

**Proposal for a common data model for
age classification information and electronic labels
MIRACLE specification v0.93 – extended documentation**

June 25th 2014

Contents

A. Background and objective	3
B. MIRACLE’s data model for machine-readable age classification data and online labels	3
Basic principles of the data model.....	3
Blocks (categories) and fields of the MIRACLE specification	5
Block 1: Issuing body - <issuer>	6
Block 2: Label metadata - <scope>	8
Block 3: Age label - <rating>	11
Block 4: Content descriptors - <content-descriptors>	12
Block 5: Feature descriptors - <feature-descriptors>.....	17
Annex 1: Exemplary data sets in full based on specification	22
Example of shortest possible version of MIRACLE XML data set.....	22
Long version with different examples	22
Annex 2: XSD for MIRACLE XML data set	25
Annex 3: Exemplary mapping of specification on existing scheme (here: age-de.xml- specification from Germany)	26

V0.93 - 25 Jun 2014

Public draft of MIRACLE specification data model

- Version for public consultation

A. Background and objective

Age ratings are highly fragmented in Europe and globally. In digital contexts, though, electronic age labels open the chance for utilizing classification knowledge across borders. MIRACLE (Machine-readable and interoperable age classification labels in Europe) is a pilot project that aims at providing a common data model for age classification information, implementing it in different national and regional schemes and showing the surplus of interoperable data in applications and end devices.

Accordingly, the first step of the project is to develop a common information exchange reference model that will be used as a data specification for all datasets provided throughout the project. The data model proposed in this paper builds on a draft proposed by the Task Force CEO Coalition's Technical Task Force. By providing a common data scheme, MIRACLE makes rating information machine-readable and technically interoperable. Such data enables software and electronic services to exchange and process data in an easy way, leading to both cost synergies for content and service providers as well as to more information available for parents and consumers. Additionally the specification proposed here will serve as a guideline for either existing players planning to implement the data model in their schemes or for new players currently considering labeling content to reduce the risk of sunk costs due to proprietary, less interoperable approaches.

B. MIRACLE's data model for machine-readable age classification data and online labels

Basic principles of the data model

The data model builds on currently existing age labeling practices, as it otherwise would undermine the efforts already taken by both companies and rating bodies as well as the classification knowledge that goes with such schemes. For companies and bodies that already provide classification data or label online content electronically, no disadvantages should result from this specification.

The three basic requirements the data model therefore takes into account are:

- (1) The data model is **technology-neutral** to reach maximum openness and compatibility between different systems and languages. Therefore, the proposal does not dictate labeling languages that have to be used, but rather the data structure, its categories, their fields and the possible values of single fields.
- (2) It **considers existing electronic labeling systems** to ensure that these are not undermined by the interoperable data model but can easily be mapped onto it.
- (3) It thoroughly **takes into account existing national and supranational classification schemes**. By doing so, existing visual labels can easily be extended by respective electronic labels, ensuring backwards-compatibility with both the data model and the underlying traditional scheme.

Proposing a data set that adheres to this principles aims at fostering a technically interoperable data structure among different rating schemes, technical implementations and distribution contexts of rated content. Information following this specification is

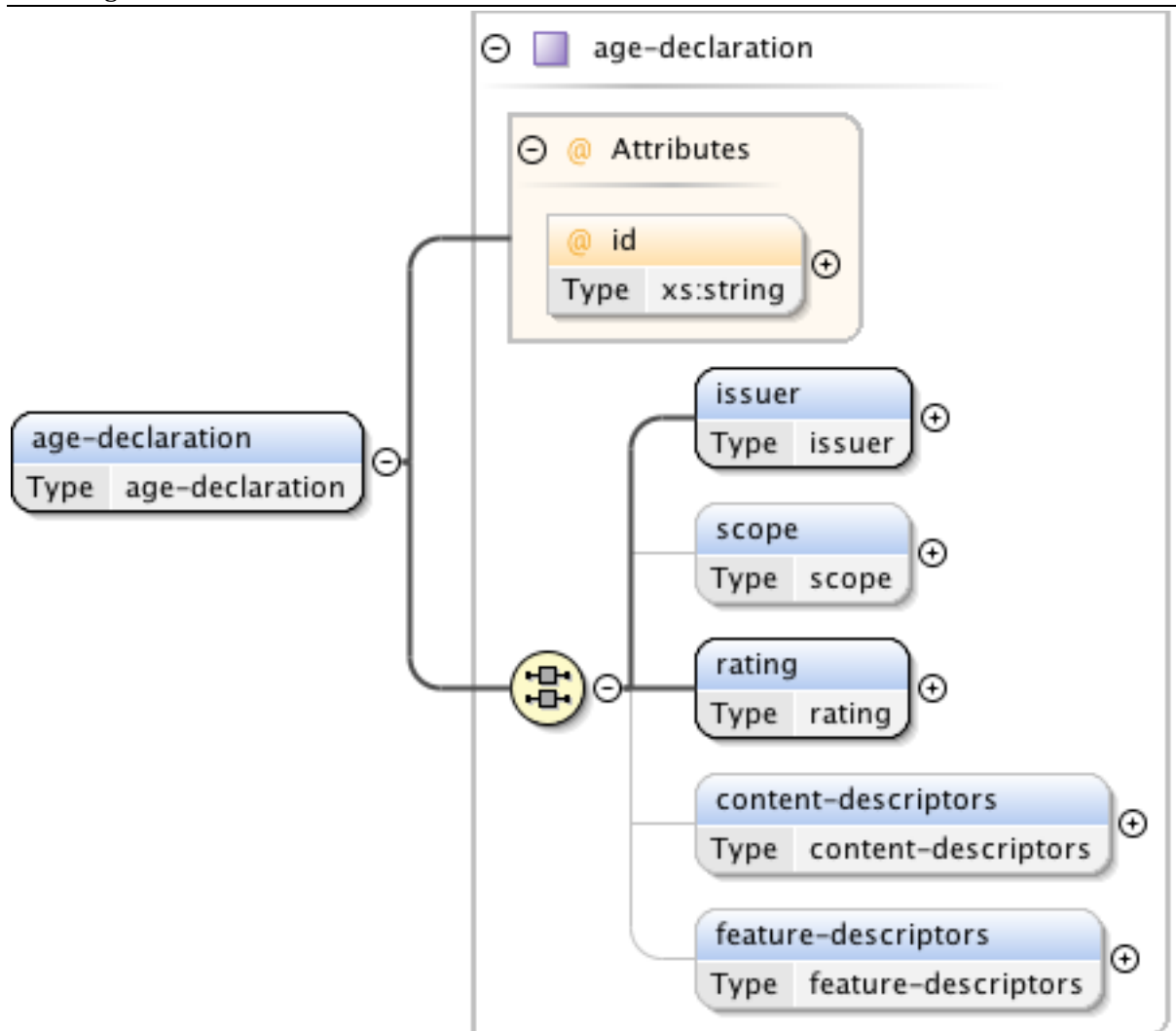
machine-readable and can be processed in different software, apps and electronic appliances. The fundamental principle of the data model is that neither existing approaches and schemes nor future ones have to provide information in *all* categories provided by the specification: As long as the data bits that are provided by a label can be mapped on any of the proposed categories, the system is technically interoperable. However, the more information a system or label provides, the better other IT systems will be able to use and process the data.

