ETSI GS ECI 001-4 V1.1.1 (2017-07)

Embedded Common Interface (ECI) for exchangeable CA/DRM solutions; Part 4: The Virtual Machine

Disclaimer

The present document has been produced and approved by the Embedded Common Interface (ECI) for exchangeable CA/DRM solutions ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference
DGS/ECI-001-4

Keywords
CA, DRM, VM

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00   Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the only prevailing document is the print of the Portable Document Format (PDF) version kept on a specific network drive within ETSI Secretariat.

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

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

Contents

List of Figures

List of Tables

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This Group Specification (GS) has been produced by ETSI Industry Specification Group (ISG) Embedded Common Interface (ECI) for exchangeable CA/DRM solutions.

The present document is part 4 of a multi-part deliverable covering the Virtual Machine for the Embedded Common Interface for exchangeable CA/DRM solutions specification, as identified below:

Part 1: "Architecture, Definitions and Overview";

Part 2: "Use cases and requirements";

Part 3: "CA/DRM Container, Loader, Interfaces, Revocation";

Part 4: "The Virtual Machine";

Part 5: "The Advanced Security System";

Part 6: "Trust Environment".

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Introduction

The present document describes the concept of a Virtual Machine that executes in a Sandbox and offers a range of instructions and System Call functions. The VM is designed to work in a variety of environments. It interoperates with other applications that exist on the same machine using well-defined interfaces and provides a combination of support for its own instruction set and a modular mechanism for the execution of elements written in the native code of the **ECI Host** CPU and interacting with the hardware and other elements of the **ECI Host** environment. This provides the VM with the means to execute readily renewable code that can provide a wide range of potential secure applications, including the implementation of CA/DRM clients.

1 Scope

The present document specifies a Virtual Machine which is intended for inclusion in the implementation of digital television receivers and Set Top Boxes, and which is able to provide a secured environment for executing Conditional Access kernel or Digital Rights Management client applications. The intention is to provide a uniform execution environment in which such clients can operate in the knowledge that minimum **ECI Host** performance requirements are met, that a standard API is provided to be used for retrieval of essential security data from content (i.e. encapsulated with content) or via external networks (e.g. the Internet) and where resources can be accessed from the **ECI Host** environment in a standardized way.

The presence and use of the VM allows to exchange CA/DRM clients at will and to support multiple simultaneous instances of such clients in **ECI Hosts** so that users and operators are not tied in to a particular content protection provider and that they can use security solutions of different types to suit differing content types. For **Content Protection system** providers, it ensures the availability of a known execution platform that does not require specific integration with any and every vendor of **ECI Host** devices.

2 References

2.1 Normative references

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

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

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

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

[1] ETSI GS ECI 001-3: "Embedded Common Interface (ECI) for exchangeable CA/DRM solutions; Part 3: CA/DRM Container, Loader, Interfaces, Revocation".

[2] "Tool Interface Standard (TIS) Executable and Linking Format (ELF) Specification, version 1.2", TIS Committee, 1995.

NOTE: Available at https://refspecs.linuxfoundation.org/elf/elf.pdf.

2.2 Informative references

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

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

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

[i.1] ISO/IEC 9899: "Information technology -- Programming Languages -- C", ISO/IEC JTC1/SC22 WG14.

[i.2] ETSI GR ECI 004: "Embedded Common Interface (ECI) for exchangeable CA/DRM solutions; Guidelines for the implementation of ECI".

3 Definitions and abbreviations

3.1 Definitions

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

bytecode: code of **ECI Client** (typically comprising a Conditional Access kernel or Digital Rights Management client) that is executed by the VM

content protection system: system that uses cryptographic techniques to manage access to digital content

NOTE: Typically, a **content protection system** is either a conditional access system or a digital rights management system.

