3GPP TSG CN Plenary Meeting #20 NP-030237
04-06 June 2003. Hämeenlinna, FINLAND

Source: **CN5 (OSA)**

Title: **Rel-4 CR 29.198-03 OSA API Part 3: Framework**

Agenda item: **7.10**

Document for: **APPROVAL**

Doc-1st-Level	Spec	CR	R	Ph	Subject	Cat	Ver-Curr	Doc-2nd-Level	WI
NP-030237	29.198-03	078	-	Rel-4	Correction to TpEncryptionCapability to correct support for Triple-DES	F	4.7.0	N5-030193	OSA1
NP-030237	29.198-03	079	-	Rel-5	Correction to TpEncryptionCapability to correct support for Triple-DES	A	5.2.0	N5-030194	OSA1
NP-030237	29.198-03	080	-	Rel-4	Correction of the Framework Service Instance Lifecycle Manager Sequence Diagram	F	4.7.0	N5-030280	OSA1
NP-030237	29.198-03	081	-	Rel-5	Correction of the Framework Service Instance Lifecycle Manager Sequence Diagram	A	5.2.0	N5-030281	OSA1
NP-030237	29.198-03	082	-	Rel-4	Correction of the use of TpDomainID in Framework initiateAuthentication method	F	4.7.0	N5-030282	OSA1
NP-030237	29.198-03	083	-	Rel-5	Correction of the use of TpDomainID in Framework initiateAuthentication method	A	5.2.0	N5-030283	OSA1

joint-API-group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) **N5-030193**
Meeting #23, San Diego, CA, USA, 19 - 23 May 2003

CR-Form-v7

CHANGE REQUEST

⌘	**29.198-03**	**CR**	**078**	⌘ **rev**	**-**	⌘	Current version:	**4.7.0**	⌘

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:	⌘	Correction to TpEncryptionCapability to correct support for Triple-DES		
Source:	⌘	Ultan Mulligan, ETSI PTCC		
Work item code:	⌘	OSA1	***Date:*** ⌘	2/05/2003
Category:	⌘	**F**	***Release:*** ⌘	*REL-4*

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:

2	*(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)*

Reason for change:	⌘	TpEncryptionCapability contains a value P_DES_128, which is described as: "A simple transfer of secret information that is shared between the client entity and the Framework with protection against interception on the link provided by the DES algorithm with a 128-bit shared secret key." DES algorithm is designed to take a 56-bit key. There is no variant of DES which can take a 128-bit key. It is unclear what behaviour is expected of implementations which select the P_DES_128 value of TpEncryptionCapability. It may be that this value was intended to identify Triple-DES, or TDEA, i.e. three DES operations in series (encrypt, decrypt, encrypt) with up to 3 unique 56-bit keys. DES can have 4 modes of operation, TDEA or Triple-DES can have seven. In each case, we should specify the intended mode of operation.
Summary of change:	⌘	Deprecate the use of P_DES_128, as it is meaningless. Add P_TDEA to the list of encryption capabilities, to permit use of Triple-DES algorithm. Specify the mode of operation of DES and Triple-DES to be the most straight-forward one: ECB for DES, and TECB for Triple-DES.
Consequences if not approved:	⌘	Our continued reliance on P_DES_128 will provoke incomprehension, if not amusement, among developers. Interworking problems may result if the modes of operation are not clarified.

Clauses affected:	⌘	10.3.3				
		Y	**N**			
Other specs affected:	⌘		**X**	Other core specifications	⌘	
			X	Test specifications		
			X	O&M Specifications		

Other comments:	⌘	Rel-5 mirror CR in N5-030194.

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.

10.3.3 TpEncryptionCapability

This data type is identical to a TpString, and is defined as a string of characters that identify the encryption capabilities that could be supported by the framework. Other Network operator specific capabilities may also be used, but should be preceded by the string "SP_". Capabilities may be concatenated, using commas (,) as the separation character. The following values are defined.

