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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
22 August 2013 (22.08.2013)

WIPO | PCT

(10) International Publication Number
WO 2013/123097 A1

(51) **International Patent Classification**:
G06F 15/16 (2006.01)

(21) **International Application Number**:
PCT/US2013/026000

(22) **International Filing Date**:
13 February 2013 (13.02.2013)

(25) **Filing Language**: English

(26) **Publication Language**: English

(30) **Priority Data**:
61/598,032 13 February 2012 (13.02.2012) US

(71) **Applicant**: **SKYKICK, INC.** [US/US]; 200 W. Thomas, Suite 400, Seattle, WA 98119 (US).

(72) **Inventors**: **DENNIS, John**; 1905 Virginia Street, Berkeley, CA 94709 (US). **RICHMAN, Evan**; 3520 Meridian Avenue North, Unit B, Seattle, WA 98103 (US). **SCHWARTZ, Todd**; 2600 Second Avenue #1106, Seattle, WA 98121 (US). **SCHWARTZ, Trent**; 2600 Second Avenue #1608, Seattle, WA 98121 (US). **TETT, Rick**; 2805 White Dove Drive, Plano, TX 75093 (US). **YOUNGE, Brad**; 23160 Timber Spring Place, Parker, CO 80138 (US).

(74) **Agents**: **COHEN, David** et al.; Perkins Coie LLP, P.O. Box 1247, Seattle, WA 98111-1247 (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, BN, 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, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

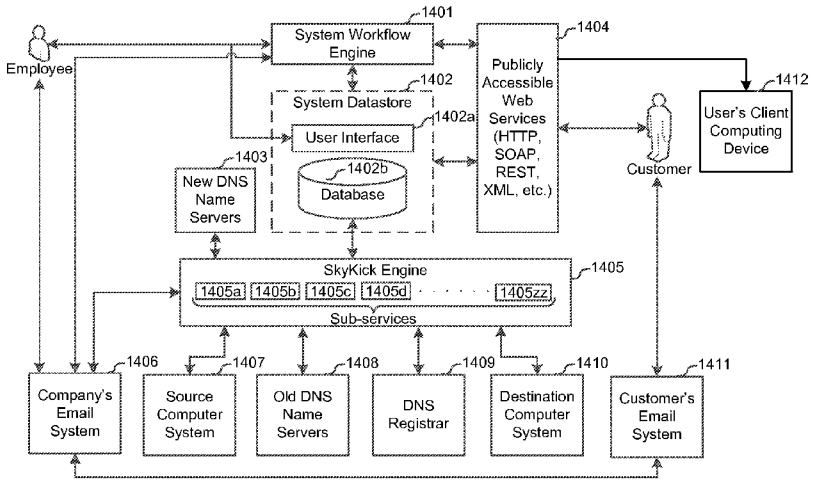
(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, 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))*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

(54) **Title**: MIGRATION PROJECT AUTOMATION, E.G., AUTOMATED SELLING, PLANNING, MIGRATION AND CONFIGURATION OF EMAIL SYSTEMS



(57) **Abstract**: Technology is generally described for automating the project management and execution of data migration from a source email system to a destination email system. In some examples, the technology can include receiving a domain name of a second computing system; obtaining domain name system (DNS) records for the received domain name; determining an email hosting provider for the second computing system; determining, by the processor, based on the obtained DNS records, an email system service type of the source email system on the second computing system; discovering mailboxes and message delivery rules of the source email system; displaying customization options for migrating discovered source email system mailboxes; migrating data items from the source email system to the destination email system; and managing migration of data from source email system client computing devices to the destination email system.

WO 2013/123097 A1

MIGRATION PROJECT AUTOMATION, E.G., AUTOMATED SELLING, PLANNING, MIGRATION AND CONFIGURATION OF EMAIL SYSTEMS

CROSS REFERENCE TO RELATED APPLICATION

[0001] This application claims the benefit of U.S. Provisional Patent Application No. 61/598,032 filed February 13, 2012, entitled "Automated selling, planning, migration and configuration of email systems", which is incorporated herein in its entirety by reference.

BACKGROUND

[0002] Migrating data to a cloud computing service such as, e.g., Microsoft® Office 365®, Google® Apps™, or Hosted Exchange, can be a complex project that requires information technology ("IT") expertise, careful planning, and significant time and expense. An email infrastructure and content system (including, e.g., email, calendar data, contact data, task data, document data, etc.) (collectively, an "email system") may, for example, be based on an obsolete computing system or an array of different systems, be heavily customized for its users, and/or be poorly documented. Data to be migrated may be stored on one or more servers and/or client computing devices that may or may not all be tightly integrated, at the same location, or even available concurrently.

[0003] To perform a project to migrate an email system from one computing system to another, with high probability to a different technology platform, and to do so without altering or losing information or requiring significant "downtime" during which the system is unavailable, generally requires IT professionals to assess the email system; choose and possibly customize appropriate software; perform a pilot migration; test the pilot project and make adjustments; and then perform and verify a complete migration.

[0004] Conventionally, this migration has been performed by IT consultants who spend considerable time and effort creating customized solutions for each customer. Moreover, traditional email migration technology ignores many of the necessary project management workloads required in a migration project, e.g., email mailbox analysis, email rules analysis, project planning and project management, process orchestration and timing, customer communications, domain name system ("DNS") configuration, and client configuration and setup.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] Figure 1 is a block diagram illustrating components of some embodiments of the present disclosure.

[0006] Figures 2–13 are flow diagrams illustrating routines performed by some embodiments of the technology to process a customer migration order.

[0007] Figure 14 is a block diagram illustrating computer systems and components arranged in accordance with some embodiments of the present disclosure.

[0008] Figure 15 is a display diagram illustrating a window with basic customer information requests.

[0009] Figure 16 is a display diagram illustrating a screen requesting customer contact information and administrative credentials.

[0010] Figure 17 is a display diagram illustrating a server plan selection screen.

[0011] Figures 18A–B are display diagrams illustrating account migration planning screens.

[0012] Figure 19 is a display diagram illustrating a subscription purchase screen.

[0013] Figure 20 is a display diagram illustrating a screen with user notification options.

[0014] Figure 21 is a display diagram illustrating a screen with other user notification options.

[0015] Figure 22 is a display diagram illustrating a migration preparation monitoring screen.

DETAILED DESCRIPTION

[0016] Technology is generally disclosed for automating the project management and execution of data migration from a source email system to a destination email system ("the technology"). The technology reduces the amounts of manual work and IT professional expertise required for such a migration project (a "migration"). In various embodiments, the technology provides a "self-service" application that can be used by a non-technical user to perform a migration in an automated fashion. In some embodiments, the technology comprises one or more of the following subcomponents (Figure 1):

[0017] A SkyKick engine subcomponent (block 102) is configured to create and maintain a customized project plan based on the architecture, system settings, and timing requirements on both the source and destination email systems; dynamically adapt the tasks depending on changes from the system workflows, hoster inputs, and customer inputs; perform quality assurance at each step; perform automated customer communications; and update the project plan and reports.

[0018] A hoster discovery subcomponent (block 101) is configured to identify a customer's current email system including but not limited to the system location (physical and virtual), the system type (e.g., Microsoft Exchange Server®, Lotus Notes®, Google Apps™, etc.), the primary communication protocol(s) that the system supports (e.g., Exchange Web Services, Web distributed authoring and versioning (WebDAV), Internet message access protocol (IMAP), post office protocol (POP), Google Data API™, etc.), and the correct format of credentials for accessing the system.

[0019] A hoster integration subcomponent (block 101) is configured to analyze a customer's email system to determine a current system architecture, server settings, port settings, and email information using a small number of non-technical inputs (e.g., an email address and password).

[0020] A mailbox discovery subcomponent (block 103) is configured to automatically interrogate a hosted email system to extract and normalize email logic and settings, e.g., a list of all mailboxes, forwarding rules, distribution groups, and aliases.

[0021] A DNS discovery and automation subcomponent (block 104) is configured to automate the discovery, relocation, quality assurance and modification of DNS settings in accordance with proper timing of the source and destination hoster(s) to successfully route email and minimize delays from the source email system to the destination email system without manual intervention.

[0022] A migration sync subcomponent (block 105) is configured to copy data from the source email system to the destination email system, including quality assurance automation, pre-migration of data, automatic coordination of what data has moved and will need to move, recurring copying of data, and on-demand data migration for users who do not provide credentials in advance.

[0023] A client application subcomponent (block 106) is configured to communicate with a user's email application to automate the timing, setup and configuration of the email

application and coordinate user data migration, including migration of local data, e.g., email messages, calendar appointments, contacts, and user email application settings.

[0024] The technology reduces the amount of work that an entity performing an email system migration (a "company") and/or a reseller partner or an entity whose documents are being migrated (a "customer") must perform to successfully migrate to a destination email system. The technology also reduces the number of electronic mail messages that are delivered to the source email system after the migration has been performed and thus attempts to ensure prompt delivery of electronic mail to the destination email system.

[0025] Several embodiments of the described technology are described in more detail in reference to the Figures. The computing devices on which the described technology may be implemented may include one or more central processing units, memory, input devices (e.g., keyboard and pointing devices), output devices (e.g., display devices), storage devices (e.g., disk drives), and network devices (e.g., network interfaces). The memory and storage devices are computer-readable media that may store instructions that implement at least portions of the described technology. In addition, the data structures and message structures may be stored or transmitted via a data transmission medium, such as a signal on a communications link. Various communications links may be used, such as the Internet, a local area network, a wide area network, or a point-to-point dial-up connection. Thus, computer-readable media can comprise computer-readable storage media (e.g., "non-transitory" media) and computer-readable transmission media.

Hoster Discovery

[0026] Figure 2 is a flow diagram illustrating a routine performed by some embodiments of the technology to process a customer migration order. The routine may begin at block 201 when, e.g., a customer representative ("user") comes to a company's website or, e.g., a partner website to place and configure a customer email system migration order. At block 202 in the illustrated embodiment, the technology prompts the user to submit an email address on the domain to be migrated and a desired date for the migration. (An illustrative example of such a prompt is described below in connection with Figure 15.) In some embodiments, the information the user submits includes, for example, the customer's basic contact information (e.g., the customer business name, the user's first and last name, customer email address(es), phone number(s), and mailing address(es)). After the requested information is submitted, the technology performs acts such as the following to deduce the

domain name registrar that the email system's domain name is registered with, the email system type and/or the Internet hosting service ("hoster") that serves the customer's email system:

[0027] At block 203, the technology uses one or more Internet domain name information ("WHOIS") databases, e.g., whois.iana.org, to find a responsible WHOIS server for the top-level domain (TLD) portion of the email system's domain name (e.g. 'com', 'org', 'info', etc.). Using the identified WHOIS service, the technology performs one or more WHOIS lookups on the domain name (e.g. 'skykick.com') to determine where the domain is registered. The technology obtains from the lookups the name and URL for the customer's domain registrar.

[0028] At block 204, the technology, using the Internet's DNS protocols, performs a DNS lookup for records including, for example, the following record types in the domain's zone: Mail Exchanger (MX) records (incoming mail servers), Address (A or AAAA) records (Internet Protocol (IP) addresses) associated with an MX record host name, any Service Location (SRV) or Canonical Name (CNAME) records (domain aliases) related to an Exchange Autodiscover service, and any TEXT records that constitute Sender Policy Framework (SPF) records (outgoing mail servers).

[0029] At block 204, the technology compares the found DNS records and IP addresses to a database of email system hosters that is created and managed by the company. This database contains one or more records for every known email system hoster. Each of these records contains information regarding possible MX, SPF, and Autodiscover values for the domains whose email systems are hosted by that hoster. In some embodiments, the technology compares found records to the database looking for matching fields in the order MX, MX IP Address, Autodiscover record, SPF record, and upon a match, the comparison is terminated. In some embodiments, the technology compares the found DNS records (up to, e.g., an exhaustive comparison of every possible record), gives a score to each match, and determines the most likely hoster based on the scores.

[0030] Those skilled in the art will appreciate that the logic illustrated in Figure 2 and described above, and in each of the flow diagrams discussed below, may be altered in various ways. For example, the order of the logic may be rearranged, substeps may be performed in parallel, illustrated logic may be omitted, other logic may be included, etc.

Hoster Integration

[0031] Referring now to Figure 3, at block 301, the technology attempts to connect to an unknown mail system for a custom domain, with only the user's email address and email password to the source email system.

[0032] After receiving the user's submittal of the requested email address and password (and optionally additional connection information), the technology attempts to derive the customer email system's connection information (block 303). The technology then initiates a process to derive a connection from those values.

[0033] In some embodiments, the technology issues DNS requests to discover a list of IP addresses corresponding to host names found in the DNS zone file for the domain name portion of the provided email address. For example, the technology issues DNS requests for hostnames derived from the provided domain name, MX records, Exchange Autodiscover service records, a list of candidate host names based on possible mail-oriented subdomains (e.g., a list of hostnames such as those in the following table), and, if provided, the user-inputted hostname.

"exchange"	"exch"	"owa"	"mail"	"email"
"webmail"	"m"	"imap"	"pop"	"smtp"
"imap1"	"imap2"	"imap3"	"imap4"	"imap5"
"pop1"	"pop2"	"pop3"	"pop4"	"pop5"
"smtp1"	"smtp2"	"smtp3"	"smtp4"	"smtp5"
"www"				

[0034] Each unique IP address from the resulting set is then interrogated for possible mail services by attempting to create a valid transmission control protocol ("TCP") connection to each of a set of ports associated with email systems, e.g., those in the following table.

Port	80	110	143	443	585	993	995
Name	"http"	"pop"	"imap"	"https"	"imap4-ssl"	"imaps"	"ssl-pop"

[0035] The technology matches each valid IP address-and-port combination from the resulting set back to hostnames with subdomains that are likely to match the IP address-and-port combination, e.g., one or more lists such as the following table.

80	110	143	443	585	993	995
"mail", "email", "webmail", "m", "www", "exchange", "exch", "owa"	"mail", "email", "webmail", "m", "www", "pop", "pop1", "pop2", "pop3", "pop4", "pop5"	"mail", "email", "webmail", "m", "www", "imap", "imap1", "imap2", "imap3", "imap4", "imap5"	"mail", "email", "webmail", "m", "www", "exchange", "exch", "owa", "mail", "email", "webmail"	"mail", "email", "webmail", "m", "www", "imap", "imap1", "imap2", "imap3", "imap4", "imap5"	"mail", "email", "webmail", "m", "www", "imap", "imap1", "imap2", "imap3", "imap4", "imap5"	"mail", "email", "webmail", "m", "www", "pop", "pop1", "pop2", "pop3", "pop4", "pop5"

[0036] In some embodiments, the technology creates permutations of possible system usernames using the email address and first and/or last name of the user as a starting point. The technology creates permutations of possible resource locators and credentials for each respondent IP address / port / hostname / username combination and attempts to log in to the email system under each of the resources using protocol-specific login methods. Full connection information for the highest ranking successfully connected mail protocol is then returned, e.g., using a ranking such as the following:

Rank	Protocol	Rank	Protocol
1	Exchange Web Services	4	Exchange ActiveSync
2	WebDAV	5	POP
3	IMAP	6	unknown

[0037] In some embodiments, the technology chains together the above-described procedures using a functionality map, e.g., a functionality map that can be predefined. The functionality map indicates to the system the incoming and outgoing value types of a particular procedure and whether the next procedure is to be executed serially (e.g., via a blocking call) or concurrently (e.g., via a non-blocking call). At each step the valid results from the prior executed procedure are reduced by a deduplication method and used to generate the function to be run in the subsequently executed procedure. For example, Internet input/output ("I/O") calls (e.g., sent UDP packets for DNS requests, TCP connections for attempts to contact mail services on specified ports, and TCP connections used to attempt mail protocol specific login methods) are executed concurrently, while procedures that have no I/O dependency (e.g., permutative procedures) are run serially. In

some embodiments, if at any point in the chain of execution a successful connection is made, currently executing and scheduled procedures that will only affect lower or same ranking mail protocols than the one returned are cancelled. Cancellation may be achieved by one or more of, e.g., two approaches. First, the technology may retain pointers to all open sockets used for outstanding Internet I/O procedures, allowing the sockets to be immediately closed regardless of their current state. Second, the technology may check a globally accessible cancellation flag at the beginning of each procedure in the functionality chain, such that setting the flag will cause downstream processes to return immediately. If at any point the top-ranked mail protocol is returned successfully, the entire chain of execution ends.

[0038] In some embodiments of the technology, at the end of the connection information deriving process described above, regardless of the outcome, the technology caches a per-domain list of discovered servers and of responding mail services on those servers. If the technology makes a discovery attempt against an already-processed domain, it bypasses the DNS queries, port scans, and resource locator permutations in favor of the cached list of servers and services, improving the speed of the process for each subsequent check.

[0039] At this point, if the connection is successful, then hoster integration is deemed successful and the user is directed to the next step in the migration.

[0040] In some embodiments of the technology, if a valid connection cannot be attained by the connection information derived from the email address and password information, the technology requests further information from the user. In some embodiments, the technology uses optimization logic to determine what additional information to ask the user for (block 301). In some embodiments of the technology, the set of heuristics includes, e.g., whether the customer's email system domain DNS records (e.g., records retrieved as described in connection with Figure 2, block 204 above) include the existence of an Exchange Autodiscover service record; whether MX, Autodiscover, or any other defining DNS records for the given domain name match one or more records in the company's hoster database; and if a match to the database is found, whether the database indicates that the hoster offers an Exchange service and/or also provides other services, e.g., POP, IMAP, and/or Microsoft Exchange ActiveSync® services. The technology may use results of the connection information deriving process (e.g., discovery of existing working mail services) as additional heuristics when calibrating the additional requested values. The

values may include more detailed credential information and mail server addressing values that are specific to known or predicted mail protocols.

[0041] If the above optimization logic indicates a possible Microsoft Exchange Server®, the technology can request input of an Exchange webmail address and/or username. If the optimization logic indicates a possible IMAP or POP server that is in the company's hoster database, the technology requests an administrative username and password to the control panel for that system. If the optimization logic indicates a possible IMAP or POP server that is not in the company's database or are ambiguous, the technology requests an email server address and username for the email system to be migrated.

[0042] In some embodiments, if the user submits an administrative username and password to a known hoster site control panel, the technology triggers an automated process (e.g., a web automation script) that logs into the control panel using the provided credentials and returns a value indicating whether the login was successful or not.

Mailbox Discovery

[0043] Referring now to Figure 4, at block 403, the technology discovers the customer's email system information architecture, including, e.g., how the email system distributes incoming and outgoing email and other types of data ("business rules"). Details of business rules include, e.g., information about email addresses (e.g., existing mailboxes), aliases (email addresses that do not have a specific mailbox but point directly to (i.e., deliver all mail to) email addresses that do have a specific mailbox), forwarding rules (rules that associate a second email address with a primary email address and send all mail received at the primary address to the secondary address), distribution groups (email addresses that do not have a specific mailbox but point directly to more than one address), shared mailboxes, and other functionality on various systems. In some embodiments, the technology discovers business rules using one or more processes corresponding to different mail hosting scenarios such as, for example, those described below.

[0044] In a first scenario, the email system provides a POP or IMAP service and the email system hoster is listed in the hoster database. The technology triggers an automated process (e.g., a web automation script) that logs into the hoster control panel using the provided credentials to return all mailbox addresses, aliases, distribution groups, and forwarding rules as they are explicitly defined in that control panel.

[0045] In a second scenario, the email system provides an Exchange service with an Exchange Web Services ("EWS") connection protocol available. The technology makes Web service calls for email address resolution using every letter of the location-specific alphabet and digits 0–9. Returned addresses are consolidated into a list of unique addresses. The technology expands any addresses that the EWS protocol specifies as a distribution list type (i.e., a list of member email addresses to which mail addressed to the distribution list address is delivered) to obtain their member email addresses. The addresses and distribution list membership information are returned and the technology starts a downstream process using social graph-enabled heuristics to determine aliases, forwarding rules, and distribution list membership, as further described below.

[0046] In a third scenario, the email system provides an Exchange service with a WebDAV connection protocol available. The technology makes Web service calls for email address resolution using every letter of the location-specific alphabet and digits 0–9. Returned addresses are consolidated into a list of unique addresses. The technology attempts to login to each of the returned addresses using a set of credentials known or expected to not work. The technology uses the HTTP response of each login attempt to classify the addresses into mailbox addresses and distribution group addresses. Mailbox addresses will return 401 (failed login attempt) and distribution group addresses will return 404 (resource not found). The addresses are returned with their classifications and the technology starts a downstream process using social graph-enabled heuristics to determine aliases, forwarding rules, and distribution list membership, as further described below.

[0047] In a fourth scenario, the email system provides a POP or IMAP service and the email system hoster is not listed in the hoster database. Using the provided mailbox credentials and the discovered connection settings, the technology logs into the given mailbox using an appropriate protocol-specific method. The technology scans message headers, e.g., for a period of time or until all messages have been scanned. The technology parses address information from the to:, from:, cc:, and bcc: fields in each message header for addresses that are on the same domain as the provided address. These are returned as candidate mailbox addresses and the technology starts a downstream process using social graph-enabled heuristics to determine aliases, forwarding rules, and distribution list membership, as further described below.

[0048] In some embodiments, as each user of the email system provides email system credentials and those credentials are tested and validated, the technology initiates a header

scanning process (Figure 6, block 608). Upon successful connection and login to a mailbox, the technology establishes a number of concurrent connections to the email system. The number of concurrent connections may be limited to avoid exceeding an estimated bandwidth limitation. The folders of the mailbox are enumerated. For each folder, a concurrent operation is started to divide the folders' contents into batches of message identifiers. For each batch, a concurrent operation is started to retrieve some or all message object headers for each message identifier found in the batch. The term "message object" includes any email system data item, including items other than email messages.

[0049] A challenge in migrating an email system (e.g., moving an entire company from one or more on-premise email servers to remote or cloud-based email servers) is that the aggregate of all users' email items can be quite large, and as a result the company's external network connection can be monopolized or severely impacted because of the bandwidth consumed by the data transfer requirements. The technology calculates and manages data transfer over the period from when a migration order is placed and when the cutover to the destination email system is completed (the "migration date"). The technology may estimate available bandwidth for data migration from the source email system and/or user devices as described below. The technology may also determine that the timing of the cutover should be adjusted to enable completion of the migration so as to impact the customer's normal Internet communications no more than necessary. A migration management system can dictate the timing of some data transfer activities, e.g., limiting some to off-peak hours and distributing individual account migrations across the period leading up to the migration date. For example, with respect to message header scanning, Internet I/O is limited throughout the process by the number of concurrent persistent protocol-specific TCP connections and by the number of threads available in the thread pools for folder and message item operations.

[0050] In some scenarios such as those described above, the technology uses a social graph-enabled heuristic to determine email business rules: In some embodiments, the technology creates a data structure from the content of the retrieved message object headers. For example, such a data structure may represent a bi-directed graph in which vertices represent either email addresses or messages, edges represent relationships based on existence of an adjacent address in the from:, to:, cc:, or bcc: field of a message, and directionality represents whether a message is going from or coming to a particular email address. The technology can utilize such a graph and heuristics such as, e.g., the following, to determine email business rules:

[0051] If the technology or a user has previously identified an address as a mailbox and/or the company has received validated credentials for a user mailbox, the address is classified as a mailbox. If the technology or a user has previously identified an address as an alias or a distribution group, the address is initially classified as an alias or distribution group, respectively.

