Web 2.0 for IPTV

SIMON LEUNG
and
JOHAN PETTERSSON

KTH Information and
Communication Technology

Master of Science Thesis
Stockholm, Sweden 2008

COS/CCS 2008-29

Web 2.0 for IPTV

Master of Science Thesis

Simon Leung
sile02@kth.se

Johan Pettersson
johpett@kth.se

Academic supervisor and examiner

Professor Gerald Q. Maguire Jr, KTH

Company supervisor

Fredrik Palmcrantz, Accedo Broadband

19 December 2008

Abstract

This master's thesis project concerns implementing and evaluating Web 2.0 services for IPTV. This thesis project was carried out at Accedo Broadband during the period August to December 2008. Accedo Broadband is a provider of interactive applications and on-demand content for IPTV. The report will analyze, specify, create, and evaluate a solution for adding Web 2.0 functionality from the most popular sites as IPTV content. We will evaluate several different web services APIs and examine how suitable they are for implementation in the context of IPTV. An IPTV set-top box has limited CPU processing power, memory capacity, screen resolution is generally limited to 720 by 576 pixels, and only a limited set of input devices and interfaces are available. The IPTV set-top box receives digitally encoded audio and video content through IP and renders this content so it is viewable on an analogue TV and the audio is output via speakers. The goal of this project is to reduce the amount of integration work and configuration necessary for creating Web 2.0 IPTV applications by creating a general framework for Web 2.0 applications. We will present and evaluate the most common web programming technologies and see which are most suitable for our purposes. We have developed four different interactive demonstration services to illustrate the usage of an IPTV platform.

Sammanfattning

Denna rapport är ett examensarbete beträffande implementering och evaluering av Webb 2.0 tjänster för IPTV. Examensarbetet har utförts på företaget Accedo Broadband under perioden augusti till december 2008. Accedo Broadband är ett företag som utvecklar interaktiva applikationer och on-demand tjänster för IPTV. Denna rapport handlar om att analysera, specificera, skapa och utvärdera en lösning för att kunna implementera en Webb 2.0 funktionalitet från de mest populära tjänsterna till IPTV. Vi utvärderar olika webbtjänsters API:er och utforska hur passande dom är. En IPTV set-up box har begränsad CPU, minne, skärmstorlek på 720×576 pixlar och en begränsad uppsättning av inmatningenheter & gränssnitt. IPTV set-top boxen tar emot digitalt kodade ljud- och videoströmmar genom IP och konverterar dem så att det går att se på en analog TV. Målet med detta projekt är reducera arbetet med integrationen och nödvändiga konfigurationer för IPTV för Webb 2.0 applikationer. Vi kommer att presentera och utvärdera den webbteknik som är vanligast förekommande och ser vilka som passar våra syften bäst. Vi har utvecklat fyra olika demonstrations applikationer för att illustrera hur en IPTV plattform fungerar.

Acknowledgements

We want to sincerely thank our examiner and supervisor Professor Gerald Q. Maguire Jr. for all the time and effort he devoted to correcting and commenting this thesis, and for giving us plenty of valuable suggestions of interesting articles to read in the scope of this thesis.

We do also want to thank our company supervisor Fredrik Palmcrantz for his important feedback and all of the people at Accedo Broadband for assisting us in different ways, and for sharing their extensive experience and knowledge regarding developing IPTV applications.

Table of Content

List of figures

List of tables

List of code examples

List of abbreviations

AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
CODEC	Compression/Decompression algorithm
CSS	Cascading Style Sheets
DHCP	Dynamic Host Configuration Protocol
DHTML	Dynamic HyperText Markup Language
DMIPS	Dhrystone Million Instructions per Second
DOM	Document Object Model
DRAM	Dynamic Random Access Memory
GFDL	GNU Free Document License
GHz	Giga Hertz
GNU	Gnu's Not Unix
GUI	Graphical User Interface
HDMI	High Definition Multimedia Interface
HTML	HyperText Markup Language
HTTP	HyperText Transport Protocol
IGMP	Internet Group Management Protocol
IPTV	Internet Protocol Television
IR	Infrared
IRC	Internet Relay Chat
JSON	JavaScript Object Notation
MB	Megabyte
Mb	Megabit
Mbps	Megabit per second
PAL	Phase Altering Line
PC	Personal Computer
PDA	Personal Digital Assistant
PHP	PHP: HyperText Preprocessor
PKI	Public Key Infrastructure
RAM	Random Access Memory
REST	Representational State Transfer
RIA	Rich Internet Applications
RPC	Remote Procedure Call
RSS	Really Simple Syndication (v 2.0)
RTSP	Real Time Streaming Protocol
SDK	Software Development Kit
SIF	Source Input Format
SOAP	Simple Object Access Protocol (until v 1.2)
SGML	Standard Generalized Markup Language
SQL	Structured Query Language
SSL	Secure Sockets Layer
STB	Set-top box
SVG	Scalable Vector Graphic
T9	Text on 9 keys
TCP	Transport Control Protocol
URL	Uniform Resource Locator

VoIP	Voice over Internet Protocol
W3C	WWW Consortium
WAP	Wireless Application Protocol
WDDX	Web Distributed Data eXchange
WHATWG	Web Hypertext Application Technology Working Group
WWW	World Wide Web
XHTML	Extensible HyperText Markup Language
XML	Extensible Markup Language
XSL	Extensible Stylesheet Language
XSLT	XSL Transformations
YAML	YAML Ain't a Markup Language
YUI	Yahoo! User Interface Library

1. Introduction

Internet access is rapidly becoming available to everyone worldwide and the TV industry is advancing at a fast pace. A technology for TV distribution via IP technology, so called IPTV is developing as a market. This new TV distribution technology will probably continue to grow, the broadband penetration has reached 200 million households by the end of 2005, and is expected to reach more than 400 million households by 2010 [1]. However, IPTV devices need not only be used for TV watching, but can also enable consumers to interact with both TV content and with other types of network services. There are potentially a lot of applications that can utilize the installed IPTV base, for example Web 2.0 services. Over the last several years Internet users have migrated from regular computers to PDAs, mobile phones, smart phones, and now TV connected devices. The web has grown from uni-directional consumption to bi-directional participation with the emergence of YouTube, Flickr, MySpace, Facebook, and others (note that each of these services will be described in detail later).

This thesis will analyze the most popular Web 2.0 services, compare them, and examine which Web 2.0 functionalities are suitable for IPTV. Our first task is to develop four interactive services and try these services on two different set-top boxes (STB) to see what limits the STBs have and how they differ from a regular computer. The STB will communicate with an Apache web-server with PHP and MySQL support. The second task is to examine if we can develop a general framework to reduce the integration and configuration work necessary when creating future Web 2.0 applications for a variety of STBs.

1.1. IPTV

The term IPTV was first used in 1995 by Judith Estrin and Bill Carrico when they built an Internet video product named IP/TV. IPTV is digital TV distributed over an IP-network instead of a traditional cable TV network. IPTV is usually delivered by a service provider using their own network to an IPTV-box. The provider breaks the video stream, generally encoded in MPEG-2 or H.264/MPEG-4, into chunks and encapsulates these to IP packets and sends this content through the network to the IPTV (set-top) box. When the IPTV-box receives the packets it re-assembles the audio-video stream and outputs the content via the display and speakers connected to the IPTV box. The advantage the provider has by using their own network is that they have end-to-end control of the audio-video stream so that the packets can be delivered with low delay and little fragmentation. They can also prioritize this over other traffic and guarantee sufficient bandwidth for their traffic at all times.[2]

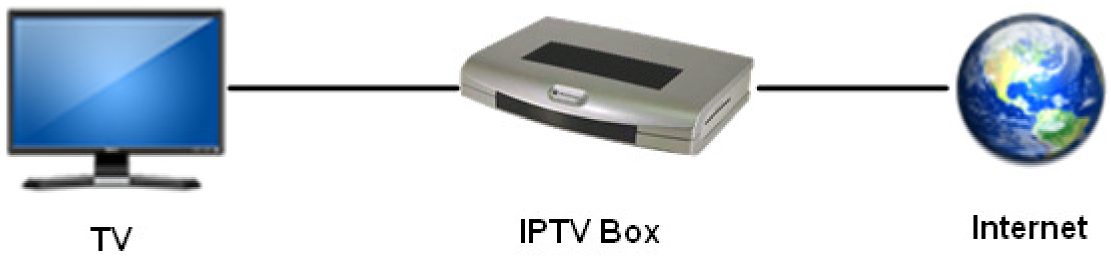


Figure 1-1: The basic components of IPTV

There are also service providers who use other network operators to route their IPTV content, they can suffer due to having a lower priority than the network operator’s own IPTV content, this is often referred to as a violation of “network neutrality” (i.e., traffic of different providers are treated differently). For the purposes of this thesis we will focus on IPTV in the context of network operator provided content (even if the operator may use data provided by others, the network operator is assumed to be operating the web server which will be used to source content to the subscriber), but we will mention other alternatives when appropriate.

Another difference between traditional TV and IPTV is that rather than sending hundreds of channels out to an IPTV subscriber - only the actual content that this user is interested in is sent. Therefore, a STB does not tune to a channel; instead the box selects which media stream it wants by using the Internet Group Management Protocol (IGMP) v2 to join the appropriate multicast group. When the subscriber logically changes to another channel, the provider receives an IGMP join request and checks if the subscriber is authorized to access this channel and if so, then this STB will be added to the channel's distribution list causing the appropriate router to send a copy of this stream to the subscriber. In this way, only the current channel is sent to the subscriber and unnecessary traffic is avoided. There are still some reasons for sending traffic associated with multiple channels - in order to decrease time when channel surfing or in the case of "picture in picture" service where the user can view multiple streams at one time, but only one is at full resolution and the other(s) is(are) shown at reduced resolution in a window.

In the past, the growth of the IPTV market has been restricted by the low broadband penetration, but today many households have broadband connectivity and the number is still increasing. Most western countries and countries with developed economies have IPTV deployments and the world's leading markets for IPTV are France, South Korea, Hong Kong, Japan, Italy, Spain, Belgium, China, and Switzerland. IPTV services have also launched in Scandinavia and Iceland. [3] According to Motorola’s EMEA Market Manager, Steve Farmer, Motorola sold as many STBs the first seven months of 2008 as they did the seven years before that, meaning the IPTV penetration are evolving in an exponential rate.[4]

1.2. Web 2.0

Web 2.0 as a term became very popular after it was used by Tim O´Reilly in 2004 to describe the more interacting web services that became popular at the time. This new type of website focuses more on user-to-user interaction, rather than the older sites that often only offered a service provider-to-user functionality. [5]

The core of Web 2.0 is that active users contribute to the site by sharing information with other users. Such information can be of almost any form, ranging from plain text to high quality video or desktop-like software. Most Web 2.0 sites profit from their large number of users by displaying commercials and other content based on the specific user’s habits (i.e., earlier behavior) in order to increase the chances that the user actually views the commercials. It is hoped that by making the commercials very relevant to the user that there is a high probability that the user will find the offer attractive.

There are many examples of Web 2.0 sites. Among the most visited pages on the Internet are Facebook's community (see section 2.2.4), Flickr's photo sharing service (see section

2.2.1), Youtube's video sharing service (see section 2.2.6), and Wikipedia - the free Internet Encyclopedia (see section 2.2.5). Another very popular part of Web 2.0 is the new public version of a diary – the blog – a simple page were you can express your thoughts about the latest happenings or other things which you find interesting.

In the 1990s the computer was the platform from which the user sent emails, edited photos, and chatted with other users. In Web 2.0 the web itself is considered to be the platform, as you can now store gigabytes of email at Google's GMail and chat with your friends directly at Facebook. These technologies are referred to as "transparent", meaning that it is possible to use your cellphone or PDA when you would like to send an email or interact with someone in other ways, thus a desktop computer is not necessarily needed any more. However, it is not simply the PC that is no longer needed for interaction; today with Skype, and other Voice over IP (VoIP) applications, users no longer need their traditional phones. "Podcasted" audio files from Apple's iTunes and others are growing in popularity as many people who previously listened to radio shows in the AM/FM radio band - now listen to a Podcast when and where they want. Applications such as Joost provide us with free television over the Internet.

Many Web 2.0 services offer their users the latest updates through RSS or other web feeds. This automated distribution is often referred to as "syndication" and it makes it possible for websites to dynamically show content published by other sites.

Some of the more advanced features associated with Web 2.0 are "mashups". These are based upon services that provide open APIs for other developers to use at their sites. One of the more famous services of this kind is Google Maps used by, amongst many others, mapmyrun.com to show your favorite running routes.

"Rich Internet Applications" describe web services that provide functionality normally found in desktop applications. Google has been very successful in this area with GMail and their word processing application Google Docs.

Sites were the user tags the content are called "folksonomy", for example flickr and the social bookmarking service del.icio.us base their search functions on collaborative tagging. Thus they are to be able to show/search data based on the **content** and not just the name.

2. Background

To understand this report you need to have a basic understanding of some of the most used techniques for Web 2.0 development. We also introduce some of the most popular services on the web and discuss advantages and disadvantages of implementing them for IPTV. We also describe the details of a representative set-top box (STB) because its hardware and software limitations are important to understand - as knowing what it supports makes the process of implementation easier. Also understanding the STB gives some guidance as to what applications will easily map to this platform.

2.1. Underlying technologies and techniques

2.1.1. HTML

The HyperText Markup Language [A] was developed in the 1980s by a group of physicists at CERN. These physicists did not want to distribute their work as a fragmented collection of pictures, sounds, and text; thus they developed HTML to bind all of the different elements of contents into a single document. HTML is a markup language for web pages. HTML uses tags surrounded with angle brackets "<...>" to describe what the interpretation of the tagged object is. This enables the author of the document to include a link, heading, list, table, etc. HTML was intended to represent the **logical structure** of a textual document, following the introduction of graphical interfaces (such as Netscape) there was a demand for web pages that were visually attractive (or interesting), and so the "font" tag was introduced to set the color, size, and font to be used for a given string. Today the visual presentation is controlled by Cascading Style Sheets, while HTML is used for the structure of a document. [6]

HTML evolved through efforts coordinated by W3C until version 4.01, but since year 2000 - W3C recommends the usage of Extensible HyperText Markup Language (XHTML) instead of HTML. XHTML is a stricter and cleaner version of HTML. XHTML does not allow a lot of the elements used in HTML, is strictly hierarchical, and is only supposed to be used for structure and content - the design should be handled by CSS. A common way in HTML to structure a site was to use tables, which are not allowed in XHTML. Instead the web designer uses div elements and implements styles through CSS. While HTML was based on SGML, XHTML is an XML application and has (amongst other features) inherited the namespaces functionality from XML. A group called WHATWG (Web Hypertext Application Technology Working Group) continues to develop HTML version 5 due their dissatisfaction with XHTML, that they believe is too much like XML and not optimal for developing web-forums or web-shops.

Dynamic HyperText Markup Language is a phrase that describes a collection of programming technologies - including HTML, JavaScript, DOM, and CSS - used to develop animated and interactive web-sites. The term, but not the technologies, is no longer used following the millennium shift, due the "discovery" of Ajax (see section 2.1.8) by a large group of developers.

Code Example 2-1: HTML

```html
<html>
	<head>
		<title>My website</title>
	</head>
	<body> <!-- This is the things that will appear on the website -->
		<p>Hello World!</p>
	</body>
</html>
```

2.1.2. CSS

CSS [B] is a stylesheet language used to control the presentation of a document. It is used in conjunction with HTML, XHTML, and SVG. CSS defines the colors, fonts, positions, and other visual layout for a document. CSS was designed to be separate from the content document, for better accessibility and control of the layout. CSS has a simple syntax that consists of a list of rules; with different rules applied to specific text blocks in a document. An important feature of CSS is absolute positioning enabling an author to have full control of the placement of all elements , i.e., text, pictures, tables, etc. [7]

Code Example 2-2: CSS

```css
body{
	background: #ff9911; /* set bgcolor to orange */
}
```

2.1.3. DOM

The Document Object Model (DOM) [C] is an interface, neutral to both platform and language that allows programs and scripts to dynamically access and update a web page. It was developed by W3C in the mid-1990s and is ordered in different layers according to which techniques are supported, the higher the level - the more modern and advanced the techniques. Since 2005, W3C DOM is well-supported by most browsers that are JavaScript-enabled. [8] [9]

2.1.4. JavaScript

JavaScript [D] was developed by Netscape and was influenced by many languages, but the script looks like Java to make it easy for non-programmers. JavaScript was initially named LiveScript, but the name was changed to attract Java programmers. JavaScript runs mostly in web browsers on the client-side.[10] When the browser loads a web page containing JavaScript, this JavaScript can dynamically change the layout by listening for event triggers from the mouse, keyboard, window, etc. In many cases the author of a web page wants to change the contents of a page without requiring that it be reloaded each time; this can be done by using the built-in XMLHttpRequest function. This function provides an API for making requests to a server (this technique is also called AJAX). For example, when you are typing in a search box the application can make suggestions of words after every key input. The standardized version of JavaScript is called ECMAscript.

Code Example2-3: JavaScript

```
<script type=”text/javascript”>
     //prints the text “Hello World!” on the screen
     document.write('Hello World!');
</script>
```

2.1.5. XML

XML [E] was specified in 1996 as a W3C recommendation for programmers to group data in a common structure. The interface is almost the same as for HTML, but in XML the programmer must define his or her own tags. These tags still have to follow a certain set of rules, for example: have one closing tag for every opening one. XML is only used for the transport of data. In order to publish the content of an XML-file you have to use HTML, CSS, or some programming language. [11] [12]

Code Example 2-4: XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<thesis>
     <!-- the tags we found necessary -->
     <area>IPTV</area>
     <persons category = “supervisor”>
               <name>Professor Gerald Q. Maguire Jr.</name>
     </persons>
</thesis>
```

2.1.6. JSON

JSON [F] is one of the most popular formats used for interchanging data. For example, JSON is used by Blogger (see section 2.2.2) to send a feed with their latest blogs, between different sites or services. JSON is based on a subset of JavaScript, but is considered language independent, both parsers and generators are available in numerous popular programming languages. The most desirable feature of JSON is that the format is easy to read and write for humans, as well as easy to generate and parse for software. JSON can represent the primitive types: string, number, Boolean, and null. It can also represent two structured types: objects and arrays. An object is an *unordered* collection of zero or more name-value pairs and an array as an *ordered* sequence of zero or more values. There are two ways of fetching information using a JSON-feed, the easy alternative is to use JavaScript’s eval() function, which interprets the JSON text to JavaScript code and executes it. There is of course a significant security risk when executing other people’s scripts on your own site. Therefore, unless you can exclude the possibility of harmful code in the received feed, it is better to use a JSON parser to transform the text into another form more suitable for your application. [13] [14]

Code Example 2-5: JSON

```
[{
    "Precision": "zip",
    "Latitude": 37.7668,
    "Longitude": -122.3959,
},
{
    "Precision": "zip",
    "Latitude": 37.371991,
    "Longitude": -122.026020,
}]
```

A JSON array containing two objects separated with a comma, containing three elements with locations values.

2.1.7. RPC

JSON-RPC [G] is a simple protocol for bi-directional communication between peers. It is peer-to-peer communication, as it is bi-directional and both calls and notifications can be sent in either direction. Because this is an asynchronous protocol, it is possible for multiple calls to a peer to be answered out of order. JSON-RPC can use either TCP/IP sockets or polling through HTTP for transport. The socket approach is encouraged by the developers of a JSON-RPC. JSON-RPC was developed in March 2004 as a lightweight alternative to XML-RPC for client-server communication on the SVG irc-channel. [15]

XML-RPC [H] was considered a lightweight protocol when developed by Microsoft and UserLand Software in 1998 with only a few data-types defined [16]. XML-RPC uses HTTP as the transport protocol and XML for encoding. It is considered easy to use and was simple predecessor of SOAP.

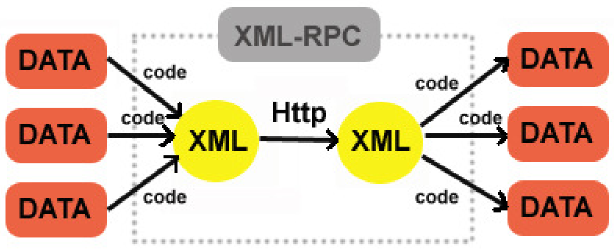


Figure 2-1: Describing the XML-RPC model

Thrift RPC [I] is a framework developed and used by Facebook for communication across programming languages. According to Facebook Thrift allows developers to define data-types and service interfaces in a single language-neutral file and generate all the code necessary to build RPC clients and servers. Facebook's developers considered other RPC mechanisms as either too heavy or too lightweight for their purposes. They have learned that the loss of performance by having an extra layer of software is less important than optimizing developer's productivity, thus they employ Thrift in applications for searching, logging, ads, and more. [17]

2.1.8. AJAX

The term AJAX (Asynchronous JavaScript and XML) [J] was coined 2005 when a group of web development technologies XHTML, CSS, JavaScript, DOM, and XML were used together. These web technologies together with the XMLHttpRequest object made it possible for a web application to retrieve data from a server *asynchronously* in the background without reloading the page. This new capability made it possible to for existing websites to become Web 2.0 sites, thus giving their website a new look and greater appeal. There are several alternatives to implement the AJAX functionality; instead of JavaScript developers can use VBScript, IFrames instead of XMLHttpRequest, and JSON instead of XML as data exchange format.

The advantages of using AJAX are a web application can request only the required data that needs to be updated; this saves bandwidth, gives a faster response, and results in a faster reload time. Disadvantages of using AJAX are dynamic web pages are hard to bookmark in a particular state, web browsers that do not support AJAX or the XMLHttpRequest object can not use this dynamic functionality (examples of such platforms are some mobile phones, PDAs, and other thin clients). [18] [19]

Code Example 2-6: AJAX (for Mozilla browsers)

```
<script type = "text/javascript">
     function ajaxFunction(){
          var xmlHttp;
          try{
               //Does not work for Internet Explorer...
               xmlHttp = new XMLHttpRequest();
          }
          catch(e){
               alert("IE uses ActiveX for AJAX...”);
               return false;
          }
     }
</script>
```

2.1.9. PHP

PHP [K] was created by Rasmus Lerdorf in 1995 using Perl and C. In the year 1998, PHP 2.2.0 was released, in 2000 version 4 was released with object-oriented programming support, and version 5 was released 2004 offering better object support. The current main implementation of PHP is produced by The PHP Group. PHP was created in order to simply produce dynamic web pages. PHP’s main task is to work as a filter. The source code is interpreted by a Zend Engine and translated to a text stream that is output as a web page. PHP usually communicates with databases to present stored information. A popular database to use with PHP is MySQL. In the year 2006 over 40 percent of all web applications were written in PHP and over 2.5 million developers used PHP. PHP is free, simple, has hundreds of base functions, and a lot of online help - which has made PHP very successful. PHP can be embedded in HTML with the tags `<?php ...?>` just like JavaScript, but it runs on the server-side instead of client-side. PHP’s latest version 5.2.6 was released 1 May 2008. [20] [21]

Code Example 2-7: PHP

```
<?php
    #define the function
    function printText($inputVar){
        print($inputVar);
    }
    //make a function call
    printText("Hello World!");
?>
```

2.1.10. SQL

SQL [L] was designed for retrieving and modify data in a relational database. In early 1970, Donald D. Chamberlin and Raymond F. Boyce of IBM developed the first version of SQL called "System R" and Structured English Query Language (SEQUEL) for manipulating and extracting data from System R. The SQL language was standardized by the American National Standards Institute (ANSI) in 1986. Database engines that support SQL include FirebirdSQL, Informix, Microsoft Access, Microsoft SQL Server, MySQL, Oracle, PostgreSQL, etc. [22]

Code Example 2-8: SQL

```
SELECT Book.title, count(*) AS Authors
FROM Bookmarked /* The table were the info is located */
JOIN Book_author ON Book.isbn = Book_author.isbn
GROUP BY Book.title
```

2.1.11. XSLT

Extensible Stylesheet Language Transformations [M] is used to transform existing XML documents into new XML, HTML, XHTML, or plain text documents. XSLT was developed by W3C in 1999. Version 2.0 was released in 2007 and is designed to be used in conjunction with XPath 2.0, thus they share the same data model. XSLT 2.0 also includes facilities to serialize the result of transformations. Nearly all major browsers have support for XML and XSLT. XSLT consists of three parts: XSLT - a language for transforming XML documents, XPath - a language for navigating in XML documents, and XSL-FO - a language for formatting XML documents. Code example 1-11 shows two XML files, one contains the data and the other works as a parser. [23]

Code Example 2-9: XSLT

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<catalog>
    <cd>
        <title>Empire Burlesque</title>
        <artist>Bob Dylan</artist>
        <country>USA</country>
        <company>Columbia</company>
        <price>10.90</price>
        <year>1985</year>
    </cd>
</catalog>
```

(a) XML file with CD info

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
    <body>
        <table border="1">
            <tr bgcolor="#999999">
                <th>Title</th>
                <th>Artist</th>
            </tr>
            <xsl:for-each select="catalog/cd[artist='Bob Dylan']">
                <tr>
                    <td>
                         <xsl:value-of select="title" />
                    </td>
                    <td>
                         <xsl:value-of select="artist" />
                    </td>
                </tr>
            </xsl:for-each>
        </table>
    </body>
</html>
</xsl:template>
</xsl:stylesheet>
```

(b) XSL file with conditions

(c) This will display:

Title	Artist
Empire Burlesque	Bob Dylan

2.1.12. SVG

Scalable Vector Graphics [N] is an XML file defining two-dimensional vector graphics, both static and dynamic. It is an open standard and is developed by W3C since 1999. Since SVG files are in the XML format it is possible to open them with any text editor and they can be searched, indexed, scripted, and compressed. All modern web browsers support SVG except Microsoft's Internet Explorer which needs a plug-in to handle SVG. There are subset versions of SVG called SVG Tiny (SVGT) and SVG Basic (SVGB) which are used for handheld devices with limited capabilities. SVG can be very useful in IPTV boxes since it does not need as much system performance as Flash. However at this time, fall 2008, it is still unusual for STBs to include an SVG engine, one of the few that does is the Motorola's 1900-series which is used by some Telia customers in Sweden. According to the Vice President at Accedo Broadband roughly five percent of the world's deployed STBs offer SVG support today. [24]

Code Example 2-10: SVG

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
    "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">

<svg xmlns="http://www.w3.org/2000/svg">
    <title>Circle</title>
    <circle r="100px" cx="200px" cy="200px"/>
</svg>
```

2.1.13. Flash

Adobe Flash [O] previously called Shockwave Flash and Macromedia Flash was created by Macromedia in 1996. Flash has become a very popular means of adding animation and interactivity to web pages. Flash can manipulate vector and raster graphics and supports bi-directional streaming of audio and video. It uses ActionScript [P] as scripting language to control and display graphics. There are many versions of ActionScript; the most recent one is version 8. Most desktop web browsers and some mobile phones support Flash, however, mobile phones generally run the Flash Lite version because of their limited capabilities. [25]

2.2. Popular Web 2.0 services

2.2.1. Flickr

Flickr (www.flikr.com) is a free community for photo and video hosting. The website is frequently used by bloggers as a photo repository; in November 2007 Flickr had over 2 billion images. Flickr was developed by the Canadian company Ludicorp in 2004. In the beginning Flickr focused on offering a multiuser chat room called FlickrLive where users shared images in real-time. These images were not primarily personal photographs, but were images found on the web; later users began to utilize Flickr as a personal image upload service. In 2005 Yahoo acquired Ludicorp and Flickr, migrated all the content from servers in Canada to servers in the U.S. Flickr uses: REST[1], XML-RPC, and SOAP as a request format and REST, XML-RPC, SOAP, JSON, and PHP as a response format. [27]

Flickr's design is simple and with a few abstractions it seems to fit very well with IPTV. The requirements for navigation are a keyboard and an input field for searching purposes. The four arrows on the remote control for the STB should be sufficient for navigating between different parts of the site. Features that probably will be used on an IPTV interface are (1) searching for user or keyword, (2) displaying and navigating between different parts of the site, and (3) navigating through the photographs.