String Value	Description
NULL	An empty (NULL) string indicates no client capabilities.
P_DES_56	A simple transfer of secret information that is shared between the client application and the Framework with protection against interception on the link provided by the DES algorithm with a 56-bit shared secret key. The ECB mode of DES is to be used.
P_DES_128	A simple transfer of secret information that is shared between the client entity and the Framework with protection against interception on the link provided by the DES algorithm with a 128-bit shared secret key. Use of the P_DES_128 value of TpEncryptionCapability is deprecated, as DES cannot be used with a 128-bit key.
P_RSA_512	A public-key cryptography system providing authentication without prior exchange of secrets using 512-bit keys.
P_RSA_1024	A public-key cryptography system providing authentication without prior exchange of secrets using 1024-bit keys.
P_TDEA	The Triple-DES or TDEA algorithm with three 56-bit secret keys. The key exchange is handled seperately, and may permit use of three, two or only one unique key. The TECB mode of Triple-DES is to be used.

joint-API-group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) **N5-030194**
Meeting #23, San Diego, CA, USA, 19 - 23 May 2003

CR-Form-v7

CHANGE REQUEST

⌘ **29.198-03** **CR** **079** ⌘ **rev** **-** ⌘ Current version: **5.2.0** ⌘

*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:	⌘	Correction to TpEncryptionCapability to correct support for Triple-DES
Source:	⌘	Ultan Mulligan, ETSI PTCC
Work item code:	⌘	OSA1
Date:	⌘	2/05/2003
Category:	⌘	**A**
Release:	⌘	*REL-5*

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:

2	*(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)*

Reason for change: ⌘

TpEncryptionCapability contains a value P_DES_128, which is described as: "A simple transfer of secret information that is shared between the client entity and the Framework with protection against interception on the link provided by the DES algorithm with a 128-bit shared secret key."

DES algorithm is designed to take a 56-bit key. There is no variant of DES which can take a 128-bit key. It is unclear what behaviour is expected of implementations which select the P_DES_128 value of TpEncryptionCapability.

It may be that this value was intended to identify Triple-DES, or TDEA, i.e. three DES operations in series (encrypt, decrypt, encrypt) with up to 3 unique 56-bit keys.

DES can have 4 modes of operation, TDEA or Triple-DES can have seven. In each case, we should specify the intended mode of operation.

Summary of change: ⌘

Deprecate the use of P_DES_128, as it is meaningless.

Add P_TDEA to the list of encryption capabilities, to permit use of Triple-DES algorithm.

Specify the mode of operation of DES and Triple-DES to be the most straight-forward one: ECB for DES, and TECB for Triple-DES.

Consequences if not approved: ⌘

Our continued reliance on P_DES_128 will provoke incomprehension, if not amusement, among developers. Interworking problems may result if the modes of operation are not clarified.

Clauses affected: ⌘ 10.3.3

Other specs affected: ⌘

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

Other comments:	⌘	Rel-5 mirror of Rel-4 CR in N5-030193.

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.

10.3.3 TpEncryptionCapability

This data type is identical to a TpString, and is defined as a string of characters that identify the encryption capabilities that could be supported by the framework. Other Network operator specific capabilities may also be used, but should be preceded by the string "SP_". Capabilities may be concatenated, using commas (,) as the separation character. The following values are defined.

String Value	Description
NULL	An empty (NULL) string indicates no client capabilities.
P_DES_56	A simple transfer of secret information that is shared between the client application and the Framework with protection against interception on the link provided by the DES algorithm with a 56-bit shared secret key. The ECB mode of DES is to be used.
P_DES_128	A simple transfer of secret information that is shared between the client entity and the Framework with protection against interception on the link provided by the DES algorithm with a 128-bit shared secret key. Use of the P_DES_128 value of TpEncryptionCapability is deprecated, as DES cannot be used with a 128-bit key.
P_RSA_512	A public-key cryptography system providing authentication without prior exchange of secrets using 512-bit keys.
P_RSA_1024	A public-key cryptography system providing authentication without prior exchange of secrets using 1024-bit keys.
P_TDEA	The Triple-DES or TDEA algorithm with three 56-bit secret keys. The key exchange is handled seperately, and may permit use of three, two or only one unique key. The TECB mode of Triple-DES is to be used.

joint-API-group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) **N5-030280**
Meeting #23, San Diego, USA, 19 – 22 May 2003

CR-Form-v7

CHANGE REQUEST

⌘	**29.198-03**	**CR**	**080**	⌘ **rev**	**-**	⌘	Current version:	**4.7.0** ⌘

*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: ⌘	Correction of the Framework Service Instance Lifecycle Manager Sequence Diagram		
Source: ⌘	AePONA – Eamonn Murray		
Work item code: ⌘	OSA1	***Date:*** ⌘	06/05/2003
Category: ⌘	**F**	***Release:*** ⌘	*REL-4*

