UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SAP America, Inc.
Petitioner

IPR2025-00419
U.S. Patent No. 7,936,738

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

TABLE OF CONTENTS

Page

EXHIBIT LIST 4
I. INTRODUCTION 6
II. QUALIFICATIONS AND PROFESSIONAL EXPERIENCE 7
III. LEVEL OF ORDINARY SKILL IN THE ART 19
IV. RELEVANT LEGAL STANDARDS 20
V. OVERVIEW OF THE '738 PATENT 22
VI. CLAIM CONSTRUCTION 26
VII. IDENTIFICATION OF HOW THE CLAIMS ARE UNPATENTABLE 26
VIII. THE PRINCIPAL PRIOR ART 27
IX. CLAIMS 1-10 AND 33 ARE UNPATENTABLE 32
A. Ground 1: Claims 1–10 and 33 Would Have Been Obvious In View of Forissier 32
1. Claim 1 33
i. [1.0] A method of storing context information in an outgoing message sent from a node including a computing device using a protocol stack having at least one layer, comprising: 33
ii. [1.1] providing, by the computing device, the outgoing message from an application to a layer of the protocol stack, the outgoing message is destined for an application on a destination node; 39
iii. [1.2] selectively indicating to the layer of the protocol stack that context information is to be obtained for that layer; 42
iv. [1.3] obtaining, by the computing device, context information in accordance with the indication; and 44
v. [1.4] adding, by the computing device, the obtained context information to the outgoing message such that a response, received from the destination node, to the outgoing message contains the obtained context information. 45
2. Claim 2 47
3. Claim 3 51

4. Claim 4 53
5. Claim 5 54
6. Claim 6 56
7. Claim 7 58
8. Claim 8 59
9. Claim 9 61
10. Claim 10 62
11. Claim 33 67

X. CONCLUSION 72

EXHIBIT LIST

No.	Exhibit Description
1101	U.S. Patent No. 7,936,738
1102	U.S. Patent No. 7,936,738 File History
1103	Declaration of Kevin Almeroth, Ph.D.
1104	CV of Kevin Almeroth, Ph.D.
1105	U.S. Patent No. 7,936,763 to Forissier et al.
1106	J. Forissier, et al., *Method and Apparatus for Load-Balancing in a Distributed Processing System*, European Patent Application 03291191.9, 2003-05-20.
1107	IEEE STANDARD GLOSSARY OF COMPUTER NETWORKING TERMINOLOGY, (IEEE Std 610.7-1995, 1995)
1108	Director Vidal's Memorandum Regarding Discretionary Denial Procedure, Dated June 21, 2022
1109	First Amended Docket Control Order in *Valtrus Innovations Ltd. v. SAP America, Inc. et al*, 2:24-cv-00021-JRG (E.D. Tex.)
1110	Motion to Dismiss for Lack of Standing (Public Version) in *Valtrus Innovations Ltd. v. SAP America, Inc. et al*, 2:24-cv-00021-JRG (E.D. Tex.)
1111	Motion to Reconsider Ruling on Motion to Dismiss for Lack of Standing (Public Version) in *Valtrus Innovations Ltd. v. SAP America, Inc. et al*, 2:24-cv-00021-JRG (E.D. Tex.)
1112	Original Complaint for Patent Infringement in *Valtrus Innovations Ltd. v. SAP America, Inc. et al*, 2:24-cv-00533-JRG (E.D. Tex.)
1113	Plaintiffs' Notice Regarding *Valtrus II* in *Valtrus Innovations Ltd. v. SAP America, Inc. et al*, 2:24-cv-00021-JRG (E.D. Tex.)
1114	SAP's Invalidity Contentions (Cover Pleading) in *Valtrus Innovations Ltd. v. SAP America, Inc. et al*, 2:24-cv-00021-JRG (E.D. Tex.)

1115	Valtrus's Infringement Contentions (Cover Pleading) in *Valtrus Innovations Ltd. v. SAP America, Inc. et al*, 2:24-cv-00021-JRG (E.D. Tex.)
1116	Neil Briscoe, Understanding the OSI 7-Layer Model (July 2000)
1117	SIP: Session Initiation Protocol, The Internet Society (2002)

I, Kevin Almeroth, Ph.D., do hereby declare as follows:

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. 7,936,738 ("the '738 Patent") to Bouat.

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 1–10 ("the Challenged Claims") and 33 of the '738 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 '738 Patent, Ex.1101.

   b. the prosecution history of the '738 Patent ("'738 File History"), Ex.1102.

   c. U.S. Patent No. 7,936,763 to Forissier et al ("Forissier") (Ex. 1105).

   d. The additional references I refer to below and/or are 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-1104. The following is a brief summary of my relevant qualifications and professional experience.

8. I am currently a Professor Emeritus in the Department of Computer Science at the University of California, Santa Barbara (UCSB). While at UCSB, I held faculty appointments and was a founding member of the Computer Engineering (CE) Program, Media Arts and Technology (MAT) Program, and the Technology Management Program (TMP). I also served as the Associate Director of the Center for Information Technology and Society (CITS) from 1999 to 2012. I have been a faculty member at UCSB since July 1997.

9. I hold three degrees from the Georgia Institute of Technology: (1) a Bachelor of Science degree in Information and Computer Science (with minors in Economics, Technical Communication, and American Literature) earned in June

1992; (2) a Master of Science degree in Computer Science (with specialization in Networking and Systems) earned in June 1994; and (3) a Doctor of Philosophy (Ph.D.) degree in Computer Science (Dissertation Title: Networking and System Support for the Efficient, Scalable Delivery of Services in Interactive Multimedia System, minor in Telecommunications Public Policy) earned in June 1997. During my education, I took a wide variety of courses as demonstrated by my minors. My undergraduate degree also included a number of courses more typical of a degree in electrical engineering including digital logic, signal processing, and telecommunications theory.

10. One of the major concentrations of my research over the past 30+ years has been the delivery of multimedia content and data between computing devices, including through various network architectures. In my research, I have studied large-scale content delivery systems, and the use of servers located in a variety of geographic locations to provide scalable delivery to hundreds or thousands of users simultaneously. I have also studied smaller-scale content delivery systems in which content is exchanged between individual computers and portable devices. My work has emphasized the exchange of content more efficiently across computer networks, including the scalable delivery of content to many users, mobile computing, satellite networking, delivering content to mobile devices, and network support for data delivery in wireless networks.

11. In 1992, the initial focus of my research was on the provision of interactive functions (e.g., VCR-style functions like pause, rewind, and fast-forward) for near video-on-demand systems in cable systems; in particular, how to aggregate requests for movies at a cable head-end and then how to satisfy a multitude of requests using one audio/video stream broadcast to multiple receivers simultaneously. This research has continually evolved and resulted in the development of techniques to scalably deliver on-demand content, including audio, video, web documents, and other types of data, through the Internet and over other types of networks, including over cable systems, broadband telephone lines, and satellite links.

12. An important component of my research has been investigating the challenges of communicating multimedia content, including video, between computers and across networks including the Internet. Although the early Internet was used mostly for text-based, non-real time applications, the interest in sharing multimedia content, such as video, quickly developed. Multimedia-based applications ranged from downloading content to a device to streaming multimedia content to be instantly used. One of the challenges was that multimedia content is typically larger than text-only content, but there are also opportunities to use different delivery techniques since multimedia content is more resilient to errors. I have worked on a variety of research problems and used a number of systems that

were developed to deliver multimedia content to users. One content-delivery method I have researched is the one-to-many communication facility called "multicast," first deployed as the Multicast Backbone, a virtual overlay network supporting one-to-many communication. Multicast is one technique that can be used on the Internet to provide streaming media support for complex applications like video-on-demand, distance learning, distributed collaboration, distributed games, and large-scale wireless communication. The delivery of media through multicast often involves using Internet infrastructure, devices and protocols, including protocols for routing and TCP/IP.

13. Starting in 1997, I worked on a project to integrate the streaming media capabilities of the Internet together with the interactivity of the web. I developed a project called the Interactive Multimedia Jukebox (IMJ). Users would visit a web page and select content to view. The content would then be scheduled on one of a number of channels, including delivery to students in Georgia Tech dorms delivered via the campus cable plant. The content of each channel was delivered using multicast communication.

14. In the IMJ, the number of channels varied depending on the capabilities of the server including the available bandwidth of its connection to the Internet. If one of the channels was idle, the requesting user would be able to watch their selection immediately. If all channels were streaming previously selected content,

the user's selection would be queued on the channel with the shortest wait time. In the meantime, the user would see what content was currently playing on other channels, and because of the use of multicast, would be able to join one of the existing channels and watch the content at the point it was currently being transmitted.

15. The IMJ service combined the interactivity of the web with the streaming capabilities of the Internet to create a jukebox-like service. It supported true Video-on-Demand when capacity allowed, but scaled to any number of users based on queuing requested programs. As part of the project, we obtained permission from Turner Broadcasting to transmit cartoons and other short-subject content. We also connected the IMJ into the Georgia Tech campus cable television network so that students in their dorms could use the web to request content and then view that content on one of the campus's public access channels.

16. More recently, I have also studied issues concerning how users choose content, especially when considering the price of that content. My research has examined how dynamic content pricing can be used to control system load. By raising prices when systems start to become overloaded (i.e., when all available resources are fully utilized) and reducing prices when system capacity is readily available, users' capacity to pay as well as their willingness can be used as factors in

stabilizing the response time of a system. This capability is particularly useful in systems where content is downloaded or streamed on-demand to users.

17. As a parallel research theme, starting in 1997, I began researching issues related to wireless devices and sensors. In particular, I was interested in showing how to provide greater communication capability to "lightweight devices," i.e., small form-factor, resource-constrained (e.g., CPU, memory, networking, and power) devices. Starting in 1998, I published several papers on my work to develop a flexible, lightweight, battery-aware network protocol stack. The lightweight protocols we envisioned were similar in nature to protocols like Bluetooth, Universal Plug and Play (UPnP) and Digital Living Network Alliance (DLNA).

18. From this initial work, I have made wireless networking—including ad hoc, mesh networks and wireless devices—one of the major themes of my research. My work in wireless networks spans the protocol stack from applications through to the encoding and exchange of data at the data link and physical layers.

19. At the application layer, even before the large-scale "app stores" were available, my research looked at building, installing, and using apps for a variety of purposes, from network monitoring to support for traditional computer-based applications (e.g., content retrieval) to new applications enabled by ubiquitous, mobile devices. For example, my research has looked at developing applications for virtually exchanging and tracking "coupons" through "opportunistic contact" among

mobile wireless devices (i.e., communication among devices moving into communication range with each other). In many of the courses I have taught there is a project component. Through these projects I have supervised numerous efforts to develop new "apps" for download and use across a variety of mobile platforms.

20. Toward the middle of the protocol stack, my research has also looked to build wireless infrastructure support to enable communication among a set of mobile devices unaided by any other kind of network infrastructure. These kinds of networks are useful either in challenged network environments (e.g., when a natural disaster has destroyed existing infrastructure) or when suitable support for network communication never existed. The deployment of such networks (or even the use of traditional network support) are critical to support services like disaster relief, catastrophic event coordination, and emergency services deployment.

21. Yet another theme is monitoring wireless networks, in particular different variants of IEEE 802.11 compliant networks, to (1) understand the operation of the various protocols used in real-world deployments, (2) use these measurements to characterize use of the networks and identify protocol limitations and weaknesses, and (3) propose and evaluate solutions to these problems. I have successfully used monitoring techniques to study wireless data link layer protocol operation and to improve performance by enhancing the operation of such protocols.

