Draft EN 301 143 V1.1.1 (1998-01)

European Standard (Telecommunications series)

Cordless Terminal Mobility (CTM); Stage 3 specifications for Service Control Function (SCF) - SCF and Call Unrelated Service Function (CUSF) / Service Switching Function (SSF) - SCF interface

European Telecommunications Standards Institute

Reference
DEN/SPS-03061 (al000ico.PDF)

Keywords
CCS, CS2, CTM, INAP, ISDN, protocol, SS7

ETSI Secretariat

Postal address
F-06921 Sophia Antipolis Cedex - FRANCE

Office address
650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16
Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

X.400
c= fr; a=atlas; p=etsi; s=secretariat

Internet
secretariat@etsi.fr
http://www.etsi.fr

Contents

Intellectual Property Rights

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

Pursuant to the ETSI Interim IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETR 314 (or the updates on http://www.etsi.fr/ipr) which are, or may be, or may become, essential to the present document.

Foreword

This European Standard (Telecommunications series) has been produced by ETSI Technical Committee Signalling Protocols and Switching (SPS), and is now submitted for the Public Enquiry phase of the ETSI standards Two-step Approval Procedure (TAP).

Proposed national transposition dates	
Date of latest announcement of this EN (doa):	3 months after ETSI publication
Date of latest publication of new National Standard or endorsement of this EN (dop/e):	6 months after doa
Date of withdrawal of any conflicting National Standard (dow):	6 months after doa

Introduction

The Cordless Terminal Mobility (CTM) service phase 1 allows subscribers of cordless terminals to be mobile within and between networks. Where radio coverage is provided and the cordless terminal has appropriate access rights the subscriber will be able to make calls from, and to receive calls at, any location within the fixed public and/or private networks.

The signalling procedures provided in the present document are supporting features required for CTM phase 1 on the SCF-SCF, SSF-SCF, and CUSF-SCF interfaces. The service to service capability extensions to the core Intelligent Network Application Part (INAP) CS2 SCF-SCF interface provided in the present document are required to support the features required for CTM phase 1.

1 Scope

The present document analyses what the Cordless Terminal Mobility (CTM) specific specifications are within the generic parameters on the Service Control Function (SCF)-SCF, Call Unrelated Service Function (CUSF)-SCF and Service Switching Function (SSF)-SCF interfaces of the ETSI core Intelligent Network Application Part (INAP). These specifications ensure internetworking between CTM networks and the establishment of a multi-vendor environment for CTM.

The generic INAP protocol contains generic parameters that can convey application specific information, such as for the CTM application. Detailed CTM specifications are described by the present document using the generic INAP SCF-SCF, CUSF-SCF and SSF-SCF interface capabilities. The SCF-Service Data Point (SDF) and SDF-SDF interfaces are not considered.

2 References

References may be made to:

a) specific versions of publications (identified by date of publication, edition number, version number, etc.), in which case, subsequent revisions to the referenced document do not apply; or

b) all versions up to and including the identified version (identified by "up to and including" before the version identity); or

c) all versions subsequent to and including the identified version (identified by "onwards" following the version identity); or

d) publications without mention of a specific version, in which case the latest version applies.

A non-specific reference to an ETS shall also be taken to refer to later versions published as an EN with the same number.

2.1 Normative references

[1] DEN/NA-020039 (1997): "Network Aspects (NA), Cordless Terminal Mobility (CTM), Service Description Phase 1".

NOTE: Document not available at the time of release of present document for PE.

[2] EN 301 140-1 (V1.1): "Intelligent Network (IN); Intelligent Network Capability Set 2 (CS2); Intelligent Network Application Protocol (INAP); Part 1: Protocol specification".

[3] EN 301 144-1: "Integrated Services Digital Network (ISDN); Digital Subscriber Signalling System No. one (DSS1) protocol and Signalling System No.7 protocol; Signalling application for the mobility management service on the alpha interface; Part 1: Protocol specification".

[4] EN 300 444 (V1.2): "Digital Enhanced Cordless Telecommunications (DECT); Generic Access Profile (GAP)".

[5] EN 301 144-1 (V1.1): "Integrated Services Digital Network (ISDN); Digital Subscriber Signalling System No. one (DSS1) protocol and Signalling System No.7 protocol; Signalling application for the mobility management service on the alpha interface Part 1: Protocol specification".

[6] CCITT Recommendation I.130 (1988): "Method for the characterization of telecommunication services supported by an ISDN and network capabilities of an ISDN".

[7] ITU-T Recommendation I.210 (1993): "Principles of telecommunication services supported by an ISDN and the means to describe them".

[8] ITU-T Recommendation I.221 (1993): "Common specific characteristics of services".

[9] CCITT Recommendation Q.9 (1988): "Vocabulary of switching and signalling terms".

[10] CCITT Recommendation X.680 (07/94): "Abstract Syntax Notation one: Specification of basic Notation".

[11] CCITT Recommendation X.219 (1988): "Remote Operations: Model, notation and service definition".

[12] CCITT Recommendation Z.100 (1988): "Specification and Description Language (SDL)".

[13] ETS 300 175-5 (1995): "Digital Enhanced Cordless Telecommunications (DECT); Common Interface (CI); Part 5: Network (NWK) layer".

[14] ETS 300 175-6 (1995): "Digital Enhanced Cordless Telecommunications (DECT); Common Interface (CI); Part 6: Identities and addressing".

[15] ETS 300 175-7 (1995): "Digital Enhanced Cordless Telecommunications (DECT); Common Interface (CI); Part 7: Security features".

[16] EN 300 196-1 (V1.2): "Integrated Services Digital Network (ISDN); Generic functional protocol for the support of supplementary services; Digital Subscriber Signalling System No. one (DSS1) protocol; Part 1: Protocol specification".

[17] ETS 300 787 (1997): "Digital Enhanced Cordless Telecommunications (DECT); Global System for Mobile communications (GSM); Integrated Services Digital Network (ISDN); DECT access to GSM via ISDN; General description of service requirements".

[18] ETS 300 788 (1997): "Digital Enhanced Cordless Telecommunications (DECT); Global System for Mobile communications (GSM); Integrated Services Digital Network (ISDN); DECT access to GSM via ISDN; Functional capabilities and information flows".

[19] EN 301 140-1: "Intelligent Network (IN); Intelligent Network Capability Set 2 (CS2); Intelligent Network Application Protocol (INAP); Part 1: Protocol specification".

[20] (reserved)

[21] CCITT Recommendation E.164 (1991): "Numbering plan for the ISDN era".

[22] ITU-T Recommendation I.112 (1993): "Vocabulary of terms for ISDN".

2.2 Informative references

(none)

3 Definitions and abbreviations

3.1 Definitions

For the purpose of the present document, the following definitions apply:

Authentication Code (AC): The AC may be held in non-volatile memory within the PP or it may be manually entered by the user when required for an authentication service. This depends on the application; see ETS 300 175-7 [15].

access rights: An indication that the cordless terminal has appropriate permission to use the CTM service.

Access Rights Identity (ARI): An identity which is globally unique to a service provider and which shows the access rights related to the service provider.

authentication: A security mechanism allowing the verification of the provided identity.

cordless terminal mobility: The ability of a cordless terminal to be mobile within and between Fixed Parts. The mobility may be continuous while the terminal is accessing and using the telecommunication services offered by the network, and it may include the capability of the networks to keep track of the cordless terminal's location throughout the entire network.

CTM identity: The identity by which a user is known to the CTM service providers and networks supporting CTM, and it is used for flexibility and security purposes. The CTM identity identifies a user unambiguously. The CTM identity does not need to be known by users.

CTM number: Number that uniquely and unambiguously identifies each CTM user. It is used by a calling party to reach the CTM user. The number is independent of the calling terminal, network or service used and conforms to CCITT Recommendation E.164 [21].

DECT Paging: A DECT procedure which establishes a link on DECT interface.

Fixed Part (FP): A physical grouping that contains all elements in the cordless network between the local network and the cordless terminal air interface.

Fixed Termination (FT): A logical group of functions that contains all of the Cordless Terminal (CT) Network specific processes and procedures on the fixed side of the air interface. A Fixed Radio Termination only includes elements that are defined in the relevant CT specifications. This includes radio transmission elements (layer1) together with a selection of layer 2 and layer 3 elements.

FT Address: The address of the FT (i.e. an E.164 address).

Portable Part (PP): A physical grouping that contains all elements between the user and air interface. Portable Part is a generic term that may describe one or several physical pieces.

Portable Termination: A logical group of functions that contains all of the CT processes and procedures on the portable side of the CT air interface. A Portable radio Termination only includes elements that are defined in the relevant CT specification.

RANDom challenge: This parameter is used for authentication (see ETS 300 175-7 [15]).

RES1: See ETS 300 175-7 [15].

RES2: See ETS 300 175-7 [15].

RS: A value used to establish authentication session keys, as defined in subclause 4.4.3 of ETS 300 175-7 [15].

roaming: Movement of the cordless terminal without a call in progress from one location area to another location area within the same and/or between different networks supporting the CTM service.

service feature: A specific aspect of a telecommunication service that can also be used in conjunction with other telecommunication services as part of a commercial offering. It is either a core part of a telecommunication service or an optional part offered as an enhancement to a telecommunication service.

service profile: A record containing all the service information related to a user.

User Authentication Key (UAK): Secret authentication data contained within the subscriber's registration data. It is uniquely associated with the particular subscriber (user) and the subscription. The UAK is held in non-volatile memory within the PP (or within a detachable DECT Authentication Module (DAM); see ETS 300 175-7 [15].

3.2 Abbreviations

AC	Allocation Code
ACM	Address Complete Message
ARC	Access Right Class
ARD	Access Right Details
ARI	Access Right Identity
ASE	Application Service Entity
BCSM	Basic Call State Machine
BCUSM	Basic Call Unrelated State Machine

CCAF	Call Control Agent Function
CCF	Call Control Function
CLIR	Calling Line Identification Restriction
CTM	Cordless Terminal Mobility
CTMid	CTM identity
CUCF	Call Unrelated Control Function
CUSF	Call Unrelated Service Function
DCK	Derived Cipher Key
DECT	Digital Enhanced Cordless Telecommunication
DN	Distinguished Name
DP	
DSS1	Digital Subscriber Signalling System 1
EN	European Norm
FE	Functional Entity
FT	Fixed Termination
FP	Fixed Part
GAP	Generic Access Profile
GFT	Generic Functional Transport
IAM	Initial Address Message
IPEI	International Portable Equipment Identity
IPUI	International Portable User Identity
IN	Intelligent Network
INAP	IN Application Part
LA	Location Area
LAL	
LE	Local Exchange
MCID	Malicious Call IDentification
MM	Mobility Management
MMF	Mobility Management Function
MSC	Message Sequence Chart
NCICS	Networked Call Independent Connection-Oriented Signalling
PARK	Portable Access Right Key
PLI	PARK Length Indicator
PP	Portable Part
PT	Portable Terminal
PUN	Portable User Number
PUT	Portable User Type
RAND	RANDom challenge
RES	RESponse
RFPI	Radio Fixed Part Identity
SCF	Service Control Function
SCFmm	SCF mobility management
SCFmmHome	SCF mobility management Home
SCFmmVisited	SCF mobility management Visited
SCFsl	SCF service logic
SCFslHome	SCF service logic Home
SCFslVisited	SCF service logic Visited
SCP	Service Control Point
SDF	Service Data Function
SDFmm	Service Data Function mobility management
SDFsl	Service Data Function service logic
SDP	Service Data Point
SMF	Service Management Function
SSF	Service Switching Function
SSI	Service to Service Information
SSP	Service Switching Point
STUI	Service To User Information
TDP	
UAK	User Authentication Key
USI	User to Service Information

UTSI User To Service Information

4 General

In this stage 3 description it is considered to separate the mobility management from the service control function. In this respect the SCF mobility management (SCFmm) (Visited and Home) handles the Mobility Management (MM) procedures in connection with the Call Unrelated Control Function (CUSF), and the SCF service logic (SCFsl) (Visited and Home) handles the Call Control (e.g. provide roaming number) and CTM supplementary services (e.g. CTM-CLIR, CTM-MCID) in connection with the Service Switching Function (SSF).

To optimize the hierarchical distribution of location information and authentication data across 3 levels (see figure 1), it should be noted that in the figure the SCFmm to SDFsl and SCFsl and SDFmm interface are not required. For optimizing the hierarchical distribution of mobility management and Call Control information, it should be noted that for phase 1 SDFmm (Visited and Home) and SDFsl (Visited and Home) are located in one node (SDFv and SDFh), and that SCFmm (Visited and Home) and SCFsl (Visited and Home) are located in one node (SCFv and SCFh).

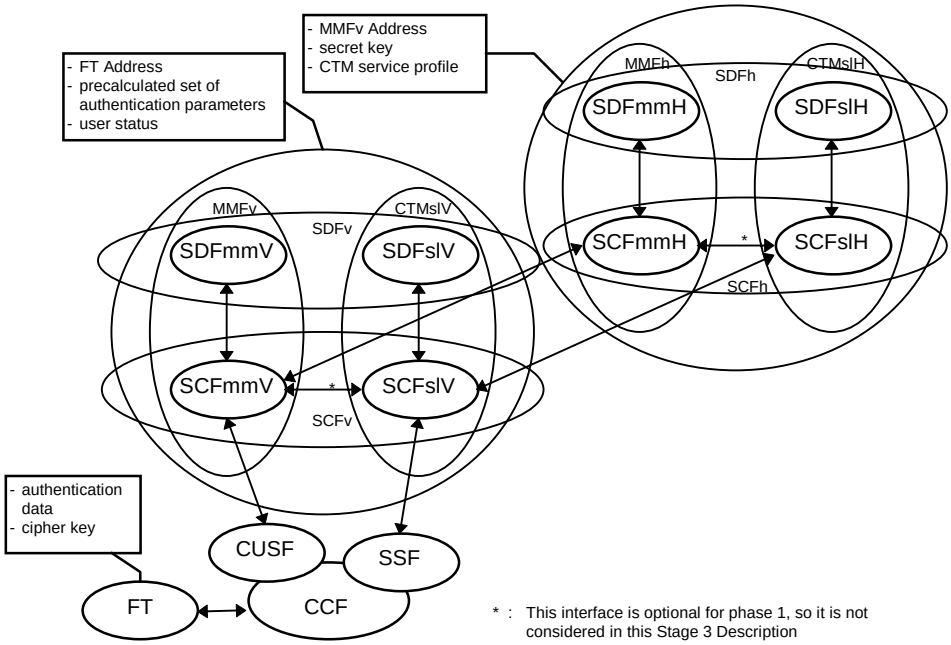


Figure 1: CTM phase 1 architecture

With regard to the architecture to be considered, it is proposed to support in a first phase the topology for one single operator but it must be ensured that easy introduction of inter-networking between operators is accomplished without major impact in the network at the introduction of interworking. This principle will entail that the procedures are to be harmonized so that it can be avoided that extra procedures are required for knowing whether the ongoing procedure (e.g. location registration, call establishment, etc.) is handled as an intra- or inter-operator procedure; this should only be an addressing issue.

4.1 Proposed functional models for the CTM procedures

Figures 2, 3 and 4 propose the functional model for the CTM mobility management (see figure 2) , outgoing CTM call (see figure 3) and incoming CTM call (see figure 4).

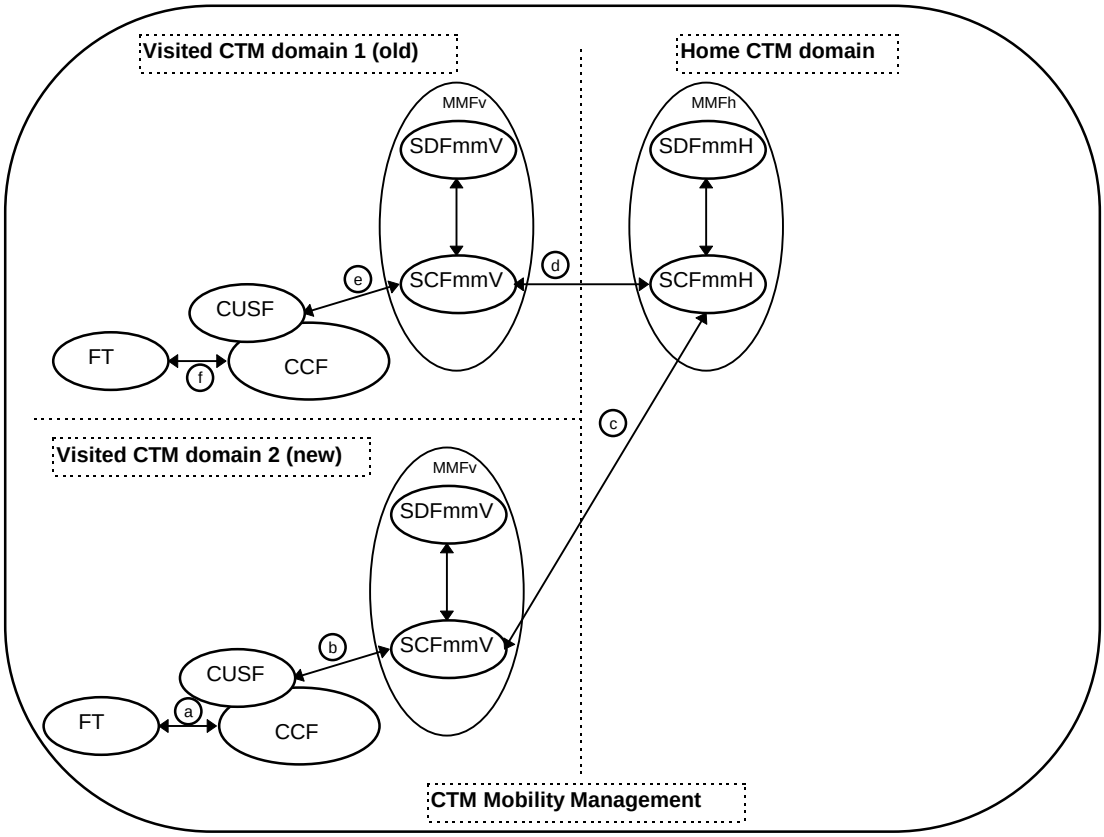


Figure 2: CTM mobility management functional domains

The following procedures are examples of the use of the indicated interfaces:

a) NCICS connection with CTM mobility management information (e.g. location registration).

b) User to Service Information (USI) mechanism containing CTM mobility management information (e.g. location registration).

c) static triggering (e.g. subscription registration) or triggering based on International Portable User Identity (IPUI) (e.g. location update), Service to Service Information (SSI) mechanism containing CTM mobility management information.

d) triggering of SCFmmVisited by SCFmmHOME$_{ome}$, SSI mechanism containing CTM mobility management information (e.g. location cancellation).

e) USI mechanism containing CTM mobility management information (e.g. location cancellation).

f) NCICS connection with CTM mobility management information (e.g. location cancellation).

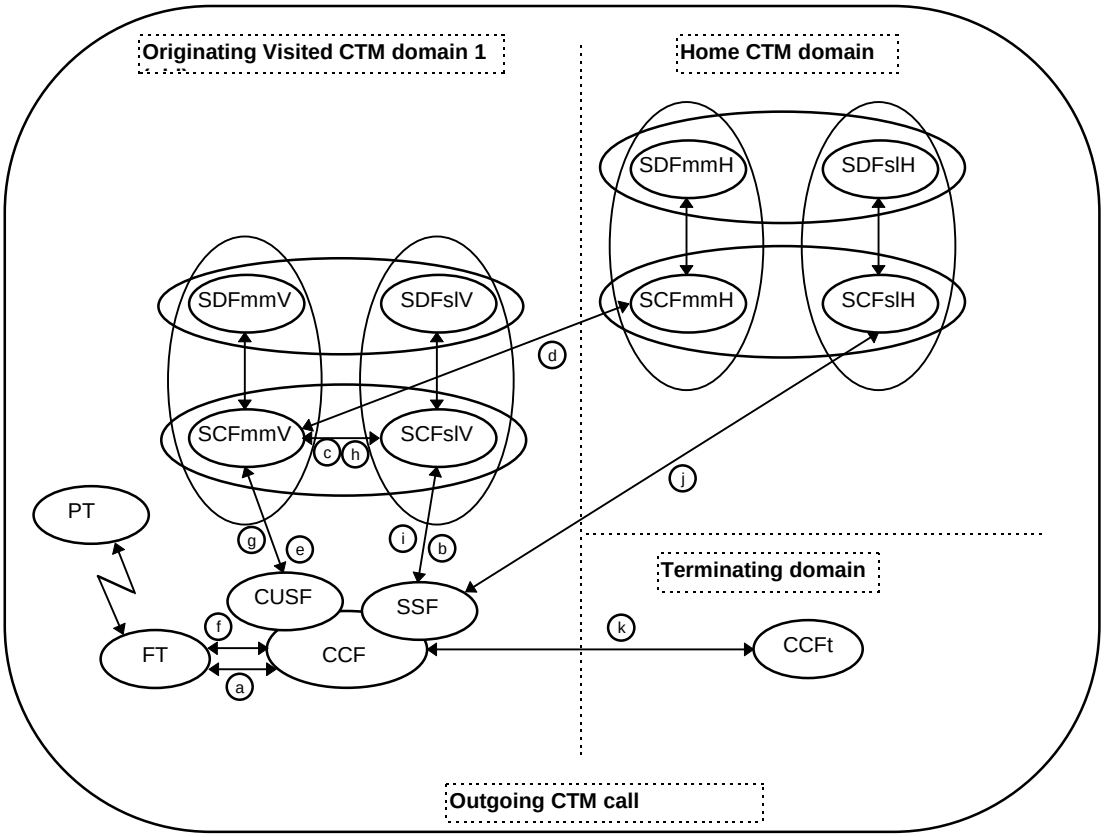


Figure 3: Outgoing CTM call functional domains

Detection and handling of emergency call setup will be completely handled in the SCFmmVisited.

The following procedures are examples of the use of the indicated interfaces:

a) call setup;

b) line base triggering;

c) (optionally) initiation of authentication and/or ciphering procedures;

d) fetching and reception of authentication data (SSI mechanism);

e) authentication and/or ciphering, initiated by SCFmmVisited (USI mechanism);

f) authentication and/or ciphering procedures and the result;

g) result of authentication and/or ciphering;

h) (optionally) action on the call setup based on the result of authentication;

i) subsequent triggering initiated by SCFslVisited;

j) triggering of SCFslHOME$_{ome}$ based on SCFid and check of CTM supplementary services;

k) call establishment to the CTM user.

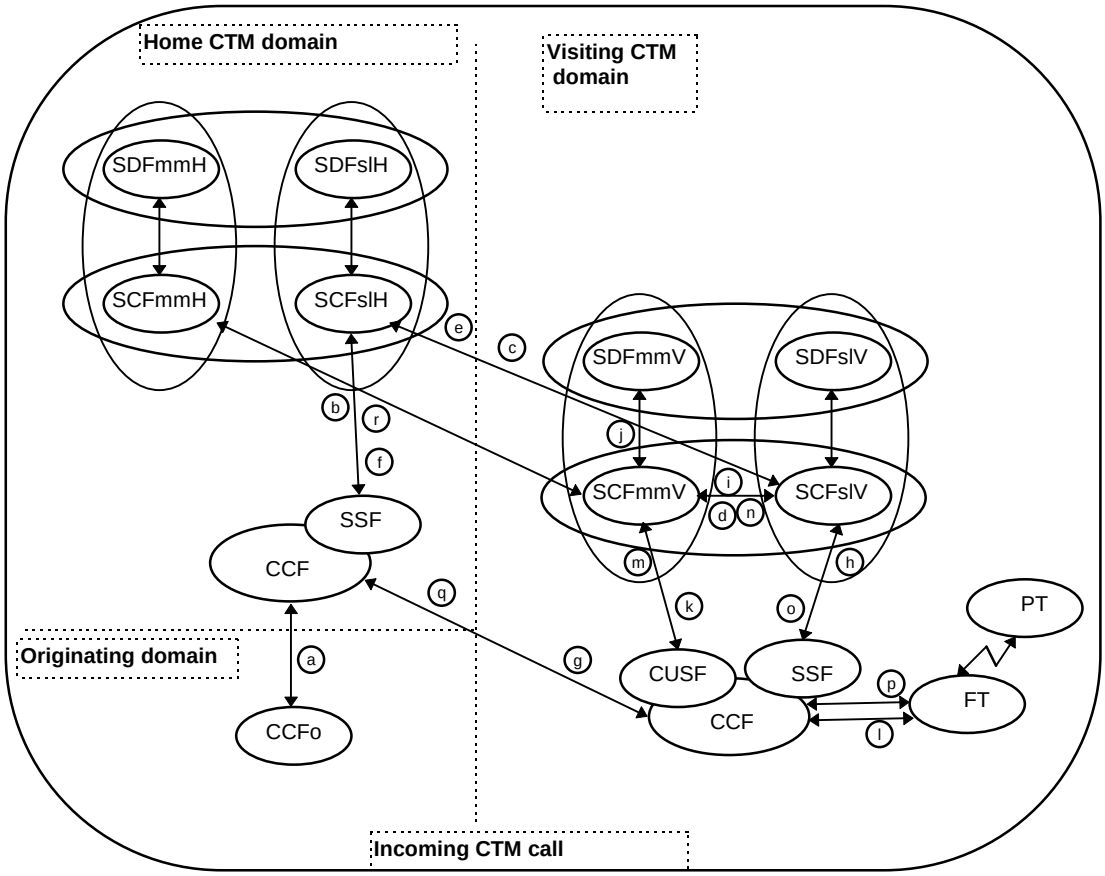


Figure 4: Incoming CTM call functional domains

For the incoming call procedure only the roaming number case is used.

The following procedures are examples of the use of the indicated interfaces:

a) CTM call setup to CTM user using CTM-number;

b) triggering based on CTM-number;

c) request roaming number (SSI mechanism);

d) (optionally) fetching of user status (e.g. not_reachable);

e) roaming number and user status provided (SSI mechanism);

f) call establishment to CTM user using roaming number;

g) call setup to visiting domain using roaming number;

h) triggering of SCFslVisited;

i) (optionally) initiation of authentication and/or ciphering procedures;

j) fetching and reception of authentication data;

k) authentication and/or ciphering procedures, initiated by SCFmmVisited;

l) authentication and/or ciphering procedures, and result;

m) report of authentication and/or ciphering to SCFmmVisited;

n) (optionally) action on call setup based on result of authentication and/or ciphering;

o) call establishment to CTM user;

p) call setup to FT;

q) report of call setup;

r) report of events to SCFslHOME$_{ome}$ to check CTM supplementary services.

A CTM user to CTM user call shall be a combination of a CTM outgoing call and a CTM incoming call, as they are presented in the figures.

5 CTM procedures

5.1 Model

The SDL model of the CTM network is included to this standard as an informative part of the standard in annex C. This subsection provides a short introduction to the SDL model.

The SDL diagrams reflect an Application Layer Structure (ALS) where the SACF performs the association co-ordination function.

The Visited SCF has four Application Service Entities (ASEs) namely:

- SCFVis, having process type SCF_SSF_CUSF_ASE_Visited, for communication with the SSF/CUSF in the Visited Domain.
- SCFmmVisApp, having process type SCFmmVisitedApplication, performing the service logic for the CTM Phase 1 mobility management call unrelated applications in the Visited SCF Domain.
- SCFslVisApp, having process type SCFslVisitedApplication, performing the service logic for the CTM Phase 1 call related applications in the Visited SCF Domain.
- SCF_SCFVis, having process type SCF_SCF_ASE_Visited, for communcation with the SCF in the Home Domain.

The Home Domain SCF has four Application Service Entities (ASEs) namely:

- SCFHom, having process type SCF_SSF_CUSF_ASE_Home, for communication with the SSF/CUSF in the Home Domain.
- SCFmmHomApp, having process type SCFmmHomeApplication, performing the service logic for the CTM Phase 1 mobility management call unrelated applications in the Home SCF Domain.
- SCFslHomApp, having process type SCFslHomeApplication, performing the service logic for the CTM Phase 1 call related applications in the Home SCF Domain.
- SCF_SCFHom, having process type SCF_SCF_ASE_Home, for communcation with the SCF in the Visited Domain.

The SCFHom and SCFVis ASEs communicate with the TCAP adaptors for the CUSF and SSF entities, while the SCF_SCFHom and SCF_SCFVis communicate with the TCAP adaptors for the peer SCF entities.

The association instance for the SCFHom and SCFVis is identified by either the cUSFdialogueID for call unrelated CUSF communication or by the sSFdialogue for call related SSF communication.

The association instance for the SCF_SCFHom and SCF_SCFVis is identified by the sCFdialogueID for communication with the peer SCF entity. The association instance for the SCFmmVisApp and SCFmmHomApp is identified by the mMdialogueID while for the SCFslVisApp and SCFslHomApp is identified by the sLdialogueID.

During one dialogue unique invoke identifiers are generated. They are obtained by calling a GetInvokeID procedure within the send signal parameter e.g. ContinueAssociationRequest((call GetInvokeID), cUSFdialogueID, conaArg)

The communication of the ASEs performing the transport functions via the SACF are given by the appropriate signal name e.g.RequestReportBCUSMEvent for the invoke component, by RequestReportBCUSMEventRes for the return result component and by RequestReportBCUSMEventErr for the return error component.

The association instances for the SCFmmVisApp, SCFmmHomApp, SCFslVisApp and SCFslHomApp ASEs are identified by the Process Instance Identifiers (PID).

Since the signals sent from or received by the transport ASEs by/from the application processes are not unique due to the fact that

- relaying of signals from the SCFHome occur in the SCFVisited SACF, and
- embedded procedures are invoked for the same entity,

it is required that different dialogue identifiers are used. In this repect a qualifier will indicate the nesting which has occured e.g. cUSF1dialogueID.

For the ASEs the creation of an application process instance is initiated by sending a CreateASE signal with a parameter indicating the process which is addressed e.g SCFmmVisApp. The processes addressed are indicated by a parameter value having the type

```
TypeOfASE ::= ENUMERATED {
-- Indicates the type of ASE to be created by SACF
  SCFmmVisApp,
  SCF_SCFVis,
  SCFmmHomApp,
  SCF_SCFHom,
  SCFVis,
  SCFHom
}
```

The process instances are terminated by sending an ASETerminated(x)signal, where x designates the TypeOfASE e.g. SCFmmVisApp.

For the TCAP simulator the signal ApplicationBegin and ApplicationEnd are used.

5.2 Subscription registration in SCFmmSubscription

An example for this procedure can be found in clause A.1.

5.2.1 Procedure description

5.2.1.1 Introduction

This procedure allows the CTM user to get his subscription data (IPUI and Portable Access Right Key (PARK)) from a dedicated SCF, called SCFmmSubscription. Since more than one SCFmmSubscription can exist in a network, the addressing of the appropriate SCFmmSubscription shall be defined by command in each SCFmmVisited.

Two options are defined for searching the IPUI/PARK pair:

1) based on the iPEI. Such iPEI shall be known in advance by the operator (when the CTM user takes his/her subscription);
2) based on the generated authentication data. Precalculated sets of authentication data are used to find the CTM user by initiating a terminal authentication and comparing the terminal authentication result with the precalculated authentication result.

5.2.1.2 Normal operation

In order to obtain the subscription data from the SCFmmSubscription, the SCFmmVisited shall send a cTMAccessRightsRequestSCF invoke component to the SCFmmSubscription by using the SSI mechanism of the SCF-SCF interface.

The following parameters have to be included:

- iPEI, identifying the equipment identity and only used in option 1;
- cTMAuthType, indicating the authentication algorithm (DSSA), the authentication key type (AC or UAK) and the authentication key number;
- cTMPortableCapabilities, to convey the cordless terminal's capabilities (tone, display, etc.).

If both the terminal authentication (initiated by either SCFmmVisited or SCFmmSubscription) and the ciphering procedures (optional procedures) are successful, and if the network authentication (initiated by the PP) is successful, then the SCFmmSubscription shall send a cTMAccessRightsRequestSCF return result component to the SCFmmVisited, using the SSI mechanism of the SCF-SCF interface with the following parameters:

- iPUI, containing the newly assigned identity of the cordless terminal;
- cTMFixedIdentity, indicating the type (PARK), the AccessRightClass (ARC), the AccessRightDetails (ARD), and the length of the identity;
- cTMServiceClass (optional), indicating the service class of the cordless terminal.

5.2.1.3 Exceptional procedure

If the SCFmmSubscription is not able to perform a subscription registration, it shall send a cTMAccessRightsRequestSCF return error component to the SCFmmVisited, using the SSI mechanism of the SCF-SCF interface with the following error values:

- portableIdentityUnknown, if the identity of the cordless terminal (iPEI), for which the request has been initiated, is not known by the SCFmmVisited;
- unspecified, if the requested procedure fails for any other reason.

5.2.2 SDL application procedures

See annex C.

5.3 Key allocation

In case of a subscription registration procedure, SCFmmSubscription may optionally initiate a ciphering procedure.

The cTMciphering invoke, return result and return error are imported from the DSS1 protocol specification; see EN 301 144-1 [3], clause "Invocation and operation", subclause "User initiated ciphering".

The cTMCiphering invoke component sent by SCFmmSubscription using the SSI mechanism over the SCF-SCF interface to SCFmmVisited will be transferred transparently by SCFmmVisited over the CUSF-SCF interface using the USI mechanism.

5.4 Subscription deregistration

An example for the procedure can be found in clause A.2.

The cTMAccessRightsTerminate invoke, return result and return error are imported from the DSS1 protocol specification (see EN 301 144-1 [3], clause "Registration and deregistration", subclause "Subscription deregistration").

When a subscription is removed to a CTM user, SCFmmHome initiates a cTMAccessRightsTerminate invoke component to SCFmmVisited using the SSI mechanism over the SCF-SCF interface. The invoke component is transferred transparently to the CUSF via the USI mechanism over the SCF-CUSF interface.

5.5 Network authentication

In case of both the subscription registration and subscription deregistration procedures, the PP may initiate a network authentication.

The cTMNetworkAuthentication invoke, return result and return error are imported from the DSS1 protocol specification (see EN 301 144-1 [3], clause "Invocation and operation" subclause "Network authentication").

The cTMNetworkAuthentication invoke component received by SCFmmVisited over the CUSF-SCF interface using the USI mechanism is transferred transparently to SCFmmHome using the SSI mechanism over the SCF-SCF interface.

5.6 Terminal authentication

In case of a subscription registration procedure, SCFmmSubscription may optionally initiate a Terminal Authentication.

The cTMTerminalAuthentication invoke, return result and return error are imported from the DSS1 protocol specification (see EN 301 144-1 [3], clause "Invocation and operation" subclause "Terminal authentication").

The cTMTerminalAuthentication invoke component sent by SCFmmSubscription using the SSI mechanism over the SCF-SCF interface to SCFmmVisited will be transferred transparently by SCFmmVisited over the CUSF-SCF interface using the USI mechanism.

5.7 Network initiated ciphering

In case of a subscription registration procedure, SCFmmSubscription may optionally initiate a ciphering procedure.

The cTMciphering invoke, return result and return error are imported from the DSS1 protocol specification; see EN 301 144-1 [3], clause "Invocation and operation" subclause "User initiated ciphering".

The cTMCiphering invoke component sent by SCFmmSubscription using the SSI mechanism over the SCF-SCF interface to SCFmmVisited will be transferred transparently by SCFmmVisited over the CUSF-SCF interface using the USI mechanism.

5.8 Location registration in the SCFmmVisited

An example for the procedure can be found in clause A.3.

The cTMLocationRegistration invoke, return result and return error are imported from the DSS1 protocol specification (see EN 301 144-1 [3], clause "Activation and deactivation", subclause "Location registration").

5.9 Location update in the SCFmmHome

An example for the procedure can be found in clause A.4.

5.9.1 Procedure description

5.9.1.1 Introduction

The CTM service provided by the network to the subscriber shall be activated when the subscriber makes its location known to the network by the location registration procedure for the first time or after a period of deactivation. The location update procedure shall provide during a location registration procedure in order to update the home domain and to check whether the CTM user is allowed to roam in the visited domain.

5.9.1.2 Normal operation

Location update procedure is used to make the cordless terminal location known to the network when a cordless terminal is roaming from a defined location area to a new one.

The SCFmmVisited shall translate the PUN part of the cTMPortableIdentity (E.212 number) received during the location registration procedure into a "mobile global title" (E.214 number) and may store it as sCFmmHomeID.

NOTE: The SCFmmHomeID is used later during the outgoing call procedure in order that the SCFmmVisited can initiate a subsequent triggering of the SCFmmHome for checking the outgoing CTM supplementary services.

In order to make the cordless terminal location known to the SCFmmHome and to perform a location update procedure, the SCFmmVisited shall send a cTMLocationUpdateSCF invoke component by using the SSI mechanism of the SCF-SCF interface.

The following parameters shall be included:

- iPUI, indicating the identity of the cordless terminal;
- sCFmmVisitedID, containing the global title of the SCFmmVisited to be addressed by the SCFmmHome.

On receiving the cTMLocationUpdateSCF invoke component, the SCFmmHome shall use the received parameters to perform the location update procedure.

Before sending the cTMLocationRegistrationSCF return result component to the SCFmmVisited, the SCFmmHome shall initiate one or more of the following procedures:

- the SCFmmHome shall check whether the CTM user is allowed to roam into the SCFmmVisited area;
- the SCFmmHome shall check whether the CTM user was previously registered in another SCFmmVisited. If registered the SCFmmHome shall initiate a location cancellation procedure in old SCFmm Visited.

The SCFmmHome having concluded that the CTM user is allowed to roam into the visited domain the location update procedure shall send a CTMLocationUpdateSCF return result component to the SCFmmVisited, by using the using the SSI mechanism of the SCF-SCF interface.

The following parameters shall be included:

- cTMNumber, indicating the E.164 number that uniquely and unambiguously identifies the CTM subscriber;
- cTMFixedIdentity, containing the PARK.

5.9.1.3 Exceptional procedure

If the CTM user is not allowed to roam into the SCFmmVisited, the SCFmmHome shall send a CTMLocationUpdateSCF return error component to SCFmmVisited, by using the SSI mechanism of the SCF-SCF interface, indicating one of the following error values:

- portableIdentityUnknown, if the identity of the cordless terminal, for which the request has been initiated, is not known to the home network;
- cTMDataMissing, if the network cannot provide the security data due to unavailability of the information;
- cTMRoamingNotAllowed, if the user of the cordless terminal is not allowed to roam in the visited network;
- unspecified, if the requested procedure fails for any other reason.

If the SCFmmVisited receives a reject component from the SCFmmHome it shall close the association for location registration.

If during the location registration procedure a release association is received from the CTM user or the FT, this event may be reported to the SCFmmVisited by arming the EDP-R "AssociationReleaseRequested" Basic Call Unrelated State Machine (BCUSM) Event. The arming is performed by means of an RequestReportBCUSMEvent operation and the reporting is performed by means of the eventReportBCUSM operation. The "CTM-user/FT release" shall not be reported to the SCFmmHome and shall close the association for location registration.

5.9.2 SDL application procedures

See annex C.

5.10 Location cancellation in SCFmmVisited

An example for the procedure can be found in clause A.5.

The cTMLocationCancellation invoke, return result and return error are imported from the DSS1 protocol specification (see EN 301 144-1 [3], clause "Activation and deactivation", subclause "Location cancellation").

When the SCFmmHome initiates a cTMLocationCancellation invoke component to SCFmmVisited using the SSI mechanism over the SCF-SCF interface, the invoke component is transferred transparently by SCFmmVisited to CUSF via the USI mechanism over the SCF-CUSF interface.

5.11 Location cancellation in the FT

An example for the procedure can be found in clause A.6.

The cTMLocationCancellation invoke, return result and return error are imported from the DSS1 protocol specification (see EN 301 144-1 [3], clause "Activation and deactivation", subclause "Location cancellation").

5.12 Restoration of location data in the SCFmmHome

An example for the procedure can be found in clause A.8.

5.12.1 Procedure description

5.12.1.1 Introduction

This procedure allows the SCFmmHome to restore the location information for all CTM users belonging to it when this data is corrupted.

5.12.1.2 Normal operation

1) The SCFmmHome has a table of all the SCFmmVisited addresses where the CTM users may roam. If the location data gets corrupted in the SCFmmHome e.g. after restart, the SCFmmHome shall send a message to all SCFmmVisited by means of a cTMRestoreDataSCF invoke component using the SSI mechanism over the SCF-SCF interface. This invoke component indicates to the addressed SCFmmVisited that the restoration of location data in the SCFmmHome shall take place and shall have the sCFmmHomeId parameter.

2) Each SCFmmVisited, on receipt of the cTMRestoreDataSCF invoke component shall mark <restoration of location info in SCFmmHome required> for all CTM users belonging to that SCFmmHome. To find all CTM users, the SCFmmVisited shall compare the SCFmmHomeId stored at the location update result procedure in the SCFmmVisited with the one received in the restoration of location data procedure.

3) When the CTM user performs an "Outgoing call" procedure or a "location registration in the SCFmmVisited" procedure , the SCFmmVisited shall initiate a "location update in the SCFmmHome" procedure and shall set mark <restoration of location info in SCFmmHome not required>. This option implies that CTM users will not be reachable until either they initiate an outgoing call or perform a location registration.

5.12.1.3 Exceptional procedure

No return result is expected for this procedure, since action on the reception of the cTMRestoreDataSCF invoke component will be performed later (only at CTM outgoing call and location registration).

No return error is expected for this procedure, since SCFmmHome is in an unstable situation.

5.12.2 SDL application procedures

See annex C.

5.13 Terminal authentication and/or ciphering

An example for the procedures can be found in clause A.9.

The cTMTerminalAuthentication and cTMCiphering invoke, return result and return error are imported from the DSS1 protocol specification (see EN 301 144-1 [3], clause "Invocation and operation", subclause "Terminal authentication and network initiated ciphering").

5.14 Download of security data to SCFmmVisited

An example for the procedure can be found in clause A.7.

5.14.1 Procedure description

5.14.1.1 Introduction

The authentication and the ciphering procedures shall always be processed by an independent dialogue using the call unrelated concept and shall always be performed by the SCFmmVisited (see note).

Terminal authentication and/or ciphering procedures shall be able to be initiated in the following cases:

- CTM outgoing call;
- CTM incoming call;
- location registration procedure;
- subscription registration procedure.

Network authentication procedure shall be able to be initiated in the following cases:

- subscription registration procedure;
- subscription deregistration procedure.

For the ciphering and authentication the updating shall be performed by an independent procedure where e.g. a set of pre-calculated quadruplet values is transferred independently from the call set-up procedures where a set of quadruplet instances will be consumed when the ciphering and authentication procedures are invoked.

The terminal authentication and/or ciphering procedures may be processed in parallel or in series with the call set-up. In case the ciphering procedure is done in parallel, the dialled digits may not be ciphered. In case the ciphering procedure is done in series, the old Derived Cipher Key (DCK) may be used.

Since two independent procedures for terminal authentication/ciphering and call set-up are done, co-ordination has to be performed in the SCFmmVisited using the IPUI. For example, if the authentication failed, the call related part shall be released.

The terminal authentication result (if ignored) will be ignored in case of an emergency call.

NOTE: In case of subscription registration procedure, as an option it may be possible to be initiated from the SCFmmSubscription instead of SCFmmVisited.

5.14.1.2 Normal operation

In order to obtain the security data for authentication and ciphering from the SCFmmHome, the SCFmmVisited shall send a cTMProvideSecurityDataSCF invoke component by using the SSI mechanism of the SCF-SCF interface.

The following parameter shall be included:

- IPUI, indicating the identity of the cordless terminal.

The SCFmmHome on receipt of the cTMProvideSecurityDataSCF invoke component shall retrieve a number of sets of authentication and optionally ciphering data from SCFmmHome and shall send to the SCFmmVisited an cTMProvideSecurityDataSCF return result component using the SSI mechanism of the SCF-SCF interface.

The following parameter shall be included:

- securityDataSets, containing the number or sets {RAND, RS, XRES1, DCK, cTMAuthType) that have been generated.

5.14.1.3 Exceptional procedure

If the provide security data procedure fails, the SCFmmHome shall send a cTMProvideSecurityDataSCF return error component to the SCFmmVisited, by using the SSI mechanism of the SCF-SCF interface, indicating one of the following error values:

- portableIdentityUnknown, if the identity of the cordless terminal for which the request has been initiated is not known;

- unspecified, if the requested procedure fails for any other reason.

5.14.2 SDL application procedures

See annex C.

5.15 Provide roaming number

An example for the procedure can be found in clause B.2.

5.15.1 Procedure description

5.15.1.1 Introduction

This procedure shall allow the SCFslHome to retrieve during an incoming call the roaming number of the CTM dialled subscriber in the SSFslVisited.

For the CTM incoming call the originating party dials the directory number of a mobile terminal (CTM number).

If the CTM number is a non-geographical number the call shall be routed to the nearest SSF of the calling party.

If a CTM number is a geographical number the call shall be routed to the Call Control Agent Function (CCF) / SSFHome.

The SSF shall trigger the SCFsl in the home domain. The SCFslHome shall retrieve the address of the visited SCFmm. With the obtained SCFmm address, the SCFslHome shall request the SCFslVisited for the roaming number.

The SCFslVisited shall check if the subscriber is registered and reachable, shall select a roaming number, shall make an association between the retrieved roaming number and the IPUI, shall start a timer for the supervision of the use of the roaming number, and shall return the roaming number to the SCFslHome.

The SCFslHome shall instruct the originating SSF to route the call with the obtained number by e.g. ISUP procedures to the visited CCF/SSF, where the call is triggered.

The visited SSF shall query the SCFslVisited based on the roaming number in order to get the FT address and the IPUI/PARK. The SCFslVisited shall release the roaming number, stop the supervision timer and shall ask the visited CCF/SSF to route the call towards the identified FT. The FT shall then page the PT.

5.15.1.2 Normal operation

The SCFslHome shall request for a roaming number in the SCFslVisited, by sending a cTMProvideRoamingNumberSCF invoke component by using the SSI mechanism of the SCF-SCF interface.

The following parameter shall be included:

- IPUI, indicating the identity of the cordless terminal.

If the CTM user is registered in the SCFslVisited, the SCFslVisited shall return the CTMProvideRoamingNumber return result component using the SSI mechanism of the SCF-SCF interface having the following parameters:

- roamingNumber, containing the roaming number to be used for routeing in the ISUP to the visited CCF/SSF;

- userStatus, set to userReachable.

If the CTM user is not registered in the SCFslVisited, the SCFslVisited shall return the cTMProvideRoamingNumberSCF return result component using the SSI mechanism of the SCF-SCF interface having the following parameter:

- userStatus, set to userNotReachable or userUnknown.

The SCFsl can execute the CTM supplementary service CFNRc , if active for the user.

5.15.1.3 Exceptional procedure

If the provide roaming number procedure fails, the SCFslVisited shall send a cTMProvideRoamingNumberSCF return error component to the SCFslHome, by using the SSI mechanism of the SCF-SCF interface, indicating one of the following error values:

- cTMFacilityNotSupported;

- cTMNoRoamingNumberAvailable;

- portableIdentityUnknown, if the identity of the cordless terminal, for which the request has been initiated, is not known to the visited network;

- unspecified, if the requested procedure fails for any other reason.

5.15.2 SDL application procedures

See annex C.

6 Use of INAP

6.1 SCF-SCF service to service extensions

6.1.1 ASN.1 module

6.1.2 Data types

```
SSI-SCF-SCF-datatypes {ccitt(0) identified-organisation(4) etsi(0) inDomain(1) in-ctm(3) modules(1)
datatypes(25) version1(0)}

DEFINITIONS ::=

BEGIN

IMPORTS
    SSI-PARAMETERS-BOUND,
```

```
FROM SSI-SCF-SCF-classes {ccitt(0) identified-organisation(4) etsi(0) inDomain(1) in-ctm(3)
modules(1) classes (22) version1(0)}

STSIInformation {SSI-PARAMETERS-BOUND : SSI-bound} ::= OCTET STRING (SIZE (
    SSI-bound.&minSSIInformationLength..SSI-bound.&maxSSIInformationLength))

-- The STSIInformation contains the service information associated with the
agreementFeatureIndicator. provided by the
-- Service Logic of the invoking SCF to be transferred to the Service Logic of the responding SCF.s
-- For USI-SSI interworking the maxUSIInformationLength and the maxSSIInformationLength shall be the
same.
-- The maximum of those values are depending also on the used TCAP version.
AgreementFeatureIndicator {SSI-PARAMETERS-BOUND : SSI-bound} ::= CHOICE {
    global  OBJECT IDENTIFIER,
    local   OCTET STRING (SIZE (
                SSI-bound.&minAgreementFeatureIndicatorLength..
                SSI-bound.&maxAgreementFeatureIndicatorLength))
    }

--This AgreementFeatureIndicator indicates the service application for which the Seville data is to
be provided.

END
```

6.1.3 Error types

```
-- No new error types are defined for SSI SCF-SCF in addition to CS2.
```

6.1.4 Operation codes

```
SSI-SCF-SCF-operationcodes {ccitt(0) identified-organisation(4) etsi(0) inDomain(1) in-ctm(3)
modules(1) operationcodes(21) version1(0)}

DEFINITIONS ::=

BEGIN

IMPORTS

    Code
FROM Remote-Operations-Information-Objects ros-InformationObjects

ros-InformationObjects FROM  IN-CS2-object-identifiers
        { ccitt(0) identified-organization(4) etsi(0) inDomain(1) in-network(1) CS-2(20) modules(0)
in-cs2-object-identifiers(17) version1(0) }

-- SCF/SCF interface including transferSTSI

    opcode- transferSTSI        Code ::= local : 200

END
```

6.1.5 Error codes

```
-- No new error codes are to be defined SSI SCF-SCF.
```

6.1.6 Classes

```
SSI-SCF-SCF-classes {ccitt(0) identified-organisation(4) etsi(0) inDomain(1) in-ctm(3) modules(1)
classes (22) version1(0)}
DEFINITIONS ::=
BEGIN


SSI-PARAMETERS-BOUND ::= CLASS
{
    &minSTSIInformationLength       INTEGER,
    &maxSTSIInformationLength       INTEGER,
    -- The maxSTSIInformationLength and shall not exceed the maximum value allowed such that the
    -- STSIInformation can be transferred by the used TCAP.
    &minAgreementFeatureIndicatorLength       INTEGER,
    &maxAgreementFeatureIndicatorLength       INTEGER
```

```
WITH SYNTAX
{
    MINIMUM-FOR-STSI-INFORMATION    &minSTSIInformationLength
    MAXIMUM-FOR-STSI-INFORMATION    &maxSTSIInformationLength
    MINIMUM-FOR-AGREEMENT-INDICATOR &minAgreementFeatureIndicatorLength
    MAXIMUM-FOR-AGREEMENT-INDICATOR &maxAgreementFeatureIndicatorLength

-- The following instance of the parameter bound is just an example
sSI-networkSpecificBoundSet  SSI-PARAMETERS-BOUND ::= {
    MINIMUM-FOR-STSI-INFORMATION    1
    MAXIMUM-FOR-STSI-INFORMATION    4
    MINIMUM-FOR-AGREEMENT-INDICATOR 1
    MAXIMUM-FOR-AGREEMENT-INDICATOR 2
}
END
```

6.1.7 Object identifiers

```
SSI-SCF-SCF-object-identifiers {ccitt(0) identified-organisation(4) etsi(0) inDomain(1) in-ctm(3)
modules(1) object-identifiers(20) version1(0)}
DEFINITIONS ::=
BEGIN
-- This module assigns object identifiers for SSI SCF-SCF interface

-- For SSI-SCF-SCF Modules
sSI-SCF-SCF-datatypes       OBJECT IDENTIFIER ::=
    {ccitt(0) identified-organisation(4) etsi(0) inDomain(1) in-ctm(3) modules(1) datatypes(25)
version1(0)}
sSI-SCF-SCF-operationcodes      OBJECT IDENTIFIER ::=
    {ccitt(0) identified-organisation(4) etsi(0) inDomain(1) in-ctm(3) modules(1) operationcodes(21)
version1(0)}
sSI-SCF-SCF-classes         OBJECT IDENTIFIER ::=
    {ccitt(0) identified-organisation(4) etsi(0) inDomain(1) in-ctm(3) modules(1) classes (22)
version1(0)}
sSI-SCF-SCF-Operations      OBJECT IDENTIFIER ::=
    {ccitt(0) identified-organisation(4) etsi(0) inDomain(1) in-ctm(3) modules(1) ops_and_args(23)
version1(0)}
sSI-SCF-SCF-Protocol            OBJECT IDENTIFIER ::=
    {ccitt(0) identified-organisation(4) etsi(0) inDomain(1) in-ctm(3) modules(1) sSI-SCF-SCF-pkgs-
contracts-acs(24) version1(0)}

-- SSI-SCF-SCF Application Context
id-ac-SSI-scf-scfOperationsAC   OBJECT IDENTIFIER ::=
    {ccitt(0) identified-organisation(4) etsi(0) inDomain(1) in-ctm(3) ac(3) 20 version1(0)}
-- SSI-SCF-SCF Contract
id-contract-SSI-scf-scf         OBJECT IDENTIFIER ::=
    {ccitt(0) identified-organisation(4) etsi(0) inDomain(1) in-ctm(3) contract(26) 20 version1(0)}
-- SSI-SCF-SCF Operation Packages
id-package-transferSTSI         OBJECT IDENTIFIER ::=
    {ccitt(0) identified-organisation(4) etsi(0) inDomain(1) in-ctm(3) package(27) 20 version1(0)}
-- SSI-SCF-SCF Abstract Syntaxes
id-as-SSI-scf-scfOperationsAS       OBJECT IDENTIFIER ::=
    {ccitt(0) identified-organisation(4) etsi(0) inDomain(1) in-ctm(3) as(5) 20 version1(0)}
END
```

6.1.8 Operations and arguments

```
-- SSI Specific CS2 operations and arguments
SSI-SCF-SCF-ops-args {ccitt(0) identified-organisation(4) etsi(0) inDomain(1) in-ctm(3) modules(1)
ops-and-args(23) version1(0)}

DEFINITIONS IMPLICIT TAGS ::=
BEGIN

IMPORTS

    OPERATION, ERROR
FROM Remote-Operations-Information-Objects ros-InformationObjects
    SecurityParameters,
    securityError
FROM DirectoryAbstractService directoryAbstractService

    OPTIONALLY-PROTECTED{}
FROM EnhancedSecurity enhancedSecurity

    opcode-transferSTSI
FROM SSI-operationcodes sSI-SCF-SCF-operationcodes
```

```
    PARAMETERS-BOUND
FROM IN-CS2-classes classes

    SSI-PARAMETERS-BOUND,
FROM SSI-SCF-SCF-classes sSI-SCF-SCF-classes

    ExtensionField {}
FROM IN-CS2-datatypes datatypes

    AgreementFeatureIndicator {}
    STSIInformation {},
FROM SSI-SCF-SCF-datatypes sSI-SCF-SCF-datatypes

    missingParameter,
    parameterOutOfRange,
    systemFailure,
    unexpectedDataValue,
    unexpectedParameter,
FROM IN-CS2-errortypes errortypes

    scfTaskRefused,
    SCFQOP
FROM IN-CS2-SCF-SCF-ops-args scf-scf-Operations

    ros-InformationObjects, ds-UsefulDefinitions, classes, errortypes, datatypes,
    scf-scf-Operations
FROM IN-CS2-object-identifiers { ccitt(0) identified-organization(4) etsi(0) inDomain(1) in-
network(1) CS-2(20) modules(0) in-cs2-object-identifiers(17) version1(0) }

    directoryAbstractService, enhancedSecurity
FROM UsefulDefinitions ds-UsefulDefinitions

    sSI-SCF-SCF-datatypes,
    sSI-SCF-SCF-operationcodes,
    sSI-SCF-SCF-classes
FROM SSI-SCF-SCF-object-identifiers {ccitt(0) identified-organisation(4) etsi(0) inDomain(1) in-
ctm(3) modules(1) object-identifiers(20) version1(0)}

transferSTSI  {PARAMETERS-BOUND : bound, SSI-PARAMETERS-BOUND : SSI-bound} OPERATION ::= {
    ARGUMENT        TransferSTSIArg {bound, SSI-bound}
    RETURN RESULT   FALSE
    ERRORS      {missingParameter |
            systemFailure |
            scfTaskRefused
            unexpectedDataValue |
            unexpectedParameter |
            parameterOutOfRange |
            securityError
            }
    CODE        opcode-transferSTSI
    }
-- Direction : controlling SCF -> supporting SCF (or IAF), or
-- supporting SCF (or IAF) -> controlling SCF, T imer Tstsi
-- This operation is used by the invoking SCF to request or report  service information from / to
the responding SCF,

TransferSTSIArg {PARAMETERS-BOUND : bound, SSI-PARAMETERS-BOUND : SSI-bound} ::= OPTIONALLY-
PROTECTED {SEQUENCE {
    sSIInfo SSIInfo {SSI-bound},
    securityParameters  [2] SecurityParameters  OPTIONAL,
    extensions  [3] SEQUENCE SIZE (1..bound.&numOfExtensions)
                 OF ExtensionField {bound}  OPTIONAL,
    },
    SCFQOP.&scfArgumentQOP{@scfqop}
    }

SSIInfo{SSI-PARAMETERS-BOUND : SSI-bound}::= SEQUENCE  {
    agreementFeatureIndicator   [0]AgreementFeatureIndicator {SSI-bound} OPTIONAL,
    sTSIInformation [1] STSIInformation {SSI-bound}
}
END
```

The following value ranges do apply for operation specific timers in INAP:
short: 1 - 10 seconds

Table 1 below lists all operation timers and the value range for each timer. The definitive value for each operation timer may be network specific and has to be defined by the network operator.

