

# The Audiotarium

A Concept for a Public Domain Sound Object Repository

**Author** Tuomo Tarkiainen

**Graduate Degree** MA New Media

**Advisors** Antti Ikonen, Nuno Correia, Andrea Botero, Teijo Pellinen

**Month / Year** October 2008

**Number of Pages** 76

**Media Lab Helsinki**

# Table of Contents

<b>Abstract</b> .....	<b>4</b>
<b>Acknowledgments</b> .....	<b>5</b>
<b>1. Introduction</b> .....	<b>7</b>
1.1 Project Background .....	7
1.2 Design Objectives.....	8
1.2.1 Public Domain to Liberate Sound Objects .....	8
1.2.2 The Audiotarium Platform to Lower the Threshold of the Play .....	10
1.2.3 The Audiotarium API to Expand the Element of the Play .....	10
1.3 An Overview of the Work .....	11
<b>2. Design Principles Explained</b> .....	<b>14</b>
2.1 The Element of the Play .....	14
2.2 Information Wants to be Free .....	17
2.3 Public Domain and the Notion of Credits.....	19
2.4 Creative Commons and Free Culture.....	20
2.5 Sound as a Physical Phenomenon .....	21
2.6 Contextually Relevant Web Services .....	24
2.6.1 YouTube.....	24
2.6.2 Flickr.....	26
2.6.3 The Freesound Project.....	26
2.7 Affordances, Perceived Affordances and Cultural Constraints .....	28
2.8 About Accessibility .....	29
<b>3. The Audiotarium Concept</b> .....	<b>31</b>
3.1 The Audiotarium Ideology and Rules .....	31
3.1.1 The Audiotarium Ideology.....	31
3.1.2 The Audiotarium Rules .....	32
3.2 Main Parts of the Concept .....	34
3.2.1 The Public Domain Sound Repository .....	34
3.2.2 The API and Sandbox Applications .....	35
3.2.3 The Web Site and The Community .....	37
3.3 Good Practises .....	38
3.3.1 Credits.....	38

3.3.2 Meta Data Conventions .....	38
3.3.3 Sandbox Application List .....	39
3.3.4 Statement About Accessibility .....	39
3.4 The Technical Structure .....	39
3.4.1 The Core System Structure .....	39
3.4.2 The API Structure.....	40
<b>4. The Prototype.....</b>	<b>42</b>
4.1 Limitations of the Prototype .....	42
4.2 About Technologies .....	43
4.2.1 MySQL .....	43
4.2.2 PHP .....	43
4.2.3 WordPress .....	44
4.2.4 Flash & ActionScript .....	44
4.2.5 The OpenSource Map UI Component .....	45
4.3 Pages.....	46
4.3.1 General pages .....	46
4.3.1.1 <i>The Main Page</i> .....	46
4.3.1.2 <i>The Info</i> .....	46
4.3.2 User related pages.....	47
4.3.2.1 <i>The List of Users</i> .....	47
4.3.2.2 <i>My Auditorium</i> .....	47
4.3.2.3 <i>The User Info</i> .....	48
4.3.3 Sounds.....	48
4.3.3.1 <i>Search Sounds</i> .....	48
4.3.3.2 <i>The Sound Repository</i> .....	49
4.3.3.3 <i>Add a Sound</i> .....	50
4.3.4 The API .....	50
4.3.4.1 <i>The API Explained</i> .....	50
4.3.4.2 <i>API URLs and Functions</i> .....	50
4.3.4.3 <i>Get an API Key</i> .....	51
4.3.5 Sandbox Applications.....	51
4.3.5.1 <i>The List of Sandbox Applications</i> .....	51
4.3.5.2 <i>The Auditorium Rotator</i> .....	52
4.4 About the Visual Identity and the Interaction Design .....	52
4.5 The Core System and the UI System .....	53
4.6 Hosting and Bandwidth Issues .....	54

4.7 The Scalability .....	54
<b>5. Results.....</b>	<b>55</b>
<b>6. Conclusion.....</b>	<b>57</b>
<b>Appendix.....</b>	<b>62</b>
A. The Database Structure of The Audiotarium Core Functionality.....	62
B. The Code of the AudiotariumInterfaceAC3 Class .....	63
C. Examples of the Audiotarium API XML Results.....	67
D. Textual Version of the Creative Commons Public Domain Dedication....	70
E. The Complete List of the Audiotarium API Functions.....	70
<b>List of Acronyms.....</b>	<b>73</b>
<b>Bibliography.....</b>	<b>74</b>

# Abstract

In creative activity, the tools of creative work form the matrix of possibilities. If these tools are not available or if they are not supporting the activity well enough, the threshold of starting to play and to experiment may be too high for most people. The purpose of this work is to understand and develop tools that would enhance everyone's possibilities to play with sounds in the context of the web. Practically the work documents a design and development exploration made by concentrating on three main themes: creativity, the Internet and sound. Creativity is condensed in the idea of "the play", the web refers to the Internet and a sound object is considered to consist of a sound file and of meta data related to it.

Based on a series of design objectives - Public Domain to Liberate Sound Objects, The Audiotarium Platform to Lower the Threshold of the Play and The Audiotarium API to Expand the Element of the Play - the work proposes The Audiotarium concept, a simple, free and open platform for creative sound work on the web. The Audiotarium service offers a possibility to contribute and get access to sounds. The concept is validated through iterative prototyping and preliminary user test of a functional beta version of the service that functions as proof of concept. The prototype uses a combination of Open Source and free tools (WordPress, MySQL and PHP) that exemplifies the main functionality of the concept and serves as a playground for further development.

In its basic description the Audiotarium service consists of a public domain repository of sound objects expanded by a standard way to access the repository programmatically (API). Building on both the public domain ideology (the idea of completely free cultural entities) behind the concept and the Audiotarium API (Application Programming Interface that makes it possible for other applications to automatically access the functionality of the Audiotarium) the work proposes a grey area for users and developers to expand the concept and continue the play with sounds.

# Acknowledgments

Everyone I have ever met or heard about has affected me somehow. In the context of this work I want to mention some people who have especially made this work possible. They all have been very understanding and helpful.

So I thank all the people on the following list.

My family for endless patience.

My teacher, instructor, and source of inspiration, Antti Ikonen.

My instructor, friend, and source of inspiration, Nuno Correia.

My teacher, friend, instructor and colleague, Andrea Botero.

My other mentors :

Shinji Kanki

Rasmus Vuori

Pipsa Asiala

Heidi Tikka

Kari-Hans Kommonen

Sanna Marttila

Teijo Pellinen

Ville Tikkanen

All the teachers and the staff of the Media Lab Helsinki.

All my fellow students in Media Lab Helsinki.

My employers during the final thesis process who made it financially possible for me to do this work.

TAMK University of Applied Sciences, School of Art and Media

The Arki research group, Media Lab Helsinki

Learning Environments research group, Media Lab Helsinki

Fjord Oy

Zipipop Ltd

Test users for allocating some of their time to test the prototype and to give invaluable feedback.

Kristine Visanen for spell check.

Last but not least some friends of mine who have been supportive and very inspiring :

Atle Larsen  
Kalle Mäntsälä  
Sami Stranden  
Pekka Mähönen  
Mikko Sääntti  
Tommi Sääntti  
Juha Tanskanen  
Miska Knapek  
Anne Naukkarinen  
Joanna Saad-Sulonen

# 1. Introduction

## 1.1 Project Background

To me, the writer of the work, the project is personally very important for many reasons. I take the liberty to use slightly informal language to introduce the background of the project.

I started playing with sounds when I was 7 years old. My parents bought me a small Casio home organ. In the beginning I was really puzzled by the instrument. I really didn't get it. Why the weird pattern of white and black keys? Why all kinds of knobs and slides? As I was randomly pressing keys and tweaking knobs, what came out of the small speakers didn't sound very organised. To be honest, it was horrible audio chaos but then again - I loved it and my parents probably hated it. The fact that I was able to produce sounds with an instrument was absolutely fascinating. I was profoundly moved by sound.

Time passed. I continued playing with sounds. My experiments made me learn a lot of things related to sound. Since I was spending most of my time doing something with the computer anyway, a couple of years after high school I ended up studying interaction design.

I'm still interested in all things which are related to sound. This master's thesis work is my personal voyage into the world of digital sound objects and the web. The work is also a tribute to all the people who have helped me dig deeper into this wonderful world of sonic things. Sound has become professionally very important to me. I work more and more with things related to sound on the web and from the perspective of a web designer, I would say that sound is becoming a more and more important part of the Internet.

So both my professional and personal life have come together in this work. Because of this duality, there is some ideology involved. I wanted to keep the concept as open and free as possible. For a long time, most of the tools I used were free tools.

Another almost ideological element which is present in everything that I do is a thing I call *the play*. Pierre Schaeffer called this element *jeu* (Schaeffer, as cited in Cambridge Encyclopedia 1994). The verb 'jeune' (play) has a double meaning.

- a. to enjoy oneself by interacting with one's surroundings**
- b. to operate an instrument**

The definition a. of the verb play represents very well the ideology of *the play* that I want to follow. In this work, I am aiming for all kinds of free, open and simple solutions to enhance the possibility to play with sound objects in the context of the web. In this work, when I speak about the play I still refer to the definition a. since it serves as an umbrella definition for all of the rest of the definitions of the verb play. I have to add that perhaps the notion of enjoyment in the definition a. is a little bit unrealistic. It is not always the case that all the action that can be called the play is enjoyable. This is a very important notion since on an abstract level the whole work is about this. I think all of the play **should** be enjoyable.

## **1.2 Design Objectives**

This part introduces the three main design objectives of this project. These objectives have been used as the guiding rules for decisions during the design process of the Audiotarium concept. The implementation of the prototype was done on the basis of the concept design so the prototype follows the same design objectives.

### **1.2.1 Public Domain to Liberate Sound Objects**

The public domain is not a licence.

*Public domain refers to the total absence of copyright protection for a creative work. [...] Public domain works (i.e., works in the public domain) are considered to be a part of the public's cultural heritage, and thus*

*anybody is entitled to make use of them for any purpose, including copying, modifying and even selling, including with a restrictive license such as a EULA (end user license agreement) (Public Domain Definition 2004)*

A work of art or a computer program or almost anything that is done by a person can be stated to belong to the public domain.

In this work, the public domain is considered as a way to liberate sound objects. The process of liberating sound objects is required to enhance people's possibilities to use sound objects to be creative. This method of announcing something to be in the public domain is close to the ideas of Open Source (Coar 2006) and Creative Commons (Creative Commons 2008) licensing. Open Source software is usually free software, but not necessarily. Creative Commons is a charitable corporation which has specialised in providing free licensing for certain purposes. Both cases differ from completely free software.

*"Nearly all open source software is free software; the two terms describe almost the same category of software. But they stand for views based on fundamentally different values. Open source is a development methodology; free software is a social movement." (Stallman 2007)*

The above quote by software freedom activist and hacker Richard Stallman describes the main difference between free software and Open Source software.

In this work, this idea of freedom is taken into the context of sound not by breaking existing regulations, licensing systems or copyright laws, but by creating a parallel system of completely free and open sound objects. The aim is to encourage and support people to liberate sounds and not licence them. Works declared to be in the public domain are free. In the context of the Audiotarium concept, all sound objects need to be free which makes the use of those sound objects easy. This is needed to keep the threshold of the play as low as possible.

### **1.2.2 The Audiotarium Platform to Lower the Threshold of the Play**

In creative activity tools of the creative work form the matrix of possibilities. If these tools are not available, or if they are not supporting the activity well enough, the threshold of starting to play and to experiment may be too high for most people. The Audiotarium concept relies on open-minded people who want to participate in a public effort of creating a public domain sound repository. The purpose is to enhance everyone's possibilities to play with sounds in the context of the web. The Audiotarium aims to provide a platform which would lower the threshold of starting the play. In this effort, the focus is on liberating sounds from the chains of the complex system of rights and ownership and on providing an inspiring set of tools for working with sounds in the context of the web. Digital technology is used to provide people a set of tools to participate in the process of liberating sound objects and to play with sound.

### **1.2.3 The Audiotarium API to Expand the Element of the Play**

API is an acronym of Application Programming Interface. Practically, an API is a set of functions. An Application Programming Interface makes it possible for an application to automatically access functionality of another application or system.

An API gives developers a possibility to make their applications communicate with someone else's applications. Usually, this provides many innovative possibilities for developers to make so-called mashup (Storani 2008) applications. Mashup applications are applications which are implemented by combining existing functionality from several existing applications. Another way to use an API is simply to make it possible for people to build applications on a certain platform. For example, the famous networking service Facebook (<http://www.facebook.com>). functions this way. Facebook provides the platform and developers use that platform to make their applications appear as parts of Facebook.

The Audiotarium API provides both options of API usage to people who are interested in exploring and developing the world of sound in the context of the web. All the possibilities of an API cannot be predicted. Within the

technical scope of the API, the imagination of a developer is the only limiting factor. The API can also be developed according to needs of developers.

