

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

CISCO SYSTEMS, INC. and FORTINET, INC.,
Petitioner

IPR2024-00678

U.S. Patent No. 8,117,645

**DECLARATION OF HENRY HOUH, PH.D., UNDER 37 C.F.R. § 1.68 IN
SUPPORT OF PETITION FOR *INTER PARTES* REVIEW**

EXHIBIT LIST

No.	Exhibit Description
1301	U.S. Patent No. 8,117,645
1302	U.S. Patent No. 8,117,645 File History
1303	Declaration of Henry Houh, Ph.D
1304	CV of Henry Houh, Ph.D
1305	U.S. Patent Pub. No. 2004/0111520 to Krantz et al
1306	U.S. Patent Pub. No. 2004/0107360 to Herrmann et al
1307	IEEE Std 802.1Q-1998 – IEEE Standard for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks
1308	U.S. Patent No. 6,430,621 to Srikanth et al.
1309	U.S. Patent No. 6,032,194 to Gai et al.
1310	Cisco IOS 12.0 Switching Services (1999)
1311	Microsoft Computer Dictionary 5 th Edition (2002)
1312	Webster’s New World Computer Dictionary (2001)
1313	[RESERVED]
1314	[RESERVED]
1315	[RESERVED]
1316	RFC 2716 – PPP EAP TLS Authentication Protocol
1317	Wireless security with 802.1x and PEAP (2003)
1318	RFC 5216 – The EAP TLS Authentication Protocol
1319	Incorporating WS-Security into a Web services-based portal by B.L. Lim et al. (2004)
1320	Newton’s Telecom Dictionary 18 th Edition

I. INTRODUCTION

1. I am making this declaration at the request of Petitioner in the matter of the *Inter Partes* Review of U.S. Patent No. 8,117,645 (“the 645 Patent”) to Lum et al.

2. I am being compensated for my work in this matter at my standard hourly rate. I am also being reimbursed for reasonable and customary expenses associated with my work and testimony in this proceeding. My compensation is not contingent on the outcome of this matter or the specifics of my testimony.

3. I have been asked to provide my opinions regarding whether the subject matter of claims 1-2 and 5-6 (“the Challenged Claims”) of the 645 Patent are anticipated or otherwise would have been obvious to a person having ordinary skill in the art (“POSITA”) at the time of the alleged invention, in light of the prior art. It is my opinion that the Challenged Claims would have been obvious to a POSITA.

4. In the preparation of this declaration, I have studied:

- a. the 645 Patent, Ex.1301.
- b. the prosecution history of the 645 Patent (“645 File History”), Ex.1302.
- c. U.S. Patent Pub. No. 2004/0111520 to Krantz et al., (“Krantz”), Ex.1305.

d. U.S. Patent Pub. No. 2004/0107360 to Herrmann et al.

(“Herrmann”), (Ex.1306).

e. The additional references I refer to below and/or identified in the exhibit list.

5. In forming the opinions expressed below, I have considered: the documents listed above; the relevant legal standards, including the standard for obviousness; and my own knowledge and experience based upon my work in the field of network communications and security as described below.

6. Unless otherwise noted, all **emphasis** in any quoted material has been added. Claim terms are *italicized*.

II. QUALIFICATIONS AND PROFESSIONAL EXPERIENCE

7. My complete qualifications and professional experience are described in my *Curriculum Vitae*, a copy of which can be found in Ex.1304. The following is a brief summary of my relevant qualifications and professional experience.

8. I received a Ph.D. in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology (“MIT”) in 1998. Beforehand, I received a Master of Science degree in Electrical Engineering and Computer Science in 1991, a Bachelor of Science degree in Electrical Engineering and Computer Science in 1989, and a Bachelor of Science degree in Physics in 1990, all from MIT.

9. I am currently self-employed as an independent technical consultant. I am also president of a company that provides supplemental science, technology, engineering, and mathematics (“STEM”) education to children of all ages.

10. I first entered telecommunications in 1987 when I worked as a summer intern at AT&T Bell Laboratories as part of a five-year dual degree program at MIT. I continued to work at AT&T Bell Laboratories as part of this MIT program. While at MIT, I was a teaching assistant (“TA”) in the Electrical Engineering and Computer Science Department’s core Computer Architectures course. I first was a TA as a senior for a role typically reserved for graduate students. I later became head TA. The course covered various topics in computer architectures. As a TA, I helped write homework assignments, lab assignments, and exams. I also taught in the recitation sections.

11. Later, as part of my doctoral research at MIT from 1991-1998, I was a research assistant in the Telemedia Network Systems (“TNS”) group at the Laboratory for Computer Science. The TNS group built a high-speed gigabit network and created applications that ran over the network. Example applications included ones for remote video capture, processing, and display of video on computer terminals. In addition to working on the design of core network components, designing and building the high-speed links, and designing and

writing the device drivers for the interface cards, I also set up the group's web server.

12. I also helped to build the web pages that initiated the above-mentioned video sessions via a web interface. Vice President Al Gore visited our group in 1996 and received a demonstration of—and remotely drove—a radio-controlled toy car with a wireless video camera mounted on it that was built by our group. This toy car device received commands transmitted over a network from a remote computer, and video data from the toy car was transmitted wirelessly then over a computer network back to the user controller. On occasion, we allowed users visiting our web site to drive the toy car from their remote computer while they watched the video on their computer. The video stream was encoded by TNS-designed hardware, streamed over the TNS-designed network, and displayed using TNS-designed software.

13. I defended and submitted my Ph.D. thesis, titled “Designing Networks for Tomorrow’s Traffic,” in January 1998. As part of my thesis research, I analyzed local area and wide area flows to show a more efficient method for routing packets in a network, based on traffic patterns at the time.

14. From 1997 to 1999, I was a Senior Scientist and Engineer at NBX Corporation, a start-up that made business telephone systems for streaming packetized audio over data networks instead of using traditional telephone lines.

NBX was later acquired by 3Com Corporation and the phone system is still used today by numerous businesses.

15. As part of my work at NBX, I designed the core audio reconstruction algorithms for the telephones, as well as the packet transmission algorithms. I also designed and validated the core packet transport protocol used by the phone system. The protocol was used for all signaling in the phone system, including for the setup of conference calls.

16. The NBX system also featured a computer interface for initiating phone calls, which could also initiate conference calls. The NBX system also supported the Telephony Application Programming Interface (“TAPI”) that allowed other computer programs to integrate with our system telephony features. We obtained U.S. Patent No. 6,967,963, entitled “Telecommunication method for ensuring on-time delivery of packets containing time-sensitive data,” as part of this work. During my time at NBX, I also evaluated and purchased VPN equipment in order to demonstrate that our system worked over long distances through VPN devices, so that our phones could be located remotely. We used the VPN equipment to connect from various trade shows, for example, in Las Vegas and Los Angeles, remotely into our home network so that phones in the remote network were able to connect to our home facility in Andover, MA.

17. From 1999-2004, I was employed by Empirix or its predecessor company, Teradyne. Empirix was a leader in test tools for telecommunications protocols and systems, providing functional testing tools as well as load testing tools. From 2000-2001, I conceived and built a test platform for testing Voice-over-IP (VoIP). The first application on this new test platform was a cloud emulator for simulating the effects of transmitting VoIP over a busy network.

18. In 2006, as part of my role at BBN Technologies, I helped found PodZinger Inc., now known as RAMP Inc. PodZinger utilized BBN's speech recognition algorithms to search through the spoken words in audio and video segments. While I was Vice President of Operations and Technology, PodZinger followed its initial prototype with a full streaming audio and video search solution. I also created a social networking web site, which BBN sold to a venture-funded startup company. In the process of creating the web site, I designed and specified the authentication and authorization protocols.

19. I have been awarded several United States patents, and I have several patent applications pending including the following examples:

- U.S. Patent No. 7,975,296, "Automated security threat testing of web pages";
- U.S. Patent No. 7,877,736, "Computer language interpretation and optimization for server testing";
- U.S. Patent No. 7,801,910, "Method and apparatus for timed tagging of media content";

- U.S. Patent 7,590,542, “Method of generating test scripts using a voice-capable markup language”;
- U.S. Patent No. 6,967,963, “Telecommunication method for ensuring on-time delivery of packets containing time-sensitive data”;
- U.S. Patent Application Publication No. 20070106685, “Method and apparatus for updating speech recognition databases and reindexing audio and video content using the same”;
- U.S. Patent Application Publication No. 20070106693, “Methods and apparatus for providing virtual media channels based on media search”;
- U.S. Patent Application Publication No. 20070106760, “Methods and apparatus for dynamic presentation of advertising, factual, and informational content using enhanced metadata in search-driven media applications”;
- U.S. Patent Application Publication No. 20070112837, “Method and apparatus for timed tagging of media content”;
- U.S. Patent Application Publication No. 20070118873, “Methods and apparatus for merging media content”; and
- U.S. Patent Application Publication No. 20090222442, “User-directed navigation of multimedia search results.”

III. LEVEL OF ORDINARY SKILL IN THE ART

20. I understand there are multiple factors relevant to determining the level of ordinary skill in the pertinent art, including (1) the levels of education and experience of persons working in the field at the time of the invention; (2) the sophistication of the technology; (3) the types of problems encountered in the field; and (4) the prior art solutions to those problems.

21. A Person of Ordinary Skill in The Art (“POSITA”) in September 2003 would have had a working knowledge of the telecommunications and networking art that is pertinent to the 645 Patent. A POSITA would have had a bachelor’s degree in computer science, computer engineering, or an equivalent, and two years of relevant experience in networking and network security, and related fields. Lack of professional experience can be remedied by additional education, and vice versa.

22. For purposes of this Declaration, in general, and unless otherwise noted, my statements and opinions, such as those regarding my own experience and what a POSITA would have understood or known generally (and specifically related to the references I consulted herein), reflect the knowledge that existed in the relevant field as of the priority date of the 645 Patent.

IV. RELEVANT LEGAL STANDARDS

23. I am not an attorney. In preparing and expressing my opinions and considering the subject matter of the 645 Patent, I am relying on certain basic legal principles that counsel has explained to me.

24. I understand that prior art to the 645 Patent includes patents and printed publications in the relevant art that predate the priority date of the 645 Patent. For purposes of this Declaration, I am applying September 24, 2003, as the priority date of the 645 Patent, but I take no position on that issue.

25. I have been informed by counsel that a claimed invention is invalid if it is “anticipated” by the prior art. I understand that, for a patent claim to be “anticipated” by the prior art, each and every requirement of the claim must be found, expressly or inherently, in a single prior art reference as recited in the claim. I understand that claim limitations that are not expressly described in a prior art reference may still be there if they are “inherent” to the thing or process being described in the prior art.

26. I understand that if a reference incorporates other documents by reference, the incorporating reference and the incorporated reference(s) should be treated as a single prior art reference for purposes of analyzing anticipation.

27. I understand that it is acceptable to consider evidence other than the information in a particular prior art document to determine if a feature is necessarily present in or inherently described by that reference.

28. I have been informed by counsel that a claimed invention is unpatentable under 35 U.S.C. § 103 if the differences between the claimed invention and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a POSITA. I have also been informed by counsel that the obviousness analysis considers factual inquiries, including the level of ordinary skill in the art, the scope and content of the prior art, and the differences between the prior art and the claimed subject matter.

29. I have been further informed by counsel that there are several recognized rationales for combining references or modifying a reference to show obviousness. These rationales include: (a) combining prior art elements according to known methods to yield predictable results; (b) simple substitution of one known element for another to obtain predictable results; (c) use of a known technique to improve a similar device (method, or product) in the same way; (d) applying a known technique to a known device (method, or product) ready for improvement to yield predictable results; (e) choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success; and (f) some teaching, suggestion, or motivation in the prior art that would have led a POSITA to modify the prior art or to combine prior art teachings to arrive at the claimed invention.

V. OVERVIEW OF THE 645 PATENT

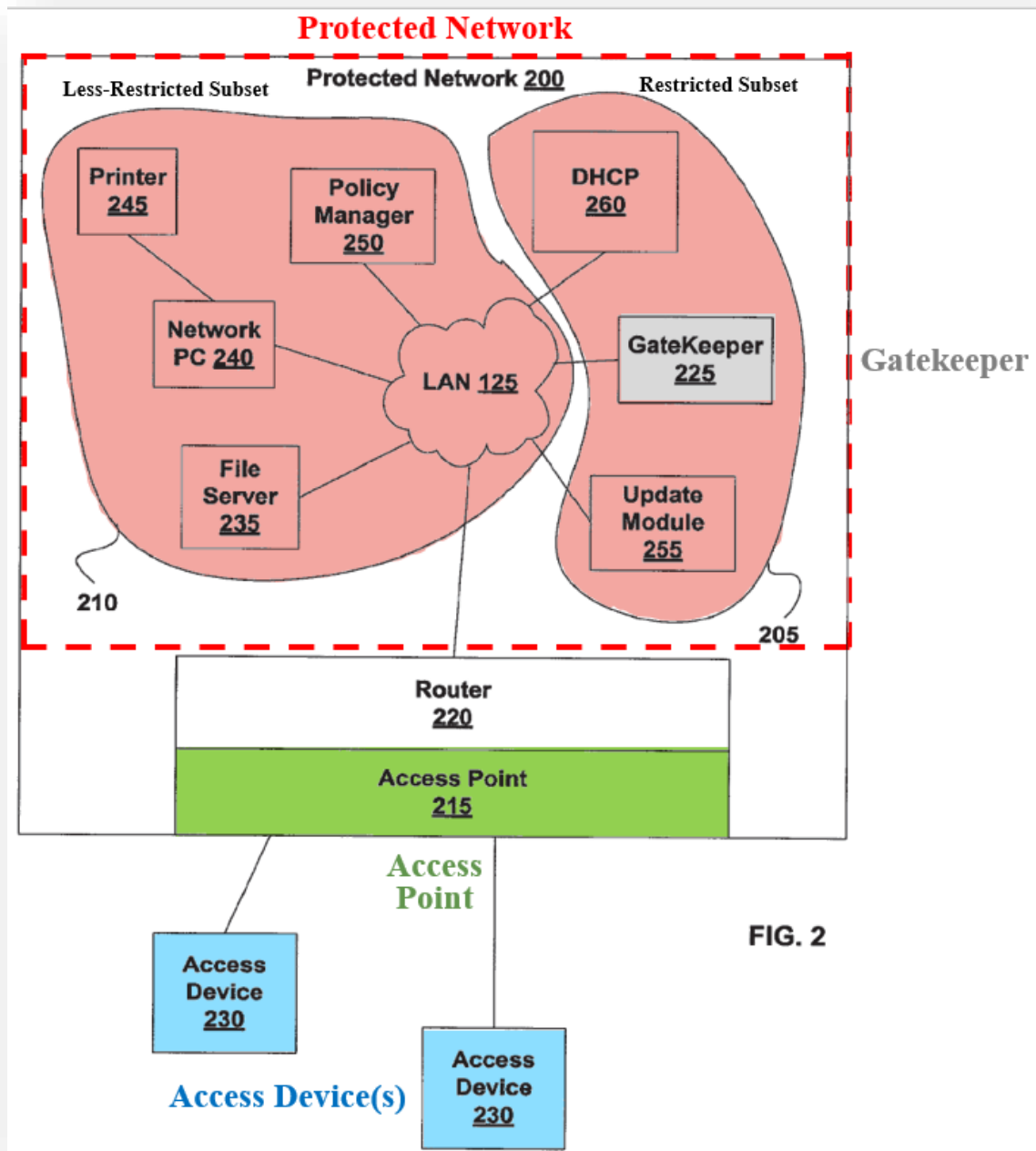
30. The 645 Patent states that it is “a continuation and claims the priority benefit of U.S. patent application Ser. No. 12/273,037 filed Nov. 18, 2008, which is a continuation and claims the priority benefit of U.S. patent application Ser. No. 10/949,179 filed Sep. 24, 2004 (now U.S. Pat. No. 7,523,484), which claims the priority benefit of U.S. Provisional Patent Application No. 60/505,582 filed Sep. 24, 2003 and entitled “Gatekeeper Inventions for Controlling Access to a Corporate Network” and U.S. Provisional Patent Application No. 60/513,080 filed

Oct. 21, 2003.” 645 Patent, 1:7-16. I have been asked to assume that **September 24, 2003**, is the effective filing date, but I take no position on that issue.

31. I have reviewed the file history of the 645 Patent, including that of U.S. Patent No. 7,523,484, to which the 645 Patent claims priority, and note that none of the activity there is pertinent to my analysis.

32. U.S. Patent No. 8,117,645 (“the 645 Patent”) is generally directed to systems and methods that describe “a protected network physically or logically divided into a restricted subset and one or more less-restricted subsets.” 645 Patent, 2:50-53. Access to less-restricted networks from a device “are first passed to the restricted subset.” 645 Patent, 2:53-54. Elements of the “restricted subset”—such as a “gatekeeper”—“are then used to apply security policies” to the device requesting access. 645 Patent, 2:54-58. If the requesting device satisfies the “security policies,” access is then granted “to a less-restricted subset of the protected network.” 645 Patent, 2:57-58. The 645 Patent further explains that both “the passing of access requests to the restricted subset” and the subsequent granting of such requests is “accomplished by configuring and reconfiguring a communication port of an access point.” 645 Patent, 2:58-61.

