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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
01 October 2020 (01.10.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/193555 A1

(51) International Patent Classification:
G06F 16/958 (2019.01) *G06F 21/41* (2013.01)
H04L 29/12 (2006.01) *H04L 29/06* (2006.01)
H04L 29/08 (2006.01)

(21) International Application Number:
PCT/EP2020/058165

(22) International Filing Date:
24 March 2020 (24.03.2020)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
19164986.2 25 March 2019 (25.03.2019) EP

(71) Applicant: SIEMENS AKTIENGESELLSCHAFT [DE/DE]; Werner-von-Siemens-Straße 1, 80333 München (DE).

(72) Inventor: RUSSO, Francesco; Via G. Leopardi 3/28, 16011 Arenzano (GE) (IT).

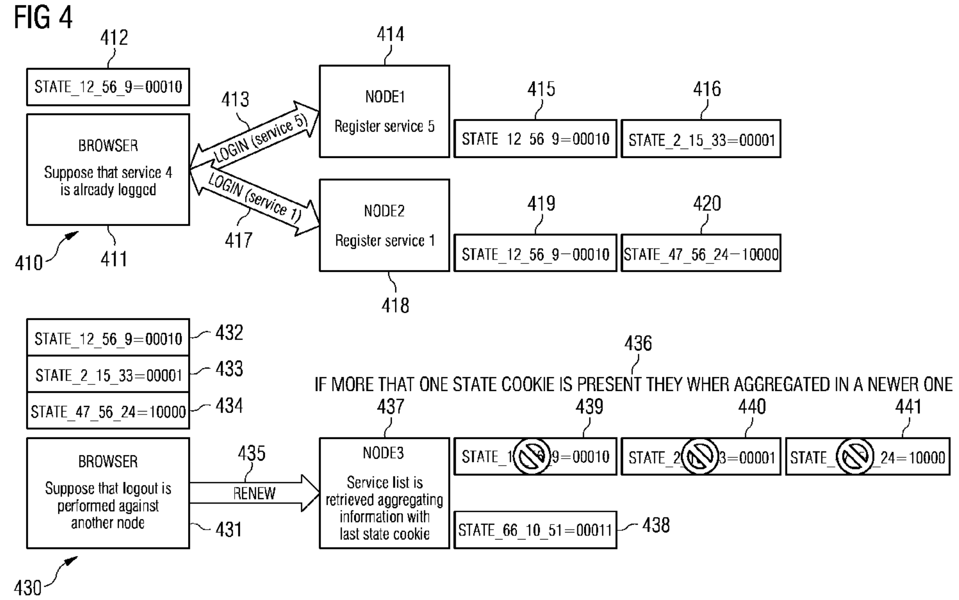
(74) Agent: MAIER, Daniel; Postfach 22 16 34, 80506 München (DE).

(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, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

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

(54) Title: MANAGING A SSO SESSION BY AN IDENTITY PROVIDER

FIG 4



(57) Abstract: The present invention provides a method, a system and a computer program for managing a SSO session by an identity provider for a plurality of services, comprising managing, by an identity provider, information on the SSO session via a cookie based protocol;persisting a list of services of relying parties participating in a same SSO session information in one session cookie and a plurality of temporary state cookies with randomly generated names, whereby the list of session services are represented with a bit mask representation within the cookies; and, whereby the plurality of temporary state cookies can be consolidated into one state cookie.

[Continued on next page]

TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW,
KM, ML, MR, NE, SN, TD, TG).

Published:

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

Managing a SSO session by an identity provider

The invention relates to a method, a system and a computer program product for managing a SSO session by an identity provider for a plurality of services.

Conveniently, embodiments can also be applied to the field of industrial software and in particularly to the field of MOM.

Most recently, the term MOM (Manufacturing Operations Management) is more and more used to replace the term MES (Manufacturing Executing System).

In the field of industrial software, in order to integrate the login between different web applications which may also be delivered by different products, identity providers are used with web Single Sign On (SSO) functionalities.

Applications, known as relying parties, can ask for a login session to the identity provider and - if it exists - they can then join a global session shared between the different relying parties. When the state of the global session changes or terminates, all the relying parties receive a notification.

Relying parties are typically web applications or services in the cloud wanting to join a web SSO session and they receive authentication information from the identity provider in a claim.

In order to increase availability, identity providers are often implemented in a cluster of nodes.

In the art, the information to manage and restore the SSO session is typically stored at the server side, for example a server side session managed by the web server or persisted to an external system (e.g. SQL, distributed cache) in case of redundancy.

In such scenarios, there is the need to increase infrastructure for redundancy purposes with the related disadvantages of cost increases in terms of extra hardware and maintenance complexities.

Therefore, techniques of web SSO session management in which the session state is persisted at the client side are desirable choices.

Unfortunately, in the art, such client based techniques may experience one of more of the below challenging technical issues:
- being able of reconstructing the service list when login requests are performed to different nodes;
- dealing with a browser response order which is different from that of the http server;
- be limited by a cookie size which cannot exceed certain thresholds.

