Real-time Communication between Web Browsers

MASTER THESIS

Pavel Smolka

Brno, 2013

Declaration

Hereby I declare, that this paper is my original authorial work, which I have worked out by my own. All sources, references and literature used or excerpted during elaboration of this work are properly cited and listed in complete reference to the due source.

Advisor: doc. RNDr. Tomáš Pitner, Ph.D.

Acknowledgement

Thanks to everyone... TODO: Petr Kunc (help with OpenTok & RealTime communication,

Abstract

TODO

Keywords

Contents

Chapter 1

Introduction

Millions, billions, trillions. That many and even more messages are exchanged every day between various people over the world. The Internet created brand new way to communicate and collaborate, even if you are located on the opposite parts of the world. Since the age of Alexander Graham Bell, the accesibility to the communication devices and their simplicity incredibly enhanced. Nowadays, almost 2.5 billion people over the world have access to the Internet and therefore they are able to use almost limitless communication possibilities it provides. [9]

However, the manner of Internet usage essentially changed during the first decade of 21st century. Using the Internet and using the web browser became almost synonyms. People use the web browser as the primary platform to do every single task on the Internet. Sometimes it's not even possible to use the other internet services without visiting certain web page in the web browser and performing the authentication there.[1] Considering the mentioned fact, web browsers became also the basic platform for the communication tools. Even though the purpose of the world wide web and HTTP protocol was completely different at first (displaying single documents connected via hypertext links), it appeared that there is the need of common rich applications running withing web browser - rich internet application. So popular social networks are built on top of the web browser platform and they are used by more than billion people over the world. [5] And the main reason why the social networks are so popular is the real-time stream of news and messages from the other people. By the beginning of the year 2013, I would say that static web is dead - users prefer interactivity.

This thesis embraces the topic of real-time application in web browsers, especially the text communication tools and the technologies being used to develop them. It also describes the problem of "inter-process" communication between various web pages which need to cooperate and exchange information as quickly as possible. Finally, the possibilities of multimedia content transfer (audio and video) and the current options of capturing multimedia directly from the web browser are described as well.

As mentioned above, the web browser became one of the most popular platforms. Celebrio, simple software for the elderly simulating the operating system interface, is typical

---

1. Two examples of such behavior. Wi-fi network in the Student Agency coaches forces the user to visit the entry page in the web browser. The second example, very well known to the students of the Faculty of Informatics at Masaryk University, is the faculty wireless network called wlan_fi. Every user has to open the web browser and log in with her credentials. It's not possible just to open the terminal or e-mail client and start working online.

example of rich internet application. [3] All the topics mentioned in the previous paragraph appeared to be very important in the system. When questioning the elderly people in the Czech Republic, it appeared that almost 90 % of the elderly computer users use the real-time communication (RTC) applications, mostly Skype. [4] Interaction with their loved ones is the most desired benefit they expect from the computer. Therefore, creating real-time application, text messenger supporting video calling, became not only programming challenge but also a business goal.

Considering the fact the people like real-time communication while using web browser brings us the question what the currently available solutions are. There are "big players" providing their own services as closed-source, without the possibility to use them. To name the most important, it's Google Talk web browser client and Facebook chat, using XMPP protocol. [8][6] Even though Facebook chat service is not pure XMPP server implementation (the message and presence engine is proprietary system of Facebook implemented mostly in C++ and Erlang), they provide the possibility to connect to the "world of Facebook Chat" via XMPP as proxy. [7] Combination of the facts that XMPP is open technology with open-sourced client and server implementations [12] and the big internet companies also use it persuaded us to use it in our communication application too. XMPP itself and its usage in web applications is described in Chapter 2, "Extensible Messaging and Presence Protocol". Other approaches to RTC are covered in Chapter 3, "Other approaches to RTC".

