



Machine-readable and interoperable
age classification labels in Europe

Grant agreement no: 621059

Implementation Report
British Board of Film Classification (BBFC)

February 12th, 2015

Editors: Chris Hunter-Brown (chunterbrown@bbfc.co.uk)

The icon for the Creative Commons Attribution-NonCommercial-NoDerivs (CC BY-NC-ND) license, showing a person, a crossed-out dollar sign, and a crossed-out document.

Contents

Summary of Implementation strategy	3
Existing data structure	4
Ratings and decisions.....	4
Anticipated amount of work.....	6
Implementation phase.....	7
Technical implementation steps.....	7
Dealing with MIRACLE limitations.....	8
Endpoints which cannot be offered under MIRACLE 1.0	8
JSON interpretations of MIRACLE are currently ambiguous	9
MIRACLE services.....	10
Overview	10
Developer portal	10
Accessing the endpoints.....	10
Response content types.....	10
Errors	10
Endpoints.....	11
Match	11
Details.....	13
Sample responses	15
A single MIRACLE age label in XML	15
An array of multiple age-labels in JSON	16
Outlook: Challenges and business case.....	17
Challenges.....	17
Licensing.....	17
Multiple API's, a lack of standardisation.....	17
Unique identifiers	17
Potential as of MIRACLE 1.0	18

Summary of Implementation strategy

To implement MIRACLE in the BBFC context, we have chosen to extend our existing web API to support the retrieval of ratings data in the MIRACLE data format. In short, a centralised approach.

The alternative would be to provide a MIRACLE age label to content owners at the point of completion of the rating process, perhaps as an .xml file for them to download from our customer extranet for each piece of content they have submitted for an age rating. Whilst this distributed approach would be very straight-forward for us to do and a good fit for MIRACLE as of version 1.0, our experience to date suggested this would have led to very little practical distribution of MIRACLE data during the project lifetime.

At the BBFC, we have been working with various parties in the film industry on the use of our age ratings online since 2008 and have had to adapt our approach significantly during this time. In our experience, many content owners only retain a minimal amount of the age rating information that we provide and this rarely ever makes it through the supply chain to the consumer intact. On many digital video platforms, ratings are either missing or inaccurate and only very rarely accompanied with any content descriptors.

This is in part due to the lack of a common standard for embedding an age rating label inside a media file, like the successful [ID3](#) containers for digital audio files. Naturally, this is something that MIRACLE is looking to address but presents something of a “chicken and egg” scenario.

In our first attempt to address this, we provided a unique digital identifier to the age rating to the content owner for every piece of content we issued a rating to. Despite our best efforts, this too would rarely make it through the supply chain. In order to supply accurate ratings to digital video platforms that were offering tens of thousands of titles, we needed a new approach.

We created our web API to attempt to solve this problem, how to reunite an age label with a piece of content. As well as offering a means for services to retrieve age rating data in a machine readable format, we refactored our data to allow us to use metadata to help disambiguate the correct rating for a piece of content. This is a service we now provide at scale. Extending our centralised approach to support MIRACLE allows us to lean on this work together with the infrastructure already in place.

At present, most data we supply to a third party is provided under a commercial digital licence and it is necessary that we extend this to data that we make available via MIRACLE also. We expected to be able to adopt our existing licencing and fees without extensive modification but are also committed to offering our modest catalogue of video game ratings under much less restrictive terms, certainly free of charge, in order to support the hackathons and experimentation during the lifetime of the MIRACLE project.

Finally, we also have a modest amount of age ratings generated via our pilot [YouRateIt](#) project. As this scheme operates on a very different basis to our statutory business we will look to publish these ratings under MIRACLE also, once agreement from the project partners is secured. We expect this to be sometime before MIRACLE 2.0 is finalised and anticipate this too will be under less restrictive terms.

Existing data structure

The BBFC has been providing age ratings since 1912 and we have an extensive catalogue of ratings. This number bolstered and complicated further by the way we are required to work.