[0052] If a number of messages exists above a threshold satisfying the following condition, the technology classifies a yet-unclassified address as an alias to the inspected mailbox: For each of the message vertices specified on all incoming edges of a inspected mailbox, the message contains no incoming or outgoing edge to the inspected mailbox and only one outgoing edge to the yet-unclassified address. In other words, at least a threshold number of messages in the mailbox are not connected by any incoming or outgoing edge to the mailbox address, and have exactly one outgoing edge to the yet-unclassified address.

[0053] If a number of incoming edges above a certain threshold and a number of outgoing edges above a certain, possibly different, threshold exist to a particular email address vertex, the email address corresponding to that vertex is classified as a mailbox.

[0054] If the previous condition is satisfied for more than one inspected address already classified as a mailbox, the yet-unclassified address is classified as a distribution list.

[0055] If the previous condition is satisfied above a possibly different threshold for an address classified as a mailbox rather than a yet-unclassified address, the relationship is classified as a forwarding rule.

[0056] Because more precision to the discovered set of rules corresponds to the amount of data in the graph, in some embodiments of the technology each addition to the graph is cached and the rules are re-run each time the graph receives more data. The business rules obtained and/or determined by the technology are in turn displayed to the customer (including, e.g., a reseller partner) so the customer can perform any further desired configuration.

[0057] In some embodiments, the technology uses the content of the retrieved message object headers to classify email addresses and associations between email addresses. The technology may de-duplicate the list of associations. Email addresses missing an associated email address are classified as existing mailboxes. Email addresses that are classified as existing mailboxes and have an associated address that is on or off the customer's domain are reclassified as forwarding rules. Email addresses that are not classified as existing

mailboxes, are in the customer's domain, and have a single associated address that is on or off the customer's domain are classified as aliases. Email addresses that are not classified as existing mailboxes, are in the customer's domain, and have multiple associated addresses that are on or off the customer's domain are classified as distribution groups. Any remaining unclassified addresses may be deemed to be outside of the customer's domain and discarded, unless the technology is also applied to the domain to which the discarded addresses belong.

Migration Customization Features

[0058] In some embodiments, once the technology has returned the mailboxes for each user and associated business rules, the technology allows the user to easily customize the migration configuration for a given email address (Figure 4, block 405). The technology enables a non-technical user to easily change, e.g., the routing rules and/or the mailbox data migration destination for any given mailbox, based on making changes in the user experience rather than code, whereas such customization has traditionally been performed by IT professionals using custom-developed code for each situation. As illustrated in Figure 18B, the technology enables a mailbox to be routed to the same email address 1811, a different email address (alias) 1813, a group of email addresses (distribution group) 1814, a shared mailbox 1812, or be deleted 1815, while directing the data to be migrated to the same mailbox 1816, a different mailbox or a shared mailbox 1817, or to no mailbox 1818. Once the user configures the mail routing and migration options, the SkyKick engine makes the appropriate routing changes in the destination system using a variety of methods (including, e.g., API calls, Powershell commands, and/or website automation scripting). After the routing changes have been made, the SkyKick engine migrates the data to the appropriate destination, and ensures that authorized users have access to that data. Further, because numerous combinations of actions can be performed on any given mailbox, set of mailboxes, email addresses, and/or destination mailboxes, the technology ensures that any changes to the revised mail flow or data migration are not in conflict with the mailbox or mailboxes at the destination email system. For example, the technology ensures that if data is being migrated from user A to user B, then user B cannot be changed to an invalid mailbox status, thus preventing accidental data loss.

User Information Gathering

[0059] In some embodiments, the technology creates a migration contact for each of the provided users or addresses and saves other information to the system datastore (block

501). The system datastore is described below in connection with Figure 14 (block 1402). A migration contact is a record in the system datastore that contains information about each user who has some sort of electronic document on the source email system that is to be migrated to the destination email system.

[0060] Referring again to Figure 5, the technology updates a status indicator for the migration process ("migration status") to reflect that the technology is gathering user information and a task scheduling subcomponent of the technology (the "system workflow engine") in turn creates a record in the system datastore that instructs the SkyKick engine to perform work (a "SkyKick engine task"), e.g., to send emails to gather user information (block 502). The system workflow engine subcomponent is described below in connection with Figure 14 (block 1401). The SkyKick engine subcomponent is described below in connection with Figure 14 (block 1405). In some embodiments of the technology, the SkyKick engine is an asynchronous process that performs the instructed type of work over a longer period of time than is typically acceptable with a user interface and returns results via, e.g., an email to the customer, a migration status update (which may trigger other processes), creation of another SkyKick engine task, creation of a task for a company employee, etc.

[0061] Referring again to Figure 5 (block 503), the technology may gather user information in, e.g., one or more of the following ways. If the customer specified that they would like an email to be generated and sent to the customer's primary contact, then the SkyKick engine generates and sends an email to the primary contact for the primary contact to in turn forward to each of the customer's users (blocks 504–505). If the customer specified that an email should be generated and sent directly to the customer's users, then the SkyKick engine generates and sends an email to each user (block 506). A link in the emails sent to the customer's users (block 507) takes the users to a website where they can submit their passwords (Figure 6, block 601).

[0062] Referring now to Figure 6, when a customer email system user submits a username and password on the website (block 601), the SkyKick engine determines whether it is valid by attempting to connect to the source email system with the supplied username and password (block 602). In some embodiments, if the credentials are not valid, the technology attempts other permutations of possible usernames such as, e.g., just the characters before the '@' sign in the customer's email address and/or provided username; the characters after the '/' sign in the user's username (if the '/' character is found in the username); the user's first name; the user's last name; the user's first name and last name

combined together; the user's first name and last name combined together with a period ('.') between them; the user's first initial and last name combined; the user's first name and last initial combined; etc. If none of the attempted permutations of the user's credentials is valid, then the user is notified so with a message on the webpage and asked to resubmit the information (block 601). If any attempted combination of username and password as described above is valid, then the migration contact record is updated in the system datastore (block 603) with the valid credentials and the user may be encouraged or instructed to download and install a client application onto their computing device (block 605). Additionally, the SkyKick engine performs an initial sync (described in further detail below and illustrated in block 607) and a header scanning process (described in further detail above and illustrated in block 608). If all of the passwords work (block 604), then the technology may update the migration status to indicate that the customer's end user passwords have been received and may send a notification email to the customer (block 606).

Client Application Setup

[0063] In some embodiments, the technology includes a client application that the customer and/or individual users can download or otherwise install on computing devices from which customer data may be migrated. The technology may provide each user an option to download and install client application software on one or more of the computing or smartphone devices that they use to access the customer email system via an email application (e.g., Microsoft® Outlook).

[0064] In some embodiments, client application software is implemented in two executable components: a background executable (the "client service") and a foreground component (the "client UI"). The client service is installed in a manner to begin execution automatically when installed and each time that the operating system is restarted. Each time the client service begins execution in the user's environment, the client service looks for previously stored data associated with this software. When executing for the first time and there is no previously stored data, the client service starts the client UI, which prompts the user to enter the user's username and password credentials for the customer email system. In some embodiments, the user's email address may be used in lieu of the username; or an alternate password may be used in lieu of the user's original password. Once a username and password are entered into the client UI form fields and submitted by the user, the client UI passes the user's credentials to the client service.

[0065] The client service establishes a connection over the Internet to an online interface for the SkyKick engine and submits an authentication request to the SkyKick engine with the user's credentials along with the user's computer's hardware media access control address or addresses ("MAC address"). The SkyKick engine attempts to find a contact matching the submitted username (or email address). If none is found, or if a matching contact is found but the password does not match a password on the contact, then the SkyKick engine returns an authentication request failure to the client service, and the client service informs the client UI which displays a failure message. In either case, the user may correct and resubmit the email system credentials.

[0066] If the technology finds a matching contact and the password matches, then the technology attempts to find a device associated with the contact having the computer's MAC address. If no such device is found, the technology creates a new device record and associates it with the identified contact. The submitting computer's MAC address is recorded in the new device record and a new unique ID is created and assigned to this device (a "device ID"). After the device record is created, the SkyKick engine returns authentication request success along with the device ID to the client service and includes information from the contact (e.g., a first name and information such as any new account credentials) and associated migration information (e.g., a scheduled migration date and time). The client service saves the validated credentials, device ID, and migration configuration data and informs the client UI, which may, e.g., advance to profile selection as described below.

[0067] If the user dismisses the client UI window without supplying a valid username and password pair, the client service will restart the client UI periodically until successful authentication has occurred. In some embodiments of the technology, the username and password are included in the client application installation package downloaded by the user, and upon initial execution, the username and password may be added to the client application's or user computer's data store.

[0068] In some embodiments of the technology, once successful authentication has been completed, the client service sends periodic requests to the SkyKick engine to obtain instructions for proceeding with particular actions. The SkyKick engine locates the associated device each time and evaluates the device information and the associated migration date to determine what instructions, if any, to return to the client service. When the client service is restarted and stored credentials are found, the client service resubmits those credentials and resumes periodically sending requests to the SkyKick engine to obtain instructions.

[0069] In some embodiments, after the technology creates a device record, the SkyKick engine sends the client service an instruction to perform a device inventory. When the client service receives this action, it collects and returns additional information about the user's computing device including, e.g., operating system version, processor "bitness", and email application version. The SkyKick engine uses the user's username, device ID, and MAC address to locate the appropriate contact and device record and adds the results of the inventory to the device record data.

[0070] Another instruction that the technology can send to the client service is a bandwidth measurement request. When the client service receives this action, it measures the time it takes to upload a known amount of data to a specific endpoint on the Internet to determine the available outbound bandwidth of the user's network and submits this measurement to the SkyKick engine. The SkyKick engine uses the user's username, device ID, and MAC address to locate the appropriate contact and device record and adds the results of the bandwidth measurement to the device record data.

[0071] In some embodiments, the client application begins profile selection by examining the mail information for the logged-in user and retrieving a list of profile names. If no profiles are present, a null selection is made by default and the block is complete. After determining that there are one or more profiles present, the client service examines the mail information and retrieves a list of the user computer's data store names associated with the first profile. If there is only one profile and one data store, the default selections of profile and data store suffice to make the selections without any user interaction. However, if there are multiple data stores or multiple profiles, the client UI may, e.g., display a profile and data store selection form. This form includes a method of choosing from the list of profiles such as dropdown selection boxes. If there is only one profile, that selection is implied. Given an implicit or explicit selection of a profile, the list of associated data stores is presented for selection. If the name of one of the data stores matches the email address of the user, this data store is made the default selection. If there is only one data store, this data store selection is implied. When the user indicates that profile/data store selections have been made the block is complete. If the client UI window is dismissed without a profile being selected, the client service will periodically restart the client UI until profile selection has occurred.

[0072] When profile and data store selection is complete (implicitly or explicitly), the client application saves the selections in a local data file, calculates the size of this data store,

and reports the information to the client service. The client service submits a profile selection request to the SkyKick engine. The technology finds the associated device record and records the user's profile and data store selection names and the data store size. The SkyKick engine then returns a response indicating success. The client UI displays to the user the results of saving and reporting the profile selection.

Initial Sync

[0073] After the technology has received a user's credentials for a source email system from which the user's data is to be migrated, the SkyKick engine begins a process to perform a migration pass (an "initial sync"). The technology creates a mailbox on the destination email system and then performs an initial user email system migration.

[0074] In some embodiments of the technology, upon successful connection and login to a mailbox, a number of concurrent connections are established to the mail service. The number of concurrent connections may be limited to avoid exceeding an estimated bandwidth limitation, as described in further detail above. The folders of the mailbox are enumerated. For each folder, a concurrent operation is started to divide the folder's contents into batches of message identifiers. The technology retrieves message identifying attributes by queries to the source email system for message object metadata. Depending on availability of this metadata for each protocol type, metadata attributes queried may include, e.g.,: message system ID, created date, received date, message size, existence of an attachment, message subject, etc. For each batch, the technology starts a concurrent operation to retrieve message object content for each message identifier in the batch.

[0075] In some embodiments, Internet I/O is limited by the number of concurrent persistent protocol-specific TCP connections and by the number of threads available in the thread pools for folder and message item operations. Internet I/O may also be attenuated by use of thread communication and blocking techniques to throttle network calls when messages over a specified size threshold are encountered. The technology may be tuned to throttle to a maximum acceptable memory footprint for any set of messages from batches still in-memory given, e.g., the size of the server image the technology is running on.

[0076] Each message object is assigned an identifier ("ID") that is replicable and guaranteed to be unique within the source email system, e.g., a globally unique identifier (GUID). Each message object is then translated to an object type acceptable to the destination email system and sent to its analogous location in the destination email system. The assigned

message ID is then stored in a cache (e.g., in memory and/or persisted to the system datastore) upon successful migration of a messaging object from source to destination email systems. Additionally, this assigned message ID is stored in the metadata of the message object itself on the destination email system, so as to be retrievable in the future.

[0077] In some embodiments, every Internet I/O call for the sync process is routed through an error tracking system to ensure fault tolerance. This system is seeded with a default number of retries and a set of error profiles that can be applied to specific exceptions that are anticipated to occur. The error profiles signal the system to make real-time changes to connection and/or message object state before retrying the Internet I/O call a number of times also signaled by the profile. The error profiles also dictate logging messages to an error tracking system. The error tracking system records error messages for each unfixable exception. The error tracking system also attempts to save source message object content that was available to the system in the case of, e.g., an exception that occurred after the content was successfully retrieved but before it was sent to or successfully saved on the destination email system. The error tracking system also provides a final report after each migration pass detailing a source email system snapshot, destination email system before and after snapshots, number of failed messages, counts of migration errors grouped by error message, and other metrics.

DNS Migration Automation

[0078] Referring now to Figures 7 and 8, after the customer's submittal of the migration order, the technology contacts the destination hosting service (e.g., through published APIs or via an Internet browser session) and, if no account exists, creates a destination account for the customer (Figure 7, blocks 703 and 704). Once this account is created and any necessary licenses are purchased, the technology updates the migration status to indicate that the technology is initiating a domain mapping process (Figure 8, blocks 801 and 802). The SkyKick engine requests a DNS verification record from the destination email system (blocks 803 and 807) and then can take various actions depending on, e.g., whether or not the customer has selected to move their DNS nameservers to a new DNS system managed by the company (block 806):

[0079] If the customer has selected to relocate their DNS nameservers, then the technology creates a new domain in the new DNS system and adds the existing DNS records from the current nameservers to the new DNS system as well as the domain verification

record returned from the destination email system (block 810). The technology can accomplish this by making queries for, e.g., the following Internet class DNS records for the customer's domain(s) from the current DNS nameservers and then creating an exact copy of the returned record(s), if any are returned, in the new DNS system:

A	
AAAA	
CNAME	email, imap, lyncdiscover, ftp, mail, pop, webmail, www, admin, administration, ads, adserver, alerts, alpha, ap, apache, app, apps , appserver, auth, autodiscover, backup, beta, blog, campus, cdn, chat, citrix, cms, console, corp, crs, cvs, database, db, demo, dev, devsql, dhcp, direct, dmz, dns, dns0, dns1, dns2, download, e, en, erp, eshop, exchange, f5, fileserver, firewall, forum, ftp0, git, gw, help, home, host, http, i, id, images, info, internal, internet, intranet, ipv6, lab, ldap, linux, local, log, m, mail2, mail3, mailgate, main, manage, mgmt, mirror, mobile, mobilemail, monitor, mssql, mta, mx, mx0, mx1, my, mysql, news, noc, ns, ns0, ns1, ns2, ns3, ns4, ntp, ops, oracle, owa, partner, partners, pbx, pda, portal, secure, server, shop, sip, smtp, sql, squid, ssh, ssl, stage, staging, stats, store, svn, syslog, test, testing, upload, vm, vnc, voip, vpn, whois, wiki, www2, www3, xml
MX	
SRV	_afpvertcp._tcp, _ssh._tcp, _autdiscver._tcp, _caldav._tcp, _client._smtp, _gc._tcp, _h323cs._tcp, _h323cs._udp, _h323ls._tcp, _h323ls._udp, _h323rs._tcp, _h323rs._tcp, _http._tcp, _iax.udp, _imap._tcp, _imaps._tcp, _jabber-client._tcp, _jabber._tcp, _kerbers-adm._tcp, _kerbers._tcp, _kerbers._tcp.dc._msdcs, _kerbers._udp, _kpasswd._tcp, _kpasswd._udp, _ldap._tcp, _ldap._tcp.dc._msdcs, _ldap._tcp.gc._msdcs, _ldap._tcp.pdc._msdcs, _msdcs, _mysqlsrv._tcp, _ntp._udp, _pp3._tcp, _pp3s._tcp, _sip._tcp, _sip._tls, _sip._udp, _sipfederatintls._tcp, _sipinternaltls._tcp, _sips._tcp, _smtp._tcp, _stun._tcp, _stun._udp, _tcp, _tls, _udp, _vlmcs._tcp, _vlmcs._udp, _wpad._tcp, _xmpp-client._tcp, _xmpp-server._tcp
TXT	

[0080] In addition to the above queries and replication logic, the technology can determine whether there are any 'wildcard' DNS records. To determine whether there is a wildcard A or CNAME record in the DNS Zone, the technology creates a string of characters that is extremely unlikely to be a domain name label and performs a request for a matching A or CNAME record. If the request returns a successful result, and the result is not equal to the requested string, then the system determines that whatever the result contains is the value of a wildcard A or CNAME record.

[0081] In some embodiments, in addition to the queries and replication logic described above, the technology uses web automation technologies to connect to the customer's current nameserver hoster's website control panel, log in with the customer's credentials, find the domain in question and retrieve some or all of the DNS records displayed in the control panel. The records are merged into the new DNS system with the other identified DNS records.

[0082] Next, the technology may update the nameserver listing at the registrar with the correct fully qualified domain name of each of the newly created nameservers (block 811). If for any reason the updating of the nameserver listing at the registrar fails, then a task is created in the system datastore instructing a company employee to manually update the nameservers at the registrar (blocks 812, 809 and 808). If, however, the SkyKick engine is successful in updating the nameserver listing at the registrar, then the SkyKick engine task is closed (Figure 9, block 903) and the technology updates the migration status to indicate that the technology is verifying the domain (Figure 9, block 902).

[0083] 2. If the customer has selected not to relocate their DNS nameservers, then a task is created in the system datastore instructing a company employee to manually add the verification record to the customer's nameservers and reduce the time-to-live (TTL) persistence of the customer's MX record(s) to the minimum possible value (blocks 806, 805 and 804).

[0084] If in the above process a task is created instructing the employee to do any work (blocks 804 or 808), then when the employee closes the task the system workflow engine sets the migration status to indicate that the technology is verifying the domain (Figure 9, block 902).

[0085] Referring now to Figure 9, when the migration status is updated to indicate that the technology is verifying the domain (block 902), then the system workflow engine creates a new SkyKick engine task (block 903) that causes the SkyKick engine to verify the mapping of the customer's domain (block 904). If verification is successful, then the SkyKick engine task is closed (block 909) and the migration status is updated to indicate that the technology is creating mailboxes (block 910). If the domain is not verified, then the SkyKick engine responds in, e.g, one of the following ways:

[0086] If 24 hours, or any other determined interval of time, has not transpired since the migration status was set to verifying the mapping of the customer's domain, then the

SkyKick engine waits for, e.g., ten minutes and then checks to see if the domain has verified (blocks 908, 905 and 904). The SkyKick engine continues this checking until 24 hours, or any other determined interval of time, has transpired since the migration status was set to verifying the domain.

[0087] If 24 hours, or any other determined interval of time, has transpired since the migration status was set to verifying the domain, then the SkyKick engine may create a task in the system datastore instructing a company employee to investigate why it is taking so long for the domain to verify (block 907). This in turn may cause the employee to investigate and fix the domain verification problem (block 906), and when complete, the employee closes the task which in turn causes the system workflow engine to create a new SkyKick engine task to verify the domain (block 903).

[0088] When the migration status is updated to indicate that the technology is creating mailboxes (block 910), the system workflow engine creates a SkyKick engine task (block 911) which in turn causes the SkyKick engine to connect to the destination email system and create the mailboxes (blocks 912 and 913) and then close the SkyKick engine task (Figure 10, block 1001) and update the migration status to verifying that access is still valid (Figure 10, block 1002).

[0089] Continuing in Figure 10, the system workflow engine may create a new SkyKick engine task to verify that access is still valid (block 1003), which in turn may cause the SkyKick engine to validate that the system still has all the necessary access to the source computer and email system (block 1004). If all the data is still valid (block 1005), then the SkyKick engine task is closed (block 1010) and the migration status may be updated to, e.g., indicate that the technology is waiting for the migration date (block 1011). If all of the data is not valid (block 1005), then the SkyKick engine closes the SkyKick engine task (block 1008) and may create a task instructing a company employee to troubleshoot the invalid source email system credentials (block 1009 and 1007). Once the employee fixes the access problems and closes the task (block 1006), the system workflow engine creates a SkyKick engine task to again attempt to verify that access is valid (block 1003).

[0090] When the migration status is updated to a state of waiting for the migration date (block 1011), the system workflow engine can create a new SkyKick engine task (block 1012) to cause the SkyKick engine to go into a wait mode until the scheduled migration date and time (block 1013).

[0091] Referring now to Figure 11, when the migration date and time arrives, the SkyKick engine closes the SkyKick engine task (block 1101) and updates the migration status to show that the technology is ready for migration (block 1102). When the migration status indicates that the technology is ready for migration, the system workflow engine may create a task for a company employee to confirm that the migration should begin (block 1103). Once the employee confirms that the migration should begin by closing the task (block 1104), or if the system does not require employee approval, the migration status is updated to indicate that the technology is updating the customer's MX record or records (block 1105). The system workflow engine creates a SkyKick engine task (block 1106), and the SkyKick engine responds in, e.g, one of the following ways:

[0092] If the nameservers were not moved to the new DNS system (block 1108), then the SkyKick engine closes the SkyKick engine task (block 1107) and creates a task instructing the employee to manually update the MX record (Figure 12, block 1201). The employee connects to the customer's nameserver and updates the MX record to the appropriate value (block 1203). When the employee closes the task the migration status is updated to indicate that the technology is migrating email system data (block 1205).

[0093] If the nameservers were moved to the new DNS system (block 1108), then the SkyKick engine connects to the new DNS system and updates the MX record (Figure 12, block 1202), closes the SkyKick engine task (Figure 12, block 1204) and updates the migration status to indicate that the technology is migrating email system data (block 1205).

[0094] When the migration status indicates that the technology is migrating email system data (block 1205), the system workflow engine creates a new SkyKick engine task (block 1206) and the technology creates a SkyKick engine task to perform an additional migration pass (a "final sync") for each migration contact associated with the migration order. This causes the SkyKick engine to copy electronic documents and records from the source email system to the destination email system for customer users (block 1207).

