

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SAP America, Inc.
Petitioner,

v.

Valtrus Innovations Ltd. and Key Patent Innovations Ltd.,
Patent Owner.

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

**PETITION FOR *INTER PARTES* REVIEW OF
U.S. PATENT NO. 7,936,738**

TABLE OF CONTENTS

	Page
I. INTRODUCTION	12
II. COMPLIANCE WITH IPR REQUIREMENTS.....	12
A. Grounds for Standing	12
B. Mandatory Notices	13
C. Fee Authorization	15
III. CHALLENGED CLAIMS	15
IV. THE '738 PATENT.....	15
A. Effective Filing Date	15
B. Person of Ordinary Skill in the Art	15
C. Overview of the '738 Patent.....	16
V. CLAIM CONSTRUCTION	19
VI. PRINCIPAL PRIOR ART.....	19
VII. CLAIMS 1–10 AND 33 ARE UNPATENTABLE	25
A. Ground 1: Claims 1–10 and 33 Would Have Been Obvious In View of <u>Forissier</u>	25
1. Claim 1	26
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:	26
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;	32
iii. [1.2] selectively indicating to the layer of the protocol stack that context information is to be obtained for that layer;.....	36
iv. [1.3] obtaining, by the computing device, context information in accordance with the indication; and	37

i.	[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.....	38
2.	Claim 2.....	40
i.	[2.0] The method of claim 1, further comprising adding context information obtained from a different protocol stack layer to the outgoing message.....	40
3.	Claim 3.....	44
i.	[3.0] The method of claim 1, wherein the outgoing message is sent from the node is [sic] to a remote node across a network.....	44
4.	Claim 4.....	45
i.	[4.0] The method of claim 1, used with a message-based communications system.....	45
5.	Claim 5.....	46
i.	[5.0] The method of claim 1, wherein the step of obtaining context information obtains context information related to the outgoing message.	46
6.	Claim 6.....	47
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.....	47
7.	Claim 7.....	49
i.	[7.0] The method of claim 1, for use with a session initiation protocol (SIP) network.....	49
8.	Claim 8.....	49
i.	[8.0] The method of claim 7, wherein the step of adding the obtained context information appends the obtained context information to a SIP TAG field.	49
9.	Claim 9.....	51

i.	[9.0] The method of claim 7, wherein the step of adding the obtained context information appends the obtained context information to a SIP extension header.	51
10.	Claim 10	52
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.	52
11.	Claim 33	56
i.	[33.0] A method of sending a message from a node through a hierarchical structure of one or more discreet layers comprising:	56
ii.	[33.1] indicating to a layer that context information is to be obtained for that layer;	57
iii.	[33.2] obtaining, by a computing device, context information in accordance with the indication; and	57
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.	58
VIII.	DISCRETIONARY DENIAL IS NOT APPROPRIATE	60
A.	The Board Should Not Deny Institution Under 35 U.S.C. § 325(d).....	60
B.	The Board Should Not Deny Institution Under 35 U.S.C. § 314(a).....	61
IX.	CONCLUSION.....	65

TABLE OF AUTHORITIES

	Page(s)
Cases	
<i>Apple, Inc. v. SEVEN Networks, LLC</i> , IPR2020-00156, Paper 10 (June 15, 2020).....	59, 60
<i>Benitec Biopharma Ltd. v. Cold Spring Harbor Lab’y</i> , IPR2016-00017, Paper 7 (P.T.A.B. Apr. 6, 2016)	19
<i>Dynamic Drinkware, LLC v. Nat’l Graphics, Inc.</i> , 800 F.3d 1375 (Fed. Cir. 2015)	20
<i>Ecobee Techs. v. Causam Enters., Inc.</i> , No. IPR2022-01339, Paper 20.....	57
<i>Samsung Bioepis Co. Ltd v. Regeneron Pharms., Inc.</i> , IPR2023-00739, Paper 9 (Oct. 20, 2023)	58
<i>Samsung Elecs. Co. Ltd. v. Dynamics Inc.</i> , IPR2020-00505, Paper 11 (Aug. 12, 2020).....	60
<i>Sand Revolution II, LLC v. Continental Intermodal Group-Trucking LLC</i> , IPR2019-01393, Paper 24 (June 16, 2020).....	59
<i>SAP America, Inc. v. Valtrus Innovations Limited et al.</i> , No. 1:24-cv-00756-GBW (D. Del.).....	13, 16, 17, 18
<i>Sony v. MZ Audio Scis., LLC</i> , No. IPR2022-01544, Paper 12 (Apr. 21, 2023).....	56
<i>Valtrus Innovations Ltd. v. SAP America, Inc. et al.</i> , No. 2:24-cv-00021-JRG (E.D. Tex.)	13
<i>Valtrus Innovations Ltd. v. SAP America, Inc. et al.</i> , No. 2:24-cv-00533 (E.D. Tex.).....	13
<i>Vivid Techs., Inc. v. Am. Sci. & Eng.’g. Inc.</i> , 200 F.3d 795 (Fed. Cir. 1999).	19

Statutes

35 U.S.C. § 102(e)	22
35 U.S.C. § 103	12, 15, 16
35 U.S.C. § 314(a)	57
35 U.S.C. § 325(d)	56, 57

Other Authorities

37 C.F.R. § 42.100(b)	19
37 C.F.R. § 42.104(a).....	12
U.S. Const., Art. III.....	57, 58, 59

CLAIM LISTING

Claim 1	
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:
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;
1.2	selectively indicating to the layer of the protocol stack that context information is to be obtained for that layer;
1.3	obtaining, by the computing device, context information in accordance with the indication; and
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.
Claim 2	
2.0	The method of claim 1, further comprising adding context information obtained from a different protocol stack layer to the outgoing message.
Claim 3	
3.0	The method of claim 1, wherein the outgoing message is sent from the node is [sic] to a remote node across a network.
Claim 4	
4.0	The method of claim 1, used with a message-based communications system.

Claim 5	
5.0	The method of claim 1, wherein the step of obtaining context information obtains context information related to the outgoing message.
Claim 6	
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.
Claim 7	
7.0	The method of claim 1, for use with a session initiation protocol (SIP) network.
Claim 8	
8.0	The method of claim 7, wherein the step of adding the obtained context information appends the obtained context information to a SIP TAG field.
Claim 9	
9.0	The method of claim 7, wherein the step of adding the obtained context information appends the obtained context information to a SIP extension header.
Claim 10	
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.
Claim 33	
33.0	A method of sending a message from a node through a hierarchical structure of one or more discreet [sic] layers comprising:

33.1	indicating to a layer that context information is to be obtained for that layer;
33.2	obtaining, by a computing device, context information in accordance with the indication; and
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.

EXHIBIT LIST

No.	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 <i>et al.</i>
1106	J. Forissier, et al., <i>Method and Apparatus for Load-Balancing in a Distributed Processing System</i> , 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 <i>Valtrus Innovations Ltd. v. SAP America, Inc. et al</i> , 2:24-cv-00021-JRG (E.D. Tex.)
1110	Motion to Dismiss for Lack of Standing (Public Version) in <i>Valtrus Innovations Ltd. v. SAP America, Inc. et al</i> , 2:24-cv-00021-JRG (E.D. Tex.)
1111	Motion to Reconsider Ruling on Motion to Dismiss for Lack of Standing (Public Version) in <i>Valtrus Innovations Ltd. v. SAP America, Inc. et al</i> , 2:24-cv-00021-JRG (E.D. Tex.)
1112	Original Complaint for Patent Infringement in <i>Valtrus Innovations Ltd. v. SAP America, Inc. et al</i> , 2:24-cv-00533-JRG (E.D. Tex.)
1113	Plaintiffs' Notice Regarding <i>Valtrus II</i> in <i>Valtrus Innovations Ltd. v. SAP America, Inc. et al</i> , 2:24-cv-00021-JRG (E.D. Tex.)
1114	SAP's Invalidity Contentions (Cover Pleading) in <i>Valtrus Innovations Ltd. v. SAP America, Inc. et al</i> , 2:24-cv-00021-JRG (E.D. Tex.)
1115	Valtrus's Infringement Contentions (Cover Pleading) in <i>Valtrus Innovations Ltd. v. SAP America, Inc. et al</i> , 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. INTRODUCTION

Petitioner SAP America, Inc. and SAP, SE submits this Petition for *Inter Partes* Review of claims 1-6, 10, and 33 (the “Challenged Claims”) of U.S. Patent No. 7,936,738 (the “’738 Patent” (Ex-1101)), assigned to Patent Owner, Valtrus Innovations Ltd. and Key Patent Innovations Ltd. Petitioner respectfully submits that the Challenged Claims of the ’738 Patent are unpatentable under 35 U.S.C. § 103 in view of the prior art references discussed herein. Accordingly, it is respectfully requested that the Board institute an *inter partes* review of the ’738 Patent.

II. COMPLIANCE WITH IPR REQUIREMENTS

A. Grounds for Standing

As required by 37 C.F.R. § 42.104(a), Petitioners certify that the ’738 Patent “is available for inter partes review and that the Petitioners are not barred or estopped from requesting an inter partes review challenging the patent claims on the grounds identified in the petition.” Neither Petitioners nor any real party in interest has filed a civil action challenging the validity of a claim of the ’738 Patent. This petition is being filed no more than one year after the date on which Petitioners or real parties in interest or privies of Petitioners were served with a complaint alleging infringement of the ’738 Patent.

