3GPP TSG CT Plenary Meeting #28 **CP-050091**
1st – 3rd June 2005 Quebec, Canada.

Source: TSG CT WG4

Title: Corrections on GUP

Agenda item: 9.6

Document for: APPROVAL

Doc-2nd-Level	Spec	CR #	Rev	Rel	Tdoc Title	CAT	C_Version
C4-050769	29.240	003	1	Rel-6	GUP HSS-IMS Component Definition	F	6.0.0
C4-050693	29.240	004		Rel-6	GUP Profile Structure	F	6.0.0
C4-050694	29.240	005		Rel-6	Security and Authentication	F	6.0.0
C4-050771	29.240	006		Rel-6	GUP SOAP Headres	F	6.0.0

3GPP TSG-CT WG4 Meeting #27 ***Tdoc ⌘C4-050693***
Cancun, Mexico. 25th to 29th April 2005.

CR-Form-v7.1

CHANGE REQUEST

⌘ **29.240** **CR** **004** ⌘ **rev** **-** ⌘ Current version: **6.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network ☐ Core Network ☒

Title:	⌘	Clarification of the content of Annex <A>
Source:	⌘	Ericsson
Work item code:	⌘	GUP
Date:	⌘	05/04/2005
Category:	⌘	**F**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

Ph2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*
Rel-7	*(Release 7)*

Reason for change:	⌘	As currently only the HSS IMS component is present in the Annex A, someone could think that the HSS IMS component is the only component to be included in GUP.
Summary of change:	⌘	A brief paragraph has been added in Annex A, in order to clarify that, although the current version of this specification only defines the IMS HSS GUP component, further versions of the specifcation could include the definition of other GUP components. This enforces the idea that the Generic User Profile is going to consist of many components, instead of **only** the IMS HSS Component.
Consequences if not approved:	⌘	The fact that only the IMS HSS Component has been defined could create confusion and lead to the erroneous belief that the only component that could be provided via GUP is the IMS HSS component.

Clauses affected:	⌘	Annex <A>

		Y	N		
Other specs affected:	⌘		X	Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	

Other comments: ⌘

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm.
Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

>>>>>>>>>>> First modified section <<<<<<<<<<<

Annex A (normative):Component Data Definitions

According to Section 5 of TS23.240 [1], a Generic User Profile consists of a number of independent GUP Components. This Annex defines the content, semantics and specific processing rules of the different GUP Data Components.

Note: For Rel-6 only the HSS IMS GUP component is defined. Further releases of this specification will define additional components.

A.1 HSS IMS GUP Component definition

>>>>>>>>>>>>>>>>>>>>>>>>>>>>> END OF CHANGE<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

3GPP TSG-CT WG4 Meeting #27 **Tdoc ⌘C4-050694**
Cancun, Mexico. 25th to 29th April 2005.

CR-Form-v7.1

CHANGE REQUEST

⌘ **29.240** **CR** **005** ⌘ **rev** **-** ⌘ Current version: **6.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network ☐ Core Network ☒

Title:	⌘	GUP Security and Authentication
Source:	⌘	Ericsson
Work item code:	⌘	GUP
Date:	⌘	15/04/2005
Category:	⌘	**F**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:
Ph2 (GSM Phase 2)
R96 (Release 1996)
R97 (Release 1997)
R98 (Release 1998)
R99 (Release 1999)
Rel-4 (Release 4)
Rel-5 (Release 5)
Rel-6 (Release 6)
Rel-7 (Release 7)

Reason for change:	⌘	Required Changes to the Security section related to Authentication as described in the accompaning discussion paper.
Summary of change:	⌘	Miscellaneus clarifications and addition for a complete specification of transport and message security mechanisms.
Consequences if not approved:	⌘	GUP security solution would remain underspecified.

Clauses affected:	⌘	10, 10.1, 12.1 and 12.2

Other specs affected: ⌘	Y	N	
		X	Other core specifications
		X	Test specifications
		X	O&M Specifications

⌘

Other comments: ⌘

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.
2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.
3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

10 Authentication, authorization and security

~~Editor's Note: description of general security, authentication and authorization mechanisms. The clauses 8 and 9 may provide additional reference point specific issues.~~

10.1 Principles

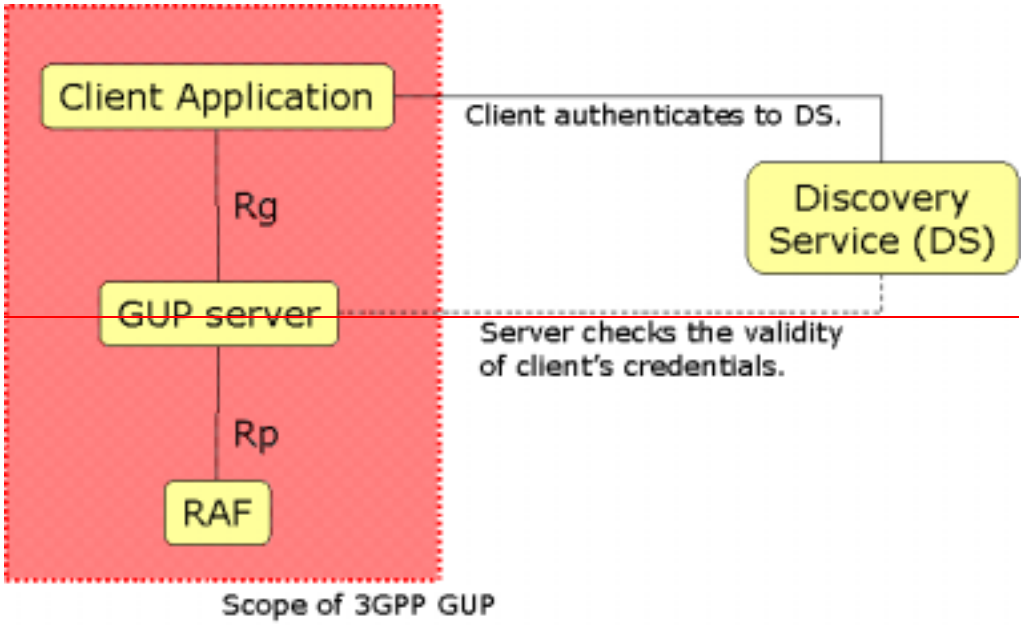
GUP Architecture and Protocols include secure mechanisms for the transfer of User Profile data to, from and between authorised entities. Access to User Profile data shall only be permitted in an authorised and secure manner.

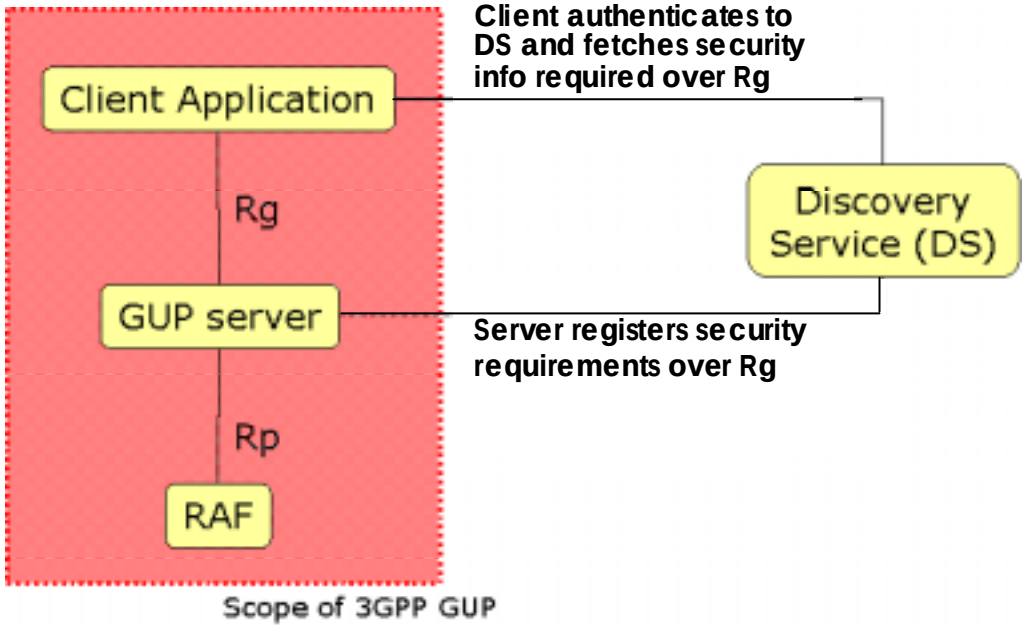
Security mechanisms to be applied over Rg and Rp reference points are defined by Liberty ID-WSF Security Mechanisms ~~Specification~~ [15] specification.

In addition to managing the registration and discovery of GUP Server and profile information a **Discovery Service** (DS) as defined in Liberty ID-WSF Discovery Service Specification [17] may also act as a centralized policy information and decision point issuing the necessary Authentication and/or Authorization statements.

Interactions of GUP requestors towards the Discovery Service shall be as described in Liberty ID-WSF Discovery Service Specification [17].

The GUP architecture with respect to Authentication, Authorization and Security is summarized in Figure XXX.





~~Liberty ID-WSF Security Mechanisms Specification [15] defines a set of combinations of peer authentication and message authentication mechanisms necessary to accommodate various deployment scenarios. Each combination is defined by a URI with the form *urn:liberty:security:2003-08:peer-mechanism:message-mechanism*.~~

Liberty ID-WSF Security Mechanisms Specification [15], Liberty ID-WSF SOAP Binding Specification [14] and Liberty ID-WSF Discovery Service Specification [17] provide normative content for the implementation of the security mechanisms that apply to GUP.

The following chapters under this section of the specification define how the different security mechanisms described in the referred Liberty specifications shall be used in the context of GUP.

The following chapters specify the required set of security mechanisms that shall be supported for GUP.

10.2 Security and Authentication

10.2.1 Rg interface (client application / GUP server)

GUP specifications shall be applicable for multiple deployment scenarios where different security, privacy and trust considerations apply. The applications that may apply GUP Rg reference point may be targeted for different purposes e.g. third party applications for value added services or operator´s own applications for subscription management.

Liberty Alliance Security Mechanisms [15] specification defines a set of combinations of peer authentication and message authentication mechanisms necessary to accommodate various deployment scenarios. Each combination is defined by a URI with the form *urn:liberty:security:2003-08:peer mechanism:message mechanism*. It is up to the security policy of the operator to choose which combination of methods to apply taking into account the security domains where the client and server reside.

~~The required security mechanism for the Rg interface is **urn:liberty:security:2003-08:TLS:SAML**.~~

- ~~The client and the server communicate over a secure channel protected by SSL/TLS (see sub-clause 12.2.3 for list of supported cipher suites).~~
- ~~The server is authenticated to the client using a x.509 server-side certificate.~~
- ~~The client is authenticated to the server using the SAML assertion issued by the Discovery Service and carried in the GUP wsse:Security header (see sub-clause 9.2.9).~~

10.2.1.1 Peer Entity Authentication and Transport Layer Channel Protection

The Peer entity authentication mechanisms required for GUP Rg reference point are based on the mechanisms prescribed by Liberty ID-WSF Security Mechanisms [15] specification, which rely upon the inherent security properties supplied by SSL 3.0 [SSL] or TLS 1.0 [RFC2246] sometimes referred to as transport-level security (also including means for its confidentiality and integrity protection).

