A service-oriented framework to promote interoperability among DRM systems

Fernando Marques Figueira Filho,
João Porto de Albuquerque, and Paulo Lício de Geus

Institute of Computing, University of Campinas, 13083-970 Campinas/SP Brazil

Abstract. Through the past years, several digital rights management (DRM) solutions for controlled dissemination of digital information have been developed using cryptography and other technologies. Within so many different solutions, however, interoperability problems arise, which increase the interest on integrated design and management of these technologies. Pursuing these goals, this paper presents a framework which aims at promoting interoperability among DRM systems, using a service-oriented architecture (SOA) and a high-level policy modeling approach.

1 Introduction

Digital Rights Management is a collection of technologies that enables controlled dissemination of digital information. Today, the majority of DRM applications are used in copyrighted content distribution, such as movies and music, but it is expected that those technologies will also benefit, in a near future, small content producers and individuals who intend to securely distribute their own information.

Although there have been considerable advances in the area, DRM systems still do not interoperate. There are differences over formats and protocols, as well as difficulties in trying to integrate management while simultaneously operating different DRM systems. Thus, content producers are forced to choose one among available platforms, which affects their content distribution covering. Moreover, the lack of operability can be used to stimulate the monopoly over proprietary software and devices by some vendors, which can be harmful for both users and content producers.

Following this motivation, this paper presents a framework which aims at promoting interoperability among DRM platforms. It is based on the fact that in every platform, the lifetime of contents follows basically the same steps: firstly, it is packaged using cryptography, in order to protect it against unauthorized users. Then, at some moment during content distribution, it is licensed to a specific user or device. A license is a file containing the rights and conditions, described in a platform-specific format, which govern contents' usage by that particular user. Our framework centers those rights and conditions in a single policy-based model, which is generic for every DRM platform.

To that effect, a service-oriented architecture (SOA) is defined, which is responsible for managing those policies and using them to generate licenses in

different DRM platform formats. Services are implemented using Web Services, allowing for easier compatibility with most computer architectures and programming languages.

The next section presents a brief of the conceptual models in which our approach is based. The system architecture is analyzed in Section 3 and we conclude this paper with some related work and expectations around future work in Section 4.

2 Policy Model

In this paper, policies are based in an object-oriented model which can be divided conceptually into levels of abstraction, as depicted in Fig. 1. The highest level is based on the role-based access control (RBAC) concepts [1] and its extension, the GRBAC [2].

Through the past 10 years, RBAC has been used to simplify permission management, especially when users are hierarchically organized or when it is possible to identify common characteristics among them. Such scenario is found in various DRM business models (e.g. service subscription or purchasing, membership of a club or organization). Instead of associating rights with each user, we apply rights to *subject-roles*, which in turn are associated with users. In this manner, a small policy set is sufficient to manage a large and complex system. Thus, policies in the abstract level are relatively static and their construction is supported by a graphical tool, similar to the one used in other policy-based management applications [3].

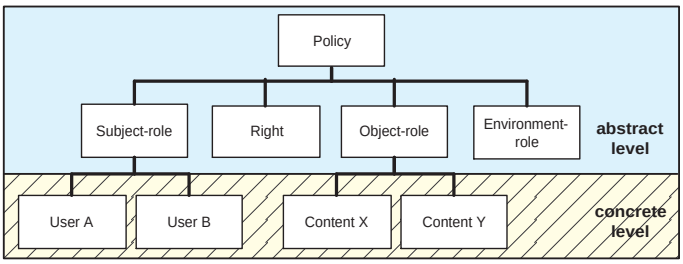

Fig. 1. Policy structure

DRM permissions, however, commonly associate conditions and restrictions to a right (e.g. play, print), based on stateful information. This information is included in the license and used by a particular DRM platform to control, for example, the number of times a user exercises a right, the time interval during which a content can be used, among others. GRBAC extends RBAC through the introduction of *environment-roles*, which are applied to our policy model to

incorporate those state-based conditions and restrictions. GRBAC also defines *object-roles*, which are used to group contents and build policies based on their characteristics, such as type (audio, video etc.) and confidentiality level.

The second abstraction level carries concrete entities from a DRM system (e.g. users, contents) and holds a much more dynamic behavior. While the upmost level is updated by human intervention by means of a graphical editor, the second level is updated by framework services according to the external DRM system activity. The architecture that comprehends these services and its functioning are covered in the next section.

