

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

LENOVO (UNITED STATES) INC. and
MOTOROLA MOBILITY LLC,

Petitioners,

v.

HEADWATER RESEARCH LLC,

Patent Owner.

Patent No. 10,749,700
Case No. IPR2024-01181

PETITION FOR *INTER PARTES* REVIEW

TABLE OF CONTENTS

I.	Introduction.....	1
II.	The '700 Patent.....	2
	A. Overview	2
	B. Prosecution History	3
III.	Level of Ordinary Skill.....	3
IV.	Claim Construction.....	4
V.	Statement of Precise Relief Requested.....	4
	A. Ground 1: Claims 1-2, 4-5, 7-11, 13, and 15-17 Are Obvious over <i>Rao</i> in View of <i>Scahill</i>	6
	1. Overview of <i>Rao</i>	6
	2. Overview of <i>Scahill</i>	10
	3. Rationale to Combine	13
	4. Claim 1	19
	a. 1[pre]: “A wireless end-user device comprising:”	19
	b. 1[a]: “at least one wireless modem to provide network access for the wireless end-user device;”	20
	c. 1[b]: “an activity classifier to, for each of a plurality of executable applications stored on the wireless end-user device, dynamically associate one or more corresponding network access policy controls with each of the plurality of executable applications;”	20
	d. 1[c]: “an Application Programming Interface (API) accessible to each of the plurality of executable applications, the API callable by each of the executable	

applications to request an execution time according to the corresponding network access policy controls for that application; and” 24

e. 1[d]: “a launch manager to schedule corresponding execution times for applications requesting an execution time through the API, the launch manager selecting corresponding execution times, for the applications requesting execution time, according to the respective network access policy controls for each such application.” 27

5. Claim 2: “The wireless end-user device of claim 1, the activity classifier dynamically associating network access policy controls with each of the plurality of executable applications using different policies that apply to blocked activities, network access controlled activities, and network access non-controlled activities.” 29

6. Claim 4: “The wireless end-user device of claim 2, the activity classifier determining that a given one of the applications is a network access non-controlled activity based on a user’s current interaction with that application.” 30

7. Claim 5: “The wireless end-user device of claim 2, the activity classifier determining that a given one of the applications is a network access non-controlled activity based on that application currently providing a foreground service.” 31

8. Claim 7: “The wireless end-user device of claim 1, wherein the API receives, from a given one of the applications, a network capacity demand as part of a request for an execution time.” 32

9. Claim 8: “The wireless end-user device of claim 1, wherein the API receives, from a given one of the applications, a network request as part of a request for an execution time.” 34

10. Claim 9: “The wireless end-user device of claim 8, wherein the network request specifies whether the network request

specifies one of a roaming network or a non-roaming network as the connected network for the network request.”...35

11.	Claim 10: “The wireless end-user device of claim 1, wherein the API receives, from a given one of the applications, a scheduled time request as part of a request for an execution time.”.....	37
12.	Claim 11: “The wireless end-user device of claim 1, further comprising a user interface allowing a user to specify one or more of the network access policy controls to be associated with a given one of the plurality of executable applications.”.....	38
13.	Claim 13: “The wireless end-user device of claim 1, wherein to dynamically associate one or more corresponding network access policy controls comprises changing an association of network access policy controls based on a device usage state.”	39
14.	Claim 15: “The wireless end-user device of claim 1, wherein the launch manager receives a list identifying executable applications that have a launch control policy in effect.”	40
15.	Claim 16: “The wireless end-user device of claim 1, wherein the launch manager is integrated with an operating system broadcast intent manager.”	41
16.	Claim 17: “The wireless end-user device of claim 1, wherein, based on a given one of the network access policy controls, the launch manager deprioritizes one or more processes associated with a given one of the executable applications.”.....	43
B.	Ground 2: Claim 3 Is Obvious over <i>Rao</i> in View of <i>Scahill</i> and <i>Oestvall</i>	44
1.	Overview of <i>Oestvall</i>	44

2.	Rationale to Combine	44
3.	Claim 3: “The wireless end-user device of claim 2, the activity classifier determining that a given one of the applications is a network access controlled activity or a blocked activity based on a lack of user interaction with that application for a period of time.”	46
C.	Ground 3: Claims 6, 14, and 18-19 Are Obvious over <i>Rao</i> in View of <i>Scahill</i> and <i>Montemurro</i>	47
1.	Overview of <i>Montemurro</i>	47
2.	Rationale to Combine	49
3.	Claim 6: “The wireless end-user device of claim 1, wherein according to at least one of the network access policy controls, the launch manager delays an execution time for one or more of the applications until a connection is made through the at least one wireless modem to an alternative network.”	52
4.	Claim 14: “The wireless end-user device of claim 1, wherein to dynamically associate one or more corresponding network access policy controls comprises changing an association of network access policy controls based on a change in a current network to which the at least one wireless modem is providing network access.”	53
5.	Claim 18: “The wireless end-user device of claim 1, wherein, based on a given one of the network access policy controls, the launch manager controls the frequency at which a given one of the executable applications is allowed network access.”	54
6.	Claim 19: “The wireless end-user device of claim 1, wherein, based on an amount of network data communicated to or from a given one of the executable applications, the activity classifier dynamically changes at	

	least one network access policy control associated with that given executable application.”	56
D.	Ground 4: Claims 12 and 20 Are Obvious over <i>Rao</i> in View of <i>Scahill</i> and <i>Araujo</i>	58
1.	Overview of <i>Araujo</i>	58
2.	Rationale to Combine	59
3.	Claim 12: “The wireless end-user device of claim 1, wherein to dynamically associate one or more corresponding network access policy controls comprises changing an association of network access policy controls based on a power state of the device.”	63
4.	Claim 20: “The wireless end-user device of claim 1, wherein the launch manager schedules corresponding execution times for applications to correspond with power state changes of the at least one wireless modem.”	65
VI.	The Petition Is Proper Under 35 U.S.C. § 325(d)	67
VII.	The Petition Is Proper Under <i>Fintiv</i>	69
VIII.	Mandatory Notices.....	70
A.	Real Party-in-Interest	70
B.	Related Matters.....	70
C.	Lead and Back-Up Counsel.....	70
D.	Service Information.....	71
IX.	Grounds for Standing.....	71
X.	Conclusion	72

TABLE OF AUTHORITIES

	Page(s)
Cases	
<i>Advanced Bionics, LLC v. MED-EL Elektromedizinische Geräte GmbH,</i> IPR2019-01469, Paper 6 (PTAB Feb. 13, 2020).....	67, 68
<i>Apple Inc. v. Fintiv Inc.,</i> IPR2020-00019, Paper 11 (PTAB Mar. 20, 2020)	69
<i>Ariosa Diagnostics v. Verinata Health, Inc.,</i> 805 F.3d 1359 (Fed. Cir. 2015)	63
<i>Becton, Dickinson, & Co. v. B. Braun Melsungen AG,</i> IPR2017-01586, Paper 8 (PTAB Dec. 15, 2017)	67
<i>Brown v. 3M,</i> 265 F.3d 1349 (Fed. Cir. 2001)	46
<i>Keynetik, Inc. v. Samsung Elecs. Co.,</i> No. 2022-1127, 2023 WL 2003932 (Fed. Cir. Feb. 15, 2023).....	<i>passim</i>
<i>KSR Int’l Co. v. Teleflex Inc.,</i> 550 U.S. 398 (2007).....	17
<i>In re Mouttet,</i> 686 F.3d 1322 (Fed. Cir. 2012)	19
<i>Navistar, Inc. v. Fatigue Fracture Tech., LLC,</i> IPR2018-00853, Paper 13 (PTAB Sept. 12, 2018).....	68
<i>Phillips v. AWH Corp.,</i> 415 F.3d 1303 (Fed. Cir. 2005) (en banc)	4
Statutes	
35 U.S.C. § 102(a)	5
35 U.S.C. § 102(b)	5

35 U.S.C. § 102(e)	5, 6
35 U.S.C. § 103	5
35 U.S.C. § 112, ¶2	4
35 U.S.C. § 311	4
35 U.S.C. § 314(a)	69
35 U.S.C. § 325(d)	67

Regulations

37 C.F.R. § 42.100(b)	4
-----------------------------	---

Other Authorities

USPTO Memorandum, Interim Procedure for Discretionary Denials in AIA Post-Grant Proceedings with Parallel District Court Litigation (June 21, 2022).....	69
--	----

LIST OF EXHIBITS

<u>Exhibit</u>	<u>Description</u>
Ex-1001	U.S. Patent No. 10,749,700 to Raleigh et al. (“the ’700 patent”).
Ex-1002	Patent Center Prosecution History of U.S. Patent No. 10,749,700.
Ex-1003	Declaration of Harry Bims, Ph.D.
Ex-1004	Curriculum Vitae of Harry Bims, Ph.D.
Ex-1005	U.S. Patent Appl. Publ. No. 2006/0039354 to Rao et al. (“ <i>Rao</i> ”).
Ex-1006	U.S. Patent Appl. Publ. No. 2009/0217065 to Araujo (“ <i>Araujo</i> ”).
Ex-1007	U.S. Patent Appl. Publ. No. 2009/0207817 to Montemurro et al. (“ <i>Montemurro</i> ”).
Ex-1008	International Appl. Publ. No. WO 2008/113986 to Scahill (“ <i>Scahill</i> ”).
Ex-1009	U.S. Patent Appl. Publ. No. 2007/0038763 to Oestvall (“ <i>Oestvall</i> ”).
Ex-1010	U.S. Patent No. 5,987,611 to Freund (“ <i>Freund</i> ”).
Ex-1011	U.S. Patent No. 8,028,060 to Wyld et al. (“ <i>Wyld</i> ”).
Ex-1012	U.S. Patent Appl. Publ. No. 2009/0093247 to Srinivasan (“ <i>Srinivasan</i> ”).
Ex-1013	U.S. Patent Appl. Publ. No. 2008/0080458 to Cole (“ <i>Cole</i> ”).
Ex-1014	U.S. Patent Appl. Publ. No. 2008/0057894 to Aleksic et al. (“ <i>Aleksic</i> ”).
Ex-1015	U.S. Patent Appl. Publ. No. 2009/0119773 to D’Amore et al. (“ <i>D’Amore</i> ”).
Ex-1016	U.S. Patent No. 8,909,211 to Huq et al. (“ <i>Huq</i> ”).
Ex-1017	U.S. Patent No. 8,381,127 to Singh et al. (“ <i>Singh</i> ”).

<u>Exhibit</u>	<u>Description</u>
Ex-1018	U.S. Patent No. 6,898,654 to Senior et al. (“ <i>Senior</i> ”).
Ex-1019	U.S. Patent Appl. Publ. No. 2009/0318124 to Haughn (“ <i>Haughn</i> ”).
Ex-1020	U.S. Patent No. 8,886,261 to Aerrabotu (“ <i>Aerrabotu</i> ”).
Ex-1021	European Patent Appl. No. 1,484,871 to Kelz et al. (“ <i>Kelz</i> ”).
Ex-1022	Scheduling Order, <i>Headwater Research LLC v. Motorola Mobility LLC, et al.</i> , No. 4:23-cv-04496-JST (N.D. Cal. Jan. 30, 2024) (ECF No. 46).
Ex-1023	Joint Claim Construction and Prehearing Statement, <i>Headwater Research LLC v. Motorola Mobility LLC, et al.</i> , No. 4:23-cv-04496-JST (N.D. Cal. June 11, 2024) (ECF No. 64).

I. Introduction

Lenovo (United States) Inc. and Motorola Mobility LLC (“Petitioners”) request *inter partes* review of claims 1-20 of U.S. Patent No. 10,749,700 (Ex-1001). The ’700 patent relates to a wireless end-user device that manages application requests on an application-by-application basis. As the patent explains, a device intercepts application requests to launch or access a network. (’700 patent, 66:38-51.) The device then handles requests differently based on priority (e.g., foreground or background applications), network requirements, power requirements, or other classifications, as set forth in a “control policy.” (*Id.*, 11:55-58, 12:48-57.) The device associates policy controls with particular applications. (*Id.*, claim 1.) Then, when an application requests to execute an activity, the device intercepts the request and schedules an execution time for that activity, according to the policy controls. (*Id.*, 66:38-51.)

There is nothing novel about any of the claimed elements or their combination. Instead, all claimed elements were known individually and in various combinations before the ’700 patent’s earliest priority date, including as taught by *Rao* (Ex-1005), *Scahill* (Ex-1008), *Oestvall* (Ex-1009), *Montemurro* (Ex-1007), and *Araujo* (Ex-1006). Considering these references, Petitioners request that the Board institute review and cancel the challenged claims.

II. The '700 Patent

A. Overview

According to the '700 patent, wireless network resources are stretched thin. “The increasing popularity of various smart phone devices, net book devices, [and] tablet computing devices” is leading to a “network capacity crunch.” ('700 patent, 3:33-41.) Network resources, e.g., base stations, are inundated with requests for network access, leading to “overloading” of base stations. (*Id.*, 6:23-7:5.) When this happens, base stations become “overwhelmed” and may begin “fall[ing] behind” in servicing the requests. (*Id.*, 6:61-7:5.)

