

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SAMSUNG ELECTRONICS CO. LTD.
Petitioner,

v.

HEADWATER RESEARCH LLC,
Patent Owner.

IPR2024-01396
Patent 9,647,918 B2

**DECLARATION OF ERIK DE LA IGLESIA IN SUPPORT OF PATENT
OWNER'S PRELIMINARY RESPONSE TO THE PETITION**

I declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true.

Executed on January 9, 2024, at Marshall, Texas.



Erik de la Iglesia

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	BACKGROUND AND QUALIFICATIONS	2
III.	LEGAL PRINCIPLES	6
A.	Claim Construction	6
B.	Burden of Proof	6
C.	Anticipation	6
D.	Obviousness	7
IV.	OVERVIEW OF THE '918 PATENT.....	8
V.	PERSON OF ORDINARY SKILL IN THE ART.....	14
VI.	DR. TRAYNOR’S CHALLENGE TO CLAIM 1 IS INCONSISTENT WITH PLAIN MEANING.....	15

I, Erik de la Iglesia, hereby declare as follows:

I. Introduction

1. I am over the age of eighteen (18) years and otherwise competent to make this declaration.

2. I have been retained as an expert witness on behalf of Patent Owner for the above-captioned *inter partes* review (“IPR”). I understand that the petition for *inter partes* review involves U.S. Patent No. 9,647,918 (“the ’918 patent”).

3. I make this declaration based on my personal knowledge, educational background and training, consideration of the materials I discuss herein, and my expert opinions.

4. My work on this matter is being billed at my normal hourly consulting rate of \$750, with reimbursement for actual expenses. My compensation is not related to the outcome of any proceeding involving the ’918 patent. I have no personal interest in the outcome of the case.

5. In preparing this Declaration, I have reviewed and considered the ’918 patent, the ’918 patent’s prosecution history, the Petition and prior art references submitted with the Petition, the declaration submitted by Dr. Traynor in this proceeding, and each document cited in my declaration.

II. Background and Qualifications

6. My qualifications for forming the opinions given in this declaration are summarized here and are addressed more fully in my curriculum vitae, which is submitted as Exhibit 2002.

7. I hold a Bachelor of Science degree in Electrical Engineering from the University of Florida, and a Master of Science degree in Electrical Engineering from Stanford University. While at Stanford, I was an NSF Graduate Research Fellow. My graduate focus included VLSI CMOS architecture, high-speed circuit design, computer networks and protocols, and artificial intelligence. I have worked in several computer and electrical engineering related fields for over 25 years resulting in 68 issued U.S. patents that have been cited over 2850 times in USPTO patent applications. Since 2003, I have served in a number of chief architect and founder roles of companies that have been acquired or gone public. My experiences include the design or implementation of networks and network-based computation and protocol systems similar to the technology of the '918 patent, including systems for content targeting based on user behavior across devices.

8. Between December 1997 and November 1999, I was a circuit design engineer at Intel Corporation within the Mobile and Handheld Products Group. My responsibilities included working on several aspects of mobile processors and chipsets including clock and power distribution, dynamic clock frequency

selection, and the circuit design, device simulation, and validation of such features. Until March of 2000, I worked on similar functionality within the Microprocessor Products Group for the Itanium 3 and 4 products.

9. My work in power distribution included thermal modeling and predictive control of silicon devices and their packaging such as multi-tiered cooling systems for embedded, mobile, and desktop environments. Such systems included multiple tiers of thermal sensors, active and passive thermal control systems, and the feedback and analytic firmware and software control. My models were part of the development of the Thermal Design Power (TDP) metrics used by OEM and system designers to plan thermal control solutions for consumer and industrial products.

10. Between March 2000 and September 2003, I worked as a logic designer and architect for WebStacks and, through acquisition, Extreme Networks. WebStacks built a fully hardware-based TCP and HTTP processing stack for network processing functions including load balancing, proxy processing, and content rewriting. I was responsible for the architecture and design of the HTTP processing engine including header extraction and processing.

11. Between August 2003 and August 2007, I worked as Founder, Chief Architect and Director of Engineering for Reconnex, later acquired by McAfee. Reconnex built a gigabit line-rate network security analyzer capable of

classification, heuristic analysis, and policy enforcement on arbitrary protocols and content types. The Reconnex product captured and analyzed content transmitted over the internet using HTTP and other protocols, including storage protocols. Design of the Reconnex system was based on a modified and customized version of the Linux kernel which included specifically designed driver source code and interfacing between the application and kernel domains. Reconnex technology was employed in the audit and compliance verticals based on its ability to quickly analyze and classify the content of transmissions and interact with protocols and communications crossing corporate and departmental boundaries.

12. Between August 2007 and September 2008, I founded Strangways, a company focused on addressing messaging security for webmail and internet messaging protocols. Products and technology developed by Strangways were sold to SendMail Inc., Iron Port (Cisco), and other companies. The Strangways product utilized a Squid caching server and ICAP content rewrite server to translate HTTP to SMTP.

13. Between September 2008 and April 2014, I worked as Chief Architect for Gridiron Systems and, through acquisition, Senior Director of Technology for Violin Memory. I held the Senior Director position with Violin Memory both before and after the company's IPO. My responsibilities included architecture and

design of flash media controllers, adaptive caching design, and machine learning for application storage behavior analytics.

