

**IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
TYLER DIVISION**

BLUE SPIKE, LLC,

Plaintiff,

vs.

**HUAWEI TECHNOLOGIES CO., LTD,
ET AL,**

Defendants.

§
§
§
§
§
§
§
§
§
§
§

**CASE NO. 6:13-cv-00679-RWS
LEAD CASE**

MEMORANDUM OPINION AND ORDER

This Memorandum Opinion construes the disputed claim terms in United States Patent No. 5,745,569 (“the ’569 Patent”) asserted by Plaintiff Blue Spike, LLC (“Blue Spike”). On February 11, 2016, the parties presented oral arguments on the disputed claim terms at a *Markman* hearing. For the reasons stated below, the court **ADOPTS** the following constructions.

BACKGROUND

The ’569 Patent relates to technology for the protection of computer code copyrights by including “digital watermarks” in the code. ’569 Patent Abstract. Through such digital watermark techniques, “while an application program can be copied in an uninhibited manner, only the licensed user having the license code” can operate the program. *Id.* The patent describes that in typical programs, certain code resources contain the countermeasures that prevent the unauthorized use of the program and that these resources typically remain in a fixed position. ’569 Patent 7:30–41. However, a skilled hacker could take a “snap-shot” of the code

in memory and determine which code resources comprise the countermeasures and then disable those countermeasures. '569 Patent at 7:41–44.

The techniques disclosed in the '569 Patent randomly reorganize program memory structure intermittently. '569 Patent at 2:21–22. The parties discuss this in the context of “shuffling” the memory. Docket No. 143 at 1; Docket No. 148 at 18. Through shuffling, attempts to capture memory by a hacker can be made more difficult. This reorganization prevents attempts at capturing code for analysis which may be used to eliminate licensing information. '569 Patent at 2:23–32. The only disputed terms appear in Claim 16, which reads in its entirety:

A method for copy protecting a software application executed by a computer system, the software application including a plurality of executable code resources loaded in a memory of the computer system, said method comprising the steps of:

determining an address within the memory of the computer system associated with each of the plurality of executable code resources; and intermittently relocating each of the plurality of executable code resources to a different address within the memory of the computer during execution of the software application.

'569 Patent at 9–20.

APPLICABLE LAW

“It is a ‘bedrock principle’ of patent law that ‘the claims of a patent define the invention to which the patentee is entitled the right to exclude.’ ” *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312 (Fed. Cir. 2005) (quoting *Innova/Pure Water, Inc. v. Safari Water Filtration Sys., Inc.*, 381 F.3d 1111, 1115 (Fed. Cir. 2004)). The Court examines a patent’s intrinsic evidence to define the patented invention’s scope. *Id.* at 1313–1314; *Bell Atl. Network Servs., Inc. v. Covad Commc’ns Group, Inc.*, 262 F.3d 1258, 1267 (Fed. Cir. 2001). Intrinsic evidence includes the claims, the rest of the specification and the prosecution history. *Phillips*, 415 F.3d at 1312–13;

Bell Atl. Network Servs., 262 F.3d at 1267. The Court gives claim terms their ordinary and customary meaning as understood by one of ordinary skill in the art at the time of the invention. *Phillips*, 415 F.3d at 1312–13; *Alloc, Inc. v. Int’l Trade Comm’n*, 342 F.3d 1361, 1368 (Fed. Cir. 2003).

Claim language guides the Court’s construction of claim terms. *Phillips*, 415 F.3d at 1314. “[T]he context in which a term is used in the asserted claim can be highly instructive.” *Id.* Other claims, asserted and unasserted, can provide additional instruction because “terms are normally used consistently throughout the patent.” *Id.* Differences among claims, such as additional limitations in dependent claims, can provide further guidance. *Id.*

