

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

AMAZON.COM, INC. and AMAZON.COM SERVICES LLC,
Petitioner

v.

NOKIA TECHNOLOGIES OY
Patent Owner

Case No. IPR2024-01043
Patent. No. 8,918,741

PATENT OWNER'S PRELIMINARY RESPONSE

Mail Stop "PATENT BOARD"
Patent Trial and Appeal Board
U.S. Patent and Trademark Office
P.O Box 1450
Alexandria, VA 22313-1450

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	BACKGROUND	3
	A. Overview of the '741 Patent.....	3
	B. Prosecution History	3
	C. Hassan (“Hassan,” Ex. 1004)	7
	D. Becker (“Becker,” Ex. 1006).....	9
	E. Person of Ordinary Skill in the Art (POSITA).....	10
	F. Claim Construction.....	11
III.	THE BOARD SHOULD NOT INSTITUTE <i>INTER PARTES</i> REVIEW	12
	A. Petitioner Has Failed to Establish a Reasonable Likelihood that Any Challenged Claim Is Unpatentable under 37 C.F.R. § 42.108(c).	12
	1) Grounds 1 and 2 – Hassan Does Not Render Obvious Claims 1, 5-7, 10-11, 15, 17, and 19 and Hassan-Nirhamo Do Not Render Obvious Claims 1-20	12
	a) Launching a Hassan application does not teach or suggest the claims.	16
	b) Hassan’s applications are fully activated.....	19
	c) Hassan’s applications are not temporarily activated.	28
	d) Dependent claims 5-7, 10, 15, and 19.....	29

2)	Ground 2 –The combination of Hassan and Nirhamo Does Not Render Obvious Claims 1-20.....	30
3)	Ground 3 – Becker Does Not Render Obvious Claims 1, 5-7, 10-11, 15, 17, and 19	30
a)	Becker’s softphone or communications application is fully activated.	31
b)	Becker’s use of a screen saver application to restrict the fully activated softphone or communications application does not teach or suggest the claims.	37
c)	Petitioner’s ground requires each of Becker’s buttons to be present on Becker’s user interface.	40
d)	Dependent claims 5-7, 10, 15, and 19.....	42
4)	Ground 4 –The combination of Becker and Nirhamo Does Not Render Obvious Claims 1-20.....	42
IV. CONCLUSION.....		43
CERTIFICATE OF WORD COUNT		44
CERTIFICATE OF SERVICE		45

TABLE OF AUTHORITIES

	Page(s)
STATUTES	
35 U.S.C. § 102(e)	4
35 U.S.C §103(a)	4, 5
OTHER AUTHORITIES	
37 C.F.R. § 42.108(c).....	12

LIST OF EXHIBITS

Ex. Number	Title
2001	Declaration of Dr. Michael Kogan
2002	<i>Curriculum Vita</i> of Dr. Michael Kogan
2003	U.S. Patent Application Publication No. 2007/0016958 ("Bodepudi")

I. INTRODUCTION

Amazon.com, Inc. and Amazon.com Services LLC, (“Petitioner” or “Amazon”) petitioned for *inter partes* review of claims 1-20 (“Challenged Claims”) of U.S. Patent No. 8,918,741 (“’741 Patent”), which is owned by Nokia Technologies Oy (“Patent Owner” or “Nokia”). Petitioner asserts four grounds of unpatentability, but these four grounds fail for similar reasons—Petitioner relies on references that disclose fully activated applications that remain fully activated even in an alleged “locked mode.” Worse still, none of the cited references disclose temporarily or partially activated applications at all. As such, Petitioner’s references fail to teach or suggest “enabling a user to select the object while the device is in a locked mode, in order to determine a state, function or utility of the application without fully activating the application while the device remains in the locked mode.” The Board should decline institution because the Petitioner has failed to meet its burden of showing a reasonable likelihood of unpatentability for any challenged claims of the ’741 Patent.

In its first two grounds, both of which rely on Hassan alone for the independent claims, Hassan never contemplates partially or temporarily activated applications, only fully activated applications whose execution may be controlled by a task changer and user profiles. The task changer of Hassan is designed to intercept certain user requests but does not otherwise modify the activated status of

the underlying applications themselves. Because the task changer does not modify Hassan's applications, Hassan's applications themselves are fully activated and remain fully activated in Hassan's locked state.

In its final two grounds, both of which rely on Becker alone for the independent claims, Petitioner argues that Becker may impose restrictions on the softphone or communications application. But Petitioner fails to carry its burden because Becker displays the same fully activated softphone or communications application having the same graphical user interface ("GUI") elements regardless of whether the screen saver mode (the so-called locked mode of the claims) is active or not. Any restrictions to the Becker applications are restrictions imposed by a different application—the screen saver application. The restrictions imposed by the screen saver application do not change the fully activated nature of Becker's applications.

Finally, the Petition lacks any explanation or evidence to support its allegations that "[s]ince the application cannot perform its full set of operations, it is not fully activated." Petition, 25. Petitioner's expert recites the same conclusory sentence verbatim. Such conclusory statements fail, especially in view of the strong evidence presented in this paper and the testimony of Dr. Kogan. Moreover, even if an application is restricted from performing its full set of operations, such as by a separate application implementing user account controls and/or parental

controls, the application itself is still fully activated. Patent Owner requests that the Board deny institution of *inter partes* review.

II. BACKGROUND

A. Overview of the '741 Patent

The '741 Patent explains that mobile devices typically rely on a “locked mode” to avoid the inadvertent activation of applications on the device. Ex. 1001, 1:22-25. Users unlock their device with, for example, a fingerprint sensor or key combination. Until an unlocking action is taken, a device generally ignores user input. Prior to the '741 Patent, conventional technology was not able to allow interaction with applications while the device was locked, nor did it provide direct access or shortcuts to specific applications while locked. *See id.*, 1:26-28.

The '741 Patent enables a user, while in “locked mode,” to select an icon for an application to determine a state, utility, or function of the underlying application without the need to unlock the device or open the application. *Id.*, 6:56-59. In this way, the '741 Patent allows for quick activation of high-level user interface elements for passive information gathering, while the device remains in the idle and locked mode. *Id.*, 7:11-13; Ex. 2001, ¶54.

B. Prosecution History

U.S. Patent Application No. 13/889,750 (“the '750 Application”) was filed May 8, 2023, as a continuation of U.S. Application No. 13/357,143, filed on

January 24, 2012, which is a division of U.S. Application No. 11/770,929 filed on June 29, 2007 (now U.S. Patent No. 8,127,254). Ex. 1002, 53; 15-16.

On October 15, 2013, the Examiner issued a Non-Final Office Action rejecting claims 1, 7, 11-13, and 17-19 under 35 U.S.C. § 102(e) as allegedly being anticipated by U.S. Patent Application No. 2009/0205041 (“Michalske”). *Id.*, 69-71. The Examiner also rejected claims 2, 8-10, 14, and 20 under 35 U.S.C §103(a) as allegedly being obvious over Michalske in view of U.S. Patent Application No. 2007/0150842 (“Chaudhri”). *Id.*, 71-72. The Examiner also rejected claims 3, 6, and 15 under 35 U.S.C §103(a) as allegedly being obvious over Michalske in view of U.S. Patent Application No. 2010/0070931 (“Nichols”). *Id.*, 73-74. Applicants overcame these rejections by demonstrating that neither Michalske nor Nichols qualified as prior art. *Id.*, 86-87.

