ETSI TR 133 995 V16.0.0 (2020-08)

Universal Mobile Telecommunications System (UMTS);
LTE;
Study on security aspects of integration of
Single Sign-On (SSO) frameworks
with 3GPP operator-controlled resources and mechanisms
(3GPP TR 33.995 version 16.0.0 Release 16)

Reference
RTR/TSGS-0333995vg00

Keywords
LTE,SECURITY,UMTS

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Legal Notice

This Technical Report (TR) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities. These shall be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between 3GPP and ETSI identities can be found under http://webapp.etsi.org/key/queryform.asp.

Modal verbs terminology

In the present document "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Contents

Foreword

This Technical Report has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

- x the first digit:
  - 1 presented to TSG for information;
  - 2 presented to TSG for approval;
  - 3 or greater indicates TSG approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

1 Scope

The present study investigates the security aspects of the service requirements specified by SA1 in TS 22.101 [11] clause 26, on the integration of SSO frameworks with 3GPP networks for various operator authentication configurations (e.g. configurations using GBA or not using GBA).

In particular, this study evaluates existing interworking solutions between SSO frameworks and 3GPP authentication mechanisms against the SA1 service requirements. The study is not limited to evaluation of existing interworking solutions and new interworking solutions may be developed as appropriate.

The study covers the security requirements to enable the operator to become the preferred SSO Identity Provider by allowing the usage of credentials on the UE for SSO services, as well as ways for the 3GPP operator to leverage its trust framework and its reliable and robust secure credential handling infra-structure to provide SSO service based on operator-controlled credentials.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TR 22.895: "Study on Service aspects of integration of Single Sign-On (SSO) frameworks with 3GPP operator-controlled resources and mechanisms".

[3] 3GPP TR 33.980: "Interworking of Liberty Alliance Identity Federation Framework (ID-FF), Identity Web Service Framework (ID-WSF) and the Generic Authentication Architecture (GAA)".

[4] 3GPP TR 33.924: "Identity management and 3GPP security interworking; Identity management and Generic Authentication Architecture (GAA) interworking".

[5] 3GPP TR 33.804: "Single Sign On Application Security for Common IMS – based on SIP Digest".

[6] 3GPP TS 33.220: "Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture".

[7] 3GPP TS 24.109: "Bootstrapping interface (Ub) and network application function interface (Ua); Protocol details".

[8] 3GPP TS 29.109: "Generic Authentication Architecture (GAA); Zh and Zn Interfaces based on the Diameter protocol; Stage 3".

[9] OpenID Foundation "OpenID Authentication 2.0", http://openid.net/.

[10] 3GPP TS 33.222, “Access to network application functions using Hypertext Transfer Protocol over Transport Layer Security (HTTPS)“

[11] 3GPP TS 22.101, “Service aspects; Service principles”.

[12] 3GPP TR 33.905, "Recommendations for trusted open platforms".

[13] OpenID Foundation "OpenID Provider Authentication Policy Extension 1.0", http://openid.net/.

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in TR 21.905 [1], TS 22.101 [11] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in TR 21.905 [1].

example: text used to clarify abstract rules by applying them literally.

Authorization: a mechanism or process which determines what a particular user or a group of users can access or do.

Multi-factor authentication: a method of logon verification where at least two different factors of proof are provided, and jointly verified. There are three generally recognized types of authentication factors:

- Type 1 - Something You Know. Type 1 includes, but is not limited to, passwords, PINs, combinations, code words, or secret handshakes. Anything that a user can remember and then type, say, do, perform, or otherwise recall when needed falls into this category.
- Type 2 - Something You Have. Type 2 includes all items that are physical objects, such as, but not limited to, keys, smart phones, smart cards, USB drives, and token devices. (A token device produces a time-based PIN or can compute a response from a challenge number issued by the server.)
- Type 3 - Something You Are. Type 3 includes any part of the human body that can be offered for verification, such as, but not limited to, fingerprints, palm scanning, facial recognition, retina scans, iris scans, and voice verification.

Multi-step authentication: a method of logon verification where the authentication can take several steps or phases to complete. Multi-step authentication differs from multi-factor authentication in that it does not strictly require that each authentication factor be different, or that multiple factors are evaluated in conjunction.

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in TR 21.905 [1], TS 22.101 [11] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in TR 21.905 [1].

IdP	Identity Provider
RP	Relaying Party
SSO	Single Sign-On

4 Relation of the present study to other related work in 3GPP

Other SSO related work in 3GPP

Completed SA1 work

- SSO requirements, TS 22.101 [11] clause 26;
- Study on integration of SSO frameworks with 3GPP, TR 22.895[2].

Completed SA3 work

- Liberty - GBA interworking, TR 33.980 [3];
- OpenID – GBA interworking, TR 33.924 [4].

- SSO with SIP Digest, TR 33.804 [5].

What is the relation of this study to other work in 3GPP

This study evaluates the completed and ongoing SA3 SSO work against the service requirements identified bySA1 in TS 22.101 [11] clause 26.

All input in this study is intended to have a clear relation to the SA1 service requirements. This study is not intended duplicate functionality supporting SA1 service requirements, when such functionality can be offered by existing SSO mechanisms. In particular existing solutions in other SA3 specifications are evaluated and new ones can be proposed only if the existing solutions would not meet the SA1 service requirements.

5 Potential requirements identified in the present study

The purpose of this clause is to identify potential security requirements in the present study, if any. The requirements may be general or specific to identified SSO frameworks as seen appropriate.

NOTE: No potential requirements were identified in the present study.

6 Solutions for Liberty Alliance/SAML – 3GPP interworking

6.1 General

The purpose of this clause is to investigate the existing (and possible new) solutions for interworking of Liberty Alliance/SAML and 3GPP authentication mechanisms and evaluate the solutions against the SA1 requirements.

NOTE: No solutions were investigated under this clause.

7 Solutions for OpenID – 3GPP interworking

7.1 General

The purpose of this clause is to investigate the existing (and possible new) solutions for interworking of OpenID and 3GPP authentication mechanisms and evaluate the solutions against the SA1 requirements.

7.2 GBA Lite

7.2.1 Rationale for solution

SSO has been identified as one of the most promising applications of GBA. Clearly, the value of this use-case for an external service provider depends on the number of supporting users. This number in turn depends on the availability of GBA-capable phones and the number of operators which have deployed the necessary GBA infrastructure

One way to overcome the initial threshold of supporting users is to simplify the deployment process. This is accomplished using an SSO specific implementation option of GBA called– GBA Lite. Later on, if an operator finds a need to support other applications as well, the SSO specific version can be extended to full GBA.

The solution presented here closely follows the GBA and OpenID interworking described in 3GPP TR 33.924 [4]. The difference is that the BSF and OP are co-located and hence the Zn interface is a matter of internal implementation. This results in a simpler implementation and deployment. All other nodes and interfaces remain unchanged.

The design goals for GBA Lite were the following:

- A simple migration path to use of full GBA
- The Client/UE and RP (Relying Party) follow TR 33.924 [4] without impact
- Aim for simplicity: keep only the core BSF functionality, remove the rest.

7.2.2 Solution description

7.2.2.1 Architecture

The architecture is identical to 3GPP TR 33.924 [4] Figure 4.3-1 except for the co-location of BSF and OP and the consequent internalization of the Zn interface.

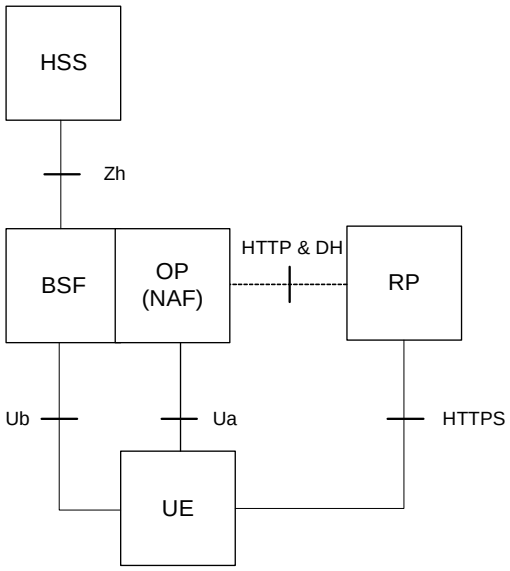

Figure 7.2.2.2-1 GBA Lite Network Architecture

7.2.2.2 BSF Implementation optimizations

No GUSS handling

In ordinary GBA the BSF has to support a wide range of applications with varying options and permissions. In GBA Lite, however, there is only one application: OpenID. This allows us to simplify both the handling of keys and of GBA user security settings (GUSS).

Key handling can be simplified since we only need to deal with OpenID specific keys. For example, the NAF identifier used in the key derivation can be static instead of dynamically determined at the run of the Zn protocol.

The information contained in the GUSS (key lifetime, UICC type, MSISDN etc) can either be statically encoded (key lifetime) or stored as part of the OpenID user account (UICC type, MSISDN). Typically, the OP will maintain a user account for each of its users where the OpenID identifier, attributes, and settings are stored.

The Zh interface can be utilized with minimal effort i.e. no support of GBA User Security Settings (GUSS) is required.

Zn implementation options

Since the Zn interface is internal the vendor or operator is free to choose whatever modifications and optimizations it sees fit. For example, the BSF can be made stateless if the bootstrapping information (B-TID, keys, etc) is pushed over Zn and stored in the OP database. Another option is to use a common database backend and replace Zn with two database calls. Of course, one could also choose not to make any changes and implement the standard Zn interface. The latter approach makes it easier to migrate to full GBA in the future.

7.2.2.3 Message Flow

The following message flow is identical to the Direct Interworking Scenario in TS 33.924 [4] except for the B-TID lookup (step 8 below) and a slightly different wording.

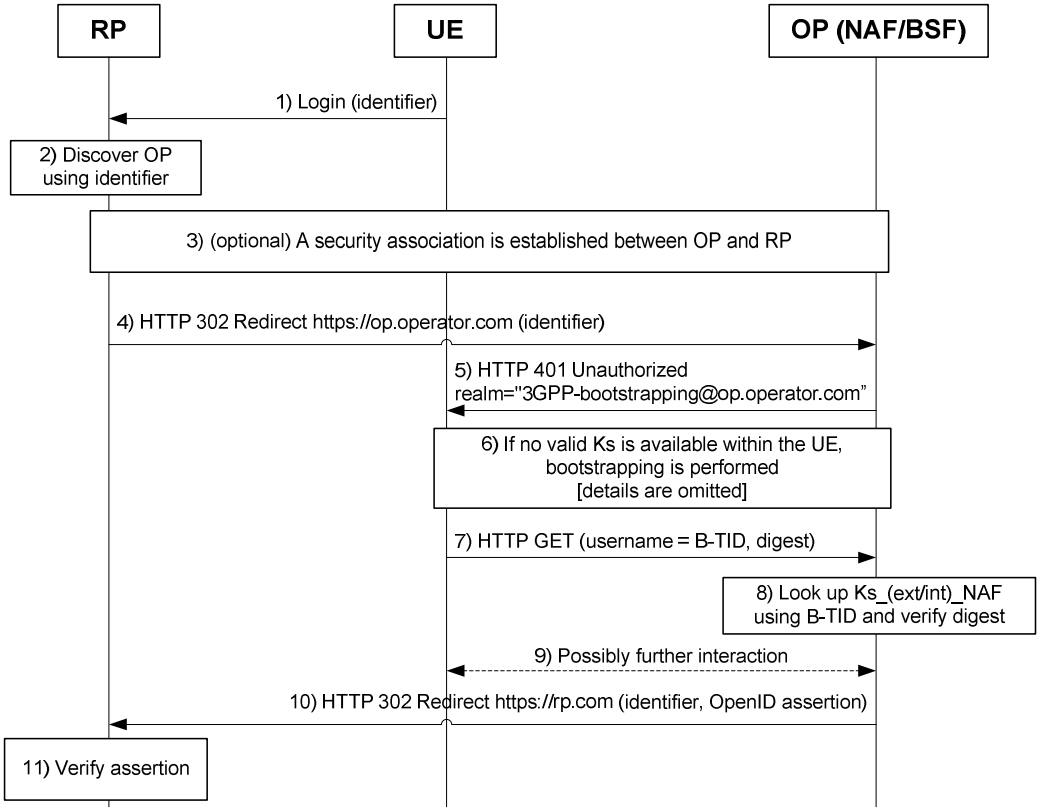

Figure 7.2.2.3-1 Interworking message flow for GBA / OpenID

1) The user initiates authentication by presenting a User-Supplied Identifier to the Relying Party via their User-Agent

2) After normalizing the User-Supplied Identifier, the Relying Party performs discovery on it and establishes the OP Endpoint URL that the end user uses for authentication.

