

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.

IPR2024-01181
Patent 10,749,700

PATENT OWNER'S RESPONSE

TABLE OF CONTENTS

I.	Introduction	1
II.	Overview of the '700 Patent	3
III.	Level of Ordinary Skill in the Art.....	5
IV.	Claim Construction	6
V.	Overview of Relevant Prior Art.....	6
	A. Rao	7
	B. Scahill	10
VI.	The Petition fails to show unpatentability of claim 1.	12
	A. The Petition's theory that queuing network packets constitutes scheduling of application execution cannot succeed.	13
	B. The Petition does not motivate or provide a reasonable expectation of success for Rao's packet filter 322 to incorporate functionality from Scahill's application schedulers 24.	16
VII.	The Petition further fails to show unpatentability of claims 4-5.....	25
	A. The Petition's theory is based on an unsupported understanding of "network access non-controlled activity" as encompassing all foreground activities.	26
	B. The Petition sets forth no plausible theory showing how claims 4 or 5 are satisfied.	29
VIII.	The Petition further present a plausible allegation of invalidity regarding claim 2 according to the testimony of Petitioner's expert.	31
IX.	The Petition further fails to set forth evidence of unpatentability for claim 16.....	33
X.	Conclusion	37

PATENT OWNER'S EXHIBIT LIST

No.	Description
Ex. 2001	Declaration of Erik de la Iglesia in Support of Patent Owner's Response
Ex. 2002	<i>Curriculum Vitae</i> of Erik de la Iglesia
Ex. 2003	May 20, 2025 Deposition Transcript of Dr. Harry V. Bims

I. Introduction

All challenged claims require a “launch manager to schedule corresponding execution times for applications” as recited by claim limitation 1[d]. Ex. 1001 at 107:47-52. The Petition relies on a combination of two references, “Scahill” and “Rao,” for this limitation. Pet. at 27–28.

The Petition alleges that this limitation would be satisfied by modifying Rao’s packet filter 322 (which is responsible for queuing and prioritizing network packets) to incorporate the functionality of Scahill’s application schedulers 24 (which are responsible for coordinating with the device’s operating system to schedule application execution). However, the Petition contends that limitation 1[d] would be satisfied “assign[ing] execution times” to *network packets*¹ in the proposed combination, rather than assigning execution times for an *application*. Pet. 28 (alleging: “In Rao/Scahill’s architecture, ... the prioritized network communications would be assigned execution times by filter 322.”); Ex. 2003 (“Bims Dep. Tr.”) at 77:5-20 (acknowledging that in the proposed combination, the Petition’s allegation that “the prioritized network communications would be assigned execution times by

¹ The Petition’s reference to assigning “execution” time for packets does not make sense—presumably, the Petition meant to say something like “assign[ing] transmission times” for network packets, because unlike executable applications, packets are transmitted and received, rather than executed. See Ex. 2001, ¶45.

filter 322” refers to “the network packets are assigned execution times by filter 322”).

The Petition’s theory fails for numerous reasons, but most obviously because scheduling packet transmission does not equate to “schedul[ing] corresponding execution times for applications” as all challenged claims require. The Petition’s argument relies on the conflation of the prior art’s disclosure of queuing *network* packets with “queue[ing] applications.” See Pet. 27 (discussing Rao’s methods of queuing network packets, and describing these as “ways to queue applications”). But, of course, *packets* are not *applications*, and the Petition presents no plausible justification for its conflation of these two distinct concepts.

The Petition additionally fails for additional reasons as to claims 2, 4–5 and 16. Regarding claims 2 and 4–5, all recite “a network access non-controlled activity.” The Petition’s *only* rationale as to why this limitation is satisfied is based on an obvious misreading of the ’700 patent that “an application in the foreground is a ‘network access non-controlled activity.’” Pet. 31. The Petition’s allegation in this regard is inconsistent with even claim 5 of the ’700 patent, which makes clear that these are two distinct concepts by reciting that “determining that a given one of the applications is a network access non-controlled activity” is “*based on* that application currently providing a foreground service,” and does *not* recite that *all*

applications providing a foreground service are *by definition* network access non-controlled activities.

Finally, claim 16 recites that “the launch manager is integrated with an operating system broadcast intent manager.” The Petition’s sole rationale for why this would have been obvious is that, allegedly, “[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,” and “[i]n the case of Android OS, this would include the ‘broadcast intent manager.’” Pet. 42–43. The Petition, however, presents absolutely no evidence that the Android OS’s broadcast intent manager was, in fact, a component of the operating system that “schedules tasks and access to system resources,” and Petitioner’s expert had no knowledge of these details. Ex. 2003 at 84:8-19 (“Q. Do you know whether the broadcast intent manager is responsible for scheduling tasks [and] access to system resources? A. I have not looked into those details.”). Accordingly, the Petition fails to set forth any competent evidence supporting its purported obviousness theory for claim 16.

Accordingly, Patent Owner respectfully requests all challenged claims be found not unpatentable.

II. Overview of the ’700 Patent

The challenged claims of the ’700 Patent (claims 1–20) are directed at a high level to a device that is capable of associating “network access policy controls

with... [a] plurality of executable applications,” wherein the applications “request an execution time according to the corresponding network access policy controls,” and wherein the device includes “a launch manager to schedule corresponding execution times for [the requesting] applications” according to the policy controls. *See* Ex. 2001, ¶¶30–32.

Independent claim 1 (the sole independent claim of the ’700 patent) is reproduced below.

1[pre]. A wireless end-user device comprising:
1[a] at least one wireless modem to provide network access for the wireless end-user device;
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;
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
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.

Ex. 1001, claim 1.

In other words, the application requests execution time, and execution time is scheduled according to the policy. The '700 patent makes clear that there is a distinction between the requested and scheduled “execution of an application” on the one hand, and communication generated as a result of execution on the other hand. *See, e.g.*, Ex. 1001 at 75:4-7 (distinguishing between “an application executing on the device” and communications sent or received from the executing application); *id.* at 102:64-103:2 (“The application can be referred to as ‘running’ on the device or as being ‘executed’ by the device in accordance with known uses of those terms.”); *id.* at Claims 1, 11, 15, 17, 18, and 19 (referring to “executable applications”); *id.* at 35:42–51 (“In some embodiments, *at or after execution of an application that is determined to require network service usage....*”); Ex. 2001 ¶33.

III. Level of Ordinary Skill in the Art

The Petition proposes that a POSITA in the field 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.” Pet. 3. The Petition further states that “[m]ore related education would compensate for fewer years of work experience (and vice versa).”

