



Categorization Based Content Screening

Approved Version 1.0 – 06 Dec 2011

Open Mobile Alliance
OMA-TS-CBCS-V1_0-20111206-A

Use of this document is subject to all of the terms and conditions of the Use Agreement located at <http://www.openmobilealliance.org/UseAgreement.html>.

Unless this document is clearly designated as an approved specification, this document is a work in process, is not an approved Open Mobile Alliance™ specification, and is subject to revision or removal without notice.

You may use this document or any part of the document for internal or educational purposes only, provided you do not modify, edit or take out of context the information in this document in any manner. Information contained in this document may be used, at your sole risk, for any purposes. You may not use this document in any other manner without the prior written permission of the Open Mobile Alliance. The Open Mobile Alliance authorizes you to copy this document, provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials and that you comply strictly with these terms. This copyright permission does not constitute an endorsement of the products or services. The Open Mobile Alliance assumes no responsibility for errors or omissions in this document.

Each Open Mobile Alliance member has agreed to use reasonable endeavors to inform the Open Mobile Alliance in a timely manner of Essential IPR as it becomes aware that the Essential IPR is related to the prepared or published specification. However, the members do not have an obligation to conduct IPR searches. The declared Essential IPR is publicly available to members and non-members of the Open Mobile Alliance and may be found on the “OMA IPR Declarations” list at <http://www.openmobilealliance.org/ipr.html>. The Open Mobile Alliance has not conducted an independent IPR review of this document and the information contained herein, and makes no representations or warranties regarding third party IPR, including without limitation patents, copyrights or trade secret rights. This document may contain inventions for which you must obtain licenses from third parties before making, using or selling the inventions. Defined terms above are set forth in the schedule to the Open Mobile Alliance Application Form.

NO REPRESENTATIONS OR WARRANTIES (WHETHER EXPRESS OR IMPLIED) ARE MADE BY THE OPEN MOBILE ALLIANCE OR ANY OPEN MOBILE ALLIANCE MEMBER OR ITS AFFILIATES REGARDING ANY OF THE IPR'S REPRESENTED ON THE “OMA IPR DECLARATIONS” LIST, INCLUDING, BUT NOT LIMITED TO THE ACCURACY, COMPLETENESS, VALIDITY OR RELEVANCE OF THE INFORMATION OR WHETHER OR NOT SUCH RIGHTS ARE ESSENTIAL OR NON-ESSENTIAL.

THE OPEN MOBILE ALLIANCE IS NOT LIABLE FOR AND HEREBY DISCLAIMS ANY DIRECT, INDIRECT, PUNITIVE, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR EXEMPLARY DAMAGES ARISING OUT OF OR IN CONNECTION WITH THE USE OF DOCUMENTS AND THE INFORMATION CONTAINED IN THE DOCUMENTS.

© 2011 Open Mobile Alliance Ltd. All Rights Reserved.

Used with the permission of the Open Mobile Alliance Ltd. under the terms set forth above.

Contents

1. SCOPE (INFORMATIVE)	5
2. REFERENCES	6
2.1 NORMATIVE REFERENCES	6
2.2 INFORMATIVE REFERENCES	6
3. TERMINOLOGY AND CONVENTIONS	7
3.1 CONVENTIONS	7
3.2 DEFINITIONS	7
3.3 ABBREVIATIONS	7
4. INTRODUCTION (INFORMATIVE)	9
5. CBCS V1.0 TECHNICAL SPECIFICATIONS	11
5.1 CBCS.PEM-1 INTERFACE	11
5.1.1 CBCS.PEM-1 Input Template	11
5.1.2 CBCS.PEM-1 Output Template	14
5.2 CBCS.PEM-2 INTERFACE	17
5.3 CBCS-1 INTERFACE	18
5.3.1 CBCS-1 Request	18
5.3.2 CBCS-1 Response	20
5.4 ICAP REPRESENTATION OF CBCS-1	21
5.4.1 Transfer of the CBCS-1 Request by ICAP	21
5.4.2 Transfer of the CBCS-1 Response by ICAP	22
5.5 CBCS-2 INTERFACE	23
5.6 CBCS-3 INTERFACE	23
5.6.1 CBCS-3 Request and Response	24
5.7 ICAP REPRESENTATION OF CBCS-3	27
5.7.1 Transfer of the CBCS-3 Messages by ICAP	27
5.8 PRE-CATEGORIZED CONTENT	29
5.8.1 Use of Standard Metadata Schemes	29
5.8.2 Precategorized Content XML Schema	29
APPENDIX A. CHANGE HISTORY (INFORMATIVE)	31
A.1 APPROVED VERSION HISTORY	31
APPENDIX B. STATIC CONFORMANCE REQUIREMENTS (NORMATIVE)	32
B.1 SCR FOR CBCS.PEM-1	32
B.1.1 SCR for CBCS.PEM-1 Client	32
B.1.2 SCR for CBCS.PEM-1 Server	33
B.2 SCR FOR CBCS.PEM-2	33
B.2.1 SCR for CBCS.PEM-2 Client	33
B.2.2 SCR for CBCS.PEM-2 Server	34
B.3 SCR FOR CBCS-1	34
B.3.1 SCR for CBCS-1 Client	34
B.3.2 SCR for CBCS-1 Server	35
B.4 SCR FOR CBCS-2	37
B.4.1 SCR for CBCS-2 Client	37
B.4.2 SCR for CBCS-2 Server	37
B.5 SCR FOR CBCS-3	37
B.5.1 SCR for CBCS-3 Client	37
B.5.2 SCR for CBCS-3 Server	41
APPENDIX C. CATEGORIZATION SCHEMES SUPPORTED BY CBCS (NORMATIVE)	45
APPENDIX D. CBCS-1: ICAP ASSESSMENT (INFORMATIVE)	48
APPENDIX E. EXAMPLES OF CBCS-1 REQUESTS AND REPLIES OVER ICAP (INFORMATIVE)	49
E.1 CATEGORIZATION REQUEST FOR A HTTP REQUEST	49

E.2 CATEGORIZATION REQUEST FOR A HTTP RESPONSE49
 E.3 CATEGORIZATION REQUEST FOR SMS50
 E.4 CATEGORIZATION REQUEST WITH CONTENT REFERENCE50
 E.5 CATEGORIZATION REQUEST FOR SMS SHORTCODE51
 E.6 CBCS-1 RESPONSE EXPRESSED AS ICAP RESPONSE.....51
 E.7 CBCS-1 ERROR IN ICAP.....52
 APPENDIX F. ON THE USE OF METADATA SCHEMES IN CBCS (INFORMATIVE).....53
 F.1 ON THE USE OF MPEG-7 IN CBCS.....53

Figures

This document contains no figures.

Tables

Table 1 (normative) Permitted combination of mode attribute, PEM status code and action attribute.....16
 Table 2 (informative) Typical content of the screeningResult element in function of action attribute17
 Table 3 (normative) CBCS-1 specific status codes.....21
 Table 4 (normative) CBCS-3 specific status codes.....24
 Table 5 (normative) Semantics of the content-category-value element for supported Content categorization schemes45
 Table 6 (informative) How ICAP matches CBCS-1 criteria48

1. Scope (Informative)

This document represents the complete Technical Specification of the Categorization Based Content Screening (CBCS) 1.0 Enabler of the Open Mobile Alliance™, and contains specifications of the CBCS.PEM-1, CBCS.PEM-2, CBCS-1, CBCS-2 and CBCS-3 interfaces described in the CBCS Architecture Document [CBCS-AD].

The following items are within the scope of this Technical Specification of the CBCS 1.0 Enabler:

- A CBCS template for the PEM-1 interface [PEM1-TS] to create CBCS.PEM-1.
- Guidelines for the use of the CBCS.PEM-2 [PEM2-TS] interface and PEL [PEL-TS] in the CBCS Enabler.
- The specification of messages to be exchanged over the CBCS-1, CBCS-2 and CBCS-3 interfaces [CBCS-AD].
- A mapping of the CBCS-1 and CBCS-3 messages onto ICAP protocol messages [RFC3507].
- Guidelines for the use of meta-data standards for Pre-Categorized Content.
- The definition of a standard format for Pre-Categorized Content that contains its Content Category (or Content Categories) as meta-data.

The following items fall outside the scope of this Technical Specification of the CBCS 1.0 Enabler:

- The specification of the messages to be exchanged between the CBCS Enabler and other enablers to enforce Screening Rules when the CBCS Enabler is used in Proxy Usage Pattern [CBCS-AD].
- The specification of messages to be exchanged with Other Resources [CBCS-AD].
- The specification of the CBCS User Profile.
- The specification of specific Screening Rules or Content Categorization Rules.
- The standardization of specific Content Categories or categorization schemes.
- The specification of specific Content categorization methods and algorithms, and pattern matching techniques.
- CBCS User interaction mechanisms (such as customer facing warnings and requests for consent).
- Delegation of management rights.

This Technical Specification defines the CBCS 1.0 interfaces in Augmented Backus Naur Form (ABNF) [RFC4234], with the exception of the CBCS.PEM-1 and CBCS.PEM-2 interfaces based on PEM-1 [PEM1-TS] and PEM-2 [PEM2-TS] respectively, which are defined as XML Schema.

2. References

2.1 Normative References

- [CBCS-AD] “Categorization Based Content Screening Framework Architecture”, Open Mobile Alliance™, OMA-AD-CBCS-V1_0, URL:<http://www.openmobilealliance.org/>
- [CBCS-RD] “Categorization Based Content Screening Framework Requirements”, Open Mobile Alliance™, OMA-RD-CBCS-V1_0, URL:<http://www.openmobilealliance.org/>
- [ICAP-Extensions] “ICAP Extensions”, M. Stecher e.a., April 2003, URL:<http://www.icap-forum.org/documents/specification/draft-stecher-icap-subid-00.txt>
- [ISO 3166-2] “Codes for the representation of names of countries and their subdivisions - Part 2: Country subdivision code”, International Standards Organization, 1998, http://www.iso.org/iso/country_codes/background_on_iso_3166/iso_3166-2.htm
- [MPEG-7] “Information Technology – Multimedia Content Description Interface – Part 5: Multimedia Description Schemes – Amendmend 2: Multimedia Description Schemes User Preferences Extensions”, ISO/IEC 15938-5:2003/Amd.2:2005(E), International Standards Organization , URL:<http://www.iso.org/>
- [PEEM-AD] “Policy Evaluation, Enforcement and Management Architecture”, Open Mobile Alliance™, OMA-AD-Policy_Evaluation_Enforcement_Management-V1_0_0, URL:<http://www.openmobilealliance.org/>
- [PEEM-RD] “Policy Evaluation, Enforcement and Management Requirements”, Open Mobile Alliance™, OMA-RD-Policy_Evaluation_Enforcement_Management-V1_0, URL:<http://www.openmobilealliance.org/>
- [PEL-TS] “PEEM Policy Expression Language Technical Specification”, Open Mobile Alliance™, OMA-TS-PEEM_PEL-V1_0, URL:<http://www.openmobilealliance.org/>
- [PEM1-TS] “Policy Evaluation, Enforcement and Management Callable Interface (PEM-1) Technical Specification”, Open Mobile Alliance™, OMA-TS-PEEM_PEM1-V1_0, URL:<http://www.openmobilealliance.org/>
- [PEM2-TS] “Policy Evaluation, Enforcement and Management – Management Interface (PEM-2) Technical Specification”, Open Mobile Alliance™, OMA-TS-PEEM_PEM2-V1_0, URL:<http://www.openmobilealliance.org/>
- [RFC2119] “Key words for use in RFCs to Indicate Requirement Levels”, S. Bradner, March 1997, URL:<http://www.ietf.org/rfc/rfc2119.txt>
- [RFC2616] “Hypertext Transfer Protocol -- HTTP/1.1”, R. Fielding e.a., June 1999, URL:<http://www.ietf.org/rfc/rfc2616.txt>
- [RFC3507] “Internet Content Adaptation Protocol (ICAP) ”, J. Elson, A. Cerpa, April 2003, URL:<http://www.ietf.org/rfc/rfc3507.txt>
- [RFC3986] “Uniform Resource Identifier (URI): Generic Syntax”, T. Berners-Lee, R. Fielding, L. Masinter, January 2005, URL:<http://www.ietf.org/rfc/rfc3986.txt>
- [RFC4234] “Augmented BNF for Syntax Specifications: ABNF”, D. Crocker, Ed., P. Overell. October 2005, URL:<http://www.ietf.org/rfc/rfc4234.txt>
- [SCRRULES] “SCR Rules and Procedures”, Open Mobile Alliance™, OMA-ORG-SCR_Rules_and_Procedures, URL:<http://www.openmobilealliance.org/>

2.2 Informative References

- [OMADICT] “Dictionary for OMA Specifications”, Open Mobile Alliance™, OMA-ORG-Dictionary-V2_7, URL:<http://www.openmobilealliance.org/>

3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes are normative, unless they are explicitly indicated to be informative.

3.2 Definitions

Categorization Based Content Screening User	A Principal whose receivable or transmitted Content is subject to a CBCS Enabler implementation [CBCS-RD]
Categorization Based Content Screening User Profile	The User Profile [OMA-DICT] applicable to the CBCS Enabler [CBCS-RD]
Categorized Content	Content for which a set of Content Categories have been assigned [CBCS-RD]
Content	Digitized work that is processed, stored, or transmitted. It includes such things as text, presentation, audio, images, video, executable files, etc. Content may have properties such as media type, mime type, etc. [OMADICT]
Content Category	A category assigned to Content, aiming to describe the characteristics of the Content [CBCS-RD]
Content Scanning	The act of determining the Content Category (or Content Categories) of the Content [CBCS-RD]
Content Screening	The act of blocking, allowing or amending Content [CBCS-RD]

3.3 Abbreviations

ABNF	Augmented Backus Naur Form
CBCS	Categorization Based Content Screening
CBCS-1	CBCS Categorization interface
CBCS-2	CBCS Categorization Rules management interface
CBCS-3	CBCS Categorization association interface
CBCS.PEM-1	PEEM Callable interface extended for use in CBCS
CBCS.PEM-2	PEEM Management interface inherited for use in CBCS
CRLF	Carriage Return Line Feed character
CS	Classification Scheme (MPEG-7)
DS	Description Scheme (MPEG-7)
ESRB	Entertainment Software Rating Board
HTTP	Hyper Text Transfer Protocol
ICAP	Internet Content Adaptation Protocol
ICRA	Internet Content Rating Association
ISAN	International Standard Audiovisual Number
ISBN	International Standard Book Number
MD-5	Message Digest algorithm 5
MPAA	Motion Pictures of America Association
MPEG	Moving Picture Expert Group

OMA	Open Mobile Alliance
PEGI	Pan European Game Information
PEM-1	PEEM Callable interface
PEM-2	PEEM Management interface
PEEM	Policy Evaluation, Enforcement and Management
RIAA	Recording Industry of America Association
SMIL	Synchronized Multimedia Integration Language
SMS	Short Message Service
URI	Universal Resource Identifier
WAP	Wireless Application Protocol
XML	Extensible Markup Language

4. Introduction (Informative)

As end-user devices become more widespread and their multimedia capabilities evolve, they provide an increasing amount of Content to users. Consequently, a user's exposure to illegal, undesired or malicious Content also increases. Protecting users such as minors from inappropriate Content becomes a growing challenge.

