3GPP TR 22.857 V6.0.0 (2002-12)

Technical Report

3rd Generation Partnership Project;
Technical Specification Group Terminals;
Runtime Independent Framework Feasibility Study;
(Release 6)

Keywords
UMTS, MExE

3GPP

Postal address

3GPP support office address
650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet
http://www.3gpp.org

Contents

Foreword

This Technical Report has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

- x the first digit:
  - 1 presented to TSG for information;
  - 2 presented to TSG for approval;
  - 3 or greater indicates TSG approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

Introduction

This document discusses the need for a Runtime Independent Framework, what it is, and how it can be provided with a minimum of changes to the existing specification.

The references to the MExE Stage 2 specification, 3GPP TS 23.057, in this TR are based on the section numbers in version 5.0.0 of the MExE specification found at:

http://www.3gpp.org/ftp/Specs/latest/Rel-5/23_series/23057-500.zip

The information and opinions in this document reflect the discussions of the 3GPP T2 SWG1 (MExE) starting with input to the SWG meetings at the T2#17 Plenary and T2#18 Plenary.

One document that formed the basis of the discussions is available at

http://www.3gpp.org/ftp/tsg_t/WG2_Capability/TSGT2_17_Vancouver/Docs/T2-020391.zip

1 Scope

The present document is a technical report consisting of a benefits analysis and a feasibility study on the creation of a framework enabling the application of MExE to arbitrary runtime environments.

2 References

The following documents contain provisions, which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TS 23.057: "Mobile Execution Environment (MExE); Stage 2", Version 5.0.0.

[2] International J Consortium, JEFF specification draft of March 7 2002, available at: http://www.j-consortium.org/jeffwg/JeffDraftSpecs2002March7.pdf

[3] 3GPP TS 22.057: "Mobile Execution Environment (MExE); Stage 1", Version 5.4.0.

3 Definitions, symbols and abbreviations

3.1 Definitions

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

RTIF mapping: A table or description of implementation details that describe how a specific runtime environment meets the requirements of the Runtime Independent Framework. Applying the Runtime Independent Framework to a specific runtime technology includes the generic RTIF framework as well as any runtime-dependent details that must be defined in order to make the runtime conformant to the RTIF.

Runtime Environment: The environment for a specific runtime technology, including APIs and access to system resources, within which an application executes.

Runtime Profile: A runtime may support one or more variations of capabilities and services using the same core runtime technology. The details of what exactly is included in a specific combination is termed a Runtime Profile. Runtime Profiles usually have names.

Runtime Technology: The technology that is provided to enable an application to execute. This includes the instruction set or script language syntax, the definition of the virtual machine or instruction processor, and the APIs available to the application programmer.

3.2 Symbols

For the purposes of the present document, the following symbols apply.

None.

3.3 Abbreviations

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

RTIF: Runtime Independent Framework

OEM: Original Equipment Manufacturer

ODM: Original Design Manufacturer

CAB: File Format for .Net

JTAPI: Java Telephony Application Programming Interface

4 Current Situation

Currently, in order to be MExE compliant, a device must implement at least one of the four run-time technologies specified by classmarks 1 through 4. A device cannot claim MExE conformance by applying portions of the MExE specification to other runtime technologies.

This leads to several problems, which can be roughly described as a delay incorporating new technology into 3GPP, an unbounded specification growth, uncertainties in implementation requirements, and a fragmentation of the application market.

4.1 Delaying new technology adoption into 3GPP

The 3GPP specifications are updated at approximately yearly intervals. Incorporation of a new runtime environment is seen as a new feature, and must correspond to a work item in a new release. Given the currently rapid advancement of runtime technology, and the large numbers of specifications and profiles now being worked on, the intersection of the two time cycles, i.e., specification approval time and development time, may result in a compelling technology being adopted more slowly into 3GPP MExE than the market demands.

One example of this is related to the MIDP 2.0 specification, currently being finalized in the Java Community Process, and likely to be ready in late 2002. MExE classmark 3 includes MIDP 1.0, and does not make any provision for future versions. There is very strong demand and support for MIDP 2.0 among both manufacturers and carriers. It is likely that the T2 SWG1 (MExE) will have to revisit MIDP 2.0 and define the means by which it will be supported in 3GPP after device manufacturers have already adopted it and released phone products that include MIDP 2.0.

4.2 Unbounded specification growth

The current specification does not provide means for implementing a mobile execution environment that is not specified as a MExE classmark. This restricts companies from making a runtime environment to work within the MExE (and by implication, 3GPP) framework. Companies are starting to recognize that their runtimes must be included into the MExE specification as unique variations of the classmarks in order for them to build a MExE device. Each MExE classmark defined for the MExE framework currently requires an additional section to the specification, making the specification longer, more imposing, and harder to read. The incorporation of classmarks into the MExE specification usually includes the listing of specific runtime features. Since the included technologies are defined by their own specifications, the listing of specific runtime features in the MExE specification is non-normative, at least, and, as discussed below, could possibly lead to conflicting interpretations with the referenced specifications.

4.3 Inefficient use of 3GPP technical resources

As has been demonstrated over the last few years, many companies are interested in having their technology be 3GPP and MExE conformant. The only way to do this is to propose the creation of a classmark specific to their runtime technology. This requires each company to present a proposal for classmark consideration in front of T2 SWG1, followed by presenting a defence in front of T2, the parent organization. After the proposal is accepted, the company needs to work for several months in making change requests to the MExE specification for approval by T2 SWG1 and T2. Understandably, a company that proposes a new runtime technology to 3GPP has a lot invested in it, and, without

an alternative means of applying their technology in the 3GPP environment, are very reluctant to accept a negative response from the T2 SWG1 or T2. This results in a lot of time from T2 and T2 SWG1 being spent in reviewing specific technologies, and a lot of effort in trying to determine whether or not the proposed technology can be used within the MExE framework and what additional value that it may provide.

This places T2 SWG1 in the role of a technology evaluator, a role that consumers and the market should serve. Unfortunately, this is not the best way to enable the growth of the capability and features in the mobile data marketplace. At best, this places T2 SWG1 and 3GPP behind the technology curve, instead of providing an environment where 3GPP can lead the adoption of new, compelling technologies.

4.4 Uncertain implementation requirements

Nothing in the MExE specification explicitly states that all classmarks must be implemented on a device for it to be MExE conformant. In fact, the specification currently states that the implementation of "one or more" classmarks is required for a device to be MExE conformant.

Currently, device manufacturers are reluctant to see new classmarks added to the specification. This may be due to an interpretation that "being a complete MExE implementation" seems to imply that all classmarks have to be implemented on a single device. It may also be due to the difficulty for manufacturers to determine, in advance, which classmarks will be important in the market, and which will not. The companies have a fear of choosing incorrectly, so they proceed with the safest choice in implementing all classmarks.

Understandably, this leads to the fear of an ever-increasing implementation burden and associated increase in demands on base platform storage, memory, power, and size.

Furthermore, it appears that including runtime environments in the MExE specification, even by reference, has implications to manufacturers and carriers that 3GPP is "recommending" that runtime technology. Currently, OEMs and ODMs appear to be interpreting the adding of MExE classmarks to be a recommendation from 3GPP which, ultimately, implies that they have to implement all classmarks on a device to adequately support the MExE service environment.

4.5 Potential fragmentation of the application market

Designating specific technologies as classmarks opens up the possibility of imposing runtime specific requirements on those technologies, and those requirements may be different in MExE from those in the runtime specification itself. The potential for this exists with classmark 2 and the designation of mandatory and optional packages listed in TS 23.057 [1] – Table 4. The required and optional packages are stated in TS 23.057 [1] – Section 6.1.2.3 to be the same as those in the Wireless Profile JavaPhone API specification, but there is no guarantee that these two specifications may not diverge at some time in the future. To maintain runtime consistency, the runtime technology needs to be defined in one, and only one, place.

This fragments the technology from the point of view of the application developer. It also puts the MExE group in the position of redefining issues that the authors of the runtime specification should be controlling. It is in the best interest of consumers, application authors, manufacturers, and carriers if a given named runtime "means the same thing" to the programmer, whether it is implemented on a MExE compliant device, or some other device conformant to the specification for that runtime.

4.6 Unclear technology requirements for classmarks

At the T2#17 Plenary, a response was drafted to a Liaison Statement from 3GPP SA1 requesting the criteria for inclusion as a classmark. The T2 SWG1 spent quite a lot of time on this question. Technical requirements on runtimes were also discussed in preparation for this TR. While T2 SWG1 and T2 were able to provide rough guidelines to SA1, these were non-binding and subject to change. These guidelines were subsequently edited down to a half page document at the following T#16 Plenary. See documents TP-020109 and TP-020170. It is clear that detailed, specific technical feature requirements for classmark adoption do not exist, and it is very likely that they would be difficult or impossible to create.

It is confusing, at best, for T2 SWG1 to apply different required functionality on one runtime versus another. Support for the JTAPI core package is mandatory for conformance with classmark 2, but the other classmarks, except the WAP classmark 1, have no such support for telephony capabilities within the JTAPI core package. This unevenness of

required support across the classmarks makes it very difficult for the T2 SWG1 to determine what is necessary for a runtime technology to meet when operating as a MExE classmark.

4.7 Summary of current situation

The numerous difficulties with the current scheme supports the idea that T2 or T2 SWG1 should diminish its role as a body that approves runtime technologies for 3GPP and focus on its role of providing a flexible, secure, extensible, managed application environment that makes 3GPP networks available to current and future runtimes demanded by the marketplace.

A new approach supporting the coexistent and parallel evolution of independent standards for runtime environments provides a means for efficient standardisation. The currently serialised standardisation process can lead to unnecessary delays.

5 Reusable technology: An alternate approach

An alternative to classmarks currently used in MExE for integration of runtime technologies is to separate out the components and aspects that are independent of any runtime technology, and reusable, from several specific runtime technologies. Along with this, aspects of the service environment can be enhanced to support the use of any runtime within the new framework. This document refers to the creation of an explicit set of runtime independent technologies and a binding framework that can be applied to any runtime environment as the *Runtime Independent Framework*, or RTIF.

The creation of an RTIF provides an answer to the problems listed in the previous section. It allows the marketplace to determine what runtime environments and profiles to deploy, and when, while doing this within the security and confidence that the new framework provides. It does this by making a clear separation between the runtime dependent aspects (the runtime environments specification) from the reusable, runtime-independent parts (components and aspects), filling in some small missing pieces of "glue" technology, and explicitly stating conformance requirements for integrating runtime environments with the resulting runtime independent framework.

MExE provides technology in the following areas that are reusable between several runtime environments and not bound or limited to any specific runtime:

- Security infrastructure
- Service environment
- Core software update
- Provisioning a runtime environment
- Multiple execution environment support

The following sections discuss each of these technologies, what reusable value they provide, and any technical issues that need to be addressed in order to use these technologies in a runtime independent framework.

5.1 Security infrastructure

Since MExE defines a service and security infrastructure that is common across all currently defined execution environments (classmarks), it is not surprising that the security infrastructure is independent of runtime technology. The security infrastructure is one of the primary values of the MExE specification.

A security infrastructure is composed of both a *mechanism* and a *policy*. While several runtime technologies, such as Personal Java, ECMA CLI, and the J2ME MIDP 2.0, define security *mechanisms*, none define a complete security *policy* like MExE does. A security policy requires agreement of the involved parties, and can be realized using any one of several security mechanisms. MExE defines a security infrastructure by providing behavioural requirements, a policy, and some common protocols to ensure interoperability. MExE relies on the security mechanisms of the runtime technology, or the implemented functions of the terminal, to actually support the security infrastructure.

The fact that the MExE standard has been ratified by the membership of 3GPP demonstrates that this security infrastructure is based on industry consensus. Any alternative wireless security infrastructure would have to develop a security policy and a set of security mechanisms, like the one done for MExE. Additionally, MExE provides a public specification and forum with which to grow and adapt the security infrastructure over time.

In general, regeneration of something that is already available is a waste of effort. A more efficient approach is to ensure that the MExE security infrastructure can be applied to a wide range of circumstances. Making the MExE security infrastructure available to a runtime technology, in general, and not just the runtime technologies included in the MExE classmarks, is one of the main goals of the RTIF.

The reusable security infrastructure is described by discussing each reusable aspect in the following sub-section.

5.1.1 Security model

5.1.1.1 Application isolation

MExE defines the means by which applications running within the MExE framework are allowed to interact. In general, these requirements inhibit unintended application interaction. They restrict the means available for applications to explicitly interact with each other to a level where corruption of another application, or its data, is unlikely.

MExE requires applications to have separate I/O streams that are not visible or modifiable to one another. The means by which this is accomplished is left as an implementation detail, but standard virtual machine and operating system memory management mechanisms are widely available. Requirements are detailed in TS 23.057 [1] – Section 8.2.3.

A definition of application isolation and a requirement to maintain that isolation is essential to any system hoping to maintain security with downloadable applications. This is true for devices having a single runtime technology, and is even more necessary for devices providing multiple runtime technologies. It would not be acceptable for a new runtime technology to be able to compromise the security of an older, widely deployed runtime. Therefore, the application isolation requirements of MExE are necessary as well as independent of any specific runtime technologies mentioned in TS 23.057 [1].

5.1.1.2 Domain definitions

In TS 23.057 [1] – Section 8.2, there are definitions of executable permissions for 3 trusted domains as well as an untrusted area. The trusted domain names are Operator, Manufacturer, and Third Party. There are specific, public key and certificate limitations for each of these domains.

Each secure domain, as well as the untrusted area, has a set of permissions that are allowed to it. These are listed in TS 23.057 [1] – Table 6. An application gains authorization to execute in a particular domain when being signed by the public key of a certificate whose certification chain verification is rooted by a specific, self-signed root key for a specific trusted domain. An application that is granted authorization to execute in a particular domain has access to the system services and resources available in that domain.

These permissions are described in terms of capabilities, called *actions*, in the specification, and not in terms of specific APIs. Therefore, this technology is independent of any runtime and is generically reusable with respect to runtime environments and RTIF. It is up to an implementer that is creating a MExE compliant implementation to determine how to enforce the capability restrictions appropriate for each domain.

The untrusted domain provides additional value, as it specifies what system resources and services an application that is not signed by a trusted party may use. The ability to run untrusted applications in a secure way is essential to enabling growth in the wireless application marketplace, since the large number of small developers will often not be easily able to get a trusted signature for their applications. Defining the capabilities of a secure environment to which they can write their applications without needing certification by another party encourages the development of new and novel applications, and encourages users to try out these applications.

Determining the domains and the untrusted area with their associated permissions was a major effort and represents the consensus of the industry in terms of what classes of entities can authorize various capabilities. Therefore, this feature provides major value to the industry and 3GPP has a strong incentive to make this as widely reusable as possible.