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:

2	*(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)*

Reason for change: ⌘	The current Framework Service Instance Lifecycle Manager sequence diagrams do not include any details regarding the establishment of Framework – SCS Access sessions. Such access sessions are the cornerstone of all Framework - SCS Management functionality, and the absence of clear information on the functionality required from SCS implementations, may result in significant interoperability problems or an inability to support key functionality.
Summary of change: ⌘	Introduce additional clarification to the Framework Service Instance Lifecycle Manager sequence diagram to indicate the functionality required from SCS implementations.
Consequences if not approved: ⌘	Multi Vendor Interoperability cannot be supported, or key Framework authentication and management capability cannot be guaranteed.

Clauses affected: ⌘ 8.1.3

	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.

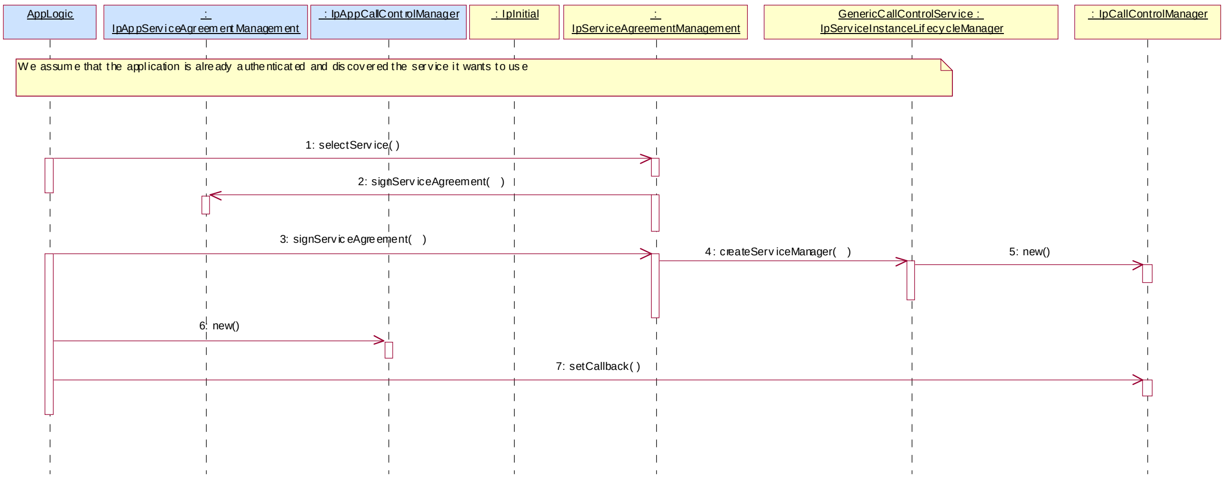
3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

*********** Start of Change # 1 ******************

8.1.3 Service Instance Lifecycle Manager Sequence Diagrams

8.1.3.1 Sign Service Agreement

This sequence illustrates how the application can get access to a specified service. It only illustrates the last part: the signing of the service agreement and the corresponding actions towards the service. For more information on accessing the framework, authentication and discovery of services, see the corresponding clauses.


AppLogic
: IpAppServiceAgreementManagement
: IpAppCallControlManager
: IpInitial
: IpServiceAgreementManagement
GenericCallControlService : IpServiceInstanceLifecycleManager
: IpCallControlManager
We assume that the application is already authenticated and discovered the service it wants to use
1: selectService( )
2: signServiceAgreement( )
3: signServiceAgreement( )
4: createServiceManager( )
5: new()
6: new()
7: setCallback( )

1: The application selects the service, using a serviceID for the generic call control service. The serviceID could have been obtained via the discovery interface. A ServiceToken is returned to the application.

2: The framework signs the service agreement.

3: The client application signs the service agreement. As a result a service manager interface reference (in this case of type IpCallControlManager) is returned to the application.

4: Provided the signature information is correct and all conditions have been fulfilled, the framework will request the service identified by the serviceID to return a service manager interface reference. The service manager is the initial point of contact to the service.

5: The lifecycle manager creates a new manager interface instance (a call control manager) for the specified application. It should be noted that this is an implementation detail. The service implementation may use other mechanism to get a service manager interface instance.

