ETSI GR ECI 004 V1.1.1 (2018-03)

Embedded Common Interface (ECI) for exchangeable CA/DRM solutions; Guidelines for the implementation of ECI

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

DGR/ECI-004

Keywords

CA, DRM, security

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

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables 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 Report (GR) has been produced by ETSI Industry Specification Group (ISG) Embedded Common Interface (ECI) for exchangeable CA/DRM solutions.

The present document on Guidelines for the implementation of **ECI** complements ETSI GS ECI 001 (all parts), [i.1] to [i.7] for the Embedded Common Interface for exchangeable CA/DRM solutions Group Specification (GS).

NOTE: The use of terms in bold and starting with capital characters in the present document shows that those terms are defined with an **ECI** specific meaning, which may deviate from the common use of those terms.

Modal verbs terminology

In the present document "**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

Service and content protection realized by Conditional Access (CA) and Digital Rights Management (DRM) are essential in the rapidly developing area of digital Broadcast and Broadband services. This includes the distribution of HD and UHD content to various types of customer premises equipment (CPE) in order to protect business models of content owners and **Service** providers, including Broadcasters and PayTV operators.

Existing CA/DRM technologies limit the freedom of many players in digital multimedia content markets. Due to technological progress, innovative, software-based CA/DRM solutions become feasible. Maximizing interoperability while maintaining a high level of security, these solutions promise to meet upcoming demands in the market, allow for new businesses, and broaden consumer choice with respect to content consumption via broadcast and broadband connections.

An **ECI Ecosystem,** compliant with ETSI GS ECI 001 (all parts) [i.1] to [i.7], addresses important attributes, such as enabling a high level of system security, flexibility and scalability due to software-based implementation, as well as exchangeability fostering a future-proof solution and enabling innovation. Further aspects are applicability to content distributed via different types of networks, including classical digital broadcasting, IPTV and OTT **Service**s. The **ECI** system specification of an open eco-system, fostering market development, provides the basis for exchangeability of CA and DRM systems in **CPE**s, at lowest possible costs for the consumers and with minimal restrictions for CA or DRM vendors to develop their target products for the PayTV market.

Complementing ETSI GS ECI 001 (all parts) [i.1] to [i.7], the present document gives further guidance and addresses beside necessary performance requirements a number of use cases and scenarios, which on one side make use of the **ECI Ecosystem** and on the other extend its possibilities.

1 Scope

The present document serves as a guidance document for the **ECI Ecosystem** as specified in ETSI GS ECI 001 (all parts) [i.1] to [i.7], including specification of the architecture of the **ECI** system as defined in ETSI GS ECI 001-1 [i.1] and specification of the requirements as defined in **ECI** Group Specification ETSI GS ECI 001-2 [i.2]. A major advantage and innovation of the **ECI Ecosystem**, compared with currently deployed systems, is a fully software-based client container architecture, backed by a standardized advanced security hardware and secure software functionality for the loading and exchanging of CA/DRM client systems in **CPE**s. **ECI** compliant solutions do not require any detachable hardware modules in **CPE**s. Software containers provide a secure ("Sandbox") environment for either CA or DRM kernels, hereafter named as **ECI Client**s, together with their individual **Virtual Machine Instance**s. The download process is embedded in a secure and trusted environment, providing a trust hierarchy for installation and exchange of **ECI Host** and **ECI Client**s and thus enabling an efficient protection against integrity- and substitution attacks. For this reason, the **ECI Ecosystem** integrates an advanced security mechanism.

The present document covers implementation guidance details in the following clauses:

- Clause 4 contains performance requirements and parameters for the **ECI Host,** the **ECI Client**, the Virtual Machine and for the **Advanced Security System**.
- Clause 5 deals with use cases and applications based on the **ECI Ecosystem**, which either complement the **ECI** multi-part Group Specification or address given scenarios in more detail.

The present document has the objective to make available to **ECI** implementers as much as possible of the common understanding captured during the work of the ISG **ECI** developing the **ECI** specification series [i.1] to [i.8]. The present document was prepared with the intention to provide know-how complementary to the content of the **ECI** specifications [i.1] to [i.8] itself and about the environment in which an **ECI Ecosystem** will be operated. It is planned to extend this guideline by further guidance and background information gained during the implementation and operation of **ECI** compliant ecosystems.