On May 20, 2014, the Examiner issued a Non-Final Office Action rejecting claims 1, 7-9, 12, 13, 16, 17, and 19 under 35 U.S.C. § 102(e) as allegedly being anticipated by U.S. Patent Application No. 2007/0016958 (“Bodepudi”). *Id.*, 709-711. The Examiner also rejected claims 3-5, 11, 15, and 18 under 35 U.S.C §103(a) as allegedly being obvious over Bodepudi in view of U.S. Patent Application No. 2007/0184112 (“Chiang”). *Id.*, 711-712. The Examiner also rejected claims 2, 6, 14, and 20 under 35 U.S.C §103(a) as allegedly being obvious over Bodepudi in view of U.S. Patent Application No. 2007/0150842

(“Chaudhri”). *Id.*, 712-713. The Examiner also rejected claim 6 under 35 U.S.C §103(a) as allegedly being obvious over Bodepudi in view of Chiang and Chaudhri. *Id.*, 713. The Examiner also rejected claim 10 under 35 U.S.C §103(a) as allegedly being obvious over Bodepudi in view of U.S. Patent Application No. 2007/0120689 (“Zerhusen”). *Id.*, 713-714.

Notably, with respect to “enabling a user to select the object while the device is in the locked mode, in order to determine a state, function, or utility of the application while the device remains in the locked mode,” the Examiner cited paragraph 0051 of Bodepudi, which states:

Referring now to FIG. 6, an illustrative diagram shows a user interface during selective lock mode. As illustrated, a user interface 600 during selective lock mode, as compared with user interface 500 during normal mode, only includes a selective lock folder window 620 representing selective lock folder 408 with a selectable link 622 associated with application A and a selectable link 624 associated with application D. Responsive to a non-authorized user selection of application A, selective lock application 208 calls a wrapper function that opens application A, as illustrated by application A window 602 within user interface 600 through which a user has full access to the functions of application A.

Ex. 2003, ¶0051.

Applicants filed a response to the Non-Final Office Action on June 23, 2014. Ex. 1002, 744-749. In the office action response, Applicants amended the claims to recite “enabling a user to select the ~~icon~~ object while the device is in the locked mode, in order to determine a state, function, or utility of the application without fully activating the application while the device remains in the locked mode.” *Id.*, 748-749. Applicants argued that this claim element was not taught or suggested by Bodepudi or any of the cited references. *Id.* Applicants noted that support for this claim amendment is provided in at least paragraph [0036] of the published application, which is column 6: line 56- column 7: line 13 of the issued patent and is provided below for reference. *Id.*, 748.

Referring to FIGS. 2A and 3, in one embodiment, the user can select or contact the dimmed icons 202 and 204 to determine a state, utility or function of the underlying application, without the need to unlock the device or open the application. For example, in FIG. 2A, dimmed icon 202 represents a time or clock application. In the dimmed mode, when the device is locked, it may not be possible to view any functions of the application, such as for example, the current time. However, the features of the disclosed embodiments allow the icon 202 to be contacted to temporarily illuminate or somehow highlight the icon, as illustrated by icon 302 in FIG. 3. This state, which might be described as a passive active state, allow for temporary activation of an application, such as a clock. This is also referred to herein as the "semi-active" state. As shown in FIG. 3, the clock application icon

302 is illuminated when that area is selected, and the current time is available and can be viewed, particularly against the dimmed and/or darkened background 304. In alternate embodiments, any suitable way to temporarily highlight the selected icon can be implemented. The clock function is illuminated temporarily, but does not become fully active. The icon 302 can remain illuminated for a pre-determined period of time, such as for example, 5 seconds. This allows the quick activation of high level user interface elements for passive information gathering, while the device remains in the idle and locked mode.

Ex. 1001, 6:56-7:13.

On June 24, 2014, the Examiner allowed the case without providing further comments. Ex. 1002, 757.

C. Hassan (“Hassan,” Ex. 1004)

Hassan is directed to situations where a user wants to share an unlocked device (e.g., open and accessible device) with another user without compromising security. Ex. 1004, ¶0001. To accomplish its goal of security, Hassan teaches that an unlocked device may be placed in a “locked state,” which allows the device owner to share the device with other users while maintaining security over aspects of the device. *Id.*, ¶0013.

Hassan discloses that a task changer application relies on profile data or user profiles to determine which applications a user can open or which tasks within a

fully running application that a user is permitted to execute. *Id.*, ¶¶0040; ¶¶0070. In one example of Hassan, a user associated with the user profile is only permitted by the task change API to use the voice call application and is prohibited from opening or accessing other applications on the device. *Id.*, ¶¶0040; ¶¶0002; Ex. 2001, ¶¶65. In another example, a user associated with the user profile is only permitted by the task change API to use gaming applications and is prohibited from opening or accessing other applications on the device. Ex. 1001, ¶¶0040; ¶¶0002; Ex. 2001, ¶¶66. In these examples, Hassan's "locked state" discloses the ability to allow other users to have full access to some applications while preventing access to other applications on the device. Ex. 2001, ¶¶66.

In operation, Hassan's task changer intercepts user requests based on user profile information. Specifically, Hassan states that "[t]he task changer is the software object or element that responds to that user input to execute or implement the requested operation or task. The task changer typically calls an application programming interface (API) instruction to implement or execute the requested function." Ex. 1004, ¶¶0036. Hassan is thus directed to a process that relies on a task manager for controlling access to applications on a device for other users by using user profiles that specify which applications a user is able to open and/or which tasks a user can perform within an open application while the device is in the "locked state." Ex. 2001, ¶¶69.

D. Becker (“Becker,” Ex. 1006)

Becker relates to “screen blankers” (e.g., screen savers) and in particular “an application communicating with a user when a screen blanker is activated.” Ex. 1006, 1:6-8. In Becker, a “screen blanker” or “screen saver” is a program that automatically blanks the screen of a computer and/or displays a selected image or collection of images when the computer has not been used or has been idle for a selected period of time and the monitor remains activated. *Id.*, 1:50-57.

Becker discloses an application where an IP softphone (e.g., Voice over IP (VoIP)) is running on a personal computer (PC). *Id.*, 1:12-13; Ex. 2001, ¶71.

Becker discloses that one common problem with prior systems is that a user may be unable to answer an incoming VoIP call when a password protected screen blanker or screen saver is activated on the PC. Ex. 1006, 1:48-50. Becker explains that, in prior systems, when the user hears the IP softphone ringing, the user must first unlock the screen blanker to access the PC. *Id.*, 1:61-66. The user must then select the software application and activate the soft button labeled “Answer” to answer the incoming call. *Id.* Only after completing these steps can the user use their handset to talk. *Id.*, 1:66-2:1. Becker explains that due to this time-consuming process, frequently, the call is transferred to voicemail or the caller terminates the call before the user has completed this process. *Id.*, 2:1-4; Ex. 2001, ¶73.