The objective of the Categorization Based Content Screening (CBCS) Enabler is to screen Content before delivering it to the user, based on Content Categories. The CBCS Enabler can be applied to any Content regardless of the Enabler or protocol that is used to deliver the Content.

A Content Category qualifies the Content according to a categorization scheme. Examples of existing categorization schemes are the *Minimum Recommended Age (MRA)*, or the Recording Industry Association of America™'s (RIAA) *Parental Advisory* warning. The CBCS Enabler can obtain the Content Category for a given piece of Content from a Categorization Entity [CBCS-RD], from categorization meta-data in the Content itself, or by analyzing the Content.

The CBCS Enabler architecture describes two components [CBCS-AD]: a *Content Categorization Component* which categorizes Content or Content References, and a *Content Screening Component* which applies Screening Rules to determine whether the Content should be categorized, modified in any way and delivered.

The application of Screening Rules may involve processing of the following information:

- Content Categories, i.e. a qualification of the Content according to a categorization scheme.
- Other Content related information, for example the sender, owner or URI of the Content.
- A CBCS User Profile containing Content Screening preferences and other information pertinent to CBCS, such as the CBCS User's date of birth or MS-ISDN.
- Screening Rules, which may apply to one specific CBCS User (e.g. Content Screening preferences specified in the CBCS User Profile), or to a collective of CBCS Users (e.g., Screening Rules to be applied to all minors for legal reasons).

The standardization of this information, and in particular Content categorization schemes, Content Categories, CBCS User Profiles, Screening Rules and other Content related information, is outside the scope of the CBCS Enabler specification.

A CBCS Enabler implementation can be deployed in either the Proxy Usage Pattern, in which case it will evaluate and enforce the Screening Rules, or in the Callable Usage Pattern, in which case it will only evaluate the Screening Rules and respond to the requestor as to how to enforce them.

The result of evaluating Screening Rules can be one of the following Screening Actions:

- Not to deliver the Content to the CBCS User.
- To deliver the Content unmodified to the CBCS User.
- To deliver the Content to the CBCS User with modifications (e.g. with certain parts omitted).
- To deliver the Content to the CBCS User in combination with a textual or graphical warning.
- To deliver the Content to the CBCS User only after receiving consent from an Authorized Principal (e.g. a parent).

The remainder of this Technical Specification is organized as follows:

- Section 5.1 specifies the CBCS.PEM-1 interface for the CBCS Content Screening Component. The CBCS Enabler reuses PEM-1 interface from the Policy Evaluation, Enforcement Management [PEEM-AD] Enabler of the Open Mobile Alliance™. Section 5.1 defines CBCS.PEM-1 through a template with CBCS parameters for PEM-1.
- Section 5.2 prescribes how the PEM-2 interface is reused from the Policy Evaluation, Enforcement Management [PEEM-AD] Enabler of the Open Mobile Alliance™ to provide management of Screening Rules to create the CBCS.PEM-2 interface.
- Section 5.3 and 5.4 contain the specification the CBCS-1 interface, the functional interface through which a requestor makes a Content categorization request to a Content Categorization Component. Section 5.3 specifies the CBCS-1 interface in abstract terms, while section 5.4 describes a normative mapping of this interface to the ICAP protocol of the IETF [RFC3507].

- Section 5.5 specifies the CBCS-2 interface for managing Categorization Rules, which is defined by re-use of the PEM-2 interface of the Policy Evaluation, Enforcement Management [PEEM-AD] Enabler.
- Section 5.6 and 5.7 contain the specification the CBCS-3 interface, the management interface through which a requestor can manage (i.e. add, delete and list) associations between Content references and Content Categories in the Content Categorization Component.
- Section 5.8 specifies a format for Pre-Categorized Content, i.e. Content which contains its Content Category as meta-data. Section 5.8 also provides guidelines on how to handle Pre-Categorized Content in standard metadata formats such as MPEG-7.

In addition, there are a number of normative and informative appendices that provide additional information for the specifications.

5. CBCS V1.0 Technical Specifications

This chapter contains the specification of the CBCS V1.0 interfaces CBCS.PEM-1, CBCS.PEM-2, CBCS-1, CBCS-2 and CBCS-3 in this order. It also specifies a format for Pre-Categorized Content, and defines how the CBCS Enabler must handle existing Content metadata.

Regarding categorization management, a valid CBCS enabler implementation MUST implement at least one of the interfaces CBCS-2 and CBCS-3.

5.1 CBCS.PEM-1 interface

The Content Screening Component is an instance of the PEEM Enabler [PEEM-RD, PEEM-AD, PEM1-TS, PEM2-TS, PEL-TS] that evaluates and possibly enforces CBCS specific Screening Rules. The CBCS.PEM-1 interface by which to request Content screening is therefore derived from the standard PEM-1 interface with a CBCS template. Sections 5.1.1 and 5.1.2 define the input and output template for the CBCS request and reply, respectively.

5.1.1 CBCS.PEM-1 Input Template

The `templateID` to be used for the CBCS template for PEM-1 SHALL be `OMA_CBCS_1_Content_Screening_Input` and the `templateVersion` for this template SHALL be `V1.0.0`

The CBCS input template is specified as a file named `cbcs_pem1InputTemplate-v1_0.xsd` with the following contents:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:pem1-i="urn:oma:xml:peem:pem1-input-template:1.0"
  xmlns="urn:oma:xml:cbcs:pem1-input-template:1.0"
  targetNamespace="urn:oma:xml:cbcs:pem1-input-template:1.0">
  <xs:import namespace="urn:oma:xml:peem:pem1-input-template:1.0"
    schemaLocation="http://www.openmobilealliance.org/tech/profiles/PEM_1_GenericInputTemplateData-v1_0.xsd"/>
  <xs:complexType name="CBCSpem1InputTemplate-V1_0Type">
    <xs:complexContent>
      <xs:extension base="pem1-i:inputTemplateType">
        <xs:sequence>
          <xs:element name="screeningRequest">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="userInformation">
                  <xs:complexType>
                    <xs:complexContent>
                      <xs:extension base="xs:anyType">
                        <xs:attribute name="userInformationType" type="xs:string"
                          use="required"/>
                      </xs:extension>
                    </xs:complexContent>
                  </xs:complexType>
                </xs:element>
                <xs:element name="contextInformation">
                  <xs:complexType>
                    <xs:complexContent>
                      <xs:extension base="xs:anyType">
                        <xs:attribute name="contextInformationType"
```

```

        type="xs:string" use="required"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<xs:element name="contentDescriptor" type="xs:string"
  minOccurs="0"/>
<xs:choice>
  <xs:element name="content" type="xs:anyType"/>
  <xs:element name="contentLocator">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute name="locatorType" type="xs:string"
            use="required"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="contentIdentifier">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute name="identifierType" type="xs:string"
            use="required"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="contentDigest">
    <xs:complexType>
      <xs:complexContent>
        <xs:extension base="xs:anyType">
          <xs:attribute name="digestType" type="xs:string"
            use="required"/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
</xs:choice>
<xs:element name="categorizationMetadata">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="contentCategoryVector" type="xs:string"/>
      <xs:element name="contentProvider" type="xs:string"
        minOccurs="0"/>
      <xs:element name="categoryProvider" type="xs:string"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="signature" minOccurs="0">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:string">
              <xs:attribute name="signatureType"
                type="xs:string" use="required"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

```

```

        </xs:element>
    </xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="mode" default="evaluate">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:enumeration value="evaluate"/>
            <xs:enumeration value="evaluate and enforce"/>
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:schema>

```

This XML Schema defines a complex type `CBCSpem1InputTemplate-V1_0Type` which is an extension of the PEM-1 abstract `inputTemplateType`. The `CBCSpem1InputTemplate-V1_0Type` has one top level element, `screeningRequest` which has an attribute called `mode`.

The attribute `mode` indicates whether the request to the CBCS Enabler is made in *callable mode* or *proxy mode* and can take on the string values "evaluate" or "evaluate and enforce". Any implementation of the CBCS Enabler **MUST** implement callable mode and **MAY** implement proxy mode. If the value of the mode attribute is "evaluate", then the Content Screening Component **SHALL** *only evaluate* the Screening Rules and return an action to be enforced by the requesting Enabler. If the value of this attribute is "evaluate and enforce" and the CBCS Enabler supports proxy mode, then the Content Screening Component **SHALL** *evaluate and enforce* the Screening Rules.

The `screeningRequest` element contains a sequence of the following:

- An **OPTIONAL** `userInfo` element: if this element is present the Content Screening Component **SHALL** use this information to determine for which user(s) to apply the Screening Rules. If it is absent, the Content Screening Component **SHALL** evaluate Screening Rules that apply to all users. The `userInfo` element has an attribute `userInfoType` that indicates the type of user information provided (e.g. MS-ISDN, age, user profile). The `userInfo` element can have any kind of mixed content.
- An **OPTIONAL** `contextInfo` element to provide additional information the Content Screening Component may need for evaluating Screening Rules. Such information **MAY** include a user's location information, presence information or information about the user's device. The `contextInfo` element has an attribute `contextInfoType` that indicates the type of context information provided (e.g. location, device, presence). The `contextInfo` element can have any kind of content.
- A choice between `content`, `contentLocator`, `contentIdentifier` or `contentDigest`: the request can carry either raw Content, or a content reference which can be a *content locator*, a *content identifier* or a *content digest* in accordance with the CBCS-1 specification in section 5.3. If the CBCS.PEM-1 request carries Content, then the `contentDescriptor` element **MAY** be used to describe the Content type. If the CBCS.PEM-1 request contains a Content reference, then a type describes the reference type (e.g. "ISBN number" or "MD5 digest").

- An OPTIONAL `categorizationMetadata` element: the CBCS.PEM-1 request MAY carry categorization information in case the content has been pre-categorized. The structure of this element is in line with the definition of pre-categorized content in section 5.8.

An (informative) example of a valid CBCS.PEM-1 input document is:

```
<?xml version="1.0" encoding="UTF-8"?>
<pem1-i:policyInputData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:cbcs1-i="urn:oma:xml:cbcs:pem1-input-template:1.0"
  xmlns:pem1-i="urn:oma:xml:peem:pem1-input-template:1.0"
  xsi:schemaLocation="urn:oma:xml:cbcs:pem1-input-template:1.0
    http://www.openmobilealliance.org/tech/profiles/cbcs_pem1InputTemplate-V1_0.xsd">

  <policyInputTemplate xsi:type="cbcs1-i:CBCSpem1InputTemplate-V1_0Type"
    templateID="OMA_CBCS_1_Content_Screening_Input"
    templateVersion="V1.0.0">
    <screeningRequest mode="evaluate and enforce">
      <userInformation userInformationType="MS-ISDN">
        +34696858585
      </userInformation>
      <contextInformation contextInformationType="presence">
        IN_MEETING
      </contextInformation>
      <contentDescriptor>Some content</contentDescriptor>
      <contentLocator locatorType="URI">
        http://www.the-content-to-be-screened.net
      </contentLocator>
      <categorizationMetadata>
        <contentCategoryVector>
          ESRB T Comic Mischief ES CN, MRA 13 US
        </contentCategoryVector>
      </categorizationMetadata>
    </screeningRequest>
  </policyInputTemplate>
</pem1-i:policyInputData>
```

Disclaimer: please note that OMA does not specify the contents of the `userInformation`, `userInformationType`, `contextInformation` and `contextInformationType` elements and attributes.

This input document describes a request to screen content in proxy mode (evaluation and enforcement of Screening Rules), for a user with MS-ISDN +34696858585 whose presence information indicates that he or she is in meeting. The content to be screened is located at the URI `http://www.the-content-to-be-screened.net` and has been pre-categorized with the ESRB category “T” (Teenagers) for Spain and China indicating that the content contains *comic mischief*, as well as a minimum recommended age 13 for the US.

5.1.2 CBCS.PEM-1 Output Template

The `templateID` to be used for the CBCS output template for PEM-1 SHALL be `OMA_CBCS_1_Content_Screening_Output` and the `templateVersion` for this template SHALL be `V1.0.0`

The CBCS output template is specified as an XML schema file with the following contents:

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:pem1-o="urn:oma:xml:peem:pem1-output-template:1.0"
  xmlns="urn:oma:xml:cbcs:pem1-output-template:1.0"
  targetNamespace="urn:oma:xml:cbcs:pem1-output-template:1.0">
```

```

<xs:import namespace="urn:oma:xml:peem:pem1-output-template:1.0"
schemaLocation="http://www.openmobilealliance.org/tech/profiles/PEM_1_GenericOutputTemplateData-v1_0.xsd" />

<xs:complexType name="CBCSOutputTemplateType">
<xs:complexContent>
<xs:extension base="pem1-o:outputTemplateType">
<xs:sequence>
<xs:element name="screeningResult">
<xs:complexType>
<xs:complexContent>
<xs:extension base="xs:anyType">
<xs:attribute name="mode" use="optional">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:enumeration value="evaluate"/>
<xs:enumeration value="evaluate and enforce"/>
</xs:restriction>
</xs:simpleType>
</xs:attribute>
<xs:attribute name="action" use="required">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:enumeration value="block"/>
<xs:enumeration value="pass"/>
<xs:enumeration value="adapt"/>
<xs:enumeration value="warn"/>
<xs:enumeration value="consent required"/>
<xs:enumeration value="other"/>
</xs:restriction>
</xs:simpleType>
</xs:attribute>
<xs:attribute name="actionId" type="xs:string" use="optional"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:schema>

```

The CBCS output template describes a single output element, `screeningResult`, with three attributes: `mode`, `action` and `actionId`.

The OPTIONAL `mode` attribute can take on the string values "evaluate" or "evaluate and enforce". The `mode` attribute SHALL have the value "evaluate" if the Content Screening Component has *evaluated but not enforced* the Screening rules. The `mode` attribute SHALL have the value "evaluate and enforce" if the Content Screening Component has *evaluated and enforced* the Screening rules.

If the `mode` attribute is omitted then it SHALL be assumed that its value is equal to the `mode` attribute issued in the request (see section 5.1.1 above) that lead to this screening result.

The `action` attribute in combination with the CBCS.PEM-1 status code provides the requestor information about the result of the screening request. The `action` attribute can have string values "block", "pass", "adapt", "warn", "consent required" and "other". Table 1 (normative) lists the permitted combinations of CBCS.PEM-1 status code, `mode` attribute and `action` attribute, and specifies the meaning of each combination.