Blocks (categories) and fields of the MIRACLE specification

Main blocks of data fields within the data model are

- the body issuing the classification (<issuer>),
- the scope of the dataset (<scope>),
- age labels (<rating>),
- content descriptors (<content-descriptors>) and
- feature descriptors (<feature-descriptors>).

XSD diagram – blocks overview



Main XML structure

```
<age-declaration>
  <issuer>...</issuer>
  <scope>...</scope>
  <rating>...</rating>
  <content-descriptors>...</content-descriptors>
  <feature-descriptors>...</feature-descriptors>
</age-declaration>
```

Blocks are not in fixed order [xs:all].

Required: <issuer>, <rating>

Optional: <scope>, <content-descriptors>, <feature-descriptors>

<age-declaration> may have an attribute "id" in case of multiple datasets in one XML-file.

Block 1: Issuing body - <issuer>

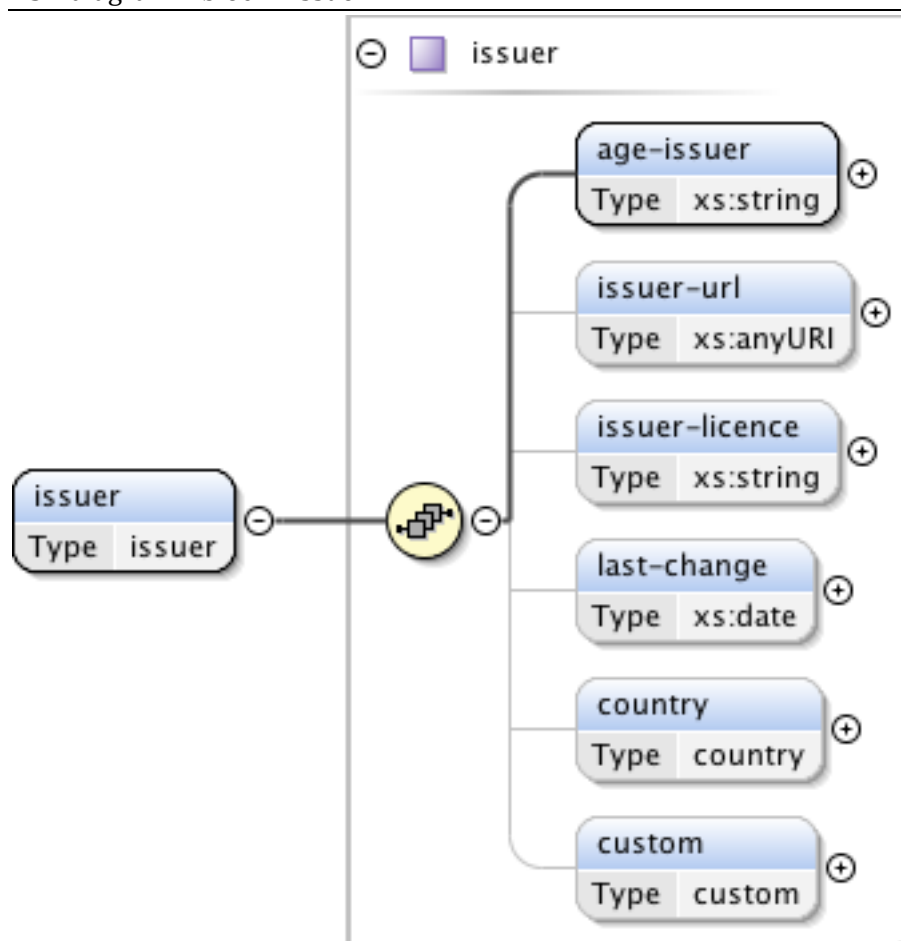
For all content ratings it is important to refer to the body that issued the specific information/label. First, this information links the age label and descriptors to brand or institutional trust. Moreover, this category allows the assessment of the regional origin of the label and its individual relevance. Also, additional information on the type of classification behind the rating procedure can be an important asset when it comes to trust. By providing the date of the last revisions of the classification, the label can also show the actuality of a rating.

Elements of <issuer> have to be in fixed order [xs:sequence].

Name	Possible values	Comments	Mandatory?
<age-issuer>	shortText [xs:string]	Since age classification can be based on self-classification, this field has to be open for all kinds of issuing bodies (FSM, PEGI, NICAM, USK, FSK, „own“, „Company Name“ etc.).	yes
<issuer-url>	URL [xs:anyURI]	Reference to address of issuing body, e.g. with additional information on institutional background and rating procedures.	no
<issuer-licence>	shortText [xs:string]	Information about type of licence for the data, or information about licensee (e.g. customer ID).	no
<last-change>	Date (yyyy-mm-dd) [xs:date]	The date of the most recent decision or update on an age classification (e.g. 2013-04-24).	no
<country> children: <country-code>	two-letter ISO country code (upper case); special values for EU and worldwide (lower case), for allowed values see XSD. [xs:string]	Country abbreviation (e.g. DE, NL, GB, etc.; EU-wide: eu; worldwide: all; multiple countries use child-tag <country-code> for each country. This tag informs about region(s) the label comes from and/or is (not	no

		necessarily legally) applied to.	
<custom> children: <custom-field class="">	shortText [xs:string]	Open text field that the issuing body can freely use for various reasons; the class-attribute is obligatory in case a <custom-field> is used; it can be used both for internal reasons as well as for expressly pointing out underlying schemes.	no

