# 3GPP TR 33.980 V11.0.0 (2012-09)

*Technical Report*

**3rd Generation Partnership Project;**
**Technical Specification Group Services and System Aspects;**
**Liberty Alliance and 3GPP security interworking;**
**Interworking of**
**Liberty Alliance Identity Federation Framework (ID-FF),**
**Identity Web Services Framework (ID-WSF)**
**and Generic Authentication Architecture (GAA)**
**(Release 11)**


Keywords
UMTS, GSM, interworking, security, architecture

***3GPP***

Postal address

3GPP support office address
650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet
http://www.3gpp.org

# 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.

# Introduction

3GPP defined the Generic Authentication Architecture (GAA) independent of the Liberty Alliance Identity Federation and Web Service Framework. Both systems were designed to be deployed independently of each other. The Liberty Alliance Identity Federation and Web Service Framework offers simplified sign-on and session management for complex web service business interaction protocols. The GAA offers a mechanism to provide a shared secret and certificates to two communicating entities for mobile applications, based on GSM and UMTS authentication and key agreement protocols.

# 1 Scope

The present document provides guidelines on the interworking of the Generic Authentication Architecture (GAA) and the Liberty Alliance architecture. The document studies the details of possible interworking methods between the Security Assertion Markup Language v2.0, SAML v2.0 (or alternatively the Liberty Alliance Identity Federation Framework, ID-FF), the Identity Web Services Framework (ID-WSF) , the Security Assertion Markup Language (SAML) and a component of GAA called the Generic Bootstrapping Architecture (GBA). This document only applies if Liberty Alliance and GBA or SAML v2.0 and GBA are used in combination.

# 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 TS 33.220: "Generic Authentication Architecture (GAA); Generic bootstrapping architecture".

[2] 3GPP TS 33.222: "Generic Authentication Architecture (GAA); Access to network application functions using Hypertext Transfer Protocol over Transport Layer Security (HTTPS)".

[3] 3GPP TS 33.221: "Generic Authentication Architecture (GAA); Support for subscriber certificates".

[4] 3GPP TS 24.109: "Bootstrapping interface (Ub) and network application function interface (Ua); Protocol details".

[5] 3GPP TS 29.109: "Generic Authentication Architecture (GAA); Zh and Zn Interfaces based on the Diameter protocol; Stage 3".

[6] Liberty Alliance Project, ID-WSF v2.0: "Liberty ID-WSF Security Mechanisms".

[7] Liberty Alliance Project, ID-FF v1.2: "Liberty ID-FF Architecture Overview".

[8] Liberty Alliance Project, ID-WSF v2.0 "Liberty ID-WSF Authentication Service Specification and Single Sign-On Service".

[9] Liberty Alliance Project, ID-WSF v2.0: "Liberty ID-WSF SOAP Binding Specification".

[10] Liberty Alliance Project, ID-WSF v2.0: "Liberty ID-WSF Discovery Service Specification".

[11] Organization for the Advancement of Structured Information Standards (OASIS), SAML v2 Core "Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0".

[12] Liberty Alliance Project, ID-FF v1.2: "Liberty ID-FF Bindings and Profiles Specification".

[13] Organization for the Advancement of Structured Information Standards (OASIS), "Profiles for the OASIS Security Assertion Markup Language (SAML) v2.0".

[14] Liberty Alliance Project, ID-WSF v1.2: "Security Mechanisms".

[15] Liberty Alliance Project Support Documents: "Authentication Context Specification" v2.0.

[16] Liberty Alliance Project, ID-WSF "Profiles for Liberty enabled User Agents and Devices".

[17] IETF RFC 2222 (1997), "Simple Authentication and Security Layer (SASL)".

[18] IETF RFC 2831 (2000), "Using Digest Authentication as a SASL Mechanism".

[19] IETF RFC 2617 (1999), "HTTP Authentication: Basic and Digest Access Authentication".

[20] Liberty Alliance Project Support Documents: "Liberty Reverse HTTP Binding for SOAP Specification" v1.1.

[21] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[22] IETF RFC 3546 (2003-06), "Transport Layer Security (TLS) Extensions".

[23] Liberty Alliance Project, ID-SIS: "Liberty Alliance ID-SIS 1.0 Specifications".

[24] IETF RFC 2246 (1999-01), "The TLS Protocol Version 1.0".

[25] IETF RFC 4279 (2005-12), "Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)".

[26] Liberty Alliance Project, ID-FF v1.2: "Liberty ID-FF Protocols and Schema Specification".

[27] Organization for the Advancement of Structured Information Standards (OASIS), "Authentication Contexts for the OASIS Security Assertion Markup Language (SAML) V2.0".

[28] Organization for the Advancement of Structured Information Standards (OASIS), SAML v2 Core "Conformance Requirements for the OASIS Security Assertion Markup Language (SAML) V2.0".

[29] 3GPP TS 23.003: "Numbering, addressing and identification".

# 3 Definitions, symbols and abbreviations

## 3.1 Definitions

For the purposes of the present document, the terms and definitions given in TR 21.905 [21] and the following apply.

**Assertion (SAML assertion)** is an XML-based data structure defined by SAML v2.0 [28]. Assertions are collections of one or more statements made by a SAML authority (also known as an issuer), such as an authentication statement or attribute statement. As used in Liberty, assertions typically concern things such as: an act of authentication performed by the Principal, attribute information about a Principal, or an authorization permission applying to a Principal with respect to a specified resource.

**Attribute:** A distinct, named characteristic of a Principal or other system entity.

**Bootstrapping Server Function (BSF):** A BSF is hosted in a network element under the control of an MNO. BSF, HSS, and UEs participate in GBA in which a shared secret is established between the network and a UE by running a bootstrapping procedure. The shared secret can be used between NAFs and UEs, for example, for authentication purposes.

**Defederate (federation termination):** To eliminate the linkage between a Principal's account at an identity provider and a service provider.

**Discovery Service (DS):** An ID-WSF service facilitating the registration, and subsequent discovery of, ID-WSF service instances, as indexed by Principal identity [10].

**Federation:** A is an act of establishing a relationship between two entities or an association compromising any number of service providers and identity providers.

**GBA Function:** A is a function on the ME executing the bootstrapping procedure with BSF (i.e. supporting the Ub reference point) and providing Ua applications with a security association to run bootstrapping usage procedure. The GBA function is called by a Ua application when the Ua application wants to use the bootstrapped security association.

**Identity Provider (IdP):** A Liberty-enabled system entity that manages identity information on behalf of Principals and provides assertions of Principal authentication to other providers e.g. other service providers.

**Liberty-Enabled User Agent or Device (LUAD):** A device (or user agent) that has specific support for one or more profiles of the Liberty specifications. A LUAD may perform one or more Liberty system entity roles as defined by the Liberty specifications it implements. For example, a LUAD LECP is a user agent or device that supports the Liberty LECP profile, a LUAD ECP is a user agent or device that supports the SAML v2.0 ECP Profile and a LUAD-DS would define a device or user agent offering a Liberty ID-WSF Discovery Service [10].

**Liberty Identity Federation Framework (ID-FF):** A system that enables identity federation and management through features such as identity/account linkage, simplified sign on, and simple session management.

**Liberty Identity Web Services Framework (ID-WSF):** A system that provides the framework for building interoperable identity services, permission based attribute sharing, identity service description and discovery, and the associated security profiles.

**Network Application Function (NAF):** A NAF is hosted in a network element. GBA may be used between NAFs and UEs for authentication purposes, and for securing the communication path between the UE and the NAF.

**Principal:** A principal is a system entity whose identity can be authenticated. In Liberty usage the term Principal is often synonymous with "user". The Principal is the legitimate user of the UE.

**Service Provider (SP):** A SP is a role donned by system entities. The SP interacts with other system entities primarily via plain HTTP. From a Principal's perspective, a Service Provider is typically a web site providing services and / or goods.

**Web Service:**

1. A service defined in terms of an XML-based protocol, often transported over SOAP, and / or a service whose instances, and possible data objects managed therein, are concisely addressable via URIs.

2. A web service utilizing [9], [6] and [10].

**Web Service Consumer (WSC):** A WSC is a role donned by a system entity when it makes a request to a web service.

**Web Service Provider (WSP):** A WSP is a role donned by a system entity when it provides a web service.

## 3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply (origin of term if GAA or LAP/SAML):

| | |
|---|---|
| AS | Authentication Service (as defined by LAP) |
| BSF | Bootstrapping Server Function (GAA) |
| B-TID | Bootstrapping Transaction Identifier (GAA) |
| DS | Discovery Service (as defined by LAP) |
| ECP | Enhanced Client or Proxy (as defined by SAML) |
| FQDN | Fully Qualified Domain Name |
| GAA | Generic Authentication Architecture |
| GBA | Generic Bootstrapping Architecture (GAA) |
| GSID | GAA Service Identifier (GAA) |
| GUSS | GBA User Security Settings (GAA) |
| HSS | Home Subscriber Server |
| ID-FF | Identity Federation Framework (as defined by LAP) |
| ID-SIS | Identity Service Interface Specification (as defined by LAP) |
| IdP | Identity Provider (as defined by LAP/SAML) |
| ID-WSF | Identity Web Services Framework (as defined by LAP) |
| LAP | Liberty Alliance Project |
| LECP | Liberty-Enabled Client or Proxy (as defined by LAP) |
| LUAD | Liberty-Enabled User Agent or Device (as defined by LAP) |
| NAF | Network Application Function (GAA) |
| PAOS | Reversed HTTP binding for SOAP (as defined by LAP/SAML) |
| SAML | Security Assertion Markup Language |
| SASL | Simple Authentication and Security Layer |
| SOAP | Simple Object Access Protocol |
| SP | Service Provider |
| SSO | Single Sign-On |

| | |
|---|---|
| SSOS | SSO Service |
| UE | User Equipment |
| UID | User Identifier |
| USS | User Security Setting |
| WSC | Web Service Consumer (as defined by LAP) |
| WSP | Web Service Provider (as defined by LAP) |

# 4 Interworking of Liberty Alliance ID-FF/ ID-WSF and Generic Authentication Architecture

## 4.1 Introduction

This document describes the interworking of GBA and the Liberty Alliance Project framework. Such interworking translates into a combined use of both frameworks by a given user. This interworking guideline may result in profiling GAA and the Liberty Alliance Project Specifications for interworking purposes or may propose extensions. The deployment of the GAA system entities and of the Liberty Alliance system entities must not be dependent on each other. Thus this guideline does not interfere with any deployment of GAA or Liberty Alliance entities where both are not interworking.

## 4.2 Architectural Description – Use of GBA within ID-FF / ID-WSF

This clause describes the GAA and ID-FF / SAML v2.0 / ID-WSF architecture. The GAA system consists of UE, BSF, NAF, and HSS (and Zn-Proxy dependent on configuration) as described in TS 33.220 [1].

In the Liberty Alliance architecture the following system entities exist: Principal (shown as UE in the figures), IdP, DS, SP, and the roles WSC, and WSP. Typical Liberty Alliance network models are shown for ID-FF in Figure 4.2.-1 and for ID-WSF in 4.2.-2.

As SAML v2.0 [28] was specified with ID-FF 1.2 taken as an input, SAML v2.0 is a superset of ID-FF 1.2 and SAML v1.1 with some relatively small differences (mostly extensions). The related system entities are: UA, SP and IdP (User Agent, Service Provider and Identity Provider, respectively). For this strong similarity, no separate discussion on SAML v2.0 is given in this section unless necessary. However, as SAML v2.0 has formally superseded ID-FF 1.2, it is recommended that the solutions implementing the interworking functionality described in this TR are based on SAML v2.0, rather than ID-FF v1.2.

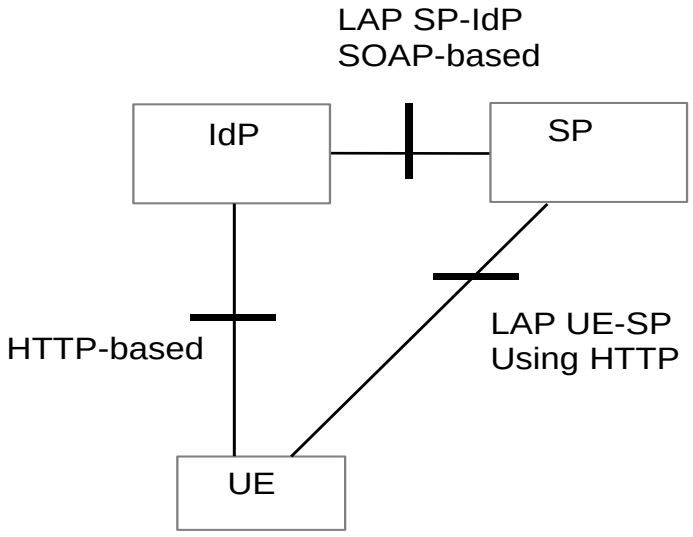


**Figure 4.2-1: Liberty Alliance network model for ID-FF**

For easy integration in current web deployment, some variants of ID-FF do not use the SOAP-based connection between IdP and SP (as shown e.g. in figure 4.2-1), but rely solely on HTTP-based connections originating in UE. Regarding SAML v2.0, the Web Browser SSO Profile [13] is used.

Regarding GAA/GBA interworking with Liberty ID-FF, in principle Liberty ID-FF Identity Provider (IdP) Specification [7] is the only specific ID-FF service that it is relevant for the discussion regarding authentication interworking.

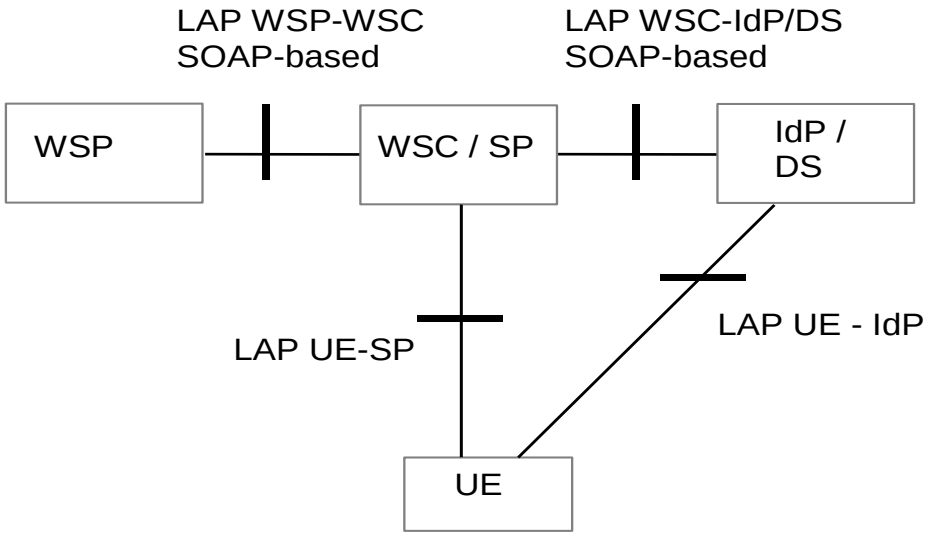


**Figure 4.2-2: General Liberty Alliance network model for ID-WSF**

Regarding GAA/GBA interworking with Liberty ID-WSF, in principle Liberty ID-WSF Authentication Service (AS) Specification [8] is the only specific ID-WSF service that it is relevant for the discussion regarding authentication interworking. Liberty Alliance specifies the AS as part of the IdP in ID-WSF taking the authentication function in ID-WSF. This is in contrast to ID-FF, where the authentication function is not a separate service within IdP. First it is outlined, how the Liberty ID-WSF Authentication Service fits together with the GBA architecture, then the more complex scenario that includes a Single Sign On Service and an Authentication Service is described.

The typical Liberty ID-WSF attribute sharing infrastructure including WSC, WSPs and DS does usually not interwork with GAA/GBA. A WSC would request end user attributes from a WSP and all the required security aspects would be supported by the DS.

Liberty ID-WSF "Authentication Service and Single Sign-On Service Specification" [8] describes procedures so that:

1. A user authenticates to an AS using SOAP-based interface;
2. A user requests a security token to access a particular SP;
3. A user presents the received security token to the SP.

