INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG11
CODING OF MOVING PICTURES AND AUDIO

ISO/IEC JTC1/SC29/WG11
MPEG2013/m29242
April 2013, Incheon, KR

Source **Distributed Multimedia Applications Group (DMAG),**
Universitat Politècnica de Catalunya (UPC), Barcelona (Spain)
Status **Proposal**
Title **MPEG-M application based on mtPlatform**
Author **Jaime Delgado, Jonathan Florido, Silvia Llorente**

Table of Contents

1 Executive Summary

This document presents an update of the application developed by DMAG-UPC to demonstrate the use of MPEG-M services (m28138), as now the middleware used by it has been changed from MIPAMS v2.1 to mtPlatform, developed by mediaTG company in collaboration with DMAG-UPC. The application has been developed to be compatible with Android mobile devices using at least Android 3.2 HoneyComb version.

This document is organized as follows. Chapter 2 presents mtPlatform. Chapter 3 presents the scenario implemented by the demo application. Chapter 4 presents MPEG-M usage in the demo application. Chapter 5 describes in detail the functionality provided by the demo application.

2 mtPlatform

2.1 Introduction

This section summarizes mtPlatform, briefly describing its components.

The demonstration application has been developed by mediaTG company [1] in collaboration with the Distributed Multimedia Applications Group (DMAG) of the UPC [2]. The demonstration is based on mtPlatform, a REST-based web services standards-based platform, also implemented by mediaTG in collaboration with DMAG-UPC.

2.2 Background: mtPlatform

mediaTG media platform (mtPlatform) is a service-oriented secure content management and distribution platform implemented using REST-based web services (REST, Representational State Transfer). In this way, it has the advantages of service-oriented functionality, like the possibility of decoupling into different subsystems, whilst providing a more efficient and light-weight web service infrastructure. mtPlatform encompasses an important part of the content value chain, from content creation and distribution to its consumption by final users.

mtPlatform is an evolution of MIPAMS v 2.1 [3] [4] and includes the following new features and improvements:

- REST-based web services design
- Complete refactoring of the web services provided by the system, according to REST design principles, that is, based on resources and not in remote operations. This refactoring has increased service productivity
- Implementation of new services to provide billing, advanced contract management, transcoding and refactoring of object registration, license, protection and upload / download services.

Figure 1 depicts mtPlatform. A general overview of its components and the different services being offered (in alphabetical order) is provided next.