XSD diagram - block <issuer>



Snippet from a hypothetical dataset

```

<issuer>
  <age-issuer>PEGI</age-issuer>
  <issuer-url>http://www.pegi.eu</issuer-url>
  <issuer-licence>C0-2124-23443</issuer-licence>
  <last-change>2014-06-25</last-change>
  <country>
    <country-code>eu</country-code>
    <country-code>NO</country-code>
    <country-code>CH</country-code>
  </country>

```

```
<custom>
  <custom-field class="PEGI-custom">PEGI-related custom
  field content</custom-field>
</custom>
</issuer>
```

Block 2: Label metadata - <scope>

Depending on the existing form of age classification data (central classification data base that can be queried from third parties vs. a specific age label attached to specific online content), each piece of interoperable classification information has to provide information that identifies the specific content the given age information applies to („scope“). This scope might either aim at specific, isolated media content that – sometimes in a nation- or region-specific version – has already been classified or it aims at a website or service or parts of those.

The problem relating to the scope of application of a label is that its form depends on the context of implementation:

a) For *labels that are provided with the content (“age labels”)*, the scope has to clarify to what parts of the content the label applies. As the content is always distributed with the label attached, there is no need to provide any unique identifiers to link the label to the content. The issue here is to clarify the scope of the label, any exemptions or overriding special cases. Since most content and services online rely on URI structures, the metadata for these kinds of labels will be URI-based,. The basic approach will be to take the age label for the whole URI authority, usually the second level domain. However, possibilities for URI-based exemptions (specific file or folder paths) are the following.

For FQDN and URLs wildcards are defined:

(1) *.example.eu covers all subdomains including www.example.eu, sub1.sub2.example.eu plus example.eu without a subdomain. It can also be used like *.sub1.example.eu with similar rules. Wildcards for TLD are not allowed (Example: *.eu or *.co.uk is NOT allowed)

(2) Interpretation of URI-based scopes behind the first slash: www.example.eu covers all URLs of www.example.eu, e.g. www.example.eu/folder1/index.php. www.example.eu/folder1 covers all URL inside this folder like www.example.eu/folder1/pic1.jpg or www.example.eu/folder1/index.php?par=1

(3) In case only the start page is within the scope, a minus can be added. Example: www.example.eu- covers only the startpage.

Different protocol do not lead to different scopes: <https://www.example.eu> has the same scope like <http://www.example.eu> (both: www.example.eu or with wildcard *.example.eu).

b) Contrary to content-wise attached labels another form of providing classification data is to offer *centralised databases with classification data*. Usually, existing rating bodies will opt for such forms of data provisions. The issue here is that for instance an online shop or a VOD service wants to query the database for valid age classification information. To get the correct information out of the database both the data provider as well as the demander will have to use unique identifiers or fuzzy approximations via object title. As the label is detached from the content, the database query has to aim at getting the correct information back from the database. In practice, such UIDs are not

being used coherently throughout all existing rating bodies – and even if a rating body uses UIDs, they aren't the same among different rating bodies. Hence, an alternative can be to base queries on the title of the media content (e.g. movie title, game title). This fuzzy approach might, however, lead to several results, as in many cases different versions of a game or film have been classified by a rating body (cinematic version, DVD version, Director's cut, uncut version, TV version etc.).

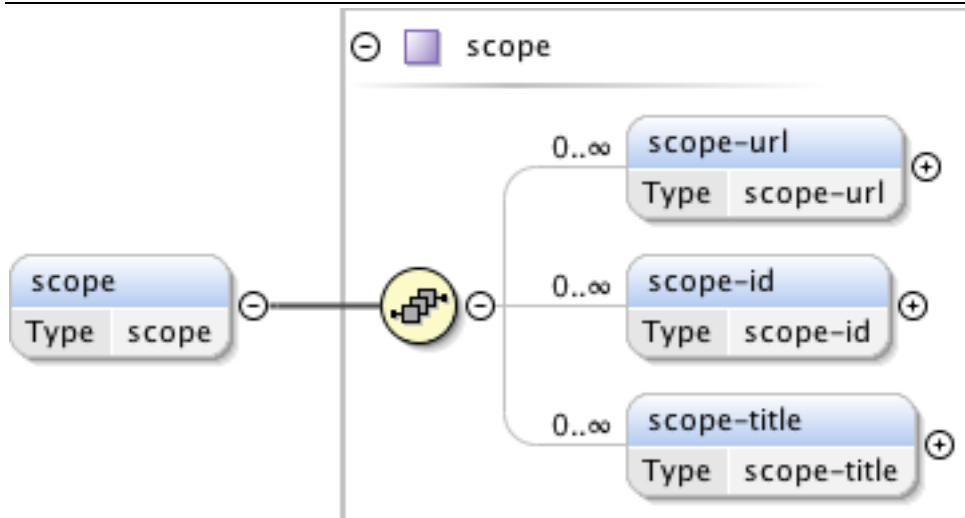
To remain flexible, the data model opts for offering all three fields: a scope URL, an UID field and a title field. These fields are not mandatory and can be used either alone or in combination. As each of the field can have one attribute, each issuing body will be able to attach system-specific UIDs to a dataset; e.g. country, rating system, producer, internal ERP or logistics numbers, EAN, GSDN, etc.).

Elements of <scope> have to be in fixed order [xs:sequence]; each element can be used more than once.

Name	Possible values	Comments	Mandatory?
<scope-url class="">	URL plus *-character *. as wildcard for subdomains including second level (*.example.eu) Minus "-" on the right side for restrictions to start page, otherwise paths are always covered, too. [anyURI]	FQDN-/URL-based scope of application of a label (not suitable for central databases of classification information that is detached from the content); the class-attribute can be used both for internal reasons as well as for expressly pointing out the underlying scheme. URL covers all protocols (e.g. http/https). No protocol-information has to be given. More than one scope-url is allowed.	No
<scope-id class="">	Alphanumeric [xs:string]	UID of classified content (system-specific, probably not suitable for labels attached to	No

		content); the class-attribute can be used both for internal reasons as well as for expressly pointing out the underlying scheme. More than one scope-id element is allowed.	
<scope-title class="">	shortText [xs:string]	Title of classified content (system-specific); the class-attribute can be used both for internal reasons as well as for expressly pointing out the underlying scheme. More than one scope-title element is allowed.	No

XSD diagram - block <scope>



Snippet from hypothetical dataset

Note: For exemplary purposes, the example includes mixed scopes that might not make sense.

```

<scope>
  <scope-url class="web-url">*.example.eu/supergame</scope-url>
  <scope-id class="PEGI-classification-no">18423</scope-id>
  <scope-title class="PEGI-title-en">Supergame Title
</scope-title>
  <scope-title class="title-de">Superspiel Titel</scope-title>
  <scope-title class="PEGI-version">1.2</scope-title>
  <scope-title class="PEGI-language">en</scope-title>
</scope>

```

Block 3: Age label - <rating>

Age labels are a common approach in content rating systems worldwide – the age rating hence is the core aspect of interoperable rating information. However, there are different schemes of how to provide age-specific information (specific age, age group, age group description, or additional age information like parental guidance). The data model will encompass these differences by making a numeric age field mandatory, but extending the data structure by additional age-related data fields.