According to the '700 patent, a device's network access requests may vary in several ways. First, they may vary in their priority. Typically, each request is associated with a particular application. (*Id.*, 4:8-24.) Because the services provided by these applications vary, the priority of requests may vary. For example, an “interactive” service, such as an application that is interacting with a user, may be higher priority because it requires “reasonably low response time” for a request, while a “background” service may be “lowest priority.” (*Id.*, 16:50-60.) For some services, it may make sense to hold off sending requests until a “scheduled transmission time.” (*Id.*, 88:36-44.)

To avoid “overloading” the base stations, the patent explains, the device intercepts requests by applications to launch or access the network. (*Id.*, 6:23-7:5,

11:24-45, 66:38-54.) The device then handles requests differently based on priority, quality of service, or other classifications, as set forth in a “control policy.” (*Id.*, 11:55-58, 12:48-57.) For example, the device associates policy controls with particular applications. (*Id.*, claim 1.) Then, when an application issues a request for a time to execute an activity, the device intercepts the request and schedules the time for execution of that activity, according to the policy controls. (*Id.*)

B. Prosecution History

The '700 patent was allowed after a double-patenting rejection was allegedly resolved via a terminal disclaimer. (*See generally* Ex-1002 at 422-426, 612-627, 650.) During prosecution, the Examiner did not apply any of this Petition’s references. (*See id.*)

III. Level of Ordinary Skill

A person of ordinary skill in the art (“POSITA”) at the effective filing date of the '700 patent would have (1) held at least a bachelor’s degree in computer science, electrical engineering, or a related field, and (2) had three to five years of experience with networking, power consumption of networked computing devices, and/or wireless digital communications systems. (Ex-1003 (“Bims”), ¶¶24-26.) More related education would compensate for fewer years of work experience (and vice versa). (*Id.*)

IV. Claim Construction

Claims in an IPR are construed under the claim construction principles set forth in *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc). 37 C.F.R. § 42.100(b). Petitioners believe no terms need to be expressly construed, so at this time Petitioners identify no terms for construction.

In the parallel district court litigation, the parties filed a Joint Claim Construction and Prehearing Statement (Ex-1023). The analysis in this Petition satisfies both parties constructions for all disputed terms. (*Id.*, 2-4.) Petitioners additionally argued the following term is indefinite under pre-AIA 35 U.S.C. § 112, ¶2: “an operating system broadcast intent manager” (claim 16). (*Id.*, 4.) But the Board need not reach that issue because the patent admits the operating system broadcast intent manager was known in the art and existed in the prior art applied in this Petition, as discussed below.

V. Statement of Precise Relief Requested

Petitioners respectfully request review under 35 U.S.C. § 311 of claims 1-20 of the '700 patent and cancellation of those claims as obvious under pre-AIA 35 U.S.C. § 103 based on the following unpatentability grounds.

Ground	References	Challenged Claims
1	<i>Rao</i> in view of <i>Scahill</i>	Claims 1-2, 4-5, 7-11, 13, and 15-17

Ground	References	Challenged Claims
2	<i>Rao</i> in view of <i>Scahill</i> and <i>Oestvall</i>	Claim 3
3	<i>Rao</i> in view of <i>Scahill</i> and <i>Montemurro</i>	Claims 6, 14, and 18-19
4	<i>Rao</i> in view of <i>Scahill</i> and <i>Araujo</i>	Claims 12 and 20

The '700 patent claims priority to multiple provisional applications, the earliest of which was filed January 28, 2009. ('700 patent, Cover.) While Petitioners do not concede that January 28, 2009, is the priority date that should be afforded to the '700 patent, all references asserted in this Petition qualify as prior art under pre-AIA 35 U.S.C. § 102(a), (b), and/or (e) even if the January 28, 2009, date is used as the priority date of the '700 patent:

Ex-1005	U.S. Patent Publication 2006/0039354 (" <i>Rao</i> "), published February 23, 2006; prior art under § 102(a), (b), and (e).
Ex-1008	International Application Publication WO 2008/113986 (" <i>Scahill</i> "), published September 25, 2008; prior art under § 102(a).
Ex-1009	U.S. Patent Publication 2007/0038763 (" <i>Oestvall</i> "), published February 15, 2007; prior art under § 102(a), (b), and (e).

Ex-1007	U.S. Patent Publication 2009/0207817 (“ <i>Montemurro</i> ”), filed February 15, 2008; prior art under § 102(e).
Ex-1006	U.S. Patent Publication 2009/0217065 (“ <i>Araujo</i> ”), filed February 26, 2008; prior art under § 102(e).

A. Ground 1: Claims 1-2, 4-5, 7-11, 13, and 15-17 Are Obvious over *Rao* in View of *Scahill*

1. Overview of *Rao*

Like the '700 patent, *Rao* “generally relates to optimizing peer-to-peer network communications, and in particular, to techniques for application-aware prioritizations of network communications on a client.” (*Rao*, [0002].) *Rao* is analogous art to the '700 patent. (Bims, ¶¶50-55.)

As *Rao* explains, in prior-art computing devices (also, “client[s]”), “[c]ommunications over a network ... are typically processed in the order generated by the activity of a user and applications of the client.” (*Rao*, [0003].) But this is not “always desirable.” (*Id.*) For example, “a network packet generated or received for an application running in the background may be processed ahead of a network packet generated or received for the application running in the foreground,” such as a voice-over-IP (VoIP) telephone call. (*Id.*) This results in increased latency and reduced quality for the foreground application. (*Id.*)

Accordingly, *Rao*'s prioritization scheme "provide[s] application-aware, client-specific prioritization of packet traffic." (*Id.*) To facilitate the prioritization, *Rao*'s computing device includes remote access client 120 that interacts with applications 338 on the computing device and the device's network stack 310, as illustrated in Figure 5A. (*Rao*, [0098]; *see also Rao*, [0101], [0180], [0184].) Remote access client 120 "manages and controls network communication prioritizations on the client." (*Rao*, [0080].)

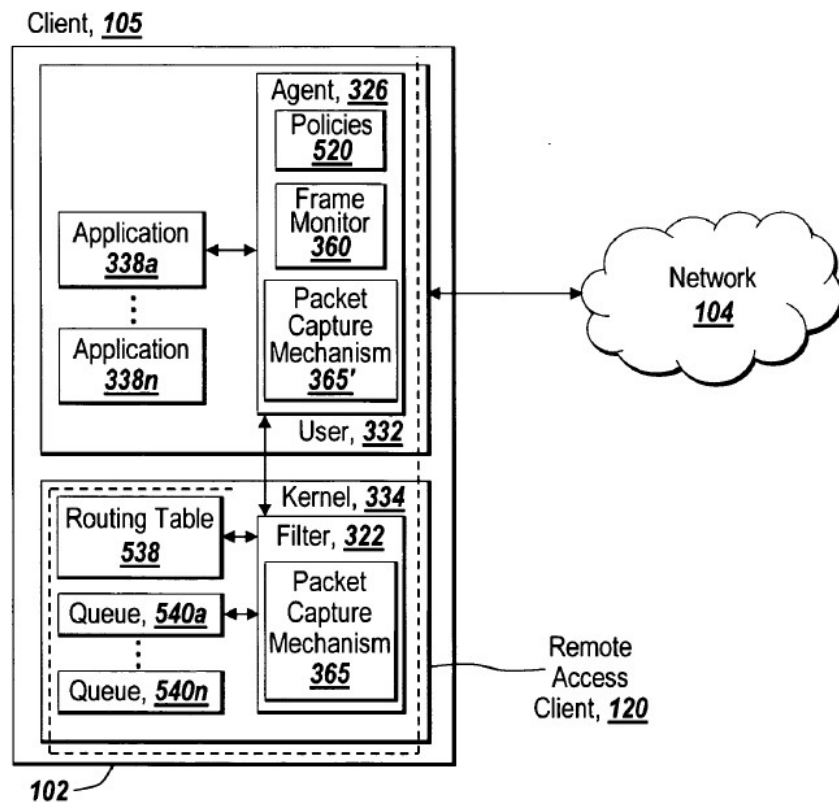


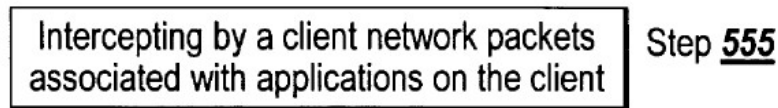
Fig. 5A

(*Rao*, Fig. 5A.)

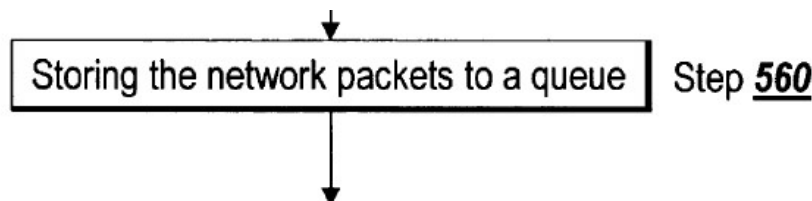
Rao's remote access client 120 includes an agent 326 that stores "[p]olicies 520." (*Id.*) These policies "specify[] client-side prioritization of network communications related to applications" at the computing device. (*Rao*, [0182].) For example, the "policies 520 define prioritization based on whether an application is running in the foreground or the background of the client 105." (*Id.*) *Rao* envisions a "multitude of ways to define client-side application priorities." (*Id.*)

Remote access client 120 further includes filter 322, which employs a packet capture mechanism 365 and communicates with queues 540a-540n. (*Rao*, [0099], [0101], [0180].) The packet capture mechanism 365 "may use any hooking application programming interface (API) to intercept, hook, or otherwise obtain ... network traffic associated with [an] application." (*Rao*, [0110].) The queues are for "queuing and prioritizing [the] network communications" obtained by the packet capture mechanism 365. (*Rao*, [0180].)

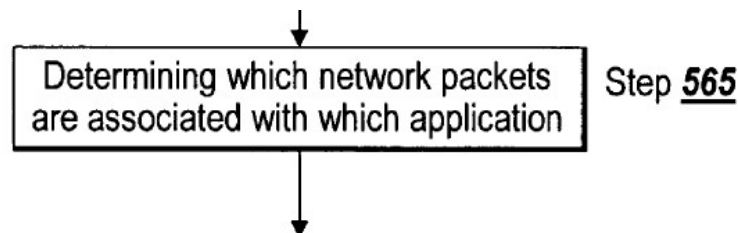
Figure 5B illustrates *Rao*'s prioritization technique. (*Rao*, [0184], Fig. 5B.) Beginning with step 555, "client 105 intercepts," e.g., using the packet capture mechanism, "one or more network[] packets associated with an application 338a-338n on the client 105."



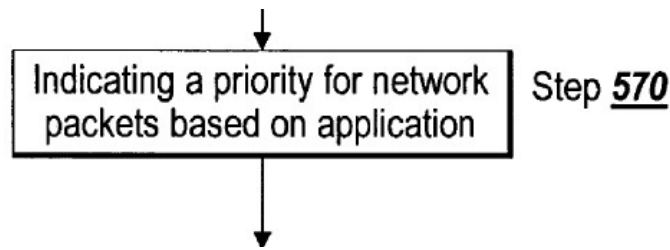
(*Rao*, [0184], [0186], Fig. 5B (cropped).) Then, “[a]t step 560, the network packets intercepted at step 555 may be stored in a queue 540a-540n,” such as a “temporary queue.”



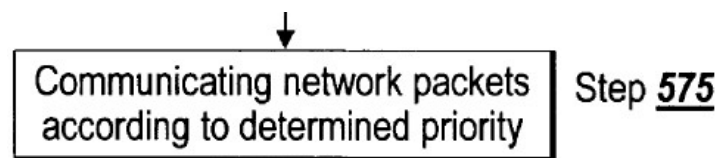
(*Rao*, [0186], Fig. 5B (cropped).) At step 565, remote access client 120 “determines the association of network packets with applications 338a-338n in order to determine priorities and apply any priority[-]based policies.”



(*Rao*, [0186], [0187], Fig. 5B (cropped).) And, at step 570, remote access client 120 “indicates a priority for intercepted ... network packets based on the application 338a-338n associated with the packet at step 565.” (*Rao*, [0189].) For example, the remote access client may “use[] the policies 520 to apply a priority to network packets of applications 338a-338n in accordance with the prioritization rules specified or indicated by the policies.”



(*Rao*, [0189], Fig. 5B (cropped).) Finally, at step 575, “the network packets are communicated from the queues 540a-540n according to the determined priorities,” “which[,] in turn, may be based or derived from the policies 520 of the client.” (*Rao*, [0194].) To this end, *Rao* explains, the prioritized packets are provided “for communications via a network stack of the client.” (*Rao*, [0038].)



(*Rao*, [0184], Fig. 5B (cropped).) Thus, “outbound network packets generated by an application 338a-338n on the client 105 are prioritized by the client 105 prior to transmission based on the type and/or priority of the application 338a-338n.” (*Rao*, [0184].)

2. Overview of *Scahill*

Scahill “seeks to mitigate and/or obviate the limitations known in the art with the timing of data transmissions from mobile communications devices.” (*Scahill*, 5:8-9.) In particular, like the ’700 patent, *Scahill* discloses a method of coordinating the execution time of a plurality of applications requiring a network

connection that are all hosted on the same mobile device. (*Scahill*, Abstract.)

Scahill is analogous art to the '700 patent. (Bims, ¶¶56-59.)