14. The Gridiron product was a storage accelerator used in many analytic environments such as big data architectures with constant ingest and processing of financial, consumer and IoT data. In my role as Chief Architect, I often worked with customers to implement and accelerate their analytics environments and spoke on related topics at trade shows and industry user groups. During product development, I worked with the data science team at Shopzilla to improve database and storage performance for real-time shopping recommendation and user tracking for advertising.

15. My work at Webstacks and Reconnex relate directly to packet processing including concepts such as priority and rules based processing relevant to the current proceeding. Approximately two dozen of my issued patents relate directly to these concepts.

16. Additional information regarding my employment history and prior expert testimony is included in Exhibit 2002. As reflected in Exhibit 2002, I have consulted on several matters involving intellectual property disputes including instances of alleged patent infringement requiring analysis of source code, reviewing and analyzing patents, writing expert reports, and testifying in court. I have testified at deposition and at trial in a number of these cases.

III. Legal Principles

A. Claim Construction

17. I understand that the first step in performing a validity analysis of the patent claims is to interpret the meaning and scope of the claims by construing the terms and phrases found in those claims. I understand that the appropriate construction of a claim term is its ordinary and accustomed meaning as understood by one of ordinary skill in the art at the time of the invention in the context of the entire patent and the prosecution history.

18. I understand that standard for claim construction in an *inter partes* review is the same standard as is applied in district court proceedings.

19. I understand that a determination of the meaning and scope of the claims is a matter of law. I have been informed that to determine the meaning of the claims, one should consider the intrinsic evidence, which includes the patent's claims, written description, and prosecution history.

B. Burden of Proof

20. I understand that in an *inter partes* review, the petitioner has the burden of proving unpatentability by a preponderance of the evidence.

C. Anticipation

21. I have been instructed by counsel and understand that a reference is anticipated if a single prior art reference discloses each and every claim element,

either explicitly or inherently, as arranged in the same way as in the claim. I understand that where even one claim element is not disclosed in a reference, a contention of anticipation fails.

22. I further understand that when a reference fails to explicitly disclose a claim element, that reference inherently discloses that element only if the reference must *necessarily* (or “inherently”) include the undisclosed claim element.

D. Obviousness

23. I have been instructed by counsel and understand that a combination of prior art references may render a claim obvious if, at the time of the invention, a person of ordinary skill in the art would have selected and combined those prior-art elements in the normal course of research and development to yield the claimed invention.

24. I understand that in an obviousness analysis, one should consider the *Graham* factors, including: the scope and content of the prior art; the differences between the claimed inventions and the prior art; and the level of ordinary skill in the art. I further understand the obviousness analysis is to be performed on a claim-by-claim basis. I understand that a person of ordinary skill in the art is a person of ordinary creativity, not an automaton.

25. I have been instructed by counsel and understand that obviousness requires more than a mere showing that the prior art includes separate references

covering each separate limitation in a claim under examination. I understand obviousness requires the additional showing that a person of ordinary skill at the time of the invention would have been motivated to combine those references in a manner that would include all limitations of the challenged claim, and, in making that combination, a person of ordinary skill in the art would have had a reasonable expectation of success.

26. I also understand that an obviousness analysis must be conducted with awareness of the distortion caused by hindsight bias and with caution of arguments reliant upon *ex post* reasoning. For instance, I understand that when considering obviousness, I should put myself in the position of a person of ordinary skill in the field at the time of the invention, rather than considering new information that is known today, but was not known before the priority date of the challenged patent.

IV. Overview of the '918 Patent

27. The challenged claims of the '918 patent are relate to a wireless end-user device that allows applications to request data transfers over a network stack either directly via a first call to a network stack application programming interface (“API”), or indirectly via a second API call to a proxy media service manager, which manager acts as a proxy to interface for the network stack in lieu of the application. Regardless of whether the application (1) transfers data via a first API call to the network stack or (2) transfers data by proxy via a second API call to the

proxy media service manager, the claimed device is able to associate the data transfers with the requesting device application, and as such is able to track an aggregate data usage for both types of calls attributable to the application.

28. Dr. Traynor treats claim 1 is as representative, and that claim recites (using Dr. Traynor's numbering of claim elements):

[1.pre] A wireless end-user device, comprising:

[1.1] a wireless modem configurable to connect to a wireless network;

[1.2] a network stack configurable to receive and transmit data via the wireless modem and the wireless network;

[1.3] *a first network stack Application Programming Interface (API), containing at least **one first call accessible to each of a plurality of device applications, the first network stack API callable by each of the plurality of device applications** to open and use data packet flows via the network stack, the wireless modem, and the at least one wireless network;*

[1.4] *a second API containing at least **one second call accessible to each of the plurality of device applications, the second API callable by each of the plurality of device applications** to make a data transfer request for a media object associated with a network resource identifier supplied by the calling device application;*

[1.5] a media service manager prompted by the second call, to manage network data transfers for the media object by

interfacing with the network stack to retrieve the media object associated with the network resource identifier via the wireless modem and the wireless network; and [1.6] one or more service classification and measurement agents to associate wireless network data usage for the media object network data transfers with the device application that requests the data transfer for the media object, to associate wireless network data usage for respective data packet flows opened and used via the first network stack API with the device application opening such respective data packet flow, and to reconcile wireless network data usage for each of the plurality of device applications to track an aggregate wireless network data usage attributable to each of the plurality of device applications via both the first network stack API and the second API.

Ex. 1001, claim 1 (124:28-63).¹

29. Relevant to my analysis below, the claim provides for two alternative pathways for an application to initiate traffic—a path where the application interfaces with the network stack directly by issuing a call to the network stack’s API (via “the first network stack API callable by each of the plurality of device applications,” or a “first” API), and a path where the application interfaces with the

¹ All emphases added unless otherwise noted.

network stack indirectly by issuing a call to a proxy “media service manager” via a different API (“a second API”).