Confidentiality and Integrity at the transport channel is ensured making use of suitable SSL/TLS cipher suites (see sub-clause 12.2.3 for list of supported cipher suites).

The server is authenticated to the client using a x.509 server-side certificate. In general the support of client-side certificates in the context of GUP is not mandated but mutual authentication of the communicating peers may be also supported.

10.2.1.2 Message Authentication

Third party applications external to the Mobile Network Operator domain shall only apply Rg reference point. More precisely, third party applications external to the Mobile Network Operator domain acting as a GUP Requestor over the Rg reference point shall at least support *urn:liberty:security:2003-08:TLS:SAML* as defined by Liberty Alliance Security Mechanisms [15] Specification. The support of other message authentication mechanisms (e.g. *urn:liberty:security:2004-04:TLS:Bearer* and/or *urn:liberty:security:2003-08:TLS:X509*) is optional for third party applications external to the Mobile Network Operator domain. However the use of the *null* message authentication profile (e.g. *urn:liberty:security:2003-08:TLS:null*) is not recommended for the Rg reference point.

Internal applications to the Mobile Network Operator domain acting as a GUP Requestor over the Rg reference point shall at least support *urn:liberty:security:2003-08:TLS:null* as defined by Liberty Alliance Security Mechanisms [15] Specification. The support of other message authentication mechanisms (e.g. *urn:liberty:security:2003-08:TLS:SAML*, *urn:liberty:security:2004-04:TLS:Bearer* and/or *urn:liberty:security:2003-08:TLS:X509*) is optional for Internal applications to the Mobile Network Operator domain.

GUP Server shall at least support *urn:liberty:security:2003-08:TLS:null* and *urn:liberty:security:2003-08:TLS:SAML* as defined by Liberty Alliance Security Mechanisms [15] Specification. The support of other message authentication mechanisms (e.g. *urn:liberty:security:2004-04:TLS:Bearer* and/or *urn:liberty:security:2003-08:TLS:X509*) is optional for the GUP Server.

GUP requestors over Rg reference point may utilise a discovery service as a Trusted Authority providing essential security related information (e.g. preferences in terms of peer entity and message authentication mechanism to be used and authentication and/or authorization assertions). Different policies may be followed in the use of the services of a discovery service. It may be used by different applications in different ways: per each operation, occasionally or not at all. In general terms, third party applications belonging to external security domains shall need to use the services of a discovery service as a normal step, but in operator's services it may not be needed at all.

A Discovery Service as defined by Liberty Alliance Discovery Service [17] specification is able to inform GUP requestors of the peer entity and message authentication mechanisms to be used. Additionally, and in the event that a particular deployment of GUP requires the use of the Web Services SAML security profile (e.g. *urn:liberty:security:2003-08:TLS:SAML* or *urn:liberty:security:2003-08:ClientTLS:SAML)* the Discovery Service shall provide the necessary Authentication assertions as defined by Liberty Alliance Security Mechanisms [15] specification. A Discovery Service may also be capable of providing necessary bearer tokens in the event the Bearer token security profile (e.g. *urn:liberty:security:2004-04:TLS:Bearer* or *urn:liberty:security:2004-04:ClientTLS:Bearer*) is used.

~~The required security mechanism for the Rg interface is **urn:liberty:security:2003-08:TLS:SAML**.~~

- ~~The client and the server communicate over a secure channel protected by SSL/TLS (see sub-clause 12.2.3 for list of supported cipher suites).~~
- ~~The server is authenticated to the client using a x.509 server-side certificate.~~
- ~~The client is authenticated to the server using the SAML assertion issued by the Discovery Service and carried in the GUP wsse:Security header (see sub-clause 9.2.9).~~

10.2.2 Rp interface (GUP server / RAF)

The required security mechanism for the Rp interface is **urn:liberty:security:2003-08:~~Client~~TLS:null**.

- The server and the RAF communicate over a secure channel protected by SSL/TLS (see 12.2.3 for list of supported cipher suites).

~~- The server is authenticated to the RAF using a x.509 client-side certificates.~~
~~- The RAF is authenticated to the server using a x.509 server-side certificate.~~

NOTE: Since the number of RAFs is limited and the connections between the server and the RAFs are long-lived (multiple requests sent on the same connections), this should not create too much overhead for either key management or cryptographic processing.

The support of other message authentication mechanisms (e.g. urn:liberty:security:2003-08:TLS:SAML, urn:liberty:security:2004-04:TLS:Bearer and/or urn:liberty:security:2003-08:TLS:X509) is optional for the Rp reference point.

10.2.3 Cryptographic requirements

~~The security infrastructure requires the following cryptographic entities:~~

~~1.existence of one or more x.509 certification authorities (CA)~~
~~2.existence of a certificate for each GUP server signed by a valid CA~~
~~3.existence of a certificate for each RAF signed by a valid CA~~
~~4.public knowledge and public access to the various certificates~~

~~Note:~~ For signing and verification of protocol messages, communicating entities SHOULD use certificates and private keys that are distinct from the certificates and private keys applied for SSL or TLS channel protection (in accordance to Liberty Alliance Security Mechanisms [15] specification). ~~[from liberty-idwsf-security-mechanisms-v1.1]~~

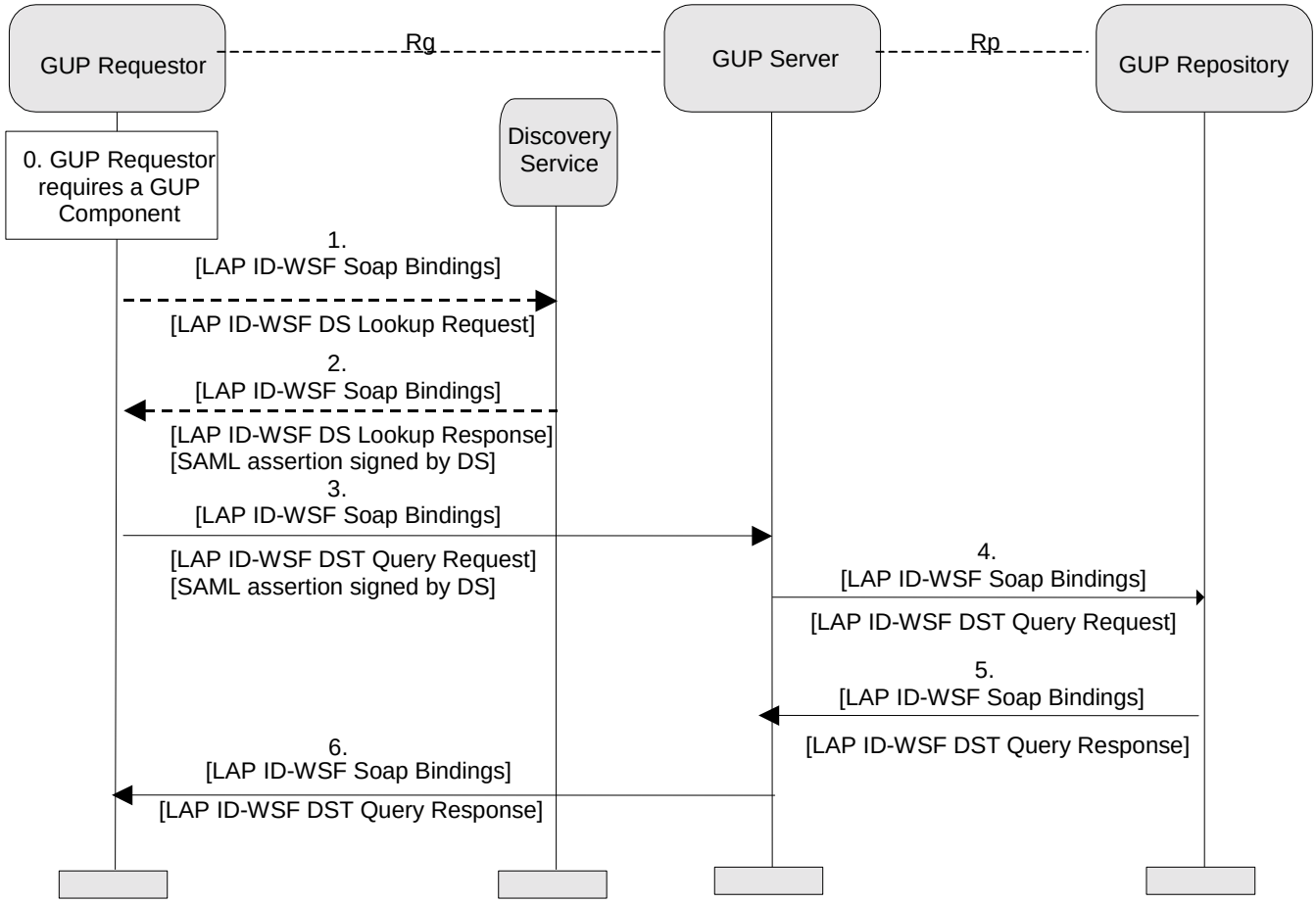
The cipher suites to be used for peer-wise encryption are:

- TLS_RSA_WITH_RC4_128_SHA
- TLS_RSA_WITH_3DES_EDE_CBC_SHA
- TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA
- TLS_RSA_WITH_AES_CBC_SHA
- TLS_DHE_DSS_WITH_AES_CBC_SHA

~~Editor's note: the choice of a set of mandated ciphers for GUP is left FFS.~~GUP entities shall at least support TLS_RSA_WITH_3DES_EDE_CBC_SHA and TLS_RSA_WITH_AES_CBC_SHA cipher suites. The support of other cipher suites is optional.

10.2.4 10.2.4 End-to-End Example (informative)

The following diagram represents a possible interaction between a client application and a GUP server.


GUP Requestor
Rg
GUP Server
Rp
GUP Repository
Discovery Service
0. GUP Requestor requires a GUP Component
1.
[LAP ID-WSF Soap Bindings]
[LAP ID-WSF DS Lookup Request]
2.
[LAP ID-WSF Soap Bindings]
[LAP ID-WSF DS Lookup Response]
[SAML assertion signed by DS]
3.
[LAP ID-WSF Soap Bindings]
[LAP ID-WSF DST Query Request]
[SAML assertion signed by DS]
4.
[LAP ID-WSF Soap Bindings]
[LAP ID-WSF DST Query Request]
5.
[LAP ID-WSF Soap Bindings]
[LAP ID-WSF DST Query Response]
6.
[LAP ID-WSF Soap Bindings]
[LAP ID-WSF DST Query Response]

