

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

J.1013

(04/2020)

SERIES J: CABLE NETWORKS AND TRANSMISSION
OF TELEVISION, SOUND PROGRAMME AND OTHER
MULTIMEDIA SIGNALS

Conditional access and protection – Exchangeable
embedded conditional access and digital rights
management solutions

Embedded common interface for exchangeable CA/DRM solutions; The virtual machine

Recommendation ITU-T J.1013

Recommendation ITU-T J.1013

Embedded common interface for exchangeable CA/DRM solutions; The virtual machine

Summary

Recommendation ITU-T J.1013 is part of a multi-part deliverable covering the virtual machine for the embedded common interface (ECI) for exchangeable conditional access/digital rights management (CA/DRM) solutions specification.

This ITU-T Recommendation is a transposition of the ETSI standard ETSI GS ECI 001-4 and is a result of a collaboration between ITU-T SG9 and ETSI ISG ECI. A minor modification was done in clause 7.3.7.1.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T J.1013	2020-04-23	9	11.1002/1000/13574

Keywords

CA, DRM, swapping.

* To access the Recommendation, type the URL <http://handle.itu.int/> in the address field of your web browser, followed by the Recommendation's unique ID. For example, <http://handle.itu.int/11.1002/1000/11830-en>.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at <http://www.itu.int/ITU-T/ipr/>.

© ITU 2020

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

Table of Contents

	Page
1 Scope.....	1
2 References.....	1
3 Definitions	1
3.1 Terms defined elsewhere	1
3.2 Terms defined in this Recommendation.....	1
4 Abbreviations and acronyms	2
5 Conventions	2
6 Conceptual principles	3
6.1 The virtual machine as a central processing unit.....	3
6.2 Characteristics of the virtual machine	3
6.3 Isolation of individual ECI Clients.....	3
6.4 Specifying the virtual machine	4
6.5 ECI Client loader.....	4
7 The virtual machine	4
7.1 Execution environment.....	4
7.2 Virtual machine architecture	5
7.3 Virtual machine instruction set.....	9
8 Interface between the ECI Client and the ECI Host	14
8.1 General principles.....	14
8.2 Error value	15
8.3 SYS_EXIT.....	15
8.4 SYS_PUTMSG.....	15
8.5 SYS_GETMSG	16
8.6 SYS_HEAPSIZE	16
8.7 SYS_STACKSIZE	16
8.8 SYS_SYNCCALL.....	16
8.9 SYS_CLIB.....	17
9 Bytecode lifecycle	17
9.1 Introduction	17
9.2 Loading a new ECI Client into the VM.....	17
9.3 Initialization of the VM	18
9.4 The central run loop.....	18
Annex A – VM system resources	19
Annex B – Op codes for the VM	20
Annex C – Standard C library routines.....	24
C.1 Introduction	24
C.2 memmove	24

	Page
C.3 strcpy	24
C.4 strncpy	24
C.5 strcat	25
C.6 strncat	25
C.7 memcmp	25
C.8 strcmp	25
C.9 strncmp	25
C.10 memchr	26
C.11 strchr	26
C.12 strcspn.....	26
C.13 strpbrk.....	26
C.14 strrchr.....	26
C.15 strspn.....	27
C.16 strstr	27
C.17 memset.....	27
Annex D – ECI Client file format.....	28
Appendix I – Areas for further development.....	29
Bibliography.....	31

Introduction

This ITU-T Recommendation¹ is a transposition of the ETSI standard [b-ETSI GS ECI 001-4] and is a result of a collaboration between ITU-T SG9 and ETSI ISG ECI. A minor modification was done in clause 7.3.7.1.

The objective of this Recommendation is to facilitate interoperability and competition in electronic communications services and, in particular, in the market for broadcast and audio-visual devices. However other technologies are available and may also be appropriate and beneficial depending on the circumstances in Member States.

This Recommendation describes the concept of a virtual machine (VM) 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 and interoperates with other applications that exist on the same machine using well-defined interfaces. It provides a combination of support for its own instruction set and a modular mechanism for the execution of elements written in the **Native Code**² of the **ECI Host** Central Processing Unit (CPU) and interacts with the hardware and other elements of the **ECI Host** environment. This provides the VM with the means to execute a readily renewable code that can provide a wide range of potential secure applications, including the implementation of CA/DRM clients.

¹ Several areas for further development have been identified in Appendix I.

² The use of boldface in the text of this Recommendation indicates terms with definitions specific to the context of the embedded common interface that may differ from common use.

Recommendation ITU-T J.1013

Embedded common interface for exchangeable CA/DRM solutions; The virtual machine

1 Scope

This Recommendation specifies a virtual machine that 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. Refer also to [b-ITU-T J.1010] and [b-ITU-T J.1011].

The presence and use of the VM allows for the exchange of CA/DRM clients at will and for the support of multiple simultaneous instances of such clients in **ECI Hosts**. This ensures that users and operators are not bound to a particular content protection (CP) provider and facilitate the use of different types of security solutions to suit various content types. For providers of content protection systems, 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

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

- | | |
|---------------------|--|
| [ITU-T J.1012] | Recommendation ITU-T J.1012 (2020), <i>Embedded common interface for exchangeable CA/DRM solutions; CA/DRM container, loader, interfaces, revocation</i> . |
| [ETSI GS ECI 001-4] | ETSI GS ECI 001-4 V1.1.1 (2017), <i>Embedded Common Interface (ECI) for exchangeable CA/DRM solutions; Part 4: The Virtual Machine</i> . |

3 Definitions

3.1 Terms defined elsewhere

None.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 bytecode: Code of **ECI Client** (typically comprising a conditional access kernel or digital rights management client) that is executed by the virtual machine (VM).