B. Mandatory Notices

Real Party-in-Interest: Petitioners identify SAP America, Inc. and SAP SE (collectively, “SAP”) as real parties-in-interest.

Related Matters: No previous IPR or post-grant proceeding has been filed challenging the ’738 Patent. Valtrus has asserted the ’738 Patent in litigation against SAP in *Valtrus Innovations Ltd. v. SAP America, Inc. et al.*, 2:24-cv-00021-JRG (E.D. Tex.); Patent Owners have asserted the ’738 Patent in litigation against SAP in *Valtrus Innovations Ltd. v. SAP America, Inc. et al.*, 2:24-cv-00533 (E.D. Tex.); and SAP filed a declaratory judgment action of non-infringement of the ’738 Patent against Patent Owners and Patent Platform Services, LLC in *SAP America, Inc. v. Valtrus Innovations Limited et al.*, 1:24-cv-00756-GBW (D. Del.).

Lead and Back-Up Counsel and Service Information

Lead Counsel	Back-up Counsel
Louis L. Campbell (Reg. # 59,963) Winston & Strawn LLP 255 Shoreline Drive, Suite 520 Redwood City, CA 94065 Tel: (650) 858-6500 LLcampbell@winston.com	Scott M. Border (Reg. # 77,744) Winston & Strawn LLP 1901 L Street NW Washington, DC 20036 Tel: (202) 282-5000 SBorder@winston.com
	Samuel W. Riebe (Reg. # 78,044) Winston & Strawn LLP 2121 N. Pearl Street, Suite 900 Dallas, TX 75201 Tel: (214) 453-6500 SRiebe@winston.com
	Michael Bittner (<i>Pro hac vice</i> to be submitted) Winston & Strawn LLP

	2121 N. Pearl Street, Suite 900 Dallas, TX 75201 Tel: (214) 453-6500 MBittner@winston.com
	Matt R. McCullough (<i>Pro hac vice</i> to be submitted) Winston & Strawn LLP <i>Pro hac vice</i> to be submitted Winston & Strawn LLP 255 Shoreline Drive, Suite 520 Redwood City, CA 94065 Tel: (650) 858-6500 MRMcCullough@winston.com
	William Logan (<i>Pro hac vice</i> to be submitted) <i>Pro hac vice</i> to be submitted Winston & Strawn LLP 800 Capitol Street, Suite 2400 Houston, TX 77002 Tel: (713) 651-2600 WLogan@winston.com
	Patrick Clark (<i>Pro hac vice</i> to be submitted) Winston & Strawn LLP <i>Pro hac vice</i> to be submitted 2121 N. Pearl Street, Suite 900 Dallas, TX 75201 Tel: (214) 453-6500 PClark@winston.com

Service Information: Service information is provided in the designation of counsel above. Petitioners consent to service of all documents via electronic mail at IPR-SAP-Val@winston.com.

C. Fee Authorization

The Office is authorized to charge all fees due in connection with this proceeding to Deposit Account No. 50-1814.

III. CHALLENGED CLAIMS

Ground 1: Claims 1-10 and 33 are unpatentable as obvious under 35 U.S.C. § 103 based on Forissier (Ex-1105).

IV. THE '738 PATENT**A. Effective Filing Date**

The '738 patent ultimately claims priority to a foreign application, European Patent Application 03291852, filed July 25, 2003. Thus, Petitioner assumes for the purposes of this Petition that **July 25, 2003**, is the effective filing date.

B. Person of Ordinary Skill in the Art

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. Ex-1103, ¶ 33. A POSITA would have had a bachelor's degree in computer science, computer engineering, or an equivalent, and two years of relevant experience in networking and network security, and related fields. Lack of professional experience can be remedied by additional education, and vice versa. *Id.*, 32-34.

C. Overview of the '738 Patent

The '738 Patent describes “a method and system for providing fault tolerance in . . . message-based communication systems.” Ex-1101, 1:2-6. To purportedly achieve this fault tolerance, the '738 Patent discloses inserting “context information” into outgoing messages, such that a response to the outgoing message will contain the context information. *Id.*, Abstract, 2:1-10, 3:29-36, 5:59-67. Some embodiments further involve “restoring the context information of a layer of a protocol stack of a node.” *Id.*, 2:28-30.

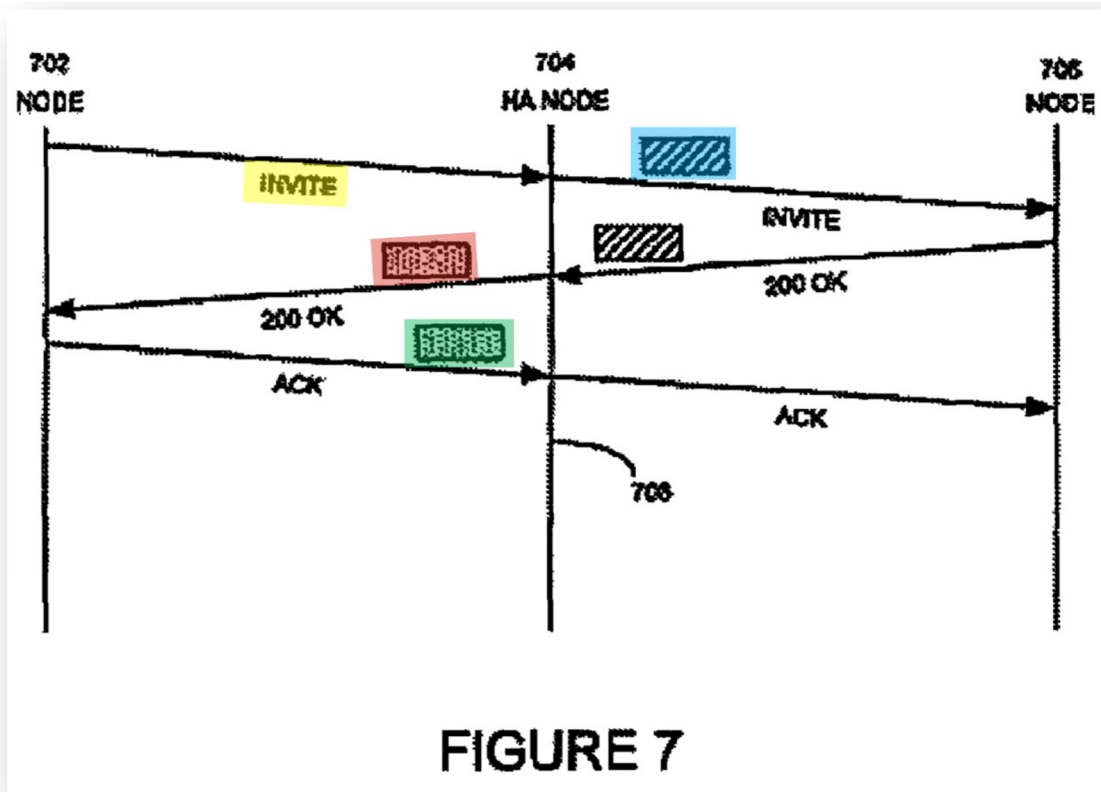
According to the '738 Patent, inserting this context information into the outgoing message is an improvement over the prior art, which instead often used a “common storage element” to store context data. *See id.*, 1:46-52. In particular, the '738 Patent contends that the prior-art practice of using a common storage area was ill suited for telephony. *Id.*, 1:53-62. More specifically, the '738 Patent posits that for telephony, “the requirement for storing context data adds to the complexity and cost of such systems.” *Id.*, 1:63-65, 4:34-36, 5:3-7.

The '738 Patent explains the purported invention within the context of a specific telephony protocol, session initiation protocol (SIP). *Id.*, 2:21-24. For SIP, instead of storing context information on a common storage area, “context information may be appended to a SIP TAG field, or to a SIP extension header.” *See id.*, 2:20-24, 2:46-50, 3:3-4 (“Suitably the obtained context information is

appended to a SIP TAG field or to a SIP extension header.”). Likewise, the ’738 Patent explains that for SIP, restoration involves “restoring the context using context information stored in a SIP TAG or a SIP extension header.” *Id.*, 3:25-28.

The ’738 Patent does not ascribe any specific form to the context information and instead generally explains that the context may “differ depending on particular circumstances,” and in telephony applications “it may be desirable to only store context information relating to [the current call] with the outgoing message.” *Id.*, 5:43-48. However, within the context of SIP, the ’738 Patent provides that “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.” *Id.*, 7:1-3. For SIP messages, that destination address may be stored in the SIP TAG, which the specification explains “is included in all related messages and response messages.” *Id.*, 7:4-10 (“The SIP specifications define that the TAG must not be altered once set.”). Alternatively, “a SIP extension header may be used to store the context field.” *Id.*, 6:54-57.

The ’738 patent explains this more specifically regarding an exemplary SIP message-flow diagram shown in Figure 7 (below), which Dr. Kevin Almeroth has annotated in his attached declaration in support of this Petition:



Ex-1103, ¶ 46-47. In this diagram, “high-availability node 704” is exchanging messages between a “calling party node 702” and a “called party node 706.” Ex-1103, ¶ 47 (citing Ex-1101, 7:16-19). In this arrangement, the calling party node 702 sends an INVITE message (yellow) to the high-availability node 704 to establish a connection, and the high availability node 704 forwards the invite to the called party node 706—but only after inserting the context information (blue) into the TAG field of the INVITE message. *Id.* (citing Ex-1101, 7:19-27). Once inserted, the context information in the TAG field is included both in the 200-OK response (red) returned to calling party node 702 and in the acknowledgement response (green) from calling party 702 to the high-availability node 704. *See id.*

(citing Ex-1101, 7:6-10) (“The TAG feature is a field in a SIP message which, if used, must be set in the first outgoing message from a node, and thereafter, the TAG is included in all related messages and response messages. The SIP specifications define that the TAG must not be altered once set.”).