“[C]laims ‘must be read in view of the specification, of which they are a part.’ ” *Id.* (quoting *Markman v. Westview Instruments, Inc.*, 52 F.3d 967, 979 (Fed. Cir. 1995)). “[T]he specification ‘is always highly relevant to the claim construction analysis. Usually, it is dispositive; it is the single best guide to the meaning of a disputed term.’ ” *Id.* (quoting *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1582 (Fed. Cir. 1996)); *Teleflex, Inc. v. Ficosa N. Am. Corp.*, 299 F.3d 1313, 1325 (Fed. Cir. 2002). In the specification, a patentee may define his own terms, give a claim term a different meaning that it would otherwise possess, or disclaim or disavow some claim scope. *Phillips*, 415 F.3d at 1316. Although the Court generally presumes terms possess their ordinary meaning, this presumption can be overcome by statements of clear disclaimer. *See SciMed Life Sys., Inc. v. Advanced Cardiovascular Sys., Inc.*, 242 F.3d 1337, 1343–44 (Fed. Cir. 2001). This presumption does not arise when the patentee acts as his own lexicographer. *See Irdeto Access, Inc. v. EchoStar Satellite Corp.*, 383 F.3d 1295, 1301 (Fed. Cir. 2004).

The specification may also resolve ambiguous claim terms “where the ordinary and accustomed meaning of the words used in the claims lack sufficient clarity to permit the scope of the claim to be ascertained from the words alone.” *Teleflex, Inc.*, 299 F.3d at 1325. For example, “[a] claim interpretation that excludes a preferred embodiment from the scope of the claim ‘is rarely, if ever, correct.’ ” *Globetrotter Software, Inc. v. Elam Computer Group Inc.*, 362 F.3d 1367, 1381 (Fed. Cir. 2004) (quoting *Vitronics Corp.*, 90 F.3d at 1583). But, “[a]lthough the specification may aid the court in interpreting the meaning of disputed language in the claims, particular embodiments and examples appearing in the specification will not generally be read into the claims.” *Constant v. Advanced Micro-Devices, Inc.*, 848 F.2d 1560, 1571 (Fed. Cir. 1988); *see also Phillips*, 415 F.3d at 1323.

Although “less significant than the intrinsic record in determining the legally operative meaning of claim language,” the Court may rely on extrinsic evidence to “shed useful light on the relevant art.” *Phillips*, 415 F.3d at 1317 (quotation omitted). Technical dictionaries and treatises may help the Court understand the underlying technology and the manner in which one skilled in the art might use claim terms, but such sources may also provide overly broad definitions or may not be indicative of how terms are used in the patent. *Id.* at 1318. Similarly, expert testimony may aid the Court in determining the particular meaning of a term in the pertinent field, but “conclusory, unsupported assertions by experts as to the definition of a claim term are not useful.” *Id.* Generally, extrinsic evidence is “less reliable than the patent and its prosecution history in determining how to read claim terms.” *Id.*

I. Claim Construction of Disputed Terms

1. “copy protecting a software application” (’569 Patent, claim 16)

| Blue Spike’s Proposed Construction | Defendants’ Proposed Construction |
|---|---|
| Plain and ordinary meaning | “protecting software from unauthorized copying” |

The parties’ dispute over this term was narrowed during the briefing and hearing. Blue Spike, relying on the specification, contends that the term would be understood by a person of ordinary skill in the art to include protecting against unauthorized copying, use, unintended use, tampering, attempts at memory capture, object code analysis, unintended modification, and attacks at disabling the system. Docket No. 174 at 5:4–8 (relying on the ’569 Patent at 2:23–32, 3:37–40, 7:16–17). Defendants respond that without construction, a juror could interpret this term to be about general computer security, rather than limiting the invention to aspects of preventing unauthorized copying and use of unauthorized copies. Docket No. 174 at 8:4–21

Given the confusion that the parties had in understanding scope of the term, jurors likely will have similar difficulty, and a construction is warranted. At the hearing, the parties agreed to the construction “protecting a software application from unauthorized copying or the use of unauthorized copies.” Docket No. 174 at 10:22–11:1.

Accordingly, the Court construes “copy protecting a software application” as “protecting a software application from unauthorized copying or the use of unauthorized copies.”

