

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

TCL CORPORATION, TCL COMMUNICATION TECHNOLOGY
HOLDINGS, LTD., TCT MOBILE LIMITED, TCT MOBILE INC., and
TCT MOBILE (US) INC.,
Petitioners,

v.

TELEFONAKTIEBOLAGET LM ERICSSON,
Patent Owner.

Case IPR2015-01628
Patent 7,149,510 B2

Before PATRICK R. SCANLON, BARBARA A. BENOIT, and
STACY B. MARGOLIES, *Administrative Patent Judges*.

MARGOLIES, *Administrative Patent Judge*.

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

I. INTRODUCTION

In this *inter partes* review, instituted pursuant to 35 U.S.C. § 314, TCL Corporation, TCL Communication Technology Holdings, Ltd., TCT Mobile Limited, TCT Mobile Inc., and TCT Mobile (US) Inc. (collectively, “Petitioners”) challenge the patentability of claim 11 of U.S. Patent No. 7,149,510 (Ex. 1001, “the ’510 patent”), owned by Telefonaktiebolaget LM Ericsson (“Patent Owner”). We have jurisdiction under 35 U.S.C. § 6. 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, Petitioners have not shown by a preponderance of the evidence that claim 11 of the ’510 patent is unpatentable.

A. Procedural History

Petitioners filed an Amended Petition for *inter partes* review of claim 11 of the ’510 patent. Paper 5 (“Pet.”). Patent Owner did not file a Preliminary Response to the Petition. On January 29, 2016, we instituted an *inter partes* review of claim 11 of the ’510 patent on certain asserted grounds of unpatentability. Paper 9 (“Dec.”).

Subsequent to institution, pursuant to an Order regarding means-plus-function claim construction briefing (Paper 13, “Order”), Patent Owner and Petitioners each filed a claim construction brief (Paper 17 (“PO Br.”) and Paper 18 (“Pet. Br.”), respectively). Patent Owner also filed a Patent Owner Response to the Petition (Paper 23, “PO Resp.”), to which Petitioners filed a Reply (Paper 28, “Pet. Reply”). Patent Owner further filed observations on deposition testimony of Dr. Sam Malek (Paper 34), to which Petitioners filed a Response (Paper 37).

An oral hearing was held on October 26, 2016. A transcript of the hearing has been entered into the record. Paper 42 (“Tr.”).

B. Related Matters

The parties identify the following district court proceedings in which the ’510 patent has been asserted: *Ericsson Inc. v. TCL Commc’n Tech. Holdings, Ltd.*, No. 2:15-cv-11 (E.D. Tex. 2015) and *Ericsson Inc. v. Apple Inc.*, No. 15-cv-291 (E.D. Tex. 2015). Pet. 2; Paper 7, 2–3.

The parties also identify other matters that would affect, or be affected by, a decision in this proceeding. See 37 C.F.R. § 42.8(b)(2); Pet. 1–2; Paper 7, 2–3; Paper 16, 1–2.

Claims of the ’510 patent also are subject to *inter partes* review in currently pending IPR2015-01605 and IPR2015-01622.

C. The ’510 Patent

The ’510 patent is titled “Security Access Manager in Middleware” and describes a system for controlling access to a software platform for a mobile terminal. Ex. 1001, [54], 1:25–28. The ’510 patent discloses that “drastically increased functionality” in cellular telecommunications systems “has placed substantial demands on the developers of mobile terminals” and that this demand “is exacerbated by the fact that a mobile terminal is a ‘resource scarce’ environment that is limited in size, memory, and power.” *Id.* at 1:55–61. The ’510 patent explains that mobile terminal platform assembly software and application software are developed separately and then later combined by installing, loading, and running the application software in the mobile terminal platform assembly. *Id.* at 2:18–24. The patent discloses that installing, loading, and running the application software may require non-native applications to run on the mobile terminal. *Id.* The

patent states that such non-native applications depend on functionality that is provided by the native code of the mobile terminal platform assembly, and unrestricted access to such native functionality may jeopardize the integrity of the mobile terminal by, for example, initiating cost-incurring events without notifying the user. *Id.* at 2:18–32. The '510 patent states that “[c]ertificates of origin are used on applications to determine the extent of trust therein and therefore grant access to a subset of the services made available by the mobile terminal platform assembly to the non-native execution environment.” *Id.* at 2:33–37. However, according to the '510 patent, permissions granted may be changed in run-time by the end user of the mobile terminal. *Id.* at 2:37–39. The '510 patent states that, therefore, there is a need for a dynamic registration of the permissions as well as dynamic filtering of the access to the native code of the mobile terminal platform assembly at any time. *Id.* at 2:40–44.

Figure 1 of the '510 patent, shown below, is a block diagram that schematically illustrates a platform system for a mobile terminal:

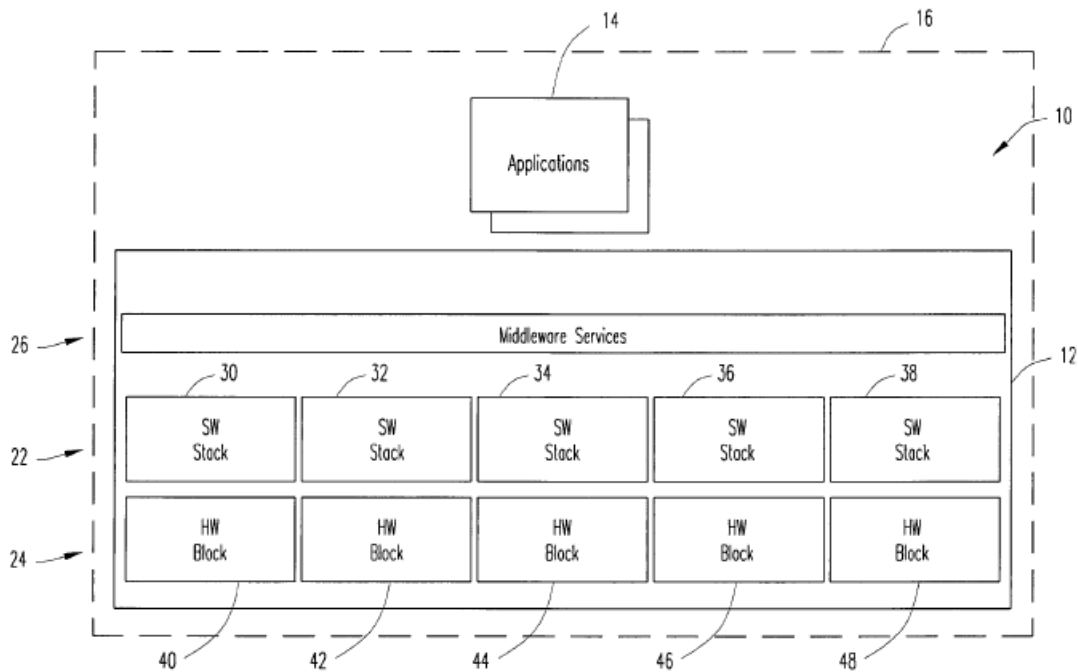


FIG. 1

Id. at 4:9–12. As shown in Figure 1 above, platform system 10 is adapted to be incorporated into mobile terminal 16 and includes mobile terminal platform assembly 12 and one or more applications (i.e., application software) 14. *Id.* at 4:12–18. Mobile terminal platform assembly 12 includes software services component 22, hardware component 24, and interface component 26. *Id.* at 4:19–21. According to the '510 patent, hardware blocks 40–48 are associated with and controlled by their respective functional software stacks 30–38. *Id.* at 4:25–32.

The '510 patent discloses that interface component 26 includes a middleware services layer having application programming interfaces (APIs) for installing, loading, and running one or more applications 14 in mobile terminal platform assembly 12. *Id.* at 4:33–40. The APIs isolate mobile terminal platform assembly 12 from applications 14.

