# Web Call SDK 2.0

A dissertation submitted to the Royal Institute of Technology (KTH) in partial fulfillment of the requirements for the degree of Master of Science

SHANBO LI

Master's Thesis at ERICSSON AB
Supervisor at Ericsson: Peter Yeung
Supervisor at KTH: Mihhail Matskin, PhD
Examiner: Mihhail Matskin, PhD

TRITA xxx yyyy-nn

## Abstract

This is a skeleton for KTH theses. More documentation regarding the KTH thesis class file can be found in the package documentation.

*To my parents, LI Chongzhi and WU Wei, who have guided me through life and encouraged me to follow my own path, and to my wife, MEI Dan, for being waiting for me and keeping faith in me.*

# Contents

# Acknowledgements

... acknowledgement text ...

## Chapter 1

# Background

### 1.1 Introduction

Voice over Internet Protocol (VoIP) enables the communications between Internet users and the endpoints in PSTN circuit switched (CS) networks. As we know, Session Initiation Protocol (SIP) is widely adopted as a signaling protocol for VoIP communications. Because of its simplicity, power and extensibility, it has also been selected by the Third Generation Partnership Project (3GPP) as a major component of IP Multimedia Subsystem (IMS) for the evolved UMTS core network.

As the world-leading supplier in telecommunications, Ericsson realizes that the position of VoIP technology in the future of telecom industry is absolutely outstanding. A project named Web Call SDK is planned by the department of Ericsson Developer Connection, to create an application that based on VoIP and make phone to phone calls.

### 1.2 Task

The task is to take all of the benefits of VoIP and create a powerful application for phone calls. The application should be simple, stable, reusable, extendable, integratable, easy to access and user friendly.

It should be a splendid application that supplies the function for users to make phone call anytime, anywhere and to anyone in this world.

## 1.3 Terminology

### Java

The Java$^{TM}$ programming language is a popular high-level language which provides a portable feature and can be used on many different operating systems.

The source code of Java is first written in plain text files which ends with the `.java`. Then the Java source files are compiled into bytecodes which ends with `.class` by Java compiler (javac). A `.class` file is platform independent. It will be executed by the Java Virtual Machine*.[1]

A Java application can be distributed in the format of Java ARchive (JAR) file.

### Java EE

Java Platform, Enterprise Edition (Java EE) is a set of coordinated technologies that significantly reduces the cost and complexity of developing, deploying, and managing multitier, server-centric applications.[10]

### Java ME

Java Platform, Micro Edition (Java ME) is a collection of technologies and specifications to create a platform that fits the requirements for mobile devices such as consumer products, embedded devices, and advanced mobile devices.[11]

### MIDlet

A MIDlet is a Java application framework for the Mobile Information Device Profile (MIDP) that is typically implemented on a Java-enabled cell phone or other embedded device or emulator.

### JAD

The Java Application Descriptor (JAD) file, as the name implies, describes a *MIDlet* suite. The description includes the name of the MIDlet suite, the location and size of the JAR file, and the configuration and profile requirements. The file may also contain other attributes, defined by the Mobile Information Device Profile (MIDP), by the developer, or both.[9]

### SSL

Secure Sockets Layer (SSL), is a cryptographic protocol which provides security and data integrity for communications over networks such as the Internet.[18]

*The terms “Java Virtual Machine” and "JVM" mean a Virtual Machine for the Java platform. [1]

### SIP

Session Initiation Protocol (SIP) is an application-layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants. These sessions include Internet telephone calls, multimedia distribution, and multimedia conferences.[15]

SIP is the format of control signal in VoIP. It describes the sender, receiver. A agent use SIP messages to register on a proxy, establish session or close session.

### SDP

SDP is short for Session Description Protocol. It is intended for describing multimedia sessions for the purposes of session announcement, session invitation, and other forms of multimedia session initiation.[6]

A SIP message may carry a SDP message. The SDP message contains protocol version, session name, information, and most important, the connection data and media descriptions. This supplies a way to manipulate the connection of media flow.

### RTP

RTP, the real-time transport protocol, provides end-to-end network transport functions suitable for applications transmitting real-time data, such as audio, video or simulation data, over multicast or unicast network services.[16]

### Mobile Front Controller

Mobile Front Controller (MFC) is a light-weight Java EE web application framework for creating web applications for web browsing and mobile browsing. [20]

The mobile front controller uses a sevlet to handle http request, and redirect request to different kind of view. All views share a same logic.

