ETSI GS MTC 009 V1.1.1 (2010-05)

Group Specification

Mobile Thin Client (MTC); Architecture

Reference

DGS/MTC-0002 Arch

Keywords

access, adaptation, architecture, mobile,
multimedia, remote

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - 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

Important notice

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

Copyright Notification

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 ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI 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 ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Group Specification (GS) has been produced by ETSI Industry Specification Group (ISG) Mobile Thin client Computing (MTC).

Introduction

The present document is intended to define a technical framework of the mobile thin client system architecture. This includes basic building blocks, components detailed at different levels and appropriate interfaces between them.

1 Scope

The present document presents the architecture for a mobile thin client system, consistent with the requirements articulated in the work item "Mobile thin client system requirements" of the MTC ISG. The deliverable first describes a high-level architecture, in order to position the approach taken, and subsequently details the functional architecture at three different levels. Components and their interactions are identified, together with their interfaces. Non-normative scenarios indicating a possible implementation of the required functionality, are included as annex to the present document.

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

2.1 Normative references

The following referenced documents are necessary for the application of the present document.

[1] ETSI MTC 008: " Mobile Thin Client (MTC); Use Cases and Requirements".

2.2 Informative references

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

Not applicable.

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

Service Management Framework (SMF): to manage the thin client service, a whole range of management components are required (such as authentication and authorization, network management, business management, session management, server management, monitoring, etc.). The collection of these management components is called the Service Management Framework (SMF). It should be noted that for scalability and efficiency reasons, the components of the SMF will be distributed. Some of the management components will run on the client, some on the server hosting the thin client session and some on dedicated management servers

thin client protocol: is responsible for delivering user input (such as keystrokes, mouse events but also data from for example webcams, etc.) to the server and sending audiovisual output from the server to the client

thin client service: is a service offering remote execution of applications. While the application logic is actually running on a remote server, output is transmitted to the terminal the user interacts with. Likewise, the user input is captured at the terminal, and transmitted to the remote server for proper handling. This basic service can be supplemented with additional functionalities, such as authentication, remote data storage and management, automatic application installation, etc.

3.2 Abbreviations

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

ADS	Application Delivery Service
DSS	Data Storage Service
E2E	End to End
E2EAUTHN	End to End authentication
E2EAUTHZ	End to End authorization
E2ECONN	End to End connection
E2ELOGISTICS	End to End logistics
ENV	ENVironment
ESS	Entity Subscription Service
I	Interface
MTC	Mobile Thin Client (implementing the client part of the service)
MTH	Mobile client Hosting environment
MTS	Mobile Thin Server (implementing the server part of the service)
NET	NETwork
NETC	NETwork Control
QOS	Quality of Service
SDM	Self Data Manager
SLM	Self Logistics Manager
SMF	Service Management Framework
TCS	Thin Client Service
TCSC	Thin client service core
TCSCC	Thin client service core core (i.e. the core of the TCSC)

4 High-level architecture

4.1 Overview

In Figure 1, the high-level software architecture is illustrated. Six basic building blocks can be distinguished: the Mobile Thin Client, the Mobile Thin Server, the Service Management Framework, the Network Control, the Data Storage Service and the Application Delivery Service.

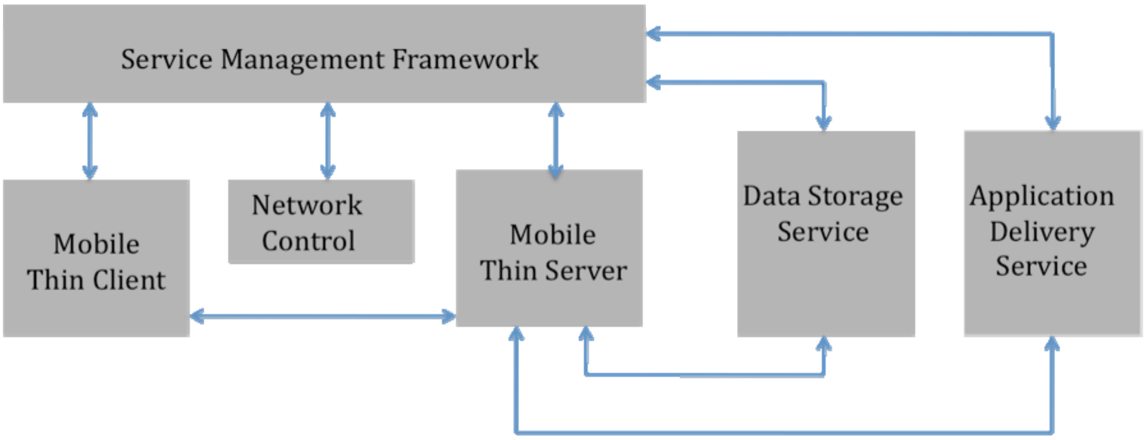

Figure 1: High-level software architecture

In Figure 1, only one client (i.e. Mobile Thin Client) and one server (i.e. Mobile Thin Server) are visualized. The multi-user aspect of the architecture is shown in Figure 2. Multiple clients can be connected to the same physical Thin Client Server running a separate User Session for each connected user. To monitor and manage the Thin Client Server, a part of the Service Management Framework (i.e. the Thin Client Server Management component) is installed at every Thin Client Server. The Mobile Thin Server runs on top of the Thin Client Server Operating System.