Table 1

TransferSTSI	Ttstsi	short

6.1.9 SSI-SCF-SCF interface

```
SSI-SCF-SCF-pkgs-contracts-acs {ccitt(0) identified-organisation(4) etsi(0) inDomain(1) in-ctm(3)
modules(1) sSI-SCF-SCF-pkgs-contracts-acs(24) version1(0)}

DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- This module describes the operation-packages, contracts and application-contexts used
--  over the SCF-SCF interface for SSI.

IMPORTS

    PARAMETERS-BOUND,
    networkSpecificBoundSet
FROM IN-CS2-classes classes

    SSI-PARAMETERS-BOUND,
    sSI-networkSpecificBoundSet
FROM SSI-SCF-SCF-classes sSI-SCF-SCF-classes

    CONTRACT, OPERATION-PACKAGE, CONNECTION-PACKAGE, OPERATION
FROM Remote-Operations-Information-Objects ros-InformationObjects

    TCMessage {}
FROM  TCAPMessages tc-Messages

    APPLICATION-CONTEXT, dialogue-abstract-syntax
 FROM TC-Notation-Extensions tc-NotationExtensions

    establishChargingRecord {},
    confirmedReportChargingInformation{},
    confirmedNotificationProvided {},
    handlingInformationRequest {},
    handlingInformationResult {},
    networkCapability {},
    notificationProvided {},
    provideUserInformation {},
    reportChargingInformation {},
    requestNotification {},
FROM IN-CS2-SCF-SCF-ops-args scf-scf-Operations

    transferSTSI {}
FROM SSI-SCF-SCF-ops-args sSI-SCF-SCF-operations

    ds-UsefulDefinitions,
    tc-Messages, tc-NotationExtensions,
    ros-InformationObjects,
    scf-scf-Operations, scf-scf-Protocol,
    ssf-scf-Operations, ssf-scf-Protocol
FROM IN-CS2-object-identifiers { ccitt(0) identified-organization(4) etsi(0) inDomain(1) in-
network(1) CS-2(20) modules(0) in-cs2-object-identifiers(17) version1(0) }

    sSI-SCF-SCF-classes,
    sSI-SCF-SCF-operations,
    id-ac-SSI-scf-scfOperationsAC
    id-contract-SSI-scf-scf
    id-package-transferSTSI
    id-as-SSI-scf-scfOperationsAS
FROM SSI-SCF-SCF-object-identifiers {ccitt(0) identified-organisation(4) etsi(0) inDomain(1) in-
ctm(3) modules(1) object-identifiers(20) version1(0)}

    activityTest
FROM IN-CS2-SSF-SCF-ops-args ssf-scf-Operations

    activityTestPackage
FROM IN-CS2-SSF-SCF-pkgs-contracts-acs ssf-scf-Protocol

    scf-scfConnectionPackage,
   activityTestPackage,
    chargingInformationPackage,
    handlingInformationPackage
    networkCapabilityPackage,
    notificationPackage,
    userInformationPackage
FROM IN-CS2-SCF-SCF-pkgs-contracts-acs scf-scf-Protocol
;

-- Contracts, Packages and Application Contexts
```

```
-- Application Context --

SSI-scf-scfOperationsAC APPLICATION-CONTEXT ::= {
    CONTRACT                SSI-scf-scfContract
    DIALOGUE MODE           structured
    TERMINATION             basic
    ABSTRACT SYNTAXES       {dialogue-abstract-syntax |
                    SSI-scf-scfOperationsAbstractSyntax}
    APPLICATION CONTEXT NAME        id-ac-SSI-scf-scfOperationsAC
}

-- Contract --

SSI-scf-scfContract CONTRACT ::= {
    CONNECTION    scf-scfConnectionPackage(networkSpecificBoundSet)
    INITIATOR CONSUMER OF    {
       activityTestPackage |
       handlingInformationPackage{networkSpecificBoundSet} |
       transferSTSIPackage{networkSpecificBoundSet, sSI-networkSpecificBoundSet }
    RESPONDER OF    {
       activityTestPackage |
       handlingInformationPackage{networkSpecificBoundSet} |
     chargingInformationPackage {networkSpecificBoundSet}|
     networkCapabilityPackage {networkSpecificBoundSet}|
     notificationPackage {networkSpecificBoundSet}|
     userInformationPackage {networkSpecificBoundSet}
       transferSTSIPackage{networkSpecificBoundSet, sSI-networkSpecificBoundSet }
    ID  id-contract-SSI-scf-scf
}

-- transferSTSI package --

transferSTSIPackage  {PARAMETERS-BOUND : bound, SSI-PARAMETERS-BOUND : SSI-bound}   OPERATION-
PACKAGE ::= {
    CONSUMER INVOKES    {transferSTSI {bound, SSI-bound }}
    SUPPLIER INVOKES    {transferSTSI {bound, SSI-bound }}
    ID                  id-package-transferSTSI
    }

-- abstract syntaxes --

SSI-scf-scfOperationsAbstractSyntax ABSTRACT-SYNTAX ::= {
     BasicSSI-SCF-SCF-PDUs
     IDENTIFIED BY          id-as-SSI-scf-scfOperationsAS}
BasicSSI-SCF-SCF-PDUs ::= TCMessage{{SSI-SCF-SCF-Invokable},{SSI-SCF-SCF-Returnable}}
SSI-SCF-SCF-Invokable {PARAMETERS-BOUND : bound,  SSI-PARAMETERS-BOUND : SSI-bound} OPERATION ::= {
    activityTest |
    establishChargingRecord {bound}|
    confirmedNotificationProvided {bound}|
    confirmedReportChargingInformation {bound} |
    handlingInformationRequest {bound} |
    handlingInformationResult {bound}|
    networkCapability {bound}|
    notificationProvided {bound}|
    provideUserInformation {bound}|
    reportChargingInformation {bound}|
    requestNotification {bound}
    transferSTSI {bound, SSI-bound }
}
SSI-SCF-SCF-Returnable {PARAMETERS-BOUND : bound, SSI-PARAMETERS-BOUND : SSI-bound} OPERATION ::= {
    activityTest |
    establishChargingRecord {bound}|
    confirmedNotificationProvided {bound}|
    confirmedReportChargingInformation {bound}|
    handlingInformationRequest {bound}|
    handlingInformationResult {bound} |
    networkCapability {bound}|
    provideUserInformation {bound}|
    requestNotification {bound}
    transferSTSI {bound, SSI-bound }
}
END
```

6.1.10 TransferSTSIInformation operation

6.1.10.1 Procedure descriptions

6.1.10.1.1 General description

This operation may be used from the controlling SCF to the supporting SCF - and vice versa - to exchange service information. The received service information will be used in the SCF to determine how the service is to be proceeded. The type of service information is used to assist the service logic execution and may consists of e.g. call unrelated as well as call related data.

The operation may be used to request service data and report service data results. It may be sent during an established relationship. If a result is requested the correlation between send service data request and reported service data result is to be made on the service application level.

6.1.10.1.1.1 Parameters

- sSIInfo

  This parameter identifies a service application (e.g. a service feature) within the service logic type supported across the SCF -SCF interface which is indicated in the agreement ID in the SCF bind operation. This parameter also contains associated service information.

  The agreement feature indicator may correspond to a service application which is standardized at the ITU level or by any regional standardization body (i.e. global). It may also correspond to a service application which is significant only within one network or group of cooperating networks in which the service application identified is significant (i.e. local).

  - agreementFeatureIndicator:

    This parameter indicates the service application for which the associated service information within the agreement ID is to be provided. The scope of the agreementFeatureIndicator is local to an AgreementID and applicable within the context of an requested type.

  - sTSIInformation:

    This parameter conveys service information provided by the service logic of the invoking SCF to the service logic of the responding SCF.

- securityParameter:

  This is an optional parameter that conveys security related information.

6.1.10.1.2 Invoking entity (controlling SCF)

6.1.10.1.2.1 Normal procedure

SCF precondition:

1) A relationship has been established between the two SCFs;

2) The SLPI has identified the need for sending service information from the controlling SCF;

3) The SCF FSM is in the state assisted mode (controlling SCF).

SCF postcondition:

1) SLPI execution continues;

2) The SCF FSM remains in the same state.

6.1.10.1.2.2 Error handling

Generic error handling for the operation related errors is described in clause 16 and the TCAP services which are used for reporting operation errors are described in clause 18.

6.1.10.1.3 Invoking entity (supporting SCF)

6.1.10.1.3.1 Normal procedure

SCF precondition:

1) A relationship has been established between the two SCFs;
2) The SLPI has identified the need for sending service information from the supporting SCF;
3) The SCF FSM is in the state assisting mode (supporting SCF).

SCF postcondition:

1) SLPI execution continues;
2) The SCF FSM remains in the same state.

6.1.10.1.3.2 Error handling

Generic error handling for the operation related errors is described in clause 16 and the TCAP services which are used for reporting operation errors are described in clause 18.

JMM These clause numbers do not exist. The reference is to be checked following PE.

6.1.10.1.4 Responding entity (supporting SCF)

6.1.10.1.4.1 Normal procedure

SCF precondition:

1) A relationship has been established between the two SCFs;
2) The SCF FSM is in the assisting mode (supporting SCF).

SCF postcondition:

1) SLPI execution continues;
2) The SCF FSM remains in the same state.

6.1.10.1.4.2 Error handling

Generic error handling for the operation related errors is described in clause 16 and the TCAP services which are used for reporting operation errors are described in clause 18.

JMM These clause numbers do not exist. The reference is to be checked following PE.

6.1.10.1.5 Responding entity (controlling SCF)

6.1.10.1.5.1 Normal procedure

SCF precondition:

1) A relationship has been established between the two SCFs;
2) The SCF FSM is in the assisted mode (controlling SCF).

SCF postcondition:

1) SLPI execution continues;

2) The SCF FSM remains in the same state.

6.1.10.1.5.2 Error handling

Generic error handling for the operation related errors is described in clause 16 and the TCAP services which are used for reporting operation errors are described in clause 18.

JMM These clause numbers do not exist. The reference is to be checked following PE.

6.2 Mapping of CTM messages/operations to INAP operations

Table 2 indicates the mapping of the CTM messages/operations to the INAP operation for every concerned interface, where the following abbreviations apply:

InitialAssDP: InitialAssociationDP;

InitiateAssoc: InitiateAssociation.

Table 2

	operation on INAP interfaces			
	CUSF/SSF-SCF		SCF-SCF	
CTM operation	invoke	result	invoke	result
cTMAccessRightsRequest	InitialAssDP	SendSTUI	n/a	n/a
cTMAccessRightsRequestSCF	n/a	n/a	TransferSTSI	TransferSTSI
cTMAccessRightsTerminate	InitiateAssoc	ReportUTSI	TransferSTSI	TransferSTSI
cTMLocationRegistration	InitialAssDP	SendSTUI	n/a	n/a
cTMLocationUpdateSCF	n/a	n/a	TransferSTSI	TransferSTSI
cTMLocationCancellation	InitiateAssoc	ReportUTSI	TransferSTSI	TransferSTSI
cTMProvideSecurityDataSCF	n/a	n/a	TransferSTSI	TransferSTSI
cTMNetworkAuthentication	ReportUTSI	SendSTUI	TransferSTSI	TransferSTSI
cTMTerminalAuthentication	InitiateAssoc/ SendSTUI (note1)	ReportUTSI	TransferSTSI (note2)	TransferSTSI
cTMKeyAllocate	SendSTUI	ReportUTSI	TransferSTSI	TransferSTSI
cTMCiphering	InitiateAssoc/ SendSTUI *	ReportUTSI	TransferSTSI (note2)	TransferSTSI
cTMOutgoingCallMobility ManagementInfo	InitialDP	n/a	n/a	n/a
cTMIncomingCallMobility ManagementInfo	SendSTUI	n/a	n/a	n/a
cTMProvideRoaming NumberSCF	n/a	n/a	TransferSTSI	TransferSTSI
ProvideSecurityData	tbc		TransferSTSI	TransferSTSI
cTMRestoreDataSCF	n/a	n/a	TransferSTSI	TransferSTSI
NOTE 1: This is dependent on whether a call unrelated dialogue is already open. NOTE 2: This corresponds to the case where the SCFmmSubscription initiates the terminal authentication and/or ciphering during the subscription registration procedures.				

7 ASN.1 specification for CTM application

-- CTM Mobility Management Application Module

```
CTM-appl-ops-args{ccitt(0) identified-organization(4) etsi(0) inDomain(1) in-Ctm(3) modules(1) cTM-
appl-ops-and-args(30) version1(0)}
DEFINITIONS IMPLICIT TAGS ::=
BEGIN
IMPORTS
    OPERATION, Code, ERROR
FROM Remote-Operations-Information-Objects ros-InformationObjects

    -- Operations
```

```
    cTMAccessRightsTerminate,
    cTMLocationCancellation,
    cTMTerminalAuthentication,
    cTMNetworkAuthentication,
    cTMCiphering,
    cTMKeyAllocate,

    -- Data Types
    AuthType,
    CipherKey,
    FixedIdentity,
    IPUI,
    IPEI,
    PortableCapabilities,
    PortableIdentity,
    Rand,
    Rs,
    Res,
    ServiceClass,
    -- Errors
    unspecified,
    portableIdentityUnknown
FROM CTM_Operations_and_Arguments{ccitt identified-organisation etsi (0) xxx operations-and-
errors(1) }
```

Editor's Note for ETSI: The above ObjectIdentifierValue "{ccitt identified-organisation etsi (0) xxx operations-and-errors(1) }" shall identify the MobilityManagement-Operations-and-Errors {ccitt identified-organisation etsi(0) xxx operations-and-errors(1)} of DE/SPS-05121 " Signalling application for the mobility management service on the alpha interface" part 1.

```
    CalledPartyNumber,
    CallingPartyNumber,
    ScfID
FROM IN-CS2-datatypes { ccitt(0) identified-organisation(4) etsi(0) inDomain(1) in-network(1)
CS-2(20) modules(0) in-cs2-datatypes (0) version1(0)};
cTMAccessRightsRequestSCF OPERATION ::= {
    ARGUMENT SEQUENCE {
        iPEI            [0] IPEI        OPTIONAL,
        cTMAuthType     [1] AuthType,
        cTMPortableCapabilities [2] PortableCapabilities,
    ...}
    RESULT SEQUENCE {
        iPUI            [0] IPUI,
        cTMFixedIdentity        [1] FixedIdentity,
        cTMServiceClass     [2] ServiceClass OPTIONAL,
    ...}
    ERRORS {pPortableIdentityUnknown |
        uUnspecified
}
    CODE opcode-CTMAccessRightsRequestSCF
    }

-- End of SCFAccessRightsRequest operation definition
cTMLocationUpdateSCF OPERATION ::= {
    ARGUMENT SEQUENCE {
        iPUI            [0] IPUI,
        sCFmmVisitedId      [1] ScfID,
    ..}
    RESULT SEQUENCE {
        cTMNumber       [0] CallingPartyNumber,
        cTMFixedIdentity    [1] FixedIdentity,
    ...}
    ERRORS {    cTMDataMissing |
            portableIdentityUnknown |
            unspecified |
            cTMRoamingNotAllowed}
    CODE opcode-CTMLocationUpdateSCF
    }
-- End of LocationUpdate operation definition

cTMProvideRoamingNumberSCF OPERATION ::= {
    ARGUMENT SEQUENCE {
        iPUI            [0] PortableIdentity,
    ...}
    RESULT SEQUENCE {
        roamingNumber       [0] CalledParty OPTIONAL,
        userStatus      [1] UserStatus,
    ...}
    ERRORS {cTMDataMissing |
            cTMFacilityNotSupported |
            cTMNoRoamingNumberAvailable |
```

```
            portableIdentityUnknown}
    CODE opcode-CTMProvideRoamingNumberSCF
    }
-- End of ProvideRoamingNumber operation definition

cTMProvideSecurityDataSCF OPERATION ::= {
    ARGUMENT SEQUENCE{
        iPUI            [0] IPUI,
    ...}
    RESULT SEQUENCE {
        securityDataSets        [2] SIZE (1.. maxSetsOfAuthenticationData) OF AuthenticationData,
    ...}
    ERRORS{
        portableIdentityUnknown |
        unspecified
    CODE opcode-CTMProvideSecurityDataSCF
    }
-- End of ProvideSecurityData operation definition
cTMRestoreDataSCF OPERATION ::= {
    ARGUMENT SEQUENCE{
        sCFmmHomeId     ScfID,
...}
    RETURN RESULT FALSE
    ERRORS FALSE
    CODE opcode-CTMRestoreDataSCF
}
-- End of SCF_RestoreData operation definition

-- Data Types

AuthenticationData      ::= SEQUENCE {
                    authType    AuthType,
                    rand        Rand            OPTIONAL,
                    rs      Rs          OPTIONAL,
                    ks      Ks          OPTIONAL,
                    res     Res         OPTIONAL,
                    cipherKey   CipherKey       OPTIONAL
}


Ks              ::= OCTETSTRING(SIZE(16))
UserStatus          ::= ENUMERATED{userUnknown(0), userReachable(1), userNotReachable(2), ... }
-- Error Types

cTMRoamingNotAllowed ERROR ::= {
    CODE    errcode-CTMRoamingNotAllowed
    }
cTMFacilityNotSupported ERROR ::= {
    CODE    errcode-CTMFacilityNotSupported
    }
cTMNoRoamingNumberAvailable ERROR ::= {
    CODE    errcode-CTMNoRoamingNumberAvailable
    }
cTMDataMissing ERROR ::= {
    CODE    errcode-CTMDataMissing
    }

-- CTM Mobility Management operation codes

opcode-CTMAccessRightsRequestSCF    Code ::= global : { cTM-SCF-SCF-application 1}
opcode-CTMLocationUpdateSCF         Code::= global : { cTM-SCF-SCF-application 2}
opcode-CTMProvideRoamingNumberSCF     Code::= global : { cTM-SCF-SCF-application 3}
opcode-CTMProvideSecurityDataSCF        Code::= global : { cTM-SCF-SCF-application 4}
opcode-CTMRestoreDataSCF        Code::= global : { cTM-SCF-SCF-application 5}

-- CTM Mobility Management error codes

errcode-CTMDataMissing          Code ::= global : { cTM-SCF-SCF-application 51}
errcode-CTMFacilityNotSupported     Code ::= global : { cTM-SCF-SCF-application 52}
errcode-CTMNoRoamingNumberAvailable    Code ::= global : { cTM-SCF-SCF-application 53}
errcode-CTMRoamingNotAllowed        Code ::= global : { cTM-SCF-SCF-application 54}

CTM-APPL-PARAMETERS-BOUND ::= CLASS
{
    &maxSetsOfAuthenticationData        INTEGER
    -- The maxSetsOfAuthenticationData shall not exceed the maximum sets allowed to be transferred
by the
    -- used TCAP.
}
WITH SYNTAX
```

```
{
    MAXIMUM-FOR-SETS-OF-AUTHENTICATION-DATA &maxSetsOfAuthenticationData
}
-- The following instance of the parameter bound is just an example
networkSpecificCTMBoundSet PARAMETERS-BOUND ::=
{
    MAXIMUM-FOR-SETS-OF-AUTHENTICATION-DATA 4   -- e.g. maximum for Bluebook TCAP
}

-- Object Identifiers

-- This module assigns object identifiers for CTM

cTM-SCF-SCF-application OBJECT IDENTIFIER ::=
    {ccitt(0) identified-organization(4) etsi(0) inDomain(1) in-Ctm(3) modules(1) cTM-
appl-ops_and_args(30) version1(0)}
id-CTM-ServiceFunctionIndicator OBJECT IDENTIFIER ::= { ccitt(0) identified-organisation(4)
etsi(0) inDomain(1) in-Ctm(3) interfaces(100) ctmAgreement(2)}.
-- This value shall be used for the USIServiceIndicator and the AgreementFeatureIndicator.
-- NOTE: Also operation codes and error codes as defined are object identifiers.
-- Timer values
```

The following value ranges do apply:
short: 1 - 10 seconds
medium: 1 - 60 seconds
long: 1 second - 30 minutes

-- Operation Timers

Ttable 3 lists all operation timers and the value range for each timer. The definitive value for each operation timer may be network specific and has to be defined by the network operator.

```
The default value for T-MM shall be 15 sec.
```

Table 3

DSS1+ defined operations	T-MM	medium
cTMAccessRightsRequestSCF	Tcarrs	medium
cTMLocationUpdateSCF	T_{clus}	medium
cTMProvideRoamingNumberSCF	T_{cprns}	medium
cTMProvideSecurityDataSCF	*Tcpsds*	long
cTMRestoreDataSCF	*Tcrds*	short

```
-- Application Timers
```
Table 4 lists all application specific timers and the value range for each timer. The definitive value for each operation timer may be network specific and has to be defined by the network operator.

Table 4

RoamingNumberTimer	Tcrn	medium

```
END
```

8 Interworking with DSS1

This clause describes the interworking with DSS1+.

8.1 Description

The following figure shows in a simplified manner the call related and the call unrelated signalling configurations to be considered in the present document.

In general the term SSP is used to consider both the call related as well as the call unrelated configuration.

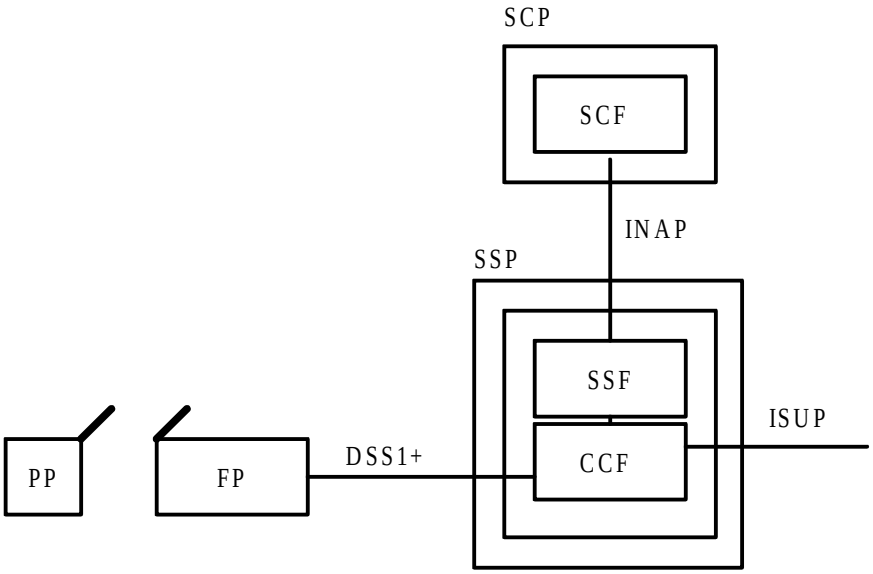
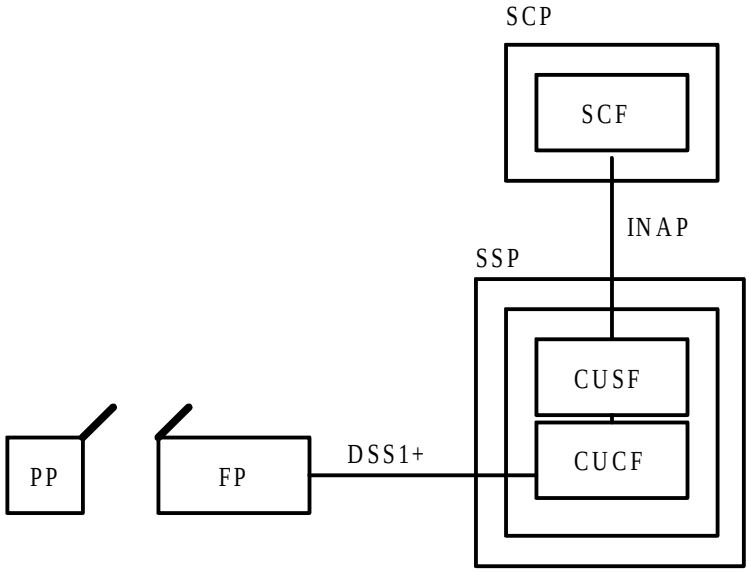


Figure 5: Signalling call related configuration



CUCF (Call Unrelated Control Function): The part of CCF necessary for the CUSF.

Figure 6: Signalling call unrelated configuration

8.1.1 Detection point processing

The normal call related and call unrelated detection points are applicable for CTM.

Of course, the triggering of the CTM service is service specific.

8.1.2 Receipt of DSS1+ messages

In general the procedures as described in EN 301 144-1 [3] (DSS1+) and EN 301 140-1 [19] (ETSI INAP CS2) are applicable. Specific aspects are defined in the following.

8.1.3 Receipt of INAP operations

In general the procedures as described in EN 301 144-1 [3] (DSS1+) and EN 301 140-1 [19] (ETSI INAP CS2) are applicable. Specific aspects are defined in the following.

8.1.4 Coding requirements

The CTM operations transferred in DSS1+ and INAP shall not exceed either the maximum size for the component part of the Facility information element or the maximum size of the octet string content of the uSIInformation. It has to be considered that the size of an INAP operation shall not exceed the maximum length capable of being transferred by TCAP.

8.2 Generic procedures

This subclause specifies the interworking on a procedural level.

The procedures as described in EN 301 144-1 (DSS1+) and EN 301 140-1 [19] (ETSI INAP CS2) are applicable. Those procedures define which messages and operations are applicable in which circumstances. The CTM specific use of IN procedures and operations flows are defined in the present document.

8.2.1 USI handling procedures

For CTM, the uSIServiceIndicator for USI transfer to be used shall be the following object identifier:

id-CTM-ServiceFunctionIndicator as defined in the CTM application module in this specification.

This subclause describes the USI related issues. Which message will be used to send this information is context dependent. The IN procedures as defined in clause 17 of en 301 140-1 [19] are to be taken into account.

8.2.1.1 GFT-control input and output for CTM

The GFT-control receives and provides respectively the value of the serviceFunction of the network facility extension in the Facility information element. This value shall be id-CTM-ServiceFunctionIndicator.

The sourceEntity received by the CUCF (user to SCP direction) will be stored as destinationEntity and will be sent later (SCP to user direction).

8.2.1.2 Receiving of DSS1+ operations / sending of UTSI information

UTSI information can be received by the SSP by the Facility information element in the following DSS1+ messages: SETUP or FACILITY.

UTSI information shall be sent to the SCP using one of the following operations for CTM as appropriate: InitialDP, InitialAssociationDP or ReportUTSI.

If and only if the SSP receives a Facility information element with a protocol profile set to "Networking extensions" and if the GFT-control has determined that the serviceFunction is to be provided locally, then the SSP shall send an appropriate IN operation to the SCP and the mapping of information shall be as follows:

- The uSIServiceIndicator of the INAP operation shall be set to the value of the serviceFunction received in the NetworkFacilityExtension received in the Facility information element.

- The octet string content of the uSIInformation shall be set to the sequence of octets which are building the "Service Components" part (octets 4 etc.) of the Facility information element.

- The other argument parameters shall be set as appropriate.

The following table summarizes the mapping in respect to the USI information.

Table 5: Mapping of DSS1+ operations to UTSI information

DSS1+			INAP	
Facility i.e.		→	**USI**	
Protocol profile	= Networking extensions			
Network Facility Extension				
serviceFunction	= id-CTM-ServiceFunctionIndicator		uSIServiceIndicator	= serviceFunction <same as received>
sourceEntity	= endTerminal			
sourceEntityAddress <opt.>				
destinationEntity				
destinationEntityAddress <opt.>				
Service Components	= components		uSIInformation	= Service Components <same as received>
NOTE: opt. means optional.				

8.2.1.3 Receiving of STUI information / sending of DSS1+ operations

STUI information can be received by the SSP by the following INAP operations: InitiateAssociation or SendSTUI operation.

STUI information will be sent to the DSS1 user using one of the following messages as appropriate: SETUP or FACILITY.

If and only if the SSP receives an IN operation including STUI information with the uSIServiceIndicator set to id-CTM-ServiceFunctionIndicator and if the SSP is in the state possible to send this information to the user then the SSP shall send an appropriate DSS1+ message including a Facility information element. The mapping of information shall be as follows:

- The protocol profile of the Facility information element shall be set to "networking extensions".
- The serviceFunction of the network facility extension in the Facility information element shall be set to the uSIServiceIndicator of the INAP operation.
- The octet content of the "service components" part of the Facility information element, i.e. "octets 4 etc.", shall be set to the service components received in the sequence of octets content of the uSIInformation.

Table 6 summarizes the mapping in respect to the USI information.

Table 6: Mapping of STUI information to DSS1+ operations

DSS1+			INAP	
Facility i.e.		←	**USI**	
Protocol profile	= Networking extensions			
Network Facility Extension				
serviceFunction	= uSIServiceIndicator <same as received>		uSIServiceIndicator	= id-CTM-ServiceFunctionIndicator
sourceEntity	<not applicable>			
sourceEntityAddress <opt.>	<not applicable>			
destinationEntity	= "endTerminal" or <provided by CUCF as previously received>			
destinationEntityAddress <opt.>	-			
Service Components	= <same as received>		uSIInformation	= Service Components
NOTE: opt. means optional.				

8.2.2 INAP component handling procedures

Not required for CTM.

8.3 Call related issues

In general the procedures as described in EN 301 144-1 (DSS1+) and EN 301 140-1 [19] (ETSI INAP CS2) are applicable.

8.3.1 Outgoing call

The user shall send a SETUP message which includes the CTMOutgoingCallMobilityManagement operation. The SSP initiates an InitialDP operation as defined in EN 301 140-1 [19]. The content of the Facility information element shall be mapped to the UTSI information according to subclause 8.2.1.2 "Receiving of DSS1+ operations / sending of UTSI information". The serviceKey shall be set as agreed for the service.

8.3.2 Emergency call

In principle the same interworking as for the normal call is applicable.

8.3.3 Incoming call

The SSP initiates an InitialDP operation as defined in EN 301 140-1 [19].

The SCP shall send SendSTUI operation including the CTMIncomingCallMobilityManagement operation. The CTMIncomingCallMobilityManagement operation shall be stored until the Connect operation is received. If the SSP receives a Connect operation the SSP shall send a SETUP message which includes the received CTMIncomingCallMobilityManagement operation. The Connect operation contains a destinationRoutingAddress set to the FT_address of the called user.

8.3.4 Call release

If the SCF sends a ReleaseCall operation then the call will be released immediately, that is without waiting for outstanding user responses.

8.4 Call unrelated issues

In the following specific interworking issues for the call unrelated case are mentioned.

As defined in EN 301 144-1 [3] the user or the network initiates a DSS1 GFP NCICS connection by the CTM application using the SETUP message. EN 301 144-1 states that "The first message establishing the NCICS connection (SETUP) shall always contain the CTM-id/PT-id and an invoke component indicating the requested operation.".

8.4.1 User initiated association

The SETUP message is received from the user and shall contain a Facility information element. The SSP shall send an InitialAssociationDP operation to the SCP.

The content of the Facility information element shall be mapped to the UTSI information according to subclause 8.2.1.2 "Receiving of DSS1+ operations / sending of UTSI information". The Network Facility Extension in the Facility information element shall be used, if at all, to route the Facility information to the SCF. It will not be included in the InitialAssociationDP operation for CTM.

The SETUP message and the InitialAssociationDP operation shall contain the information and use the mapping as indicated in the table 7. Further information may be included as appropriate.

The cUApplicationIndication parameter shall not be included.

Table 7: User initiated association

DSS1+			INAP	
SETUP		→	**InitialAssociationDP**	
			serviceKey	= <service specific>
			eventTypeBCUSM	= activation Received and Authorized
Bearer capability	= "Call independent Signalling Connection"		bearerCapability	= "Call independent Signalling Connection"
Calling party number	= FT_Address <optional>		callingPartyNumber	= FT_Address (note)
Called party number	<not used>			
Facility i.e.			USI (s. table 4)	
NOTE: In case, that the FT_Address is not provided by DSS1+ the default number of the access will be used as FT_Address in the InitialAssociationDP operation.				

8.4.2 Network initiated association

The InitiateAssociation operation is received from the SCP and shall contain STUI information. The SSP shall send a SETUP message to the user. The STUI information shall be mapped to the content of the Facility information element in the SETUP message according to subclause 8.2.1.3 "Receiving of STUI information / sending of DSS1+ operations".

The SETUP message and the InitiateAssociation operation shall contain the information and use the mapping as indicated in table 8. Further information may be included as appropriate.

Table 8: Network initiated association

DSS1+			INAP	
SETUP		←	**InitiateAssociation**	
Bearer capability	= "Call independent Signalling Connection"			
Calling party number	<not used>			
Called party number	= FT_Address <optional>		calledPartyNumber	= FT_Address
Facility i.e. (see table 5)			USI	

8.4.3 Information exchange during the association

The USI information exchange during the association shall use the procedures as defined in subclause 8.2.1 "USI handling procedures".

If the SSP receives USI information in the FACILITY message it will send those information to the SCP using the ReportUTSI operation.

If the SSP receives USI information in the SendSTUI operation it will send those information to the DSS1+ user using the FACILITY message.

8.4.4 Association release

The association release shall be as defined in en 301 140-1 [19].

The SCP shall use cause #31 "normal, unspecified" in the ReleaseAssociation operation. This cause shall be send to the user in the RELEASE message towards the user.

If the CUSF receives a ReleaseAssociation operation, then the DSS1+ association will be released immediately, i.e. without waiting for user responses.

Annex A (informative): Message Sequence Charts

This annex contains all MSCs and the related textual descriptions for the CTM mobility management procedures. They describe example cases for all possible procedures. The MSCs and their description are informative information only.

Clause A.1: Subscription registration

Clause A.2: Subscription deregistration

Clause A.3: Location registration in the SCFmmVisited

Clause A.4: Location update in the SCFmmHome

Clause A.5: LocationCancellation in the SCFmmVisited

Clause A.6: LocationCancellation in the FT

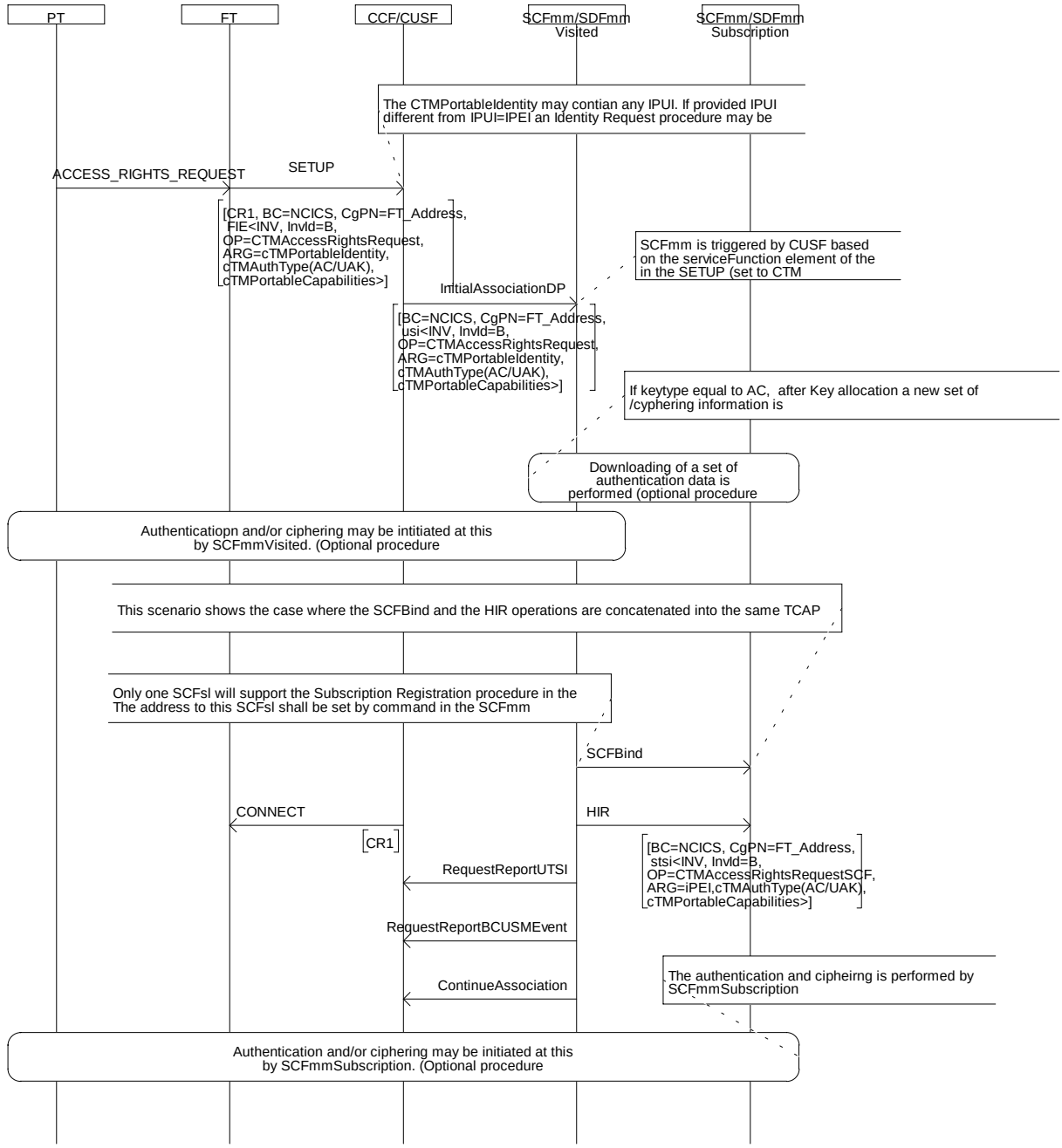
Clause A.7: Download of security data

Clause A.8: Restoration of location data in the SCFmmHome

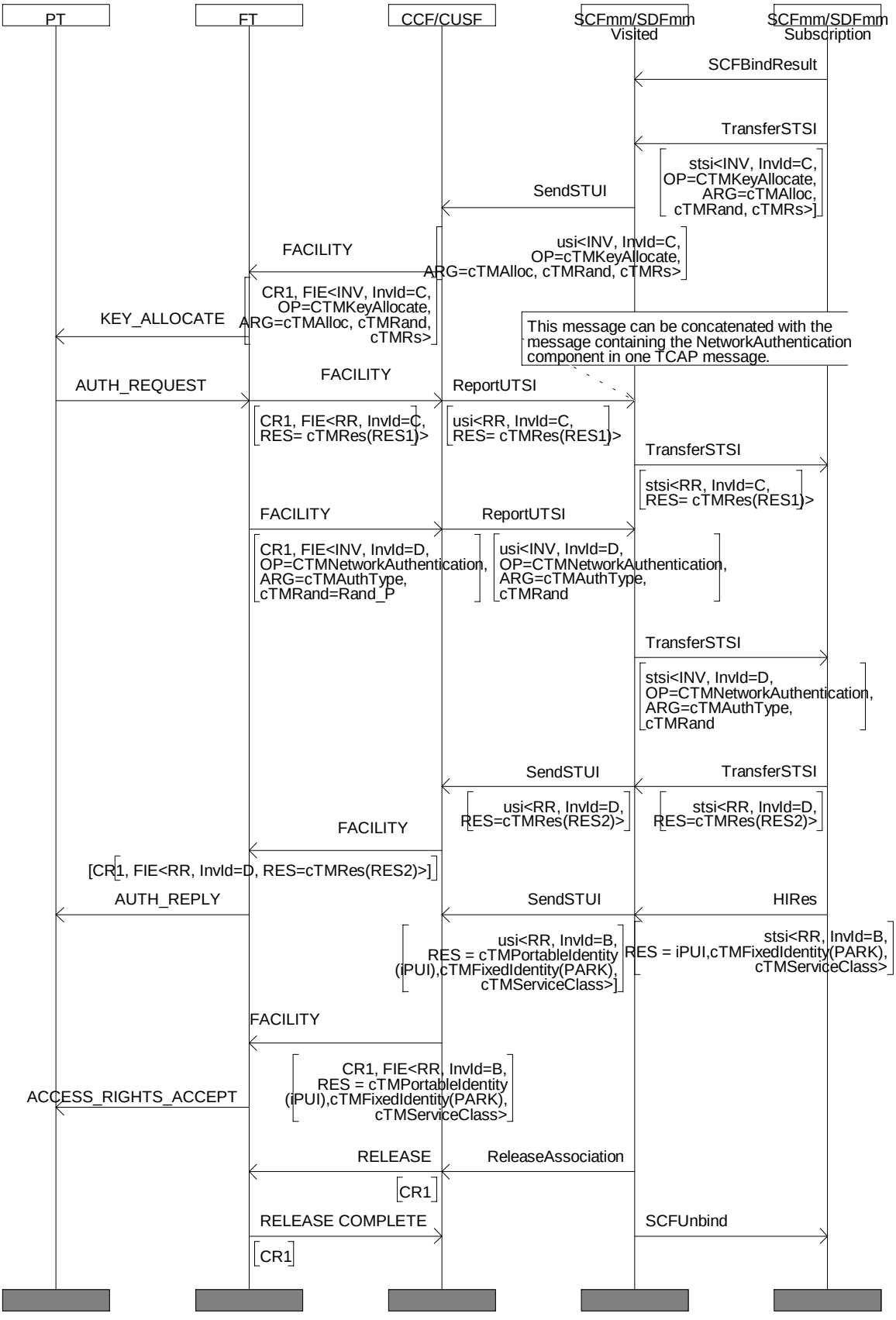
Clause A.9: Terminal authentication ciphering

A.1 Subscription registration

MSC SubscriptionRegistration (1/2)



MSC SubscriptionRegistration (2/2)



Normal operation

1) The PT initiates the subscription registration procedure when allowed by FT by sending an <cTMAccessRightsRequestSCF> invoke operation to the CUSF.

This invoke operation includes the following parameters:

- cTMPortableIdIdentity depends on the value previously assigned to the subscription record in the PT. If no value has been stored, the default IPUI-N=IPEI is used. If the IPEI is not provided an IdentityRequest procedure may be started by the SCFmmVisited.

- cTMAuthType containing the authentication algorithm (DSSA), the authentication key type (AC or UAK since the PT may contain either an AC or UAK before the subscription registration procedure is initiated) which is stored in the PT and the authentication key number.

- cTMPortableCapabilities to convey the cordless terminal capabilities such as tone capability, display capability, profile indicator and control codes.

2) The SCFmmVisited is triggered by the CUSF by means of an InitialAssociationDP operation based on the Service Function field of the NFE received in the DSS1 SETUP message with TDP "ActivationReceivedAndAuthorised" and as Triggering Criteria equal to CTM Application.

3) Prior to initiating the <cTMAccessRightsRequestSCF> in an SCFmmSubscription the SCFmmVisited may as a first possible option ask the SCFmmSubscription for the authentication data to be used for ciphering the key allocate procedure and to authenticate the CTM user. The SCFmmVisited then initiates authentication and/or ciphering. If authentication data is related to the AC Key a new set of authentication data is needed after the Key allocation procedure.

4) Since one or more dedicated SCFmmSubscription will support the Subscription Registration procedure in a network, one SCFmmSubscription shall be addressed. The SCFmmVisited shall send the <cTMAccessRightsRequestSCF> invoke operation to the SCFmmSubscription by means of the SSI mechanism.

5) The SCFmmSubscription may as another option initiate authentication and/or ciphering, first authenticate the CTM user and then cipher the Key Allocation procedure.

6) The SCFmmSubscription shall now initiate the key allocation procedure by translating the AC value stored in the SCFmmSubscription and the PT into the UAK. The key allocation procedure is initiated by the network to replace the Authentication Code (AC) by a more secure User Authentication Key (UAK).

7) The SCFmmSubscription sends the <cTMKeyAllocate > invoke operation to the SCFmmVisited including Alloc_type (in cTMAllocType), RAND_F (in cTMRand) and Rs (in cTMRs) via the SSI mechanism and this operation is transported to the PT via the USI mechanism of the CUSF.

8) If the content of the received parameters is acceptable for the PT, the PT responds with:

- a <cTMKeyAllocate> return result operation including the cTMRes parameter indicating the RES1 number to provide the terminal authentication result calculated by the cordless terminal, and;

- a <cTMNetworkAuthentication> invoke operation which includes the AuthType (in the cTMAuthenticationType) in addition to the RAND_P (in cTMRand) value. The <cTMNetworkAuthentication> operation is transported to the SCFmmSubscription using the CUSF and SCF-SCF interface.

These <cTMKeyAllocate> and <cTMNetworkAuthentication> components are transported in respectively sendSTUI and reportUTSI messages over the CUSF-SCF interfaces and via TransferSTSI over the SCF-SCF interface.

9) The SCFmmSubscription checks if the RES1 value corresponds with the XRES1 for the valid subscription.

The SCFmmSubscription on receiving the cTMNetworkAuthentication invoke component reuses the random session number cTMRs previously generated for key allocation and use the received random parameter. The SCFmmSubscription, having successfully performed the network authentication procedures, sends a cTMNetworkAuthentication return result to the SCFmmVisited via the SSI mechanism and this operation is transported to the PT via the USI mechanism of the CUSF. The cTMNetworkAuthentication return result includes as parameters the calculated result RES2.

10) The SCFmmSubscription performs the access rights procedure by sending a cTMAccessRightsRequestSCF return result component containing the following parameters:

- a cTMPortableIdentity which contains in the IPUI the PUT and PUN of the cordless terminal which requested the subscription registration;
- cTMFixedIdentity containing the type PARK, the AccessRightClass (ARC), AccessRightDetails (ARD) and the length of the identity (PLI). The pair IPUI and PARK provides the network with an unique cordless terminal identity;
- cTMServiceClass which is the service class related to the current active IPUI.

The cTMAccessRightsRequestSCF return result component is sent via the SSI mechanism on the SCFmmSubscription to SCFmmVisited interface and transported to the PT via the USI mechanism of the CUSF.

Exceptional procedure

11) If the PT is unable to perform the key-allocate procedure, a cTMKeyAllocate return error operation is sent via a ReportUTSI message via the CUSF and via a TransferSTSI over the SCF-SCF interface containing one of the following error values:

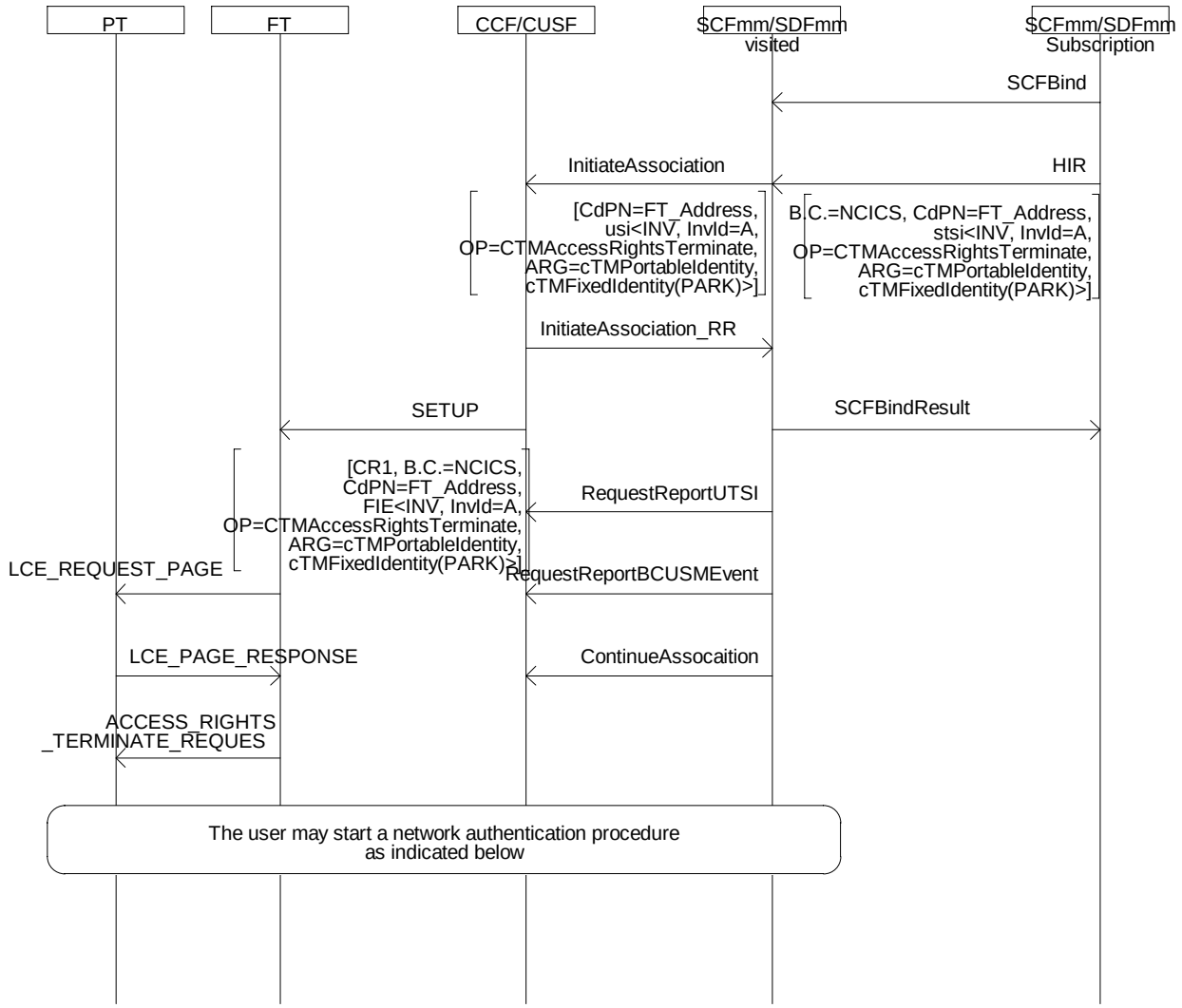
- terminal reject; with this value an optional additional parameter cTMTerminalRejectValue indicating the reject reason received on the air interface may be sent;
- pagingFailure (only if key allocate procedure in not embedded);
- radioConnectionFailure.

12) If the SCFmmSubscription detects on receipt of the cTMKeyAllocate return result operation that the RES1 value does not correspond with the XRES1 for the valid subscription, a cTMNetworkAuthentication return error component with value "networkRejected" and a cTMAccessRightsRequest return error component with value "networkRejected" shall be send via the SSI mechanism on the SCFmmSubscription to SCFmmVisited interface and these operations are transported to the PT via the USI mechanism of the CUSF.

13) If the SCFmmSubscription is unable to perform a network authentication, it sends a cTMNetworkAuthentication return error with value "networkRejected" and will subsequently forward a cTMAccessRightsRequest return error with value "networkRejected" via the SSI mechanism on the SCFmmSubscription to SCFmmVisited interface, and these operations are transported to the PT via the USI mechanism of the CUSF.

A.2 Subscription deregistration

MSC SubscriptionDeregistration (1/2)



Normal operation

1) This procedure is initiated by the network to cancel a subscription. At the end of the procedure execution, the subscriber data will be erased in the PT.

   To perform the subscription deregistration procedure, the SCFmmHome sends a cTMAccessRightsTerminate invoke component to the SCFmmVisited including in cTMPortableIdentity the IPUI and the PARK (in cTMFixedIdentity).

2) The SCFmmVisited forwards the cTMAccessRightsTerminate invoke component to the CUSF using the USI mechanism.

3) The CUSF sends the cTMAccessRightsTerminate invoke component to the SCUAF. The SCUAF then performs paging to the PT. The PT may start a Network Authentication procedure.

4) To perform a network authentication, the SCUAF sends a cTMNetworkAuthentication invoke component to the CUSF using the USI mechanism by means of a ReportUTSI message. The cTMNetworkAuthentication invoke component contains the cTMAuthenticationType, the cTMRand and optionally the cTMPortableIdentity.

5) The CUSF forwards the cTMNetworkAuthentication invoke component to the SCFmmVisited using the USI mechanism by means of a ReportUTSI message.

6) The SCFmmVisited forwards the cTMNetworkAuthentication invoke component to the SCFmmHome using the SSI mechanism by means of a TransferSTSI message.

7) The SCFmmHome, on receiving the cTMNetworkAuthentication invoke component, performs authentication and sends a cTMNetworkAuthentication return result component to the SCFmmVisited containing the calculated result RES2 (in cTMRes) and the random session number the network has used to calculate the RES2 value (in cTMRs). The SCFmmHome forwards the cTMNetworkAuthentication return result component to the SCFmmVisited using the SSI mechanism by means of a TransferSTSI message.

8) The SCFmmVisited forwards the cTMNetworkAuthentication return result component to the CUSF using the USI mechanism by means of a SendSTUI message.

