Microsponsorships: A New Payment Model for Music Distribution

by

Gregory Herrera

Dissertation

Presented to the

University of Dublin, Trinity College

in partial fulfillment

of the requirements

for the Degree of

Master of Science in Computer Science

University of Dublin, Trinity College

September 2005

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Gregory Herrera

September 12, 2005

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Gregory Herrera

September 12, 2005

Acknowledgments

I would like to thank my supervisor Mads Haahr for his advice and guidance throughout the dissertation. A special thanks to the Networks and Distributed Systems class of 2005 and to ENSEEIHT who allowed me to study in Trinity Colllege this year. Finally I would like to thank Jenny Kennedy for her help during the last days of the dissertation and, obviously, my family.

Gregory Herrera

University of Dublin, Trinity College
September 2005

Abstract

Microsponsorships: A New Payment Model for Music Distribution

Gregory Herrera

University of Dublin, Trinity College, 2005

Supervisor: Mads Haahr

Today's music industry on the Internet is dominated by the record companies who charge users to download MP3s. In most cases, users do not have the opportunity to listen to the entire song prior to downloading and purchasing it which means they may end up paying for content that they don't really like. Since no refund policy exists under the current system, consumer rights are somewhat compromised. Further to this, the record companies profit greatly from each transaction with the artists themselves only receiving a small percentage of the profits. Thus it is clear that the present model for purchasing music from the Internet is restrictive for both artists and consumers.

The objective of this dissertation was to design a new model for purchasing music from the Internet that would be less restrictive for artists and users alike. The aim was

to produce a working prototype, namely an MP3 player, that would allow the user, at the click of a button, to donate money directly to the artist they are listening to, and to support their favourite artist.

The mechanism of the proposed model is such that when a user listens to a song that they like they can click on a Donate button on the MP3 player which will take them directly to the web site of that particular artist. The web site will contain payment options (in the case of this prototype the online payment system PayPal is used) should the user wish to donate money to the artist. Obviously, such a system is dependant on the artist having payment data stored within the ID3 tag which is located in the MP3. In the working prototype, when this payment data is present in the MP3, the Donate button on the MP3 player is activated. In contrast, if no payment data is contained in the MP3, this button remains inactive.

Because this system allows consumers to support artists directly, it takes away the need for the middle man, in this case, the record companies. Consequently, even unknown artists could use the Internet as a platform from which to promote their music. Needless to say, if such a system were to be put into action, it would encounter a strong opposition from record companies.

While ethical and copyright issues obviously exist with the above prototype, it does provide an example of a less restrictive model for purchasing music from the Internet.

Contents

List of Figures

Chapter 1

Introduction

Nowadays, Digital Rights Management (DRM) controls and restricts the use of digital media content. Today's environment of peer-to-peer file sharing makes it difficult to ensure the rights of the content's owner. With regard to music, the current business model is mainly based on a per song downloaded payment, and some online music stores like iTunes, Real Player or Rhapsody that allow users to purchase songs and albums. However, if a user purchased an MP3 and removed it from his computer by accident, they would have to pay again to download it from the music store. Also, if a user changed his mind about a song a few days after downloading it and did not like it anymore there would be no refund. These restrictions are due to the current business model. Other online music services exist, for example Napster [3], which is based on a subscription model and users are permitted to transfer an unlimited number of songs to their own devices provided they are "Napster To Go compatible". Then, these songs can be listened to only while the Napster subscription is active, and when the user unsubscribes he will no longer be able to listen to them.

Thus it is clear that current business models for music distribution are very restrictive, and, for the most part, favour music companies. For this reason and because the price of Compact Discs (CDs) is too high, there is a tendency for web users to share music for free on the Internet by using Peer-to-Peer (P2P) networks. These practices have become an issue of contempt to music companies and, in order to limit their use, music companies have launched advertising campaigns to fight against piracy. At the same

time, new technologies have been introduced to prevent consumers from being able to copy music from CD to their computers. The concept here was to include DRM in CDs and to provide the CDs with copy protection. Indeed, being able to extract the audio data from a CD to convert them into MP3 files may potentially give rise to piracy because it subsequently allows the sharing of these MP3 files on the Internet. However, if a user buys a CD, they should be allowed to burn it to make a copy for themselves. Once again, DRM stands as a barrier between the consumer and the product, even if these technologies are not quite up to scratch yet. Thus, nowadays, the only means of protection against piracy is the legal action taken by the music companies and the Recording Industry Association of America (RIAA) against people who download copyrighted MP3 using P2P networks.

When it comes to downloading MP3s, there are several different consumer types. On one hand, according to Lawrence Lessig [1], there are those who use shared networks instead of purchasing content. For example, when a new CD is released, rather than buying the CD, these users simply download it. In this instance, it could be argued as to whether everyone who downloads songs or albums from the Internet would actually have bought them if they had to pay the full price. Clearly there are some who would and some who definitely would not. On the other hand, there are those who use sharing networks to listen to a sample of music before purchasing it. The result of using this type of shared network would be to increase the quantity of music that would be purchased. In this case, the Internet may serve as a means of promoting a given artist and encouraging the sales of their CD. Finally, there are many who use sharing networks to get access to copyrighted content that is no longer sold or that they would not have purchased in stores because the transaction costs off the Net are too high. This use of sharing networks is among the most rewarding for many as it enables people to download songs from their youth that may be difficult find in stores.

Thus, downloading MP3s is not merely a way to avoid buying CDs. In the last few years, the music companies have identified a decrease in CD sales and their solution is to restrict the distribution of MP3s on the Internet. This practice is very restrictive and aggressive, and may not be justified. Indeed, the decrease in CD sales may not be directly linked to the use of MP3s since the majority of people do not mind paying

for CDs with content that they like. The notion of copyright used nowadays seems rather out of date, and it may be time to revise it. On the other hand, there is another way of thinking that supports the idea of putting music and culture within everyone's reach. Under this way of thinking, MP3 files would be freely accessible to everybody. Obviously, this point of view cannot be considered as a realistic one, but a fair solution would be at the half way point between these two concepts.

Moreover, unknown artists who want to become famous or known use the network as a way of spreading their music. They do not expect people to pay for it, instead, they just make it available for free. However, they could, potentially, use it as a way of raising money to release a CD later, but no system exists today to allow users to donate money to this kind of artist. This principle of sponsorship seems interesting and requires further development. One can imagine a user playing a song in an MP3 player that would allow them at any moment to donate money to the artist they are listening to. This microsponsorship model could be used as an alternative business model for music distribution.

The objectives of this dissertation are to design a novel system for music distribution that would enable users to pay only for the music that they really wished to purchase and to give users the option to support their favourite artists. The aim is to produce a user friendly working prototype, namely an MP3 player, that would allow the user, at the click of a button, to donate money directly to the artist they are listening to.

The proposed system, as a new business model for music distribution, may encounter strong opposition from the music industry. It is important to bear in mind, however, that the music industry has historically reacted strongly against new technologies that later turned out to be beneficial to them. Examples of this would include the music cassette and cable TV. There is therefore a need to explore business models for network distribution of digital music.

This project may open some doors and provide some ideas on finding a fairer way to distribute MP3s, both for the artist and the consumer. The project does not aim to resolve the issues of copyright involved in MP3 sharing on the internet, but to show

that a less-restrictive solution could be technically feasible. Thus, it will investigate an alternative business model, completely different from today's model. In the next chapter, the current trends and the main applications for music distribution will be described in further detail. Then, a design of the application will be demonstrated, and in the following chapter the different technical solutions will be discussed. Last but not least, an evaluation of the developed application will be given as well as an analysis of the advantages and disadvantages of a new microsponsorship model.

Chapter 2

State of the art

As stated in the introduction, nowadays music distribution on the Internet is via online music stores such as iTunes, whose music store version was launched in 2003 [2]. iTunes is a combination of a player and an online store. Today, iTunes is the leader in music sales on the Internet with more than 2 million songs in store. With iTunes, Apple found a solution that seems to be fair both for the artist and the consumer. Users can listen to a 30-second preview of a song and, if they are interested in it, they can then choose to buy it with the click of a button. Songs purchased from iTunes are copy protected with Apple's digital rights management implementation called 'FairPlay' that produces songs in an Advanced Audio Coding (AAC) format. 'Fairplay' is an Apple's proprietary implementation, not AAC that is a standard. The user owns the songs purchased forever, and they can even burn a CD legally if they want. One of the other advantages of iTunes is that the user can buy two or three songs of a particular artist without having to buy the whole album. Finally, iTunes is much more than an online store because it allows users to organize their music and store all of their songs in the same jukebox.