30. Then—regardless of whether the application interfaced ***directly*** with the network stack by issuing a call to the first network stack API, or whether the application made use of the network stack ***indirectly*** by issuing a call to the second API to prompt the proxy media service manager to interface with the network stack on the application’s behalf—the system has at least one service and classification agent to associate the corresponding traffic with the requesting application in order “to track an aggregate wireless network data usage attributable to each of the plurality of device applications.”

31. An example of this dual-path approach (where either path is an option) is depicted, for example, in FIG. 32 of the ’918 patent, reproduced below.

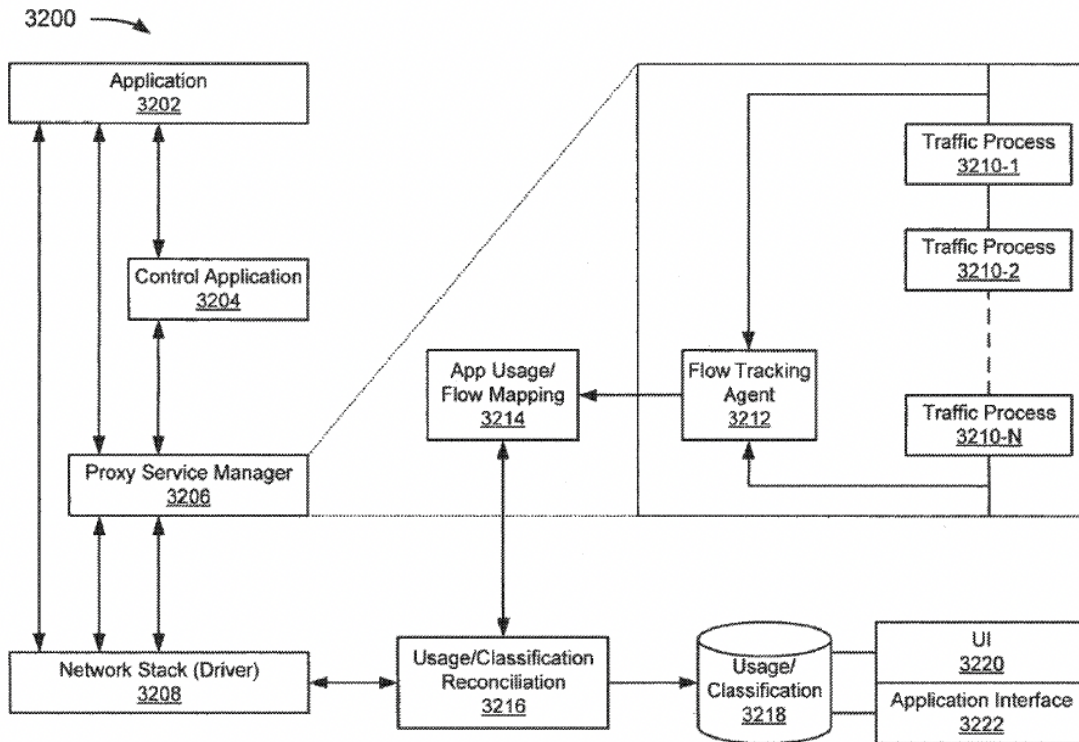


FIG. 32

Ex. 1001, FIG. 32. As shown above, Application 3202 is able to **directly** interface with the Network Stack 3208 via the left-most vertical arrow (by issuing an API call to the Network Stack 3208), and is also able to **indirectly** interface with the Network Stack 3208 via Proxy Service Manager 3206 (by issuing an API call to either the Proxy Service Manager 3206, or by issuing an API call to Control Application 3204 which would then call the Proxy Service Manager API).

32. In the event that “the application 3202 does **not** use the proxy service manager 3206,” and instead calls the network stack API via the left-most arrow, data at the socket will accurately identify the app as the initiator of the data flow” because the application issued a call to the network stack API to open the socket

and to initiate the data flow. Ex. 1001, 116:66-117:3. However, “[i]n an instance or embodiment where the application uses the proxy service manager 3206” rather than calling the network stack API, “data at the socket will be insufficient to identify the application 3202 as the initiator of the data flow.” Ex. 1001, 117:3-6. To solve the socket-level data being insufficient to identify the application 3202 as the initiator of the data flow, the ’918 patent recognizes the advantage “to add a virtual tag at the proxy API,” such that the virtual tag is “sent from the flow tracking agent 3212” (which is depicted as being part of the proxy service manager) “to the app usage/flow mapping engine 3214, which communicates with the usage/classification reconciliation 3216, which in turn communicates with the network stack (driver) 3208.” Ex. 1001, 117:12-20. A POSITA would understand that this virtual tag helps the system track the application initiating the data transfer, even in the scenario where the application does not call the network stack API (and instead calls a different API which in turn causes a different component to call the network stack API).

33. Alternatively to the use of a virtual tag, “the proxy service manager can perform application identification and classification, usage monitoring, and counting” for traffic flows where an application uses the proxy instead of calling the network stack API. Ex. 1001, 117:29-33. This process “allows service usage to

be classified and associated with an application prior to proxy processing.” Ex. 1001, 117:31-33.