2.2.2. Blogger

Blogger (www.blogger.com) provides a blog publishing service with some extra features, such as the Picasa photo-sharing utility and the availability to post by email or a cellphone. Their website provided templates follow the web-standards, so you do not have to code anything by yourself. It is possible to comment on other blogs, restrict who can read and comment on your blog, and to search for those who have a similar profile to you [28]. Blogger allows third-party applications and has a long list of recommended applications at their site. It is not difficult to understand why there are a lot of applications – as Blogger provides an excellent API which is open for any developer to use [29]. There are developer guides and libraries for many popular programming languages, amongst them: .NET, Java, and PHP. Blogger also provides a JSON feed, for those who want to access blog content from their own application.

Blogger's API uses XML as a response format. It is unlikely that user will write blog content using the remote control, but a user is likely to search and read blog articles by others. The main use in conjunction with IPTV is likely to be to searching for a specific user, specific content, or top-rated blogs, and displaying them in a viewable manner. An extra feature could be to present blog content based on what the user was recently watching on their IPTV.

[1] Representational State Transfer (REST) is described in detail in section 2.3.1 of [26].

2.2.3. Del.icio.us

Delicious (www.delicious.com) formerly known as del.icio.us, is a free social bookmarking web service for storing and sharing web bookmarks. The site was founded by Joshua Schachter in 2003 and acquired by Yahoo in 2005. It has over five million users and 150 million bookmarked URLs. Delicious uses a keyword categorization technique where users tag their bookmarks with a number or keyword. This feature makes it possible to view bookmarks generated by similar-minded users. The rich set of features Delicious provides has made it popular. These features include: human-readable URLs, a novel domain name, a simple API, and RSS feeds. Their API uses XML as the data response format. This service is not suitable for IPTV as we use it today, as the bookmarking feature is a better fit for services that you navigate to using a URL, as on the web with a keyboard. Although a version that utilizes a display of tags relevant to this user might be useful. [30]

2.2.4. Facebook

Facebook (www.facebook.com) is the social networking service used by a very large number of, especially younger, people to chat, find new friends, share photos, and much more. It was originally only for university students, but has grown and does now accept everyone older than 13 years of age as a member. It is possible to join different networks based on where you live, study, and so on. They offer a toolkit for developing third-party applications and have a nice developer API. Facebook develops and uses own protocols to optimize the performance of their system. The Facebook Query Language was created for third-parties who want to fetch data. Facebook Markup Language is an evolved subset of HTML for the application developers to use to customize the design of their applications. The main competitor is Google's MySpace which offers similar functionality and generates almost as much traffic, but has more than twice the number of monthly visitors as Facebook. MySpace also allows their own users to design their own pages with HTML and CSS while Facebook only allows plain text. Facebook is a very complex service - some of its features could fit well with the IPTV interface, while others seem to be almost impossible to implement in a way that is easy to use without a keyboard. To search for a person and see that person's list of friends, watch other user's images, and being able to read your latest chat messages are popular features that probably fit the IPTV format best. [31] [32] [33]

2.2.5. Wikipedia

Wikipedia (www.wikipedia.org) is the largest, popular, and multilingual free encyclopedia. It was created by Jimmy Wales and Larry Sanger. In April 2008, Wikipedia had more than 10 million articles in 253 languages; a quarter of the articles were in English. The articles have been written by volunteers around the world and everyone is free to edit the existing entries or make their own new entries. Due to the very large number of users, faulty articles get changed rapidly and the quality of the articles based on science is often very high. There are even professors who give their students as an assignment to write new articles as a part of their education [34]. There is also a forum for discussing the quality, lack of sources, and other aspects of the articles. Articles with only a few references are tagged as having poor quality, in order to inform the reader that the article may not be correct. Wikipedia was one of the first Web 2.0 service and was soon followed by YouTube and MySpace. [35] If Wikipedia were to be implemented for the IPTV it is more likely that the user will be primarily interested in searching for and read articles, than to post new content or be part of a discussion in a forum. It may be possible to scale down large articles to make them easier to read on the limited screen space available to the STB. The KTH School of Computer Science and Communication (CSC) have a project [36] [37] for automatic text summarization.

2.2.6. YouTube

YouTube (www.youtube.com) is a video sharing website where users can upload, watch, and share video clips. It was created in 2005 by three formal Paypal employees: Steve Chen, Chad Hurley, and Jawed Karim. YouTube uses Adobe Flash Player to display video clips, thus it only requires a web browser plug-in in order to view these videos. An advantage of using the Flash format is that roughly 90% of all online computers use it. In November 2006, Google bought the company for $1.65 billion in Google stock. YouTube has about 6.1 million videos which require about 600 terabytes of storage space, 500 thousand user accounts, and it now consumes as much bandwidth as the entire Internet had in the year 2000, due to approximately 13 hours of video being uploaded every minute [38]. In March 2008, its bandwidth costs were estimated to be $1 million a day. Videos are limited to ten minutes in length and the maximum file size is 1 gigabyte. There has been criticism of YouTube due to the many unauthorized clips that have been uploaded by users from TV shows, movies, and music videos. After a lawsuit from Viacom, the Premier League, and others - YouTube introduced a system that checks uploaded videos against a database with known copyrighted material to reduce violations [39]. YouTube is more socially accepted as a medium many trailers and commercials even the newly elected American president Barrack Obama used YouTube when he campaigned and gave his nation speech [40]. Creating an IPTV YouTube application would require a Flash player. Unfortunately, YouTube does not allow users of their open API to retrieve a video with resolution greater than 176x144 pixels for the H.263 CODEC. Note that the Motorola/Kreatel 1510 STB supports only a H.263 CODEC; although other versions support additional video CODECs. However, content could be received as *.flv* files and transcoded to H.263 *.mpg* files which the Motorola 1510 does support. If the user is given a way to search for content, it would be possible to categorize videos by topics to minimize typing; for example, by having lists of top-rated movie clips, the most viewed clips, and more. These topics could be accessed by a simple menu bar, which eliminates unnecessary typing. [41]

2.2.7. IMDb

IMDb, the Internet Movie Database (www.imdb.com) is a large on-line database with information about movies, actors, TV-shows, direct-to-video products, videos in production, video games, etc. IMDb was launched in 1990 by Col Needham. In 1998, Amazon acquired it. All the basic information is freely available, but if you want to submit or edit information then you need to be a registered user. IMDb had 57 million visitor and 17 million registered users as of 2007. The database content is provided and updated by volunteers, but 20 members of the IMDb staff are dedicated to monitoring received data. In 2002, the site introduced "IMDbPro", a service which offers additional information about business professionals, such as: contact details, titles in development, movie event calendars, and a greater range of industry news. [42] The basic feature "Now Playing" shows information for movies currently playing at cinemas, some of the movie top lists, offers the ability to watch trailers, and of course provides free searching for movies and TV-series. This last aspect is a very interesting application that could couple IMDb with

IPTV. The site requires users to log in as members to be able to rate movies. As this would definitely be an interesting feature for IPTV users, there needs to be a way for the user to do this - for example, by using a web browser on their computer to browse to the IPTV device and edit a IMDb account profile - thus when using the STB, the user would only have to select their profile and they would be automatically logged in. IMDb also provides a news blog with gossip about actors and movies; this would also probably be interesting for many IPTV users.

2.2.8. MySpace

MySpace (www.myspace.com) is a social networking service developed by several employees from eUniverse. It offers personal profiles, blogs, group photos, music, and video content. In June 2006, MySpace was the most popular social networking site in the U.S. MySpace was sold in 2005 for $580 million to the News Corporation which also owns Fox Broadcasting and other media enterprises. MySpace increases by 230 thousand new users per day. MySpace's biggest competitor is Facebook. MySpace has a feature - MySpace Music – were musicians can upload their songs in the mp3 format. Unsigned musicians often use MySpace to post and sell their own music. Over eight million artists have been discovered by MySpace, some well known singers such as Lilly Allen and Sean Kingston gained fame through MySpace. [43] MySpace offers APIs that are free for everyone to use and they provide a forum were you can ask questions to professional developers and learn more about how to develop MySpace applications. These APIs are for applications running on subparts of the MySpace website, similar to the Gadgets [44] in the Microsoft Vista operating system, rather than as stand-alone applications. [45]

2.2.9. Twingly

Twingly (www.twingly.com) is a Swedish search engine developed 2006 in Linköping. Many big newspapers in a number of different countries uses Twingly to discuss the news with each other; examples include **Dagens Nyheter** in Sweden, **Helsingin Sanomat** in Finland, **Politiken** in Denmark, **Dagbladet** in Norway, **Público** in Portugal, **De Telegraaf** in the Netherlands, and the **Sunday Times** in South Africa. Twingly makes it possible for users to add their blogs if the search engine can not find it by adding the URL to Twingly. Twingly is currently developing a spam-free blog search by utilize a massive scale chain of trust and creating a white list of approved blogs. [46]

2.2.10. Statistics

There is a service, called Alexa rank, which analyzes web usage and creates a ranking list sorted by the amount of traffic based on number of page views and percentage of all global Internet users who visit a given site during the last three months. The two websites at the top of this list are the search engines Yahoo! (Alexa rank 1) and Google (Alexa rank 2).

Table 2-1: Web 2.0 sites statistics as of 17 December 2008

Name	Launched	Owner	Monthly users [47]	Alexa rank [48]	Global Internet users in % 3mos. Avg. [48]
IMDb	1990	Amazon	11 million	44	1.33 %
Blogger	2000	Google	-	9	6.97 %
Wikipedia	2001	Wikimedia foundation	43 million	8	8.34 %
Del.icio.us	2003	Yahoo	1.3 million	2,683	0.04 %
MySpace	2003	Fox Interactive Media	53 million	7	5.94 %
Facebook	2004	Private	25 million	5	11.41 %
Flickr	2004	Yahoo	9.7 million	30	1.78 %
YouTube	2005	Google	32 million	3	17.11 %
Twingly	2007	Private	-	42,819	0.002 %

2.3. Useful functions

2.3.1. Tag-cloud

Tag-cloud or word-cloud is a machine generated visual box containing words that displays the most popular search words with different font sizes and colors. Usually a cloud contains about 30 to 150 words; the weights are represented using different font sizes or different colors. The first use of tag-clouds was by Stewart Butterfield, the Flickr Co-founder and interaction designer. A tag-cloud can help a user to find new search topics or discover what is popular right now. [49]

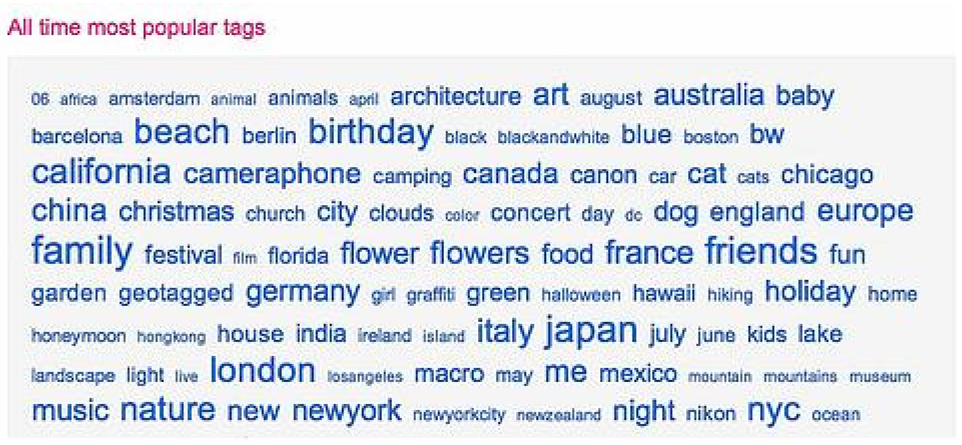
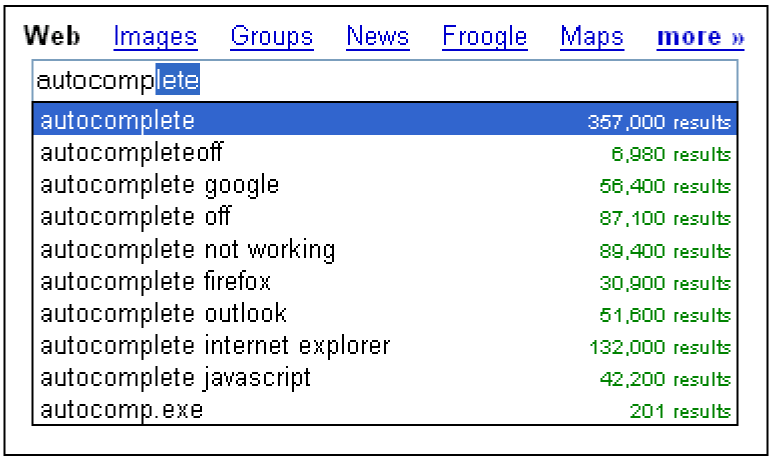

All time most popular tags
06 africa amsterdam animal animals april architecture art august australia baby
barcelona beach berlin birthday black blackandwhite blue boston bw
california cameraphone camping canada canon car cat cats chicago
china christmas church city clouds color concert day dc dog england europe
family festival film florida flower flowers food france friends fun
garden geotagged germany girl graffiti green halloween hawaii hiking holiday home
honeymoon hongkong house india ireland island italy japan july june kids lake
landscape light live london losangeles macro may me mexico mountain mountains museum
music nature new newyork newyorkcity newzealand night nikon nyc ocean

Figure 2-2: Flickr's tag-cloud

2.3.2. Auto-suggest

Auto-suggest or auto-complete is a feature widely used in text editors, email software, and web browsers to predict common words or phrase that user wants to type without typing it completely. Web browsers use it in search boxes and forms, email software to fill in email addresses, and word processors with repeated words in text documents. This feature is especially useful for services that do not have a keyboard available, as in our case where the STB user only has a remote control. Google-suggest was one of the first services that used auto-suggest for the web. [50]



autocomplete

autocomplete	357,000 results
autocompleteoff	6,980 results
autocomplete google	56,400 results
autocomplete off	87,100 results
autocomplete not working	89,400 results
autocomplete firefox	30,900 results
autocomplete outlook	51,600 results
autocomplete internet explorer	132,000 results
autocomplete javascript	42,200 results
autocomp.exe	201 results

Figure 2-3: Google's auto-suggest

2.3.3. Virtual Keyboards

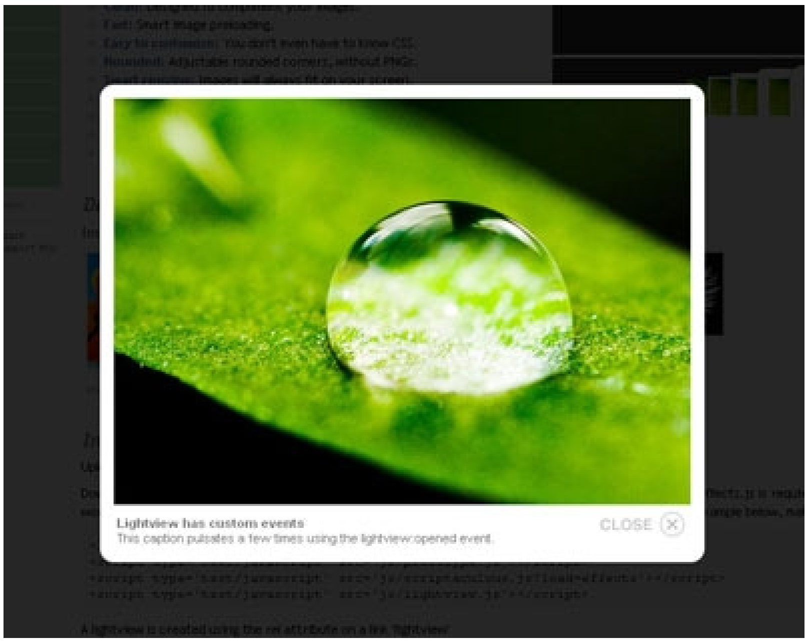
A virtual keyboard is a software component that allows a user to enter characters. The purpose is to provide an alternative input mechanism for users that can not use a physical keyboard. Many JavaScript keyboards have been created allowing users to type in their own languages on foreign keyboards (as often found in Internet cafes). [51]

2.3.4. T9

T9 [52] stands for "Text on 9 keys" and is a predictive text technology mainly for mobile phones developed by Tegic Communications, now part of Nuance Communications. T9 allows single key presses for each letter instead of multi-taps, it combines groups of letters on each key and looks up in the dictionary of all words corresponding to the sequence of key presses and orders them by frequency of use. T9 can add new words and phrases enabling them to be recognized in the future. T9 could be a useful feature for IPTV since almost all remote controls have the buttons 0-9. As of autumn 2008 it seems that Apple is the only popular STB vendor that does not have these numeric keys on their remote controls. A problem with the Motorola 1720 STB remote control is that although it has the numeric keys, there are no letters written beneath them, e.g. the user does not know which letters each numeric key represents. On most modern remote controls it is exceptional if they do not provide the letters, in a similar way as a cellular phone's keyboard.

2.3.5. Modal window

A modal window is a child window that requires the user to interact with it before interaction returns to the main window. It is commonly used in graphical user interface (GUI) applications to block the application flow until the information that is required to proceed is entered, for example a login process that is required before an image is displayed. There are some criticisms of modal windows because this window blocks the program until it is closed and many users do not recognize this behavior, hence may be confused that the main window is not responding. Examples of modal windows are Lightview (see Figure 2-4), ThickBox, TinyBox, Lightbox, Submodal, etc. [53]


Lightview has custom events
This caption pulsates a few times using the lightview:opened event.
CLOSE
Figure 2-4: Lightview's Modal window

2.3.6. Speech recognition

Speech recognition converts spoken words to machine-readable inputs, for example key presses or to identify the person speaking. There are speech recognition applications that include voice dialing, for example you can program some phones to call home when you say "Call home". For simple data entry for IPTV, speech-to-text could be a useful tool to input search strings. Additionally people also people with disabilities can benefit from using speech recognition programs [54]. However a TV is usually located several meters away from the user, the user would need to utilize a microphone with some type of wired or wireless link to the STB in order to have high accuracy speech recognition. Few users have tried speech recognition applications. It is likely that at least initially they will find the experience quite frustrating (for example when using the speech recognition built-into Vista). This frustration is to lead to a negative attitude regarding speak recognition applications. At this point in time, this technology is probably not an application for the masses.

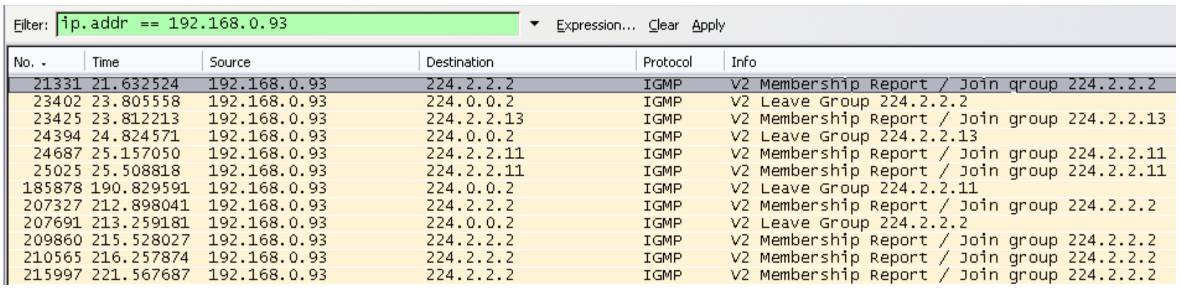
2.3.7. Bluetooth

Bluetooth is a wireless communication protocol for exchanging data over short distances from mobile phones, telephones, laptops, PC, printers, digital cameras, video game consoles, etc. It is also possible to use a Bluetooth equipped mouse or a keyboard as input devices. A Bluetooth equipped STB could enable you to use your mobile phone as remote control. In IPTV applications you could use Bluetooth to upload pictures or text to web services through the STB (if your STB supports it), for example upload pictures taken with your mobile phone to your Flickr account. If the STB does not support a Bluetooth interface by default (most likely) there are often USB ports on the STB (such as the Motorola 1720) and there are lots of cheap Bluetooth devices that could be used with this USB interface. If the STB vendor does not provide a driver for Bluetooth you have to write your own, which is not a simple matter. However, as a very large number of STB's utilize Linux as their operating system, there exists Linux Bluetooth drivers for most USB equipped Bluetooth devices. There are three different Bluetooth versions - version 1.2 has a maximum data rate of 1 Mbps, version 2.0+EDR 3 Mbps, and WiMedia Alliance has (proposed) a version with maximum data rate of 53-380 Mbps. [55]

2.4. Set-top box

A set-top box (STB) is a device for connecting a TV with an external information source, in this case a TCP/IP network. The typical STB device uses an Ethernet port for the physical connection, but some STBs use other networks such as HomePNA, broadband cable, etc. Many STB providers have a software development kit (SDK) which they offer their partners, in order to ease the development of programs for their STBs. Since these programs have a very high development cost and long time for deployment, it is often simpler for IPTV developers to use the STB's web browser to run web applications rather than creating stand-alone programs.