Figure 4B of the '510 patent illustrates the relationships between application domain 500, middleware domain 501, and platform domain 502:

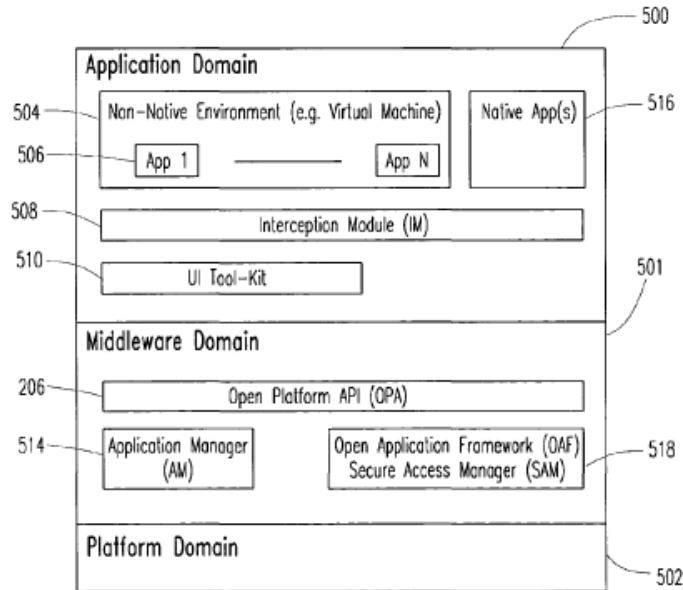


FIG. 4B

Id. at 3:43–46. As illustrated in Figure 4B above, application domain 500 includes non-native applications 506(1)–(N) and native applications 516. *Id.* at 7:6–9. According to the '510 patent, service requests from applications 506 and 516 are subject to access control via interception module 508 before they are passed on to lower-level services, such as graphical support. *Id.* at 7:11–17. Via Open Platform API (OPA) 206, applications may communicate with application manager 514 in order to request updates of access permissions. *Id.* at 7:17–20. Application manager 514 informs secure access manager (SAM) 518 of any such requests. *Id.* at 7:20–21.

Figure 5 of the '510 patent, shown below, illustrates software modules in the Open Application Framework (OAF) API domain:

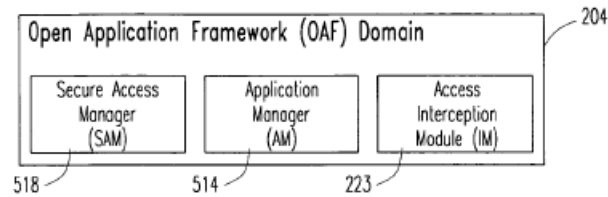


FIG. 5

Id. at 7:26–29. As illustrated in Figure 5 above, OAF domain 204 includes secure access manager (SAM) 518, application manager (AM) 514, and access interception module (IM) 223. *Id.* at 7:29–31, Fig. 5. The ’510 patent discloses that SAM 518 decides whether or not a call from non-native applications should be permitted. *Id.* at 7:31–37. According to the patent, “SAM 518 holds and maintains the security policies related to the access of the platform services.” *Id.* at 7:37–39. The patent also discloses that IM 223 monitors service requests from applications running in non-native environment 504 (shown in Figure 4B). *Id.* at 7:41–44.

The ’510 patent explains that SAM 518 may grant access to the native platform services in a variety of ways, such as the method illustrated in Figure 7 below:

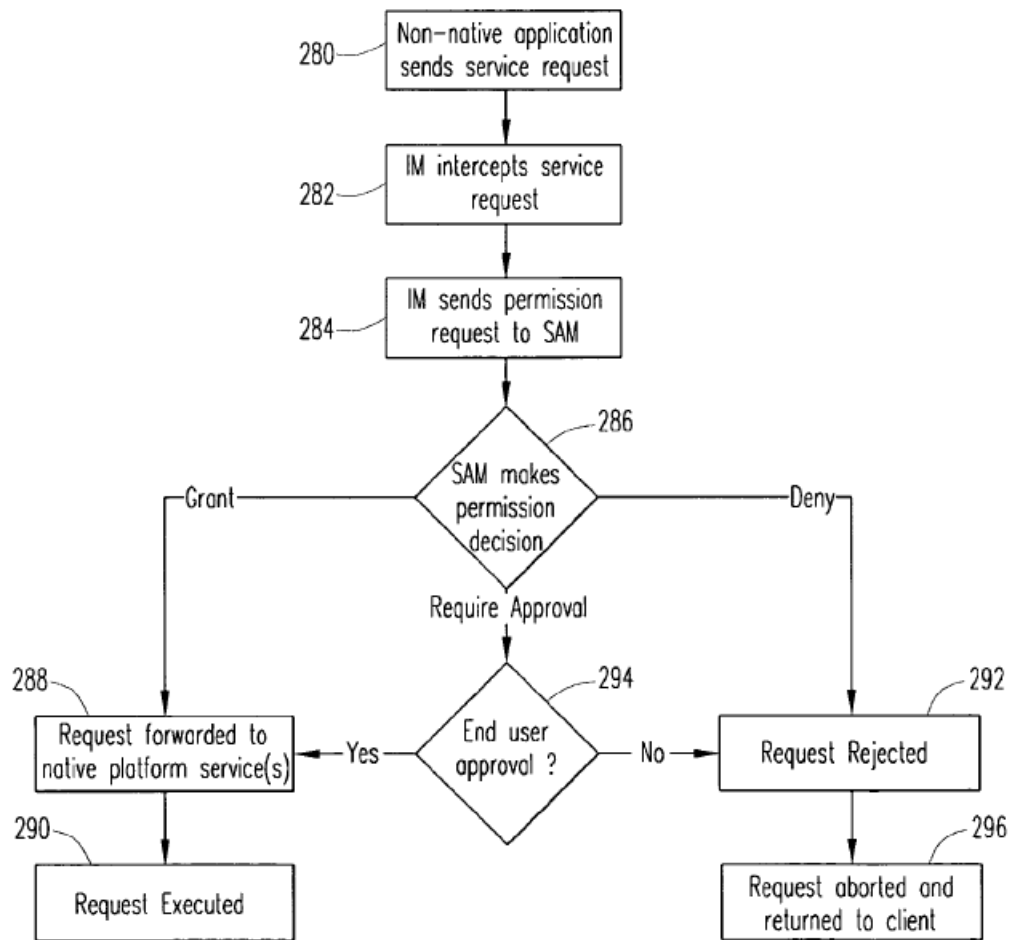


FIG. 7

Id. at 7:61–63. Figure 7 above is a flow chart illustrating a method for requesting access and receiving a permission decision from SAM 518. *Id.* at 3:58–60. At step 280, non-native application 250 (shown in Fig. 6A) requests a service that requires access to the native platform services. *Id.* at 8:1–3. At step 282, IM 223 intercepts the service request, which according to the patent “includes an ID tag of the requesting non-native application 250.” *Id.* at 8:3–5. At step 284, IM 223 sends a permission request to SAM 518 along with the ID tag included in the service request. *Id.* at 8:5–7. According to the patent, IM 223 may also send additional access information

and an identification of the native platform service that the non-native application desires to access. *Id.* at 8:7–10. SAM 518 reviews the security policies of the native platform services to determine if access may be granted. *Id.* at 8:11–13. At step 286, SAM 518 forms a permission decision and forwards the decision to IM 223. *Id.* at 8:13–15. If the permission request is granted, then at step 288 the service request is forwarded to the requested native platform service and executed at step 290. *Id.* at 8:15–18. If the permission request is denied, then at step 296 a reject response is sent to the non-native application. *Id.* at 8:18–20. The '510 patent discloses that if the ID tag does not match one of the ID tags included in the located access record, the request is rejected at step 292 and the request is aborted and returned to the requesting non-native application at step 296. *Id.* at 8:21–25.