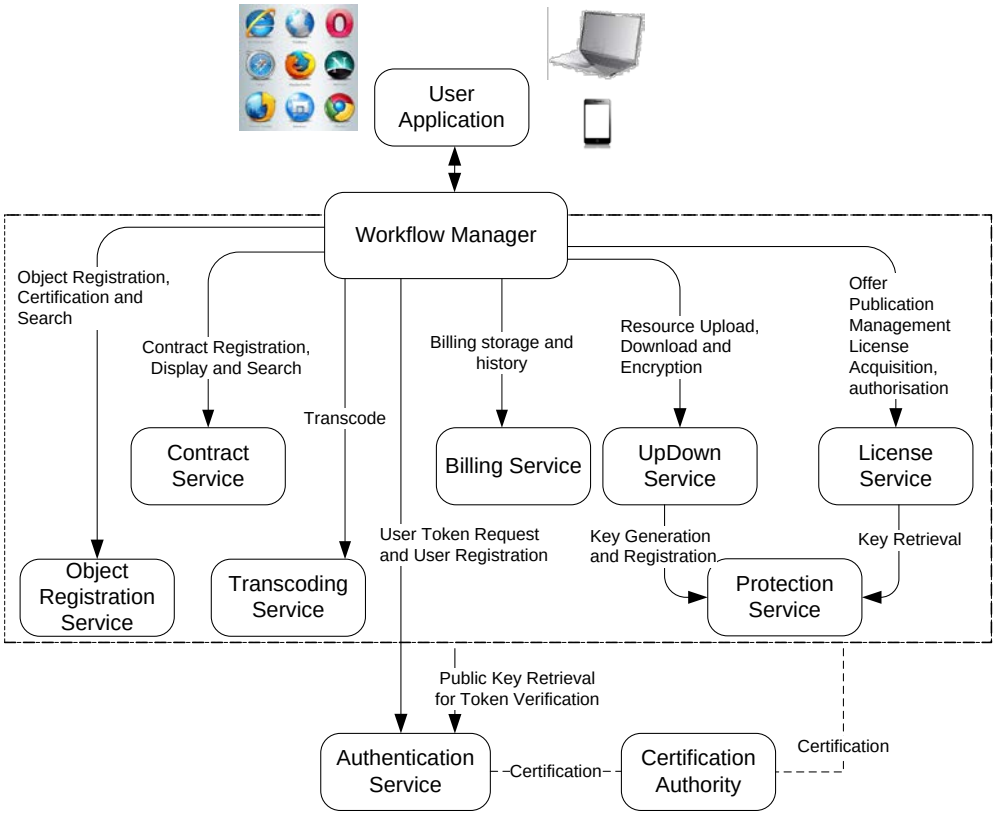

Figure 1. mtPlatform.

2.2.1 mtPlatform Services

The Authentication Service (ATS) is needed to authenticate the identity of users. It generates SAML (Security Assertion Markup Language) based tokens. Any service in mtPlatform requires a token argument to be provided in order to authenticate users. Tokens are digitally signed, so that they can be checked for authenticity and integrity by the receiving service. Moreover, the ATS deals with user registration and management (i.e. personal data modification, user account deactivation, etc.).

The Billing Service (BS) deals with the payment operations originated by the users' operations performed in the platform. There are different kinds of payment information supported by this module, as it is differentiated the payment for the service from the payment for the purchases done (licenses purchased). To do so, different billing conditions are taken into account. The service provides operations for service providers (sellers) like retrieval and generation of bills according to different criteria, like date range or specific user. It also offers operations to service clients (purchasers or buyers) for checking consumption done according to different parameters, like number of purchases or date range.

The Certification Authority (CA), which issues credentials for the different Components and Actors in the system, as X.509 certificates and private keys for the different architectural components.

The Contract Service (CS) manages contracts expressed in MPEG-21 CEL (Contracts Expression Language). It provides different search operations over the contracts stored in the

system. The main objective of these operations is to help contracts' parties in the management of their contracts, providing contract search by country, rights issued, exclusivity, etc.

The License Service (LS) deals with all operations related to licenses and offers: creation, search, authorization of user operations based on licenses she owns and reporting of authorizations performed. Licenses are expressed using MPEG-21 REL, guaranteeing standards support. The reason for this has been a redesign of the service in order to treat licenses and offers as a unique resource, providing into only one service the operations related to them.

The Object Registration Service (ORS) enables applications to request a digital representation of content and metadata (i.e. digital objects) to be generated and registered in the system. It also provides operations for searching objects and reporting of new object creation. Content and metadata are packaged together following the MPEG-21 Digital Item approach.

The Protection Service (PS), as introduced before, generates encryption keys upon request, registers encryption keys associated to uniquely identified content and provides the encryption keys for protected content to the AS.

The Transcoding Service (TS) is used to convert video files into different formats for its visualization into different platforms and devices. It is connected with the UpDown service, as the current transcoding occurs when the file is uploaded into the system. In this precise moment, several versions of the uploaded video files are created, including one version for video streaming in a lower quality than the original file.

The UpDown Service (CS) enables applications to upload and download digital resources such as audio or video files, text documents, etc. There are several versions of this service, depending on the needs of the application using it and the storage options required. Currently, this service has an FTP version, an HTTP version and an Amazon Web Services version. Resources can be optionally encrypted under request, asking for protection keys to the Protection Services (PS), described next.

3 Selected Scenario: Buyer point of view

This section describes a complete buyer scenario, which shows the steps for searching, buying and consuming content, demonstrating how one of the aggregated services described in MPEG-M part 5 can be implemented.

