



(86) Date de dépôt PCT/PCT Filing Date: 2001/03/07

(87) Date publication PCT/PCT Publication Date: 2001/09/27

(85) Entrée phase nationale/National Entry: 2002/08/15

(86) N° demande PCT/PCT Application No.: US 2001/007282

(87) N° publication PCT/PCT Publication No.: 2001/072009

(30) Priorité/Priority: 2000/03/17 (09/528,189) US

(51) Cl.Int.⁷/Int.Cl.⁷ H04L 29/06

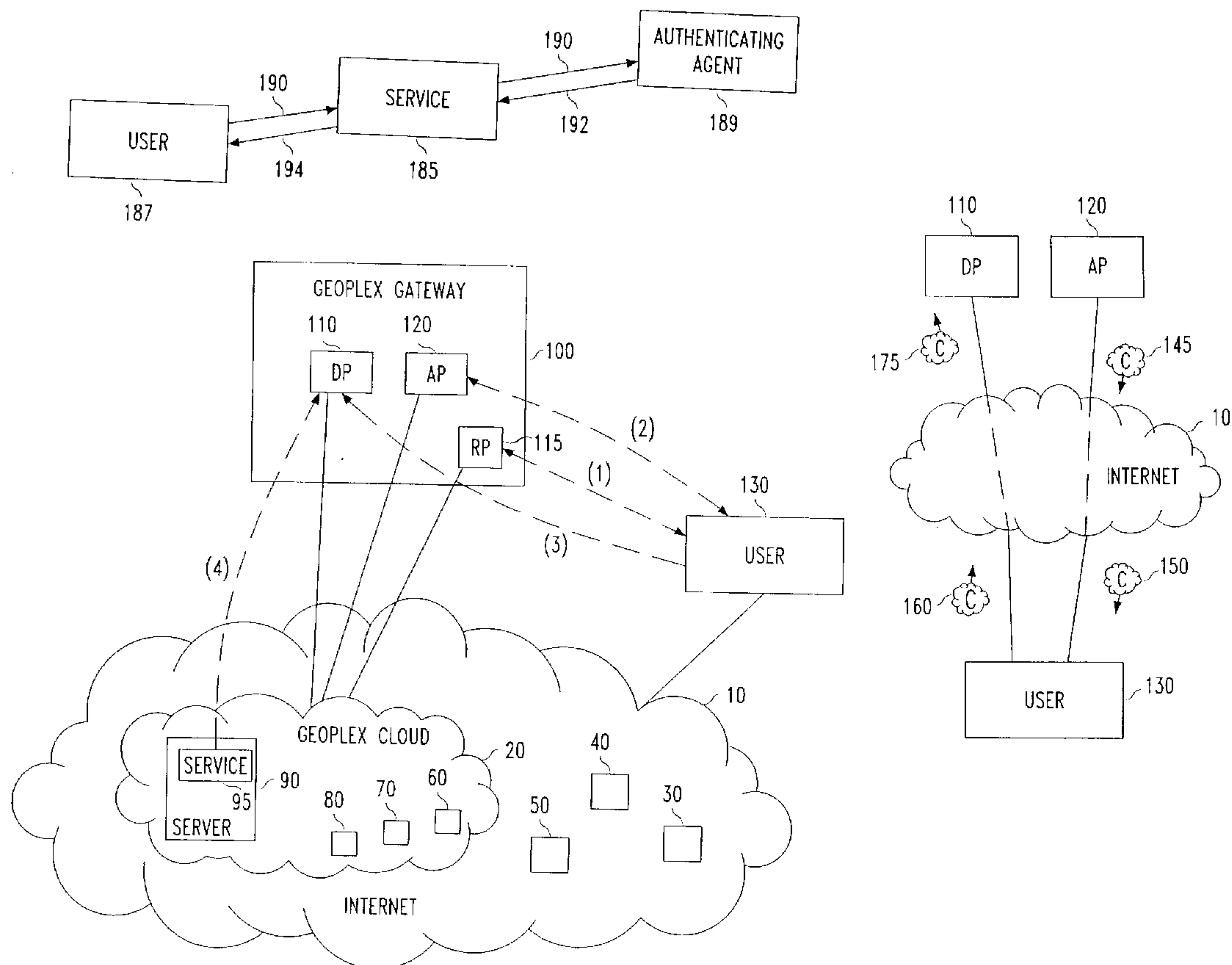
(71) Demandeur/Applicant:
AT&T CORP., US

(72) Inventeurs/Inventors:
BALABINE, IGOR, US;
DUTTA, PARTHA P., US;
KUMAR, MAHESH M., US;
TSELOVALNIKOV, ALEX, US

(74) Agent: KIRBY EADES GALE BAKER

(54) Titre : **MECANISME D'AUTHENTIFICATION BASE SUR LE WEB ET POSSEDANT UNE PROCEDURE
D'OUVERTURE UNIQUE**

(54) Title: **WEB-BASED SINGLE-SIGN-ON AUTHENTICATION MECHANISM**



(57) **Abrégé/Abstract:**

A method and apparatus are disclosed for a single sign-on method and system for accessing a plurality of services distributed over a network in which authentication-related functionality is separated from the services, and in which authentication need not be renegotiated for access to a new service from the plurality of services during a session. Additional benefits accruing from



(57) **Abrégé(suite)/Abstract(continued):**

embodiments of the invention include notification of the plurality of services when a user has terminated a session, and the use of secure, short-lived authentication tokens to verify a user's identity for subsequent access to the plurality of services. The steps in a method embodiment comprise receiving a request from a user for authorization to access a service; transmitting a token corresponding to the service to the user; receiving the token corresponding to the service from the user; determining whether the user is authorized to receive the service based on the token; and connecting the user to the service, if the user is authorized to use the service.

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

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
27 September 2001 (27.09.2001)

PCT

(10) International Publication Number
WO 01/72009 A3

(51) International Patent Classification⁷: **H04L 29/06**

TSELOVALNIKOV, Alex; 1470 English Drive 22, San Jose, CA 95129 (US).

(21) International Application Number: **PCT/US01/07282**

(74) Agents: **CANAVAN, Robert, T.** et al.; AT & T Corp., P.O. Box 4110, Middletown, NJ 07748-4110 (US).

(22) International Filing Date: 7 March 2001 (07.03.2001)

(25) Filing Language: English

(81) Designated States (*national*): BR, CA, MX.

(26) Publication Language: English

(84) Designated States (*regional*): European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).

(30) Priority Data:
09/528,189 17 March 2000 (17.03.2000) US

Published:

— with international search report

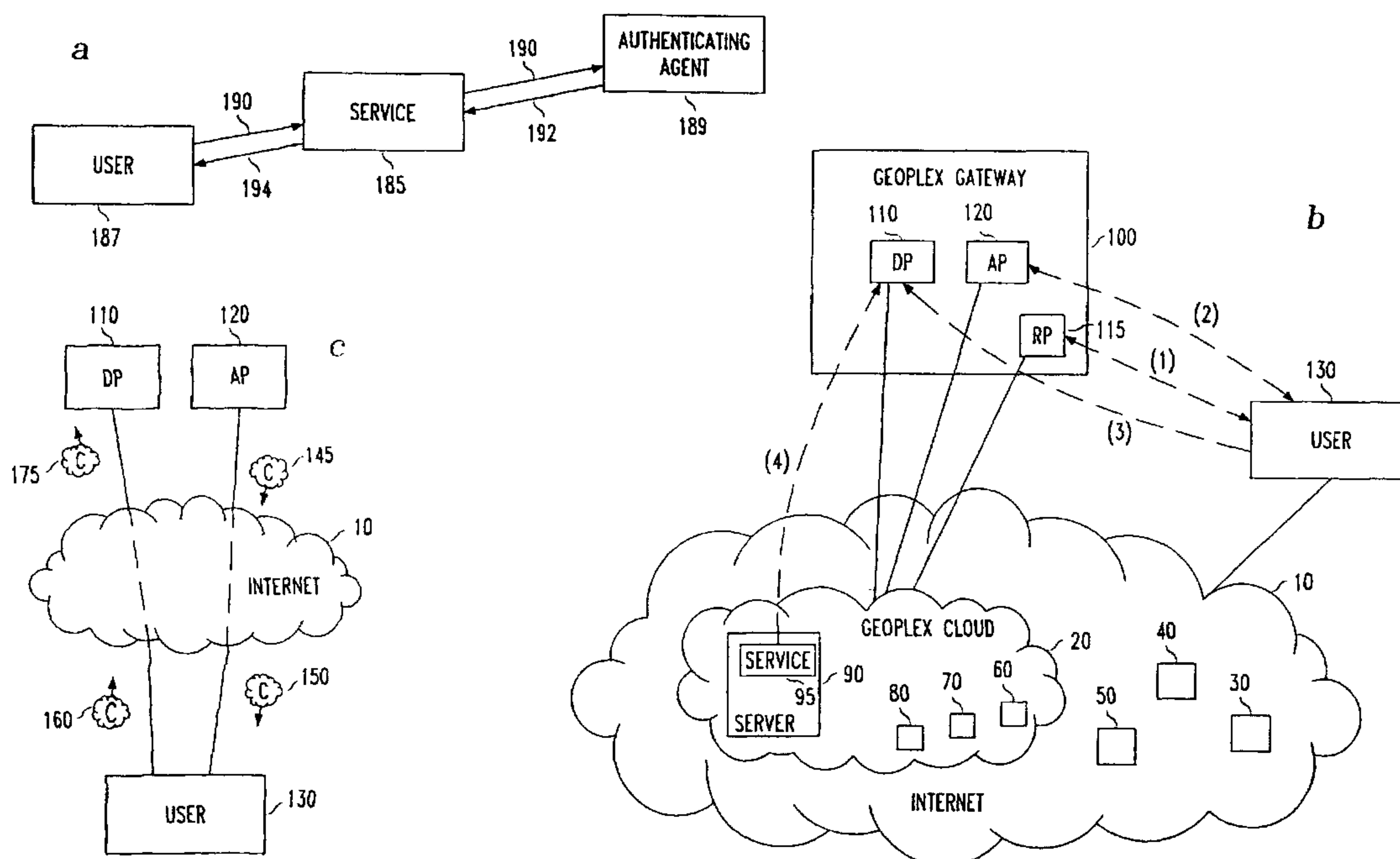
(71) Applicant: **AT & T CORP.** [—/US]; 32 Avenue of the Americas, New York, NY 10013-2412 (US).