The '510 patent also discloses an embodiment in which instead of IM 223 sending a permission request with the ID tag to SAM 518, IM 223 makes a decision locally. *Id.* at 8:35–38, 8:50–52. In this embodiment, IM 223 maintains access records for the native platform services. *Id.* at 8:52–53. According to the patent, “[e]ach access record includes the ID tags of specific applications that have permission to access the requested native platform service.” *Id.* at 8:53–55.

D. Challenged Claim

Claim 11 is the only challenged claim and reads as follows:

11. A system for controlling access to a platform, the system comprising:

a platform having a software services component and an interface component, the interface component having at least one interface for providing access to the software services component for enabling application domain software to be installed, loaded, and run in the platform; and

an access controller for controlling access to the software services component by a requesting application domain software via the at least one interface, the access controller comprising:

an interception module for receiving a request from the requesting application domain software to access the software services component;

a decision entity for determining if the request should be granted;

wherein such interception module includes a cache with the rules and policies of the decision entity; and

wherein the requesting application domain software is granted access to the software services component via the at least one interface if the request is granted.

Id. at 12:21–43.

E. Asserted Grounds of Unpatentability

We instituted an *inter partes* review of claim 11 of the '510 patent on the following grounds:

References	Basis	Challenged Claim
Usui, ¹ Gong, ² and Ramamurthy ³	35 U.S.C. § 103 ⁴	11
Usui, Gong, Spencer, ⁵ and Ramamurthy	35 U.S.C. § 103	11

II. DISCUSSION

A. Principles of Law

To prevail in challenging Patent Owner’s patent claims, Petitioners must demonstrate by a preponderance of the evidence that the claims are unpatentable. 35 U.S.C. § 316(e); 37 C.F.R. § 42.1(d). “In an [*inter partes* review], the petitioner has the burden from the onset to show with particularity why the patent it challenges is unpatentable.” *Harmonic Inc. v. Avid Tech., Inc.*, 815 F.3d 1356, 1363 (Fed. Cir. 2016) (citing 35 U.S.C. § 312(a)(3) (requiring *inter partes* review petitions to identify “with particularity . . . the evidence that supports the grounds for the challenge to each claim”)). This burden never shifts to Patent Owner. *See Dynamic*

¹ Kazutoshi Usui et al., *Design and Implementation of Java Application Environment and Software Platform for Mobile Phones*, NEC Res. & Dev., Vol. 42, No. 4 (Oct. 2001) (Ex. 1010, “Usui”).

² U.S. Patent No. 6,125,447, issued Sept. 26, 2000 (Ex. 1011, “Gong”).

³ U.S. Patent No. 7,080,777, issued July 19, 2006 (Ex. 1401, “Ramamurthy”).

⁴ The Leahy-Smith America Invents Act (“AIA”), Pub. L. No. 112-29, 125 Stat. 284, 287–88 (2011), revised 35 U.S.C. § 103, effective March 16, 2013. Because the challenged patent was filed before March 16, 2013, we refer to the pre-AIA version of § 103 in this decision.

⁵ Ray Spencer et al., *The Flask Security Architecture: System Support for Diverse Security Policies*, Proc. 8th USENIX Security Symp., Aug. 23–29, 1999 (Ex. 1019, “Spencer”).

Drinkware, LLC v. Nat'l Graphics, Inc., 800 F.3d 1375, 1378 (Fed. Cir. 2015) (citing *Tech. Licensing Corp. v. Videotek, Inc.*, 545 F.3d 1316, 1326–27 (Fed. Cir. 2008)) (discussing the burden of proof in *inter partes* review). Furthermore, a petitioner cannot satisfy its burden of proving obviousness by employing “mere conclusory statements.” *In re Magnum Oil Tools Int’l, Ltd.*, 829 F.3d 1364, 1380 (Fed. Cir. 2016).

Petitioners assert that claim 11 of the ’510 patent is unpatentable under 35 U.S.C. § 103(a) as obvious. A claim is unpatentable under 35 U.S.C. § 103(a) if the differences between the claimed subject matter and the prior art are such that the subject matter, as a whole, would have been obvious at the time of the invention to a person having ordinary skill in the art. *KSR Int’l Co. v. Teleflex, Inc.*, 550 U.S. 398, 406 (2007). The question of obviousness is resolved on the basis of underlying factual determinations including: (1) the scope and content of the prior art; (2) any differences between the claimed subject matter and the prior art; (3) the level of ordinary skill in the art; and (4) objective evidence of nonobviousness. *Graham v. John Deere Co.*, 383 U.S. 1, 17–18 (1966). “A determination of whether a patent claim is invalid as obvious under § 103 requires consideration of all four *Graham* factors, and it is error to reach a conclusion of obviousness until all those factors are considered.” *Apple Inc. v. Samsung Elecs. Co.*, 839 F.3d 1034, 1048 (Fed. Cir. 2016) (en banc) (citations omitted). “This requirement is in recognition of the fact that each of the *Graham* factors helps inform the ultimate obviousness determination.” *Id.*

B. Level of Ordinary Skill

In determining whether an invention would have been obvious at the time it was made, 35 U.S.C. § 103 requires us to resolve the level of

ordinary skill in the pertinent art at the time of the invention. *Graham*, 383 U.S. at 17. “The importance of resolving the level of ordinary skill in the art lies in the necessity of maintaining objectivity in the obviousness inquiry.” *Ryko Mfg. Co. v. Nu-Star, Inc.*, 950 F.2d 714, 718 (Fed. Cir. 1991). The person of ordinary skill in the art is a hypothetical person who is presumed to have known the relevant art at the time of the invention. *In re GPAC, Inc.*, 57 F.3d 1573, 1579 (Fed. Cir. 1995). Factors that may be considered in determining the level of ordinary skill in the art include, but are not limited to, the types of problems encountered in the art, the sophistication of the technology, and educational level of active workers in the field. *GPAC*, 1573 F.3d at 1579. In a given case, one or more factors may predominate. *Id.* Generally, it is easier to establish obviousness under a higher level of ordinary skill in the art. *Innovation Toys, LLC v. MGA Entm’t, Inc.*, 637 F.3d 1314, 1323 (Fed. Cir. 2011) (“A less sophisticated level of skill generally favors a determination of nonobviousness . . . while a higher level of skill favors the reverse.”).

Petitioners’ declarant Dr. Malek opines that one of ordinary skill in the art would have had “a bachelor’s degree in computer engineering, computer science, or a related field, and one to two years of experience with mobile software code security architecture or equivalent education and experience.” Ex. 1406 ¶ 49. Dr. Malek explains that he determined the characteristics of a hypothetical person of ordinary skill in the art of the ’510 patent as of September 23, 2002 based on his review of the ’510 patent, the prior art references cited in the ’510 patent and referenced in his declaration, and on his “knowledge of access control mechanisms for mobile software.” *Id.*

Patent Owner does not challenge Dr. Malek's testimony regarding the level of skill in the art. *See generally* PO Resp. However, Dr. Muftic, Patent Owner's declarant, opines that a person of ordinary skill in the art "would probably have had a Bachelor's degree in computer science, with at least two years' experience in the field of software security." Ex. 2010 ¶ 22. Dr. Muftic's proposed level of ordinary skill differs slightly from Dr. Malek's. Patent Owner, however, does not cite the above testimony in its Patent Owner Response. Nor does Patent Owner or Dr. Muftic quarrel with the definition proposed by Petitioner and Dr. Malek. *See generally* PO Resp.

We determine that the level of ordinary skill proposed by Petitioners' declarant is consistent with the challenged patent and the asserted prior art, and we therefore adopt Dr. Malek's definition of the level of ordinary skill in the art for the purposes of the analysis below. However, our analysis would not differ if we were to adopt Dr. Muftic's definition.