It is extremely common for us to issue more than one rating for a particular “film” or “video”. This can be due to slightly different versions of the content, director’s cuts or otherwise edited versions to achieve a particular age rating. The format of the work can also mandate a separate rating. Ratings issued for physical media have a different legal basis than those issued for exhibition in a cinema or for online distribution so these are necessarily separate. Having been in operation for over 100 years, our rating standards have also evolved over time. In some cases, a film given an adult rating in the early half of the last century may be rated for a much wider audience today. Conversely, films rated under a previous set of BBFC guidelines may accrue a more restrictive age rating if submitted again when different guidelines are in place.

Some of these issues are likely to apply to other age rating organisations, others are specific to the BBFC domain but they are of acute importance to us in terms of providing accurate and useful age rating information to consumers. It is imperative MIRACLE remains flexible enough to cover these requirements.

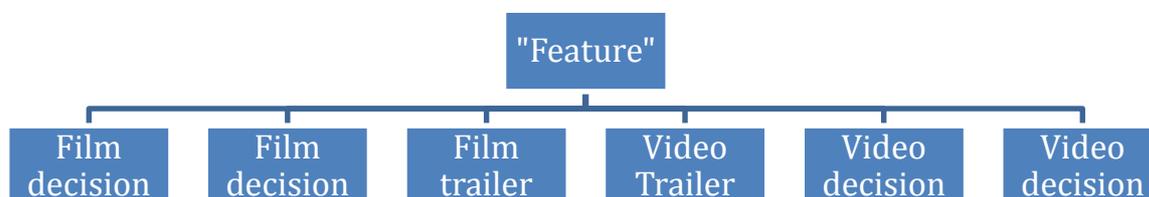
A particular title may have several different ratings some of which are only valid in a particular context. Combined with the number of ratings we have issued, now just shy of 340,000, this can make isolating the correct rating for a given piece of content a challenge for a third party. Particularly in the absence of any usable unique identifiers. In 2012, we enhanced our data structure to try to address this.

Ratings and decisions

Traditionally, we keep an individual record for every age rating decision that we make, regardless if one of those decisions refers to a slightly different version or even identical content as a previous decision. Here is a breakdown of age rating decisions we have made to date.

Type	Since	Notes	No. of decisions
Film (theatrical)	1912	Works exhibited in cinemas, rated under the Licensing Act. These ratings cannot be used for online content.	75304
- Film Trailers	1912		13961
- Film Adverts	1912		12320
AsLive	2011	7 day pre-certificate for theatrical exhibition of live performances (sport, opera etc.)	108
Video (DVD/Blu-ray)	1984	Packaged goods (DVD, Blu-ray etc.), works released on physical media rated under the Video Recordings Act. These ratings can also be used for online content	204589
- Video Trailers	1984		27103
- Video Adverts	1984		2102
VoD (online only)	2008	These ratings cannot be used for content released on physical media	2205
Video games	1986 - 2012	Only a subset of games released in this period required a BBFC rating	1918

Our decisions have traditionally been organised by the “type of decision” we were making. Where in the past we have only been concerned about “films” this was sufficient. As we started to move into “videos” which no longer just comprise “films” but TV series, documentaries and so forth this structure became less useful. This was further aggravated as content made its way online. As a result we created an abstraction layer on top of our database of decisions



In this structure, multiple decisions are represented by a “rating” object. Additionally this parent object aggregates information contained within the decisions and itself has an age rating, the “current rating” which is determined by evaluating the descendent decisions. This is to be considered only as a general answer to the question “What rating is this thing”, for a particular use case such as exhibiting the work in a cinema or offering it for streaming online, the answers may be different and it is the age rating from a particular *decision* that should be used. This is best shown by an example.



- The “current” rating for the movie *Jaws* is ‘12A’ as we rated it most recently in 2012. The 1975 rating is effectively obsolete.
- For theatrical exhibition, the rating is still ‘12A’.
- For packaged goods or online distribution, the rating is a less restrictive **PG**. This is because the theatrical rating cannot legally be used for packaged goods and is not authorised to be used online.
- Were this title to be submitted for a new age rating for online and/or packaged goods, it would likely be given a ‘12’ under our more modern guidelines
- The three PG decisions relate to different edits, it would depend on the edit in question as to which one actually applies.

It is unrealistic to expect a third party to deal with this complexity. The role of the rating object is to isolate the correct title and provide a gateway to the relevant decisions. Our API can provide the correct age rating when informed of the context required.