9) The CUSF forwards the cTMNetworkAuthentication return result component to the SCUAF.

10) If the authentication of the network is successful, the user deletes PARK and IPUI and sends an cTMAccessRightsTerminate return result component to the SCUAF. The SCUAF forwards the cTMAccessRightsTerminate return result component towards the CUSF.

11) The CUSF forwards the cTMAccessRightsTerminate return result component to the SCFmmVisited using the USI mechanism by means of a ReportUTSI message.

Exceptional procedure

12) The SCFmmVisited forwards the cTMAccessRightsTerminate return result component to the SCFmmHome using the SSI mechanism by means of a TransferSTSI message.

13) If the PT is unable to perform the access rights terminate procedure, an cTMAccessRightsTerminate return error operation is sent via a ReportUTSI message via the CUSF and via TransferSTSI over the SCF-SCF interface containing one of the following error values:

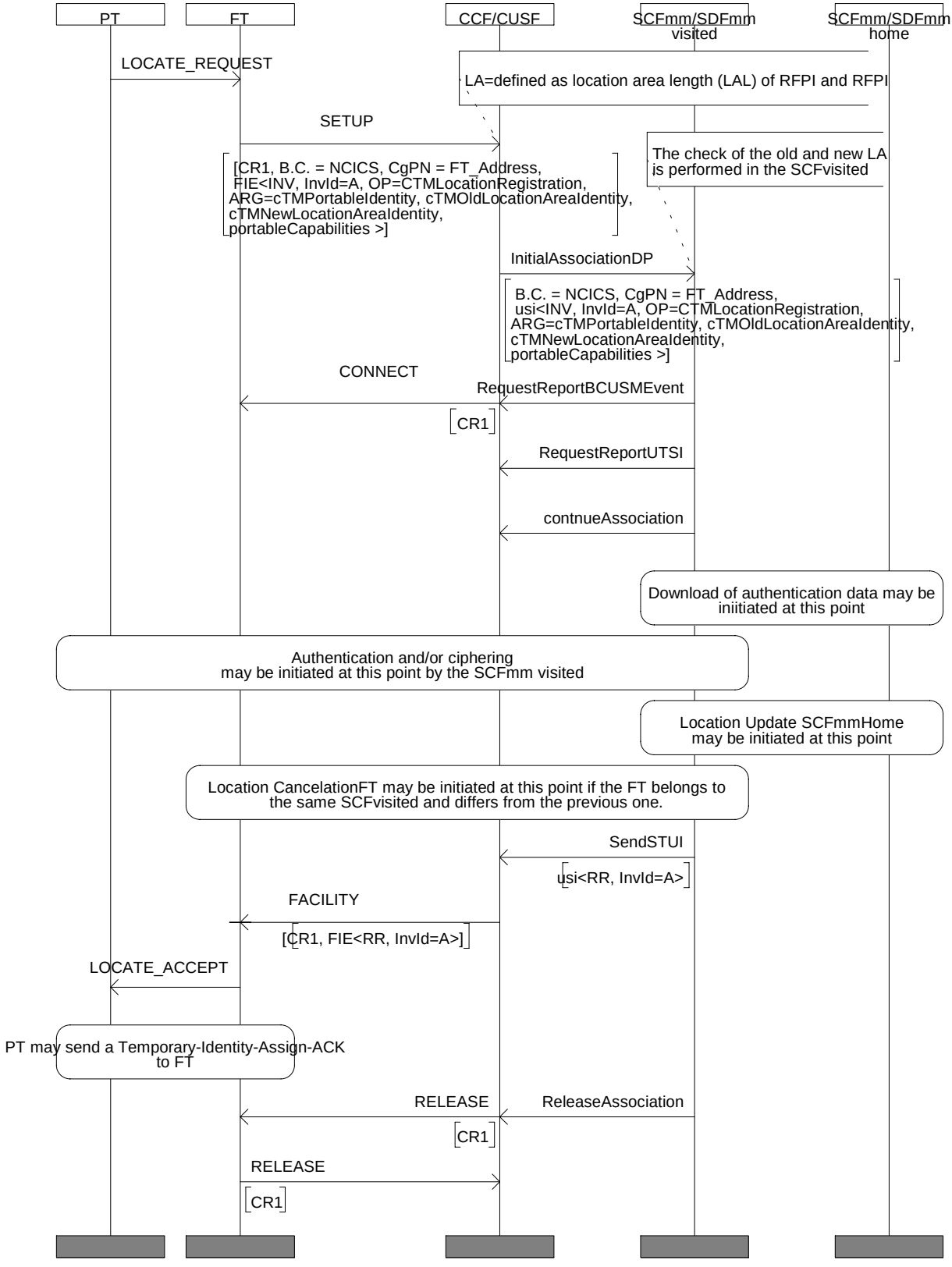
- terminalRejected (with this value an optional parameter cTMTerminalRejectValue indicating the reject reason received on the air interface may be sent);
- pagingFailure;
- radioConnectionFailure.

The SCFmmHome takes action as appropriate.

14) If the SCFmmHome is unable to perform a network authentication, the SCFmmHome sends a cTMcTMNetworkAuthentication return error with the value "networkRejected".

A.3 Location registration in the SCFmmVisited

MSC LocationRegistration



Normal operation

1) The cTMLocationRegistration invoke component is sent from the FT via the CUSF to the SCFmmVisited using the USI mechanism; this component contains the following parameters:

   - cTMPortableIdentity indicating the IPUI (identity of the portable terminal);
   - cTMOldLocationAreaIdentity and cTMNewLocationAreaIdentity containing the old and new LA (where LA is defined as the LAL of RFPI and the RFPI);
   - cTMPortableCapabilities indicating the characteristics of the cordless terminal.

NOTE 1: The CUSF provides the CUSFID as Calling Party Address in the TC-BEGIN message.

NOTE 2: The FT_Address is provided in the CallingPartyNumber of the CUSF InitialAssociationDP operation.

2) The CUSF triggers the SCFmmVisited based on the Service Function field of the NFE received in the DSS1 SETUP message with TDP "ActivationReceivedAndAuthorised" and as Triggering Criteria equal to CTM Application.

3) The SCFmmVisited may at this point download authentication data from the SCFmmHome e.g. if not yet registered.

4) The SCFmmVisited may perform authentication and/or ciphering procedure.

5) If PT is already registered in the SCFmmVisited:

   - the SCFmmVisited shall check the "Active LA" and "Active FT address".

6) If the "Active FT Address" differs from the received "FT Address" the SCFmmVisited shall initiate a cTMLocation Cancellation_FT (see relevant MSC).

7) If the "Active LA" differs from the "Old LA" received the SCFmmVisited shall initiate a location update SCFmmHome (see relevant MSC).

8) If data restoration procedure for the addressable SCFmmHome is ongoing, the SCFmmVisited shall initiate a location update SCFmmHome(see relevant MSC).

9) If the PT is not yet registered in the SCFmmVisited, the SCFmmVisited shall initiate a location update SCFmmHome (see relevant MSC).

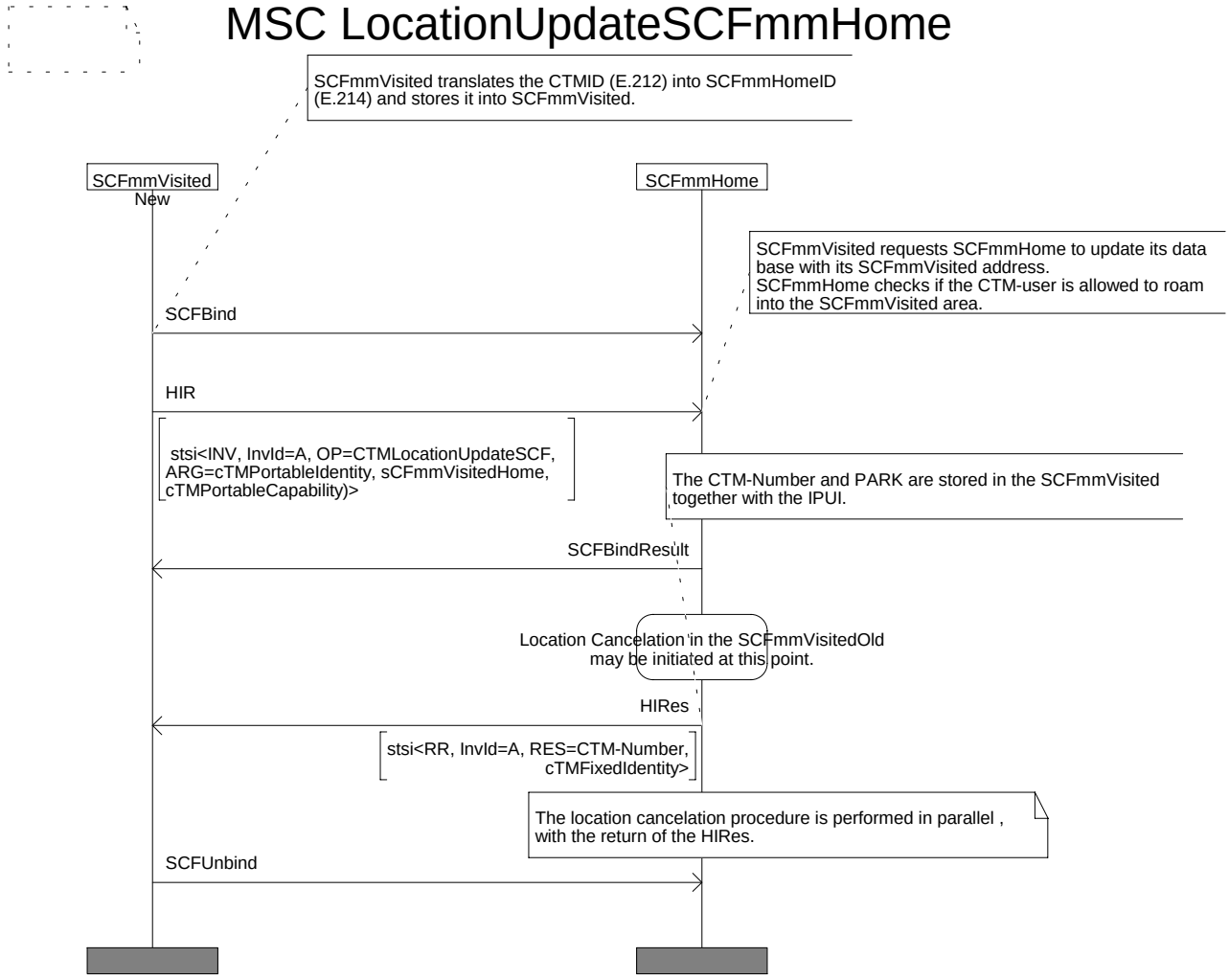
10) If the SCFmmVisited / SCFmmHome has successfully performed the location registration/update, the SCFmmVisited sends via a CUSF SendSTUI operation a cTMLocationRegistration return result component to the PT without parameters. The SCFmmVisited stores the "New LA" as "Active LA" and the received FT address as the Active FT Address.

Exceptional procedure

11) If the terminal authentication fails, or the location update SCFmmHome fails than the SCFmmVisited sends a LocationRegistration return error component via an SendSTUI operation to the CUSF with the value "NetworkRejected".

12) If during the location registration procedure a release association is received from the CTM user or the FT, this event may be reported to the SCFmmVisited by arming the EDP-R "AssociationReleaseRequested" BCUSM Event. The arming is performed by means of a requestReportBCUSMEvent operation and the reporting is performed by means of the eventReportBCUSM operation.

A.4 Location update in the SCFmmHome



Normal operation

1) The SCFmmVisited translates the PUN part of the cTMPortableIdentity (E.212 number) into a "Mobile global title" (E.214 number) and stores it as SCFmmHomeId. The SCFmmHomeId will be used later during the outgoing call procedure in order that the SCFmmVisited can initiate a subsequent triggering of the SCFmmHome for checking the outgoing CTM supplementary services. The SCFmmVisited shall send a cTMLocationUpdateSCF invoke operation using the SSI mechanism of the SCF-SCF interface containing its SCFmmVisitedId together with the associated IPUI.

2) The SCFmmHome shall check whether the CTM user is allowed to roam into the SCFmmVisited area; if so a cTMLocationUpdateSCF return result is sent with the cTMNumber and the type PARK.

3) The SCFmmHome shall check whether the CTM user was previously registered in another SCFmmVisited. If registered, the SCFmmHome initiates a location cancellation in SCFmmVisited procedure.

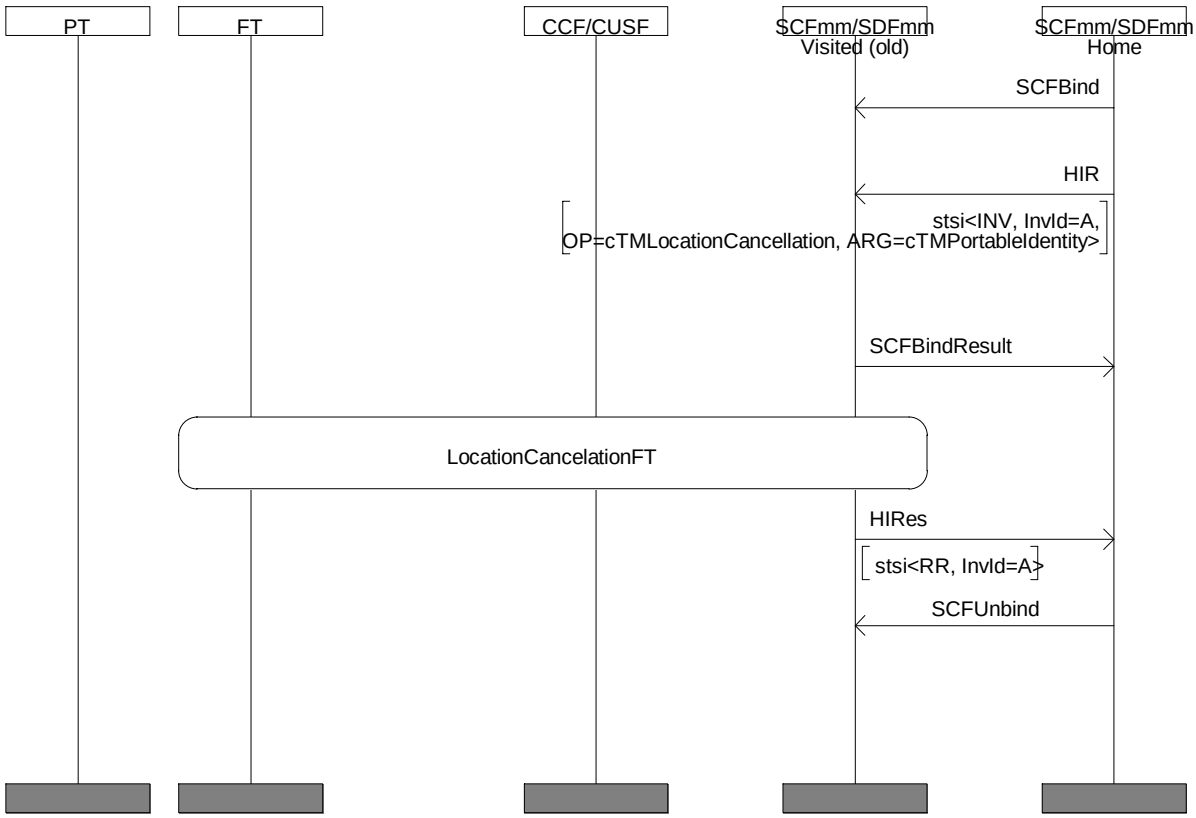
Exceptional procedure

4) If the CTM user is not allowed to roam into the SCFmmVisited area, on receipt of an cTMLocationUpdateSCF invoke operation the SCFmmHome sends a cTMLocationUpdateSCF return error operation to the SCFmmVisited using the SSI mechanism of the SCF-SCF interface.

5) If during the location registration procedure a release association is received from the CTM user or the FT, this event may be reported to the SCFmmVisited by arming the EDP-R "AssociationReleaseRequested" BCUSM Event. The arming is performed by means of an requestReportBCUSMEvent operation and the reporting is performed by means of the eventReportBCUSM operation.

A.5 LocationCancellation in the SCFmmVisited

MSC LocationCancellationSCFmmVisited

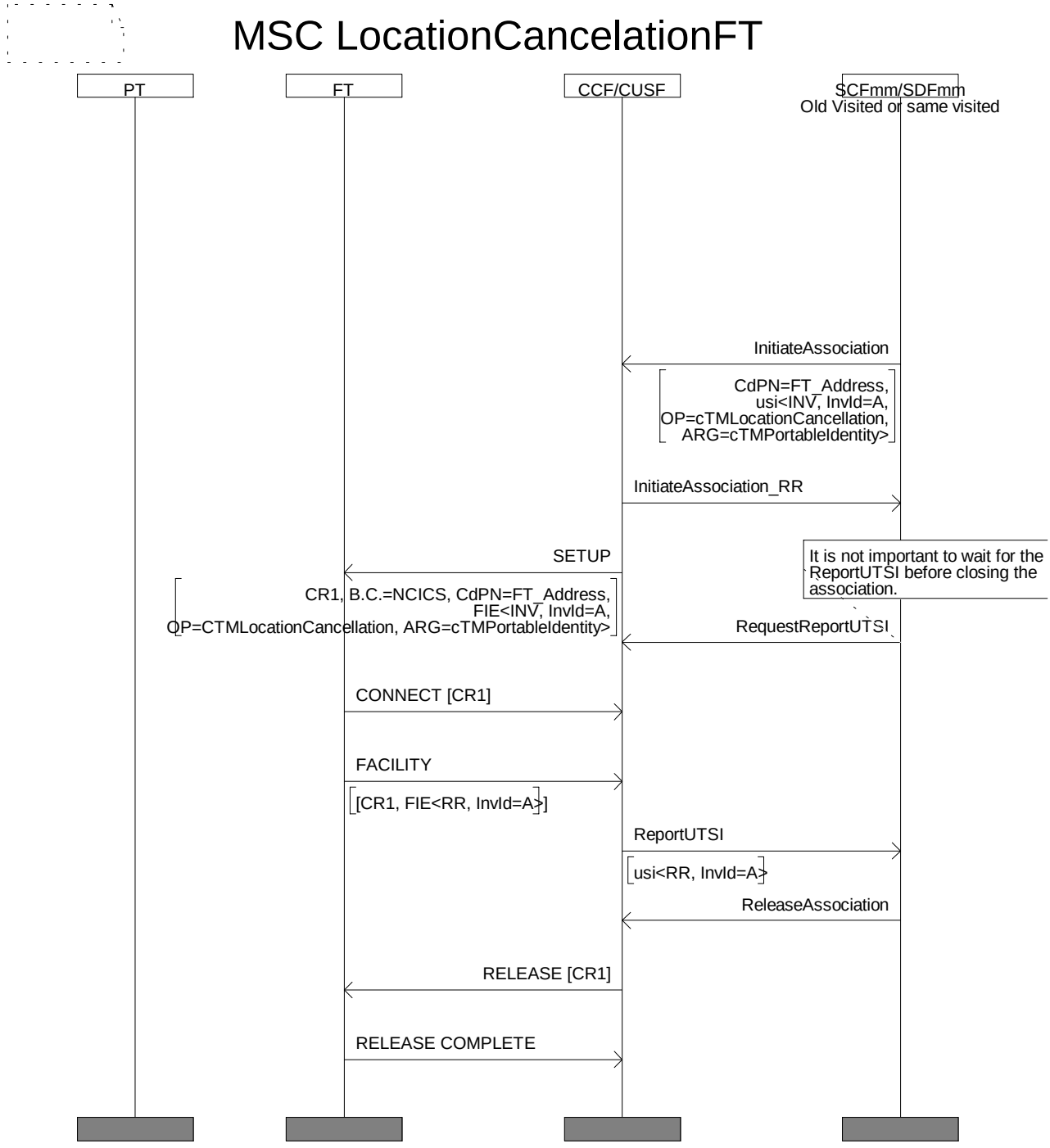


Normal operation

1) The SCFmmHome sends a cTMLocationCancellation invoke component using the SSI mechanism of the SCF-SCF interface containing as parameter the cTMPortableIdentity.

2) The SCFmmVisited removes, on receipt of the LocationCancellation invoke component, the cTMPortableIdentity and sends a cTMLocationCancellation return result to the SCFmmHome.

A.6 LocationCancellation in the FT



Normal operation

1) The location cancellation procedure is initiated by the SCFmmVisited towards the old FT to delete all data related to a Cordless Terminal e.g. because the terminal has moved to another location area.

   The SCFmmVisited sends a cTMLocationCancellation invoke component using the CUSF USI mechanism. The cTMLocationCancellation component contains the cTMPortableIdentity indicating the IPUI. The CUSF forwards the cTMLocationCancellation invoke component to the FT where the cTMPortableIdentity was previously registered.

2) The SCUAF removes the cTMPortableIdentity from its data base and responds by sending a cTMLocationCancellation return result component without parameters. This component is send via the CUSF to the SCFmmVisited using the USI mechanism.

Exceptional procedure

3) If during the location cancellation procedure a release association is received from the CTM user or the FT, this event may be reported to the SCFmmVisited by arming the EDP-R "AssociationReleaseRequested" BCUSM Event. The arming is performed by means of an requestReportBCUSMEvent operation and the reporting is performed by means of the eventReportBCUSM operation.

4) If the FT is unable to act on the cTMLocationCancellation invoke component, the FT sends a cTMLocationCancellation return error component via the CUSF to the SCFmmVisited using the USI mechanism. The component contains the portableIdentityUnknown error value if the location cancellation is not successfully performed due to the fact that the cTMPortableIdentity to be cancelled is not present.

A.7 Download of security data

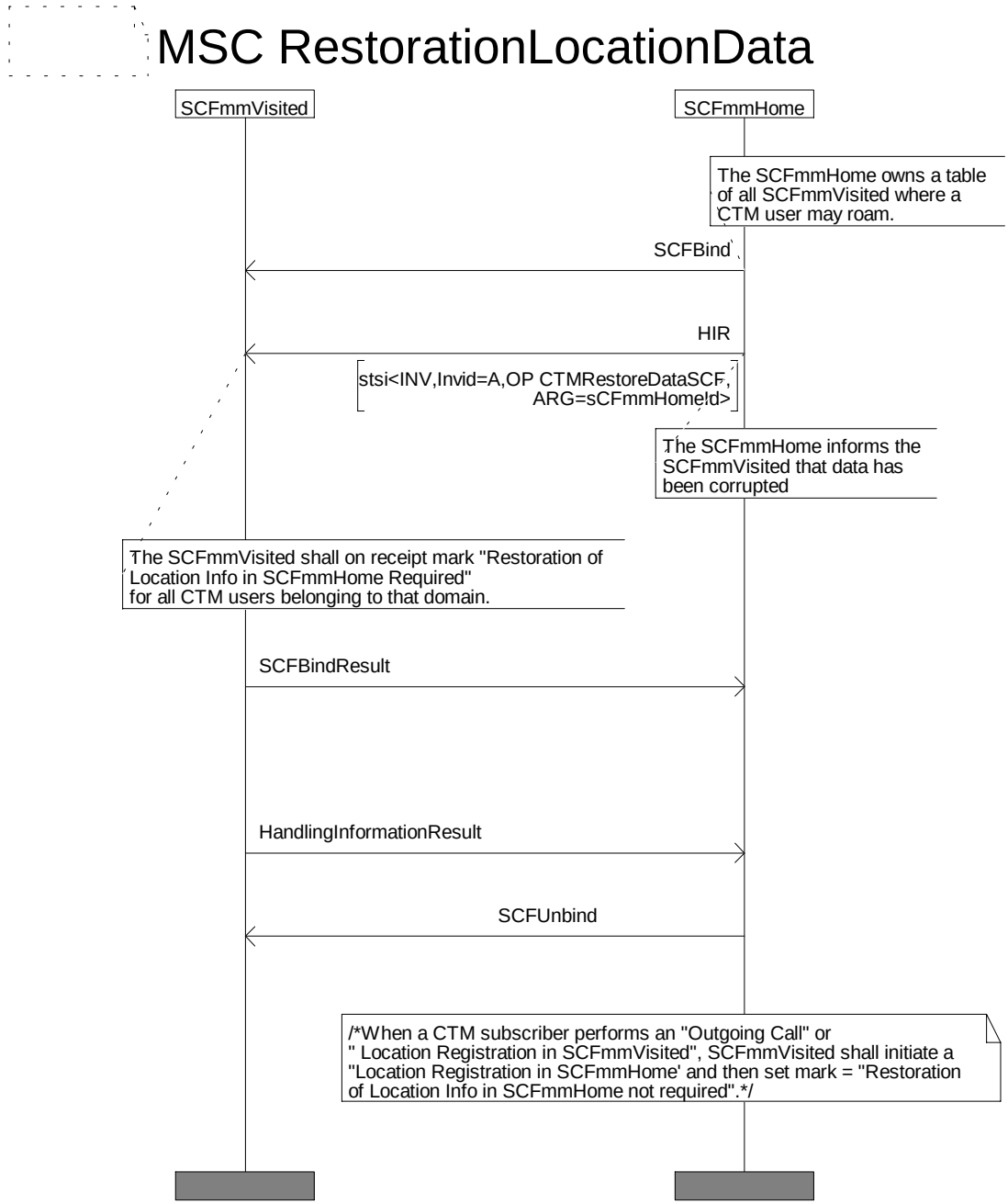
MSC DownloadSecurityData

Normal operation

1) The SCFmmVisited requests authentication and/or ciphering data from the SCFmmHome by sending a cTMProvideSecurityDataSCF invoke component using the SSI mechanism of the SCF_SCF interface. This component contains the IPUI.

2) The SCFmmHome, on receipt of the cTMProvideSecurityDataSCF invoke component, generates a number of sets of authentication and/or ciphering data and sends to the SCFmmVisited a cTMProvideSecurityDataSCF return result component using the SSI mechanism of the SCF_SCF interface. This component contains the sets of Authentication Data (RAND, RS, XRES1, DCK, cTMAuthType) that have been generated.

A.8 Restoration of location data in the SCFmmHome



Normal operation

1) The SCFmmHome has a table of all the SCFmmVisited addresses where the CTM subscribers may roam. If the location data gets corrupted in the SCFmmHome (e.g. after restart), the SCFmmHome sends a message to all

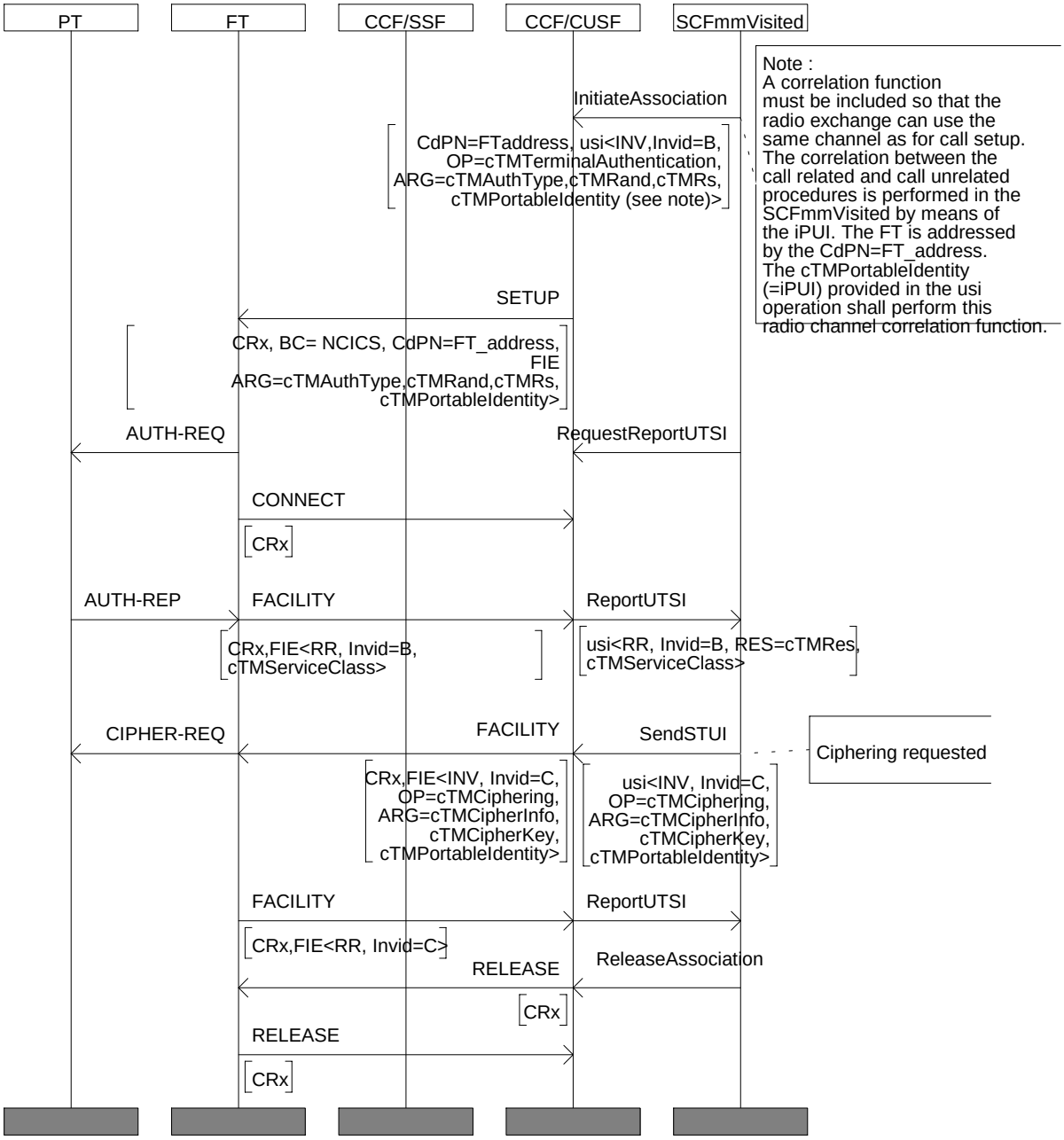
SCFmmVisited by means of a cTMRestoreDataSCF invoke component using the SSI mechanism over the SCF-SCF interface. This invoke component indicates to the addressed SCFmmVisited that the restoration of location data in the SCFmmHome shall take place and shall have the SCFmmHomeID parameter.

2) The SCFmmVisited, on receipt of the cTMRestoreDataSCF invoke component, marks <restoration of location info in SCFmmHome required> for all subscribers belonging to that SCFmmHome. To find all users belonging to the SCFmmHome, the invoked SCFmmVisited compares the SCFmmHomeID stored at the location registration in the SCFmmHome procedure with the one received in the restoration of location data procedure.

3) When the CTM subscriber performs an "outgoing call" procedure or a "location registration SCFmmVisited" procedure, the SCFmmVisited initiates a "location update SCFmmHome" procedure and sets mark <restoration of location info in SCFmmHome not required>. This option implies that CTM users will not be reachable until either they initiate an outgoing call or perform a location registration.

A.9 Terminal authentication ciphering

MSC TerminalAuthenticationCiphering



Normal operation

1) The SCFmmVisited may initiate an Terminal Authentication by sending a cTMTerminalAuthentication invoke component using the USI mechanism on CUSF-SCF interface. If a call-unrelated dialogue is already open, the same dialogue will be used for the Terminal Authentication, and the invoke component will be sent via a sendSTUI operation.

If a call-unrelated dialogue is not open, a call-unrelated dialogue will be opened, and the invoke component will be sent via the initiateAssociation operation.

The cTMTerminalAuthentication component shall contain the following information:

- cTMAuthType, indicating the authentication key type (AC and UAK), the authentication algorithm, and the authentication key number related to the iPUI to use for calculation of the authentication result;
- cTMRand, indicating the RAND number, which is used for calculation of the authentication result;
- cTMRs, indicating the Rs number, which is used for calculation of the authentication result;
- cTMPortableIdentity, identifying the cordless terminal.

2) The result of the terminal authentication (cTMRes) will be sent in the cTMTerminalAuthentication return result component. Optionally the service class (cTMServiceClass) may be included in the return result.

3) Independently from the Terminal Authentication procedure (for example, old cipher key can be used), the SCFmmVisited may initiate a ciphering procedure by sending a cTMCiphering invoke component using the USI mechanism on CUSF-SCF interface. If a call-unrelated dialogue is already open, the same dialogue will be used for the ciphering, and the invoke component will be sent via a sendSTUI operation.

If a call-unrelated dialogue is not open, a call-unrelated dialogue will be opened, and the invoke component will be sent via the initiateAssociation operation.

The cTMCiphering component contains the following information:

- cTMCipherInfo, indicating the cipher key type, cipher key number and the ciphering algorithm, related to the iPUI;
- cTMCipherKey, indicating the numeric value of the ciphering key to be used by the FP;
- cTMPortableIdentity (optionally), indicating the IPUI.

4) As soon as the ciphering is completed, the FT will initiate a cTMCiphering return result component without any parameter.

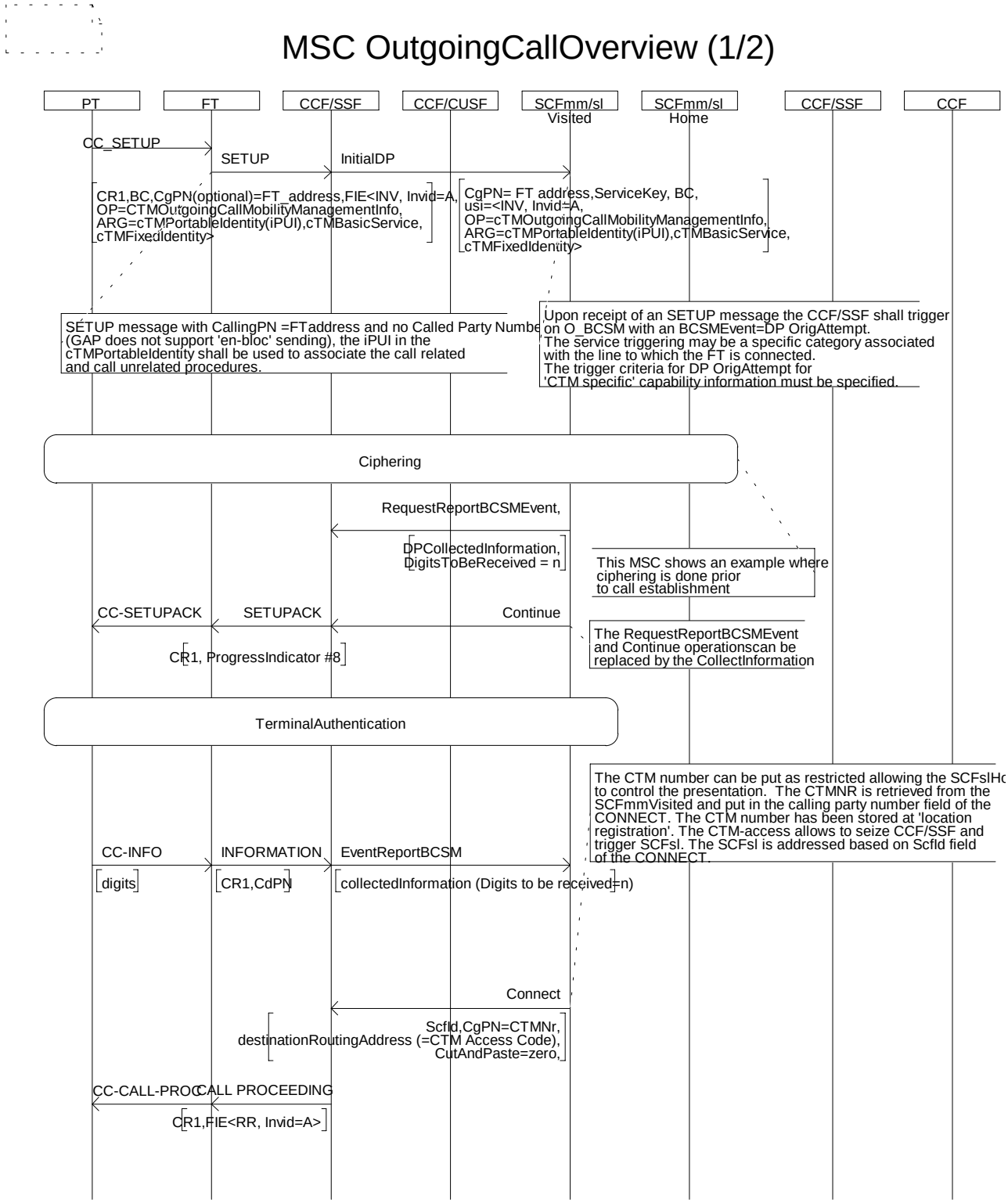
Annex B (informative): Call control Message Sequence Charts

This annex contains all MSCs and the related textual descriptions for the CTM Call Control procedures. They describe example cases for all possible procedures. The MSCs and their description are informative only.

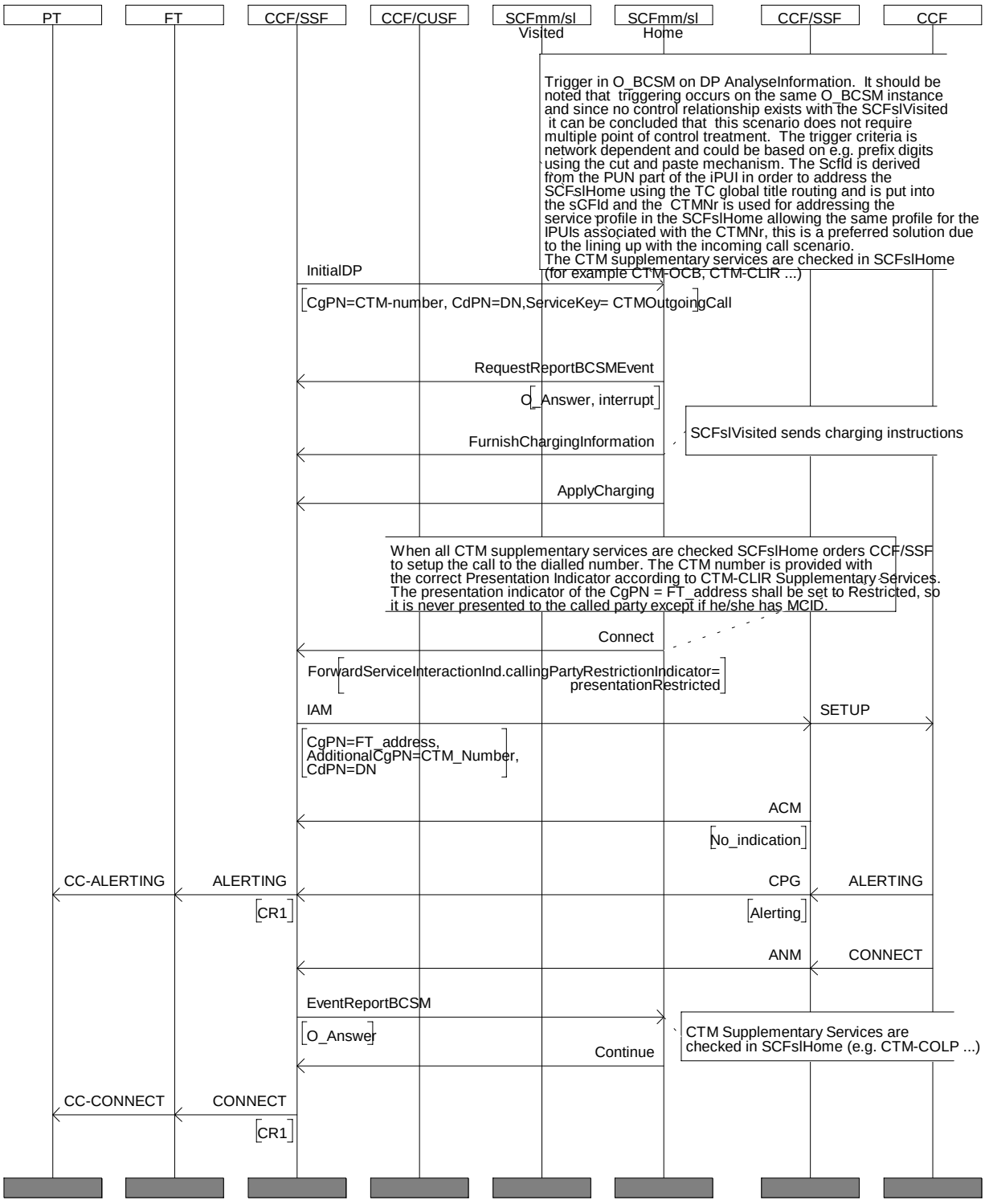
Clause B.1 Outgoing call

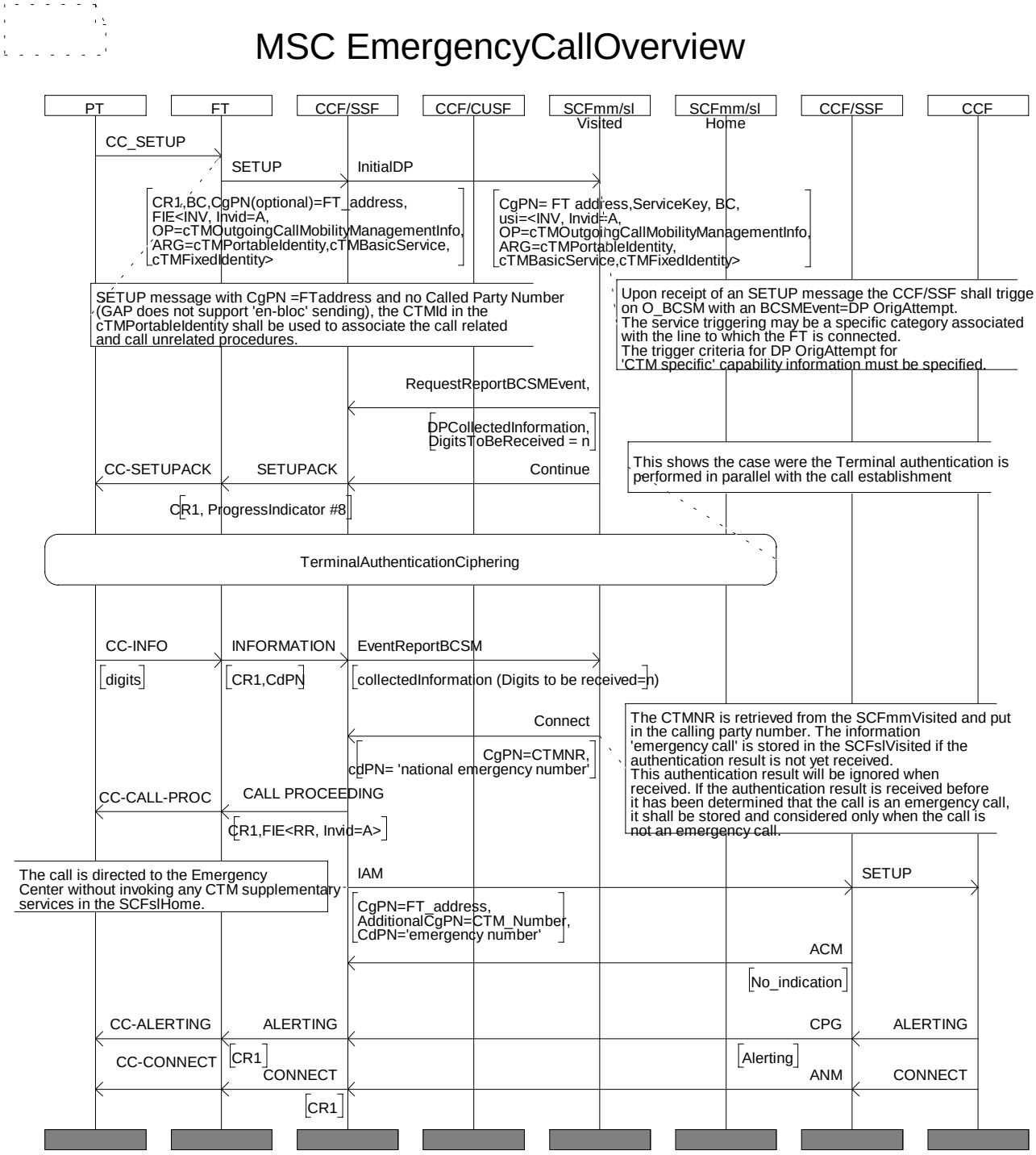
Clause B.2 Incoming call

B.1 Outgoing call



MSC OutgoingCallOverview (2/2)





Normal operation

1) When the CTM user initiates a call, the FT may include in the SETUP message the FT Address as Calling Party Number, the cTMOutgoingCallMobilityManagementInfo invoke component and no Called Party Number since the GAP profile does not support en-bloc sending. The cTMOutgoingCallMobilityManagementInfo shall contain the following parameters:

   - cTMPortableIdentity: indicating the IPUI (PUT and PUN), which shall be used to associate the call related and call unrelated procedures so that the same radio channel is used;

   - cTMBasicService indicating normal call setup;

   - cTMFixedIdentity indicating the PARK of the cordless terminal.

The call is routed to the CCF/SSF.

2) The SSF triggers the SCFslVisited on the O_BCSM in DP Originating_Attempt_Authorised with the trigger criteria based on the Service Function (part of the Facility information element of the MobilitiyManagementInfo) equal to CTM application. A InitialDP operation is sent to the SCFslVisited transferring the cTMOutgoingCallMobilityManagementInfo component via the USI mechanism.

3) The SCFslVisited may initiate terminal authentication and/or ciphering via the CUSF mechanism. The service script in the SCFslvisited decides whether authentication and ciphering is to be performed in parallel or in sequence with the call establishment or ciphering is performed prior to call establishment. The correlation between the call related procedures (normal call setup) and the call unrelated procedures (authentication, ciphering) isperformed via the IPUI (PUN). The authentication result is considered when it is determined that the call is a normal call.

4) The SCFslVisited requests, by means of e.g. a RequestReportBCSMEvent (DPCollectedInformation, DigitsToBeReceived = n) operation or a CollectInformation operation, the appropriate number of digits in order to be able to identify that the call is an emergency call. If the SCFmmVisited has decided that the ciphering is to be performed in sequence with the call establishment, the sending of SETUPACK message over the air interface and the alpha interface is coupled with the receipt of an appropriate INAP operation (e.g. CollectInformation) and/or associated ServiceInteractionIndicatorsTwo element of the continue.

5) When the requested digits are received in the CCF/SSF, an EventReportBCSM operation is sent to the SCFslVisited containing in the calledPartyNumber of the collectedInfoSpecificInfo the requested digits. The SCFslVisited checks the dialled digits with regard to the standard emergency number (in Europe, "112").

6) If the digits match, then the SCFslVisited orders the CCF/SSF to route the call towards the emergency centre by sending a connect operation. The result of the authentication procedure SCFmmVisited is ignored and the check on CTM supplementary services (for example CTM-OCB) in the SCFslHome is not performed. The SCFslVisited provides in the connect operation the CTM-Number, if available, as CallingPartyNumber (this CTM number has been stored previously during location registration). The FT address (E.164) and the CTM-Number (E.164) are sent in the IAM message towards the emergency centre as respectively calling party number and as generic number (additional calling party number).

7) If the digits do not match and the terminal authentication check fails then the call is released (see exceptional procedures) by SCFmmVisited through SCFslVisited.

8) If the digits do not match and the terminal authentication check does not fail, the SCFslVisited will order to the CCF/SSF a subsequent triggering of the SCFslHome.

The SCFslVisited includes the following parameters in the Connect operation sent to the SSF:

- the SCFmmHomeIdd which was previously stored during the location registration and place it in the SCFId parameter;

- the CTM-number of the calling CTM user in the CallingPartyNumber with the Presentation Indicator set to restricted;

- the "CTM access code" is added before the dialled digits in the SSF using the CutAndPaste parameter (CalledPartyNumber = "CTM access code + DN") in order to seize CCF/SSF and trigger the SCFslHome. In order to perform the above, the SCFslVisited puts the "CTM access code" into the destinationRoutingAddress parameter and sets the cutAndPaste parameter to zero in the connect operation. The SSF now pastes the remaining dialled digits received in overlap from the CTM calling user on to the end of the "CTM access code" digits received in the destinationRoutingAddress parameter from the SCFslVisited. The "CTM access code" ise deleted from the CalledPartyNumber by the CCF/SSF after triggering of the SCFslHome.

9) The CCF/SSF triggers in O_BCSM on DP AnalysedInformation based on the trigger criterion "CTM access code", previously received in the calledPartyNumber of the Connect operation from the SCFslVisited. The O_BCSM is re-triggered on the same instance but another service key is indicated in the InitialDP. Since no control relationship exists with the SCF it can be concluded **that this scenario does not require multiple point of control**. A monitor relationship will be maintained for the case where authentication is performed in parallel so that the SCFmmVisited can send a releaseCall operation if the authentication fails.

   The SCFId parameter of the Connect operation is used to address the SCFslHome while the CTM-number is used to access the CTM user’s Service Profile in the SDFslHome. This number is passed as AdditionalCallingPartyNumber in the InitialDP operation to the SCFslHome.

10) The SCFslHome checks the Service Profile of the CTM-user in order to invoke the CTM supplementary service (for example, the presentationIndicator of the FT_address will be set to restricted by the SCFslHome, so that it is not presented to the called user; the Presentation Indicator of the CTM-number is set according to the CTM-CLIR service) and sends a Connect operation to the CCF/SSF in order to establish the call towards the dialled number (DN). The CCF/SSF sets up the call towards the DN by sending an IAM message, including the FT_address as CallingPartyNumber and the CTM number as AdditionalCallingPartyNumber.

Exceptional procedure

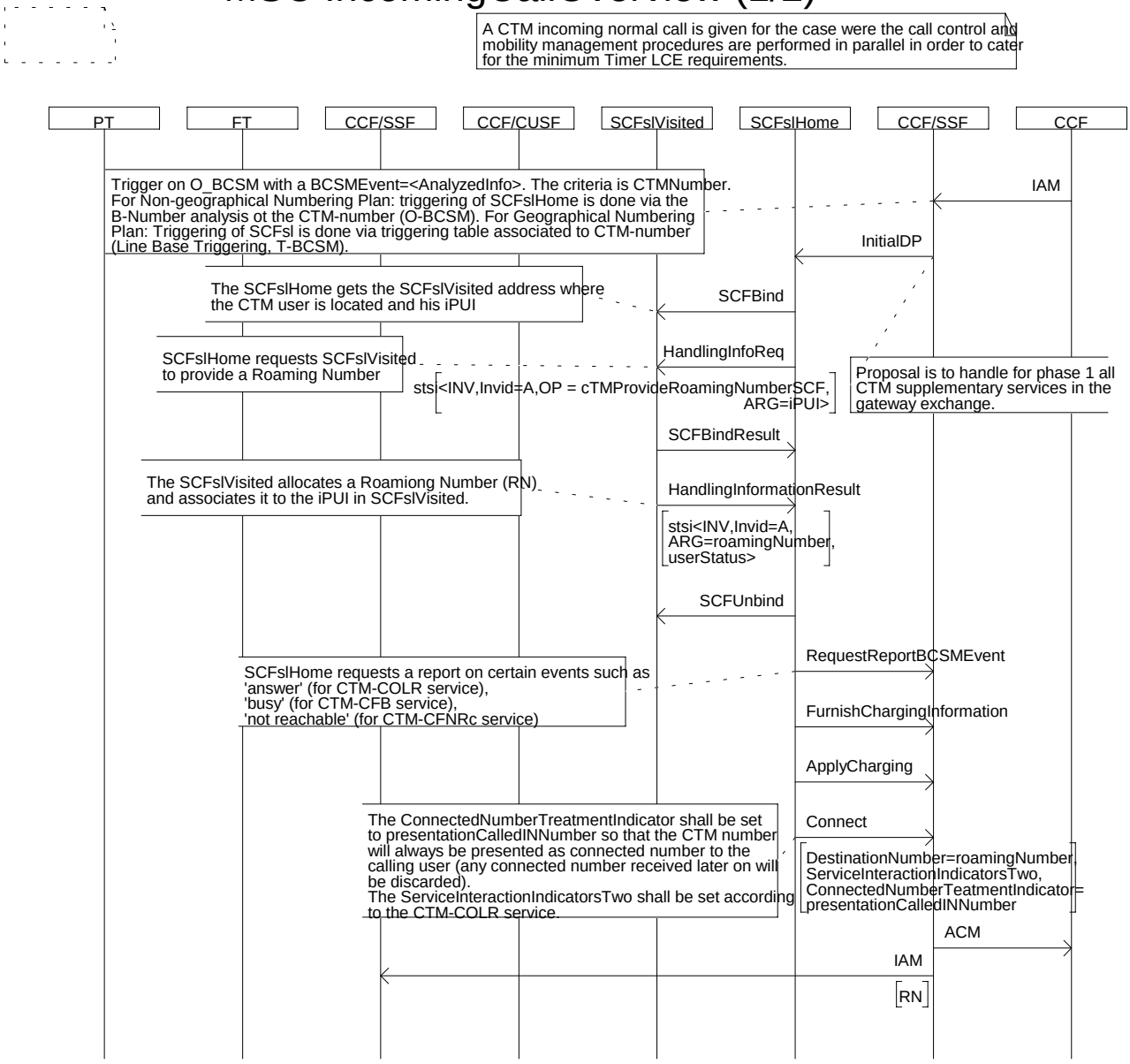
1) If the SCFslVisited determines that the received cTMOutgoingCallMobilityManagementInfo invoke component part of the USI parameter of the InitialDP operation is incorrect, or the SCFmmVisited does not receive an expected terminal authentication result for normal calls, the SCFslVisited sends a cTMOutgoingCallMobilityManagementInfo return error component with error value "networkRejected" in the SendSTUI operation and releases the call by sending a ReleaseCall operation to the SSF. The ReleaseCall operation contains no specific cause so that the cause value (#31) "Normal, unspecified" is sent towards the CTM user. The SCFslVisted is also able to perform a ReleaseCall operation during the active phase of a call (for the case where authentication is performed in parallel with the call); therefore a monitor relationship will be maintained.

