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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
28 April 2011 (28.04.2011)

PCT

(10) International Publication Number
WO 2011/047734 A1

(51) International Patent Classification:
H04L 29/06 (2006.01)

(21) International Application Number:
PCT/EP2009/063971

(22) International Filing Date:
23 October 2009 (23.10.2009)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant *(for all designated States except US)*: **NOKIA SIEMENS NETWORKS OY** [FI/FI]; Karaportti 3, FIN-02610 Espoo (FI).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **MARTON, Gabor** [HU/HU]; Jagelló út 42/A, H-1124 Budapest (HU). **BODI, Miklos Tamas** [HU/HU]; Egri uti lakopark 55, H-3326 Ostoros (HU).

(74) Common Representative: **NOKIA SIEMENS NETWORKS OY**; COO RTP IPR/Patent administration, 80240 Munich (DE).

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

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, 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, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

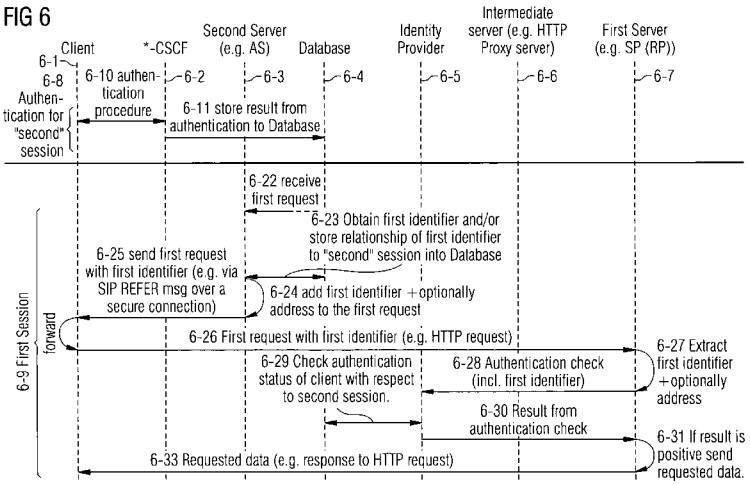
Published:

— *with international search report (Art. 21(3))*

WO 2011/047734 A1

(54) Title: METHOD, APPARATUS AND RELATED COMPUTER PROGRAM PRODUCT FOR A CLIENT AUTHENTICATION WHEN STARTING A NEW SESSION

FIG 6



(57) Abstract: Disclosed is a method for optimizing authentication procedures of a client. When a client starts multiple sessions with one or more service provider in parallel, and must authenticate for at least two of those sessions separately, an earlier successfully performed authentication will be used for authenticating the later started session (s). Re-using of the authentication is handled by the network (s) where the client is connected to without requesting any data or information from the client. A first identifier, identifying the client, is added by a network element to a first request for the later session. This identifier is used to check if the client was authenticated already earlier. Further on it is disclosed to add the address of an identity provider, in addition to the client identifier, to the first request.

DESCRIPTION

TITLE

Method, Apparatus and related computer program product for a client authentication when starting a new session

FIELD OF THE INVENTION

[0001] The present invention relates to the field of authentication and identity management. More specifically, the present invention relates to a method, an apparatus and a related computer program product being part of an authentication procedure when a new session between a client and a server is started and the client must authenticate for the new session.

[0002] Examples of the present invention are applicable, but not limited, to elements of an Internet Protocol Multimedia System (IMS).

BACKGROUND

[0003] A client can set-up multiple sessions of different types, for example via an IP Multimedia Subsystem. Those sessions might be multimedia communications sessions (like e.g. voice and video calls) or data sessions (like e.g. Web sessions). Typically the client must authenticate separately for sessions of different types.

[0004] If a client is involved in a multimedia session (for example a voice or videoconference call), and a parallel data session shall be started (for example for retrieving data stored on a web server), the client must actively authenticate itself with the data service provider in order to be able to establish the data connection and obtain the requested data.

[0005] Even if the client is already authenticated, for example when he registered with the IMS for starting the multimedia session, the client must authenticate itself again with the data service provider in order to start the parallel data session. If more (data) sessions (e.g. with different data service providers) shall be started in parallel the client must authenticate for each session separately.

[0006] As an example it is assumed that Bob and Alice are in a voice call, started via SIP by Bob. During the voice call Alice wants to show Bob a photo which is available at a photo sharing service (e.g. over HTTP). Alice sends the URL of the photo to Bob. When Bob tries to access the photo, he starts to establish a parallel data session with the photo sharing service. The photo sharing service requires that Bob first authenticates himself and presents a logon dialog to him asking for his logon credentials, even if Bob has already authenticated himself when establishing the voice call with Alice.

[0007] The separate authentication actions for the different sessions require each time active manual actions from the client (e.g. providing provider specific user identifier and password), which is a time consuming and annoying task to be performed by the client.

[0008] From the state of the art OpenID is known as one concept for single sign on. The OpenID concept requires that the client first registers with an OpenID provider and both agree on a dedicated OpenID for the client. Once this is done, the OpenID Provider (OP) can perform the authentication of the client and provide it as a service to Relying Parties (RPs) the client is trying to access. When for example a client tries to access a resource at a service provider, which is an OpenID RP, the service provider asks the client for its OpenID identifier. The OpenID identifier is typically a URL pointing to the OP, on which basis the service redirects the client to the OP for authentication (asked by

means of attaching an authentication request). Having authenticated the client, the OP redirects the client back to the service provider (attaching an authentication response). Note that the client must provide the OpenID identifier to the service provider, so that the service provider can redirect the client to the OP. In order to be able to provide the OpenID identifier to the service provider, the user typically has to type it into an online form.

[0009] However, situations will occur where no OpenID has been defined for the client or where an OpenID is defined but the client is not aware about it. Furthermore, the client will typically be asked for authentication related data, for example an OpenID identifier or user credentials, before a session is started and a new service or data can be accessed, thus active input is required from the client at least once before a session is started.

[0010] In consideration of the above, it is an object of the present invention to overcome the above drawbacks. In particular, the present invention provides a method, an apparatus and a related computer program product for optimizing an authentication procedure of a client when establishing sessions to e.g. multiple services and/or service providers in parallel.

[0011] According to an example of the present invention, in a first aspect, this object is for example achieved by **a method** being part of an authentication procedure of a client when a first session between the client and a first server is started, where the client has not been authenticated for the first session and where a first request is requesting data related to the first session from the first server, characterised by

- obtaining a first identifier, identifying the client,
- adding the first identifier to the first request,
- transmitting the first request with the first identifier to the first server,

wherein the first identifier is related to a second identifier identifying the client and wherein the second identifier was used for authenticating the client for a second session.

[0012] According to further refinements of the example of the present invention as defined under the above first aspect, the method further comprises
- obtaining the first identifier by generating the first identifier or by retrieving the first identifier from a database
- storing the generated first identifier in the database
- generating relationship information between the first identifier and the second identifier and storing it in the database
- requesting with the first request for the first time data related to the first session from the first server
- requesting data or information from the first server as part of the first request which is not related to the authentication of the client
- adding the first identifier to the first request by a network element located between the client and the first server,
- wherein the network element is located between the client and the first server is a proxy server
- wherein the proxy server is a HTTP proxy sever
- adding the first identifier to the first request by a second server, where the second server sends the first request, including the first identifier, to the client which then forwards the request, including the first identifier, to the first server
- where the connection between the second server and the client is a secure connection
- where the second server is an Application Server of a first Operator
- wherein the first request and the first identifier are included in a SIP message
- wherein the SIP message is a SIP REFER message

- wherein the first Operator is an IMS Operator
- wherein the first identifier is a nonce identifier
- wherein the first identifier is an OpenID identifier
- wherein the first identifier includes address information of an Identity Provider
- wherein the first session is a HTTP session, the first request is a HTTP request and the first server is a HTTP server
- wherein the authentication of the client for the first session is assumed to be successfully completed if the client was already authenticated successfully for the second session

[0013] According to an example of the present invention, in a second aspect, this object is for example achieved by **a device** being part of an authentication procedure of a client when a first session between the client and a first server is started, where the client has not been authenticated for the first session and where the first request is requesting data related to the first session from the first server, the device comprising
- an obtaining means obtaining a first identifier identifying a client,
- an adding means for adding the first identifier to the first request,
- a transmitting means for transmitting the first request with the first identifier towards the first server,

wherein the first identifier is related to a second identifier identifying the client and where the second identifier was used for authenticating the client for a second session.

[0014] According to further refinements of the example of the present invention as defined under the above second aspect,
- the obtaining means is configured to generate the first identifier or to retrieve the first identifier from a database

- the obtaining means is configured to store the generated first identifier in the database
- the obtaining means is configured, if the first identifier is generated, to generate relationship information between the first identifier and the second identifier and to store the generated first identifier together with the relationship information in the database
- wherein the first request is requesting for the first time data related to the first session from the first server
- wherein at least one part of the first request is requesting data or information from the first server which is not related to the authentication of the client
- wherein the device is located between the client and the first server
- where the device is a proxy server
- where the proxy server a HTTP proxy server
- wherein the device is a second server and where the second server sends the first request, including the first identifier, to the client which then forwards the request, including the first identifier, to the first server
- where the connection between the second server and the client is a secure connection
- wherein the second server is an Application Server of a first Operator
- wherein the first request and the first identifier are included in a SIP message
- wherein the SIP message is a SIP refer message
- wherein the first Operator is an IMS operator
- wherein the first identifier is a nonce identifier
- wherein the first identifier is an OpenID identifier
- wherein the first identifier includes address information of an Identity Provider
- wherein the first session is a HTTP session, the first request is a HTTP request and the first server is a HTTP server
- wherein the authentication of the client for the first session is assumed to be successfully completed if the client was already authenticated successfully for the second session

[0015] According to an example of the present invention, in a third aspect, this object is for example achieved by **an apparatus** being part of an authentication procedure of a client when a first session between the client and a first server is started, where the client has not been authenticated for the first session and where the first request is requesting data related to the first session from the first server, the device comprising