2 References

2.1 Normative references

Normative references are not applicable in the present document.

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] ETSI GS ECI 001-1 (V1.2.1): "Embedded Common Interface (ECI) for exchangeable CA/DRM solutions; Part 1: Architecture, Definitions and Overview".

[i.2] ETSI GS ECI 001-2 (V1.2.1): "Embedded Common Interface (ECI) for exchangeable CA/DRM solutions; Part 2: Use cases and requirements".

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

[i.4] ETSI GS ECI 001-4: "Embedded Common Interface (ECI) for exchangeable CA/DRM solutions; Part 4: The Virtual Machine".

[i.5] ETSI GS ECI 001-5-1: "Embedded Common Interface (ECI) for exchangeable CA/DRM solutions; Part 5: The Advanced Security System; Sub-part 1: ECI specific functionalities".

[i.6] ETSI GS ECI 001-5-2: "Embedded Common Interface (ECI) for exchangeable CA/DRM solutions; Part 5: The Advanced Security System; Sub-part 2: Key Ladder Block".

[i.7] ETSI GS ECI 001-6: "Embedded Common Interface (ECI) for exchangeable CA/DRM solutions; Part 6: Trust Environment".

[i.8] ETSI GS ECI 002: "Embedded Common Interface (ECI) for exchangeable CA/DRM solutions; System Validation".

[i.9] ISO/IEC 23001-12:2015: "Information technology -- MPEG systems technologies -- Part 12: Sample Variants in the ISO base media file format".

3 Definitions and abbreviations

3.1 Definitions

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

Advanced Security System (AS System): function of an **ECI** compliant **CPE**, which provides enhanced security functions (hardware and software) for an **ECI Client**

NOTE: The details are specified in [i.5] and [i.6].

AS slot: resources of the Advanced Security block provided exclusively to an **ECI Client** by the **ECI Host**

AS slot session: resources and computing in an **AS slot** related to the de-cryption or re-encryption of a content element

Certificate: data structure as defined in clause 5 of [i.3] with a complementary secure digital signature that identifies an **Entity**

NOTE: The holder of the secret key of the signature attests to the correctness of the data - authenticates it - by signing it with its secret key. Its public key can be used to verify the data.

CPE: ECI compliant customer premises equipment

NOTE: A **CPE** can be a stationary device (e.g. SetTopBox or iDTV) or any kind of mobile or portable device, which is able to process digital media content within an **ECI Ecosystem**.

CPE Manufacturer: company that manufactures **ECI** compliant **CPE**s

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 Client**s in customer premises equipment (**CPE**) and thus provides interoperability of **CPE**s with respect to **ECI**

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

NOTE: It is the software module in a **CPE** which provides all means to receive, in a protected manner, and to control execution of a consumer's entitlements and rights concerning the content that is distributed by a content distributor or **Operator**. It also receives the conditions under which a right or an entitlement can be used by the consumer, and the keys to decrypt the various messages and content.

ECI Client Image: file with software as VM code, and initialization data required by the **ECI Client Loader**

ECI Client Loader: software module part of the **ECI Host** which allows downloading, verifying and installing new **ECI Client Image**s in an **ECI Host**

ECI Ecosystem: commercial operation consisting of a **TA** and several platforms and **ECI** compliant **CPE**s in the field

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

NOTE: The **ECI Host** is one part of the **CPE** firmware.

Entity, (Entities): organization (e.g. **Manufacturer, Operator** or **Security Vendor**) or real world item (e.g. **ECI Host**, **Platform Operation** or **ECI Client**) identified by a unique ID in an **ECI Ecosystem**

Manufacturer: Entity which develops and sells **CPE**s, which accommodate an implementation of the **ECI** system and allow **ECI Host**s and **ECI Client**s to be installed per software download

Media Handle: reference to a single program decryption or re-encryption processing setup between an **ECI Client** and an **ECI Host**

Operator: organization that provides **Platform Operation**s and is enlisted with the **ECI TA** for signing the **ECI Ecosystem**

NOTE: An **Operator** may operate multiple **Platform Operation**s.