(88) Date of publication of the international search report:
11 April 2002

(72) Inventors: **BALABINE, Igor**; 11063 Bel Aire Court, Cupertino, CA 95014 (US). **DUTTA, Partha, P.**; 1164 Malibu Drive, San Jose, CA 95129 (US). **KUMAR, Mahesh, M.**; 5009 Oak Hollow Terrace, Fremont, CA 94536 (US).

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: WEB-BASED SINGLE-SIGN-ON AUTHENTICATION MECHANISM



(57) Abstract: A method and apparatus are disclosed for a single sign-on method and system for accessing a plurality of services distributed over a network in which authentication-related functionality is separated from the services, and in which authentication need not be renegotiated for access to a new service from the plurality of services during a session. Additional benefits accruing from embodiments of the invention include notification of the plurality of services when a user has terminated a session, and the use of secure, short-lived authentication tokens to verify a user's identity for subsequent access to the plurality of services. The steps in a method embodiment comprise receiving a request from a user for authorization to access a service; transmitting a token corresponding to the service to the user; receiving the token corresponding to the service from the user; determining whether the user is authorized to receive the service based on the token; and connecting the user to the service, if the user is authorized to use the service.



WO 01/72009 A3

WEB-BASED SINGLE-SIGN-ON AUTHENTICATION MECHANISM

5 **BACKGROUND OF THE INVENTION**

The present invention relates to authentication mechanisms for computer networks, and more specifically, relates to single sign-on authentication mechanisms for enforcing authorized access to a plurality of services distributed over a network (for
10 example, the Internet).

As an increasing number of services begin to be offered over the Internet, the ability to provide an effective and secure single sign-on mechanism for access to such services has become extremely important. Services offered over the Internet are often
15 distributed on a plurality of servers that are in remote locations with respect to each other. Traditionally, each server offering a service or services would have to implement its own authentication mechanism for security purposes. Thus, a consumer of multiple services on the Internet would often have to
20 store and recall a plurality of user names, passwords, and/or digital certificates. Moreover, each server or service would have to implement its own authentication protocol. The total sum of resources, time and expenses required for implementing separate authentication procedures on each of a plurality of resources is
25 large enough to justify alternate solutions.

One known solution to this problem is to implement a single sign-on mechanism through which a user can authenticate his identity and authorization to use a plurality of services distributed over a plurality of remote servers through an
30 authentication procedure running on one or a small number of

servers. For example, in one known solution shown in Figure 1a, server 185 offering a service passes authentication request 190 received from user 187 to authenticating agent 189.

Authenticating agent 189, if it successfully authenticates the user, transmits data 192 to server 185 authorizing the provision of the service to user 187. Then, server 185 provides service 194 to user 187. In this way, authentication can be performed for a plurality of servers offering services through a single authenticating agent, and hence a single sign-on mechanism.

One deficiency of this method is that the authenticating procedures are not entirely divorced from the service server. In this method, the service server needs to have functionality allowing it to recognize a user's request for authentication and forward this request to the authenticating agent. Further, the service server has to have functionality allowing it to recognize a transmission from the authenticating server authorizing or refusing to authorize the user's request. Thus, each new service added to the plurality of services secured by the authenticating agent requires installation and maintenance of this functionality.

Second, in this method, a service server has to initiate an authentication negotiation with the authentication agent anew each time the user attempts to use a service on a new server from the plurality of servers during a session. Consequently, the fact that the user has already been successfully authenticated during a session is of no benefit at all for verifying the authorization of a user to access a new service during the session; the new service must forward a new user request for authentication to the authentication agent and wait for a response.

Thus, a single sign-on mechanism for accessing a plurality of services distributed over a network is needed in which authentication-related functionality is separated from the services,

and in which authentication need not be renegotiated for access to a new service from the plurality of services during a session.

SUMMARY OF THE INVENTION

5 The deficiency in prior known systems and methods are overcome by embodiments of the present invention, which disclose a novel, single sign-on authentication method and system for accessing a plurality of services distributed over a network. In accordance with an embodiment of the present invention, a request
10 is received from a user for authorization to access a service. In response, a token corresponding to the service is transmitted to the user. Subsequently, the token corresponding to the service is received from the user. Then, whether the user is authorized to receive the service is determined, based on the token.
15 Subsequently, the user is connected to the service, if the user is authorized to use the service.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1a is a block diagram of a system in accordance
20 with a prior, known method.

Figure 1b is a block diagram of a system in accordance with the present invention.

Figure 1c is a block diagram of a section of a system in accordance with an embodiment of the present invention.

25 Figure 2 illustrates a flow diagram of a first embodiment of the present invention

Figure 3 illustrates a flow diagram of a second embodiment of the present invention.

Figure 4 is a block diagram of a system in accordance with
30 the second embodiment.

DETAILED DESCRIPTION OF THE INVENTION

Embodiments of the present invention disclose a single sign-on method and system for accessing a plurality of services distributed over a network in which authentication-related
5 functionality is separated from the services, and in which authentication need not be renegotiated for access to a new service from the plurality of services during a session. Additional benefits accruing from embodiments of the invention include notification of the plurality of services when a user has terminated a session,
10 and the use of secure, short-lived authentication tokens to verify a user's identity for subsequent access to the plurality of services.

Figure 1b shows a system in accordance with embodiments of the present invention. Figure 1b also shows paths of data flows in the system that are depicted with dotted lines labeled with
15 numbers in braces (the dotted lines may signify network connections between the entities at the end points of the dotted lines). In Figure 1b, user 130 is connected to Internet 10. Internet 10 includes a plurality of resources in a network, such as servers 30, 40, 50 and Cloud 20. Cloud 20 comprises a collection of
20 servers, for example servers 60, 70, 80 and 90, that offer a plurality of services for authorized Cloud users. In particular, service 95, which is located on server 90, is one of the services offered on Cloud 20.

A "remote" connection is said to exist between two
25 entities, if the entities are connected through one or more network connections. If two connected entities are not connected through at least one network connection, a "local" connection may be said to exist between the two entities. An entity connected to another entity by a remote connection may be said to be "remote" from the
30 other entity.

Gateway 100 is responsible for registering users of Cloud

20 and authenticating registered users for connection to services offered on Cloud 20. In Figure 1b, Gateway 100 is shown to be connected to user 130, the Internet 10 and Cloud 20. However, Gateway 100 may be connected to a plurality of users and/or a plurality of networks. Gateway 100 includes a registration proxy 115, an authentication proxy 120 and a data proxy 110. Registration proxy 115, authentication proxy 120 and data proxy 110 are logical units that may be implemented in hardware or software. In particular, any or all of the functionality within Gateway 100, including that of authentication proxy 120, data proxy 110 and registration proxy 115, may be spread over one or more physical units. For example, data proxy 110 may comprise a plurality of logical units, each located on a server at a different node within a network, e.g., the Internet, or on the same unit.