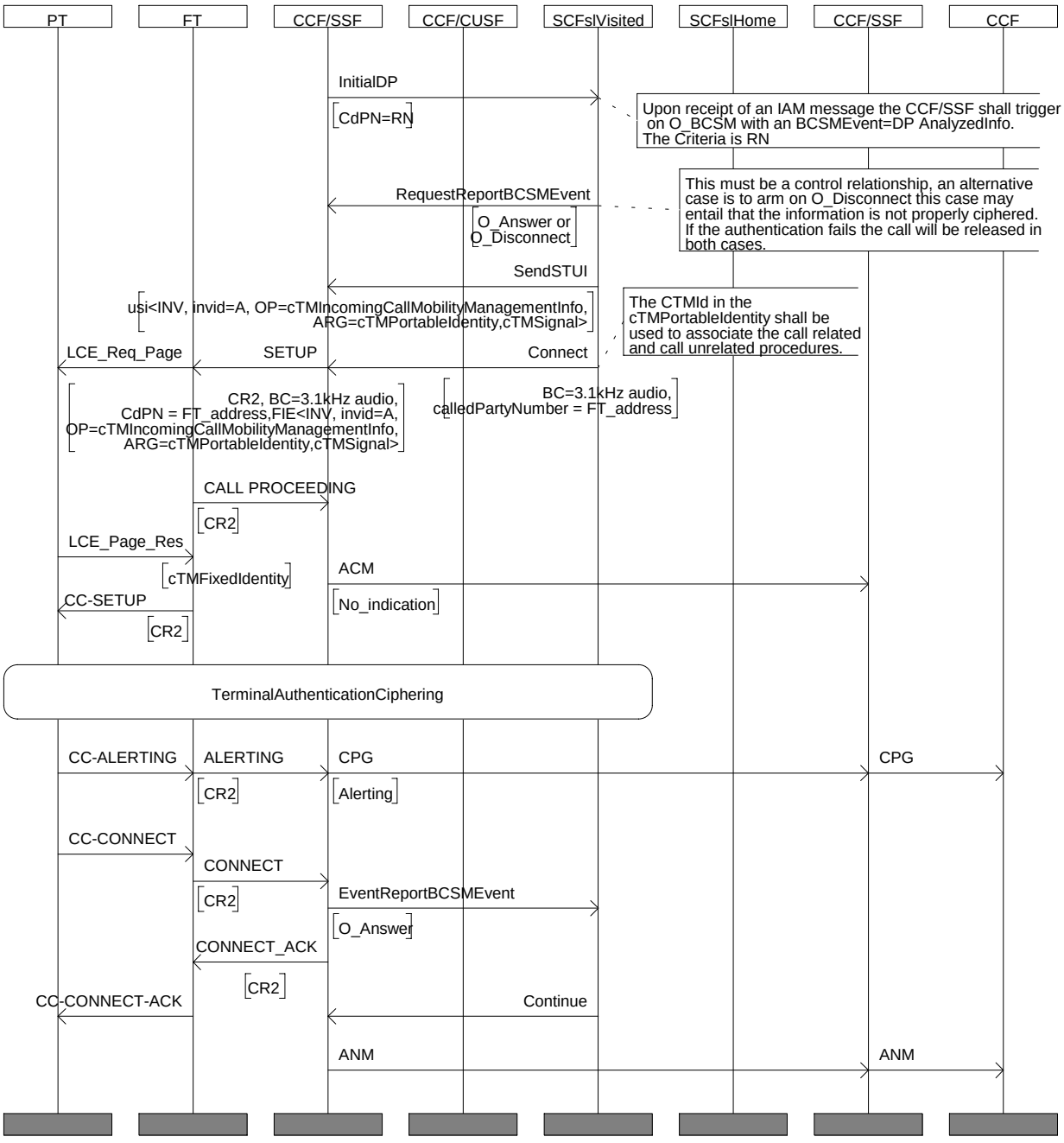
2) When the check in the SCFslVisited of the requested collected dialled digits with regard to the standard emergency number (in Europe, "112") does match and the terminal authentication check fails the SCFslVisited does not release the call. The SCFslVisited ensures that the called party number received cannot be an emergency number before releasing the call due to the failure of the terminal authentication procedure.

3) When the SCFmm receives the Ciphering return error component, the SCFslVisited may either release the call or proceed in the existing mode.

B.2 Incoming call

MSC IncomingCallOverview (1/2)





Normal operation

1) When a CTM-number is received the call is routed by means of the ISUP IAM message to the nearest CCF/SSF.

2) The SSF triggers the SCFslHome by sending an InitialDP and is based on whether the CTM-number is belonging to a geographical or non-geographical numbering plan.

For the non-geographical numbering plan, the SSF triggers based on the B party analysis on the O_BCSM with a BCSMEvent equal to AnalysedInformation and the service criteria is based on the CTM-number.

For the geographical numbering, plan the SSF triggers based on a triggering table associated with the CTM-number (line based triggering) on the T_BCSM with the BCSMEvent equal to TerminationAttempt and the service criteria is based on the CTM-number.

In order to address the SCFslHome, the CTM-number is used to be translated to a global title.

3) The SCFslHome requests the SCFslVisited for a Roaming Number (RN) by sending an cTMProvideRoamingNumberSCF invoke component using the SSI mechanism on the SCF-SCF interface. The cTMProvideRoamingNumberSCF invoke component contains the IPUI as parameter.

4) On receipt of the cTMProvideRoamingNumberSCF, the SCFslVisited checks whether the CTM user is registered or not.

If the CTM user is defined as not reachable in the SCFslVisited, the SCFslVisited returns the cTMProvideRoamingNumberSCF return result component using the SSI mechanism on the SCF-SCF interface, containing the status of the user (userNotReachable).The SCFslHome can execute the CTM supplementary service CTM-CFNRc, if active for the user.

The SCFslHome in the Gateway will be the network where all terminating CTM supplementary services are executed (e.g. CTM-COLR, CTM-CLIP, CTM Call Forwarding services). For this reason the SCFslHome requests via a requestReportBCSMEvent operation Basic Call State Machine (BCSM) events such as O_Answer, O_NoAnswer, O_NotReachable, O_Busy, O_Disconnect.

5) The SCFslHome instructs the SSF by sending a Connect operation to route the call with the following parameters:

   - DestinationNumber containing the roaming number;

   - ServiceInteractionIndicatorsTwo indicating that the dialled number is to be used as connected number;

   - ConnectedNumberTreatmentIndicator set to presentationRestricted if the CTM user has CTM-COLR.

6) The call is routed towards the local exchange where the CTM user is located.

7) In the visited domain the CCF/SSF triggers the SCFslVisited on the O_BCSM with a BCSMEvent equal to AnalysedInfoDP and the service criteria is based on the roaming number.

8) The SCFslVisited releases the roaming number and the SCFmmVisited initiates terminal authentication and/or ciphering via the CUSF mechanism. The SCFmmVisited service script decides whether the authentication and/or ciphering is to be performed in parallel or in sequence with the call setup. A correlation function is performed between the call related and call unrelated procedures by means of the IPUI parameter.

If the terminal authentication and ciphering is run in sequence to the call setup, the call may in some cases not be ciphered because the DECT GAP "Link maintain timer": LCE.02 may be set to zero in the PP. This causes the radio link to be taken down when the ciphering is completed and before the call setup is sent to the PP.

If the terminal authentication and ciphering is run in parallel to the call setup, the signalling information on the air is not ciphered before the cipher request is sent to the PP.

It is noted that for non-geographical numbers **in the case the CTM user has not roamed away from the node where the SCFslHome was triggered, there will be a multiple point of control for the AnalysedInformation DP**.

The SCFslVisted generates an announcement towards the A subscriber if the call setup takes too long a time and an early Address Complete Message (ACM) message in the ISUP protocol will be generated.

9) The SCFslVisited requests the CCF/SSF to route the call towards the towards the FT by sending a connect operation containing a calledPartyNumber equal to the FT_address. The SCFslVisited includes the cTMIncomingCallMobilityManagementInfo invoke component using the USI mechanism in the SendSTUI operation.

The cTMIncomingCallMobilityManagementInfo invoke component contains the following parameters:

   - cTMPortableIdentity containing the IPUI;

   - cTMSignal containing the signal to be generated by the portable.

10) The CCF/SSF routes the call towards the FT.

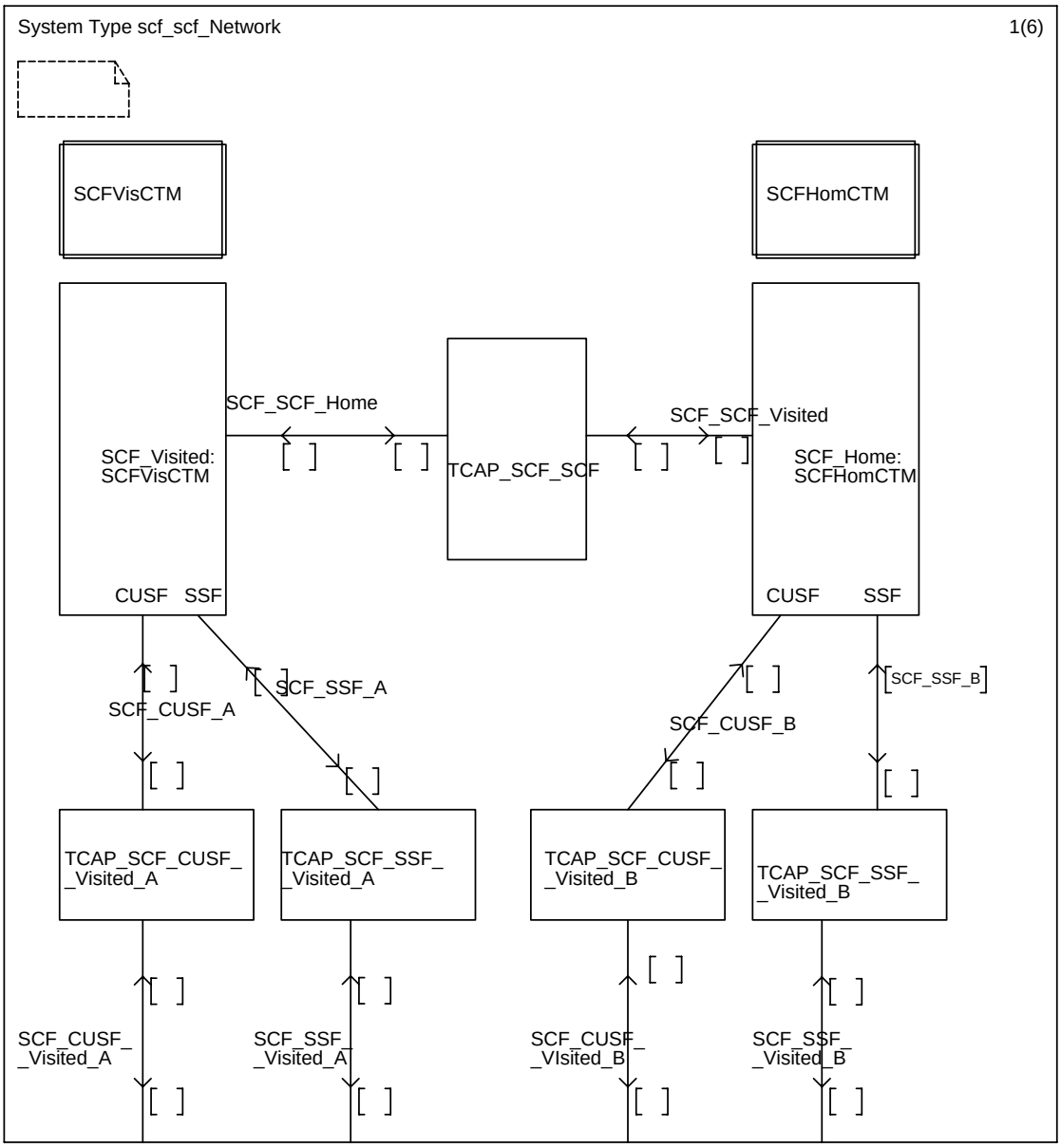
Annex C (informative): SDL model

This annex contains the SDL model of the CTM application. The SDL model is informative only.

Annex C1 : System Type scf_scf_Network

- Annex C2 : Block Type SCFVisCTM
  - Annex C3 : Process Type SCF_SSF_CUSF_ASE_Visited
    - Annex C4 : Procedure DetermineCTMArg
    - Annex C5 : Procedure GetInvokeID
  - Annex C6 : Process Type SCFmmVisitedApplication
    - Annex C7 : Procedure LocationRegistrationSCF
    - Annex C8 : Procedure LocationCancellationFT
    - Annex C9 : Procedure DownloadSecurityData
    - Annex C10 : Procedure TerminalAuthentication
    - Annex C11 : Procedure Ciphering
    - Annex C12 : Procedure IdentityRequest
    - Annex C13 : Procedure SubscriptionRegistration
    - Annex C14 : Procedure LocationRegistration
    - Annex C15 : Procedure LocationRegistrationSuggest
  - Annex C16 : Process Type SCFslVisitedApplication
    - Annex C17 : Procedure ProvideRoamingNumber
  - Annex C18 : Process Type SCF_SCF_ASE_Visited
    - Annex C19 : Procedure DetermineCtmArg_SCFVis
- Annex C20 : Block Type SCFHomCTM
  - Annex C21 : Process Type SCFmmHomeApplication
    - Annex C22 : Procedure LocationRegistration_SCF
    - Annex C23 : Procedure SubscriptionDeregistration_SCF
    - Annex C24 : Procedure RestoreData_SCF
    - Annex C25 : Procedure TerminalAuthentication_SCF
    - Annex C26 : Procedure Ciphering_SCF
    - Annex C27 : Procedure NetworkAuthentication_SCF
    - Annex C28 : Procedure KeyAllocate_SCF
    - Annex C29 : Procedure LocationCancellation_SCF
    - Annex C30 : Procedure SubscriptionRegistration_SCF
    - Annex C31 : Procedure DownloadSecurityData_SCF

Annex C1 : System Type scf_scf_network



System Type scf_scf_Network 2(6)

```
/*
The SDL diagrams reflect an Application Layer Structure (ALS) where the SACF performs the association
co-ordination function.
The Visited SCF has four Application Service Entities (ASEs) namely:

 SCFVis   having process type SCF_SSF_CUSF_ASE_Visited,for communication with the SSF/CUSF
   in the Visited Domain.
 SCFmmVisApp having process type SCFmmVisitedApplication, performing the service logic for the
   CTM Phase 1 mobility management call unrelated applications in the Visited SCF Domain.
 SCFslVisApp having process type SCFslVisitedApplication, performing the service logic for the
   CTM Phase 1 call related applications in the Visited SCF Domain.
 SCF_SCFVis  having process type SCF_SCF_ASE_Visited, for communcation with the SCF in the Home
   Domain.


The Home Domain SCF has four Application Service Entities (ASEs) namely:

 SCFHom   having process type SCF_SSF_CUSF_ASE_Home,for communication with the SSF/CUSF
   in the Home Domain.
 SCFmmHomApp having process type SCFmmHomeApplication, performing the service logic for the
   CTM Phase 1 mobility management call unrelated applications in the Home SCF Domain.
 SCFslHomApp having process type SCFslHomeApplication, performing the service logic for the
   CTM Phase 1 call related applications in the Home SCF Domain.
 SCF_SCFHom  having process type SCF_SCF_ASE_Home, for communcation with the SCF in the Visited
   Domain.

The SCFHom and SCFVis ASEs communicate with the TCAP adaptors for the CUSF and SSF entities, while the
SCF_SCFHom and SCF_SCFVis communicate with the TCAP adaptors for the peer SCF entities.

The association instance  for the SCFHom and SCFVis is identified by either the cUSFdialogueID for call unrelated
CUSF communication or by the sSFdialogue for call related SSF communication.
The association instance for the SCF_SCFHom and SCF_SCFVis is identified by the sCFdialogueID for communication
with the peer SCF entity. The association instance for the SCFmmVisApp and SCFmmHomApp is identified by the
mMdialogueID while for the SCFslVisApp and SCFslHomApp is identified by the sLdialogueID.

During one dialogue unique invoke identifiers are generated.  They are obtained by calling a GetInvokeID
procedure within the send signal parameter e.g. ContinueAssociationRequest((call GetInvokeID), cUSFdialogueID, conaArg)

The communication of the ASEs performing the transport functions via the SACF are given by the appropriate
signal name e.g.RequestReportBCUSMEvent for the invoke component, by RequestReportBCUSMEventRes for the return
result component and by RequestReportBCUSMEventErr for the return error component.

The association instances for the SCFmmVisApp, SCFmmHomApp, SCFslVisApp and SCFslHomApp ASEs are identified by the
Process Instance Identifiers (PID).
Since the signals send from or received by the transport ASEs by/from the application processes are not unique due to
the fact that :
    * relaying of signals from the SCFHome occur in the SCFVisited SACF, and;
    * embedded procedures are invoked for the same entity
it is required that different dialogue identifiers are used. In this repect a qualifier will indicate the nesting which
 has occured e.g. cUSF1dialogueID.

For the ASEs the creation of an application process instance is initiated by sending a CreateASE signal
with  a parameter indicating the process  which is addressed e.g SCFmmVisApp.
The processes addressed are indicated by a parameter value having the type
TypeOfASE ::= ENUMERATED { -- Indicates the type of ASE to be created by SACF
  SCFmmVisApp,
  SCF_SCFVis,
  SCFmmHomApp,
  SCF_SCFHom,
  SCFVis,
  SCFHom
}
The process instances are terminated by sending an ASETerminated(x)signal, where x designates the TypeOfASE
e.g. SCFmmVisApp.

For the TCAP simulator the signal ApplicationBegin and ApplicationEnd are used.
```

System Type scf_scf_Network 3(6)

/* Data type definitions used by the TCAP Simulator */

```
/* Needed for the initialization of the TCAP Simulator */
NEWTYPE IHroleType
  LITERALS
    A_Side, B_Side;
ENDNEWTYPE;

/* Dialog IDs */
/* - Total valid numbers 1 - 100 */
/* - Direction SCF -> SSF: 1 - 50 */
/* - Direction SSF -> SCF: 51 - 100 */
/* - 0 is used as flag, if something strange happened, e.g., for a given csaID no dialogID is found */
SYNONYM maxDialogIDtotal Integer = 100;
SYNONYM maxDialogIDtoSSF Integer = 50;
SYNTYPE DialogIDtype = Integer CONSTANTS 0:maxDialogIDtotal
ENDSYNTYPE;

/* Invoke IDs */
/* - Total valid numbers 1 - 200 */
/* - Direction SCF -> SSF: 1 - 100 */
/* - Direction SSF -> SCF: 101 - 200 */
/* - 0 is used as dummy */
SYNONYM maxInvokeIDtotal Integer = 200;
SYNONYM maxInvokeIDtoSSF Integer = 100;
SYNTYPE InvokeIDtype = Integer CONSTANTS 0:maxInvokeIDtotal
ENDSYNTYPE;

/* For indicating the origin of TCAP messages */
NEWTYPE TCoriginType
  LITERALS
    oSSF, oSCF;
ENDNEWTYPE;

/* refers to basic and prearranged end in TC_EndReq/Ind primitives */
NEWTYPE TCAPterminationType
  LITERALS
    basic, prearranged;
ENDNEWTYPE;

/* For Timeout values for operations, mandatory parameter in TC_InvokeReq primitives */
NEWTYPE TimeoutValType
  LITERALS
    short, medium, long;
ENDNEWTYPE;
```

System Type scf_scf_Network 4(6)

```
/* Reasons for a TCAP failure */
NEWTYPE TCAPfailReasonType
  LITERALS
    wrongDiaID, /* incorrect dialog ID: incorrect range or no dialog for ID present/available */
    wrongInvokeID, /* incorrect invoke ID */
    wrongCSaID, /* incorrect CSA ID */
    beginRequired, /* TC_BeginReq is required */
    noComp, /* TC_BeginReq without components */
    unknownOP, /* Unknown operation code (within TC Invoke) */
    noDiaIDavail, /* No dialogID available, 'maxDialogIDtoSCF' Diallogs are established */
    cSAnotAvail, /* the provide CSAid is not available */
    unexpSignal, /* the TCAP IH received an unexpected signal */
    signalNotProcessed, /* signal cannot be processed by the TCAP signal handler */
    preArrangedEndReq; /* End signal with wrong termination parameter */
ENDNEWTYPE;
```

```
/* Operation Codes */
NEWTYPE OpCodeType
  LITERALS
  /* No Operation */ NoOperation,
  /* CLASS 1 */ CSA, DL, MTD, MC, MCS, ML, RCS, RES, SL,
  /* CLASS 2 */ ASF, AC, CIRQ, CAN, CSR, CI, CON, CTR, CWA,
            DFC, DFCWA, ETC, FCI, ICA, RE, RFS, RNC, RRB, RRU, RT, SCI, SS,
  /* CLASS 4 */ AT, CG, CUE, RC,
  /* FROM SSF */ ACR, ARI, CIR, IDP, ER, ENC, ERB, RU, SFR, SRP,
  /* RETURN RESULTS FROM SSF */ AT_R, ASF_R, SL_R, DL_R, MCS_R, MC_R,
            ML_R, RCS_R, RES_R, MTD_R, CSA_R;
ENDNEWTYPE;

/* Operation Classes */
SYNTYPE OpClassType = Integer CONSTANTS 0:4
ENDSYNTYPE;
/*- 0 means no class */
```

System Type scf_scf_Network 5(6)

```
/* SIGNAL Definitions for the TCAP Adapter*/
```

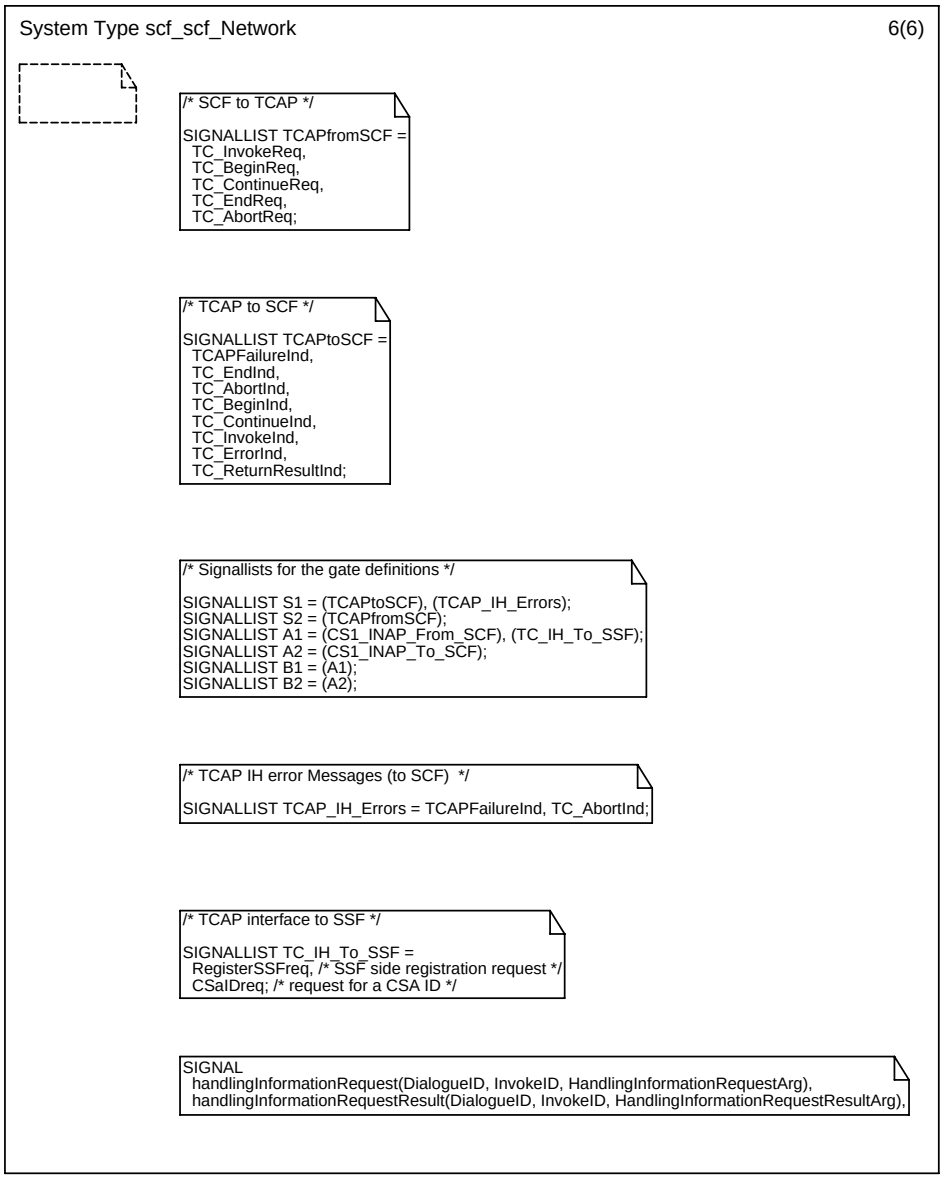
```
/* No Operations, but additional Signals exchanged between TCAP Adapter, SSF and SCF */

SIGNAL
  CSaIDreq(DialogIDtype),
  CSaIDresp(DialogIDtype,CSAID),
  RegisterSSFreq(IHroleType),
  RegisterSSFresp(IHroleType),
  TCAPFailureInd(TCAPfailReasonType);
```

```
/* TCAP Primitives */

SIGNAL
/* TC_nameReq: Direction SCF->TCAP */
  TC_InvokeReq(InvokeIDtype,DialogIDtype,OpClassType,OpCodeType,TimeoutValType,ArgType),
  TC_BeginReq(DialogIDtype,TCoriginType),
  TC_ContinueReq(DialogIDtype,TCoriginType),
  TC_EndReq(DialogIDtype,TCAPterminationType),
  TC_AbortReq(DialogIDtype),

/* TC_nameInd: Direction TCAP->SCF */
  TC_BeginInd(DialogIDtype,TCoriginType,Boolean),
  TC_ContinueInd(DialogIDtype,TCoriginType,Boolean),
  TC_InvokeInd(InvokeIDtype,DialogIDtype,OpCodeType,Boolean,ArgType),
  TC_EndInd(DialogIDtype,TCAPterminationType,Boolean),
  TC_AbortInd(DialogIDtype),
  TC_ErrorInd(InvokeIDtype,DialogIDtype,Boolean),
  TC_ReturnResultInd(InvokeIDtype,DialogIDtype,Boolean,opCodeType,ArgType);
```

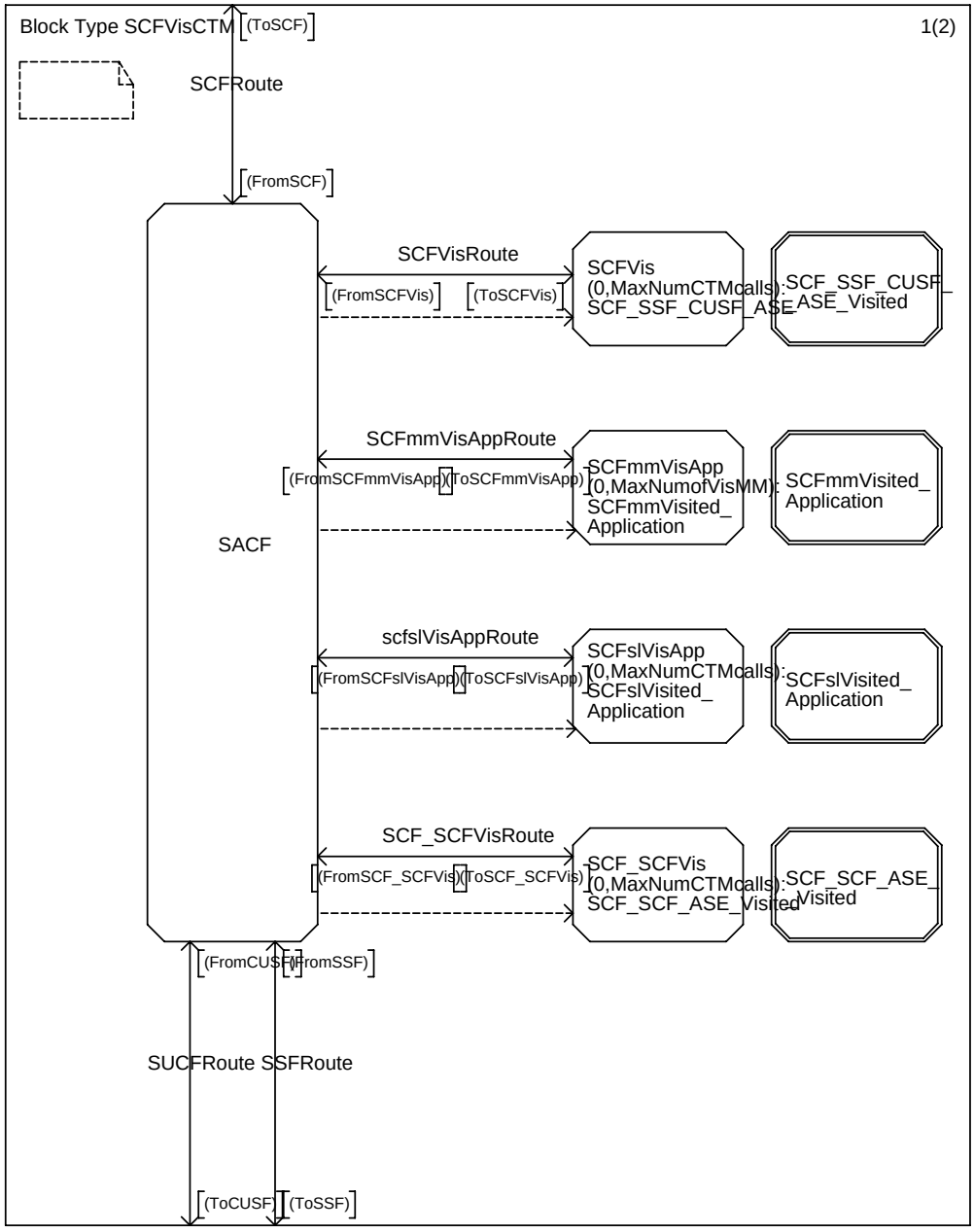
```
/**** SIGNAL DEFINITIONS FOR THE IN CS-1 ERROR AND DIALOUGE HANDLING ****/

/* Note: Used between the HalfCalls (SSF/CCF_A and SSF/CCF_B) and the TCAP Adaptor. */
```

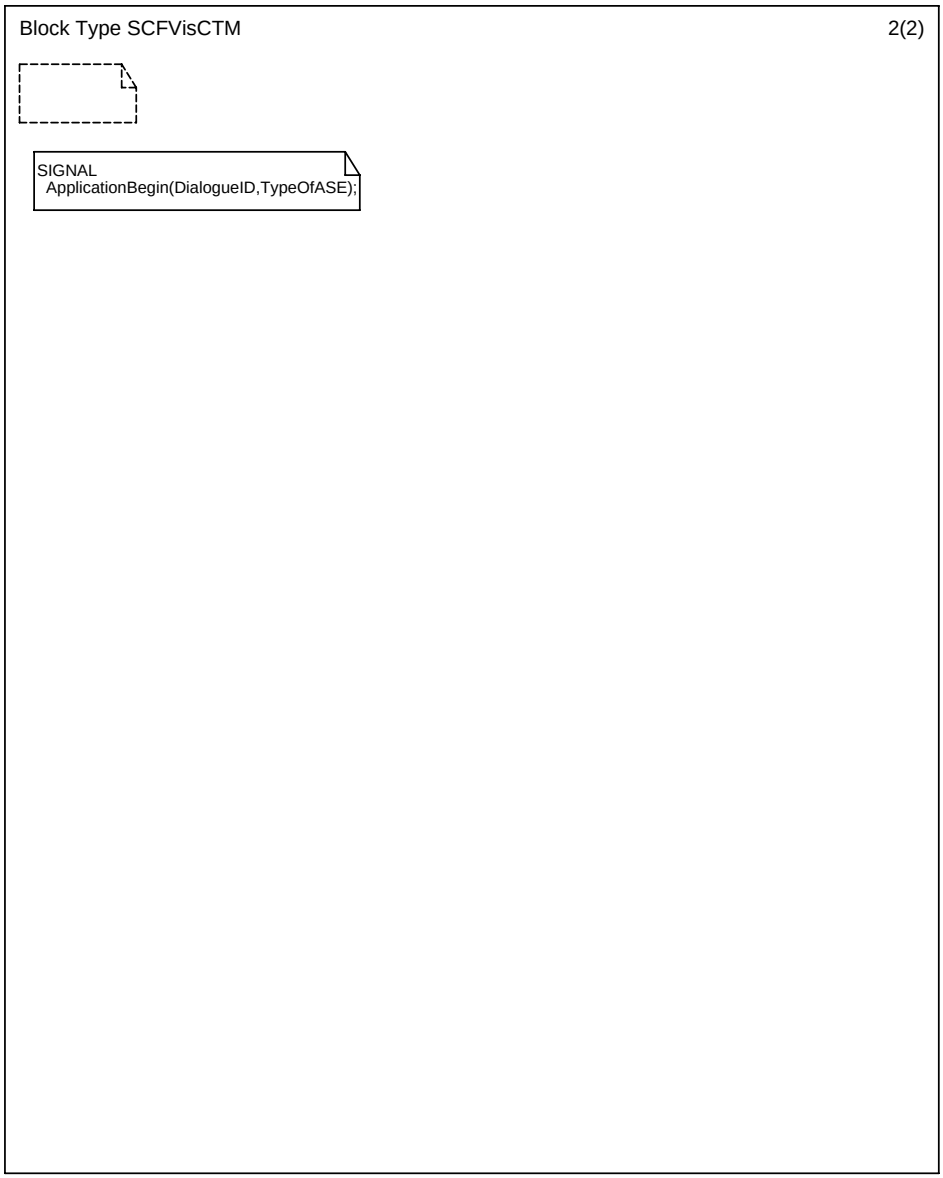
```
/* Type definition used for the return error. The values
correspond to the error codes defined in the ASN.1 */
SYNTYPE ErrorArg = NATURAL
  CONSTANTS 0:23
ENDSYNTYPE;
```

```
SIGNAL
  Errror(InvokeID,CSAID,ErrorArg), /* Return error. */
  ApplicationBegin(CSAID),        /* Begin a dialouge. */
  ApplicationContinue(CSAID),
  ApplicationAbort(CSAID),        /* Abort a dialouge. */
  ApplicationEnd(Boolean,CSAID);  /* End a dialouge
                                     indicating whether prearranged end or not. */
```

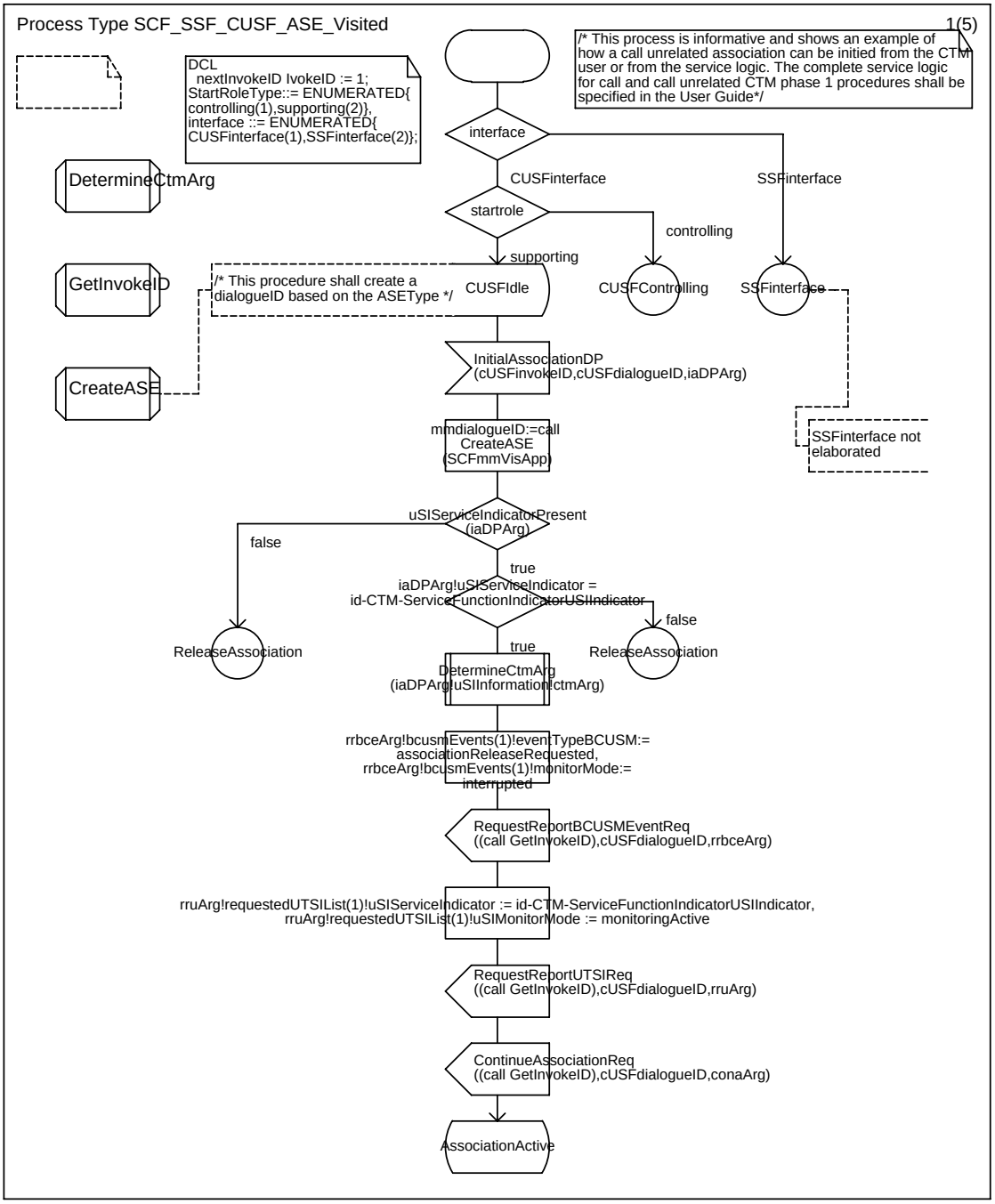

System Type scf_scf_Network
6(6)
/* SCF to TCAP */
SIGNALLIST TCAPfromSCF =
TC_InvokeReq,
TC_BeginReq,
TC_ContinueReq,
TC_EndReq,
TC_AbortReq;
/* TCAP to SCF */
SIGNALLIST TCAPtoSCF =
TCAPFailureInd,
TC_EndInd,
TC_AbortInd,
TC_BeginInd,
TC_ContinueInd,
TC_InvokeInd,
TC_ErrorInd,
TC_ReturnResultInd;
/* Signallists for the gate definitions */
SIGNALLIST S1 = (TCAPtoSCF), (TCAP_IH_Errors);
SIGNALLIST S2 = (TCAPfromSCF);
SIGNALLIST A1 = (CS1_INAP_From_SCF), (TC_IH_To_SSF);
SIGNALLIST A2 = (CS1_INAP_To_SCF);
SIGNALLIST B1 = (A1);
SIGNALLIST B2 = (A2);
/* TCAP IH error Messages (to SCF) */
SIGNALLIST TCAP_IH_Errors = TCAPFailureInd, TC_AbortInd;
/* TCAP interface to SSF */
SIGNALLIST TC_IH_To_SSF =
RegisterSSFreq, /* SSF side registration request */
CSaIDreq; /* request for a CSA ID */
SIGNAL
handlingInformationRequest(DialogueID, InvokeID, HandlingInformationRequestArg),
handlingInformationRequestResult(DialogueID, InvokeID, HandlingInformationRequestResultArg),

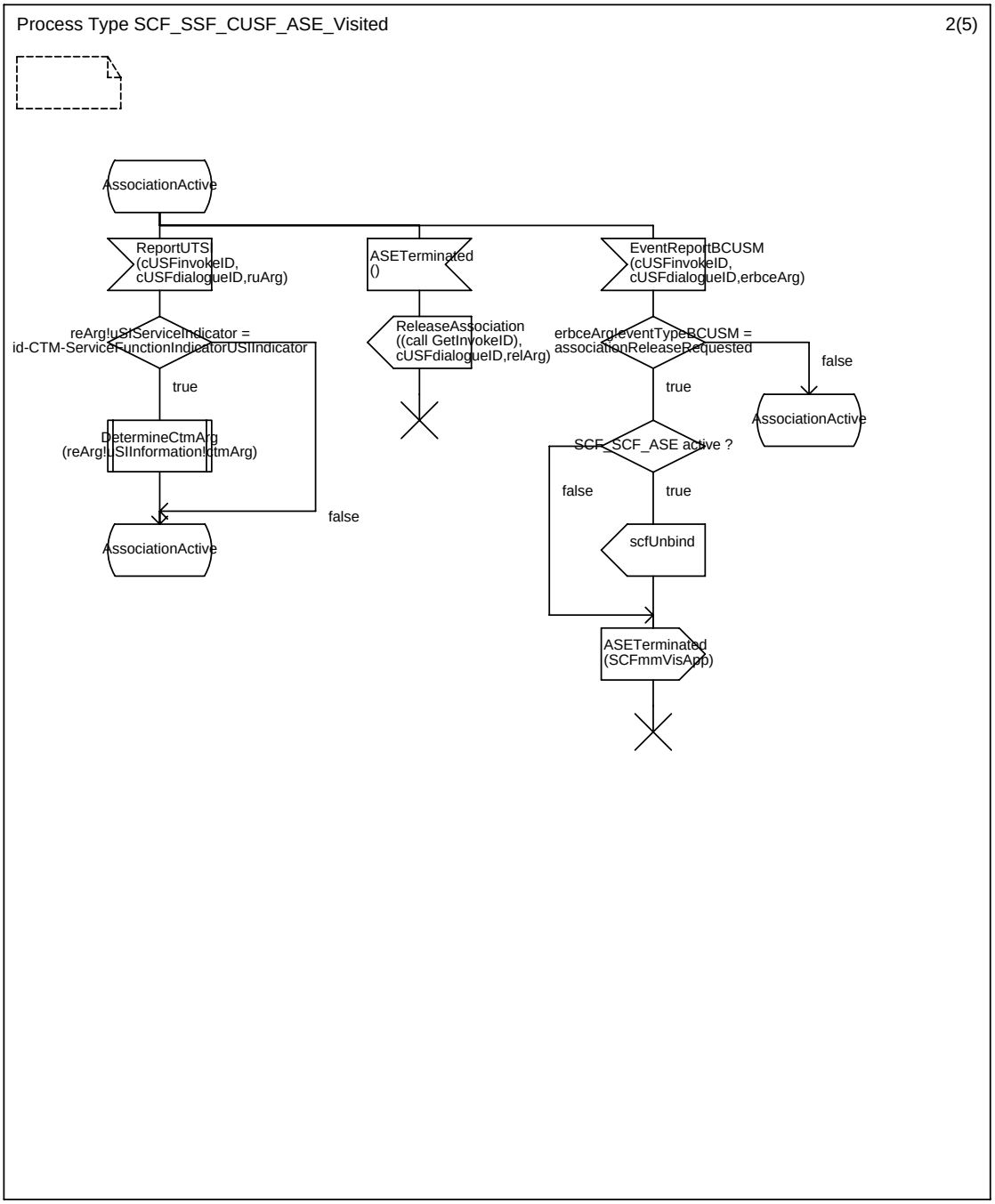
Annex C2 : Block Type SCFVisCTM

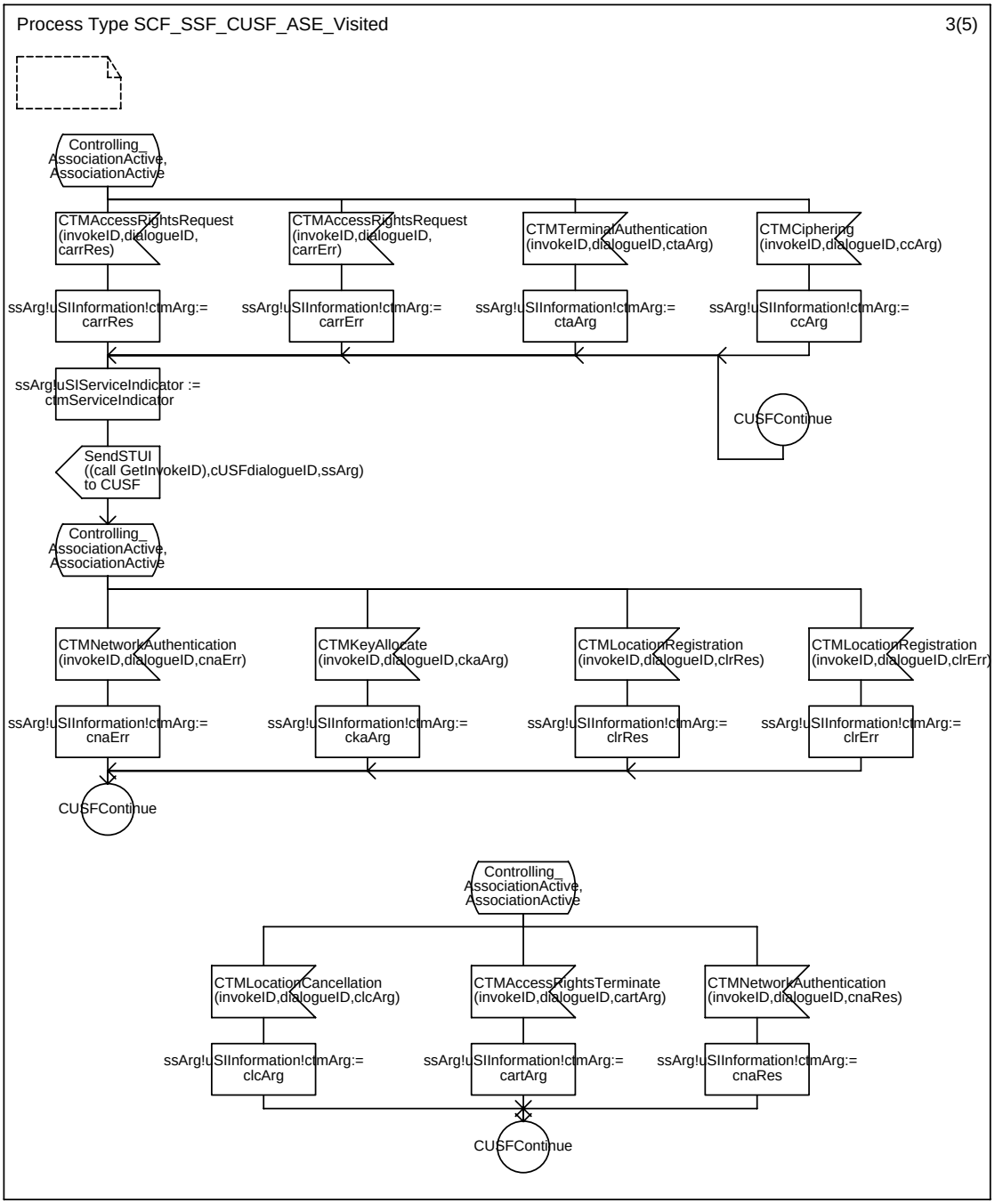



Block Type SCFVisCTM
2(2)
SIGNAL
ApplicationBegin(DialogueID,TypeOfASE);

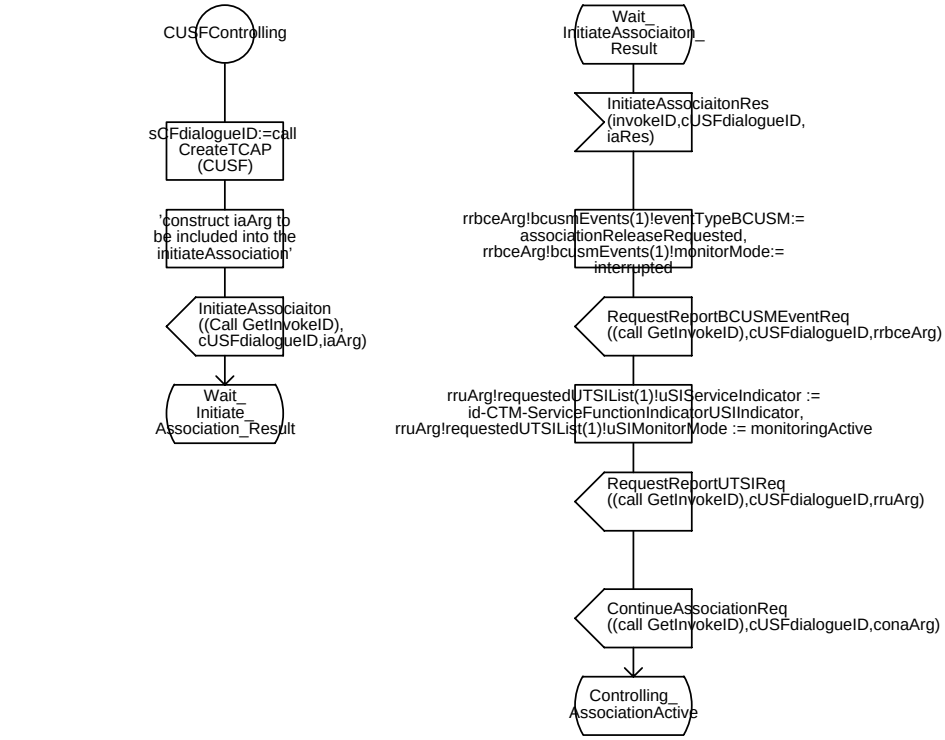
Annex C3 : Process Type SCF_SCF_CUSF_ASE_Visited



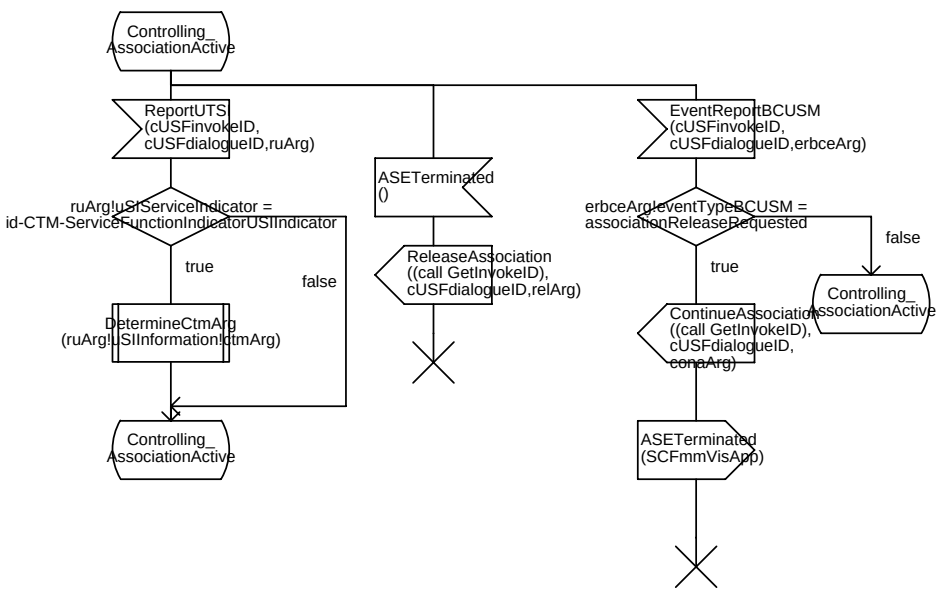

Process Type SCF_SSF_CUSF_ASE_Visited
2(5)
AssociationActive
ReportUTS
(cUSFinvokeID,
cUSFdialogueID,ruArg)
ASETerminated
()
EventReportBCUSM
(cUSFinvokeID,
cUSFdialogueID,erbceArg)
reArg!uSIServiceIndicator =
id-CTM-ServiceFunctionIndicatorUSIIndicator
ReleaseAssociation
((call GetInvokeID),
cUSFdialogueID,relArg)
erbceArg!eventTypeBCUSM =
associationReleaseRequested
false
true
true
AssociationActive
DetermineCtmArg
(reArg!uSIInformation!ctmArg)
SCF_SCF_ASE active ?
false
false
true
scfUnbind
AssociationActive
ASETerminated
(SCFmmVisApp)


Process Type SCF_SSF_CUSF_ASE_Visited
3(5)
Controlling_
AssociationActive,
AssociationActive
CTMAccessRightsRequest
(invokeID,dialogueID,
carrRes)
CTMAccessRightsRequest
(invokeID,dialogueID,
carrErr)
CTMTerminalAuthentication
(invokeID,dialogueID,ctaArg)
CTMCiphering
(invokeID,dialogueID,ccArg)
ssArg!uSIInformation!ctmArg:=
carrRes
ssArg!uSIInformation!ctmArg:=
carrErr
ssArg!uSIInformation!ctmArg:=
ctaArg
ssArg!uSIInformation!ctmArg:=
ccArg
ssArg!uSIServiceIndicator :=
ctmServiceIndicator
CUSFContinue
SendSTUI
((call GetInvokeID),cUSFdialogueID,ssArg)
to CUSF
Controlling_
AssociationActive,
AssociationActive
CTMNetworkAuthentication
(invokeID,dialogueID,cnaErr)
CTMKeyAllocate
(invokeID,dialogueID,ckaArg)
CTMLocationRegistration
(invokeID,dialogueID,clrRes)
CTMLocationRegistration
(invokeID,dialogueID,clrErr)
ssArg!uSIInformation!ctmArg:=
cnaErr
ssArg!uSIInformation!ctmArg:=
ckaArg
ssArg!uSIInformation!ctmArg:=
clrRes
ssArg!uSIInformation!ctmArg:=
clrErr
CUSFContinue
Controlling_
AssociationActive,
AssociationActive
CTMLocationCancellation
(invokeID,dialogueID,clcArg)
CTMAccessRightsTerminate
(invokeID,dialogueID,cartArg)
CTMNetworkAuthentication
(invokeID,dialogueID,cnaRes)
ssArg!uSIInformation!ctmArg:=
clcArg
ssArg!uSIInformation!ctmArg:=
cartArg
ssArg!uSIInformation!ctmArg:=
cnaRes
CUSFContinue

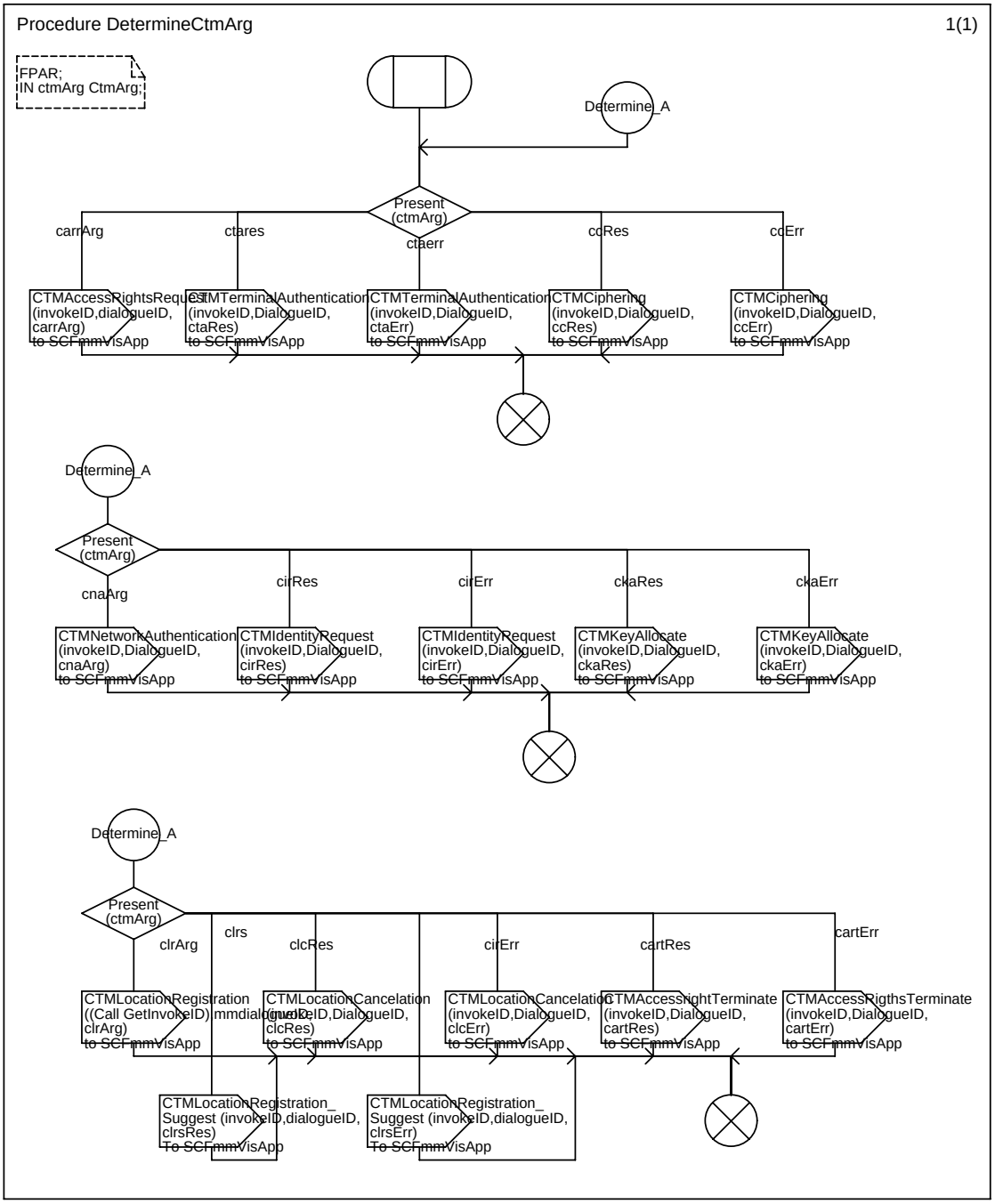
Process Type SCF_SSF_CUSF_ASE_Visited 4(5)