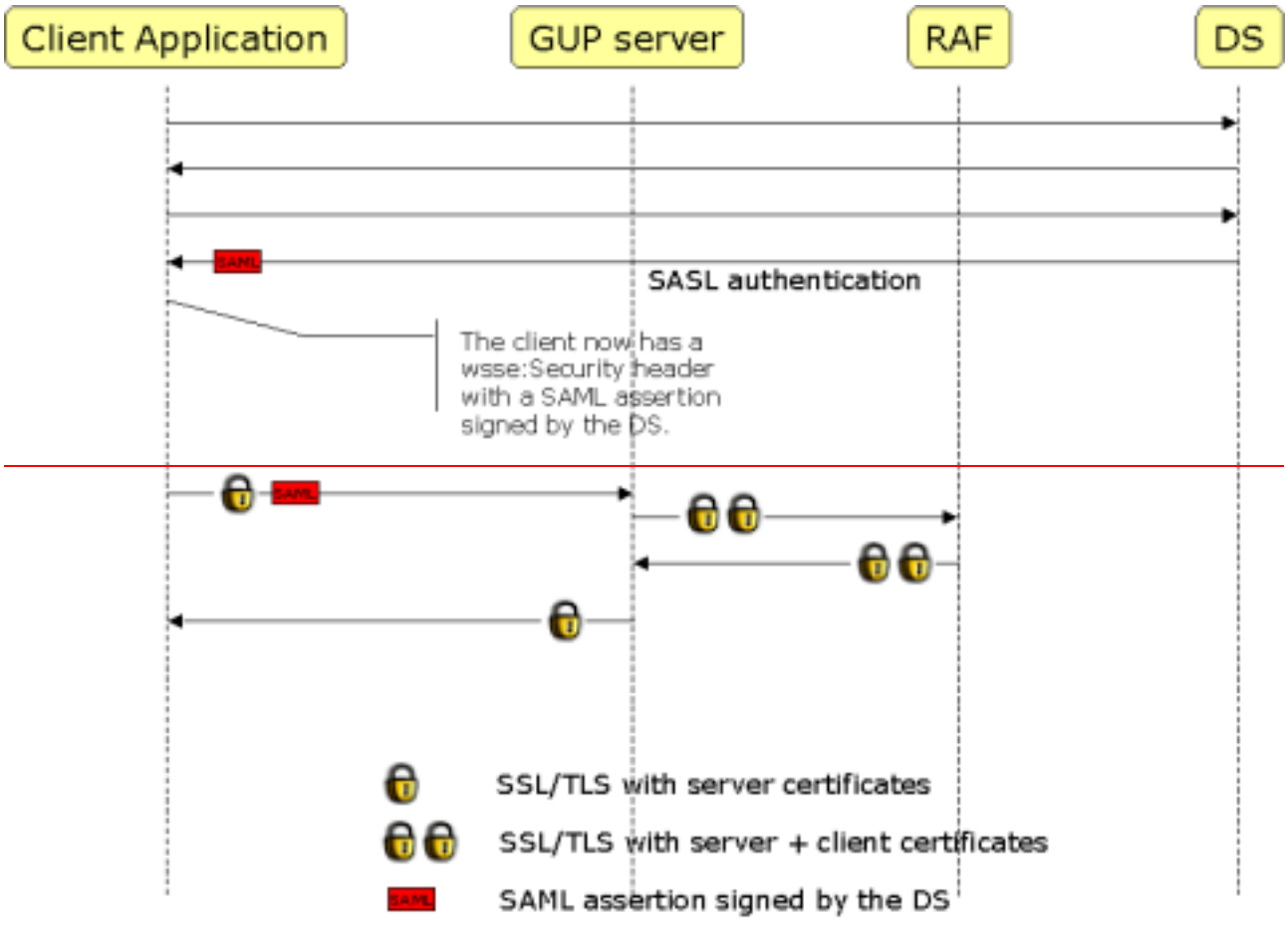

Client Application
GUP server
RAF
DS
SAML
SASL authentication
The client now has a wsse:Security header with a SAML assertion signed by the DS.
SAML
SSL/TLS with server certificates
SSL/TLS with server + client certificates
SAML
SAML assertion signed by the DS

Figure 2x: GUP security flow

10.2.5 Example of GUP wsse:Security header (informative)

The following header authenticates the application defined by `application1@3gpp.org`.

```
<wsse:Security>
<saml:Assertion
xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
MajorVersion="1" MinorVersion="0"
AssertionID="2sxJu9g/vvLG9sAN9bKp/8q 0NKU="
Issuer="DS.example.com"
IssueInstant="2004-04-01T16:58:33.173Z">

<!-- By placing an audience restriction on the assertion we
can limit the scope of which entity should consume
the information in the assertion. -->

<saml:Conditions
NotBefore="2004-04-01T16:57:20Z"
NotOnOrAfter="2004-04-01T21:42:43 Z">
<saml:AudienceRestrictionCondition>
<saml:Audience>http://gupserver.com</sa ml:Audience>
</saml:AudienceRestrictionConditi on>
</saml:Conditions>

<!-- The AuthenticationStatement carries information
that describes the identity of the entity this assertion
was issued to (the Subject) and the method the Subject
authenticated to the assertion issuing authority -->

<saml:AuthenticationStatement
AuthenticationMethod="urn:ietf:rfc:2246"
AuthenticationInstant="2004-04-01T16:57:30.000Z">
<saml:Subject>
<saml:NameIdentifier
format="urn:liberty:iff:nameid:entityID"
NameQualifier="http://AffiliationStation.com /">
http:// application1@3gpp.org
</saml:NameIdentifier>
</saml:Subject>
</saml:AuthenticationStatement>
<!-- signature by the authority over the assertion -->
<ds:Signature>...</ds:Signature>
</saml:Assertion>

<!-- this is the reference to the bearer token -->
<wsse:SecurityTokenReference>
<wsse:Reference URI="#2sxJu9g/vvLG9sAN9bKp/8q0NKU="
ValueType="saml:Assertion" />
</wsse:SecurityTokenReference>
</wsse:Security>
<wsse:Security>
  <saml:Assertion>
    <saml:Conditions>
<!
      e.g. not before now, not after 5 minutes from now
      only for GUP servers from provider.com

-->
    </saml:Conditions>
    <saml:AuthenticationStatement>
      <saml:Subject>application1@3gpp.org</saml:Subject>
    </saml:AuthenticationStatement>
    <ds:Signature>
<!-- Digital signature of the whole assertion, signed by the DS -->
    </ds:Signature>
  </saml:Assertion>
<wsse:Security>
```

3GPP TSG-CT WG4 Meeting #27
Cancun, Mexico. 25th to 29th April 2005.

Tdoc ⌘C4-050769 ***(Rev of C4-050691)***

CR-Form-v7.1

CHANGE REQUEST

⌘ **29.240** **CR** **003** ⌘**rev** **1** ⌘ Current version: **6.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network ☐ Core Network **X**

Title:	⌘	GUP HSS-IMS Component Definition
Source:	⌘	Ericsson
Work item code:	⌘	GUP
Date:	⌘	05/04/2005
Category:	⌘	**F**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

Ph2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*
Rel-7	*(Release 7)*

Reason for change: ⌘ As shown in the accompanying discussion paper, the current structure of the GUP HSS IMS component is not correct. Some information elements are missing and some information needs to be removed from the model described in the annex A.

There is a risk related to consistency maintenance with the specifications were the data are specified, as some descriptions are duplicated.

The DST tables in the version 6.0.0 are empty.

Summary of change: ⌘ The content of the data model described in the annex has been changed. Appropriate references to the TSs where the data are specified have been added.

Mainly the content of the Annex A has been changed, adding missing data such as Charging inforamtion and removing other info such as Authentication data.

The details of the specific attribute part of the HSS IMS component will not be described in this annex, they will be reffered to the correct TS.

The UML model included in the Normative annex is proposed to be kept as an example.

DST Tables are completed.

Consequences if not approved:	⌘	The GUP IMS Component would contain some elements that are of no interest to applications and at the same time it would miss some that are interesting. A risk for consistency problem would remain. DST tables would not be completed.

Clauses affected:	⌘	Annex <A>			
		Y	**N**		
Other specs affected:	⌘		**X**	Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	
Other comments:	⌘				

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

>>>>>>>>>>> First modified section <<<<<<<<<<<

Annex A (normative):

Component Data Definitions

A.1 HSS IMS GUP Component definition

The purpose of this annex is to provide GUP Profile Component definition for the IMS data of the HSS.

A.1.1 General description

The following Figure A.3 gives an outline of the UML model of the logical view of the HSS IMS GUP Components. The main Component is called HSSIMSData.

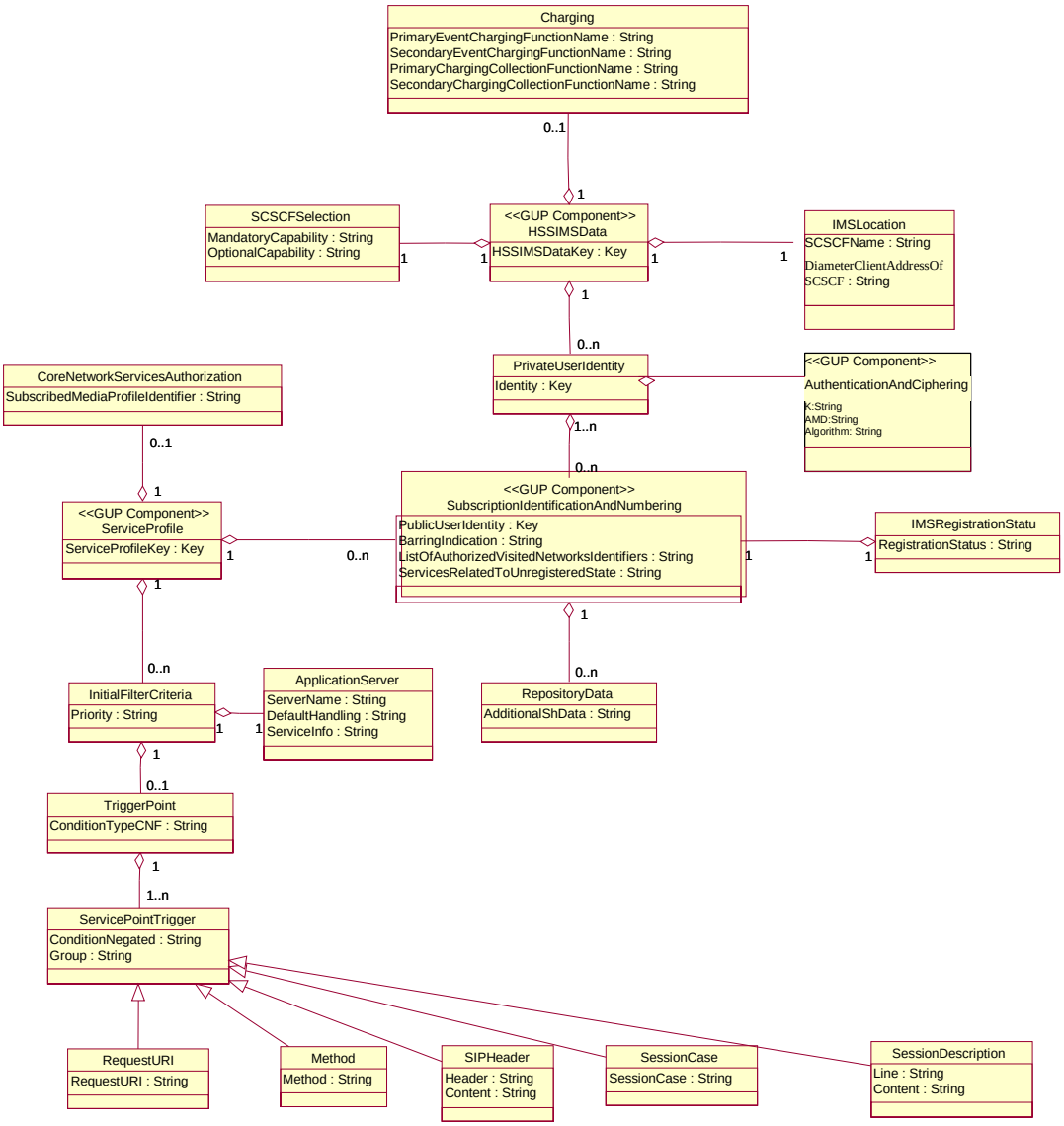
Each element in that figure represents a part of the XML Schema structure, either a GUP Component or a lower level block of data contained in a GUP Component. Elements marked with the same background colour make up an independently manageable GUP Component, whose root is marked with '<<GUP Component>>'.