For wireless protocols, this research includes functions like network acquisition and channel bonding.

22. One theme in my wireless network research has been cross-layer solutions and innovations. As mentioned above, with greater wireless device use and network support, we envisioned new application paradigms and services, for example, when mobile devices come into contact with each other. Instead of relying on existing infrastructure to relay communication, the devices are able to discover each other and communicate directly. Other examples include discovering and using location information to enhance users' experiences. Network support and novel applications span use a variety of network architectures supporting users on foot, in vehicles, and across varying terrains and environments. Finally, we studied how communication efficiency can be supported through intelligent handoffs as well as location and movement prediction.

23. Protecting networks, including their operation and content, has been an underlying theme of my research almost since the beginning of my research career. Starting in 2000, I have been involved in several projects that specifically address security, network protection, and firewalls. After significant background work, a team on which I was a member successfully submitted a $4.3M grant proposal to the Army Research Office (ARO) at the Department of Defense to propose and develop a high-speed intrusion detection system. Key aspects of the system included

associating streams of packets and analyzing them for viruses and other malware. Once the grant was awarded, we spent several years developing and meeting the milestones of the project. A number of my students worked on related projects and published papers on topics ranging from intrusion detection to developing advanced techniques to be incorporated into firewalls. I have also used firewalls, including their associated malware detection features, in developing techniques for the classroom to ensure that students are not distracted by online content.

24. My recent work ties some of the various threads of my past research together. I have investigated content delivery in online social networks and proposed reputation management systems in large-scale social networks and marketplaces. On the content delivery side, I have looked at issues of caching and cache placement, especially when content being shared and the cache has geographical relevance. We were able to show that effective caching strategies can greatly improve performance and reduce deployment costs. Our work on reputation systems showed that reputations have economic value, and as such, creates a motivation to manipulate reputations. In response, we developed a variety of solutions to protect the integrity of reputations in online social networks. The techniques we developed for content delivery and reputation management were particularly relevant in peer-to-peer communication and recommendations for downloadable "apps."

25. As an important component of my research program, I have been involved in the development of academic research into available technology in the market place. One aspect of this work is my involvement in the Internet Engineering Task Force (IETF). The IETF is a large and open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. I have been involved in various IETF groups including many content delivery-related working groups like the Audio Video Transport (AVT) group, the MBone Deployment (MBONED) group, Source Specific Multicast (SSM) group, the Inter-Domain Multicast Routing (IDMR) group, the Reliable Multicast Transport (RMT) group, the Protocol Independent Multicast (PIM) group, etc. I have also served as a member of the Multicast Directorate (MADDOGS), which oversaw the standardization of all things related to multicast in the IETF. Finally, I was the Chair of the Internet2 Multicast Working Group for seven years.

26. My involvement in the research community extends to leadership positions for several academic journals and conferences. I am the co-chair of the Steering Committee for the ACM Network and System Support for Digital Audio and Video (NOSSDAV) workshop and on the Steering Committees for the International Conference on Network Protocols (ICNP), ACM Sigcomm Workshop on Challenged Networks (CHANTS), and IEEE Global Internet (GI) Symposium. I

have served or am serving on the Editorial Boards of IEEE/ACM Transactions on Networking, IEEE Transactions on Mobile Computing, IEEE Network, ACM Computers in Entertainment, AACE Journal of Interactive Learning Research (JILR), and ACM Computer Communications Review. I have co-chaired a number of conferences and workshops including the IEEE International Conference on Network Protocols (ICNP), IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), International Conference on Communication Systems and Networks (COMSNETS), IFIP/IEEE International Conference on Management of Multimedia Networks and Services (MMNS), the International Workshop On Wireless Network Measurement (WiNMee), ACM Sigcomm Workshop on Challenged Networks (CHANTS), the Network Group Communication (NGC) workshop, and the Global Internet Symposium, and I have served on the program committees for numerous conferences.

27. Furthermore, in the courses I taught at UCSB, a significant portion of my curriculum covered aspects of the Internet and network communication including the physical and data link layers of the Open System Interconnect (OSI) protocol stack, and standardized protocols for communicating across a variety of physical media such as cable systems, telephone lines, wireless, and high-speed Local Area Networks (LANs). The courses I have taught also cover most major topics in Internet communication, including data communication, multimedia

encoding, and mobile application design. My research and courses have covered a range of physical infrastructures for delivering content over networks, including cable, Integrated Services Digital Network (ISDN), Ethernet, Asynchronous Transfer Mode (ATM), fiber, and Digital Subscriber Line (DSL). For a complete list of courses I have taught, see my *curriculum vitae* (CV) (Ex-1104).

28. In addition, I co-founded a technology company called Santa Barbara Labs that was working under a sub-contract from the U.S. Air Force to develop very accurate emulation systems for the military's next generation internetwork. Santa Barbara Labs' focus was in developing an emulation platform to test the performance characteristics of the network architecture in the variety of environments in which it was expected to operate, and, in particular, for network services including IPv6, multicast, Quality of Service (QoS), satellite-based communication, and security. Applications for this emulation program included communication of a variety of multimedia-based services, including video conferencing and video-on-demand.

29. In addition to having co-founded a technology company myself, I have worked for, consulted with, and collaborated with companies for nearly 30 years. These companies range from well-established companies to start-ups and include IBM, Hitachi Telecom, Turner Broadcasting System (TBS), Bell South, Digital Fountain, RealNetworks, Intel Research, Cisco Systems, and Lockheed Martin.

30. Through my graduate education, leadership with CITS, involvement in TMP, role in the development of the Internet2 infrastructure, and consulting with ISPs, I have gained a strong understanding in the role of the Internet in our society and the challenges of deploying large-scale production networking infrastructure. CITS, since its inception, has looked at the role of the Internet in society, including how the evolution of technology have created communication opportunities and challenges, including, for example through disruptive technologies like P2P. TMP looks to focus on non-purely technical issues, including, for example, state-of-the-art business methods, strategies for successful technology commercialization, new venture creation, and best practices for fostering innovation. Through my industry collaborations and Internet2 work, I have developed significant experience in the challenges of deploying, monitoring, managing, and scaling communication infrastructure to support evolving Internet services like streaming media, conferencing, content exchange, social networking, and e-commerce.

31. I am a Member of the Association of Computing Machinery (ACM) and a Fellow of the Institute of Electrical and Electronics Engineers (IEEE).

III. LEVEL OF ORDINARY SKILL IN THE ART

32. 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.

33. A Person of Ordinary Skill in The Art ("POSITA") in July 2003 would have had a working knowledge of the telecommunications and networking art that is pertinent to the '738 Patent. A POSITA would have had a bachelor's degree or graduate 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.

34. 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 '738 Patent.

IV. RELEVANT LEGAL STANDARDS

35. I am not an attorney. In preparing and expressing my opinions and considering the subject matter of the '738 Patent, I am relying on certain basic legal principles that counsel has explained to me.

36. I understand that prior art to the '738 Patent includes patents and printed publications in the relevant art that predate the priority date of the '738 Patent. For

purposes of this Declaration, I am applying July 25, 2003, as the priority date of the '738 Patent, but I take no position on that issue.

37. 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.

38. 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.

39. 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.

40. 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.

41. 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 ’738 PATENT

42. The ’738 Patent is directed to “a method and system for providing fault tolerance in . . . message-based communication systems.” Ex. 1101, 1:2-6. As explained by the ’738 Patent, in the field of telecommunications, fault tolerance is the concept of designing a system in such a way that a failure of part of that system does not “cause significant disruption to the services provided thereby.” Ex. 1101, 1:6-16. “Such systems are often also referred to as high-availability (HA) systems.”

Ex. 1101, 1:15-16. The ’738 Patent claims that fault tolerance can be achieved by “storing context information in an outgoing message sent from a node using a protocol stack having at least one layer.” Ex. 1101, 2:1-4. Some embodiments further involve “restoring the context information of a layer of a protocol stack of a node.” Ex. 1101, 2:28-30.

43. The ’738 Patent notes that “appending context data to a message sent from a HA system” removes the need for the HA system “to store context data in a central data store,” a purported problem of prior art. Ex. 1101, 5:3-7. According to the ’738 Patent, without the need for a central data store or “common storage element,” the system becomes less complex and therefore less costly. Ex. 1101, 4:34-36. The ’738 Patent describes this method of appending context data to a message specifically as it relates to a session initiation protocol network, otherwise known as a SIP network. Ex. 1101, 2:21-24.

44. A POSITA would recognize that SIP is the foundation for Internet telephony: it is a signaling protocol used for establishing, maintaining, and terminating live Internet-based communication sessions, including voice calls or instant messaging. These types of communication use SIP tags that send and receive information to and from the SIP protocol stack. This information includes context information, or information that can be used to characterize the situation of a given thing—usually as it relates to location or time. A POSITA would recognize that

certain tags may contain context information that corresponds to actual addresses and defines where a message is to be sent.

45. In my opinion, this understanding of SIP and SIP tags is consistent with that of the '738 Patent. Though the '738 Patent does not define what "context information" is, it notes that it may "differ depending on particular circumstances" and "may be desirable to only store context information relating to [the current call] with the outgoing message." Ex. 1101, 5:43-48. It expressly provides that "context data"—which a POSITA would recognize is interchangeable with "context information"—"may include a destination address of the SIP call, which would enable the SIP messages related to that call to be routed to the correct destination." Ex. 1101, 7:1-3. As expected, that destination address may be stored in the SIP TAG, which "is included in all related messages and response messages" or a SIP extension header. Ex. 1101, 7:4-10; *see also* Ex. 1101, 2:20-24 ("[C]ontext information may be appended to a SIP TAG field, or to a SIP extension header."); Ex. 1101, 3:25-28 ("[T]he restoration means is adapted for restoring the context using context information stored in a SIPTAG or a SIP extension header.").

46. Figure 7 below shows an exemplary SIP message-flow diagram as described in the '738 Patent:

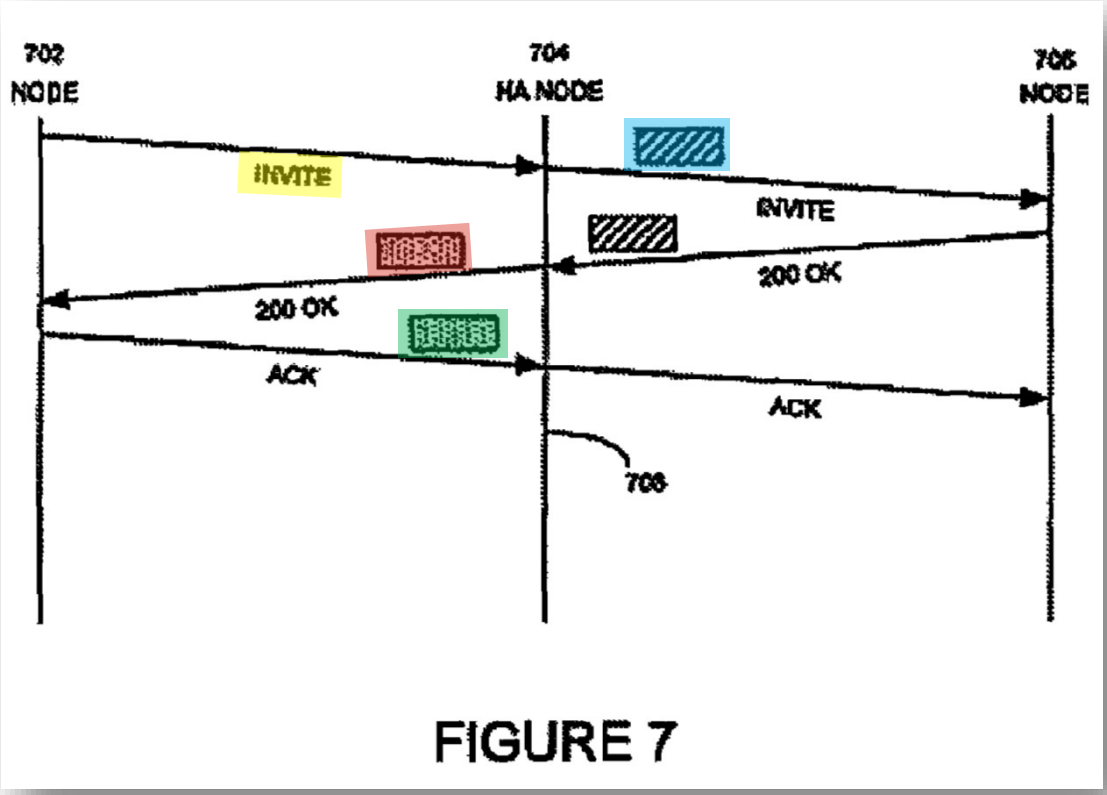


FIGURE 7

Ex. 1101, Fig. 7 (annotated).