For this master's thesis project we are developing our own web based applications for the Motorola 1720 STB. It is a widely deployed STB and compared to many other STB vendors, Motorola has focused a bit more on the web browser and has optimized the performance of web applications. This STB runs a relatively modern Mozilla browser on top of a Linux operating system. The support for JavaScript, CSS, etc, is high. The Motorola 1720 STB offers some security mechanisms, such as log-in. In figure 2-5 you can see the packets captured by Wireshark when the STB is booting, the STB uses the IGMP protocol to communicate with the boot server. This STB can request a dynamic IP address from a DHCP server or be manually assigned a static IP address (see figure 2-6). It is possible, to access the STB via Telnet/SSH. Motorola has implemented the ability to retrieve information about memory consumption by accessing a certain port on the STB (UDP port: 19999) with Motorolas Logclient v1.1 (see Appendix A). Motorola's code sends all "shell output" information to a buffer which streams its content at this port.


```
Filter: ip.addr == 192.168.0.93

No.     Time        Source          Destination   Protocol  Info
  21331 21.632524   192.168.0.93    224.2.2.2     IGMP      V2 Membership Report / Join group 224.2.2.2
  23402 23.805558   192.168.0.93    224.0.0.2     IGMP      V2 Leave Group 224.2.2.2
  23425 23.812213   192.168.0.93    224.2.2.13    IGMP      V2 Membership Report / Join group 224.2.2.13
  24394 24.824571   192.168.0.93    224.0.0.2     IGMP      V2 Leave Group 224.2.2.13
  24687 25.157050   192.168.0.93    224.2.2.11    IGMP      V2 Membership Report / Join group 224.2.2.11
  25025 25.508818   192.168.0.93    224.2.2.11    IGMP      V2 Membership Report / Join group 224.2.2.11
 185878 190.829591  192.168.0.93    224.0.0.2     IGMP      V2 Leave Group 224.2.2.11
 207327 212.898041  192.168.0.93    224.2.2.2     IGMP      V2 Membership Report / Join group 224.2.2.2
 207691 213.259181  192.168.0.93    224.0.0.2     IGMP      V2 Leave Group 224.2.2.2
 209860 215.528027  192.168.0.93    224.2.2.2     IGMP      V2 Membership Report / Join group 224.2.2.2
 210565 216.257874  192.168.0.93    224.2.2.2     IGMP      V2 Membership Report / Join group 224.2.2.2
 215997 221.567687  192.168.0.93    224.2.2.2     IGMP      V2 Membership Report / Join group 224.2.2.2
```

Figure 2-5: Motorola 1720 start-up sequence with Wireshark

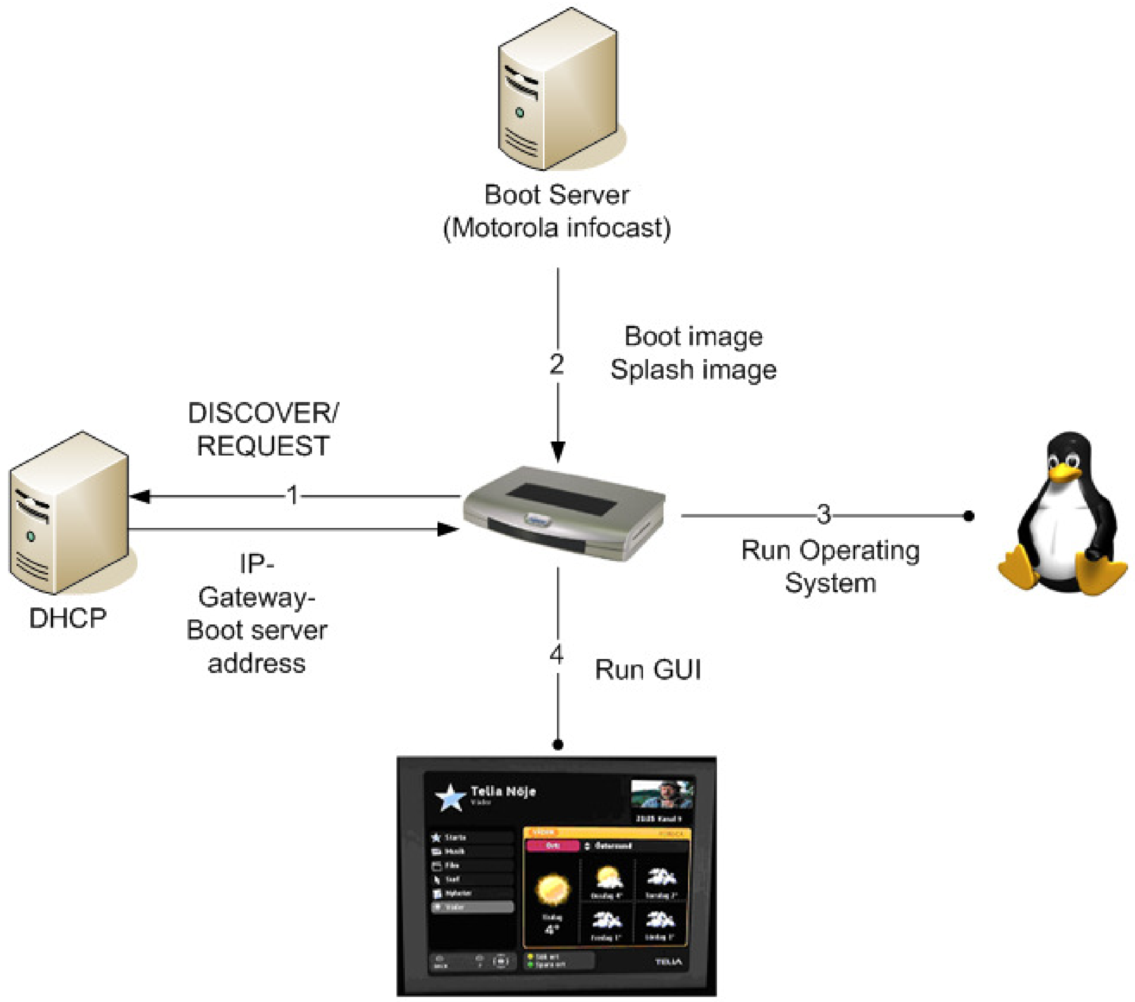


Figure 2-6: Motorola 1720 start-up sequence

There are some features missing from the 1720 STB that are available from other STBs, these are support for Flash plug-ins, SVG, and HDTV. STBs which include these features are not yet widely used in the IPTV world market. Telia are deploying the Motorola 1910 (which is a HDTV STB) to some of its IPTV customers in Sweden, but the majority of customers are using either the model 1510 or 1720 STBs. The 1900-series provides HDTV while watching TV, but the included Mozilla browser does not run at HDTV resolution. According to developers at Accedo Broadband, HDTV STBs are still in quite an early deployment phase and are often only upgraded with a new decoder for the input signal with a higher resolution and a HDMI port instead of the older S-Video port. The need to support higher resolution images results in a higher workload on the processor and the memory consumption increases which leads to slower operations. The Flash plug-in does not enable a much higher degree of flexibility since it is just a plug-in to the web browser. SVG does not yet appear to be deployed in any STB provided by the major IPTV operators, but there are SVG rendering engine companies that are partners with STB vendors, for example Motorola (a STB vendor) and Ekioh (SVG engine), thus it is likely that SVG enabled STBs will be deployed in the near future. SVG will probably contribute a lot more than Flash when creating interactive applications for the IPTV format in the future, since SVG is an open format with many different developers competing with each others, rather than Adobe's monopoly for Flash.

2.4.1. Motorola 1720 STB

The first STB we have used in this project is the Motorola (formerly the Kreatel) 1720 (shown in figure 2-7) which has a MIPS processor (with a performance of 420 DMIPS) and a 128 MB of DRAM. It can handle MPEG-2 streaming video at SIF (352x288) resolution, with a variable bit rate ranging from 2 to 25 Mbps. This STB supports a resolution of 720x576 pixels for PAL computer graphics at 24 bits per pixel. This STB uses SSL and PKI for encrypted data. [56]

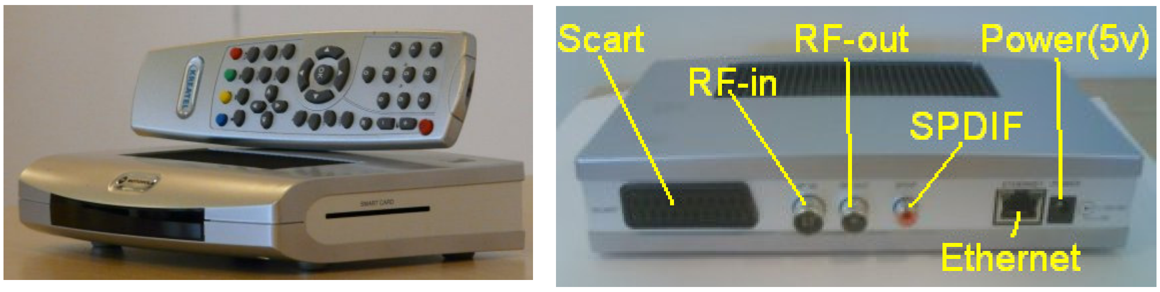


Figure 2-7: Kreatel/Motorola 1720 STB

This Motorola STB utilizes a modified Linux version as its operating system. Figure 2-8 shows the Motorola Layered architecture with five layers. The lowest level is the STB physical hardware, second is the Hardware Abstraction Layer which is responsible for communication between the hardware and applications. Third are the operating system and device drivers. The second is the service layer which the applications can reach through the Motorola Terminal Open Interface (TOI), see Appendix B, this is a vendor specific library. On the top we have the application layer where our services run.

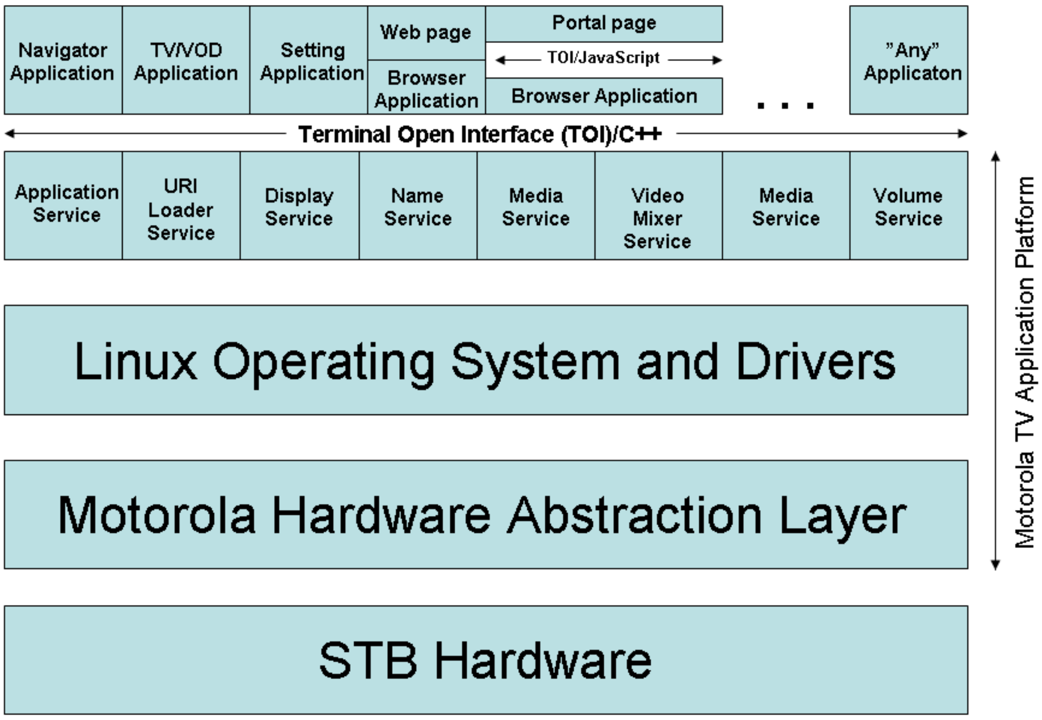


Figure 2-8: Motorola Layered Architecture

2.5. Related Work

We have not yet found any general framework developed for Web 2.0 services such as this master's thesis proposes. There are some web services that have been converted to an IPTV service, but all have been individually coded. Others have tried to interact with different Web 2.0 services. Johan Ekström at Växjö University presented a report [57] in 2006 about Wikerpedia, were Wikipedia interacted with Flickr; his conclusion was that it increased the usability when photographs from Flicker were matched with the sites at Wikipedia. He used XML feeds from Flickr and integrated this content with the wiki by presenting the photographs as thumbnails at the end of each Wikipedia page. Social Bookmark Script is a service for generating scripts for different social bookmarking sites. [58]

Marine partners is a company that has developed middleware called Frostt, to run on top of the IP layer. Middleware is software that connects different software components or applications to enable services that allow multiple processes running on one or more machines to interact across a network. The services Frostt offers are Parent control, billing, profile management, and authorization control. Frostt also has the possibility to integrate Web 2.0 services in their middleware as so called Frostt Support Services. [59]

Lars Mikael Liljeroth's master's thesis, "Hardware accelerated user interface architecture for IPTV set-top boxes", is about developing a fast and responsive user interface (UI) in EMCAscript (see section 2.1.4) for an IPTV STB. This project required more advanced programming skills and much time was spend on dealing on the hardware problems because of the need to exploit low level functionality and to perform memory allocation. [60]

According to a presentation by John Allen at the JavaOne^{SM} Conference 2007, Java is a major potential technology for IPTV. There are already a lot of electronic devices that support Java today, approximately 10 billion devices in 2008 and still growing. The advantages of using Java technology for IPTV applications are that Java is flexible for developers, partners, and customers; portable – a cross platforms; scalable; there are large numbers of Java solutions available; and that there is a large support infrastructure. [61]

According to the newspaper Metro Teknik; Ericsson is currently developing Web 2.0 applications for their IPTV program. The first applications are Flickr and Facebook, and these applications are to be launched at the Mobile World Conference in February 2009. The services will run as web interfaces and they will be included in a package called Life store. [62]

Ola Haraldson Halset's master's thesis, "IP Television - Content on Demand: A usability and user experience evaluation of an IP television portal by the Norwegian broadcaster TV 2 Interaktiv" evaluates the Norwegian broadcaster TV2's portal for IPTV. [63]

Robert Högberg's master's thesis, "Video telephony in an IP-based set-top box environment" describes the implementation of a video telephony solution for networked connected set-top boxes based on the SIP protocol. This thesis uses a USB attached camera to make a video telephony system using the Kreatel STB: [64]

John Brännström's master's thesis, "Bokningssystem för gemensamhetsresurser via TV" concerns the implementation if a reservation system for common resources through a TV-interface. The system will cooperate with the company Dreampark and is built to work in their IPTV platform. [65]

Behzad Hamzei Tavosloi’s master’s thesis, "Design and Implementation of an Application Server for an IPTV Environment Architecture, Prototype System Design, Implementation and valuation" describes the design and implementation of an application server for an IPTV environment. The goal was to find a way to simplify the administration of the IPTV environment based on Java and a Tomcat server. [66]

Camilla Isaksson’s master’s thesis, ”Gränssnitt och tjänster för IPTV” concerns the development of user friendly interfaces for IPTV. The user interface should work on an Amino STB and on a workstation. The user interface will replace the middleware in order to simplify updates and changes. The interface was written in HTML/CSS, JavaScript, and PHP. [67]

3. Method

As a part of this master's thesis we have implemented interfaces to four Web 2.0 applications to fit the IPTV format. We choose Flicker (as a photography viewer), Twingly (for presenting blog content), IMDb (to show movie information), and Wikipedia (The free Internet encyclopedia). We think that these four services suit the IPTV well and Wikipedia and IMDb, Flickr are well known services. We wanted to make a light version of an interface to these four services and use a similar layout for each application. We kept the design as simple as possible; with a virtual keyboard, a search field, and a submit button for input in all four applications. All four services are easy to navigate through using a remote control. We used these four applications to find similarities and differences amongst popular Web 2.0 services. The knowledge gained in this first phase was used to develop generic modules to ease the development of future Web 2.0 applications for the IPTV format.

In the beginning we experimented with HTML to learn what would work on the STB. We tried to use predesigned virtual keyboards, but the problem was how to type when the STB does not have a mouse cursor, so we had to create our own virtual JavaScript keyboard – using the remote control for the STB (see figure 3-1). The same problem occurred with the search button, we could not press the button without a cursor so we had to make our own search button as a picture and switch to another picture when the button is marked. The next problem was to make it look like we are navigating around the page (using the up arrow, down arrow, left arrow, and right arrow buttons on the remote control of the STB but without a cursor); the solution was to change the font to bold on the current letter when it is selected (using the arrow keys button on the remote control) on the keyboard, for pictures we changed the opacity, and for the blogs we used a frame. We had to keep track of each virtual cursor movement on the page by defining coordinates for each position and making different parts of the page visible or hidden depending on where the user has moved or what they have pressed. We parse the IR key code sent by the remote control when the user navigates through our application (see table 3-1).

Table 3-1: Motorola keycodes

Key	Keycode
Backspace	0x00080008
Tab	0x00090009
Return	0x000D000D
Shift	0x00100010
Control	0x00110011
Alt	0x00120012
Capslock	0x00140014
Stop	0x001B001B
Standby	0x011B001B
Space	0x00200020
Left	0x00250025
Up	0x00260026
Right	0x00270027
Down	0x00280028
Scroll up	0x01260026
Scroll left	0x01250025

Scroll right	0x01270027
Scroll down	0x01280028
0	0x00300030
1	0x00310031
2	0x00320032
3	0x00330033
4	0x00340034
5	0x00350035
6	0x00360036
7	0x00370037
8	0x00380038
9	0x00390039
Info	0x00700070
Text	0x00710071
TV	0x00720072
WWW	0x00730073
Red	0x00740074
Green	0x00750075
Yellow	0x00760076
Blue	0x00770077
Portal	0x00780078
Back	0x00790079
OK	0x007A007A
Menu	0x007B007B
Volume Up	0x01720072
Volume Down	0x01710071
Mute	0x01730073
Media Play	0x01740074
Media Rew	0x01750075
Media FF	0x01760076
Media Stop	0x01770077

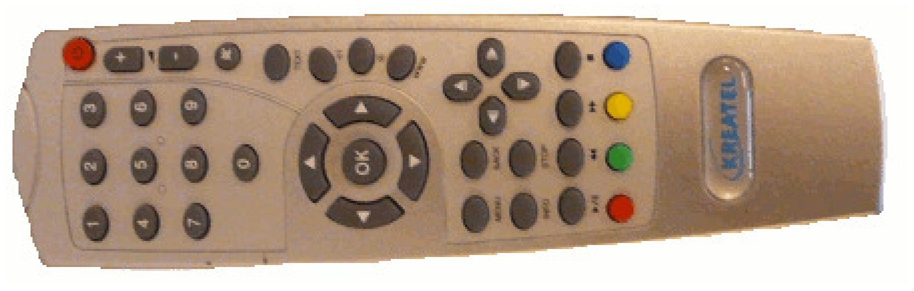
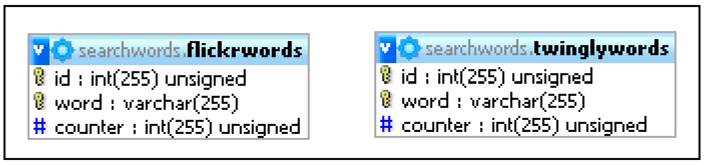


Figure 3-1: Motorola remote control

We have implemented two important Web 2.0 tools: Tag-cloud/Word-cloud and Auto-suggest. The Tag-cloud contains the users most popular search words, the bigger the font the more popular the word. We have chosen to list the 10 most popular search words alphabetically. The search words are stored in two separate MySQL tables (see Figure 3-2), one for Flickr and one for Twingly. The table has three fields: id, word, and counter. The id is a unique identifier for each word, word contains the text of the word, and the counter contains number of times the word has been searched for. A new word is stored only if a blog is found in Twingly; while in Flickr a new word (a user name) is stored only if the username exists and has public photos. The font size is a function of the value of the counter; using one of five different font sizes. The database can add the most common words from the articles we request to increase the number of word we can use for auto-suggestion. This could be useful in the future when users are searching for articles or blogs, as according to Ani Nenkova the 30 most common words account for 30% of the tokens in written text. [68]



searchwords.flickrwords
id : int(255) unsigned
word : varchar(255)
counter : int(255) unsigned

searchwords.twinglywords
id : int(255) unsigned
word : varchar(255)
counter : int(255) unsigned

Figure 3-2: MySQL tables

The Auto-suggest drop-down menu uses the Tag-cloud database to list words based on each key input, if a word does not match a word in database, then the drop-down menu will display “no suggestions”, this could easily be changed to nothing at all but we wanted to show to the user that we have that feature. The drop-down menu lists at most five words that a user can choose with the 1-5 numeric keys. The word will then be inserted in the search field and a search will occur. This is a very useful feature because it will reduce the number of key presses. The number of key presses will be one of the metrics we use to evaluate our applications with.

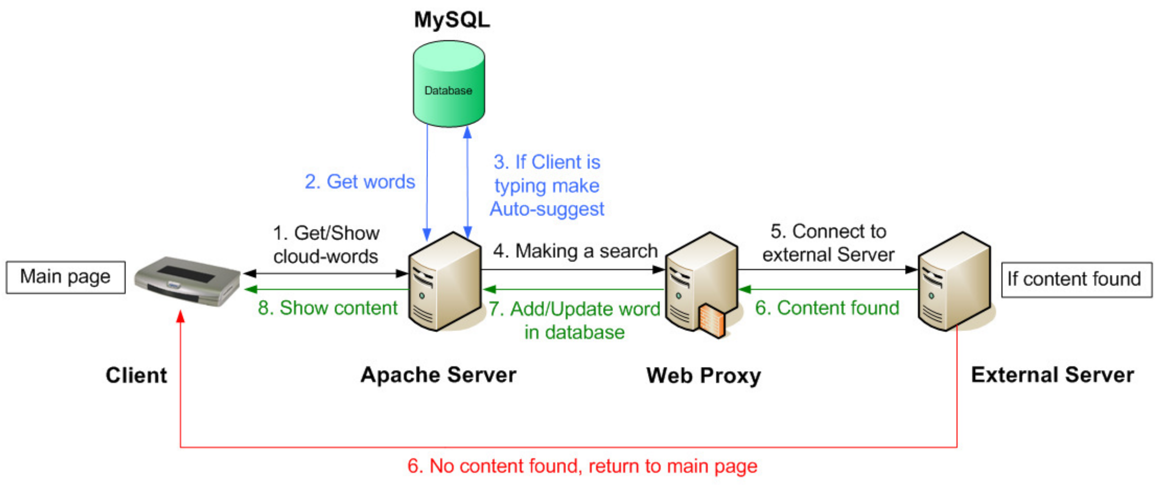


Figure 3-3: Overview of client/server interaction

Figure 3-3 presents an overview of how our service works. When a client navigates to the main page a connection will be established to an Apache server that retrieves potential words from the MySQL database and creates a tag-cloud for the client. When a user types into the search field a suggestion will appear if there is one or more entry in our database starting with those letters. When the user presses the search button or selects a word in the cloud-word, the Apache server will make a URL/image request to a web proxy that will forward the request to the external server. If the result of this request is a success (i.e., a 200 OK is returned from the external web server), then we will insert or update the word counter in the database and display received contents. If a search string does not exist, then the user will be notified and return to main page (i.e. the page where the user were when making the search request).

The services we have developed are created for testing purposes only; if any of these services are going to be commercial products then the company needs to make agreements with the original provider to be able to use the appropriate trademarks and content for the IPTV market. These issues are outside the scope of our master theses.

3.1. Flickr

Flickr (http://www.flickr.com/services/api/) has a very good API for external developers. Flickr uses URL, SOAP, and XML-RPC as request formats and supports many response formats, such as REST, XML-RPC, SOAP, JSON, and PHP. They also provide many library methods to retrieve contact info, group info, picture info, comments, and much more.

To make a method requests to Flickr server it is mandatory to have an authentication key, which is free after signing up for one on their homepage. This authentication key enables Flickr to know who is making requests and if some one is burdening their server too much.

When a user enters a username in our application we create a URL, specifying the Flickr method we want to use, which response format we would like, our authentication key, and the search string (see Code example 3-1). Then we make a request through our proxy, to Flickr's server. After receiving a response (see code example 3-2), we parse it and extract the interesting information to use as input in another method call. When we receive the information the user is interested in, e.g., links to the user's pictures, then we use this to generate an update to display them on our webpage.

Code Example 3-1: An example URL request

```
http://www.flickr.com/services/rest/
     ?method=flickr.people.findbyusername
     &api_key=XXX
     &format=json
     &username=urbanfeel
```

Code Example 3-2: The JSON response

```
jsonFlickrApi
(
     {"user":
          {
               "id":"30003006@N00",
               "nsid":"30003006@N00",
               "username":
               {
                    "_content":"urbanfeel"
               }
          },
     "stat":"ok"
     }
)
```

In code example 3-2, the response contains the following information:

```
jsonFlickrApi   - A function that returns the array containing user and
                  the status
id              - Userid
nsid            - Userid
username        - The username you searched for
stat            - Status, if found or not
```

Figure 3-4 shows the main page of our Flickr application. At the top there is a search field and a search button. To the left is a tag-cloud presenting the ten most searched for strings and on the right side a virtual keyboard. Note that in this figure the letter "a" is currently highlighted so if the user pushes the “OK” button this letter would be entered into the search string. The user will navigate around the screen using the up, down, left, and right arrow buttons.