5.1.1.3 User permission types

In TS 23.057 [1] – Section 8.3, there are definitions for the types of permission that a user may give an application requesting the ability to access certain restricted system resources and services. This includes blanket, session, and single action permission. At a minimum, the user must have control via single action permissions, but the MExE specification provides options that allow the user to exercise very flexible control over application behaviour. This is a finer grained, user-centric control of application resources. These permission types are described independently of the resources, and are, therefore, a reusable permission granting framework applicable to any situation that would benefit from providing user permission control.

5.1.1.4 Control of application connections and network activity

Because connection to the network often involves user charges, and may have privacy issues, it is essential that the user have control of network connections, and be informed whenever an application is using the network. MExE defines that the user must have control over network connections, and that the user should be informed of that activity. These are defined in TS 23.057 [1] – Section 4.11 and TS 23.057 [1] – Section 4.13. The control and notification requirements are defined behaviourally, and do not have dependencies on any specific runtime or user interface, and are widely applicable.

5.1.2 Certificates and certificate management

Another component of MExE that is widely applicable to any, and all, runtime technologies in a secure environment is the handling and management of certificates, and the authentication and authorization mechanisms that use certificates. This forms the basis of a consistent, universal authentication and authorization mechanism for all applications, and all runtimes, operating in the MExE environment. The MExE certificate and authorization architecture is defined in TS 23.057 [1] – Section 8.4.

Two open issues with using the certificates are the means by which they are distinguished for a particular secure domain, and the means by which they are associated with a specific executable. The RTIF requires a mechanism to provide both of these capabilities. This will be discussed in Section 6.

5.1.2.1 Certificate format requirements

MExE specifies that X.509 Certificates (Version 3) must be supported. Furthermore, support for the "SHA1WithRSA" signature algorithm is required. A maximum supported key length requirement of 2048 bits can also be inferred from the referenced specifications. Certificate details are specified in TS 23.057 [1] – Sections 8.4.1.1 and 8.6.1.1.

This certificate format provides what is necessary, and is completely independent from a runtime technology.

5.1.2.2 Domain-based certificate requirements

MExE specifies that an individual certificate, and its associated public key, can only be used to certify an application for one of the trusted domains. This keeps the certificate hierarchy, and associated processing, straightforward, since only one certificate chain needs to be checked for any application. While simple, the system is flexible, in that a certifying entity needs to only have a certificate and public key for the domains that it can certify, and the maximum that any entity may need in order to certify applications to run in any domain or the untrusted area is three. This is detailed in TS 23.057 [1] – Section 8.5.

All of this is independent of any runtime technology and applies equally well to each of them.

5.1.2.3 Certificate chain structure and authorization

The MExE specification defines one certificate hierarchy to be used and shared by all runtime environments installed on a particular device. At any moment, a device may have at most one active root operator key, one active root manufacturer key, and any number of root trusted third party keys. This is termed the *trust hierarchy* in the MExE specification.

Any MExE application has at most one certification path through the certificate chain to a root key. The type of the root key at the top of the certification chain determines which secure domain, if any, the application is authorized to enter. An application that cannot be certified by following a chain to the root key is usually permitted to run as an untrusted application. This is detailed in TS 23.057 [1] – Section 8.4.4.

Furthermore, the domain of an application certified through a non-root certificate is solely determined by the type of the root key at the top of the certification chain for that certificate.

All certificates and keys can potentially apply to applications destined to execute in any runtime. MExE chose this approach because it is more efficient in terms of processing and storage than a scheme that has a separate trust hierarchy for each runtime. There are several other benefits of this for the RTIF. The size and complexity of the trust hierarchy can remain constant, even if there is an increase in the number of runtimes that the MExE specification supports. Additionally, if the system software on a device is upgraded to support additional runtimes, no change needs to be made to the trust hierarchy; it can be used, as is, to authorize applications for the added runtime.

In summary, the MExE certificate trust hierarchy and authorization mechanism is flexible and reusable and applies equally well to current runtimes, and future runtimes that may be supported on MExE devices.

5.1.2.4 Certification Configuration Message (CCM)

MExE also defines a means of managing the enabling or disabling of trusted third party certificates via a certification configuration message (CCM). TS 23.057 [1] – Section 8.7 provides the format of the CCM and outlines the protocols for a device accepting a CCM. TS 23.057 [1] – Section 8.7.4 details how CCM messages are to be securely downloaded. This is well integrated with the concepts of the certificate trust hierarchy and the administrator role.

5.1.2.5 Handling of root public key stored on an installed security device

The MExE specification details how root public keys stored on an installed security device, such as a USIM, should be handled. The specifics of how, what, and when root public keys on the USIM shall take precedence over those on the UE are detailed in TS 23.057 [1] – Section 8.5.

Again, this is reusable technology, independent of the runtime, and this is necessary in an environment providing secure execution of downloadable applications under a wide range of device configurations.

5.1.3 Administrator role

The MExE specification provides a key abstraction, that of the device *administrator*, which is distinct from the role of the device *user*.

- The administrator is a specially designated entity that plays a key role in managing the security configuration of the device, including installing and updating third party public root keys, deleting public root keys, and accepting CCM messages.
- The user is the person actually using the device to make phone calls, review and make entries to the address book, etc.

The MExE specification details how the administrator is determined in TS 23.057 [1] – Section 8.8.1. Basically, a separate public key may be installed in the MExE device for determining the administrator. The lack of an installed administrator key makes the user operate as the administrator. If there is an administrator key installed on the device, any party designated by the key can become the administrator. Rules for determining the administrator when an administrator key is present on an installed security device, such as a USIM, are detailed in TS 23.057 [1] – Section 8.8.1.2.

The device administrator may be the device user, the device owner, the carrier, or any other designated party. A distinction between user and administrator provides more flexibility in managing the device. For example, a corporation can provide cell phones to its employees and restrict third party applications to those that the corporation has signed.

The MExE scheme provides quite a lot of management flexibility with little additional implementation complexity. Any system providing secure downloadable applications for mobile devices will need a means of determining who controls the security of the device. MExE provides a solution that can be applied to a wide range of devices, runtimes, and usage models.

5.2 Service environment

Several aspects of the MExE service environment detailed in the MExE specification are reusable across runtime technologies with little or no modification.

5.2.1 Capability negotiation

MExE specifies the use of WAP UAProf and CC/PP attributes for capability negotiation. In MExE, this technology is used to communicate the classmark support from the terminal to the MExE Service Environment (MSE). One way that this could be used is to limit the downloadable content visible to the user on the browsing device to MExE executables that the device can execute. TS 23.057 specifies the current set of UAProf properties identifying the supported MExE classmarks, the supported version of the MExE specification, and the supported security domains.

While the basic technology is present in the current MExE specification, the specific attributes needed to support a flexible RTIF are not currently available. While several runtime independent MExE properties (MexeSpec, MexeSecureDomains, Vendor, Model, ScreenSize, etc.) are supported, the properties that designate runtime support are closed ended and not flexible enough to support the RTIF. Currently, the designated properties are identified as MexeClassmarks, JavaPlatform, and, possibly, CLIPlatform.

A small proposed set of additional attributes and value formats necessary to support the RTIF with an unbounded set of runtimes will be presented in a following section.

5.2.2 Provisioning

MExE relies on a browser offering HTTP or WAP transfer protocols to download and provision applications. This model has worked well on the wired Internet, and is expected to succeed equally well on mobile devices. One issue that arises on the wireless Internet that has been addressed on the wired Internet is determination of content type.

Content, downloaded from the Internet, depends upon use of MIME types in the header to provide the first step in determining the actual type of the content, and how it should be handled. In some cases, knowledge of the MIME type is sufficient to determine how the content of downloadable MExE applications should be handled. In other cases, the MIME type is just the first step in the logic that determines how the content should be handled on the device. The content, itself, must contain enough information to make this determination. This is all implied by the MExE requirements for browser support in TS 23.057 [1] – Section 4.10, and applies equally well to all runtime technologies.

The latter is likely to be more common. This is demonstrated in the cases for Java, where there are multiple profiles and configurations, all of which will be contained in downloadable files of the JAR content type. This leads to an additional requirement on the RTIF mapping for a runtime technology profile to describe how to determine whether content is appropriate for that runtime mapping.

5.2.3 Management requirements

The MExE management requirements, specified in TS 23.057 [1] – Section 4.9, detail high-level aspects of service discovery, transfer, installation and configuration, census, and termination. These aspects are independent of the runtime technology and apply equally well to all runtime technologies.

5.3 Core software update

MExE provides security for downloaded *core software*. Obviously, the ability to upgrade the core software on the terminal device in a secure manner under the manufacturer's control applies equally well to all runtime technologies for MExE classmarks and RTIF. The details for secure downloading are presented in TS 23.057 [1] – Section 4.14, and the elements provided for the manufacturer domain can be easily reused for downloading core software.

5.4 Provisioning a runtime environment

The RTIF provides a means for manufacturers and operators to upgrade terminal devices in the field with new runtime technologies as they grow in demand in the marketplace. The MExE specification needs no changes in order to provide this capability.

5.5 Multiple execution environment support

MExE defines the way that applications and execution environments are to behave in the presence of other execution environments. This idea can be easily extended to include, both, runtimes implemented as MExE classmarks and

runtimes using the RTIF. Essentially, MExE requires that the applications and runtimes behave functionally consistent, with a possible difference of timing performance, whether one or many runtime environments are installed in a device.

It is clear that this condition is necessary in order to enable growth in mobile applications and expansion of capabilities and features of the runtimes for which they are written. To be useful, applications must run predictably, regardless of whether other software, beyond that required to provide the runtime environment, is installed in the device.

6 Integrating the Runtime Independent Framework into the Current MExE Specification

This section will detail the additions and changes to TS 22.057 [3] and TS 23.057 [1] that are necessary to introduce the RTIF into the current TS 23.057 (MExE) specification.

6.1 RTIF conformance requirements

At a very high level, what is necessary to introduce the Runtime Independent Framework to TS 22.057 [3] and TS 23.057 [1] is a set of requirements to be conformant with the framework. For a runtime, or a device, for that matter, to be conformant to MExE, it must have a specific set of conformance requirements in TS 22.057 [3] and TS 23.057 [1]. By definition, the RTIF does not require creation of new classmarks. However, a runtime will need some criteria of conformanceother than classmark conformance.

Therefore, to support the RTIF, a section in the TS 22.057 [3] or TS 23.057 [1] will have to be added that details what the requirements are for conformance. In general, these requirements fall into two categories: runtime generic and runtime mapping requirements.

6.1.1 Runtime generic requirements

These are requirements on the behaviour of the runtime and system software as implemented on a MExE device in a RTIF conformant manner.

The RTIF will define conformance to runtime generic requirements in terms of compliance with the reusable components of MExE listed in Section 5 of this report. The specific, corresponding sections of the TS 23.057 (MExE) specification should be explicitly listed in the RTIF compliance section. If additional features and requirements are added to the MExE specification, it will have to be determined whether these need additional reference in the sections with RTIF conformance requirements. Alternatively, the RTIF sections could require compliance of the entire specification while explicitly stating exceptions for specific implementations of technology for a classmark's environment.

6.1.2 Runtime mapping requirements

These are requirements that the runtime mapping must specify in order to "fill in the details" and make an RTIF mapping reproducible and not conflict with other RTIF mappings. These are requirements that a runtime mapping must specify before it can claim conformance with the RTIF. These will usually take the form of a published document detailing how the profile for the runtime technology has been made to conform to the MExE specification. The following requirements apply to the definition of how that runtime conforms to the framework, as well as to the "filled in details" for the implementation of the RTIF mapped runtime.

- Provide a complete definition of the runtime environment including a specification of the runtime technology, i.e., mandatory and optional APIs. This must be published and available to those who would use the runtime to create applications.
- Provide a description of how the MExE requirements, in particular, the security requirements, have been fulfilled. This must be published and available to those who need to review how the MExE requirements are met in order to make decisions on implementing that RTIF mapping into a MExE device.
- Provide a description of the algorithmic means of determining whether content of a given MIME type is executable by the RTIF mapped runtime. This is likely to be published along with the assignment, or registration, of a particular MIME type.

- Provide a unique identifier for the runtime mapping. This identifier will be used to identify device support and content associated with this RTIF mapping. In particular, this name will be used in UAProf attributes during capability negotiation, and may be used inside the metadata of a content package to differentiate from non-compliant content of the same MIME type.
  The suggested UAProf extensions use the URI mechanism to ensure that the namespace of identifiers is extensible, and identifiers do not collide. It is recommended, although, not required, that the RTIF mapping define how a client should handle different versions of the RTIF mapping that is expressed through similar, although not identical, identifiers. See Section 6.2 in this report, UAProf extensions.

- Provide a description of how the required X.509 Certificates are associated with an executable for that runtime. This may use a runtime-specific archive format, such as JAR files, or some other means.

Alternatively, the sections on RTIF mappings could be published as informative text. This implies that the sections on generic RTIF requirements formulate the complete set of normative materials. In this fashion, the runtime mappings show, that the MExE classmarks follow the requirements and guidelines established by the Runtime Independent Framework.

This pattern of RTIF requirements followed by informative mappings to runtimes of a classmark clearly shows that there is no longer a need for additional classmarks in the MExE specification. Any runtime environment that meets the requirements listed under the generic RTIF section, implicitly conforms to the MExE requirements, and descriptions that are specific to runtime technologies are strictly informative. Adoption of informative text requires less processing within the standards groups, and the new pattern for the MExE specification allows for many options of making annexes, chapters, or sections for easier inclusion of RTIF mappings.

6.2 UAProf extensions

The current set of UAProf attributes do not allow specification of an arbitrary runtime that has a compliant RTIF mapping and has been implemented on the client device. Clearly, some kind of flexible identifier is required. Since there will be no central control of the RTIF identifiers, the mechanism has to be both extensible and provide collision avoidance.

While there are many approaches to solve this problem, perhaps the simplest is to extend the UAProf attributes with a Literal Bag named "SupportedMexeRTIFs":

Attribute	Description	Type	Sample
SupportedMexeRTIFs	List of URIs designating supported RTIF mapped runtime profiles on this device.	Literal (Bag)	"http://www.sun.com/j2me/midp/2.0", "http://www.j-consortium.org/RTJWG/1.0"

Note that URIs are NOT intended to be web accessible resources, although, they may be. Instead, they are RTIF mapping identifiers that are under the sole control of the definer of the RTIF mapping, providing extensibility along with avoidance of collisions. If there does exist a web resource associated with the URI, typically, the URI is a document containing the specification of the RTIF mapping, itself.