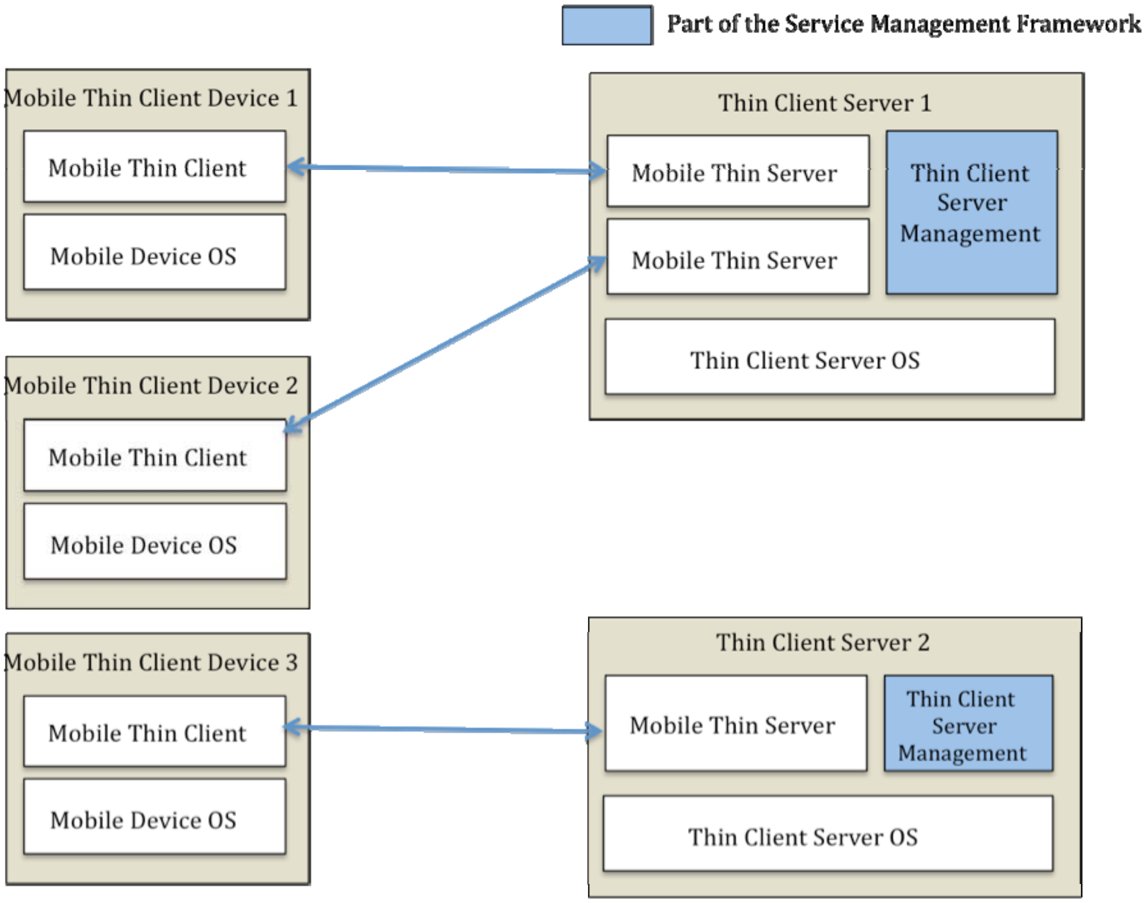

Figure 2: Multi-user aspect of the system Architecture

4.2 Basic Building Blocks

From a high-level perspective, the functionality of the 6 basic blocks is summarized Table 1. In clause 5, an exhaustive description of the functionality of all basic blocks will be discussed.

Table 1: Basic building blocks of architecture

Basic Building Block	Description
Mobile Thin Client	This is the software running on the mobile client device. When a user wants to start the thin client service, he starts this client on his thin client device.
Service Management Framework (SMF)	When a user logs in, he has to identify himself to the SMF, which is responsible for managing the complete thin client service and guaranteeing the desired QoS to the users. The components of the SMF are distributed over the different building blocks.
Mobile Thin Server	This is the software running on the Thin Client Server selected by the SMF. All applications of the users are executed in their Mobile Thin Server. Audiovisual output from a user's applications is transported over the network to the user's device. Input from the user is transported in the opposite direction.
Network Control	This is the service of the network operator. The SMF will interact with the Network Control block to set the appropriate QoS classes for the traffic between the Mobile Thin Client and the Mobile Thin Server.
Data Storage Service	This service maintains the personal data of the users.
Application Delivery Service	For scalability reasons (not every application should be installed in every Mobile Thin Server), applications are delivered by an application delivery service to the Mobile Thin Server.

4.3 Philosophy of the architecture

A first introduction of the system architecture is shown in Figure 3. Details on the functional aspects of the system architecture will be given in clause 5. Two major components can be distinguished: the Mobile Thin Client (MTC) and the Mobile Thin Server (MTS). Both entities communicate with each other over the network and exchange thin client protocol messages and management data. The architecture is quite symmetric, with some additional components on the MTS side to support specific services (e.g. data storage service).

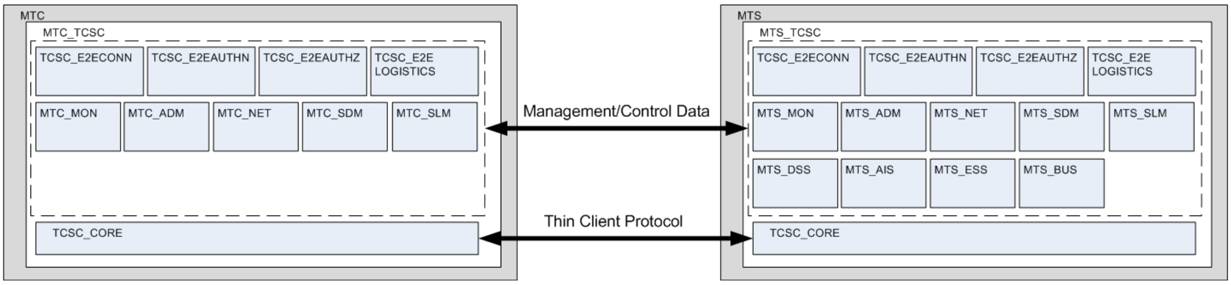

Figure 3: Overview of the System Architecture

A short description of the components of Figure 3 is given in Table 2. Note that the functionality or implementation might differ between the MTC and MTS side. This rather high-level description can help the reader to understand the composition of the overall system architecture.

Table 2: Components of the System Architecture

TCSC_E2ECONN	Starts a connection between MTC and MTS, and thus acts as a portal to the system.
TCSC_E2EAUTHN	Authentication of MTS and MTC.
TCSC_E2EAUTHZ	Authorization of MTS and MTC.
TCSC_E2ELOGISTICS	Negotiates session settings, reserves resources, …
MTS_MON	Monitors the state of the MTS.
MTS_ADM	Handles administration tasks.
MTS_NET	(De)packetizes data, sends and receives network packets (general network functionality), including network control functionality.
MTS_SDM	Contains data about the system, and provides a means for interacting with it.
MTS_SLM	Handles self management tasks such as resilience.
MTS_DSS	Endpoint to deal with data storage service.
MTS_ADS	Handles application delivery from application delivery service.
MTS_ESS	Handles access to the entity subscription service.
MTS_BUS	Provides required information for external business systems.
TCSCC	Handles the core communication between MTC and MTS. This basically means the thin client protocol communication, functionally equivalent to e.g. VNC. RDP, NX, …

5 Functional Architecture

5.1 General Overview

As already stated in the requirement document [1], the mobile thin client service as an entity would operate within the following three main environments:

The **Terminal Device environment** is composed of the terminal hardware and software. This is the environment where the Mobile Thin Client will operate.

Th**e Hosting environment** is the environment, where the Mobile Thin Server will operate. This environment is supposed to provide components that are not included in the system scope but that are necessary for the Mobile Thin Server to communicate with the external world. For instance, components needed for configuring routers are not in the scope of the system. Another example is that the service should provide billing data to external billing systems but should not include a billing system. So if a billing system is needed it should be considered as part of the environment of the system.

Network environment is the pipe that links the two previous environments. This environment is composed of one or many networks. This environment is taken into account because the Mobile Thin Client and Server will need to interact with it, for instance to ask for specific network QoS support. A goal of the service is to allow end-to-end network adaptation in order to provide the best possible end user experience combined with the best possible energy economy.

5.2 Elaboration of the architecture

5.2.1 Introduction to the Architecture

To elaborate on the architecture, we proceed using an approach that will consist of:

- Elaborating the top level (level 0) architecture layer.
- Recursively building detailed layers (level 1 up to level 3): to achieve this, each functional block in a previous layer will be split into deeper sub-blocks, according to the sub-functionality they provide. This will be done only when it is appropriate. For instance, at some level, some blocks will not need to be more detailed, so they will appear in a layer as they are in less detailed layer.

In the present document, the top level layer will be considered as the layer level zero. The immediately underlying layer will be considered as the layer level one, etc.