Platform Operation: specific instance of a technical **Service** delivery operation having a single **ECI** identity with respect to security

Request: message from a sender to a receiver asking for certain information or to perform a certain operation within an **ECI Ecosystem**, which is specified in the data fields of that request

NOTE: More details are given in clause 9.2.3 of [i.3].

Response: message within an **ECI Ecosystem** answering a **Request**

NOTE: More details are given in clause 9.2.3 of [i.3].

Revocation List (RL): list of **Certificate**s that have been revoked and therefore should no longer be used

Root: public key or **Certificate** containing a public key that serves as the basis for authenticating a chain of **Certificate**s

Root Certificate: trusted **Certificate** that is the single origin of a chain of **Certificate**s

Secure Authenticated Channel (SAC): communication path (channel) that has been established between two **Entities** where the **Entities** have securely identified themselves to each other (authenticated) and agreed on an encryption of data transferred between them (secure)

Security Vendor: company providing **ECI** security systems including **ECI Client**s for **Operator**s of **ECI Platform Operation**s

Service: content that is provided by a **Platform Operation**

NOTE: In the context of **ECI** only protected content is considered.

Trust Authority (TA): an organization governing all rules and regulations that apply to a certain implementation of **ECI** and targeting at a certain market

NOTE: The Trust Authority has to be a legal **Entity** to be able to achieve legal claims. The Trust Authority needs to be impartial to all players in the **ECI Ecosystem** it is governing.

User: person who operates an **ECI** compliant **CPE**

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:

AES	Advanced Encryption Standard
API	Application Programming Interface

AS	Advanced Security
BAT	Bouquet Association Table
CA	Conditional Access
CA/DRM	Conditional Access/Digital Rights Management
CAT	Conditional Access Table
CI	Common Interface
CPE	Customer Premises Equipment
CPS	Certificate Processing Subsystem
CPU	Central Processing Unit
CW	Control Word
DMIPS	Dhrystone Million Instructions Per Second
DRM	Digital Rights Management
DVB	Digital Video Broadcasting
ECM	Entitlement Control Message
EIT	Event Information Table
EITpf	EIT related to the present and the following content event
GS	Group Specification
HD	High Definition (Television)
HDD	Hard Disk Drive
HTTP	Hypertext Transfer Protocol
HTTP(S)	Hypertext Transfer Protocol Secure
iDTV	integrated Digital Television
IP	Internet Protocol
IPTV	TV using the Internet Protocol (IP)
MPEG	Motion Picture Experts Group
NIT	Network Information Table
NV	Non-Volatile (memory)
OTT	Over The Top (over the open Internet)
PAT	Program Association Table
PayTV	Pay Television
PID	MPEG Packet IDentifier
PMT	Program Map Table
PVR	Personal Video Recorder
RL	Revocation List
SAC	Secure Authenticated Channel
SD	Standard Definition (Television)
SDT	Service Description Table
SI	Service Information
SOC	System-On-a-Chip
SW	Software
TA	Trust Authority
TCP	Transmission Control Protocol
TECM	Time (delay) ECM
UHD	Ultra High Definition (Television)
URI	Usage Rights Information
VM	Virtual Machine

4 Guidelines for the implementation of an ECI compliant CPE

4.1 Introduction

Performance of CPE controllers is growing especially due to enhanced silicon technologies. Therefore performance figures for the **ECI**-implementation in **CPE**s have been defined separately in the present document, allowing **ECI** following any technological development in an easy way by updating or extending the present document.

4.2 The relevance of the ECI Implementation Guidelines for ECI Eco-Systems

An **ECI Ecosystem** is based on the **ECI** specifications, allowing its framework to be further developed based on commercial and business related considerations. The guidelines for the implementation of an **ECI** compliant **CPE** can be part of that framework. However, most likely adaptations and extensions will be necessary. This is usually the case in an open system and does not cause any conflict with the **ECI** philosophy.

4.3 Performance requirements for ECI Clients and ECI Hosts

4.3.1 Introduction

In this clause performance figure-related specification elements defined in part 3 of the **ECI** specifications [i.3] are discussed.

4.3.2 Execution time

The **ECI Host** should ensure that the **ECI Client** can be executed (when ready to do so) with a maximum interval of 25 ms.