Figure 2 shows a sequence diagram to clarify the order of the operations that can be done from the web application from the buyer point of view.

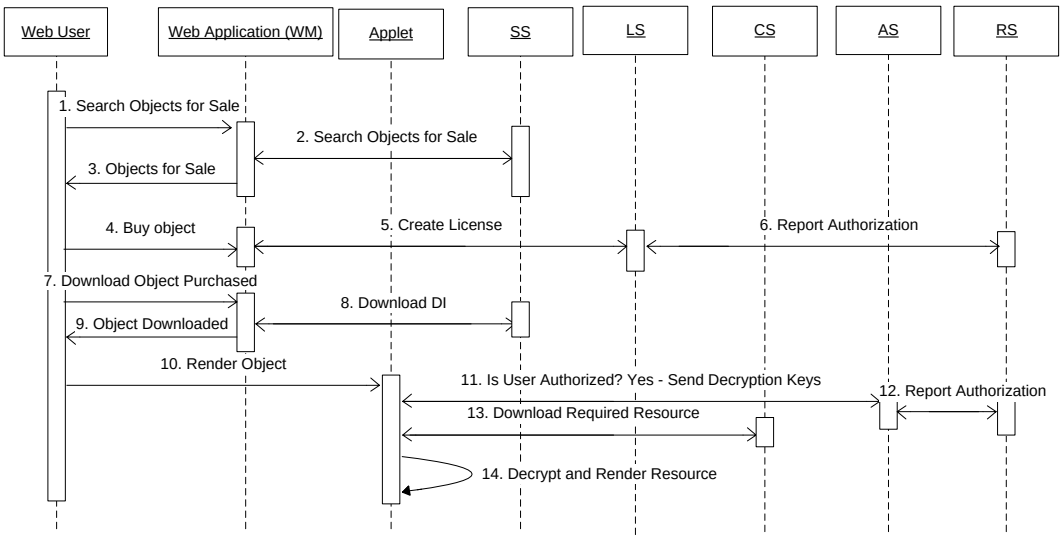

Figure 2. Object search and purchase on the web application. Authorization and rendering of content on the applet.

Operation order is as follows:

1. User selects Search and Buy menu option to look for the objects for sale. They can be filtered by Title and Creator.
2. WM sends object search request to the SS. Then, SS sends the search result to the WM.
3. Objects for sale are presented to the user.
4. User selects an object from the ones presented. She has to select an offer from the ones provided by seller in order to purchase the object.
5. WM asks for license creation to LS.
6. LS sends a report to RS informing of the license creation.
7. User downloads object she has purchased to render its content.
8. WM requests DI to SS.
9. User receives object.
10. User wants to render the object. Applet player is started.
11. Before rendering resource, applet asks AS for user authorization. If authorization is positive, keys for rendering content are provided.
12. User requests a resource, applets gets it from CS.
13. AS sends a report informing of positive authorization to the user.
14. The received resource is decrypted and shown to the user. If user is not authorized, then the resource is not shown and user is informed of this fact.

The login operation is omitted but it should be done before step 1.

4 MPEG-M usage

4.1 Introduction

This section describes how the aggregated service described in subclause 6.4.3 of ISO/IEC 23006-5 (buyer scenario) can be implemented with mtPlatform using an Android-based mobile application.

The mobile application makes use of some Elementary Services in order to build an Aggregated Service that represents the MPEG-M buyer scenario. The Elementary Services used are:

1. Authenticate user: performs the authentication of the user to mtPlatform
2. Search license: allows the user to search content by title or author
3. Present license: shows information about a license
4. Create license: when the user purchases a new license, the system creates a new one in order to grant the user access to the content he has purchased
5. Authorize user: checks if the user owns any license that allows him to perform a right against the content
6. Render content: plays the content

In order to implement the Aggregated Service, we are providing a MPEG-M interface for mtPlatform. In other words, applications will call MPEG-M Elementary Services that will map to kmtPlatform services to be executed.

