US 20110321147A1

(19) United States

(12) Patent Application Publication

(10) Pub. No.: US 2011/0321147 A1

(43) Pub. Date: Dec. 29, 2011

CHAKRA et al.

(54) **DYNAMIC, TEMPORARY DATA ACCESS TOKEN**

(75) Inventors: **Al CHAKRA**, Apex, NC (US); **Yongcheng LI**, Cary, NC (US); **Yuping C. WU**, Cary, NC (US)

(73) Assignee: **INTERNATIONAL BUSINESS MACHINES CORPORATION**, Armonk, NY (US)

(21) Appl. No.: **12/825,291**

(22) Filed: **Jun. 28, 2010**

Publication Classification

(51) **Int. Cl.**

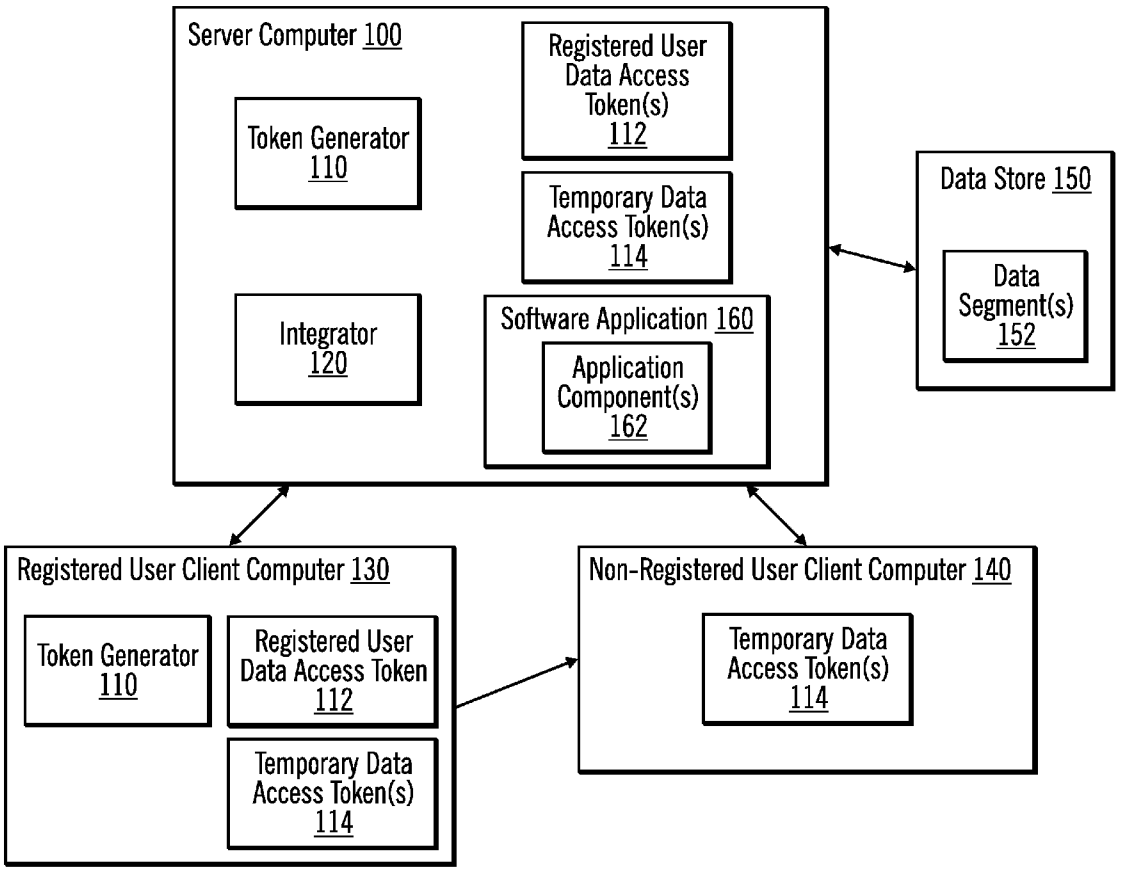
H04L 9/32 (2006.01)

G06F 21/00 (2006.01)

(52) **U.S. Cl.** **726/9**

(57) **ABSTRACT**

Provided are techniques for generating a temporary data access token for a subset of data for a specific period of time for a non-registered user who did not register with a computer providing access to the subset of the data. In response to the non-registered user attempting to access the subset of data with the temporary data access token, it is determined whether the temporary data access token is valid for the subset of data based on the specified period of time. In response to the temporary data access token being valid, the subset of data is provided to the non-registered user. In response to the temporary data access token not being valid, access is denied to the subset of data by the non-registered user.