34. The purpose of these disclosures is to inform a POSITA how traffic initiated by applications can be tracked both (1) in an instance where the application directly interfaces with the network stack by calling the network stack API and (2) in an instance where the application does not call the network stack, but instead uses a proxy to initiate the data transfer. Specifically, regardless of which access path is used (a call to the network stack API or a call that initiates a network transfer via a proxy), the system is able to “associate wireless network data usage” with the application initiating the data flow in order “to track an aggregate wireless network data usage attributable to each of the first plurality of device applications via both the first network stack API and the second API.” Ex. 1001, 124:51-63.

V. Person of Ordinary Skill in the Art

35. Dr. Traynor proposes that a POSITA in the field of the ’918 patent “would have had (1) at least a bachelor’s degree in computer science, computer engineering, electrical engineering, or a related field, and (2) at least two years of industry experience in wireless communication network applications and software. Additional graduate education could substitute for professional experience, and *vice versa*.” Ex. 1003 ¶21. For purposes of this my analysis, I do not disagree with

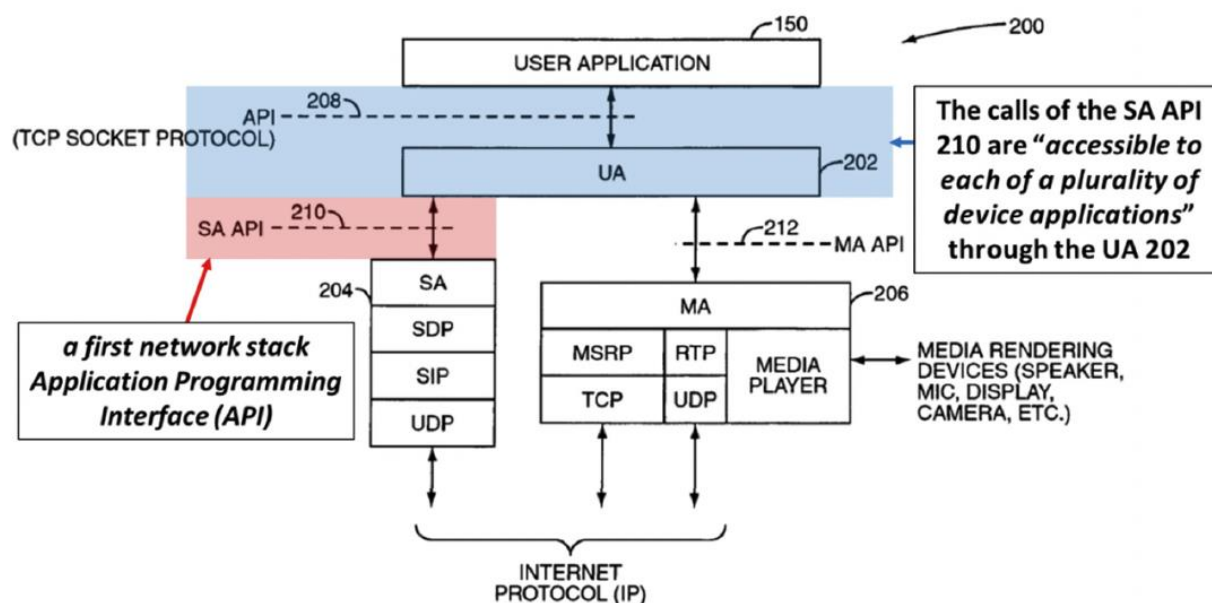
this proposed level of skill in the art. I meet at least these requirements of a POSITA, and have met those requirements since before the priority date of the '918 patent.

VI. Dr. Traynor's challenge to claim 1 is inconsistent with plain meaning.

36. As I discussed above in Section IV, claim 1 requires that a plurality of device applications be able to initiate data transfers over the network in two different ways. The first way is through a “first call accessible to each of a plurality of device applications,” which “first network stack API [is] callable by each of the plurality of device applications.” Ex. 1001, 124:33-39 (limitation [1.3]). The second way is through “a second API containing at least one second call accessible to each of the plurality of device applications,” which “second API [is] callable by each of the plurality of device applications to make a data transfer request for a media object.” Ex. 1001, 124:40-45 (limitation [1.4]). The second API call will prompt “a media service manager... to manage network data transfers... by interfacing with the network stack” instead of the application interfacing with the network stack. Ex. 1001, 124:46-50 (limitation [1.5]).

37. Then, **regardless** of whether applications use the first network stack API or the second API to initiate data transfer, the claimed device is able to “track an aggregate wireless network data usage attributable to each of the plurality of