33. Figure 2 depicts various embodiments of the invention described in the 645 Patent, and is annotated below:



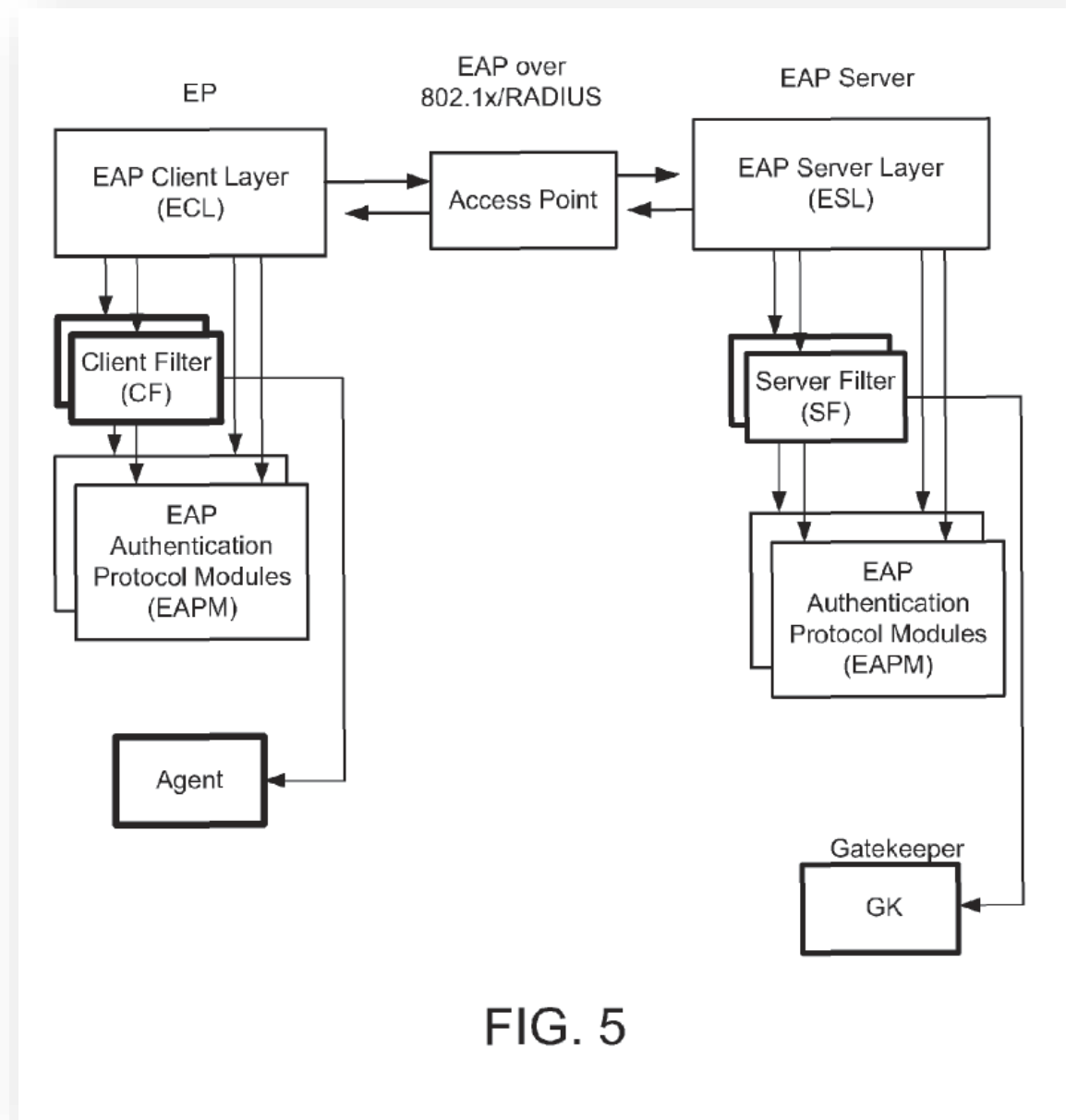
Ex.1301 (645 Patent), Fig. 2 (annotated)

34. For example, Figure 2 includes “Protected Network 200,” which is described in the 645 Patent as “either physically or logically divided into a Restricted Subset 205 and a Less-Restricted Subset 210.” 645 Patent, 5:55-58.

The 645 Patent also explains that access to Protected Network 200 is “under the control of a GateKeeper 225,” which “enforce[s] a security policy against one or more Access Device 230 requesting access.” 645 Patent, 5:59-64. The 645 Patent describes that the security policy may proscribe up-to-date virus software and encryption protocols, (Ex.1301 2:16-27), include requirements related to user identification and passwords or configuration requirements related to the access device (such as aspects of the operating system), (Ex.1301, 6:38-52), or authenticate an access device (or user) of the access device. 645 Patent, 10:36-41, 12:2-6.

35. The 645 Patent articulates that “[i]n some embodiments, Authenticate User Step [] is considered part of the security policy audit.” Ex.1301, 12:2-6; *see also*, Ex.1301, 11:43-54 (giving examples of audit/system data including, operating system status, antivirus software status, information about devices connected to the Access Device, MAC address, IP address, etc.). Moreover, “[a]ccording to an exemplary embodiment, client filters (CF) and server filters (SF) are added along with an agent and GK [i.e., gatekeeper] to perform audits, as highlighted in FIG. 5. There may be one or more instances of the CF and SF, which correspond to the separate EAP Authentication Modules (EAPM) that are supported.” 645 Patent, 14:22-27. Notably, “the agent and [gatekeeper] are logical components. The functionality of the agent and [gatekeeper] can be embedded into the CF and SF,

respectively, or ... may reside on a separate server.” 645 Patent, 14:27-30. Fig. 5 is shown below for reference.



Ex.1301 (645 Patent), Fig. 5

36. However, as demonstrated below, it was well-known in the prior art to provide access control mechanisms utilizing gatekeepers that facilitate access to protected networks by user devices through the use of security policies.

VI. CLAIM CONSTRUCTION

37. It is my understanding that in order to properly evaluate the 645 Patent, the terms of the claims must first be interpreted. It is my understanding that for the purposes of this *inter partes* review, the claims are to be construed under the so-called *Phillips* standard, under which claim terms are given their ordinary and customary meaning as would have been understood by a POSITA in light of the specification and prosecution history, unless the inventor has set forth a special meaning for a term. I have also been informed that claim terms only need to be construed to the extent necessary to resolve the unpatentability inquiry.

VII. IDENTIFICATION OF HOW THE CLAIMS ARE UNPATENTABLE

38. The discussion in this Declaration provides a detailed analysis of how the asserted prior art teaches each limitation of the Challenged Claims.

39. As part of my analysis, I have considered, and discuss in detail, the scope and content of the prior art and any differences between the alleged invention and the prior art.

40. It is my opinion that the alleged invention recited in the Challenged Claims would have been obvious in view of the teachings of the asserted prior art and the knowledge of a POSITA.

VIII. THE PRINCIPAL PRIOR ART

A. Summary of Krantz

41. U.S. Patent Pub. No. 2004/0111520 to Krantz et al. (Ex.1305) was filed on December 6, 2002, and published on June 10, 2004. I understand that it is therefore prior art to the 645 Patent at least under pre-AIA 35 U.S.C. § 102 (e).

42. Krantz is generally directed to “computer-based systems and method for facilitating controlled access to protected networks and in particular, increasing the level of automation when provisioning and configuring a computer system to access a network.” Ex. 1305, [0015].

43. More specifically, Krantz discloses a system in which a device attempts to access a protected network. *See, e.g.*, Ex.1305, [Abstract]. To gain access to such a protected network, Krantz includes a server which can authorize at least partial access to the protected network resources based on credentials supplied by the user. *Id.*, [0016]. An embodiment of this network architecture is depicted below with respect to Fig. 2, which I have annotated:



Krantz, Fig. 2 (Ex.1305) (Annotated)

44. As seen above, and described in Krantz, the invention can utilize Virtual Local Area Network (“VLAN”) techniques. *Id.*, [0057]. Moreover, Krantz discloses that the network architecture is logical in nature and can connect sets of disparate computer systems together into one protected network. For instance, it discloses that “network architecture 200 data routing device 214 is connected to network 213, server 215, and access point 209 by corresponding logical communication links 233, 234, and 235 respectively. Network 213 can be virtually any type of network, such as, for example, a corporate or enterprise-wide network

... [and] data routing device 214 can be a device capable of grouping computer systems together in a single broadcast domain based on criteria other than physical location.” *Id.*, [0060]. As shown above with respect to Fig. 2, Access Point 209 utilizes ports which can be configured for data transfer in particular ways. *Id.*, [0062].

45. Krantz discloses that a client (access device) requesting access to the network can send credentials to a server to authenticate itself. *Id.*, [0016]. Based upon the credentials supplied, the server “may authorize the client for at least partial access.” *Id.* This can include “at least partial access to [a] single network, such as, for example, by authorizing client access to selected computer systems or modules on the single network.” *Id.*, [0017]. Authentication may be made by authentication module 217 in server 215. Ex.1305, [0073], [0076]. The server is also responsible for ensuring that the client device adheres to certain configuration settings—including the established encryption protocol—for gaining access to the network authorized to be accessed. Ex.1305 [0110]-[0111].

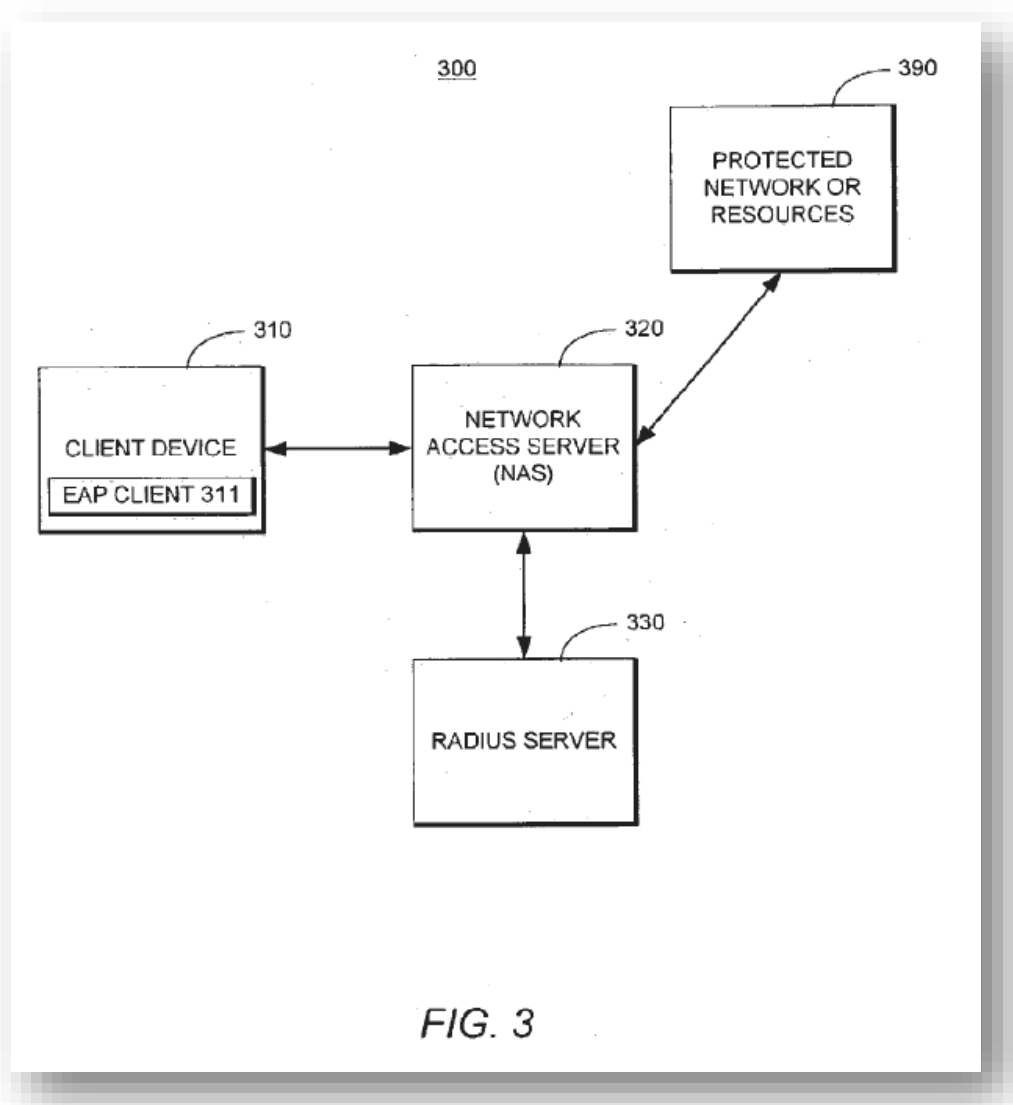
B. Summary of Herrmann

46. U.S. Patent Pub. No. 2004/0107360 to Herrmann et al. (Ex.1306) was filed on March 13, 2003, published on June 3, 2004, and claims priority to U.S. Provisional Application No. 60/430,458, which was filed on December 2, 2002. I

understand that it is therefore prior to the 645 Patent at least under pre-AIA 35 U.S.C. § 102 (e).

47. Hermann is directed to enforcing security policies when providing access to a network. Hermann describes that if a client device connected to a network . . . [and] is infected with a virus or worm, it may infect other machines on the same network. Ensuring that devices connected to the network are running current anti-Virus programs is particularly important, as virus suppression methods are very time sensitive. Ex.1306 [0012].

48. As a solution, Hermann proposes the collection of information from a client device which is used during the authentication process to enforce network security policies. For example, Herrmann teaches a system in which a client device attempts to access a network and information is collected to determine whether the client can gain access to protected network resources. Ex.1306, [Abstract], [0063]. The network access system is described with respect to Figures 3 and 4. The basic architecture of the system includes at least a client access device, network access server, and protected network and/or resources are shown below in Figure 3 of Herrmann. Ex.1305, [0018], [0053], [0063].

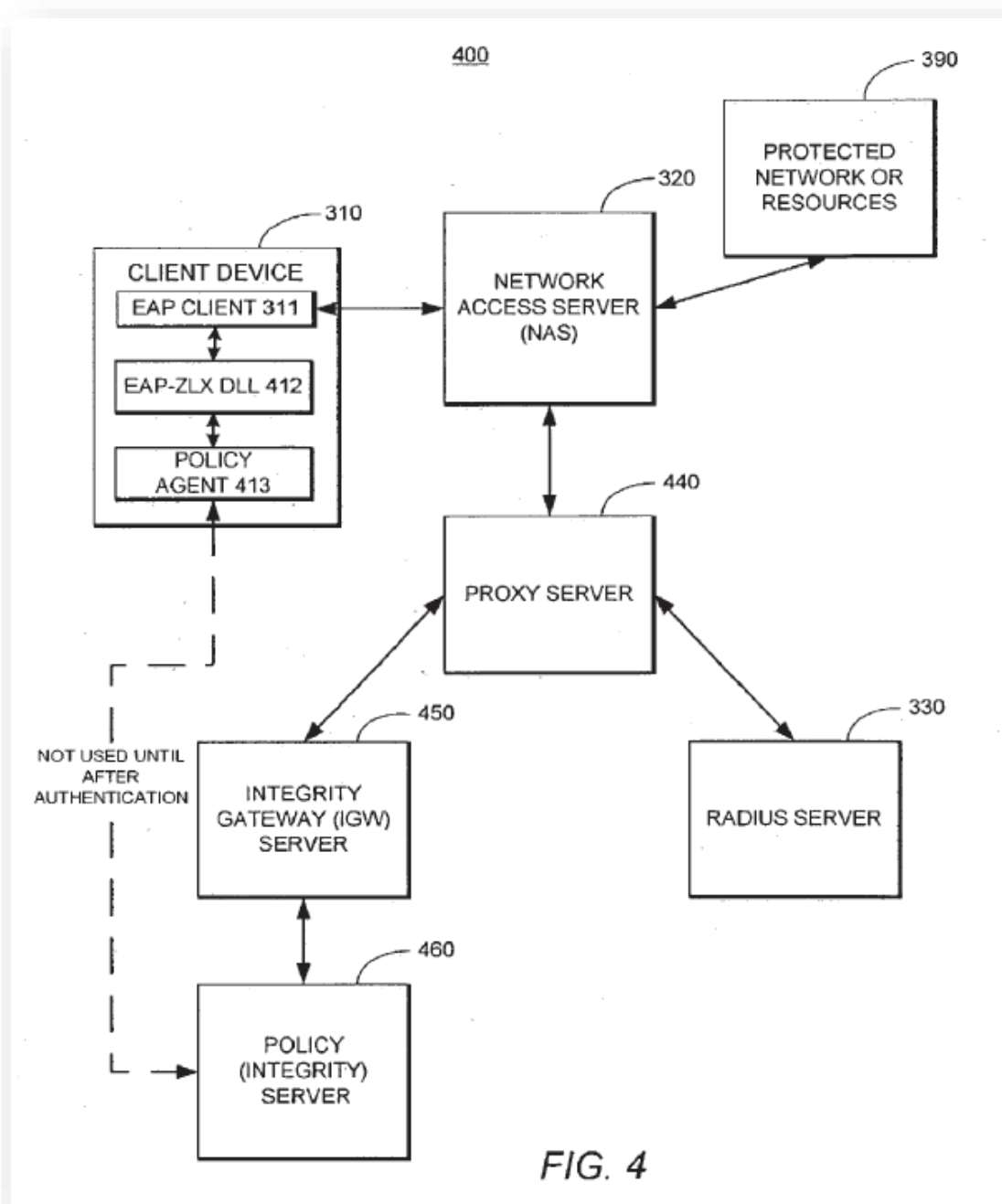


Herrmann, Fig. 3 (Ex.1306)

49. More specifically, Fig. 3 of Herrmann depicts a “network access server environment 300 ... which includes a RADIUS server providing authentication services.” *Id.*, [0053]. As shown above, “a client device 310 requesting access to a protected network 390 (e.g., the Internet, a corporate LAN or other resources) typically connects to a network access server (NAS) 320 using

client software and/or hardware such as a VPN client ... The NAS 320 acts as an access point (i.e., gateway) to a group of resources or collection of data (e.g., the protected network or resources 390) ...” *Id.*

50. Similar to Fig. 3, in the preferred embodiment of Fig. 4, “a client device 310 connects to the NAS 320 to access the protected network or resources 390.” *Id.*, [0062]. However, the system, includes an additional “policy (integrity) server 460” ... which “ensures that all users accessing a particular system or network comply with specified security policies, including access rights and cooperative anti-virus enforcement” and can deny network access for access devices that are not in compliance. *Id.*, [0067].