To provide for future versions of an RTIF mapping, it is suggested that RTIF mappings use the following URI format for creating identifiers:

<Issuing party base URI> + "/" + <runtime technology name> + "/" <profile name> + <version number>

Example applying this to MIDP 2.0:

http://www.sun.com/j2me/midp/2.0

This scheme can even be applied to the current set of classmarks in order to bring all runtimes associated with MExE into the name identifier system. Some examples are provided in the following list:

http://www.3gpp.org/mexe/classmark1/5.0

http://www.3gpp.org/mexe/classmark2/5.0

http://www.3gpp.org/mexe/classmark3/5.0

http://www.3gpp.org/mexe/classmark4/5.0

6.3 Other MExE specification changes

6.3.1 RTIF conformance

TS 23.057 [1] – Section 4, "Generic MExE aspects", specifically requires support of at least one classmark for MExE devices to comply with the MExE specification. It does contain a forward-looking statement that makes it clear that the authors thought that a one-size-fits-all (and by implication, a fixed set of supported runtimes) was unrealistic.

This section will have to be revised to provide for conformance with the RTIF. One approach is to replace the classmark scheme with the RTIF scheme.

Another approach is to modify the MExE specification to specifically define two types of conformance: classmark conformance and RTIF conformance.

> *Classmark conformance* is defined to be identical to the conformance requirement for implementing one of the 4 current classmarks, with the addition that a classmark conformant device may optionally support the Runtime Independent Framework.
>
> *RTIF conformance* is defined as the compliance with the requirements set forth in Sections 5 and 6 of this document.

Alternatively, the MExE specification can be limited to requiring RTIF conformance with informative text demonstrating a softer aspect of classmark conformance. If the MExE specification builds a pattern with normative descriptions for generic RTIF elements, the classmark descriptions build an informative description of a specific runtime environment complying to the minimum, essential elements of MExE aspects.

The RTIF conformance is a complete set of the minimum, essential aspects of MExE requirements and there should be no further need in making requirements within an implementation of a specific runtime environment that meets the general functions, services, and characteristics of a MExE device.

6.3.2 Multiple execution environment and runtime support

TS 23.057 [1] – Section 4.4, "Multiple classmark support", must be expanded to include the possibility of support for the RTIF and include runtime technologies executing within the RTIF. It should also discuss support for one execution environment, or more, on the same device.

In general, the approach taken in the current specification states that applications executing on a device supporting multiple execution environments must behave the same and meet the same requirements as when executing on a device supporting only that execution environment. These same requirements apply to a device simultaneously supporting one or more classmarks and/or the RTIF that includes one or more runtime technologies.

7 Additional open issues

7.1 Binding executables to certificates and metadata

Currently, MExE does not define any runtime independent manner to associate, or *bind*, an executable with its associated certificates or metadata. Each classmark does this in its own way. Classmarks 2 and 3 use JAR files, while Classmark 4 uses a CAB file format. While this approach can be extended to RTIF runtimes, it is inefficient in terms of code size. A binding mechanism, common to all RTIF mapped runtimes, would decrease the implementation burden of supporting the RTIF as well as supporting multiple runtimes mapped to the RTIF in a single device. A common mechanism would also simplify the choices needing to be made when creating an RTIF mapping.

One simple approach would be to standardize on a single archive format for all runtimes complying with the RTIF. The binding between an executable, a certificate, and metadata is accomplished by placing them all in the same archive.

There are several archive formats available in the public domain that would be sufficient for this purpose, including the ZIP file format, and the JEFF [2] file format, now an ISO standard.

7.2 Root key certificate packaging and metadata

A related issue to binding executables to certificates is how to package certificates and bind them to metadata. This is specifically necessary for root key certificate packages intended to be installed on MExE devices. X.509 Certificates do not include an internal means of specifying which secure domain for which they are associated. Since the domain of a non-root certificate can be determined by tracing to the domain of the root, this is only an issue for root key certificates, and, especially, for certificates containing root keys for the trusted third party domain. Some metadata, external to the certificate, is required for designating the domain.

The only way to do this with the current specification is to use the JAR file format and manifest attributes associated with classmarks 2 and 3. This is discussed in TS 23.057 [1] – Sections 8.10 and 8.10.2. This solution is tied to Java technology, and, in practice, is related to devices that support classmarks 2 or 3.

One simple, runtime independent solution, is to place the certificates in a runtime independent archive using the subdirectory of the root of the archive to identify the domain. Each secure domain would have a specific directory path defined for its use. This technique reuses the archive format discussed in Section 7.1, above, and is already used for the storage format described in TS 23.057 [1] – Annex A.3.

7.3 Handling of existing MExE classmarks

No changes to the current classmarks are required to create the RTIF. However, it may be desirable to align the future versions of the current classmarks with the RTIF for technology such as archive formats and UAProf extensions. The changes to existing classmarks should be discussed separately from those necessary to support the creation of the RTIF.

As stated earlier, the creation of the RTIF imposes no additional requirements on future classmarks. Integration with the RTIF demonstrates a proof of the feasibility of a technology working within the MExE framework while meeting all the requirements of the MExE framework. In summary, the RTIF can support the current classmark structure for backwards compatibility issues, or it can be used to support a system without classmarks. Furthermore, it does not require any new classmark constructions to be created for successful implementation.

8 Out of scope issues

During examination of the RTIF, several issues were discussed and determined to be separate from the creation of an RTIF. While some of these issues may be important in setting the future direction of the MExE standard, it was decided that the RTIF should be created independently from discussion of these issues:

- The MExE specification could establish a minimum level of functionality in areas such as media support, telephony, XML processing, etc., for RTIF mapped runtimes. The MExE SWG decided that mandatory or optional features of a runtime technology are the decision of the runtime creator and the drafters of the RTIF mapping document.

- It was decided that it was not necessary for an RTIF mapping to specify which function calls were affected by the domain encapsulating an executable. It was discussed that T2 needs some means of evaluating how MExE security requirements are met, and it may be in the interest of the party proposing a new classmark to provide information at this level of detail, but it is not strictly required for either RTIF compliance or for proposing a new MExE classmark.

- A standardized secure transport format and protocol would be generally useful across all runtime technologies, especially RTIF mapped runtimes. However, creation of this is a separate task.

- The issue of architectural constraints on runtimes, such as are binary runtime environments, providing acceptable security guarantees was discussed, but determined to be more an issue for classmark adoption rather than conformance with the RTIF. No runtime architectural constraints for the RTIF have been proposed. However, questions were raised on the complexity of the system software required to support binary runtime environments.

- It was maintained that support for all the secure domains, as well as the untrusted area, is critical to the success of downloadable applications, MExE, and the RTIF. No allowance was made for RTIF mappings that only support the trusted domains, or RTIF mappings that support a subset of the trusted domains.
- Definition of which media or content types must be supported by RTIF runtimes was determined to be out of scope.
- The manner in which the user profile information is to be integrated with the RTIF was felt to be the same as the issue of integration with the current classmarks. This work will be separately considered as the generic user profile work proceeds.

9 Conclusion

In order to encourage the growth and popularity of downloadable applications on mobile devices, application authors need powerful runtime environments to program, users and carriers need security and provisioning support that they can rely on, and mobile device manufacturers need a means of incorporating new technology as it becomes compelling and the market demands.

The current MExE Stage 1 (TS 22.057 [3]) and Stage 2 (TS 23.057 [1]) documents provide important, reusable technology that goes a long way to address these issues. Much MExE technology applies equally well to current and future mobile runtime environments. Additionally, MExE provides components based on industry consensus, such as the security domain policy model, that are not available anywhere else. However, the current MExE specification limits the application of this technology to runtime environments adopted as classmarks.

This technical report shows that the creation of a Runtime Independent Framework (RTIF) for execution environments is technically feasible . It outlines the aspects of the MExE framework that are reusable, and describes a small number of technical additions that are necessary to provide a working RTIF. The resulting proposed Runtime Independent Framework provides for a means of conforming to the MExE framework and the reusable MExE technology components independent of the details of the runtime technology.

This is a report of the feasibility study and not a conclusion of the analyses.

Annex A: Generic MExE Security

The following section gives an example baseline for of how an implementation following the RTIF security guidelines could be achieved (based on the 3GPP TS 23.057 version 6.1.0).

A.1 Introduction

In order to manage the MExE and prevent attack from unfriendly sources or transferred applications unintentionally damaging the MExE device a security system is required. This clause defines the MExE security architecture.

The basis of MExE security is:

- a framework of permissions which defines the permissions transferred MExE executables have within the MExE device;
- the secure storage of these permissions (and permission type as defined in clause A.5 "User permission types");
- conditions within the execution environment that ensure that MExE executables can only perform actions for which they have permission.

The MExE permissions framework is defined in 3GPP TS 22.057 [32] and is as follows (there is no implied hierarchy):

- MExE Security Operator Domain (MExE executables authorised by the HPLMN operator, as described in clause A.3.1 "MExE executable permissions for operator, manufacturer and third party security domains");
- MExE Security Manufacturer Domain (MExE executables authorised by the ME manufacturer, as described in clause A.3.1 "MExE executable permissions for operator, manufacturer and third party security domains");
- MExE Security Third Party Domain (trusted MExE executables authorised by trusted third parties, as described in clause A.3.1 "MExE executable permissions for operator, manufacturer and third party security domains");
- MExE Untrusted Area. Untrusted MExE executables are not permitted to execute in a security domain (i.e. Operator domain, Manufacturer domain or Third Party domain) and execute in the Untrusted area, and have very reduced privileges as described in clause A.3.2. "MExE executable permissions for untrusted MExE executables".

A MExE device shall support either all three security domains or no domains. If the security domains are not supported, then all applications shall be untrusted. The MExE device shall not support any subset of the three security domains. Support of the MExE Untrusted area is mandatory.

A.2 MExE executable integrity

If the 3 MExE security domains defined in clause A.1 "Generic security" are not supported, then the MExE device shall ensure application integrity immediately prior to application execution. the pre-verification of MExE executables at launch time described in this clause is optional.

A potential threat is that MExE executables may be securely authenticated at the time of download, but tampered with or corrupted prior to being launched. Further a certificate may be compromised or expired. Authentication of a MExE executable at the time of download does not ensure that the MExE executable has not been modified when it is subsequently launched. Furthermore, authentication of a MExE executable at the time of launch does not ensure that the MExE executable is not modified during execution. Similarly, verification of the certificate at the time of download may not ensure that the certificate is valid at time of application launch, and verification of the certificate at the time of launch does not ensure that the certificate remains valid during execution.

Therefore, the MExE device shall ensure application integrity immediately prior to application execution.

Application integrity is defined as the state in which:-

- application code has not been modified since authentication; and

- the certificate containing the root public key is checked and known to be valid.

The mechanism by which the device preserves integrity is an implementation detail, dependant on the application storage mechanism and access. Examples of mechanisms that contribute to such application integrity could include :

- Storage of applications in a memory area that cannot be compromised on the device;
- Preventing launch of the application when the MExE device becomes aware that the certificate is invalidated;
- Full signature verification prior to each application invocation (see clause A.2.1 “Full signature verification”);
- Optimised pre-launch signature verification (see clause A.2.2 “Optimised pre-launch signature verification”);
- Periodic full signature verification by separate process during application execution.

The list of examples is not exhaustive and any other mechanisms ensuring application integrity may be equally considered.

A MExE device may furthermore ensure that the application code has not been modified during application execution.

A.2.1 Full signature verification

Full signature verification assumes that the procedure of validation for downloaded MExE executables and certificates is used. For more details see clause A.7 “Certification and Authorization Architecture”.

A.2.2 Optimised pre-launch signature verification

This is an optional feature which is used to eliminate the potentially excessive overhead of checking a signature again after initial full certificate verification has already been performed.

To use this process the MExE device shall create a hash of the executable object (executable object fingerprint) as if checking the signature. This shall be stored in a protected verified application list, along with indication of the domain permissions for the application. The hash used shall be the same type as that used for signing the object. When launching an application or downloading an applet, the hash shall be performed as for when computing the signature. The verified application list shall then be checked; if the hash value is present and the entry has not expired then the application or applet may execute. If no list entry exists for this object, or the entry has expired, the process shall then proceed with the full signature verification. Note that the lists for applications and applets should be separate and that an implementation determines management policy for the lists (e.g., ageing policy, which entries to delete when trying to add a new entry to a full list etc.). One restriction imposed that shall be enforced is that the maximum number of uses for an entry before it is marked invalid is limited to some maximum value.

In the event that a new CCM is new certificate information has been received by the MExE device, all verified application list entries shall be marked invalid unless some mechanism to determine the validity of an authorising certificate entry for each application is provided by the MExE device implementation.

A.3 MExE executable permissions

Support of MExE executable permissions as detailed in this clause is mandatory.

A.3.1 MExE executable permissions for operator, manufacturer and third party security domains

The following table A.1 "Security domains and actions" specifies the permissions of operator, manufacturer and third party security domains in the order of restriction.

The actions listed in the security table A.1 "Security domains and actions" are generic actions. These actions can only be performed by MExE executables via application programming interfaces (APIs) (which are intrinsically part of the MExE implementation) The security restrictions shall apply to MExE executables whether the API functionality is called directly or indirectly by the MExE executable. Explicit user permission is required for all actions by MExE executables in all domains. Types of user permission are defined in clause A.5 "User permission types".

Untrusted MExE executables are not permitted access to any actions which access the phone functionality (phone functionality includes all the actions in table A.1 "Security domains and actions") except for the exceptions identified in clause A.3.2 "MExE executable permissions for untrusted MExE executables".

Actions available using interfaces giving access to the phone functionality (either in existence at the time of approval of this specification or not) that are not listed in the security table A.1"Security domains and actions" shall be categorised into one of the groups in the security table A.1 "Security domains and actions" by comparing its action against the groups in order as they are listed in the table A.1 "Security domains and actions". If an action can be categorised into a more restrictive group near the top of the table, then it shall not be again categorised into another, less restrictive, group further down in the table. For example, if a new action eventually results in forwarding a call, it shall be categorised into Network access. If the action is totally new, it shall be categorised into some of the groups by comparing its functionality to the group description below and by comparing with the list of actions listed in the table within the group.