Table 1 (normative) Permitted combination of mode attribute, PEM status code and action attribute

mode	CBCS.PEM-1 status code	action	meaning
evaluate	2101 ALLOW	pass	the requesting Enabler may deliver the content to the user without modification
	2102 ALLOW (specified)	adapt	the requesting Enabler may only deliver the content to the user after adaptation.
		warn	the requesting Enabler should deliver the content to the user with a warning message.
		consent required	the requesting Enabler may only deliver the content to the user after soliciting consent from another Principal.
		other	The requesting Enabler should process the content as specified.
2401 DENY	block	the requesting Enabler should not deliver the content to the user	
evaluate and enforce	2701 SUCCESS	pass	the Content Screening Component has passed the content on for delivery to the user, without modification.
		block	the Content Screening Component has blocked the content
	2702 SUCCESS (specified)	adapt	the Content Screening Component has passed the content on for delivery to the user, with adaptations.
		warn	the Content Screening Component has passed the content on for delivery to the user, with a warning message.
		other	The Content Screening Component has processed the content as specified.

The OPTIONAL `actionId` attribute allows the Content Screening Component to return a unique identifier for the returned action. The value of the `actionId` attribute can be any string as long as it uniquely identifies the action. The `actionId` MAY be used by the requestor to provide the Content Screening Component with feedback about the screening result in a separate message exchange which is not specified as part of this PEM-1 template.

The contents of the `screeningResult` element can be any content (i.e. binary content, string or XML) and depend on the value of the `action` attribute. Table 2 (informative) describes what content the `screeningResult` may carry depending on the `action` attribute.

Table 2 (informative) Typical content of the screeningResult element in function of action attribute

mode	CBCS.PEM-1 status code	action	meaning
evaluate	2101 ALLOW	pass	N/A (normally no content)
		adapt	the adapted content, or instructions for how to adapt the content
	2102 ALLOW (specified)	warn	a warning to be displayed to the user before or while delivering the content
		consent required	identification of the Principal who has to provide consent before the content can be delivered
		other	instructions for how to process the content
2401 DENY	block	a message to the user indicating that the content has been blocked, and possibly why	
evaluate and enforce	2701 SUCCESS	pass	N/A (normally no content)
		block	description of the reason why the content was blocked
	2702 SUCCESS (specified)	adapt	the adapted content, or a description of how the content was adapted before delivery
		warn	the warning displayed to the user before delivery, or a description of the warning
		other	description of how the content was processed

An (informative) example of a valid CBCS.PEM-1 output document is:

```
<?xml version="1.0" encoding="UTF-8"?>
<pem1-o:policyOutputData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:cbs1-o="urn:oma:xml:cbsc:pem1-output-template:1.0"
  xmlns:pem1-o="urn:oma:xml:peem:pem1-output-template:1.0"
  xsi:schemaLocation="urn:oma:xml:cbsc:pem1-output-template:1.0
    http://www.openmobilealliance.org/tech/profiles/cbsc_pem1OutputTemplate-v1_0.xsd">

  <policyOutputTemplate xsi:type="cbsc1-o:CBCSOutputTemplateType"
    templateID="OMA_CBCS_1_Content_Screening_Output"
    templateVersion="V1.0.0">
    <StatusCode>2401</StatusCode >
    <StatusText>DENY</StatusText >
    <screeningResult mode="evaluate" action="block" actionId="Bx9V0Gis3n8NfZwOy5U">
      This content is for users 13 years of age and over.
    </screeningResult>
  </policyOutputTemplate>
</pem1-o:policyOutputData >
```

Disclaimer: please note that OMA does not specify the contents of the `actionId` attribute.

This example indicates that the Content Screening Component has evaluated but not enforced the policy rules. The result of the evaluation returned by the Content Screening Component is a recommendation to block the content. The reason for the blocking, which may be displayed to the user by the requesting Enabler, is *"This content is for users 13 years of age and over"*. The result also carries a unique `actionId` that may be user for later reference to this screening result, for example to provide feedback.

5.2 CBCS.PEM-2 Interface

The CBCS Enabler SHALL support the PEM-2 interface [PEM2-TS] for the management of Screening Rules. Screening Rules SHALL be instantiations of PEEM Policies [PEL-TS].

5.3 CBCS-1 Interface

The CBCS-1 interface allows a Content Categorization Requestor to request a Content Category (or Content Categories) for given Content (or a reference to Content) from a Content Categorization Component. The protocol on this interface is essentially a request-response protocol that carries Content or a Content reference in the request, and a list of zero or more Content Categories in the response.

The approach to the specification of CBCS-1 is the following: this section provides an abstract, implementation neutral specification of the CBCS-1 messages. That is, it specifies only the CBCS-1 request and response message parameters but not the request-response protocol used to transfer these, nor the message encoding scheme that is used.

Section 5.4 then specifies how the Internet Content Adaptation Protocol (ICAP) [RFC3507] can transfer the CBCS-1 request and response parameters. Though this Technical Specification does not specify encodings of the CBCS-1 messages in terms of other request-response protocols than ICAP, future versions of the CBCS Enabler MAY include additional CBCS-1 representations.

This section uses Augmented Backus Naur Form (ABNF) as defined in [RFC4234].

The CBCS-1 protocol is an application level protocol with two types of messages: request and response.

```
CBCS1-Message = CBCS1-Request / CBCS1-Response
```

The following paragraphs specify the CBCS-1 request and response structures in detail.

5.3.1 CBCS-1 Request

A CBCS-1 request can take the form of a request for categorization or a request for categorization capabilities.

A CBCS-1 categorization request carries either Content or a Content reference, preceded by a Content descriptor and followed by an OPTIONAL filter that specifies the categorization schemes to be used, and an OPTIONAL string parameter that specifies the categorization method to be used.

A CBCS-1 request for capabilities allows the requestor to discover which capabilities a Content Categorization Component supports, such as types of Content, Content references, categorization schemes, or categorization methods. A CBCS-1 capabilities request would typically be invoked before making a CBCS-1 categorization request. It allows a requestor to discover for example, if a Content Categorization Component can categorize Content by providing its URL.

The ABNF definition of the CBCS-1 request parameters is the following:

```
CBCS1-Request = CBCS1-categorization-request / CBCS1-capabilities-request
CBCS1-categorization-request = content-descriptor ( content / 1* content-reference )
                               [ filter ] [ categorization-method ]
CBCS1-capabilities-request = "CAPABILITIES"
content-descriptor = * CHAR
content = 1* OCTET                ; the content itself consists of any
                                   ; sequence of octets
content-reference = content-locator / content-identifier / content-digest
content-locator = content-locator-type content-locator-value
content-locator-type = token                ; eg. URI, SMS shortcode
content-locator-value = * CHAR
content-identifier = content-identifier-type content-identifier-value
content-identifier-type = token             ; eg. ISBN, title
content-identifier-value = 1* CHAR
content-digest = content-digest-type content-digest-value
```

```

content-digest-type = token          ; eg. MD5, RIPEMD-160, thumbnail
content-digest-value = 1* OCTET
filter = 1* categorization-scheme-identifier
categorization-scheme-identifier = token    ; eg. ESRB , ICRA , MPAA , MRA , PEGI , RIAA
categorization-scheme-identifier = / 1* CHAR ; free text for extension
categorization-method = 1* CHAR

```

The Content descriptor helps the Content Categorization Component determine how to decode and process the Content. Although it is a free text field allowing for any Content meta-data, it is RECOMMENDED to use the Content-Type element from [RFC2616] which describes the type of Content (for example "text/plain" or "image/jpeg"). The Content-Encoding, Content-Language and Content-Length elements defined in [RFC2616] are RECOMMENDED for use in the Content descriptor parameter as well.

The CBCS-1 request carries the Content to be categorized as a series of octets. This allows for text encoding of binary Content, though encoding techniques are not defined in this section. These encoding techniques are handled by the transfer protocol that carries the CBCS-1 parameters.

The CBCS-1 request message can also carry a Content reference rather than Content itself. A Content reference can take three forms:

- A Content locator provides information about where or how to access the Content to be categorized. A Content locator MAY be an URI or an SMS short code (a combination of a three or four digit short number and a keyword typically used to access WAP Push Content), or any other string value. If the CBCS-1 request carries a Content locator, then the Content Categorization Component SHALL determine the Content Categories of the Content at the specified location.
- A Content identifier is a string that provides an unambiguous textual description of the Content. A Content identifier MAY be an ISBN number for a book or an ISAN number for audiovisual Content. It MAY also be a movie title such as "Casablanca, 1949" or any other string value, provided it can unambiguously reference the Content. If the CBCS-1 request carries a Content identifier, then the Content Categorization Component SHALL use this Content identifier to determine the Content Categories to be returned to the requestor. The Content Categorization Component MAY use this Content identifier to retrieve cached Content Categories.
- A Content digest is a binary value that can be used as an unambiguous reference to Content. A Content digest MAY be a reduced form of the Content which conserves the main properties of the Content for categorization, e.g. a thumbnail or movie with strongly reduced frame size and rate. A Content digest MAY also be a digital digest computed on the Content, for example a MD5 or RIPEMD-160 hash. If the CBCS-1 request carries a Content digest, then the Content Categorization Component SHALL use this Content digest to determine the Content Categories to be returned to the requestor. If the Content digest is a digital digest (hash) then the Content Categorization Component MAY use this digital digest to retrieve cached Content Categories.

Any implementation of the CBCS Enabler MUST support at least one of these three content reference types (content-locator, content-identifier, content-digest).

The content-reference element has specific semantics. If the content-locator-type is URI then the content-locator-value MUST have the syntax defined in [RFC3986]. If the content-locator-type is SMS short code, then the content-locator-value MUST consist of one or more digits, which MAY be followed by a plain text keyword or message of no more than 161 characters.

If the content-identifier-type is ISBN, then the content-identifier-value MUST consist of 13 digits. If the content-identifier-type is ISAN, then the content-identifier-value MUST consist of 24 hexadecimal digits.

If multiple `content-reference` elements are provided in the `CBCS-categorization-request`, the Content Categorization Component can use any number of these to categorize the Content applying the `categorization-method` specified in the request, or if none is specified in the request, the default categorization method supported.

The OPTIONAL `categorization-method` element in the CBCS-1 request is of string value and allows the requestor to specify which categorization method it wants to be used. A typical use of this parameter is the following: after obtaining the supported categorization methods through a CBCS-1 capabilities request the requestor makes its Content categorization request with the categorization method according to the supported categorization method(s) obtained from the Content Categorization Component.

An example of a categorization method (expressed in plain text) may be *“first check if the category for an ISAN or the URL to the same content is cached; if so, return the cached categories; if not, retrieve the content from the URL and categorize it”*. Another example of a categorization method is *“ignore any cached categories for this content and first try to categorize the content digest; if this is impossible, retrieve the content from its URL and categorize it”*.

The OPTIONAL `filter` parameter in the CBCS-1 request allows the requestor to specify that it is only interested to receive Content Categories from specific categorization schemes. It does so by simply listing the names of the categorization schemes it wants the Content Categorization Component to use. These MAY be predefined categorization scheme identifiers, or they MAY be any string value so as to allow for the CBCS Enabler to use any categorization schemes.

The `categorization-scheme-identifier` element MUST support tokens for the following categorization schemes: ESRB (Entertainment Software Rating Board), ICRA (Internet Content Rating Association), MPAA (Motion Picture Association of America), MRA (Minimum Recommended Age), PEGI (Pan European Game Information) and Recording Industry Association of America (RIAA), as defined in Appendix C.

5.3.2 CBCS-1 Response

The CBCS-1 response message provides a status code, followed by a response to the categorization request or categorization capabilities request.

A response to a categorization request contains a (possibly empty) vector of Content Categories, an OPTIONAL signature and an OPTIONAL index.

A response to a categorization capabilities request contains a string with a description of capabilities supported by the Content Categorization Component.

The ABNF definition of the CBCS-1 response parameters is the following:

```

CBCS1-Response = status-code ( CBCS1-categorization-response / CBCS1-capabilities-response )
CBCS1-categorization-response = content-category-vector [ signature ] [ index ]
CBCS1-capabilities-response = 1* CHAR
status-code = *CHAR
content-category-vector = * ( content-category [ * region-code ] )
content-category = [ categorization-scheme-identifier ] content-category-value
content-category-value = * CHAR
region-code = 2 CHAR ; two characters, conforming to [ISO3166-2]
signature = signature-type signature-value
signature-type = token ; eg. DSA, ECDSA
signature-value = * OCTET
index = content-identifier / content-digest ; content-identifier and content-digest as above

```

The `status-code` element provides the requestor with information about the status of the response. The `status-code` element SHALL contain one of the CBCS-1 status codes listed in Table 3 below.

Table 3 (normative) CBCS-1 specific status codes

Normal response	
200	OK
Client errors	
400	Bad request
440	Badly formed filter
441	Corrupt or incomplete content
442	Unable to resolve content reference
443	No content at specified location
Server errors	
500	Server error
550	Server does not support requested categorization scheme
551	Server does not support content type
552	Server does not support content encoding

The `content-category-vector` element contains the Content Categories (zero or more) returned to the requestor. The CBCS Enabler supports a number of widely used categorization schemes, which are specified in Appendix C (normative). The CBCS-1 response MAY carry also any other type of Content Category (not specified in Appendix C) in plain text format.

The OPTIONAL `signature` argument in the CBCS-1 response carries a digital signature of the Content Categorization Component calculated on the Content and its Content Categories. The requestor MAY use this signature to create trusted Pre-Categorized Content, while a Content Screening Component MAY use this signature to verify the authenticity and integrity of Pre-Categorized Content it receives.

The OPTIONAL `index` element MAY be used by the Content Categorization Component if it caches the categorization result. It allows future requests to be made with the `index` value alone, rather than with the complete Content. This can considerably improve the efficiency of the CBCS-1 protocol if categorization requests occur often for a piece of Content.

5.4 ICAP Representation of CBCS-1

This section defines how the CBCS-1 request and response parameters can be transferred using a widely accepted protocol for Content vectoring, the Internet Content Application Protocol (ICAP) [RFC3507].

The ICAP representation CBCS-1 messages defined in this section adheres to the following principles:

- If the Content to be categorized is a HTTP request or HTTP response, then it MUST be possible to use ICAP as described in [RFC3507].
- If the Content to be categorized is not a HTTP request or HTTP response, then the content or content reference element in a CBCS-1 request SHALL be transferred in the encapsulated body of the ICAP message.
- CBCS-1 parameters other than content or content-reference SHALL be transferred in ICAP headers. Where necessary, new user-defined X-extension headers are introduced for these parameters.

5.4.1 Transfer of the CBCS-1 Request by ICAP

A `CBCS1-request` message consisting of a `CBCS1-categorization-request` SHALL be transferred by an ICAP `REQMOD` or `RESPMOD` method [RFC3507]. A `CBCS1-request` message consisting of a `CBCS1-capabilities-request` SHALL be transferred by an ICAP `OPTIONS` method [RFC3507].

If the `CBCS1-request` contains a content element representing a HTTP request or HTTP response, then:

- If the content element consists of a HTTP request, then it SHALL be encapsulated in an ICAP `REQMOD` method as specified in [RFC3507].