3) (optional) The Relying Party and the OP establish an association – a shared secret established using Diffie-Hellman Key Exchange. The OP uses an association to sign subsequent messages and the Relying Party to verify those messages; this removes the need for subsequent direct requests to verify the signature after each authentication request/response.

4) The Relying Party redirects the end user's User-Agent to the OP with an OpenID Authentication request (Requesting Authentication).

5) The OP (NAF) initiates the UE authentication and responds with a HTTPS response code 401 “Unauthorized”, which contains a WWW Authenticate header carrying a challenge requesting the UE to use Digest Authentication with GBA as specified in TS 33.222 [10] with server side certificates.

6) If no valid Ks is available, then the UE bootstraps with the BSF as described in TS 33.220, which results in the possession of the UE of a valid Ks. From this the UE can derive the application specific (OpenID specific) Ks_(ext/int)_NAF key(s).

7) The UE generates a HTTP GET request to the NAF. The HTTP request carries an authorization header containing the B-TID received from the BSF and a response digest.

8) Using the B-TID the NAF retrieves the shared application specific NAF key and validates the response digest.

NOTE: Since BSF–OP/NAF interface is internal, several implementation options are possible. E.g. the standard Zn interface could be implemented.

9) Possibly further interaction where e.g. the user is made aware that he is logging in to RP with OpenID.

10) The OP redirects the end user's User-Agent back to the Relying Party with either an assertion that authentication is approved or a message that authentication failed.

11) The Relying Party validates the assertion received from the by using either the shared key established during the association or by sending a direct request to the OP. If the validation is successful, then the user is logged in to the service of the RP.

7.2.3 Evaluation against SA1 requirements

The collocated GBA architecture shows an easy entry solution for an operator that has not yet deployed GBA, but would like to have an extensible system.

7.3 Third Party IdP binding for two-factor authentication

7.3.1 Rationale for solution

Enterprises and “Over-The-Top” application services providers (OTT) need a means of asserting users’ identities for their subsequent authorization. Current use of user ID/password credentials is considered as inadequate security for value added applications such as mobile payments and access to enterprise applications.

The most widespread two-factor authentication is based on the user’s ID/password as a first authentication factor (for user’s presence authentication) as well as a hardware-based token as a second authentication factor (confirming a user’s possession of a physical entity such as a token or device on which such token functionality resides).

When a smartphone containing UICC mutually authenticates with its MNO, reuse of the user’s UICC as a second authentication factor allows MNOs to become ID Providers (IDP) and inherently provide more security than the sole use of user ID/password credentials. Existing 3GPP SSO solutions do not provide a means to confirm the presence of a registered user of a data application, nor do they provide a means for binding (e.g. cryptographically) the results of two discrete authentication mechanisms.

Traditionally, 3GPP was focusing on the developing the means to authenticate subscriptions, rather than subscribers (i.e., presence of registered users). Existing SSO solutions do not provide adequate mechanisms to confirm presence of a registered user, since it is the subscription credentials (vs. User credentials) that are being authenticated by existing SSO solutions.

Some of the existing solutions might be deemed capable of providing means for two-factor authentication. Their analysis is presented below.

GBA – Liberty interworking via using GBATwoFactor authentication as described in TS 29.109

TR 33.980 [3] describes 3GPP framework for GBA-Liberty Alliance interworking while not having specific provisions for multi-factor authentication. TS 29.109 [8] in its informative Annex E defines the following information elements and with Associated 3GPP URIs and Class schemas for invoking two-factor authentication using interworking with Liberty Alliance:

GBATwoFactorUnregistered

GBATwoFactorContract

It is, however, unclear how such authentication proceeds, what entity is the Master IDP, and how the binding of authentication factors is being achieved. It is presumed that such binding is possible to accomplish.

GBA – OpenID interworking via using PAPE extensions

PAPE (Provider Authentication Policy Extension) [13] defines a mechanism which allows an OpenID Relying Party to achieve the following:

- request identity providers to use specific authentication policies when authenticating a user.
- require an identity provider to inform the relying party of the authentication policies used during authentication.
- require an identity provider to communicate the levels of authentication used as defined in sets of requested custom assurance levels.

It is possible to use PAPE for the GBA service to request, and to successfully perform GBA authentication. It seems reasonable to have both factors authenticated either in sequence or concurrently. However, PAPE does not seem to provide a mechanism to bind authentication processes for different factors. While PAPE is defined outside of 3GPP, such binding mechanism arguably needs to be defined in 3GPP to be successfully used for multi-factor authentication by 3GPP operators.

SA1 Service Requirements to be taken into SA3 consideration

As part of the technical specification work for Rel-12, 3GPP SA1 defined requirements (see TS 22.101 section 26.1) on providing Single Sign-On service for the UE and the SSO Provider. One of the requirements states that the UE and the SSO Service Provider have mechanisms in place in order to confirm the presence of a registered user of a data application.

In addition, the 3GPP SSO Service is required to support flexibility regarding user configuration of third party SSO identities in the process of gaining access to a service using 3GPP SSO Service. It is required to interwork with such SSO technologies as OpenID (see TS 22.101 section 26.1).

MNO Benefits

Customer records are the biggest MNO asset, together with the MNO's ability to authenticate subscriptions based on AKA credentials residing in the MNO network and UICC. When presence of the user's UICC in the smartphone is verified to serve as a second authentication factor, the MNO becomes an IDP. MNO-provisioned IDP services, anchored on the trust in the MNOs, can be revenue-producing and more importantly, allow MNOs to leverage their ability to provide value-adding authentication services to either over-the-top application services or to enterprises.

Application Services/Enterprise Benefits

Over-the-top application services and enterprises need a secure way of authenticating their users. Two-factor authentication, with user ID/password as the first factor and possession of a token as the second factor, is considered to be a strong form of user authentication.

7.3.2 Considerations on multi-step and multi-factor authentication

Based on the SA1 requirement, this TR is attempting to solve the problem of user authentication. Such authentication is rather new for 3GPP and requires user input for authentication. Using multi-factor and multi-step authentication for user authentication is being described in the following sections of this TR.

The orthogonal relationship between multi-factor and multi-step authentication methods is described in the following diagram 7.3.2-1. For simplicity this diagram lists examples of single step/multi-step and single-factor/multi-factor authentication in a 2x2 table.

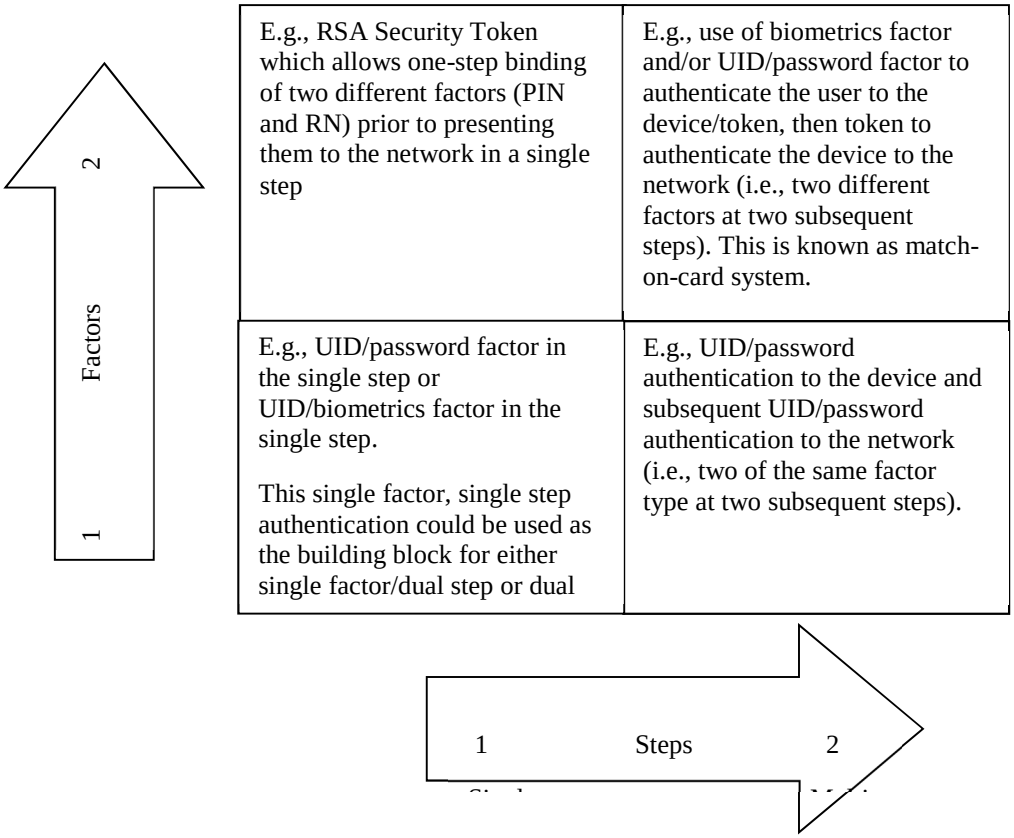

Figure 7.3.2-1. Orthogonal relationship between Multi-step and Multi-factor authentication

Moving from the single-factor quadrant to the quadrant with two-factor (or multi-factor) authentication offers the potential to provide better authentication strength and better authentication assurance. However, increasing authentication steps without increasing authentication factors (i.e., lateral horizontal move from left to right) in most cases provides only marginal authentication strength increase and authentication assurance. In addition, it is useful to observe that multi-step authentication while even employing multiple factors could only be as strong as its weakest step. Particularly, in some of the examples of multi-step authentication (match-on-card) provided above, a token is used as an intermediary, which carries out the first factor authentication and then performs the second factor authentication toward another verifier. In this situation, overall authentication strength also relies on the trust in the intermediary. This exhibits the chain-like nature of multi-step authentication. Even in the example given for the case of single step, single factor, authentication, trust in the intermediary becomes important to achieve optimal assurance. Cached credentials in a browser can weaken the assurance achievable with a two-step authentication.

7.3.3 Solution 1 description

7.3.3.1 General

Example high-level Flow: OTT as a master IDP and MNO as authenticator for factor 2. Figure 7.3.3.1-1

A User attempts to login to an application service (or to an enterprise network) requiring two-factor authentication.

Upon verification of the first authentication factor by an over-the-top (OTT) application service, the OTT initiates a second factor authentication (token-based) with the user's MNO.

When the second factor authentication is completed, the results of the two authentications (from the OTT based on the first factor and from the user's MNO based on the second factor) are bound together by the OTT. Such authentication binding can be achieved either cryptographically or on the protocol level.

1) User Authentication: OTT performs first factor authentication (e.g. using UID/Password) and decides, based on policy, whether to proceed with a second authentication factor;

2) Second Factor Authentication: OTT forwards a request to the Browser Agent for second factor authentication;

3) UE Authentication Request: Browser Agent forwards authentication request to the UE;

4) UICC based Authentication: GBA based authentication occurs based on AKA credentials;

5) Send Result to OTT: Upon successful completion of Step 4, OP/NAF (MNO) asserts UE Identity to the OTT. The functional interface between OP/NAF/MNO and RP/OTT can be realized via OpenID indirect requests using HTTP re-direct;

6) Conclude Second Factor Authentication: OTT receives confirmation of second authentication factor and binds the two authentication factors.