5.2.2 Top level architecture description (level 0)

For the sake of simplicity, the architecture is conceived as a traditional client server architecture where the client is a piece of software running in its own environment and the server is also a piece of software running in its environment. The two pieces communicate with each other through the network. This is illustrated in Figure 4.

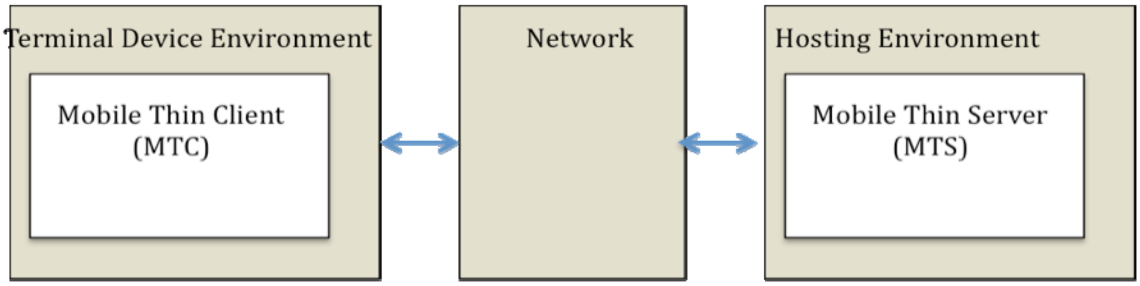

Figure 4: Top Level Architecture

5.2.2.1 Network considerations

As far as the network is concerned, the Mobile Thin Client and the Mobile Thin Server will provide two high level types of data:

- Control Data: the destination of this data is the network itself, meant to be interpreted by the network.
- End-to-end data: for this type of data, the network will transmit the data unmodified between the two end points.

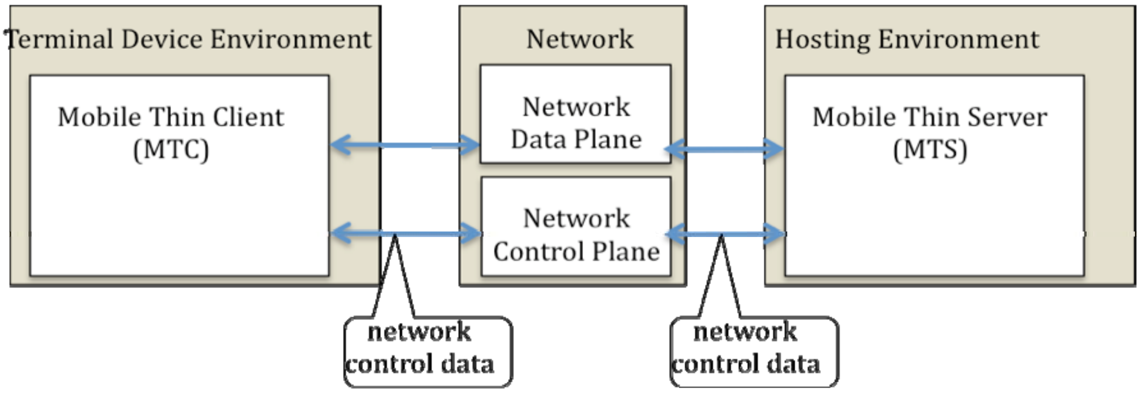

Figure 5: Types of data

For network control data, it is the responsibility of the network to provide the needed interfaces for the system to be able to set the adequate parameters. The Mobile Thin Client and Mobile Thin Server will include components that will ensure providing data for network control.

According to Figure 5, two interface categories are expected to be provided by the network:

- The interface needed for end-to-end communication: this interface will be named I_NET_E2E.
- The interface needed for network control: this interface will be named I_NET_NETC.

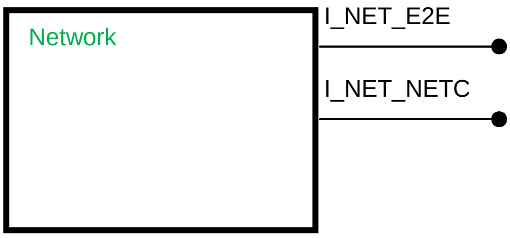

Figure 6: Network interfaces

The interface named I_NET_E2E will most likely be composed of a collection of standard network interfaces. This interface may have nothing special to the system so it may not be of special interest. The interface named I_NET_NETC will deal with the network control: it will most likely be composed of a collection of existing network control interfaces. The system has to know all about this interface in order to support it. In the remainder of the present document we will consider a generic network interface providing both network access interface and network control interface.

5.2.2.2 End point considerations

Since the system endpoints (i.e. MTS and MTC) are interconnected through the network, they need to interface with the two interfaces from the network operator as described in the previous clause. This means each endpoint needs two high level interfaces: one for network communication (NET) and one for communication with the other end point (E2E).

Table 3: Interfaces from MTS and MTC

Communication type	Interfaces to be provided by MTS	Interfaces to be provided by MTC
End-to-End	I_MTS_E2E	I_MTC_E2E
to Network	I_MTS_NET	I_MTC_NET

Each interface is related to at least one of the internal components of each end point. For the two high level interfaces described above, similar components are thus required. This means for MTC the components MTC_NET and MTC_E2E are dealing with communication over respectively the interfaces I_MTC_NET and I_MTC_E2E. The same components are required on MTS (I_MTS_NET and I_MTS_E2E).

To illustrate the purpose of the interfaces I_MTC_NET and I_MTC_E2E, we will consider a small scenario where the Mobile Thin Server wants to send video output to the Mobile Thin Client. Therefore, MTS_E2E creates the video images then the end-to-end data transfer process takes place as follows:

1) MTS_E2E hands off the images to MTS_NET.

2) MTS_NET sends the images over the network (via I_NET_E2E interface).

3) The packets go through the network and finally arrive at MTC_NET (through I_MTC_NET).

4) The component MTC_NET will see that the packets contain E2E data and will handle over the data to MTC_E2E (through I_MTC_E2E).

5) Finally, the component MTC_E2E receives and processes the video images and the result is shown on the client's screen.

Another functionality required for each endpoint is the ability to communicate with the environment of the endpoint. In other words, the MTS and MTC should be able to know about the environment in which they are active and the system should deal with changes in these environments in order to optimize the quality of experience for the users. For example, when an external screen is attached to the user's terminal, this is informed to the component MTC_ENV through interface I_MTC_ENV to allow the system to adapt to the changed terminal device environment.

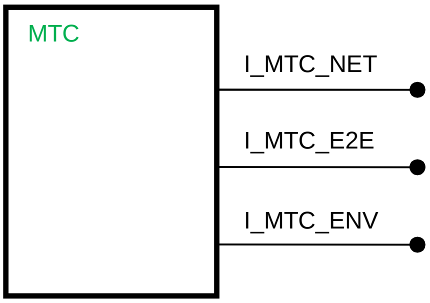

Figure 7: Mobile Thin Client interfaces

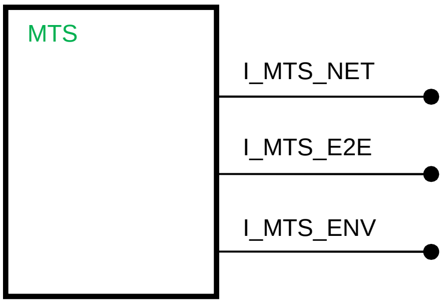

Figure 8: Mobile Thin Server interfaces

5.2.2.3 Self Management

Each system end point needs its self management that will mainly deal with the following tasks:

- Starting, stopping and coordinating internal components.
- Acquiring needed dynamic resources and freeing them when appropriate.
- Optimizing the user experience by adapting the system to changes in the environment.