1. Device core function access includes functions, which are an essential part of the phone functionality .
2. Support of core software download, which allows updating the ME radio, characteristics and properties by changing the core software in the ME (e.g. a new CODEC may be loaded into a ME, a new air interface, etc.)
3. (U)SIM smart card low level access includes functions, which allow communications at the transport service access point (send and receive application protocol data unit).
4. Network security access includes all functionalities which relate to CHV, CHV2, UNBLOCK CHV and UNBLOCK CHV2 (verification, management, reading or modifying), GSM authentication, GSM ciphering.
5. Network property access includes functions, which enable the management of operator-related data parameters and network settings.
6. Network services access includes all functionalities which result in or need interaction via the operator's network.
7. User private data access includes all functionalities which relate to management, reading or modifying of data that the user has stored in the MExE device including user preferences.
8. MExE security functions access includes all functionalities which, through an API relate to certificate handling in the MExE device; end to end encryption, signed content, hashing, access to public, private, secret keys stored in the MExE device or in a smart card.
9. Application access includes the functionalities which relate to launch provisioned functionality, MExE executables, external executables ((U)SIM tool kit application,…) usage.
10. Lifecycle management includes the functionalities which are needed for installing or removing MExE executables in the MExE device.
11. Terminal data access includes the functions which relate to accessing terminal data, i.e. not user data.
12. Peripheral access includes the functionalities related to peripherals other than user interface peripherals usage through a high level software application interface.
13. Input output user interface access includes the functionalities related to the user interface and user notification means usage.

Table A.1: Security domains and actions

	MExE Security Domains		
Actions	**Operator**	**Manufacturer**	**Third Party**
Device core function access Start/stop radio Turn on/off device Write time and/or date Activate a user profile Modify a user profile	No		
Support of Core Software Download e.g. Update ME software	No	Yes	No
(U)SIM smart card low level access [11] Send APDU Slot management (power on/off, reset, port lock…)	No		
[11] – Access to (U)SIM is provided using more high level API as phonebook, application launching			
Network Security access Run algorithm Verify CHV/2 or UNBLOCK CHV/2 Activate/deactivate CHV Modify CHV/2	No		
Network property access Get IMSI Get home network Select network	Yes	No	
Network services access Initiate a voice/data connection [3] Accept a voice/data connection [3] Call forward [4] Multiparty call [4] Call deflection [4] Explicit call transfer [4] Terminate an existing connection Hold an existing connection Resume an existing connection Send point-point message (e.g. SMS, USSD) [4] Query network status Get signal level Get call list QoS management	Yes		Yes [6]
[3] – A network connection may be via any supported bearer service [4] – Multiparty, deflection, and explicit call transfer shall be permitted only to numbers explicitly supplied by the user to the MExE Executable. Modification of call forward numbers stored in the network shall only be permitted to numbers explicitly supplied by the user to the operator. [6] – The Third Party domain's permission to access the networking action depends on the provisioning mechanism as described in clause A.13.3 "Determining the administrator of the MExE device"			
User private data access [1] Read Write Get properties Delete Get Location Information Read stored SMS Delete stored SMS Modify user preferences	 Yes [2] Yes [2] Yes [2] Yes [2] Yes [2] Yes [2] Yes [2] Yes [7]		
[1] – User private data includes user files, phonebook, MSISDN, etc located on the MExE device. [2] – The user shall be able to specify data access permissions within the capabilities of the MExE device. It is not applied to user preferences [7] – Trusted applications only have permission to modify user preferences, and not to activate or de-activate them. The user shall be able to specify for each domain, the preferences that applications in that domain can access. All other preferences shall not be accessible to that domain. The default shall be that there is no access. Single action user permission is the only type of user permission that shall be possible for changes to User Preferences.			

Actions	MExE Security Domains		
	Operator	Manufacturer	Third Party
MExE security functions access			
Install a certificate for a given domain	Yes [5]		
Uninstall a certificate for a given domain			
Replace a certificate for a given domain	Yes [5]		
Data encryption API			
Verify a signature API	Yes [5]		
Compute a digital signature API			
Hash a content API	Yes		
Non repudiation API	Yes		
	Yes		
	Yes		
	Yes		
[5] – Only the organisation whose public key is certified (or the organisation that certified the public key) can add, delete or replace a particular certificate.			
Application access			
Get application list	Yes[8]		
Launch an application	Yes[8]		
Get application status	Yes[8]		
Stop, suspend, resume an application	Yes[9]		
[8] – ME provisioned functionality access is limited to manufacturer domain. (U)SIM tool kit application access is limited to operator domain. MExE executable access is limited to MExE executable issued by the same issuer (identify by the certificate) of launched MExE executable [9] – Access is limited to MExE executable which launch the application. But the end user, shall have a way to stop the launched application, MExE environment may stop the launched application or launched application may stop itself.			
Lifecycle management	Yes		
Install a MExE Executable			
Uninstall a MExE executable			
Terminal data access			
Get manufacturer software version	Yes		
Read time and date	Yes		
Peripheral access	Yes		
Sound generation to speaker (e.g. via stream)			
Set speaker volume			
printer access			
Monitor the power state			
Change the power state			
Activate/ access Serial port (RS232, IrDA, Bluetooth, USB …) access			
Activate/access Parallel port			
Activate/access Smart card other than (U)SIM card (Send APDU, Slot management)			
Input output User interface access			
Input device (keyboard, mouse …)	Yes[10]		
Output device (display)	Yes[10]		
Output notification device(smart icon, sound, light, vibrator …)	Yes		
[10] – Access request requires no user permission.			

The lists in the groups in table 6 "Security domains and actions" are not exhaustive, and other actions which are of the same category shall be included in the group for the purposes of requesting user permission.

This clause identifies the permissions for MExE executables in the 3 security domains (operator, manufacturer and Third Party). The permissions do not apply to untrusted MExE executables which are not permitted to execute within the domains.

A.3.2 MExE executable permissions for untrusted MExE executables

When the Security Domains are not supported then all executables are untrusted and they execute in the untrusted area for which the executable permissions are defined as follow in table A.2 "Executable permissions for untrusted MExE executables".

In order to facilitate untrusted MExE executables having some limited access to MExE device functionality beyond their very limited privileges, some of the access permissions in the previous table 6 "Security domains and actions" are extended to untrusted MExE executables and described in table A.2 "Executable permissions for untrusted MExE executables" as well as in clause A.11 "Separation of I/O streams".

The untrusted MExE executables permitted to use these facilities shall be MExE executables the user has downloaded him or herself, and not be MExE executables that have been pushed to the user. MExE executables on the MExE device due to the user having visited a particular Web site are considered to be MExE executables that the user had downloaded him or herself.

Untrusted MExE executables shall not be permitted access to any other functions.

Table A.2: Executable permissions for untrusted MExE executables

	Classmark 1	Classmark 2	Classmark 4	Classmark 3
User Interface	An untrusted, uninstalled MExE executable (e.g. an applet) can access the user interface output and input without user permission, but the sending of user data to a server to which the MExE executables has a session connection (e.g. as part of a browser session) requires user permission. An installed untrusted MExE executable shall only be able to access the user interface output and input with user permission (clearly, for the usability of untrusted MExE executables such as games, blanket user permission should be sought and given, and this is permissible).			Untrusted MExE executables can access the user interface output and input without the user permission.
File, Persistent Data	File access is not permitted for untrusted MExE executables.			
	But, untrusted MExE executables can access files only in the MExE executable's own directory.			But, persistent data may be stored via the MIDP record management system (stores are shared between MIDlets in the same MIDlet Suite).
Transmission over the Access Network	Untrusted MExE executables shall be able to exchange data, voice, HTTP requests, etc. over the Access Network under the following conditions: The recipient of a transmission (e.g. a phone number, a URL, a server name, etc.) shall be presented to the user for permission by a provisioned functionality of the MExE device itself, even if this recipient was already presented by the executable (this facility would support, for example, "click to dial" buttons/links in untrusted MExE executables). It shall not be possible for an application to use a transmission channel that it did not initiate (except for MIDlets within the same MIDlet suite). Note however that some execution environments define *application* as a collection of smaller units of executables provisioned in a single package or suite.			
Generate DTMF	Untrusted MExE executables shall be able to generate DTMF tones under the following conditions: An untrusted MExE executable is only permitted to send DTMF tones in a currently active call. The request to generate DTMF tones in the currently active call, shall result in the characters which the tones represent being presented to the user for permission by a provisioned functionality of the MExE device.			
Add Phonebook Entry	Untrusted MExE executables shall be able to add a phonebook entry (i.e. name and number only) under the following conditions: The name and the number to be added shall be displayed to the user for permission by a provisioned functionality of the MExE device and not by the MExE executable itself. The phonebook entry shall not be added without user permission. The function shall not be able to modify or delete any phonebook entry.			
Executable Interaction	Executable interaction is not permitted for untrusted MExE executables (except for MIDlets between units of executables within the same MIDlet application or suite).			

NOTE: The functionality of "Generate DTMF tones" and "Add Phonebook Entry" is not supported by the MIDP at the moment.

A.4 Handling of MExE executables when their valid root public key is not available

This clause considers the effect on MExE executables when the root public key of a secure domain (e.g. operator, manufacturer, third party) is no longer available (e.g. when the UICC is being physically removed, or the root public key is no longer valid).

A.4.1 Launching of MExE executables when their valid RPK is not available

It shall not be possible to launch a MExE executable to run in a security domain unless the root public key of that security domain is available and valid.

A.4.2 Currently executing secure MExE executables when their valid RPK is no longer available

On detection that the valid root public key of a secure domain is no longer present, the MExE device shall permit MExE executables currently executing in the secure domain controlled by that root public key to continue executing. Furthermore, if the same RPK is available again, the executable is allowed to keep on executing. However, if a different RPK is validated, the currently running MExE executables (under the old RPK) in that secure domain shall be terminated.

A.5 User permission types

Support of user permission types is mandatory.

The term "user permission" is defined to mean that the user can give permission for a specific action in one of the ways defined in table A.3 "User Permissions". Support of single action permission is mandatory, but support of blanket permission and session permission is optional.

Any request for user permission as described in table A.3 "User Permissions" must display a user friendly name identifying the signer of the corresponding MExE executable, if available. The "subject" field of the certificate of the signer ("subject" here refers to the "subject" fields of WTLS and X.509 certificates and an equivalent field for any other format of certificate) shall be made available to the user upon request. If an application, for which user permission is being sought, is untrusted, the fact that the application is untrusted shall be at least visually indicated to the user, if the MExE device is capable of visual indication, whenever user permission is sought. Other means of indication are additionally permitted. If the MExE device is not capable of visual indication, or is not designed for use by a human user, other means of indication shall be used.

The MExE device shall allow user control of permissions relating to all action groups listed in the table 6 "Security domains and actions" that are required by the MExE executable and supported by the MExE device.

Multiple action group permissions may be controlled in a single user action on the MExE device regardless of the permission type as listed in table A.3 "User Permissions". In such case, these action group permissions shall be made explicit to the user.

Note that blanket permission cannot be used for uninstalled MExE executables e.g. applets, WMLS.

Table A.3: User Permissions

	User Permissions		
Permission Type	**Description**	**Invocation**	**Revocation**
blanket permission	The user gives blanket permission to the MExE executable for the specified action, and the MExE executable subsequently uses the user's original permission for the identified subsequent actions whenever the MExE executable is running.	Typically such permission would be given at MExE executable configuration or run time.	The blanket permission maybe revoked by the user at any time. The user permission no longer applies once the MExE executable has been removed.
session permission	The user gives permission to the MExE executable for the specified action during a specific run time session of an MExE executable, and the MExE executable subsequently uses the user's permission for the identified subsequent actions whilst the MExE executable session is still running.	Typically such permission would be given at MExE executable run time.	The session permission maybe revoked by the user at any time. The user permission no longer applies once the MExE executable run time session has terminated.
single action permission	The user gives a single permission to the MExE executable for the specified action; if the MExE executable subsequently wishes to repeat the action it must again request the user's permission for the identified subsequent action.	Typically such permission would be given at MExE executable run time.	The user permission no longer applies once the action has terminated.

A.6 Root Public keys

If the 3 MExE security domains defined in clause A.1 "Generic security" are not supported, then the root public key management described in this clause is optional.

The definition implementation of the secure mechanism in this clause to mark as valid a root public key certificate on the ME, is out of the scope of this specification.

A.6.1 Operator root public key

The ME may support secure storage for one or more certificates, each of which contains an operator root public key. The ME shall support the use and management of a certificate containing an operator root public key stored on the MExE-(U)SIM and in the ME. The ME shall behave according to clause A.6.1.2 "ME actions on SIM insertion and/or power up". For support of public key management on the SIM and the USIM refer to 3GPP TS 51.011 [27] and 3GPP TS 31.102 [39] respectively. The certificate contains a root public key generated either by the operator, or by a CA trusted by the operator.

If the MExE device does not contain a valid operator root public key, then the certificate chain to MExE executable previously executing in the Operator Domain will be invalid, and the MExE executables will be excluded from the operator domain.

The user shall not be able to add or delete any type of operator public key (root or contained in a certificate).

Optionally, the operator may install a corresponding disaster-recovery root public key stored in the MExE device, enabling the operator to use a secure mechanism (involving the disaster-recovery key) to replace the certificate containing the standard operator root public key. It shall not be possible to use the disaster recovery operator root public key to replace the operator root public key unless both public keys are from the same operator.

There shall be no more than one valid operator root public key on the MExE device at any one time. A valid operator root public key on the (U)SIM shall always have precedence over any operator root public key on the ME. Any operator root public key(s) on the ME shall be marked invalid when a valid operator root public key is present on the (U)SIM.

An application signed by an operator shall not be able to execute in the Operator Domain unless the root public key of that operator is installed in the MExE device (either ME or MExE-(U)SIM) and is marked as trusted.

A.6.1.1 Caching of root public keys

The ME shall behave as if it reads the operator root public key from the secure area every time the ME needs the key to verify a signature. Examples of the secure area include an area on a (U)SIM or a secure, persistent area on the ME.

If the ME uses a mechanism for caching public keys, it shall do so in a way that maintains the integrity of the secure area and is consistent with the keys stored in the secure area. With the exception of improved performance, the operation of the device using cached public keys must be indistinguishable from that of a device that reads the key from the secure area every time it uses the key for verification.

No cached version of a key may exist beyond the expiration or termination of the key in the secure area. For example, if the ME caches a root public key held on the (U)SIM, the ME shall purge the cache when the (U)SIM application is stopped (or the SIM card is withdrawn).

A.6.1.2 MExE device actions on detection of valid (U)SIM application and/or power up

This clause defines the sequence of actions on identification by the MExE ME that a valid SIM card, or USIM application on the UICC, has been detected (e.g. through insertion of (U)SIM card, power up of MExE device etc.). More specifically, these actions relate to the enabling or disabling of the operator domain and the status of the operator applications on the ME.

The requirements in this clause ensure that the operator domain on the ME belongs to the same operator as the operator that issued the valid (U)SIM application (if detected) in the MExE device and, if there is an operator root public key (ORPK) on the MExE-(U)SIM, that trusted operator applications on the MExE device were verified using that ORPK.

The ME shall support the use and management of an Operator root public key (ORPK) on the MExE-(U)SIM.