47. In the above diagram, "high-availability node 704" exchanges SIP messages between "calling party node 702" and "called party node 706." *Id.*, 7:16-19. "The calling party node 702 sends an INVITE message"—highlighted in yellow—"to the HA node 704 to establish a connection with the called party node 706." *Id.*, 7:19-21. Before forwarding the INVITE message to the called party node 706, the HA node must "include the context data"—marked in blue—"in the TAG field of the INVITE message since this is the first message sent from the HA node 704 to the called party 706." *Id.*, 7:21-27. This is because "the TAG is included in all related messages and response messages" and "must not be altered once set." *Id.*,

7:6-10. Including the context data in the initial INVITE message allows the HA node "to store the context data representing the state of the protocol stack of the HA node." *Id.*, 7:23-27. In the above diagram, the context information is included in both the 200-OK response (red) returned to calling party node 702 and in the acknowledgement response (green) from calling party 702 to the HA node 704.

VI. CLAIM CONSTRUCTION

48. It is my understanding that in order to properly evaluate the '738 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.

49. I do not believe that any claim constructions would affect my analysis or conclusion that the claims are obvious over the asserted prior art.

VII. IDENTIFICATION OF HOW THE CLAIMS ARE UNPATENTABLE

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

51. 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. As detailed below, the scope and content of the Forissier prior art reference includes every element of the asserted claims without any meaningful difference between the teachings of those references and the elements of the asserted claims.

52. The Patent Owners have not identified any secondary considerations of nonobviousness so I have not considered any in my analysis.

53. I have considered the teachings in the prior art from the perspective of a POSITA and it is my conclusion that a POSITA would have found it obvious to combine the teachings of those known elements in the way they are recited in the Challenged Claims as of July 25, 2003. I do not believe any secondary considerations would alter my overall conclusion that the claims would have been obvious to a POSITA as of July 25, 2003. Nevertheless, I reserve the right to analyze any secondary considerations that the Patent Owners identify.

VIII. THE PRINCIPAL PRIOR ART

A. Overview of Forissier

54. U.S. Patent No. 7,936,763 to Forissier et al ("Forissier") (Ex-1105) was filed on May 18, 2004, and claims priority to foreign application European Patent Application 03291191 (published as EP1480407A1), which was filed on May 20,

2003. Ex-1106. I understand that Forissier is therefore prior art to the '738 Patent at least under pre-AIA 35 U.S.C. § 102(e).

55. Forissier is directed to "improvements in load-balancing systems and methods" in "the field of distributed processing systems." Ex. 1101, 1:5-7. A specific embodiment of Forissier is aimed at providing "a method of routing a message to one of a plurality of available processing system" through the use of a "destination identifier" in the message. Ex. 1105, Abstract. Forissier explains that the destination identifier both facilitates forward messages to the appropriate destination processing system and, in an embodiment where the destination identifier is determined to be missing, can also be inserted into the message to restore context regarding the appropriate destination for the message. Ex. 1105, Abstract, 2:42-61, 3:18-34.

56. In my opinion, Forissier is analogous to the '738 Patent because (i) they are both directed to message-based communication systems, (ii) Forissier directly addresses the problem described by the '738 Patent of overcoming the complexity and cost of "maintain[ing] a shared or common storage element" in a fault-tolerant message system, particularly as it relates to telephony applications (Ex. 1101, 4:34-36), and (iii) Forissier explains its teachings using an exemplary SIP message-flow diagram, just as the '738 Patent does.

57. *First*, Forissier and the '738 Patent are both directed to message-based communication systems. Ex. 1101, 1:3-6 ("The present invention . . . relates to a

method and system for providing fault tolerance in . . . **message-based communication systems**."); Ex. 1105, Abstract ("According to an embodiment of the present invention, there is provided a method of **routing a message to one of a plurality of available processing systems**, comprising the steps of detecting the presence of a destination identifier in the message, and where the presence of the destination identifier is detected, forwarding the message to the processing system identified thereby.").

58. *Second*, Forissier directly addresses a material problem described by the '738 Patent. For instance, the '738 Patent explains that it aims to overcome the complexity and cost of "maintain[ing] a shared or common storage element" in a fault-tolerant message system, particularly as applied to telephony systems. Ex. 1101, 4:34-36. Forissier similarly discusses that its invention is advantageous over the prior art because it alleviates the need for using traditional storage to retain context information about calls:

> Traditional load-balancing approaches, such as maintaining a database of context information for all active calls at a load-balancer, may therefore, not be suitably adapted for use with SIP for a number of reasons.
>
> Firstly, since the length of a SIP call is theoretically unbounded and the fact that the load-balancer must store call context information for the duration of each and every the call, it is possible that the load-balancer may become overwhelmed, especially if the number of simultaneous calls and the number of backend servers is large.

> Additionally, in order to clean up the stored context information, the load-balancer may be required to receive all incoming and outgoing SIP messages to determine when a call has finished in order to be able to remove the stored context information for that call when it is no longer required. These constraints may impact the performance capabilities of such a load-balancer and limit the message handling capabilities thereof.

Ex. 1105, 2:14-32. Notably, both the '738 Patent and Forissier contend that such traditional storage is inefficient for applications such as SIP. *Id.*; *see also*, Ex. 1101, 1:63-65 ("However, the requirement for a common storage area for storing context data adds to the complexity and cost of [telephony networks which switchover from active to standby] systems"—i.e., systems using SIP.). Forissier further explains that its approach "is particular[ly] useful for fault tolerance purposes as the load-balancer maintains full control of the routing of messages, for example, in the event of a backend server failing." Ex. 1105, 7:4-6.

59. *Third*, Forissier explains its teachings by example to the same technological application as the '738 Patent: SIP. For example, Forissier explains that an embodiment of its purported invention is disclosed by Figure 4 (annotated below), which is an exemplary SIP message-flow diagram:

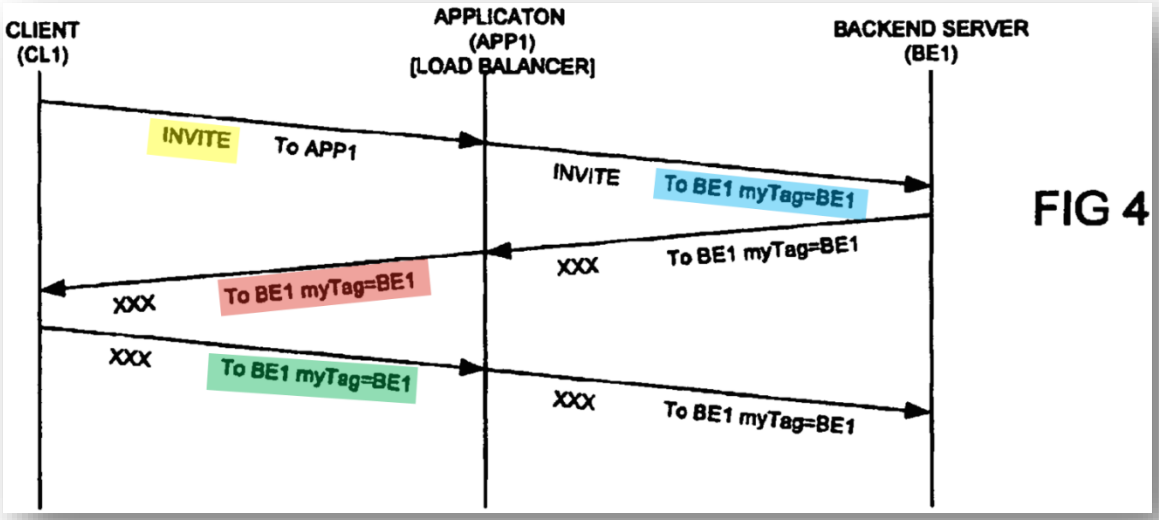


FIG 4

Ex. 1105, Fig. 4 (annotated). In this diagram, a SIP user agent (CL1) sends a SIP Invite message (yellow) to a load-balancer application (APP1), which inserts a tag ("myTag", blue) into the message that identifies a chosen backend server (BE1) and forwards the message to BE1. Ex. 1105, 6:11-20. The myTag information identifying BE1 as the destination server (red) is then included in the response to CL1. Ex. 1105, 6:2-23. If the SUP user agent at CL1 sends a further response related to the same call, the previously inserted myTag (green) will be included in the response. Ex. 1105, 6:23-25. APP1 then examines the response to determine the presence of the tag and, based on that myTag information, will forward the message to BE1. Ex. 1105, 6:25-33. This process greatly resembles that process disclosed in the Figure 7 of the '738 Patent, reproduced below:

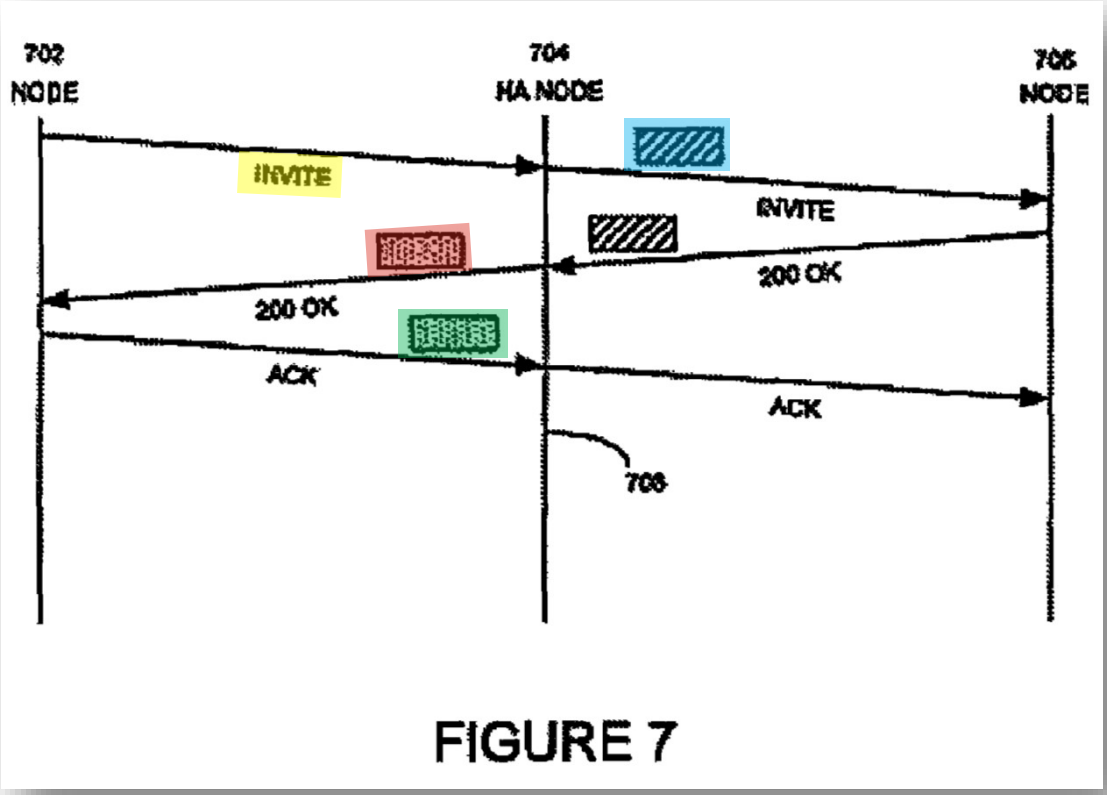


FIGURE 7

Ex. 1105, Fig. 7 (annotated).