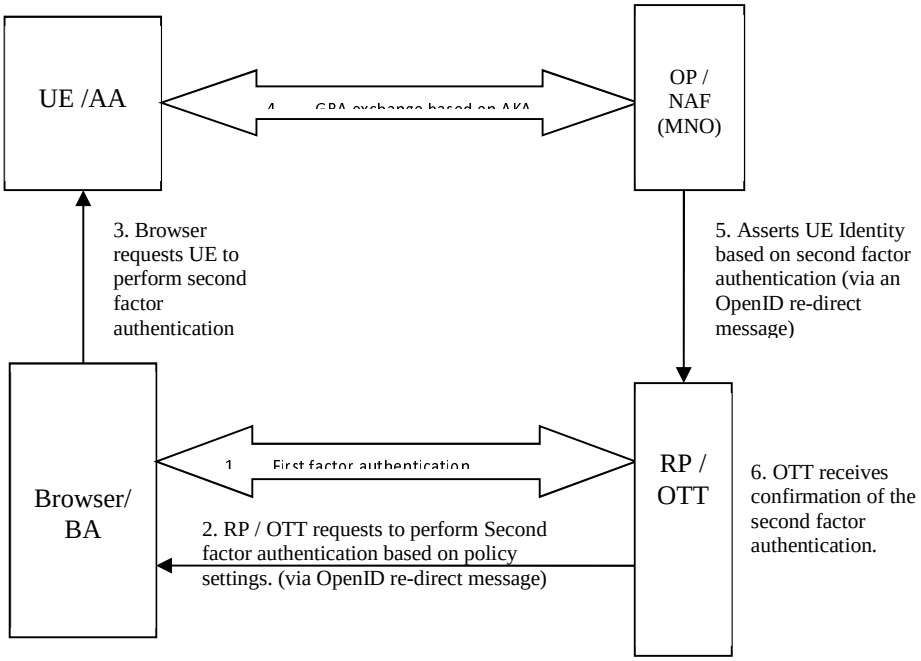

Figure 7.3.3.1-1

Steps 1 through 6 create a "proof of possession", thus providing two-factor authentication for the OTT.

Caching/storing user identity credentials (e.g., user identity and password) in the browser has to be avoided since such caching can potentially interfere with confirming the presence of the "registered user of the data application" and effective user authentication. Preventative measures against storage/caching of user identity credentials can include the use of a freshness indication (e.g. when the password was supplied by the user) in the authentication protocol by utilizing appropriate policies. Defining such preventative measures is outside of the scope of this TR.

Example high-level flow: MNO as Master IdP (authenticator for factor 1 & 2). Figure 7.3.3.1-2.

A User attempts to login (using MNO credentials) to an over-the-top application service (or to an enterprise network).

The OTT, based on policies, determines that two-factor authentication is required and requests the User to perform two-factor authentication with the MNO that works as the master IdP.

Upon verification of the first authentication factor by the OP / NAF, the MNO initiates a second factor authentication (token-based).

When the second factor authentication is completed, the results of the two authentications (first factor based on the User authentication and second factor based on the user's UICC-) are bound together. Such authentication binding can be achieved either cryptographically or on the protocol level.

1) OTT/RP decides, based on its policy, to request the User to perform two-factor authentication;
2) User Authentication: MNO/OP/NAF performs first factor authentication e.g. using UID/Password;
3) UE Authentication Request: Browser forwards authentication request to the UE;
4) UICC based Authentication: GBA based authentication occurs based on AKA credentials;
5) Bind UE/AA and Browser/BA: Upon successful completion of Step 4 and step 2, OP/NAF (MNO) asserts User and UE Identity based on the success of two-factor authentication. The functional interface between OP/NAF/MNO and RP/OTT can be realized via OpenID indirect request using HTTP re-direct;
6) Conclude Second Factor Authentication: OTT receives confirmation of second authentication factor.

This high-level message flow example is amplified in Section 7.3.3.2.

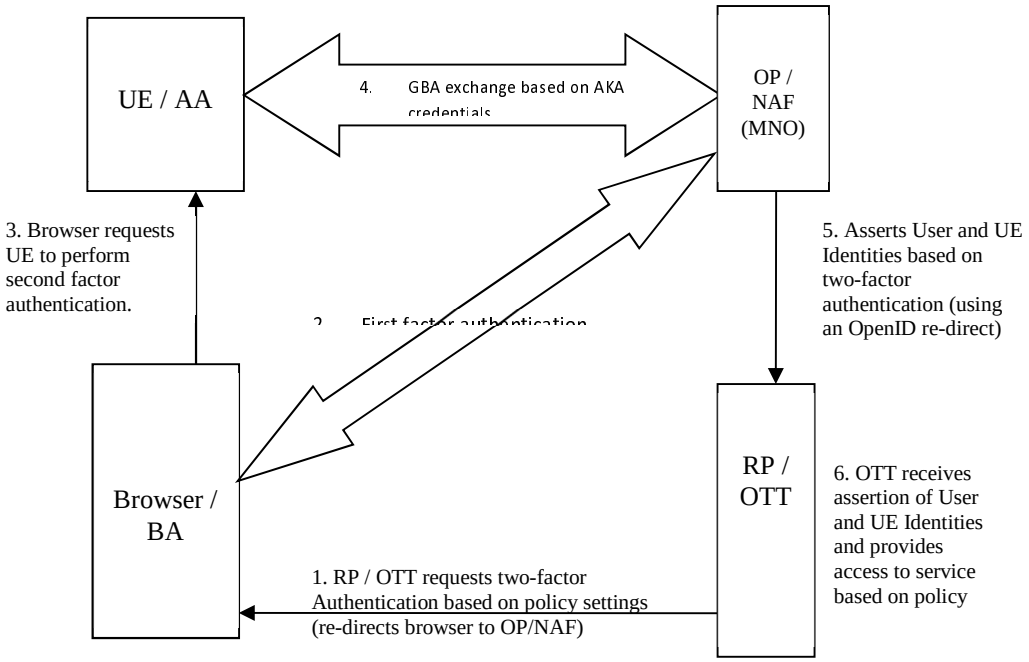

Figure 7.3.3.1-2

7.3.3.2 Example solutions for two factor authentication

Variant 1,

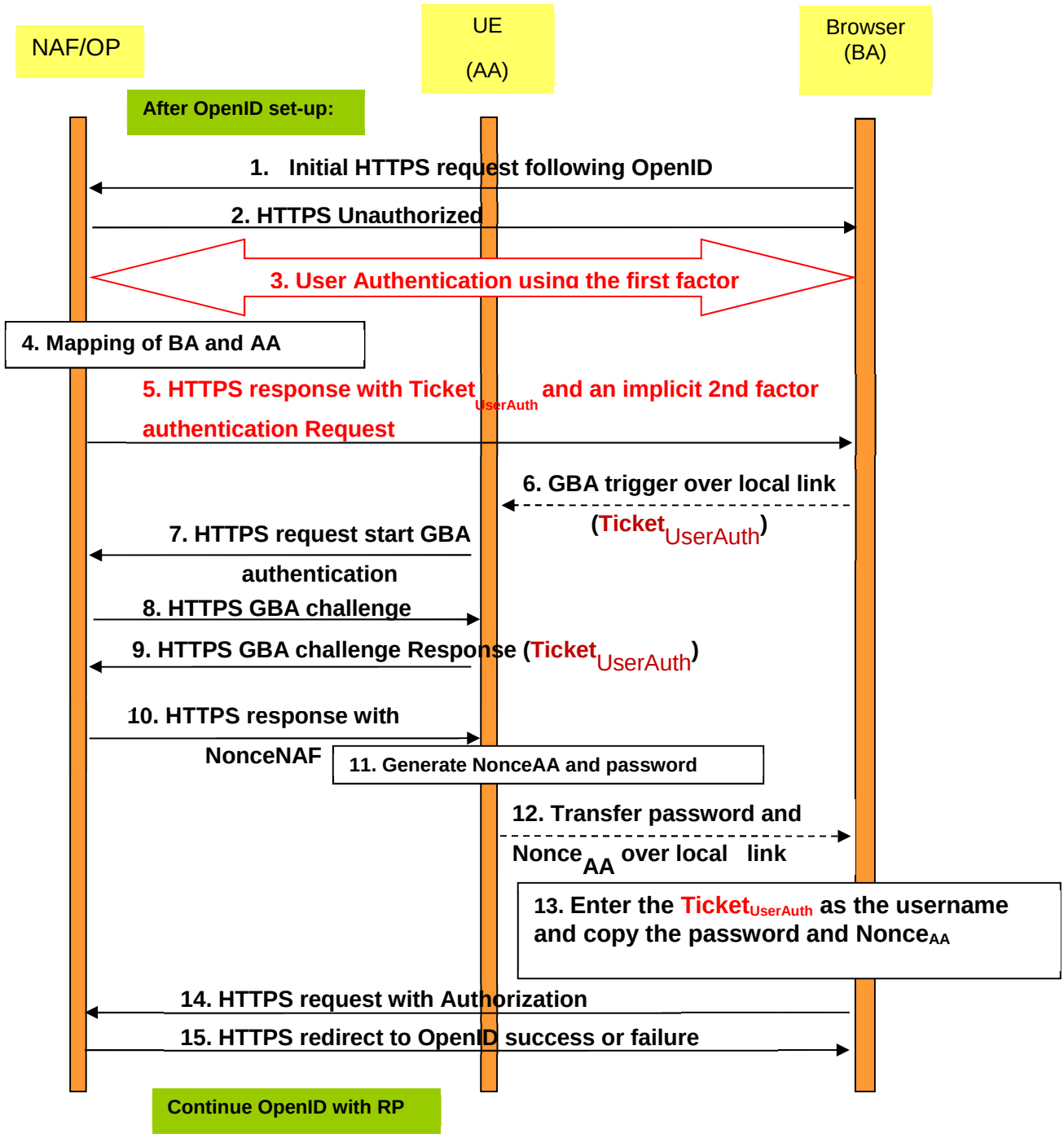

Figure 7.3.3.2 -1

Detailed call flow description

For better understanding of the higher-level diagram 7.3.3-1 and the detailed call flow presented here, note that the RP or Service Provider can be an OTT and OP/NAF can be a MNO.

After the OpenID setup as per specification:

1) Initial HTTPS request following OpenID redirect (same as in TR 33.924, Section 4.4.2. , Fig. 4.4.2.4-3)
2) HTTP Unauthorized Response (same as in TR 33.924, Section 4.4.2. , Fig. 4.4.2.4-3)

3) Message 3 is an aggregate of more than one actual message. It is shown as a single message for simplicity with the intent of being agnostic to any particular authentication mechanism or protocol. User proceeds with the First Factor authentication to OP (e.g., user ID and password). Note that the first factor freshness, e.g. password being cached in the Browser, etc. has to be addressed by the OP policy. To enforce such policies, a Trusted Execution Environment (TEE), a TPM or a similar trusted entity such as a UICC could be needed for policy control (e.g., Policy Enforcement Point and Policy Decision Point.) The way OP addresses enforcement of policies is outside of this Technical Report's scope. Upon successful first factor authentication, a HTTP request is sent by the BA to the OP/NAF requesting a Ticket. This HTTP request is an implicit request within Message 3.

4) Mapping of BA and AA is performed at the NAF / OP.

5) The OP generates a $Ticket_{UserAuth}$ (e.g. a nonce) and sends it within the HTTPS response message, which is in response to the HTTPS request that was sent by the BA as part of the Message 3 exchange. Sending of the $Ticket_{UserAuth}$, has to be interpreted as an implicit request for second factor authentication. Response to this request message is Message 12.

