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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
21 December 2006 (21.12.2006)

PCT

(10) International Publication Number
WO 2006/135508 A2

(51) International Patent Classification:
H04B 7/212 (2006.01)

(21) International Application Number:
PCT/US2006/017579

(22) International Filing Date: 5 May 2006 (05.05.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11/149,650 10 June 2005 (10.06.2005) US

(71) Applicant *(for all designated States except US)*: **MICROSOFT CORPORATION** [US/US]; One Microsoft Way, Redmond, WA 98052-6399 (US).

(72) Inventors: **BRACE, Colin Harrison**; One Microsoft Way, Redmond, WA 98052-6399 (US). **JACK, William, S., III**; One Microsoft Way, Redmond, WA 98052-6399 (US). **MUGGLI, Nathan, Daniel**; One Microsoft Way, Redmond, WA 98052-6399 (US).

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

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

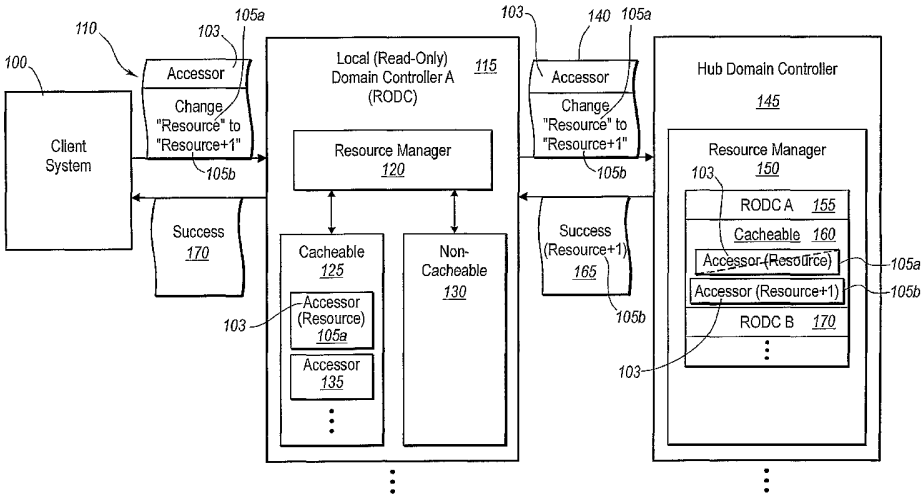
— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: TRANSPARENT RESOURCE ADMINISTRATION USING A READ-ONLY DOMAIN CONTROLLER



(57) Abstract: A domain controller hierarchy in accordance with implementations of the present invention involves one or more local domain controllers, such as one or more read-only local domain controllers in communication with one or more writable hub domain controllers. The local domain controllers includes a resource manager, such as a Security Account Manager ("SAM"), that manages resources and/or other accounts information received from the writable hub domain controller. When a local user attempts to change the resource at the local domain controller, however, the resource manager chains the request, along with any appropriate identifiers for the request, to the writable hub domain controller, where the request is processed. If appropriate, the hub domain controller sends a response that the resource has been updated as requested and also sends a copy of the updated resource to be cached at the local domain controller.

WO 2006/135508 A2

TRANSPARENT RESOURCE ADMINISTRATION USING A READ-ONLY DOMAIN CONTROLLER

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] N/A

1. Technical Field

[0002] This invention relates to systems, methods, and computer program products for managing resources among domain controllers.

2. Background

[0003] As computerized systems have increased in popularity, so have the needs to distribute files and processing resources of computer systems in networks both large and small. In general, computer systems and related devices communicate information over a network for a variety of reasons, for example, to exchange personal electronic messages, sell merchandise, provide account information, and so forth. One will appreciate, however, that as computer systems and their related applications have become increasingly more sophisticated, the challenges associated with sharing data and resources on a network have also increased.

[0004] Generally, there are a number of different mechanisms and protocols for distributing resources among computer systems. For example, two or more computers in a corporate network can share resources, such as files, application programs, or the like, over, for example, a Local Area Network ("LAN"), or a Wide Area Network ("WAN"). The computers can share these resources using any number of currently available transmit and receive communication protocols established between them.

[0005] More complicated schematics for sharing resources on a network include, for example, a domain controller hierarchy scheme, which is used in some implementations to

organize and share both secure and non-secure resources in an efficient manner. For example, a central hub domain controller might be used to manage user names, passwords, computer names, network identifiers, or the like, and provide the information through a hierarchy of remote and local servers (i.e., local domain controllers). The various domain controllers, in turn, are configured with a resource manager, such as a Security Account Manager ("SAM"), which provides interfaces and protocols for storing or authenticating resources in the domain. When one or more individual client computer systems requests a resource, the request may be handled by a local domain controller based on a policy provided by a higher level computer in the domain.

[0006] In one example, a large pharmaceutical company that has several local branch offices (e.g., neighborhood pharmacies), might want to establish a local domain controller at each of the different local pharmacies. The company might do so by establishing a domain controller at each branch office. Each different branch office might therefore be part of a sub-domain in the hierarchy, or might even represent its own individual sub-domain of yet another sub-domain in the company's domain hierarchy. The established domain controller is typically configured for, among other things, operating in accordance with resource guidelines pushed downward from the centralized hub domain controller.