Following the creation of the service manager outlined above, a unique instance of the service particular to the application client results. This service instance is assigned a serviceInstanceID by the Framework, which is provided to the Service Instance Lifecycle manager during the createServiceManager operation. ~~It is then~~If it is necessary that Framework Integrity Management functionality and operations are to be supported between the Framework and the service instance identified by the defined serviceInstanceID, it is then necessary for the new service instance to establish an access session with the Framework. ~~This step is mandatory in order to ensure that Framework Integrity Management functionality and operations can be supported between the Framework and the service instance identified by the defined serviceInstanceID~~. This provides theFramework with the ability to manage and monitor the operation of the service instance that relates to a particular application client. The steps required to establish a Framework access session are outlined in chapter 6 of this specification.

6: The application creates a new IpAppCallControlManager interface to be used for callbacks.

7: The Application sets the callback interface to the interface created with the previous message.

************** End of Change # 1 ************************

joint-API-group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) N5-030281
Meeting #23, San Diego, USA, 19 – 22 May 2003

CR-Form-v7

CHANGE REQUEST

⌘	**29.198-03**	**CR**	**081**	⌘**rev**	**-**	⌘	Current version:	**5.2.0**	⌘

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:	⌘	Correction of the Framework Service Instance Lifecycle Manager Sequence Diagram		
Source:	⌘	AePONA – Eamonn Murray		
Work item code:	⌘	OSA1	***Date:*** ⌘	06/05/2003
Category:	⌘	**A**	***Release:*** ⌘	*REL-5*

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:

2	*(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)*

Reason for change:	⌘	The current Framework Service Instance Lifecycle Manager sequence diagrams do not include any details regarding the establishment of Framework – SCS Access sessions. Such access sessions are the cornerstone of all Framework - SCS Management functionality, and the absence of clear information on the functionality required from SCS implementations, may result in significant interoperability problems or an inability to support key functionality.
Summary of change:	⌘	Introduce additional clarification to the Framework Service Instance Lifecycle Manager sequence diagram to indicate the functionality required from SCS implementations.
Consequences if not approved:	⌘	Multi Vendor Interoperability cannot be supported, or key Framework authentication and management capability cannot be guaranteed.

Clauses affected:	⌘	8.1.3				
		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.

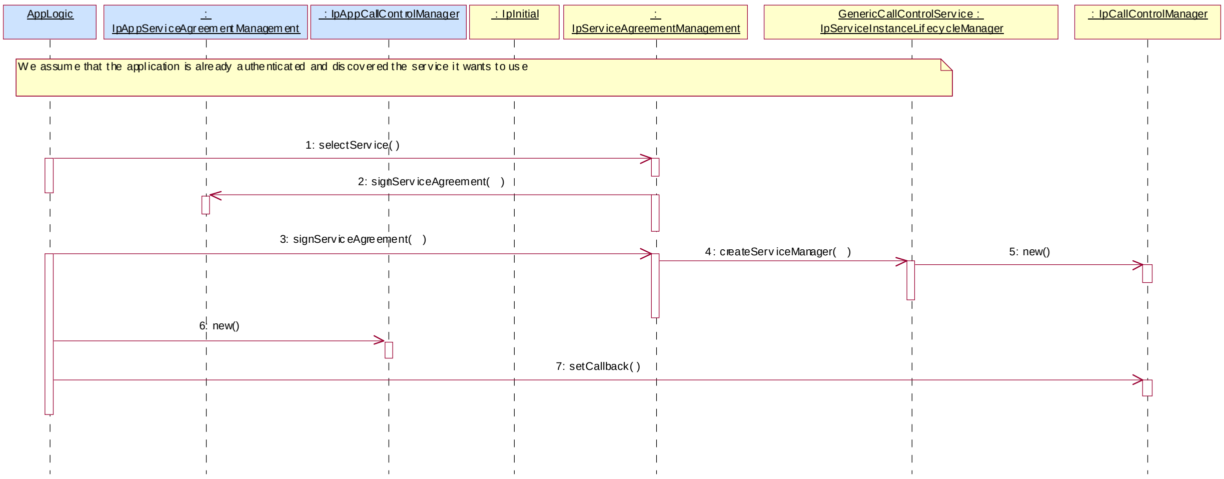
3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

*********** Start of Change # 1 ******************

8.1.3 Service Instance Lifecycle Manager Sequence Diagrams

8.1.3.1 Sign Service Agreement

This sequence illustrates how the application can get access to a specified service. It only illustrates the last part: the signing of the service agreement and the corresponding actions towards the service. For more information on accessing the framework, authentication and discovery of services, see the corresponding clauses.