Herrmann, Fig. 4 (Ex.1306)

51. Fig. 4 of Herrmann depicts a “policy agent (or integrity agent) 413 to retrieve information about the current security policy in operation on the client device 310. *Id.*, [0064]. This allows “the policy server 460 [to] evaluate whether

Ex.1303

or not the client device 310 has a correct, up-to-date version of the applicable security policy.” *Id.*, [0067].

C. The Combination of Krantz and Herrmann

52. It would have been obvious to include Herrmann’s techniques related to enforcing security policies (including checking for anti-virus protection at the client) and obtaining system data from an access device in the system of Krantz for the reasons explained below. In particular, the Krantz system already discloses that “access point 209 may block [] protocols ... from being transferred **until client 205 is authenticated (and authorized)**. Ex.1305, [0072]. Specifically, Krantz explains that as part of its security policies, client devices are authenticated, (Ex.1305, [0067]-[0074], [099]) and users of client devices are authenticated, Ex.1305, [0074], [0076]. Ex.1305, [0076].

53. A POSITA would recognize that checking for anti-virus software as part of a security policy would merely be an additional item for server 215 to check in the process already disclosed by Krantz. Ex.1305, [Fig. 3], [0068]-[0076] (describing the steps of authenticating client devices and users). Like Krantz, Herrmann discloses a “policy server 460 [which] may enforce a variety of different types of security policies or rules.” Ex.1306, [0068]. One such rule, to be authorized for network access, would be “a policy requiring that a **specific virus protection program** is operational on each client device that is connected to a

network.” Ex.1306, [0012]. Herrmann expressly recites that the policy server determines whether the client device 310 is in compliance with applicable security policies. Ex.1306 [0066].

54. A POSITA would also be motivated to modify the teachings of Krantz to include determining whether the requirements of a security policy that include checking for virus protection are satisfied as taught by Herrmann for several reasons. As explained in detail below, Krantz already describes a security policy that includes requiring authentication of a client device and users. Krantz also teaches providing a configuration file that specifies the proper encryption protocol (and other necessary configuration information), to ensure the client device is configured to for compatible operation with the network. Herrmann discloses that it is “particularly important” that devices connected to the network are running current anti-virus programs, and that one way of ensuring this is implementing a “policy requiring that a **specific virus protection program.**” Ex.1306, [0068]; *see also*, Ex.1306, [0012] (noting it is critical that all machines in a network maintain updated antivirus software so the network is not infiltrated by newly released viruses). Herrmann expressly recites that the policy server determines whether the client device 310 is in compliance with applicable security policies, which would include specific virus detection. Ex.1306, [0066]. Thus, a POSITA would be motivated to modify the teachings of Krantz to include in a security policy

requirements that a specific virus protection program be utilized, and to make a determination if the client device satisfies the applicable security policies before allowing the client device to access protected networks in order to prevent viruses from infecting the protected network and to ensure network compatibility

55. Krantz's server 215, which compares the credentials of a user against a database (e.g., user-identifier and password), as modified by Herrmann, would additionally compare a virus protection program on an access device against approved antivirus programs. Information regarding approved antivirus software and approved connected devices could be stored in the User Database or an additional Database could be added.

56. The 645 Patent itself confirms my opinion that a POSITA would recognize the need for the authentication module of Krantz to check for up-to-date and approved antivirus programs. In the background section of the 645 Patent, it discusses GateKeeper 110 (Ex.1301, 2:30-31) and articulates that "GateKeeper 110 may make certain that Access Device 160 has up-to-date virus software and encryption protocols as proscribed by [a] security policy." Ex.1301, 2:23-26.

57. Both Krantz and Herrmann are analogous art to the 645 Patent for at least two reasons. *First*, Krantz and Herrmann are both directed toward the same field of endeavor of the 645 Patent—network security. Ex.1301, 1:21-22 ("The invention is in the field of computing systems and more specifically in the field of

network security.”); Ex.1305, Abstract (“A computer system attempts to authenticate with a server to gain authorization to access a first network.”); Ex.1306, Abstract (“A system and methodology for policy enforcement during authentication of a client device for access to a network is described.”).

58. *Second*, both Krantz and Herrmann are reasonably pertinent to a problem described by the 645 Patent: the reliance “on mere user authentication” and the need to “assure that **an access device** used by the authenticated user meets network security policies” such as current anti-virus software. *See, e.g.*, Ex.1301, 1:46-48; *see also, id.*, 1:53-60 (“A greater level of access control may be achieved by including a gatekeeper between the secure network and the access device. The gatekeeper is configured to ensure that the access device conforms to a predetermined security policy.”).

59. Krantz discloses a security policy beyond user authentication and encryption protocols. For instance, “[a]fter an EAP type is selected . . . **computer systems can attempt to authenticate with one another** through the transfer of EAP messages . . . in accordance with the selected EAP type. For example, when EAP-TLS is selected, **client 205 may communicate with server 215** through a series of EAP messages in accordance with EAP-TLS **so as to be authenticated and possibly authorized to access resources located on VLAN C.**” Ex.1305, [0068]. The use of EAP-TLS (*e.g.*, Ex.1305, [0067]-[0074], [0100]) was a known

authentication protocol that would be used to authenticate client devices. Ex.1316 [RFC2716] at 1, 2. Krantz further discloses that the security policy can include “a configuration sub-file to indicate whether or not **operating system credentials** should be used for authentication.” *Id.*, [0099].

60. Krantz discloses that “[i]n some embodiments, a server on a network can provide a client with limited access to the network. For example, a server located on network 413 may implement IP filtering to provide client 405 with limited access to network 413. Included in the limited access can be access to computer systems or modules that store **provisioning files needed for gaining full access to network 413.**” *Id.*, [0056].

61. Herrmann similarly discloses a “policy agent (or integrity agent) 413 to retrieve information about the current security policy in operation on the client device 310. The information collected by the policy agent 413...” Ex.1306, [0064]. “In response to a message from the IGW server 450 (or a subsequent message from the policy agent 413 on the client device 310), the policy server 460 evaluates whether or not **the client device 310 has a correct, up-to-date version of the applicable security policy.**” *Id.*, [0067]. The policy server is notified by the client, containing policy agent 413, through periodic status messages called “IA_HEARTBEAT(s)” in order to authenticate the access device. *See, id.*, [0080], [0098].

62. Herrmann further discloses that “[n]ew viruses are frequently released that cannot be identified using older anti-virus engines and definition files. It becomes critical, therefore, to promptly update **anti-virus applications on all machines in a network** in a timely fashion before the network is infiltrated by a newly released virus.” *Id.*, [0012]. Herrmann also discloses that “a network administrator may require all users accessing the network to implement a **virus definition update (e.g., DAT file)** that is targeted at a particular virus.” *Id.*, [0071].

63. It would have been obvious to combine Herrmann’s techniques related to determining whether an access device is authorized for network access (including checking its antivirus software) with the authentication and authorization system of Krantz. Each performs the same function (authentication and authorization to access network resources) with each yielding not more than what one would expect from such an arrangement (automated network access techniques including a higher level of security by checking antivirus software), as the Krantz/Herrmann system demonstrates.

64. It would have been obvious to combine Herrmann with Krantz because a POSITA would have been motivated to do so based on the teachings of Krantz. For example, Krantz explains that its system relates to a server that “includes the functionality to authorize clients to access a second network”

EX.1305, [0016]. EX.1305, [0002]. Similarly, Herrmann explains that “[t]he present invention relates to ... policy enforcement on **computer systems** connected on one or more networks...” EX.1306, [0003].

65. A POSITA would have also been motivated to apply Herrmann’s technique of checking connecting computer systems for antivirus software as it would not only provide added security but increase automation when connecting to a network. *Id.*, [0068] (“a system administrator may establish a policy requiring that **a specific virus protection program is operational on each client device** that is connected to a network. **The policy server would then evaluate whether or not each client device was in compliance** with the specified policy before approving the client device for network access.”). These are two stated goals of Krantz. EX.1305, [0014] (“Therefore what would be advantageous are systems, methods, computer program products, and data structures for increasing the level of automation when provisioning and configuring a computer system to access a network.”); *id.*, [0019] (“If credentials do not authorize full access ... the server can respond by sending a notification ... [including] conditions (e.g., signup, renewal etc.) a user must satisfy to become authorized for full access to resources ...” Thus, A POSITA would have recognized that Herrmann’s disclosure of a policy server checking that a computer device complies with, e.g., an antivirus security policy, would be complimentary to the system of Krantz, which seeks an

automated process for providing network security. In addition, a POSITA would have had a reasonable expectation of success in implementing these techniques into a combined Krantz/Herrmann system as doing so would involve minimal changes to the existing Krantz system.

IX. KRANTZ, OR KRANTZ AND HERRMANN RENDER OBVIOUS CLAIMS 1-2 AND 5-6

A. Detailed Analysis of the Challenged Claims

1. Claim 1

a. [1.0] A network access control system, comprising:

66. Krantz discloses “[a] network access control system.”

67. The “*network*” of the 645 Patent can include multiple networks as opposed to single network implementation. For instance, it explains that “[i]n various embodiments, Protected Network 200 includes a corporate network, telephone network, private network, local area network, wide area network, wireless network, communication network, government network, university network, and/or the like.” Ex.1301, 6:5-9.

68. The network disclosed in Krantz is a “*network*” as described in the 645 Patent. As will be explained in the following paragraphs, VLAN B and VLAN C of Krantz (e.g., as depicted in Fig. 4) constitute a protected network consistent with the “*network*” of the 645 Patent. Krantz explains that the disclosed networks “can be implemented using any number of techniques, such as, for

example, **Virtual Local Area Networks (“VLANs”), Internet Protocol (“IP”) filtering, Virtual Private Networks (“VPNs”), or use of IPSecurity (“IPSec) protocols.”** Ex.1305, [0017]. Each of these techniques are methods of creating a “*network*” utilizing a “*network access control system*” because they facilitate, for example, the access control capabilities described in Krantz. Ex.1305, [0017] (explaining that the above techniques can be used such that “the server (based at least in part on received credentials) may authorize the client for at least partial access to the single network, such as, for example, by authorizing client access to selected computer systems or modules on the single network.”).

69. Krantz discloses network access protocols whereby “[a] server includes the **functionality to authorize clients to access a second network ...**” Ex.1305, [0016] (i.e., *network access control*). For example, Krantz explains that “[a] client sends credentials to the server to attempt to authenticate with the server. Based on the credentials, the **server may authorize the client for at least partial access to the second network** and/or may deny access to the second network and **authorize the client for at least partial to access to the first network.**” Ex.1305, [0016].

70. Based on Krantz, a POSITA would further understand that combining the first and second networks into a logical “*network*” would be done either physically or logically, although Krantz focuses on logical representations:

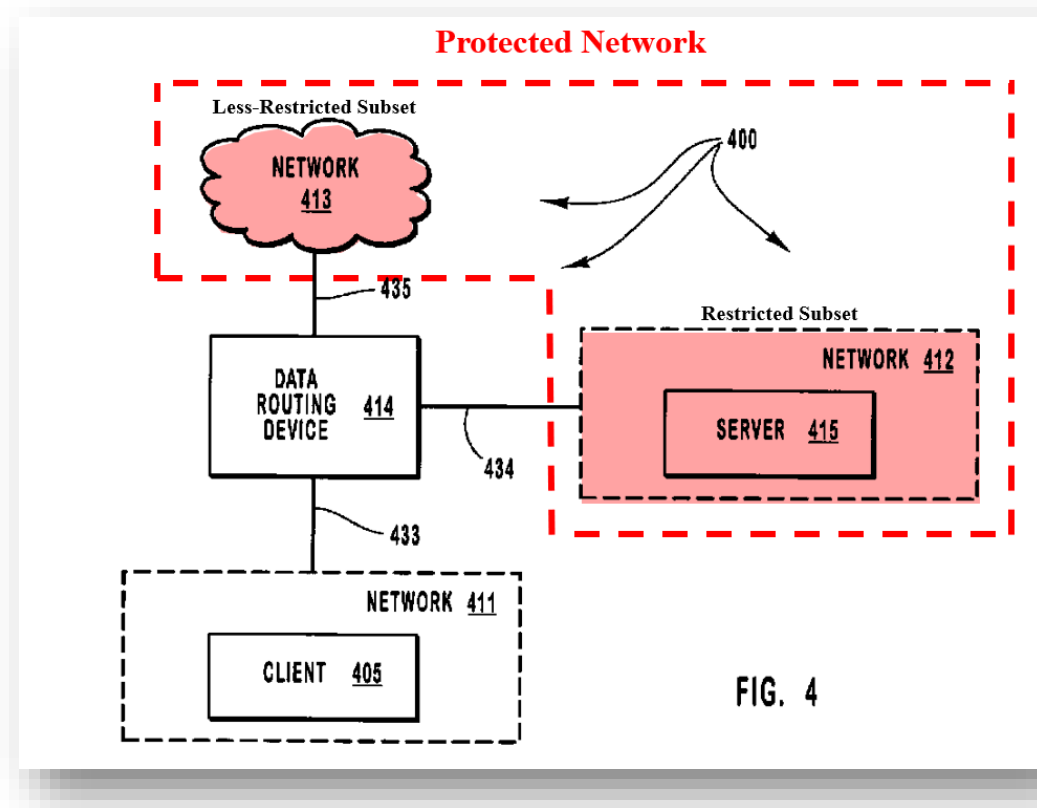
[A] “logical communication link” is defined as any communication path that can enable the transport of electronic data between two entities such as computer systems or modules. The actual physical representation of a communication path between two entities is not important and can change over time. A logical communication link can include portions of . . . a local area network (e.g., an Ethernet network), a wide area network, the Internet, combinations thereof, or portions of any other path that may facilitate the transport of electronic data. . . . Logical communication links may also include portions of a virtual network, such as, for example, Virtual Private Network (“VPN”) or a Virtual Local Area Network (“VLAN”).

Ex.1305, [0038].

71. In fact, combining two (or more) physical networks together was a common technique in networking that was well-known to POSITAs at the time. *See, e.g.*, Ex.1311, 223 (defining a “logical network” as “[a] network as it appears to the user. In fact, the network could be composed of two or more physical networks, or portions of these, that are linked and coordinated in a such a way that they appear to be a unit.”); Ex.1312, 317 (explaining that the “difference between a network as logical or physical is sometimes subtle because the physical network (the actual layout of hardware and cabling) doesn’t necessarily resemble the logical network (the path followed by transmissions.”). Similarly, Krantz explains that “[l]ogical communication links **can include hardwired links, wireless links, or a combination**” thereof. Ex.1305, [0038].

72. The 645 Patent describes some of the “*network’s*” characteristics. For example, the 645 Patent explains that a *network* may be “**physically or logically divided** into a restricted subset and one or more less-restricted subsets.” Ex.1301, 2:50-52. The 645 Patent also explains that such restricted and less-restricted subsets are typically “characterized by Virtual Local Area Networks (e.g., VLANs),” (Ex.1301, 2:64-3:2), and further explains that the “protected *network*” may include “access restrictions.” Ex.1301, 6:21-37.

73. The protected “*network*” described in Krantz is depicted in Figures 4 and 2, which are reproduced and annotated below.



Ex.1305 (Krantz), Figure 4 (annotated).