[0007] In this example, the hub domain controller might be "writable" or configured to be written-to by an administrator in the main organization. By contrast, the local domain controllers to which the central writable domain controller connect in this example, however, might be "read-only", and not therefore configured to be written-to in any meaningful way by the local users, or sometimes even the network administrator. In such an example, each local domain controller would be configured primarily to operate in accordance with guidelines provided by the writable domain controller. For user requests, the local domain might also be configured simply to relay the user requests to the writable hub domain controller, and then pass along the relevant account approval information sent

back from the hub domain controller. For example, a user logs onto a client machine, and the local domain controller forwards the request to the hub domain controller for authentication. If the hub domain controller verifies the user's entered information, the hub domain controller instructs the local domain controller to allow the user to logon to the client computer system.

[0008] While this example schematic might have an advantage of being highly centralized, it also has a number of different difficulties, such as a low degree of local configurability (or none at all) for the various local domain controllers. For example, in order for a user to change a password (or reconfigure another resource), the user will usually need to contact an administrator managing the hub domain controller, who will then change the password (or resource) at the hub domain controller before the user can use the new password (or resource) at the local branch. That is, read-only domain controllers are typically unable to chain, or forward, secure account management requests from a user to hub domain controller.

[0009] Furthermore, although minimizing the amount of technical support staff needed at the local branch, this centralized domain controller schematic represents a single point of failure throughout the entire company's network. For example, when the hub domain controller is unavailable for any reason, users at the local branch might be unable to access a certain resource (e.g., logon to their respective client computer systems). That is, the local read-only domain controller will not store the relevant information from the hub domain controller that could be used to validate the client's request(s).

[0010] Alternative implementations to the foregoing examples include local and hub domain controllers that are each writable. Under this type of schematic, the hub domain controller sends not just resource configuration information, but also security account information for the company or organization, such that each local domain controller stores all security accounts for the company. Thus, when any user of the company logs onto any

client machine at any local branch, the local domain controller, rather than the hub domain controller, authenticates the user(s), and provides client computer system access.

[0011] Since resources such as these are stored locally, the user may also be able to configure other resources at the local domain controller, such as the share-ability of a certain file, or access to another file system, or the like. Furthermore, an appropriately authorized user, can also change or update certain security account (or other configuration) information at the local domain controller, as opposed to changing this information by dealing with the network administrator of the hub domain controller. Changes at the user's local domain controller are then propagated throughout the other local and remote domain controllers in the hierarchy.

[0012] While this writable local and remote domain controller schematic has an advantage of decentralizing domain controller configurability and access, this schematic can also present other potential problems, such as security issues, when used in local branch offices that do not have trained technical support staff. For example, there is a heightened risk that one of the local users at a local branch might inadvertently modify a resource at the local domain controller that should not be modified. This is possible in part since many resources in present operating systems now come with a high degree of configuration granularity, which is difficult, if not impossible, for many non-technical branch users to successfully navigate. If there is no local technical support when a configuration mistake is made, the local branch might have to wait until a trained network administrator can fix the problem on-site, or fix it over the company's network as available.

[0013] One can appreciate that the writable local and remote domain controller schematic also presents a variety of exposure concerns. For example, the local domain controllers in the prior example may be accessible by other local domain controllers on the company network, and sometimes also by others on an outside, non-corporate network (e.g., Internet). Furthermore, the writable local domain controllers can present heightened

security risks due to physical intrusion concerns, since, as a practical matter, organizations tend not to place the local domain controllers in physical secure locations. That is, the organization may want the benefits of a writable local domain controller, but not want to expend the resources to guard against virtual and physical security risks. Such physical risks can include removal or replacement of server hardware (e.g., hard drives, and the like).

[0014] As such, a number of difficulties can be found both with schematics with one writable domain controller and those with all domain controllers being writable, particular when implemented in branch locations that are separated from a head office location.

BRIEF SUMMARY OF SELECTED EMBODIMENTS

[0015] The present invention solves one or more of the foregoing problems with systems, methods, and computer program products configured to allow for the handling of security account management through a local read-only domain controller. In particular, implementations of the present invention relate to a local read-only domain controller receiving a request for account management from a local user, and implementing one or more application program interfaces to chain, or forward, the request to a writable hub domain controller for processing. The local domain controller also responds to the local user with a success or failure message, such that the local user is unaware that the writable hub domain controller was actually the computer system handling and processing the actual request.

[0016] For example, one method in accordance with an implementation of the present invention from the read-only domain controller perspective involves receiving a request from an accessor to change a resource. The read-only domain controller then identifies that the request involves a write operation, and forwards the request to a writable hub domain controller. If the writable hub domain controller is able to process the request for a change in the resource successfully, the read-only domain controller will also receive into cache a copy of a changed version of the resource from the writable hub domain controller. Furthermore, the read-only domain controller will send a reply to the accessor that the resource has been changed. Thus, the accessor is unaware that the resource was changed at the writable hub domain controller since the interaction of requesting and granting of request was done primarily between the accessor and the read-only domain controller.

[0017] A method in accordance with an implementation of the present invention from the writable hub domain controller perspective involves receiving a request from a read-only domain controller for a change in a resource, where the request comprises an identifier for

the read-only domain controller. The writable hub domain controller also identifies that the read-only domain controller is authorized to make the request on behalf of an accessor initiating the request, and identifies that the accessor is authorized to change the resource. As appropriate, the writable hub domain controller then changes the resource, and sends a reply message to the read-only domain controller that the resource has been changed.