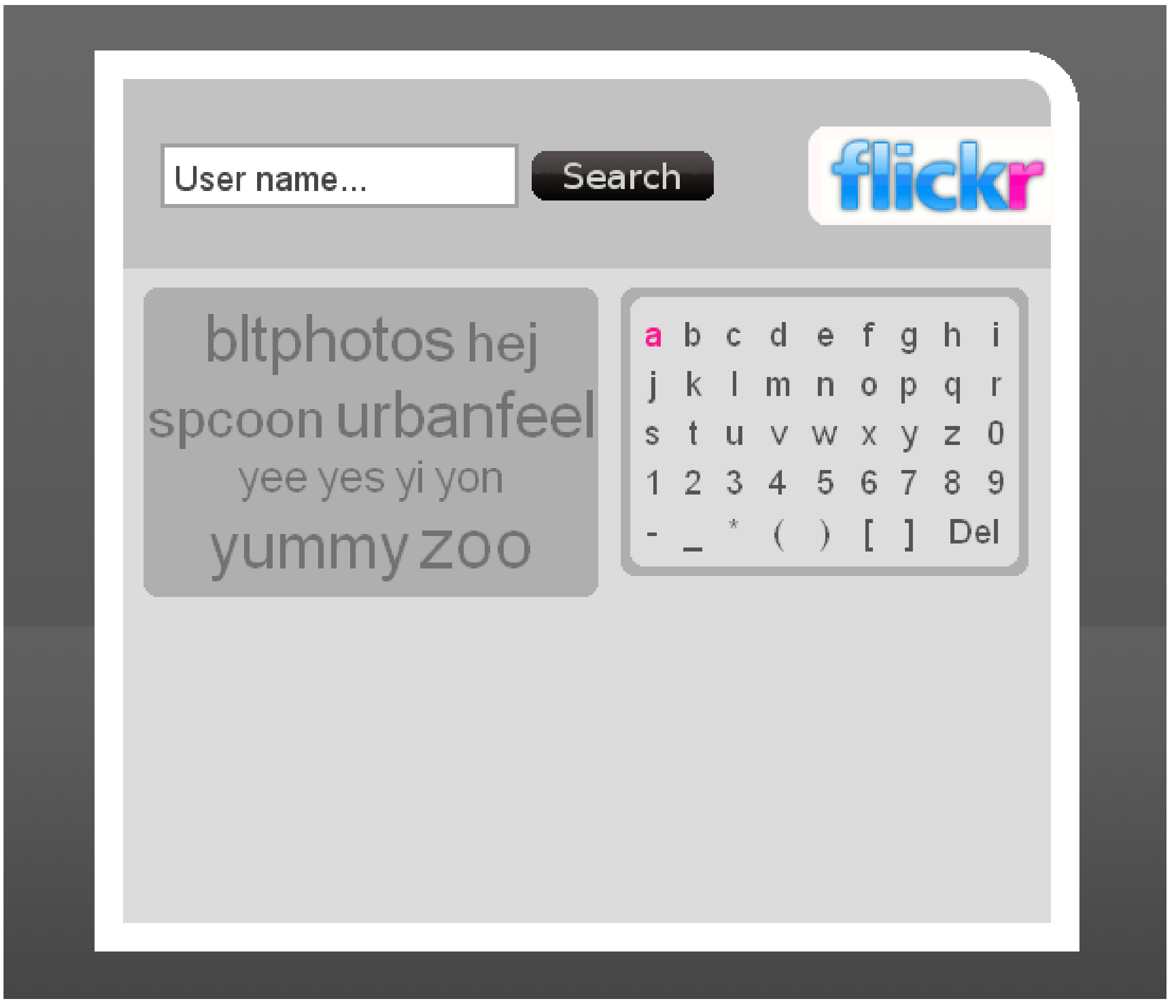


Figure 3-4: Flickr main page

A user can use the tag-cloud to search for a username instead of typing it (see figure 3-5). This significantly reduces the button presses – as a single button push is equivalent to entering the full text string of the user name (see section 4.3).

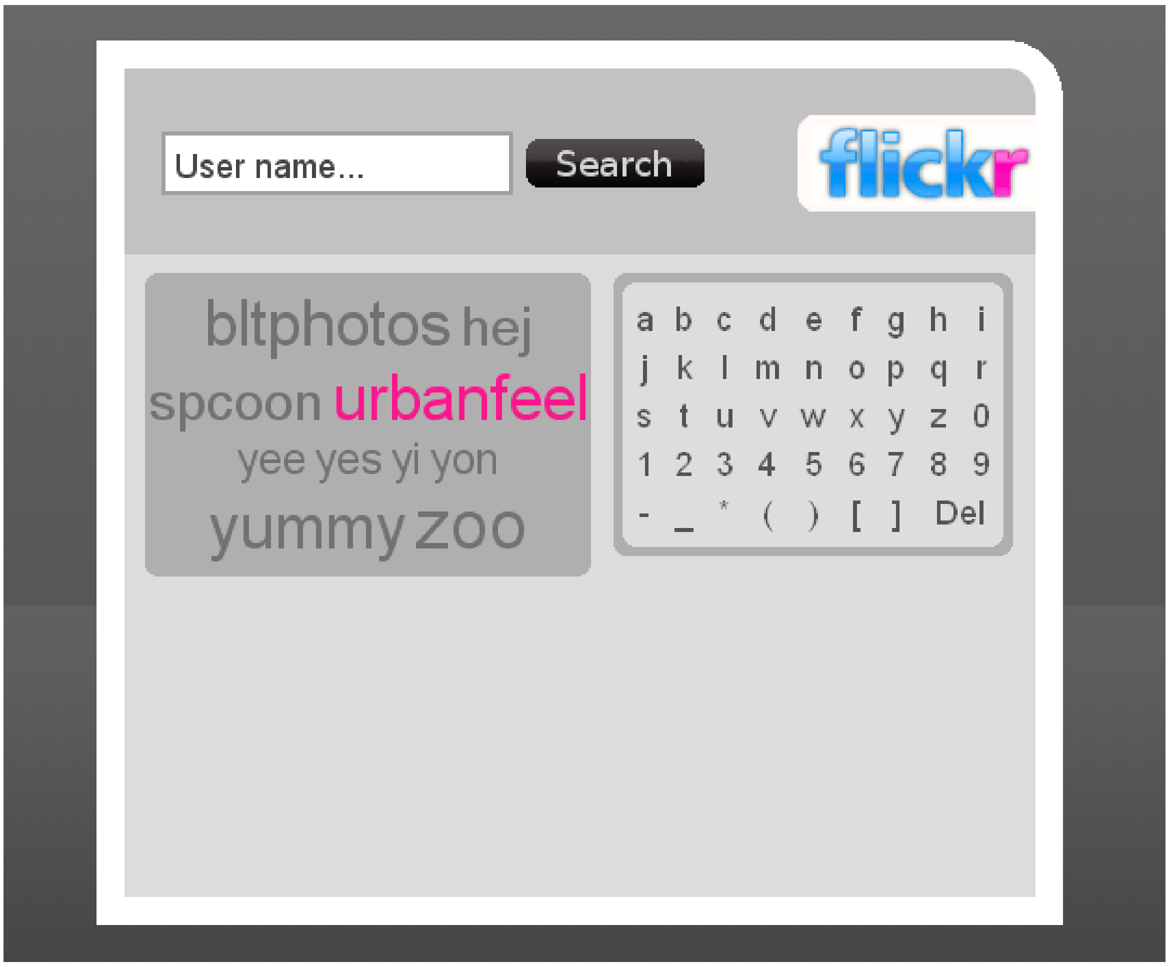


Figure 3-5: Using the tag-cloud to find pictures

If the user wants to type using the virtual keyboard an auto-suggest feature will offer suggested expansions by listing usernames which have the same prefix as the user has already entered. The user can choose one of the suggested usernames with the numeric keys 1 to 5. Figure 3-6 shows that the dropdown list, which present the suggestions, have a transparent background to enable the user to see the whole tag-cloud even when there are the maximum number (five) of alternatives. We have tried placing the tag-cloud and keyboard at other positions, but after discussions with a graphics designer at Accedo Broadband we ended up with this design. This design works well, because when you are typing you are not interested in the tag-cloud, so it does not matter that a part of the tag-cloud is hidden.

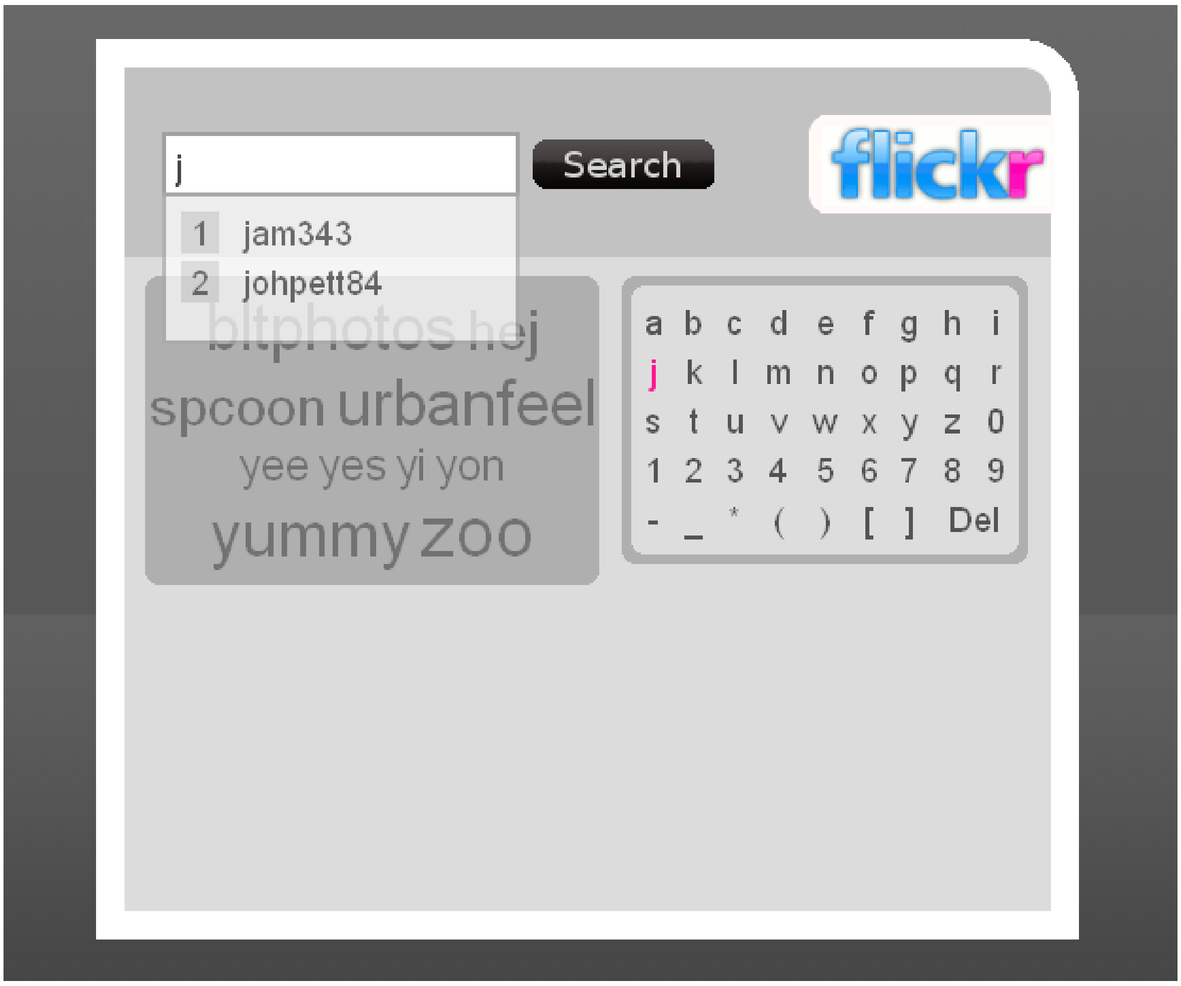


Figure 3-6: Typing with the auto-suggest feature

After entering a username that exists, the 12 latest pictures of that user will be displayed as thumbnails in the bottom of the page. The user can now navigate, with the left and right arrow buttons on the remote control, to focus on the thumbnail that he or she is interested in, after pressing the OK button the image will be shown on the full-screen (as shown in figure 3-7).

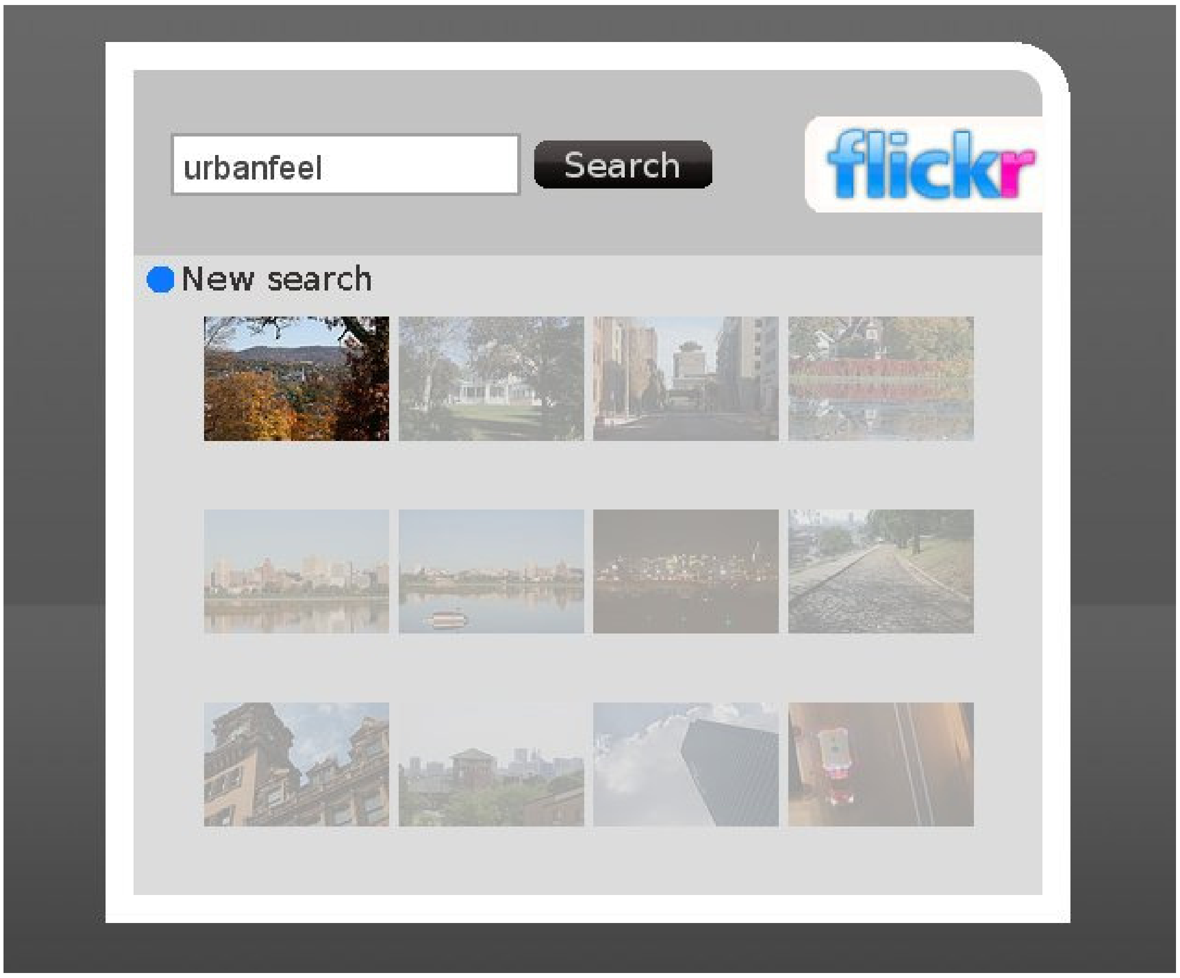


Figure 3-7: Showing the thumbnails

When a user presses the OK button while focus is on one of the thumbnails, that picture will be displayed in full screen with a black background and a white border see figure 3-9. He or she can press left or right key to change to another picture without leaving the full-screen mode.

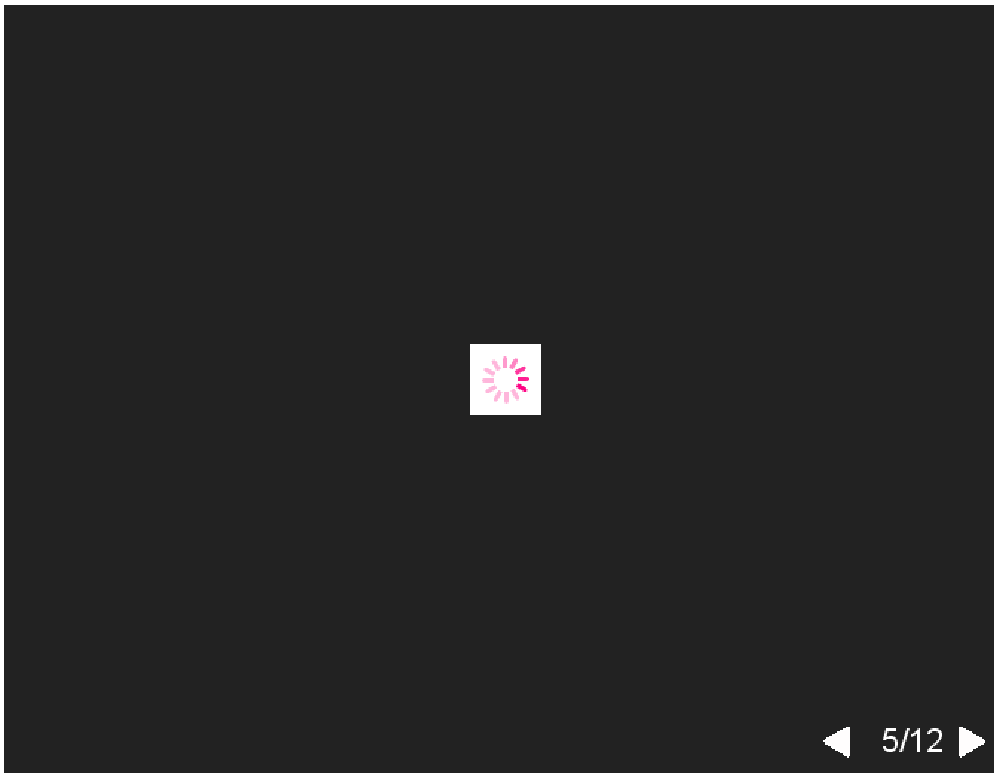


Figure 3-8: Preloader image while loading full-screen picture

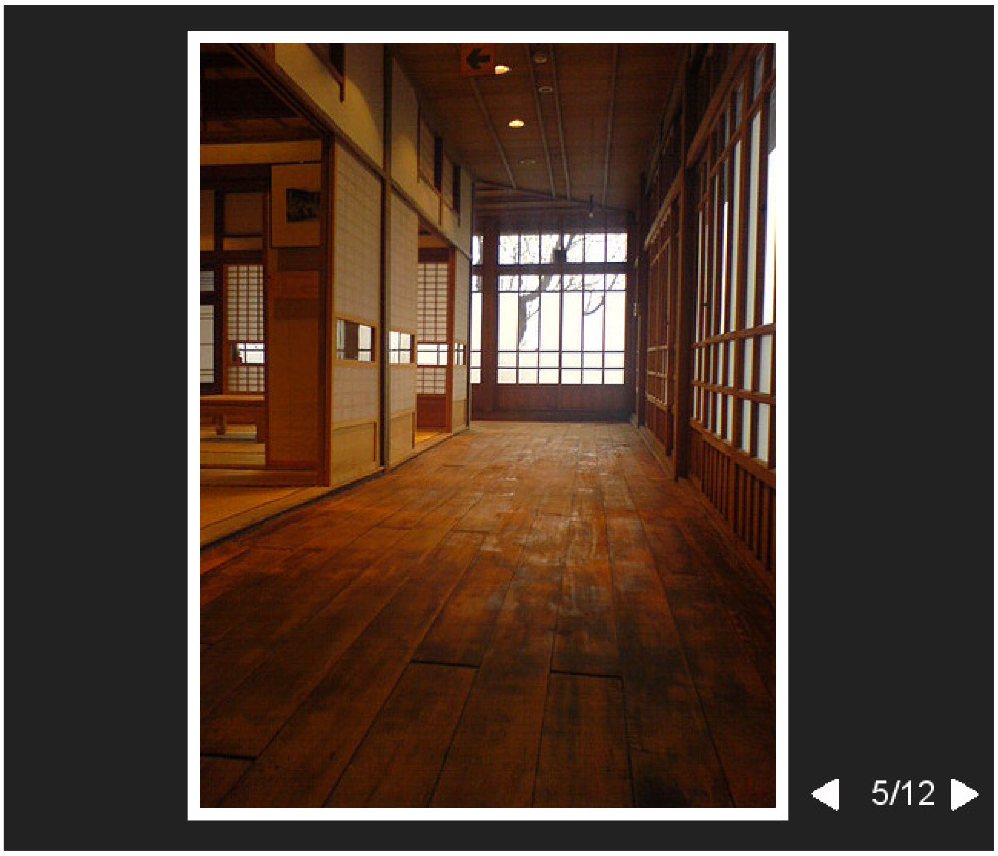
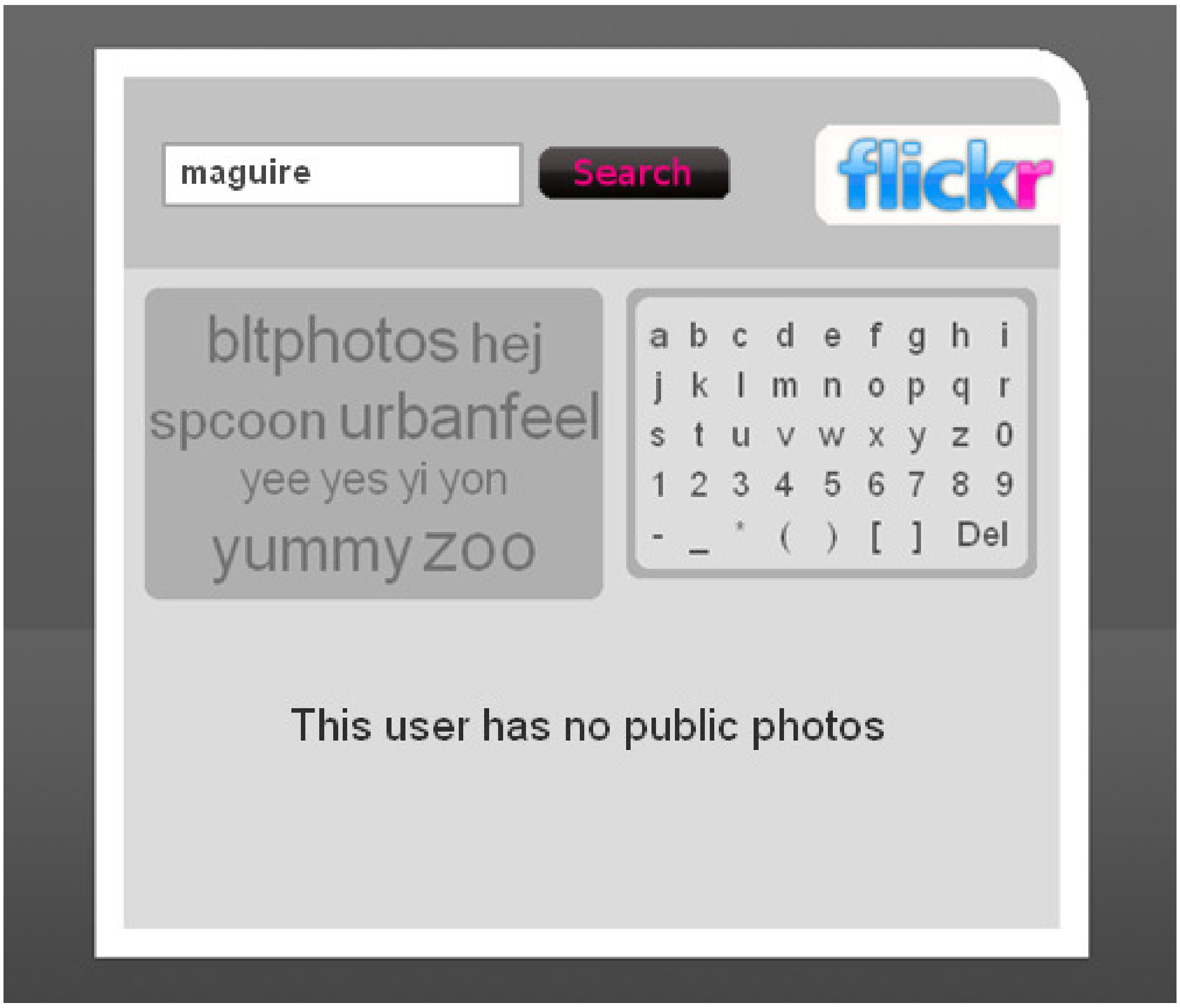


Figure 3-9: A picture in full-screen

If a username that does not exist is entered in the search field, then the user will be informed by displaying a string “User not found”. Some Flickr users do not allow all users to view their photos, thus if someone searches for such a user and this user is not one of the users allowed to see these photos we display the text “This user has no public photos”.



This user has no public photos

Figure 3-10: A user with no public photos

3.2. Twingly

Twingly (http://www.twingly.com/help) does not have a public API for their services, but we received a copy directly from Twingly's system developer Oskar Skoog. Twingly uses a URL as request format (see code example 3-3) and JSON for the response format (see code example 3-4). For this web application we reused some code that we had developed for our Flickr application, thus saving some time. Twingly's response in JSON has some limitations, as it only contains at most ten blogs. Twingly does not use an authentication key. The JSON response was parsed in the same way for Flickr.