- an obtaining block configured to obtain a first identifier identifying a client,
- an adder configured to add the first identifier to the first request,
- a transmitter configured to transmit the first request with the first identifier towards the first server,

wherein the first identifier is related to a second identifier identifying the client and where the second identifier was used for authenticating the client for a second session

[0016] According to further refinements of the example of the present invention as defined under the above third aspect,

- the obtaining block is configured to generate the first identifier or to retrieve the first identifier from a database
- the obtaining block is configured to store the generated first identifier in the database
- the obtaining block is configured, if the first identifier is generated, to generate relationship information between the first identifier and the second identifier and to store the generated first identifier together with the relationship information in the database
- wherein the first request is requesting for the first time data related to the first session from the first server
- wherein at least one part of the first request is requesting data or information from the first server which is not related to the authentication of the client

- wherein the device is located between the client and the first server
- where the device is a proxy server
- where the proxy server a HTTP proxy server
- wherein the device is a second server and where the second server sends the first request, including the first identifier, to the client which then forwards the request, including the first identifier, to the first server
- where the connection between the second server and the client is a secure connection
- wherein the second server is an Application Server of a first Operator
- wherein the first request and the first identifier are included in a SIP message
- wherein the SIP message is a SIP refer message
- wherein the first Operator is an IMS operator
- wherein the first identifier is a nonce identifier
- wherein the first identifier is an OpenID identifier
- wherein the first identifier includes address information of an Identity Provider
- wherein the first session is a HTTP session, the first request is a HTTP request and the first server is a HTTP server
- wherein the authentication of the client for the first session is assumed to be successfully completed if the client was already authenticated successfully for the second session

[0017] According to an example of the present invention, in a fourth aspect, this object is achieved by **a computer program product** comprising code means for performing method steps of a method according to any one of the above first aspect, when run on a processing means or module.

[0018] According to an example of the present invention, in a fifth aspect, this object is achieved by **a computer program product** embodied on a computer-readable medium encoded with instructions that, when executed on a computer, perform a method according to any one of the above first aspect.

[0019] The embodiments of the present invention can provide one or more of the following advantages:

- A client is not required to perform a dedicated authentication for the first session if he already performed a successful authentication for another session (for example for a session with a different service provider)
- The first server, where the first request is targeted to, is not required to know the address of the identity provider (which the first server must contact to verify the authentication status of the client) in advance, since the first server might receive it together with the first request and the first identifier identifying the client.
- The client is not asked for any authentication related information (for example for an Open Identifier) with respect to a newly started first session when he already authenticated himself successfully earlier (for example for a different session with a different service provider).
- The client does not need to know or store the first identifier.
- The client is not required to register to an OpenID provider and agree with the OpenID provider on an OpenID in advance.

BRIEF DESCRIPTION OF THE DRAWINGS

[0020] Examples of the present invention are described herein below with reference to the accompanying drawings, in which:

[0021] Fig. 1 illustrates a first example embodiment of a high level network architecture where the invention is used; and

[0022] Fig. 2 presents a second example embodiment of a high level network architecture where the invention is used; and

[0023] Fig. 3 shows method steps related to the present invention; and

[0024] Fig. 4 illustrates possible internal structures and functions of devices implementing aspects of the invention; and

[0025] Fig. 5 shows a first example signaling diagram of authenticating a first session according to the present invention; and

[0026] Fig. 6 presents a second example signaling diagram of authenticating a first session according to the present invention.

DETAILED DESCRIPTION OF THE PRESENT INVENTION

[0027] Examples of the present invention are described herein below with reference to the accompanying figures. The figures include mandatory, as well as optional elements, related to the present invention. Furthermore, the figures only include elements that are important for the present invention, or that are useful in the context of describing the present invention. Not important network elements, messages or signals (like for example elements where information is just relayed/passed through, or messages just acknowledging the receipt of another message) might have been left out of the figures and the description for simplification purposes.

[0028] In the description the terms OpenID Provider (OP), Proxy Server, OpenID identifier, nonce identifier, Internet Protocol Multimedia Subsystem (IMS), HyperText Transfer Protocol (HTTP) and Session Initiation Protocol (SIP) are

examples for elements, functions, protocols and networks without restricting or limiting them to functions, elements, protocols or networks of this specific type, or excluding any possible alternatives. The described embodiments are not limited to the mentioned network elements, messages and signals.

[0029] The following paragraphs define certain terms and elements used throughout this application. These definitions are related to example embodiments of the invention as described below and might not be directly applicable to other, alternative, embodiments of the invention not described within this document.

[0030] An OpenID identifier is an identifier used to identify a client, for example by different service providers. An OpenID identifier typically includes sections directly related to the client, for example in the OpenID identifier http://csp.com/bob123, "bob123" identifies the client within the realm of the OpenID Provider (OP). Furthermore, the OpenID identifier might include the domain name of the OP (for example "csp.com"), which could be for example an operator operating one of the networks where the client (e.g. bob123) is authenticated and/or connected with. Since the OpenID identifier is a fixed, predictable identifier, it might be a target for impersonation attacks. Therefore it is a recommended option to transfer the OpenID identifier over secure connections.

[0031] An OpenID provider (OP) is a provider offering authentication of a client based on an OpenID identifier. If the client has been authenticated once successfully with the OP, the OP will automatically (without involving the client again) assert further authentication requests to other service providers for the same client, based on the provided (and authenticated) OpenID identifier of the client.

A "nonce" identifier is a generated, temporary, client identifier. The nonce identifier could be a separate identifier used for example instead of an OpenID identifier, or it could be used for example as a "temporary valid" OpenID identifier where the "URI" part might be the nonce. The nonce identifier does not include any further information about the client it identifies (e.g. it does not contain a real client name). However, it might include address information related to an identity provider. Furthermore, the nonce identifier might be a random identifier which makes it to an unpredictable identifier, so it is a good measure against impersonation attacks and might be preferred to "predictable", fixed identifiers. An example nonce identifier is: "http://csp.com/FIAkGJhQRAAAAAAAAAAAAA==". In the example "csp.com" is the address information related to an identity provider.
In the present invention, the nonce identifiers are valid by possession, therefore it is a recommended option to transfer them only via secure channels, connections and protocols.

[0032] By the term identifier, an identifier identifying a single client is meant. Such an identifier could be for example an OpenID identifier, a nonce identifier, any kind of other identifier related to a single client or a combination of different identifiers (for example of an OpenID identifier and a nonce identifier). Multiple identifiers might be assigned to a single client (for example the same client might have a different identifier for each service provider he is communicating with).

[0033] The term identity provider is a general term used for any kind of provider being able to provide an identification / authorization service on behalf of the client. A special type of an identity provider is an OpenID provider as known from the state of the art.

[0034] The terms "first" and "second", as used throughout this document, do not define any order of elements or

occurrences, or where a certain element belongs to. For example a "second" session might be set-up before the "first" session and vice versa, and a "second" server is not required to be automatically located in the network of the "second" operator.
In general, the terms "first" and "second" shall not cause and limitation with respect to occurrences, ordering or grouping of network elements, actions, sessions or anything else.

[0035] The term "first request" in this application is used as a generic term for a first request sent towards a first server. The "first request" might pass through various intermediate network elements before the first server is reached. Between the different intermediate network elements and/or the first server the used protocols might vary and the first request might be modified by intermediate network elements (for example by adding or modifying information related to the first request and/or encapsulating the first request in another message).

[0036] In general all shown figures relate to example scenarios where a client, who is already authenticated and possibly connected to a network of a first operator, wants to establish a session with an application server located possibly in the network of second operator. In order to establish the session the client has to authenticate first with the application server in the second operator network. Since the client already authenticated when connecting to the network of the first operator, this authentication could be reused when establishing the session with the application server located in the network of the second operator, so no extra authentication is needed. For this purpose a first identifier identifying the client, and optionally additional information (for example the address of the identity provider - the node to be contacted for performing the check if the client is already authenticated), are added by another network element (not the client) to a first request targeted

to the application server when starting the session. With this information the application server is able to perform an authentication check without requesting any further information from the client. If the authentication check reveals that the client was authenticated successfully earlier (for example for a session with the network of the first operator), the application server regards the client as authenticated and sends directly the requested data without challenging the client for authentication purposes.

[0037] The main advantages of the above outlined concept are:

a) No dedicated authentication of the client for the session to be started with the application server in the network of the second operator is needed before data can be requested by the client from the application server.
b) The client is not required to be registered to an Identity Provider.
c) The client is not required to know the used identifier, identifying the client.
d) The client is not asked for any authentication related information (like for example a used identifier for identifying the client).
e) The application server is informed about the address of the identity provider, there is no need to pre-program the address or query it from another network element.

[0038] The first identifier, and other related authentication information, are transferred from the first to the second operator domain without any active involvement of the client (the client is not actively asked to provide information like the first identifier, instead the client might be requested to passively forward a received first request including the first identifier and the authentication information).

[0039] The first request towards the application server is used as a "transport medium" for authentication and identity related information (for example a first identifier and/or an Identity Provider address). Furthermore, the first request is used as a trigger to start the checking of the authentication status of the client at the application server. The whole authentication checking procedure happens in the background and is not actively involving the client.

[0040] Fig. 1 to fig 6 illustrate exemplary embodiment of the present invention. Only elements necessary to describe the present invention are included in the figures, all other network elements, not involved in authentication and identity management related to the present invention, are not shown.

[0041] Following elements are included in fig. 1 and fig. 2:

- First operator 1-1/2-1 network, for example an IP Multimedia Subsystem (IMS) providing multimedia services to the client.
- Second operator 1-7/2-7 network providing additional services to the client, for example data services like a photo sharing service.
- The client (UE) 1-5/2-5, connected directly to the network of the first operator (e.g. to an application server in an IMS network). The client could be for example a mobile or fixed device.
- A database 1-2/2-2 storing authentication related information. The database might contain one or more tables providing client specific information like client specific identifiers (for example SIP identifier, OpenID identifier, nonce identifiers ...) used for different sessions and/or relationship information between them. One client might have multiple assigned identifiers, for example for different Service Providers.