The VM performance should be at least as high as 5 DhrystoneMIPS (refer to annex A). The code and data memories should not be paged and be permanently allocated in main memory.

The scheduling of **ECI Client**s can depend on collaborative sharing of a single CPU resource. **ECI Client**s should limit the number of instructions executed between thread blocking events to 12 ms of 20 DhrystoneMIPS equivalent. In case this limit is exceeded the **ECI Host** may reset the **ECI Client** and/or mark it as possibly dysfunctional and prevent it from being loaded.

4.3.3 NV file storage

The **ECI Client** should have access to NV file storage resource provided by the **ECI Host** through the file system API (see clause 9.4.5 of [i.3]). The maximum amount of NV storage available to an **ECI Client** should be limited to 128 kBytes.

4.3.4 Minimum storage resources provided by the ECI Host for storage of an ECI Client

The **ECI Host** should be able to store at minimum two **ECI Client Image**s. The maximum size of a client image is given by the maximum segment sizes for code and initialized data as defined in clause 4.4.3.

4.3.5 Minimum storage resources provided by the ECI Host to an ECI Client for data storage

The resources to be provided by the **ECI Host** to an **ECI Client** for data storage are defined in clause 4.4.3.

4.3.6 Resources for storage of Root Certificate

ECI Hosts should store at minimum 3 **Root Certificate** extracts (at least the public key and the associated version) in non-modifiable memory (ROM/OTP) managed by the CPS.

4.3.7 Minimum repetition rate for acquisition of different DVB SI tables

The **ECI Host** should at least update DVB SI table data with the following repetition rate:

- NIT, SDT, BAT: at least every 30 minutes.

EITpf:	at least every 2 minutes.
PAT, CAT, PMT:	at least every 20 seconds.

4.3.8 Performance requirements for Responsiveness Monitoring

The **ECI Host** should use a timeout of 5 seconds on the acceptance of a new message by the **ECI Client** as per [i.3], clause 9.2.6.

4.3.9 Performance requirements for the ECI system software update policies

ECI Hosts should attempt to check for updates at least every 30 minutes during power-on, provided network access resources are available, and every 6-hours during standby and, if required, perform a download of new items using the network **Service** provided by each **Platform Operation**. In case this is prevented (no network access, no power or other temporary reasons, etc.) the **User** should be warned after 14 days and asked to check the network access. Update data items should get higher priority than any user request to access a certain service, in case such a user-requested access would prevent accessing update services. Alternatively, the **ECI Host** can offer the **User** the choice to suspend loading of the affected **ECI Client**.

NOTE: In case update checking prevents an **ECI Service** from rendering services to a **User**, there has to be a suitable warning on the screen.

4.3.10 Performance requirements for the TCP server

For TCP server mode the **ECI Host** should be able to queue at least 10 incoming connection requests.

The TCP server should be able to handle at least 5 incoming connection requests in parallel.

4.3.11 Performance requirements for the HTTP(S) server

The **ECI Host** should support at minimum 3 simultaneous outstanding HTTP requests per **ECI Client**. In case of multiple simultaneous HTTP requests the **ECI Host** may queue these. The maximum file size to be loaded is 1 Mbyte. Each outstanding HTTP request can diminish the number of available IP sockets for an **ECI Client** by one.

The **ECI Host** should support at least 3 redirects to complete a HTTP(S) request.

4.3.12 Performance requirements for timers

The **ECI Host** should support a minimum number of outstanding timers for each **ECI Client** of 50.

4.3.13 Performance requirements for power management

The **ECI Host** should repeat sending messages for requesting a change of the power status every 10 seconds in case the **ECI Client Response** is negative (not ready). The **ECI Host** is not obliged to refrain from going to Standby state after more than 30 seconds.

Time accurateness of the wakeup time implementation is allowed to be ±2 minutes in case an **ECI Host** is not impeded in waking up from standby and starting an **ECI Client.**

4.3.14 Buffering requirements for the reqEncrTsData Message

For the reqEncrTsData message the **ECI Host** should buffer the data of the message appropriately (as associated data to the content) and respond to the next within 1 second.

