computing systems

(Ex. 6 at 8:6-10 (emphasis added); Ex. 7 at 7:16-20 (emphasis added).)

The Network Component Patents' repeated statements concerning modular document-based computing confirm that Motorola's proposed construction is correct. The specification describes these aspects as "the present invention," not a preferred embodiment. Such aspects of "the present invention" must be reflected in claim construction. *See, e.g., Verizon Servs. Corp. v. Vonage Holdings Corp.*, 503 F.3d 1295, 1308 (Fed. Cir. 2007) ("When a patent thus describes the features of the 'present invention' as a whole, this description limits the scope of the invention."); *Honeywell Int'l, Inc. v. ITT Indus., Inc.*, 452 F.3d 1312, 1318 (Fed. Cir. 2006) (construing a term to include fuel filter because "[o]n at least four occasions, the written description refers to the fuel filter as 'this invention' or 'the present invention'"); *SciMed Life Sys.*, 242 F.3d at 1343 ("[T]he characterization of the coaxial configuration as part of the 'present invention' is strong evidence that the claims should not be read to encompass the opposite structure."). Indeed, as described earlier, the use of modular document-based computing rather than application-based computing is the alleged point of novelty for these patents; it is the design choice that allegedly sets the patents apart from prior art applications.

The prosecution histories also support Motorola's proposed construction. For example, the RE '486 Patent was initially rejected over two pieces of prior art: (1) Reinhardt, Andy, *The Network with Smarts*, BYTE, Oct. 1994 ("Reinhardt") and (2) Lippman, Stanley B., C++ Primer 2nd edition, Addison-Wesley, 1991 ("Lippman"). (*See* Ex. 17, First Office Action dated October 14, 2003 at 4-5.) According to the Examiner, the combination of Reinhardt (which teaches a network navigation object coupled to a computer network) and Lippman (which teaches the use of abstract classes in the context of object-oriented hierarchies) rendered then claim 22 of the RE

'486 Patent obvious.[14] In response, Applicants distinguished their invention over the prior art, asserting:

> The ***claimed invention*** beneficially employs a "component-based" approach to ***browsing and retrieving network-oriented information as opposed to the monolithic application-based approach*** of prior browsing systems.

(Ex. 17, Response to Office Action dated July 7, 2004 at 8.)

By contrast, Apple's proposed construction – "arrangement of software components" – has no meaning and will not assist the jury in understanding the term "software component architecture" in the context of the specification. Motorola's proposed construction is adopted directly from the specification's description of the "invention," is consistent with the purpose behind that invention, and will resolve the issue of infringement for the Network Component Patents. Thus, the Court should adopt Motorola's proposed construction.

G. United States Patent Number 5,946,647

1. Introduction Of The Disputed Phrase "Linking Actions To The Detected Structures"

The parties dispute the construction of the phrase "linking actions to the detected structures." This phrase is found in asserted claims 1 and 8 of U.S. Patent No. 5,946,647 (the "'647 Patent"). The parties' proposed constructions for this phrase are set forth below:

Disputed Claim Phrase	Motorola's Proposed Construction	Apple's Proposed Construction
linking actions to the detected structures	"creating a specified connection between a detected structure and at least one specific computer subroutine which performs operations on the structure"	"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"

[14] Claim 22 was eventually rejected by the Examiner. However, Claim 22 used the same "software component architecture" phrase that is contained in the asserted claims of the Network Component Patents.

Apple asserts claim 1 and 8 from the '647 Patent against certain of Motorola's Android-based products.[15] The parties dispute the meaning of the phrase "linking actions to the detected structures," which is found in both of the asserted claims. Claim 1 from the '647 Patent is representative of how the disputed term is used in the asserted claims:

> 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.

(Ex. 8, '647 Patent, at 7:8-24 (emphasis added).) Asserted claims 1 and 8 require, among other things, an analyzer server for (i) detecting structures in the data; and for (ii) linking actions to the detected structures. Motorola's proposed construction for "linking actions to the detected structures" properly defines the linking function performed by the analyzer server. Apple's proposed construction imprecisely repeats and re-arranges the claim language in an attempt to create an infringement position, but Apple's construction is not consistent with the intrinsic evidence. Accordingly, Motorola's construction should be adopted.

---

[15] Asserted claim 8 from the '647 Patent depends from claim 1. Claim 8 states:

> 8. The system recited in claim 1, wherein the user interface highlights detected structures.

2. Description Of The '647 Patent

The '647 Patent relates generally to a computer system that "identifies structures in computer data, associates candidate actions with each detected structure, enables the selection of an action, and automatically performs the selected action on the identified structure." (Ex. 8 at 2:6-9.)

Structures in data include "recognizable structures that have semantic significance such as phone numbers, e-mail addresses, post-office addresses, zip codes and dates." (*Id.* at 1:14-16.) "[A] user may receive extensive files from word-processing programs and e-mail that contain several of these structures." (*Id.* at 1:16-17.)

An "action" is a computer subroutine that performs operations on a structure. (*Id.* at 2:31-34.) For example, "[a]n action may specify opening another application, loading the identified structure into an appropriate field, and closing the application." (*Id.* at 2:34-36.) An action may also include "storing phone numbers in an electronic phone book, addresses in an electronic address book, appointments on an electronic calendar, and external actions such as returning phone calls, drafting letters, sending facsimile copies and e-mail, and the like." (*Id.* at 2:36-41.)

Specifically, the '647 Patent discloses a "computer system with a central processing unit (CPU), input/output (I/O) means, and a memory that includes a program (the "Program") to identify structures in a document and perform selected computer-based actions on the identified structures." (*Id.* at 2:21-25.) "The [P]rogram includes program subroutines that include an analyzer server, an application program interface, a user interface and an action processor." (*Id.* at 2:25:27.)

The analyzer server is described in the '647 Patent as "program subroutines" within the Program that "receive[] data from a document having recognizable structures," "use[] patterns to

detect the structures," and "upon detection of a structure, link[] actions to the detected structure." (*Id.* at 2:28-31.) These patterns can include any patterns that can be recognized, such as patterns for "text, pictures, graphs, voice, etc." (*Id.* at 2:10-12.)

The user interface comprises "program subroutines" within the Program that identify, present, and enable selection of detected structures by the user. (*Id.* at 2:25-27, 46-49.) Additionally, "[u]pon selection of a detected structure, the user interface presents and enables selection of candidate actions." (*Id.* at 2:49-51.) After the analyzer server detects a particular structure, the user interface enables selection of the detected structure by the user.

3. The Court Should Construe The Phrase "Linking Actions To The Detected Structures" Because It Will Confirm That Motorola's Accused Products Do Not Infringe The Asserted Claims of the '647 Patent

Apple asserts claims 1 and 8 from the '647 Patent against certain of Motorola's Android-based products. The asserted claims require, among other things, an analyzer server for (i) detecting structures in the data; and for (ii) linking actions to the detected structures. Therefore, defining what is meant by "linking actions to the detected structures" is essential to understand the precise scope of the '647 Patent.

In its infringement contentions, Apple asserts that Motorola's accused products link actions to the detected structures in conclusory fashion:

> The '647 Accused Products have an analyzer server for detecting structures in data and for linking actions to the detected structures.
>
> - By way of one example, the Droid X includes Android's "Linkify" functionality, which "take[s] a piece of text and a regular expression and turns all of the regex matches in the text into clickable links. This is particularly useful for matching things like email addresses, Internet URLs, etc. and making them actionable." Exh. I-1 [Android Developer Site at Linkify]. In particular, the matching functionality within Android's "Linkify" engine searches text strings for structures representative of Internet URLs, telephone numbers, email addresses, and map addresses. *Id.*

- By way of an example, when the Droid X receives a message comprising an e-mail address using the IM or MMS applications, ***the Droid X detects the e-mail address information and links actions to it***, such as composing an e-mail to that address in email applications on the device. *See* screenshots below.

(Ex. 26, Apple's '647 Patent Infringement Contentions Chart, at 3 (emphasis added).)

Apple's lack of specificity concerning how the accused products "link" actions to the detected structures is telling: it offers no specificity because the accused products do not link one or more actions to a detected structure as required by the asserted claims. Consequently, as properly construed, Motorola's products do not infringe the asserted claims.

4. The Court Should Adopt Motorola's Proposed Construction

The parties agree that the analyzer server described in the '647 Patent detects structures in the data. The parties also agree that the term "actions" in the disputed phrase refers to a computer subroutine(s) that performs operations on the detected structure. Therefore, the dispute centers on how the analyzer server "links" those actions to a detected structure.

Apple merely repeats and re-arranges the words in the phrase, not defining the terms "linking" or "detected structures" at all, and only defines "actions" to be "computer subroutines that cause the CPU to perform a sequence of operations on the particular structures to which they are linked." Apple through its construction attempts to keep the "linking" process as ambiguous as possible to stretch the scope of the claim for its infringement allegations. Motorola, on the other hand, construes this phrase in a manner that is consistent with the specification; namely, once the analyzer server detects a structure, it links that structure to a specific computer subroutine that performs one or more actions on the structure.

The "Summary of the Invention" repeatedly states that the system associates or "links" the candidate actions with "each detected structure," (Ex. 8 at 2:7-8), or "each identified structure," (*id.* at 2:18-20). The "Summary of the Invention" later defines "[t]he present

invention" as including an analyzer server, where "[u]pon detection of a structure, the analyzer server links actions to the detected structure." (*Id.* at 2:30-31.) A statement in the "Summary of the Invention" section of a patent can support a limiting definition of a claim term, especially where, as here, the statement "describe[s] the invention as a whole, rather than … only [a] preferred embodiment[] . . . ." *C.R. Bard, Inc. v. U.S. Surgical Corp.*, 388 F.3d 858, 864 (Fed. Cir. 2004); *see also Am. Med. Sys., Inc. v. Biolitec, Inc.*, 618 F.3d 1354, 1366 (Fed. Cir. 2010) (The Federal Circuit has "repeatedly held that the use of the words 'the present invention' can be read to limit the invention to what is described as such." (internal quotation marks omitted)).

The "Detailed Description of the Preferred Embodiment" also requires that the detection of a structure is followed by the linking of associated actions. For example:

> The program 165 of the present invention is stored in RAM 170 and causes CPU 120 to identify structures in the data presented by application 167, ***to associate actions with the structures identified in the data***, to enable the user to select a structure and an action, and to automatically perform the selected action on the identified structure.

(Ex. 8 at 3:38-44.)

> Upon detection of a structure, ***analyzer server 220 links actions associated with the responsible pattern to the detected structure, using conventional pointers***.

(*Id.* at 3:65-67.)

> Upon identification of a structure in the text, ***parser 310 links the actions associated with the grammar to the identified structure***. More particularly, parser 310 retrieves from grammar file 320 pointers attached to the grammar and ***attaches the same pointers to the identified structure.*** These pointers direct the system to the associated actions contained in associated actions file 330. Thus, upon selection of the identified structure, user interface 240 can locate the linked actions.

(*Id.* at 4:65-5:5.)

> When analyzer server 220 identifies an address using the "e-mail address" grammar, ***actions*** for sending e-mail to the identified address and putting the identified address in an e-mail address book are ***linked to the address***.

(*Id.* at 5:15-18.)

> Upon detection of a structure based on a particular pattern, ***actions associated with the particular pattern are linked 825 to the detected structure***.

(*Id.* at 5:59-61.)

Furthermore, the specification requires an actual association between the detected structures and the actions, often specifying the use of pointers. (*See, e.g., id.* at 3:65-67, 4:64 – 5:5.) A "pointer" is a term of art referring to an explicit reference to a computer subroutine using its exact location in the computer's memory.

Thus, the specification teaches that there is an explicit link between the detected structure and one or more actions. "The claims of the patent must be read in light of the specification's consistent emphasis on this fundamental feature of the invention." *Praxair, Inc. v. ATMI, Inc.*, 543 F.3d 1306, 1324 (Fed. Cir. 2008); *see also Ormco Corp. v. Align Tech., Inc.*, 498 F.3d 1307, 1313-14 (Fed. Cir. 2007) (construing claims to include limitation where doing so "most naturally aligns with the patent's description of the invention" (internal quotation marks omitted)); *Alloc, Inc. v. International Trade Comm'n*, 342 F.3d 1361, 1370 (Fed. Cir. 2003) (looking to "whether the specification read as a whole suggests that the very character of the invention requires the limitation be a part of every embodiment").