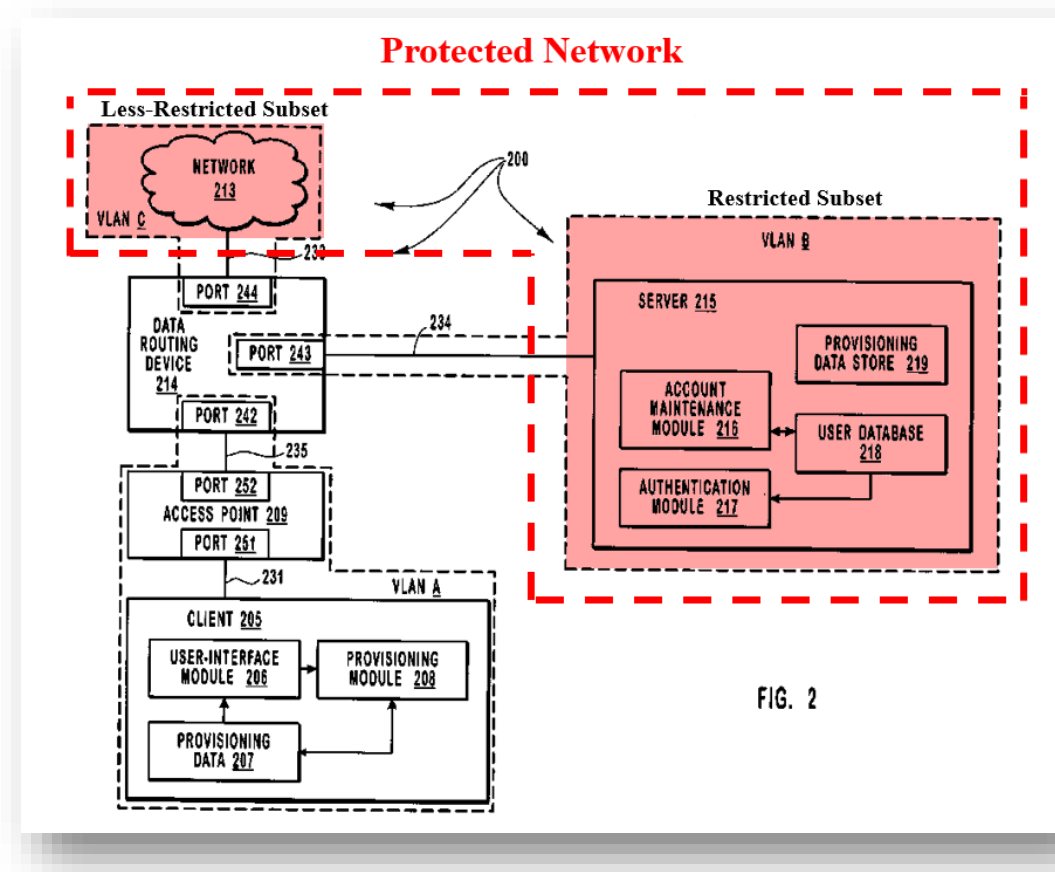


FIG. 2

Ex.1305 (Krantz), Figure 2 (annotated).

74. Krantz describes Figure 4 as “illustrat[ing] generally an example of a network architecture 400 that can provide access to a first network while restricting access to a second network.” Ex.1305, 5:65-6:2. Krantz further explains that “in some embodiments, access to resources located in network 413 can be restricted by the use of Virtual Local Networks (“VLANs”).” Ex.1305, [0052]; *see also* [0054] (similarly explaining that access to resources located on network 413 can also be restricted using VPNs) (i.e., utilizing *network access control*). Figure 2 provides an example of such an embodiment implementing the general architecture of

Figure 4 and “using VLAN techniques,” which “can facilitate increasing the level of automation when provisioning a client.” Ex.1305, [0057]. Figure 2 and its description thus provide certain specific techniques to be implemented in Krantz’s networking architecture. *See generally* Ex.1305, Figures 2 and 4 and related descriptions. Accordingly, and as depicted above, the combination of Network 413 and Network 412 (also annotated as VLAN C and VLAN B) constitute a single logical “*network*.” Given the similarity in disclosure between the embodiment described in Figure 2 and the 645 Patent, my analysis will focus on Figure 2 and its corresponding description, although I will refer to the general architecture represented in Figure 4 and the corresponding description as necessary.

75. To the extent the “*network*” is required to be a single network, Krantz explains that its “*access*” control techniques can also be used “**within a single network.**” Ex.1305, [0017] (“In some embodiments, similar techniques can also be used within a single network. In these embodiments, the server (based at least in part on received credentials) may authorize the client for at least partial access to the single network, such as, for example, by **authorizing client access to selected computer systems or modules on the single network.**”); *see also* Ex.1305, [0055].

76. As Krantz explains with respect to the single network embodiment, the disclosed system “**may authorize the client for at least partial access to the single network**” (“*network access control*”) and as explained in more detail below, one example of granting such “*access*” in Krantz is “by authorizing client access to selected computer systems or modules on the single network.” Ex.1305, [0017]; *see also* Ex.1305, [0016], [0071] [0074]-[0076], e.g., [0068] (“[a]fter an EAP type is selected . . . **computer systems can attempt to authenticate with one another** through the transfer of EAP messages . . . in accordance with the selected EAP type. For example, when EAP-TLS is selected, **client 205 may communicate with server 215** through a series of EAP messages in accordance with EAP-TLS **so as to be authenticated and possibly authorized to access resources located on VLAN C.**”) *see also* Ex.1305, [0104], [0107]-[0109].

77. Accordingly, Krantz discloses “[a] *network access control system.*”

- b. **[1.1] an authentication module stored in memory and executable by a processor to authenticate a device seeking network access based at least on authentication information received from the device,**

78. Krantz discloses this limitation. As an initial matter, the components of the system described in the 645 Patent can be logical in nature and thus reside on the same or different physical machines. For instance, the 645 Patent articulates that “[a]ccording to an exemplary embodiment, client filters (CF) and server filters (SF) are added along with an agent and GK [i.e., gatekeeper] to perform audits ...

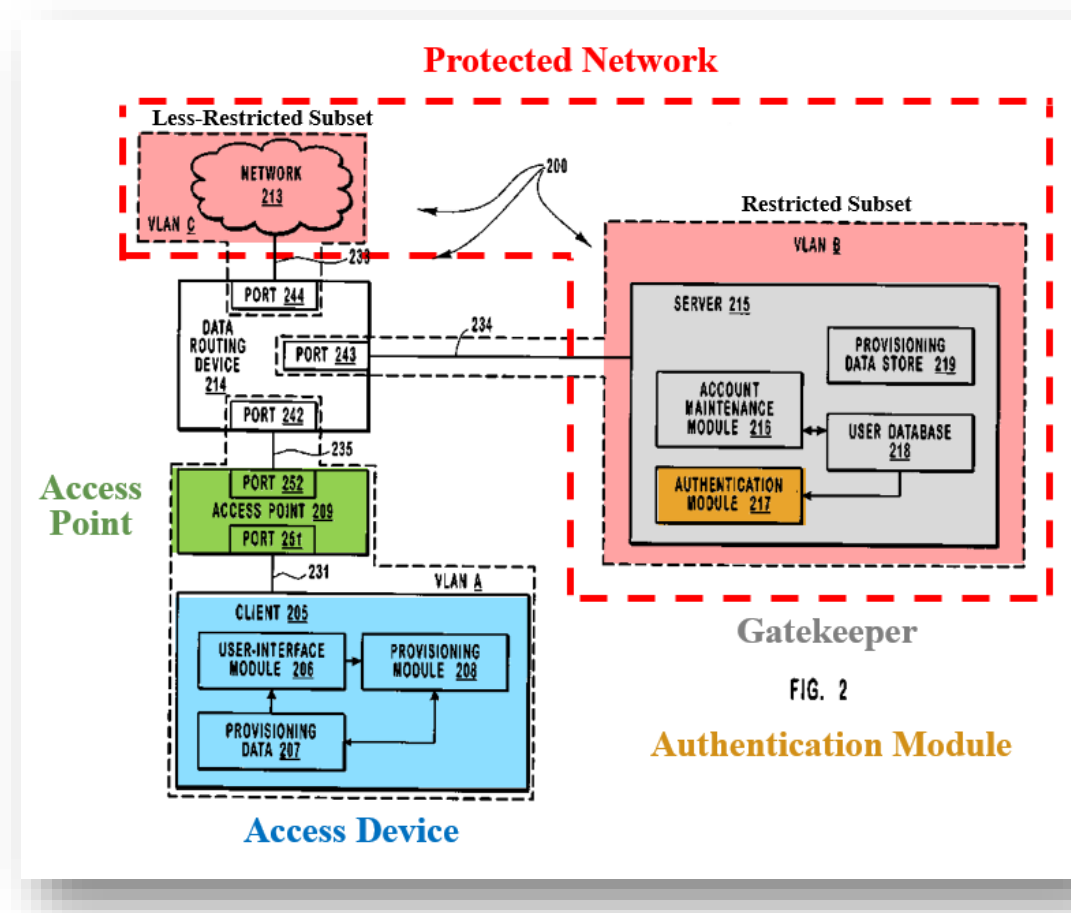
There may be one or more instances of the CF and SF, which correspond to the separate EAP Authentication Modules (EAPM) that are supported.” 645 Patent, 14:22-27. Notably, “the **agent and [gatekeeper] are logical components**. The functionality of the agent and [gatekeeper] **can be embedded into the CF and SF, respectively, or ... may reside on a separate server.**” 645 Patent, 14:27-30.

79. Similarly, Krantz discloses a logical structure in which the “*authentication module*” can be stored on the same server as the “*gatekeeper*” or a separate server. *See, e.g.*, Ex.1305, [0058] (“In some embodiments, server 215 logically represents a Remote Authentication Dial-In User Service (‘RADIUS’) server. That is, **although visually depicted by a single system, server 215 may include** a separate Network Access Server (‘NAS’), **a separate authentication server**, and a separate shared accounting server.”).

80. Krantz discloses “*an authentication module stored in memory and executable by a processor.*” As explained in Krantz, “if an authenticated user of client 205 is authorized to access resources located **on VLAN C** (e.g., the user is up to date on payments for their account), client 205 can then be authorized to access resources located on VLAN C.” Ex.1305, [0076]. Specifically, Krantz discloses authentication techniques utilizing “*an authentication module.*” Ex.1305, [0076] (“[w]hen credentials are sent to server 215, **authentication module 217** [“*authentication module*”] **can receive the credentials (e.g., user-identifier and**

password) and compare the credentials to entries in user database 218.”).

Krantz explains that after the client is authenticated, “client 205 can then be authorized to access resources located on VLAN C.” Ex.1305, [0076]; *see also* Ex.1305, [0031]. The authentication module 217 (“*authentication module*”) of Krantz is shown as part of Server 215, with respect to Fig. 2, below.



Ex.1305 (Krantz), Figure 2 (annotated) (notably, as discussed *supra*, and consistent with the 645 Patent disclosures, the authentication module and gatekeeper can both be considered logical components and can reside as part of the same server 215).

81. Because **authentication module 217** is part of a server (server 215), a POSITA would have understood that the “*authentication module*” is “*stored in memory and executable by a processor*” based on corresponding teachings of Krantz. For instance, Krantz explains that “a computer system [] may implement the principles of the present invention ... defined broadly as **any hardware component or components that are capable of using software** to perform one or more functions.” Ex.1305, [0047]. “Examples of computer systems include desktop computers, laptop computers, Personal Digital Assistants (‘PDAs’) ... **multi-processor systems, microprocessor-based** or programmable consumer electronics, network PCs, minicomputers, mainframe computers, or any other system or device that has processing capability.” Ex.1305, [0047]; *see also*, Ex.1305, [0013], [0037] (“**Computer-executable instructions comprise**, for example, any instructions and data which cause a general-purpose computer system, special-purpose computer system, or special-purpose **processing device** to perform a certain function or group of functions.”), [0042] (“[A]n exemplary system for implementing the invention includes a general-purpose computing device in the form of a conventional computer 120, **including a processing unit 121, a system memory 122**, and a system bus 123 that couples various system components **including the system memory 122 to the processing unit 121.**”),

[0044]. Based on these disclosures, the “*authentication module*” is “*stored in memory and executable by a processor*” as a POSITA would expect.

82. Indeed, with respect to modules, such as, e.g., authentication module 217, Krantz explains what would have been common knowledge to a POSITA at the time, that it is stored in memory (for execution by a processor as I described in the preceding paragraph):

[P]ortions of such modules and portions of associated program data may be included in operating system 135, application programs 136, **program modules 137 and/or program data 138, for storage in system memory 122**. When a mass storage device, such as, for example, magnetic hard disk 139, is coupled to computer 120, **such modules and associated program data may also be stored in the mass storage device**. In a networked environment, program modules depicted relative to computer 120, or portions thereof, can be stored in **remote memory storage devices, such as, for example, system memory and/or mass storage devices** associated with remote computer system 149a and/or remote computer system 149b. Execution of such modules may be performed in a distributed environment as previously described.

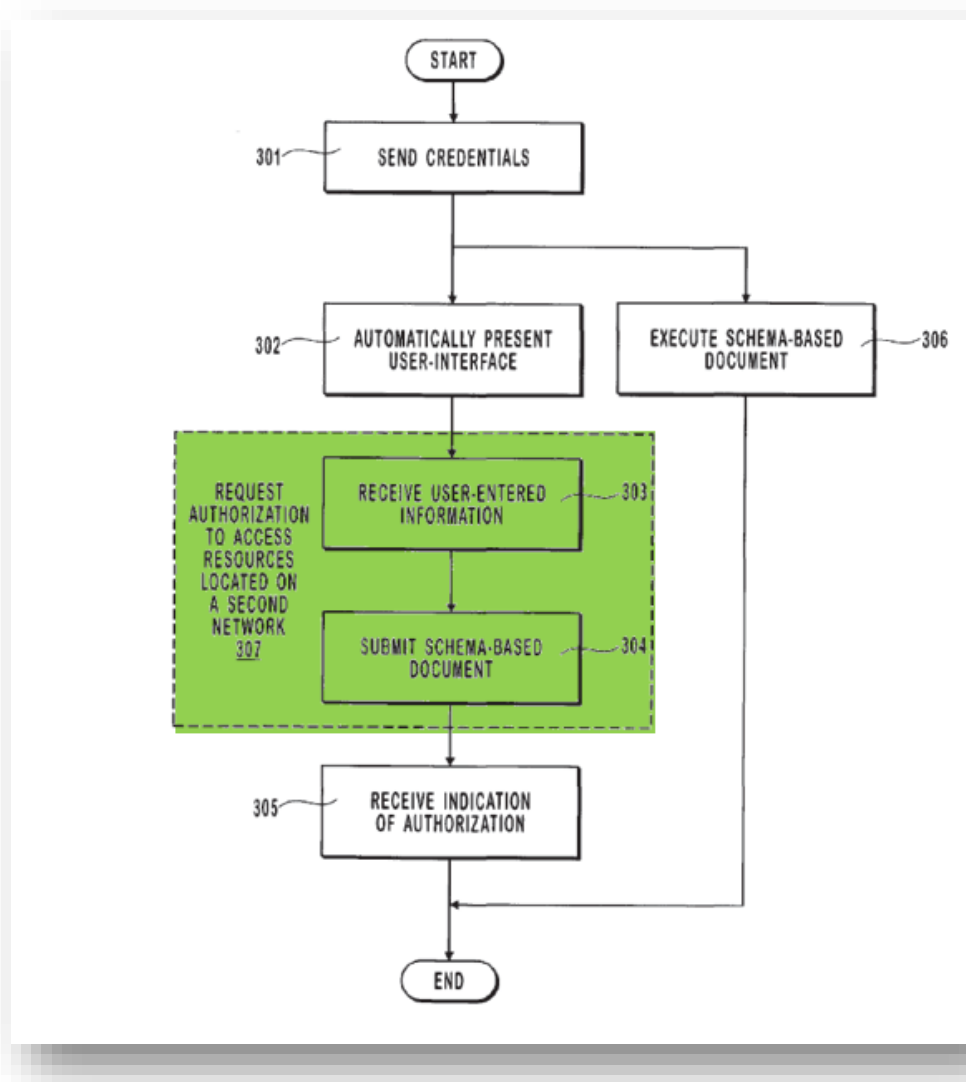
Ex.1305, [0049]; *see also*, Ex.1305, [0041], [0042], [0045], [0048] (“In a distributed system environment, **program modules may be located in both local and remote memory storage devices.**”).

83. The “*authentication module*” of Krantz is used “*to authenticate a device seeking network access based at least on authentication information received from the device.*” For example, Krantz discloses that “a server has the functionality to authorize access” to a network and/or selected elements within a network after the client 205 [“*device seeking network access*”] sends credentials to the server [“*based at least on authentication information received from the device*”]. See, e.g., Ex.1305, [0016], [0071]. Krantz discloses that the “server **(based at least in part on received credentials) may authorize the client for at least partial access** to the single network.” Ex.1305, [0017]; see also [0068]-[0076], [0099], [0104], [0107]-[0109] (collectively describing techniques for requesting and “*authentica[ting] a device*” seeking access in a VLAN environment). Other credentials provided by Krantz’s “*device*” may include credentials based on the “*device*,” such as operating system or other credentials used to authenticate a device. Ex.1305, [0067]-[0074], [0099]-[0100].

84. Such credentials, i.e., “*authentication information*,” “can include sending an Extensible Authentication Protocol (“EAP”) Response/Identity message from the client to the server. If the client is attempting to connect to a network via an access point, the EAP Response/Identity message may be encapsulated within some other type of message that can be transferred through the access point.” Ex.1305, [0018]. Other credentials or “*authentication information*” provided by

Krantz's "*device*" may include, for example, a "user-identifier[s] and password[s]." Ex.1305, [0076].

85. Krantz also explains that Figure 3 "is a flow diagram illustrating a method 300 for provisioning a computer system," and that the disclosed "method 300 [is] described with respect to components depicted in network architecture 200." Ex.1305, [0070]. Krantz discloses that "the method 300 includes a functional, result-oriented step for **requesting authorization to access resources located on a second network (step 307)** [*i.e.*, "*device seeking network access*"]." Ex.1305, [0104]; *see also* Ex.1305, [0071] (describing the client sending credentials to server to authenticate). Krantz further explains that "[s]tep 307 may include any corresponding acts for accomplishing the result of requesting authorization to access resources located on the second network." Ex.1305, [0104].



Ex.1305 (Krantz), Figure 3 (annotated).

86. In the example depicted in Figure 3, Krantz explains that step 307 “includes [the] corresponding act of receiving user-entered information (act 303). A presented user-interface can be used to receive user-entered information (e.g., user-identifier, password, name, address, credit card information, etc.) from a user at client 205.” Ex.1305, [0104].