[0018] Additional features and advantages of exemplary implementations of the invention will be set forth in the description which follows, and in part will be obvious from the description, or may be learned by the practice of such exemplary implementations. The features and advantages of such implementations may be realized and obtained by means of the instruments and combinations particularly pointed out in the appended claims. These and other features will become more fully apparent from the following description and appended claims, or may be learned by the practice of such exemplary implementations as set forth hereinafter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0019] In order to describe the manner in which the above-recited and other advantages and features of the invention can be obtained, a more particular description of the invention briefly described above will be rendered by reference to specific embodiments thereof which are illustrated in the appended drawings. Understanding that these drawings depict only typical embodiments of the invention and are not therefore to be considered to be limiting of its scope, the invention will be described and explained with additional specificity and detail through the use of the accompanying drawings in which:

[0020] Figure 1A illustrates an overview schematic diagram in accordance with an implementation of the present invention having a local read-only domain controller and a writable hub domain controller, wherein a local user is able to manage a resource through the read-only domain controller;

[0021] Figure 1B illustrates the schematic diagram shown in Figure 1A after the resource has been updated, and in which an accessor requests access of the updated resource through the local read-only domain controller; and

[0022] Figure 2 illustrates acts in methods from the local domain controller perspective and from the hub domain controller perspective for handling a local request to change a resource.

DETAILED DESCRIPTION OF EMBODIMENTS

[0023] The present invention extends to systems, methods, and computer program products configured to allow for the handling of security account management through a local read-only domain controller. In particular, implementations of the present invention relate to a local read-only domain controller receiving a request for account management from a local user, and implementing one or more application program interfaces to chain, or forward, the request to a writable hub domain controller for processing. The local domain controller also responds to the local user with a success or failure message, such that the local user is unaware that the writable hub domain controller was actually the computer system handling and processing the actual request.

[0024] Accordingly, and as will be understood more fully from the following description and claims, a domain controller hierarchy can be implemented with writable hub domain controllers that interact with one or more local read-only domain controllers. This provides a number of advantages, at least in part related to security issues at the local level, as well as ease of management. For example, a read-only domain controller is typically less prone to inadvertent or malicious access that could alter configuration resources or comprise sensitive account information. Thus, an organization can realize the benefits of using local domain controllers that have advantages associated with both conventional read-only local domain controllers, and with writable local domain controllers.

[0025] For example, Figure 1A shows a system in accordance with an implementation of the present invention that includes a client computer system 100 connected to a local read-only domain controller A 115 (i.e., "local domain controller A" or "RODC A"). In one implementation, the client computer system 100 and local domain controller A 115 are each found in a branch office (not shown), such as a local neighborhood pharmacy that is part of large pharmaceutical organization. The local domain controller A 115, however,

need not necessarily be located in the same physical location as the client computer system 100, and there may also be more than one local domain controllers located at the given branch location. Figure 1A also shows that a hub domain controller 145 is connected to the local domain controller A 115 over a network. In some cases, the hub domain controller 145 is also located at the branch location, particularly where there may be several levels of hub domain controllers in the domain controller hierarchy. Nevertheless, however, the hub domain controller 145 will normally be present at a remote location.

[0026] Figure 1A also shows that an accessor request 110 is sent from the client computer system 100 to the local domain controller A 115. This can occur in many different ways. For example, a user (not shown) accesses the client computer system 100 and initiates a request to change a resource, such as a password or other such information in the user's account. Alternatively, an application at the client computer system 100 requests a change for the computer systems 100, such as a password change, a change of name on the network, a change of domain (i.e., joining a new domain), or needs to change some other configuration resource. Accordingly, the client computer system 100 sends message 110 via a remote procedure call (or "RPC") on behalf of the relevant accessor 103.

[0027] As shown, message 110 includes an identity for the accessor 103, as well as a request to change the resource, such as changing from "Resource" 105a to "Resource + 1" 105b. The message 110 is then sent to the local domain controller A 115, where one or more application program interfaces make one or more corresponding determinations about the request based on domain policy. For example, a resource manager 120, such as a Security Account Manager (or "SAM") at the local domain controller, identifies that the request 110 involves a "write" operation on a secure resource.. Since the local domain controller is configured primarily for read operations with respect to local accessors, the local domain controller therefore determines that it cannot independently handle the

request 110. That is, the local domain controller determines that it will need to involve a writable domain controller to complete the request.

[0028] The resource manager 120 also identifies that write functions for this type of resource can be "chained", or relayed to a writable hub domain controller.. For example, the resource manager 120 might be configured to chain certain requests to a writable hub domain controller in some situations, but not in others. Furthermore, this configuration information may be unique to how the resource manager 120 is set up at local domain controller A 115. For example, each local domain controller in the domain controller system may be set up to have different chaining (or delegation) policies, such that certain functions at one branch can be chained, while these same functions might not be able to be chained by a different local domain controller.

[0029] In any event, assuming that the request is one that is appropriate or allowed for chaining, the resource manager 120 prepares to send the request to the hub domain controller 145 via message 140. To do so, the resource manager 120 calls a series of one or more handles to help with the request, where the one or more handles target the writable hub domain controller 145. For example, the resource manager 120 calls a handle to communicate with the writable hub domain controller 145. The resource manager 120 also calls one or more handles to write to any relevant accessor and/or resource objects. The resource manager 120 then prepares and passes message 140 to the hub domain controller 145.