- If the content element consists of a HTTP response, then it SHALL be encapsulated in an ICAP RESPMOD method as specified in [RFC3507], using chunk-encoding to encapsulate the body of the HTTP response.
- The CBCS-1 content-descriptor element SHOULD be encoded as one or more [RFC2616] entity-header elements in the encapsulated header section of the ICAP message, e.g. if it comes in the form of a Content-Type or Content-Language entity header as recommended in section 5.3.1.
- The CBCS-1 content-descriptor element MAY be transferred in the user-defined X-Content-Descriptor extension header if it cannot be expressed as [RFC2616] entity-header. The syntax of the X-Content-Descriptor extension header is:

```
X-Content-Descriptor = "X-Content-Descriptor:" content-descriptor
                        ; content-descriptor as defined in section 5.3.1.
```

If the CBCS1-request does not contain a content element representing a HTTP request or HTTP response, then:

- The content or content-reference element of the CBCS1-request SHALL be transferred in the encapsulated body part of an ICAP RESPMOD message, using chunk encoding.
- If the CBCS1-request contains a content element, then it SHOULD be encoded as a single chunk.
- If the CBCS1-request contains a content-reference element, then it SHALL be encoded in two chunks. The first chunk SHALL contain the locator-type, identifier-type or digest-type and the second chunk SHALL contain the locator-value, identifier-value or digest-value.
- The content-descriptor element of the CBCS1-request SHOULD be transferred in the chunk encoded encapsulated body [RFC3507] of the ICAP message, e.g. if it comes in the form of a Content-Type or Content-Language entity header as recommended in section 5.3.1.
- The content-descriptor element of the CBCS1-request MAY be transferred in the user-defined X-Content-Descriptor extension header if it cannot be expressed by the RECOMMENDED Content-Type, Content-Encoding, Content-Language or Content-Length elements [RFC2616] in the chunk encoded encapsulated body of the ICAP message.

The CBCS-1 filter element SHALL be transferred in the user-defined X-Filter extension header. The syntax of the X-Filter extension header is:

```
X-Filter = "X-Filter:" categorization-scheme-identifier
           * ( separator categorization-scheme-identifier
               ; categorization-scheme-identifier is a token or string
               ; as defined in section 5.3.1
           )
separator = token ; for example comma
```

A CBCS1-capabilities-request SHALL be transferred in an ICAP OPTIONS method by adding the keyword "CAPABILITIES" to end of the Net_Path part of the ICAP URI, preceded by the "/"-encoding. Example of a capabilities request:

```
OPTIONS icap://192.168.1.34:1344/CAPABILITIES ICAP/1.0
```

Note that this codification of the CBCS-1 request in an ICAP request message fully conforms to [RFC3507].

5.4.2 Transfer of the CBCS-1 Response by ICAP

A CBCS1-response message consisting of a CBCS1-categorization-response SHALL be transferred by an ICAP REQMOD or RESPMOD response [RFC3507]. A CBCS1-response message consisting of a CBCS1-capabilities-response SHALL be transferred by an ICAP OPTIONS response [RFC3507].

The ICAP status code SHALL match the CBCS-1 status-code element (see Table 3 in section 5.3.2).

If the CBCS1-response consists of a CBCS1-categorization-response, then the X-Attribute ICAP extension header in the ICAP response SHALL contain the CBCS-1 content-category-vector element if and only if it is not null. The elements of the content-category-vector MUST be comma separated. The X-Attribute ICAP extension header MUST be omitted if the content-category-vector element is null, to comply with [ICAP-Extensions].

If the CBCS1-response consists of a CBCS1-categorization-response, then the OPTIONAL CBCS-1 signature element SHALL be transferred in the user-defined X-Signature extension header in the ICAP response. The syntax of the X-Signature extension header is:

```
X-Signature = "X-Signature:" signature-type separator signature-value
separator = token           ; for example comma
signature-type = token      ; as defined in section 5.3.2
signature-value = * OCTET  ; as defined in section 5.3.2
```

If the CBCS1-response consists of a CBCS1-categorization-response, then the OPTIONAL CBCS-1 index element SHALL be transferred in the encapsulated body part of the ICAP response message, as it represents a modified (reduced) form of the original Content submitted in the request. If the CBCS-1 index element is not present, then the ICAP encapsulated body SHALL be the null-body [RFC3507].

If the CBCS1-response consists of a CBCS1-categorization-response, then the OPTIONAL X-Response-Desc ICAP extension header [ICAP-Extensions] MAY be used to provide the requestor with additional information about the response. In case the ICAP status code is 200, the X-Response-Desc extension header MAY contain:

- The string “categorized” if the CBCS-1 content-category-vector element is non-empty and the CBCS-1 signature and index elements are not present.
- The string “categorized with index” if the CBCS-1 content-category-vector element is non-empty, the CBCS-1 index element is present and the CBCS-1 signature element is not present.
- The string “categorized with signature” if the CBCS-1 content-category-vector element is non-empty, the CBCS-1 signature element is present and the CBCS-1 index element is not present.
- The string “categorized with index and signature” if the CBCS-1 content-category-vector element is non-empty, and both the CBCS-1 signature and index elements are present.

If the CBCS1-response consists of a CBCS1-capabilities-response, then the result parameters SHALL be transferred in the encapsulated body of the ICAP OPTIONS response with the following syntax:

```
X-CBCS1-capabilities = "X-CBCS1-capabilities:" 1* CHAR
```

Note that this codification of the CBCS-1 response in an ICAP response message fully conforms to [RFC3507].

5.5 CBCS-2 Interface

The CBCS-2 interface for the management of Content Categorization Rules SHALL comply with [PEM2-TS]. Content Categorization Rules MAY be instantiations of PEEM Policies [PEL-TS], but MAY also be other types of rules.

5.6 CBCS-3 Interface

The CBCS-3 interface allows a Categorization Management Requestor to associate Content references (e.g., URIs) with Content Categories.

The protocol on this interface is essentially a request-response protocol that can also be used to request listings of the supported Content Categories and the associated Content references, or modifications of these.

The approach to the specification of CBCS-3 is the following: this section provides an abstract, implementation neutral specification of the CBCS-3 messages. That is, it specifies only the CBCS-3 request and response message parameters but not the request-response protocol used to transfer these, nor the message encoding scheme that is used.

Section 5.7 then specifies how the Internet Content Adaptation Protocol (ICAP) [RFC3507] can transfer the CBCS-3 request and response parameters. Though this Technical Specification does not specify encodings of the CBCS-3 messages in terms of other request-response protocols than ICAP, future versions of the CBCS Enabler MAY include additional CBCS-3 representations.

This section uses Augmented Backus Naur Form (ABNF) as defined in [RFC4234].

The CBCS-3 protocol defines two types of messages: request and response.

```
CBCS3-Message = CBCS3-Request / CBCS3-Response
```

The following paragraphs specify the CBCS-3 request and response structures in detail.

5.6.1 CBCS-3 Request and Response

A CBCS-3 request can be used to request listings of the supported Content Categories and the associated Content references, or modifications of these.

Content Categories can be created (and removed) and Content references can be associated with Content Categories (see section 5.6.1.2). The already supported Content Categories can be listed, as well as the Content references that are associated with a particular Content Category (see section 5.6.1.1).

The CBCS-3 request can also be used to query the Content Categorization Component for its capabilities, and in particular for the Content references it supports. A CBCS-3 capabilities query would typically be invoked before making a request to associate a Content reference with a category. It allows a requestor to discover for example, if a Content Categorization Component can cache an association between a Content Category and a particular Content reference type such as URL.

The ABNF definition of the CBCS-3 request parameters is the following:

```
CBCS3-Request = operation-descriptor [ list-req-parameter / add-remove-req-parameter ]
operation-descriptor = "LIST" ; see section 5.6.1.1
                      / "ADD" ; see section 5.6.1.2
                      / "REMOVE" ; see section 5.6.1.2
                      / "CAPABILITIES" ; see section 5.6.1.3
```

The CBCS-3 request consists of an operation descriptor and a set of operation parameters.

The operation-descriptor identifies the type of operation that is to be performed (LIST, ADD, REMOVE or CAPABILITIES). In case the LIST operation is to be performed, then the list-req-parameter SHALL be used as described per section 5.6.1.1 In case the ADD or REMOVE operation is to be performed, then the add-remove-req-parameter SHALL be used as described per section 5.6.1.2. If the CAPABILITIES operation is to be performed, then the add-remove-req-parameter and list-req-parameter SHALL NOT be used.

The CBCS-3 response consists of a status code and a particular set of response parameters that differ per operation request.

```
CBCS3-Response = status-code ( list-res-parameter / add-remove-res-parameter
                             / capabilities-res-parameter )
```

The status-code element provides the requestor with information about the status of the response. The status-code element SHALL contain one of the CBCS-3 status codes listed in Table 4 below.

Table 4 (normative) CBCS-3 specific status codes

Normal response	
200	OK
Client errors	
400	Bad request
Server errors	

500	Server error
-----	--------------

5.6.1.1 List Operation

The LIST operation can be used to request listing of the Content references that are associated to a particular Content Category which in turn may be associated with a particular categorization scheme (LIST <content-reference-type> <content-category>). Notice that the content-category definition MAY include a categorization-scheme-identifier as defined in section 5.3.2. The content-reference-type is described in line with section 5.3.1; being either a content-locator-type, a content-identifier-type or a content-digest-type. A valid CBCS-3 implementation SHALL support at least one of these types. As Content references can be of different types (as described per section 5.3.1), the requested type is to be indicated in the request. CBCS-3 MUST for a particular Content Category support the listing of content-locator-type Content References and MAY support the content-identifier-type Content References and content-digest-type Content References. The Content Category that the listing applies to is defined as per the content-category description in section 5.3.2. The LIST operation can also be used for the inverse: to list the Content Categories associated to a particular Content reference scheme (LIST <content-reference >).

The LIST operation can also be used to request listing of the already supported categorization schemes (LIST CATEGORIZATIONSCHMES) or Content Categories associated with a particular categorization scheme (LIST CATEGORIES <categorization-scheme-identifier>); the response carries the already supported categorization schemes (categorization-scheme-identifier) respectively the already supported Content Categories (content-category).

The following paragraph describes the request parameters:

```
list-req-parameter = "CATEGORIZATIONSCHMES"
                    ; request list of supported categorization schemes
                    / "CATEGORIES" categorization-scheme-identifier
                    ; request list of supported Content Categories associated with a
                    ; particular categorization scheme
                    / ( content-reference-type content-category )
                    ; request list of Content references of a certain type that are
                    ; associated with a certain Content category for a certain
                    ; categorization scheme
                    ; for the definition of content-category see section 5.3.2
                    / content-reference
                    ; request list of Content Categories associated with a certain
                    ; Content reference

content-reference-type = content-locator-type ; see section 5.3.1
                       / content-identifier-type ; see section 5.3.1
                       / content-digest-type ; see section 5.3.1
```

The following paragraph describes the response parameters:

```
list-res-parameter = ( * categorization-scheme-identifier )
                    / ( * content-category )
                    / ( content-category content-reference-list )
                    ; for the definition of categorization-scheme-identifier
                    ; see section 5.3.1
                    ; for the definition of content-category see section 5.3.2

content-reference-list = *content-reference
```

The response to the list request of the already supported categorization schemes provides the status code (see description above) and the supported categorization schemes.

The response to the list request of the already supported Content Categories provides the status code (see description above) and the supported Content Categories.

The response to the request of the listing of the Content references that are associated to a particular Content Category, provides the status code (see description above) and carries also the Content Category (content-category) and the list of associated Content references (content-reference-list). If the request contained a particular content-

reference-type, then the response MUST contain the matching content-reference. If the Content Category is associated to one or more categorization schemes, then the response MUST list these schemes.

The response to the list request for Content Categories associated to a particular Content reference provides the status code (see description above) and the associated Content Categories.

5.6.1.2 Add / Remove Operation

The ADD operation request can be used to introduce categorization scheme (ADD CATEGORIZATIONScheme < categorization-scheme-identifier >), or to introduce a Content Category to a particular categorization scheme (ADD CATEGORY < content-category >, where the < categorization-scheme-identifier > is part of the content-category as defined per section 5.3.2) that can consequently be used to associate Content references with that Content Category (ADD < content-reference > < content-category > < include-list-in-response >). Also (just) a content reference can be added; this is to facilitate an implementation that is able to determine by itself what categories should be associated with that content reference (ADD < content-reference >).

Notice that a content-reference entails various types: a content-locator-type, a content-identifier-type or a content-digest-type. A valid CBCS-3 implementation SHALL support at least one of these types.

The REMOVE operation request can be used to remove a supported categorization scheme (REMOVE CATEGORIZATIONScheme < categorization-scheme-identifier >) including all its associations, or to remove one or more supported Content Categories for a certain categorization scheme (REMOVE CATEGORY * < content-category-value > * < categorization-scheme-identifier >) including all their associations. It can be used to remove particular Content references (REMOVE * < content-reference >) resulting in the removal of all appropriate associations with those content references and it can also be used to remove particular Content-references-to-Content-Category associations (REMOVE * < content-reference > * < content-category >).

A valid CBCS-3 implementation SHALL support at least one of the following two options; either the set of ADD/REMOVE operations for categorization schemes, category-scheme associations and Content reference-Content Category associations, or the set of ADD/REMOVE operations for categorization schemes, category-scheme associations and Content references.

In the ADD and REMOVE operation request the OPTIONAL include-list-in-response parameter can be used to denote whether the response should carry the list of categorization schemes (as a reply to the add/remove categorization scheme request), the list of Categories (as a reply to the Add/Remove Content Category request) or the list of Content references associated with a particular Content Category (as a reply to the Add/Remove Content references-to-Content Category-association request).

```
add-remove-req-parameter =
    ( ( "CATEGORIZATIONScheme" categorization-scheme-identifier )
      / ( "CATEGORY" * content-category-value * categorization-scheme-identifier )
      / ( * content-reference * content-category )
      / ( * content-reference )
    )
    ["include-list-in-response"]
    ; for content-reference-type see section 5.6.1.1
    ; for content-category see section 5.3.2
```

The ADD/REMOVE operation response provides the status-code (see description above), the OPTIONAL list (see list-res-parameter description above when "include-list-in-response" was included in the request) and an OPTIONAL operation response description (response-description). The operation response description provides extra information as to what extent the request has been lived up to, for example "TestCategory added without error", "www.test.com added without error".

```
add-remove-res-parameter = [response-description] [list-res-parameter]
response-description = 1* CHAR
```

5.6.1.3 Capabilities Operation

The CAPABILITIES operation can be used to query a Content Categorization Component as to which Content reference types (e.g. URLs, Content identifiers, digests) it supports.

The CAPABILITIES response parameter consists of a list of Content reference types:

```
capabilities-res-parameter = * content-reference-type
```

5.7 ICAP Representation of CBCS-3

This section defines how the CBCS-3 request and response parameters can be transferred using the Internet Content Application Protocol (ICAP) [RFC3507].