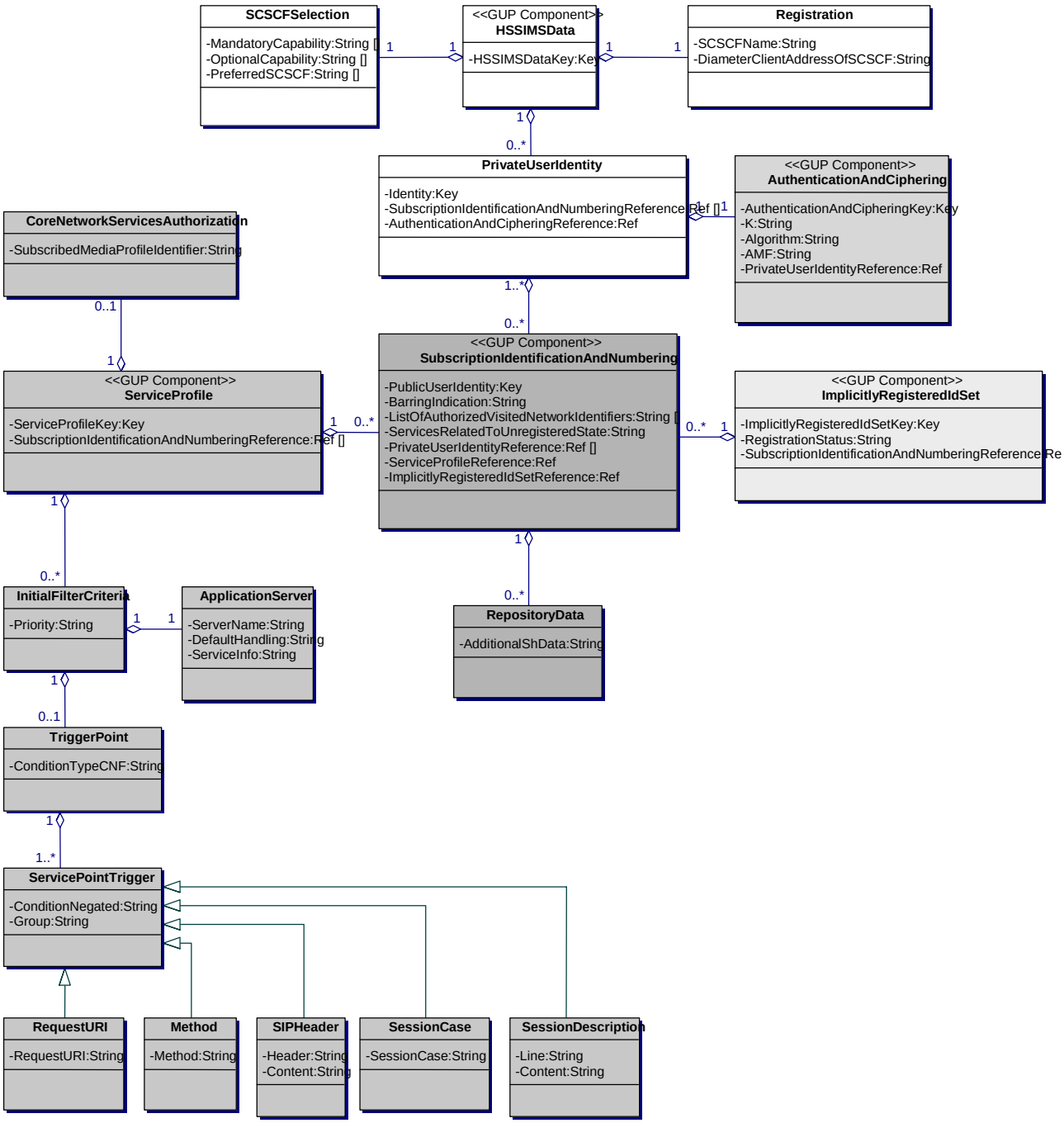
All HSS IMS GUP Components can be managed with the procedures provided by GUP.

Component and attribute naming has been taken mainly from 3GPP TS 23.008 [8] where appropriate. The details about the specific attributes shown in the picture are described in others specs. The ones related Cx protocol are described in the 3GPP TS 29.228 [7] and the ones related Sh interface are described in the 3GPP TS 29.328 [10]. ~~If a name consists of more than one word or abbreviated word, capitalization is used to improve the readability of long names. Each word begins with capital letter.~~

~~If Component does not have a unique key of its own from the reference specification, it has been created for it.~~

The identified set of GUP data elements with regard to the IMS HSS component should receive different treatment (only read or both read/write access rights), depending on the nature of the data and the nature of the application requesting the data (e.g. provisioning application). In order to provide such differentiated treatment access control mechanisms shall be applied. These access control mechanisms should take into consideration the rights that can be offered over each attribute.


Charging
PrimaryEventChargingFunctionName : String
SecondaryEventChargingFunctionName : String
PrimaryChargingCollectionFunctionName : String
SecondaryChargingCollectionFunctionName : String
SCSCFSelection
MandatoryCapability : String
OptionalCapability : String
<<GUP Component>>
HSSIMSData
HSSIMSDataKey : Key
IMSLocation
SCSCFName : String
DiameterClientAddressOf
SCSCF : String
PrivateUserIdentity
Identity : Key
<<GUP Component>>
AuthenticationAndCiphering
K:String
AMD:String
Algorithm: String
CoreNetworkServicesAuthorization
SubscribedMediaProfileIdentifier : String
<<GUP Component>>
ServiceProfile
ServiceProfileKey : Key
<<GUP Component>>
SubscriptionIdentificationAndNumbering
PublicUserIdentity : Key
BarringIndication : String
ListOfAuthorizedVisitedNetworksIdentifiers : String
ServicesRelatedToUnregisteredState : String
IMSRegistrationStatu
RegistrationStatus : String
InitialFilterCriteria
Priority : String
ApplicationServer
ServerName : String
DefaultHandling : String
ServiceInfo : String
RepositoryData
AdditionalShData : String
TriggerPoint
ConditionTypeCNF : String
ServicePointTrigger
ConditionNegated : String
Group : String
RequestURI
RequestURI : String
Method
Method : String
SIPHeader
Header : String
Content : String
SessionCase
SessionCase : String
SessionDescription
Line : String
Content : String


SCSCFSelection
-MandatoryCapability:String []
-OptionalCapability:String []
-PreferredSCSCF:String []
<<GUP Component>>
HSSIMSData
-HSSIMSDataKey:Key
Registration
-SCSCFName:String
-DiameterClientAddressOfSCSCF:String
PrivateUserIdentity
-Identity:Key
-SubscriptionIdentificationAndNumberingReference:Ref []
-AuthenticationAndCipheringReference:Ref
<<GUP Component>>
AuthenticationAndCiphering
-AuthenticationAndCipheringKey:Key
-K:String
-Algorithm:String
-AMF:String
-PrivateUserIdentityReference:Ref
CoreNetworkServicesAuthorization
-SubscribedMediaProfileIdentifier:String
<<GUP Component>>
ServiceProfile
-ServiceProfileKey:Key
-SubscriptionIdentificationAndNumberingReference:Ref []
<<GUP Component>>
SubscriptionIdentificationAndNumbering
-PublicUserIdentity:Key
-BarringIndication:String
-ListOfAuthorizedVisitedNetworkIdentifiers:String []
-ServicesRelatedToUnregisteredState:String
-PrivateUserIdentityReference:Ref []
-ServiceProfileReference:Ref
-ImplicitlyRegisteredIdSetReference:Ref
<<GUP Component>>
ImplicitlyRegisteredIdSet
-ImplicitlyRegisteredIdSetKey:Key
-RegistrationStatus:String
-SubscriptionIdentificationAndNumberingReference:Re
InitialFilterCriteria
-Priority:String
ApplicationServer
-ServerName:String
-DefaultHandling:String
-ServiceInfo:String
RepositoryData
-AdditionalShData:String
TriggerPoint
-ConditionTypeCNF:String
ServicePointTrigger
-ConditionNegated:String
-Group:String
RequestURI
-RequestURI:String
Method
-Method:String
SIPHeader
-Header:String
-Content:String
SessionCase
-SessionCase:String
SessionDescription
-Line:String
-Content:String

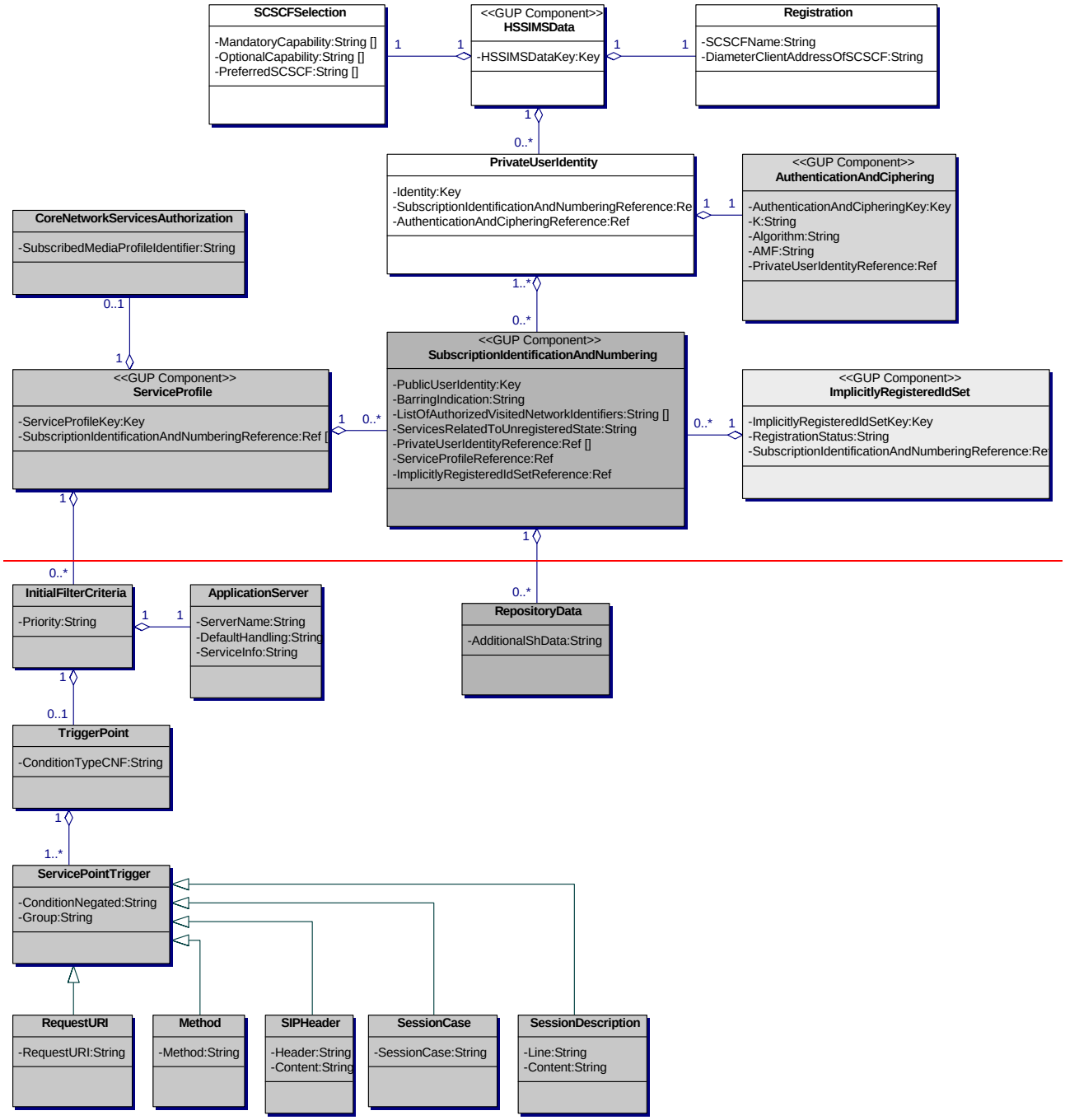


Figure A.3 Logical data model of the HSS IMS GUP Components

A.1.1.1 XML Schema files for HSS IMS GUP Components

XML Schema files attached to this specification combine together whole HSS IMS GUP Component data. XML Schema files are intended to be used by an XML parser.