However, iTunes did not resolve all of the problems associated with the current business model. Firstly, the user still has to pay for music that he likes to listen to occasionally even if they are not very interested in it. Providing a 30 second preview of a song is not always sufficient to make up one's mind about whether or not to purchase a song. Also, if a user purchased a song and realised a few hours later that they did not really like it

there is no way of getting a refund or exchanging this song for another one. Likewise, if a song was accidently removed from the computer, the user would have to pay again for getting the same content. Another major drawback of iTunes is that it is only available on Mac OS X and Microsoft Windows platform, that is to say Linux platform users can't access any version of iTunes. Further to this, the digital rights management technology 'FairPlay' which is used in iTunes is a proprietary implementation of Apple which prevents iTunes downloads from being played, for instance, on Microsoft digital music players and vice versa. In the USA, the Congress has been considering a plan to outlaw music that is protected by proprietary DRM technology like 'FairPlay', to allow interoperability between the different music services. However, many of the industry representatives disagreed with this plan and advised the government to take a hands-off approach to the digital music industry. Once again, it seems that the music industry wishes to retain the right to make decisions in order to preserve their restrictive business model. Last but not least, iTunes uses huge servers to store all the songs available and, as a result, there is an obvious problem with regard to scalability. Moreover, the management of such big servers is not only very expensive, but is very difficult as a great deal of vigilance is required to inter-operate the different databases.

At the same time, some users do not want to pay for music because of the price of the CDs in stores and because of the restrictive nature of the current business model on the Internet. Thus, they have opted to use peer-to-peer (or P2P) networks to share their music and download the songs they like or they want to listen to occasionally free of charge. The P2P revolution was launched in 1999 by Napster [3] and allowed users to share MP3s on the Internet. Napster had to shut down in 2001 after a number of lawsuits concerning copyright laws, and other file sharing programs like KaZaA, eMule, bitTorrent and freenet emerged to improve and replace the original Napster. P2P technologies are not only a way of sharing MP3s, and any digital content can be shared in the same manner. Because it is not only a means of sharing MP3s but also other types of files that are not protected by copyright, it has proven very difficult for music companies and the RIAA to fight against this technology.

P2P technologies are not only used by a minority of individuals to download MP3s. This practice is commonplace and so, a description of this technology is warranted.

A peer-to-peer computer network is a network that uses the computing power and bandwidth of the participants in the network instead of concentrating it within a few servers. In this case, P2P network is an extension of the concept of the client-server model as all of the computers in the network behave as clients and servers for the other nodes in the network. Different protocols and applications exist with this technology, but they are all more efficient than a client-server model in spreading digital content to a large number of users.

The current business model for music distribution does not use the efficiency of P2P networks. In fact, on the contrary, it attempts to make them disappear. However, one would wonder whether it would be possible to use the technological advances to develop another model for music distribution. The goal of this dissertation is to develop an initial approach to a model using a sponsorship-oriented solution, where P2P technologies would have an important role to play in music distribution. In the proposed system, people would download songs from any file sharing application on the Internet, and, when playing a song on their MP3 player, they could donate money to the artist if they wished to support them in creating another album for example. This model would be fairer for users as they could now select the songs that they really wish to pay for. Moreover, it would give fans the option of supporting their favourite artist. For such a model to be feasible, MP3 players would have to be extended to permit a direct online secure payment method from any user. Obviously, the security and scalability of this solution would be of up most importance.

This microsponsorship model has never been implemented before, that is to say that this project presents a novel idea and can open doors in an area where music companies, for the time being, are the main dictators. Among the existing MP3 players, for example Windows Media Player and Winamp, there are none that have the functionality that allows users to donate money directly to the artist. Also, because of the mini-crisis surrounding music companies, it is nowadays more difficult than ever for an artist to get a record deal. Thus, this microsponsorship concept would be even more interesting for unknown artists who do not have the backing of any music company because the model promotes support for the artists themselves rather than for music companies. Consequently, this simple and flexible model developed for the purpose of

this dissertation has some important advantages for the music industry on the Internet. It would appear that the only group who would be negative towards such a concept would be the music companies themselves.

Chapter 3

Design

3.1 Microsponsorship mechanism

The primary requirement of the microsponsorship model is that it has to be user friendly so that users do not need any special computing skills to donate money to any artist. After all, if the model was difficult to operate, the likelihood is that people would not use this system at all because of its complexity. Thus, the concept was to add a 'Donate' button to an MP3 player that would become activated if the MP3 playing contained some payment data. A user could press this button at any time, while playing a song, to send money to the artist they are listening to. This transaction must be secure, and for this project the secure online payment system PayPal was used to transfer money. Obviously, the model could use any other method of payment such as credit card payment for example, but this was not implemented in the prototype developed for this dissertation. Future extensions for other methods of payment were taken into consideration, however, and as a result, the architecture was developed in such a way that the addition of any new methods of online payment could be easily implemented.

Once the user has clicked on the Donate button in the MP3 player, they arrive on the artist's official web page. Here, they can choose any online payment system, and for using PayPal for instance, they would click on the PayPal Donate button. Finally, the user enters the amount of money they wish to donate and their PayPal login in-

formation to complete the transaction.

The following flow diagram summarizes the microsponsorship mechanism:

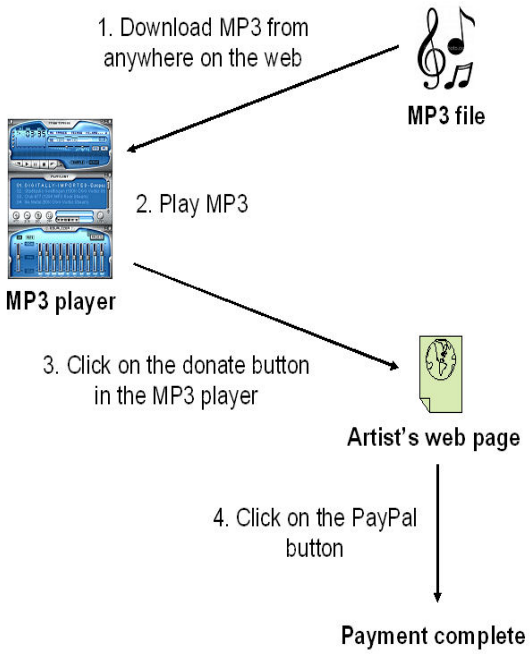

Figure 3.1: Microsponsorship mechanism

The payment data that needs to be stored and secured within the MP3 player in order to complete a transaction with the artist is actually the Uniform Resource Locator (URL) of the artist's official web page and the methods of payment that are accepted by this artist. Even though it was not necessary to store information relating to different payment methods that the artist would accept in this proposed mechanism, it was added in anticipation of future extensions of the MP3 player. For instance, a future extension could be to allow payment to an artist only if they accept the payment systems that the user has configured earlier within the MP3 player. The next section will describe the strengths and weaknesses of the two types of architecture that could potentially have been used to design the application.

3.2 Architecture of the system

The design of the application is very important as this defines its architecture and determines the technical solutions that could be implemented in order to guarantee that it works well and is secure. There are two possible solutions that could be used for this purpose. On the one hand, the application could use a centralised model, and a description of this solution will be given in the first section of this chapter. On the other hand, a distributed solution could be used. Such a solution would be considerably more scalable. The distributed solution will be discussed further in the second section of this chapter. Because of this issue of scalability, the distributed model is the one that was adopted for this project.

3.2.1 The centralised solution

To remedy the problem of the artist's account authentication, the easiest solution would be to use a central server to identify the official URL of the artists. The central server would behave as a huge database that would contain hashes of MP3 files with the matching URL of the artist's website. If this solution is implemented, the MP3 player would have to be configured with the URL of this central server. When playing a song, the MP3 player would create a hash of the MP3 file and would send it to the central server which would reply with the URL corresponding to the hash received. The communication between the MP3 player and the central server needs to be secured in order to ensure the integrity of the data. Indeed, if the data was not encrypted, it would allow a hacker to send a fake URL in reply to the hash sent. Thus, the dialogue between the server and the MP3 player would use public and private key encryption to ensure the integrity of the data. The first connection would consist of a key exchange. Then, should the user wish to donate to the artist, the MP3 player would create a hash of the MP3 file, encrypt it with the server's public key and send it to the central server. The server would decrypt the message, extract the hash, find the URL corresponding to the hash in the database, encrypt it with the user's public key and send it back to the user, signing the message with its private key. The player would then verify the authenticity of the server using its signature and contact the artist's web server, where, by clicking a button on the artist's web site, the user would donate money to them by way of an online payment system such as PayPal.