[0030]

[0031] Upon performing any appropriate authentication, the hub domain controller 145 then processes message 140 with its own resource manager 150, which involves calling a set of corresponding one or more APIs for the request. In one instance, an API at the resource manager 150 identifies that the local domain controller A 115 from which message 140 is sent corresponds to a partition 155 at the hub domain controller. If the

resource manager 150 does not find any such partition, the hub domain controller 145 replies with an error.

[0032] Nevertheless, assuming the local domain controller sending the request is valid, the resource manager 150 processes the remainder of the request in message 140. In one implementation, for example, this processing includes determining that the local domain controller 115 can make the request on behalf of the accessor. For example, Figure 1A shows that the accessor (e.g., user or client computer system 100) is part of a cacheable partition 160 in the RODC A partition 155. If the requesting accessor were in a non-cacheable group for the partition of RODC A 155, or if the requesting accessor were only in a non-matching cacheable group, such as in a cacheable group for "RODC B 170", the resource manager 150 might deny the request. In any event, because the accessor of Figure 1A is found in cacheable partition 160 for the valid RODC A partition, the resource manager 150 allows the resource 105a to be changed to resource 105b (i.e., to "Resource + 1").

[0033] The resource manager 150 of the hub domain controller 145 then sends a success message 165 back to the local domain controller A 115. In at least one implementation, the resource manager 150 also sends the updated accessor account 105b (i.e., "Resource + 1") back to the local domain controller A 115 so that the resource 105b can be cached at the local domain controller A 115 for subsequent actions by this accessor.

[0034] For example, Figure 1B shows the schematic of Figure 1A after the resource has been updated to resource 105b, and the accessor attempts to access the resource. As shown, client computer system 100 sends a message 180 requesting access of "Resource + 1" on behalf of accessor 103. The resource manager 120 at the local domain controller 115 then takes the message 180 and determines that the request is being processed at a local read-only domain controller 115. The resource manager 120, however, also identifies that the updated resource 105b is cached in the cacheable partition 125. Since

resource 105b is found locally, the resource manager 120 does not have to open up any additional handles or interfaces to interact with the hub domain controller 145, but can simply grant access through message 185. Accordingly, the schematic diagrams in accordance with the present invention illustrate how one or more local read-only domain controllers can be implemented in a manner that provides one or more of the benefits of a writable local domain controller.

[0035] The present invention can also be described in terms of acts for accomplishing one or more methods in accordance with an implementation of the present invention. For example, Figure 2 illustrates flowcharts of series of acts from the perspective of the local domain controller 115 and from the perspective of the hub domain controller 145 for managing resources through a local read-only domain controller and a writable hub domain controller 145. The acts illustrated in Figure 2 are described below with respect to the diagrams of Figures 1A and 1B.

[0036] For example, Figure 2 shows that a method from the perspective of the local read-only domain controller 115 comprises an act 200 of receiving a resource change request. Act 200 includes receiving a request from an accessor to change a local resource. For example, client computer system 100 sends message 110, which bears the identity 105a of an accessor, and requests a change of a resource from "Resource" to "Resource + 1". The message 110 is then received at the local domain controller 115 through a resource manager 120.

[0037] The method also comprises an act 210 of identifying that the request involves a write operation. Act 210 includes identifying that the request involves a write operation on the resource. For example, the resource manager 120 examines the request 110, and determines that the request involves a write operation that is not ordinarily handled by the local (read-only) domain controller. This determination of what operations can and cannot be forwarded is provided in advance by the hub domain controller The resource manager

120 150 authorizes the forwarding of the request to a writable hub domain controller 145 by resource manager 120 at the local read only domain controller 115, subject to any other local or otherwise domain-driven policies.

[0038] Accordingly, Figure 2 also shows that the method comprises an act 220 of forwarding the request to a writable hub. Act 220 includes forwarding the request to a writable hub domain controller. For example, the resource manager 120 prepares a new message 140 that includes information for the local read-only domain controller's identity, and also includes the request information from the message 110, including the accessor identification, as well as the request for a resource change from 105a (i.e., "Resource") to 105b (i.e., "Resource + 1").

[0039] Thus, Figure 2 further shows that a method from the perspective of the hub domain controller 145 comprises an act 230 of receiving a request for a resource change. Act 230 includes receiving a request from a read-only domain controller for a change in a resource, wherein the request comprises an identifier for the read-only domain controller. For example, as shown in Figure 1A, the hub domain controller 145 receives message 140 through the hub's resource manager 150. The resource manager 150 examines the message to determine that the message 140 contains a request for a change in a resource, such as the change in a password, or other account information, or such as the change in a computer name, change in domain, or the like.

[0040] The method from the hub domain controller 145 perspective also comprises an act 240 of identifying that the domain controller is authorized. Act 240 includes identifying that the read-only domain controller is authorized to make the request on behalf of an accessor initiating the request. For example, the resource manager 150 at the hub domain controller 145 identifies a partition 155 for the local domain controller A 115, and identifies that the local domain controller is associated with a cacheable group that includes the accessor 103 making the request. Since the local domain controller A 115

can cache the accessor's 103 resources (i.e., 105a 105b, etc.), the local domain controller A 115 is authorized to send the request 140 on behalf of the accessor. Similarly, therefore, the method from the hub domain controller 145 perspective comprises an act 250 of identifying that the accessor is authorized. Act 250 includes identifying that the accessor is authorized to change the resource. For example, as shown in Figure 1A, the accessor is associated with a resource 105a that is in a partition 160 of cacheable objects for the local domain controller A.

