



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY. DOCKET NO	TOT CLAIMS	IND CLAIMS
61/096,617	09/12/2008		210	021735-004800US		

CONFIRMATION NO. 5378

20350
TOWNSEND AND TOWNSEND AND CREW, LLP
TWO EMBARCADERO CENTER
EIGHTH FLOOR
SAN FRANCISCO, CA 94111-3834

FILING RECEIPT



OC000000032171726

Date Mailed: 09/24/2008

Receipt is acknowledged of this provisional patent application. It will not be examined for patentability and will become abandoned not later than twelve months after its filing date. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please submit a written request for a Filing Receipt Correction. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections**

Applicant(s)

David Brady, Oakland, CA;

Power of Attorney: The patent practitioners associated with Customer Number 20350

If Required, Foreign Filing License Granted: 09/18/2008

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 61/096,617**

Projected Publication Date: None, application is not eligible for pre-grant publication

Non-Publication Request: No

Early Publication Request: No

Title

METHODS AND SYSTEMS FOR POLLING AN ON DEMAND SERVICE

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at <http://www.uspto.gov/web/offices/pac/doc/general/index.html>.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, <http://www.stopfakes.gov>. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4158).

LICENSE FOR FOREIGN FILING UNDER

Title 35, United States Code, Section 184

Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign Assets Control, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

PROVISIONAL APPLICATION FOR PATENT COVER SHEET — Page 1 of 2

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53(c).

INVENTOR(S)		
Given Name (first and middle [if any])	Family Name or Surname	Residence (City and either State or Foreign Country)
David	Brady	Oakland, CA
<i>Additional inventors are being named on the _____ separately numbered sheets attached hereto</i>		
TITLE OF THE INVENTION (500 characters max)		
METHODS AND SYSTEMS FOR POLLING AN ON DEMAND SERVICE		
<i>Direct all correspondence to:</i> CORRESPONDENCE ADDRESS		
☒ The address corresponding to Customer Number: 20350		
OR ☐ Firm or ☐ Individual Name		
Address		
City	State	Zip
Country	Telephone	Email
ENCLOSED APPLICATION PARTS (check all that apply)		
☒ Application Data Sheet. See 37 CFR 1.76 ☐ CD(s), Number of CDs _____		
☒ Drawing(s) Number of Sheets <u>2</u> ☒ Other (specify) <u>Title Page</u>		
☒ Specification (e.g., description of the invention) Number of Pages <u>23</u>		
Fees Due: Filing fee of \$210 (\$105 for small entity). If the specification and drawings exceed 100 sheets of paper, an application size fee is also due, which is \$260 (\$130 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).		
METHOD OF PAYMENT OF THE FILING FEE AND APPLICATION SIZE FEE FOR THIS PROVISIONAL APPLICATION FOR PATENT		
☐ Applicant claims small entity status. See 37 CFR 1.27.		
☐ A check or money order is enclosed to cover the filing fee and application size fee (if applicable).		
☐ Payment by credit card. Form PTO-2038 is attached.		
☒ The Director is hereby authorized to charge the filing fee and application size fee (if applicable) or credit any overpayment to Deposit		
Account Number: <u>20-1430</u> A duplicative copy of this form is enclosed for fee processing.		
210 TOTAL FEE AMOUNT (\$)		

USE ONLY FOR FILING A PROVISIONAL APPLICATION FOR PATENT

PROVISIONAL APPLICATION COVER SHEET
Page 2 of 2

PTO/SB/16 (10-07)

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.

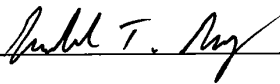
☒ No.

☐ Yes, the name of the U.S. Government agency and the Government contract number are: _____

WARNING:

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to the USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

SIGNATURE _____



Date September 11, 2008

TYPED or PRINTED NAME Gerald T. Gray

REGISTRATION NO. 41,797
(if appropriate)

TELEPHONE 925-472-5000

Docket Number: 021735-004800US

Attorney Docket No.: 021735-004800US
Client Ref.: 129 PROV

PROVISIONAL

PATENT APPLICATION

METHODS AND SYSTEMS FOR
POLLING AN ON DEMAND SERVICE

Inventor: David Brady, a citizen of The United States, residing at
1866 Mountain Blvd.
Oakland, CA 94611

Assignee: salesforce.com, inc.
The Landmark @ One Market, Suite 300
San Francisco, CA 94105

Entity: Large

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, Eighth Floor
San Francisco, California 94111-3834
Tel: 925-472-5000

METHODS AND SYSTEMS FOR POLLING AN ON DEMAND SERVICE

COPYRIGHT NOTICE

5 [0001] A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

BACKGROUND

10 [0002] The present invention relates generally to refreshing displayed data, and more particularly to refreshing data displayed on a page after the data has been changed in the underlying database.

BRIEF DESCRIPTION OF THE DRAWINGS

15 [0003] In the following drawings like reference numbers are used to refer to like elements. Although the following figures depict various examples of the invention, the invention is not limited to the examples depicted in the figures.

 [0004] FIG. 1 illustrates a block diagram of an example of an environment wherein an on-demand database service might be used.

20 [0005] FIG. 2 illustrates a block diagram of an embodiment of elements of FIG. 1 and various possible interconnections between these elements.

DETAILED DESCRIPTION

25 [0006] The present invention provides systems and methods for automatically polling for data changes in an on-demand database service environment.

General Overview

 [0007] Systems and methods are provided for querying for application server changes without incurring the overhead of direct application server cells.

[0008] As used herein, the term *multi-tenant database system* refers to those systems in which various elements of hardware and software of the database system may be shared by one or more customers. For example, a given application server may simultaneously process requests for a great number of customers, and a given database table may store rows for a potentially much greater number of customers. As used herein, the term *query plan* refers to a set of steps used to access information in a database system.

[0009] Next, mechanisms and methods for providing automatic polling in an on-demand service will be described with reference to example embodiments.

System Overview

[0010] FIG.1 illustrates a block diagram of an environment 10 wherein an on-demand database service might be used. Environment 10 may include user systems 12, network 14, system 16, processor system 17, application platform 18, network interface 20, tenant data storage 22, system data storage 24, program code 26, and process space 28. In other embodiments, environment 10 may not have all of the components listed and/or may have other elements instead of, or in addition to, those listed above.

[0011] Environment 10 is an environment in which an on-demand database service exists. User system 12 may be any machine or system that is used by a user to access a database user system. For example, any of user systems 12 can be a handheld computing device, a mobile phone, a laptop computer, a work station, and/or a network of computing devices. As illustrated in FIG. 1 (and in more detail in FIG. 2) user systems 12 might interact via a network 14 with an on-demand database service, which is system 16.