This process is summarised in the following diagram:

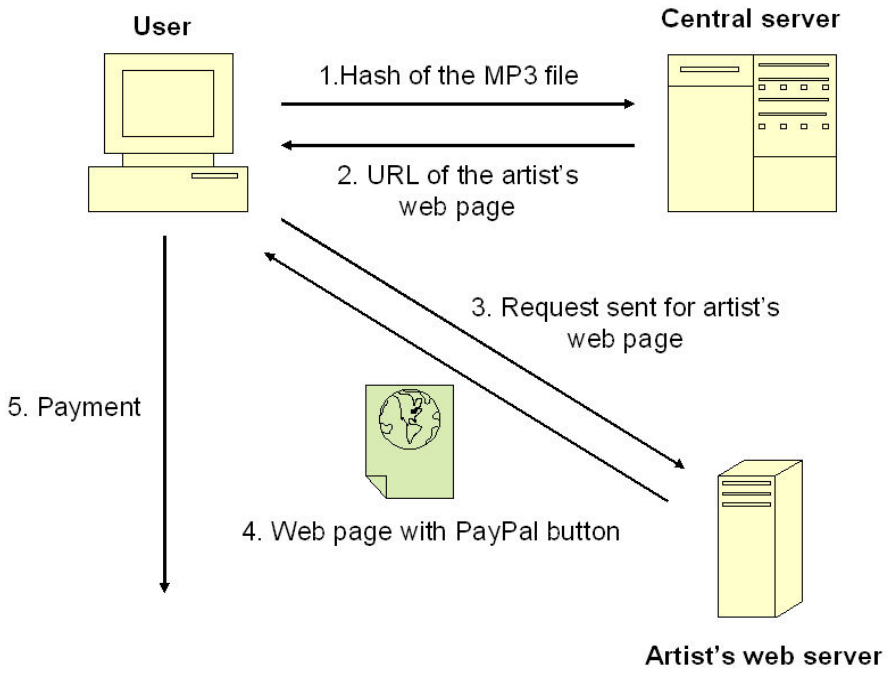

Figure 3.2: The centralised architecture

This would appear to be the easiest method but it has some limitations. Firstly, the centralised architecture of the application creates a bottleneck at the server since interaction with the server is required to pay for any song of any artist. As a result, there would be a great deal of latency as a lot of users would try to connect to the server at the same time. Also, there is the problem of scalability. That is to say that the large number of songs from different artists can make the management of a database very difficult. Finally, by storing the hash of an MP3 in the central database, it would not be possible to get the payment data of an artist with an MP3 representing the same song but with a different encoding for example. Because of all of the drawbacks listed above, a distributed solution was selected for the microsponsorship model.

3.2.2 The distributed solution

A more flexible and scalable way of designing the architecture of the application is to use a distributed model. In this case, the payment data is contained in the MP3 itself, using a storage element called ID3 tag. The ID3 tag is where the information about the MP3 file is stored, for example: the title of the song, the author, the name of the album, and so on However, the use of the ID3 tag increases the need for security because it can be easily modified by anybody. The issues of security in relation to the distributed solution will be discussed in the next section. At this stage, it is not possible to give a design of the distributed system because it relies on the technical solutions used to implement it.

3.3 Security issues

There are a number of security issues regarding the online payment. As seen in the previous section, the distributed solution uses the ID3 tag of the MP3 in order to store the payment data. The problem with this tag is that it is easy to modify and lots of editors and libraries exist on the Internet that are free of charge. Thus, the chosen technical solution must ensure the authenticity of the MP3 issuer and the integrity of the payment data. Indeed, a third person could unrightfully claim another artist's song, store their own payment data in the ID3 tag and earn the money without anyone's knowledge. This is the main security issue and cryptography will play an important role in its prevention.

In addition, the payment transaction itself must also be secure. This is not a real problem for the proposed application because it will use an existing online payment system like PayPal [4]. PayPal supports the SSL protocol to encrypt data and to authenticate the PayPal server. The user has then to log in and enter a password to authenticate them. Thus, PayPal allows the authentication of both endpoints and the confidentiality and integrity of the transaction. Micropayment schemes will not be dealt with in this paper because no real micropayment system exists today. In any case, the security in such a system would be less important as the amount of money involved in micropayments is very low.

In the next part of this paper, possible implementations to solve the above issues will be discussed. A detailed description of the application and of the different steps to donate money to an artist will also be explained.

Chapter 4

Implementation

This section of the report will discuss the implementation choices that were encountered during the development of this application. Firstly, the technical solutions that exist to implement the distributed model will be considered. The working of each component of the proposed prototype will then be explained. This will include a description of the different interactions between the artist, the user and a third element that will handle the security of the model. In addition to this, a description of the ID3 tag will be given in relation to the storage of the payment data.

4.1 The distributed solutions

As seen in the previous chapter, the main concern with regard to the distributed model is the security. The use of the ID3 tag to store the payment data makes the authentication of the MP3 issuer (the artist) the main priority. Consequently, one solution comes to mind to authenticate the artist: the use of X.509 certificates. But the authentication of the artist is not the only concern and, indeed, additional information will have to be stored in the element that is used in order to guarantee the authentication of the artist. The additional information in question is the payment data that are used to donate money to the artist (a more detailed explanation of the payment data will be given later). For this purpose, another technical solution exists: Security Assertion Markup Language (SAML) [6]. The payment data is personal to the artist, and this is the reason why it has to be stored within the certificate (or the SAML assertion).

In this way, it would prevent any hacker from modifying it as, if this was the case, the signature of the certificate (or of the SAML assertion) would no longer be valid.

However, storing a X.509 certificate or a SAML assertion alone in the ID3 tag would not be sufficient. Indeed, it would be very easy for a hacker to replace any artist's certificate or assertion by their own and nobody would know the identity of the hacker. It is for this reason that another element has to be stored in the ID3 tag: the artist's signature of the MP3 file. The signature used is a public key digital signature and the artist signs the MP3 file with their private key. To verify the signature, the artist's public key, which is contained in the certificate or the assertion, is used. Thus, in order for the artist to confirm that they were responsible for the storage of their own payment data in the ID3 tag, their public key which is contained in the certificate or the assertion has to collaborate with the artist's private key. The private key is unique to each artist and the artist is the only person who has access to their private key. Obviously, the artist will only sign the audio data of the MP3 file. If this were not the case, ie: if they signed the entire MP3 file including the ID3 tag then even by changing the title of the song for example this would render the signature of the artist invalid.

Thus, for the distributed solution to work effectively and to guarantee the security of the microsponsorship model, two elements will be stored in the ID3 tag: the artist's certificate (or their assertion) and the artist's signature of the MP3. To determine whether it would be more appropriate to use a X.509 certificate or SAML, a more in depth description of these two elements is required and this will follow in the next section.

4.1.1 X.509 Certificate

A certificate is an encoded digital document that declares the binding of a public key to a given entity. A certificate is issued by a Certificate Authority (CA) that signs it with its private key. Then, an entity can check whether the certificate is valid by verifying its signature with the Certificate Authority's public key. This will prove that the certificate was issued by the Certificate Authority. The next step is then to make sure that the expiration date of the certificate has not been reached. Thus, a certifi-

cate allows verification that a given public key does in fact belong to a given individual and helps to prevent someone from using a phoney key to impersonate someone else. In 1988, the International Telecommunication Union - Telecom (ITU-T) created the X.509 certificate standard which became the most widely accepted format for certificates [5]. Since 1996 and X.509 certificate version 3, it has become possible to add extensions to X.509 certificates. This is vital for the application developed for this dissertation because the artist's payment data have to be stored inside the certificate itself. Thus, the certificate will contain both authentication of the artist themselves and their payment data.

A X.509 certificate version 3 consists of the following fields:

- Version
- Serial Number
- Signature Algorithm Identifier
- Issuer Name
- Validity Period
- Subject Name
- Subject Public Key Information
- Issuer unique identifier
- Subject unique identifier
- Extensions

All of the data in a certificate are encoded using two related standards called ASN.1/DER. Abstract Syntax Notation 1 (ASN.1) is used to describe the data while the Definite Encoding Rules (DER) is used to describe a single way to store and transfer that data. Used together, these standards are powerful and flexible in handling data.