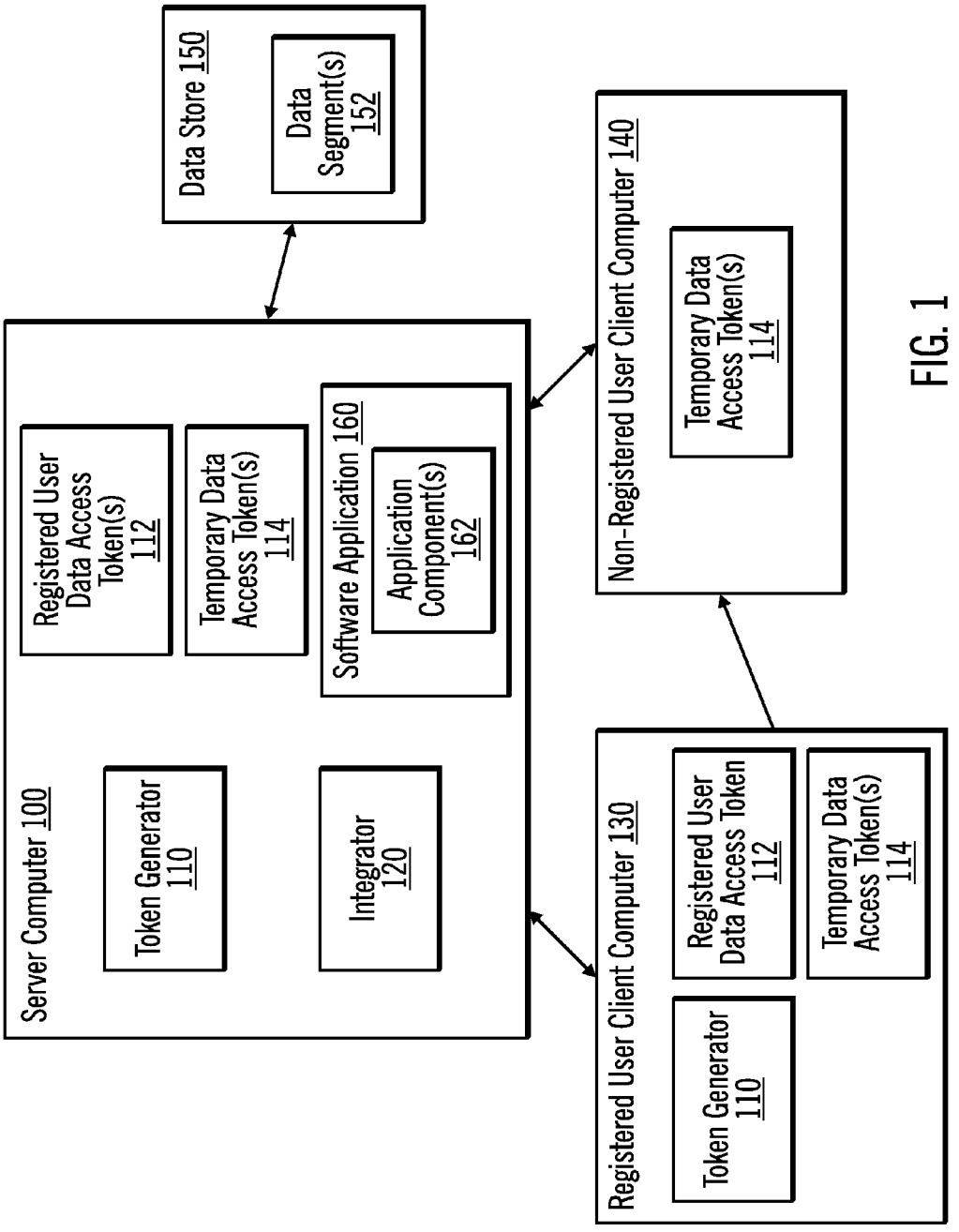

Data Store 150
Data Segment(s) 152
Server Computer 100
Registered User Data Access Token(s) 112
Temporary Data Access Token(s) 114
Software Application 160
Application Component(s) 162
Token Generator 110
Integrator 120
Non-Registered User Client Computer 140
Temporary Data Access Token(s) 114
Registered User Client Computer 130
Registered User Data Access Token 112
Temporary Data Access Token(s) 114
Token Generator 110