Code Example 3-3: An example URL request

```
http://www.twingly.com/json/search?q=”obama”
```

Code Example 3-4: Twingly JSON response

```
{
  "totalFound":236005,
  "items":[
    {
      "id":"9667003650863044375",
      "summary":"Hej alla glada !F&#246;rsta bilden &#228;r vad jag
       hade p&#229; mig idag, de resterande tv&#229; &#228;r fr&#229;n
       f&#246;rra veckan, (n&#228;r jag var hos mamma). Skolan idag, kortaste
       dage...",
      "url":"http://baraeellen.devote.se/blogs/559399/today---.htm",
      "pubDateMs":1220446879000,
      "title":"Today '",
      "websiteName":"baraeellen - Devote.se",
      "websiteUrl":"http://baraeellen.devote.se/",
      "websiteRssUrl":"http://baraeellen.devote.se/index.rss",
      "inlinks":"0",
      "likes":"0",
      "blogoscopeX":null,
      "blogoscopeY":null,
      "tags":[]
    },
    {
     ... etc. ...
    }
}
```

The information which is found in a Twingly JSON response can contain the following information:

totalFound	- Number of blogs found
items	- Contains an array with ten blogs

Items contain this info:

id	- The blog id
summary	- A short summary of the blog
url	- The URL address to the blog
pubDateMs	- The publish date in milliseconds since 1970. It can easily be converted in JavaScript
title	- Title of the blog
websiteName	- Name of the blogs site
websiteUrl	- URL to the site that hold that blog
websiteRssUrl	- RSS URL to the blog
inlinks	- Number of inlinks (other pages that links to this blog)
tags	- Contains an array with tags that the blog have been tagged with

As seen in Figure 3-11 our Twingly interface is quite similar in design to our Flickr application. The functionality is also quite similar; we search for, and then present a summary of information in a simple way. The design is modular and based on `<div>` elements in different layers, formatted through CSS files.

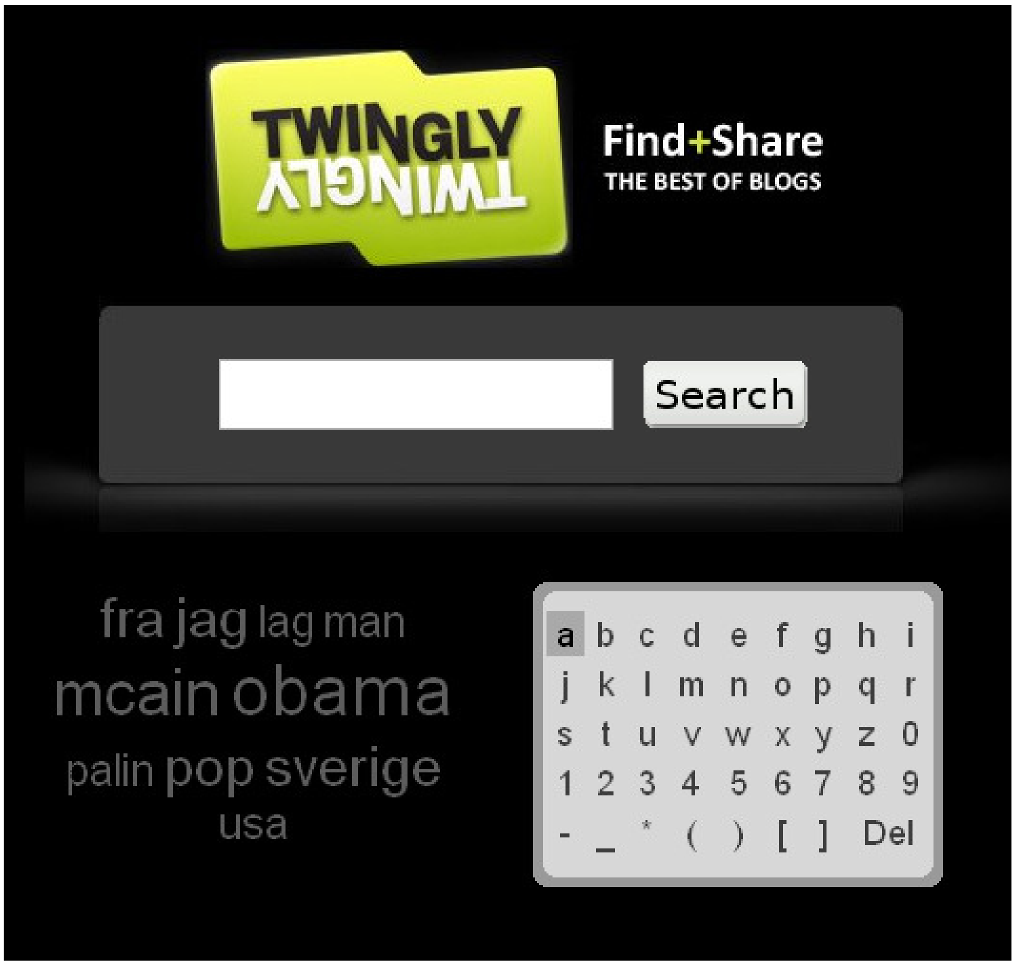


Figure 3-11: The index of our Twingly application

Note that the "Search" button has changed color since it was selected, in order to request the results of searching for “Obama”.

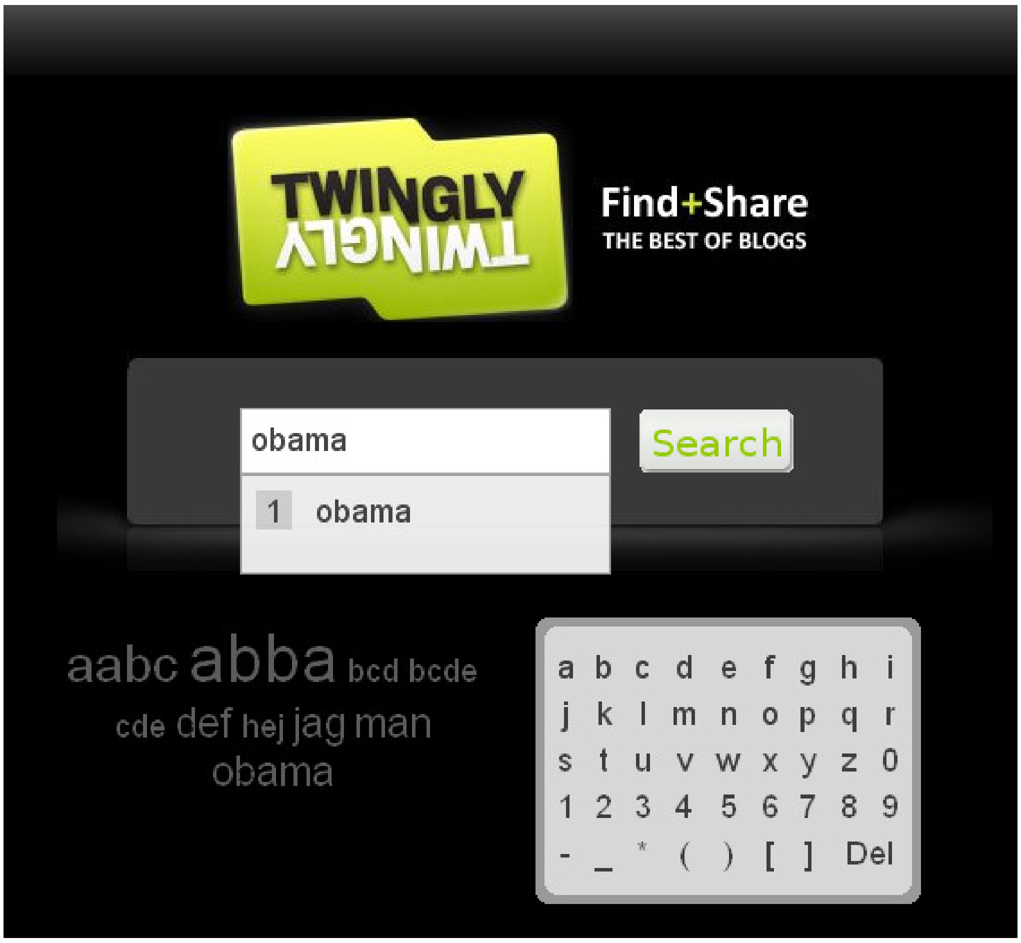
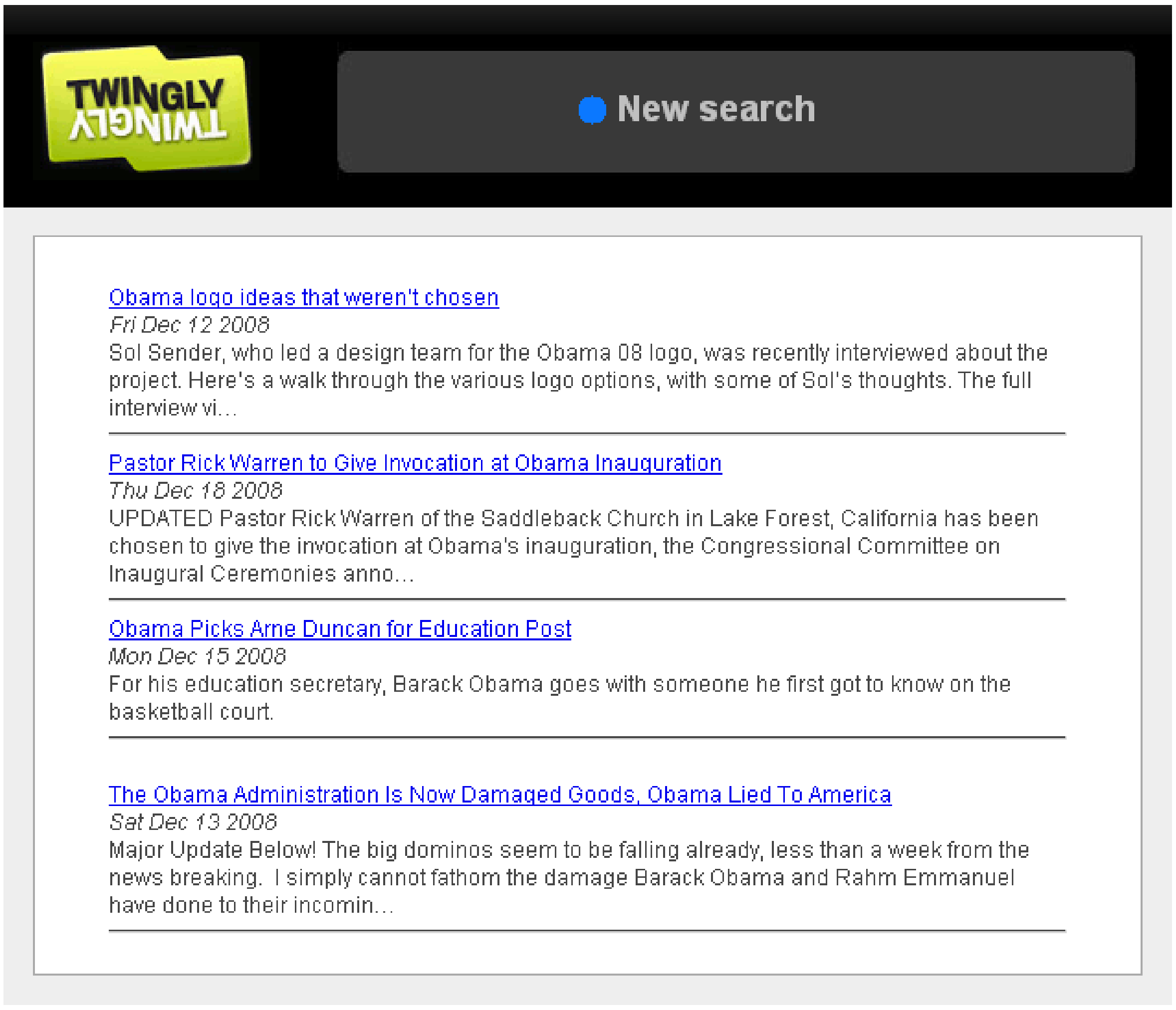


Figure 3-12: Searching with our Twingly application

In this application we present a summary of the blogs that Twingly's server returns in a very simple fashion, separated by horizontal lines.



Obama logo ideas that weren't chosen
Fri Dec 12 2008
Sol Sender, who led a design team for the Obama 08 logo, was recently interviewed about the project. Here's a walk through the various logo options, with some of Sol's thoughts. The full interview vi...

Pastor Rick Warren to Give Invocation at Obama Inauguration
Thu Dec 18 2008
UPDATED Pastor Rick Warren of the Saddleback Church in Lake Forest, California has been chosen to give the invocation at Obama's inauguration, the Congressional Committee on Inaugural Ceremonies anno...

Obama Picks Arne Duncan for Education Post
Mon Dec 15 2008
For his education secretary, Barack Obama goes with someone he first got to know on the basketball court.

The Obama Administration Is Now Damaged Goods, Obama Lied To America
Sat Dec 13 2008
Major Update Below! The big dominos seem to be falling already, less than a week from the news breaking. I simply cannot fathom the damage Barack Obama and Rahm Emmanuel have done to their incomin...

Figure 3-13: Publishing the blog summaries

3.3. Wikipedia

Wikipedia (http://www.mediawiki.org/wiki/API) provides an API, but it does not provide the developer with as much interesting and useful data as the two previous services do. Wikipedia utilizes URL requests and responds with one of the following formats: JSON, XML, PHP, WDDX, or YAML. The problem is that the responds do not contain the data we are interested in, you can for example try if a search string is valid, or check the id of a page, but you can not get a summary of an article or the images related to the text (which is of most interest in the scope of this thesis).

Code Example 3-5: Wikipedia URL request

```
http://en.wikipedia.org/w/api.php?
     action=query&
     titles=Albert%20Einstein&
     prop=info&
     format=jsonfm
```

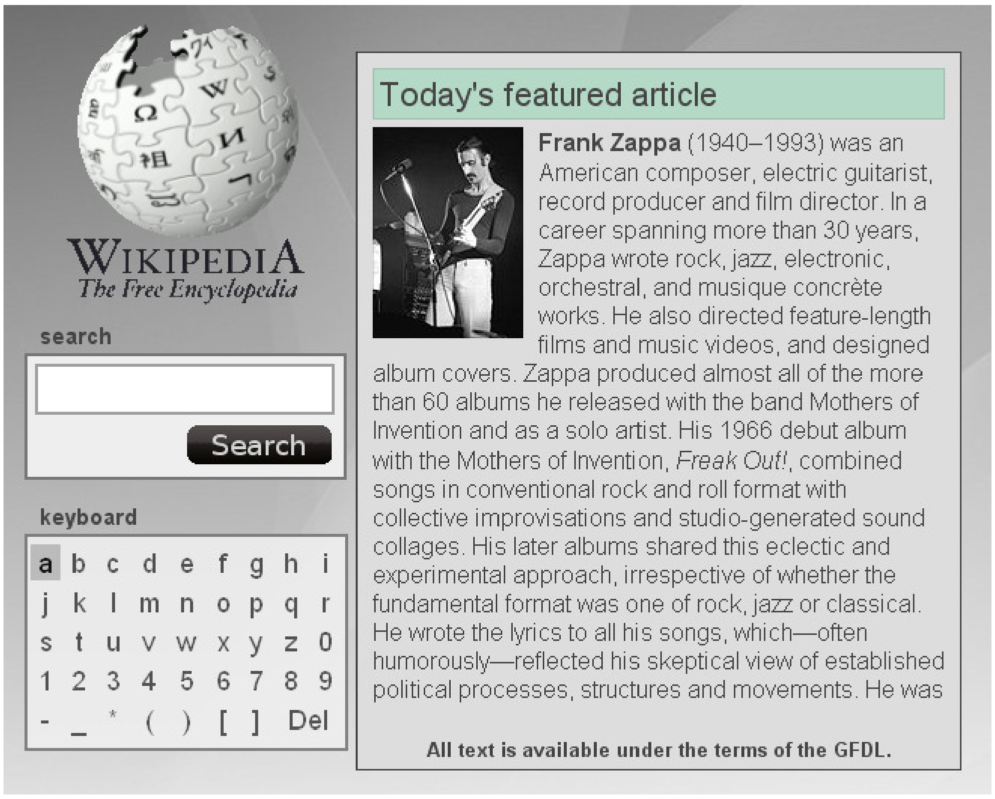
Code Example 3-6: Wikipedia JSON response

```
{
     "query": {
          "pages": {
               "736": {
                    "pageid": 736,
                    "ns": 0,
                    "title": "Albert Einstein",
                    "touched": "2008-12-08T06:24:14Z",
                    "lastrevid": 255695230,
                    "counter": 4698,
                    "length": 91520
               }
          }
     }
}
```

Due to the lack of functions in the Wikipedia API we have decided to create our own interface to this application by parsing the data received from Wikipedia's servers our selves. The content published at Wikipedia, is available under the GNU Free Document Licence (GFDL). An exception to this license is the statement "The only contents about the use of which you should contact the Wikimedia Foundation are the trademarked Wikipedia/Wikimedia logos" that is found on Wikipedia's Copyrights page. The GFDL require the following actions when copied [69]:

- Your materials in turn have to be licensed under GFDL,
- You must acknowledge the authorship of the article (section 4B), and
- You must provide access to the "transparent copy" of the material (section 4J). (The "transparent copy" of a Wikipedia article is any of a number of formats available from Wikipedia, including the wiki text, the HTML web pages, XML feed, etc.)

When starting the application the user will see a page similar to Figure 3-14. This page includes the possibility to search for content and it publishes "Today's featured article".



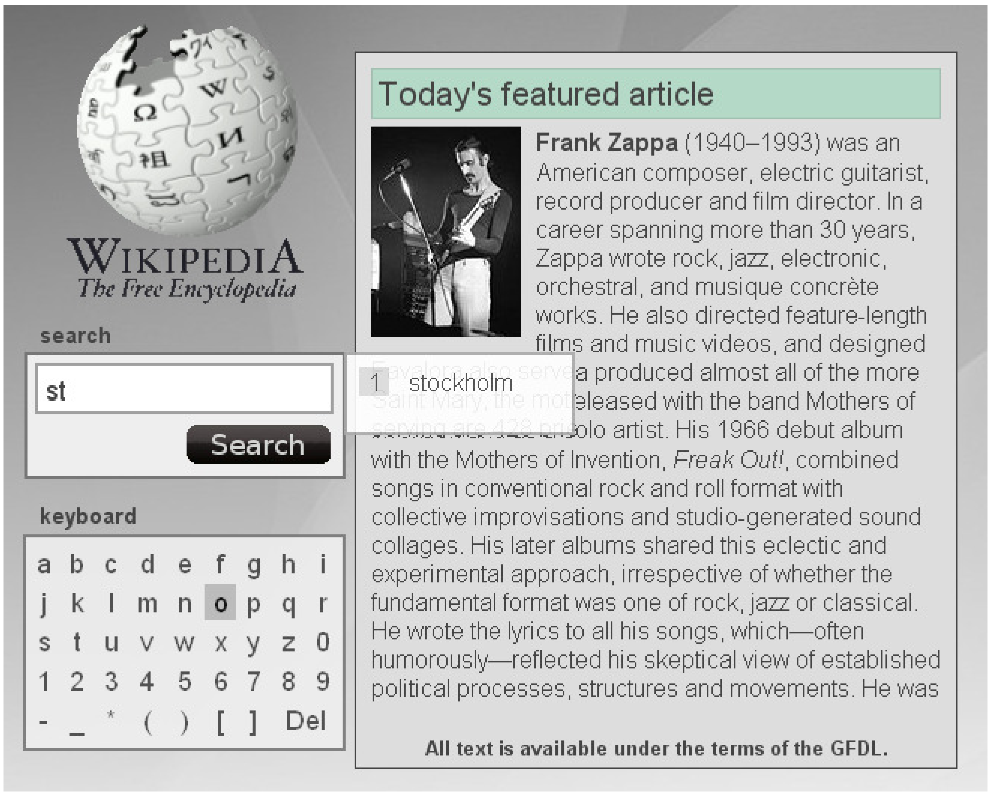
Today's featured article

Frank Zappa (1940–1993) was an American composer, electric guitarist, record producer and film director. In a career spanning more than 30 years, Zappa wrote rock, jazz, electronic, orchestral, and musique concrète works. He also directed feature-length films and music videos, and designed album covers. Zappa produced almost all of the more than 60 albums he released with the band Mothers of Invention and as a solo artist. His 1966 debut album with the Mothers of Invention, *Freak Out!*, combined songs in conventional rock and roll format with collective improvisations and studio-generated sound collages. His later albums shared this eclectic and experimental approach, irrespective of whether the fundamental format was one of rock, jazz or classical. He wrote the lyrics to all his songs, which—often humorously—reflected his skeptical view of established political processes, structures and movements. He was

All text is available under the terms of the GFDL.

Figure 3-14: Wikipedia application start page

When the user starts to type in a search string a suggestion will appear (as seen in figure 3-15).



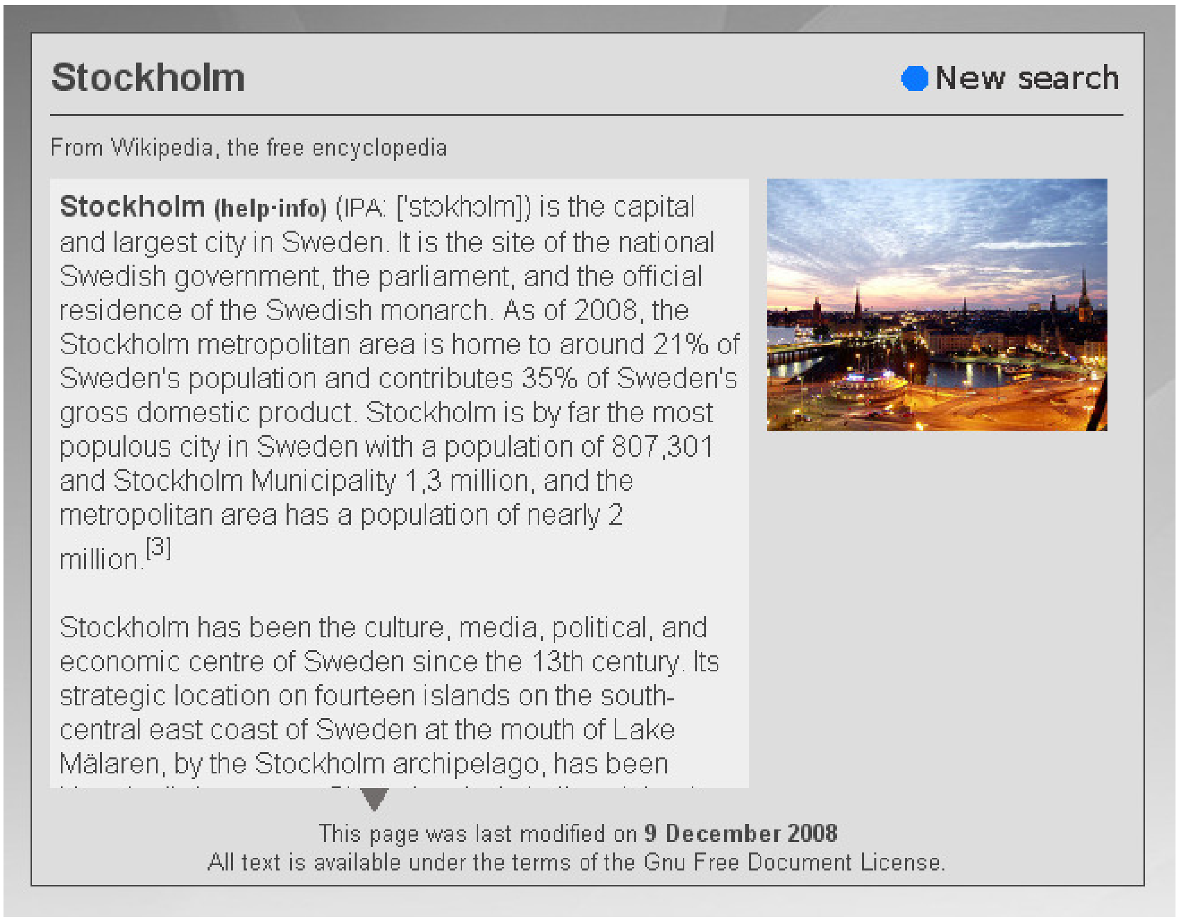
Today's featured article

Frank Zappa (1940–1993) was an American composer, electric guitarist, record producer and film director. In a career spanning more than 30 years, Zappa wrote rock, jazz, electronic, orchestral, and musique concrète works. He also directed feature-length films and music videos, and designed a produced almost all of the more eleased with the Mothers of olo artist. His 1966 debut album with the Mothers of Invention, *Freak Out!*, combined songs in conventional rock and roll format with collective improvisations and studio-generated sound collages. His later albums shared this eclectic and experimental approach, irrespective of whether the fundamental format was one of rock, jazz or classical. He wrote the lyrics to all his songs, which—often humorously—reflected his skeptical view of established political processes, structures and movements. He was

All text is available under the terms of the GFDL.

Figure 3-15: Wikipedia application searching

The user can now choose to press the "1" key continue using the virtual keyboard to generate a search string. The result will be an article such as shown in Figure 3-16. To maximize the content that can be displayed we have removed the logotype and other unnecessary elements. In order to make a new search (e.g. to get back to the start page) the user will have to press the blue key on the remote control. We do also present the first image, if there are any, of the article, when the last edit took place and some legal terms (these are required - as mentioned earlier in this section). To scroll down and read more of the article (as we have chosen to only present the summary of each article) we use a YUI JavaScript library function to change which part of the `<div>` (including the content) that is currently visible in an animated way (similar to using the vertical scroll bar in a normal web browser). This is triggered when the user presses the down arrow on the remote control.



