

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

UBISOFT, INC. and SQUARE ENIX, INC.,
Petitioner,

v.

UNILOC 2017 LLC,
Patent Owner.

Case IPR2017-01291
Patent 6,728,766 B2

Before SALLY C. MEDLEY, MIRIAM L. QUINN, and
JESSICA C. KAISER, *Administrative Patent Judges*.

QUINN, *Administrative Patent Judge*.

FINAL WRITTEN DECISION
35 U.S.C. § 318(a); 37 C.F.R. § 42.73(a)

I. INTRODUCTION

In this *inter partes* review, instituted pursuant to 35 U.S.C. § 134, Petitioner, as listed in the caption above, challenged the patentability of claims 1, 3, 7, 9, 13, and 15 of U.S. Patent No. 6,728,766 B2 (“the ’766 patent”), owned by Uniloc 2017 LLC, (“Patent Owner” or “Uniloc”).¹ This Final Written Decision is entered pursuant to 35 U.S.C. § 318(a) and 37 C.F.R. § 42.73. For the reasons discussed below, Petitioner has not shown by a preponderance of the evidence that claims 1, 3, 7, 9, 13, and 15 of the ’766 patent are unpatentable.

A. PROCEDURAL HISTORY

Petitioner filed a Petition to institute *inter partes* review of claims 1, 3, 7, 9, 13, and 15 of the ’766 patent. Paper 4 (“Pet.”). Patent Owner filed a Preliminary Response. Paper 7 (“Prelim. Resp.”). On November 1, 2017, we instituted *inter partes* review as to claims 1 and 3 only.

After institution, Patent Owner filed a Patent Owner Response. Paper 13 (“PO Resp.”). Petitioner filed a Reply. Paper 15 (“Reply”). On April 24, 2018, the Supreme Court held that a decision to institute under 35 U.S.C. § 314 may not institute on fewer than all claims challenged in a petition. *SAS Inst., Inc. v. Iancu*, 138 S. Ct. 1348, 1355 (2018). Consistent with SAS, we modified our Institution Decision to institute on all of the challenged claims, including claims 7, 9, 13, and 15, on the grounds

¹ Patent Owner’s Mandatory Notice filed on May 10, 2017 identified both Uniloc USA, Inc. and Uniloc Luxembourg S.A. as the patent owners in this case. Paper 5, 1. Updated Mandatory Notices, however, indicate that the patent owner entity changed to Uniloc Luxembourg S.A., and most recently to Uniloc 2017 LLC. Papers 17, 25.

presented in the Petition and authorized Petitioner to respond to these modifications to our Institution Decision. Paper 18, 3 (stating that “Petitioner’s Institution Response Brief is for identifying matters that Petitioner believes the Board misapprehended or overlooked, or otherwise erred in its institution decision discussing the newly-added claims” (emphasis omitted)). Petitioner then filed a Petitioner’s Institution Response Brief. Paper 21 (“Pet. Inst. Resp.”).

We heard oral arguments on August 7, 2018. A transcript of the hearing is in the record. Paper 24 (“Tr.”).

B. RELATED MATTERS

Petitioner identifies the ’766 patent as the subject of two district court cases pending in the U.S. District Court for the Eastern District of Texas (Case No. 2:16-cv-00397-RWS and Case No. 2:16-cv-00872-RWS). Pet. 32; Ex. 1006; Ex. 1007.

C. THE ’766 PATENT (EX. 1001)

The ’766 patent is entitled “Methods, Systems and Computer Program Products for License Use Management on a Network.” Ex. 1001, (54). The ’766 patent relates in particular to application program management on a computer network. *Id.* at 1:22–24. According to the ’766 patent, control over software, such as application programs, is a challenge with respect to “maintaining proper licenses for existing software and deploying new or updated applications programs across the network.” *Id.* at 1:45–57. In particular, the ’766 patent states that “[a] distributed network environment with a plurality of client stations and a plurality of different users accessing the applications from different clients increases the challenge associated

with managing license use to [ensure] compliance with limitations established by software designers.” *Id.* at 3:28–32. According to the ’766 patent, management of license use for a network is provided as follows:

License management policy information for a plurality of application programs is maintained at a license management server. Requests are received at the license management server for a license availability of a selected one of the plurality of application programs from a user at a client. The license management server determines the license availability for the selected one of the plurality of application programs for the user based on the maintained license management policy information and provides an unavailability indication to the client responsive to the selection if the license availability indicates that a license is not available for the user or an availability indication if the licensed availability indicates that a license is available for the user.

Id. at 5:39–52.

