3GPP TSG-T (Terminals) Meeting #17
Biarritz, France, 4 - 6 September, 2002

Tdoc TP-020251

Presentation of Specification/Report to TSG-T

Presentation to:	**TSG-T Meeting #17**
Document for presentation:	**TR 22.857 Run-Time Independent Framework Feasability Study; Version 1.0.0**
Presented for:	**Information**

Abstract of document:

This document discusses the need for a Runtime Independent Framework for MExE, what it is, and how it can be provided with a minimum of changes to the existing specification. It consists of a benefits analysis and a feasibility study on the creation of a framework enabling the application of MExE to arbitrary runtime environments.

Changes since last presentation:

New technical report.

Outstanding Issues:

Binding executables to certificates and metadata, Root key certificate packaging and metadata, Handling of classmarks.

Contentious Issues:

Yes

Debate ongoing in T2

3GPP TR 22.857 V1.0.0

Technical Report

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

GLOBAL SYSTEM FOR
MOBILE COMMUNICATIONS

The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP.
The present document has not been subject to any approval process by the 3GPP Organizational Partners and shall not be implemented.
This Specification is provided for future development work within 3GPP only. The Organizational Partners accept no liability for any use of this Specification.
Specifications and reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organizational Partners' Publications Offices.

Keywords

<keyword[, keyword]>

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

Copyright Notification

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 for MExE, 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, 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 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 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 runtime environment, or 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 before 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 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.

5 Reusable technology: An alternate approach

An alternative to classmarks for integration of runtime technologies into MExE is to separate out the components and aspects of MExE that are independent of any runtime technology, and reusable, from the specific runtime technologies of the classmarks. Along with this, aspects of the MExE service environment can be enhanced to support the use of any runtime within the MExE framework. This document refers to the creation of an explicit set of runtime independent MExE technologies and a binding framework that can be applied to any runtime 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 runtimes and profiles to deploy, and when, while doing this within the security and confidence that the MExE framework provides. It does this by making a clear distinction between the runtime dependent aspects of the MExE specification from the reusable, runtime-independent parts of the specification, filling in some small missing pieces of “glue” technology, and explicitly stating conformance requirements for integrating with the resulting runtime independent framework.

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

- Security infrastructure
- Service environment
- Core software update
- Multiple classmark 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 current 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 extremely 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 the specification.

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. These will be discussed later.

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 section 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 enablement or disablement 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 second case 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 classmark support

MExE defines the way that applications and classmarks are to behave in the presence of other classmarks. This idea can be easily extended to include, both, 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 and TS 23.057 that are necessary to introduce the RTIF into the current 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 and TS 23.057 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 and TS 23.057. Since, by definition, the RTIF does not require creation of new classmarks, a runtime will need some criteria of conformance other than classmark conformance.

Therefore, to support the RTIF, a section in the TS 22.057 or TS23.057 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. The specific, corresponding sections of the 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 MExE 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. Therefore, 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, 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. It would increase clarity if the MExE specification were modified 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 classmark 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 classmark, or more, on the same device.

In general, the approach taken in the current specification states that applications executing in a classmark running on a device supporting multiple classmarks must behave the same and meet the same requirements as when executing on a device supporting only that classmark. 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. Of course, this solution is tied to Java technology, and, practically, is obviously 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 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 proposed, the creation of the RTIF imposes no additional requirements on future classmarks. However, it may be desirable to impose RTIF integration as a necessary precursor for a runtime technology to be considered as a MExE classmark. This could provide great value to 3GPP and the wireless application community, and is not, simply, a means of decreasing the number of future proposed 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 MExE. In this way, the RTIF could serve as a first step in an integration path for new runtime technologies into MExE.

In summary, the RTIF can support the current classmark structure, or it can be used to support a system without classmarks. Furthermore, it does not require any classmark constructions 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 and Stage 2 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 for MExE is both feasible and useful. It outlines the aspects of MExE 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:
Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
2002-09	T#17	TP-020207	-		Presented to TSG-T for Information	1.0.0	1.0.0