2. “during execution of the software application” (’569 Patent, claim 16)

| Blue Spike’s Proposed Construction | Defendants’ Proposed Construction |
|---|--|
| Plain and ordinary meaning | “while the software application is running after it has been loaded” |

The dispute is whether executing software is limited to running the software program (as Defendants argue) or whether executing software can include loading of software program as

well (as Blue Spike argues). Defendants argue claim 16 distinguishes between “loading” in the preamble and “executing” in the body. Docket No. 148 at 20. Defendants rely on the preamble to show that the software application is first loaded, and only then the code resources are intermittently relocated *during execution* of the software. *Id.* at 20–21. Defendants also argue the specification treats loading and executing differently. *Id.* at 21 (relying on ’569 Patent at 7:30–34, to argue that the “invention sought to rearrange code resources while the application was running, *i.e.* during execution time as opposed to system time”). Finally, Defendants cite to extrinsic evidence for the general proposition that loading and executing are separate and that software programs must first be loaded into memory before execution. *Id.* at 22.

To show the difference between execution and loading, Defendants provided an example of launching an app on one’s phone, and then moving that app to the background. Docket No. 174 at 17:20–18:4. Defendants argue that when the app is idle, then it is not running. *Id.* at 17:17–19 (“In the background there may be a few things happening, but there's no actual execution happening at that point. And then you flip to your active app, and that's the one that's executing.”).

Blue Spike argues for the plain and ordinary meaning, relying on the Fifth Edition of the Microsoft Computer Dictionary (2002) for the proposition that “execution implies loading machine code of the program into memory and then performing the instructions.” Docket No. 143 at 11, Ex. 3. Blue Spike also criticizes Defendants for “unnecessarily limit[ing] the term ‘execution’ to exclude the loading phase of execution.” *Id.* At the hearing, Blue Spike argued that execution should include loading time, and offered the example of double-clicking on a program, which leads to the operating system both loading and executing the program.

“THE COURT: So just -- just loading the program without running the program would be encompassed within ‘execution of the software application’? MR.

ANDERSON: That's right, Your Honor. Once you click on and execute the software, it loads and then it -- and executes. And so loading is necessarily a part of the execution phase.”).

Docket No. 174 at 12:11–24.

Defendants are correct that the '569 Patent treats loading and executing differently, that loading precedes execution, and that the code resources are intermittently relocated during software execution. For example, loading is the act of “transfer[ring] a program from a storage device into a computer’s memory” and executing is “run[ning] a program, carry[ing] out a command, or perform[ing] a function.” Docket No. 148, Ex. D, Dictionary of Computer Words, Am. Heritage Dictionaries Revised Ed., at 94, 160 (1995). However, the claim does not require that loading be completed before a program is executed. Likewise, the '569 Patent does not exclude “idle” time from “execution time.” That is, referring to Defendants’ app example, just because the application is in the background or waiting for an input from a user does not necessarily mean the app is not executing. As acknowledged by Defendants, there may still be “a few things happening.” Docket No. 174 at 18:1–4. As long as the app has started to perform instructions, it could potentially read onto this term.

Accordingly, the Court construes “during execution of the software application” as “while the software application is running.”

3. “intermittently relocating” ('569 Patent, claim 16)

| Blue Spike’s Proposed Construction | Defendants’ Proposed Construction |
|---|--|
| Plain and ordinary meaning | “intentionally shuffling at random or pseudo-random intervals” |

Blue Spike contends that both “intermittently” and “relocating” are readily understood by one of skill in the art. Docket No. 143 at 10. Blue Spike relies on the Oxford Dictionary’s definition of “intermittent” as being “not continuous or steady.” *Id.* at 9. Blue Spike submits

this conforms to Claim 16 which requires the location of a code resource in memory to be determined and then the code resource is “intermittently” relocated to a different location in memory. *Id.* Blue Spike also argues that Defendants improperly rely upon a single passage of the specification. *See* ’569 Patent at 8:4–5 (stating that a memory scheduler “can be called periodically, or at random or pseudo-random intervals”). According to Blue Spike, this is an example of how code resources may be intermittently shuffled and there is no indication the inventors intended to limit their invention by those terms. Docket No. 143 at 10.