The database can be located inside or outside the network of the first operator and might be integrated into another network element (for example the second server 2-3 of the first network operator).

- An identity provider 1-4/2-4, which might be part of the first operator network or located outside. The identity provider is used as a central point for identity provisioning and checking. It can be contacted for example by different application servers for client authentication checking purposes. The identity provider could be for example an OpenID Provider (OP).
- A first server 1-8/2-8 (e.g. an application server) of a service provider located in a network of the second operator. The first server could be for example an application server for sharing data (e.g. photos). The first server is typically located outside of the network of the first operator.
- Only in fig. 1: An intermediate network element 1-6 located between a client and an application server of a service provider accessed by the client. The intermediate network element 1-6 is part of the first operator network and could be for example a HTTP proxy server.
- Only in fig. 2: A second server 2-3, for example an Application Server (AS) for serving SIP sessions. The second server is typically located in the network of the first operator, but might be also located outside. The database 1-2/2-2 could be integrated into the second server.

[0042] Fig 5 and fig 6 include similar elements. Therefore the above listed elements and their description apply also to the corresponding elements in fig 5 and fig 6.

[0043] As a pre-requisite to the example embodiments shown in fig. 1 and fig. 2 it is assumed, that the client 1-5/2-5 has already authenticated successfully with the network of the first operator, for example when the client established a session (e.g. a phone call, called a "second" session) via the network of the first operator. The network of the first operator could be for example an IMS network. Such an authentication could happen for example in the case of an IMS

involving *-CSCF (P-CSCF, S-CSCF, I-CSCF) and the HSS network elements. Information about the successful authentication for the second session is stored in a database 1-2/2-2 (for example the SIP identifier of the client, relationship of the SIP identifier to one or more identifiers of the same client stored in the database, the authentication status of the client …).

[0044] If this pre-requisite is not fulfilled, the client can be "pushed" to initially authenticate with the network of the first operator in order to fulfill the pre-requisite (this is further detailed below in the description related to fig 1 and fig 2).

[0045] Turning to fig. 1. Client 1-5 wants to start a first session with a first server 1-8 of a service provider located in a network of a second operator. A first request to start the first session could be either generated by the client itself (for example initiated by the user or queried to be started from external 1-10) or received from external 1-10. The query to start a first session or the first request received from external 1-10 at the client 1-5 might be issued for example by a remote client (a remote client where client 1-5 is already in a call with) or by another source/network element (not shown). If the client 1-5 receives a query to start a first session the client generates the first request by itself. If the client 1-5 receives the first request (for example encapsulated in a message like a SIP REFER message) it extracts the first request and "forwards" it to the first server.

[0046] A query to start a first session might be a push message received by the client, which could be for example a WAP Push message, a GBA Push message or a SMS request.

[0047] The first session might be a HTTP session. The first request might be a HTTP request to retrieve the content

related to a HTTP URL from the first server (for example a HTTP server), for example a HTTP GET request.

[0048] In order to utilize an already performed authentication of the client 1-5 for the second session with the network of the first operator (the prerequisite mentioned above), an intermediate network element 1-6 located in the network of the first operator 1-1 (for example a HTTP proxy server), will receive the first request 1-11 from the client 1-5 targeted to the first server 1-8. The intermediate network element 1-6 will analyze the request, identify the client who has sent the request, and start an action to obtain a first identifier for the client, which can be used for checking the authentication status of the client.

[0049] Obtaining the first identifier in the intermediate network element 1-6 could be done by issuing a request 1-18 to the database 1-2 in order to check if the client 1-5 was already authenticated earlier for a different session (for example the second session mentioned in the prerequisite section above). If the client was already authenticated earlier, a first identifier might be either retrieved from the database 1-2, or a new first identifier might be generated for the client 1-5 by the intermediate network element 1-6. Generating a new first identifier can be done for example by using a pseudo-random number generator (generating a nonce), or any other client related information (for example an OpenID identifier or a SIP address of the client) available in the intermediate network element and/or information retrieved from the database 1-2.

[0050] The first identifier is preferably a “nonce” identifier or an OpenID identifier, or a combination of both, and if it is a new generated first identifier it might be stored in the database 1-2 optionally together with relationship information to at least one other identifier for the same client 1-5. The at least one other identifier might be already stored in the database or stored together with the

newly generated first identifier and the relationship information in the database. At least one other identifier could be for example the IP address of the client or a SIP identifier of the client. The SIP identifier of the client might be already stored in the database when the client authenticated for the seconds SIP session earlier.

[0051] The first identifier might also include an address of an OpenID Provider 1-4, the address might be retrieved also from the database 1-2.

[0052] The intermediate network element 1-6 inserts the first identifier into the first request 1-11 received from the client 1-5 and forwards the first request together with the inserted first identifier 1-12 towards the server 1-8 in the network of the second operator 1-7.

[0053] The connection between the intermediate network element 1-6 located in the network of the first operator 1-1 and the first server 1-8 located in the network of the second operator 1-7 is preferably (but not limited to) a secure connection.

[0054] The first server 1-8 extracts the first identifier from the received first request 1-12 and starts an authentication check 1-13 with an identity provider 1-4. In order to contact the identity provider 1-4 and perform the check the first server 1-8 needs to known the address of the identity provider 1-4. The first server 1-8 is either already aware of the address of the identity provider 1-4 (e.g. pre-programmed or informed about it beforehand), requests it from the some other node or extracts it from the first identifier of the received request 1-12.

[0055] The identity provider 1-4 is then contacting in step 1-14 the database 1-2 in order to check if for the client, related to the first identifier provided by the first server 1-8, a successful authentication was performed earlier (for

example for an earlier started second session like a voice session). Via the database 1-2 the received first identifier of the client 1-5 could be matched to other (for example a second) identifier(s) of the client 1-5 used when authenticating for the second session. The identity provider 1-4 could be for example an OpenID provider.

[0056] If the result of the authentication check is negative, the identity provider 1-4 might either send an authentication challenge request (not shown) directly to the client 1-5 or send a message (for example a push message, not shown) to the client 1-5 in order to force the client 1-5 to create a "virtual" second session and authenticate with the first operator 1-1. The performed authentication could then be used when repeating the authentication check for the received first request 1-12 as described above without challenging the client again. If the result of the authentication check is still negative, no further actions might be taken by the identity provider.

[0057] The identity provider 1-4 signals the result of the authentication check 1-15 back to the first server 1-8, which then in turn either accepts the request (if the check was successful indicating that the client has been already authenticated earlier successfully for the second session) and sends the requested data/information 1-16 towards the client 1-5 via the intermediate network element 1-6, or rejects the request if the authentication check with the identity provider 1-4 was not successful (not shown).

[0058] Turning now to fig 2 showing a second preferred embodiment of the present invention. Fig. 2 has a lot in common with the first embodiment shown in Fig 1 (for example the basic principle to pass a first identifier of the client together with the first request to the first server, in order to perform a check of the authentication status of the

client), therefore mainly the differences compared to fig 1 are outlined in the description of fig 2 below.

[0059] In Fig 2 the first request 2-20 is received by a second server 2-3 located preferably in the network of the first operator 2-1. The second server can be for example a SIP application server of the first operator network.

[0060] Similar to fig 1 a first request 2-20 to start a first session between the client 2-5 and a first server 2-8 might be originated by a remote client (not shown), where client 2-5 could be already in a call with, or by another network element (not shown). In contrast to fig. 1 the first identifier (and possible additional information like the address of the network element performing the authentication check) is added to the first request before the first request arrives at the client 2-5.

[0061] The first request 2-20 might include for example a HTTP URL encapsulated in a SIP REFER message. The HTTP URL should be further transmitted via an intermediate network element, for example a second server 2-3, and the client 2-5 towards the first server 2-8 which might then provides the requested HTTP content.

[0062] Before the second server 2-3 transmits the first request 2-10 to the client 2-5, the second server 2-3 adds a first identifier, indentifying the client, to the first request. The first identifier to be added might be retrieved 2-15 by the second server 2-3 from database 2-2 (the database includes information if the client 2-5 is authenticated earlier for a different session, like the second session mentioned in the prerequisite section above, which could be for example a SIP session), or a new first identifier might be generated for the client 2-5 by the second server 2-3. Generating a new first identifier can be done for example by using a pseudo-random number generator (generating a nonce), or any other client related information (for example an

OpenID identifier or a SIP address of the client) available in the intermediate network element and/or information retrieved from the database 2-2.

[0063] Like in the description related to fig. 1 the first identifier is added to the first request in order to be able to perform an authentication check 2-13/2-15 initiated by the first server 2-8 located at the second operator 2-7 network.

[0064] The first identifier is preferably a "nonce" identifier or an OpenID identifier, or a combination of both and might be stored in the database 2-2 optionally together with relationship information to at least one identifier for the same client 2-5 in the database 2-2. Further on same details as mentioned in the description to fig. 1, with respect to the at least one other identifier, apply also here.

[0065] The first identifier added by the second server 2-3 might also include an address of an identity provider that will be contacted by the first server 2-8 in order to perform the authentication check, for example an OpenID Provider 2-4 (for more details on this refer to the description of fig 1).

[0066] Sending of the first request with the first identifier from the second server 2-3 to the client 2-5 might be done via a secure connection (for example the first request and the first identifier are transferred via a secure SIPS session inside a SIP REFER message).

[0067] The client 2-5 receives the first request 2-10 with the first identifier from the second server 2-3, extracts it (for example if the requested is encapsulated in a SIP REFER message the client extracts the first request and the first identifier from the SIP REFER message) and issues the first request 2-12 including the first identifier to the first server 2-8.

[0068] The first request 2-12 including the first identifier could be for example a request for retrieving the content of a HTTP URL, the first server 2-8 could be for example a HTTP server.

[0069] Preferably the first request 2-12 is transferred over a secure connection (for example a HTTPS connection).