87. The “device”—i.e., “Client 205”—“seeking network access” in Krantz is depicted in Figure 2 in blue and annotated below:

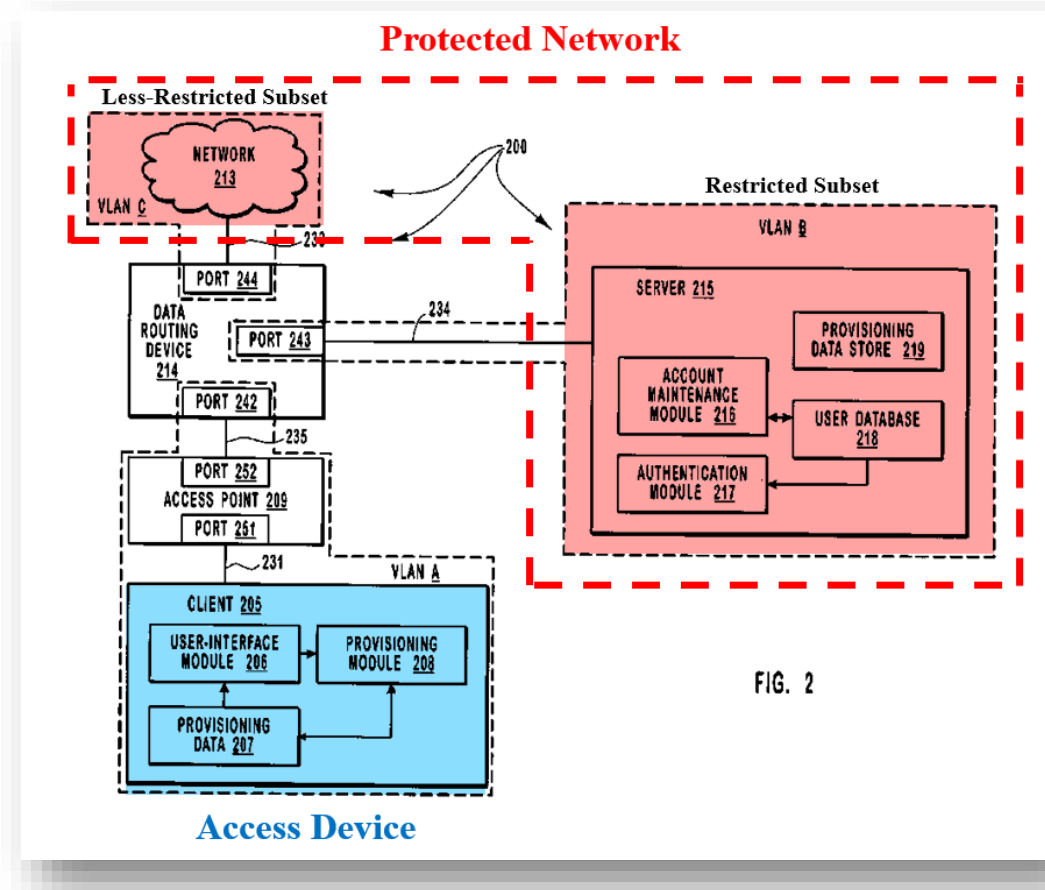


FIG. 2

Ex.1305 (Krantz), Figure 2 (annotated).

88. Moreover, as described above, *see* § [1.0], the combination of second network VLAN C and first network VLAN B constitute a “network.” Krantz discloses that “[b]ased on [client] credentials, “the server **may authorize the client for at least partial access to the second network [i.e., VLAN C] and/or may deny access to the second network** and authorize the client for at least partial to

access to the first network” (*i.e.*, “*authenticate a device seeking network access based at least on authentication information received from the device.*”) Ex.1305, [0016]; *see also* Ex.1305, [0017] (explaining that “the server (based at least in part on received credentials) may authorize the client for at least partial access to the single network, such as, for example, by authorizing client access to selected computer systems or modules on the single network.”). Similarly, Krantz explains that its access methods “include[] a functional, result-oriented step **for requesting authorization to access resources located on a second network.**” Ex.1305, [0104].

89. Accordingly, Krantz discloses “*an authentication module stored in memory and executable by a processor to authenticate a device seeking network access based at least on authentication information received from the device.*”

c. [1.2] a gatekeeper stored in memory and executable to

90. Krantz discloses “*a gatekeeper stored in memory and [which is] executable.*” As explained above, *see* [1.1], Krantz discloses “*memory*” as well as a “*processor*” capable of executing modules stored “*in memory.*” As also discussed in [1.1], the components of the system described in the 645 Patent can be logical in nature. Notably, “the agent and **[gatekeeper] are logical components.** The functionality of the agent and [gatekeeper] **can be embedded into the CF and SF, respectively, or ... may reside on a separate server.**” 645 Patent, 14:27-30.

Because of this disclosure, a POSITA would have recognized that the gatekeeper can include the authentication module, because these components are logical in nature and work together to authorize and authenticate devices. This is particularly the case for claim 1, where the gatekeeper **is not required to reside on a separate server** from the authentication module, as opposed to the more limiting claim 4 which does include that requirement. *See*, Ex.1301, claim 4 (“The network access control system of claim 1, wherein the **gatekeeper is hosted at a computing device distinct from a computing device hosting the EAPM [authentication module]**, ESL, and SF.”).

91. Krantz includes such a “*gatekeeper*,” labeled as server 215, shown below with respect to Fig. 2. As an initial matter, the Abstract of the 645 Patent notes that “[t]he gatekeeper⁵ is configured to enforce a security policy ... before granting access to a less-restricted subset of the computing network.” Ex.1301, Abstract.

⁵ Notably, Applicant admits that this Gatekeeper is prior art. *See*, Ex.1301, 2:28-30 (“GateKeeper ... [is] practiced in the prior art.”).

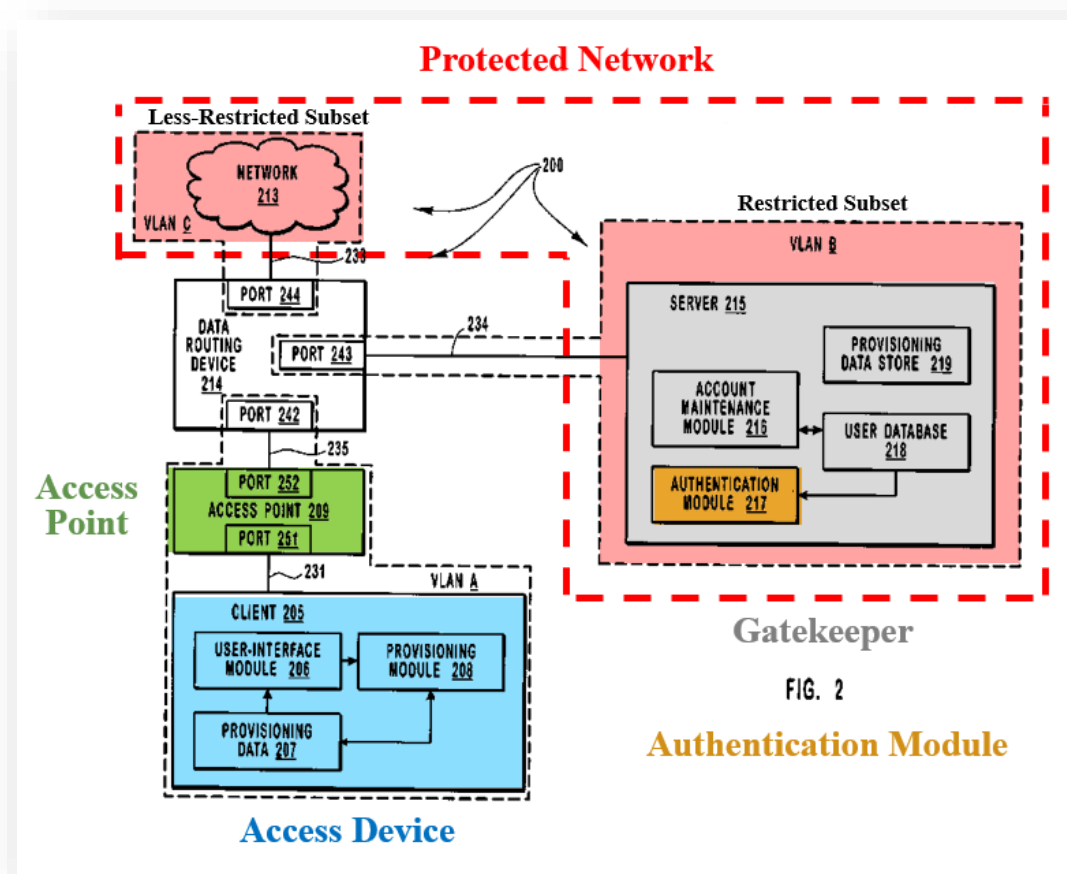


FIG. 2

Authentication Module

Ex.1305 (Krantz), Figure 2 (annotated).

92. Furthermore, server 215 (the “gatekeeper”) is “stored in memory and executable” because Krantz explains that server 215 may optionally be comprised of a variety of services having a number of program modules, including a **logical** authentication service. Initially, a Skilled Artisan would understand that a “server” in the context of “the internet or other network,” is “a computer or program that responds to commands from a client.” Ex.1311, 474. For example, Krantz explains that “[i]n some embodiments, server **215** logically represents a Remote

Ex.1303

Authentication Dial-In User Service (“RADIUS”) server. That is, although visually depicted by a single system, server **215** may include a separate Network Access Server (“NAS”), a separate authentication server, and a separate shared accounting server. The servers logically represented by server **215** can be configured to communicate using the RADIUS protocol and, more specifically, may support the RADIUS attributes EAP-Message and Message-Authenticator.” Ex.1305, [0058]. RADIUS was a well-known authentication software system at the time, *see, e.g.*, Ex.1320, 3 (“RADIUS is a client/server-based authentication software system); *see also* Ex.1301, 14:40-43, that would be “*stored in memory and executable.*”

93. Krantz also explains, “[s]erver 215 may have modules from a variety of different services **loaded into system memory** ... so as **to facilitate implementation** of the principles of the present invention.” Ex.1305, [0058]. A POSITA would understand that this means the disclosure of the gatekeeper in Krantz is stored in memory and executable based on the teachings of Krantz itself. For instance, Krantz teaches:

[T]he invention will be described in the general context of **computer-executable instructions, such as program modules, being executed by computers in network environments**. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. **Computer-executable instructions, associated data structures, and program modules represent examples of the**

program code means for executing steps of the methods disclosed herein.

Ex.1305, [0041].

94. Accordingly, Krantz discloses that “*a gatekeeper stored in memory and executable.*”

d. [1.3] formulate an audit request,

95. Krantz discloses that the gatekeeper “*formulate[s] an audit request.*”

96. As an initial matter, the 645 Patent broadly discloses that a “**security audit typically includes determining if Access Device 230 satisfies requirements of the security policy.** The security audit optionally includes active probing, scanning, and/or collection of third party data regarding Access Device 230. In some embodiments, **Authenticate User Step 440 is considered part of the security policy audit.**” Ex.1301, 11:67-12:6. The specification of the 645 Patent does not include the term “*formulate,*” but a POSITA would recognize the term’s plain and ordinary meaning in view of the specification and prosecution history to simply mean to create or generate. *See, e.g.,* Ex. 1306, [0079].

97. Krantz discloses that “[s]erver 215 [*‘gatekeeper’*] **may request that client 205 provide a password** [*‘an audit request’*] that is associated with a user-identifier. A user of client 205 can respond by providing a password to server 215. **It may also be that client 205 and server 215 exchange EAP-messages to pass certificates, keys, and supported cipher suites.** Depending on the EAP type,

other credential information may also be exchanged between client 205 and server 215.” Ex.1305, [0074].

98. Additionally, in the case of a denial of access to resources, “**server 215 can send an EAP-Notification ... [which] can [] include conditions a user must satisfy** [*‘audit request’*] to become authorized to access resources located on VLAN C (e.g., signup, renewal, etc.). If client 205 is to perform downloads by accessing a provided URI, access point 209 can enable the transfer of HTTP packets for port 251. Server 215 may send commands to access point 209 to cause access point 209 to enable the transfer of HTTP packets.” Ex.1305, [0078].

99. In order to make such a request to the access device, it is understood that the gatekeeper of Krantz must first “*formulate*” the “*request*.” This is formulated and transmitted within a packet. As one example, Krantz discloses:

Numeric Values can be included within an options field of LCP packets (e.g., Configure-Request, Configure-Ack, Configure-Nak and Configure-Reject packets) to negotiate the[] connection settings. Numeric values may be included in a type field (e.g., the numeric value 3 for authentication-protocol negotiation) within the options field of an LCP packet to indicate the type of configuration option being negotiated. Numeric values may be included in a data field (e.g., the hexadecimal value C227 to indicate EAP) within the options field of an LCP packet to provide corresponding data for the type of negotiation indicated in the type field. **When the configuration of EAP is indicated, further numeric values (e.g., the numeric value 13 to**

indicate EAP-TLS or the numeric value 25 to indicate PEAP) can be included in a LCP packet to indicate the specific EAP type of authentication that is desired [i.e., “*formulate*” the audit request].

Ex. 1305, [0067]. In addition, Krantz states that “[i]t may be that a particular EAP type is **deterministically selected** (e.g., by an administrator of **server 215** [“*gatekeeper*”] and/or a user of client 205) when both a client and a server support the particular EAP type.” Ex. 1305, [0068]. In other words, Krantz discloses that the “*gatekeeper formulates [the] audit request*” to be transmitted to the client, which can involve an EAP protocol such as EAP-TLS, as I will discuss in greater detail with respect to element [1.7].

100. Krantz similarly discloses that the “server (**based at least in part on received credentials**) may authorize the client for at least partial access to the single network.” Ex.1305, [0017]; *see also* [0068]-[0076], [0099], [0104], [0107]-[0109].

101. To the extent one may assert that Krantz does not expressly or implicitly disclose “*formulate*” with respect to the gatekeeper’s “*audit request*,” it would have been obvious that the gatekeeper of Krantz formulates (i.e., generates) such a request in view of Herrmann, which states:

When the access implementation module 553 on the IGW server 450 receives the response message from the policy server 460, the access implementation module 553 **formulates** a RADIUS Access-Accept

package **for transmission** by the proxy server 440. In the currently preferred embodiment, if the client does not have the correct security policy in place, the RADIUS packet that is **generated** for return includes a restrictive filter identifier that limits access for the session to a limited group of IP addresses (i.e., a defined “sandbox” area).

Ex. 1306, [0079]. In other words, a POSITA would have recognized that in common parlance, “*formulate*” simply means to “generate” in the context of the 645 Patent. For example, as Herrmann demonstrates, formulate and generate are used interchangeably. A POSITA would have been motivated to combine Herrmann and Krantz for the reasons I demonstrate above. *See* § VIII.C.

102. Accordingly, Krantz, or Krantz and Herrmann, disclose that the gatekeeper “*formulate[s] an audit request.*”

e. [1.4] send the audit request to the device,

103. Krantz discloses that the gatekeeper “*send[s] the audit request to the device.*” As I discuss with respect to [1.3], which I incorporate herein by reference, the “*gatekeeper*” (server 215) formulates audit requests.

104. The teachings of Krantz make it further apparent that it “*send[s] the audit requests to the device.*” For instance, as just discussed, Krantz discloses that “[s]erver 215 [*gatekeeper*] **may request that client 205 provide a password** that is associated with a user-identifier. **A user of client 205 can respond** by providing

a password to server 215. It may also be that **client 205 and server 215 exchange EAP-messages** [*‘send the audit request’*] to pass certificates, keys, and supported cipher suites. Depending on the EAP type, other credential information may also **be exchanged between client 205 and server 215.**” Ex.1305, [0074]. Krantz’s use of the terms “respond” and “exchange” to reference the back and forth between server 215 (the “*gatekeeper*”) and client device 205 (“*the device*”) demonstrates that the audit request is “*sen[t]*” from the “*gatekeeper*” of Krantz “*to the device.*”

105. Additionally, in the case of a denial of access to resources, “**server 215 can send** [*‘send’*] **an EAP-Notification** [*‘the audit request’*] ... [which] can [] include conditions **a user** [i.e., user of ‘*the device*’] **must satisfy** to become authorized to access resources located on VLAN C (e.g., signup, renewal, etc.). If **client 205** [*‘the device’*] is to perform downloads by accessing a provided URI, access point 209 can enable the transfer of HTTP packets for port 251. Server 215 may send commands to access point 209 to cause access point 209 to enable the transfer of HTTP packets.” Ex.1305, [0078].

106. Accordingly, Krantz discloses that the gatekeeper “*send[s] the audit request to the device.*”

f. [1.5] evaluate device information received from the device in response to the audit request,

107. Krantz discloses that the gatekeeper “*evaluate[s] device information received from the device in response to the audit request.*”

108. For example, Krantz discloses that “a server has the functionality to authorize access” to a network and/or selected elements within a network after the client 205 (“*device*”) sends credentials to server 215 (“*gatekeeper*.”). *See, e.g.*, Ex.1305, [0016], [0071]. Krantz discloses that the “server (**based at least in part on received credentials**) **may authorize** the client for at least partial access to the single network” (in other words, the gatekeeper “*evaluate[s] device information received from the device*”). Ex.1305, [0017]; *see also* [0018] (noting an “Extensible Authentication Protocol (“EAP”) Response/Identity message from the client to the server.”); *see also*, [0074]-[0076], [0104], [0107]-[0109] (collectively describing techniques for requesting access in a VLAN environment and “*evaluating device information received from the device*,” e.g., such as passing certificates, keys, and supported cipher suites, as well as other credential information).