Defendants argue that Blue Spike has failed to identify what the plain and ordinary meaning is. Defendants contend that in prosecution, the examiner questioned what was meant by “intermittently relocating” and, the patentee responded:

Applicants respectfully direct the Examiner’s attention to the disclosure at [Col.8:1-19] of the Specification, which describes an embodiment of the present invention wherein the code resources are “shuffled” in memory at periodic, random or pseudo-random intervals. *See also* [Col. 3:38-41 of the] Specification, (“Attempts to tamper or ‘patch’ substitute code resources can be made highly difficult by randomizing the location of said resources in memory on an intermittent basis . . .”). Given the foregoing, the Applicant respectfully submits that the language of claim[] 28 would be clear to a person of ordinary skill in the art.

Docket No. 148 Ex. K, Response to Office Action dated June 26, 1997 at 7. Defendants argue that the patentee, therefore, relied upon the specific portions of the specification to articulate the meaning of the term. Defendants also contend their construction is consistent with an object of the invention that is intended to prevent someone from copying a “snapshot” of the code located in memory by “intentional shuffling” code resources during execution time at “random or pseudo-random intervals”. Docket No. 148 at 19 (citing ’569 Patent at 7:41–44). This also conforms to the specification statement:

Under the present invention, the application contains a special code resource which knows about all the other code resources in memory. During execution time, this special code resource, called a “memory scheduler,” can be called periodically, or at random or pseudo random intervals, at which time it intentionally shuffles the other code resources randomly in memory, so that someone trying to analyze snapshots of memory at various intervals cannot be sure if they are looking at the same code or organization from one “break” to the next.

'569 Patent at 8:1–10.

As to the prosecution history, Blue Spike notes that the citation states “the code resources are ‘shuffled’ in memory at periodic, random or pseudo-random intervals.” Docket No. 148 Ex. K, Response to Office Action dated June 26, 1997 at 7. Blue Spike argues that the key portion of the passage and the larger passage do not include “intentional,” and Defendants have failed to include in their construction the word “periodic” which is in the passage.

At the hearing, it became clear that the primary dispute is whether the code resources must be shuffled more than a single time. Docket No. 174 at 24:8–25:3 and 25:15–18. Looking at the cited prosecution history and specification, there is no clear disclaimer or disavowal that excludes a single shuffle. *See Home Diagnostics, Inc. v. LifeScan, Inc.*, 381 F.3d 1352, 1358 (Fed.Cir.2004) (“Absent a clear disavowal in the specification or the prosecution history, the patentee is entitled to the full scope of its claim language.”). However, the prosecution history does show that “intermittently relocating” has some inherent ambiguity. The patentee addressed that ambiguity by stating the term means “ ‘shuffled’ in memory at periodic, random or pseudo-random intervals.” Docket No. 148 Ex. K, Response to Office Action dated June 26, 1997 at 7. The patentee’s statement could include multiple shuffles but does not necessarily exclude a single shuffle. For example, a shuffle at a random time would be consistent with the patentee’s statement. Defendants further clarified that “intentional” does not require user initiated relocating or shuffling, addressing Blue Spike’s concern on this point. Docket No. 148 at 19.

Accordingly, the Court construes “intermittently relocating” as “intentionally shuffling at periodic, random or pseudo-random intervals.”

4. “a software application” (’569 Patent, claim 16)

| Blue Spike’s Proposed Construction | Defendants’ Proposed Construction |
|------------------------------------|---|
| Plain and ordinary meaning | “a software program run by an operating system” |

The parties dispute whether an operating system is a software application. Blue Spike argues that the specification treats the terms “computer program” and “software application” as synonymous. Docket No. 143 at 5 (relying on ’569 Patent, at 3:44–45). According to Blue Spike, this means that operating systems, which no one disputes is an executable computer program, is also a software application. *Id.* Defendants argue that the patentee used the term “software application” in claim 16, but used the more broad term “computer software” in claim 1. Docket No. 148 at 10. Defendants submit that because “different words or phrases used in separate claims are presumed to have different meanings, the term ‘software application’ cannot encompass all ‘computer software’ as Blue Spike contends.” Docket No. 148 at 10. Defendants also rely on the specification’s disparate treatment of “application code” and “operating system code” to show that “software application” does not encompass operating systems. Docket No. 148 at 11 (citing ’569 Patent at 7:30–34, stating that system time is when the “operating system code, not application code, is running”).