There are many existing real-time chat-based applications over the Internet we could have used. However, none of them suited our needs perfectly. Celebrio has very specific graphical user interface (GUI) and there is a need to integrate both text-based chat and video calling. Just to mention, there is commercial chat module Cometchat[2] or even open project Jappix.[3] Video calling web browser applications are provided for example by Tok-Box Inc.[4] Nevertheless, following the rule that "If you have to customize 1/5 of a reusable component, its likely better to write it from scratch", [2] just very generic existing libraries (Strophe.js) and APIs (OpenTok) were used for implementing brand new application called *Celebrio Talker*. The general approaches when building web browser based chat application are mentioned in Chapter 4, "XMPP client in Javascript". Within the programming part of the thesis, the real-time text chat application, video calling application and simple "inter-process" communication tool for Celebrio has been implemented. Celebrio Talker application itself, its architecture and the specific procedures used to create it are described in Chapter 5, "Talker".

It has been said that Skype[5] is the most favorite communication tool among the target audience. If it had been implemented, the existing customer base could be used and converted to our messaging application. However, there is one big pitfall in this approach. Skype license strictly prohibits incorporating their software into mobile devices. [11] They support only prompting the official Skype client to be opened via Skype URI, which is insufficient

---

2. http://www.cometchat.com/
3. https://project.jappix.com/
4. http://www.tokbox.com/
5. http://www.skype.com/

for Celebrio since the messaging client has to be built in the system, with the corresponding user interface.

Finally, there are several notes about "inter-process communication" between different applications running separately in various browser frames, tabs or even windows. Chapter 6, "Inter-process Communication Framework" covers this topic and describes the issues we came accross when implementing such functionality for Celebrio, where every application runs in separate iframe.

Chapter 2

Extensible Messaging and Presence Protocol

Extensible Messaging and Presence Protocol (XMPP) technologies were invented by Jeremie Miller in 1998. [13] It is one of the most widespread technologies for instant messaging (IM),[1] i.e. exchanging the text or multimedia data between several endpoints. The very essence of every instant messaging is bidirectional stream where so that both sides can immediately *push* new data and the other side (or other sides, respectively) is promptly notified without the need to perform any manual *pull* (update) action. The best transport protocol for such use case is Transmission Controll Protocol (TCP) and the "native" implementation of XMPP works right on top of TCP protocol: XMPP endpoint (called client as it represents the first actor in client-server architecture) opens long-lived TCP connection. Then, both the client and the server negotiate and open XML streams so there is one stream in each direction. [13] When the connection is established, both client and server can push any changes as XML elements to the stream and the other side obtains them immediately.

Nevertheless, the implementation described in the previous paragraph is unavailable when creating common web application. That is the reason why the bidirectional stream emulation is frequently used in many rich internet applications in order to assure interactivity and freshness of the data. Such emulation is called BOSH and it is described closely in Section 2.2. The other approach utilizes HTML5 technology called *WebSockets* and the common XMPP servers support it via various plugins. This approach is covered in Section 2.3.

2.1 Fundamental bulding blocks in XMPP

Mention the terms as stanza, roster, describe the subscription mechanism.

2.2 Bidirectional-streams Over Synchronous HTTP

Describe BOSH (http://xmpp.org/extensions/xep-0124.html)
and XMPP Over BOSH (http://xmpp.org/extensions/xep-0206.html).
Later on, description of BOSH extension, including the advantages and limitations. Describe also the connection to HTTP protocol.

---

1. Acutally, the IM client or even the techology itself is sometimes called "Instant Messenger". This term is registered as a trademark by AOL company. [1]

2.3 XMPP over WebSockets

luajit - lua jit compiler
    It was neccessary to install: apt balíky luajit + liblua5.1-bitop0

2.4 Jingle

Jingle extension - multimedia.
    http://xmpp.org/extensions/xep-0166.html

2.5 Inter-process communication

Pub-sub extensions for inter-process communication.
    http://xmpp.org/extensions/xep-0060.html

Chapter 3

Other approaches to RTC

Describe other approaches to RTC in WB than XMPP. Tell why we didn't use them (or that we used - OpenTok).