The development of the Audiotarium happens in collaboration with the Audiotarium community. Everyone who registers to be a member of the Audiotarium shall also become a member of the Audiotarium community, and the Audiotarium Group. Applications done using the Audiotarium API are referred to as Audiotarium Sandbox Applications. In the beginning, they are entities of their own but there is also a possibility to develop the Audiotarium web site to be a facilitator for activity which brings users and developers closer to each other. The technical structure of the Audiotarium makes it easy to build the experimental section on the web site. As an example, this kind of section could have a web radio which plays things done in Audiotarium Sandbox Applications. There could also be possibilities to expand the API to support collaborative playing and experimenting. In this kind of situation, the Audiotarium web site could offer an output channel like some kind of virtual club where people could observe art done by means of the Audiotarium. Also, live online spectacles may be possible.

The Audiotarium API opens doors of the Audiotarium sound object repository to expand the element of *the play*. Together with the public domain thinking, the API lowers the threshold of playing with sound on the web. The automated use of a big quantity of licensed sound objects would be a very complex and time consuming process. In some cases, it would be practically impossible. But in the case of the Audiotarium, all of the sounds are public domain sounds so there are no problems with crediting because it is not obligatory. However, crediting is considered to be an act of goodwill and good manners and the Audiotarium API supports it by providing the user name of the user who has added the sound as meta data of the sound object.

### **1.3 An Overview of the Work**

The work has two parts, a written part and a prototype. The whole work is called the Audiotarium concept. The short version is simply called the Audiotarium. The prototype of the concept is a web service and is online at: **<http://www.audiotarium.net/proto>**. Only test users are able to log in and use the main functionality of the Audiotarium prototype. The prototype has been built to be a proof of concept and to get user test data for further development of the project.

Chapter one of the written part of the work is the introduction. The introduction clarifies my relationship to sound and the web. The introduction also briefly explains design objectives, provides some reflections on sound and the web in general, describes the structure of the written part of the work, and introduces the concept giving the reader some kind of idea of what the Audiotarium is.

Chapter two goes into more detail about design principles by introducing concepts related to the Audiotarium.

Chapter three describes the Audiotarium concept. The description goes through the four elements of the concept which are the Audiotarium ideology and rules, the main parts of the concept, good practices, and the technical structure.

Chapter four describes the Audiotarium prototype. The prototype description tells what the prototype is and how it was developed.

Chapter five is results and it describes concrete results of the work.

Chapter six is a conclusion. In the conclusion all the key findings and learning outcomes are presented. Also, possible future development plans are introduced.

The reader is expected to be somewhat familiar with basics of web design and digital sound.

After the Appendix there is a list of acronyms which briefly describes each acronym used and provides the reader a link for more information about

phenomena related to the acronym.

## 2. Design Principles Explained

This chapter explains principles of the design of the Audiotarium. These principles have served as guidelines for the concept and for the prototype. The design objectives of the work introduced in chapter 1.2 have been defined to follow principles introduced in this chapter. These principles have been developed observing phenomena within the discourse of new media.

### 2.1 The Element of the Play

Playing has a very important role when human beings are trying to learn something. Experiments, tests, and prototypes are essential in all creative work. They are all forms of playing. To play is to try, to find out, to experiment, and sometimes to learn. One of the pioneers of a movement called Musique Concrète, Pierre Henri Marie Schaeffer developed a practice of using sound objects as building blocks of compositions. His ideas made him experiment a lot with sound recordings. Finally, he ended up stating the idea of a movement called Musique Concrète.

*This determination to compose with materials taken from an existing collection of experimental sounds, I name Musique Concrète to make well the place in which we find ourselves, no longer dependent upon preconceived sound abstractions, but now using fragments of sound existing concretely and considered as sound objects defined and whole. (Schaeffer, as cited in Balloran 2008)*

In the context of the Audiotarium, the idea of Musique Concrète has a fundamentally important role. To detach himself from the conventions of music and from the idea of preconceived sound, Schaeffer had to make lot of experiments. He had to play with sound. The element of the play is the motivation, the main inspiration and a tool for the Audiotarium concept. What I call *the play* is an idea which is built upon work done by Musique Concrète pioneers who opened their minds and inspected sound objects as fundamentally usable elements of an artistic composition. They took a step

further from the conventions of Western music culture by widening the idea of a composition and by introducing a new way to see sounds as objects.

The Audiotarium is designed to honour the work of pioneers of Musique Concrète and especially the notion of play. Built in to the Audiotarium there is also an objective to expand the world of sound by using modern technology as a facilitator for operating with sound objects. The global network, the Internet, provides many interesting ways for sound artists to collaborate and to use Audiotarium sound objects to play together.

The Audiotarium concept treats sound objects like sonic quotes of reality. In the Audiotarium, the sound object is a digital representation of the original sound. These sonic quotes are made and shared by users of the Audiotarium. Because of the public domain nature of the Audiotarium, no-one owns these quotes or copyrights to these quotes. It is important to communicate this to users before they participate by adding sound objects to the Audiotarium. Thus, the public domain ideology has to be emphasised carefully within the whole Audiotarium web site. Since all sound objects in the Audiotarium are public domain sound objects, users of the Audiotarium have complete freedom to play with them and maybe even produce something that they themselves can call a work of art. If this is the case, they have all the copyrights regarding the end product created by using sound objects from the Audiotarium.

In the Audiotarium, people are encouraged to create their sound objects by recording them by themselves. In most cases, this procedure should ensure that users have a right to announce the sound object to be a public domain sound object. There is still a considerable grey area of what should be allowed and what should not. The famous case of Amen Break (Harrison 2004) from The Winstons is a good example taken from the context of sampling culture. The case of the Amen Break has shown that it is not easy to always define who has a right to what. Amen Break is a drum loop sampled from an old The Winstons album. It is about 6 seconds long and was originally recorded in 1969 by The Winstons. After that, the loop has been used in a vast amount of musical pieces. Richard Spencer, a founding member of The Winstons, owns the copyright to the song where the loop appears. For many years, the Amen Break was used as a part of new

pieces of art made by other people. Spencer let this happen without any regulations or demands. Because of this, the Amen Break achieved some kind of status in the cultural public domain. Legally, however, Spencer still owned the copyright to the loop. Later, a company called Zero-G used the Amen Break on their loop collection CD which was copyrighted in 2002. So now there exists two copyrights of the same material. The use of Amen Break has become legally quite confusing. The case of Amen Break shows that copyright systems are not always watertight. Rich public domain is essential for all innovations within a culture. The case of Amen Break shows that overprotecting intellectual property is likely to lead into a situation which slows down or even stops the development of culture. If existing cultural entities are difficult to use or if the usage of those entities is impossible because of confusion about rights and laws the system either forces people to break laws to produce cultural phenomena like remixes and mashups or in the worse case the whole culture slowly vanishes because of these restrictions. In addition, it is remarkable to note that all innovations we have are derived from previous cultural innovations. Even the Western scientific system is partly based on the idea of referencing. Generations of people have been working and finding out different kinds of things before current generations. and current generations benefit from that information in multiple ways.

In the context of the Audiotarium, users are encouraged to discuss what is acceptable and what is not. In practical terms, administrators still need to reserve a right to remove any sound object at any time from the Audiotarium repository to ensure that the service can preserve a legally durable status.

The element of the play is the key. The experimental nature of the service has been planned to encourage users to play, to continue traditions of Musique Concrète, to do it legally, to respect other people, and to expand the element of the play by means of the web and digital technology. All of this can be ensured by building a public domain sound repository.

## 2.2 Information Wants to be Free

*Information Wants To Be Free. Information also wants to be expensive. Information wants to be free because it has become so cheap to distribute, copy, and recombine---too cheap to meter. It wants to be expensive because it can be immeasurably valuable to the recipient. That tension will not go away. It leads to endless wrenching debate about price, copyright, 'intellectual property', the moral rightness of casual distribution, because each round of new devices makes the tension worse, not better. (Brand 1987)*

The Audiotarium concept has been built to make the price of getting sound objects to the web even lower than it is already. Digital recording technology and the development of personal computers have increased possibilities to create and share sound objects. Now, in the year 2008, sound samples are used everywhere. It is not difficult to get samples, it is not difficult to record samples, and it is not difficult to share samples. In the context of the Audiotarium, there is an effort to go around most of the copyright issues simply by encouraging users to produce and liberate their own sound objects. The Audiotarium is still not a solution to all the copyright issues, the tension between the value and zero value will be there. The Audiotarium provides the user a possibility to participate an effort to create repository of free sound objects.

The Audiotarium concept has been built to passively fight the idea of owning sound objects. This fight is not active in the sense that users of the Audiotarium would be encouraged to break existing laws. The passive way is to build a free system of sound objects. One possible outcome of this kind of system is that one day there will not be any sense in charging for any kind of use of sound objects because all the sounds will be available also for free.

The freedom of sounds is not only about the price. Digital sound objects and software both consist of bits so they can be seen as belonging to digital domain. Therefore, it is relevant to make some comparison between sound and software.

The GNU project defines the free software. These two sentences are quotes from The Free Software Definition :

*Free software" is a matter of liberty, not price. Free software is a matter of the users' freedom to run, copy, distribute, study, change and improve the software. (gnu.org 1996)*

Digital tools provide interesting possibilities considering the liberation of sound objects. It is easy to make a digital copy of a sound. It does not require anything but skill to copy a file. Because of the Internet, it is also easy to distribute sounds. People with basic web skills can share sounds using a computer that is connected to the Internet. To change and to improve sounds requires a little bit more professional skill. On the other hand, there are lot of tools available for people who are interested in working with sound. Sounds do not have to be edited to distribute them or to play with them. The Audiotarium concept aims to lower the threshold to study sound and to play with sound by making this whole process easy and simple.

Richard Stallman, the father of the GNU General Public License writes :

*Since free refers to freedom, not to price, there is no contradiction between selling copies and free software. (Stallman as cited in DiBona et al. 1999)*

This is a very important notion about the freedom of digital entities. The freedom is not only about the price, but it is more about the liberty to use digital entities like digital sound objects without having to worry if someone owns the sound object or not. If software is released under GNU General Public License, it has been liberated in a sense that no-one holds the right to say what can be made with the software or how it will be modified. The Audiotarium concept can be seen as a tool to liberate sound objects in similar manner as the GNU General Public License can be seen as a tool to liberate source code of software. The Audiotarium does not provide any way to release sound objects under a license, but instead gives the user a

possibility to declare the sound object to be a public domain sound object. The idea of the Audiotarium differs this way slightly from the GNU General Public License idea since there are no licenses used at all in the Audiotarium. The user only accepts the public domain dedication when he or she is adding the sound object to the Audiotarium. After that, the sound has entered irreversibly the public domain.

## **2.3 Public Domain and the Notion of Credits**

Sounds which are already copyrighted cannot be declared to be public domain sounds. Public domain sound objects cannot be protected by copyrights, either. The process of declaring a sound object to be a public domain sound object is irreversible. It is important for the user of the Audiotarium to be aware of all of the consequences of the process of adding a sound object to the Audiotarium and declaring the sound object to be a public domain sound. The importance of understanding one's own actions when using the Audiotarium to declare sound objects as public domain sound objects will be emphasised in the UI design of the final version of the Audiotarium.

The Creative Commons Public Domain Dedication (2008) is used as a guideline for the public domain declaration in the context of the Audiotarium concept. The declaration can be found from the Creative Commons web site and within the scope of this work from Appendix D.

The public domain nature of the concept is in a very important role as a supporter of the element of the play. The fact that the Audiotarium's sound objects are public domain sound objects makes the use of sound objects very easy. The user does not have to spend energy to think about what kind of licences and copyrights are involved. There are none.

When the creator of the sound object declares the sound object to be a public domain sound object, he or she gives up any legal rights to control possible usage of the sound object. There is no obligation for the user of the sound object to credit the creator of the sound object.

However, good manners are respected in the context of the Audiotarium and when sound objects are listed in the Audiotarium, the information of the user who uploaded the sound is provided. When the Audiotarium API is used to access the Audiotarium sound repository, the API provides the Audiotarium user the name of the user who has added the sound object. Good practices of the Audiotarium include instructions for crediting sounds. This is not obligatory, but very strongly recommended since public domain intellectual property only exists because of the people who are willing to participate in the common effort to build a public pool of free cultural entities.

## 2.4 Creative Commons and Free Culture

*The Creative Commons is a nonprofit corporation established in Massachusetts, but with its home at Stanford University. Its aim is to build a layer of reasonable copyright on top of the extremes that now reign. It does this by making it easy for people to build upon other people's work, by making it simple for creators to express the freedom for others to take and build upon their work. Simple tags, tied to human-readable descriptions, tied to bulletproof licenses, make this possible. (Lessig 2004, p. 282)*

Creative Commons was founded by Lawrence Lessig, a professor at Stanford Law School. The ideological foundation of the company is noble. Free culture is about liberating cultural entities by seeing them as belonging to public domain. This is exactly what the Audiotarium is about. So in principal, the Audiotarium belongs to a movement which can be called called Free Culture. On a practical level, the Audiotarium is pretty far from the system of Creative Commons. The Audiotarium can be seen representing one extreme which now "reigns" as Lessig puts it. This extreme is complete freedom, the real, pure idea of public domain.

Creative Commons provides a system of licensing one's work. Licenses the corporation provides are Open Source licences. They all aim to have free usage of cultural entities. However, there is a possibility to customise a license if one wants to license one's work. The user of a Creative Commons

license has a possibility to choose a set of conditions which regulates the usage of the product licensed using Creative Commons licences.