By creating a two level hierarchy, the number of objects is greatly reduced. This makes it much more feasible to isolate the correct title using metadata such as one or more known titles, directors, distributors, publishers running times, year of release etc. Work on building out this hierarchy is an ongoing task. We focused on “movies” first as that reflects the greatest demand but are working on building a more sophisticated graph for TV that categories episodes into seasons and series’.

Type	Notes	No. of “rating” objects
“Features”	Includes films that have been exhibited in a cinema, released online or on DVD/Blu-ray. A “feature” rating will comprise all the known edits of a movie that have been issued an age rating.	94437
“TV Episodes”	In the near future, this will be better organised into series’/seasons/episodes	62386
“Video games”	A “video game” rating will be guaranteed to be cross-platform and potentially comprise several versions of the same game with minor differences in content	1563

For MIRACLE, both types of object can be returned by the API. Whilst the specifics of how to obtain the “correct” age rating for a title from our services are not strictly a part of the project or standard, they are none-the-less important in terms of creating an effective implementation that meets our standards.

Anticipated amount of work

At the proposal stage, we anticipated it would take a total of four person months to complete the implementation.

The bulk of the implementation is made up of development work on our existing API to translate our existing age rating records into MIRACLE labels, and the provision of new endpoints via which labels can be retrieved. As we have an internal IT resource, we had no need to sub-contract this work to a third party and envisaged the work would be carried out by the developer of the existing API. However, we built some leeway into this resource budget to account for the possibility the work may have to be undertaken by another developer less familiar with these systems.

Additionally, we budgeted to add an additional API server to ensure that provision of MIRACLE services is robust and able to scale up to any increased demand generated by the project.

Finally, our implementation budget incorporates some time for our business development and legal teams to evaluate our current licensing arrangements for the provision of data services and produce a suitable licence for consumers of data via MIRACLE. We did not anticipate our current licences would require a great deal of modification but our goal was to provide a subset of our data with as few restrictions as possible to support the project including the proposed hackathons.

Implementation phase

The bulk of the implementation work was planned to be carried out between November 2014 and January 2015 with a view to having a demonstrable implementation ready for the project meeting in mid-January. This period was chosen on the basis that it is a quieter time of the year in terms of internal projects but other work was carefully planned to avoid interference as much as possible. Although the work was to be undertaken by internal resources, as a small organisation these are by no means extensive so leave and other responsibilities of key members of staff impacted on our schedule a little, and some tasks over-ran into February. However, we were able to ensure these were weighted to supporting tasks such as the implementation report, documentation, internal arrangements for managing licensees etc. to ensure we could still deliver a working alpha-level demonstration by mid-January.

As we had infrastructure already in place, we were able to replicate some of the steps taken when constructing our existing API. This not only saved us some of the physical implementation work that might be required to build a MIRACLE API from scratch, it also meant we did not have to produce the architecture design. This afforded us the luxury of being able to implement MIRACLE in the BBFC context with a relatively modest amount of resources. Only four person months were budgeted, largely allocated to a sole developer tasked with extending the existing API.

At the time of completing our periodic review, we had used up 2.26 person months of the 4 person months allocated for the implementation phase WP2, with much of the remaining budget dedicated to documentation and time for our business development and legal teams to review our licensing requirements. With this infrastructure already in place, our only direct financial costs were staff costs.

Technical implementation steps

Our primary data stores are [IBM Domino](#) databases. Domino underpins our internal systems and we find the schema-less, document orientated storage model to be the best fit for the information we need to record to produce an age rating decision. From our internal systems, a database containing the subset of decisions suitable for publication is updated and the parent rating objects in the abstraction layer are generated from these in a separate Domino database. Using the IBM Domino replication technology, these are then replicated to several replicas on our web facing API server although no requests are served by these databases directly.

Using the façade pattern, a separate Domino application serves as the front-end application server to the databases. A Domino agent is written to facilitate each API endpoint with a common library of code to manage APIkey authentication, conversion of internal fields to our output format and to access the correct records in the correct databases. This is then served behind an [NginX](#) proxy that allows for us to publish cleaner URL's, log requests and provide rate limiting and load balancing should we require it in the future.