C. Claim Construction

In an *inter partes* review, we construe claim terms in an unexpired patent 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); *Cuozzo Speed Techs., LLC v. Lee*, 136 S. Ct. 2131, 2144–46 (2016) (upholding the use of the broadest reasonable interpretation standard). Consistent with the broadest reasonable construction, claim terms are presumed to have their ordinary and customary meaning as understood by a person of ordinary skill in the art in the context of the entire patent disclosure. *In re Translogic Tech., Inc.*, 504 F.3d 1249, 1257 (Fed. Cir. 2007). An inventor may provide a meaning for a term that is different from

its ordinary meaning by defining the term in the specification with reasonable clarity, deliberateness, and precision. *In re Paulsen*, 30 F.3d 1475, 1480 (Fed. Cir. 1994).

Petitioners propose constructions for a number of terms, and asserts that “interface component,” “interception module,” “decision entity,” “security access manager,” and “access controller” are means-plus-function limitations. Pet. 16–20; Pet. Br. 5–15. Patent Owner disagrees that the foregoing terms are means-plus-function limitations and asserts that only “software services component” be explicitly construed. PO Br. 6–16; PO Resp. 8–9.

As explained below, the dispositive issue in this case is whether Petitioners provide a persuasive articulated reason with rational underpinning to support their contention that a person of ordinary skill would have been motivated to combine *Usui* and *Gong*. In reaching this issue, we determine that no claim terms require express construction.

D. Asserted Obviousness over Usui, Gong, and Ramamurthy

Petitioners contend that claim 11 is unpatentable under 35 U.S.C. § 103 as obvious over *Usui*, *Gong*, and *Ramamurthy*. Pet. 3, 20–53. Petitioners cite to the prior art references and rely on declaration testimony of Dr. Malek. *Id.* at 20–53 (citing Ex. 1406). Furthermore, Petitioners provide analysis purportedly explaining how the combination would have conveyed to one of ordinary skill in the art the claim limitations and provide reasons why one of ordinary skill in the art would have combined the references. *Id.*

With support of the declaration of Dr. Muftic (Ex. 2010), Patent Owner challenges Petitioners’ contentions, arguing among other things that

a person of ordinary skill in the art would not have been motivated to combine the references in the manner proposed by Petitioner. PO Resp. 2–3, 13–31. In reply, Petitioners maintain their position that the subject matter of claim 11 would have been obvious. *See generally* Pet. Reply.

We have reviewed the Petition, Patent Owner’s Response, and Petitioners’ Reply, as well as the relevant evidence discussed in those papers. For the reasons that follow, we determine that Petitioners have not shown by a preponderance of the evidence that the subject matter of claim 11 would have been obvious in view of Usui, Gong, and Ramamurthy.

1. Summary of Usui

Usui is an October 2001 article titled “Design and Implementation of Java Application Environment and Software Platform for Mobile Phones.” Ex. 1010, 2. Usui describes an “efficient JAE (Java Application Environment) for mobile phones and the layer-structured flexible software platform on which the JAE runs.” *Id.*

Figure 1 of Usui, shown below, illustrates the layered structure of the new software platform:

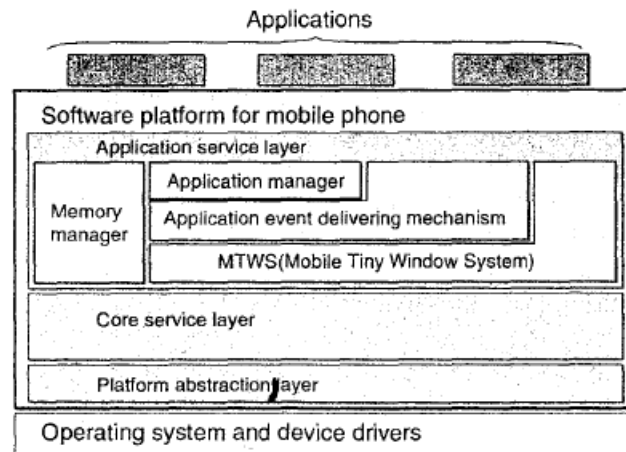


Fig. 1 Software platform for mobile phones.

Id. at 3. As shown in Figure 1 above, the software platform for mobile phones includes a platform abstraction layer, a core service layer, and an application service layer. *Id.* The application service layer includes an application manager, which, according to Usui, “receives requests from applications and sends commands to them using the application event delivering mechanism.” *Id.* Usui states that “[a]pplication programmers can develop applications easily because the application manager prevents applications from interfering with the shared resources of each other.” *Id.*

Usui discloses that security management, “which controls how a Java program is restricted to access to hardware resources in the phone,” is “important.” *Id.* at 4. Usui discloses that one security management model is a sandbox model, which gives a Java program no access rights to the hardware resources in the mobile phone. *Id.* Usui explains that this model is for an untrusted program. *Id.* Usui also discloses that the Mobile Execution Environment (MExE) standard provides for security domains for trusted programs. *Id.* According to Usui, MExE “defines three security domains in which the program can access hardware resources within a level

to be permitted in the domain in order to provide hardware-supported services including telephony and device maintenance.” *Id.*

Figure 3 of Usui below illustrates its JAE for mobile phones:

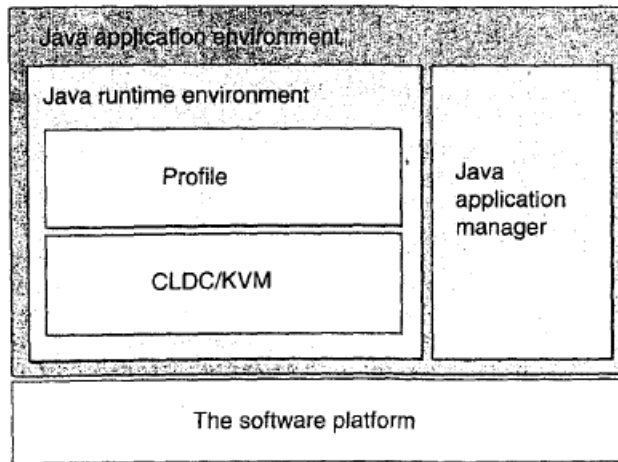


Fig. 3 Java application environment.

Id. at 4. As illustrated in Figure 3 above, Usui’s JAE includes a Java application manager (JAM) and a Java runtime environment (JRE). *Id.* Usui discloses that the JAM handles downloading Java programs and security management. *Id.* The JAM includes a security manager which, according to Usui, “controls access to hardware resources including network communications.” *Id.* Usui discloses that the security manager “encapsulates the security management mechanism, making it possible to implement the security domain defined in MExE.” *Id.*

Usui also discloses that the JRE consists of two main parts—a Connected Limited Device Configuration (CLDC) and a profile. *Id.* at 5. Usui discloses that the CLDC “defines the basic J2ME (Java 2 Micro Edition) runtime environment, which comprises Java technologies for embedded systems.” *Id.* Usui states that CLDC “has functions that are a subset of Java language features” and “a subset of the functions of the Java

virtual machine.” *Id.* Usui also discloses that the authors designed the user interface (UI) “so that the Java part was as small as possible” to “achieve good performance of the UI capability in the mobile phone.” *Id.* Usui further describes a resource release mechanism and explains that “[r]eleasing resources that are no longer used by Java applications is very important for resource-limited systems such as mobile phones.” *Id.*

2. Summary of Gong

Gong is a U.S. patent titled “Protection Domains to Provide Security in a Computer System.” Ex. 1011, [54]. Gong describes creating protection domains for code written in an object-oriented programming language and an access controller that uses those protection domains to determine whether access to computer resources should be granted. *See id.* at Abstract, 2:52–3:8, 6:24–44, 9:54–56, 11:34–39. Gong explains as background that one security method is the sandbox method, which “permits all trusted code full access to [a] computer system’s resources and all remote code limited access to a computer system’s resources.” *Id.* at 1:66–2:6. Gong states that a drawback of the sandbox approach is that it “is not very granular because all remote code is restricted to the same limited set of resources.” *Id.* at 2:10–13. Gong also explains that providing security measures that allow more granularity than the sandbox method and that address receiving requests from multiple sources on a vast network such as the Internet requires “developing complex security software.” *Id.* at 2:23–35; *see also id.* at 6:24–31.