IX. CLAIMS 1-10 AND 33 ARE UNPATENTABLE

A. Ground 1: Claims 1–10 and 33 Would Have Been Obvious In View of Forissier

60. Claims 1-10 and 33 would have been obvious in view of Forissier. To the extent one may assert that Forissier does not teach or suggest the following limitations, each would have been obvious to a POSITA in view of the teachings and suggestions in Forissier.

1. Claim 1

i. *[1.0] A method of storing context information in an outgoing message sent from a node including a computing device using a protocol stack having at least one layer, comprising:*

61. To the extent the preamble is limiting, Forissier discloses "*a method of storing context information in an outgoing message sent from a node including a computing device using a protocol stack having at least one layer*."

62. Forissier discloses "*storing context information in an outgoing message sent from a node*." Ex. 1105, 1:51-56, 2:14-32. Forissier aims to eliminate the storage of context information for calls in a database by storing the context information in an outgoing message. As Forissier explains, generally prior-art load balancers were "required to store some context information, for example the call ID of each message." *Id.*, 1:51-56. The prior-art approaches included "maintaining a database of context information for all active calls may…not be suitably adapted" for then-emerging telephony protocols such as SIP. *Id.*, 2:14-32.

63. The solution in Forissier is similar to how the '738 patent describes "the requirement for a common storage area for storing context data adds…complexity and cost [for telephonic] systems, [and it is the] aim of the ['738] invention…to overcome…the above-mentioned problem[]." Ex. 1101 at 1:63-67.

64. One resounding problem that Forissier identifies with prior-art approaches was the complexity in "maintaining a database of context information

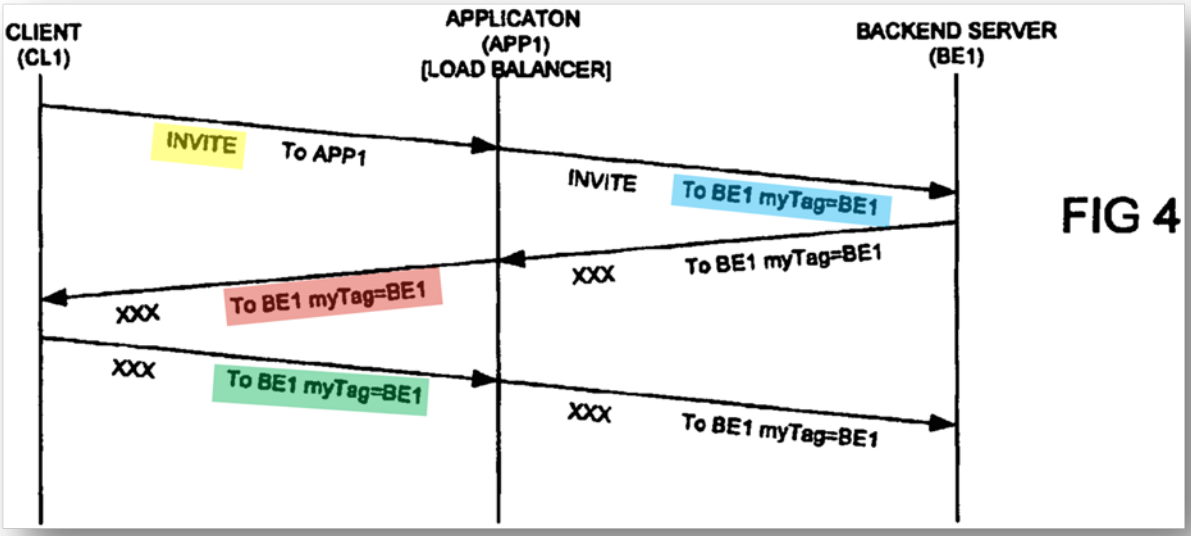
for all active calls," given the potential length of SIP calls would potentially "overwhelm[]" the load balancer, "impact[ing its] performance capabilities…and limit[ing its] message handling capabilities." *Id*., 2:15-16, 2:22, 2:29-31.

65. Forissier's invention "overcomes at least some of the above-mentioned problems" with respect to storing context information. *Id.*, 2:33-35. This is done by the addition of a "destination identifier" in the outgoing message (*storing context information in an outgoing message*) as opposed to storing context information about active calls in a database:

> Where the presence of the destination field is not detected the further steps of determining a destination processing system for processing the message, **inserting into the message a destination identifier identifying the determined destination processing system**, and forwarding the message to the determined processing system may be performed.
>
> **Advantageously, this removes the need for a load-balancing element to maintain a database of all current calls being processed, thus reducing the processing load of the load balancer**.

Id., 2:42-52, Abstract, 5:6-15, 7:8-15.

66. Forissier illustrates a tangible implementation of the above novelty, using an exemplary SIP message-flow diagram in Figure 4 (annotated below):



Id. at Figure 4 (annotated). "[S]hown in FIG. 4, a SIP user agent (CL1) sends a SIP INVITE message (yellow) to [a load-balancer] application (APP1)." *Id*., 6:11-12. While Forissier does not explicitly use the term *node*, a POSITA would know that it is common within networked messaging environments with a client-server-load balancer architecture for the clients, load balancers and backend servers to be considered nodes. According to the IEEE Standard Glossary of Computer Networking Terminology, "[i]n networking, [a node is] a point or a junction in a transmission system where lines or trunks from one or more systems meet [and]…[i]n data communications, [a node is] a device or station that implements some part of the communications protocol." Ex. 1107, 3.567 Clearly, the load-balancer application is a *node*, it "receives a message (step **702),** for example a SIP INVITE message, from, for example, a SIP user agent **102.** When a message arrives at the load- balancer relating to a new call, the load-balancer chooses (step **710)** one

of the available backend servers **206, 208 or 210** to send the message to using a suitable load-balancing algorithm," Ex. 1105, 4:67-5:5. The load-balancer comprises an application running on *a node including* one or more *computing device[s]*, such as network servers, being that it is "integral to the B2BUA 112,…[and] may be located externally therefrom, for example in a distributed manner." *Id.*, 4:39-41.

67. As depicted in Figure 4 of Forissier, when application APP1 receives a message, the load-balancer "analyses the message to determine the presence of an inserted tag. If no tag is found, the load-balancer chooses a suitable backend server (BE1) and inserts a tag, for example myTag [(blue)], identifying the chosen backend server…and forwards the message to the chosen backend server." Ex. 1105, 6:14-20. The myTag information identifying BE1 as the destination server persists throughout all communications related to the call, for instance in the response to CL1 (red) and SIP user agent at CLI's further response (green). *Id*., 6:2-23. Forissier notes, "the tag is preferably inserted into the message" (*storing context information in an outgoing message*) "such that the tag will be included in all future messages sent in response to the message. For example, in SIP the message may be suitably inserted as an extension header." *Id*., 5:12-15. For the SIP user agent's further responses, the load-balancer examines the message "to determine whether there is a tag present…If there is a tag present, the load-balancer forwards the message to the backend server" (*node including a computing device*) "indicated by the tag," in the

Figure 4 example to BE1 based on information in myTag. *Id*., 6:25-29. The "destination identifier" contained in the myTag information disclosed in Forissier is the "*stor[ed] context information*" as recited in the claim as it enables SIP messages to reach the correct destination (e.g., BE1) by utilizing application-specific data from the load-balance application The '738 Patent confirms this conclusion, noting that for telephony networks, "*context information*" may include "any application specific data related to individual calls" and "a destination address of a SIP call, which would enable SIP messages related to that call to be routed to the correct destination." Ex. 1101, 1:53-56, 7:1-3. The "destination identifier" in Forissier is an example of a "destination address" mentioned in the '738 patent. It "determin[es a] destination processing system" and uses said destination identifier to forward messages to that determined destination processing system. *Id.*, Abstract.

68. Furthermore, Forissier "provide[s] a method of routing a message to one of a plurality of available processing systems, comprising the steps of detecting the presence of a destination identifier in the message, and where the presence of the destination identifier is detected, forwarding the message to the processing system identified thereby." *Id*., 2:36-42.

69. In my opinion, a POSITA would understand Forrisier to disclose "*a method for storing context information in an outgoing message sent from a node including a computing device,*" given that Forissier discloses a load-balancer

application inserting a destination identifier into a message sent from a client or server ("*a method for storing context information in an outgoing message sent from a node*") where the clients and servers are deployed on nodes such as network servers (*"node including a computing device*").

70. Forissier also discloses *using a protocol stack having at least one layer*. A POSITA would know that the SIP protocol can be carried by several transport layer protocols which is supported by Forissier's disclosure that it "is an application-layer control or signaling protocol for establishing, modifying, and terminating real-time calls and conferences over, primarily, Internet protocol (IP) networks." *Id.*, 2:3-5; Ex-1116, 1-2. As described in the '738 Patent, a POSITA would also have known that the SIP routing performed by Forissier's load-balancer application involves multiple layers:

> The systems 202 and 204 may be any kind of system or network nodes which communicate with one another using messages, across a network, **using one or more layers of a protocol stack**. For example, as shown in FIG. 3, systems 202 and 204 may represent, respectively, a SIP user agent 300 and a SIP back-to-back user agent (B2BUA) 302 of a session initiation protocol (SIP) network.

Ex. 1101, 4:63-5:2.

According to the IEEE Standard Glossary of Computer Networking Terminology, a "layer…[i]n the OSI model, [is] one of a collection of network-processing functions representing a level of a hierarchy of function….[and a] protocol…[is a] formal set

of conventions governing the format and relative timing of message exchange in a computer system" Ex. 1107, 3.451, 3.655. A POSITA would know the OSI model (*protocol stack having at least one layer*) is used to understand and categorize the functions ("network-processing" adding destination information) of the SIP *protocol*.

71. Accordingly, Forissier thus discloses "*a method of storing context information in an outgoing message sent from a node including a computing device using a protocol stack having at least one layer*."