In order to implement MIRACLE, we developed an additional façade application that contained the MIRACLE specific endpoints. This is so we could offer MIRACLE services alongside our existing API but still have them function slightly differently due to constraints with the MIRACLE format and provide the two tiers of free and paid access via separate APIkeys. As we did not previously offer XML output a separate library was developed containing functions to build a MIRACLE label from our internal database

records. We avoided technologies such as XSLT and XML functions provided by Domino as they vastly exceeded our need to just be able to create the tags and attributes that make up the relatively simple format.

This separation of concerns allows us to version our API's appropriately. For MIRACLE 2.0 we will develop a second façade application and be able to maintain access to 1.0 and 2.0 indefinitely if required. All of the façade applications work from the same set of data.

We use the API documentation tool [swagger](#) to document the API endpoints. A swagger-ui instance is hosted on our developer portal to provide a front-end to the documentation and facilitate the running of sample queries against live data.

Dealing with MIRACLE limitations

Endpoints which cannot be offered under MIRACLE 1.0

Our existing API offers the following endpoints;-

Match

The Match API will attempt to isolate the exact rating for a given title, director and year. When accurate values for all three of these fields are provided, the match will be highly reliable and will often return an unambiguous single result. However, applications should always be tolerant to the possibility of multiple matches and consider the confidence of any given match.

The BBFC database dates back to 1912 and covers the vast majority of film and video works released in the UK during this time. A title alone will nearly always be too ambiguous. The string "Top Gun" for example will return four different results.

Search

The search API operates on a wider scope than the match API and will instead, return the widest array of results possible for a given query akin to a free-text search of the BBFC database. Rating objects are prioritised over decision objects in that the latter are only returned if no ratings can be found.

Suggest

Similar to the search API, the suggest API will return up to five suggestions for the given query and is largely to facilitate using BBFC data in an auto-complete bar. Only rating objects and a subset of the normal fields will be returned.

Ratings

Will return a feed of new or updated *ratings*

Decisions

Will return a feed of new or updated *decisions*

Details

The details endpoint will return the full details for any rating or decision in exchange for a BBFC identifier or known to the BBFC third party identifier.

For version 1.0 of our MIRACLE implementation, only the **Match** and **Details** endpoints will be implemented. This is due to a limitation of the MIRACLE XML format whereby the root element is the containing element of a specific age-declaration, rather than a collection of age-declarations. This makes it impossible at present to return an XML response containing more than one age rating label. The **Search**, **Suggest**, **Ratings** and **Decisions** endpoints are all designed to return multiple results as a matter of course. Conversely, the **Details** endpoint will only ever return a single result and the **Match** endpoint can be configured to return only the most likely result. For XML responses, this is the exact approach that we take. For responses in JSON, we will return multiple age labels where appropriate by way of proposing a method of addressing this limitation in future iterations of MIRACLE. This is made possible by the more open-ended nature of JSON where a schema is not required. However, for the time being this should be considered non-standard behaviour and subject to change.

A proposal to add a new container root element to the XML standard is already in progress and will specifically allow multiple age labels to be returned. Therefore, this issue is likely to already be addressed come MIRACLE 2.0, if not before.

JSON interpretations of MIRACLE are currently ambiguous

Whilst the open-ended nature of JSON affords us flexibility, it also means there are multiple ways one might describe a MIRACLE label using this format. Specifically, there is no direct equivalent in JSON for XML attributes and these might be implemented directly as objects, with multiple attributes stored as objects in an array or as properties of the containing object, with multiple values stored in an array specific to that attribute.

Leaving this open to interpretation will be problematic to consumers of the API so it would be incumbent on the MIRACLE project to define exactly the particular format of JSON responses. For MIRACLE 1.0 we have adopted our own interpretation but this should be considered non-standard and merely a proposal at this stage. When it comes to MIRACLE 2.0, a slightly enhanced format just as JSON-LD might be considered more suitable as it affords the opportunity to incorporate better semantics and structure that is offered by the XSD that underpins the XML format.