Final Sync

[0095] The technology performs a final sync migration pass that is similar to the initial sync pass described above. Differences between a final sync and the initial sync may include, e.g., the following: 1) as a source email system message identifier is created for each message in a batch, that identifier is checked against the cache of identifiers created in the initial sync pass. If the identifier exists in the cache, indicating that a message object has

already been migrated, the message object is marked to skip the actual transfer step (thus avoiding a duplicate item being created on the destination); and 2) as the source message object metadata is collected, current message state information (e.g., flags for read/unread, starred, draft) is stored in an in-memory data structure.

[0096] The technology verifies that that the final sync was successful (e.g., verifying that the number of exceptions encountered was below a preset number and that the migrated data passes a series of source-destination reconciliation checks). If the pass is valid, any migration object IDs that were not encountered on this migration pass are assumed to have been deleted from the source email system between the time of the initial sync and the final sync and are thus marked for deletion on the destination email system. A scan of the destination email system message headers is then performed (as described above in connection with the mailbox discovery process) and the state of changed message state information is updated to reflect the most recent state of the source email system. The technology returns the destination email system migration IDs (e.g., message IDs) of any message objects that were marked for deletion. Using those destination email system IDs, the technology sends the destination email system concurrent requests to delete each of the messages marked for deletion.

[0097] Once the SkyKick engine completes the copying of electronic documents and records it makes a determination of whether or not the migration was successful (block 1211). This determination is based on the tracking of successfully copied items, unsuccessfully copied items and the total number of items in the source email system. If the percent of unsuccessfully copied items is less than a pre-determined threshold (e.g., 1%), then the migration is considered successful and the SkyKick engine closes the SkyKick engine task (block 1210) and updates the migration status to indicate that the technology is ready to periodically sweep incoming email (Figure 13, block 1301). If the percent of unsuccessfully copied items is greater than a pre-determined threshold, then the migration is considered unsuccessful and the SkyKick engine closes the SkyKick engine task (block 1210) and then creates a task instructing the employee to troubleshoot the migration errors (block 1209). After troubleshooting the migration errors and closing the task (block 1208), the system workflow engine creates a new SkyKick engine task to migrate email system data (block 1206).

Client Application User Migration

[0098] In some embodiments of the technology, for any device for which the client application setup has been completed as described above (e.g., the client application installed, the user's credentials authenticated, and the user's profile/data store selection(s) reported), when the client service is executing, it will periodically request instructions from (i.e., poll) the SkyKick engine. Upon receiving and processing such a request (e.g., before the migration date), the SkyKick engine examines the device record and determines that the client application setup is complete. As a result, it updates a status to indicate that the device is ready to start a profile creation process. In response to a subsequent request for instructions from the client service running on that device, the technology returns an instruction to the client service to set up the user's email application profile for migration, and updates a record of the device's state. Additional instructions that can be returned by the SkyKick engine to the client service include actions to create a profile, delete a profile, and set a migration date and/or time.

[0099] The client application proceeds to configure the user's email application to send messages and to synchronize contents with the migration destination email system server(s) over the Internet. Configuration typically involves creating an email profile (a "destination profile"), unless such a profile is already present (e.g., having been previously created either manually or by a prior execution of the client service). The instruction issued by the SkyKick engine to set up a destination profile includes a specific name for the destination profile. When the client service receives such an instruction to configure a destination profile, the client service checks to see if the client UI is active and, if not, starts client UI execution. The client service issues a command to the client UI via inter-process communication with the command name and destination profile name. The client application establishes an API session with the user's email application, obtains a list of existing profiles, and looks for any that are named with the destination profile name. If any are so named, successful completion is reported to the client service, which in turn reports successful completion to the SkyKick engine.

[00100] If a destination profile is not present, the client application begins to configure a new destination profile. In some embodiments, all of the information for the configuration is supplied together with the instruction. In some embodiments, the client application obtains additional required information, e.g., by using an automatic discovery process to locate the email server information necessary to create the profile. Once all the required information is

available, the client application makes an API request to create a new profile with the destination profile name and configuration information. If the user's email application requires authentication credentials to be entered through the user interface to complete the request to create and/or configure a new profile, the client application launches a user interface automation thread that automatically detects and fills in a sign-in form with the associated username and password. The thread is terminated once the profile creation process has been completed. If the profile creation succeeds, the client service returns a success status to the SkyKick engine, which updates the progress state in the associated device record.

[00101] If either the client UI or the client service encounters an error in fulfilling the client profile configuration action, the client service returns a failure status to the SkyKick engine, which records the error state in the associated device record. That status may be reset to a "ready to start" state either by a manual change by a technician or by automatic repair and retry logic. A result of changing the state back to "ready to start" is that the instruction to set up the user's email application profile for migration will be returned to the client service upon a subsequent poll from the user device.

[00102] If the technology determines that the user's email application setup has been completed, then the technology identifies the next step as migration. Once the migration date and time have arrived, the SkyKick engine returns an instruction to the client service to migrate the email application data and updates the state of each migration element. The client application proceeds to copy data and settings from the user's email application source data store into the destination profile's default data store. The migrate instruction issued by the SkyKick engine includes a list of scopes (e.g., email data, calendar data, contact, data, task data, notes, rules, etc.) reflecting device migration scope items that are in the active migration process.

[00103] When the client service receives an instruction to migrate, the client service checks to see if the client UI is active and, if not, starts client UI execution. In some embodiments, the client service sets the migration date and time to the current date and time. Whenever running, the client application checks to see if the migration date has been reached. When it finds that it has, the client application begins a sequence of steps to complete the configuration of a new profile. The client application first makes an API request to change the default email profile to the destination profile name. The client application then determines if the email application process is running and, if so, stops (or kills) the process.

The client application subsequently reopens the email application in the new profile. If for any reason, the e-mail application does not stop, the client UI displays a message for the user to close the application and waits a reasonable period of time for that to occur.

[00104] Once the email application has been reopened, the migration action continues in one or more of a variety of paths depending on the source profile and store. The client application first determines whether there is source data to migrate. If there is no data store, the migration is considered complete immediately and success if reported to the technology and recorded in the device record as completed on each applicable scope.

[00105] If there is no local data to transfer, the client application skips past the data transfer step. If it is determined that there is source data to transfer, the client application opens the source data store and scans it to count the number of folders and messages in that store. The client application then opens the destination default data store and scans it to count the existing number of folders and messages in that store. The statistics for both source and destination stores is then reported to the SkyKick engine where it is recorded in the associated device record.

[00106] If the source contains local data to be migrated, the client application traverses the source data store folder hierarchy to find any folders that are of a type included in the migration scopes. If a folder is included, the client application determines whether the folder already exists in the destination store. If not, the client application creates the folder in the destination store. After that, a pass is made to compare each item in the source folder with items in the destination folder by, e.g., a combination of all the header values and body content. Those items that are not present in the destination are then copied to the respective destination folder. This is repeated for all subfolders within each source folder until all items are present in the destination store.

[00107] In some embodiments, additional filters may be included in the migration action instructions based on settings on either the migration order or the device record (in case of a conflict, the technology may prioritize, e.g., the device settings to override the migration order settings). Examples of such filters include, e.g., a maximum transfer size limit, a message receipt cutoff date by which messages will be copied to the destination only if received before that date, and an appointment age cutoff date by which appointments will be filtered to include only those non-recurring appointments after the given date (recurring appointments may be included regardless of date because of the complexity of determining

whether they recur at a compliant date). Scope, date and size limits reduce the amount of data copied to the destination store. Items in scope that are not copied to the destination due to a date or size filter may be copied to a newly created archive data store that is attached to the destination email system profile so that the archive is accessible to the user from the destination email system profile.

[00108] In some embodiments of the technology, if a maximum overall size for migration has been specified, the client application determines whether the total size of the source items (within scope and date criteria) exceeds this maximum size. If not, the entire set of source items is copied to the destination. Otherwise the technology may prioritize data to be migrated. For example, if the entire set of source items exceeds the maximum limit, but the Inbox and Sent Items and their folders are less than the maximum, then only the Inbox and Sent Items are copied to the destination. If the entire Inbox and Sent Items will not fit, the client application copies items from the Inbox and Sent Items until the size limit is reached. Items in scope that are not copied to the destination are copied to a newly created archive data store that is attached so that the archive is accessible to the user from the destination email system profile.

[00109] After the local source data has been copied to the destination store, the destination store is scanned again and the statistics reported to the SkyKick engine, where it can be determined to what extent the data migration succeeded. In addition, the technology copies custom settings from the source profile in the email application to the destination profile. Such settings may include, for example, default signature selections, categories, junk mail settings, out-of-office settings, quick steps, an autocompletion dictionary, etc. A migration scope for "rules" or "categories" may be included with folder type scopes in a migration action or may be issued as a separate migration action. If the latter, the client application uses the API to extract custom rules from the source and inserts those rules into the destination.

[00110] The migration identifies any data stores attached to the source profile other than the selected store that was migrated. These data stores are attached to the new profile so that the items in these stores are available to the user when using the email application opened with the new profile.

[00111] Because the user's email application may present sign-in and other dialogs through the user interface to complete the migration actions, the client application launches a

User Interface Automation thread that automatically detects and fills in a sign-in form with the associated username and password as well as dispenses with other dialogs emanating from the email application. This thread is terminated once the migration process has been completed.

[00112] If the migration executes to completion, the client service returns a success status to the SkyKick engine which updates the status of each migration step on the associated device to a "completed" state. If at any point the client UI or the client service encounters an error in fulfilling the migration actions that interferes with completion of the process, the client service returns a failure status to the SkyKick engine that sets the state of applicable migration scope items on the associated device record to an error state. The state(s) may be reset to a ready to start migration state either by a manual change by a technician or by automatic repair and retry logic. The result of changing the state(s) back to "ready to start" is that the instruction(s) to begin migration will be returned to the client service upon a subsequent poll from the user device.

[00113] In some embodiments of the technology, during a migration, the client UI displays information about tasks taking place, which may include a progress bar that shows one or more task completion percentages. If the client UI is closed by the user while any action including migration is in process, the migration process will continue as long as the computer is running. However, if the computer is shut down, the action will be processed and begin again when the computer is restarted. The client service may continue polling the technology for instructions during various actions. Among the actions that can be issued by the technology are, e.g., commands to suspend migration and resume migration.

[00114] Once the local data migration has been completed, the client UI can be closed by the user at will. If subsequent actions are issued by the technology, the client service will determine whether the client UI needs to be started. Among possible actions are a repeat of previous actions as well as optional actions. For example, an action to display a message to the user can be issued by which the client UI will display the supplied message in either a notification window or in a traditional application window based on the command parameters. An action to upgrade the client application software will cause the client application to download the new upgrade package and offer to run the upgrade (or alternatively to instruct the user to run it). An uninstall action can be issued to have the client application disable and uninstall itself.

Alternative Migration Mechanics

[00115] In some embodiments of the technology, the client application performs a local data copying operation on the user device to accomplish remote data migration. For example, when transferring items from a local data file to a new cloud or server-based destination email system account, the client application extracts each item to be migrated from the source data store on the device through an email application API and adds them to the destination local email store via the same API. Those items (which may be email messages, IM, appointments and other calendar items, tasks, notes, contacts, or other records managed and stored by an email application) are then copied to the destination email server's storage by the user's email application through a process (often referred to as "synchronization") that is not controlled by the client application.

[00116] When transferring items from a local data file to a new cloud or server-based destination email system account, it may be preferable in some situations to bypass the local store when inserting the items into the destination account.

[00117] In some embodiments of the technology, instead of the client application inserting data into a local data store, a migration agent executing on a server (e.g., in the cloud) inserts messages into the destination account. For example, the client application extracts the items from the local store and transfers them to the cloud or server-based migration agent for insertion. In some embodiments, the client application locates the data file (mail store) and transfers it to the migration agent, which extracts all or selected items from the data file according to migration order filters. The extraction may be either by directly reading the file and parsing it for metadata and content, or by simulating the user's email environment and using a compatible email application API. Once items have been extracted, the migration agent may use any method supported by the destination email account to add or insert the messages (e.g. Exchange Web Services for Office 365).

[00118] The client application communicates with the migration agent over a web service interface or other data transfer connection and may use an intermediate storage location or queue to "hand off" the data. Both the client application and the migration agent communicate with a migration management system as well. The migration management system instructs the client application when to transfer data based on bandwidth availability so as not to monopolize the data link from the customer's business premises. Once all desired items from the source account are present in the destination account, an email

application on any of a user's devices can access (or "synchronize") all or some items to that device.

[00119] Such an embodiment of the technology is potentially beneficial, for example, in the case where the device does not have sufficient storage space for both the original data file and a new data file containing copies of the items. It may also be beneficial when the email API is unable to insert messages into a new local email store due to errors that may occur due to corruption of the data store or for other uncontrolled reasons. Such an embodiment may be used for migrating a customer that will be primarily accessing the customer's destination email system account via a Web-based interface (e.g. Outlook Web Access) or through the use of email tools on devices other than the source device (e.g., when the existing mail client is incompatible with the destination email system account, or when a smart phone or other mobile device will be a primary tool for accessing the destination email system account).

[00120] In some embodiments, the migration takes place at the migration date (when the MX records have been switched) so that only a single migration pass is performed after the client application has been associated with the new account profile. This approach can be effective if no more mail will be downloaded to the client application and no more items will be added by the user once the source account is no longer accessed. It therefore requires that all users migrate at more-or-less the same time, e.g., in a single night. If there are large amounts of mail to copy, this can take many hours to make the local copies and even more hours to synchronize the items to the destination cloud account storage. If a failure occurs partway through the process, determining which items have been successfully copied to the destination may require an item-by-item comparison.

[00121] In some embodiments, the technology begins initial migrations for some or all users several days prior to the migration date. Changes may thus take place to the local data file after the initial migration; e.g., new items added, items deleted, and flags changed. Because of this, a subsequent reconciliation pass is required at the time of the switch. This multistep migration can be implemented using one of several approaches:

[00122] A first approach is to use a source/destination migrated item reference index table. The client application maintains a data table of both source and destination data store reference indexes, the date and time, and the source and destination folder path for each item copied to the destination. During a migration pass, each item in the source for which a source reference index is not present in the data table is copied to the destination. Those source item

indexes in the data table for which the items are no longer found in the source (i.e., items that have since been deleted locally) are deleted from the destination using the destination data store reference index.

[00123] A second approach is checking last modified timestamps. The source email application maintains a last modified timestamp on each item (e.g., for content, folder location, and metadata alterations). During any migration pass, in addition to additions and deletions, the client application re-copies and resets metadata for each item whose last modified time is more recent than the last migration to the destination. In addition, if a source item is in a different folder than recorded in the data table, the item is either moved in the destination or deleted and recreated in the correct location. This includes the movement of items that are in the source email system's "Deleted Items" folder.

[00124] A third approach is to use a copied item identification hash. The client application creates an M-bit hash for each item in the source, which is saved along with a destination reference in a data table for each item copied to the destination. The hash is based on, e.g., the item folder path, flags, metadata, file length, and the first N bytes of content. M and N are both defined so that the hash may be assumed to be a unique identifier for each item and its source state. During each migration pass, the client application traverses the entire source item set, determines the hash for the item, and if it is already in the data table, updates the table to show that item is still present in the source in the previously copied state. If it is not found in the data table, the client application will copy that item to the destination and add that entry to the data table. Any item previously copied but not still present in the source in its previously copied state is removed from the destination using the reference associated with that entry in the data table.

[00125] In some embodiments, the technology migrates data from a source email system to a destination email system by dividing source email system data into batches of item identifiers; retrieving item metadata from the source email system; for each batch of item identifiers, retrieving all item content for each item identifier in the batch; and for each item in a batch to be transferred, assigning a unique item ID to the item, translating the item to an item type compatible with the destination email system, copying the translated item to the destination email system, and saving the assigned unique item ID in the item metadata on the destination email system. In some embodiments the technology further receives an indication to limit the scope of data to be migrated; determines whether an item is within the indicated scope of data; and copies the item to the destination email system only if the item is

determined to be within the indicated scope of data. Copying an item to the destination email system may include, e.g., copying the item from a first data store on a client computing device to a second data store on the client computing device, and configuring an email application on the client device to synchronize the second data store with the destination email system. It may also include, e.g., copying a data store on a client computing device to a destination email system server for processing by a migration agent on the destination email system server. It may also include, e.g., processing a data store copied from a client computing device to a destination email system server, which may involve, e.g., directly reading the data store and parsing it for metadata and content, or simulating the email environment of the client computing device and using a compatible email application API to extract the item.

[00126] The process of migrating data from a source email system to a destination email system may further include performing multiple migration passes; for each pass, determining, for each item, whether identical items exist on the source email system and the destination email system; and for each item for which it is determined that identical items do not exist on the source email system and the destination email system, resolving the inconsistency by updating the destination email system. Determining, for each item, whether identical items exist on the source email system and the destination email system may include, e.g., referencing a data structure that maps a unique ID to every item copied to the destination. It may also include, e.g., identifying an item with a last modified time more recent than the last migration to the destination email system, so that resolving the inconsistency for an item with a last modified time more recent than the last migration to the destination includes re-copying and resetting metadata for the item. For a moved item, resetting metadata for the item may include modifying the path to the item in the destination email system, or deleting the item from the previous location and recreating the item in the moved location.

Sweep

[00127] Referring now to Figure 13, when the migration status is updated to indicate that the technology is ready to periodically sweep incoming email (block 1301), the system workflow engine creates a SkyKick engine task (block 1302), sends an email to the customer stating that the migration is complete (block 1307), and creates a task for a company employee to help get the customer set up and trained on the destination email system (blocks 1307 and 1306).

[00128] When the SkyKick engine task to sweep incoming mail system items is created (block 1302), the SkyKick engine goes into a sweeping mode for, e.g., 72 hours, or any other specified time period. In the sweeping mode, every, e.g., ten minutes, or at any other specified time interval, the SkyKick engine connects to the source email system and copies any newly arrived emails or other new email system items over to the destination email system.

[00129] At each interval, the technology performs an additional migration pass (a "sweep") that is similar to the final sync pass described above. Differences between each sweep and the final sync may include, e.g., the following: 1) as a source email system message identifier is created for each message in a batch, that identifier is checked against the cache of identifiers created in the initial sync or final sync pass. If the identifier exists in the cache, indicating that a message object has already been migrated, the message object is marked to skip the actual transfer step (thus avoiding a duplicate item being created on the destination); and 2) a cutoff date is determined based on the last recorded migration pass (e.g., from the final sync or a prior sweep). When querying message object metadata for batch assembly, the technology limits queries to messages that are later than the cutoff date.

[00130] Once 72 hours (or another specified time period) have elapsed, the SkyKick engine closes the SkyKick engine task (block 1305) and updates the migration status to indicate that the migration is complete (block 1304).

Illustrative Technology Subsystems

[00131] Referring now to Figure 14, in some embodiments, the technology can be implemented through the development of, e.g., five illustrative subsystems: a system workflow engine, a system datastore, a new DNS nameserver, a website, and a SkyKick engine (blocks 1401, 1402, 1403, 1404 and 1405, respectively) and communication with, e.g., six illustrative external systems: a company email system, a source computer system, a current DNS nameserver, a DNS registrar, a destination computer system, and a customer email system (blocks 1406, 1407, 1408, 1409, 1410 and 1411, respectively).

[00132] Block 1401 is the system workflow engine. This subsystem is responsible for scheduling tasks to be performed by either a company employee or the SkyKick engine subsystem (block 1405). The system workflow engine uses the storage capabilities in the system datastore's (block 1402) database (block 1402b) to store instructions for the employee

or SkyKick engine. These instructions are in turn read and processed by either the employee or the SkyKick engine.

[00133] Block 1404 is the company's website, which is built to interact with the customer. The website has direct communication with the system workflow engine (block 1401) so that it can schedule tasks to be completed by either the employee or the SkyKick engine. The website has direct connectivity with the system datastore (block 1402) so that it can read and write information about the customer (e.g. contact information, selected order information, desired migration date, employee names and email addresses, billing information, etc.).

[00134] Block 1403 is the new DNS nameserver. This new nameserver is controlled by the SkyKick engine, creating new nameserver records and deleting and updating nameserver records when appropriate.

[00135] Block 1402 is the system datastore. It includes two subcomponents, a user interface (block 1402a) and a database (block 1402b). The system datastore user interface is used by a company employee to view and interact with tasks that are assigned to him or her. The system datastore database is used to store persistent data related to the technology as well as employee tasks and SkyKick engine tasks that are created by one of the other subcomponents of the technology and in turn read by one or more of the other subcomponents.

[00136] Block 1405 is the SkyKick engine, which in the illustrated embodiment is in turn comprised of a flexible number of sub-services (blocks 1405a through 1405zz). As a whole, the SkyKick engine is an asynchronous process which performs the instructed type of work over a longer period of time than is typically acceptable with a user interface and then provides results back to the subcomponents of the technology through, e.g., one or more of an email to the customer via the company's email system (block 1406), an update of the migration status in the system datastore, the creation of another SkyKick engine task in the system datastore, or the creation of an account rep task in the system datastore

[00137] Since the SkyKick engine performs work which may take several seconds to several minutes, or longer, to complete, and the system as a whole has been designed to handle a large amount of volume, the SkyKick engine may be built as a collection of sub-services (blocks 1405a through 1405zz) that actually perform the work. The number of sub-

services can be either static or non-static and can be all executing on a single computer system or distributed across multiple computer systems.

[00138] Blocks 1406, 1407, 1408, 1409, 1410 and 1411 illustrate six external systems that the technology may interact or integrate with.

[00139] The company's email system (block 1406) is used to send and receive electronic mail messages needed as part of the process. The company's email system is accessible by a company employee and may be used to communicate with the customer's email system (block 1411).

[00140] The source computer system (block 1407) is a computer system housing source email system data (e.g., electronic mail, electronic contacts, electronic calendar appointments and any other type of electronic document or file) that the technology migrates to the destination computer system (block 1410).

[00141] The current DNS nameservers (block 1408) are queried by the SkyKick engine to read the values of the customer's DNS records prior to beginning the migration.

[00142] The DNS registrar (block 1409) stores records identifying the customer's DNS nameservers and is updated by the SkyKick engine during the migration.

Illustrative Website Pages

[00143] Figures 15–22 are display diagrams illustrating screens that may be shown in various embodiments of the technology. Figure 15 is a display diagram illustrating a window with basic customer information requests. The customer is asked three questions to get started: their business email address, the number of employees in the company, and a desired migration date. When the GO button is clicked the customer progresses to Figure 16.

[00144] Figure 16 is a display diagram illustrating a screen requesting customer contact information and administrative credentials. The customer is asked for several items of information (block 1601), including, e.g., the website user's name, email address, phone number, preference for being contacted, an alternate contact's name, email address and phone number, and the desired migration date and time.

[00145] Next on the page are identification of the customer's email hoster (block 1602) and the customer's email registrar (block 1603). These two blocks are determined, e.g., by a background process while the user is completing the elements in block 1601. After the values are determined by the background process, they are rendered on the web page for the

user to see. The email hoster and email registrar values are determined as described above in connection with Figure 2.

[00146] In the illustrated embodiment, once the email hoster and email registrar values have been determined, then the last three blocks on the web page (blocks 1604, 1605 and 1606) are made visible (they may initially not be visible) and the website user is required to provide the information asked for in blocks 1604 and 1605. Once complete, the user clicks on the Submit button (block 1606).