The claim language is also consistent with the specification. Claim 1 recites "an analyzer server for detecting structures in the data, and for linking actions to the detected structures." (Ex. 8 at 7:16-17.) Thus, the claimed "detecting" and the claimed "linking" are performed by the same component, the analyzer server. Claim 1 provides that a different component (the "user interface") enables user selection of the detected structure. (*See id.* at 7:18-19.) This language implies, particularly in view of the rest of the specification, that the claimed "detecting" and "linking" procedures are intimately connected and separate and apart from the "enabling"

process. Thus, the linking must be explicitly determined at the time of detection, rather than allowing for an uncertain or unspecified connection to be resolved by another component.

Apple's proposed construction, on the other hand, diverges from the explicit intent of the specification because it could cover a system that links a set of structures to a set of actions, without any direct correspondence between those structures and associated actions. This is contrary to the teachings of the specification, which shows that for each detected structure, there is one or more corresponding actions. For example, Fig. 7 demonstrates that particular actions (*i.e.*, "Call #," "Put in electronic telephone book") are linked to a particular detected structure (*i.e.*, "(415 555-1234") instead of merely presenting a set of detected structures and associated actions.

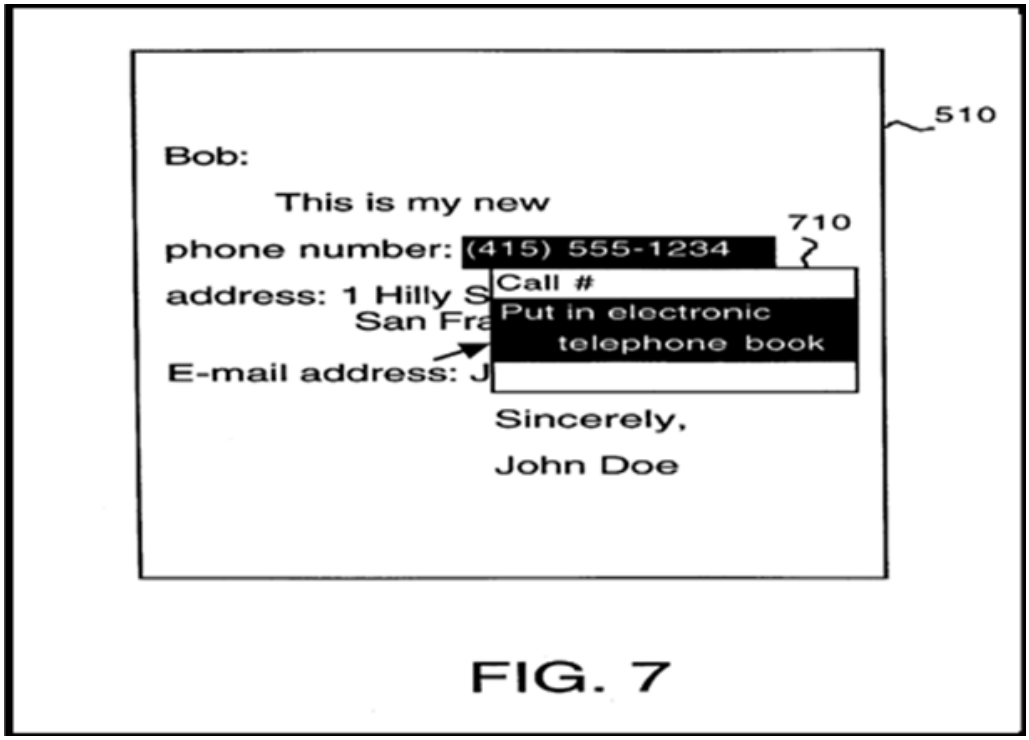


FIG. 7

Because the claims cannot be construed broader than what is supported by the specification, the claimed "linking" phrase should be construed as "creating a specified connection between a detected structure and at least one specific computer subroutine which

performs operations on the structure." *See, e.g., On Demand Mach. Corp. v. Ingram Indus., Inc.*, 442 F.3d 1331, 1340 (Fed. Cir. 2006) ("[T]he claims cannot be of broader scope than the invention that is set forth in the specification."); *Toro Co. v. White Consol. Indus., Inc.*, 199 F.3d 1295, 1299-1302 (Fed. Cir. 1999) (construing "including" in accordance with specification when specification repeatedly described "including" limitation in particular way), *vacated on other grounds,* 266 F.3d 1367 (Fed. Cir. 2001).

H. United States Patent Number 5,481,721

1. Introduction Of The Disputed Term "Dynamic Binding"

The parties dispute the construction of the term "dynamic binding" as used in claims 1 and 5 of U.S. Patent 5,481,721 ("the '721 Patent"). The parties' proposed constructions for "dynamic binding" are set forth below:

Disputed Claim Term	Motorola's Proposed Construction	Apple's Proposed Construction
dynamic binding	"binding messages to actual methods to be invoked, allowing objects of any class to be substituted at run time"	"binding messages to the actual methods to be invoked during runtime"

Apple asserts claims 1 and 5 of the '721 Patent against certain of Motorola's Android-based products. The term "dynamic binding" appears in claim 1, set forth below, and indirectly in claim 5 because it depends from claim 1.

> 1. A method for sending an object oriented programming language based message having ***dynamic binding*** from a first object in a first process to a second object in a second process, said method comprising the steps of:
>
> transmitting, using a first processing means, said object oriented programming language based message to a first proxy in said first process;
>
> using said first proxy and said first processing means, encoding said object oriented programming language based message into an operating system based message at run time;

transmitting said operating system based message to said second process in said second processing means at run time;

decoding, using a second process, said operating system based message into a language based message;

transmitting, using said second processing means, said object oriented programming language based message to said second object in said second process;

executing said object oriented programming language based message by said second object in said second process.

(Ex. 9, '721 Patent, at 71:1-23 (emphasis added).)

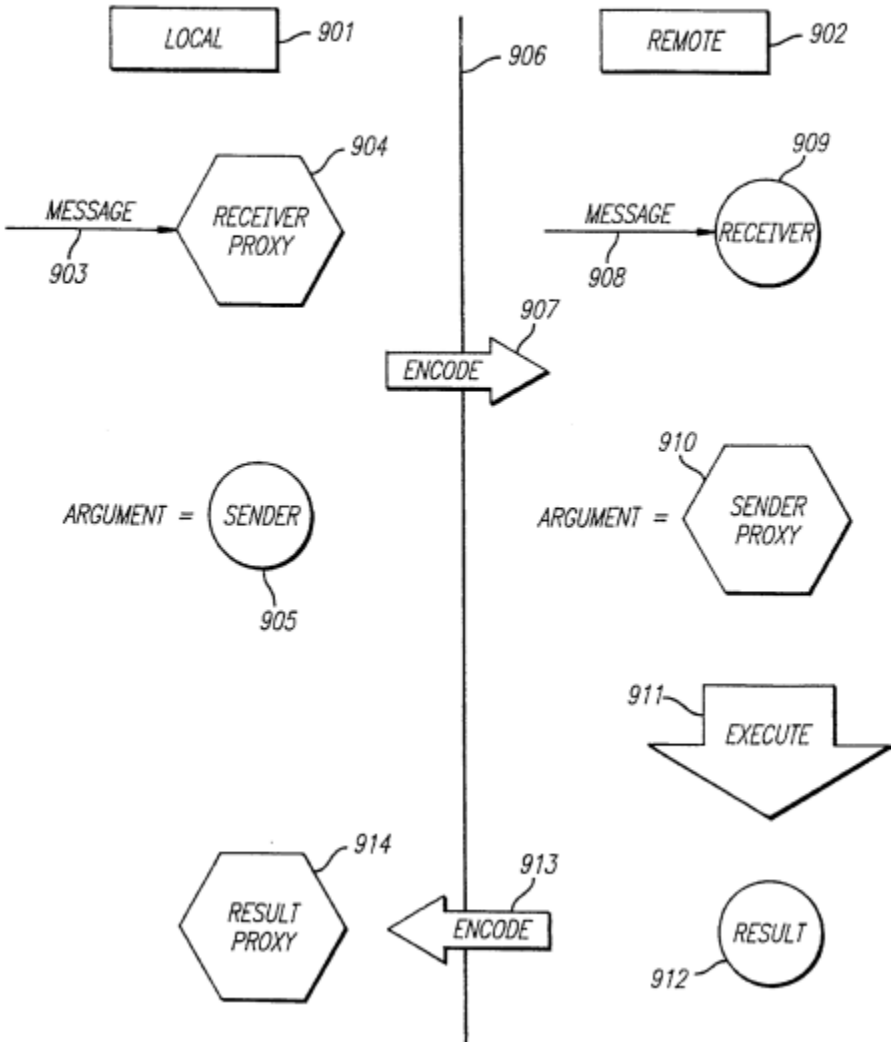
Motorola's proposed construction for "dynamic binding" is consistent with the intrinsic and extrinsic evidence. Apple's construction is overly broad and contradicts the '721 patent specification. Accordingly, the Court should adopt Motorola's proposed construction.

2. Description Of The '721 Patent

The '721 Patent describes a method for passing messages between objects that are located in different processes. (*See* Ex. 9 at 6:53-63.) In object oriented programming a "message" generally refers to a specific set of instructions called a "method" and the data values that must be passed to that method. (*See id.* at 1:33-37.) In the '721 Patent, messages are sent between objects located in two different "processes." The term "process" refers to an executing program and its memory space. More than one process may be running on the computer at any one time. Generally, one process does not have access to another process's memory space. This means, among other things, that an object in a local process cannot communicate directly with an object in a remote process. In order for these objects to communicate, there must be a mechanism for inter-process communication. The stated invention of the '721 Patent allows an object in one process to communicate with an object in another process, using a local "proxy" object that represents a remote object. This local "proxy" object acts as a local receiver for the remote

object. (*Id.* at 6:53-63.) When the proxy receives a message, it encodes the message, which is then transmitted by the operating system to the remote process where the message is decoded and then delivered to the remote object for execution. These steps are shown in Figure 3C (*see id.* at 10:64-11:47):

FIG. 3C



Importantly, the stated invention depends on the ability to use one "proxy" object as the receiver for all messages being sent to a particular remote object. (*See id.* at 10:53-63.) This means that the "proxy" must be able to learn how to encode new messages dynamically, since the

remote object may receive any number of messages. (*See id.* at 12:30-44.) The patent describes this capability as a distinguishing feature over the prior art; unlike "prior art methods" using a "pre-defined set of messages" to communicate between processes (*Id*. at 10:3-7), the '721 patent promised a system with a "dynamic aspect" that "allows one program to 'learn' how to talk to another program" during runtime – and which allows proxy objects to learn new messages "dynamically by asking the receiver [object] how to encode the arguments for each message as it is encountered." (*Id*. at 12:31-45.) As described below, the patent relies on this dynamic functionality to create a mechanism that can forward any message from a local object to a remote object.

3. The Court Should Construe The Term "Dynamic Binding" Because It Will Confirm That Motorola's Accused Products Do Not Infringe The Asserted Claims Of The '721 Patent

Apple's infringement allegations against Motorola's Android products are directed to the use of the Binder framework. (*See* Ex. 27, Apple's '721 Patent Infringement Contentions Chart.) According to Apple, the messages Binder passes between processes in Android are "dynamically bound" because at runtime a certain set of objects from a certain set of classes related to the remote object may be able to receive the message sent to the remote process. But in the asserted claims, and in the '721 patent specification, "dynamic binding" means that that ***any*** object can be substituted for the target object at runtime. Android does not allow this.

The difference between the parties' constructions is not esoteric. It is critical. The '721 patent relies on the ability to send ***any*** message to ***any*** object. Without this capability, the proxy objects would be unable to perform their claimed function: send ***any*** message to ***any*** remote object that the proxy represents. Apple's construction, which does not require this capability, would render the invention described in the specification inoperable.

In Motorola's Android products it is not possible to send a message to an object of ***any*** class. Accordingly, if "dynamic binding" is properly construed to mean "binding messages to actual methods to be invoked, allowing objects of any class to be substituted at run time," it will confirm that the accused Motorola Android products do not infringe the asserted claims of the '721 patent.