Therefore improved techniques are desirable.

The aforementioned aim is achieved by a method, a system and a computer program product for managing a SSO session by an identity provider for a plurality of services where SSO session information is persisted at the client side via a cookie-based protocol.

The aforementioned aim is achieved by a method, a system and a computer program product for managing a SSO session by an identity provider for a plurality of services including:
a) by an identity provider, managing information on the SSO session via a cookie based protocol;
b) persisting a list of services of relying parties participating in a same SSO session information in one session cookie and a plurality of temporary state cookies with randomly generated names, whereby the list of session services are represented with a bit mask representation within the cookies; and, whereby the plurality of temporary state cookies can be consolidated into one state cookie.

In embodiments, the bit mask may be advantageously represented in a compressed format.

In embodiments, the SSO session state may conveniently be saved via cookies and can be recovered via cookies and there is no need of persistency of the SSO session state at the server side.

In embodiments, the session static information may preferably be represented as a signed ticket whereby both ticket and signature are stored in separate cookies.

In embodiments, when a new relying party joins a global SSO session, the session cookie may conveniently be updated by adding the new relying party into the list of the global session participants.

In embodiments, the information on the session state is stored on the client side at the browser.

In embodiments, needed session information is stored via cookies.

In embodiment, the list of relying party of the SSO session is persisted in one or more state cookies.

In embodiments, state cookies have randomly generated names.

In embodiments, session services are listed with a bit mask representation.

In embodiments, a plurality of state cookies are consolidated in one consolidated state cookie and temporary state cookies are deleted.

In embodiments, the bit mask may preferably be compressed.

Furthermore, a computer program element can be provided, comprising computer program code for performing steps according to the above mentioned method when loaded in a digital processor of a computing device.

Additionally, a computer program product stored on a computer usable medium can be provided, comprising computer readable program code for causing a computing device to perform the mentioned method.

In embodiments, a SSO protocol implementation uses a client side storage of the session state.

In embodiments, once a new session is created all the needed session state information are stored via cookies and sent back to the browser.

In embodiments, the needed session information is stored in the memory-cache of one node and, in case it is not present, it is possible to recover the session information from the session and state cookies.

In embodiments, a SSO protocol implementation conveniently persists the list of services of the relying parties that are participating in the same authenticated session in one or more cookies, so that the identity provider can dispatch session notifications to each relying party of the session.

In embodiments, if a valid ticket is received by the node, the web SSO session is reconstructed.

Advantageously, there is no need of having centralized session information stored on the server side in order to verify if the session is still valid and not invalidated.

When a new relying party joins the global session, the session state is updated and this new relying party is added in the list of the global session participants. This list contains the end points registered in order to receive notifications. Examples of session requests include but are not limited to authentication, renew and logout requests.

In embodiments, the service must be registered to the whitelist.

In embodiments, the service is preferably represented within the service list within a "state" cookie with one bit in a binary mask. Hence, cookie size problems are advantageously minimized. For example, Set-Cookie: STATE=00010 the forth bit ON refers to the forth service.

In embodiments, the representation via a bit mask may be compressed by various compression techniques.

In embodiments, a base-32 encoding may conveniently be used. Hence, the size is advantageously reduced by five times so as to mitigate size issues of the http heater due to the binary string representation.

In embodiments, a substring composed by all zeros may conveniently be replaced with their count. Hence, the size may also be advantageously reduced.

In embodiments, a state cookie with a randomly generated name is used to enable to persist the session state with cookies. Hence, any loss of bits due to the unpredictability of the browser response is advantageously avoided so that a cookie defined first by the server and set after in the browser does not overwrite the previous value.

With embodiments, it is provided a client side session management for web SSO identity provider with a cookie state protocol.

In embodiment, session static information (that do not change over time) are represented as a signed ticket: both ticket and signature (encoded base64) are stored in separate cookies.

In embodiments, information on the session is persisted with one session cookie and a plurality of temporary state cookies, that may then be conveniently reduced/consolidated to one state cookie.

With embodiments, the session state is saved via cookies and can be recovered via cookies.

With embodiments, there is no persistence of the session state at the server side.

With embodiments, when requesting a single sign off functionality on multiple cluster nodes there is no need of saving the state on a physical storage or in-memory cache.

With embodiments, the cookie state protocol in the Web SSO identity provider gives the possibility to persist the session state at the client side. Advantageously, it is not necessary to have a specific server infrastructure in order to manage the session.

With embodiments, installation, license and maintenance costs are reduced since in case of redundancy it is necessary to buy specific software or use the OSS equivalent.

With embodiments, scalability is improved since it is enough to add a new server in the load balancer.

The invention will now be described in preferred but not exclusive embodiments with reference to the accompanying drawings Figure 1 to Figure 4.

Figure 1, Figure 2 and Figure 3 are diagrams of examples of information exchange flow between the browser, the identity provider nodes and the relying parties according to exemplary embodiments of the invention.

