

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Liberty Mutual Insurance Company, Liberty Mutual Technology Group, Inc.,
Liberty Mutual Holding Company Inc., Liberty Mutual Group Inc., Liberty Mutual
Plano LLC, Comparion Insurance Agency, LLC, Ironshore Holdings (U.S.) Inc.,

and

Comerica Incorporated

Petitioners

v.

Intellectual Ventures I LLC

Patent Owner

IPR2025-00200
U.S. Patent No. 8,407,722

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 8,407,722**

Mail Stop Patent Board
Patent Trial and Appeal Board
U.S. Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	GROUND FOR STANDING.....	2
III.	IDENTIFICATION OF CHALLENGE	2
IV.	THE 722 PATENT	3
	A. Overview	3
	B. Prosecution History	7
	C. POSA.....	9
V.	CLAIM CONSTRUCTION	9
	A. Claims 17-19, 23-25, 29-31, 35-37: “activation module”.....	10
	B. Other claims.....	12
VI.	GROUND 1: Claims 1-37 are obvious over Lawson alone or combined with Tsutsumitake	12
	A. Overview	12
	1. Lawson (EX1005)	12
	2. Tsutsumitake (EX1006)	14
	3. Motivation to Combine	14
	B. Detailed Analysis.....	17
	1. Claims 14-37	17
	Claim 14	17
	Claim 15	34
	Claim 16	34
	Claim 17	36
	Claim 18	37
	Claim 19	39
	Claims 20-37	41
	2. Claims 1 and 7.....	41
	3. Claim 2 and 8	43
	4. Claims 3-5 and 9-13.....	44

5. Claim 6	44
VII. GROUND 2: Claims 1-37 are obvious over Kostas alone or combined with Tsutsumitake.....	46
A. Overview	46
1. Kostas (EX1007).....	46
2. Tsutsumitake (EX1007).....	48
3. Motivation to Combine	48
B. Detailed Analysis.....	50
1. Claims 14-37	50
Claim 14	50
Claim 15	57
Claim 16	58
Claim 17	59
Claim 18	59
Claim 19	60
Claims 20-37	61
2. Claims 1 and 7.....	61
3. Claim 2 and 8	63
4. Claims 3-5 and 9-13.....	64
5. Claim 6.....	64
VIII. DISCRETIONARY DENIAL IS NOT APPROPRIATE	65
A. 35 U.S.C. § 325(d).....	65
B. 35 U.S.C § 314(a).....	65
IX. MANDATORY NOTICES	67
A. Real Party-in-Interest	67
B. Related Matters.....	68
C. Lead and Backup Counsel and Service Information.....	69
X. CONCLUSION.....	69

EXHIBIT LIST

EX	Description
1001	U.S. Patent No. 8,407,722 B2, entitled “Asynchronous messaging using a node specialization architecture in the dynamic routing network,” filed 3/30/2006 and issued 3/26/2013, issued to Tuttle (“ <u>722 patent</u> ”)
1002	File History for U.S. Patent Application No. 11/396,251 (issued as 722 patent)
1003	Expert Declaration of Dr. William C. Easttom II (Chuck Easttom), Ph.D., D.Sc. (“ <u>Easttom Decl.</u> ”)
1004	Curriculum Vitae of Dr. William C. Easttom, Ph.D., D.Sc.
1005	U.S. Patent No. 5,721,825, entitled “System and method for global event notification and delivery in a distributed computing environment,” filed 10/3/1996 and issued 2/24/1998, issued to Lawson (“ <u>Lawson</u> ”)
1006	U.S. Patent No. 6,480,883, entitled “Real-Time Information Transmission System,” filed 6/29/1999, issued to Tsutsumitake (“ <u>Tsutsumitake</u> ”)
1007	WO 99/26127, entitled “System and Method For Displaying Multiple Sources of Data In Near Real-Time,” filed 11/14/1997 and published 5/27/1999, named inventor Kostas (“ <u>Kostas</u> ”)
1008	File History for U.S. Patent Application No. 60/256,613
1009	File History for U.S. Patent Application No. 60/276,847
1010	File History for U.S. Patent Application No. 60/278,303
1011	File History for U.S. Patent Application No. 60/279,608
1012	File History for U.S. Patent Application No. 60/280,627
1013	File History for U.S. Patent Application No. 10/017,182
1014	File History for U.S. Patent Application No. 10/105,018
1015	722 patent Claims 14-37 Concordance
1016	722 patent Claims 1-13 Concordance
1017	<i>Intellectual Ventures v. Liberty Mutual</i> , No. 23-cv-525 (E.D. Tex.) – Infringement Contentions
1018	<i>Intellectual Ventures v. Comerica</i> , No. 23-cv-524 (E.D. Tex.) – Infringement Contentions
1019	Reserved
1020	Reserved

EX	Description
1021	Hohpe, Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions, Chapter 3: Messaging Systems
1022	Cornell University, Nodes; Parallel Programming Concepts and High Preformance Computing
1023	Northeastern, ECE 333: Introduction Communication Networks Fall 2002
1024	Zen and the Art of the Internet, Glossary
1025	Microsoft Computer Dictionary, 5th Edition, 2002
1026	Internetworking Technologies Handbook, Routing Basics, Chapter 5
1027	Juniper Networks, Inc., Routing Protocols Overview
1028	CloudFlare, What is Routing? IP routing
1029	Barrons, Dictionary of Computer and Internet Terms, 10th Edition
1030	NYU Networks at ITP, Proxies
1031	Amazon, What is Routing?
1032	IPXO, A Comprehensive Guide to Network Routing, August 28. 2024
1033	Dictionary of Electrical and Computer Engineering, The McGraw Hill Companies
1034	NETGEAR DSL Modem Internet Gateway Installation Guide

Index of Challenged Claims

Claims
[1.0] A method comprising:
[1.1] receiving, using a processing device, an update message from an input source, the update message identifying a live object and containing data for updating a property of the live object;
[1.2] identifying a category of the update message based on the input source;
[1.3] determining a node having a node type to which the update message is to be routed based on a mapping of categories of update messages to node types, the mapping controlling an amount of update message traffic through nodes of a routing network;
[1.4] routing, using the processing device, the update message to the node having the determined node type;
[1.5] causing the node, through the update message, to determine a client, different from the input source, that has registered for updates of the live object;
[1.6] causing the node to route the update message from the node to the client; and
[1.7] causing the client to process the update message and to update the property of the live object.
[2.0] The method of claim 1, further comprising: causing the node to extract an object identifier (ID) from the update message, to establish a connection with the client, and to determine if the client has registered for updates of the live object based on the object ID.
[3.0] The method of claim 2, further comprising:
[3.1] causing the node to determine at least one client proxy with which the client communicates;
[3.2] causing the node to route the update message to the client proxy; and
[3.3] causing the client proxy to route the update message to the client to determine the connection.

Claims
[4.0] The method of claim 3, further comprising causing the node to maintain client registration information concerning the client connection.
[5.0] The method of claim 3, further comprising causing the node to maintain client registration information concerning the client connection at the client proxy.
[6.0] The method of claim 1, wherein the property of the live object has a direct effect on a visual representation of the live object in a data representation, has an effect on an internal aspect of the live object and has no effect on the visual representation of the live object in the data representation, or has a direct effect on one aspect of the visual representation of the live object in the data representation and has no effect on other aspect of the visual representation of the live object in the data representation.
[7.0] A routing network comprising:
[7.1] a gateway device configured to:
[7.2] receive an update message from an input source, the update message identifying a live object and containing data for updating a property of the live object,
[7.3] identify a category of the update message based on the input source;
[7.4] determine a node having a node type to which the update message is to be routed based on a mapping of a categories of update messages to node types, the mapping controlling an amount of update message traffic through nodes of a routing network; and route the update message; and
[7.5] the node configured to:
[7.6] receive the update message from the gateway device, wherein the node is configured to be mapped to the node type,
[7.7] determine a client, different from the input source, that has registered for updates of the live object, and
[7.8] route the update message from the node to the client, wherein the client is adapted to process the update message and to update the property of the live object.

Claims
[8.0] The routing network of claim 7, wherein the node is configured to extract an object identifier (ID) from the update message and to determine a connection to the client to route the data to the client.
[9.0] The routing network of claim 8, wherein
[9.1] the node is configured to determine the connection by determining at least one client proxy with which the client communicates and
[9.2] to route the update message to the client proxy.
[10.0] The routing network of claim 9, wherein client registration information concerning the client connection is configured to be maintained at the node.
[11.0] The routing network of claim 9, wherein client registration information concerning the client connection is configured to be maintained at the client proxy.
[12.0] The routing network of claim 7, wherein:
[12.1] the gateway device is configured to route the update message to the node adapted to receive messages of more than one message category, and
[12.2] the node is configured to determine a connection with at least one client proxy with which the client communicates and to route the data to the client proxy.
[13.0] The routing network of claim 7, wherein the node type is configured to identify the node that receives the update message from the gateway device.
[14.0] A method comprising:
[14.1] providing, using a processing device of an input source, a data representation to a client device, different from the input source, coupled to a routing network, wherein the data representation includes at least one live object recognizable by the client device, and
[14.2] causing the client device to respond to the live object of the data representation by determining an object identifier (ID) of the live object and to register for updates of the live object with the routing network, such that registering the client device with the routing network provides client connection information to a node in the routing network; and

Claims
[14.3] sending, using the processing device of the input source, an update message to the routing network, wherein the update message identifies the live object and contains update data that updates a property of the live object,
[14.4] wherein a gateway device at the routing network is configured to identify a category of the update message based on the input source, to determine a node type to which the identified category maps, and to route the update message to the node, having the node type, at the routing network,
[14.5] wherein the node is configured to identify the client device as a registered device and to route the update message to the client device, and
[14.6] wherein the client device processes the update message upon receipt to update the property of the live object at the client device.
[15.0] The method of claim 14, wherein providing the data representation to the client device includes providing the live object that causes the client device to register with a client proxy of the routing network.
[16.0] The method of claim 14, wherein providing the data representation to the client device includes providing the live object that causes the client device to register with the node of the routing network.
[17.0] The method of claim 14, wherein providing the data representation to the client device includes providing an activation module that is executed by the client device and that registers the live object with the routing network.
[18.0] The method of claim 17, wherein providing the activation module includes providing the activation module that is configured to determine a node type that handles registration and that causes the client device to register with the node corresponding to the node type.
[19.0] The method of claim 17, wherein providing the activation module includes providing the activation module that is configured to determine a message category of the data representation and that causes the client device to register with a node having a node type corresponding to the message category.
[20.0] An apparatus comprising:

Claims
[20.1] an input source device configured to provide a data representation to a client device, different from the input source, coupled to a routing network, wherein the data representation includes at least one live object that is recognizable by the client device, and
[20.2] that causes the client device to determine an object identifier (ID) of the live object to register for updates of the live object with the routing network, such that registering the client device with the routing network provides client connection information to the routing network,
[20.3] wherein the input source device is configured to route an update message to the routing network, wherein the update message identifies the live object and contains update data for updating a property of the live object,
[20.4] wherein a gateway device at the network is configured to identify a category of the update message based on the input source, to determine a node type to which the identified category maps, and to route the update message to a node of the node type at the routing network,
[20.5] wherein the node is configured to identify the client device as a registered device and route the update message containing the update data to the client device, and
[20.6] wherein the client device is configured to process the update message upon receipt to update the property of the live object at the client device.
[21.0] The apparatus of claim 20, wherein the live object of the data representation is configured to cause the client device to register with a client proxy of the routing network.
[22.0] The apparatus of claim 20, wherein the live object of the data representation is configured to cause the client device to register with the node of the routing network.
[23.0] The apparatus of claim 20, wherein the received data representation includes an activation module that is configured to be executed by the client device and adapted to register the live object with the routing network.

Claims
[24.0] The apparatus of claim 23, wherein the activation module is configured to determine a node type for handling registration and to cause the client device to register with the node of the node type.
[25.0] The apparatus of claim 23, wherein the activation module determines a message category of the data representation and causes the client device to register with the node having a node type corresponding to the message category.
[26.0] An article of manufacture including a computer-readable storage medium having instructions stored thereon, execution of which by a computing device causes the computing device to perform operations comprising:
[26.1] providing, using a processing device of an input source, a data representation to a client device, different from the input source, coupled to a routing network, wherein the data representation includes at least one live object that is recognizable by the client device, and
[26.2] that causes the client device to respond to the live object by determining an object identifier (ID) of the live object to register for updates of the live object with the routing network, such that registering the client device with the routing network provides client connection information to the routing network; and
[26.3] sending, using the processing device of the input source, an update message to the routing network, wherein the update message identifies the live object and contains update data for updating a property of the live object,
[26.4] wherein a gateway device at the routing network is configured to identify a category of the update message based on the input source, to determine a node type to which the identified category maps, and to route the update message to a node of the node type at the routing network,
[26.5] wherein the node is configured to identify the client device as a registered device and to route the update message to the client device, and
[26.6] wherein the client device is configured to process the update message upon receipt to update the property of the live object at the client device.
[27.0] The article of manufacture of claim 26, wherein the live object of the data representation causes the client device to register with a client proxy of the routing network.

Claims
[28.0] The article of manufacture of claim 26, wherein the live object of the data representation causes the client device to register with the node of the routing network.
[29.0] The article of manufacture of claim 26, wherein the received data representation includes an activation module executed by the client device and adapted to register the live object with the routing network.
[30.0] The article of manufacture of claim 29, wherein the activation module determines a node type that handles registration and causes the client device to register with the node of the node type.
[31.0] The article of manufacture of claim 29, wherein the activation module determines a message category of the data representation and causes the client device to register with the node having a node type corresponding to the message category.
[32.0] A non-transitory computer readable storage medium having instructions stored thereon, the instructions comprising:
[32.1] instructions for providing a data representation to a client device coupled to a routing network, wherein the data representation includes at least one live object recognizable by the client device, and
[32.2] wherein the client device is configured to respond to the live object of the data representation by determining an object identifier (ID) of the live object to register for updates of the live object with the routing network, such that registering the client device with the routing, network provides client connection information to the routing network; and
[32.3] instructions for providing, using a processing device of an input source, different from the client device, an update message to the routing network, wherein the update message identifies the live object and contains update data for updating a property of the live object,
[32.4] wherein a gateway device at the routing network is configured to identify a category of the update message based on the input source, to determine a node type to which the identified category maps, and to route the update message to a node of the node type at the routing network,

Claims
[32.5] wherein the node is configured to identify the client device as a registered device and to route the update message containing the update data to the client device, and
[32.6] wherein the client device is configured to process the update message upon receipt to update the property of the live object at the client device.
[33.0] The non-transitory computer readable storage medium of claim 32, wherein the live object of the data representation causes the client device to register with a client proxy of the routing network.
[34.0] The non-transitory computer readable storage medium of claim 32, wherein the live object of the data representation causes the client device to register with the node of the routing network.
[35.0] The non-transitory computer readable storage medium of claim 32, wherein the data representation includes an activation module that is executed by the client device and that is adapted to register the live object with the routing network.
[36.0] The non-transitory computer readable storage medium of claim 35, wherein the activation module determines a node type for handling registration and causes the client device to register with the node of the node type.
[37.0] The non-transitory computer readable storage medium of claim 35, wherein the activation module determines a message category of the data representation and causes the client device to register with the node having the node type corresponding to the message category.