This procedure is described in clause 4.3.5 and does not require any further interaction with WSCs, WSPs or DSs. The Liberty ID-WSF Authentication Service may also be used by WSCs to be able to interact with a DS (e.g. when a Liberty ID-FF infrastructure is not available and a WSC needs to interact with a DS in order to discover user attributes). Here the DS would act as a SP that needs to authenticate the WSC. This would be an entity peer authentication rather than a GBA/GAA based end-user authentication. Thus the only potential for interworking between the ID-WSF Authentication Service and GAA/GBA is where a Liberty implementation of a WSC in a User Equipment (i.e. a Liberty User Agent or Device, LUAD-WSC) wants to get access to a SP (e.g. a DS or any other SP). Therefore, the roles and architecture elements relevant are described in figure 4.2-3.

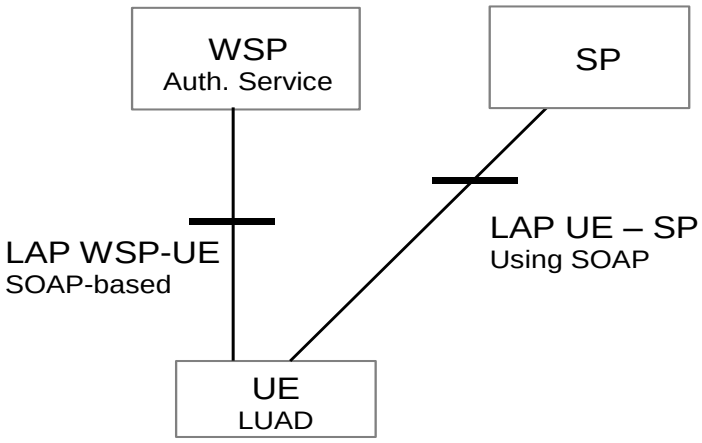


**Figure 4.2-3: Liberty Alliance network model for ID-WSF Authentication Service**

The Liberty Alliance Architecture might also not only contain an Authentication Service (AS), but also a separate Single Sign On Service (SSOS) that interacts directly with an SP. The AS provides security tokens to the UE which may be used with all services offered in the domain of the same provider. The scenario with SSOS is necessary when either the communication between UE and SP may by some reason only be based on ID-FF protocols, or if the service is offered by some other provider. The network model for this scenario is depicted in 4.2-4:

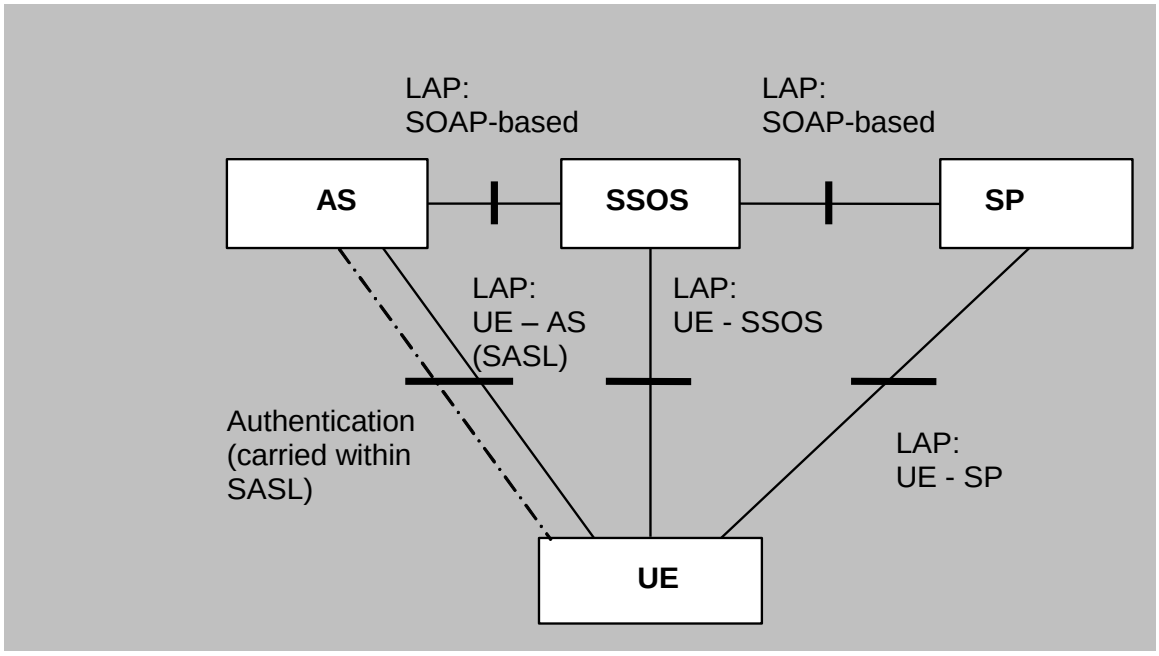


**Figure 4.2-4: Liberty Alliance network model for ID-WSF Authentication Service with Single Sign On Service**

NOTE 1: The dashed line indicates the authentication which is out of scope of Liberty Alliance ID-FF and ID-WSF specifications. The solid lines and boxes indicate Liberty Alliance reference points and elements.

The scenarios where the GBA architecture is combined with the ID-WSF AS have the following interworking elements:

- For the UE: UE comprises both GBA and LAP functionality and thus has Ub interface to BSF.
- For the AS: AS contains authentication functionality and thus has to interwork with GBA. Details depend on the actual collocation of elements and are given in the following sub-clauses.

The reference point between UE and AS is affected in this scenario, as can be seen e.g. in Figure 4.2-4. The reference point between UE and AS utilizes the Simple Authentication and Security Layer (SASL) protocol (RFC 2222 [17]) as authentication support layer according to Liberty Alliance specifications.

The UE-AS reference point may utilize digest authentication as a SASL mechanism (RFC 2831 [18]). This would be a specific implementation of the Ua protocol similar to TS 33.222 [2]. The protocols could use the shared secret of GBA (Ks_NAF) for authentication, e.g. digest-MD5 or other authentication methods within SASL

NOTE 2: There are further interworking cases possible, but all require more new specifications or adaptations of existing specifications compared with the above-mentioned way. In particular, one case stands out where the AS acts as BSF. Then a version number information of the used AKA protocol must be transported within SASL, but this would no longer fall within the realm of GAA/GBA. There would be no Ub and Ua protocols as specified in TS 33.220 [1], but only a straight-forward use of e.g. digest AKA within SASL for authentication. All other features of GBA would not be used.

The Liberty-specific interfaces are secured using methods described in [14] and [6]. There are several possibilities for the UE interfaces towards Liberty entities e.g. pure HTTP-based or PAOS-based [20]. For ID-WSF, the reference points between the UE and the SP, respectively the UE and the IdP might also be SOAP-based.

For a mobile network operator deploying 3GPP GBA system and the Liberty ID-FF or ID-WSF, there are two alternative architectures possible. The Liberty Authentication function might be collocated with the NAF, or it might be collocated with the BSF and the SP collocated with the NAF. For ID-WSF, the reference points between the UE and the SP, respectively the UE and the IdP might also be SOAP based. These alternative architectures are discussed in the following sub-clauses.

In practice one or the other collocation scenario will be deployed, depending on the specifics of the operator's network and on its business requirements. The following could be used as a guideline which collocation setting to use:

1. If the operator wants to play the role of both, a BSF and an IdP, having both similar roles, it seems natural to collocate them (optimized network flows and optimized costs for the operator).

2. If the operator wants to play the BSF role, but the IdP role will be played by somebody else (being that for example another organization inside the operator, or a 3rd Party), in this case it may not be possible to collocate IdP and BSF but still there is a possibility to reuse GBA-based strong authentication performed by the collocation of IdP and NAF.

## 4.2.1 Architecture for collocation of NAF with Liberty Alliance Authentication Function

Interworking of GAA and LAP/SAML applies only to the authentication used within LAP/SAML. Thus the machinery provided by GAA is a natural extension to a Liberty Alliance ID-FF Identity Provider (IdP), a SAML Identity Provider (IdP) or a Liberty Alliance ID-WSF Authentication Service (AS). The following sub-clauses handle the LAP/SAML IdP and the LAP AS cases separately.

NOTE: Interworking of GAA and Liberty Alliance or SAML is independent of any other deployment of ID-FF, SAML or ID-WSF. Only the type of communication between UE and the network element responsible for authentication is relevant.

If the subscriber's home operator does not host the NAF described hereinafter, then the architecture also includes a Zn-Proxy as described in TS 33.220 [1].

### 4.2.1.1 Collocation of IdP/NAF in Liberty Alliance ID-FF (alternatively SAML v2.0)

If the IdP is collocated with the NAF, then the IdP/NAF authenticates the UE using GBA credentials. There is only one reference point carrying both Liberty Alliance and GBA related information, i.e. the reference point between the IdP/NAF and the UE. The protocols and profiles, that are used to trigger the authentication of the UE and the successful authentication information transfer are defined in Liberty ID-FF [7] or SAML v2.0 [11], [13]. The architecture for a collocated IdP/NAF together with the Liberty ID-FF is outlined in Figure 4.2-5.

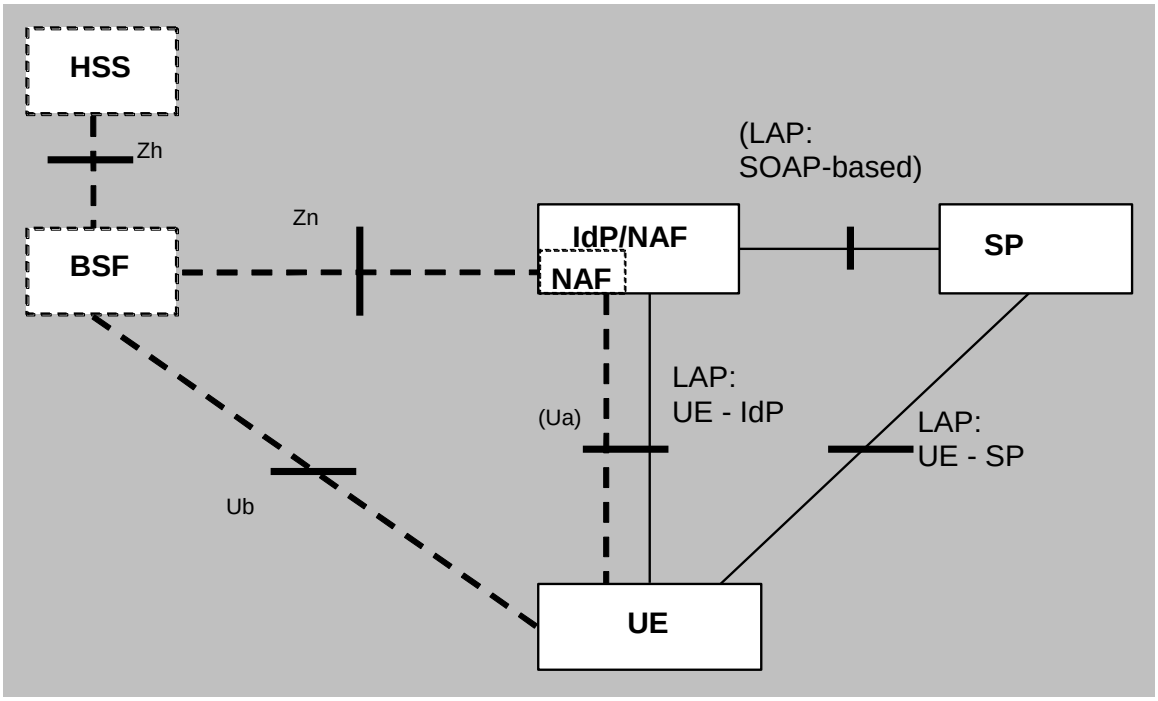


**Figure 4.2-5: Combined Liberty Alliance ID-FF and GAA architecture with collocated NAF and IdP.**

NOTE: The dashed lines and boxes are 3GPP reference points and network elements defined in TS 33.220 [1]. The solid lines and boxes indicate that these are Liberty Alliance reference points and elements.

Figure 4.2.5 shows a Liberty Alliance ID-FF environment on the right hand side. The same arrangement is valid if other Liberty Alliance network elements (except the UE) deploy ID-WSF protocols between them.

### 4.2.1.2 Collocation of AS/NAF in Liberty Alliance ID-WSF

If the GBA architecture is deployed together with the Liberty ID-WSF Authentication Service as described in 4.2-6 then the architecture is similar to the Liberty ID-FF case as depicted in Figure 4.2-5. The main difference is that the Ua reference point is a SOAP-based interface for the usage of the authentication service.

In principle, Liberty Alliance ID-FF and ID-WSF specifications do not care how authentication is performed. But if authentication is carried within the same communication path as the Liberty Alliance SOAP messages between UE and AS, then Liberty Alliance mandates the use of SASL [17] as wrapper for the authentication protocol. Guidance on the use of digest authentication [18] similar to the use within TS 33.222 [2] is given in Annex A.