[0070] Like described for fig 1, the first server 2-8 receives the first request 2-12, extracts the first identifier, performs a authentication check 2-13/2-15 via the Identity Provider 2-4, and sends the requested data 2-16 towards the client 2-5 if the authentication check result 2-15 reveals that the client 2-5 was authenticated for a second session earlier. If the authentication check result 2-15 is negative, similar actions as described for fig 1 apply.

[0071] With the above described preferred embodiments in fig 1 and fig 2 the client is able to establish a first session with a first server without the need to explicitly authenticate for this session by reusing an earlier performed authentication of the client.

[0072] Figure 3 shows an example method for starting a first session with a first server. In a first step 3-1 of the method a first session between a client and a first server is started by generating a first request 3-1-1. The first request might be either generated at the client 3-1-1-1 or at another element 3-1-1-2 (for example a remote client or another network element). The first request might be a request including a HTTP URL to retrieve the content of a web page from a first server, and if the first request is generated at another element it might be encapsulated into another message, for example in a SIP REFER message.

[0073] The first request is received in a second step 3-2 of the method where a first identifier is added. The first identifier is identifying a client, which will be then later

on used to check if the client has been authenticated for another (second) session earlier. The first identifier might be obtained 3-2-2 by either retrieving it from a database 3-2-2-1, or by generating it, for example by using a pseudo-random number generator (generating a nonce), or any other client related information (for example an OpenID identifier or a SIP address of the client) available in the intermediate network element and/or information retrieved from the database 3-2-2-1. If a new first identifier is generated the identifier might be stored in the database 3-2-2-1 optionally together with relationship information to at least one other identifier for the same client . Further on same details as mentioned in the description to fig. 1, with respect to the at least one other identifier, apply also here.

[0074] Adding the first identifier to the first request happens either at a network element located between the client and the first server 3-2-1 (for example a HTTP proxy server) or at a second server 3-2-3 (for example an application server), which will then forward the first request, together with the first identifier, to the client. Optionally also an address of a network element, that will perform the authentication check later on (for example an identity provider), will be added to the first request. The address might be retrieved from the database 3-2-2-1 as well.

[0075] The forwarding of the first request from the second server to the client might happen for example via encapsulating the first request into a SIP REFER message. The client extracts then the first request from the SIP REFER message and forwards it to the first server 3-2-4.

[0076] Next the first request, including the first client identifier (and possibly the address of the network element which will perform an authentication check for the client based on the first identifier later on), will be transmitted 3-3 towards the first server.

[0077] The first server receives the first request 3-4, extracts the first identifier and the optional address from the received first request 3-5 and performs the authentication check of the client 3-6 utilizing the extracted first identifier. For performing the authentication check an identity provider is addressed, either via the extracted address from the received first request or by some other means (for example the identity provider address might be pre-programmed, informed beforehand or could be fetched from another node).

[0078] If the authentication check was successful (indicating that the client was already authenticated earlier) the requested data is send to the client 3-7. If the authentication check was not successful the identity provider might send a push message 3-8 to the client in order to "force" the client to authenticate for a second session first. This authentication will be then used in a repeated authentication check 3-6, which shall be then completed successfully.

[0079] Fig 4 shows an example block diagram of a network element 4-1 implementing the claimed invention of adding a first identifier to a first request for a first session started with a first server. Preferably such a network element could be a proxy server (for example a HTTP server) located between a client and the first server, or an application server (for example a SIP AS) located between the request originator and the client.

[0080] The network element 4-1 receives the first request 4-2-1 from external. The first request might be a HTTP request or a SIP message encapsulating a request to retrieve a HTTP URL from a first server. The SIP message might be a SIP REFER message. The first request might be received via a secure connection (for example HTTPS or SIPS).

[0081] In block 4-1-1 a first identifier related to the client involved in the first session is added. The first identifier might optionally also contain an address of a network element able to perform an authentication check utilizing the added first identifier (for example an identity provider).

[0082] The first identifier (and optionally the network element address) to be added might be obtained by a separate block 4-1-4. The first identifier (and optional the network element address) might be either retrieved 4-2-5 from a database 4-1-5 or a new first identifier might be generated 4-1-4 in the network element 4-1 (for example by using a pseudo-random number generator or the client IMPU), or any other available client related information). If a new first client identifier is generated it might be stored 4-2-7 in the database 4-1-5, optionally together with relationship information to at least one other identifier of the same client . The same details as mentioned in the description to fig. 1, with respect to the at least one other identifier, apply also here. The database 4-1-4 itself might be part of the network element 4-1 or it might be an external database. The database 4-1-4 might be also accessible by other (external) network elements (illustrated by 4-2-6).

[0083] After the first identifier, and optionally the address, has been added to the first request by block 4-1-1 the first request will be forwarded 4-2-2 to a transmitter block 4-1-3 which will then transmit the first request further on towards the first server 4-2-3. Preferably the first request, including the first identifier (and optionally the address as described above), will be send to the client, which will then forward it to the first server, or it will be sent to the first server directly. Transmission shall be done over a secure connection, however this is not mandatory. The connection 4-2-3 between the network element 4-1 and the receiving element (not shown) could be for example a HTTP(S) connection or a SIP(s) connection. In case of a SIP(S)

connection the first request, including the first identifier + optionally the address, might be transferred encapsulated into a SIP refer message.

[0084] Turning now to fig 5 and fig 6. Both figures show an example message sequence diagram for two preferred embodiments of the present invention. The figures are related to the embodiments illustrated in fig 1 and fig 2. Shown elements are basically the same and interaction between the elements is closely related. The message sequence diagram of fig. 5 relates to fig. 1, where the first identifier is added by an intermediate network element 5-6 (1-6) located between the client 5-1 (1-5) and the first server 5-7 (1-8), while fig 6 relates to fig 2 where the first identifier is added at a second server 6-3 (2-3) located between the origin of the first request and the client 6-1 (2-5).

[0085] In addition to fig 1 and fig 2, fig 5 and fig 6 illustrate also the "prerequisite", the authentication of the client with the network (or potentially any other service or application residing in the network) of the first operator performed for the second session (for example for a voice call session between the client 5-1/6-1 and another client). However the authentication process for the second session (5-8/6-8) is only shown in a simplified way and as a not limiting example for the case where the first network might be an IP Multimedia Subsystem.

[0086] The client 5-1/6-1 starts the authentication procedure by contacting (5-10/6-10) a P-CSCF via a SIP REGISTER message, which then forward the SIP message further via I-CSCF (if present) and S-CSCF (shown in a simplified way in fig 5 and fig 6 as *-CSCF 5-2/6-2 covering all three CSCF types). The S-CSCF performs the authentication with the client involving the HSS (not shown) which provides authentication related data.

[0087] After a successful authentication of the client 5-1/6-1 the *-CSCF 5-2/6-2 stores (5-11/6-11) the result of the authentication to a database 5-4/6-4. The stored information might include a second identifier (identifying the client) like for example a SIP identity of the client.

[0088] Turning now to fig 5. The client 5-1 issues a first request 5-20 (for example a HTTP request) targeted to the first server 5-7 for establishing a first session, in order to obtain for example data from the first server. The first server could be for example an application server located in a second operator network.

[0089] The first request 5-20 is received by an intermediate server 5-6 (for example a HTTP proxy server). The intermediate server 5-6 examines the request in order to be able to obtain and add a first identifier to the first request. The intermediate server 5-6 either generates the first identifier by itself (for example by using a pseudo-random number generator), or any other client related information available in the second server element and/or information retrieved from the database 5-4) or retrieves 5-21/5-22 a first identifier from the database 5-4. If the intermediate server 5-6 generates the first identifier by itself, the newly generated first identifier might be stored 5-21 in the database 5-4 optionally together with relationship information to at least one other identifier of the same client which might be already stored in the database (for example to the second identifier (e.g. the SIP identity) of the client stored from the authentication related to the second session as outlined above). Further on same details as mentioned in the description to fig. 1, with respect to the at least one other identifier, apply also here. The first identifier might be for example an OpenID identifier, a nonce identifier or a combination of both.

[0090] Optionally the intermediate server 5-6 also request with the message 5-21 the address of the identity provider 5-

5 from the database 5-4, which is then later on used by the first server to perform the authentication check of the client.

[0091] After the intermediate server 5-6 has obtained the first identifier related to the client, and optionally also the address of the identity provider 5-5, the intermediate server 5-6 adds 5-23 them to the first request and send the first request 5-24 (now including the first identifier and the address) to the first server 5-7. The transmission towards the first server could optionally happen over a secure session, for example a HTTPS session.

[0092] The first server 5-7 receives the first request 5-24 and extracts 5-27 the first identifier and optionally also the address of the identify provider. If the address of the identity provider is not included in the received first request 5-24 the first server 5-7 was either informed about it beforehand (received it earlier or it was pre-programmed) or the first server must request the address from another external network element (not shown).

[0093] The first server 5-7 send then an authentication check request 5-28 including the first identifier to the identity provider 5-5. The Identity Provider could be in a preferred embodiment of the invention an Open ID Provider.

[0094] The identity provider 5-5 extracts the first identifier from the received authentication check request 5-28 and performs a check 5-29 with the database 5-4 in order to check the authentication status of the client (for example to check if client has been authenticated earlier like for the second session as described above). The result of the check is transmitted 5-30 from the identity provider 5—5 back to the first server 5-7.

[0095] If the result of the authentication check is positive 5-31, the first server 5-7 regards the client as

authenticated and sends 5-32 the data requested by the client in the first request. The data is send via the intermediate server 5-6 towards the client 5-1.

[0096] If the result from the authentication check 5-30 is negative, similar actions are performed as described already for fig 1 and fig 2. Actions in case of a negative result from the authentication check 5-30 are not shown in fig 5.

[0097] In Fig 6 the first request 6-22 is received by a second server 6-3, which is preferably located in the network of the first operator. The second server can be for example an application sever, preferably a SIP server. The received first request might include a HTTP URL for retrieving the content of a web page encapsulated in a SIP REFER message, targeted to establish a first session between a client 6-1 and a first server 6-7. The first request might be generated by a remote client, where for example the client 6-1 is already in call with, or by some other network element (not shown). Before forwarding the first request to the client 6-1, the second server will add a first identifier, identifying the client 6-1, to the first request, so that the first server is able to perform an authentication check for the client 6-1. Optionally the second server may also add an address of an Identity Provider 6-5.

