



(86) Date de dépôt PCT/PCT Filing Date: 2009/08/13  
 (87) Date publication PCT/PCT Publication Date: 2010/02/18  
 (45) Date de délivrance/Issue Date: 2015/10/06  
 (85) Entrée phase nationale/National Entry: 2011/02/01  
 (86) N° demande PCT/PCT Application No.: US 2009/004654  
 (87) N° publication PCT/PCT Publication No.: 2010/019258  
 (30) Priorité/Priority: 2008/08/14 (US12/191,585)

(51) Cl.Int./Int.Cl. *H04L 12/26* (2006.01)  
 (72) Inventeurs/Inventors:  
 WENIG, ROBERT, US;  
 KNUDSEN, TED, US  
 (73) Propriétaire/Owner:  
 INTERNATIONAL BUSINESS MACHINES  
 CORPORATION, US  
 (74) Agent: WANG, PETER

(54) Titre : PROCÉDE ET SYSTÈME DE COMMUNICATION ENTRE UN SYSTÈME CLIENT ET UN SYSTÈME SERVEUR

(54) Title: METHOD AND SYSTEM FOR COMMUNICATION BETWEEN A CLIENT SYSTEM AND A SERVER SYSTEM

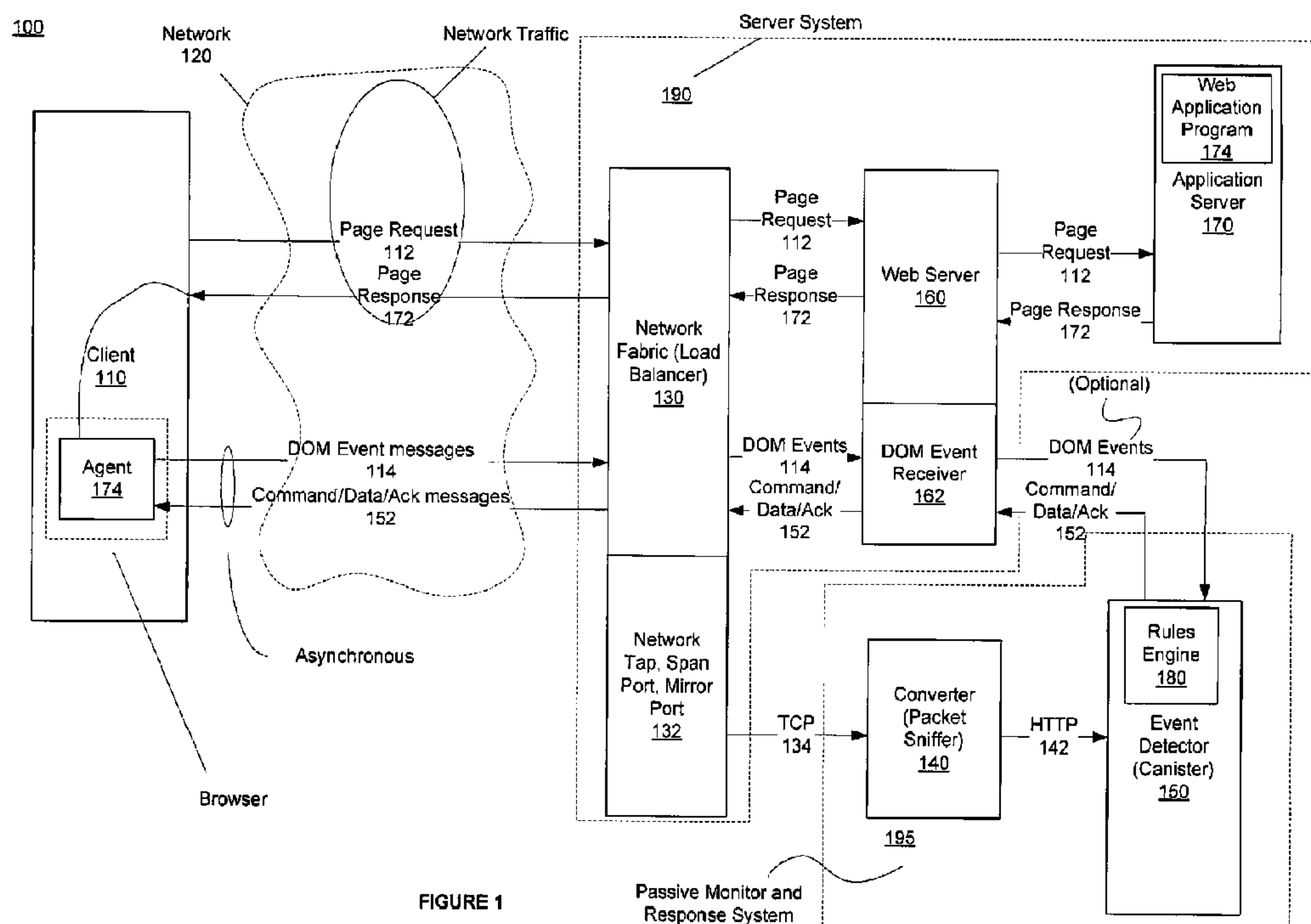


FIGURE 1

(57) Abrégé/Abstract:

A method of a server system communicating with a client system. The method includes a passive system monitoring network traffic between the client-server system. The server sending a page response including an agent, e.g., a Javascript that becomes operable on the client system. The agent monitors events on the client side, e.g., DOM events, and communicates these to the server side. A set of rules that define prescribed user events are maintained by the passive system and applied to the monitored network traffic to determine whether a prescribed user event occurred. Responsive to the prescribed user event occurring, an asynchronous message is generated and sent to the agent in response to a received DOM event message, the asynchronous message comprising a command to be executed by the agent to alter the users web application flow at the client in a manner that is substantially transparent to the web application.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization  
International Bureau(43) International Publication Date  
18 February 2010 (18.02.2010)(10) International Publication Number  
**WO 2010/019258 A1**(51) International Patent Classification:  
*G06F 15/16* (2006.01)(21) International Application Number:  
PCT/US2009/004654(22) International Filing Date:  
13 August 2009 (13.08.2009)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:  
12/191,585 14 August 2008 (14.08.2008) US(71) Applicant (for all designated States except US):  
**TEALEAF TECHNOLOGY, INC.** [US/US]; 45 Fremont Street, Suite 1450, San Francisco, CA 94105 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **WENIG, Robert** [US/US]; 181 Buena Vista Avenue, Mill Valley, CA 94941 (US). **KNUDSEN, Ted** [US/US]; 2243 Fifth Avenue, San Rafael, CA 94901 (US).(74) Agent: **TABARROK, Amir, A.**; Murabito, Hao and Barnes LLP, Two North Market Street, Third Floor, San Jose, CA 95113 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— with international search report (Art. 21(3))

(54) Title: METHOD AND SYSTEM FOR COMMUNICATION BETWEEN A CLIENT SYSTEM AND A SERVER SYSTEM

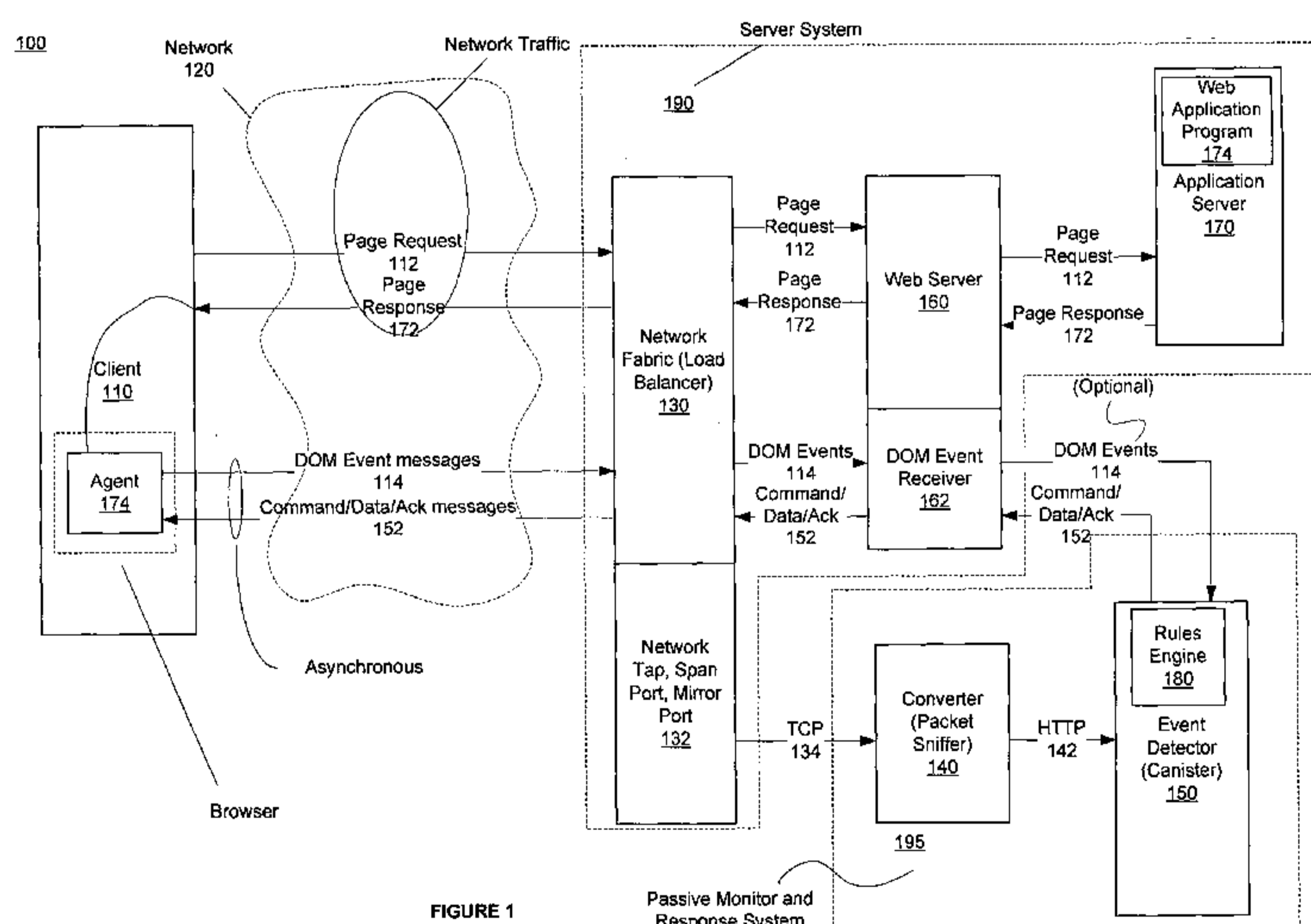


FIGURE 1

(57) Abstract: A method of a server system communicating with a client system. The method includes a passive system monitoring network traffic between the client-server system. The server sending a page response including an agent, e.g., a Javascript that becomes operable on the client system. The agent monitors events on the client side, e.g., DOM events, and communicates these to the server side. A set of rules that define prescribed user events are maintained by the passive system and applied to the monitored network traffic to determine whether a prescribed user event occurred. Responsive to the prescribed user event occurring, an asynchronous message is generated and sent to the agent in response to a received DOM event message, the asynchronous message comprising a command to be executed by the agent to alter the users web application flow at the client in a manner that is substantially transparent to the web application.