4. The Court Should Adopt Motorola's Proposed Construction

The '721 patent specification, which is "the single best guide to the meaning of a disputed term," *see Phillips*, 415 F.3d at 1315, explains the role dynamic binding plays in the invention. In the specification, the Applicants consistently convey that "dynamic binding" refers to the ability to substitute objects of ***any*** class for the target object at runtime. The specification first confirms that any object that implements the method may be substituted for the target object at runtime.

> One feature of objective C is ***"dynamic binding" of messages to the actual methods to be invoked***, depending on the class of the receiver. A programmer writing code in objective C can create code that sends a message "doSomething" to an object. The actual method corresponding to the class of the target object does not need to be determined until the message must be sent. This ***allows objects of any classes that implement[] the doSomething method to be substituted for the target object at run time*** without having to modify the part of the program that sends the message.

(Ex. 9 at 8:29-39 (emphasis added).)

The example above indicates that "dynamic binding" is used in the '721 patent to refer to a form of binding that allows objects of ***any*** classes that implement a method to be substituted for the target object at run time. Thus, if a form of binding does not allow objects of any classes that implement a method to be substituted for the object at runtime, those forms of binding are not "dynamic binding" in the context of the '721 patent. Moreover, the specification also indicates that the stated invention can only be implemented in a system that allows objects of ***any***

class to receive a dynamically bound message, whether or not that object implements the message.

Figure 5, described as an illustration of the "present invention," demonstrates that the steps claimed in the patent can only occur if the proxy object (Object B) receives a message that it does not implement:

FIG. 5

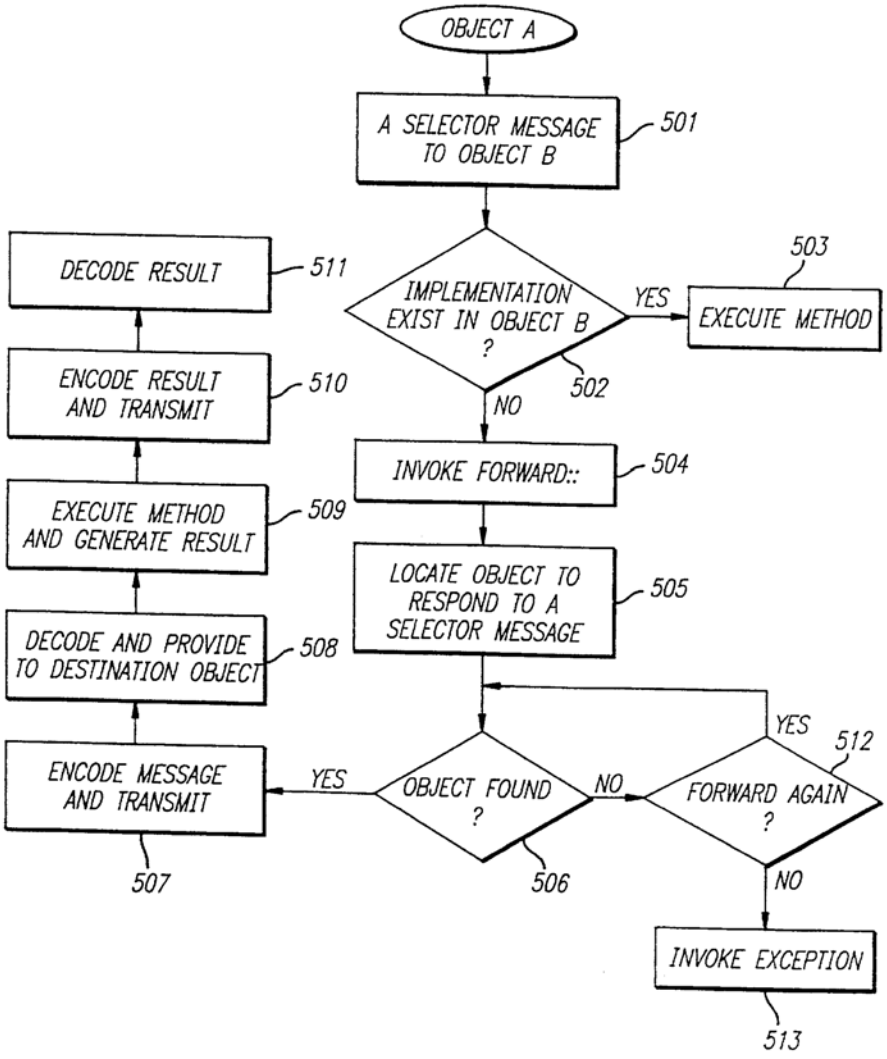


Figure 5 and the corresponding portions of the specification limit the scope of the claims because they describe the "present invention," not a mere embodiment. (Ex. 9 at 14:59-60.) *See Verizon Servs. Corp.*, 503 F.3d at 1308 ("When a patent thus describes the features of the 'present invention' as a whole, this description limits the scope of the invention."). In step 501, object A sends a message to object B. Object B may be a local object that has an implementation of the message and can execute it (step 503). But where the claimed steps are performed, object B is a proxy object; it does not have an implementation of the message. (*See* Ex. 9 at 15:6-9.) In step 502, it is determined whether object B has the required method. If not, object B (a proxy object) invokes the "forward::" method (step 504). The "present invention" illustrated in Figure 5 will not perform the claimed functions of encoding, transmitting, decoding, etc. unless the question posed at Box 502 results in "no" – a condition that can only occur if the dynamically bound message was received by an object that did ***not*** implement the message. Thus, the patent relies on a dynamically bound language that allows objects of ***any*** class to receive ***any*** message at runtime. The patent therefore explicitly teaches, as part of the claimed "present invention," objects (object B in Figure 5) to which ***any*** message can be sent, including both messages that the object does implement (when the answer at Box 502 is "yes") and messages that the object does not implement but instead forwards on to another object (when the answer at Box 502 is "no"). This is possible only with dynamic binding that allows ***any*** message to be sent to the object. The only programming languages in which this is possible are languages that support messages with dynamic binding that allow the program to send ***any*** message to objects of ***any*** class at runtime. Indeed, one of the claimed advantages of the invention is that the substitution of the proxy object for the remote object that it represents is "transparent to the programmer" (Ex. 9 at 10:4-5.) This transparency refers to the fact that it is possible to substitute the proxy

object for the remote object "without having to modify the part of the program that sends the message." (*Id.* at 8:37-38.) This transparency is possible only for messages with dynamic binding that can be sent to objects of ***any*** class at runtime.

Motorola's proposal is further supported by a publication issued by NeXT Computer, assignee of the '721 Patent and Apple's predecessor in interest, that describes "dynamic binding." (*See* Ex. 29, NeXTSTEP Object-Oriented Programming And The Objective C Language, at pp. 20-21.) The publication distinguishes "dynamic binding" from "late binding," which places constraints on the methods that a program may call. (*Id*. at 21.) In a system that practices late binding, a message cannot be sent to an object that does not implement the required method. In contrast, NeXT described dynamic binding as an "unconstrained" form of binding that allows any message to be sent to any method. (*Id*.)

> Some object-oriented programming languages (notably C++) require a message receiver to be statically typed in source code, but don't require the type to be exact. An object can be typed to its own class or to any class that it inherits from.
>
> The compiler therefore can't tell whether the message receiver is an instance of the class specified in the type declaration, an instance of a subclass, or an instance of some more distantly derived class. Since it doesn't know the exact class of the receiver, it can't know which version of the method named in the message to invoke.
>
> In this circumstance, the choice is between treating the receiver as if it were an instance of the specified class and simply bind the method defined for that class to the message, or waiting until run time to resolve the situation. In C++, the decision is postponed to run time for methods (member functions) that are declared virtual.
>
> This is sometimes referred to as 'late binding' rather than 'dynamic binding.' While 'dynamic' in the sense that it happens at run time, it carries with it strict compile time type constraints. As discussed here (and implemented in Objective C), ***'dynamic binding' is unconstrained***.

(Ex. 29 at 21 (emphasis added).)

The "strict compile time type constraints" that are part of late binding are incompatible with the "dynamic binding" disclosed as a required element of the invention, both at col. 8, ll. 29-39 and in Figure 5. This required element is possible only with messages that have unconstrained dynamic binding as disclosed in the NeXT publication quoted above. Apple's proposed construction of dynamic binding would encompass "late binding," meaning a message cannot be sent to an object that does not implement the required method. As described above, the '721 Patent specification and prosecution history do not support and in fact directly contradict such a broad construction. Accordingly, the Court should adopt Motorola's proposed construction.

I. United States Patent Number 6,493,002

1. Introduction Of The Disputed Term "Programming Module"

The parties dispute the construction of the term "programming module" found in the asserted claims of U.S. Patent No. 6,493,002 ("the '002 Patent"). The parties' proposed constructions for "programming module" are set forth below:

Disputed Claim Term	Motorola's Proposed Construction	Apple's Proposed Construction
programming module	"an individual program added to the control strip that provides a specific status or control function"	"code that performs a function"

Apple asserts claims 1, 21, and 46 of the '002 Patent against certain of Motorola's smartphones, tablets, and other devices. Claim 1 is representative of how the disputed term "programming module" is recited in the claims:

1. An interactive computer-controlled display system comprising:

a processor;

a data display screen coupled to the processor;

a cursor control device coupled to said processor for positioning a cursor on said data display screen;

a window generation and control logic coupled to the processor and data display screen to create an operating environment for a plurality of individual ***programming modules*** associated with different application programs that provide status and/or control functions, wherein the window generation and control logic generates and displays a first window region having a plurality of display areas on said data display screen, wherein the first window region is independently displayed and independently active of any application program, and wherein each of the plurality of display areas is associated with one of the plurality of individual ***programming modules***, the first window region and the plurality of independent display areas implemented in a window layer that appears on top of application programming windows that may be generated; and

an indicia generation logic coupled to the data display screen to execute at least one of the plurality of individual ***programming modules*** to generate information for display in one of the plurality of display areas in the first window region, wherein at least one of the plurality of display areas and its associated ***programming module*** is sensitive to user input, and further wherein the window generation and control logic and the indicia generation logic use message-based communication to exchange information to coordinate activities of the indicia generation logic to enable interactive display activity.

(Ex. 10, '002 Patent, at 22:11-43 (emphasis added).)

Motorola's proposed construction for "programming module" is consistent with the intrinsic evidence, particular the '002 prosecution history in which the '002 inventor repeatedly disclaimed subject matter to overcome prior art. Apple's proposed construction, on the other hand, precisely matches up with the prior art distinguished by the inventor. Accordingly, the Court should adopt Motorola's proposed construction.

2. Description Of The '002 Patent

The '002 Patent describes a control strip that includes a plurality of individual programming modules which provide status and control functions:

- The '002 Patent describes "a control and/or status window for display on the desktop of the computer system . . . referred to herein as the control strip." (Ex. 10 at 5:62-67.)

- "The control strip of the present invention is a window of graphics depicting one or more display areas for control and/or status indicia." (*Id.* at 5:65-67.)
- "The control strip is a control panel that provides an operating environment for control strip modules," (*id.* at 6:35-36), which "provide status and control functions," (*id.* at 7:58-60).

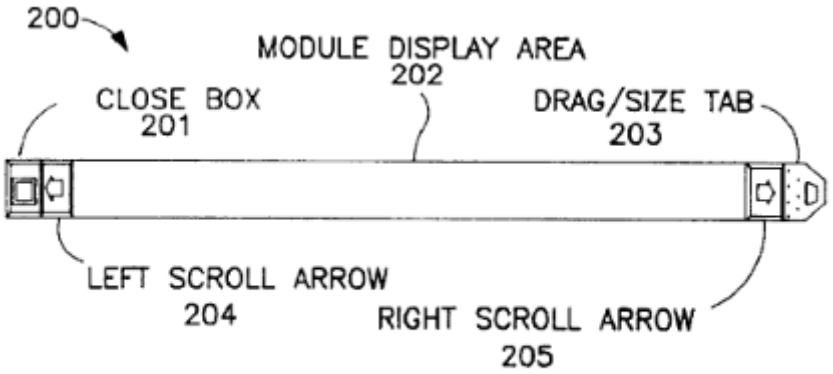


FIG. 2B

As explained throughout the '002 specification, the "programming modules" that appear within each independent claim – including asserted claims 1, 21, and 46 – are not single-function pieces of code (*e.g.*, Apple's proposed construction), but complex programs that are separately initialized, have detailed processing logic, multiple "features," can draw themselves, have distinct idle states and help states and particular functions associated with each, and can have their own settings which can be saved. (*See, e.g.,* Ex. 10 at Figs. 6-8, 10.)