[00147] When the Submit button (block 1606) is clicked, the web server uses the information provided in blocks 1604 and 1605 to attempt to authenticate with the email hoster and email registrar. If this authentication fails for either of these, then an error message stating which system the server failed to authenticate with may be displayed and the user may be required to re-enter the failed information until the authentication succeeds. When the authentication succeeds, the user is taken to the webpage shown in Figure 17.

[00148] Figure 17 is a display diagram illustrating a server plan selection screen. The user is asked a series of questions to determine which destination server plan may be best for the user. These questions are dependent on the types of plans offered by the destination email system, and the questions in Figure 17 are only an example of what may be asked. Once the questions in Figure 17 are answered and the Submit button is clicked, the user can be taken to the webpage shown in Figure 18A.

[00149] Figures 18A–B are display diagrams illustrating account migration planning screens. When the web page shown in Figure 18A is arrived at, if the process that is gathering the information from the email hoster has not yet completed, then the page shows a message stating that email hoster information is being retrieved. After the process completes, or if it had completed before the customer arrived at the page, then the elements shown in Figure 18A can be rendered on the page.

[00150] A list of all mailboxes found at the email hoster is displayed (block 1801). The user may have the ability to add additional mailboxes to the list by clicking on the Add button (block 1802) and may also be able to specify whether or not a mailbox's information should be migrated. A list of all distribution lists and/or forwarding accounts found at the Email Hoster may also be displayed (block 1803). The user may have the ability to add additional forwarding accounts by clicking on the Add buttons 1802 or 1804.

[00151] A migration customization detail screen is illustrated in Figure 18B. The screen displays options for the user to specify how email addressed to a mailbox should be routed at the destination email system, e.g., delivered to the same email address 1811, to a different email address 1813 (making the original mailbox address an alias), to a group of email addresses 1814 (making the original mailbox address a distribution group), to a shared mailbox 1812, or to no mailbox (deleting the original mailbox address) 1815. The screen also displays options for the user to specify how existing mailbox data should be migrated to the destination email system, e.g., to the same mailbox 1816, to a different mailbox 1817, to a shared mailbox, or not to migrate the mailbox data at all 1818. Clicking the Update button 1820 saves the user's choices and returns the user to Figure 18A. When the user has completed any changes needed on the page, the Submit button 1805 is clicked and the user may be displayed the page shown in Figure 19.

[00152] Figure 19 is a display diagram illustrating a subscription purchase screen. This webpage displays the recommended destination service plan and company fees (block 1901). The possibilities here are dependent on the destination computer system and email system, and on the company implementing the technology and its service offerings. Also on the page may be a set of form elements the customer may fill out to specify how they would like to pay for the service (block 1902). Once the form is complete the user can click the Submit button 1903 to be taken to the webpage shown in Figure 20.

[00153] Figure 20 is a display diagram illustrating a screen with user notification options. The user may be shown a synopsis of all the user mailboxes that will be created on the destination email system and have mail migrated (block 2001). On the bottom half of the page the user may be allowed to select the way in which they would like the users notified about the migration along with instructions on submitting the password to their source email mailbox so that the migration technology will have the necessary access to read the existing email from the source location during the migration step. The user can select the "Have Company Notify Your Team" tab 2002, in which case they may see the options to preview the email 2003, add a custom introduction 2004 and have the company send the notification emails 2005.

[00154] Alternatively, the user may select the "You Notify Your Team" tab 2002, in which case the technology will display the page in Figure 21. Figure 21 is a display diagram illustrating a screen with other user notification options. When this option is selected, the user may be able to preview the notification email 2103 and have it sent to himself or herself

2104. Once the user receives the notification email, he or she can in turn forward it to other users.

[00155] Once the user has decided how he or she would like to notify the customer's employees and clicks the "Send" button 2005 or 2104, the system sends the appropriate email and then takes the user to the page shown in Figure 22.

[00156] Figure 22 is a display diagram illustrating a migration preparation monitoring screen. The customer is able to see the list of their employees who have been notified to visit a web page and enter the password for their source email system mailbox. As employees open the email, visit the webpage, share their access information and/or perform any other type of action that is tracked by the system, the chart shown in block 2201 may be updated with that information. Once all users have performed all the necessary steps required to migrate their email, then the chart in Figure 22 (block 2201) can reflect that progress and the Schedule Migration button 2202 may become enabled (it may initially be disabled). Furthermore, if the user leaves this page or closes their browser, upon returning to the website, they can be taken directly to this page and may be able to see the current migration preparation status. Once all information has been entered by all the employees, the user can click the Schedule Migration button 2202 for all the collected information to be delivered to the system datastore and for the technology to take over the remainder of the automated migration process.

[00157] It is to be understood that the subject matter defined in the appended claims is not necessarily limited to the specific structural features and/or methodological routines described above. Rather, the specific features and routines described above are disclosed as example embodiments of the claims. Accordingly, the invention is not limited except as by the appended claims.

CLAIMS

We claim:

1. A computer-readable storage medium storing computer-executable instructions for causing a first computing system having a processor to perform a method for automating a project of data migration from a source email system to a destination email system, comprising:

receiving a domain name of a second computing system;

obtaining domain name system (DNS) records for the received domain name;

determining an email hosting provider for the second computing system;

determining, by the processor, based on the obtained DNS records, an email system service type of the source email system on the second computing system;

discovering mailboxes and message delivery rules of the source email system;

displaying customization options for migrating discovered source email system mailboxes;

migrating data items from the source email system to the destination email system; and

managing migration of data from source email system client computing devices to the destination email system.

2. A method performed by a first computing system having a processor to determine email hosting information about a second computing system, comprising:

receiving a domain name of the second computing system;

querying a database that maps domain names to domain registrars;

receiving, in response to the querying, identification of the domain registrar for the second computing system;

requesting, from the identified domain registrar, domain name system (DNS) records for the received domain name;

receiving, in response to the requesting, DNS records for the received domain name;

accessing a database of known email hosting provider DNS records;

comparing, by the processor, the received DNS records and the database of known email hosting provider DNS records; and

determining, based on the comparing, an email hosting provider for the second computing system.

3. The method of claim 2, wherein receiving a domain name of the second computing system comprises receiving an email address of the second computing system.

4. The method of claim 2, wherein receiving DNS records for the received domain name comprises receiving one or more of an MX record, an Autodiscover record, or an SPF record.

5. The method of claim 2, wherein:

comparing the received DNS records and the database of known email hosting provider DNS records comprises:

searching DNS fields in a predetermined order; and

identifying at least one received DNS record that matches a corresponding known email hosting provider DNS record; and

determining an email hosting provider for the second computing system comprises selecting a known email hosting provider with the first matching record.

6. The method of claim 5, wherein the predetermined order is MX, MX IP Address, Autodiscover Record, SPF Record.

7. The method of claim 2, wherein:

comparing the received DNS records and the database of known email hosting provider DNS records comprises scoring received DNS records that match corresponding known email hosting provider DNS records; and

determining an email hosting provider for the second computing system comprises selecting a known email hosting provider based on the scoring.

8. A method performed by a first computing system having a processor to identify sufficient email system login credentials for an email system having an email system service type, comprising:

obtaining an email address and an email password for the email system;

identifying a second computing system that hosts the email system;

obtaining domain name system (DNS) records for the second computing system;

determining, by the processor, based on the obtained DNS records, an email system service type of the second computing system email system;

deriving, based on the obtained email address and email password and the determined email system service type, email system login credentials; and

connecting to the email system using the derived email system login credentials.

9. The method of claim 8, wherein obtaining an email address and an email password for the email system comprises obtaining no other email system login credentials.

10. The method of claim 8, wherein determining the email system service type of the second computing system email system comprises identifying a DNS record associated with an email system type;

11. The method of claim 10, wherein the DNS record is an Exchange Autodiscover record;

12. The method of claim 8, wherein determining the email system service type of the second computing system email system comprises:

accessing a database of known email hosting provider DNS records and associated known email hosting provider email system service types;

comparing, by the processor, the obtained DNS records and the database of known email hosting provider DNS records;

determining, based on the comparing, an email hosting provider for the second computing system; and

determining an email system service type associated with the email hosting provider.

13. The method of claim 12, wherein determining an email hosting provider for the second computing system comprises selecting a known email hosting provider for which at least one obtained DNS record matches the corresponding known email hosting provider DNS record.

14. The method of claim 12, wherein:

comparing the obtained DNS records and the database of known email hosting provider DNS records comprises scoring obtained DNS records that match corresponding known email hosting provider DNS records; and

determining an email hosting provider for the second computing system comprises selecting a known email hosting provider based on the scoring.

15. The method of claim 8, wherein determining an email system service type comprises selecting a most likely email system service type.

16. The method of claim 8, wherein deriving email system login credentials and connecting to the email system using the derived email system login credentials comprises executing concurrent Internet input/output (I/O) calls;

17. The method of claim 16, further comprising:

assigning a ranking to connections; and

when a successful connection is made, canceling currently executing and scheduled I/O calls for connections with a same or lower ranking.

18. The method of claim 8, further comprising, if the derived email system login credentials are insufficient to connect to the email system:

selecting, based on the email system service type, email system login credentials to request; and

prompting for the selected email system login credentials.

19. The method of claim 18, wherein prompting for the selected email system login credentials comprises requesting a username.

20. The method of claim 18, wherein prompting for the selected email system login credentials comprises requesting a host computer name.

21. The method of claim 8, further comprising:

creating a list of possible email system hosts using the obtained DNS records; and

for each host in the list of hosts:

attempting to connect to the host on a port associated with email systems; and

for each successful connection:

determining a supported communication protocol.

22. The method of claim 21, wherein determining a supported communication protocol comprises:

attempting to log in to the email system using protocol-specific login methods in a predetermined order of communication protocols; and

identifying the protocol associated with the first successful login attempt.

23. A method performed by a computing system having a processor to discover substantially all mailboxes and message delivery rules in an email system, comprising:

obtaining a list of email addresses on the email system; and

classifying, by the processor, each of substantially all the obtained addresses as a mailbox, alias, forwarding rule, or distribution list.

24. The method of claim 23, wherein obtaining a list of email addresses comprises:

receiving administrative credentials for logging in to a control panel for the email system hoster; and

executing a web automation script that logs into the email system hoster control panel, instructs the email system hoster to provide email address data, and parses the email system address data.

25. The method of claim 23, wherein obtaining a list of email addresses comprises:

making Web service calls to the email system for email address resolution using every letter of the location-specific alphabet and digits 0–9;

consolidating returned addresses into a list of unique addresses; and

expanding distribution list addresses.

26. The method of claim 23, wherein obtaining a list of email addresses comprises:

logging into an email account with messages on the email system;

scanning the headers of messages to parse address information from the to:, from:, cc:, and bcc: fields in each message header; and

identifying addresses that are on the same domain as the email account.

27. The method of claim 23, wherein classifying each of substantially all the obtained addresses as a mailbox, alias, forwarding rule, or distribution list comprises:

attempting to log in to one or more returned address; and

using the login attempt failure response code to distinguish mailbox addresses from distribution group addresses.

28. The method of claim 23, wherein classifying each of substantially all the obtained addresses as a mailbox, alias, forwarding rule, or distribution list comprises:

creating a social graph data structure containing obtained email addresses; and

using, by the processor, the social graph data structure to classify one or more addresses as a mailbox, alias, forwarding rule, or distribution list.

29. The method of claim 28, wherein creating a social graph data structure containing obtained email addresses comprises creating a bi-directed graph in which vertices represent email addresses and/or messages, edges represent relationships based on existence of an address in a message header field, and directionality represents whether a message is going from or coming to a particular email address.

30. The method of claim 29, wherein using the social graph data structure to classify one or more addresses comprises classifying a yet-unclassified address as a mailbox if a number of incoming edges above a certain threshold and a number of outgoing edges above a certain, possibly different, threshold exist to a particular email address vertex, the email address corresponding to that vertex is classified as a mailbox.

31. The method of claim 29, wherein using the social graph data structure to classify one or more addresses comprises classifying a yet-unclassified address as an alias to

a mailbox address if at least a threshold number of messages in the mailbox are not connected by any incoming or outgoing edge to the mailbox address, and have exactly one outgoing edge to the yet-unclassified address.

32. The method of claim 29, wherein using the social graph data structure to classify one or more addresses comprises classifying a yet-unclassified address as a distribution list if at least a threshold number of messages in a plurality of mailboxes are not connected by any incoming or outgoing edge to the addresses of the mailboxes they are in, and have exactly one outgoing edge to the yet-unclassified address.

33. The method of claim 29, wherein using the social graph data structure to classify one or more addresses comprises reclassifying a mailbox-classified address as a forwarding rule if at least a threshold number of messages in the mailbox are not connected by any incoming or outgoing edge to the mailbox address, and have exactly one outgoing edge to the yet-unclassified address.

34. A method performed by a computing system having a processor to enable a user to customize an email system migration from a source email system to a destination email system, comprising:

displaying a list of all source email system mailboxes that may be migrated;

for each source email system mailbox in the list:

providing a set of options for routing email addressed to the mailbox after migration to the destination email system; and

providing a set of options for migrating mailbox data to the destination email system; and

ensuring, by the processor, that selected options do not conflict.

35. The method of claim 34, wherein the set of options for routing email addressed to the mailbox after migration to the destination email system comprises routing to the mailbox, an alias, a distribution group, or a shared mailbox, or deleting the mailbox address.

36. The method of claim 34, wherein the set of options for migrating mailbox data to the destination email system comprises migrating data to the mailbox, a different mailbox, or a shared mailbox, or not migrating the mailbox data.

37. A method performed by a computing system having a processor to migrate data items from a source email system to a destination email system, comprising:

creating a data structure for recording items copied to the destination email system;

performing a first migration pass, comprising:

dividing source email system data into batches of item identifiers;

retrieving item metadata from the source email system;

for each item in a batch to be transferred:

assigning a unique item ID to the item;

retrieving item content;

creating an identification hash for the item;

translating the item to an item type compatible with the destination email system;

copying the item to the destination email system;

saving the assigned unique item ID in the item metadata on the destination email system; and

saving in the data structure the identification hash and a reference to the item in the destination email system; and

performing a second migration pass, comprising:

determining, for each item, whether identical items exist on the source email system and the destination email system; and

for each item for which it is determined that identical items do not exist on the source email system and the destination email system, resolving the inconsistency by updating the destination email system.

38. The method of claim 37, wherein the basis for the identification hash includes the item folder path, flags, file length, and at least a portion of the item's content.

39. The method of claim 37, wherein the length of the identification hash is defined so that the hash may be assumed to be a unique identifier for each item and its source state.

40. The method of claim 37, wherein determining, for each item, whether identical items exist on the source email system and the destination email system comprises determining the hash for the item, and if it is already in the data structure, updating the data structure to show that the item is still present in the source email system in the previously copied state.

41. The method of claim 37, wherein for each item for which it is determined that identical items do not exist on the source email system and the destination email system, resolving the inconsistency comprises:

for an item whose identification hash is not found in the data structure, copying the item to the destination email system and adding the identification hash to the data structure; and

for an identification hash in the data structure for which the corresponding item is not found in the source email system, deleting the item from the destination email system using the reference to the item contained in the data structure and removing the identification hash from the data structure;

42. A computer-readable storage medium storing computer-executable instructions for causing a computing system having a processor to perform a method for domain name system (DNS) record discovery for an identified domain, comprising:

creating a list of potential DNS label records;

creating a set of discovered DNS records;

for each DNS label in the list:

performing, by the processor, DNS requests for records matching the label;

for each request that returns an alias to a record on the identified domain, recursively performing DNS requests; and

for each request that returns an IP address, adding the record to the set of discovered DNS records;

identifying wildcard DNS records; and

removing DNS records having the same IP address or alias as a wildcard DNS record from the set of discovered DNS records.

43. The computer-readable storage medium of claim 42, further comprising, for each DNS label in the list, for each request that returns an alias to a record on a domain different than the identified domain, adding the record to the set of discovered DNS records.

44. The computer-readable storage medium of claim 42, wherein identifying wildcard DNS records comprises:

creating a string of characters that is extremely unlikely to be a domain name label;

performing a DNS request for a record matching the created string in the identified domain; and

determining, if the performed DNS request returns a successful result, that the returned result is the value of a wildcard DNS record.

45. The computer-readable storage medium of claim 44, wherein the requested record is an A record.

46. The computer-readable storage medium of claim 44, wherein the requested record is a CNAME record.

47. The computer-readable storage medium of claim 44, further comprising executing a web automation script that logs into an identified domain hoster control panel, instructs the domain hoster to provide DNS data for the identified domain, and parses the DNS data.

48. A computer-readable storage medium storing instructions for causing a computing system having a processor to perform a method for managing email system migration from a plurality of source email system client computing devices to a destination email system, comprising:

receiving information about source email system client computing devices;

creating, by the processor, a migration plan for each source email system client computing device; and

transmitting the migration plan to each source email system client computing device, such that the source email system client computing devices copy email system data to the destination email system according to the migration plan.

49. The computer-readable storage medium of claim 48, wherein a migration plan comprises an instruction to copy email system data to the destination email system at a specified time.

50. A computer-readable storage medium storing instructions for causing a source email system client computing device having a processor to perform a method for email system migration to a destination email system, comprising:

transmitting information about the source email system client computing device to a migration coordination system;

receiving a command from the migration coordination system to migrate email system data from the source email system client computing device to the destination email system; and

copying data from a data store on the source email system client computing device to a cached data store that is configured to automatically synchronize data with the destination email system.

51. The computer-readable storage medium of claim 50, wherein the synchronization of the cached data store with the destination email system is performed in a background process.

52. The computer-readable storage medium of claim 50, further comprising:

receiving an indication of a maximum overall size for data to be migrated to the destination email system;

determining, by the processor, the total size of all data to be migrated from the distinguished source email system client computing device; and

if the total size of all data to be migrated from the distinguished source email system client computing device exceeds the maximum overall size for data to be migrated to the destination email system, migrating a portion of the data according to a schedule prioritizing data to be migrated.

53. The computer-readable storage medium of claim 52, wherein determining the total size of all data to be migrated from the distinguished source email system client computing device comprises determining the size of data that matches scope and date criteria for the migration.

54. The computer-readable storage medium of claim 52, wherein, for an email system having Inbox and Sent Items folders, migrating a portion of the data according to a schedule prioritizing data to be migrated comprises:

if the combined size of the Inbox and Sent Items folders is less than the maximum overall size, migrating only the Inbox and Sent Items data to the destination email system; and

if the combined size of the Inbox and Sent Items folders is greater than the maximum overall size, migrating items from the Inbox and Sent Items until the size limit is reached.

55. The computer-readable storage medium of claim 52, further comprising:

creating an archive data store;

copying data exceeding the maximum overall size that is not migrated to the destination email system to the archive data store; and

configuring the destination email system to make the archive data store accessible from the destination email system.

56. A computer-readable storage medium storing computer-executable instructions for causing a computing system having a processor and a network connection to perform a method for managing migration data transfer bandwidth, comprising:

testing, by the processor, the network connection outbound bandwidth;

determining a migration data transfer bandwidth limitation; and

managing migration data transfer bandwidth over the network connection according to the determined migration data transfer bandwidth limitation.

57. The computer-readable storage medium of claim 56, wherein testing the network connection outbound bandwidth comprises uploading a known amount of data to a specific endpoint on the Internet.

58. The computer-readable storage medium of claim 56, wherein determining a migration data transfer bandwidth limitation comprises:

establishing a baseline network connection outbound bandwidth for Internet communication other than migration data transfer; and

subtracting the baseline network connection outbound bandwidth from the tested network connection outbound bandwidth.

59. The computer-readable storage medium of claim 58, wherein the baseline network connection bandwidth is time-dependent.

60. The computer-readable storage medium of claim 56, wherein managing migration data transfer bandwidth comprises establishing no more than a restricted number of concurrent persistent protocol-specific TCP connections for transferring data over the network connection.

61. The computer-readable storage medium of claim 60, wherein the restricted number of connections is time-dependent.

62. The computer-readable storage medium of claim 56, wherein managing migration data transfer bandwidth comprises using thread communication and blocking techniques to throttle network calls when messages over a threshold size are encountered.

63. The computer-readable storage medium of claim 56, wherein managing migration data transfer bandwidth comprises establishing no more than a restricted number of threads for processing migration data to be transferred over the network connection.

64. The computer-readable storage medium of claim 56, wherein managing migration data transfer bandwidth comprises controlling the timing of migration data transfer operations.

65. The computer-readable storage medium of claim 64, wherein controlling the timing of migration data transfer operations comprises scheduling at least one operation to occur only during off-peak hours.

66. The computer-readable storage medium of claim 64, wherein controlling the timing of migration data transfer operations comprises distributing individual account migrations across the period leading up to a specified migration date.

67. The computer-readable storage medium of claim 64, wherein controlling the timing of migration data transfer operations comprises adjusting the specified migration date.