Stockholm (help·info) (IPA: ['stokholm]) is the capital and largest city in Sweden. It is the site of the national Swedish government, the parliament, and the official residence of the Swedish monarch. As of 2008, the Stockholm metropolitan area is home to around 21% of Sweden's population and contributes 35% of Sweden's gross domestic product. Stockholm is by far the most populous city in Sweden with a population of 807,301 and Stockholm Municipality 1,3 million, and the metropolitan area has a population of nearly 2 million.[3]

Stockholm has been the culture, media, political, and economic centre of Sweden since the 13th century. Its strategic location on fourteen islands on the south-central east coast of Sweden at the mouth of Lake Mälaren, by the Stockholm archipelago, has been

This page was last modified on **9 December 2008**
All text is available under the terms of the Gnu Free Document License.

Figure 3-16: Wikipedia application article

3.4. IMDb

IMDb does not provide an open API that is suitable for IPTV. In order to use IMDb, the developer has to fulfill distinct license requirements in order to be able to use these services and their content for non-commercial use. In other cases it is mandatory to have an agreement (including paying a license fee) between the original provider and the IPTV content provider. IMDb's lack of an open API has been a well debated issue (for the last few years) for both major Internet personalities, such as Tim O'Reilly [70] and less famous developers. According to IMDb's website [71] they are currently searching for a software developer with "Evidence of excellent API design and architecture skills and an understanding of best-practises". This may indicate that they are moving towards creating an open API. Until they do, there are several other companies and programmers that provide their own IMDb APIs under the GNU Public License, but it seems unclear if they actually have the legal rights to do so. As an example, IMDBPHP [72] presents movie covers which IMDb explicit does not allow [73]. Another IMDb service provider is Trynt:Tech [74], if you use the Trynt API, then you need to say that you use their web services. However, IMDb does allow developers to download much of their database, thus it is possible to search and present their contents without loading their servers. IMDb states [75]: "Please refer to the copyright/license information listed in each file for instructions on allowed usage. The data is NOT FREE although it may be used for free in specific circumstances." IMDb does not state what this means and suggests that you should contact their license department if you want to use their data for other use than strictly personal.

Many of the service providers without open APIs do not accept data mining, screen scraping, or similar methods for retrieving and parsing their content, especially IMDb [76].

The code example 3-7 shows how to search for the IMDb id representing the movie title "Dogma". The response sent by Trynt's API for such a request is shown in code example 3-8. The interesting part of this XML response is the text "tt0120655" which is later user to get the actual information about the movie from IMDb's website.

Code Example 3-7: Trynt IMDb URL request

```
http://www.trynt.com/movie-imdb-api/v2/?t=dogma
```

Code Example 3-8: Trynt IMDb XML response

```
<trynt>
     <movie-imdb>
          <search>dogma</search>
          <matched-id>tt0120655</matched-id>
          <matched-url>http://www.imdb.com/title/tt0120655/</matched-url>
          <matched-title>Dogma (1999)</matched-title>
          <data>cache</data>
     </movie-imdb>
     <server>10.254.95.50</server>
</trynt>
```

When starting the application the user will see a page as in Figure 3-17. This page includes the possibility to search for content through a virtual keyboard or a tag-cloud.

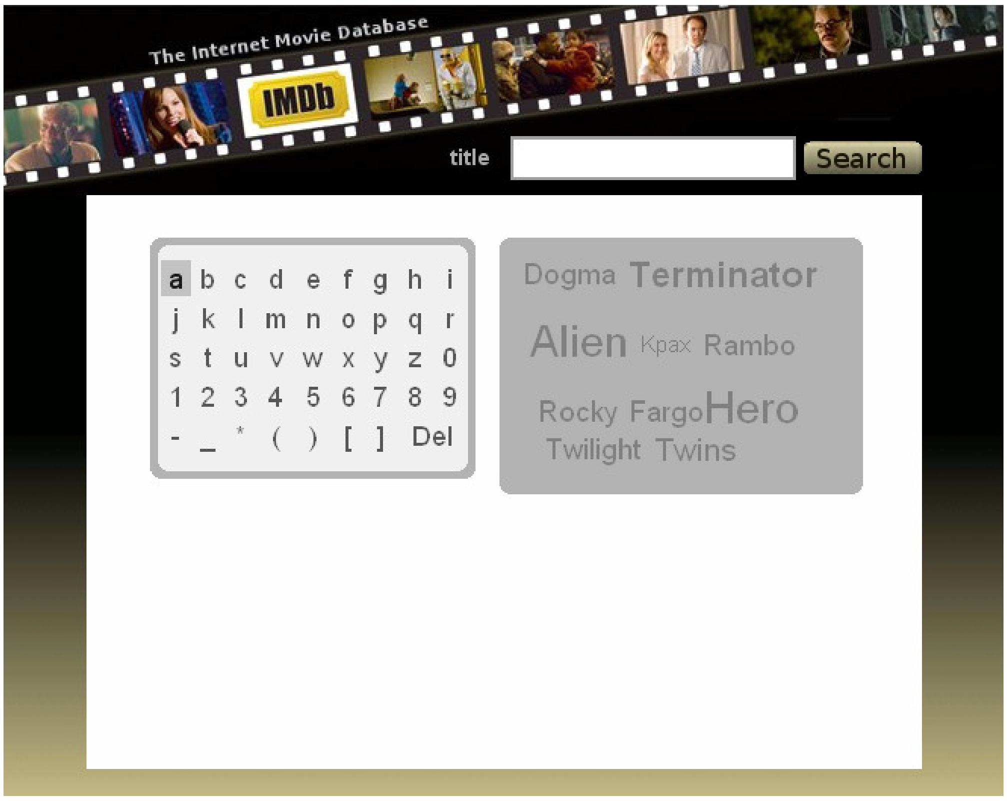


Figure 3-17: IMDb main page

When the user starts to type with the virtual keyboard an auto-suggest window will appear, including a suggestion based on entities in our database of earlier searched movie titles.

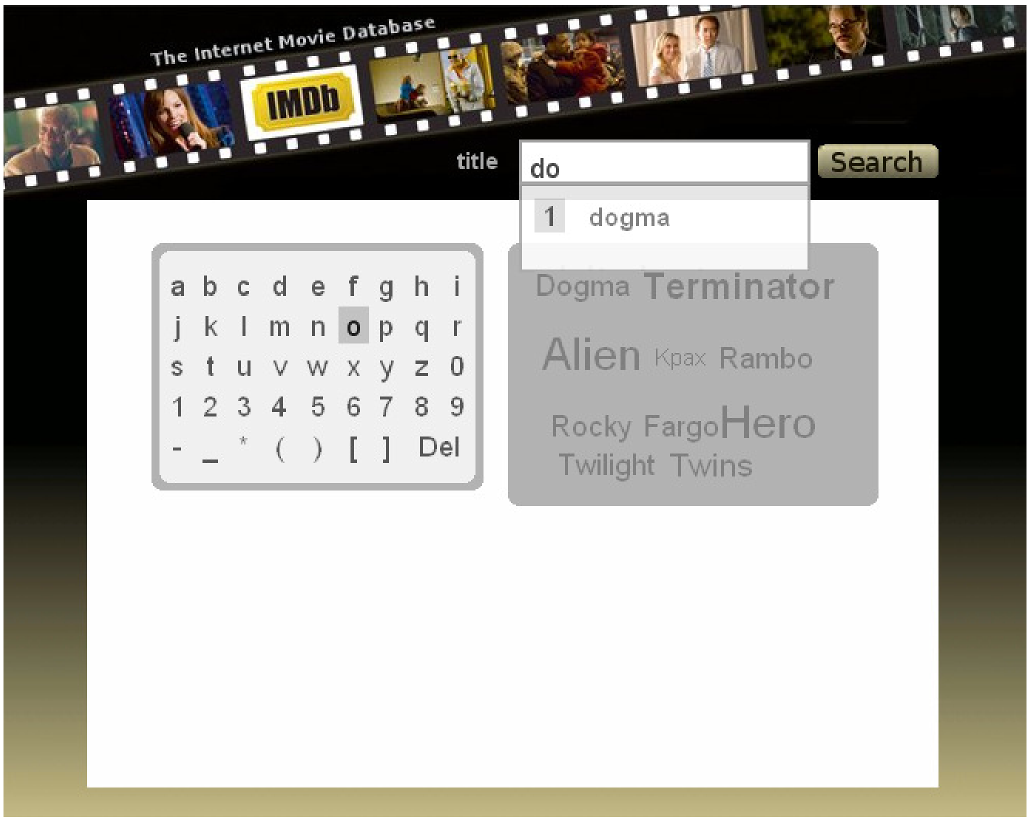
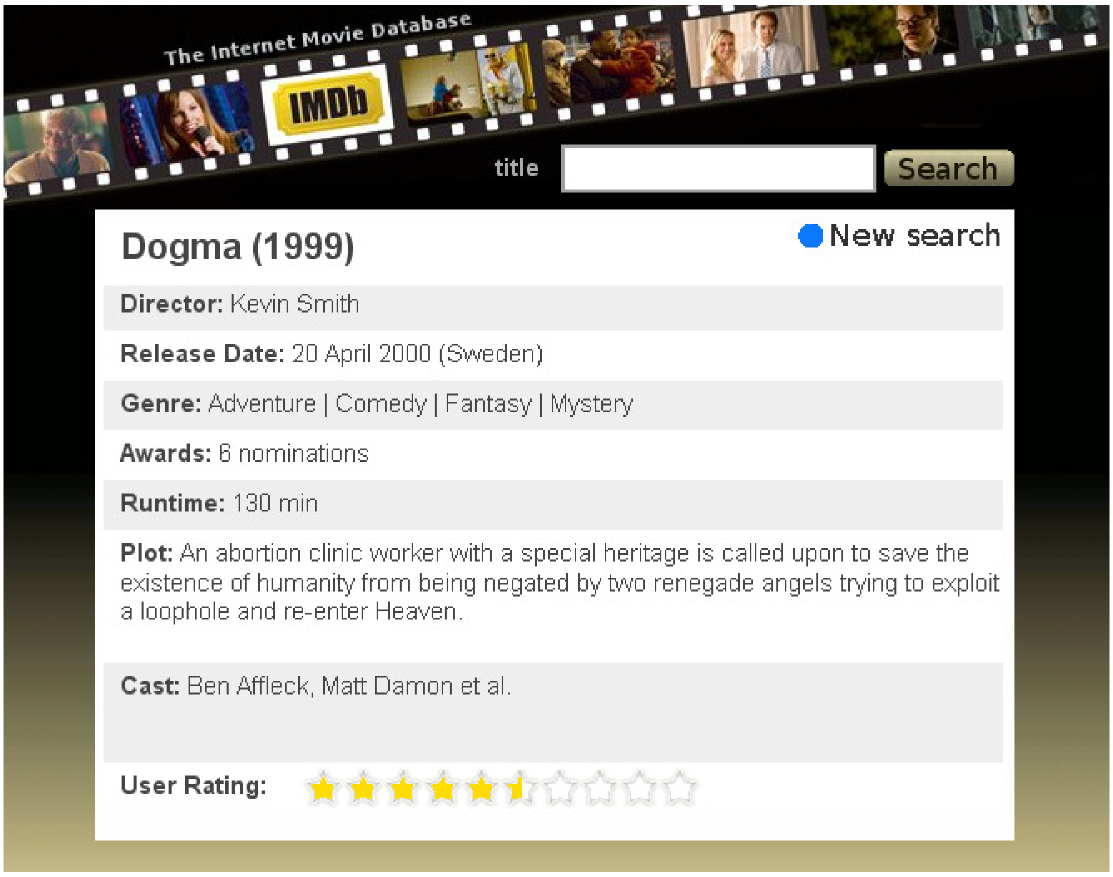


Figure 3-18: IMDb Typing

The final page is presenting basic facts about the movie. We use our own PHP function for printing a yellow bar, based on the user rating (in the case of Dogma it is 5.5), and have a transparent image with ten stars placed above to be able to illustrate the rating graphically.



Dogma (1999)

Director: Kevin Smith

Release Date: 20 April 2000 (Sweden)

Genre: Adventure | Comedy | Fantasy | Mystery

Awards: 6 nominations

Runtime: 130 min

Plot: An abortion clinic worker with a special heritage is called upon to save the existence of humanity from being negated by two renegade angels trying to exploit a loophole and re-enter Heaven.

Cast: Ben Affleck, Matt Damon et al.

User Rating:

Figure 3-19: IMDb movie information

3.5. Other services with open APIs

There are a number of other services with open APIs. Thus far, we have looked at EBay, YouTube, and Facebook. Their APIs offer a lot of support and help and are quite user friendly for developers.

3.5.1. EBay

With the EBay API [77] you can:

- Submit items for listing on eBay
- Get the current list of eBay categories
- View information about items listed on eBay
- Get high bidder information for items you are selling
- Retrieve lists of items a particular user is currently selling through eBay
- Retrieve lists of items a particular user has bid on
- Display eBay listings on other sites
- Leave feedback about other users at the conclusion of a commerce transaction

In code example 3-9 you can see the URL request for making a search for iPod and we want the response to be in XML. In code example 3-10 is the XML response file for an iPod skin case with detailed EBay info about the object.

Example code 3-9: EBay URL request

```
http://open.api.ebay.com/shopping
        ?callname=FindItems
        &version=517
        &siteid=0
        &appid=XXX
        &QueryKeywords=iPod
        &responseencoding=XML
```

Example code 3-10: EBay XML response

```
<FindItemsResponse>
        <Timestamp>2008-11-11T13:34:45.327Z</Timestamp>
        <Ack>Success</Ack>
        <Build>e589_core_Bundled_7495152_R1</Build>
        <Version>589</Version>
        <Item>
                <ItemID>110307863757</ItemID>
                <EndTime>2008-11-11T13:34:49.000Z</EndTime>
                <ViewItemURLForNaturalSearch>
                        http://cgi.ebay.com/Silicone-Skin-Case-for-New-Apple-iPod-Touch-
                        Black_W0QQitemZ110307863757QQcategoryZ56170QQcmdZViewItem
                </ViewItemURLForNaturalSearch>
                <ListingType>FixedPriceItem</ListingType>
                <GalleryURL>
                        http://thumbs1.ebaystatic.com/pict/1103078637578080_1.jpg
                </GalleryURL>
                <PrimaryCategoryID>56170</PrimaryCategoryID>
                <PrimaryCategoryName>
                        Electronics:iPod & MP3 Accessories:For iPod:Cases
                </PrimaryCategoryName>
                <ConvertedCurrentPrice currencyID="USD">3.98</ConvertedCurrentPrice>
                <ListingStatus>Active</ListingStatus>
                <TimeLeft>PT4S</TimeLeft>
                <Title>
                        Silicone Skin Case for New Apple iPod Touch (Black)
                </Title>
                <ShippingCostSummary>
                        <ShippingServiceCost currencyID="USD">0.0</ShippingServiceCost>
                        <ShippingType>Flat</ShippingType>
                </ShippingCostSummary>
        </Item>
        <Item>
                ... Etc. …
        </Item>
        <TotalItems>104437</TotalItems>
        <ItemSearchURL>
                http://search.ebay.com/ws/search/SaleSearch?fsoo=1&fsop=1&satitle=iPod
        </ItemSearchURL>
</FindItemsResponse>
```

3.5.2. YouTube

The YouTube API [78] provides the ability to retrieve feeds related to videos, users, and playlists. It also provides the ability to manipulate these feeds, such as creating new playlists, adding videos, to favorites, uploading videos and sending messages.

In example code 3-11 you can see the URL request for making a search for dancing ordered by number of tags. Example code 3-12 shows the XML response file with detailed YouTube information about the video.

Example code 3-11: YouTube URL request

```
http://www.youtube.com/api2_rest
        ?method=youtube.videos.list_by_tag
        &tag=dancing
        &per_page=10
        &dev_id=XXX
        &page=1
```

Example code 3-12: YouTube XML response

```
<video_list>
        <video>
                <author>youtuberocks</author>
                <id>k0gEeue2sLk</id>
                <title>My First Motion Picture</title>
                <length_seconds>16</length_seconds>
                <rating_avg>3.75</rating_avg>
                <rating_count>10</rating_count>
                <description>This is the video description shown on the YouTube site.</description>
                <view_count>170</view_count>
                <upload_time>1121398533</upload_time>
                <comment_count>1</comment_count>
                <tags>feature film documentary</tags>
                <url>http://www.youtube.com/watch?v=k04Eeue24Lk</url>
                <thumbnail_url>
                        http://static.youtube.com/get_still?video_id=k04Eeue24Lk
                </thumbnail_url>
                <embed_status>ok</embed_status>
        </video>
</video_list>
```

3.5.3. Facebook

The Facebook API [79] offers quite a good API, but it is not as user friendly as the API of either EBay or YouTube. The Facebook API provides the developer with the ability to create Facebook applications, find user information, get/modify photos, create/modify events, etc. For example the video providing site Joost offers the possibility to log on to Facebook through their website and thus be able to share events and logging your friends feed directly.

4. Analysis

We have decided to analyze our applications by measuring the memory consumption-, response time-, and execution time. We have compared different search methods to see the number of key presses it takes and the code reuse by counting lines of common code the applications has. We have compared our virtual alphabetic keyboard with a standard qwerty-keyboard to evaluate the movement efficiency. Finally we have analyzed our design to see if it fit the IPTV environment.

4.1. Memory consumption

As the Motorola STB provides us with the possibility to measure how much free memory it has, if we compare the amount we have before and after different actions we see how much memory that particular action consumes. Tables 4-1 to 4-4 shows how much of the Motorola STB's memory usage (see memory usage column) is consumed compared to the memory in use at the portal by using the different search methods (Auto-suggest, Tag-cloud or through keyboard). Note that the 1720 STB runs a modified version of the Mozilla 1.5 [80] web browser. After running several applications we have noticed increased memory consumption thus it is clear that the garbage collection is not well optimized - this is a well known problem for Mozilla browsers [81]. Unfortunately, there are no methods to invoke garbage collection in JavaScript because everything is handled automatically [82]. The rightmost column presents the average difference in memory consumption of each function (as computed by the difference between the memory available when using the function and the memory consumption when reaching the main portal). The averages are computed over 5 repetitions of the measurements. Wikipedia web service consumes most memory because it has to parse a lot of data and process a number of pictures before it can display our page. You can also see in section 4.2 that the execution time is longer than the other services.

Table 4-1: Memory consumption for Flickr compared to the portal (measured in kB)

	Test 1	Test 2	Test 3	Test 4	Test 5	Avg. Memory usage
Portal	0	0	0	0	0	**-**
Flickr index	700	792	156	156	712	**503,2**
Auto-suggest	732	792	176	176	812	**537,6**
Tag-cloud	800	856	388	388	820	**650,4**
Show picture	752	1016	332	332	864	**659,2**
Switch picture	752	1016	388	388	780	**503,2**

Table 4-2: Memory consumption for Twingly compared to the portal (measured in kB)

	Test 1	Test 2	Test 3	Test 4	Test 5	Avg. Memory usage
Portal	0	0	0	0	0	**-**
Twingly index	-24	96	124	124	-16	**61**
Auto-suggest	56	100	188	188	-4	**106**
Tag-cloud	-4	132	220	220	48	**123**
Keyboard	160	136	264	264	4	**166**

Table 4-3: Memory consumption for Wikipedia compared to the portal (measured in kB)

	Test 1	Test 2	Test 3	Test 4	Test 5	Avg. Memory usage
Portal	0	0	0	0	0	**0**
Wikipedia index	4084	2376	3236	3236	2352	**3056,8**
Auto-suggest	4160	2416	3392	3392	2344	**3140,8**
Keyboard	4144	2516	3408	3408	2328	**3160,8**

Table 4-4: Memory consumption for IMDb compared to the portal (measured in kB)

	Test 1	Test 2	Test 3	Test 4	Test 5	Avg. Memory usage
Portal	0	0	0	0	0	**0**
IMDb index	76	24	-20	-20	-28	**6,4**
Auto-suggest	168	52	48	48	-88	**45,6**
Tag-cloud	200	84	56	56	48	**88,8**
Keyboard	204	96	72	72	59	**100,6**

We have also compared the values for Flickr, Twingly, Wikipedia, and IMDb with the memory consumption of popular games at Accedo. Table 4-5 shows the memory consumption of the worst case in our applications and two popular JavaScript games. Kaboom! is a game similar to the popular Minesweeper (installed with most versions of Microsoft's Windows operating system). Memory is a mind game where you need to remember and find matching pairs. As seen in the table, our applications use quite little memory compared to even the simplest games.

Table 4-5: Memory consumption for different applications (measured in kB)

Service	Memory usage
Flickr	3123
Twingly	264
Kaboom!	5019
Memory	6246

While using a "Modal-window" (see section 2.3.3) for our full screen pictures in Flickr, we observed that all the different `<div>` parts were not faded simultaneously, resulting in an awful and slow feeling while clicking the thumbnails to get a full-size picture. However, the results in table 4-6 shows that this behavior is **not** due to the transparent background consuming more memory than a plain black colored one.

Table 4-6: Memory diffrence for transparent and black backgrounds in Flickr (measured in kB)

	Difference in memory consumption
Test 1	128
Test 2	96
Test 3	68
Test 4	328
Test 5	132
Test 6	416
Test 7	-72
Test 8	348
Test 9	-8
Test 10	292
Average	**173**

4.2. Response time

We used a desktop PC[2] and the 1720 STB with Wireshark 1.0.4 [Q] to measure the response time for our GET requests to several services. We wanted to examine if there was a big difference running the services on the STB or PC. We measured the time from sending a GET request to we got the response back. In tables 4-7 and 4-8 we see that the Flickr library method used to convert a username to a user-id consumes hundreds of milliseconds more time for each request than the method used to get URLs to that user's public photos which we expected to take the similar amount of time. Table 4-9 presents the equivalent timings for Twingly. You can see the difference in the row "Get photo URLs" for Flickr. We have noticed that the response time is between 500 to 1300 ms depending on how many public pictures the user has, because we have to get all the information for all the pictures. If there are more than hundreds of pictures it takes over 1000 ms to retrieve them all. According to the Flickr API there is an option to set a maximum number of pictures for which information will be returned, but when we tried this option, it did not work. In IMDb and Wikipedia the "Get webpage" is the most time consuming because we have to parse all the text before we display any of it.

Table 4-7: Response time of Flickr measured with Wireshark on a PC (ms)

Test #	1	2	3	4	5	6	7	8	9	10	Average
Get user id	569	595	497	583	886	526	548	501	603	569	**587,7**
Get photo URLs	1423	1163	1255	1381	556	562	1347	523	1338	1060	**1060,8**
Update tag-cloud	7	7	8	5	5	9	5	5	7	8	**6,6**

[2] Dell Optiplex 210L (Intel Pentium 4 2.8 GHz CPU and 2 GB DDR2 RAM (533 MHz)) running Microsoft Windows XP Professional SP2.

Table 4-8: Response time of Flickr measured with Wireshark on a 1720 STB (ms)

Test #	1	2	3	4	5	6	7	8	9	10	Average
Get user id	635	602	602	550	595	603	566	621	556	568	**590,0**
Get photo URLs	1233	1263	1205	1389	1626	606	1226	612	530	1270	**1096,0**
Update tag-cloud	1	3	2	5	6	5	7	6	5	3	**4,7**

Table 4-9: Response time of Twingly measured with Wireshark on a PC (ms)

Test #	1	2	3	4	5	6	7	8	9	10	Average
Get blog info	261	272	586	437	621	446	268	262	487	262	**390,2**
Update tag-cloud	4	4	5	5	5	5	5	5	5	5	**4,8**

Table 4-10: Response time of Twingly measured with Wireshark on a 1720 STB (ms)

Test #	1	2	3	4	5	6	7	8	9	10	Average
Get blog info	346	728	496	863	357	605	266	549	379	576	**516,0**
Update tag-cloud	5	5	4	5	5	5	4	5	5	5	**4,8**

Table 4-11: Response time of Wikipedia measured with Wireshark on a PC (ms)

Test #	1	2	3	4	5	6	7	8	9	10	Average
Get index.php	244	278	235	243	285	238	231	240	240	243	**247,7**
Get auto-suggest	4	5	5	5	5	8	4	6	9	4	**5,5**
Get webpage	737	648	726	714	885	725	518	544	979	561	**703,7**
Add/update searchword	5	5	5	5	8	5	7	9	9	6	**6,4**
Get image	104	245	96	231	165	129	157	82	102	59	**137,0**

Table 4-12: Response time of Wikipedia measured with Wireshark on a 1720 STB (ms)

Test #	1	2	3	4	5	6	7	8	9	10	Average
Get index.php	274	297	275	298	271	358	333	291	270	285	**295,2**
Get auto-suggest	4	5	6	4	4	6	5	5	4	5	**4,8**
Get webpage	651	1107	920	890	1067	847	898	853	810	722	**876,5**
Add/update searchword	5	5	5	4	6	6	5	5	4	5	**5,0**
Get image	199	200	148	177	178	355	102	127	135	129	**175,0**

Table 4-13: Response time of IMDb measured with Wireshark on a PC (ms)

Test #	1	2	3	4	5	6	7	8	9	10	Average
Get index.php	8	5	8	9	9	10	9	9	9	9	**8,5**
Get auto-suggest	5	5	4	5	5	5	4	5	5	4	**4,7**
Get webpage	1907	1895	1784	1686	1643	1678	1399	1492	1787	1545	**1681,6**
Add/update searchword	6	7	7	6	5	6	6	7	5	5	**6,0**

Table 4-14: Response time of IMDb measured with Wireshark on a 1720 STB (ms)

Test #	1	2	3	4	5	6	7	8	9	10	Average
Get index.php	8	9	8	8	8	8	7	8	7	7	**7,8**
Get auto-suggest	4	4	5	4	4	4	4	5	4	5	**4,3**
Get webpage	1812	1878	1595	1828	1498	1542	1686	1785	1736	1581	**1694,1**
Add/update searchword	5	5	5	5	5	5	5	5	5	5	**5,0**

4.3. Execution time

The measurements summarised in tables 4-15 to 4-18, were captured through the Firefox plug-in Firebug 1.2.1 [R] and present the execution time when a PC[3] requests the start page from our Apache web server (both machines are located in the same office and connected to the same Internet provider). Thus these measurements compare the different parts of the application, if one function continually needs five times more time than an other to execute on a PC it is likely that there is a similar difference when they run on a STB (altough the exact execution times will probably differ).