D. ILLUSTRATIVE CLAIM

Challenged claims 1, 7, and 13 of the ’766 patent are independent. Illustrative claim 1 is reproduced below.

1. A method for management of license use for a network comprising the steps of:
 - maintaining license management policy information for a plurality of application programs at a license management server, the license management policy information including at least one of a user identity based policy, an administrator policy override definition or a user policy override definition;
 - receiving at the license management server a request for a license availability of a selected one of the plurality of application programs from a user at a client;
 - determining the license availability for the selected one of the plurality of application programs for the user based on the maintained license management policy information; and
 - providing an unavailability indication to the client

responsive to the selection if the license availability indicates that a license is not available for the user or an availability indication if the licensed availability indicates that a license is available for the user.

Ex. 1001, 14:64–15:16.

E. ASSERTED REFERENCE AND GROUND OF UNPATENTABILITY

Petitioner asserts one ground of unpatentability based on the anticipation of claims 1, 3, 7, 9, 13, and 15 by U.S. Patent No. 5,758,069, issued to Olsen (Ex. 1002, “Olsen”). Pet. 2.

I. ANALYSIS

A. CLAIM CONSTRUCTION

In an *inter partes* review, claim terms in an unexpired patent are interpreted according to their broadest reasonable construction in light of the specification of the patent in which they appear. 37 C.F.R. § 42.100(b) (2012); *Cuozzo Speed Techs., LLC v. Lee*, 136 S. Ct. 2131, 2142–46 (2016). Consistent with that standard, claim terms also are given their ordinary and customary meaning, as would be understood by one of ordinary skill in the art in the context of the entire disclosure. *See In re Translogic Tech., Inc.*, 504 F.3d 1249, 1257 (Fed. Cir. 2007).

In our Institution Decision, we analyzed two terms drafted in means-plus-function format: “means for maintaining” and “computer readable program code means.” Paper 9, 6–9 (“Dec.”). We recognized that construing a means-plus-function limitation requires first defining the particular function of the limitation and then identifying the corresponding structure for that function in the specification. *Golight, Inc. v. Wal-Mart*

Stores, Inc., 355 F.3d 1327, 1333–34 (Fed. Cir. 2004). Our claim construction analysis follows.

1. Means for Maintaining

The parties agreed that the function for the claim 7 term is “maintaining license management policy information for a plurality of application programs at a license management server, the license management policy information including at least one of a user identity based policy, an administrator policy override definition or a user policy override definition.” Pet. 3; Prelim. Resp. 5. With regard to corresponding structure, Petitioner identified “a database and equivalents thereof.” Pet. 3.

Patent Owner argued that identifying a database in general is misleading. Prelim. Resp. 5–6. Patent Owner further pointed out the claim language describing that the “means must reside ‘at a license management server’” and that the server *stores* license use management policy information *in a database*. *Id.* at 5. Consequently, Patent Owner argued that Petitioner does not tie the structure recited in the claim with the database alleged to perform the function. *Id.* at 5–6.

We agreed with Patent Owner that the claim language plainly states that “maintaining” occurs “at a license management server,” not a database as Petitioner proposed. Ex. 1001, 15:48–50. The specification supports that position, as it describes the server system storing license use management policy information in hierarchical centralized preference database 208. *Id.* at 12:50–52. That description, however, refers only to “storing” the information in a specific type of database, which may be useful while the server “maintains” licenses across a distributed network. We determined, thus, that Petitioner failed to tie the “storing” to the entire function of

“maintaining” the information, as recited. A structure disclosed in the specification qualifies as “corresponding” structure only if the specification or prosecution history clearly links or associates that structure to the function recited in the claim. *B. Braun Med. v. Abbott Labs.*, 124 F.3d 1419, 1424 (Fed. Cir. 1997). The “clear linkage or association” in the specification of the structure to the function recited in the claim is determined based on the understanding of an artisan of ordinary skill. *See AllVoice Computing PLC v. Nuance Commc’ns, Inc.*, 504 F.3d 1236, 1242 (Fed. Cir. 2007).

Further, we found that the portions of the specification Petitioner relied on do not convey that database 208 performs the function of maintaining the recited information. Dec. 7–8. Petitioner proffered no other evidence or testimony regarding this issue. Consequently, we were not persuaded that Petitioner showed that a person of ordinary skill would understand that a database, *alone*, is clearly linked to “maintaining license management policy information for a plurality of application programs at a license management server” *Id.* at 7.