4.2 Translating between MPEG-M and mtPlatform format

Although MPEG-M and mtPlatform follow the same concepts, their syntax and functionality are slightly different. Therefore, since the input query formats of MPEG-M and mtPlatform are different, a translation mechanism between the two formats must exist in order to perform the mtPlatform operations.

The steps of executing a new mtPlatform operation by using MPEG-M architecture are:

1. An MPEG-M query is created containing the destination ES and the parameters
2. The MPEG-M workflow of the ES finishes at the corresponding Orchestrator Engine. The translation of the query from MPEG-M to mtPlatform format must be done here. The MPEG-M query is translated into mtPlatform format by using some classes called *wrappers.*
3. The corresponding mtPlatform module is called with the translated query.
4. As the mtPlatform response is generated, it must be translated into MPEG-M format in order to generate the proper response.
5. The response is returned.

4.3 Architecture of the application

Since MPEG-M becomes the entry point of the operations against mtPlatform, the architecture adds a layer between the user and mtPlatform. The architecture is depicted in Figure 3.

The MPEG-M is a layer between the user and mtPlatform, and it is the responsible for dealing with MPEG-M and mtPlatform queries in order to translate and generate the proper requests and responses.

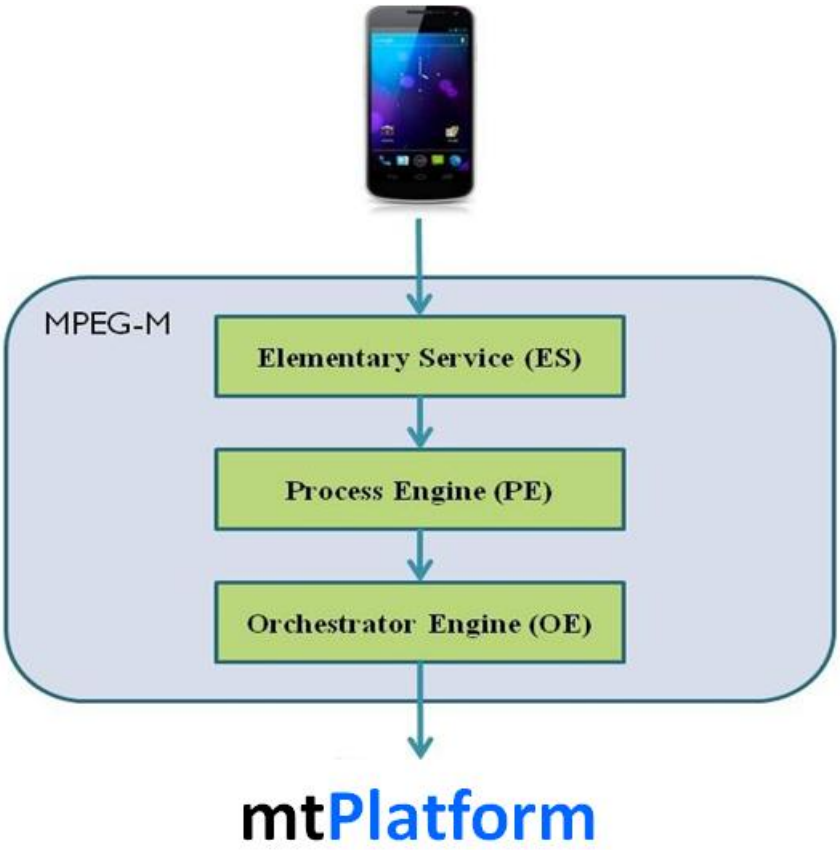

Figure 3. Application architecture.

5 Demo functionality

5.1 Introduction

This section introduces the functionalities the application allows to perform against the mtPlatform services.

The Android application allows the user to perform these actions:

- Create a new account to login into mtPlatform
- Login to mtPlatform
- Search objects by title or author of the content
- Get metadata information about licenses and objects of the system
- Purchase licenses
- Play content

The relation between these functionalities and the MPEG-M ES used to perform these actions is listed in sub-section 3.1.