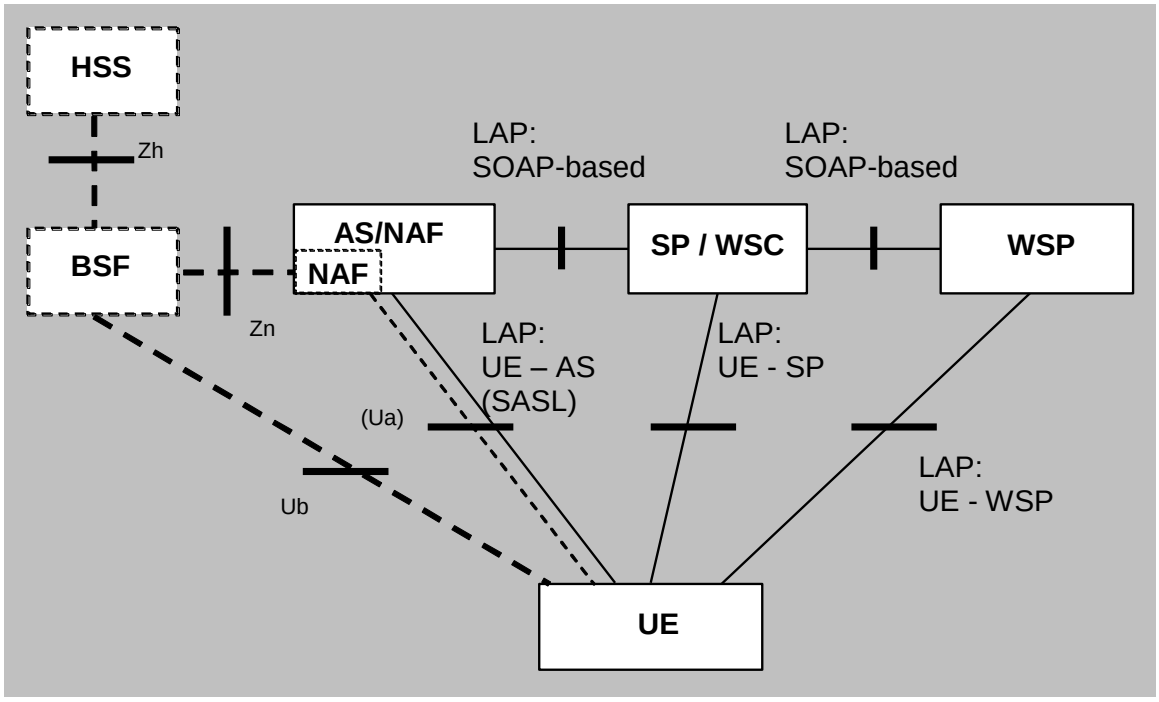


**Figure 4.2-6: Combined Liberty Alliance ID-WSF and GAA architecture with collocated NAF and AS**

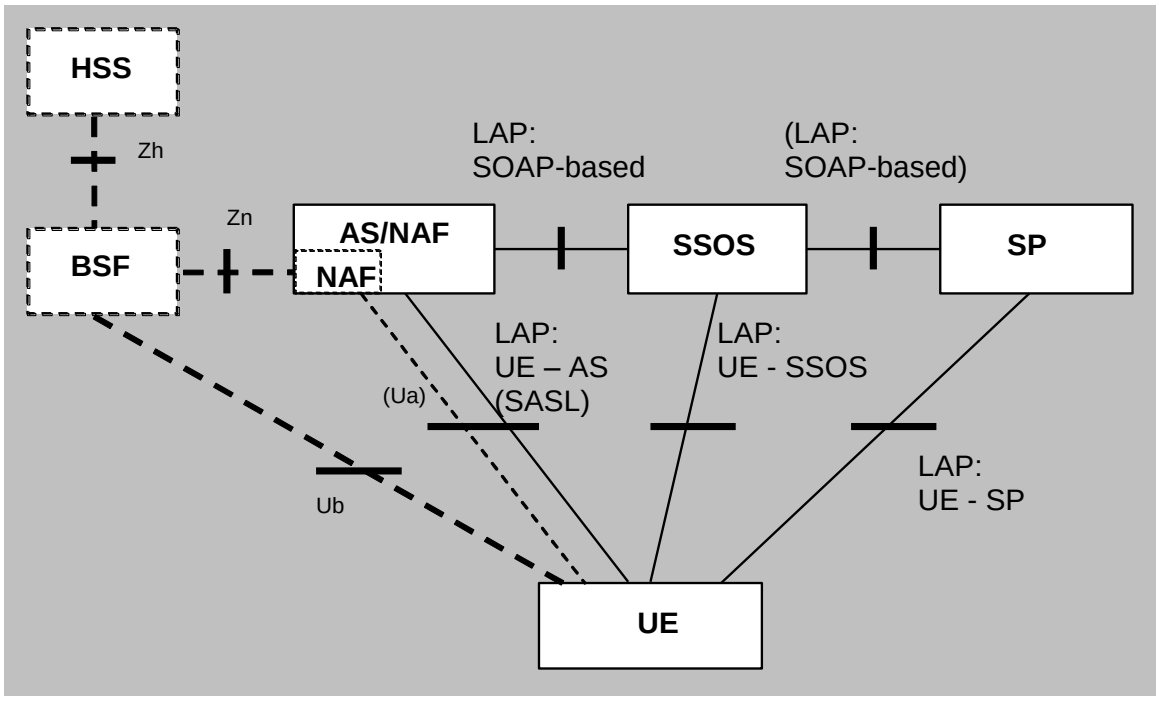


**Figure 4.2-7: Combined Liberty Alliance ID-WSF and GAA architecture with collocated NAF and AS and deployment of ID-FF for UE-SP communication**

The network model for GBA interworking scenario with the Liberty ID-WSF Authentication Service ,where the AS and SSOS are separate, is shown in Figure 4.2-7, According to the Liberty Alliance Project, ID-WSF: "Authentication Service and Single Sign-On Specification" [8], the SSOS may also be collocated with the AS, which can also be applied in Figure 4.2-4.

In an interworking scenario with the Liberty ID-WSF Authentication Service, a service provider that wants to request user authentication would redirect the user to the AS/NAF. This is depicted in Figure 4.2-7. The user will be authenticated in a first step to the AS/NAF for which GBA procedures may be used. The AS/NAF interaction with the BSF would be transparent to the SP. In a second step the user will request a security token from the Single Sign On

Service (SSOS) to be granted access to the particular SP. Finally the user will present the security token received from the SSOS to the SP which would analyze it and decide whether the user deserves access to the service or not.

### 4.2.2 Architecture for collocation of BSF with Liberty Alliance authentication function

If the IdP or AS (Liberty ID-WSF Authentication Service) is collocated with the BSF, then the IdP/BSF authenticates the UE by executing the Ub bootstrapping authentication procedure. In the same way as with the IdP/NAF collocation option with regard to the UE-IdP/NAF reference point, in this collocation option the reference point between the UE and the IdP/BSF carries not only the GBA bootstrapping procedure (Ub) but also Liberty/SAMLv2-related information as follows:

- In addition to the GBA bootstrapping authentication procedures carried over the UE-IdP/BSF reference point, SAML v2.0/Liberty ID-FF v1.2 related information information (e.g. SAMLv2/ID-FF Web Browser SSO protocol messages, if such profile is used) shall is also be carried. Thus, the protocols used to trigger the authentication of the UE (by using the Ub bootstrapping authentication procedure) and the transfer of authentication information will be those defined in LAP ID-FF / SAML v2.0. It is important to highlight that the transfer of information over different protocols in the UE-IdP/BSF reference point does not require any modification on the actual GBA bootstrapping procedures as defined in relevant GBA specifications (as with regard to Ua and LAP/SAML procedures over the UE-IdP/NAF reference point). The interaction between the UE and the IdP/BSF is defined by the composition of SAMLv2/LAP ID-FF and GBA messages over such interface (in SAML parlance, this would be typically referred to as a "profile").

- In addition to Liberty related information carried over the UE-IdP/BSF reference point, the IdP/BSF shall be able to trigger the execution of the standard GBA bootstrapping procedure when an authentication request is received by means of LAP ID-FF v1.2 / SAML v2.0 procedures. This shall be done by sending back a bootstrapping required indication to the UE.

  Note that, according to the current GAA/GBA specifications, this feature is typically implemented by NAFs, that is, upon indication from the NAF, the UE starts a Ub bootstrapping procedure, by accessing the BSF. However, in this case, it is required that the IdP/BSF entity makes use of such protocol fragment, as part of the authentication process of the IdP. The use of this indication by the IdP/BSF does not represent a change in the Ub bootstrapping procedure, as used by standard BSFs.

- If artefact transfer is supported, an additional SOAP based reference point to service providers shall be provided by the IdB/BSF..

The protocols and profiles that are used to trigger the authentication of the UE and the successful authentication information transfer are defined in SAML v2.0 [11], [13], or Liberty ID-FFv1.2 [7]. The architecture for a collocated BSF together with a SAMLv2/Liberty ID-FF IdP is outlined in Figure 4.2-8

NOTE: In Liberty Alliance the IdP or AS does not need, in general, to belong to the same organizational domain as the key provisioning entity. A collocation of the BSF with the IdP or AS actually materializes the scenario where the IdP/BSF or AS/BSF actually belongs to the operator trust domain.

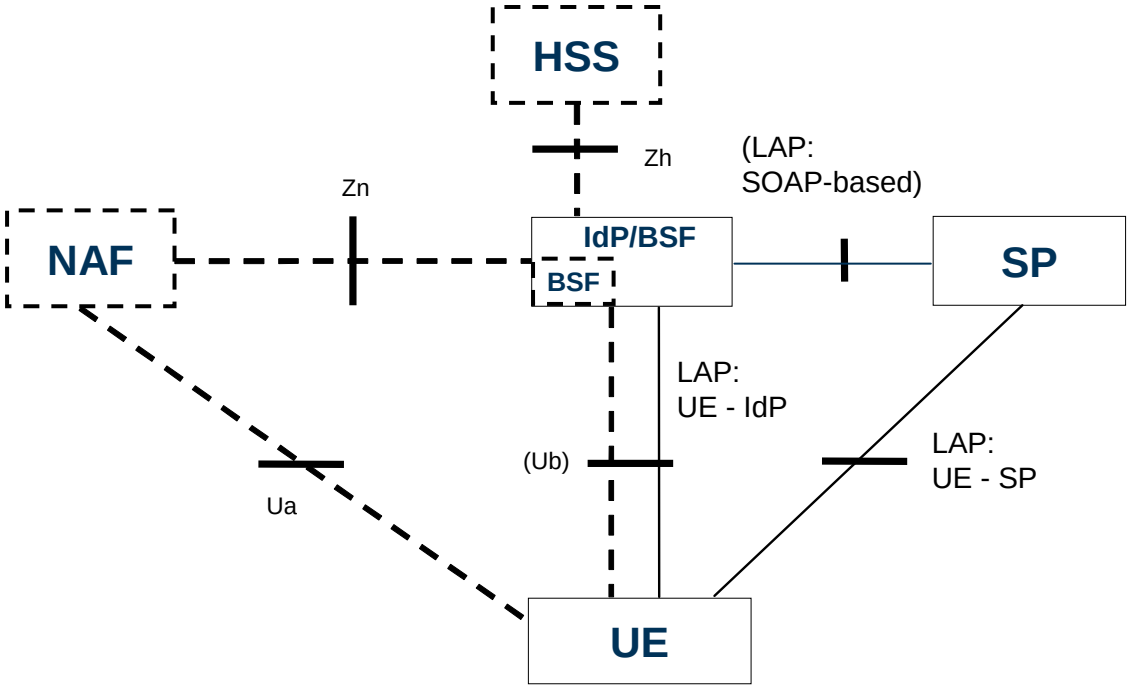


**Figure 4.2-8: Combined Liberty Alliance ID-FF and GAA architecture with collocated BSF and IdP.**

Clause 4.3a outlines details for GBA interworking with the SAMLv2/Liberty Alliance ID-FF specifications when BSF and Identity Provider are collocated.

### 4.2.2a Logical data model of the Liberty Alliance Authentication Function (IdP/AS)

The main role of an IdP/AS is to get user authenticated and share such authentication event information with the service providers belonging to the Circle of Trust. In addition to that, per service provider identifiers may also be delivered to SPs. Therefore, user profiles handled by IdP/ASs comprise, at least, a user identifier that allows to link the authentication procedure with the user profile and also, the per service provider identifiers that may apply to each service provider. Such user profiles will be identified by means of one or several indexes.

It can be assumed that in interworking scenarios, one of the indexes used to locate the profile of a given user at the Liberty Alliance Authentication Function is available in the GBA environment as a user identity. Therefore, upon reception of such identity, the IdP/AS will look up the appropriate user profile from its storage.

### 4.2.3 User Registration to Interworking Service

Participation of a given user in GAA-LAP interworking requires the Liberty Alliance authentication function (IdP/AS) in ID-FF and ID-WSF to get knowledge of some persistent subscriber-specific data. Most LAP-specific data may be stored persistently in the IdP (c.f. clause 4.3 and 4.3a). However, two kinds of persistent data need to be stored in the GBA environment:

- persistent user identity that allows the IdP to locate the appropriate user profile (IMPI or UID, e.g. IMPU or IdP/AS-specific pseudonym) unless anonymous user access is used, only when IdP and NAF are collocated (c.f. clause 4.3.1), and

- data which should be under control of the operator, e.g. access rights to IdP or authorization flags.

These data are not provided by the UE to the IdP (except in the IdP/BSF scenario - since the IMPI is always present in the IdP/BSF as it is provided by the UE in the GBA bootstrapping procedure - and when the IMPI is used as persistent user identity in a) above). Therefore, the addition of persistent user-specific data at the GBA environment is necessary. The GUSS in the HSS may be used to store and transfer this additional persistent user-specific data, a) and b) above. Such data are transferred through the Zh reference point to the BSF and, if the IdP/NAF co llocation option is used, from the BSF to the IdP/NAF through the Zn reference point.

Conceptually the user registration to GAA-LAP interworking may be subdivided into two parts:

- registration with general LAP usage at some IdP (the user may have been previously registered to the IdP), and
- registration with the GAA-LAP interworking service at that IdP

This distinction facilitates the deployment of GAA-LAP interworking in case the user has an ongoing contract with some LAP IdP or a preference for a certain LAP circle of trust.

#### 4.2.3.1 Registration with Operator

Participation of a given user in GAA-LAP interworking may be initiated by different procedures:

- Each user explicitly subscribes to GAA-LAP interworking. This may be accompanied by provisioning of user-specific and/or user selected data in the relevant entities: HSS, IdP/NAF, IdP/BSF or AS.
- Each subscriber being able to use GBA is automatically provided with access to GAA-LAP interworking. This requires either to provide each subscriber on start of subscription with interworking-specific data (at least a UID, e.g. IMPU or IdP/AS-specific pseudonym in USS), or alternatively to only use data for interworking which is existing for the subscriber anyhow (e.g. IMPI).

  NOTE: In the IdP/NAF collocation scenario, interworking specific identities may be created upon subscription to the interworking service (specific identities for the interworking service) or existing identities may be used instead.

For provisioning of interworking specific data stored with the operator see clause 4.2.4. To allow clear separation of data, LAP-specific data used in LAP environment only should not be stored in HSS with the operator.

#### 4.2.3.2 Registration with IdP

The registration to participate in LAP federation in general may be independent of the usage of the interworking service. Thus also all data used only within LAP framework is more suited to be provisioned and stored with the IdP that is taking care of all federation and further LAP tasks anyway.

Registration with IdP may be not bound by organisational means to registration with the operator. Otherwise, and depending on the collocation option, different tasks have to be carried out:

- When IdP/NAF collocation is used, then the user has to indicate to IdP the GBA-related identity (IMPI or UID) which shall be used to identify him/her at the IdP and therefore play the role of link between LAP, IdP and the GBA environment. In addition the user has to prove to the IdP that he is entitled to use the GBA-related identity. This is best accomplished by using a GBA-authenticated communication for the registration procedure at IdP/NAF also, as this proves the legitimate use by the current user of the GBA-related identity, which is provided to IdP/NAF via the Zn reference point. In case the USS transferred to IdP/NAF contains multiple public user identities, the user may indicate which identity (or identities) shall be used for LAP authentication.

  NOTE 1: When the Liberty Alliance Authentication Function is deployed, it shall be configured to know which type of GBA-related identity it has to use (either an IMPI or an UID).

  NOTE 2: When the user is asked to introduce a GBA-related identity at the IdP to identify him/her, the user shall be previously instructed to introduce the type of identity that is expected, over the Zn reference point, by the IdP/NAF.

- When the BSF is collocated with the IdP, the IMPI is always made available to the IdP/BSF via the Ub interface. Only if the IdP/BSF has been configured to use as user identity an identity other than the IMPI, it shall require that the user indicates which identity shall be used.

### 4.2.4 Provisioning of User Data for Interworking Service

Provisioning of user data at the (mobile) operator may be done in different ways. A general view of data stored in GUSS is given in clause 4.4.

#### 4.2.4.1 Service based on standard user data

If no user-specific interworking data is used for GAA-LAP interworking then the subscriber may start using the interworking functionality at any time.

This requires the interworking to be based on data which exists for the subscriber due to the underlying subscription. The only data generally available to BSF , since it is provided by the UE over the Ub reference point, is the IMPI (which can be used directly by the IdP if IdP/BSF collocation is used or tranferred to IdP/NAF). When IdP/NAF collocation is usedas the BSF may be configured by local policy to transfer the IMPI to a NAF over Zn, this is completely viable (c.f. TS 33.220 [1]). Only privacy considerations may apply in this case with respect to the trustworthiness of the IdP/NAF, as it gets knowledge of the IMPI of the subscriber.

#### 4.2.4.2 Service based on pre-provisioned interworking data

Each subscriber may be pre-provisioned with GAA-LAP interworking specific data on start of (mobile or IMS) subscription. This requires that for each subscriber at least one USS for the GAA-LAP interworking service is created (when IdP/BSF collocation is in place, this requirement is not always needed, since, if the IMPI is used at the IdP as user identity, no USS is needed).

This specific USS contains at least a persistent identity (UID) for use by the IdP. This shall be a public user identity, e.g. an IMPU, either generally used by the subscriber, or used specifically as a pseudonym for interworking with LAP.

In case authorization flags are specified for the interworking service, these may be set to some default values. These may depend on data available at time of subscription, e.g. type prepaid or postpaid.

If NAF groups are deployed by the operator, appropriate data elements have to be added to USS.

#### 4.2.4.3 Service based on explicitly added interworking data

On start of usage of GAA-LAP interworking (or on explicit subscription to this service) each user may be provisioned with specific interworking data. This may be done in addition to data provisioned according to clause 4.2.3.2, or as only data provisioning for GAA-LAP interworking service. Location for storage of this persistent data is also the GUSS in HSS.

NOTE: The storage and management of user service specific data may be done using operator specific means.