According to *Scahill*, “[t]he connection requests generated by ... [a] client application 12a may require network connections which vary in time and/or which vary depending on the amount of data and/or type of task to be performed.”

(*Scahill*, 19:9-11.) Thus, when an application requests execution of a task, *Scahill*'s method entails (1) determining task completion conditions, including necessary network or device conditions; (2) retrieving stored data indicating network characteristics for an available network connection during a predetermined time period; (3) processing the task completion conditions to determine if the network characteristics retrieved for the predetermined period of time are sufficient to complete the task; and (4) in the event the network characteristics are sufficient, scheduling the task for execution. (*Scahill*, Abstract.)

Scahill's Figure 1B illustrates a client application 12a for carrying out *Scahill*'s method. Client application 12a includes “a network aware scheduling software client library 14a.” (*Scahill*, 18:15-16.) This “client library 14a is arranged to allow a connection requesting component 11 of the application 12a to interface with an application scheduler 24 library component of the client library 14a.” (*Scahill*, 18:16-18.)

the client application 12a to perform the relevant task. (*Scahill*, 20:5-8.) The operating system launches the client application 12a when all necessary conditions are satisfied, such as at a specific time requested by the application. (*Scahill*, 20:9-10.) *Scahill* discloses that the request may specify “task completion conditions” for the task, including a “start delay” (also called a “timing interval Δt ”) and a required connection “duration.” (*Scahill*, 24:19-23, 26:17-20.)

3. Rationale to Combine

It would have been obvious to a POSITA to improve *Rao*’s remote access architecture by including *Scahill*’s scheduling method to coordinate network communications of a plurality of applications. (Bims, ¶¶60-72.)

Rao discloses “application-aware prioritization of client-side network communications.” (*Rao*, Abstract.) *Rao*’s remote access client uses policies to prioritize network communications from certain applications over others. (Bims, ¶61; *Rao*, [0038].) *Rao* discloses that the remote access client also “considers other network factors in determining what network packets to communicate from what queue.” (*Rao*, [0195].) For example, *Rao* discloses that “remote access client 120 may receive an indication of network congestion,” and as a result, “may throttle back or not communicate [certain] network packets.” (*Id.*) Thus, according to *Rao*, “client 105 controls and manages the prioritization of network communications of the client 105 on an application 338a-338n basis and in accordance with any

policies 520 for the client 105 and in further view of any network statistics and other [network] factors.” (*Id.*)

Scahill goes further, seeking “to mitigate and/or obviate the limitations known in the art with the timing of data transmissions from mobile communications devices.” (*Scahill*, 5:8-9.) *Scahill* recognized that, in prior art such as *Rao*, the scheduling of network communications “may not coincide with appropriate network conditions to support the type of data transmission required to complete the scheduled task.” (*Scahill*, 2:14-17.) The result is that “[e]ach attempt at transmission [with prior-art systems] needlessly wastes the resources of the device and if a data transmission is aborted as the network conditions alter in the course of the transmission, network resources have also been wasted.” (*Scahill*, 2:17-20.) Thus, “it is useful if the task execution can be timed to coincide with appropriate network conditions to prevent having any necessity to perform such a delay operation.” (*Scahill*, 2:24-25.) *Scahill* thus “seek[s] to preserve device resources so that they are used only when there is a likelihood of a connection having characteristics which will allow an application to successfully complete one or more scheduled tasks in a particular connection window.” (*Scahill*, 13:31-34.) To this end, *Scahill*’s application scheduler schedules application tasks at times when appropriate network conditions (or other device conditions) are likely present. (*Scahill*, 19:20-25.)

Scahill discloses that its method is beneficial for other reasons as well. (Bims, ¶67.) First, *Scahill*'s method "reduces or removes the likelihood of any conflict with other applications trying to also establish a connection over the same wireless communications link." (*Scahill*, 4:11-15.) According to *Scahill*, "[e]ven when a wireless communications device can support more than one network connection[,] it is useful if conflicts can be resolved to ensure applications ... are coordinated[,] as otherwise[,] conflicts could still occur." (*Scahill*, 4:15-18.)

Second, *Scahill*'s method reduces the financial costs associated with certain data transmissions. (Bims, ¶68.) For example, "under certain [network] conditions, e.g., if located in a roaming data network, the user may be on a tariff." (*Scahill*, 2:20-22.) *Scahill*'s method reduces the occurrences of these roaming costs by allowing each application to specify a particular network for execution and scheduling the application's task at an execution time when the specified network is available. (Bims, ¶68.)

Third, *Scahill*'s method optimizes a device's power use. (Bims, ¶69.) According to *Scahill*, prior-art transmission methods faced many situations that unnecessarily drain battery resources, including where (1) a device must "continually scan for available networks and trigger applications when a particular network is available"; and (2) "there may be long periods of time ... between the user being in range of a suitable network." (*Scahill*, 4:20-30.) *Scahill*'s method

optimizes power consumption in these scenarios by scheduling tasks in a particular connection window. (*Id.*; Bims, ¶69.)

In view of these benefits, a POSITA would have been motivated to modify *Rao*'s architecture to further include *Scahill*'s scheduling method. (Bims, ¶69.) In addition to improving *Rao*'s architecture, *Scahill*'s method aligns with *Rao*'s intent of optimizing the data transmission process. (*Id.*)

Scahill's application scheduling method is complementary to *Rao*'s application prioritization method. (Bims, ¶70.) For example, *Rao*'s remote access client may use an application's characteristics, such as whether that application is running in the foreground or background, to determine a network packet's priority. (*Rao*, [0189].) And the remote access client may use network characteristics, such as congestion, in determining which network packets to communicate from which queue. (*Rao*, [0195].) *Scahill* builds on *Rao*'s disclosure, disclosing how to schedule the transmission of network communications to reduce conflicts and utilize ideal network conditions. (Bims, ¶70.) Thus, rather than just prioritizing certain applications over others in queues, *Rao/Scahill*'s architecture would also schedule transmissions of the prioritized applications' network communications to optimize network conditions. (*Id.*) Such scheduling minimizes transmission conflicts, reduces financial costs, and optimizes a device's power consumption. (*Id.*)

A POSITA would have had a reasonable expectation of success in modifying *Rao*'s architecture in view of *Scahill*'s method. (Bims, ¶71.) *Rao* discloses flexibility with respect to how network communications are queued and communicated. (*Id.*) Regarding queuing, *Rao* discloses that “queues 540a-540n may comprise any type and/or form of suitable means and/or mechanisms for storing and/or arranging network packets,” and discloses that an ordinary artisan “will recognize and appreciate that the network packets may ... be placed or arranged in the queue in any suitable manner.” (*Rao*, [0180], [0191].) Regarding communicating, *Rao* discloses the prioritized packets are provided “for communications via a network stack of the client,” which “may comprise one or more network layers,” “one or more protocols,” and/or “one or more network drivers” “included as part of the operating system ... or as part of any network interface cards or other network access components of the computing device.” (*Rao*, [0038], [0100].) A POSITA would have appreciated that *Rao*'s architecture would be improved by queuing and communicating based on *Scahill*'s scheduling method (e.g., assigning an execution time to an application's task). (Bims, ¶71.)

The modification would have entailed nothing “more than the predictable use of prior art elements according to their established functions.” *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 417 (2007). Under *Rao*'s architecture, remote access client 120 prioritizes network communications on an application 338a-338n basis,

in accordance with policies 520, and communicates the network communications in view of any network statistics or other network factors. (*Rao*, [0195].) A POSITA would have understood how to implement *Scahill*'s scheduling method in *Rao*'s architecture. (Bims, ¶72; *see, e.g.*, '700 patent, 10:16-23.) For instance, *Scahill* discloses that its scheduling method is implemented through a scheduling software client library (and related components). (*Id.*; *Scahill*, 18:15-22.) In *Rao*'s modified architecture, device applications would likewise include or otherwise interface with this scheduling software client library (and related components) to schedule transmissions. (Bims, ¶72.) The scheduling library (also referred to as an application scheduler) would assign execution times for transmitting the applications' network communications in accordance with the policies 520, taking into consideration likely network conditions as *Scahill* describes. (*Id.*) In the *Rao/Scahill* combination, then, *Scahill*'s application scheduler and related components are integrated into *Rao*'s architecture and work together to prioritize network communications into queues and assign execution times according to *Rao*'s policies, taking into consideration *Scahill*'s scheduling requests and other network factors. (*Id.*)

A POSITA would have understood how to modify *Rao*'s architecture to accommodate *Scahill*'s scheduling functionality. (*Id.*) As the Federal Circuit has explained, “[n]ormally, once the function to be performed by software has been

identified, writing code to achieve that function is within the skill of the art.”

Keynetik, Inc. v. Samsung Elecs. Co., No. 2022-1127, 2023 WL 2003932, at *2 (Fed. Cir. Feb. 15, 2023) (citation omitted). Such is the case here where the functions (e.g., *Scahill*’s scheduling method) have been identified. (Bims, ¶72.)

Thus, a POSITA would have incorporated *Scahill*’s teachings into *Rao*’s device. In view of the advantages of *Scahill*’s scheduling method, and in particular how *Scahill*’s scheduling method furthered *Rao*’s aim of optimizing data transmissions from a client device running multiple applications, a skilled artisan would have had reason to modify *Rao*’s architecture in view of *Scahill*. *See In re Mouttet*, 686 F.3d 1322, 1332 (Fed. Cir. 2012) (bodily incorporation is not required to demonstrate obviousness).

4. Claim 1

a. 1[pre]: “A wireless end-user device comprising:”

Rao/Scahill renders obvious the preamble, to the extent it is limiting. (Bims, ¶73.) *Rao* discloses “multiple computing devices 102a-102c” (also referred to as “clients 105a-105c”). (*Rao*, [0086].) “[C]lient 105 can be any type and/or form of computing device 102 that can run one or more applications 338.” (*Rao*, [0088].) For instance, *Rao*’s device may also be a “mobile client,” such as a “notebook, personal digital assistant (PDA), a smart phone,” or a “telecommunication device,” that “can connect to one or more networks ... through a variety of connections

including ... wireless connections.” (*Rao*, [0095], [0198]; *see also Rao*, [0130].)

Rao’s device is an “end-user device” because *Rao* discloses a device “user.” (*Rao*, [0183], [0199]; Bims, ¶73.)

b. 1[a]: “at least one wireless modem to provide network access for the wireless end-user device;”

Rao/Scahill renders obvious element 1[a]. (Bims, ¶74.) *Rao*’s device includes network interface 118, which enables connectivity to a “Local Area Network (LAN), Wide Area Network (WAN) or the Internet through a variety of connections,” such as “standard telephone lines, LAN or WAN links,” “broadband connections,” “*wireless* connections,” or some combination of the above. (*Rao*, [0095], [0125].)¹ Network interface 118 includes “a built-in network adapter, network interface card, PCMCIA network card, card bus network adapter, ... USB network adapter, [or] *modem*.” (*Rao*, [0125].)

c. 1[b]: “an activity classifier to, for each of a plurality of executable applications stored on the wireless end-user device, dynamically associate one or more corresponding network access policy controls with each of the plurality of executable applications;”

Rao/Scahill renders obvious element 1[b]. (Bims, ¶¶75-80.) *Rao* discloses the claimed “plurality of executable applications stored on the wireless end-user

¹ All emphases added unless otherwise noted.

device” because each client 105 includes “one or more applications 338a-338n.”

(*Rao*, [0087]; Bims, ¶75.) *Rao* explains that “applications 338a-338n provide one or more real-time data communications, such as VoIP,” or “may provide email, collaboration, online meeting, and/or desktop sharing related services or functionality.” (*Rao*, [0179]; *see also Rao*, [0088] (describing other applications).)