Table A.1 lists HSS IMS GUP Components and XML Schema files that include those. CommonDataTypes file contains common data types, i.e. it is not a stand alone GUP Component on its own.

Table A.1 HSS IMS GUP Components in XML Schema files

HSS IMS GUP Component	XML Schema file
AuthenticationAndCiphering	AuthenticationAndCiphering.xsd

HSSIMSData	HSSIMSData.xsd
ImplicitlyRegisteredIdSet	ImplicitlyRegisteredIdSet.xsd
SubscriptionIdentificationAndNumbering	SubscriptionIdentificationAndNumbering.xsd
ServiceProfile	ServiceProfile.xsd
Component independent data	**XML Schema file**
-	CommonDataTypes.xsd

A.1.2 Element addressing

Contents for HSS IMS GUP Components can be addressed with XPath representation format as an input parameter for select clauses. Each GUP Component has a unique key.

Example
Following Select element addresses the BarringIndication element in the SubscriptionIdentificationAndNumbering GUP Component. Public User Identity is specified by ResourceID in the procedure message element, e.g. Query.

```
<Select>
    <ComponentType>
            http://3gpp/gup/ns/comp/SubscriptionIdentificationAndNumbering
        </ComponentType>
        <GCLRef>
            //SubscriptionIdentificationAndNumbering/BarringIndication
        </GCLRef>
</Select>
```

~~A.1.3 Relationships of the HSS IMS GUP Components~~

~~Each HSS IMS GUP Component contains a native key element, or if that is not available, an artificial key has been generated for it. Table A.1 shows how HSS IMS GUP Components can refer each other.~~

~~Referencing Component is the referencing GUP Component. Referencing element is the referencing element of that GUP Component.~~

~~Referenced Component is a referenced GUP Component. Referenced key element is the referenced key element of that GUP Component.~~

~~ResourceID Identifier (see 7.1 ResourceID) type describes which type of value is contained in the key. Each instance of these GUP Components can be addressed directly with a proper ResourceID value.~~

~~Table A.2 Relationships of the HSS IMS GUP Components~~

~~**Referencing Component**~~	~~**Referencing element**~~	~~**Referenced Component**~~	~~**Referenced key element**~~	~~**ResourceID Identifier type**~~
~~AuthenticationAndCiphering~~	~~PrivateUserIdentityReference~~	~~HSSIMSData~~	~~Identity of PrivateUserIdentity~~	~~IMS Private Identity~~
~~HSSIMSData~~	~~AuthenticationAndCipheringReference~~	~~AuthenticationAndCiphering~~	~~AuthenticationAndCipheringKey~~	~~Generic Data Reference~~

	~~SubscriptionIdentificationAndNumbering Reference~~	~~SubscriptionIdentificationAndNumbering~~	~~PublicUserIdentity~~	~~IMS Public Identity~~
~~ImplicitlyRegisteredIdSet~~	~~SubscriptionIdentificationAndNumbering Reference~~	~~SubscriptionIdentificationAndNumbering~~	~~PublicUserIdentity~~	~~IMS Public Identity~~
~~SubscriptionIdentificationAndNumbering~~	~~PrivateUserIdentityReference~~	~~HSSIMSData~~	~~Identity of PrivateUserIdentity~~	~~IMS Private Identity~~
	~~ServiceProfileReference~~	~~ServiceProfile~~	~~ServiceProfileKey~~	~~Generic Data Reference~~
	~~ImplicitlyRegisteredIdSetReference~~	~~ImplicitlyRegisteredIdSet~~	~~ImplicitlyRegisteredIdSet Key~~	~~Generic Data Reference~~
~~ServiceProfile~~	~~SubscriptionIdentificationAndNumbering Reference~~	~~SubscriptionIdentificationAndNumbering~~	~~PublicUserIdentity~~	~~IMS Public Identity~~
		~~HSSIMSData~~	~~HSSIMSDataKey~~	~~Generic Data Reference~~

A.1.2 HSS IMS GUP Component structure

HSS IMS GUP Component model includes following Components:

- HSSIMSData
- AuthenticationAndCiphering
- SubscriptionIdentificationAndNumbering

~~- ImplicitlyRegisteredIdSet~~

- ServiceProfile

This model is mainly based on the HSS data described in 3GPP TS 23.008 [8]. Authentication information is described in TS 33.203 [12] and TS 33.102 [11]. Data relationships are mainly based on descriptions from 3GPP TS 23.228 [9], 3GPP TS 29.328 [10] and 3GPP TS 29.228 [7].

~~Editor's note: CAMEL related data is FFS~~

~~Editor's note: Charging related data is FFS~~

A.1.2.1 HSSIMSData GUP Component

HSS IMS Data contains S-CSCF Selection-, ~~Registration~~IMS location, charging- and Private User Identity information.

HSS IMS Data contains also following data for Component linking purposes.

- HSSIMSDataKey
  - HSS IMS Data Key is a primary key to uniquely identify HSS IMS Data Component.
  - This value must be unique within all HSS IMS Data Components.

A.1.2.1.1 SCSCFSelection

S-CSCF Selection contains the following data, which are described in 3GPP TS 23.008 [8].

- MandatoryCapability
- OptionalCapability
- PreferredSCSCF (optional parameter)

These data shall be offered only for provisioning purposes.

A.1.2.1.2 IMS Location~~Registration~~

~~Registration~~ IMS Location contains the following data, which are described in 3GPP TS 23.008 [8].

- SCSCFName
- DiameterClientAddressOfSCSCF (optional parameter)

A.1.2.1.3 PrivateUserIdentity

Private User Identity contains the following data, which is described in 3GPP TS 23.008 [8].

- Identity (Private User Identity)

This data shall be offered only for provisioning purposes.

~~In addition to the data listed above, Private User Identity element contains the following data for component linking purposes.~~

~~- SubscriptionIdentificationAndNumberingReference~~

~~o Subscription Identification And Numbering Reference list links this Private User Identity element to zero or more Public User Identities in Subscription Identification And Numbering GUP Component.~~

~~- AuthenticationAndCipheringReference~~

~~o Authentication And Ciphering Reference links this Private User Identity element to the Authentication And Ciphering GUP Component.~~

A.2.2 AuthenticationAndCiphering GUP Component

~~[Editor's note: to be defined in more detail]~~

Authentication and Ciphering contains the following data, which are described in 3GPP TS 33.102 [11].

- K
- Algorithm
- AMF

K parameter can contain either Secret Key or Encrypted Secret Key value. The encryption algorithm is implementation specific.

These data shall be offered only for provisioning purposes.

~~In addition to data listed above, Authentication And Ciphering Component contains the following data for component linking purposes.~~

~~- AuthenticationAndCipheringKey~~

~~o Authentication And Ciphering Key is a primary key to uniquely identify Authentication And Ciphering Component.~~

~~- PrivateUserIdentityReference~~

- Private User Identity Reference links this Authentication And Ciphering GUP Component to one Identity element of the Private User Identity element in HSS IMS Data GUP Component.

A.1.2.3 SubscriptionIdentificationAndNumbering GUP Component

Subscription Identification And Numbering Component contains the following data, which are described in 3GPP TS 23.008 [8].

- PublicUserIdentity
- BarringIndication (This data shall be offered only for provisioning purposes)
- ListOfAuthorizedVisitedNetworkIdentifiers (This data shall be offered only for provisioning purposes)
- ServicesRelatedToUnregisteredState

Public User Identity is a primary key element for the Subscription Identification And Numbering Component.

In addition to the data listed above, Subscription Identification And Numbering Component contains the following data for component linking purposes.

- PrivateUserIdentityReference
  - Private User Identity Reference list links this Subscription Identification And Numbering Component to one or more Identity elements of the Private User Identity element in HSS IMS Data GUP Component.
- ServiceProfileReference
  - Service Profile Reference links this Subscription Identification And Numbering Component to a Service Profile Key element in Service Profile GUP Component.
- ImplicitlyRegisteredIdSetReference
  - Implicitly Registered Id Set Reference links this Subscription Identification And Numbering Component to an Implicitly Registered Id Set Key element in Implicitly Registered Id Set GUP Component.

A.1.2.3.1 RepositoryData

Repository Data contains additional Sh-interface data, which are described in 3GPP TS 29.328 [10].

- AdditionalShData

Editor's note: to be defined in more detail

Editor's note: CAMEL related data is FFS

A.1.2.4 ServiceProfile GUP Component

The present sub clause presents the Service Profile GUP Component contents, which are defined based onin the 3GPP TS 29.228 [7].

There exist some naming convention differences between Service Profile GUP Component XML Schema and XML Schema for Cx interface described in 3GPP TS 29.228 [7]. GUP XML Schema uses data type in a format XxxType, where XML Schema for Cx interface uses format tXxx.

E.g. corresponding GUP data element type for ServiceProfileType is tServiceProfile - Cx element type. This rule is valid to data types under ServiceProfileType - data type.

Service Profile contains the following data for Component linking purposes.

- ServiceProfileKey
  - Service Profile Key is a primary key to uniquely identify Service Profile Component.

~~oThis value must be unique within all Service Profile data.~~

~~- SubscriptionIdentificationAndNumberingReference~~

~~oSubscription Identification And Numbering Reference list links this Service Profile Component to zero or more Public User Identity element in Subscription Identification And Numbering Components.~~

A.1.2.4.1 CoreNetworkServiceAuthorization

Core Network Service Authorization contains the following data, which are described in 3GPP TS 29.228 [7].

- SubscribedMediaProfileIdentifier

This data shall be offered only for provisioning purposes

A.1.2.4.2 InitialFilterCriteria

Initial Filter Criteria contains the following data, which are described in 3GPP TS 29.228 [7]. All data under Initial Filter Criteria are described in 3GPP TS 29.228 [7].

- Priority

A.1.2.4.2.1 ApplicationServer

The present sub clause presents the Application Server element contents, which are defined based on the 3GPP TS 29.228 [7].

- ServerName
- DefaultHandling
- ServiceInfo

A.1.2.4.2.2 TriggerPoint

The present sub clause presents the Trigger Point element contents, which are defined based on the 3GPP TS 29.228 [7].

- ConditionTypeCNF

A.1.2.4.2.2.1 ServicePointTrigger

The present sub clause presents the Service Point Trigger element contents, which are defined based on the 3GPP TS 29.228 [7].