As seen in the tables below, most of the execution time is due to processing of the stylesheet files (.css) and the main page (index.php), while the JavaScript files are executed very fast. The reason for the low execution time for JavaScript files is that they include client-side code that mostly only run when the user navigates through the site, rather than running at "start-up" time (when a page is first loaded).

Table 4-15: Execution time for Flickr, measured with Firebug in (ms)

Test #	1	2	3	4	5	6	7	8	9	10	Average
Index.php	20	22	23	23	35	25	37	36	33	26	**28**
Base.css	23	28	12	11	11	10	12	14	10	11	**14**
Keyboard.css	21	32	17	20	22	20	26	21	20	18	**22**
Tagcloud.css	34	38	17	21	22	20	20	25	21	20	**24**
AutoSuggest.css	29	32	22	24	23	25	26	27	28	27	**26**
Action.js	31	30	24	27	26	28	28	29	28	25	**28**
Movement.js	4	4	6	3	3	3	3	3	3	3	**4**
Include_Keyhandler.js	3	5	4	3	3	3	3	4	3	4	**4**
Keyhandler_workstation.js	4	3	4	3	3	4	4	4	3	3	**4**
Keyboard.js	3	5	4	3	3	5	3	3	3	3	**4**
Navigation.js	3	4	3	3	3	3	3	3	4	3	**3**
Ajax.js	4	3	3	3	3	3	3	2	3	3	**3**
AutoSuggest.js	5	8	3	3	3	5	3	3	2	3	**4**

[3] Dell Optiplex 210L (Intel Pentium 4 2.8 GHz CPU and 2 GB DDR2 RAM (533 MHz)) running Microsoft Windows XP Professional SP2.

Table 4-16: Execution time for Twingly, measured with Firebug in (ms)

Test #	1	2	3	4	5	6	7	8	9	10	Average
Index.php	24	24	25	34	24	35	24	25	24	26	**27**
Base.css	14	19	13	16	12	12	12	12	11	13	**13**
Keyboard.css	26	30	24	23	25	23	25	26	24	22	**25**
Tagcloud.css	30	33	29	29	28	35	32	28	30	26	**30**
AutoSuggest.css	33	29	23	26	30	35	30	31	27	29	**29**
Action.js	38	34	33	30	31	26	27	35	32	35	**32**
Movement.js	7	3	3	4	9	4	3	4	3	3	**4**
Include_Keyhandler.js	4	3	4	5	3	7	3	3	3	3	**4**
Keyhandler_workstation.js	3	4	3	3	4	5	5	4	3	4	**4**
Keyboard.js	3	3	3	3	3	3	3	3	3	3	**3**
Navigation.js	3	3	3	3	4	3	3	3	3	3	**3**
Ajax.js	3	2	3	5	3	3	3	3	3	3	**3**
AutoSuggest.js	4	3	3	3	4	3	3	3	2	4	**3**

Table 4-17: Execution time for Wikipedia, measured with Firebug in (ms)

Test #	1	2	3	4	5	6	7	8	9	10	Average
Index.php	282	286	264	262	257	260	257	285	246	259	**266**
Base.css	19	22	7	7	7	7	17	7	7	7	**11**
Keyboard.css	23	25	14	17	16	17	18	18	16	15	**18**
AutoSuggest.css	27	26	14	17	12	13	19	14	16	16	**17**
Action.js	28	28	15	17	15	16	19	17	16	16	**19**
Movement.js	3	3	3	3	3	3	3	3	3	2	**3**
Include_Keyhandler	3	3	3	3	2	3	3	2	3	4	**3**
Keyhandler_ workstation.js	3	4	3	3	4	3	3	4	3	4	**3**
Keyboard.js	3	3	3	3	3	4	2	3	3	4	**3**
Navigation.js	6	3	6	5	3	3	4	5	6	7	**5**
AutoSuggest.js	3	3	3	3	3	3	3	3	2	3	**3**
Ajax.js	3	3	3	2	3	3	4	3	3	5	**3**
Yahoo-DOM-event.js	3	3	3	3	3	3	5	3	3	4	**3**
Animation-min.js	5	7	4	4	5	4	3	3	4	3	**4**

Table 4-18: Execution time for IMDb, measured with Firebug in (ms)

Test #	1	2	3	4	5	6	7	8	9	10	Average
Index.php	58	57	58	57	65	60	60	58	53	63	**59**
Base.css	29	26	27	27	33	29	29	34	26	29	**29**
Keyboard.css	49	41	35	38	46	39	39	50	50	45	**43**
tagCloud.css	54	43	42	41	48	41	41	51	52	46	**46**
AutoSuggest.css	45	46	45	47	55	48	48	54	57	50	**50**
Action.js	56	48	47	54	57	49	49	58	60	53	**53**
Movement.js	8	7	6	6	6	7	7	7	6	6	**7**
Include_Keyhandler	6	5	5	8	6	5	5	5	5	6	**6**
Keyhandler_ workstation.js	9	5	6	6	7	8	8	6	7	7	**7**
Keyboard.js	6	7	9	6	7	7	7	7	9	6	**7**
Navigation.js	8	8	5	6	6	9	9	9	5	8	**7**
Ajax.js	5	6	6	6	5	6	6	6	5	6	**6**
AutoSuggest.js	5	6	5	5	6	6	6	6	6	5	**6**

4.4. Different search methods

Since every search string is unique we can not say that we save a specific number of key presses by using our tag-cloud or auto-suggest functionalities for navigation. However, we can say when the search string is already in the database (e.g. enabling the auto-suggest functionality) it will only take one button press more than needed to navigate to the first letter in the string you want to search for. To show this we present the number of buttons pressed to search for the ten most popular strings in our database (see Table 4-19). We assume that the user starts with their focus on the letter "a" on the virtual keyboard, which is the state when the page has just been loaded. As seen in the table, combining tag-cloud and auto-suggest means that we never have to press more than nine buttons to reach any of the ten most popular search strings, this is a lot better than if we only use the virtual keyboard. As of fall 2008 there are no mice or similar devices for STBs. According to Motorola/Kreatel technical specifications there is mouse support for the STB [83], but it is not just plug-and-play (e.g. you cannot just put a mouse into the USB port and expect a cursor as on a PC). While using a mouse would possibly make the navigation more user-friendly, the user would need to have yet another device (the mouse) besides the remote control; thus while more user friendly it would not necessarily be *convenient*.

Table 4-19: Number of key presses to search for popular strings in Flickr

Search string	Keyboard	Auto-suggest	Tag-cloud
Zoo	18	11	2
Yummy	27	10	3
Urbanfeel	41	6	8
Yes	25	10	6
Spcoon	27	4	9
Yi	16	10	5
Yon	17	10	4
Yee	17	10	7
Bltphotos	45	3	10
Hej	21	9	11
Average	**25,4**	**8,3**	**6,5**

4.5. Library functions

The main purpose of this master's thesis is to ease the implementation of Web 2.0 applications for the IPTV format, so every line of code we can reuse by abstracting it to a library function is an improvement. Figure 4-1 illustrates our catalogue structure. We have a number of global JavaScript, CSS, and PHP files which we can include in future applications to provide the same functionality in different services without writing similar code. We still need to create some files for the service specific code and layout needed for each specific application. These files will be named: *action.js, movement.js,* and *base.css* and would be placed inside each specific local service folder. In table 4-20 you see the code reuse we have on Flickr and Twingly. For IMDb and Wikipedia we could not reuse as much of the code because we could not use the same method to retrieve the info. Our Wikipedia and IMDb implementations are created in a different way than Flickr and Twingly and do have much of the logic on other places (more server-side and less client-side) compared to them. Since IMDb and Wikipedia do not share all the 1200 lines of common code with the other two services they are presented by themselves in table 4-21.

We could use an XSLT parser to modify data that is unique for every service, such as the requests and responses to and from the services. As an example there are open source XSLT parsers which convert XML requests to JSON requests and the opposite. Such parsers would reduce the work when implementing a new service since the developer can write the request in the format he or she is most familiar with and use the XSLT file to convert the request to the proper format. [84]

Table 4-20: Code reuse - API-based applications

Service	# lines of code	Percent of reused code
Common code	1200	-
Flickr	433	73 %
Twingly	322	79 %

Table 4-21: Code reuse – Other applications

Service	# lines of code	Percent of reused code
Common code	905	-
IMDb	542	63 %
Wikipedia	461	66 %

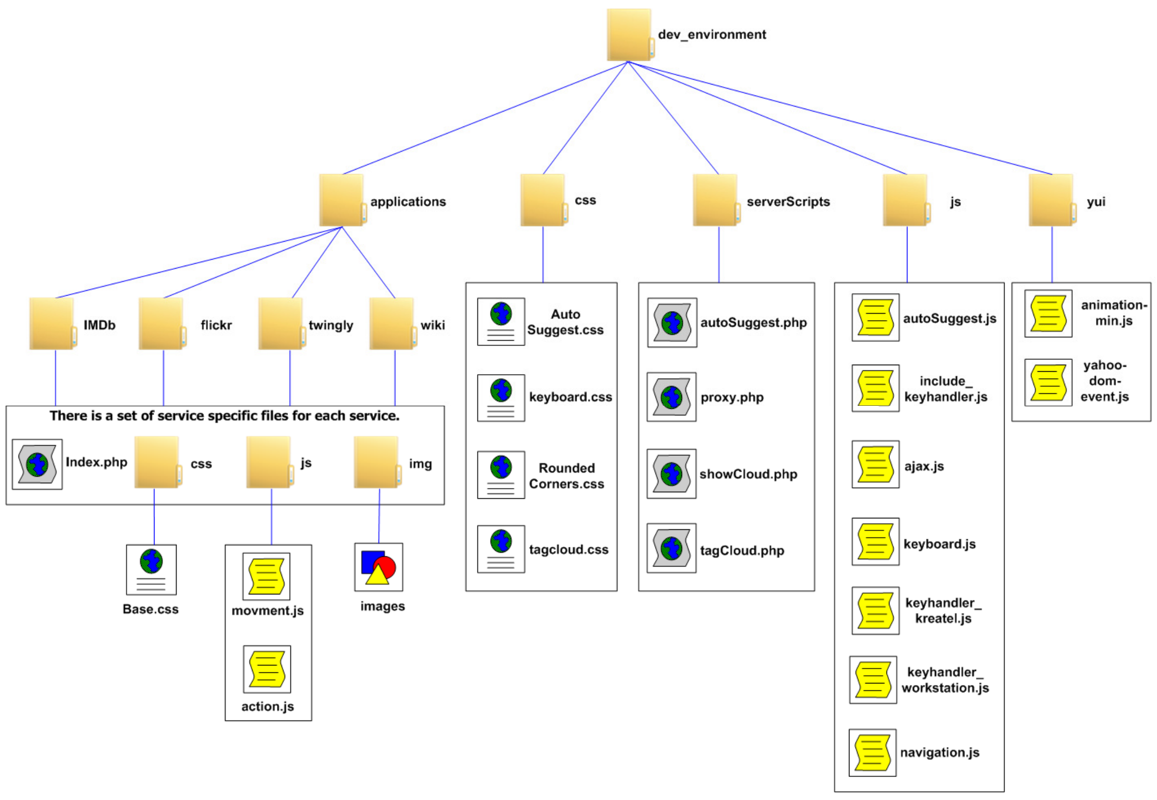


Figure 4-1: Organization of our code

autosuggest.js	Contains a function that calls the autoSuggest.php every time a key is pressed to retrieve matching words so far with ajax.
include_keyhandler.js	Contains code that checks what platform the client is using by checking the DOM user agent info.
ajax.js	Contains the ajax-function that return an ajax object.

keyboard.js	Defines the virtual keyboard.
keyhandler_kreatel.js	Defines every key code on the remote control to human readable variables.
keyhandler_workstation.js	Defines every key code on the PC keyboard to human readable variables.
navigation.js	Contains the generic navigation example are virtual keyboard, tag-cloud, search button, search field, etc.
autoSuggest.php	Connects to the MySQL database and retrieves the words that matches the input parameter.
proxy.php	Forwards requests for images and urls.
showCloud.php	Connects to MySQL database and retrieves the 10 most frequently searched words and displays them alphabetically with different font size depending on how high the counter is
tagCloud.php	Updates the counter or inserts a new word to the MySQL database if it does not exist.
autoSuggest.css	Defines the layout for the auto-suggest menu.
keyboard.css	Defines the layout for the keyboard.
tagCloud.css	Defines the layout for the tag-cloud.

4.6. Implementation

We have divided the code into different folders and files so it will be easy to find and easy to include the tag-cloud, keyboard, or auto-suggest features into a new web services. Functions that run on the STB are written mostly in HTML or JavaScript. Functions that run on server side are written mostly in JavaScript or PHP. Our virtual keyboard is in alphabetic ordered because it is easiest to navigate and people recognize it.

If we would like to change the alphabet (or add more special characters) we simply change the values in an array in the keyboard.js file. To get other types of Unicode characters than the ones we use there are converters available on the Internet. [85]

Some remote controls are prepared for triple touch typing by having letters printed next to the digits (see figure 4-2). If we combined this with the T9 functionality (similar to our auto-suggest function) it will be easier for the user to type compared to a virtual keyboard. It is also possible to achieve this for those remote controls that do not have the letters printed on them by having a picture on screen mapping the digits to the letters instead of virtual keyboard.

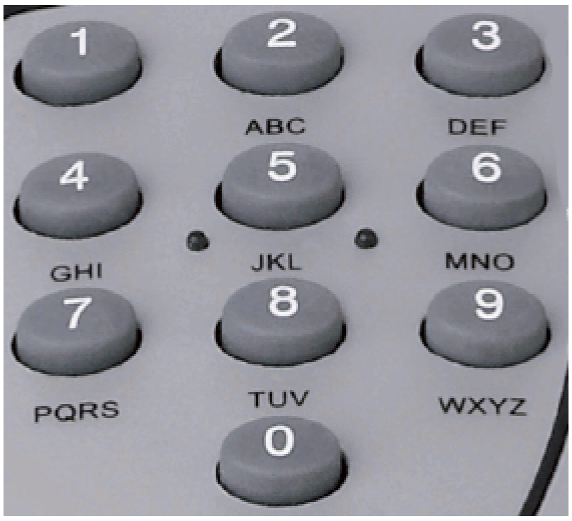


Figure 4-2: One of Motorola's remote controls with printed letters

According to the article Performance Optimization of Virtual Keyboards [86] there are two factors that may minimize key movements. One is the transitional frequencies from one letter to another and the other is the relative distances between the keys. An ideal keyboard should have the most frequent keys in the centre and the frequently connected letters should be closer to each other than less frequently connected letters. By using Fitt's law the average movement time (MT) can be predicted by the formula below (were Dij is the distance between the middle part of two keys and Wj is the width of the key).

$$MT = a + b \times \log_2\left(\frac{D_{ij}}{W_j} + 1\right) \qquad a = 0 \qquad b = \frac{1}{4.9}$$

$$t = \sum_{i=1}^{z}\sum_{i=1}^{z}\frac{P_{ij}}{IP}\left[\log_2\left(\frac{D_{ij}}{W_j} + 1\right)\right]$$

z = total number of keys
P_{ij} = frequency of letter j to follow by letter i
i = j = 1 is when the user taps on the same key successively (e.g., oo as in look)
IP = Index of performance (a = 0, b = 4.9 are values used in the article and can be determined experimentally by fitting a straight line to measured data.)

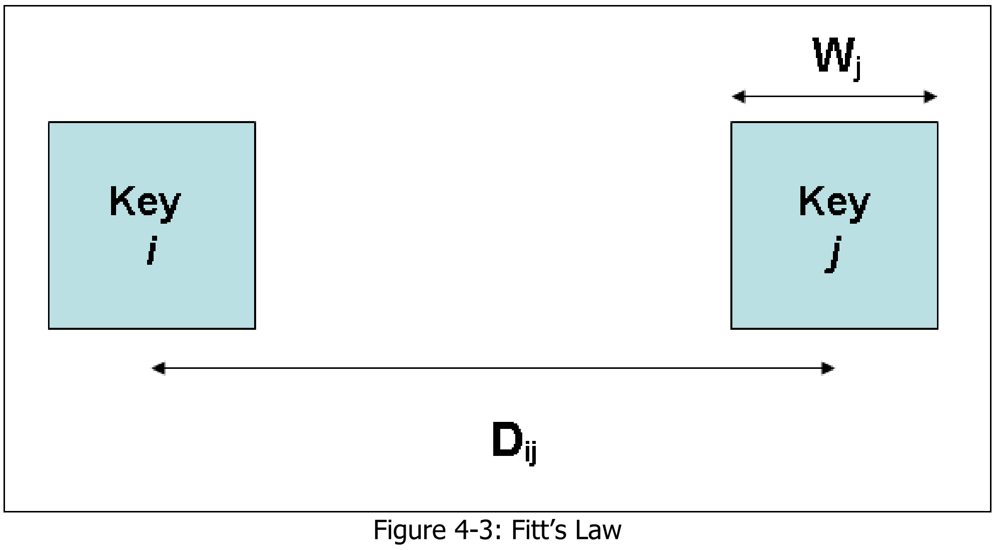


Figure 4-3: Fitt's Law

This research is targeting touch-screen devices and their conclusions do not necessarily match our applications. They also exclude special characters such as a dot or a slash which are important for our purposes. We have chosen to use a square alphabetic keyboard since most people recognize it, the writing speed is longer compared to the qwerty-keyboard, and the layout is more suitable for IPTV usage. Table 4-21 presents movement efficiency measured in words per minute (wpm) based on Fitts's law "Index of performance (IP) levels" for two different keyboard layouts. Fitt's law is very mathematical and does not consider if the layouts are user friendly. If a user has not seen the layout before it will take time to learn and find the keys. However, it is widely used and provides an upper bound on what the performance can be.

Table 4-22: Movement efficiency in words per minute (IP = Index of Performance)

Keyboard	IP = 4.9 bits/sec	IP = 6.0 bits/sec	IP = 8.0 bits/sec
Qwerty	28,0	34,3	45,7
5x6 Alpahbetic	33,5	41,0	54,7

4.7. Design

As stated earlier we kept our design very simple to make it easier for the user to navigate within the service. The layout and colors are similar to the original web service so it will be recognizable. We tried different layouts in the beginning and decided upon a layout that we thinks is most suitable on a TV. We had to use big fonts because the distance to a TV is further than to the display a regular workstation/PC. The screen space is limited so we had to show and hide different objects depending on what the user is doing. When the user is searching, then the virtual keyboard is shown, but when a search is successful we needed to hide the virtual keyboard to show the result of the search. We have not carried out any large scale user study, but did speak with application designers inside Accedo Broadband to get their feedback regarding our design choices.

5. Conclusions and Future work

In section 5.1 includes the knowledge that we have gained during this period. We tell about the limits of applications running through an STB and how to best utilize the STB to offer the user the most experience. We do also discuss interesting features and applications that are exciting to work with as an extension of this thesis in section 5.2.

5.1. Conclusions

Some of the practical limitations of IPTV devices are due to the user interface generally being limited to navigating using a remote control, limited screen resolution, and a limited level of DOM support in the STB web browsers. Thus we expect that suitable applications will be services with a simple GUI which do not require a lot of user input or navigation. From a user's perspective it seems more likely that a user will use a service for reading blog content, rather than posting through the IPTV interface. With this in mind we will probably not add extra features that require the user to log-in to websites for all the services. But extra value is often provided when logging in to some popular services, so it will still be interesting to try it out. Services that seem to fit well with the IPTV experience are for example IMDb (for getting information about movies), Wikipedia (read encyclopedia on TV), Flickr (for using an IPTV STB to view photographs), and Twingly (a service to read blogs). There are wireless keyboards available for many STBs, but we have found that the majority of users do not have such a keyboard. Therefore our applications were developed to function well with only the remote control. The main requirement is to provide a means for the user to be able to search for contents in a clever way. The GUI was kept very simple so that the design could be easily implemented using only a few of the development techniques described in this report. We found that there are a lot of similarities among the Web 2.0 services that we found interesting and relevant for the IPTV format. All of these serves rely on a search function for most of the initial navigation and the OK button for selection. The sites should have a limited amount of this content using XHTML and CSS. The largest difference amongst the four proposed services is that IMDb do not offer an open API for their services, while Flickr and many blog content providers have JSON or XML-RPC feeds that are free for everyone to use.

Although STBs do provide web browsers, most STB vendors prohibit their users from being able to surf on the web. There are vendors (for example Sharp) who accept a few pre-defined sites, which have been tested to support IPTV, but there are very few IPTV operators who allow the user to surf to all web sites. The few IPTV operators who actually allow surfing (see figure 5-3 for Telia's web interface) seem to provide a really poor user experience, both due to technical reasons (specifically poor screen resolution on the TV which leads to a lot of scrolling) and due to a poor web browser which makes it an effort just to type in an address. Some of the problems can be solved by using a service like Del.icio.us (see section 2.2.3) to provide clickable links instead of typing each website's URL into the browser, but the problem with lots of scrolling will occur until the either the browser becomes more clever or more websites start to offer sites well suited for IPTVs. There is already a market for websites specifically designed for mobile devices, so it is possible that we will experience the same for IPTV devices in the future. See figure 5-4 for an example of a WAP site in Telia's IPTV web browser.

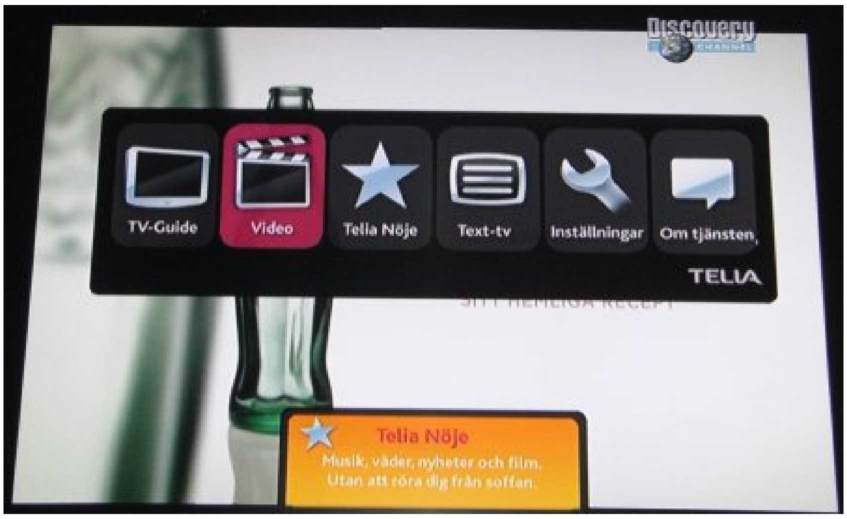


Figure 5-1: Telia start

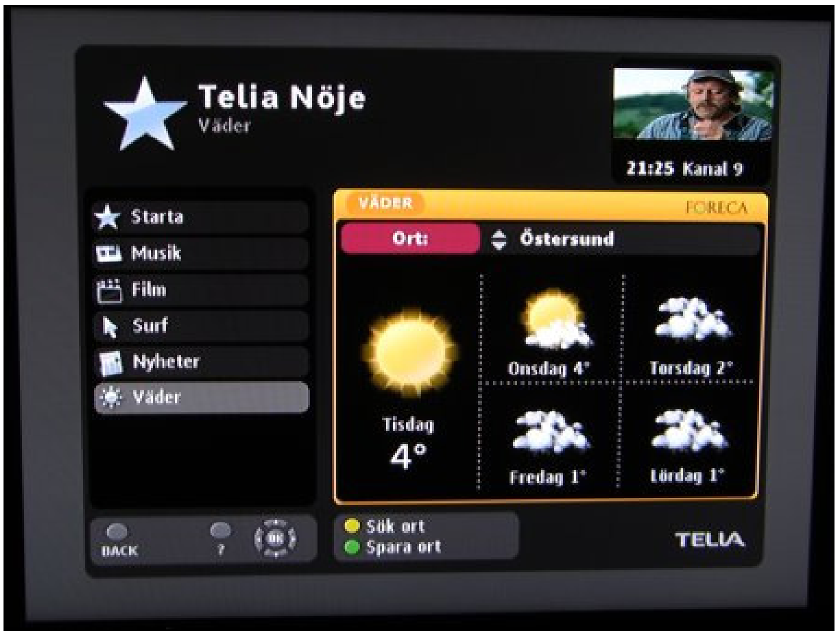
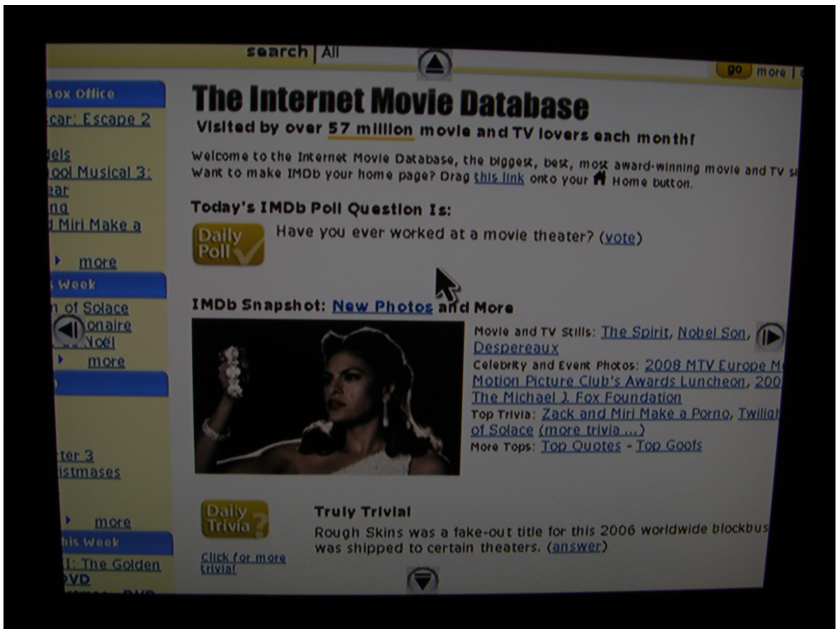


Figure 5-2: Telia "Nöje" (Entertainment) menu



The Internet Movie Database
Visited by over 57 million movie and TV lovers each month!