Becker discloses a solution that enables a PC to automatically modify a screen blanker or screen saver in response to detecting the occurrence of one or more selected events. Ex. 1006, 2:34-36; Ex. 2001, ¶74. Becker explains that when a communication device receives an event or communication (e.g., an incoming call) while the screen saver is active, the GUI associated with the softphone or communication application is displayed to permit the user to answer the communication without first disabling or terminating the screen saver. Ex. 1006, 2:64-3:1.

Becker is configured such that when an incoming communication is received by a softphone or communications application, an event notification agent is notified, which in turn notifies the screen saver application. *Id.*, 4:25-28; FIG. 3; Ex. 2001, ¶75. The screen saver application then causes the standard graphical user interface of the softphone application to be displayed by a display device even though the display device is still in the screen saver mode. Ex. 1006, 4:33-37; FIG. 3; Ex. 2001, ¶75.

E. Person of Ordinary Skill in the Art (POSITA)

Patent Owner contends that a person of ordinary skill in the art (“POSITA”) for the ’741 Patent would have had at least a bachelor’s degree in electrical engineering, computing engineering, computer science, or a related field, with two to three years of experience in user interfaces, operating systems, and device power

management. Ex. 2001, ¶25. Additional work or practical experience may qualify one not having the requisite education as a person with ordinary skill in the art while a higher level of education could offset less experience. *Id.*

While Petitioner provides a different definition of a POSITA for the '741 Patent, Patent Owner does not believe the ultimate definition of a POSITA impacts the issues presented herein for the purposes of Institution. *See* Petition at 7. As such, while Patent Owner reserves all rights to challenge Petitioner's definition of the POSITA in this proceeding, for the purposes of Institution, Patent Owner does not dispute Petitioner's definition.

F. Claim Construction

Petitioner contends that claims should be construed according to the ordinary and customary meaning as understood by a POSITA. Petition, 8. While Patent Owner agrees that the claims should be construed according to the ordinary and customary meaning as understood by a POSITA, Patent Owner also reserves the right to address construction of any terms if this proceeding is instituted and/or in any other action involving the '741 Patent.

III. THE BOARD SHOULD NOT INSTITUTE *INTER PARTES* REVIEW

A. **Petitioner Has Failed to Establish a Reasonable Likelihood that Any Challenged Claim Is Unpatentable under 37 C.F.R. § 42.108(c).**

Petitioner has failed to meet its obligations with respect to the challenged claims in the Petition. Pursuant to 37 C.F.R. § 42.108(c), Petitioner must demonstrate a reasonable likelihood “at least one of the claims challenged in the petition is unpatentable” to warrant institution of an *inter partes* review for that claim. Petitioner has not done so.

1) Grounds 1 and 2 – Hassan Does Not Render Obvious Claims 1, 5-7, 10-11, 15, 17, and 19 and Hassan-Nirhamo Do Not Render Obvious Claims 1-20

Petitioner alleges that Hassan alone teaches or renders obvious independent Claims 1, 11, and 17.¹ *See* Petition, 8-9; 12-26. Petitioner’s allegations fail.

Claim 1 of the ’741 Patent recites: “enabling a user to select the object while the device is in the locked mode, in order to determine a state, function or utility of the application *without fully activating the application* while the device remains in the locked mode.” In this way, the ’741 Patent claims provide user-access to

¹ Petitioner has grouped Grounds 1 and 2 together for the purposes of its Petition. Petition, 8-9. The Petition relies on Hassan alone in its assertion that claims 1, 11, and 17 are obvious and Nirhamo is only applied to the dependent claims. *Id.*

specific functionality of an application that is not fully activated, without unlocking the device the device itself. Ex. 2001, ¶78.

The determination of a state, function or utility of the application *without fully activating the application* refers to a constraint on the type of information that can be viewed in the locked mode because the claims require the application to not be fully activated. Ex. 2001, ¶79. The '741 Patent provides multiple examples of such a constraint when determining a state, function or utility of the application. For example, “the icon 204 of FIG. 2A, representing a communication application, opens to user interface elements for contacts 222 and applications 224. This provides the user with the advantage of determining, prior to unlocking a device from an inactive state, an application or set of user interface elements for the device to open up to.” Ex. 1001, 6:48-53. Here, high level user interface elements, such as contacts 222 and applications 224, provide high level user interface elements that provide information regarding the functions of the fully activated communications application. Ex. 2001, ¶79.

In addition, the '741 Patent provides a second example related to an icon 202, which represents a time or clock application. Ex. 2001, ¶80. In this example, “when the device is locked, it may not be possible to view any functions of the application, such as for example, the current time.” Ex. 1001, 6:61-63. However, when the clock application is activated to determine a state, function or utility, the

“clock application icon 302 is illuminated when that area is selected, and the current time is available and can be viewed, particularly against the dimmed and/or darkened background 304.” *Id.*, 7:3-6. Here again, the high-level user interface element, the clock, is presented to the user but the application itself is not fully activated. Ex. 2001, ¶80. In fact, the ’741 Patent teaches “the quick activation of high level user interface elements for passive information gathering, while the device remains in locked mode” Ex. 1001, 7:11-13.

In contrast to the claims and the ’741 Patent specification, neither Hassan nor Becker discloses or contemplates partially or temporarily activated applications. Ex. 2001, ¶81. This lack of teaching alone should be fatal to all of the asserted grounds.

Hassan and Becker disclose standard applications (e.g., Windows applications) that are stored in memory and fully activated upon execution. *Id.* In each reference, the applications, even in a locked state / screen saver mode, include the same complete and unconstrained GUI used in an unlocked state / non-screen saver mode. *Id.* As such, neither reference contemplates the quick activation of high level user interface elements or operating an application in less than a fully activated state. *Id.*

Moreover, both references rely on a separate application or functions that operates to restrict a user request (e.g., a selection of an icon) based on

permissions. *Id.*, ¶82. But neither the claims nor the specification of the '741 Patent contemplate restrictions that are imposed by another application (e.g., a task changer / screen saver application) result in an application that is not fully activated. *Id.* Instead, similar to Windows restrictions imposed at the user/administrator level (e.g., certain functions within an application are reserved for administrators), the restrictions of the cited references merely limit what task(s) that a user is ultimately able to perform without affecting the fully activated nature of the application.² *Id.*

As such permissions (or lack thereof) from a *separate* application that result in certain functions of a fully activated application being restricted or blocked does not change the underlying fact that the application itself is fully activated and able to execute all user requests that are passed to it. *Id.*, ¶83.

The Petition never addresses the impact of the *separate* applications in both Hassan and Becker and instead merely asserts that “[s]ince the application cannot perform its full set of operations, it is not fully activated.” Petition, 25. But such a broad sweeping and unsupported statement overlooks the evidence in this case and cannot support a finding of unpatentability.

² Petitioner does not put forth a proposed construction for “without fully activating the application.”

For these reasons and the reasons below, Petitioner fails to show that the cited references teach or suggest the claims of the '741 Patent.

a) Launching a Hassan application does not teach or suggest the claims.

Petitioner argues that “Hassan teaches enabling a user to select the icon for an application to perform operations such as launching an application when the user requests a task change.” Petition, 23. However, Hassan never discloses or contemplates partially or temporarily activated applications. Ex. 2001, ¶106. Instead, Hassan only discloses fully activated applications having unconstrained user interface elements whose execution may be controlled by a task changer. As such, launching or opening an application to such a fully activated state, even if in the locked mode, does not teach or suggest “determin[ing] a state, function or utility of the application *without fully activating the application*,” as recited by the claims. *Id.*, ¶85.