Instead, for purposes of institution, we were persuaded that Patent Owner had shown sufficiently that the corresponding structure for the “means for maintaining” is, at a minimum, a processor executing algorithms for performing the function. Patent Owner argued that the specification discloses a processor for executing instructions because the specification, for example, describes all the flowcharts depicted in Figures 2–5 as follows:

It will be understood that each block of the flowchart illustrations, and combinations of blocks in the flowchart illustrations, can be implemented by computer program instructions. These program instructions may be provided to a processor to produce a machine, such that the instructions which execute on the processor create [a] means for

implementing the functions specified in the flowchart block or blocks. The computer program instructions may be executed by a processor to cause a series of operational steps to be performed by the processor to produce a computer implemented process such that the instructions which execute on the processor provide steps for implementing the functions specified in the flowchart block or blocks.

Ex. 1001, 7:26–40.

Patent Owner further pointed out that the specification does disclose algorithms for maintaining license management policy information. Prelim. Resp. 6. For example, the specification states that license use “is managed by setting certain policies, such as the limit of the number of users, whether crossing the limit of users is allowed or not and how users are counted (simultaneous number logged on, total number of clients that can use the application, and so on).” Ex. 1001, 12:37–46. The specification also describes that the license management policies have default values that are customized by applying administrator overrides (*id.* at 12:42–44) and user license policy preferences (*id.* at 12:45–47).

Therefore, we determined in our Institution Decision that the structure corresponding to the “means for maintaining” is, at a minimum, a processor programmed to carry out the algorithms for maintaining license management policy information for a plurality of application programs at a license management server, where the algorithms are setting policies such as limiting the number of users, whether crossing the limit of users is allowed or not, and how users are counted. Dec. 8. Patent Owner stated during oral argument that it believes the Board’s claim construction is correct. Tr. 67:8–15.

During trial Petitioner was given an opportunity to argue its position on claim construction in light of the Board’s preliminary construction discussed above. Pet. Inst. Resp. 7–9. Petitioner contends that the ’766 patent does not disclose algorithms and the claim is, therefore, indefinite. *Id.* In particular, Petitioner argues that those certain algorithms are only examples and are not detailed enough to be proper algorithms. Pet. Inst. Reply 8. We are not persuaded by this argument. An algorithm may be expressed in any understandable terms including as a mathematical formula, in prose, in a flow chart, or “in any other manner that provides sufficient structure.” *Finisar Corp. v. DirecTV Group, Inc.*, 523 F.3d 1323, 1340 (Fed. Cir. 2008); *see also Intel Corp. v. VIA Techs., Inc.*, 319 F.3d 1357, 1366 (Fed. Cir. 2003) (stating that structure disclosed is not inadequate merely because the written description omits details of circuitry or computer code). Here, the algorithms are stated in prose, and provide sufficient detail to apprise a person of ordinary skill in the art on what is needed to implement the computer code running at the server. Petitioner has directed us to no evidence to the contrary.

Accordingly, we conclude that the corresponding structure for the “means for maintaining” is unmodified from the structure identified in the Institution Decision: a processor programmed to carry out the algorithms for maintaining license management policy information for a plurality of application programs at a license management server, where the algorithms are setting policies such as limiting the number of users, whether crossing the limit of users is allowed or not, and how users are counted.

2. Computer Readable Program Code Means

Claim 13 recites “a computer readable storage medium having computer-readable program code means embodied in said medium” and “computer readable program code means for maintaining license management policy information,” among several additional computer readable program code means limitations. This claim language utilizes the word “means,” which triggers a rebuttable presumption that § 112, ¶ 6 applies. *Williamson v. Citrix Online*, 792 F.3d 1339, 1349 (Fed. Cir. 2015). If the “computer readable program code means” terms recited in claim 13 are indeed means-plus-function terms, then the Petition is deficient, as it fails to present the required claim construction analysis.

37 C.F.R. § 42.104(b)(3). In other words, in light of the presumption that applies to a claim reciting “means,” we look for an explanation by Petitioner, who has the burden of showing in its Petition how the challenged claim is to be construed and how that construed claim is unpatentable. *Id.* Petitioner proffers no analysis of the scope of these claim terms and, more particularly, why or how the presumption is rebutted.²

Because neither party rebutted the presumption that § 112, ¶ 6 applies to the “computer readable program code means,” we address this term as a means-plus-function term. Accordingly, the limitation “computer readable program code means for maintaining license management policy information,” at a minimum, encompasses the embodied algorithms linked

² We recognize that in discussing the prior art, the Petition maps the “computer readable program means” to a license service provider software. Pet. 27. This showing, however, is insufficient to explain a rebuttal of the presumption borne by the use of “means.”

to the function “maintaining license management policy information.”

Those algorithms, following our analysis for the term “means for maintaining” discussed above are, at a minimum: setting policies such as limiting the number of users, whether crossing the limit of users is allowed or not, and how users are counted.