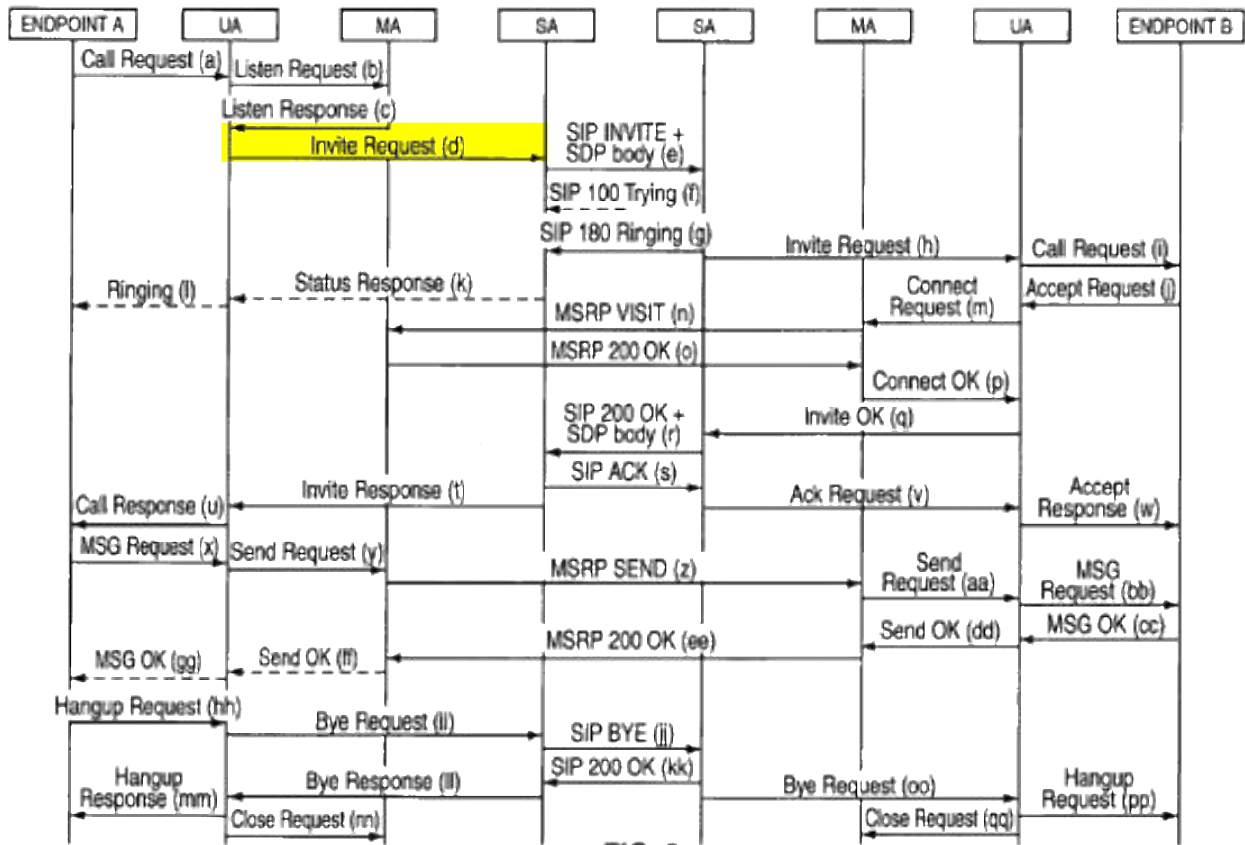
device applications via both the first network stack API and the second API.” Ex. 1001, 124:51-63 (limitation [1.6]).



Ex. 1003, ¶56 (Dr. Traynor’s annotation of Ex. 1041, FIG. 3). As shown in the annotation above, the application 150 (i.e., one of a “plurality of device applications”) contacts API 208 to user agent (UA) 202, which in turn contacts “SA API 210.” Ex. 1003, ¶55.

39. As a point of clarification, I note that Bennett does not **explicitly** discuss any API “calls”; instead, its discussion of a “call” is in the more traditional sense of, e.g., a telephone “call” which would initiate, e.g., “ringing” another user using a different device. *See, e.g.*, Ex. 1041, ¶¶63-64. A POSITA would understand Bennett does imply API calls, which are represented by direct communication between system components (wherein the sending component issues an API call to the receiving component—e.g., user application 150 calling UA 202, or UA 202 calling SA 204 (or conversely, SA 204 calling UA 202)).

40. Dr. Traynor alleges that “SA API 210[’s]” “requests” (i.e., requests transmitted between UA 202 and SA API 210) constitute the “at least one first call accessible to each of a plurality of device applications.” Ex. 1003, ¶56. These SA API “requests” are depicted, e.g., in FIG. 6 of Bennett, an annotated version of which is reproduced below:



Ex. 1041, FIG. 3 (annotated); *see also* Ex. 1003, ¶56 (explicitly identifying the “INVITE” request as the “at least one first call” which is allegedly “accessible to each of a plurality of device applications”).

41. In Bennett’s FIG. 6, the communications between “ENDPOINT A” and “UA” represent communications between “user application 150” and “UA 202,” which a POSITA would understand because the patent discloses the FIG. 6 communications between the endpoints as being communications between the “user application 150” and the “UA 202.” *See, e.g.*, Ex. 1041 ¶34 (“The CALL request is sent by the user application 150 to the UA 202....”); *id.* ¶36 (“The

ACCEPT request is sent by the user application 150 responsive to a CALL request to instruct the UA 202 to either accept or reject an incoming call.”); *id.* ¶39 (“When the HANGUP request is sent by the user application 150 to the UA 202, the UA 202 sends a HANGUP response to confirm that the call is ended.”).

42. However, Dr. Traynor does not show that that these requests are “accessible to each of a plurality of device applications” in Bennett. Instead, Bennett is clear that the user agent is not able to access the “invite request” (the only example alleged or a “bye request,” but instead that only the UA 202 has access to these requests; the user agent can issue a “call request” that may **cause** the UA 202 to issue an “invite request,” but a POSITA would understand that the “invite request” itself is not accessible to the user application. In my opinion, Dr. Traynor does not provide any explanation of how these requests, which cannot be made by user application 150, are “accessible to” user application 150 as the claim requires. Instead, he appears to acknowledge that the user application **cannot** issue an invite request.

43. My understanding (and the understanding of a POSITA) is that in order for an API request to be “accessible” to an application, the application must be able to actually make the API request. Thus, if an application **cannot** make a request, that request is **not accessible** to the application. And it is clear from Bennett (e.g., Bennett’s FIG. 6) that Bennett does not disclose or otherwise suggest

that user application 150 (the “ENDPOINT” in FIG. 6) is able to make either the “invite” request or any other alleged call of any alleged “network stack API,” and I do not understand Dr. Traynor to allege otherwise.