On power up the MExE device shall behave as dictated by figure A.1 "MExE device behaviour on power up" below.

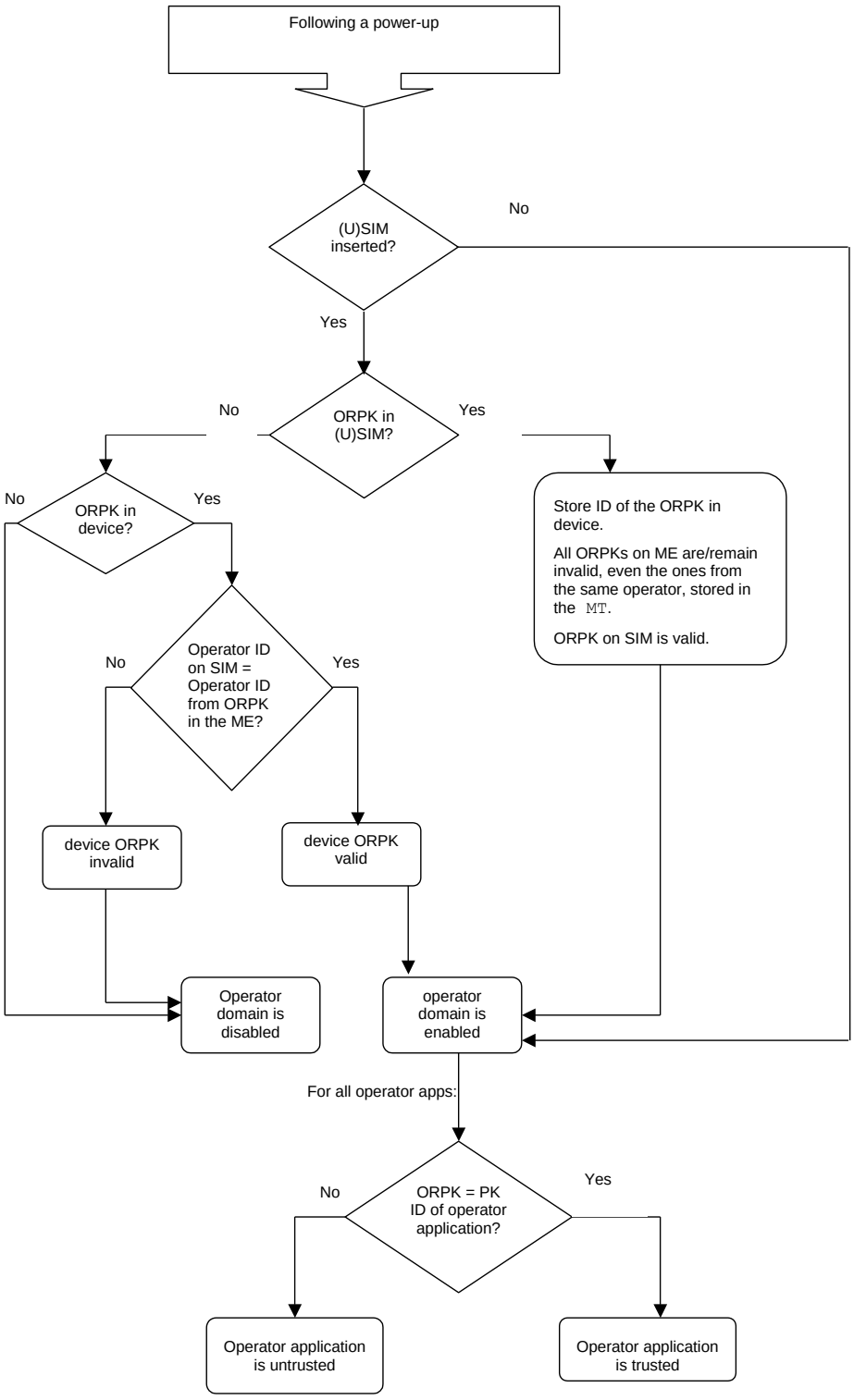

Figure A.1: MExE device behaviour on power up

Note that the procedure in Figure A.1 "MExE device behaviour on power up" checks for a match between the Operator ID on the (U)SIM and the Operator ID from the ORPK in the ME. Currently, one mechanism for defining the Operator ID on the (U)SIM is through use of the MCC+MNC. As an additional note, on DCS1900, the MCC+MNC is 6 digits, but elsewhere it is 5 digits. The MExE device needs to know how many digits to use. However, this is outside the scope of this specification. The implementations of MExE devices need to establish agreements on using the MCC+MNC as the Operator ID on the (U)SIM. Likewise, the implementations of MExE devices need to establish agreements on how to define the Operator ID belonging to the ORPK.

The ME shall only read the ORPK from the MExE-(U)SIM when required and shall not store a ORPK from the MExE-(U)SIM on the ME in a manner inconsistent with that detailed in clause A.6.1.1 "Caching of root public keys".

When an operator root public key stored on the ME is marked as invalid, all operator applications verified using that root public key or by certificates verified by a chain that terminates with that root public key, shall cease operation as soon as possible and shall be marked as untrusted.

Removal of the (U)SIM shall not cause the status (i.e. valid or invalid) of any operator root public key on the MExE device to change.

A.6.2 Manufacturer root public key

The ME shall support secure storage for a certificate containing a manufacturer root public key. The certificate contains a root public key generated by the manufacturer of the MExE device, or by a CA trusted by the manufacturer of the MExE device.

If the ME does not contain a valid manufacturer root public key, then the certificate chain to MExE executable previously executing in the Manufacturer Domain will be invalid, and the MExE executables will be excluded from the manufacturer domain and marked as untrusted.

The user shall not be able to add or delete any type of manufacturer public key (root or contained in a certificate).

The Manufacturer shall put a root public key and optionally its corresponding disaster-recovery key in the ME at the time of manufacture, and use a proprietary secure mechanism (e.g. using the disaster-recovery key) to replace the certificate containing the manufacturer root public key. It shall not be possible to use the disaster recovery manufacturer root public key to replace the standard manufacturer root public key unless both public keys are from the same manufacturer.

An application signed by a manufacturer shall not be able to run in the Manufacturer Domain unless the root public key of that manufacturer is installed in the ME and is marked as trusted.

The manufacturer, and only the manufacturer, may use a secure mechanism to mark as valid/invalid a certificate containing the manufacturer root public key on the ME. It shall only be possible to use this mechanism to mark a certificate containing a new manufacturer root public key on the ME as valid, when all manufacturer root public keys are marked as invalid.

There shall be no more than one valid manufacturer root public key on the ME at any one time. Any other manufacturer root public key(s) on the ME device shall be marked invalid when a different manufacturer root public key is marked as valid on the ME.

A.6.3 Third party root public key

The ME shall support secure storage for at least one certificate containing a third party root public key. The ME shall support the use and management of certificates containing Third Party root public keys stored on the MExE-(U)SIM and in ME. For support of public key management on the SIM and the USIM refer to 3GPP TS 51.011 [27] and 3GPP TS 31.102 [39] respectively. The MExE device may contain root public key (s) generated by CA(s) implicitly trusted by the user. The user will be able to securely install (using a secure transport) or remove Third Party root public keys at any time using a system administrative tool.

The Manufacturer, Operator and Administrator may at their discretion, securely install certificates containing Third Party root public key(s) on behalf of the user, e.g. at the time of manufacture by the Manufacturer. See clause A.10 "Certificate management" for details of Administrator control of Third Party certificate download.

If a Third Party public key is deleted or becomes invalid, then the certificate chain to MExE executables previously executing in the Third Party Domain certified by that public key will become "untrusted".

There may be any number of Third Party root public keys on the MExE device.

The third party domain administrator, i.e. the Administrator (user or other body) shall be able to enable and disable Third Party root public keys by using CCM, see clause A.10.1 "Certificate configuration message (CCM)". The process of adding/removing public keys and enabling/disabling public key are independent.

All third party certificates shall be subject to restrictions imposed by valid certificate configuration messages.

See clause A.10 "Certificate management" for the management of Third Party root public keys.

A.7 Certification and authorisation architecture

If the 3 MExE security domains defined in clause A.1 "Generic security" are not supported, then the certificate and authorisation architecture described in this clause is optionalmandatory.

In order to enforce the MExE security framework a MExE device is required to operate an authentication mechanism for verifying downloaded MExE executables. A successful authentication will result in the MExE executable being trusted; and able to be executed in a security domain (as determined by the root public key of its certification tree).

As the MExE device may want to authenticate content from many sources, a public key based solution is mandatory. Before trusting MExE executables, the MExE device will therefore check that the MExE executable was signed with a private key, for which the MExE device has the corresponding public key. The corresponding public key held in the MExE device must either be a root public key (securely installed in the MExE device, e.g. at manufacture) or a signed public key provided in a certificate. The MExE device must be able to verify certificates, i.e. have the public key (as a root key or in a certificate) corresponding to the private key used to sign the certificate. Support of certificate chains is therefore mandatory.

The requirements on authorisation and certification are given in clause A.7.1 "Certification requirements". An example authorisation and certification process is described in clause A.7.3 "Example certification process".

A.7.1 Certification requirements

A MExE device cannot verify certified MExE executables of a particular domain unless it has a root public key for that particular domain.

Root public keys shall be securely installed in the MExE device, say, at the time of manufacture.

It is recommended that a "disaster recovery" root public key be securely installed on the MExE device, to be used to install new root public keys when all other root public keys on the MExE device are invalid.

Third Party Domain root public keys will typically be installed along with and integrated into the MExE device browser, as is done for PC-based browsers.

A MExE executable can only be verified if the MExE device contains a valid root or certified public keys corresponding to the private key used to sign the MExE executable.

A MExE device shall support at least one level of certificate under operator, manufacturer or Third Party root public keys. The MExE device shall support at least one level of certificate chain analysis in a signed content package, as shown in Figure A.2 "Trust hierarchy".

A certificate (other than one containing a root public key) shall only be considered valid if the signature on the certificate is verified by a valid public key (root or contained in a certificate) already present on the MExE device and if the certificate being verified has not expired.

Public keys shall not be shared between domains.

A.7.1.1 MExE terminal requirements for certificate processing

A MExE device shall support the processing of X509 certificates based on the profiled in the “WAP Certificate and CRL Profile” [47] together with additional requirements defined in the MExE specification, see clause A.9.1.1 “X509 version 3”. The certificate chain depth is still mandated to be one level only, as mentioned in clause A.7.1 “Certificate requirements” and indicated in Figure A.2 "Trust hierarchy".

A MExE device shall support the SHA1WithRSA signature algorithm. The object identifier value can be found in [49]. A MExE device may also support other signature algorithms.

MExE devices may also support the processing of other certificate formats.

NOTE: A specific certificate profile will be defined at a later stage.

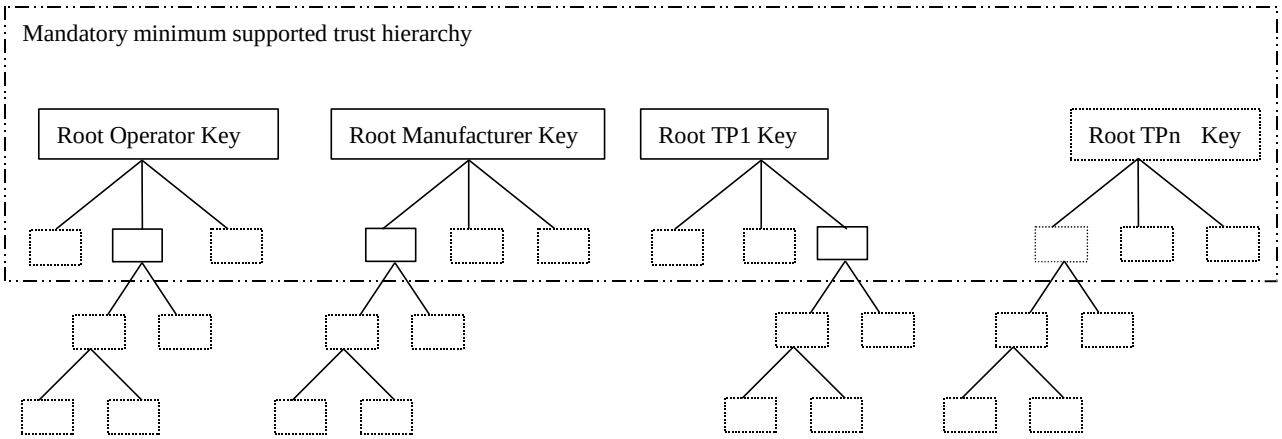

Figure A.2: Trust hierarchy

The boxes below the root keys represent individual public key certificates. The solid boxes represent the minimum MExE, and the dotted boxes represent possible further support for public key certificates (either at the first or subsequent levels).

A.7.2 Certification administration requirements

For control of third party certificates, the MExE device supports storage of a certificate containing an administrator root public key as detailed in clause A.13.1 "Administrator root public key".

This certificate is managed separately from the hierarchy of Figure A.2 "Trust Hierarchy" discussed in clause A.7.1 "Certification requirements". The administrator root public key in this certificate is primarily used for designating an administrator of the third party certificates. Note, the administrator root public key does not implicitly define a security domain, and is used in complement with the root public keys of the operator, manufacturer, and third party domains.

The relationship of the administrator certificate (and root public key) to the management of third party certificates is detailed in part of clause A.10 "Certificate management".

The relationship of the administrator certificate to the mechanism for determining if a third party certificate is trusted is detailed in part of clause A.10.1 "Certificate configuration message (CCM)".

Mechanisms for designating an administrator are detailed in clause A.13.2 "Provisioned mechanism for designating administrative responsibilities and adding third parties in a MExE device".

A.7.3 Example certification process

The following processes might be followed in order to securely download a Third Party application to a MExE device.

Root public keys for a number of Certification Authorities (CAs) are installed in the MExE device, along with the MExE device browser, at manufacture. These root public keys can be used to verify certificates for Third Party MExE executables.

1. A third party software developer generates a private and public key pair (or obtains such a pair from a CA).
2. The third party software developer obtains a certificate for the public key from a CA. The certificate contains the developer public key, signed with the private key of the CA.
3. The 3rd party software developer adds all the certificates required in the key chain in the JAR.
4. The MExE device downloads a MExE executable of the third party software developer.

5. The MExE device verifies the certificate using the root public key, contained in the browser, of the relevant CA, and extracts the third party software developer public key and may store it in the certificate store for future use.

6. The MExE device verifies that the MExE executable was signed using the private key corresponding to the third party software developer public key and installs or rejects the MExE executable accordingly.

All downloaded applications shall follow the procedure described in clause A.7.4 "Certificate Chain Verification" in order to verify the application signature and the certificate chain. If the 3 security domains are not supported, the procedure described in the next clause is optional.