WO 2010/019258 PCT/US2009/004654  
METHOD AND SYSTEM FOR COMMUNICATION BETWEEN A CLIENT SYSTEM  
AND A SERVER SYSTEM

## TECHNICAL FIELD

**[0001]** Embodiments of the present invention relate to the field of client-server communications. More particularly, embodiments of the present invention relate to a method and system for passively monitoring network communication between a server and a client, and based thereon, communicating commands and data in a client-server interaction.

## BACKGROUND ART

**[0002]** Web applications have become important business, entertainment and social tools. Commerce is conducted over the Internet via merchandise transactions, financial transactions, etc. Internet communication typically involves a client computer system, e.g., having a browser, communicating with a remote server system. The server system typically has installed thereon a web application program. The client system transmits page requests to the server system which responds with page responses to create a web experience for the client user. The requests and responses are called "network traffic."

**[0003]** Designers of web application programs would like to gain information about client experience including the client behavior and application behavior when interacting with a web application in order to better serve the client. In order to understand the manner in which users interact with a web application, prior art systems have been developed that are able to passively monitor and store the network traffic (without disturbing the web application program or the client) and further these passive systems can separate and store the data by sessions that are unique to a particular client. By analyzing the recorded network traffic, important client behavior may be understood. Also, by recording the network traffic by sessions, a specific client interaction can be replayed to an operator during a client telephone call in order to troubleshoot a web application program or determine how a client can be better assisted.

**[0004]** Unfortunately, it is not an easy task to modify a web application program to take advantage of information that may be obtained by passive monitoring and recording of the network traffic. Server based systems can be very complex and large. Many companies that run such servers resist frequent modification of them and their resident web application programs.

## SUMMARY

**[0005]** Accordingly, a need exists for a system and method that can provide real-time user feedback (to enhance a user's web experience) based on passive monitoring of client-server network traffic without involving substantial modifications to the server system and without involving any substantial modifications to the resident web application program installed on the server system. A need exists for a system and method that can alter a web application's program flow, according to the user's perspective, in a manner that is transparent to the web application program. It will become apparent to those skilled in the art after reading the detailed description of the present invention that the embodiments of the present invention satisfy the above mentioned needs.

**[0006]** For instance, in response to passive monitoring of network traffic, a communication loop may be used to send specific commands and/or information to a client system to cause certain actions to take place on that client system, e.g., a chat window may be opened, a cookie value may be cleared, a document object model (DOM) of a browser may be altered, a session may be terminated, etc. The commands and/or information may originate from a system that is separate from the server system and whose operation does not substantially alter the server system or the web application program installed thereon.

**[0007]** By employing such a system and method, client actions in the context of the history of the user's current session can be used to provide personalized assistance, e.g., by opening a chat window, enhancing security by preventing click fraud, terminating an active session, clearing a cookie value, etc., in a manner that is transparent to the web application program.



- [0008]** More specifically, embodiments of the present invention are directed to a passive monitoring system that is separate from a server system. The passive monitoring system is able to monitor and record network traffic that includes client requests and server responses. The server responses originate from a web application program that is resident on the server system. The recorded network traffic is sessionized. The network traffic can be fed to a rules engine of the passive system which can use the network traffic to recognize when one or more user events occur.
- [0009]** In accordance with embodiments of the present invention, an agent is made operable on the client system by the server system. The agent operates within the client browser and collects various events occurring on the client system, generally referred to as document object model (DOM) events, e.g., mouse movements, mouse clicks, key strokes, the recorded delay between key strokes, etc. The agent, using an asynchronous communication channel, then posts the event data to the server, e.g., transmits the event data to the server system in an asynchronous fashion at prescribed time intervals. In response to the asynchronous post message, the client system expects to receive, and in fact listens for, an acknowledgement signal from the server system.
- [0010]** The passive monitoring system takes advantage of this acknowledgement signal requirement by providing, in addition to the acknowledgement signal, certain commands and/or data that may be generated by the rules engine in response to a recognized user event. These commands and/or data can then be conveniently sent asynchronously to the agent in order to alter the flow of the web based application. Since these commands originate from the passive system, and not the web server, the program flow of the web application is altered in a way that is transparent to the web application program on the server.
- [0011]** An advantage of the embodiments of the present invention is that the user experience of a web application can be augmented and improved (based on user interaction with the web application program) without requiring changes to the web application program on the server side aside from including the Javascript in the response page. The logic that determines if commands and/or data should be provided to the agent can be provided separate from the server based application program, e.g., integrated into a system designed to passively collect and monitor network traffic

between the server system and client systems. The only change required of the server is that it place the agent with a page response.

**[0012]** More particularly, the passive monitoring system may maintain a set of logical rules that are programmable. Application of the set of logical rules to the collected network traffic for a given client determines whether a prescribed event has occurred. In response to the determination, the passive monitoring system may generate a command signal for execution by the agent. Upon the next agent communication to the server, a command/data package may be generated and sent to the client system (alternatively, a mere acknowledgement signal may be sent if no commands are contemporaneously required).

**[0013]** The command/data package may cause the agent operating on the client system to take a certain action, e.g., open a chat window such that an operator can provide help to the user, alter a cookie value, terminate an active session between a browser of the client system and the web application, alter a DOM of a browser of the client system, etc. The agent can take almost any action in response to a command/data package.

**[0014]** Accordingly, events occurring on the client system may be understood by collecting and monitoring network traffic. As a result, the passive monitoring system may provide assistance to a client in real-time, e.g., by opening a chat window. Moreover, security may be enhanced by enabling the agent on the client system to take certain actions in real-time, e.g., terminate an active session, clear a cookie value, prevent click fraud, etc. Importantly, these actions may be taken without substantial alteration to the web application program.

**[0015]** In an alternative embodiment of the present invention, the rules engine of the passive monitoring system may also take into consideration, in addition to the recorded network traffic, the client information sent from the agent. In other words, in determining if an event occurred, the rules engine may, in one

embodiment, consider both events communicated from the engine, and the recorded network traffic.

**[0015a]** In accordance with an embodiment of the invention, there is provided a method of communicating with a client system, said method comprising: responding to page requests from said client system by sending page responses to said client system, said responding performed by a web application; monitoring and storing network traffic communicated between said server system and said client system, said network traffic comprising said page requests and said page responses; monitoring and storing event data received asynchronously from an agent operating within said client system; maintaining a set of rules that define prescribed user events; based on said set of rules and stored network traffic, determining that a prescribed user event occurred within said client system; and in response to said prescribed user event occurring, sending an asynchronous message to said agent, said asynchronous message comprising a command to be executed by said agent to provide web content and functionality that are in addition to said page responses and are substantially transparent to said web application.

**[0015b]** In accordance with another embodiment of the invention, there is provided a system comprising: a server system comprising: network fabric components comprising a network tap, said network fabric components for communicating network traffic between a client system and an application tier; a converter unit coupled to said tap and operable to receive network traffic from said tap and for converting said network traffic to formatted data; a web server comprising a receiver for receiving event data communicated asynchronously from an agent operable within said client system; an application server comprising a web application and operable to receive requests from said client system and operable to generate responses in response thereto, said application server coupled to said web server; and a passive system coupled to said server system, said passive system comprising an event detector system coupled to receive

said formatted data and said event data, said event detector system operable to identify specific user events based on applying a set of prescribed rules to said formatted data, said event detector system operable to communicate commands and data to said receiver in response to identifying a specific user event, said commands and data for communication to said agent in response to an event data received from said agent, said commands and data for providing web content and functionality that are substantially transparent to said web application.



**BRIEF DESCRIPTION OF THE DRAWINGS**

**[0016]** Embodiments of the present invention are illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements and in which:

**[0017]** Figure 1 shows an exemplary system diagram in accordance with one embodiment of the present invention.

**[0018]** Figure 2 shows an exemplary event detector of a passive system in accordance with one embodiment of the present invention.

**[0019]** Figure 3 shows an exemplary communication flow between a client system, the server system and the passive system in accordance with one embodiment of the present invention.

**[0020]** Figures 4A, 4B, 4C and 4D show an exemplary sequence of events observed on the client system in accordance with one embodiment of the present invention.

**[0021]** Figure 5 shows an exemplary flow diagram in accordance with one embodiment of the present invention.

**[0022]** Figure 6 illustrates a general purpose computer system that may serve as a platform for embodiments of the present invention.

**DETAILED DESCRIPTION**

**[0023]** Reference will now be made in detail to embodiments of the present invention, examples of which are illustrated in the accompanying drawings. While the invention will be described in conjunction with these embodiments, it will be understood that they are not intended to limit the invention to these embodiments. On the contrary, the invention is intended to cover alternatives, modifications and equivalents, which may be included within the scope of the invention as defined by the appended claims. Furthermore, in the following detailed description of the present invention,

numerous specific details are set forth in order to provide a thorough understanding of the present invention. However, it will be evident to one of ordinary skill in the art that the present invention may be practiced without these specific details. In other instances, well known methods, procedures, components, and circuits have not been described in detail as not to unnecessarily obscure aspects of the invention.

## NOTATION AND NOMENCLATURE

**[0024]** Some portions of the detailed descriptions which follow are presented in terms of procedures, steps, logic blocks, processing, and other symbolic representations of operations on data bits that can be performed on computer memory. These descriptions and representations are the means used by those skilled in the art to most effectively convey the substance of their work to others skilled in the art. A procedure, computer executed step, logic block, process, etc., is here, and generally, conceived to be a self-consistent sequence of steps or instructions leading to a desired result. The steps are those requiring physical manipulations of physical quantities.

**[0025]** Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated in a computer system. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like.

**[0026]** It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the following discussions, it is appreciated that throughout the present invention, discussions utilizing terms such as "processing" or "creating" or "transferring" or "executing" or "determining" or "instructing" or "issuing" or "halting" or "clearing" or "accessing" or "aggregating" or "obtaining" or "selecting" or "calculating" or "measuring" or "querying" or "responding" or "monitoring" or "storing" or "maintaining" or "sending" or "converting" or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system



memories or registers or other such information storage, transmission or display devices.

**[0027]** Referring now to Figure 1, an exemplary system 100 diagram in accordance with one embodiment of the present invention is shown. The system 100 comprises a client computer system 110, a network 120, a server system 190 and a passive monitoring and response system ("passive system") 195. The passive system 195 may be located with or near the server system 190, or it may be located remote to the server system. The server system includes a network fabric component 130, which may be a load balancer and having a network tap 132. The network tap 132 may also be a span port or a mirror port. Herein, "tap" refers to a network tap, a span port and/or a mirror port. The server system also includes a web application program 176 resident on an application server 170. The server system may also include a web server system 160. The server system also includes a DOM event receiver 162. The DOM event message receiver 162 is operable to receive asynchronous DOM event messages 114 from the client 110. The web server 160 and the application server 170 may be referred to as an application tier.