According to Gong, “[t]he use of protection domains provides a relatively simple mechanism to implement otherwise complex security policies.” *Id.* at 6:40–42. Gong discloses a system that “facilitates

development of more granular security mechanisms.” *Id.* at 11:1–4. For example, according to Gong, “objects based on code from a first untrusted source can be granted one set of permissions and objects based on code from a second untrusted source can be granted a second set of permissions.” *Id.* at 11:4–7. Specifically, Gong discloses using protection domains to represent security policies that apply to a computer system. *Id.* at 2:52–56. Gong describes establishing an association between protection domains and classes of objects that may be invoked by the computer system. *Id.* at 2:60–62. Gong discloses that the association between a protection domain and a class is based on a code identifier. *Id.* at 3:11–12. According to Gong, “the code identifier may contain data describing the source of code that defines a class, a set of public cryptographic keys associated with the source of code, or other information which describes the source of code.” *Id.* at 3:11–16. Gong explains that “because the permissions authorized to an object are based on the code identifier associated with the object’s class, an object’s permission can [be] based on the source of code creating the object.” *Id.* at 10:64–11:1.

Gong discloses that a code executor, such as a Java virtual machine, maintains for each thread or process a call stack of the object methods invoked by the thread or process. *Id.* at 11:40–42. Gong discloses that objects corresponding to the method invocations in the call stack are each associated with a protection domain. *Id.* at 11:55–57. Gong describes a specific example based on the calling stack and protection domains illustrated in Figure 6 below:

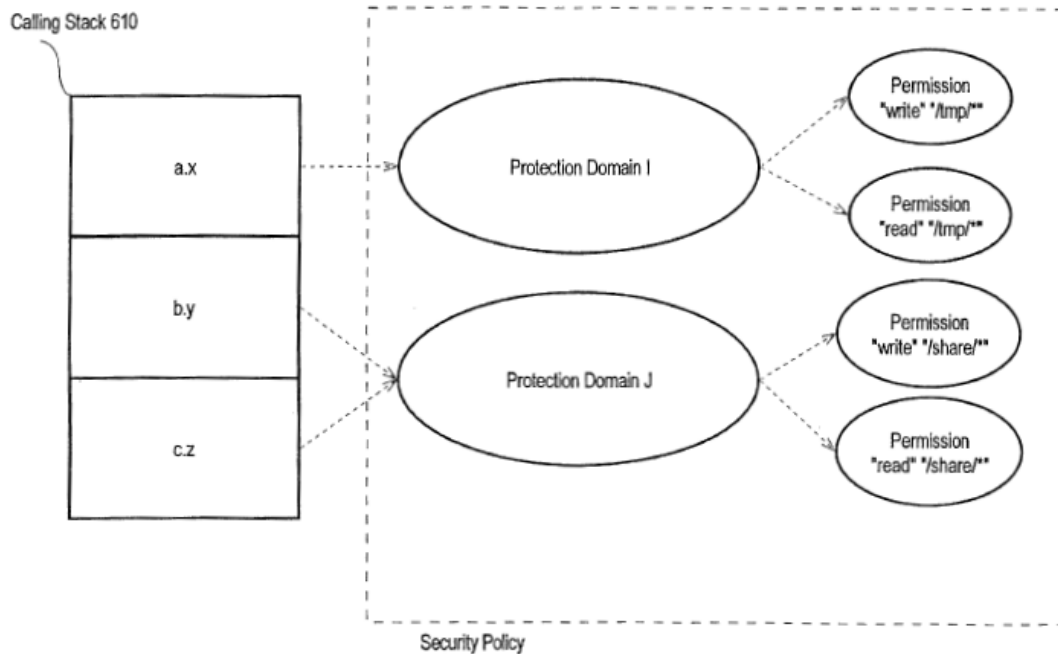


Fig. 6

Figure 6 of Gong, above, shows calling stack 610 that represents the calling hierarchy of the methods invoked by the thread but that have not yet been completed by the thread. *Id.* at 11:50–53. In the example illustrated in Figure 6, a thread executes a.x and a.x invokes b.y, which in turn invokes c.z, where the first letter (e.g., “a”) is an object, and the second letter (e.g., “x”) is a method associated with that object. *Id.* at 11:46–50, 11:65–67. As illustrated in Figure 6, object a is associated with protection domain I and objects b and c are associated with protection domain J. *Id.* at 11:55–59. Also as illustrated above, each protection domain object is associated with permission objects. *Id.* at 11:59–60.

Figure 5 of Gong is illustrated below:

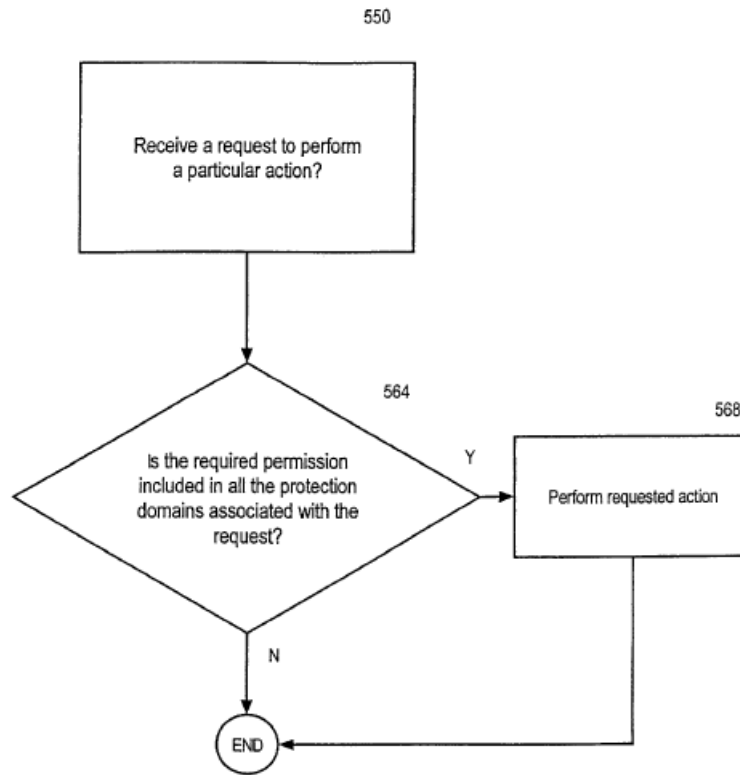


Fig. 5

Figure 5 above illustrates the steps followed by the access controller for determining whether a desired action from executing code is permitted. *Id.* at 3:36–38, 4:1–4, 11:34–36, 11:34–39. Considering Figures 5 and 6 together, Gong discloses that, per step 550 of Figure 5, a request to perform a particular action is received from the thread. *Id.* at 11:67–12:2. In Gong’s example, object a is requesting to write to file “/tmp/temporary.” *Id.* at 12:7–8. A resource manager receives the request from object a, and invokes an access controller. *Id.* at 12:16–19. According to Gong, “[t]he access controller determines whether the permission required is authorized for the entity requesting access.” *Id.* at 12:19–21. Specifically, per step 564 of Figure 5, the access controller determines whether the required permission for the requested action is included in all of the protection domains

associated with the request. *Id.* at Fig. 5, 12:24–27. Gong discloses that “[e]ach permission object of each particular protection domain object associated with each object represented by the call stack is examined in order to determine whether the permission object authorizes the required permission.” *Id.* at 12:34–38.

Returning to the example, Gong discloses that the access controller first examines the permission object associated with object a, and determines that the first permission object represents an authorized permission to write to any file in directory “tmp.” *Id.* at 13:3–9. Access controller then examines the protection domain for the class of the next object represented in the call stack, which is b. *Id.* at 13:10–12. Protection domain object J is the protection domain associated with the class of object b. *Id.* at 13:12–13. Gong discloses that when the validation method of the second permission object in protection domain object J is invoked, the validation method indicates the required permission is not authorized. *Id.* at 13:19–21. Thus, according to Gong, execution of the steps ceases, per step 560 of Figure 5. *Id.* at 21–22.

