ISITEP

D6.4.2 - PPDR TERMINAL APPLICATIONS DESIGN DESCRIPTION

Document Manager:	Marco Carli	RM3	Editor

Programme:	Inter System Interoperability for Tetra-TetraPol Networks
Project Acronym:	ISITEP
Contract Number:	312484
Project Coordinator:	Selex ES
SP Leader:	NETTECHN

Document ID N°:	ISITEP_D6.4.2_20151021_V1.0	**Version:**	V1.0
Deliverable:	D6.4.2	**Date:**	21/10/2015
		Status:	Approved

Document classification	**PUblic**

Approval Status	
Prepared by:	Marco Carli (RM3)
Approved by (WP Leader):	Federica Battisti (RM3)
Approved by (SP Leader):	George Mitsoupolos (NETTECHN)
Approved by (Coordinator)	Paolo Di Michele (SES)
Security Approval (Advisory Board Coordinator)	Etienne Lezaack (BFP)

CONTRIBUTING PARTNERS

Name	Company / Organization	Role / Title
Federico Frosali	SES	Revision
Claudia Olivieri	SES	Contributor
Federica Battisti	RM3	Contributor
Marco Carli	RM3	Contributor

DISTRIBUTION LIST

Name	Company / Organization	Role / Title
All Company Project Managers	All involved companies	Members of the Steering Committee
Elina MANOVA	EC DG REA	EC Programme Officer
General Public	NA	NA

REVISION TABLE

Version	Date	Modified Pages	Modified Sections	Comments
V0.1	22\09\2015	All	All	First draft (RM3)
V0.2	07/10/2015	All	All	Main Revision (SES)
V1.0	21/10/2015	All	All	Final release

Publishable extended abstract

This deliverable (D6.4.2) is issued by WP6.4. WP6.4 is in charge of defining the terminal side of the interoperability enabling applications, while the design of the terminal side is provided in WP6.5 and specifically in D6.5.2.

D6.4.2 “PPDR Terminal Applications Design Description” presents the design of the terminal components of the enhanced applications available within the ISITEP framework.

In particular, this document addresses the design of applications enabling interoperability that are described in the following:

- Dynamic Functional Numbering (DFN): this service allows to match the numbers of the operating talkgroups used in a country in the relevant numbers used for the same operating talkgroup in the neighboring country;
- Location Dependent Addressing (LDA): this application provides functional numbers to contact the right PPDR operational center using the GPS position of the terminal instead of the Location Area position;
- Enhanced Message Exchange (EME): this application provides the capability of exchanging messages between authorized end-users using a different native language exploiting the Semantic and Syntactic translator capabilities.

CONTENTS

1. INTRODUCTION

D6.4.2 “PPDR Terminal Applications Design Description” presents the design of the terminal components of the enhanced applications available in the ISITEP enhanced terminal (IET).

In particular, this document addresses the design of PPDR cloud added-value applications that are described in the following:

- Dynamic Functional Numbering (DFN): this is a service that allows to remap the numbers of the operating talkgroups used in a country in the relevant numbers used for the same operating talkgroup in the neighboring country;
- Location Dependent Addressing (LDA): this application is able to provide functional numbers to contact the local PPDR operational centers using the GPS position of the terminal instead of the Location Area position;
- Enhanced Message Exchange (EME): this application provides the capability to exchange messages between end-users using a different native language exploiting the Semantic and Syntactic translator capabilities (as detailed in the deliverables of WP5.5).

WP6.4 is developed taking as input the requirements related to the interoperability enabling applications defined in the ISITEP enhanced terminal requirements (WP 5.1). The outcome of WP6.4 is the design of PPDR Terminal Applications.

In the following sections each application is detailed together with the design description of their functionalities and an explanation of the relations between each application and the ISITEP framework.

2. DEFINITIONS AND ABBREVIATIONS

2.1. Definitions

This section is intended to capture the definitions of some key terms used in the document for the purpose of increased consistency. Most of the definitions are obtained from official 3GPP and ETSI documents:

Authentication: the act of positively verifying that the true identity of an entity (network, user) is the same as the claimed identity.

Confidentiality: the property that information may not be available or disclosed to unauthorized individuals, entities or processes.

Key: a sequence of symbols that controls the operations of encipherment and decipherment.

Migration: act of changing to a location area in another network (either with different Mobile Network Code and/or Mobile Country Code) where the user does not have subscription (e.g. ITSI in TETRA) for that network. In this document, migration is used as a synonym of roaming.

2.2. Abbreviations

For the purposes of the present document, the following abbreviations apply:

Acronym	Definition
HMI	Human Machine Interface

IET	ISITEP Enhanced Terminal
ISI	Inter System Interface
LIP	Location Information Protocol
MCC	Mobile Country Code
MNC	Mobile Network Code
PPDR	Public Protection and Disaster Relief
SDS	Short Data Service
SM	Security Manager
SST	Semantic and Syntactic Translator
SwMI	Switching and Management Infrastructure
TETRA	TErrestrial Trunked Radio
WFM	Workflow manager

3. SYSTEM OVERVIEW

In the ISITEP framework, operational interoperability among PPDR teams belonging to different countries relies on legal, operational, and technical improvements.

Among them, the enhanced ISITEP terminal (IET), based on TETRA and TETRAPOL modems, represent a key-factor for a real functionality integration.

The IET will be developed in both tablets (vehicular solution) and smartphones (hand-held solution) by exploiting the open source Android OS thus allowing the development of software applications for enhancing the user experience. In particular, a set of added-value applications has been designed for allowing the coordination of operative teams and to overcome language barriers when end users of different nationalities need to cooperate easily communicate among them both by voice and text-based media.

In particular, the PPDR cloud added-value applications, designed in this WP, are composed by a server and a client side. The server applications are deployed inside an operational unit connected to a hosting SwiMi while the client side is deployed inside the ISITEP enhanced terminal (IET).

A general application scenario is based on a network infrastructure (TETRA or TETRAPOL), a central room in which application servers are located and administrated, and a set of radio terminals (both tablets or smartphones) as equipment of the units operating on the field.