15 A user of Cloud 20 initially registers with registration proxy 115 in order to use the web-based authentication mechanism for access to services available on Cloud 20. The user begins the registration procedure by connecting to the registration proxy. For example, the user may use a standard web browser to connect to registration proxy 115 using the Uniform Resource Locator ("URL") of Gateway 100 or registration proxy 115. The connection may be made using the Secure Sockets Layer ("SSL") protocol for ensuring data security during the registration procedure. Alternatively, this connection may be made through any secure communications protocol, e.g., the Transport Layer Security ("TLS") protocol (e.g., RFC 2246). A secure communications protocol allows communication in a way that is designed to prevent eavesdropping, tampering, or message forgery.

30 The user may register by choosing and communicating to registration proxy 115 credentials that may be used for

authentication to Cloud 20. For example, the user may choose a user name and a password, or a user name and a digital certificate, e.g., a X.509 standard certificate, as credentials for authentication purposes. The user must choose at least one credential to

5 complete the registration procedure. The user, during the registration process, may additionally be required to specify billing information, including a credit card number or a bank account number, for use in paying for Cloud services. The dotted line labeled “(1)” in Figure 1b between user 130 and registration

10 proxy 115 indicates the data flow in connection with the registration to Cloud 20. This connection may be made through Internet 10. Once the registration process has been completed, this connection may be dropped.

Once user 130 registers with Cloud 20, he may access one

15 or more services on Cloud 20. However, before user 130 can be connected to a Cloud service, he may be required to authenticate his identity to Gateway 100. A user that authenticates his identity is authorized to access one or more services on Cloud 20. User 130 authenticates his identity by connecting to authentication

20 proxy 120 and presenting his authentication credentials. User 130 may establish a connection to authentication proxy 120 by specifying the URL of Gateway 100, authentication proxy 120, or any service within the Cloud to his web browser. User 130 may then specify his user name and password, or he may specify his

25 user name and present a digital certificate, in response to a prompt from authentication proxy 120. The connection between authentication proxy 120 and user 130 may be made using the SSL protocol, or any other secure communications protocol, for data security purposes. The dotted line labeled “(2)” in Figure 1b

30 between user 130 and authentication proxy 120 indicates data flow during authentication (as well as the flow of tokens from

authentication proxy 120 to user 130, which will be discussed shortly). This connection, which may be said to establish a channel between user 130 and authentication proxy 120, may be made through Internet 10. In one embodiment, once user 130 authenticates his identity to authentication proxy 120, the original connection between authentication proxy 120 and user 130 is severed and replaced by a new, secure connection (or channel) called a network control connection. For example, after user 130 successfully authenticates his identity, authentication proxy 120 may send user 130's browser a web page with a JavaScript program that contains a unique URL, and then terminate the connection to user 130. The JavaScript program received by user 130's browser may then reconnect to authentication proxy 120 through the unique URL. This connection may in particular be a mutually authenticated SSL connection, or a connection using any other secure communications protocol.

After user 130 authenticates his identity to authentication proxy 120, he may receive services on Cloud 20 for which he is authorized without any other authentication procedures that are apparent to user 130 (i.e., transparently), as long as the network control connection between authentication proxy 120 and user 130 remains intact. After authentication, authentication proxy 120 may store user 130's identification information in an authenticated connections table ("ACT"). User 130's identification information may remain in the ACT until the connection between user 130 and authentication proxy 120 is severed. Authentication proxy 120 may share the list of active Cloud users stored in the ACT with data proxy 110.

Additionally, while the network control connection between authentication proxy 120 and user 130 remains intact, authentication proxy 120 may continually transmit a plurality of

tokens to user 130 over the connection. Each token may contain information specific to one or more services, for example, identification numbers of the services to which the token corresponds. Additionally, each token transmitted to user 130
5 may include identification information specific to user 130 and a time value indicating the expiration time of the token. Moreover, each token may be encrypted by authentication proxy 120 using a secret key that is generated by authentication proxy 120. Authentication proxy 120 may also store a unique identifier
10 corresponding to the secret key in the ACT. Authentication proxy 120 may also place the unique identifier in the token that is encrypted by the corresponding key. Further, each token may be embedded in a http-protocol cookie.

In one embodiment, a list of registered users together with
15 the services the registered users are authorized to access may be stored in a memory in Gateway 100. In that embodiment, once a user has properly authenticated his identity to access services on Cloud 20, he may be sent only tokens corresponding to the services for which he is registered.

20 Assuming that the network control connection remains intact, for each token that is about to expire, authentication proxy 120 may transmit a token that is substantially similar to the token about to expire, but having a new expiration time value, and/or encrypted with a different key. The length of the interval of time
25 in which a token is valid may be predetermined and equal to a multiple of the lifetime of the encryption key. In other embodiments, the length of time in which a token is valid may be generated randomly. A token may be invalidated by removing the unique identifier corresponding to the key used for encrypting the
30 token from the ACT. This has the effect of invalidating all tokens corresponding to that key in the system. Additionally, when user

130 terminates a work session by, for example, closing the browser, the browser may remove all cookies containing the tokens stored in user 130's browser. Furthermore, tokens may be invalidated upon the system comprising Gateway 100 determining
5 that a security event indicating a possible breach or unauthorized access has occurred. Gateway 100 may invalidate tokens, for example, by simply changing keys and distributing the new keys to data proxy 110, which effectively logs off all users. Alternatively, if keys are issued on a per user basis, Gateway 100
10 may log off one or more users individually.

In another embodiment, the interruption of the network connection between authentication proxy 120 and user 130 may cause authentication proxy 120 to remove user 130's identification information from the ACT. In this embodiment, authentication
15 proxy 120 treats any token received from a user whose identification information does not appear in the ACT as being invalid. Thus, in this embodiment, a user's tokens are invalidated upon removal of the user's identification information from the ACT.

20 After the authentication procedure, user 130 may attempt to connect to a Cloud service by specifying the URL of the service on user 130's browser. Alternatively, user 130 may be automatically redirected to the desired service. The domain name entered by user 130 in his browser may cause the browser to
25 detect that the requested service is protected by Gateway 100. The browser may then connect to data proxy 110 and transmit the cookie containing the token (or tokens) corresponding to the service requested by user 130. The connection to data proxy 110 (forming a second channel, in addition to the first channel between
30 user 130 and authentication proxy 120) may be made using the SSL protocol, or any other secure communications protocol, to

assure data security. Data proxy 110 may extract an encrypted token (or tokens) from the cookie, decrypt the token (or tokens), verify its validity and extract the identification information contained within. In particular, data proxy 110 may extract the unique key identifier corresponding to the key used to encrypt the token. Data proxy 110 may verify that connection of user 130 to the requested service is authorized by checking the ACT for a match with the extracted unique key identifier. Additionally, data proxy 110 may check that the extracted identification information matches one of the users listed in the ACT. Matches between information stored in the token and information stored in the ACT may indicate the validity of the token. The dotted line labeled “(3)” in Figure 1b between user 130 and data proxy 110 indicates the data flow for transmission of tokens from user 130 and the verification of their validity at data proxy 110. This connection may be made through Internet 10.

If the token is determined to be valid, data proxy 110 may initiate a proxy connection to the service requested by user 130 using the SSL protocol, or any other secure communications protocol, on user 130's behalf. Thus, user 130 may begin to use the requested service through its connection to data proxy 110 and through data proxy 110's connection to the service, after completion of the procedure as described above. The dotted line labeled “(4)” in Figure 1b between service 95 and data proxy 110 indicates the data flow over the connection between user 130 and data proxy 110.

Figure 1c shows the transmission of cookies among authentication proxy 120, user 130 and data proxy 110 over Internet 10, in one embodiment of the present invention. After user 130 authenticates his identity, he receives a stream of tokens through a first connection (i.e. a first channel) from authentication

proxy 120, wherein each token corresponds to one or more services on Cloud 20. For example, user 130 may receive cookies 145 and 150. When user 130 attempts to connect to a Cloud service, his browser will be redirected and connect to data proxy 110 through a second connection (i.e. a second channel). User 130's browser will then transmit the cookie or cookies corresponding to the requested service to data proxy 110 through the second connection. For example, user 130's browser may transmit cookies 160 and 175 to data proxy 110.