3.2.2 customer premises equipment (CPE): A customer device that provides embedded common interface (**ECI**) specified decryption and encryption functions.

3.2.3 ECI (Embedded CI): The 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 the interoperability of **CPE** devices with respect to **ECI**.

3.2.4 ECI Client (Embedded CI Client): The implementation of a conditional access/digital rights management (CA/DRM) client which is compliant with the embedded common interface (**ECI**) specifications.

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

3.2.6 Native Code: Programmatic code written in the native executable instruction set of the **ECI Host** processor.

3.2.7 VM Instance: The instantiation of a virtual machine (VM) established by an **ECI Host** that appears to an **ECI Client** as an execution environment in which to operate.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

API	Application Programming Interface
CA	Conditional Access
CAS	Conditional Access System
CI	Common Interface
CP	Content Protection
CPE	Customer Premises Equipment
CPU	Central Processing Unit
DRM	Digital Rights Management
ECI	Embedded Common Interface
ECP	Enhanced Content Protection
ELF	Executable and Linkable Format
EPG	Electronic Programme Guide
ID	Identification/Identity/Identifier
OS	Operating System
OTT	Over The Top
PC	Program Counter
POSIX	Portable Operating System Interface
RISC	Reduced Instruction Set Computer
VM	Virtual Machine

5 Conventions

The use of terms in bold and starting with capital letters in this Recommendation indicates that those terms are defined with an ECI specific meaning that may deviate from the common use of those terms.

6 Conceptual principles

6.1 The virtual machine as a central processing unit

In essence, the virtual machine (VM) comprises a virtual central processing unit (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 in this Recommendation the code it executes is called **Bytecode**. Since the VM is a simulation of a general-purpose reduced instruction set computer (RISC) processor, it is able to execute a variety of applications.

6.2 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.

Consequently, 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.

6.3 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) Isolation of individual ECI Clients – The Operating System (OS) shall allocate sufficient resources to each VM Instance such that the performance requirements are met by all instances running simultaneously; proposed values for performance requirements are laid out in [b-ITU-T J-Suppl.7].
- 2) The libraries defined in clause 8 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 that 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 the VM shall make all required checks to prevent an **ECI Client** from inducing unintended behaviour in the **ECI Host** or VM implementations that may, for instance, lead directly or indirectly to the **ECI Client** being able to manipulate (hack) the **ECI Host**.

6.4 Specifying the virtual machine

In subsequent clauses of this Recommendation, 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.

6.5 ECI Client loader

In order to execute the **ECI Client**, the **Bytecode** shall first be loaded into the code space of the VM memory and the data space initialized. Clause 9 addresses some specific aspects of the format of the **ECI Client** container and initialization of the VM.

7 The virtual machine

7.1 Execution environment

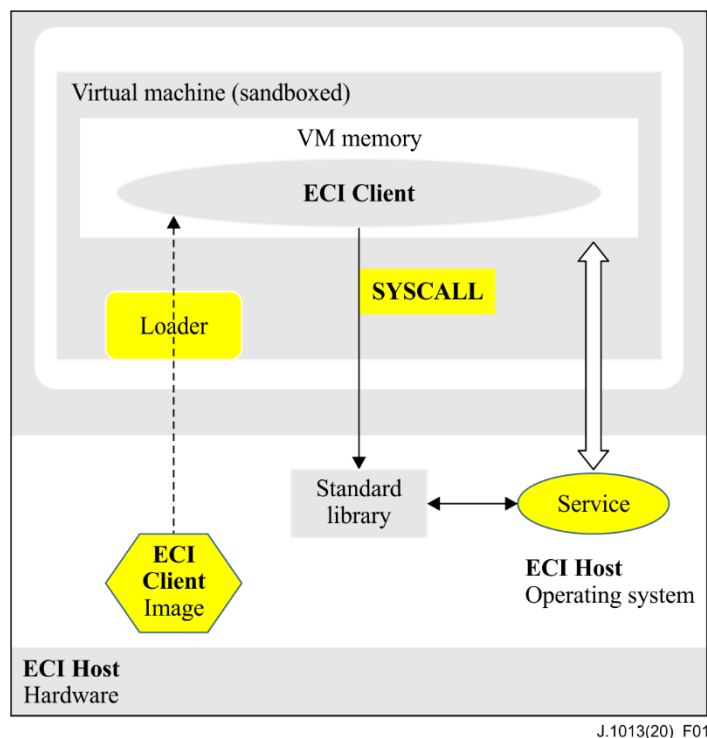


Figure 1 – VM Host environment

As depicted in Figure 1, the VM shall be executed in a sandboxed environment that ensures isolation from the **ECI Host's** operating system, other virtual machine instances and any other applications executing in the **ECI Host**.

The VM comprises a native application of the **ECI Host**, with associated memory, and interface library and a loader for installing the **Bytecode** forming an **ECI Client**. The interface library provides the **ECI Client** with access to features of the **ECI Host** operating system and hardware as well as to other applications that may be executing in the **ECI Host** and with which the **ECI Client** may need to interact. A typical example would be an interaction with an Electronic Programme Guide (EPG) application that would require authorization status for specific content for display to the user.