MIRACLE services

Overview

Access to our MIRACLE API is authenticated with an API key and subject to acceptance of a BBFC digital licence. Film and video data is likely to be subject to a fee, our catalogue of video game age ratings issued between 1986 and 2012 is provided free of charge under a more open licence with a view to encourage experimentation with MIRACLE data sets. The exact fee that may be charged is dependent on the nature of the third party's organisation, their intended uses of the data and the anticipated volume of requests. In the first instance, interested parties should contact digital@bbfc.co.uk for an API key where our business development team will be able to recommend the next steps. For research and development purposes, an API key allowing access to our free video game age ratings should be issued almost instantly, subject to the key owner adhering to the terms and conditions of our general licence. Third parties who wish to engage in a commercial relationship for access to our film and video data should allow sufficient time for both parties to agree terms before receiving their key.

The MIRACLE API apes many conventions of existing web API's that adopt the principles as REST. Where data is only provided on a read-only basis, many of these conventions are unnecessary but any developer comfortable with consuming REST API's in either XML or JSON will find accessing the endpoints below straight-forward. However, care should be taken to understanding the structure of BBFC age ratings data in order to form successful requests.

Developer portal

Extensive documentation with an interactive explorer to submit test queries is available at our developer portal, <http://developer.bbfc.co.uk>. Presently, an account is required for access and credentials will normally be delivered alongside the API key. A generic login is provided for internal use. The username and password are both **miracle**

Accessing the endpoints

The base URL is <http://api.bbfc.co.uk/miracle/v1>

All HTTP requests must be authenticated with your [APIKey](#). There are no unauthenticated endpoints to the API. Your key should be appended as a parameter to a GET request named apiKey e.g.-

[http://api.bbfc.co.uk/miracle/v1/games/match?title=MIRACLE&apiKey=\[API_KEY\]](http://api.bbfc.co.uk/miracle/v1/games/match?title=MIRACLE&apiKey=[API_KEY])

Response content types

Clients should set the HTTP_Accept header to `application/xml` or `application/json` accordingly. JSON is the default if this is omitted. JSONP is supported by appending the `callback=XXX` parameter to requests. The server will respond with `XXX(response)`.

Errors

HTTP response codes are used where deemed appropriate and applications should respect these. The **Match** endpoint will return an HTTP 200 to indicate a successful request but the query given may return zero results.

Endpoints

Please refer to the live documentation at <http://developer.bbfc.co.uk> as this reference is only guaranteed to be current at the time of publication.

Match

The **Match** endpoint can return multiple age labels but this is presently only supported in JSON and should be considered non-standard behaviour. Clients consuming JSON should expect to receive an array of labels. For XML responses, only the highest scoring match is returned. This behaviour is expected to change in MIRACLE 2.0.

GET */games/match*

Implementation Notes

Match can map BBFC ratings to a given game with a high degree of accuracy when a title, distributor and year are provided. Titles should be stripped of any metadata and be formatted in the way they might be read aloud, i.e. 'The Game' rather than 'Game, The'

Response content types

application/json

application/xml

Parameters

Name	Description	Type	Data type
title	Game title	query	string
distributor	Game distributor or publisher	query	string
year	Year of production or first release	query	string

Error status codes

HTTP status code	Reason
400	Bad request – the parameter values are malformed
401	Invalid API key – a valid API key was not provided
500	Internal server error – There was an error generating a response

GET /features/match

Implementation Notes

Match can map BBFC ratings to a given feature film with a high degree of accuracy when a title, director(s) and year are provided. Including a duration in minutes can improve the accuracy, especially when a rating for a specific context is requested. Titles should be stripped of any metadata and be formatted in the way they might be read aloud, i.e. 'The Movie' rather than 'Movie, The'

Response content types

application/json

application/xml

Parameters

Name	Description	Type	Data type
title	Feature title	query	string
directors	Feature director(s) in a comma separated list	query	string
year	Year of production or first release	query	string
duration	Duration in minutes	query	integer
matchFor	Return the correct rating for use online, in cinemas or for packaged goods	query	string "VoD", "Video" or "Cinema"

Error status codes

HTTP status code	Reason
400	Bad request – the parameter values are malformed
401	Invalid API key – a valid API key was not provided
403	Not licenced – The API key is valid but not licenced to receive this response
500	Internal server error – There was an error generating a response

Details

The **Details** endpoint allows for the direct retrieval of a single dataset in exchange for a unique identifier. These references can be a BBFC codenumber or registration number, which are internal references provided to content owners and 3rd party partners. Alternatively, in rare cases a 3rd party reference can be used but this is typically only available to partners for which we have undertaken a large scale matching exercise. In this instance a partner identifier must be provided with the unique id itself.