B. ANTICIPATION BY OLSEN

Olsen is entitled “Electronic Licensing System.” Ex. 1002, [54]. It states by way of background that conventional licensing systems use a fixed set of licenses, and that license information is stored in a license database, along with information regarding which applications are in use and how many units are still available. *Id.* at 1:44–48. In related software licensing systems, the application has embedded code that requests a license from the server. *Id.* at 1:51–52. Typically, the server checks the database of licenses and grants the request, if the appropriate license is available. *Id.* at 1:53–54. As requests are received and licenses granted, the information is logged into a file to track the usage of the various applications. *Id.* at 1:55–57. However, when a license is not available at that server, the client contacts another server to find a license. *Id.* at 1:58–59. Thus, the client typically has the responsibility to obtain the licenses from various servers, and individual servers provide the licenses at the client’s requests. *Id.* at 1:59–62.

The problem with such systems is that the programs required at the client occupy a significant portion of the client’s memory resources. *Id.* at 2:1–4. And because the license information is stored in the client’s memory, the license information may be lost and difficult to reconstruct, if the client

crashes. *Id.* at 2:4–7. Another problem with conventional licensing systems is that they rely on the code embedded in the application to establish licensing attributes. *Id.* at 2:17–19. This code also establishes the licensing parameters from information it receives from the server, which means that the “behavior” of the license is not established until after the request is made and the license is obtained. *Id.* at 2:19–23. The code in the application also must be revised in order to change the licensing terms. *Id.* at 2:24–26.

The Olsen system solves these problems by, first, providing license information stored in a distributed database among several servers and a license service provider (“LSP”) that constructs license certificate objects associated with a client request to adjust the information in the database and reflect the granting of requested licenses. *Id.* at 2:34–35, 48–54. Second, the client is equipped with a suitable library of application programming interfaces (“APIs”) for acquiring and managing licenses. *Id.* at 2:37–39. The client “merely bundles the arguments for a license request and transmits them to the server.” *Id.* at 2:57–58. For example, the client assembles a request for the license and provides the license criteria: publisher, product, version, and number of license units. *Id.* at 2:40–42. Other information may be provided, such as the user’s name. *Id.* at 2:42–43. The server, upon receipt of the request, searches the local database and passes the information to other servers to check their respective databases. *Id.* at 2:44–48. Thus, “most of the license transactions occur at the servers.” *Id.* at 2:56–57. Finally, because the license information is stored in the database, and the client only stores “the license handle associated with the license request,”

the license information may be reconstructed by finding in the database the particular transaction associated with the license handle. *Id.* at 2:61–67.

1. *User Versus Client Issue*

At the heart of the challenged claims lies a distinction of “user” versus “client.” For instance, claim 1 recites that the license management server receives “a request for a license availability of a selected one of the plurality of application programs from *a user at a client*.” Ex. 1001, 15:5–7 (emphasis added). Claim 1 further requires that the license availability is determined *for the user* based on the maintained license management policy information. *Id.* at 15:8–11. Finally, claim 1 recites “providing an unavailability indication *to the client* . . . if the license availability indicates that a license is not available *for the user* or an availability indication if the licensed availability indicates that a license is available *for the user*.” *Id.* at 15:12–16 (emphases added). Thus, the plain and ordinary meaning of the claim language requires that *a user at a client* request license availability, and that the license availability is determined *for the user*, with an indication of availability or unavailability provided *to the client*.

The user is, thus, an individual interfacing with the server via a client, which is hardware, such as a computing device. *See, e.g., id.* at 6:5–9 (“It further provides an essentially hardware transparent ability for an individual user to interface to an on-demand server supported client station while maintaining the user’s personal preferences for each application program.”); *id.* at 3:67–4:2 (“An icon is displayed on the screen of the client station through which a user may initiate execution of the application program.”); *id.* at 4:14–18 (“A profile manager program on the on-demand server provides a user interface and maintains profile information for *each client*”).

(*hardware related*) and *user (preferences)* for all managed configurable application programs available on the on-demand server.” (emphases added)). The claim language is, therefore, particular about the license availability being determined on a user basis, instead of for the client. The difference is important because Patent Owner contends that Olsen’s licensing system determines and assigns licenses to clients, not users, and therefore, Olsen does not request or determine license availability *for a user*, as required by the claims. PO Resp. 5–10. We agree with Patent Owner.