Patent Owner believes that Petition's stated level of skill is too vague. For example, it is unclear why the Petition proposes "*three to five* years of experience," when three years of experience already qualifies a person as a POSITA under the Petition's definition. *See Ex.* And the Petition's proposal that "[m]ore related education would compensate for fewer years of work experience (and vice versa)" also lacks clarity. *See Nespresso USA, Inc. v. K-Fee Sys. GmbH*, IPR2022-01574, Paper No. 9 (PTAB Apr. 10, 2024) ("As for [the contention] that '[a]dditional education might substitute for some of the experience, and substantial experience might substitute for some of the educational background,' we find that this characterization is too vague.'").

For purposes of this response, Patent Owner agrees that a person having a bachelor's degree in computer science, electrical engineering, or a related field, plus at least three years of experience with networking, power consumption of networked computing devices, and/or wireless digital communication systems qualifies as a POSITA. The three years of additional experience can be in either a professional setting or a research setting (e.g., through a graduate degree). *See Ex. 2001 ¶¶27-29.*

IV. Claim Construction

The Petition identifies no terms for construction. Pet. 4. For purposes of this response, Patent Owner agrees that no formal claim construction is necessary.

V. Overview of Relevant Prior Art

A. Rao

At a high level, Rao relates to “prioritization of network communications.” See Ex. 1005 at Title. Rao prioritizes network communications by having a packet filter 322 “determine what action to take against packets,” including dropping those packets or placing them into a queue for transmission based on priority:

The filter 322 may use a filter table for filtering packets. The filtering table is used to determine what action to take against packets, such as packets intercepted by the packet capture mechanism 365. The filter 322 can inspect the contents of the packets, such as routing information, to determine the action to take based on the filtering table. In some embodiments, the filter 322 may drop or accept the network packets depending on their content. In other embodiments, the filter 322 may route the network packets to the agent 326 based on the packets content and/or the filtering table. The filter table may also be used to ensure that unwanted packets are discarded.

Ex. 1005, [0102]; *id.* at [0191] (“Based on the indicated priorities for applications 338 a-338 n, in some embodiments, *the filter 322 may place, arrange, or coordinate network packets into queues 540a-540n* in support of the priorities.”); *see also* Pet. 6-10 (the Petition’s overview of Rao, which explains that Rao’s prioritization policies are implemented by queuing network packets according to their priority); Ex. 2001, ¶¶35.

Rao's packet filter is shown, for example, in Rao's FIG. 1C., which indicates that Rao's filter 322 exists outside of application 338 (which is an application that requests transmission of packets which are ultimately filtered and prioritized by filter 322).

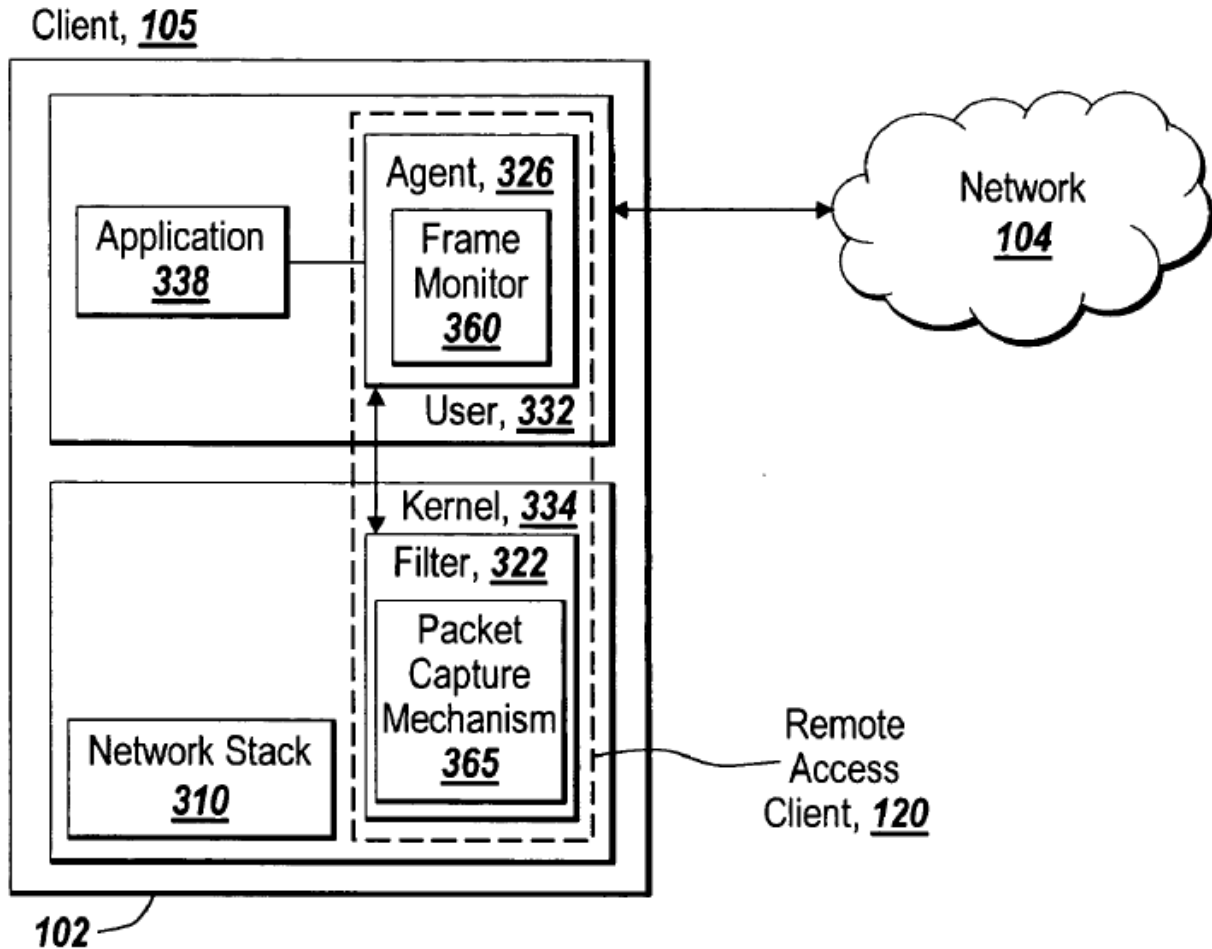


Fig. 1C

Ex. 1005, FIG. 1C; Ex. 2001, ¶36.

Rao also discloses, in Figure 5B (reproduced below) “a flow diagram depicting one embodiment of the steps performed to provide client-side application-aware prioritization.” Ex. 1005, [0071]; *see also* Ex. 2001 ¶¶37-38.