The PPDR cloud added-value client applications deployed in the IET are:

- Dynamic Functional Numbering (DFN);
- Location Dependent Addressing (LDA);
- Enhanced Message Exchange (EME).

The current work package will address the design specifications of DFN, LDA, and EME client side, while the corresponding server side applications are addressed in WP 6.5.

Among the provided applications, only EME is provided with a HMI and can be used upon request. On the other hand, DFN and LDA are operating in a seamless way for the final user.

In any case, there is a common framework offered to all ISITEP applications and it is described in this document.

In Figure 1, a general scheme of the system, in which the ISITEP applications are deployed, is shown.

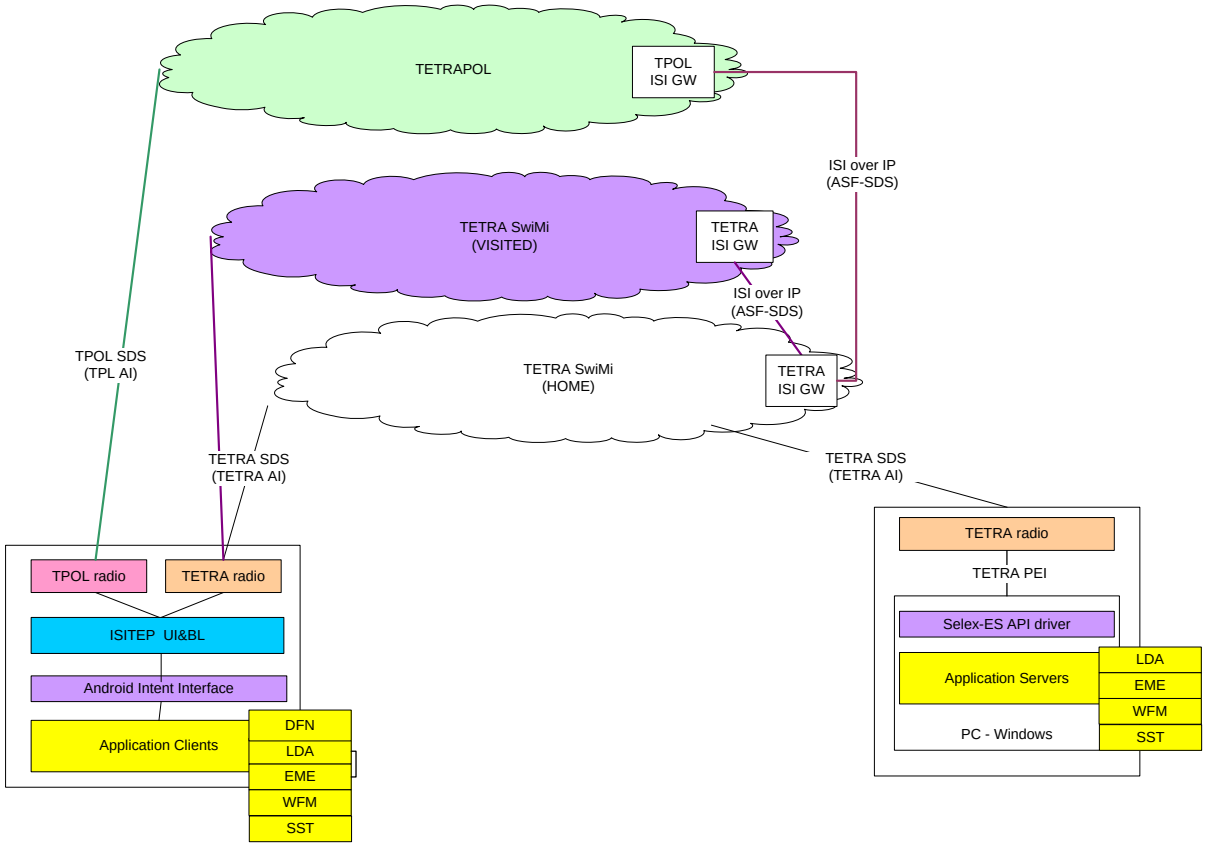

Figure 1: General framework in which the ISITEP enhanced applications are deployed.

4. DESIGN REQUIREMENTS

The need of coordination among all applications running on the terminal pushes for the definition of design requirements for the applications. Those requirements are listed in the following.

REQ#1 The ISITEP Cloud added-value Apps shall be able to exploit the SDS App IF exported by TETRA and TETRAPOL App in order to access to the unreliable data transport layer based on TETRA and TETRAPOL short data service.

REQ#2 The ISITEP Cloud added-value Apps shall be able to detect if the network service is available exploiting the Control IF exported by the Communication Manager App.

REQ#3 The ISITEP added-value Apps can be in one of two possible states: “Active” or “Deactivated”. They are in “Deactivated” state when network service is not available. The Communication Manager App informs the ISITEP Cloud added-value Apps on network service availability.

5. DESIGN DESCRIPTION

A common framework characterizes the applications designed for the IET.

In more details:

- an application has access to the Android Broadcast Bus;

- an application exploits the interfaces defined in D5.2.2 (it allows to verify the presence of communication service) and D 5.6.2 (it allows the EME exchange);
- an application is composed by an always running service and a HMI for user interaction;

From a logic functionality point of view, an application can be model as a two-state machine characterized by the following states:

- Active: the application can be used; if the user activates the corresponding HMI, all the functionalities of the application can be used.
- Inactive: the application cannot be used. This situation occurs when the user try to use the application during a handover procedure or when the radio coverage of both TETRA and TETRAPOL networks is missing. In this situation, the communication manager informs the app of no-coverage situation and the app display a message to the user.

It is important to underline that each service is always running in background, so that it can receive updates (messages) even if the corresponding HMI is not active.

5.1. Functional Description

5.1.1. Dynamic Functional Numbering

Dynamic Functional Numbering is a service that allows the identification of the number used by PPDR resources in charge of a mission in a specific area. When a terminal moves from one country to a neighboring country, in order to perform an international PPDR operation the Dynamic Functional numbering automatically updates the address book containing the international talk groups used in the visited country.

In the IET, the address books containing the group definition are separated for TETRA and TETRAPOL applications. TETRA modem contains the group definitions valid within the TETRA Network while TETRAPOL is equipped with group definitions valid in TETRAPOL network.