6) GBA is triggered by Message 6, carrying $Ticket_{UserAuth}$ from the Browser (BA) to the UE (AA). This message is corresponding to the (analogous to message in TR 33.924, Section 4.4.2., Fig. 4.4.2.4-3)

7) HTTPS request start GBA authentication (same as in TR 33.924, Section 4.4.2., Fig. 4.4.2.4-3)

8) HTTPS GBA challenge (same as in TR 33.924, Section 4.4.2., Fig. 4.4.2.4-3)

9) HTTPS GBA challenge Response carrying $Ticket_{UserAuth}$ with B-TID from the UE (AA) to the NAF/OP. This message is corresponding to the (analogous to message in TR 33.924, Section 4.4.2. , Fig. 4.4.2.4-3). At this time NAF/OP received $Ticket_{UserAuth}$ and is able to verify that the second factor authentication (UICC-based) is bound to the first factor in Step 3.

10) NAF/OP responds with a NonceNAF

11) The AA generates $Nonce_{AA}$ and uses the NonceNAF and $Nonce_{AA}$ in order to generate a password.

12) The password and $Nonce_{AA}$ is copied over a local link to the BA.

13) Copy NonceAA as Username and password received over the local link

14) Steps 14-15 are reproduced here only for referential integrity with the Solution 3 from TR 33.924. They are not germane for the purpose of this Section.

Variant 2.

For better understanding of the higher-level diagram 7.3.3-1 and the detailed call flow presented here, note that the RP or Service Provider can be an OTT and OP/NAF can be a MNO.

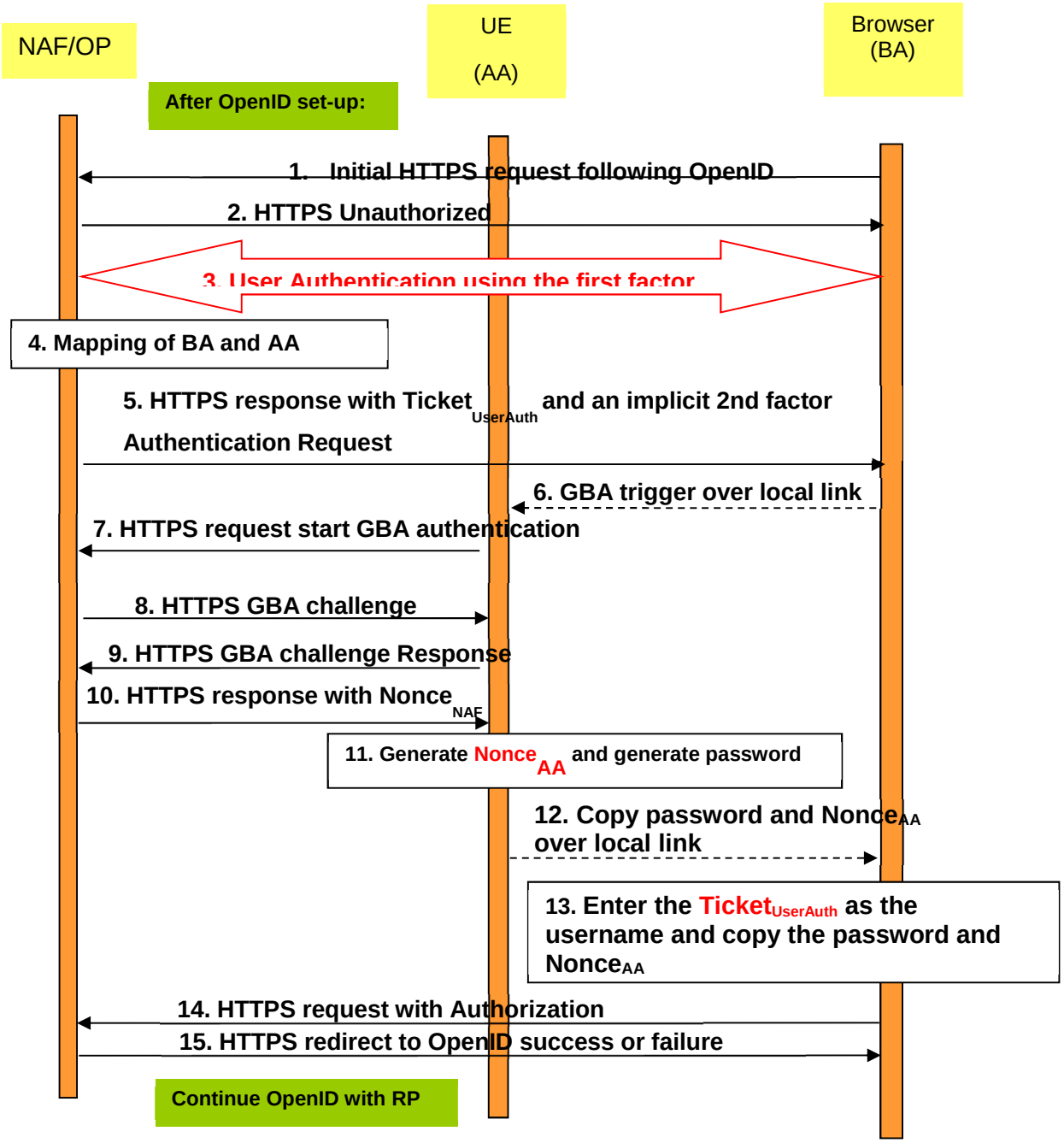

Figure 7.3.3.1-2

After the OpenID setup as per specification:

1) Initial HTTPS request following OpenID redirect (same as in TR 33.924, Section 4.4.2. , Fig. 4.4.2.4-3)

2) HTTP Unauthorized Response (same as in TR 33.924, Section 4.4.2. , Fig. 4.4.2.4-3)

3) Message 3 is an aggregate of more than one actual message. It is shown as a single message for simplicity with the intent of being agnostic to any particular authentication mechanism or protocol. User proceeds with the First Factor authentication to OP (e.g., user ID and password). Note that the first factor freshness, e.g. password being

cached in the Browser, etc. has to be addressed by the OP policy. To enforce such policies, Trusted Execution Environment, similar to the UICC could be needed for execution of policy control and enforcement (e.g., Policy Enforcement Point and Policy Decision Point.) The way OP addresses the first factor freshness e.g. password being cached in the Browser, etc. is outside of this Technical Report's scope. Upon successful first factor authentication, a HTTP request is sent by the BA to the OP/NAF requesting a Ticket. This HTTP request is an implicit request within Message 3

4) Mapping of BA and AA is performed at the NAF / OP

5) The OP generates a TicketUserAuth (e.g. a nonce) and sends it within the HTTPS response message, which is in response to the HTTPS request that was sent by the BA as part of the Message 3 exchange. Sending of the $Ticket_{UserAuth}$, has to be interpreted as an implicit request for second factor authentication. While Message 12 is the response to this request.

6) GBA is triggered by Message 6. This message is corresponding to the (analogous to message in TR 33.924, Section 4.4.2. , Fig. 4.4.2.4-3)

7) HTTPS request start GBA authentication (same as in TR 33.924, Section 4.4.2. , Fig. 4.4.2.4-3)

8) HTTPS GBA challenge (same as in TR 33.924, Section 4.4.2. , Fig. 4.4.2.4-3)

9) HTTPS GBA challenge Response with B-TID from the UE (AA) to the NAF/OP. This message is corresponding to the (analogous to message in TR 33.924, Section 4.4.2.4, Scenario 3).

10) NAF/OP responds with a NonceNAF

11) The AA uses the NonceNAF and $Nonce_{AA}$ in order to generate a password.

12) The password and $Nonce_{AA}$ is copied over a local link to the BA.

13) The $Ticket_{UserAuth}$ is copied into the Username field while the password and $Nonce_{AA}$ received over local link is copied into the Password field. The functionality of $Nonce_{AA}$ and NonceNAF are dedicated to binding AA with BA, and preserved in this example for conformance with the solution described in TR 33.924. The functionality of TicketUserAuth is devoted to binding authentications procedure/for the 1st Factor with the authentication procedure for the 2nd factor.

14) Steps 14-15 are reproduced here only for referential integrity with the Solution 3 from TR 33.924. They are not germane for the purpose of this Section.

7.3.4 Solution 2 description

7.3.4.1 Solution based on OpenID-GBA interworking where OTT performs username/password authentication

The solution presented here is based on OpenID – GBA interworking. Two factor authentication is achieved by the additional step in the beginning where the RP authenticates the user using username/password. Provided the first factor authentication is successful, the RP will redirect the user to the IdP for the second factor GBA based authentication. Once the authentication is done the IdP sends an OpenID token back to the RP via the user, asserting the user's identity.

Since the RP receives the username/password and OpenID token in the same TLS tunnel/HTTP session it is assured that the they were both provided by the same entity, In other words the "binding" between the first and second factor of authentication is accomplished by the TLS tunnel/HTTP session.

A benefit of this solution is that it requires no additional standardization. This is because the first factor of authentication and the binding is handled by the RP on its own, and the RP is not a 3GPP entity.

A high-level message flow is presented below. Note that the order in which the authentications are performed does not matter, An alternative flow would be to perform the username/password authentication after the OpenID authentication. The message flow is based on OpenID 2.0 but it can be possible to use OpenID Connect as well with some small changes to to the message flow.

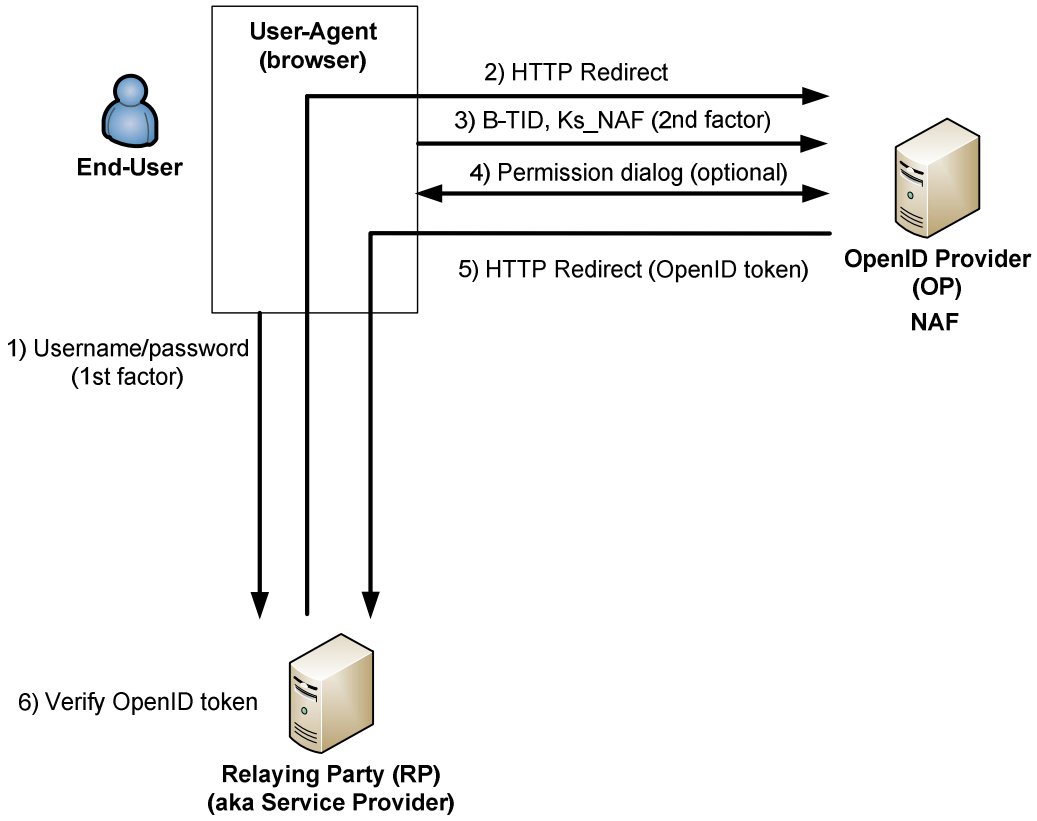

Figure 7.3.4.1-1: Two factor authentication based on OpenID – GBA interworking where OTT performs username/password authentication

1) The user initiates the login process by sending his username/password to the Relying Party via the User-Agent.
2) The Relying Party verifies the username/password, and if successful, redirects the end user's User-Agent to the OP and thereby requests OpenID authentication.
3) The OP initiates GBA authentication which triggers the User-Agent to start authentication using its GBA credentials with the OP.
4) The OP verifies the GBA credentials and, optionally, presents a permission dialog asking the user whether OpenID data can be sent to the OP.
5) If the user gives his approval in step 4, the OpenID assertion is sent to the RP via the User-Agent.
6) The Relying Party verifies the OpenID assertion and if the verification is successful the user is considered logged in.