[0041] Figure 2 further shows that the method from the perspective of the hub domain controller 145 comprises an act 260 of changing the resource. For example, upon identifying that the local domain controller 115 and the accessor account are authorized for the request, Figure 1A shows that the resource manager 150 now has an updated resource (i.e., "Resource + 1") for the accessor account, now accessor account 105b.

[0042] Figure 2 also shows that the method from the hub domain controller 145 perspective comprises an act 270 of sending a reply that the resource was changed. Act 270 includes sending a reply message to the read-only domain controller that the resource has been changed. For example, as shown in Figure 1A, hub domain controller 145 sends message 165 to the local domain controller 115, which indicates that the resource 105a has been updated successfully to 105b. In general, the hub domain controller also sends a copy of the updated resource 105b to the local domain controller 115, so that the updated resource 105b can be cached locally in cache 125 (e.g., Figure 1B). In other implementations, however, the hub domain controller might send the updated account 105b separately from an indication of success message.

[0043] In addition, Figure 2 shows that the method from the local domain controller perspective comprises an act 280 of caching a copy of the changed resource. Act 280 includes receiving into cache a copy of a changed version of the local resource from the writable hub domain controller. For example, the resource manager 120 at the local

domain controller receives message 165, which includes the updated resource 105b, and stores the updated resource 105b in cache 125 with other cacheable resources or accounts.

[0044] In addition, Figure 2 shows that the method from the local domain controller perspective comprises an act 290 of sending a reply to the accessor. Act 290 includes sending a reply to the accessor that the local resource has been changed, such that the accessor is unaware that the resource was changed at the writable hub domain controller. For example, as shown in Figure 1A, the local domain controller 115, via the resource manager 120, sends message 170 to the client computer system 100, indicating that the resource 105a has been successfully updated to resource 105b (i.e., "Resource + 1"). At least in part since the messages 110 and 170 were received by or sent from the local domain controller A 115, the accessor 103 (i.e., user, application, or client system 100) is not necessarily aware of the interaction between the local domain controller 115 and the hub domain controller 145.

[0045] The schemes and methods described herein therefore provide a number of mechanisms for implementing a domain controller hierarchy such that a local read-only domain controller has the benefits of conventional read-only local domain controllers as well as the benefits of conventional writable local domain controllers. In particular, the domain hierarchy described herein provides security within a local branch office, at least in part since the local domain controller is read-only. The domain hierarchy described herein also provides expected flexibility since the local domain controller is able to chain resource commands (e.g., changes in secure account information) to a writable hub domain controller.

[0046] One will appreciate that embodiments of the invention include or are incorporated in computer-readable media having computer-executable instructions or related data structures stored thereon. Examples of computer-readable media or computer program products include the volatile or non-volatile storage media, including but not limited to

RAM, ROM, EEPROM, Flash media, CD-ROM, DVD, or other optical or magnetic storage, as well as any corresponding optical or magnetic storage devices, and/or any other media capable of storing electronic computer-executable instructions or related electronic data structures that are capable of being accessed and/or processed by a general purpose or special purpose computerized system. Computer-readable media also encompasses any appropriate combinations of the foregoing.

[0047] Computer-executable instructions comprise, for example, general text instructions in the case of scripts, or compiled instructions in the case of compiled program code, and/or relevant data that are read by one or more components of a general purpose or special purpose computerized system. When read, interpreted, and/or executed, these instructions cause one or more processors of the general purpose or special purpose computerized system (or special purpose processing device) to execute a function or group of functions. As such, computer-executable instructions and associated data structures represent an example of program code means for executing the acts or steps of the invention disclosed herein.

[0048] The present invention may be embodied in other specific forms without departing from its spirit or essential characteristics. The described embodiments are to be considered in all respects only as illustrative and not restrictive. The scope of the invention is, therefore, indicated by the appended claims rather than by the foregoing description. All changes that come within the meaning and range of equivalency of the claims are to be embraced within their scope.

CLAIMS

We claim:

1. At a read-only domain controller in a computerized system in which the local domain controller communicates information to and from a writable hub domain controller, a method of managing resources through the read-only domain controller such that resources can be updated in a simple and secure manner, comprising the acts of:

receiving a request from an accessor to change a resource;

identifying that the request involves a write operation on the resource;

forwarding the request to a writable hub domain controller;

receiving into cache a copy of a changed version of the resource from the writable hub domain controller; and

sending a reply to the accessor that the resource has been changed, such that the accessor is unaware that the resource was changed at the writable hub domain controller.

2. The method as recited in claim 1, wherein the accessor is any of a client computer system, an application installed on the client computer system, or a user.

3. The method as recited in claim 2, wherein the request to change the resource includes any of changing a name of the client computer system on a network, changing a password for the client computer system, or joining a network domain.

4. The method as recited in claim 2, wherein the accessor is a user, and wherein the request includes changing a user name, a password, or a domain change for the user.

5. The method as recited in claim 1, wherein the request is received at the local domain controller by a resource manager configured to manage secure accounts based on one or more policies.

6. The method as recited in claim 5, further comprising:

identifying a local policy that indicates what resources can be chained and what resources cannot be chained to the writable hub domain controller; and

identifying that the requested change in the resource is a request that can be chained to the writable hub domain controller.