GET */details/{bbfc_reference}*

Implementation Notes

A BBFC reference can be either our internal codenumber, or the work's registration number. This method will return details for either ratings or individual decisions depending on the reference supplied.

Response content types

application/json

application/xml

Parameters

Name	Description	Type	Data type
bbfc_ref	BBFC reference for a given rating or decision	path	string

Error status codes

HTTP status code	Reason
400	Bad request – the parameter values are malformed
401	Invalid API key – a valid API key was not provided
403	Not licenced – The API key is valid but not licenced to receive this response
500	Internal server error – There was an error generating a response

GET /details

Implementation Notes

A partner reference is one which has been mapped to a BBFC record.

Response content types

application/json

application/xml

Parameters

Name	Description	Type	Data type
partnerId	A string identifying the name of a partner reference with which the BBFC database can be queried	query	string
linkedId	A linked identifier which has been mapped to an entry in the BBFC database.	query	string

Error status codes

HTTP status code	Reason
400	Bad request – the parameter values are malformed
401	Invalid API key – a valid API key was not provided
403	Not licenced – The API key is valid but not licenced to receive this response
500	Internal server error – There was an error generating a response

Sample responses

A single MIRACLE age label in XML

<http://api.bbfc.co.uk:80/miracle/v1/games/match?title=halo%203&apiKey=DD6CEC795E3ACB8380257DD2004E0829>

```
<?xml version="1.0" encoding="UTF-8"?>
<age-declaration
  xmlns="http://www.miracle-label.eu/ns/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.miracle-label.eu/ns/ http://www.miracle-label.eu/ns/miracle-1-0.xsd">
  <issuer>
    <age-issuer>BBFC</age-issuer>
    <issuer-url>http://www.bbfc.co.uk</issuer-url>
    <issuer-licence>http://www.bbfc.co.uk/digital/licence</issuer-licence>
    <last-change>2013-05-07</last-change>
    <country>
      <country-code>GB</country-code>
    </country>
  </issuer>
  <scope>
    <scope-ids>
      <scope-id class="BBFC-codenumber">AZG238100</scope-id>
    </scope-ids>
    <scope-titles>
      <scope-title class="BBFC-title">HALO 3</scope-title>
    </scope-titles>
  </scope>
  <rating>
    <age>15</age>
    <age-adds>
      <age-add class="BBFC-category">15</age-add>
    </age-adds>
    <age-icons>
      <age-icon class="BBFC-symbol">
        http://api.bbfc.co.uk/symbols/15.png
      </age-icon>
    </age-icons>
  </rating>
  <content-descriptors>
    <cd-opentext>
      <cd-opentext-desc class="BBFCinsightShort">
        Contains strong violence
      </cd-opentext-desc>
    </cd-opentext>
  </content-descriptors>
</age-declaration>
```

An array of multiple age-labels in JSON

<http://api.bbfc.co.uk:80/miracle/v1/games/match?title=area%2051&apiKey=DD6CEC795E3ACB8380257DD2004E0829>

```
[
  {
    "age-declaration": {
      "issuer": {
        "age-issuer": "BBFC",
        "issuer-url": "http://www.bbfc.co.uk",
        "issuer-licence": "http://www.bbfc.co.uk/digital/licence",
        "country": ["GB"]
      },
      "scope": {
        "scope-ids": {
          "BBFC-codenummer": "AZG208484"
        },
        "scope-titles": {
          "BBFC-title": "AREA 51"
        }
      },
      "rating": {
        "age": 15,
        "age-adds": {
          "BBFC-category": "15"
        },
        "age-icons": {
          "BBFC-symbol": "http://api.bbfc.co.uk/symbols/15.png"
        }
      },
      "content-descriptors": {
        "cd-opentext": {
          "BBFCInsightShort": "Strong language, and violence"
        }
      }
    }
  },
  {
    "age-declaration": {
      "issuer": {
        "age-issuer": "BBFC",
        "issuer-url": "http://www.bbfc.co.uk",
        "issuer-licence": "http://www.bbfc.co.uk/digital/licence",
        "country": ["GB"]
      },
      "scope": {
        "scope-ids": {
          "BBFC-codenummer": "AZG129299"
        },
        "scope-titles": {
          "BBFC-title": "AREA 51"
        }
      },
      "rating": {
        "age": 12,
        "age-adds": {
          "BBFC-category": "12"
        },
        "age-icons": {
          "BBFC-symbol": "http://api.bbfc.co.uk/symbols/12.png"
        }
      }
    }
  }
]
```