V. CLAIM CONSTRUCTION

Claims are given their “ordinary and customary meaning” as understood by a POSITA and the prosecution history pertaining to the patent. 37 C.F.R. § 42.100(b). Because a POSITA would find the challenged claims unpatentable under any interpretation consistent with their plain and ordinary meaning in the context of the ’738 Patent, the Board need not expressly construe the claim terms. *See Vivid Techs., Inc. v. Am. Sci. & Eng. ’g. Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999).

VI. PRINCIPAL PRIOR ART

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 (“EPA 03291191”) (published as EP1480407A1), which was filed on May 20, 2003. Ex-1106. EPA 03291191 was filed in the European Patent Office in English. *See id.*

Forissier is entitled to the filing date of EPA 03291191 because, as the below claim chart demonstrates, EPA 03291191 (Ex-1106) fully supports at least one

claim of Forissier. See *Benitec Biopharma Ltd. v. Cold Spring Harbor Laboratory*, IPR2016-00017, Paper 7 at 7 (P.T.A.B. Apr. 6, 2016) (citing *Dynamic Drinkware, LLC v. Nat’l Graphics, Inc.*, 800 F.3d 1375, 1381 (Fed. Cir. 2015)) (To show a challenged patent is entitled to the earlier filing date of its priority provisional application, “Petitioner must demonstrate that the provisional application provides written description support for at least one claim of the [challenged] patent.”).

Forissier Claim 1	EPA 03291191 (Ex-1106) Disclosures
1. A method of routing a message to one of a plurality of available processing systems, comprising the steps of:	In other situations, context information must be maintained by the load-balancer in order to determine to which backend server a message should be forwarded to . (Ex-1106, 1:30-31) According to a first aspect 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 . (Ex-1106, 3:13-17)
detecting for a presence of a destination identifier in the message; and	According to a first aspect 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. (Ex-1106, 3:13-17)
when the presence of the destination identifier is detected, forwarding the message to a processing system	According to a first aspect 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 . (Ex-1106, 3:13-17)

Forissier Claim 1	EPA 03291191 (Ex-1106) Disclosures
identified thereby; and	
when the presence of a destination identifier is not detected:	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. (Ex-1106, 3:17-21)
determining a destination processing system for processing the message;	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. (Ex-1106, 3:17-21)
inserting into the message a destination identifier identifying the determined destination processing system without storing the destination identifier outside the message; and	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. (Ex-1106, 3:17-25)
forwarding the message to the determined processing system.	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. (Ex-1106, 3:17-21)

Thus, Forissier is entitled to the filing date of EPA 03291191 and is, thus, prior art to the '738 Patent at least under pre-AIA 35 U.S.C. § 102(e). In the analysis below, parallel cites to Forissier and Forissier's European parent application are provided. This demonstrates that all portions of Forissier that Petitioner relies upon were present in Forissier's European parent application.

Forissier describes methods and systems related to routing messages to a plurality of available processing systems. Ex-1105, Abstract; *see also* Ex-1106, Abstract. More specifically, it describes inserting and detecting a “destination identifier” in the messages to facilitate forwarding messages to the appropriate destination processing system. Ex-1105, Abstract, 2:42-48; Ex-1106, Abstract, 16:8-11. In some embodiments (e.g., where the destination identifier is determined to be missing), Forissier describes also using a message identifier inserted into the message to restore context regarding the appropriate destination for the message. *E.g.*, Ex-1105, 2:56-61, 3:18-34; *see also* Ex-1106, 3:27-33, 4:29-31.

Forissier is analogous to the '738 Patent for at least three reasons.

First, it is directed to the same field of endeavor as the '738 Patent—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; *see also* Ex-1106, 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.”); Ex-1103, ¶ 57.

Second, Forissier directly addresses a problem described by the ’738 Patent: overcoming the complexity and cost of “maintain[ing] a shared or common storage element” in a fault-tolerant message system, particularly for telephony applications. Ex-1101, 4:34-36; Ex-1103, ¶ 58. Like the ’738 Patent, Forissier explains that its approach is advantageous over the prior art because it alleviates the need for using traditional storage to retain context information about calls, which Forissier (similarly to the ’738 Patent) contends is inefficient for applications such as SIP:

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.

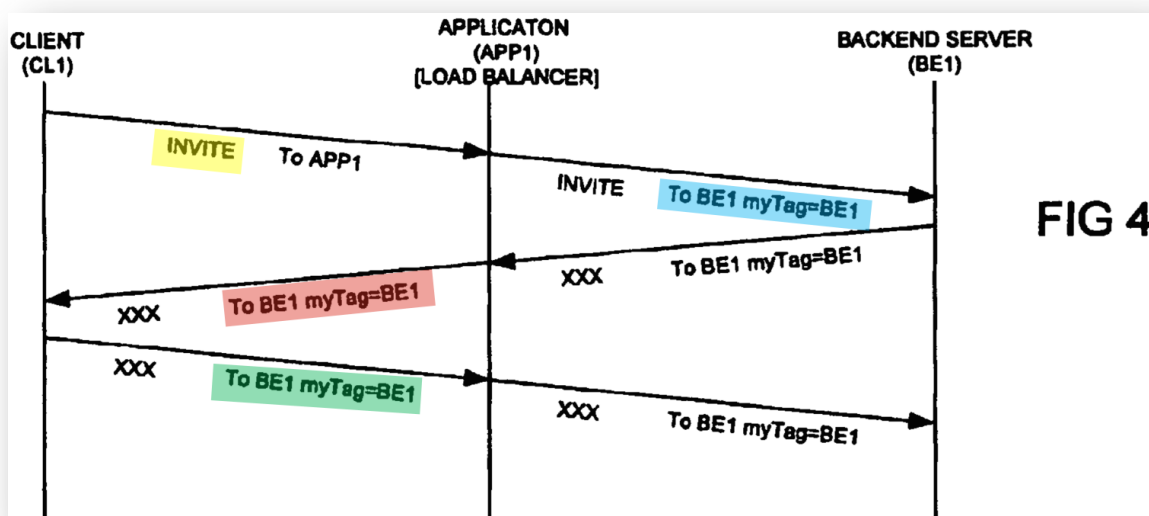
Firstly, since the length of a SIP call is theoretically unbounded and the fact that the load-balancer must store call context information for the duration of each and every the call, it is possible that the load-balancer may become overwhelmed, especially if the number of simultaneous calls and the 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; *see also* Ex-1106, 2:31-3:8. Forissier 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; *see also* Ex-1106, 11:6-8.

Third, Forissier explains its teachings as relating to the same technological application as the '738 Patent: SIP. Ex-1103, ¶ 59. More specifically, like the '738 Patent and its discussions regarding Figure 7, Forissier describes an embodiment of its purported invention within the context of an exemplary SIP message-flow diagram in Figure 4 (annotated below):



Ex-1103, ¶ 59. In this diagram, a SIP user agent (CL1) sends a SIP Invite message (yellow) to a load-balancer application (APP1). *Id.*; Ex-1105, 6:11-12; *see also* Ex-1106, 10:17-18. The load-balancer application on APP1 inserts a tag (“myTag”, blue) into the message that identifies a chosen backend server (BE1) and forwards the message to BE1. Ex-1103, ¶ 59; Ex-1105, 6:14-20; *see also* Ex-1106, 9:28-32. The myTag information identifying BE1 as the destination server (red) is then included in the response to CL1. Ex-1103, ¶ 59; Ex-1105, 6:20-23; *see also* Ex-1106, 9:33-10:1. 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-1103, ¶ 59; Ex-1105, 6:23-25; *see also* Ex-1106, 10:1-2. The load-balancer application APP1 will then examine the response to determine if there is a tag present in it and, based on the information in myTag, will forward the message to BE1. Ex-1103, ¶ 59; Ex-1105, 6:25-33; *see also* Ex-1106, 10:2-8.

VII. CLAIMS 1-10 AND 33 ARE UNPATENTABLE

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

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:***

To the extent the preamble is limiting, the 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*” in the ’738 Patent was earlier disclosed as a part of the method for routing messages using a load balancing application in Forissier. Ex-1103, ¶¶ 61, 71.

Forissier discloses “*storing context information in an outgoing message sent from a node.*” See, e.g., Ex-1105, 1:51-56, 2:14-32; see also Ex-1106, 2:5-9, 2:31-3:8. Like the ’738 Patent, Forissier is concerned with eliminating the storage of context information for calls in a database and instead storing that context information in an outgoing message. It explains that prior-art load balancers were generally “required to store some context information, for example the call ID of each message.” Ex-1105, 1:51-57; see also Ex-1106, 2:5-9. Those prior-art approaches (which included “maintaining a database of context information” about active calls) would generally “not be suitably adapted” for then-emerging telephony protocols (e.g., SIP). Ex-1105, 2:14-32; see also Ex-1106, 2:31-3:8.