4.3.15 Timing requirements for the reqEncrTsEcm Message

For the reqEncrTsEcm the **ECI Host** should insert the ECM in the Transport Stream within 400 ms of receiving the message. The ECM should be repeated at a reasonable interval (between 200 and 400 ms content time). The ECM PID should be a free PID and is generated by the **ECI Host**.

4.3.16 Timing requirements for the reqEncrMsgRecv Message

For the reqEncrMsgRecv message the **ECI Host** should buffer the data of the message appropriately (as associated data to the content) and respond to the next within 10 seconds.

4.3.17 Buffering requirements for the reqParAuthCid Message

The **ECI Client** should maintain a non-volatile record of content identifications that have been authenticated with this function. It may discard the oldest records and records which are no longer valid in the future in case it lacks storage space. It should be able to maintain a record of at least 2 000 content identifications.

4.3.18 Timing requirements for the reqParAuthChk and the reqParAuthDel Message

The **ECI Client** should use a timeout value for requesting parental authentication for the **reqParAuthChk** or **reqParAuthDel** messages (see clause 9.8.2.10 in [i.3]) that will terminate within a reasonable period if there is no person present or willing to perform the authentication. The value for the timeout should be higher than 15 seconds and smaller than 2 minutes.

4.3.19 Constraints for the ECI Application container directory structure and files

The application container directory structure and files should be no more than 5 levels deep.

4.3.20 Constraints for the ECI Application container size

The maximum decompressed size of the application container is 8 MBytes, counting a directory as 4 kBytes and rounding each file up to a 4 kBytes multiple.

4.3.21 Maximum time to cancel a Media Handle Session

The maximum time required by an **ECI Host** for cancelling a **Media Handle** Session is 1 minute.

4.4 Performance requirements for the ECI Virtual Machine

4.4.1 Introduction

In this clause specification elements related to performance requirements, defined in part 4 of the **ECI** specifications [i.4], are discussed.

4.4.2 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 should be possible to invoke multiple instances of the Virtual Machine, each potentially running a different **ECI Client**. This places a fundamental requirement on the **ECI Host** operating environment:

- The Operating System should allocate sufficient resource to each **VM Instance** such that the performance requirements laid out in the present document are met by all instances running simultaneously.

4.4.3 VM System Resources

ECI Hosts need to guarantee the availability of minimal storage resources for code, data and stack in the **VM Instance** of each **ECI Client**. The resources are defined in [i.4] annex A as C-language macro's. The following resources should be available to the **ECI Client**:

- Minimum byte code size: 2 Mbytes
  `#define CODE_SIZE (0x200000)`

- Minimum number of registers in the register file: 256 × 16 = 4 096
  `#define REGISTER_FILE_SIZE (0x1000)`

NOTE: This also defines the size of the control stack as 256: i.e. a 256 deep nesting of calls is supported.

- Minimum amount of **ECI Client** data address space available: 1 Mbyte
  `#define ADDRESSABLE_DATA_SIZE (0x100000)`

- Maximum amount of **ECI Client** address space reserved for **ECI Host** applications: 128 kbytes
  `#define VM_RESERVED_SIZE (0x020000)`

- The total amount of data memory available to **ECI Client**s has to be:
  `ADDRESSABLE_DATA_SIZE - VM_RESERVED_SIZE`
  This includes any initialized data associated with the **ECI Client**.

- The location of the data memory is defined by a start address. The value is 16 Mbytes:
  `#define DATA_BASE_ADDRESS (0x1000000)`

- For convenience of the **ECI Client** the default stack size is set at 16 kbytes at initialization:
  `#define DEFAULT_STACK_SIZE (0x4000)`

4.5 Performance requirements for the Advanced Security System

4.5.1 Introduction

In this clause performance-requirements-related specification elements defined in ETSI GS 001-5-1 [i.5] of the **ECI** specifications are discussed.

4.5.2 Discrepancy between encryption parameters and imported Content Properties

The content processing system should not permit a discrepancy between the encryption parameters and the imported Content Properties for more than 3 seconds.

4.5.3 Time constraints for the performance of symmetrical cryptography functions

Functions invoking one or multiple symmetrical cryptography operations should be performed by the **AS system** on the following basis: each **AS slot session** should be able to perform one function per 100 ms.