Conditions which Creative Commons provides are Attribution, Noncommercial, No Derivative Works and Share Alike. All of these are good for certain circumstances. However, the Audiotarium is about liberating sound and about making sound free. Only the public domain dedication can provide a way to make a work of art completely free. Unfortunately, there is no compromise between freedom and regulation. Regulation is not freedom.

The way Creative Commons works is interesting and the Audiotarium is not opposed to this way of making open licenses. The Audiotarium project is parallel to this system, not against it.

The practical reason for the Audiotarium to stick to public domain only rules is that if there are several different ways to license sound objects, the automated usage and crediting of providers of sound objects becomes too time consuming, too complex and too difficult from the perspective of the user. The Creative Commons licensing system results into the play becoming a complex process of trying to take care of crediting in order to avoid using sound objects. In the context of the Audiotarium, the play is everything and the process of the play should remain as easy and fun as possible. Technical or licence related issues must not increase the threshold of starting the play.

## **2.5 Sound as a Physical Phenomenon**

Sound is a physical phenomenon. Sound waves are vibrations transmitted through a solid, liquid, or gas and sound waves have properties and characteristics. In a physical sense, the main properties of a sound wave are frequency, wavelength, period, amplitude, intensity and speed. (Keller et al. 1993)

Sound waves can be recorded with a device called microphone. What happens when a sound wave is being recorded is that the microphone

converts changes in atmospheric pressure, meaning sound waves, into an electrical signal. This electrical signal can then be analysed, processed, or stored.

The recording process does not capture the original sound wave, but creates an electrical representation of the sound wave. Analogue recording refers to the process of producing an electrical representation of a sound. There are many things which affect the result of the recording process. Two main factors affecting the result are the recording space and the recording device. A recording can be produced by using natural sound sources or artificial sound sources like sound synthesis software.

The electrical representation of a sound is not the original sound and it can be said that by making decisions about the recording space and the recording device, the person who is recording a sound is doing creative work. However, it is debatable how creative this work is since there will always have to be a microphone to catch the representation of a sound wave. This means that the person who records the sound just produces a sonic quote of a sound source with a device designed for this quoting.

Even if the recording process would be considered to be creative activity, the person who records a sound still has the freedom to release the representation of a sound as a public domain sound object if he or she wants to.

Someone can claim to own a recording of wind blowing on a mountain top, but anyone can go to the mountain top and record the same sound. Sound itself is a process which happens because of the nature of the physical world we live in. This process can be owned only if physical phenomena can be owned. Sound as a phenomenon happens and it would happen even if there was not anyone to call it sound or record it. It is a matter of debate if natural phenomena should be owned by someone but in the context of the Audiotarium the assumption is that no-one should be allowed to own natural phenomenon like sound.

In the Audiotarium, users will have to produce and record their sounds. By doing so they will have all the rights to release those sounds as public

domain sound objects. It is good to remember that someone might be able to produce and record sound waves which resemble, for example, classical drum machine sounds. If this happens, the person should have a right to announce these sounds to be public domain sounds if he or she wants to.

An electrical representation of a sound wave can be transformed again into digital form which is the form that computers are able to deal with. The process of transforming an electrical representation of a sound wave into digital form is often called sampling. The electrical representation is said to be analogue and the binary representation is said to be digital. Digital representation of recorded sound is not mathematically continuous, but depending on the sampling rate the digital representation may be exact enough to be used to produce an audible sound that can be perceived to be a representation of the original sound. Technically, this is very interesting since recording tools and speakers are never optimal and the sound that is played from a speaker can never be exact copy of the original sound. Also, the space where the sound is played affects the way the observer perceives the sound.

Digitally it is possible to make exact copies of sounds, but even digital copies sound different depending on how they are used and what are the properties of playback devices and the playback environment. So it is very difficult to say when a sound has been copied and when not. It takes a lot of work to find out and prove if two sounds are copies of each other.

Considering how easy it is to digitally edit and change sounds so that they do not resemble each other, it can be said that it is almost impossible to know sources of all the sound files. This can be seen as a problem when a sound artist is using a sound from a public sound repository and he or she does not exactly know what is the origin of the sound. The work process becomes very difficult if the artist has to check backgrounds of all the sounds he or she is using. In most of the cases, this is probably not even possible. In the context of the Audiotarium the user who adds the sound object is responsible for the added material.

In the case of the Audiotarium, a digital sound is treated as a physical phenomenon which has been recorded. The original sound has been transformed into computer readable bits. This means that the focus of the

Audiotarium is on dealing with data objects which describe rules for reproducing sound waves which have been recorded by someone or something. These sound objects consist of a sound file and of meta data related to that sound file.

From a practical viewpoint, this all means that the process of adding a sound object to the Audiotarium means that someone or something records a sound, digitises the sound, uploads the sound file to the Audiotarium, and provides some meta data about the sound. By doing this, the user declares the sound object to be a public domain sound object. This process should be as easy and clear as possible to maintain the ease of the play. The process leads to new, free sound object which is not controlled by copyright regulations. In the case of the Audiotarium, users are encouraged to be prepared to prove that they have recorded the sound they have released. It is still possible that the origin of the sound cannot be known, but this process of user generated content creation relies on the fact that the user has a responsibility for what he or she announces to be a public domain sound object.

## **2.6 Contextually Relevant Web Services**

Some existing web services were observed when the Audiotarium prototype was designed. Not all these services are built around sound, but they provide quite the same kind of functionality as the Audiotarium. Some comparison between these services and the Audiotarium is needed to introduce similarities, adopted ideas, and main differences. There are many services quite similar to the Audiotarium, but these three have been picked because they are the most visible ones and I, the writer of this work, was the most familiar with these three services.

### **2.6.1 YouTube**

YouTube (<http://www.youtube.com>) is a video repository service. Users of YouTube can upload their videos to the service where they are publicly available for anyone to see. Users of YouTube have to be copyright owners of the videos they have uploaded. Lately, copyright violations have been a

major issue for YouTube because people have not followed the rules of YouTube very thoroughly. It is likely that because of the massive quantity of data, it is very difficult to arrange sufficient monitoring, filtering and administration for the service. YouTube has own media servers taking care of the traffic.

From the UI perspective YouTube offers great solutions to pack lots of information into a simple looking UI. There are several different ways to browse the content of YouTube. Search is the main function and it is located on the first page where it is very visible. In addition to the search functionality, there are tags, user ratings, and categories to help browsing videos. Users who have uploaded videos have public profile pages where anyone can see a list of videos uploaded by the user. All the popular ways to browse large media repositories are available.

YouTube also has an API. Their API is divided into four sections. They have a Data API, Player APIs, Custom Player and Widgets. This means that they provide almost all of their functionality for developers to build applications. This is not the idea of the Audiotarium. The Audiotarium focuses on providing the content, sound objects. The idea of the Audiotarium is not to bind developers too tightly to the Audiotarium functionality. In the Audiotarium the functionality should mainly be built on the application side of things.

A media server would make it easier to monitor traffic since the Audiotarium functionality could be bound together with the server software. It would also make scalability only a matter of getting more hardware. To build the Audiotarium on its own separate server would require much more financial resources than available at the moment.

YouTube's rules and copyright regulations are more complex than in the Audiotarium, due to the fact that the Audiotarium deals only with public domain content.

In the Audiotarium there is no technical support for categories and groups. Good practices of the Audiotarium encourage users to group sound objects by tagging them in an intelligent way.

In the Audiotarium, there is only one kind of search functionality available and this search is used to browse the repository. The search results of the Audiotarium are grouped in four groups : hits by tag, hits by title, hits by description and hits by id. All of these groups are always shown even if there would not be any hits in the group with the given search term.

### **2.6.2 Flickr**

Flickr (<http://www.flickr.com>) is a photo repository service. It works almost the same way as YouTube except that the focus is on photos. Flickr also has search functionality, tags, and user pages. Flickr allows users to make some of the images public and some private. The UI is simple and provides good ideas for the future development of the Audiotarium UI. The Flickr design has been successful in packing lots of information into small space.

The most inspirational part of Flickr is the API. It is very comprehensive and provides sophisticated access to Flickr image data. The Audiotarium API is much simpler, and has only four different functions. The Flickr API focuses on content just as the Audiotarium API. The Flickr API is simple to use and provides easy access to the Flickr database.

Flickr's rules and copyright regulations are more complex than rules of the Audiotarium. Users can decide how they want to release their material in the Flickr. To use images from Flickr one has to be pretty careful. The Flickr API has many request formats and many response formats. The Audiotarium API has many response formats as well.

### **2.6.3 The Freesound Project**

The most interesting and the most inspirational service from the perspective of the Audiotarium is the Freesound Project (<http://www.freesound.org>). The Freesound Project provides a solid and very well designed platform for a sound repository. The project provides all of the same things as the Audiotarium, except for the API.

The Freesound Project has technically much better solutions for UI

components than the Audiotarium. For example, the sound player shows the visual wave form of the sound. This visual waveform can be used to estimate how distorted the sound is. The Freesound Project also provides support for formats other than mp3s.

The Freesound Project lets users use a Creative Commons Sampling plus 1.0 license. This affects the use of sounds. One has to always remember to find out whose sound was used and one also needs to credit those whose sounds one has been using. The one who has created the sound has a right to define how the crediting should be done.

The Freesound Project has an API but their API has been implemented using an OpenSource programming library called Mootcher. This affects the element of the play quite a lot. Mootcher is kind of an equivalent of an API, but requires slightly more programming skills than the Audiotarium API. Mootcher is aimed mainly for sound professionals who are able to use programming languages like C (Ritchie 1974) or graphical programming environments like Max MSP (Cycling '74 1997) or Pure Data (Pucket 1996). Another option of using sounds in Freesound Project is to download sounds from the website <http://www.freesound.org>. In this case, users have to manually credit each of the people who have originally created sounds.

The biggest difference between the Audiotarium and the Freesound Project is the issue of licensing. The Audiotarium is about liberating sounds, not about licensing them.

There are differences in the crediting system as well. Good practises propose crediting in the Audiotarium, but it is easier to do than in the Freesound Project since the Audiotarium user name mentioned in the context of the sound object is already more than polite and one can get it automatically via the API. After all the crediting is not obligatory in the Audiotarium.

Some kind of collaboration might be possible between the Audiotarium and the Freesound Project, but it is not necessary. The Audiotarium is not competing with the Freesound Project and these two services seem to have slightly different perspective on free sound.

## **2.7 Affordances, Perceived Affordances and Cultural Constraints**

In this work, design principles have been affected by ideas of affordances and constraints. The idea of affordances was introduced by perceptual psychologist J.J. Gibson (1979) to refer actionable properties between the world and the actor. The idea of an affordance was popularised afterwards by Donald Norman (jnd.org 2004), a professor emeritus of Cognitive Science and a professor of Computer Science. Gibson's idea of affordances is that they are relationships and they exist naturally, meaning that they do not have to be visible, known or desirable. Norman's idea of perceived affordances is that they are visible and the designer deals mainly with them. In screen based design almost all constraints are perceived since computers come with real affordances which a designer cannot change. Norman also uses the term cultural constraints which he explains to be conventions shared by a cultural group (2004).

Since the Audiotarium concept provides a platform, the design has mainly been guided by cultural constraints. However, there are also some real affordances. Perceived affordances will be more important when the final UI design is complete.

Real affordances of the Audiotarium are mainly hardware based. The Audiotarium is a screen-based system. It is also a web application. Thus, the main affordances are related to the screen and to the web. The web offers the affordance of sharing things if there is a place on the web where to share. This is what the Audiotarium is meant to be, a place for sharing sound objects and experiments. The web also offers an affordance of collaboration and communication. There are many ways to collaborate. The web service itself can be seen as a collaborative effort to build something together. Especially in the final version of the Audiotarium where there will be tools to afford a web community to become active by discussing and sharing media.

The design of the Audiotarium has mainly been benefiting from cultural constraints. People, especially young people today, are very accustomed to use media repositories like YouTube and Flickr. They have learned conventions to technically create and upload media to these services. They also have conventions for sharing personal media and building web communities around innovative phenomena. All of these things have been considered when the Audiotarium concept has been planned. Later on in the final UI design conventions will have an even bigger role. The user should be familiar with actions needed to get media into the repository so conventions will help the UI designer choose the proper perceived affordances to be used. The interaction design of the UI will benefit from the fact that most of the users who are familiar with digital recording are very likely familiar with conventions of how to deal with digital sound objects.

The design of the Audiotarium API relies on cultural constraints. APIs are normal things to have in media repository services. The way APIs work has become a standard. So there is very strong convention on how the API can be used and what can be done with it. Many people who are not familiar with programming still know approximately what can be done using, for example, the API of Flickr.

On the other hand, there is a cultural constraint that the Audiotarium is trying to change. It is the idea of owning sound. A major purpose of the design of the Audiotarium is to break this cultural constraint so that there is not the constraint of complex restrictions to disturb the fluent and continuous process of the play.