Examples of relying parties may include, but are not limited to, an application SCADA, a MOM system, MES Simatic IT and/or TIA Portal.

Figure 1 is a diagram of a session creation in accordance with embodiments.

In Figure 1, a new session first session is created for a first relying party RyP_1 101 and then a second relying party RyP_2 102 joins this web SSO session. When service 1 is registered and joins the session, the first status cookie is created 103 "STATUS_123456=10000", the first bit of the mask is ON to indicate that service 1 (the end point of relying party RyP_1) is in the session. A second status cookie "STATUS_234567=01000" (with a different random name) is created 104 when service 2 (the end point of relying party RyP_2) joins this same session. It is noted that in this simple example, the first status cookie is created for service 1 and the second status cookie is created for service 2, the skilled in the art easily understand that other options/embodiments are possible, for example the first status cookie may instead be created for service 5 (fifth bit of the mask) for relying party 5 not shown.

Figure 2 is a diagram of an example of cookie consolidation in accordance with embodiments.

In Figure 2, the first two status cookies are consolidated in one status cookie STATUS_456789=11000 where the first two bits of the bit mask are ON for relying party 1 and 2. The previous status cookies are deleted by using expiration attribute 231.

With embodiments, by consolidating the state cookies into one state cookie, size requirements are advantageously reduced. In embodiments, when there is an authentication request and more than one state cookies are received, there is a consolidation into one single state cookie. In other embodiments, other scenario may trigger cookie consolidation.

Figure 3 is a diagram of a parallel join request (silent login) in accordance with embodiments. In Figure 3, two parallel silent logins are sent from the browser to the identity provider: in the example they are routed to two different nodes. The status cookies are always added and no relaying party registration is lost.

Figure 4 is a diagram of generation of state cookies according to an exemplary embodiment.

In the upper part 410 of Figure 4, assume that service 4 is already logged, a browser requests a parallel login of two services service 5 and services 1 which are sent to two nodes Node 1 and Node 2 respectively by a load balancer (not shown). New state cookies are created for service 5 with the fifth bit ON and for service 1 with the first bit ON.

In the lower part 430 of Figure 4, assume that logout is performed against another node 434, if more than one state cookie is present they are aggregated 437 in newer one, with the aggregated bit mask first, fourth and fifth bits are ON for Service 1, 4 and 5.

For example: Set-Cookie: STATE_12_56_9=00010
If simultaneous login requests are performed to different node, each of the following/successive requests sent by the browser contains all the state cookies. Advantageously, each node can gather the overall information on the service list.

In embodiments, when a new gathered cookie is added, older cookies will be removed.

In embodiment, the state cookie has a name with an initial prefix which is common (e.g. STATE) and a random part (e.g. 456789, 567890, 789012).

In embodiments, requests arriving from different nodes are sent by the browser to the http server that responds with cookie “set”.

Table 1 below is an example embodiment of a session cookie format.

Session Cookie format.

ticket_id → *this cookie contains all the information in order to reconstruct a session information (e.g. including identity expiration, …)*

Format: <header>.<claim>.<signature>
This fields are base64 format and url encoded.

header → *fixed header*
{"alg":"RS256","kid":"F9AF369D12A12DEBE24922E99CFD2E15F67450E5","typ":"JWT"}

claim → *the session description*
{"exp":"2018-03-26T14:51:26.4719099Z","iat":"2018-03-26T14:41:26.4719099Z","iss":"UMC Identity Provider","sub":<innerclaim>}
Innerclaim → *session information*

```
{  "issuerCommonName": "Siemens Issuing CA EE Auth 2013",
  "subject": "",
  "computerTicket": "",
  "validity": "600",
  "sessionEnd": "2018-03-27T09:02:28Z",
  "authnmethod": "pwd",
  "subjectCommonName": "root",
  "securityLevel": 80,
  "ticketUsage": "",
  "subjectAlternativeName": ""}
```

signature → *signature of the previous fie*

Table 1

The skilled in the art easily appreciate that other various not disclosed but advantageous embodiments of the claimed invention are possible.

None of the description in the present application should be read as implying that any particular element, step, or function is an essential element which must be included in the claim scope: the scope of patented subject matter is defined only by the allowed claims.

Reference signs and text of the figures

Figure 1:

101 RyP_1 acronym for Relying Party 1
102 RyP_2 acronym for Relying Party 2
103 creation of first status cookie
110 Browser
120 IdP_1 acronym for Identity Provider 1
121 IdP_2 acronym for Identity Provider 2
130 Set-Cookie:
STATUS_123456=10000
131 Cookie:
STATUS_123456=10000
132 Set-Cookie:
STATUS_234567=01000
In this case only the new session recording is sent back (existing STATUS_XXX cookies are preserved in the browser)
133 Cookies in the browser:
STATUS_123456=10000
STATUS_234567=01000
150 Need Authentication
151 credentials
152 Create WebSSO Session