Welcome to the Internet Movie Database, the biggest, best, most award-winning movie and TV si
Want to make IMDb your home page? Drag this link onto your Home button.

Today's IMDb Poll Question Is:

Have you ever worked at a movie theater? (vote)

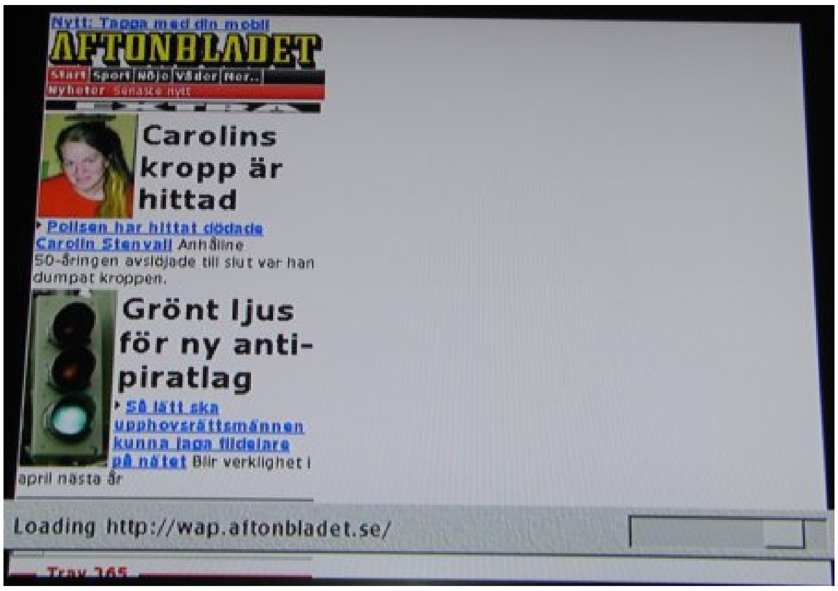
IMDb Snapshot: New Photos and More

Movie and TV Stills: The Spirit, Nobel Son, Despereaux
Celebrity and Event Photos: 2008 MTV Europe M Motion Picture Club's Awards Luncheon, 200 The Michael J. Fox Foundation
Top Trivia: Zack and Miri Make a Porno, Twiligh of Solace (more trivia ...)
More Tops: Top Quotes - Top Goofs

Truly Trivial
Rough Skins was a fake-out title for this 2006 worldwide blockbus was shipped to certain theaters. (answer)

Click for more trivia!

Figure 5-3: Telia Surf (http://www.imdb.com)


Nytt: Tagga med din mobil

AFTONBLADET

Start | Sport | Nöjo | Väder | Mer..

Nyheter Senaste nytt

EXTRA

Carolins kropp är hittad

Polisen har hittat dödade Carolin Stenvall Anhållne 50-åringen avslöjade till slut var han dumpat kroppen.

Grönt ljus för ny anti-piratlag

Så lätt ska upphovsrättsmännen kunna jaga fildelare på nätet Blir verklighet i april nästa år

Loading http://wap.aftonbladet.se/

Figure 5-4: Telia Surf (http://wap.aftonbladet.se) WAP site

We have found XSLT to be very interesting for our project because it can be used for rearranging/transforming data from XML files. This will probably help us when we want to abstract information from XML or JSON feeds from different web services and to create output that suits our purposes. We think that PHP is a suitable server-side language for the applications that will run on the service provider's web server. JavaScript/AJAX will fill the need for client-side operations. If we want to save sessions for a user with a 1720 STB, then we will need a database to save the information because cookies will not be saved when the power is disconnected from the STB. It should be noted that there are STBs that have a hard disk or a flash memory and thus can store cookies locally on the STB. A problem is that many IPTV providers offer their customers a very simple STB when subscribing to their service, thus to reach a wider audience we should develop our applications for the popular Motorola 1720 STB. However, high-end STBs that provide a better TV experience with HDTV, large storage capacity, digital audio, and better picture will become widely available in the near future. Thus we should be sure to enable our solutions to evolve with the market adoption of such STBs.

Unfortunately, we can not use all the library functions in JavaScript on the STB. For example, the onFocus-function is very useful for navigation on regular websites, but this function does not work at all on the Motorola 1720 STB, see table 5-1. As a replacement we wrote our own implementation based on `<div>` tags in HTML, positioning in CSS and our own JavaScript functions. The visibility part of CSS is very useful for presenting features such as a keyboard, buttons, information, etc, based on the user's input and navigation. Since it is much more difficult to type on a remote control than a keyboard, an auto-suggest function based on AJAX will probably increase the usability of any application quite a lot.

Table 5-1: Common events and buttons that does not work with IPTV box

Mouse events	onClick, onDblClick, onMouseDown, onMouseOver, onMouseUp, and onMouseOut
Keyboard Events	onKeyDown, onKeyPress, onKeyUp
Button Object Methods	blur – Removes focus from a button click – Simlulates a mouse click on a button focus – Gives focus to a button
Submit buttons, radio buttons, checkboxes, and dropdown box	We can not use these html elements because we simply do not have a mouse device.

When studying web technologies we also noticed a trend - scripting languages have gained a lot of popularity recently. Programming languages as Java and the .NET framework seem to have lost developers to PHP, Ruby on Rails, JavaScript, etc. when it comes to development of applications running in a web browser [87]. Large software providers are contributing to the development of some scripting languages to increase their performance. Thus scripting languages should seen as another style of programming language , rather than as some kind of amateur alternative to regular programming languages (as many see them today). This will probably lead to greater usage by professionals, as scripting languages further increase the ability to develop advanced functionalities by web developers.

In order to test the portability of our applications we ran them on the Amino AmiNET130 STB, which is an HD quality IPTV STB running an Opera browser. The only part we had to change was the key-handler since this STB uses another coding for the remote control IR key codes. Since the Amino STB offers a resolution of 1080p our applications are presented in a higher quality. We consider our code is quite portable!

During the development period it was desirable to use only scripting in this thesis project.

5.2. Future Work

These kinds of services can always be improved by both evolving technology and increasing functionality since the original versions constantly are developing. Most services are easy to extend thanks to large libraries which offer lots of interesting functions. New features and updates can easily be adopted by all our services, since most of our own code and design are library functions.

It would be fairly easy and very useful to implement a way for the user to choose what to search for when typing a string in the search field (or other field). This can be done with a multiple choice dropdown menu with potential entries as; username, image name, and image tag. It is also possible to retrieve comments and information about the images we present, but we found it a bit disturbing while viewing a slideshow and decided to not use this in our application. A screensaver showing pictures with a simple animation is feasible and if we download five or ten images at start-up, then continually download additional images as the slideshow runs, this feature would probably run in the available memory. Some countries, or families, are probably very keen to use the parental control filter to filter out adult images and content that they believe is unsuitable. As for all applications of this kind, it is of course interesting to present a top list of the best photographs or most popular users, there is space left for this kind of information on the bottom of our start page.

Twingly does not provide an open API; if they will do so in the future, it would be a nice feature to increase the numbers of blogs you receive by the JSON response, 10 blogs are probably too few for a blog search service. We would also like to be able to retrieve the whole blogs without needing to follow the link provided by Twingly's service, and then parse them for interesting content. Some users might also be interested in leaving comments after reading a blog or rate it with, thus there is a need for them to enter text.

An interesting feature for many applications is for the user to be able to grade the content in a simple way. In the case of Flickr and Twingly, this could enable two types of lists – most viewed and highest quality/most popular. Such lists would probably make it easier to navigate through out the service and users would be able to reach popular content faster.

There are APIs which provide good META data, such as Flickr, and an increasing use of this data can lead to more clever ways to search for content. Stacey Anklam at Red Bee Media [88] have stated that their company had an increase of 1/3 in the usage of On-Demand services when they started to use META data in a proper way and redesigned their GUI to be more usable.

An innovative way of increasing the user ability could be to use a touch screen Smartphone as advanced remote control for example the Apple iPhone. This could lead to increasing amount of advanced features such as adding content to your blog through your IPTV interface. This could be implemented in two ways either by using the IR/Bluetooth interface (the iPhone does only provide the Bluetooth) or on a web application that go to the web server that host that current application.

With the increasing performance of future IPTV STBs and the more advanced web based media services are gaining popularity, such as Joost and Spotify, it would be interesting to see how they work for the IPTV environment.

References

[1] Gartner - Worldwide Consumer Broadband Penetration, visited 2008-09-09
http://www.gartner.com/it/page.jsp?id=501276

[2] Wikipedia, IPTV, visited 2008-09-09
http://en.wikipedia.org/wiki/Iptv

[3] Television is changing, Published 2006-03-12
http://arstechnica.com/guides/other/iptv.ars

[4] Steve Farmer, Panel discussion "IPTV: Is TV the latest to join the world of 2.0?", TelecomTV, 2008-08-16

[5] Tim O'Reilly, What Is Web 2.0, Published 2005-09-30
http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html

[6] Bill Kennedy & Chuck Musciano, *HTML & XHTML.The Definitive Guide 6th Edition*, O'Reilly, Published 2006, ISBN: 0-596-52732-2

[7] Wikipedia, CSS, visited 2008-09-09
http://en.wikipedia.org/wiki/Cascading_Style_Sheets

[8] W3 DOM, visited 2008-09-09
http://www.w3.org/DOM/

[9] W3C Document Object Model, W3C Document Object Model
http://xml.coverpages.org/dom.html

[10] Shelley Powers, *Learning JavaScript*, O'Reilly, Published 2006
ISBN: 0-596-52746-2

[11] Extensible Markup Language (XML), visited 2008-09-09
http://www.w3.org/XML/

[12] XML - Managing Data Exchange, visited 2008-09-09
http://en.wikibooks.org/wiki/XML

[13] Introducing JSON, visited 2008-09-09
http://www.json.org/

[14] The application/json Media Type for JSON, visited 2008-09-09
http://tools.ietf.org/html/rfc4627

[15] JSON-RPC Specifications, visited 2008-09-09
http://json-rpc.org/wiki/specification

[16] XML-RPC Home Page, visited 2008-09-09
http://www.xmlrpc.com/

[17] Thrift: Scalable Cross-Language Services Implementation,
visited 2008-09-09
http://developers.facebook.com/thrift/thrift-20070401.pdf

[18] Anthony T. Holdener III, *Ajax: The Definitive Guide*,
O'Reilly, 2008, ISBN: 0-596-52838-8

[19] Wikipedia, AJAX, visited 2008-09-10
http://en.wikipedia.org/wiki/Ajax_(programming)

[20] Rasmus Lerdorf, Peter MacIntyre, Kevin Tatroe, *Programming PHP, 2nd Edition*, O'Reilly, Published 2006, ISBN: 0-596-00681-0

[21] Wikipedia, PHP, visited 2008-09-10
http://en.wikipedia.org/wiki/Php

[22] Wikipedia, SQL, visited 2008-09-10
http://en.wikipedia.org/wiki/Sql

[23] Wikipedia, XSLT, visited 2008-09-10
http://en.wikipedia.org/wiki/Xslt

[24] Wikipedia, SVG, visited 2008-11-04
http://en.wikipedia.org/wiki/Svg

[25] Wikipedia, Flash, visited 2008-11-04
http://en.wikipedia.org/wiki/Adobe_Flash

[26] Erik Hedin, Social networks and mobile devices, KTH, 2006-03-14
http://web.it.kth.se/~maguire/DEGREE-PROJECT-REPORTS/080915-Erik_Hedin-with-cover.pdf

[27] Wikipedia, Flickr, visited 2008-09-10
http://en.wikipedia.org/wiki/Flickr

[28] Blogger, visited 2008-09-10
http://www.blogger.com/tour_pub.g

[29] Blogger Developer Documentation, visited 2008-09-10
http://code.blogger.com/

[30] Wikipedia, Delicous, visited 2008-09-11
http://en.wikipedia.org/wiki/Del.icio.us

[31] Wikipedia, Facebook feature, visited 2008-09-11
http://en.wikipedia.org/wiki/Facebook_features

[32] Facebook Markup Language, visited 2008-09-11
http://oren.blogs.com/praxis/2007/05/facebook_markup.html

[33] Facebook developers, visited 2008-09-11
http://wiki.developers.facebook.com/index.php/Main_Page

[34] Annika Hamrud, Dagens Nyheter, published 2008-05-19
http://www.dn.se/DNet/jsp/polopoly.jsp?d=147&a=771168

[35] Wikipedia, Wikipedia, visited 2008-09-11
http://en.wikipedia.org/wiki/Wikipedia

[36] Martin Hassel, Evaluation of Automatic Text Summarization, Licentiate thesis, School of Computer Science and Communication, Royal Institute of Technology (KTH), published 2004
http://www.csc.kth.se/~xmartin/papers/licthesis_xmartin_notrims.pdf

[37] Martin Hassel, Resource Lean and Portable Automatic Text Summarization, Doctoral thesis, School of Computer Science and Communication, Royal Institute of Technology (KTH), published 2007
http://www.csc.kth.se/~xmartin/thesis/spikblad.pdf

[38] Lee Gomes, *Will All of Us Get Our 15 Minutes On a YouTube Video?*, Wall Street Journal, 2006-08-30, visited 2008-09-11
http://online.wsj.com/public/article/SB115689298168048904-5wWyrSwyn6RfVfz9NwLk774VUWc_20070829.html

[39] YouTube law fight 'threatens net', visited 2008-09-11
http://news.bbc.co.uk/1/hi/technology/7420955.stm

[40] YouTube Barack Obama, visited 2008-12-18
http://www.youtube.com/barackobama

[41] Wikipedia, YouTube, visited 2008-09-11
http://en.wikipedia.org/wiki/Youtube

[42] Wikipedia, Imdb, visited 2008-09-11
http://en.wikipedia.org/wiki/Imdb

[43] Wikipedia, MySpace, visited 2008-10-06
http://en.wikipedia.org/wiki/Myspace#cite_note-2

[44] Wikipedia, Windows Sidebar, visited 2008-11-24
http://en.wikipedia.org/wiki/Windows_Sidebar

[45] MySpace API, visited 2008-10-06
http://developer.myspace.com/community/myspace/anatomyOfAnApp.aspx

[46] Twingly, visited 2008-10-06
http://www.newsdesk.se/pressroom/twingly

[47] eBizMBA Monthly visitors, visited 2008-09-10
http://www.ebizmba.com/articles/user-generated-content

[48] Alexa top sites, visited 2008-09-11
http://www.alexa.com/site/ds/top_sites?ts_mode=global&lang=none

[49] Wikipedia, Tag-cloud, visited 2008-10-28
http://en.wikipedia.org/wiki/Tag_cloud

[50] Wikipedia, Auto Complete, visited 2008-10-28
http://en.wikipedia.org/wiki/AutoComplete

[51] Wikipedia, Virtual keyboards, visited 2008-11-26
http://en.wikipedia.org/wiki/Virtual_keyboard

[52] Nuance T9 Text Input : The global standard for mobile text input, visited 2008-11-24
ftp://ftp.scansoft.com/nuance/datasheets/ds_t9.pdf

[53] Wikipedia, Modal window, visited 2008-10-28
http://en.wikipedia.org/wiki/Modal_window

[54] Wikipedia, Speech Recognition, visited 2008-12-02
http://en.wikipedia.org/wiki/Speech_recognition

[55] Wikipedia, Bluetooth, visited 2008-12-02
http://en.wikipedia.org/wiki/Bluetooth

[56] Kreatel 1720 Technical specification, visited 2008-09-11
http://www.ohd.no/downloads/pdf/Kreatel_IP-STB_1720.pdf

[57] Johan Ekström, Wickrpedia Integrering av sociala tjänster, Undergraduate thesis C-level MSI/Medieteknik Växjö University, 2006-03-14
http://www.diva-portal.org/diva/getDocument?urn_nbn_se_vxu_diva-483-2__fulltext.pdf

[58] Social Bookmark Script - Script code generator, visited 2008-09-16
http://www.social-bookmark-script.com/generator.htm

[59] Trent Pomeroy and Curtis Howe, "Frostt: Accelerating IPTV Innovation", White Paper, Mariner Partners, Inc., Saint John, NB, Canada, 8 April 2008
http://www.frostt.tv/accelerating_iptv_innovation_whitepaper.pdf

[60] Lars Mikael Liljeroth , Hardware accelerated user interface architecture for IPTV set-top boxes, KTH, 2008-08-24
http://web.it.kth.se/~johanmon/theses/liljeroth.pdf

[61] John Allen, CEO DigiSoft.tv, JavaOne^{SM} Conference, 2007
http://pl.sun.com/sunnews/events/2007/about_content/pdf/jallen.pdf

[62] Jonas Rydberg, *Ditt privatliv ska sälja Ericssons IP-tv*, Metro Teknik, visited 2008-11-05
http://www.metro.se/se/article/2008/11/04/16/5547-45/index.xml

[63] Ola Haraldson Halset, IP Television - Content on Demand:
A usability and user experience evaluation of an IP television portal by the Norwegian broadcaster TV 2 Interaktiv, Master thesis, Department of Numeric Analysis and Computer Science (NADA), Royal Institute of Technology (KTH), 2006 February
http://www.nada.kth.se/utbildning/grukth/exjobb/rapportlistor/2006/rapporter06/halset_ola_06025.pdf

[64] Robert Högberg, "Video telephony in an IP-based set-top box environment", Master thesis, Linköping Institute of Technology, LITH-IDA/DS-EX, 2004-04-08
http://liu.diva-portal.org/smash/get/diva2:19682/FULLTEXT01

[65] John Brännström, "Bokningssystem för gemensamhetsresurser via TV", Master Thesis, Institutionen för Systemteknik Luleå Tekniska Universitet, 2006 January
http://epubl.luth.se/1404-5494/2006/011/LTU-HIP-EX-06011-SE.pdf

[66] Behzad Hamzei Tavosloi, "Design and Implementation of an Application Server for an IPTV Environment Architecture, Prototype System Design, Implementation and Evaluation", Master of Science Thesis, School of Information and Communication Technology, Royal Institute of Technology (KTH), 2008 January
http://web.it.kth.se/~johanmon/theses/hamzei-tavosloi.pdf

[67] Camilla Isaksson, "Gränssnitt och tjänster för IPTV", Luleå Tekniska Universitet, Institutionen för Systemteknik, 2005
http://epubl.luth.se/1402-1617/2006/027/LTU-EX-06027-SE.pdf

[68] Ani Nenkova, Vocabulary size and term distirubtion: tokenization, text normalization and stemming, visited 2008-11-20
www.cis.upenn.edu/~nenkova/Courses/cis430/Lecture02.ppt

[69] Wikipedia, Wikipedia:Copyrights, visited 2008-12-05
http://en.wikipedia.org/wiki/Wikipedia:Copyrights

[70] Tim O'Reilly, O'Reilly Radar, 2006-05-05
http://radar.oreilly.com/archives/2006/05/imdb-api-1.html

[71] I want to work at IMDb!, visited 2008-11-25
http://www.imdb.com/help/show_leaf?jobatimdb#software-uk

[72] IMDBPHP, visited 2008-11-25
http://projects.izzysoft.de/trac/imdbphp

[73] IMDb photos copyright, visited 2008-12-04
http://www.imdb.com/help/show_leaf?usephotos

[74] Trynt:tech IMDb, visited 2008-11-25
http://www.trynt.com/trynt-movie-imdb-api/

[75] IMDb license, visited 2008-12-04
http://www.imdb.com/help/show_article?conditions

[76] IMDb interfaces, visited 2008-12-04
http://www.imdb.com/interfaces

[77] EBay API, visited 2008-11-29
http://developer.ebay.com/common/api/

[78] YouTube API, visited 2008-11-29
http://code.google.com/apis/youtube/overview.html

[79] Facebook API, visited 2008-11-29
http://developers.facebook.com/

[80] Statement of compliance Kreatel TV Software 2.4
http://www.elekta.lt/~mindaugas/sat/dvbt/SoC.pdf

[81] Gregg Keizer, Fix Firefox's memory problems, visited 2008-11-20
http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9046561

[82] Javascript The definitive Guide, visited 2008-12-03
http://www.unixmexico.org/files/html/kore.hack.se/oreilly/web/jscript/ch11_07.html

[83] Kreatel TV software technical specification
http://www.elekta.lt/~mindaugas/sat/dvbt/KreatelTVsoftware.pdf

[84] XSLTJSON, XML to JSON using XSLT, visited 2008-11-27
http://www.bramstein.com/projects/xsltjson/

[85] Unicode Code Converter v6, visited 2008-12-15
http://people.w3.org/rishida/scripts/uniview/conversion.php

[86] Shumin Zhai, et. al., Performance Optimization of Virtual Keyboards, visited 2008-11-17
http://www.almaden.ibm.com/cs/people/zhai/papers/ZhaiHunterSmithHCIGalley.pdf

[87] Martin Wallström, Skriptspråk allt hetare, ComputerSweden, 2008-09-03, visited 2008-11-03
http://www.idg.se/2.1085/1.177287

[88] Stacey Anklam, Panel discussion “IPTV: Is TV the latest to join the world of 2.0?”, TelecomTV, 2008-08-16
http://web20.telecomtv.com/pages/?id=c2a3b30f-edea-4aee-925a-72fdcd51b846&vidid=2399&view=video&page=1

Further information

[A] http://www.w3schools.com/html/default.asp, visited 2008-09-09

[B] http://www.w3schools.com/css/default.asp, visited 2008-09-09

[C] http://www.w3schools.com/js/default.asp, visited 2008-09-09

[D] http://www.w3schools.com/htmldom/default.asp, visited 2008-09-09

[E] http://www.w3schools.com/xml/default.asp, visited 2008-09-09

[F] http://www.json.org/, visited 2008-09-09

[G] http://json-rpc.org/, visited 2008-09-09

[H] http://www.xmlrpc.com/, visited 2008-09-09

[I] http://incubator.apache.org/thrift/, visited 2008-09-10

[J] http://www.w3schools.com/ajax/default.asp, visited 2008-09-10

[K] http://www.w3schools.com/php/default.asp, visited 2008-09-10

[L] http://www.w3schools.com/sql/default.asp, visited 2008-09-10

[M] http://www.w3schools.com/xsl/default.asp, visited 2008-09-10

[N] http://www.w3.org/Graphics/SVG/, visited 2008-11-14

[O] http://www.w3schools.com/flash/flash_intro.asp, visited 2008-11-14

[P] http:// www.adobe.com/support/flash/action_scripts/actionscript_tutorial/, visited 2008-11-14

[Q] www.wireshark.com, visited 2008-11-25,

[R] http://getfirebug.com/, visited 2008-09-16,

Appendices

Appendix A	Motorola/Kreatel Troubleshooting, Logging Only available for Motorola/Kreatel developers
Appendix B	Motorola/Kreatel Terminal Open Interface/JavaScript – API Reference Only available for Motorola/Kreatel developers

COS/CCS 2008-29