The ’569 Patent distinguishes between “application” and “operating system.” 7:30-34, 8:1-4 (and to a lesser extent all of 8:1–11). It also distinguishes between “software applications” and “computer software.” ’569 Patent 8:30 (claim 1 preamble stating “a method for copy protection of computer software”) and 10:9 (claim 16 preamble stating “a method for copy protecting a software application”). Considering the Patent as a whole, it is more accurate to say

that the intrinsic record treats applications as a subset of a computer programs rather than equating the two. *See Phillips*, 415 F.3d at 1314 (the specification is usually “the single best guide to the meaning of a disputed term”). This reading is also consistent with the extrinsic sources cited by both parties. Despite the flaws in these sources—Blue Spike’s citation to a dictionary that predates the patent by 17 years and Defendant’s citation to dictionaries that are several years after the patent was issued—the sources show that though operating systems and software applications are related, operating systems are used to facilitate running software applications. *See, e.g.*, Docket No. 174 at 35:21–36:7 (citing to a 1972 IBM Reference Library that applications are distinct from operating systems); Docket No. 148-7 Ex. G, Douglas A. Downing, Michael A. Covington, Melody Mauldin Covington, *Dictionary of Computer and Internet Terms*, 22 (Barron’s Educational Series, Inc., 6th ed. 1998) (contrasting applications programs to operating systems); Ex. H, *Microsoft Computer Dictionary* 378 (Microsoft Press 5th ed. 2002) (“The operating system is the foundation software on which applications depend.”).

Accordingly, the Court construes “software application” as “a software program run by an operating system.”

5. “a plurality of executable code resources” / “executable code resources” (’569 Patent, claim 16)

| Blue Spike’s Proposed Construction | Defendants’ Proposed Construction |
|---|--|
| Plain and ordinary meaning | “(a plurality of) compiled, indivisible functions or procedures of the software application” |

At the hearing, the Court proposed “a plurality of one or more compiled functions or procedures.” Defendants agreed to this proposed construction as given. Blue Spike agreed with this construction with a minor change. Blue Spike requested that the Court only construe the term “executable code resources” (as opposed to “a plurality of executable code resources”) to mean “one or more compiled functions or procedures.” Docket No. 174 at 37:4–15. That way,

according to Blue Spike, the word “plurality” remains in the claim language rather than in the Court’s construction.

The parties do not dispute the ultimate meaning of this term, but only disagree on what would be the least ambiguous to a jury. *See* Docket No. 174 at 37:20–38:2, 38:21–39:14. The term “plurality” appears in the claim, is readily understood by a jury, and not subject to any dispute from the parties. It does not need to be construed.

Accordingly, the Court construes “executable code resources” as “one or more compiled functions or procedures.”

6. “a computer system” / “the computer” (’569 Patent, claim 16)

| Blue Spike’s Proposed Construction | Defendants’ Proposed Construction |
|---|--|
| Plain and ordinary meaning | No construction necessary |

At the hearing, the parties resolved their primary dispute concerning this term. Docket No. 174 at 39:23–40:2 (Defendants stating they believe the term does encompass the types of devices listed in the specification at 5:31–39). The parties further agreed to the following: “no construction necessary, plain and ordinary meaning.”

Accordingly, for “a computer system” / “the computer,” the Court will note “no construction necessary, plain and ordinary meaning.”

SIGNED this 16th day of May, 2016.


ROBERT W. SCHROEDER III
UNITED STATES DISTRICT JUDGE

APPENDIX A

United States Patent Number 5,745,569:

| Claim Term | Court's Construction |
|--|---|
| “copy protecting a software application” claim 16 | “protecting a software application from unauthorized copying or the use of unauthorized copies” |
| “during execution of the software application” claim 16 | “while the software application is running” |
| “intermittently relocating” claim 16 | “intentionally shuffling at periodic, random or pseudo-random intervals” |
| “a software application” claim 16 | “a software program run by an operating system” |
| “executable code resources” claim 16 | “one or more compiled functions or procedures” |
| “a computer system” / “the computer” claim 16 | no construction necessary, plain and ordinary meaning |