153	Claim
154	Claim
155	Need Authentication
156	Silent Login
157	Get Session From Cookies
158	Claim
159	Claim

Figure 2:

110	Browser
120	IdP_1 acronym for Identity Provider 1
121	IdP_2 acronym for Identity Provider 2
201	RyP_3 acronym for Relying Party 3
202	RyP_4 acronym for Relying Party 4

230	STATUS_123456=10000
	STATUS_234567=01000

231	Set-Cookie: STATUS_456789=11000 STATUS_123456 (delete using expiration attribute) STATUS_234567 (delete using expiration attribute)
232	Cookies in the browser: STATUS_456789=11000
250	Need Authentication
251	silent
252	Get Session From Cookies
253	Claim
254	Claim

Figure 3:

110	Browser
120	IdP_1 acronym for Identity Provider 1
121	IdP_2 acronym for Identity Provider 2
301	RyP_3 acronym for Relying Party 3

302 RyP_4 acronym for Relying Party 4

330 STATUS_456789=11000

331 STATUS_456789=11000

332 Set-Cookie:
STATUS_789012=00001
In this case only the new session recording is sent back (existing STATUS_XXX cookies are preserved in the browser)

333 Set-Cookie:
STATUS_567890=00010

334 Cookies in the browser:
STATUS_456789=11000
STATUS_567890=00010
STATUS_789012=00001

350 Need Authentication

351 Need Authentication

352 Silent (silent service 4)

353 Silent (silent service 5)

354 Get Session From Cookies

355 Get Session From Cookies

356 Claim

357 Claim

358 Claim

359 Claim

Figure 4:

410 upper part

411 BROWSER
Suppose that service 4 is already logged

412 STATE_12_56_9=00010

413 LOGIN(service 5)

414 NODE1
Register service 5

415 STATE_12_56_9=00010

416 STATE_2_15_33=00001

417 LOGIN(service 1)

418 NODE2

Register service 1

419 STATE_12_56_9=00010

420 STATE_47_56_24=10000

430 lower part

431 BROWSER

Suppose that logout is performed against another node

432 STATE_12_56_9=00010

433 STATE_2_15_33=00001

434 STATE_47_56_24=10000

435 RENEW

436 IF MORE THAN ONE STATE COOKIE IS PRESENT THEY ARE AGGREGATED IN A NEWER ONE

437 NODE3

Service list is retrieved aggregating information with last state cookie

438 STATE_66_10_51=10011

439 ~~STATE_12_56_9=00010~~

440 ~~STATE_2_15_33=00001~~

441 ~~STATE_47_56_24=10000~~

Claims

1. A method for managing a SSO session by an identity provider for a plurality of services, the method including the following steps:
a) by an identity provider, managing information on the SSO session via a cookie based protocol;
b) persisting a list of services of relying parties participating in a same SSO session information in one session cookie and a plurality of temporary state cookies with randomly generated names, whereby the list of session services are represented with a bit mask representation within the cookies; and, whereby the plurality of temporary state cookies can be consolidated into one state cookie.

2. The method of claim 1, wherein the bit mask is represented in a compressed format.

3. The method of any of the previous claims, wherein the SSO session state is saved via cookies and can be recovered via cookies and there is no need of persistency of the SSO session state at the server side.

4. The method of any of the previous claims, wherein session static information are represented as a signed ticket whereby both ticket and signature are stored in separate cookies.

5. The method of any of the previous claims, wherein when a new relying party joins a global SSO session, the session cookie is updated by adding the new relying party into the list of the global session participants.

6. A system for managing a SSO session by an identity provider for a plurality of services, the method including the following steps:

a) means for managing information by an identity provider on the SSO session via a cookie based protocol;

b) means for persisting a list of services of relying parties participating in a same SSO session information in one session cookie and a plurality of temporary state cookies with randomly generated names, whereby the list of session services are represented with a bit mask representation within the cookies; and, whereby the plurality of temporary state cookies can be consolidated into one state cookie.

7. The system of claim 6, wherein the bit mask is represented in a compressed format.

8. The system of any of the claims 6 to 7, wherein the SSO session state is saved via cookies and can be recovered via cookies and there is no need of persistency of the SSO session state at the server side.

9. The system of any of the claims 6 to 8, wherein session static information are represented as a signed ticket whereby both ticket and signature are stored in separate cookies.

10. The system of any of the claims 6 to 9, wherein when a new relying party joins a global SSO session, the session cookie is updated by adding the new relying party into the list of the global session participants.