44. Indeed, Dr. Traynor appears to acknowledge the first API is **not** “directly ‘accessible to each of a plurality of device applications,’” and that the application **cannot** “directly call the ‘first’ API.” Ex. 1003, ¶57. Rather than allege that the alleged network stack API (SA 210) has a request (the alleged “call”) that is “accessible” to the user application under his opinion regarding the **plain meaning** of “accessible” (an opinion on which Dr. Traynor does not actually express), he contends that it is “accessible” because the ’918 patent teaches that applications can use a proxy or some other middleman entity that will facilitate network access, rather than having the application call the API. See Ex. 1003 ¶¶57-58 (alleging that FIGs. 30, 35, and 12 as supposedly “indicating that such an interpretation [where the application need not actually have access to the API request] is within the scope of the claims [which require the API request is ‘accessible’ to the application”).

45. I have reviewed the cited teachings of the ’918 patent in this regard, and I believe Dr. Traynor overlooks or mischaracterizes the relationship of these teachings to the challenged claims. Indeed, it is my opinion that his interpretation is significantly at odds with the plain meaning and purpose of the claim language,

sufficiently so that Dr. Traynor’s interpretation would render superfluous the claim requirement that a “first network stack [API]” contains “at least one first call accessible to each of a plurality of device applications.”

46. For example, FIG. 30 (reproduced below) relied upon by Dr. Traynor clearly discloses to a POSITA that one way an application may initiate network traffic is by issuing an API call to a “proxy” 3012, which will then initiate network traffic by opening a thread 3014 which issues an API call to socket 3016.