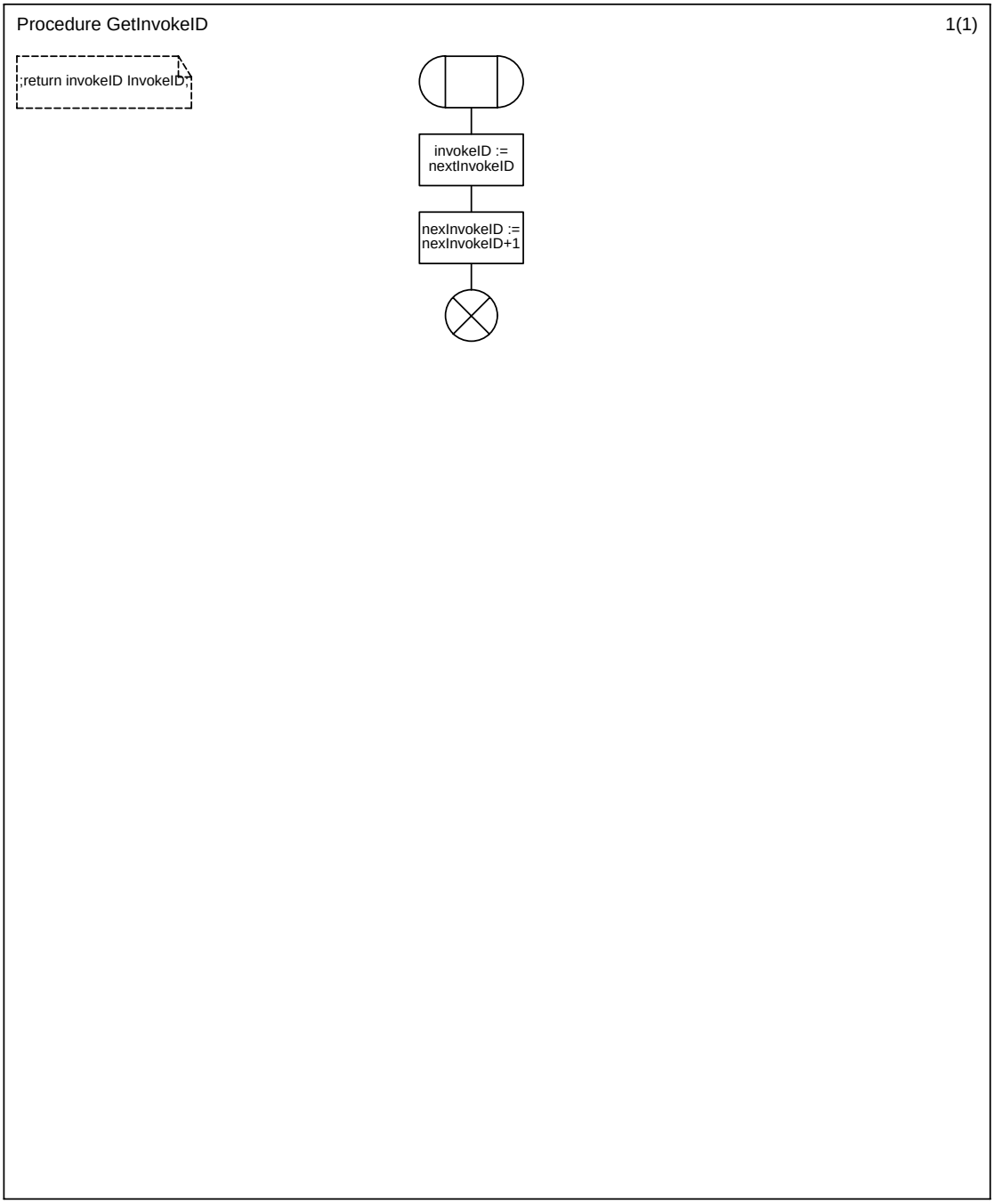
Process Type SCF_SSF_CUSF_ASE_Visited 5(5)



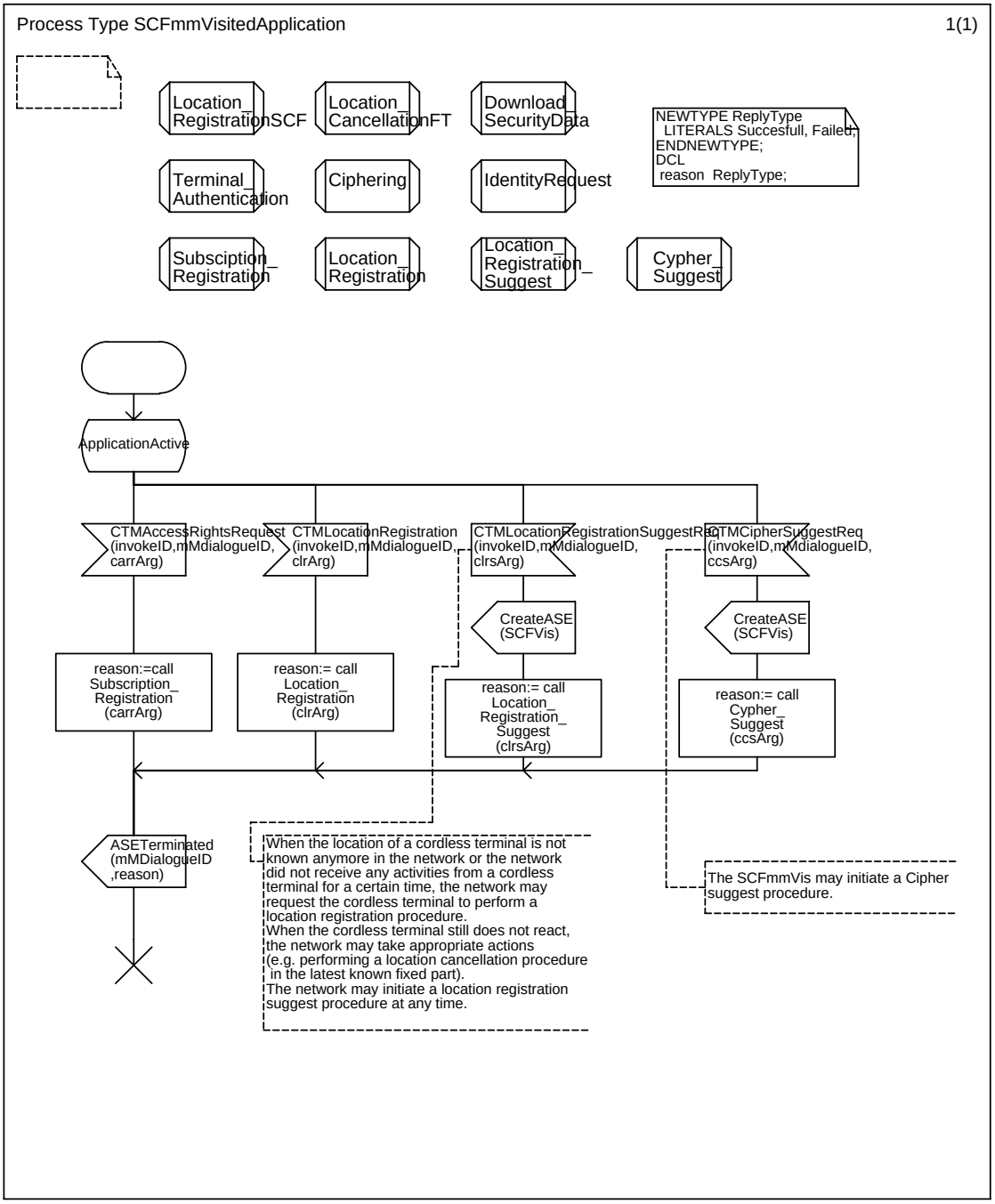
Annex C4 : Procedure DetermineCtmArg



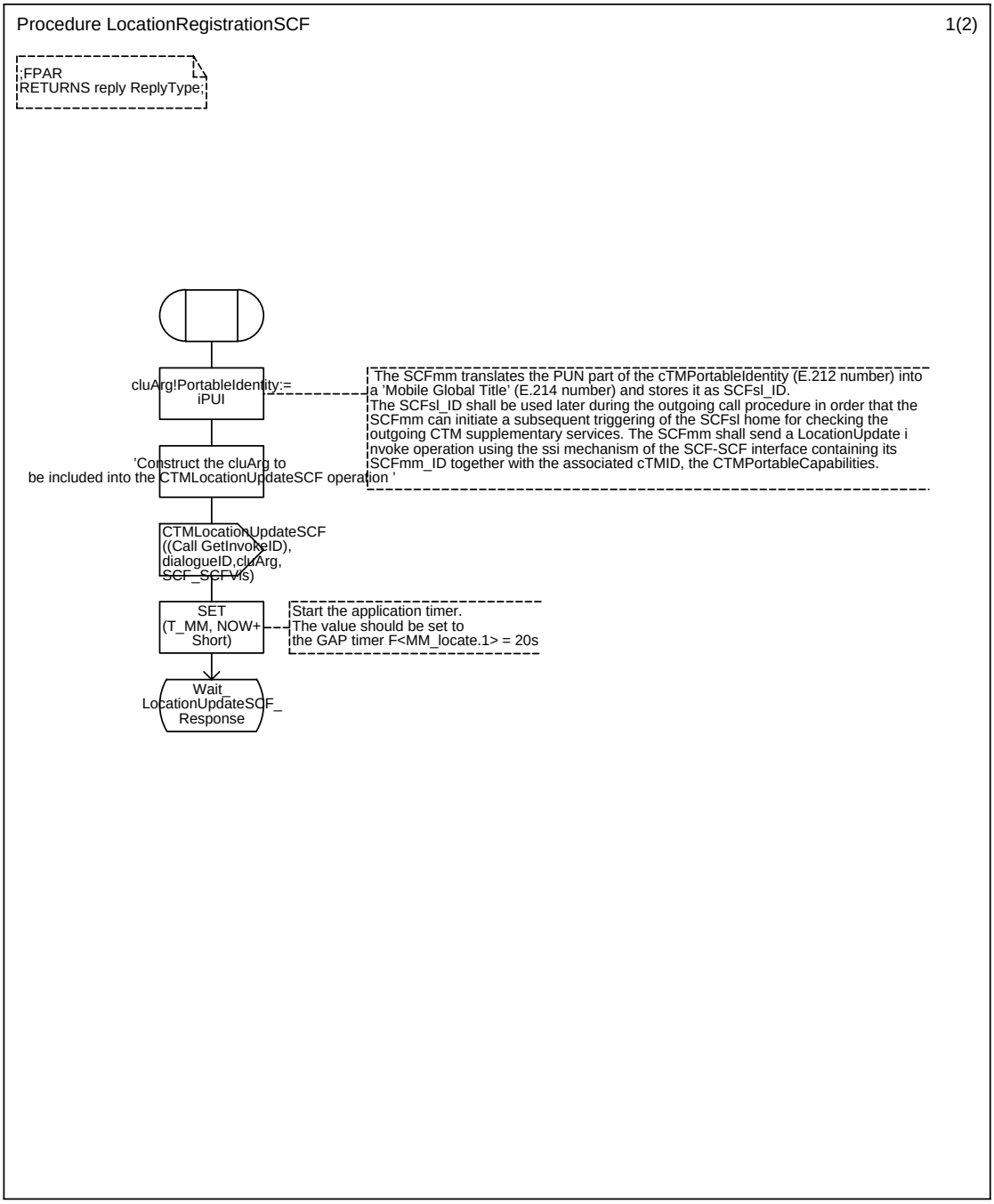
Annex C5 : Procedure GetInvokeID

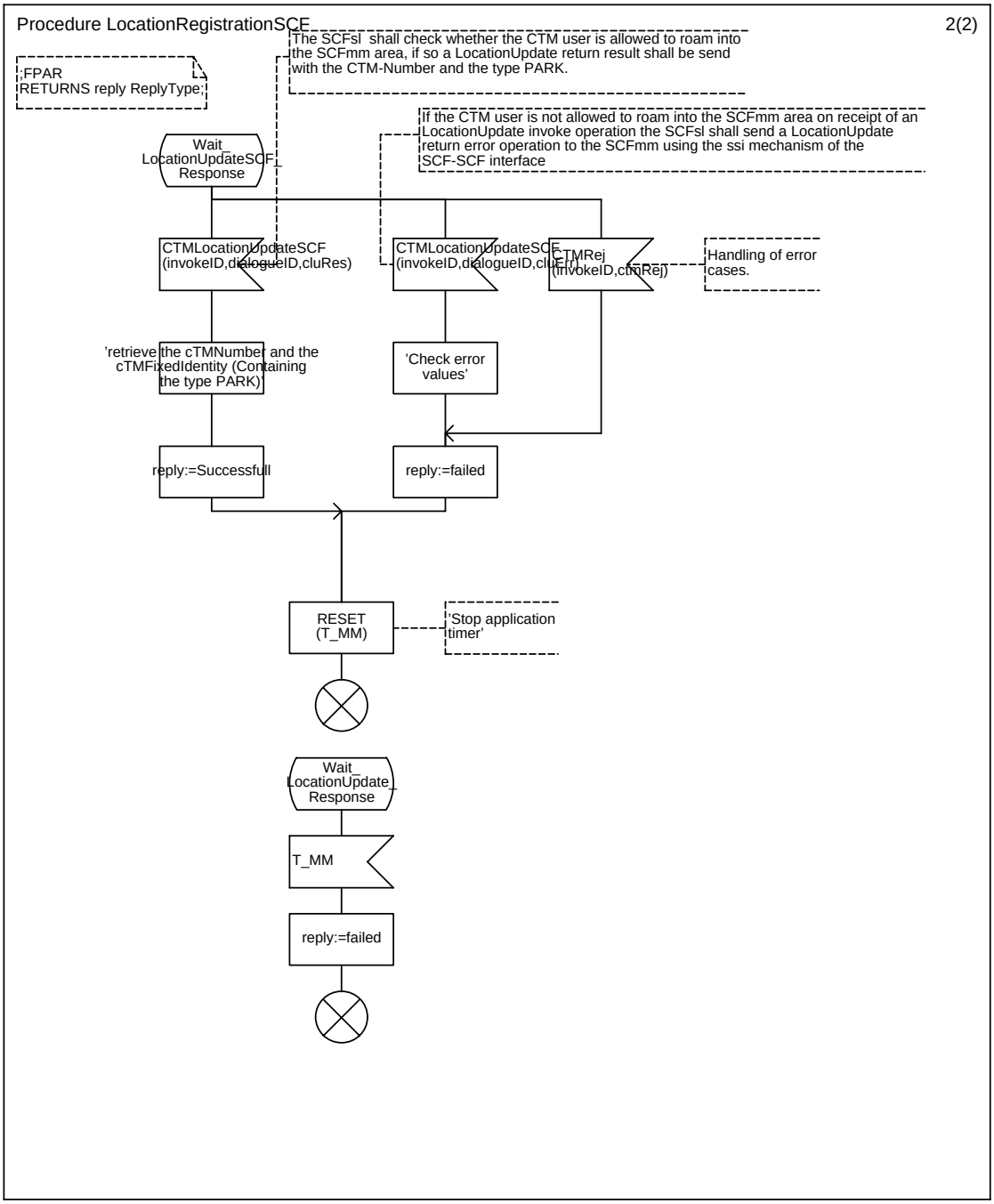


Annex C6 : Process Type SCFmmVisitedApplication

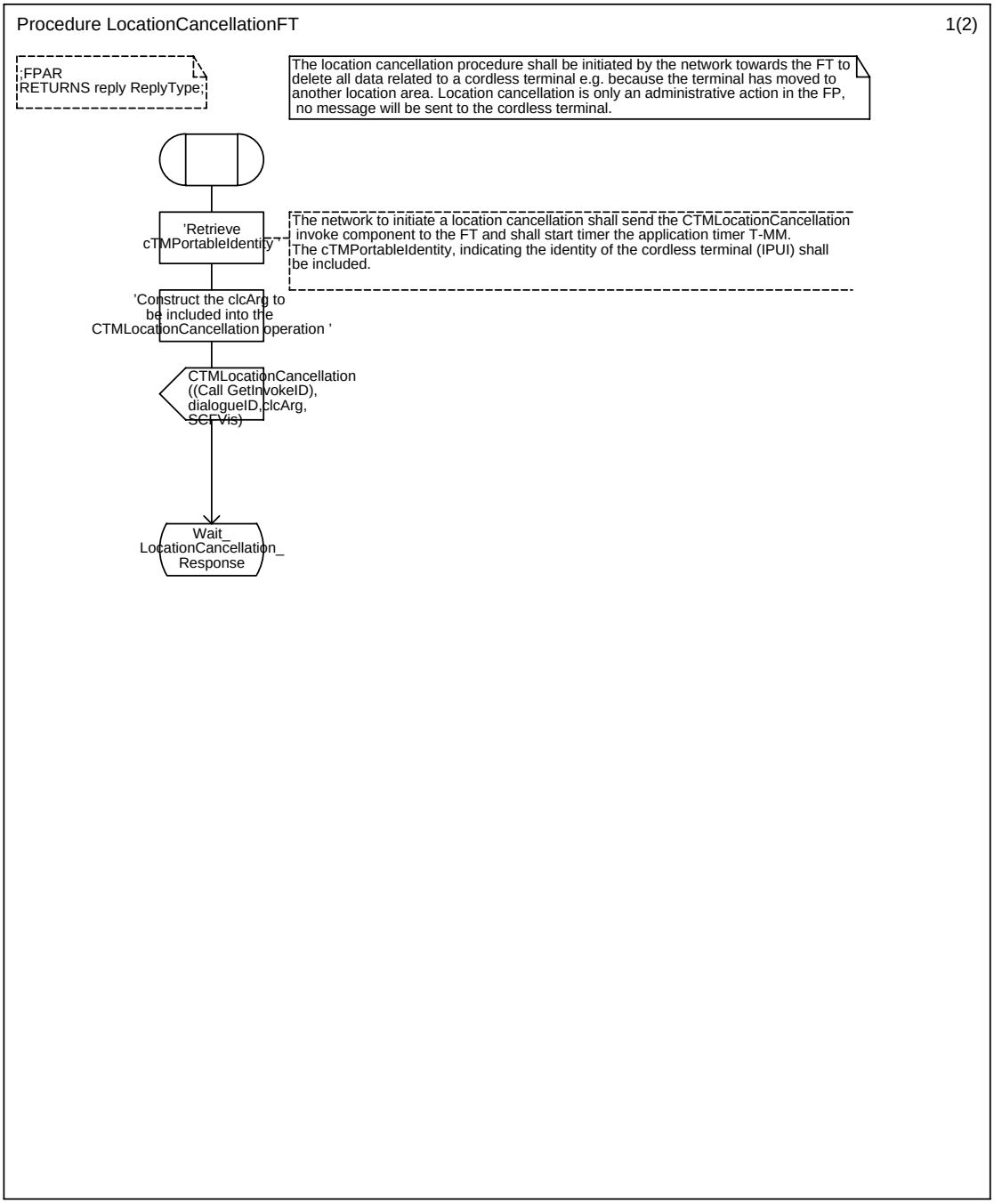


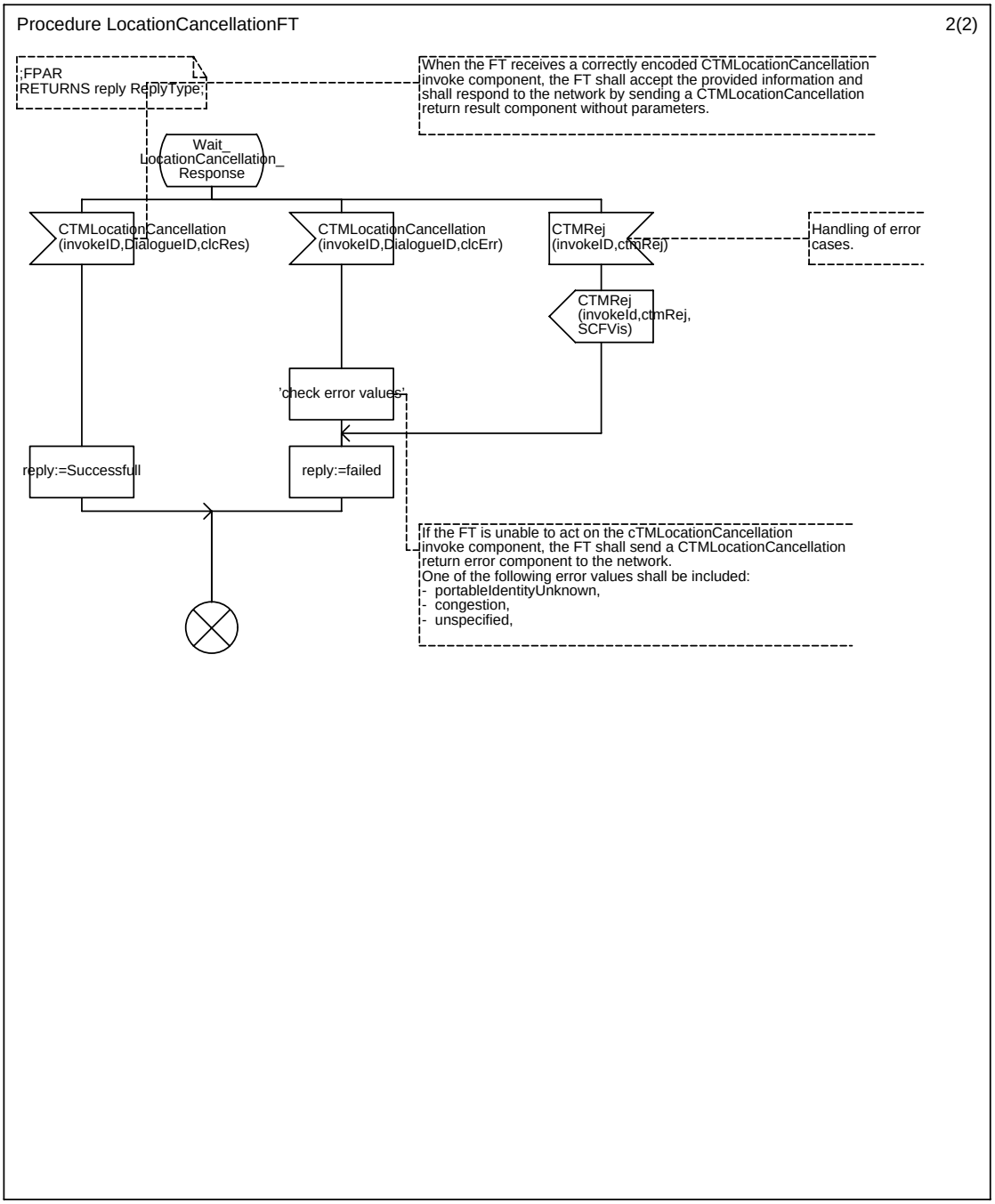
Annex C7 : Procedure LocationRegistrationSCF



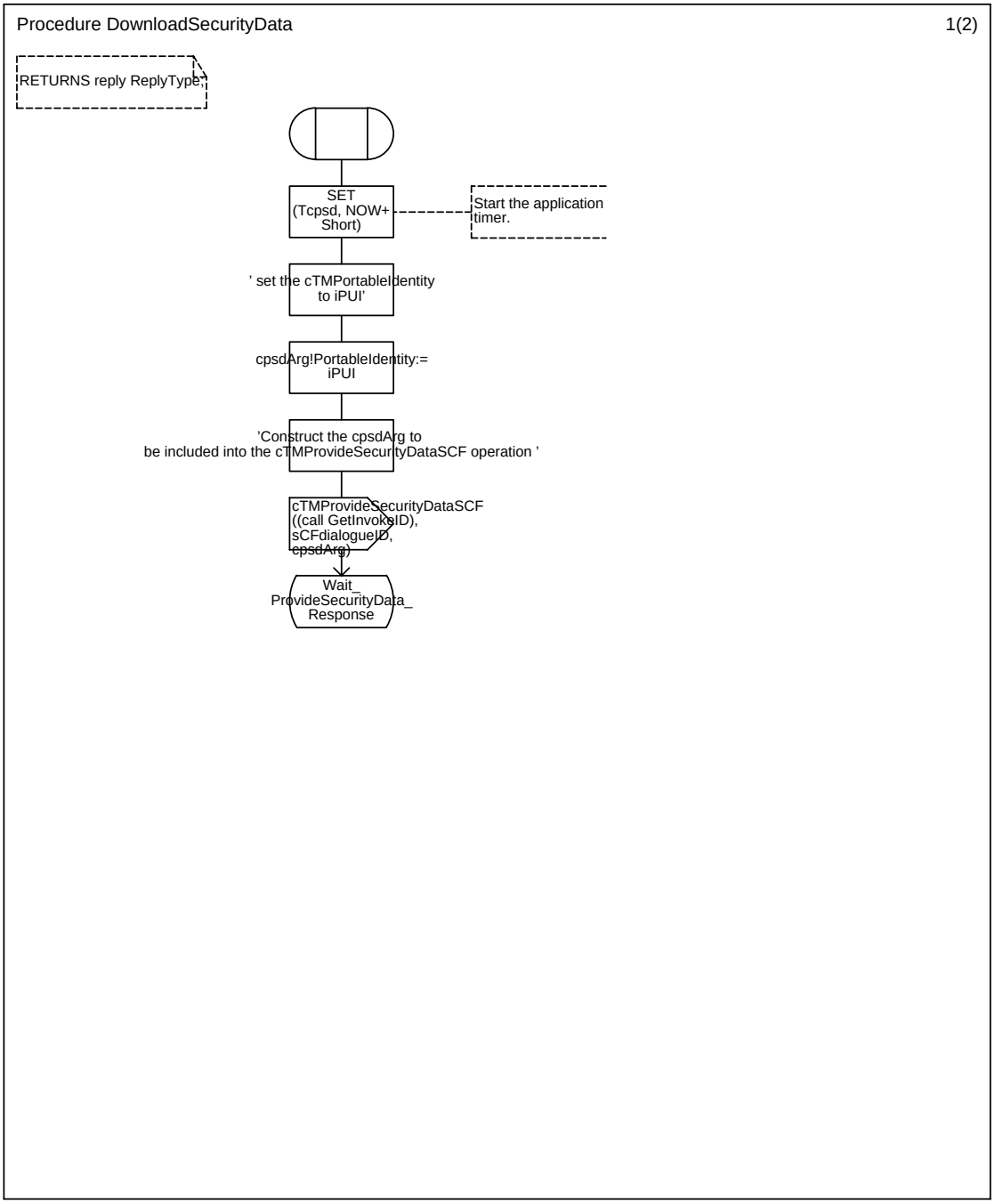

Procedure LocationRegistrationSCF
2(2)
The SCFsl shall check whether the CTM user is allowed to roam into the SCFmm area, if so a LocationUpdate return result shall be send with the CTM-Number and the type PARK.
:FPAR
RETURNS reply ReplyType;
If the CTM user is not allowed to roam into the SCFmm area on receipt of an LocationUpdate invoke operation the SCFsl shall send a LocationUpdate return error operation to the SCFmm using the ssi mechanism of the SCF-SCF interface
Wait_ LocationUpdateSCF_ Response
CTMLocationUpdateSCF (invokeID,dialogueID,cluRes)
CTMLocationUpdateSCF (invokeID,dialogueID,cluErr)
CTMRej (invokeID,ctmRej)
Handling of error cases.
'retrieve the cTMNumber and the cTMFixedIdentity (Containing the type PARK)'
'Check error values'
reply:=Successfull
reply:=failed
RESET (T_MM)
'Stop application timer'
Wait_ LocationUpdate_ Response
T_MM
reply:=failed

Annex C8 : Procedure LocationCancellationFT

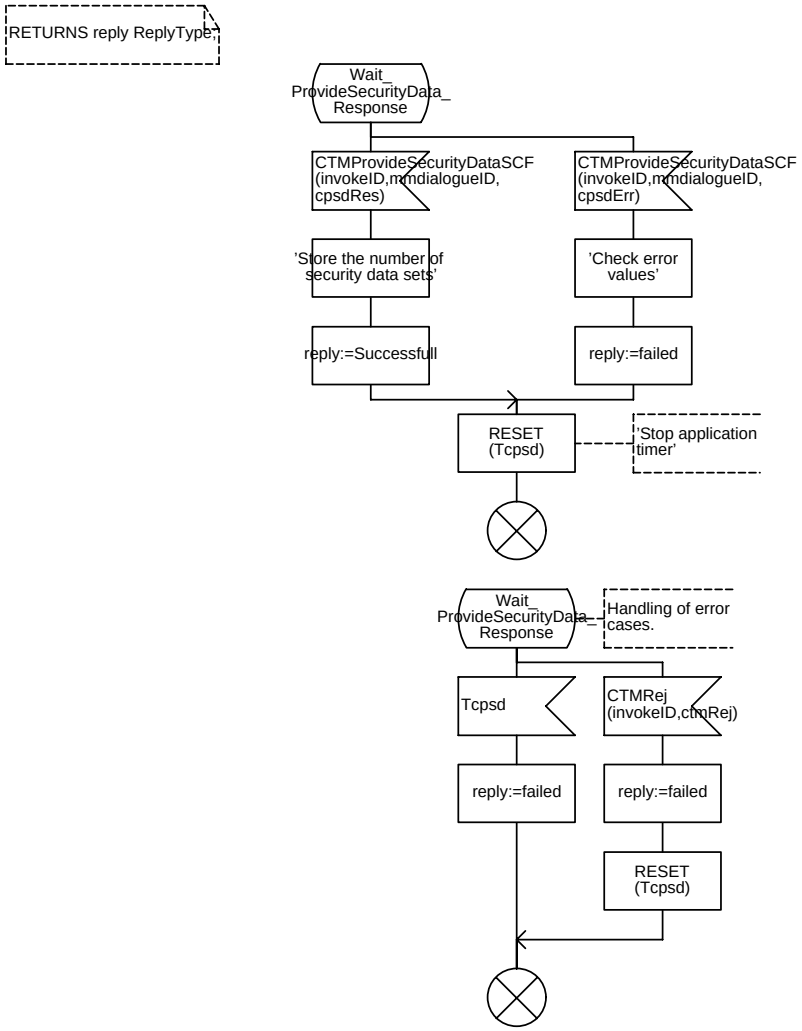



Procedure LocationCancellationFT
2(2)
;FPAR
RETURNS reply ReplyType;
When the FT receives a correctly encoded CTMLocationCancellation invoke component, the FT shall accept the provided information and shall respond to the network by sending a CTMLocationCancellation return result component without parameters.
Wait_ LocationCancellation_ Response
CTMLocationCancellation (invokeID,DialogueID,clcRes)
CTMLocationCancellation (invokeID,DialogueID,clcErr)
CTMRej (invokeID,ctmRej)
Handling of error cases.
CTMRej (invokeId,ctmRej, SCFVis)
'check error values'
reply:=Successfull
reply:=failed
If the FT is unable to act on the cTMLocationCancellation invoke component, the FT shall send a CTMLocationCancellation return error component to the network.
One of the following error values shall be included:
- portableIdentityUnknown,
- congestion,
- unspecified,

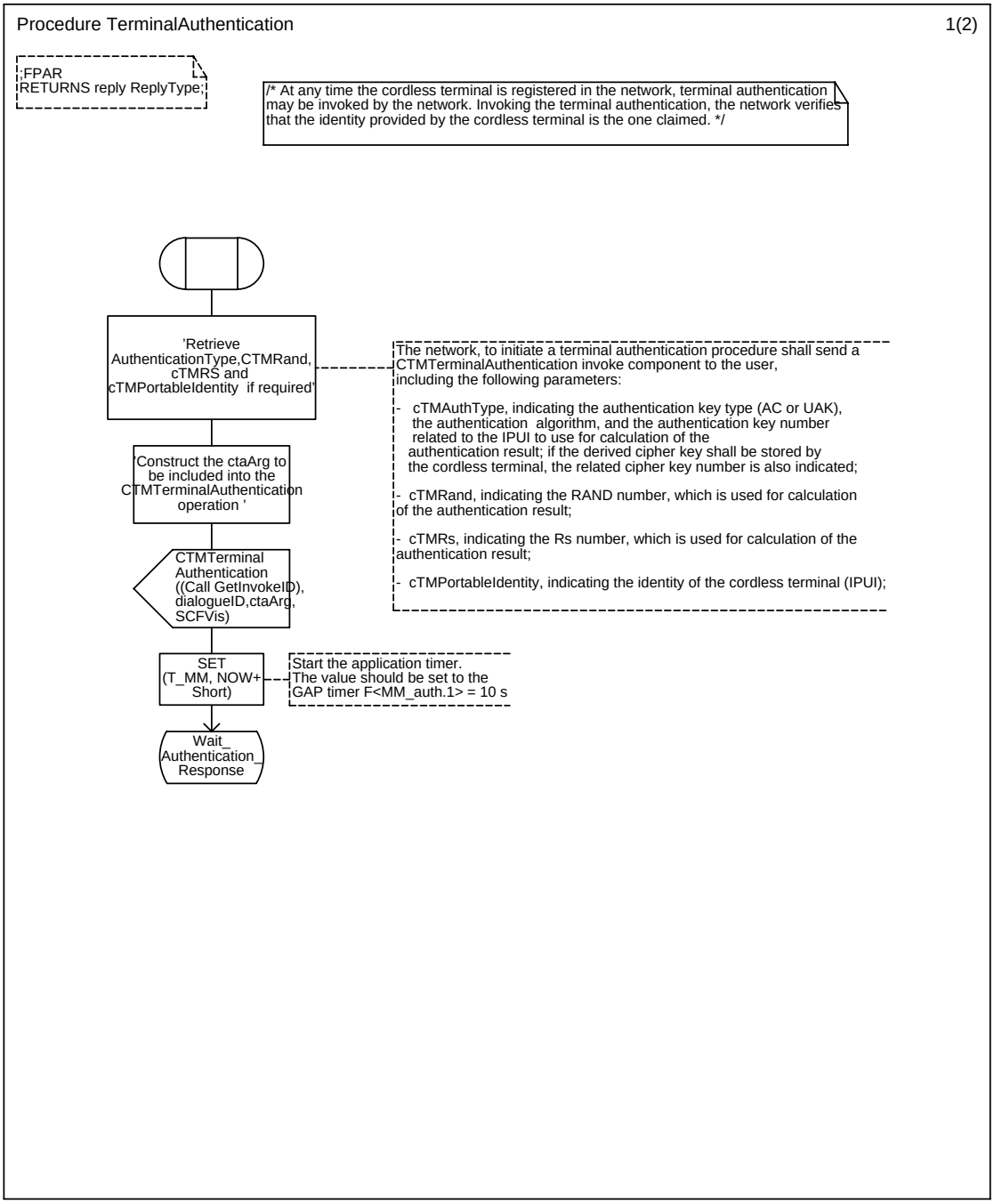
Annex C9 : Procedure DownloadSecurityData

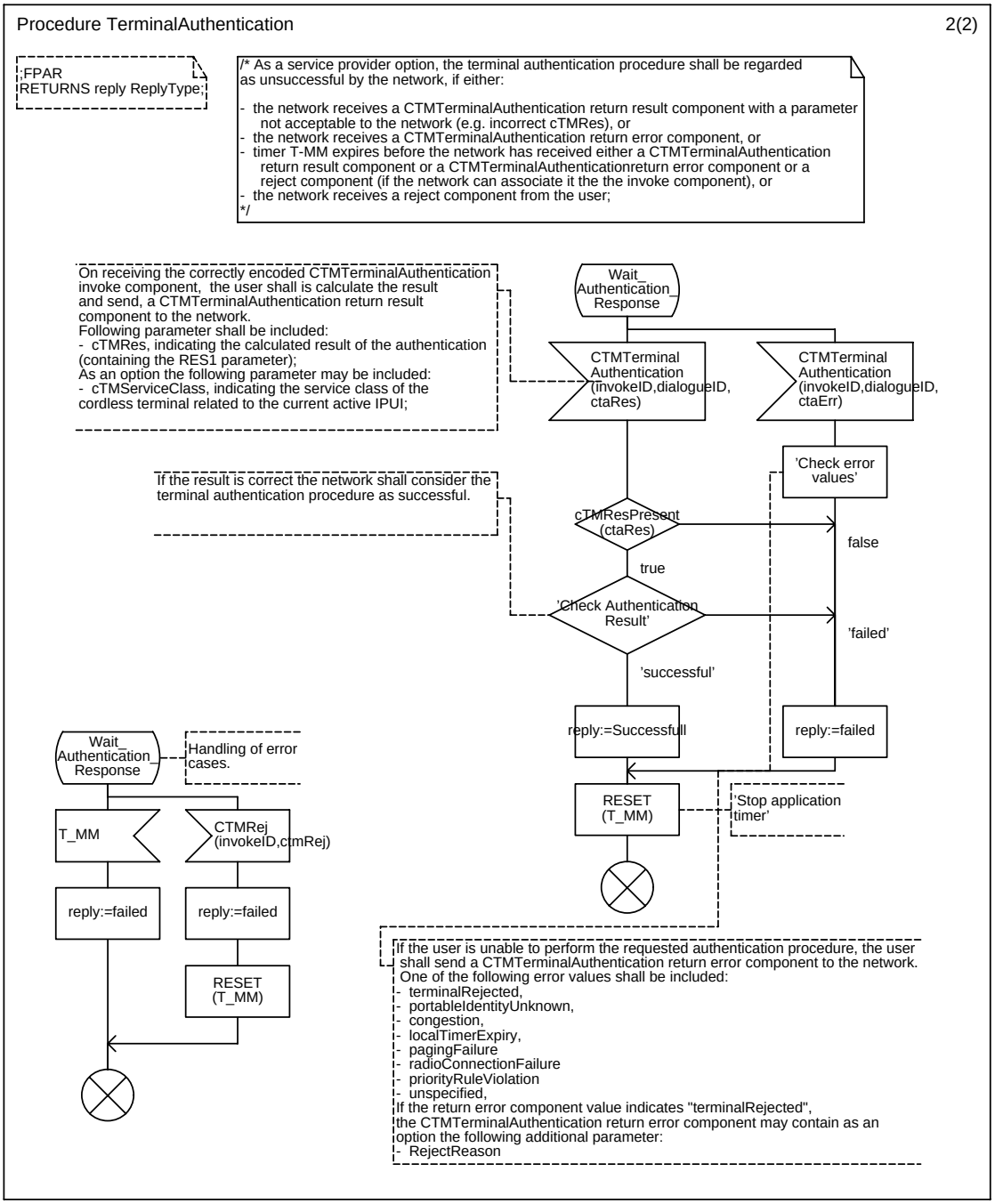


Procedure DownloadSecurityData 2(2)

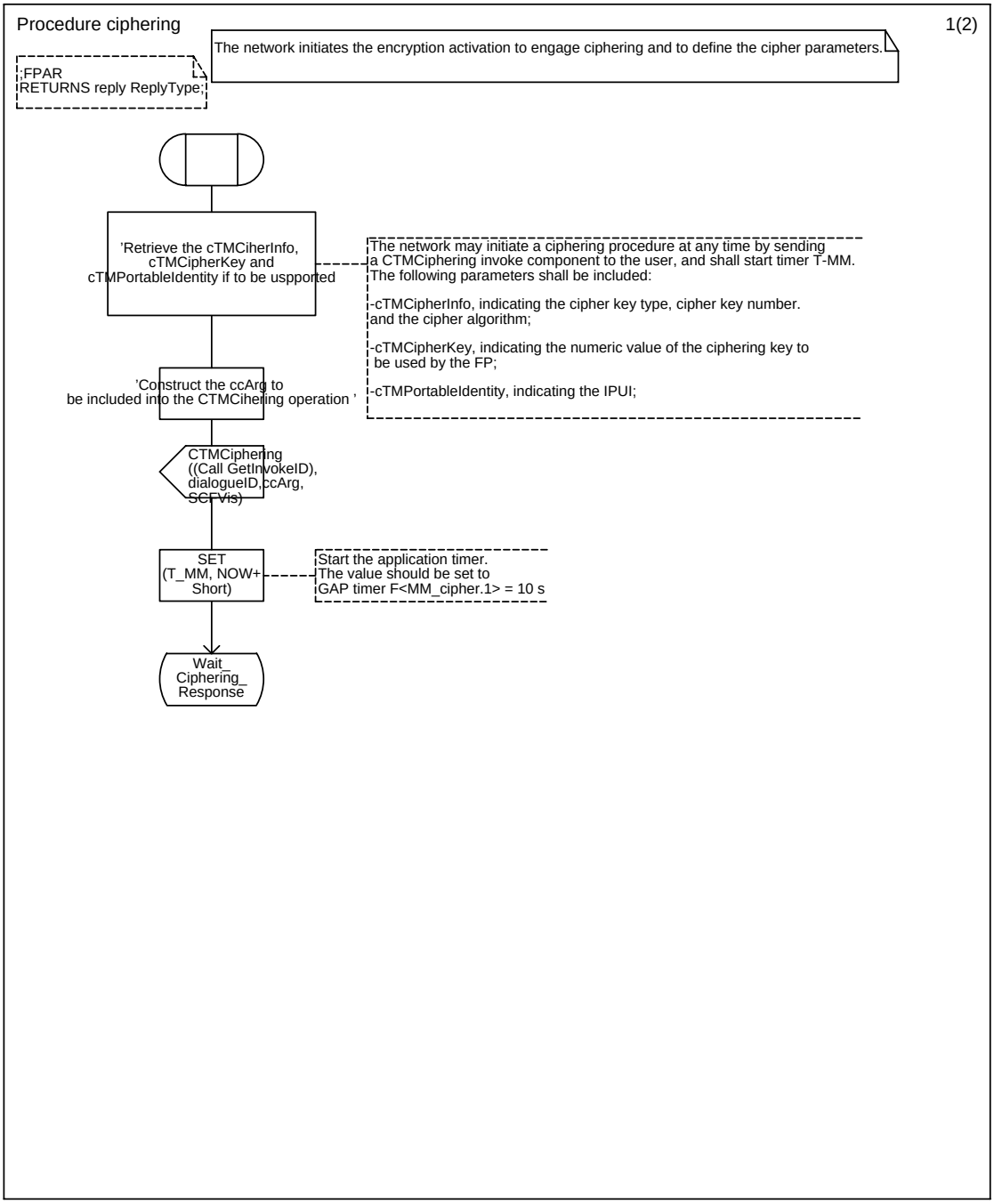


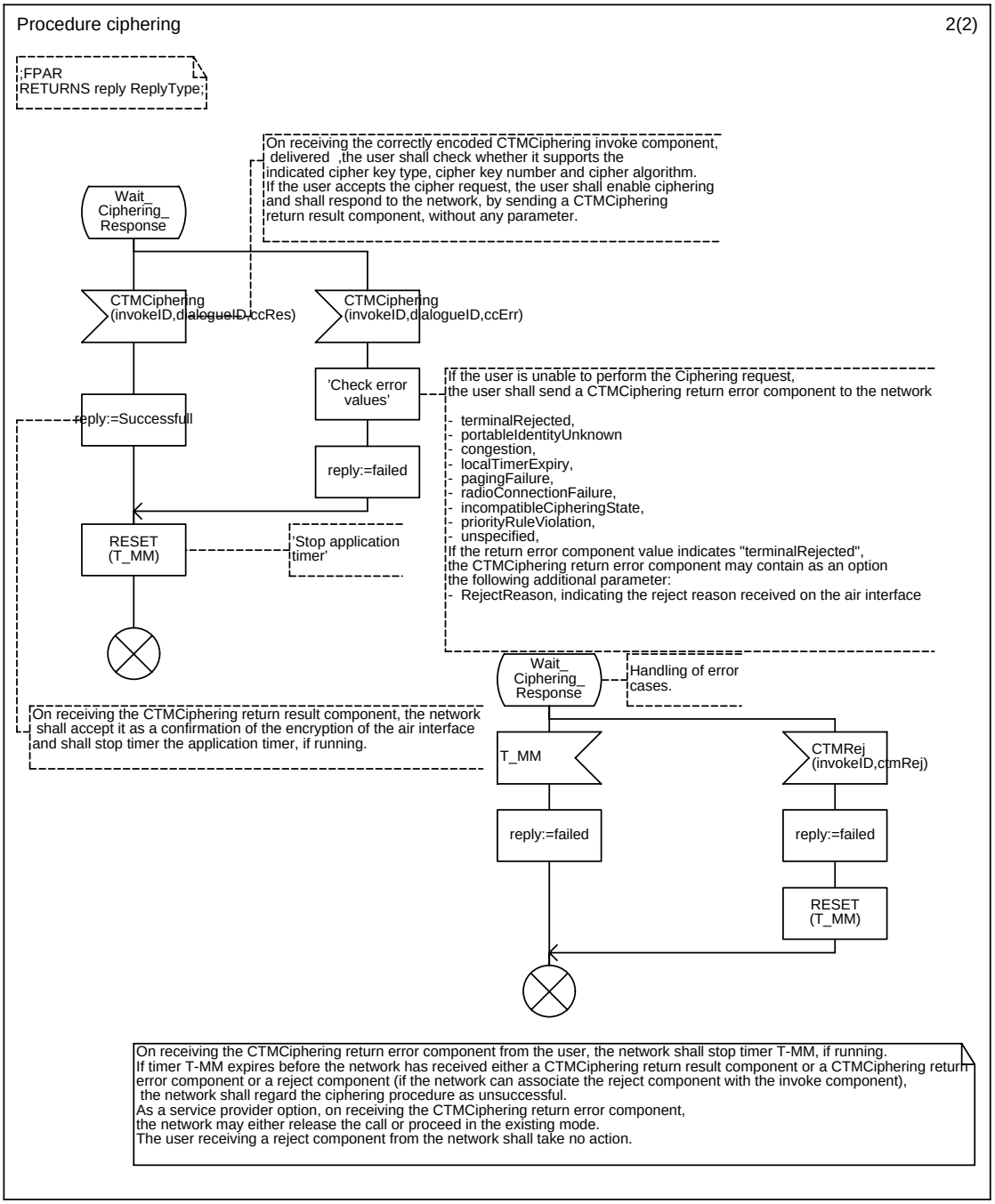
Annex C10 : Procedure TerminalAuthentication



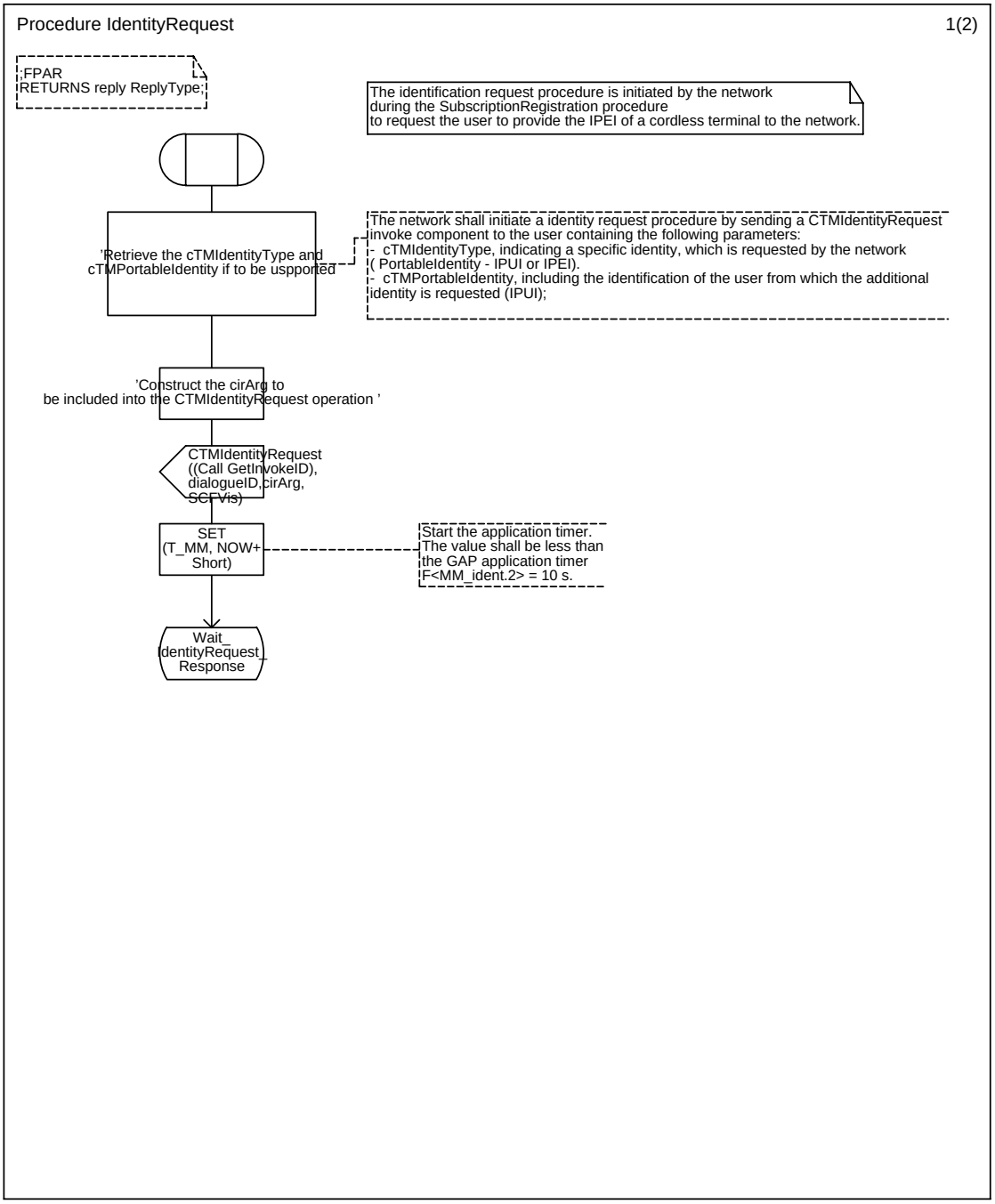

Procedure TerminalAuthentication
2(2)
;FPAR
RETURNS reply ReplyType;
/* As a service provider option, the terminal authentication procedure shall be regarded as unsuccessful by the network, if either:
- the network receives a CTMTerminalAuthentication return result component with a parameter not acceptable to the network (e.g. incorrect cTMRes), or
- the network receives a CTMTerminalAuthentication return error component, or
- timer T-MM expires before the network has received either a CTMTerminalAuthentication return result component or a CTMTerminalAuthenticationreturn error component or a reject component (if the network can associate it the the invoke component), or
- the network receives a reject component from the user;
*/
On receiving the correctly encoded CTMTerminalAuthentication invoke component, the user shall is calculate the result and send, a CTMTerminalAuthentication return result component to the network.
Following parameter shall be included:
- cTMRes, indicating the calculated result of the authentication (containing the RES1 parameter);
As an option the following parameter may be included:
- cTMServiceClass, indicating the service class of the cordless terminal related to the current active IPUI;
Wait_ Authentication_ Response
CTMTerminal Authentication (invokeID,dialogueID, ctaRes)
CTMTerminal Authentication (invokeID,dialogueID, ctaErr)
'Check error values'
If the result is correct the network shall consider the terminal authentication procedure as successful.
cTMResPresent (ctaRes)
false
true
'Check Authentication Result'
'failed'
'successful'
reply:=Successfull
reply:=failed
RESET (T_MM)
'Stop application timer'
Wait_ Authentication_ Response
Handling of error cases.
T_MM
CTMRej (invokeID,ctmRej)
reply:=failed
reply:=failed
RESET (T_MM)
If the user is unable to perform the requested authentication procedure, the user shall send a CTMTerminalAuthentication return error component to the network.
One of the following error values shall be included:
- terminalRejected,
- portableIdentityUnknown,
- congestion,
- localTimerExpiry,
- pagingFailure
- radioConnectionFailure
- priorityRuleViolation
- unspecified,
If the return error component value indicates "terminalRejected", the CTMTerminalAuthentication return error component may contain as an option the following additional parameter:
- RejectReason

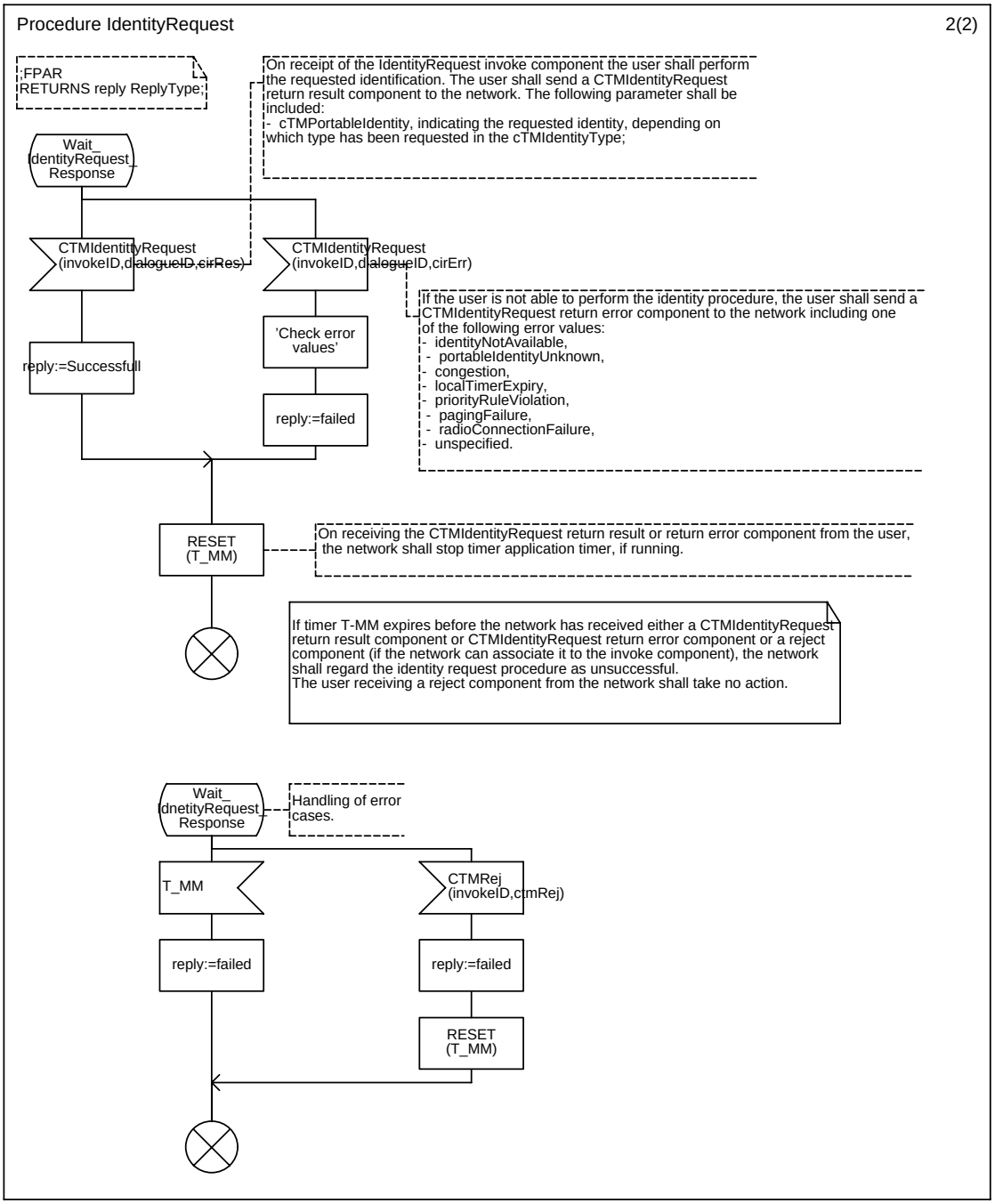
Annex C11 : Procedure Ciphering



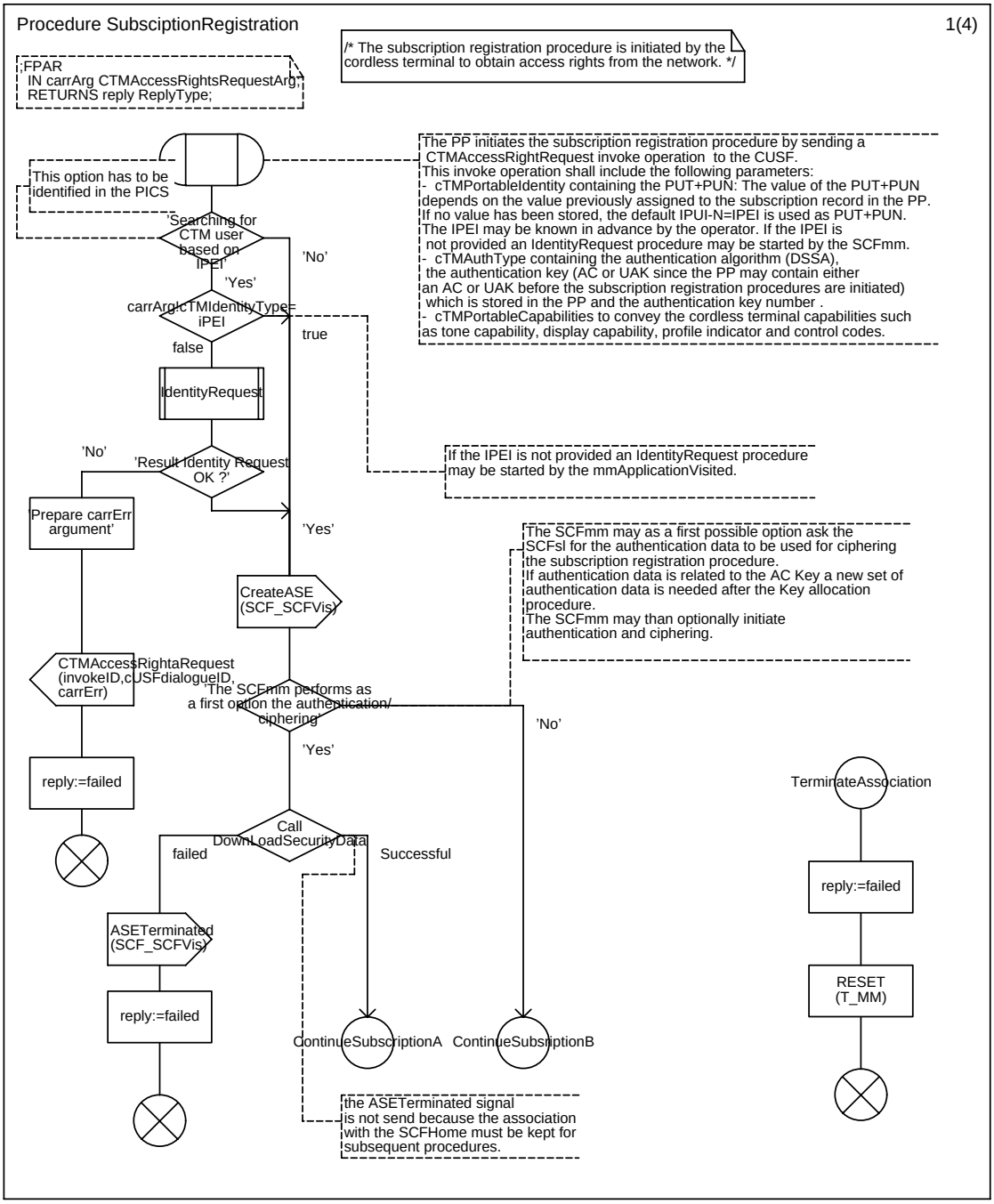

Procedure ciphering
2(2)
;FPAR
RETURNS reply ReplyType;
On receiving the correctly encoded CTMCiphering invoke component,
delivered ,the user shall check whether it supports the
indicated cipher key type, cipher key number and cipher algorithm.
If the user accepts the cipher request, the user shall enable ciphering
and shall respond to the network, by sending a CTMCiphering
return result component, without any parameter.
Wait_
Ciphering_
Response
CTMCiphering
(invokeID,dialogueID,ccRes)
CTMCiphering
(invokeID,dialogueID,ccErr)
'Check error
values'
If the user is unable to perform the Ciphering request,
the user shall send a CTMCiphering return error component to the network
- terminalRejected,
- portableIdentityUnknown
- congestion,
- localTimerExpiry,
- pagingFailure,
- radioConnectionFailure,
- incompatibleCipheringState,
- priorityRuleViolation,
- unspecified,
If the return error component value indicates "terminalRejected",
the CTMCiphering return error component may contain as an option
the following additional parameter:
- RejectReason, indicating the reject reason received on the air interface
reply:=Successfull
reply:=failed
RESET
(T_MM)
'Stop application
timer'
Wait_
Ciphering_
Response
Handling of error
cases.
On receiving the CTMCiphering return result component, the network
shall accept it as a confirmation of the encryption of the air interface
and shall stop timer the application timer, if running.
T_MM
CTMRej
(invokeID,ctmRej)
reply:=failed
reply:=failed
RESET
(T_MM)
On receiving the CTMCiphering return error component from the user, the network shall stop timer T-MM, if running.
If timer T-MM expires before the network has received either a CTMCiphering return result component or a CTMCiphering return
error component or a reject component (if the network can associate the reject component with the invoke component),
the network shall regard the ciphering procedure as unsuccessful.
As a service provider option, on receiving the CTMCiphering return error component,
the network may either release the call or proceed in the existing mode.
The user receiving a reject component from the network shall take no action.