It is important to underline that DFN application is not used to change groups when the IET moves from TETRA to TETRAPOL and vice-versa, since when the terminal moves from one network to the other one, the operating modem is switched and therefore the used groups are changed too.

As prove of concept of the DFN application the group management in TETRA-to-TETRA migration scenario will be demonstrated.

The TETRA modem has the capability to define more separated “directories” where group subscriber identities are defined. Only one directory a time is active. One directory for each TETRA Network is defined. The task of the DFN is to select the proper active directory when the TETRA network changes.

The TETRA App and TETRAPOL App export the SMAppIF interface that enable the DFN to select the active directory whenever the MCC-MNC changes. DFN application is informed by TETRA App when the MCC-MNC changes.

Group directories are pre-configured in the TETRA modem, DFN has a configuration file that remaps the MCC-MNC with the correct directory.

5.1.2. Location Dependent Addressing

The LDA application is a client-server application enhancing the terminal capabilities. It provides the capability to assign a number of an operational center to a certain geographical area. In D 6.5.2, the LDA Server Application is described. The IET exploits the LDA Application server as localization system by sharing with the LDA Application server its GPS position using the LIP protocol. The LDA Server Application receives the GPS position by the terminal and track on the map its position.

As soon as the terminal moves to a different location, the LDA Server application sends, by using the LDA-protocol, an SDS to the terminal containing the telephone number of the operational center related to the area where the IET is located.

5.1.3. Enhanced Message Exchange

During PPDR operations, unexpected emergency scenarios shall be faced by many resources belonging to different PPDR organizations. Unexpected emergency scenarios may require PPDR joint action outside the pre-established patterns of a standard workflow. Often in this kind of emergencies, it is not clear how the emergency should be faced and the solution may be clear to some skilled persons but not to the entire PPDR organization. In this context, it would be very helpful if the useful information could be disseminated quickly and efficiently to the involved PPDR forces.

In an international context, the language barrier is an issue for fast and efficient communication. In this regard, one of the ISITEP project goals is to reduce language barriers in PPDR operations by deploying an Enhanced Message Exchange application.

The Enhanced Message Exchange application shall be used to provide written communications (i.e. orders or information) to the PPDR resources, which shall be translated into the proper language of the end-user. This application would help in overcoming the language barriers in international PPDR operations, where PPDR forces that speak different languages may compose intervention teams. Moreover, it has been verified that, often in the PPDR operations, there is high background noise that prevents from speech understanding. Therefore, written communications also in this case could help in improving understanding.

Written communications shall be "real-time" translated by a server application after the source and destination language have been detected.

For security reasons, only authorized end-users in charge for providing communications to a certain PPDR forces shall be allowed to send written communications using the Enhanced Message Exchange application to the relevant PPDR group.

The EME application is composed by an authentication center deployed on the server side and by an EME client deployed inside the IET.

Only those user authenticated to the EME authentication center are enabled sending EME messages, all the users are able to receive translated messages using the EME application.

In this paragraph it will be described the EME client application, for the EME authentication center design description refer to D 6.5.2 [4].

The EME client application is deployed inside each terminal. At the start up the end-user, that is enable to send translated text, shall provide his credentials on the EME application, from that moment until the IET is switched off the EME application is enabled also to send messages.

The EME application is invoked by the end-user clicking on the EME icon, selecting the send message menu the end-user is enable to write the text and select the destination identity.

The EME application HMI is automatically prompted to the end-user whenever a translated text has been received. EME application service is in charge to interface the Semantic and Syntactic Translator (SST) providing the text to translate the source language and the destination language.

SST provides the translation back to the EME application that prompt the translated text message to the end-user.

5.2. Application architecture

ISITEP added value App requires the availability of the PPDR services on the IET platform to the Communication Manager App using the commServiceAvailRequest (D5.2.2). The Communication Manager App provides the information on the service availability in the commServiceAvailIndication (D5.2.2). When the CM provides that service is not available, the ISITEP added Value App shall move into deactivated state. End-user is informed by the ISITEP added value app when the service is not available.

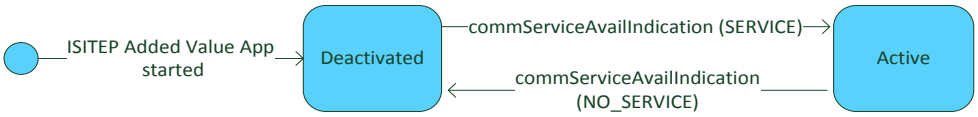

Figure 2 ISITEP Added Value App State Machine

In the deactivate status, the behavior of the DFN, LDA, and EME applications is the same: no service is provided to the end user and they cannot be started by touching on the screen the corresponding icon. This status is get when:

- the startup or reset of the CM Service starts;
- during TETRA-TETRAPOL handover
- when TETRA and TETRAPOL coverage is not available

In deactivated state, the corresponding icons appear in gray scale to inform the user about the unavailability of the system.

5.2.1. Dynamic Functional Numbering

DFN is activated upon IET start. This service asks the CM to send updates on the status of the communication services by using the *PlatformIF* described in D5.2.2. The CM notifies the DFN when a MCC/MNC change happens. For each MCC/MNC supported by the current available network, the DNF can choose from a list all available groups on that network and pushes the active app (TETRA App or TETRAPOL App) to change groups through the SMAppIF described in §5.3.3.

The DFN is not provided with an HMI and it is composed by:

- Broadcast Messages Handler: this is the interface of the DFN towards the other entities in the terminal. It is used for receiving updates on MCC/MNC changes and to push the TETRA\TETRAPOL app to update the group list. This is performed through the SMAppIF If that is detailed in §5.4.
- Configuration file: it contains the list of available MCC/MNC supported by the network
- DFN core: it is in charge of listening and understanding the data received by the Broadcast Receiver and, in case a change in MCC/MNC is registered, the configuration file is checked and the request towards the TETRA\TETRAPOL app is performed for activating the corresponding group directory.