Rao also discloses that “[e]ach client 105a-105n includes a remote access client 120a-120c.” (*Rao*, [0087].) Figure 5A (below) shows a system having the remote access client for routing network packets from a client 105 to a network 104 (via network interface 118):

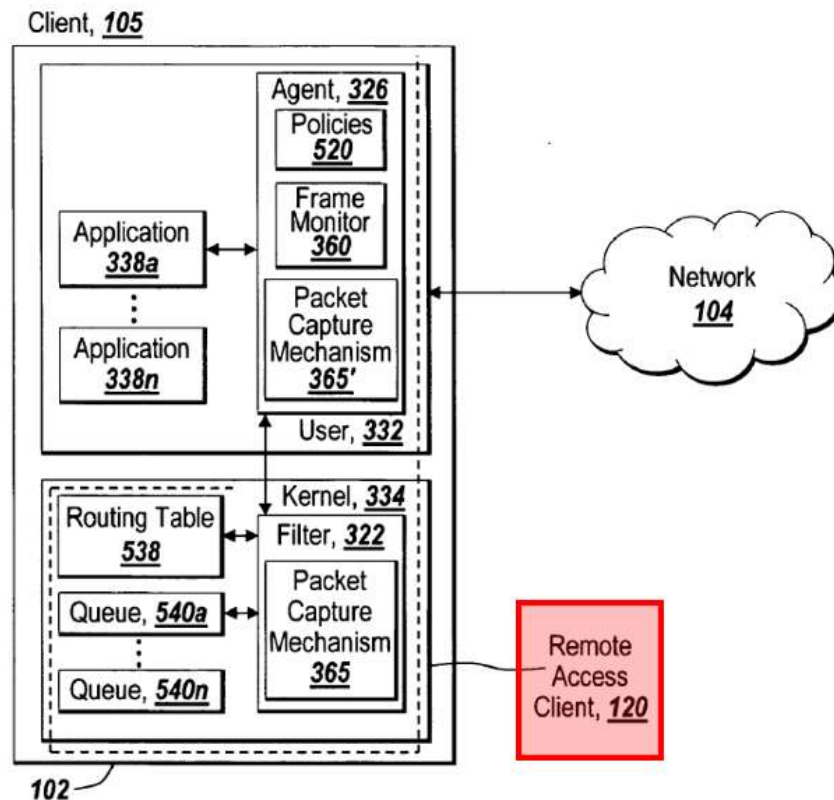


Fig. 5A

(*Rao*, Fig. 5A (annotated).) “[T]he remote access client 120 may have one or more policies 520 for specifying client-side prioritization of network communications related to applications 338a-338n.” (*Rao*, [0182].) These policies are the claimed “network access policy controls.” (Bims, ¶76.) According to *Rao*, “[t]hese policies 520 may be specified by any suitable means and/or mechanisms,” including “the name of the application 338a-338n and/or the type of application 338a-338n,” “the type of one or more protocols used by the applications 338a-338n,” and/or “whether an application is running in the

foreground or the background.” (*Rao*, [0182].) *Rao* explains that a POSITA “will recognize and appreciate the multitude of ways to define client-side application priorities.” (*Id.*)

Rao’s remote access client 120, via “agent 326,” operates as the claimed “activity classifier” to “dynamically associate one or more corresponding network access policy controls with each of the plurality of executable applications.” (Bims, ¶77.) Under *Rao*’s step 565 (Figure 5B), remote access client 120 “determines the association of network packets with applications 338a-338n in order to determine priorities and apply any priority[-]based policies 520.” (*Rao*, [0187].) *Rao* explains that “client 105, such as via agent 326, can associate network traffic with applications 338a-338n by any suitable means and/or mechanisms.” (*Id.* (further listing how agent 326 identifies a network packet).) A POSITA would have understood *Rao*’s network packets correspond to “activit[ies]” of the applications. (Bims, ¶78.) For instance, *Rao* describes the “network packets of [a] VoIP telephone call” made by a “VoIP application.” (*Rao*, [0003]; *see also Scahill*, 17:6-10 (disclosing “a one-to-one mapping between a task and a client application”).) The VoIP application’s network packets thus correspond to an activity (i.e., a VoIP telephone call) of the VoIP application. (Bims, ¶78.)

Remote access client 120 associates policies “dynamically” because, for example, the policy currently associated with the application may depend on an

application's present characteristics, including whether the application "is running in the foreground" (and in active use by the user) or "is running in ... the background." (Bims, ¶79; *Rao*, [0188].) Similarly, "remote access client 120 may determine any other characteristics or statistics of the application 338a-338n such as size, memory usage, total execution time, and/or frequency of use." (*Rao*, [0188].) These characteristics reflect an application's current state and may change over time, requiring dynamic association by the remote access client. (Bims, ¶79.)

d. 1[c]: "an Application Programming Interface (API) accessible to each of the plurality of executable applications, the API callable by each of the executable applications to request an execution time according to the corresponding network access policy controls for that application; and"

Rao/Scahill renders obvious element 1[c]. (Bims, ¶¶81-85.) In *Rao/Scahill*'s architecture, "client library 14a is arranged to allow a connection requesting component 11 of the application 12a to interface with an application scheduler 24 library component of the client library 14a." (*Scahill*, 18:16-18.) According to *Scahill*, "[t]he connection requesting component 11 of the client application 12a is effectively any part of the client application 12a which generates a connection request to perform a task." (*Scahill*, 18:30-31; *see also Scahill*, 19:18-20.) This connection requesting component of the client application thus requests an execution time for the application. (Bims, ¶81.)

Scahill discloses that “[w]hen a client application 12a wants to know its next launch time, the client library 14a of the client application 12a calls up, via an application programming interface [“API”], the client library application scheduling component 24.” (*Scahill*, 43:3-5.)

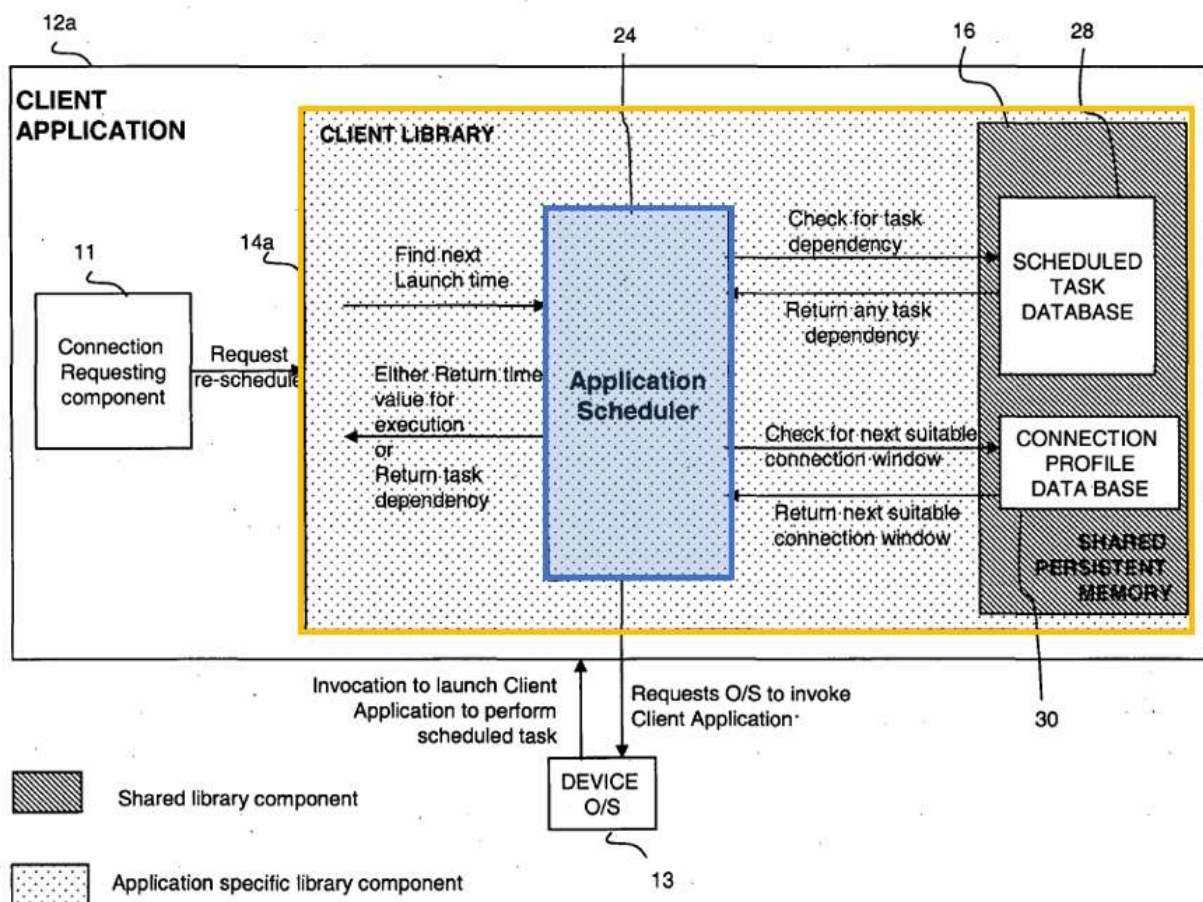


FIGURE 1B

(*Scahill*, Fig. 1B (annotated).) A POSITA would have understood that this API is the “Application Programming Interface (API) accessible to each of the plurality of executable applications,” as claimed. (*Bims*, ¶¶82-83.) This API is accessed by

each application to request an execution time because the API provides an interface to the client library application scheduling component 24 (application scheduler). (*Id.*) *Scahill* teaches that client library 14a is “accessible by all client applications,” and the scheduling component API (within client library 14a) is thus callable by each of the executable applications to request an execution time. (*Scahill*, 16:29-32; Bims, ¶82.)

A POSITA would have understood that a client application’s “request to perform a task” in *Scahill* is a request for an execution time because it entails a request for a scheduled time slot. (Bims, ¶84; *Scahill*, 18:30-31.) In *Scahill*, the application’s request is made to a “scheduler” that schedules a “connection window” for the task, where “connection window ... refers to a period of time of predetermined duration” that can be scheduled at “an absolute time value.” (Bims, ¶84; *Scahill*, 19:25-26, 26:17-23.) Moreover, in *Scahill*, “[t]here are a number of application specific requirement(s),” also called “connection conditions,” “which the application scheduler library component 14a,b of each client application 12a,b ... must take into account when seeking to schedule performance of a task.” (*Scahill*, 19:3-7, 24:14-17.) This can include a “timing interval Δt requested by the client application since the task was last performed.” (*Scahill*, 24:19-23.) For example, an application’s request could specify a “delay of 172800[]secs and duration of 62 secs.” (*Scahill*, 42:25-31.) In each case, the API

receives a request for a scheduled time slot, the “connection window,” as part of a request for an execution time from an application. (Bims, ¶84.)

As discussed in Section V.A.3 (Rationale to Combine), in *Rao/Scahill*’s architecture, *Rao*’s remote access client 120 manages the prioritization of network communications in accordance with at least *Rao*’s policies and *Scahill*’s scheduling requests. (Bims, ¶85.) *Rao*’s architecture, as modified in view of *Scahill*, thus includes an API that is both accessible to each of the plurality of executable applications and callable by each of the executable applications to request an execution time according to the corresponding network access policy controls for that application. (*Id.*)

e. 1[d]: “a launch manager to schedule corresponding execution times for applications requesting an execution time through the API, the launch manager selecting corresponding execution times, for the applications requesting execution time, according to the respective network access policy controls for each such application.”

Rao/Scahill renders obvious element 1[d]. (Bims, ¶¶86-88.) *Rao* discloses that “filter 322” (“launch manager”) arranges network communications from the applications into a queue according to their respective policies. (Bims, ¶86.)

According to *Rao*, “[i]n some embodiments, the agent 326 indicates the priority to the filter 322 for management of network packet queues 540a-540n to apply the indicated priorities.” (*Rao*, [0190].) Based on the priorities indicated for

applications 338a-338n, “filter 322 may place, arrange, or coordinate network packets into queues 540a-540n in support of the priorities.” (*Rao*, [0191].) *Rao* discloses that “filter 322 may move network packets from temporary queues 540a-540n or from memory or other storage to a queue 540a-540n associated with the application 338a-338n, a queue 540a-540n associated with a priority, or a queue 540a-540n associated with both an application and a priority.” (*Id.*) *Rao* provides additional examples of the many potential ways to queue applications. (*Id.*) According to *Rao*, a POSITA “will recognize and appreciate that the network packets may ... be placed or arranged in the queue in any suitable manner in practicing the operations of the present invention described herein.” (*Id.*)

In *Rao/Scahill*’s architecture, filter 322 coordinates network communications into queues using *Rao*’s policies, and the prioritized network communications would be assigned execution times by filter 322, implemented to incorporate the scheduling functionality of *Scahill*’s application scheduler 24, taking into account *Scahill*’s scheduling requests and other network factors. See Section V.A.4.d (element 1[c]). Once queued, remote access client 120 communicates the network packets. (*Rao*, [0194], [0195].) Filter 322 thus selects corresponding execution times for the applications requesting execution time in accordance with respective network access policy controls for each application. (Bims, ¶88.)

5. Claim 2: “The wireless end-user device of claim 1, the activity classifier dynamically associating network access policy controls with each of the plurality of executable applications using different policies that apply to blocked activities, network access controlled activities, and network access non-controlled activities.”

Rao/Scahill renders obvious claim 2. (Bims, ¶¶89-92.) *Rao* discloses that the remote access client indicates a priority based on whether an application is running in the background (“network access controlled activities”) or foreground (“network access non-controlled activities”).² (Bims, ¶89.) *Rao*’s “policies 520 define prioritization based on whether an application is running in the foreground or the background.” (*Rao*, [0182].) With these policies defined, at step 570, the remote access client’s “agent 326 [(the activity classifier)] uses the policies 520 to apply a priority to network packets of applications 338a-338n in accordance with the prioritization rules specified or indicated by the policies 520.” (*Rao*, [0189].) Agent 326 may also “use characteristics of the application 338a-338n, such as running in the foreground or background, to indicate priority for a network packet of the application 338a-338n.” (*Id.*)

² This is consistent with the ’700 patent’s specification. (Bims, ¶89; ’700 patent, 64:13-21, 108:1-4.)

Rao discloses that agent 326 also associates policies for “blocked activities.” (Bims, ¶91.) *Rao* discloses that “one or more queues 540a-540n may be used for network packets intercepted but not prioritized because ... policies 520 indicate to ignore or not process the network packet for prioritization.” (*Rao*, [0193].) By using agent 326 to associate a policy that requires ignoring or declining to process a network packet, *Rao* discloses an activity classifier that associates a policy based on blocked activities. (Bims, ¶91.)