Hence, the MP3 issuer (the artist) will have to register first with a Certificate Authority to obtain a X.509 certificate that they will attach to the MP3 file in the ID3 tag along with their own signature of the MP3 file.

A summary of this is given in the following illustration (the ID3 tag will be analysed in greater detail in following sections):

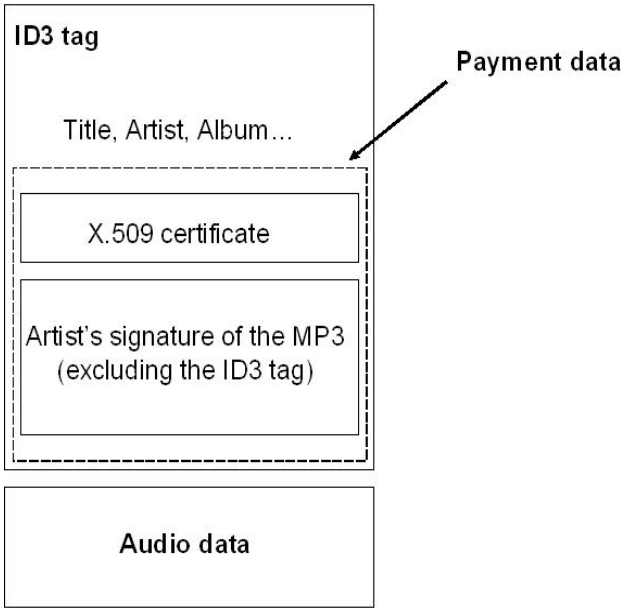

Figure 4.1: X.509 certificate in ID3 tag

When playing a song, the MP3 player will search for payment data in the ID3 tag. If payment data are present, the MP3 player will:

- Check the validity of the X.509 certificate
- Verify the artist's signature of the MP3 file with the artist's public key which is obtained from the X.509 certificate

If these two points are verified, the Donate button will become activated in the MP3 player. The user will then be able to donate money to the artist by clicking on this button. In this model, there is no interaction between a Certificate Authority and the

user (through their MP3 player). The only person who interacts with the Certificate Authority is actually the artist.

4.1.2 SAML Assertion

SAML (Security Assertion Markup Language) is an XML-based language for transferring both authentication and authorization information.

It appears to be similar to an X.509 certificate solution in some ways except that:

- A certificate is now called a SAML assertion
- A Certificate Authority (CA) is called a SAML Authority

SAML was designed to allow XML Signature and XML Encryption standards from the World Wide Web Consortium to be used. Thus, as for a X.509 certificate, the SAML Authority will sign the assertion to prevent forgery. A SAML document contains a SAML element, and within this element, 3 different statements are possible. These statements are listed below:

- Authentication statement
- Authorization decision statement
- Attribute statement

All of these statements are optional, so it is very easy to adapt this technology to the requirements of the application that was developed for this dissertation. As is the case when an X.509 certificate solution is used, the artist would have to create an account on the SAML Authority. They could then request an assertion that they would attach to the MP3 file in the ID3 tag along with their signature of the MP3 file. However, unlike with the X.509 certificate solution, the assertion and the artist's signature of the MP3 file would be stored in the same XML file.

When playing a song, the MP3 player will search the ID3 tag to see if some payment data have been stored. If payment data were present, the MP3 player would:

- Check the validity of the SAML assertion
- Verify the artist's signature of the MP3 file with the artist's public key which is obtained from the X.509 certificate

If these two points are verified, the Donate button will become activated. Payment by the user follows the same principle as for the X.509 certificate. Finally, like with X.509 certificates, there is no interaction between the user (via the MP3 player) and a SAML Authority.

A summary of the storage of the payment data in the ID3 tag using SAML is given in the illustration below:

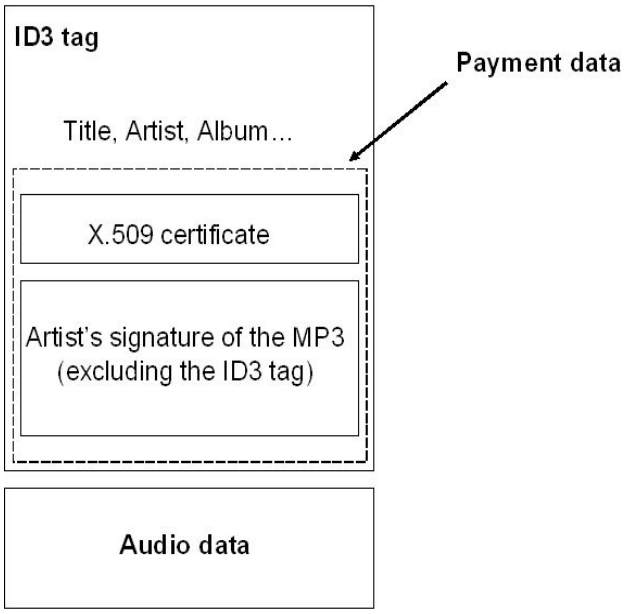

Figure 4.2: SAML assertion in ID3 tag

4.1.3 Comparison

As described above, two technologies exist to make implementation of the distributed model possible: the X.509 certificate or the SAML assertion. After a brief overview of

these technical solutions, it was considered that SAML assertions were more flexible than X.509 certificates. Firstly, a SAML assertion whose element tag is ¡assertion¿ can be used within any XML file. It would permit storage of different information in the same document. For instance, the assertion and the artist's signature of the MP3 file could be stored in the same XML document. Moreover, SAML benefits from some of the advantages of XML, namely:

- It is very easy to create and parse an XML document
- XML is platform and vendor independent

To put it in a nutshell, the SAML solution appears to be more suitable for the proposed application due to its flexibility and the simplicity of the XML language. Furthermore, it would be better to have the artist's signature of the MP3 file, the authentication of the artist and the payment data all contained within the same XML document as is the case when the SAML solution is used. For the reasons mentioned above, SAML was used to implement the distributed model. The next section will describe in detail the payment information that needs to be stored and in the subsequent section, the storage of the payment data within the ID3 tag will be explained.

4.2 Payment data

In order for the user to be able to donate money to the artist they need to know where they can pay them initially and then, how they can pay them. Firstly, when the user clicks on the Donate button, the web page of the artist they are listening to is immediately displayed. The URL of the artist's web page is where the user can donate money to the artist. This web page, in turn, displays the different payment options giving the user the choice of the payment system they wish to use. For the purpose of the application that was developed for this dissertation, it was not necessary to store the different payment options accepted by the artist in the payment data that is contained within the MP3 file. This is because all that was required of this application was that when the user clicked on the Donate button on the MP3 player, they would be brought directly to the artist's web page from where they could pay the artist using different payment options (PayPal was the only payment option available in the prototype developed). However, in anticipation of future extensions that could be potentially made

to the MP3 player, the different payment methods accepted by the artist were stored in the payment data. For instance, one could imagine an MP3 player that would allow the user to configure the payment systems with which they wished to pay. When playing an MP3, if the payment methods selected by the user were not accepted by the artist (or if the MP3 contained no payment data at all), the Donate button would remain inactive.

As seen in the previous section, the technology chosen to provide security to the application is SAML and the use of a SAML assertion. The assertion would be stored within an XML document along with the artist's signature of the MP3 file. An example of an XML document generated for an artist can be seen in Appendix 1 (page 41).

As shown in Appendix 1, the root element of the XML document is termed 'PaymentData'. Within this root element are nested two other elements, namely, 'Assertion' and 'ds:Signature'. The 'Assertion' element represents the assertion that the artist requested from the SAML authority and this contains, primarily, the artist's public key and the payment data for that artist. The 'ds:Signature' element contains the artist's signature of the MP3 file. The 'Assertion' element will now be described in further detail. Firstly, inside the'Assertion' element an'AttributeStatement' was used. This tag was chosen above the 'AuthorizationStatement' as it would permit the storage of information such as the URL of the artist's web page and the payment methods accepted, information which could not have been stored should the 'AuthorizationStatement' have been used. The 'AttributeStatement' contains a 'Subject' element that, in turn, contains the name of the artist and the information about their public key. In the given example, the artist's public key is an RSA. In this case, the 'AttributeStatement' contains two 'Attribute' elements. These are 'PaymentSystems' and 'PaymentUrl'. These elements store the payment data about the artist. Should the artist accept a payment system other than PayPal, an additional 'Attribute' termed 'PaymentSystems' would have to be added. The attribute statement is not the only element contained in the 'Assertion' element. It also contains a 'ds:Signature' element. This is the SAML Authority's signature of the assertion. This element contains the canonicalization algorithm, the signature algorithm and the digest algorithm used during the signature. It also contains the signature value that the MP3 player will verify when playing an

MP3. Indeed, the SAML Authority's certificate that contains the SAML Authority's public key will always be a component of the MP3 player and, thus, the two will be installed together.