[0012] An on-demand database service, such as system 16, is a database system that is made available to outside users that do not need to necessarily be concerned with building and/or maintaining the database system, but instead may be available for their use when the users need the database system (e.g., on the demand of the users). Some on-demand database services may store information from one or more tenants stored into tables of a common database image to form a multi-tenant database system (MTS). Accordingly, “on-demand database service 16” and “system 16” will be used interchangeably herein. A database image may include one or more database objects. A relational database management system (RDMS) or the equivalent may execute storage and retrieval of information against the database object(s). Application platform 18 may be a framework that allows the applications

of system 16 to run, such as the hardware and/or software, e.g., the operating system. In an embodiment, on-demand database service 16 may include an application platform 18 that enables creation, managing and executing one or more applications developed by the provider of the on-demand database service, users accessing the on-demand database service via user systems 12, or third party application developers accessing the on-demand database service via user systems 12.

[0013] The users of user systems 12 may differ in their respective capacities, and the capacity of a particular user system 12 might be entirely determined by permissions (permission levels) for the current user. For example, where a salesperson is using a particular user system 12 to interact with system 16, that user system has the capacities allotted to that salesperson. However, while an administrator is using that user system to interact with system 16, that user system has the capacities allotted to that administrator. In systems with a hierarchical role model, users at one permission level may have access to applications, data, and database information accessible by a lower permission level user, but may not have access to certain applications, database information, and data accessible by a user at a higher permission level. Thus, different users will have different capabilities with regard to accessing and modifying application and database information, depending on a user's security or permission level.

[0014] Network 14 is any network or combination of networks of devices that communicate with one another. For example, network 14 can be any one or any combination of a LAN (local area network), WAN (wide area network), telephone network, wireless network, point-to-point network, star network, token ring network, hub network, or other appropriate configuration. As the most common type of computer network in current use is a TCP/IP (Transfer Control Protocol and Internet Protocol) network, such as the global internetwork of networks often referred to as the "Internet" with a capital "I," that network will be used in many of the examples herein. However, it should be understood that the networks that the present invention might use are not so limited, although TCP/IP is a frequently implemented protocol.

[0015] User systems 12 might communicate with system 16 using TCP/IP and, at a higher network level, use other common Internet protocols to communicate, such as HTTP, FTP, AFS, WAP, etc. In an example where HTTP is used, user system 12 might include an HTTP client commonly referred to as a "browser" for sending and receiving HTTP messages to and

from an HTTP server at system 16. Such an HTTP server might be implemented as the sole network interface between system 16 and network 14, but other techniques might be used as well or instead. In some implementations, the interface between system 16 and network 14 includes load sharing functionality, such as round-robin HTTP request distributors to balance
5 loads and distribute incoming HTTP requests evenly over a plurality of servers. At least as for the users that are accessing that server, each of the plurality of servers has access to the MTS' data; however, other alternative configurations may be used instead.

[0016] In one embodiment, system 16, shown in FIG. 1, implements a web-based customer relationship management (CRM) system. For example, in one embodiment, system 16
10 includes application servers configured to implement and execute CRM software applications as well as provide related data, code, forms, webpages and other information to and from user systems 12 and to store to, and retrieve from, a database system related data, objects, and Webpage content. With a multi-tenant system, data for multiple tenants may be stored in the same physical database object, however, tenant data typically is arranged so that data of one
15 tenant is kept logically separate from that of other tenants so that one tenant does not have access to another tenant's data, unless such data is expressly shared. In certain embodiments, system 16 implements applications other than, or in addition to, a CRM application. For example, system 16 may provide tenant access to multiple hosted (standard and custom) applications, including a CRM application. User (or third party developer) applications,
20 which may or may not include CRM, may be supported by the application platform 18, which manages creation, storage of the applications into one or more database objects and executing of the applications in a virtual machine in the process space of the system 16.

[0017] One arrangement for elements of system 16 is shown in FIG. 1, including a network interface 20, application platform 18, tenant data storage 22 for tenant data 23, system data
25 storage 24 for system data 25 accessible to system 16 and possibly multiple tenants, program code 26 for implementing various functions of system 16, and a process space 28 for executing MTS system processes and tenant-specific processes, such as running applications as part of an application hosting service. Additional processes that may execute on system 16 include database indexing processes.

30 [0018] Several elements in the system shown in FIG. 1 include conventional, well-known elements that are explained only briefly here. For example, each user system 12 could include a desktop personal computer, workstation, laptop, PDA, cell phone, or any wireless

access protocol (WAP) enabled device or any other computing device capable of interfacing directly or indirectly to the Internet or other network connection. User system 12 typically runs an HTTP client, e.g., a browsing program, such as Microsoft's Internet Explorer browser, Netscape's Navigator browser, Opera's browser, or a WAP-enabled browser in the case of a cell phone, PDA or other wireless device, or the like, allowing a user (e.g., subscriber of the multi-tenant database system) of user system 12 to access, process and view information, pages and applications available to it from system 16 over network 14. Each user system 12 also typically includes one or more user interface devices, such as a keyboard, a mouse, trackball, touch pad, touch screen, pen or the like, for interacting with a graphical user interface (GUI) provided by the browser on a display (e.g., a monitor screen, LCD display, *etc.*) in conjunction with pages, forms, applications and other information provided by system 16 or other systems or servers. For example, the user interface device can be used to access data and applications hosted by system 16, and to perform searches on stored data, and otherwise allow a user to interact with various GUI pages that may be presented to a user. As discussed above, embodiments are suitable for use with the Internet, which refers to a specific global internetwork of networks. However, it should be understood that other networks can be used instead of the Internet, such as an intranet, an extranet, a virtual private network (VPN), a non-TCP/IP based network, any LAN or WAN or the like.

[0019] According to one embodiment, each user system 12 and all of its components are operator configurable using applications, such as a browser, including computer code run using a central processing unit such as an Intel Pentium® processor or the like. Similarly, system 16 (and additional instances of an MTS, where more than one is present) and all of their components might be operator configurable using application(s) including computer code to run using a central processing unit such as processor system 17, which may include an Intel Pentium® processor or the like, and/or multiple processor units. A computer program product embodiment includes a machine-readable storage medium (media) having instructions stored thereon/in which can be used to program a computer to perform any of the processes of the embodiments described herein. Computer code for operating and configuring system 16 to intercommunicate and to process webpages, applications and other data and media content as described herein are preferably downloaded and stored on a hard disk, but the entire program code, or portions thereof, may also be stored in any other volatile or non-volatile memory medium or device as is well known, such as a ROM or RAM, or provided on any media capable of storing program code, such as any type of rotating media

including floppy disks, optical discs, digital versatile disk (DVD), compact disk (CD), microdrive, and magneto-optical disks, and magnetic or optical cards, nanosystems (including molecular memory ICs), or any type of media or device suitable for storing instructions and/or data. Additionally, the entire program code, or portions thereof, may be transmitted and downloaded from a software source over a transmission medium, e.g., over the Internet, or from another server, as is well known, or transmitted over any other conventional network connection as is well known (*e.g.*, extranet, VPN, LAN, *etc.*) using any communication medium and protocols (*e.g.*, TCP/IP, HTTP, HTTPS, Ethernet, *etc.*) as are well known. It will also be appreciated that computer code for implementing embodiments of the present invention can be implemented in any programming language that can be executed on a client system and/or server or server system such as, for example, C, C++, HTML, any other markup language, Java™, JavaScript, ActiveX, any other scripting language, such as VBScript, and many other programming languages as are well known may be used. (Java™ is a trademark of Sun Microsystems, Inc.).