11. A computer program product for performing steps of the method according any of the claims 1 to 5.

FIG 1

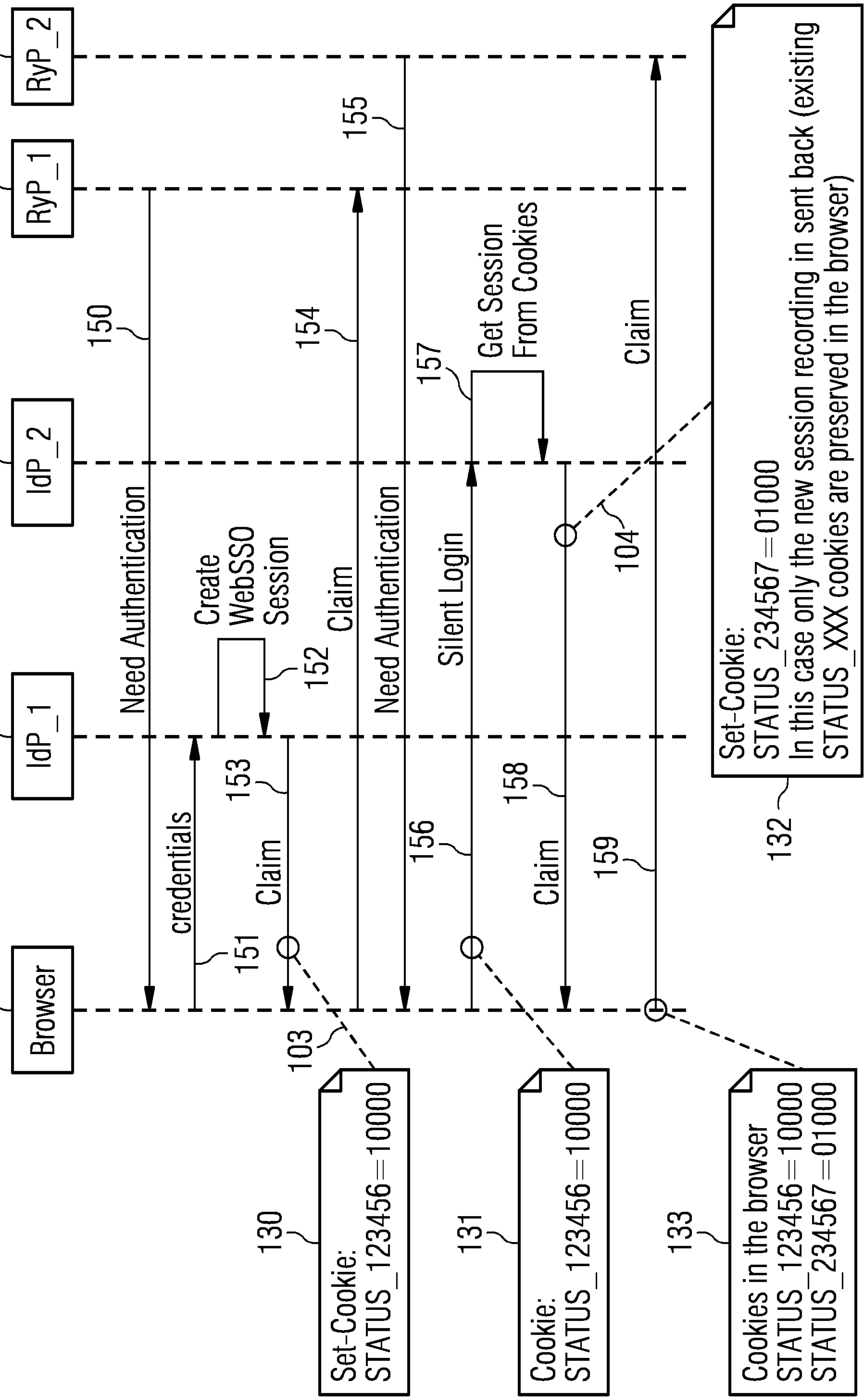

110
Browser
120
IdP_1
121
IdP_2
101
RyP_1
102
RyP_2
150
Need Authentication
151
credentials
152
Create
WebSSO
Session
153
Claim
154
Claim
155
Need Authentication
156
Silent Login
157
Get Session
From Cookies
158
Claim
159
Claim
103
104
130
Set-Cookie:
STATUS_123456=10000
131
Cookie:
STATUS_123456=10000
132
Set-Cookie:
STATUS_234567=01000
In this case only the new session recording in sent back (existing
STATUS_XXX cookies are preserved in the browser)
133
Cookies in the browser
STATUS_123456=10000
STATUS_234567=01000