Corresponding data element in Cx reference point is SePoTri.

- ConditionNegated
- Group

Service Point Trigger contains one of the following data structures described below.

A.1.2.4.2.2.1.1 RequestURI

The present sub clause presents the Request URI element contents, which are defined based on the 3GPP TS 29.228 [7].

- RequestURI

A.1.2.4.2.2.1.2 Method

The present sub clause presents the Method element contents, which are defined based on the 3GPP TS 29.228 [7].

- Method

A.1.2.4.2.2.1.3 SIPHeader

The present sub clause presents the SIP Header element contents, which are defined based on the 3GPP TS 29.228 [7].

- Header
- Content

A.1.2.4.2.2.1.4 SessionCase

The present sub clause presents the Session Case element contents, which are defined based on the 3GPP TS 29.228 [7].

- SessionCase

A.1.2.4.2.2.1.5 SessionDescription

The present sub clause presents the Session Description element contents, which are defined based on the 3GPP TS 29.228 [7].

- Line
- Content

~~A.2.5 ImplicitlyRegisteredIdSet GUP Component~~

~~Implicitly Registered Id Set contains the following data, which is described in 3GPP TS 29.228 [7].~~

- ~~RegistrationStatus~~

~~In addition to data listed above, Implicitly Registered Id Set Component contains the following data for component linking purposes.~~

- ~~ImplicitlyRegisteredIdSetKey~~
  - ~~oImplicitly Registered Id Set Key is a primary key to uniquely identify Implicitly Registered Id Set Component.~~
  - ~~oThis value must be unique within all Implicitly Registered Id Set data.~~
- ~~SubscriptionIdentificationAndNumberingReference~~
  - ~~oSubscription Identification And Numbering Reference list links this Implicitly Registered Id Set Component to zero or more Public User Identity element in Subscription Identification And Numbering Components.~~

A.1.3 Common Data Types

GUP Components share some common data types defined in the CommonDataTypes.xsd - XML Schema file.

That XML Schema file includes the following elements or element types used by other GUP XML Schema files:

- Extension
  - Will be used as an extension element
- ExtensionType
  - Extension element type
- GenericDataType
  - Will be used as a generic data type
- GenericComponentReferenceType

- Will be used as a generic Component reference type

A.1.4 Data Services Template Checklist tables

GUP is an instance of a data service as described by Liberty Alliance Data Services Template [13] specification and all its stipulations are hereby incorporated unless expressly waived or modified in this specification.

This section presents the checklist tables as defined in Liberty Alliance Data Service Template [13] specification filled-in with the values corresponding of the use of defined ~~HSS-IMS~~ GUP Data components.

Table A.1.4.1/1: General Service Parameters (1/2)

Parameter	Value
<ServiceType>	urn:3gpp:gup-hss-ims:2005-04
Discovery Options	Discovery Options MAY be supported If a Discovery Service is employed, the following Discovery Option Keywords as defined by Liberty ID-WSF Data Services Template Specification [13] MAY be used: urn:liberty:dst:allPaths urn:liberty:dst:can:extend urn:liberty:dst:changeHistoryS upported urn:liberty:dst:extend urn:liberty:dst:fullXPath urn:liberty:dst:multipleResources urn:liberty:dst:multipleQueryItems urn:liberty:dst:multipleModification urn:liberty:dst:noModify urn:liberty:dst:noPagination urn:liberty:dst:noQuery urn:liberty:dst:noQuerySubscriptions urn:liberty:dst:noSorting urn:liberty:dst:noStatic urn:liberty:dst:noSubscribe No specific Discovery Option Keyword is defined for the HSS-IMS GUP Data Componet.
Data Schema	See Apendix C in this specification
SelectType Definition	The SelectType definition for the HSS-IMS GUP data component is described in the subclause 9.3.2.
Semantics of the <Select> element	See Chapter 6.5 in this specification

Table A.1.4.1/2: General Service Parameters (2/2)

Parameter	Value
Element uniqueness	Not applicable. No multiple elements with same name used.
Data Extension Supported	The <Extension> element MAY be used to add additional HSS-IMS GUP elements, but the use is not specified by this specification.

Table A.1.4.2/1: Query Parameters (1/2)

Parameter	Value
Support querying	Queries MUST be supported
Multiple <Query> elements	Multiple <Query> elements MAY be supported.
Multiple <QueryItem> elements	Multiple <QueryItem> elements MAY be supported.
Support sorting	Not Applicable

SortType definition	Not Applicable as sorting is not supported, empty definition used: <xs:complexType name="SortType"> <xs:complexContent> <xs:restriction base="EmptyType"/> </xs:complexContent> </xs:complexType>
Support changedSince	changedSince MAY be supported.
Supported formats	All

Table A.1.4.2/2: Query Parameters (2/2)

Parameter	Value
Support includeCommon Attributes	Common Attributes MUST be supported. See Chapter 6.4 in this specification.
Support pagination	Not Applicable. MUST NOT be used.
Support static sets	Not Applicable. MUST NOT be used.
<Extension> in <Query>	The <Extension> element in <Query> MAY be used for new parameters, but the use is not specified by this specification.

Table A.1.4.3: Modify Parameters

Parameter	Value
Support modification	Modifications MUST be supported.
Multiple <Modify> elements	Multiple <Modify> elements MAY be supported.
Multiple <Modification> elements	Multiple <Modification> elements MAY be supported.
Support partial success	Partial Success MAY be supported.
Support notChangedSince	notChangedSince MAY be supported.
<Extension> in <Modify>	The <Extension> element in <Modify> MAY be used to specify new parameters, but the use is not specified by this specification.

Table A.1.4.4/1: Subscribe Parameters (1/2)

Parameter	Value
Support subscribing to notifications	Subscriptions to Notifications MUST be supported.
Use of the <Subscribe> element for modifying and renewing subscriptions.	Creation, Renewal, Cancellation and Modification of subscriptions MUST be supported.
Multiple <Subscribe> elements	Multiple <Subscribe> elements MAY be supported.
Multiple <Subscription> elements	Multiple <Subscription> elements MAY be supported.
Use of the <NotifyEndedTo> element	The <NotifyEndedTo> element MUST be supported, if end notifications are used.
TypeType definition	The <Type> element is not used, so an empty definition is used for it: <xs:complexType name="TypeType"> <xs:complexContent> <xs:restriction base="EmptyType"/> </xs:complexContent> </xs:complexType>

Table A.1.4.4/2: Subscribe Parameters (2/2)

Parameter	Value

`TriggerType` definition	The `<Trigger>` element is not used for the HSS-IMS GUP Data Component (just default change notifications are supported).
	An empty type definition shall be used:
	`<xs:complexType name="TriggerType">`
	`  <xs:complexContent>`
	`  <xs:restriction base="EmptyType"/>`
	`  </xs:complexContent>`
	`</xs:complexType>`

Start of a subscription	`starts` attribute MUST be supported.
Subscription expiration	Subscription expiration MUST be used.
Use of `expires` and `duration` attributes	Both `expires` and `duration` MUST be supported.
Support `expires==starts`	The same value for both the `starts` and the `expires` attribute MUST be supported.
Support zero `duration`	The value zero MUST be supported for the `duration` attribute.
`<Extension>` in `<Subscribe>`	The `<Extension>` element in `<Subscribe>` MAY be used for new parameters, but the use is not specified by this specification.

Table A.1.4.5: QuerySubscriptions Parameters

Parameter	Value
Support querying existing subscriptions	Query Subscriptions MAY be supported.
Multiple `<QuerySubscriptions>` elements	Multiple `<QuerySubscriptions>` elements MAY be supported.
`<Extension>` in `<QuerySubscrip`	`<Extension>` element in `<QuerySubscriptions>` MAY be used for new parameters, but the use is not specified by this specification.

Table A.1.4.6: Notify Parameters

Parameter	Value
Support notifications	Notifications MUST be supported.
Are notifications acknowledged	End notifications MUST be acknowledged.
`<Extension>` in `<Notify>`	The `<Extension>` element in `<Notify>` MAY be used to pass additional data, but the use is not specified by this specification.

Table A.1.4.7: EndNotify Parameters

Parameter	Value
Support end notifications	End notifications MAY be supported.
Are end notifications acknowledged	End notifications SHOULD be acknowledged.
`<Extension>` in `<Ended>`	The `<Extension>` element in `<Ended>` MAY be used to pass additional data, but the use is not specified by this specification.

3GPP TSG-CT WG4 Meeting #27 ***Tdoc ⌘C4-050771(Rev of C4-050696)***
Cancun, Mexico. 25th to 29th April 2005.

CR-Form-v7.1

CHANGE REQUEST

⌘ **29.240** **CR** **006** ⌘**rev** **1** ⌘ Current version: **6.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network ☐ Core Network **X**

Title:	⌘	GUP SOAP Headers
Source:	⌘	Ericsson
Work item code:	⌘	GUP
Date:	⌘	15/04/2005
Category:	⌘	**F**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

Ph2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*
Rel-7	*(Release 7)*

Reason for change:	⌘	Description of GUP SOAP Headers is currently duplicated in 7.2 and Annex E. In the 7.2 there should only be a generic description that can be used for any binding, not just to SOAP.
Summary of change:	⌘	Description of GUP SOAP Headers is consolidated in the Annex E.
Consequences if not approved:	⌘	Unclear description of the SOAP binding for GUP.

Clauses affected:	⌘	7.2, Annex E			
Other specs affected:	⌘	**Y**	**N**		
			X	Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	
Other comments:	⌘				

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.
2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.
3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

5 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 23.240: "3GPP Generic User Profile - Architecture; Stage 2

[2] "Web Services Description Language (WSDL) 1.1," Christensen, Erik, Curbera, Francisco, Meredith, Greg, Weerawarana, Sanjiva, eds. World Wide Web Consortium W3C Note (15 March 2001). http://www.w3.org/TR/2001/NOTE-wsdl-20010315

[3] Thompson, H.S., Beech, D., Maloney, M., Mendleshon, N., eds. (May 2002). "XML Schema Part 1: Structures," Recommendation, World Wide Web Consortium http://www.w3.org/TR/xmlschema-1/

[4] Biron, P.V., Malhotra, A., eds. (May 2002). "XML Schema Part 2: Datatypes," Recommendation, World Wide Web Consortium http://www.w3.org/TR/xmlschema-2/

[5] "Simple Object Access Protocol (SOAP) 1.1," Box, Don, Ehnebuske, David , Kakivaya, Gopal, Layman, Andrew, Mendelsohn, Noah, Nielsen, Henrik Frystyk, Thatte, Satish, Winer, Dave, eds. World Wide Web Consortium W3C Note (08 May 2000). http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

[6] RFC 2616 (June 1999): "Hypertext Transfer Protocol – HTTP/1.1"

[7] 3GPP TS 29.228: "IP Multimedia (IM) Subsystem Cx and Dx interfaces; Signalling flows and message contents"

[8] 3GPP TS 23.008: "Organization of subscriber data"

[9] 3GPP TS 23.228: "IP Multimedia Subsystems (IMS); Stage 2"

[10] 3GPP TS 29.328: "IP Multimedia (IM) Subsystem Sh interface; Signalling flows and message contents"

[11] 3GPP TS 33.102: "3G Security; Security architecture"

[12] 3GPP TS 33.203: "3G security; Access security for IP-based services"