15 **[0020]** According to one embodiment, each system 16 is configured to provide webpages, forms, applications, data and media content to user (client) systems 12 to support the access by user systems 12 as tenants of system 16. As such, system 16 provides security mechanisms to keep each tenant's data separate unless the data is shared. If more than one MTS is used, they may be located in close proximity to one another (*e.g.*, in a server farm located in a single building or campus), or they may be distributed at locations remote from one another (*e.g.*, one or more servers located in city A and one or more servers located in city B). As used herein, each MTS could include one or more logically and/or physically connected servers distributed locally or across one or more geographic locations. Additionally, the term "server" is meant to include a computer system, including processing hardware and process space(s), and an associated storage system and database application (*e.g.*, OODBMS or RDBMS) as is well known in the art. It should also be understood that "server system" and "server" are often used interchangeably herein. Similarly, the database object described herein can be implemented as single databases, a distributed database, a collection of distributed databases, a database with redundant online or offline backups or other redundancies, *etc.*, and might include a distributed database or storage network and associated processing intelligence.

25

30

[0021] FIG. 2 illustrates a block diagram of an embodiment of elements and various possible interconnections between these elements. In the embodiment illustrated by FIG. 2,

one or more middle tier servers 150 exist between system 16 and user systems 12. Middle tier servers 150 are termed middle tier because these servers are interposed between the system 16 and the user systems of a particular organization. As described above, network 14 may be used for communication between system 16 and system 12. In one embodiment, the same network 14 is used between a middle tier servers 150 and user systems 12. In another embodiment, a different network is used between a middle tier server 150 and user systems 12. For example, a tenant network 155_N may be a wireless network, and network 14 may provide communicable coupling via fiber-optics. Each network 14 or tenant network 155_N may also be a combination of different types and protocols.

10 **[0022]** In one embodiment, each middle tier server 150 manages data of a different organization or tenant, however other embodiments may include information of more than one tenant coupled to a single middle tier server. In another embodiment, each middle tier server 150 may contain a plurality of servers, which collectively provide communication between system 16 and user systems 12 of an organization. The tenant network 155 of each organization may be of a different type (e.g. wireless, optical, ...) or protocol. Examples of wireless protocols include Wireless LAN, Global System for Mobile Communications (GSM), Personal Communications Service (PCS), D-AMPS, Wi-Fi, General Packet Radio Service (GPRS), 3G wireless systems such as those using Code division multiple access (CDMA), High Performance Radio LAN (HIPERLAN), and Worldwide Interoperability for

15 Microwave Access (WiMAX).

[0023] Additionally, FIG. 2 further illustrates elements of system 16 and various interconnections. FIG. 2 shows that user system 12 may include processor system 12A, memory system 12B, input system 12C, and output system 12D. As shown in FIG. 2, network 14 couples user systems 12 and system 16. FIG. 2 also shows that system 16 may include tenant data storage 22, tenant data 23, system data storage 24, system data 25, User Interface (UI) 30, Application Program Interface (API) 32, PL/SOQL 34, save routines 36, application setup mechanism 38, applications servers 100₁-100_N, system process space 102, tenant process spaces 104, tenant management process space 110, tenant storage area 112, user storage 114, and application metadata 116. In other embodiments, environment 10 may not have the same elements as those listed above and/or may have other elements instead of, or in addition to, those listed above.

20
25
30

[0024] Regarding user system 12, processor system 12A may be any combination of one or more processors. Memory system 12B may be any combination of one or more memory devices, short term, and/or long term memory. Input system 12C may be any combination of input devices, such as one or more keyboards, mice, trackballs, scanners, cameras, and/or
5 interfaces to networks. Output system 12D may be any combination of output devices, such as one or more monitors, printers, and/or interfaces to networks. As shown by FIG. 2, system 16 may include a network interface 20 (of FIG. 1) implemented as a set of HTTP application servers 100, an application platform 18, tenant data storage 22, and system data storage 24. Also shown is system process space 102, including individual tenant process spaces 104 and
10 a tenant management process space 110. Each application server 100 may be configured to tenant data storage 22 and the tenant data 23 therein, and system data storage 24 and the system data 25 therein to serve requests of user systems 12. The tenant data 23 might be divided into individual tenant storage areas 112, which can be either a physical arrangement and/or a logical arrangement of data. Within each tenant storage area 112, user storage 114
15 and application metadata 116 might be similarly allocated for each user. For example, a copy of a user's most recently used (MRU) items might be stored to user storage 114. Similarly, a copy of MRU items for an entire organization that is a tenant might be stored to tenant storage area 112. A UI 30 provides a user interface and an API 32 provides an application programmer interface to system 16 resident processes to users and/or developers at user
20 systems 12. The tenant data and the system data may be stored in various databases, such as one or more Oracle databases.

[0025] Application platform 18 includes an application setup mechanism 38 that supports application developers' creation and management of applications, which may be saved as metadata into tenant data storage 22 by save routines 36 for execution by subscribers as one
25 or more tenant process spaces 104 managed by tenant management process 110 for example. Invocations to such applications may be coded using PL/SOQL 34 that provides a programming language style interface extension to API 32. A detailed description of some PL/SOQL language embodiments is discussed in commonly owned co-pending U.S. Provisional Patent Application 60/828,192 entitled, PROGRAMMING LANGUAGE
30 METHOD AND SYSTEM FOR EXTENDING APIS TO EXECUTE IN CONJUNCTION WITH DATABASE APIS, by Craig Weissman, filed October 4, 2006, which is incorporated in its entirety herein for all purposes. Invocations to applications may be detected by one or

more system processes, which manages retrieving application metadata 116 for the subscriber making the invocation and executing the metadata as an application in a virtual machine.

[0026] Each application server 100 may be communicably coupled to database systems, e.g., having access to system data 25 and tenant data 23, via a different network connection.

5 For example, one application server 100₁ might be coupled via the network 14 (e.g., the Internet), another application server 100_{N-1} might be coupled via a direct network link, and another application server 100_N might be coupled by yet a different network connection.

Transfer Control Protocol and Internet Protocol (TCP/IP) are typical protocols for communicating between application servers 100 and the database system. However, it will
10 be apparent to one skilled in the art that other transport protocols may be used to optimize the system depending on the network interconnect used.