Among the problems Forissier identifies with prior-art approaches was the complexity in “maintaining a database of context information for all active calls,”

which given the potential length of SIP calls would potentially “overwhelm” the load balancer. *See* Ex-1105, 2:14-32; *see also* Ex-1106, 2:31-3:8. Likewise, Forissier explains that managing the stored context information for SIP calls would likely “impact the performance capabilities” of the load balancer and “limit [its] message handling capabilities.” *See* Ex-1105, 2:14-32; *see also* Ex-1106, 2:31-3:8.

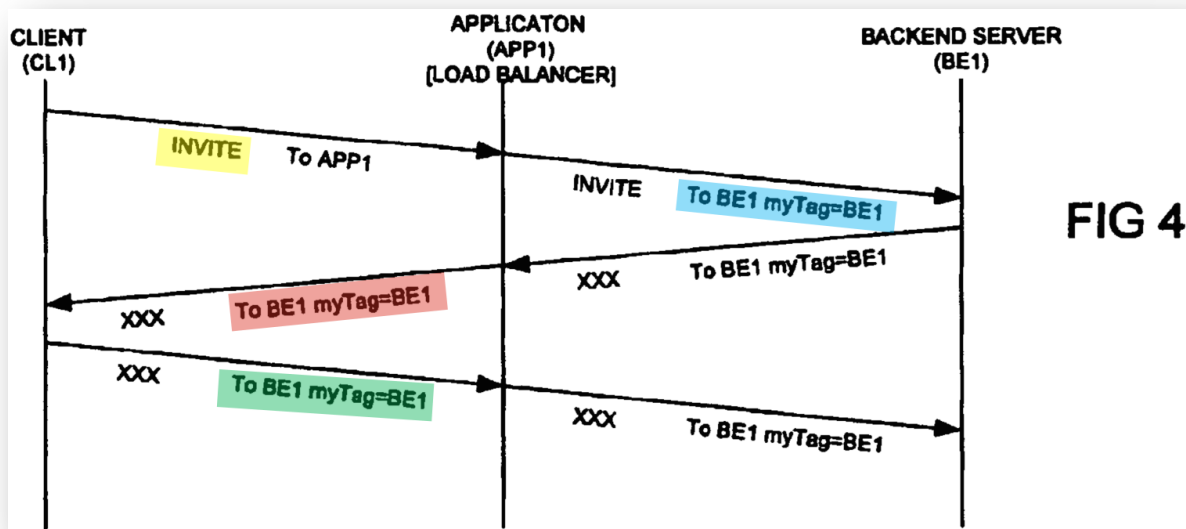
Forissier’s teachings overcome “at least some of the above-mentioned problems” with storing context information. Ex-1105, 2:33-35; *see also* Ex-1106, 3:10-11. More specifically, Forissier teaches inserting a “destination identifier” “*in an outgoing message*” instead of “*storing context information*” 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.

See Ex-1105, 2:42-52, Abstract, 5:6-15, 7:8-15; *see also* Ex-1106, Abstract, 7:33-8:6, 11:11-16; Ex-1103, ¶ 65.

Forissier further explains these teachings using an exemplary SIP message-flow diagram in Figure 4 (annotated below):



Ex-1103, ¶ 66. In this diagram, a SIP user agent (CL1) sends a SIP Invite message (yellow) to a load-balancer application (APP1). Ex-1105, 6:11-12; *see also* Ex-1106, 9:26-27. The load-balancer application is a node, and it comprises an application running on one or more computing devices, such as network servers. Ex-1105, 6:12-14 (“As described above, the application is an application using a load-balancer and a number of distributed backend servers.”), 4:37-41 (“It will be appreciated that although the load-balancer and back-end servers are shown as being integral to the B2BUA 112, these elements may be located externally therefrom, for example in a distributed manner.”); *see also* Ex-1106, 9:27-28, 7:4-6; Ex-1101, 4:63-5:2 (describing “SIP user agent 300” and “SIP back-to-back user agent (B2BUA)” in the ’738 patent as “network nodes”); Ex-1103, ¶ 66.

The load-balancer application on APP1 inserts a tag (“myTag”, blue) into the message that identifies a chosen backend server (BE1) and forwards the message to BE1. Ex-1105, 6:14-20; *see also* Ex-1106, 9:28-32. The myTag information identifying BE1 as the destination server (red) is then included in the response to CL1. Ex-1105, 6:20-23; *see also* Ex-1106, 9:33-10:1. If the SIP 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, 5:12-15 (“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, [i]n SIP the message may be suitably inserted as an extension header.”); *see also* Ex-1106, 10:1-2, 8:3-6. The load-balancer application APP1 will then examine the response to determine if there is a tag present in it and, based on the information in myTag, will forward the message to BE1. Ex-1105, 6:25-33; *see also* Ex-1106, 10:2-8.

The “destination identifier” in Forrister is thus application-specific data from the load-balance application that enables SIP messages to reach a correct destination (e.g., BE1), which is the “*context information*” recited in the claim. Ex-1103, ¶ 67. The ’738 Patent confirms this, broadly explaining that for telephony networks, “context information” may include “any application specific data related to individual calls.” Ex-1101, 1:53-56. As one example, the ’738 Patent confirms that “context data may include a destination address of a SIP call, which would

enable SIP messages related to that call to be routed to the correct destination.” *Id.*, 7:1-3.

The destination identifier in Forissier is an example of a “destination address” discussed in the ’738 patent: it identifies a “determined destination processing system” so messages containing the destination identifier can be forwarded to that determined destination processing system. *See* Ex-1105, Abstract, 2:36-42 (“According to a first aspect 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.”); *see also* Ex-1106, 3:13-17; Ex-1103, ¶ 67.

Thus, because Forissier teaches a load-balancer application (“*a method*”) inserting a destination identifier into an outgoing message (“*storing context information in an outgoing message sent from a node*”) and because Forissier teaches that applications running on nodes like network servers (“*node including a computing device*”), Forissier teaches or suggests “*a method for storing context information in an outgoing message sent from a node including a computing device.*” Ex-1103, ¶ 69.

Forissier also discloses using a protocol stack having at least one layer. The SIP protocol that Forissier teaches “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-1105, 2:3-5; *see also* Ex-1106, 2:22-24; Ex-1103, ¶ 70. As described in the ’738 Patent, a POSITA would have known that using Forissier’s load-balancer application to perform SIP routing as described above, 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; Ex-1103, ¶ 70.

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; *see also id.*, 3.575 (defining “open systems interconnection (OSI) model” as [a] computer network architecture model proposed as a standard model by the International

Organization for Standardization . . . consist[ing] of seven layers”), 3.33 (defining “application layer” as “[t]he seventh and highest layer of the seven-layer OSI model providing the only interface between the user and the application program.”). Dr. Almeroth explains that 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*,” and that a POSITA would have recognized the “application-layer” that SIP operates at as one of the layers in the OSI model. Ex-1103, ¶ 70 (citing Ex-1107).

Forissier thus discloses this limitation.

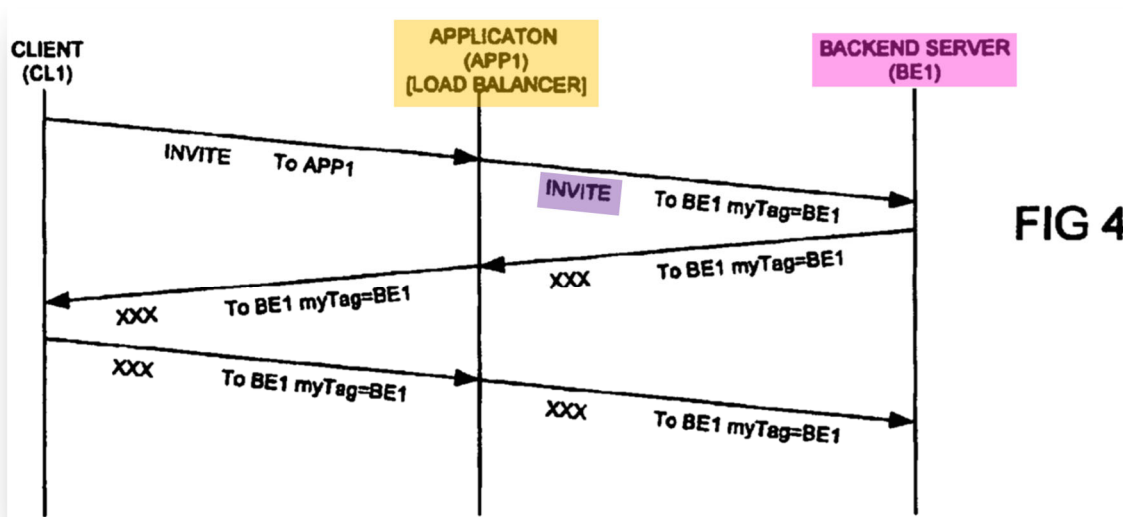
- ii. ***[1.1] providing, by the computing device, the outgoing message from an application to a layer of the protocol stack, the outgoing message is destined for an application on a destination node;***

Forissier teaches “*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, 2:3-5, 6:11-14; see also Ex-1106, 2:22-24, 9:26-28.