[13] "Liberty ID-WSF Data Services Template Specification", Liberty Alliance Project. http://www.projectliberty.org/specs/draft-liberty-idwsf-dst-v2.0-01.pdf (draft)

[14] "Liberty ID-WSF SOAP Binding Specification", Liberty Alliance Project. http://www.projectliberty.org/specs/liberty-idwsf-soap-binding-v1.1.pdf

[15] "Liberty ID-WSF Security Mechanisms Specification", Liberty Alliance Project. http://www.projectliberty.org/specs/liberty-idwsf-security-mechanisms-v1.0.pdf

[16] IETF RFC 2246: "The TLS Protocol"

[17] "Liberty ID-WSF Discovery Service Specification", Liberty Alliance Project. http://www.projectliberty.org/specs/liberty-idwsf-disco-svc-v1.0.pdf

[18] IETF RFC 2396: "Uniform Resource Identifiers (URI): Generic Syntax"

[19] IETF RFC 3261: "SIP: Session Initiation Protocol"

[20] IETF RFC 2486:" The Network Access Identifier"

[21] 3GPP TS 23.003: "Numbering, addressing and identification"

[22] IETF RFC 2821:" Simple Mail Transfer Protocol"

[23] "Liberty ID-WSF Interaction Service Specification", Liberty Alliance Project. http://www.projectliberty.org/specs/liberty-idwsf-interaction-svc-1.0-errata-v1.0.pdf

[24] Hallam-Baker, Phillip, Kaler, Chris, Monzillo, Ronald, Nadalin, Anthony, eds. (January, 2004). "Web Services Security: SOAP Message Security," OASIS Standard V1.0 [OASIS 200401], Organization for the Advancement of Structured Information Standards *http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf*

7.2 GUP headers

~~Editor´s note: This is the current approved text on this:~~

Liberty Alliance defines some headers needed when passing around messages. Since Liberty Alliance only defines a SOAP binding, these headers are defined in terms of the SOAP protocol. But this is NOT mandatory, as mentioned in [soap-binding],
"Although this binding [SOAP binding] is the only one given in this specification, other protocols could be used to convey ID-* messages, with appropriateness depending on the protocol selected and the target operational context. "
In the context of GUP, ~~we will define~~ a set of abstract headers are defined. They are needed by the messages exchanged between the various parties, against the Rp and Rg interfaces. The information contained in these headers will be described as XML data even though the binding may decide to map the information using a different syntax (e.g. ASCII, ASN.1, etc.). In the following sections, namespace "S" corresponds to the SOAP namespace.
~~We provide a n~~Normative bindings for the GUP headers to SOAP are provided in the~~(see~~ Annex E~~F) for the GUP headers~~.

7.2.1 Correlation header

7.2.1.1 Description

The **correlation** header block provides a means for correlating messages. Message correlation is achieved by using the `messageID` attribute to identify individual messages. Additionally, a message may refer to another message by setting its `refToMessageID` attribute to the value of the `messageID` of the message of interest.

7.2.1.2 Content

The content of the correlation header can be defined using the following type.

```
<xs:complexType name="correlationType">
 <xs:attribute name="messageID" type="IDType" use="required"/>
 <xs:attribute name="refToMessageID" type="IDType" use="optional"/>
 <xs:attribute name="timestamp" type="xs:dateTime" use="required"/>
 <xs:attribute name="id" type="xs:ID" use="optional"/>
 <xs:attribute name="mustUnderstand" type="xs:boolean" use="optional"/>
 <xs:attribute name="actor" type="xs:anyURI" use="optional"/>
</xs:complexType>
```

7.2.2 Provider header

7.2.2.1 Description

This header block provides a means for a sender to claim that it 545
is represented by a given providerID value. The sender may also claim that it is a member of a given affiliation. Such claims are generally verifiable by receivers by looking up these values in the sender's metadata ~~[LibertyMetadata].~~

7.2.2.2 Content

```
<xs:complexType name="ProviderType">
  <xs:attribute name="providerID" type="xs:anyURI" use="required"/>
  <xs:attribute name="affiliationID" type="xs:anyURI" use="optional"/>
  <xs:attribute name="id" type="xs:ID" use="optional"/>
  <xs:attribute name="mustUnderstand" type="xs:boolean" use="optional"/>
  <xs:attribute name="actor" type="xs:anyURI" use="optional"/>
</xs:complexType>
```

7.2.3 ProcessingContext header

7.2.3.1 Description

This header block may be employed by a sender to signal to a receiver that the latter should add a specific additional facet to the overall *processing context* in which any action(s) are invoked as a result of processing any message also conveyed in the overall message.

7.2.3.2 Content

```
<xs:complexType name="ProcessingContextType">
  <xs:simpleContent>
```

```
  <xs:extension base="xs:anyURI">
  <xs:attribute name="id" type="xs:ID" use="optional"/>
  <xs:attribute name="mustUnderstand" type="xs:boolean" use="optional"/>
  <xs:attribute name="actor" type="xs:anyURI" use="optional"/>
  </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

7.2.4 Consent header

7.2.4.1 Description

The Consent header block element MAY be employed by either a requester or a receiver. For example, the Principal may be using a Liberty-enabled client or proxy (common in the wireless world), and in that sort of environment the mobile operator may cause the Principal's terminal (AKA: cell phone) to prompt the principal for consent for some interaction.

7.2.4.2 Content

```
<xs:complexType name="consentType">
  <xs:attribute name="uri" type="xs:anyURI" use="required"/>
  <xs:attribute name="timestamp" type="xs:dateTime" use="optional"/>
  <xs:attribute name="id" type="xs:ID" use="optional"/>
  <xs:attribute name="mustUnderstand" type="xs:Boolean" use="optional"/>
  <xs:attribute name="actor" type="xs:anyURI" use="optional"/>
</xs:complexType>
```

7.2.5 UsageDirective header

7.2.5.1 Description

Participants in the ID-WSF framework may need to indicate the privacy policy associated with a message. To facilitate this, senders, acting as either a client or a server, may add one or more UsageDirective header blocks to the message being sent. A UsageDirective header appearing in a request message expresses *intended usage*. A UsageDirective header appearing in a response expresses *how* the receiver of the response is to use the response data.

7.2.5.2 Content

```
<xs:complexType name="UsageDirectiveType">
  <sequence>
    <xs:any namespace="##other" processContents="lax" maxOccurs="unbounded"/>
</sequence>
  <attribute name="ref" type="reference" use="required"/>
  <attribute name="id" type="id" use="optional"/>
  <attribute name="mustUnderstand" type="xs:boolean" use="optional"/>
  <attribute name="actor" type="xs:anyURI" use="optional"/>
</complexType>
```

7.2.6 ServiceInstanceUpdate header

7.2.6.1 Description

It may be necessary for an entity receiving a message to indicate that messages from the sender should be directed to a different endpoint, or that they wish a different credential to be used than was originally specified by the entity for access to the requested resource.
The ServiceInstanceUpdate header allows a message receiver to indicate that a new endpoint, new credentials, or new security mechanisms should be employed by the sender of the message.
The use of this header block allows the sender of the message to convey updates to security tokens, essentially providing a token renewal mechanism. This is not discussed further in this specification.

7.2.6.2 Content

```
<xs:complexType name="ServiceInstanceUpdateType">
  <xs:sequence>
    <xs:element name="SecurityMechID" type="xs:anyURI" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="Credential" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:any namespace="##any" processContents="lax"/>
        </xs:sequence>
        <xs:attribute name="notOnOrAfter" type="xs:dateTime" use="optional"/>
      </xs:complexType>
    </xs:element>
    <xs:element name="Endpoint" type="xs:anyURI" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:ID" use="optional"/>
  <xs:attribute name="mustUnderstand" type="xs:boolean" use="optional"/>
  <xs:attribute name="actor" type="xs:anyURI" use="optional"/>
</xs:complexType>
```

7.2.7 Timeout header

7.2.7.1 Description

A requesting entity may wish to indicate that they would like a request to be processed within some specified amount of time. Such an entity would indicate their wish via the Timeout header block.

7.2.7.2 Content

```
<xs:complexType name="TimeoutType">
 <xs:attribute name="maxProcessingTime" type="xs:integer" use="required"/>
 <xs:attribute name="id" type="xs:ID" use="optional"/>
 <xs:attribute name="mustUnderstand" type="xs:boolean" use="optional"/>
 <xs:attribute ref="actor" type="xs:anyURI" use="optional"/>
</xs:complexType>
```

7.2.8 CredentialsContext header

7.2.8.1 Description

~~It may be necessary for an entity receiving an ID * message to indicate the type of credentials that should be used by the requester in submitting a message.~~

The receiver of a GUP message might indicate that credentials supplied in the request did not meet its policy in allowing access to the requested resource. The `<CredentialsContext>` header block allows such policies to be expressed to a GUP requester.

7.2.8.2 Content

```
<xs:complexType name="CredentialsContextType">
  <xs:sequence>
    <xs:element ref="lib:RequestAuthnContext" minOccurs="0"/>
    <xs:element name="SecurityMechID" type="xs:anyURI" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:ID" use="optional"/>
  <xs:attribute ref="S:mustUnderstand" use="optional"/>
  <xs:attribute ref="S:actor" use="optional"/>
</xs:complexType>
```

~~We need to provide a SOAP agnostic definition of lib:RequestAuthnContext.~~

7.2.9 wsse:Security header ~~[LibertySecMech]~~

7.2.9.1 Description

Relevent security information is communicated in the Security header. The details for the specific bindings are provided in the Annexes.

7.2.9.2 Content

The details of the content for the specific binding are provided in the Annexes.
~~The content of this header is defined in WS-Security.~~

7.2.10 is:UserInteraction header ~~[LibertyInteract]~~

~~Editor's note: do we need it?~~
In some scenarios it might be necessary for GUP Servers and/or RAFs to interact with the owner of GUP component data exposed in order to e.g. ask for explicit end-user consent for the release of the requested information. To this end, GUP messages may include a `<UserInteraction>` header, which controls the possibilities that GUP Servers and/or RAFs have to engage in interactions with end-users when serving a request from a GUP Requestor.

7.2.10.1 Description

The details for the specific bindings are provided in the Annexes.

7.2.10.2 Content

The details for the specific bindings are provided in the Annexes.

Annex E (~~informative/~~normative): SOAP binding for GUP headers

~~Editor's note: have to be clarified if Annex is Informative or Normative.~~
~~Editor's note: copy/paste from Chapter 9.2 from version 0.5.0.~~ ~~Needs to be revised.~~
The SOAP protocol provides a mechanism for exchanging structured and typed information between peers using XML. It is a very generic protocol, which can also be used to carry remote procedure calls. Each SOAP message has an element "Envelope" and its immediate child elements "Header" and "Body".

SOAP is applied in the Rp and Rg reference points. SOAP carries the GUP procedure elements in its body part in compliance with the SOAP standard [5]. The GUP Procedure elements are placed immediately below the Body element. If there are several requests or responses, the GUP Procedure elements are carried one after another.

SOAP headers used in the context of GUP shall be as defined by Liberty Alliance ID-WSF SOAP Bindings [14] specification. This Liberty Alliance specification includes the definition of a number of SOAP header blocks for simple message correlation, as well as provider declaration, processing context, consent claims, usage directives and a number of other optional headers. Liberty Alliance ID-WSF SOAP Bindings [14] specification also defines how messages are bound into SOAP message bodies, and how the Liberty defined SOAP header blocks are bound into SOAP message headers.

Additionally, other SOAP header blocks defined by other Liberty Alliance specifications (i.e. Liberty Alliance ID-WSF Security Mechanisms [15] and Liberty Alliance ID-WSF Interaction Service [23]) shall be also applicable in the context of GUP and may be used concurrently with the header blocks defined in Liberty Alliance ID-WSF SOAP Bindings [14] specification.

~~The SOAP protocol is applied in the Rp reference point. SOAP provides a mechanism for exchanging structured and typed information between peers using XML. It is a very generic protocol which can also be used to carry remote procedure calls. Each SOAP message has an element "Envelope" and its immediate child elements "Header" and "Body". SOAP carries the GUP procedure elements in its body part in compliance with the SOAP standard [5]. The GUP Procedure elements are placed immediately below the Body element. If there are several requests or responses, the GUP Procedure elements are carried one after another.~~

~~GUP SOAP messages are specified to run over standard http [6] as specified in [5] but implementations may also support other transport mechanisms. If any SOAP level error is reported, no application data are returned. The used SOAP binding and error reporting mechanisms are defined in Liberty ID-WSF SOAP Binding Specification [14].~~

~~There are a number of SOAP Header elements defined for GUP. The first part of each header is defined according to the Liberty ID-WSF SOAP Binding Specification [14] which specifies the following header blocks that are also applicable in GUP:Editor's note: Whether the GUP framework will support a subset of SOAP headers defined by the SOAP specification and also by Liberty ID-WSF SOAP Binding Specification [14] is FFS. Additionally some GUP specific SOAP headers may also be required FFS. The implementation is FFS. Editor's note: Namespaces for~~

~~SOAP headers are FFS.~~

This Annex presents a brief description of the different SOAP header blocks used in the context of GUP. Other sections in this specification also refer to the use of such headers in the context of GUP.

E.1 Correlation header

SOAP does not define a mechanism to correlate one SOAP message with another message, such as in a request-response paradigm. The **correlation** header block provides a means for being able to correlate one SOAP message with another SOAP message. Message correlation is achieved by inserting a `<Correlation>` element in SOAP-bound GUP message headers and using a `messageID` attribute to identify individual messages. Additionally, a message may refer to another message by setting its `refToMessageID` attribute to the value of the `messageID` of the message of interest.

Normative semantics, definitions, schemas and processing rules for the correlation header block are defined in Liberty Alliance ID-WSF SOAP Bindings [14] specification.
~~The correlation header is implemented using the CorrelationType defined by the LA SOAP binding.~~