[0027] In certain embodiments, each application server 100 is configured to handle requests for any user associated with any organization that is a tenant. Because it is desirable to be able to add and remove application servers from the server pool at any time for any
15 reason, there is preferably no server affinity for a user and/or organization to a specific application server 100. In one embodiment, therefore, an interface system implementing a load balancing function (e.g., an F5 Big-IP load balancer) is communicably coupled between the application servers 100 and the user systems 12 to distribute requests to the application servers 100. In one embodiment, the load balancer uses a least connections algorithm to
20 route user requests to the application servers 100. Other examples of load balancing algorithms, such as round robin and observed response time, also can be used. For example, in certain embodiments, three consecutive requests from the same user could hit three different application servers 100, and three requests from different users could hit the same application server 100. In this manner, system 16 is multi-tenant, wherein system 16 handles
25 storage of, and access to, different objects, data and applications across disparate users and organizations.

[0028] As an example of storage, one tenant might be a company that employs a sales force where each salesperson uses system 16 to manage their sales process. Thus, a user might maintain contact data, leads data, customer follow-up data, performance data, goals and
30 progress data, etc., all applicable to that user's personal sales process (e.g., in tenant data storage 22). In an example of a MTS arrangement, since all of the data and the applications to access, view, modify, report, transmit, calculate, etc., can be maintained and accessed by a

user system having nothing more than network access, the user can manage his or her sales efforts and cycles from any of many different user systems. For example, if a salesperson is visiting a customer and the customer has Internet access in their lobby, the salesperson can obtain critical updates as to that customer while waiting for the customer to arrive in the lobby.

[0029] While each user's data might be separate from other users' data regardless of the employers of each user, some data might be organization-wide data shared or accessible by a plurality of users or all of the users for a given organization that is a tenant. Thus, there might be some data structures managed by system 16 that are allocated at the tenant level while other data structures might be managed at the user level. Because an MTS might support multiple tenants including possible competitors, the MTS should have security protocols that keep data, applications, and application use separate. Also, because many tenants may opt for access to an MTS rather than maintain their own system, redundancy, up-time, and backup are additional functions that may be implemented in the MTS. In addition to user-specific data and tenant-specific data, system 16 might also maintain system level data usable by multiple tenants or other data. Such system level data might include industry reports, news, postings, and the like that are sharable among tenants.

[0030] In certain embodiments, user systems 12 (which may be client systems) and/or middle tier servers 150 communicate with application servers 100 to request and update system-level and tenant-level data from system 16 that may require sending one or more queries to tenant data storage 22 and/or system data storage 24. System 16 (e.g., an application server 100 in system 16) automatically generates one or more SQL statements (e.g., one or more SQL queries) that are designed to access the desired information. System data storage 24 may generate query plans to access the requested data from the database.

[0031] Each database can generally be viewed as a collection of objects, such as a set of logical tables, containing data fitted into predefined categories. A "table" is one representation of a data object, and may be used herein to simplify the conceptual description of objects and custom objects according to the present invention. It should be understood that "table" and "object" may be used interchangeably herein. Each table generally contains one or more data categories logically arranged as columns or fields in a viewable schema. Each row or record of a table contains an instance of data for each category defined by the fields. For example, a CRM database may include a table that describes a customer with fields for

basic contact information such as name, address, phone number, fax number, etc. Another table might describe a purchase order, including fields for information such as customer, product, sale price, date, etc. In some multi-tenant database systems, standard entity tables might be provided for use by all tenants. For CRM database applications, such standard
5 entities might include tables for Account, Contact, Lead, and Opportunity data, each containing pre-defined fields. It should be understood that the word "entity" may also be used interchangeably herein with "object" and "table."

[0032] In some multi-tenant database systems, tenants may be allowed to create and store custom objects, or they may be allowed to customize standard entities or objects, for example
10 by creating custom fields for standard objects, including custom index fields. U.S. Patent application Serial No. 10/817,161, filed April 2, 2004, entitled "Custom Entities and Fields in a Multi-Tenant Database System," and which is hereby incorporated herein by reference, teaches systems and methods for creating custom objects as well as customizing standard
15 entity data rows are stored in a single multi-tenant physical table, which may contain multiple logical tables per organization. It is transparent to customers that their multiple "tables" are in fact stored in one large table or that their data may be stored in the same table as the data of other customers.

[0033] The following detailed description will describe polling in accordance with aspects
20 and embodiments of the present invention.

Polling Overview

[0034] It is desirable that a page that displays task/event data (task/event page), or other data, automatically refresh within a reasonable amount of time after task/event data for that user has changed. (In certain aspects, change means that an event or task was created,
25 modified, or deleted. Changes can be made locally - i.e., the user uses a UI on their workstation to change events, or non-locally - i.e., task/event changes are made elsewhere.

[0035] Local activity might include:

1. User A changes their own task/events through normal Salesforce pages.
- 30 2. User A's events changed through newer UI methods - drag and drop.

[0036] Non-local activity might include:

1. User B changes user A's task/events.
2. User A's events changed through workflow/triggers.
3. User A's events changed through API.

5 [0037] There is an expectation that changes made to a user's tasks/events show up in a reasonable amount of time in the task/event page. The source of the change influences what "reasonable" means. For changes made through local activity on their workstation, reasonable may be a few seconds. For non-local change sources, reasonable may be a few seconds or a few minutes.

Current Solution

10 [0038] The task/event page refreshes itself every 15 minutes. Any change to task/events - either local or non-local - will be reflected in 15 minute intervals. This is also the sole means of displaying non-local changes.

[0039] If local activity indicates that an event/task has changed, a process is used to cause the task/event page to refresh itself. This allows nearly immediate reflection of changes
15 when a user changes their own task/events through normal Salesforce pages, but does not address any of the user change sources.

[0040] However, there are numerous issues with this approach:

- The page refresh, even at the 15 minute intervals, is extremely expensive and will not be supportable as ML users ramp up.
- 20 • The local activity process is fragile, and only supports one of the potential sources of task/event changes.
- A 15 minute interval for refreshes is too coarse for active use.

Polling Service Detailed Description

25 [0041] In one embodiment, the task/event page is enabled to ask a service if the task/event page should refresh itself. This is the "polling service." From the task/event page - the page asks the polling service periodically (e.g., every 15 seconds, or every 2 to 3 minutes) if the task/event page should be refreshed, and refreshes only when needed. The frequency of this request allows a "reasonable" delay in showing up-to-date data.

[0042] Given the amount of polling - users*polls per minute - direct use of an application server is generally ruled out as direct application server accesses include unavoidable database use, and this service could quickly consume all available database connections.

Major components

5 **Managed Polling Package**

Polling Client - e.g., javascript poller

Polling Service - responds to the polling client, and coordinates bulk access to the polling table.