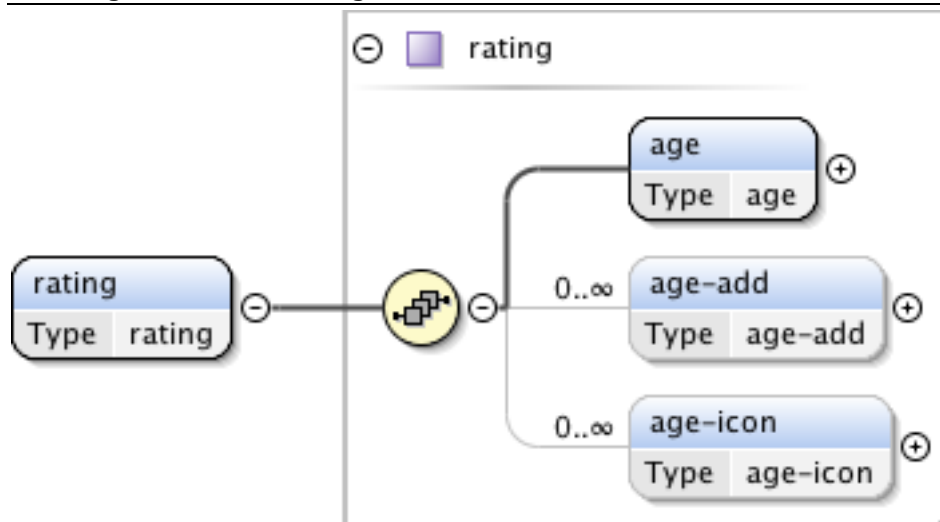
Elements of <scope> have to be in fixed order [xs:sequence].

Field	Possible values	Comments	Mandatory?
<age>	Numeric [xs:integer] min=0 max=99	Minimum age for that the content is deemed suitable. Most age classification systems already use numeric values. However, systems that do not will have to provide translation tables to provide numeric values here (e.g. US ESRB: E→0; E10+→10; T→13; M→17; A→18). Schemes that do not include any age classification (e.g. just content or feature descriptors) might use the attribute “class” with “na” and zero as a value. Each dataset must only have one <age> element.	Yes
<age-add class="">	shortText [xs:string]	If a system uses additional values to specify an age classification, these additional age information has to go here, e.g. PG, R18, 12A, XXX etc. Systems that rely on non-numeric classification (e.g. ESRB) can provide their original rating in this field, too. By providing an attribute, the data structure is able to provide a scheme-consistent explanation. More than one <age-add> element is allowed.	No
<age-icon class="">	URL [xs:anyURI]	To provide trustful classifications, this field provides the URL to the original age rating icon that can be used for displaying a visual age rating. Protocol (e.g. http://) has to be included. More than one <age-icon> element is allowed.	No

As numeric values are superior for machine-based processing than open text fields, it seems more feasible to translate textual age classification like „teens“ or „mature“ into numbers on side of the classification scheme (and its API) rather than to translate

textual values in the data model into number on processor-/client-side. An additional field might be necessary to cope with age-label-specific additional information, e.g. „parental guidance“.

XSD diagram – block <rating>



Snippet from hypothetical dataset

```
<rating>
  <age>12</age>
  <age-add class="PEGI-age">12+</age-add>
  <age-icon class="PEGI-icon">http://pegi.eu/label/12.png
</age-icon>
</rating>
```

Block 4: Content descriptors - <content-descriptors>

Many existing classification schemes use content descriptors to give additional information about the content and their reasons for a specific age rating. The categories of these descriptors are quite comparable worldwide. However, there are and always will be peculiarities of single systems or schemes, showing the need for a flexible approach of the data model. The data model opts for a mixed data field model here, where common and agreed categories are predefined, while the category allows for providing additional content descriptor fields for the sake of flexibility. It already is foreseeable that the uptake of such additional field will be comparably slow, since fragmented forms of content categories will result in the loss of synergy effects of an interoperable data model. To minimise these effects one option is to monitor the additional or new content categories and regularly decide on potential new fields that will become pre-defined later on.

In this category, too, icons are commonly used as content descriptors. The data model hence provides URL references to these icons, too.

If a pre-defined content descriptor is applicable, the allowed values are yes/no only (in this context it doesn't matter if the issuing body considers the respective relevant content as mild or strong, as long as it is relevant for the age classification decision). All content descriptor fields are optional, since many existing schemes do not provide content descriptors. Hence if a field is not provided by a data set, this doesn't default to "no".

Elements of <content-descriptors> and children must not be in fixed order [xs:all].