FIG. 1

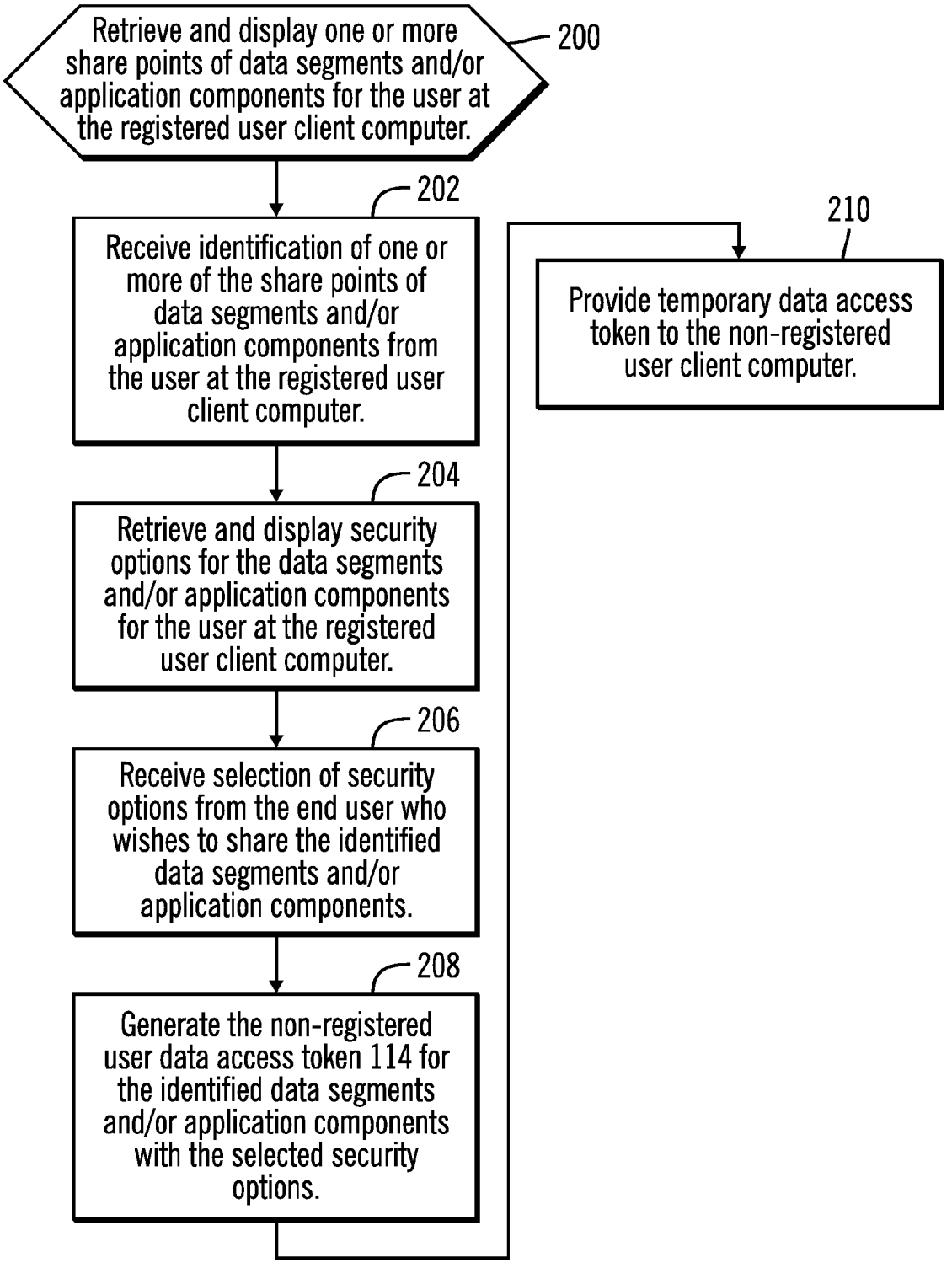

Retrieve and display one or more share points of data segments and/or application components for the user at the registered user client computer.
200
202
Receive identification of one or more of the share points of data segments and/or application components from the user at the registered user client computer.
204
Retrieve and display security options for the data segments and/or application components for the user at the registered user client computer.
206
Receive selection of security options from the end user who wishes to share the identified data segments and/or application components.
208
Generate the non-registered user data access token 114 for the identified data segments and/or application components with the selected security options.
210
Provide temporary data access token to the non-registered user client computer.

FIG. 2

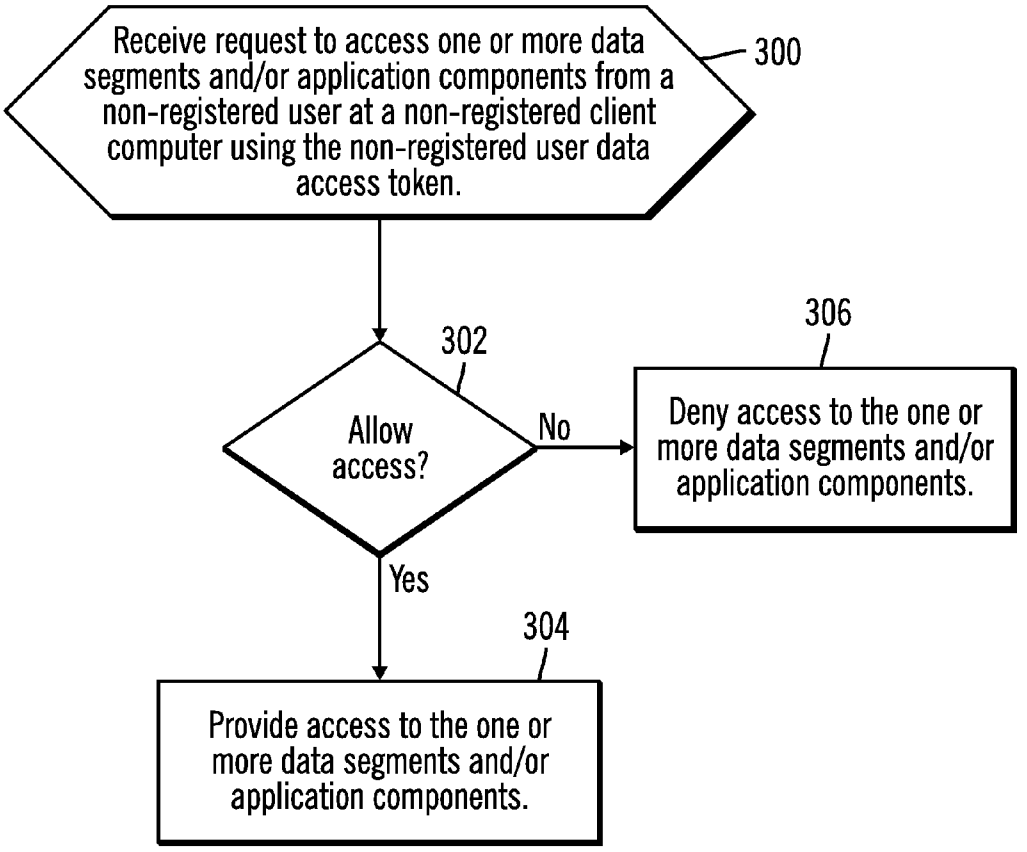

Receive request to access one or more data segments and/or application components from a non-registered user at a non-registered client computer using the non-registered user data access token.
300
302
Allow access?
No
306
Deny access to the one or more data segments and/or application components.
Yes
304
Provide access to the one or more data segments and/or application components.