Managed Polling Package

10 [0043] In one embodiment, a Polling Package includes:

Custom Polling Table (org, user, poll_type, create_date)

Workflow rule and Outbound Messaging

[0044] Triggers against tasks/events will upsert a new polling for the impacted user with poll_type="task_event_refresh" and create_date=current timestamp. In one embodiment, an
15 insert or update into this table triggers a workflow rule, which will cause an outbound message to be sent to a Polling Service endpoint.

Polling Client

[0045] The task/event page includes a poller, that runs periodically (e.g., every minute this time is tunable). The poller calls the polling service with the poll_type (e.g.
20 "task_event_refresh"), and session token. The polling service can return multiple responses.

- HTTP UNAUTHORIZED - the credentials supplied are invalid (or the service believes the poller to be abusive.) The poller should stop making requests.
- HTTP OK - the polling row id is returned. The client must keep track of the last few ids returned and ignore ids that it has seen before. If the id is new, the polling client
25 should cause the page to be refreshed.
- HTTP NOT MODIFIED - no new polling data.
- HTTP (something) - the service returns the maximum polling frequency - only returned once if the service believes the client is calling too often.

Polling Service

30 [0046] The polling service responds to polling requests. In certain aspects, the polling service is a stand-alone servlet, with minimal filters and little or no standard context.

Polling Service - Authentication

[0047] The first time a session token is seen the session is authenticated through the api; a child session is created for polling and a long timeout - e.g., 8 hours is set and the user/org is noted for the session. If authentication fails HTTP UNAUTHORIZED is returned. (Or
5 HTTP NOT MODIFIED - Polls from this child session will continue working until the session timeout - which will force a new authentication request.

Polling Service - Polling Client Requests

[0048] In one embodiment, each polling client request includes a session token (cookie), and poll_type. The service checks for org/user/poll_types in the in-memory polling data. If
10 data is found, and it is newer than the time of the previous polling request (minus some delta to allow for timing slop) - HTTP OK and the related polling id is returned, and the data is marked as consumed. Otherwise, HTTP NOT MODIFIED is returned.

[0049] Keeping this data in-memory is completely feasible -
(id(15)+org(15)+user(15)+poll_type(25)+create_date(8))*32000 = 174,720,000bytes - and
15 that's worse case. The ML model indicates 7 event/task changes a day per user (221200 total)- so in any 1 minute period over a 10 hour day (for example), and assuming an event distribution - there will be approximately 368 polling rows in-memory. If the window for polling data is 3 minutes - that is, get the last 3 minutes worth of data - a bit over 1000 rows will be produced.

20 [0050] If the current request is found to be too soon after the last request, the server responds with HTTP (something) and the minimum frequency. This same response can be used for all requests if we need to (for whatever reason) slow down polling clients.

Polling Service - Feeding

[0051] The polling service exposes a simple webservice endpoint. This endpoint accepts
25 org/user/poll_type, and uses this information to keep an in-memory list of recent polls.

Polling Service - Load Balancing

[0052] Outbound messages sent to a polling service will be propagated to other polling services. A few approaches include:

- Use Apache JCS - <http://jakarta.apache.org/jcs/> - configured with:
30 o Memory: LRU Memory Cache?

- Lateral: TCP Lateral Cache - <http://jakarta.apache.org/jcs/LateralTCPAuxCache.html> - allows service/servers to be kept in sync
- 5 • Use Jboss Cache 2.0 - http://labs.jboss.com/file-access/default/members/jboss-cache/freezone/docs/2.1.0.CR1/userguide_en/html_single/index.html#d0e138
 - Each service/server configured as a cluster.

Polling Service - Initial State

10 [0053] When a server starts up and discovers that it does not have polling data it will query the polling table, through the api, for all rows.

Overview of Pollmatic

[0054] Pollmatic allows a page (e.g., Visualforce page) to send repeated queries for application server changes without incurring the overhead of a direct application server call. Pollmatic can be used, for example, to force an automatic refresh of a page that shows events
15 when an event is changed on the app server.

[0055] Using Pollmatic requires minor configuration on an org and integrating the Pollmatic IFrame into a Visualforce page.

ADDING POLLMATIC TO A VISUALFORCE PAGE – POLLMATIC IFRAME

20 [0056] The Pollmatic IFrame does most of the work; in one embodiment it includes a robust Javascript polling mechanism that continually asks the Pollmatic service if something has changed on the application server. In certain aspects, the IFrame requires a specific URL and parameters to work:

- Server URL: `://apps.salesforce.com /pollmatic/iframe.poll`
- Parameters:
- 25 • `sessionId={!$Api.Session_ID}`
 - The user's session
 - `apiurl = {!$Api.Partner_Server_URL_90}`
 - Server URL for partner API 9.0
 - `pollingInterval=50000`
 - 30 ○ The interval between poll requests is in milliseconds. (TBD: The minimum polling interval is five minutes.)
 - `notice=<yournoticetype>`

- The notice used here must match the notice specified when configuring an org to use Pollmatic. If integrating Pollmatic into a page that shows calendar info, a notice could be *notice=calrefresh*.

The following is an example of an Iframe component:

```
5 <apex:iframe id="pollmatic"
  src="http://apps.salesforce.com/pollmatic/iframe.poll?pollingInterval=1000&sessionId={!$Api.Session_ID}&notice=calrefresh&apiurl={!$Api.Partner_Server_URL_90}"/>
```

ADDING POLLMATIC TO A VISUALFORCE PAGE – JAVASCRIPT CALLBACK

[0057] When the Pollmatic Iframe gets new information via polling, the Iframe calls a Javascript method. This method could do a simple page refresh; take advantage of `apex:actionFunction` to re-render part of a page or do anything else that Javascript enables.

[0058] In certain aspects, the following Javascript needs to be added to a page:

- `document.domain="salesforce.com"`
 - The Pollmatic service runs on a different server than the application server, and it must reset your document's domain to the less specific "salesforce.com" to allow the Pollmatic Iframe to call into your page.
- `function gotInfo(result) { <do something interesting> }`
 - The Pollmatic Iframe calls this function when your polling notice has been updated.