The ICAP representation CBCS-3 messages defined in this section adheres to the following principles:

- CBCS-3 Request parameters SHALL be transferred in the OPTIONS method request-line.
- CBCS-3 Response parameters SHALL be transferred in ICAP headers. Where necessary, new user-defined X-extension headers are introduced for these parameters.

5.7.1 Transfer of the CBCS-3 Messages by ICAP

A CBCS-3 request message SHALL be encoded in an ICAP OPTIONS method [RFC3507]:

- The CBCS-3 parameters SHALL be added to the end of the ICAP URI, following the guidance provided by [RFC3507]; using the “?”-encoding to denote the parameters.
- The operation-descriptor (LIST, ADD, REMOVE) SHALL be added to end of the Net_Path part of the ICAP URI, preceded by the “/”-encoding.

A CBCS-3 response message shall be encoded in an ICAP OPTIONS response

- The CBCS-3 status-code SHALL match the ICAP status code field.

5.7.1.1 Request / Response for the List operation

- If the CBCS3-request is a request to list categorization schemes, then the operation-descriptor (LIST) SHALL be followed by “CATEGORIZATIONSCHEMES”.

For example: OPTIONS icap://192.168.1.34:1344/LIST?CATEGORIZATIONSCHEMES ICAP/1.0

- The response carries the list of the supported categorization schemes in the encapsulated body, each separated by a new line/CRLF character:

```
X-list-categorization-schemes = ( "X-list-categorization-schemes:"
                                CRLF categorization-scheme-list )
```

```
categorization-scheme-list = * ( categorization-scheme-identifier CRLF )
                               ; categorization-scheme-identifier see section 5.3.1
```

- If the CBCS3-request is a request to list Content Categories, then the operation-descriptor (LIST) SHALL be followed by “CATEGORIES”, followed by the appropriate categorization-scheme-identifier of which the associated Content Categories should be listed.

For example: OPTIONS icap://192.168.1.34:1344/LIST?CATEGORIES?TestCategoryScheme ICAP/1.0

- The response carries the list of the supported Content Categories in the encapsulated body, each separated by a new line/CRLF character:

```
X-list-categories = "X-list-categories:" CRLF content-category-list
```

```
content-category-list = * ( content-category CRLF )
                        ;content-category see section 5.3.2
```

- If, in the response, the content-category includes the OPTIONAL categorization-scheme-identifier, then each content-category-value (see 5.1.2) SHALL be followed by a “ ”/space-character, followed by the appropriate categorization-scheme-identifier.

- If the CBCS3-request is a request to list the Content references that are associated to a particular Content Category, then the operation-descriptor (LIST) SHALL be followed by the content-reference-type (e.g. URI), followed by the content-category.

For example: OPTIONS icap://192.168.1.34:1344/LIST?URI?TestCategory ICAP/1.0

- If in the request, the content-category includes the OPTIONAL categorization-scheme-identifier, then the operation-descriptor (LIST) SHALL be followed by the content-reference-type (e.g. URI), followed by the content-category followed by the appropriate categorization-scheme-identifier.

For example: OPTIONS icap://192.168.1.34:1344/LIST?URI?TestCategory?TestCategorizationScheme ICAP/1.0

- The response carries in its encapsulated body the Content Category, which MAY be followed by the associated categorization scheme identifier, followed by a list of the associated Content references (see following description). The Content Category SHALL be mapped to the X-Attribute ICAP extension header [ICAP-Extensions]. The content-category, categorization-scheme-identifier and the Content-references are each separated by a new line/CRLF character:

```
X-list-references = "X-list-references:" CRLF content-reference-list
```

```
content-reference-list = *(content-reference-value CRLF)
                        ;content-reference-value see section 5.3.1
```

5.7.1.2 Request / Response for the Add / Remove operation

- If the CBCS3-request is a request to add or remove a categorization scheme (identifier), then the operation-descriptor (ADD or REMOVE) SHALL be followed by "CATEGORIZATIONScheme", followed by a categorization-scheme-identifier(s), each separated by the "?"-encoding, followed by the OPTIONAL "include-list-in-response".

For example: OPTIONS icap://192.168.1.34:1344/ADD?CATEGORIZATIONScheme?TestCategory?include-list-in-response ICAP/1.0

- If the CBCS3-request is a request to add or remove a Content Category, then the operation-descriptor (ADD or REMOVE) SHALL be followed by a "CATEGORY", potentially followed by the categorization scheme identifier(s) (as defined per section 5.3.1) to indicate the scheme(s) that should be extended/decreased, followed by a content-category-value, each separated by the "?"-encoding, followed by the OPTIONAL "include-list-in-response".

For example: OPTIONS icap://192.168.1.34:1344/ADD?CATEGORY?TestCategory?include-list-in-response ICAP/1.0

- If the CBCS3-request is a request to associate/dissociate a Content reference with a Content Category, then the operation-descriptor (ADD or REMOVE) SHALL be followed by the content-reference, followed by the content-category (which includes an OPTIONAL categorization-scheme-identifier (as defined per section 5.3.2) followed by the content-category-value) followed by the OPTIONAL "include-list-in-response".

For example: OPTIONS icap://192.168.1.34:1344/ADD?URI?www.test.com?TestCategory?include-list-in-response ICAP/1.0

- If the CBCS3-request is a request to add or remove (just) a Content reference, then the operation-descriptor (ADD or REMOVE) SHALL be followed by the content-reference.

For example: OPTIONS icap://192.168.1.34:1344/ADD?URI?www.test.com ICAP/1.0

- The response provides the status-code (see description above), the OPTIONAL list (when "include-list-in-response" was included in the request) and an OPTIONAL operation response description (response-description).

- When the `CBCS3-request` is a request to add or remove a categorization scheme, then returned list is a list of supported categorization schemes (`X-list-categorization-schemes`) as described per section 5.6.1.1.
- When the `CBCS3-request` is a request to add or remove a Content Category, then returned list is a list of supported Content Categories (`X-list-categories`) as described per section 5.6.1.1.
- When the `CBCS3-request` is a request to add or remove a Content reference association to a certain Content Category, then the returned list is a list of the remaining associations (`X-list-references`) as described per section 5.6.1.1.
- When the `CBCS3-request` is a request to add or remove a Content reference, then the return of a list does NOT need to be supported.

The response description SHALL be represented as per the following ICAP extension header:

```
X-response-description = "X-response-description:" * CHAR CRLF
```

5.7.1.3 Request / Response for the Capabilities operation

The ICAP response transfers the list of supported Content references in the encapsulated body of the ICAP OPTIONS response as follows:

```
X-CBCS3-capabilities = "X-CBCS3-capabilities:" * ( CRLF content-reference-type ) CRLF
```

5.8 Pre-Categorized Content

Pre-Categorized Content is Categorized Content which includes its Content Category and the Content Provider's identity. It is used on the CBCS.PEM-1 interface and proxy interface to submit requests to the Content Screening Component with Content that has been categorized previously.

5.8.1 Use of Standard Metadata Schemes

This section specifies how the CBCS Enabler should handle Pre-Categorized Content encoded by standard metadata schemes specified elsewhere, outside the CBCS Technical Specifications.

Content submitted to the CBCS Enabler for screening may have metadata associated with it, described in one of the existing standard metadata schemes such as MPEG-7, SMIL, TV Anytime, EBU P/Meta, I3A DIG35 Metadata Specification, etc. Some of these standards support metadata that can be interpreted as Pre-Categorized Content.

This section specifies which standard metadata schemes the CBCS Enabler should be able to process, and which parts of these metadata schemes are suitable for encoding Pre-Categorized Content:

5.8.1.1 MPEG-7

In case the content is provided in MPEG-7, the CBCS Screening Component SHOULD be able to parse and interpret the `ParentalGuidanceType` of the MPEG-7 Classification Description Scheme and the `DisseminationType` of the Availability Description Scheme, both defined in [MPEG-7]. The CBCS Screening Component MAY be able to parse and interpret other MPEG-7 Description Schemes to obtain additional metadata that can be used in Screening Rules.

5.8.2 Precategorized Content XML Schema

In addition, this section specifies a simple and open format for Pre-Categorized Content that allows for the association of Content Categories and Content Provider Identity with any type of Content.

Pre-Categorized Content consists of the following five elements:

- `content-category-vector` is a list of content categories according to various categorization schemes, with OPTIONAL region codes (see section 5.3.2 for its definition).
- `content-provider` is a token or string that identifies the Content Provider.

- `category-provider` is a token or string that identifies the Content Category provider. Note that this identifier differs from the `categorization-scheme-identifier`. The latter identifies a categorization scheme, while the `category-provider` identifies the entity that categorized the Content. The Pre-Categorized Content SHOULD include this element if the Content was categorized by a Categorization Entity [CBCS-RD] other than the Content Provider itself. The CBCS Enabler MAY use the `category-provider` to determine whether it considers the source of the Content Categories in the Pre-Categorized Content trustworthy. If so, then the CBCS Enabler MAY decide to use the Content Categories provided in the Pre-Categorized Content to screen the content, or otherwise the CBCS Enabler SHOULD request Content Categories from a trusted Categorization Entity before screening the Pre-Categorized Content.
- `signature` allows a Content Screening Component to verify the integrity and origin of Pre-Categorized Content. This OPTIONAL parameter consists of a signature type and a signature value (see section 5.3.2 for its definition). The digital signature SHOULD be used when a Content Provider can't be trusted to associate the appropriate Content Category with the Content it provides. In this case the Content SHOULD be pre-categorized by a trusted Categorization Entity, which digitally signs the Content and its Content Categories. The digitally signed Pre-Categorized Content (consisting of the Content, its Content Categories and the digital signature of both) can then be distributed and forwarded by non trusted parties.
- `content-descriptor` is free text that can be used to describe the content (see section 5.3.1 for its definition). It can contain any additional meta-data to form part of the Pre-Categorized Content.
- `content` can be any series of octets (see section 5.3.1 for its definition). This allows for XML encoded content as well as character encoded binary Content, however such encoding techniques are not defined at this level.

The ABNF definition of Pre-Categorized Content is the following:

```
pre-categorized-content = content-category-vector
                          [ content-provider ]
                          [ * category-provider ]
                          [ signature ]
                          [ content-descriptor ]
                          content

content-provider = * CHAR      ; free text to identify the content provider
category-provider = * CHAR    ; free text to identify the category provider
```

The specifications of the `content-descriptor` and `content` elements are given in section 5.3.1. The specifications of the `content-category-vector` and `signature` elements are given in section 5.3.2.

The following semantic restriction applies: if the OPTIONAL `category-provider` sequence is present, then it MUST have the same number of elements as the `content-category-vector`, and element *i* in the `category-provider` sequence SHALL identify the `category-provider` for content-category *i* in the `category-provider` sequence. If `category-provider` element in any position *j* is the empty string, this means that no `category-provider` identity is provided for content-category *j* in the `content-category-vector`.

One may express only pre-categorization information (Content Categories + Content Provider identity + categorization provider identity + signature) for use with external meta-data schemes or external Content containers. To this effect, the element `categorization-metadata` is defined as consisting of only the `content-category-vector`, the OPTIONAL `content-provider`, OPTIONAL `category-provider` sequence and OPTIONAL `signature`:

```
categorization-metadata = content-category-vector
                          [ content-provider ]
                          [ * category-provider ]
                          [ signature ]
```

Appendix A. Change History

(Informative)

A.1 Approved Version History

Reference	Date	Description
OMA-TS-CBCS-V1_0-20111206-A	06 Dec 2011	Approved by TP Ref TP Doc# OMA-TP-2011-0429-INP_CBCSF_V1_0_ERP_for_Final_Approval

Appendix B. Static Conformance Requirements (Normative)

B.1 SCR for CBCS.PEM-1

B.1.1 SCR for CBCS.PEM-1 Client

B.1.1.1 SCR for CBCS.PEM-1 Client Support

Item	Function	Reference	Requirement
CBCS-PEM1-SUPPORT-C-001-M	Support for the PEM-1 interface [PEM1-TS]	5.1.1	

B.1.1.2 SCR for CBCS.PEM-1 Client Request

Item	Function	Reference	Requirement
CBCS-PEM1-REQ-W-CBCSTEMPLATE-C-001-M	PEM-1 Request with CBCS Input Template	5.1.1	CBCS-PEM1-SUPPORT-C-001-M AND (CBCS-PEM1-REQ-W-CBCSTEMPLATE-C-002-O OR CBCS-PEM1-REQ-W-CBCSTEMPLATE-C-003-O)
CBCS-PEM1-REQ-W-CBCSTEMPLATE-C-002-O	Request with Content	5.1.1	
CBCS-PEM1-REQ-W-CBCSTEMPLATE-C-003-O	Request with Content reference	5.1.1	
CBCS-PEM1-REQ-W-CBCSTEMPLATE-C-004-O	Support for Content descriptor	5.1.1	
CBCS-PEM1-REQ-W-CBCSTEMPLATE-C-005-O	Support for user information	5.1.1	
CBCS-PEM1-REQ-W-CBCSTEMPLATE-C-006-O	Support for context information element	5.1.1	
CBCS-PEM1-REQ-W-CBCSTEMPLATE-C-007-O	Support for pre-categorization information	5.1.1	

B.1.1.3 SCR for CBCS.PEM-1 Client Response

Item	Function	Reference	Requirement
CBCS-PEM1-RES-W-CBCSTEMPLATE-C-001-M	PEM-1 Response with CBCS Output Template	5.1.2	
CBCS-PEM1-RES-W-CBCSTEMPLATE-C-002-M	Support for action	5.1.2	
CBCS-PEM1-RES-W-CBCSTEMPLATE-C-003-O	Support for mode attribute	5.1.2	
CBCS-PEM1-RES-W-CBCSTEMPLATE-C-004-O	Support for action identifier attribute	5.1.2	

B.1.2 SCR for CBCS.PEM-1 Server

B.1.2.1 SCR for CBCS.PEM-1 Server Support

Item	Function	Reference	Requirement
CBCS-PEM1-SUPPORT-S-001-M	Support for the PEM-1 interface [PEM1-TS]	5.1.1	

B.1.2.2 SCR for CBCS.PEM-1 Server Request

Item	Function	Reference	Requirement
CBCS-PEM1-REQ-W-CBCSTEMPLATE-S-001-M	PEM-1 Request with CBCS Input Template	5.1.1	CBCS-PEM1-SUPPORT-S-001-M AND (CBCS-PEM1-REQ-W-CBCSTEMPLATE-S-002-O OR CBCS-PEM1-REQ-W-CBCSTEMPLATE-S-003-O)
CBCS-PEM1-REQ-W-CBCSTEMPLATE-S-002-O	Request with Content	5.1.1	
CBCS-PEM1-REQ-W-CBCSTEMPLATE-S-003-O	Request with Content reference	5.1.1	
CBCS-PEM1-REQ-W-CBCSTEMPLATE-S-004-O	Support for Content descriptor	5.1.1	
CBCS-PEM1-REQ-W-CBCSTEMPLATE-S-005-O	Support for user information	5.1.1	
CBCS-PEM1-REQ-W-CBCSTEMPLATE-S-006-O	Support for context information element	5.1.1	
CBCS-PEM1-REQ-W-CBCSTEMPLATE-S-007-O	Support for pre-categorization information	5.1.1	