FIG. 3

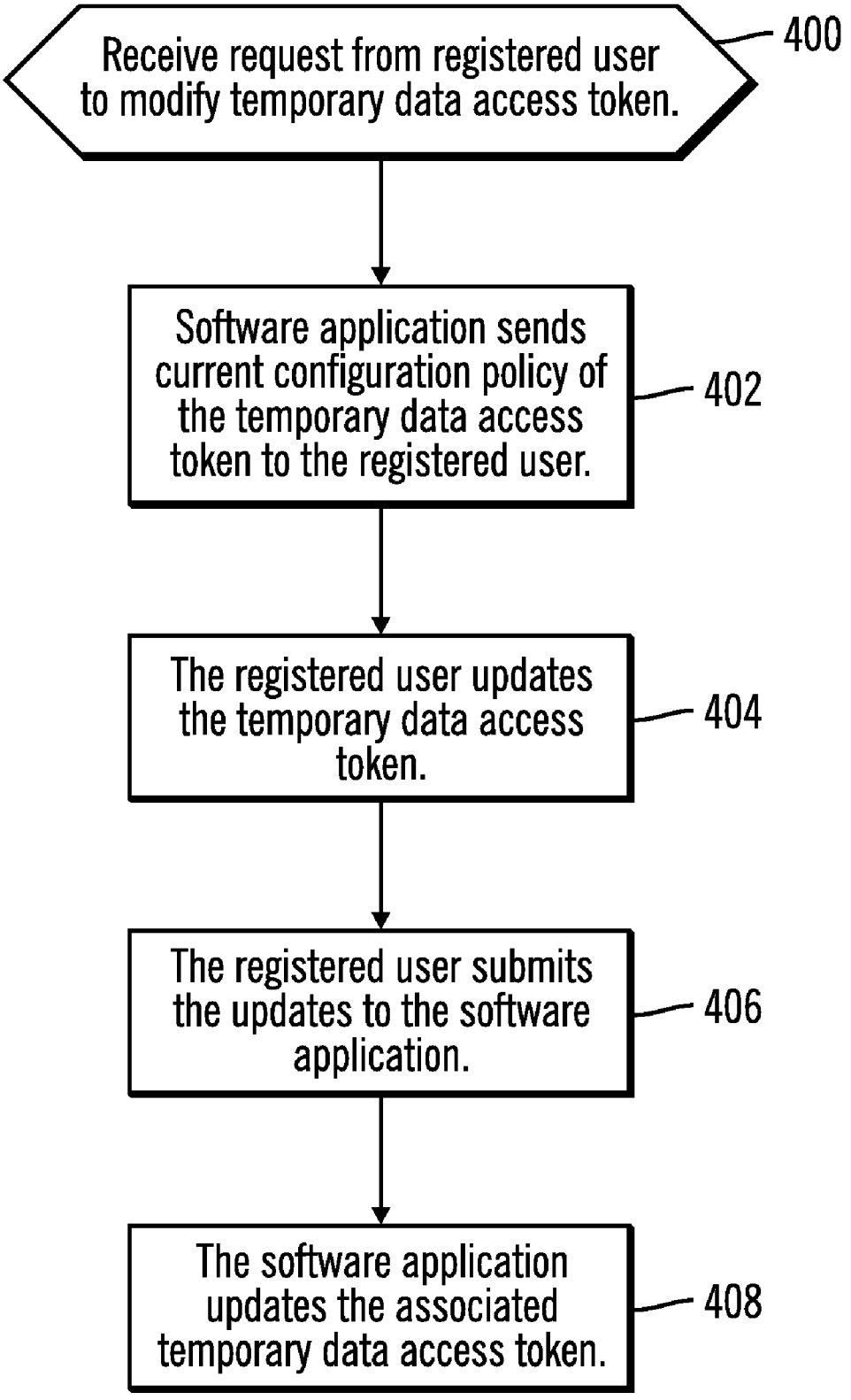

Receive request from registered user to modify temporary data access token.
400
Software application sends current configuration policy of the temporary data access token to the registered user.
402
The registered user updates the temporary data access token.
404
The registered user submits the updates to the software application.
406
The software application updates the associated temporary data access token.
408