**[0028]** It is appreciated that DOM events generally refer to events that change the DOM. However, "events" referred to herein are not limited to those events that change the DOM. For example, mouse movements, mouse clicks, key strokes, the recorded delay between key strokes, etc. may be DOM events. Moreover, an agent, e.g., a Javascript, operable on the user computer can generate events, e.g., posting event data to the server.

**[0029]** The passive system 195 is communicatively coupled with the server system 190. The passive system 195 includes a converter (packet sniffer) 140 and an event detector (canister) 150. The event detector 150 comprises a rules engine 180. The passive system 195 may monitor and record network traffic. Various methods for monitoring, capturing and encrypting the network traffic flow between the client system and the server system are described in United States Patent Numbers 6,286,098 and 6,286,030. Monitoring and synchronizing the captured network traffic between the client system and the server system are described in United States Patent Number 8,127,000.



**[0030]** In one embodiment of the present invention, in operation, the client 110 can send a page request 112 to the server system 190. For example, the client 110 may request a particular webpage. The page request 112 travels through the network 120. The network fabric 130 receives the page request 112 and forwards the page request 112 to the web server 160 which in turn forwards the page request 112 to the application server 170. The network tap 132 forwards a copy of the page request 112 to the converter 140.

**[0031]** The server system 190 responds to the request by sending a page response 172. For instance, a web application program resident on the application server 170 responds by transmitting the page response 172. The page response 172 may comprise the requested webpage and the agent 174. The page response 172 is transmitted from the application server 170 to the web server 160 and thereafter to the network fabric 130. The network fabric 130 receives the page response 172 and forwards the page response 172 to the client 110 via the network 120. The network tap 132 forwards a copy of the page response 172 to the converter 140.

**[0032]** According to one embodiment, the page response 172 is a webpage along with an agent 174, e.g., a Javascript. The client 110 receives the page response 172 that includes the requested webpage along with the agent 174. The agent may be an AJAX (Asynchronous Javascript Execution) agent in one implementation. In one embodiment, the agent 174 may be loaded onto the webpage when transmitted. The agent 174 is operable within the client browser to monitor and collect various event data, e.g., document object model (DOM) events 114, occurring on the client 110 and is operable to send the event data to the server system 190. Event data is any action performed by the user that changes the DOM on the browser of the client 110. In one example, event data may comprise keyboard key presses, mouse clicks, mouse movements, a

measure of delay in the user entering data in on-screen fields, data entered in the onscreen fields and interaction of an application with the requested webpage, etc.

- [0033]** Importantly, network traffic comprising requests 112 and responses 172 are monitored and collected by the passive system 195. It is appreciated that the network tap 132 supplies network traffic, e.g., page request 112 and page response 172, in a TCP format from the network fabric 130. The network traffic in TCP format 134 are transmitted to the converter (packet sniffer) 140. The converter converts the network traffic from TCP format into an HTTP/HTTPS format and sends the HTTP/HTTPS formatted network traffic 142 to the event detector (canister) 150. In other words, the converter 140 re-assembles packets such that requests and responses between the client 110 and the server system 190 can be recreated and replayed as webpages.
- [0034]** The HTTP/HTTPS formatted network traffic is stored and sessionized in the event detector 150 in accordance with a unique session established for the client 110. It is appreciated that each client may have its own corresponding session stored in the detector 150.
- [0035]** The passive system 195, via the rules engine 180, may determine whether a prescribed action or event has occurred, based on applying a programmable set of rules to the collected network traffic. As a result, an appropriate command, data and/or acknowledgement signal 152 ("command message") may be generated and transmitted to the client system 110. Importantly, the command message 152 is communicated in response to an asynchronous message received from the agent of the client system 110. In this sense, the command message is a "loop back" signal because it is sent responsive to the post signals 114 that originate from the agent 174. The command message 152 causes the agent 174 to take some action on the client side in real-time to alter or enhance the web application program flow in a way that is transparent to the web application program.
- [0036]** According to one embodiment, the agent 174 operating on the client 110 collects event data and communicates the collected event data asynchronously in discrete messages, e.g., DOM event messages 114, and at prescribed time intervals to the server system 190. For example, the agent 174 collects and sends DOM events 114 asynchronously via the network 120 to the network fabric 130. The network fabric 130



forwards the received DOM events 114 to the web server 160 wherein the DOM event receiver 162 receives and detects the DOM events 114. It is appreciated that the discrete messages sent by the agent could also be triggered based on a certain amount of detected events occurring rather than at prescribed time intervals, or a mixture of both may be used. While this event data may or may not be considered by the rules engine of the passive system, the communication is useful because it provides the passive system with an opportunity to communicate with the client system.

**[0037]** The DOM event messages 114 are forwarded to the event detector (canister) 150 of the passive system 195 by the DOM event receiver 162 in one embodiment. The DOM events 114, similar to the HTTP/HTTPS formatted network traffic, may optionally be stored in the event detector 150 in a sessionized fashion. For example, event data, e.g., DOM events 114, may be stored in accordance with a unique session established for the client 110. However, it is appreciated that the network traffic and/or the event data, e.g., DOM events 114, can be processed on the fly without a need to be stored. As will be discussed more fully below, the receipt of an asynchronous DOM event message 114 states the passive system 195 with a mechanism for communicating back to the client system 110 outside of the operation of the web application program 176 by using a command message 152.

**[0038]** Command messages are generated as a result of certain user actions being recognized by the passive system. A plurality of logical rules may be maintained and stored by the event detector 150. The plurality of logical rules may be user programmable and user configurable at the server side typically. The plurality of logical rules may be used by the rules engine 180 to identify specific user events and/or a prescribed action. For example, a plurality of logical rules may be used to determine (by operation of the passive system 195) whether the client 110 needs help or whether the client 110 should be disconnected for security reasons. In response to the determination, a variety of commands and/or data can be sent back to the agent indicating that a value of a cookie should be altered, an active session should be terminated, a chat window should be opened, a DOM of a browser on the client 110 should be altered, a window should be opened and a new uniform resource locator (URL) should be sent, etc.



**[0039]** The rules engine 180 of the event detector 150 applies the plurality of rules maintained in the event detector 150 to the received and stored HTTP/HTTPS formatted network traffic 142, and optionally, the DOM events 114. As a result, the rules engine 180 may generate a command message 152 which is transmitted in response to a received DOM event message 114. Message 152 may include a command, data and/or an acknowledgement signal to be generated. In other words, the rules engine 180 detects a compound event and by applying a set of prescribed rules, maintained in the event detector 150, to the received HTTP/HTTPS formatted network traffic (and optionally the received DOM events 114) for a given client. For example, based on the application of the logical rules, the rules engine 180 may determine that a chat window for the client 110 should be opened. In this case, in response to a received DOM event message 114 the acknowledgment package 152 will include commands and data to be executed by the agent 174 to open the chat window. Alternatively, if no commands and/or data is required, then message 152 will merely be an acknowledgement of the received DOM events from message 114.

**[0040]** The command signal 152 is transmitted from the event detector 150 to the DOM event receiver 162 and forwarded to the client 110 via the network 120. The signal 152 may be transmitted asynchronously and is always sent in response to a received DOM event message 114. Using this loop back communication scheme, the event detector 150 is operable to communicate commands, data and/or acknowledgement signal 152 to the DOM event receiver 162 responsive to identifying a specific user event. As such, the command, data and/or acknowledgment signal 152 is based on user actions of the client system 110 but is sent responsive to a received DOM event message 114. User actions are detected based on the stored network traffic and optionally may also be based on the received DOM events.

**[0041]** It is appreciated that in accordance with embodiments of the present invention the command message 152 may provide additional web content and functionality that is transparent to the operation of the web application. In other words, the actions taken by the agent 174 can alter the program flow of the web application from the user's perspective. But these alterations originate from the detector 150 and not from the web application program. Therefore, embodiments of the present invention provide a mechanism for passively monitoring network traffic and based thereon for altering the apparent flow of a web application at the client system without requiring any changes to



the server based web application program. For instance, situations in which a user needs help can be detected and mechanisms can be communicated to the client for that help, all done in a manner that is transparent to the application program.

**[0042]** It is further appreciated that the event data 114 and command messages 152 are transmitted using an asynchronous channel created by the agent 174 while network traffic comprising page requests 112 and page responses 172 are transmitted using a channel different from the channel created by the agent 174. In either case, the stream of network traffic and event data (optionally including DOM event messages 114) are observed, sessionized and analyzed by the detector 150. Based on the analysis, an appropriate command signal 152 is transmitted via the created channel in order to alter the application flow at the client 110 and to provide additional web content and behavior based thereon, e.g., by opening a window, altering a cookie value, terminating an active session, etc. For instance, a chat window opened on the client 110 may allow a live operator to communicate with the user. The additional web content and functionality may be operable to cause the agent 174 to alter a DOM of a browser of the client 110. In one embodiment, the additional web content and functionality is operable to cause the agent 174 to terminate an active session between a browser of the client 110 and the web application. In one exemplary embodiment, the additional web content and functionality is operable to cause the agent 174 to alter a cookie value of the client browser, e.g., erase a cookie value, used by a browser of the client 110. The additional web content and functionality may also cause a window to open and display the contents of a specified URL at the client side. Thus, the user experience may be enhanced or terminated without substantial alteration of the web application program at the application server 170.

**[0043]** Referring now to Figure 2, an exemplary event detector 150 of the passive system 195 in accordance with one embodiment of the present invention is shown. According to one embodiment, the event detector 150 comprises the rules engine 180, a memory component for maintaining a set of rules 210 and a memory component for storing sessionized database 220 of page request and page responses.

**[0044]** The event detector 150 receives the HTTP/HTTPS formatted network traffic 142 from the converter unit 140. The received HTTP/HTTPS formatted network traffic 142 may be stored in a sessionized database 220 so that the HTTP/HTTPS formatted

network traffic 142 may be stored in accordance with a unique session established for the client 110. Similarly, the received DOM events 114 may optionally be received and stored in the sessionized database 220. According to one alternative embodiment, the storing of the DOM events 114 may also be in accordance with a unique session established for the client 110.

**[0045]** A plurality of logical rules, e.g., Boolean expressions, may be maintained and stored in the set of rules memory component 210. The rules may be complex to determine compound events. It is appreciated that the plurality of logical rules maintained in the set of rules 210 may be programmable and user configurable by a server administrator. As presented and as discussed above, the plurality of logical rules may be used to determine whether a prescribed event has occurred at the client side. For example, the rules may be used to determine whether the client 110 needs help based on application of the rules to the monitored information, e.g., HTTP/HTTPS formatted network traffic 142 (and optionally DOM events 114). For instance, a set of rules may be defined for a user that enters all fields of an entry screen, but presses the "home" icon instead of the "accept" icon. A user exhibiting this behavior may need help completing the transaction. In this case, an operator window may be useful to the user. Alternatively, rules can be established to detect click fraud or other security risks. In such case, the client session can be terminated.