As discussed in Section V.A.4.c (element 1[b]), *Rao/Scahill*’s policies are associated “dynamically” because the policy may depend on an application’s present characteristics, such as whether the application is running in the foreground or background. (Bims, ¶92; *Rao*, [0188].) These characteristics (as well as others) change over time, requiring dynamic association by the remote access client. (Bims, ¶92.)

6. Claim 4: “The wireless end-user device of claim 2, the activity classifier determining that a given one of the applications is a network access non-controlled activity based on a user’s current interaction with that application.”

Rao/Scahill renders obvious claim 4. (Bims, ¶¶93-95.) As *Rao* explains, “remote access client 120 [via agent 326] may determine whether the application 338a-338n associated with the network packet is running in the foreground.” (*Rao*, [0188]; *see also Rao*, [0189] (disclosing how agent 326 may

use an application's characteristics, "such as running in the foreground or background").) As discussed in Section V.A.5 (claim 2), an application running in the foreground is a "network access non-controlled activity." (Bims, ¶93.)

A POSITA would have understood that *Rao/Scahill* classifies an application as "running in the foreground" because of a user's current interaction with that application. (Bims, ¶¶94-95.) According to *Rao*, an "application is running in the foreground" where that application is "in active use by the user." (*Rao*, [0003].) Similarly, *Scahill* describes "foreground applications" as "applications that the user is interacting with." (*Scahill*, 36:4-8.) These disclosures are each consistent with other prior-art disclosures describing the classification of foreground applications. (Bims, ¶94 (citing *Freund*, 3:13-16, 10:17-42; *Oestvall*, [0006])).)

7. Claim 5: "The wireless end-user device of claim 2, the activity classifier determining that a given one of the applications is a network access non-controlled activity based on that application currently providing a foreground service."

Rao/Scahill renders obvious claim 5. (Bims, ¶96.) As discussed in Section V.A.6 (claim 4), *Rao/Scahill* discloses an activity classifier determining that a given application is a network access non-controlled activity by virtue of that application running in the foreground (i.e., providing a foreground service). (*Id.*; *Rao*, [0003]; *Scahill*, 36:4-8.)

8. Claim 7: “The wireless end-user device of claim 1, wherein the API receives, from a given one of the applications, a network capacity demand as part of a request for an execution time.”

Rao/Scahill renders obvious claim 7. (Bims, ¶¶97-100.) As discussed in Section V.A.3 (Rationale to Combine), a POSITA would have been motivated to improve *Rao*’s architecture to include *Scahill*’s scheduling method. This method includes an application’s request to schedule an execution time via an API. (Bims, ¶97.)

As explained in Section V.A.4.d (element 1[c]), *Scahill* discloses that an application’s request specifies “application specific requirement(s)” (also called “connection conditions”) for the application, “which the application scheduler library component 14a,b of each client application 12a,b ... must take into account when seeking to schedule performance of a task.” (*Scahill*, 19:3-7, 24:14-17.) One such connection condition in *Scahill* is a “connection bandwidth exceeding a minimum value.” (*Scahill*, 19:3-7; *see also Scahill*, 19:9-11.)

For example, when requesting an execution time via an API, “client application 12a first provides a tuple of data,” including “the required network profile.” (*Scahill*, 29:13-16; *see also* Section V.A.4.d (element 1[c])). “[T]he network profile” includes “the following fields: Network type, Min Uplink BW, Min Downlink BW, Max RTT, Max PacketLoss, and/or Max Jitter.” (*Scahill*, 42:21-23.) “[E]ach field may be optionally specified by the client application.”

(*Scahill*, 42:25-27.) For example, “an application requiring to transfer 1 Mb over a min 128[kb/s downlink over a WLAN network could specify a network profile as follows: ‘WLAN, -1,128000,-1,-1,-1,-1’ with delay of 172800[secs and duration of 62 secs.”³ (*Scahill*, 42:28-31.)

Thus, *Scahill* discloses that the scheduler receives, from an application, a connection bandwidth (“network capacity demand”) as part of its request for an execution time via an API. (Bims, ¶¶98-100; *see also* ’700 patent, 5:61-6:12, 8:49-62, 74:54.) From this, a POSITA would have been motivated and found it obvious to implement *Rao/Scahill*’s architecture such that *Rao/Scahill*’s application scheduler receives, from an application, a connection bandwidth as part of its request for an execution time via an API. (Bims, ¶100.) As *Scahill* explains, a network connection attempt that does “not coincide with appropriate network conditions to support the type of data transmission required to complete the scheduled task” “needlessly wastes the resources of the device and ... network resources have also been wasted.” (*Scahill*, 2:14-20.) By receiving the connection bandwidth in the application’s request, *Rao/Scahill*’s application scheduler

³ “[N]on-specified fields” have a default value, such as “-1.” (*Scahill*, 42:27-28.)

schedules an execution time that “coincide[s] with appropriate network conditions.” (Bims, ¶100; *Scahill*, 2:24-25.)

9. Claim 8: “The wireless end-user device of claim 1, wherein the API receives, from a given one of the applications, a network request as part of a request for an execution time.”

Rao/Scahill renders obvious claim 8. (Bims, ¶¶101-102.) As discussed in Section V.A.8 (claim 7), when sending a request via an API, “client application 12a first provides a tuple of data,” including “the required network profile.” (*Scahill*, 29:13-16.) “[T]he network profile” includes “the following fields: **Network type**, Min Uplink BW, Min Downlink BW, Max RTT, Max PacketLoss, and/or Max Jitter.” (*Scahill*, 42:21-23.) For example, an application “could specify a network profile as follows: ‘WLAN, -1,128000,-1,-1,-1,-1’ with delay of 172800[]secs and duration of 62 secs.” (*Scahill*, 42:25-31.) The API thus receives an application’s request for a WLAN network as part of a request for an execution time. (Bims, ¶101.)

From this, a POSITA would have been motivated and found it obvious to implement *Rao/Scahill*’s architecture such that *Rao/Scahill*’s application scheduler likewise receives, from an application, a network request as part of its request for an execution time via an API. (Bims, ¶102.) By receiving the network request in the application’s request, *Rao/Scahill*’s scheduler can schedule an execution time

that “coincide[s] with appropriate network conditions.” (*Id.*; *Scahill*, 2:24-25; see Section V.A.8 (claim 7).)

10. Claim 9: “The wireless end-user device of claim 8, wherein the network request specifies whether the network request specifies one of a roaming network or a non-roaming network as the connected network for the network request.”

Rao/Scahill renders obvious claim 9. (Bims, ¶¶103-108.) As quoted in Section V.A.9 (claim 8), applications in *Rao*’s architecture as modified by *Scahill* can request a “[n]etwork type” as part of the request for an execution time via an API. (Bims, ¶103.) These network types include either a non-roaming network (e.g., a network associated with a home network operator) or a roaming network (e.g., a partner of a home network operator that can provide roaming services). (*Id.*)

Indeed, *Scahill* discloses that its method improves on prior-art scheduling systems that struggle to schedule transmissions under “appropriate network conditions to support the type of data transmission required to complete the scheduled task.” (*Scahill*, 2:14-17.) This includes where a user is “located in a roaming data network” and “may be on a tariff.” (*Scahill*, 2:20-22.) *Scahill*’s method addresses this issue by allowing an application to specify a particular network to avoid roaming costs. (*Scahill*, 29:13-16, 42:21-23; Bims, ¶104.)

Scahill discloses an exemplary network request for a home, non-roaming network, such as a WLAN network: “WLAN, -1,128000,-1,-1,-1,-1.” (*Scahill*, 42:25-31.) A POSITA would have understood that *Scahill*’s “WLAN” request is a request for a non-roaming network because the user may access the network without roaming network charges. (Bims, ¶105.)

In addition, a POSITA would have understood the distinction between a non-roaming and roaming network in the WWAN context, including the financial costs associated with using a roaming network relative to a non-roaming network. (Bims, ¶106; *see Srinivasan*, [0003]; *Scahill*, 2:20-22; *Cole*, [0008].) A POSITA would thus also have been motivated to implement a network request to specify the need for the more cost-effective, non-roaming WWAN network for certain tasks. (Bims, ¶107.) For example, where an application requires transmission of significant data (e.g., a large media file), it would be desirable for the application to specify the use of a non-roaming network to minimize user costs. (*Id.*; *Cole*, [0008].) From this, a POSITA would have been motivated and found it obvious to implement *Rao/Scahill*’s network requests (as discussed in claim 8) to allow an application to specify one of a roaming network or a non-roaming network to use for the network request. (Bims, ¶108.) Such a modification would be well within a POSITA’s abilities (e.g., adding a single field to *Rao/Scahill*’s network profile

requests), and a POSITA would thus have had a reasonable expectation of success in carrying out this modification. (*Id.*)

11. Claim 10: “The wireless end-user device of claim 1, wherein the API receives, from a given one of the applications, a scheduled time request as part of a request for an execution time.”

Rao/Scahill renders obvious claim 10. (Bims, ¶¶109-111.) As explained in Section V.A.4.d (element 1[c]), *Scahill* (as implemented in *Rao*’s modified architecture) discloses that an application’s request specifies “application specific requirement(s)” (also called “connection conditions”) for the application, “which the application scheduler library component 14a,b of each client application 12a,b ... must take into account when seeking to schedule performance of a task.” (*Scahill*, 19:3-7, 24:14-17.) One such connection condition is a “timing interval Δt requested by the client application since the task was last performed.” (*Scahill*, 24:19-23.) For example, an application could specify a “delay of 172800[]secs and duration of 62 secs.” (*Scahill*, 42:25-31.) A POSITA would have understood *Scahill*’s timing interval Δt to be a “scheduled time request” because it is a request for a particular time, as indicated by the delay of precise duration (specified in seconds) after the scheduled connection window (the “period of time of predetermined duration” that was scheduled at “an absolute time value”). (Bims, ¶110; *Scahill*, 19:25-26, 26:17-23.)

Thus, *Scahill* discloses that the scheduler receives, from an application, a timing interval (“scheduled time request”) as part of its request for an execution time via an API. (Bims, ¶111.) From this, a POSITA would have been motivated and found it obvious to implement *Rao/Scahill*’s architecture such that *Rao/Scahill*’s application scheduler receives, from an application, a timing interval as part of its request for an execution time via an API. (*Id.*) A POSITA would have recognized that, by receiving the timing interval in the application’s request, *Rao/Scahill*’s application scheduler schedules an execution time for the next performance of the application’s task that is appropriate for the task. (*Id.*) As *Scahill* explains, “[t]he timing interval Δt may vary for some client applications,” e.g., depending on the “amount of data” that “require[s] backing up” or dependency among tasks. (*Scahill*, 24:26-32.)

12. Claim 11: “The wireless end-user device of claim 1, further comprising a user interface allowing a user to specify one or more of the network access policy controls to be associated with a given one of the plurality of executable applications.”

Rao/Scahill renders obvious claim 11. (Bims, ¶112.) According to *Rao*, “policies 520 may be configurable by a user by any suitable means and/or mechanism. For example, the agent 326 may provide a configuration mechanism such as a user interface, graphical or otherwise, design[ed] and constructed for

configuring or specifying the policies 520.” (*Rao*, [0183].) These policies are associated with *Rao/Scahill*’s applications. (*Rao*, [0182], [0189]; Bims, ¶112.)

13. Claim 13: “The wireless end-user device of claim 1, wherein to dynamically associate one or more corresponding network access policy controls comprises changing an association of network access policy controls based on a device usage state.”

Rao/Scahill renders obvious claim 13. (Bims, ¶¶113-114.) *Rao* discloses that, in addition to making a determination based on whether the application is in the foreground, *Rao*’s remote access client “may determine any other characteristics or statistics of the application 338a-338n [for the purpose of assessing priority] such as ... total execution time[] and/or frequency of use.” (*Rao*, [0188]-[0189].) A POSITA would have understood these characteristics reflect a “device usage state” because they reflect how the device is being used. (Bims, ¶113.)

From *Rao/Scahill*, a POSITA would have understood that, based on these “device usage state” characteristics, *Rao/Scahill*’s remote access client may dynamically change an association of network access policy controls. (Bims, ¶114.) For instance, when the application requesting network access is in the background, *Rao/Scahill*’s remote access client may associate a policy to deprioritize that application (and prioritize foreground applications) in light of how the device is being used (e.g., a total execution time or frequency of use does not

exceed a predetermined threshold). (*Id.*) *Rao/Scahill*'s remote access client may then dynamically change the associated policy to prioritize that background application if how the device is being used changes (e.g., the total execution time or frequency of use now exceeds a predetermined threshold). (*Id.*)

14. Claim 15: “The wireless end-user device of claim 1, wherein the launch manager receives a list identifying executable applications that have a launch control policy in effect.”

Rao/Scahill renders obvious claim 15. (Bims, ¶¶115-117.) *Rao* discloses that filter 322 (“launch manager”) receives an indication of the applications that have one of policies 520 (“launch control policy”) in effect. (Bims, ¶115.)

Rao discloses identifying the applications that have a policy in effect because, in *Rao*, “the policies 520 may be specified by the name of the application 338a-338n.” (*Rao*, [0182].) *Rao* discloses that filter 322 (“launch manager”) receives a list identifying these applications because “[t]he policies may be accessible by the agent 326, configured into the agent 326, or loaded by the agent 326,” and “the agent 326 indicates the priority to the filter 322 for management of network packet queues 540a-540n to apply the indicated priorities.” (*Rao*, [0183], [0190].) A POSITA would have understood such an indication of the policies, specified by the name of the application, to be the claimed “list.” (*Bims*, ¶115.)