FIG. 4

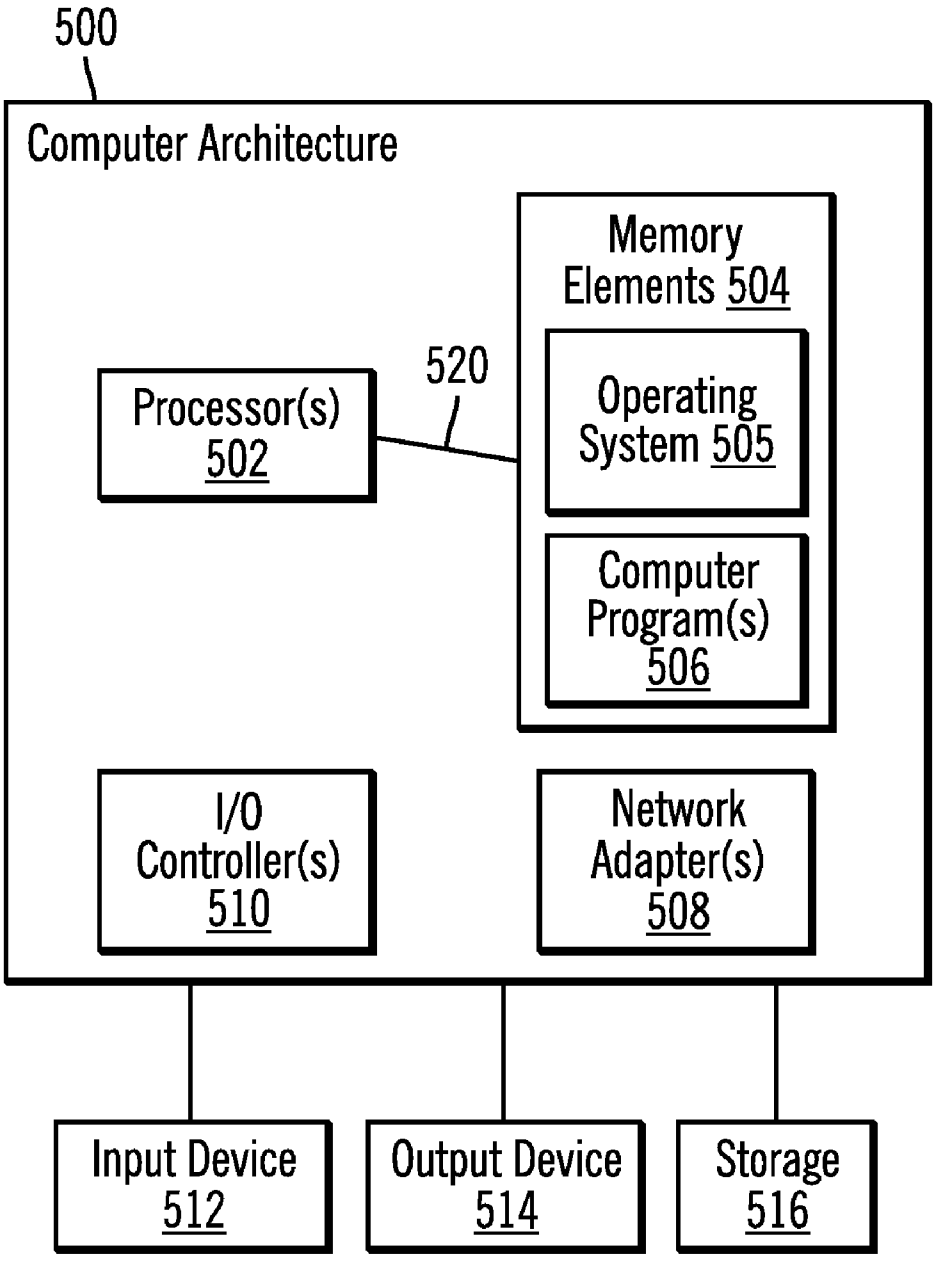

500
Computer Architecture
Memory Elements 504
520
Processor(s) 502
Operating System 505
Computer Program(s) 506
I/O Controller(s) 510
Network Adapter(s) 508
Input Device 512
Output Device 514
Storage 516

FIG. 5

DYNAMIC, TEMPORARY DATA ACCESS TOKEN

BACKGROUND

[0001] 1. Field

[0002] Embodiments of the invention relate to a dynamic, temporary data access token.

[0003] 2. Description of the Related Art

[0004] Dynamic data (i.e., information) is typically stored and managed in a centralized management system and retrieved and rendered to end users through client applications, such as browser applications. Examples of such applications are: portal, widget-based web application, etc. Access to the data requires that a user first signs in to the centralized management system by, for example, providing a user identifier (ID) and password. Thus, the requirement of signing in makes it difficult for a first user to share portions of the data in a restricted application with a second user who does not have a user identifier and password for signing in to the centralized management system.

[0005] For example, it would be useful to share corporate resources, such as conference rooms, a knowledge base, and personal information, to facilitate interaction with customers. As another example, it would be useful to print out paper documents with select information to share with others. For example, a problem occurs when a user wants to print out a paper document that contains both confidential and non-confidential data and wants to print the non-confidential part of the document to share with others).

[0006] Some conventional systems build customized applications for outside users, but these are expensive and not flexible.

[0007] Thus, there is a need for a dynamic, temporary data access token.

BRIEF SUMMARY

[0008] Provided are a method, computer program product, and system for generating a temporary data access token for a subset of data for a specific period of time for a non-registered user who did not register with a computer providing access to the subset of the data. In response to the non-registered user attempting to access the subset of data with the temporary data access token, it is determined whether the temporary data access token is valid for the subset of data based on the specified period of time. In response to the temporary data access token being valid, the subset of data is provided to the non-registered user. In response to the temporary data access token not being valid, access is denied to the subset of data by the non-registered user.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

[0009] Referring now to the drawings in which like reference numbers represent corresponding parts throughout:

[0010] FIG. **1** illustrates a computing architecture in accordance with certain embodiments.

[0011] FIG. **2** illustrates logic for sharing data with a non-registered user in accordance with certain embodiments.

[0012] FIG. **3** illustrates logic for determining access in accordance with certain embodiments.

[0013] FIG. **4** illustrates logic for dynamically updating a temporary data access token in accordance with certain embodiments.

[0014] FIG. **5** illustrates a computer architecture that may be used in accordance with certain embodiments.

DETAILED DESCRIPTION

[0015] In the following description, reference is made to the accompanying drawings which form a part hereof and which illustrate several embodiments of the invention. It is understood that other embodiments may be utilized and structural and operational changes may be made without departing from the scope of the invention.

[0016] FIG. **1** illustrates a computing architecture in accordance with certain embodiments. A server computer **100** is coupled to a registered user client computer **130** and a non-registered user client computer **140**. The registered user client computer **130** is used by a user who registers with the server computer **100** by, for example, providing the server computer **100** with a user ID and password. The registered user client computer **130** is also coupled to the non-registered user client computer **140**. The non-registered user client computer **140** is used by a user who does not register with the server computer **100** (e.g., someone who would not normally be given access to data at the server computer **100**).

[0017] The server computer **100** includes a token generator **110** and an integrator **120**. The token generator **110** generates one or more registered user data access tokens **112** and generates one or more temporary data access tokens **114**. The temporary data access tokens **114** may also be referred to as non-registered user data access tokens, limited tokens, temporal tokens, or delegated tokens. An integrator **120** may be described as a software component that integrates the output from various other software components.

[0018] The server computer **100** stores at least one software application **160**, which includes one or more application components **162**. An application component **162** may be described as part of the software application **160** and serves a specific function, such as weather report application component. Other examples of application components **162** are portions of the application that produce data for the web page (e.g., an applet or a servlet). Although one software application **162** is illustrated, embodiments may include any number of software applications.

[0019] In certain embodiments, the registered user client computer **130** also includes a token generator **110**. In certain embodiments, the token generator **110** at the registered user client computer **110** generates one or more one or more temporary data access tokens **114**. The registered user client computer **130** stores a registered user data access token **112** that is provided by the server computer **100**. The registered user client computer **130** uses the registered user data access token **112** to access data at the server computer **100**.

[0020] The non-registered user client computer stores a non-registered use data access token **114** and uses the non-registered use data access token to access the data at the server computer **100**.

[0021] The server computer **100** is coupled to a data store **150**. The data store **150** stores one or more data segments **152**. Data segments **152** may be described as a database server that can be used to store and retrieve data. An example of a data segment **152** is a dynamic content of a page of a web site.

[0022] Embodiments provide a new security model in which data segments **152** and/or application components **162** are associated with data access tokens (also referred to as security tokens) to provide more granular access control. The token generator **110** allows a user who has registered with the

server computer **100** to specify a temporary data access token **114**. In certain embodiments, the temporary data access token **114** is shared by one or more non-registered users for a specific period of time (i.e., for a time period in which a web conference is occurring), for a subset of the data available at the server computer **100**, for a number of times the temporary data access token **114** can be used, and/or other factors.