3. Analysis of asserted motivation to combine

Petitioners maintain, among other things, that an ordinarily skilled artisan would have been motivated to combine Usui and Gong. *See* Pet. 29–32. For the reasons explained below, we do not find Petitioners’ arguments persuasive.

a. Petitioners’ contentions

Petitioners maintain that a person of ordinary skill in the art would have been motivated “to implement the access controller of Gong as the security management mechanism of Usui to provide the fine-grained access

control identified in Usui as this is precisely what the access control permission enforcement mechanism of Gong was designed to do.” Pet. 30 (citing Ex. 1406 ¶ 179). Petitioners argue that “both references are in the same general field of endeavor and, as recognized by the Supreme Court, known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on design incentives or other market forces if the variations are predictable to one of ordinary skill in the art.” *Id.* at 30–31 (emphasis omitted) (citing *KSR*, 550 U.S. at 417).

Petitioners also argue that the Third Generation Partnership Project (3GPP)—an association of telecommunication companies and mobile terminal manufacturers that developed standards for compatibility of telecommunications systems operated around the world—sought to create MExE to ensure that applications taking advantage of 3G services could be developed once and run on any compliant mobile terminal. *Id.* at 31 (citing Ex. 1024, 14); *see also* Ex. 1406 ¶ 104. Petitioners state that the MExE standard required mobile phones to provide a standard execution environment with multiple security domains, providing varying levels of access to applications. Pet. 31 (citing Ex. 1025, 8). Petitioners assert that “[t]his market force” would have motivated an ordinarily skilled artisan “to implement known applicable methods of providing varying levels of access control to Java programs—such as the protection domain-based access controller of Gong—within the mobile phone environment to be compliant with the MExE requirements.” *Id.* (citing Ex. 1406 ¶ 181).

Petitioners further assert that Usui “provides an implied motivation” to implement the access controller of Gong. *Id.* (citing Ex. 1406 ¶ 182). According to Petitioners, Usui describes its security manager as

encapsulating the security management mechanism, thereby enabling the required MExE security domain model. *Id.* at 31–32 (citing Ex. 1010, 4; Ex. 1406 ¶ 182). Petitioners argue that Usui thus “contemplates the use of protection domains, similar to those taught in Gong, to control access of Java applications to mobile phone resources.” *Id.* at 32 (citing Ex. 1406 ¶ 182). Petitioners argue that “[i]t would have been within the common sense of a [person of ordinary skill in the art] to use the known applicable method for controlling access by Java applications taught in the primary embodiment of Gong to provide the type of access control model encapsulated by the security manager of Usui.” *Id.* (citing Ex. 1406 ¶ 182).

b. Patent Owner’s response

Patent Owner asserts that “Petitioners’ argument that a person of ordinary skill in the art would have combined the **mobile** Java platform discussed in the Usui reference . . . with the resource-intensive **desktop** Java implementation discussed in the Gong reference . . . is fundamentally flawed and ignores the technological realities of the relevant timeframe.” PO Resp. 2.

First, Patent Owner argues that an ordinarily skilled artisan would not have combined the resource-heavy access controller of Gong with the mobile Java platform discussed in Usui, which had limited CPU capacity and memory. *Id.* at 12–26. Patent Owner asserts that the Java mobile platform at the time of the claimed invention—Java 2 Micro Edition (J2ME)—used a security model different from that described in Gong. *Id.* at 14–21. Specifically, according to Patent Owner, Sun Microsystems—the creator of the Java platform—“chose to implement the simpler sandbox-based model for mobile phones rather than using the [Java 2 Standard

Edition (J2SE)] security model of Gong.” *Id.* at 16. Patent Owner asserts that “[t]he fact that Sun Microsystems did not incorporate the fine-grained access control mechanisms described by Gong into J2ME speaks volumes about motivation to combine.” *Id.* at 22. Patent Owner also asserts that even if there was a reason to combine the disparate teachings, a skilled artisan would not have had a reasonable expectation of success in doing so. *Id.* at 14.

In response to Petitioners’ reliance on MExE as a market force, Patent Owner argues that Petitioners have not explained how the MExE standard would have motivated a skilled artisan to venture outside the standard and use the security of Gong. *Id.* at 28. Patent Owner asserts that in fact there was no market force to implement Gong with Usui because Usui “was already compliant with the ‘market’ by implementing MExE.” *Id.* at 29. Patent Owner also argues that there was no implied motivation in Usui to “swap out the J2ME sandbox model for the much larger and processor taxing Gong mechanism that was designed for desktop and enterprise computers.” *Id.*

Patent Owner further argues, in response to Petitioners’ reliance on common sense (*see* Pet. 32), that a person of ordinary skill in the art developing a mobile platform “would use common sense and apply the security mechanisms already developed by Sun Microsystems for the J2ME platform”—specifically, J2ME sandboxing. PO Resp. 29–30.

c. Petitioners’ reply

In their reply, Petitioners argue that Usui is not limited to J2ME and Gong is not limited to the standard edition of Java. Pet. Reply 17. Petitioners argue that all of the claim elements are present within the

combination of Usui and Gong and are being used for the same purpose. *Id.* at 18. Petitioners also argue that Patent Owner’s arguments rest on an overly rigid application of the teaching/suggestion/motivation test in a manner inconsistent with *KSR*. *Id.* at 19–22. Petitioners further argue that Usui does not eschew the teachings of Gong, and there is at least one example (as purportedly described in Exhibits 1034 and 1035) of a system that ran the entire J2SE implementation on a mobile phone as early as 2001. *Id.* at 20–22.

d. Analysis

Petitioners rely on a particular theory for the motivation to combine: that a person of ordinary skill in the art, prior to the time of the alleged invention of the ’510 patent, “would have been motivated . . . to implement the access controller of Gong as the security management mechanism of Usui to provide the fine-grained access control identified in Usui as this is precisely what the access control permission enforcement mechanism of Gong was designed to do.” Pet. 30 (citing Ex. 1406 ¶ 179). Petitioners provide several reasons in support of their theory. *Id.* at 30–32. Having reviewed the parties’ arguments and supporting evidence, we determine that Petitioners fail to persuasively show that a skilled artisan would have been motivated to implement the access controller of Gong as the security management mechanism in Usui to provide the purported fine-grained access control identified in Usui. We therefore determine that Petitioners fail to persuasively establish an articulated reasoning with rational underpinning to support the legal conclusion of obviousness. *See KSR*, 550 U.S. at 418 (quoting *In re Kahn*, 441 F.3d 977, 988 (Fed. Cir. 2006)).

As explained above, Usui describes an implementation of the Java application environment and software platform for mobile phones. Ex. 1010, 2. Usui discloses using the CLDC, which “defines the basic J2ME (Java 2 Micro Edition) runtime environment” for embedded systems. *Id.* at 5. Usui states that the CLDC “has functions that are a subset of Java language features” and “a subset of the functions of the Java virtual machine.” *Id.* Usui discloses using the K Virtual Machine (KVM), which is the virtual machine for mobile devices. *Id.*; *see also* Ex. 2010 ¶¶ 207, 208; Ex. 2013, 16. Usui discloses that the authors considered the constrained resources of mobile devices in designing their software. Specifically, Usui discloses that the authors designed their user interface (UI) “so that the Java part was as small as possible” to “achieve good performance of the UI capability in the mobile phone.” Ex. 1010, 5. Usui further states that “[r]eleasing resources that are no longer used by Java applications is very important for resource-limited systems such as mobile phones.” *Id.*

Gong, on the other hand, describes Java security mechanisms for computers. Ex. 2010 ¶¶ 92–98, 101–105, 106 (“Gong describes solutions applicable in a very specific environment, to . . . very specific type of applications, using very specific methods of protection domains and with very specific prerequisites, which are required to be present in the Java platform.”); *see also* Ex. 1011, Abstract, Fig. 1, 4:25–57, 7:15–24; Pet. 24 (“Gong discloses a method and system for enforcing access control permissions for application code downloaded to and executing on a computer.”); Ex. 1406, ¶¶ 168 (same), 175. Gong describes a fine-grained security mechanism using an access controller. *See, e.g.*, Ex. 1011, 10:63–11:7, 11:32–13:22; Ex. 2010 ¶ 223. According to the Petition, Gong

discloses an access controller that identifies protection domains associated with a requesting application, and after identifying the relevant protection domain, searches the permissions associated with the requesting application to determine whether access should be granted. Pet. 46–47. As Petitioners explain, Gong discloses that the access controller invokes several methods to identify information necessary for determining whether access should be permitted, including the method of the domain mapper, to return the protection domain associated with a requesting application. *Id.* at 46.