**[0046]** The rules engine 180 applies the rules maintained in the set of rules 210 memory component to the sessionized database 220 in order to detect a prescribed event. For example, application of a set of rules to a sequence of events as stored by the sessionized database 220 may detect a prescribed event, e.g., detect that a user needs help. In response to detecting the prescribed event, the event detector 150 generates a command message 152 that may include a command, data and/or an acknowledgment signal. The generated signal 152 may be transmitted to the client 110 in response to a received DOM event message 114 in order to enhance the client's experience. For example, the command message 152 may cause the agent 174 operating on the client 110 to open a chat window, enabling a live operator to chat with the client 110 in order to resolve an issue encountered.

**[0047]** It is appreciated that in one embodiment the rules engine 180 may apply the logical rules to the HTTP/HTTPS formatted network traffic 142 (and optionally the DOM



events 114) on the fly. In other words, the rules engine 180 may apply the logical rules without a need to store the HTTP/HTTPS formatted network traffic 142 and further without a need to store DOM events 114 in the sessionized database 220.

- [0048]** As presented and discussed above, the generated command message 152 may provide web content and functionality that is in addition to the requested web application. For example, the command message 152 signal may cause the agent 174 operating on the client 110 to open a chat window, clear a cookie value, terminate an active session, alter a DOM of a browser of the client 110, etc. It is appreciated that the command message 152 signal is transmitted to the client asynchronously and in response to the received DOM events 114. The message 152 is forwarded to the DOM event receiver 162. The DOM event receiver 162 forwards the command message 152 to the agent 174 in response to an asynchronous DOM event message 114.
- [0049]** Referring now to Figure 3, an exemplary communication flow 300 between a client system 310, the server system 330 and passive system 195 in accordance with one embodiment of the present invention is shown. According to one embodiment, a browser operating on the client system 310 may send an application request 312 to the server system 330. The application request 312 may be a request for a webpage, for example.
- [0050]** The server system 330 responds by sending a response 314 to the application request. The response 314 may comprise the actual requested application page along with a Javascript agent. The server system 330 may load the Javascript agent to the application prior to transmitting the response 314.
- [0051]** When the client system 310 receives the response, the requested web application, e.g., webpage, may be rendered by the client system 310. The received agent is then operable on the client system 310 for the requested web application, e.g., webpage.
- [0052]** The agent may collect and monitor various information on the client system 310 referred to as event data. Event data may comprise DOM events, as described above. The monitored data and/or DOM events 318 may be sent to the server system 330 asynchronously. It is appreciated that the server system 330 also collects network traffic

information, e.g., requests 312 and responses 314. Based on the network traffic (and optionally on the monitored data and/or DOM events), the passive system 195 may detect a prescribed event. It is appreciated that a lack of event may also be an event, e.g., lack of an action within a prescribed time period.

**[0053]** Based on the prescribed event, the passive system 195 generates and transmits a command, data and/or acknowledgement signal 320 ("command message") to the client system 310 asynchronously and in response to a message 318. As presented above, the generated command message 320 transmitted to the client system 310 provides additional content and functionality that is transparent to the web application program of the server system 330. For example, the command may cause the agent operating on the client system 310 to open a chat window, alter a cookie value, terminate an active session, alter a DOM for the browser operating on the client system 310, etc. If no command and/or data is required, then command message 320 may merely be a simple acknowledgement of the latest DOM event message 318.

**[0054]** Referring now to Figures 4A, 4B, 4C and 4D, an exemplary sequence of events 400 observed on the client system in accordance with one embodiment of the present invention is shown. In this example, the client system 110 has requested a webpage for a particular bank in order to apply for a credit card. The server system 190 responsive to the request, sends the webpage to the client 110. Moreover, an agent is sent along with the webpage to the client.

**[0055]** The client browser receives the webpage and renders the webpage for the user. The webpage may comprise a plurality of fields and icons. For example, a first field 410 may be for the user's first name. A second field 420 may be for the user's last name. A third field 430 may be for a routing number for the user. A fourth field 440 may be an account number and the fifth field 450 may be the user's date of birth. A plurality of functional icons may also be provided. For example, a submit icon 460 may be used to submit the entered information. A cancel 470 icon may be used to cancel any transaction and a home icon 480 may be used to go to the home webpage effectively terminating any transaction before it is completed.

**[0056]** In Figure 4A, the user begins by filling out the first name 410, e.g., typing "John." In Figure 4B, the user continues to fill out the fields by filling out the last name 420, e.g.,



typing "Smith." The user may skip the third field and fill out the fourth field 440 for the account number, e.g., typing "7779542031," as shown in Figure 4C. The user may proceed by requesting to go to a home webpage by pressing on the HOME icon 480.

- [0057]** It is appreciated that the agent operable on the client 110 monitors, collects and transmits the sequence of events occurring on the client 110 in between page requests. For example, the sequence of events that the third field 430 was skipped may be transmitted to the server system 190 as event data, e.g., DOM events 114. Moreover, other event data may include delays between key strokes, mouse movements, mouse clicks, delays between key strokes, overall delay, etc., may similarly be monitored, collected and sent to the server system 190 as DOM events 114. While this event data may or may not be considered by the rules engine of the passive system, it does provide the passive system with an opportunity to communicate with the client system.
- [0058]** The passive system 195 may identify a prescribed action based on the sequence of events, e.g., DOM events 114, collected from the client 110 and the collected network traffic information. For example, if the home button 480 is pressed without the user filling out the required fields, the server system 190 by applying the logical rules to the network traffic may determine that the user is lost and may need help. As a result, the passive system 195 may generate a command signal 152 and transmit it to the client 110 in response to the next DOM event message 114.
- [0059]** The generated command signal 152 may cause the agent operating on the client 110 to provide content in addition to the webpage rendered and/or to perform a particular action. For example, when the passive system 195 identifies that the user is lost and may need help, the home webpage may be rendered on the client 110 since that was the last action taken by the user, as shown in Figure 4D. However, the command 152 may cause the agent to open a chat window 490 (transparent to the web application program) asking the user whether the user would like some additional help by connecting the user to an operator. Window 490 is transparent to the operation of the web application program and as such it does not originate from the web application program. It is appreciated that additional help may automatically be provided to the user by connecting the user to an operator in a separate window without prompting the user whether the user needs help. The user may choose to connect or reject. Thus, the



original application, e.g., original webpage, is not altered but rather additional content, help, etc., is provided to enhance user's experience.

- [0060]** In one embodiment, when the user chooses to connect to an operator, the information entered by the user, e.g., DOM events 114 and the recorded network traffic information for this session may be transferred to the operator such that the operator can better assist the user. Moreover, it is appreciated that the command, data and/or acknowledgment 152 signal may cause the agent to perform other actions. For example, the command may cause the agent to alter a cookie value, open a chat window, send a new URL to the client in a new window, terminate an active session, alter a DOM of a browser of the client 110, etc. As a result, the user experience is enhanced, e.g., the program flow from the user point of view is altered, without substantial alteration to the original web application, e.g., the webpage.
- [0061]** Referring now to Figure 5, an exemplary computer controlled flow diagram 500 in accordance with one embodiment of the present invention is shown. At step 510, a page request 112 from the client 110 is received at the application tier. For example, the page request 112 may be a request for a particular webpage. At step 520, in response to the request 112, the server system 190 responds by sending a page response 172 to the client 110. The page response 172 may comprise the requested webpage and along with the agent 174 that has been loaded onto the webpage by the server system 190.
- [0062]** At step 530, the passive system 195 monitors the network traffic that comprises page requests and page responses. The page requests and page responses may be in TCP format. As such, at step 540, the passive system 195 converts the network traffic from TCP format into HTTP/HTTPS format. In other words, the user's request and responses may be reassembled by converting the TCP format into the HTTP/HTTPS format in order to replay the user experience.
- [0063]** It is appreciated that in one embodiment at step 550, the HTTP/HTTPS formatted network traffic may be stored in accordance with a unique session established for the client 110. Each client may have a unique session and the network traffic for each client may be stored in its corresponding session.

- [0064]** When the response is sent to the client, the agent 174, e.g. a Javascript, begins operating on the browser of the client 110 for the requested webpage. Thus, the agent 174 begins monitoring and collecting event data, e.g., DOM events. According to one embodiment, event data may comprise data regarding any events that occur on the client 110 that alter the DOM maintained on the browser of the client 110. In one example, event data may comprise keyboard key presses, mouse clicks, mouse movements, the recorded delay in key strokes when the user is entering data in on-screen fields, the data entered in the on-screen fields and interaction of an application with the requested webpage, etc. While this event data may or may not be considered by the rules engine of the passive system, it does provide the passive system with an opportunity to communicate with the client system.
- [0065]** Event data, e.g., DOM events, collected and monitored by the agent 174 are sent to the server system 190 in an asynchronous manner. In other words, the event data may be communicated to the server system 190 asynchronously in discrete messages based on prescribed conditions or intervals. It is appreciated that the event data may be communicated from the agent at prescribed intervals that may be user programmable. As such, at step 560, event data is monitored as it is received from the agent operating on the client. It is appreciated that the received event data may also be stored in accordance with a unique session established for the client 110.
- [0066]** At step 570, the passive system 195 maintains a set of rules that define prescribed user events that may need alteration. It is appreciated that the set of rules may be user programmable and user configurable. At step 580, the set of rules may be applied to the received network traffic and/or optionally to the event data to determine whether a prescribed user event has occurred within the client system 110. For example, based on the application of the logical rules to the received network traffic and the received event data, it may be determined that the user needs help. As such, based on the determination an appropriate command, data and/or acknowledgement signal may be generated.
- [0067]** At step 590, the command message is sent to the agent operating on the client 110. It is appreciated that the command, data and/or acknowledgement signal is a message that is communicated asynchronously to the client responsive to the receipt of a message from the agent containing event data.



**[0068]** According to one embodiment, the asynchronous message comprises a command to be executed by the agent in order to provide web content and functionality that is in addition to the page responses sent to the client at step 595. Thus, the web application, e.g., webpage, is not altered. In one exemplary embodiment, the web content and functionality that is provided in addition to the page responses may cause the agent to open a chat window on the client 110 system. The web content and functionality may be operable to cause the agent to alter a DOM of a browser of the client 110, terminate an active session between a browser of the client 110 system and the web application, alter a cookie value, erase a cookie value, etc. Almost any action may be taken by the agent.

**[0069]** As a result, many events occurring on the client 110 are understood by collecting and analyzing network traffic and optionally event data. Thus, web applications may be augmented by providing assistance in real-time, e.g., by opening a chat window without altering the web application program at the server. Moreover, security may be enhanced by enabling the agent on the client 110 to take certain actions, e.g., terminate an active session, alter a cookie value, prevent click fraud, etc. Therefore, the user experience may be enhanced without substantial alteration to the web application program at the server.