I. INTRODUCTION

Liberty Mutual Insurance Company, Liberty Mutual Technology Group, Inc., Liberty Mutual Holding Company Inc., Liberty Mutual Group Inc., Liberty Mutual Plano LLC, Comparion Insurance Agency, LLC, Ironshore Holdings (U.S.) Inc. (collectively, “LM”), and Comerica Incorporated (“Comerica”) (collectively, “Petitioners”) petition for *inter partes* review (“IPR”) of claims 1-37 (all claims) (the “challenged claims”) of U.S. Patent No. 8,407,722 titled “Asynchronous Messaging Using a Node Specialization Architecture in the Dynamic Routing Network” (“the 722 patent,” EX1001).¹

The 722 patent generally relates to routing update messages from a server over a network to update data displayed by clients in real-time. 722 patent, Abstract. The purported novelty is the system’s ability to selectively send update messages based on the identity of the sender, instead of the content of the message (i.e., route messages based on the publisher or source (e.g., CNN), instead of the message content (e.g., stock quote)). See EX1002, 1014, 1058. However, message update systems (also called “event/eventing” systems) that could filter and route update messages (also called “events”) based on any criteria, including the identity of the publisher or source were well-known before the 722 patent’s earliest priority date.

¹ The Office is authorized to charge fees to Deposit Account 50-3725.

Petitioners, supported by the declaration of William C. Easttom II, Ph.D., D.Sc. who has 30 years of experience, demonstrate that the challenged claims are unpatentable. (“Easttom Decl.” EX1003, CV EX1004.)

II. GROUNDS FOR STANDING

Petitioners certify the 722 patent is available for IPR, and Petitioners are not barred or estopped from requesting IPR of the challenged claims on the grounds below.

III. IDENTIFICATION OF CHALLENGE

Petitioners request IPR and cancellation of all claims (claims 1-37) of the 722 patent on the following grounds:

Ground	Challenged Claims	Basis for Rejection
1	1-37	§ 103: Lawson (US 5,721,825) and Tsutsumitake (US 6,480,883)
2	1-37	§ 103: Kostas (WO 99/26127) and Tsutsumitake

The 722 patent is **not** entitled to its earliest claimed priority date of 12/18/2000, but every prior art reference in Grounds 1-2 precedes that date.

Reference	Prior Art Date	Status
Lawson (EX1005)	Filed 10/3/1996 Issued 2/24/1998	§ 102(e) § 102(b)
Tsutsumitake (EX1006)	Filed 6/29/1999	§ 102(e)
Kostas (EX1007)	Filed 11/14/1997 Published 5/27/1999	§ 102(e) § 102(b)

IV. THE 722 PATENT

A. Overview

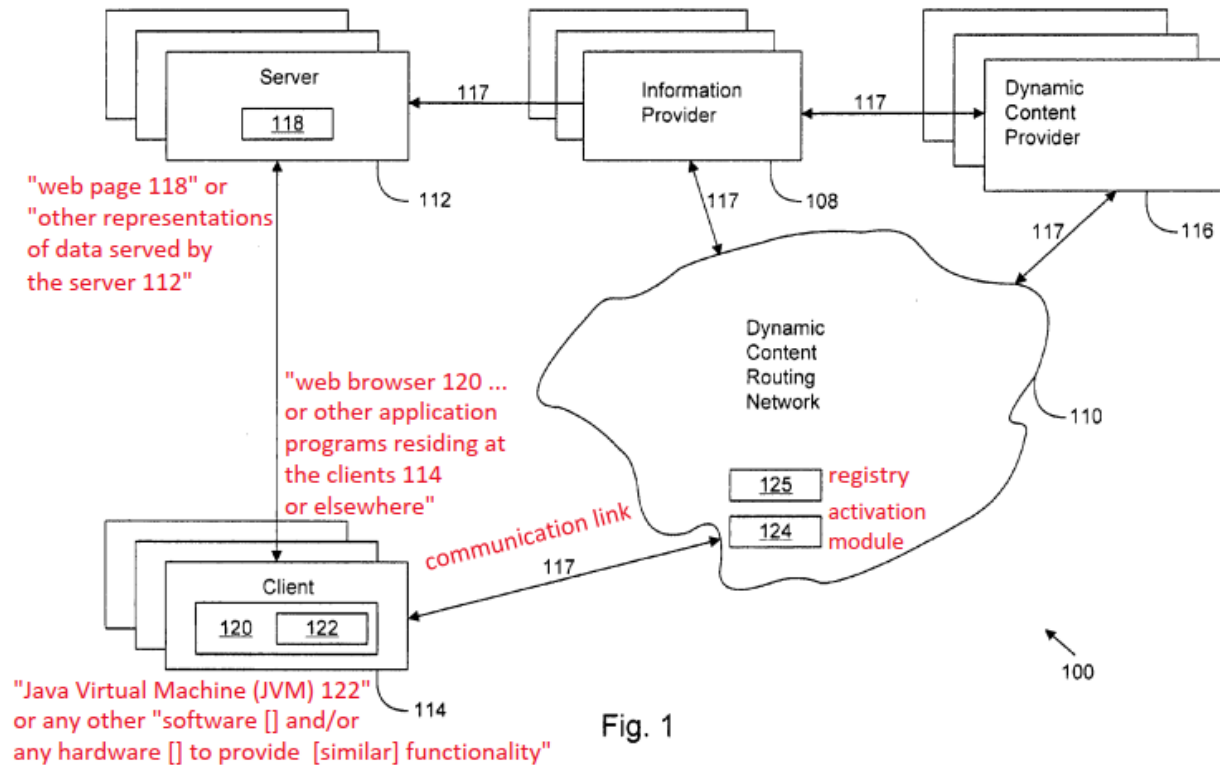
The 722 patent “pertains ... to transferring information for remotely updating content at client devices through the digital networks.” 1:24-27. Representative claim 14 recites:

14. A method comprising:
providing, using a processing device of an input source, a data representation to a client device, different from the input source, coupled to a routing network, wherein the data representation includes at least one live object recognizable by the client device, and causing the client device to respond to the live object of the data representation by determining an object identifier (ID) of the live object and to register for updates of the live object with the routing network, such that registering the client device with the routing network provides client connection information to a node in the routing network; and sending, using the processing device of the input source, an update message to the routing network, wherein the update message identifies the live object and contains update data that updates a property of the live object, wherein a gateway device at the routing network is configured to identify a category of the update message based on the input source, to determine a node type to which the identified category maps, and to route the update message to the node, having the node type, at the routing network, wherein the node is configured to identify the client device as a registered device and to route the update message to the client device, and wherein the client device processes the update message upon receipt to update the property of the live object at the client device.

Fig. 1 (annotated below)² illustrates the input source (e.g., dynamic content provider 116, information provider 108, server 112 alone or combined), client device 114 (e.g., PC, phone, PDA, web-browsing appliance, other devices), routing network 110, and registry 125 identifying clients who have registered for live objects, and a “web page” (which can be any text, data, and/or programs) to clients 114 over communication links 117 (e.g., network, Internet). 4:11-12, 4:55-5:9, 6:19-30, 6:53-

² All emphasis in quotations and annotations/highlights in figures has been added, unless otherwise noted. Citations under named headings are generally to the named documents (e.g., “Lawson” heading to EX1005 for Lawson).

60, 10:14-23. A few servers and clients are shown in Fig. 1, but there could be thousands or even millions. 4:60-63.



The “routing network [] routes messages [from the input source (e.g., information/content provider or its server)] containing data for updating properties of live objects to clients displaying web pages or other representations of data.” 3:9-12, 6:22-29; *see also* 11:21-43, 12:24-25, Figs. 2, 3. “The web server that initially provides the web pages to the clients does not need to track which clients are currently displaying the live objects. Instead, the ... input source [] ... simply sends an update message to the routing network” and “the update messages are provided to the clients only when the live object changes.” 3:12-18. The “live objects,” which

are designated to be real-time dynamically-updateable objects[, are] identified by an object identifier, or object ID.” 5:28-31. The “objectID [] is preferably a string. The string can encode any information ... [and can be] a simple textual and/or numeric identifier” (e.g., “Bang\$” for the stock symbol “BANG”). 8:34-37, 8:62-65.

“The clients 114 contact the routing network 110 and register for update messages for the objectIDs [of the live objects] on the web page[/‘other application programs residing at the clients 114 or elsewhere’]. The routing network 110, in turn, preferably maintains a registry indicating which clients have registered 35 for which object IDs.” 5:32-36, 6:58-59. Input sources send update messages “any time the[y] want[] to update a property of a live object. Each update message preferably identifies a live object [by object ID] and contains data for updating a property of the identified live object. The routing network 110 accesses the registry and determines which clients have registered for the identified object. Then, the routing network 110 routes the update message to the appropriate clients. Upon receipt of an update message, the clients 114 update the specified property of the live object.” 5:41-48, 9:59-62. The “property” to update “include[s] any modifiable data related to the [live] object or referenced with respect to the object” (e.g., updates to “stock prices and/or sports scores” or any other data at the client which “may be a simple numerical or textual value, e.g., ‘4’”). 7:23-24, 8:57-63, 9:53-56. Fig. 4 shows two

differently-formatted webpages 410, 412 with the same “scores object 416” set to the score of the game “42” to “7.” 12:37-46, 13:3-18.

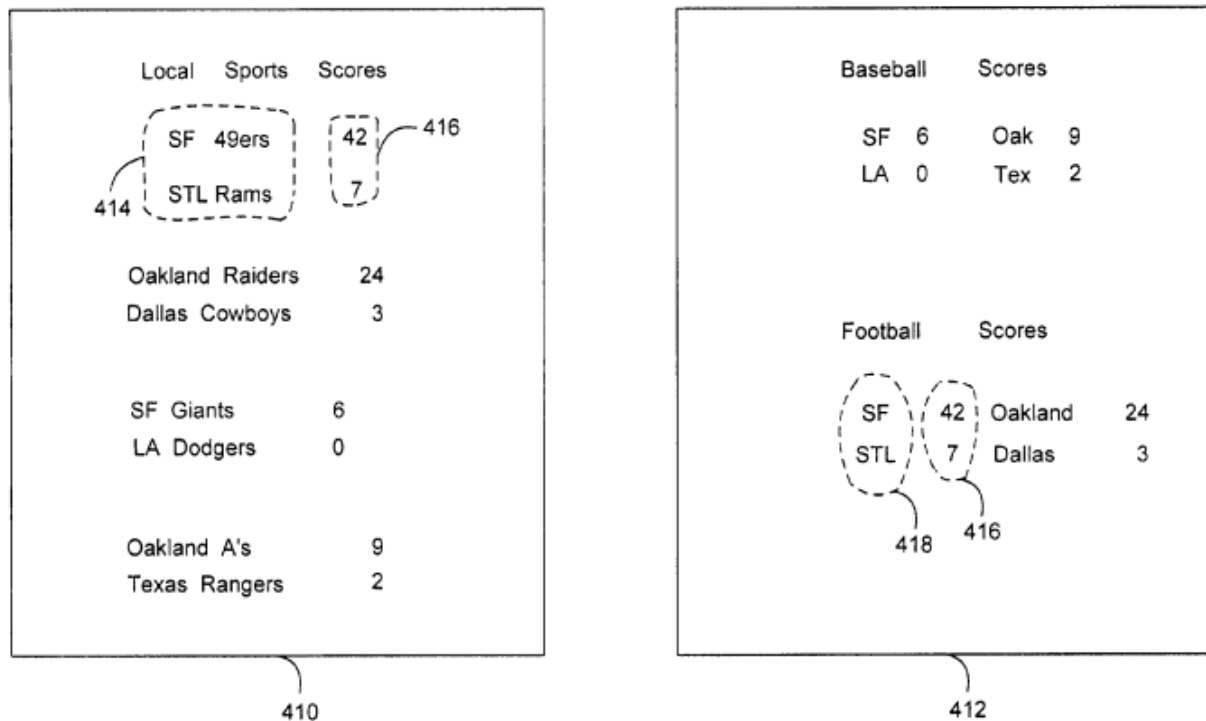


Fig. 4

“[T]o update the score of ... the game on both web pages, the input source 210 simply sends an update message to the routing network 110 specifying the objectID for the scores object 416 and the update to the score property. The routing network 110 routes the update message to the appropriate clients 114, and the clients update the appropriate score regardless of the particular page layout.” 13:18-25.

“The routing network is adapted to selectively send messages to the nodes in the network” via a gateway. 3:21-27. In an embodiment, updates “from an input source are assigned to one or more of N categories” and “the nodes are assigned to

one or more of M types, and mappings are created between message categories and node types. Each gateway keeps track of these mappings” (e.g., category 1 data sent to type 1 nodes). 3:39-46; *see also* 18:24-51, Fig. 9. One way the 722 patent teaches assigning messages to categories “is to assign all message from a given input source [(e.g., information/content provider or its server] into a certain category.” 18:52-53, 6:22-29. “When a gateway receives messages from input sources, the gateway identifies the categories of the messages and routes the messages to the nodes of the types to which the categories are mapped,” and the nodes send the update messages to registered client devices. 3:43-46, 18:66-19:5.

B. Prosecution History

During prosecution, the 722 patent claims were rejected by the examiner numerous times, including under § 102 and 103 over various prior art. EX1002, 413-31 (non-final rejection), 496-517 (non-final rejection), 575-606 (final rejection), 646-81 (non-final rejection), 723-63 (final rejection), 802-41 (non-final rejection), 892-938 (final rejection).³ Of particular relevance to allowance, applicant amended the independent claims in response to the second final rejection to recite “identifying a category of the update message based on the input source” (or similar language). EX1002, 792-801. In the third final rejection, however, the examiner maintained the

³ The provisional and parent file histories are at EX1008-14.

rejections and found the claims obvious under § 103 over U.S. 6,091,724 (“Chandra”), U.S. 6,510,323 (“Stocker”), and U.S. 6,324,578 (“Trenbeath”). EX1002, 892-938.

Applicants appealed. EX1002, 997-1030. Applicants alleged the limitation “identifying a category of the update message based on the input source” (or similar language) distinguished the claims over the prior art. EX1002, 1001-18. In particular, applicants argued “Chandra teaches identifying categories ***based on category/topic of the message content***” (i.e., “the ***data content of the message***”), but the claimed invention “identif[ies] a category of the update message ***based on the input source.***” EX1002, 1010-11 (emphasis original). Applicant argued:

Moreover, following the Examiner’s example, if a particular subscriber selects a category of “stocks”, the particular subscriber is subscribed to “stocks” type of messages. Chandra determines whether a data content of a message matches the specified values of the subscription (such as “stock”), then the message is delivered to the particular subscriber. Chandra does not identify messages based on the publisher, which the Examiner analogizes as input source of independent claims, to which Applicants do not concede.

EX1002, 1014.

The claims were allowed based on these appeal arguments. EX1002, 1058.