Because of a crowded prior art field in the area of status bars at the time the applicant filed for the '002 Patent, a long and contentious eight-year prosecution led to significant amendments and remarks confining the claims of the '002 Patent to the specific "control strip" and "programming modules" described at length in the '002 Patent. The '002 Patent issued with the term "programming module" used repeatedly and deliberately throughout every single independent claim. By necessity due to the crowded prior art field, the '002 Patent claims a very specific type of control strip that provides an operating environment for a plurality of individual

programs that provide specific status and control functions. Any other construction of "programming module" would allow Apple to claim not only what it did not invent, but what it specifically distinguished as part the '002 prosecution history.

3. The Court Should Construe The Term "Programming Module" Because It Will Confirm That Motorola's Accused Products Do Not Infringe The Asserted Claims of the '002 Patent

Apple asserts claims 1, 21, and 46 of the '002 Patent against certain of Motorola's smartphones, tablets, and other devices. Asserted claims 1, 21, and 46 require, among other things, an "operating environment for a plurality of individual programming modules associated with different application programs that provide status and/or control functions." (Ex. 10 at 22:18-20, 25:810, 26:40-44.)

Apple alleges that certain "indicia" satisfy the "programming modules" limitations of the asserted claims. (*See* Ex. 28, Apple's '002 Patent Infringement Contentions Chart, at p. 28 ("For example, if a user touches the 'new email' indicia in the above reproduced Notification Window, an Intent will result in an Activity including the opening of an email programming module and if a user touches the "new voicemail" indicia in the above reproduced Notification Window, an Intent will result in an Activity including opening a voicemail programming module.").) Those indicia, however, are not "individual programs added to the control strip that provide specific status and control functions."

As discussed above, the specification of the '002 Patent makes plain that a "programming module" is an individual program added to the control strip that provides a specific status or control function. The prosecution history of the '002 Patent confirms this interpretation, as explained below. For at least the reason that the accused "indicia" of Motorola's accused products are not "individual programs added to the control strip that provide specific status or

control functions," the accused products do not infringe any asserted claim of the '002 Patent under a proper construction for the claim term "programming module."

4. The Court Should Adopt Motorola's Proposed Construction

Apple's proposed construction, which is completely untethered from the '002 specification and renders eight years of significant amendments and remarks in prosecution meaningless, should be rejected in favor of Motorola's proposed construction, which holds true to the claims, the specification, and the prosecution history of the '002 Patent.

The patent specification is considered "the single best guide to the meaning of a disputed term." *Phillips*, 415 F.3d at 1315 (quoting *Vitronics Corp.*, 90 F.3d at 1582). One should also look to the prosecution history, which "can often inform the meaning of the claim language by demonstrating how the inventor understood the invention and whether the inventor limited the invention in the course of prosecution, making the claim scope narrower than it otherwise would be." *Id.* at 1317.

Here, the specification and prosecution history confirm that "programming module" means "an individual program added to the control strip that provides a specific status or control function," not "code that performs a function."

The specification and drawings of the '002 Patent demonstrate that a "programming module" is hardly a single-function piece of code (*i.e.*, Apple's proposed construction). Instead, it is an individual program added to the control strip capable of performing multiple tasks and having a variety of states. Modules are added to the control strip, have detailed processing logic, and can have their own settings which can be saved. (*See, e.g.*, Ex. 10 at 8:52-55, 9:15-29, 9:58-66, Fig. 8.) Modules have "features," not a single function, (*see, e.g., id.* at Fig. 6), and modules are called to draw themselves, (*see, e.g., id.* at 11:44-48, Fig. 7.) Modules have distinct help states and idle states and have functions associated with both. (*See, e.g., id.* at 12:9-22, 13:53-60,

Figs. 8,10.) A module can send a message to the control strip indicating a desire to update its settings, to resize the display, to be closed, to change its help state, or to save its settings. (*See, e.g., id.* at 12:22-36; 13:15-28, 37-44, Figs. 8,10.)

Moreover, to the extent the applicant for the '002 Patent sought to claim a status bar beyond the particular control strip with a plurality of individual status and control programs actually described in the '002 Patent, these attempts were squarely rejected by the PTO because the ***concept*** of a status bar was well-known, and robustly taught, in the art by the time the '237 Application was filed in late 1994.

The parent application to the '002 Patent, U.S. Patent App. No. 08/316,237 ("the '237 Application"), was filed September 30, 1994 into a field crowded with prior art. As window-based graphical user interfaces became popular in the 1980s, the problem of screen "clutter" and organization of multiple open windows became an important issue for developers, and the concept of a status bar was proposed no later than 1986. (*See, e.g.*, Ex. 35, D.A. Henderson, Jr. & S.K. Card, *Rooms: The Use of Multiple Virtual Workspaces to Reduce Space Contention*, 5 ACM Transactions on Graphics, No. 3, July 1986, at 221-42.)

By September 30, 1994, the general concept of a status bar – and even a status bar with control functions – was hardly novel, having been taught by a variety of references throughout the 1980s and early 1990s. In fact, Apple itself filed a patent application for a "Status Bar for Application Windows" two years before it filed the parent application to the '002 Patent. (*See* Ex. 33, U.S. Patent 5,588,105 ("Foster '105").) Apple's 1992 status bar "ha[d] a common format to provide a more uniform graphical user interface for the user of the computer system" and "carrie[d] at least one active area" which could "be used to display information derived from the application program, display information derived from the computer system ('global

information'), or control a function of the computer system ('global control')." (*Id.* at 1:58-2:4.) The preferred embodiment of the 1992 Apple status bar was the Apple Newton pen-based mobile device, which displayed only one application "window" at a time, so that as used in Foster '105, "a 'window' [wa]s the entire screen or any portion of an entire screen which Is dedicated to a particular application program," 5:12-15, as displayed below in figure 2 from Foster '105:

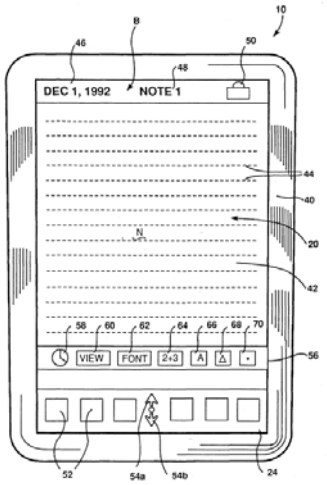
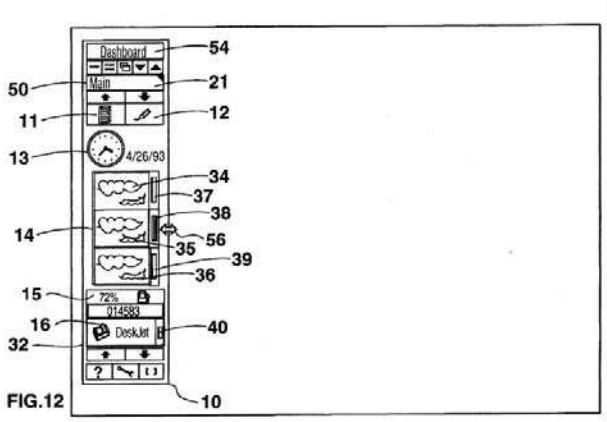


FIG. 2

Several months before Apple filed its 1992 status bar application, and over two years before Apple applied for what would become the '002 Patent, George Hansen et al. filed an application for a patent on a "User Interface with Individually Configurable Panel Interface for Use in a Computer System." (*See, e.g.*, Ex. 34, U.S. Patent No. 5,659,693 ("Hansen '693").)[16] Hansen '693 taught a status bar with a plurality of individual panels corresponding to various programs, windows, and system and other functions, as indicated in figure 12 below. The modules displayed status information to a user in a centralized location and allowed the user to interact with programs and system functions by clicking on the modules.

[16] A number of patents claim priority to U.S. Patent Application No. 07/936,692, the patent application filed August 27, 1992 to which Hansen '693 claims priority. However, Hansen '693 was the patent particular patent in this family cited repeatedly by the examiner as the basis for rejections of the claims of the '002 patent in prosecution, so Hansen '693 will be the focus here.



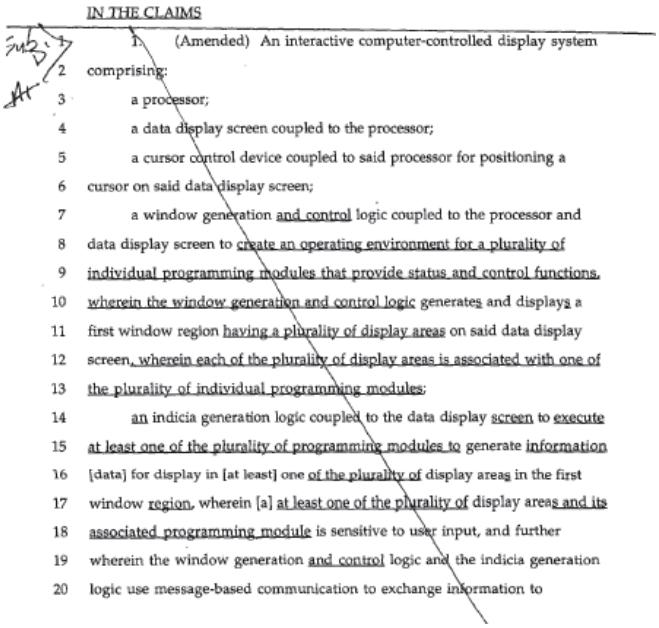
Hansen '693 at fig. 12

On September 30, 1994, nearly a decade after Henderson et al. had described a status and control bar and almost two years after Apple had filed for its own patent on a status bar providing application-specific and "global" status and control functions, Steve Christensen filed the '237 Application for what would eventually become the '002 Patent. Christensen – the sole named inventor on the '002 Patent – was not one of the named inventors on Apple's 1992 status bar patent application, and the '237 Application was framed in a very specific way to attempt to distinguish itself from Apple's previous status bar applications and the crowded field of prior art relating to status and control bars. In particular, as explained in the overview of the specification and drawings of the '002 Patent outlined earlier in this brief, the '237 Application focused not on the ***concept*** of providing status and control information in a central location, but on the particular ***programming methodology*** used to present this information: a "control strip" that created a separate, distinct program for each type of status or content information that was to be presented to the user. The '237 Application (and ultimately the '002 Patent) described these individual programs within the control strip as "programming modules."

When the applicant for the '002 Patent filed the '237 Application on September 30, 1994, claim 1 read as follows:

> An interactive computer-controlled display system comprising:
>
> a processor;
>
> a data display screen coupled to said processor for positioning a cursor on said data display screen;
>
> a window generation logic coupled to the processor and the data display screen to generate and display a first window region on said data display screen;
>
> indicia generation logic coupled to the data display to generate data for display in at least one display area in the first window, wherein a display area is sensitive to user input, and further wherein the window generation logic and the indicia generation logic use message-based communication to exchange information to coordinate activities of the indicia generation logic to enable interactive display activity.

(Ex. 37, U.S. Patent App. No. 08/316,237, at 51-52.) The claim did not include any limitations relating to "programming modules" or to "an operating environment" for these programming modules (*i.e.*, the control strip). The PTO quickly disabused the applicant of the notion that such a broad claim to a status bar could be patentable over the voluminous prior art in the field, rejecting all claims of the '237 Application under section 102(b) as being unpatentable over U.S. Patent No. 5,202,961. (*See generally* Ex. 36, U.S. Patent No. 5,202,961 ("Mills").) In response, claim 1 was amended to include specific limitations for both the programming modules and the control strip:



IN THE CLAIMS

1. (Amended) An interactive computer-controlled display system comprising:

a processor;

a data display screen coupled to the processor;

a cursor control device coupled to said processor for positioning a cursor on said data display screen;

a window generation and control logic coupled to the processor and data display screen to create an operating environment for a plurality of individual programming modules that provide status and control functions, wherein the window generation and control logic generates and displays a first window region having a plurality of display areas on said data display screen, wherein each of the plurality of display areas is associated with one of the plurality of individual programming modules;

an indicia generation logic coupled to the data display screen to execute at least one of the plurality of programming modules to generate information [data] for display in [at least] one of the plurality of display areas in the first window region, wherein [a] at least one of the plurality of display areas and its associated programming module is sensitive to user input, and further wherein the window generation and control logic and the indicia generation logic use message-based communication to exchange information to

(Ex. 32, '002 Patent Prosecution History, Amendments and Remarks dated Aug. 20, 1996, at p. 27.)