550

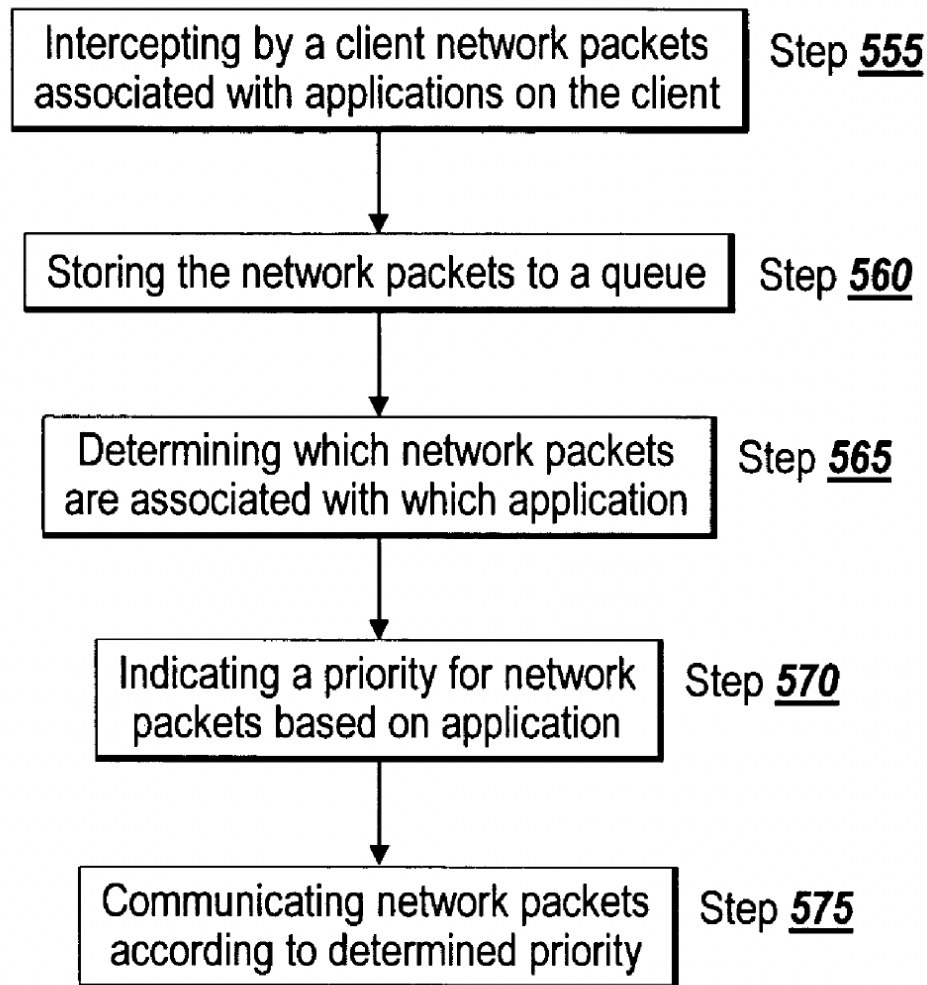


Fig. 5B

Ex. 1005, FIG. 5B. The Petition acknowledges that “Figure 5B illustrates Rao’s prioritization technique,” and that the end result of Rao’s prioritization technique is that “outbound network packets generated by an application... are prioritized by the client 105 prior to transmission based on the type and/or priority of the application 338a-338n” after being placed in appropriate queues 540a-540n Pet. 8-10 (quoting Ex. 1005, [0184], [0186]).

B. Scahill

Unlike Rao’s filter 322 which influences network communication by placing network packets into queues, Scahill’s application scheduler 24 assigns execution time to applications so that they can generate network communications in the first place. *See, e.g.*, Pet. 10 (“Scahill discloses a method of coordinating the execution time of a plurality of applications...”). This is accomplished, not through a queueing system, but by a logic structure in which task pre-requisites and dependencies are evaluated before allowing any given task to proceed, as illustrated in Scahill’s Figure 3 (reproduced below).

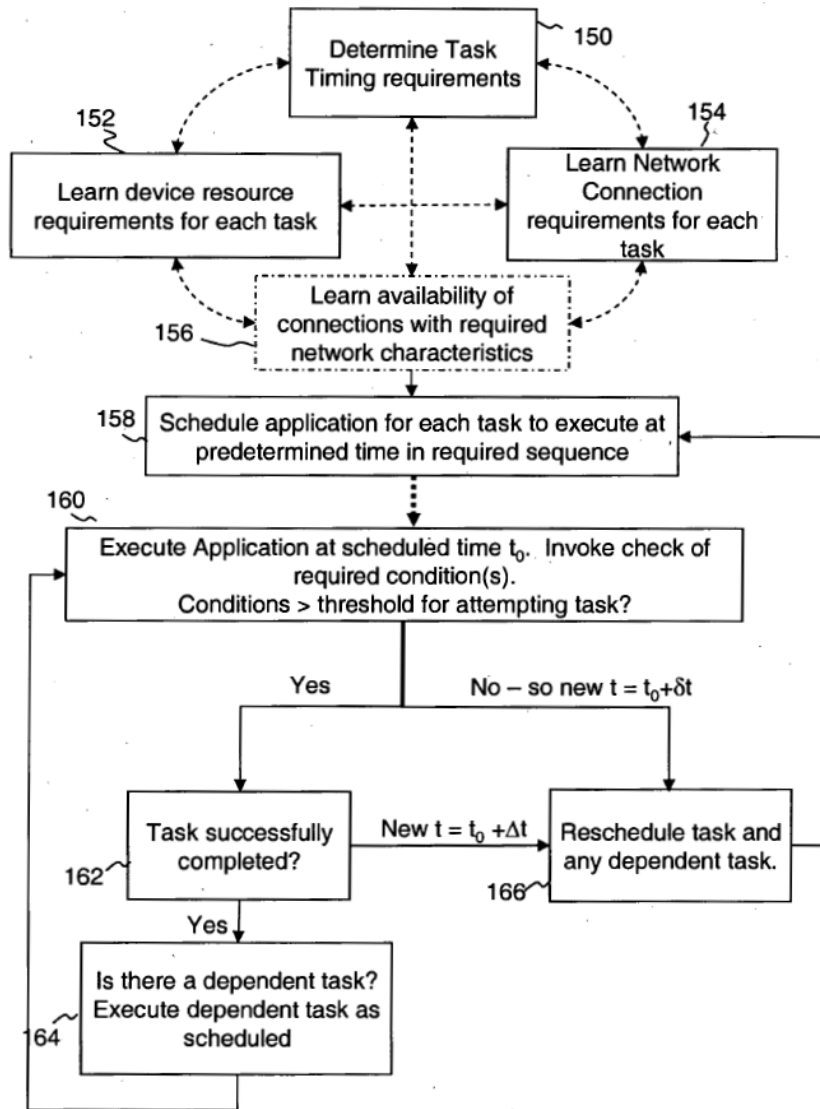


FIGURE 3

Ex. 1008, FIG. 3; Ex. 2001, ¶39. Scahill does not teach or suggest that its application scheduler would be used to schedule the timing of transmissions of any individual packet, as opposed to scheduling when a particular application task (which may ultimately generate packets) is executed. *See id.* ¶¶39-40.