7.3.4.2 Solution based on OpenID-GBA interworking where MNO performs both GBA and username/password authentication

The solution presented here is based on OpenID – GBA interworking. Two factor authentication is achieved by adding an additional step before the GBA authentication where the OP requests username/password from the user.

Since OP receives the username/password and GBA credentials in the same TLS tunnel or HTTP session it is assured that they were both provided by the same entity, In other words the "binding" between the first and second factor of authentication is accomplished by the TLS tunnel or HTTP session.

The message flow is based on OpenID 2.0 but it can be possible to use OpenID Connect as well with some small changes to to the message flow.

Note that the method for distributing username/password pairs to end-users is considered out-of-scope.

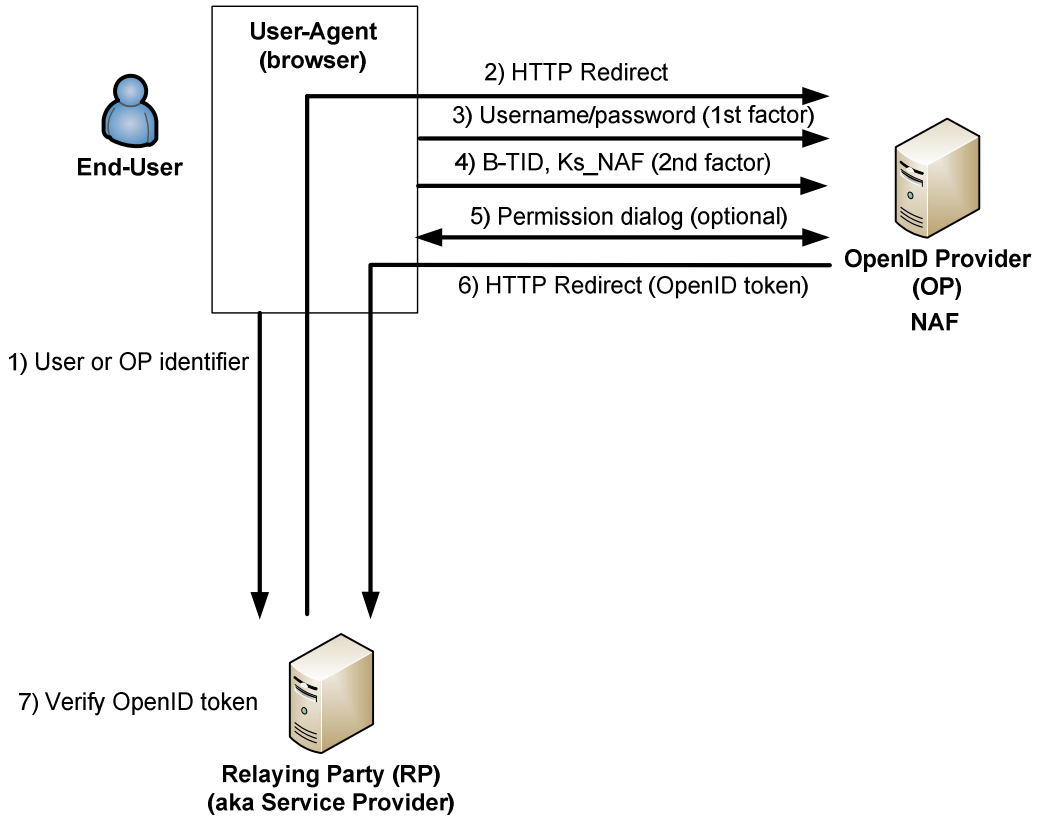

Figure 7.3.4.2-1: Two factor authentication based on OpenID – GBA interworking where OP also performs username/password authentication

1) The user initiates the login process by presenting an identifier of himself or the OP to the Relying Party via the User-Agent

2) The Relying Party redirects the end user's User-Agent to the OP and thereby requests OpenID authentication

3) The OP requests username/password which the end user supplies via the User-Agent

4) Provided the username/password pair is valid, the OP initiates GBA authentication which triggers the User-Agent to start authentication using its GBA credentials with the OP

5) The OP verifies the GBA credentials and, optionally, presents a permission dialog asking the user whether OpenID data can be sent to the RP

6) If the user gives his approval in step 4, the OpenID assertion is sent to the RP via the User-Agent. Optionally, the OP can indicate to the Relaying Party that two-factor authentication was used via the OpenID PAPE extension [13].

7) The Relying Party verifies the OpenID assertion and if the verification is successful the user is considered logged in.

7.3.5 Evaluation against SA1 requirements

The following table summarizes and illustrates conformance of solutions in this TR with SA1 service requirements specified by SA1 in TS 22.101 [11] clause 26, on the integration of SSO frameworks with 3GPP networks for various operator authentication configurations.

Table 7.3.5-1 Summary of conformance with SA1 service requirements specified by SA1 in TS 22.101 [11] clause

##	SA1 service requirement	Solutions in Section 7.3.3.1 and and Section 7.3.4.1, OTT as a master IDP and MNO as authenticator for factor 2	Solutions in Section 7.3.3.2 and and Section 7.3.4.2, MNO as Master IDP	Comments
1	UE shall support 3GPP SSO Authentication, without user intervention, based on Operator-controlled credentials	YES. Since it is based on GBA/AKA credentials	YES. Since it is based on GBA/AKA credentials	Support Requirement #1
2	UE may support a request for SSO Local User Authentication from a Data Application Provider	No.	No.	Request for Local User authentication does not come from DAP Requirement #2 is not supported
3	UE may support a request for SSO Local User Authentication from an Identity Provider	No	No	Requirement #3 is not supported
4	UE may support a request for SSO Local User Authentication…to confirm the presence of the registered user of the data application	No.	No.	Requirement #4 is not supported
5	The 3GPP SSO Service shall be able to interwork with Identity Management (IdM) specifications (e.g., OpenID [51])	Yes.	Yes.	Requirement #5 is satisfied
6	The 3GPP SSO Service shall support 3GPP SSO Authentication based on Operator-controlled credentials and policies	Yes. Only for a single “What you have” factor. Static policies only.	Yes. Only for a single “What you have” factor. Static policies only.	Requirement #5 is satisfied
7	The 3GPP SSO Service may support negotiation and use of an agreed authentication method between the UE and the 3GPP SSO Identity Provider	No negotiation. Pre-provisioned authentication method.	No negotiation. Pre-provisioned authentication method.	Requirement #7 is not satisfied
8	The 3GPP SSO Service may support mechanisms to ensure the presence of the registered user of the data application	Yes. The MNO asserts subscription authentication but has no visibility on User authentication	Yes.Limited. MNO performs User authentication; MNO leaves thejudgment on the presence of the “registerd user” to RP	Requirement #8 is mostly satisfied.
9	The 3GPP SSO Service may support mechanisms to…. satisfy policies of the Data Application Provider	Yes but Limited. It asserts “what you have” factor to RP / DAP. Policies are static.	Yes. MNO as the M-IDP satisfies policies on behalf of RP/DAP. Policies are static.	Requirement #9 is partially satisfied

10	3GPP SSO Service shall be transparent from a user perspective when the user accesses a data application using an identity created through a 3rd Party SSO Identity Provider	Yes. Very much so. Since MNO provides a service ("vouching for what you have") to a UE that has an identity associated with the OTT	No. User/UE has identity associated with MNO	Requirement #10 only partially satisfied. Assumption: "transparency" == Predictability, i.e., the same SSO service with 3rd Party identity as with MNO identity
11	The user shall be able to configure which 3rd party SSO identities are used with the 3GPP SSO Service	No.	No. Not currently and could be expanded	Requirement #11 is not currently satisfied

7.4 Using user consent for GBA and SSO

7.4.1 Rationale for solution

This solution is based on user giving her consent, or authorization, for the GAA server in terminal to derive NAF keys for a specific GAA client. The consent is achieved by a local user authentication (e.g. a PIN) between the user and the User Equipment. The intention of the local user authentication is to confirm the presence of the authorized user according to SA1 requirements in TS 22. 101 [11] and thereby avoid that GBA-based authentication would be used to access services in the background without the user noticing it, and ensure that only authorized persons are able to use GBA-based authentication.

The solution enables confirming that the authorized user is present and gives consent for using GBA keys for an application. Using a nonce approach ensures that the NAF keys are always fresh and not cached in the GAA client.

7.4.2 Solution description

7.4.2.1 General

The solution uses the concepts defined in TR 33.905 [12] "Recommendations for trusted open platforms", where the realization of GBA functionality in a trusted open terminal platform is divided into so called GAA server and GAA client. The GAA server in the terminal is the counterpart of the BSF, and the GAA client in the terminal is the counterpart of the NAF. This is assumed to be a typical division in a terminal implementing GBA. Typically the terminal internal interfaces or APIs are not standardized, and it is not the intention here either. The internals of a terminal are shown in order to explain the solution.

The flow is very similar to the regular GBA flow where the GAA client in the terminal contacts the NAF in order to access a service. The NAF then indicates to the GAA client to use GBA-based keys to secure the Ua application protocol, but in addition the NAF also requires that the presence of the authorized user needs to be confirmed (by sending $Nonce_{UI}$). "UI" stands for "User Involvement". When the GAA client requests NAF keys from the GAA server, the GAA client also consequently requests local user authentication.

The exact mechanism for local user authentication does not need to be specified. It can be for example a PIN code which the user has defined for the GAA server. It can be noted that it is not the same as the PIN to activate the USIM application.

7.4.2.2 GBA_ME-based solution

By local user authentication, the GAA server can locally confirm that the authorized user is present. For instance, the GAA server can present a dialog box to the user asking to authorize that application "Bank.com" can use GBA authentication.

If and only if the GAA server has locally authenticated the user, the GAA server derives new type of NAF keys which are bound to the ongoing transaction by taking the $Nonce_{UI}$ in the NAF key derivation. It can be noted that the result of the local user authentication (e.g. a PIN) is *not* taken into the NAF key derivation. Instead, the GAA server is a trusted element in the terminal which, in addition to performing bootstrapping and deriving NAF keys for applications, is trusted to perform local user authentication when the GAA client indicates that local user authentication is needed. If the GAA client does not indicate that local user authentication is needed, the GAA server derives the regular NAF keys. This approach avoids the burden and complexity of syncing the user authentication credentials, e.g. a PIN, with the network.

The GAA client uses the received NAF keys for authentication in the Ua application protocol. The NAF requests the NAF keys from the BSF and includes the $Nonce_{UI}$ in the Zn request and gets the same NAF keys as the GAA client did.