Olsen describes a method that allows *a user* to choose a license among options from a list (Ex. 1002, 10:45–51), determines license units available by reviewing *user information* associated with the request (*id.* at 2:38–43, 11:40–46), accesses license records using the *user’s login information* (*id.* at 11:32–34); and provides an administrator the option to assign a license to an individual, machine, group, container, or other selected *users* (*id.* at 8:29–31). Petitioner has relied on these disclosures in particular to argue that Olsen’s “license availability request is user-specific.” Reply 4–5. These disclosures paint the picture that a user is involved in requesting the license and providing information (user information) that is reviewed in determining whether to grant the license. But mere involvement by the user or user information in the process of license availability determination is insufficient to meet the claim. We look for evidence that the request and determination of license availability are directed to answering the question of whether there is a license available *for the user*.³

³ Petitioner does not dispute the scope of “request for license availability” beyond its plain and ordinary meaning. See Tr. 20:12–19 (Petitioner stating

Olsen describes determining whether a license unit is available for the client, even though a user or user information appears to be involved in the process. For example, although the request for a license includes the user's login information and the determination involves reviewing the user information, Olsen "determines whether the license units corresponding to the license record *are available to the requesting client.*" Ex. 1002, 11:40–46 (emphasis added). Further "[t]o consume the units, [Olsen] checks the license information for any existing assignments, [and i]f an assignment exists on the license certificate, it performs a security equivalency check to *determine whether the requesting client 106 is among those assigned to the license certificate.*" *Id.* at 12:13–18 (emphasis added). The license units may be reinstated and become available "for use by *other clients*, if appropriate." *Id.* at 12:56–58 (emphasis added). And Olsen describes that when "client 106 no longer requires the license units, the application transmits a release notification to LSP 110." *Id.* at 12:64–65.

Therefore, based on the above disclosures, we find that Olsen describes determining license availability for a client. Not only does Olsen plainly states this at column 11 lines 40 to 46, but Olsen also explains that if the client is assigned to a license certificate (where a security equivalency check ensures proper assignment), the license certificate object consumes the license for the client. *Id.* at 12:13–18.

that "request for license availability does not require any construction" and that the plain language says, "you're making a request to determine if a license available--is available.").

As stated above, the claims require, however, that the license availability request is *from a user* and that the license availability is determined for *that user*. The best disclosures in Olsen for supporting the contention that Olsen's request is user-specific come from the descriptions of the user information and the assignment of a license by an administrator. *See id.* at 8:29–37, 11:32–34, 11:40–46. Petitioner argued that it is irrelevant that Olsen's license response is always going back to a client, as the '766 patent specification also describes this. Tr. 14:10–17. According to Petitioner, if the license request considers user information and “ties the request to that user by looking at the licensing record with the user information and doing that comparison . . . that request and that license is tied to the user and satisfies the limitations” of the claim. *Id.* at 15:5–10. Petitioner argued that such a request is “now from a user, whether the client actually does the transmitting or not, and that request is responsive to a user and tied to that user, whether the client is receiving the license back or not.” *Id.* at 15:13–17. The argument, however, is not commensurate with the scope of the claim. The claim requires that, in addition to the request for a license availability, the method determines the license availability *for the user*. If Olsen's request includes the user information, there may be evidence tending to show that Olsen receives a request from a user for license availability. But this same evidence does not show that Olsen *determines license availability for the user*.

We agree with Petitioner that Olsen's user chooses the application for which a license is needed (Ex. 1002, 10:46–48) and Olsen's client assembles a request (*id.* at 10:44–45, 10:56–67). The request seems to be associated with a user because Olsen describes the LSP transmitting the request and

granting a corresponding license handle, and storing this handle in the transaction database “along with other information associated with the license request.” *Id.* at 11:1–11. The information includes the user’s NDS distinguished name. *Id.* at 11:13–16. Thus, by the time Olsen stores the license handle, the user’s login information is associated with the license request.

After this step of storing the information and the license handle, however, Petitioner’s evidence falls short of the preponderance of the evidence standard. The Petitioner bears the burden of proving a proposition of unpatentability by a preponderance of the evidence. 35 U.S.C. § 316(e). The burden of showing something by a preponderance of the evidence requires the trier of fact to believe that the existence of a fact is more probable than its nonexistence before the trier of fact may find in favor of the party who carries the burden. *Concrete Pipe & Products of Cal., Inc. v. Construction Laborers Pension Trust for Southern Cal.*, 508 U.S. 602, 622 (1993).