15. Claim 16: “The wireless end-user device of claim 1, wherein the launch manager is integrated with an operating system broadcast intent manager.”

Rao/Scahill renders obvious claim 16. (Bims, ¶¶118-121.) Notably, the ’700 patent neither purports to have invented an “operating system broadcast intent manager” nor explains how to integrate a launch manager into an operating system. Instead, it discloses that the claimed techniques “that allow for intercepting a service activity intention to launch, and applying a background service policy set or a network protection service policy set can be designed into the OS itself.” (’700 patent, 67:33-37.) For example, the ’700 patent discloses that “the intercept and policy implementation functions can be designed into the activity manager, broadcast intent man[a]ger, media service manager, service manager, or other application or service activity management function in the Android OS.” (*Id.*, 67:37-41.) The ’700 patent thus recognizes the claimed “broadcast intent manager” as a known type of application manager in the Android OS.⁴ (Bims, ¶118.)

⁴ These disclosures are consistent with the specification citations identified by Patent Owner as relating to this term in the district court litigation. (Ex-1023, 4.)

The '700 patent further acknowledges that “[o]ne of ordinary skill in the art will recognize that ... [the claimed prioritization techniques] can be designed into application launch management functions in the iPhone OS, windows mobile OS, windows PC OS, Blackberry OS, Palm OS, and other OS designs.” ('700 patent, 67:41-49.) The '700 patent, however, provides no details regarding this integration, recognizing it was within the abilities of a POSITA. (Bims, ¶119.)

As discussed in Section V.A.4.e (element 1[d]), *Rao*'s filter 322 selects and schedules execution times and is the claimed “launch manager.” *Rao* discloses that its devices “typically operate under the control of operating systems, which control scheduling of tasks and access to system resources.” (*Rao*, [0128].) *Rao* further discloses that its device “can be running any operating system,” including “any embedded operating system, any real-time operating system, any open source operating system, any proprietary operating system, any operating systems for mobile computing devices, or any other operating system capable of running on the computing device and performing the operations described herein.” (*Id.*) A POSITA would thus have been motivated to implement *Rao/Scahill*'s architecture in known, commercially available operating systems. (Bims, ¶¶120-121.) As of the '700 patent's priority date, this would have included Android OS. (Bims, ¶121; *Huq*, 7:14-17.) A POSITA would have understood that *Rao*'s filter would be implemented into the component of the operating system that schedules tasks and

access to system resources. (Bims, ¶121.) In the case of Android OS, this would include the “broadcast intent manager” for managing scheduling and intents. (*Id.*; ’700 patent, 67:37-41.)

16. Claim 17: “The wireless end-user device of claim 1, wherein, based on a given one of the network access policy controls, the launch manager deprioritizes one or more processes associated with a given one of the executable applications.”

Rao/Scahill renders obvious claim 17. (Bims, ¶122.) As discussed in Section V.A.4.e (element 1[d]), filter 322 (“launch manager”) arranges network communications from applications into a queue according to their respective policies. For example, *Rao*’s policies “define prioritization based on whether an application is running in the foreground or the background.” (*Rao*, [0182].) These policies “may be specified conditionally, such as if one application 338a is running, a second application 338b may have a higher or lower priority.” (*Id.*) Thus, where a first application moves from the foreground to the background and a second application moves from the background to the foreground, filter 322 deprioritizes the first application (and its associated processes or network packets) based on the policies identified by agent 326. (Bims, ¶122; *see also Rao*, [0189], [0190].)

B. Ground 2: Claim 3 Is Obvious over *Rao* in View of *Scahill* and *Oestvall*

1. Overview of *Oestvall*

Oestvall discloses techniques to “preserve or conserve resources” on mobile devices. (*Oestvall*, [0002].) Like the ’700 patent, *Oestvall* describes “multi-tasking devices, i.e., devices with an operating system that can run several applications at the same time.” (*Oestvall*, [0007].) This includes foreground and background applications. (*Oestvall*, [0006].) *Oestvall* is analogous art to the ’700 patent. (Bims, ¶¶123-124.)

According to *Oestvall*, conventional schedulers used for multitasking may be configured such that “applications will continue to run even when not actually in active use,” which results in the applications continuing to “use some system resources, even when residing in the ‘background’.” (*Oestvall*, [0006].) *Oestvall* addresses these resource challenges using techniques that “deny[] system resources and services to background applications that do not meet predefined ‘trust’ or certification criteria.” (*Oestvall*, [0010].)

2. Rationale to Combine

Rao contrasts “an application [that] is running in the foreground and currently in active use by the user” with “an application running in the background.” (*Rao*, [0003]; Bims, ¶125.) *Rao*, however, does not expressly

describe how to determine whether an application is running in the background.

(Bims, ¶125.) As a result, a POSITA implementing *Rao*'s teachings (in the context of *Rao/Scahill*) would have looked to similar references, including *Oestvall*, to understand how to classify an application as running in the background. (*Id.*)

Oestvall teaches “[a]n application is in the background if it is not being interacted with by an end-user and it presents no user interface with which a user could interact.” (*Oestvall*, [0006].) A POSITA would have understood that *Oestvall* addresses concerns associated with conventional classification methods. (Bims, ¶126; *see Freund*, 8:63-9:3, 10:5-43 (discussing concerns).) *Oestvall*'s approach would advantageously allow *Rao/Scahill*'s architecture to more accurately classify background applications and enable *Rao/Scahill*'s prioritization based on whether an application is running in the foreground or background. (Bims, ¶126.) A POSITA would have been motivated to identify applications running in the background because, as *Rao* and *Oestvall* both recognize, such background applications can cause latency that affects foreground applications. (*Id.*) A POSITA would have understood that the proposed combination involves (1) combining prior-art elements according to known methods to yield predictable results; and (2) use of a known technique to improve similar devices in the same way. (*Id.*)

A POSITA would have also had a reasonable expectation of success in modifying *Rao/Scahill*'s architecture in view of *Oestvall*'s classification method. (Bims, ¶127.) *Rao*'s device already evaluates whether a device is in active use (e.g., to classify whether an application is running in the foreground), and *Oestvall*'s classification method only requires the additional step of determining that the application presents no user interface with which a user could interact. (*Id.*) Such a modification would have been well within a POSITA's abilities to implement. (*Id.*) See *Keynetik*, 2023 WL 2003932, at *2.

3. Claim 3: “The wireless end-user device of claim 2, the activity classifier determining that a given one of the applications is a network access controlled activity or a blocked activity based on a lack of user interaction with that application for a period of time.”

Rao/Scahill/Oestvall renders obvious claim 3. (Bims, ¶¶128-129.) Claim 3 recites an “or” between “network access controlled activity” and “blocked activity,” so only one of the two limitations must be taught to satisfy claim 3. *Brown v. 3M*, 265 F.3d 1349, 1352-53 (Fed. Cir. 2001).

Rao discloses that remote access client 120, including agent 326 (“activity classifier”), determines whether an application associated with the network packet is running in the background. (*Rao*, [0188]; see also *Rao*, [0189].) As discussed in Section V.A.5 (claim 2), a background application is a “network access controlled activit[y]” because it may be deprioritized for network access. (Bims, ¶128.)

As discussed above, a POSITA would have been motivated to modify *Rao/Scahill*'s architecture to classify whether an application is running in the background based on *Oestvall*'s assessment that an application "is not being interacted with by an end-user and it presents no user interface with which a user could interact." (*Oestvall*, [0006]; Bims, ¶129.) As Dr. Bims explains, this would have been evaluated over a period of time (rather than happening instantaneously) so as to ensure that a user was indeed not interacting with a particular application. (Bims, ¶129; *Wyld*, 2:38-43, 4:49-57, 5:12-19.) In *Rao/Scahill/Oestvall*'s architecture, filter 322 classifies whether an application is running in the background based on a determination that an application is not being interacted with by an end-user for a period of time and the application presents no user interface with which a user could interact. (Bims, ¶129.)

C. Ground 3: Claims 6, 14, and 18-19 Are Obvious over *Rao* in View of *Scahill* and *Montemurro*

1. Overview of *Montemurro*

Montemurro teaches application-aware optimization of network transmissions from a mobile device. Just as *Rao* teaches application-aware prioritization, *Montemurro* discloses application-aware "policy-based routing of communications among two or more modes of wireless communication."

(*Montemurro*, [0001].) *Montemurro* is analogous art to the '700 patent. (Bims, ¶¶130-133.)

Devices with such connectivity capabilities are “dual or multi-mode devices,” with “radio access technologies that provide access to” multiple network types, including WLAN and WWAN. (*Montemurro*, [0003].) But “[t]here are costs associated with application access from these different networks,” and “[i]t is therefore desirable to have a mechanism that seeks to optimize communications.” (*Montemurro*, [0004].)

To optimize communications, *Montemurro* discloses “a rules engine 206A ... using rules to configure TCP/IP network stack 206B and connection table 206C to coordinate communications for applications 204 using communications interfaces [208].” (*Montemurro*, [0027].) This rules engine “configures the communication operations with a set of rules/policies that could include various factors such as radio access technology (e.g., [.] for high/low bandwidth properties), cost, presence, time of day, location (e.g., [.] geo-based policies, network roaming), destination IP address, application type, and Quality of Service (QoS) requirements.” (*Montemurro*, [0029].)

Rules engine 206A “configures ... connection table 206C and routing table 210 to optimize the flow of communications over multiple communications modes.” (*Montemurro*, [0034].) In response to a state change (e.g., “time of day”

or “connecting/disconnecting of device 102 with a specific network 104 and 106”), rules engine 206A executes and modifies “routing table 210 to ensure that data goes out [to] the most appropriate network (via respective network interface 208).” (*Montemurro*, [0035].) Rules engine 206A further “determine[s] which interface would be best used to service a particular application.” (*Id.*)

2. Rationale to Combine

As described in Section V.A.3 (Ground 1, Rationale to Combine), a POSITA would have been motivated to improve *Rao*’s architecture by including *Scahill*’s scheduling method. (*See Rao*, [0179]-[0195].) In implementing this modification, a POSITA would have recognized opportunities to provide further optimization in scenarios where *Rao/Scahill*’s device has access to multiple networks.⁵ (*Bims*,

⁵ Although *Scahill*’s scheduling method allows an application to request a particular network (*e.g.*, claim 8), this network request is driven by the application—not the device (*i.e.*, remote access client). (*Bims*, ¶134 n.3.) *Montemurro*’s user-configurable network policies allow the remote access client (or a user through a user interface) to optimize the communication flow over multiple communications modes. (*Id.*; *see also Montemurro*, [0036] (disclosing a user interface).)

¶134; *Rao*, [0095] (describing the ability to connect to multiple networks); *see also Montemurro*, [0003], [0004].) This would have motivated a POSITA to seek solutions beyond *Rao* and *Araujo* to achieve such additional efficiencies. (Bims, ¶¶134-138.)

According to *Montemurro*, devices with multiple radio access technologies can access more than one network, including at least “wireless local area networks (WLAN)” and “wireless wide area networks (WWAN) (e.g.,] cellular technologies[)].” (*Montemurro*, [0003].) But “[t]here are costs associated with application access from these different networks.” (*Montemurro*, [0004].) And “[t]here are also quality considerations such as the speed at which content can be delivered, or in the case of streaming media, the quality of service at which the content is delivered.” (*Id.*) “It is therefore desirable to have a mechanism that seeks to optimize communications for multi-mode capable devices, that is, that seeks to improve communications for multi-mode capable devices.” (*Id.*)

Given *Rao*’s disclosure of multi-mode devices that “connect to one or more networks” (*Rao*, [0095]), a POSITA would have been motivated to improve *Rao/Scahill*’s architecture by augmenting *Rao/Scahill*’s policies to include *Montemurro*’s policy-based data routing for multimode operations. (Bims, ¶¶135-136.) Incorporating these teachings into *Rao/Scahill* would have resulted in the ability to “offer different usage models depending on the mode of wireless

operation selected.” (*Montemurro*, [0003].) *Montemurro*’s policies would have enabled *Rao*’s prioritization architecture to further account—at a policy level—for “radio access technology (e.g.,] for high/low bandwidth properties), cost, presence, time of day, location (e.g.,] geo-based policies, network roaming), destination IP address, application type, and Quality of Service (QoS) requirements.” (*Montemurro*, [0029].)

A POSITA would have had a reasonable expectation of success in modifying *Rao/Scahill*’s architecture in view of *Montemurro*’s policies to augment *Rao/Scahill*’s existing policies. (Bims, ¶137.) *Rao* discloses that a POSITA “will recognize and appreciate the multitude of ways to define client-side application priorities.” (*Rao*, [0182].) *Rao* explains that policies: (1) “may be provided by or downloaded via the gateway 340,” (2) “may comprise any type and format of syntax and/or language for specifying a policy,” and (3) “may be provided via any type and/or form of medium.” (*Rao*, [0183].) These policies also “may be configurable by a user by any suitable means and/or mechanism.” (*Id.*) The result is that the modification would have been well within the abilities of an ordinary artisan to implement. (Bims, ¶137.) See *Keynetik*, 2023 WL 2003932, at *2.