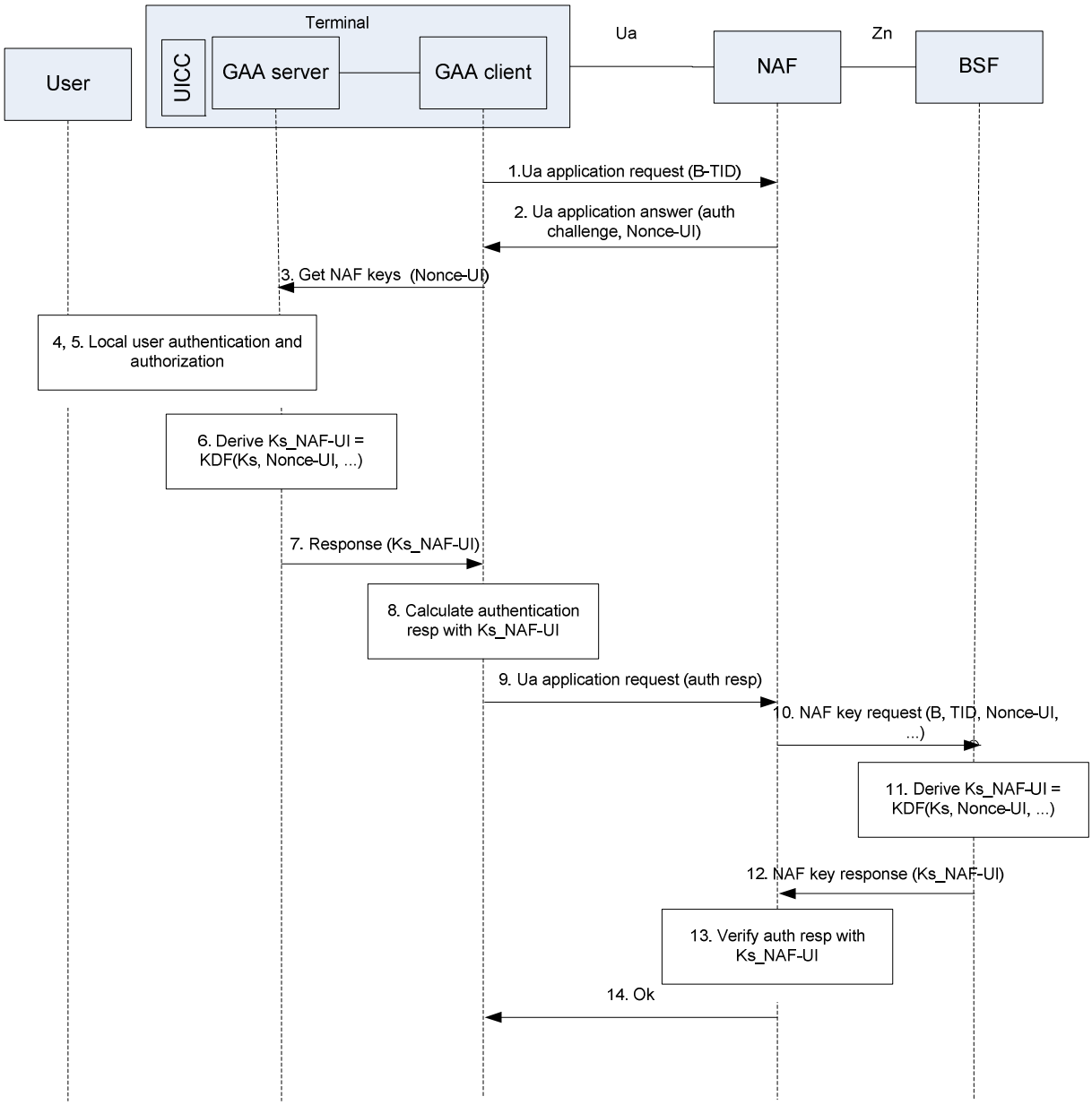

Figure 7.4.2.2.-1: Using User consent for GBA_ME

1) The GAA client in the terminal sends an Ua application request to the application server (i.e. NAF). The request includes the B-TID. In case of GBA – Open ID interworking the UE has been redirected by the RP to contact OP/NAF.

2) The NAF sends back an Ua application answer with an authentication challenge and $Nonce_{UI.}$ The $Nonce_{UI}$ could be sent for exmple in HTTP product token.

3) When the GAA client requests NAF keys from the GAA server in the terminal it includes the $Nonce_{UI}$ in the request.

4) When the GAA server in the terminal receives a request for NAF keys with $Nonce_{UI,}$ the local GAA server requests for local user's authentication and authorization credentials (e.g. a PIN, UID/password, etc.) to derive the NAF keys for this GAA client.

5) The local user provides authentication response/authorization (e.g. PIN, UID/password, etc).

6) If the user authorization was given, (e.g. local authentication of the user based on the provided PIN, UID/password, etc. is correct), , the local GAA server in the terminal derives NAF keys using $Nonce_{UI}$ as an input in the following way Ks_NAF-UI = KDF(Ks, $Nonce_{UI}$, …), where Ks_NAF-UI derivation takes the same input as Ks_NAF derivation, but added with the NonceUI (and with a different FC value). If needed, the GAA server runs bootstrapping before step 6.

7) The GAA server provides Ks_NAF-UI to the GAA client.

8) The GAA client uses the Ks_NAF-UI as the key to calculate the authentication response for the Ua application request.

9) The GAA client sends the Ua application request to the NAF.

10) The NAF requests NAF keys, and optionally USS, from the BSF over Zn. $Nonce_{UI}$ is included in the request.

11) When the BSF receives the Zn request with $Nonce_{UI}$, the BSF calculates the Ks_NAF-UI using $Nonce_{UI}$ as an input in the NAF key derivation similarly as in step 6.

12) The BSF sends Zn response with Ks_NAF-UI to the NAF.

13) The NAF uses the received Ks_NAF-UI to verify authentication response received from the GAA client in step 9.

14) The NAF sends an Ua response to the GAA as a result of a successful authentication. In case of GBA – Open ID interworking the UE is re-directed back to the the RP.

The flow shows a generic authentication handshake between the GAA client and the NAF over Ua relying on GBA_ME to illustrate how the mechanism works, and it can be noted that the derived NAF keys could be used to protect in principle any Ua application protocol.

Note that trusted platform is required for deployment of GAA Server and GAA Client in ME, to fulfill the requirements of TR 33.905[12]. The definition of such trusted platform is outside of 3GPP scope.

In addition, an appropriate protocol for negotiation UE-supported local user authentication capabilities vs. required by the NAF authentication capabilities could be needed.

7.4.2.3 GBA_U-based solution

By local user authentication, the UICC can locally confirm that the authorized user is present. For instance, the GAA server can present a dialog box to the user asking to authorize that application "Bank.com" can use GBA authentication. The GAA server computes and sends to the UICC the hash of the $Nonce_{UI}$ concatenated with the user answer.

If and only if the UICC application has locally authenticated the user, the UICC derives new type of NAF keys which are bound to the ongoing transaction by taking the $Nonce_{UI}$ in the NAF key derivation. It can be noted that the result of the local user authentication (e.g. a PIN) is *not* taken into the NAF key derivation. Instead, the UICC is a tamper resistant device in the User Equipment which, in addition to performing bootstrapping and deriving NAF keys for applications, is trusted to perform local user authentication when the GAA client indicates that local user authentication is needed. If the GAA client does not indicate that local user authentication is needed, the UICC derives the regular NAF keys. This approach avoids the burden and complexity of the user authentication credentials synchronization, e.g. a PIN, with the network.

The GAA client uses the received NAF keys for authentication in the Ua application protocol. The NAF requests the NAF keys from the BSF and includes the $Nonce_{UI}$ in the Zn request and gets the same NAF keys as the GAA client did.

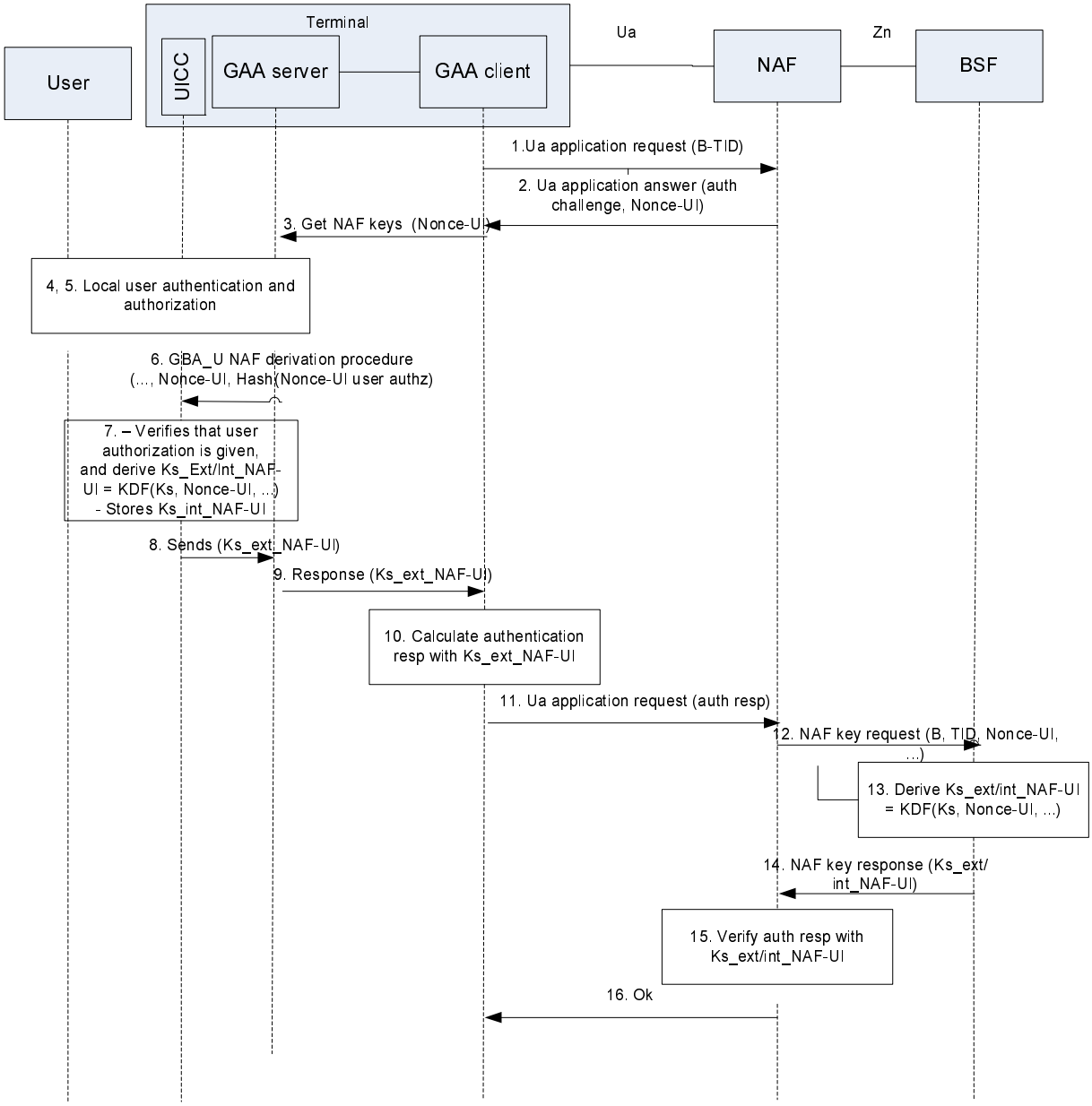

Figure 7.4.2.3.-1: Using User consent for GBA_U

1) The GAA client in the terminal sends an Ua application request to the application server (i.e. NAF). The request includes the B-TID. In case of GBA – Open ID interworking the UE has been redirected by the RP to contact OP/NAF.

2) The NAF sends back an Ua application answer with an authentication challenge and $Nonce_{UI.}$ The $Nonce_{UI}$ could be sent for exmple in HTTP product token.

3) When the GAA client requests NAF keys from the GAA server in the terminal it includes the $Nonce_{UI}$ in the request.

4) When the GAA server in the terminal receives a requst for NAF keys with $Nonce_{UI,}$ the local GAA server requests for local user's authentication and authorization (e.g. a PIN, UID/password, etc.) to derive the NAF keys for this GAA client.

5) The user provides authentication response/authorization (e.g. PIN, UID/password, etc.).

6) The GAA server in the terminal sends GBA_U NAF Derivation procedure to the UICC application including as additional parameters the $Nonce_{UI}$ and hash value of the user's authorization (e.g. a PIN) concatenated $Nonce_{UI}$ (Hash ($Nonce_{UI}$ || user authz)) .