As it is anticipated that this type of provisioning may be more dependent on user needs, user selected UIDs or subscription specific authorization flags may be set. Also e.g. additional IdPs may be subscribed to, as GAA-LAP interworking is not restricted to one IdP only (when it comes to the IdP/NAF collocation). Different public identities (pseudonyms) for different IdP/NAFs are possible, distinguished in USS by NAF group.

Also a set of UIDs (bound to the same IMPI) may be stored in USS, allowing the user to indicate an intended identity (selected from the set of UIDs) on communication to IdP. Transfer of this intended identity between user and IdP is outside the scope of this document.

## 4.3 Co-hosting of NAF and IdP

In this clause it is assumed that the GBA NAF contains a Liberty IdP as defined in [7]. The creation of the authentication and re-authentication credentials is handled by GBA.

NOTE: When the UE contacts the IdP/NAF with a valid B-TID from an earlier bootstrapping run, then the NAF can have its local policy that can be stricter than the BSF policy, when to require a new bootstrapping run [1].

The GBA procedure is triggered by IdP/NAF as defined in TS 33.220 [1]. All [6] and [7] specific tasks are fulfilled by the IdP implementation in the NAF, this is transparent to the GBA function in the UE.

This clause also applies to the case where GAA interworks with Liberty Alliance ID-WSF. In this case the AS/NAF as part of IdP takes the role of the IdP/NAF in ID-FF. For the sake of brevity only IdP/NAF is mentioned in the following text.

### 4.3.1 Federation Concept in GBA

The Liberty Alliance has the concept of federating Principal identities together. This act of establishing a relationship between two entities requires a mapping. To map the GBA credential information and the Liberty Alliance information the NAF/IdP must maintain a table. In the case of non-anonymous access the IdP/NAF has two options how to label the user table:

- IMPI. Then the BSF must be configured always to send the IMPI to this NAF/IdP upon receiving the B-TID if the NAF/IdP is fully trusted by the BSF. The IMPI is used as a persistent user identifier.
- UID. The UID may be the IMPU. Then the NAF must insert the GSID into the request over the Zn reference point [5] to request the USS and then extract the UID from the USS. The UID is used as a persistent user identifier.

The IMPI or UID will be used as a permanent user identifier for the table. The table stores also the user's B-TID, key lifetime data, key generation time and the corresponding service related opaque handles (service specific user identifiers). The service specific user identifiers should be different for each service to ensure the user's privacy. This table might also contain the NAF specific key material, USS and further service provider related data. The table should logically separate temporary GBA related data i.e. B-TID, key, expiry time, bootstrapping time from the IdP related data and persistent data e.g. SP related data, SP name, user identifier for this SP, opaque handle, USS etc. The temporary GBA data shall be deleted on key expiry or Liberty session expiry. The IdP related data, and the persistent user identifier are persistent. The USS may be deleted upon defederation.

If the user is allowed to use the service anonymously, then the user is an authorized GBA participant. In this case, the B-TID is used as a temporary user identifier for the table. The federation then lasts as long as the Liberty session and the maximal length of the federation is the key lifetime. Since the whole table is of temporary nature the GBA related data in the table will be deleted, if the B-TID expires or the session is terminated. If the federation is terminated and the B-TID is still valid, then only the opaque handle and service provider related information should be deleted. In this anonymous user case, the whole table is of temporary nature. The table consists of two logically separate data blocks: Liberty service provider related data and GBA related data. For the anonymous access case, the Liberty service provider related data will be deleted upon termination of federation and the GBA related data upon session termination or expiration of key lifetime.

NAF/IdP can manage defederation (termination of the federation) by deleting the opaque handles and service provider related information out of the table. This may apply to single SPs or to all federations. The NAF could still then use the B-TID in GBA-based applications. The UE informs the NAF/IdP about the defederation using [12]. The NAF/IdP may also trigger the defederation, e.g. in case the service agreement with the SP ends or the user's subscription ends. In case of subscription end, the whole table should be deleted. The notification to the NAF/IdP of the termination of the subscription is out of the scope of this document.

If the B-TID expires and the user wants to use a GBA-based service then the NAF/IdP may, depending on the NAF policy, trigger a new bootstrapping run and update the B-TID or may delete the B-TID, related key material and key information.

### 4.3.2 Session Concept at IdP

The session concept of Liberty Alliance is mapped to the key lifetime of the NAF-specific key material. The maximum Liberty Alliance session lifetime must be equal to or shorter than the remaining lifetime of the key. When the Liberty session expires the temporary GBA related data is deleted from the table described in 4.3.1. This implies that if a key expires, then either the login session also expires (if no successful re-authentication has taken place). Then the related sessions should be terminated using Single Logout. If a session is explicitly terminated e.g. via Single-Logout, then the temporary GBA related data is deleted in the NAF/IdP. For the next login, the UE would be required to execute the bootstrapping usage procedure again, since he has no shared keys with the NAF/IdP. If a new bootstrapping procedure was executed since the last contact between UE and NAF, the new temporary GBA related data is inserted into the table described in 4.3.1. If the freshness of the received key material is not satisfactory, then NAF/IdP sends a re-negotiation request to the UE as outlined in TS 33.220 [1] and uses the new key material for the Liberty session.

When a user starts a Liberty session with the IdP, then it contacts the IdP via Ua reference point and mutual authentication as outlined in [2] is done. Depending on the entries in the table of the IdP, three possibilities exist:

1) In case the B-TID exists in the table and is not expired, the IdP has all required data and can start communication with the UE without communication over Zn. If the IdP decides that the remaining lifetime of the B-TID is too short, it may indicate bootstrapping re-negotiation required to the UE. Then the procedure is similar to case 2.

2) In case the B-TID does not exist in the table, and the USS received over Zn contains a user identity which does already exist in the table, then the entry in the table is updated with B-TID and related information.

3) In case the B-TID does not exist in the table, and the USS received over Zn contains a user identity which does not exist in the table or there is no user identity sent, then the IdP creates a new entry in the table.

This could be applied to a BSF/IdP and a NAF/IdP solution.

For anonymous user access, the B-TID is used as the user identifier. If such an anonymous Liberty session is terminated, then all the GBA related data is deleted, including the B-TID.

Liberty Alliance has the concept of authentication time. In GBA the bootstrapping time is available to the IdP/NAF. Since the bootstrap procedure requires Digest AKA, the bootstrapping time should be taken as Liberty authentication time.

If a user with ongoing LAP IdP session contacts the LAP IdP for authentication, and the <lib:AuthnRequest> contains the element <ForceAuthn> (cf. [26], section 3.2.1.1), then the IdP shall send to the user a Bootstrapping Renegotiation Request according to section 4.5.3 of [1]. This is necessary as this may be a reauthentication request issued for liveness validation within LAP (cf. [7], section 4.4.2), requiring a new bootstrapping, as the bootstrapping time is taken as Liberty authentication time. If the <ForceAuthn> fails, then the NAF/IdP should get a corresponding notification. The NAF/IdP may trigger then a Single Logout.

## 4.3.2a Single-Logout Concept

In this section, we describe briefly the procedure involved in a Single Logout process triggered by the user:

1. The user contacts the SP to perfomr a Single Logout.

2. The SP performs an HTTP redirect to the NAF/IdP.

3. The NAF/IdP may redirect the UE to other SPs which have an ongoing session with the user (the NAF/IdP has to keep track to which SPs the user was previously redirected during the lifetime of the current key). The NAF/IdP should provide the UE with his address for the case, that the redirect back in step 4 fails.

4. If redirected the UE performs a logout at the second SP and is redirected back to the IdP/NAF.

5. The NAF/IdP redirects the user back to the first SP to finalize there the logout procedure and then deletes the Ks_(ext/int)_NAF keys and related data.

6. UE performs logout at SP.

The logout procedure should be done using SAML 2.0 logout procedure [13] as outlined in Liberty Profiles and Binding Specification [12].

NOTE1: The NAF/IdP might contact the BSF for deletion of all data related to this Ks used to derive the Ks_(ext/int)_NAF, but that would be an even larger sort of logout and is not further elaborated on in this document. This does not eliminate the valid keys already given out to NAFs, those keys are valid up to the key lifetime or shorter, due to the local policy of the NAF.

NOTE2: If an operator wants to cancel the session of an user at a NAF/IdP, then he needs to inform the NAF/IdP of this.The Zn reference point can not be utilized as is for this directly, due to the direction of the message flows.

## 4.3.3 SSO scenario: ID-FF with <lib:AuthnResponse> transfer

### 4.3.3.1 HTTPS with conventional TLS

In this scenario the UE is not LAP aware. All protocol elements are taken from within ID Federation Framework [7] and complemented by the GAA-specific details from [2]. First the steps are outlined that are needed when utilizing HTTPS deploying conventional TLS [24] according to [2], clause 5.3:

1) The UE contacts the SP to gain access to a service provided by the SP by sending an HTTP Request. This request shall contain the GBA-based authentication support indication (cf. step 3), as this is required for the redirection of the request according to step 3.

2) On receipt of the HTTP request from UE, the SP obtains the identity provider and sends a redirect HTTP Response with < lib:AuthnRequest> to UE. The means by which the identity provider address is obtained is implementation-dependent and up to the service provider.

3) The UE in turn contacts the IdP under the URL given in the Location header field and the UE must access the NAF/IdP URL with an HTTP Request with <lib:AuthnRequest> information [12].

   The UE shall indicate to the NAF/IdP that GBA-based authentication is supported by adding a constant string to the "User-Agent" HTTP header as a product token as specified in IETF RFC 2616 [12]. This constant string shall be set according to step 2 of clause 5.3 of TS 33.222 [2].

   If a bootstrapped security association between UE and IdP exists, then UE and IdP/NAF share the keys to protect reference point Ua and the UE possesses all necessary data to perform HTTP Digest Authentication from previous messages. In this case step 3 is combined with the request in step 5, and step 4 is omitted.

4) As the IdP is collocated with the NAF, the HTTP Digest authentication is conducted in the accordance to 3GPP TS 33.222 [2] and a HTTP response with Unauthorized status and WWW-Authenticate header field is sent to the UE. The method and details of this authentication are defined by TS 33.222 [2] and not in [7].

   If the UE does not contain a valid bootstrapping session or the freshness of the key material is not sufficient for the IdP, then the UE will execute a new bootstrapping procedure with the BSF. This is transparent to the SP.

5) The UE returns the Authorization data, using the B-TID as a username and the Ks_(ext/int)_NAF as password to the IdP. The UE may include further LAP related user data.

   If the IdP is collocated with the NAF, then this happens as outlined in TS 33.222 [2]. The USS might contain Liberty specific information.

6) The <lib:AuthnRequest> is processed. The IdP responds with an <lib:AuthnResponse> in the HTTP Response redirect URL [12]. The IdP may include further LAP-related data.

7) The UE contacts the SP again using this URL and HTTP Request with <lib:AuthnResponse>.

8) The SP answers with a HTTP Response.

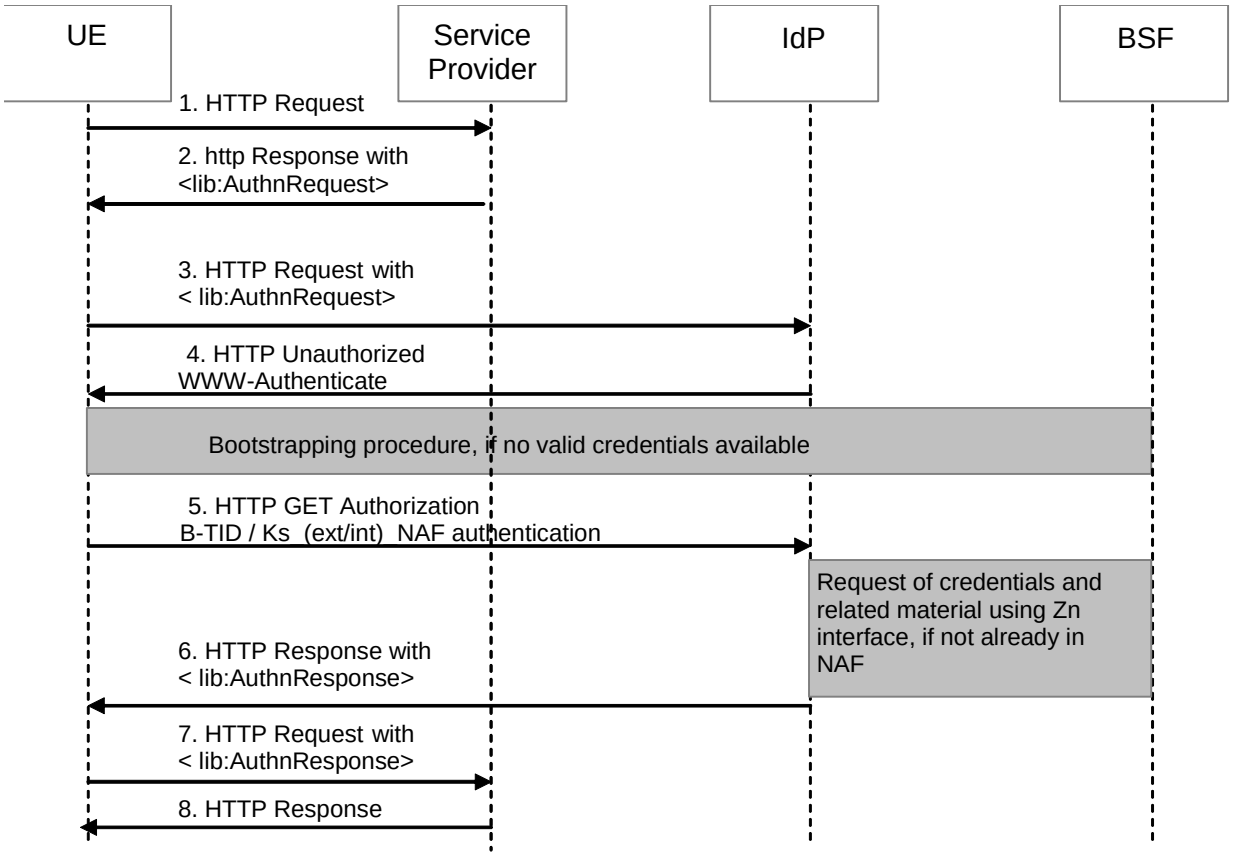


**Figure 4.3-1: Message flow for SSO with <lib:AuthnResponse> and conventional TLS with GBA**

NOTE 1: As the IdP is collocated with the NAF i.e. Ua is chosen for authentication as outlined in TS 33.222 [2], then each request over Ua is authenticated by itself, as each request carries the full Authorization Header. There is no difference between first request and follow-up requests.

NOTE 2: LAP ID-FF specification [7] defines also a POST-based communication between UE and IdP besides a GET-based request with a query string. This is in conformance with TS 33.222 [2], as there only a HTTP request is specified without any explicit method stated.

NOTE 3: The SP may use the GBA-based authentication support indication received in step 1 to select an appropriate identity provider address.

### 4.3.3.2 HTTPS with PSK TLS

When HTTPS with PSK TLS according to TS 33.222 [2], clause 5.4, is utilized, then the steps are the following:

1) The UE contacts the SP to gain access to a service provided by the SP by sending an HTTP Request. This request shall contain the GBA-based authentication support indication (cf. step 3 of clause 4.3.3.1), as the UE may be forced by the IdP/NAF to use conventional TLS, even if the UE offers the usage of PSK TLS.

2) On receipt of the HTTP request from UE, the SP obtains the identity provider and sends a redirect HTTP Response with < lib:AuthnRequest> in the URL to the UE. The means by which the identity provider address is obtained is implementation-dependent and up to the service provider.

3) The UE starts to set up a PSK TLS tunnel to the IdP/NAF as specified in clause 5.4 in TS 33.222 [2]. This is in preparation of sending the redirected request to the IdP/NAF (cf. step 4). During TLS tunnel setup the UE indicates possibility to use PSK TLS, and the IdP/NAF may select to use PSK TLS with GBA.

   The UE recognizes from the TLS ciphersuite selected by IdP/NAF if the IdP/NAF will use PSK TLS.