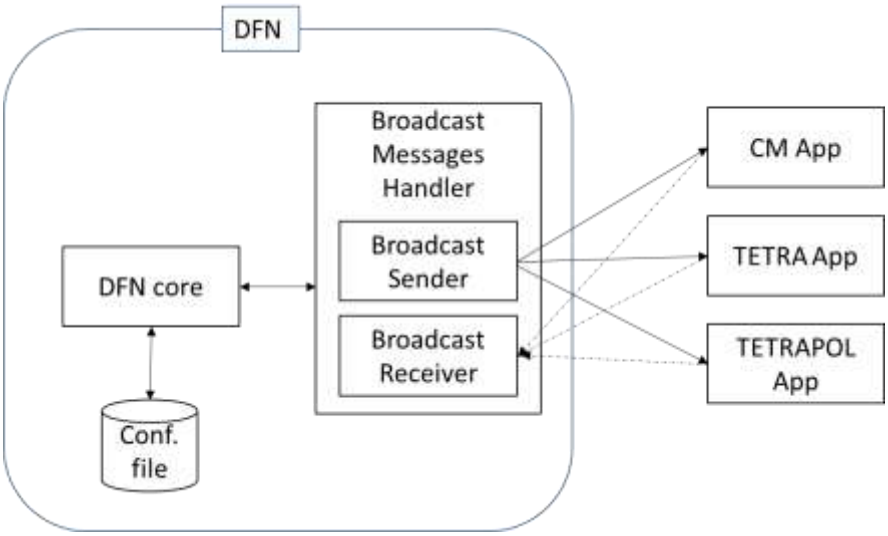

Figure 3: DNF app architecture

5.2.2. Location Dependent Addressing

LDA is activated upon IET start. LDA asks to the CM App to be notified on the available communication services and listens to messages sent from the LDA server through the SDS App IF (described in D6.5.2). When it receives a message addressed to its own appID, it updates the address book through the SMAppIF descrived in §5.3.3.

The LDA is not provided with an HMI and it is composed by:

- Broadcast Messages Handler: this is the interface of the LDA towards the other entities in the terminal. It is used for receiving the message that requests the TETRA\TETRAPOL app to update their address book. This is performed through the SM App If that is detailed in §6.4.
- LDA core: it verifies if the recipient of the incoming message corresponds to the terminal appID, and, if the message requests an address book update, it is performed through the SM App IF.

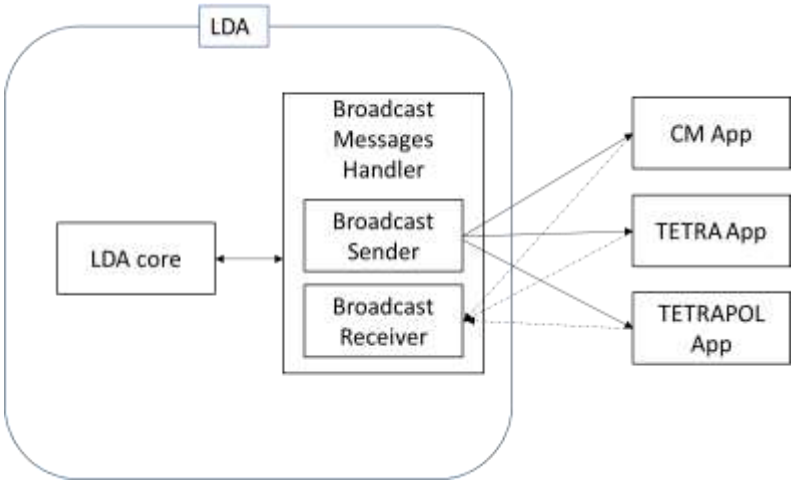

Figure 4: LDA app architecture

5.2.3. Enhanced Message Exchange

The service provided by the EME app is activated upon IET start.

EME asks to the CM App to be notified on the available communication services. Until a communication service is not available, it is not possible to activate the EME HMI.

The EME application is based on a HMI, a client and a server side service. The client side is divided in two entities: EME sender and EME receiver.

The EME application exploits the SDS App IF (as described in D5.2.2) to send and receive messages to be translated and to receive translated messages.

At message reception, the user can display and read the content. After the first reading, the EME application stores, in a local DB, the content of the received (and sent) messages, together with the sender-receiver details. In order to protect the message confidentiality, before storage, the messages are encrypted. If the user needs to access those encrypted data, he will have to input a combination of username-password for authentication purposes that will be dealt with by the Security Manager services.

The Security Manager is also invoked during the startup phase. In more details, the EME sender sends to the EME server the UserID. The EME server will verify the received ID with the list of authorized users and, based on the results of this check, it will send back to the EME client an authorization token. If the token is positive, the EME sender is authorized to send EME messages and in the EME HMI, the “send” button is enabled.

Otherwise, the EME sender is disabled, the menu is disabled and the user can only receive enhanced messages.

The authentication process is performed by exploiting the digest authentication scheme. By default the MD5 algorithm is used. In more details:

1. The EME client sends the EME server a request for an access-protected resource.
2. The EME server receives the request and, based on the RequestID recognizes that the request is for the EME delivering service that requires authentication. The EME server checks whether the client has provided authentication information. The server responds by returning an EME to the client, including a header with the following information: timestamp, nonce and an HTML tag that references the Uniform Resource Indicator (URI) requested, and it is protected in the server’s private key. The message body is a 401 message code, indicating that the client’s request has generated an unauthorized access error.
3. The EME client receives the server’s challenge and gathers the required credentials. The EME client’s challenge response is then hashed with the user’s secret key, including the original nonce protected with the user’s secret key, the user name and domain. The original nonce and timestamp sent by the server, the number of times this number has been used, and a new random number, called a nonce number, which is unique and is used only once. To prevent replay attacks nonce counts are used.
4. The EME server receives the message and it verifies that the challenge response refers to a challenge issued by the server itself, and that the nonce number has not been used before in challenge responses, to ensure that this is not a replay attack. Then it checks if the ID is included in the pre-authorized list and if the check is successful, an EME message with an authorization token is returned to the client.

It is useful to underline that the password needed for retrieving stored data is different from the ID need to enabling the EME sender.