C. POSA

A person of skill in the art (“POSA”) for the 722 patent, as of the claimed earliest priority date 12/18/2000, would have had: a bachelor’s degree in an academic area emphasizing computer networking (i.e. computer science, computer information systems, computer engineering, etc.) and (2) at least two years of work experience with Internet routing with additional education substituting for less experience and vice versa. For example, someone with a relevant master’s degree and 1 year of experience could qualify as one of ordinary skill in the art. Similarly, someone with no degree but five years of relevant experience might qualify as one of ordinary skill in the art. Easttom Decl. ¶24.

Petitioners’ independent expert, Dr. Easttom, was qualified as POSA on the earliest claimed priority date and remains qualified to testify about what POSA would have understood. Easttom Decl. ¶¶1-6, 23; EX1004.

V. CLAIM CONSTRUCTION

Claim terms are construed according to the *Phillips* standard in an IPR. *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005); 37 C.F.R. § 42.100. However, claim construction is only required where necessary to resolve a dispute. *See, e.g., Toyota Motor Corp. v. Cellport Sys., Inc.*, IPR2015-00633, Paper 11 at 16 (Aug. 14, 2015) (citing *Wellman, Inc. v. Eastman Chem. Co.*, 642 F.3d 1355, 1361 (Fed. Cir. 2011)). Petitioners propose the following claim constructions.

A. Claims 17-19, 23-25, 29-31, 35-37: “activation module”

Claims 17-19, 23-25, 29-31, 35-37 recite an “activation module”⁴ for performing certain functions. “Module” is a nonce word that substitutes for “means” for purposes of § 112, ¶6. *See Williamson v. Citrix Online, LLC*, 792 F.3d 1339, 1350 (Fed. Cir. 2015) (en banc). Claim set 17-19, discussed below, is illustrative. Claims 23-25, 29-31, 35-37 are substantially identical. *See* EX1015 (claims 14-37 concordance). Easttom Decl., ¶¶37-38.

[17.0] The method of claim 14, wherein providing the data representation to the client device includes providing an activation module that is executed by the client device and that registers the live object with the routing network.	Nonce word for “means” function without recitation of structure to perform it
[18.0] The method of claim 17, wherein providing the activation module includes providing the activation module that is configured to determine a node type that handles registration and that causes the client device to register with the node corresponding to the node type.	
[19.0] The method of claim 17, wherein providing the activation module includes providing the activation module that is configured to determine a message category of the data representation and that causes the client device to register with a node having a node type corresponding to the message category.	

⁴ In the litigation, Patent Owner asserts claims 17 and 18 and reads the “activation module” very broadly against commonly and widely used open-source software known as Kafka. *See* EX1017-18, claims 17-18. Patent Owner contends this term is not a means-plus-function limitation.

Means-plus-function claims are construed by identifying the claimed function and the corresponding structure in the specification for performing it. See *Williamson*, 792 F.3d at 1351. The chart below provides these constructions.

Claims	Function	Structure in 722 Specification
17, 23, 29, 35	“executing instructions by the client for registering the live object with the routing network”	“activation module 124[: (1)] parses 610 the web page 118 received from the server 112 and identifies the objectIDs of the live objects,” (2) “opens 612 a connection between the client 114 and the routing network 110,” and (3) “uses the connection ... to register 614 the object IDs by communicating to the routing network 116 [] the identified object IDs [(e.g., as a vector, list, or array)]” and equivalents thereof. 14:33-15:1, Fig. 6; <i>see also</i> 11:1-8.
18, 24, 30, 36	“determining a node type that handles registration and causing the client device to register with the node corresponding to the node type”	activation module 124: (1) determines a node type in the routing network that handles registration, (2) makes a connection to that node, and (3) “registers the appropriate object IDs with the appropriate node[]” and equivalents thereof. 21:43-52.
19, 25, 31, 37	“determining a message category of the data representation and causing the client device to register with a node having a node type corresponding to the message category”	activation module 124: (1) “determines which categories of messages it needs and to which nodes types it should connect,” (2) “makes a connection to one node of each relevant type,” and (3) “registers the appropriate object IDs with the appropriate nodes” and equivalents thereof. 21:43-52.

B. Other claims

No express construction of the other challenged claims is necessary in this IPR because the prior art renders these claims obvious under any reasonable construction.⁵

VI. GROUND 1: Claims 1-37 are obvious over Lawson alone or combined with Tsutsumitake

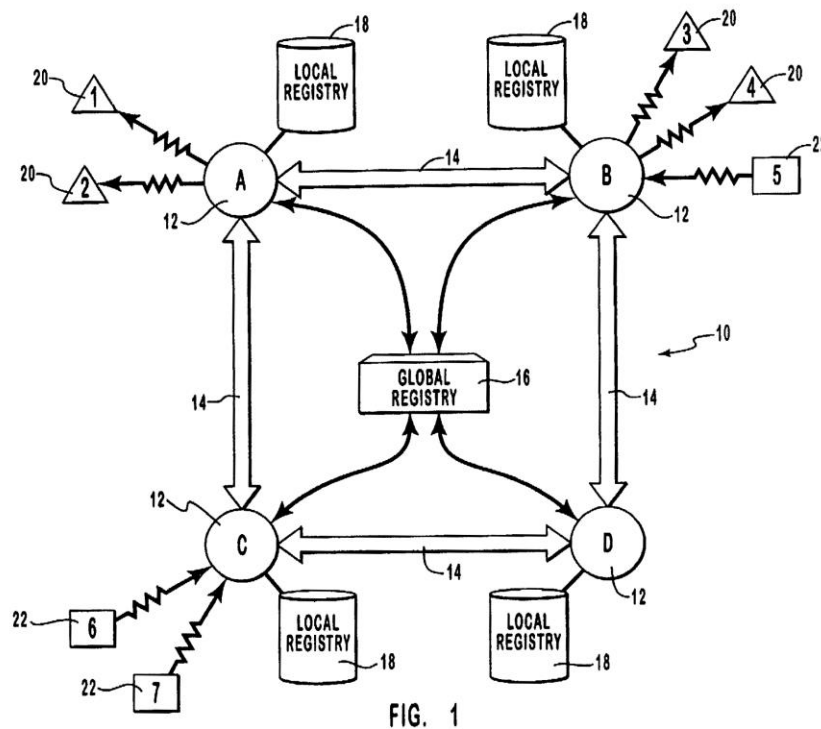
A. Overview

1. Lawson (EX1005)

Lawson, which was not considered during prosecution, discloses “[a] system [and] method [illustrated in Fig. 1 below] for global event notification in a distributed computer environment ... a local event registry [is used] to identify local event consumers that should be notified when an event occurs.... [A] global event registry [] identifies other servers which need notification when an event occurs. These other servers will then, in turn, notify their local event consumers of the event. The system and method ... incorporate[] multiple levels of filtering to allow [servers and] event consumers to remove notification of events having little or no interest.”

⁵ Petitioners do not concede no construction of the claims may ever be necessary or any other claim construction argument, and they reserve the right to propose constructions and respond to constructions proposed by others.

Abstract. The “filtering can be performed on any of the information available from a particular event ... include[ing] not only the information available in the event data itself, but also in the event packet, such as the [event type, event producer, server, etc. field in the] event packet.” 12:3-11, 20:10-13; *see also* 5:26-65, 16:63-17:6, 17:67-18:3, 22:9-11, Figs. 5, 6.

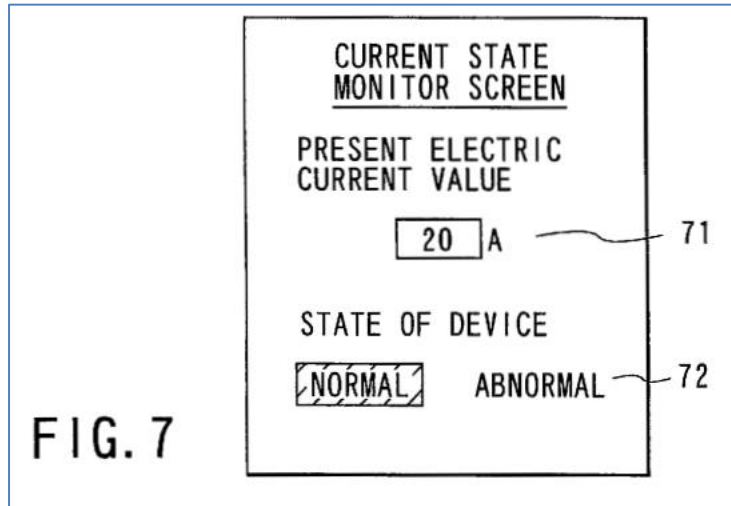


Example of Lawson’s Event Packet

Event Producer	Server	Type of Event	Time of Event	Size of Event Data	Event Data
----------------	--------	---------------	---------------	--------------------	------------

2. Tsutsumitake (EX1006)

Tsutsumitake, also not considered during prosecution, discloses “[a] real-time information transmission system [and method that] can immediately transmit



information updated on a server to a client” in a web-enabled environment. Abstract; 1:6-32. A web “client analyzes [a web]page. If an event request is included in the analyzed page ... [a] connection ID [is used by] the server [to] send[] an event to the client each time the event has occurred. The client processes the event and reflects the processed result on a screen of a display” to, e.g., update in real time powerplant data 71 and 72 being monitored via the web as shown in Fig. 7. Abstract, 13:31-65.

3. Motivation to Combine

The Federal Circuit has adopted a flexible approach when evaluating whether a patent claiming a combination of elements of prior art is obvious. *KSR International Co. v. Teleflex Inc.*, 550 U.S. 398 (2007). Under *KSR*, a POSA would have been motivated to combine Lawson with Tsutsumitake for numerous reasons. Lawson is a system for global event notification in a distributed computer environment. It is known that any event notification is a form of information

transmission. Tsutsumitake is a real-time information transmission system which describes information providing servers connected to the Internet—which, a POSA would have known, is the quintessential distributed computer system. *See Easttom Decl.* ¶¶58-60.

A POSA would have known that ensuring that events are transmitted in real-time using the combination of Lawson and Tsutsumitake would increase efficiency on the Internet, reduce bandwidth usage, and greatly expand the commercial value of the combined system and be in accord with design trends at the time, and thus, would have been motivated to combine Lawson with Tsutsumitake. *Easttom Decl.* ¶60. For example, Tsutsumitake discloses use of popular, known Internet programming methods with eventing systems like Lawson’s, which would make the system of Lawson combined with Tsutsumitake attractive to event producers and developers desiring to display updated information in real-time on webpages. *Id.*

A POSA also would have recognized much overlap and many similarities between Lawson and Tsutsumitake that would have facilitated combining them (e.g., same field, similar problems addressed, discloses known methods, similar components adapted for use in similar ways). *See Easttom Decl.* ¶61. Indeed, Lawson explicitly suggests combination with other systems: “The current system and method can be used with virtually any underlying event notification system. In one preferred embodiment, the present invention is designed to work in conjunction

with current event notification systems to achieve the desired functionality” (EX1005, 4:27-32), and Lawson is explicitly designed to be integrated with other systems: “the present invention, however, can also be used with any special purpose computer or other hardware system and all should be included within its scope.” (EX1005, 7:17-21). Thus, modifying Lawson with Tsutsumitake’s real-time Internet-based information transmission system would have been straightforward, have had numerous benefits, and would not have required extensive experimentation or imposed undue burden. *See Easttom Decl.* ¶62.

A POSA would also have a high expectation of success in the combination, because Lawson can be used with any event data, including transmitting events in real time over the Internet as disclosed by Tsutsumitake. For the same reasons, the combination would have yielded predictable results well within the skill of POSA. *See Easttom Decl.* ¶62.

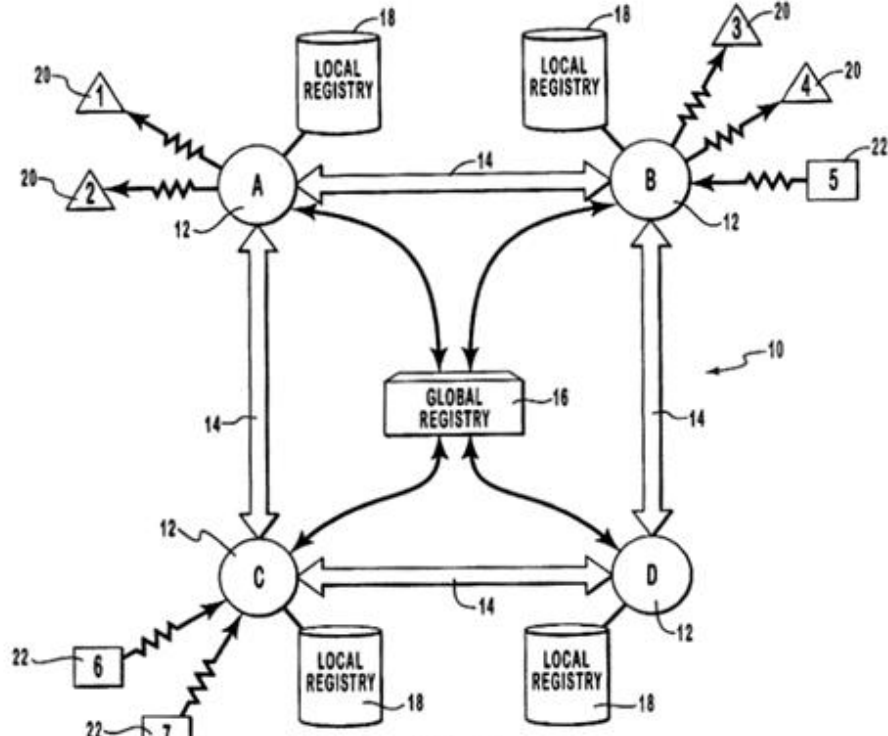
B. Detailed Analysis

1. Claims 14-37

Claims 14-37 recite four groupings in different statutory classes (e.g., method, apparatus, etc.), but are substantially identical within those groupings, so are discussed together. *See* EX1015 (claims 14-37 concordance).