Thus, the disclosed system provides authentication functionality allowing authorized users to access services over a network. The system comprises a gateway that authenticates users for access to the services. Moreover, all of the authentication functionality of the system exists only on the gateway. For example, a new service added to the services for which the gateway performs authentication functionality does not require any augmentation for proper functioning of the authentication functionality of the system.

An authentication token may be passed to the user in a the "Set-Cookie" field of a Hypertext Transfer Protocol ("http") header. In one embodiment, the structure of this field is given as: Set-Cookie:

```

    <Cookie-Name>=<Cookie-Value>
    ;EXPIRES=<expiration time>
    ;DOMAIN=<destination-domain>
    ;PATH=<destination-path>
    [;SECURE]
  
```

The following fields appear within the "Set-cookie" field:

cookie-name – the name of the cookie. This

value is equal to GeoPlexIdSecure for the cookies used over the secure connections (https). For regular http connections this name is set to GeoPlexId.

5 *expiration-time* - the date string formatted
as:

Wdy, DD-Mon-YY HH:MM:SS GMT

Wdy - is the day of the week (for example, Mon or Tues); DD is a two-digit representation of the day of the month; Mon is a three-letter abbreviation for the month (for example, Jan or Feb); YY is the last two digits of the year; HH:MM:SS are hours, minutes, and seconds, respectively. The authentication proxy may set this value to be the current time plus the cookie validity period. The expiration of a cookie may be enforced by the authentication proxy. The presence of an expiration value on the user's side sets the guidelines for the use of the cookie for the user's browser.

destination-domain – the attributes of a protected domain that may be accessed using this authentication token;

20 destination-path – the path in the protected domain for which this
cookie is intended;

25 **SECURE** – this specifier denotes authentication tokens that are sent only over secure connections. The authentication proxy sets this attribute only for cookies destined to SSL protected domains (https). The presence of this value on the users's side sets the guidelines for the use of the cookie for the user's browser.

```

30  Cookie-Value – base-64 encoded authentication token:
      base-64 - encoded struct {

```

13

```

uint8 TokenMagic[4];
uint32 Version;
uint32 HashType;
opaque HashValue[20];
5  uint32 WorkKeyId;
uint32 CipherType;
uint32 ContentLength;
encrypted struct {
    opaque
10  TokenContent[AuthenticationToken.ContentLength];
    select (CipherType) {
        case block:
            uint8 padding[Padding.padding_length];
            uint8 padding_length;
15    }Padding;
    } EncryptedContent;
    } GeoAuthenticationToken;

```

TokenMagic is a four-byte value that allows the quick recognition
20 of the structure of AuthenticationToken:

```
TokenMagic = { 'G', 'E', 'O', 'X' };
```

Version – the version of the AuthenticationToken structure.

Currently the value of this field is 1.

25

HashType – the identifier of a hash algorithm used to compute the
cryptographic checksum stored in the HashValue field.

```

enum {
    HashType_md5,
30  HashType_sha1
    } HashType;

```

HashValue – is the hash value of the following data fields in the cryptographic token:

WorkKeyId, CipherType, ContentLength, TokenContent
 5 (unencrypted), Padding (unencrypted).

If the md5 hash function is used, the hash value is stored in first 16 bytes of the HashValue field. The hash value for the sha1 hash function is 20 bytes long.

10 WorkKeyId – the identifier of an encryption key used to encrypt the token.

CipherType – the identifier of a cipher used to encrypt the token.

```
enum {
    CipherType_des,
    15    CipherType_3des,
    CipherType_rc4
} CipherType;
```

Content-Length – the length of the token (plain text), in bytes.

20

Token-Type – the identifier of the token.

```
enum {
    TokenType_http
    25    } TokenType;
```

TokenContent – the encrypted token.

```
struct {
    opaque random[8];
    uint32 GeoPlexUserid;
    30    TokenType type;
    select (type) {
```

15

```
case http:
    uint32 Secure;
    uint32 DomainLength;
    uint8 Domain[TokenContent.DomainLength];
5    } TokenContent;
    } GeoTokenContent;
```

Random – eight random bytes help to prevent a known plain text attack when a stream cipher such as RC4 is used.

10

Secure – this flag is set if this authentication token should be used only over secure connections.

Domain – attributes of the domain (URI) for which this cookie
15 was issued.

GeoPlexUserid – user id of the token owner.

Padding – padding of the TokenContent field to be a multiple of
20 eight. This field is present only when a block cipher was used to encrypt the token.

In this embodiment, authentication tokens have a limited validity period, which is enforced by the authentication proxy.

25 When the token validity period is about to expire, the authentication proxy updates the tokens issued to all active users by sending a new set of http cookies.

Each updated set of tokens issued to a user is encrypted with a key different from the one used to encrypt the previous set
30 of tokens. This measure prevents replay attacks against the authentication proxy when a malicious user may send a stolen

token to gain access to a protected resource. Immediate replay attacks are made impossible since fresh tokens are delivered over a SSL protocol-protected (or other secure communications protocol-protected) network connection.

5 In order to provide the keying material for the next generation of authentication tokens the authentication proxy updates its pool of random data, called the master secret, every T minutes. When a master secret object is created, it is assigned a handle that uniquely identifies the object. The authentication
10 proxy keeps one preceding instance of the master secret so that the tokens issued under the older master secret are still valid for some time. This is done in order to avoid the race condition when new tokens are already issued but have not reached the user yet. The update period T is a configurable parameter
15 (“KeyGenerationInterval”) set by the System Administrator. Each newly generated master secret object is marshaled by the authentication proxy and is saved in the ACT. The authentication proxy stores the two most recent marshaled master secret objects in the ACT. A new marshaled master secret object replaces the
20 older, saved, marshaled master secret object in the ACT.

 In one embodiment, authentication proxy 120 periodically and automatically sends tokens to user 130 over the network control connection. In particular, in this embodiment, authentication proxy 120 automatically sends user 130 tokens
25 encrypted with new keys after an update of the master secret. Thus, in this embodiment, the gateway “pushes” updated tokens to the user. This embodiment may be implemented, for example, where user 130's browser supports multi-part data types defined in the MIME 1.0 specification.

30 Alternatively, in another embodiment, authentication proxy 120 will send user 130 a meta http tag after user 130 reconnects to

the authentication proxy through the network control connection. The meta http tag causes user 130's browser to periodically and automatically send a request signal to authentication proxy 120 through the network control connection. When the master secret is updated, authentication proxy 120 sends tokens encrypted with new keys to user 130, as long as authentication proxy 120 has received the last scheduled request signal, and/or has received at least one request signal during a predetermined amount of time. Thus, in this embodiment, the user "pulls" updated tokens from the gateway. This embodiment may be implemented, for example, where user 130's browser does not support multi-part MIME data types.

Figure 2 shows one embodiment of the present invention. At step 200, a request from a user for authorization to access a service is received. The user may request authorization to access services on Cloud 20, for example, by attempting to log on through authentication proxy 120 for access to services on Cloud 20. User 130 may make such a request by directing his browser to a web site corresponding to authentication proxy 120.

After the user logs on, at step 210, a token corresponding to a service is transmitted to the user. For example, authentication proxy 120 may transmit a token corresponding to a Cloud service after user 130 properly authenticates himself to authentication proxy 120 (or logs on) at step 200. Additionally, tokens corresponding to other Cloud services may also be transmitted to user 130.

At step 220, the token corresponding to the service is received from the user. For example, user 130, to access the service, may direct his browser to a web site corresponding to the service, e.g., the web site "geoplex.service.com." The browser may then be redirected to the web address of data proxy 110 and

transmit the cookie containing the token corresponding to the service to data proxy 110. The cookie, and hence the token, may then be received at data proxy 110.

At step 240, a determination of whether the user is
5 authorized to receive the service is made. For example, data proxy 110 may extract and decrypt the token embedded in the received cookie. Data proxy 110 may then determine whether the received cookie authorizes the user to receive the requested service. For example, data proxy 110 may extract the user identification
10 information in the token and compare it to the user identification list contained in the ACT. If the user identification information does not match any of the users in the ACT, then the user will not be connected to the requested service. Further, data proxy 110 may extract the expiration time of the token and compare it to the
15 current time. If data proxy 110 determines that the token has expired, then the user will not be connected to the requested service.

If the user identification information in the received token matches a user in the ACT and the token has not yet expired, then
20 the user may be connected to the requested service and may begin using the requested service at step 250.