Petitioners thus rely on using Gong’s Java access controller for computers as the security mechanism in Usui’s mobile phone system, and that a skilled artisan would have been motivated to make such a combination to provide the fine-grained access control identified in Usui. *See* Pet. 30, 42; Ex. 1406 ¶ 179. We are not persuaded by Petitioners’ arguments purportedly explaining why a skilled artisan would have been so motivated.

First, Petitioners argue that “both [Usui and Gong] are in the same general field of endeavor and, as recognized by the Supreme Court, known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on design incentives or other market forces if the variations are predictable to one of ordinary skill in the art.” Pet. 30–31 (emphasis omitted). Petitioners’ declarant, Dr. Malek, repeats this statement from the Petition but fails to provide any additional explanation for why, simply because the two references are in the same field of endeavor, a skilled artisan would have had reason to make the proposed combination. *See* Ex. 1406 ¶ 180. Nor do Petitioners or Dr. Malek elucidate what design incentives would have prompted a skilled artisan to use the fine-grained security control of Gong with the mobile phone platform of Usui,

given the design constraints of less memory and resources of a mobile phone. *See, e.g.*, Ex. 1010, 5; Ex. 2014, 14, 25; Ex. 2015, 8.

Indeed, Patent Owner cites uncontroverted evidence that in designing the J2ME platform for mobile phones, Sun Microsystems did not use the full security of the Java 2 Standard Edition and its fine-grained protection domains like that described in Gong. *See* Ex. 2010 ¶¶ 210, 211, 223, 224. For example, the specification for the CLDC of J2ME, which defines a “standard, minimum-footprint Java platform for small, resource-constrained, connected devices,” states that “the J2SE security model is too memory-consuming to be included in a CLDC device that allows only a few hundred kilobytes of total memory budget.” Ex. 2014, 14, 25; *see also* PO Resp. 21–26. In addition, a 2000 JavaOne presentation from a Sun Microsystems employee explains that the “J2SE platform security model is much larger than the entire CLDC implementation” and that the “full J2SE™ platform security model” cannot be supported in CLDC target devices. Ex. 2015, 48. In view of this context, Petitioners lack of explanation regarding purported design incentives that would have prompted a skilled artisan to make the proposed combination is significant. *See, e.g.*, Ex. 1010, 5 (disclosing that the Usui team designed the user interface (UI) “so that the Java part was as small as possible” to “achieve good performance of the UI capability in the mobile phone” and noting that “[r]eleasing resources that are no longer used by Java applications is very important for resource-limited systems such as mobile phones”); Ex. 1001, 1:55–61 (disclosing that “drastically increased functionality” in cellular telecommunications systems “has placed substantial demands on the developers of mobile terminals” and this demand

“is exacerbated by the fact that a mobile terminal is a ‘resource scarce’ environment that is limited in size, memory, and power”).

As to the purported presence of “market forces,” Petitioners rely on the MExE standard, which according to Petitioners required mobile phones to provide a standard execution environment with multiple security domains, providing varying levels of access to applications. Pet. 31 (citing Ex. 1025, 8). Petitioners assert that “[t]his market force” would have motivated an ordinarily skilled artisan “to implement known applicable methods of providing varying levels of access control to Java programs—such as the protection domain-based access controller of Gong—within the mobile phone environment to be compliant with the MExE requirements.” *Id.* Again, Dr. Malek repeats the assertions set forth in the Petition without any additional explanation. See Ex. 1406 ¶ 181.

Usui mentions MExE in the context of security policies for trusted programs, not untrusted programs. Ex. 1010, 4; *see also* Ex. 2010 ¶ 225. For untrusted programs, Usui describes a sandbox model, which gives a Java program no access rights to the hardware resources of the mobile phone. Ex. 1010, 4. This sandbox model is the model specifically designed for J2ME. Ex. 2010 ¶ 226. And it is not fine-grained. See Ex. 1010, 4; Pet. Reply 22. Usui then explains that MExE “discusses various security policies such as the security domain for trusted programs.” Ex. 1010, 4. Usui states that “MExE defines three security domains in which the program can access hardware resources within a level to be permitted in the domain in order to provide hardware-supported services including telephony and device maintenance.” Ex. 1010, 4. The three security domains to which Usui is referring are the MExE Security Operator Domain, the MExE Security

Manufacturer Domain, and the MExE Security Third Party Domain. Ex. 2010 ¶¶ 218–221, 225, 226, 288, 295–297. Petitioners do not explain how those three security domains—which do not require the Gong security management mechanism (Ex. 2010 ¶ 226)—would have prompted a skilled artisan to use the Gong security features. *See* Ex. 2010 ¶¶ 218–221. Nor do Petitioners explain how those security domains provide fine-grained access control. *See* Pet. 29–32.

Patent Owner submits persuasive evidence that using J2ME—as the Usui team did—would have been sufficient to be compliant with MExE. *See* PO Resp. 26–29. And, as explained above, J2ME did not use the full security of the Java 2 Standard Edition used for desktop computers. Petitioners do not persuasively show how MExE was a market force that would have prompted a skilled artisan to use the access controller of Gong in a resource-constrained mobile phone.

Second, Petitioners argue that Usui “provides an implied motivation” to implement the access controller of Gong. Pet. 31 (citing Ex. 1406 ¶ 182). According to Petitioners, Usui describes its security manager as encapsulating the security management mechanism, thereby enabling the required MExE security domain model. *Id.* at 31–32 (citing Ex. 1010, 4; Ex. 1406 ¶ 182). Petitioners argue that Usui thus “contemplates the use of protection domains, similar to those taught in Gong, to control access of Java applications to mobile phone resources.” *Id.* at 32 (citing Ex. 1406 ¶ 182). We disagree. Petitioners fail to persuasively explain how the protection domains contemplated by Usui are similar to those taught in Gong. As explained above, Usui discloses that for untrusted programs, Usui discloses using the sandbox model that was specifically designed for the

J2ME platform. *See* Ex. 2010 ¶ 226. For trusted programs, Usui states that MExE defines security domains “in which the program can access hardware resources within a level to be permitted in the domain.” Ex. 1010, 4.

Petitioners do not explain how the three security domains provide fine-grained access control or explain how they control access similar to the way security is handled by the access controller described in Gong. *See* Pet. 31–32. In short, Petitioners fail to show an implied motivation in Usui to use the larger and processor-taxing Gong mechanism that was designed for computers.

Third, Petitioners argue that “[i]t would have been within the common sense of a [person of ordinary skill in the art] to use the known applicable method for controlling access by Java applications taught in the primary embodiment of Gong to provide the type of access control model encapsulated by the security manager of Usui.” *Id.* at 32 (citing Ex. 1406 ¶ 182). Dr. Malek repeats the statement from the Petition. Ex. 1406 ¶ 182. Neither Petitioners nor Dr. Malek, however, explain why or how it would have been within the common sense of a skilled artisan to use the fine-grained security mechanism of Gong with Usui’s resource-constrained mobile phone system. *Id.*; *see Arendi S.A.R.L. v. Apple Inc.*, 832 F.3d 1355, 1362 (Fed. Cir. 2016) (“[R]eferences to ‘common sense’ . . . cannot be used as a wholesale substitute for reasoned analysis and evidentiary support.”). Petitioners’ and Dr. Malek’s conclusory statements are unsupported and unpersuasive in light of evidence showing that the technical design constraints of a mobile phone system counseled against such a modification. *See, e.g.*, Ex. 2014, 14, 25; Ex. 2015, 8; Ex. 1001, 1:55–61. Moreover, as Patent Owner argues, it would have been “common sense” for a skilled

artisan to use Sun Microsystems' security mechanism from J2ME (sandboxing), which was specifically designed for embedded systems such as mobile telephones. *See* PO Resp. 29–30 (citing Ex. 2010 ¶ 227).