4.5.4 Time constraints for the performance of asymmetrical cryptography functions

Functions invoking asymmetrical cryptography operations should be performed by the **AS System** on the following basis: each **AS Slot** should be able to perform one function at a time; an encryption or signature validation operation (operation with a public key) takes a maximum of 50 ms, and a decryption operation (operation with a private key) should take a maximum of 100 ms.

4.5.5 Content property change timing interface convention

The selection of timing parameters is important for the seamless handover of content between **ECI Client**s. Typical values for delay parameters are given as an example:

- TECM: 3 s
- TCASCADE: 2 s
- TDELAY: 0,3 s
- TMAXWARN: 10 s

5 Use cases and scenarios associated with an ECI Ecosystem

5.1 Introduction

The following use cases and scenarios are closely related to the **ECI** architecture represented by the **ECI** Group Specification series in order to give some further guidance and to open up opportunities to expand the possibilities of the **ECI Ecosystem**; refer among other GS to ETSI GS ECI 001-1 [i.1].

5.2 Management of protected content

5.2.1 Introduction

The **ECI** architecture specifies mechanisms for re-encryption, transfer, and streaming of protected content. The following clauses describe some related use cases and scenarios, which might occur.

5.2.2 Local storage of content within a CPE (PVR)

This use case is fully specified in [i.3] and [i.5]. Further dependencies of involved **Entities** can be derived from associated flow diagrams in clause 9 of [i.8] with regard to re-encryption of content.

5.2.3 Replacement of a CPE by a new CPE

Subject of this scenario is the transfer of stored, encrypted content, which has been legally acquired by the customer, to a new **ECI** compliant **CPE** under a trusted environment and in line with the associated usage rights information (URI); refer to [i.2].

Use case: the customer has stored a copy-protected piece of content on the HDD of an **ECI**-compliant **CPE** and intends now to transfer the stored copy protected content including usage rights information to a new **CPE**, which will then replace the original **CPE**. The new **CPE** does not have to be from the same **Manufacturer** as long as it is **ECI** compliant.

5.2.4 Export from primary CPE to secondary ECI compliant CPE

Use case: a DRM vendor produces an **ECI Client** for **ECI**-compliant tablets and/or other **CPE**s, which are suited for communication with a primary **ECI**-compliant **CPE** in a home domain and in order to handover protected content. The DRM vendor solely relies on the **ECI**-specific procedures as specified in [i.3] and [i.5] without any need for direct contractual relations with other CA/DRM vendors.

The **ECI Client** on the tablet/**CPE** communicates with the security instance on the primary **CPE** with regard to allowed streaming and/or export (copy or move) to the secondary **CPE**. In case this is permitted, the tablet/**CPE** receives information about the necessary protection level (usage rights) and may then control the content and the associated security-related functions.

5.2.5 Export from primary CPE to secondary non-ECI compliant CPE

Use case: a DRM vendor produces a client for a non-**ECI**-compliant CPE, which is suited for communication with a primary **ECI**-compliant **CPE** in a home domain and in order to handover protected content. The DRM vendor solely relies on the **ECI**-specific procedures as specified in [i.3] and [i.5] without any need for direct contractual relations with other CA/DRM vendors.

The **ECI Client** in the primary **CPE** opens a data pipe to the client on the secondary CPE (see clause 9.9 in [i.3]) after having established trust with the client of the secondary CPE using proprietary mechanisms. This is only possible if the vendors of both CA/DRM-system generally trust each other and have agreed on the required communication mechanisms. If a transfer of the requested content element to the other CPE is allowed according to the usage rights information, the **ECI Client** of the primary **CPE** will transfer the content element, including a full copy of the related URI data (mapped to the CA/DRM capabilities of the secondary CPE) to the secondary CPE, using proprietary mechanisms. From now onward the secondary CPE is fully responsible for the usage and the security of that content element.

5.3 Implementation of a Secure Authenticated Channel (SAC) between two ECI Clients

In clause 9.9.2 of [i.3] APIs are defined, allowing two **ECI Client**s to communicate. The means to secure this communication are out of scope for **ECI**. However **ECI Client**s may use all API tools provided by the **ECI** specification, including the **Advanced Security System**, specified in [i.6], to establish trust with another **ECI Client**. Specific information on a **SAC** for inter client communication is discussed in clauses 9.3.5 and 9.5.2 of [i.3].