   If a bootstrapped security association between UE and IdP/NAF exists, then UE and IdP/NAF share the keys to protect reference point Ua. Thus the UE possesses all necessary data to set up the PSK TLS tunnel according to TS 33.222 [2] and the next step can be approached immediately without executing a bootstrapping procedure.

If no bootstrapped security association between UE and IdP/NAF exists, but the UE does contain a valid bootstrapping key Ks, then the UE establishes a PSK TLS tunnel with the IdP/NAF based on the related Ks_(ext)_NAF.

If the UE does not contain a valid bootstrapping session or the freshness of the key material is not sufficient for the IdP/NAF, then the UE will execute a new bootstrapping procedure with the BSF. This is transparent to the SP.

4) The UE accesses the IdP/NAF URL with the HTTP GET Request with <lib:AuthnRequest> information [12] within the established PSK TLS tunnel.

5) The IdP extracts the <lib:AuthnRequest>, processes it, uses the UE authentication done during the PSK TLS tunnel establishment, and sends a redirect HTTP Response to the UE, which redirects the UE back to the SP. The URL may contain a SAML artefact or a <lib:AuthnResponse>.

6) The SP extracts the SAML artefact or the <lib:AuthnResponse>, processes it and answers with a HTTP Response.

7) The SP answers with a HTTP Response.

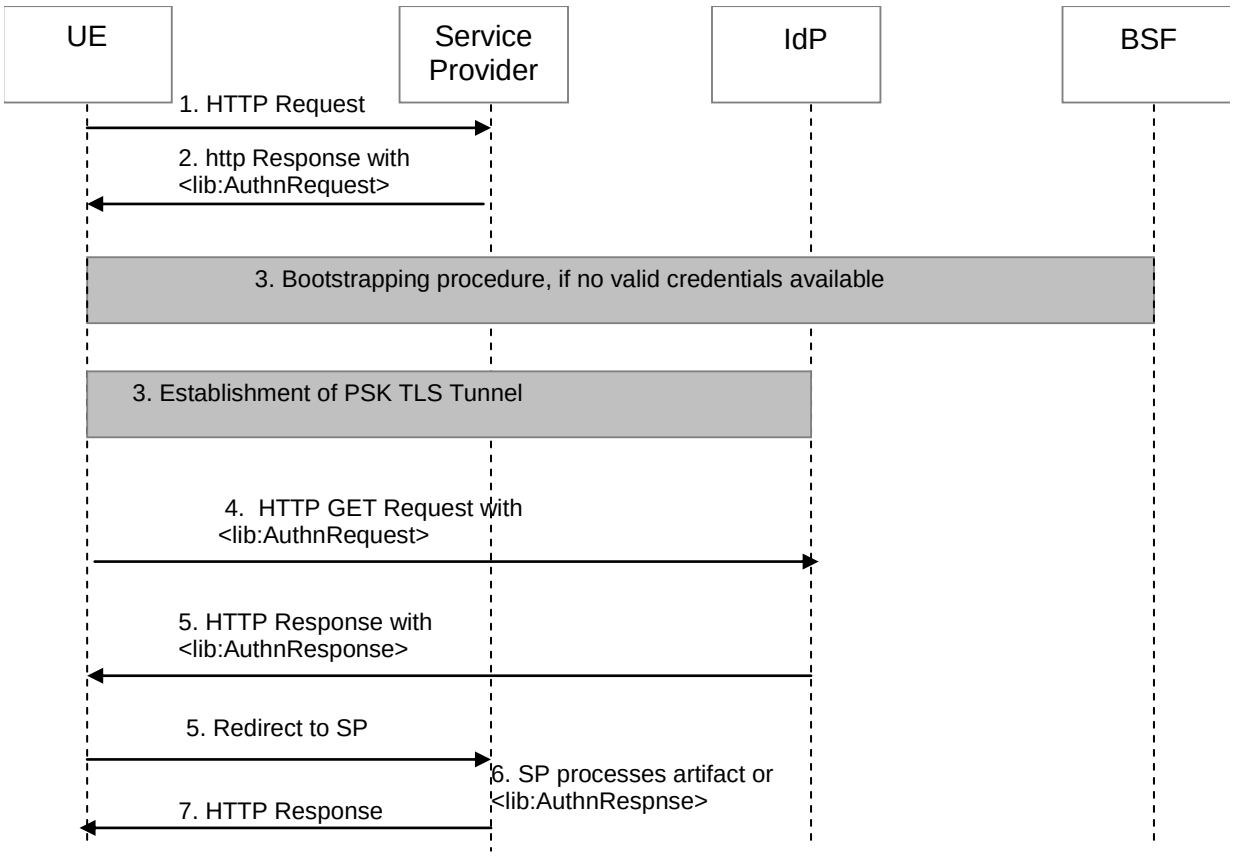


**Figure 4.3-1a: Message flow for SSO with <lib:AuthnResponse> and usage of PSK TLS with GBA**

NOTE: The notes given in clause 4.3.3.1 are also applicable for usage of PSK TLS as defined in this clause.

## 4.3.4 SSO scenario: ID-FF with artefact transfer

This scenario is similar to the scenario given in clause 4.3.3, with the extension that the service provider is able to contact the IdP directly.

NOTE: As the basic message flow is the same for artefact and for <lib:AuthnResponse> usage, the same differences between usage of conventional TLS and PSK TLS as in clause 4.3.3 apply to this clause also. Message flows given in this clause refer to conventional TLS, Analogous usage of PSK TLS is also possible.

The IdP must support an additional interface to SP, to allow the SP retrieval of the authentication assertion. This interface is not completely separated from GBA, as this authentication information may include GBA related information, e.g. user identity, pseudonym and further information from GUSS, restrictions based on GBA, etc.

1) The UE contacts the SP to gain access to a service provided by the SP by sending an HTTP Request. This request shall contain the GBA-based authentication support indication (cf. step 3), as this is required for the redirection of the request according to step 3.

2) On receipt of the HTTP request from UE, the SP obtains the identity provider and sends a redirect HTTP Response with < lib:AuthnRequest> to UE. The means by which the identity provider address is obtained is implementation-dependent and up to the service provider.

3) The UE in turn contacts the IdP under the URL given in the Location header field and the UE must access the NAF/IdP URL with an HTTP Request with <lib:AuthnRequest> information [12].

   The UE shall indicate to the NAF/IdP that GBA-based authentication is supported by adding a constant string to the "User-Agent" HTTP header as a product token as specified in IETF RFC 2616 [12]. This constant string shall be set according to step 2 of clause 5.3 of TS 33.222 [2].

   If a bootstrapped security association between UE and IdP/NAF exists, then UE and IdP/NAF share the keys to protect reference point Ua and the UE possesses all necessary data to perform HTTP Digest Authentication from previous messages. In this case step 3 is combined with the request in step 5, and step 4 is omitted.

4) If the UE is not yet authenticated with the IdP, then the authentication has to take place here, as defined in TS 33.222 [2]. The method and details of this authentication are not defined by Liberty Alliance in [7]. The IdP sends a HTTP response with Unauthorized status to the UE as defined in TS 33.222 [2].

   If there is no valid NAF specific key material in the NAF, or the freshness of the key material is not to the satisfaction of the NAF or IdP, then the bootstrapping procedure has to be performed as defined in TS33.220 [1]. This is transparent to the SP.

5) The UE answers with a HTTP GET request with Authorization header field containing as a username the B-TID and as a password the Ks_(ext/int)_NAF. The UE may include further LAP related user data.

   The IdP/NAF can request the credentials and related material, if it does not have it stored already. The received USS may contain further Liberty specific information.

6) The IdP responds with a SAML artefact in the HTTP Response redirect URL [12]. The IdP may include further LAP related data.

7) The UE contacts the SP again using this URL and HTTP Request with the SAML artefact.

8) The SP sends an HTTP Request with the SAML artefact to the IdP. The request contains a <samlp:Request> SOAP Request message to the identity provider's SOAP endpoint, requesting the assertion by providing the SAML assertion artefact in the < samlp:AssertionArtefact> element as specified in [12]

9) The IdP can now construct or find the requested assertion and responds with a <samlp:Response> SOAP Response message with the requested <saml:Assertion> or an status code as defined [13]. The IdP sends the authentication assertion that corresponds to the artefact.

10) The SP processes the SOAP message with the <saml:Assertion> returned in the <samlp:Response>, verifies the signature on the <saml:Assertion> and processes the message as defined in [12] and then answers with a HTTP Response.

The SAML authentication assertion should have a lifetime equal to or less than the B-TID. The assertion should be stored together with the B-TID in the table described in clauses 4.3.1 and 4.3.2.

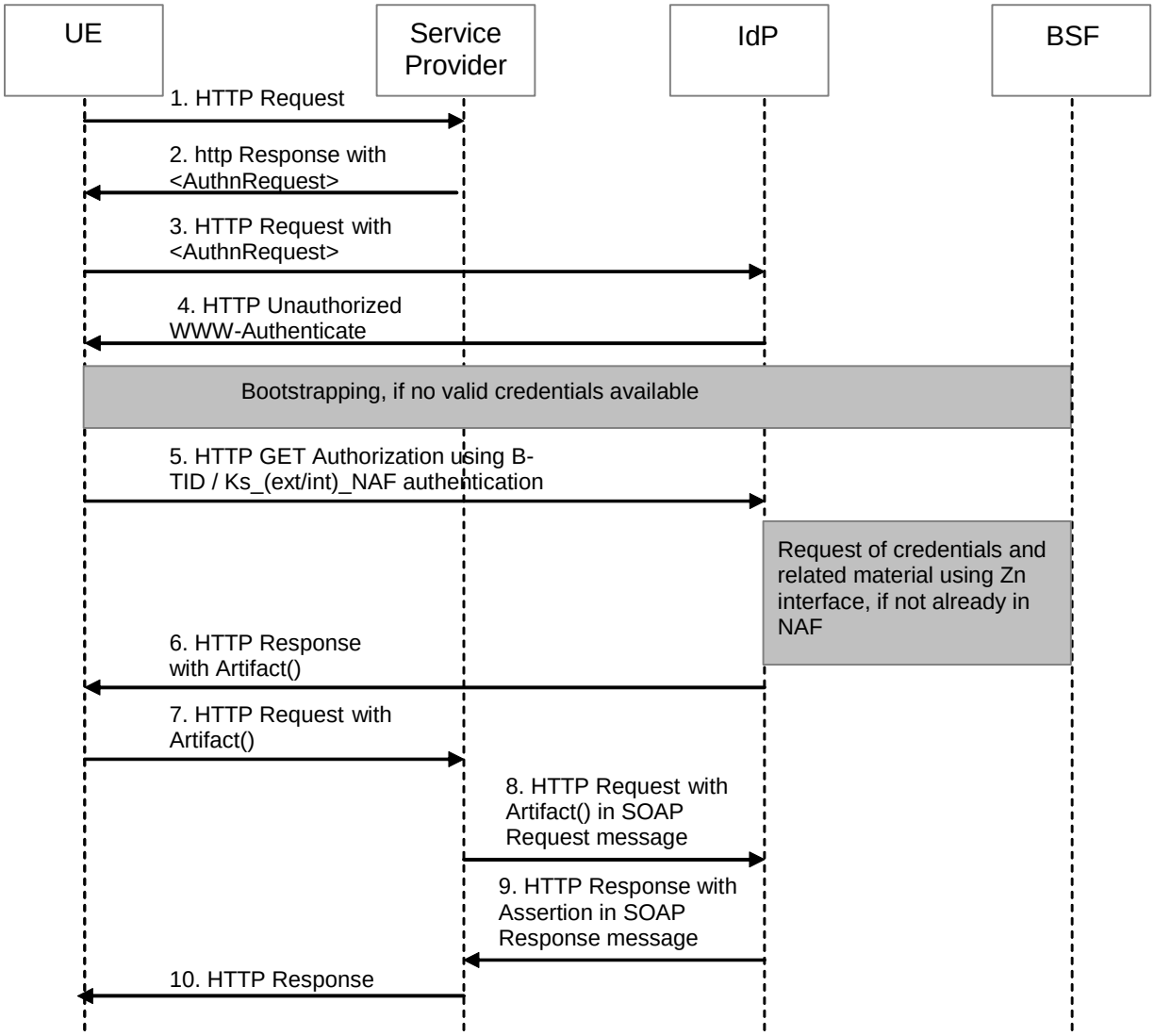


**Figure 4.3-2: Message flow for SSO with Artefact transfer and usage of GBA**

### 4.3.5 SSO scenario: ID-WSF Authentication Service

In this scenario the UE is LAP enabled, i.e. a LUAD (Liberty enabled User Agent or Device as defined in Liberty ID-WSF Profiles for Liberty enabled User Agents and Devices specification [16]). The protocol elements used are taken from ID–WSF Authentication Service [8], and the interaction of UE with IdP comprises two consecutive protocol runs. The active LUAD client contacts the NAF/IdP first before accessing the service provided by the SP.

1. The UE authenticates with the Authentication Service (AS) of the IdP and retrieves a security token, which entitles the UE to invoke some services.

2. The UE invokes the Single-Sign-On Service (SSOS) of the IdP using the security token. In this step the UE receives the authentication assertion (authentication and authorisation information) to be used at the SP.

3. The UE presents the authentication assertion to the SP acting as a WSP for web service access.

In case the WSP providing the web service to the user is part of the domain of the IdP operator, the LUAD client may also contact the WSP directly with the security token. In this case the SSOS contact may be left out.

Mapping of the three steps to GBA is done in the following way:

- The first step is mapped to the communication between user (LUAD) and AS as specified within LAP [8]. The authentication protocol is embedded in the SASL protocol as described in clause 4.2.1.2. The Ub run must be executed by the UE if necessary. This is not based on LAP protocols [6], [7] or [8], but only on GBA protocols [1].

- The second and third steps are completely as defined in LAP (no connection to GBA). The only dependency on GBA is in the content of the SAML authentication assertion depending partly on GBA results (protocol parameters, e.g. execution time, and user-specific parameters, e.g. taken from USS).

  The following gives a message flow for the SSO scenario of the ID-WSF authentication service with response transfer. This can also applies when the SSOS also offers an ID-WSF authentication service, in which case the SSOS is collocated with the AS.