The exemplary explanations in Forissier describe the “nodes” (e.g., CL1, APP1, and BE1 from Figure 4) as including telephony applications, particularly SIP applications. Ex-1105, 6:11-14; see also Ex-1106, 9:26-28. SIP is “an application-layer control or signaling protocol” that is communicating over computer networks, and more particularly, “Internet protocol (IP) networks.” Ex-

1105, 2:3-5; *see also* Ex-1106, 2:22-24; Ex-1103, ¶ 73. Forissier explains that the SIP messages are directed to the applications running on the respective nodes (e.g., CL1, APP1, and BE1), including Forissier's load-balancing application (APP1). *See, e.g.*, Ex-1105, 6:11-14 (“As shown in FIG. 4, a SIP user agent (CL1) sends a SIP INVITE message to an application (APP1); *see also* Ex-1106, 9:26-28. As described above, the application is an application using a load-balancer and a number of distributed backend servers.”), Ex-1105, Fig. 4; *see also* Ex-1106, Fig. 4.

In Forissier's Figure 4 (annotated below), the load-balancer application (“an application”) on APP1 (“the computing device”) (orange) sends an outgoing message (a SIP Invite) (purple) (“providing, by the computing device, the outgoing message”) destined for an application on BE1 (“the outgoing message is destined for an application on a destination node”) (pink):



Ex-1103, ¶ 74; Ex-1105, 6:11-14; *see also* Ex-1106, 9:26-28. The application on BE1 may, for instance, facilitate SIP calls and/or provide for value-added services, such as for pre-paid billing, authentication, or credit-card authorizations. *See, e.g.*, Ex-1105, 4:10-41; *see also* Ex-1106, 6:3-7:6.

Dr. Almeroth explains that a POSITA would understand that the message (INVITE) would originate from an application on APP1 and would be destined for an application on BE1. *See* Ex-1103, ¶ 75; Ex-1105, 4:10-41 (describing various telephony applications that may be receiving messages on BE1), 6:11-14 (describing the previous INVITE message from CL1 as being directed to an application on APP1); *see also* Ex-1106, 6:3-7:6, 9:26-28.

Dr. Almeroth further explains that a POSITA would have recognized that to send a message such as the SIP Invite to BE1, the application on APP1 would provide the outgoing message to a layer of the protocol stack (*“providing, by the computing device, the outgoing message from an application to a layer of the protocol stack”*). Ex-1103, ¶ 75 (citing 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.”)). To the extent that one asserts that “*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*” is not explicitly disclosed in these teachings, it would have been obvious in light of Forissier’s disclosures, and been a matter of common sense, because (as described above) this is how messages, including SIP messages over IP networks, are ordinarily sent by application across a network by providing those messages to a layer of the protocol stack. Ex-1103, ¶ 75; Ex-1116, 1-2.

A POSITA would have understood that to facilitate sending the SIP Invite, the computing device of APP1 provides the SIP Invite message destined for the application on BE1 from the load-balancer application on APP1 to the SIP application layer. *Id.* 76. A POSITA would understand that the message passes through the other layers of the protocol stack (e.g., transport and network layers) to transport the message over the network. Ex-1103, ¶¶ 75-76; Ex-1116, 1-2.

Forissier thus teaches or suggests this limitation.

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

Forissier teaches “*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; see also Ex-1106, 8:19-33, 9:28-32.

When a message arrives at the APP1 node, Forissier teaches indicating to the APP1 node whether (“*selectively indicating*”) to obtain the identity of the chosen backend server (“*that context information [the identity of the chosen backend server] is to be obtained*”) by either (“*selectively indicating*”) including a tag (in which case, no context information needs to be obtained) or not including a tag (which indicates that the node must obtain and insert the context information). See, e.g., Ex-1105, 5:29-48, 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, [i]dentifying the chosen backend server into the message, and forwards the message to the chosen backend server.”); see also Ex-1106, 8:19-33, 9:28-32; Ex-1103, ¶ 79. If the message is new, APP1 indicates that the load-balancer application should obtain the identity for an appropriate backend server. See, e.g., Ex-1105, 5:29-48, 6:14-20; see also Ex-1106, 8:19-33, 9:28-32; Ex-1103, ¶ 79.

A POSITA would have understood, or found it obvious, including based on common sense, that this indication (including or omitting the tag) would be directed “*to the layer of the protocol stack*” and would be “*for that layer*” (i.e., the layer to which the indication is being directed is the layer that requires the indication), as this is ordinarily and customarily how information would pass between the load-balancing application on APP1 and the network, including for SIP messages. Ex-1103, ¶ 80 (citing Ex-1116, 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”))).

Forissier thus teaches or suggests this limitation.

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

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 see also Ex-1106, 7:27-8:3, 9:30-32, Fig. 8.

When a new SIP Invite arrives at the load-balancer application, Forissier teaches that the load-balancer application should determine a suitable backend

server for handling the call. *See* Ex-1105, 4:65-5:9; *see also* Ex-1106, 7:27-8:3; Ex-1103, ¶ 83. Particularly, the load-balancer application should obtain “sufficient information to enable the load-balancer to route the message without requiring a further call context.” (i.e., *obtaining, by the computing device, context information in accordance with the indication*). *See* Ex-1105, 5:6-12, 6:16-20 (“If no tag is found, the load-balancer chooses a suitable backend server (BE1) and inserts a tag, for example myTag, [i]dentifying the chosen backend server into the message, and forwards the message to the chosen backend server.”), Fig. 8; *see also* Ex-1106, 7:33-8:3, 9:30-32, Fig. 8; Ex-1103, ¶ 83-84.

Forissier thus teaches or suggests this limitation.

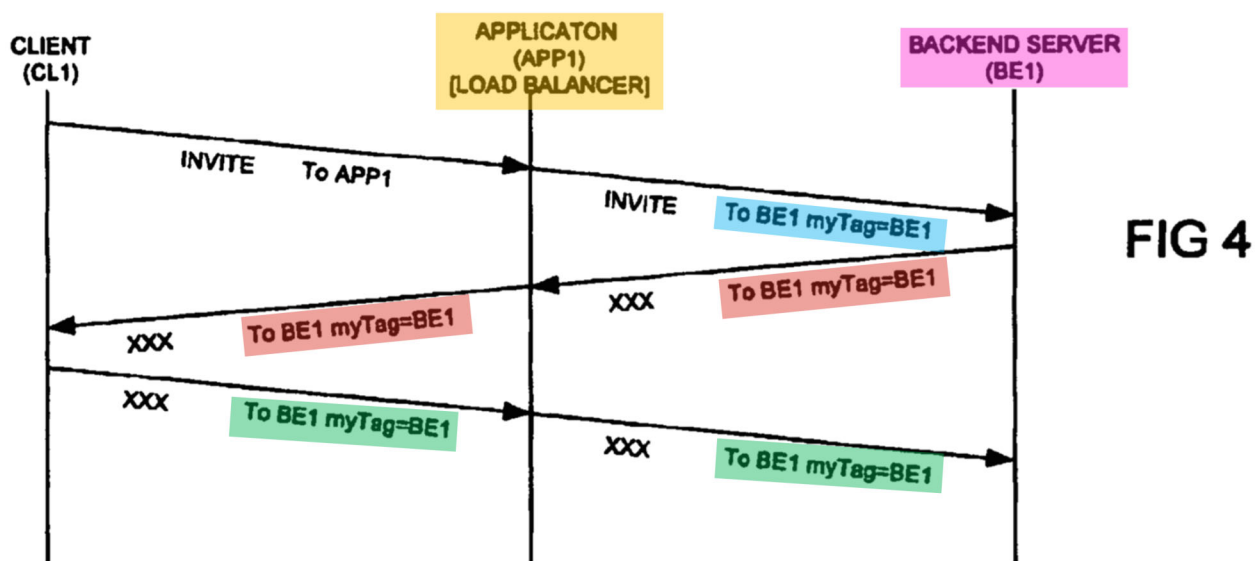
- i. ***[1.4] adding, by the computing device, the obtained context information to the outgoing message such that a response, received from the destination node, to the outgoing message contains the obtained context information.***

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; *see also* Ex-1106, 7:33-8:6, 9:26-10:8, Fig. 4.

Specifically, Forissier teaches inserting a tag into the SIP Invite that indicates the previously obtained context information about the chosen backend

server (adding, by the computing device, the obtained context information to the outgoing message). Ex-1105, 5:12-15, 6:14-20; see also Ex-1106, 8:3-6, 9:28-32.

This tag should contain “sufficient information to enable the load-balancer to route the message without requiring a further call context” and should be “inserted into the message such that the tag will be included in all future messages sent in response to the message” (*such that a response, received from the destination node, to the outgoing message contains the obtained context information*). Ex-1105, 5:12-15, 6:14-20; see also Ex-1106, 8:3-6, 9:28-32. This is shown in Figure 4 (annotated below), where APP1 (orange) adds the contextual information about the identity of the backend server (BE1) (pink) to the outgoing SIP Invite (purple) as a tag in the SIP Header (“myTag”) (blue)—such that future responses (e.g., those highlighted red and green) contain the context information about BE1 that is stored in the “myTag” header field:



Ex-1103, ¶ 87; Ex-1105, 5:6-15 (“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) ; *see also* Ex-1106, 7:33-8:6. 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.”), 6:11-33; *see also* Ex-1106, 9:26-10:8; Ex-1103, ¶ 87.

Forissier thus teaches or suggests this limitation and therefore renders obvious claim 1.