One straightforward mechanism of secure inter-client communication is through the secure provisioning by a server or headend of a shared (symmetrical) key to two clients that may then wish to authenticate each other and possibly even encrypt data exchanged between them each other based on such a key (or keys). Protocols for such **SAC**s are described in literature. The secure provisioning can be based on different mechanisms. Examples are:

1) Use the reqAsComputeAkClient and reqAsClientChalResp messages as defined in [i.3], clause 9.5.2.2. The encrypted messages can be stored in the **ECI Client**'s file system (see [i.3], clause 9.4.5).

2) Use serialized and encrypted client images as defined in [i.3], clauses 7.4 and 7.8.3. The encrypted client image can contain the keys.

Rather than using symmetrical cryptography as a basis, **ECI Client**s may choose asymmetrical cryptography to establish authenticity and (if deemed required) security on the communication channel. The same mechanisms as above may be used to provision an **ECI Client** with a secret or public key. For the provisioning of a public key, encryption between the server or the headend and the **ECI Client** or **ECI Client Loader** is not required in general.

5.4 Mechanism for future update or extension of API messages

Any API defined in **ECI** has its own version number. An **ECI Client** will use the so called **ECI Host** interface discovery resource (see clause 9.4.3 in [i.3]) to ask the **ECI Host** which APIs und which related version numbers are supported. More details are discussed in clause 9.4.3 of [i.3].

The **Trust Authority**, refer to [i.7], of an **ECI Ecosystem** should define its policy concerning which API and which version have to be supported by compliant **ECI Host**s. This mechanism allows the **Trust Authority** to define a clear migration path for the introduction of new or extended features within the deployed basis of **CPE**s. It should be noted that not all compliant **CPE** need to support the latest version of all APIs. As an **ECI Client** negotiates the API versions individually with the **ECI Host** it is installed in, the system is very flexible. However, if a new version with extended API features is available, those features are only usable in case an updated **ECI Host** is available and has been installed by the customer on his **CPE**.

5.5 Mechanism for future extension of content properties

For the content property APIs the versioning mechanism as discussed in clause 5.4 applies as well. The usage rights information (URI) APIs offer three different possibilities to signal URI:

a) The standard URI API allows to signal URI comparable to the today deployed solutions as described in clause 9.8.2.3 of [i.3].

b) The basic URI API is a very interesting new feature of **ECI**, which allows protecting the delivery of URI with the **ECI Advanced Security System**. In case the delivery of the basic URI would be manipulated the decryption of the related content is not possible because the **ECI Client** is no longer able to compute the correct control word (CW); refer to clause 9.8.2.5 of [i.3].

c) The customer URI API allows to deliver platform-specific (non-standardized) URI information. This feature could be used in case it is necessary to deliver complex private content license information to an **ECI** compliant **CPE**; refer to clause 9.8.2.4 of [i.3].

All three URI signalling mechanisms can be updated or extended individually in case new features or business models are emerging in the market.

5.6 Watermarking

The **ECI Ecosystem** supports Watermarking and specifies a Watermarking API according to clause 9.8.2.7 of [i.3].

The marking API permits **ECI Clients** to discover embedded (water) marking systems available through the **ECI Host**, and then engage in a "setup" control dialogue with such systems. The marking systems may be able to engage in a dialogue with only a limited number of **ECI Clients** and may be able to mark only a limited number of **Media Handle** sessions simultaneously.

ECI does not specify any specifics regarding buffering or (possibly extensive) intermediate processing like transcoding or watermarking that may be performed on the decrypted content passing from decryption to encryption resource. Such processes may cause significant delays. **CPE Manufacturer**s may select appropriate implementations causing a time-offset between decryption resource and a connected re-encryption resource. The re-encryption slot and **ECI Client** synchronize with the encryption of content; refer to clause 7.1 in [i.5].

In addition to the watermarking API, **ECI** supports a default **CPE** bound watermarking system that can be activated through the application of the Output Control Vector as defined in [i.3], clause 9.8.2.6.

ECI can accommodate any headend or server based watermarking. **ECI Client**s can be securely provisioned for exclusive decryption of specifically watermarked content.