[0059] For example:

```
25 <apex:page controller="pollmaticTestPageController">
  <apex:iframe id="poller" src="//apps.salesforce.com
    /pollmatic/iframedebg.poll?pollingInterval=1000&sessionId={!$Api.Session_ID}&notice=calrefresh&apiurl={!$Api.Partner_Server_URL_90}"/>

  <script>
    document.domain="salesforce.com";
    var lastResult = 0;
    function gotInfo(result) {
      callActionFunction();
    }
  </script>

  <apex:form id="output">
    <apex:actionFunction action="{!pinged}" rerender="output"
      name="callActionFunction"/>
  </apex:form>

  </apex:page>
```

ADDING POLLMATIC TO A VISUALFORCE PAGE – AUTHENTICATION

[0060] The request for the `iframe.poll` page is authenticated with the passed in `sessionId`, against the passed in `apiurl`. The session in Pollmatic can be in one of three states after authentication:

- 5 1. *Authenticated* - The session was successfully authenticated, and will not be checked again for four hours. After four hours, the session will be authenticated again. This means that if a session is authenticated in Pollmatic, and within those four hours, the application server session is made invalid (through logging out, presumably), that Pollmatic will continue to accept poll requests from the
- 10 Pollmatic IFrame.
2. *Unauthenticated* - The session was not authenticated. The session will never be checked again.
3. *Uncheckable* - The session could not be checked due to various issues: a network issue, the API server is unavailable, org rate limits have been exceeded, or other
- 15 issues. This session is authenticated again within 15 to 30 (TBD) minutes. The re-check time is random so that the API server does not get hit with multiple, simultaneous authentication requests when the API server becomes available again.

[0061] The Pollmatic IFrame makes polling requests to `check.poll`. These requests are

20 authenticated in the same manner as `iframe.poll`.

ADDING POLLMATIC TO A VISUALFORCE PAGE – POLLING NUANCES

[0062] Polling requests will not start for ten seconds after page load. If requests to `check.poll` are not answered in a timely manner (say, the Pollmatic service is down), polling

25 requests are stopped for a period of time defined as two times your polling interval, and then restarted.

ADDING POLLMATIC TO A VISUALFORCE PAGE – DEBUGGING

[0063] A second Pollmatic IFrame page includes multiple debugging callbacks. These are not required in the hosting page.

- 30 • `function requestInfoHandler(request) { }` – Called when a request is made to the `check.poll`; request is an `XMLHttpRequest` object.
- `Function responseInfoHandler(response) { }` - Called when a response is received from the `check.poll` request; response is an `XMLHttpRequestResponse` object.
- `Function requestAbortedHandler() { }` – Called when a request is aborted.
- 35 • `Function debugHandler() { }` –

[0064] For convenience, a second iframe is available: `iframedebg.poll`. This iframe includes visible debugging information.

```
5 <apex:page controller="pollmaticTestPageController">
  <apex:iframe id="poller" src="http://apps.salesforce.com
  /pollmatic/iframedebg.poll?pollingInterval=50000&sessionId={!$Api.Session_ID}&noti
  ce=calrefresh&apiurl={!$Api.Partner_Server_URL_90}"/>

  <script>
    document.domain="salesforce.com";
    var lastResult = 0;
    function gotInfo(result) {
      callActionFunction();
    }
  </script>

  <apex:form id="output">
    <apex:actionFunction action="{!pinged}" rerender="output"
    name="callActionFunction"/>
  </apex:form>

  </apex:page>
```

TELLING POLLMATIC ABOUT APPLICATION SERVER CHANGES

[0065] The Pollmatic service gathers information about polling notices in two ways:

1. When a session is authenticated, Salesforce will check to see if the organization has been seen by the service since the service was started. If not, all polling information from the application server for the org within 30 seconds is gathered.
2. The Pollmatic service accepts outbound messages from the application server describing polling information for a single user.

[0066] In certain aspects, pollmatic setup requires a few steps:

1. Create a custom object, Poll Notice.
2. Create an “always on” workflow rule against all Poll Notice changes.
3. Create an Outbound Message action, triggered by the workflow rule.
4. Finally, create a trigger which updates Poll Notice.

TELLING POLLMATIC ABOUT APPLICATION SERVER CHANGES –POLL NOTICE CUSTOM OBJECT

[0067] In certain aspects, the Pollmatic service requires a custom object called Poll Notice.

[0068] The object must be named “Poll_Notice” and have the following custom fields:

- User - Lookup(User) - The user that should be alerted to changes.
- Date – Date/time – The date/time that the change took place.
- Notice – Text(25) – The notice type. For more information, *see Adding Pollmatic to a Visualforce Page – Pollmatic Iframe*.

[0069] Rows inserted into this table should ensure that there is only one row per User/Notice combination. That is, if a row is inserted for user A and notice “calrefresh,” you should only update that row for future user A and notice “calrefresh” date changes. This object must be secured to allow any user API access to all rows. The first time an organization uses the polling service, the pollmatic service selects all rows from this organization’s object.

TELLING POLLMATIC ABOUT APPLICATION SERVER CHANGES – OUTBOUND MESSAGES

[0070] The Pollmatic service accepts outbound messages that include the user, date, and notice. The service keeps track of these messages, and uses them to let the pollmatic Iframe know that a notice has arrived.

[0071] In certain aspects, the Outbound Message is defined as:

- Name: Poll Notice Modified
- Object: Poll_notice__c
- Fields: id, date__c, user__c, notice__c
- 5 • Endpoint URL: <http://apps.salesforce.com/pollmatic/set.poll>

TELLING POLLMATIC ABOUT APPLICATION SERVER CHANGES – WORKFLOW

[0072] All changes to the Poll Notice object should trigger workflow.

- Object: Poll_notice__c
- 10 • Rule Criteria: if(true, true, true)
- Immediate Workflow Action: Outbound Message, Poll Notice Modified

TELLING POLLMATIC ABOUT APPLICATION SERVER CHANGES – TRIGGERS

- [0073] Triggers should be written to update the Poll Notice object. Here's an example that
- 15 will react to changes to an Event. Updates and inserts to the Poll Notice table should be designed to avoid duplicate notice values for a user.

```
trigger PollNotification on Event (after insert,after update,before delete) {  
20     Set<String> impactedUsers = new Set<String>();  
  
    if (trigger.isdelete){  
        for (Event e : Trigger.old) {  
            String u = e.OwnerId;  
25            if (u.substring(0,3)=='005') {  
                impactedUsers.add(u);  
            }  
        }  
    } else {  
30        for (Event e : Trigger.new) {  
            String u = e.OwnerId;  
            if (u.substring(0,3)=='005') {  
                impactedUsers.add(u);  
            }  
35        }  
    }  
  
    List<Poll_Notice__c> notices =  
40    [select user__c, notice__c, date__c from poll_notice__c  
        where user__c in :impactedUsers  
        and notice__c='calrefresh'];  
  
    for(Poll_Notice__c notice : notices) {
```

```

    impactedUsers.remove(notice.user__c);
    notice.date__c = System.now();
}
5    for(String user : impactedUsers) {
        notices.add(new Poll_Notice__c(user__c=user, notice__c='calrefresh',
date__c=System.now()));
    }
10    upsert(notices);
}

```

TELLING POLLMATIC ABOUT APPLICATION SERVER CHANGES – DIRECT WORKFLOW

15 **[0074]** It is also possible to tell Pollmatic about application server changes without the use of the Poll Notice object. This could minimize the executed DML statements during, say, an Event or Task change, but does have some major caveats:

1. Workflow does not support deletion of objects. Therefore, if an object is deleted you can not send a notification to the poll notice service.
- 20 2. The Pollmatic service will not be aware workflow/outbound messages that were sent prior to startup.