Pre-defined data fields (standardised content descriptors)

Name	Possible values	Comments
<cd-sexuality>		
<cd-sexuality-exist>	true/false or 1/0 [xs:boolean]	Sex/erotism/nudeness is a reason for the age classification
<cd-sexuality-desc>	shortText [xs:string]	Description of content
<cd-sexuality-icon>	URL [xs:anyURI]	Address of original sex/erotic/nudeness icon
<cd-violence>		
<cd-violence-exist>	true/false or 1/0 [xs:boolean]	Violence/weapons/blood is a reason for the age classification
<cd-violence-desc>	shortText [xs:string]	Description of content
<cd-violence-icon>	URL [xs:anyURI]	Address of original violence/weapons/blood icon
<cd-discrimination>		
<cd-discrimination-exist>	true/false or 1/0 [xs:boolean]	Discrimination/ racism/ hate speech is a reason for the age classification
<cd-discrimination-desc>	shortText [xs:string]	Description of content
<cd-discrimination-icon>	URL [xs:anyURI]	Address of original discrimination/ racism/ hate speech icon
<cd-cursing>		
<cd-cursing-exist>	true/false or 1/0 [xs:boolean]	Obscene language/bad language/cursing is a reason for the age

		classification
<cd-cursing-desc>	shortText [xs:string]	Description of content
<cd-cursing-icon>	URL [xs:anyURI]	Address of original obscene/bad language/cursing icon
<cd-drugs>		
<cd-drugs-exist>	true/false or 1/0 [xs:boolean]	Drugs/tobacco/alcohol is a reason for the age classification
<cd-drugs-desc>	shortText [xs:string]	Description of content
<cd-drugs-icon>	URL [xs:anyURI]	Address of original drugs/tobacco/alcohol icon
<cd-fear>		
<cd-fear-exist>	true/false or 1/0 [xs:boolean]	Fear/shock is a reason for the age classification
<cd-fear-desc>	shortText [xs:string]	Description of content
<cd-fear-icon>	URL [xs:anyURI]	Address of original fear/shock icon
<cd-gambling>		
<cd-gambling-exist>	true/false or 1/0 [xs:boolean]	Gambling is a reason for the age classification
<cd-gambling-desc>	shortText [xs:string]	Description of content
<cd-gambling-icon>	URL [xs:anyURI]	Address of original gambling icon

If new relevant content categories emerge or a system is making use of additional content descriptors than the ones pre-defined by the data model, it is possible to provide additional content descriptors in the following way within a block <cd-other>.

Example for additional content descriptors

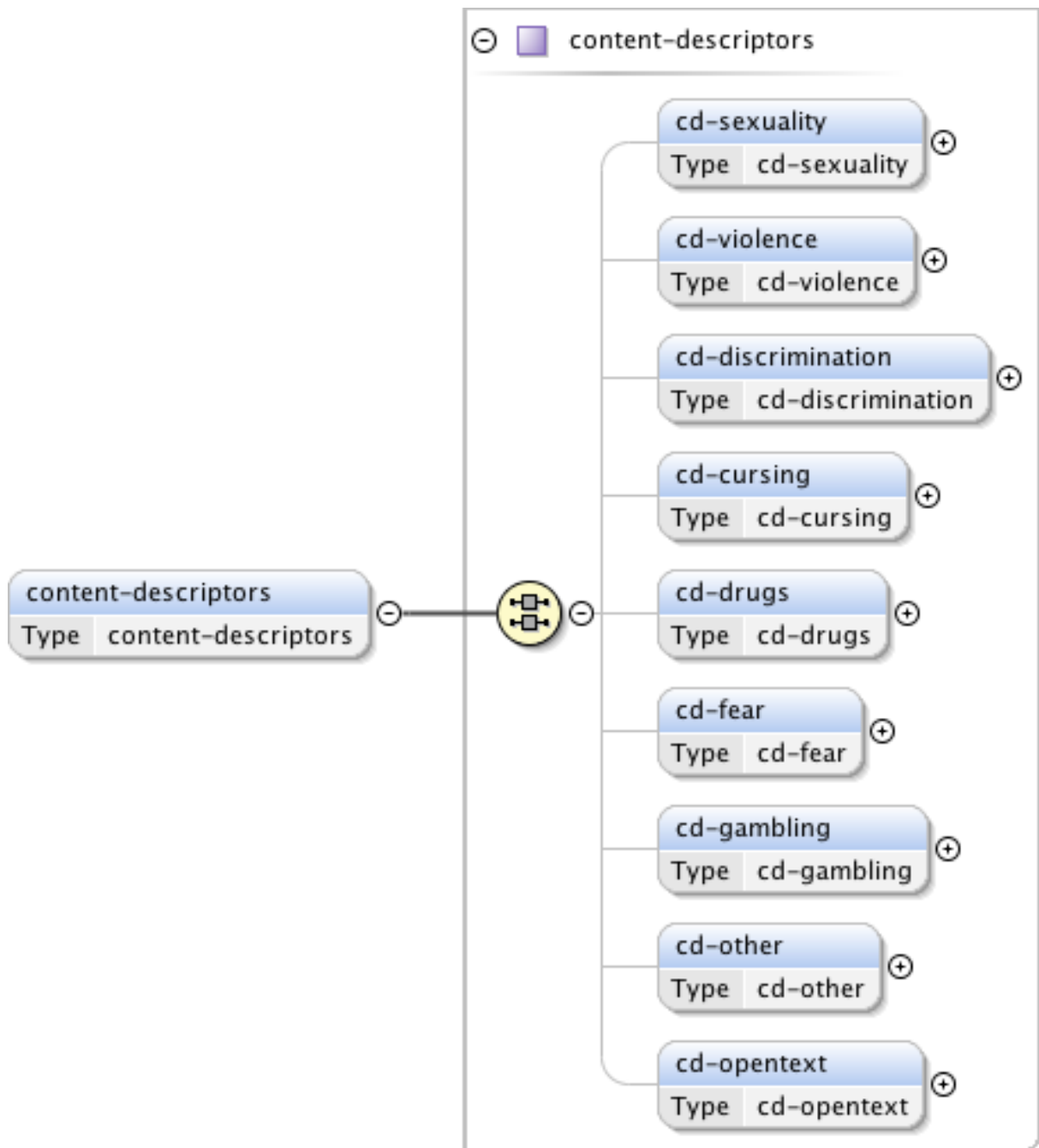
Field name	Possible values	Comments
<cd-add class="cd-xyz"> (e.g. class="cd-selfharm";		The class-attribute is mandatory here, as it

class="cd-antisocial")		contains information about the additional descriptor.
<cd-add-exist>	true/false or 1/0 [xs:boolean]	
<cd-add-desc>	shortText (short description of additional content descriptor) [xs:string]	
<cd-add-icon>	URL; address of original additional icon [xs:anyURI]	

Some systems do not provide systematic content descriptors, but offer additional information regarding the reasoning for a specific age rating in text form. Such information is harder to structure and to process technically but it still provides relevant information. Later on, systems might be able to process this data automatically, too. As some systems use such open textual descriptions (e.g. BBFC), the data model will provide such a field, too.