3. Claim 6: “The wireless end-user device of claim 1, wherein according to at least one of the network access policy controls, the launch manager delays an execution time for one or more of the applications until a connection is made through the at least one wireless modem to an alternative network.”

Rao/Scahill/Montemurro renders obvious claim 6. (Bims, ¶¶139-141.) As discussed in Section V.C.2 (Ground 3, Rationale to Combine), a POSITA would have been motivated to augment *Rao*’s policies to additionally account for “various factors such as radio access technology (e.g.,] for high/low bandwidth properties)” and “location (e.g.,] geo-based policies, network roaming),” as taught by *Montemurro*. (*Montemurro*, [0029].) In particular, a POSITA would have been motivated to augment *Rao*’s policies to allow an application differing network access on different networks. (Bims, ¶139.) Such augmented policies would cause a launch manager to delay an application’s execution time until a connection is made through a device’s modem to an alternative network. (*Id.*)

For example, a POSITA would have been motivated to augment *Rao*’s policies to allow an application differing network access over a WLAN network than over a WWAN network. (Bims, ¶140.) When a WLAN network is available, an application’s network access may be specified by a policy that permits large transmissions of data, while when only a WWAN network is available, an application’s network access may be specified by a policy that does not permit large transmissions of data. (*Id.*) For example, in *Montemurro*, a “CRM application

(i.e., one of 204B) only synchronizes the sales contact database (e.g., 120B) over a WLAN network 106.” (*Montemurro*, [0029].) In *Rao/Scahill/Montemurro*’s architecture, agent 326 would identify a radio access technology policy for a CRM application, and filter 322 (“launch manager”) would delay executing the CRM application synchronization while *Rao*’s is connected to a WWAN, to minimize data usage. (Bims, ¶140.) But, once the device connects to a home WLAN, filter 322 would allow execution of the application under the network policy. (*Id.*)

4. Claim 14: “The wireless end-user device of claim 1, wherein to dynamically associate one or more corresponding network access policy controls comprises changing an association of network access policy controls based on a change in a current network to which the at least one wireless modem is providing network access.”

Rao/Scahill/Montemurro renders obvious claim 14. (Bims, ¶142.) As discussed in Section V.C.2 (Ground 3, Rationale to Combine), a POSITA would have been motivated by *Montemurro* to augment *Rao*’s policies to specify one policy for an application’s network access over a WLAN network and another policy for an application’s network access over a WWAN network to account for “various factors such as radio access technology (e.g., for high/low bandwidth properties)” and “location (e.g., geo-based policies, network roaming).” (*Montemurro*, [0029].) A POSITA would have understood that a radio access technology policy or a location-based policy focused on network roaming would

meet claim 14. (Bims, ¶142; *see also* Section V.A.10 (claim 9) (discussing roaming).)

As a device moves from one location connected to a home network (e.g., WLAN) to a second, out-of-network location (e.g., WWAN), the corresponding network access policy would change. (Bims, ¶142.) For example, a “CRM application (i.e., one of 204B) only synchronizes the sales contact database (e.g., 120B) over a WLAN network 106.” (*Montemurro*, [0029].) If the device with the CRM application moves from one location with a non-WLAN connection to a location with a WLAN connection, the network access policy would change with the network change. (Bims, ¶142.) As discussed in Section V.A.4.c (element 1[b]), this change would be dynamic; that is, the policy would change along with the location change and associated network change. (*Id.*)

5. Claim 18: “The wireless end-user device of claim 1, wherein, based on a given one of the network access policy controls, the launch manager controls the frequency at which a given one of the executable applications is allowed network access.”

Rao/Scahill/Montemurro renders obvious claim 18. (Bims, ¶¶143-146.) *Montemurro* discloses policies that consider “radio access technology (e.g., for high/low bandwidth properties), cost, presence, time of day, location (e.g., geo-based policies, network roaming),” and “Quality of Service (QoS) requirements.” (*Montemurro*, [0029].) As discussed in Section V.C.2 (Ground 3, Rationale to

Combine), a POSITA would have been motivated to modify *Rao/Scahill*'s architecture in view of *Montemurro*'s policies to augment *Rao*'s existing policies. (Bims, ¶143.) These policies (including those discussed above) would each enable filter 322 ("launch manager") to control the frequency at which a given one of the executable applications is allowed network access. (*Id.*)

For example, a policy based on QoS requirements includes transmission rates. (Bims, ¶144.) Indeed, Dr. Bims describes an example where one application requires a transmission every 20 ms but another requires less frequent transmission to satisfy QoS requirements. (*Id.*) A POSITA would have understood that a policy directed to transmission rates would advantageously allow optimization of transmission on an application-by-application basis. (*Id.*) Additionally, a policy based on time of day, as disclosed by *Montemurro* and incorporated into the *Rao/Scahill* combination above, would alter the frequency at which an application is allowed network access to a particular network type. (*Montemurro*, [0035].) This would include a policy that only allows certain applications, such as high-bandwidth applications, to access a given network type (such as a cellular network or WLAN) during off-peak hours (e.g., before 9:00 a.m. or after 5:00 p.m.) so as to optimize data usage. (Bims, ¶145.) If that policy were removed (or changed), the frequency at which that application is allowed network access would change accordingly. (*Id.*)

From this, a POSITA would have been motivated and found it obvious to augment *Rao/Scahill*'s policies in view of *Montemurro* to include policies that control the frequency at which that application is allowed network access. (Bims, ¶146.) Such a modification would be well within a POSITA's abilities (e.g., adding additional policies to *Rao*'s remote access client), and a POSITA would thus have had a reasonable expectation of success in carrying out this modification. (*Id.*) See *Keynetik*, 2023 WL 2003932, at *2.

6. Claim 19: “The wireless end-user device of claim 1, wherein, based on an amount of network data communicated to or from a given one of the executable applications, the activity classifier dynamically changes at least one network access policy control associated with that given executable application.”

Rao/Scahill/Montemurro renders obvious claim 19. (Bims, ¶¶147-149.) *Montemurro* discloses “a set of rules/policies that could include various factors such as radio access technology (e.g., [] for high/low bandwidth properties), cost, ... [and] location (e.g. geo-based policies, network roaming).” (*Montemurro*, [0029].) As discussed in Section V.C.2 (Ground 3, Rationale to Combine), a POSITA would have been motivated to modify *Rao/Scahill*'s architecture in view of *Montemurro*'s policies to augment *Rao*'s existing policies. (Bims, ¶147.) *Rao/Scahill*'s “agent 326,” which operates as the claimed activity classifier, would dynamically apply cost policies based on an amount of network data communicated to or from a given one of the executable applications. (*Id.*)

A POSITA would have understood that different network connections have different bandwidth availability, different associated costs, and in some instances may only allow a fixed amount of data transmission or bandwidth before incurring costs or the network administrator throttles the network speed. (Bims, ¶148.) The result is that, if a network access policy is cost-based, as taught by *Montemurro*, *Rao/Scahill*'s agent 326 would advantageously only allow an application to use a certain amount of data before dynamically associating a new network access policy for that application so as to prevent the user from incurring unexpected costs (e.g., by going over a fixed monthly data allotment). (*Id.*) This application-specific limit might be, for example, a predetermined percentage of the overall fixed data allotment (for an entire device) to ensure that one application does not alone lead to exceeding allowable limits and preserves network resources for other applications. (*Id.*)

From this, a POSITA would have been motivated and found it obvious to augment *Rao/Scahill*'s policies in view of *Montemurro* to apply cost policies based on an amount of network data communicated to or from a given one of the executable applications. (Bims, ¶149.) Such a modification would be well within a POSITA's abilities (e.g., adding additional policies to *Rao*'s remote access client), and a POSITA would thus have had a reasonable expectation of success in carrying out this modification. (*Id.*)

D. Ground 4: Claims 12 and 20 Are Obvious over *Rao* in View of *Scahill* and *Araujo*

1. Overview of *Araujo*

Araujo teaches application-aware optimization of power management.

(Bims, ¶150.) Like the '700 patent and *Rao*, *Araujo* implements policies to control activities of “wireless telephones” on an application-by-application basis. (*Araujo*, [0014].) *Araujo* is analogous art to the '700 patent. (Bims, ¶¶150-153.)

Araujo describes “a power-management policy” for managing “power usage.” (*Araujo*, Abstract.) The policy is implemented on “[e]lectronic devices,” such as “wireless telephones.” (*Araujo*, [0014].) The policy is applied to “justify [a program] consuming power, depending on what the program is doing.” (*Araujo*, [0004].) By implementing this policy, a device better “use[s] the energy in a way that strikes a balance between providing functionality and maintaining longevity of the charge.” (*Araujo*, [0001].)

Araujo classifies “programs” using various criteria and selectively provides resources based on their classifications. (*Araujo*, [0027].) A program’s ability to consume power is “based on a finding as to whether the program’s status justifies the consumption of power under the circumstances that are present.” (*Id.*) “[S]ome programs may have a ‘VIP’ status that allows them to consume power in situations where other programs would not be permitted to consume power.” (*Id.*)

Araujo teaches that a device can provide resources to programs differently, based on a program's "worthiness" to consume power. (*Araujo*, [0027].) A program generates "a request to perform an action using a particular device," such as "a request to send data over a wireless network." (*Araujo*, [0020], [0024].) The device then performs various responsive operations, including allowing the request (*Araujo*, [0035], [0037], Fig. 3 (302)) or blocking the request (*Araujo*, [0035], [0040], [0050], Fig. 4 (406)). These operations are provided by a "request deflector component" that is "interposed" between a "program" and a "[device] driver," and configured to "intercept[] requests and respond[] to the requests based on power management considerations before the request reach[es] a device driver." (*Araujo*, [0024], [0050].)

2. Rationale to Combine

It would have been obvious to a POSITA to modify *Rao/Scahill's* architecture to include *Araujo's* power management policy to provide a combination that better "use[s] the energy in a way that strikes a balance between providing functionality and maintaining longevity of the charge." (*Araujo*, [0001]; Bims, ¶¶154-162.)

Battery-powered devices "have a finite amount of energy stored between charges." (*Id.*) These devices therefore "attempt to use the energy in a way that strikes a balance between providing functionality and maintaining longevity of the

charge,” and “employ power management schemes to govern the use of energy.”

(*Id.*) The prior art, however, recognized a need “for an improved power management operation in portable devices.” (*Aleksic*, [0004].) This was especially true for devices running multiple applications and having access to multiple network connections. (Bims, ¶155; *see also D’Amore*, [0031]; *Araujo*, [0002]-[0005].) One method of improving a device’s power management was an application-aware prioritization of network communications that optimized the power consumption of individual applications. (Bims, ¶155; *Araujo*, [0002]-[0005]; *D’Amore*, [0031]; *Rao*, [0080].)

A POSITA would have also understood that improved, application-specific, power management schemes reduce competition among applications for other device resources, including network resources and processing power. (Bims, ¶156.) By selectively allowing applications to access network resources, power management schemes reduce competition among applications for network resources. (*Id.*) The result is that certain applications access network resources at the expense of others. (*Id.*) This is particularly helpful, for example, when accessing a low bandwidth network or running a high-priority application requiring more processing power. (*Id.*) Using a power management scheme to manage an individual application’s power consumption optimizes a device’s resource consumption, improving an application’s operation and user experience. (*Id.*)

Indeed, *Rao* recognized that its policy-based prioritization techniques “may improve or increase the performance, operational characteristics, and user experience on the client based on the applications running on the client.” (*Rao*, [0080].) A POSITA would have understood that “performance, operational characteristics, and user experience” each implicate battery power, longevity of a device, and availability of network resources. (Bims, ¶157.) A POSITA would have thus been motivated to modify *Rao/Scahill* to include well-known, application-aware, power-monitoring capabilities that reduce power consumption, decrease energy expenses, optimize network resource allocation, and increase device longevity. (*Id.*)

Araujo’s power management policy optimizes power consumption by applying application-aware prioritization of network communications based on a power state of a device. (*Araujo*, [0003]-[0005].) *Rao/Scahill*’s architecture, as modified in view of *Araujo*’s power management policy, uses “energy in a way that strikes a balance between providing functionality and maintaining longevity of the charge.” (*Araujo*, [0001].)

To implement these improvements, a POSITA would have looked to *Araujo*’s disclosure to selectively allow or block an application from transmitting data to the network depending on various factors, including the power state of the device and the “status” of the particular application requesting network access.

(*Araujo*, [0027], Fig. 2; Bims, ¶¶157-159.) Using this functionality, *Rao/Scahill*'s architecture would assign a "status" to an application, use that status to determine if allowing an application request "justifies the consumption of power under the circumstances that are present," and regulate (e.g., blocking, allowing) the request based on that determination. (*Araujo*, [0025], [0027].)

Araujo's request deflector functionality may be implemented in connection with or as part of the filter 322 of *Rao*'s remote access client, which—like *Araujo*'s request deflector—applies application-specific policies to determine how to handle applications' network access attempts. (Bims, ¶¶160-161; *Araujo*, [0024], [0050]; *Rao*, [0191]-[0193].) The request deflector determines whether applications have a "status [that] justifies the consumption of power under the circumstances that are present," and selectively blocks packets from proceeding to the network driver based on the determination. (*Araujo*, [0027]; Bims, ¶160.)