Scahill further discloses that what it describes as “application scheduler 24” actually constitutes a plurality of application schedulers, one for each application

that requires scheduling. Ex. 1008, 17:14-17 (indicating that “[e]ach software client software library 14a,b comprises a plurality of library components, some specific to the respective client application 12a,12b); FIG. 1B (indicating that application scheduler 24 is one of these “[a]pplication specific library component[s]”); Ex. 2003 at 73:14-75:10 (Dr. Bims testifying that “application scheduler 24 is actually an application-specific application scheduler such as 24a corresponding to application 12a,” and “that would also be true for application 12b,” and “same thing for application 12c[,] [i]t would have an application scheduler 24c”); Ex. 2001 ¶41.

VI. The Petition fails to show unpatentability of claim 1.

Claim 1 of the ’700 Patent relates specifically to scheduling *application* execution based on application-specific policies. Specifically, all challenged claims require a “launch manager to schedule corresponding execution times[] for applications requesting execution time” as recited by claim limitation 1[d].

As explained in further detail below, the Petition’s proposal for how its proposed combination satisfies this limitation is deficient, because the Petition conflates queuing network packets with scheduling application execution. And at an even higher level, the Petition fails to show the obviousness of its proposed combination or Rao and Scahill, in which the functionality associated with a plurality of application schedulers would be implemented within a single packet filter.

A. The Petition’s theory that queuing network packets constitutes scheduling of application execution cannot succeed.

All challenged claims require a “launch manager to schedule corresponding execution times, for the applications requesting execution time” as recited by claim limitation 1[d]. Ex. 1001 at 107:47-52.

The Petition alleges that Rao’s packet filter 322 would be the “launch manager” in its proposed combination. Pet. 27. And the Petition further proposes that Rao’s filter 322 would somehow be modified to incorporate the functionality of Scahill’s application schedulers 24. *See generally* Pet. 27-28 (not specifying precisely what functionality of Scahill’s application schedulers 24 would be included in Rao’s filter 322); *see also generally* Pet. 13-19 (Petition’s rationale to combine section, providing only vague allegations about how Scahill’s application scheduler functionality would be combined with Rao’s system, but providing few details about how the resulting system would operate in practice).

Other than two entirely conclusory sentences,² the Petition’s only allegation as to how limitation 1[d] is satisfied by the Petition’s proposed combination is that

² *See* Pet. 27 (alleging: “Rao/Scahill renders obvious element 1[d]”); Pet. 28 (alleging that in the proposed combination, filter 322 “selects corresponding execution times for the applications requesting execution time in accordance with respective network access policy controls for each application,” a near-verbatim recitation of limitation 1[d]).

Rao's filter 322 as modified would "assign[] execution times" to "*network communications*" in the proposed combination. Pet. 28 (alleging: "In Rao/Scahill's architecture, ... the prioritized network communications would be assigned execution times by filter 322."). And the only "network communications" that Rao teaches assigning transmission times to are *network packets* (Ex. 2001 ¶44), and Petitioner's expert confirmed that under his combination theory, the purported "execution times" that are assigned by filter 322 would be assigned to "network packets" as opposed to the any application activity that might generate those packets. Ex. 2003 at 77:5-20 (acknowledging that in the proposed combination, the Petition's allegation that "the prioritized network communications would be assigned execution times by filter 322" refers to "the network packets are assigned execution times by filter 322"). In other words, the Petition's theory rests on the premise that "assign[ing] execution [i.e., transmission]³ times" to *network packets* constitutes "selecting corresponding execution times for applications."

³ Assigning "execution times" for packets (as Dr. Bims confirmed the Petition's theory entails) does not make sense, because unlike executable applications, packets are not "executed." Ex. 2001, ¶45. Instead, packets are transmitted and received, and Rao's filter 322 operates such that it prioritizes and schedules transmission of outgoing network packets. *Id.* ("To the extent Rao's filter 322 assigns particular times to packets at all, it would be transmission times being assigned.").

However, as is known in the art, there is a clear distinction between *packets* and *application execution*. Ex. 2001, ¶¶46-47; *see also id.* at ¶¶52-54 (discussing differences between application scheduling and packet prioritization / filtering). For example, Rao itself makes clear that *packets* are “generated or received for an *application*.” *See, e.g.,* Ex. 1005, [0003] (discussing “network packets *for these applications*” and the order in which the applications are processed); *id.* at [0024] & [0027] (“[T]he present invention relates to a method for transmitting *packets from an application* using an unreliable transport protocol”). Thus, there is a clear distinction between *application execution* (which may *cause* packets to be generated), and the resulting *packets* (which are the potential *result of* application execution).

Contrary to this straightforward understanding that packets are generated or received *for* or on behalf of applications, the Petition’s analysis assumes without any supporting evidence or logic that Rao’s queue of “network packets” is a “way[] to queue applications.” *See* Pet. 28 (quoting Rao’s teaching of queuing network packets, then alleging that “Rao provides additional examples of the many potential ways to queue applications,” and then providing another discussion in Rao of “packets [being] placed or arranged in the queue”).

In other words, the Petition asserts that Rao’s queueing of *packets* constitutes a queuing of *applications*. The Petition’s theory, wherein queuing *packets*

constitutes queuing *application execution*, ignores the fundamental distinction between *packets* on the one hand, and *application execution* on the other hand. Ex. 2001 ¶¶47-48.

The Petition’s theory fails, because scheduling transmission of individual packets does not constitute “selecting corresponding execution times, for the applications requesting execution time.” Ex. 2001 ¶49. As noted above, packets are potentially generated *as a result of* application execution, such that scheduling the transmission of already-generated packets does not constitute assigning an execution time for an application as the claims require. Indeed, as explained by Mr. de la Iglesia, the fact that there are packets to queue in the first place *assumes* that whatever task that generated those applications has *already* been executing, such that queuing of the relevant packets could not constitute scheduling of the process that generated those packets. Ex. 2001, ¶49.

B. The Petition does not motivate or provide a reasonable expectation of success for Rao’s packet filter 322 to incorporate functionality from Scahill’s application schedulers 24.

As discussed above, the Petition’s mapping of its Rao-Scahill combination to claim 1 of the ’700 Patent is clear that Petitioner believes and contends that scheduling transmission of *network packets* is the same as scheduling *application execution*. Because there is no support for that understanding, Patent Owner can

only assume that in Petitioner's Reply, Petitioner will attempt to introduce a new and different theory.

While Patent Owner does not know what theory Petitioner might allege (and believes that any new theory would be improper under the Board's regulations and practices), Patent Owner does note that at a high level, implementations of Scahill's application scheduling functionality into Rao's filter 322 would not be motivated or be reasonably expected to succeed.