Field name	Possible values
<cd-opentext>	
<cd-opentext-desc class="">	shortText [xs:string]

XSD diagram – block <content-descriptors>



Snippet from hypothetical dataset

```

<content-descriptors>
  <cd-violence>
    <cd-violence-exist>true</cd-violence-exist>
    <cd-violence-icon>http://pegi.eu/label/violence.png
  </cd-violence>
  <cd-fear>
    <cd-fear-exist>true</cd-fear-exist>
    <cd-fear-icon>http://pegi.eu/label/fear.png
  </cd-fear>
  <cd-other>
    <cd-add class="self-harm">
      <cd-add-exist>true</cd-add-exist>
  </cd-other>
</content-descriptors>

```



```

        <cd-add-icon>http://pegi.eu/label/self-harm.png
        </cd-add-icon>
    </cd-add>
</cd-other>
    <cd-opentext class="PEGIONline">
        <cd-opentext-desc>Online game</cd-opentext-desc>
    </cd-opentext>
</content-descriptors>

```

Block 5: Feature descriptors - <feature-descriptors>

A different type of descriptors relate to information about features or functionalities that the content (or better: the content-related service or platform) provides to the user. Depending on the scheme, this information sometimes results in a specific age classification result, sometimes it is only regarded as additional information for end users without an effect on the actual age rating decision. Information regarding such features are relevant for minors and other consumers, too; e. g. user generated content might contain relevant depictions that would change existing age classifications, chat functionalities result in unknown people approaching (underage) user in a harmful way or location-based services log and display the movement and/or other person-related information to third parties. The PEGI scheme already started to implement such descriptors, hence a first step is to take those as predefined ones, while leaving the feature descriptor field open to new ones, too (see above additional content descriptors). All feature descriptor fields are optional, since many existing schemes do not provide content descriptors. If a field is not provided by a data set, it defaults to “no”. Elements of <feature-descriptors> and children must not be in fixed order [xs:all].

Pre-defined data fields regarding features (standardised feature descriptors)

Name	Possible values	Comments
<fd-inapppurchase>		
<fd-inapppurchase-exist>	true/false or 1/0 [xs:boolean]	The service contains elements enabling the consumer to purchase additional content or functionality, regardless of whether the app itself was acquired for free or not.
<fd-inapppurchase-desc>	shortText [xs:string]	Description of feature
<fd-inapppurchase-icon>	URL [xs:anyURI]	Address of original icon for in-app purchase features.
<fd-personaldatasharing>		

<fd-personaldatasharing-exist>	true/false or 1/0 [xs:boolean]	The service gives its provider (or a third party) access to personal data such as home address, contact details or bank account numbers.
<fd-personaldatasharing-desc>	shortText [xs:string]	Description of feature
<fd-personaldatasharing-icon>	URL [xs:anyURI]	Address of original icon for personal data sharing features.
<fd-locationdatasharing>		
<fd-locationdatasharing-exist>	true/false or 1/0 [xs:boolean]	The service contains the option to share exact location on a map when using the service. The location information may be shared publicly or with a specific network inside the service or elsewhere online.
<fd-locationdatasharing-desc>	shortText [xs:string]	Description of feature
<fd-locationdatasharing-icon>	URL [xs:anyURI]	Address of original icon for location data sharing features.
<fd-chat>		
<fd-chat-exist>	true/false or 1/0 [xs:boolean]	The service includes an option for a user to chat with other users of the app. These users may operate under a pseudonym or anonymously.
<fd-chat-desc>	shortText [xs:string]	Description of content
<fd-chat-icon>	URL [xs:anyURI]	Address of original icon for chat features.

Similarly to the content descriptors, additional feature descriptors will emerge during time. Hence, the data has to be open to additional or new descriptors, too.

Example for additional feature fields (additional feature descriptors)

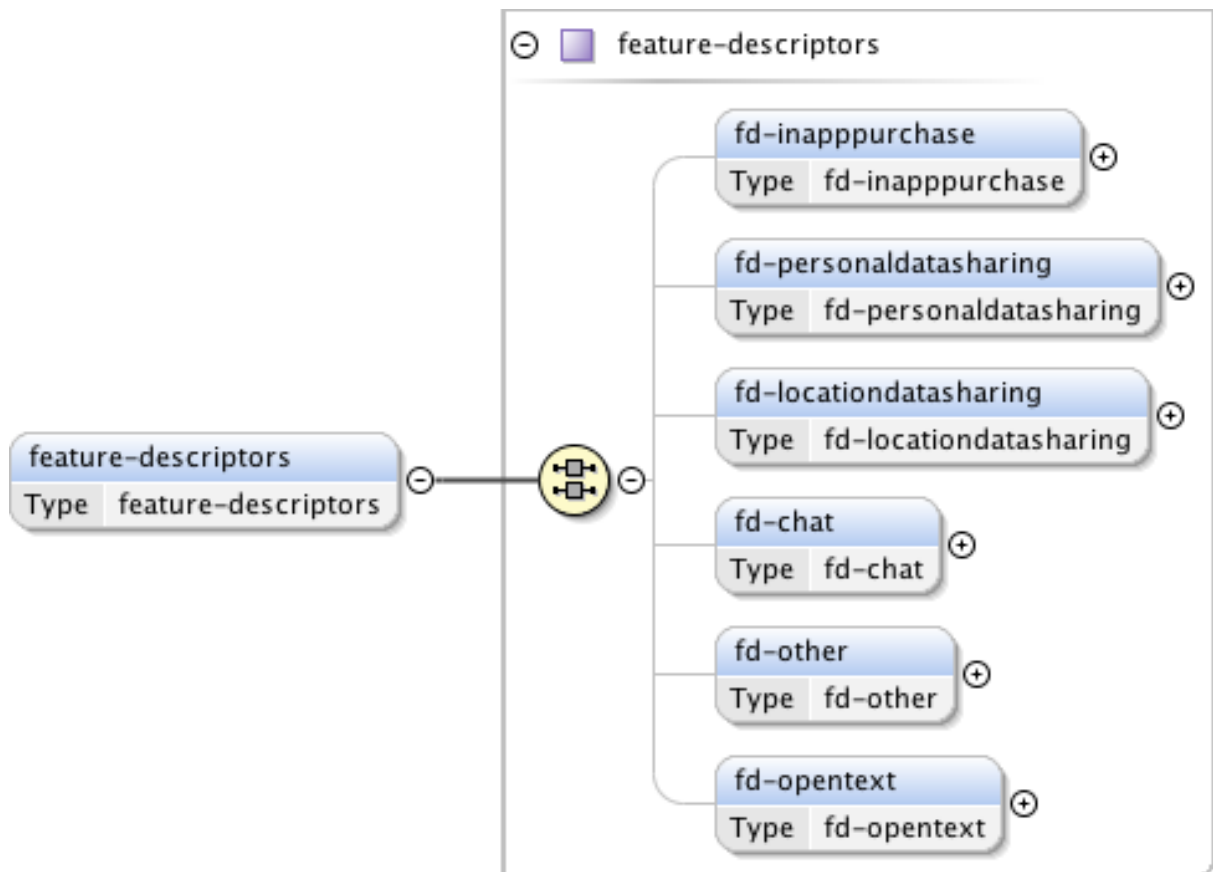
Field name	Possible values	Comments
<fd-add class="fd-xyz"> (e.g. class="fd-upload")		The class-attribute is mandatory here, as it contains information about the additional descriptor
<fd-add-exist>	true/false or 1/0 [xs:boolean]	
<fd-add-desc>	shortText (short description of additional feature descriptor) [xs:string]	
<fd-add-icon>	Address of original additional icon [xs:anyURI]	