Mention:

OpenTok

SignalR (.NET)

WebRTC (http://www.webrtc.org/)

Google hangouts API

Chapter 4

XMPP client in Javascript

Describe the tools that can be used to implement RTC in WB (and which were used to implement Celebrio Talker)
  Strophe (simple XMPP in Javascript)
  Strophe connecting/attaching - security issues. TODO programming
  Strophe plugins
  Possible server-side implementations (JAXL, XMPPHP, ...)

Chapter 5

Talker

Describe the Talker application in Celebrio.

Mention what we expected from the app (value proposition)

Then, application analysis, design and implmementation.

Describe the architecture and used tools&frameworks: JS + Ember.js, OpenTok, Web-Sockets in new OpenTok

Don't forget to use UML: use case diagram, class diagram (if any), sequence/action diagram

5.1   Ember.js

Describe Javascript client-side MVC frameworks overall, compare, tell why we used Ember

5.1.1 Subsection about Ember...

Chapter 6

Inter-process Communication Framework

Implement and describe framework for inter-process communication in Celebrio.

First, the lightweight one (which we already have), then implement the "heavy" one, if there's enough time.

Chapter 7

Conclusion

conclusion

7.1 Another part of the conclusion

Another part of the conclusion... just to have subchapter here

Bibliography

[1] AOL Inc.: *AOL Trademark List*, 3/15/2011 [retrieved 2/20/2013], from <http://legal.aol.com/trademarks/>. 1

[2] Staatss, B. (Brianstaats): *"If you have to customize 1/5 of a reusable component, its likely better to write it from scratch @trek at #embercamp"*, 2/15/2013 [retrieved 2/23/2013], Tweet. 1

[3] Donko, P. and Kunc, P. and Novák, M. and Smolka, P. and Volmut, J.: *Celebrio System*, 2013 [retrieved 2/19/2013], from <http://www.celebriosoftware.com/celebrio-system>. 1

[4] Smolka, P. and Novák, M.: *Elderly people and the computers*, 2/11/2013 [retrieved 2/19/2013], from <http://infogr.am/Seniori-a-pocitace>. 1

[5] Olanoff, D.: *Facebook Announces Monthly Active Users Were At 1.01 Billion As Of September 30th*, TechCrunch, 10/23/2012 [retrieved 2/19/2013], from <http://techcrunch.com/2012/10/23/facebook-announces-monthly-active-users-were-at-1-01-billion-as-of-septem 1

[6] Facebook Developers: *Facebook Chat API*, 2/12/2013 [retrieved 2/20/2013], from http://xmpp.org/about-xmpp/history/ <http://legal.aol.com/trademarks/>. 1

[7] Letuchy, E.: *Facebook Chat*, 5/14/2008 [retrieved 2/23/2013], from <https://www.facebook.com/note.php?note_id=14218138919>. 1

[8] Google Developers: *Google Talk Developer Documentation*, 3/23/2012 [retrieved 2/20/2013], from <https://developers.google.com/talk/talk_developers_home>. 1

[9] Miniwatts Marketing Group: *Internet Users in the World - 2012 Q2*, Internet World Stats, 2/17/2013 [retrieved 2/19/2013], from <http://www.internetworldstats.com/stats.htm>. 1

[10] Ward, J.: *What is a Rich Internet Application?*, 10/17/2007 [retrieved 2/19/2013], from <http://www.jamesward.com/2007/10/17/what-is-a-rich-internet-application/>.

[11] Smith, A.: *Does SkypeKit work on Android?*, 8/7/2012 [retrieved 2/23/2013], from <http://devforum.skype.com/t5/SkypeKit-FAQs/Does-SkypeKit-work-on-Android/m-p/16490/thread-id/78>. 1

[12] The XMPP Standards Foundation: *History of XMPP*, 1/27/2010 [retrieved 2/20/2013], from http://xmpp.org/about-xmpp/history/ <http://legal.aol.com/trademarks/>. 1

[13] Saint-Andre, P. and Smith, K. and Tronçon, R.: *XMPP: The Definitive Guide*, Sebastopol: O'Reilly, 2009, 978-0-596-52126-4, 310 (7, 16, ...). 2

Index

Appendix A

Screenshots of the application

Some screenshots from Celebrio Talker