As an initial matter, it would not make any sense to implement an application scheduler into a packet filter. These two components operate at entirely different timescales—packets would have been transmitted on the order of hundreds of times per second using the technologies available in the relevant timeframe, whereas application scheduling would occur at human-observable timescales. *See, e.g.,* Ex. 1008, 4:2-6 (discussing “several data transmissions to be performed within a 24 hour period” that need to be scheduled). In other words, packet filters make timing decisions that would typically be on the order of thousands of times faster than the timing decisions that application schedulers make, such that the design and timing considerations of these two different components are very different, and result in very different design considerations. *See* Ex. 2001 ¶¶51-52 (providing additional details regarding the differences between how packet filters and application schedulers operate).

And as explained by Mr. de la Iglesia, the vast majority of applications are not even *aware* of what packets are generated on its behalf. Ex. 2001 ¶53; *see also* Ex. 1001, 10:16-19 (distinguishing between the packet level and application level). Applications in systems such as Rao and Scahill simply request that data be transmitted (e.g., after requesting that a socket be opened), and lower-level system elements (such as the operating system) are responsible for generating, managing, and transmitting packets in order to effectuate the transmission of the data that the application has requested be transmitted. Accordingly, applications in the context of Rao and Scahill would not request transmission times for packets, as the applications would not even be aware of the packets that would be transmitted.

In light of these significant differences between scheduling *packets* and scheduling *application execution*, a POSITA would not have been motivated to modify a packet filter to incorporate application scheduling functionality. Ex. 2001, ¶54.

The Petition's sole motivation to incorporate *application scheduler* functionality into Rao's *packet filter* 322 appears to be based on a significant misunderstanding of what Rao's packet filter 322 actually does. As previously noted, the Petition is explicit that it believes that Rao's filter 322 "queue[s] applications," even though Petitioner's expert was not able to identify *any* disclosure in Rao of using its prioritization policies "to influence system operation other than by

providing the result of prioritizing, ignoring, or not processing network packets.” Ex. 2003 at 47:20-48:17; *see also id.* at 19:17-49:10 (lengthy line of questions where Dr. Bims was unable to identify any effect of Rao’s policies other than to ultimately queue or not process network packets). But with the *correct* understanding of Rao’s packet filter 322 (wherein filter 322 queues *packets* that may have been generated by applications, rather than actually queuing *applications* or application execution), there is no record motivation to incorporate application scheduler functionality into a packet filter as the Petition proposes. Ex. 2001, ¶¶55-56.

The lack of evidence regarding motivation to make the Petition’s proposed combination is underscored by the fact that the Petition identifies no evidence that anyone has *ever* attempted to implement application scheduling functionality within a packet filter. And as Mr. de la Iglesia explains, he is *also* not aware of anyone having done so. Ex. 2001, ¶57. Thus, because there is no evidence of packet filters incorporating application scheduler functionality (and because, as discussed above, packet filter and application scheduling functionality are significantly different), this is *not* as the Petition contends “nothing ‘more than the predictable use of prior art elements according to their established functions.’” *See* Pet. 17; Ex. 2001, ¶57.

And the lack of identifiable motivation becomes even more problematic for the Petition’s theory, in view of the specific teachings of regarding Rao’s filter 322 and Scahill’s application schedulers 24. *See* Ex. 2001, ¶¶58-65. For example, Rao’s

filter 322 prioritizes packet transmission by placing packets into queues, which are linear data structures that place certain packets in front of others. Ex. 2001, ¶58. In contrast, Scahill's application schedulers 24 do *not* use queues because they must account for dependencies between tasks. *Id.*, ¶¶58-61.

In the context of Scahill, "a dependent task is a task whose execution by a client application 12a...c is dependent on the successful completion of another task by the same or a different client application 12a...c." Ex. 1008 at 20:18-20. An example dependency might be that the activity of "uploading an activities journal for various files" must be performed before another activity of "downloading all time-charged activities" can be performed. *See* Ex. 1008 at 21:25-28. Scahill teaches that "[i]n the event that a task has a task dependency, ... the relevant application scheduler 24b,c will reschedule the execution time for [the] dependent task whenever the first task is rescheduled." *Id.* at 20:20-23.

An illustration of the logic flow for Scahill's application scheduling is shown in Scahill's FIG. 3, reproduced below.