The component that will deal with this task will be called "Self Manager".

5.2.2.4 Self Data

Each system end point needs persistent data for its own needs (configuring and tuning its self components, user data for internal usage only).

The component that will deal with this task will be called "Self Data Manager".

5.2.2.5 Top level view of the architecture

Based on the previous clauses, the top level view of the architecture can be summarized as follows. Considering the system endpoints (i.e. MTC and MTS), a generic architecture has been derived. The interfaces of an endpoint are illustrated in Figure 9. An endpoint has 3 interfaces to the external world:

- I_MT_NET for network communication with the network operator's interfaces I_NET_NETC and I_NET_E2E.
- I_MT_E2E for communication between two endpoints.
- I_MT_ENV to exchange information about the state of the endpoint's environment.

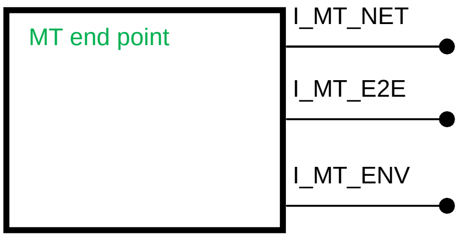

Figure 9: System endpoint interfaces

Not only the interfaces of an endpoint can be described in a generic way, also the components of an endpoint are generic. The components of an endpoint are illustrated in Figure 10. Each endpoint consists of 5 generic components:

- MT_NET to handle incoming and outgoing packets from interface I_MT_NET.
- MT_E2E to handle communication between two endpoints (such as audiovisual information and user input) through interface I_MT_E2E.
- MT_ENV to communicate with the endpoint's environment (such as device capabilities) over interface I_MT_ENV.
- MT_SLM to handle self management task (such as resilience mechanisms).
- MT_SDM to deal with self data management tasks (such as access to logging information for business support).

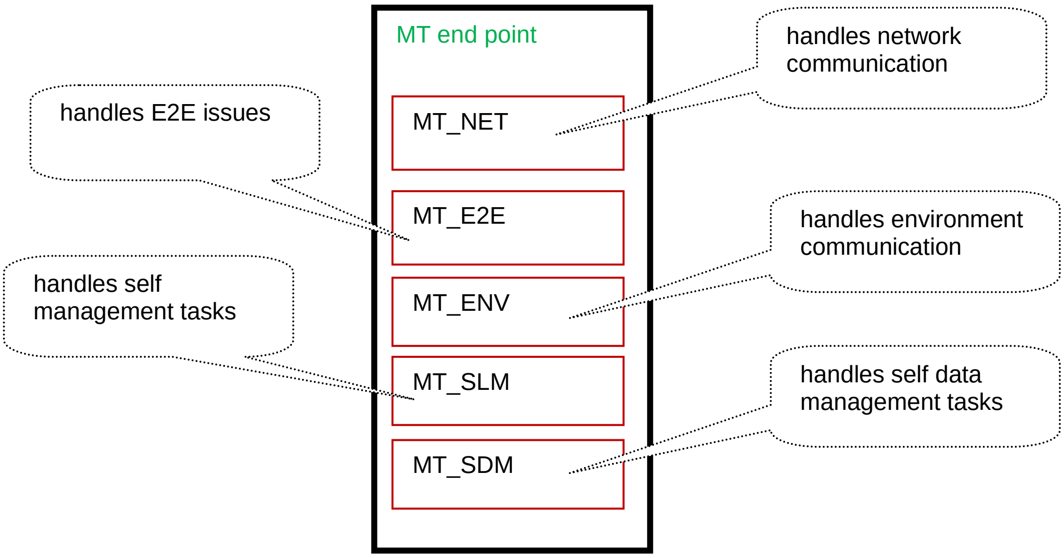
MT end point
handles network communication
handles E2E issues
MT_NET
MT_E2E
handles environment communication
handles self management tasks
MT_ENV
MT_SLM
handles self data management tasks
MT_SDM

Figure 10: System endpoint components

5.2.2.6 Top level architecture

Figure 11 presents a complete overview of level 0 of the architecture.

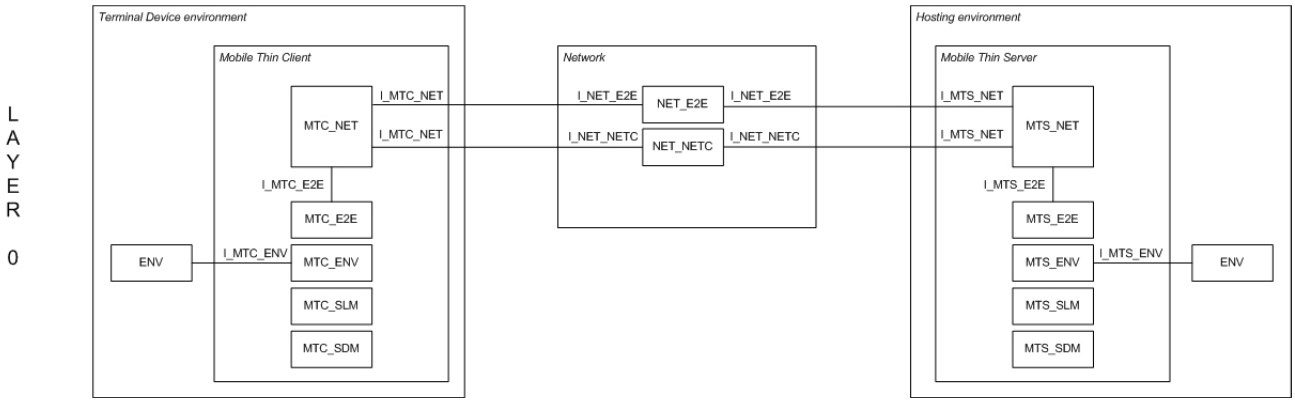

Figure 11: Level 0 architecture overview

5.2.3 Level one architecture description

5.2.3.1 End-to-end communications

For end-to-end communications, each system endpoint will provide needed interfaces to the other end point.

Table 4: End-to-End communication interfaces from MTS and MTC

Communication type	Interfaces to be provided by MTS	Interfaces to be provided by MTC
End-to-End	I_MTS_E2E	I_MTC_E2E

From the requirements step, the end-to-end communication category might be divided into three sub-categories:

- The Thin Client Service Core (TCSC) communication deals with the body of the Thin Client Service (TCS). It encompasses all the communications being actively involved in delivering thin client services. This includes end-to-end connection set-up and teardown, end-to-end authentications, end-to-end protocols for delivering audio data, video data, etc.
- Monitoring communication concerns with monitoring of system endpoints (i.e. monitoring of battery load of client device or CPU load in the hosting environment).
- Administration communication deals with the administration of system end points (i.e. administration user profiles, connection of data storage service, etc.).

Table 5 provides the resulting interfaces.