In addition, Petitioner’s bare assertion that opening an application while the Hassan device is in a “locked state” meets the claimed “without fully activating the application” is analogous to a rejection issued during prosecution. Specifically, the Examiner relied on paragraph 51 of Bodepudi, which disclosed a fully activated application that launched in a wrapper application (which the Examiner appeared

to have likened to a “locked mode”) to reject the then pending claims.³ Ex. 1002,

709. Paragraph 51 of Bodepudi states:

Referring now to FIG. 6, an illustrative diagram shows a user interface during selective lock mode. As illustrated, a user interface 600 during selective lock mode, as compared with user interface 500 during normal mode, only includes a selective lock folder window 620 representing selective lock folder 408 with a selectable link 622 associated with application A and a selectable link 624 associated with application D. Responsive to a non-authorized user selection of application A, *selective lock application 208 calls a wrapper function that opens application A*, as illustrated by application A window 602 within user interface 600 *through which a user has full access to the functions of application A*.

Ex. 2003, ¶0051 (emphasis added); *see also* ¶0034; ¶0041; ¶0042.

Based on the disclosure of Bodepudi, a POSITA would have understood, and the examiner did understand, that the Bodepudi application was fully activated even though the overall functionality was constrained by the wrapper application.

Ex. 2001, ¶87. In response to Bodepudi, Applicants amended the claims to recite “without fully activating the application.” *Id.*, ¶88. As a result of Applicants’

³ The then pending claims did not include the phrase “without fully activating the application.”

amendments, and in view of the claim as a whole, Applicants overcame the then pending rejection in view of Bodepudi. *Id.*, ¶89. Thus, by amending the claims to recite that the claimed application is not fully activated, Applicants distinguished the independent claims as a whole from references where a fully activated application was launched in a wrapper application that intercepts user actions and selectively blocks access. *Id.*

Like Bodepudi, which discloses a wrapper function that opens a fully activated application while still restricting the user, Hassan discloses the use of a type of wrapper function, a task changer, to restrict access to certain fully activated applications. *Id.*, ¶90. Because Hassan's task changer is operating in Hassan's "locked state" and restricts access to what applications can be launched it is the same as or equivalent to Bodepudi's wrappers restricting what applications can be launched, launching an application in Hassan when the user requests a task change thus does not satisfy the claim limitation "without fully activating the application." *Id.*

As such, merely launching an application while in Hassan's locked state does not satisfy "determin[ing] a state, function or utility of the application ***without fully activating the application,***" as recited by the claims.

b) Hassan’s applications are fully activated.

Petitioner asserts that any restriction in functionality, even if it is based on permissions (or lack thereof)⁴, results in an application that is not fully activated. Petition, 25. Specifically, Petitioner asserts that “Hassan teaches that the application (e.g., e-mail) is not fully activated because users may only access a limited set of ‘permitted’ tasks or operations within the application, rather than the full set of operations.” *Id.* Moreover, Petitioner and its expert both incorrectly assert that “[s]ince the application cannot perform its full set of operations, it is not fully activated.” *Id.* However, Hassan never contemplates partially or temporarily activated applications. Ex. 2001, ¶91. Instead, Hassan discloses fully activated applications whose execution is restricted by a task changer. *Id.* Such restriction to a limited set of “permitted” tasks by a task changer fails to teach or suggest “determin[ing] a state, function or utility of the application ***without fully activating the application***,” as recited by the claims.

Hassan teaches a task changer and task change API which controls user access to fully loaded and activated applications and their operations while Hassan is operating in a “locked state.” *Id.*, ¶92. For example, Hassan discloses “[t]ask

⁴ Hassan relies on profile data, which defines the permitted and/or impermissible tasks associated with a locked state. Ex. 1004, ¶0039.

change API calls by a task changer, such as a task switcher or window launcher, may be evaluated by a process control level of the operating system to determine whether the requested task change is permitted.” Ex. 1004, ¶0014. Hassan’s task change API that is used by a task changer uses a process control level of the operating system to determine which aspects of the fully activated application’s functions are permitted to be used or not. Ex. 2001, ¶93; *see also* Ex. 1004, ¶0037 (“The term “task change” or “task changer” as used herein is meant to encompass both a change in the active window as between two or more open windows, and the launch or opening of a new window. In some embodiments, ***it is also meant to encompass selection of a specific operation within an application, where the operation implements an application-specific feature.***” (emphasis added)). None of these disclosures of Hassan indicate that the underlying application in Hassan is modified but instead merely indicate that Hassan’s task changer intercepts user requests.⁵

⁵ Similar to Windows restrictions imposed at the user/administrator level (e.g., certain functions within an application are reserved for administrators), the restrictions of the cited reference merely limit what task(s) that a user is able to perform without affecting the fully activated nature of the application (e.g., the

Hassan provides examples illustrating that the user interacts with a Hassan application in a normal and expected manner (e.g., all of the operations/functions of the application are displayed and available for selection), which indicates to the POSITA that the Hassan application itself is fully activated. *Id.*, ¶94. For example, Hassan discloses “in an e-mail application, a user may request a certain function like forwarding a message, replying to a message, opening a message, through an appropriate user input (like clicking on a designated icon).” Ex. 1004, ¶0036. Regardless of whether Hassan’s device is in a “locked state” or an unlocked state, Hassan’s application thus includes all of the relevant functionality because the user is able to, for example, activate an icon to reply to a message. Ex. 2001, ¶94. The only difference in Hassan’s “locked state” is that the task changer processes task change requests (e.g., the selection of a “reply” icon) and may intercept the task request unbeknownst to the fully activated application to request user authentication.

ability for a user to select an icon to launch a task from within the application even though that task was ultimately prevented by the OS). *See* Ex. 2001, ¶82

The activated state of Hassan's applications remains the same in the locked state as it would be in the unlocked/open and accessible state.⁶ *Id.*, ¶95. That is, the presence of the task changer does not alter the fact that Hassan's applications are fully activated. *Id.* The sole difference in the locked state is that the background or wrapper application (e.g., task changer) is intercepting each request and confirming whether or not a particular task or operation is authorized before it allows a task to be executed. *Id.* Figure 6 of Hassan confirms this understanding of Hassan's teachings and is provided below for reference. *Id.*

⁶ There should not be any dispute that an application in Hassan is fully activated when the Hassan device is in an unlocked state. As such, because there is no difference in a Hassan application between the locked state and the unlocked state, it follows that the Hassan application remains fully activated in the locked state. *See Ex. 2001*, ¶95.