In connection with the above amendments, the applicant remarked: "Thus, the present invention provides logic that creates an operating environment like a shell for other programming modules to ***provide status and control functions***. Mills does not provide such an environment." (*Id.* (emphasis added).)

In a Final Office Action mailed November 20, 1996, the PTO again rejected every claim of the '237 Application, this time under section 102(a) as being anticipated by EP Patent No. 0584392A1 ("Cohausz"). This rejection was met with additional amendments and remarks in which the applicant characterized "the plurality of individual programming modules," each "associated with" one of "the plurality of display areas" as "a ***plurality of programs corresponding to a plurality of fields as claimed in the present invention***." (*Id.* at 18 (emphasis added).)

After these Amendments and Remarks, the applicant abandoned the '237 Application and filed a continuation application, U.S. Patent Application No. 08/821,004 ("the '004 Application"), on March 20, 1997. In a Final Office Action mailed August 14, 1997, the PTO rejected all claims of the '004 Application under section 102(a) based on Cohausz. In response, the applicant further amended the claims and again made remarks regarding "programming modules." (*Id*. at 23-25)

A few months later, in an Office Action dated March 25, 1998, the PTO again rejected all claims of the '004 Application. This time, claims 1-3 and 8-24 were rejected under Cohausz in view of Foster '105, the 1992 Apple status bar application discussed earlier in this brief. The applicant distinguished Foster '105 by remarking to the PTO that

> in order to generate [Foster '105's] status bar, the process initially starts a new application program and then couples the status bar to the application window. . . . Foster further teaches that the icons on the status bar relate to actions within the application window. Thus, for example, Foster teaches a notepad which has a status bar including items such as a view button, font button, nib button, close button, etc. In addition to these specific buttons that are associated with the application, a global clock button is also taught by Foster to display the current time. . . .
>
> ***Neither Cohausz nor Foster teach or suggest the plurality of display areas associated with individual programming modules.***

(*Id.* at 33 (emphasis added).)

The distinction drawn by the applicant in his June 25, 1998 Amendments and Remarks would be wholly illusory under Apple's proposed construction, since Foster '105 clearly meets the limitation "a plurality of display areas associated with individual programming modules" if this limitation means "a plurality of display areas associated with individual code that performs a function" (Apple's proposed construction inserted for "programming modules"). Indeed, this is

exactly how the applicant characterized Foster '105, explaining that "icons on the status bar relate to actions within the application window."

Motorola's proposed construction, on the other hand, makes perfect sense in connection with the representations made by the applicant to the PTO: Foster '105, as characterized by the applicant, arguably does not meet the limitation "a plurality of display areas associated with individual programs added to the control strip that perform a specific status or control function," because these are (as characterized by the applicant) "icons on the status bar [that] relate to actions within the application window."

After the applicant's June 25, 1998 Amendments and Remarks, the claims of the '004 Application were again rejected by the PTO under § 102 and/or § 103 two additional times, on September 18, 1998 and May 6, 1999, respectively. The prior art cited as the basis for these rejections included Hansen '693, discussed earlier in this brief. Both rejections were met with additional Amendments and Remarks by the applicant, on December 10, 1998, November 8, 1999, respectively.

On June 1, 2001, the applicant submitted an appeal brief to the PTO. In this appeal brief, the applicant discussed at length the purported distinctions between the claims of the '002 Patent and the prior art cited by the examiner, including Cohausz and Hansen '693. In a section of the appeal brief titled "Summary of the Invention," the applicant explained:

> "The control strip of the present invention provides a standard screen location for a collection of individual modules that provide status and control functions. . . . ***[T]he control strip acts as a status and control function bar, or windowing area, that provides running modules to be displayed in an arrangement that is to be displayed . . . The control strip includes a plurality of individual modules. Each module includes its own initialization process. Thus, as the control strip is initiated, the modules are each called, and if a module is able, it is added to the control strip in***

> ***the appropriate locations. After initialization, during an idle period, module tasks are run.***"

(*Id.* at 59 (emphasis added).) As represented by the applicant to the PTO, the claimed "programming modules" were not simply "code that provides a function," but actual "running" programs that were initialized by and added to the control strip with "module *tasks*."

Furthermore, to distinguish Hansen '693 in his appeal brief, the applicant stated: "Hansen discusses a dashboard including a plurality of buttons, to permit management of various items. Hansen provides a plurality of functions, such as rolodex, calendar, etc. . . . However, Hansen does not teach or suggest a plurality of independent application programs associated with a plurality of independent areas in a display system." (*Id.* at 13.) As with the applicant's asserted distinction between the claims of the '002 Patent and Foster '105, the alleged distinction between the claims of the '002 Patent and Hansen '693 would be illusory under Apple's proposed construction of "programming module." Under Apple's proposed construction, "a plurality of individual programming modules" equates to "a plurality of individual code that performs a function." But this is ***exactly*** what Hansen '693 (as well as Foster '105) represents, as expressly stated by the applicant: "Hansen provides a plurality of functions . . . ." In short, the applicant's statements in prosecution of the '002 Patent ***expressly*** rule out Apple's proposed construction.

Motorola's proposed construction, on the other hand, fits within the representations made by the applicant in prosecution of the '002 Patent. As illustrated in the excerpts above, the applicant repeatedly asserted that the "programming modules" of the claims were individual programs or applications added to the control strip that provide status or control functions.

In short, the specification, claims, and prosecution history of the '002 Patent support Motorola's proposed construction for the claim term "programming module." At the same time, Apple's construction was disclaimed in prosecution of the '002 Patent in order to distinguish

prior art that included a plurality of individual [segments of] code that perform a function, such as Foster '105 and Hansen '693. Accordingly, the court should adopt Motorola's proposed construction of the term "programming module": "an individual program added to the control strip that performs a specific status or control function."

IV. INTRODUCTION TO MOTOROLA'S PATENTS

The parties' present briefing involves claim construction disputes concerning four of the six patents asserted by Motorola in this litigation. These patents each describe aspects for communicating data in a communication system such as a cellular telephone network or a wireless network. Specifically, U.S. Patent No. 6,175,559 (the "'559 Patent") relates to a method for generating preamble sequences, which are used to identify and coordinate timing of messaging in a cellular telephone network. U.S. Patent No. 5,490,230 (the "'230 Patent") relates to a system and method for coding speech information and recovering and reproducing speech from the speech coded information. U.S. Patent No. 5,319,712 (the "'712 Patent") relates to a method for encrypting data in a data stream prior to transmitting the data. And U.S. Patent No. 5,572,193 (the "'193 Patent") relates to a method for authenticating a subscriber unit in a communication system.

The value of these four patents is demonstrated by their relationship to certain standards related to cellular telephone networks and wireless networks. The 3rd Generation Partnership Project ("3GPP") has developed several Technical Standards ("T.S.") used by manufacturers to configure handsets for communication in a cellular telephone network. The 3GPP T.S. 25.213 standard relates to spreading and modulation, which includes requirements on generating preamble sequences. Claims in the '559 Patent recite methods for generating preamble sequences that meet these requirements. Thus, products that generate preamble sequences in accordance with the 3GPP T.S. 25.213 standard necessarily infringe the '559 Patent.

Another standard promulgated by the 3GPP is the T.S. 26.090 standard relating to Adaptive Multi-Rate ("AMR") speech coding and decoding. This standard includes requirements about how speech data is coded on the transmission side and decoded to produce recovered speech components on the receiver side. Claims in the '230 Patent recite a system and method for recovering speech components on the receiver side that meet these requirements. As a result, products that generate recovered speech components in accordance with the 3GPP T.S. 26.090 standard necessarily infringe the '230 Patent.

Another standard setting group, the Institute for Electrical and Electronics Engineers ("IEEE") has developed standards relating to wireless networks. Primary among these wireless standards is the 802.11 standard, which has gone through several iterations and improvements. The 802.11 standard sets forth many requirements for complying with the standard including the manner in which data is encrypted and the manner in which devices are authenticated. Claims in the '712 Patent recite a method for encrypting data that meets the encryption requirements of the 802.11 standard. Similarly, claims in the '193 Patent recite a method for authenticating a device that meets the authentication requirements of the 802.11 standard. Accordingly, products that comply with these aspects of the 802.11 standard necessarily infringe both the '712 and '193 patents.

Apple has several products that are capable of communicating in a cellular telephone network including the iPhone and the iPad (collectively "Apple's cellular products"). Apple also has several products that are capable of communicating in a wireless network including the iPhone, iPad, iPod Touch, MacBook, and others (collectively "Apple's wireless products"). As part of its configuration for communicating in a cellular telephone network, Apple's cellular products are identified as being compliant with Wideband Code Division Multiple Access

("WCDMA"). Compliance with WCDMA is governed by the Technical Specifications developed by 3GPP, including T.S. 25.213 and 26.090. Accordingly, Apple's cellular products necessarily infringe both the '559 Patent and the '230 Patent. With respect to operation in wireless networks, Apple's wireless products are identified as being compliant with the relevant portions of the IEEE 802.11 standard. Thus, Apple's wireless products infringe both the '712 Patent and the '193 Patent.

V. CONSTRUCTION OF TERMS IN MOTOROLA'S PATENTS

A. United States Patent Number 6,175,559

1. Introduction Of The Disputed Phrase "Preamble Sequence"

The parties dispute the construction of the phrase "preamble sequence" found in U.S. Patent No. 6,175,559. The parties' proposed constructions for "preamble sequence" are set forth below:

Disputed Claim Phrase	Motorola's Proposed Construction	Apple's Proposed Construction
preamble sequence	"a signal preceding the transmission of a message that is the product of an inner code and an outer code"	"a signal that is sent by the transmitter prior to the transmission of the information-bearing signal"

Motorola asserts claims 4 and 5 from the '559 Patent against certain of Apple's products. The parties dispute the meaning of the term "preamble sequence," which is found in claim 1, from which asserted claim 4 depends, and asserted claim 5. Claims 1 and 5 are set forth below:

> 1. A ***method for generating preamble sequences in a CDMA system***, the method comprising the steps of:
>
> forming an outer code in a mobile station, wherein the outer code is formed from a generator from the group consisting of a maximum length sequence generator, a Kasami sequence generator, and a Gold sequence generator;
>
> forming an inner code in the mobile station; and
>
> ***multiplying the outer code by the inner code to generate a preamble sequence***.

5. A ***method for generating preamble sequences in a CDMA system***, the method comprising the steps of:

forming an outer code in a mobile station;

forming an inner code in the mobile station utilizing the following equation:

$$c_i(k) = \sum_{j=0}^{M-1} s_j(k - jP)$$

where s_j, j=0,1, . . . ,M-1 are a set of orthogonal codewords of length P, where M and P are positive integers; and

multiplying the outer code by the inner code to generate a preamble sequence.

(Ex. 11, '559 Patent, at 4:66-5:8, 5:20-35.)

Motorola's proposed construction for the term "preamble sequence" is consistent with the intrinsic evidence and, in particular, the relevant claim language. Apple's construction seeks to add a limitation that is not present in the claim language and is not required by the intrinsic record. Accordingly, the Court should adopt Motorola's proposed construction.

2. Description Of The '559 Patent

U.S. Patent No. 6,175,559 is titled "Method for Generating Preamble Sequences in a Code Division Multiple Access System" and issued on January 16, 2001. The '559 Patent discloses a method for generating preamble sequences in a CDMA communication system from the product of an outer code and an inner code. (Ex. 11 at 2:52-57.) Preamble sequences enable a receiving device, such as a base station, to recognize the timing and frequency of a message being sent from a mobile station.

3. The Court Should Adopt Motorola's Proposed Construction of "Preamble Sequence"

The parties appear to agree that a "preamble sequence" is a signal that precedes the transmission of a message. The main dispute between the parties is whether the "preamble sequence" as used in the claims of the patent must be a signal "that is sent by the transmitter" as

Apple proposes. Motorola proposes that this term be construed consistent with the plain and ordinary meaning of the term and in accordance with the claim language and specification of the '559 Patent. In contrast, Apple's construction attempts to incorporate a transmission requirement that is plainly lacking from the claim language and is not described by the specification of the '559 Patent as a requirement in a method for generating a "preamble sequence."