7) The UICC verifies that the user is authorized, e.g. the provided user credential (e.g., PIN UID/password, etc.) is correct by retrieving the user authorization value already stored on the UICC to compute the corresponding Hash value ($Nonce_{UI}$ || user authz) and compare it with hash value sent by the GAA server as input data of the GBA_U NAF derivation procedure. If the user authorization was given, the UICC application derives NAF keys using $Nonce_{UI}$ as an input in the following way Ks_ext/int_NAF-UI = KDF(Ks, $Nonce_{UI}$, …), where Ks_ext/int_NAF-UI derivation takes the same input as Ks_ext/int_NAF derivation, but added with the $Nonce_{UI}$ (and with a different FC value). If needed, the GAA server runs bootstrapping before step 6. The UICC stores Ks_int_NAF-UI.

NOTE: The user authorization reference value is stored as TLV (Tag Length Value) object in a file of the UICC protected by Access Conditions.The usage of TLV object lets open the type and format of the user authorization value (e.g. PIN) that could be chosen. The user authorization reference value could be set by the user and stored in the UICC by the GAA server.

8) The UICC sends back to the GAA server Ks_ext_NAF-UI

9) The GAA server provides Ks_ext_NAF-UI to the GAA client.

10) The GAA client uses the Ks_ext_NAF-UI as the key to calculate the authentication response for the Ua application request.

11) The GAA client sends the Ua application request to the NAF.

12) The NAF requests NAF keys, and optionally USS, from the BSF over Zn. $Nonce_{UI}$ is included in the request.

13) When the BSF receives the Zn request with $Nonce_{UI}$, the BSF calculates the Ks_ext/int_NAF-UI using $Nonce_{UI}$ as an input in the NAF key derivation similarly as in step 6.

14) The BSF sends Zn response with Ks_ext/int_NAF-UI to the NAF.

15) The NAF uses the received Ks_ext_NAF-UI to verify authentication response received from the GAA client in step 11.

16) The NAF sends an Ua response to the GAA as a result of a successful authentication. In case of GBA – Open ID interworking the UE is re-directed back to the the RP.

The flow shows a generic authentication handshake between the GAA client and the NAF over Ua relying on GBA_U to illustrate how the mechanism works.

Note that trusted platform is required for deployment of GAA Server and GAA Client in ME, to fulfill the requirements of TR 33.905[12]. The definition of such trusted platform is outside of 3GPP scope.

In addition, an appropriate protocol for negotiation UE-supported local user authentication capabilities vs. required by the NAF authentication capabilities could be needed.

7.4.3 Functional Architecture

An example of the functional architecture of the solution with local user authentication is depicted in Figure 7.4.3-1.

The Authorization Function on the UE works as a proxy to the multi-factor authentication server (e.g., OP/NAF) and carries out authentication on behalf of the server. The role of the proxy is to carry out policies as specified by the server and to provide an authorization to use the GBA authentication. The server might delegate more than one factor of authentication (this can be based on knowledge of the capability of the UE and on the server policy) to the local proxy and provision the proxy with policies on how the authentications are to be carried out, how often, under what circumstances, and a minimum level of confidence in the user identity that can be achieved based on Service Provider (SP) requirements. It is assumed that the proxy operations are protected by a secure environment on the UE.

The operation of a solution implementing such functional architecture is as follows:

- A user requests service from an SP.
- The SP wishes to authenticate the user with a minimum level of confidence in the user identity to allow access to use the GBA authentication for the requested service, leveraging the availability of a diverse set of authentication capabilities becoming available on user devices.
- The user provides input of credentials over the UE user interface.
- User credentials are matched and assertions generated.
  Note: The mechanism for matching user credentials as well as specifying types of credentials are outside of the scope of this document.
- The assertions are analysed by the Authorization Function.
- The Authorization Function on the UE confirms the assertions and provides the authorization to use the GBA authentication.
- Upon successful conclusion of a GBA authentication, the SP receives implicit confirmation of the local user authentication and then allows access to the service requested by the user.

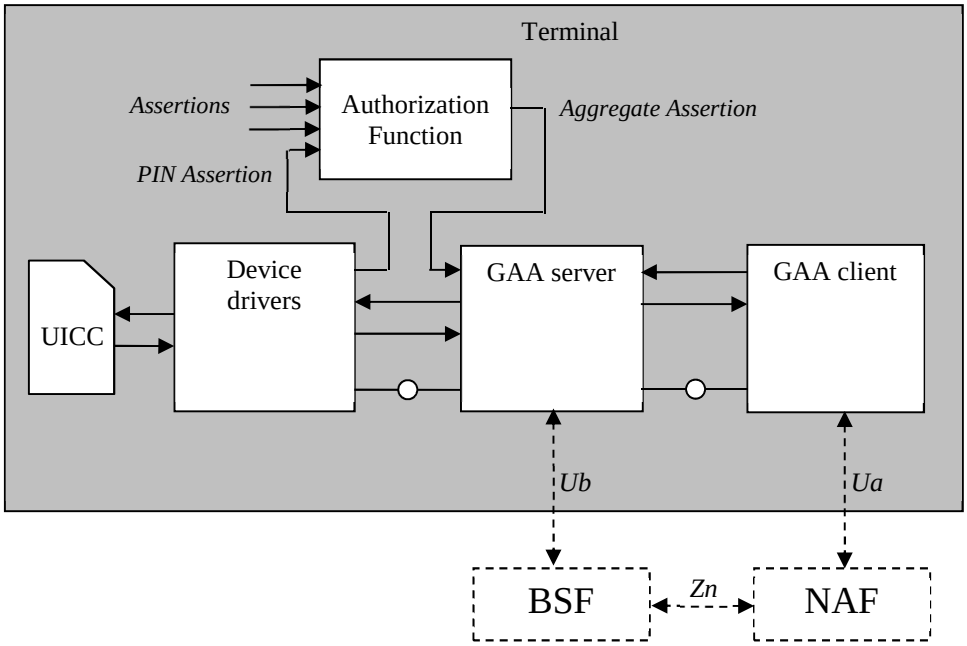

Figure 7.4.3-1 Functional Architecture of a GAA Solution with Local User Authentication.

NOTE : Device drivers, Authorization Function, GAA Server, GAA Client, and user interface have to operate in a secure environment (e.g., UICC, external Smart Card, or Secure Environment on ME)

The benefits of the local user authentication approach used as either the only method of authentication or in combination with the more traditional server-based authentication (e.g., AKA, GBA, etc.) include the following:

- A high level of assurance that the rightful subscriber has provided consent and authorization for the use of GBA authentication on the UE.
- Users' credentials never leave the UE, and could reside in the Secure Environment on the UE. This can be potentially very useful with credentials that are difficult to revoke and re-issue (e.g., biometric credentials). Such approach could alleviate privacy concerns of potential users and help to accelerate adoption of the service.
- Autonomous local user authentication becomes achievable, allowing user authentication when network connectivity is not possible (e.g. to unlock the phone after first power on).

7.4.4 Evaluation against SA1 requirements

The following table summarizes and illustrates conformance of solutions in this Section with SA1 service requirements specified by SA1 in TS 22.101 [11] clause 26, on the integration of SSO frameworks with 3GPP networks for various operator authentication configurations.

Table 7.4.4-1 Summary of conformance with SA1 service requirements specified by SA1 in TS 22.101 [11] clause

##	SA1 service requirement	Solution in Section 7.4, User Consent	Comments
1	UE shall support 3GPP SSO Authentication, without user intervention, based on Operator-controlled credentials	YES. Since it is based on GBA/AKA credentials	Support Requirement #1
2	UE may support a request for SSO Local User Authentication from a Data Application Provider	Solution does not provide for DAP to request Local User Authentication	Request for Local User authentication does not come from DAP Requirement #2 is not supported
3	UE may support a request for SSO Local User Authentication from an Identity Provider	Yes partially (implied request for local authentication).	Requirement #3 is partially supported
4	UE may support a request for SSO Local User Authentication…to confirm the presence of the registered user of the data application	Yes. Partially. It is still not a true User authentication	Requirement #3 is partially supported
5	The 3GPP SSO Service shall be able to interwork with Identity Management (IdM) specifications (e.g., OpenID [51])	Yes. Maybe needs to be better reflected in call flows	Requirement #5 satisfied
6	The 3GPP SSO Service shall support 3GPP SSO Authentication based on Operator-controlled credentials and policies	Yes. Since it is based on AKA, and PIN can be provisioned by MNO, making both credentials operator-controlled.	Requirement #5 is satisfied
7	The 3GPP SSO Service may support negotiation and use of an agreed authentication method between the UE and the 3GPP SSO Identity Provider	No negotiation. Pre-provisioned authentication method.	Requirement #7 is not satisfied
8	The 3GPP SSO Service may support mechanisms to ensure the presence of the registered user of the data application	Yes, limited. PIN is a weak form of shared secret. User has to provide PIN to UICC to trigger GBA process. One of the advantages is that Local authentication can be achieved offline and can be useful for continuous authentication.	Requirement #8 is mostly satisfied.
9	The 3GPP SSO Service may support mechanisms to…. Satisfy policies of the Data Application Provider	Yes, only if the local PI authentication is the DAP policy	Requirement #9 is partially satisfied
10	3GPP SSO Service shall be transparent from a user perspective when the user accesses a data application using an identity created through a 3rd Party SSO Identity Provider	No. The identity of the User / UE is associated to the MNO and NOT a third party IdP	Requirement #10 only partially satisfied. Assumption: “transparency” == Predictability, i.e., the same SSO service with 3rd Party identity as with MNO identity
11	The user shall be able to configure which 3rd party SSO identities are used with the 3GPP SSO Service	Not currently. May be expanded	Requirement #11 is not currently satisfied

7.5 3rd party SSO identity mapping

7.5.1 Rationale for solution

SA1 TS 22.101 [11] has the following requirement:

"The user shall be able to configure which 3rd party SSO identities are used with the 3GPP SSO Service."

It is assumed that "3rd party SSO identities" mean web user identities used with the 3GPP SSO Service, which were not assigned by the 3GPP operator, but a 3rd party. For example, when web service, like a social network site uses a 3GPP SSO Service provided by a 3GPP operator, an example of a 3rd party SSO identity could be user-x@socialnetwork.com. For simplicity we will use the term web user identity in the following.

GBA provides the possibility to use different user identities, including 3rd party SSO identities, for user authentication. This is facilitated by the User Security Settings (USS), which can include a list of user identities, which the NAF can use to authenticate the user with the NAF specific key.

The USS is stored in the HSS as part of the user specific GBA USS (GUSS). It is not defined in GBA specifications how the user identities are allocated to the USS, i.e. mapped to the 3GPP subscription, but this is left for configuration and thus out of scope of 3GPP specifications. Consequently, also the security measures for the identity mapping are left out of scope. If adequate security measures are not in place to verify that a person is authorized to request a mapping, it could be possible that an attacker could be able to map a victim's web user identity to the 3GPP subscription of the attacker. This could make the attacker able to access the victim's web service account.

7.5.2 Solution description

A solution is described which allows only authorized entities to map a web user identity to a 3GPP subscription.