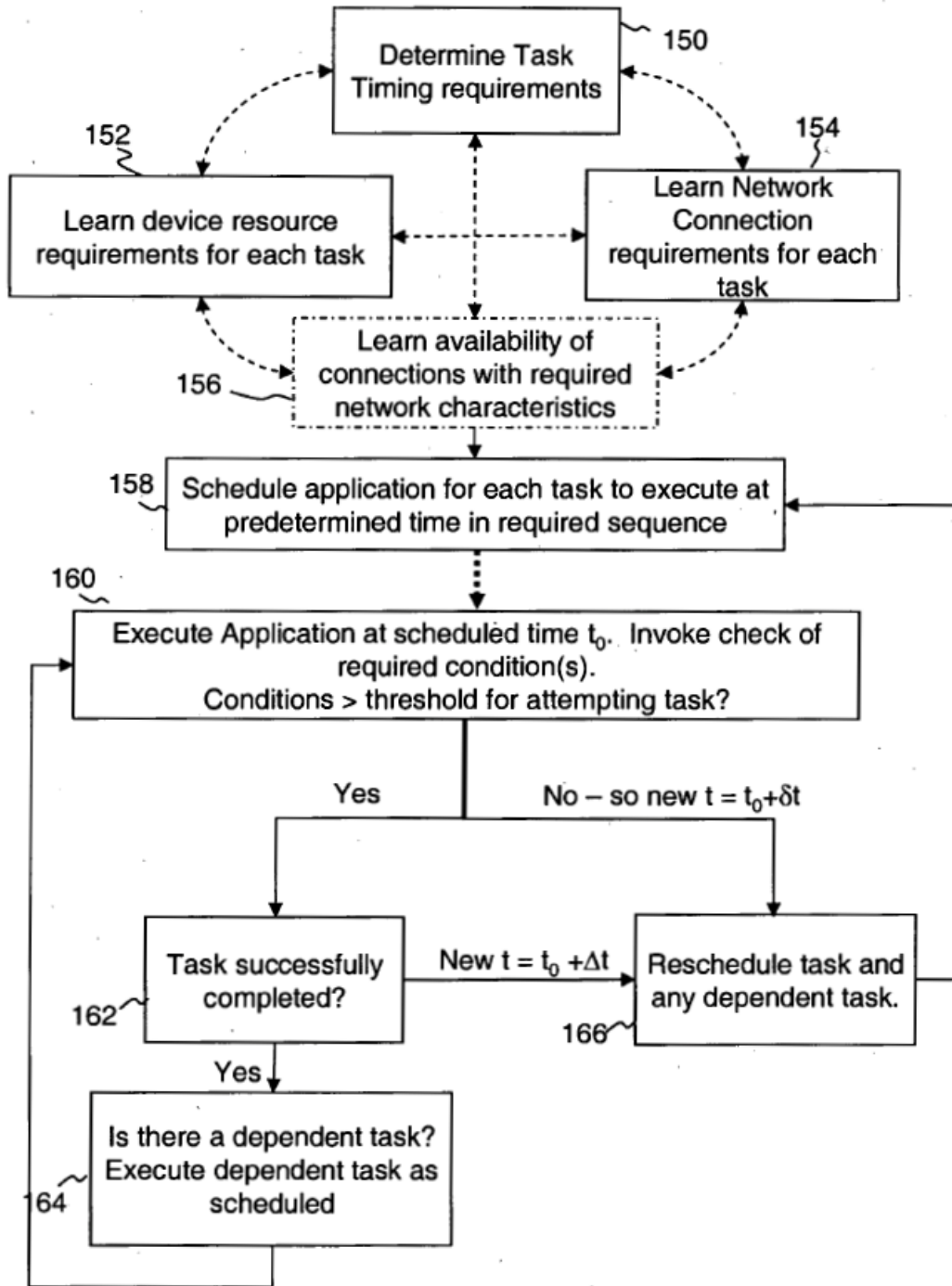


FIGURE 3

Ex. 1008, FIG. 3.

As shown above, even *after* an application's task has been scheduled for a particular time (in step 158), the system must "[i]nvoke [a] check of required condition(s)" such as whether the required dependencies have been satisfied. Ex. 2001, ¶61. There is no indication that this complex logic would be (or could be) implemented in a queuing structure, much less by the type of queuing structure controlled by Rao's *packet* filter, and the Petition provides no proposed modification to either filter 322 or application schedulers 24 to reconcile this discrepancy. *See id.*

Finally, the Petition's proposed moving of functionality from Scahill's application schedulers 24 into Rao's filter 322 is contrary to Scahill's teaching that each application is assigned its *own* application scheduler that is "specific to the respective client application 12a,12b." Ex. 1008, 17:14-17; Ex. 1008 at 32-34 (discussing how tasks "are scheduled... by the respective application schedulers 24a,b,c of each client application 12a...c"); Ex. 2003 at 73:14-75:10 (Dr. Bims testifying that "application scheduler 24 is actually an application-specific application scheduler such as 24a corresponding to application 12a," and "that would also be true for application 12b," and "same thing for application 12c[,] [i]t would have an application scheduler 24c"); Ex. 2001, ¶¶62-63.

Unlike in Scahill, where each application has its own application scheduler that operates within the application, the Petition proposes that the application scheduler functionality be taken *outside* of the applications and placed within a

single filter 322 which “is separate from the applications.” Ex. 2003, 67:15-19 (Dr. Bims testifying that in his proposed implementation, “the filter resides in the remote access client, which is separate from the applications”); Ex. 2001, ¶63.

The Petition provides no motivation whatsoever to modify Scahill’s application schedulers (which Scahill teaches are specific to each application and operate within the applications themselves), such that there would only be a *single* component (filter 322), located *outside* of the applications, that incorporates that functionality. *See generally* Petition (particularly at 27-28) (not even addressing this deviation from Scahill’s teachings, much less providing any motivation for deviating in the proposed manner); *see* Ex. 2001, ¶64. Nor does the Petition provide any factual basis regarding why a POSITA would have even allegedly expected to succeed in making this modification. *See* Ex. 2001, ¶64.

Far from being obvious, a POSITA would have avoided making this proposed modification to Scahill’s application schedulers due to the teachings of Scahill, because a POSITA would have recognized that “[t]here are a number of *application specific requirement(s)* which the application scheduler library component 14a,b of each client application 12a,b... must take into account” in performing its functionality. Ex. 1008, 24:14-17. Because requirements are specific to each individual application, a POSITA would understand it to be far more sensible to leave application scheduling functionality within the application-specific scheduling

components 24a, 24b, etc., rather than implementing them in a single component (such as filter 322) that is external to the relevant applications. Ex. 2001, ¶65

Accordingly, there are numerous reasons why a POSITA would *not* have found it obvious to modify the prior art in the manner the Petition alleges.

Regarding reasonable expectation of success, the Petition makes no showing how a POSITA, would reasonably expect success in implementing the proposed combination in view of the significant differences between the individual references and how the Petition’s description of their combination. Neither the Petition nor Dr. Bims has actually explained specifically *how* the necessary modifications would be made in practice, so Patent Owner cannot address any specific implementation details. But as explained by Mr. de la Iglesia, the differences are significant enough that a POSITA (which can be satisfied by a bachelor’s degree plus three years of experience under Petitioner’s proposal) would not have reasonably expected to succeed in implementing the functionality of Scahill’s application schedulers 24 into Rao’s packet filter 322, because given the significant differences in how those components operate, the proposed modification would have taken (at least) an extraordinary level of skill to accomplish in a functioning manner. *See* Ex. 2001, ¶66

* * *

In sum, the Petition’s mapping of Rao-Scahill to limitation 1[d] fails because queuing of network packets does not satisfy the claim requirement of “selecting

corresponding execution times, for the applications requesting execution time.” And to the extent Petitioner proposes some different claim mapping or theory in reply, even the high-level concept of implementing Scahill’s application scheduler functionality into Rao’s packet filter would not have been obvious, because the Petition identifies no motivation to do so, and because doing so would be inconsistent with the teachings of both Rao and Scahill. *See* Ex. 2001 ¶¶42-66.⁴

VII. The Petition further fails to show unpatentability of claims 4-5.

Claims 4 and 5 both require “determining that a given one of the applications is a network access non-controlled activity” based on two different criteria. Both claims 4 and 5 depend from dependent claim 2. For context, claim 2 recites:

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.⁵

Ex. 1001, claim 2.

⁴ Patent Owner provides expert testimony only with regard to claim 1, because the Petition’s failures with respect to claims 2, 4, 5, and 16 do not require resolution of any disputed technical issues; instead, the Petition’s theories with respect to those claims are apparent in view of the Petition itself and Dr. Bims’ admissions.

⁵ The Petition’s failure with respect to claim 2 is addressed separately in Section VIII, below.

Claim 4 additionally requires that “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.*” *Id.*, claim 4. And claim 5 recites “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.*” *Id.*, claim 5. In short, both claims 4 and 5 require an activity classifier that determines whether an application “is a network access non-controlled activity” based on specific inputs to the determination.

As discussed below, the Petition’s theories for both claims 4 and 5 of these claims rely on the incorrect assumption that all foreground activities in the prior art are “network access non-controlled activit[ies],” such that the Petition sets forth no plausible theory as to how the requirements of claims 4 and 5 are satisfied.