Figure 3 shows another embodiment of the present invention. This embodiment illustrates the interaction of Alice, a registered Cloud user, with the Gateway. At step 300, Alice
25 directs her browser to the authentication web site (for example, the web site of the Gateway) that is managed by the authentication proxy. For example, if the address of this web site is “www.login.domain/login.html”, then Alice may enter the following command in her browser:
30 “https://www.login.domain/login.html”. This will establish a connection between Alice’s browser and the authentication proxy

that uses the Secure Sockets Layer protocol for data security.

At step 305, the authentication proxy presents Alice with a form that asks for her login name and, possibly a password.

At step 310, Alice provides her login name in the form.

5 She also types in the login password if she uses the password-based authentication method. If she uses certificate-based authentication, she does not need to type her password and this field is left blank. Alice presses the "login" button on her browser to submit the form to the authentication proxy.

10 At step 315, the authentication proxy verifies Alice's login name. If the authentication proxy does not recognize the name provided by Alice, it asks her to repeat step 310.

If Alice uses a password as a credential, then at step 320, the authentication proxy verifies the validity of the password. If
15 the password provided by Alice is incorrect, the authentication proxy asks Alice to repeat step 310. If Alice uses certificate-based authentication, then step 320 is skipped and step 325 is executed.

At step 325, the authentication proxy sends Alice a web page with a JavaScript program that contains a unique URL
20 reachable through a SSL protocol connection. The web page contains Alice's login page for the new session. The authentication proxy then closes the network connection with Alice's browser.

At step 330, Alice receives the login page. The embedded
25 JavaScript program creates a new pop-up window in Alice's browser. This window attempts to reconnect to the login page using the unique URL provided by the authentication proxy.

At step 335, the authentication proxy waits for Alice to reconnect to the authentication proxy through the unique URL.

30 At step 340, the authentication proxy accepts the new connection if Alice provided a proper certificate when the SSL

protocol negotiation for the connection took place. Otherwise, the new connection from Alice is rejected.

At step 345, the authentication protocol sends Alice a stream of http protocol cookies, where each cookie corresponds to a domain. The authentication proxy also sends a logout page to the pop-up window on Alice's computer.

At step 350, Alice's browser receives the cookies. The pop-up window displays the received logout button.

At step 360, the authentication proxy sends standby messages to the pop-up window to maintain a persistent connection. The authentication protocol also updates Alice's cookies when they are about to expire.

At step 365, the pop-up window in Alice's browser is in standby mode. Alice may discontinue her session at any time by pressing the logout button or by simply closing the pop-up window. Alice uses the cookies to transparently authenticate herself to the data proxy when establishing connections with services in the Cloud.

Figure 4 shows another illustration of the embodiment of the invention discussed in connection with Figure 3. Browser state 410 shows the state of Alice's browser after Alice directs her browser to the authentication web site (i.e., after step 300 of Figure 3). Browser state 410 is essentially a form that Alice may complete in order to logon to use services in the Cloud (i.e. to login to the Gateway), which are located behind firewall 415. Firewall 415 is implemented through authentication proxy 420 and data proxy 440.

After Alice properly authenticates herself, authentication proxy 420 stores Alice's user identification in the list of active users in ACT 435. Authentication proxy 420 also sends Alice a page with a JavaScript program that contains a unique URL.

Furthermore, authentication proxy 420 closes its connection with Alice's browser.

The JavaScript program received by Alice's browser creates pop-up window 425 and causes Alice's browser to

5 reconnect to authentication proxy 420 through the unique URL. While the connection to the unique URL is extant, Alice receives a stream of cookies, for example, cookies 450, 455, 460 and 465, from authentication proxy 420 containing tokens corresponding to services in the Cloud. When Alice logouts of the Gateway by, for

10 example, clicking on the logout box within pop-up window 425, the stream of cookies sent by authentication proxy 420 is stopped. While Alice remains logged on, she may access any Cloud service by simply directing her browser to that service. For example, she may direct her browser to the web site

15 "geoplex.domain.com/server." Her browser, detecting that the requested web site corresponds to a Cloud service, connects to data proxy 440 and sends cookie 450 corresponding to the requested Cloud Service to data proxy 440. Data proxy 440 then extracts the user identification information and other validity

20 information embedded in the cookie (or embedded in a token embedded in the cookie). Other validity information, for example, may include the token expiration date, or information on the security grade over which the cookie is sent. For example, if security grade information in the cookie specifies that the cookie is

25 to be transmitted over a connection secured through the SSL protocol, but the cookie is transmitted to data proxy 440 through an unsecured connection, then data proxy 440 may determine that the token is invalid and refuse to connect Alice to the requested service. Data proxy 440 may also check the user identification

30 information against the list of users stored in ACT 435. If no match is established, then data proxy 435 may determine that the

received token is invalid and refuse to connect Alice to the requested service.

Finally, if data proxy 440 establishes that the received token is valid, then data proxy 440 connects Alice's browser to service 445. Alice will then see home page 430 of the requested service on her screen.

Several of the embodiments above discuss a single-sign-on mechanism for services, or more generally, resources, distributed over the Internet. However, other embodiments are possible in which the services or resources are distributed over a network other than the Internet; for example, an intranet, or any other type of network.

As described above, the various embodiments of the present invention describe a single sign-on method and system for accessing a plurality of services distributed over a network in which authentication-related functionality is separated from the services, and in which authentication need not be renegotiated for access to a new service from the plurality of services during a session. Additional benefits accruing from embodiments of the invention include notification of the plurality of services when a user has terminated a session, and the use of secure, short-lived authentication tokens to verify a user's identity for subsequent access to the plurality of services.

The present invention has been described in terms of several embodiments solely for the purpose of illustration. Persons skilled in the art will recognize from this description that the invention is not limited to the embodiments described, but may be practiced with modifications and alterations limited only by the spirit and scope of the appended claims.

Claims:

1. A method for authenticating a user for connection to a service,
the service implemented on at least one server, the method
5 comprising:
receiving a request from a user for authorization to access a
service;
transmitting a token corresponding to the service to the
user;
10 receiving the token corresponding to the service from the
user;
determining whether the user is authorized to receive the
service based on the token; and
connecting the user to the service, if the user is authorized
15 to receive the service.
2. The method of claim 1 further comprising the steps of
determining whether the user is authorized to access the service
and transmitting the token to the user only if the user is authorized
20 to access the service.
3. The method of claim 2 further comprising the step of
registering the user for authorization to access to the service, the
registering including assigning the user a username and at least
25 one of a password and a digital certificate for use in authorization
to access the service.
4. The method of claim 3 wherein the user is authorized to access
the service only after receiving from the user the username and the
30 at least one of the password and the digital certificate.

5. The method of claim 1 wherein the token is transmitted to the user over a first channel and wherein the token is received from the user over a second channel.
- 5 6. The method of claim 5 wherein connection through the first channel is maintained while the token is received over the second channel.
7. The method of claim 5 wherein the token is transmitted to the user in response to a request signal from the user.
- 10 8. The method of claim 5 wherein the token is transmitted to the user automatically.
- 15 9. The method of claim 5 wherein the token is valid only for a period of time and the user is not authorized to receive the service after the period of time expires.
10. The method of claim 9 wherein the length of the period of time is predetermined.
- 20 11. The method of claim 9 wherein the length of the period of time is random.
12. The method of claim 9 wherein the period of time expires upon the occurrence of a security event.
13. The method of claim 5 wherein the token is embedded in a cookie.
- 30 14. The method of claim 5 wherein the token contains information

specifying a security grade of the second channel.

15. The method of claim 5 wherein the token that is transmitted and received is a first token and a second token corresponding to the service is transmitted to the user over the first channel.

16. The method of claim 15 wherein the first token is encrypted using a first key, the first key not being known to the user, and the second token is encrypted using a second key, the second key not being known to the user.

17. The method of claim 1 wherein all the steps are performed on at least one server, not including the at least one server on which the service is implemented

15

18. The method of claim 17 wherein the at least one server on which all the steps are performed is remote from the at least one server on which the service is implemented.

19. A method for authenticating a user for connection to a plurality of services, each service of the plurality implemented on at least one server, the method comprising:

receiving a request from a user for authorization to access a plurality of services;

transmitting a token corresponding to a service from the plurality of services to the user through a channel;

receiving a request to access the service from the user;

receiving the token corresponding to the service from the user;

determining whether the user is authorized to receive the service based on the token; and

connecting the user to the service, if the user is authorized to receive the service.

20. The method of claim 19 wherein all the steps are performed
5 on at least one server, the at least one server not including the at least one server on which the service is implemented.

21. The method of claim 20 wherein the at least one server on
which all of the steps are performed is remote from the at least one
10 server on which the service is implemented.