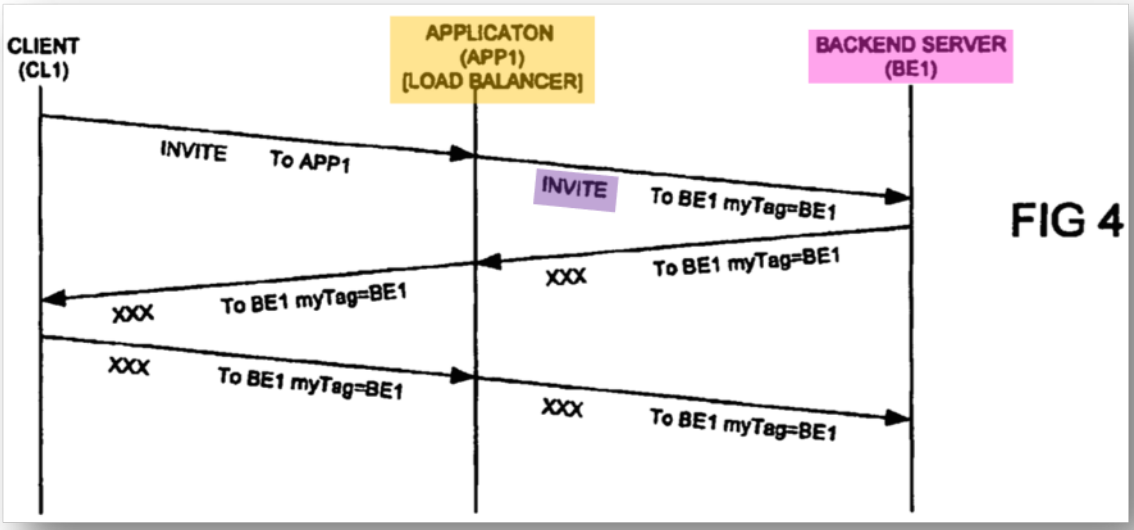
ii. *[1.1] providing, by the computing device, the outgoing message from an application to a layer of the protocol stack, the outgoing message is destined for an application on a destination node;*

72. Forissier discloses "*providing, by the computing device, the outgoing message from an application to a layer of the protocol stack, the outgoing message is destined for an application on a destination node*." *See, e.g.*, Ex. 1105, 2:2-5, 6:11-14.

73. Forissier describes the "nodes" (e.g., CL1, APP1, and BE1 from Figure 4) as including telephony applications, particularly SIP applications with messages passed between them. *Id*., 6:11-14. SIP is "an application-layer control or signaling protocol" for communicating over computer networks, "primarily, Internet protocol (IP) networks." *Id.*, 2:2-5. A POSITA would understand that Forissier explains that the SIP messages are *outgoing message[s]…*destined *for an application on a*

destination node (e.g., CL1 and BE1), because it discloses that "[a]s shown in FIG. 4, a SIP user agent (CL1) sends a SIP INVITE message to an application (APP1)" (*outgoing message*) "and forwards the message to the chosen backend server [(BE1)]." (*application on a destination node*) "When the backend server sends a response message to the user agent…the load-balancer 302…forwards the message to the SIP user agent (CL1)". *Id*., 6:11-23.

74. Figure 4 from Forissier (annotated below), illustrates the load-balancer application APP1 (orange) sending an *outgoing message* (a SIP Invite) (purple) *destined for an application* on BE1 (pink):



Id. at Figure 4 (annotated); *See also, Id.*, 6:11-14. The application on BE1 may, for instance, facilitate SIP calls and/or "provide for value-added services, such as…pre-paid billing, authentication, or credit-card authorizations". *Id.*, 4:10-41.

75. A POSITA would have understood that the message (INVITE) (*outgoing message*) originating *from an application* on CL1 and APP1 is *destined for an application* on APP1 and BE1 (*destination node*), respectively. Forissier further describes various telephonic applications with messages sent from CL1 to APP1 to BE1. *Id.*, 4:10-41, 6:11-14. A POSITA would have also understood, that in sending messages to APP1 and BE1 using the SIP protocol and myTag information, the applications CL1 and APP1 would have to *provid[e]…the outgoing message…to a layer of the protocol stack.* This is obvious in view of Forissier's disclosures and a POSITA's experience, since this is how messages, including SIP messages over IP networks, are sent by applications across a network: *provid[ing]* those messages *to a layer of the protocol stack*. As described in Section I.A.i, SIP consists of *layer[s] of the protocol stack*, where the layers perform "network-processing functions," such that messages can be sent across the network. Indeed, a POSITA would be familiar with the OSI model, and would know that one of the layers, layer 3, is the network layer. *See, e.g.*, Ex-1116, 1 ("Layer 3 is the Network Layer, providing a means for communicating open systems to establish, maintain and terminate network connection") ("Layer 4 is the Transport Layer, and is where TCP lives."), 2 ("Consider the following example. An application forms a packet of data to be sent; this takes place at Layer 7. As the packet descends the stack, it is wrapped in headers and trailers, as required by the various protocols, until, having reached Layer 1, it is

transmitted as a frame across the medium in use. Upon reaching device B, it reascends the stack, as the device strips off the appropriate headers and trailers, delivering just the application data to the application.").

76. A POSITA would understand that by facilitating sending the SIP Invite, CL1 and APP1, which are deployed on computing devices, provide the SIP Invite message to the SIP application layer and it passes through other layers of the protocol stack (e.g., transport and network layers).

77. Accordingly, Forissier teaches or suggests "*providing, by the computing device, the outgoing message from an application to a layer of the protocol stack, the outgoing message is destined for an application on a destination node*."

iii. *[1.2] selectively indicating to the layer of the protocol stack that context information is to be obtained for that layer;*

78. Forissier discloses "*selectively indicating to the layer of the protocol stack that context information is to be obtained for that layer*." *See, e.g.*, Ex. 1105, 5:29-48, 6:14-20.

79. Forissier discloses that the load-balancer application on the APP1 node selectively obtains context information, by including a tag, based on whether or not the message already includes a tag. As stated, "[w]hen the load balancer receives the INVITE message, it analyses the message to determine the presence of an inserted tag. If no tag is found, the load-balancer chooses a suitable backend server (BE1)

and inserts a tag, for example myTag, identifying the chosen backend server into the message, and forwards the message to the chosen backend server." *Id.*, 6:14-20. If the message is new, "a suitable backend server is chosen [by the load-balancer application] to process the message." *Id.*, 5:41-48.

80. A POSITA would understand that it is common practice to pass information between the load-balancing application on APP1 and the network, especially for SIP messages because doing so is required for the load-balancer to analyze the messages. As stated above, the load-balancer analyses the message to determine if a tag (*context information*) *is to be obtained* and inserted by a *layer of the protocol stack*, it intentionally will choose to add a tag to some messages (those without tags) and not others (messages with tags), thus *selectively*[1] *indicating…that context information is to be obtained*. As discussed in Section I.A.i, in a *protocol stack* layers perform different network-processing functions, including *obtain[ing] context information*, and in the case of Forissier, the load balancer indicates to the respective layer whether or not to *obtain[]* the *context information* for insertion.

81. Accordingly, Forissier teaches or suggests "*selectively indicating to the layer of the protocol stack that context information is to be obtained for that laye*r."

[1] Cambridge Dictionary Definition of "selective": intentionally choosing some things and not others

iv. *[1.3] obtaining, by the computing device, context information in accordance with the indication; and*

82. Forissier discloses "*obtaining, by the computing device, context information in accordance with the indication*." *See, e.g.*, Ex. 1105, 4:65-5:12, 6:16-20, Fig. 8.

83. Forissier teaches that the load-balancer application determines a suitable backend server for handling the call when it receives a new SIP Invite message. *Id.*, 4:65-5:9. For a new SIP Invite message, "no tag is found, [so] the load-balancer chooses a suitable backend server (BE1) and inserts a tag, for example myTag, Identifying the chosen backend server, into the message, and forwards the message to the chosen backend server." *Id.*, 6:16-20. In determining the suitable backend server, the context information or tag "[p]referably…contains sufficient information to enable the load-balancer to route the message without requiring a further call context." *Id.*, 5:6-12.

84. A POSITA would understand that this process of "choos[ing] a suitable backend server (BE1) and insert[ing] a tag," only when "no tag is found," would first require *obtaining…the context information in accordance with the indication* of no tag present because the process must obtain the necessary context information in order to choose a correct, i.e. suitable, backend server and insert a correct tag. Finally, the load-balancer application and other SIP layers reside on APP1, which as

discussed in Section I.A.i is a node composed of *computing device[s]* that are used to *obtain* said *context information*.

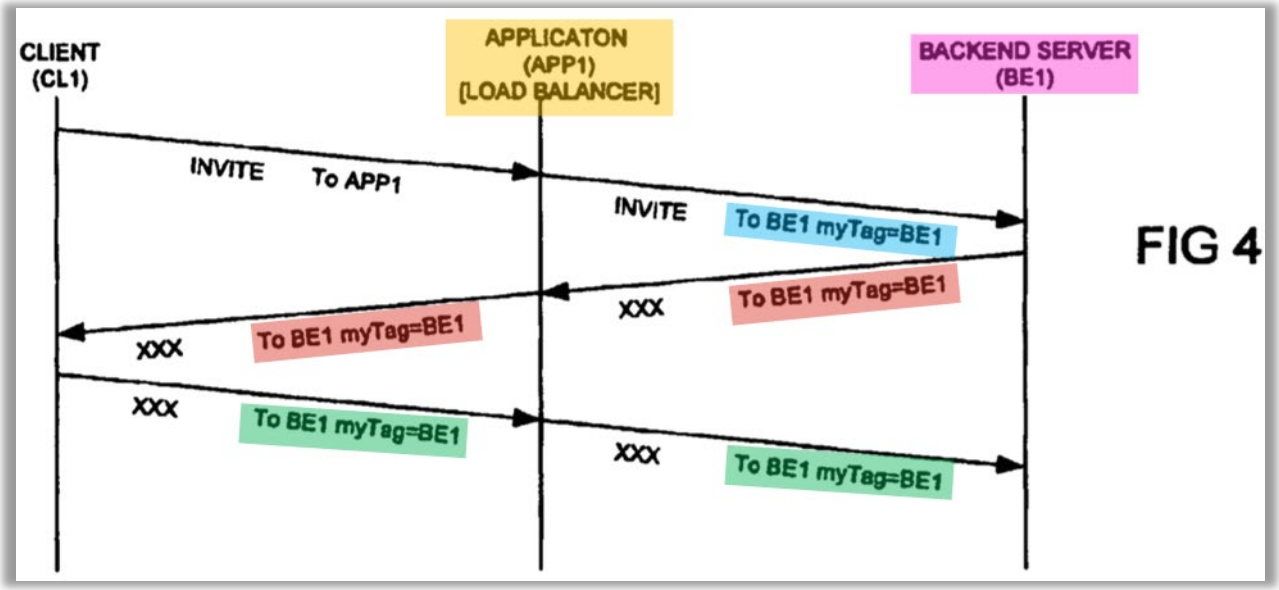
85. Accordingly, Forissier discloses "*obtaining, by the computing device, context information in accordance with the indication*."

v. *[1.4] adding, by the computing device, the obtained context information to the outgoing message such that a response, received from the destination node, to the outgoing message contains the obtained context information.*

86. Forissier discloses "*adding, by the computing device, the obtained context information to the outgoing message such that a response, received from the destination node, to the outgoing message contains the obtained context information*." *See, e.g.*, Ex. 1105, 5:6-15, 6:11-33, Fig. 4.