AppLogic
: IpAppServiceAgreementManagement
: IpAppCallControlManager
: IpInitial
: IpServiceAgreementManagement
GenericCallControlService : IpServiceInstanceLifecycleManager
: IpCallControlManager
We assume that the application is already authenticated and discovered the service it wants to use
1: selectService( )
2: signServiceAgreement( )
3: signServiceAgreement( )
4: createServiceManager( )
5: new()
6: new()
7: setCallback( )

1: The application selects the service, using a serviceID for the generic call control service. The serviceID could have been obtained via the discovery interface. A ServiceToken is returned to the application.

2: The framework signs the service agreement.

3: The client application signs the service agreement. As a result a service manager interface reference (in this case of type IpCallControlManager) is returned to the application.

4: Provided the signature information is correct and all conditions have been fulfilled, the framework will request the service identified by the serviceID to return a service manager interface reference. The service manager is the initial point of contact to the service.

5: The lifecycle manager creates a new manager interface instance (a call control manager) for the specified application. It should be noted that this is an implementation detail. The service implementation may use other mechanism to get a service manager interface instance.

Following the creation of the service manager outlined above, a unique instance of the service particular to the application client results. This service instance is assigned a serviceInstanceID by the Framework, which is provided to the Service Instance Lifecycle manager during the createServiceManager operation. ~~It is then~~If it is necessary that Framework Integrity Management functionality and ~~ ~~operations are to be supported between the Framework and the service instance identified by the defined serviceInstanceID, it is then necessary for the new service instance to establish an access session with the Framework. ~~This step is mandatory in order to ensure that Framework Integrity Management functionality and operations can be supported between the Framework and the service instance identified by the defined serviceInstanceID~~. This provides theFramework with the ability to manage and monitor the operation of the service instance that relates to a particular application client. The steps required to establish a Framework access session are outlined in chapter 6 of this specification.

6: The application creates a new IpAppCallControlManager interface to be used for callbacks.

7: The Application sets the callback interface to the interface created with the previous message.

************** End of Change # 1 ************************

joint-API-group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) **N5-030282**
Meeting #23, San Diego, USA, 19 – 22 May 2003

CR-Form-v7

CHANGE REQUEST

⌘ **29.198-03 CR 082** ⌘ **rev - ⌘** Current version: **4.7.0** ⌘

*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:	⌘	Correction of the use of TpDomainID in Framework initiateAuthentication method
Source:	⌘	AePONA – Eamonn Murray
Work item code:	⌘	OSA1
Date:	⌘	06/05/2003
Category:	⌘	**F**
Release:	⌘	*REL-4*

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:
2 (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)

Reason for change:	⌘	Correct the desciptive text in the Framework initiateAuthentication method that relates to the TpDomainID of the entity authenticating with the Framework. In the definition of TpDomainID it is not possible to define a TpServiceID element. The current description of the initiateAuthentication method incorrectly makes reference to the TpServiceID element and fails to detail TpServiceInstanceID as a valid value for the TpDomainID.
Summary of change:	⌘	Correct the description of initiateAuthentication
Consequences if not approved:	⌘	Significant Interoperability issues are likely to result from varying vendor interpretation of the incorrect description that currently exists.

Clauses affected:	⌘	6.3.1.3			
		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.

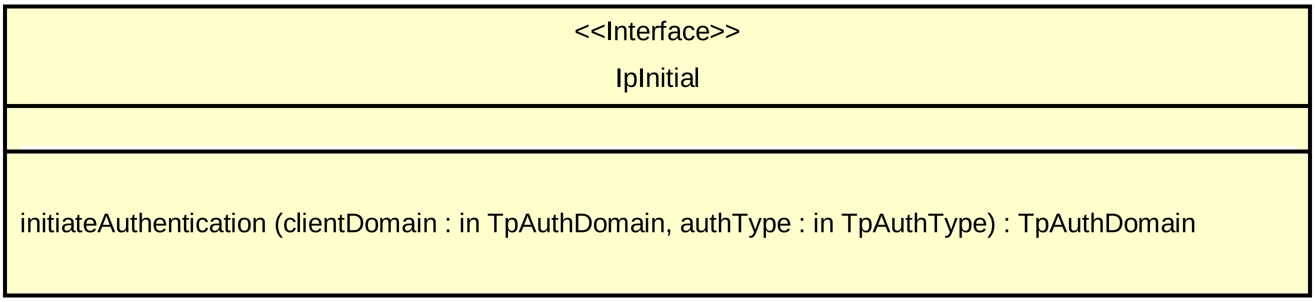
3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

*********** Start of Change # 1 ******************

6.3.1.3 Interface Class IpInitial

Inherits from: IpInterface.