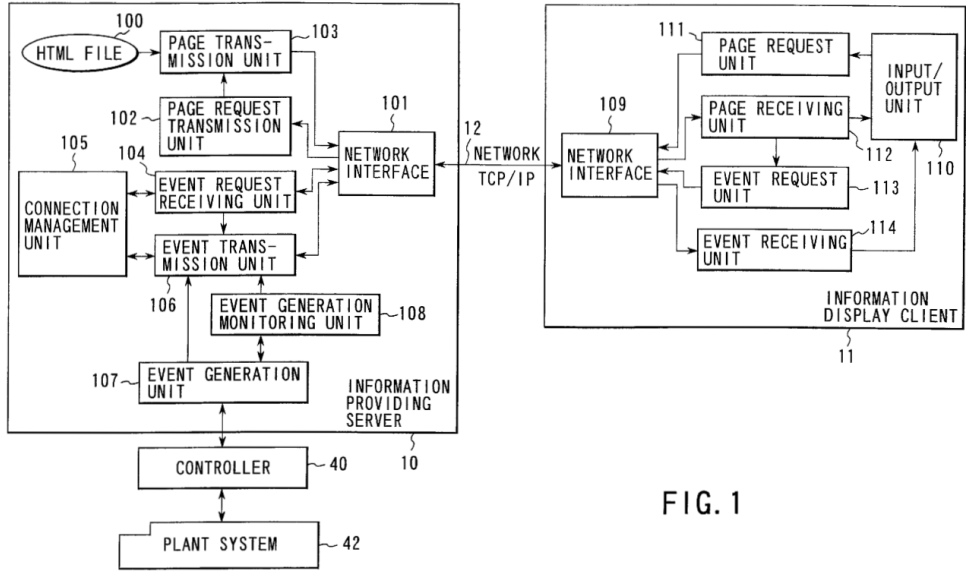
Claims	Lawson (EX1005) and Tsutsumitake (EX1006)
[14.0] A method comprising:	If the preamble is limiting, Lawson discloses “a system and method for globalizing event notifications in a distributed computing environment.” Abstract, 4:25-27. <i>See</i> Easttom Decl. ¶¶193-194.
[14.1] providing, using a processing device of an input source, a data representation to a client device, different from the input source, coupled to a routing network, wherein the data representation includes at least one live object recognizable by the client device, and	Lawson alone or combined with Tsutsumitake discloses providing, using a processing device of an input source (event producer, server), a data representation (data/records, webpage) to a client device (event consumer, client), different from the input source, coupled to a routing network (network of servers implementing event globalization process), wherein the data representation includes at least one live object recognizable by the client device (records/data recognized by event consumers, updateable webpage recognized by client). <u>Lawson</u> Lawson discloses that “[i]n Fig. 1, event consumers are illustrated by the triangular regions 20 and are numbered 1-4. Event producers are identified by rectangular regions 22 and are labeled 5-7. Event producer and event consumer are meant to be generic terms that are interpreted broadly. Event producer will be used to refer to a process, user, device, or other item that produces or generates an event. An event consumer

Claims	Lawson (EX1005) and Tsutsumitake (EX1006)
	should be broadly interpreted to refer to any process, user, device, or other item that desires notification of an event. As used throughout this application, event notification will refer to not only notification of an event but also distribution of information associated with that event. The details of how event information is distributed in conjunction with notification [] is described in greater detail below.” 9:1-15; <i>see also</i> 1:51-60. Lawson’s event producers and event consumers are different from each other and each can be or include general or special purpose computers and software. 7:17-39, Figs. 1-3; <i>see also</i> claims 38-39, 43-44. “Each server [labeled A, B, C, D in the network of in Fig. 1] has running thereon a local event globalization process that registers for events desired by local event consumers. When an event consumer registers for an event, the event globalization process of the event consumer’s local server places an entry into the local event registry for the local server. An entry is also placed into the global event registry for the local server where the event consumer is located. Thus, the global event registry is updated to contain an entry identifying the server where the event consumer is located and the desired event and the local event registry of that server is updated to identify the event consumer and the desired event.” 4:54-65, 7:52-8:4; <i>see also</i> 9:16-35, 10:16-35, Fig. 2 (noting any network can be used). Lawson “achieves global event notification by storing a global event registry comprising a list of events and a corresponding list of servers which need notification when the corresponding event occurs. In addition to the global registry, each server stores a local event registry comprising a list of events and a corresponding list of local event consumers that need notification when an event occurs.” 4:45-52; <i>see also</i> 24:17-19, Figs. 3-6.

Claims	Lawson (EX1005) and Tsutsumitake (EX1006)
	 FIG. 1 Lawson's "event packet" corresponds to the 722 patent's "live object recognizable by the client device" (i.e., event packet recognizable by Lawson's event consumers) because Lawson's event consumers use data/records in the event packet to update other data/records at the event consumers. Likewise, in the 722 patent "live objects" are "real-time dynamically-updateable objects" containing any data (e.g., "text characters and strings, images, frames, tables, audio, video, applets, scripts, ... variables and other

Claims	Lawson (EX1005) and Tsutsumitake (EX1006)												
	information used by applets, scripts, and/or code,” “[a]pplication and operating system constructs ... [such as] cells of spreadsheets, text in word processor documents, and title bars and messages displayed by the operating system or applications”) that “can be dynamically updated in real-time at the client” (e.g., updates to “stock prices and/or sports scores” or any other data at the client which “may be a simple numerical or textual value, e.g., ‘4’”). 722 patent 5:27-29, 6:63-7:12, 8:57-63, 9:53-56. Lawson discloses “event consumer 4 [in Fig. 1] is a company phone directory” and “event consumer 1 [in Fig. 1] is a payroll database.” 23:64-65, 24:1. These event consumers dynamically update data/records they store based upon receipt and processing of an event packet (e.g., update phone number, update payroll database based on data/records in the event packet). 21:16-25, 24:64-24:6, Fig. 7 #178 (phone directory), 24:1-4, 25:16-21, Fig. 7 #186 (payroll database). Lawson illustrates these two examples via Fig. 7, which is “an example of how [Lawson’s system and method] works ... [where t]he network topology used for th[e] example is presented in Fig. 1” and the “event globalization process is described in Figs. 3-6.” 23:60-63, 24:17-18. With respect to the “event packet,” Lawson discloses that “[f]or most events, packaging event data into an event packet has the benefit of adding data that may be useful in processing the event. In one embodiment, an event packet comprises: <table><tr><th>Event Producer</th><th>Server</th><th>Type of Event</th><th>Time of Event</th><th>Size of Event Data</th><th>Event Data</th></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	Event Producer	Server	Type of Event	Time of Event	Size of Event Data	Event Data						
Event Producer	Server	Type of Event	Time of Event	Size of Event Data	Event Data								

Claims	Lawson (EX1005) and Tsutsumitake (EX1006)
	[T]he event producer field is an identifier that identifies the process, device, user, etc., which generated the event. The server field identifies the server location where the event was produced. The type of event field is an identifier describing the type of event which occurred. This [event type] identifier preferably identifies an event in the global event registry and/or local event registry so that appropriate servers or event consumers can be identified to receive the event... The event data is the data associated with the event.” 12:3-26. The “event data” in Lawson can be any data or records, such as employee’s name, phone number, address, payroll data, program function call, computer program code, etc. 6:4-15, 7:23-51, 21:15-26, 24:1-4. <u>Tsutsumitake</u> Tsutsumitake discloses “a real-time information transmission system for transmitting information in real time from a server to a client via a network.” 3:39-41. As shown in Fig. 1, “a [electric] plant system 42 is connected to the [information providing] server 10 ... A [web]page request receiving unit 102 [on server 10] receives a [web]page request sent from the client 11 via the network interface 101, analyzes it, and informs a [web]page transmission unit 103 of a [web]page to be sent to the client 11... The page transmission unit 103 transmits an HTML file stored in the server 10 to the network 12 via the network interface 101 [to client 11]. Information to be transmitted is not limited to the HTML file (100) [webpage], and may be freely chosen one such as image or voice.” 8:39-49; <i>see also</i> 3:39-56. “In this structure, the client [11], which received the requested [web]page information from the server, can automatically issue the event request to the server according to the event request information set in the [web]page information, and can activate the event notification. Specifically, by associating the [web]page information with the event, the [web]page information opened on the client

Claims	Lawson (EX1005) and Tsutsumitake (EX1006)
	side can be dynamically changed according to the event notification sent from the server.” 4:22-29; <i>see also</i> 1:7-43, 7:63-8:23, 9:43-68, 13:32-14:3, Fig. 7.  The diagram illustrates the architecture of an information providing system. On the left, the INFORMATION PROVIDING SERVER (10) includes a CONTROLLER (40) connected to a PLANT SYSTEM (42). The server's internal components are: EVENT GENERATION UNIT (107) connected to EVENT GENERATION MONITORING UNIT (108); EVENT TRANSMISSION UNIT (106) connected to EVENT REQUEST RECEIVING UNIT (104); CONNECTION MANAGEMENT UNIT (105) connected to EVENT REQUEST RECEIVING UNIT (104); PAGE REQUEST TRANSMISSION UNIT (102) connected to EVENT REQUEST RECEIVING UNIT (104); PAGE TRANSMISSION UNIT (103) connected to PAGE REQUEST TRANSMISSION UNIT (102); and HTML FILE (100) connected to PAGE TRANSMISSION UNIT (103). A NETWORK INTERFACE (101) connects the server to a NETWORK TCP/IP (12). On the right, the INFORMATION DISPLAY CLIENT (11) includes an INPUT/OUTPUT UNIT (110) connected to PAGE REQUEST UNIT (111), PAGE RECEIVING UNIT (112), EVENT REQUEST UNIT (113), and EVENT RECEIVING UNIT (114). A NETWORK INTERFACE (109) connects the client to the NETWORK TCP/IP (12). FIG. 1

Claims	Lawson (EX1005) and Tsutsumitake (EX1006)
	Fig. 2 shows an example of the webpage sent by server 10 to client 11. “In this example, an event request to the server [from the client] is included in the [web]page. The event request is expressed by an attribute ‘URL’ in a tag ‘EVENT’ in a tag ‘EMBED’.” 10:20-23. The client 11 recognizes the webpage as containing the 722 patent’s “live object” because upon receiving the webpage, “the page receiving unit 112 [in client 11] informs an event request unit 113 of the URL of the event request.” 9:61-68. <i>See also</i> 14:3-29, Fig. 8. <pre> < HTML > < HEAD > < TITLE > SAMPLE PAGE < /TITLE > < /HEAD > < BODY > < H1 > SAMPLE PAGE < /H1 > THIS PAGE INCLUDES EVENT REQUEST < EMBED > < EVENT URL=http://abc.def/hij.evt > < /EMBED > < /BODY > < /HTML > </pre> FIG. 2 <i>See Easttom Decl.</i> ¶¶195-204.
[14.2] causing the client device to respond to the live object of the data representation by determining an object identifier (ID) of the live object and to register for updates of the live object	Lawson alone or combined with Tsutsumitake discloses causing the client device (event consumer/client) to respond to the live object of the data representation (records/data, webpage) by determining an object identifier (ID) (event type, event ID/URL) of the live object and to register for updates of the live object (event consumer/client registers for events) with the routing network (network of servers implementing event globalization process), such that registering the client device with the routing network provides client connection information to a node in the routing network (updating a

Claims	Lawson (EX1005) and Tsutsumitake (EX1006)
with the routing network, such that registering the client device with the routing network provides client connection information to a node in the routing network; and	server's local event registry to identify event consumer, updating connection management unit to identify client). <u>Lawson</u> Lawson discloses “[w]hen an event consumer receives notification that a certain event happened on an event producer, the event consumer can take appropriate action” and that “event consumers which desire to receive notification of certain events ‘register’ with the event producing system.” 1:61-63, 2:6-9. “When an event consumer needs to register for a new event or create a new event, the event consumer can send a ‘register for event’ packet or a ‘create event packet’ to the event globalization process on the local server. When an event is created, the event globalization process will create an entry in the global event registry, as for example global event registry 68 of Fig. 3. This entry will include the event identifier of the new event. When an event consumer registers for an event, the event globalization process will place an entry in the local event registry, as for example local event registry 72 of Fig. 3, identifying the event and the event consumer wishing to receive notification of the event.” 15:41-56; <i>see also</i> 10:50-11:2, 16:41-46, 19:63-20:7, 22:27-23:2, 23:11-28, Fig. 3 #52, 76, Fig. 4 #102, claim 9. For example, “the employee phone book database could register for notification of ‘a modified object’ type event.” 21:20-21. As another example, “a payroll database [could] also register[] for the add user event so that information can be extracted to update the employee payroll database.” 24:1-4. <i>See also</i> 10:20-35, 10:47-67, 12:46-52, 14:54-64, 15:35-57, 16:23-62, Figs. 1-4 (additional examples of event consumers registering for event types with the server network event globalization process).

Claims	Lawson (EX1005) and Tsutsumitake (EX1006)
	Lawson’s “event type” corresponds to the 722 patent’s “object identifier (ID) of the live object” because the 722 patent explains an “objectID [] is preferably a string. The string can encode any information ... [and can be] a simple textual and/or numeric identifier... For example, the object ID for a live object representing the real time price of a stock having the symbol ‘BANG’ might be[gin with] ‘Bang\$...’” and contain additional identifying data (e.g., “Bang\$homeScoreID” as the object ID for a live object that updates a sports score). 722 patent 8:34-37, 8:63-9:2, 14:14-22. In the 722 patent, an “update message will change the specified properties of all of the objects having the given object ID” (i.e., the object ID identifies the updated data/records from the input source to be provided to the client). Likewise, in Lawson “[t]he type of event field [(event type)] ... identifies an event in the global event registry and/or local event registry so that appropriate servers or event consumers can be identified to receive the event.” 12:15-20. “[T]he only required field [in Lawson] is the type of event field. The type of event field is used to extract from the global event registry, either directly or indirectly, those servers that should receive notification of the event and is used to extract from the local event registry, either directly or indirectly, those local event consumers that should receive notification of the event.” 12:28-34. The “registry compris[es] a list of local event consumers and associated list of event types [] identify which local event consumers are registered for events of the type included in [the] event packet.” 27:61-64 (claim 19). “In addition to the type of event field, the event data should also be included so that the event information can be transferred to the appropriate recipients” (e.g., in the phone directory example, ‘a modified object’ event type and an updated phone number in the event data; in the payroll example, an “add user” event type and information in the event data that “can be extracted to update the employee payroll database”). 12:34-37, 21:15-26, 24:1-4.