22. The method of claim 19 wherein the token is embedded in a cookie.

15 23. The method of claim 19 wherein the token contains information specifying a security grade of a connection over which the token is received.

24. The method of claim 19 wherein the token is encrypted.

20

25. The method of claim 24 wherein the token is valid for a predetermined amount of time.

26. The method of claim 24 wherein the token is valid for a
25 random period of time.

27. The method of claim 19 wherein the steps of receiving the request to access the service, receiving the token corresponding to the service and connecting the user to the service are performed
30 through at least one connection, the at least one connection not including the channel.

28. The method of claim 27 wherein the channel is maintained during the steps of receiving the request to access the service, receiving the token corresponding to the service and connecting
5 the user to the service.

29. The method of claim 19 wherein the channel is implemented using a secure communications protocol.

10 30. The method of claim 29 wherein the secure communications protocol is the SSL protocol.

31. The method of claim 29 wherein the secure communications protocols is the TLS protocol.

15

32. The method of claim 19 wherein the request from the user for authorization to access the plurality of services includes at least one of:

- a) a username and a password, and
- 20 b) the username and a digital certificate.

33. A system for authenticating a user for connection to a plurality of services, each service of the plurality implemented on at least one server, each at least one server capable of being
25 connected to at least one of a data proxy and an authentication proxy, the system comprising:

- an authentication proxy that is connected to a user through a first channel and that transmits a plurality of tokens to the user through the first channel; and
- 30 a data proxy that is connected to the authentication proxy, the user and a plurality of services, the data proxy

(a) receiving at least one token from the plurality of tokens from the user through a second channel, the at least one token corresponding to at least one service from the plurality of services, and

5 (b) determining whether the user is authorized to receive the at least one service based on the at least one token.

34. The system of claim 33 wherein each of the authentication proxy and the data proxy is remote from the at least one service.

10

35. The system of claim 33 wherein each token from the plurality of tokens is embedded in a cookie.

36. The system of claim 33 wherein the at least one token
15 contains information specifying a security grade of a connection over which the at least one token is transmitted from the user.

37. The method of claim 33 wherein the at least one token is encrypted.

20

38. The method of claim 37 wherein the at least one token is valid for a predetermined amount of time.

39. The method of claim 37 wherein the at least one token is valid
25 for a random amount of time.

40. The method of claim 33 wherein the first channel is implemented using a secure communications protocol.

30 41. The method of claim 40 wherein the secure communications protocol is the SSL protocol.

42. The method of claim 40 wherein the secure communications protocols is the TLS protocol.

5 43. A system for providing authentication functionality allowing authorized users to access a plurality of services, the system comprising a gateway that authenticates a user for access to at least one service, all of the authentication functionality of the system exists only on the gateway.

10

44. The system of claim 43 wherein a new service added to the plurality of services for which the gateway performs authentication functionality does not require augmentation for proper functioning of the authentication functionality of the
15 system.

45. The system of claim 44 wherein the gateway and the at least one service are connected through a remote connection.

20 46. The system of claim 44 wherein:

a first connection between the user and gateway is used for transmitting a plurality of tokens from the gateway to the user; and

a second connection between the user and gateway is used for transmitting at least one token from the plurality of tokens
25 from the user to the gateway, the at least one token corresponding to at least one service and authorizing access of the user to the at least one service.

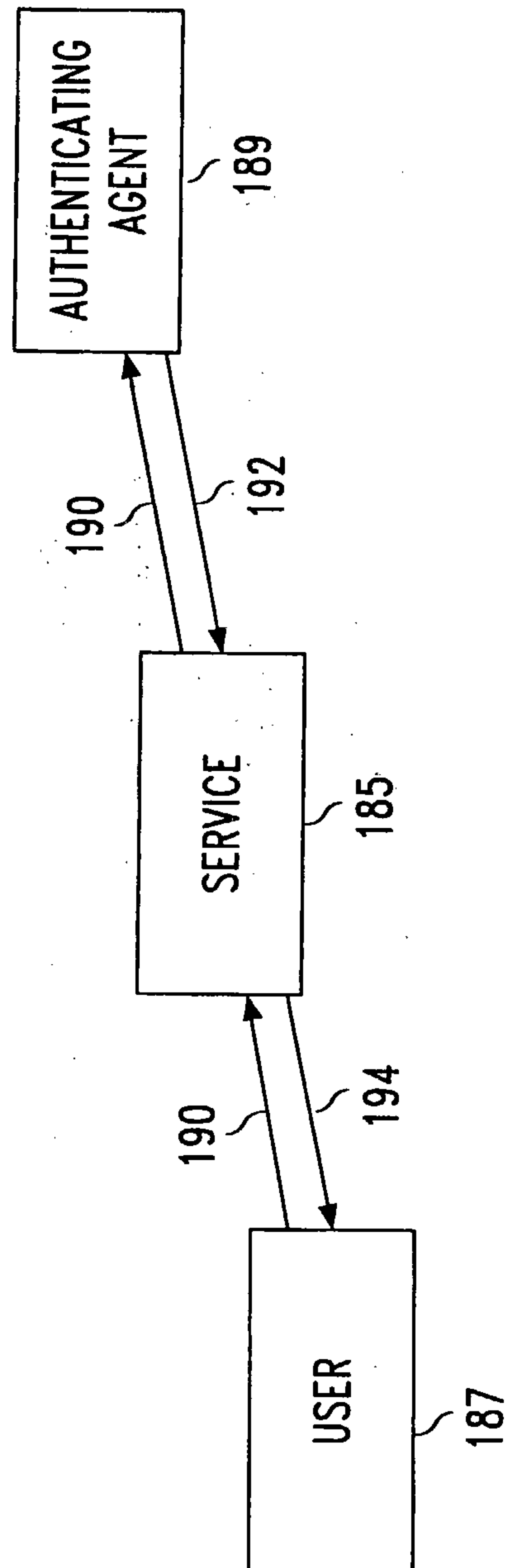
47. The system of claim 46 wherein the user is connected to the at
30 least one service through the second connection and a third connection, the third connection existing between the gateway and

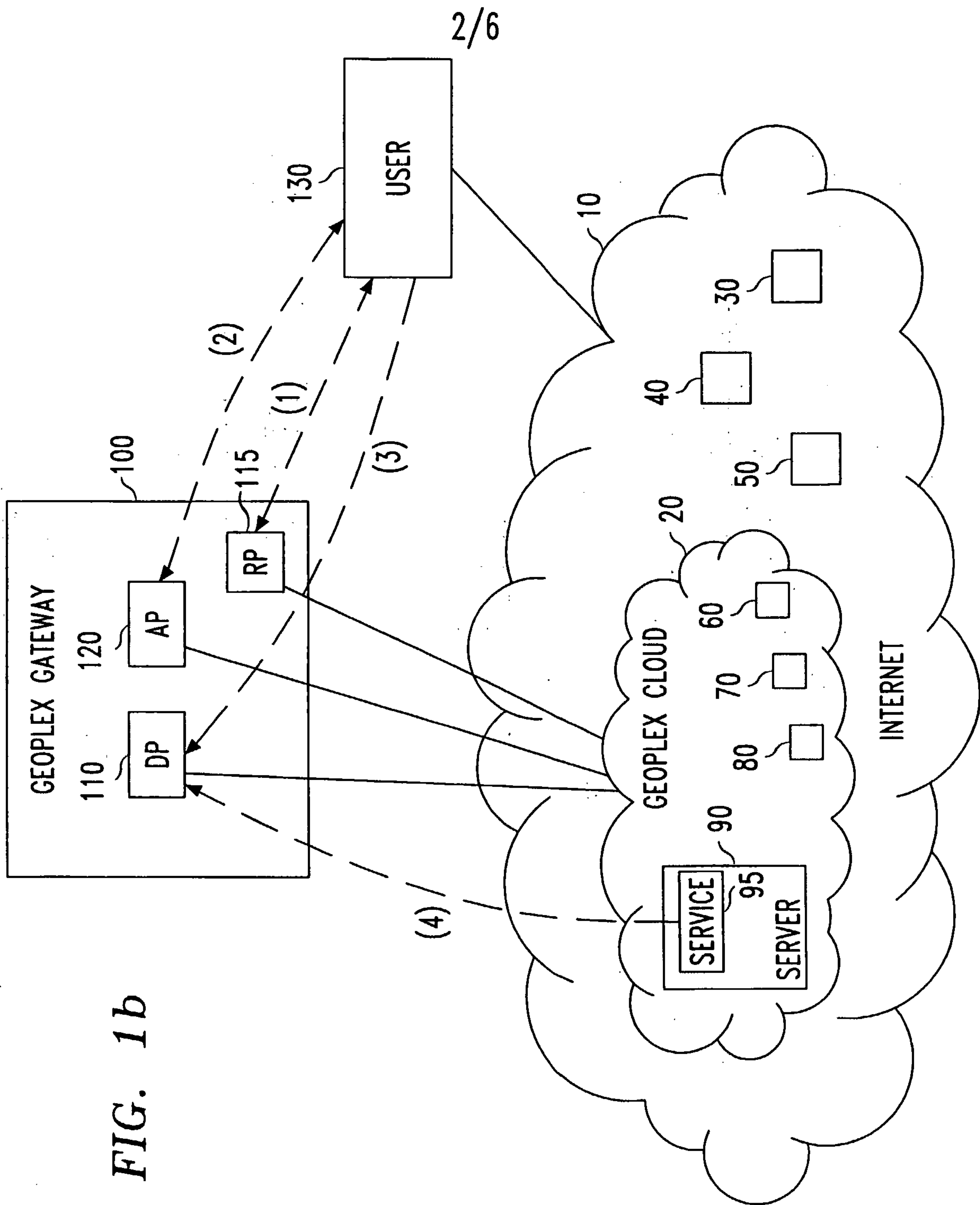
the at least one service.