FIG 2

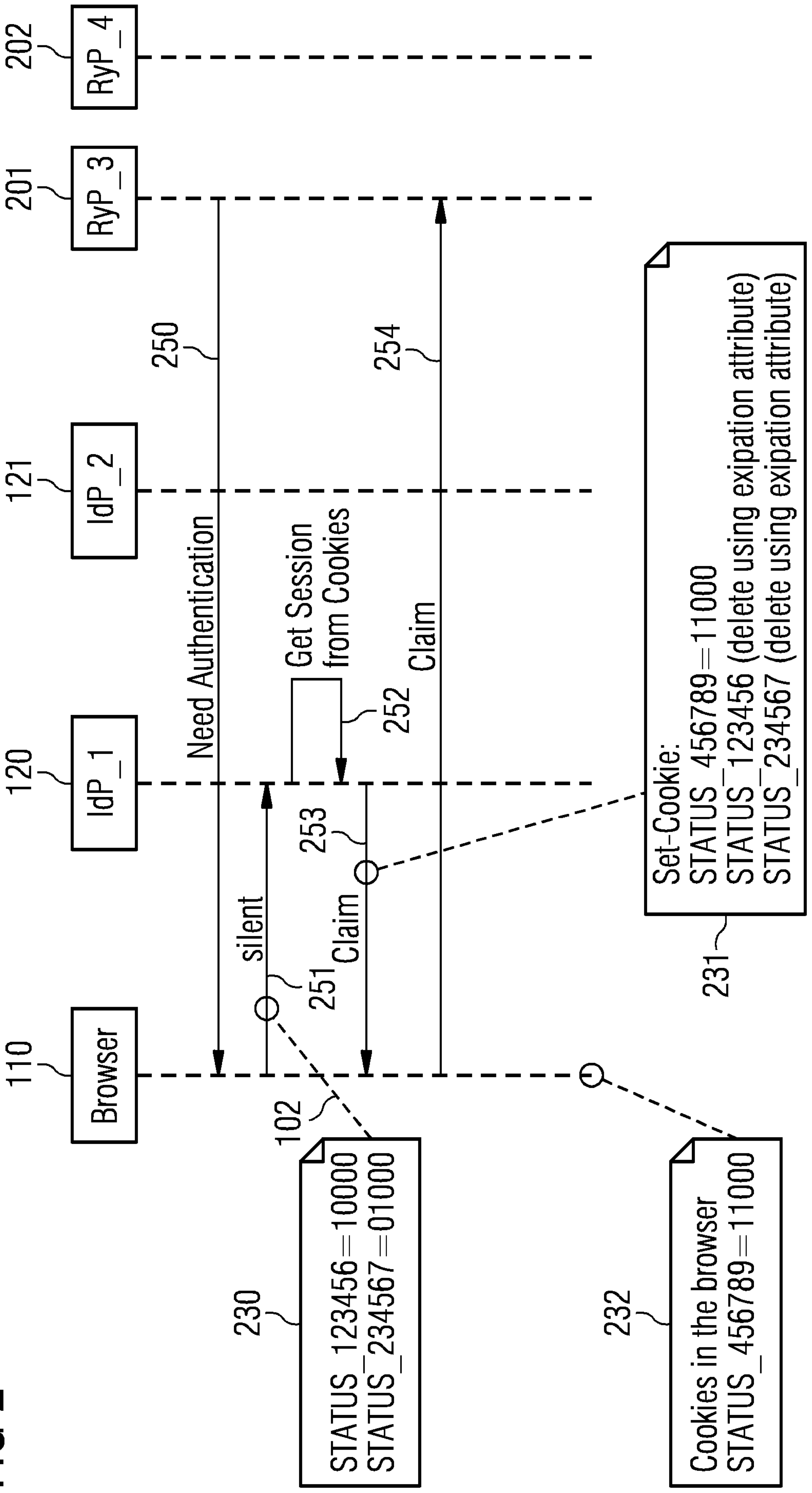

110
Browser
120
IdP_1
121
IdP_2
201
RyP_3
202
RyP_4
250
Need Authentication
251
silent
252
Get Session from Cookies
253
Claim
254
Claim
102
230
STATUS_123456=10000
STATUS_234567=01000
231
Set-Cookie:
STATUS_456789=11000
STATUS_123456 (delete using exipation attribute)
STATUS_234567 (delete using exipation attribute)
232
Cookies in the browser
STATUS_456789=11000