109. Krantz discloses that “[s]erver 215 [*‘gatekeeper’*] may request that client 205 provide a password that is associated with a user-identifier. A user of client 205 can respond by providing a password to server 215. **It may also be that client 205 and server 215 exchange EAP-messages to pass certificates, keys, and supported cipher suites.** [*“device information”*]. Depending on the EAP type, **other credential information may also be exchanged between client 205 and server 215.**” Ex.1305, [0074]. Based on this exchange, the gatekeeper of

Krantz will “*evaluate device information received from the device.*” More specifically, “[w]hen credentials are sent to server 215 [the ‘gatekeeper’], authentication module 217 can receive the credentials [] and **compare the credentials** to entries in user database 218 ...” Ex.1305, [0076].

110. Krantz discloses several other specific examples of “*evaluat[ing] device information received from the device.*” For instance, Krantz explains that access to the network may be controlled by “configuration settings,” which can include “supported types of encryption” or “supported types of authentication,” which can include an indication for EAP, which is short for “Extensible Authentication Protocol.” Ex.1305, [0066]-[0067]. Krantz further explains that “[a]fter an EAP type is selected . . . **computer systems can attempt to authenticate with one another** through the transfer of EAP messages . . . in accordance with the selected EAP type. For example, when EAP-TLS is selected, **client 205 may communicate with server 215** through a series of EAP messages in accordance with EAP-TLS **so as to be authenticated and possibly authorized to access resources located on VLAN C.**” Ex.1305, [0068]. The use of EAP-TLS (*e.g.*, Ex.1305, [0067]-[0074], [0100]) was a known authentication protocol that would be used to authenticate client devices. Ex.1316 [RFC2716] at 1; *see also*, Ex.1316 [RFC2716] at 2 (“The Extensible Authentication Protocol (EAP) ... provides a standard mechanism for support of additional authentication methods

within PPP ... [and] support for a number of authentication schemes may be added, **including smart cards, Kerberos, Public Key, One Time Passwords, and others. To date however, EAP methods such as [6] [Security Considerations] have focussed [sic] on authenticating a client to a server.”**)

111. Krantz similarly discloses that “[w]hen EAP-MSChap V2 is indicated, the “MS Chap V2 Parameters element at line 67 can be included in a configuration sub-file to indicate whether **or not operating system credentials should be used for authentication.**” Ex.1305, [0099].

112. Krantz also explains that EAP-TLS and EAP-MSCHAP V2 can both be required from a client seeking access and authorization. Ex.1305, [0099] (“[B]oth of the elements . . . can be included in a configuration sub-file to indicate different EAP types, such as, for example, EAP-TLS and EAP Microsoft/Challenge/Replay Handshake Protocol Version (“EAP-MSChapV2”). Accordingly, a POSITA would understand that each of the above techniques in Krantz would be used to audit and (authenticate⁶) the client 205 (“*evaluat[ing] device information received from the device.*”) which can be in response to an audit

⁶ This information is “*device information*” because it includes information unique, specific to, or related to the device. The use of some of this “*information*” to perform authentication is described in greater detail below. See claim [1.8], below.

request. *E.g.*, Ex.1317, at 2 (“EAP-TLS. TLS (SSL) is the encryption method used to protect the EAP exchange. In Windows, TLS uses certificates for machine logon and user logon. The computer first authenticates to the network using a machine certificate; this allows for computer policies in the Active Directory and SMS packages to be applied before a user logs on.”)(“In Windows, PEAP supports MS-CHAPv2 in addition to TLS. PEAP-MSCHAPv2 uses machine and user credentials for authentication. MSCHAPv2 uses rotating keys to encrypt both machine and user password hashes. Certificates are still required for the authentication sever (IAS) and are used for server-to-client authentication; credentials, carried inside MSCHAPv2, are used for client-to-server authentication.”).

113. Similar to my discussion in [1.4], which I incorporate herein, Krantz’s use of the terms “response,” “respond,” “exchange,” “handshake” and (*see* ¶¶108, 109, 112 *supra*) to reference the back and forth between server 215 (the “gatekeeper”) and client device 205 (“*the device*”) demonstrates that the audit request is “*sent*” from the “gatekeeper” of Krantz “*to the device*” and information is received from the device “*in response to the audit request.*” *See, e.g.*, Ex.1305, [0018] (“Sending credentials can include sending an Extensible Authentication Protocol (‘EAP’) **Response**/Identity message from the client to the server.”), [0074] (“Server 215 may **request** that client 205 provide ... A user of client 205

can **respond** ... client 205 and server 215 **exchange** EAP-messages ... other credential information may also be **exchanged** between client 205 and server 215.”).

114. To the extent that Krantz does not expressly disclose evaluating “*device information*,” it would have been obvious in view of Herrmann. A POSITA would have been motivated to combine Krantz with Herrmann for the reasons described above. *See* § VIII.C. In particular, it would have been obvious to include Herrmann’s techniques for “*evaluating device information*” (e.g., evaluating whether or not the access device includes antivirus software) in the system of Krantz. In particular, the Krantz system already discloses that “access point 209 may block [] protocols ... from being transferred **until client 205 is authenticated (and authorized)**. Ex.1305, [0072]. Furthermore, “[w]hen credentials are sent to server 215, authentication module 217 can receive the credentials (e.g., user-identifier and password) **and compare the credentials to entries in user database 218**.” Ex.1305, [0076].

115. A POSITA would recognize that checking for anti-virus software would merely be an additional item for server 215 (*i.e.*, the “*gatekeeper*”) to evaluate in the authentication process already disclosed by Krantz. Like Krantz, Herrmann discloses a “policy server 460 [which] may enforce a variety of different types of security policies or rules.” Ex.1306, [0068]. One such rule, to be

authorized for network access, would be “a policy requiring that a **specific virus protection program** is operational on each client device that is connected to a network.” *Id.*; *see also, id.*, [0012] (noting it is critical that all machines in a network maintain updated antivirus software so the network is not infiltrated by newly released viruses). Moreover, a POSITA would be motivated to modify the system of Krantz to include the additional policy requiring a specific virus protection program because Krantz is specifically focused on network authentication and security. *See*, Ex.1305, [Abstract].

116. Krantz’s server 215 “*gatekeeper*” and its authentication module, which compares the credentials of a user against a database (e.g., user-identifier and password), as modified by Herrmann, would additionally compare a virus protection program on an access device against approved antivirus programs (“*evaluate device information received from the device in response to the audit request*”). Information regarding approved antivirus software and approved connected devices could be stored in the User Database or an additional Database could be added. *See, e.g.*, Ex.1305, Fig 2 (and corresponding description). A POSITA would have had a reasonable expectation that this additional provision would be successful because Krantz already discloses conditions for client access, such as “signup” or “renewal” conditions to authorize access to network resources. Ex.1305, [0019].

117. Accordingly, Krantz, or Krantz and Hermann, discloses that the gatekeeper “*evaluate[s] device information received from the device in response to the audit request.*”

g. [1.6] approve the device based on evaluation of the device information,

118. Krantz discloses that the gatekeeper “*approve[s] the device based on evaluation of the device information.*”

119. As I discuss with respect to claim element [1.5], the gatekeeper of Krantz discloses the “*evaluation of the device information received from the device in response to the audit request.*” I incorporate that discussion herein. The “gatekeeper” of Krantz “*approves the device based evaluation of the device information.*” For instance, Krantz discloses that the “server (**based at least in part on received credentials**) may authorize the client for at least partial access to the single network” (in other words, the gatekeeper evaluate[s] “*device information*” received from the device and “*approves the device based on evaluation of the device information*”). Ex.1305, [0017]. Krantz further explains that “[a]fter an EAP type is selected . . . **computer systems can attempt to authenticate with one another** through the transfer of EAP messages . . . in accordance with the selected EAP type. For example, when EAP-TLS is selected, **client 205 may communicate with server 215** through a series of EAP messages

in accordance with EAP-TLS **so as to be authenticated and possibly authorized to access resources located on VLAN C.**” Ex.1305, [0068].

120. To the extent that Krantz does not expressly disclose evaluating “*approve[s] the device based on evaluation of the device information,*” it would have been obvious in view of Herrmann for the same reasons described above for claim element [1.5].

121. Accordingly, Krantz, or Krantz and Herrmann, disclose that the gatekeeper “*approve the device based on evaluation of the device information.*”

h. [1.7] an extensible authentication protocol (EAP) server layer (SL) stored in memory and executable to:

122. Krantz discloses “*an extensible authentication protocol (EAP) server layer (SL) stored in memory and executable to.*” As an initial matter, the 645 Patent does not appear to accord any special meaning to the term “*server layer.*” A POSITA would understand, from the context of the claims that this simply refers to the EAP protocol being stored in memory and executable at the server as opposed to, e.g., the client device. This is consistent with the description of Fig. 5 of the 645 Patent, which shows the “EAP Server Layer” as part of the block diagram generally showing a RADIUS server setup (below). Ex.1301, Fig. 5; *see also, e.g.,* Ex.1301, 14:15-16 (repeatedly referring to the “**EAP server**”); *see also, e.g.,* Ex.1301, 14:31-33 (distinguishing the “EAP Server Layer” from the “EAP Client

Layer” which further indicates that “EAP Server Layer” simply means at the “Server” as opposed to the “Client”).

123. The 645 Patent explains that “[a]ccording to an exemplary embodiment, client filters (CF) and server filters (SF) are added along with an agent and GK [i.e., gatekeeper] to perform audits, as highlighted in FIG. 5 [depicted below]. There may be one or more instances of the CF and SF, which correspond to the separate EAP Authentication Modules (EAPM) that are supported.” 645 Patent, 14:22-27. Notably, “the agent and [gatekeeper] are logical components. The functionality of the agent and [gatekeeper] can be embedded into the CF and SF, respectively, or ... may reside on a separate server.” 645 Patent, 14:27-30.

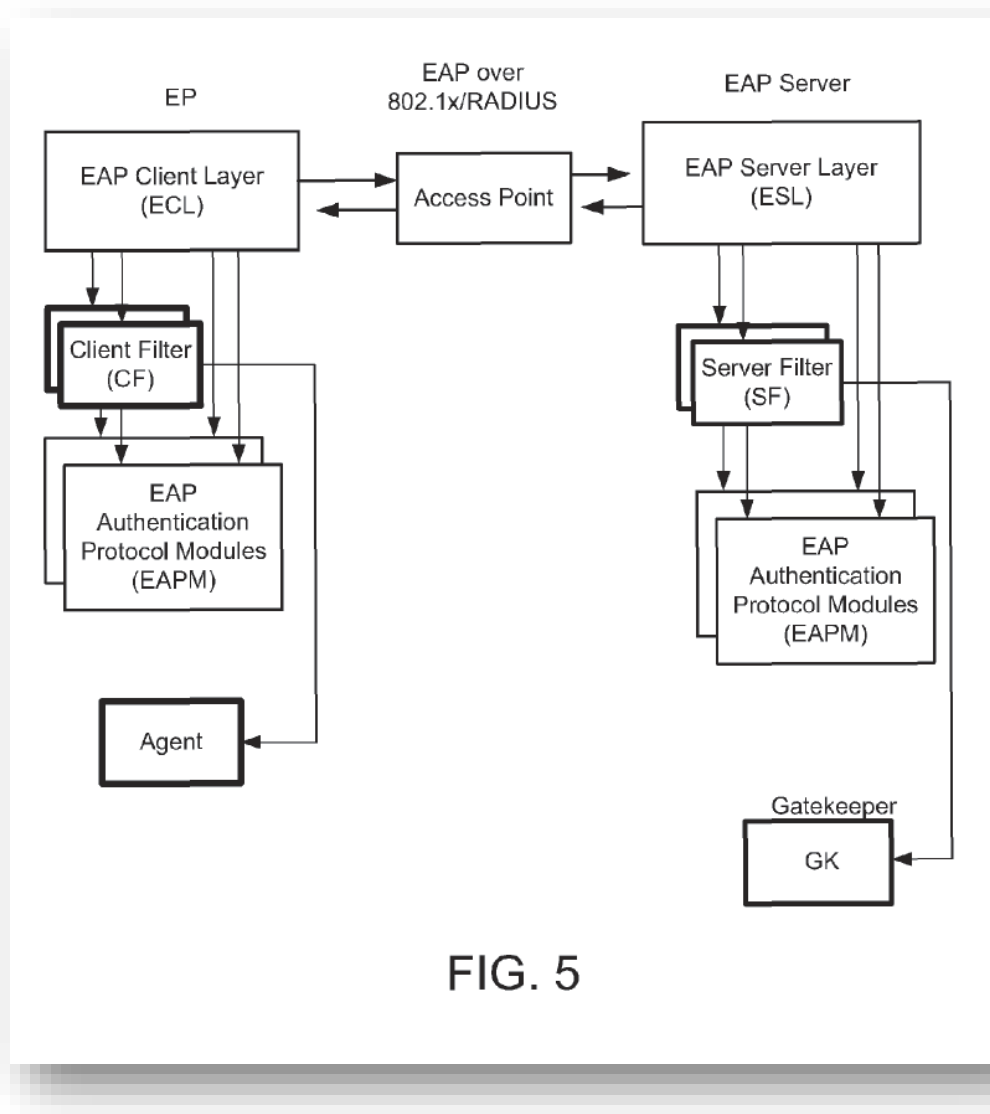


FIG. 5

Ex.1301 (645 Patent), Fig. 5

124. Krantz similarly discloses “an extensible authentication protocol (EAP) server layer (SL).” Krantz discloses the same EAP over 802.1x/RADIUS system as shown above with respect to Fig. 5 of the 645 Patent. Specifically, Krantz discloses sending EAP messages to a server layer capable of receiving such messages (i.e., “an extensible authentication protocol (EAP) server layer”):

Sending credentials can include sending an **Extensible Authentication Protocol (‘EAP’) Response/Identity message** from the client to the server. If the client is attempting to connect to a network via an access point, the EAP Response/Identity message may be encapsulated within some other type of message that can be transferred through the access point. In VLAN embodiments, the access point can insert a Tag Header into the encapsulating message to indicate to a data routing device that the encapsulating message is to be transferred to the first network. **The access point may also be configured to block communications from the client that use protocols other than EAP and 802.1X.**

Ex.1305, [0018].

125. Furthermore, like the 645 Patent, such servers are logical in nature and may be on one more physical servers, including a RADIUS server. For instance, Krantz discusses how “[i]n some embodiments, **server 215 logically represents a Remote Authentication Dial-In User Service (‘RADIUS’) server** [‘*server layer*’]. That is, although visually depicted by a single system, server 215 may include a separate Network Access Server (‘NAS’), a separate authentication server, and a separate shared accounting server.” Ex.1305, [0058]. Krantz explains that “[t]he servers logically represented by server 215 can be **configured to communicate using the RADIUS protocol and, more specifically, may support the RADIUS attributes EAP-Message and Message-Authenticator.**” Ex.1305, [0058]. Thus, Krantz discloses an “(EAP) *server layer*.”

126. Krantz provides numerous other disclosures related to using an “EAP server layer.” For instance, Krantz discloses that “[i]t may be that a particular EAP type is deterministically selected (e.g., by an administrator of server 215 and/or a user of client 205) **when both a client and a server support the particular EAP type.**” Ex.1305, [0068]. Furthermore, Krantz discloses that “when EAP-TLS is selected, client 205 **may communicate with server 215 through a series of EAP messages in accordance with EAP-TLS** so as to be authenticated and possibly authorized to access resources located on VLAN C.” Ex.1305, [0068].

127. The use of EAP-TLS (e.g., Ex.1305, [0067]-[0074], [0100]) was a known authentication protocol that would be used to authenticate client devices. Ex.1316 [RFC2716] at 1. A POSITA would have understood that EAP-TLS involved an extensible authentication protocol (EAP) server layer (SL). *See, e.g.*, Ex.1316 [RFC2716] at 2 (“[W]hile nominally the EAP conversation occurs between the PPP authenticator and the peer, the authenticator MAY act as a passthrough device, with the EAP packets received from the peer being encapsulated for transmission **to a RADIUS server or backend security server.** In the discussion that follows, we will use the term ‘**EAP server**’ to denote the **ultimate endpoint conversing with the peer.**”); *see also, id.* (“Once having received the peer’s Identity, **the EAP server** MUST respond with an EAP-

TLS/Start packet, which is an EAP-Request packet with EAP-Type=EAP-TLS.”).

Notably, EAP-TLS initially stood for “Transport Level Security” but the acronym was later modified to be “Transport **Layer** Security,” and the two terms were used interchangeably around the time of the earliest priority date for the 645 Patent.

Compare Ex.1316 (RFC2716) at p. 1 with Ex.1318 (RFC5216) at p. 3; see also, e.g., Ex.1319 (whitepaper titled “Incorporating WS-Security into a Web services-based portal”) at p. 214 (“The birth of WS-Security allows for another level of security to be added to **transport-level security, a commonly adopted method of security solution in Web applications** because of its simplicity and ease of implementation. **An example of transport-level security solution is Secure Socket Layer (SSL) or Transport Layer Security (TLS).**”). This provides further evidence that the EAP-TLS disclosure of Krantz renders obvious the claimed “*EAP server layer.*”

128. Krantz also discloses that “the computer system may be able to transfer an encapsulated EAP message. One protocol used for encapsulation is the **802.lx protocol, which may be referred to as EAP encapsulation over LANs (‘EAPOL’)**. ... Another protocol used for encapsulation is **EAP-RADIUS, which encapsulates EAP messages within RADIUS messages.**” Ex.1305, [0069]; see also, [0071-0074], [0078-0079], [0098-0101].