## **2.8 About Accessibility**

Since the Audiotarium is about sound, it is likely that visually impaired people would like to use it. This must be possible, but in the context of this work the deliberation on how to make this possible is not dealt. The accessibility is such a big part of the service that it needs a lot of focus and within this work it was not possible to design all of the accessibility required to reach visually impaired people. However, the WordPress ([wordpress.org](http://wordpress.org))

2003) platform offers a structure of a web site which is well designed in consideration of visually impaired people. Flash is not the best thing to have, so in the final version of the Audiotarium the sound player could be built using JavaScript. The Audiotarium concept does not give any guidelines for developers on how to approach challenges of accessibility, but many web designers know how to do web applications to make them as accessible as possible. In good practises of the Audiotarium, there is a statement about accessibility when using the Audiotarium sound objects and the Audiotarium API.

## 3. The Audiotarium Concept

This chapter describes the Audiotarium concept. Not all of the things mentioned in the concept description have been implemented in the prototype. The prototype tolerantly implements only the most essential functional parts of the whole concept.

### 3.1 The Audiotarium Ideology and Rules

Together, the Audiotarium ideology and the Audiotarium rules are what defines the practical manifestation of the Audiotarium. They form an ideological framework for the Audiotarium web service. This ideological framework serves as a guideline for the design and binds together all of the parts of the Audiotarium concept. The ideological framework is the heart and veins of the Audiotarium concept.

#### 3.1.1 The Audiotarium Ideology

The Audiotarium ideology is quite simple.  
Let's make sounds free.

A work of art is something that an artist calls a work of art.  
If one calls one's sound object as a work of art, one should not add it to the Audiotarium. All of the sound objects in the Audiotarium are public domain sound objects.

#### **The Audiotarium does :**

- encourage people to create and share public domain sound objects
- encourage people to think what can be thought to be a sound and what is a work of art
- encourage people to be creative and to play
- encourage people to join public effort and to do things together
- encourage people to questionize existing moral values and laws and to act democratically to change them

## **The Audiotarium does not :**

- encourage people to break laws
- encourage people to break existing copyright laws
- support any kind of illegal activity
- support piracy

### **3.1.2 The Audiotarium Rules**

#### **1. No copyrighted material!**

Users are not allowed to upload copyrighted sound material to the Audiotarium web service. If they do so, they shall take all the responsibility for their actions meaning both moral and legal responsibility. All of the uploaded sound material has to be recorded or synthesized by the user who adds the sound material to the Audiotarium web service.

By uploading files to the Audiotarium web service, the user testifies that those files can be declared to be public domain sound objects. Failure to comply with the rules of the Audiotarium web service (for example by uploading copyrighted material) will result in the user being banned from the Audiotarium web service and legal actions will be taken against the violator. Uploads are traced and directly linked to the user of the system. The personal information of the violator will be handed to possible third parties of the violation.

It is a matter of philosophical discussion what can be considered a sound object and what is a copyright violation, so the Audiotarium Group reserves the right to remove any sound at any time for any any reason from the Audiotarium. The user also accepts that unacceptable behaviour or illegal lingual content may lead to the removal of the sound and banning from the Audiotarium.

#### **2. All the sounds in the Audiotarium are Public Domain sounds!**

When the user uploads a sound to the Audiotarium the sound becomes a

public domain sound. Creative Commons Public Domain Dedication explains what it means that the sound is a public domain sound. This definition can be found from the Creative Commons web site and as a text only version from Appendix D.

### **3. The Membership**

When the user is registering to become a member of the Audiotarium Group (all registered users of the Audiotarium web service also belong to the Audiotarium Group) a user has to provide his / her real name, real birthday and real contact information. This information is needed for possible cases of violations. The personal information of the user will not be handed to third parties except in the case that the user has been breaking the law or in the case that the user can be suspected of violation of law.

The user is fully responsible for the accuracy of the information he or she provides when he or she is registering to the Audiotarium web service.

By becoming a member of the Audiotarium web service the user accepts all of the Audiotarium rules.

## 3.2 Main Parts of the Concept

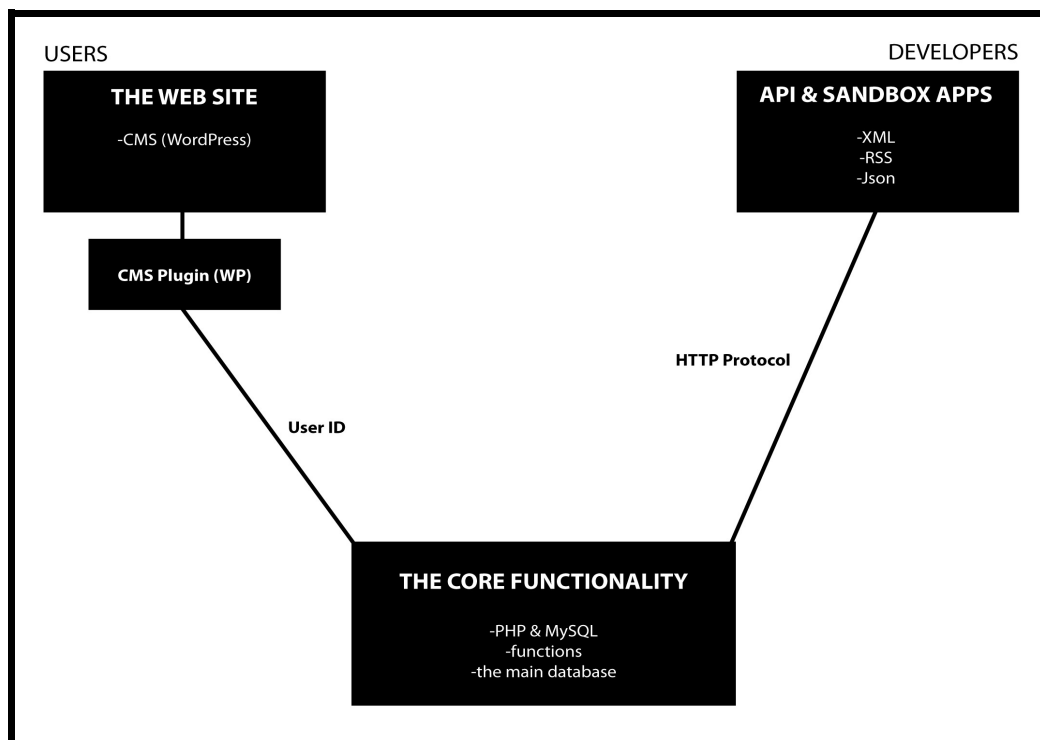


Image 1 : The Main Structure of the Audiotarium Concept

### 3.2.1 The Public Domain Sound Repository

The technical solution of the Audiotarium concept focuses on providing a web based solution to collect public domain sound objects into one service. In the context of the Audiotarium, a sound is more than just a sound file. A sound object is a sound file plus all the meta data related to the sound file.

#### The Essential Sound Object Data

- The id of the user who adds the sound to the Audiotarium
- The id of the sound
- The sound file
- The title of the sound
- The description of the sound

#### The Additional Sound Object Data

- The technical quality of the sound file (bit rate or sample rate)

- Recording location information (longitude and latitude)
- Tags

The Audiotarium sound object has a minimum stage of existence. In this minimum stage, the object consists of a sound file, a system generated user id (which is created when the user registers himself or herself to the service), a user provided sound object name, a user provided sound object description and a system generated individual id. The system generated individual id is used to individualise sound objects in the Audiotarium. From the users perspective the title and the description of a sound object are obligatory which means that the user who adds a sound object to the Audiotarium has to provide this information in addition to the actual sound file. Even though all of the sounds in the Audiotarium are public domain sounds, some kind of user system is needed to control the usage and to track possible rule breakers. The user id provides a way to know who has added the sound to the Audiotarium.

The Audiotarium sound repository does not only consist of sound objects, but also of basic functionality which makes it possible for the Audiotarium UI layer to communicate with the repository. Together, this functionality and the repository are called the core system of the Audiotarium.

The sound object repository is a fundamental part of the Audiotarium concept. The technical platform of the repository provides users a possibility to build a collection of free sounds. Main roles of users can be seen as builders of the web community and the repository. The technology is nothing without users.

### **3.2.2 The API and Sandbox Applications**

API is an acronym for Application Programming Interface (API). In practical terms, an API is a set of functions which provides services to support requests made by other computer programs. The Audiotarium API provides a standardised access to the Audiotarium sound repository so that developers can make their software to use Audiotarium sound repository sounds straight from the Audiotarium database instead of first downloading sounds manually. This approach provides many more possibilities than the

manual download approach. Developers can use the Audiotarium API to make their software do different kinds of requests to the Audiotarium sound repository based on sound meta data. For example, location information can be used as a rule to fetch sounds from the Audiotarium repository.

All of the possibilities of the API cannot be predicted and it is not even rational to try since the Audiotarium API exists to encourage developers to play with the Audiotarium. The idea is not to give instructions what to do with sounds, but to provide them for any kind of use.

Applications which use the Audiotarium API are called Audiotarium Sandbox Applications. The Sandbox metaphor refers to the element of the play. If the developer provides an URL to his or her application, the application is listed on the Audiotarium web site. The idea is to present as much creative work as possible based on the Audiotarium concept, in the context of the service. This is done to inspire the community to play and to build their own applications.

Often, an API is considered to be a technical approach and this is true. But non-technical people should also be perfectly capable to design applications that use the API because of the simplicity of the idea. The actual implementation is not difficult, but requires a little bit of coding skills. In the context of the Audiotarium, there are two potential groups who could benefit from using the service, users and developers. There is a grey area between the two which is ignored way too often.

In the context of the Audiotarium users and developers can get closer and even mix if developers develop applications which can be used, for example, as a integral part of the web site. All kinds of experimental applications are welcome to the experimental corner of the site. There are also some interesting possibilities worth exploring when it comes to collaborative activity and output channels of products done within the Audiotarium. For example, a virtual club where users inspect a live spectacle of people using Audiotarium tools to play together is definitely inspiring idea. So are all kinds of streams and public domain web radio station ideas. All this and much more is possible because of the API of the Audiotarium.

### 3.2.3 The Web Site and The Community

The visible entity of the Audiotarium is the web site, <http://www.audiotarium.net>. To make the threshold of starting the play as low as possible, the UI of the final version of the web site will be optimised so that the most crucial operations regarding sound objects are as intuitive as possible.

To inspire and to make the Audiotarium a real web phenomenon the service has to become a web community. The Audiotarium web community has tools for discussions and chatting. The discussion board is technically a separate part, but visually uniform with the rest of the service. It is also possible to discuss about each sound object separately. Additionally, the chat is technically separate but visually uniform with the rest of the service.

There is also an event organiser for the community so that they can organise Audiotarium based events either in a virtual space or in a real space. An experimental corner of the web site serves as some kind of aggregator for Audiotarium Sandbox Applications.

The administration of the service is designed to rely on the activity level of the community. There are super administrators who monitor the system, but the best way to arrange quite reliable monitoring is to let users tell what they think of sounds. There is a way to inform about copyright violation suspicions by clicking the button "I consider this to sound object to break copyrights". When even one user has pressed this button, the sound object goes to "questionable" mode and is only visible for registered users. The administration decides if the sound object stays or not but users can tell their opinion about the sound object in a discussion related to the sound. If a sound object has been found not to break copyrights it will stay in the Audiotarium and it cannot be reported again as questionable.

There is also a rating system which makes it possible for the user to tell his or her opinion about three factors of the sound object. These factors are technical quality, artistic quality, and the user's general opinion. All the factors have two possible values: good and not good. Over time, this should produce similar system of evaluations as Amazon's (Amazon.com, Inc 2008)

system where one can evaluate how the shopping went.

By using the WordPress blog engine as a content management system, the web site of the Audiotarium can be built as a combination of a WordPress theme and a WordPress plugin. This provides a sustainable way to maintain the service since the WordPress platform is actively developed and there are dozens of plugins available to WordPress. Updates of the Audiotarium are easy to install if the core system of the WordPress does not have to be modified to be used as content management system for the Audiotarium web site.

All of the communal features have been left out of the prototype .

### **3.3 Good Practises**

Good practises are not rules. They are not suggestions either. They are something in-between. The idea of good practices is to introduce practices that have been found to be good and well mannered when doing things in the context of the service. The idea is to avoid forcing technical solutions where conceptual, user centred solutions are possible.

#### **3.3.1 Credits**

The Audiotarium has been built to support the public domain thinking. This means that the user of a sound object does not have to give any credits to the user who added the sound object. Still, good practises of the Audiotarium encourage users to give some credit for the person who has added the sound. It is enough to somehow mention the Audiotarium user name of the one who added the sound. Technologically, this practise is supported by providing user names in the context of sound objects both when sounds are listed on the web site and when they are accessed via the Audiotarium API.

#### **3.3.2 Meta Data Conventions**

There are no groups, categories, or any other technological ways to group sound objects in the Audiotarium. Tagging is suggested to be used in an intelligent way to group things. For example, long random character tags can be used to group sounds. Users are also encouraged to tag sounds so that main themes related to the sound object are used as tags. The user can also provide a recording location information for the sound object. For this purpose, there is an OpenSource map UI component which allows a user to define the recording location by pointing and clicking the location on the map.