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.***

Forissier teaches “*adding context information obtained from a different protocol stack layer to the outgoing message.*” Ex-1103, ¶ 91; Ex-1105, 2:3-5, 4:6-9, 5:1-15, Fig. 8; *see also* Ex-1106, 2:22-24, 6:12-14, 7:29-8:6, Fig. 8.

Dr. Almeroth explains that SIP messages are sent over protocol stack that comprises, for instance, an application layer and a transport layer. *See* Ex-1103, ¶ 91; Ex-1005, 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). In Forissier, the load-balancer application at APP1 obtains the relevant context information (the identity of the chosen backend server, BE1) at the application layer, then passes that context information to the transport layer, where it is added as the destination for the message:

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.

See Ex 1105, 5:1-15, 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.”), 6:11-14 (“As shown in FIG. 4, a SIP user agent (CL1) sends a SIP INVITE message to an application (APP1); see also Ex-1106, 7:29-8:6, 2:22-24, 9:26-9:28. As described above, the application 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.”); Ex-1103, ¶ 91.

This is further illustrated in Figure 8 from Forissier (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. Ex-1103, ¶ 92; Ex-1105, 4:6-9, Fig. 8; see also Ex-1106, 6:12-14, Fig. 8. At step 710 (yellow), “the load -balancer chooses ... one of the available backend servers.” Ex-1105, 5:1-6; see also Ex-1106, 7:29-33. Then at step 714, “[t]he load-

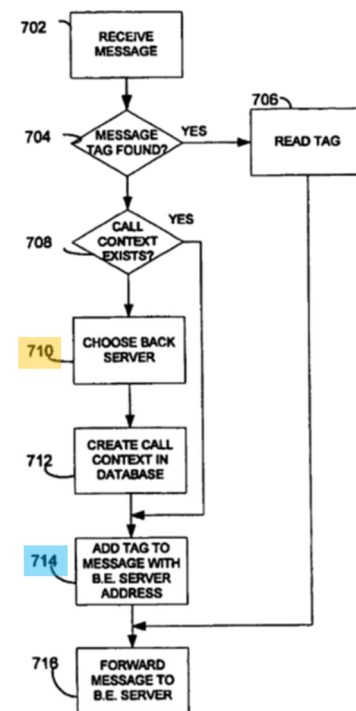
U.S. Patent

May 3, 2011

Sheet 5 of 5

US 7,936,763 B2

FIGURE 8



balancer 302 inserts a tag in the SIP message” (such as in the SIP extension header) that identifies the chosen backend server (“*context information obtained from a different protocol stack layer*”). Ex-1105, 5:6-15; *see also* Ex-1106, 7:33-8:6.

A POSITA would understand that when the message is then transmitted to the chosen backend server, the backend server identity that was obtained by a load-balancing application at the application layer is added to the outgoing message as the destination at the transport layer to facilitate sending the message. Ex-1103, ¶ 92; Ex-1116, 1-2 (describing operation of layers in sending messages).

To the extent one may assert that Forissier does not expressly disclose “*adding context information obtained from a different protocol stack layer to the outgoing message,*” as Dr. Almeroth explains, a POSITA would have found it obvious, including based on common sense, that a load-balancing application at the application layer would obtain the backend server identity, whereas the identity of the backend server would be added to the outgoing message as the destination address at the transport layer to facilitate message delivery. Ex-1103, ¶ 93; Ex-1116, 1-2. This is so because a POSITA would have understood that information obtained at the application layer that needs to be transmitted across the network is ordinarily and customarily, and as a matter of common sense, added to the outgoing message at a different layer of the protocol stack (e.g., the transport layer). Ex-1103, ¶ 93; Ex-1116, 1-2.

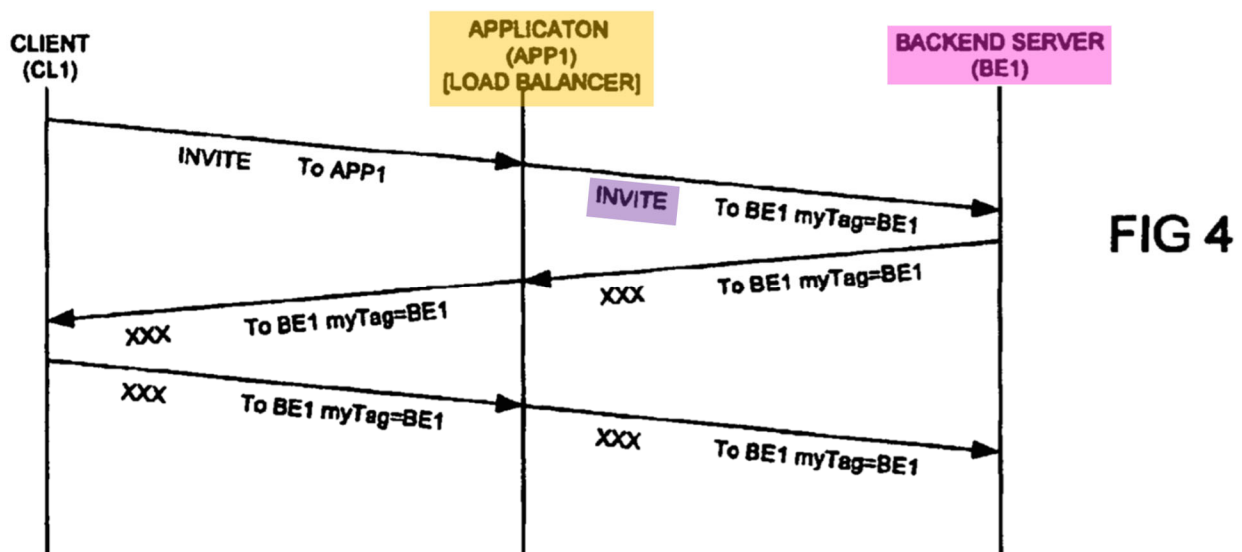
Forissier thus renders this claim obvious.

3. Claim 3

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

Forissier discloses that the “outgoing message is sent from the node is to a remote node across a network.” Ex-1105, 4:37-41, 6:14-20, Fig. 4; see also Ex-1106, 7:3-6, 9:28-32, Fig. 4.

As shown in Figure 4 (annotated below), the “outgoing message” (the SIP Invite, purple) is being sent from the node (“the node”) APP1 (yellow) to a remote backend server node (“remote node”) BE1 (pink):



Ex-1103, ¶ 97; Ex-1105, 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.”); *see also* Ex-1106, 9:28-32. Forissier further explains that the back-end servers may be integrated with the load-balancer or “may be located externally therefrom, for example in a distributed manner [(“*across a network*”).” Ex-1105, 4:37-41; Ex-1106, 7:3-6; Ex-1103, ¶ 97.

Forissier thus renders this claim obvious.

4. Claim 4

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

Forissier discloses that its method is “*used 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; *see also* Ex-1106, 2:22-24, 2:26-29, 3:3-8, 4:15-16, 6:18-21, 7:8-14, 7:28-8:17, 9:26-10:8, 10:16-25, 10:30-32, 11:10-16.

Specifically, Forissier discloses its method 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.” Ex-1105, 2:3-5; *see also* Ex-1106, 2:22-24. Forissier discloses SIP as “*messaging-based communication*” and explains at length the sending, receiving, and analyzing of SIP messages. *See, e.g.*, Ex-1105, 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; *see also* Ex-1106, 2:26-29, 3:3-8, 4:15-16, 6:18-21, 7:8-14, 7:28-8:17, 9:26-10:8, 10:16-25, 10:30-32, 11:10-16; Ex-1103, ¶ 101. That a POSITA would have recognized SIP as “a *message-based communication system*” is further evident from the ’738 Patent, which identifies SIP as the exemplary message-based communication system being used with the putative invention. *See, e.g.*, Ex-1101, 7:15-27, Fig. 7; Ex-1103, ¶ 101.

Forissier thus renders this claim obvious.

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.***

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; *see also* Ex-1106, 8:1-6, 9:28-32.

In Forissier, the relevant “*context information*” includes the identity of the chosen backend server. *See* Ex-1105, 6:14-20; *see also* Ex-1106, 9:28-32; Ex-1103, ¶ 105. This information, which is added to the “*outgoing message*” (i.e., the SIP Invite from APP1 to BE1), contains “*context information related to the outgoing message,*” including, for instance, “sufficient information to enable the load-balancer to route the message without requiring ... further call context” (“*context*

information”). Ex-1105, 5:9-15; *see also* Ex-1106, 8:1-6. As Dr. Almeroth explains, a POSITA would have understood the identity of the chosen backend server (i.e., the destination of the outgoing message) to be context information related to the outgoing message. Ex-1103, ¶ 105; Ex-1105, 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”).

Forissier thus renders this claim obvious.

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.***

Forissier discloses “*adding the obtained context information by appending the obtained context information to a separate field of the message.*” *See, e.g.,* Ex-1105, 5:6-15, 6:14-29, 3:8-9 (“The destination identifier are preferably inserted into an extension header of a SIP message.”), Fig. 4, Fig. 5, Fig. 6, Fig. 7, Fig. 8 (illustrating step 714 as “add tag to message with b.e. server address”); *see also* Ex-1106, 7:33-8:6, 9:28-10:5, 4:15-16, Fig. 4, Fig. 5, Fig. 6, Fig. 7, Fig. 8.

For example, Forissier 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.