B.1.2.3 SCR for CBCS.PEM-1 Server Response

Item	Function	Reference	Requirement
CBCS-PEM1-RES-W-CBCSTEMPLATE-S-001-M	PEM-1 Response with CBCS Output Template	5.1.2	
CBCS-PEM1-RES-W-CBCSTEMPLATE-S-002-M	Support for action	5.1.2	
CBCS-PEM1-RES-W-CBCSTEMPLATE-S-003-O	Support for mode attribute	5.1.2	
CBCS-PEM1-RES-W-CBCSTEMPLATE-S-004-O	Support for action identifier attribute	5.1.2	

B.2 SCR for CBCS.PEM-2

B.2.1 SCR for CBCS.PEM-2 Client

Item	Function	Reference	Requirement
CBCS-PEM2-	Support for the PEM-2	5.2	

Item	Function	Reference	Requirement
SUPPORT-C-001-M	interface [PEM2-TS]		

B.2.2 SCR for CBCS.PEM-2 Server

Item	Function	Reference	Requirement
CBCS-PEM2-SUPPORT-S-001-M	Support for the PEM-2 interface [PEM2-TS]	5.2	

B.3 SCR for CBCS-1

B.3.1 SCR for CBCS-1 Client

B.3.1.1 SCR for CBCS-1 Client Request-Response

Item	Function	Reference	Requirement
CBCS-CBCS1-C-REQ_RESP-001-M	Request-response	5.3.1	CBCS-CBCS1-REQ-C-001-M AND CBCS-CBCS1-RESP-C-001-M AND CBCS-CBCS1-ICAP-C-001-O

B.3.1.2 SCR for CBCS-1 Client Request

CBCS-CBCS1-REQ-C-001-M	Request	5.3.1	CBCS-CBCS1-REQ-C-007-M AND (CBCS-CBCS1-REQ-C-002-O OR CBCS-CBCS1-REQ-C-003-O)
CBCS-CBCS1-REQ-C-002-O	Request carrying Content	5.3.1	CBCS-CBCS1-REQ-C-004-M AND CBCS-CBCS1-REQ-C-005-O
CBCS-CBCS1-REQ-C-003-O	Request carrying Content reference	5.3.1	CBCS-CBCS1-REQ-C-004-M AND CBCS-CONTREF-C-001-O
CBCS-CBCS1-REQ-C-004-M	Support of Content descriptor	5.3.1	
CBCS-CBCS1-REQ-C-005-O	Support of Content	5.3.1	
CBCS-CBCS1-REQ-C-006-O	Support of filter	5.3.1	
CBCS-CBCS1-REQ-C-007-M	Request capabilities	5.3.1	

B.3.1.3 SCR for CBCS-1 Client Response

Item	Function	Reference	Requirement
CBCS-CBCS1-RESP-C-001-M	Response to CBCS-1 request	5.3.2	CBCS-CBCS1-RESP-C-002-M AND (CBCS-CBCS1-RESP-C-003-M OR CBCS-CBCS1-RESP-C-007-M)
CBCS-CBCS1-RESP-C-002-M	Support of status code	5.3.2	
CBCS-CBCS1-RESP-C-003-M	Support of Content Category vector	5.3.2	CBCS-CATEGORIES-C-001-M
CBCS-CBCS1-RESP-C-004-O	Support of region code	5.3.2	
CBCS-CBCS1-RESP-C-005-O	Support of signature	5.3.2	
CBCS-CBCS1-RESP-C-	Support of index	5.3.2	

Item	Function	Reference	Requirement
006-O			
CBCS-CBCS1-RESP-C-007-M	Support of capabilities	5.3.2	

B.3.1.4 B.4 SCR for CBCS-1 Client ICAP Protocol Binding

Item	Function	Reference	Requirement
CBCS-CBCS1-ICAP-C-001-O	ICAP protocol binding	5.4, 5.4.1, 5.4.2	CBCS-CBCS1-ICAP-C-002-O AND CBCS-CBCS1-ICAP-C-003-O AND CBCS-CBCS1-ICAP-C-004-O AND CBCS-CBCS1-ICAP-C-005-O
CBCS-CBCS1-ICAP-C-002-O	Use of the REQMOD and RESPMOD method	5.4.1	
CBCS-CBCS1-ICAP-C-003-O	When the request contains a HTTP request/reponse	5.4.1	
CBCS-CBCS1-ICAP-C-004-O	When the request does not contain HTTP request/response	5.4.1	
CBCS-CBCS1-ICAP-C-005-O	Transfer of the response	5.4.2	

B.3.1.5 SCR for Client Support of Content References

Item	Function	Reference	Requirement
CBCS-CONTREF-C-001-O	Support of Content reference	5.3.1	CBCS-CONTREF-C-002-O OR CBCS-CONTREF-C-003-O OR CBCS-CONTREF-C-004-O
CBCS-CONTREF-C-002-O	Support of Content locator	5.3.1	
CBCS-CONTREF-C-003-O	Support of Content identifier	5.3.1	
CBCS-CONTREF-C-004-O	Support of Content digest	5.3.1	

B.3.1.6 SCR for Client Support of Content Categories

Item	Function	Reference	Requirement
CBCS-CATEGORIES-C-001-M	Support for content categories	5.3.2	CBCS-CATEGORIES-C-003-O
CBCS-CATEGORIES-C-002-O	Support of categorization scheme identifier	5.3.1	
CBCS-CATEGORIES-C-003-O	Support for particular categorization schemes	5.3.1	

B.3.2 SCR for CBCS-1 Server

B.3.2.1 SCR for CBCS-1 Server Request-Response

Item	Function	Reference	Requirement
CBCS-CBCS1-S-REQ_RESP-001-M	Request-response	5.3.1	CBCS-CBCS1-REQ-S-001-M AND CBCS-CBCS1-RESP-S-001-M AND CBCS-CBCS1-ICAP-S-001-O

B.3.2.2 SCR for CBCS-1 Server Request

Item	Function	Reference	Requirement
CBCS-CBCS1-REQ-S-001-M	Request	5.3.1	CBCS-CBCS1-REQ-S-007-M AND (CBCS-CBCS1-REQ-S-002-O OR CBCS-CBCS1-REQ-S-003-O)
CBCS-CBCS1-REQ-S-002-O	Request carrying Content	5.3.1	CBCS-CBCS1-REQ-S-004-M AND CBCS-CBCS1-REQ-S-005-O
CBCS-CBCS1-REQ-S-003-O	Request carrying Content reference	5.3.1	CBCS-CBCS1-REQ-S-004-M AND CBCS-CONTREF-S-001-O
CBCS-CBCS1-REQ-S-004-M	Support of Content descriptor	5.3.1	
CBCS-CBCS1-REQ-S-005-O	Support of Content	5.3.1	
CBCS-CBCS1-REQ-S-006-O	Support of filter	5.3.1	
CBCS-CBCS1-REQ-S-007-M	Request capabilities	5.3.1	

B.3.2.3 SCR for CBCS-1 Server Response

Item	Function	Reference	Requirement
CBCS-CBCS1-RESP-S-001-M	Response to CBCS-1 request	5.3.2	CBCS-CBCS1-RESP-S-002-M AND (CBCS-CBCS1-RESP-S-003-M OR CBCS-CBCS1-RESP-S-007-M)
CBCS-CBCS1-RESP-S-002-M	Support of status code	5.3.2	
CBCS-CBCS1-RESP-S-003-M	Support of Content Category vector	5.3.2	CBCS-CATEGORIES-S-001-M
CBCS-CBCS1-RESP-S-004-O	Support of region code	5.3.2	
CBCS-CBCS1-RESP-S-005-O	Support of signature	5.3.2	
CBCS-CBCS1-RESP-S-006-O	Support of index	5.3.2	
CBCS-CBCS1-RESP-S-007-M	Support of capabilities	5.3.2	

B.3.2.4 SCR for CBCS-1 Server ICAP Protocol Binding

Item	Function	Reference	Requirement
CBCS-CBCS1-ICAP-S-001-O	ICAP protocol binding	5.4, 5.4.1, 5.4.2	CBCS-CBCS1-ICAP-S-002-O AND CBCS-CBCS1-ICAP-S-003-O AND CBCS-CBCS1-ICAP-S-004-O AND CBCS-CBCS1-ICAP-S-005-O
CBCS-CBCS1-ICAP-S-002-O	Use of the REQMOD and RESPMOD method	5.4.1	
CBCS-CBCS1-ICAP-S-003-O	When the request contains a HTTP request/reponse	5.4.1	
CBCS-CBCS1-ICAP-S-004-O	When the request does not contain HTTP request/response	5.4.1	

Item	Function	Reference	Requirement
CBCS-CBCS1-ICAP-S-005-O	Transfer of the response	5.4.2	

B.3.2.5 SCR for Server Support of Content References

Item	Function	Reference	Requirement
CBCS-CONTREF-S-001-O	Support of Content reference	5.3.1	CBCS-CONTREF-S-002-O OR CBCS-CONTREF-S-003-O OR CBCS-CONTREF-S-004-O
CBCS-CONTREF-S-002-O	Support of Content locator	5.3.1	
CBCS-CONTREF-S-003-O	Support of Content identifier	5.3.1	
CBCS-CONTREF-S-004-O	Support of Content digest	5.3.1	

B.3.2.6 B.12 SCR for Server Support of Content Categories

Item	Function	Reference	Requirement
CBCS-CATEGORIES-S-001-M	Support for content categories	5.3.2	CBCS-CATEGORIES-S-002-O
CBCS-CATEGORIES-S-002-O	Support of categorization scheme identifier	5.3.1	
CBCS-CATEGORIES-S-003-M	Support for particular categorization schemes	5.3.1	

B.4 SCR for CBCS-2

B.4.1 SCR for CBCS-2 Client

Item	Function	Reference	Requirement
CBCS-CBCS2-C-001-M	Support for the PEM-2 interface [PEM2-TS]	5.5	

B.4.2 SCR for CBCS-2 Server

Item	Function	Reference	Requirement
CBCS-CBCS2-S-001-M	Support for the PEM-2 interface [PEM2-TS]	5.5	

B.5 SCR for CBCS-3

B.5.1 SCR for CBCS-3 Client

B.5.1.1 SCR for CBCS-3 Client Operations

Item	Function	Reference	Requirement
CBCS-CBCS3-CLIENT-C-001-M	Support of Associating Content references with Content Categories	5.6	
CBCS-CBCS3-CLIENT-C-002-M	Support of List operation	5.6.1	CBCS-CBCS3-LISTCONTREFTYPE-C-001-M AND CBCS-CBCS3-LISTScheme-C-001-M AND CBCS-CBCS3-LISTCATEGORY-C-001-M AND CBCS-CBCS3-

Item	Function	Reference	Requirement
			ICAP-C-001-O
CBCS-CBCS3-CLIENT-C-003-M	Support of add and remove operation	5.6.1	CBCS-CBCS3-ADDREMOP-C-001-M AND CBCS-CBCS3-ICAP-C-001-O
CBCS-CBCS3-CLIENT-C-004-M	Support of capabilities operation	5.6.1	CBCS-CBCS3-CAP-C-001-M AND CBCS-CBCS3-ICAP-C-001-O

B.5.1.2 SCR for CBCS-3 Client Capabilities operation

Item	Function	Reference	Requirement
CBCS-CBCS3-CAP-C-001-M	Support of Capabilities operation	5.6.1	
CBCS-CBCS3-CAP-C-002-M	Support of response to Capabilities operation	5.6.1	

B.5.1.3 SCR for CBCS-3 Client List Content references

Item	Function	Reference	Requirement
CBCS-CBCS3-LISTCONTREFTYPE-C-001-M	Support of listing content references of a particular type	5.6.1.1	CBCS-CBCS3-LISTCONTREFTYPE-C-002-O AND (CBCS-CBCS3-LISTCONTREFTYPE-C-003-O) OR CBCS-CBCS3-LISTCONTREFTYPE-C-004-O OR CBCS-CBCS3-LISTCONTREFTYPE-C-005-O)
CBCS-CBCS3-LISTCONTREFTYPE-C-002-O	Support of response to list request with a particular content reference type	5.6.1.1	
CBCS-CBCS3-LISTCONTREFTYPE-C-003-O	Support of listing content references of type content locator	5.6.1.1	
CBCS-CBCS3-LISTCONTREFTYPE-C-004-O	Support of listing content references of type content identifier	5.6.1.1	
CBCS-CBCS3-LISTCONTREFTYPE-C-005-O	Support of listing content references of type content digest	5.6.1.1	

B.5.1.4 SCR for CBCS-3 Client List Categorization schemes

Item	Function	Reference	Requirement
CBCS-CBCS3-LISTSCHEME-C-001-M	Support of listing the supported Categorization Schemes	5.6.1.1	
CBCS-CBCS3-LISTSCHEME-C-002-M	Support of response to request to list the supported Categorization schemes	5.6.1.1	

B.5.1.5 SCR for CBCS-3 Client List Categories

Item	Function	Reference	Requirement
CBCS-CBCS3-LISTCATEGORY-C-001-	Support of listing the supported Content	5.6.1.1	

Item	Function	Reference	Requirement
M	Categories of a scheme		
CBCS-CBCS3- LISTCATEGORY-C-001- M	Support of response to request to list the supported Content Categories of a scheme	5.6.1.1	

B.5.1.6 SCR for CBCS-3 Client Add/Remove operation

Item	Function	Reference	Requirement
CBCS-CBCS3- ADDREMOP-C-001-M	Support Add/remove operation	5.6.1.2	(CBCS-CBCS3- ADDREMScheme-C-001-O AND CBCS-CBCS3- ADDREMASSOCCATScheme- C-001-O AND CBCS-CBCS3- ADDREMASSOCCONTREFCAT- C-001-O) OR (CBCS-CBCS3- ADDREMScheme-C-001-O AND CBCS-CBCS3- ADDREMASSOCCATScheme- C-001-O AND CBCS-CBCS3- ADDREMONTREF-C-001-O)
CBCS-CBCS3- ADDREMOP-C-001-O	Include list in response	5.6.1.2	

B.5.1.7 SCR for CBCS-3 Client Add/Remove Categorization Schemes

Item	Function	Reference	Requirement
CBCS-CBCS3- ADDREMScheme-C-001- O	Support Adding/removing supported Categorization Schemes	5.6.1.2	
CBCS-CBCS3- ADDREMScheme-C-002- O	Support of response to request to add/remove supported Categorization schemes	5.6.1.2	

B.5.1.8 SCR for CBCS-3 Client Add/Remove Categories

Item	Function	Reference	Requirement
CBCS-CBCS3- ADDREMASSOCCATSCH EME-C-001-O	Support adding/removing Content Categories to a scheme	5.6.1.2	
CBCS-CBCS3- ADDREMASSOCCATSCH EME-C-001-O	Support the response to request to add/remove supported Content Categories to a scheme	5.6.1.2	