Table 5: End-to-End communication interfaces

Communication type	Interfaces to be provided by MTS	Interfaces to be provided by MTC
TCSC	I_MTS_TCSC	I_MTC_TCSC
TCS monitoring	I_MTS_TCSMon	I_MTC_TCSMon
TCS administration	I_MTS_TCSAdm	I_MTC_TCSAdm

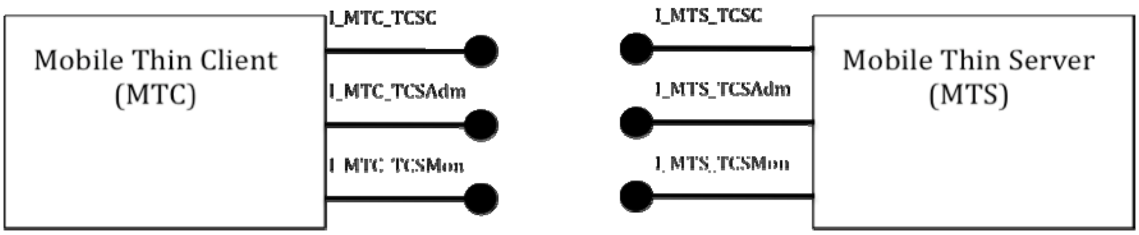

Figure 12: End-to-End communication interfaces

5.2.3.2 Mobile Thin Server and its hosting environment

The Mobile Thin Server expects that its hosting environment will make its hardware and software resources available for its usage. In particular, the following data are expected:

- Application delivery data includes all data required for the installation and execution of an application. It is transferred by the Application Delivery Service via the I_MTH_ADS and I_MTS_ADS.

- Private User data is the data from a user stored and transferred in the data storage service via I_MTH_DSS and I_MTS_DSS.

- Entity subscription data, e.g. user subscription data, is accessed via the interfaces I_MTH_ESS and I_MTS_ESS.

The server is expected to provide the following data to the system hosting environment:

- business data;

- optional data might be:
  - monitoring data;
  - administration data.

Each type of previous data will be given through a specific interface.

The way this data will be exchanged will be determined is an implementation choice. For now we will consider each component to provide all the interfaces to import and export the data.

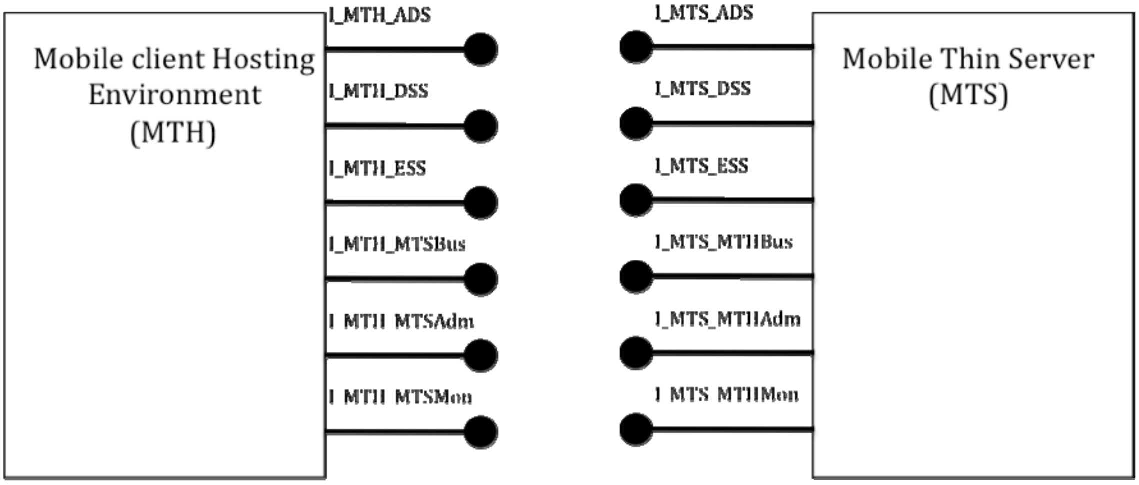

Figure 13: Mobile Thin Server and its hosting environment interfaces (detailed)

NOTE: More generally, data exchanged might be classified into two categories: data to export and data to import. So to simplify, associated interfaces might be generalized to:

- I_MTS_Export: this is the collection of interfaces that export data from the Mobile Thin Server.
- I_MTS_Import: this is the collection of interfaces that import data to the Mobile Thin Server.
- I_MTH_Export: this is the collection of interfaces that export data from the hosting environment.
- I_MTH_Import: this is the collection of interfaces being used to import data to the hosting environment.

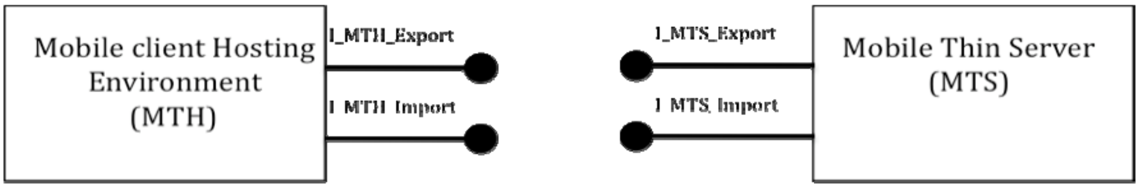

Figure 14: Mobile Thin Server and its hosting environment interfaces (generic)

5.2.3.3 System architecture: level one

The previous clauses brought up the identification of the following blocks for both Mobile Thin Client and Mobile Thin Server.

Table 6: Required components for Mobile Thin Client and Server

Component name	Description
Network control block	This is the block that will handle network control tasks.
Self Data manager (SDM)	This is the block that will manage self persistent data for configuring and tuning system self components.
Self logistics manager (SLM)	This is the block that will be responsible for the self logistics tasks (mainly coordinate different internal blocks, acquire/free resources, etc.).
TCSC block	This is the block that will handle the core of thin client sessions.
TCS monitoring block (TCSMon)	This is the block that will handle monitoring tasks between the Mobile Thin Client and the Mobile Thin Server.
TCS administration block (TCSAdm)	This is the block that will handle administration tasks between the Mobile Thin Client and the Mobile Thin Server.

For the Mobile Thin Server we have identified the following additional components.

Table 7: Optional components for Mobile Thin Client and Server

Component name	Description
ESS block	This is the block that will handle the access tasks to the entity subscription location (e.g. user subscription).
ADS block	This is the block that will handle the access tasks to the Application Image Servers.
DSS block	This is the block that will handle the access tasks to the DSS.
BUS block	This is the block that will provide needed information for external business systems (billing systems).