87. Forissier discloses *adding* a tag *contain[ing]* previously *obtained context information* about the chosen backend server into the SIP Invite (*outgoing message*) and including the tag in all subsequent messages in response to the message (*a response, received from the destination node, to the outgoing message contains the obtained context information*). *Id.*, 5:12-15, 6:14-20. Forissier explicitly mentions this tag (*obtained context information*) should contain "sufficient information to enable the load-balancer to route the message without requiring a further call context" and is "inserted into the message such that the tag will be included in all future messages sent in response to the message." *Id.*, 5:12-15, 6:14-

20. Figure 4 (annotated below), illustrates APP1 (orange) adding the contextual information as a tag in the SIP Header ("myTag") (blue) to the outgoing SIP Invite (*outgoing message*) containing the identity of the backend server (BE1) (pink)—and additionally future responses (e.g., those highlighted red (*response, received from the destination node*) and green) contain the *context information* about BE1 stored in the "myTag" header field:



Id. at Figure 4 (annotated); Forissier states, "[t]he load-balancer 302 inserts" (*add[s]*) "a tag" (*obtained context information*) "in the SIP message (step 714)," (*to the outgoing message*) "for example myTag, indicating the identity of the chosen backend server" (*obtained context information*) "and forwards the message to the chosen backend server (step 716). Preferably the tag contains sufficient information to enable the load-balancer to route the message without requiring a further call

context. Additionally, the tag" (*obtained context information*) "is preferably inserted into the message such that the tag will be included in all future messages sent in response to the message" (*a response, received from the destination node…contains the obtained context information*). "For example. In SIP the message may be suitably inserted as an extension header." *Id.*, 6:6-15.

88. Accordingly, Forissier discloses "*adding, by the computing device, the obtained context information to the outgoing message such that a response, received from the destination node, to the outgoing message contains the obtained context information*."

2. Claim 2

i. *[2.0] The method of claim 1, further comprising adding context information obtained from a different protocol stack layer to the outgoing message.*

89. Forissier teaches or suggests "*[t]he method of claim 1, further comprising adding context information obtained from a different protocol stack layer to the outgoing message*."

90. For the reasons I explain above (*see supra* Section IX.1), Forissier discloses the method of Claim 1.

91. Forissier teaches or suggests "*adding context information obtained from a different protocol stack layer to the outgoing message*." *See e.g.*, Ex. 1105, 2:3-5, 4:6-9, 5:1-15, Fig. 8. As mentioned in Section I.A.ii, it would have been

known by a POSITA that SIP messages are sent through multiple layers of the protocol stack, including an application layer and a transport layer. *See* Ex-1116. In Forissier, a POSITA would understand that the *context information [is] obtained* by the load-balancer application at APP1 at the application layer, which is *a different protocol stack layer* than the transport layer, where the *context information* is *add[ed]…to the outgoing message* (as the destination to facilitate delivery):

> When a message arrives at the load-balancer relating to a new call, the load-balancer chooses (step 710) one of the available backend servers 206, 208 or 210 to send the message to using a suitable load-balancing algorithm, as will be appreciated by those skilled in the art.
>
> **The load-balancer 302 inserts a tag in the SIP message (step 714), for example myTag**, **indicating the identity of the chosen backend server** and forwards the message to the chosen backend server (step 716). Preferably the tag contains sufficient information to enable the load-balancer to route the message without requiring a further call context.
>
> Additionally, the tag is preferably inserted into the message such that the tag will be included in all future messages sent in response to the message. **For example. In SIP the message may be suitably inserted as an extension header.**

Id., 5:1-15; *See also, id.*, 2:3-5 ("SIP is an application-layer control or signaling protocol for establishing, modifying, and terminating real-time calls and conferences over, primarily, Internet protocol (IP) networks.") Ex-1117, 10 ("SIP is an application-layer control protocol that can establish, modify, and terminate multimedia sessions (conferences) such as Internet telephony calls"); Ex. 1116, 1-2

(describing protocol stacks and layers). As was previously mentioned, Forissier discloses this adapted SIP methodology, "SIP is an application-layer control or signaling protocol for establishing, modifying, and terminating real-time calls and conferences over, primarily, Internet protocol (IP) networks" and "shown in FIG. 4, a SIP user agent (CL1) sends a SIP INVITE message to an application (APP1). As described above, the application is an application using uses a load-balancer and a number of distributed backend servers. When the load-balancer receives the INVITE message, it analyses the message to determine the presence of an inserted tag." *Id.*, 2:3-5, 6:11-14; Ex-1116, 1-2 (describing sending application data through layers).

92. Forissier further discloses the details of the method in Figure 8 (annotated right), which shows a flow diagram outlining example processing steps of a load-balancer operating in accordance with an embodiment of the present invention. *Id.*, 4:6-9, Fig. 8. Following the check determining there is no message tag and call context, at step 710 (yellow), "the load -balancer chooses … one of the available backend servers" and

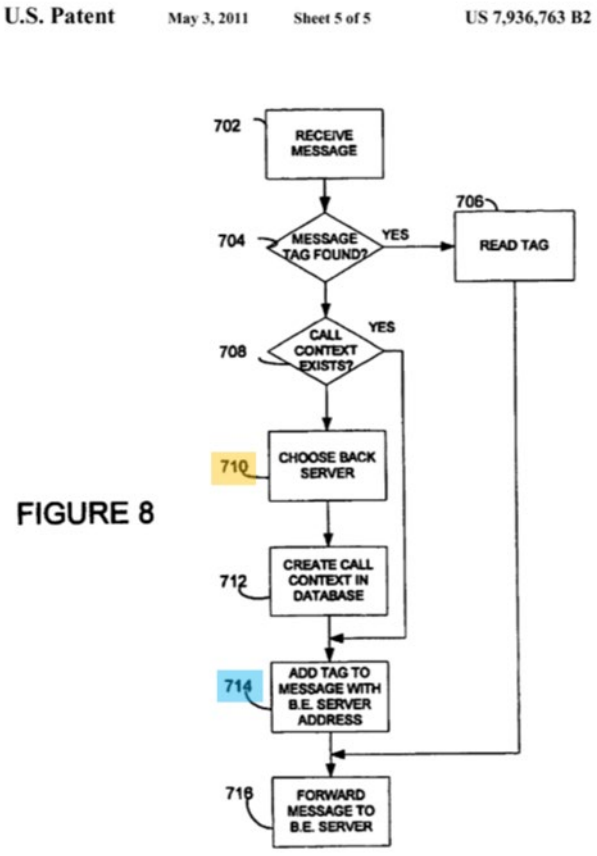


FIGURE 8

at step 714 (blue), "inserts a tag in the SIP message" (such as in the SIP extension header) that identifies the chosen backend server. *Id*., 5:1-6, 5:6-15. A POSITA would understand that the load-balancer application at the application layer identifies the chosen backend server. Additionally, a POSITA would know that the transport layer is responsible for adding information to manage communication sessions, such as routing information, including the address (identity) of the chosen backend server. Ex-1116, 1-2 (describing protocol stacks and layers). Thus, the *adding [of] context information…to the outgoing message* is done in one layer and *context information* is *obtained from a different protocol stack layer*.

93. To the extent one may assert that what was discussed does not sufficiently disclose "*adding context information obtained from a different protocol stack layer to the outgoing message*," it is my understanding that a POSITA would have known understood that the application layer of the load-balancing application obtains the identity of the chosen backend server and the transport layer adds that context information to the message. This is so because a POSITA would know the differences between the application and transport layer. Ex-1116, 1-2 (describing protocol stacks and layers). The application layer is responsible for network services to end-user applications, such as load-balancing, while the transport layer is responsible for managing communication sessions, including adding routing information to messages. It would have then been obvious that the *context*

information [is] obtained from a different protocol stack layer (application layer) than where it is *add[ed]* to facilitate delivery of the message (transport layer). Ex-1116, 1-2 (describing protocol stacks and layers).

94. Accordingly, Forissier teaches or suggests "*[t]he method of claim 1, further comprising adding context information obtained from a different protocol stack layer to the outgoing message*."

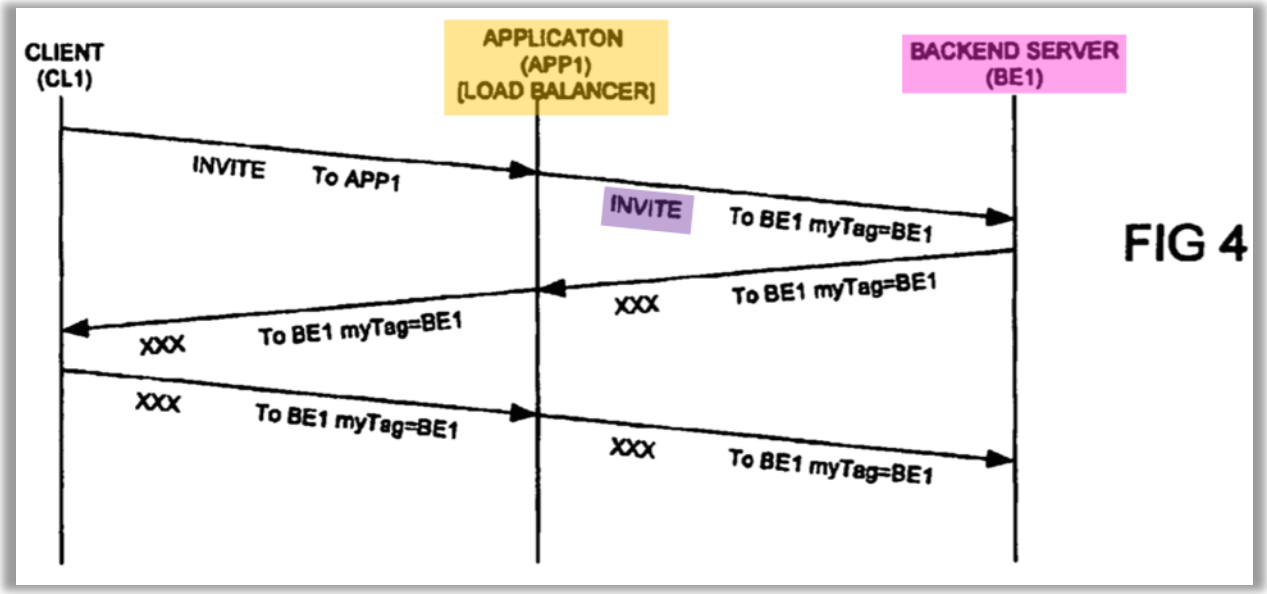
3. Claim 3

i. *[3.0] The method of claim 1, wherein the outgoing message is sent from the node is to a remote node across a network.*

95. Forissier discloses "*[t]he method of claim 1, wherein the outgoing message is sent from the node is to a remote node across a network*."

96. For the reasons I explain above (*see supra* Section IX.1), Forissier discloses the method of Claim 1.

97. Forissier discloses that "*the outgoing message is sent from the node is to a remote node across a network*." *See, e.g.*, Ex. 1105, 4:37-41, 6:14-20, Fig. 4. Shown in Figure 4 (annotated below), is the outgoing message (the SIP Invite, purple) sent from the node APP1 (yellow) to a backend server node BE1 (pink):



Id. at Figure 4 (annotated); *See also, Id.*, 6:14-20 ("When the load balancer receives the INVITE message, it analyses the message to determine the presence of an inserted tag. If no tag is found, the load-balancer chooses a suitable backend server (BE1) and inserts a tag, for example myTag, Identifying [sic] the chosen backend. [sic] server into the message, and forwards the message to the chosen backend server."). Forissier further explains that the back-end servers "may be located externally therefrom [the load balancer], for example in a distributed manner" and that SIP is a "protocol for [communication] over, primarily, Internet protocol (IP) networks." *Id.*, 4:37-41, 2:3-5. A POSITA would know that distributed nodes in IP networks are generally remote, thus the backend server BE1 may be remote from the load-balancer APP1 (i.e., *across a network*). So, a POSITA would have understood that *the outgoing message* (SIP Invite message) *is sent from the node* (load balancer application, APP1)…*to a remote node across a network* (backend server, BE1)

98. Accordingly, Forissier discloses *"[t]he method of claim 1, wherein the outgoing message is sent from the node is to a remote node across a network*."