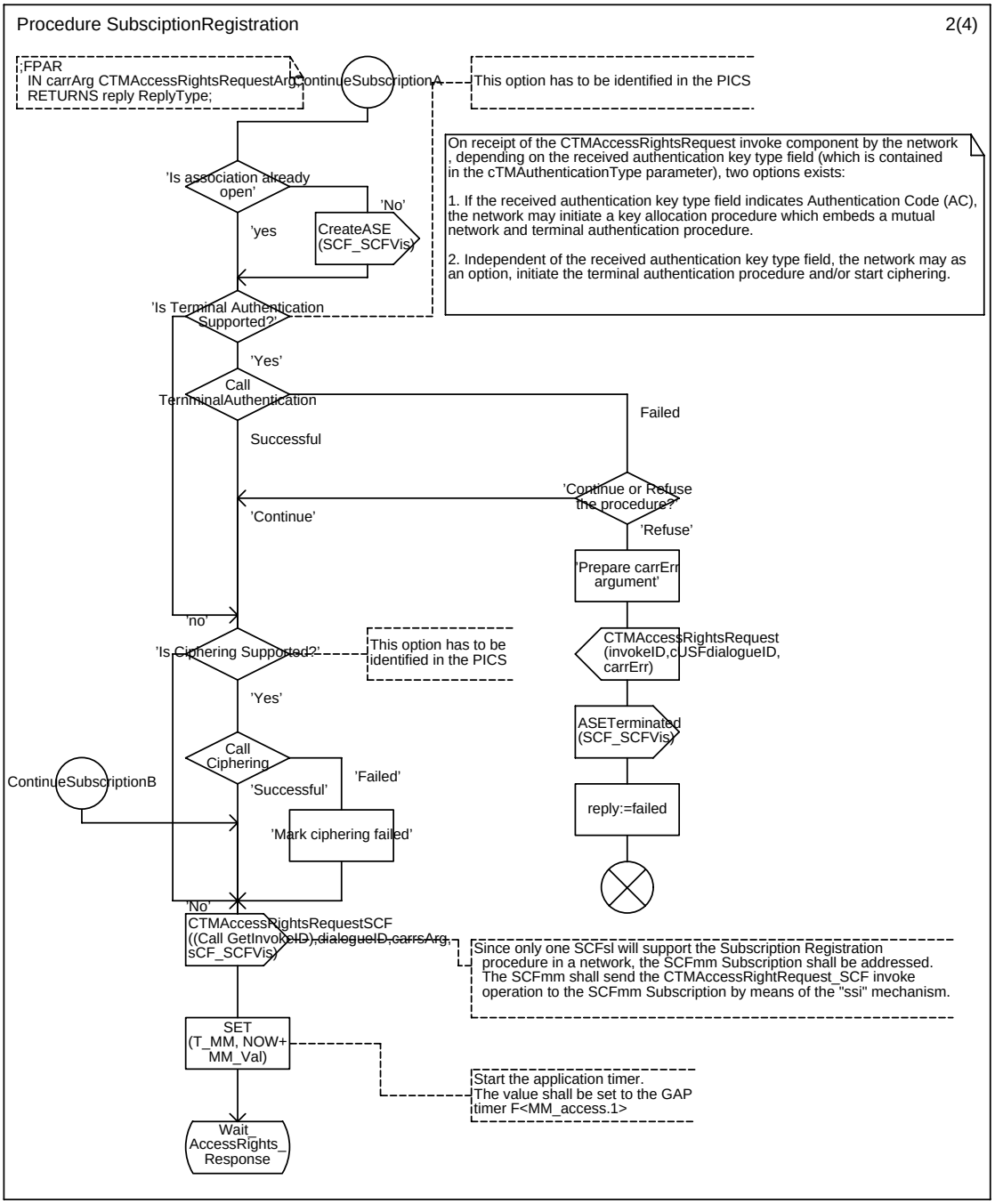
Annex C12 : Procedure IdentityRequest

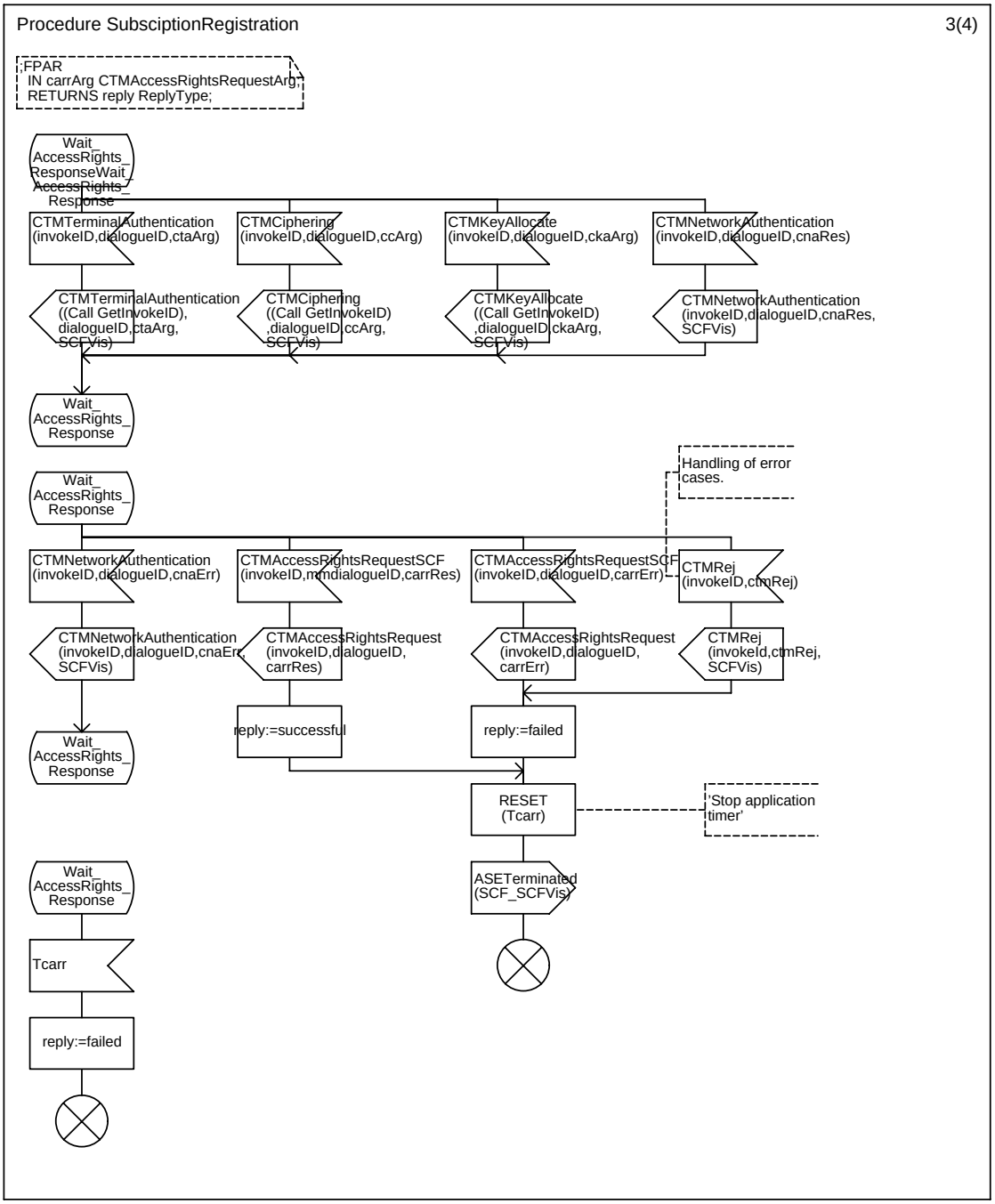


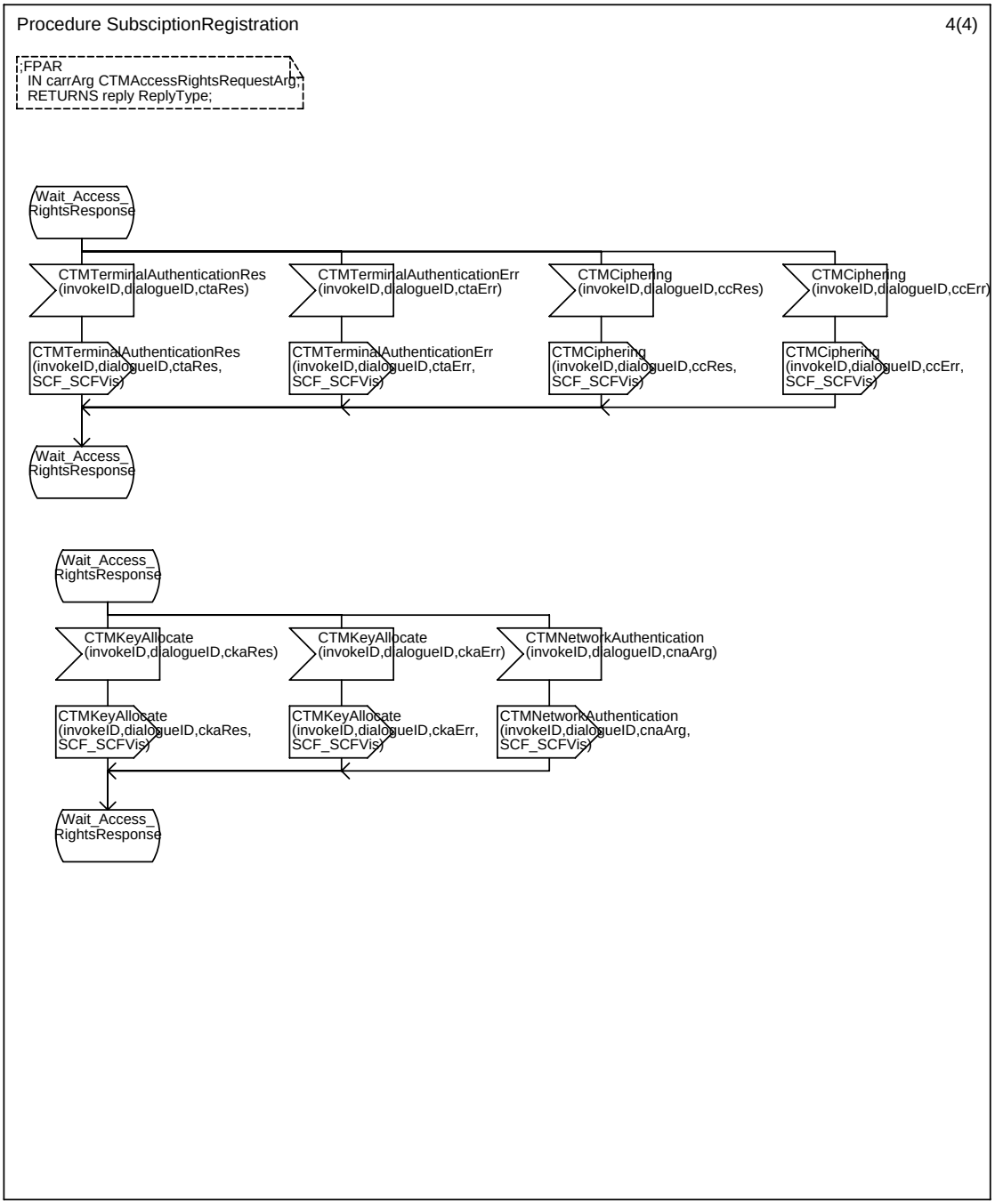

Procedure IdentityRequest
2(2)
;FPAR
RETURNS reply ReplyType;
On receipt of the IdentityRequest invoke component the user shall perform the requested identification. The user shall send a CTMIdentityRequest return result component to the network. The following parameter shall be included:
- cTMPortableIdentity, indicating the requested identity, depending on which type has been requested in the cTMIdentityType;
Wait_ IdentityRequest_ Response
CTMIdentityRequest (invokeID,dialogueID,cirRes)
CTMIdentityRequest (invokeID,dialogueID,cirErr)
If the user is not able to perform the identity procedure, the user shall send a CTMIdentityRequest return error component to the network including one of the following error values:
- identityNotAvailable,
- portableIdentityUnknown,
- congestion,
- localTimerExpiry,
- priorityRuleViolation,
- pagingFailure,
- radioConnectionFailure,
- unspecified.
'Check error values'
reply:=Successfull
reply:=failed
RESET (T_MM)
On receiving the CTMIdentityRequest return result or return error component from the user, the network shall stop timer application timer, if running.
If timer T-MM expires before the network has received either a CTMIdentityRequest return result component or CTMIdentityRequest return error component or a reject component (if the network can associate it to the invoke component), the network shall regard the identity request procedure as unsuccessful.
The user receiving a reject component from the network shall take no action.
Wait_ IdnetityRequest_ Response
Handling of error cases.
T_MM
CTMRej (invokeID,ctmRej)
reply:=failed
reply:=failed
RESET (T_MM)

Annex C13 : Procedure SubscriptionRegistration

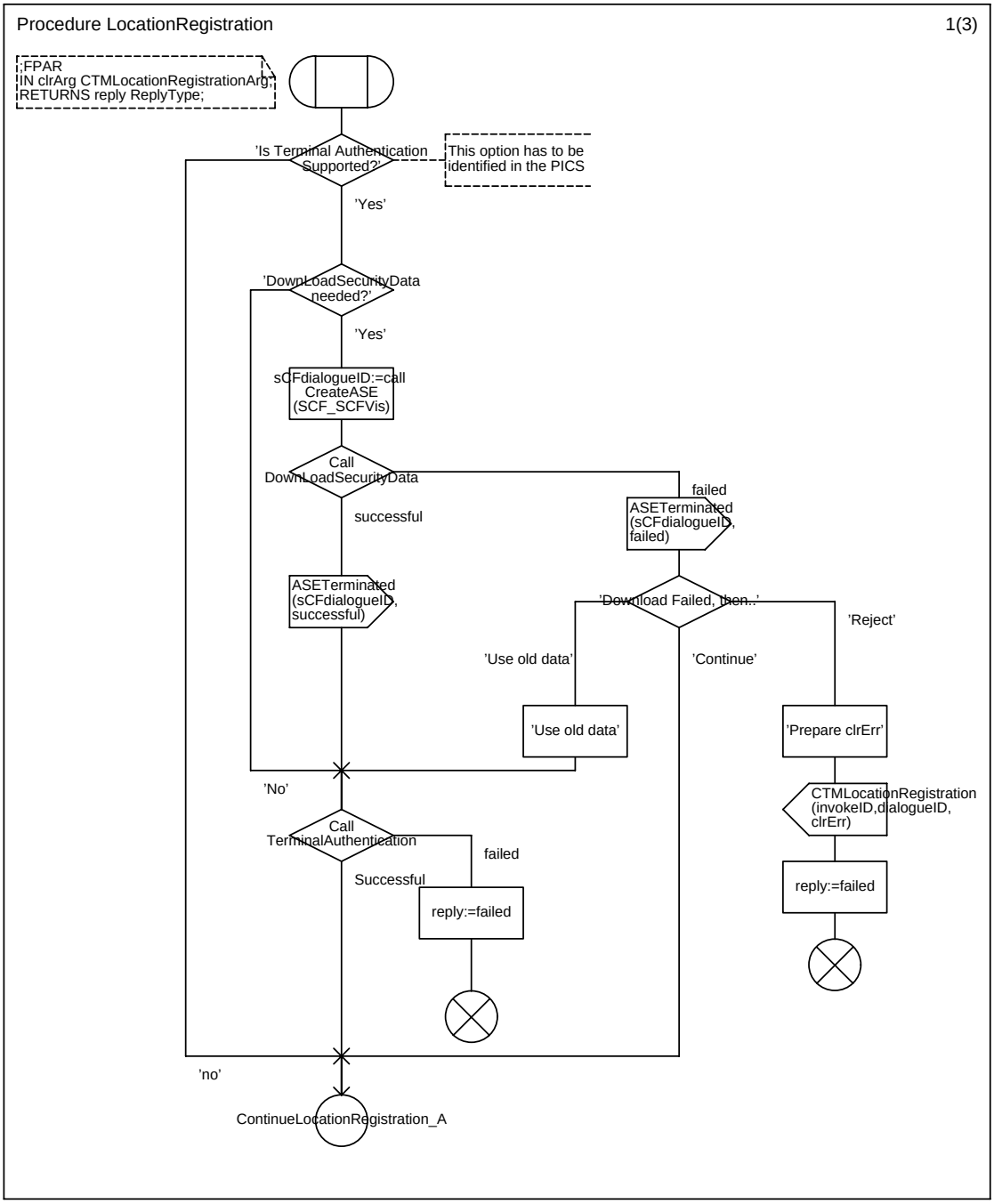


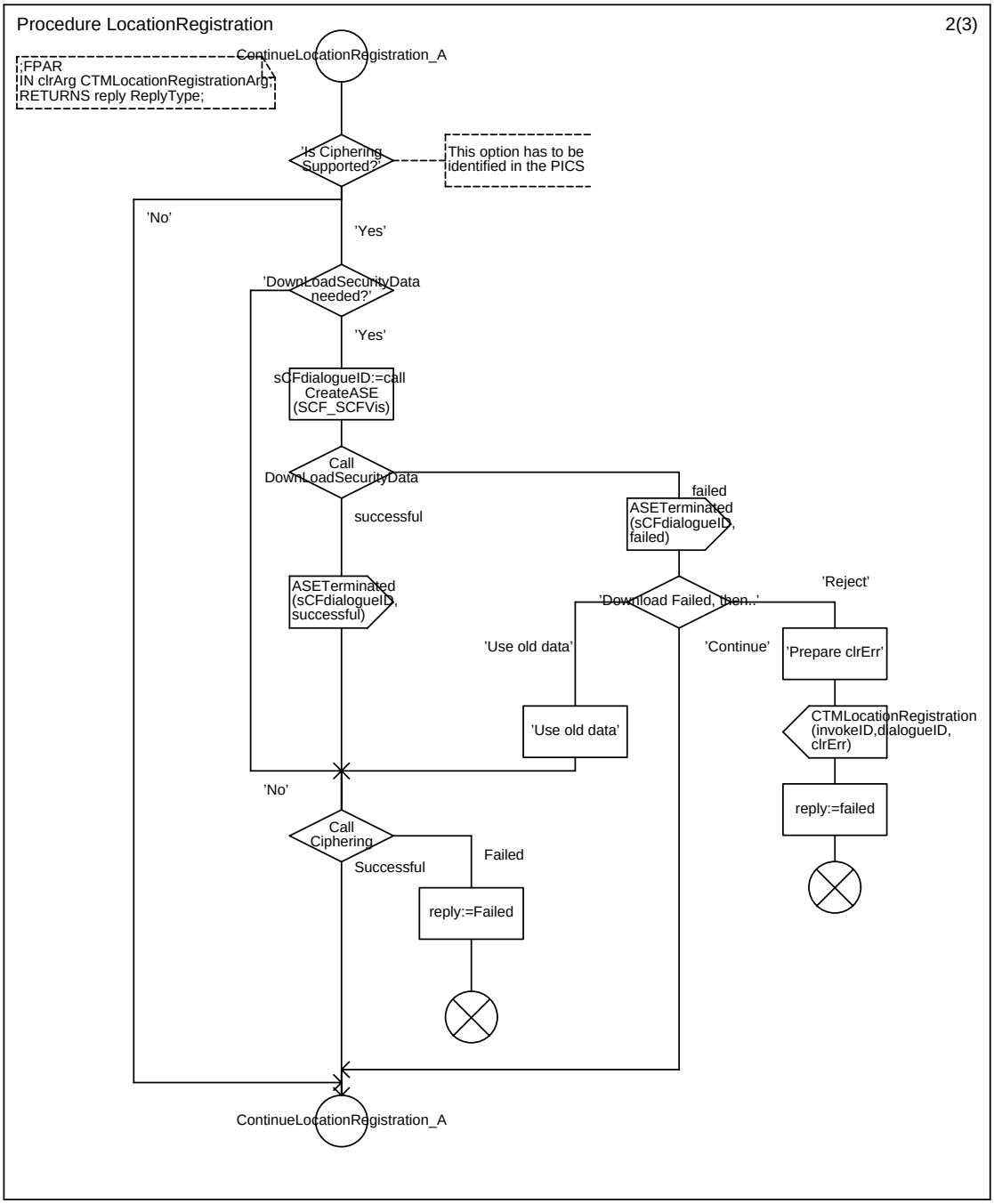

Procedure SubsciptionRegistration
2(4)
;FPAR
IN carrArg CTMAccessRightsRequestArg,
RETURNS reply ReplyType;
ContinueSubscriptionA
This option has to be identified in the PICS
On receipt of the CTMAccessRightsRequest invoke component by the network , depending on the received authentication key type field (which is contained in the cTMAuthenticationType parameter), two options exists:
1. If the received authentication key type field indicates Authentication Code (AC), the network may initiate a key allocation procedure which embeds a mutual network and terminal authentication procedure.
2. Independent of the received authentication key type field, the network may as an option, initiate the terminal authentication procedure and/or start ciphering.
'Is association already open'
'No'
'yes'
CreateASE (SCF_SCFVis)
'Is Terminal Authentication Supported?'
'Yes'
Call TerminalAuthentication
Failed
Successful
'Continue or Refuse the procedure?'
'Continue'
'Refuse'
'Prepare carrErr argument'
'no'
'Is Ciphering Supported?'
This option has to be identified in the PICS
CTMAccessRightsRequest (invokeID,cUSFdialogueID, carrErr)
'Yes'
ASETerminated (SCF_SCFVis)
Call Ciphering
ContinueSubscriptionB
'Failed'
'Successful'
reply:=failed
'Mark ciphering failed'
'No'
CTMAccessRightsRequestSCF ((Call GetInvokeID),dialogueID,carrsArg, sCF_SCFVis)
Since only one SCFsl will support the Subscription Registration procedure in a network, the SCFmm Subscription shall be addressed. The SCFmm shall send the CTMAccessRightRequest_SCF invoke operation to the SCFmm Subscription by means of the "ssi" mechanism.
SET (T_MM, NOW+ MM_Val)
Start the application timer.
The value shall be set to the GAP timer F<MM_access.1>
Wait_ AccessRights_ Response

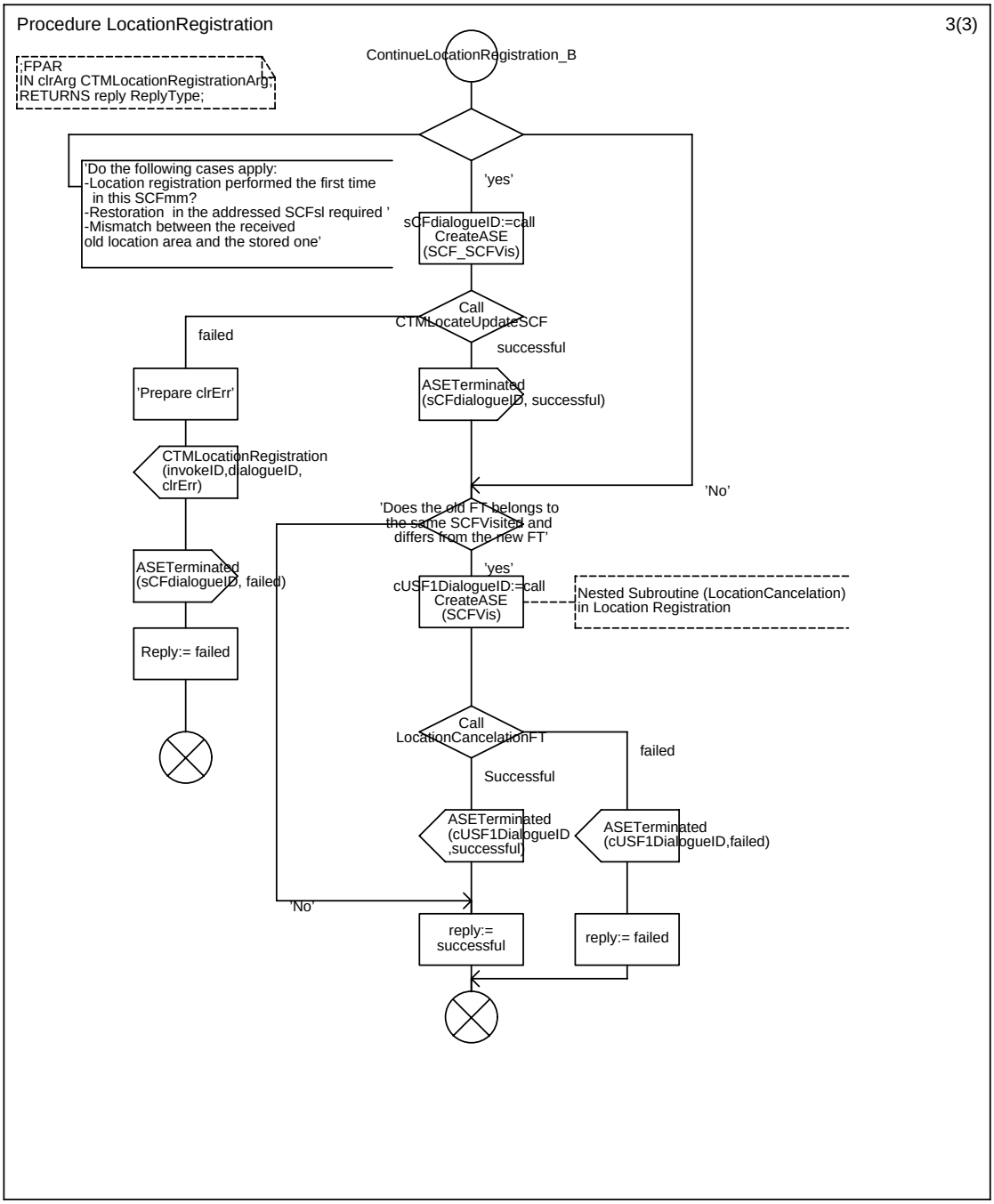

Procedure SubsciptionRegistration
3(4)
;FPAR
IN carrArg CTMAccessRightsRequestArg,
RETURNS reply ReplyType;
Wait_ AccessRights_ ResponseWait_ AccessRights_ Response
CTMTerminalAuthentication (invokeID,dialogueID,ctaArg)
CTMCiphering (invokeID,dialogueID,ccArg)
CTMKeyAllocate (invokeID,dialogueID,ckaArg)
CTMNetworkAuthentication (invokeID,dialogueID,cnaRes)
CTMTerminalAuthentication ((Call GetInvokeID), dialogueID,ctaArg, SCFVis)
CTMCiphering ((Call GetInvokeID) ,dialogueID,ccArg, SCFVis)
CTMKeyAllocate ((Call GetInvokeID) ,dialogueID,ckaArg, SCFVis)
CTMNetworkAuthentication (invokeID,dialogueID,cnaRes, SCFVis)
Wait_ AccessRights_ Response
Handling of error cases.
Wait_ AccessRights_ Response
CTMNetworkAuthentication (invokeID,dialogueID,cnaErr)
CTMAccessRightsRequestSCF (invokeID,mmdialogueID,carrRes)
CTMAccessRightsRequestSCF (invokeID,dialogueID,carrErr)
CTMRej (invokeID,ctmRej)
CTMNetworkAuthentication (invokeID,dialogueID,cnaErr, SCFVis)
CTMAccessRightsRequest (invokeID,dialogueID, carrRes)
CTMAccessRightsRequest (invokeID,dialogueID, carrErr)
CTMRej (invokeId,ctmRej, SCFVis)
Wait_ AccessRights_ Response
reply:=successful
reply:=failed
RESET (Tcarr)
'Stop application timer'
Wait_ AccessRights_ Response
ASETerminated (SCF_SCFVis)
Tcarr
reply:=failed


Procedure SubsciptionRegistration
4(4)
;FPAR
IN carrArg CTMAccessRightsRequestArg,
RETURNS reply ReplyType;
Wait_Access_
RightsResponse
CTMTerminalAuthenticationRes
(invokeID,dialogueID,ctaRes)
CTMTerminalAuthenticationErr
(invokeID,dialogueID,ctaErr)
CTMCiphering
(invokeID,dialogueID,ccRes)
CTMCiphering
(invokeID,dialogueID,ccErr)
CTMTerminalAuthenticationRes
(invokeID,dialogueID,ctaRes,
SCF_SCFVis)
CTMTerminalAuthenticationErr
(invokeID,dialogueID,ctaErr,
SCF_SCFVis)
CTMCiphering
(invokeID,dialogueID,ccRes,
SCF_SCFVis)
CTMCiphering
(invokeID,dialogueID,ccErr,
SCF_SCFVis)
Wait_Access_
RightsResponse
Wait_Access_
RightsResponse
CTMKeyAllocate
(invokeID,dialogueID,ckaRes)
CTMKeyAllocate
(invokeID,dialogueID,ckaErr)
CTMNetworkAuthentication
(invokeID,dialogueID,cnaArg)
CTMKeyAllocate
(invokeID,dialogueID,ckaRes,
SCF_SCFVis)
CTMKeyAllocate
(invokeID,dialogueID,ckaErr,
SCF_SCFVis)
CTMNetworkAuthentication
(invokeID,dialogueID,cnaArg,
SCF_SCFVis)
Wait_Access_
RightsResponse

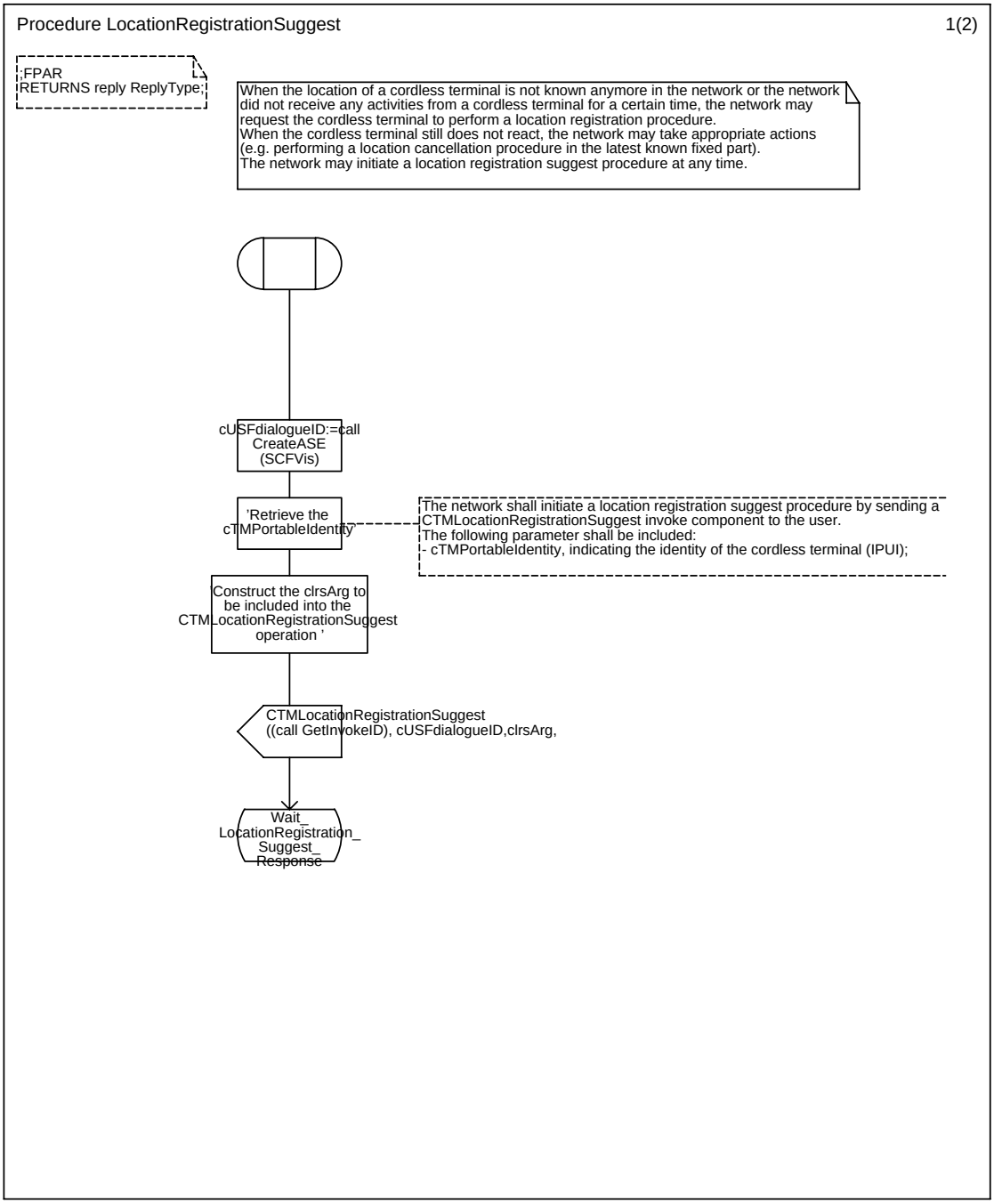
Annex C14 : Procedure LocationRegistration

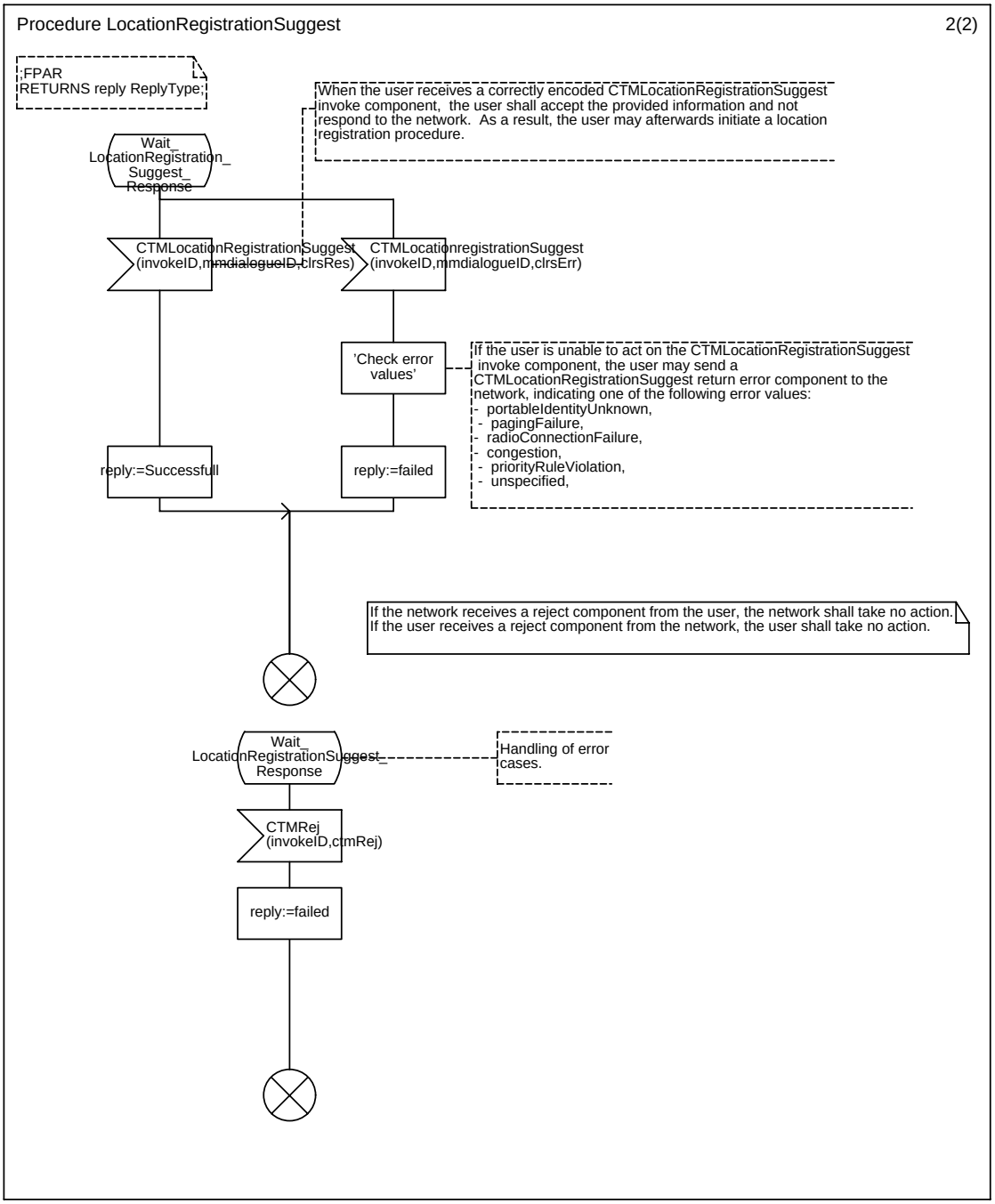



Procedure LocationRegistration
2(3)
ContinueLocationRegistration_A
;FPAR
IN clrArg CTMLocationRegistrationArg;
RETURNS reply ReplyType;
'Is Ciphering Supported?'
This option has to be identified in the PICS
'No'
'Yes'
'DownLoadSecurityData needed?'
'Yes'
sCFdialogueID:=call CreateASE (SCF_SCFVis)
Call DownLoadSecurityData
failed
ASETerminated (sCFdialogueID, failed)
successful
ASETerminated (sCFdialogueID, successful)
'Download Failed, then..'
'Reject'
'Use old data'
'Continue'
'Prepare clrErr'
'Use old data'
CTMLocationRegistration (invokeID,dialogueID, clrErr)
'No'
reply:=failed
Call Ciphering
Failed
Successful
reply:=Failed
ContinueLocationRegistration_A

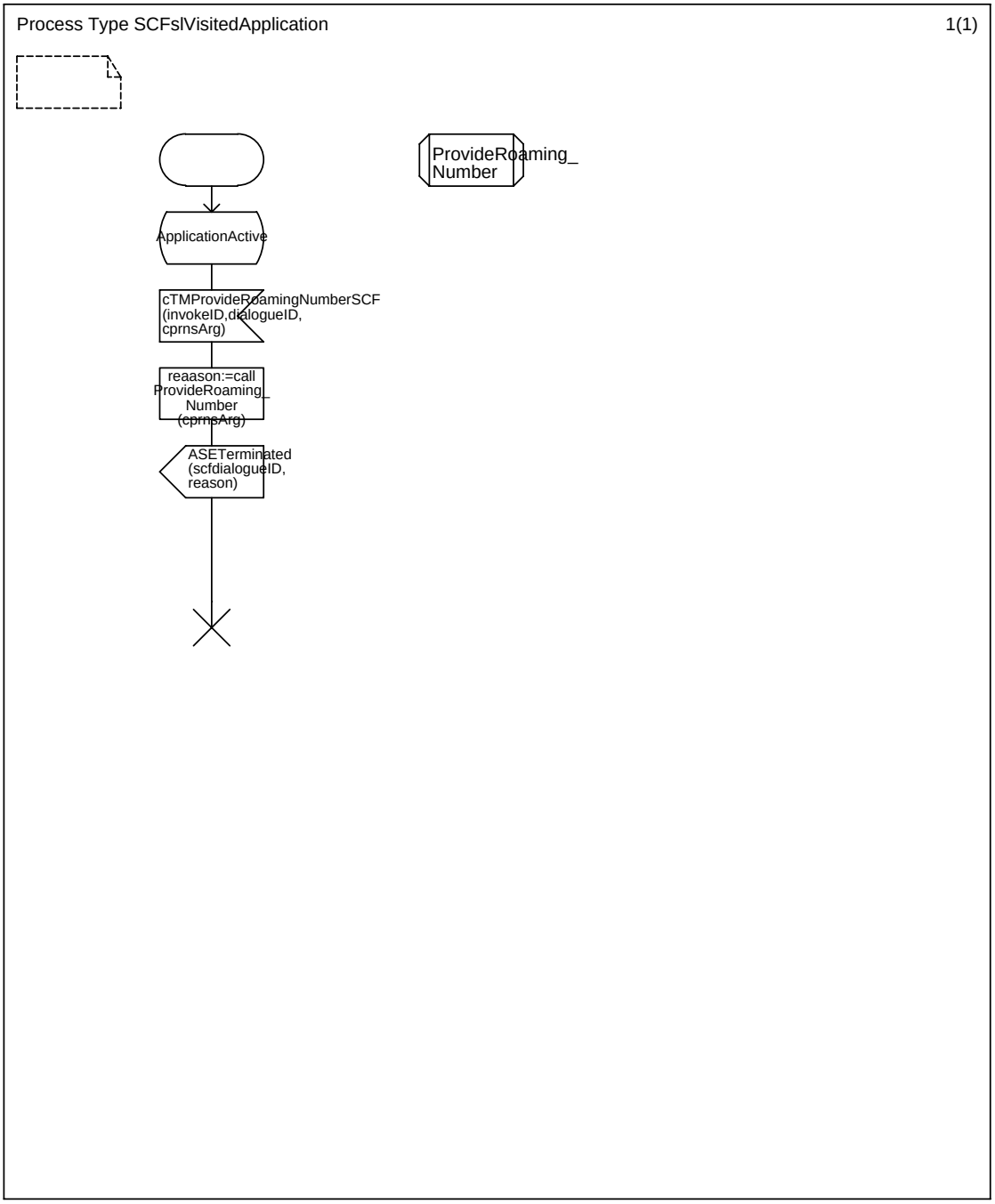

Procedure LocationRegistration
3(3)
ContinueLocationRegistration_B
;FPAR
IN clrArg CTMLocationRegistrationArg;
RETURNS reply ReplyType;
'Do the following cases apply:
-Location registration performed the first time
in this SCFmm?
-Restoration in the addressed SCFsl required '
-Mismatch between the received
old location area and the stored one'
'yes'
sCFdialogueID:=call
CreateASE
(SCF_SCFVis)
Call
CTMLocateUpdateSCF
failed
successful
'Prepare clrErr'
ASETerminated
(sCFdialogueID, successful)
CTMLocationRegistration
(invokeID,dialogueID,
clrErr)
'No'
'Does the old FT belongs to
the same SCFVisited and
differs from the new FT'
ASETerminated
(sCFdialogueID, failed)
'yes'
cUSF1DialogueID:=call
CreateASE
(SCFVis)
Nested Subroutine (LocationCancelation)
in Location Registration
Reply:= failed
Call
LocationCancelationFT
failed
Successful
ASETerminated
(cUSF1DialogueID
,successful)
ASETerminated
(cUSF1DialogueID,failed)
'No'
reply:=
successful
reply:= failed

Annex C15 : Procedure LocationRegistrationSuggest

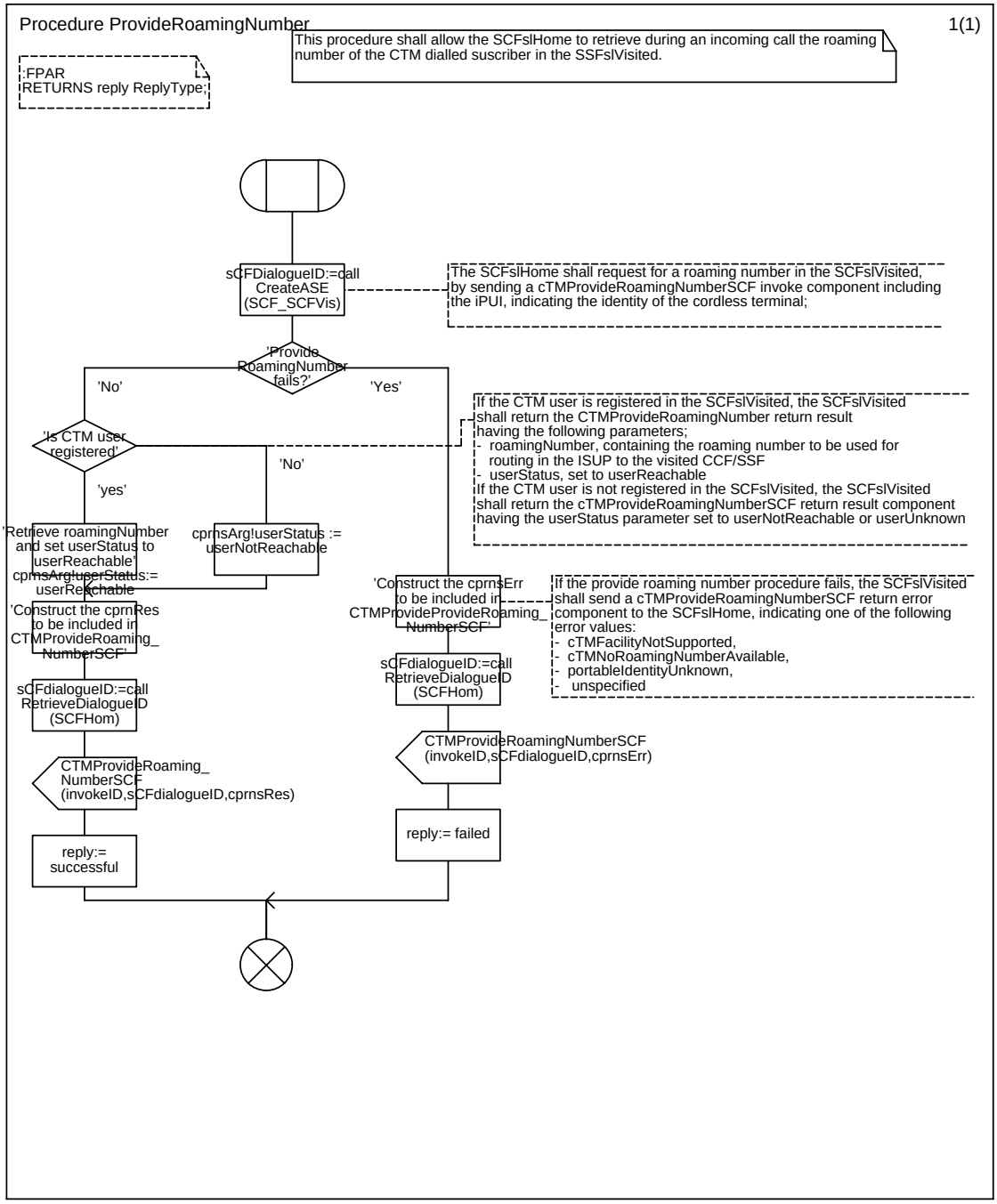



Procedure LocationRegistrationSuggest
2(2)
:FPAR
RETURNS reply ReplyType;
When the user receives a correctly encoded CTMLocationRegistrationSuggest invoke component, the user shall accept the provided information and not respond to the network. As a result, the user may afterwards initiate a location registration procedure.
Wait_ LocationRegistration_ Suggest_ Response
CTMLocationRegistrationSuggest (invokeID,mmdialogueID,clrsRes)
CTMLocationregistrationSuggest (invokeID,mmdialogueID,clrsErr)
'Check error values'
If the user is unable to act on the CTMLocationRegistrationSuggest invoke component, the user may send a CTMLocationRegistrationSuggest return error component to the network, indicating one of the following error values:
- portableIdentityUnknown,
- pagingFailure,
- radioConnectionFailure,
- congestion,
- priorityRuleViolation,
- unspecified,
reply:=Successfull
reply:=failed
If the network receives a reject component from the user, the network shall take no action.
If the user receives a reject component from the network, the user shall take no action.
Wait_ LocationRegistrationSuggest_ Response
Handling of error cases.
CTMRej (invokeID,ctmRej)
reply:=failed