Olsen’s user information/login information is used in two processes: (1) to access license records in the database in the process of creating the license certificate object (*id.* at 11:32–34); and (2) in the process of determining whether the requested license units are available to the client (*id.* at 11:40–46). Olsen describes the process of “actually obtaining” the license by first determining “whether all of the required license units are available.” *Id.* at 11:46–48. Olsen’s license certificate object does this by reviewing the policy attributes of the license, the user information associated with the request, any existing license assignments, and the raw number of units originally installed. *Id.* at 11:40–46. Thus, there is evidence of the

user information being part of the license availability determination. Olsen is clear, however, in stating that the license certificate object “suitably determines whether the license units corresponding to the license record *are available to the requesting client*.” *Id.* (emphasis added). Olsen describes the client as a “conventional network terminal, for example a personal computer” that is connected to the server. *Id.* at 5:65–6:1. The client of Olsen is not an individual, person, or virtual representation of a network user.

Therefore, we find that Olsen determines license availability for the client—not the user. No other passage from Olsen describes determining license availability *for* the requesting user. While user information may be used in determining the license availability for the client, Olsen, we find, does not describe what role the user information plays in the determination such that the license availability determination would be user-specific. Thus, the nonexistence of the user-specific determination is more probable than its existence. *See Concrete Pipe*, 508 U.S. at 622. And, therefore, Petitioner has not shown by a preponderance of the evidence that Olsen discloses “determining the license availability for the selected one of the plurality of application programs for the user based on the maintained license management policy information,” as required by claim 1.

2. “License Availability” Versus “Security Equivalency Check”

Our determination is further supported by Olsen’s description of what happens after determining the license availability for the client. Olsen performs an additional check before consuming the available licenses. Ex. 1002, 12:9–17. This check is a “security equivalency check to determine whether the requesting client 106 is among those assigned to the

license certificate (step 828).” *Id.* That is, the licenses may be available for the client, but Olsen checks further to ensure that the user *is authorized* to access the application program. Ex. 2001 ¶¶ 19–21. Olsen’s assignment to the license certificate was effected when the license certificates were created, allowing “users to an application for access.” Ex. 1002, 4:46–50, 5:20–25, 8:29–31. Olsen, thus, describes two different determinations: (1) determining license availability, and (2) determining authorization to access the application. Again, although Olsen suggests security assignments on a user-specific basis, Olsen describes the security check as verifying whether a particular user is authorized under the license certificate, and ultimately, is permitted to access the application. *Id.* at 8:31–33. In sum, Olsen determines the license availability *for the client*, and then checks whether the user at the requesting client has authorization to access the license certificate.

The parties disagree as to whether there are differences between determining license availability and performing a security check in Olsen. Patent Owner argued there is a distinction between these two determinations, during oral argument, in response to Petitioner’s Reply. Tr. 34:11–36:16. Petitioner’s Reply explains the disclosure of U.S. Patent No. 6,510,466 B1,⁴ which the ’766 Patent incorporates by reference, in the context of what constitutes a “license availability request.” Reply 6. During oral argument, Petitioner argued that a “license availability request” is simply a license

⁴ Ex. 1003. Petitioner also admits that the ’766 patent incorporates by reference the ’466 patent. Pet. 5. Petitioner also requested we take judicial notice of the ’466 patent disclosure. Tr. 77:22–78:4.

request that can be denied. Tr. 18:1–20:11. Petitioner also argued that in response to the client’s request to the LSP, Olsen’s determination is “just like in the ’766 patent.” *Id.* at 22:1–4. “If there are insufficient license[s]--license units located, a detailed error code is issued.” *Id.* According to Petitioner, the security check that Olsen performs is to check “the credentials that are supplied . . . to make sure it’s an authorized and appropriate user.” *Id.* at 22:10–14. Petitioner argued that, even if the security checks out, Olsen may deny the request if there are not sufficient license units available. *Id.* at 22:16–19. The arguments highlight Petitioner’s contention that the “security equivalency check” is part of Olsen’s license availability determination. *See* Pet. 16.

Patent Owner disagrees with Petitioner’s contention because there is a difference between a license availability and a security check. Patent Owner takes issue with Petitioner’s argument that if you can deny a request based on security, then what you have is a license availability check that is for a user. Tr. 34:20–22. Patent Owner then explained the differences between the two concepts with respect to the ’466 patent disclosure, particularly regarding Figure 7, reproduced below.