Such additional fields can become pre-defined fields in later versions of the data model, when deemed necessary.

Some systems do not provide systematic feature descriptors, but offer additional information regarding the reasoning for a specific age rating or other information regarding existing features in text form. Such information is harder to structure and to process technically but it still provides relevant information (see above).

Field name	Possible values
<fd-opentext>	
<fd-opentext-desc class="fd-xyz">	

XSD diagram – block <feature-descriptors>



Snippet from hypothetical dataset

```

<feature-descriptors>
  <fd-inapppurchase>
    <fd-inapppurchase-exist>true
    </fd-inapppurchase-exist>
    <fd-inapppurchase-icon>http://pegi.eu/label/iap.png
    </fd-inapppurchase-icon>
  </fd-inapppurchase>
  <fd-personaldatasharing>
    <fd-personaldatasharing-exist>true
    </fd-personaldatasharing-exist>
    <fd-personaldatasharing-icon>
    http://pegi.eu/label/pds.png
    </fd-personaldatasharing-icon>
  </fd-personaldatasharing>
  <fd-chat>
    <fd-chat-exist>1</fd-chat-exist>
    <fd-chat-icon>http://pegi.eu/label/chat.png
    </fd-chat-icon>
  </fd-chat>
  <fd-other>
    <fd-add class="self-harm">
      <fd-add-exist>true</fd-add-exist>
      <fd-add-icon>http://pegi.eu/label/self-harm.png
      </fd-add-icon>
    </fd-add>
  </fd-other>
  <fd-opentext>

```

```
<fd-opentext-desc class="PEGI-fd-info">  
  Text about additional features  
</fd-opentext-desc>  
</fd-opentext>  
</feature-descriptors>
```

Annex 1: Exemplatory data sets in full based on specification

Disclaimer: This is a completely fabricated, hypothetical data set for demonstration purposes only. All content is exemplary and names of content titles or organisations are only used for practical clarification.

Example of shortest possible version of MIRACLE XML data set

```
<?xml version="1.0" encoding="UTF-8"?>
<age-declaration xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="miracle-093.xsd">
  <issuer>
    <age-issuer>myhomepage.cz</age-issuer>
  </issuer>
  <scope>
    <scope-url>*.myhomepage.cz</scope-url>
  </scope>
  <rating>
    <age>6</age>
  </rating>
</age-declaration>
```

Long version with different examples

(Note: for example reasons some information might not make sense in real life)

```
<?xml version="1.0" encoding="UTF-8"?>
<age-declaration xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="miracle-093.xsd">
  <issuer>
    <age-issuer>PEGI</age-issuer>
    <issuer-url>http://www.pegi.eu</issuer-url>
    <issuer-licence>C0-2124-23443</issuer-licence>
    <last-change>2014-06-25</last-change>
    <country>
      <country-code>eu</country-code>
      <country-code>NO</country-code>
      <country-code>CH</country-code>
    </country>
    <custom>
      <custom-field class="PEGI-custom">
        PEGI-related custom field content
      </custom-field>
    </custom>
  </issuer>
  <scope>
    <scope-url class="web-url">*.example.eu/supergame
  </scope-url>
    <scope-id class="PEGI-classification-no">18423</scope-id>
    <scope-title class="PEGI-title-en">Supergame Title
  </scope-title>
    <scope-title class="title-de">Superspiel Titel
  </scope-title>
    <scope-title class="PEGI-version">1.2</scope-title>
    <scope-title class="PEGI-language">en</scope-title>
```

```

</scope>
<rating>
  <age>12</age>
  <age-add class="PEGI-age">pegi12+</age-add>
  <age-icon class="PEGI-icon">
    http://pegi.eu/label/12.png</age-icon>
</rating>
<content-descriptors>
  <cd-violence>
    <cd-violence-exist>>true</cd-violence-exist>
    <cd-violence-icon>http://pegi.eu/label/violence.png
    </cd-violence-icon>
  </cd-violence>
  <cd-fear>
    <cd-fear-exist>>true</cd-fear-exist>
    <cd-fear-icon>http://pegi.eu/label/fear.png
    </cd-fear-icon>
  </cd-fear>
  <cd-other>
    <cd-add class="self-harm">
      <cd-add-exist>>true</cd-add-exist>
      <cd-add-icon>http://pegi.eu/label/self-harm.png
      </cd-add-icon>
    </cd-add>
  </cd-other>
  <cd-opentext>
    <cd-opentext-desc class="PEGIONline">
      Online game</cd-opentext-desc>
    </cd-opentext>
</content-descriptors>
<feature-descriptors>
  <fd-inapppurchase>
    <fd-inapppurchase-exist>>true</fd-inapppurchase-exist>
    <fd-inapppurchase-icon>http://pegi.eu/label/iap.png
    </fd-inapppurchase-icon>
  </fd-inapppurchase>
  <fd-personaldatasharing>
    <fd-personaldatasharing-exist>>true
    </fd-personaldatasharing-exist>
    <fd-personaldatasharing-icon>
      http://pegi.eu/label/pds.png
    </fd-personaldatasharing-icon>
  </fd-personaldatasharing>
  <fd-chat>
    <fd-chat-exist>>true</fd-chat-exist>
    <fd-chat-icon>http://pegi.eu/label/chat.png
    </fd-chat-icon>
  </fd-chat>
  <fd-other>
    <fd-add class="self-harm">
      <fd-add-exist>>true</fd-add-exist>
      <fd-add-icon>http://pegi.eu/label/self-harm.png
      </fd-add-icon>
    </fd-add>
  </fd-other>
  <fd-opentext>
    <fd-opentext-desc class="PEGI-fd-info">
      Text regarding additional features</fd-opentext-desc>

```

```
    </fd-opentext>  
  </feature-descriptors>  
</age-declaration>
```


Annex 2: XSD for MIRACLE XML data set

There is a XML Schema Definition for the MIRACLE specification. It documents all elements and types in detail and can be used to validate conformity of MIRACLE-compatible XML instances.