[0075] The endpoint url for direct workflow messages is slightly different:

1. Uses wfset.poll (instead of set.poll).
2. The url must indicate which fields correspond to the date and user entries in the message.

[0076] Using Event as an example:

- Outbound Message Name – suggest Event Modified
- Endpoint URL:
<http://apps.salesforce.com/pollmatic/wfset.poll?user=OwnerId¬ice=calrefresh&date=LastModifiedDate>
 - 30 ○ User=OwnerId – this tells the Pollmatic service that the OwnerId parameter is to be used to identify the affected user.
 - Date=LastModifiedDate – this tells the Pollmatic service that LastModifiedDate should be used to identify the time of the change.
 - 35 ○ Notice=calrefresh – this *notice* is the same as the notice you specified in the iframe url.
- Include 3 fields:
 - Id

- OwnerId
- LastModifiedDate

[0077] You'll also need to set up the workflow rule. Again, using Event, as an example:

- 5 • Object: Task or Event
- Rule Criteria: if(true, true, true)
- Immediate Workflow Action: Outbound Message – and either the Task Modified or Event Modified rule as appropriate.

10 **[0078]** While the invention has been described by way of example and in terms of the specific embodiments, it is to be understood that the invention is not limited to the disclosed embodiments. To the contrary, it is intended to cover various modifications and similar arrangements as would be apparent to those skilled in the art. Therefore, the scope of the appended claims should be accorded the broadest interpretation so as to encompass all such

15 modifications and similar arrangements.

WHAT IS CLAIMED IS:

- 1 1. A method for polling an on-demand service, comprising:
2 sending a poll including a poll_type and a session token to an on-demand
3 service; and
4 waiting for the on-demand service to respond.
- 1 2. The method of claim 1, further comprising receiving a response from
2 the on-demand service.
- 1 3. The method of claim 1, wherein the response includes a return code
2 including at least one selected from:
3 • HTTP UNAUTHORIZED;
4 • HTTP OK;
5 • HTTP NOT MODIFIED; and
6 • HTTP (something).
- 1 4. A method for responding to a poll at an on demand service,
2 comprising:
3 returning a return code including at least one selected from:
4 • HTTP UNAUTHORIZED;
5 • HTTP OK;
6 • HTTP NOT MODIFIED; and
7 • HTTP (something).

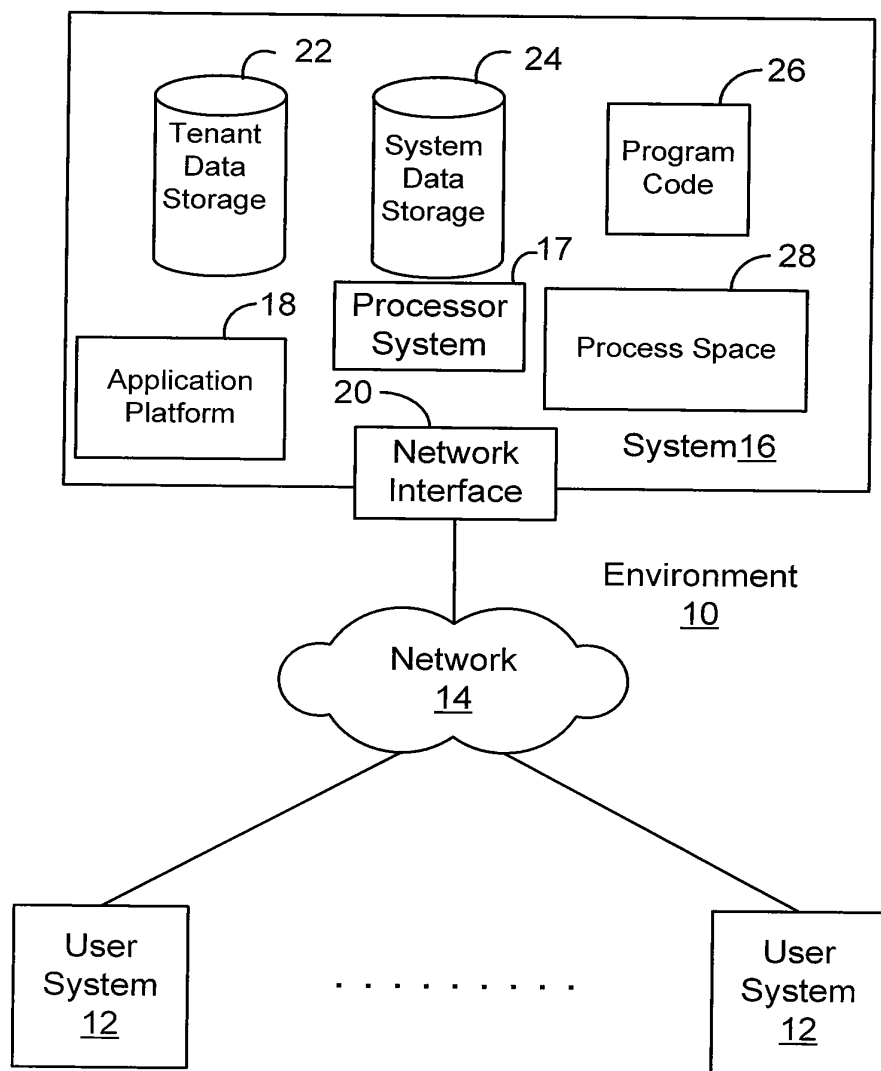
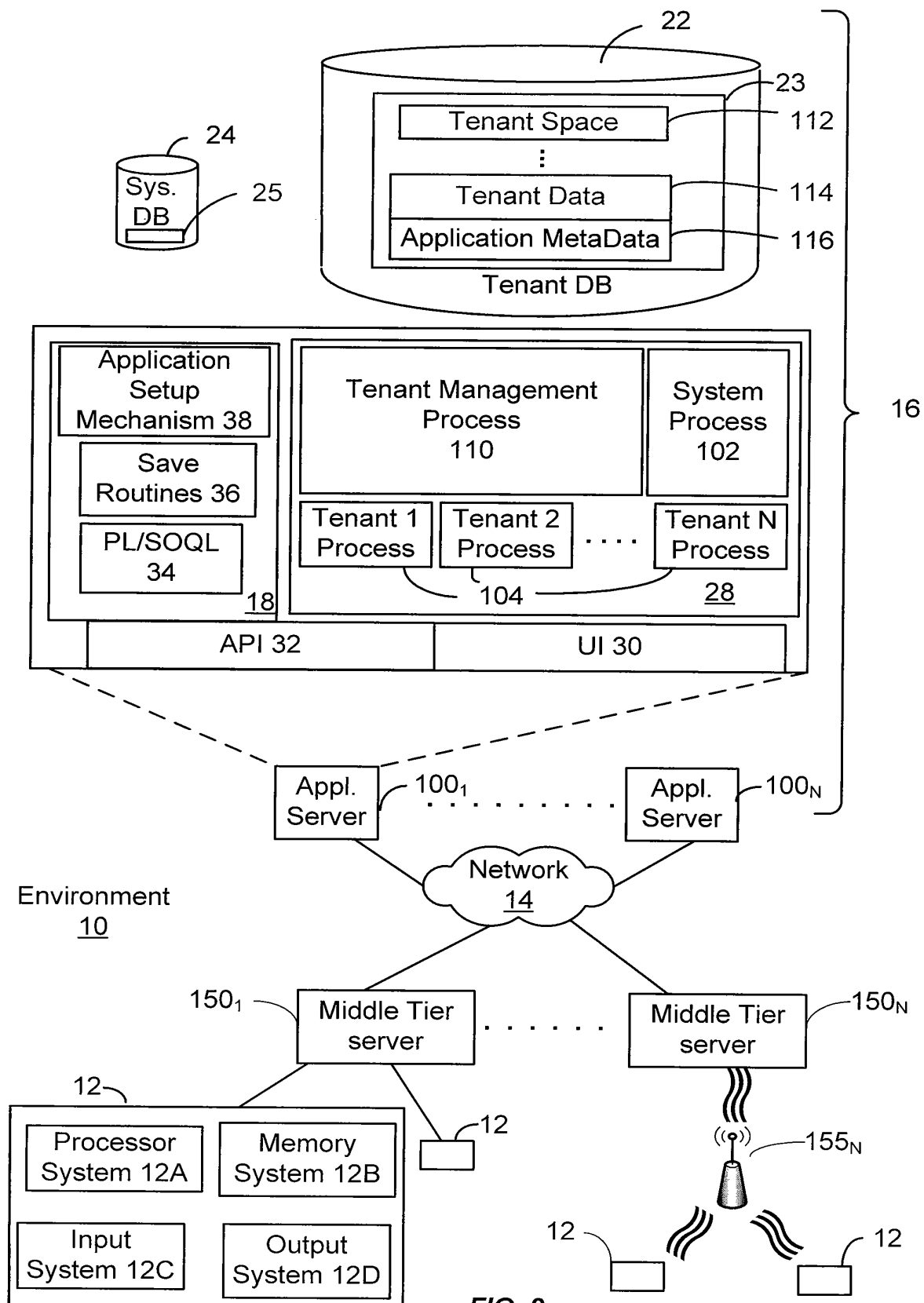