### **3.3.3 Sandbox Application List**

Developers using the Audiotarium API need an API key to access the API. The key system makes it possible to monitor who is using the API and possible attacks on server can be tracked and prevented.

When a developer is applying for a key, he or she needs to provide some information about his or her Sandbox Application project. The developer is required to provide a title and a description for the project. If the developer chooses to provide also the URL of the project web site, the project gets listed in the sidebar of the Audioratium web site. It is a good practice to provide the URL. The list of Sandbox Applications is meant to inspire the Audiotarium web community and to promote Sandbox Applications.

### **3.3.4 Statement About Accessibility**

Whenever it is technically feasible, developers using sounds from the Audiotarium should make their works accessible to everyone. Developers who use the Audiotarium API are especially encouraged to pay attention to design for visually impaired people.

## **3.4 The Technical Structure**

### **3.4.1 The Core System Structure**

The core system of the Audiotarium has two parts. The database and the

main functionality take care of the communication between the different parts and handle the database queries. The main functionality manages API requests. The table structure of the database is described in full detail in Appendix A.

The main functionality is implemented using PHP . PHP does not provide very sophisticated tools for dealing with audio. On the other hand, there is not very much to do with the actual sound file except to move it from one place to another which can be very easily done by using PHP. In the final version of the Audiotarium, some things need to be done using some other programming language besides PHP since there is a need to access the sound file meta data to parse the bit rate of the file.

The core system is built so that it is able to communicate with the Audiotarium plugin for WordPress. This makes it possible to keep the web site completely separated from the core functionality.

### **3.4.2 The API Structure**

The Audiotarium API is a set of functions that can be accessed by using a http protocol and URL parameters. In practice, this means that the Sandbox Application creates a http connection to certain URL and provides required URL parameters for the Audiotarium API. The API does database queries and returns either an error or requested data. The result can be received in several formats which are XML, RSS and JSON depending on needs of the developer. If the result is not an error, the result is a sound object or a list of sound objects. Three XML result examples can be found from Appendix C.

After the Sandbox Application has received a non-error result from the Audiotarium API, the application can load the actual sound file since the result includes a path from where the sound file can be downloaded.

A complete list of the Audiotarium API functions and parameters can be found from Appendix E.

The API URL is the same as the Audiotarium web site URL with the API folder name added after the main service URL. In the case of the

Audiotarium prototype, the API url is <http://www.audiotarium.net/API>. The default API format is XML. Other formats are available in API sub folders called RSS and JSON. RSS is a special case of XML format designed for purposes of delivering news. Json is an acronym of JavaScript Object Notation. JSON can be used to transmit JavaScript coded objects between different systems. For example in Python JSON objects will become automatically Python objects because of the same kind of object notation. The prototype provides only the default API.

All the API requests are formed following the same general pattern which is demonstrated in the following example request :

**`http://www.theaudiotariumdomain.net/API/?action=action_name&api_key=md5_coded_key&parameter_name=parameter_value...`**

The Audiotarium API is accessible only for registered users of the Audiotarium. All the developers who use the API need to get an API key. This key is needed to log all the actions done via the API. This ensures that all attempts to attack the server by using the API can be tracked. The API key is also used to monitor the traffic that Sandbox Applications cause by fetching data from the Audiotarium. There are no limits for this traffic, but it is important to monitor the traffic to know possible needs for widening the bandwidth. The API key can be requested by sending a web form which is available only to registered users.

## 4. The Prototype

This part of the work describes the prototype of the work.

The prototype was implemented during summer 2008 and took about three months to design and implement all the parts of the prototype. From the perspective of the creator of this work, the prototype was technically simple to do but because the process was iterative (meaning that the prototype was built couple of times from remains of old versions of the prototype), the structure of the prototype is not as simple as it could be. The WordPress engine was used non-orthodox way and this resulted in the situation where WordPress updates are not possible to do without losing the functionality of the Audiotarium.

### 4.1 Limitations of the Prototype

The prototype has some limitations regarding the functionality. Test users are not able to edit or delete any data. This has been a conscious decision since the implementation of that part of the functionality would had been extra work and it would not have enhanced the user experience very much.

Due to the bandwidth limit, the prototype only accepts mp3 files which are smaller than 300 kilobytes. Wav and aiff files should be supported in the final version of the Audiotarium.

When a test user is adding a sound to the Audiotarium, he or she has to provide the bit rate of the file manually. The bit rate should be parsed automatically from the sound file in the final version of the Audiotarium.

Communal parts of the web site are completely missing from the prototype. The whole web community section of the web site will have to be designed, implemented, and tested for the final version. The community section of the web site is very important and requires a very through design process to be usable and appealing, so it had to left out of the context of this work.

## 4.2 About Technologies

### 4.2.1 MySQL

The prototype uses a MySQL (MySQL AB 2008) database system. The WordPress blog engine which is used to offer the user account management creates needed tables and in addition to those tables there are the core Audiotarium system tables. The detailed table structure can be found from Appendix A.

There are five main tables for the core functionality of the prototype. A table called *audiotarium\_api\_key* is for storing the Audiotarium API keys. It has a reference to the user id so that keys can be linked to users.

A table called *audiotarium\_log\_api* is for logging all the API actions. The API writes a line to this table each time there is a request to the Audiotarium API. The IP of the requester is saved, the time of the request is saved, the API key used is saved, and the whole request string is saved. Therefore, all the Audiotarium API requests can be tracked down later if needed.

A table called *audiotarium\_sound\_info* includes all the essential information about sound objects. This information is linked to the user id so that each sound object can be linked to the user who has added the sound object.

A table called *audiotarium\_sound\_tags* includes tags related to sound objects.

A table called *audiotarium\_sound\_locations* includes the location information related to sound objects.

### 4.2.2 PHP

The programming language used to implement the functionality of the prototype is PHP. PHP is a server side programming language which can be integrated very tightly with HTML. This makes PHP very handy for doing fast tests and building a proof of concept type of web service prototypes.

PHP was chosen for the prototype because it is the most familiar programming language to me and because the WordPress blog engine is built using PHP and MySQL.

### **4.2.3 WordPress**

WordPress is an OpenSource blog engine which can be used to build web sites, blogs and other web services. The engine is available in [wordpress.org](http://wordpress.org). WordPress provides the user account management, theme support and a large number of plugins. WordPress was chosen to be used for the Audiotarium prototype because it is built using PHP and MySQL. There are also some other benefits to using WordPress such as it is easy to install manage and tweak, and it provides an access to library of plugins and pre-made functional parts.

WordPress was found to be a very good platform for the possible future development of the Audiotarium UI. The native structure of WordPress can be customised to work with the Audiotarium functionality so that user account information is synchronised between these two systems. Additionally, the post system of WordPress can be synchronised with the Audiotarium sound object system.

### **4.2.4 Flash & ActionScript**

In the prototype, there are two things which are made in Flash. The sound player on the Audiotarium web site is an OpenSource sound player from <http://www.webresourcesdepot.com/open-source-flash-mp3-player/>. The player is a little bit buggy and should not be used in the final version. Somehow, the playhead gets stuck even though the sound is playing.

The other thing done in Flash / ActionScript (Adobe Systems Incorporated 2008) is the example Sandbox Application called the Audiotarium Rotator. The Audiotarium Rotator is a simple application. There are three rotating objects. Each object makes a sound when it hits certain point on the orbit. The user of the Audiotarium Rotator can change sounds registered for

objects by copy-pasting public sound ids from the Audiotarium web site to text fields in the Audiotarium Rotator. This Sandbox Application was done only to demonstrate that it is possible to build applications that use the Audiotarium API. The Audiotarium Rotator uses one ActionScript3 class that takes care of the communication between the Audiotarium Rotator and the Audiotarium API. The code for this class can be found from Appendix B.

Flash / ActionScript technology is just one way to build applications which use the Audiotarium API. Practically any environment or programming language which is able to make a http request is able to communicate with the Audiotarium API.

#### **4.2.5 The OpenSource Map UI Component**

The map UI component of the prototype uses OpenSource map data from OpenStreetMap (<http://openstreetmap.org>). The map engine is the OpenLayers (<http://openlayers.org>) JavaScript map engine. This combination works well and maps are accurate enough for a user to define the recording location of the sound object.

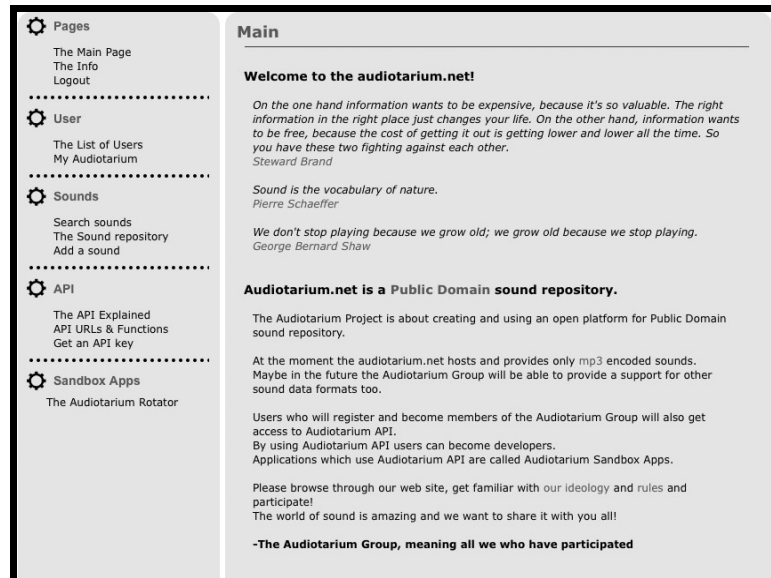
## 4.3 Pages

This part of the work describes pages of the Audiotarium prototype web site.

### 4.3.1 General pages

#### 4.3.1.1 The Main Page

The main page of the Audiotarium prototype is the usual welcome page. It describes the basic idea of the service and provides some information for test users if they are logged in. In the final version, there is a plan to put a map UI element on the first page. It would show either all of the sound objects with recording location information, or just randomly a geo-located sound object of the day. The main page is publicly available.



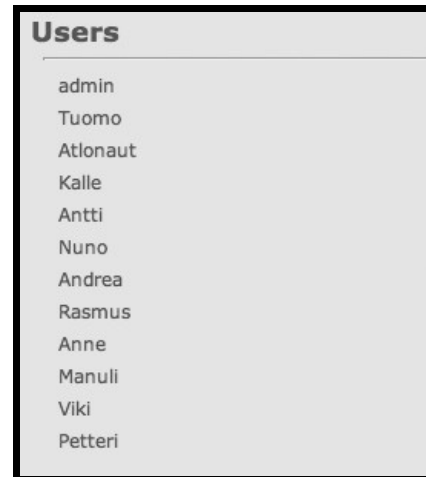
#### 4.3.1.2 The Info

The info page introduces both the Audiotarium Ideology and the Audiotarium Rules for the user. This page is supposed to provide all the information that is needed to use the Audiotarium web service. However, in the prototype there is only the Audiotarium ideology and the Audiotarium rules available. Test users are not required to follow rules because the content of the prototype is not public. It is also important to test the wrong kind of usage of the service. The info page is publicly available.

## 4.3.2 User related pages

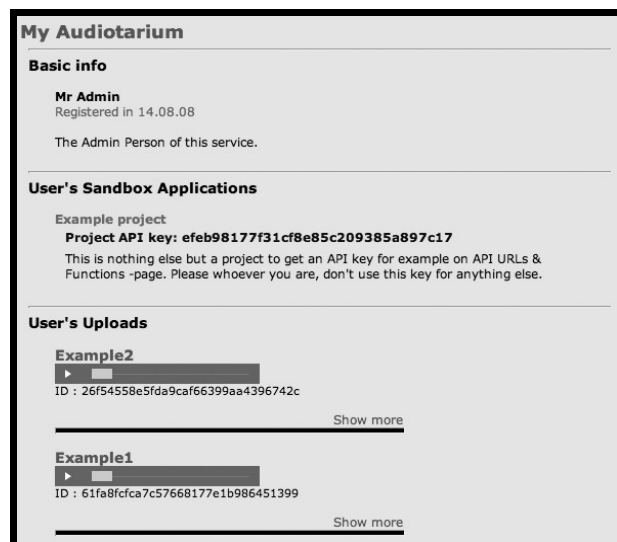
### 4.3.2.1 The List of Users

The list of users is very simple and clear in the prototype. There is just a list of user names. To get more information about certain user, the Audiotarium user needs to click the user name on the list. The user list is not publicly available in the prototype. The list will not be publicly available in the final version, either.



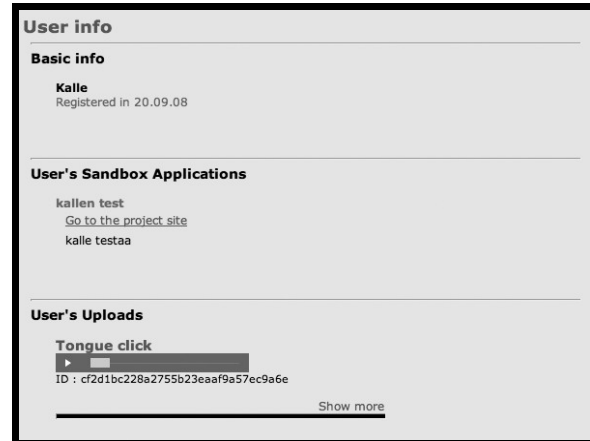
### 4.3.2.2 My Audiotarium

The My Audiotarium page is a page where a user can observe his or her personal information. The user information has been grouped in three groups. The basic user information includes the user name and the registration day of the user. On the My Audiotarium page there is also a list of the user's Sandbox Application projects. API keys of a user's projects can be found from the Sandbox Application list. The My Audiotarium page is available only to the user who has logged into the Audiotarium.