The **ECI** API supports decoding of the keys for MPEG variants for file-format based decryption [i.9]. For transport stream based decryption an accepted industry format signalling of MPEG variants stream sections and transients is not available at the time of creation of the present document.

5.7 Update mechanism for RL

In addition to the mechanisms specified in [i.3], the following principles for a **Revocation List** update apply:

As a general rule **ECI Host**s should store the **TA Revocation Lists** of all **Certificate**s required to verify the **Entities** that are loaded by the **ECI Host**. **ECI Host**s should replace a stored **Revocation List** for a **Certificate** or item by a newly received **Revocation List** with a later version number; refer to clause 5.3 of [i.3].

5.8 Uninstallation of an ECI Client

While the installation of **ECI Client**s is specified in [i.3] and their related procedures further specified in [i.8], uninstallation of **ECI Client**s is not specifically addressed in [i.3] and subject to implementation details. Some general aspects are given in this clause.

Several cases for the necessity to uninstall **ECI Clients** may occur:

- An **ECI Client** needs to be removed to allow the installation of an updated version of this client
- An **ECI Client** is candidate to be removed due to storage limitations
- An **ECI Client** has been revoked and has to be removed for security reasons and also in order not to block any internal resources of the **CPE**
- An **ECI Client** with harmful code was loaded to the **CPE** and after detection has to be prevented from operation and has to be uninstalled

It is in the responsibility of the **ECI Host** to include appropriate uninstallation mechanisms.

Annex A: General VM computing performance

Considering Dhrystone (DMIPS) as a suitable benchmark shows the following:

1) Dhrystone 2.1 is a synthetic benchmark. A significant part of the performance is defined by c-library functions for string manipulation, which are at raw CPU level performance if executed by the SYS_CLIB syscall. This "taints" Dhrystone to a somewhat unknown extent.

   It also does not measure the overhead of synchronous and asynchronous message passing, though this may have some resemblance to the SYS_CLIB syscall.

Using this benchmark for VM computing performance thus may have some constraints, but may be used for orientation.

2) In order to find an adequate performance level, it is assumed, that the c-lib speed benchmark distortion is small with the result, that 5 DMIPS should be more than sufficient for a client.

Considering a factor of 10 as VM overhead, a calculation for 4 **ECI Client**s results in:

Performance needed: 4 × 5 × 10 = 200 DMIPS of raw CPU; (20 % of a single core processor with reduced instruction set; see below)

Some performance references:

• PC CPU@50MHz (CPU for PCs, mid 90's):	22 DMIPS
• Advanced PC processor of unspecified generation @ 4GHz:	12 000 DMIPS
• Processors with reduced instruction set of today with 1 DMIPS/MHz (single core):	around 1 000 DMIPS for a single 1 GHz core
• Typical broadcast zapper SOC:	600 DMIPS CPU performance

Taking these values into account, it seems realistic to run 2 **ECI Client**s (100 DMIPS) with some effort.

At least any state-of-the-art chip architecture appears suitable for **ECI,** while "legacy" SD settop box chips will not offer the demanded performance.

NOTE: Assumption of 5 DMIPS for hash and AES operations performed in SW on smaller amounts of data seems to be a realistic approach, while operating asymmetrical cryptography in SW would not be recommended.

For benchmarks refer to:

- http://www.roylongbottom.org.uk/android%2064%20bit%20benchmarks.htm
- http://www.roylongbottom.org.uk/dhrystone%20results.htm

Annex B:
Authors & contributors

The following people have contributed to the present document:

Rapporteur:
Christoph Schaaf, Vodafone Kabel Deutschland

Peter Mann, BNetzA (from 02/2017 onwards)

Other contributors:
Marnix Vlot, Vodafone Kabel Deutschland

Annex C: Bibliography

- Klaus Illgner, Christoph Schaaf, Marnix Vlot: "Embedded Common Interface (**ECI**) for Digital Broadcasting Applications: Security and Interoperability combined", Broadband Journal of the SCTE, Vol. 38, No. 3, August 2016.

Annex D:
Change History

Date	Version	Information about changes
February 2018	0.0.1	first version of the document

History

Document history		
V1.1.1	March 2018	Publication