#### GENERAL PURPOSE COMPUTER SYSTEM PLATFORM

**[0070]** Figure 6 is a block diagram that illustrates a general purpose computer system 600 upon which embodiments of the invention may be implemented. Computer system 600 may be distributed, one at the client and one at the server to implement a method for a client and a server to communicate as shown in Figures 1-5 and includes a bus 602 or other communication mechanism for communicating information, and a processor 604 coupled with bus 602 for processing information. Computer system 600 also includes a main memory 606, such as a random access memory (RAM) or other dynamic storage device, coupled to bus 602 for storing information and instructions to be executed by processor 604. Main memory 606 also may be used for storing temporary variables or other intermediate information during execution of instructions to be executed by processor 604. Computer system 600 further includes a read only memory (ROM) 608 or other static storage device coupled to bus 602 for storing static information and instructions for processor 604. A non-volatile storage device 610, such

as a magnetic disk or optical disk, is provided and coupled to bus 602 for storing information and instructions and may store the persistent internal queue.

- [0071]** Computer system 600 may be coupled via bus 602 to an optional display 612, such as a cathode ray tube (CRT), a liquid crystal display (LCD), etc., for displaying information to a computer user. An optional input device 614, including alphanumeric and other keys, may be coupled to bus 602 for communicating information and command selections to processor 604. Another type of user input device is cursor control 616, such as a mouse, a trackball, or cursor direction keys for communicating direction information and command selections to processor 604 and for controlling cursor movement on display 612.
- [0072]** The term "computer-readable medium" as used herein refers to any medium that participates in providing instructions to processor 604 for execution. Such a medium may take many forms, including but not limited to, non-volatile media, volatile media, and transmission media. Non-volatile media includes, for example, optical or magnetic disks, such as storage device 610. Volatile media includes dynamic memory, such as main memory 606. Transmission media includes coaxial cables, copper wire and fiber optics, including the wires that comprise bus 602. Transmission media can also take the form of acoustic or light waves, such as those generated during radio wave and infrared data communications.
- [0073]** Common forms of computer-readable media include, for example, a floppy disk, a flexible disk, hard disk, magnetic tape, or any other magnetic medium, a CD-ROM, any other optical medium, punch cards, paper tape, any other physical medium with patterns of holes, a RAM, a PROM, and EPROM, a FLASH-EPROM, any other memory chip or cartridge, a carrier wave as described hereinafter, or any other medium from which a computer can read.
- [0074]** Computer system 600 can send and receive messages through the network(s), network link 620 and communication interface 618. In the Internet example, a server 630 might transmit a requested code for an application program through Internet 628, ISP 626, local network 622 and communication interface 618. The received code may be executed by processor 604 as it is received, and/or stored in storage device 610, or other non-volatile storage for later execution.



**[0075]** In summary, events occurring on the client system are understood by monitoring and collecting network traffic and optionally by monitoring asynchronously transmitted event data. In response to the monitored and collected data, a loop back communication channel may be used to provide assistance in real-time. Moreover, security may be enhanced by enabling the agent on the client system to take certain actions, e.g., terminate an active session, alter a cookie value, etc., to prevent click fraud, for instance. Since these agent actions take place outside the web application program, the user experience may be enhanced or changed without substantial alteration to the web application program.

**[0076]** In the foregoing specification, embodiments of the invention have been described with reference to numerous specific details that may vary from implementation to implementation. Thus, the sole and exclusive indicator of what is, and is intended by the applicants to be, the invention is the set of claims that issue from this application, in the specific form in which such claims issue, including any subsequent correction. Hence, no limitation, element, property, feature, advantage or attribute that is not expressly recited in a claim should limit the scope of such claim in any way. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

## CLAIMS

What is claimed is:

1. A method of communicating with a client system, the method comprising:
  - monitoring and storing network traffic communicated between a web application on a server system and a client system, the network traffic comprising page requests and page responses for a web session over a first transmission channel, wherein the first transmission channel is used by the web application and a client device for conducting the web session;
  - monitoring and storing event data received separately from the network traffic from an agent operating within the client system wherein the event data is received over a second transmission channel;
  - maintaining a set of rules;
  - determining that a prescribed user event occurred within the client system based on a comparison of the set of rules with the network traffic from the web session; and
  - in response to the prescribed user event occurring, sending a command message to the agent over the second transmission channel, in response to determining the user event occurred within the client system wherein the command message instructs the agent to open a new window with a new uniform resource locator (URL) independently of the page requests and the page response for the web session.
2. The method of claim 1 wherein the agent is loaded into one of the page responses.
3. The method of claim 1 further comprising receiving the event data asynchronously, from the network traffic.
4. The method of claim 1 wherein the event data comprises data regarding events occurring within the client system that alters a document object model (DOM) of a browser operating within the client system.



5. The method of claim 1 wherein the new window comprises a chat window.
6. The method of claim 1 wherein the event data comprises one or more of keyboard key presses, mouse clicks, mouse movements, a recorded delay in a user entering data in on-screen fields and data entered in said on-screen fields.
7. An apparatus, comprising:
  - a monitor and response system configured to:
    - receive webpage data from a first transmission channel, wherein the first transmission channel is used by a web application and a client device for conducting a web session;
    - receive event data associated with the webpage data from an agent operating on the client device, wherein the event data is received over a second transmission channel;
    - determine an action for operation on the client device based on the webpage data received from the first transmission channel and the event data received from the second transmission channel; and
    - send a command message to the agent over the second transmission channel, wherein the command message is configured to instruct the agent to open a chat window on the client device.
8. The apparatus of claim 7, wherein the monitor and response system comprises:
  - a converter unit configured to receive the webpage data from the first transmission channel; and
  - an event detector system configured to:
    - receive the webpage data from the converter unit;
    - receive the event data from the second transmission channel;
    - identify user events based on applying a set of prescribed rules to the webpage data and the event data; and
    - send the command message to the agent over the second transmission channel based on the user events.

9. The apparatus of claim 8, wherein:

an application server is configured to operate the web application and transmit the webpage data over the first transmission channel in response to a page request from the client device;

a web server is configured to relay the page request from the client device to the application server and relay the webpage data from the web application to the client device;

a receiver is configured to operate on the web server;

the converter unit is configured to receive a copy of the webpage data from a network tap; and

the event detector is configured to receive the copy of the webpage data from the converter unit, receive the event data from the receiver, and send the command message to the agent through the receiver.

10. The apparatus of claim 7, wherein the monitor and response system is further configured to:

receive the event data from the agent asynchronously from the web application; and

send the command message in response to the event data asynchronously from the web application.

11. The apparatus of claim 7, wherein the command message is configured to initiate opening the chat window transparently from the web application.

12. The apparatus of claim 7, wherein the command message is further configured to cause the agent to alter a value of a cookie on the client device.

13. The apparatus of claim 7, wherein the command message is further configured to cause the agent open a new window with a new uniform resource locator (URL).



14. A computer program product comprising a non-transitory computer readable storage medium having program code embodied thereon, which when executed by a computer directs the computer to:

monitor network traffic for a web session between a web application on a server and a client system, the network traffic comprising page requests sent from the client system to the web application and webpage content received by the client system from the web application in response to the requests for a web session over a first transmission channel, wherein the first transmission channel is used by the web application and a client device for conducting the web session;

detect user inputs associated with the webpage content;

send event data to a monitor and response system in response to detecting the user inputs, wherein the event data is received over a second transmission channel;

receive a command message from the monitor and response system over the second transmission channel in response to sending the event data, wherein the command message is based on a comparison of a set of rules with the network traffic from the web session comprising the event data and the webpage content using the set of rules maintained by the monitor and response system; and

opening a new window with a new uniform resource locator (URL) based on the command message received from the monitor and response system independently of the page requests and the page response for the web session.

15. The computer program product of claim 14, wherein the program code when executed by a computer further directs the computer to:

send the event data to the monitor and response system independently of the requests sent from the client system; and

receive the command message back from the monitor and response system independently from the webpage content received from web application.

16. The computer program product of claim 14, wherein the program code when executed by a computer further directs the computer to:

open the new window transparently from the web application.

17. The computer program product of claim 14, wherein the command message is based on the set of rules applied to the webpage content and the event data identifying a user event during the web session.

18. The computer program product of claim 14, wherein:

the requests and the webpage content are transmitted asynchronously over a first transmission channel; and

the event data and the command message are transmitted asynchronously over a second transmission channel.

19. The computer program product of claim 14, wherein the program code when executed by a computer further directs the computer to:

send the event data to the monitor and response system and receive the command message back from the monitor and response system independently of the network traffic transmitted between the client system and the web application.