Customer Premises Equipment (CPE): customer device that provides **ECI** specified decryption and encryption functions

ECI (Embedded CI): architecture and the system specified in the ETSI ISG "Embedded CI", which allows the development and implementation of software-based swappable **ECI Clients** in customer premises equipment (**CPE**) and thus provides interoperability of **CPE** devices with respect to **ECI**

ECI Client (Embedded CI Client): implementation of a CA/DRM client which is compliant with the ECI specifications

ECI Host: hardware and software system of a **CPE**, which covers **ECI** related functionalities and has interfaces to an **ECI Client**

ecosystem: content and system environment in which the Virtual Machine described in the present document exists

NOTE: It takes into account the wider perspective of content preparation, delivery, authorization, etc. and is not limited to a specific device or implementation.

interface specification: wrapper document that describes the extension, restrictions or any other modifications to the present document that are required to meet the specific needs of a wider **ecosystem** in which the VM is required to operate

native code: programmatic code written in the native executable instruction set of the **ECI Host** processor

sandbox: application execution environment limiting application access to only those resources defined by the **sandbox** API

VM Instance: instantiation of VM established by an **ECI Host** that appears to an **ECI Client** as an execution environment to run in

3.2 Abbreviations

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

API	Application Programming Interface
CA	Conditional Access
CI	Common Interface
CP	Content Protection
CPU	Central Processing Unit
DRM	Digital Rights Management
ELF	Executable and Linkable Format
EPG	Electronic Programme Guide
ID	Identification/Identity/Identifier
OS	Operating System
PC	Program Counter
POSIX	Portable Operating System Interface
RISC	Reduced Instruction Set Computer
VM	Virtual Machine

4 Conceptual principles

4.1 The Virtual Machine as a CPU

In essence, the Virtual Machine (VM) comprises a virtual CPU with its own code and data memory and a set of system interfaces that provide access to hardware features of the **ECI Host** machine. The emulated CPU executes code in the manner of a virtual 32-bit CPU, and the code it executes is called **bytecode** in the present document. Since the VM is a simulation of a general purpose RISC processor it is able to execute a variety of applications.

4.2 Characteristics of the Virtual Machine

The VM shall provide a single-process, single-threaded environment.

The interface to the **ECI Host** hardware and other functions is provided in the form of a standard library of calls, termed SYSCALLs. The SYSCALL instruction is one of the customized instructions of the VM and it is generally executed after preparing the parameters required by the library routine (i.e. passed in "registers" of the VM).

All interaction between the **ECI Client** and the **ECI Host** is achieved through this operation. No **interrupt** architecture is defined and, once started, the **ECI Client** runs to completion. Therefore, there is no opportunity to invoke calls into the VM. Whilst restricting flexibility to a certain extent, this is outweighed by the enhanced control of the VM execution (ensuring robustness of operation), the avoidance of race conditions, interference with time-critical operations, etc.

As a consequence, the only means of passing data or messages to the **ECI Client** executing in the VM is on the basis of requests issued by the **ECI Client** by invoking the appropriate SYSCALLs.

4.3 Isolation of individual **ECI Clients**

The **ECI Client** executes in a Virtual Machine, which exists as an application running in the firmware of the **ECI Host**. It shall be possible to invoke multiple instances of the Virtual Machine, each potentially running a different **ECI Client**. This places three fundamental requirements on the **ECI Host** operating environment:

1) The Operating System shall allocate sufficient resource to each **VM Instance** such that the performance requirements laid out in [i.2] are met by all instances running simultaneously.

2) The libraries defined in clause 6 and annex C shall be fully re-entrant or implemented separately for each instance of the VM.

3) The Operating System and VM shall ensure no information can be exchanged between running **ECI Clients** and the outside world, including other **ECI Clients** by means other than those explicitly specified for such purpose as part of the SYSCALL interface. This among others implies that all memory mapped into the data space of a **VM Instance** is wiped from its previous content beforehand, and any attempts to use exceptional conditions in the VM to trigger unspecified behaviour shall be prevented. This also implies that there is no means for an **ECI Client** to change its **bytecode**. It specifically implies that the **ECI Host** and VM shall make all required checks to prevent an **ECI Client** inducing unintended behaviour in the **ECI Host** or VM implementations that may for instance lead directly or indirectly lead to the **ECI Client** being able to manipulate (hack) the **ECI Host**.

4.4 Specifying the Virtual Machine

In subsequent clauses of the present document, the following are explicitly detailed regarding the VM itself:

1) The technical architecture of the VM.

2) The instruction set of the VM.

3) The **ECI Host** interface.