The 'ds:Signature' element used for the assertion's signature actually has the same format as the one used for the artist's signature of the MP3 file. The only difference is that within the assertion's signature there is an element termed 'ds:Transforms' that is not present in the artist's signature of the MP3 file. This element contains a list of 'ds:Transform' elements that describe the transformation algorithms used to transform the data that is to be signed before it is digested. As the assertion's signature is an enveloped signature, the enveloped transform algorithm is used so that the signature element itself is removed before calculating the signature value.

Finally, the entire XML document will be stored in the ID3 tag of the MP3 file, and an explanation of how this storage is achieved will be given in the next section.

4.3 ID3 tag

The XML document described in the above section has to be stored in the MP3 file to allow donation of money to the artist while playing an MP3. A storage element called an ID3 tag actually exists in any MP3 file for this purpose. The ID3 tag is a block of data that is added to the audio data of an MP3 file in order to store some interesting information such as the title of the song, the name of the artist, the name of the album, and so on. Two versions of this tag exist today: ID3v1 and ID3v2. ID3v1 was created in 1996, and its size is only 128 bytes. The 'Comment' field could have been used for this project for storing the payment data, but its size is limited to only 30 characters!

ID3v2 was created in order to be more flexible than ID3v1 and to allow the user to extend the tag by adding frames to it. The ID3v2 tag is situated at the beginning of the MP3 file and some frames already exist in the tag by default. One of these frames was cosidered promising for the purpose of the proposed application and is termed 'WPAY'. This frame is defined in the ID3v2.4 tag documentation as: "The 'Payment' frame WPAY is a URL pointing at a web page that will handle the process of paying

for this file"[7].

So, this frame was created for the same purpose as the proposed application: to pay the artist. Despite the fact that its definition describes the frame as containing a URL, a longer string was stored since the size of each frame was limited to 16MB. Thus, the frame 'WPAY' was used for the proposed application to store the XML document which is shown in Appendix 1 (page 41).

4.4 SAML Authority

As the solution chosen for the distributed model was SAML, a SAML Authority had to be set up. The goal of the SAML Authority is to send a SAML assertion in reply to an artist's request. The SAML assertion contains some important information about the artist, namely the artist's name, the artist's public key, the URL of the artist's web page and the payment systems accepted by them. Thus, in order to construct an assertion, the SAML Authority needs to know this information.

The first step for the artist is to create an account on the SAML Authority and to fill in most of the above details (except for the public key).

The creation of an account is summarized in the illustration below:

The URL of the artist's web page and the payment systems accepted by the artist should be modifiable at anytime in order to allow the artist to update this information. The creation of an account on the SAML Authority will provide the artist with a login and a password which they could use to request an assertion. Additional information is required during the creation of an account in order to have all of the important details of each artist registered to the SAML Authority. For example, the artist's address, their phone number, their e-mail address are stored and should be verified before the artist's account is validated. This is very important in the case of theft of another artist's MP3 by a hacker and this will be highlighted in '5.2 Security threats'. Finally, the communication between the artist and the SAML Authority must be secure to prevent hackers from modifying the artist's data.

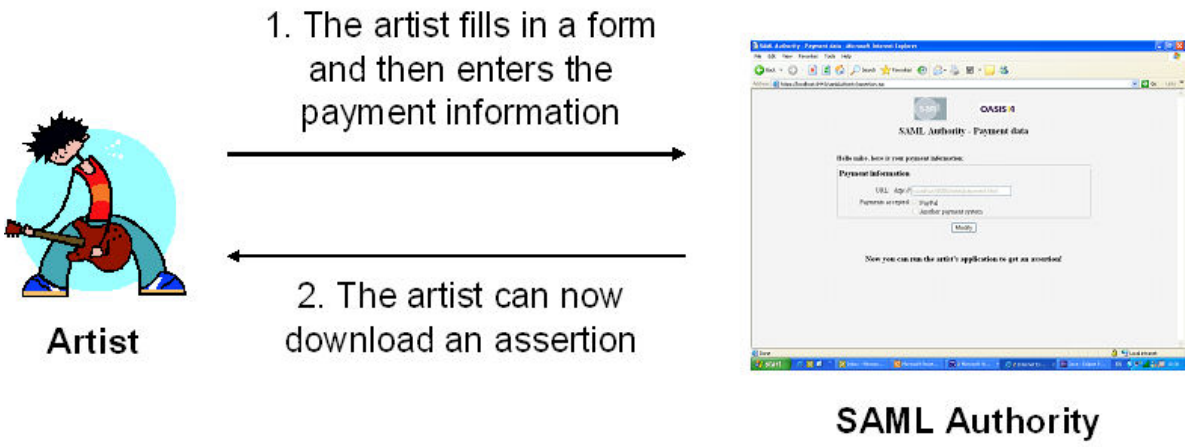

Figure 4.3: Step 1: Creation of an account on the SAML Authority

The second step for the artist is to request a SAML assertion from the SAML Authority and to store it in the ID3 tag of their MP3 file. A special ID3 tag editor for the artist was developed for the proposed application, and, by clicking on a 'Get assertion' button, the SAML assertion is requested and then stored in the ID3 tag. More details with regard to the above will be given in the next section.

In this prototype the SAML Authority was implemented using an Apache Tomcat 5.5 server [8]. The Secure Socket Layer (SSL) was added to this server to guarantee the security of the communication between the artist and the SAML Authority. In order to store the artists' data, a MySQL 4.1 database server [9] was used. Finally, to handle and create SAML assertions, a java open source SAML library called OpenSAML 1.0 was used [10].

In order for an artist to obtain a SAML assertion, they must send a request to the SAML Authority which contains their public key as well as their login and password information for their SAML account. When the SAML Authority receives a request from the artist for an assertion, the servlet first verifies from within the database that the requesting artist is registered on the SAML Authority by verifying the given login and password. If this is verified, the servlet will search the database to establish whether

the artist's public key has already been stored. If it has not, the public key will be stored at this point. The servlet then builds the SAML assertion with the information about the user which is stored in the database (public key, URL, payment systems). Finally, it sends the assertion back to the user.

The different components of an artist's request for a SAML assertion are illustrated below:

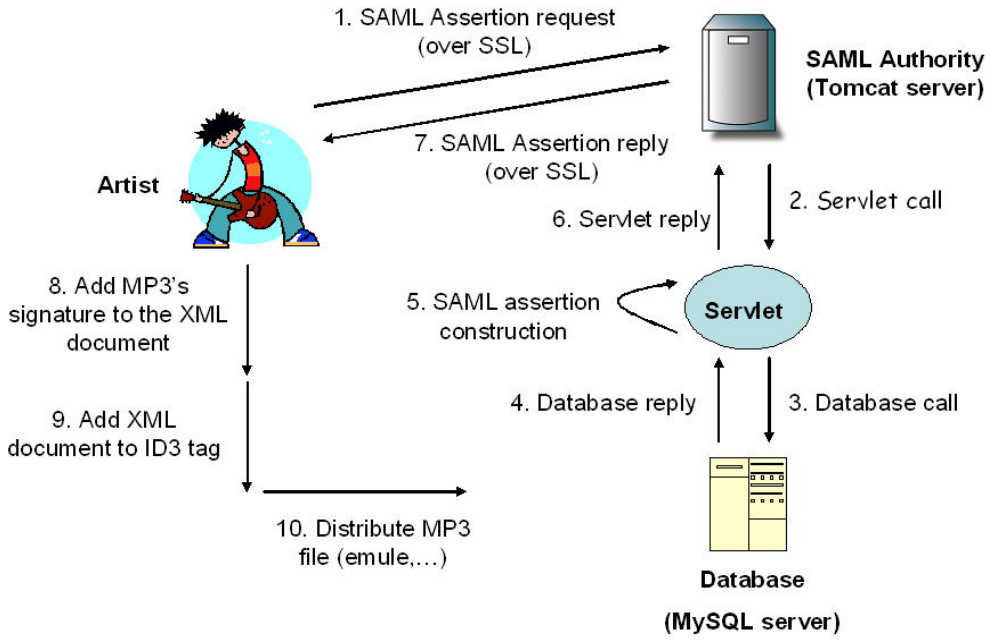

Figure 4.4: Step 2: Artist's request for a SAML assertion

The web pages used to create an account were dynamically generated using Java Server Pages (JSP) and HyperText Markup Language (HTML). Only one servlet was created on the SAML Authority: this is the servlet required to reply to an assertion request from an artist. So, the 'web.xml' file was modified to add this servlet. For the presentation of the pages, Cascasding Style Sheets (CSS) was employed. The requests to the MySQL database were made by a Java DataBase Connectivity (JDBC) - Open DataBase Connectivity (ODBC) Bridge. Basically, an ODBC driver for MySQL was installed and then the Application Program Interface (API) JDBC was used in order