A. The Petition’s theory is based on an unsupported understanding of “network access non-controlled activity” as encompassing all foreground activities.

For both claims 4 and 5, the Petition refers back to its analysis for claim 2, in which the Petition asserts that “an application... running in the... foreground” is a “network access non-controlled activity.” *See* Pet. 31 (the Petition’s claim 4 analysis citing the Petition’s claim 2 analysis, and claim 5 analysis citing claim 4 analysis); Pet. 29 & n.2 (Petition’s claim 2 analysis). The entirety of applications’ allegation

that applications running in the foreground are network “access non-controlled activities” is reproduced below:

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”). (Bims, ¶89.) [A footnote adds: “This is consistent with the ’700 patent’s specification. (Bims, ¶89; ’700 patent, 64:13-21, 108:1-4.)”]

Pet. 29 & n.2.

The cited portions of the ’700 patent, however, do not suggest that prior art foreground activities are “network access non-controlled activities.” For instance, the first of the Petition’s citations is reproduced below.

In some embodiments, classifying the network service usage activity includes classifying the network service usage activity into one or more of a plurality of classification categories for differential network access control for protecting network capacity (e.g., one or more network capacity controlled service classifications and/or a priority state classification, such as a background services classification and/or a background priority state classification).

Ex. 1001, 64:13-21. As shown above, the cited portion indicates that a classification system *can* be established such that “a background services classification” can be one example of a “classification categor[y] for protecting network capacity”; it says

nothing about how any *prior art* disclosure of a background service would necessarily constitute a “category[y] for protecting network capacity,” and more importantly discloses nothing of a “network access non-controlled activity.”

Likewise, the second of the Petition’s citations (which is actually to claim 5 of the ’700 patent) is reproduced below:

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.

Ex. 1001, 108:1-4 (claim 5).

This portion of the ’700 patent (which explicitly defines a *claimed invention* of the ’700 patent) does nothing to indicate that a prior art disclosure of a determination that an application is running in the foreground constitutes a disclosure that the application “is a network access non-controlled activity.” Instead, it states that *the invention of claim 5 determines* whether an application is a “network access non-controlled activity *based on*” whether that application is “currently providing a foreground service.” In other words, *even in the claimed invention*, an application providing a foreground service is not necessarily determinative of that application being “a network access non-controlled activity,” but rather the determination need only be *based on* the provision of a foreground

service. Thus, even claim 5 itself draws a distinction between a “foreground service” on one hand, and a “network access non-controlled activity” on the other.

As explained below, without the (incorrect) understanding that all applications running in the foreground are “network access non-controlled activities,” the Petition provides no plausible theory for how either claims 4 or 5 are satisfied.

B. The Petition sets forth no plausible theory showing how claims 4 or 5 are satisfied.

Claim 4 recites, in part, an “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.”

The Petition’s sole argument for how this limitation is satisfied is an assertion that the Rao/Scahill combination “classifies an application as ‘running in the foreground’ because of a user’s current interaction with that application.” Pet. 30-31. However, this argument relies on a conflation of a prior art determination that an application running in the foreground in the prior art, with the determination that the application is a network access non-controlled activity. As noted in the preceding subsection, there is no support for conflating these concepts in the manner the Petition proposes.

Accordingly, the Petition presents no plausible theory regarding how claim 4 is satisfied. At most, the Petition has alleged that Rao or Scahill disclose determining whether an application is *running in the foreground* based on a user's current interaction with the application, which is not what claim 4 recites.

For instance, Rao discloses that “filter 322 may be used... to prevent unauthorized access from remote computers by discarding packets to specified destination addresses.” Ex. 1005, [0102]. If a foreground activity only sent packets to one of these specified destination addresses such that all of its packets would be discarded, that activity would clearly be “blocked” and thus not “network access non-controlled.” *See, e.g.*, Ex. 1001 at claim 2 (drawing a distinction between blocked activities and network access non-controlled activities). Accordingly, in the context of the prior art, an application providing a foreground service cannot be sufficient to determine that the application is a “network access non-controlled activity.”

Regarding claim 5, which recites an “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,” the Petition presents only a single conclusory sentence that this limitation is disclosed. Pet. 31. And the Petition's citations to evidence cite to Rao's discussion of “Background Information” (Ex. 1005, [0003] (cited by Pet. 31)), which describes only prior art

that Rao disfavors, and Scahill’s discussion of foreground applications vs background applications (Ex. 1008 at 36:4-8 (cited by Pet. 31)), which provides no disclosure of any applications being network access non-controlled on the basis of an application providing a foreground service.

Accordingly, the Petition presents no plausible theory as to how its proposed combination of Rao and Scahill satisfies either claim 4 or 5.

VIII. The Petition further present a plausible allegation of invalidity

regarding claim 2 according to the testimony of Petitioner’s expert.

Claim 2 recites: “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.”

As noted above in Section VII, the Petition fails to set forth a plausible theory regarding how the “network access non-controlled activities” limitation is disclosed in the prior art. This flaw applies equally to claim 2.

Additionally, claim 2 recites not only “network access non-controlled activities,” but *also* “blocked activities” *and* “network access controlled activities” as three separate types of activities. According to Dr. Bims, however, as long as *any* activity on the system is blocked from network access, then all other activities that

are *not* blocked from network access are “network access non-controlled activit[ies]”:

Q. Okay. So is it your opinion that if an application gets a higher priority than another application, the first application is a network access non-controlled activity?

A. Yes.

Q. Okay. So is it your opinion that if the second

application in this example is effectively deprioritized entirely or blocked and the first application is allowed to transmit in some way, that the first application would be a network access non-controlled activity?

A. Yes, because the concept of priority is a relative concept. In other words, for the -- for an application to be prioritized, it has to be at the expense of another application that is deprioritized. So the first application which is given the higher priority relative to the second application would thus be a network access non-controlled activity application.

Ex. 2003, 80:17-81:11 (annotated).

To the extent that Dr. Bims’ understanding is accepted as correct (a position with which Patent Owner disagrees, at least because it is nonsensical in the context of claim 2), then there could not be three separate classifications of activities that include both “blocked activities,” “non-network controlled activities,” and another class of activities (here, “network controlled activities”). Such a classification would be impossible because “blocked activities” are obviously blocked. But then *all other* activities that are *not* blocked would be “network access *non*-controlled activities.” Ex. 2003 at 80:22-81:11 (acknowledging that under his interpretation, if a second application is “deprioritized entirely or blocked and the first application is allowed to transmit in some way, [then] the first application would be a network access *non*-controlled activity”). Under Dr. Bim’s logic, assuming the presence of a blocked activity in the first place, *all* activities on a system would fall into one of two categories: blocked activities or network access non-controlled activities. This leaves no room for the third recited category of claim 2 of “network access *controlled* activities.” Thus, to the extent Dr. Bims’ claim interpretation is given credence, claim 2 cannot be satisfied for that additional reason.⁶

IX. The Petition further fails to set forth evidence of unpatentability for claim 16.

⁶ To the extent the requirements of claim 2 are not satisfied, that is an additional reason why claims 4 and 5 (which depend from claim 2) are not satisfied.