20. The computer program product of claim 14, wherein the new window comprises a chat window.



1/9

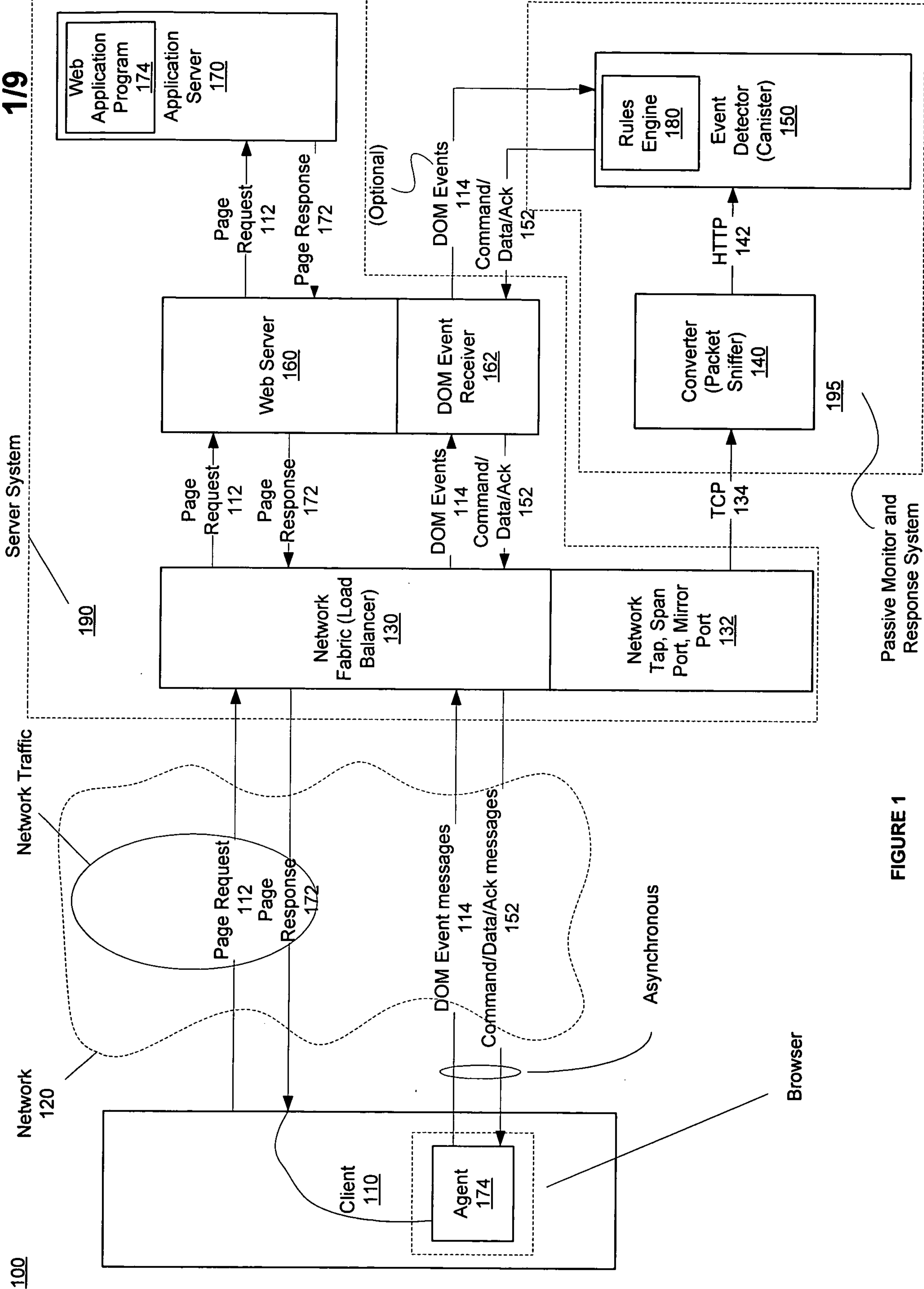


FIGURE 1

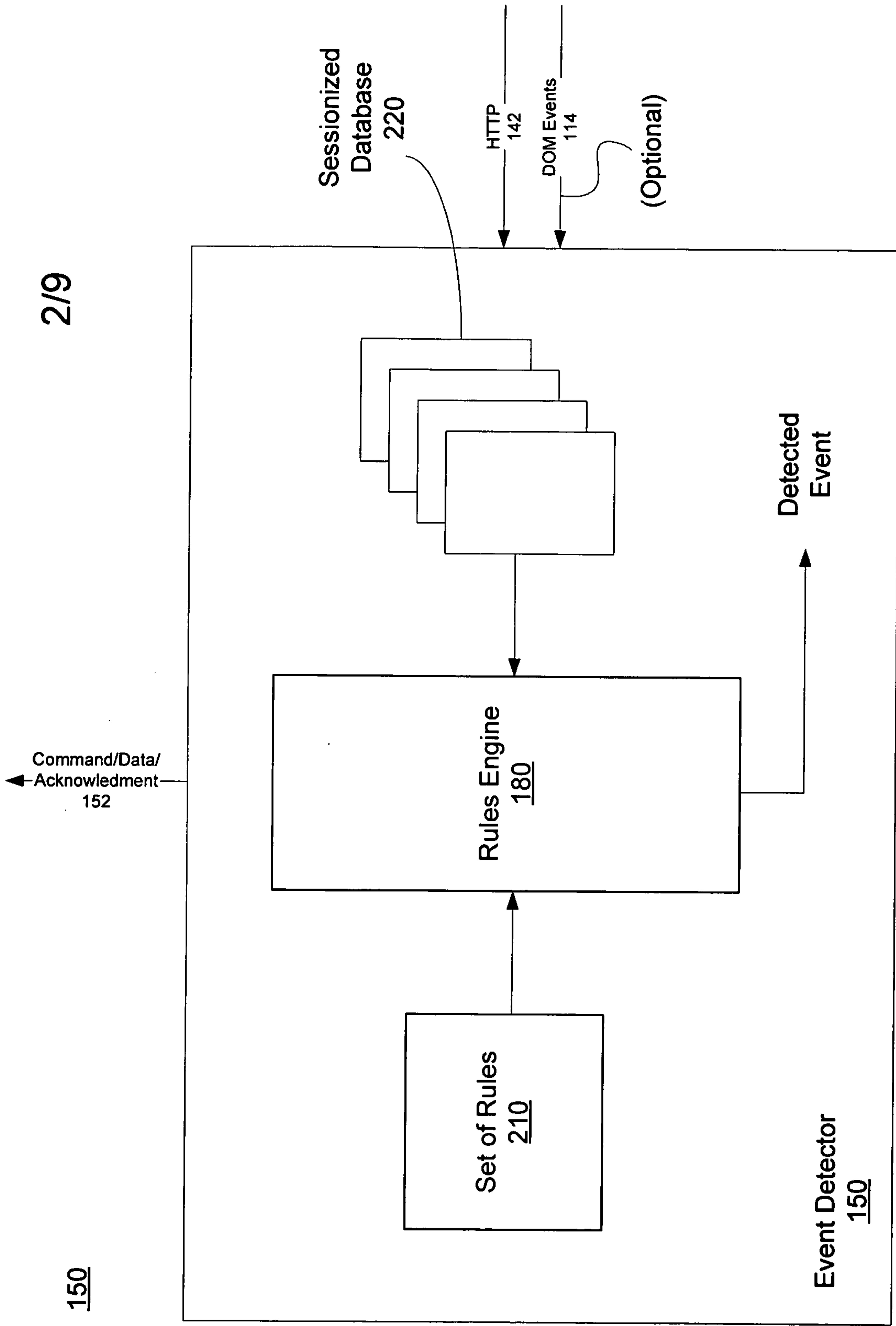


FIGURE 2



300

3/9

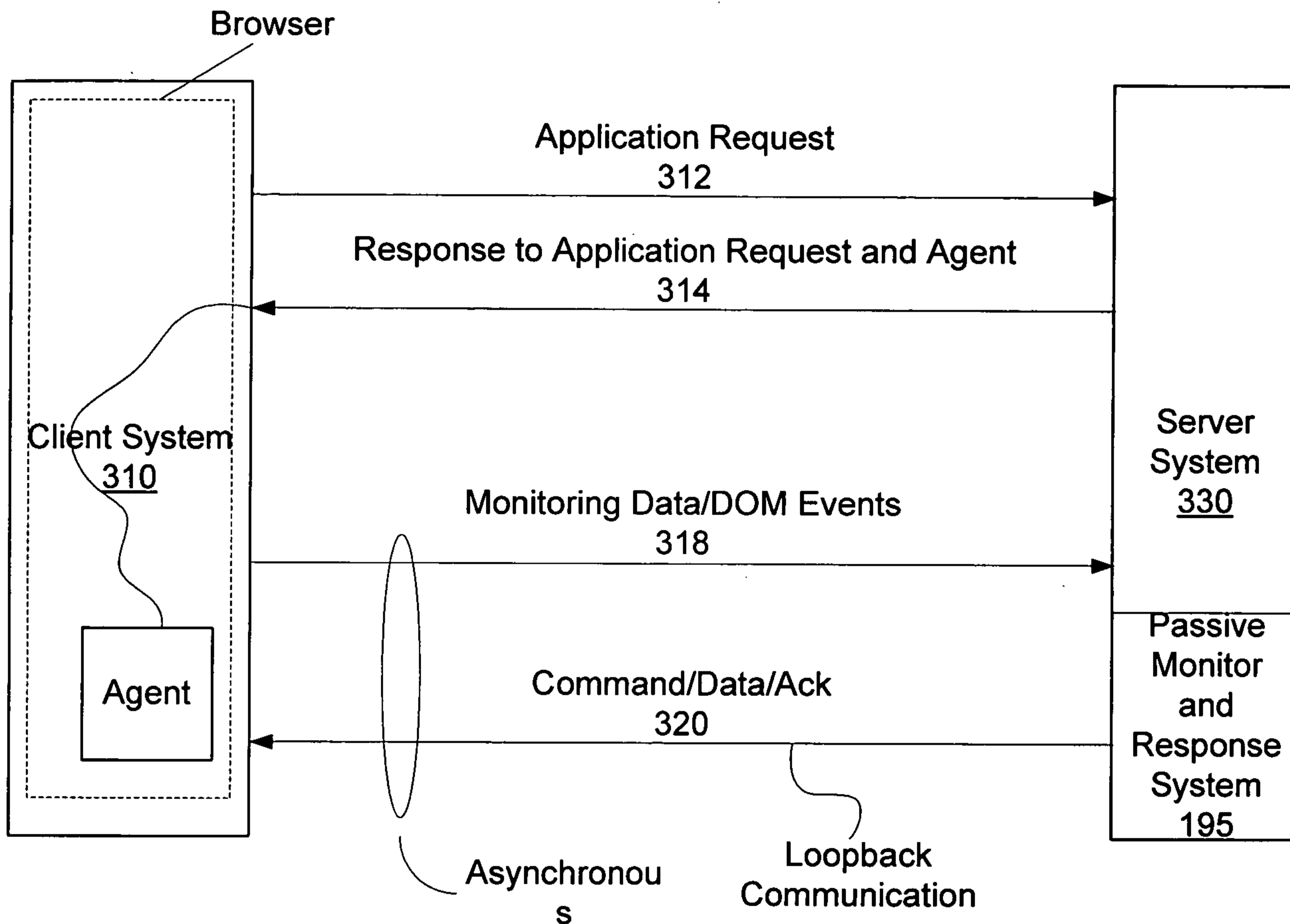


FIGURE 3

4/9

410