Claims	Lawson (EX1005) and Tsutsumitake (EX1006)
	<u>Tsutsumitake</u> Tsutsumitake discloses “[t]he page receiving unit 112 [on the client 11] analyzes the content of the [web]page sent from the server 10 (step A6) to determine whether an event request is included in the page (step A7).” 10:56-59. “[I]f the event request is included, the page receiving unit 112 informs the event request unit 113 of the URL of the event and causes the event request unit 113 to send the event request to the server 10 (step A8).” 10:64-67, Figs. 1, 3. At the server 10, “[t]he event request receiving unit 104 asks the connection management unit 105 whether or not an identifier (event ID) of the event requested by the event request and an identifier (client ID) of the client 11 at the request originating point have already been registered (steps B8, B9)... If they are not registered, a pair of the ... event ID ... and ... client ID ... are registered in the connection management unit 105 (step B11).” 11:45-55, Figs. 1, 4; <i>see also</i> 9:30-33, 12:66-67. <i>See Easttom Decl.</i> ¶¶205-211.
[14.3] sending, using the processing device of the input source, an update message to the routing network, wherein the update message identifies the live object and contains update data	Lawson alone or combined with Tsutsumitake discloses sending, using the processing device of the input source (event producer/client), an update message (event) to the routing network (network of servers implementing event globalization process), wherein the update message identifies the live object (contains field “event type,” contains event ID/responsive event URL) and contains update data (updated event data) that updates a property of the live object (updates records/data at the event consumer/client). <u>Lawson</u>

Claims	Lawson (EX1005) and Tsutsumitake (EX1006)
that updates a property of the live object,	Lawson discloses “an event producer notifies event consumers of events.” 2:4-5. “[E]vent notification [] refer[s] to not only notification of an event but also distribution of information associated with that event.” 9:9-11. Referring to Fig. 7 (which uses the same network topology as Fig. 1), “assume for this example that event producer 5 [at server B] added a new user to a database stored on the network” and “that event consumer 4 [at server B] is a company phone directory that has registered to receive an ‘add user’ [or ‘modified object’] event so that it can extract the phone number of the added user and place [or update] it in the company phone book. Further, assume that event consumer 1 [at server A] is a payroll database that has also registered for the add user event so that information can be extracted to update the employee payroll database.” 21:16-26, 23:60-24:4. “After the user has been added to the database, step 156 indicates that an event packet is created [by event producer 5 at server B].” 24:7-9. Event packets include, among other things, an “event type” identifying the “appropriate servers or event consumers [] to receive the event” and “event data” comprising any information, data, or records “associated with the event.” 12:7-10, 12:17-20, 12:25-26. “After the event packet has been created, the event is sent to the local event globalization process.” 24:15-16. As part of the event globalization process, “the registered local event consumers and registered remote servers [for that event type] are identified in steps 166 and 168 [of Fig. 7].” 24:35-37. <i>See also</i> 4:66-5:12, 10:16-11:2 (additional examples of event producers sending events with updated data to event consumers). “A single event may need to be transferred to multiple event consumers” via the network. 22:62-63. In Lawson’s phone directory example, “the event notification process on server B dispatches the event packets [over the network] to registered local event consumers [for

Claims	Lawson (EX1005) and Tsutsumitake (EX1006)
	the event type as] indicated in Fig. 7 by step 174” and “[a]s indicated by step 176 [in Fig. 7], the event packet is received by the company phone directory [event consumer 4 in Fig. 1].” 24:60-66. “After the event has been transferred, the company phone directory creates a new record[/updates a record] in the company phone book as indicated by step 178.” 25:4-6. In Lawson’s payroll example, “step 170 [on server B in Fig. 7] dispatches the events to the event globalization process running on server A [where event consumer 1 is located as shown in Fig. 1].” 24:44-47. “[S]erver A proceeds to identify the registered local event consumers for this event [type as] indicated in Fig. 7 by step 180 ... by checking the local event registry on server A. In this example, the only registered local event consumer on server A is event consumer 1. In step 182, the event globalization process of server A transfers the event packet [over the network] to the local event consumer which, in this case, is the payroll database process. The payroll database process receives the information as indicated by step 184 and proceeds to create a new entry[/update an entry] in the employee payroll database as indicated by step 186.” 25:8-20. <u>Tsutsumitake</u> Tsutsumitake discloses “[i]f an event has occurred, the server 10 sends it to the client 11 with use of [a] connection ID. Subsequently, as shown in Fig. 6, each time an event has occurred, it is sent [to client 11] repeatedly with use of the [connection] ID.” 13:1-5; <i>see also</i> 12:10-15, Figs. 1, 5. The example in Fig. 7 discloses “the event [data] indicat[es] the present value of current and [] the state of the device being transmitted in real time [from electric plant 42]) sent from the server 10 in response to one event

Claims	Lawson (EX1005) and Tsutsumitake (EX1006)
	request issued [by client 11] in accordance with the displayed page ... [shown in Fig. 7 #71, 72].” 13:59-66. “[E]ach time an event has occurred in the server the event notification has to be sent to the [registered] client[s].” 4:36-38; <i>see also</i> 12:23-28. <i>See Easttom Decl.</i> ¶¶212-216.
[14.4] wherein a gateway device at the routing network is configured to identify a category of the update message based on the input source, to determine a node type to which the identified category maps, and to route the update message to the node, having the node type, at the routing network,	Lawson discloses wherein a gateway device (remote event processing block) at the routing network (network of servers implementing event globalization process) is configured to identify a category of the update message based on the input source (identify servers to receive events by filtering, e.g., based on the event producer or server field of event packet), to determine a node type to which the identified category maps (servers needing event), and to route the update message to the node, having the node type, at the routing network (transfer event only to servers needing event). Lawson’s “remote event processing block 66[, as shown in Figs. 3 and 5,] monitors event queue 64 and when events are placed therein by incoming event processing block 60 that need to be transferred to other servers in the network, remote event processing block 66 will establish a connection to those servers and transfer the events to the other server. Embodiments within the scope of the present invention may, therefore, comprise means for accessing a global event registry to identify which servers should receive which events [(e.g., filtering based on event type, event producer, or server)].... In such a case, remote event processing block 66 identifies the servers that should receive events queued in event queue 64 by checking global registry 68.” 13:12-24; <i>see also</i> 12:12-20, 12:29-34. “[W]hen an event is transferred from one server to another server, a connection must be established between the two servers.” 13:25-27. “[T]here is a possibility of events being transferred both to and from the local server. This is

Claims	Lawson (EX1005) and Tsutsumitake (EX1006)
	indicated in Fig. 3 by remote event 56 and remote event 58.” 13:33-35. “Remote event 58 represents a local event that is transferred to another server via network connection 52. [I]n Fig. 3 network connection 52 represents an example of the networking means for interconnect servers in the present invention.” 11:51-56. Lawson discloses “filtering can be performed on any of the information available from a particular event. The available information for an event includes not only the information available in the event data itself, but also in the event packet, such as the [event type, event producer, server, etc. field in the] event packet previously described.” 20:10-13; <i>see also</i> 5:26-65, 16:63-17:6, 17:67-18:3, 22:9-11, Fig. 5. For example, as shown in Fig. 2, “[w]hen server B recognized that event producer 42 [(e.g., a specific publisher, producer)] had initiated an event, server B would check global event registry 34. Global event registry 34 would identify that server E needed notification of the event. Server B would then send the event through gateway 32 to server E.” 10:23-28. The “event notification was transferred from server B to server E without broadcasting the event to servers A, C and D. Thus, such a method can globalize an event to required locations while, simultaneously, reducing message traffic due to broadcast type events.” 10:43-47; <i>see also</i> 22:62-63. <i>See Easttom Decl.</i> ¶¶217-223.
[14.5] wherein the node is configured to identify the client device as a registered device and to	Lawson alone or combined with Tsutsumitake discloses wherein the node (server) is configured to identify the client device as a registered device (identify local event consumer/client registered to receive event) and to route the update message to the client device (transfer event to event consumer/client), and

Claims	Lawson (EX1005) and Tsutsumitake (EX1006)
route the update message to the client device, and	<u>Lawson</u> Lawson’s “local event processing block 70[, as shown in Figs. 3 and 6,] monitors event queue 64 for events that should be transferred to a local event consumer. Embodiments of the present invention may, therefore, comprise means for accessing a local event registry to identify which local event consumers should receive which events. Such means may be incorporated into local event processing block 70. In such a case, local event consumers that should receive events queued in event queue 64 are identified by checking local event registry 72 and any associated filtering criteria... In Fig. 3, the process of transferring an event to a local event consumer is illustrated by event 76. Event 76 is shown going out over network connection 52.” 13:43-52, 13:65-67, Fig. 3 #52, 76. Similarly, Fig. 7 at steps 174 and 182 shows events routed from a server to an event consumer. 24:60-64, 25:16-18. <i>See also</i> 4:66-5:12, 8:58-60, 8:66-67, 10:28-36, 12:15-20, 12:29-34, 13:67-14:12, 19:63-20:9, 22:27-23:2, 25:34-36, 25:31-45, claim 9 (examples of servers routing events to event consumers). In Lawson’s phone directory example, “the event notification process on server B dispatches the event packets [over the network] to registered local event consumers [for the event type as] indicated in Fig. 7 by step 174” and “[a]s indicated by step 176 [in Fig. 7], the event packet is received by the company phone directory [event consumer 4 in Fig. 1].” 24:60-66 In Lawson’s payroll example, “step 170 [on server B in Fig. 7] dispatches the events to the event globalization process running on server A [where event consumer 1 is located as shown in Fig. 1].” 24:44-47. “[S]erver A proceeds to identify the registered local event consumers for this event [type as] indicated in Fig. 7 by step 180 ... by checking

Claims	Lawson (EX1005) and Tsutsumitake (EX1006)
	the local event registry on server A. In this example, the only registered local event consumer on server A is event consumer 1. In step 182, the event globalization process of server A transfers the event packet [over the network] to the local event consumer which, in this case, is the payroll database process.” 25:8-17. <u>Tsutsumitake</u> Tsutsumitake discloses “each time an event has occurred in the server the event notification has to be sent to the [registered] client[s].” 4:36-38; <i>see also</i> 13:1-5, 12:23-28, 13:59-66, Figs. 6, 7. <i>See Easttom Decl.</i> ¶¶224-228.
[14.6] wherein the client device processes the update message upon receipt to update the property of the live object at the client device.	Lawson alone or combined with Tsutsumitake discloses wherein the client device (event consumer/client) processes the update message upon receipt to update the property of the live object at the client device (update records/data recognized by event consumer/client). <u>Lawson</u> Lawson discloses “that event consumer 4 [in Fig. 1] is a company phone directory.” 23:64-65. “As indicated by step 176 [of Fig. 7], the event packet is received by the company phone directory” and “[a]fter the event has been transferred, the company phone directory process creates a new record[/updates a record] in the company phone book as indicated by step 178.” 24:64-66, 25:4-6; <i>see also</i> 21:16-19. Lawson also discloses that “[i]n step 182 [of Fig. 7], the event globalization process of server A transfers the event packet to the local event consumer which, in this case, is

Claims	Lawson (EX1005) and Tsutsumitake (EX1006)
	the payroll database process. The payroll database process receives the information as indicated by step 184 and proceeds to create a new entry[/update an entry] in the employee payroll database as indicated by step 186.” 25:16-21; <i>see also</i> 24:1-4. Lawson’s updating of data/records as described above corresponds to the 722 patent’s “update the property of the live object at the client device” because the 722 patent explains “[l]ive objects have associated ‘properties’ which include any modifiable data related to the object or referenced with respect to the object.” 722 patent 7:22-24. <u>Tsutsumitake</u> Lawson discloses “generated events [from server 10] are transmitted in real time to the client 11. The browser (client 11) processes a plurality of events (both the event [data] indicating the present value of current and [] the state of the device being transmitted in real time [from electric plant 42]) sent from the server 10 in response to one event request issued in accordance with the displayed page, and reflects on the screen the event processing results on the current value display section 71 or device state display section 72” shown in Fig. 7. 13:59-66. FIG. 7

Claims	Lawson (EX1005) and Tsutsumitake (EX1006)
	See Easttom Decl. ¶¶229-234.
[15.0] The method of claim 14, wherein providing the data representation to the client device includes providing the live object that causes the client device to register with a client proxy of the routing network.	Lawson discloses this claim. See G1-[14]. In addition: Lawson discloses that “[w]hen a local event globalization process receives notification of an event that has occurred locally, the local process looks in the local registry in order to identify local event consumers that need notification of the event. This process then transfers the event to any identified local event consumers. In addition, the process checks the global event registry to identify any additional servers which also need notification of the event. This process then sends the event to the corresponding event globalization process running on the identified servers. These corresponding event globalization processes, in turn, check their local event registries and provide the event to any event consumers identified therein. In this way, local events can be routed to any event consumer in the network without the need to register at each server.” 4:66-5:12; <i>see also</i> Figs. 1-2, 5. This corresponds to the 722 patent’s “client proxy,” which “merely passes messages among the clients and nodes” and/or “stores client registration information ... [to] determine which client proxies have registered for the messages[] and [to] send the messages to the appropriate client proxies” to thereafter send to registered clients. 3:60-4:2. See Easttom Decl. ¶¶235-240.
[16.0] The method of claim 14, wherein	Lawson alone or combined with Tsutsumitake discloses this claim. See G1-[14]. In addition:

Claims	Lawson (EX1005) and Tsutsumitake (EX1006)
providing the data representation to the client device includes providing the live object that causes the client device to register with the node of the routing network.	<u>Lawson</u> Lawson discloses “[w]hen an event consumer receives notification that a certain event happened on an event producer, the event consumer can take appropriate action,” such as the event consumer causing “an associated process [] to be initiated which receives the custom event and takes appropriate action.” 1:61-63, 6:2-4. “[C]omputers having no local program storage means and that receive program code means across a communication link should also be included within the scope of [Lawson’s disclosure].” 7:29-32. As previously discussed, the data/records Lawson provides to event consumers via events and event packets can include any data, including computer program code to register the event consumers with the Lawson’s network of servers implementing event globalization process, such as via Lawson’s “launch program event type.” 6:7. “Program code means comprises, for example, executable instructions and data which cause a general purpose computer or special purpose computer to perform a certain function or a group of functions.” 7:48-51. One such function in Lawson is “the event consumer [] send[ing] a ‘register for event’ packet ... to the event globalization process on the local server” to register the event consumer for the event. 15:41-44. <u>Tsutsumitake</u> Tsutsumitake discloses “[t]he page transmission unit 103 [in server 10] transmits an HTML file stored in the server 10 to the network 12 via the network interface 101 [to client 11].” 8:39-49; <i>see also</i> 3:39-56. “In this structure, the client [11], which received the requested [web]page information from the server, can automatically issue the event request to the server according to the event request information set in the [web]page

Claims	Lawson (EX1005) and Tsutsumitake (EX1006)
	information, and can activate the event notification. Specifically, by associating the [web]page information with the event, the [web]page information opened on the client side can be dynamically changed according to the event notification sent from the server.” 4:22-29; <i>see also</i> 1:7-43, 7:63-8:23, 9:43-68, 13:32-14:3, Fig. 7. Upon receipt of the webpage at client 11, “[t]he page receiving unit 112 [on the client 11] analyzes the content of the [web]page sent from the server 10 (step A6) to determine whether an event request is included in the page (step A7).” 10:56-59. “[I]f the event request is included, the page receiving unit 112 informs the event request unit 113 of the URL of the event and causes the event request unit 113 to send the event request to the server 10 (step A8).” 10:64-67, Figs. 1, 3. At the server 10, “[t]he event request receiving unit 104 asks the connection management unit 105 whether or not an identifier (event ID) of the event requested by the event request and an identifier (client ID) of the client 11 at the request originating point have already been registered (steps B8, B9) If they are not registered, a pair of the identifier (event ID) of the event requested by the event request and the identifier (client ID) of the client 11 at the request originating point are registered in the connection management unit 105 (step B11).” 11:45-55, Figs. 1, 4; <i>see also</i> 9:30-33, 12:66-67. <i>See Easttom Decl.</i> ¶¶241-246.
[17.0] The method of claim 14, wherein providing the data representation to the client device includes	Lawson combined with Tsutsumitake discloses this claim. <i>See</i> G1-[14]. In addition: <u>Lawson</u> Lawson discloses “[w]hen an event consumer receives notification that a certain event happened on an event producer, the event consumer can take appropriate action,” such

Claims	Lawson (EX1005) and Tsutsumitake (EX1006)
providing an activation module that is executed by the client device and that registers the live object with the routing network.	as the event consumer causing “an associated process [] to be initiated which receives the custom event and takes appropriate action.” 1:61-63, 6:2-4. One such function in Lawson may execute is “the event consumer [] send[ing] a ‘register for event’ packet ... to the event globalization process on the local server” to register the event consumer for the event. 15:41-44; <i>see also</i> 7:48-51. <u>Tsutsumitake</u> Tsutsumitake discloses that in the webpage the server sends to client containing the event request “[i]t is also possible to adopt a method wherein the event request [to register the client 11 with the server 10] is generated dynamically from a certain program incorporated in the page (e.g. ‘Applet’ of Java developed by Sun Microsystems, or “JavaScript” developed by Netscape).” 10:20-33. This Java/JavaScript code, for example, parses the webpage to identify “an attribute ‘URL’ in a tag ‘EVENT’ in a tag ‘EMBED’” of the webpage to send to the server to register the client for the event. <i>See</i> 10:20-23; <i>see also</i> 10:64-67, 11:45-55, Fig. 4. This corresponds to the “activation module 124” of the 722 patent, which “preferably contains JAVA® instructions ... [to] register[] object IDs from the web page 118 downloaded by the client 114 with the routing network 110” by, for example, “pars[ing] 610 the web page 118 received from the server 112 and identif[ing] the object IDs of the live objects ... to register [] by communicating to the routing network 116.” 722 patent 11:1-14, 14:33-35, 14:55-59, Fig. 6. <i>See Easttom Decl.</i> ¶¶247-250.
[18.0] The method of claim 17, wherein	Lawson combined with Tsutsumitake discloses this claim. <i>See</i> G1-[17]. In addition:

Claims	Lawson (EX1005) and Tsutsumitake (EX1006)
providing the activation module includes providing the activation module that is configured to determine a node type that handles registration and that causes the client device to register with the node corresponding to the node type.	<u>Lawson</u> Lawson discloses “[w]hen an event consumer receives notification that a certain event happened on an event producer, the event consumer can take appropriate action,” such as the event consumer causing “an associated process [] to be initiated which receives the custom event and takes appropriate action.” 1:61-63, 6:2-4. One such function in Lawson may execute is “the event consumer [] send[ing] a ‘register for event’ packet ... to the event globalization process on the local server” to register the event consumer for the event. 15:41-44; <i>see also</i> 7:48-51. “Registering” for the event is a prerequisite for receiving “notification of events that occur.” <i>See</i> 4:33-37. Because of Lawson’s local and global event registries, “local events [(e.g., an event at a given event producer)] can be routed to any event consumer in the network without the need [for the event consumer] to register at each server.” 5:11-12; <i>see also</i> 4:66-5:10. As such, Lawson discloses the event consumer is configured to determine a server that handles registration and to cause the event consumer to register with that server, which results in the event consumer receiving the events it is registered for. <u>Tsutsumitake</u> Tsutsumitake discloses “the client [11], which received the requested [web]page information from the server, can automatically issue the event request to the server according to the event request information set in the [web]page information,” including via Java/JavaScript code. 4:22-25, 10:64-67, 11:45-55, Fig. 4. Tsutsumitake determines the server to receive the event registration via “[t]he page receiving unit 112 [on the client 11, which] analyzes the content of the [web]page sent from the server 10 (step A6) to determine whether an event request is included in the page (step A7).” 10:56-59. “[I]f the event request is included, the page receiving unit 112 informs the event

Claims	Lawson (EX1005) and Tsutsumitake (EX1006)
	request unit 113 of the URL of the event and causes the event request unit 113 to send the event request to the server 10 (step A8) [for registration to receive events].” 10:64-67, Figs. 1, 3; <i>see also</i> 11:45-55, Fig. 4. As such, Tsutsumitake discloses Java/JavaScript code at the client is configured to determine a server that handles registration and cause the client to register with that server to receive events. Lawson’s and Tsutsumitake’s disclosures correspond to the 722 patent’s recited “activation module that is configured to determine a node type that handles registration and that causes the client device to register with the node corresponding to the node type,” which the 722 patent explains is “functionality [on the client] for determining which node type handles which registration requests. 722 patent 19:61-63. One way the 722 patent’s “activation module” does so is by “determine[ing] which categories of messages it needs and to which nodes types it should connect [to get them]. The activation module 124 then makes a connection to one node of each relevant type and registers the appropriate object IDs with the appropriate node.” 21:47-52. <i>See Easttom Decl.</i> ¶¶251-254.
[19.0] The method of claim 17, wherein providing the activation module includes providing the activation module that is configured to determine a message category of	Lawson combined with Tsutsumitake discloses this claim. <i>See</i> G1-[17]. In addition: Lawson and Tsutsumitake disclose program or Java/JavaScript code at the event consumer/client to determine the producer or server an event came from and to register for those events. <i>See</i> G1-[18]. The 722 patent explains that “[o]ne way [to assign messages into categories] is to assign all messages from a given input source 710 [(e.g, a publisher, server)] into a certain category.” 18:52-54; EX1002, 1014.

Claims	Lawson (EX1005) and Tsutsumitake (EX1006)
the data representation and that causes the client device to register with a node having a node type corresponding to the message category.	

Claims 20-37 are obvious for the reasons stated in the corresponding limitations of claims 14-19, which are identified in the table below. *See also* EX1015 (claims 14-37 concordance).

Apparatus	Article	Software	Corresponding G1-Argument	Easttom Decl. ¶
[20.0]	[26.0]	[32.0]	[14.0]	262-64, 317-18, 385
[20.1]	[26.1]	[32.1]	[14.1]	265-69, 319-22, 386-88
[20.2]	[26.2]	[32.2]	[14.2]	270-74, 323-27, 328-27, 389-92
[20.3]	[26.3]	[32.3]	[14.3]	275-78, 328-33, 393-99
[20.4]	[26.4]	[32.4]	[14.4]	279-83, 334-38, 400-04
[20.5]	[26.5]	[32.5]	[14.5]	284-86, 339-41, 405-07
[20.6]	[26.6]	[32.6]	[14.6]	287-92, 342-47, 408-13
[21.0]	[27.0]	[33.0]	[15.0]	293-97, 348-57, 414-19
[22.0]	[28.0]	[34.0]	[16.0]	298-302, 358-63, 420-25
[23.0]	[29.0]	[35.0]	[17.0]	303-05, 364-69, 427-29
[24.0]	[30.0]	[36.0]	[18.0]	305-07, 370-75, 430-33
[25.0]	[31.0]	[37.0]	[19.0]	308-15, 376-84, 434-38

2. Claims 1 and 7

Except as discussed below, claims 1 and 7 are substantially identical and obvious for the reasons stated in the corresponding limitations of claim 14 and the

Easttom Declaration, which are identified in the tables below. *See also* EX1016 (claims 1-13 concordance).

Claim 1	Corresponding G1 Argument	Easttom Decl. ¶
[1.0]	[14.0]	64-65
[1.1]	[14.3]	66-71
[1.2]	[14.4]	72-73
* [1.3]	[14.4]	74-78
[1.4]	[14.4]	79-83
[1.5]	[14.1] , [14.2] , [14.5]	84-88
[1.6]	[14.5]	89-90
[1.7]	[14.6]	91-94

Claim 7	Corresponding G1 Argument	Easttom Decl. ¶
[7.0]	[14.0]	131-36
[7.1]	[14.3] , [14.4]	137-38
[7.2]	[14.3] , [14.4]	139-41
[7.3]	[14.3] , [14.4]	142-43
* [7.4]	[14.3] , [14.4]	144-48
[7.5]	[14.2] , [14.5] , [14.6]	149-50
* [7.6]	[14.2] , [14.5] , [14.6]	151-53
[7.7]	[14.2] , [14.5] , [14.6]	154-58
[7.8]	[14.2] , [14.5] , [14.6]	159-61

In addition, Lawson discloses the “mapping” to “node type” and for “controlling ... message traffic” elements in claims 1.3, 7.4, and 7.6.

Lawson's Fig. 2 example discloses an event producer on server B initiated an event, server B checked the global event registry to identify that server E needed the event, and server B sent the event only to server E, which Lawson discloses "reduc[es] message traffic." 10:43-47. Lawson also discloses "filtering events so that an event is only transferred if the event meets designated filtering criteria." 20:10-13; *see also* 17:67-18:3, Fig. 5; Easttom Decl. ¶¶74-78, 144-48, 151-53.

3. Claim 2 and 8

Claims 2 and 8 are obvious for the reasons stated in [G1-\[14.2\]](#) and Easttom Decl. ¶¶97-101, 163-67. In addition, Lawson alone or combined with Tsutsumitake disclose the underlined "determine if the client has registered for updates"/"a connection to the client" based on the "objectID" elements in claims 2 and 8.

Lawson discloses when "the local event consumer that is attempting to register for the event ... [a] step is to check and see if the event consumer has already registered for this event. This is performed by decision block 100 [in Fig. 4]. If the consumer is already registered for the designated event, execution returns to the start. If the event consumer has not already registered for the event, then entries are made into the local and global event registries in order to achieve registration for this event by the event consumer [at block 102 of Fig. 4]." 16:35-44. This registration "identif[ies] the event and the event consumer wishing to receive notification of the

event” via the network. 15:48-56; *see also* 4:61-5:12, 10:16-35, 24:35-44, 24:59-62, 25:8-11, Figs. 5, 7.

Tsutsumitake discloses “[t]he event request receiving unit 104 asks the connection management unit 105 whether or not an ... event ID ... and ... client ID ... have already been registered (steps B8, B9). If they are not registered, [the] ... event ID ... and ... client ID ... are registered in the connection management unit 105 (step B11).” 11:45-55, Fig. 4. This registration identifies the event and client to receive the event. 11:56-60; *see also* 4:36-37, 12:10-15, Figs. 1, 5.

4. Claims 3-5 and 9-13

Lawson alone or combined with Tsutsumitake renders claims 3-5 and 9-11 obvious for the reasons stated in [G1-\[15.0\]](#), [G1-\[14.2\]](#), [G1-\[14.5\]](#), and Easttom Decl. ¶103-26, 168-92. In addition, Lawson’s local registry corresponds to claim 4 and Lawson’s global registry corresponds to claim 5. *See* [G1-\[15.0\]](#), § VI.B.3. Lawson’s local and/or global stored at a local server correspond to claims 10 and 11. 8:5-67, Figs. 1-2. Claims 12-13 are obvious for the reasons stated in § VI.B.2, [G1-\[15.0\]](#), [G1-\[14.4\]](#), and Easttom Decl. ¶185-92 because Lawson’s remote event processing block filters and routes events to registered servers and event consumers.

5. Claim 6

Lawson alone or combined with Tsutsumitake discloses claim 6, which recites various ways that the “property of the live object” directly effects or does not directly

effect a “visual representation of the live object in a data representation.” Lawson’s phone directory and payroll examples and Tsutsumitake’s electric plant monitoring example disclose events updating data having a direct effect on a visual representation (e.g., user sees updated phone number, payroll data, current value 71, device state 72). See [G1-\[14.6\]](#), Easttom Decl. ¶127-30.

Tsutsumitake also discloses events updating data that would have no direct effect on a visual representation (e.g., refresh “period,” “target,” “name” in Fig. 8’s HTML code). 14:4-15. See Easttom Decl. ¶128.

```
< HTML >
< HEAD >
< TITLE > CURRENT MONITOR < /TITLE >
< /HEAD >
< BODY >
< H1 > CURRENT MONITOR SCREEN < /H1 >
< H2 > PRESENT CURRENT VALUE < /H2 >
  < HANDLER type=text name=currency > A
  < EMBED >
    < EVENT url=http://abc.def/current period=1 target=currency >
  < /EMBED >
< H2 > STATE OF DEVICE < /H2 >
  < HANDLER type=text value=normal name=valid >< HANDLER type=text value=abnormal name=invalid >
  < EMBED >
    < EVENT url=http://abc.cef/state target=valid,invalid >
  < /EMBED >
< /BODY >
< /HTML >
```

FIG. 8

VII. GROUND 2: Claims 1-37 are obvious over Kostas alone or combined with Tsutsumitake

A. Overview

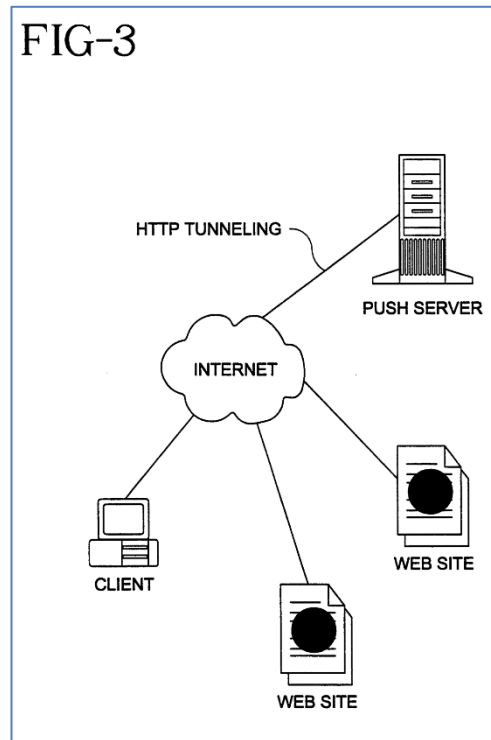
1. Kostas (EX1007)

Kostas, which was not considered during prosecution, discloses “a system and method for displaying, in near real-time, data published from various computer systems and networks, such as the Internet.” 1:22-24. The system is “web-enabled, providing tight integration with all popular [web] browsers.” 30:17-20. “In [Kostas’s] publish/subscribe architecture [(aka ‘model-view’ architecture)] ... various information sources [(e.g., different sources such as Sportsticker, CNN, Reuters, etc.)], most likely under a contractual agreement with [Kostas’s] push server, constantly publish and broadcast data to the push server, using any desired communication method. [A] client computer subscribes to desired types of information available from the push server.” 7:12-18, 16:10-13, 2:11-20, 3:9-14.

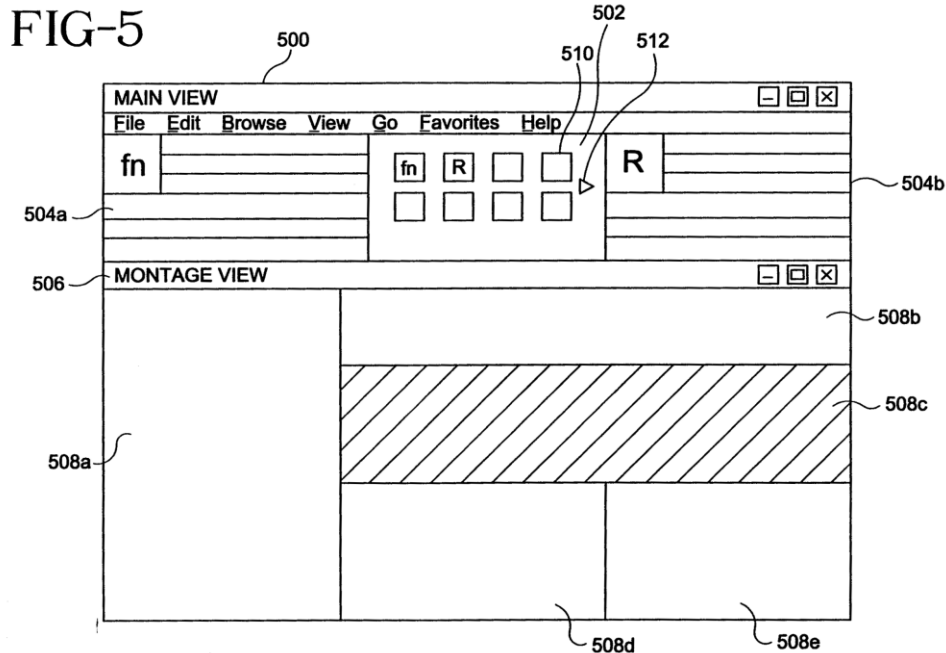
Fig. 3 shows, as an illustration, “a private push server [] made accessible to the client via the Internet,” but there would be numerous push servers and clients. 10:5-6; *see also* Figs. 1-2.