Figure 5: EME app architecture

5.3. Interface Description

In the current section for the completeness of the design description, a general description of the interfaces exported or used by the ISITEP Added Value Apps has been reported; for a detailed description of the interfaces refer to D 5.5.2 [2] and D 5.6.2 [3].

5.3.1. Control IF

The ControlIF is implemented on top of the Android platform by means of Intents. As depicted in the following Figure 6, this interface is decomposed into:

- CCmAServiceControlIF, which is used to transfer the signaling from the Communication Manager App and the TETRA and TETRAPOL App.
- CCmAServiceControlListenerIF, which is used to transfer the signaling from the TETRA and TETRAPOL Apps to the Communication Manager App.
- CCmAPlatformServicesIF, which is used to transfer signaling from ISITEP Added Value Applications to Communication Manager App.
- CCmAPlatformServicesListenerIF, which is used to transfer the service availability of the enhanced terminal platform to the ISITEP Added Value Apps.
- CCmAAppControlIF, which is used to transfer the signaling from the ISITEP Added Value Apps to the TETRA/TETRAPOL App.

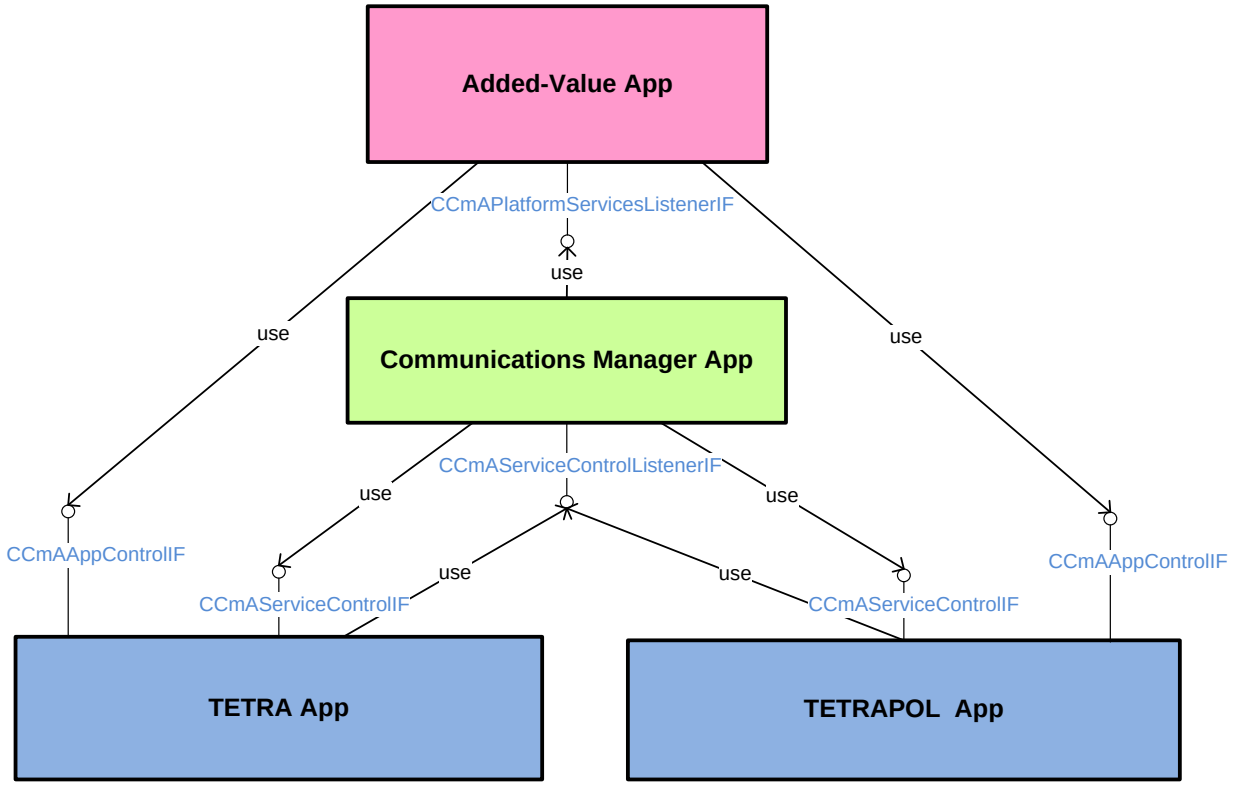

Figure 6: Block diagram of the Control interfaces

All the interfaces exported or exploited by the Communication Manager App (CCmAServiceControlIF, CCmAServiceControlListenerIF, CCmAPlatformServicesIF and CCmAPlatformServicesListenerIF) are described in D 5.2.2 [2].

The CCmAAppControlIF is described in D 5.6.2 [3].

5.3.2. SDS App IF

The SDS App IF is implemented on top of the Android platform by means of Intents.

In the following diagram the two ways of the interface have been out lighted:

Top-down way:

CCmAMessageIF is the interface exported by the TETRA App and used by Added-Value applications to send messages over the TETRA / TETRAPOL radio channel.

Bottom-up way:

CCmAMessageListenerIF is the interface that is exported by Added-Value applications in order to receive notifications relevant to incoming messages on the TETRA / TETRAPOL radio channel.