The application also implements a disconnected mode to keep the application working if the device connectivity is not available. This means that many mtPlatform functionalities need to be implemented in the application, like the authorization of licenses or the login. In addition, the application also implements a synchronization mechanism in order to keep mtPlatform updated of the activity the user has performed in disconnected mode.

5.2 Application usage

This sub-section describes the functionalities the application implements. Some screenshots and explanation are provided.

The login screen is depicted in Figure 4. It allows to login in mtPlatform by providing a username and a password. It also allows creating a new user.

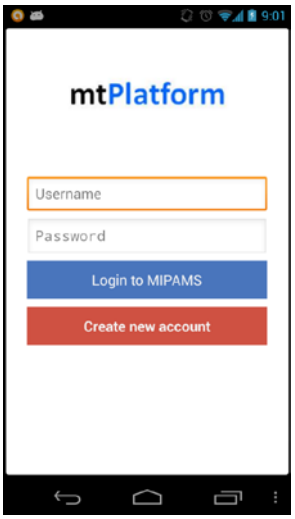

Figure 4. Login screen.

When the user is logged in, the user menu is shown. The menu contains an action bar indicating the username, the number of items the user has purchased in mtPlatform, a search icon and a refresh icon. The user menu is depicted in figure 5.

The screen displays the content the user has already purchased in a scrollable list. The user can render the objects (with a previous license based authorization) and get the metadata of both the purchased Digital Item and its licenses. The refresh icon allows getting again the purchased items.

Figure 5. User menu.

By pressing the search icon, a new screen is displayed. A search bar allows the user to search by author or title of the content. When the user presses the search button, the results are shown in a scrollable list like the user menu. For each result of the search, the user can display both the information about the content or its purchasable licenses. The search menu and an example of search result are depicted in figures 6 and 7.

Figura 6. Search menu.

Figure 7. Search example.

The  icon displays the metadata of a Digital Item. It is accessible from both the user menu and the search menu. The information is presented as follows:

- The next information of the Digital Item is presented at the top: title, creator, description and creation date.
- The information of all the object resources is presented down: title, description and file format.

The licenses of the object can also be displayed if the “View licenses” button is pressed. The screen is depicted in Figure 8.

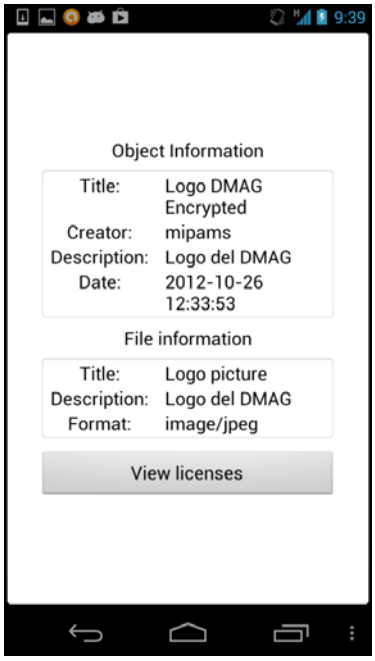

Figure 8. Object information screen.

The  icon displays information about the licenses of the object. If the icon is pressed in the user menu, it displays the information about the licenses the user has purchased for this item. Otherwise, if the user presses the button from the search menu, it shows all the purchasable licenses of this object. The licenses are presented in a scrollable list with the following information:

- License title.
- The amount of Euros to purchase the license.
- A time interval in which the license is valid (i.e. the interval in which the user can render the content).
- The maximum number of views of the content.
- The territory in which the render of the content is allowed.
- A time interval when the render is valid since the first view.

The screen is depicted in figure 9:

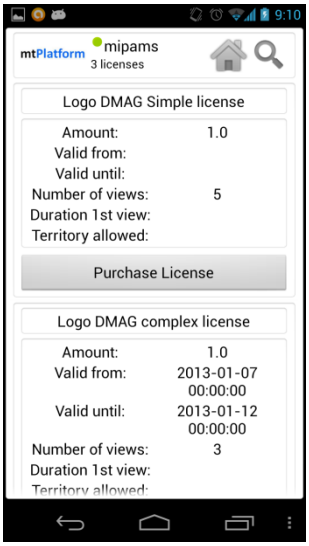

Figure 9. License information screen.