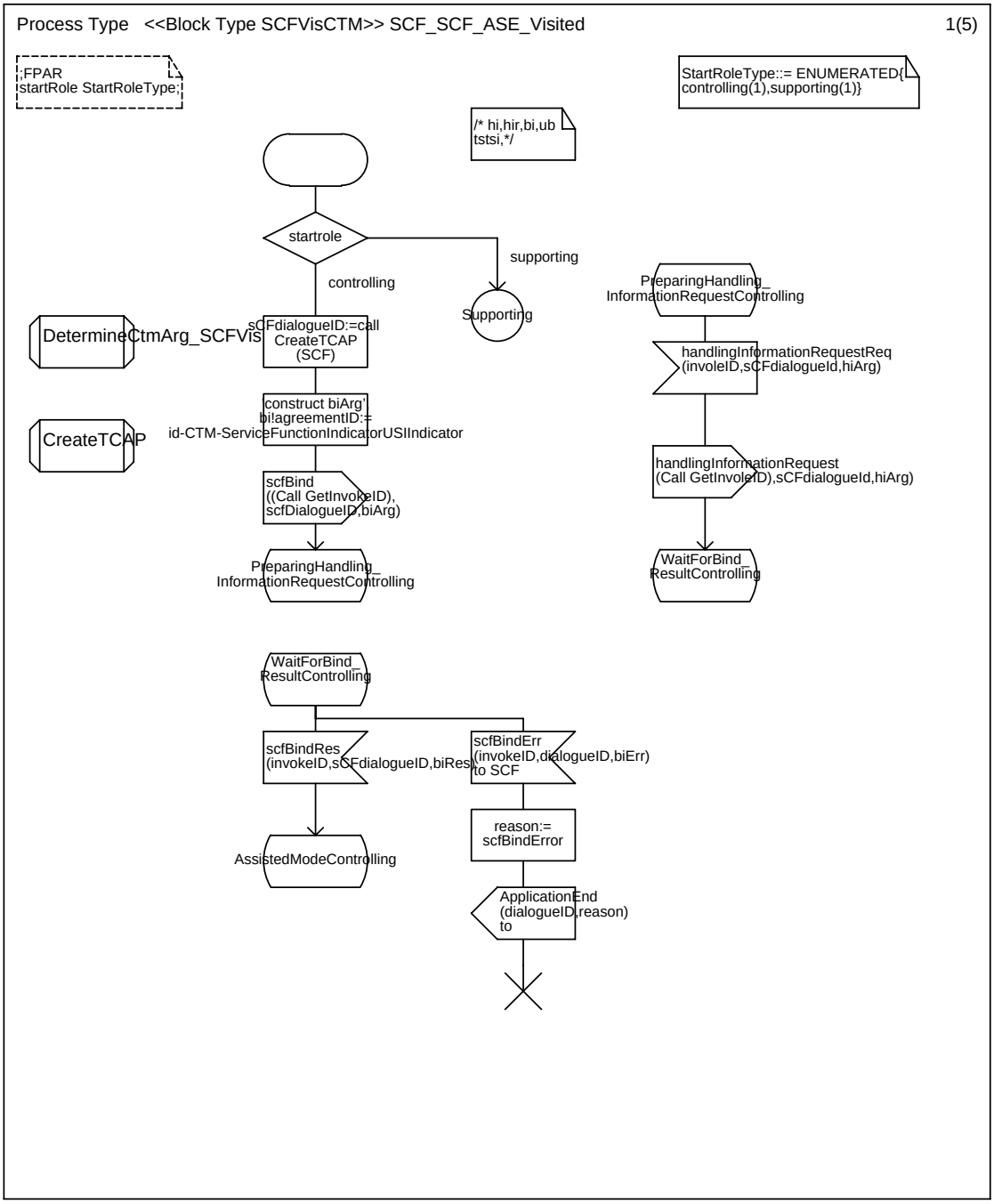
Annex C16 : Process Type SCFslVisitedApplication



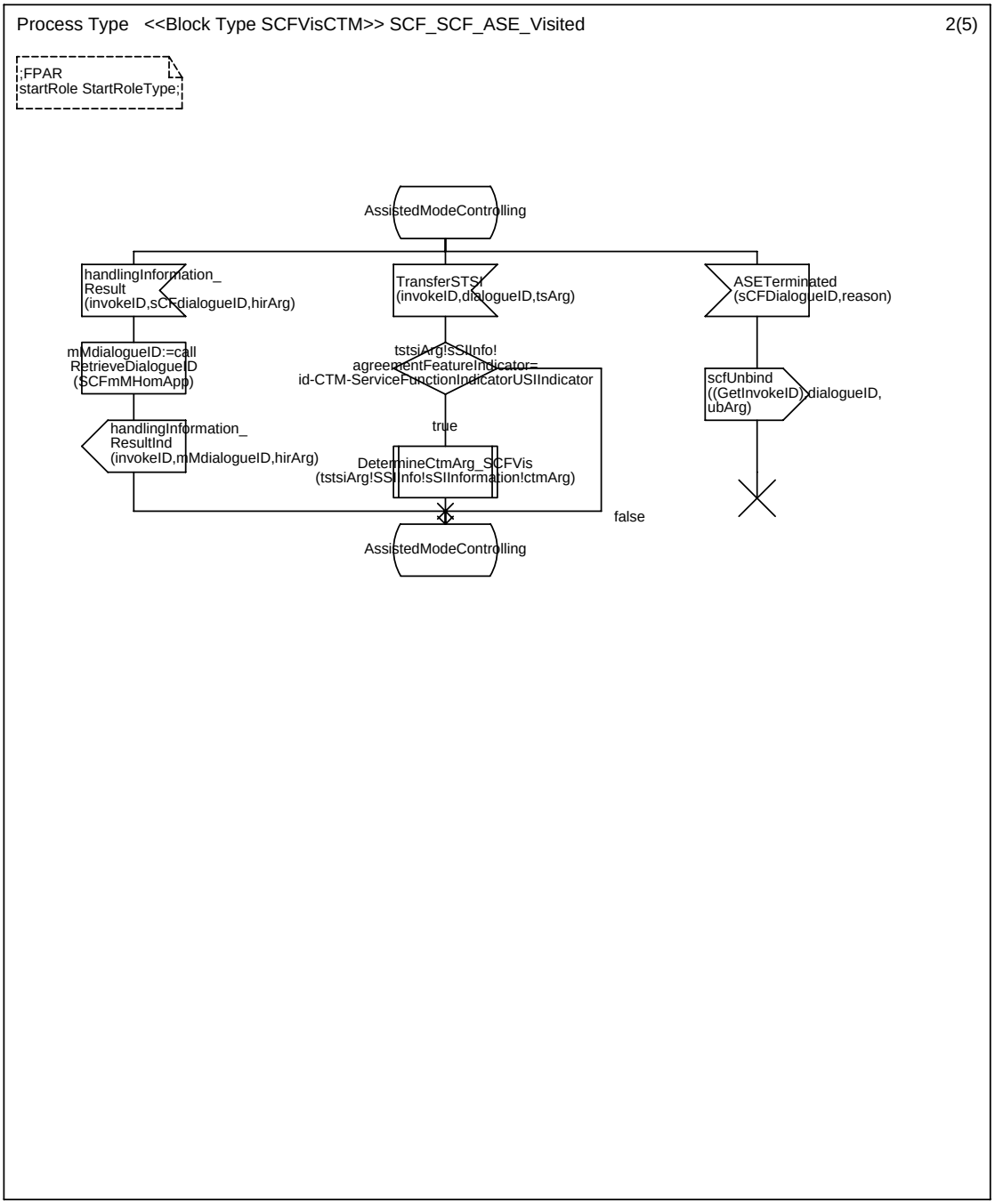
Annex C17 : Procedure ProvideRoamingNumber

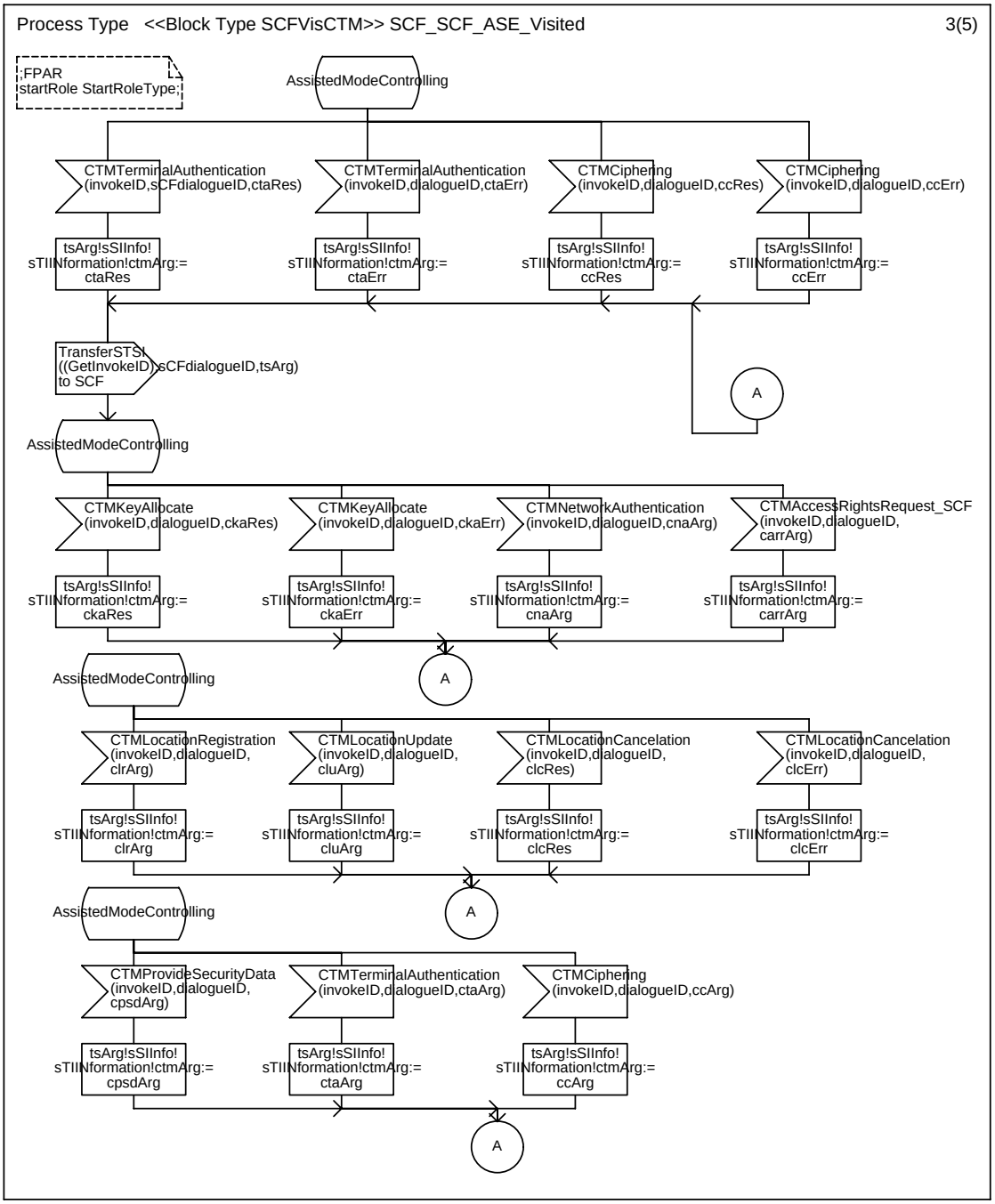


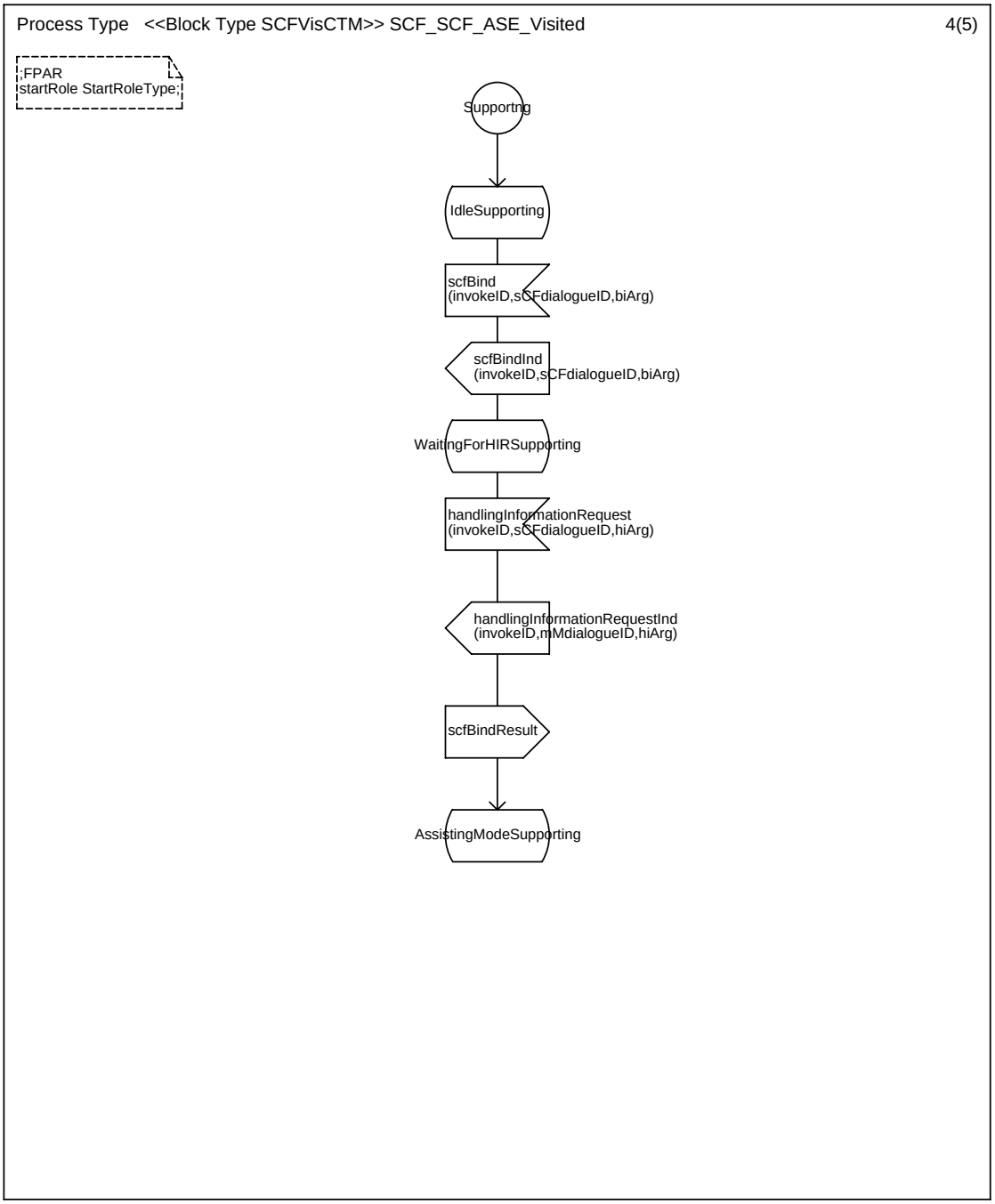
Annex C18 : Process Type SCF_SCF_ASE_Visited

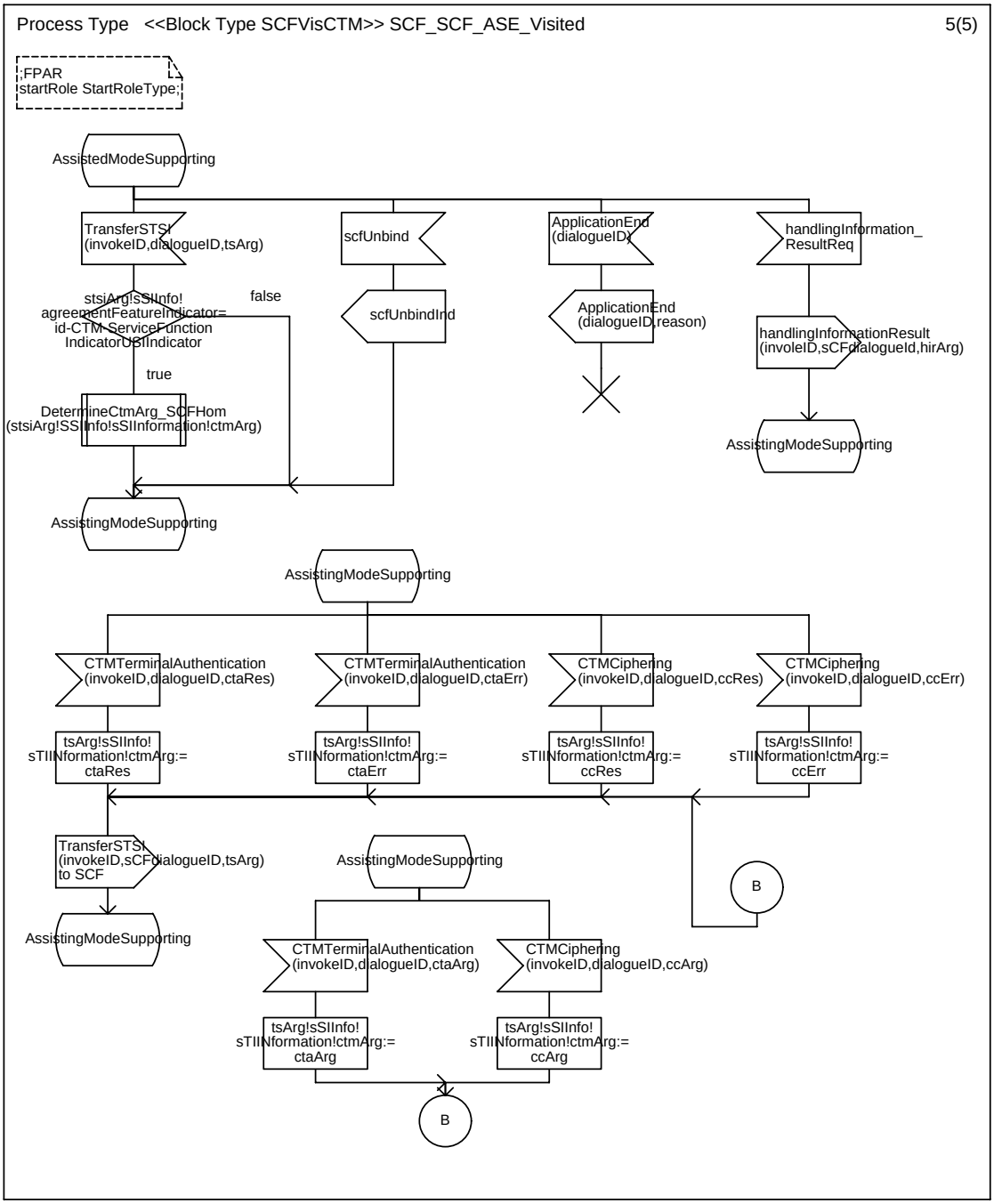


Process Type <<Block Type SCFVisCTM>> SCF_SCF_ASE_Visited 2(5)

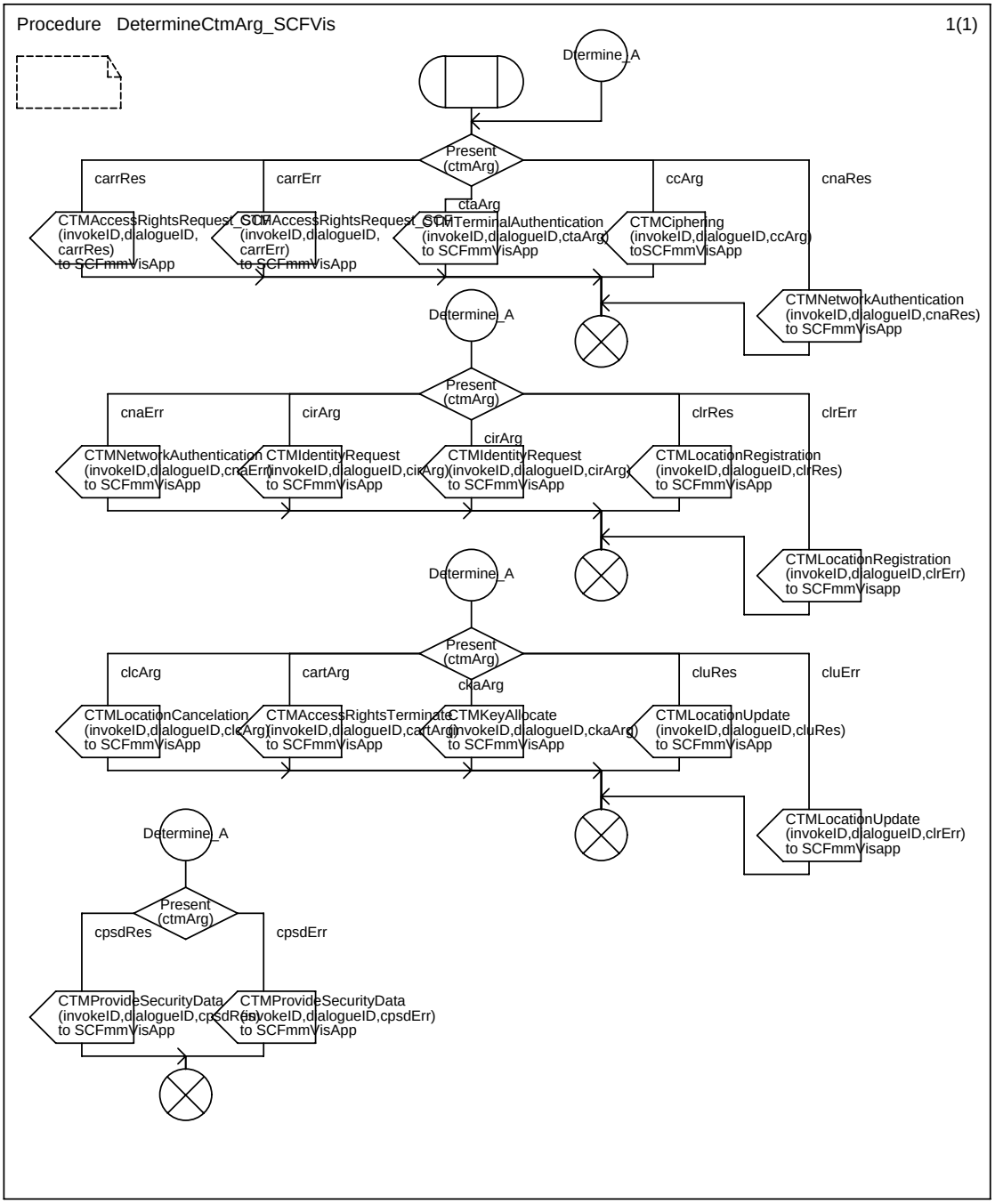



Process Type <<Block Type SCFVisCTM>> SCF_SCF_ASE_Visited 3(5)
;FPAR
startRole StartRoleType;
AssistedModeControlling
CTMTerminalAuthentication (invokeID,sCFdialogueID,ctaRes)
CTMTerminalAuthentication (invokeID,dialogueID,ctaErr)
CTMCiphering (invokeID,dialogueID,ccRes)
CTMCiphering (invokeID,dialogueID,ccErr)
tsArg!sSIInfo! sTIINformation!ctmArg:= ctaRes
tsArg!sSIInfo! sTIINformation!ctmArg:= ctaErr
tsArg!sSIInfo! sTIINformation!ctmArg:= ccRes
tsArg!sSIInfo! sTIINformation!ctmArg:= ccErr
TransferSTSI ((GetInvokeID),sCFdialogueID,tsArg) to SCF
A
AssistedModeControlling
CTMKeyAllocate (invokeID,dialogueID,ckaRes)
CTMKeyAllocate (invokeID,dialogueID,ckaErr)
CTMNetworkAuthentication (invokeID,dialogueID,cnaArg)
CTMAccessRightsRequest_SCF (invokeID,dialogueID, carrArg)
tsArg!sSIInfo! sTIINformation!ctmArg:= ckaRes
tsArg!sSIInfo! sTIINformation!ctmArg:= ckaErr
tsArg!sSIInfo! sTIINformation!ctmArg:= cnaArg
tsArg!sSIInfo! sTIINformation!ctmArg:= carrArg
AssistedModeControlling
A
CTMLocationRegistration (invokeID,dialogueID, clrArg)
CTMLocationUpdate (invokeID,dialogueID, cluArg)
CTMLocationCancelation (invokeID,dialogueID, clcRes)
CTMLocationCancelation (invokeID,dialogueID, clcErr)
tsArg!sSIInfo! sTIINformation!ctmArg:= clrArg
tsArg!sSIInfo! sTIINformation!ctmArg:= cluArg
tsArg!sSIInfo! sTIINformation!ctmArg:= clcRes
tsArg!sSIInfo! sTIINformation!ctmArg:= clcErr
AssistedModeControlling
A
CTMProvideSecurityData (invokeID,dialogueID, cpsdArg)
CTMTerminalAuthentication (invokeID,dialogueID,ctaArg)
CTMCiphering (invokeID,dialogueID,ccArg)
tsArg!sSIInfo! sTIINformation!ctmArg:= cpsdArg
tsArg!sSIInfo! sTIINformation!ctmArg:= ctaArg
tsArg!sSIInfo! sTIINformation!ctmArg:= ccArg
A

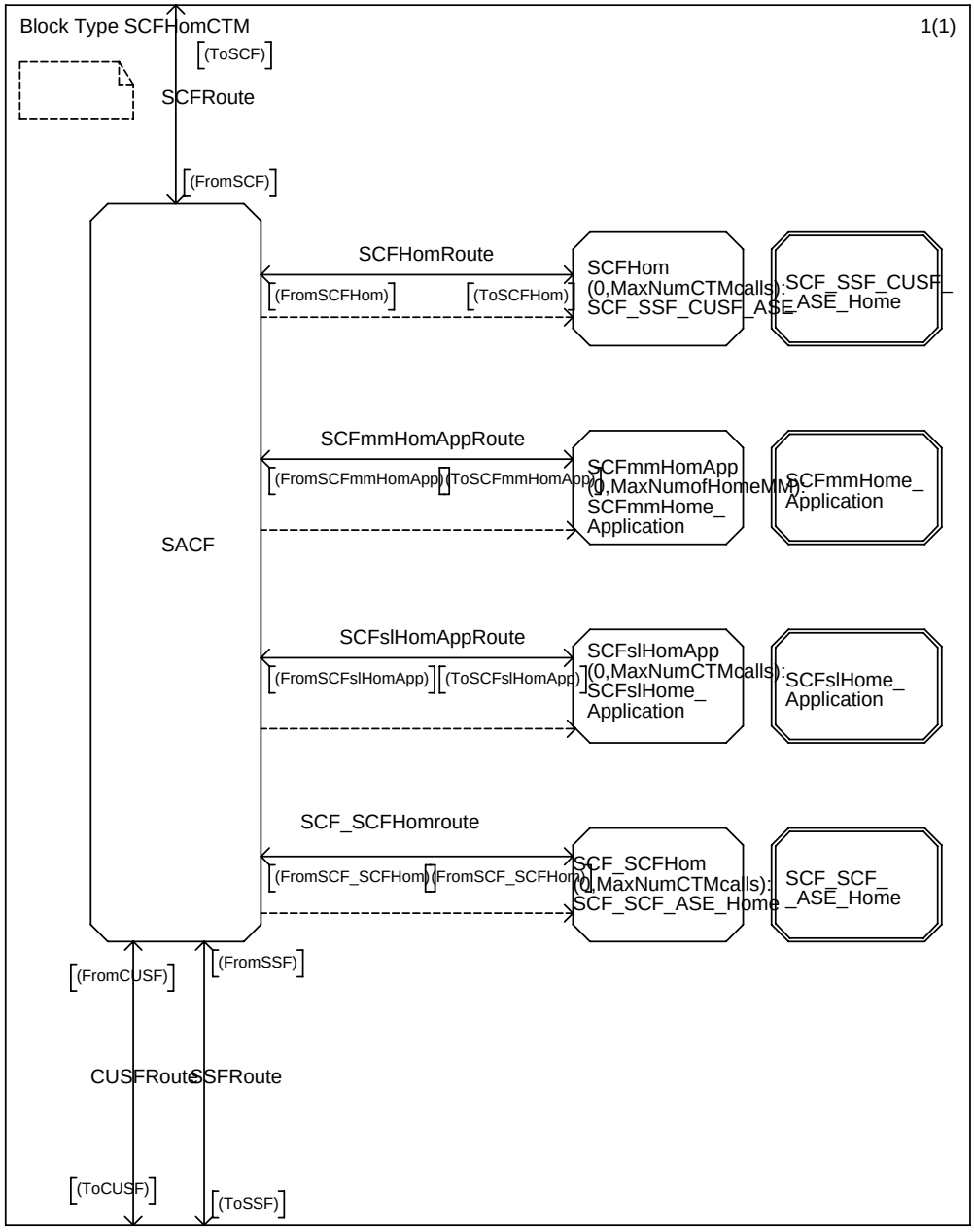

Process Type <<Block Type SCFVisCTM>> SCF_SCF_ASE_Visited
4(5)
;FPAR
startRole StartRoleType;
Supportng
IdleSupporting
scfBind
(invokeID,sCFdialogueID,biArg)
scfBindInd
(invokeID,sCFdialogueID,biArg)
WaitingForHIRSupporting
handlingInformationRequest
(invokeID,sCFdialogueID,hiArg)
handlingInformationRequestInd
(invokeID,mMdialogueID,hiArg)
scfBindResult
AssistingModeSupporting


Process Type <<Block Type SCFVisCTM>> SCF_SCF_ASE_Visited 5(5)
;FPAR
startRole StartRoleType;
AssistedModeSupporting
TransferSTSI
(invokeID,dialogueID,tsArg)
scfUnbind
ApplicationEnd
(dialogueID)
handlingInformation_
ResultReq
stsiArg!sSIInfo!
agreementFeatureIndicator=
id-CTM-ServiceFunction
IndicatorOSIIndicator
false
true
scfUnbindInd
ApplicationEnd
(dialogueID,reason)
handlingInformationResult
(involeID,sCFdialogueId,hirArg)
DetermineCtmArg_SCFHom
(stsiArg!SSIInfo!sSIInformation!ctmArg)
AssistingModeSupporting
AssistingModeSupporting
AssistingModeSupporting
CTMTerminalAuthentication
(invokeID,dialogueID,ctaRes)
CTMTerminalAuthentication
(invokeID,dialogueID,ctaErr)
CTMCiphering
(invokeID,dialogueID,ccRes)
CTMCiphering
(invokeID,dialogueID,ccErr)
tsArg!sSIInfo!
sTIINformation!ctmArg:=
ctaRes
tsArg!sSIInfo!
sTIINformation!ctmArg:=
ctaErr
tsArg!sSIInfo!
sTIINformation!ctmArg:=
ccRes
tsArg!sSIInfo!
sTIINformation!ctmArg:=
ccErr
TransferSTSI
(invokeID,sCFdialogueID,tsArg)
to SCF
AssistingModeSupporting
B
AssistingModeSupporting
CTMTerminalAuthentication
(invokeID,dialogueID,ctaArg)
CTMCiphering
(invokeID,dialogueID,ccArg)
tsArg!sSIInfo!
sTIINformation!ctmArg:=
ctaArg
tsArg!sSIInfo!
sTIINformation!ctmArg:=
ccArg
B

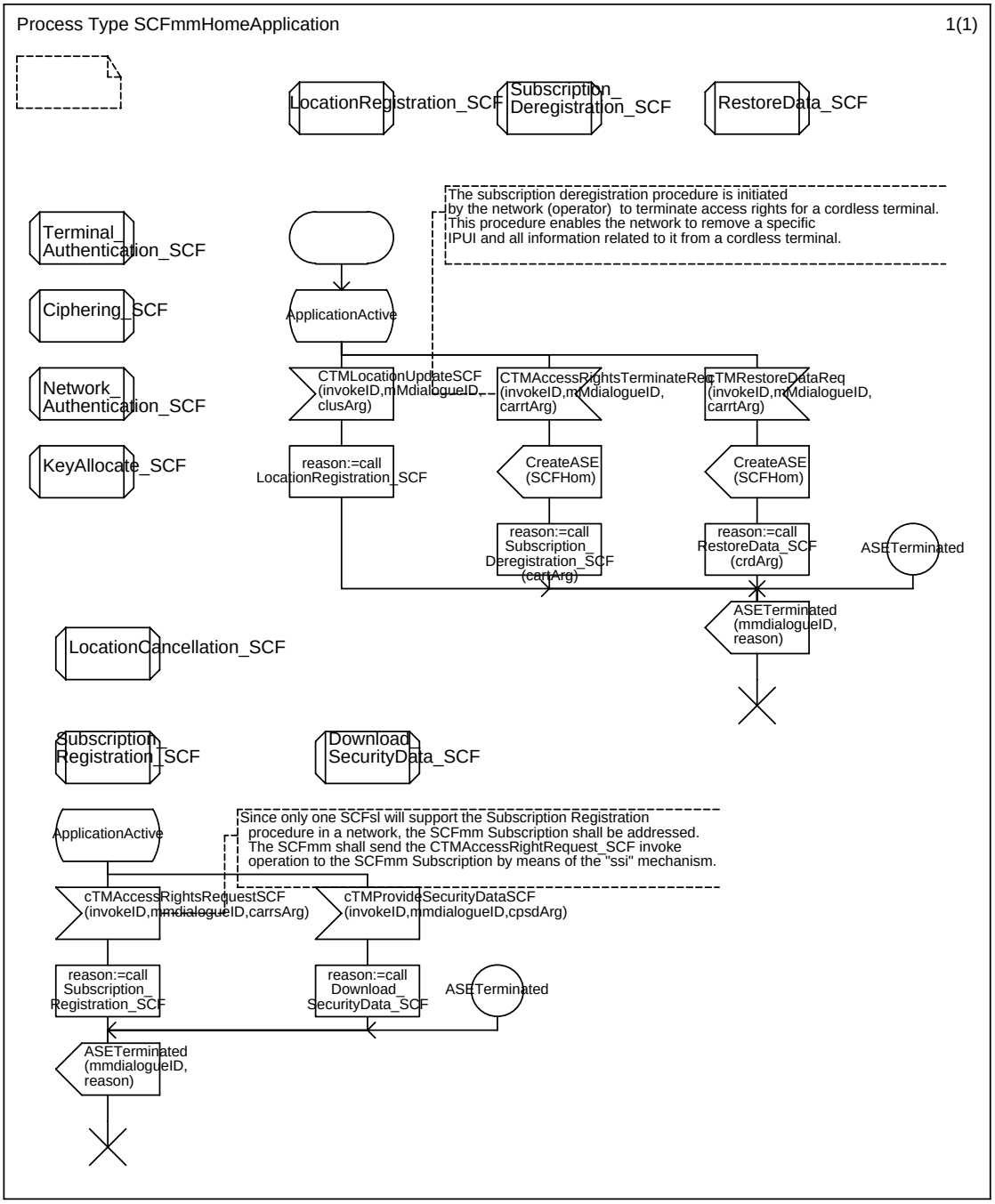
Annex C19 : Procedure DetermineCtmArg_SCFVis



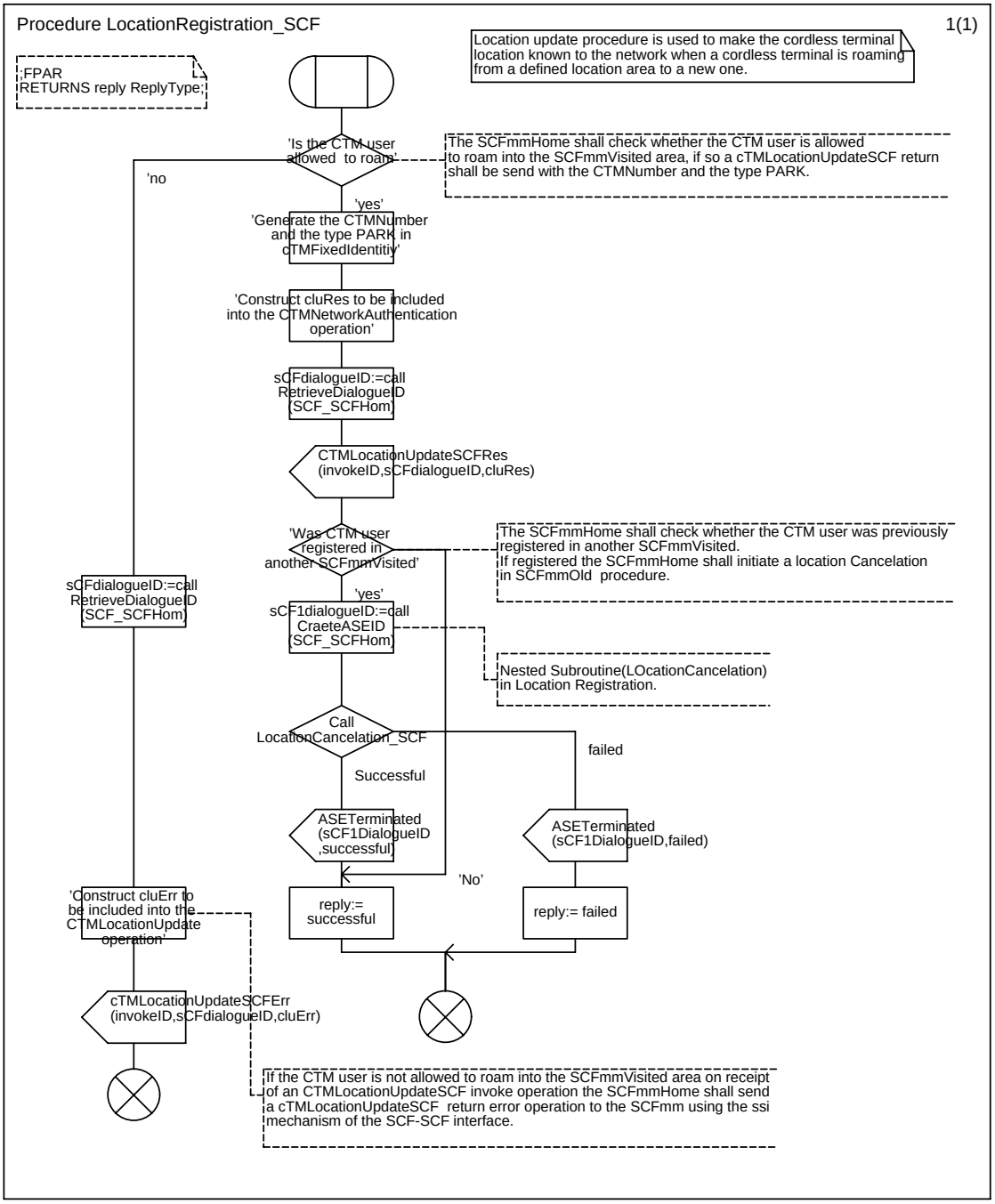
Annex C20 : Block Type SCFHomCTM



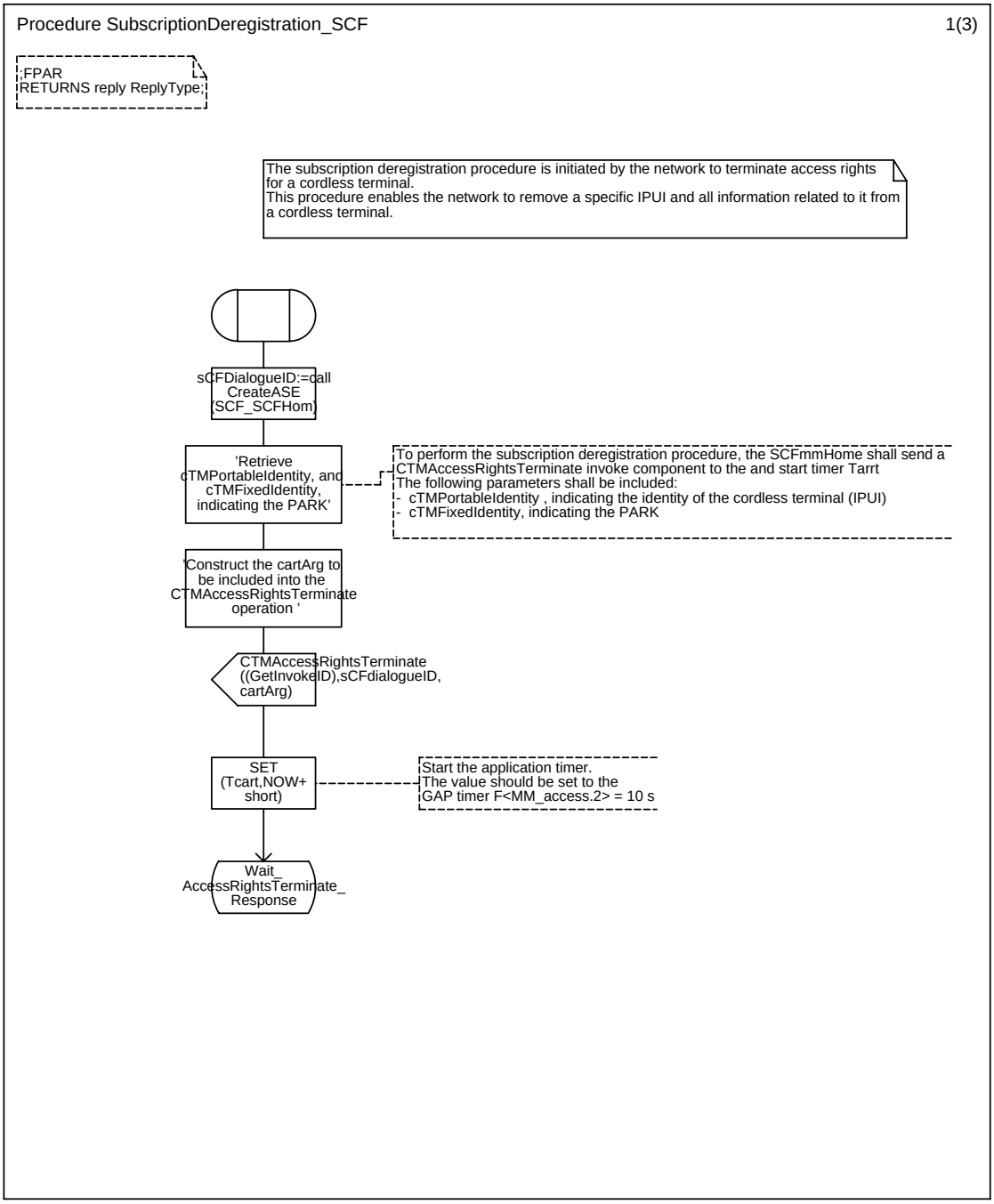
Annex C21 : Process Type SCFmmHomeApplication

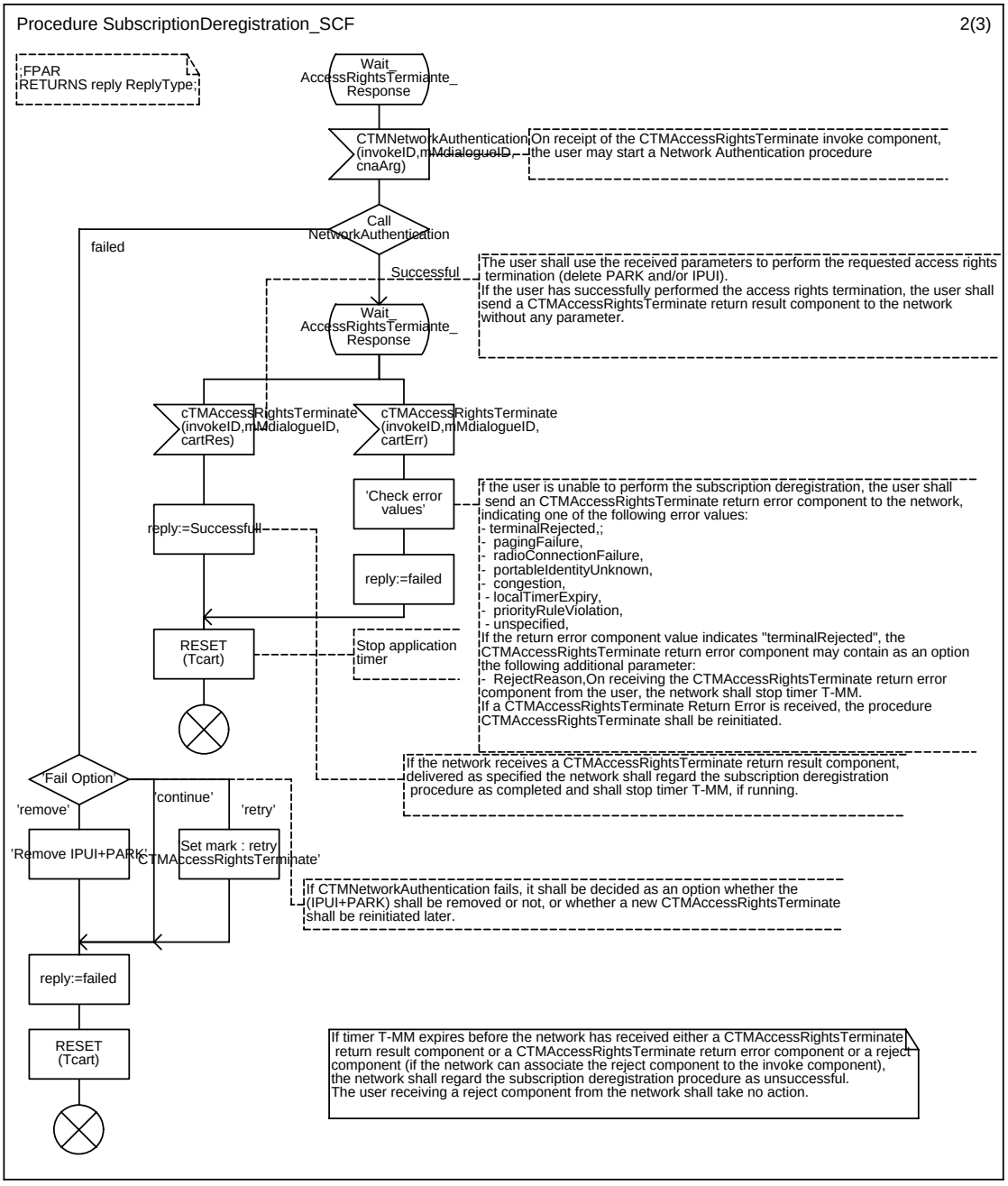


Annex C22 : Procedure LocationRegistration_SCF

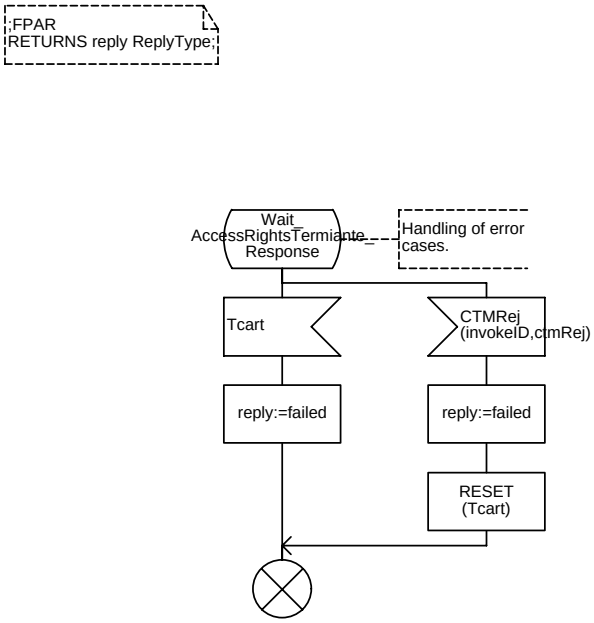


Annex C23 : Procedure SubscriptionDeregistration_SCF

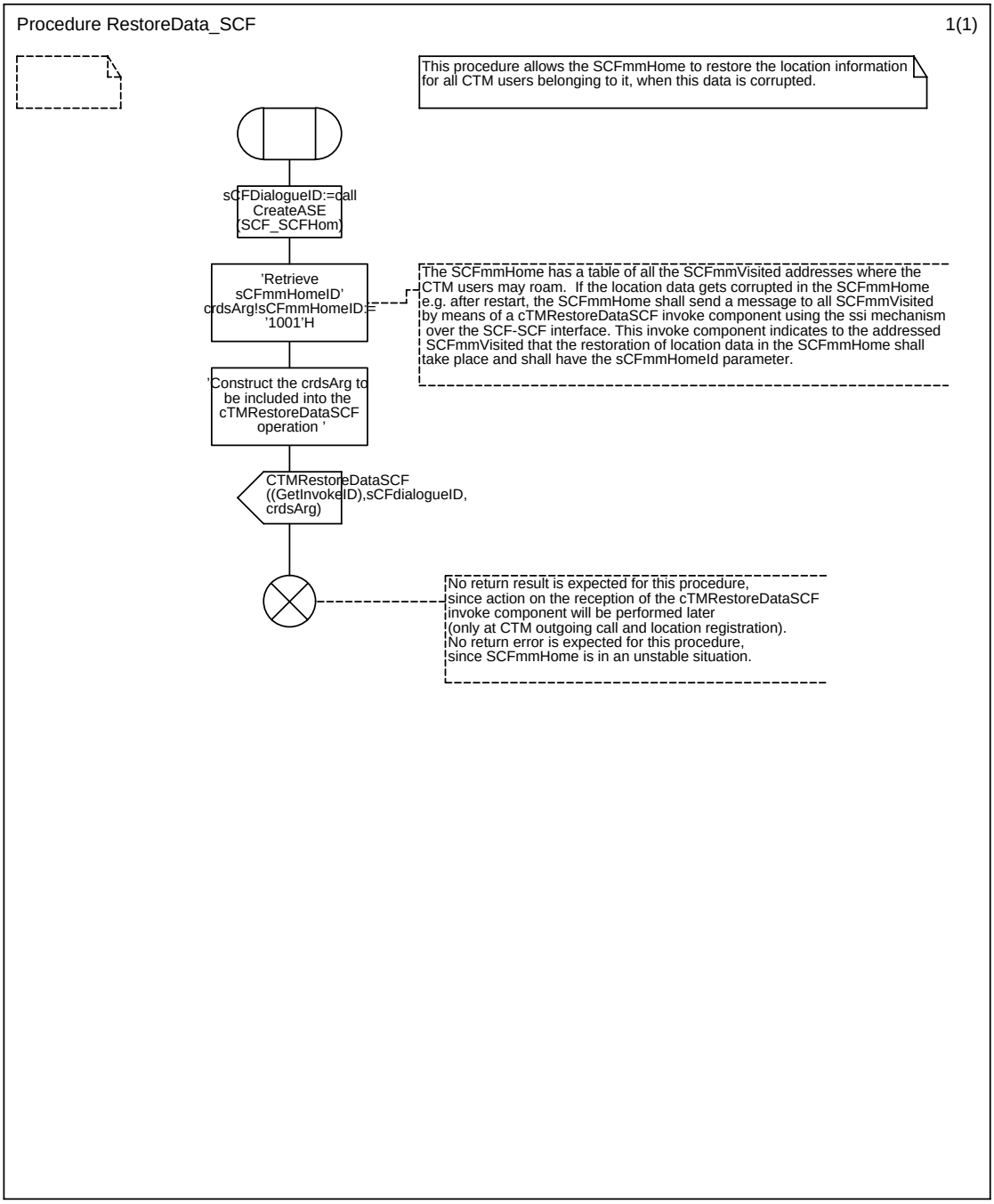



Procedure SubscriptionDeregistration_SCF
2(3)
;FPAR
RETURNS reply ReplyType;
Wait_
AccessRightsTermiante_
Response
CTMNetworkAuthentication
(invokeID,mMdialogueID,
cnaArg)
On receipt of the CTMAccessRightsTerminate invoke component, the user may start a Network Authentication procedure
Call
NetworkAuthentication
failed
Successful
The user shall use the received parameters to perform the requested access rights termination (delete PARK and/or IPUI).
If the user has successfully performed the access rights termination, the user shall send a CTMAccessRightsTerminate return result component to the network without any parameter.
Wait_
AccessRightsTermiante_
Response
cTMAccessRightsTerminate
(invokeID,mMdialogueID,
cartRes)
cTMAccessRightsTerminate
(invokeID,mMdialogueID,
cartErr)
'Check error values'
If the user is unable to perform the subscription deregistration, the user shall send an CTMAccessRightsTerminate return error component to the network, indicating one of the following error values:
- terminalRejected,;
- pagingFailure,
- radioConnectionFailure,
- portableIdentityUnknown,
- congestion,
- localTimerExpiry,
- priorityRuleViolation,
- unspecified,
If the return error component value indicates "terminalRejected", the CTMAccessRightsTerminate return error component may contain as an option the following additional parameter:
- RejectReason,On receiving the CTMAccessRightsTerminate return error component from the user, the network shall stop timer T-MM.
If a CTMAccessRightsTerminate Return Error is received, the procedure CTMAccessRightsTerminate shall be reinitiated.
reply:=Successful
reply:=failed
RESET
(Tcart)
Stop application timer
If the network receives a CTMAccessRightsTerminate return result component, delivered as specified the network shall regard the subscription deregistration procedure as completed and shall stop timer T-MM, if running.
'Fail Option'
'remove'
'continue'
'retry'
'Remove IPUI+PARK'
'Set mark : retry CTMAccessRightsTerminate'
If CTMNetworkAuthentication fails, it shall be decided as an option whether the (IPUI+PARK) shall be removed or not, or whether a new CTMAccessRightsTerminate shall be reinitiated later.
reply:=failed
RESET
(Tcart)
If timer T-MM expires before the network has received either a CTMAccessRightsTerminate return result component or a CTMAccessRightsTerminate return error component or a reject component (if the network can associate the reject component to the invoke component), the network shall regard the subscription deregistration procedure as unsuccessful.
The user receiving a reject component from the network shall take no action.