The term "preamble sequence" is used in the preamble of both claim 1 (from which claim 4 depends) and claim 5, as well as in the last step of both claims. There is no language in claims 1, 4, or 5 that requires that the "preamble sequence" be "sent by the transmitter." [17] Rather, the method recited in these claims simply provides, in short, that an "inner code" and an "outer code" be formed and then multiplied together to generate a "preamble sequence." To add that the preamble must be transmitted would be to add an additional step to the method that is not present in the claims. And while the preferred embodiment in the specification describes that the "preamble sequence" is ultimately to be transmitted by the mobile station, there is no basis in the specification from which to draw the conclusion that a preamble sequence must, by definition, be transmitted in order to be a "preamble sequence" as used in the claims.

The parties also differ as to whether "preamble sequence" should be defined to indicate that it is generated as the "product of an inner code and an outer code." That such language should be included, however, is required by the intrinsic evidence. Claims 1 and 5 both specify that the "preamble sequence" is generated from the product of the outer and inner codes.

---

[17] There is also no antecedent basis in the claims or in the specification for the term "the transmitter" contained in Apple's construction. If one were to insert Apple's construction into the claim, it is unclear what "transmitter" Apple refers to; the term does not appear otherwise in the claim. As such, Apple's proposed construction would introduce an ambiguity into the term "preamble sequence" which could be argued by Apple will make the claims unclear or indefinite. The Court should decline any construction which could render the patent invalid. *See Liebel-Flarsheim Co. v. Medrad, Inc.*, 358 F.3d, 898, 911 (Fed. Cir. 2004).

Moreover, the specification provides description explicitly supporting the language in each claim that the "preamble sequence" is formed from the product of outer and inner codes (*see, e.g.*, Ex. 11 at 4:27-30.) Because the "preamble sequence" ***is*** the product of the inner code and outer code in the patent, it is consistent in light of the patent to construe the term accordingly.

Only Motorola's proposed construction of the term "preamble sequence" is consistent with the intrinsic evidence. Apple's construction is not. Thus, the Court should adopt Motorola's proposed construction for "preamble sequence."

4. Introduction Of The Disputed Phrase "Outer Code"

The parties also dispute the construction of the phrase "outer code" found in the '559 Patent.

Disputed Claim Phrase	Motorola's Proposed Construction	Apple's Proposed Construction
outer code	"a code formed from a pseudorandom sequence generated in the mobile station"	"a code that consists of multiple repetitions of a sequence of chips, and that is common for all handset transmitters"

The parties dispute the meaning of the term "outer code," which is found in claim 1, from which asserted claim 4 depends, and asserted claim 5. Claims 1 and 5 are set forth below:

> 1. A method for generating preamble sequences in a CDMA system, the method comprising the steps of:
>
> ***forming an outer code in a mobile station***, wherein ***the outer code is formed from a generator from the group consisting of a maximum length sequence generator, a Kasami sequence generator, and a Gold sequence generator***;
>
> forming an inner code in the mobile station; and
>
> ***multiplying the outer code by the inner code to generate a preamble sequence***.
>
> 5. A method for generating preamble sequences in a CDMA system, the method comprising the steps of:
>
> ***forming an outer code in a mobile station;***

forming an inner code in the mobile station utilizing the following equation:

$$c_i(k) = \sum_{j=0}^{M-1} s_j(k - jP)$$

where s_j, j=0,1, . . . ,M-1 are a set of orthogonal codewords of length P, where M and P are positive integers; and

multiplying the outer code by the inner code to generate a preamble sequence.

(Ex. 11, '559 Patent, at 4:66-5:8, 5:20-35.)

Motorola's proposed construction for the term "outer code" is consistent with the intrinsic evidence and the relevant claim language. Apple's construction seeks to add multiple specific limitations into the construction that are not found in the claim language and are merely preferred aspects of the "outer code" that are not required by the intrinsic record. Accordingly, the Court should adopt Motorola's proposed construction.

5. The Court Should Adopt Motorola's Proposed Construction of "Outer Code"

Motorola proposes that this term be construed consistent with the plain and ordinary meaning of the phrase in light of the claim language and specification of the '559 Patent. Apple's proposed construction, on the other hand, seeks to import limitations from the specification into the claims, contrary to the teachings of *Phillips*. *See* 415 F.3d at 1323. The Federal Circuit has maintained that a "fine line" exists between permissible and impermissible uses of the specification. *Id*. Apple's proposed construction crosses that line.

The parties disagree whether the "outer code" in the context of the patent claims must be "common for all handset transmitters" and one "that consists of multiple repetitions of a sequence of chips," as Apple proposes, or whether the code is "formed from a pseudorandom sequence," as Motorola proposes. Nothing in the claims, the specification, or prosecution history

requires or even suggests that the term "outer code" should be read to include the additional limitations imposed by Apple's proposed construction.

There is no basis whatsoever in the claims to find that the "outer code" must be "common for all handset transmitters." Claims 1 and 5 simply recite "forming an outer code in *a mobile station*." Nowhere do they require, or even suggest, that the "outer code" formed in one "mobile station" must be the same as the "outer code" formed in another mobile station.

At best, the specification indicates that the "[o]uter code is *preferably* common for all transmitters." (Ex. 11 at 3:46-47 (emphasis added).) Elements of the preferred embodiments, however, should not be read into the limitations of the claims. *See Fuji Photo Film Co. v. International Trade Comm'n*, 386 F.3d 1095, 1106 (Fed. Cir. 2004) ("It is a familiar axiom of patent law, however, that the scope of the claims is not limited to the preferred embodiments described in the specification.") Apple's proposed construction, therefore, must be rejected.

Apple's assertion that the "outer code" should be interpreted as "a code that consists of multiple repetitions of a sequence of chips" also must be rejected as that feature is merely a preferred aspect of the "outer code" and not required by either the claim language or by the specification of the '559 Patent. Like Apple's construction requiring the "outer code" be "common for all handset transmitters," there is no basis whatsoever in the claims to find that the "outer code" must be "a code that consists of multiple repetitions of a sequence of chips." Such language is completely absent from claims 1, 4, and 5.

Moreover, the language in other claims of the '559 Patent confirms that the absence of language in claims 1, 4, and 5 relating to "multiple repetitions of a sequence of chips" was intentional and not intended to be a limitation on the "outer code." Claim 2, which depends from claim 1, adds the limitation that the period of the "outer code" comprises K symbols. A code is

periodic when the code is formed from repetitions of a sequence of symbols of a fixed length. Thus, an "outer code" with a period of K symbols would include repetitions of a sequence of K symbols.

Claim 2, therefore, expressly requires the generation of "code that consists of multiple repetitions of a sequence of chips" –claim 1 does not. No such language exists in claim 5 either. Apple's proposed construction would impose reading the limitation of claim 2 into the "outer code" recited in claims 1, 4, and 5. To read the term "outer code" as Apple proposes would impermissibly conflate claims 1 and 2, render claim 2 meaningless, and import limitations into claim 5 that do not exist. *See Liebel-Flarsheim Co. v. Medrad, Inc.*, 358 F.3d 898, 910 (Fed. Cir. 2004) ("As this court has frequently stated, the presence of a dependent claim that adds a particular limitation raises a presumption that the limitation in question is not found in the independent claim.")

The specification also explicitly states that the "outer code" in "the *preferred* embodiment . . . is periodic with period K chips . . . ." (Ex. 11 at 3:41-45 (emphasis added).) Again, this describes "a code that consists of multiple repetitions of a sequence of chips" as merely a "preferred embodiment," not a requirement of the "outer code." Because Apple's construction attempts to import a limitation into the construction of "outer code" that is completely absent from the language of claims 1, 4, and 5, is referenced as merely a preferred embodiment in the specification, and is already embodied in claim 2, Apple's construction of "outer code" in regards to "multiple repetitions of a sequence of chips" must fail.

Motorola's proposed construction, on the other hand, is consistent with the claim language itself and the specification of the '559 Patent in a manner that does not improperly limit the claims to a preferred embodiment. According to the '559 Patent, the "outer code" is

preferably formed from sections of a maximal-length sequence, Gold sequence or Kasami sequence. (*See id.* at 3:41-45.) All three of these sequences are pseudorandom sequences. Since the '559 Patent specifically identifies the three sequences as preferred, it would be improper to limit the "outer code" to being one of the three sequences. The fact that claim 4 explicitly recites the "outer code" being one of the three sequences, whereas claim 5 does not, confirms that the "outer code" should not be interpreted as being limited to one of the three sequences. However, based on the specification, one of ordinary skill in the art would understand the "outer code" to be formed from a pseudorandom sequence, albeit not necessarily one of the three identified sequences.

Thus, the Court should adopt Motorola's proposed construction for "outer code."

B. United States Patent Number 5,490,230

1. Introduction Of The Disputed Phrase "Long Term Energy Value for [the/a] frame of information"

The parties dispute the construction of the phrase "long term energy value for [the/a] frame of information" found in U.S. Patent No. 5,490,230.

Disputed Claim Phrase	Motorola's Proposed Construction	Apple's Proposed Construction
long term energy value for [the/a] frame of information	Plain meaning of the terms; or "a value representative of the energy in a block of data"	"the total energy for the current frame of speech"

Motorola asserts claims 6 to 8 from the '230 Patent against certain of Apple's products. The parties dispute the meaning of the phrase "long term energy value for [the/a] frame of information," which is found in each of the asserted claims. Claims 6 to 8 are set forth below:

> 6. A method for recovering information that relates to gain information for excitation components of a speech sample, ***wherein the speech sample is digitized to provide a frame of information*** comprising at least one subframe, the method comprising the steps of:

A) ***receiving at least one parameter comprising a long term energy value for the frame of information***;

B) receiving excitation component definition information for at least one excitation component;

C) processing the excitation component definition information to provide a pre-component, which pre-component has an energy value;

D) ***determining a gain value that is proportional to the long term energy value*** and inversely proportional to the energy value; and

E) applying the gain value to the pre-component, to provide a recovered excitation component of the speech sample.

7. A method for recovering information that relates to gain information for excitation components of a speech sample, wherein ***the speech sample is digitized to provide a frame of information*** comprising at least one subframe, the method comprising the steps of:

A) receiving a radio signal;

B) demodulating the radio signal to provide a recovered signal;

C) ***extracting from the recovered signal at least one parameter comprising a long term energy value for the frame of information***;

D) extracting from the recovered signal excitation component definition information for at least one excitation component;

E) processing the excitation component definition information to provide a pre-component, which pre-component has an energy value;

F) ***determining a gain value that is proportional to the long term energy value*** and inversely proportional to the energy value; and

G) applying the gain value to the pre-component to provide a recovered component of the speech sample.

8. A radio that receives speech coded information and that synthesizes speech in response thereto, comprising:

A) RF means for receiving and demodulating a radio signal that includes speech coded information;

B) excitation source means operably coupled to the RF means for receiving the speech coded information; and for:

1) ***extracting from the speech coded information at least one parameter comprising a long term energy value for information***, wherein ***a speech sample is digitized to provide the frame of information*** comprising at last one subframe;

2) extracting from the speech coded information excitation component definition information for at least one excitation component;

3) processing the excitation component definition information to provide a pre-component, which pre-component has an energy value;

4) ***determining a gain value that is proportional to the long term energy value*** and inversely proportional to the energy value;

5) applying the gain value to the pre-component to provide a recovered component of the speech sample;

6) providing an excitation signal using the recovered component; and

C) LPC filter means for receiving the excitation signal and for providing a synthesized speech signal in response thereto.

(Ex. 12, '230 Patent, at 9:7-25, 10:7-37.)

Motorola's proposed construction for the phrase "long term energy value for [the/a] frame of information" is consistent with the intrinsic evidence and in particular, the relevant claim language. Apple's construction seeks to add extraneous limitations that are lacking in the claims and are not required by the intrinsic record, complicate the meaning of the claim language, and create issues of indefiniteness. Accordingly, the Court should adopt Motorola's proposed construction.