The Initial Framework interface is used by the client to initiate the mutual authentication with the Framework. This interface and the initiateAuthentication() method shall be implemented by a ~~client~~Framework.



<<Interface>>
IpInitial
initiateAuthentication (clientDomain : in TpAuthDomain, authType : in TpAuthType) : TpAuthDomain

Method
initiateAuthentication()

This method is invoked by the client to start the process of mutual authentication with the framework, and request the use of a specific authentication method.

Returns <fwDomain> : This provides the client with a framework identifier, and a reference to call the authentication interface of the framework.

```
structure TpAuthDomain {
    domainID:        TpDomainID;
    authInterface:   IpInterfaceRef;
    };
```

The domainID parameter is an identifier for the framework (i.e. TpFwID). It is used to identify the framework to the client.

The authInterface parameter is a reference to the authentication interface of the framework. The type of this interface is defined by the authType parameter. The client uses this interface to authenticate with the framework.

Parameters

clientDomain : in TpAuthDomain

This identifies the client domain to the framework, and provides a reference to the domain's authentication interface.

```
structure TpAuthDomain {
    domainID:        TpDomainID;
    authInterface:   IpInterfaceRef;
};
```

The domainID parameter is an identifier either for a client application (i.e. TpClientAppID) or for an enterprise operator (i.e. TpEntOpID),~~ or for an existing registered service (i.e. TpServiceID),~~ or for an instance of a service for which a client application has signed a service agreement (i.e. TpServiceInstanceID), or for a service supplier (i.e. TpServiceSupplierID). It is used to identify the client domain to the framework, (see authenticate() on IpAPILevelAuthentication). If the framework does not recognise the domainID, the framework returns an error code (P_INVALID_DOMAIN_ID).

The authInterface parameter is a reference to call the authentication interface of the client. The type of this interface is defined by the authType parameter. If the interface reference is not of the correct type, the framework returns an error code (P_INVALID_INTERFACE_TYPE).

authType : in TpAuthType

This identifies the type of authentication mechanism requested by the client. It provides operators and clients with the opportunity to use an alternative to the API level Authentication interface, e.g. an implementation specific authentication mechanism like CORBA Security, using the IpAuthentication interface, or Operator specific Authentication interfaces. OSA API level Authentication is the default authentication mechanism (P_OSA_AUTHENTICATION). If P_OSA_AUTHENTICATION is selected, then the clientDomain and fwDomain authInterface parameters are references to interfaces of type Ip(Client)APILevelAuthentication. If P_AUTHENTICATION is selected, the fwDomain authInterface parameter references to interfaces of type IpAuthentication which is used when an underlying distribution technology authentication mechanism is used.

Returns

TpAuthDomain

Raises

TpCommonExceptions, P_INVALID_DOMAIN_ID, P_INVALID_INTERFACE_TYPE, P_INVALID_AUTH_TYPE

************** End of Change # 1 ************************

joint-API-group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) **N5-030283**
Meeting #23, San Diego, USA, 19 – 22 May 2003

CR-Form-v7

CHANGE REQUEST

⌘ **29.198-03** **CR** **083** ⌘**rev** **-** ⌘ Current version: **5.2.0** ⌘

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:	⌘	Correction of the use of TpDomainID in Framework initiateAuthentication method		
Source:	⌘	AePONA – Eamonn Murray		
Work item code:	⌘	OSA1	***Date:*** ⌘	06/05/2003
Category:	⌘	**A**	***Release:*** ⌘	*REL-5*

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:

2	*(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)*

Reason for change:	⌘	Correct the desciptive text in the Framework initiateAuthentication method that relates to the TpDomainID of the entity authenticating with the Framework. In the definition of TpDomainID it is not possible to define a TpServiceID element. The current description of the initiateAuthentication method incorrectly makes reference to the TpServiceID element and fails to detail TpServiceInstanceID as a valid value for the TpDomainID.
Summary of change:	⌘	Correct the description of initiateAuthentication
Consequences if not approved:	⌘	Significant Interoperability issues are likely to result from varying vendor interpretation of the incorrect description that currently exists.