48. The system of claim 47 wherein the gateway comprises an authentication proxy and a data proxy, the authentication proxy
5 connected to the user through the first connection and the data proxy connected to the user through the second connection.

PRIOR ART
FIG. 1a

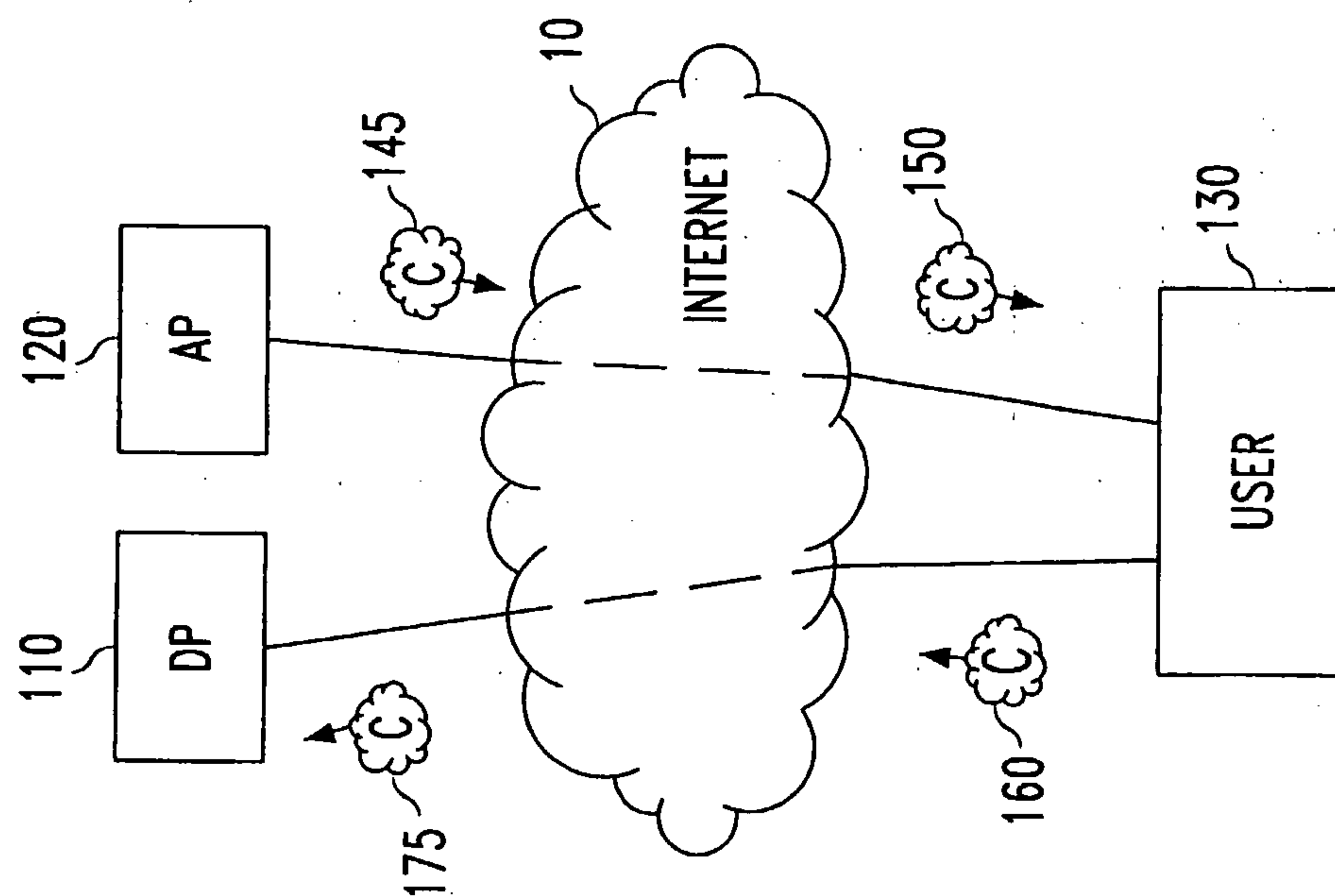
1/6



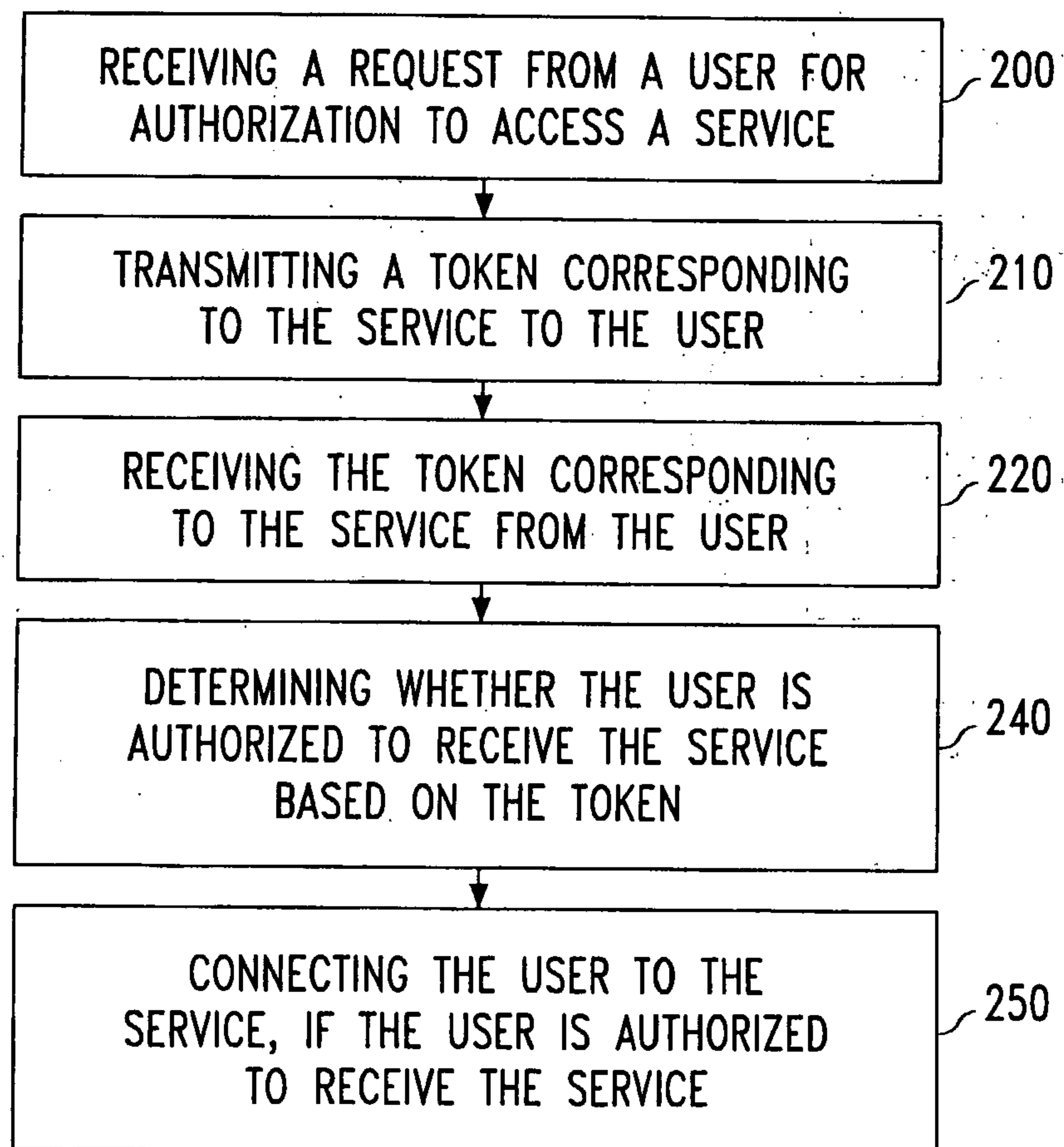


3/6

FIG. 1C

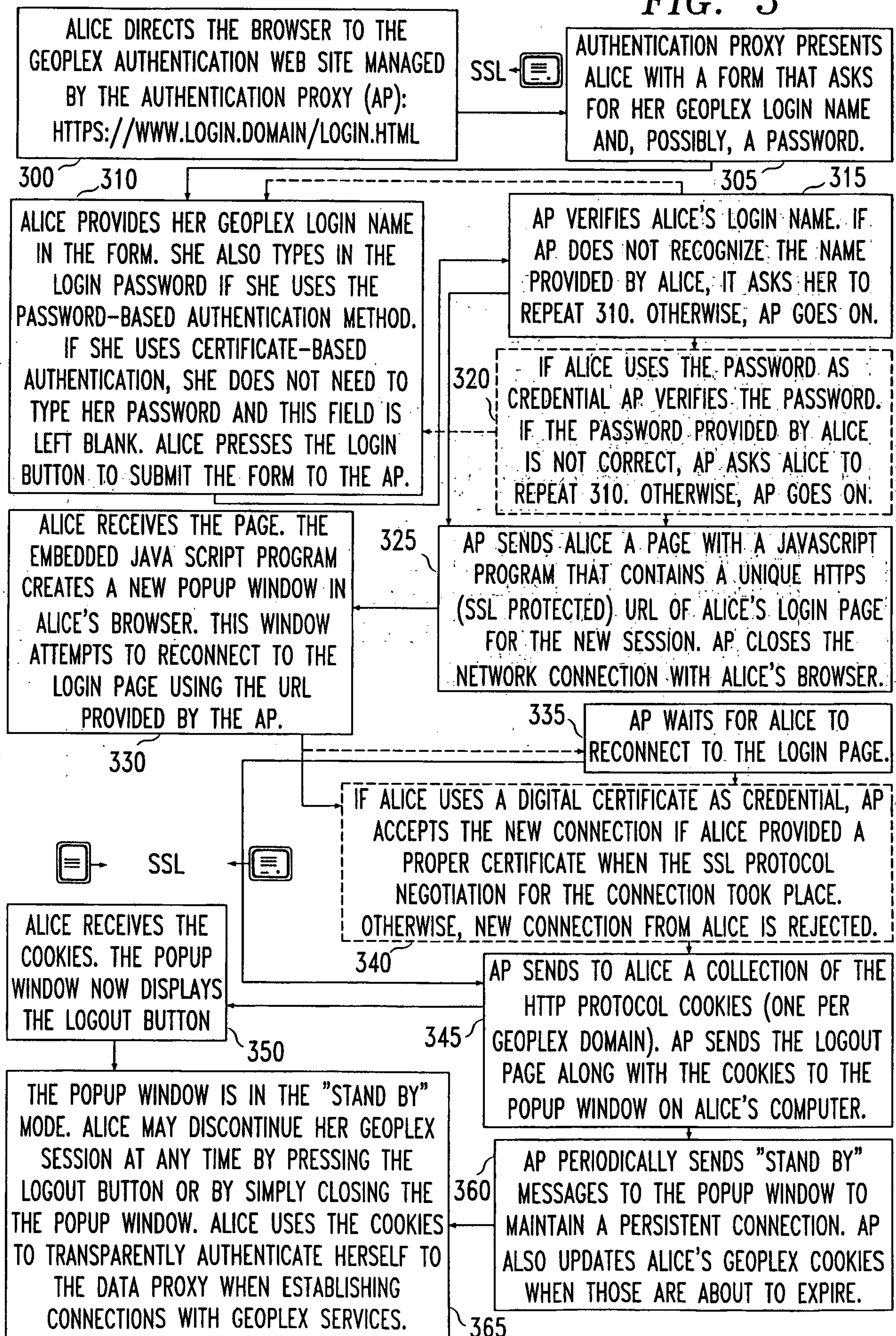


4/6

FIG. 2

5/6

FIG. 3



6/6

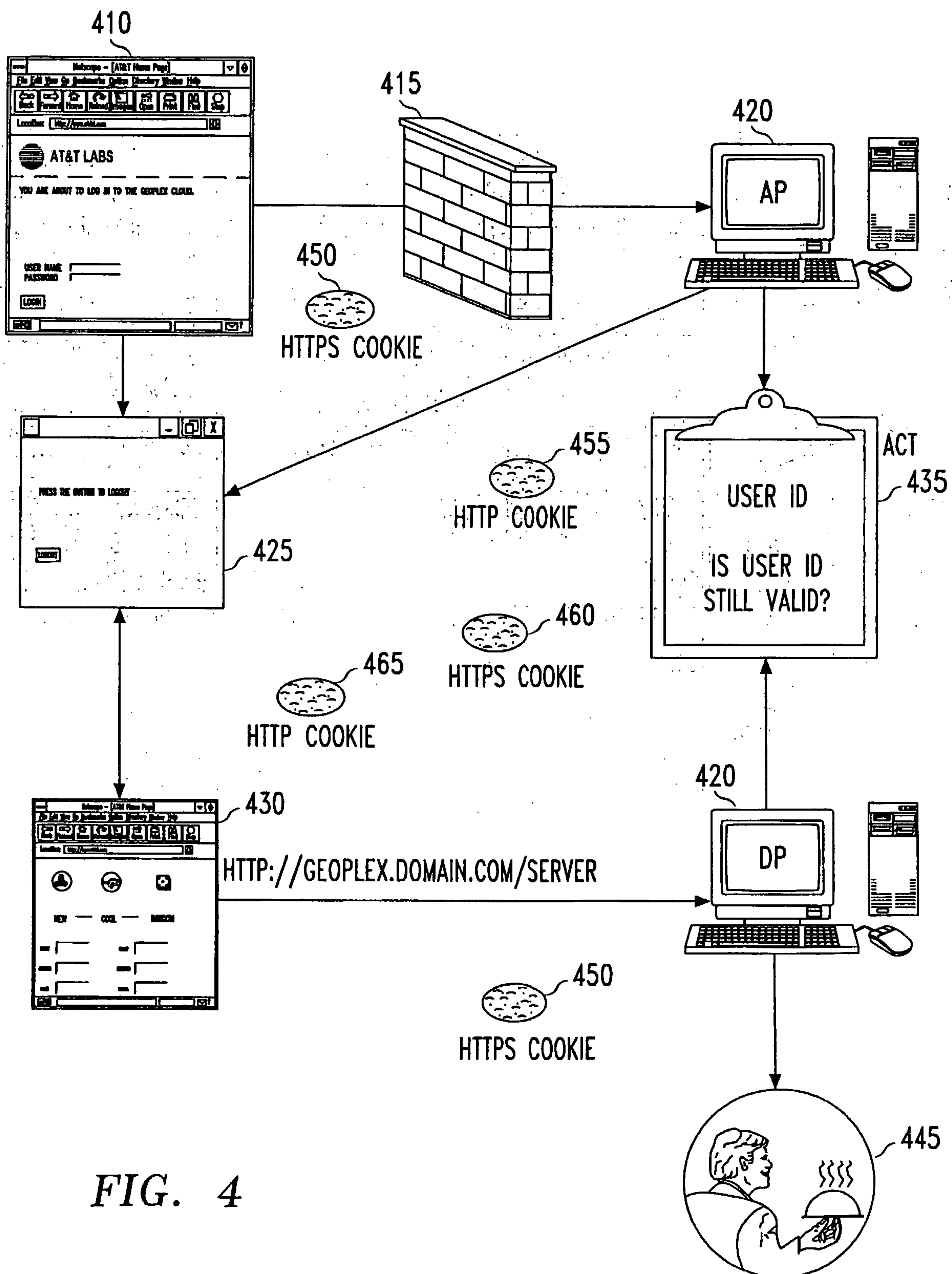


FIG. 4