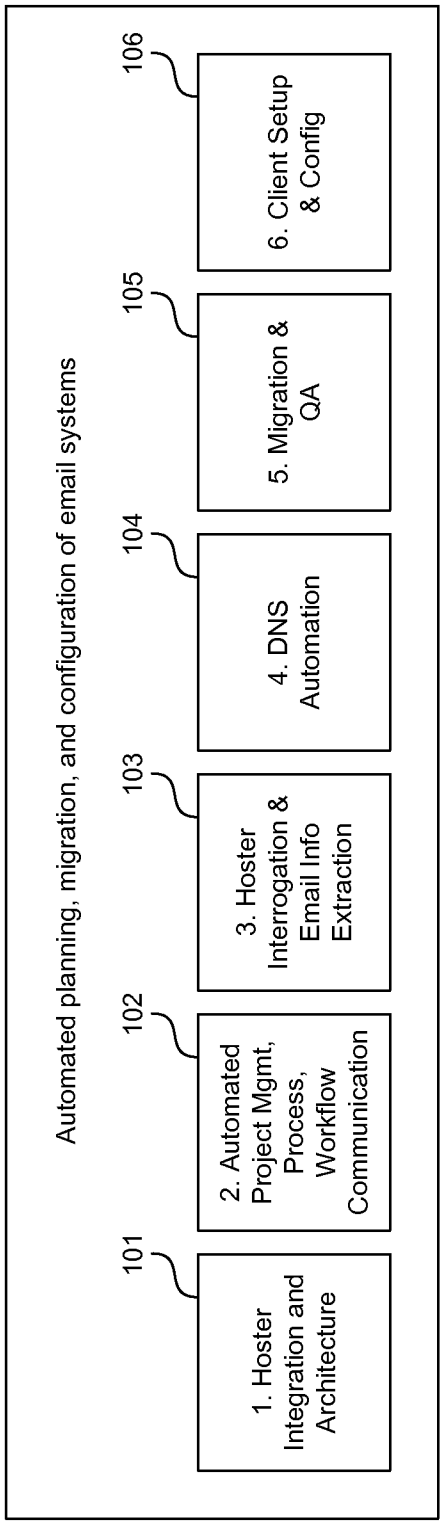

Automated planning, migration, and configuration of email systems
101
1. Hoster Integration and Architecture
102
2. Automated Project Mgmt, Process, Workflow Communication
103
3. Hoster Interrogation & Email Info Extraction
104
4. DNS Automation
105
5. Migration & QA
106
6. Client Setup & Config

FIG. 1

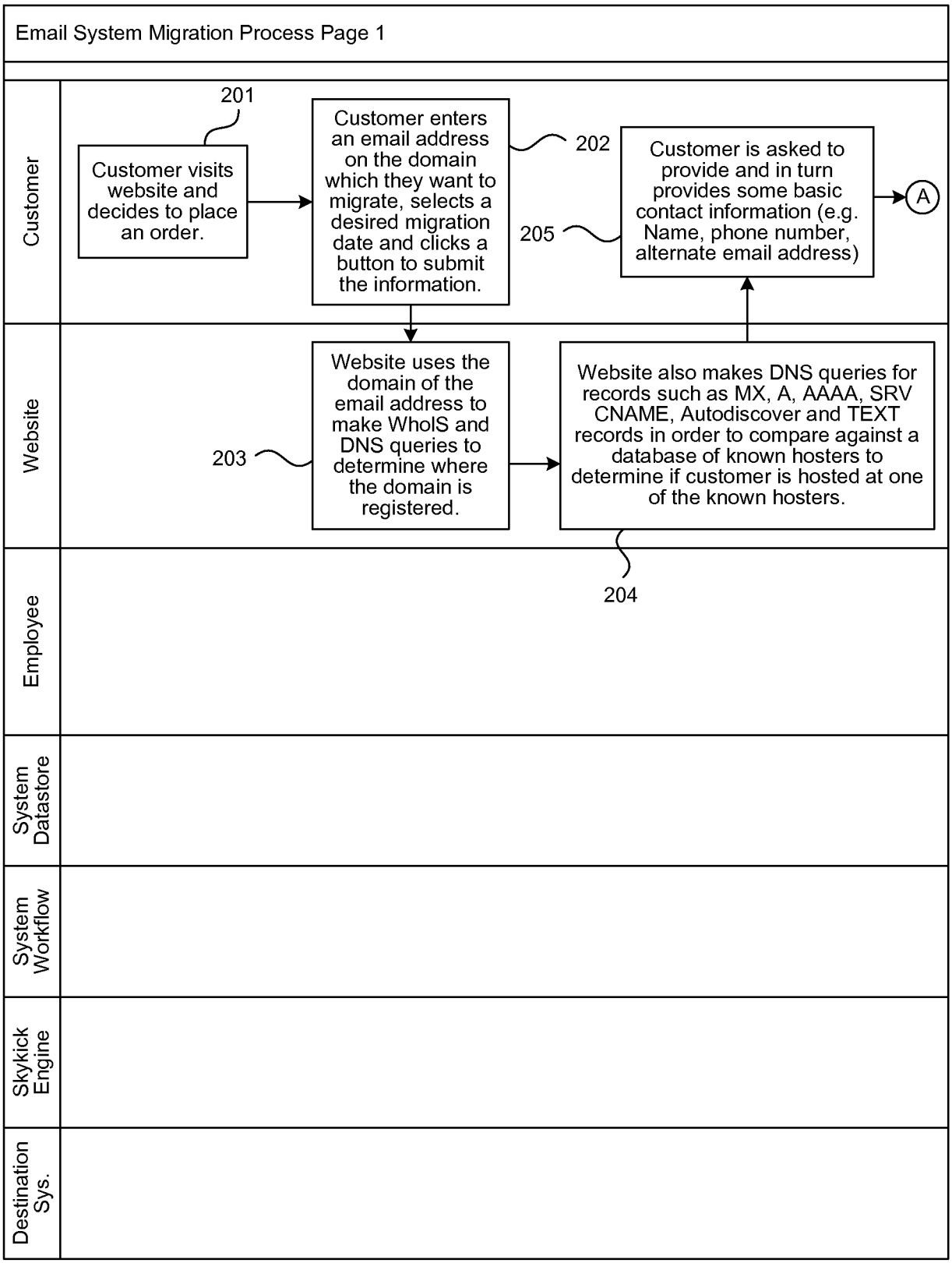

Email System Migration Process Page 1
Customer
Website
Employee
System Datastore
System Workflow
Skykick Engine
Destination Sys.
201
Customer visits website and decides to place an order.
Customer enters an email address on the domain which they want to migrate, selects a desired migration date and clicks a button to submit the information.
202
205
Customer is asked to provide and in turn provides some basic contact information (e.g. Name, phone number, alternate email address)
A
203
Website uses the domain of the email address to make WhoIS and DNS queries to determine where the domain is registered.
Website also makes DNS queries for records such as MX, A, AAAA, SRV CNAME, Autodiscover and TEXT records in order to compare against a database of known hosters to determine if customer is hosted at one of the known hosters.
204

FIG. 2

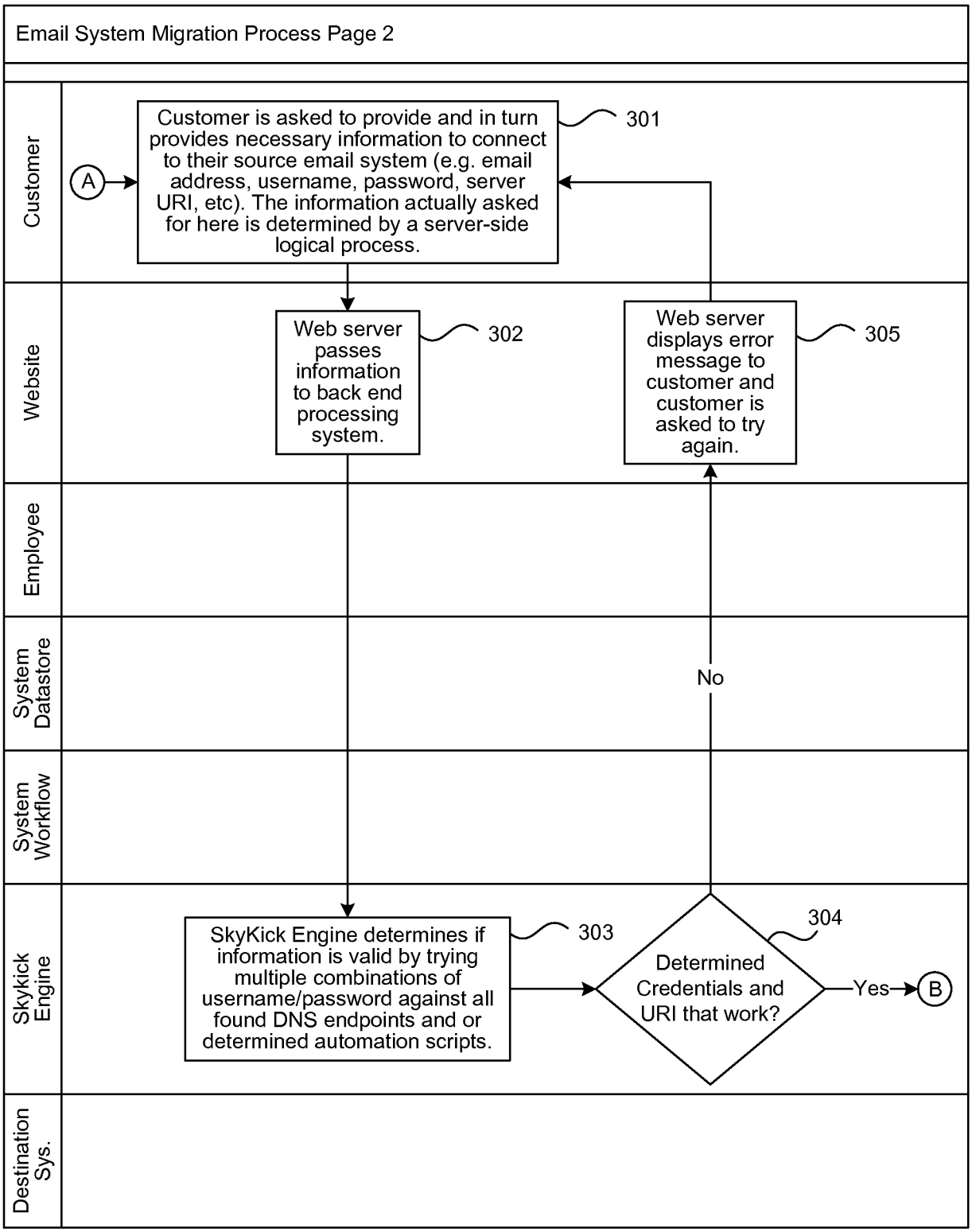

Email System Migration Process Page 2
Customer
Website
Employee
System Datastore
System Workflow
Skykick Engine
Destination Sys.
A
Customer is asked to provide and in turn provides necessary information to connect to their source email system (e.g. email address, username, password, server URI, etc). The information actually asked for here is determined by a server-side logical process.
301
Web server passes information to back end processing system.
302
Web server displays error message to customer and customer is asked to try again.
305
No
SkyKick Engine determines if information is valid by trying multiple combinations of username/password against all found DNS endpoints and or determined automation scripts.
303
Determined Credentials and URI that work?
304
Yes
B

FIG. 3

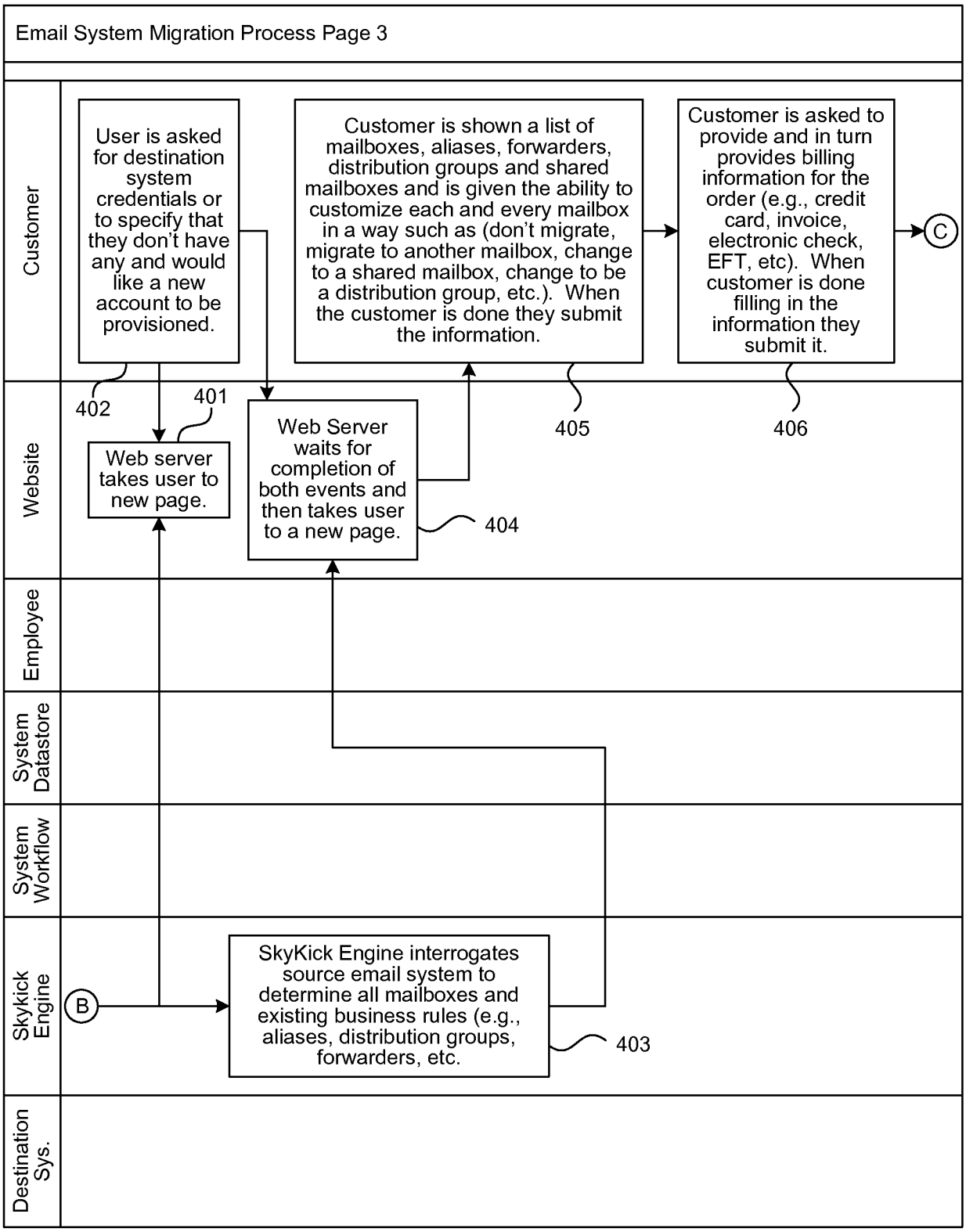

Email System Migration Process Page 3
Customer
Website
Employee
System Datastore
System Workflow
Skykick Engine
Destination Sys.
User is asked for destination system credentials or to specify that they don't have any and would like a new account to be provisioned.
Customer is shown a list of mailboxes, aliases, forwarders, distribution groups and shared mailboxes and is given the ability to customize each and every mailbox in a way such as (don't migrate, migrate to another mailbox, change to a shared mailbox, change to be a distribution group, etc.). When the customer is done they submit the information.
Customer is asked to provide and in turn provides billing information for the order (e.g., credit card, invoice, electronic check, EFT, etc). When customer is done filling in the information they submit it.
C
402
401
405
406
Web server takes user to new page.
Web Server waits for completion of both events and then takes user to a new page.
404
B
SkyKick Engine interrogates source email system to determine all mailboxes and existing business rules (e.g., aliases, distribution groups, forwarders, etc.
403

FIG. 4

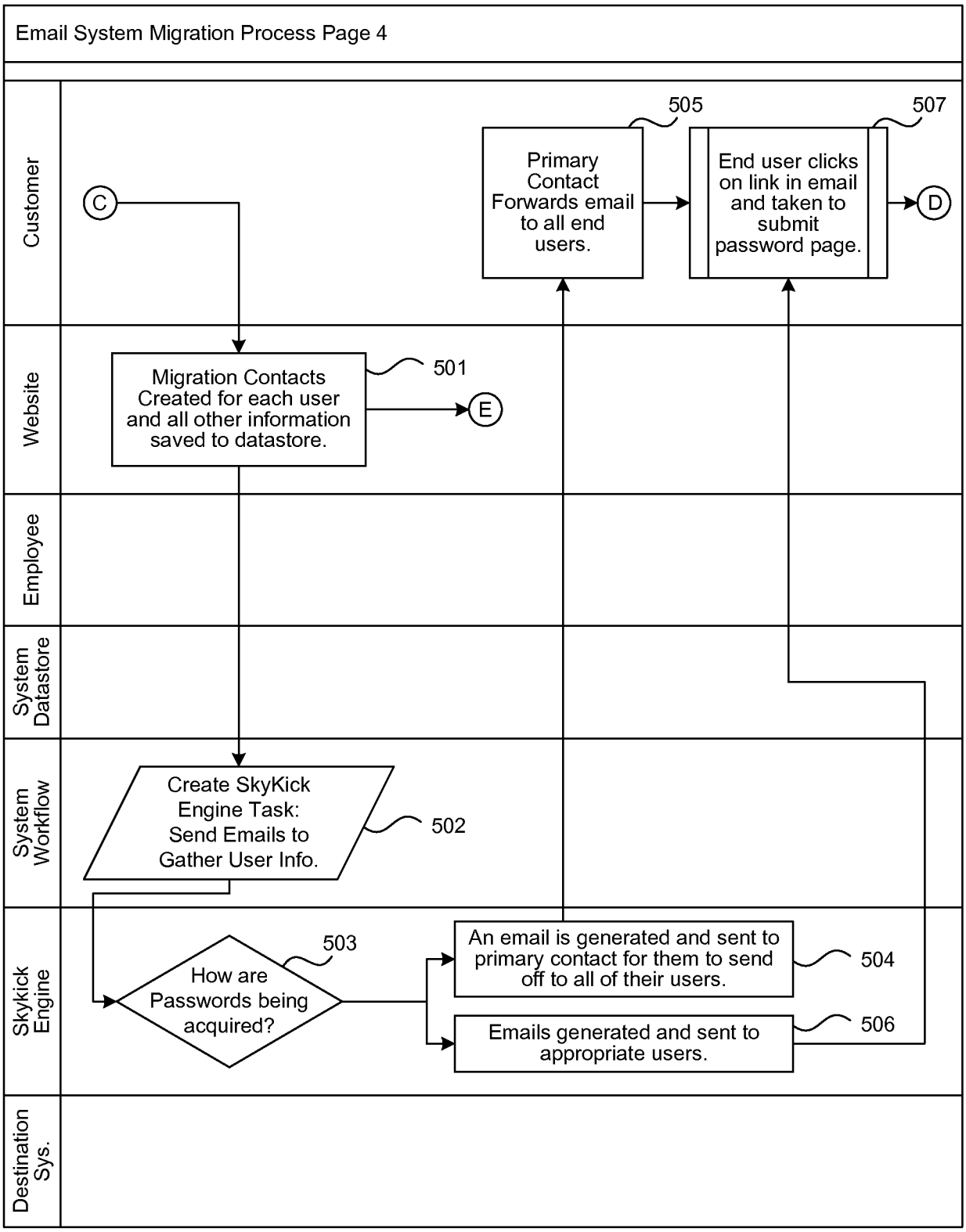

Email System Migration Process Page 4
Customer
Website
Employee
System Datastore
System Workflow
Skykick Engine
Destination Sys.
C
Primary Contact Forwards email to all end users.
505
End user clicks on link in email and taken to submit password page.
507
D
Migration Contacts Created for each user and all other information saved to datastore.
501
E
Create SkyKick Engine Task: Send Emails to Gather User Info.
502
How are Passwords being acquired?
503
An email is generated and sent to primary contact for them to send off to all of their users.
504
Emails generated and sent to appropriate users.
506

FIG. 5

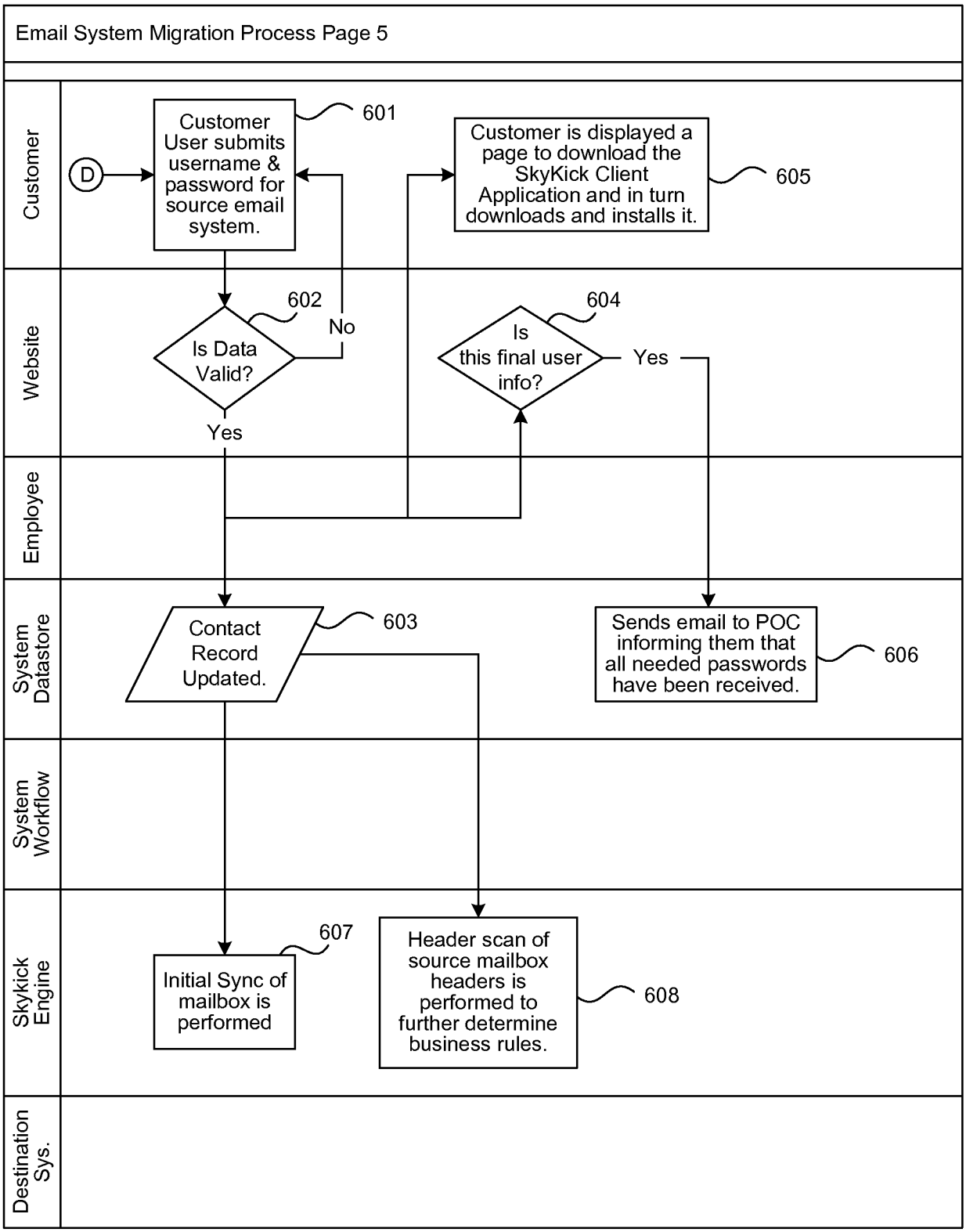

Email System Migration Process Page 5
Customer
Website
Employee
System Datastore
System Workflow
Skykick Engine
Destination Sys.
D
Customer User submits username & password for source email system.
601
Customer is displayed a page to download the SkyKick Client Application and in turn downloads and installs it.
605
Is Data Valid?
602
No
Yes
Is this final user info?
604
Yes
Contact Record Updated.
603
Sends email to POC informing them that all needed passwords have been received.
606
Initial Sync of mailbox is performed
607
Header scan of source mailbox headers is performed to further determine business rules.
608