Claim 16 depends from claim 1, and recites:

The wireless end-user device of claim 1, wherein the launch manager is integrated with an operating system broadcast intent manager.

Ex. 1001, claim 16. The Petition alleges that it would have been obvious to implement Rao's filter 322 (i.e., the alleged "launch manager") into the "broadcast intent manager" for the Android OS. Pet. 42-43.

The Petition's only rationale, however, as to why a POSITA would have implemented filter 322 into the "broadcast intent manager" (as opposed to any other component of, e.g., the Android OS) is the assertion that "Rao's filter would be implemented *into the component of the operating system that schedules tasks and access to system resources*," which the Petition alleges "would include 'the broadcast intent manager.'" Pet. 42-43.

In other words, the Petition's logic regarding claim 16 is structured as follows:

- **Premise #1:** A POSITA would have been motivated to incorporate Rao's filter into the OS component that schedules tasks and access to system resources.
- **Premise #2:** In Android OS, the component of the OS that schedules tasks and access to system resources includes the "broadcast intent manager."
- **Conclusion:** A POSITA would have been motivated to incorporate Rao's filter into Android OS's "broadcast intent manager."

See Pet. 42-43. As a simple matter of logic, if *either* Premise #1 or #2 is unsupported, the Petition's theory cannot succeed.

The Petition fails to set forth a prima facie case of invalidity because *the Petition provides no plausible support for Premise #2*. Specifically, the Petition's only cited support for the allegation that the component of Android OS that schedules tasks and access to system resources includes the "broadcast intent manager" is a citation to the '700 patent itself. See Pet. 43 (citing Ex. 1001, 67:37-41). The cited portion of the '700 patent is reproduced in its entirety below.

For example, the intercept and policy implementation functions can be designed into the activity manager, broadcast intent manger, media service manager, service manager, or other application or service activity management function in the Android OS.

Ex. 1001, 67:37-41.

As can be clearly seen above, this portion of the '700 patent does *not* state that the broadcast intent manager is involved in "schedul[ing] tasks and access to system resources." *Id.* And while the '700 patent *does* teach that policy implementation functionality can be designed into the broadcast intent manager, it is, of course, fundamentally improper to use the invention *as disclosed in the '700 patent itself* as a roadmap for obviousness. Instead, the Petition must show how the *prior art* discloses or renders obvious the relevant limitations.

The Petition's reliance on Dr. Bims' declaration fares no better. Specifically, although Dr. Bim's declaration largely parrots the Petition's conclusory allegations in relevant respect, Dr. Bims admitted at his deposition that he had not reviewed any documentation for the Android OS, and he admitted that he "ha[d] not looked into th[e] details" of "whether the broadcast intent manager is responsible for scheduling tasks [and] access to system resources":

Q. Okay. And in preparing your declaration, do you recall reviewing any documentation of the Android operating system?

A. No.

...

Q. Okay. And do you know what the broadcast intent manager of the Android operating system is?

A. I'm aware of the broadcast intent manager of the Android operating system, but not familiar with the details of its implementation.

Q. Okay. And is that also a component of the Android operating system?

A. Yes, as described in the '700 patent.

Q. Okay. Do you know whether the broadcast intent manager is responsible for scheduling tasks in access to system resources?

A. I have not looked into those details.

Ex. 2003, 14:1-4; 84:8-19 (annotated).

Accordingly, because Dr. Bims did not disclose any reasoned basis supporting his declaration testimony (and in fact admitted that he *did not know* whether the broadcast intent manager was responsible for scheduling tasks and access to system resources), his declaration testimony to that effect should be disregarded as unreliable.

Accordingly, the Petition presents no competent evidence supporting its invalidity theory for claim 16. Claim 16 should thus be found not unpatentable for this additional reason.

X. Conclusion

For the foregoing reasons, Patent Owner respectfully requests that all challenged claims be found not unpatentable.

Date: June 3, 2025

Respectfully submitted,

/James A. Milkey/

Reza Mirzaie, Reg. No. 69,138

Marc A. Fenster, *pro hac vice* forthcoming

Dale Chang, Reg. No. 61,476

Neil A. Rubin, Reg. No. 67,030

Philip X. Wang, Reg. No. 74,621

James A. Milkey, Reg. No. 79,503

RUSS, AUGUST & KABAT

12424 Wilshire Blvd., 12th Floor

Los Angeles, CA 90025

Counsel for Patent Owner

CERTIFICATION REGARDING WORD COUNT

Pursuant to 37 C.F.R. §42.24(d), Patent Owner certifies that there are 6,865 words in the paper excluding the portions exempted under 37 C.F.R. §42.24(a)(1).

Date: June 3, 2025

/James A. Milkey/

Reza Mirzaie, Reg. No. 69,138

Marc A. Fenster, *pro hac vice* forthcoming

Dale Chang, Reg. No. 61,476

Neil A. Rubin, Reg. No. 67,030

Philip X. Wang, Reg. No. 74,621

James A. Milkey, Reg. No. 79,503

RUSS, AUGUST & KABAT

12424 Wilshire Blvd., 12th Floor

Los Angeles, CA 90025

Counsel for Patent Owner

CERTIFICATE OF SERVICE (37 C.F.R. § 42.6(e)(1))

The undersigned hereby certifies that the above document was served on June 3, 2025, by filing this document through the Patent Trial and Appeal Case Tracking System as well as delivering a copy via electronic mail upon the following attorneys of record for the Petitioner:

Joshua L. Goldberg (Reg. No. 59,369)
Benjamin A. Saidman (Reg. No. 69,325)
Daniel C. Cooley (Reg. No. 59,639)
Grace K. Mills (Reg. No. 66,946)
joshua.goldberg@finnegan.com
benjamin.saidman@finnegan.com
daniel.cooley@finnegan.com
gracie.mills@finnegan.com
Motorola-Headwater-IPRs@finnegan.com

Date: June 3, 2025

/James A. Milkey/
Reza Mirzaie, Reg. No. 69,138
Marc A. Fenster, *pro hac vice* forthcoming
Dale Chang, Reg. No. 61,476
Neil A. Rubin, Reg. No. 67,030
Philip X. Wang, Reg. No. 74,621
James A. Milkey, Reg. No. 79,503
RUSS, AUGUST & KABAT
12424 Wilshire Blvd., 12th Floor
Los Angeles, CA 90025

Counsel for Patent Owner