A.7.4 Certificate Chain Verification

This clause presents the procedure of validation of any downloaded MExE executable. It checks for the presence of the signature used to sign the application as well as the presence and integrity of all the certificates needed to successfully verify the signature. As a result, the application under scrutiny is deemed trusted or untrusted, i.e. will be allowed execution in one of the secure domains or in untrusted area, or otherwise the application will not be allowed to be executed and will be deleted. In any outcome of the verification, the user is notified about the result. The user also may wish to see certificate details if the application is allowed to be executed on the MExE device.

The MExE device shall follow "certificate verification" procedure as described below. The procedure shall contain at least the following logical phases (not necessarily in the order stated below):

Signature and Certificate Verification Supported: Checks whether signature and certificate verification procedure is supported on the MExE ME.

Executable with Signature and End Entity Certificate (note): Checks whether the executable contains a signature together with the corresponding end entity certificate.

Valid Application Signature (note): This phase comprises the following checks:

- Check if the signature and the end entity certificate formats are supported by the device. If this check fails, the application is classified as untrusted.

- Check if the signature algorithm is supported/known by the device. If this check fails, the application is classified as untrusted.

- Check if the signature can be cryptographically verified by using the accompanying end entity certificate . If this check fails, the application is not allowed execution and is deleted.

Complete set of Intermediate Certificates Available (note): Checks if all the necessary intermediate certificates (certificates between the RPK and the end entity certificate) are available.

Valid RPK on (U)SIM/ME: Checks if a valid RPK (not expired) exists on the (U)SIM or on the ME that could verify a certificate chain originating from the end entity certificate accompanying the application.

NOTE: These steps could include validation (e.g. expiration, revocation, etc.) checking by means of e.g. OCSP, SCVP, CRL-Consultation, and etc. The use of certificate revocation checking is recommended but is not mandated or defined in this specification.

Certificate Chain Cryptographically Verified: Checks if all the certificates from the end entity certificate to the RPK can be verified cryptographically. Certificate verification shall be performed according to the functional requirements given in clause A.1 "Basic Path Validation" of RFC 2459 [43] excluding revocation checking.

Secure Domains Supported: Checks whether MExE ME supports secure domains.

Only if all the above checks are successful, the downloaded application is deemed trusted and is allowed to be executed in the designated trusted domain (operator, manufacturer, trusted third party). Otherwise, the application is either untrusted (execution in the untrusted area only is allowed) or deleted (execution is not allowed at all) as per the Figure A.3 "Certificate Chain Verification Diagram" and as explained above. The executable shall only be designated into one of the trusted domains, and it shall be possible to verify the certificate chain unambiguously to one and only one root public key.

The MExE ME shall allow for a "user notification" procedure as described below.

It shall be possible to display certificate details to the user if requested, however, since the terminal might not have a display or might not be meant for a human user the methods presented in "user notification" section are not discussed any further in this specification. Figure A.3 “Certificate Chain Verification Diagram” shows an example of the certificate chain verification procedure.

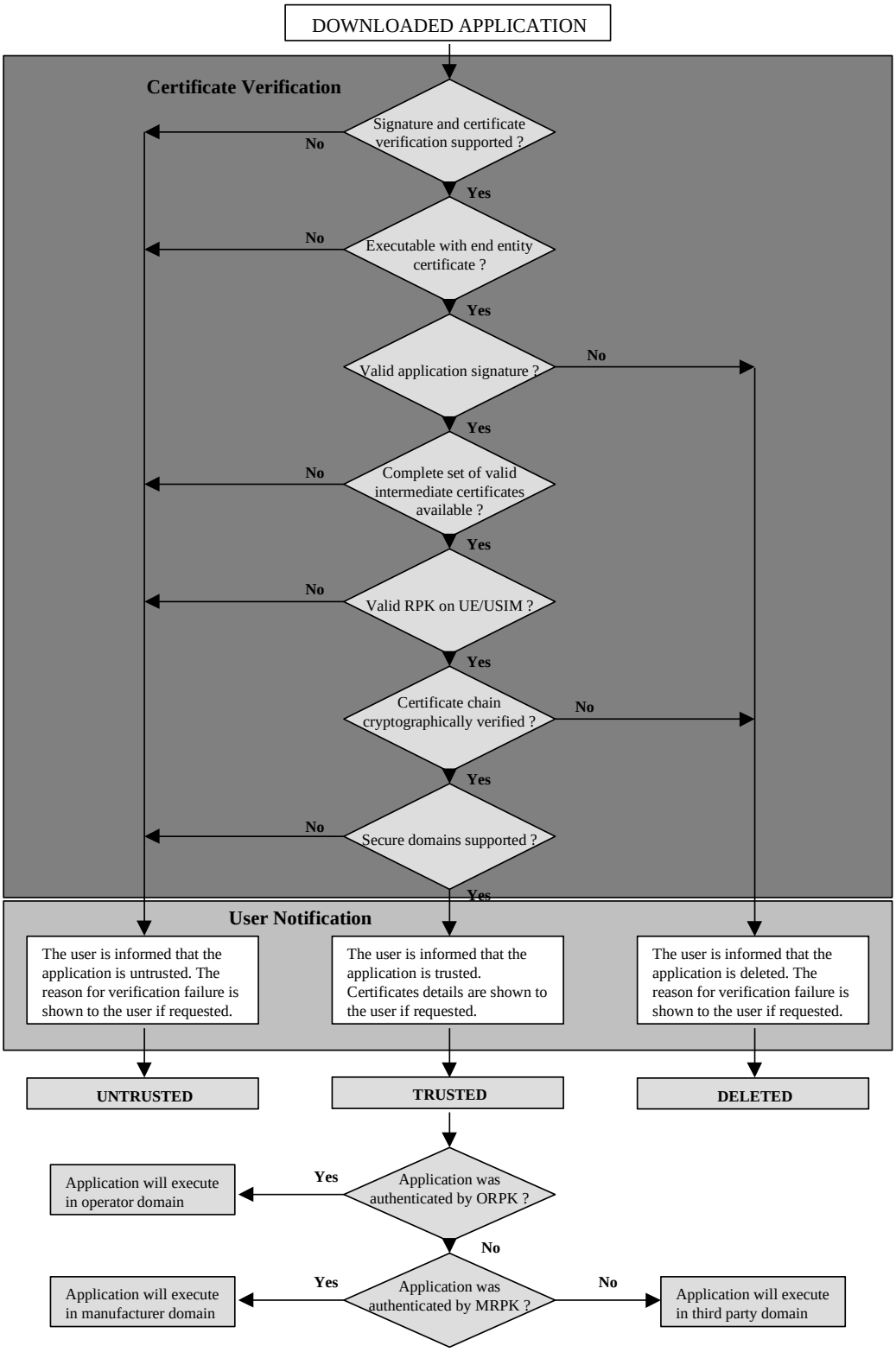

Figure A.3: Certificate Chain Verification Diagram

A.8 Usage of Signed Content

A technology neutral approach for signed content is recommended in section 7.2 of this report.

If the 3 MExE security domains defined in clause B.1 "Generic security" are supported, then a technology neutral approach of signed packages used for installation is mandatory.

The following is an example based on JAR-file technology.

A.8.1 Example of signed packages used for installation

If the 3 MExE security domains defined in clause 6.1 "Generic security" are not supported, then the signed packages used for installation, described in this clause, are optional.

The Java Archive (JAR) file format, if shall be supported on classmark 2 and 3 MExE devices, shall support the following for securely packaging objects that are to be downloaded and installed on the ME. The method for securely packaging objects for MExE classmark 1 devices may be referenced from the WAP specifications in a future release of this specification. A MExE device may support other proprietary means of downloading and installing objects.

The JAR file shall contain a manifest file that has at least the following attribute:

`MExE-Implementation-Type`
The information contained within the manifest file is represented as so-called "name: value" pairs, where "name" is represented by `MExE-Implementation-Type`. Groups of name-value pairs are known as a "section", where sections are separated from other sections by empty lines.

The `MExE-Implementation-Type` value shall be one of the following:-

- "**`MExENativeLibrary`**"

  in the case of a MExE Native Library (as described in 8.3.2 "Installing MExE native libraries" in 23.057);

- "**`TTPCertificate`**"

  in the case of a certificate containing a 3rd party root public key (as described in A.8.2 "Installation of root certificates in a signed data package");

- "**`ManufacturerCertificate`**"

  in the case of a certificate containing a manufacturer root public key (as described in A.8.2 "Installation of root certificates in a signed data package");

- "**`OperatorCertificate`**"

  in the case of a certificate containing an operator root public key (as described in clause A.8.2 "Installation of root certificates in a signed data package");

- "**`AdminCertificate`**"

  in the case of an administrator certificate, which shall consist of a section containing both the administrator certificate and a CCM (as described in clause A.8.2 "Installation of root certificates in a signed data package"); or

- **"OrdinaryTTPCertificate"**

  in the case of a certificate or certificate list containing 3rd party public key(s). An example of a certificate list syntax can be found in [52]

- **"OrdinaryManufacturerCertificate"**

  in the case of a certificate or certificate list containing manufacturer public key(s). An example of a certificate list syntax can be found in [52]

- **"OrdinaryOperatorCertificate"**

in the case of a certificate or certificate list containing operator public key(s). An example of a certificate list syntax can be found in [52]

- "**CCM**"

  in the case of a CCM (as described in clause A.8.2 "Installation of root certificates in a signed data package"); or

- *-free-format-value-*

  in the case of proprietary binaries or Java classes such as native DSP code, provisioned functionality upgrades and patches (as described in clause A.8.3 "Installation of other signed data").

Refer to [42] for full details of how to encode the "name: value" pairs and "section" in a JAR manifest file.

See Figure A.4 "Signed packages". When a download of a JAR file is completed, the system installer shall read the manifest to determine what types of files are contained in the JAR, and install them appropriately.

Note that a signed package containing a library which does not have a manifest attribute "MExE-`Implementation-Type: MExENativeLibrary`" shall be considered to be some type of upgrade to libraries that are intrinsically part of the MExE device implementation rather than a "MExE native library". E.g.

```
MExE-Implementation-Type: ManufacturerUpgrade (something.dll)
```

(Recommended behaviour for the server is that it uses the capability information supplied from the MExE device to determine how to offer appropriate upgrades.)

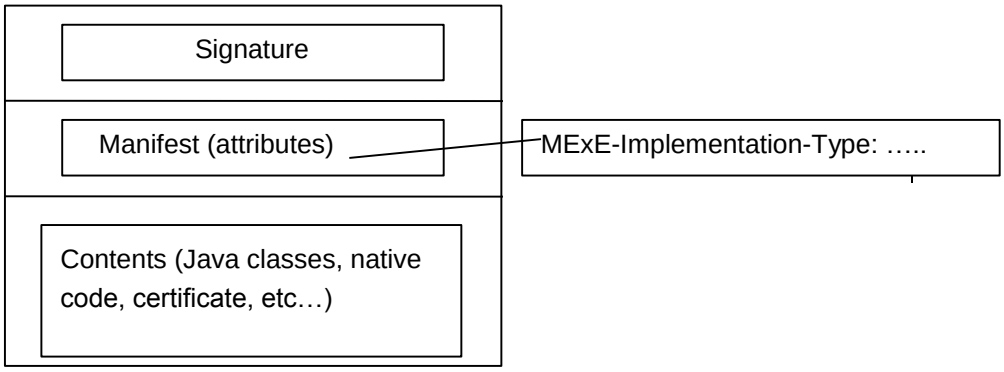

Figure A.4: Signed packages

A.8.2 Installation of root certificates in a signed data package

Root certificates in a signed package (whose signature verifies as described in clause A.6 "Root Public keys" to the Manufacturer root, Operator root, or the Administrator root), may be installed to the root public key store on the MExE device. Note that the certificate thus packaged does not necessarily belong to the manufacturer domain. The types of certificate that can be present and installed by packages are given in table A.4 "Allowed certificate types in signed packages". The MExE device shall store the root public key as indicated by the certificate type.

When a certificate containing an Administrator root public key is thus contained in a signed package, the signed package (e.g. a JAR file in the case of Java based MExE classmarks) shall contain two files: the Administrator root public key and the CCM.

Table A.4: Allowed certificate types in signed packages

Signature on Package	Allowed Certificate types in package
Administrator	Third Party
Manufacturer	Administrator, Manufacturer, Operator, Third Party
Operator	Administrator, Operator, Third Party

6.8.3 Installation of other signed data

A signed package of proprietary binaries or Java classes such as native DSP code, provisioned functionality upgrades and patches, whose signature verifies as described in clause 6.6.2 "Manufacturer root public key" as belonging to the Manufacturer Domain may be installed. The use of such binaries is outside the scope of MExE, but the manufacturer shall be responsible for ensuring that the integrity of MExE is not compromised.

Support of this feature is optional.

A.9 Certificate format

This clause defines a possible feature addition to the X.509 certificate. The feature is optional.

A.9.1 Certificate extension for removal of network access

MExE defines the certificate extension (attribute) " access-Restriction". If the access-Restriction extension is present in a certificate used to verify the signature on a trusted application or in any certificate in the certificate chain used to verify that signature, then the application shall not be permitted the capabilities listed under "network service access" in the security table, (table 6 "Security domains and actions"). This restriction applies irrespective of any user permission for network service access that may or may not be requested by the application and/or given by the user.

The extension prevents the trusted applications of developers who do not need network service access from writing applications that can perform network service access.

The support of this extension in the operator domain is mandatory. The support of this extension in the manufacturer and third party domains is optional.

The extension is defined for X.509 version 3 only. Support for WTLS, X9.68 certificate formats is for further study.

A.9.1.1 X.509 version 3

The MExE certificate format as specified in clause A.7.1.1 “MExE terminal requirements for certificate processing” shall support the X.509 version 3 access-Restriction extension.

X509 version 3 provides a mechanism to define extensions. An Object identifier (OID) is defined for each private extension as defined in X509 [26]. The extension is defined to be within the ETSI Object Identifier (OID) name space.

This extension shall apply irrespective of the presence or otherwise of any other X.509 key usage or extended key usage field.

Normal use of the "critical" flag for extensions apply. That is, if this extension is marked as critical in the certificate used to verify the signature on the application or in any certificate in the chain used to verify the signature and this extension cannot be processed in the MExE device then the certificate shall be considered invalid.

The syntax of the extension is defined in annex C “Access restriction certificate extension” in 23.057.

A.10 Certificate management

If the 3 MExE security domains defined in clause A.1 "Generic security" are not supported, then the certificate management described in this clause is mandatoryoptional.