FIG. 6

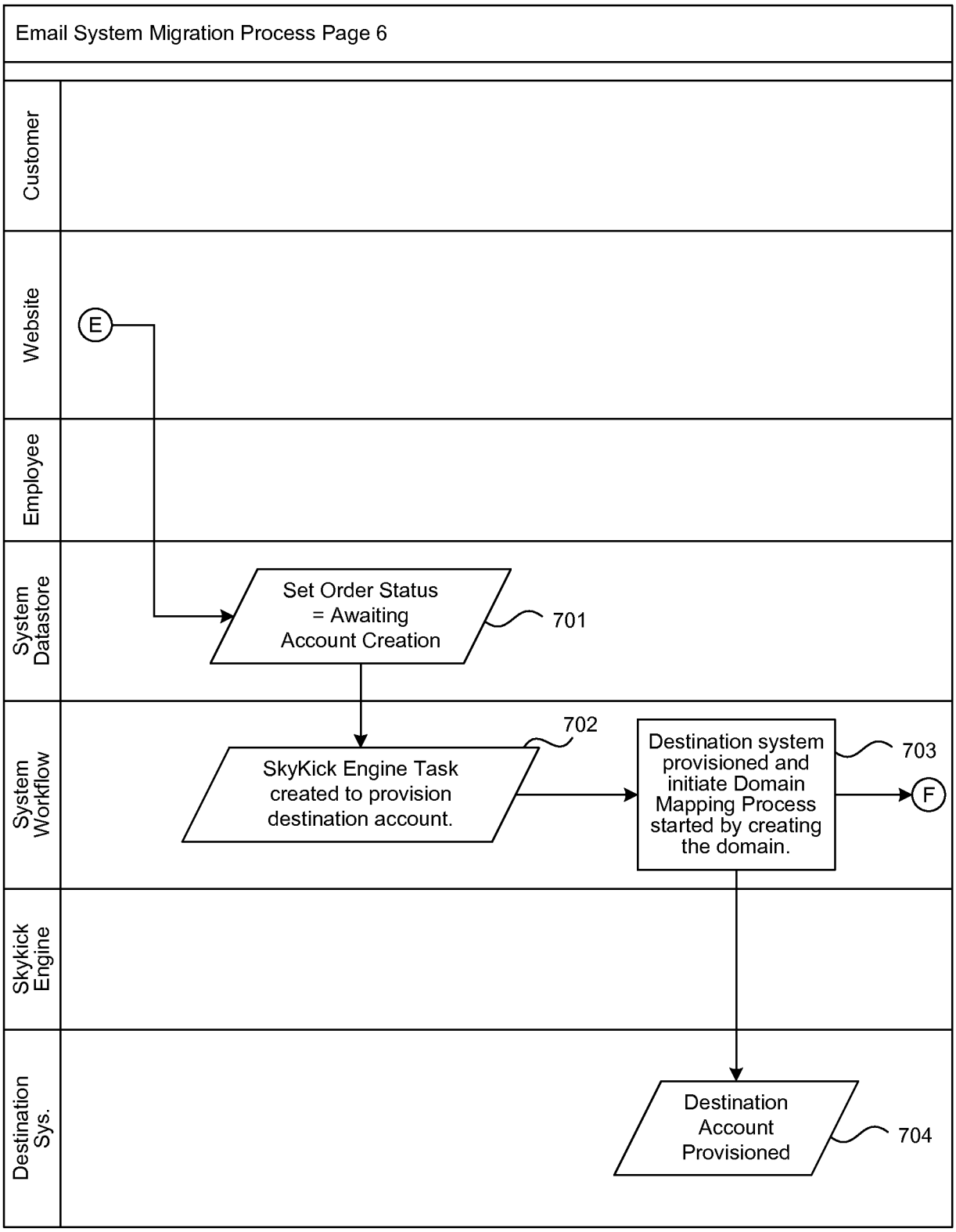

Email System Migration Process Page 6
Customer
Website
Employee
System Datastore
System Workflow
Skykick Engine
Destination Sys.
E
Set Order Status = Awaiting Account Creation
701
SkyKick Engine Task created to provision destination account.
702
Destination system provisioned and initiate Domain Mapping Process started by creating the domain.
703
F
Destination Account Provisioned
704

FIG. 7

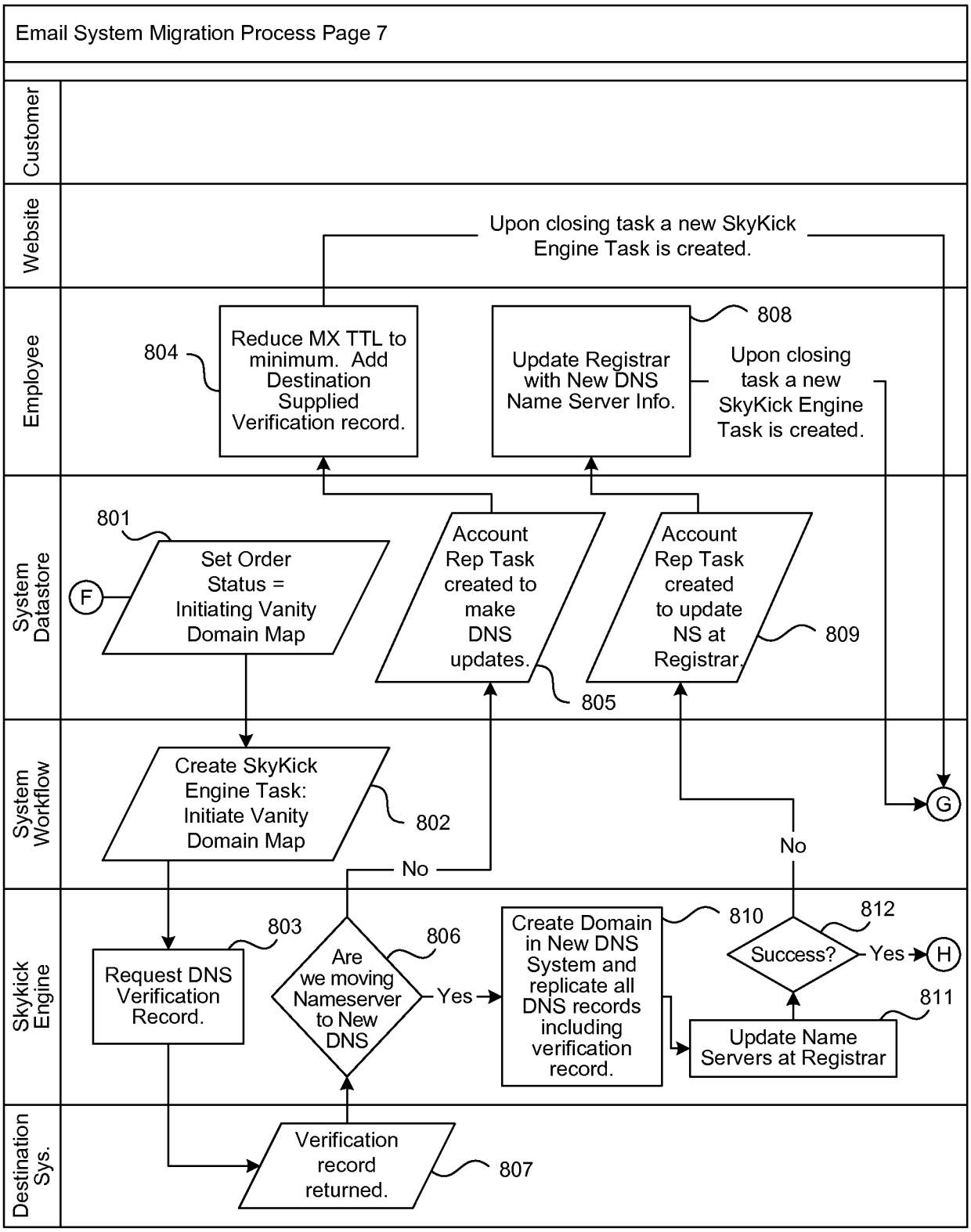

Email System Migration Process Page 7
Customer
Website
Employee
System Datastore
System Workflow
Skykick Engine
Destination Sys.
Upon closing task a new SkyKick Engine Task is created.
804
Reduce MX TTL to minimum. Add Destination Supplied Verification record.
808
Update Registrar with New DNS Name Server Info.
Upon closing task a new SkyKick Engine Task is created.
801
F
Set Order Status = Initiating Vanity Domain Map
Account Rep Task created to make DNS updates.
805
Account Rep Task created to update NS at Registrar.
809
Create SkyKick Engine Task: Initiate Vanity Domain Map
802
No
No
G
803
Request DNS Verification Record.
806
Are we moving Nameserver to New DNS
Yes
Create Domain in New DNS System and replicate all DNS records including verification record.
810
812
Success?
Yes
H
811
Update Name Servers at Registrar
Verification record returned.
807

FIG. 8

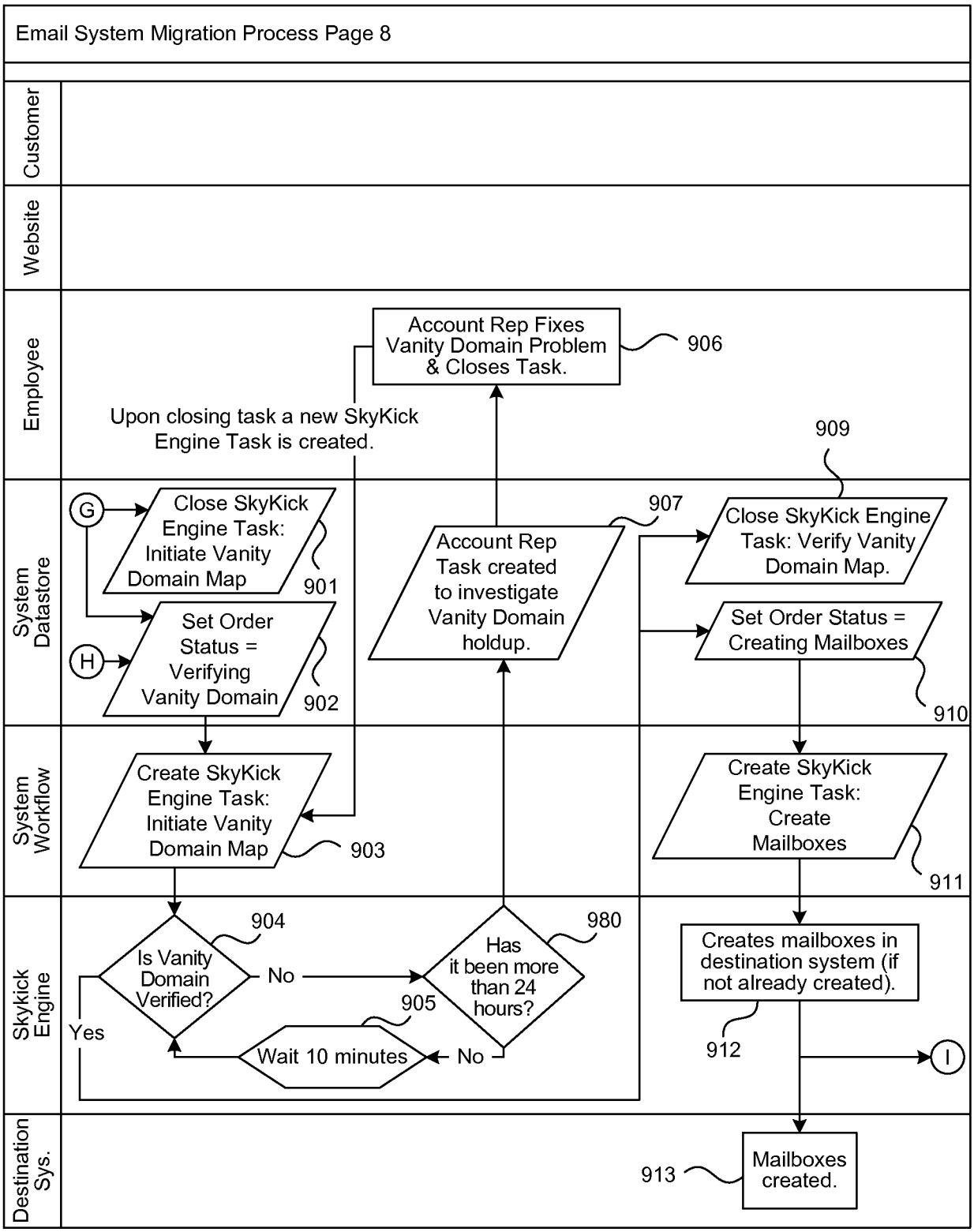

Email System Migration Process Page 8
Customer
Website
Employee
Account Rep Fixes Vanity Domain Problem & Closes Task.
906
Upon closing task a new SkyKick Engine Task is created.
909
System Datastore
G
Close SkyKick Engine Task: Initiate Vanity Domain Map
901
907
Account Rep Task created to investigate Vanity Domain holdup.
Close SkyKick Engine Task: Verify Vanity Domain Map.
H
Set Order Status = Verifying Vanity Domain
902
Set Order Status = Creating Mailboxes
910
System Workflow
Create SkyKick Engine Task: Initiate Vanity Domain Map
903
Create SkyKick Engine Task: Create Mailboxes
911
Skykick Engine
904
Is Vanity Domain Verified?
No
980
Has it been more than 24 hours?
905
Yes
Wait 10 minutes
No
Creates mailboxes in destination system (if not already created).
912
I
Destination Sys.
913
Mailboxes created.

FIG. 9

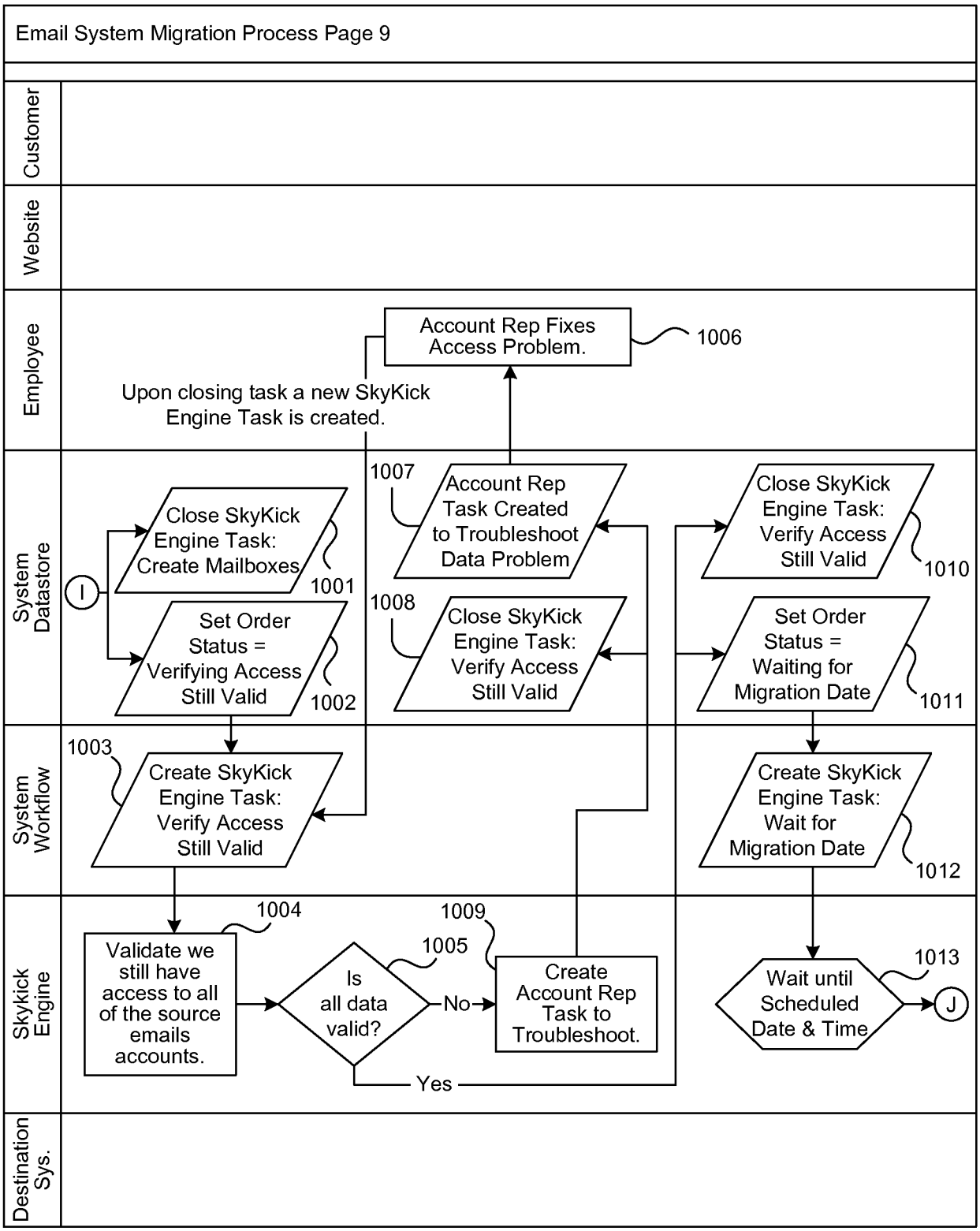

Email System Migration Process Page 9
Customer
Website
Employee
Account Rep Fixes Access Problem.
1006
Upon closing task a new SkyKick Engine Task is created.
System Datastore
Close SkyKick Engine Task: Create Mailboxes
1001
Set Order Status = Verifying Access Still Valid
1002
1007
Account Rep Task Created to Troubleshoot Data Problem
1008
Close SkyKick Engine Task: Verify Access Still Valid
Close SkyKick Engine Task: Verify Access Still Valid
1010
Set Order Status = Waiting for Migration Date
1011
System Workflow
1003
Create SkyKick Engine Task: Verify Access Still Valid
Create SkyKick Engine Task: Wait for Migration Date
1012
Skykick Engine
1004
Validate we still have access to all of the source emails accounts.
1005
Is all data valid?
No
Yes
1009
Create Account Rep Task to Troubleshoot.
1013
Wait until Scheduled Date & Time
J
Destination Sys.

FIG. 10

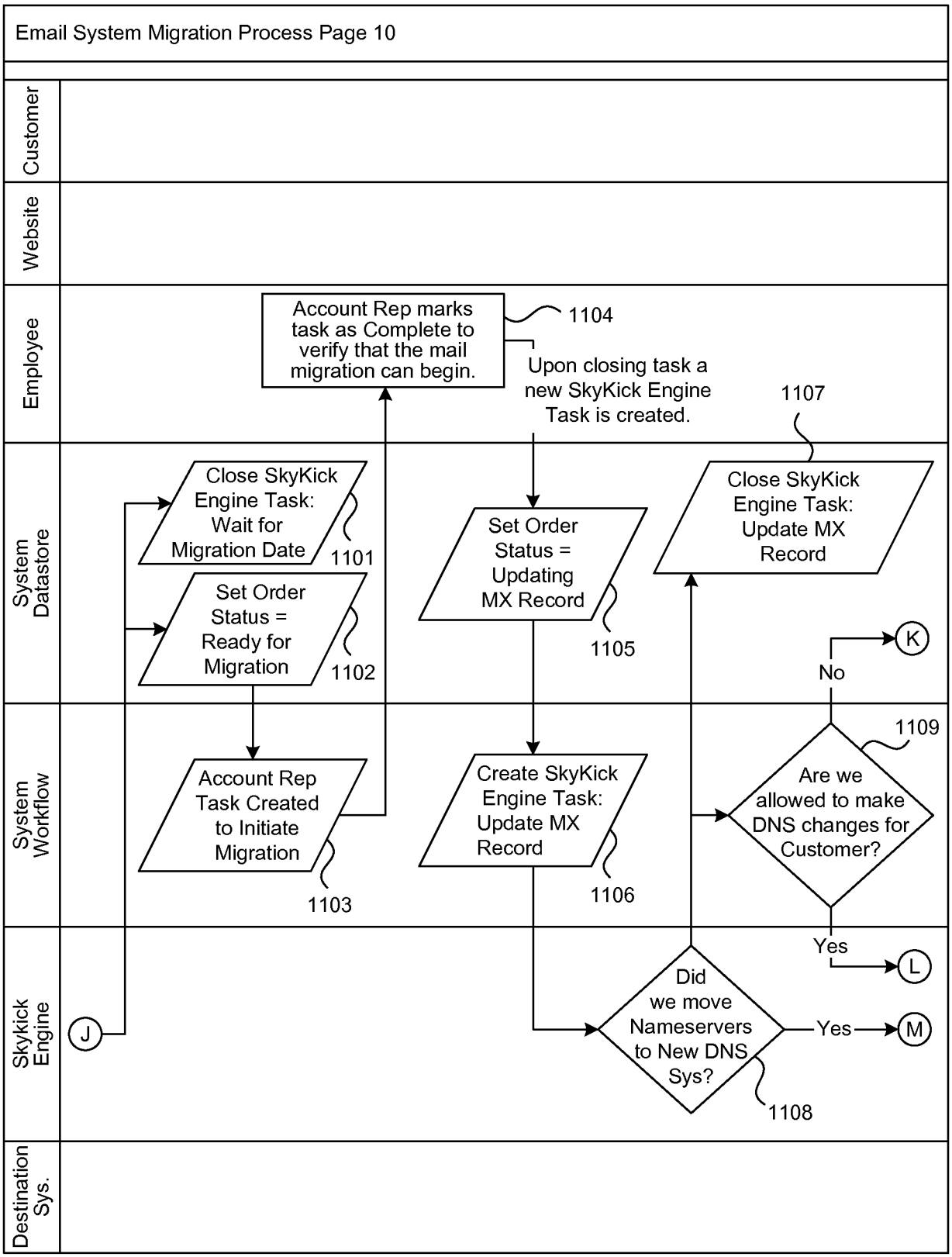

Email System Migration Process Page 10
Customer
Website
Employee
System Datastore
System Workflow
Skykick Engine
Destination Sys.
Account Rep marks task as Complete to verify that the mail migration can begin.
1104
Upon closing task a new SkyKick Engine Task is created.
1107
Close SkyKick Engine Task: Wait for Migration Date
1101
Set Order Status = Ready for Migration
1102
Account Rep Task Created to Initiate Migration
1103
Set Order Status = Updating MX Record
1105
Create SkyKick Engine Task: Update MX Record
1106
Close SkyKick Engine Task: Update MX Record
No
K
1109
Are we allowed to make DNS changes for Customer?
Yes
L
Did we move Nameservers to New DNS Sys?
1108
Yes
M
J