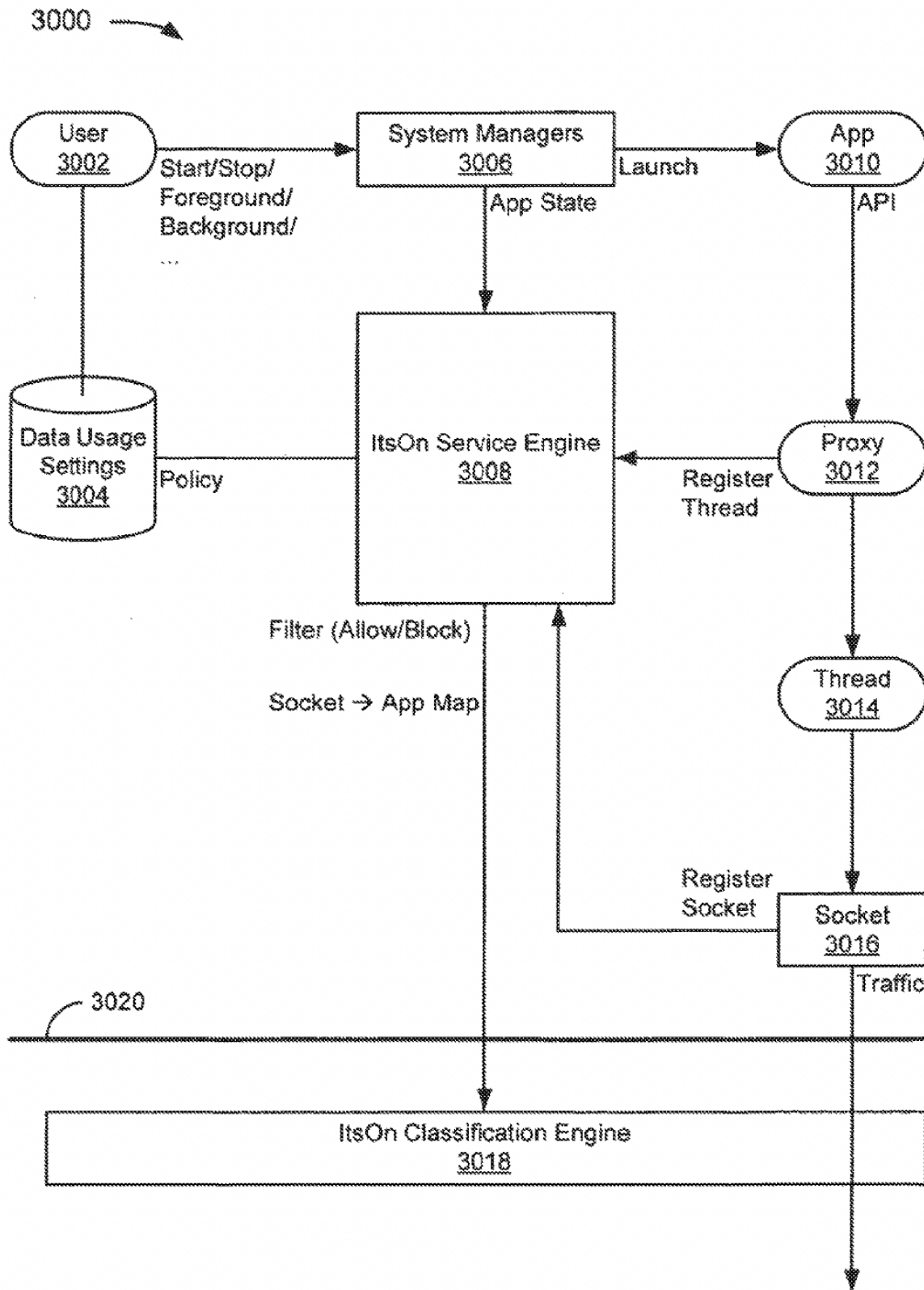


FIG. 30
 22

Ex. 1001, FIG. 30; *id.* at Ex. 1001 110:46-48 (“In the example of FIG. 30, the app 3010 utilizes an API to trigger the proxy 3012 which in turn passes through a socket connection at the socket 3016 as traffic.”). Nothing in FIG. 30, or the description of FIG. 30, would imply to a POSITA that the app “accesses” a network stack API call or calls the network stack API in this example.² Instead, a POSITA would understand this example traffic flow as showing the app calling a proxy API by accessing a proxy API call. Ex. 1001, 110:46-53 (“In the example of FIG. 30, the app 3010 utilizes an API to trigger the proxy 3012 which in turn passes through a socket connection at the socket 3016 as traffic.”).

47. In relationship to the claim language, the example FIG. 30 traffic flow shows the application making use of the **second** API, which is a “proxy” (an example of which the ’918 patent discloses as being a “service manager” such as the claimed “media services manager”). Ex. 1001, 114:11-21 (discussing “an application... that requests from a proxy service manager a data transfer (a data transfer request)”); *id.* at 114:24-26 (discussing “each application that requests a

² The fact that this application flow does not show the application accessing the network stack API call does not mean that the network stack API call is not **accessible** to the application, it just means that FIG. 30 does not itself depict an access to a network stack API call. A POSITA would understand FIG. 30 to be depicting one example of a possible data flow, which corresponds, e.g., with the application’s call to the “second API” recited in the claim, rather than the “first network stack” API recited in the claim.

data transfer from the proxy service manager or that conducts a data transfer with the proxy service manager”); Ex. 1003 ¶24 (Dr. Traynor’s discussion of a “proxy network service manager”).

48. I further note that if Dr. Traynor **were** correct in his implication that a plurality of applications’ ability to access a call of a **proxy** API (e.g., an API to prompt “a media service manager” to “manage network data transfers... by interfacing with the network stack” as recited by limitations [1.4] and [1.5]) were sufficient for a “first network stack [API]” to “contain[] at least one first call accessible to each of a plurality of device applications” (as recited by limitation [1.3]), then limitation [1.3] would not need to recite that the first network stack API contains a call accessible to the applications; this would be redundant with the claim’s other recitations of the “second API” having a call which prompts the media service manager to “manage network data transfers” on behalf of the application “by interfacing with the network stack.” See Ex. 1001, 124:40-50 (limitations [1.4] and [1.5]).

49. A POSITA would not understand the claim language to be superfluous in this manner. Instead, a POSITA would understand the claims to mean precisely what they say: that both the network stack API **and** the second API (e.g., proxy API) must contain calls that are accessible to the application, such that the application can thus be able to issue a call to either API (which is further

enforced by the claim requirements that each API be “**callable** by each of the plurality of device applications”). Not only is this the plain meaning of the claim language, but it is also consistent with the remainder of the claim language (and an important purpose of the invention of the ’918 patent), which is to “track an aggregate wireless network data usage attributable to each of the device applications” **regardless** of which API the application calls. Ex. 1001, 124:60-63 (“track an aggregate wireless network data usage attributable to each of the plurality of device applications via **both** the first network stack API and the second API”).