to translate high-level operations in Java into ODBC operations. Owing to the ODBC driver for MySQL, ODBC could interact with MySQL and return the results of the queries to JDBC.

Furthermore, in order to create the SAML Authority's private key and certificate, the 'keytool' command was used. This command creates a new file named ".keystore" in the home directory of the user under which the command is run. For instance, for the SAML Authority developed for this dissertation, the file was stored in 'C:\Documents and Settings\Administrator\.keystore'. The Java Development Kit (JDK) defines this file to contain the different keys and certificates of the system. This file is protected by a password and different aliases make it possible to add several different certificates or keys. Also, for the purpose of this prototype, a self-signed certificate was generated for the SAML Authority using the RSA algorithm. The SAML Authority certificate was stored under the alias 'tomcat' in the keystore. The command is as follows: 'keytool -genkey -alias tomcat -keyalg RSA'.

Finally, the SAML Authority's URL is 'https://localhost:8443/samlAuthority/index.html', and this is the address the artists have to connect to in order to create an account.

4.5 Artist's ID3v2 tag editor

To simplify the process of acquiring an assertion and storing that assertion within the ID3 tag of an MP3 file at the same time, an ID3v2 tag editor was developed in Java for the artists.

The ID3v2 tag editor that was developed allows the artist to enter their name, the title of the song and the name of the album. They can then save the changes by clicking on the 'Save' button. By clicking on the 'Get Assertion' button, the editor first creates a secure connection to the SAML Authority using SSL, sends the artist's certificate with their login and password and waits for the SAML Authority's reply. When the assertion is received by the ID3v2 editor, the editor starts building the XML document using Document Object Model (DOM). This is the XML document that will contain the payment data and that will be stored in the ID3 tag. Firstly, a root

element called 'PaymentData' is created and the assertion is then nested within this element. Furthermore, the signature of the MP3 file (excluding the ID3 tag) is created and is contained within the 'PaymentData' element. Finally, the entire XML document is stored in the ID3 tag of the MP3 file and, since the MP3 now contains all of the information necessary to donate money to the artist, it is ready to be distributed on the Internet.

The illustration below gives a screenshot of this editor:

Figure 4.5: A screenshot of the artist's ID3v2 tag editor

In order to edit the ID3v2 tag of an MP3 file, a Java open source library called JID3 was used. It allowed the title of the song, the name of the artist, the name of the album, but also the payment data to be modified [11].Another library was used to generate the artist's XML signature of the MP3 file. This was called Apache XML Security 1.2.1 [12].

With regard to the artist's public and private keys, the 'keytool' command was used as described in the previous section. This time, the alias was 'artist' and not 'tomcat'!

4.6 MP3 Player

The MP3 player is the principal application for the microsponsorship model. Indeed, it is from their MP3 player that the users can donate money directly to the artist. For the purpose of this dissertation, the MP3 player was not developed from scratch. Instead, an existing MP3 player called jlGui was extended. jlGui is a Java open source implementation of an MP3 player, and it is said to be 'a Java WinAmp clone' [13].

The reason for choosing this MP3 player was threefold. Firstly, the MP3 player needed to be extensible, and so the source code of the player was required. Moreover, the player had to support ID3v2 in order to be able to read the payment data. Finally, the source code needed to be free to reuse. The open source project jlGui satisfied all of these requirements.

The illustration below shows a screenshot of the MP3 player prior to modification:

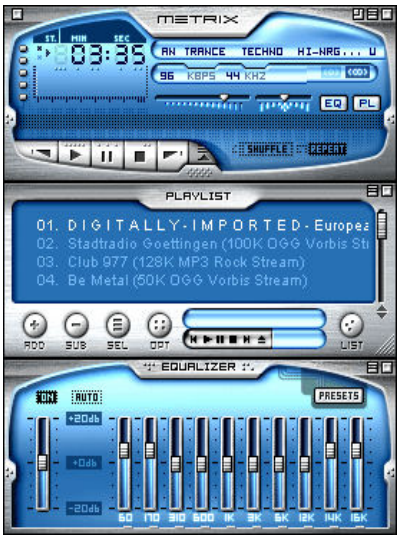

Figure 4.6: A screenshot of the jlGui player prior to modification

The first modification that was made to the MP3 player was to display a 'Donate' button. The jlGui player actually uses a compressed file called 'metrix.wsz' that contains images in .bmp format to draw the Graphical User Interface (GUI). So, the

first step was to add the 'Donate' button which was drawn in 'payment.bmp' to this file.

The 'payment.bmp' image is displayed below:

Figure 4.7: The 'payment.bmp' image

There are actually three buttons drawn in the 'payment.bmp' file. Indeed, they correspond in order to when the MP3 file contains no payment data or corrupt payment data (grey), when it contains some valid payment data and the button is not clicked (yellow), when it contains some valid payment data and the button is clicked (blue). It is more efficient in terms of efficiency to have all three buttons in the same file because, in this way, the file is loaded only once. Then, a virtual window is created of the same dimensions as the Donate button. Changing the location of the virtual window will determine whether the grey, yellow or blue button is displayed. This allows the colour of the button to change quickly depending on the circumstances.

The next step was to add a listener to this button so that an action would take place when the button was clicked by the user, but only when some payment data were contained in the MP3 that was playing. A class called 'PaymentActiveComponent.java' was created to handle this. The details of the implementation can be found on a CD that was submitted with this report.

The final step was to add coding that would enable the MP3 player to read the ID3v2 tag and to check for valid payment data. No particular library was used for this purpose and the 'WPAY' frame of the ID3 tag was read as a byte array. Then, the XML document was created by DOM from this byte array. The SAML assertion and the artist's XML signature of the MP3 file were then extracted. The OpenSAML 1.0 library [10] and the Apache XML Security 1.2.1 library [12] were used to verify and validate the SAML assertion and the XML signature respectively.

The next section will be a brief explanation of the artist's web page that was developed for the proposed application.

4.7 Artist's web page

The artist's web page is very important because it is from here that the user donates money to the artist. In the microsponsorship model, after clicking on the Donate button of the MP3 player, the user arrives to the artist's web page. On the artist's web page, a finance bar is displayed to show the amount of money raised to date, and the amount of money that still needs to be raised before releasing an album. This would encourage people to donate as they could see the bar grow day after day. In the prototype, only the online payment system PayPal was implemented. This system can be used by clicking on the PayPal button on the web page.