“Push channels [at the push server] use Model Services to subscribe to a particular data topic [from an information source (e.g., Sportsticker, CNN, etc.)].... When new data for this topic is generated, it is sent to the [model]

FIG-3



container, which determines the [net]channel[(s)] for which the data is destined and sends the appropriate message.” 21:10-16. At the clients, “netchannel widgets,” which are special objects programmed to listen for events generated by netchannel models[,] ... register interest in updates from netchannel models.” 23:12-22. “These widgets are graphical information controls, such as stock tickers, scrolling headline displays, and dynamic text labels.” 23:16-18. “As [] updates are received, the widget adds the new data to its information display and shows the data onscreen.” 23:34-35. An example in Fig. 5 shows “panels 504a, 504b for viewing ‘netchannels,’ which each display continuously updated information from specific data sources.” 11:34-12:1.



2. Tsutsumitake (EX1007)

Tsutsumitake was summarized in § VI.A.2 above.

3. Motivation to Combine

Under *KSR*, a POSA would have been motivated to combine *Kostes* with *Tsutsumitake*. Both are concerned with seeking to provide real-time information such as through the Internet, so they are in the same field, address similar problems, and have similar components. *Kostes* is directed towards displaying, in near real-time, data published from various computer systems and networks, such as the internet. *Tsutsumitake* describes information providing servers connected to the Internet. Both inventions work via the Internet, and both seek to provide real-time information. A POSA would have combined them using known methods to follow known techniques. See Easttom Decl. ¶440. Moreover, combining *Tsutsumitake* to

Kostes would have been in accord with design trends at the time, and a POSA would have been motivated to add features together in a combined system that would be more attractive to content producers, developers, and consumers. *See id.*

The combination also would solve known difficulties with known solutions. Tsutsumitake, for example, noted that the conventional push mechanism used on the World Wide Web to update page information in a browser can have problems with time-sequential data that is updated in short intervals. Thus, in addressing this, a POSA would have been motivated to combine Tsutsumitake with the subscription request of Kostes which would facilitate efficient update of page information. *See id.* ¶441. A POSA would recognize that the netchannels of Kostes, which display continuously updated information from specific sources would improve the updating of page information in Tsutsumitake. *See id.*

Combining Tsutsumitake and Kostes would have been straightforward for a POSA, within POSA's skill level, and would not have required extensive experimentation or imposed undue burden, and a POSA could have combined these elements using known methods. *See id.* ¶442. This is especially so given that both references have many points of similarity and overlap that would facilitate this improvement in a combined system. *See id.* This natural overlap makes such a combination yield predictable results and as such a POSA would also have had a high expectation of success. *See id.*

B. Detailed Analysis

1. Claims 14-37

Claims	Kostes (EX1007) and Tsutsumitake (EX1006)
[14.0] A method comprising:	If the preamble is limiting, Kostes discloses a “a system and method for displaying, in near real-time, data published from various computer systems and networks, such as the Internet” 1:22-24. <i>See Easttom Decl.</i> ¶¶584-85.
[14.1] providing, using a processing device of an input source, a data representation to a client device, different from the input source, coupled to a routing network, wherein the data representation includes at least one live object recognizable by the client device, and	Kostes alone or combined Tsutsumitake with discloses providing, using a processing device of an input source (information sources), a data representation (data/webpage) to a client device (client/netchannel widget), different from the input source, coupled to a routing network (push server model services), wherein the data representation includes at least one live object recognizable by the client device (data recognized by client/netchannel widget). <u>Kostes</u> Kostes discloses a “publish/subscribe architecture [(aka “model-view’ architecture’)] ... [where] various information sources [(e.g., different sources such as Sportsticker, CNN, Reuters, etc.)], most likely under a contractual agreement with the push server, constantly publish and broadcast data to the push server, using any desired communication method. [A] client computer subscribes to desired types of information available from the push server.” 7:12-18, 16:10-13, 2:11-20, 3:9-14. “In summary, published information is pushed from the [information] source to the [push] server, and a subset thereof is passed on from the server to the client.” 7:20-23. “[E]very component in [Kostes’s] display system ... [is] web-enabled.” 30:17-20.

Claims	Kostes (EX1007) and Tsutsumitake (EX1006)
	Fig. 3 shows, as an illustration, “a private push server [] made accessible to the client via the Internet,” but there would be numerous push servers and clients. 10:5-6; <i>see also</i> 9:10-19, 9:32-10:3, Figs. 1-2 (alternate server and network configurations), 18:25-19:2, 26:33-27:1, Fig. 8 (repository for instantiating servers and model-view objects). “[I]ncoming push events [(e.g., from ‘various information sources,’ such as CNN, etc.)] are [] directed to appropriate ‘push’ channels.” 17:28-30, 7:12-16, 7:20-23, 2:11-20, 3:9-14. “Push channels [at the push server] use Model Services to subscribe to a particular data topic. Model Services takes the request and formats a subscription request for the push engine in use. When new data for this topic is generated, it is sent to the [model] container, which determines the [net]channel[(s)] for which the data is destined and sends the appropriate message.... The models used in these cases simply add the content of the event to their data store and notify all [clients/netchannels/]observers that new content is available.” 21:10-21; <i>see also</i> 16:15-35, 20:10-12, 23:4-6, 17:32-18:4, Fig. 7. “Model classes are implemented as ‘Observable’ classes. “Observable” objects, as is well known in the art, are a type of object which generates events that may be observed, or processed, by arbitrary ‘Observer’ objects. (This pattern allows models FIG-3 <pre> graph TD Internet((INTERNET)) Client[CLIENT] PushServer[PUSH SERVER] WebSite1[WEB SITE] WebSite2[WEB SITE] Internet --- Client Internet --- PushServer Internet --- WebSite1 Internet --- WebSite2 PushServer -- HTTP TUNNELING --- Internet </pre>

Claims	Kostes (EX1007) and Tsutsumitake (EX1006)
	to generate events when new information is delivered without “knowing” anything about the objects that are interested in receiving these events.)” 16:28-35. “Netchannels [at the client] are capable of presenting virtually any type of dynamic content, including stock quotes, news headlines and special alerts.” 12:15-18; <i>see also</i> 11:19-25. “[N]etchannel widgets,” which are special objects programmed to listen for events generated by netchannel models.... These widgets are graphical information controls, such as stock tickers, scrolling headline displays, and dynamic text labels.... [N]etchannel widgets register interest in updates from netchannel models.” 23:12-22. “As [] updates are received, the widget adds the new data to its information display and shows the data onscreen.” 23:34-35. <u>Tsutsumitake</u> See G1-[14.1]. See Easttom Decl. ¶587-96.
[14.2] causing the client device to respond to the live object of the data representation by determining an object identifier (ID) of the live object and to register for updates of the live object with the routing network,	Kostes alone or combined with Tsutsumitake discloses causing the client device (client/netchannel widget) to respond to the live object of the data representation by determining an object identifier (ID) of the live object (ID code) and to register for updates of the live object with the routing network (client/netchannel widget registers), such that registering the client device with the routing network provides client connection information to a node in the routing network (models). <u>Kostes</u>

Claims	Kostes (EX1007) and Tsutsumitake (EX1006)
such that registering the client device with the routing network provides client connection information to a node in the routing network; and	Kostes discloses “[netchannels] are the ‘observers’ in the observer-observable pattern. This means that a widget can register as an observer of a model class, thus receiving events when data is changed and automatically updating its display.” 17:9-13; <i>see also</i> 7:17-20, 18:30-34. The registration is associated with an “ID code [] assigned to the channel model.” 22:27-31, 23:18-22. “Each model automatically broadcasts [update data] to all widgets that have previously registered interest in that model’s updates.” 23:31-33. Numerous models may be instantiated on numerous push servers. <i>See</i> 18:25-19:2, 26:33-27:1, Fig. 8. <u>Tsutsumitake</u> <i>See</i> G1-[14.2]. <i>See</i> Easttom Decl. ¶597-601.
[14.3] sending, using the processing device of the input source, an update message to the routing network, wherein the update message identifies the live object and contains update data that updates a property of the live object,	Kostes alone or combined with Tsutsumitake discloses sending, using the processing device of the input source (information sources), an update message (topic/event) to the routing network (push server model services), wherein the update message identifies the live object (contains ID code) and contains update data (updated data) that updates a property of the live object (updates data at client/netchannel widget). <u>Kostes</u> Kostes discloses “[w]hen new data for [an event/]topic is generated, it is sent to the [model] container, which determines the [net]channel[(s)] for which the data is destined and sends the appropriate message.” 21:13-16; <i>see also</i> 17:32-18:4, 18:30-19:2, Fig. 7. The update message includes the “ID code.” <i>See</i> 22:27-23:22, 23:24-33.

Claims	Kostes (EX1007) and Tsutsumitake (EX1006)
	At the client, when “updates are received, the widget adds the new data to its information display and shows the data onscreen.” 23:34-35. <u>Tsutsumitake</u> See G1-[14.3]. See Easttom Decl. ¶602-08.
[14.4] wherein a gateway device at the routing network is configured to identify a category of the update message based on the input source, to determine a node type to which the identified category maps, and to route the update message to the node, having the node type, at the routing network,	Kostes discloses wherein a gateway device at the routing network (incoming events portion of model services at push server) is configured to identify a category of the update message based on the input source (identify models to receive topics/events based on, e.g., information source, server), to determine a node type to which the identified category maps (models needing topic/event), and to route the update message to the node, having the node type, at the routing network (transfer topic/event only to models needing topic/event). Kostes discloses “incoming push events are [] directed to appropriate [models for] ‘push’ channels” by a portion of model services at the push server. 17:29-30, 20:19-21, 21:10-16, 23:4-6, Fig. 7. “The models used in these cases simply add the content of the event to their data store and notify all [clients/netchannels/]observers that new content is available.” 21:18-21. “[O]nly data that is desired and subscribed to by [a] user [of that model] is communicated from the server [(and eventually)] to the client computer[], when means] the bandwidth required for the communication ... is significantly reduced.” 7:27-31; <i>see also</i> 8:4-19. “[A] channel is concentration of data from a resource such as a web server.” 11:1-3. “For example, one netchannel might display breaking business headlines from a major news source, while another could

Claims	Kostes (EX1007) and Tsutsumitake (EX1006)
	be used to display firm announcements relating to a specific business unit ... [, and another could] monitor events received from an enterprise computer management system.” 12:15-21. <i>See also</i> 24:1-27 (additional filtering options). <i>See</i> Easttom Decl. ¶609-13.
[14.5] wherein the node is configured to identify the client device as a registered device and to route the update message to the client device, and	Kostes alone or combined with Tsutsumitake discloses wherein the node (model) is configured to identify the client device as a registered device (identify registered client/netchannel widget) and to route the update message to the client device (transfer topic/event to client/netchannel widget). <u>Kostes</u> Kostes discloses “[e]ach model automatically broadcasts [update data] to all widgets that have previously registered interest in that model’s updates.” 23:31-33; <i>see also</i> 17:9-13, 18:30-19:2, 21:13-16, Fig. 7. The update message includes the “ID code” recognized by the client/netchannel widget. <i>See</i> 22:27-23:22, 23:24-35. <u>Tsutsumitake</u> <i>See</i> G1-[14.5]. <i>See</i> Easttom Decl. ¶614-17.
[14.6] wherein the client device processes the update message upon	Kostes alone or combined with Tsutsumitake discloses wherein the client device (client/netchannel widget) processes the update message upon receipt to update the

Claims	Kostes (EX1007) and Tsutsumitake (EX1006)
receipt to update the property of the live object at the client device.	property of the live object at the client device (updates data at client/netchannel widget). <u>Kostes</u> Kostes discloses when “updates are received [from models], the widget [at the client] adds the new data to its information display and shows the data onscreen.” 23:34-35, Figs. 5, 7; <i>see also</i> 11:19-25, 12:13-21, 23:24-33, 30:22-31:16. “These widgets are graphical information control, such as stock tickers, scrolling headline displays, and dynamic text labels.” 23:16-18. An example in Fig. 5 shows “panels 504a, 504b for viewing ‘netchannels,’ which each display continuously updated information from specific data sources.” 11:34-12:1.

Claims	Kostes (EX1007) and Tsutsumitake (EX1006)
	FIG-5 Tsutsumitake See G1-[14.6]. See Easttom Decl. ¶618-21.
[15.0] The method of claim 14, wherein providing the data representation to the client	Kostes discloses this claim. See G2-[14]. In addition: Kostes discloses “widget[s] can register as an observer of a model class, thus receiving events [from the models] when data is changed.” 17:9-13; <i>see also</i> 7:17-20, 18:30-34.

Claims	Kostes (EX1007) and Tsutsumitake (EX1006)
device includes providing the live object that causes the client device to register with a client proxy of the routing network.	The registration is stored at the model and updates are sent to registered nodes. <i>See</i> 22:27-31, 23:18-22, 23:31-33. This corresponds to the 722 patent’s client proxy, in which “client registration information is stored at the nodes, and the client proxy merely passes messages among the clients and nodes.” 722 patent 3:50-59; <i>see also</i> 21:29-31 (“each node type can serve as a node and as a client proxy”). <i>See</i> Easttom Decl. ¶622-26.
[16.0] The method of claim 14, wherein providing the data representation to the client device includes providing the live object that causes the client device to register with the node of the routing network.	Lawson alone or combined with Tsutsumitake discloses this claim. <i>See</i> G2-[14]. In addition: <u>Kostes</u> Kostes discloses “[u]sing a push server also permits the automatic update of the client application software. For example, when existing netchannels (described below) need to be updated, or new netchannels become available and need to be added, the push server can push the new binary code required to accomplish these tasks directly to the client computer.” 8:18-26. “[Netchannels] are the ‘observers’ in the observer-observable pattern. This means that a widget can register as an observer of a model class, thus receiving events when data is changed and automatically updating its display.” 17:9-13; <i>see also</i> 7:17-20, 18:30-34. <u>Tsutsumitake</u> <i>See</i> G1-[16.0]. <i>See</i> Easttom Decl. ¶627-31.

Claims	Kostes (EX1007) and Tsutsumitake (EX1006)
[17.0] The method of claim 14, wherein providing the data representation to the client device includes providing an activation module that is executed by the client device and that registers the live object with the routing network.	Lawson alone or combined with Tsutsumitake discloses this claim. See G2-[14]. In addition: <u>Kostes</u> Kostes discloses netchannels may be implemented as “Java™ applications” or Java “applets ... that run within a web browser” (both of which run on “Java™ Virtual Machines (VM’s)”). 27:14-21; <i>see also</i> 11:19-25, 27:5-6, Fig. 7. As explained in G2-[16.0], push servers may push updates of netchannel code to clients and netchannel widgets at clients register for updates with the push server model services. <u>Tsutsumitake</u> <i>See</i> G1-[17.0]. <i>See</i> Easttom Decl. ¶632-35.
[18.0] The method of claim 17, wherein providing the activation module includes providing the activation module that is configured to determine a node type that handles registration and that causes the client device to register with the node	Lawson alone or combined with Tsutsumitake discloses this claim. See G2-[17.0]. In addition: <u>Kostes</u> Kostes discloses netchannel widgets at clients register with models at the push server to receive events the clients/netchannel widgets are registered for from a given model. <i>See</i> 17:9-20, 18:30-34, 23:31-33. <u>Tsutsumitake</u> <i>See</i> G1-[18.0].