FIG. 11

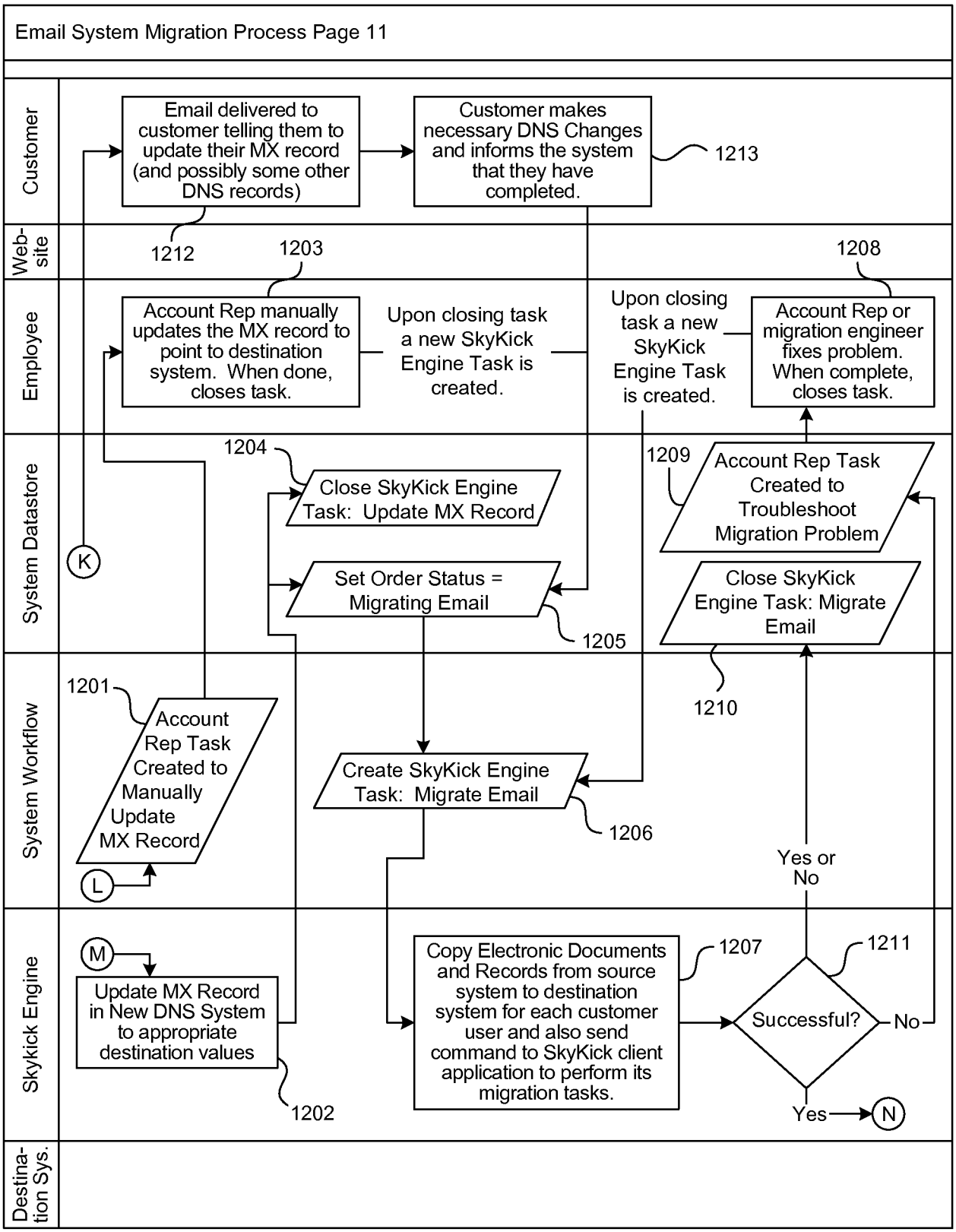

Email System Migration Process Page 11
Customer
Web-site
Employee
System Datastore
System Workflow
Skykick Engine
Destina-tion Sys.
Email delivered to customer telling them to update their MX record (and possibly some other DNS records)
Customer makes necessary DNS Changes and informs the system that they have completed.
1213
1212
1203
1208
Account Rep manually updates the MX record to point to destination system. When done, closes task.
Upon closing task a new SkyKick Engine Task is created.
Upon closing task a new SkyKick Engine Task is created.
Account Rep or migration engineer fixes problem. When complete, closes task.
1204
Close SkyKick Engine Task: Update MX Record
1209
Account Rep Task Created to Troubleshoot Migration Problem
K
Set Order Status = Migrating Email
1205
Close SkyKick Engine Task: Migrate Email
1210
1201
Account Rep Task Created to Manually Update MX Record
Create SkyKick Engine Task: Migrate Email
1206
Yes or No
L
M
Update MX Record in New DNS System to appropriate destination values
1202
Copy Electronic Documents and Records from source system to destination system for each customer user and also send command to SkyKick client application to perform its migration tasks.
1207
1211
Successful?
No
Yes
N

FIG. 12

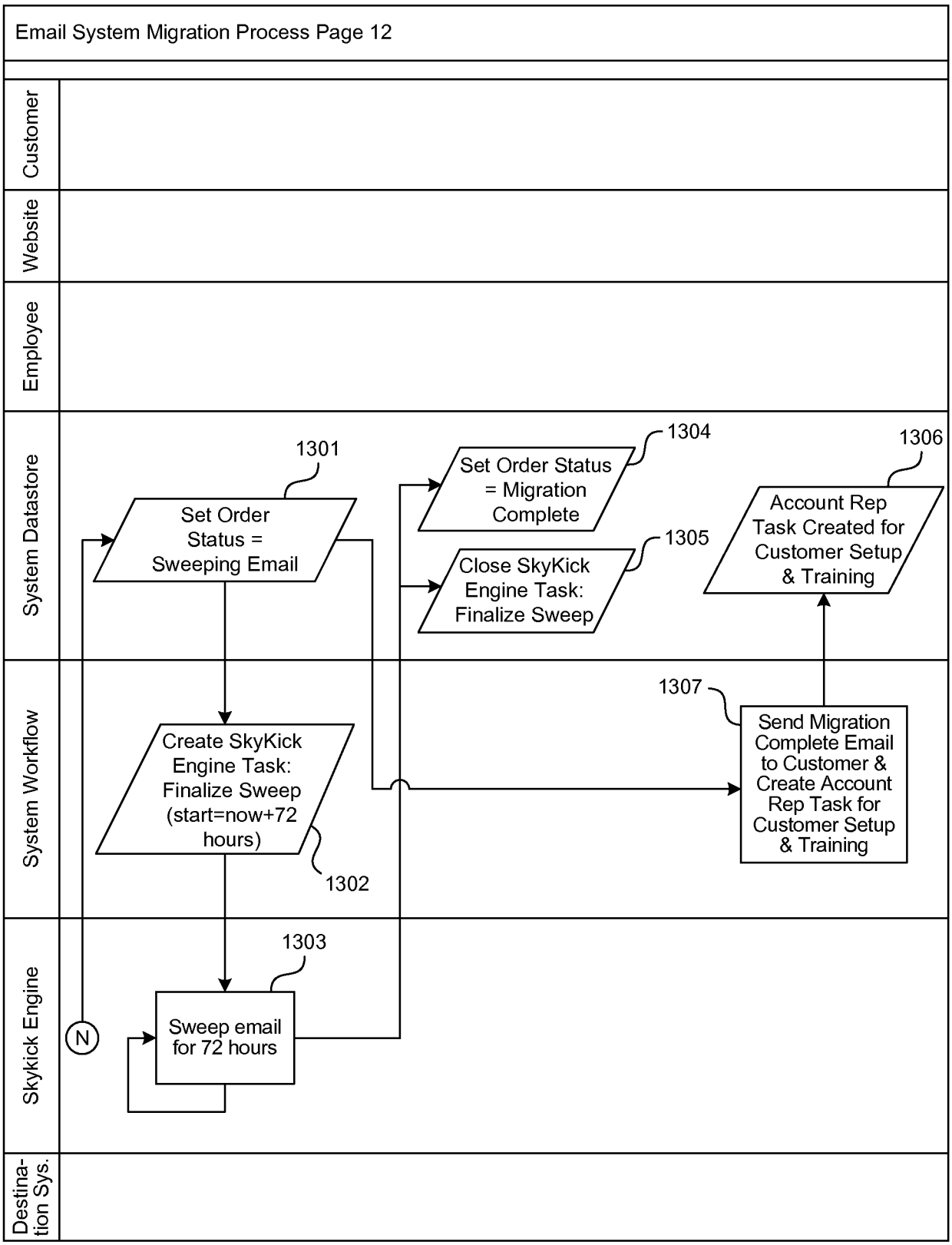

Email System Migration Process Page 12
Customer
Website
Employee
System Datastore
System Workflow
Skykick Engine
Destina-tion Sys.
1301
Set Order Status = Sweeping Email
1304
Set Order Status = Migration Complete
1305
Close SkyKick Engine Task: Finalize Sweep
1306
Account Rep Task Created for Customer Setup & Training
1307
Send Migration Complete Email to Customer & Create Account Rep Task for Customer Setup & Training
Create SkyKick Engine Task: Finalize Sweep (start=now+72 hours)
1302
1303
N
Sweep email for 72 hours

FIG. 13

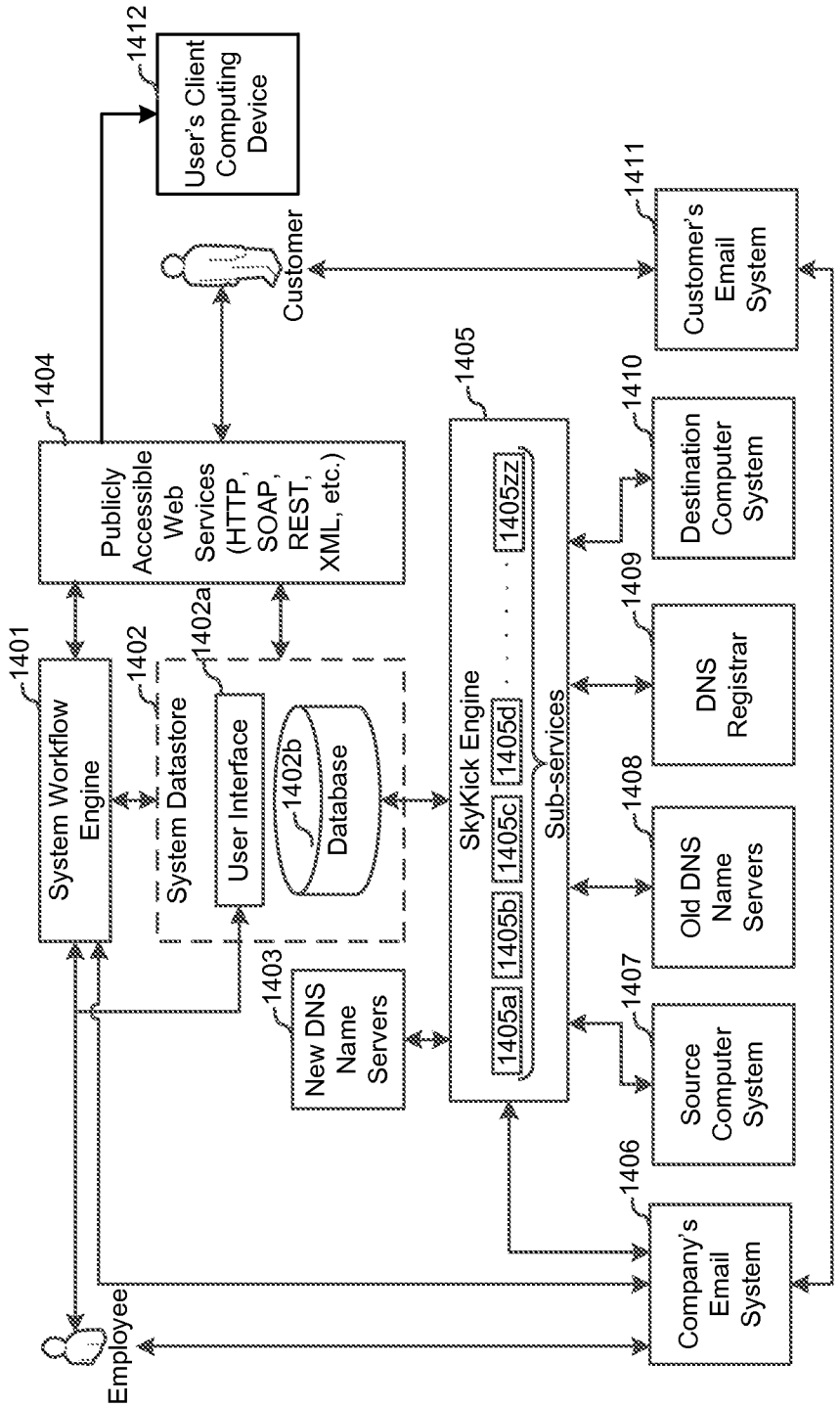

1412
User's Client Computing Device
Customer
1411
Customer's Email System
1404
Publicly Accessible Web Services (HTTP, SOAP, REST, XML, etc.)
1405
1410
Destination Computer System
1401
System Workflow Engine
1402
System Datastore
1402a
User Interface
1402b
Database
SkyKick Engine
1405a
1405b
1405c
1405d
1405zz
Sub-services
1409
DNS Registrar
1403
New DNS Name Servers
1408
Old DNS Name Servers
1407
Source Computer System
1406
Company's Email System
Employee

FIG. 14


GET STARTED
Your Business Email
Number of Employees
Desired Migration Date
GO

FIG. 15

Share Your Info & Admin Access
*Required
Your Info:
Your Name:*
Your Email Address:*
johndoe@biz.com
Your Phone Number:*
Preferred Way to Contact?
O Email
O Phone
O Text
Alternate Contact: (Optional):
Name:
Email Address:
Phone Number:
Words go here:
Desired Migration Date:*
5/25/2012
1601
Based on the business email you gave us we determined:
Email Hoster:
Hosting Company Name
1602
Email Registrar:
Registrar Company Name
1603
In order to lorem ipsum, we will need you to share:
EMAIL HOSTER:
Admin User Name:*
Admin Password:*
1604
EMAIL REGISTRAR:
Admin User Name:*
Admin Password:*
1605
Submit
1606

FIG. 16

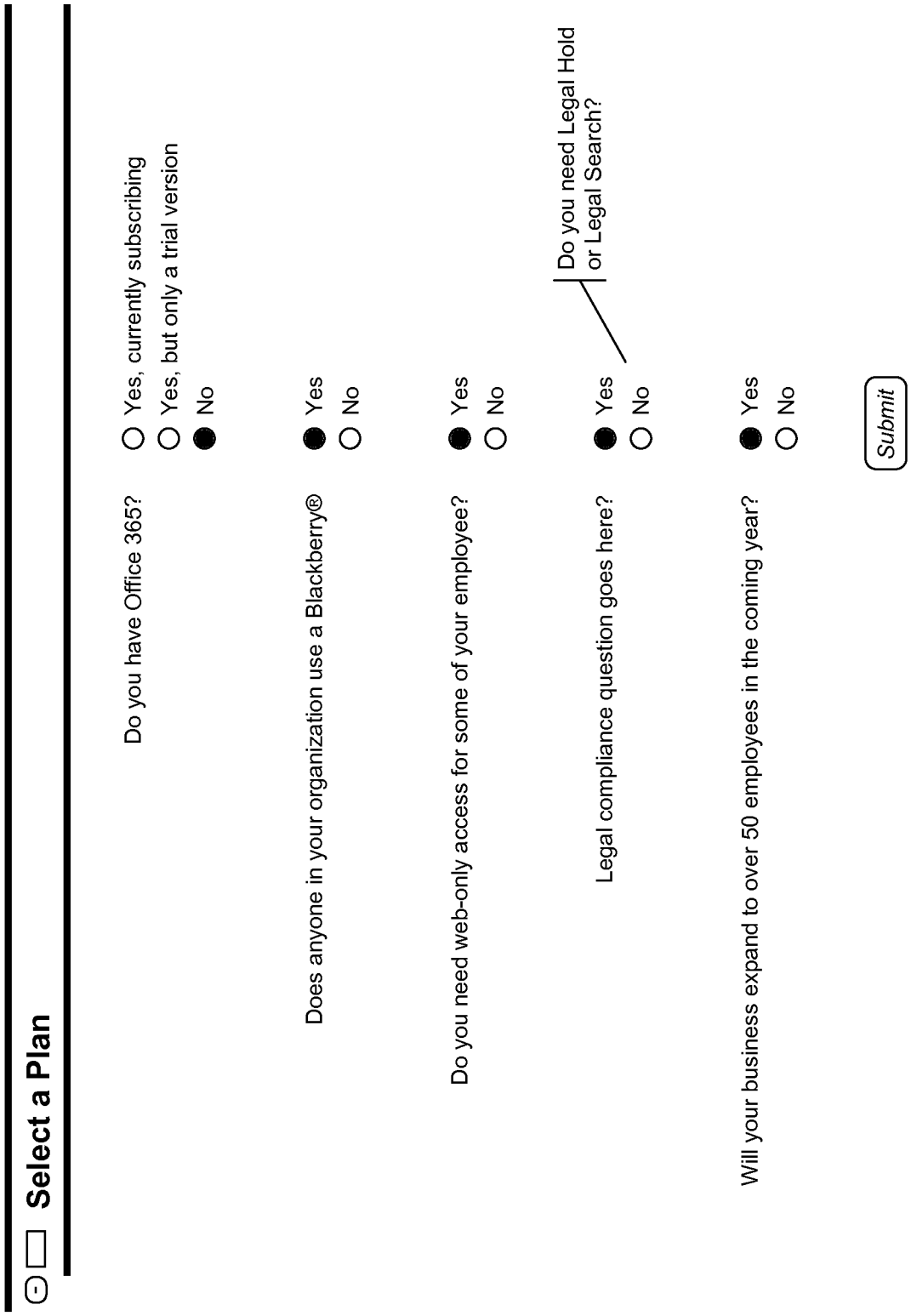

Select a Plan
Do you have Office 365?
Yes, currently subscribing
Yes, but only a trial version
No
Does anyone in your organization use a Blackberry®
Yes
No
Do you need web-only access for some of your employee?
Yes
No
Do you need Legal Hold or Legal Search?
Legal compliance question goes here?
Yes
No
Will your business expand to over 50 employees in the coming year?
Yes
No
Submit

FIG. 17

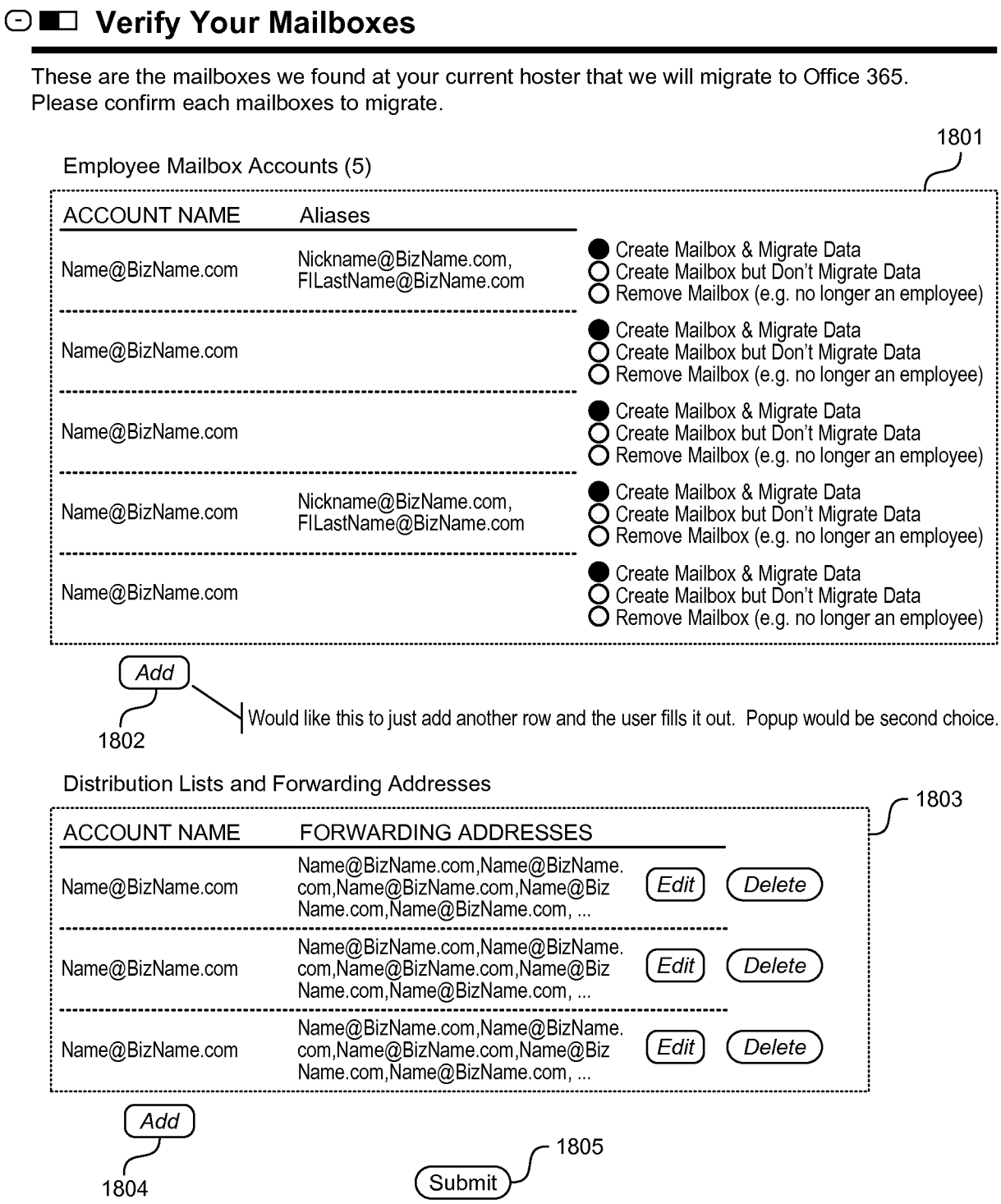

Verify Your Mailboxes
These are the mailboxes we found at your current hoster that we will migrate to Office 365.
Please confirm each mailboxes to migrate.
1801
Employee Mailbox Accounts (5)
ACCOUNT NAME
Aliases
Name@BizName.com
Nickname@BizName.com, FILastName@BizName.com
Create Mailbox & Migrate Data
Create Mailbox but Don't Migrate Data
Remove Mailbox (e.g. no longer an employee)
Name@BizName.com
Name@BizName.com
Name@BizName.com
Nickname@BizName.com, FILastName@BizName.com
Name@BizName.com
Add
1802
Would like this to just add another row and the user fills it out. Popup would be second choice.
Distribution Lists and Forwarding Addresses
1803
ACCOUNT NAME
FORWARDING ADDRESSES
Name@BizName.com
Name@BizName.com,Name@BizName.com,Name@BizName.com,Name@BizName.com,Name@BizName.com, ...
Edit
Delete
Name@BizName.com
Name@BizName.com
Add
1804
1805
Submit