Outlook: Challenges and business case

Challenges

Licensing

For commercial reasons, we are only able to make a large portion of our ratings data available under a paid licence and other rating bodies may face similar constraints. This is likely to be seen as a barrier by some but even putting aside the fees, the effort required to understand and ensure compliance with the terms of licences, even free ones, may also be a barrier to entry. Terms and conditions are nothing new, almost every web API has them and access can be revoked for breaches but in terms of the BBFC, the legal and political context that underpins our ratings is likely to make it complex to understand what is allowed and what isn't.

In the immediate future, it is difficult to see what could be done to address this as the roots of BBFC age ratings were set long before the availability of content online became a reality but as new rating systems emerge they will at least be designed for the digital world.

Multiple API's, a lack of standardisation

There is a tension between the two implementation contexts of MIRACLE, the distributed approach and the centralised approach, which is not easy to address. Some features of the specification are only really applicable to one context or the other. Furthermore, whilst we have agreed on a MIRACLE standard, in the centralised context there has been no discussion amongst those who need to implement an API as to how we should do so. Therefore, whilst the responses will be in the MIRACLE format, the APIs to get at the data will likely be different. This will hinder interoperability.

We could agree to standardise of course. It is likely, for example, that each API will a method for retrieving an age label by some kind of identifier and we could agree the method signature. E.g.:-

<http://api.bbfc.co.uk/miracle/id?{identifier}>

<http://api.kijkwijzer.nl/miracle/id?{identifier}>

<http://pegi.info/miracle/id?{identifier}>

Even so, an application that wants to consider data from multiple centralised sources will have to query several different services. This may be seen as a barrier. We can look at a centralised point for all MIRACLE queries, but other constraints may make that unfeasible and this is likely only possible with a tightly defined set of services. In the BBFC context, there are some domain specific requirements which wouldn't necessarily apply to other bodies but are nonetheless critical to making our data available.

Unique identifiers

In the centralised context, the single biggest challenge for MIRACLE is the lack of usable unique identifiers for audio/visual media that can be mapped to the correct age-rating labels when they are separated from the content itself.

This is a far larger problem than that which MIRACLE is trying to solve and impacts on different partners to different degrees. In the BBFC context, it is important to us that not only is the rating referring to the correct film but also that the rating used is one suitable

for the application, be that for use on online video platforms or local cinema listings. Whilst technologies such as [EIDR](#) and fingerprinting are looking to address this area, there are no catch-all solutions. Even with EIDR, some degree of mapping identifiers to content still needs to take place. With respect to fingerprinting, some ratings bodies do not even get a copy of the content they are rating to generate a usable fingerprint and it may be prohibitively expensive to bring these techniques to bear “just” to get an age rating.

The matching endpoint of our API was developed to try and tackle this problem but it is not necessarily applicable to other datasets or domains.

Potential as of MIRACLE 1.0

As it stands, MIRACLE offers the greatest potential to those seeking to implement age ratings for the first time, particularly if they are required to consider age-ratings from more than one provider.

Whilst our existing customers will have already secured a licence to use our data, removing a significant barrier to entry, they will likely see our existing data services as sufficient for now. Our JSON API offers a wider range of endpoints and a greater amount of metadata about the content and the rating itself including information on cuts, which is likely specific to the BBFC. There is a limit to how many fields we can add to the existing MIRACLE data model without hindering interoperability. The majority of our data customers at present are seemingly only interested in BBFC data and so won't enjoy the benefits of being able to implement a new source of ratings more readily.

Some of these issues we expect to be addressed with MIRACLE 2.0 where we expect to be able to offer all of our existing endpoints and spend more time on considering the fragmentation of MIRACLE compliant API's.