Petitioners' arguments that Patent Owner applies too rigid a test for obviousness lack merit. *See* Pet. Reply 16–20. Petitioners set forth a specific theory for motivation to combine (*see* Pet. 29–32) and it is that theory to which Patent Owner responds. Moreover, Petitioners' obviousness theory in general is premised on using the security mechanism of Gong as the security mechanism in Usui. *See, e.g.*, Pet. 42 (An ordinarily skilled artisan “would have been motivated to implement the security mechanism of Gong as the security manager of Usui Usui and Gong combine to disclose the recited functionality and identified elements of the access controller.”), 44 (“Gong discloses [the claimed] interception module, which *when implemented within the security manager of Usui*, receives a request from the requesting application domain software” (emphasis added)). Petitioners' obviousness contentions rely on implementing the fine-grained security mechanism of Gong in Usui. Patent Owner properly responded to that contention, and in particular to Petitioners' theory that a person of ordinary skill in the art would have been motivated “*to implement the access controller of Gong as the security management mechanism of Usui* to provide the fine-grained access control identified in Usui.” Pet. 30 (emphasis added).

Petitioners additionally argue that Patent Owner ignores the full disclosures of Usui and Gong—that Usui is not limited to J2ME and that Gong uses Java merely as an example. Pet. Reply 17. We are not convinced that Patent Owner ignores the full disclosure of either reference. With

respect to Usui, Figure 3 of Usui depicts an environment “for mobile phones” that runs with the CLDC/KVM—features of the J2ME platform. Ex. 1010, 4–5; *see also* Ex. 2010 ¶¶ 70–83. Patent Owner properly considers Usui’s Java Application Environment disclosure in addition to Usui’s disclosure of using J2ME and the KVM. *See, e.g.*, Ex. 2010 ¶¶ 73, 143; *see also id.* ¶¶ 70–72, 74–83; Ex. 1010, 4–5. With respect to Gong, Patent Owner’s declarant opines that Gong describes “solutions applicable in a very specific environment, to a very specific type of application[], using very specific methods of protection domains and very specific prerequisites, which are required to be present in the Java platform.” Ex. 2010 ¶ 106; *see also id.* at ¶ 105 (explaining what Gong describes in the context of the Java platform). We find this testimony persuasive and supported by Gong—a patent issued to Sun Microsystems—which describes specifics for creating and enforcing protection domains using object-oriented programming. *See* Ex. 1011, [73], 6:46–7:59; *see also* Ex. 2010 ¶¶ 84–103.

Finally, we are not persuaded by Petitioners’ reliance on an example (as described in Exhibits 1034 and 1035) of a system that purportedly ran the entire J2SE implementation on a mobile phone as early as 2001. Pet. Reply 21–22. It is unclear from those documents whether the system included the entire J2SE implementation, and in particular the security functionality specified in Gong. *See* Exs. 1034, 1035. As Dr. Muftic testified at deposition, “definitely this cannot be the Java platform with full functionality and full scope of code from the large scale IT platform,” given the inherent size constraints of a mobile phone. Ex. 1033, 215:7–23, 216:12–15; *see also id.* at 213:22–219:21. Moreover, it is unclear what was purportedly available prior to September 23, 2002, the earliest possible

effective filing date for the challenged claim and the timeframe that serves the basis for Petitioners' contentions. *See, e.g.*, Pet. 7, 14; Ex. 1034 (describing features that "will be delivered in Version 2.0 in 2002").

In sum, we are not persuaded by Petitioners' conclusory and unsupported arguments as to why a skilled artisan would have combined Usui and Gong. Accordingly, we determine that Petitioners have not provided a persuasive articulated reason with rational underpinning to support their contention that a person of ordinary skill would have had a reason or motivation to combine Usui and Gong. We thus determine that Petitioners have not shown by a preponderance of the evidence that the subject matter of claim 11 would have been obvious in view of Usui, Gong, and Ramamurthy.

e. Conclusion

Having reviewed the parties' arguments and supporting evidence, for the reasons explained above, we determine that Petitioners have not shown by a preponderance of the evidence that claim 11 is unpatentable under 35 U.S.C. § 103 as obvious in view of Usui, Gong, and Ramamurthy.

E. Asserted Obviousness over Usui, Gong, Spencer, and Ramamurthy

Petitioners also contend that claim 11 is unpatentable under 35 U.S.C. § 103 as obvious over Usui, Gong, Spencer, and Ramamurthy. Pet. 3, 53–58. For this ground, Petitioners rely on the same arguments and evidence for why a skilled artisan would have combined Usui and Gong, and rely on additional arguments and evidence in support of combining with Usui and Gong with Spencer and Ramamurthy. *See* Pet. 29–32, 54–56. For the reasons explained above in connection with the first ground, having reviewed the parties' arguments and supporting evidence, we determine that

Petitioners have not provided a persuasive articulated reason with rational underpinning to support their contention that a person of ordinary skill would have had a reason or motivation to combine Usui and Gong. We thus determine that Petitioners have not shown by a preponderance of the evidence that the subject matter of claim 11 would have been obvious in view of Usui, Gong, Spencer, and Ramamurthy.

III. CONCLUSION

Based on the evidence and arguments, Petitioners have not demonstrated by a preponderance of the evidence that (i) claim 11 of the '510 patent is unpatentable under 35 U.S.C. § 103 as obvious over Usui, Gong, and Ramamurthy or (ii) claim 11 of the '510 patent is unpatentable under 35 U.S.C. § 103 as obvious over Usui, Gong, Spencer, and Ramamurthy.

IV. ORDER

Accordingly, it is:

ORDERED that claim 11 of U.S. Patent No. 7,149,510 has not been shown to be unpatentable;

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.

IPR2015-01628
Patent 7,149,510 B2

FOR PETITIONER:

Stephen S. Korniczky
Martin R. Bader
Nam H. Kim
Hector A. Agdeppa
David A. Randall
Daniel N. Yannuzzi
David E. Heisey
Daniel T. Taskalos
SHEPPARD MULLIN RICHTER & HAMPTON LLP
skorniczky@sheppardmullin.com
mbader@sheppardmullin.com
nkim@sheppardmullin.com
hagdeppa@sheppardmullin.com
drandall@sheppardmullin.com
dyannuzzi@sheppardmullin.com
dheisey@sheppardmullin.com
dtaskalos@sheppardmullin.com
LegalTm-TCL-IPRs@sheppardmullin.com

John D. Zele
Bradford A. Cangro
Jeremy D. Peterson
Robert J. Gaybrick
Leeger Yu
MORGAN LEWIS & BOCKIUS LLP
john.zele@morganlewis.com
bradford.cangro@morganlewis.com
jeremy.peterson@morganlewis.com
robert.gaybrick@morganlewis.com
leeger.yu@morganlewis.com
TCL-Ericsson-IPRs@morganlewis.com

FOR PATENT OWNER:

W. Todd Baker
Alexander Englehart
Lisa Mandrusiak

IPR2015-01628
Patent 7,149,510 B2

Alex Kuo
OBLON, McCLELLAND, MAIER & NEUSTADT, L.L.P.
CPDocketBaker@Oblon.com
CPDocketEnglehart@Oblon.com
CPDocketMandrusiak@Oblon.com
CPDocketKuo@Oblon.com