7. The method as recited in claim 5, further comprising:

identifying that the request for change in the resource is being handled at a read-only domain controller; and

identifying that the read-only domain controller cannot complete the requested change in the resource since the request involves the write operation.

8. The method as recited in claim 1, further comprising preparing the request in a new message to be sent to the writable hub domain controller, wherein the new message includes an identifier for the local domain controller.

9. The method as recited in claim 1, further comprising receiving a separate success message from the writable hub domain controller that the requested change was processed successfully, such that the changed resource and the success message are received separately.

10. The method as recited in claim 1, further comprising:

receiving a new request from the accessor to access the changed version of the resource;

identifying that the changed version of the resource is stored in cache; and

granting access to the changed version of the resource, such that the writable hub domain controller is not involved in authenticating the new request.

11. At a writable hub domain controller in a computerized system in which the writable hub domain controller communicates information to and from a read-only domain controller, a method of managing resources such that resources can be updated by one or more accessors in a simple and secure manner through the read-only domain controller, comprising the acts of:

receiving a request from a read-only domain controller for a change in a resource,

identifying that the read-only domain controller is authorized to make the request on behalf of an accessor initiating the request;

identifying that the accessor is authorized to change the resource;

changing the resource; and

sending a reply message to the read-only domain controller that the resource has been changed.

12. The method as recited in claim 11, wherein identifying that the read-only domain controller is authorized comprises determining that the resource is cacheable at the read-only domain controller.

13. The method as recited in claim 11, wherein identifying that the accessor is authorized comprises identifying that the accessor is found in a partition of cacheable resources associated with the read-only domain controller.

14. The method as recited in claim 13, wherein identifying that the accessor is authorized further comprises identifying that the resource is associated with the accessor in the partition of cacheable resources.

15. The method as recited in claim 11, wherein sending a reply message that the resource has been changed comprises sending a changed version of the resource to the read-only domain controller, such that the changed version of the resource can be stored in cache at the read-only domain controller.

16. The method as recited in claim 11, further comprising sending local policy information to the read-only domain controller that indicates what resources can be accessed at the writable domain controller through access to the read-only domain controller.

17. The method as recited in claim 11, wherein the accessor is a user or an application installed on a client computer system, the client system being connected to the read-only domain controller.

18. The method as recited in claim 11, wherein the request for a change in a resource comprises a request to change a user password, a computer name, a user name, domain access, file access, or file system access.

19. At a local read-only domain controller in a computerized system in which the local domain controller communicates information to and from a writable hub domain controller, a computer program product having computer-executable instructions stored thereon that, when executed, cause one or more processors of the local read-only domain controller to perform a method comprising the acts of:

receiving a request from an accessor to change a resource;

identifying that the request involves a write operation on the resource;

forwarding the request to a writable hub domain controller;

receiving into cache a copy of a changed version of the resource from the writable hub domain controller; and

sending a reply to the accessor that the resource has been changed, such that the accessor is unaware that the resource was changed at the writable hub domain controller.