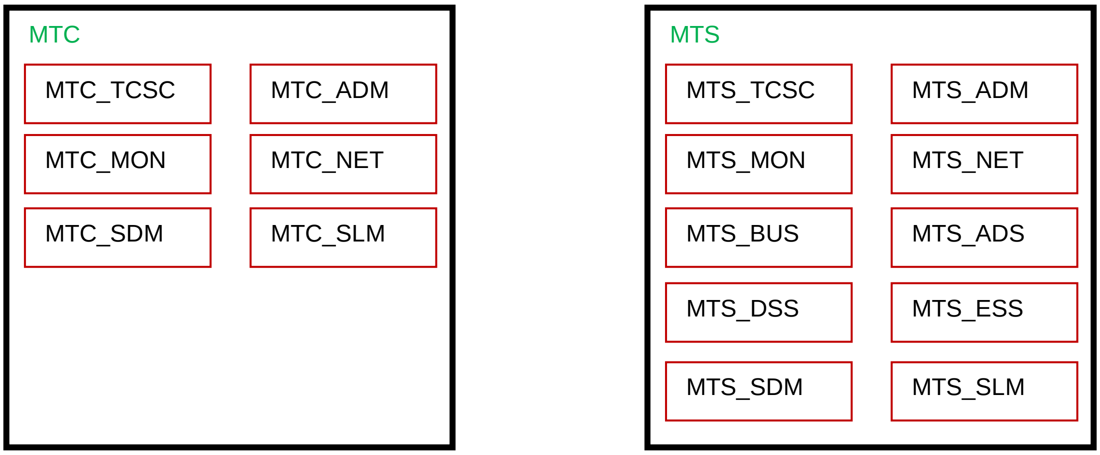

Figure 15: For Mobile Thin Client and Server

5.2.3.4 Level one architecture

Figure 16 presents a complete overview of level 1 of the architecture.

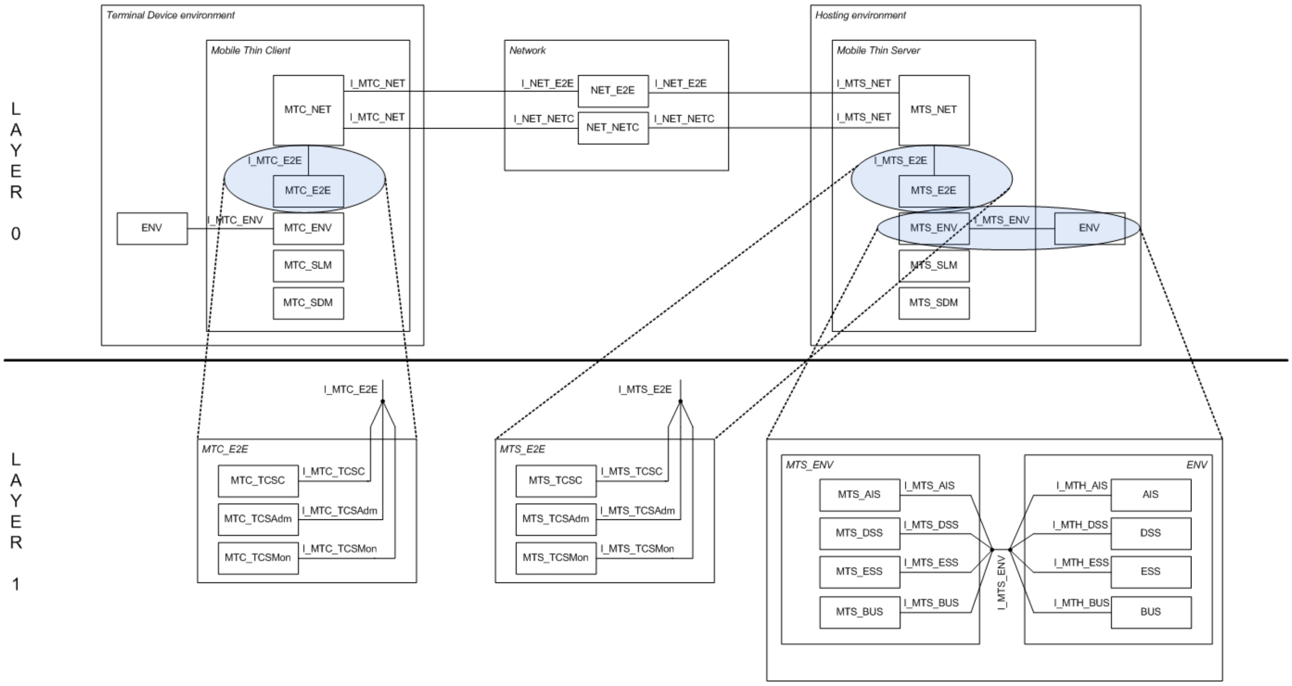

Figure 16: Level 1 architecture overview

5.2.4 Level two architecture description

5.2.4.1 Introduction to level two

The core interaction process between MTC and MTS is described through the following steps:

- E2E connection step: In this step, an endpoint will initiate a connection to the other endpoint. This is a sort of hand-shaking step.

- Entity authentication: In this step both entities will proceed to the authentication of each other. In this step, the authenticator will just verify that the other end point entity is really what it claims to be.

- Entity authorization: In this step the entity "authorizer" will check that the connecting entity is authorized to go further and use the thin client service.

- Logistics issues: In this step, the MTC and MTS will negotiate the session settings (their respective capacity, the initial required QoS, session token, etc.) and reserve adequate resources for the ongoing session. References of the real attachment points for pure service delivery operations should be provided. For instance, if the MTS operates in listening mode then the MTS should provide the address of the server and socket numbers where the MTC should connect. If it is the MTC who will operate in listening mode then the MTC should provide its address and its attachment points and the MTS should connect to that MTC points. Attachment points might be multiple (for instance, one point for video data, one point for printer data, one point for audio data, etc.).

- Service utilization: this step encompasses the body of the session. The delivery of the service expected to be provided by the system will be done here. This is the core of TCSC (TCSCC) and the internals of this component are not specified.

- E2E Session Termination: this step terminates the E2E session between the two end-points. It might be under the initiative of one or the other end-point.

5.2.4.2 Architecture of TCSC

Both MTS core and MTC core will:

- participate in connection establishment. This will be done through a connection handler;
- need to authenticate/authorize the other end point;
- participate in settings negotiation and logistics;
- participate in the service session body;
- participate in the termination of a session. This will be done through a connection handler;
- need to deal with their self management;
- need to access their proper self data.

So basically, at this point, the logical view of TCSC architecture is similar for MTS and MTC (symmetric architecture). Figure 17 shows the composition of a generic TCSC.

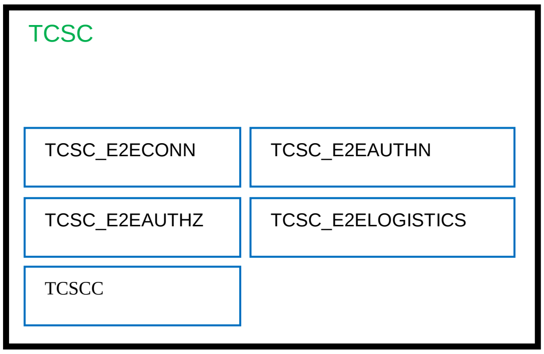

Figure 17: TCSC components

Of course each component of TCSC will provide needed information for administration, monitoring, billing and logging. TCSC will need its self management and its proper tuning data. This will be done through interfaces being exported by internal system components.

5.2.4.3 Level two architecture

Figure 18 presents a complete overview of level 2 of the architecture.