129. Moreover, the “*EAP server layer*” is “*stored in memory and executable*.” As Krantz explains, “[s]erver 215 may have modules from a variety of different services **loaded into system memory** [i.e., ‘*stored in memory*’], such as, for example, Internet Information Services (‘IIS’) modules, Internet Authentication Services (‘IAS’) modules, Dynamic Host Control Protocol (‘DHCP’) modules, and Active Directory (‘AD’) modules, so as to facilitate implementation [i.e., it is *executable*] of the principles of the present invention.” Ex.1305, [0058].

130. As I discussed *supra*, a POSITA would understand these components, including the EAP server layer of Krantz, are “*stored in memory and executable*” based on the teachings of Krantz itself. For instance, Krantz teaches:

[T]he invention will be described in the general context of **computer-executable instructions, such as program modules, being executed by computers in network environments**. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. **Computer-executable instructions, associated data structures, and program modules represent examples of the program code means for executing steps of the methods disclosed herein.**

Ex.1305, [0041].

131. Accordingly, Krantz discloses “*an extensible authentication protocol (EAP) server layer (SL) stored in memory and executable to.*”

- i. **[1.8] receive the authentication information from the device using an EAP encryption protocol, and**

132. Krantz discloses an EAP server layer (SL) executable to “*receive the authentication information from the device using an EAP encryption protocol.*”

133. The “*EAP SL*” discussed with respect to claim element 1.7 *supra*, receives information from the device using an EAP encryption protocol. For instance, Krantz discloses that “the computer system may be able to transfer an encapsulated EAP message. **One protocol** used for encapsulation is the **802.1x protocol, which may be referred to as EAP encapsulation over LANs (‘EAPOL’)**. ... **Another protocol** used for encapsulation is **EAP-RADIUS**, which encapsulates EAP messages within RADIUS messages.” Ex.1305, [0069]. As Krantz explicitly states above, these are “*EAP encryption protocol[s].*”

134. Furthermore, Krantz explains that “[s]ending credentials can include sending an Extensible Authentication Protocol (**‘EAP’**) **Response/Identity message from the client to the server** [i.e., ‘*receiv[ing] the authentication information from the device*’].” Ex.1305, [0018]; *see also, id.* (“In VLAN embodiments, the access point can insert a Tag Header into the encapsulating message to indicate to a data routing device that **the encapsulating message is to be transferred to the first network**. The access point may also be configured to block communications from the client that use protocols other than EAP and

802.1X.”). As I discussed earlier, the Response/Identity message includes “*authentication information.*” See, ¶84.

135. Indeed, Krantz explicitly discloses that the EAP server layer “*receives authentication information from the device using [the disclosed] EAP encryption protocols.*” For instance, Krantz discloses that “**when EAP-TLS is selected, client 205 may communicate with server 215** through a series of EAP messages in accordance with EAP-TLS **so as to be authenticated** [i.e., the EAP server layer ‘*receive[s] the authentication information from the device*’] and possibly authorized to access resources located on VLAN C.” Ex.1305, [0068].

136. Krantz discloses that “[s]erver 215 may request that client 205 provide **a password that is associated with a user-identifier.** A user of client 205 can respond by providing a password to server 215. [“*authentication information*”] **It may also be that client 205 and server 215 exchange EAP-messages to pass certificates, keys, and supported cipher suites.** [“*authentication information*”] Depending on the EAP type, **other credential information may also be exchanged between client 205 and server 215.**” Ex.1305, [0074]. Based on this exchange, the gatekeeper of Krantz will “*receives authentication information from the device*” More specifically, “[w]hen credentials are sent to server 215 [the ‘gatekeeper’], authentication module 217 can receive the credentials [] and **compare the credentials** to entries in user database 218 ...” Ex.1305, [0076].

137. These disclosures of Krantz are consistent with the 645 Patent’s description that audit data can be used to authenticate a user. *See*, Ex.1301, 12:4-6 (“In some embodiments, **Authenticate User Step 440** is considered part of the **security policy audit**”); *see also*, Ex.1301, 11:35-42 (“In optional Authenticate User Step 440, Policy Auditor 325 **authenticates the identity of Access Device 230 and/or a user of Access Device 230** ... the authentication may include receipt of a **user name and password** ... or the like.”).

138. Moreover, as discussed *supra*, Krantz also explains that EAP-TLS and EAP-MSCHAP V2 can both be required from a client seeking access and authorization using encryption techniques. Ex.1305, [0099] (“[B]oth of the elements . . . can be included in a configuration sub-file to indicate different EAP types, such as, for example, EAP-TLS and EAP Microsoft/Challenge/Replay Handshake Protocol Version (“EAP-MSChapV2”); *see also*, Ex.1305, [0111] (“A configuration sub-file can be executed to configure connection type, communication protocols, authentication type, **encryption type**, etc.”); *see also*, *e.g.*, Ex.1317, at 2 (“EAP-TLS. TLS (SSL) is the **encryption method** used to protect the EAP exchange. In Windows, TLS uses certificates for machine logon and user logon. The computer first authenticates to the network using a machine certificate; this allows for computer policies in the Active Directory and SMS packages to be applied before a user logs on. ... In Windows, PEAP supports MS-

CHAPv2 in addition to TLS. PEAP-MSCHAPv2 uses machine and user credentials for authentication. MSCHAPv2 uses rotating keys **to encrypt** both machine and user password hashes. Certificates are still required for the authentication sever (IAS) and are used for server-to-client authentication; credentials, carried inside MSCHAPv2, are used for client-to-server authentication.”).

139. Accordingly, Krantz discloses an EAP server layer (SL) executable to “*receive the authentication information from the device using an EAP encryption protocol.*”

j. [1.9] configure an access point in response to approval of the device by the gatekeeper.

140. Krantz discloses an EAP server layer (SL) executable to “*configure an access point in response to approval of the device by the gatekeeper.*”

141. Krantz discloses that “access point 209” (*i.e.*, “*access point*”) is a wireless access point that enables access to wired computer systems on VLAN B (e.g., server 215) and VLAN C (e.g., computer systems included in network 213).” Ex.1305, [0064].

142. As explained in Krantz, “if an authenticated user of client 205 is authorized to access resources located **on VLAN C . . .** client 205 can then be authorized to access resources located on VLAN C.” Ex.1305, [0076]; *see also id.* Ex.1305, [0068]. As explained above, Krantz discloses utilizing a “*gatekeeper*” to

“evaluate device information received from the device in response to [an] audit request” and “approve the device based on evaluation of the device information.”

See discussion regarding claim elements [1.5] and [1.6] *supra*.

143. Thus, *“in response to approval of the device by the gatekeeper,”* the access point facilitates the transfer of data between client 205 (the access *“device”*) and, ultimately VLAN C. Ex.1305, at [0063]-[0064]. As shown below, *“configur[ing] an access point”* to allow access to VLAN C.”

144. Additionally, Krantz explains that “[a]ccess point 209 can be configured as a VLAN aware device and can insert Tag Headers into data frames received from client 205 (or any other computer system on VLAN A) that are to be transferred to VLAN B and/or VLAN C.” Ex.1305, [0064]; *see also* Ex.1305, [0061]. As Krantz also describes:

In VLAN embodiments, when a client is authorized to access resources located on the second network, the access point may insert Tag Headers into client data to indicate to the data routing device that client data is to be transferred to the second network.

Ex.1305, [0023].

145. Krantz is therefore explaining that when configured as a “VLAN aware device,” Krantz’s *“access point”* is *“configure[d]”* because Access Point

209 (via Port 251) will insert VLAN Tag Headers into incoming data frames from a client **that may now access VLAN C**. Ex.1305, [0107]. Access Point 209/Port 251 is “*configure[d]*” because it was previously tagging incoming data from client 205, as described above, with headers that enabled access to VLAN B.

146. In particular, Krantz discloses “[w]hen client 205 is authorized to access resources located on VLAN C, access point 209 may enable any protocols that are currently blocked for port 251 and may insert Tag Headers indicating that data from client 205 is to be transferred to VLAN C. Server 215 can send commands to access point 209 to cause access point 209 to enable protocols and cause access point 209 to insert appropriate Tag Headers.” As already explained above, configuring input ports in VLAN aware devices to tag incoming data was a well-known technique to POSITAs by 2003. Ex.1307 (IEEE 802.1Q), 11 (Table 5-1) (explaining that when the “Reception Port receives as (and does not discard)” untagged data, it “shall support the insertion of tag headers of a format appropriate to the destination MAC, carrying a non-null VID.”); Ex.1308 (U.S. Patent No. 6,430,621), 4:42-47 (“[S]ource node A transmits the packet. No VLAN tag information is present in a packet transmitted from a source node. At 415, the packet is received at switch 305, port 2. Port 2 is configured as a member of LAN 1. Thus, a VLAN tag header is inserted in the packet with a VID=1.”); Ex.1309 (U.S. Patent No. 6,032,194), 16:4-13 (“every time a message is received by the

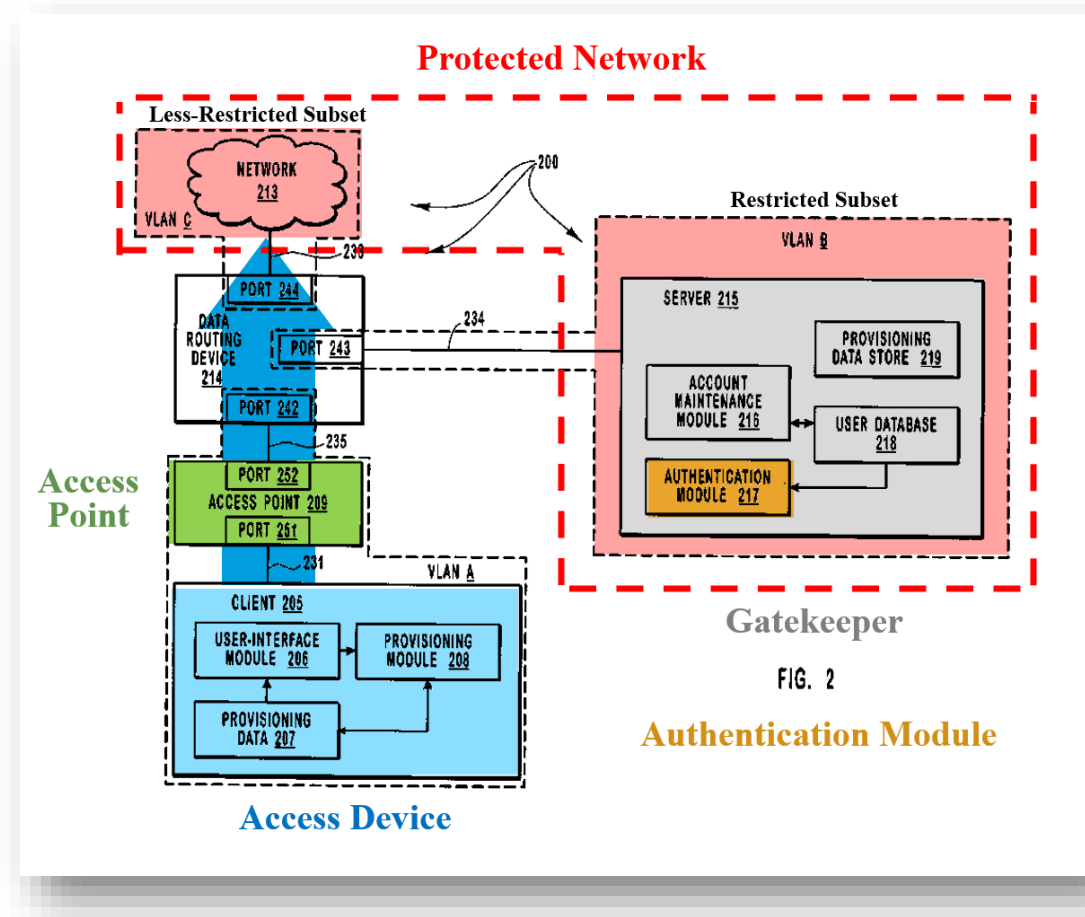
switch on a local port the VLAN designation of that port is associated with the message. Association is accomplished by a flow processing element which looks up the VLAN designation in a memory based on the local port where the message originated. In addition to the 402 patent, an IEEE Standards committee is preparing a standard for Virtual Bridged Local Area Networks. *See* IEEE Standard 802.1Q (draft).”); Ex.1310 (Cisco IOS 12.0 Switching Services), p.210 (“The switch that receives the frame from the source station inserts the VLAN ID and the packet is switched onto the shared backbone network. When the frame exits the switched LAN, a switch strips the header and forwards the frame to interfaces that match the VLAN color.”).

147. Krantz explains that “data routing device 214 can be configured to separate computer systems into different VLANs. As depicted in FIG. 2, **data routing device 214 has been configured to separate network architecture 200 into VLANs A, B, and C.** Data routing device 214 can be configured to transfer both tagged data frames and untagged data frames between the VLANs depicted in FIG. 2.” *Id.* at [0060]; *see also*, e.g., *id.* at [0064] (“Access point 209 can be configured as a VLAN aware device and can insert Tag Headers into data frames received from client 205 (or any other computer system on VLAN A) that are to be transferred to VLAN B and/or VLAN C.”). Accordingly, by configuring the flow of data in this manner, the “*gatekeeper*” (server 215) may evaluate device

information and approve the access device to access data (“*in response to approval of the device by the gatekeeper*”)—such as the resources on VLAN C. Ex.1305, [0062] (“Any of the ports included in data routing device 214 (e.g., ports 242, 243, and 244) can be configured to transfer tagged data frames but drop untagged data frames.”).

148. As discussed above, a number of different types of device information can be utilized for the audit, such as exchanging a user-identifier and password, EAP-messages to pass certificate keys, as well as cipher suites. *See*, Ex.1305, [0074]. “When [the] credentials are sent to server 215, authentication module 217 can receive the credentials (e.g., user-identifier and password) and compare the credentials to entries in user database 218. If credential information matches an entry in user database 218, the identity of a user is authenticated (i.e., server 215 believes that the user represented by the user-identifier is the user who entered the credential information). **If an authenticated user of client 205 is authorized to access resources located on VLAN C (e.g., the user is up to date on payments for their account), client 205 can then be authorized to access resources located on VLAN C.**” *Id.*, [0076]. Thus, a POSITA would understand that the “*access point*” is “*configure[d]*” (*id.*, [0053]) when the audit data demonstrates that the access “*device*” is properly audited. In particular, based on Krantz’s disclosure, the communication path for an authorized access “*device*” would flow through

“communication port” 244, as depicted in the annotated figure below in blue.



Ex.1305 (Krantz), Figure 2 (annotated).

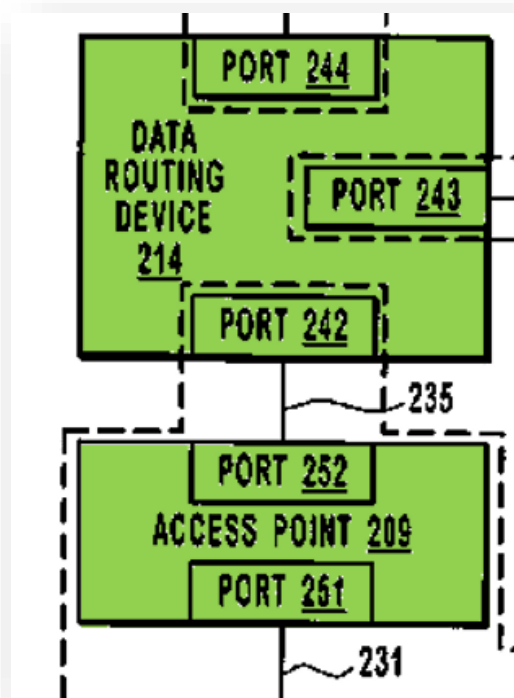
149. Indeed, the above shows, and a POSITA would additionally understand that the “access point” is “configure[d]” (*id.*, [0053]) when the audit data demonstrates that the access “device” is properly audited because, in Krantz, access point 209 is initially configured to allow access only to VLAN B—the restricted subset of the “network” (as shown below):



FIG. 2

150. To the extent one may assert that Krantz does not expressly disclose “*configur[ing] an access point*,” it would have been obvious to implement the Data Routing Device disclosure of Krantz within the Access Point 209 itself.

Ex.1303
Cisco Systems, Inc. / Fortinet, Inc. / Page 79



Ex.1305 (Krantz), Figure 2 (annotated and focused).

152. Krantz discloses that “data routing device 214 can be configured to separate computer systems into different VLANs. As depicted in FIG. 2, data routing device 214 has been configured to separate network architecture 200 into **VLANs A, B, and C**. Data routing device 214 can be configured to transfer both tagged data frames and untagged data frames between the VLANs depicted in FIG. 2.” Ex.1305, [0061].