2. Description Of The '230 Patent

U.S. Patent No. 5,490,230 is titled "Digital Speech Coder Having Optimized Signal Energy Parameters" and issued on February 6, 1996. Systems requiring transmission of speech signals, such as telecommunication systems, typically use a speech coding process. This process converts analog speech signal into digital speech signals and compresses them to reduce the amount of data that needs to be transmitted. The inventive method of the '230 Patent provides an

improved speech coding process that reduces demands on bandwidth while simultaneously providing increased error correction protection for gain factor information.

3. The Court Should Adopt Motorola's Proposed Construction of "Long Term Energy Value for [the/a] Frame of Information"

Although the parties construction of "long term energy value for [the/a] frame of information" (hereinafter the "long term energy" limitation) appear facially similar, the constructions differ significantly. As a threshold matter, the phrase has an ordinary meaning and Apple's position is to swap one word out for another. For instance, Apple replaces the phrase "long term" with the word "total." This is unnecessary and inaccurate. Second, Apple replaces the word "information" with "speech". Again this is unnecessary and again alters the claim language. Finally, Apple inserts the word "current" where it does not exist in the claim language. None of these changes to the claim language are justified. The Court should adopt the plain meaning of the words in the claim phrase which do not require replacement by different words with different meanings or the addition of words. However, for the sake of clarity, Motorola refers to its proposed construction below.

As an example, Apple's construction requires that the "frame of information" be for a "current frame." This requirement proposed by Apple finds no support in the claim language and is not supported by the specification or the prosecution history. In contrast, Motorola's construction of the "long term energy" limitation as "a value representative of the energy in a block of data" is completely consistent with the claim language itself, the specification, and the prosecution history of the '230 Patent.

The "long term energy" limitation is used in each of claims 6, 7, and 8 in essentially the same manner. Claim 6 recites "receiving at least one parameter comprising a long term energy value for the frame of information" and claims 7 and 8 similarly recite "extracting … at least one

parameter comprising a long term energy value for [the/a] frame of information." In both claims 6 and 7, "the frame of information" refers to the language in the preamble that "the speech sample is digitized to provide a frame of information." Claim 8 similarly recites earlier in the body of the claim that "a speech sample is digitized to provide the frame of information."

There is nothing in the claim language that defines the "frame of information" in reference to any other "frame of information." As such, there is nothing in the claim language that gives any context as to which "frame of information" is being received or extracted. Without this context, it would be impossible to understand what is meant by Apple's added phrase "the current frame," as required by Apple's construction. In particular, it is unclear if "the current frame" is, for example, a frame that has just been received, a frame that is used to provide a recovered component, a recently received frame, or some other frame. There is likewise nothing in the specification to clarify what makes a frame "current." Thus, adding the limitation "current" to the construction not only lacks any basis in the claim language, but also complicates the meaning of the "long term energy" limitation and makes it more indefinite.

Moreover, the performance of the claimed invention can be achieved (or implemented) regardless of whether the long term energy value is for some "current" frame. To meet claims 6, 7, or 8, the only requirement is that at least one parameter comprising the long term energy value for the frame of information is received or extracted and that the determined gain value is proportional to the long term energy value.

Like the claim language, neither the specification nor the prosecution history requires limiting the "frame of information" to being "current." With respect to the specification, the '230 Patent simply explains that "the long term energy value comprises an energy value *that is*

generally representative of a single frame." (Ex. 12 at 4:34-36.) This language certainly fails to require that the long term energy value be representative of some "current" frame.

During prosecution of the '230 Patent, the Applicant indicated that the "long term energy value is defined as $E_q(0)$ on page 9 of the Specification [Lines 1-2]." (Ex. 18, '230 Patent Prosecution History, Amendment dated June 11, 1993, at p. 10.) The lines in the specification referenced by the applicant state that "$E_q(0)$=quantized long term signal energy for total frame…." (Ex. 12 at 4:46.) Like the language of the claim, neither the statement by the applicant during prosecution nor the referenced lines in the specification places any temporal limitation on the "frame" relative to the "long term energy." Instead, they describe only that the "long term energy" is for a particular frame, not necessarily the current frame.

Accordingly, the Court should adopt the plain meaning of the words in the claim or in the alternative Motorola's proposed construction for the "long term energy" limitation.

4. Introduction Of The Disputed Phrase "Extracting From [the Recovered Signal/the Speech Coded Information] At Least One Parameter"

The parties also dispute the construction of the phrase "extracting from [the recovered signal/the speech coded information] at least one parameter" found in the '230 Patent.

Disputed Claim Phrase	Motorola's Proposed Construction	Apple's Proposed Construction
extracting from [the recovered signal/the speech coded information] at least one parameter	"processing performed on [the recovered signal/the speech coded information] to obtain a parameter"	"taking a transmitted value out of the [speech signal/speech coded information]"

The parties dispute the meaning of the phrase "extracting from [the recovered signal/the speech coded information] at least one parameter," which is found in asserted claims 7 and 8. Claims 7 and 8 are set forth below:

7. A method for recovering information that relates to gain information for excitation components of a speech sample, wherein the speech sample is digitized to provide a frame of information comprising at least one subframe, the method comprising the steps of:

A) receiving a radio signal;

B) demodulating the radio signal to provide a recovered signal;

C) ***extracting from the recovered signal at least one parameter comprising a long term energy value for the frame of information***;

D) extracting from the recovered signal excitation component definition information for at least one excitation component;

E) processing the excitation component definition information to provide a pre-component, which pre-component has an energy value;

F) determining a gain value that is proportional to the long term energy value and inversely proportional to the energy value; and

G) applying the gain value to the pre-component to provide a recovered component of the speech sample.

8. A radio that receives speech coded information and that synthesizes speech in response thereto, comprising:

A) RF means for receiving and demodulating a radio signal that includes speech coded information;

B) excitation source means operably coupled to the RF means for receiving the speech coded information; and for:

1) ***extracting from the speech coded information at least one parameter comprising a long term energy value for information***, wherein a speech sample is digitized to provide the frame of information comprising at last one subframe;

2) extracting from the speech coded information excitation component definition information for at least one excitation component;

3) processing the excitation component definition information to provide a pre-component, which pre-component has an energy value;

4) determining a gain value that is proportional to the long term energy value and inversely proportional to the energy value;

5) applying the gain value to the pre-component to provide a recovered component of the speech sample;

6) providing an excitation signal using the recovered component; and

C) LPC filter means for receiving the excitation signal and for providing a synthesized speech signal in response thereto.

(Ex. 12 at 9:26-10:37.)

Motorola's proposed construction for the phrase "extracting from [the recovered signal/the speech coded information] at least one parameter" is consistent with the intrinsic evidence including the relevant claim language and the specification. Apple's construction seeks to add a limitation into the construction that is plainly lacking from the claim language and is not required by the intrinsic record. Accordingly, the Court should adopt Motorola's proposed construction.

5. The Court Should Adopt Motorola's Proposed Construction Of "Extracting From [the Recovered Signal/the Speech Coded Information] At Least One Parameter"

The main dispute between the parties is whether the construction of the phrase "extracting from [the recovered signal/the speech coded information] at least one parameter" requires that the parameter extracted from the recovered signal/speech coded information be "a transmitted value," as required by Apple's construction. There is no basis in the claim language for Apple's position that the "parameter" derived from the data has to include an actual piece of data from the data being transmitted. The "parameter" simply needs to be derived from the data, but not necessarily include actual pieces of the data.

In RF communication, a digital bit stream that is being transmitted is used to modulate (*i.e.*, modify) a carrier signal. The carrier signal is an RF analog signal. The carrier signal is received by a receiver, demodulated, and sampled to reproduce the digital bit stream that was transmitted. Claim 7 and 8 both recite receiving and demodulating a radio signal.

The digital bit stream reproduced in the receiver may itself contain values that are useful to the receiver. In other words, specific bits of the reproduced digital bit stream may represent a particular value that can be used by the receiver. According to Apple's construction, it appears that a parameter is "a transmitted value" only if specific bits in the reproduced digital bit stream are a value corresponding to a parameter.

More commonly, however, the digital bit stream contains data that the receiver uses to reconstruct a value or some other piece of information. For example, the data in the digital bit stream may have been modified on the transmitting end, such as through encoding or compression, and the receiver must process the digital bit stream to decode and/or decompress the data. The decoded or decompressed data can represent a value used by the receiver to perform a function, such as reproducing speech. It is also possible that specific bits of the digital bit stream must first be processed to develop a useful value. For example, these specific bits may need to be combined in some manner, such as mathematically, to reproduce a value that is then used by the receiver to perform a function. Thus, it would be clear to one of ordinary skill in the art that it may be necessary to perform some form of processing first on the demodulated radio signal to extract a parameter.

Indeed, the relevant description from the specification of the '230 Patent specifically contemplates the need for such processing to extract a parameter from a demodulated radio signal. The '230 Patent, in reference to Fig. 2, discloses that a radio includes an antenna (202) for receiving a speech coded signal (201) and an RF unit (203) that processes the received signal to recover speech coded information. (*See* Ex. 12 at 7:55-58.) "This [speech coded] information is provided to *a parameter decoder (204) that develops control parameters for various subsequent processes*." (*Id*. at 7:58-60 (emphasis added).) By using the term "decoder," the '230

Patent expressly recognizes that some form of processing is done on the speech coded information in order to develop the control parameters.

Apple's construction that the extracted parameter must be "a transmitted value" is thus not supported by the intrinsic record. There is no language in the "extracting" limitation nor elsewhere in claims 7 or 8 that would require the extracted parameter to be "a transmitted value." Moreover, Apple's construction is plainly at odds with the description in the specification of the '230 Patent, which specifically contemplates the use of a parameter decoder to effect processing of the speech coded information to develop parameters.

Motorola therefore proposes construing the "extracting" limitation to mean "processing performed on [the recovered signal/the speech coded information] to obtain a parameter." This construction is completely consistent with the ordinary meaning of the phrase as would be understood by one of ordinary skill in the art, the claim language itself, and the specification of the '230 Patent. In contrast, Apple's construction attempts to incorporate a requirement which has no basis in the claim language and is not required by the specification of the '230 patent. Accordingly, this court should adopt Motorola's proposed construction for the "extracting" limitation.

C. United States Patent Number 5,319,712

1. Introduction Of The Disputed Phrase "transmit overflow sequence number"

The parties dispute the construction of the phrase "transmit overflow sequence number" found in U.S. Patent No. 5,319,712.

Disputed Claim Phrase	Motorola's Proposed Construction	Apple's Proposed Construction
transmit overflow sequence number	"a multi-bit incrementing number that updates within the transmitter when the packet sequence number rolls over"	"a number that is updated within the transmitter when the packet sequence number rolls over, but is not communicated to the receiver"

Motorola asserts claim 17 from the '712 Patent against certain of Apple's products. The parties dispute the meaning of the phrase "transmit overflow sequence number," which is found in the asserted claim. Claim 17 is set forth below:

> 17. In a communication system having a physical layer, data link layer, and a network layer, a method for providing cryptographic protection of a data stream, comprising:
>
> (a) assigning a packet sequence number to a packet derived from a data stream received from the network layer;
>
> (b) ***updating a transmit overflow sequence number as a function of the packet sequence number***; and
>
> (c) ***encrypting, prior to communicating the packet and the packet sequence number on the physical layer, the packet as a function of*** the packet sequence number and ***the transmit overflow sequence number***.

(Ex. 13, '712 Patent, at 8:65-9:12 (emphasis added).)

Motorola's proposed construction for the phrase "transmit overflow sequence number" is consistent with the intrinsic evidence including the relevant claim language and the specification. Apple's construction seeks to add a negative limitation into the construction that is plainly lacking from the claim language and is not required by the intrinsic record. Accordingly, the Court should adopt Motorola's proposed construction.

2. Description Of The '712 Patent

U.S. Patent No. 5,319,712 is titled "Method and Apparatus for Providing Cryptographic Protection of a Data Stream in a Communication System" and issued on June 7, 1994. The inventions of the '712 Patent relate to cryptographic protection of data in communication systems. These inventions provide robust protection of data while also allowing efficient processing of information within the system.