4. Claim 4

i. *[4.0] The method of claim 1, used with a message-based communications system.*

99. Forissier discloses *"[t]he method of claim 1, used with a message-based communications system*."

100. For the reasons I explain above (*see supra* Section IX.1), Forissier discloses the method of Claim 1.

101. Forissier discloses "*use[] with a message-based communications system*." *See, e.g.*, Ex. 1105, 2:3-5, 2:9-14, 2:24-32, 3:8-9, 4:13-18, 4:42-51, 4:66-5:28, 6:11-33, 6:41-54, 6:58-61, 7:1-3, 7:7-15. Specifically, Forissier discloses its invention being used with SIP, "an application-layer control or signaling protocol for establishing, modifying, and terminating real-time calls and conferences over, primarily, Internet protocol (IP) networks." *Id.*, 2:3-5. A POSITA would know that calls can be considered part of message-based communication systems, given that signaling protocols, like SIP, use messaged-based communication to establish, manage, and terminate calls, Unified Communications integrate calls with other message-based communication methods, VoIP and SIP trunking use internet-based message exchanges to route calls through the network. There are numerous examples where Forissier discloses SIP as messaged-based communication and explains at

length sending ("the SIP user agent sends a further message"), receiving ("the load-balancer receives the INVITE message"), and analyzing SIP messages ("the load-balancer 302 may comprise suitable logical or functional elements such as a message analyzer"). *Id.*, 2:9-14, 2:24-32, 3:8-9, 4:13-18, 4:42-51, 4:66-5:28, 5:66-6:1, 6:11-33, 6:41-54, 6:58-61, 7:1-3, 7:7-15. Furthermore, the '738 Patent itself, identifies SIP as a message-based communication system, disclosing "SIP messages related to that [SIP] call to be routed to the correct destination." Ex. 1105, 7:2-3, 7:15-27, Fig. 7.

102. Accordingly, Forissier discloses "*[t]he method of claim 1, used with a message-based communications system*."

5. Claim 5

i. *[5.0] The method of claim 1, wherein the step of obtaining context information obtains context information related to the outgoing message.*

103. Forissier discloses *"[t]he method of claim 1, wherein the step of obtaining context information obtains context information related to the outgoing message*."

104. For the reasons I explain above (*see supra* Section IX.1), Forissier discloses the method of Claim 1.

105. Forissier discloses "*obtaining context information where the context information is related to the outgoing message*." *See, e.g.*, Ex. 1105, 5:9-15, 6:14-

20. As previously discussed, Forissier mentions that the load balancer "chooses a suitable backend server (BE1) and inserts a tag" (*obtains* and adds *context information*), "for example myTag, Identifying the chosen backend server." *Id.*, 6:14-20. The *obtain[ed]* information, added to the outgoing message (i.e., the SIP Invite from APP1 to BE1), contains information *related to the outgoing message*, including, the destination of the outgoing message, that is, "sufficient information to enable the load-balancer to route the message without requiring … further call context." *Id.*, 5:9-15; *see also id.* 1:40-42 ("context information must be maintained by the load-balancer in order to determine to which backend server a message should be forwarded to"); Ex-1101, 7:1-3 ("the context data may include a destination address of the SIP call, which would enable the SIP messages related to that call to be routed to the correct destination"). It would be clear to a POSITA, that the identity of the chosen backend server (i.e., the destination of the outgoing message) is *context information related to the outgoing message*.

106. Accordingly, Forissier discloses *"[t]he method of claim 1, wherein the step of obtaining context information obtains context information related to the outgoing message*."

6. Claim 6

i. *[6.0] The method of claim 1, wherein the step of adding the obtained context information appends the obtained context information to a separate field of the message.*

107. Forissier discloses *"[t]he method of claim 1, wherein the step of adding the obtained context information appends the obtained context information to a separate field of the message*."

108. For the reasons I explain above (*see supra* Section IX.1), Forissier discloses the method of Claim 1.

109. Forissier discloses "*adding the obtained context information [by] append[ing] the obtained context information to a separate field of the message*." *See, e.g.*, Ex. 1105, 5:6-16, 6:14-29, 3:8-9 Forissier mentions in the instance of SIP messages, "[t]he destination identifier are preferably inserted into an extension header of [the] SIP message." *Id.*, 3:8-9, Fig. 4, Fig. 5, Fig. 6, Fig. 7, Fig. 8; Ex-1117, 11 ("The INVITE request contains a number of header fields. Header fields are named attributes that provide additional information about a message."). A POSITA would know that an extension header is a field used to add additional information or capabilities to SIP messages beyond what is provided by the standard headers. Forissier also discloses that the obtained context information (e.g., the identity of the chosen backend server BE1) should be appended as a tag to the SIP Invite, preferably

in the SIP extension header, such that the tag will be retained in future related messages:

> The load-balancer 302 inserts a tag in the SIP message (step 714), for example myTag, indicating the identity of the chosen backend server and forwards the message to the chosen backend server (step 716). Preferably **the tag contains Sufficient information to enable the load-balancer to route the message without requiring a further call context. Additionally, the tag is preferably inserted into the message such that the tag will be included in all future messages sent in response to the message. For example. In SIP the message may be suitably inserted as an extension header**.

Id., 5:6-16; *See also Id.*, 6:14-29, 3:8-9, Fig. 4, Fig. 5, Fig. 6, Fig. 7, Fig. 8 (illustrating step 714 as "add tag to message with b.e. server address"). A POSITA would recognize that since "that tag is preferably inserted into the message such that the tag will be included in all future messages sent in response to the message," "the tag contains sufficient information to enable…rout[ing] the message without requiring a further call context," and message fields serve a distinct purpose and help in the proper processing of the message, that the proper implementation of a tag (*obtained context information*) in the message is through (*append[ing] to*) a *separate field of the message*.

110. Accordingly, Forissier discloses "*[t]he method of claim 1, wherein the step of adding the obtained context information appends the obtained context information to a separate field of the message*."

7. Claim 7

i. *[7.0] The method of claim 1, for use with a session initiation protocol (SIP) network.*

111. Forissier discloses *"[t]he method of claim 1, for use with a session initiation protocol (SIP) network*."

112. For the reasons I explain above (*see supra* Section IX.1), Forissier discloses the method of Claim 1.

113. As explained for claim 1 above (*see supra* Section IX.1), Forissier discloses "*use of its methods and systems with a session initiation protocol (SIP) network*." Forissier provides a number of examples, "[t]he method may be used in conjunction with a session initiation protocol (SIP) based network," "[t]he load-balancing system may be adapted for use in a session initiation protocol (SIP) based network," and "[t]he load-balancer 302 inserts a tag in the SIP message (step 714)" when the "SIP user agent (CL1) sends a SIP INVITE message to an [a load-balancer] application (APP1)." Ex. 1105, 3:6-7, 3:4-45, 5:6-7, 6:11-12. Many of the exemplary embodiments utilize a SIP-based system and claim 7 states the method of detecting and inserting a destination identifier "for use in a session initiation protocol (SIP) based network." *Id.*, Claim 7.

114. Accordingly, Forissier discloses *"[t]he method of claim 1, for use with a session initiation protocol (SIP) network*."

8. Claim 8

i. *The method of claim 7, wherein the step of adding the obtained context information appends the obtained context information to a SIP TAG field.*

115. Forissier discloses *"[t]he method of claim 7, wherein the step of adding the obtained context information appends the obtained context information to a SIP TAG field*."

116. For the reasons I explain above (*see supra* Section IX.7), Forissier discloses the method of Claim 7.

117. As explained above for claims 1 and 6 (*see supra* Section IX.1 and IX.6), Forissier discloses *adding the obtained context information* (i.e., the destination identifier identifying the chosen backend server) by *append[ing] the obtained context information into a SIP TAG field* (i.e., "MyTag"). *See, e.g.*, Ex. 1105, 5:6-9, 6:14-20, Fig. 8. Forissier explicitly discloses that "[t]he load-balancer 302 inserts a tag in the SIP message (step 714), for example 'myTag', indicating the identity of the chosen backend server and forwards the message to the chosen backend server (step 716)." *Id.*, 5:6-9. From the above analysis of Claim 6, a POSITA would know that the tag goes into a specific SIP TAG field. *See* Ex-1117, 35 ("Note that in some instances, header fields that have been communicated in the contact address may instead append to existing request header fields in the original redirected request. As a general rule, if the header field can accept a comma-

separated list of values, then the new header field value MAY be appended to any existing values in the original redirected request."), 82, 115 ("The tag parameter is used in the To and From header fields of SIP messages. It serves as a general mechanism to identify a dialog, which is the combination of the Call-ID along with two tags, one from each participant in the dialog."). The load balancer *appends the obtained context information*, "[w]hen the load balancer receives the INVITE message, it analyses the message to determine the presence of an inserted tag. If no tag is found, the load-balancer chooses a suitable backend server (BE1) and inserts a tag, for example myTag, Identifying the chosen backend. [sic] server into the message, and forwards the message to the chosen backend server."*Id.*, 6:14-20, Fig. 8. Furthermore, "[t]he destination identifier" (*obtained context information*) "are preferably inserted into an extension header of a SIP message" – extension headers are appended to the basic headers. *Id.*, 3:8-9. A POSITA would have understood, given that (1) the load balancer adds the obtained context information ("myTag") (2) this context information is inserted into a separate field and (3) the context information is preferably inserted into an extension header (*append[ed]* to the basic header) that the *obtained context information appends the obtained context information to a SIP TAG field*.

118. Accordingly, Forissier discloses "*[t]he method of claim 7, wherein the step of adding the obtained context information appends the obtained context information to a SIP TAG field*."

9. Claim 9

i. *The method of claim 7, wherein the step of adding the obtained context information appends the obtained context information to a SIP extension header.*

119. Forissier discloses *"[t]he method of claim 7, wherein the step of adding the obtained context information appends the obtained context information to a SIP extension header*."

120. For the reasons I explain above (*see supra* Section IX.7), Forissier discloses the method of Claim 7.

121. As explained above for claims 1, 6, and 8 (*see supra* Section IX.1, IX.6, and IX.8), Forissier discloses that "*adding the obtained context information appends the obtained context information to a SIP extension header*." *See, e.g.*, Ex. 1105, 3:8-9, 3:49-50, 5:14-15. Forissier explicitly mentions that it is preferable to insert (*append[]*) the destination identifier (*obtained context information*) *to a SIP extension header*. *See, e.g.*, *Id.*, 3:8-9 ("The destination identifier are preferably inserted into an extension header of a SIP message."), 3:49-50 ("The message processor preferably inserts the destination identifier into an extension header of a SIP message."), 5:14-15 ("For example. In SIP the message may be suitably inserted

as an extension header."); Ex-1117, 35 ("Note that in some instances, header fields that have been communicated in the contact address may instead append to existing request header fields in the original redirected request. As a general rule, if the header field can accept a comma-separated list of values, then the new header field value MAY be appended to any existing values in the original redirected request."), 82, 115 ("The tag parameter is used in the To and From header fields of SIP messages. It serves as a general mechanism to identify a dialog, which is the combination of the Call-ID along with two tags, one from each participant in the dialog."). From the above discussion and disclosures, a POSITA would understand that the step of inserting the *obtained context information* (destination identifier) into a SIP TAG field *appends the obtained context information to a SIP extension header.*

122. Accordingly, Forissier discloses *"[t]he method of claim 7, wherein the step of adding the obtained context information appends the obtained context information to a SIP extension header*."