The mechanism involved in the use of the PayPal system is illustrated below:

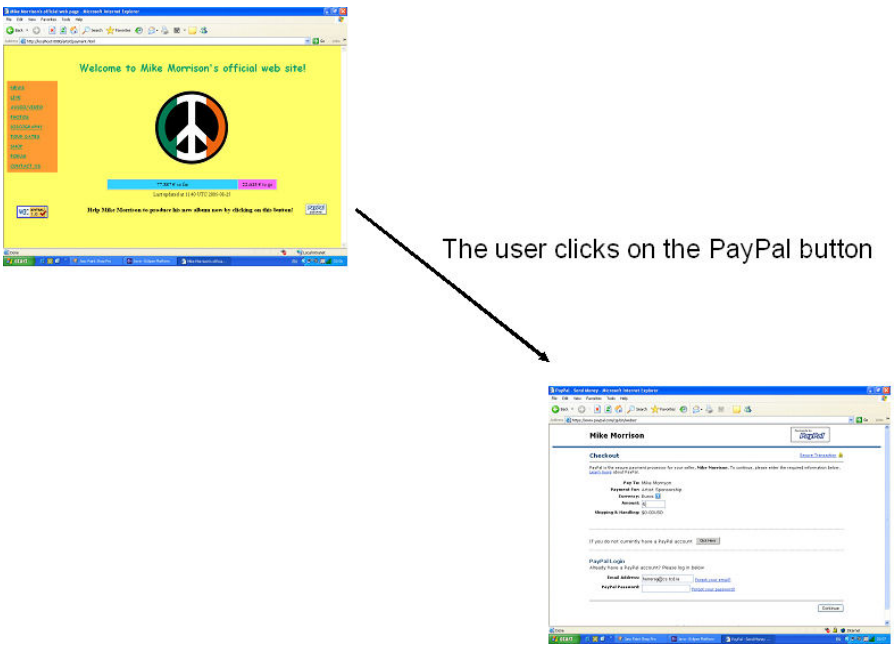

Figure 4.8: The payment mechanism for PayPal

After clicking on the PayPal button on the artist's web page, the user arrives to the PayPal web page where they are required to enter the amount of money they wish to donate to the artist. The user must then log into their PayPal account and confirm the transaction.

Chapter 5

Evaluation

The evaluation of the project will serve as an analysis of the entire application to ensure that the objectives given have been fulfilled. Moreover, it will discuss the security threats that are an integral part of the system.

5.1 Benefits

As stated in the introduction, the objectives of this dissertation were to design a novel system for music distribution that would enable users to pay only for the music that they really wished to purchase and to give users the option to support their favourite artists. The aim was to produce a user friendly working prototype, namely an MP3 player, that would allow the user, at the click of a button, to donate money directly to the artist they are listening to.

The application that was developed followed the objectives given. However, the real test was to see how it worked in a real environment, namely, while playing MP3s that contained payment data to allow donation of money to the artist and with other MP3s containing no payment data at all or invalid payment data. Obviously, the MP3 player has to be capable of playing both types of MP3, and the Donate button has to be capable of activating or not depending on whether payment data are contained in the MP3. The tests that needed to be implemented were all very simple and consisted of playing songs that either contained or did not contain payment data.

Below are two illustrations of the MP3 player respectively in the case that the MP3 being played contains no payment data or some invalid payment data and in the case that the MP3 being played contains valid payment data:

Figure 5.1: The 'Donate' button displayed in grey because inactive

Figure 5.2: The 'Donate' button displayed in yellow because active

As expected, in Figure 5.1, the 'Donate' button is inactive and its colour is grey. This means that it is not possible to pay this artist because either the MP3 does not contain any payment data or the payment data are invalid. This would be the case if either the assertion's signature was not verified or the artist's signature of the MP3 file was not verified. When the button is inactive, clicking on it would have no effect.

Moreover, in Figure 5.2, the 'Donate' button is active and its colour is yellow. This means that it is possible to donate money to this artist by clicking on this button. In this case, a new window would open displaying this artist's web page from where the user could pay them.

The results showed that the application worked successfully. Thus, money could potentially be donated to the artist by some form of online payment, Paypal in this instance. Likewise, the donate button remained suitably inactive where no payment data was contained in the MP3 and thus it was not possible to donate money to the artist using the microsponsorship model. This meant that there was no confusion for the user as to whether or not they could actually donate money to the artist. Thus the system was very user friendly.

So, the main goal of this dissertation was reached, that is, to deliver a working prototype that fulfils its objectives and that could potentially replace the existing business model for music distribution. The model developed in this dissertation could have used two different types of architecture as seen in Chapter **??**. The solution chosen was the distributed model with the use of the ID3 tag to store the payment data. This model offers huge benefits as it is scalable and no latency would occur should a large number of individuals use the microsponsorship system at the same time. Also, the prototype developed uses the secure online payment system PayPal, which is one of the most well-known systems to make transactions on the Internet. In addition to this, the application was developed in such a way that the addition of other payment systems in the form of future extensions would be possible.

The system of music distribution developed for this dissertation could, potentially, be very useful in the future for unknown artists who want to use the network as a way to become famous and to have the possibility of releasing a CD without the support of a music company.

5.2 Security threats

Unfortunately, there are still some technical and ethical issues that have not been resolved in the proposed model. Firstly, one could wonder how the system would handle the theft of an MP3 by a hacker who would claim to own it by adding their payment data in the ID3 tag.

This threat is illustrated in the flow diagram below:

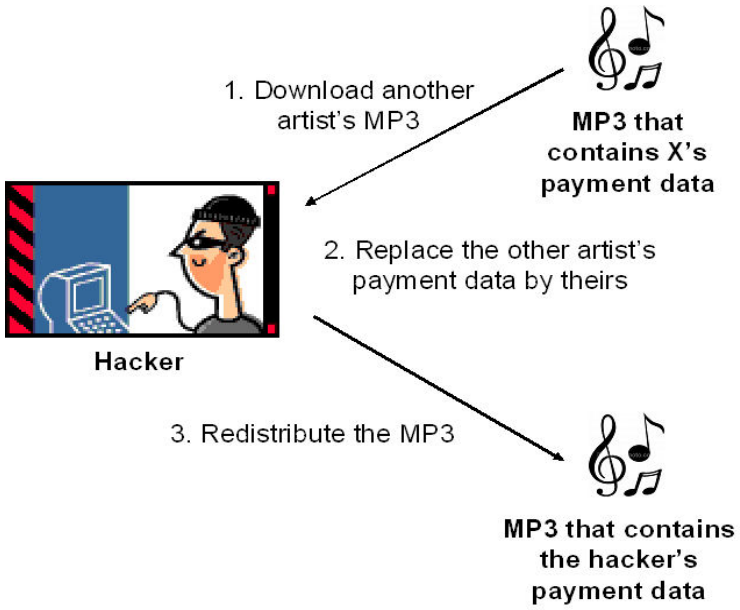

Figure 5.3: Theft of an artist's MP3 by a hacker

The payment data consists of two parts. Firstly, it consists of a valid assertion that the artist must download from the SAML Authority and secondly, the artist's signature of the MP3. When playing an MP3, the MP3 player checks the validity of the payment data contained in the ID3 tag by validating the assertion initially and then by verifying the artist's signature of the MP3. Indeed, the MP3 player is configured with the SAML Authority's certificate so that an assertion can be checked with the SAML Authority's public key. If the SAML assertion is valid, it means that the data contained in the assertion such as the artist's public key, the URL of the artist's web

page and the payment methods accepted by the artist are legitimate. The final step is to verify the signature of the MP3 with the artist's public key which is obtained from within the assertion. Therefore, if a hacker wanted their payment data to be verified they would have to have an account on the SAML Authority in order to obtain a valid assertion. The hacker would have to sign the MP3 with their private key, and the public key contained in the assertion would be used to check the signature. The signature is actually the fingerprint of the artist, that is to say, its use is just to prove that an owner of an assertion stored their payment data in the ID3 tag. The binding of an assertion to an MP3 follows a lock and key principle. The signature is the catalyst required for the binding to take place. Thus, if a hacker stole another artist's MP3, they would have to replace the existing assertion with their own one and then sign their fraud. For this reason, it would be easy to identify and sue such a hacker because their information would be known. To make the system even more secure, certain complementary papers could be demanded on the creation of an account to confirm its validity. To put it in a nutshell, no technical solution was implemented in the given prototype, but legal action would be easy to take and would dissuade any hackers from steeling the payment data of an artist. This issue is open to future work.

There is another issue concerning the payment of the artist with the microsponsorship model. As stated earlier, the ID3 tag is easy to modify, and one could wonder what would happen if someone removed the payment data from the MP3, and then redistributed it on the Internet. If this were to happen, some of the MP3s released by an artist would not permit the donation of money to that particular artist. Furthermore, in the case of an individual removing the payment data from the MP3, there is no way to detect the person that is responsible for it. However, this kind of practice would be marginal or even inexistent as there is no motive for anyone to do it except, perhaps, for the purpose of ruining one or a number of a particular artist's MP3s that are available on the Internet. In any case, as the number of valid MP3s would be far greater than the number of MP3s whose payment data has been removed, it would not alter the effectiveness of the model.