FIG. 7

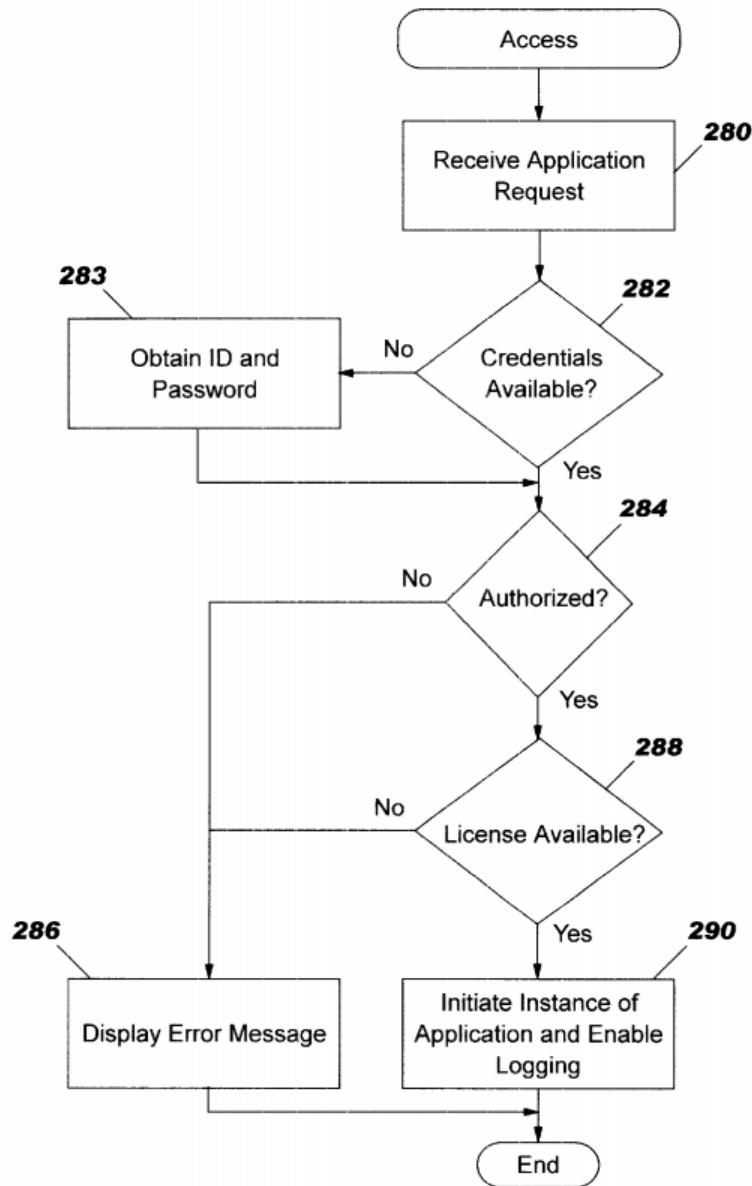


Figure 7 of the '466 patent depicts a flowchart of application access operations for an application program. Ex. 1003, 6:25–27, Fig. 7. According to Patent Owner, Figure 7 shows steps 284 and 288 as distinct steps in the process of requesting an application and receiving license requests from an application program. Tr. 35:18–36:7. At step 288, for

instance, if the requesting user is an authorized user for the requested application, the system accepts a license request from the application. *Id.* at 35:18–21. According to Patent Owner, the ’466 patent disclosure, which is part of the ’766 patent disclosure by incorporation,⁵ clarifies that user authorization and determining whether a license is available are distinct steps. *Id.* at 35:22–36:7 (arguing that at block 286, “[i]f no licenses are available, the system . . . provide[s] an error message display and stop[s] processing,” and distinguishing this step from the authorization step). The security check “is not a license availability check within the . . . context of the patent.” Tr. 38:7–20.

We agree with Patent Owner. The ’766 patent claims are directed to a license availability request and determination, with no indication that license availability would subsume a security check or determination that the user is actually authorized to use the application program. Further, the ’766 patent specification describes “authorization” distinctly from “availability.” For instance, the ’766 patent describes how *authorized* users may launch an application program. Ex. 1001, 5:11–22; *see also id.* at 8:26–34 (“[A]ll users may be considered authorized users” and “access to an application program may be limited to only specific users . . . [or] to only specific hardware locations regardless of user.”), 11:5–8 (describing that the server confirms the authorized status of the user), 4:18–20 (describing the profile manager program on the server which controls access to applications based on user authorization information). With regard to license management,

⁵ *See* Ex. 1001, 7:17–21 (incorporating by reference U.S. Patent Application No. 09/211,528, which became the ’466 patent).

however, the '766 patent describes that each time a user request instantiates an application, the license information is provided to the application. *Id.* at 13:7–10; *see also id.* at 13:36–38 (describing that a license request is received prior to initiating execution of an instance of the application responsive to a user request). The '766 patent describes the request for and determination of license availability without any mention or interplay with the authorization checks performed when actually accessing the application. *Id.* at 5:37–60; *see also id.* at 13:47–48 (describing that the license policy determinations are returned to the *initiated* instance of the application, implying that the user is already authorized); *id.* at 14:1–12 (describing that the license use may not need to be counted for some users but counted for others).