B.5.1.9 SCR for CBCS-3 Client Add/Remove Association of Content Reference with Content Category

Item	Function	Reference	Requirement
CBCS-CBCS3- ADDREMASSOCCONTRE FCAT-C-001-O	Support adding/removing associations of a Content reference with a Content Category	5.6.1.2	CBCS-CBCS3- ADDREMONTREFTYPE-C-001- O

Item	Function	Reference	Requirement
CBCS-CBCS3-ADDREMASSOCCONTREFCAT-C-001-O	Support the response to request to add/remove associations of a Content reference with a Content Category	5.6.1.2	CBCS-CBCS3-ADDREMCONTREFTYPE-C-002-O

B.5.1.10 SCR for CBCS-3 Client Add/Remove Content Reference

Item	Function	Reference	Requirement
CBCS-CBCS3-ADDREMCONTREF-C-001-O	Support adding/removing Content reference	5.6.1.2	CBCS-CBCS3-ADDREMCONTREFTYPE-C-001-O
CBCS-CBCS3-ADDREMCONTREF-C-001-O	Support the response to request to add/remove Content reference	5.6.1.2	CBCS-CBCS3-ADDREMCONTREFTYPE-C-002-O

B.5.1.11 SCR for CBCS-3 Client Content Reference Type

Item	Function	Reference	Requirement
CBCS-CBCS3-ADDREMCONTREFTYPE-C-001-O	Support request to add/remove Content references of a particular type or Content Category associations with a Content reference of a particular type	5.6.1.2	CBCS-CBCS3-ADDREMCONTREFTYPE-C-003-O OR CBCS-CBCS3-ADDREMCONTREFTYPE-C-004-O OR CBCS-CBCS3-ADDREMCONTREFTYPE-C-005-O
CBCS-CBCS3-ADDREMCONTREFTYPE-C-002-O	Support response to request to add/remove Content references of a particular type or Content Category associations with a Content reference of a particular type	5.6.1.2	
CBCS-CBCS3-ADDREMCONTREFTYPE-C-003-O	Support adding/removing content references or associations with these of type content locator	5.6.1.2	
CBCS-CBCS3-ADDREMCONTREFTYPE-C-004-O	Support adding/removing content references or associations with these of type content identifier	5.6.1.2	
CBCS-CBCS3-ADDREMCONTREFTYPE-C-005-O	Support adding/removing content references or associations with these of type content digest	5.6.1.2	

B.5.1.12 SCR for CBCS-3 Client ICAP Protocol Binding

Item	Function	Reference	Requirement
CBCS-CBCS3-ICAP-C-001-O	Support the ICAP protocol binding	5.7	
CBCS-CBCS3-ICAP-C-002-O	Use of the OPTIONS method	5.7.1	

Item	Function	Reference	Requirement
CBCS-CBCS3-ICAP-C-003-O	List request/response binding to ICAP	5.7.1.1	
CBCS-CBCS3-ICAP-C-004-O	Add/remove request/response binding to ICAP	5.7.1.2	

B.5.2 SCR for CBCS-3 Server

B.5.2.1 SCR for CBCS-3 Server Operations

Item	Function	Reference	Requirement
CBCS-CBCS3-SERVER-S-001-M	Support associating Content references with Content Categories	5.6	
CBCS-CBCS3-SERVER-S-002-M	Support of List operation	5.6.1	CBCS-CBCS3-LISTCONTREFTYPE-S-001-M AND CBCS-CBCS3-LISTScheme-S-001-M AND CBCS-CBCS3-LISTCATEGORY-S-001-M AND CBCS-CBCS3-ICAP-S-001-O
CBCS-CBCS3-SERVER-S-003-M	Support of Add and remove operation	5.6.1	CBCS-CBCS3-ADDREMOP-S-001-M AND CBCS-CBCS3-ICAP-S-001-O
CBCS-CBCS3-SERVER-S-004-M	Support of capabilities operation	5.6.1	CBCS-CBCS3-CAP-S-001-M AND CBCS-CBCS3-ICAP-S-001-O

B.5.2.2 SCR for CBCS-3 Server Capabilities operation

Item	Function	Reference	Requirement
CBCS-CBCS3-CAP-S-001-M	Support of Capabilities operation	5.6.1	
CBCS-CBCS3-CAP-S-002-M	Support of response to Capabilities operation	5.6.1	

B.5.2.3 SCR for CBCS-3 Server List Content references

Item	Function	Reference	Requirement
CBCS-CBCS3-LISTCONTREFTYPE-S-001-M	Support of listing content references of a particular type	5.6.1.1	CBCS-CBCS3-LISTCONTREFTYPE-S-002-O AND (CBCS-CBCS3-LISTCONTREFTYPE-S-003-O) OR CBCS-CBCS3-LISTCONTREFTYPE-S-004-O OR CBCS-CBCS3-LISTCONTREFTYPE-S-005-O)
CBCS-CBCS3-LISTCONTREFTYPE-S-002-O	Support of the response to list request with a particular content reference type	5.6.1.1	
CBCS-CBCS3-LISTCONTREFTYPE-S-003-O	Support of listing content references of type content locator	5.6.1.1	
CBCS-CBCS3-LISTCONTREFTYPE-S-	Support of listing content references of	5.6.1.1	

Item	Function	Reference	Requirement
004-O	type content identifier		
CBCS-CBCS3-LISTCONTREFTYPE-S-005-O	Support of listing content references of type content digest	5.6.1.1	

B.5.2.4 SCR for CBCS-3 Server List Categorization schemes

Item	Function	Reference	Requirement
CBCS-CBCS3-LISTScheme-S-001-M	Support listing of the supported Categorization Schemes	5.6.1.1	
CBCS-CBCS3-LISTScheme-S-002-M	Support of the response to the request to list supported Categorization schemes	5.6.1.1	

B.5.2.5 SCR for CBCS-3 Server List Categories

Item	Function	Reference	Requirement
CBCS-CBCS3-LISTCATEGORY-S-001-M	Support of listing supported Content Categories of a scheme	5.6.1.1	
CBCS-CBCS3-LISTCATEGORY-S-001-M	Support of the response to the request to list supported Content Categories of a scheme	5.6.1.1	

B.5.2.6 SCR for CBCS-3 Server Add/Remove operation

Item	Function	Reference	Requirement
CBCS-CBCS3-ADDREMOP-S-001-M	Support Add/remove operation	5.6.1.2	(CBCS-CBCS3-ADDREMScheme-S-001-O AND CBCS-CBCS3-ADDREMASSOCCATScheme-S-001-O AND CBCS-CBCS3-ADDREMASSOCCONTREFCAT-S-001-O) OR (CBCS-CBCS3-ADDREMScheme-S-001-O AND CBCS-CBCS3-ADDREMASSOCCATScheme-S-001-O AND CBCS-CBCS3-ADDREMCONTREF-S-001-O)
CBCS-CBCS3-ADDREMOP-C-001-O	Support to include the list in the response	5.6.1.2	

B.5.2.7 SCR for CBCS-3 Server Add/Remove Categorization Schemes

Item	Function	Reference	Requirement
CBCS-CBCS3-ADDREMScheme-S-001-O	Support of adding/removing supported Categorization Schemes	5.6.1.2	
CBCS-CBCS3-ADDREMScheme-S-002-O	Support of the response to the request to add/remove supported Categorization schemes	5.6.1.2	

B.5.2.8 SCR for CBCS-3 Client Add/Remove Categories

Item	Function	Reference	Requirement
CBCS-CBCS3-ADDREMASSOCCATSCH EME-S-001-O	Support of adding/removing Content Categories to a scheme	5.6.1.2	
CBCS-CBCS3-ADDREMASSOCCATSCH EME-S-001-O	Support of the response to the request to add/remove supported Content Categories to a scheme	5.6.1.2	

B.5.2.9 SCR for CBCS-3 Server Add/Remove Association of Content Reference with Content Category

Item	Function	Reference	Requirement
CBCS-CBCS3-ADDREMASSOCCONTREFCAT-S-001-O	Support of adding/removing associations of a Content reference with a Content Category	5.6.1.2	CBCS-CBCS3-ADDREMCONTREFTYPE-S-001-O
CBCS-CBCS3-ADDREMASSOCCONTREFCAT-S-001-O	Support of the response to the request to add/remove associations of a Content reference with a Content Category	5.6.1.2	CBCS-CBCS3-ADDREMCONTREFTYPE-S-002-O

B.5.2.10 SCR for CBCS-3 Server Add/Remove Content Reference

Item	Function	Reference	Requirement
CBCS-CBCS3-ADDREMCONTREF-S-001-O	Support of adding/removing Content reference	5.6.1.2	CBCS-CBCS3-ADDREMCONTREFTYPE-S-001-O
CBCS-CBCS3-ADDREMCONTREF-S-001-O	Support of the response to the request to add/remove Content reference	5.6.1.2	CBCS-CBCS3-ADDREMCONTREFTYPE-S-002-O

B.5.2.11 SCR for CBCS-3 Server Content Reference Type

Item	Function	Reference	Requirement
CBCS-CBCS3-ADDREMCONTREFTYPE-S-001-O	Support request to add/remove Content references of a particular type or Content Category associations with a Content reference of a particular type	5.6.1.2	CBCS-CBCS3-ADDREMCONTREFTYPE-S-003-O OR CBCS-CBCS3-ADDREMCONTREFTYPE-S-004-O OR CBCS-CBCS3-ADDREMCONTREFTYPE-S-005-O
CBCS-CBCS3-ADDREMCONTREFTYPE-S-002-O	Support response to request to add/remove Content references of a particular type or Content Category associations with a Content reference of a particular type	5.6.1.2	
CBCS-CBCS3-	Support of	5.6.1.2	

Item	Function	Reference	Requirement
ADDREMCNTREFTYPE-S-003-O	adding/removing content references or associations with these of type content locator		
CBCS-CBCS3-ADDREMCNTREFTYPE-S-004-O	Support of adding/removing content references or associations with these of type content identifier	5.6.1.2	
CBCS-CBCS3-ADDREMCNTREFTYPE-S-005-O	Support of adding/removing content references or associations with these of type content digest	5.6.1.2	

B.5.2.12 SCR for CBCS-3 Server ICAP Protocol Binding

Item	Function	Reference	Requirement
CBCS-CBCS3-ICAP-S-001-O	Support of the ICAP protocol binding	5.7	
CBCS-CBCS3-ICAP-S-002-O	Use of the OPTIONS method	5.7.1	
CBCS-CBCS3-ICAP-S-003-O	List request/response binding to ICAP	5.7.1.1	
CBCS-CBCS3-ICAP-S-004-O	Add/remove request/response binding to ICAP	5.7.1.2	

Appendix C. Categorization Schemes Supported by CBCS (Normative)

This normative appendix species the semantics of the `content-category` element (section 5.3.2) for supported Content categorization schemes. Table 5 below should be interpreted as follows: “if the `categorization-scheme-identifier` is as displayed in the left column, then the syntax of the `content-category-value` MUST have the syntax specified in the right column”.

Table 5 (normative) Semantics of the `content-category-value` element for supported Content categorization schemes

categorization-scheme-identifier	content-category-value
ESRB	<pre> content-category-value =/ ESRB-rating [ESRB-content-descriptor] ESRB-rating = "EC" ; Early Childhood / "E" ; Everybody / "E10+" ; Everybody 10 and over / "T" ; Teenagers (13 and over) / "M" ; Mature (17 and over) / "AO" ; Adults Only / "RP" ; Rating Pending ESRB-content-descriptor = "Alcohol Reference" /"Animated Blood" /"Blood" /"Blood and Gore" /"Cartoon Violence" /"Comic Mischief" /"Crude Humor" /"Drug Reference" /"Fantasy" /"Intense Violence" /"Language" /"Lyrics" /"Mature Humor" /"Nudity" /"Partial Nudity" /"Real Gambling" /"Sexual Content" /"Sexual Themes" /"Sexual Violence" /"Simulated Gambling" /"Strong Language" /"Strong Lyrics" /"Strong Sexual Content" /"Suggestive Themes" /"Tobacco Reference" /"Use of Drugs" /"Use of Alcohol" </pre>

	<p>/"Use of Tobacco" /"Violence" /"Violent References"</p>
<p>ICRA</p>	<pre> content-category-value =/ 1* ICRA-label ICRA-label = "na 1" ; Exposed breasts / "nb 1" ; Bare buttocks / "nc 1" ; Visible genitals / "nz 0" ; Nudity may be present / "nz 1" ; No nudity / "sa 1" ; Passionate kissing / "sb 1" ; Obscured or implied sexual acts / "sc 1" ; Visible sexual touching / "sd 1" ; Explicit sexual language / "se 1" ; Erections/explicit sexual acts / "sf 1" ; Erotica / "sz 0" ; Sexual material may be present / "sz 1" ; No sexual material / "va 1" ; Assault/rape / "vb 1" ; Injury to human beings / "vc 1" ; Injury to animals / "vd 1" ; Injury to fantasy characters / "ve 1" ; Blood and dismemberment, human beings / "vf 1" ; Blood and dismemberment, animals / "vg 1" ; Blood and dismemberment, fantasy / "vh 1" ; Torture or killing of human beings / "vi 1" ; Torture or killing of animals / "vj 1" ; Torture or killing of fantasy / "vz 0" ; Violence may be present / "vz 1" ; No violence / "la 1" ; Abusive or vulgar terms / "lb 1" ; Profanity or swearing / "lc 1" ; Mild expletives / "lz 0" ; Potentially offensive language / "lz 1" ; No potentially offensive language / "oa 1" ; Depiction of tobacco use / "ob 1" ; Depiction of alcohol use / "oc 1" ; Depiction of drug use / "od 1" ; Depiction of the use of weapons / "oe 1" ; Gambling / "of 1" ; Bad example for young children / "og 1" ; Fear, intimidation, horror / "oh 1" ; Discrimination or hate incitement / "oz 0" ; Potentially harmful activities / "oz 1" ; No potentially harmful activities / "ca 1" ; User-generated content (moderated) / "cb 1" ; User-generated content (unmoderated) </pre>

	<pre> / "cz 0" ; User generated content may be present / "cz 1" ; No user-generated content / "xa 1" ; Artistic context / "xb 1" ; Educational context / "xc 1" ; Medical context / "xd 1" ; Sports context / "xe 1" ; News context </pre>
MPAA	<pre> content-category-value =/ "G" ; General Audiences / "PG" ; Parental Guidance / "PG-13" ; Parents strongly cautioned / "R" ; Restricted / "NC-17" ; No one 17 and under admitted </pre>
MRA	<pre> content-category-value =/ 2 DIGIT </pre>
PEGI	<pre> content-category-value =/ PEGI-rating [PEGI-descriptor] PEGI-rating = *2 DIGIT PEGI-descriptor = "Bad language" / "Discrimination" / "Drugs" / "Fear" / "Gambling" / "Sex" / "Violence" </pre>
RIAA	<pre> content-category-value =/ "" / "Parental advisory" </pre>

Appendix D. CBCS-1: ICAP Assessment (Informative)

Table 6 below presents an analysis to what extent ICAP fulfills CBCS-1 needs.