3 Framework Architecture

The framework proposed in this work has a service-driven architecture composed by five services. Some are platform-dependent and interface DRM systems with which the framework operates, while others interact with the policy database, as depicted in Fig. 2.

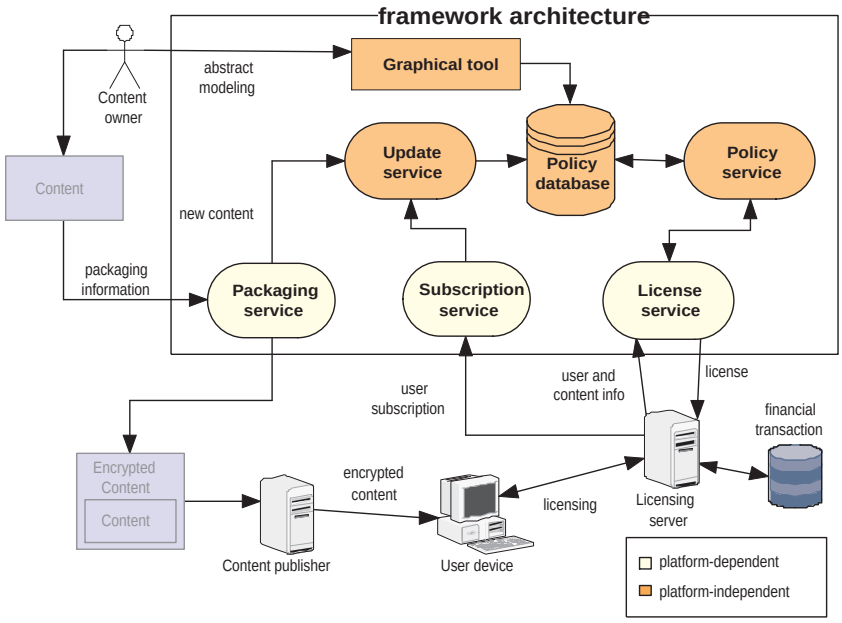

Fig. 2. Framework architecture

In the beginning of the content lifetime it is supposed that abstract policies have already been defined through the graphical tool. After finishing the abstract modeling step, the content can be packaged using a platform-specific

packaging service, which receives a plain file and references to which *object-roles* that content will be associated with. The *packaging service* then requests the update to the *update service*.

On the other hand, users interact by licensing content on a payment-basis or, when there is no financial transaction involved, by only subscribing to new services and having their access levels changed (e.g. when a company that uses DRM to manage classified documents hires a new employee). In these cases, the *subscribing service* receives user information and references to which *subject-roles* that user will be associated with.

Finally, when a license has to be generated in a specific platform format, the licensing server contacts the *license service* which serves that particular platform, passing user and content identifications, as well as some other platform-specific information. The *license service*, in turn, contacts the *policy service*, which searches the database for all policies related to those user and content, returning the results. The *license service* then interprets the returned policies and generates a license.

4 Related and Future Work

Some recent work analyze interoperability issues, sometimes proposing solutions, as in Sun's project called DReaM [4], which also employs a service-oriented architecture. However, none of them uses a policy-based management approach or any abstract modeling technique.

The proposed architecture aims at providing interoperability through a centered, platform-independent policy model, which interfaces to other systems using specialized services that will be implemented using Web Services. The conceptual division of policies in two layers allows for a system view with an appropriate abstraction level. The high-level policy design is also supported by a graphical editor, to be developed using Java and applying the visualization improvements used in [3].

References

1. Ferraiolo, D., Kuhn, R.: Role-based access control. In: Proceeedings of 15th NIST-NCSC National Security Computer Conference, Baltimore, MD (1992)
2. Covington, M.J., Moyer, M.J., Ahamad, M.: Generalized role-based access control for securing future applications. In: 23rd National Information Systems Security Conference Proceedings. (2000)
3. Porto de Albuquerque, J., Isenberg, H., Krumm, H., de Geus, P.L.: Improving the configuration management of large network security systems. In: Ambient Networks: 16th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management, DSOM 2005, Proceedings. Volume 3775 of Lecture Notes in Computer Science., Berlin Heidelberg, Germany, Springer-Verlag (2005) 36–47
4. Fernando, G., Jacobs, T., Swaminathan, V.: Project DReaM - An Architectural Overview. White Paper. Open Media Commons. Available at: http://www.openmediacommons.org/ (2005)