50. I disagree with Dr. Traynor’s reliance on FIG. 35 (*see* Ex. 1003, ¶57), reproduced below, for similar reasons.

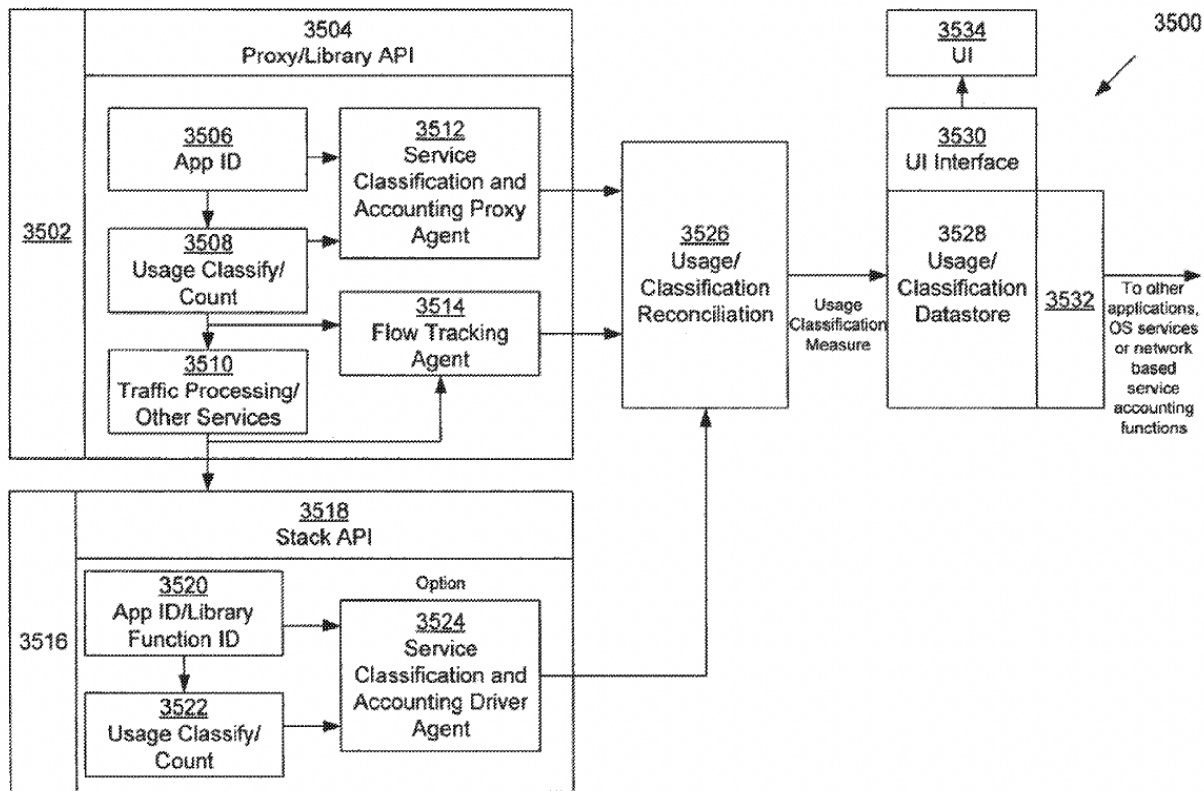


FIG. 35

Ex. 1041, FIG. 35 (cited by Ex. 1003, ¶157).

51. As shown above, FIG. 35 discloses a network stack API (3518) being called by a proxy API (3504), which is precisely what the claim contemplates when it references the second API causing a media service manager to “interface with the network stack” on the application’s behalf. Ex. 1001, 124:40-50. Nothing in FIG. 35 discloses or even remotely implies that a proxy API interfacing with a network stack API on an application’s behalf is itself sufficient for the network stack API to have an API call accessible to an application that can **only** call the proxy API (which is what Dr. Traynor’s theory entails).

52. Dr. Traynor’s reliance on FIGs. 12-13 (Ex. 1003, ¶58) is similarly inapposite. Specifically, FIGs. 12-13 disclose applications interfacing with an “Application Interface” or an “Application Service Interface Agent.” But again, the patent does not teach or suggest that an application interfacing with an “application interface” or an “application service interface agent” is somehow accessing an API call for a **different** element of the system. To the contrary, the patent implies that the “application interface” can be part of the “network stack” level in that embodiment. Ex. 1001, 8:45-52 (referring to a “network stack application interface level/layer”). Furthermore, the Petition does not allege that FIGs. 12-13 are even embodiments of claim 1. In embodiments that **do** clearly provide examples of the claimed invention (e.g., FIG. 32), the “application interface” is not shown as being interposed between the application 3202 and network stack 3208.

53. Dr. Traynor’s reliance on “network based APIs” that are located “on a network element” (Ex. 1003 ¶58) also does not imply that a network stack API can have a call accessible to an application even if the application cannot call that API. Dr. Traynor does not explain **why** he believes this teaching of the ’918 patent supports his deviation from plain meaning, and it does not. For instance, a “network **based** API” is not the same thing as “network **stack** API”—a “network based API” is simply any API that is based in (e.g., located in) or accessed through a network, where as a “network stack API” would be understood by a POSITA to

be an API for accessing the stack of network layers (e.g., OSI network layers) within a system. Additionally, even if a network stack somehow existed outside of the device itself, that does not mean that the network stack API would have a call that is accessible to any particular application. Instead, a POSITA would understand that for a call to be accessible to an application, the application must actually have access to that specific call (not merely be able to perform actions that may ultimately result in the call being made, depending on the actions of other system components).

54. I also note that Dr. Traynor does not even allege the specification provides for lexicography (e.g., a definition of the claim terms), and in any event the specification is entirely consistent with the plain meaning of the claim language which is that the API must actually be callable by the application. Ex. 1001, 124:33-39 (limitation [1.3]).

55. In sum, I do not believe that Dr. Traynor's invalidity theory is based on the plain meaning of the claim language. While he relies on the patent specification in an attempt to justify his deviation from the plain meaning of the claim, the portions of the specification he relies on do not support his interpretation; instead, they confirm what the claims say, which is that the applications may call **either** a network stack API **or** a proxy / service manager API. Dr. Traynor also overlooks an important purpose of the claimed invention which is

to evaluate traffic initiated by API calls to *either* a network stack API *or* an API that will initiate a proxy “media service manager” to “manage network data transfers... by interfacing with the network stack” on behalf of the application, in order to “track an *aggregate* wireless network data usage attributable to” the applications. Ex. 1001, 124:40-63. If the claimed system did not need to include the ability of an application to call the network stack API, it would defeat this purpose.

56. Dr. Traynor’s reliance on the specification involve either entirely irrelevant examples that are not alleged to be claimed embodiments (e.g., FIGs. 12-13 and Ex. 75:26-37), or focus on examples in the specification where an application calls a proxy API to interface with the network stack on the application’s behalf as supposed evidence of claim broadening (e.g., FIGs. 30, 35). In the latter instance, the use of a proxy API is entirely consistent with the *additional* claim requirement that a network stack API is *itself* callable by applications.

57. Accordingly, it is my opinion that Dr. Traynor’s invalidity theory is inconsistent with the requirements of limitation [1.3]; the Petition’s theory does not show “a first network stack [API] containing at least one first call accessible to each of a plurality of device applications [and] the first network stack API callable by each of the plurality of device applications” as claim 1 requires.

58. I also believe that his theory is inconsistent with the requirements of limitation [1.4], as he also does not allege that the applications are able to issue a call to his alleged “second” API (“MA API 212”). *See* Ex. 1003, ¶¶60-62.