[0098] Before the second server 6-3 can add the first identifier it starts an obtaining process in order to either retrieve the identifier 6-23 from a database 6-4 (possibly together with the address of the identity provider 6-5) or to generate the first identifier by itself (for example by using a pseudo-random number generator, the client's IMPU or other identity (for example "http://csp.com/sip:bob123@csp.com" , where „URI" part is a SIP URI) or any other client related information available in the second server and/or information retrieved from the database 6-4). If the second server 6-3 generates the first identifier by itself, the resulting generated first identifier might be stored in the database 6-

4 optionally together with relationship information to for example at least one other identifier of the same client 6-1 stored already in the database 6-4 (for example the second identifier (the SIP identity) of the client 6-1 stored in the database with respect to the already authentication performed for the second session as described earlier). Further on same details as mentioned in the description to fig. 1, with respect to the at least one other identifier, apply also here.

[0099] The second server adds then the first identifier (and optionally the address of the identity provider 6-5) to the first request 6-24 and transmits the first request towards the client 6-1. The transmission 6-25 is preferably done via a secure session, for example a SIPS session, where the first request includes for example a HTTP URL together with the first identifier and optionally the address. The first request might be encapsulated in a SIP REFER message.

[00100] The client receives the first request 6-25, extracts it (for example from the SIP REFER message) and forwards 6-26 it to the first server 6-7. The forwarding of the request from the client 6-1 to the first server 6-7 is preferable done via a secure connection, preferably via HTTPS.

[00101] The first server 6-7 receives the first request 6-26 and starts to perform actions 6-27 to 6-31, which are similar to actions 5-27 to 5-31 already described in relationship to fig 5. Therefore those steps are not described here in detail again (refer to the corresponding description passage for fig 5).

[00102] If the result of the authentication check is positive 6-31 the first server will send the requested data 6-33 directly to the client 6-1. If the authentication check is negative, similar actions as for fig 5 and figures 1 and 2 will be performed.

[00103] Figs 1 to Fig 6 include the option to add an address, in addition to the identifier, to the first request. The purpose for adding the address is to inform the first server (for example the application server located in the network of the second operator) about the address of the network element that is able to perform the authentication check, for example the identity provider address. By providing the identity provider address together with the first identifier, the first server is not required to retrieve the address by itself. The identity provider address could be a part of the first identifier or added as a separate element to the first request.

[00104] Alternatively to adding the address together with the identifier to the first request, the network element that adds the first identifier (for example the second server 2-20/6-3 or the intermediate node 1-6/5-6) could send the address in parallel to the first request to the first server 1-8/2-8/5-7/6-7. This could happen for example simultaneously, or shortly after/before, the first request, including the first identifier, is transmitted. Sending the address to the first server might happen on the direct way, so possible intermediate nodes, where the first request is passed through, might be bypassed by the message including the address.

[00105] Sending the address separately to the first server is especially relevant for the preferred embodiments shown in fig 1 and fig 5 (where the first request is received at an intermediate network element, for example a HTTP proxy server, located between the client and the first server). Note, the above described alternative of transmitting the address separately is not shown in the figures.

[00106] For all the described embodiments, and all other embodiments where the invention could be used and which are not described in this application, the first identifier, identifying the client, might identical for the first and the

second session (identical to the second identifier). However, in the majority of the cases the first and second identifier will be not identical.

[00107] Even if in the description of the preferred embodiments the first request is quite often mentioned to be a HTTP request, a request including a HTTP URL or a request encapsulated into a SIP message like a SIP refer message, it shall be understood that all these are just examples of a first request and do not limit the first request to any specific protocol type or protocol command. Further on, in a preferred embodiment of the invention the first request is requesting data "for the first time" from the first server. However, the invention can be also used in situations where the first request is not requesting data "for the first time" from the first server.

[00108] Functions (or at least parts of the functions) for starting a first session between a client and a first server (where the client has not been authenticated for the first session and where a request is requesting data related to the first session from the first server), adding a first identifier (identifying the client) to the first request, transmitting the first request with the first identifier to the first server, wherein the first identifier is related to a second identifier identifying the client and wherein the second identifier was used for authenticating the client for a second session, as described above may be implemented by code means, as software, and loaded into memory of a computer. Different computer instances might be used for different parts of the functions.

[00109] Time aspects included in Figs. 1 to 6 do not restrict any one of the shown steps to be limited to the step sequence as outlined. This applies in particular to method steps that may be functionally disjunctive with each other.

[00110] Although the present invention has been described herein before with reference to particular embodiments thereof, the present invention is not limited thereto and various modification can be made thereto.

[00111] For example the described invention can be also adapted to other authentication solutions than OpenID like for example Liberty/SAML or Windows CardSpace.

Used abbreviations:

AS	Application Server
CSCF	Call Session Control Function
*-CSCF	Placeholder for different CSCF types
P-CSCF	Proxy Call Session Control Function
I-CSCF	Interrogating Call Session Control Function
ID	Identifier
S-CSCF	Serving Call Session Control Function
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
HSS	Home Subscriber Service
IMPU	IP Multimedia Public Identity
IMS	Internet protocol Multimedia Subsystem
OP	OpenID Provider
RP	Relying Party (with respect to an OP)
UE	User Equipment
URI	Uniform Resource Identifier
URL	Uniform Resource Allocator
SAML	Security Assertion Markup Language
SIP	Session Initiation Protocol
SIPS	Session Initiation Protocol Secure
SP	Service Provider

CLAIMS:

1. A method being part of an authentication procedure of a client when a first session (5-9) between the client (5-1) and a first server (5-7) is started, where the client has not been authenticated for the first session and where a first request (5-20) is requesting data related to the first session from the first server, characterised by
obtaining (4-1-4) a first identifier, identifying the client,
adding (5-23) the first identifier to the first request,
transmitting (5-24) the first request with the first identifier to the first server,
wherein the first identifier is related to a second identifier identifying the client and wherein the second identifier was used for authenticating the client for a second session (5-8).

2. A method of claim 1 wherein obtaining the first identifier is done by generating (4-1-4) the first identifier or by retrieving (4-2-5) the first identifier from a database (4-1-5).

3. A method of claim 2 wherein the generated first identifier is stored (4-2-7) in the database (4-1-5).

4. A method of claim 2 wherein if the first identifier is generated, also relationship information between the first identifier and the second identifier is generated, and the generated first identifier together with the relationship information is stored (4-2-7) in the database (4-1-5).

5. A method of any of the preceding claims wherein the first request (5-20) is requesting for the first time data related to the first session from the first server (5-7).

6. A method of any of the preceding claims wherein at least one part of the first request (5-20) is requesting data or

information from the first server (5-7) which is not related to the authentication of the client.

7. A method of any of the preceding claims where the first identifier is added (5-23) to the first request by a network element (5-6) located between the client (5-1) and the first server (5-7).

8. A method of claim 7 wherein the network element is a proxy server (5-6).

9. A method of claim 8 wherein the proxy server is a HTTP proxy server (5-6).

10. A method of any of the claims 1 to 6 wherein the first identifier is added (6-24) to the first request (6-22) by a second server (6-3) and where the second server sends (6-25) the first request, including the first identifier, to the client (6-1) which then forwards (6-34) the request, including the first identifier, to the first server (6-7).

11. A method of claim 10 where the connection (6-25) between the second server and the client is a secure connection.

12. A method of claim 10 or 11 wherein the second server is an Application Server of a first Operator (6-3).

13. A method of any of the claims 10 to 12 wherein the first request and the first identifier are included in a SIP message (6-25).

14. A method of claim 13 wherein the SIP message is a SIP REFER message (6-25).

15. A method of any of claims 12 to 14 wherein the first Operator is an IMS Operator (1-1).

16. A method of any of the preceding claims wherein the first identifier (3-2) is a nonce identifier.

17. A method of any of claims 1 to 15 wherein the first identifier (3-2) is an OpenID identifier.

18. A method of any of the preceding claims wherein the first identifier (3-2) includes address information of an Identity Provider.

19. A method of any of the preceding claims wherein the first session is a HTTP session (5-9), the first request is a HTTP request (5-20) and the first server is a HTTP server (5-6).

20. A method of any of the preceding claims wherein the authentication of the client for the first session is assumed to be successfully completed if the client was already authenticated successfully for the second session (3-7).

21. A device (4-6) being part of an authentication procedure of a client when a first session (5-9) between the client (5-1) and a first server (5-7) is started, where the client has not been authenticated for the first session and where a first request is requesting data related to the first session from the first server, the device comprising:

- an obtaining means (4-1-4) obtaining a first identifier identifying a client,
- an adding means (4-1-1) for adding the first identifier to the first request,
- a transmitting means (4-1-3) for transmitting the first request with the first identifier towards the first server, wherein the first identifier is related to a second identifier identifying the client and where the second identifier was used for authenticating the client for a second session (3-6).

22. A device of claim 21 wherein the obtaining means is configured to generate (4-1-4) the first identifier or to retrieve (4-2-5) the first identifier from a database (4-1-5).

23. A device of claim 22 wherein the obtaining means is configured to store (4-2-7) the generated first identifier in the database (4-1-5).

24. A device of claim 23 wherein the obtaining means is configured, if the first identifier is generated, to generate relationship information between the first identifier and the second identifier and to store (4-2-7) the generated first identifier together with the relationship information in the database (4-1-5).

25. A device of any of the claims 21 to 24 wherein the first request (5-20) is requesting for the first time data related to the first session from the first server (5-7).

26. A device of any of the claims 21 to 25 wherein at least one part of the first request (1-11) is requesting data or information from the first server (1-8) which is not related to the authentication of the client (1-5).

27. A device of any of the claims 21 to 26 wherein the device is located between the client (1-5) and the first server (1-8).