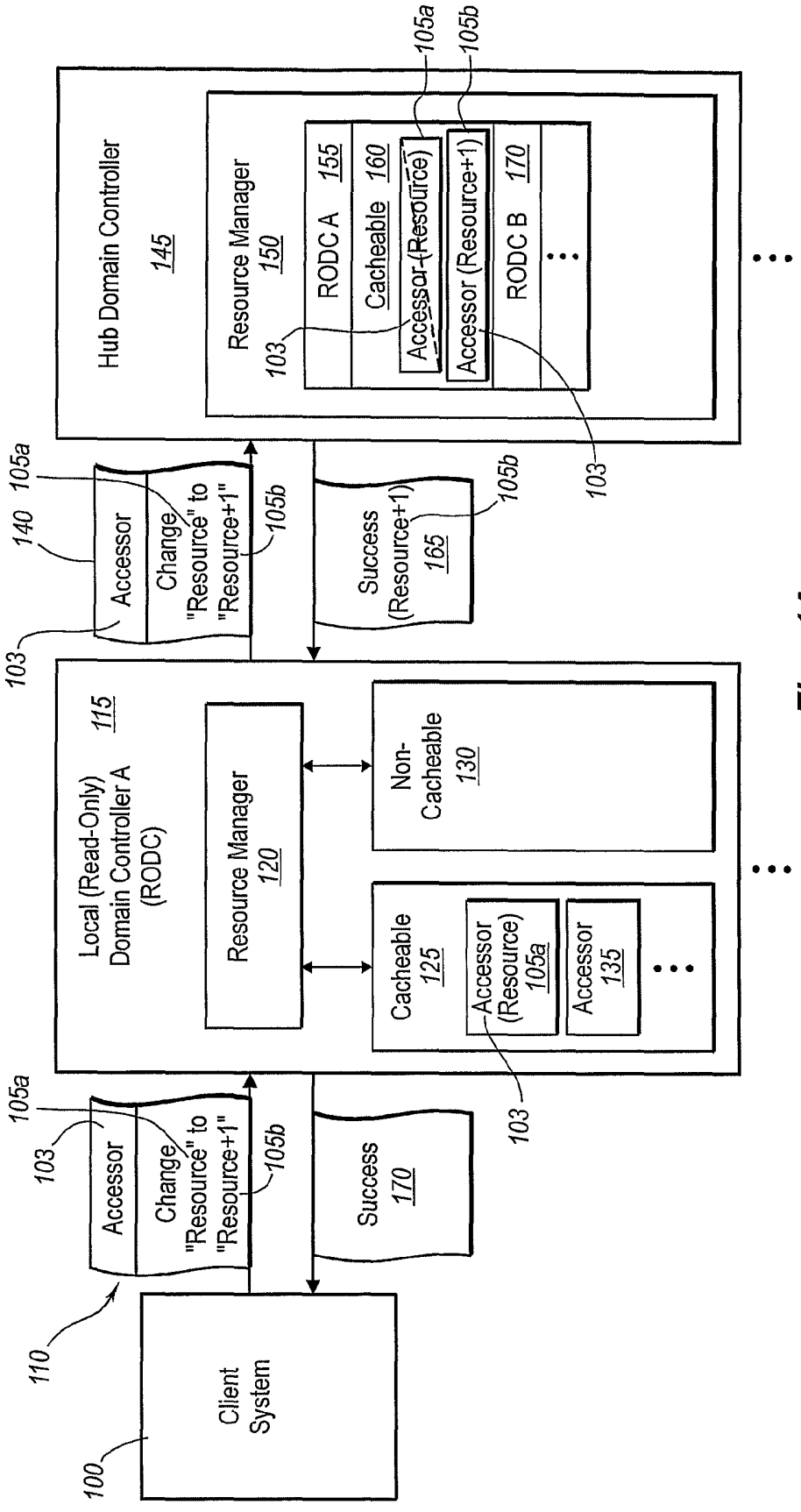

Client System
100
110
Accessor
Change "Resource" to "Resource+1"
105a
105b
103
Success 170
Local (Read-Only) Domain Controller A (RODC) 115
Resource Manager 120
Cacheable 125
Accessor (Resource) 105a
Accessor 135
Non-Cacheable 130
Accessor
Change "Resource" to "Resource+1"
140
Success (Resource+1) 165
Hub Domain Controller 145
Resource Manager 150
RODC A 155
Cacheable 160
Accessor (Resource)
Accessor (Resource+1)
RODC B 170