The manufacturer may load initial third party certificates on the ME. Downloaded certificates shall be verified by an existing trusted certificate and placed in the domain defined by the root public key at the top of the verification chain for the downloaded certificate.

The administrator root certificate shall be provided on the (U)SIM if support for certificate storage on the (U)SIM exists (e.g. MExE-(U)SIM) or in the MExE device. For (U)SIMs not having certificate storage the administrator root may be downloaded using the root download procedure described in clause A.13.4 "Administrator root certificate download mechanism".

The actions that may be performed for a given certificate are:

- addition;
- deletion;
- mark un-trusted (un-trusted certificates cannot be used to verify applications or other certificates. This process may be preferred to certificate deletion as there is a chance that the certificate may become trusted again in the near future);
- mark trusted (marking as trusted is the process of allowing an untrusted certificate to come into use again);
- modify fine grain access permissions (proposed as a future enhancement).

The ability to perform these actions depend on the certificate type being modified as well as the access level of the entity performing the operation.

Users may add a third party certificate as long as it is certified by an existing trusted certificate. Using a provisioned functionality, users may delete Third Party certificates.

The Administrator may mark trusted/untrusted Third-Party certificates using Certificate Configuration Messages (see clause A.10.1 “Certificate configuration message (CCM)”.

Users cannot add or delete any Operator or Manufacturer certificate containing a root public key.

An example of public key infrastructure certificate management protocols can be found in [33].

A.10.1 Certificate configuration message (CCM)

If the 3 MExE security domains defined in clause A.1 "Generic security" are not supported, then the certificate configuration message described in this clause is mandatedoptional.

The MExE device shall use the CCM to determine the third party certificates (and only the Third Party certificates) that are trusted for use on the MExE device. The CCM shall only be used to enable or disable third party certificates and can not be used to delete certificates. The CCM may be periodically fetched or downloaded to a MExE device by the Administrator to dynamically configure the third party list using the mechanisms defined in clause A.10.1.4 "Authorised CCM download mechanisms".

The Certificate Configuration Message shall be as shown in Figure A.5 "Format of a CCM". This message is essentially a simplified version of a certificate revocation list to satisfy a particular use case. More complex usage requires a full certificate revocation list.

The MExE device may additionally support other means of enabling/disabling root certificates.

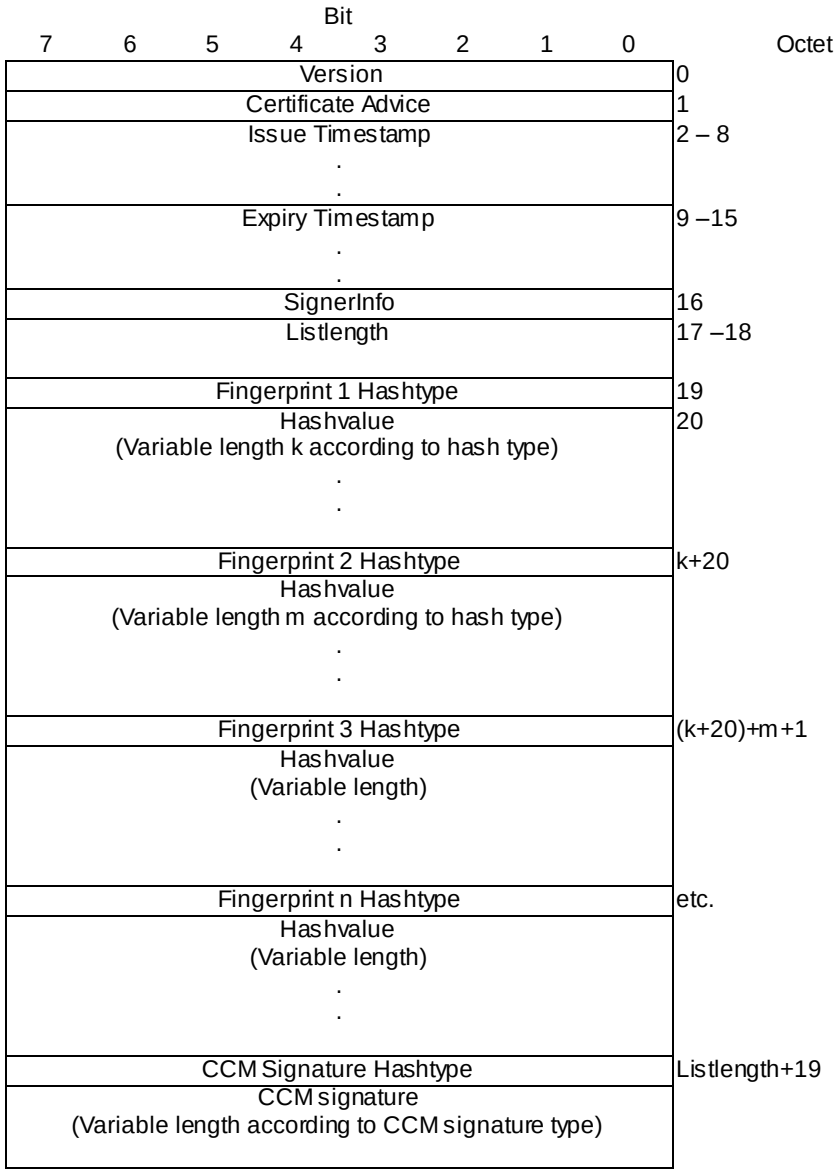

Figure A.5: Format of a CCM

Version = The CCM format version is 0. All other values are reserved for future use.

Certificate Advice = enumerated { enable all present and future Third Party certificates (0), disable all present and future Third Party certificates (1), enable present list only (2),enable CCM list (3), disable CCM list (4) }. All other values are reserved for future use.

Issue and Expiry Timestamps = Fields used to identify the issue and expiry date of the CCM. The issue timestamp indicates a time before the current time of day (GMT) when a CCM message must be considered invalid. The expiry timestamp (GMT) identifies the time when a CCM is to be deemed no longer valid. The receiver shall use these parameters to detect a replay attack. A MExE device maintains information on the last valid CCM message received. A replay attack is an attacker replaying a previous valid CCM message to a MExE device in order to change the security settings. This is particularly dangerous for CCM messages used to enable certificates. Administrators should try and set the expiration time to be no longer than the next expected system update time of CCM information. CCM messages used to enable-all (rather than disable-all) certificates should be very short lived as the danger of these being used in a replay attack should be considered serious.

The encoding of time (GMT) shall be coded as an OCTET SEQUENCE of seven octets in length as follows:

Octet 0	1	2	3	4	5	Octet 6
Year		Month	Day	Hour	Minute	Second

Element	Size (bits)	Range
Year	16	$(0 – 65535)_{10}$
Month	8	$(1 – 12)_{10}$
Day	8	$(1 - 31)_{10}$
Hour	8	$(0- 23)_{10}$
Minute	8	$(0 – 59)_{10}$
Second (see note)	8	$(0 – 60)_{10}$
NOTE: The second field range includes the value 60 in order to accommodate leap seconds.		

For example, 1st January, 2001 00:00:30 would be encoded as: 07 d1 01 01 00 00 1E.

SignerInfo = one octet indicating the type of signer information for this CCM. The only currently defined value is device-admin = 0. In this case, no further signer information follows as it is implicit. All other values are reserved for future use.

Listlength = The total length of the fingerprint list not including the final CCM signature. Shall be zero when certificateAdvice = enable-all, disable-all or enable present list.

Hashtype = enumerated { signature (0), MD5 (1), SHA-1 (2) } All other values are reserved for future use.

The length of the Hashvalue field, number of octets output by the selected hash type, is 16 for MD5 [23] or 20 for SHA-1 [24].

The list entries shall contain certificate *fingerprints* in the form of hashes of the encoded signed certificates. The full hash output for the specified algorithm shall be used to generate the fingerprint. A list generator shall check to insure that no two list entries match when creating a list. For an X509v3 [26] or X9.68 (currently being drafted) certificate the fingerprint hash shall be computed over the ASN.1 encoded signed certificate object, first octet to last octet. For WTLS certificates the hash shall be computed over the signed WTLS certificate in network transmission format, first octet to last octet.

The signature type and length shall be indicated by the administrator certificate, which shall be present on the MExE device. If no administrator certificate is on the MExE device or if the signature is not verified, the message shall be rejected.

Upon receipt of a valid certificate configuration message the MExE device shall go through the third party certificate list, computing fingerprints if they are not stored with the certificate and enabling or disabling each certificate according to the following conditions:

- certificateAdvice is enable-all — all Third Party certificates shall be enabled;
- certificateAdvice is disable-all — all Third Party certificates shall be disabled;
- certificateAdvice is enable present list only — enable all Third Party certificates currently on MExE device, do not enable any future certificates (this option allows the list to be frozen at time of manufacture) until Administrator changes;
- certificateAdvice is enable-list — if its fingerprint occurs in the CCM, it shall be enabled, otherwise it shall be disabled;
- certificateAdvice is disable-list — if its fingerprint occurs in the CCM, it shall be disabled, otherwise it shall be enabled.

For future releases, the setting of fine grained permissions for each certificate is expected to be supported.

An implementation shall keep track of the domain that authorised a given executable. If a CCM message is received while MExE executables are currently executing, the implementation shall check to ensure that any executables no longer in the Third Party domain, have their permissions re-configured appropriately and actions that are no longer permissible are terminated.

A.10.1.1 CCM numbering convention

Bits are grouped into octets. The bits of an octet are shown horizontally and are numbered from 0 to 7. Multiple octets are shown vertically and are numbered from 0 to n.

A.10.1.2 CCM order of transmission

Frames are transferred in units of octets, in ascending numerical octet order (i.e., octet 0, 1, …, n-1, n). The order of bit transmission is specific to the underlying protocols used to transport the CCM.

A.10.1.3 CCM field mapping convention

When a field is contained within a single octet, the lowest bit number of the field represents the lowest-order value. When a field spans more than one octet, the order of bit values within each octet progressively decreases as the octet number increases. In that part of the field contained in a given octet the lowest bit number represents the lowest-order value.

For example, a 16 bit number can be represented as shown in Figure A.6 "Field mapping convention".

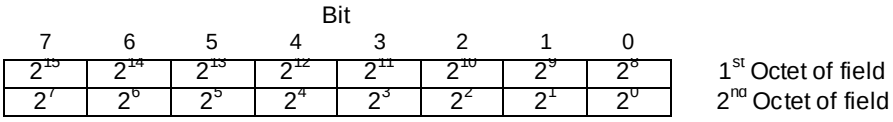

Figure A.6: Field mapping convention

A.10.1.4 Authorised CCM download mechanisms

The download of third party certificate lists by a remote administrator shall be performed by using a secure mechanism as defined below. The download mechanisms shall use HTTP over IP and/or the WAP Protocol. The URL from which the CCM is downloaded shall be in the administrator certificate if the CCM was not downloaded with the Administrator certificate. The format for storing the URL information with the certificate shall be as shown in Figure A.7 "CCM Message URL storage format":

Urltype	**CharacterSet**	**UrlLength**	**URL**

Figure A.7: CCM Message URL storage format

Urltype= one byte, enumerated {WAP (0), HTTP (1)}. All other values are reserved for future use.

CharacterSet = one byte, Internet Assigned Numbers Authority assigned character set.

UrlLength = one byte unsigned integer, length of the URL in octets.

URL = a field where the format for storing the URL information in the certificate shall be defined as part of the enhanced administrator mechanism.

When the administrator is changed, then the CCM shall also be changed. If there is URL information with the certificate as described in Figure A.7 "CCM Message URL storage format", then the new CCM shall be obtained using the URL. If the Administrator certificate was downloaded in a JAR file, the CCM shall be obtained from the same JAR file.

A.11 Separation of I/O streams

Support of the separation of I/O streams is mandatory.

Except for the MExE Classmark 3 executables (MIDlets) from the same MIDlet Suite, tThere shall be strict separation of the user interface input and output streams between different MExE executables, i.e. it shall not be possible for one MExE executable to access the user interface input or output of another MExE executable. In particular, it shall not be possible for an untrusted MExE executable to access the user interface input and output destined for or proceeding from a trusted MExE executable. This requirement prevents a malicious MExE executable from eavesdropping on or interfering with communications between the user and a trusted MExE executable (for instance, intercepting PINs or passwords).(This requirement is to prevent a long lived malicious MExE executable from eavesdropping upon on interfering with the user to MExE executables communications, for instance PINs, of a trusted MExE executable).

6.12 Core software download

Support of core software download is optional.

Core software download enables the MExE device radio, characteristics and properties to be updated by changing the software in the MExE device. E.g. a new CODEC may be loaded into a MExE device, a new air interface, etc. This process could include the transfer of executable code and software patches over the air.

This updating of core software (e.g. the Software Defined Radio (SDR) concept) can in principle be generically supported within the MExE framework by a MExE service that executes in the manufacturer security domain, and uses handset manufacturer proprietary APIs. Possible scenarios for the support of this functionality include:

- A MExE service that can be transferred to, and executed in, the manufacturer domain. The service would use manufacturer APIs to perform the software update, radio re-configuration, etc.
- A core software download application that executes in the manufacturers' domain that acts like a user agent in conjunction with a server to transfer software as needed or requested by the user. The core software download application uses manufacturer APIs to perform the software update, radio re-configuration, etc.

Similar functionality may be supported by a downloaded MExE service using manufacturer's OEM classes. All such OEM classes shall comply with the MExE security requirements in table 6 "Security domains and actions" and table A.2 " Executable permissions for untrusted MExE executables".

The support of core software download functionality in a MExE device shall only be under the control of the MExE device manufacturer.

A.13 Administrator Concept

A.13.1 Administrator root public key

To help with the control of Third-Party certificates, the ME shall support secure storage for a certificate containing an administrator root public key. The ME shall support the use and management of a certificate containing an Administrator root public key stored on the MExE-(U)SIM and in the ME. The ME shall behave according to clause A.13.3 “Determining the administrator of the MExE MS”. For support of public key management on the SIM and the USIM refer to 3GPP TS 51.011 [27] and 3GPP TS 31.102 [39] respectively.

A secure mechanism may be used to mark as valid/invalid a certificate containing the administrator root public key on the MExE device. It shall only be possible to use this mechanism to mark a certificate containing a new administrator root public key on the ME as valid, when all administrator root public keys are marked as invalid.

There shall be no more than one valid administrator root public key on the MExE device at any one time. A valid administrator root public key on the (U)SIM shall always have precedence over any administrator root public key on the ME. Any administrator root public key(s) on the ME shall be marked invalid when a valid administrator root public key is present on the (U)SIM.

The MExE device shall support the administrator designation mechanism explained in clause A.13.2 "Provisioned mechanism for designating administrative responsibilities and adding third parties in a MExE device" and the secure downloading of CCMs explained in clause A.10.1.4 “Authorised CCM download mechanisms”.