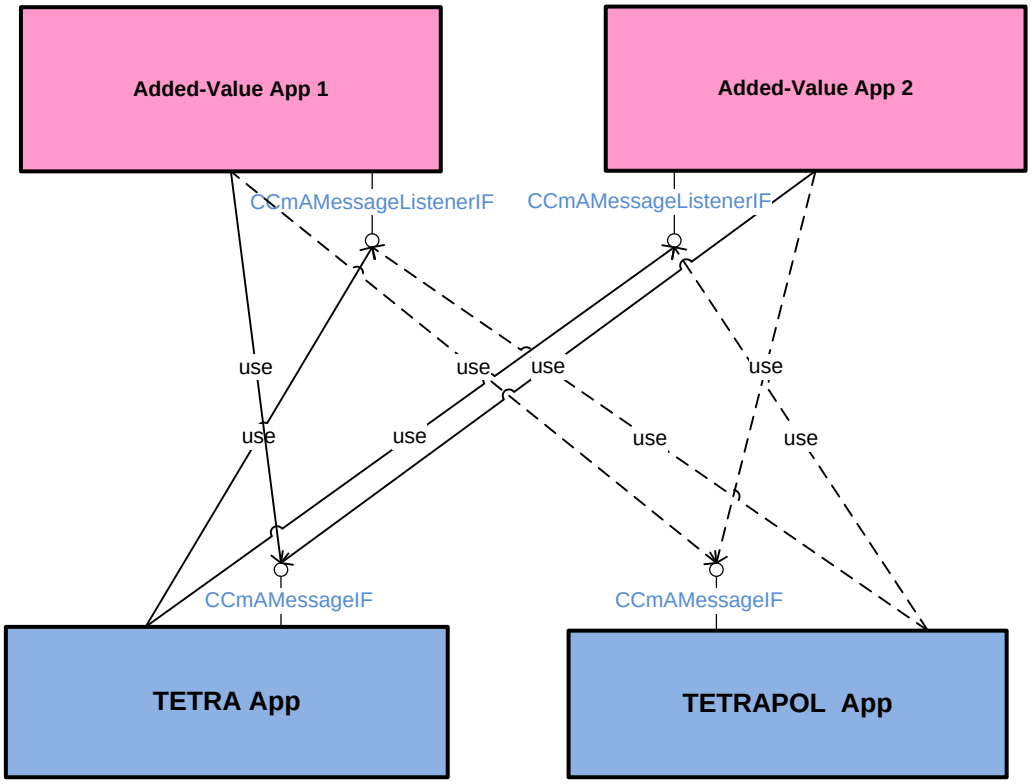

Figure7: Interface Identification

The Detailed description of CCmAMessageIF and CCmAMessageListenerIF is provided in D 5.6.2 [3].

5.3.3. SMAppIF

The SMApp IF is implemented on top of the Android platform by means of Intents. This interface is exported by TETRA and TETRAPOL App in order to allow the LDA and DFN application to modify the IET address book.

In the following diagram the two-ways of interface have been out lighted:

Top-down way:

CCmASMAppIF is the interface exported by the TETRA App and used by LDA/DFN applications to modify the IET address book.

Bottom-up way:

CCmASMAppListenerIF is the interface that is exported by LDA/DFN applications in order to receive acknowledge to the address book modification.

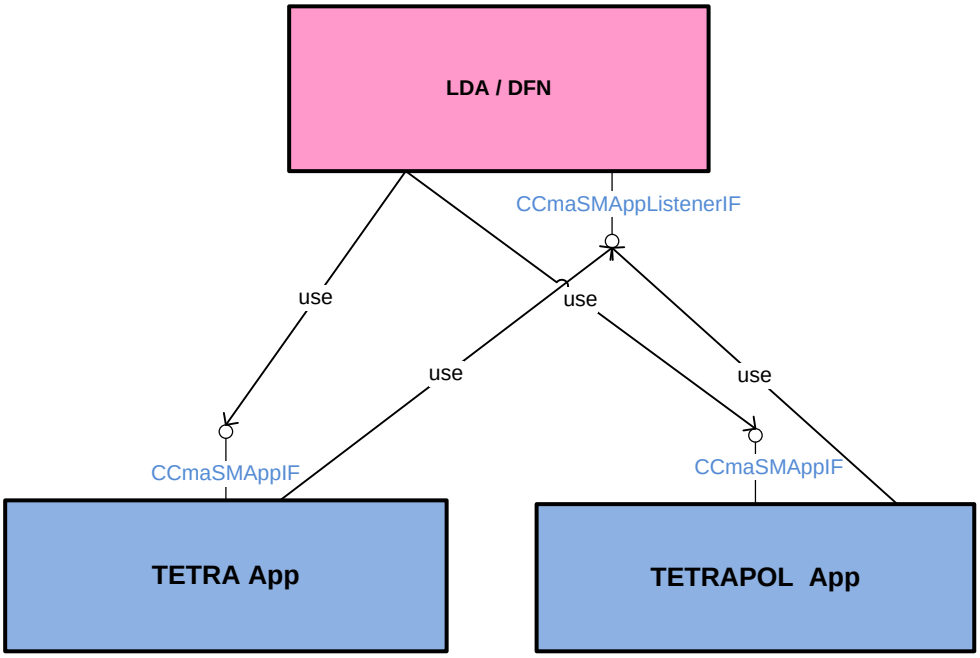

Figure 8: CCmaSMAppIF interface identification

5.3.4. CCmASMAppIF

5.3.4.1. changeGroupListRequest (int mcc, int mnc, int groupListID)

This command is used by the DFN to request to the TETRA and TETRAPOL App to change the active group list in use in the current network.

The following parameters shall be populated when sending the changeGroupListRequest:

- int mcc: the Mobile Country Code (MCC) of the serving network.
- int mnc: the Mobile Network Code (MNC) of the serving network.
- Int groupListID: this is an integer that identify the group list to be used for the MCC-MNC where the IET is registered.

5.3.4.2. modifyGroupNumberRequest (String groupName, String groupNumber)

This command is used by the LDA in order to request to the TETRA and TETRAPOL App to change, in the address book, the telephonic group address for a certain groupName.

The following parameters shall be populated when sending the modifyGroupNumberRequest:

- String groupName: the group Name registered inside the address book
- String groupNumber: telephonic group subscriber identity of the network entity to be linked to the groupName.

5.3.5. CCmASMAppListenerIF

5.3.5.1. changeGroupListConfirm (String serviceProviderName, Boolean acknack)

This command is used by TETRA/TETRAPOL App in order to confirm to the DFN App that the group list has been changed (successfully or not).

The following parameters shall be populated when sending the *changeGroupListConfirm*:

- String serviceProviderName: String that identifies the radio channel TETRA or TETRAPOL.
- Boolean acknack: set to TRUE when the active group list has been successfully changed, set to FALSE when it has not been changed.

5.3.5.2. changeGroupNumberConfirm (String serviceProviderName, Boolean acknack)

This command is used by TETRA/TETRAPOL App in order to confirm to the LDA App that the group identity has been changed (successfully or not).

The following parameters shall be populated when sending the *changeGroupNumberConfirm*:

- String serviceProviderName: String that identifies the radio channel TETRA or TETRAPOL.
- Boolean acknack: set to TRUE when the group number has been updated, set to FALSE when it has not been updated.