### Web Service

A web service is defined by the W3C as “a software system designed to support interoperable machine-to-machine interaction over a network”.[7]

### SOAP

SOAP is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment. It uses XML technologies, an extensible messaging framework containing a message construct that can be exchanged over a variety of underlying protocols.[5]

### WSDL

WSDL is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information.[3]

## 1.4 About

**Web Call SDK *2.0*** is a open source project at Ericsson Developer Connection[†] (EDC), Ericsson[‡]. It is the successor project of **Web Call SDK**[2] which is developed by Yuening Chen at Ericsson AB during the year 2007 and 2008. After Yuening finished Web Call SDK, the people at EDC found it is not stable enough and many of the functions are not usable. So they decided to start a new project follow Web Call SDK to make the it stable and add some new feature to it. The new project is called **Web Call SDK *2.0***. As the main developer, I took over the new project at March 2008 and finished it at February 2009. I rewrite most code of the old project to make it stable and runnable and added some new feature to it. As a result, the **Web Call SDK *2.0*** is used as the base library of Ericsson's demo of **Using REST and Web Services to Mash Up Communications Capabilities**[4][13] at JavaOne™[§] 2009.

[†] Ericsson Developer Connection (former Ericsson Mobility World Developer Porgram) is a department of Ericsson. It helps developers to create applications that incorporate telecommunication network capabilities, such as location-based services, charging, messaging and presence, with sustainability in mind.

[‡] Ericsson is a world-leading provider of telecommunications equipment and related services to mobile and fixed network operators globally.

[§] JavaOne is an annual conference (since 1996) put on by Sun Microsystems to discuss Java technologies

## Chapter 2

# Requirement

### 2.1 Programming Language

To make the application potable, **Java** is chosen as the programming language of Web Call SDK. The Java programming language is a popular high-level language which provides a portable feature and can be used on many different operating systems. For the introduction and more detail of Java please refer to 1.3.

### 2.2 Simple

The word “Simple” means, for community developers, the Web Call SDK should supplies a set of API that are easy to understand and convenient to use. The developers who use this API do not need much experience on java language and deep understanding of VoIP technology.

### 2.3 Stability

The application should be stable and has as less bugs as possible. The application should be designed for deploying on a server for long term use. The concurrent request users may more than one hundred.

### 2.4 Reusability

The code should be made as generic and reusable. The interface should not constrain on any specific network or service provider. It should follow a common accepted standard. Session Initiation Protocol (SIP) is a signalling protocol, which defined in RFC 3261 SIP: Session Initiation Protocol [15], widely used for multimedia communication sessions such as voice and video calls over the Internet. It should be used as the main signal protocol of Web Call SDK.

## 2.5 Extendibility

The application should be able to add new feature according to customer's requirement, e.g. add video call and instant message.

## 2.6 Integration

The Web Call SDK should supply a web service API that can be used by other applications. This interface should contain most of the functions of Web Call SDK. And can be easily used on Web 2.0[*] mashup[†].

[*] "Web 2.0" refers to a perceived second generation of web development and design, that facilitates communication, secure information sharing, interoperability, and collaboration on the World Wide Web.[19] See also: What Is Web 2.0[12]

[†] Mashup is a Web application that combines data or functionality from two or more sources into a single integrated application.

## Chapter 3

# Background Study

## 3.1 VoIP Market

In the telecom market, VoIP technology has gained more and more customers. The advantage of VoIP is obviously, much cheaper fee and almost same quality as traditional telephone. Report from Infonetics indicates that, in the year 2007, the subscribers for VoIP are under 80 all around world. Most of them are in the Asia Pacific region. However by the year of 2011 the user will be 135 million, predicted by MarketResearch.com. And a UK research company Disruptive Analysis Ltd. predicts the users of mobile-VoIP will be 250 million by the year of 2012.[8]

The analyses and figures above draw a brilliant future of VoIP market.

### 3.1.1 VoIP Service Provider

VoIP service provider is the company which supplies the products of VoIP/PSTN gateway. Or the ones who supply the service that customers can call a PSTN phone by a VoIP phone via their service/network. There are hundreds of such companies in the world. PSTN is short for Public Switched Telephone Network. It is the network of the world's public circuit-switched telephone networks. In another word, it is just the traditional phone network.

### 3.1.2 VoIP Client

A VoIP client is a common SIP client software or a IMS client software. This kind of software runs on a computer or mobile device and implements the SIP or/and IMS standard. It can work like a phone to dial or answer VoIP calls.