Ex-1105, 5:6-15, 6:14-29, 3:8-9 (“The destination identifier are preferably inserted into an extension header of a SIP message.”), Fig. 4, Fig. 5, Fig. 6, Fig. 7, Fig. 8 (illustrating step 714 as “add tag to message with b.e. server address”); *see also* Ex-1106, 7:33-8:6, 9:28-10:5, 4:15-16, Fig. 4, Fig. 5, Fig. 6, Fig. 7, Fig. 8. As Dr. Almeroth explains, a POSITA would have understood that a tag that has been inserted into an extension header of a SIP message is in a separate field of the message. Ex-1103, ¶ 109; Ex-1117, 11 (“The INVITE request contains a number of header fields. Header fields are named attributes that provide additional information about a message.”).

Forissier thus renders this claim obvious.

7. Claim 7

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

As has been explained claim 1, Forissier discloses use of its methods and systems “*with a session initiation protocol (SIP) network.*” See also, e.g., Ex-1105, 3:6-7 (“The method may be used in conjunction with a session initiation protocol (SIP) based network.”), 3:4-45 (“The load-balancing system may be adapted for use in a session initiation protocol (SIP) based network.”); see also Ex-1106, 4:12-13, 4:9-5:16; Ex-1103, ¶ 113.

Forissier thus renders this claim obvious.

8. Claim 8

- i. ***[8.0] The method of claim 7, wherein the step of adding the obtained context information appends the obtained context information to a SIP TAG field.***

As has been explained for claims 1 and 6, Forissier discloses “*adding the obtained context information*” (i.e., the identity of the chosen server) by “*appending 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; see also Ex-1106, 7:33-8:1, 9:28-32, Fig. 8.

For instance, Forissier 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)” (“*a SIP TAG field*”). Ex-1105, 5:6-9, 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 the chosen backend. [sic] server into the message, and forwards the message to the chosen backend server.”), Fig. 8; *see also* Ex-1106, 7:33-8:1, 9:28-32, Fig. 8. Dr. Almeroth further explains that a POSITA, reading these disclosures within the context of SIP, would have understood that the discussions of adding the context information to a SIP message as a tag means that the context information is being appended to a SIP TAG field. Ex-1103, ¶ 117; 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.”).

Forissier thus renders this claim obvious.

9. Claim 9

- i. ***[9.0] The method of claim 7, wherein the step of adding the obtained context information appends the obtained context information to a SIP extension header.***

As has been explained for claims 1 and 6, 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; *see also* Ex-1106, 4:15-16, 5:25-26, 8:5-6. Specifically, Forissier explains that appending the obtained context information to a SIP extension header is preferable. *See, e.g.*, Ex-1105, 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.”); *see also* Ex-1106, 4:15-16, 5:25-26, 8:5-6. Dr. Almeroth explains that a POSITA would have understood these disclosures of inserting the context information (i.e., the identity of the chosen backend server) as an extension header of a SIP message to mean appending the obtained context information to a SIP extension header. Ex-1103, ¶ 121; Ex-1117, 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.”).

Forissier thus renders this claim obvious.

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.***

Forissier teaches “*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; see also Ex-1106, 8:3-6, 8:21-33, 9:28-32, Fig. 7.

As explained regarding limitation [1.4], Forissier teaches that a tag containing the relevant context information (the identity of the backend server) should be “inserted into the message such that the tag will be included in all future messages sent in response to the message.” Ex-1105, 5:12-15, 6:14-20; see also Ex-1106, 8:3-6, 9:28-32. There is accordingly an expectation that future related messages will contain this context information.

However, Forissier further teaches that incoming messages should be inspected to determine whether the context information is incomplete (i.e., missing). Forissier explains, for instance, that the load-balancer may first inspect an incoming SIP message to determine whether it includes the identity of a chosen backend sever—if that identity has previously been chosen but is missing (i.e., “*the context information is inaccurate or incomplete*”), then Forissier teaches using 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 before it is forwarded to the chosen backend server:

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.

See, e.g., Ex-1105, 5:31-48, Fig. 7; *see also* Ex-1106, 8:21-33, Fig. 7. On the other hand, if the call is new, then 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.

See, e.g., Ex-1105, 5:31-48; *see also* Ex-1106, 8:21-33.

To the extent one might assert that Forissier does not expressly 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,*” as Dr. Almeroth explains, a POSITA would have understood this process of retrieving the identity of an already-chosen backend server to take place in situations where the context information (i.e., the identity of the backend server) is missing (i.e., incomplete), such that Forissier teaches this process should take place “where it is determined that the obtained context information is potentially inaccurate or incomplete,” and that this process results in adding a tag to the message identifying the already-chosen backend server (i.e., adding, to the message, an indication associated with the obtained context information). Ex-1103, ¶ 127.

However, to the extent one argues that omission of the tag does not satisfy “inaccurate or incomplete,” Dr. Almeroth further explains that a POSITA would have found it obvious, and a matter of common sense, to employ 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). Ex-1103, ¶ 128. This would comport with Forissier’s teachings of fault tolerance, and it would be a natural (and obvious) extension of the teachings in Forissier to deal with an incomplete or inaccurate identification of a previously chosen backend server as it would to deal with an omitting identification of a previously chosen backend server. Ex-1103, ¶ 128.

This scenario (involving an incoming response message containing an inaccurate backend server tag, or omitting it entirely) also still satisfies the limitations in claim 1 for the same reasons described for an initial incoming message above. The response message is processed at APP1 and then sent an outgoing message from APP1 to BE1 (limitation 1.1); 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). Ex-1103, ¶ 129. The only difference here is that the message happens to be related to one received earlier.

Forissier thus renders this claim obvious.

11. Claim 33

- i. ***[33.0] A method of sending a message from a node through a hierarchical structure of one or more discreet¹ layers comprising:***

Claim 33 would have been obvious in view of Forissier. For the same reasons described above regarding claim limitation 1.0, to the extent this preamble is limiting, Forissier discloses “*sending a message from a node through a hierarchical structure of one or more discrete layers.*” See, e.g., Ex-1105, 2:3-5, 3:4-45; see also Ex-1106, 2:22-24, 4:9-5:16. Forissier describes using SIP. See, e.g., Ex-1105, 3:6-7 (“The method may be used in conjunction with a session initiation protocol (SIP) based network.”), 3:4-45 (“The load-balancing system may be adapted for use in a session initiation protocol (SIP) based network.”); see also Ex-1106, 4:12-13, 4:9-5:16. Forissier further explains 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.” Ex-1105, 2:3-5; see also Ex-1106, 2:22-23.

¹ The word “discreet” appears to be an obvious typographical error, specifically a misspelling of the word “discrete.” As such, Petitioner has applied the claim as if it used the word “discrete.”

As Dr. Almeroth explains, a POSITA would have understood that SIP, including when used over IP networks, involves sending messages through a hierarchical structure of “*one or more discrete layers.*” Ex-1103, ¶ 133; Ex-1116, 1-2. SIP messaging, for instance, involves multiple layers, including application and transport layers. *Id.* 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. Ex-1103, ¶ 133; Ex-1116, 1-2. Thus, for a SIP message delivered over an IP network, the message will pass through multiple tiered layers (“*hierarchical structure*”), including the application layer and the transport layer. Ex-1103, ¶ 133; Ex-1116, 1-2.

Forissier thus discloses this limitation.

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

For the same reasons described above regarding claim limitation [1.2],

Forissier discloses this limitation.

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

For the same reasons described above regarding claim limitation [1.3],

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.***

For the same reasons described above regarding claim limitation [1.4], 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; see also Ex-1106 7:33-8:3, 9:30-32, 10:12-14, 11:6-8.

To the extent that one may argue that the foregoing does not expressly disclose this limitation, as Dr. Almeroth explains, a POSITA would have understood that the obtained context information (i.e., the identity of the backend server) is context information needed to restore a pre-switchover context of the layer. Ex-1103, ¶ 141. A switchover may occur, for instance, when the chosen backend server goes offline and another backend server must handle the ongoing messages. *Id.*, ¶ 141.

Forissier anticipates such a scenario, explaining 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.” See, e.g., Ex-1105, 6:36-40, 7:4-6; see also Ex-

1106, 10:13-15; 11:6-8; Ex-1103, 142. (“This 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.”).

Dr. Almeroth further explains that a POSITA would have understood these disclosures to mean, or to render obvious, that the added context information (i.e., the obtained identity of the backend server), which is retained in further responses to the message, provides information needed to restore a pre-switchover context (i.e., maintain the SIP call) if the previously obtained backend server is offline and the message must instead be handled by a different backend server. *See* Ex-1103, ¶ 143. This is consistent with both common sense and the disclosures in Forissier, which consistently focus on storing the identity of the appropriate 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. Ex-1105, 5:6-12, 6:16-20; *see also* Ex-1106, 7:33-8:3, 9:30-32; Ex-1103, ¶ 144.

Given these disclosures, and those suggesting that the load-balancing should be fault tolerant, such that future messages can be routed to a new backend server if the previously chosen backend server goes down, a POSITA would have found it obvious to replace the identity of the original chosen backend server (e.g., in the SIP TAG) with the identity of the new backend, thus restoring the pre-switchover context to the transport layer (e.g., the appropriate destination). Ex-1103, ¶¶ 143-

145; 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.

Forissier thus renders this claim obvious.