FIG 3

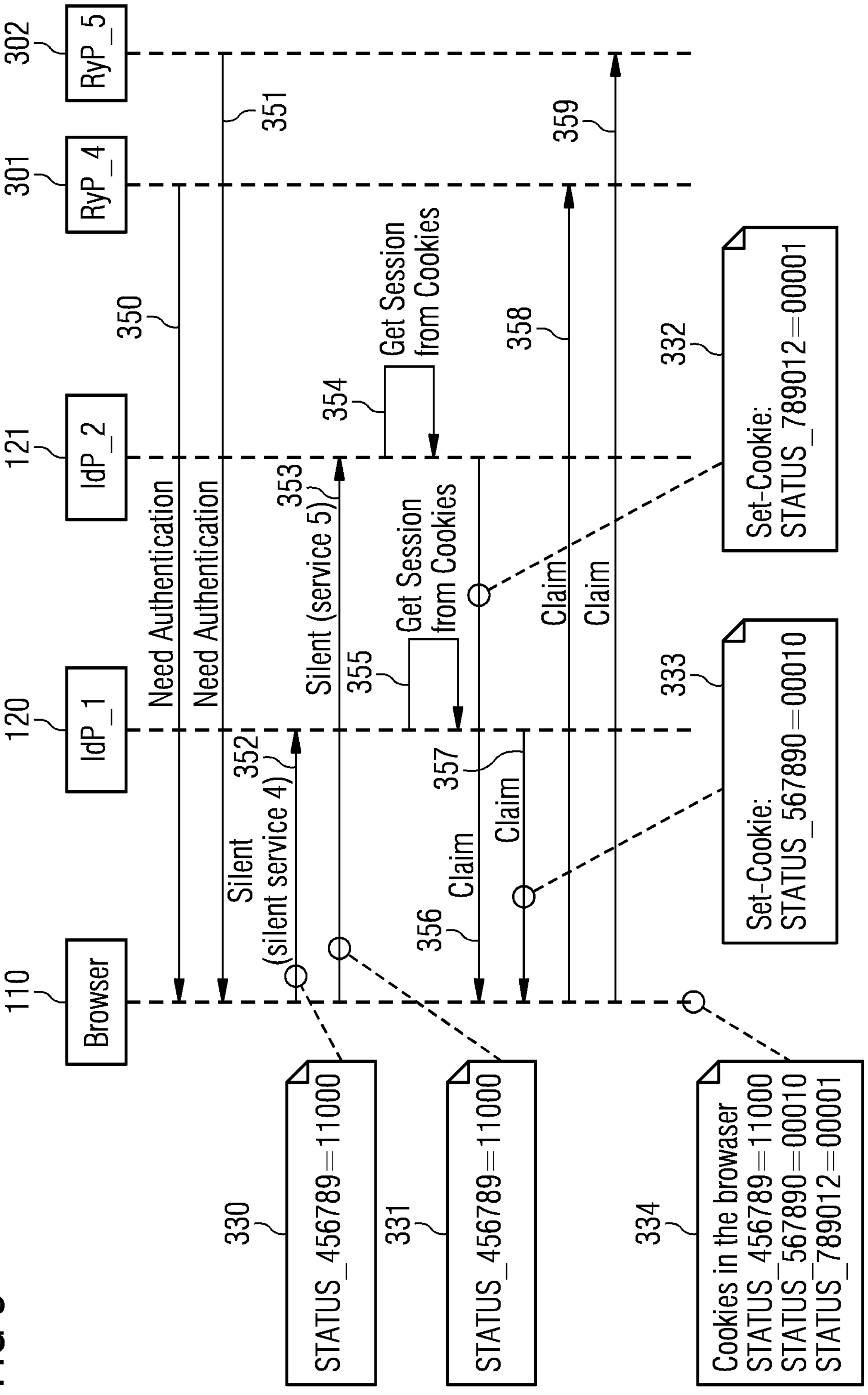
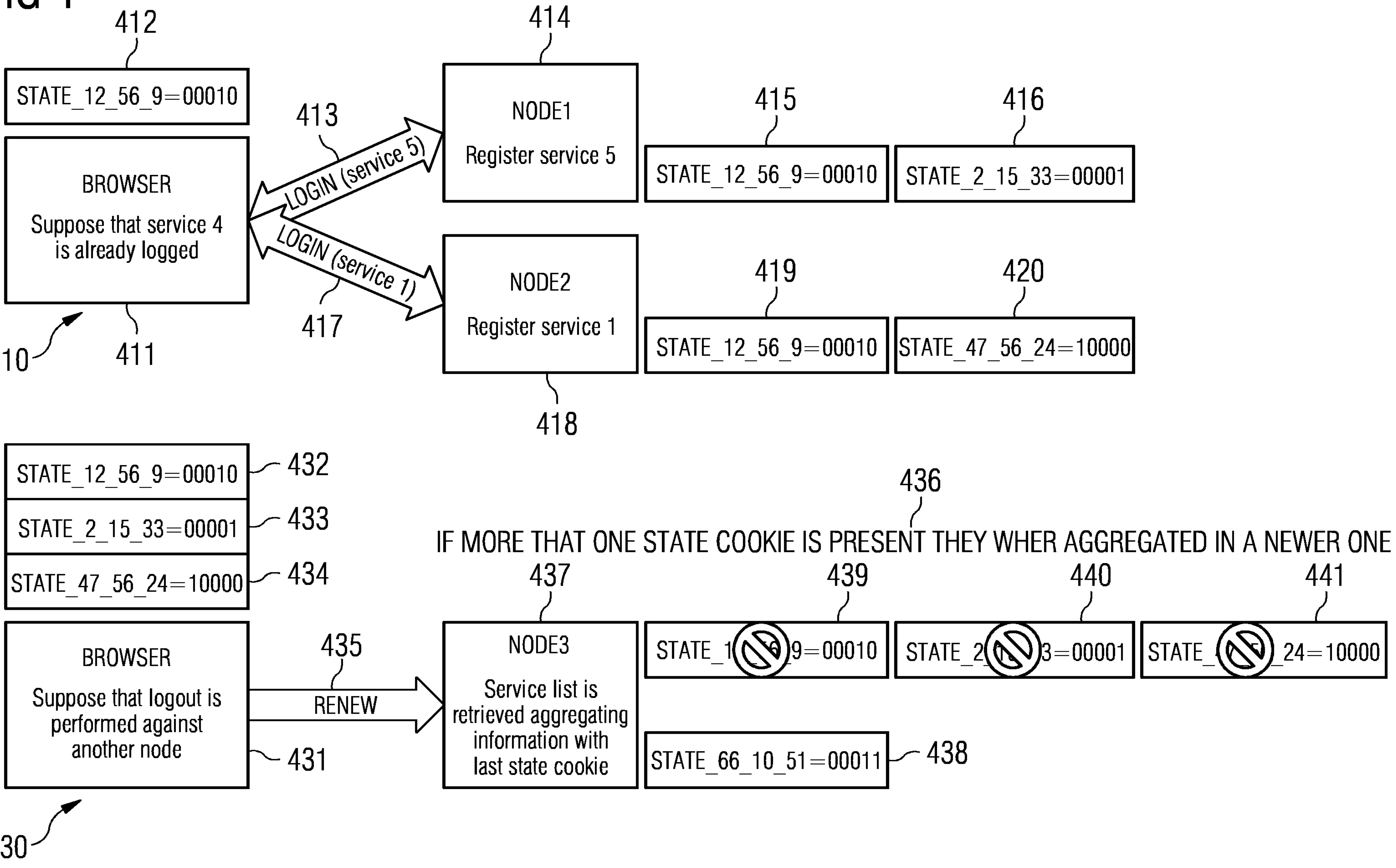