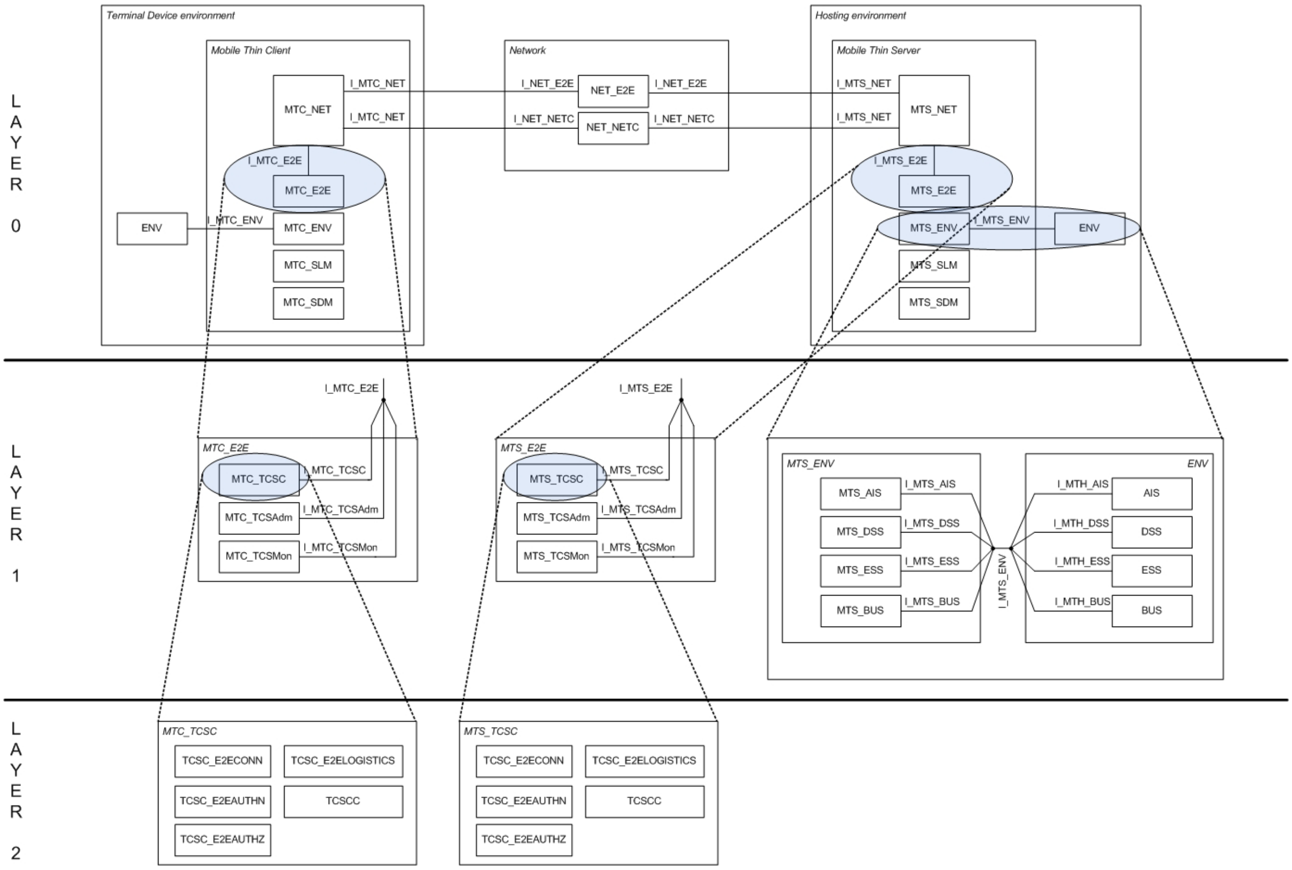

Figure 18: Level 2 architecture overview

Annex A (informative): Scenario mapping

In annex A of [1], five scenarios were described to connect the thin client system to "real" life. From these scenarios, several use cases can be extracted such as starting and stopping the service, starting an application and migrating a session.

The interactions between the 6 basic building blocks in the different use cases are illustrated with a sequence diagram in Figures A.1 to A.3

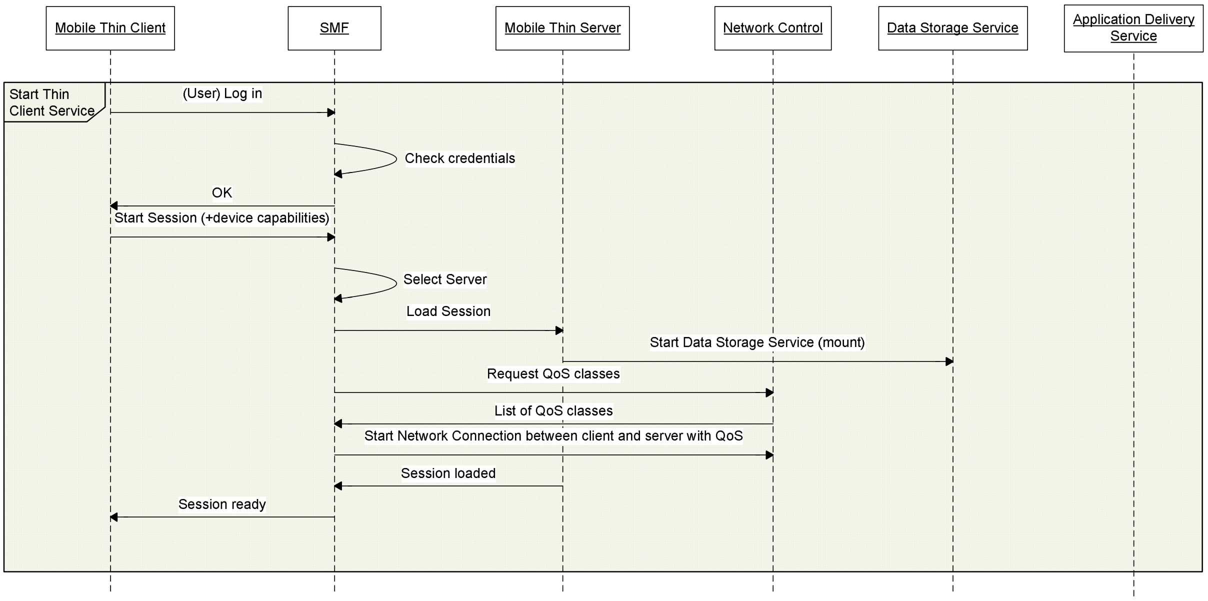

Figure A.1: Sequence diagram "Start the thin client service"