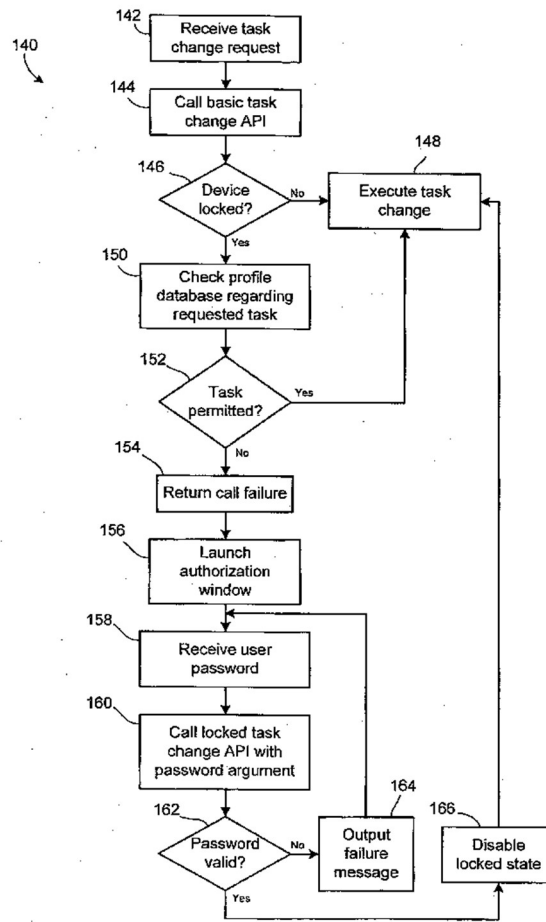


FIG. 6

Ex. 1004, FIG. 6.

Hassan discloses that “[t]he task change request is received in step 142.” Ex. 1004, ¶0064. For there to be a task change request, there must be a request from a user to select an application, an operation within an application, or a task change. Ex. 2001, ¶97. A POSITA would have understood that for the user to make such a request the menu or icons of available functions, such as functions of an applications, must be displayed. *Id.* Additionally, the ability to display all the

available functions or operations would also suggest to a POSITA that an application is fully activated. *Id.*

Hassan provides numerous examples of its applications being fully activated in that each available operations (e.g., icons) are activated and shown to a user. Ex. 2001, ¶98. For example, “within an e-mail messaging program, clicking on a message in an ‘inbox’ view launches a new window for displaying the selected message. Within the displayed message, selection of a ‘reply’ or ‘forward’ button or menu item launches yet another window for composing the reply or forward message.” Ex. 1004, ¶0035. In this example, the application publishes the “reply” or “forward” button and it is the Hassan task changer that intercepts any user input related to these buttons so as to create the “task change request” that is processed at step 142. Ex. 2001, ¶98. The fact that the application publishes the “reply” or “forward” button even though they may ultimately be unauthorized by the task changer indicates to the POSITA that the Hassan application is fully activated despite the presence of the task changer. *Id.*

Referring again to FIG. 6 of Hassan and after a task change request is received, Hassan discloses that “process control layer 64 assesses whether the device is locked in step 146” and “[i]f not, the change request is passed to the operating system for implementation, as is shown in step 148.” Ex. 1004, ¶0064. Here, the POSITA would understand that the Hassan device is unlocked and

passing a “change request” or task request that originated from an application to the operating system would result in the device operating normally. Ex. 2001, ¶100. That is, in response to a request for a new task or application, the operating system would process the request and execute the task or open the application in the normal course. *Id.* Such an operation bolsters a POSITA’s understanding that the application is fully activated and is operating as normal even though the task changer is performing the steps of Figure 6. *Id.* Hassan does not disclose or suggest that there is any change to Hassan’s application to be able to perform the tasks, only that the intercepted task is passed along to the application for execution. *Id.*

Continuing with Figure 6 of Hassan, if the device is locked at step 146, the process control layer determines whether the requested task change is a permitted task change. Ex. 1004, ¶0065. Hassan discloses that “[t]his evaluation as to whether the task is permitted, as shown in step 152, may be based upon the identity of the application program requested, the type of task or operation being performed, or other factors.” *Id.* Regardless, if the task change is permitted, “the method 140 continues to step 148, whereupon the task change API call is passed through to the operating system 52 for implementation of the requested change.” *Id.*; Ex. 2001, ¶101. Once again, because a task change is implemented just the same as it was implemented in an unlocked state, Hassan is teaching that the application is fully activated and is operating as normal even though the task

changer is intercepting any requests and performing the steps of Figure 6. Ex.

2001, ¶101.

Hassan further discloses that, if the task is impermissible, “in step 156, the locking component 60 launches an authorization window to prompt the user to provide input authorization data, such as a password. Through the authorization window, the locking component 60 receives user input in step 158.” Ex. 1004, ¶0066. The presence of an authorization window does not change the fact that the application itself is still fully activated and is operating as normal because the user was able to request an unauthorized task or operation in the application. Ex. 2001, ¶102. Hassan confirms this understanding by stating that “[i]f the password is validated, then in step 164 the lock state may be disabled and the method 140 proceeds to step 148 to implement the requested task change.” Ex. 1004, ¶0067. That is, upon authorization, the “change request” or task request is forwarded to the operating system for implementation, and again, there is no change to the application contemplated upon authorization only that the task changer will process a task request. Ex. 2001, ¶102.

Even if the task request is ultimately restricted by the task changer, nothing has changed with the application and its status of being fully activated between steps 146 and 156 of Figure 6. *Id.*, ¶103. The fully activated application at step 146 remains fully activated at step 156 such that, from the application’s perspective,

nothing has changed at all (e.g., the user interface and application functionality has not changed). *Id.* Indeed, as discussed above, all of the user interface elements of the Hassan application are active and displayed. The only distinction or change that has occurred in the Hassan “locked state” is that a separate application or function, a task changer, has launched an authorization window as a result of the intercepted request. *Id.* This authorization window has not changed the activated status of the application or altered the application, but instead has intercepted a task request before it is implemented by the operating system even though the fully activated application is itself able to execute the task. *Id.* As such, the POSITA would have understood that the Hassan application is fully activated. *Id.*

Thus, Figure 6 of Hassan demonstrates that, regardless of success or failure of a task request, an application is fully activated in Hassan. *Id.*, ¶105. Hassan is configured such that, from the applications perspective, the user is able to execute a change request or task request, such as by selecting an icon within the application, rendering the application fully activated. *Id.* Regardless of the decision made by the task changer, the application itself remains fully activated. *Id.*

Petitioner never addresses these disclosures in Hassan. Instead, Petitioner merely alleges, without support, that “[s]ince the application cannot perform its full set of operations, it is not fully activated.” Petition, 25. Petitioner’s expert fares no better. Such conclusory testimony cannot stand in the face of the evidence

presented herein. Especially evidence that confirms that the application itself is fully activated. In the face of overwhelming evidence in Hassan, Petitioner's theory conclusory theory fails and Petitioner cannot carry its burden.

Hassan never contemplates partially or temporarily activated applications, only fully activated applications. Ex. 2001, ¶106. As a result, Hassan's fully activated applications fails to teach or suggest "without fully activating the application" as is recited in claims 1, 11, and 17. *Id.*

c) Hassan's applications are not temporarily activated.

Petitioner asserts that "Hassan teaches that the application (e.g., email) is not fully activated while the device remains in locked mode; instead, it is *temporarily* activated with limited functionality, which serves the underlying purpose of Hassan." Petition, 26 (emphasis added). However, this assertion is also wrong. As discussed above, Hassan's applications are fully activated, and furthermore, Hassan does not teach or suggest that the activation of its applications is temporary in nature. Ex. 2001, ¶107. Instead, as Petitioner admits, Hassan only discloses a locked mode that may be temporarily activated. *Id.*, ¶109.