400

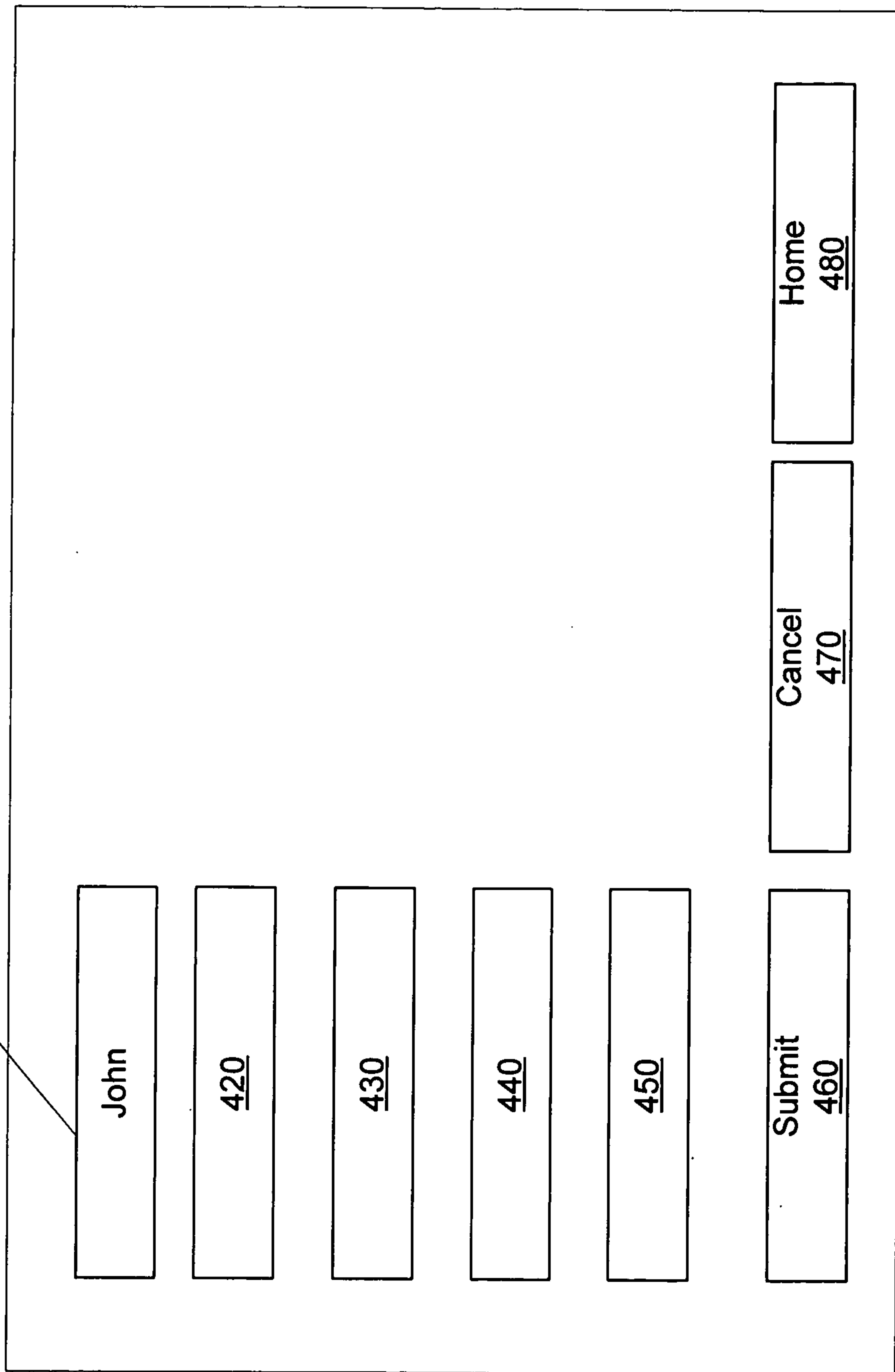


FIGURE 4A



5/9

400

410

420

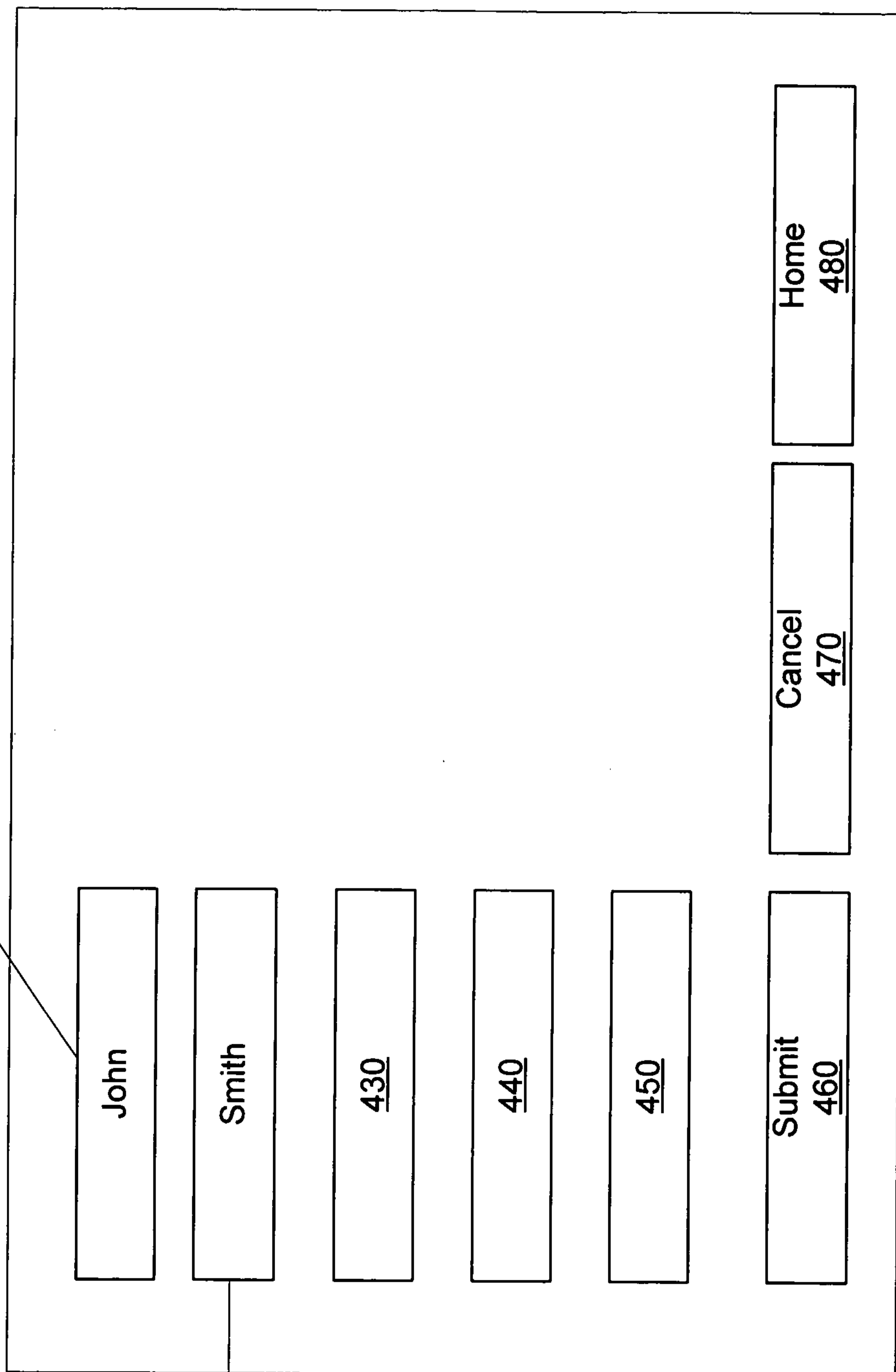


FIGURE 4B

6/9

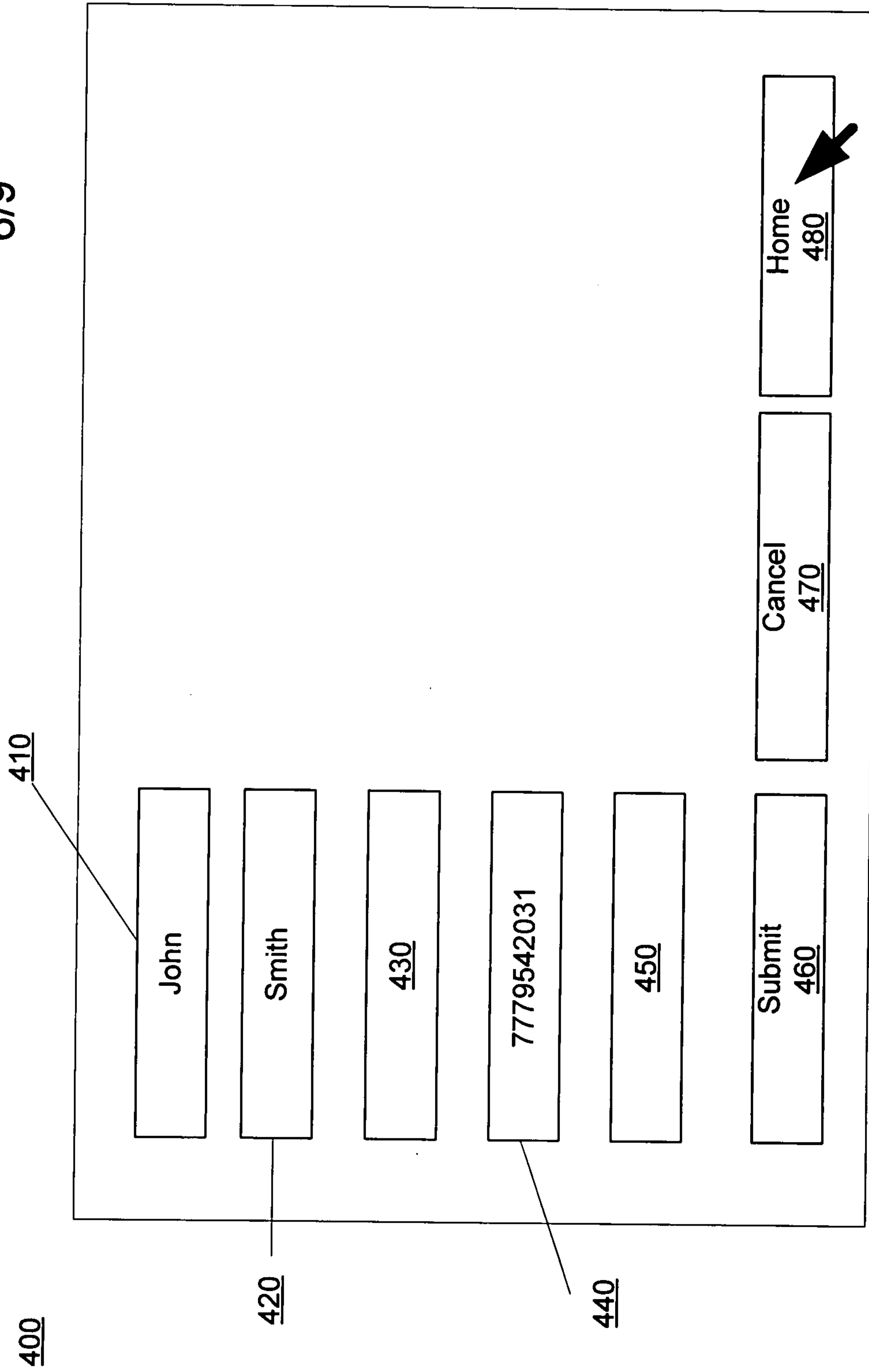


FIGURE 4C



7/9

490

400

Would You Like Help By Chatting Online With One Of Our Representatives?