[0023] Moreover, the temporary data access token **114** has attributes, and these attributes are dynamic. The attributes are dynamic in that the attributes can be changed (i.e., updated or modified) after the temporary data access token **114** has been issued. For example, the expiration date can be modified. The subset of the data that can be viewed by the temporary data access token **114** can be modified.

[0024] Initially, the user at the registered user client computer **130** registers with the server computer **100** and obtains a registered user data access token **112**. Then, the registered user at the registered user client computer **130** uses the registered user data access token **112** to access data at the server computer **100**.

[0025] FIG. **2** illustrates logic for sharing data with a non-registered user in accordance with certain embodiments. Control begins in block **200** with the token generator **110** at the server computer **100** retrieving and displaying one or more share points of data segments **152** and/or application components **162** (e.g., via a Graphical User Interface (GUI)) for the user at the registered user client computer **130**. In certain embodiments, the token generator **110** displays the one or more share points in response to a request from the registered user at the registered user client computer **130**.

[0026] In block **202**, the token generator **110** at the server computer **100** receives identification of one or more of the share points of data segments **152** and/or application components **162** from the user at the registered user client computer **130**. Then, a user at the registered user client computer **130** identifies the one or more share points of data segments **152** and/or application components **162** via the GUI.

[0027] Thus, in embodiments, the token generator **110** at the server computer **100** provides application logic to enforce the security configuration.

[0028] In block **204**, the token generator retrieves and displays security options for the identified data segments **152** and/or application components **162** for the user at the registered user client computer **130**. In certain embodiments, there are different security options associated with different data segments **152** and/or application components **162**. Examples of security options are the capability of view and edit the content).

[0029] In block **206**, the token generator **110** at the server computer **100** receives selection of security options from the end user who wishes to share the identified data segments **152** and/or application components **162**.

[0030] In block **208**, the token generator **110** generates the temporary data access token **114** for the identified data segments **152** and/or application components **162** with the selected security options. In certain embodiments, the token generator **110** at the server computer **100** generates the temporary data access token **114**. In certain embodiments, the token generator **110** at the registered user client computer **130** generates the temporary data access token **114**.

[0031] In block **210**, the temporary data access token **114** is provided to the non-registered user client computer **140**. In certain embodiments, the token generator **110** at the server computer provides the temporary data access token **114** to the non-registered user client computer **140**. In certain embodiments, the token generator at the registered client computer **130** provides the temporary data access token **114** to the non-registered user client computer **140**.

[0032] Then, a non-registered user at the non-registered client computer **140** accesses the identified data segments **152** and/or application components **162** using the temporary data access token **114**. Thus, a user can access (e.g., view or edit) the identified data segments **152** and/or application components **162** without signing in to the server computer **100**.

[0033] In certain embodiments, a password is created for the temporary data access token **114**. In such a case, the password and the temporary data access token **114** are provided to the non-registered user at the non-registered user client computer **140**.

[0034] In embodiments, there are various levels of access control. At the highest application component level, each application component **162** registers to the integrator **120**. When the integrator **120** receives a request to access an application component **162**, the integrator **120** uses the temporary data access token **114** to determine whether to allow or deny access to the application component **162**. When the application component **162** is allowed for access, the integrator **120** provides the application component **162** to the non-registered user client computer **140**, otherwise, the application component **162** is bypassed. Thus, if multiple application components **162** are requested, and a subset (one or more) of the application components **162** are not allowed for access, these application components **162** are not provided to the non-registered user client computer **140**, while the allowed application components **162** are provided.

[0035] At the data record level are related to the data segments **152**. For example, the data records are like personal profile fields, such as name, contact, projects, etc. Each data record registers itself with the integrator **120**, and the temporary data access token **114** is used to decide when a subset of the records are to be provided to the non-registered user client computer **140**. When the integrator **120** receives a request to access a data record, the integrator **120** uses the temporary data access token **114** to determine whether to allow or deny access to the data record. When the data record is allowed for access, the integrator **120** provides the data record to the non-registered user client computer **140**, otherwise, the data record is bypassed. Thus, if multiple data records are requested, and a subset (one or more) of the data records are not allowed for access, these data records are not provided to the non-registered user client computer **140**, while the allowed data records are provided.

[0036] FIG. **3** illustrates logic for determining access in accordance with certain embodiments. Control begins at block **300** with the integrator **120** receiving a request to access one or more data segments **152** and/or application components **162** from a non-registered user at a non-registered client computer **140** using the temporary data access token **114**. In block **302**, the integrator **120** determines whether to allow access. The determination is based on the data segments **152** and/or application components **162** and the security options associated with the temporary data access token **114**. If the access is allowed, processing continues to block **304**, otherwise, processing continues to block **306**. In block **304**, the integrator **120** provides access to the one or more data segments **152** and/or application components **162**. In block **306**, the integrator **120** denies access to the one or more data segments **152** and/or application components **162**.

[0037] In embodiments, there are various configuration types. As a first example, the software application **160** running at the server computer **100** displays the options of granular access, and the user then chooses a subset. As a second example, the software application **160** displays the options of access policies (e.g., insider, partner, and customer), and the user at the registered user client compute **130** then selects a policy. As a further example, the configuration type may be a combination of the first and second examples.

[0038] FIG. **4** illustrates logic for dynamically updating the temporary data access token **114** in accordance with certain embodiments. The temporary data access token **114** is associated with a registered user **112**. Control begins in block **400** with the software application **160** receiving a request from the registered user (at the registered use client computer **130**) to modify the temporary data access token **112**. In certain embodiments, the software application **160** is a server application. In block **402**, the software application **160** sends the current configuration policy of the temporary data access token **114** to the registered user. In block **404**, the registered user updates the temporary data access token **114** by updating the configuration policy. In block **406**, the registered user submits the updates to the software application **160** by sending the updated configuration policy. In block **408**, the software application **160** updates the associated temporary data access token **114** by saving the updated configuration policy. For example, the attributes (e.g., the period of time) of the temporary data access token **114** may be updated. Thus, in certain embodiments, the temporary data access token **114** is dynamically updated (without updating the associated registered user data access token **112**). The updates are saved at the server computer **100**. Then, when the non-registered user at the non-registered user client computer **140** attempts to access one or more data segments **152** and/or application components **162**, the integrator **130** uses the updated information to determine whether to allow access.