When the user presses the  icon from the user menu, an authorization request is performed. The authorization grants the access to the content if exist at least one license that allows the user to render this object. If the authorization is positive, the content is downloaded and the default Android player opens the file to render it. An example of authorization and content render is depicted in figures 10 and 11.

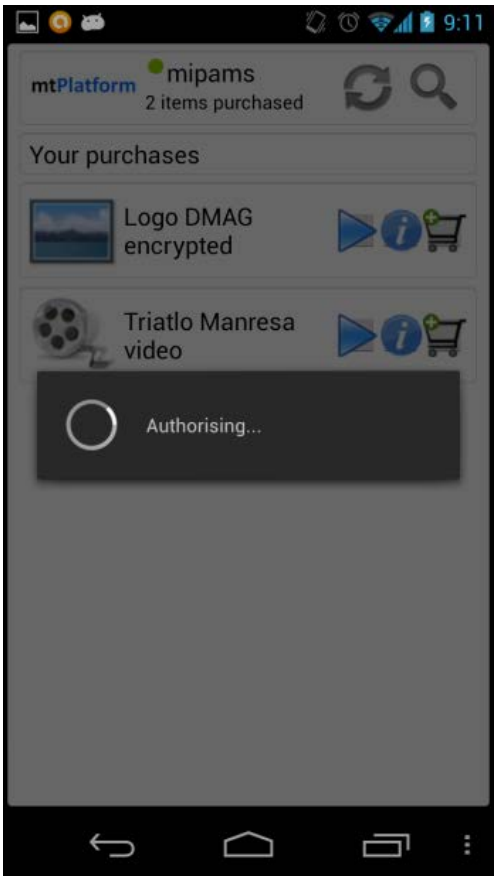

Figure 10. Authorizing the content.

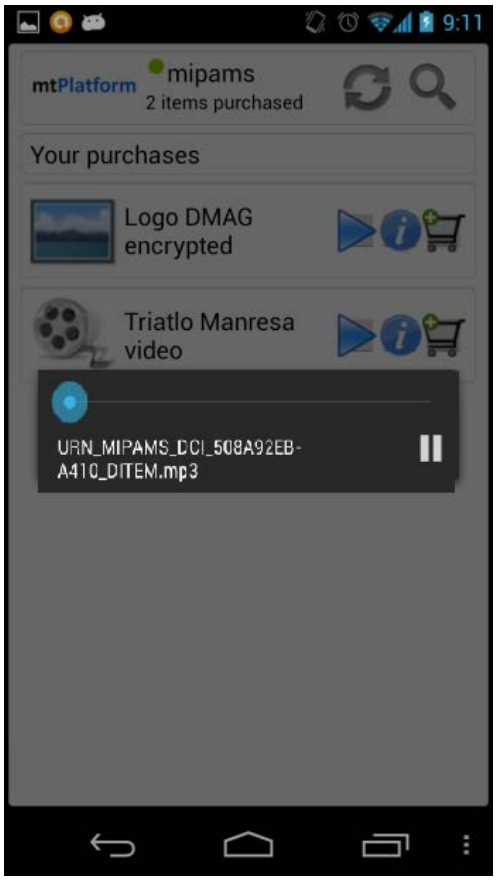

Figure 11. Rendering a song.

6 References

[1] media technology Group (mediaTG), http://mediatg.com

[2] Distributed Multimedia Applications Group (DMAG), http://dmag.ac.upc.edu.

[3] Delgado, J., Torres, V., Llorente, S., Rodríguez, E., Rights management in architectures for distributed multimedia content applications, Trustworthy Internet. Heidelberg: Springer, 2011, ISBN 978-88-470-1817-4.

[4] Delgado, J., Florido, J., Llorente, S., m28138, MPEG-M demo based on the MIPAMS platform.