While the independent claims of the '741 Patent do not recite the term "temporary," the '741 Patent discloses that an application may be temporarily activated when the device is in the locked mode. For example, the '741 Patent discloses "[a]n application that does not become active, or fully active, when

selected, such as a clock that can be temporarily illuminated, may be displayed with a 60% illumination level.” Ex. 1001, 3:65-4:1. The ’741 Patent further discloses “[t]he clock function is illuminated temporarily, but does not become fully active.” *Id.*, 7:8-9. Based on the teachings of the ’741 Patent, certain aspects of an application may be temporarily activated, but the application does not become fully activated when the device is in the locked mode. Ex. 2001, ¶108. In contrast to the ’741 Patent, Hassan discloses, and Petitioner confirms, that it is not the applications that are temporarily activated but rather it is Hassan’s “locked mode [that] is only temporary.” Petition, 26. In Hassan, any application that is opened will remain fully activated until it is closed, which would not be considered temporary to a POSITA. Ex. 2001, ¶109.

Hassan never contemplates partially or temporarily activated applications, only fully activated applications. *Id.*, ¶106. As a result, Hassan’s fully activated applications fail to teach or suggest “without fully activating the application” as is recited in claims 1, 11, and 17. *Id.*

d) Dependent claims 5-7, 10, 15, and 19

Dependent claims 5-7, 10, 15, and 19 depend from independent claims 1, 11, and 17, respectively. As discussed above, Hassan fails to disclose or render obvious independent claims 1, 11, and 17. For at least this reason, Hassan also fails

to disclose or render obvious dependent claims 5-7, 10, 15, and 19 and Nirhamo has not been alleged to and does not make up for the shortcomings in Hassan.

2) Ground 2 –The combination of Hassan and Nirhamo Does Not Render Obvious Claims 1-20

Petitioner alleges that the combination of Hassan and Nirhamo teaches or renders obvious Claims 1-20. *See* Petition, 8-9. However, Nirhamo is only applied to the dependent claims. As discussed above, Hassan fails to disclose or render obvious at least independent claim elements 1[c], 11[c], and 17[c]. Nirhamo alone or in combination with Hassan does not cure the deficiencies of Hassan. For at least this reason, the combination of Hassan and Nirhamo also fails to disclose or render obvious Claims 1-20.

3) Ground 3 – Becker Does Not Render Obvious Claims 1, 5-7, 10-11, 15, 17, and 19

Petitioner alleges that Becker alone teaches or renders obvious independent Claims 1, 11, and 17.⁷ *See* Petition, 43-44; 46-60. Petitioner's allegations fail.

⁷ Petitioner has grouped Grounds 3 and 4 together for the purposes of its Petition. Petition, 43-44. The Petition relies on Becker alone in its assertion that claims 1, 11, and 17 are obvious and Nirhamo is only applied to the dependent claims. *Id.*

a) Becker's softphone or communications application is fully activated.

Petitioner asserts that: “Becker teaches that the communication application is not fully activated because the user may access only a restricted set of the features that only ‘allow[s] for limited calling with specified call permissions (or limited calling to specified destinations) while the PC is password protected.” Petition, 57. Moreover, Petitioner and its expert both incorrectly assert that Becker’s application is not fully activated because Becker discloses “limiting user access ‘to a restricted set of features of the communication application.’” Petition, 57-58; Ex. 1003, ¶130. But Becker does not disclose partially or temporarily activated applications. Instead, Becker discloses an application, a softphone or communications application, that is fully activated in the screen saver mode or not. Becker thus fails to teach or suggest the claims.

Becker discloses a “screen saver mode,” where a personal computer may “modify automatically a screen saver in response to the occurrence of one or more selected events.” Ex. 1006, 2:34-36. Becker further discloses that “[w]hen the communication device is receiving the communication, *a graphical user interface associated with the communication application is displayed* to permit the user to answer the communication without first disabling or terminating the screen saver mode.” *Id.*, 2:61-3:1 (emphasis added). In operation, Becker provides an “event

notification agent” that notifies a screen saver application when an incoming communication is received so that the screen saver application can cause the GUI of the softphone or communications application to be presented. *Id.*, 4:25-28, 4:33-38, 5:1-16; Ex. 2001, ¶115. The event notification agent and the screen saver application control the presentation of the GUI and any interactions therewith. Ex. 2001, ¶¶114-115.

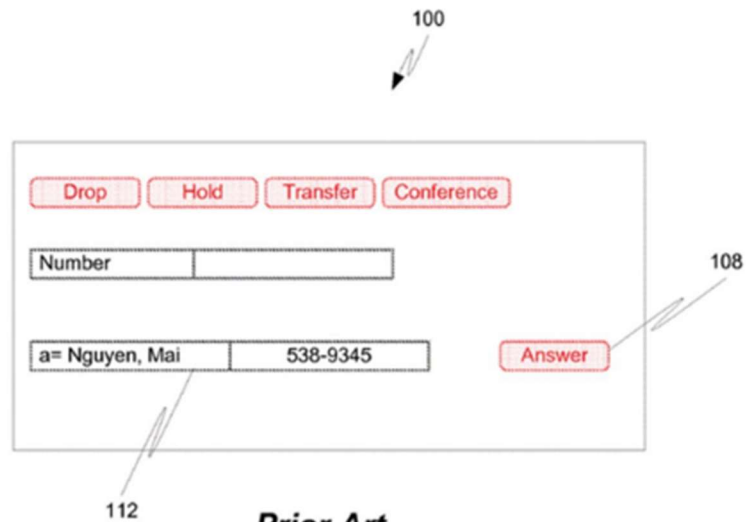
Details of Becker’s GUI of the softphone or communications application are disclosed with respect to Figures 1A, 1C, and 1D of Becker. Ex. 2001, ¶116. For example, Becker discloses:

With reference to the Avaya IP Softphone™ display 100 of FIG. 1A, the user may answer the call by selecting the softphone software application (such as by clicking on the corresponding icon 104 of FIG. 1B) and selecting a button 108 labeled “answer”. Before and after the call is answered, the caller's name and number, if available, is depicted in display field 112. The user then conducts a conversation with the caller via either the handset or the headset. When the call is answered, the display changes automatically to the display 120 of FIG. 1D, which includes buttons for drop 124, hold 128, transfer 132, conference 136, and hangup 140. The call is terminated by the called user by selecting the hangup button 140. With reference to the Avaya IP Softphone™ display 150 of FIG. 1C, to place a call the user once again selects the software and selects a soft button 154 labeled “Number” and proceeds to dial the destination number using a keypad

or a pre-stored list of telephone numbers. The number appears in the field 158. The keypad can be either a soft keypad or the PC's keyboard.

Ex. 1006, 1:29-47. Becker further discloses that, when the device is in screen saver mode, the user can receive and place outgoing calls using the same fully activated softphone or communications application GUI as described above in Becker at column 1, lines 29-47. *Id.*, 2:61-3:1; 6:52-54; Ex. 2001, ¶116.