28. A device of claim 27 where the device is a proxy server (1-6).

29. A method of claim 28 wherein the proxy server is a HTTP proxy server (1-6).

30. A device of any of the claims 21 to 26 wherein the device is a second server (2-3) and where the second

server sends (2-10) the first request, including the first identifier, to the client (2-5) which then forwards the request, including the first identifier, to the first server (2-8).

31. A device of claim 30 where the connection (2-10) between the second server and the client is a secure connection.

32. A device of claim 30 or 31 wherein the second server is an Application Server (2-3) of a first Operator (2-1).

33. A device of any of the claims 30 to 32 wherein the first request and the first identifier are included in a SIP message (6-25).

34. A device of claim 33 wherein the SIP message is a SIP refer message (6-25).

35. A device of any of the claims 32 to 34 wherein the first Operator is an IMS operator (1-1).

36. A device of any of claims 21 to 35 wherein the first identifier (3-2) is a nonce identifier.

37. A device of any of claims 21 to 35 wherein the first identifier (3-2) is an OpenID identifier.

38. A device of any of claims 21 to 37 wherein the first identifier (3-2) includes address information of an Identity Provider.

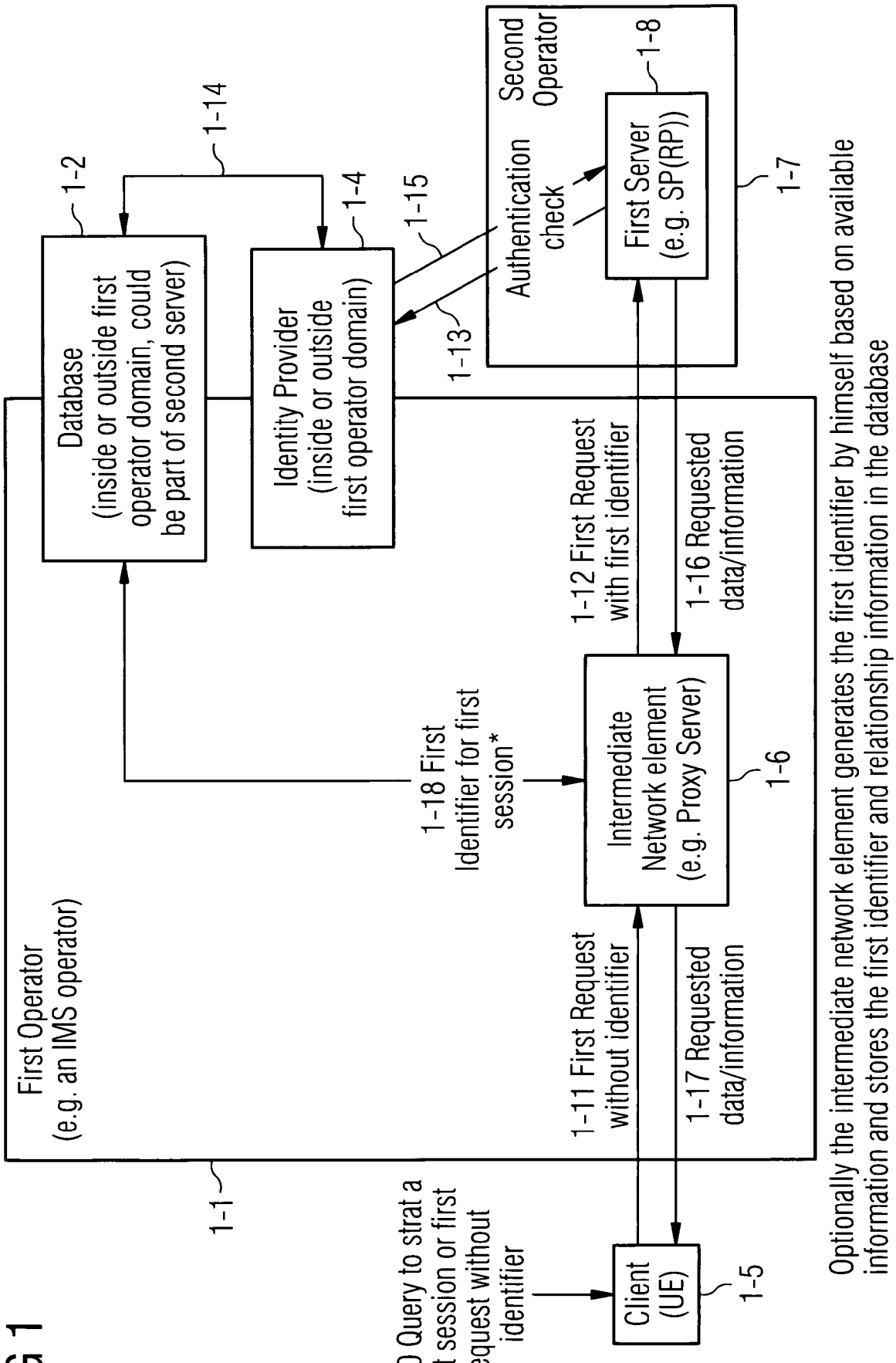
39. A device of any of the claims 21 to 38 wherein the first session is a HTTP session (5-9), the first request is a HTTP request (5-20) and the first server is a HTTP server (5-6).

40. A device of any of the claims 21 to 39 wherein the authentication of the client for the first session is

assumed to be successfully completed if the client was already authenticated successfully for the second session (3-7).

41. A computer program product comprising code means for starting a first session (3-1) between a client (1-5) and a first server (1-8), where the client has not been authenticated for the first session, and where a first request is requesting for the first time data related to the first session from the first server (3-1-1), adapted to produce the steps of any one of the claims 1 to 20 when loaded into the memory of a computer.

FIG 1


First Operator
(e.g. an IMS operator)
1-1
Database
(inside or outside first operator domain, could be part of second server)
1-2
Identity Provider
(inside or outside first operator domain)
1-4
1-14
1-15
1-13
Second Operator
Authentication check
First Server
(e.g. SP(RP))
1-8
1-7
1-12 First Request with first identifier
1-16 Requested data/information
1-18 First Identifier for first session*
Intermediate Network element
(e.g. Proxy Server)
1-6
1-11 First Request without identifier
1-17 Requested data/information
1-10 Query to strat a first session or first request without identifier
Client
(UE)
1-5
Optionally the intermediate network element generates the first identifier by himself based on available information and stores the first identifier and relationship information in the database