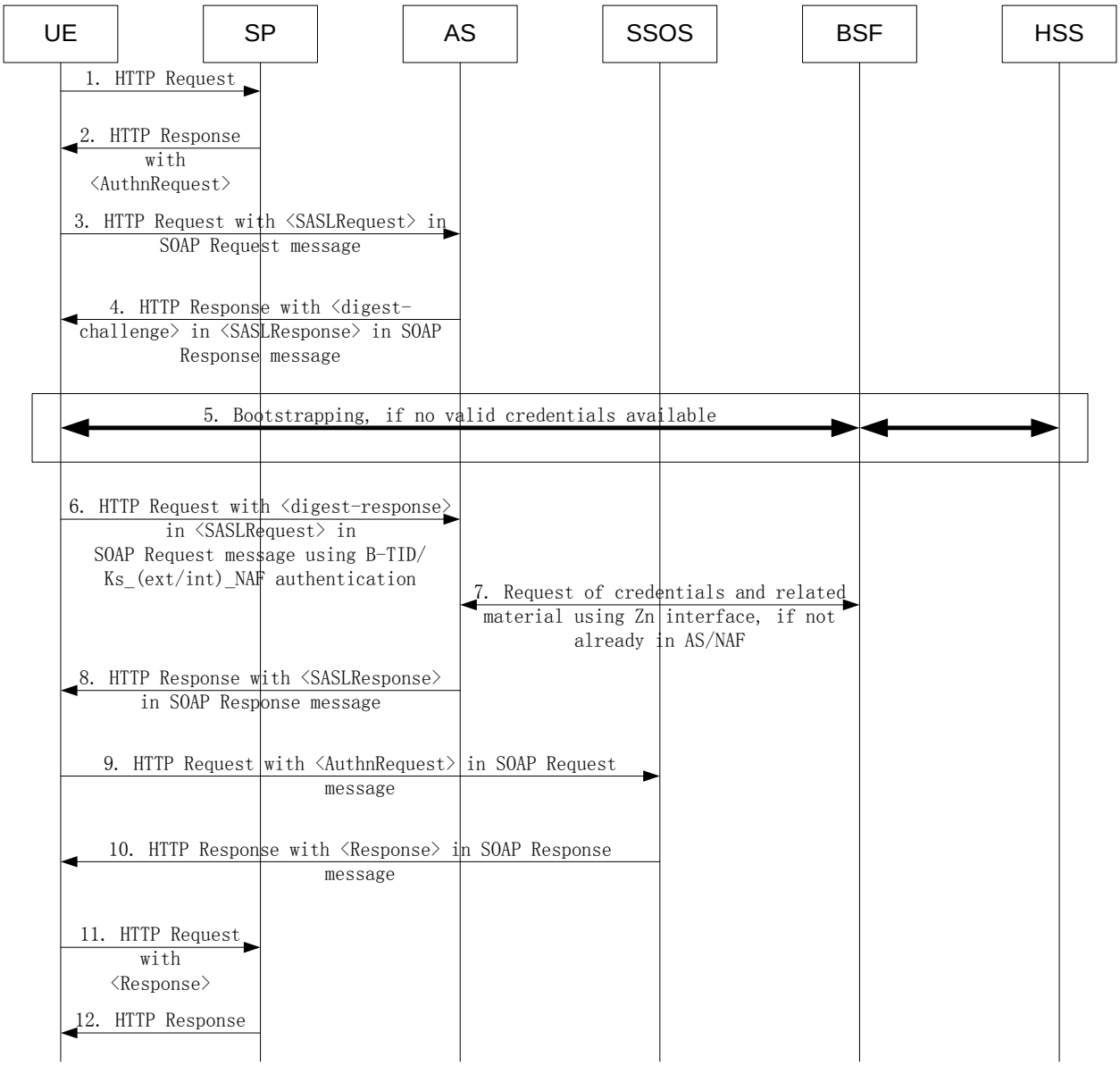


**Figure 4.3-3: Message flow for ID-WSF AS and SSO with Response transfer and usage of GBA**

1. The UE contacts the SP to gain access to a service provided by the SP by sending an HTTP request.
2. On receipt of the HTTP request from the UE, the SP obtains the AS address and sends a redirect HTTP response to the UE. The HTTP response may or may not contain an < lib:AuthnRequest> header according to the application or deployment model. The means by which the AS's address is obtained is implementation-dependent.
3. The UE (LUAD-WSC) sends an HTTP request to the AS. The request contains a soap-bound <SASLRequest> header, where the "mechanism" parameter is filled with a list of one-or-more client-supported SASL mechanism names.

The UE shall indicate to the NAF/AS that GBA-based authentication is supported by adding a constant string to the "User-Agent" HTTP header as a product token as specified in IETF RFC 2616 [12]. This constant string shall be set according to step 2 of clause 5.3 of TS 33.222[2].

If a bootstrapped security association between UE and NAF/AS exists, then UE and NAF/AS share the keys to protect reference point Ua and the UE may perform a subsequent authentication procedure if the SASL profile allows. In this case step 3 is combined with the request in step 6, and step 4 and step 5 are omitted.

4 The AS sends a HTTP response to the UE. The response contains a soap-bound <SASLResponse> header, where the "serverMechanism" parameter is filled with a selected SASL mechanism name (i.e. DIGEST authentication) from the client-supported SASL mechanism list and in this case the <SASLResponse> header also contains a <digest-challenge> parameter. The method and details of this parameter are compliant to RFC2831.

5 If the UE does not contain a valid bootstrapping session or the freshness of the key material is not sufficient for the AS, then the UE will execute a new bootstrapping procedure with the BSF and obtain a shared key Ks_(ext/int)_NAF. This is transparent to the SP.

6 The UE re-sends a HTTP request to the AS. The request contains a soap-bound <SASLRequest> header, where the "mechanism" parameter is filled with the returned SASL mechanism in step 4 and in this case the <SASLRequest> header also contains a <digest-response> parameter, where the authorization data is computed using the B-TID as a username and the Ks_(ext/int)_NAF as the password. The method and details of this parameter are compliant to RFC2831. The UE may include further LAP related user data.

7 As the AS is collocated with the NAF, the AS requests Ks_(ext/int)_NAF and other materials from the BSF using the Zn interface if they are not available yet.

8 The AS processes the <digest-response> parameter in the <SASLRequest> header. Then the AS responds with a soap-bound <SASLResponse> header in the HTTP Response. The <SASLResponse> header contains an ID-WSF EPR (EndpointReference) parameter which refers to the SSOS instance and the Service type URI is set according to [8] to identify the ID-WSF SSOS. The <SASLResponse> header also contains some necessary credentials for the UE to invoke the SSOS. The AS may include further LAP-related data.

9 The UE sends a HTTP request to the SSOS. The request contains a soap-bound <samlp2:AuthnRequest> header, where the ProtocolBinding attribute is set according to [8] to identify the SAML protocol binding to be used .The request also contains a <wsse:security> header which includes the returned credentials in step 8.The UE may have to construct the <samlp2:AuthnRequest> header by itself if it does not receive such a header in step 2 according to the application or deployment model.

10 The <samlp2:AuthnRequest> is processed. The SSOS responds with an <samlp2: Response> header in the HTTP Response redirect URL [12]. The <samlp2: Response> header contains a <saml2:Assertion> parameter . The SSOS may include further LAP-related data.

11 The UE contacts the SP again using this URL and HTTP Request with <samlp2: Response >.

12 The SP answers with a HTTP Response.

NOTE: If the IdP is co-hosted with the BSF, then the first step could be mapped to Ub reference point of GBA [4]. The second step could be mapped to Ua interface of GBA.

Despite having this formal analogy of executing two consecutive protocol runs required by both protocol worlds, it seems that a simple mapping is not possible. The syntax and semantic of the information elements transferred between GBA and LAP protocols differ substantially. This is one of the reasons, why clause 4.2.2 above states that , the ID-WSF IdP/BSF co-hosting scenario will not be elaborated further in this document.

## 4.3.6 SSO scenario: SAML v2.0 with <samlp:Response> transfer

### 4.3.6.1 HTTPS with TLS

This scenario is a version of the scenario in clause 4.3.3.1 with the difference that all protocol elements are taken from within SAML v2.0 [28] implementing the Web Browser SSO Profile from [13]. Hence all the steps described there apply here as well, after replacing <lib:AuthnRequest> with <samlp:AuthnRequest> and <lib:AuthnResponse> with <samlp:Response>. The steps are not repeated here, only an adapted version of Figure 4.3-1 is included.

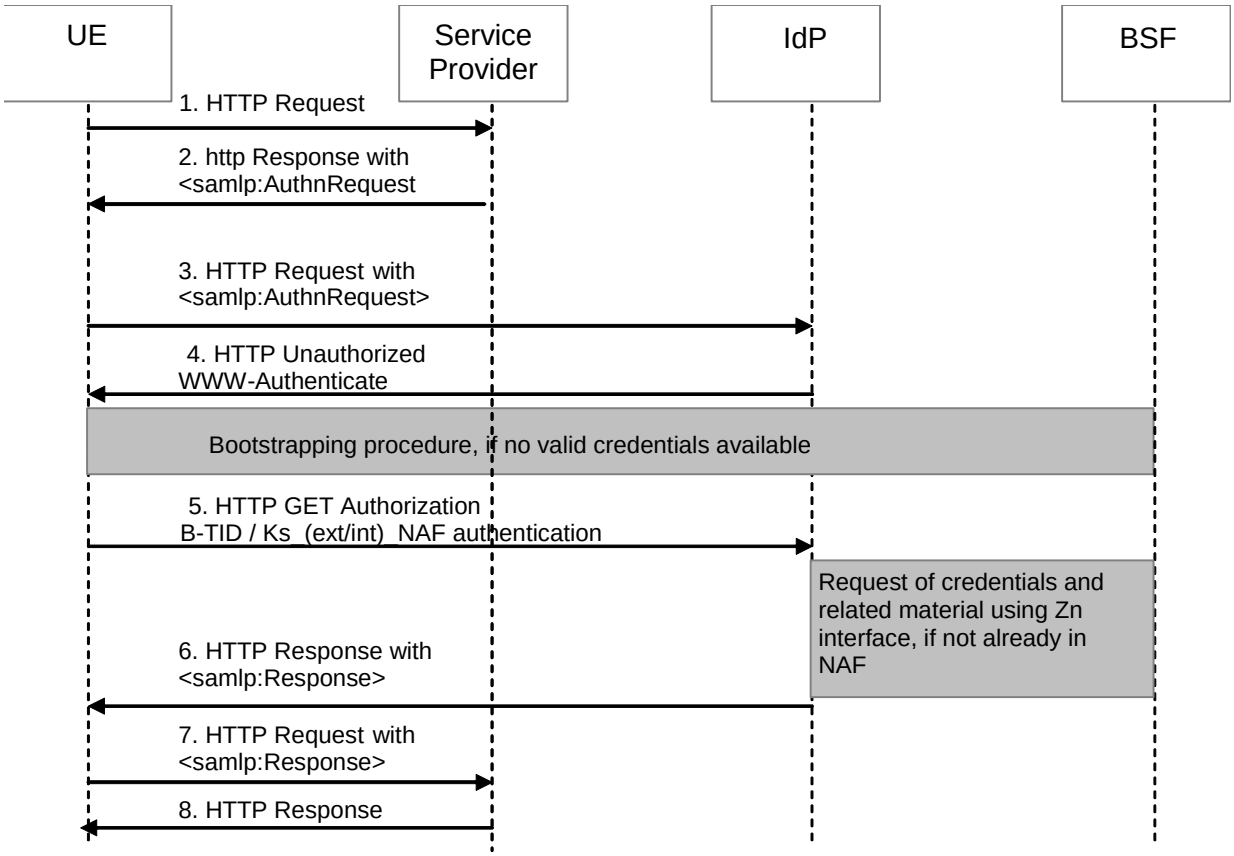


**Figure 4.3-4: Message flow for SSO with <samlp:Response> and TLS with GBA**

### 4.3.6.2 HTTPS with PSK TLS

This scenario is a version of the scenario in clause 4.3.3.2 with the difference that all protocol elements are taken from within SAML v2.0 [28] implementing the Web Browser SSO Profile from [13]. Hence all the steps described there apply here as well, after replacing <lib:AuthnRequest> with <samlp:AuthnRequest> and <lib:AuthnResponse> with <samlp:Response>. The steps are not repeated here, only an adapted version of Figure 4.3-1a is included.

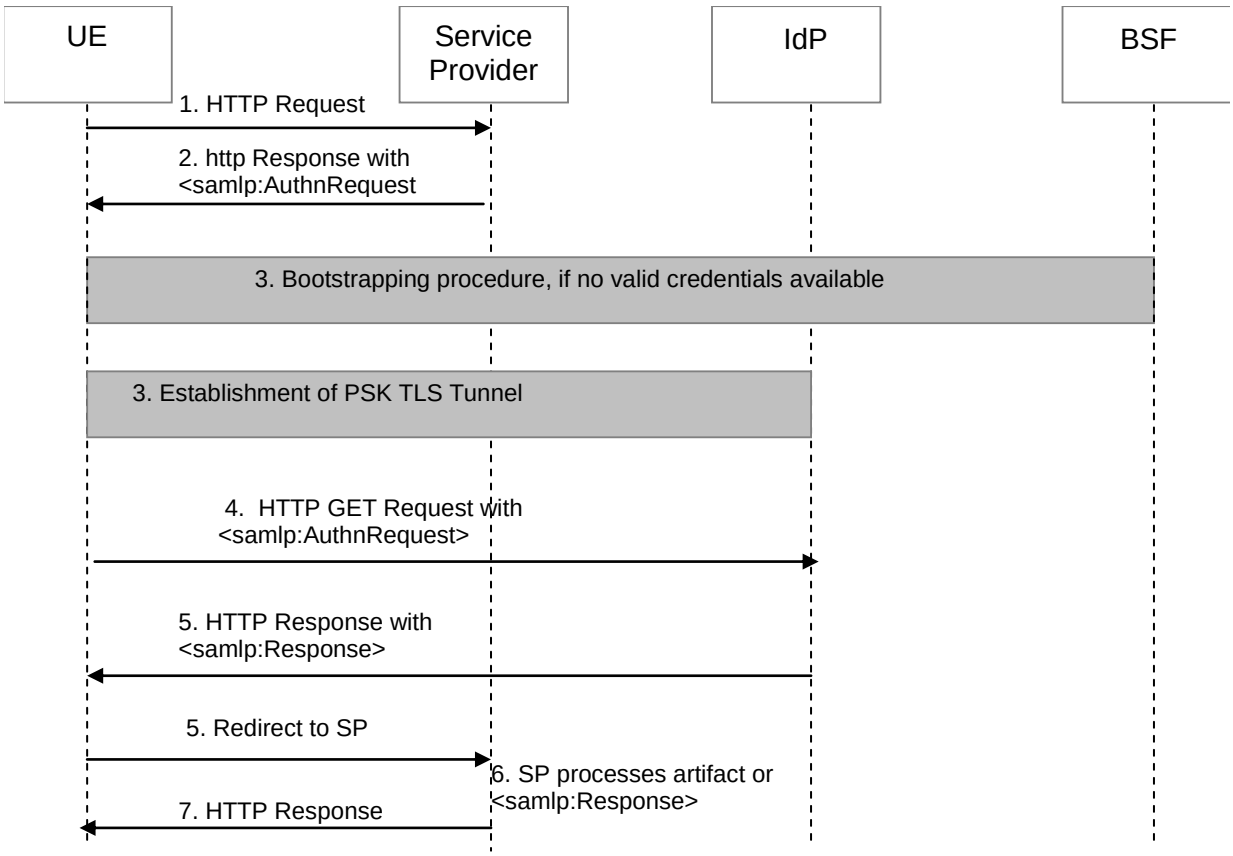


**Figure 4.3-5: Message flow for SSO with <samlp:Response> and usage of PSK TLS with GBA**

## 4.3.7 SSO scenario: SAML v2.0 with artefact transfer (resolution)

This scenario is a version of the scenario in clause 4.3.4 with the difference that all protocol elements are taken from within SAML v2.0 [28] implementing the Web Browser SSO Profile from [13]. Hence all the steps described there apply here as well, after replacing <lib:AuthnRequest> with <samlp:AuthnRequest>. The steps are not repeated here, only the adapted version of Figure 4.3-2 is included.