10. Claim 10

i. *[10.0] The method of claim 1, further comprising adding, to the message, an indication associated with the obtained context information where it is determined that the obtained context information is potentially inaccurate or incomplete.*

123. Forissier teaches or suggests "*[t]he method of claim 1, further comprising adding, to the message, an indication associated with the obtained*

context information where it is determined that the obtained context information is potentially inaccurate or incomplete."

124. For the reasons I explain above (*see supra* Section IX.1), Forissier discloses the method of Claim 1.

125. Forissier teaches or suggests "*adding, to the message, an indication associated with the obtained context information where it is determined that the obtained context information is potentially inaccurate or incomplete*." *See, e.g.*, Ex. 1105, 5:12-15, 5:31-48, 6:14-20, Fig. 7. As explained above for limitation 1.4 (*see supra* Section I.A.v), Forissier mentions that a tag containing the relevant *context information* (a destination identifier marking the identity of the backend server) "is preferably inserted into the message such that the tag will be included in all future messages sent in response to the message." *Id.*, 5:12-15, 6:14-20. Consequently, subsequent messages are expected to carry this context information and used in processing of messages.

126. Forissier discloses there are checks or *indication[s]* that *obtained context information is potentially inaccurate or incomplete.* For instance, Forissier discusses a potential situation which could arise with potentially inaccurate or incomplete obtained context information:

> SIP provides for the retransmission of messages in the event that a response is not received within a predetermined amount of time. One problem that this can create is that if a backend server is slow to respond, or if

> an initial message is lost, for example, the SIP user agent may retransmit the same message. If this message happens to be the first message related to a call (i.e. there is no tag present), the load-balancer is likely to send this message to a different backend server than that dealing with the first message, which may lead to the system creating several call contexts in different backend servers for a single SIP call. This may result in protocol violations for example if identical responses are sent to a user agent client, or sub-optimal processing.
>
> In order to prevent this, the load-balancer 302 preferably maintains a database 304 of all messages which relate to new calls.

Ex. 1105, 5:16-31. In order to prevent the *potentially inaccurate or incomplete obtained context information*, the load-balancer first inspects the incoming SIP message to determine if it includes the identity of a chosen backend server-if that identity has previously been chosen but is missing (*the obtained context information is…incomplete*), the method uses other context about the call (e.g., the SIP call ID field) to retrieve and add the identity of the chosen backend server to the outgoing message:

> For example, when a message is received it is determined (Step 704) whether a previously inserted tag is present. If not, this indicates that the received message may relate to a new call. The database 304 is searched to determine whether a message with the same call identification is present In the database (step 708). If yes, then the message may be, for example, a retransmitted message or a CANCEL message sent shortly after an initial INVITE message, and is forwarded to the backend server indicated in the database once a suitable tag has been inserted in the message.

Id., 5:31-48; *See also, Id.*, Fig. 7. A POSITA would have understood that the method may also check messages with complete context information against the database to avoid any *potentially inaccurate obtained context information* that could result in several contexts for a single call. In the case that the call is new, Forissier describes choosing a backend server to handle the call and inserting the context information (the identity of the chosen backend server) into the message:

> If no message having the same call ID is found, this indicates that this is the first message relating to a call, in which case a suitable backend server is chosen to process the message (step 710). A call context is subsequently created in the database (step 712), a tag is added to the received message identifying the chosen backend server (step 714), and finally the message is forwarded to the chosen backend server.

Id., 5:31-48.

127. To the extent one might assert the foregoing does not disclose "*adding, to the message, an indication associated with the obtained context information where it is determined that the obtained context information is potentially inaccurate or incomplete*," a POSITA would have understood that having a master repository that tracks context information in relation to calls, would enable the method to enforce rules that govern accuracy and completeness of *context information,* such as message destination and call identification, and indicate if there are any inconsistencies in the *obtained context information*. Particularly in the case that the database contains the same call identification as a message without a tag or a

different tag, this could indicate that the information is incomplete or inaccurate, respectively. In that case the *add[ed]…indication*, "may be…a CANCEL message sent shortly after an initial INVITE message." Ex. 1105, 5:37-39.

128. And also, that omission of the tag does not satisfy the meaning of "*inaccurate or incomplete*," it would have been common sense and a POSITA would have understood to utilize the same process (e.g., identify the appropriate backend server from other information, such as the "call ID") and insert the identity of the appropriate backend server in similar situations where the identity of the backend server is incomplete or otherwise inaccurate (e.g., the original backend server has gone offline and a failover backend server has been chosen). The '738 Patent states that this "[c]ontext data relating to each task…may be used in the event of a switchover for reinitializing the new active system with the context of any tasks that were in progress when the switchover occurred." Ex. 1101, 1:48-52. Forissier also discusses fault tolerance, "in the event of a failure of a backend server, the load-balancer can forward the message to a backup server other than that indicated by the inserted tag, without the SIP user agent ever being aware that a failure occurred." Ex. 1105, 6:36-40. As previously mentioned, it would be a natural and obvious extension of that disclosed in Forissier to handle an incomplete or inaccurate identification of a previously chosen backend server as it would for an omission of identification of a previously chosen backend server.

129. The above discussion and reasoning applies to response messages containing inaccurate backend server tags or omitting them entirely while still satisfying the limitations in claim 1. The response message is sent from BE1, processed at APP1 and then sent as an outgoing message from APP1 to CL1 (limitation 1.1); as part of the processing at APP1, it is selectively indicated to a layer of the protocol stack that APP1 should obtain context information (the accurate backend server identity) (limitation 1.2); APP1 obtains the accurate backend server identity (limitation 1.3); and the context information is added to the message (e.g., as a tag) (limitation 1.4). The only difference between the initial and response message is the response message is related to an earlier received message.

130. Accordingly, Forissier teaches or suggests "*[t]he method of claim 1, further comprising adding, to the message, an indication associated with the obtained context information where it is determined that the obtained context information is potentially inaccurate or incomplete*."

11. Claim 33

i. *[33.0] A method of sending a message from a node through a hierarchical structure of one or more discreet layers comprising:*

131. Forissier discloses "*[a] method of sending a message from a node through a hierarchical structure of one or more discreet layers*."

132. To the extent the preamble is limiting, Forissier discloses "*sending a message from a node through a hierarchical structure of one or more discrete layers*" for the same reasons described above regarding claim limitation 1.0 (*see supra* Section I.A.i). *See, e.g.*, Ex. 1105, 2:3-5, 3:4-45. On numerous occasions, Forissier describes using SIP-based communication, "[t]he method may be used in conjunction with a session initiation protocol (SIP) based network," and "[t]he load balancing system may be adapted for use in a session initiation protocol (SIP) based network." *Id.*, 3:6-7, 3:4-45. Forissier further discloses that "SIP is an application-layer control or signaling protocol for establishing, modifying, and terminating real-time calls and conferences over, primarily, Internet protocol (IP) networks." *Id.*, 2:3-5.

133. A POSITA would know that the SIP protocol, including when used over IP networks, sends messages through a hierarchical structure of one or more discrete layers. As mentioned in Section IX.2, SIP messaging, involves multiple layers, including application and transport layers. In normal convention, SIP messages are sent through a hierarchical structure of layers, particularly over IP networks, such that each layer of the protocol stack may communicate with the layer directly above and below that layer with each layer performing different functions. For instance, the application layer provides network services directly to end-user applications, the transaction layer manages all transactions between SIP entities, the transport layer is

responsible for transmission of SIP messages between clients and servers, and the transport layer is responsible for setting up, managing and terminating communication sessions. Thus, for a SIP message delivered over an IP network, the message will pass through a hierarchy of multiple layers, including the application layer and the transport layer.

134. Accordingly, Forissier discloses "*[a] method of sending a message from a node through a hierarchical structure of one or more discreet layers*."

ii. *[33.1] indicating to a layer that context information is to be obtained for that layer;*

135. Forissier discloses "*indicating to a layer that context information is to be obtained for that layer*."

136. For the same reasons described above regarding claim limitation 1.2 (*see supra* Section I.A.iii), Forissier discloses this limitation.

iii. *[33.2] obtaining, by a computing device, context information in accordance with the indication; and*

137. Forissier discloses "*obtaining, by a computing device, context information in accordance with the indication*."

138. For the same reasons described above regarding claim limitation 1.3 (*see supra* Section I.A.iv), Forissier discloses this limitation.

iv. *[33.3] adding, by the computing device, the obtained context information to the message, such that a response to the message contains the obtained context information needed to restore a pre-switchover context of the layer.*

139. Forissier discloses "*adding, by the computing device, the obtained context information to the message, such that a response to the message contains the obtained context information needed to restore a pre-switchover context of the layer*."

140. For reasons described above regarding claim limitation 1.4 (*see supra* Section I.A.v), Forissier discloses "*adding, by the computing device, the obtained context information to the message, such that a response to the message contains the obtained context information needed to restore a pre-switchover context of the layer*." *See, e.g.*, Ex. 1105, 5:6-12, 6:16-20, 6:36-40, 7:4-6.

141. Additionally, a POSITA would realize that the *obtained context information* (i.e., the identity of the backend server) is context information that can be used to restore a pre-switchover context of the layer. The chosen backend server can go offline for any number of reasons, and in that event, it is imperative to perform a switchover.

142. Forissier addresses such an occurrence, stating that "in the event of a failure of a backend server, the load-balancer can forward the message to a backup server other than that indicated by the inserted tag, without the SIP user agent ever

being aware that a failure occurred" and "is particular[ly] useful for fault tolerance purposes as the load-balancer maintains full control of the routing of messages, for example, in the event of a backend server failing." *Id.*, 6:36-40, 7:4-6.

143. A POSITA would have understood that the *obtained context information* (i.e. the identity of the chosen backend server), that persists in subsequent response messages, is used to *restore a pre-switchover context* (i.e., maintain the SIP call) in the event the chosen backend server is offline and the message must be handled by a different backend server.

144. Furthermore, a POSITA would have understood that *obtained context information* (i.e. identity of the backend server) *add[ed]* by the load-balancer application node (*computing device*) *to the message [is] information needed to restore a pre-switchover context* (i.e., maintain the SIP call) when the chosen backend server is offline. This is consistent with a POSITA's intuition and disclosures by Forissier, indicating storage of the identity of the chosen backend server as a tag in future messages, such that the messages can be routed to the appropriate backend server without further processing at the load-balancer application node. *Id.*, 5:6-12, 6:16-20; Ex-1105, 7:4-6 ("This is particular useful for fault tolerance purposes as the load-balancer maintains full control of the routing of messages, for example, in the event of a backend server failing."); Ex-1117, 101 ("When the server transaction sends a response to the transport layer to be sent, the

following procedures are followed if the transport layer indicates a failure. First, the procedures in [4] are followed, which attempt to deliver the response to a backup."); Ex-1116, 1-2.

145. In light of the above disclosures and discussion of fault tolerance, where future messages are routed to a new backend server if the chosen backend server is offline, it would be obvious to a POSITA to update the *context information* (e.g., the SIP TAG) to *restore a pre-switchover context of the layer* (e.g., the appropriate destination).

146. Accordingly, Forissier discloses "*adding, by the computing device, the obtained context information to the message, such that a response to the message contains the obtained context information needed to restore a pre-switchover context of the layer*."

X. CONCLUSION

147. 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.

January 14, 2025
Date

Kevin C. Almeroth
Kevin C. Almeroth, Ph.D.