### 4.3.2.3 The User Info

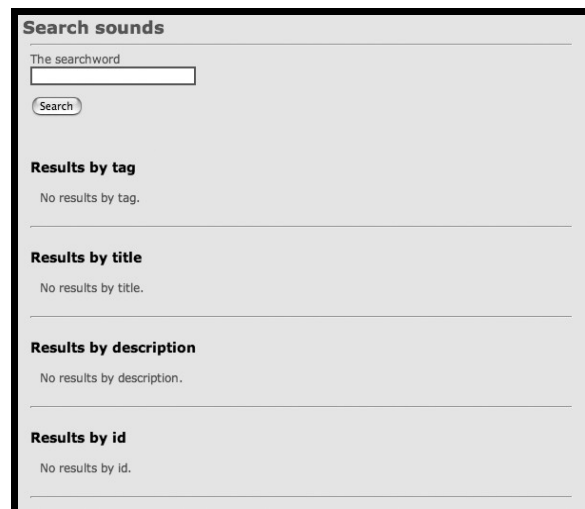
The user info page is similar to the My Audiotarium page with the exception that it is available to all the users of the Audiotarium who have logged in. This is the page that opens when a user clicks a user name link in the list of users. Users are not able to see each other's API keys.



### 4.3.3 Sounds

#### 4.3.3.1 Search Sounds

This page of the prototype is for browsing the Audiotarium sound repository by doing searches. There is one search field and search results are grouped into four groups based on the meta data that matched the search term. This approach is a little bit experimental and the idea is to test it with users to find out if it is a useful way to



implement the search functionality. It is very common that a user does not know what it means to do the search based on tags or keywords or whatever. Instead of confusing the user by asking him or her to choose which kind of search to use, this search method teaches the user how the search finds things. The search result is quite comprehensive.

Based on the information gathered from user tests, it seems to be a good idea to combine the search page with the actual sound repository page.

In the prototype the sound search page is available only to users who have logged in, but in the final version of the Audiotarium the search will be publicly available.

#### 4.3.3.2 The Sound Repository

The sound repository page of the prototype is very tricky from the perspective of the UI design. At the moment, the page lists all of the sound objects in the Audiotarium but of course this is not possible when there are more than ten or fifteen sound objects. In the future, if there are hundreds of sound objects even paging of the list is useless. The amount of data is crucial and this page needs more attention when the final version of the Audiotarium is designed. In the prototype this list is present, but in the final version the browsing of the repository must be arranged differently. One solution could be to have the search and some kind of listings based on user ratings. If the user wants to see all the sound objects, they could be shown as some kind of list but the user must be warned before the whole list is shown. And the list must have pages.



In the prototype the sound repository page is available only to users who

have logged in, but in the final version the page will be publicly available.

#### **4.3.3.3 Add a Sound**

On this page, there is a form for adding a sound object to the Audiotarium. In the prototype the form is ugly, but does what it needs to do. Obligatory fields have been marked with red and non-obligatory fields have been marked with black. The title and the description of the sound object are obligatory. Also, the actual sound file is obligatory. In the prototype the bit rate of the sound object has to be provided by the user, but in the final version of the Audiotarium it will be automatically parsed from the sound file.

There are five non-obligatory fields for tags and two non-obligatory fields for location coordinates, longitude and latitude. Longitude and latitude can be defined by using the map UI component so they don't have to be written by the user.

#### **4.3.4 The API**

##### **4.3.4.1 The API Explained**

This page explains the general idea of an API and gives a user an outlook of what can be done with the Audiotarium API. In the prototype this page is available only to users who have logged in, but in the final version the page will be publicly available.

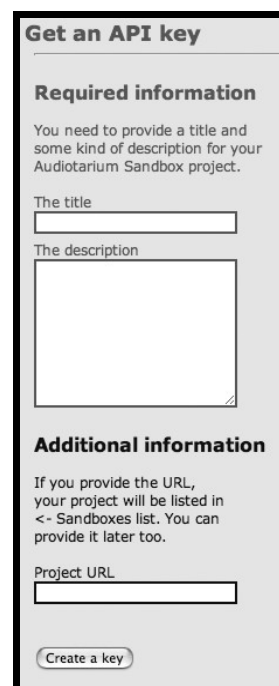
##### **4.3.4.2 API URLs and Functions**

From this page the user can find instructions on how to use the Audiotarium API. There is a complete list of the API functions and parameters (see Appendix E). Some functional examples of API requests are available so that the user can get familiar with the logic of the API. Also, URLs for different kinds of APIs can be found on this page. The

prototype provides only the XML API. In the prototype this page is available only to users who have logged in, but in the final version of the Auditorium the will be publicly available.

#### 4.3.4.3 Get an API Key

This page is where a developer goes to get an API key. There is a simple web form with three fields. The title and the description of the project are obligatory fields, and the URL of the project is not. If the user provides the URL of the project, the application will be listed on the Auditorium web site. In the final version, there has to be a page where user can edit the project information. All kinds of editing and deleting operations have been left out from the prototype. Obviously only registered users can get an API key.



The screenshot shows a web form titled "Get an API key". It is divided into two sections: "Required information" and "Additional information".

**Required information**  
You need to provide a title and some kind of description for your Auditorium Sandbox project.

The title

The description

**Additional information**  
If you provide the URL, your project will be listed in <- Sandboxes list. You can provide it later too.

Project URL

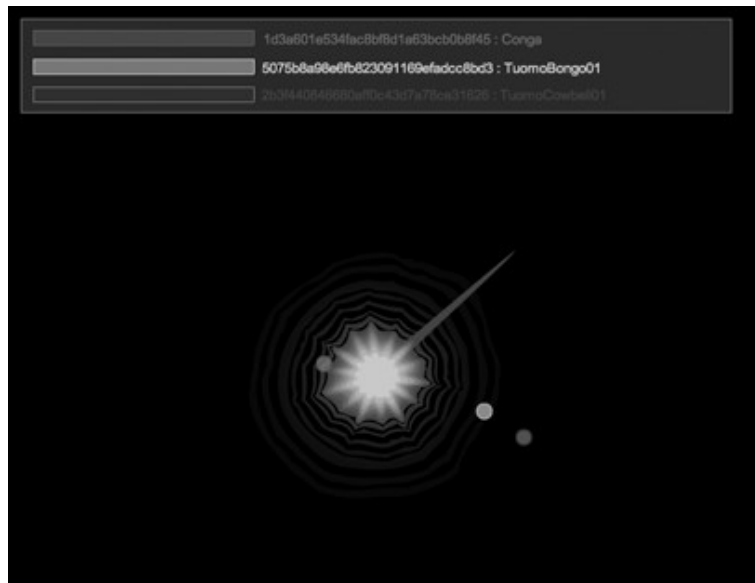
#### 4.3.5 Sandbox Applications

##### 4.3.5.1 The List of Sandbox Applications

In the side bar, there is a list of Auditorium Sandbox Applications. Names of the applications are links to web sites of projects. A developer can make his or her Sandbox Application appear on the list by providing the Sandbox Application project URL.

### 4.3.5.2 The Audiotarium Rotator

The Audiotarium Rotator is an example of a Audiotarium Sandbox Application. The application is physically located at a different URL than the Audiotarium web site just to demonstrate that the application



does not have to be in the same place as the Audiotarium web site. The Audiotarium Rotator is a strange, small application. In the Audiotarium Rotator, three small circular objects go around a bigger graphical object on different orbits and at different speeds. Each object has a sound attached to it and when the object passes a certain point on the orbit, the sound is played. The Audiotarium Rotator fetches sounds from the Audiotarium sound repository based on the IDs of the sound objects. The user of the Audiotarium Rotator can change sounds of moving objects by copy-pasting ids of sounds from the Audiotarium web site to text fields of the Audiotarium Rotator. There are no instructions on how to use the Audiotarium Rotator, because the application is not a matter of great importance in the test use phase. And after all, it is all about playing.

## 4.4 About the Visual Identity and the Interaction Design

There is a definite need for a uniform visual identity for the Audiotarium. Colours black, red, and blue are fine but the UI needs to be designed completely again. The interaction design needs to be done from the scratch so that the UI would be more intuitive and key actions of the service would

be emphasised.

The nature of a sound object needs to be especially emphasised. At the moment, it is not very clearly stated that the sound object includes both the actual sound file and meta data related to the file.

One option would be to make a fancy version of the UI in Flash. Probably by making sliding panels and semi-transparent elements, the user experience could be made a little bit faster and a bit more intuitive which would again enhance the experience of the play. At the moment, the prototype is only a proof of concept so it does not have to be fast to use.

Test user feedback of the prototype seems to signal that the UI is what most people consider to be the service. Thus, the UI needs to represent the service in the way that makes the user experience fluent and amusing.

## **4.5 The Core System and the UI System**

The prototype does not actually implement the concept idea of having a lower level core functionality which would take care of the communication between the database and other components of the concept. At the moment the UI and the core system are bound together a little bit too tightly. UI updates require more and more hacking of the core system and other way around. This is not sustainable a arrangement.

If WordPress would be used as a UI platform for the final service, there would have to be a WordPress plugin which would integrate the Audiotarium functionality to WordPress logic. At the moment, WordPress is hacked to serve only as a user account manager and as a UI framework manager. This is not good solution in the long run. Expanding needs and issues with WordPress updates require that the system needs to be as modular as possible. WordPress is a very fast developing platform, so it would be best to follow practises which are native in the WordPress environment.

## **4.6 Hosting and Bandwidth Issues**

The ultimate solution regarding bandwidth would be a personal server. Web hosts have strict rules and they charge quite a lot for a bandwidth. Since the Audiotarium is a public domain project there might be a possibility to get some public funding for development. Mainly the bandwidth issues are money issues.

## **4.7 The Scalability**

The sound repository of the prototype can be separated from all of the other parts of the functionality. Practically, the prototype database could be used for the end product with only slight modifications. Almost everything else but the main database would have to be designed, built, and tested again. There are not very many test users and they all have been informed that the data they add to the Audiotarium will not be public and will not be there in the possible end product.

The prototype is not at all scalable. It is a proof of concept and the final version of the Audiotarium needs to be designed almost completely again. Still, the prototype provides a considerable quantity of invaluable data for the final design.

## 5. Results

The Audiotarium concept was defined and the prototype was built. The service prototype serves as a proof of concept that a public domain sound object repository can be built and it can be made simple and easy to use. It is possible to make this kind of service and it is possible to make it with a very low budget. The cost of maintenance rises with the number of users.

The money spent on this was 54 Euros per year for the domain and hosting of the service. The development was done on a computer which had been bought for another purpose. All of the software used, except Macromedia Flash, was free software. The Flash license was bought previously for other study reasons. The work was not paid work and was done to create a final thesis so there was no need to pay anyone a salary. The final result was that money needed for the project was 54 Euros.

The Audiotarium Rotator Sandbox Application is also a proof of concept that proves that it is possible to build applications which use the Audiotarium sounds. It shows that the user can interact with the service via the Audiotarium Sandbox Applications.

One important outcome is the realisation of the theory that prototyping is needed to develop web services. Problems which occur during the development process are not possible to predict and it very likely that some problems force the developer to see the project from different perspective. This happened in this development process as well. The prototype showed that there are some technical limitations which can be overcome only with money. The most important limitation was the bandwidth issue. The cost of the bandwidth limits dramatically the potential of web innovations. This knowledge is very important considering future development of the concept.

Preliminary user testing was partly successful and partly not. There were 11 user accounts created for friends of the developer. Google Analytics was installed to monitor the activity level of these preliminary test users. This was not meant to be any extensive usability testing session nor a formal technical evaluation of the service but it provided good feedback mechanism through

informal discussions. However test users were not very active during the 3 weeks period of testing. Preliminary user tests and informal discussions with test users give some kind of guidelines from where to begin to find the possible target group. Preliminary user tests were arranged mainly to find out if the technical solution on server side is working and to find out if some essential functionality is missing from the core of the Auditorium. So far it seems that the core functionality is there and it is working well enough.

## 6. Conclusion

To me, the writer of the work, it seems that the concept is worth developing, especially because there are two strong areas. The public domain ideology behind the concept makes the service exceptional and the Audiotarium API provides interesting possibilities for developers to expand the concept. During the process it became clear that technically the service is feasible to be implemented.

The Audiotarium Rotator is a simple and coherent example of a Sandbox Application. It gives some kind of idea of what can be done by using the Audiotarium API and how sound objects can be used straight from the Audiotarium sound object repository. The Audiotarium Rotator demonstrates a little bit of creative possibilities available to developers. Especially this application provides a good example of how to bring developers and users closer to each other. Developers have a possibility to build interactive applications using the Audiotarium API and ordinary users can explore the world or sound by interacting with Sandbox Applications. The Audiotarium Rotator was successful in showing how the Audiotarium can lower the threshold of the play.