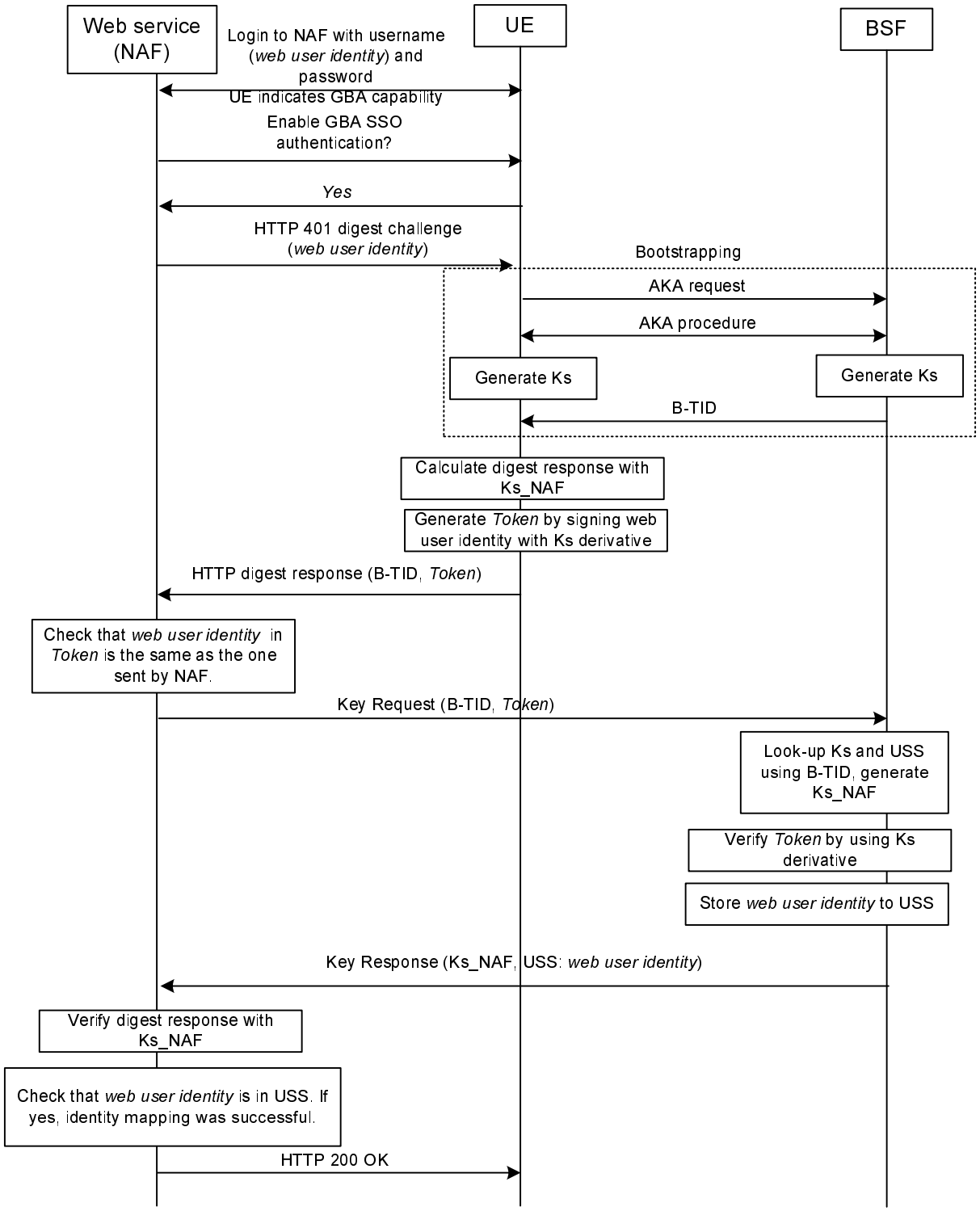

Figure 7.5.2-1 Identity mapping

The procedure works as follows:

- User logs in to a web service using the credentials of the web service (e.g. username (web user identity) and password). The UE indicates capability to use GBA for SSO.
- The web service asks the user whether she would like map her web user identity to her 3GPP subscription and use GBA and SSO when authenticating to the web service in the future.
- The user replies yes in an appropriate HTTP message.

- The web service sends 401 digest challenge with the web user identity included.
- If valid Ks is not available, GBA bootstrapping is performed.
- The UE calculates the digest response with NAF specific key. The UE also generates a Token, which includes the web user identity signed by a key derived from the Ks.
- The UE sends the digest response including the Token to the web service.
- The web service checks that the web user identity in the Token matches the web user identity sent in 401 challenge.
- The web service requests NAF specific key and USS over Zn and includes the Token.
- Upon receiving the request, the BSF looks up the Ks and USS using the B-TID and generates the NAF specific key. The BSF verifies the Token using a key derived from the Ks. If the verification is successful, the web user identity in the Token is stored in the USS. Since the Token was signed by the UE and the Token came from trusted node (i.e. NAF), the BSF can be sure that identity mapping was legitimate.
- The BSF sends Zn response to the web service including the NAF specific key and the USS. The USS includes the newly mapped web user identity.
- The web service checks if the web user identity is in the USS. If yes, mapping was successful. If not, the procedure is aborted. The web service verifies the digest response with NAF specific key.
- Web service sends 200 OK to the UE. User is now authenticated with web user identity and GBA to the web service. In the future, the user can be authenticated to the web service using GBA.

7.5.3 Evaluation against SA1 requirements

The following table summarizes and illustrates conformance of solutions in this Section with SA1 service requirements specified by SA1 in TS 22.101 [11] clause 26.

Table 7.5.3-1 Summary of conformance with SA1 service requirements specified by SA1 in TS 22.101 [11] clause 26

##	SA1 service requirement	Solution in 7.Y.2 3rd party identity mapping	Comments
1	UE shall support 3GPP SSO Authentication, without user intervention, based on Operator-controlled credentials	YES. Since it is based on GBA/AKA credentials	Requirement #1 is supported.
2	UE may support a request for SSO Local User Authentication from a Data Application Provider	No.	Requirement #2 is not supported. Conditionally yes if this combined with User consent solution.
3	UE may support a request for SSO Local User Authentication from an Identity Provider	No.	Requirement #3 is not supported. Conditionally yes if this combined with User consent solution.
4	UE may support a request for SSO Local User Authentication…to confirm the presence of the registered user of the data application	No.	Requirement #4 is not supported. Conditionally yes if this combined with User consent solution.
5	The 3GPP SSO Service shall be able to interwork with Identity Management (IdM) specifications (e.g., OpenID [51])	Yes. Maybe needs to be better reflected in call flows.	Requirement #5 supported.
6	The 3GPP SSO Service shall support 3GPP SSO Authentication based on Operator-controlled credentials and policies	Yes. Since it is based on GBA/AKA.	Requirement #6 is supported.
7	The 3GPP SSO Service may support negotiation and use of an agreed authentication method between the UE and the 3GPP SSO Identity Provider	Partially.	Requirement #7 is partially supported. Use of GBA is agnostic to the Ua protocol used for authentication.
8	The 3GPP SSO Service may support mechanisms to ensure the presence of the registered user of the data application	No.	Requirement #8 is not supported. Conditionally yes if this combined with User consent solution
9	The 3GPP SSO Service may support mechanisms to…. Satisfy policies of the Data Application Provider	Yes, only if the local PI authentication is the DAP policy	Requirement #9 is partially supported.
10	3GPP SSO Service shall be transparent from a user perspective when the user accesses a data application using an identity created through a 3rd Party SSO Identity Provider	Yes. This is the main target of this solution.	Requirement #10 is supported.
11	The user shall be able to configure which 3rd party SSO identities are used with the 3GPP SSO Service	Yes. This is the main target of this solution.	Requirement #11 is supported

8 Conclusions

The present study has investigated and evaluated existing interworking solutions between SSO frameworks and 3GPP authentication mechanisms against the SA1 requirements in TS 22.101. New solutions have also been proposed. The objective was to identify if the SA1 requirements give rise to further specification work in SA3.

Issue 1 Presence of the registered user

Most of the investigation focused on SA1 requirement on confirming the presence of the registered user (clause 7.3). Since the current 3GPP defined SSO mechanisms authenticate the USIM and not the human user, a set of solutions for performing two-factor authentication (e.g., username/password as first factor and GBA-OpenID as second factor) were proposed. Username/password mechanisms, which are not using 3GPP credentials, are out of scope of 3GPP. However, most of the solutions proposed in this TR allow either 3GPP entities or non-3GPP entities to control the second factor credentials (e.g., username/password). At least some of the proposed two-factor authentication mechanisms could be implemented without any impacts to 3GPP specifications.

Also related to the SA1 requirement on confirming the presence of the registered user, a solution for ensuring user consent for using GBA for a service like SSO was proposed (clause 7.4). GBA, as specified today, does not provide means for confirming that the authorized user is present and gives consent for using GBA keys for an application. Specifying such a solution would require TS changes, e.g. in TS 33.220.

Issue 2 Authorization function in the UE

Clause 7.4 includes a proposal for functional architecture of an authorization function within the UE to perform the local user authentication. Such authorization function and mechanisms for providing authentication or authorization policies from the Service Provider/NAF to the authorization function in the UE or mechanisms for negotiating local user authentication capabilities could be regarded to be in scope of 3GPP, since such local credentials could be provisioned by operator, similar to currently used UICC credentials.

Issue 3 Third party user identities

Another investigated SA1 requirement was on transparency of 3GPP SSO service when using third party user identities and configuration of those identities. The requirement is met by the current 3GPP SSO mechanisms (especially GBA-OpenID), since third party user identities are enabled by USS (User Security Settings) of GBA. A solution for mapping the third party user identities to USS instead of using configuration (as proposed in clause 7.5) could be beneficial for using GBA with SSO frameworks. Specifying such solution would require TS changes, e.g. in TS 33.220.

Issue 4 Negotiation of authentication method

SA1 has a requirement which states "The 3GPP SSO Service may support negotiation and use of an agreed authentication method between the UE and the 3GPP SSO Identity Provider. The negotiation of an authentication method could be repeated each time the user accesses a DAP's service." This requirement is met by the existing standardized mechanisms between the NAF and UE to negotiate which GBA variant is to be used.

Issue 5: GBA-lite

Clause 7.2 describes an implementation option for co-locating NAF and BSF and simplifying some GBA functionality accordingly in order to enable a step-wise introduction of GBA.

Conclusion It is recommended that no further 3GPP work is conducted for Single Sign-On in the context of the present study. However, the findings of the present study could be useful in the context of other 3GPP activities, e.g. Mission Critical Push To Talk (MCPTT).

Annex A:
Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
2011-01					Skeleton and scope agreed in SA3 #62		0.1.0
2011-05					S3-110507 agreed in SA3 #63.	0.1.0	0.2.0
2011-12					Added S3-111056 agreed in email discussion after SA3 #65.	0.2.0	0.3.0
2012-12					Updated with modified version of S3-121158 after email approval after SA3 #69.	0.3.0.	0.4.0
2013-01					Updated with modified version of S3-130229 at SA3 #70.	0.4.0	0.5.0
2013-05					Updated after SA3 #71 due to email approval of S3-130570	0.5.0	0.6.0
2013-10					Updated after SA3 #72 due to email approval of S3-130703, S3-130704, S3-130705, S3-130706, S3-130707, S3-130723, and S3-130891.	0.6.0	0.7.0
2013-12					Updated after SA3 #73 due to email approval of S3-131031, S3-131082, S3-131204, and S3-131205.	0.7.0	0.8.0
2014-01					Updated at SA3 #74 due to approval of S3-140057and S3-140278.	0.8.0	0.9.0
2014-08					Updated at SA3 #76 due to approval of S3-142039 and S3-142040.	0.9.0	0.10.0
2014-12					Updated at SA3 #77 due to approval of S3-142486, S3-142500 and S3-142501.	0.10.0	0.11.0
2015-04					Updated at SA3 #79 due to approval of S3-151379, S3-151381, and S3-151483.	0.11.0	0.12.0
2015-06	*SA#68*	*SP-150293*			Presented for information and approval (MCC editorial corrections included)	0.12.0	1.0.0
2015-08					Updated due to comments from SA plenary in S3-152107.	1.0.0	1.1.0
2015-09	*SA#69*	*SP-150467*			Presented for approval	1.1.0	2.0.0
					Upgrade MCC	2.0.0	13.0.0

Change history							
Date	**Meeting**	**TDoc**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**New version**
2017-03	SA#75					Promotion to Release 14 without technical change	14.0.0
2018-06	-	-	-	-	-	Update to Rel-15 version (MCC)	**15.0.0**
2020-07	-	-	-	-	-	Update to Rel-16 version (MCC)	**16.0.0**

History

Document history		
V16.0.0	August 2020	Publication