The user shall not be able to delete an administrator root public key or certificate.

The system shall support a mechanism (as part of a provisioned functionality and/or inherently part of the MExE implementation) allowing the owner of the MExE device to manage the administrator root public key (including the download of a new administrator root public key) as defined in clause A.13.3.2.1 "Administrator of the MExE device is the user". This mechanism shall be secure so that only the owner can use this functionality.

The administrator root public key can be downloaded to the MExE device as described in clause A.13.4 "Administrator root certificate download mechanism".

If the Administrator root public key is stored in the (U)SIM, the ME shall only read the Administrator root public key from the MExE-(U)SIM when required and shall not store the Administrator root public key from the MExE-(U)SIM on the ME in a manner inconsistent with that detailed in clause A.6.1.1 "Caching of root public keys".

See clause A.10 "Certificate management" for the management of Administrator root public keys.

The same root public key may be used for both the Administrator role and the operator or manufacturer domain. This facility does not imply any increased right of the manufacturer or operator to take the Administrator role.

If the same root public key is used for the operator domain and Administrator role and this root public key is stored on the MExE-(U)SIM (see [27] and [39]), there shall be separate entries relating to each use of the root public key in the operator and administrator trusted certificate directory files. These entries in the operator and Administrator trusted certificate directory files may point to the same root public key in the certificate data file.

If the root public key to be shared is not stored on the (U)SIM, then procedures relating to this are out of the scope of this specification.

A.13.2 Provisioned mechanism for designating administrative responsibilities and adding third parties in a MExE device

If the 3 MExE security domains defined in clause A.1 "Generic security" are not supported, then the administrator concept described in this clause is optional.

All applications in the Third-Party security domain are to be signed by a key which shall be verified back to a Third Party root public key on the MExE device. The Third Party root public keys shall be managed (e.g. addition/mark trusted/mark untrusted) by an administrator that is designated by the owner of the MExE device using the MExE administrator provisioning mechanism. A mechanism is required to be provided to enable the owner of the MExE device to dynamically assign an administrator. The mechanism shall support the following cases:

- the user is the owner;
- the owner is at a remote location. In this case the owner could be the operator, a service provider or a third party;
- the owner of the MExE-(U)SIM wants to be a temporary administrator.

A.13.3 MExE administrator determination mechanism

The administrator of the MExE device shall be determined by a two part logical process with the first part shown in the flowchart in Figure A.8 "MExE administrator determination mechanism". The second part of the logical process is in Figure A.9 "MExE administrator determination mechanism, for MExE-(U)SIM supporting third party certificates".

During power-up or MExE-(U)SIM insertion event, the provisioned mechanism shall look for an administrator root public key that is stored on the MExE-(U)SIM.

A.13.3.1 Determining the administrator of the MExE device

If an administrator root public key cannot be found on the MExE-(U)SIM, the provisioned mechanism shall look for one on the MExE device. This leads to the following two cases:

- administrator root public key is absent

  if the administrator root public key is absent, then the user shall automatically become the administrator of the MExE device.

- administrator root public key is present

  if an administrator root public key is present, this root public key shall be used for all remote administration authentication, implying that the owner of the administrator root public key is the administrator. Note that the owner of the administrator root public key could be the user.

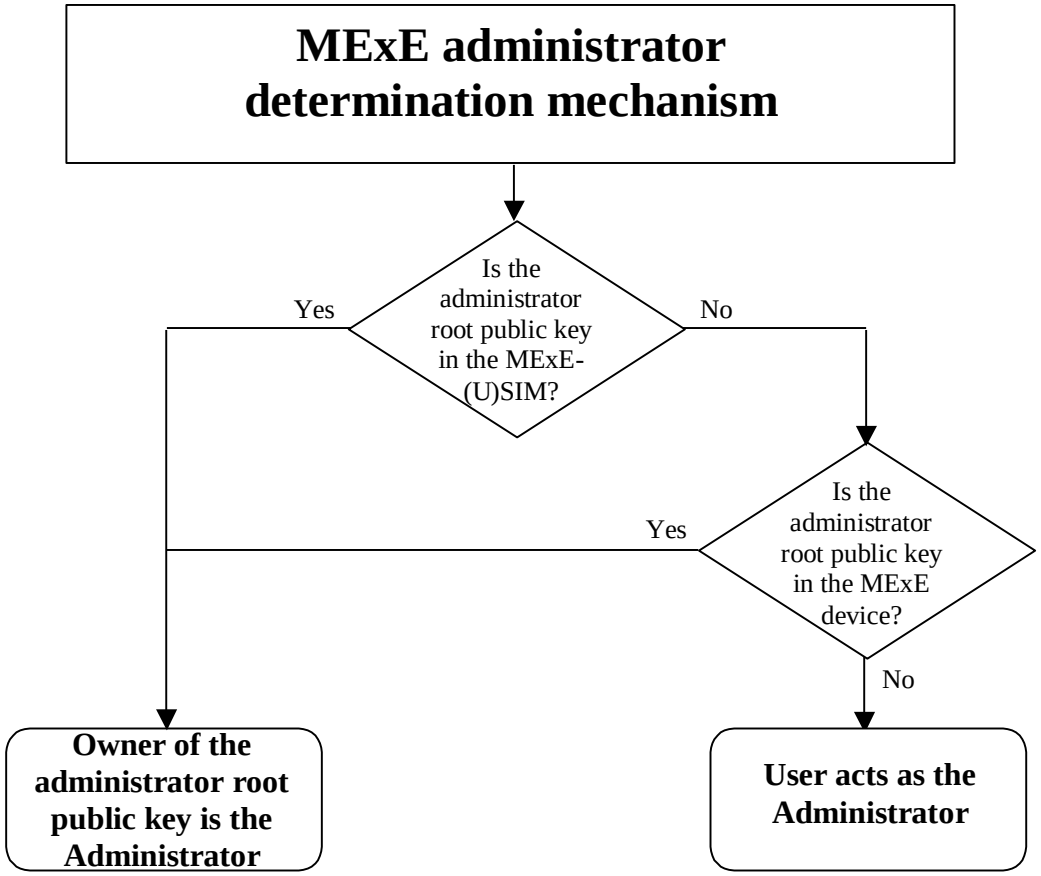

Figure A.8: MExE administrator determination mechanism

A.13.3.2 Determining the administrator of the MExE device, for MExE-(U)SIM supporting third party certificates

The second part of the administrator determination mechanism is subsequently defined (see Figure A.9 "MExE administrator mechanism, for MExE-(U)SIM supporting third party certificates "), and shall be initiated after a power-up or MExE-(U)SIM insertion event is processed.

The following clauses A.13.3.2.1 "Administrator of the MExE device is the user" and A.13.3.2.2 "Administrator of the MExE device is not the user" assume that Third Party certificates can be added using the MExE-(U)SIM, however Third Party certificates may be added using a non-(U)SIM approach (e.g. inserted at the time of manufacture, signed package download etc.).

A.13.3.2.1 Administrator of the MExE device is the user

If the administrator is the user, then a check shall be made to determine whether there is a MExE-(U)SIM. If a MExE-(U)SIM is present, then a check shall then be made to determine whether there is a Third Party or an Administrator certificate containing a root public key in the MExE-(U)SIM. The second part of the administrator determination mechanism shall allow the MExE device to determine (via a format) what type of certificate is present:

- certificate present - third party (CP-TP)

  If a Third Party certificate containing a root public key is present in the MExE-(U)SIM then this certificate shall be considered by the MExE device as a Third Party certificate, whilst that valid MExE-(U)SIM application is present in the MExE device. The user shall be queried to allow or disallow the certificate as a Third Party.

- certificate present - administrator (CP-Admin)

  If a temporary Administrator certificate containing a root public key is present in the MExE-(U)SIM, the user shall be queried whether to allow the certificate on the MExE-(U)SIM to take temporary control of the third party domain. By temporary control, it is meant that once the card is removed the administrator reverts back to the user administrator settings (i.e. the Administrator becomes the User). The above mechanism implies that the previous configuration settings for the administrator shall be saved, so that they may be restored. If the user disallows the MExE-(U)SIM certificate, the Third Party Domain shall not be able to use any of the network capabilities in the third party domain as identified in the network access section of the security table 6 "Security domains and actions".

If an administrator certificate is not present on the MExE-(U)SIM and the administrator is the user, the user shall continue to be the administrator and may make use of all functionality.

A.13.3.2.2 Administrator of the MExE device is not the user

If the administrator is not the user, then a check is made to determine if there is a MExE-(U)SIM. If a MExE-(U)SIM is present, then a check is made to see if there is a Third Party or an Administrator certificate containing a root public key in the MExE-(U)SIM.

If an Administrator certificate containing a root public key is present in the MExE-(U)SIM, then a comparison is made of this certificate's root public key with the Administrator root public key on the MExE device for the following cases:

- Case (a): they are the same;
- Case (b): they are not the same, but the MExE device certificate is cross-certified with the MExE-(U)SIM certificate (a cross-certificate exists on the MExE device);
- Case (c): they are not the same, but the MExE device certificate has a line of trust back to the MExE-(U)SIM certificate domain;
- Case (d): they are not the same.

If the owner of the public key in the certificate on the MExE-(U)SIM is to be a temporary administrator (CP-Admin), then in cases (a), (b) and (c), the temporary administrator shall be the owner of the CP-Admin root public key. In case (d), the Third Party domain shall not use any of the network capabilities in the third party domain as identified in the network access section of the security table 6 "Security domains and actions".

If the certificate is to be a Third Party certificate containing a root public key, then the certificate (CP-TP) shall be verified with the CCM and based on the content and permissions of the CCM, the certificate shall be added to the Third Party list or rejected.

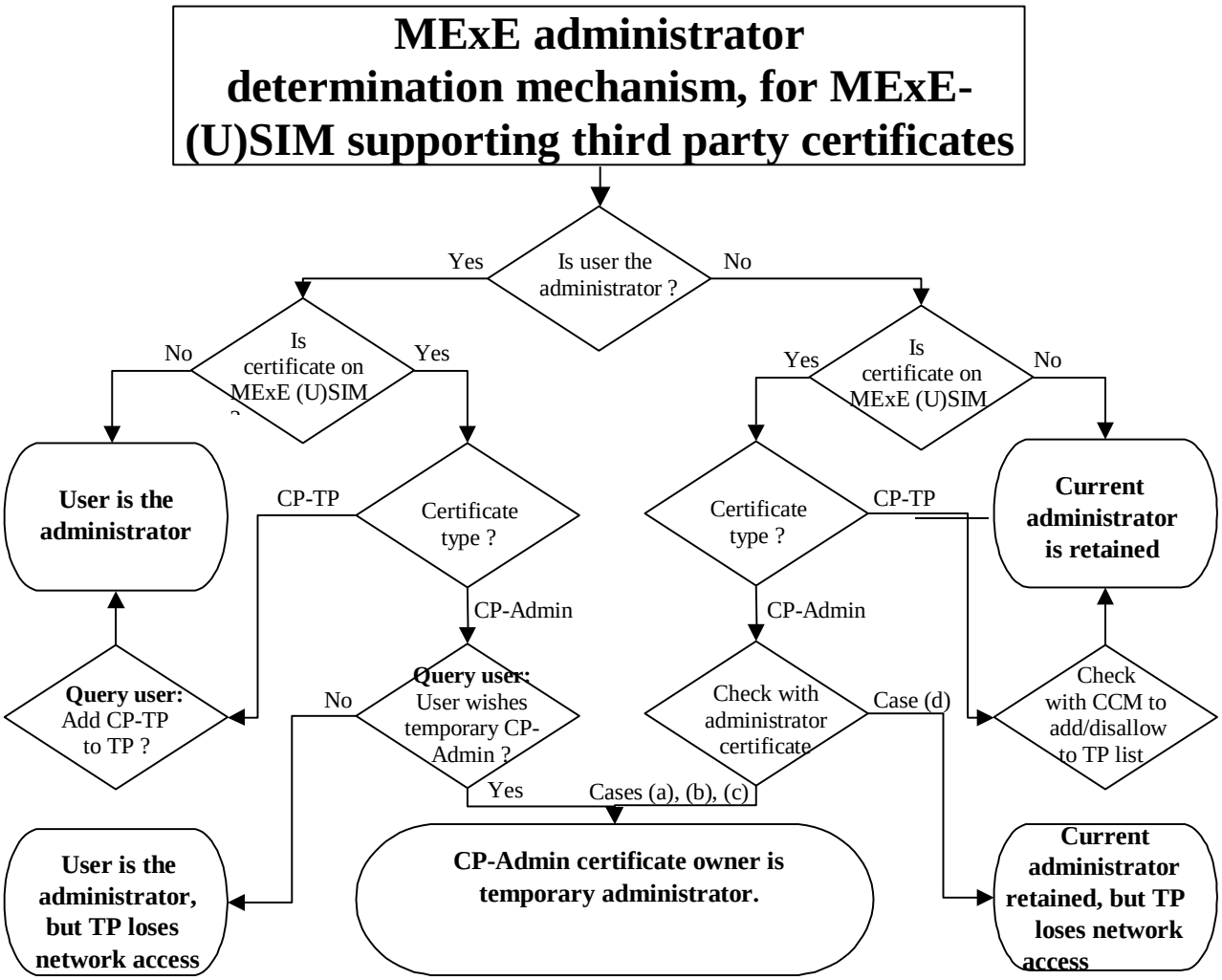

Figure A.9: MExE administrator determination mechanism, for MExE-(U)SIM supporting third party certificates

A.13.4 Administrator root certificate download mechanism

MExE devices supporting (U)SIMs without certificates shall at least support the following procedure to download the administrator root certificate.

1. Upon sign-up with an administrator the user and administrator will make contact.
2. The administrator service centre will obtain any required information from the user and inform the user by SMS or other means of the location of the administrator root certificate.
3. The user will initiate the download of the Administrator root certificate using a signed package.
4. Once the procedure is complete the MExE device shall compute the hash of the received Administrator certificate containing root public key.
5. The user will contact the administrator and enters on the MExE device at least the first 8 bytes using decimal value of the hash of the Administrator root public key information provided by the administrator . The MExE device compares the beginning of computed hash value and the abbreviated hash value entered by the user If these two values are the same ,the provisioning process will be complete. If the two values are different this shall be indicated to the user who should inform the administrator of this.

Alternative methods to download an administrator root certificate may be used where appropriate but must insure that the certificate is received by the MExE device unaltered.

Annex B:
Change history

Change history							
Date	TSG #	TSG Doc.	CR	Rev	Subject/Comment	Old	New
2002-12	T#18	TP-020274			First approved version	2.0.0	6.0.0