There were many things to learn during the process. In the beginning of the project, I needed to find out what exists and what can be done in the context of the web. I also became familiar with some new tools like the WordPress blog engine and ActionScript 3. The most challenging part of the thesis was to find a proper balance between theory and practical work. Towards the end of the process, when writing took more time than coding I felt really exhausted and somehow annoyed by the fact that the prototype could have been better in many ways. The visual design, accessibility design, and the design of communal elements could have been done by reducing the amount of theoretical deliberation. On the other hand, I learned to explain ideas behind my work and managed to produce a reasonable amount of text which binds together the theory and the practical part.

However, the prototype is far from perfect. The work had to stay in some kind of frame so there was a need to leave certain parts out. Timewise it

would have been too much to build a fully functional prototype with a well designed UI. The next iteration of the prototype will be built using a more solid basis thus the focus of the project can be kept in the design of essential and more practical parts. Now that the scope of the concept is defined and the most fundamental technical barriers have mainly been overcome, the next version of the Audiotarium can be developed to be the first public release of the service. This way the prototype serves its purpose as a proof of concept.

Being partly practical and partly theoretical, master's thesis work this work taught me to see the difference between visual-only design and design which really aims to reach sustainable and well-reasoned results. When there is some kind of ideological foundation behind the design process, it is much easier to draw guidelines and make decisions. This is good, considering that the infinite quantity of possibilities quite often seems to block creativity. On the other hand, it is much more work to really anchor one's design on theory. Especially in the case of the Audiotarium, it was really difficult to stay true and honest to the idea of the play that was partly developed during the process. When all of the decisions are dependent on a set of rules, it forces one to give up many things which would be interesting to explore. This has to be done for the sake of continuity in following one's principles.

During the process of making the thesis I learned a lot about my own working skills and habits. I am a bit more confident now as a professional web designer because I know what I can do and what I can leave others to do. I have also realised that writing my thoughts coherently is not one of my best skills. The written part of this work had to go through many iterations to find its final form.

In many ways, the concept is not special at all. The idea was not to create something marvelous, fancy, shiny, or even new but to see some existing things in a new light. There are services which are quite similar to the Audiotarium, but they are not exactly the same. The main difference and the main advantage is that sound objects are free in the Audiotarium. This removes complexity from the process of using the sound objects. This can be demonstrated by using the example application. One can explore sound objects by playing with the Audiotarium Rotator. There is no need for the

developer or the user of the Audiotarium Rotator to spend time thinking about licenses or crediting or copyrights of sounds which are used within the scope of the Audiotarium Rotator.

The concept can be seen as belonging to the free culture movement. In the context of the Audiotarium, freedom means complete freedom of digital sound objects. A web repository of public domain sound objects is needed to maintain and develop possibilities to play with sound in the context of the web. Licensing systems do not support the play well enough since they require extra energy from the user of sound objects. The user needs to do lots of research about licenses in order to be able to use licensed sound objects. The Audiotarium provides an alternative for licensing systems. This lowers the threshold of starting to play with sounds. It can be seen as a situation where people can go straight into what they want to do without having to deal with restrictions, regulations and legal issues.

There are some ideas I have for the future development of the concept. One future vision related to the Audiotarium is the mobile use case. One can record audio with most of the new mobile phones, so it would be interesting to have a simple mobile application which could record sounds and offer the Audiotarium upload form for the user to upload the sound straight into the Audiotarium repository. The mobile device would know the location already so this would save some trouble from the user. The mobile situation is very different from working with a desktop computer or laptop computer. This area would require a completely new research project. It seems likely that this mobile research will not happen very soon in the context of the Audiotarium.

The most interesting area of the future development is the API. The Audiotarium API provides interesting views on sound and the web. There are so many possibilities with sound objects and the API that all of them cannot be predicted. On a general level, some ideas can be formulated.

For example, output channels of recordings done using the Audiotarium Sandbox Applications might provide very interesting ways to bring users and developers closer to each other. As an example, a public domain web radio which would play things done with Sandbox Applications could work through

the Audiotarium web service. This would probably inspire both the developer and the user to participate actively. On the Audiotarium web site there could also be some kind of an experimental corner for code level integration of the Audiotarium web site and some Sandbox Applications. For example, collaborative systems which would use the Audiotarium sound objects could be viewed through the Audiotarium web site. In a way, this would be a virtual club where users of certain Sandbox Application could be performing and users of the Audiotarium web site would be the audience. Also the API itself can and will be developed according to needs of developers.

The API can be developed thinking some special cases also. I have been discussing with people who might be interested in using the API and the trend is that people would like to have some special features designed for their purposes. This all is possible and easy to do since the structure of the core of the Audiotarium is technically quite simple. There are some plans to implement special tags which are available only to certain users of the API so that there would be a possibility to group sounds for some special purposes. These sounds would be available to anyone but they would also have special tags which are only editable by the user who has added the sound. This is interesting area of development and requires more research about grouping of sound objects. This kind of progress may be a little bit in contradiction with the ideological basis of the Audiotarium. All kinds of extra features for some user groups are exclusive features and therefore the system wouldn't be that open anymore. So this direction of the development needs to be examined very carefully.

Overall the work managed to achieve defined aspirations to develop a platform which uses means of the web and the idea of public domain to lower the threshold of the play within the context of the web. If people are interested in playing with sounds in the context of the web remains to be seen. Preliminary user tests suggest that there is some kind of interest but examples of what to do and how are needed to boost users' creativity.

To me, the creator of the work, the concept is fascinating and personally challenging and motivating. Work started by Musique Concrète pioneers must be continued and the Audiotarium is one effort to do that.

The work itself has been an experimental journey into the world of sound. My personal and professional opinion about the process is that the work has definitely been part of the play.

# Appendix

## A. The Database Structure of The Audiotarium Core Functionality

### audiotarium\_api\_key

	Field	Type	Collation	Attributes	Null	Default	Extra	Action							
<input type="checkbox"/>	id	int(11)			No		auto_increment								
<input type="checkbox"/>	api_key	varchar(32)	latin1_swedish_ci		No										
<input type="checkbox"/>	id_user	int(11)			No										
<input type="checkbox"/>	ip_create	varchar(15)	latin1_swedish_ci		No										
<input type="checkbox"/>	time_create	datetime			No										
<input type="checkbox"/>	title	text	latin1_swedish_ci		No										
<input type="checkbox"/>	url	text	latin1_swedish_ci		No										
<input type="checkbox"/>	description	text	latin1_swedish_ci		No										

### audiotarium\_log\_api

	Field	Type	Collation	Attributes	Null	Default	Extra	Action							
<input type="checkbox"/>	id	int(11)			No		auto_increment								
<input type="checkbox"/>	api_key	varchar(32)	latin1_swedish_ci		No										
<input type="checkbox"/>	api_action	text	latin1_swedish_ci		No										
<input type="checkbox"/>	time_action	datetime			No										
<input type="checkbox"/>	ip_action	text	latin1_swedish_ci		No										
<input type="checkbox"/>	url_action	text	latin1_swedish_ci		No										

### audiotarium\_sound\_info

	Field	Type	Collation	Attributes	Null	Default	Extra	Action							
<input type="checkbox"/>	id	int(11)			No		auto_increment								
<input type="checkbox"/>	id_public	varchar(32)	latin1_swedish_ci		No										
<input type="checkbox"/>	id_user	int(11)			No										
<input type="checkbox"/>	ip_upload	varchar(15)	latin1_swedish_ci		No										
<input type="checkbox"/>	time_upload	datetime			No										
<input type="checkbox"/>	title	text	latin1_swedish_ci		No										
<input type="checkbox"/>	description	text	latin1_swedish_ci		No										
<input type="checkbox"/>	filesize	int(11)			No										
<input type="checkbox"/>	bitrate	text	latin1_swedish_ci		No										

## audiotarium\_sound\_location

	Field	Type	Collation	Attributes	Null	Default	Extra	Action						
<input type="checkbox"/>	id	int(11)			No		auto_increment							
<input type="checkbox"/>	id_sound	int(11)			No									
<input type="checkbox"/>	id_user	int(11)			No									
<input type="checkbox"/>	lon	text	latin1_swedish_ci		No									
<input type="checkbox"/>	lat	text	latin1_swedish_ci		No									
<input type="checkbox"/>	time	datetime			No									

## audiotarium\_sound\_tags

	Field	Type	Collation	Attributes	Null	Default	Extra	Action						
<input type="checkbox"/>	id	int(11)			No		auto_increment							
<input type="checkbox"/>	id_sound	int(11)			No									
<input type="checkbox"/>	id_user	int(11)			No									
<input type="checkbox"/>	tag	text	latin1_swedish_ci		No									
<input type="checkbox"/>	time	datetime			No									

## B. The Code of the AudiotariumInterfaceAC3 Class

```

package
{
    import flash.display.MovieClip;
    import flash.display.Sprite;
    import flash.events.*;
    import flash.net.URLRequest;
    import flash.net.URLRequestMethod;
    import flash.net.URLLoader;
    import flash.xml.*;
    import flash.errors.*;
    import flash.media.*;

    public class AudiotariumInterfaceAC3 extends MovieClip
    {
        //-----
        //VARIABLES
        //-----

        //URL VARIABLES
        private var param_action:String = "action";
    }

```

```

private var value_action_value_get_sound:String      = "get_sound";
private var param_api_key:String                    = "api_key";
private var value_api_key:String                   = "null";
private var param_sound_id:String                  = "sound_id";
private var value_sound_id:String                  = "null";
private var url_api:String                         = "";
private var url_get_sound_body:String              = "";
private var url_get_sound:String                  = "";

//Data variables
public var data_as_object                          = new Object();

//Sound variables
private var sound_channel:SoundChannel;

//-----
//THE CONSTRUCTOR
//-----
public function AudiotariumInterfaceAC3 (url_api:String, api_key:String)
{
    //Don't change order of these!
    this.setUrlApi(url_api);
    this.setValueApiKey(api_key);
    this.setUrlGetSoundBody();
}

//-----
//SETTERS
//-----
private function setUrlApi(url_api:String):void
{
    this.url_api = url_api;
}
private function setValueApiKey(api_key:String):void
{
    this.value_api_key = api_key;
}
private function setUrlGetSoundBody():void
{
    this.url_get_sound_body      = this.url_api + "?" + this.param_action +
    "=" + this.value_action_value_get_sound + "&" + this.param_api_key +
    "=" + this.value_api_key + "&" + this.param_sound_id + "=";
}

```

```

}
private function setValueIdSound(id_sound:String):void
{
    this.value_sound_id    = id_sound;
}
private function getUrlGetSound():void
{
    this.url_get_sound = this.url_get_sound_body + this.value_sound_id;
}

//-----
//API Functionality
//-----
public function apiGetSound(id_sound:String)
{
    //Both required!
    this.setValueIdSound(id_sound);
    this.getUrlGetSound();
    this.readSoundXML();
}

private function readSoundXML()
{
    var XML_URL:String          = this.url_get_sound;
    var myXMLURL:URLRequest     = new URLRequest(XML_URL);
    var myLoader:URLLoader      = new URLLoader(myXMLURL);

    myLoader.addEventListener("complete", xmlLoaded);
    function xmlLoaded(event:Event):void
    {
        var myXML = new XML(myLoader.data);
        parseResultXML(myXML);
    }
}

public function parseResultXML(xml_result:XML):void
{
    var error:String = xml_result.elements("error");
    trace(xml_result.elements("error"));

    if(error == "null")
    {
        this.data_as_object.id_public    =

```

```

xml_result.elements().elements().elements("id_public");
        this.data_as_object.title           =
xml_result.elements().elements().elements("title");
        this.data_as_object.description     =
xml_result.elements().elements().elements("description");
        this.data_as_object.user_nick      =
xml_result.elements().elements().elements("user_nick");
        this.data_as_object.time_upload    =
xml_result.elements().elements().elements("time_upload");
        this.data_as_object.bitrate        =
xml_result.elements().elements().elements("bitrate");
        this.data_as_object.filesize       =
xml_result.elements().elements().elements("filesize");
        this.data_as_object.sound_url      =
xml_result.elements().elements().elements("sound_url");
        this.loadSoundFromAudiotarium();
    }
    else
    {
        this.data_as_object.sound = new Sound();
    }
}

function loadSoundFromAudiotarium():void
{
    var req:URLRequest    = new URLRequest(this.data_as_object.sound_url);
    var s1:Sound          = new Sound(req);
    s1.addEventListener(Event.COMPLETE, onSoundLoaded);
}

function onSoundLoaded(event:Event):void
{
    this.data_as_object.sound    = (event.target as Sound);
}

function playSound():void
{
    var transform:SoundTransform    = new SoundTransform(0.5, 0);
    this.sound_channel              = this.data_as_object.sound.play();
    this.sound_channel.soundTransform    = transform;
}