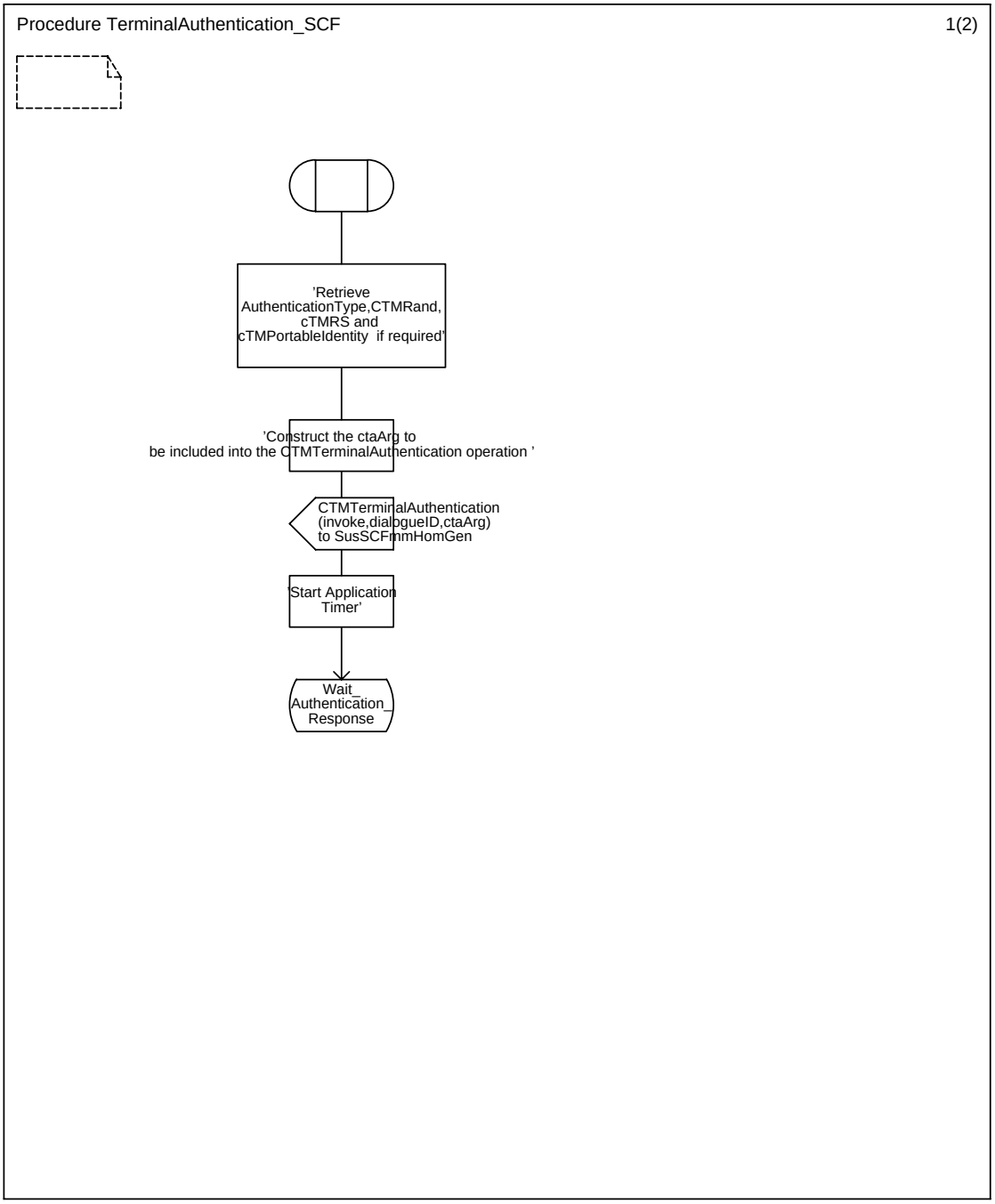
Procedure SubscriptionDeregistration_SCF 3(3)



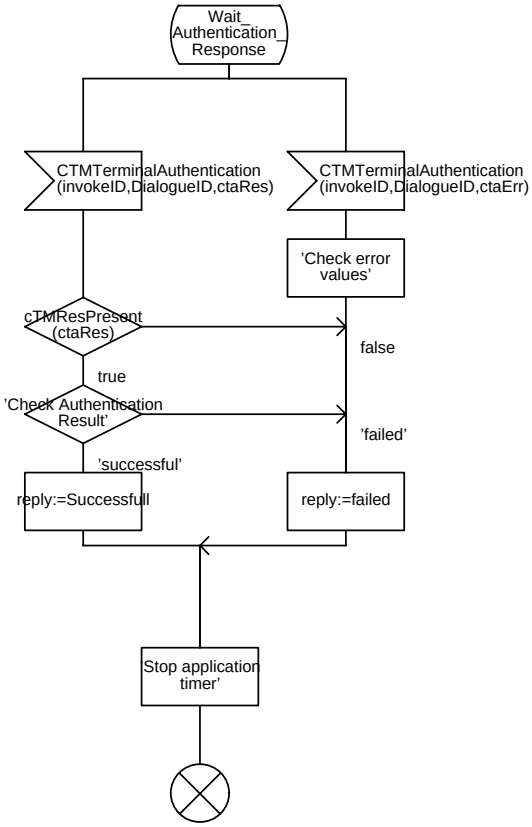
Annex C24 : Procedure RestoreData_SCF



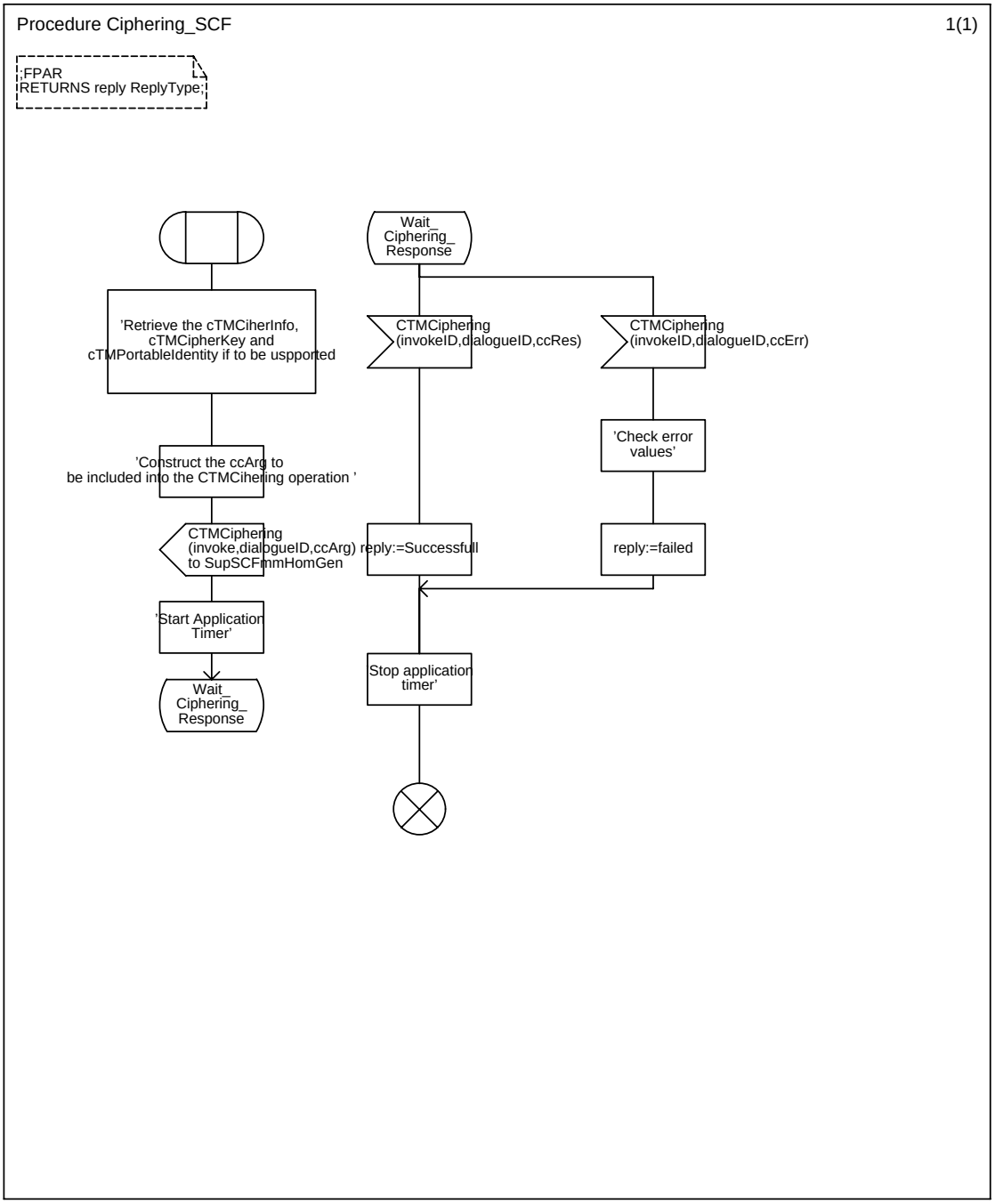
Annex C25 : Procedure TerminalAuthentication_SCF



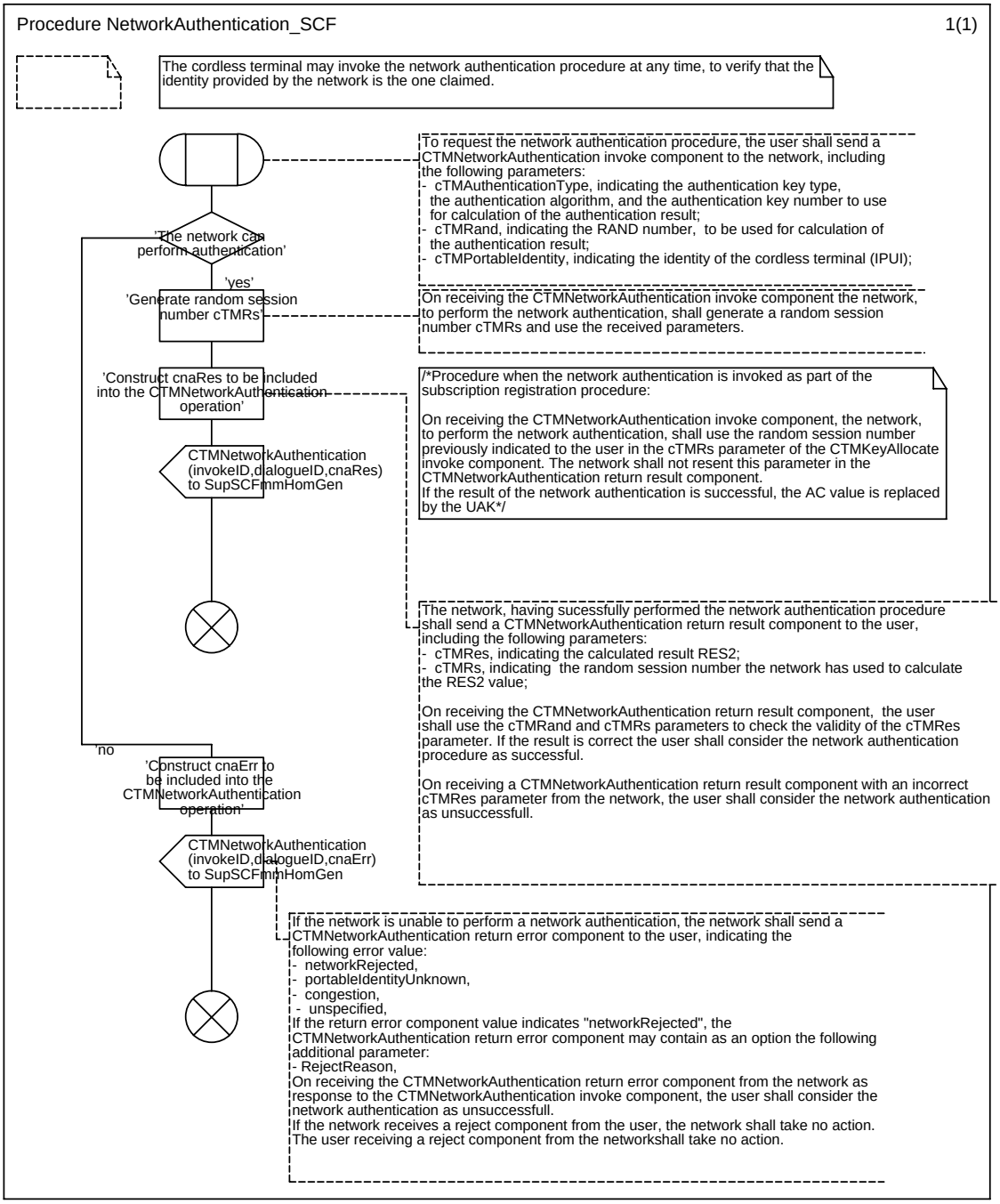
Procedure TerminalAuthentication_SCF 2(2)



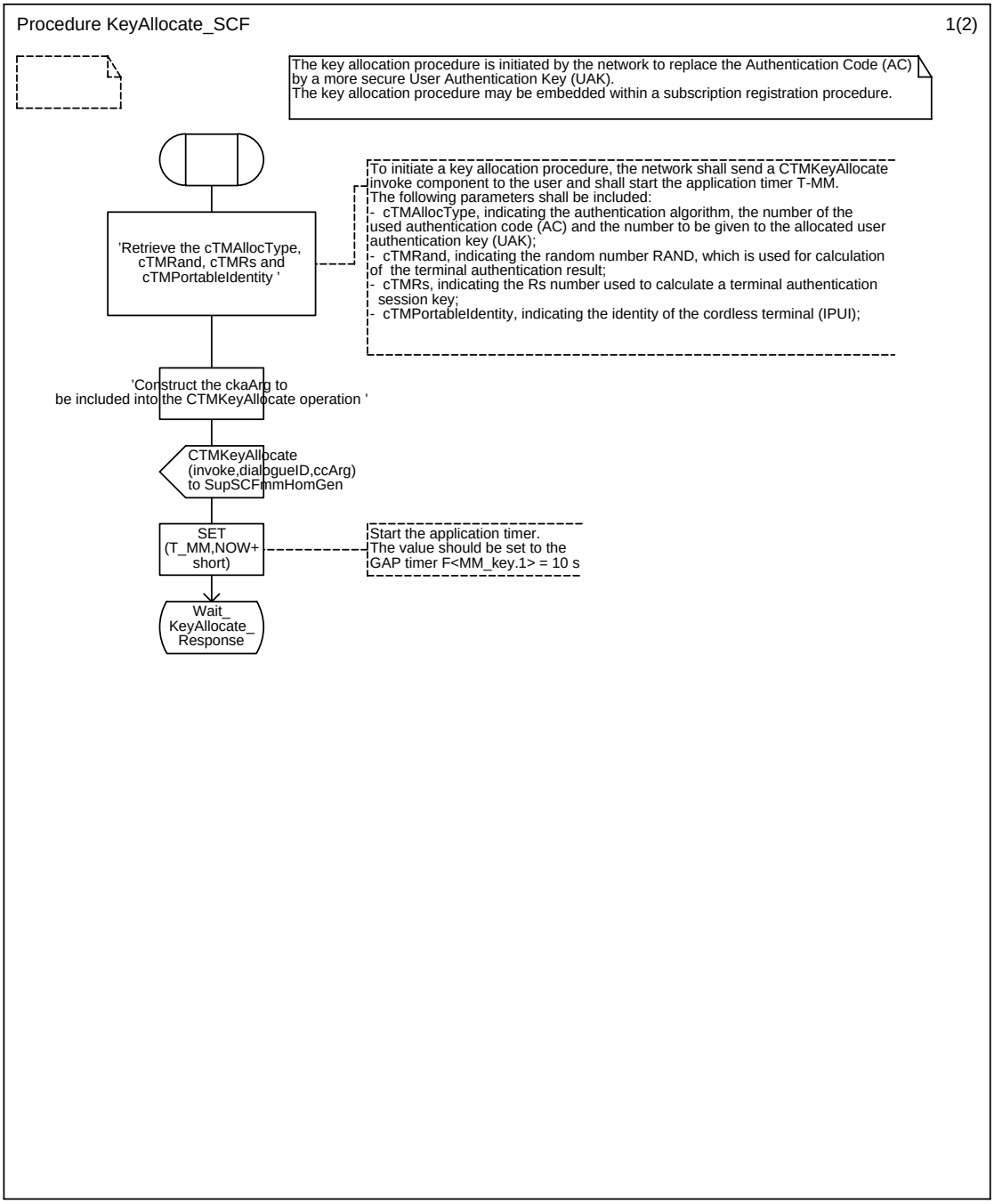
Annex C26 : Procedure Ciphering_SCF



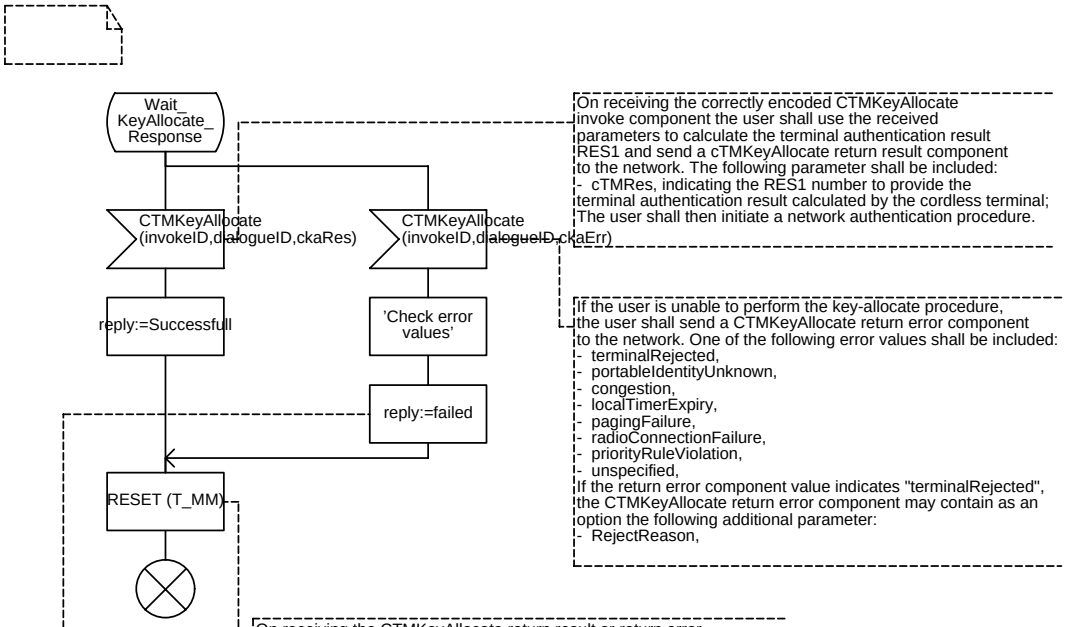
Annex C27 : Procedure NetworkAuthentication_SCF



Annex C28 : Procedure KeyAllocate_SCF



Procedure KeyAllocate_SCF 2(2)



On receiving the correctly encoded CTMKeyAllocate invoke component the user shall use the received parameters to calculate the terminal authentication result RES1 and send a cTMKeyAllocate return result component to the network. The following parameter shall be included:
- cTMRes, indicating the RES1 number to provide the terminal authentication result calculated by the cordless terminal;

The user shall then initiate a network authentication procedure.

If the user is unable to perform the key-allocate procedure, the user shall send a CTMKeyAllocate return error component to the network. One of the following error values shall be included:
- terminalRejected,
- portableIdentityUnknown,
- congestion,
- localTimerExpiry,
- pagingFailure,
- radioConnectionFailure,
- priorityRuleViolation,
- unspecified,

If the return error component value indicates "terminalRejected", the CTMKeyAllocate return error component may contain as an option the following additional parameter:
- RejectReason,

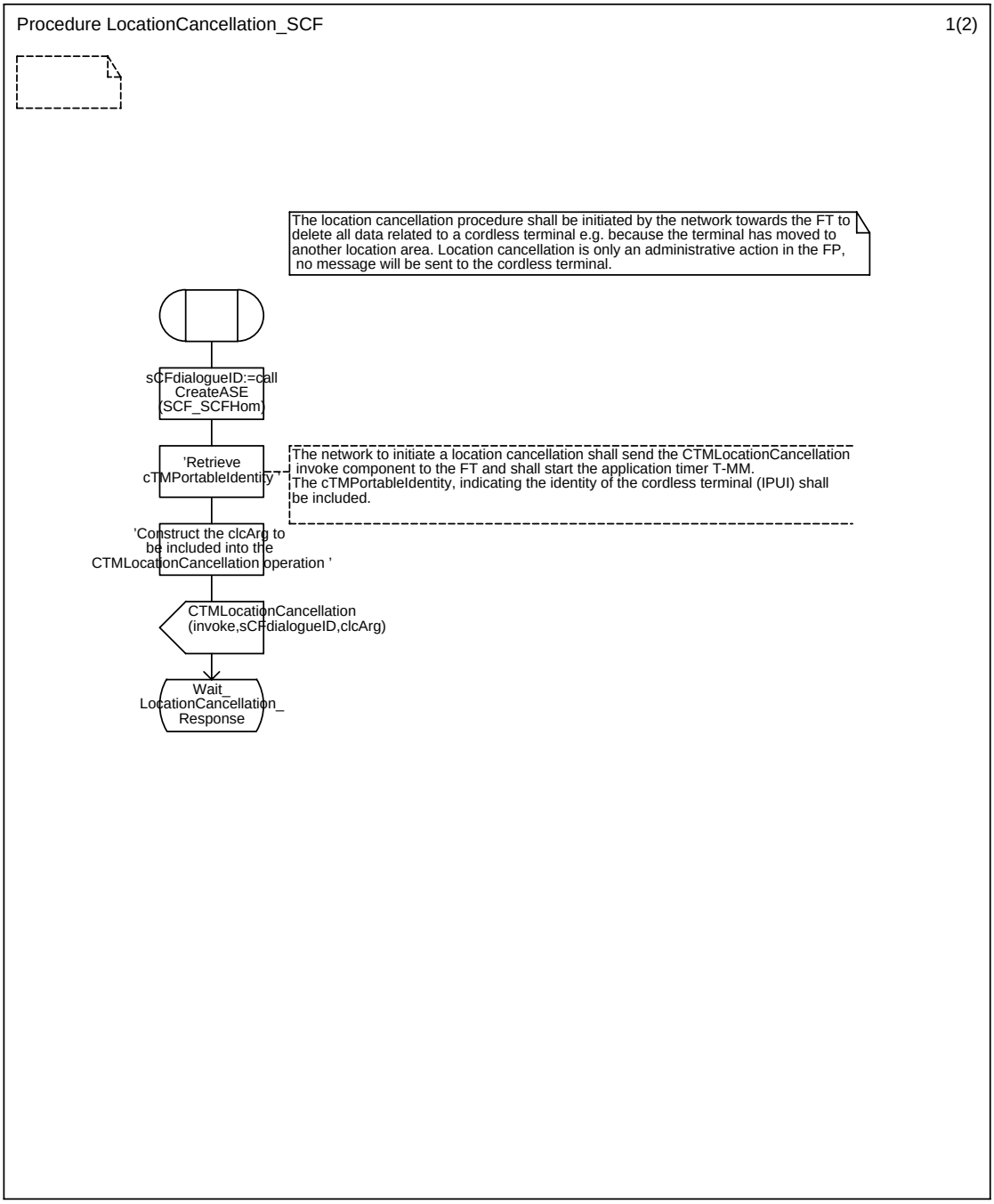
On receiving the CTMKeyAllocate return result or return error component from the user, the network shall stop timer T-MM, if running.

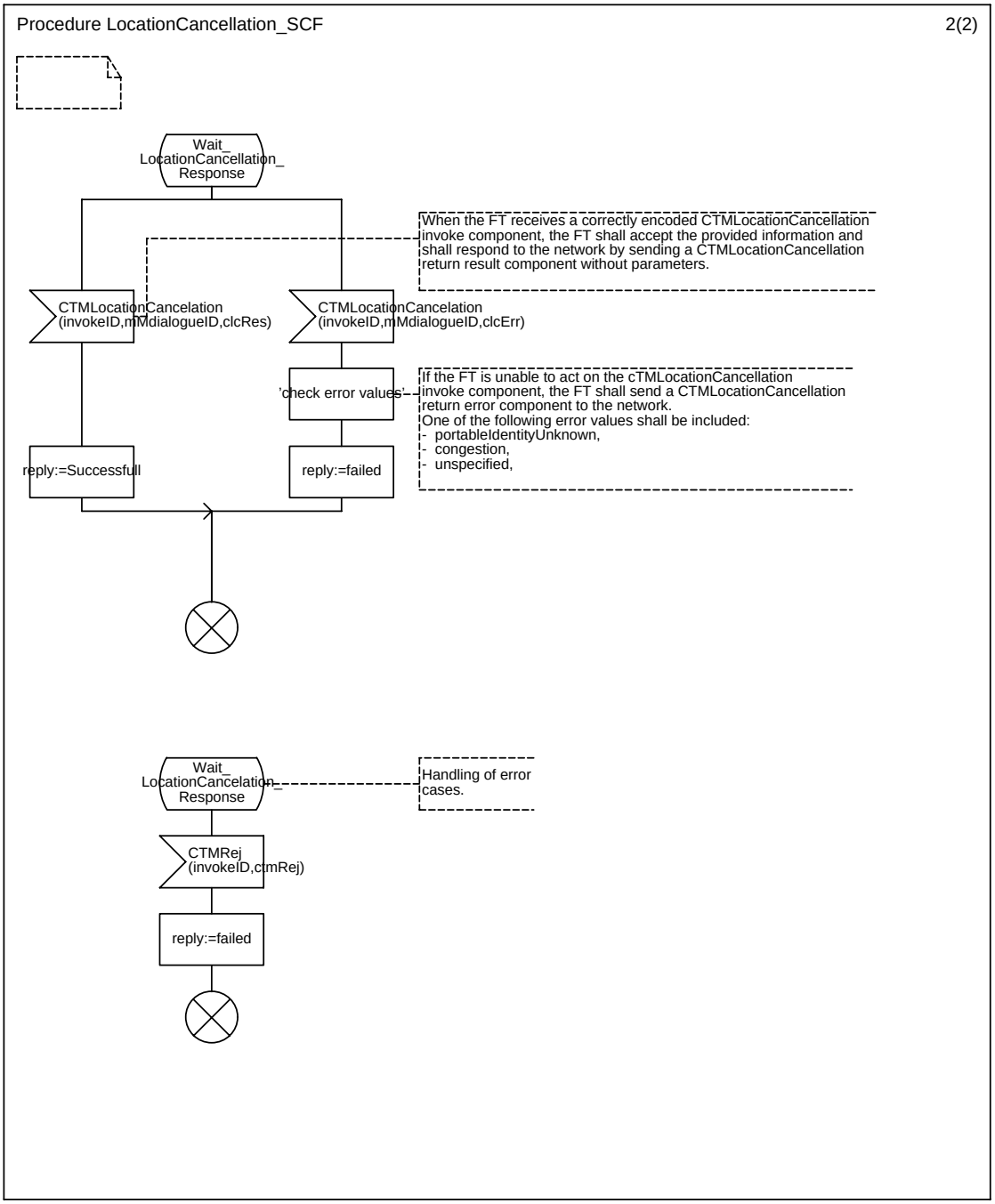
On receiving the CTMKeyAllocate return error component from the user, the network shall stop the application timer T-MM.
The network shall consider the key allocation procedure as unsuccessful, and may take appropriate actions, if either:
- the network receives a CTMKeyAllocate return error component, or
- the network receives a CTMKeyAllocate return result component (with an incorrect parameter) and the authentication of the user fails, or
- the network receives a reject component and the network can correlated it to the CTMKeyAllocate invoke component, or

- the timer T-MM expires before the network has either received a CTMKeyAllocate return result component or a CTMKeyAllocate return error or a reject component (if the network can associate the reject component with the invoke component);

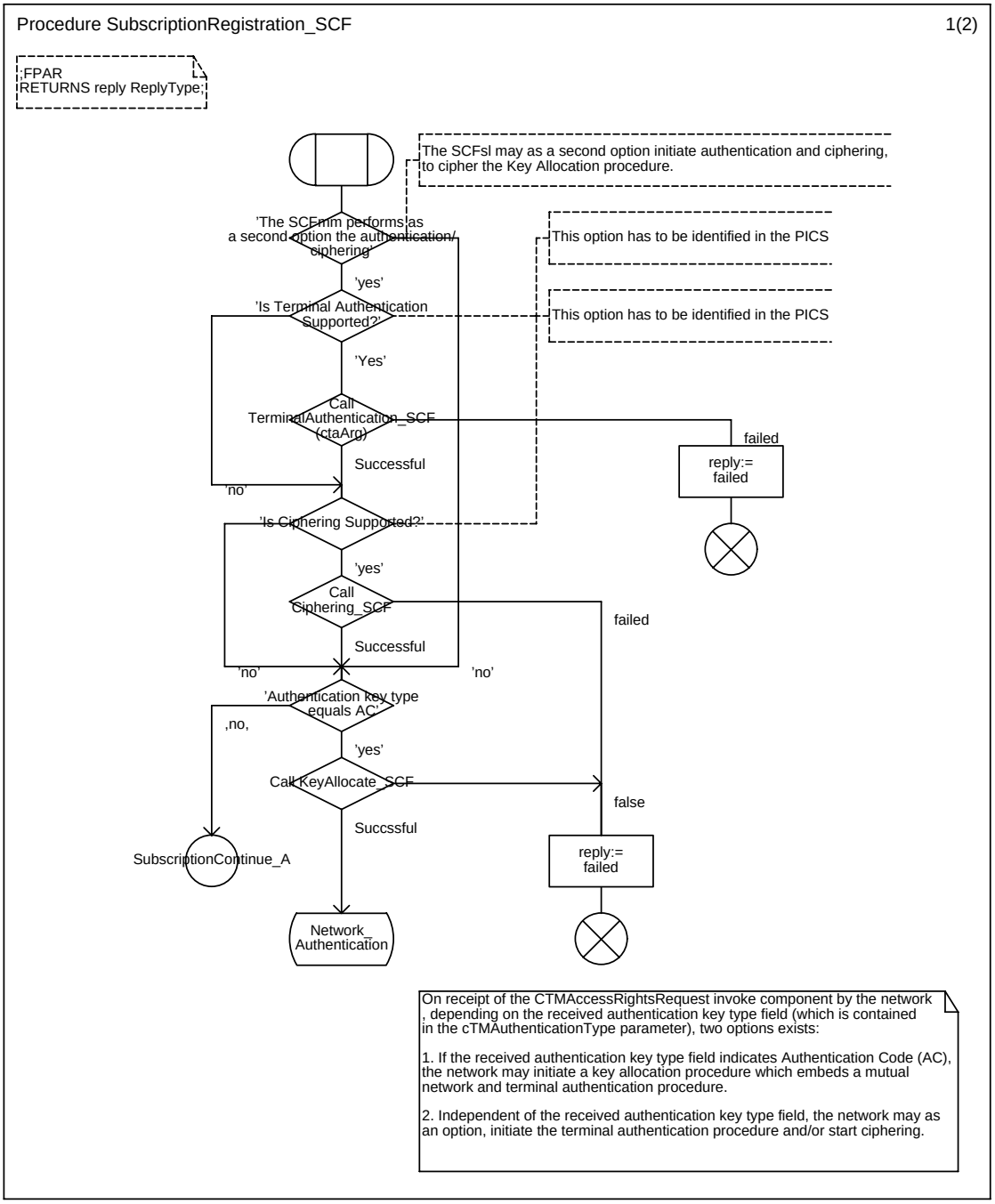
If the user returns a CTMKeyAllocate return error component or a reject component in response to the CTMKeyAllocate invoke component, no network authentication procedure shall be initiated.
In the second case, if the authentication of the user fails, the network shall respond with a CTMNetworkAuthentication return error component to the first (subsequent) following CTMNetworkAuthentication invoke component and include the error value "networkRejected".
If either the key allocation procedure or the network authentication procedure are not successful, the AC value shall not be replaced by the UAK.
If the network receives a reject component from the user, and the network can correlate it to the CTMNetworkAuthentication return result component, the network may take appropriate actions.
The user receiving a reject component from the network shall take no action.

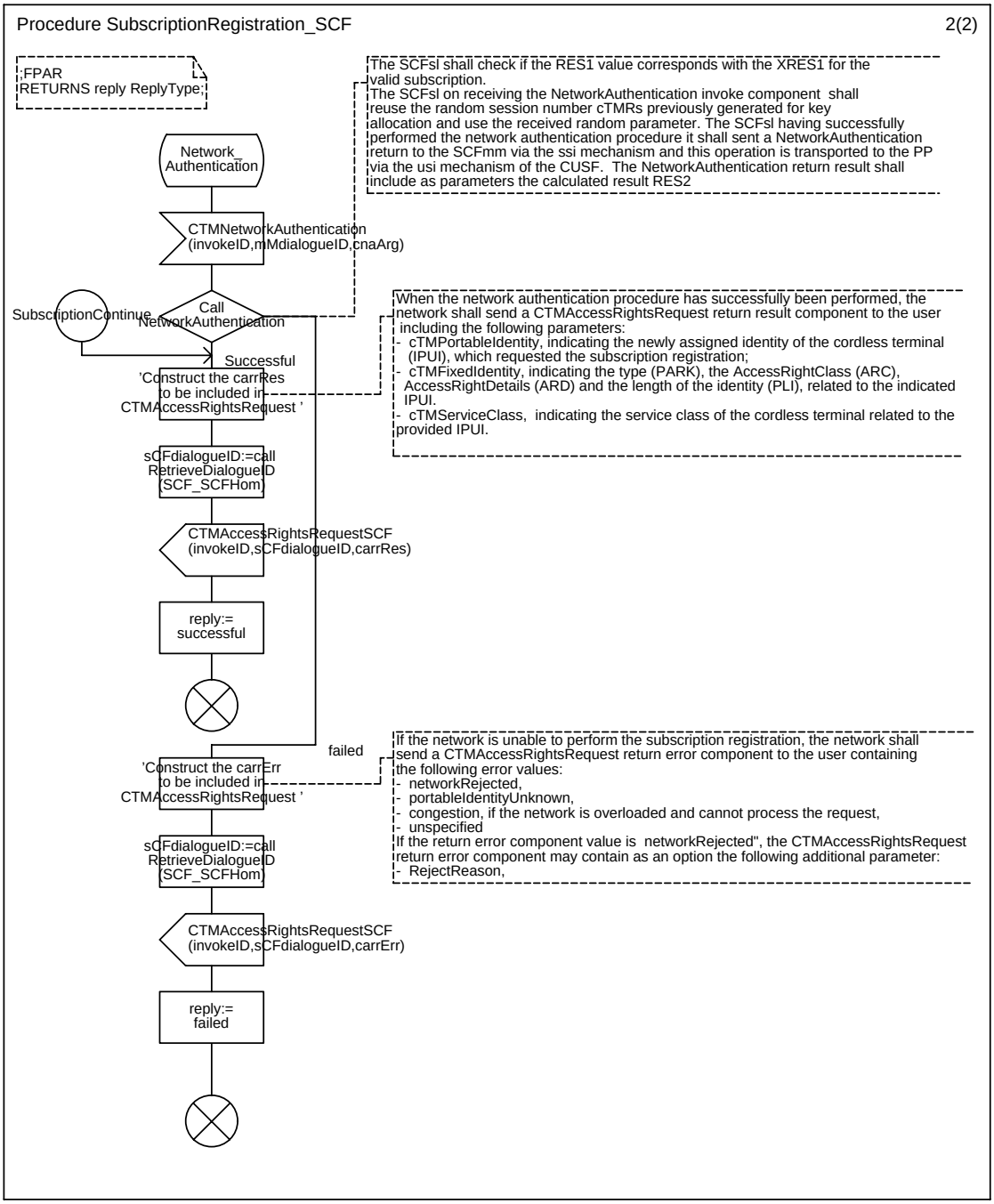
Annex C29 : Procedure LocationCancellation_SCF



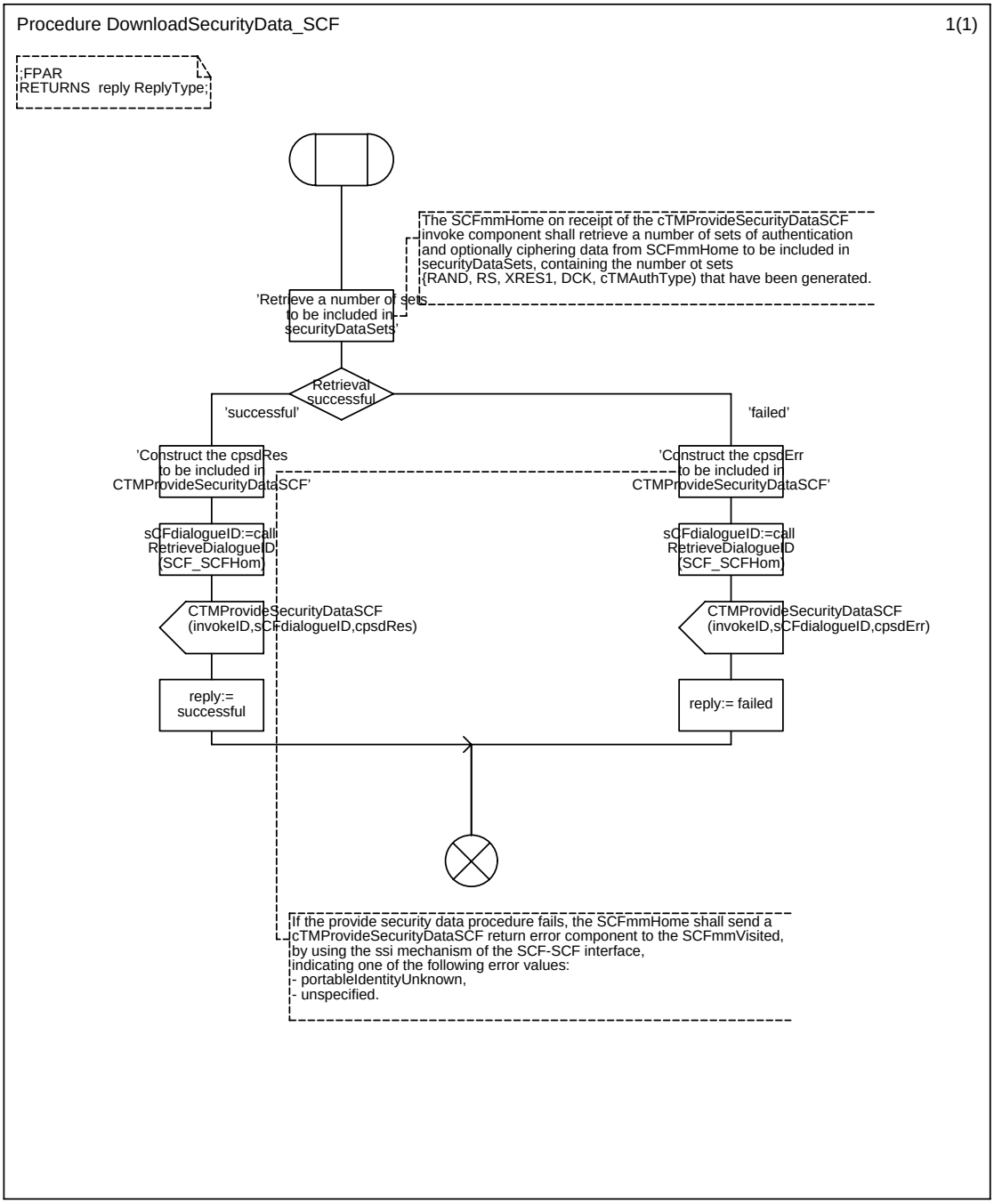

Procedure LocationCancellation_SCF
2(2)
Wait_ LocationCancellation_ Response
When the FT receives a correctly encoded CTMLocationCancellation invoke component, the FT shall accept the provided information and shall respond to the network by sending a CTMLocationCancellation return result component without parameters.
CTMLocationCancelation (invokeID,mMdialogueID,clcRes)
CTMLocationCancelation (invokeID,mMdialogueID,clcErr)
'check error values'
If the FT is unable to act on the cTMLocationCancellation invoke component, the FT shall send a CTMLocationCancellation return error component to the network.
One of the following error values shall be included:
- portableIdentityUnknown,
- congestion,
- unspecified,
reply:=Successfull
reply:=failed
Wait_ LocationCancelation_ Response
Handling of error cases.
CTMRej (invokeID,ctmRej)
reply:=failed

Annex C30 : Procedure SubscriptionRegistration_SCF

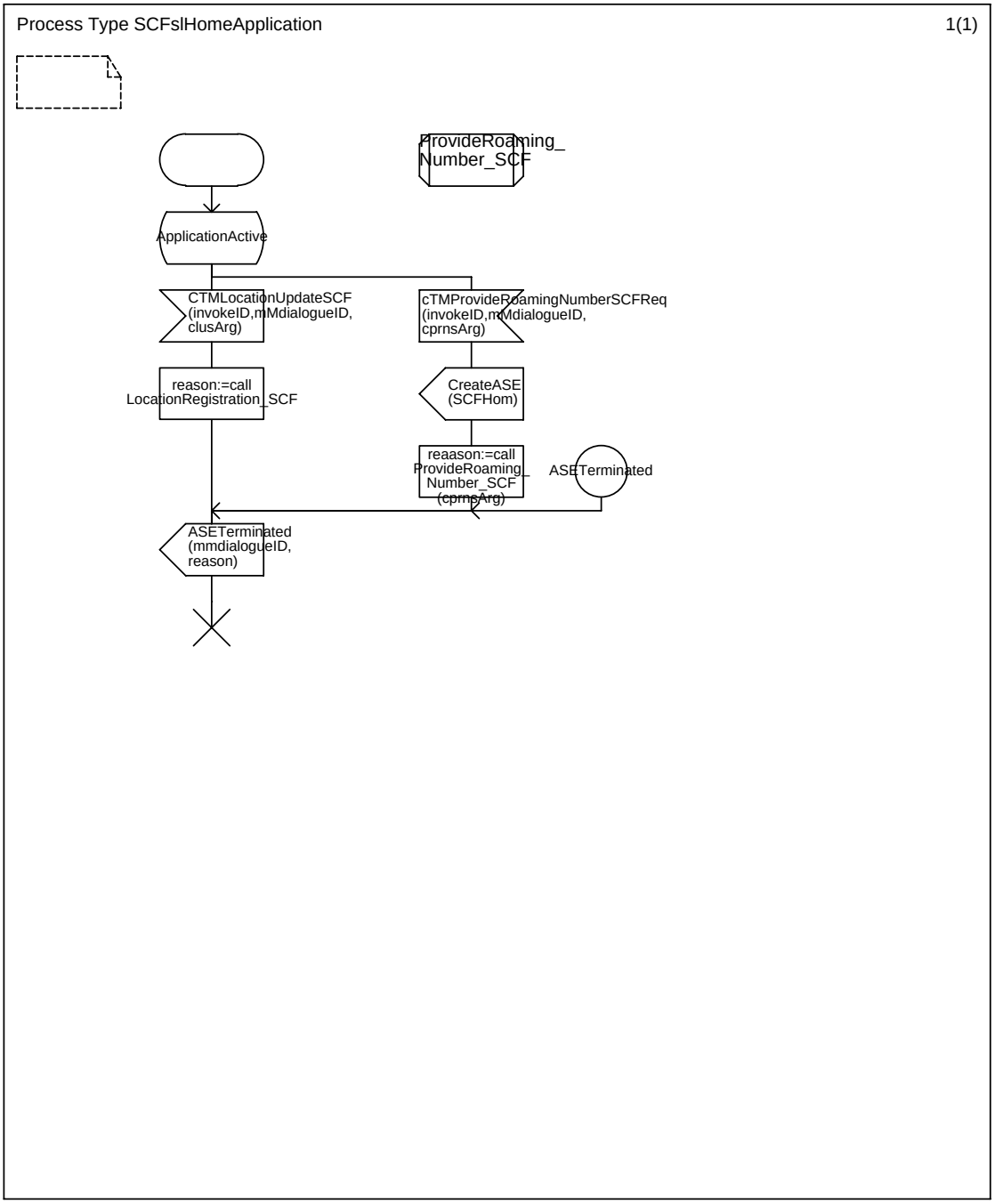



Procedure SubscriptionRegistration_SCF
2(2)
;FPAR
RETURNS reply ReplyType;
Network_
Authentication
CTMNetworkAuthentication
(invokeID,mMdialogueID,cnaArg)
SubscriptionContinue
Call
NetworkAuthentication
Successful
'Construct the carrRes
to be included in
CTMAccessRightsRequest '
sCFdialogueID:=call
RetrieveDialogueID
(SCF_SCFHom)
CTMAccessRightsRequestSCF
(invokeID,sCFdialogueID,carrRes)
reply:=
successful
failed
'Construct the carrErr
to be included in
CTMAccessRightsRequest '
sCFdialogueID:=call
RetrieveDialogueID
(SCF_SCFHom)
CTMAccessRightsRequestSCF
(invokeID,sCFdialogueID,carrErr)
reply:=
failed
The SCFsl shall check if the RES1 value corresponds with the XRES1 for the valid subscription.
The SCFsl on receiving the NetworkAuthentication invoke component shall reuse the random session number cTMRs previously generated for key allocation and use the received random parameter. The SCFsl having successfully performed the network authentication procedure it shall sent a NetworkAuthentication return to the SCFmm via the ssi mechanism and this operation is transported to the PP via the usi mechanism of the CUSF. The NetworkAuthentication return result shall include as parameters the calculated result RES2
When the network authentication procedure has successfully been performed, the network shall send a CTMAccessRightsRequest return result component to the user including the following parameters:
- cTMPortableIdentity, indicating the newly assigned identity of the cordless terminal (IPUI), which requested the subscription registration;
- cTMFixedIdentity, indicating the type (PARK), the AccessRightClass (ARC), AccessRightDetails (ARD) and the length of the identity (PLI), related to the indicated IPUI.
- cTMServiceClass, indicating the service class of the cordless terminal related to the provided IPUI.
If the network is unable to perform the subscription registration, the network shall send a CTMAccessRightsRequest return error component to the user containing the following error values:
- networkRejected,
- portableIdentityUnknown,
- congestion, if the network is overloaded and cannot process the request,
- unspecified
If the return error component value is networkRejected", the CTMAccessRightsRequest return error component may contain as an option the following additional parameter:
- RejectReason,

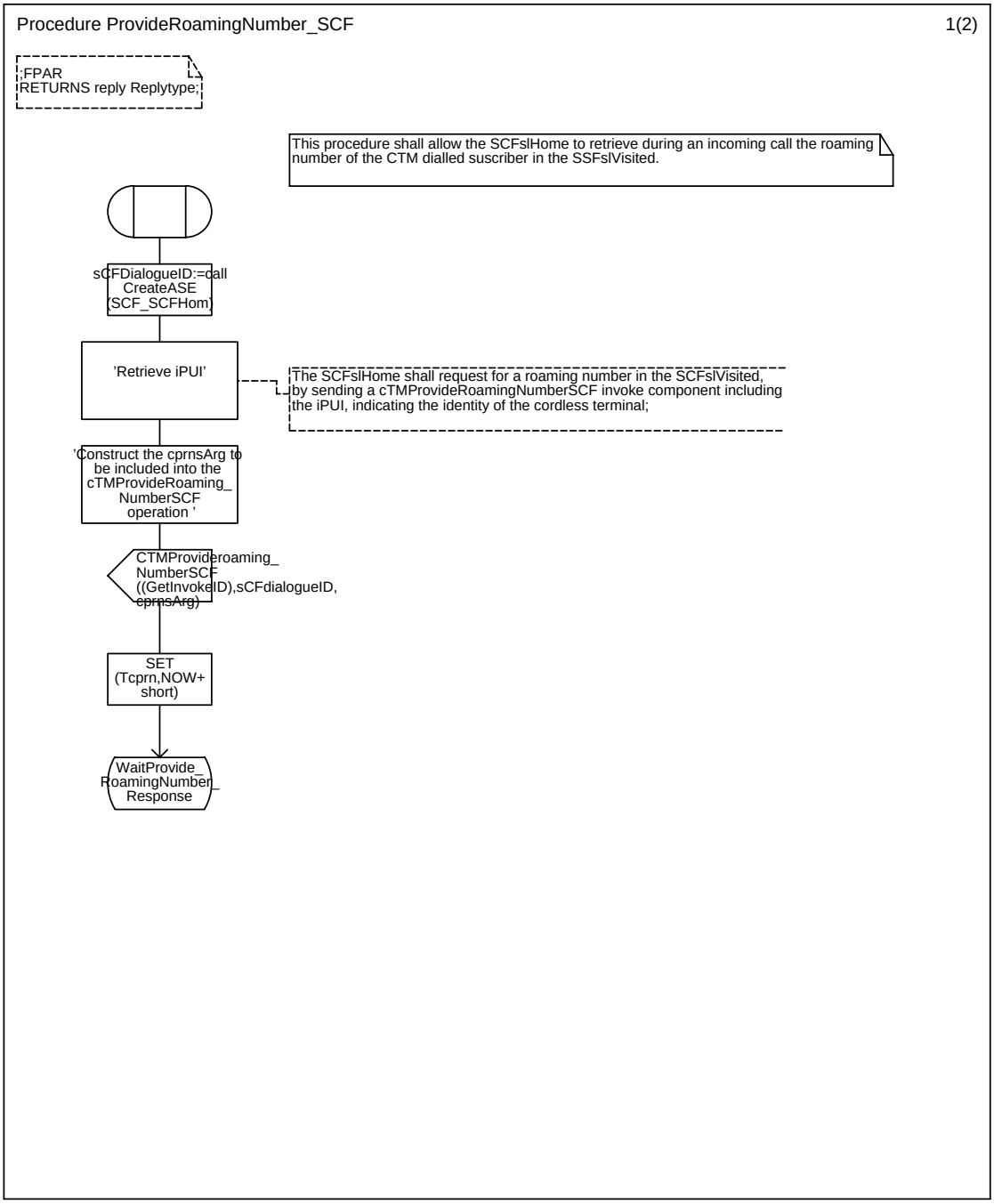
Annex C31 : Procedure DownloadSecurityData_SCF

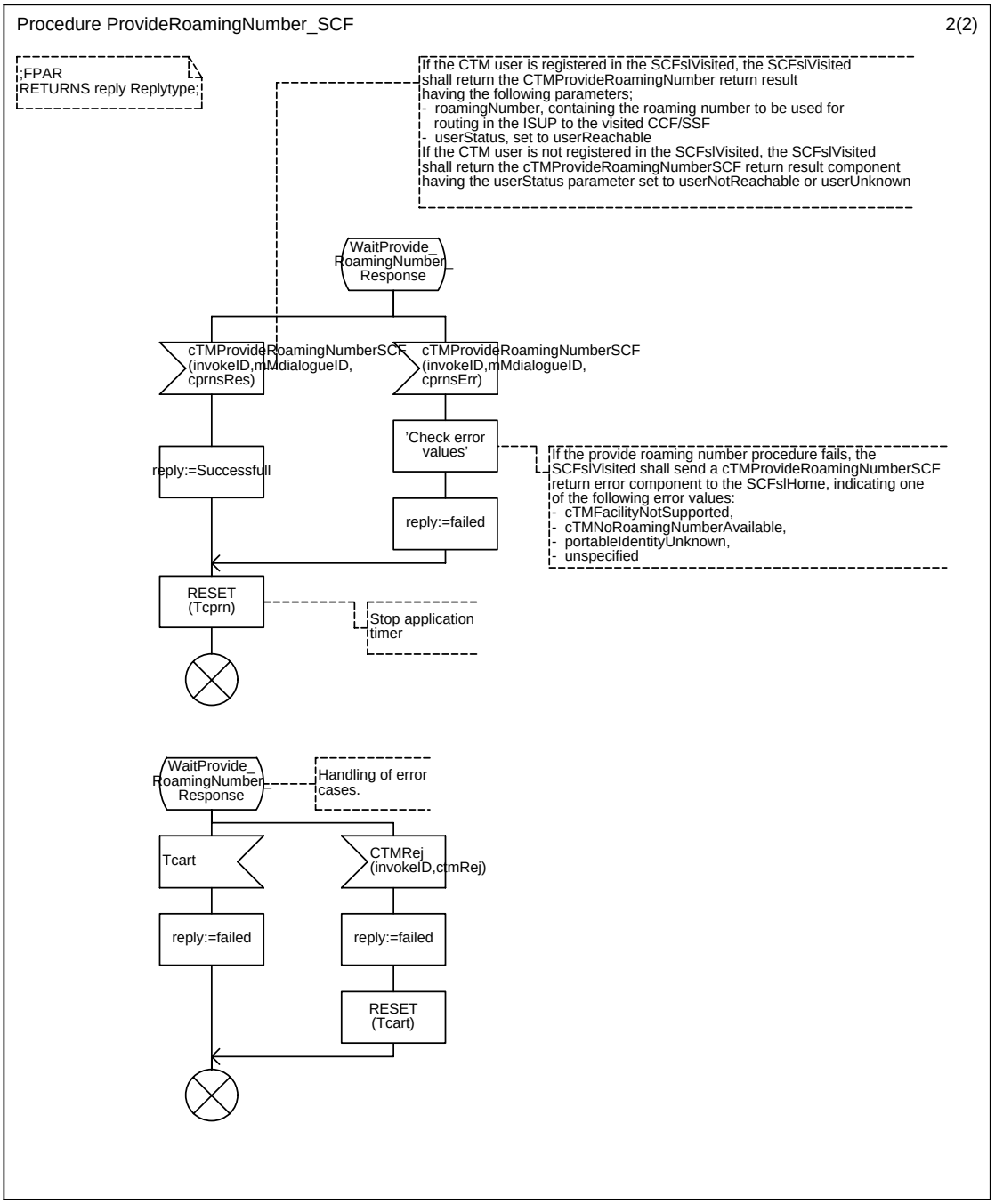


Annex C32 : Process Type SCFslHomeApplication



Annex C33 : Procedure ProvideRoamingNumber_SCF




Procedure ProvideRoamingNumber_SCF
2(2)
;FPAR
RETURNS reply Replytype;
If the CTM user is registered in the SCFslVisited, the SCFslVisited shall return the CTMProvideRoamingNumber return result having the following parameters;
- roamingNumber, containing the roaming number to be used for routing in the ISUP to the visited CCF/SSF
- userStatus, set to userReachable
If the CTM user is not registered in the SCFslVisited, the SCFslVisited shall return the cTMProvideRoamingNumberSCF return result component having the userStatus parameter set to userNotReachable or userUnknown
WaitProvide_RoamingNumber_Response
cTMProvideRoamingNumberSCF (invokeID,mMdialogueID, cprnsRes)
cTMProvideRoamingNumberSCF (invokeID,mMdialogueID, cprnsErr)
'Check error values'
If the provide roaming number procedure fails, the SCFslVisited shall send a cTMProvideRoamingNumberSCF return error component to the SCFslHome, indicating one of the following error values:
- cTMFacilityNotSupported,
- cTMNoRoamingNumberAvailable,
- portableIdentityUnknown,
- unspecified
reply:=Successfull
reply:=failed
RESET (Tcprn)
Stop application timer
WaitProvide_RoamingNumber_Response
Handling of error cases.
Tcart
CTMRej (invokeID,ctmRej)
reply:=failed
reply:=failed
RESET (Tcart)

Annex D (informative): Bibliography

- EG 201 096-1: "Intelligent Network (IN); Cordless Terminal Mobility (CTM); IN architecture and functionality for the support of CTM".

- EG 201 096-2: "Intelligent Network (IN); Cordless Terminal Mobility (CTM); IN architecture and functionality for the support of CTM; Part 2: CTM interworking between public Intelligent Networks".

- EG 201 096-3: "Intelligent Network (IN); Cordless Terminal Mobility (CTM); IN architecture and functionality for the support of CTM; Part 3: CTM interworking between private networks and public Intelligent Network".

History

Document history			
V1.1.1	January 1998	Public Enquiry	PE 9820: 1998-01-16 to 1998-05-15