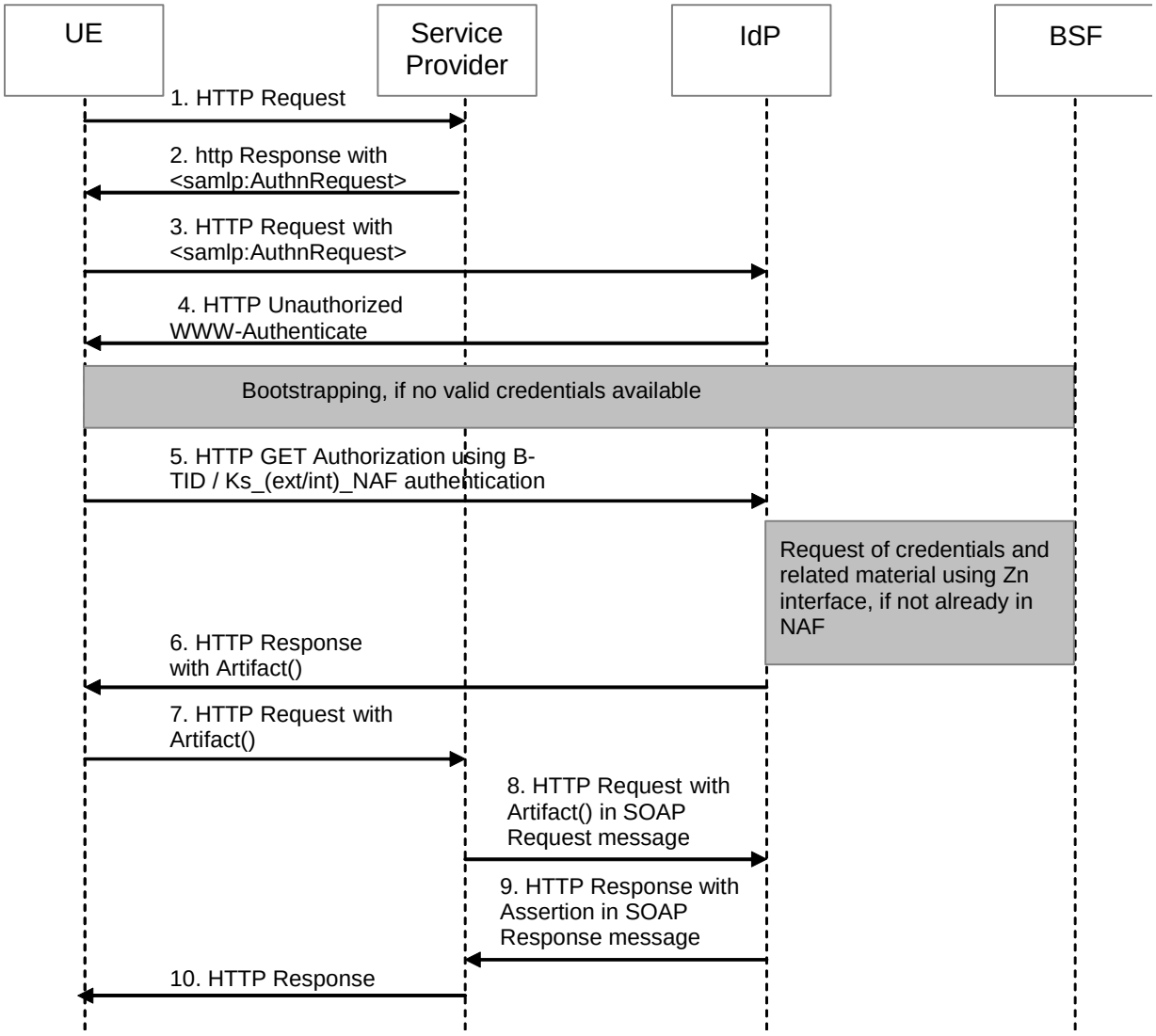


**Figure 4.3-6: Message flow for SSO with Artefact resolution (SAML v2.0) and usage of GBA**

# 4.3a Co-hosting of BSF and IdP

## 4.3a.1 General

In this clause it is assumed that the GBA BSF is collocated with a Liberty IdP as defined in [7]. Therefore, IdP/BSF is able to authenticate users by making use of the Ub bootstrapping procedure against the UE as defined in TS24.109 [4] (together with some other protocol fragments as defined in section 4.2.2). The combination of all these protocol fragments define the IdP/BSF Profile ("profile" term as used in SAML parlance).

According to clause 4.3 in TS24.109 [4], the actual Ub bootstrapping procedure is triggered by the UE itself, when it is sent to the appropriate BSF by a NAF (as clause 16.2 in TS23.003 [X] describes, there is no need to define a BSF discovery procedure, since the UE is able to derive the address of the appropriate BSF and therefore start a Ub bootstrapping procedure when no shared key is available). However, in the Liberty Alliance model the Identity Provider is in charge of starting the authentication procedure upon request from the service provider, which usually asks for a specific authentication method as defined in [7] and [12], by using an Authentication Context (if GBA-based authentication is wished, it would use the Authentication Context for GBA as defined in Annex E of TS 29.109 [5]).

On the other hand, as described in clause 4.3a.1 below, the IdP may know that the UE supports the GBA-based authentication since a "product" token is included into the "User-Agent" HTTP header within the HTTP requests issued by the UE (a given IdP may use other methods to determine that the Principal supports GBA-based authentication, such as obtaining such information directly from the user. These are, however, out of scope of this specification).

When a service provider requires the Principal to be authenticated by means of GBA, s/he is usually redirected to the IdP passing on a <samlp:AuthnResponse> message and also indicating that it requires such kind of authentication method (the use of the Web Browser SSO Profile is assumed). The IdP may trigger the beginning of the authentication procedure by sending back a bootstrapping required indication. All [6] and [7] specific tasks are fulfilled by the IdP/AS implementation, this is transparent to the GBA function in the UE.

This clause also applies to the case where GAA interworks with Liberty Alliance ID-WSF. In this case the AS/BSF as part of IdP works similarly as the IdP/BSF in ID-FF. For the sake of brevity only IdP/BSF is mentioned in the following text. Following the same principle, only the SSO scenario based on the <samlp:AuthnResponse> transfer by using the Web Browser SSO Profile is fully described. SSO scenario based on the transfer of an artifact is not shown, since it is a combination of the former scenario and that the one described in clause 4.3.4.

NOTE: The BSF/IdP collocation scenario presents certain security vulnerability under certain circumstances (i.e. a MitM attack over the Ub interface may become possible as the GBA bootstrapping over Ub is not cryptographically bound to the Liberty procedures). The attack becomes very difficult to implement (and therefore, almost negligible) if the Ub procedures are run end-to-end over the operator administrated network which is properly enrypted (UE initiates the bootstrapping procedure directly with the preconfigured BSF-address, and if all the communication runs over the properly encrypted operator administrated network, there is practically little chance for the MitM to actually get "in the middle" during this Ub procedure). In this case, the security of the solution is basically ensured by relying on the trustworthiness and non-vulnerability of the operator's routing infrastructure

## 4.3a.2 UE behaviour

When the UE is redirected to the IdP/BSF in order to request an authentication assertion (conveying a <samlp:AuthnRequest> message), it shall indicate to the IdP that GBA-based authentication is supported according to step 2 in clause 5.3 of TS 33.222 [2] and following the rule in clause 5.2.1 in [4], which state that the UE shall that support 3GPP-bootstrapping-based HTTP Digest authentication and shall indicate it by including a "product" token into the "User-Agent" header in each outgoing HTTP request.

## 4.3a.3 IdP/BSF behaviour

When the Identity Provider receives a <samlp:AuthnResponse> requiring the use of GBA-based authentication and, in the same message, an indication of the UE supporting such kind of authentication procedure (by means of the User-Agent HTTP header) it may, according to its internal policies, trigger the beginning of the authentication procedure by sending back a bootstrapping required indication (as if it were a NAF) as described in clause 5.2.4 in [4].

This behaviour does not impact current functionality in standard BSFs. It is the responsibility of those IdPs able to authenticate according to the GBA procedures to trigger such GBA-based authentication. Therefore an IdP/BSF shall be able to behave as a standard BSF and, at the same time, trigger GBA-based authentication in this specific case.

In order to avoid MitM attacks during this authentication procedure, the operator could for instance only accept GBA-based authentications towards the IdP/BSF in those deployments where it is ensured that all the authentication procedures are run end-to-end over the operator's network. I.e., in this case, the security of the solution relies on the trustworthiness and non-vulnerability of the operator's routing infrastructure.

## 4.3a.4 Federation Concept in GBA with IdP/BSF collocation

As described in clause 4.3.1, the Liberty Alliance technologies relies on the concept of "federation" (pairwise sharing of Principal identifiers between two sites) . This act of establishing a relationship between the digital information of a Principal at two entities requires a mapping between the identifiers used for the Principal at each entity. Following the same pattern, to be able to map the user GBA-related information (Ks and B-TID as part of the security association between UE and BSF and security related information about subscriber, such as the user's private identity) and the Liberty Alliance profile at the Identity Provider, the IdP/BSF shall receive as part of the whole bootstrapping procedure execution a user identity that allows it to locate the LAP user profile. Such user identity shall be already available in the user profile managed by the IdP. There are two options with regard to the user identity it may receive:

- IMPI. The IMPI has been received from the UE when running the Ub bootstrapping procedure (it is also received back from the HSS over the Zh reference point).

- UID. The UID is retrieved as part of the GBA User Security Settings from the HSS over the Zh reference point. It may be an IMPU or any other user identity.

The IMPI or UID will be used by the IdP business logic to link the user GBA-related information received and the LAP user profile stored by the IdP/BSF. The way the IdP indexes its user profile database is outside the scope of this recommendation (i.e., the IMPI or UID may be database indexes, but the only real requirement is that one of them can be used by the IdP logic to locate the right user profile). Apart from the LAP-corresponding service-related opaque handles (service specific user identifiers; in certain scenarios when privacy is a requirement, the service specific user identifiers should be different for each service to ensure the user's privacy), this user identity (IMPI or UID) is the only GBA-related information that the IdP/BSF permanently stores. The rest of user GBA-related information (user's B-TID, key lifetime data (Ks), bootstrapping time and GUSS) is obtained upon successful execution of the GBA bootstrapping procedure and stored by the IdP. The temporary GBA data shall be deleted on key expiry. It may also be removed on Liberty session expiry. The IdP-related data, and the persistent user identifier are persistent.

The IdP/BSF may handle defederation (termination of the federation) between the user GBA-related information and the LAP user profile by simply removing the GBA-related user identifier (IMPI or UID). The procedures to do so, in the same way as with the federation, are implementation-dependent and outside the scope of this document. Handling of defederation of the user identity at the IdP with other LAP service providers is done according to LAP specifications, as described in clause 4.3.1.

## 4.3a.5 Session Concept at the IdP

In LAP-GAA/GBA interworking scenarios, the session concept of Liberty Alliance shall be mapped to the key lifecycle lifetime of the entity collocated with the IdP (in the IdP/BSF collocation option, to the Ks). Therefore, the maximum Liberty Alliance session lifetime shall be equal to or shorter than the remaining lifetime of Ks. When the Liberty session expires, the temporary GBA related data may be deleted from the IdP storage described in clause 4.3a.3. If a Liberty session is explicitly terminated e.g. via Single-Logout, then the temporary GBA related data may be also deleted in the IdP, depending on the IdP/BSF internal policies. If removed, for the next user login, the UE would be required to execute the Ub bootstrapping procedure again, since it has no shared keys (Ks) with the IdP/BSF. If a new Ub bootstrapping procedure was executed since the last contact between UE and IdP/BSF, the new temporary GBA related data is stored (as the IdP is also a BSF).

In the case of IdP/BSF collocation, when a user starts a Liberty session with the IdP and it decides that the GBA bootstrapping procedure shall be executed, it triggers the authentication through the Ub reference point. If it the first time the UE contacts the BSF or the B-TID has expired, the whole Ub bootstrapping procedure is executed. If the B-TID has not expired, a new Ub procedure is not needed. However, if the IdP, according to internal policies policies, decides that the remaining lifetime of the B-TID is too short, it may goes on with the whole bootstrapping procedure.

Liberty Alliance specifications have the concept of Authentication Instant (the time when the authentication procedure took place). In interworking scenarios, the bootstrapping time is available to the IdP/BSF. Since the bootstrap procedure requires Digest AKA, the bootstrapping time should be typically taken as Liberty Authentication Instant (according to the internal IdP policies the Liberty authentication time could be set to a time more recent than the bootstrapping time),

It is important to note that the IdP might need to initiate a new Ub bootstrapping procedure, due to two possible reasons:

- According to IdP internal configuration policy (Liberty re-authentication settings shorter than key expiration time), or

- If a user with an ongoing LAP IdP session contacts the LAP IdP for authentication, and the <samlp:AuthnRequest> contains the element <ForceAuthn> (cf. [26], section 3.2.1.1), then the IdP shall send to the user a Bootstrapping Renegotiation Request according to section 4.5.3 of [1]. This is necessary as this may be a reauthentication request issued for liveness validation within LAP (cf. [7], section 4.4.2), requiring a new bootstrapping, as the bootstrapping time is typically taken as Liberty authentication time.

## 4.3a.6 SSO scenario: ID-FF with <samlp:AuthnResponse> transfer

In this scenario the UE is not LAP aware. All protocol elements are taken from within ID Federation Framework [7] and complemented by the GAA-specific details from [2]:

1) The UE contacts the LAP SP to gain access to a service provided by the SP by sending an HTTP Request. This request may contain the GBA-based authentication support indication (cf. step 3) in order to aid the SP

to select the appropriate IdP (input to be taken into account for the redirection of the request according to step 3).

NOTE: The GBA-based authentication support indication is not extrictly necessary. The UE may not send such indication. The SP will in every case be in charge of determining the appropriate IdP for this specific service request and user. This is known as "IdP Discovery" procedures and are out of the scope of the present document (although SAMLv2 documents some non-normative means of achieving that).

2) On receipt of the HTTP request from UE, the SP obtains the identity provider and sends a redirect HTTP Response with <samlp:AuthnRequest> to UE. The means by which the identity provider address is obtained is implementation-dependent and up to the service provider.

3) The UE in turn contacts the IdP under the URL given in the Location header field and the UE shall access the IdP/BSF URL with an HTTP Request with <samlp:AuthnRequest> information [12].

   The UE shall indicate to the IdP that GBA-based authentication is supported by adding a constant string to the "User-Agent" HTTP header as a product token as specified in IETF RFC 2616 [12]. This constant string shall be set according to clause 5.3 of TS 33.222 [2].

   The IdP/BSF checks if the UE has a valid bootstrapping sessions with the IdP/BSF. If the UE does not have a valid bootstrapping session with IdP/BSF, then the the process continues in step 4. If the UE does have a valid bootstrapping session with IdP/BSF, then the the process continues in step 5.

4) The IdP, acknowledging that the UE supports GBA-based authentication and that such authentication method is required (either upon explicit requirement from the SP or due to internal policies), sends back a bootstrapping required indication according to clause 5.2.4 in [4] for the UE to start the bootstrapping procedure.

   In order to be authenticated according to the GBA-defined procedures, the UE shall follow the specifications of clause 4.2 of TS 24.109 [4].

5) Upon successful run of the bootstrapping procedure the UE contacts the IdP under the URL given in the Location header field and the UE shall access the IdP/BSF URL with an HTTP Request with <samlp:AuthnRequest> information [12]. The IdP/BSF makes the same checks as in step 3.

6) The IdP responds with an <samlp:AuthnResponse> in the HTTP Response redirect URL [12]. The IdP may include further LAP-related data.

7) The UE contacts the SP again using this URL and HTTP Request with <samlp:AuthnResponse>.

8) The SP answers with an HTTP Response.

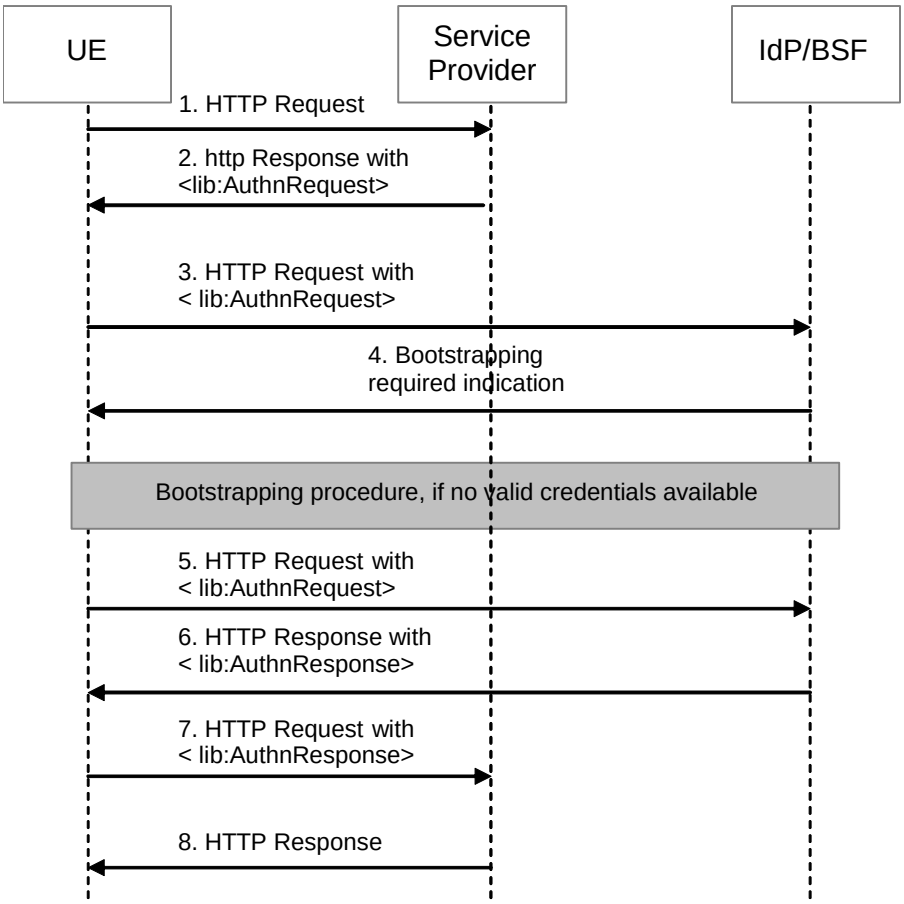


**Figure 4.3a-1: Message flow for SSO with <lib:AuthnResponse>**

NOTE 1: LAP ID-FF specification [7] defines also a POST-based communication between UE and IdP besides a GET-based request with a query string. The present description is consistent with TS33.222 [2], which only reflects an HTTP request, no any specific method explicitly stated.