Fig. 1A

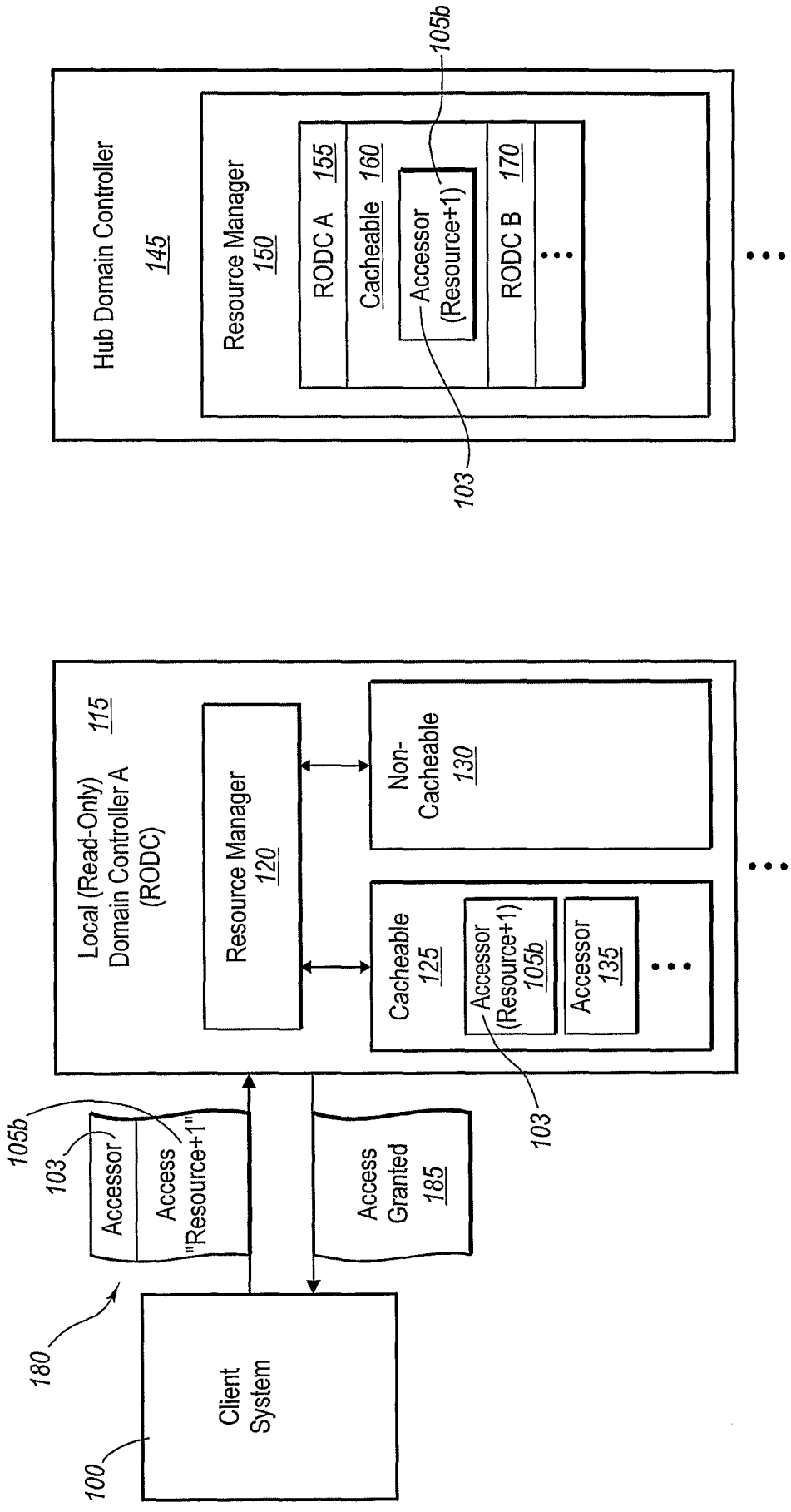

Hub Domain Controller
145
Resource Manager
150
RODC A 155
Cacheable 160
Accessor (Resource+1)
RODC B 170
105b
103
Local (Read-Only) Domain Controller A (RODC) 115
Resource Manager 120
Non-Cacheable 130
Cacheable 125
Accessor (Resource+1) 105b
Accessor 135
Accessor
Access "Resource+1"
Access Granted 185
Client System
100
180

Fig. 1B

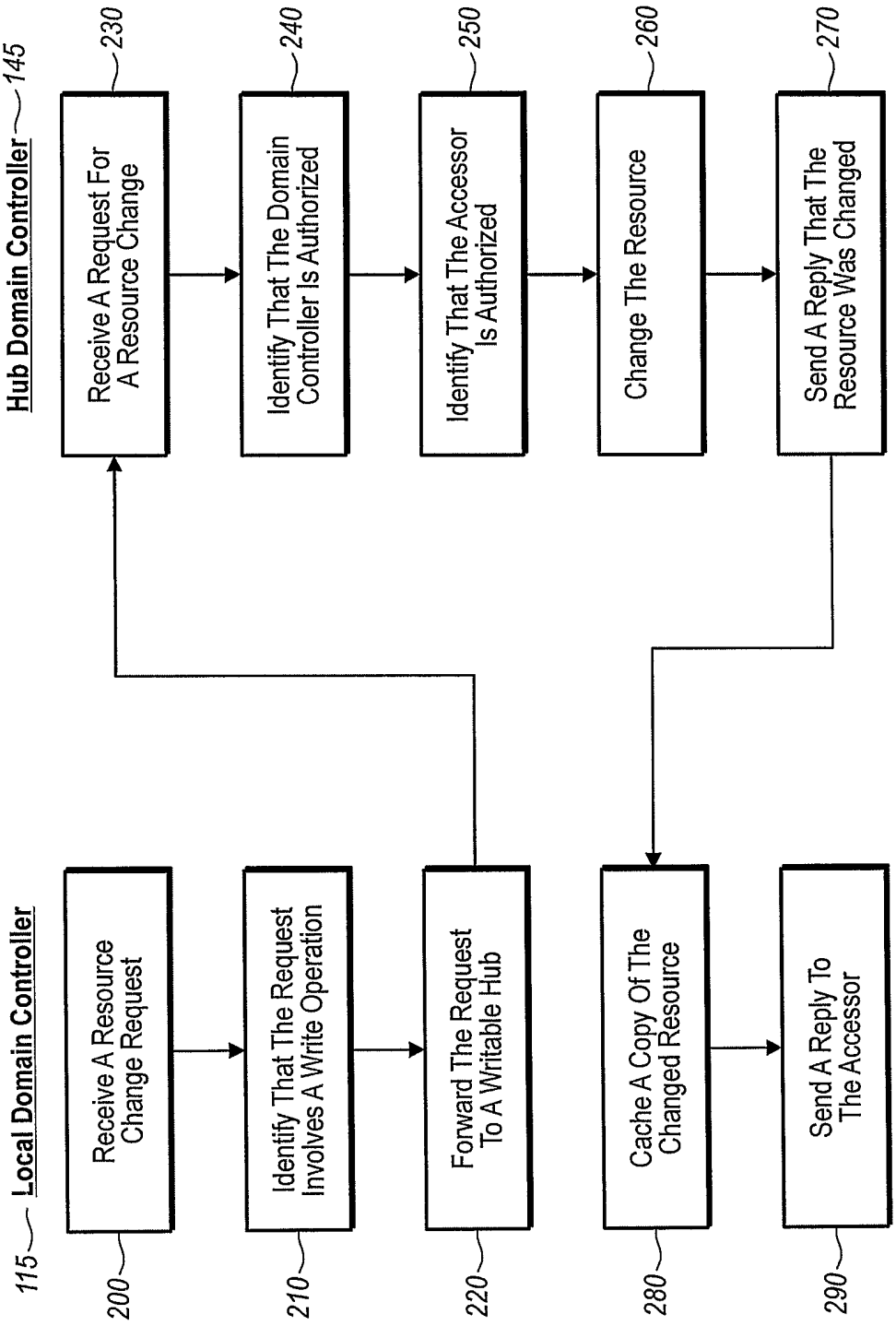

Hub Domain Controller
145
Receive A Request For A Resource Change
230
Identify That The Domain Controller Is Authorized
240
Identify That The Accessor Is Authorized
250
Change The Resource
260
Send A Reply That The Resource Was Changed
270
Local Domain Controller
115
Receive A Resource Change Request
200
Identify That The Request Involves A Write Operation
210
Forward The Request To A Writable Hub
220
Cache A Copy Of The Changed Resource
280
Send A Reply To The Accessor
290

Fig. 2