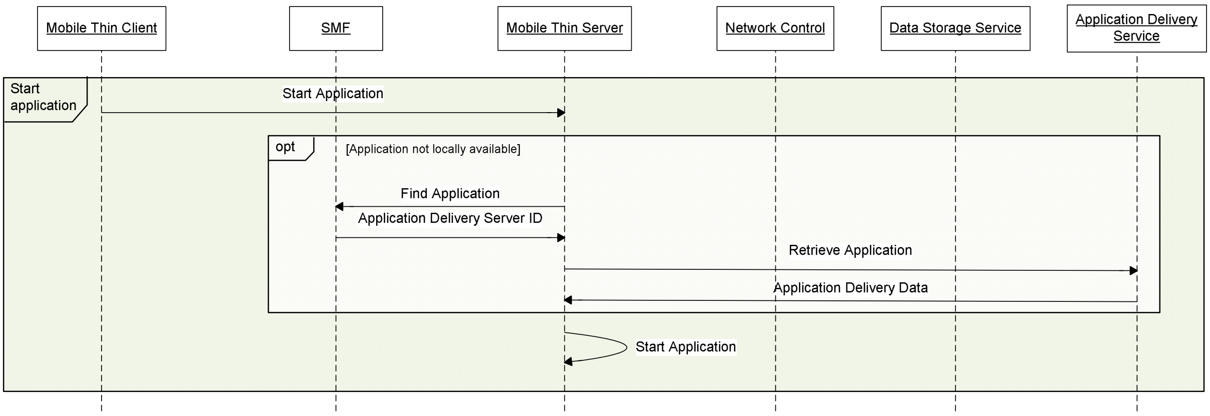

Figure A.2: Sequence diagram "Start application"

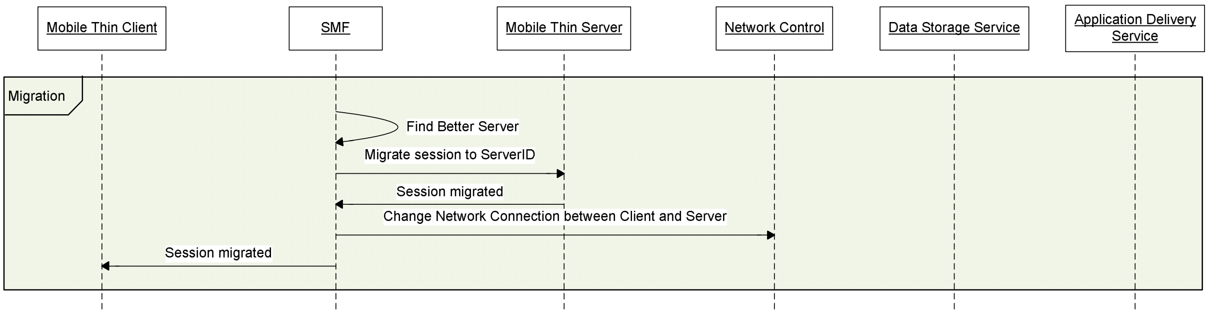

Figure A.3: Sequence diagram "Migration"

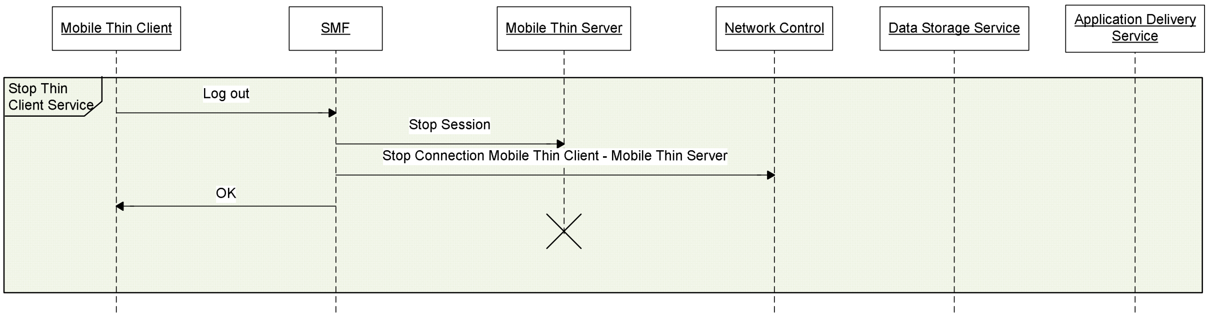

Figure A.4: Sequence diagram "Stop thin client service"

An exemplarily mapping of the basic building blocks and high-level interfaces is described for a first scenario: Bringing the hospital to the patient's home and the second scenario: Getting in touch with Paris. The other scenarios can be mapped in a similar way. Not to overload the mapping, the details of the use cases presented before are replaced by a "high-level message".

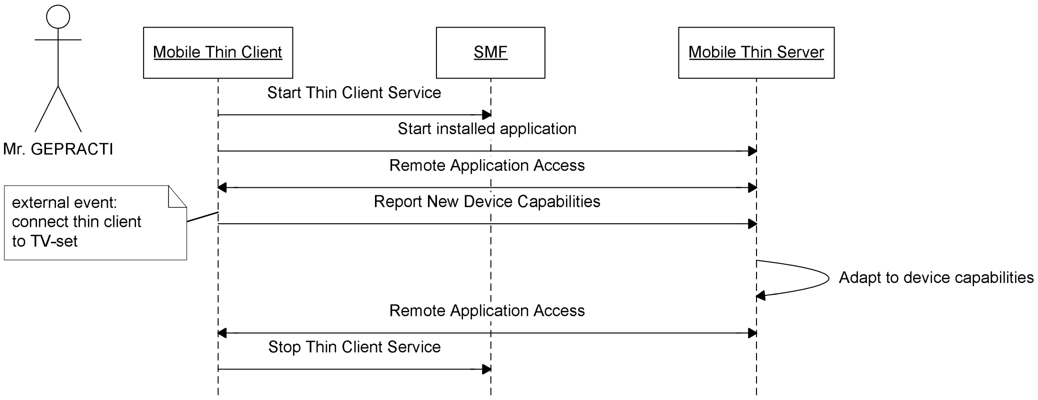
Mr. GEPRACTI
Mobile Thin Client
SMF
Mobile Thin Server
Start Thin Client Service
Start installed application
Remote Application Access
Report New Device Capabilities
external event: connect thin client to TV-set
Adapt to device capabilities
Remote Application Access
Stop Thin Client Service

Figure A.5: Scenario 1: Bringing the hospital to the patient's home

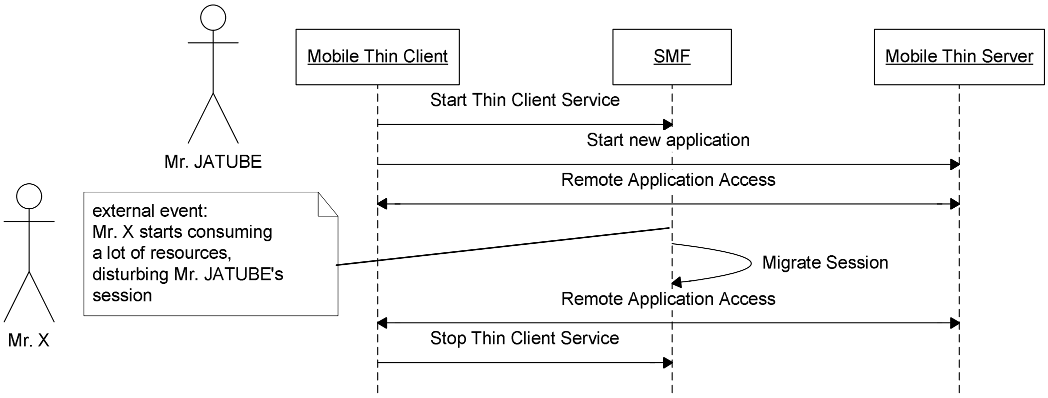

Figure A.6: Scenario 2: Getting in touch with Paris

History

Document history		
V1.1.1	May 2010	Publication