The assessment criteria (left hand column) have been derived from [CBCS-RD] and [CBCS-AD].

The right hand column describes how ICAP fulfills the assessment criterion.

Table 6 (informative) How ICAP matches CBCS-1 criteria

CBCS-1 Assessment Criterion	ICAP fulfillment
Reuse an existing standard if possible, as per OMA principles to rather reuse instead of re-invent	Supported: ICAP has been around since 2003.
Is endorsed by content categorization industry	Supported: check out the members of the ICAP forum (www.icap-forum.org) and the registered implementations.
The CBCS content screening component must be able to retrieve categorization information from the content categorization entity.	Supported: this is fulfilled by the ICAP response message [RFC3507] combined with the ICAP-forum endorsed extension for content classification.
The CBCS enabler may use information from multiple content categorization entities	Supported, ICAP supports a parameter for the ICAP server name (i.e. content categorization server). The specification does not limit the ICAP client to interact with multiple servers.
It may be possible to send multiple request/response pairs over a single connection to/from a content categorization entity	Supported, as described per section 4 of [RFC3507]
The interaction with the content categorization entity shall be lightweight	Supported, the ICAP specification characterizes ICAP as: <i>"ICAP is, in essence, a lightweight protocol for executing a "remote procedure call"</i>
The request for content categorization can contain the reference to content (e.g. URI) or the content itself or the content plus associated metadata	Supported: Practically any content can be sent to the Content Categorization Component: <ul style="list-style-type: none"> • By means of the RESPMOD message, using the ICAP chunked-encoding, for any content that is not an HTTP request or response; • By means of the REQMOD message for HTTP request content and the RESPMOD message for HTTP response content (the latter using ICAP chunked-encoding)

Appendix E. Examples of CBCS-1 Requests and Replies over ICAP (Informative)

This appendix describes some examples of the use of ICAP to transport requests and replies on the CBCS-1 interface. These examples have an informative status and serve only to illustrate the normative ICAP representation of CBCS-1 as defined in section 5.3.

E.1 Categorization Request for a HTTP Request

When the CBCS-1 request for categorization contains an HTTP request that is to be categorized, then the REQMOD method as specified per ICAP [RFC3507] shall be used to encapsulate the HTTP request.

An example of request for categorization containing an HTTP request, encapsulated in an ICAP REQMOD message:

ICAP Message	Description
REQMOD icap://icap-server.net/ContentCategorizationSvc ICAP/1.0	Request Line
Host: icap-server.net	Request Header
X-Filter: ESRB, MRA	Request Header
Encapsulated: req-hdr=0; null-body=102	
get / HTTP/1.1	Encapsulated HTTP Request
Host: www.cnn.com	''
Accept: text/html, text/plain	''
Accept-Encoding: compress	''
Pragma: no-cache	''

Note that this example shows the use of the OPTIONAL X-Filter extension header, which the requestor can use to indicate it is only interested in receiving Content Categories according to specific categorization schemes.

E.2 Categorization Request for a HTTP Response

When the CBCS-1 request for categorization contains an HTTP response that contains data that is to be categorized, then the RESPMOD message including chunk-encoding as specified per ICAP [RFC3507] shall be used to encapsulate the HTTP response.

An example of a request for categorization containing an HTTP response, encapsulated in a RESPMOD message:

ICAP Message	Description
RESPMOD icap://icap-server.net/ContentCategorizationSvc ICAP/1.0	Request Line
Host: icap-server.net	Request Header
Encapsulated: res-hdr=0, res-body=39	Request Header
HTTP/1.1 200 OK	Encapsulated HTTP response header(s)
Content-Type: image/jpeg	
3ef7	
<chunk of 16119 octets>	Chunk encoding of HTTP response body
0	

Note that this example shows how the encapsulated HTTP response header includes a content descriptor in the form of a HTTP Content-Type entity header.

E.3 Categorization Request for SMS

When the CBCS-1 request for categorization contains Content other than HTTP requests or responses, then the ICAP RESPMOD method will be used and the content will be chunk-encoded as specified per ICAP [RFC3507].

The following example shows how a CBCS-1 request represented in ICAP carries a short message (SMS) with the text “Having an excellent time in Ibiza!”:

ICAP Message	Description
<pre>RESPMOD icap://icap-server.net/ContentCategorizationSvc ICAP/1.0 Host: icap-server.net Trailer : Content-Type Encapsulated: res-body=0 3a Content-Type: text/plain Having an excellent time in Ibiza! 0</pre>	<p>Request Line</p> <p>Request Header</p> <p>Request Header</p> <p>Request Header</p> <p>Chunk encoded SMS with entity header that represents the content descriptor</p>

There are a few points to note in this example:

- As the Content is not HTTP Content, the ICAP message only contains an encapsulated body, and no encapsulated (HTTP) headers. Note that ICAP allows the encapsulated headers to be void in a RESPMOD method.
- Assuming that the original CBCS-1 request contained a content-descriptor element with value “Content-Type: text/plain”, the corresponding ICAP message transports this as the first line in the chunk encoded encapsulated body, as shown in the example.

E.4 Categorization Request with Content reference

A CBCS-1 request may contain a Content reference, rather than the Content itself. The following is an example of an ICAP request corresponding to a CBCS-1 request with a Content digest (i.e. a binary value computed on the Content, which can serve to reference the Content, for example if it is cached):

ICAP Message	Description
<pre>RESPMOD icap://icap-server.net/ContentCategorizationSvc ICAP/1.0 Host: icap-server.net X-Content-Descriptor: content digest Encapsulated: res-body=0 3 MD5 20 40555161d127e31e1e8cabb7a073c638 0</pre>	<p>Request Line</p> <p>Request Header</p> <p>Request Header</p> <p>Request Header</p> <p>Chunk encoded content digest. The first chunk holds the digest type, the second chunk holds the digest value</p>

The `content-digest` element of the CBCS-1 request is a two field structure, which is mapped onto two chunks. The first represents the content digest type (MD5), the second represents the content digest value (a 32 character HEX number). Note also that this example shows the use of the `X-Content-Descriptor` extension header to hold a content descriptor (“content digest”) that can’t be expressed as standard entity header.

E.5 Categorization Request for SMS Shortcode

SMS shortcodes are often used to subscribe to premium Content services. A message to an SMS short code often carries a single keyword (for example “SUBSCRIBE”). Taking this keyword at face value and categorizing it is meaningless; what should be categorized is the Content service requested by this keyword. For this reason, SMS shortcodes have to be considered as Content references, rather than as Content in their own right.

The following example shows how a CBCS-1 request to screen a message to SMS shortcode 1234 with message “SUBSCRIBE” is represented in ICAP:

ICAP Message	Description
RESPMOD icap://icap-server.net/ContentCategorizationSvc ICAP/1.0	Request Line
Host: icap-server.net	Request Header
X-Content-Descriptor: content locator	Request Header
Encapsulated: res-body=0	Request Header
d	Chunk encoded SMS body.
SMS shortcode	The first chunk holds the content locator type.
e	
1234 SUBSCRIBE	The second chunk holds the SMS shortcode and keyword.
0	

As in the example of appendix E3, the `X-Content-Descriptor` extension header is used to tell the Content Categorization Component that the payload carries a content locator. Content locators are always encapsulated in two chunks. The first chunk contains the locator type (“SMS shortcode”). The second chunk contains the locator value, which is a combination of the shortcode (“1234”) followed by the message (the keyword “SUBSCRIBE”).

E.6 CBCS-1 Response Expressed as ICAP Response

So far, appendices F1-F5 have only shown examples of requests. This section will provide two examples of a CBCS-1 response, expressed as ICAP responses. The first example is a plain response:

ICAP Message	Description
ICAP/1.0 200 OK	Status Line
Server: ICAP-Server-Software/1.0	Response Header
Date: Tue Sep 25 11:24:15 2007 GMT	Response Header
X-Response-Desc: categorized	Response Header
X-Attribute: ESRB M Strong Language ES, MRA 17 NL	Response Header: Content Categories
Encapsulated: null-body=0	Response Header
	Null Body

The Content Categories are returned in the `X-Attribute` extension header, in comma separated format (this is necessary to comply with [ICAP-Extensions]). In this example, two Content Categories are returned. The first is an ESRB category

with rating “M” and descriptor “Strong Language”. In addition the two characters “ES” indicate that this category is valid in Spain. The second Content Category is a minimum recommended age (MRA) of 17, valid in The Netherlands.

Note that the ICAP response does not carry any Content as the CBCS-1 response does not require the Content to be returned. The ICAP response therefore returns a null body.

The second example is similar to the above, except that the server returns an index in the encapsulated body:

ICAP Message	Description
<pre> ICAP/1.0 200 OK Server: ICAP-Server-Software/1.0 Date: Tue Sep 25 11:24:15 2007 GMT X-Response-Desc: categorized with index X-Attribute: ESRB M Strong Language ES, MRA 17 NL Encapsulated: res-body=0 3 MD5 20 71b93ef9ea687f19ebf2ae4a036d0d7f 0 </pre>	<pre> Status Line Response Header Response Header Response Header Response Header: Content Categories Response Header Chunk encoded body which contains the CBCS-1 response index, in this case an MD5 digest </pre>

In this case the index is an MD5 digest computed on the original content, chunk-encoded in the same way as in section E4 above. The index may be used in future categorization requests assuming that the Content Categorization Component has cached the Content Categories with this index.

This can significantly improve the performance of the CBCS-1 protocol as it avoids the full Content having to be sent in every request, if the Content Categorization Component is capable of caching categorization results.

E.7 CBCS-1 Error in ICAP

It is also possible that an error occurs following a CBCS-1 request. In this case the ICAP response code will indicate the type of error and the OPTIONAL X-Response-Desc extension header can provide additional information:

ICAP Message	Description
<pre> ICAP/1.0 400 Bad request Server: ICAP-Server-Software/1.0 Date: Thu Nov 15 12:50:15 2007 GMT Connection: close X-Response-Desc: Unable to resolve content reference Encapsulated: null-body=0 </pre>	<pre> Status Line: error Response Header Response Header Response Header Response Header Response Header Response Header Null Body </pre>

In this case the ICAP status code notifies the requestor that it sent a bad request. The OPTIONAL X-Response-Desc extension header provides additional CBCS-1 specific information to the requestor, for example that it was unable to resolve the content reference sent in the request.

Appendix F. On the Use of Metadata Schemes in CBCS (Informative)

This appendix provides informative guidelines on the use of specific existing metadata schemes in CBCS for representing Pre-Categorized Content.

F.1 On the Use of MPEG-7 in CBCS

This informative section provides some detail on how MPEG-7 can be used to represent Pre-Categorized Content as defined in the CBCS Requirements Document.

Part 5 of the MPEG-7 standard [MPEG-7] defines the Classification Description Scheme (DS), and describes it as follows: *“The Classification DS describes the classification of the multimedia content. The resulting descriptions facilitate searching and filtering of multimedia content based on user preferences (e.g, language, style, genre, and o forth) and service-oriented classifications (e.g, purpose, parental guidance, market segmentation, media review, and so forth).”*

The Classification DS part that can be used to express the Content Categories part of Pre-Categorized Content is the `parentalGuidanceType` which defines complex data for the form:

```
<ParentalGuidance>
  <ParentalRating href="urn:mpeg:mpeg7:cs:MPAAParentalRatingCS:2001">
    <Name>PG-13</Name>
  </ParentalRating>
  <Region>us</Region>
</ParentalGuidance>
```

A parental guidance description always starts with an element `<ParentalGuidance>`. The children of this element are `<ParentalRating>` elements, which have an URI attribute referring to the MPEG-7 Classification Scheme (CS) that defines the rating terms. MPEG-7 defines Classification Schemes for various Content rating schemes, including RIAA, MPAA and ICRA.

The `<ParentalRating>` element in turn has a `<Name>` child element which contains the rating value in textual format. The `<Region>` sub-element of the `<ParentalGuidance>` element is OPTIONAL, and may be used to specify one or more regions where the rating scheme applies.

Part 5 of the MPEG-7 standard [MPEG-7] also defines the Availability Description Scheme (DS), and describes it as follows: *“The Availability DS describes the availability for use (broadcasting, on demand delivery, CD sales, and so forth) of the multimedia content.”*

The Availability DS part that can be used to express the Content Provider’s identity part of Pre-Categorized Content is the `DisseminationType`. This type can express several dissemination related properties of the Content, such as the format and region. Within the `DisseminationType`, the `<Disseminator>` element identifies the organization that makes the Content available.

The `<Disseminator>` element is of type `MediaAgentType`, which describes the agents involved in handling multimedia and their roles. An element of type `DisseminationType` then has the following form:

```
<Dissemination>
  <source>
    <!-- optional technical source information, e.g. a streaming server -->
  </source>
  <format>
    <!-- optional format information, e.g. CD-ROM -->
  </format>
  <Disseminator>
    <Role href="urn:mpeg:mpeg-7:cs:RoleCS:2001:PUBLISHER"/>
    <Agent xsi:type="OrganizationType">
      <Name>Discovery Channel</Name>
    </Agent>
  </Disseminator>
  <location>
    <!-- Optional location information where the content is released -->
  </location>
</Dissemination>
```

The MPEG-7 MediaAgent type defines several roles, among which the following (this definition taken from [MPEG-7] is not complete and merely serves as informational example):

```
<ClassificationScheme uri="urn:mpeg:mpeg7:cs:RoleCS:2001"
    domain="//CreationInformation/Creation/Creator/Role
    //UsageInformation/Dissemination/Disseminator/Role">

  <!--some content omitted here-->

  <Term termID="PUBLISHER">
    <Name xml:lang="en">Publisher</Name>
    <Definition>A person or organization that prepares and issues material
      for distribution or sale.
    </Definition>
  </Term>
  <Term termID="DISTRIBUTOR">
    <Name xml:lang="en">Distributor</Name>
    <Definition>A person or organization that markets merchandise.</Definition>
  </Term>
  <Term termID="SYNDICATOR">
    <Name xml:lang="en">Syndicator</Name>
    <Definition>A person or organization that sells material for publication in
      a number of venues simultaneously.
    </Definition>
  </Term>
  <Term termID="AGGREGATOR">
    <Name xml:lang="en">Aggregator</Name>
    <Definition>A person or organization that gathers material into a sum or
      whole.
    </Definition>
  </Term>
  <Term termID="BROADCASTER">
    <Name xml:lang="en">Broadcaster</Name>
    <Definition>A person or organization that sends out or communicates material,
      especially by radio or television.
    </Definition>
  </Term>
  <Term termID="WEBCASTER">
    <Name xml:lang="en">Webcaster</Name>
    <Definition>A person or organization that sends out or communicates material
      on the Internet by audio and/or video.
    </Definition>
  </Term>

  <!--some content omitted here-->
</ClassificationScheme>
```