The XSD file and its technical documentation is available at <http://www.miracle-label.eu/consultation>

Annex 3: Exemplatory mapping of specification on existing scheme (here: age-de.xml-specification from Germany)

This example shows how information from neutral MIRACLE data-sets can be mapped downwards compatible to the age-de.xml-specification from Germany.

```
<age-declaration>

<ageblock-basic>
<age-issuer>ratingbody</age-issuer>
<issuer-url>http://www.ratingbody.eu</issuer-url>
<issuer-licence>123456789</issuer-licence>
<last-change>2014-06-22</last-change>
<label-version>2.0</label-version>
<revisit-after>always</revisit-after>
</ageblock-basic>

<ageblock-country>
<country class="age-de.xml">
<country-code>DE</country-code>
<country-code>CH</country-code>
<country-code>AT</country-code>
<country-code>NL</country-code>
</country>
<country class="age-eu.xml">
<country-code>eu</country-code>
</country>
<country-default>age.xml</country-default>
</ageblock-country>

<ageblock-labeltype>
<xmlfile>>true</xmlfile>
<httpheader>>false</httpheader>
<htmlmeta>>false</htmlmeta>
<savefilter>>false</savefilter>
<default-age>0</default-age>
<alternate age="16">http://www.youthpage-example.eu</alternate>
<alternate>http://www.kidspage-example.eu</alternate>
</ageblock-labeltype>

<ageblock-labeltype-definition>

<labeltype-xmlfile>
<label class="example-scope1">
<age>18</age>
<age-add class="pegi">pg18</age-add>
<age-add class="bbfc">bbfc18</age-add>
<age-icon class="pegi">http://www.pegi.info/icon/pg18.png</age-icon>
<age-icon class="bbfc">http://www.bbfc.co.uk/icon/violence-icon.gif</age-icon>
<scope>*.website.eu/games/violence/game1.html</scope>
<scope-id class="pegi">4567</scope-id>
<scope-id class="bbfc">7654</scope-id>
<scope-id class="ean">4260192682132</scope-id>
</label>

<label class="example-scope2">
<age>12</age>
<scope>*.website.eu</scope>
</label>
</labeltype-xmlfile>

</ageblock-labeltype-definition>

<ageblock-descriptors class="PEGI">
<content-descriptors>
```

```

    <cd-violence>
      <cd-violence-exist>true</cd-violence-exist>
      <cd-violence-desc>Description of violence in content</cd-
violence-desc>
      <cd-violence-icon>http://pegi.eu/label/violence.png</cd-violence-icon>
    </cd-violence>
    <cd-fear>
      <cd-fear-exist>true</cd-fear-exist>
      <cd-fear-icon>http://pegi.eu/label/fear.png</cd-fear-icon>
    </cd-fear>
      <cd-opentext class="PEGI-cd-info">some text</cd-opentext>
    </content-descriptors>
<feature-descriptors>
  <fd-inapppurchase>
    <fd-inapppurchase-exist>true</fd-inapppurchase-exist>
    <fd-inapppurchase-desc>Description of inapppurchase in
content</fd-inapppurchase-desc>
    <fd-inapppurchase-icon>http://pegi.eu/label/iap.png</fd-inapppurchase-
icon>
  </fd-inapppurchase>
  <fd-personaldatasharing>
    <fd-personaldatasharing-exist>true</fd-personaldatasharing-exist>
    <fd-personaldatasharing-icon>http://pegi.eu/label/pds.png</fd-
personaldatasharing-icon>
  </fd-personaldatasharing>
  <fd-chat>
    <fd-chat-exist>true</fd-chat-exist>
    <fd-chat-icon>http://pegi.eu/label/chat.png</fd-chat-icon>
  </fd-chat>
    <fd-opentext class="PEGI-fd-info">some text</fd-opentext>
  </feature-descriptors>
</ageblock-descriptors>

<ageblock-descriptors class="BBFC">
  <content-descriptors>
    <cd-violence>
      <cd-violence-exist>true</cd-violence-exist>
      <cd-violence-desc>Description of violence in content</cd-
violence-desc>
      <cd-violence-icon>http://bbcf.co.uk/label/violence.png</cd-violence-
icon>
    </cd-violence>
    <cd-fear>
      <cd-fear-exist>true</cd-fear-exist>
      <cd-fear-icon>http://bbcf.co.uk/label/fear.png</cd-fear-icon>
    </cd-fear>
      <cd-opentext class="BBFC-cd-info">some text</cd-opentext>
    </content-descriptors>
  <feature-descriptors>
    <fd-inapppurchase>
      <fd-inapppurchase-exist>true</fd-inapppurchase-exist>
      <fd-inapppurchase-desc>Description of inapppurchase in
content</fd-inapppurchase-desc>
      <fd-inapppurchase-icon>http://bbcf.co.uk/label/iap-icon.gif</fd-
inapppurchase-icon>
    </fd-inapppurchase>
    <fd-personaldatasharing>
      <fd-personaldatasharing-exist>true</fd-personaldatasharing-exist>
      <fd-personaldatasharing-icon>http://bbcf.co.uk/label/pds-icon.gif</fd-
personaldatasharing-icon>
    </fd-personaldatasharing>
    <fd-chat>
      <fd-chat-exist>true</fd-chat-exist>
      <fd-chat-icon>http://bbcf.co.uk/label/chat-icon.gif</fd-chat-icon>
    </fd-chat>
      <fd-opentext class="BBFC-fd-info">some text</fd-opentext>
    </feature-descriptors>
</ageblock-descriptors>

</age-declaration>

```