The '466 patent specification confirms these distinctions as Patent Owner described with regards to Figure 7. Incorporation by reference ““provides a method for integrating material from various documents into a host document . . . by citing such material in a manner that makes clear that the material is effectively part of the host document as if it were explicitly contained therein.”” *Cook Biotech Inc. v. Acell, Inc.*, 460 F.3d 1365, 1376 (Fed. Cir. 2006) (quoting *Advanced Display Sys., Inc. v. Kent State Univ.*, 212 F.3d 1272, 1282 (Fed. Cir. 2000)). Here, there is no dispute that the '466 patent is incorporated by reference in the '766 patent, and therefore, its disclosure is treated as if it were part of the '766 patent specification.

The '466 patent describes a database for controlling access or authorization to applications. Ex. 1003, 8:60–9:5. Before the user can use an application, the user logs in to the desktop and provides user identification, which the system uses to determine whether the user is

authorized to access the application. *Id.* at 15:19–41. The application program is customized to provide for license use management by “determining license availability before initiating execution of the application program.” *Id.* at 4:3–9; *see also id.* at 5:8–20. “If the requesting user is an authorized user for the requested application, the server system 22 accepts a license request from the application (block 288).” *Id.* at 16:43–47 (also sending error message if “no licenses are available”). The ’466 patent states that “[i]f a license is available, an instance of the requested application is executed.” *Id.* at 16:53–55. In sum, the ’466 patent performs first the user authorization determination, and separately performs the license availability determination. The language of the claims of the ’766 patent focus on the license availability determination, and the notification produced as a result of that determination. Although the claims do not preclude a user-authorization determination from occurring, to anticipate, Olsen must perform, at a minimum, the license availability determination for the user. But the license availability determination and the authorization determination are neither equal to each other nor co-extensive according to the disclosures of the ’466 and ’766 patents.

Thus, we find that the ’466 patent describes user authorization to access an application distinctly from the determination of whether a license is available for the user. Therefore, Olsen’s “security equivalency check” alone is not sufficient evidence of anticipation for the “determining license availability” step. Olsen’s security equivalency check determines whether the user is authorized to access the application because a user is assigned to a license certificate. Petitioner relied on this security equivalency check, in addition to Olsen’s license availability check, for the limitation of a license

availability determination for a user. Pet. 16–18. We determined above, that Olsen does not perform a license availability check for *the user*. And according to our discussion above, we determine that Olsen’s security equivalency check, alone, is insufficient to disclose the limitation.

At oral argument, Petitioner provided further explanation concerning the ’466 patent disclosure in comparison with Olsen. Tr. 81:12–84:2. The explanation is unpersuasive. Again, although the ’466/’766 patents and Olsen describe the two determinations (license availability and security checks), Olsen describes the license availability for *the client only*, with no disclosure of the determination being made *for the user*. Olsen’s security check is performed on a user-specific basis, but this check alone cannot supplant the requirement of determining whether a license is available for the user. And we have determined that Olsen does not disclose this determination.

3. Conclusion as to all Challenged Claims

The independent challenged claims all recite the “determining” step in one form or another. Specifically, claim 7 recites the “means for determining the license availability . . . for the user.” Ex. 1001, 15:57–60. In addition, claim 13 recites a “computer readable program code means for determining the license availability . . . for the user.” *Id.* at 16:46–50. For the same reasons as stated for claim 1, Petitioner has not shown that Olsen anticipates any of the further challenged claims.

II. CONCLUSION

Petitioner has failed to prove by a preponderance of the evidence that the challenged claims are unpatentable as anticipated over Olsen.

III. ORDER

It is ORDERED that claims 1, 3, 7, 9, 13, and 15 of the '766 patent have not been shown to be unpatentable; and

FURTHER ORDERED that, because this is a Final Written Decision, parties to the proceeding seeking judicial review of the decision must comply with the notice and service requirements of 37 C.F.R. § 90.2.

IPR2017-01291
Patent 6,728,766 B2

PETITIONER:

Eric A. Buresh (Reg. No. 50,394) (LEAD)
eric.buresh@eriseip.com

Mark C. Lang (Reg. No. 55,356) (Back-up)
mark.lang@eriseip.com

Kathleen D. Fitterling (Reg. No. 62,950) (Back-up)
kathleen.fitterling@eriseip.com

PATENT OWNER:

Brett Mangrum
brett@etheridgelaw.com

Ryan Loveless
ryan@etheridgelaw.com

Sean Burdick
sean.burdick@unilocusa.com