NOTE 2: The SP may use the GBA-based authentication support indication received in step 1 to select an appropriate identity provider address (one more possible "IdP Discovery" mechanism).

## 4.4 Use of GUSS / USS in Support of ID-FF and ID-WSF

ID-FF and ID-WSF frameworks have the need for additional information elements not existent in basic GBA. These elements may be stored in HSS GUSS to ease MNO administration work. As stated in clause 1, this document describes interworking of GAA and LAP framework with changes to both as small as possible.

In consequence, the MNO part, and more precisely HSS GUSS, should only store data relevant for interworking. This corresponds well with the requirement that data in HSS should be quite static in nature, and that GUSS is only fetched by BSF on occasion of a Ub bootstrapping run, but not on every communication with a NAF.

The difference in interworking of GAA with ID-FF and ID-WSF is not reflected within GUSS which is defined to contain security settings. The deployment of ID-FF and ID-WSF is mainly characterised by use of different LAP protocol suites only, not by the use of difference security mechanisms.

All data used within LAP environment only is outside the scope of this document and as such assumed to be stored within LAP network elements or accessible from there. This applies e.g. to LAP Id-SIS [23] profiles or access rights in DS. On the other hand, this document does not preclude that e.g. user self-administration of her data at IdP is secured by GBA or by LAP SSO based on GBA.

A basic requirement for identity federation between GAA and LAP is a user identity commonly known to GAA and LAP, or a mechanism that allows for the mapping between the identifiers used in each of the domains, which is outside the scope of this document. From a GAA point of view, this may be the IMPI of the user, an IMPU, or any other public

user identity. If the IMPI is used as a common identifier for interworking purposes then this does not require the usage of GUSS, as the IMPI is known to the BSF from Ub bootstrapping run (straightforward implementation for the IdP/BSF alternative; when the IdP is collocated with NAF, the BSF may be configured by local policy to send the IMPI to NAF). Any other UID must always be transferred from HSS to IdP using USS.

### 4.4.1 GAA-LAP Interworking Service

Interworking of GAA and LAP is a service offered by an Identity Provider or Authentication Service that is collocated with some NAF or with the BSF in the framework of GAA: in other words, this feature is provided by a GAA network element (NAF or BSF) which also provides IdP functionality. If existing, USSs for this service are marked with the GAA Service Identifier (GSID) for this service.

NOTE: At the time being there is only one type of interworking service defined. Thus the GSID (GAA Service Identifier) for GAA-LAP interworking is the same as the GAA Service Type Code as defined in clause 4.4.2.

### 4.4.2 GAA-LAP Interworking USS

The following text profiles the definition of USS attributes as given in [5]:

- The value of the attribute "id" in the element "uss" is the service identifier (GSID) given in clause 4.4.1.
- The value of attribute "type" in the element "uss" is the GAA service type code for GAA-LAP interworking service as defined in Annex B in [5].
- The value of attribute "nafGroup" in the element "uss" is an operator internal group designator for a NAF group the USS is valid for. This attribute may be used by the operator to enforce distinction between different IdPs or circles of trust within LAP.
- Values of the element "uid" are user's public authentication identities from the HSS. These may be IMPUs or any other public user identities by which the user is known to the IdP.

NOTE: The value of the attribute "uid" in the element "uss" in TS 29.109 [5] can be used in the GAA Liberty Alliance Interworking case for the pertinent user identifier at the Liberty Alliance Identity Provider.

- Values of element "flag" are not defined for GAA-LAP interworking service.

### 4.4.2a GUSS / USS when IdP/AS is collocated with BSF

When the BSF runs the Ub bootstrapping procedure, it retrieves through the Zh reference point the GBA User Security Settings (GUSS) for the given IMPI.

Depending on the user identifier shared between the LAP IdP and the GBA environment, a specific desing of GUSS may be needed. As stated in clause 4.4, if the IMPI is chosen, no data from the GUSS is needed for the IdP itself, since the IdP/BSF will use the IMPI (provided by the UE over the Ub reference point).

If other identifier is used, then a specific USS shall be defined in the HSS containing such user idenfier. Besides, the BSF shall be configured to choose such a specific USS, the one that contains the user identity that allows it to locate the LAP user profile (the Ub bootstrapping procedure is not executed upon request of a NAF).

## 4.5 Liberty Alliance Authentication Context and GBA

The authentication context needs to contain information to describe that GBA was used for trust establishment and to describe how GAA/GBA was used e.g. Username / password in HTTPS. In addition the strength of the GBA authentication and the security of key storage in UE have to be taken into account (c.f. TS 33.220 [1]: GBA_ME, GBA_U, or 2G GBA). The Liberty authentication context specification is based on [15]. The SAML v2.0 authentication contexts are defined in [27].

NOTE: In case that GBA and Liberty Alliance Interworking extensions are needed by standardisation bodies other than 3GPP, then also the definition of authentication contexts has to be done in the applicable specifications.

The Liberty Alliance ID-FF v1.2 (also valid for SAML v2.0) Authentication Context for GBA is defined in Annex E of TS 29.109 [5].

# Annex A: Digest Authentication within SASL for Ua protocol between UE and AS/NAF

Liberty Alliance specifications define an ID-WSF Authentication Protocol based on a profile of the Simple Authentication and Security Layer (SASL) framework [17] mapped onto ID-* SOAP-bound messages. As SASL provides only a wrapper for many kinds of authentication protocols, this report suggests the usage of digest authentication within SASL for authentication of UE to AS within GBA. This annex defines the usage of MD5 digest authentication according to RFC 2831 [18] within SASL for as an instance of the Ua reference point. This annex keeps as close as possible to TS 33.222 [2], where digest authentication according to RFC 2617 [19] is used with Ks_NAF for authentication.

RFC 2831 defines a slightly different variant of MD5 digest authentication, compatible to the algorithm "MD5-sess" as specified in [19], which is similar to the "MD5" used in TS 33.222. These differences are not important for the use within GAA Liberty alliance interworking, except for the discussion on subsequent authentication and authentication context in clause A.5.

Digest authentication within SASL is used without Integrity and Confidentiality protection as specified in [18]. Both are catered for by the HTTPS protocol as described below.

## A.1 HTTPS deployment

Liberty Alliance recommends the use of a security protocol for all communications between UE and network elements. Section 4.5 of the Liberty ID-WSF Authentication Service Specification and Single Sign-On Service [8] recommends TLS with server certificates for server authentication. Thus, in the scope of this Annex, HTTPS is defined as the security protocol. All statements about TLS deployment and relevant security checks in TS 33.222 [2] apply.

The Liberty ID-WSF Authentication Service Specification and Single Sign-On Service [8] requires all service providers offering ID-WSF authentication services to support at least the security mechanism "urn:liberty:security:2003-08:TLS:null". This LAP security mechanism is specified in [6] and requires server authentication with X.509v3 certificates. The requirement is fulfilled by the deployment of TS 33.222 [2]. As no message authentication is needed from a GBA-LAP interworking point of view, this LAP security mechanism is also sufficient in the context of this annex.

The TLS profile according to TS 33.222 [2] applies.

NOTE: The Liberty ID-WSF Authentication Service Specification and Single Sign-On Service [8] requires the support of TLS extensions as specified in RFC 3546 [22], while TS 33.222 [2] only mandates to support a subset from RFC 3546 [22], i.e. the "server name" extension. This is an additional requirement beyond this annex, but it does not constitute any contradiction to it.

## A.2 Digest challenge

The digest challenge sent from server to client is defined as follows:

digest-challenge = 1#( realm | nonce | qop-options | stale | maxbuf | charset | algorithm | cipher-opts | auth-param )

In the context of this annex the following values for the digest challenge are profiled:

- realm: the realm shall be set according to TS 33.222 [2].
- qop-options: only qop-option "auth" shall be used, as there is no body to be integrity protected and no need to encrypt subsequent messages based on the result of SASL protocol run.
- maxbuf: this value is not relevant in the context of this annex, as only qop="auth" is used. This value may be left out.

- charset: support of charset utf-8 is not mandatory, as digest auth authentication according to RFC 2617 [19] does only support ISO-8859-1.
- algorithm: this value must be " md5-sess" as this is the only value specified within RFC 2831 [18].
- cipher-opts: this value is not relevant in the context of this annex, as only qop="auth" is used. This value shall not be set.

## A.3 Digest response

The digest response sent from client to server is defined as follows:

digest-response = 1#( username | realm | nonce | cnonce | nonce-count | qop | digest-uri | response | maxbuf | charset | cipher | authzid |auth-param )

In the context of this annex the following values for the digest response are profiled in addition to the values handled in clause A.2:

- username: this value is set according to TS 33.222 [2].
- digest-uri: the "serv-type" shall be the service name "idwsf" according to [8]. "host" shall be the FQDN of the AS.

## A.4 Response auth

[18] requires the server to send a "response auth" to the client after successful authentication of the client. This is the same mechanism as the (optional) use of the Authentication-Info-Header in [19].

This "response auth" should be checked by the client to provide a second assurance that it connected to the correct server (besides the server authentication by server certificate). The security analyses should not rely on the client to perform this check correctly.

## A.5 Subsequent authentication

Use of the "subsequent authentication" mechanism depends on local policy in the AS.

If the client never sends an "initial response", then this mechanism is not used anyway. If the client sends an "initial response", then the server may accept or reject it depending on local policy.

Whether "subsequent authentication" should be used depends also on the security requirements of the authentication, as "md5-sess" has a session concept where subsequent authentications are not as independent as with "md5". For this topic c.f. discussion of "session concept" in [19]. This also influences the definition of "authentication context" as handled in clause 4.5.

# Annex Z: Change history

| Change history | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Date** | **TSG #** | **TSG Doc.** | **CR** | **Rev** | **Cat** | **Subject/Comment** | **Old** | **New** | **WI** |
| 2005-04 | | | | | | Creation of document | 0.0.0 | 0.0.1 | |
| 2005-07 | SA3#39 | S3-050478 | | | | Integration of S3-050478 to Liberty 3GPP Security Interworking TR | 0.0.1 | 0.1.0 | |
| 2005-09 | SA3#40 | S3-050666 | | | | Integration of S3-050520, S3-050522, S3-050523, S3-050529, S3-050536 approved in SA3#40 | 0.1.0 | 0.2.0 | |
| 2005-09 | SA3#40 | S3-050666 | | | | Integration of S3-050520, S3-050522, S3-050523, S3-050529, S3-050536 approved in SA3#40 | 0.2.0 | 1.0.0 | |
| 2005-12 | SA3#41 | S3-050858 | | | | Integration of S3-050856, S3-050712, S3-050854, S3-050855 approved in SA3#41 | 1.0.0 | 1.1.0 | |
| 2006-02 | SA3#42 | S3-060178 | | | | Integration of S3-060050 and S3-060121 | 1.1.0 | 2.0.0 | |
| 2006-02 | SA3#42 | | | | | Minor editorial changes made before presentation to SA | 2.0.0 | 2.0.1 | |
| 2006-03 | SP-31 | SP-06070 | - | - | - | Approved at SA #31 | 2.0.1 | 7.0.0 | |
| 2006-06 | SP-32 | SP-060389 | 0001 | - | B | Integration of CT4 approved CR on Authentication Context Definition | 7.0.0 | 7.1.0 | LibSec |
| 2006-06 | SP-32 | SP-060389 | 0002 | - | B | Addition of interworking details between GAA and LAP, especially USS details. | 7.0.0 | 7.1.0 | LibSec |
| 2006-06 | SP-32 | SP-060389 | 0003 | - | F | Liberty-3GPP interworking security | 7.0.0 | 7.1.0 | Liberty-GBA interworking |
| 2006-06 | SP-32 | SP-060450 | 0004 | 1 | F | Message flow for SSO scenario of ID-WSF authentication service | 7.0.0 | 7.1.0 | Liberty-GBA interworking |
| 2006-09 | SP-33 | SP-060506 | 0005 | - | F | Usage of HTTP POST method | 7.1.0 | 7.2.0 | LibSec |
| 2006-09 | SP-33 | SP-060506 | 0006 | - | F | Service based data management | 7.1.0 | 7.2.0 | LibSec |
| 2006-09 | SP-33 | SP-060506 | 0007 | - | F | Liberty ID-WSF and GBA interworking architecture | 7.1.0 | 7.2.0 | 6.25 Liberty-GBA interworking |
| 2006-09 | SP-33 | SP-060506 | 0008 | - | F | Clarifications and corrections | 7.1.0 | 7.2.0 | Liberty-GBA interworking |
| 2006-12 | SP-34 | SP-060812 | 0009 | 1 | F | Addition of a reference to the GAA Service Type Code in TS 29.109. | 7.2.0 | 7.3.0 | LibSec |
| 2006-12 | SP-34 | SP-060812 | 0010 | 1 | F | Removal of editor's note and validation of received identity | 7.2.0 | 7.3.0 | LibSec |
| 2006-12 | SP-34 | SP-060812 | 0011 | 1 | F | Addition of Note pointing to work related to 3GPP external entities. | 7.2.0 | 7.3.0 | LibSec |
| 2006-12 | SP-34 | SP-060812 | 0012 | 1 | F | Clarification of usage of TLS mechanism and of indication of GBA-based authentication support in GAA - ID-FF interworking. | 7.2.0 | 7.3.0 | LibSec |
| 2006-12 | SP-34 | SP-060812 | 0013 | 1 | F | Replacement of Editor's Note concerning LAP re-authentication. | 7.2.0 | 7.3.0 | LibSec |
| 2007-01 | - | - | - | - | - | Updated to include the entries of history box and to correct the version number on the cover sheet | 7.3.0 | 7.3.1 | - |
| 2007-03 | SP-35 | SP-070164 | 0014 | - | F | Removal of identity validation inconsistency | 7.3.0 | 7.4.0 | LibSec |
| 2007-03 | SP-35 | SP-070164 | 0015 | 1 | B | SAMLv2.0 Integration to TR 33.980 | 7.3.0 | 7.4.0 | LibSec |
| 2007-06 | SP-36 | SP-070333 | 0016 | 1 | F | Specifying compliance levels | 7.4.0 | 7.5.0 | LibSec |
| 2007-06 | SP-36 | SP-070333 | 0017 | - | F | Clarification of SAML v2.0 description | 7.4.0 | 7.5.0 | LibSec |
| 2007-09 | SP-37 | SP-070596 | 0020 | 3 | F | Enhancements on the architecture for the collocation of BSF with Libery Alliance IdP | 7.5.0 | 7.6.0 | LibSec |
| 2008-12 | SP-42 | -- | -- | -- | -- | Upgrade to Release 8 | 7.6.0 | 8.0.0 | -- |
| 2009-12 | SP-46 | SP-090819 | 0022 | -- | B | Missing logout for interworking | 8.0.0 | 9.0.0 | GBA-IdM |
| 2011-03 | - | - | - | - | - | Update to Rel-10 version (MCC) | - | 9.0.0 | 10.0.0 |
| 2012-09 | - | - | - | - | - | - | Update to Rel-11 version | 10.0.0 | **11.0.0** |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | (MCC) | | |