5.4. SST IF

The SST IF is implemented on top of the Android platform by means of Intents.

In the following diagram the available interfaces are described

CCmSSTAppIF is the interface exported by the TETRA App and TETRAPOL App and used by the SST application to send messages over the TETRA / TETRAPOL radio channel.

CCmaSSTListenerIF is the interface that is exported by the SST application to receive notifications relevant to incoming messages on the TETRA / TETRAPOL radio channel.

TSListener IF is the interface exported by the SST application to receive translated messages from the EME app.

TSIF is the interface exported by the EME and used by the SST application to send messages to be translated.

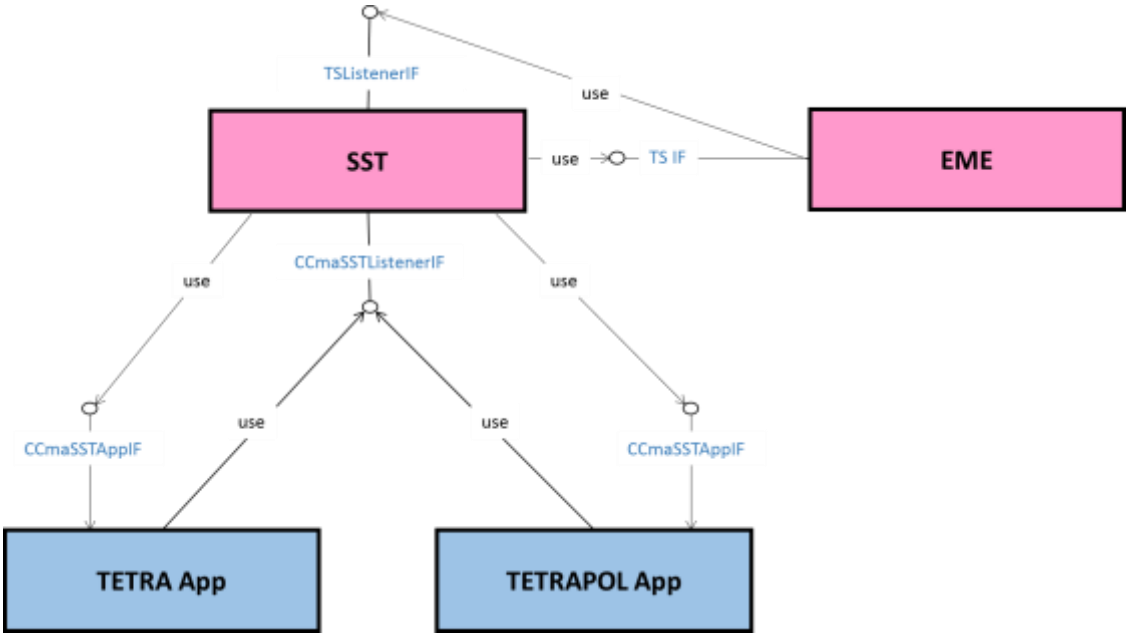

Figure 9: SST IF

5.5. TSListener IF

5.5.1. SSTTranslationReq (int SL, int RL, string message, int userID, int appID)

This command is used by the EME app for requesting to the SST the translation of a message.

The parameters used in this command are:

- int SL: is a code associated to the language of the sender
- int RL: is a code associated to the language of the receiver
- string message: is the message to be translated
- int userID: is the identification number of the user
- int appID: is the identification number of the app that is requesting the translation service.

5.6. TSIF

5.6.1. SSTranslationResult (string Tmessage, int userID, int appID)

This command is used by the SST app for sending the translated message.

The parameters used in this command are:

- string Tmessage: is the message translated by the SST
- int userID: is the identification number of the user
- int appID: is the identification number of the app that is requesting the translation service.

5.7. LDA Application Protocol

LDA application protocol exploits the standard TETRA localization protocol provided by the TETRA modem, LIP defined in document [5] par. 29.5.12.1. Periodically the TETRA modem provides the information of the GPS position to the LDA Server. The LDA Server is configured to provide an « LDA-SDS update number » to the LDA client when the IET moves from one geographical area to another geographical area.

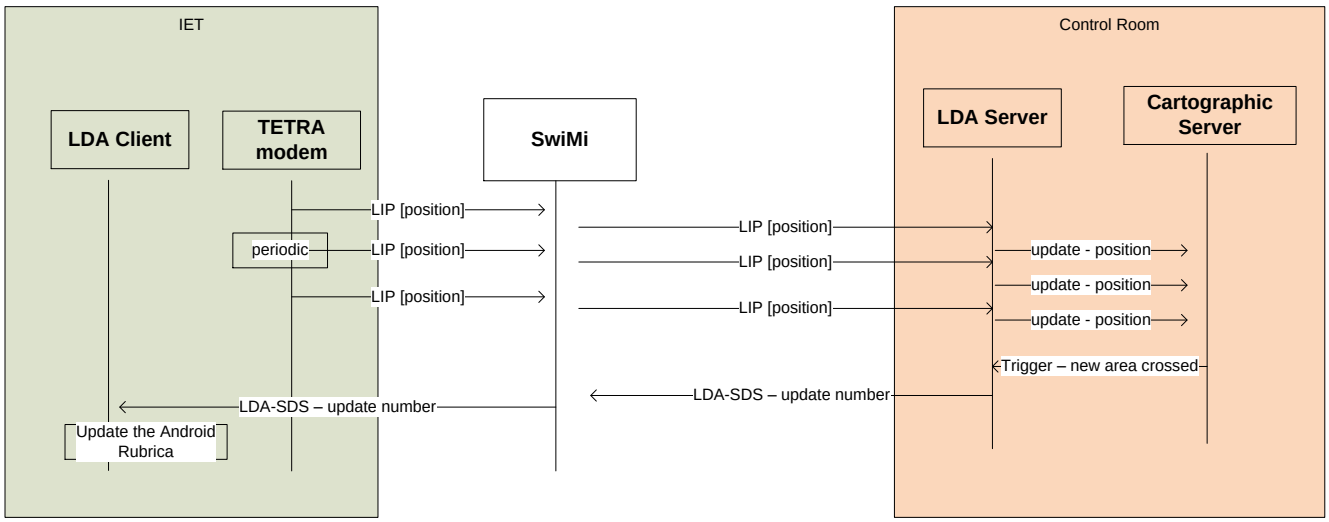

Figure 10: LDA Message Sequence

5.7.1. LDA-SDS update number

The LDA-SDS update number is provided through the Android Intent interface to the TETRA / TETRAPOL App using the SDS App If see par. 6.2.