Clauses affected:	⌘	6.3.1.3				
		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.

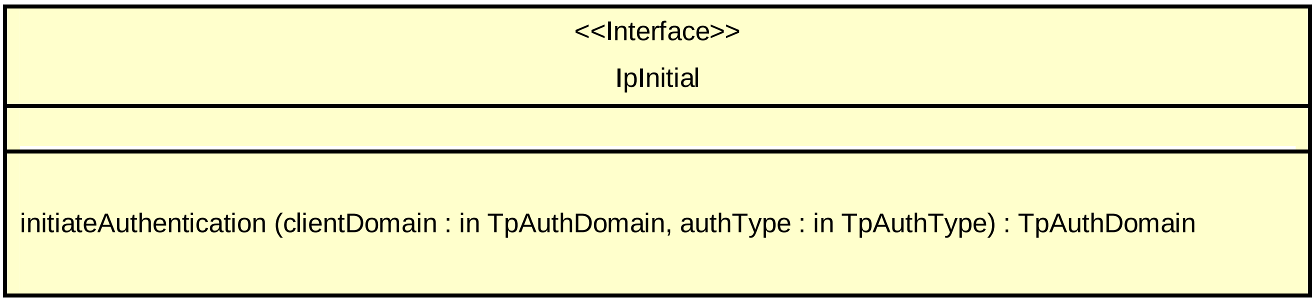
3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

*********** Start of Change # 1 ******************

6.3.1.3 Interface Class IpInitial

Inherits from: IpInterface.

The Initial Framework interface is used by the client to initiate the mutual authentication with the Framework. This interface and the initiateAuthentication() method shall be implemented by a ~~client~~Framework.


<<Interface>>
IpInitial
initiateAuthentication (clientDomain : in TpAuthDomain, authType : in TpAuthType) : TpAuthDomain

Method

initiateAuthentication()

This method is invoked by the client to start the process of mutual authentication with the framework, and request the use of a specific authentication method.

Returns <fwDomain> : This provides the client with a framework identifier, and a reference to call the authentication interface of the framework.

```
structure TpAuthDomain {
    domainID:        TpDomainID;
    authInterface:   IpInterfaceRef;
    };
```

The domainID parameter is an identifier for the framework (i.e. TpFwID). It is used to identify the framework to the client.

The authInterface parameter is a reference to the authentication interface of the framework. The type of this interface is defined by the authType parameter. The client uses this interface to authenticate with the framework.

Parameters

clientDomain : in TpAuthDomain

This identifies the client domain to the framework, and provides a reference to the domain's authentication interface.

```
structure TpAuthDomain {
    domainID:        TpDomainID;
    authInterface:   IpInterfaceRef;
};
```

The domainID parameter is an identifier either for a client application (i.e. TpClientAppID) or for an enterprise operator (i.e. TpEntOpID),~~ or for an existing registered service (i.e. TpServiceID),~~ or for an instance of a service for which a client application has signed a service agreement (i.e. TpServiceInstanceID), or for a service supplier (i.e. TpServiceSupplierID). It is used to identify the client domain to the framework, (see authenticate() on IpAPILevelAuthentication). If the framework does not recognise the domainID, the framework returns an error code (P_INVALID_DOMAIN_ID).

The authInterface parameter is a reference to call the authentication interface of the client. The type of this interface is defined by the authType parameter. If the interface reference is not of the correct type, the framework returns an error code (P_INVALID_INTERFACE_TYPE).

authType : in TpAuthType

This identifies the type of authentication mechanism requested by the client. It provides operators and clients with the opportunity to use an alternative to the API level Authentication interface, e.g. an implementation specific authentication mechanism like CORBA Security, using the IpAuthentication interface, or Operator specific Authentication interfaces. OSA API level Authentication is the default authentication mechanism (P_OSA_AUTHENTICATION). If P_OSA_AUTHENTICATION is selected, then the clientDomain and fwDomain authInterface parameters are references to interfaces of type Ip(Client)APILevelAuthentication. If P_AUTHENTICATION is selected, the fwDomain authInterface parameter references to interfaces of type IpAuthentication which is used when an underlying distribution technology authentication mechanism is used.

Returns

TpAuthDomain

Raises

TpCommonExceptions, P_INVALID_DOMAIN_ID, P_INVALID_INTERFACE_TYPE, P_INVALID_AUTH_TYPE

************** End of Change # 1 ************************