FIG. 1



Application Data Sheet

Application Information

Application number::

Filing Date::

Application Type:: Regular

Subject Matter:: Utility

Suggested classification::

Suggested Group Art Unit::

CD-ROM or CD-R??::

Number of CD disks::

Number of copies of CDs::

Sequence Submission::

Computer Readable Form (CRF)?::

Number of copies of CRF::

Title:: METHODS AND SYSTEMS FOR POLLING
AN ON DEMAND SERVICE

Attorney Docket Number:: 021735-004800US

Request for Early Publication:: No

Request for Non-Publication:: No

Suggested Drawing Figure::

Total Drawing Sheets:: 2

Small Entity?:: No

Latin name::

Variety denomination name::

Petition included?:: No

Petition Type::

Licensed US Govt. Agency::

Contract or Grant Numbers One::

Secrecy Order in Parent Appl.: No

Applicant Information

Applicant Authority Type:: Inventor
Primary Citizenship Country:: US
Status:: Full Capacity
Given Name:: David
Middle Name::
Family Name:: Brady
Name Suffix::
City of Residence:: Oakland
State or Province of Residence:: CA
Country of Residence:: US
Street of Mailing Address:: 1866 Mountain Blvd.
City of Mailing Address:: Oakland
State or Province of mailing address:: CA
Country of mailing address:: US
Postal or Zip Code of mailing address:: 94611

Correspondence Information

Correspondence Customer Number:: 20350

Representative Information

Representative Customer Number:: 20350

Domestic Priority Information

Application:: Continuity Type:: Parent Application:: Parent Filing Date::

Foreign Priority Information

Country:: Application number:: Filing Date::

Assignee Information

Assignee Name::

Street of mailing address::

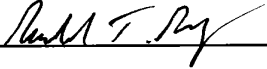
City of mailing address::

State or Province of mailing address::

Country of mailing address::

Postal or Zip Code of mailing address::

Submitted by:

Signature 

Date 9/11/08

Printed Name Gerald T. Gray

Registration Number 41,797

Electronic Patent Application Fee Transmittal				
Application Number:				
Filing Date:				
Title of Invention:		METHODS AND SYSTEMS FOR POLLING AN ON DEMAND SERVICE		
First Named Inventor/Applicant Name:		David Brady		
Filer:		Gerald Todd Gray./Sylvia Arnold		
Attorney Docket Number:		021735-004800US		
Filed as Large Entity				
Provisional Filing Fees				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Provisional application filing	1005	1	210	210
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Miscellaneous:				
Total in USD (\$)				210

Electronic Acknowledgement Receipt	
EFS ID:	3936639
Application Number:	61096617
International Application Number:	
Confirmation Number:	5378
Title of Invention:	METHODS AND SYSTEMS FOR POLLING AN ON DEMAND SERVICE
First Named Inventor/Applicant Name:	David Brady
Customer Number:	20350
Filer:	Gerald Todd Gray./Sylvia Arnold
Filer Authorized By:	Gerald Todd Gray.
Attorney Docket Number:	021735-004800US
Receipt Date:	12-SEP-2008
Filing Date:	
Time Stamp:	17:13:45
Application Type:	Provisional

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	\$210
RAM confirmation Number	2555
Deposit Account	201430
Authorized User	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
-----------------	----------------------	-----------	----------------------------------	------------------	------------------

1		2008-09-12_FWR_Prov_DWG_ ADS_022735-004800US.PDF	1274771 ff8fb61031fec8aab08d0dcefc6cc60eb04b5c1e5	yes	31
	Multipart Description/PDF files in .zip description				
	Document Description		Start	End	
	Provisional Cover Sheet (SB16)		1	2	
	Specification		3	25	
	Claims		26	26	
	Drawings-only black and white line drawings		27	28	
	Application Data Sheet		29	31	
Warnings:					
Information:					
2	Fee Worksheet (PTO-06)	fee-info.pdf	29586 7a9e26aa2e4847b1036639619c19003399bc4e28	no	2
Warnings:					
Information:					
Total Files Size (in bytes):			1304357		
This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503. <u>New Applications Under 35 U.S.C. 111</u> If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application. <u>National Stage of an International Application under 35 U.S.C. 371</u> If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course. <u>New International Application Filed with the USPTO as a Receiving Office</u> If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.					