The “LDA-SDS update number” message has the following field transported over a text message.

Name	Value	Note
AppProt	1	LDA
MessID	1	Update Number
Service	String	Talkgroup Name
MCC	<0 … 999>	Destination MCC [6]
MNC	<0 … 9999>	Destination MNC [6]
SSI	<0 … 16777215>	Destination SSI [5]

5.8. EME Application Protocol

5.8.1. EME client to EME Authentication Center

This interface is described in document D.6.5.2 [4].

When the user tries to run the EME app, an authentication procedure starts.

5.8.2. EME sender to EME receiver

The EME sender is able to send a message to the EME receiver exploiting the SDS App If see par. 6.2. Only authenticated user are authorized to send an EME message, the EME sender then sends and EME-SDS-TextSent to the EME receiver deployed on a different terminal.

In order to translate text message the SST client deployed on the receiving IET may need to send requests to the SST Server in the Control Room.

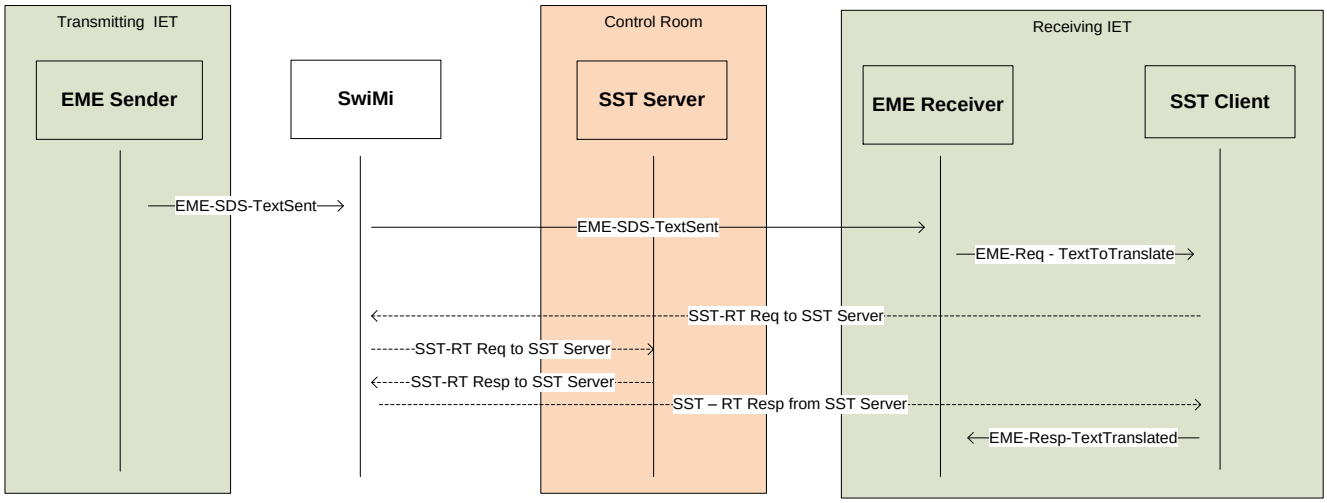

Figure 11: EME message sequence completed with SST interaction

5.8.3. EME-SDS-TextSent

The EME-SDS-TextSent is provided through the Android Intent interface to the TETRA / TETRAPOL App using the SDS App If see par. 6.2.

The Android Intent interface is invoked using a configurable protocol identifier, as described in D5.6.2 PID shall have a value inside the [192 – 254] range, In order to avoid conflict with PID values already adopted by different vendors the PID value shall be configurable on the application.

The short text message will contain EME protocol fields and the text provided by the end-user.

EME protocol fields are reported in the following table:

Field	Value	Note
AppProt	2	EME
MessID	3	EME-SDS-TextSent
Src		Language Code according ISO 639.
Dst		NULL

Text		Maximum number of character 100

EME:

Src=de,Dst=NULL, Dies ist ein Beispielbeitrag

5.8.4. EME-Req-TextToTranslate

The EME-Req-TextToTranslate is provided through the Android Intent interface to the SST Client see par 5.4.

The EME Receiver compares the language ID of the incoming enhanced message with the language ID set by the terminal user. If the IDs do not correspond, the SST client is invoked to start the message translation. In particular, the source language ID, the destination language ID and the text to be translated are given to the Semantic and Syntactic Translator.

Src=de,Dst=fr, Dies ist ein Beispielbeitrag

Field	Value	Note
AppProt	2	EME
MessID	4	EME-Req-TextToTranslate
Src		Language Code according ISO 639.
Dst		Language Code according ISO 639.
Text		Maximum number of character 100

5.8.5. EME-Resp-TextTranslated

The Semantic and Syntactic Translator returns to the EME receiver the translated message using the EME-Resp-TextTranslated message on the Android Broadcast Bus used the interface specified at par. 5.4 . The EME Receiver prompt the translated message.

“Ce est un example d’apres”

In the following table the EME-RespTextTranslated message structure:

Field	Value	Note
AppProt	2	EME
MessID	5	EME-Resp-TextTranslated
Text		Maximum number of character 100

6. REFERENCES

[1] D 5.1.1 Enhanced Terminal Requirements
[2] D 5.5.2 Adaptation / Communication Manager Design Description
[3] D 5.6.2 User Interface and Business Logic Manager Design Description
[4] D 6.5.2 PPDR Server Applications Design Description
[5] ETSI EN 300 392-2 Terrestrial Trunked Radio (TETRA); Voice plus Data (V+D); Part 2: Air Interface (AI)
[6] ITU-T Recommendation E.218 - Management of the allocation of terrestrial trunk radio Mobile Country Codes