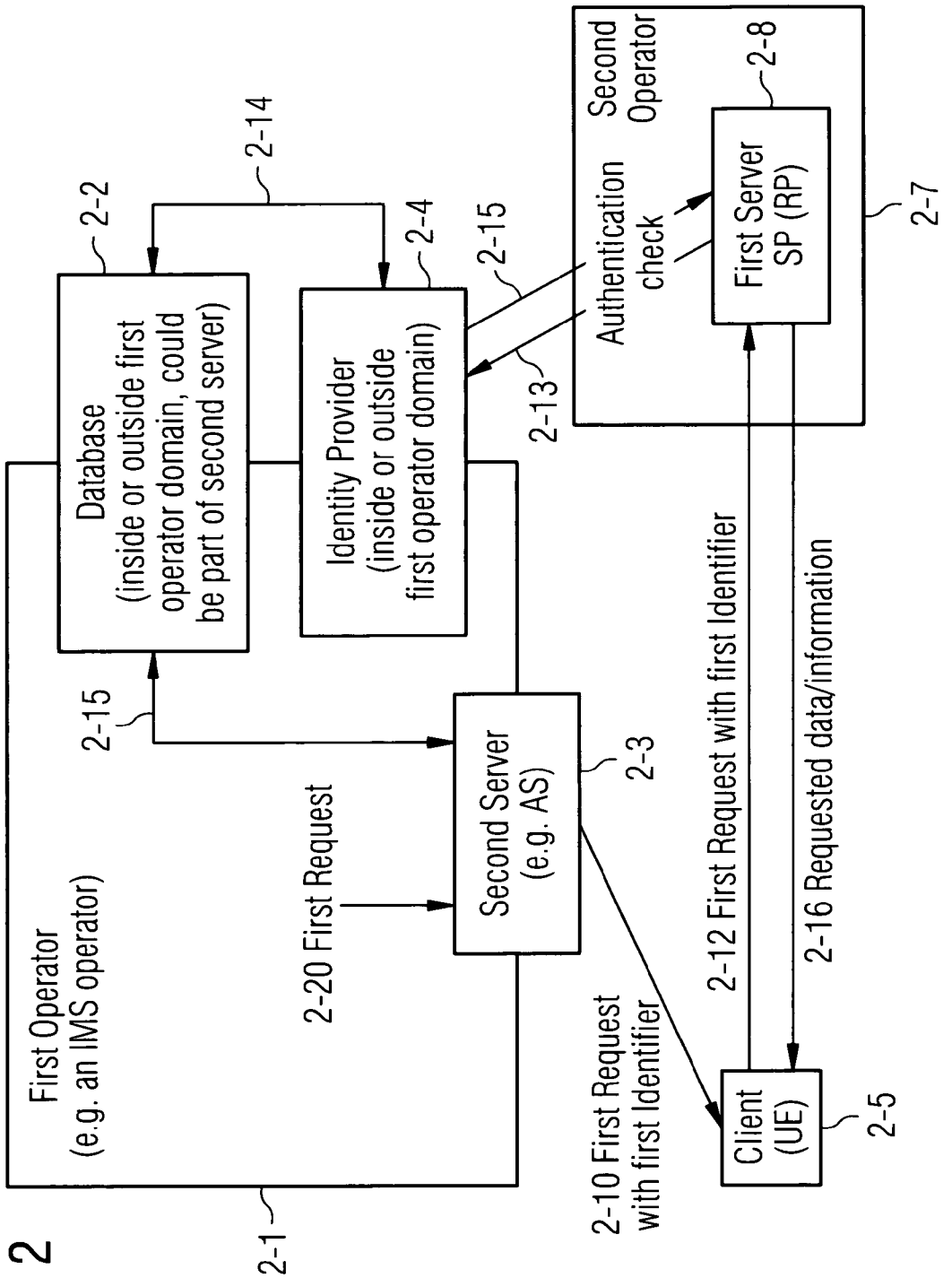
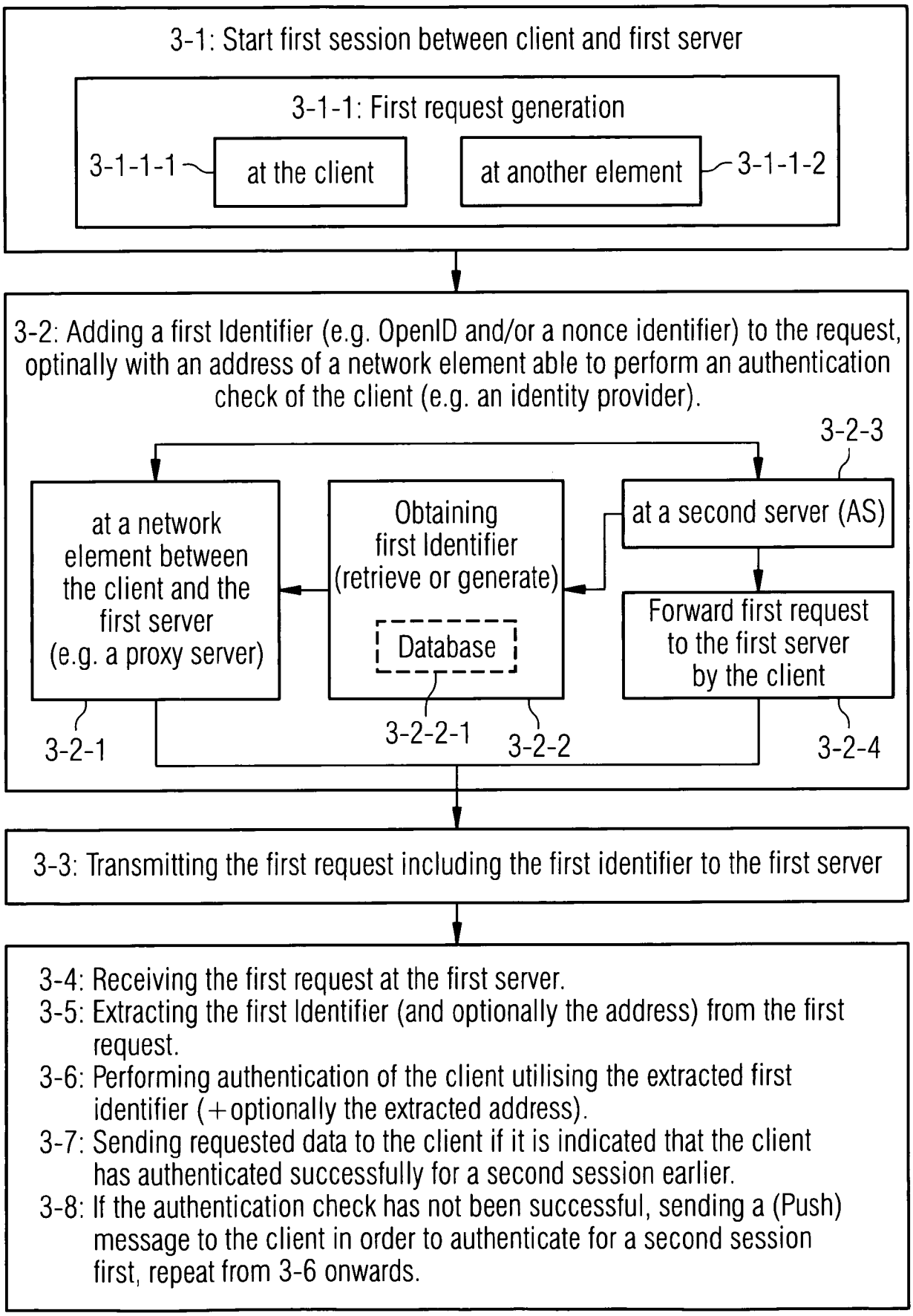

First Operator
(e.g. an IMS operator)
2-1
Database
(inside or outside first
operator domain, could
be part of second server)
2-2
Identity Provider
(inside or outside
first operator domain)
2-4
2-14
2-15
2-13
Second Operator
Authentication
check
First Server
SP (RP)
2-8
2-7
2-15
Second Server
(e.g. AS)
2-3
2-20 First Request
2-10 First Request
with first Identifier
2-12 First Request with first Identifier
2-16 Requested data/information
Client
(UE)
2-5

FIG 2

FIG 3


3-1: Start first session between client and first server
3-1-1: First request generation
3-1-1-1
at the client
at another element
3-1-1-2
3-2: Adding a first Identifier (e.g. OpenID and/or a nonce identifier) to the request, optinally with an address of a network element able to perform an authentication check of the client (e.g. an identity provider).
3-2-3
at a network element between the client and the first server (e.g. a proxy server)
Obtaining first Identifier (retrieve or generate)
Database
at a second server (AS)
Forward first request to the first server by the client
3-2-1
3-2-2-1
3-2-2
3-2-4
3-3: Transmitting the first request including the first identifier to the first server
3-4: Receiving the first request at the first server.
3-5: Extracting the first Identifier (and optionally the address) from the first request.
3-6: Performing authentication of the client utilising the extracted first identifier (+optionally the extracted address).
3-7: Sending requested data to the client if it is indicated that the client has authenticated successfully for a second session earlier.
3-8: If the authentication check has not been successful, sending a (Push) message to the client in order to authenticate for a second session first, repeat from 3-6 onwards.

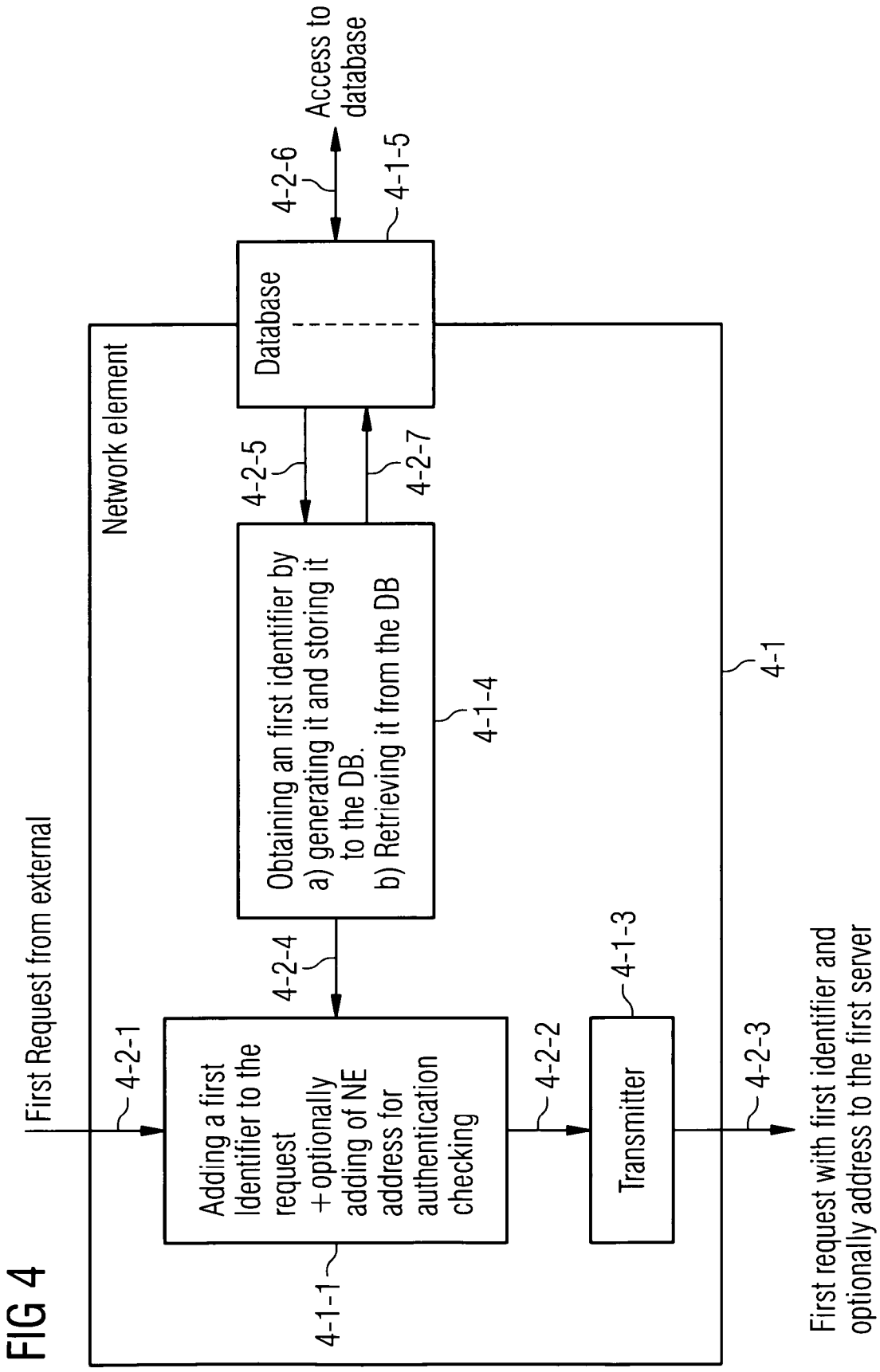

Access to database
4-2-6
4-1-5
Database
Network element
4-2-5
4-2-7
Obtaining an first identifier by
a) generating it and storing it to the DB.
b) Retrieving it from the DB
4-1-4
4-1
First Request from external
4-2-1
4-2-4
4-1-3
Adding a first Identifier to the request +optionally adding of NE address for authentication checking
4-2-2
Transmitter
4-2-3
4-1-1
First request with first identifier and optionally address to the first server