3. The Court Should Adopt Motorola's Proposed Construction

The parties apparently agree as to what the "transmit overflow sequence number" is: "a multi-bit incrementing number that updates when the packet sequence number rolls over," which is used in the transmitter. This is consistent with how "transmit overflow sequence number" is used in the claims and patent specification. The main dispute between the parties is whether the definition of "transmit overflow sequence number" should be defined not only in terms of what the number is, but also whether there are restrictions on how it may be used after encryption has occurred. Apple would add a negative limitation to the definition of "transmit overflow sequence number," barring it from being communicated to the receiver.

But what a "transmit overflow sequence number" is and how it may or may not be used are two different things. There are many ways to use a "transmit overflow sequence number" consistent with the inventions of the '712 Patent, but not all of them are part of the asserted claims. The claim at issue here, claim 17, focuses only on the encryption process. It requires only one use for a "transmit overflow sequence number," which is to encrypt a packet "prior to communicating the packet and the packet sequence number on the physical layer." (Ex. 13 at 9:7-12.) Whether the "transmit overflow sequence number" is transmitted to a receiver after it is used for encryption is irrelevant to the asserted claim's focus on just the encryption process.

The construction proposed by Apple nonetheless would add limitations restricting the use of transmit overflow sequence numbers after encryption. Apple's construction would require withholding transmission of the transmit overflow sequence number after it has been used to generate an encrypt mask. The '712 Patent, and in particular the asserted claim, however, leave to the preferences of the system designer the decision as to what to do with the "transmit overflow sequence number" after encryption. Neither the asserted claims nor the specification forbid transmission of a transmit overflow sequence number. Indeed, by expressly using the

word "transmit," the claim language itself suggests the transmission of the overflow sequence number.

Other claims, not asserted here, do include additional limitations which make it unnecessary to communicate the transmit overflow sequence numbers after encryption. Claim 11 requires, for example, independent generation of a "receive overflow sequence number" (the receive-side version of the "transmit overflow sequence number") to be used for decryption purposes. (*Id*. at 8:14-29.) But even this claim, and others with similar limitations, do not forbid communication of the transmit overflow sequence number to the receiver.

Apple's construction therefore should be rejected and Motorola's adopted.

D. United States Patent Number 5,572,193

1. Introduction Of The Disputed Phrase "transmitting… from the subscriber unit to the communication system"

The parties dispute the construction of the phrase "transmitting… from the subscriber unit to the communication system" found in U.S. Patent No. 5,572,193.

Disputed Claim Phrase	Motorola's Proposed Construction	Apple's Proposed Construction
transmitting… from the subscriber unit to the communication system	"sending [the authentication message and the at least part of the plurality of information bits] from the subscriber unit to the communication system"	"transmitting from a mobile or portable unit over a cellular radio telephone system"

Motorola asserts claim 31 from the '193 Patent against certain of Apple's products. The parties dispute the meaning of the phrase "transmitting… from the subscriber unit to the communication system," which is found in claim 29 from which asserted claim 31 depends. Claims 31 is set forth below:

> 29. A ***method of authenticating a subscriber unit in a communication system***, comprising:

(a) providing the subscriber unit with at least part of a plurality of information bits which uniquely identify a target communication unit;

(b) generating an authentication message in the subscriber unit as a function of the at least part of the plurality of information bits; and

(c) ***transmitting the authentication message and the at least part of the plurality of information bits from the subscriber unit to the communication system***.

(Ex. 14, '193 Patent, at 10:12-10:23.)

Motorola's proposed construction for the phrase "transmitting… from the subscriber unit to the communication system" is consistent with the intrinsic evidence and with the plain and ordinary meaning of the terms. Apple's construction seeks to add a narrow limitation into the construction that is plainly lacking from the claim language and is not required by the intrinsic record. Accordingly, the Court should adopt Motorola's proposed construction.

2. Description Of The '193 Patent

The '193 Patent is titled "Method for Authentication and Protection of Subscribers in Telecommunications Systems" and issued on November 5, 1996. Before being permitted to communicate in a communication system, subscriber units, such as mobile or portable units, are first authenticated. If the authentication process is insufficiently robust, a thief may be able to imitate a legitimate user and gain improper access to the communication system. The '193 Patent discloses an inventive method for authenticating subscriber units in a communication system which allows for the detection of fraudulent users and efficiently protects identification numbers from unauthorized detection.

3. The Court Should Adopt Motorola's Proposed Construction

The phrase "transmitting … from the subscriber unit to the communication system" as used in the '193 Patent should be read according to its plain and ordinary meaning and thus does not require construction. To the extent that the Court believes that the identified term should be

construed, Motorola's construction as explained herein remains true to the plain and ordinary meaning of the disputed term according to the standards established by *Phillips,* 415 F.3d 1303. Apple's proposed construction, on the other hand, again seeks to import limitations from the specification into the claims, contrary to the teachings of *Phillips*. *See id*. at 1323.

The parties do not appear to have a dispute over the term "transmitting" in the step of "transmitting … from the subscriber unit to the communication system." Apple does not even propose a construction for the term "transmitting," and Motorola simply proposes using the common and ordinary meaning of "transmitting" as being "sending." As Motorola believes the step of "transmitting … from the subscriber unit to the communication system" does not require construction, Motorola accepts leaving the term "transmitting" as is or using its common and ordinary meaning as "sending." *See, e.g., Virginia Panel Corp. v. MAC Panel Co.,* 133 F.3d 860, 865-66 (Fed. Cir. 1997) (claim term "reciprocating" given ordinary meaning and not limited to linear reciprocation).

The main dispute between the parties is whether "transmitting … from the subscriber unit to the communication system" must be "over a cellular radio telephone system" as Apple proposes. Apples construction essentially asks this court to interpret "communication system" as used in the claims to mean a "cellular radio telephone system." Nothing in the claim language or the specification of the '193 Patent, however, requires such a limitation.

The step of "transmitting … from the subscriber unit to the communication system" is recited as the last step of asserted claim 29 (from which claim 31 depends). There is no language in claims 29 or 31 that requires that the "transmitting" be "over a cellular radio telephone system." Rather, the method simply provides that the "transmitting" is "to the communication system" without specifying the type of "communication system." If the inventors had meant to

limit the claim to "transmitting over a cellular radio telephone system," the claims would have been drafted to include that limitation. Apple's proposed construction, however, impermissibly attempts to do just that.

Although the patent describes the preferred embodiment in terms of a cellular telecommunications system, the patent claims make it clear that the inventors did not intend to so limit their invention, instead anticipating that the methods claimed would be used in any a wide variety of other "communications systems." Apple's proposed construction is thus inconsistent with the language of the claims and would render elements of the claims and entire claims meaningless.

That Apple's proposed construction is inappropriate is made even more stark when claim 29 is read in context with other claims, in particular claims 20, 23 and 32. Claims 20 and 29 are very similar and both contain the disputed step of "transmitting" as the final limitation in the method. Claims 23 and 32 depend on claims 20 and 29, respectively. Under Apple's proposed construction, the scope of all four of the claims would be limited to transmitting the authentication message over a "cellular radio telephone system." However, based on the differences between claims 20 and 29, and the limitations added by claims 23 and 32, limiting the transmitting of the authentication message to being over a "cellular radio telephone system" would be inconsistent and improper. *See, e.g., Tandon Corp. v. International Trade Comm'n,* 831 F.2d 1017, 1023 (Fed. Cir. 1987) (discussing claim differentiation).

Claim 20 generally describes generating and transmitting an authentication message as a function of "at least a plurality of digits which uniquely identify a target communication unit." As set forth in the abstract and in claim 3, such digits include the "dialed digits" of the called telephone number described in the specification. Although the "digits" are applicable to cellular

radio telephone systems, dependent claim 23 further specifies that the authentication message generated in claim 20 can be transmitted across at least "radio communication link, satellite link, fiber optic cable, coaxial cable, and wireline." Only the first of these mediums, the "radio communication link," is used in communications between mobile stations and base stations in cellular telephone systems. The rest clearly relate to other types of "communications systems," such as landline telephone communications and satellite telephony, among other things. Thus, "communications systems," as that phrase is used in claim 20, is at least applicable to telephone systems generally, not just "cellular radio telephone systems" as would be required by Apple's construction.

Claim 29, while quite similar to claim 20, uses broader language that makes it clear that its authentication method covers any type of communication system, such as computer networks, and not just telephonic communications systems. Instead of "digits" as recited in claim 20, claim 29 recites "information bits" and describes generating and transmitting an authentication message as a function of "at least a plurality of information bits which uniquely identify a target communication unit." In addition, whereas the authentication message in claim 20 is transmitted to "an intermediate communication unit," claim 29 simply recites that the authentication message is transmitted to "the communication system." The use of "information bits" instead of "digits" and transmission to "the communication system" instead of "an intermediate communication unit" clearly demonstrates that claim 29 was intended to have a broader scope than claim 20.

The use of "information bits" instead of "digits" in claim 29 impacts directly on the type of communication system. Whereas a user of a telephonic device would enter "digits" to identify a target telephonic device, "information bits" can be used to identify any type of target device including computing devices, such as a server or a printer, as well as telephonic devices.

Similarly, the absence in claim 29 of "an intermediate communication unit," such as a base station in cellular radio telephone system, and merely requiring transmission to "the communication system" clearly suggests to one of ordinary skill in the art that the "communication system" in claim 29 is a broad, general term that could apply to any system in which a subscriber unit communicates with a target communication unit. Such a system would cover not only a cellular radio telephone system, but also systems like computer networks.

Moreover, similar to the discussion above regarding claim 20 and claim 23, that claim 29 is not limited to cellular communication systems is further confirmed by claim 32. Claim 32, which depends on claim 29 and is identical to claim 23, claims communications of "information bits" via "radio communication link, satellite link, fiber optic cable, coaxial cable, and wireline." In context, it is clear that "information bits" can be transmitted not only through telecommunications systems but also through other "communication systems" such as wired and wireless computer networks, which send information between communication units as bits of data or information. Given the language of claim 29 and the additional limitations of dependent claim 32, claim 29's "communication system" cannot be limited to a cellular radio communication system, as Apple contends.

The specification also confirms the impropriety of limiting the transmission to a cellular radio communication system. In particular, the specification of the '193 Patent discloses that "the invention relates generally to communication systems and more particularly to radio frequency (RF) cellular telecommunication systems." (Ex. 14 at 1:21-23.) This statement expressly recognizes that the "communication system" identified in the claims is a more general term than a cellular radio telephone system. Since the specification identifies the invention as

relating to communication systems generally, and not just cellular radio telephone systems, it is improper to limit the "transmitting" as necessarily being "over a cellular radio telephone system."

Thus, the Court should adopt Motorola's proposed construction for "transmitting … from the subscriber unit to the communication system."

VI. CONCLUSION

For the foregoing reasons, Motorola respectfully requests that the Court adopt Motorola's proposed construction for each of the disputed terms and phrases.

Dated: June 17, 2011

Respectfully submitted,

MOTOROLA MOBILITY, INC.

By: s/ Scott W. Hansen
Scott W. Hansen

Edward J. DeFranco
Alexander Rudis
Richard W. Erwine
Quinn Emanuel Urquhart & Sullivan, LLP
51 Madison Avenue, 22nd Floor
New York, NY 10010
Telephone: (212) 849-7000
Facsimile: (212) 849-7100
Email: eddefranco@quinnemanuel.com
alexanderrudis@quinnemanuel.com
richarderwine@quinnemanuel.com

David A. Nelson
500 West Madison St., Suite 2450
Chicago, IL 60661
Email: davenelson@quinnemanuel.com

Robert W. Stone
Brian Cannon
555 Twin Dolphin Drive, Suite 560
Redwood Shores, CA 94065
Email: robertstone@quinnemanuel.com
briancannon@quinnemanuel.com

Charles K. Verhoeven
50 California Street, 22nd Floor
San Francisco, CA 94111
Email: charlesverhoeven@quinnemanuel.com

Scott W. Hansen
Lynn M. Stathas
Lisa Nester Kass
Reinhart Boerner Van Deuren s.c.
22 East Mifflin Street
P.O. Box 2018
Madison, WI 53701-2018
Telephone: (608) 229-2200
Facsimile: (608) 229-2100
1000 North Water Street, Suite 1700
Milwaukee, WI 53202
Telephone: (414) 298-1000
Facsimile: (414) 298-8097
Email: shansen@reinhartlaw.com
lstathas@reinhartlaw.com
lkass@reinhartlaw.com

Attorneys for Defendants Motorola, Inc. and Motorola Mobility, Inc.