110
Browser
120
IdP_1
121
IdP_2
301
RyP_4
302
RyP_5
350
Need Authentication
351
Need Authentication
352
Silent
(silent service 4)
353
Silent (service 5)
354
Get Session
from Cookies
355
Get Session
from Cookies
356
Claim
357
Claim
Claim
358
Claim
359
Claim
330
STATUS_456789=11000
331
STATUS_456789=11000
332
Set-Cookie:
STATUS_789012=00001
333
Set-Cookie:
STATUS_567890=00010
334
Cookies in the browaser
STATUS_456789=11000
STATUS_567890=00010
STATUS_789012=00001

FIG 4


412
STATE_12_56_9=00010
BROWSER
Suppose that service 4 is already logged
410
411
413
LOGIN (service 5)
LOGIN (service 1)
417
414
NODE1
Register service 5
NODE2
Register service 1
418
415
STATE_12_56_9=00010
416
STATE_2_15_33=00001
419
STATE_12_56_9=00010
420
STATE_47_56_24=10000
STATE_12_56_9=00010
432
STATE_2_15_33=00001
433
STATE_47_56_24=10000
434
BROWSER
Suppose that logout is performed against another node
431
430
435
RENEW
436
IF MORE THAT ONE STATE COOKIE IS PRESENT THEY WHER AGGREGATED IN A NEWER ONE
437
NODE3
Service list is retrieved aggregating information with last state cookie
439
STATE_1 9=00010
440
STATE_2 3=00001
441
STATE_ _24=10000
STATE_66_10_51=00011
438

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2020/058165

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F16/958 H04L29/12 H04L29/08 G06F21/41 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)
G06F 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 practicable, search terms used)

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2017/099299 A1 (SUTTON WESLEY MARLIN [US] ET AL) 6 April 2017 (2017-04-06) abstract paragraph [0018] - paragraph [0030] paragraph [0034] - paragraph [0038] paragraph [0042] - paragraph [0057] paragraph [0063] - paragraph [0078] paragraph [0094] claim 1 -----	1-11
X	US 2005/154887 A1 (BIRK PETER D [US] ET AL) 14 July 2005 (2005-07-14) abstract paragraph [0002] - paragraph [0012] paragraph [0018] - paragraph [0033] paragraph [0043] - paragraph [0048] paragraph [0054] - paragraph [0064] paragraph [0069] - paragraph [0078] ----- -/--	1-11

☒ 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 application or patent but published on or after the international filing date

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search 18 May 2020	Date of mailing of the international search report 29/05/2020
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Authorized officer Boyadzhiev, Yavor

2

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	GB 2 364 408 A (IBM [US]) 23 January 2002 (2002-01-23) abstract page 3, line 15 - page 3, line 35 -----	1-11
A	NORDBOTTEN N A ET AL: "Methods for service discovery in Bluetooth scatternets", COMPUTER COMMUNICATIONS, ELSEVIER SCIENCE PUBLISHERS BV, AMSTERDAM, NL, vol. 27, no. 11, 1 July 2004 (2004-07-01), pages 1087-1096, XP004503638, ISSN: 0140-3664, DOI: 10.1016/J.COMCOM.2004.01.013 abstract page 1088, right-hand column, line 40 - page 1088, right-hand column, line 53 -----	1-11

2

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2020/058165

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2017099299 A1	06-04-2017	US 9531719 B1 US 2017099299 A1	27-12-2016 06-04-2017
US 2005154887 A1	14-07-2005	NONE	
GB 2364408 A	23-01-2002	NONE	