FIG 4

FIG 5

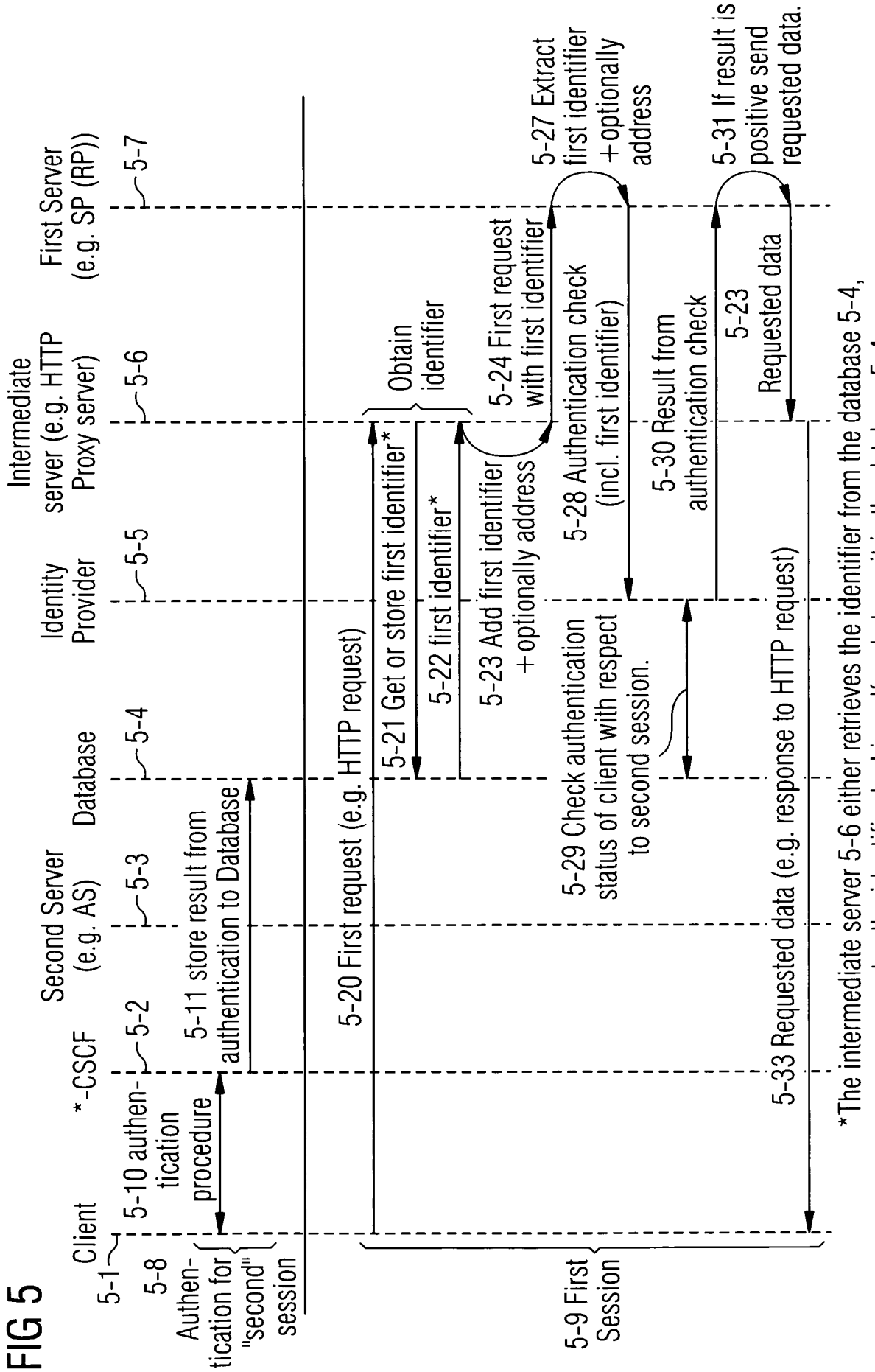
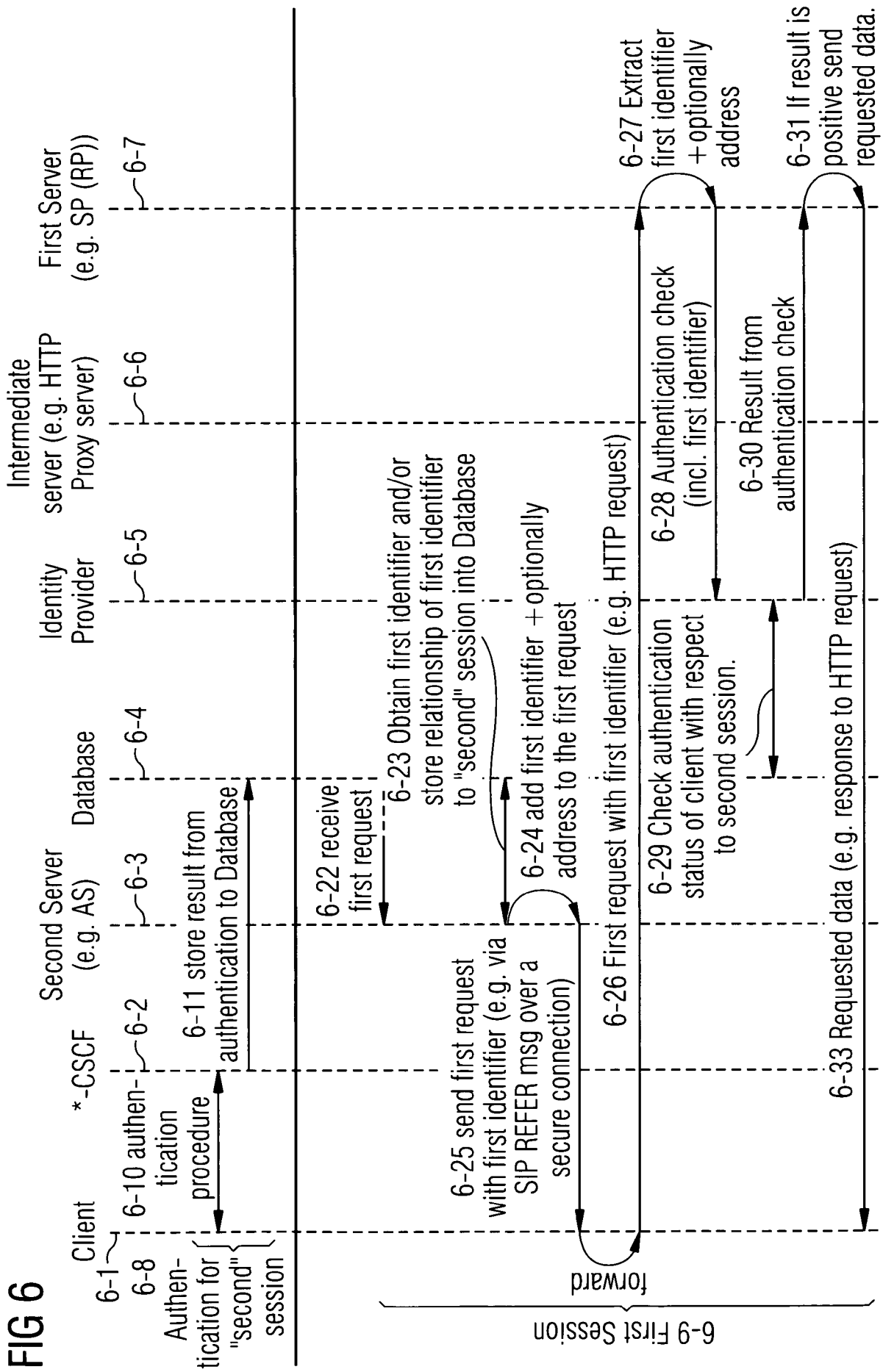

Client
5-1
*-CSCF
5-2
Second Server (e.g. AS)
5-3
Database
5-4
Identity Provider
5-5
Intermediate server (e.g. HTTP Proxy server)
5-6
First Server (e.g. SP (RP))
5-7
5-8 Authentication for "second" session
5-10 authentication procedure
5-11 store result from authentication to Database
5-9 First Session
5-20 First request (e.g. HTTP request)
5-21 Get or store first identifier*
Obtain identifier
5-22 first identifier*
5-23 Add first identifier + optionally address
5-24 First request with first identifier
5-27 Extract first identifier + optionally address
5-28 Authentication check (incl. first identifier)
5-29 Check authentication status of client with respect to second session.
5-30 Result from authentication check
5-31 If result is positive send requested data.
5-23 Requested data
5-33 Requested data (e.g. response to HTTP request)
*The intermediate server 5-6 either retrieves the identifier from the database 5-4, or generates the identifier by himself and stores it in the database 5-4

FIG 6


Client
6-1
6-8 Authentication for "second" session
*-CSCF
6-2
Second Server (e.g. AS)
6-3
Database
6-4
Identity Provider
6-5
Intermediate server (e.g. HTTP Proxy server)
6-6
First Server (e.g. SP (RP))
6-7
6-10 authentication procedure
6-11 store result from authentication to Database
6-22 receive first request
6-23 Obtain first identifier and/or store relationship of first identifier to "second" session into Database
6-24 add first identifier + optionally address to the first request
6-25 send first request with first identifier (e.g. via SIP REFER msg over a secure connection)
6-26 First request with first identifier (e.g. HTTP request)
6-27 Extract first identifier + optionally address
6-28 Authentication check (incl. first identifier)
6-29 Check authentication status of client with respect to second session.
6-30 Result from authentication check
6-31 If result is positive send requested data.
6-33 Requested data (e.g. response to HTTP request)
forward
6-9 First Session

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2009/063971

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L29/06
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04L

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

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, INSPEC, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2008/120705 A1 (BEYER LORAINE [US] ET AL) 22 May 2008 (2008-05-22) * abstract paragraph [0025] - paragraph [0026] paragraph [0032] - paragraph [0044] figures 1-8	1-41

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

☒ See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
30 December 2010	11/01/2011

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Dujardin, Corinne

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2009/063971

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2008120705 A1	22-05-2008	NONE	