[0039] Thus, embodiments may be used in business-to-business (B2B) applications in which content or work is shared between collaborating businesses. One example might be in sharing specific aspects of internal content repositories with potential suppliers that are submitting bids to provide products/services.

[0040] In certain embodiments, non-registered users receive Uniform Resource Locators (URLs) to access data segments **152** and/or application components **162** with an embedded token that grants access, and these non-registered users are not required to establish accounts, logins, and so on, ahead of time.

Additional Embodiment Details

[0041] As will be appreciated by one skilled in the art, aspects of the present invention may be embodied as a system, method or computer program product. Accordingly, aspects of the present invention may take the form of an entirely hardware embodiment, an entirely software embodiment (including firmware, resident software, micro-code, etc.) or an embodiment combining software and hardware aspects that may all generally be referred to herein as a "circuit," "module" or "system." Furthermore, aspects of the present invention may take the form of a computer program product embodied in one or more computer readable medium(s) having computer readable program code embodied thereon.

[0042] Any combination of one or more computer readable medium(s) may be utilized. The computer readable medium may be a computer readable signal medium or a computer readable storage medium. A computer readable storage medium may be, for example, but not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, or device, or any suitable combination of the foregoing. More specific examples (a non-exhaustive list) of the computer readable storage medium would include the following: an electrical connection having one or more wires, a portable computer diskette, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM or Flash memory), an optical fiber, a portable compact disc read-only memory (CD-ROM), an optical storage device, a magnetic storage device, solid state memory, magnetic tape or any suitable combination of the foregoing. In the context of this document, a computer readable storage medium may be any tangible medium that can contain, or store a program for use by or in connection with an instruction execution system, apparatus, or device.

[0043] A computer readable signal medium may include a propagated data signal with computer readable program code embodied therein, for example, in baseband or as part of a carrier wave. Such a propagated signal may take any of a variety of forms, including, but not limited to, electro-magnetic, optical, or any suitable combination thereof. A computer readable signal medium may be any computer readable medium that is not a computer readable storage medium and that can communicate, propagate, or transport a program for use by or in connection with an instruction execution system, apparatus, or device.

[0044] Program code embodied on a computer readable medium may be transmitted using any appropriate medium, including but not limited to wireless, wireline, optical fiber cable, RF, etc., or any suitable combination of the foregoing.

[0045] Computer program code for carrying out operations for aspects of the present invention may be written in any combination of one or more programming languages, including an object oriented programming language such as Java, Smalltalk, C++ or the like and conventional procedural programming languages, such as the "C" programming language or similar programming languages. The program code may execute entirely on the user's computer, partly on the user's computer, as a stand-alone software package, partly on the user's computer and partly on a remote computer or entirely on the remote computer or server. In the latter scenario, the remote computer may be connected to the user's computer through any type of network, including a local area network (LAN) or a wide area network (WAN), or the connection may be made to an external computer (for example, through the Internet using an Internet Service Provider).

[0046] Aspects of the present invention are described below with reference to flowchart illustrations and/or block diagrams of methods, apparatus (systems) and computer program products according to embodiments of the invention. It will be understood that each block of the flowchart illustrations and/or block diagrams, and combinations of blocks in the flowchart illustrations and/or block diagrams, can be implemented by computer program instructions. These computer program instructions may be provided to a processor of a general purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions, which execute via the processor of the computer or other programmable data pro-

cessing apparatus, create means for implementing the functions/acts specified in the flowchart and/or block diagram block or blocks.

[0047] These computer program instructions may also be stored in a computer readable medium that can direct a computer, other programmable data processing apparatus, or other devices to function in a particular manner, such that the instructions stored in the computer readable medium produce an article of manufacture including instructions which implement the function/act specified in the flowchart and/or block diagram block or blocks.

[0048] The computer program instructions may also be loaded onto a computer, other programmable data processing apparatus, or other devices to cause a series of operational steps to be performed on the computer, other programmable apparatus or other devices to produce a computer implemented process such that the instructions which execute on the computer or other programmable apparatus provide processes for implementing the functions/acts specified in the flowchart and/or block diagram block or blocks.

[0049] The code implementing the described operations may further be implemented in hardware logic or circuitry (e.g., an integrated circuit chip, Programmable Gate Array (PGA), Application Specific Integrated Circuit (ASIC), etc.

[0050] FIG. **5** illustrates a computer architecture **500** that may be used in accordance with certain embodiments. Server computer **100**, registered user client computer **130**, and/or non-registered user client computer **140** may implement computer architecture **500**. The computer architecture **500** is suitable for storing and/or executing program code and includes at least one processor **502** coupled directly or indirectly to memory elements **504** through a system bus **520**. The memory elements **504** may include local memory employed during actual execution of the program code, bulk storage, and cache memories which provide temporary storage of at least some program code in order to reduce the number of times code must be retrieved from bulk storage during execution. The memory elements **504** include an operating system **505** and one or more computer programs **506**.

[0051] Input/Output (I/O) devices **512**, **514** (including but not limited to keyboards, displays, pointing devices, etc.) may be coupled to the system either directly or through intervening I/O controllers **510**.

[0052] Network adapters **508** may also be coupled to the system to enable the data processing system to become coupled to other data processing systems or remote printers or storage devices through intervening private or public networks. Modems, cable modem and Ethernet cards are just a few of the currently available types of network adapters **508**.