The Petition provides an annotated Figure 1A of Becker as an example of the GUI that is displayed in the screen saver mode. Petition, 51-52; Ex. 1003, ¶121. As shown in this example, the unconstrained GUI of Becker includes all the user interface elements as they would be included if the softphone or communications application was operating outside of Becker's screen saver mode. Ex. 2001, ¶117. According to Petitioner, there are no constraints to the Becker GUI. *See* Petition, 51-52.



**Prior Art
Fig. 1A**

Petition, 52.

Petitioner's expert confirms that Becker relies on this same fully activated and unconstrained GUI application window when operating in the screen saver mode. Ex. 1003, ¶121 (citing Ex. 1006, 1:29-47, 3:40-42,3:50, 4:25-40, 5:25-47, 6:4-16, 6:48-7:17, Figs. 1A-1D, Fig. 5); Ex. 2001, ¶117. There can be no dispute that the softphone or communications application is fully activated outside of the screen saver mode. Ex. 2001, ¶117. As such, the fact that an application itself operates the same in both the screen saver mode and out of it suggests that the application is also fully activated in the screen saver mode. *Id.*

Similar to how a POSITA would have understood an application to be run in a typical operating system environment, such as a Microsoft Windows

environment, a POSITA would have understood that Becker's softphone or communications application is stored in memory and fully activated when executed. Ex. 2001, ¶119. Support for this understanding is illustrated in Figure 3 of Becker. Becker discloses that "FIG. 3 depicts, in block diagram form, the structure of the software instructions stored in the memory 322 (or a computer readable medium) of the PC 216 and executed by the processor 312 to provide functions for the handset or for the headset and for the display device 316." Ex. 1006, 4:51-55. As shown in Figure 3 of Becker, softphone application 300 is one example of a software instruction that is stored in memory. Ex. 2001, ¶119.

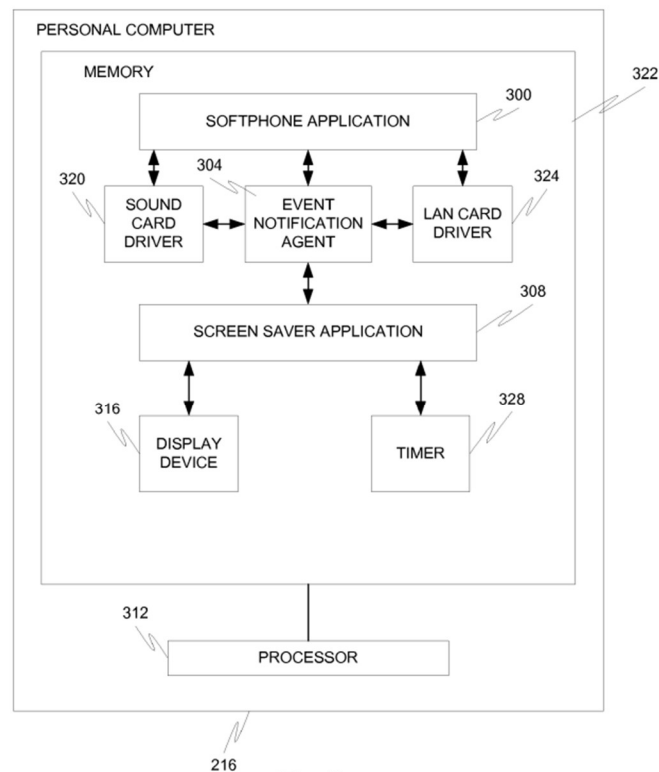


Fig. 3

Ex. 1006, Fig. 3.

A POSITA would have further understood that the softphone or communications application of Becker is fully activated in Becker because it is configured to continuously monitor for incoming communications. Ex. 2001, ¶120. For example, Becker discloses that “[w]hen an incoming communication is received by softphone application intended for the user/subscriber of the PC 216, an event notification agent 304 (FIG. 3) is notified, which in turn notifies a screen saver application 308 (FIG. 3).” Ex. 1006, 4:25-28; 5:1-9.

In Becker, when an incoming communication is detected, the fully activated softphone or communications application configures its complete and unconstrained GUI to be displayed in the Becker screen saver mode. Ex. 2001, ¶121. For example, Becker discloses that “[i]n response, the screen saver application 308 causes the GUI of the softphone application, such as the GUI of FIGS. 1A and 1B, to be displayed by the display device 316 (FIG. 3), even though the display device is still in screen saving mode.” Ex. 1006, 4:33-37. As such, a POSITA would have understood that Becker’s softphone or communications application would be fully activated for its complete GUI and unconstrained GUI to be displayed. Ex. 2001, ¶121.

Finally, Becker discloses the use of a type of wrapper function, a screen saver application, to restrict access to certain applications, which is also analogous

to the wrapper function used in Bodepudi during prosecution. *Id.*, ¶122. As discussed above with respect to Ground 1 and Bodepudi during prosecution, Applicants distinguished the claims from references where a fully activated application was bounded by a wrapper application.

As such, the Becker application does not satisfy the claim limitation “determin[ing] a state, function or utility of the application *without fully activating the application*,” as recited by the claims.

b) Becker’s use of a screen saver application to restrict the fully activated softphone or communications application does not teach or suggest the claims.

Petitioner asserts that “Becker further teaches enabling a user to select the object (e.g., button) in order to determine a state, function or utility of the application **without fully activating the application** while the device remains in the locked mode.” Petition, 57 (emphasis in original). In support for this conclusory assertion, Petitioner relies on Becker’s teaching of limiting user access to “a restricted set of the features of the communications application” while the PC is in the screen saver mode and the buttons of Becker’s communication application to demonstrate that Becker enables a user to select an object. Petition, 56-58.

However, any restrictions on the softphone or communication application are imposed by the Becker screen saver application. As such, the identified teachings of Becker to limit or restrict the softphone or communication applications do not

change the fact that Becker's softphone or communications application is fully activated. Ex. 2001, ¶¶124-125.

Becker discloses that it is the *separate* screen saver application, in conjunction with the event notification agent, that controls any restrictions. *Id.*, ¶126. For example, Becker discloses "Softphone application 300 provides the overall control of the communication of calls with PBX 204. Screen saver application 308 provides the ***overall control of the display device 316*** as discussed below. The event notification agent 304 provides the interface between the softphone and screen saver applications 300 and 308." Ex. 1006, 3:61-68 (emphasis added). Becker further discloses "If so, the [screen saver] application, in step 422, displays the GUI for the communication application (e.g., softphone or instant messaging application) (without receiving the defined information from a user) but ***continues to restrict the user's access to a defined set of applications.***" *Id.*, 6:21-26 (emphasis added). Notably, other than the teachings that the screen saver application imposes restrictions, Becker provides no other teachings with regards to how its "restrictions" would be implemented and neither Petitioner nor its expert identify such a teaching. Ex. 2001, ¶126.

Instead of identifying any specific teaching in Becker to support its theory regarding how Becker restricts an application's set of features, Petitioner asserts, without support, that "[]Becker enables users to select the

drop/hold/transfer/conference buttons to determine whether or not the application supports the corresponding functionality during screen saver mode.” Petition, 56. Assuming Petitioner’s reading of Becker is correct, Petitioner’s interpretation thus necessarily relies on the presence of each of the drop/hold/transfer/conference buttons and a *separate* application to restrict a button’s corresponding functionality. Ex. 2001, ¶127. That is, even under Petitioner’s apparent reading, the *separate* screen saver application of Becker is configured to restrict a user’s access to the functionality of some buttons such that a user could select a button like “transfer” and, after selection, “determine whether or not the application supports the corresponding functionality during screen saver mode.” See Ex. 1006, 3:61-68, 6:21-26; *see also* Ex. 2001, ¶126.