YES  
NO

Checking Account

Saving Account

Business Loan

Home Loan

Car Loan

Login

Contact Us

Locations

HOME

FIGURE 4D

500

8/9

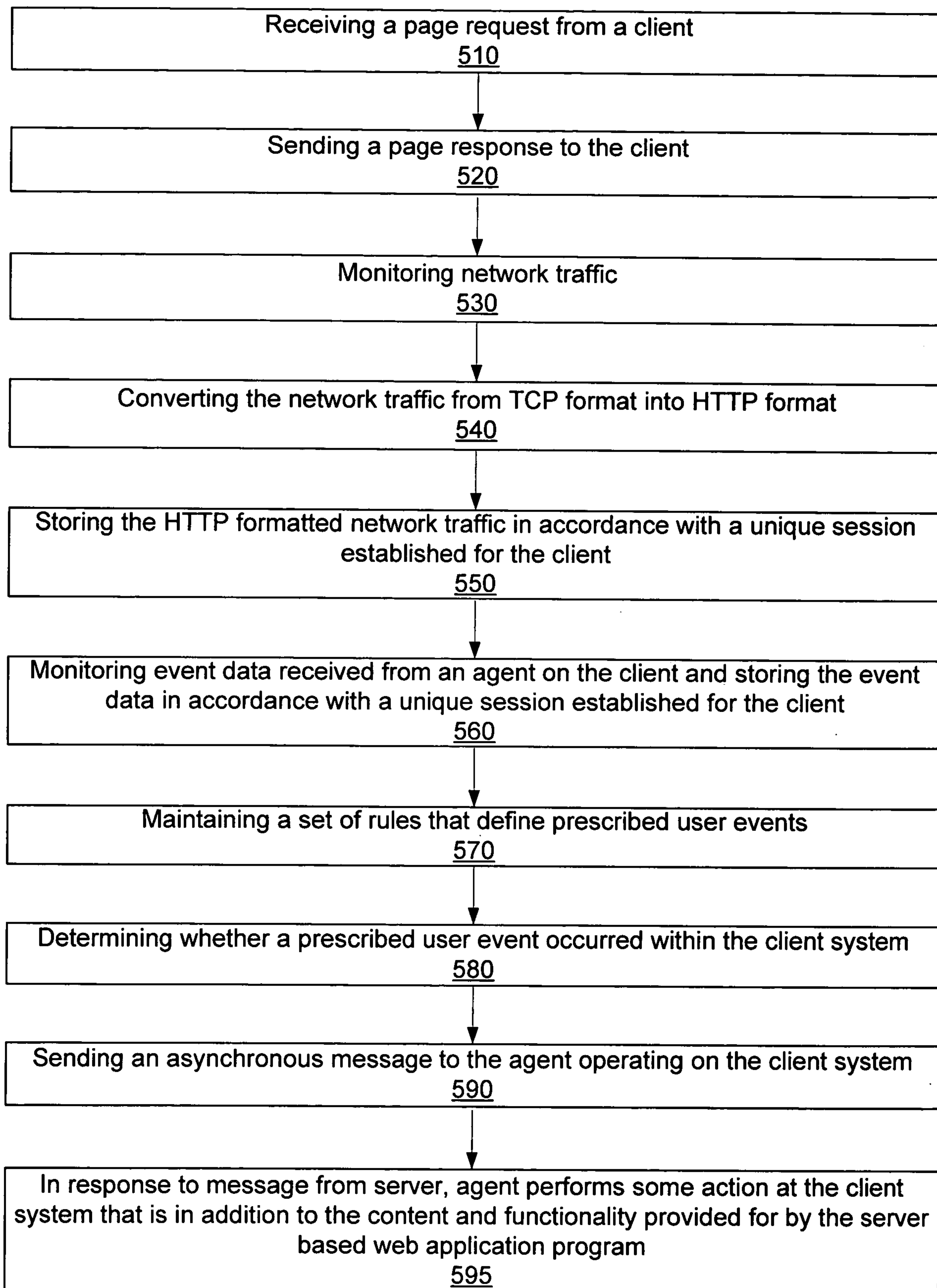


FIGURE 5



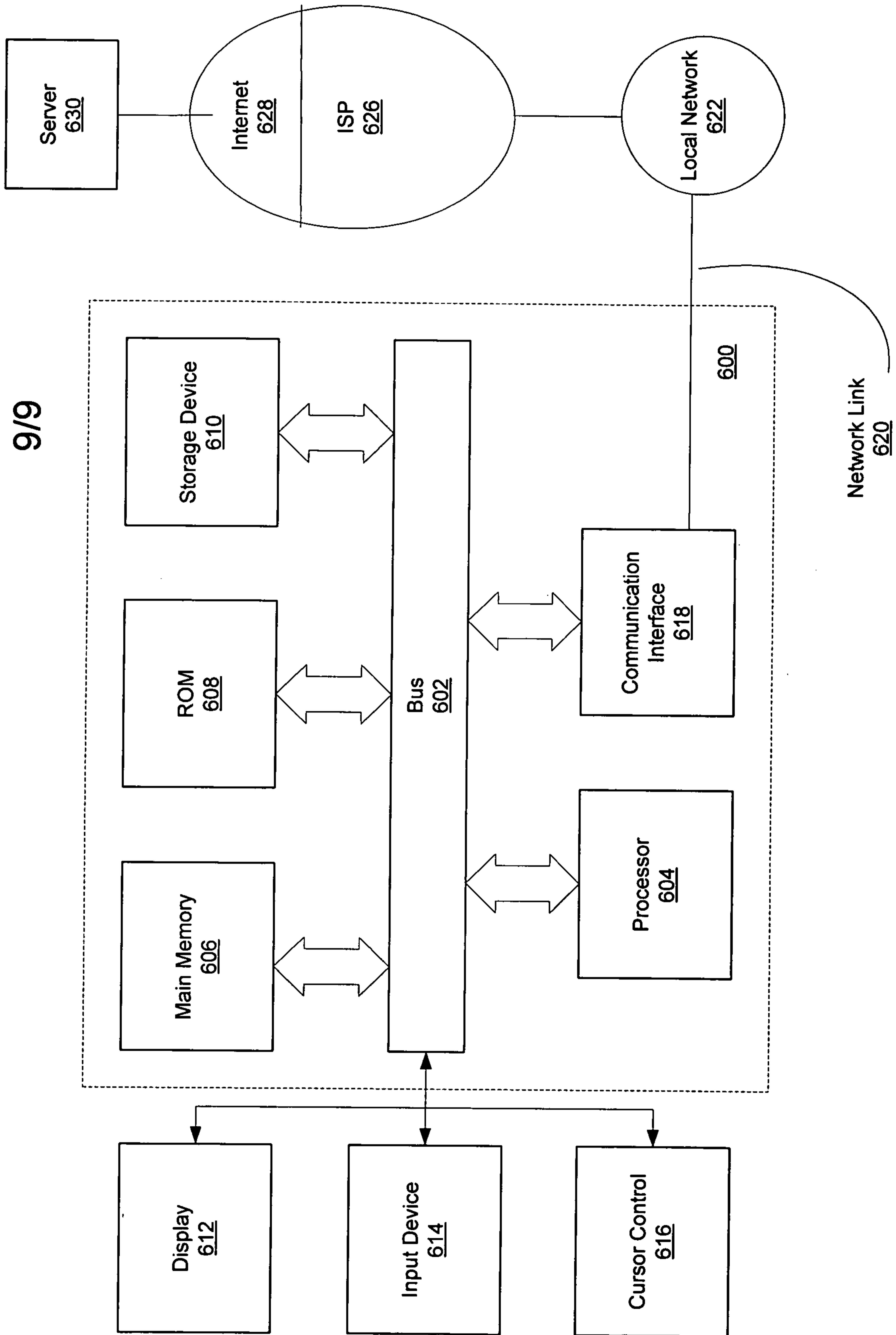


FIGURE 6

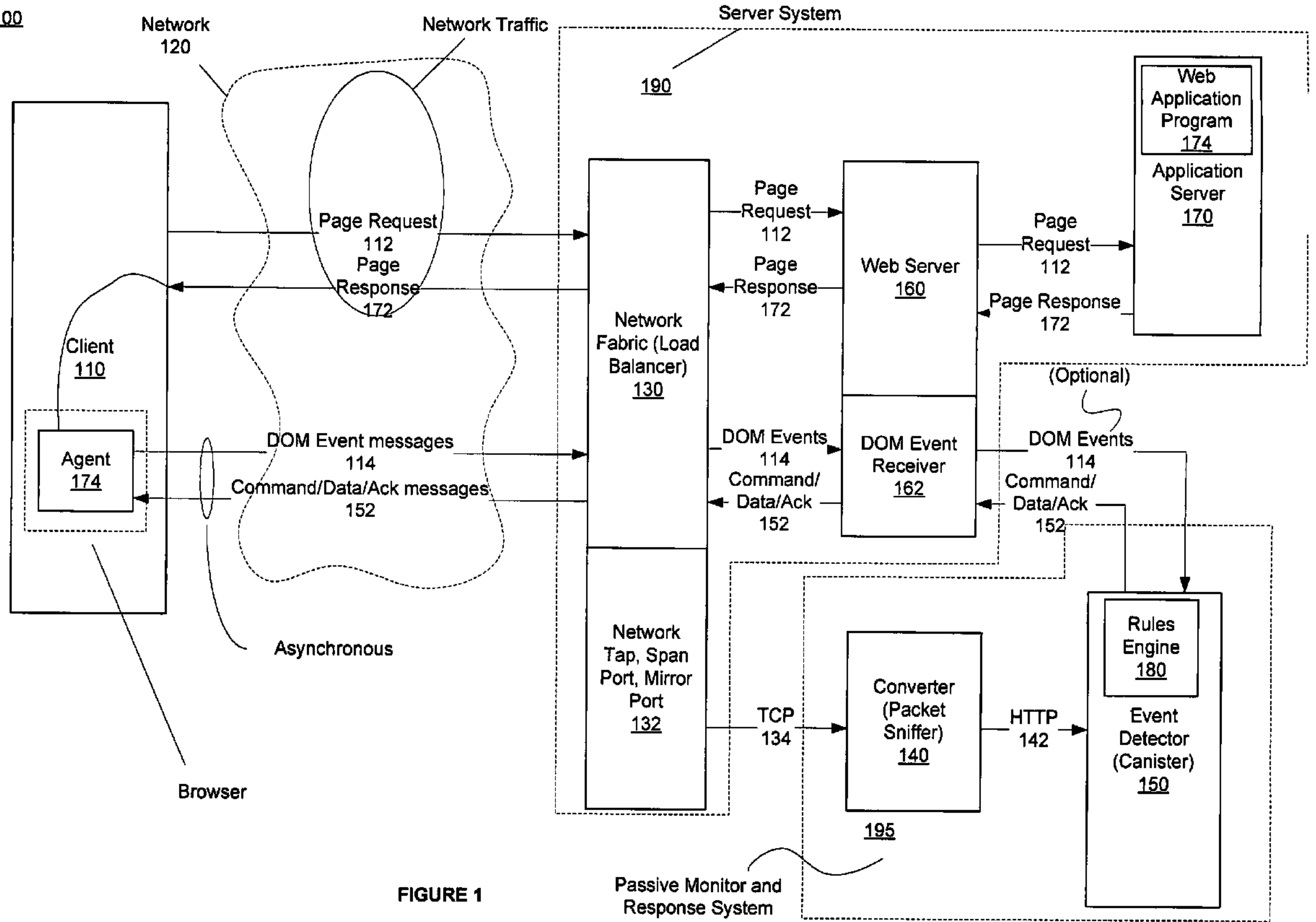


FIGURE 1

Passive Monitor and Response System