Finally, an ethical issue also exists. That is to say, that if such a system was to be put into action, how could one ensure that people would actually donate money

to the artists they like? Indeed, if Internet users could have unlimited music for free, what would urge them to pay for it? For this reason it would be logical to implement the sponsorship based model only for unknown artists initially as this would not affect any pre-existing business model for them. If the microsponsorship model proved effective on this population, the concept could then be extended to any artists and could become the de facto standard in music distribution. One means of encouraging users to donate money to their favourite artists would be to display the donate button in different colours or to make it flash in cases where the user has listened to the same artist previously but not yet made a donation. A similar colour change on the donate button could also occur if the user has listened to a particular song more than ten times for instance or if they have previously donated money to the artist they are listening to.

Chapter 6

Conclusion

Nowadays, the music industry on the Internet is such that music companies benefit mostly from users downloading MP3s. Needless to say, music companies are content with the current system and offer great resistance to the development of new technology that would potentially take from their earnings.

The previous sections have shown that the present environment for downloading music from the Internet is very restrictive. Individuals are not fully informed as to the music that they are downloading prior to paying for it as they may only be able to sample a short preview of the song or, in some cases, they will be able to hear nothing at all. Essentially, users are blind as to what exactly they are paying to download. Further to this, if they don't like the music they have downloaded, a refund is not possible. Nor is it possible for the user to support their favourite artists directly. Rather, they support the music companies with the artist only receiving a small percentage of the profits. Thus the system is restrictive for both users and artists alike.

The purpose of this dissertation was to design a novel system for music distribution that enables users to pay only for the music that they really wish to purchase and to give users the option to support their favourite artists.

The microsponsorships concept discussed in this dissertation would provide a fairer system of downloading MP3s from the Internet for both the user and the artists. Un-

der this system, users would have the opportunity to download an MP3 from the Internet and to listen to it. If they like the music, they could then support the artist directly by using the 'donate' button on the MP3 player. The overall effect such a system would have on the music industry would be to promote music sales of all artists and not just those that have been signed by record companies. This would result in fairer competition among artists and perhaps bring about a fall in the price of music.

Implementation of the microsponsorships model would require each artist to have a web page from where the user could pay that particular artist. Because money is donated by the user directly to the artist, this removes the need for the middle man, in this case, the music companies. Obviously, for this reason, the proposed microsponsorships model would encounter strong opposition from music companies should it be put into action. It is important to bear in mind, however, that the music industry has historically reacted strongly against new technologies that later turned out to be beneficial to them. Examples of this would include the music cassette and cable TV. There is therefore a need to explore business models for network distribution of digital music.

Nevertheless, this concept demonstrates another feasible system for purchasing music from the Internet. It is easy to use and promotes fairer competition among artists, allowing even unknown artists to have a platform from which to promote their music. This would potentially increase the variety of music available on the Internet. Such a system would also allow record companies to search the net for popular artists, artists who have already received attention or proved themselves, so to speak, in the music world.

The current evidence would suggest that the microsponsorship concept proposed in this dissertation is the first of its kind. That is not to say that the current music business model does not need to or cannot change. Moreover, the proposed system would promote fairer competition among artists and would increase music variety. Because of the restrictive nature of the current business model, further research into the feasibility of exciting new models such as the microsponsorship concept is merited in order to change the face of the music industry on the Internet.

Appendix A

Payment data in XML Document


```
  <?xml version="1.0" encoding="UTF-8" ?>
- <PaymentData>
- <Assertion xmlns="urn:oasis:names:tc:SAML:1.0:assertion"
xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol"
AssertionID="b85916a405e6bea9abcec5440897f6ab"
IssueInstant="2005-08-31T15:00:43.584Z"
Issuer="https://localhost:8443/samlAuthority/index.html" MajorVersion="1"
MinorVersion="1">
- <AttributeStatement>
- <Subject>
  <NameIdentifier>Mike Morrison</NameIdentifier>
- <SubjectConfirmation>
  <ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:holder-of-key
</ConfirmationMethod>
- <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:KeyName>Mike MorrisonKey</ds:KeyName>
- <ds:KeyValue>
- <ds:RSAKeyValue>
  <ds:Modulus>mAcMi9nMzekp+bfggFQAZNyCIeMg+R+fh4ALHMCWBWGxk1BePDWAhTnAI5xstC
WChrhpqmnZcZhh98JJPEaWm7mvtsChMEPhCr5hQf9DV1aWZeKR0Lmqs2yaXmPXxDYvwjOsxMl5h/
```

```
FXCaNzZ6esVhW4 vXYcUjsh/GMl3M5El1U=</ds:Modulus>
  <ds:Exponent>AQAB</ds:Exponent>
  </ds:RSAKeyValue>
  </ds:KeyValue>
  </ds:KeyInfo>
  </SubjectConfirmation>
  </Subject>
- <Attribute AttributeName="PaymentSystems"
AttributeNamespace="urn:tcd.ie:attributeNamespace:uri">
  <AttributeValue>PayPal</AttributeValue>
  </Attribute>
- <Attribute AttributeName="PaymentUrl"
AttributeNamespace="urn:tcd.ie:attributeNamespace:uri">
  <AttributeValue>localhost:8080/artist/payment.html</AttributeValue>
  </Attribute>
  </AttributeStatement>
- <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
- <ds:SignedInfo>
  <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"
 />
  <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
- <ds:Reference URI="#b85916a405e6bea9abcec5440897f6ab">
- <ds:Transforms>
  <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
signature" />
- <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
  <ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#"
PrefixList="#default code ds kind rw saml samlp typens" />
  </ds:Transform>
  </ds:Transforms>
  <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
  <ds:DigestValue>94/+y27e1EmzZxw1oB/35MukWAo=</ds:DigestValue>
  </ds:Reference>
```

```
  </ds:SignedInfo>
  <ds:SignatureValue>YGs7YP8J9/9Up9aoD9MiiCD1T8yU/BOC6Yi1xNI75NMrVKTCMD8SI/V7iL
z1cUQdm3ZM4Ja1VQdPgvtFv2UljqdUlVQkyN6o8+WmZDU2SMmJZYtIzoY0axHX6p71LZYPo9JCTmOWF
snP7TxHfLziVRSt yGHePlTv/3P7JKtDiC0=</ds:SignatureValue>
  </ds:Signature>
  </Assertion>
- <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
- <ds:SignedInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"
 xmlns:ds="http://www.w3.org/2000/09/xmldsig#" />
  <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#" />
- <ds:Reference URI="C:\temp\fileToSign.tmp"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#" />
  <ds:DigestValue
xmlns:ds="http://www.w3.org/2000/09/xmldsig#">oeAS12ZruWL39n/E65/b/ovddIY=
</ds:DigestValue>
  </ds:Reference>
  </ds:SignedInfo>
  <ds:SignatureValue
xmlns:ds="http://www.w3.org/2000/09/xmldsig#">ZxO8jDlNixnUR8ttjiqFJmesgR+Uu5S4ygE
HBYRG/NUDNJLHdwYSyNzwIQNAB9dAxTi+0nWNjt2U8s/0CxeMLSZe5ATPdN4jK2cRBzUxH/gy48B5In9s
KWb8+7IXL6vegQvxYEKxdmcfVFQpN8aiEg +bJxDC+kUzJXviN1tbQ=</ds:SignatureValue>
  </ds:Signature>
  </PaymentData>
```

Bibliography

[1] Lawrence Lessig, *Free Culture*. Penguin Books, 2005.

[2] iTunes, *http://www.apple.com/itunes/*.

[3] Napster, *http://www.napster.com/*.

[4] The online payment system PayPal, *https://www.paypal.com/*.

[5] X.509 certificate, *http://java.sun.com/j2se/1.3/docs/guide/security/cert3.html*.

[6] OASIS Security Service SAML, *http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security*.

[7] ID3 tag version 2.4.0 Documentation, *http://www.id3.org/id3v2.4.0-frames.txt*.

[8] Apache Tomcat 5.5, *http://jakarta.apache.org/tomcat/*.

[9] MySQL 4.1 Server, *http://www.mysql.com/products/database/mysql/*.

[10] OpenSAML 1.0, *http://www.opensaml.org/*.

[11] Java ID3 tag library JID3, *http://jid3.blinkenlights.org/*.

[12] Apache XML Security 1.2.1, *http://xml.apache.org/security/*.

[13] jlGui Player from Javazoom project, *http://www.javazoom.net/jlgui/jlgui.html*.