Similar to what was discussed above with respect to Hassan and Grounds 1-2, restrictions that are implemented by a separate application (e.g., a task changer or a screen saver application) do not change the fact that the underlying application (e.g., the softphone or communications application) is fully activated, is displaying its complete and unconstrained GUI, and allows for selection of any of its buttons by a user. Ex. 2001, ¶127. That is, because the buttons are active in the softphone or communications application and it is the *separate* screen saver application and the event notification agent that is preventing the execution of certain available features shown on the Becker GUI, there are no restrictions to the Becker

communications application itself and, thus, it is fully activated just as it would be outside of the screen saver mode. *Id.*

Thus, Becker's softphone or communications application is fully activated and, if anything, restricted by the *separate* screen saver application. *Id.* Such restrictions do not change the fully activated nature of the application. *See* Section 3.a, *infra.*

Petitioner never establishes that restrictions imposed by the *separate* screen saver application change the fully activated nature of the Becker softphone or communications application. Instead, Patent Owner has presented evidence that confirms that the softphone or communications application itself is fully activated. In the face of overwhelming evidence presented herein, Petitioner's conclusory theory fails.

For at least these reasons, Becker fails to teach or suggest "without fully activating the application" as is recited in claims 1, 11, and 17.

c) Petitioner's ground requires each of Becker's buttons to be present on Becker's user interface.

To the extent that Petitioner argues that Becker restricts its application by removing functionality (e.g., buttons) from its GUI in the screen saver mode, this interpretation also fails and is inconsistent with how a POSITA would read and interpret Becker. Ex. 2001, ¶129.

In its application of Becker to the claim element, “enabling a user to select the object while the device is in the locked mode,” Petitioner relies on the selection of a “button in the GUI display” of Becker. Petition, 54. More specifically, Petitioner identifies that “Becker teaches that users select the GUI display, including, for example, ‘Answer,’ ‘Drop,’ ‘Hold,’ ‘Transfer,’ ‘Conference’ buttons, to perform permitted operations when the PC is in screen saver mode.” *Id.* Petitioner only identifies and relies on the buttons within the GUI of Becker for its arguments. Ex. 2001, ¶130. That is, Petitioner does not identify any other “objects” that would teach or suggest “enabling a user to select the object while the device is in the locked mode.” *Id.* Based on the alleged presence of these buttons, Petitioner asserts “Becker enables a user to select its buttons, to determine a function of the application (e.g., the ability to answer, drop, hold, transfer, or conference incoming calls) during screen saver mode.” Petition, 56. As such, Petitioner’s alleged rejection relies on each of Becker’s buttons being present so that they are selectable by the user to determine whether or not the application supports the corresponding functionality during screen saver mode. Ex. 2001, ¶131.

In the event that Petitioner argues that Becker’s restriction means that a restricted button (e.g., dial or answer) is removed from the GUI, then Petitioner’s reliance on selection of such a button to determine its functionality is no longer possible. *Id.* As such, if Becker was interpreted to remove a restricted button from

its GUI and to keep unrestricted buttons on the GUI, then Petitioner would no longer be able to show “enabling a user to select the object while the device is in the locked mode.” *Id.*, ¶132. Thus, Petitioner proposed rejection also fails under this interpretation. *Id.*

d) Dependent claims 5-7, 10, 15, and 19

Dependent claims 5-7, 10, 15, and 19 depend from independent claims 1, 11, and 17, respectively. As discussed above, Becker fails to disclose or render obvious independent claims 1, 11, and 17. For at least this reason, Becker also fails to disclose or render obvious dependent claims 5-7, 10, 15, and 19.

4) Ground 4 –The combination of Becker and Nirhamo Does Not Render Obvious Claims 1-20

Petitioner alleges that the combination of Becker and Nirhamo teaches or renders obvious Claims 1-20. *See* Petition, 43-44. However, Nirhamo is only applied to the dependent claims. As discussed above, Becker fails to disclose or render obvious at least independent claim elements 1[c], 11[c], and 17[c]. Nirhamo alone or in combination with Becker does not cure the deficiencies of Becker. For at least this reason, the combination of Becker and Nirhamo also fails to disclose or render obvious Claims 1-20.

IV. CONCLUSION

For the above-stated reasons, Patent Owner requests that the Board deny institution of the Petition.

Dated: October 24, 2024

Respectfully submitted,

/Christopher TL Douglas/
Christopher TL Douglas
Reg. No. 56,950
christopher.douglas@alston.com

John Haynes
Reg. No. 44,754
john.haynes@alston.com

Nicholas T. Tsui
Reg. No. 71,055
nick.tsui@alston.com

Dana Zottola
Reg. No. 65,942
dana.zottola@alston.com

Bryan Rodriguez
Reg. No. 71,274
bryan.rodriguez@alston.com

*Attorneys for Patent Owner,
Nokia Technologies Oy*

CERTIFICATE OF WORD COUNT

Under the provisions of 37 C.F.R. § 42.24, the undersigned hereby certifies that the word count for the foregoing Patent Owner Preliminary Response totals 9,047 words, which is less than the 14,000 allowed words under 37 C.F.R. § 42.24(a)(1)(i),(b)(1).

Dated: October 24, 2024

Respectfully submitted,

/Christopher TL Douglas/
Christopher TL Douglas
Reg. No. 56,950
christopher.douglas@alston.com

John Haynes
Reg. No. 44,754
john.haynes@alston.com

Nicholas T. Tsui
Reg. No. 71,055
nick.tsui@alston.com

Dana Zottola
Reg. No. 65,942
dana.zottola@alston.com

Bryan Rodriguez
Reg. No. 71,274
bryan.rodriguez@alston.com

*Attorneys for Patent Owner,
Nokia Technologies Oy*

CERTIFICATE OF SERVICE

The undersigned hereby certifies that the foregoing **PATENT OWNER'S PRELIMINARY RESPONSE** was served electronically via e-mail on October 24, 2024, in its entirety on the following counsel of record for Petitioner:

Harper Batts	hbatts@sheppardmullin.com
Chris Ponder	cponder@sheppardmullin.com
Jeffrey Liang	jliang@sheppardmullin.com

Electronic Service Address LegalTM-Amazon-Nokia-IPRs@sheppardmullin.com

Dated: October 24, 2024

Respectfully submitted,

/Christopher TL Douglas/
Christopher TL Douglas
Reg. No. 56,950
christopher.douglas@alston.com
John Haynes
Reg. No. 44,754
john.haynes@alston.com
Nicholas T. Tsui
Reg. No. 71,055
nick.tsui@alston.com
Dana Zottola
Reg. No. 65,942
dana.zottola@alston.com
Bryan Rodriguez
Reg. No. 71,274
bryan.rodriguez@alston.com

*Attorneys for Patent Owner,
Nokia Technologies Oy*