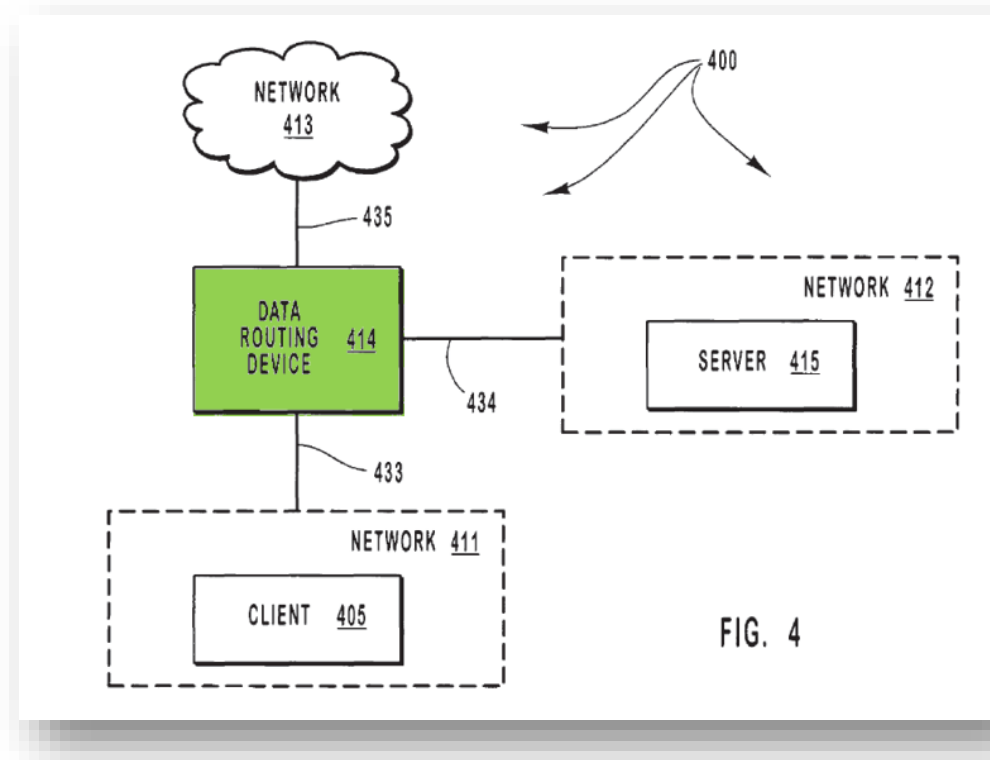
153. For example, Krantz states that “[n]etwork architecture 200 also includes data routing device 214. **Data routing device 214** logically represents a special-purpose computer system, such as, for example, **a router and/or a switch, that can determine how to transfer data between the ports included in data**

Ex.1303

routing device 214. That is, when a portion of data, such as, for example, a data frame, is transferred in at a first port (e.g., port 242) **data routing device 214 can determine, based on configuration rules, that the frame is to be transferred out at a second port** (e.g., port 243). For example, data routing device 214 can determine that a data frame received from access point 209 is to be sent to server 215. It may also be that a data frame is transferred in and out of data routing device 214 at the same port.” *Id.*, [0059]; *see also*, e.g., *id.*, [0060] (“Thus, data routing device 214 can be configured to separate computer systems into different VLANs.); *see also generally*, *id.* [0064]. Such configuration that sends data to VLAN C (*i.e.*, the “*network*” that has controlled access) would be “*in response to approval of the device by the gatekeeper*” as Krantz requires authorization and authentication as described in Krantz in detail.

154. A POSITA would have found it obvious that the technique discussed with respect to the “data routing device 214” could simply be implemented **at the access point itself** as the data routing device is merely a “logical representation” of, e.g., a router and/or switch. Ex.1305, [0059]. In such a scheme, the Access Point 209 would implement that port configuration and data transfer techniques described with respect to “data routing device 214”).

155. In fact, Krantz already contemplates combining the functionality of the “access point” and “data routing device” in a single physical or logical representation. For example, Figure 4 of Krantz demonstrates that such functionality can be readily combined with the data routing device as the access point:



156. In particular, Krantz explains that:

Data routing device 414 logically represents a computer system that can determine where to transfer data received from network 411, such as, for example, data received from client 405. That is, when data routing device 414 receives data from network 411, data routing device 414 can determine if the data is to be transferred to network

412 or network 413. As such, data routing device 414 can be configured to allow access to one network (e.g., network 412) while at the same time restricting access to another network (e.g., network 413).

Ex.1305, [0051] (emphasis added).

157. Indeed, with respect to the disclosures in Figure 2, Krantz expressly discloses that “Data routing device 214 **logically represents** a special-purpose computer system, such as, for example, a router and/or a switch, that can determine how to transfer data between the ports included in data routing device 214,” Ex.1305, [0059] so a POSITA would understand in view of that disclosure that such a “logical” architecture could readily be implemented in a single device such as an access point. Moreover, Krantz explains that such a logical implementation would include functionality that enables a “data frame” to be “transferred in and out of data routing device 214 **at the same port.**” Ex.1305, [0059] (emphasis added). In fact, Krantz repeatedly describes the “data routing” device and “access point” device as having the same functionality. *Compare, e.g.,* Ex.1305, [0060] (“Data routing device 214 can be configured to transfer both tagged data frames and untagged data frames between the VLANs depicted in FIG. 2.); [0064] (“Access point 209 can be configured as a VLAN aware device and can insert Tag Headers into data frames received from client 205 (or any other computer system on VLAN A) that are to be transferred to VLAN B and/or VLAN C.”); *see also*

[0069], [0073] (describing the same functionality for both the data routing device and access point). Therefore, it is clear that the functionality can occur at the access point and data routing device combined **or** the access point alone, without having any impact on the overall functionality of the system described in Krantz. Using one technique or the other would merely have been a matter of design choice.

158. Thus, it would have been obvious to combine the functionality of the Data Routing Device and Access Point of Krantz in the same device for the reasons described above.

159. Accordingly, Krantz discloses “*configur[ing] an access point in response to approval of the device by the gatekeeper.*”

2. Claim 2

- a. [2.0] The network access control system of claim 1, wherein the gatekeeper is hosted at the same computing device as the EAPM, ESL, and SF.**

160. Krantz discloses “[t]he network access control system of claim 1, wherein the gatekeeper is hosted at the same computing device as the EAPM, ESL, and SF.”

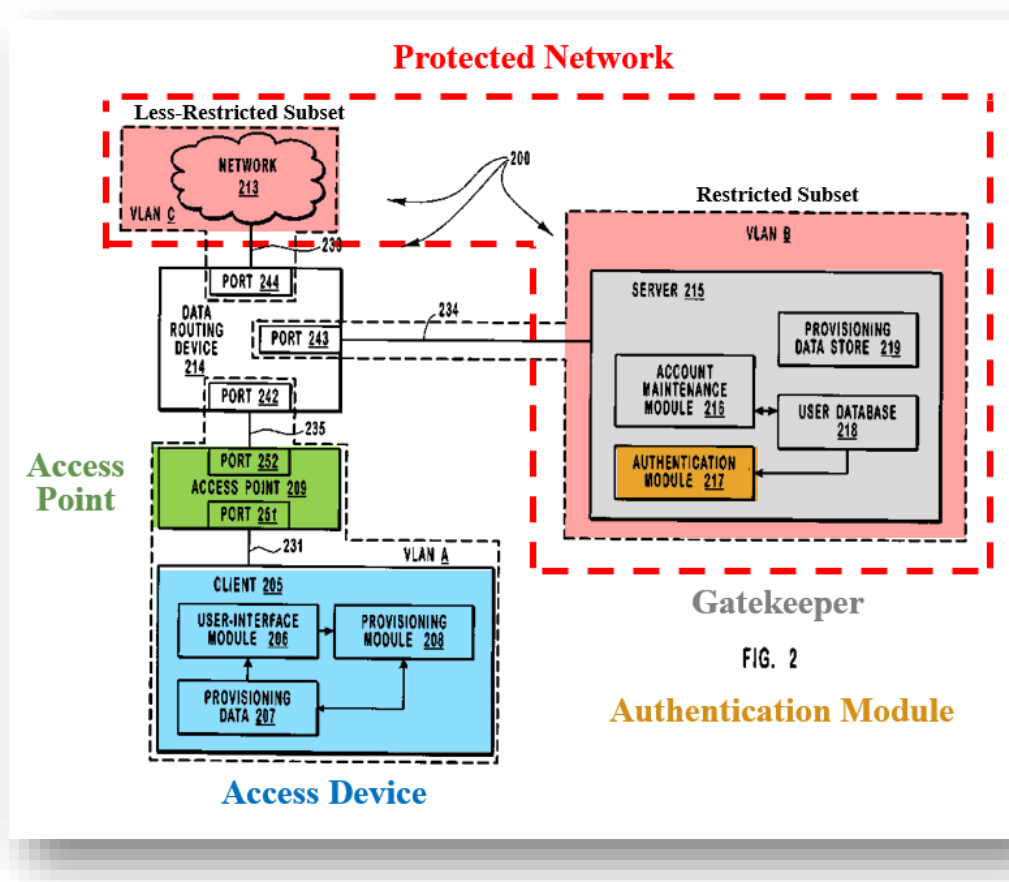
161. As I discuss above, the components of the system described in the 645 Patent can be logical in nature and thus reside on the same or different physical machines. For instance, the 645 Patent articulates that “[a]ccording to an

exemplary embodiment, client filters (CF) **and server filters (SF)** are added along with an agent and GK [i.e., gatekeeper] to perform audits ... There may be one or more instances of the CF and SF, which correspond to the separate **EAP Authentication Modules (EAPM)** that are supported.” 645 Patent, 14:22-27.

Notably, “the agent and **[gatekeeper]** are logical components. The functionality of the agent and **[gatekeeper]** can be embedded into the CF and SF, respectively, or ... may reside on a separate server.” 645 Patent, 14:27-30.

162. Similarly, Krantz discloses a logical structure in which the “*authentication module*” can be stored on the same server that functions as the “*gatekeeper*” or a separate server. *See, e.g.*, Ex.1305, [0058] (“In some embodiments, server 215 logically represents a Remote Authentication Dial-In User Service (‘RADIUS’) server. That is, **although visually depicted by a single system, server 215 may include** a separate Network Access Server (‘NAS’), **a separate authentication server**, and a separate shared accounting server.”).

163. Krantz discloses that “*the gatekeeper is hosted at the same computing device as the EAPM, ESL, and SF.*” In particular, the entire functionality of the gatekeeper in Krantz can be hosted “*at the same computing device*” as depicted in Fig. 2 of Krantz. This is seen below (in gray) with respect to server 215—the “*gatekeeper*”—as well as the authentication module (“*EAPM*”), an account maintenance module, a user database, and a provisioning data store.



Ex.1305 (Krantz), Figure 2 (annotated).

164. As I discussed prior, Krantz discusses how “[i]n some embodiments, **server 215 logically represents a Remote Authentication Dial-In User Service (‘RADIUS’) server** [‘server layer’]. That is, although visually depicted by a single system, server 215 may include a separate Network Access Server (‘NAS’), a separate authentication server, and a separate shared accounting server.” Ex.1305, [0058]. Krantz explains that “[t]he servers logically represented by server 215 can be configured to communicate using the RADIUS protocol and, more

specifically, may support the RADIUS attributes EAP-Message and Message-Authenticator.” Ex.1305, [0058].

165. Krantz also discloses that “[i]t should be understood that VLAN, **IP filtering**, and VPN techniques are merely examples of some of the techniques that can be used to differentiate between networks so as to allow access to one network while restricting access to another network.” Ex.1305, [0055]. Krantz further states that “[i]t should be understood that access to a single network can be limited using VLANs, **IP filtering**, or VPNs. **For example, a server located on network 413 can provide client 405 with limited access to resources located on network 413.** It would be apparent to one skilled in the art, after having reviewed this description, that other techniques, in addition VLAN, IP filtering, and VPN techniques, can be used to allow access to one network, while restricting access to another network.” Ex.1305, [0055].

166. The 645 Patent provides little explanation regarding the functionality of the server filters beyond being “added along with an agent and GK to perform audits, as highlighted in FIG. 5” and showing that it can (but is not required to) relate to an “SF API” (Server Filter Application Programming Interface) as seen in Fig. 6B. Nonetheless, as I just described in the preceding paragraph, Krantz discloses filtering techniques at the server level.

167. The “*SF*” of the 645 Patent is simply part of the audit process that the gatekeeper formulates. As I discussed prior, according to the 645 Patent, the functionality of the gatekeeper **can be embedded into the SF (server filter)**. Ex.1301, 14:28-29. The ESL “can call the CF and SF instances corresponding a particular authentication method.” Ex.1301, 14:32-34. In other words, these components appear to work with (or as part of) the gatekeeper to implement network access control. Indeed, the 645 Patent explains that to both the client layer and server layer, the CF and SF simply appear “to be an EAPM [‘EAP authentication module’] which incorporates the functionality of the original EAPM...” Ex.1301, 14:35-38; *see also*, Ex.1301, 14:46-53 (explaining that the CF, SF, ESL, and EAPM are all part of an iterative authentication process that “continues until the authentication either succeeds or fails”). In Krantz, as I have explained with respect to claim [1.0], the “*network access control*” can be performed by the logical components found in server 215 (e.g., the gatekeeper and authentication module).” I have also explained *supra*, e.g., in ¶163, how the entire functionality of the “*gatekeeper*” in Krantz (which includes the authentication module or “*EAPM*”), can be hosted at the same server 215. Thus “*the gatekeeper is hosted at the same computing device as the EAPM, ESL, and SF*” in Krantz (i.e., server 215).

168. Accordingly, Krantz discloses “[t]he network access control system of claim 1, wherein the gatekeeper is hosted at the same computing device as the EAPM, ESL, and SF.”

3. Claim 5

a. [5.0] The network access control system of claim 1, wherein the SF is part of the ESL.

169. Krantz discloses “[t]he network access control system of claim 1, wherein the SF [(server filter)] is part of the ESL [(EAP Server Layer)].”

170. As I discussed prior, the components of the system described in the 645 Patent can be logical in nature and thus reside on the same or different physical machines. For instance, the 645 Patent articulates that “[a]ccording to an exemplary embodiment, client filters (CF) **and server filters (SF)** are added along with an agent and GK [i.e., gatekeeper] to perform audits ... There may be one or more instances of the CF and SF, which correspond to the separate **EAP Authentication Modules (EAPM)** that are supported.” 645 Patent, 14:22-27. Notably, “the agent and **[gatekeeper]** are logical components. The functionality of the agent and **[gatekeeper]** can be embedded into the CF and SF, respectively, or ... may reside on a separate server.” 645 Patent, 14:27-30. As I also discussed prior, Krantz discusses how “[i]n some embodiments, **server 215** [the ‘gatekeeper’] **logically represents a Remote Authentication Dial-In User Service (‘RADIUS’) server** [‘server layer’]. Thus, given the logical nature of the

systems described in both the 645 Patent and Krantz, a POSITA would understand that the “*gatekeeper*” can be part of the “*server layer*” which means the “*server filter (SF)* can be *part of the EAP Server Layer (ESL)*” (as the gatekeeper can be embedded into the server filter (SF)). Moreover, as I explained with respect to claim 2, which I incorporate herein by reference, the entire functionality of the “*EAPM*” and “*gatekeeper*,” including the “*SF*” can be implemented in the same location, e.g., the “*ESL*.”

171. Accordingly, Krantz discloses “[*t*]he network access control system of claim 1, wherein the *SF* is part of the *ESL*.”

4. Claim 6

- a. **[6.0] The network access control system of claim 1, wherein the EAPM includes a plurality of authentication modules configured to perform different authentication operations.**

172. Krantz discloses “[*t*]he network access control system of claim 1, wherein the EAPM [*(EAP Authentication Module)*] includes a plurality of authentication modules configured to perform different authentication operations.”

173. As an initial matter, Krantz makes clear that in some embodiments, “server 215 [i.e., the “*gatekeeper*” which includes the “*EAPM [(EAP Authentication Module)]*” logically represents a [RADIUS] server. ... **Server 215 may have modules from a variety of different services loaded into system memory**, such as, for example, Internet Information Services (“*IIS*”) **modules**,

Internet Authentication Services (‘IAS’) **modules**, Dynamic Host Control Protocol (‘DHCP’) **modules**, and Active Directory (‘AD’) **modules**, so as to facilitate implementation of the principles of the present invention.” Ex.1305, [0058]. A POSITA would have recognized that the logical nature of these embodiments meant that it would have been obvious that the EAPM could “*include[] a plurality of [these] authentication modules [which were] configured to perform different authentication operations.*” For instance, the EAPM could include the “Internet Authentication Services” module. This is because program modules are simply components that work together in a system, e.g., on server 215, that “[g]enerally [] include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types.” Ex.1305, [0041].

174. Indeed, Krantz explains that the authentication module (“EAPM”) “*includes*” or at minimum works in conjunction with “*a plurality of authentication modules configured to perform different authentication operations.*” For example, Krantz discloses that “[w]hen credentials are sent to server 215, **authentication module 217 can receive the credentials** (e.g., user-identifier and password) and compare the credentials to entries **in user database 218.**” Ex.1305, [0076]. Additionally, “**account maintenance module 216** may access user database 218 to retrieve previously entered user information. ...” Ex.1305, [0104]. As seen in Fig. 2 above, “account maintenance module 216” is hosted on the same logical server

environment (server 215) as the authentication module (“*EAPM*”). As seen in Figure 2, “account maintenance module 216” is hosted on the same logical server environment (server 215) as the authentication module (“*EAPM*”) but is not shown as a separate module within the *EAPM*. However, given the logical nature of the invention in Krantz, authentication module would be understood to optionally include other modules and data structure elements that each perform particular tasks as part of the authentication methods of Krantz.

175. Accordingly, Krantz discloses “[t]he network access control system of claim 1, wherein the *EAPM* includes a plurality of authentication modules configured to perform different authentication operations.”

176. Accordingly, Krantz, or Krantz and Hermann, render obvious claims 1-2, and 5-6.

X. CONCLUSION

177. 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, and that these statements were made with knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code.

Dated: March 28, 2024
Henry Houh, Ph.D.