```

```
}  
}
```

## C. Examples of the Audiotarium API XML Results

### Example 1: Result with one sound entity

```
<?xml version="1.0" encoding="UTF-8"?>  
<audiotarium>  
  <error>  
    null  
  </error>  
  <error_type>  
    null  
  </error_type>  
  <data>  
    <sound>  
      <id_public>26f54558e5fda9caf66399aa4396742c</id_public>  
      <sound_url>http://www.audiotarium.net/proto/API/sound.php?s=26f54558e5fda9caf66399aa4396  
742c</sound_url>  
      <user_nick>admin</user_nick>  
      <time_upload>2008-09-19 23:50:28</time_upload>  
      <title>Example2</title>  
      <description>This sound is also for examples on the API URLs & Functions -page. The sound is  
made from some old bass sound by distorting it and adding some room.</description>  
      <filesize>30770</filesize>  
      <bitrate>192</bitrate>  
      <tags>  
        <tag>test</tag>  
        <tag>example</tag>  
        <tag>synth</tag>  
        <tag>room</tag>  
        <tag>distortion</tag>  
        <tag>bass</tag>  
      </tags>  
      <locations>  
        <location>  
          <lon>2655499.463665379</lon>  
          <lat>8734229.99677515</lat>  
          <time>2008-09-19 23:50:28</time>  
        </location>  
      </locations>
```

```
</sound>
</data>
</audiotarium>
```

## Example 2: Result with many sound entities

```
<?xml version="1.0" encoding="UTF-8"?>
<audiotarium>
  <error>
    null
  </error>
  <error_type>
    null
  </error_type>
  <data>
    <sound>
      <id_public>26f54558e5fda9caf66399aa4396742c</id_public>

      <sound_url>http://www.audiotarium.net/proto/API/sound.php?s=26f54558e5fda9caf66399aa4396
742c</sound_url>
      <user_nick>admin</user_nick>
      <time_upload>2008-09-19 23:50:28</time_upload>
      <title>Example2</title>
      <description>This sound is also for examples on the API URLs & Functions -page. The sound is
made from some old bass sound by distorting it and adding some room.</description>
      <filesize>30770</filesize>
      <bitrate>192</bitrate>
      <tags>
        <tag>test</tag>
        <tag>example</tag>
        <tag>synth</tag>
        <tag>room</tag>
        <tag>distortion</tag>
        <tag>bass</tag>
      </tags>
      <locations>
        <location>
          <lon>2655499.463665379</lon>
          <lat>8734229.99677515</lat>
          <time>2008-09-19 23:50:28</time>
        </location>
      </locations>
```

```

</sound>
<sound>
  <id_public>61fa8fcfa7c57668177e1b986451399</id_public>
  <sound_url>http://www.audiotarium.net/proto/API/sound.php?s=61fa8fcfa7c57668177e1b986451
399</sound_url>
  <user_nick>admin</user_nick>
  <time_upload>2008-09-19 23:47:14</time_upload>
  <title>Example1</title>
  <description>This sound is uploaded for examples on API URLs & Functions -page. It is made of
cymbal recording by removing the attack.</description>
  <filesize>154622</filesize>
  <bitrate>192</bitrate>
  <tags>
    <tag>example</tag>
    <tag>test</tag>
    <tag>shure</tag>
    <tag>tuomo</tag>
    <tag>ambience</tag>
    <tag>drone</tag>
    <tag>cymbal</tag>
  </tags>
  <locations>
    <location>
      <lon>2655346.589608825</lon>
      <lat>8734349.42963171</lat>
      <time>2008-09-19 23:47:14</time>
    </location>
  </locations>
</sound>
</data>
</audiotarium>

```

### Example 3: Result with error

```

<?xml version="1.0" encoding="UTF-8"?>
<audiotarium>
  <error>
    Error: Action doesn't exist
  </error>
  <error_type>
    action
  </error_type>

```

```
<data>  
null  
</data>  
</audiotarium>
```

## **D. Textual Version of the Creative Commons Public Domain Dedication**

The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of the country from which the work is published, or (b) hereby dedicates whatever copyright the dedicators holds in the work of authorship identified below (the "Work") to the public domain. A certifier, moreover, dedicates any copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognises that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors. Dedicator intends this dedication to be an overt act of relinquishment in perpetuity of all present and future rights under copyright law, whether vested or contingent, in the Work. Dedicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.

Dedicator recognises that, once placed in the public domain, the Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.

## **E. The Complete List of the Audiotarium API Functions**

## ***get\_sound***

Returns sound data by public ID of the sound.

### **PARAMETERS**

api_key [obligatory]	:	Needed for all the functions.
sound_id [obligatory]	:	Public ID of the sound requested.

## ***get\_sounds\_by\_tag\_search***

Returns data of sounds which match the tag search. Maximum amount of returned sounds is 50, default is 10.

### **PARAMETERS**

api_key [obligatory]	:	Needed for all the functions.
tag [obligatory]	:	Returns data of sounds with tag given in request.
order	:	Possible values are title_asc, title_desc, time_asc, time_desc. Order is not obligatory, the default order is time_desc.
amount	:	Possible values are values from 1 to 50. Amount is not obligatory, the default amount is 10 (or less if there are not that many sounds with the tag given).

## ***get\_sounds\_by\_title\_search***

Returns data of sounds which match the title search. Maximum amount of returned sounds is 50, default is 10.

### **PARAMETERS**

api_key [obligatory]	:	Needed for all the functions.
search [obligatory]	:	Returns data of sounds with the search text in their title.
order	:	Possible values are title_asc, title_desc, time_asc, time_desc. Order is not obligatory, the default order is time_desc.

amount : Possible values are values from 1 to 50. Amount is not obligatory, the default amount is 10 (or less if there are not that many sounds with the string given).

### ***get\_sounds\_by\_description\_search***

Returns data of sounds which match the description search. Maximum amount of returned sounds is 50, default is 10.

#### **PARAMETERS**

api\_key [obligatory] : Needed for all the functions.

search [obligatory] : Returns data of sounds with the search text in their description.

order : Possible values are title\_asc, title\_desc, time\_asc, time\_desc. Order is not obligatory, the default order is time\_desc.

amount : Possible values are values from 1 to 50. Amount is not obligatory, the default amount is 10 (or less if there are not that many sounds with the string given).

# List of Acronyms

## **API = Application Programming Interface**

*A set of functions that a service provides to enable communication between computer programs.*

Bray, Mike (Lockheed-Martin Ground Systems) 1997, *Application Programming Interface*, Software Engineering Institute Pittsburgh, viewed 2 October 2008, <<http://www.sei.cmu.edu/str/descriptions/api.html>>

## **HTML = HyperText Markup Language**

*The markup language used to create Web pages.*

W3C® (MIT, ERCIM, Keio) 1999, *HTML 4.01 Specification*, W3C®, viewed 2 October 2008, <<http://www.w3.org/TR/REC-html40/>>

## **HTTP = HyperText Transfer Protocol**

*A protocol for the transfer of information on the Internet.*

W3C® (MIT, ERCIM, Keio) 2003, *HTTP - Hypertext Transfer Protocol*, W3C®, viewed 2 October 2008, <<http://www.w3.org/Protocols/>>

## **JSON = JavaScript Object Notation**

*A lightweight data interchange format*

json.org 2008, json.org, viewed 2 October 2008, <<http://www.json.org/>>

## **PHP = PHP: Hypertext Processor**

*A recursive acronym. An open source, server-side, HTML-embedded scripting language used to create dynamic Web pages.*

The PHP Group 2008, The PHP Group, viewed 2 October 2008, <<http://php.net/>>

## **RSS = Really Simple Syndication and Rich Site Summary**

*A group of Web feed formats used to publish information.*

Winer, Dave 2003, *RSS 2.0 Specification, RSS 2.0 at Harvard Law*, Berkman Center, viewed 2 October 2008, <<http://cyber.law.harvard.edu/rss/rss.html>>

## **XML = eXtensible Markup Language**

*A general specification for markup languages*

3C XML Core Working Group (WG) 2006, *Extensible Markup Language (XML) 1.0 (Fourth Edition)*, W3C® (MIT, ERCIM, Keio), viewed 2 October 2008, <<http://www.w3.org/TR/REC-xml/>>

# Bibliography

## Books

Brand, Stewart 1987, *The Media Lab : inventing the future at MIT*, Viking Penguin, New York, N.Y., p.202

Gibson, J.J. 1979, *The Ecological Approach to Visual Perception*, Houghton Mifflin, Boston. (Currently published by Lawrence Erlbaum, Hillsdale, NJ.)

Keller, J.Frederick, Gettys, W.Edward, Skove, Malcolm J. 1993, *Physics Classical and Modern Second Edition*, McGraw-Hill Inc, Chapter 33 Sound

Lessig Lawrence 2004, *Free Culture*, The Penguin Press, p.282, <<http://www.free-culture.cc/freeculture.pdf>>

DiBona Chris, Ockman Sam Stallman, Richard M. 1999, *The GNU Operating System and the Free Software Movement* -article as cited in *Open Sources : Voices from the Open Source Revolution*, O'Reilly

## World Wide Web

### Online Videos

Harrison, Nate 2004, *Can I Get An Amen?*, online video, viewed 2 October 2008, <[http://www.nkhstudio.com/pages/amen\\_mp4.html](http://www.nkhstudio.com/pages/amen_mp4.html)>

### Web documents

Cambridge Encyclopedia Vol. 59 1994, *Pierre Schaeffer*, Cambridge Encyclopedia, U.S. University Directory, viewed 2 October 2008, <<http://encyclopedia.stateuniversity.com/pages/17450/Pierre-Schaeffer.html>>

Coar Ken 2006, *The Open Source Definition*, The Open Source Initiative, viewed 10 October 2008, <<http://www.opensource.org/docs/osd/>>

Creative Commons 2008, *About – Creative Commons*, Creative Commons San Francisco California United States, viewed 2 October 2008, <<http://creativecommons.org/about/>>

Creative Commons 2008, *Public Domain Dedication*, Creative Commons San Francisco California United States, viewed 2 October 2008, <<http://creativecommons.org/licenses/publicdomain/>>

Norman, Donald A. 2004, *Affordance, Conventions and Design (Part 2)*, ORIGINALLY PUBLISHED AS AFFORDANCE, CONVENTIONS, AND DESIGN IN THE MAY 1999 ISSUE OF INTERACTIONS, PP 38-43, viewed 11 October 2008, <[http://www.jnd.org/dn.mss/affordance\\_conv.html](http://www.jnd.org/dn.mss/affordance_conv.html)>

Free Software Foundation Inc 2008, *The Free Software Definition*, viewed 2 October 2008, <<http://www.gnu.org/philosophy/free-sw.html>>

Ballora M. 2008, INART 55 (Department of Integrative Arts), *History of Electroacoustic Music Musique Concrète*, The Pennsylvania State University, viewed 2 October 2008, <<http://www.music.psu.edu/Faculty%20Pages/Ballora/INART55/concrete.html>>

*Public Domain Definition* 2004, The Linux Information Project, viewed 2 October 2008, <<http://www.linfo.org/publicdomain.html>>

Ritchie, Dennis M. (Bell Labs/Lucent Technologies) 2003, *The Development of the C Language*, viewed 2 October 2008, <<http://plan9.bell-labs.com/who/dmr/chist.html>>

Stallman Richard 2007, *Why "Open Source" misses the point of Free Software*, Free Software Foundation Inc., <<http://www.gnu.org/philosophy/open-source-misses-the-point.html>>

## **Weblogs**

Storani, Maurizio 2008, *Mashup (hybrid web application): new generation of Web-based applications*, viewed 2 October 2008, <<http://mauriziorstori.wordpress.com/2008/09/18/mashup-hybrid-web-application-new-generation-of-web-based-applications/>>

## **Web sites**

Adobe Systems Incorporated 2008, *Adobe Systems Incorporated*, viewed 2 October 2008, <<http://www.adobe.com/devnet/flash/>>

Amazon.com, Inc 2008, Amazon.com, Inc, viewed 2 October 2008, <<http://www.amazon.com/>>

Cycling '74 2008, *Welcome to Max 5: Max for the next 20 years*, Cycling '74, viewed 2 October 2008, <<http://www.cycling74.com/products/max5/>>

Facebook 2008, Facebook, viewed 2 October 2008, <<http://www.new.facebook.com/>>

The Freesound Project 2008, The Freesound Project, viewed 2 October 2008, <<http://www.freesound.org/>>

MySQL AB 2008, Sun Microsystems, Inc, viewed 2 October 2008, <<http://www.mysql.com/>>

Norman, Donald A. 2004, jnd.org, viewed 2 October 2008, <<http://www.jnd.org/>>

Pure Data Portal 2008, PD-community, viewed 2 October 2008, <<http://puredata.info/>>

openlayers.org 2008, openlayers.org, viewed 2 October 2008, <<http://openlayers.org/>>

openstreetmap.org 2008, openstreetmap.org, viewed 2 October 2008, <<http://www.openstreetmap.org/>>

wordpress.org 2008, wordpress.org, viewed 2 October 2008, <<http://wordpress.org/>>

wordpress.org 2008, WordPress > About, wordpress.org, viewed 2 October 2008, <<http://wordpress.org/about/>>

Yahoo! Inc. 2008, Yahoo! Inc., viewed 2 October 2008, <<http://www.flickr.com/>>

YouTube, LLC, YouTube, viewed 2 October 2008, <<http://www.youtube.com/>>