Claims	Kostes (EX1007) and Tsutsumitake (EX1006)
corresponding to the node type.	See Easttom Decl. ¶636-41.
[19.0] The method of claim 17, wherein providing the activation module includes providing the activation module that is configured to determine a message category of the data representation and that causes the client device to register with a node having a node type corresponding to the message category.	Kostes alone or combined with Tsutsumitake discloses this claim. See G2-[17.0]. In addition: <u>Kostes</u> Kostes discloses netchannel widgets at clients register with models at the push server to receive events the clients/netchannel widgets are registered for from a given model, including registering for events from a particular information source, channel, or server. See 11:1-3, 12:15-21, 17:9-20, 17:29-30, 18:30-34, 21:10-21. See also 24:1-27 (additional filtering options). <u>Tsutsumitake</u> See G1-[19.0]. See Easttom Decl. ¶642-46.

Claims 20-37 are obvious for the reasons stated in the corresponding limitations of claims 14-19, which are identified in the table below. *See also* EX1015 (claims 14-37 concordance).

Apparatus	Article	Software	Corresponding G2 Argument	Easttom Decl. ¶
[20.0]	[26.0]	[32.0]	[14.0]	647-48, 708-10, 769-71
[20.1]	[26.1]	[32.1]	[14.1]	649-59, 711-20, 772-76
[20.2]	[26.2]	[32.2]	[14.2]	660-63, 721-23, 777-81
[20.3]	[26.3]	[32.3]	[14.3]	664-68, 724-30, 782-86
[20.4]	[26.4]	[32.4]	[14.4]	669-72, 731-35, 787-91
[20.5]	[26.5]	[32.5]	[14.5]	673-76, 736-40, 792-95
[20.6]	[26.6]	[32.6]	[14.6]	677-80, 741-43, 796-803
[21.0]	[27.0]	[33.0]	[15.0]	681-85, 744-47, 804-06
[22.0]	[28.0]	[34.0]	[16.0]	686-90, 748-54, 807-810
[23.0]	[29.0]	[35.0]	[17.0]	691-95, 755-58, 811-15
[24.0]	[30.0]	[36.0]	[18.0]	696-701, 759-63, 816-18
[25.0]	[31.0]	[37.0]	[19.0]	702-07, 764-68, 819-22

2. Claims 1 and 7

Except as discussed below, claims 1 and 7 are substantially identical and obvious for the reasons stated in the corresponding limitations of claim 14 and the

Easttom Declaration, which are identified in the tables below. *See also* EX1016 (claims 1-13 concordance).

Claim 1	Corresponding G2 Argument	Easttom Decl. ¶
[1.0]	[14.0]	444-48
[1.1]	[14.3]	444-48
[1.2]	[14.4]	449-52
* [1.3]	[14.4]	453-61
[1.4]	[14.4]	462-65
[1.5]	[14.1] , [14.2] , [14.5] ,	466-71
[1.6]	[14.5]	472-74
[1.7]	[14.6]	472-74

Claim 7	Corresponding G2 Argument	Easttom Decl. ¶
[7.0]	[14.0]	515-19
[7.1]	[14.3] , [14.4]	520-25
[7.2]	[14.3] , [14.4]	526-27
[7.3]	[14.3] , [14.4]	528-29
* [7.4]	[14.3] , [14.4]	530-32
[7.5]	[14.2] , [14.5] , [14.6]	533-36
* [7.6]	[14.2] , [14.5] , [14.6]	537-39
[7.7]	[14.2] , [14.5] , [14.6]	540-41
[7.8]	[14.2] , [14.5] , [14.6]	542-43

In addition, Kostas discloses the “mapping” to “node type” and for “controlling ... message traffic” elements in claims 1.3, 7.4, and 7.6.

Kostes discloses models at push servers store events for subsequent distribution to clients/netchannels. 21:18-21. “[O]nly data that is desired and subscribed to by [a] user [of that model] is communicated from the server [(and eventually] to the client computer[], when means] the bandwidth required for the communication ... is significantly reduced.” 7:27-31. “As a result, redundant network traffic is significantly reduced, which in turn greatly reduces bandwidth between the source and the server, and between the server and the client.” 8:13-19. Kostes also discloses “granular filtering of information channel-by-channel,” which can map events to necessary models at push servers. 24:2-3. *See also* Easttom Decl. ¶453-61, 530-32, 537-39.

3. Claim 2 and 8

Claims 2 and 8 are obvious for the reasons stated in [G2-\[14.2\]](#) and Easttom Decl. ¶475-79, 544-46. In addition, Kostes alone or combined with Tsutsumitake disclose the “determine if the client has registered for updates”/“a connection to the client” based on the “objectID” elements in claims 2 and 8.

Kostes discloses “all communication with the push server is conducted via Model Services [and] a unique ID code is assigned to the channel model at the time of subscription for purposes of tracking incoming and outgoing messages” between the model and registered clients/netchannels. 22:21-31, 23:24-35, 18:30-19:2.

For Tsutsumitake, *see* § VI.B.3

4. Claims 3-5 and 9-13

Kostes alone or combined with Tsutsumitake renders claims 3-5 and 9-11 obvious for the reasons stated in [G2-\[15.0\]](#), [G2-\[14.2\]](#), [G2-\[14.5\]](#), and Easttom Decl. ¶¶480-508, 547-583. In addition, Kostes’s registry of clients/netchannels stored at the model corresponds to claims 4, 5, 10, and 11. See [G2-\[15.0\]](#), § VII.B.3. Claims 12-13 are obvious for the reasons stated in § VII.B.2, [G2-\[15.0\]](#), [G2-\[14.4\]](#), and Easttom Decl. ¶¶573-83 because Kostes’s models filter and route events to registered models and clients/netchannels.

5. Claim 6

Kostes alone or combined with Tsutsumitake discloses claim 6, , which recites various ways that the “property of the live object” directly effects or does not directly effect a “visual representation of the live object in a data representation.” Kostes discloses events updating data having a direct effect on a visual representation (e.g., user sees updated data, change to “foreground and background color”). See [G2-\[14.6\]](#), 23:27-31, Easttom Decl. ¶¶509-14. Kostes also discloses events updating data having no direct effect on a visual representation (e.g., updated timestamp, XML schema, binary software code, java/script source code/function calls). 8:21-26, 9:5-7, 23:31, 27:29-28:10, 28:27-31, Fig. 5. See Easttom Decl. ¶¶513.

For Tsutsumitake, see § VI.B.5.

VIII. DISCRETIONARY DENIAL IS NOT APPROPRIATE

A. 35 U.S.C. § 325(d)

The prior art references relied on herein (Lawson, Tsutsumitake, and Kostas) were not cited or substantively addressed during prosecution, and nothing suggests that their teachings relied on herein—much less the combination of those teachings as informed by a POSA’s knowledge and expert testimony—were considered by the examiner. *See, e.g., Mylan Pharm. Inc., v. Regeneron Pharm., Inc.*, IPR2022-01226, Paper 22 at 27-29 (PTAB Jan. 11, 2023). Further, this petition relies on the references Lawson and Kostas, which were not cited in the *ex parte* reexamination request filed by Unified Patents, LLC (Control No. 90/019,662), and seeks review of additional claims that are not part of the reexamination. Accordingly, this petition should not be denied under Section 325(d).

B. 35 U.S.C § 314(a)

Fintiv and the Office’s interim guidelines dated June 21, 2022 (“Interim Procedures”) favor institution, which best serves the efficiency and integrity of the patent system. Factor 1 is at worst neutral because no request for a stay pending IPR has been filed yet. *Sand Revolution II LLC v. Continental Intermodal Group - Trucking LLC*, IPR2019-01393, Paper 24, 7 (PTAB June 16, 2020) (informative). Nonetheless, Petitioners are considering requesting a stay, which favors institution.

Factors 2 and 3 also favor institution. The related district court cases against Petitioners, filed on November 15, 2023, are in their early stages. The *Markman* process has not begun, no claim construction order has issued, fact discovery remains open and no merits fact depositions have been taken, and expert discovery has not begun. In October 2024, the trial date was extended to September 22, 2025, but on November 5, 2024, after the trial date was moved back, Patent Owner filed a Second Amended Complaint against Liberty Mutual,⁶ an Amended Complaint against Comerica, and finally served source code infringement contentions. Notably, Patent Owner received a lengthy amount of time to provide source code contentions, *i.e.*, from default of 30 days to 75 with extension, and then an additional about 60 days past the 75-day deadline because of Patent Owner's delay. Without the code contentions, Petitioners did not actually know the full details of Patent Owner's infringement theories. In *Sand Revolution*, the district court had "conducted a *Markman* hearing and entered a related Order." IPR2019-01393, Paper 24 at 10. Despite this fact, the Board found that the third *Fintiv* factor "weighs only marginally, if at all, in favor of exercising discretion to deny institution." *Id.* Here,

⁶ Liberty Mutual also has not answered and has a pending motion to dismiss Patent Owner's entire complaint based on improper venue under 28 U.S.C. § 1400(b) since January 2024.

the parallel proceeding is at a substantially earlier phase. Accordingly, these factors weigh against discretionary denial.

As to factors 4 and 5, certain challenged claims were not asserted in the district court, so this Petition addresses issues that will not be litigated there, and this weighs against discretionary denial. Moreover, while Petitioners and Patent Owner are the same parties in the district court, this is “far from an unusual circumstance” in IPR and does not warrant discretionary denial. *Sand Revolution*, IPR2019-01393, Paper 24 (instituting where parties were the same).

The Board only addresses Factor 6 if Factors 1-5 support discretionary denial, which they do not. *CommScope Techs. LLC. v. Dali Wireless, Inc.*, IPR2022-01242, Paper 23 (Feb. 27, 2023) (precedential).

This Petition presents a compelling unpatentability challenge. The cited prior art, none of which was cited by the examiner, rebuts the Patent Owner’s arguments during prosecution that persuaded the examiner to allow the claims. Accordingly, the Board should not deny institution under *Fintiv* and the Interim Procedures.

IX. MANDATORY NOTICES

Pursuant to 37 C.F.R. § 42.8(b), Petitioners state:

A. Real Party-in-Interest

The real parties-in-interest are Liberty Mutual Insurance Company, Liberty Mutual Technology Group, Inc., Liberty Mutual Holding Company Inc., Liberty

Mutual Group Inc., Liberty Mutual Plano LLC, Comparion Insurance Agency, LLC, and Ironshore Holdings (U.S.) Inc. (“Liberty Mutual”) and Comerica Incorporated (“Comerica”).

B. Related Matters

Patent Owner Intellectual Ventures I LLC (along with Intellectual Ventures II LLC), has asserted the 722 patent (and three other unrelated patents) in:

- *Intellectual Ventures I LLC et al. v. Liberty Mutual Holding Company Inc., et al.*, No. 23-cv-525 (E.D. Tex. filed November 15, 2023)
- *Intellectual Ventures I LLC et al. v. Comerica Incorporated*, No. 23-cv-524 (E.D. Tex. filed November 15, 2023)
- *Intellectual Ventures I LLC et al. v. JP Morgan Chase Bank, National Association et al.*, No. 23-cv-523 (E.D. Tex. filed November 15, 2023)

These cases were consolidated, with the latter Case No. 23-cv-523 as the lead case.

The 722 patent is also asserted in the following cases:

- *Intellectual Ventures I LLC et al., v. American Airlines, Inc.*, No. 24-cv-980 (E.D. Tex. filed November 2, 2024)
- *Intellectual Ventures I LLC et al., v. Southwest Airlines, Co.*, No. 24-cv-277 (W.D. Tex. filed November 2, 2024).

The action against JP Morgan has been dismissed with prejudice. The actions against Petitioners and the other cases remain pending. There has been no ruling on claim construction in any case.

The 722 patent is also the subject of an *ex parte* reexamination request filed by Unified Patents, LLC on September 16, 2024 (Control No. 90/019,662). Reexamination was ordered on October 16, 2024.

Petitioners are not aware of any other civil actions or proceedings filed in connection with the 722 patent.

C. Lead and Backup Counsel and Service Information

Lead Counsel	Back-up Counsel
Joshua D. Curry josh.curry@lewisbrisbois.com LEWIS BRISBOIS BISGAARD SMITH LLP 600 Peachtree St. NE, Suite 4700 Atlanta, GA 30308 Ph: 404.348.8585 Fax: 404.467.8845 Reg No. 65,478	François O. Ecclesiaste francois.ecclesiaste@lewisbrisbois.com LEWIS BRISBOIS BISGAARD SMITH LLP Wells Fargo Center 90 S 7th Street, Suite 2800 Minneapolis, MN 55402 Ph: 612.428.5048 Fax: 612.428.5001 Reg No. 75,836

Petitioners consent to service by email to the counsel above.

X. CONCLUSION

For at least the foregoing reasons, Petitioners request IPR be instituted and claims 1-37 (all claims) of the 722 patent be cancelled.

Respectfully submitted,

/Joshua D. Curry/

Joshua D. Curry – Lead Counsel
Registration No. 65,478
LEWIS BRISBOIS BISGAARD SMITH LLP
600 Peachtree St. NE, Suite 4700
Atlanta, GA 30308
Ph: 404.348.8585
Fax: 404.467.8845
josh.curry@lewisbrisbois.com

François O. Ecclesiaste
Registration Reg No. 75,836
LEWIS BRISBOIS BISGAARD SMITH LLP
Wells Fargo Center
90 S 7th Street, Suite 2800
Minneapolis, MN 55402
Ph: 612.428.5048
Fax: 612.428.5001
francois.ecclesiaste@lewisbrisbois.com

*Counsel for Liberty Mutual Insurance
Company, Liberty Mutual Technology
Group, Inc., Liberty Mutual Holding
Company Inc., Liberty Mutual Group Inc.,
Liberty Mutual Plano LLC, Comparion
Insurance Agency, LLC, Ironshore
Holdings (U.S.) Inc.*

-and-

*Counsel for Petitioner Comerica
Incorporated*

CERTIFICATE OF SERVICE

Pursuant to 37 C.F.R. § 42.6(e) and 42.105(b) and agreement of the parties, the undersigned certifies that on the date indicated below the foregoing document with any attachments was served by email on counsel for the Patent Owner as indicated below.

Brandon R. Theiss
Volpe Koenig
30 South 17th Street, 18th Floor
Philadelphia, PA 19103
btheiss@vklaw.com

Jonathan K. Waldrop
Kasowitz Benson Torres LLP
333 Twin Dolphin Drive, Suite 200
Redwood Shores, CA 94065
jwaldrop@kasowitz.com
IntellectualVentures@kasowitz.com

Date: 11/20/2024

/Joshua D. Curry/_____

CERTIFICATION OF WORD COUNT

Pursuant to 37 C.F.R. § 42.24(d), Petitioners certify that the foregoing IPR petition contains 13,976 words, excluding the portions as permitted by § 42.24(a).

Date: 11/20/2024

/Joshua D. Curry/_____