FIG. 18A

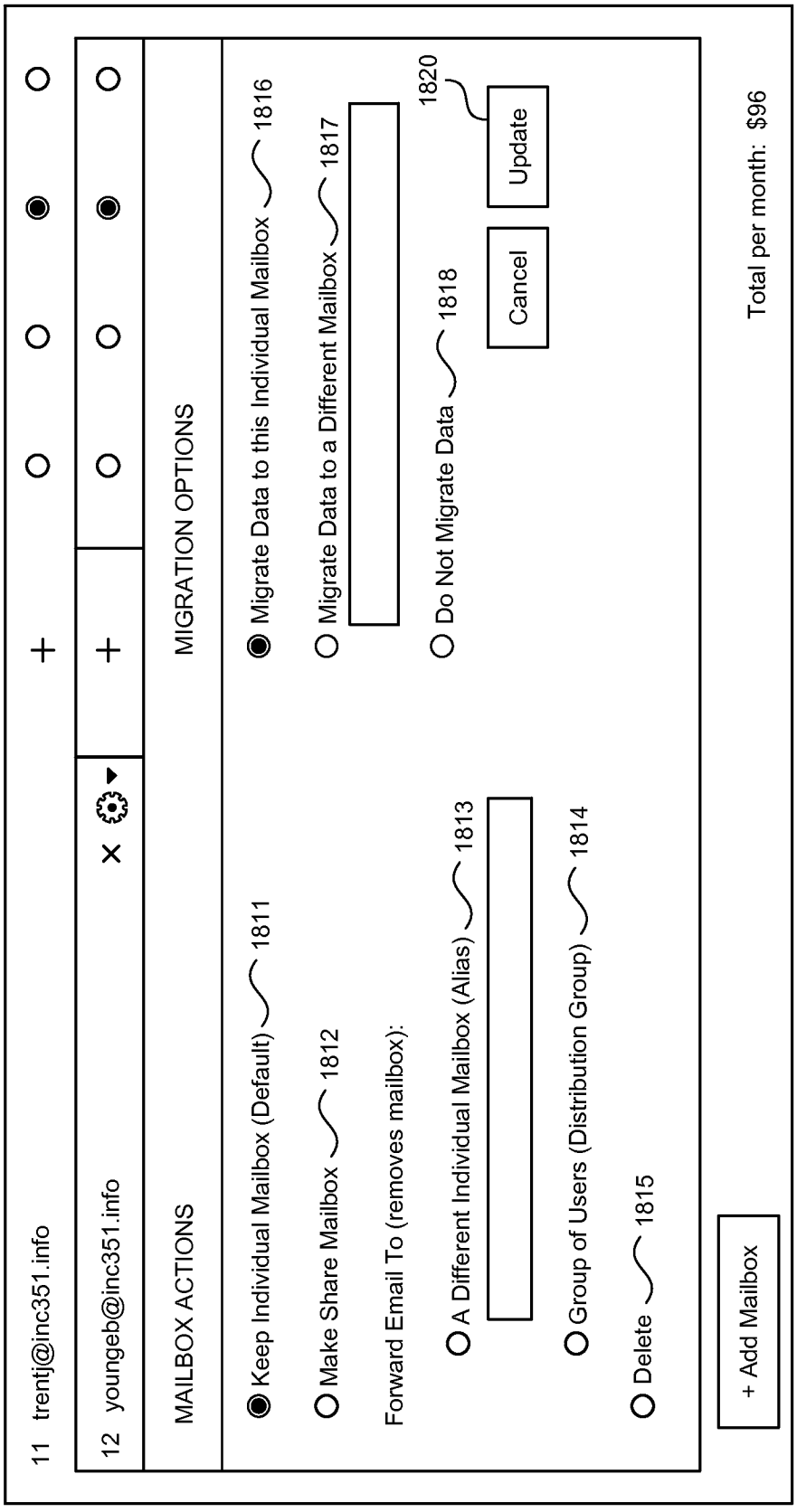

11 trentj@inc351.info
12 youngeb@inc351.info
MAILBOX ACTIONS
MIGRATION OPTIONS
Keep Individual Mailbox (Default) 1811
Make Share Mailbox 1812
Forward Email To (removes mailbox):
A Different Individual Mailbox (Alias) 1813
Group of Users (Distribution Group) 1814
Delete 1815
Migrate Data to this Individual Mailbox 1816
Migrate Data to a Different Mailbox 1817
Do Not Migrate Data 1818
1820
Cancel
Update
+ Add Mailbox
Total per month: $96

FIG. 18B

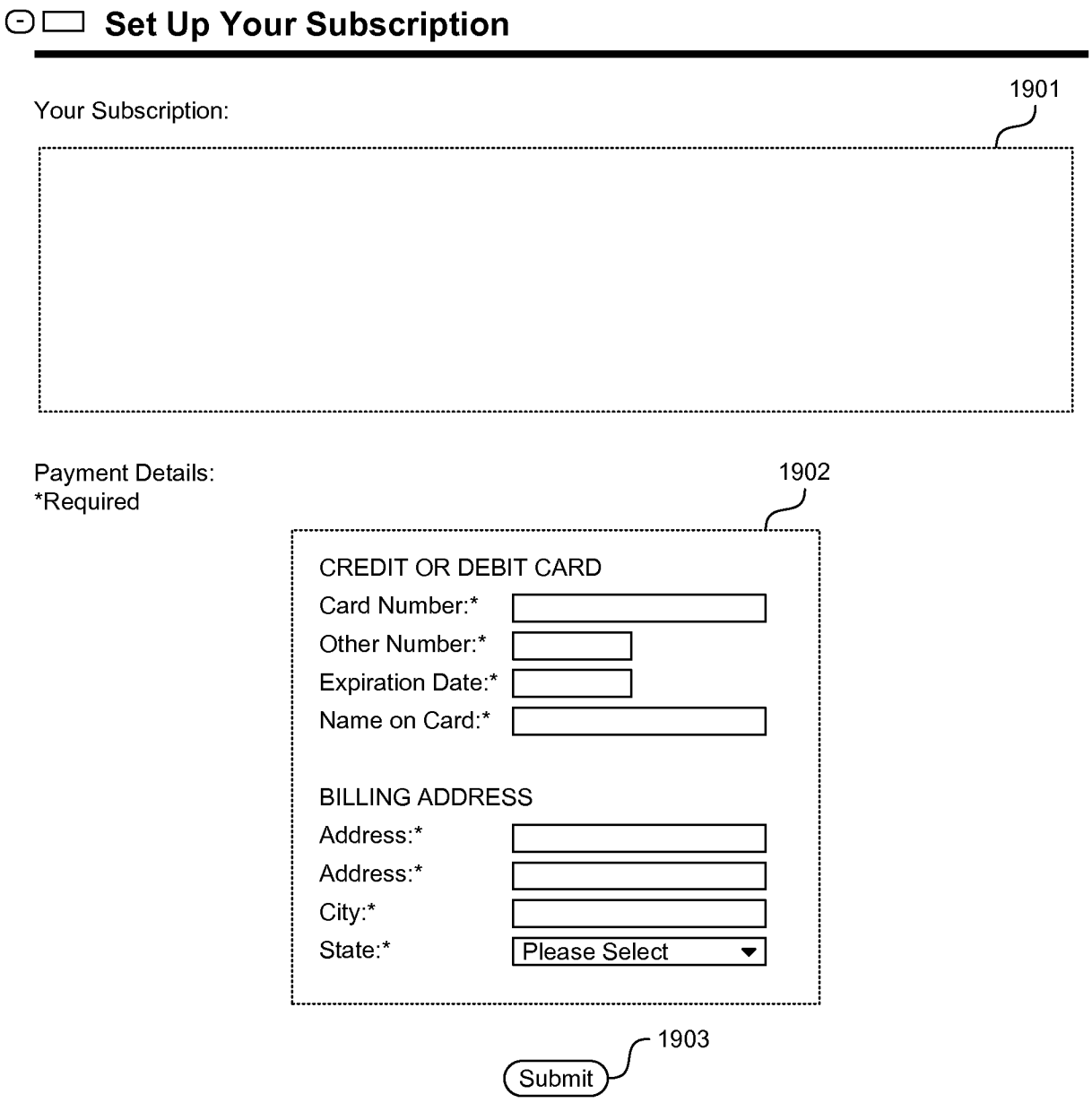

Set Up Your Subscription
Your Subscription:
1901
Payment Details:
*Required
1902
CREDIT OR DEBIT CARD
Card Number:*
Other Number:*
Expiration Date:*
Name on Card:*
BILLING ADDRESS
Address:*
Address:*
City:*
State:*
Please Select
1903
Submit

FIG. 19

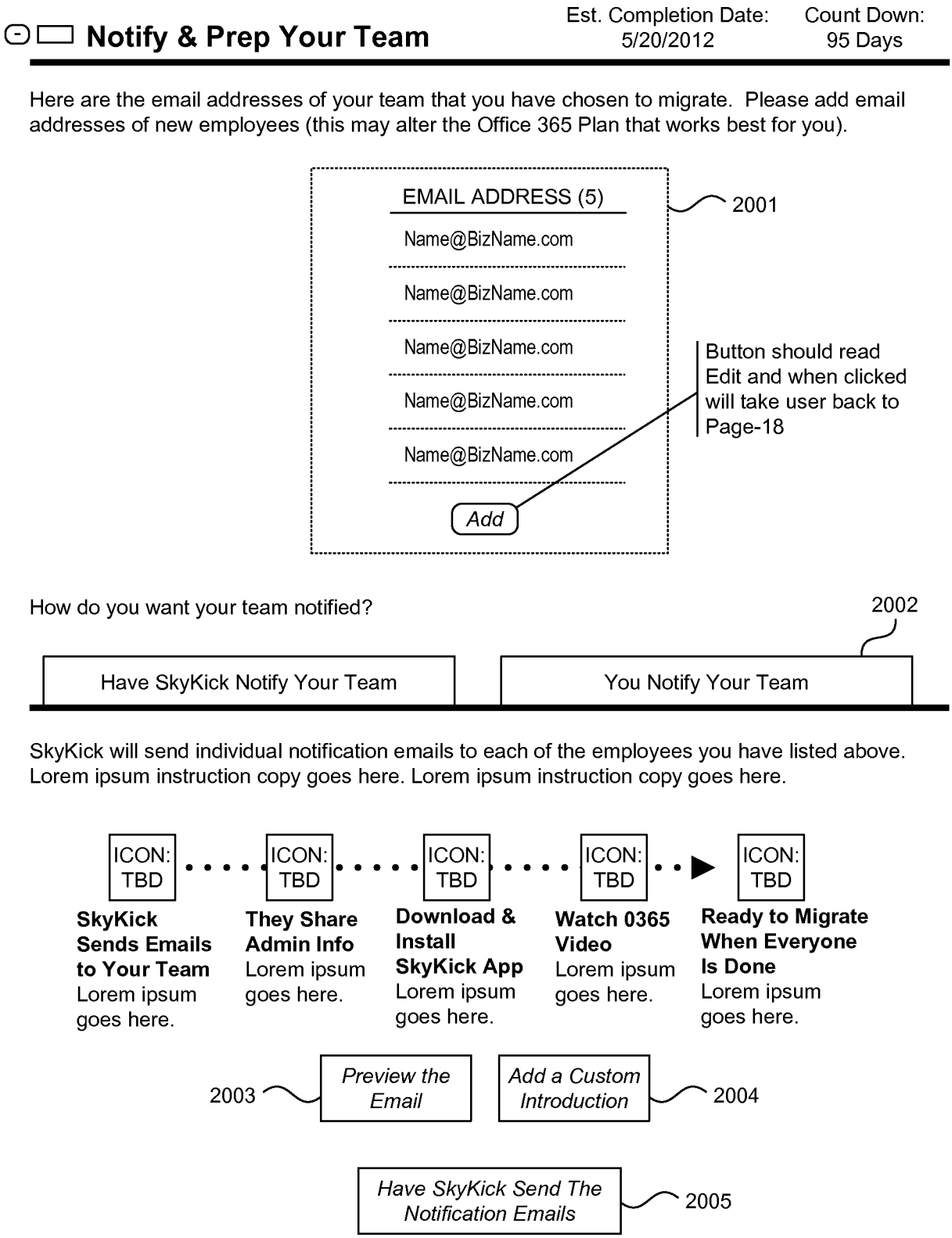

Notify & Prep Your Team
Est. Completion Date: 5/20/2012
Count Down: 95 Days
Here are the email addresses of your team that you have chosen to migrate. Please add email addresses of new employees (this may alter the Office 365 Plan that works best for you).
EMAIL ADDRESS (5)
2001
Name@BizName.com
Name@BizName.com
Name@BizName.com
Name@BizName.com
Name@BizName.com
Add
Button should read Edit and when clicked will take user back to Page-18
How do you want your team notified?
2002
Have SkyKick Notify Your Team
You Notify Your Team
SkyKick will send individual notification emails to each of the employees you have listed above. Lorem ipsum instruction copy goes here. Lorem ipsum instruction copy goes here.
ICON: TBD
SkyKick Sends Emails to Your Team
Lorem ipsum goes here.
They Share Admin Info
Lorem ipsum goes here.
Download & Install SkyKick App
Lorem ipsum goes here.
Watch 0365 Video
Lorem ipsum goes here.
Ready to Migrate When Everyone Is Done
Lorem ipsum goes here.
2003
Preview the Email
Add a Custom Introduction
2004
Have SkyKick Send The Notification Emails
2005

FIG. 20

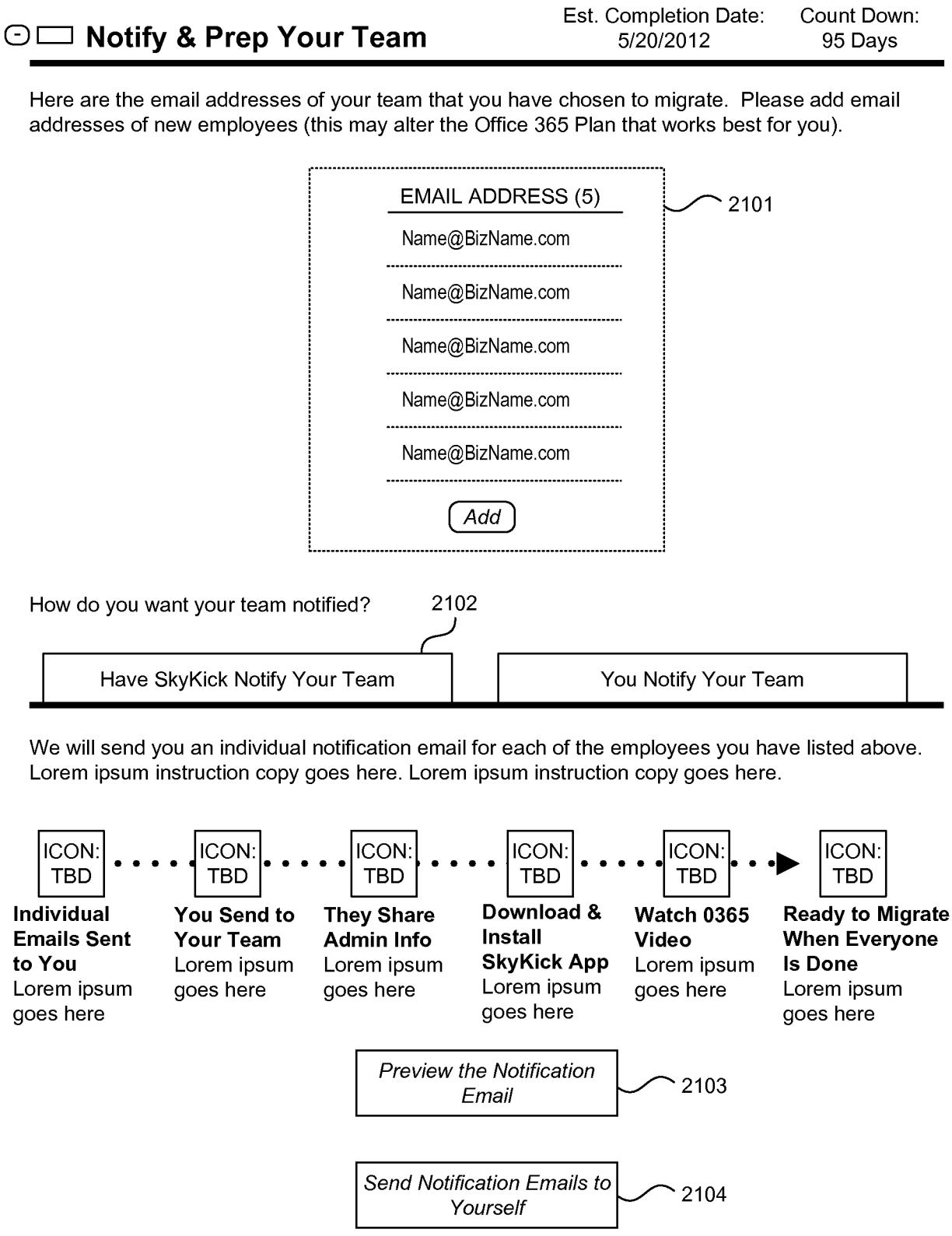

Notify & Prep Your Team
Est. Completion Date:
5/20/2012
Count Down:
95 Days
Here are the email addresses of your team that you have chosen to migrate. Please add email addresses of new employees (this may alter the Office 365 Plan that works best for you).
EMAIL ADDRESS (5)
2101
Name@BizName.com
Name@BizName.com
Name@BizName.com
Name@BizName.com
Name@BizName.com
Add
How do you want your team notified?
2102
Have SkyKick Notify Your Team
You Notify Your Team
We will send you an individual notification email for each of the employees you have listed above. Lorem ipsum instruction copy goes here. Lorem ipsum instruction copy goes here.
ICON: TBD
Individual Emails Sent to You
Lorem ipsum goes here
ICON: TBD
You Send to Your Team
Lorem ipsum goes here
ICON: TBD
They Share Admin Info
Lorem ipsum goes here
ICON: TBD
Download & Install SkyKick App
Lorem ipsum goes here
ICON: TBD
Watch 0365 Video
Lorem ipsum goes here
ICON: TBD
Ready to Migrate When Everyone Is Done
Lorem ipsum goes here
Preview the Notification Email
2103
Send Notification Emails to Yourself
2104

FIG. 21

Track Progress & Approve Migration

Migration Date: 5/25/2012

Count Down: 5 Days

Monitor the progress of you and your team.

2201

EMAIL ADDRESS (5)	OPENED EMAIL	SHARED ACCESS INFO	DOWNLOADED SKYKICK APP	READY TO MIGRATE	+ WATCHED VIDEO
Name@BizName.com	◉	◉	◉	◉	◉
Name@BizName.com	◉	◉	◉	◉	○
Name@BizName.com	◉	◉	◉	◉	◉
Name@BizName.com	◉	◉	◉	◉	◉
Name@BizName.com	◉	◉	◉	◉	◉

Congratulations!

Your team is prepped and ready.

SkyKick is ready to migrate your business to the cloud.
Please authorize by clicking below and we'll tell you when everything is complete.

2202

Schedule Migration

FIG. 22

SUBSTITUTE SHEET (RULE 26)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US13/26000

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - G06F 15/16 (2013.01)
USPC - 709/206
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC(8) Classification(s): G06F 15/16, 15/167; G06Q 10/00 (2013.01)
USPC Classification(s): 709/216, 206, 245

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
MicroPatent (US-G, US-A, EP-A, EP-B, WO, JP-bib, DE-C,B, DE-A, DE-T, DE-U, GB-A, FR-A); DialogPRO; IEEE/IEEEXplore; Google/Google Scholar; IP.com; Search Terms Used: domain name, identify, lookup, database, map, registrar, email, electronic mail, host, provider, server*1, dns

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X -- Y	US 2008/0028100 A1 (ADELMAN, W et al.) January 31, 2008; figures 1, 3; paragraphs [0051], [0057], [0062], [0063], [0066], [0072], [0081], [0082], [0084], [0085]	2, 3, 7 -------- 4-6
Y	US 2007/0073818 A1 (GARDNER, D et al.) March 29, 2007; paragraphs [0022], [0084], [0090]	4-6
Y	US 2011/0004629 A1 (THORAT, N et al.) January 6, 2011; figure 1; paragraphs [0014], [0016]-[0018], [0022], [0023]	1
Y	US 2008/0109448 A1 (ABOEL-NIL, S et al.) May 8, 2008; paragraphs [0057], [0072], [0112], [0123], [0356]	1

☐ Further documents are listed in the continuation of Box C. ☐

*. Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
30 May 2013 (30.05.2013)	21 JUN 2013
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201	Authorized officer: Shane Thomas PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

Form PCT/ISA/210 (second sheet) (July 2009)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US13/26000

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

Group I: Claims 1-7; Group II: Claims 8-22; Group III: Claims 23-33; Group IV: Claims 34-36 and 48-55; Group V: Claims 37-41; Group VI: Claims 42-47; Group VII: Claims 56-67

-***-Please see Supplemental Page-***-

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☒ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:
1-7

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

☐ No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US13/26000

-***-Continued from Box No. III - Observations where unity of invention is lacking-***-

This application contains the following inventions or groups of inventions which are not so linked as to form a single general inventive concept under PCT Rule 13.1. In order for all inventions to be examined, the appropriate additional examination fee must be paid.

Group I: Claims 1-7 are directed toward a computer-readable storage medium storing computer-executable instructions for causing a first computing system having a processor to perform a method for automating a project of data migration from a source email system to a destination email system, comprising: receiving a domain name of a second computing system; obtaining domain name system (DNS) records for the received domain name; determining an email hosting provider for the second computing system; determining, by the processor, based on the obtained DNS records, an email system service type of the source email system on the second computing system; discovering mailboxes and message delivery rules of the source email system; displaying customization options for migrating discovered source email system mailboxes; migrating data items from the source email system to the destination email system; and managing migration of data from source email system client computing devices to the destination email system.

Group II: Claims 8-22 are directed toward a method performed by a first computing system having a processor to identify sufficient email system login credentials for an email system having an email system service type, comprising: obtaining an email address and an email password for the email system; identifying a second computing system that hosts the email system; obtaining domain name system (DNS) records for the second computing system; determining, by the processor, based on the obtained DNS records, an email system service type of the second computing system email system; deriving, based on the obtained email address and email password and the determined email system service type, email system login credentials; and connecting to the email system using the derived email system login credentials.

Group III: Claims 23-33 are directed toward a method performed by a computing system having a processor to discover substantially all mailboxes and message delivery rules in an email system, comprising: obtaining a list of email addresses on the email system; and classifying, by the processor, each of substantially all the obtained addresses as a mailbox, alias, forwarding rule, or distribution list.

Group IV: Claims 34-36 and 48-55 are directed toward a method performed by a computing system having a processor to enable a user to customize an email system migration from a source email system to a destination email system, comprising: displaying a list of all source email system mailboxes that may be migrated; for each source email system mailbox in the list: providing a set of options for routing email addressed to the mailbox after migration to the destination email system; and providing a set of options for migrating mailbox data to the destination email system; and ensuring, by the processor, that selected options do not conflict.

Group V: Claims 37-41 are directed toward a method performed by a computing system having a processor to migrate data items from a source email system to a destination email system, comprising: creating a data structure for recording items copied to the destination email system; performing a first migration pass, comprising: dividing source email system data into batches of item identifiers; retrieving item metadata from the source email system; for each item in a batch to be transferred: assigning a unique item ID to the item; retrieving item content; creating an identification hash for the item; translating the item to an item type compatible with the destination email system; copying the item to the destination email system; saving the assigned unique item ID in the item metadata on the destination email system; and saving in the data structure the identification hash and a reference to the item in the destination email system; and performing a second migration pass, comprising: determining, for each item, whether identical items exist on the source email system and the destination email system; and for each item for which it is determined that identical items do not exist on the source email system and the destination email system, resolving the inconsistency by updating the destination email system.

Group VI: Claims 42-47 are directed toward a computer-readable storage medium storing computer-executable instructions for causing a computing system having a processor to perform a method for domain name system (DNS) record discovery for an identified domain, comprising: creating a list of potential DNS label records; creating a set of discovered DNS records; for each DNS label in the list: performing, by the processor, DNS requests for records matching the label; for each request that returns an alias to a record on the identified domain, recursively performing DNS requests; and for each request that returns an IP address, adding the record to the set of discovered DNS records; identifying wildcard DNS records; and removing DNS records having the same IP address or alias as a wildcard DNS record from the set of discovered DNS records.

Group VII: Claims 56-67 are directed toward a computer-readable storage medium storing computer-executable instructions for causing a computing system having a processor and a network connection to perform a method for managing migration data transfer bandwidth, comprising: testing, by the processor, the network connection outbound bandwidth; determining a migration data transfer bandwidth limitation; and managing migration data transfer bandwidth over the network connection according to the determined migration data transfer bandwidth limitation.

The common technical feature shared by Groups I-VII are a method for automating a project of data migration from a source email system to a destination email system. However, this common feature is previously disclosed by US 7,831,676 B1 (Nagar). Nagar discloses a method for automating a project of data migration from a source email system to a destination email system (make email data accessible to users in the event their primary email system (source email system) becomes unavailable by allowing migration path of other email systems (destination email system, column 9, line 63 to column 10, line 11).

Since the common technical feature is previously disclosed by the Nagar reference, this common feature is not special and so Groups I-VII lack unity.