A POSITA would have had a reasonable expectation of success in modifying *Rao* to incorporate *Araujo*'s power management functionality. (Bims, ¶162.) Such a modification would have been well within the abilities of a POSITA. (*Id.*) For example, *Araujo*'s power management policy "can be implemented as software that is stored in one or more of the data remembrance component(s) 704 and that executes on one or more of the processor(s) 702." (*Araujo*, [0055].) This scheme could thus be implemented using the processor and memory units in

Rao/Scahill's architecture. (See *Rao*, [0119], [0120]; Bims, ¶162.) See *Keynetik*, 2023 WL 2003932, at *2. The resulting modification would have involved combining prior-art elements according to known methods to yield predictable results and use of a known technique to improve similar devices in the same way.

3. Claim 12: “The wireless end-user device of claim 1, wherein to dynamically associate one or more corresponding network access policy controls comprises changing an association of network access policy controls based on a power state of the device.”

Rao/Scahill/Araujo renders obvious claim 12. (Bims, ¶¶163-167.) According to *Araujo*, “[e]lectronic devices (e.g., computers, wireless telephones, audio/video equipment, etc.) normally employ some type of power management framework.” (*Araujo*, [0014].) *Araujo* uses the term “power state” to describe the power condition of a device, such as a “wireless telephone.” (*Araujo*, [0004], [0014] (referring to “a wireless telephone” that “may turn off its display” in a low-power mode).) *Araujo* discloses policies to be employed in a power management framework that are based on a device’s power state. (*Araujo*, [0004], [0014].)

A POSITA, when reviewing *Araujo*, would also read *Araujo*’s disclosure in the context of the prior art that formed a POSITA’s background knowledge. See *Ariosa Diagnostics v. Verinata Health, Inc.*, 805 F.3d 1359, 1365 (Fed. Cir. 2015). This includes *Cole*, which teaches that a “power state”—the same phrase used in *Araujo*—applies to mobile devices. (Bims, ¶164; see *Cole*, [0090].) Thus, a

POSITA would have understood that, in *Rao/Scahill/Araujo*'s architecture, a power control state is a power state of the device. (Bims, ¶164.)

According to *Araujo*, “devices may have various power states,” including “[o]n” and “off,” as well as “power states other than on and off.” (*Araujo*, [0004]; *see also Cole*, [0090].) Other examples include “a lower-power-consumption mode.” (*Araujo*, [0014].) *Rao/Scahill*'s architecture, as modified by *Araujo*, applies network policies based on a device's power state (e.g., “a lower-power-consumption mode”) and based on whether an application's status is worthy of consuming power by accessing the Internet. (*Araujo*, Abstract, [0004], [0014], [0027].)

Araujo discloses that one example of consuming power is executing a network transmission by using a network card (i.e., modem). (*Araujo*, [0021].) “For the purpose of power management, a program [that is, application] may have a status that reflects its worthiness to consume power [e.g., by accessing the Internet]. For example, some programs may have a ‘VIP’ status that allows them to consume power in situations where other programs would not be permitted to consume power.” (*Araujo*, [0027].) “Programs could be assigned binary VIP/non-VIP status, or their status could be indicated on a scale.” (*Id.*) Ultimately, “[w]hether a program is permitted to consume power could be based on a finding

as to whether the program's status justifies the consumption of power under the circumstances that are present.” (*Id.*)

Thus, when a power conservation policy is in effect in *Rao/Scahill/Araujo*'s architecture, a non-VIP application may be subject to a network access policy that restricts (or blocks) that application's ability to consume power by utilizing a network connection. (Bims, ¶¶165-167; *Araujo*, [0027].) But when a power conservation policy is not in effect, that same non-VIP application may be subject to a different policy that does not restrict power consumption, including network access. (Bims, ¶¶165-167; *Araujo*, [0027].) As the device's power state changes, *Rao/Scahill/Araujo*'s remote access client dynamically changes the policy associated with the non-VIP application. (Bims, ¶167; *Araujo*, [0025]; *Cole*, [0090].)

4. Claim 20: “The wireless end-user device of claim 1, wherein the launch manager schedules corresponding execution times for applications to correspond with power state changes of the at least one wireless modem.”

Rao/Scahill/Araujo renders obvious claim 20. (Bims, ¶¶168-171.) As discussed in Section V.D.2 (Ground 4, Rationale to Combine), a POSITA would have been motivated to modify *Rao/Scahill*'s architecture to include *Araujo*'s power management policy. (Bims, ¶168.) When *Rao/Scahill/Araujo*'s device

operates in “batch” mode, this policy schedules task executions based on whether a wireless modem is powered on or off. (*Id.*)

A POSITA would have understood that wireless devices, including *Rao*’s device, have the ability to operate “in a ‘batch’ mode that conserves power by allowing network transmissions to happen in bursts rather than at the time ... that the outbound network traffic is generated.” (*Araujo*, [0043]; *Bims*, ¶169; *see also Haughn*, [0049].) A POSITA would have further understood that this mode, which was a common feature of devices by the ’700 patent’s priority date, “facilitate[s] a more efficient utilization of radio resources” by controlling when large data transmissions occur. (*Haughn*, [0049].)

Araujo discloses that “[t]he power management policy for such a mode might call for the network card [(i.e., modem)] to be turned on at specific intervals (e.g., every fifteen minutes).” (*Araujo*, [0043].) When the modem is powered on and powered off, the modem undergoes a “power state change[.]” (*Bims*, ¶170.) According to *Araujo*, “[i]f a request to transmit data over a network is made between these intervals, the request may be delayed until the next time that the network card is scheduled to be powered on, at which time the request could be allowed.” (*Araujo*, [0043].)

From this, a POSITA would have been motivated and found it obvious to augment *Rao/Scahill*’s policies in view of *Araujo* to include policies that schedule

task executions based on whether a wireless modem is powered on or off when *Rao/Scahill*'s device operates in "batch" mode. (Bims, ¶171.) The result is that *Rao/Scahill*'s filter 322 ("launch manager") applies *Araujo*'s power management policy to schedule execution times for applications to correspond with modem power state changes. (*Id.*) Such a modification would be well within a POSITA's abilities (e.g., adding additional policies to *Rao*'s remote access client), and a POSITA would thus have had a reasonable expectation of success in carrying out this modification. (*Id.*)

VI. The Petition Is Proper Under 35 U.S.C. § 325(d)

Discretionary denial under § 325(d) is not warranted. *See Advanced Bionics, LLC v. MED-EL Elektromedizinische Geräte GmbH*, IPR2019-01469, Paper 6, 7-9 (PTAB Feb. 13, 2020) (precedential) ("*Advanced Bionics*") (citing *Becton, Dickinson, & Co. v. B. Braun Melsungen AG*, IPR2017-01586, Paper 8 (PTAB Dec. 15, 2017) and applying *Becton, Dickinson* factors (a)-(f) within the *Advanced Bionics* analysis).

Oestvall, Montemurro (issued as U.S. Patent No. 8,825,109), *Araujo*, and a relative of *Rao* (WO 2006/012610A2) were each cited during prosecution of the '700 patent. ('700 patent, References Cited.) Patent Owner submitted these references on an information disclosure statement, and they are among the over 1,775 references listed on the patent. (*Id.*) There is no evidence the Examiner

substantively considered these references. (*See* Ex-1002.) Simply citing a reference on an information disclosure statement, without more, does not favor denial of institution. *See Navistar, Inc. v. Fatigue Fracture Tech., LLC*, IPR2018-00853, Paper 13, 16-17 (PTAB Sept. 12, 2018). Moreover, the same or substantially the same arguments were not previously presented to the Office. Indeed, there could be no overlap between the arguments made before the Office because the Examiner issued no prior-art rejections during prosecution. (*See generally* Ex-1002 at 422-426, 612-627, 650.)

To the extent it is argued that the same or substantially the same art or arguments were previously presented to the Office, the second prong is not met. *Advanced Bionics*, 8-10. Although *Oestvall*, *Montemurro*, *Araujo*, and *Rao*'s relative were cited on the face of the '700 patent, none were addressed during prosecution. As this Petition shows, the Examiner failed to appreciate *Oestvall*'s, *Montemurro*'s, *Araujo*'s, and *Rao*'s disclosures and how those references in combination with other prior-art references not previously before the Office (e.g., *Scahill*) render the challenged claims obvious. The Examiner thus erred in a manner material to the patentability of the claims.

Moreover, Petitioners have shown a reasonable likelihood that at least one of the challenged claims is unpatentable over the applied art based on the current

record. Petitioners have demonstrated material error by the Office, and discretionary denial is not warranted.

VII. The Petition Is Proper Under *Fintiv*

The factors set forth in *Apple Inc. v. Fintiv Inc.*, IPR2020-00019, Paper 11, 5-6 (PTAB Mar. 20, 2020) (precedential), weigh in favor of instituting review under 35 U.S.C. § 314(a). Here, there is a parallel district court proceeding involving the same parties and patent. However, no trial date has been set. Moreover, the claim construction process has just begun, the district court has not provided any analysis of prior-art invalidity issues, and the parties are only in the preliminary stages of discovery. (Ex-1022.)

The Petition's merits are compelling and clear evidence of the unpatentability of the claims. Where the Board "determines that the information presented at the institution stage presents a compelling unpatentability challenge, that determination alone demonstrates that the PTAB should not discretionarily deny institution under *Fintiv*." USPTO Memorandum, Interim Procedure for Discretionary Denials in AIA Post-Grant Proceedings with Parallel District Court Litigation 4-5 (June 21, 2022). Because the factors favor instituting review, the Board should institute under § 314(a).

VIII. Mandatory Notices

A. Real Party-in-Interest

The real parties-in-interest for this Petition are Lenovo (United States) Inc. (“Lenovo US”) and Motorola Mobility LLC (“Motorola”). Petitioners further identify Lenovo Group Ltd. as a real party-in-interest solely because Lenovo Group Ltd. was previously named as a defendant in the parallel district court case listed below.

B. Related Matters

Patent Owner has asserted the ’700 patent against Lenovo US and Motorola in *Headwater Research LLC v. Motorola Mobility LLC, et al.*, No. 4:23-cv-04496-JST (N.D. Cal.) (filed Aug. 30, 2023).

The ’700 patent is also related to U.S. Patent No. 9,189,076, which is the subject of an *inter partes* review petition filed concurrently with this Petition.

C. Lead and Back-Up Counsel

<u>Lead Counsel</u>	<u>Back-Up Counsel</u>
Joshua L. Goldberg Reg. No. 59,369 joshua.goldberg@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 901 New York Avenue, NW Washington, DC 20001-4413 Tel: 202-408-6092 Fax: 202-408-4400	Benjamin A. Saidman Reg. No. 69,325 benjamin.saidman@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 271 17th Street, NW, Suite 1400 Atlanta, GA 30363-6209 Tel: 404-653-6510 Fax: 404-653-6444

<u>Lead Counsel</u>	<u>Back-Up Counsel</u>
	Daniel C. Cooley (Reg. No. 59,639) daniel.cooley@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 1875 Explorer Street Suite 800 Reston, VA 20190-6023 Tel: 571-203-2700 Fax: 202-408-4400 Grace K. Mills (Reg. No. 66,946) gracie.mills@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 901 New York Avenue, NW Washington, DC 20001-4413 Tel: 202-408-4000 Fax: 202-408-4400

D. Service Information

Petitioners consent to being served electronically at the e-mail addresses provided above and at Motorola-Headwater-IPRs@finnegan.com.

IX. Grounds for Standing

Petitioners certify that the '700 patent is available for IPR and that
Petitioners are not barred or estopped from requesting IPR challenging the patent
claims on the grounds in this Petition.

X. Conclusion

Petitioners request IPR and cancellation of claims 1-20. The Office may charge any required fees to Deposit Account No. 06-0916.

Date: August 12, 2024

Respectfully submitted,

By: /Joshua L. Goldberg/
Joshua L. Goldberg, Lead Counsel
Reg. No. 59,369

CERTIFICATION OF WORD COUNT

Pursuant to 37 C.F.R. § 42.24, Petitioners certify that the word count of Petitioners' Petition for *Inter Partes* Review (exclusive of any table of contents, table of authorities, mandatory notices under § 42.8, certificates of service and word count, and list of exhibits or claim listing) as measured by Microsoft Word is 13,674.

Dated: August 12, 2024

By: /Joshua L. Goldberg/
Joshua L. Goldberg, Lead Counsel
Reg. No. 59,369

CERTIFICATE OF SERVICE

Pursuant to 37 C.F.R. §§ 42.6(e) and 42.105(a), the undersigned certifies that on August 12, 2024, a copy of the foregoing Petition for *Inter Partes* Review, the associated powers of attorney, and Exhibits 1001-1023 were served by FedEx Priority Overnight on the correspondence address of record indicated in the Patent Office's Patent Center website for U.S. Patent No. 10,749,700:

Headwater Research LLC
c/o Farjami & Farjami LLP
26522 La Alameda Ave., Suite 360
Mission Viejo, CA 92691

A courtesy copy has also been mailed to Patent Owner's litigation counsel at:

Mark Fenster
Russ August & Kabat
12424 Wilshire Boulevard, 12th Floor
Los Angeles, CA 90025

Date: August 12, 2024

By: /Lisa C. Hines/
Lisa C. Hines
Case Manager
FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, LLP