```
<xs:complexType name="correlationType">
 <xs:attribute name="messageID" type="IDType" use="required"/>
 <xs:attribute name="refToMessageID" type="IDType" use="optional"/>
 <xs:attribute name="timestamp" type="xs:dateTime" use="required"/>
 <xs:attribute name="id" type="xs:ID" use="optional"/>
 <xs:attribute ref="S:mustUnderstand" use="optional"/>
 <xs:attribute ref="S:actor" use="optional"/>
</xs:complexType>
```

E.2 Provider header

A Provider in the context of GUP is an entity that is able to handle (issue and/or receive) SOAP messages, and which is univocally identified at the SOAP level by its providerID. This header block provides means for a sender to claim that it is represented by a given providerID value.

Normative semantics, definitions, schemas and processing rules for the provider header block are defined in Liberty Alliance ID-WSF SOAP Bindings [14] specification.
~~The provider header is implemented using the ProviderType defined by the LA SOAP binding.~~

```
<xs:complexType name="ProviderType">
  <xs:attribute name="providerID" type="xs:anyURI" use="required"/>
  <xs:attribute name="affiliationID" type="xs:anyURI" use="optional"/>
  <xs:attribute name="id" type="xs:ID" use="optional"/>
  <xs:attribute ref="S:mustUnderstand" use="optional"/>
  <xs:attribute ref="S:actor" use="optional"/>
</xs:complexType>
```

E.3 ProcessingContext header

A GUP requestor may need to express additional context for a given request, for example indicating that the requester expects to make such requests in the future when the end-user may or may not be online. This header block may be employed by a sender to signal to a receiver that the latter should add a specific additional facet to the overall *processing context* in which any action(s) are invoked as a result of processing any message also conveyed in the overall message.

Normative semantics, definitions, schemas and processing rules for the processing context header block are defined in Liberty Alliance ID-WSF SOAP Bindings [14] specification.
~~The ProcessingContext header is implemented using the ProcessingContextType defined by the LA SOAP binding.~~

```
<xs:complexType name="ProcessingContextType">
  <xs:simpleContent>
  <xs:extension base="xs:anyURI">
  <xs:attribute name="id" type="xs:ID" use="optional"/>
  <xs:attribute ref="S:mustUnderstand" use="optional"/>
  <xs:attribute ref="S:actor" use="optional"/>
  </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

E.4 Consent header

GUP applications may wish to claim whether they obtained the end-user's consent for carrying out any given operation. This header block is used to explicitly claim that the Principal consented to the present interaction.

Normative semantics, definitions, schemas and processing rules for the consent header block are defined in Liberty Alliance ID-WSF SOAP Bindings [14] specification.
~~The consent header is implemented using the consentType defined by the LA SOAP binding.~~

```
<xs:complexType name="consentType">
  <xs:attribute name="uri" type="xs:anyURI" use="required"/>
  <xs:attribute name="timestamp" type="xs:dateTime" use="optional"/>
  <xs:attribute name="id" type="xs:ID" use="optional"/>
  <xs:attribute ref="S:mustUnderstand" use="optional"/>
  <xs:attribute ref="S:actor" use="optional"/>
</xs:complexType>
```

E.5 UsageDirective header

GUP requestors may wish to indicate their policies for handling data at the time of data request, while GUP server and RAFs may wish to specify their policies for the subsequent use of data at the time of data release. To facilitate this, GUP requestors may add one or more UsageDirective header blocks to the message being sent. A UsageDirective header appearing in a request message expresses *intended usage*. A UsageDirective header appearing in a response expresses *how* the receiver of the response is to use the response data.

Normative semantics, definitions, schemas and processing rules for the usage directive header block are defined in Liberty Alliance ID-WSF SOAP Bindings [14] specification.
~~The UsageDirective header is implemented using the UsageDirectiveType defined by the LA SOAP binding.~~

```
<complexType name="UsageDirectiveType">
  <sequence>
    <xs:any namespace="##other" processContents="lax" maxOccurs="unbounded"/>
  </sequence>
  <attribute name="ref" type="reference" use="required"/>
  <attribute name="id" type="id" use="optional"/>
```

```
  <attribute ref="S:mustUnderstand" use="optional"/>
  <attribute ref="S:actor" use="optional"/>
</complexType>
```

E.6 ServiceInstanceUpdate header

It may be necessary for an entity receiving a message to indicate that messages from the sender should be directed to a different endpoint, or that they wish a different credential to be used than was originally specified by the entity for access to the requested resource.

The ServiceInstanceUpdate header allows a message receiver to indicate that a new endpoint, new credentials, or new security mechanisms should be employed by the sender of the message.
The use of this header block allows the sender of the message to convey updates to security tokens, essentially providing a token renewal mechanism. This is not discussed further in this specification.

Normative semantics, definitions, schemas and processing rules for the service instance update header block are defined in Liberty Alliance ID-WSF SOAP Bindings [14] specification.
The ServiceInstance header is implemented using the ServiceInstanceType defined by the LA SOAP binding.

```
<xs:complexType name="ServiceInstanceUpdateType">
  <xs:sequence>
    <xs:element name="SecurityMechID" type="xs:anyURI" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="Credential" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:any namespace="##any" processContents="lax"/>
        </xs:sequence>
        <xs:attribute name="notOnOrAfter" type="xs:dateTime" use="optional"/>
      </xs:complexType>
    </xs:element>
    <xs:element name="Endpoint" type="xs:anyURI" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:ID" use="optional"/>
  <xs:attribute ref="S:mustUnderstand" use="optional"/>
  <xs:attribute ref="S:actor" use="optional"/>
</xs:complexType>
```

E.7 Timeout header

A requesting entity may wish to indicate that they would like a request to be processed within some specified amount of time. Such an entity would indicate their wish via the Timeout header block.

Normative semantics, definitions, schemas and processing rules for the timeout header block are defined in Liberty Alliance ID-WSF SOAP Bindings [14] specification.
The Timeout header is implemented using the TimeoutType defined by the LA SOAP binding.

```
<xs:complexType name="TimeoutType">
 <xs:attribute name="maxProcessingTime" type="xs:integer" use="required"/>
 <xs:attribute name="id" type="xs:ID" use="optional"/>
 <xs:attribute ref="S:mustUnderstand" use="optional"/>
 <xs:attribute ref="S:actor" use="optional"/>
</xs:complexType>
```

E.8 CredentialsContext header

The receiver of a GUP message might indicate that credentials supplied in the request did not meet its policy in allowing access to the requested resource. The <CredentialsContext> header block allows such policies to be expressed to a GUP requester.

Normative semantics, definitions, schemas and processing rules for the credentials context header block are defined in Liberty Alliance ID-WSF SOAP Bindings [14] specification.
~~The CredentialsContext header is implemented using the CredentialsContextType defined by the LA SOAP binding.~~

```
~~<xs:complexType name="CredentialsContextType">~~
~~  <xs:sequence>~~
~~    <xs:element ref="lib:RequestAuthnContext" minOccurs="0"/>~~
~~    <xs:element name="SecurityMechID" type="xs:anyURI" minOccurs="0"~~
~~maxOccurs="unbounded"/>~~
~~  </xs:sequence>~~
~~  <xs:attribute name="id" type="xs:ID" use="optional"/>~~
~~  <xs:attribute ref="S:mustUnderstand" use="optional"/>~~
~~  <xs:attribute ref="S:actor" use="optional"/>~~
~~</xs:complexType>~~
```

E.9 ~~wsse:~~Security header ~~[LibertySecMech]~~

OASIS WS-Security compliant header elements as defined in OASIS SOAP Message Security [24] specification communicate the relevant security information in SOAP.

The construction and decoration of the <wsse:Security> header element in the context of GUP MUST adhere to the rules specified in OASIS SOAP Message Security [24] specification.

Additionally the construction, decoration and usage of the <wsse:Security> header element in the context of GUP MUST also adhere to the mechanisms defined in Liberty Alliance Security Mechanisms [15] specification.

Normative semantics, definitions, schemas and processing rules for the security header block are defined both in OASIS SOAP Message Security [24] and Liberty Alliance ID-WSF Security Mechanisms [15] specifications.

E.10 ~~is:~~UserInteraction header ~~[LibertyInteract]~~

In some scenarios it might be necessary for GUP Servers and/or RAFs to interact with the owner of GUP component data exposed in order to e.g. ask for explicit end-user consent for the release of the requested information. To this end, GUP messages may include a <UserInteraction> SOAP header, which controls the possibilities that GUP Servers and/or RAFs have to engage in interactions with end-users when serving a request from a GUP Requestor.

Normative semantics, definitions, schemas and processing rules for the user interaction header block are defined in Liberty Alliance ID-WSF Interaction Service [23] specification.

~~E.11 Example~~

~~We now provide a simple example of the usage of the GUP headers in the context of the GUP SOAP binding.~~
~~Editor's note: copy paste from the LA soap binding spec and make it GUP compliant.~~