[0053] The computer architecture **500** may be coupled to storage **516** (e.g., a non-volatile storage area, such as magnetic disk drives, optical disk drives, a tape drive, etc.). The storage **516** may comprise an internal storage device or an attached or network accessible storage. Computer programs **506** in storage **516** may be loaded into the memory elements **504** and executed by a processor **502** in a manner known in the art.

[0054] The computer architecture **500** may include fewer components than illustrated, additional components not illustrated herein, or some combination of the components illustrated and additional components. The computer architecture **500** may comprise any computing device known in the art, such as a mainframe, server, personal computer, workstation, laptop, handheld computer, telephony device, network appliance, virtualization device, storage controller, etc.

[0055] The flowchart and block diagrams in the figures illustrate the architecture, functionality, and operation of possible implementations of systems, methods and computer program products according to various embodiments of the present invention. In this regard, each block in the flowchart or block diagrams may represent a module, segment, or portion of code, which comprises one or more executable instructions for implementing the specified logical function (s). It should also be noted that, in some alternative implementations, the functions noted in the block may occur out of the order noted in the figures. For example, two blocks shown in succession may, in fact, be executed substantially concurrently, or the blocks may sometimes be executed in the reverse order, depending upon the functionality involved. It will also be noted that each block of the block diagrams and/or flowchart illustration, and combinations of blocks in the block diagrams and/or flowchart illustration, can be implemented by special purpose hardware-based systems that perform the specified functions or acts, or combinations of special purpose hardware and computer instructions.

[0056] The terminology used herein is for the purpose of describing particular embodiments only and is not intended to be limiting of the invention. As used herein, the singular forms "a", "an" and "the" are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will be further understood that the terms "comprises" and/or "comprising," when used in this specification, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof.

[0057] The corresponding structures, materials, acts, and equivalents of all means or step plus function elements in the claims below are intended to include any structure, material, or act for performing the function in combination with other claimed elements as specifically claimed. The description of embodiments of the present invention has been presented for purposes of illustration and description, but is not intended to be exhaustive or limited to the invention in the form disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art without departing from the scope and spirit of the invention. The embodiments were chosen and described in order to best explain the principles of the invention and the practical application, and to enable others of ordinary skill in the art to understand the invention for various embodiments with various modifications as are suited to the particular use contemplated. The foregoing description of embodiments of the invention has been presented for the purposes of illustration and description. It is not intended to be exhaustive or to limit the embodiments to the precise form disclosed. Many modifications and variations are possible in light of the above teaching. It is intended that the scope of the embodiments be limited not by this detailed description, but rather by the claims appended hereto. The above specification, examples and data provide a complete description of the manufacture and use of the composition of the embodiments. Since many embodiments may be made without departing from the spirit and scope of the embodiments, the embodiments reside in the claims hereinafter appended or any subsequently-filed claims, and their equivalents.

1. A computer-implemented method, comprising:
generating a temporary data access token for a subset of data for a specific period of time for a non-registered user who did not register with a computer providing access to the subset of the data; and
in response to the non-registered user attempting to access the subset of data with the temporary data access token,
determining whether the temporary data access token is valid for the subset of data based on the specified period of time;
in response to the temporary data access token being valid,
providing the subset of data to the non-registered user; and
in response to the temporary data access token not being valid, denying access to the subset of data by the non-registered user.

2. The method of claim 1, wherein the subset of data includes one of a data segment and an application component.

3. The method of claim 1, further comprising:
dynamically updating attributes of the temporary data access token.

4. The method of claim 1, wherein the non-registered user is a user who does not provide sign in information to the computer when trying to access the subset of data.

5. The method of claim 1, further comprising:
generating a registered user data access token for the registered user; and
associating the temporary data access token with the registered user.

6. The method of claim 1, further comprising:
receiving selection of security options for the temporary data access token.

7. A system, comprising:
hardware logic capable of performing operations, the operations comprising:
generating a temporary data access token for a subset of data for a specific period of time for a user who did not register with a computer providing access to the subset of the data; and
in response to a non-registered user attempting to access the subset of data with the temporary data access token,
determining whether the temporary data access token is valid for the subset of data based on the specified period of time;
in response to the temporary data access token being valid,
providing the subset of data to the non-registered user; and
in response to the temporary data access token not being valid, denying access to the subset of data by the non-registered user.

8. The system of claim 7, wherein the subset of data includes one of a data segment and an application component.

9. The system of claim 7, further comprising:
dynamically updating attributes of the temporary data access token.

10. The system of claim 7, wherein the non-registered user is a user who does not provide sign in information to the computer when trying to access the subset of data.

11. The system of claim 7, further comprising:
generating a registered user data access token for the registered user; and
associating the temporary data access token with the registered user.

12. The system of claim 7, further comprising:
receiving selection of security options for the temporary data access token.

13. A computer program product comprising a computer readable storage medium including a computer readable program, wherein the computer readable program when executed by a processor on a computer causes the computer to:
generate a temporary data access token for a subset of data for a specific period of time for a user who did not register with a computer providing access to the subset of the data; and
in response to a non-registered user attempting to access the subset of data with the temporary data access token,
determine whether the temporary data access token is valid for the subset of data based on the specified period of time;
in response to the temporary data access token being valid,
provide the subset of data to the non-registered user; and
in response to the temporary data access token not being valid, deny access to the subset of data by the non-registered user.

14. The computer program product of claim 13, wherein the subset of data includes one of a data segment and an application component.

15. The computer program product of claim 13, wherein the computer readable program when executed by the processor on by computer causes the computer to:
dynamically update attributes of the temporary data access token.

16. The computer program product of claim 13, wherein the non-registered user is a user who does not provide sign in information to the computer when trying to access the subset of data.

17. The computer program product of claim 13, wherein the computer readable program when executed by the processor on by computer causes the computer to:
generate a registered user data access token for the registered user; and
associate the temporary data access token with the registered user.

18. The computer program product of claim 13, wherein the computer readable program when executed by the processor on by computer causes the computer to:
receive selection of security options for the temporary data access token.

* * * * *