VIII. DISCRETIONARY DENIAL IS NOT APPROPRIATE

A. The Board Should Not Deny Institution Under 35 U.S.C. § 325(d)

Denial under § 325(d) is not warranted because the challenges presented in this Petition are neither cumulative nor redundant to the prosecution of the ’738 Patent, as Forissier was neither cited nor considered by the Examiner during prosecution of the ’738 Patent. The references in this Petition were neither relied upon during prosecution nor are cumulative of art relied upon during prosecution. Further, the ’738 Patent has not been previously challenged in an IPR. In addition, the Examiner also did not have the benefit of Dr. Almeroth’s detailed expert testimony and the other evidence of record.

Even if it were asserted that this reference was substantially the same or cumulative of the art and arguments raised during prosecution, this Petition’s Grounds identify reference combinations that are not substantially the same as any

considered references, specifically by relying on Forissier (a reference never considered by the Examiner) as disclosing the majority of the claim limitations. *See, e.g., Sony v. MZ Audio Scis., LLC*, No. IPR2022-01544, Paper 12, at 7 (Apr. 21, 2023) (finding Section 325(d) discretion improper where “the Examiner did not consider the specific combination of references asserted during prosecution”). The Board “did not expressly consider” any art with substantially the same teachings as the art in this Petition, rendering it “impossible . . . to determine the consideration the Examiner gave” to such references, and demonstrating that “the Office erred in a manner material to patentability.” *Ecobee Techs. v. Causam Enters., Inc.*, No. IPR2022-01339, Paper 20, at 20-21 (declining to exercise Section 325(d) discretion).

Thus, the Board should not exercise discretion to deny institution.

B. The Board Should Not Deny Institution Under 35 U.S.C. § 314(a)

The merits of Petitioner’s arguments are compelling and the evidence in support is substantial. That “alone demonstrates that the PTAB should not discretionarily deny institution under *Fintiv*.” Ex-1108 at 4-5. But in any event, the six *Fintiv* factors do not justify denying institution.

Factor 1 (stay) is **neutral**. SAP has not yet moved to stay pending this IPR.

Factor 2 (proximity to trial) 2 is **neutral or weighs against denial**. In the originally filed litigation by Patent Owner Valtrus (*Valtrus I*), the trial date is not currently set (though jury selection is set for September 22, 2025). Ex-1109.

However, the Board should not rely on the current schedule in *Valtrus I* because Patent Owner Valtrus filed suit alone (without the other named Patent Owner, Key Patent Innovations) and thus did not have Article III Standing. Ex-1110; Ex-1111. Petitioner filed a Motion to Dismiss for Lack of Standing in *Valtrus I*, reconsideration of which remains pending and casts serious doubt on whether *Valtrus I* will actually proceed along the current schedule. Indeed, in response to Petitioner's Motion to Dismiss for Lack of Standing, Patent Owners essentially concede that Valtrus alone lacked standing because Patent Owners together filed a second, duplicative lawsuit (*Valtrus II*) in an effort to rectify the Article III standing deficiencies in *Valtrus I*. Ex-1112; Ex-1113. This conduct confirms that Patent Owners themselves doubt that the trial schedule in *Valtrus I* will hold, further supporting that the Board should not rely on that schedule at all. Additionally, the other pending litigations involving the challenged patent (*Valtrus II* and a declaratory judgment action filed by Petitioners) do not have any trial date currently set, meaning there is no reliable trial date for the Board to consider in relation to this Petition.

Even if the Board assumes the *Valtrus I* trial date applies, this factor is neutral because there will be other claims not litigated at trial that are covered by this Petition. Therefore, a final written decision will issue first at least on the claims not litigated at trial but covered by this Petition.

Factor 3 (investment in parallel proceedings) **weighs against denial**. Importantly, claims 7-9 are not at issue in the related litigations. Ex-1115, 2; *Samsung Bioepis Co. Ltd v. Regeneron Pharmaceuticals, Inc.*, IPR2023-00739, Paper 9 at 57 (Oct. 20, 2023) (“[B]ecause certain of the challenged claims, . . . are not expressly at issue in the district court litigation, *Fintiv* factor 3 favors institution.”). The related litigations are in its early stages of discovery, motions to transfer and dismiss have been filed, and the investment by the parties has been relatively minimal. All litigations involving the challenged patent are at an early stage, with major deadlines several months away. Ex-1109. Patent Owners may argue that the parties have invested substantial resources on procedural motions, but the resources expended have been directed toward resolving procedural missteps perpetuated by Patent Owner Valtrus’s choice to file alone (without Article III standing) and Patent Owners’ later attempts to fix that with procedurally improper duplicative litigation. These procedural disputes (caused by Patent Owners’ actions) should not be held against Petitioner under this factor.

Factor 4 (overlap) **weighs against denial** because Petitioner hereby stipulates that, if the IPR is instituted, it will not pursue the IPR grounds in the district court. Indeed, Forissier is not at issue in the District Court Litigation. Ex-1114. Thus, “[i]nstituting trial here serves overall system efficiency and integrity goals by not duplicating efforts and by resolving materially different patentability issues.” *Apple*,

Inc. v. SEVEN Networks, LLC, IPR2020-00156, Paper 10 at 19 (June 15, 2020); *see also Sand Revolution II, LLC v. Continental Intermodal Group-Trucking LLC*, IPR2019-01393, Paper 24 at 12 (June 16, 2020). In addition, Petitioner challenges more claims than Patent Owners will ultimately assert at trial. Patent Owner Valtrus agreed to narrow to 30 total asserted claims by April 30, 2025, and a maximum of six asserted claims per asserted patent. Ex-1109, 4. Thus, Patent Owner will be limited to asserting up to six of the eight asserted claims of the '738 Patent, merely a subset of the claims challenged in this Petition. *See, e.g., Apple*, No. IPR2020-00156, Paper 10, at 17, 21-22.

Factor 6 (other circumstances) **weigh against denial**. Here, the merits of the Petition are particularly compelling—for example, Forissier is a strikingly similar disclosure to that of the '738 Patent, is directed at the same problem and proposes the same solutions as the '738 Patent and discloses nearly identical architecture, including several nearly identical figures. The evidence presented, if unrebutted in trial, would plainly lead to a conclusion that one or more claims are unpatentable by a preponderance of the evidence.

While Factor 5 (parties) may weigh slightly in favor of denial because the parties are the same as in district court, under the “holistic view” of the remaining factors and whether integrity of the system and efficiency is best served, institution is

avored. *Samsung Elecs. Co. Ltd. v. Dynamics Inc.*, IPR2020-00505, Paper 11 at 15 (Aug. 12, 2020).

IX. CONCLUSION

Claims 1-10 and 33 of the '738 Patent are unpatentable for the reasons demonstrated above.

Dated: January 15, 2025

Respectfully submitted,

/s/ Louis L. Campbell

Louis L. Campbell (Lead Counsel)

llcampbell@winston.com

USPTO Reg. No. 59,963

Winston & Strawn LLP

255 Shoreline Dr., Ste. 520

Redwood City, CA 94065

T: (650) 858-6500

Lead Counsel for Petitioner

CERTIFICATE OF COMPLIANCE

This Petition complies with the word count limits set forth in 37 C.F.R. § 42.24(a)(1)(i), because this Petition contains 11,103 words, excluding the parts of the Petition exempted by 37 C.F.R. § 42.24(a)(1), determined using the word count provided by Microsoft Word, which was used to prepare this Petition.

Dated: January 15, 2025

Respectfully submitted,

/s/ Louis L. Campbell

Louis L. Campbell (Lead Counsel)

llcampbell@winston.com

USPTO Reg. No. 59,963

Winston & Strawn LLP

255 Shoreline Dr., Ste. 520

Redwood City, CA 94065

T: (650) 858-6500

Lead Counsel for Petitioner

CERTIFICATE OF SERVICE

Under 37 C.F.R. §§ 42.6(e) and 42.105(a), this is to certify that on January 15, 2025, I caused to be served a true and correct copy of the foregoing “**PETITION *INTER PARTES* REVIEW OF U.S. PATENT NO. 7,936,738**” and accompanying exhibits by Federal Express priority mail on the following:

Botos Churchill IP Law LLP - Malik
430 Mountain Avenue
Suite 401
New Providence, NJ 07974

With a courtesy copy of this Petition and supporting material was also served on litigation counsel for Patent Owners via email:

Matt Berkowitz

mberkowitz@reichmanjorgensen.com

Patrick R. Colsher

pcolsher@reichmanjorgensen.com

Aaron Morris

amorris@reichmanjorgensen.com

Navid Cyrus Bayar

nbayar@reichmanjorgensen.com

Reichman Jorgensen Lehman & Feldberg LLP
100 Marine Parkway
Suite 300
Redwood Shores, CA 94065

Khue Van Hoang

khoang@reichmanjorgensen.com

Reichman Jorgensen Lehman & Feldberg LLP
400 Madison Avenue
Suite 14d
New York, NY 1001

Philip Eklem

peklem@reichmanjorgensen.com

Savannah Harman Carnes
scarnes@reichmanjorgensen.com
Reichman Jorgensen Lehman & Feldberg LLP
1909 K Street, NW
Suite 800
Washington, DC 20006

/s/ Louis L. Campbell
Louis L. Campbell (Lead Counsel)
llcampbell@winston.com
USPTO Reg. No. 59,963
Winston & Strawn LLP
255 Shoreline Dr., Ste. 520
Redwood City, CA 94065
T: (650) 858-6500

Lead Counsel for Petitioner