### 3.1.3 Solution Provider

A solution provider is a company that supplies both VoIP service and software client, such as Skype™[*], VoipStunt[†] and JAJAH[‡]. Among them Skype is the most famous one. It has a very good quality of voice and functionality client. However, the Skype is not following the standard of SIP. So it means, only the Skype client itself can use the service of Skype. VoipStund and JAJAH supply relevant lower fee and less quality of audio.

## 3.2 Third Party Call Control

In the traditional telephony context, third party call control allows one entity (which we call the controller) to set up and manage a communications relationship between two or ore other parties. Third party call control (referred to as **3pcc**) is often used for operator services (where an operator creates a call that connects two participants together) and conferencing.[14]

A general work flow of 3pcc is shown in Figure 3.1. The initial side of the phone is the *controller*. The *controller* sends a signal "connect *client A* and *client B*" to the server ①. And the server establishes a call between *client A* and *B* ②.

## 3.3 Why Web Call SDK?

Based on IMS/SIP technology, Web Call SDK integrated call functions into Web containers. This presents a simple way to implement the communication convergence of Web, IMS/SIP network, and CS networks. It does not require the installation of the plug-in clients on the browser or other special client software as most VoIP services.

The Web Call SDK is nether a service provider or a client that described above. It is more like a controller which acts as a initial side in third party call control. That is, it support all standard client and service provider. Another advantage of Web Call SDK is that The desktop view, mobile browser view and JavaME client all share a same database. The user can access a same contact book and use a same service account from different platform. None of the solution provider or client have the same function.

[*] http://www.skype.com
[†] http://www.voipstunt.com
[‡] http://www.jajah.com

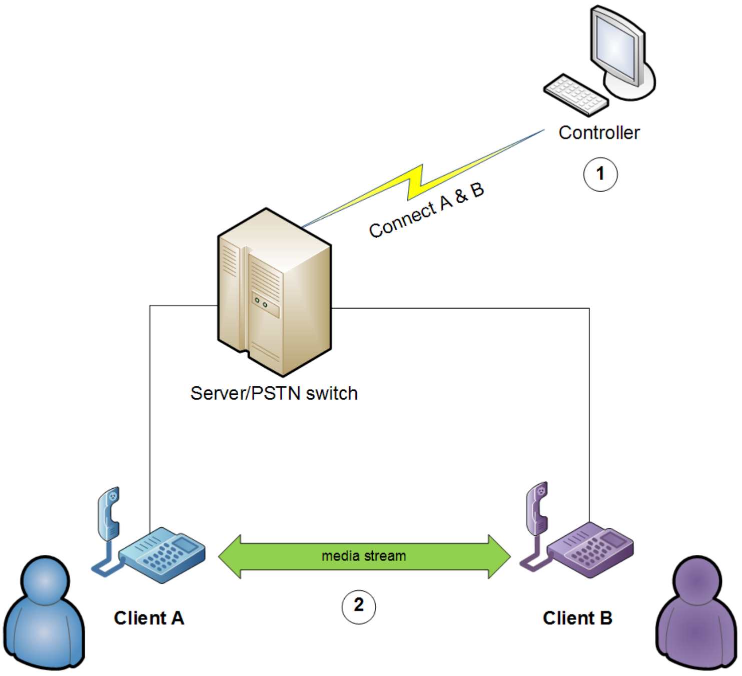

**Figure 3.1.** Third party call control

Chapter 4

# Solution

There are two kind of connection solution of the core of Web Call, the Relay Call and Third Party Call. The different between them is the way they handle media stream.

The Relay Call Controller works as a back-to-back agent and forwards media streams, while Third Party Call Controller only establishes connections by sending out SIP messages and it does not handles any streams itself.

## 4.1 Relay Call

The signal and media flow of Relay Call is shown in Figure 4.1. In this scenario, the Web Call Example Application acts as a back-to-back user agent. It sets up the connection and forwards the media stream. It can be seen from the picture that both signal and media are handled by Web Call Server. When it starts, it try to call client A. After it establishes a session with client A, it will try to call client B and also establish a session with client B. After that, it will work as a media stream bridge and forward media stream from client A to B, as well as from client B to A.

For a detail description and mechanism of Relay Call, please refer to the master thesis of **Web Call SDK** by *Yuening Chen*[2].

## 4.2 Problem of Relay Call

### 4.2.1 The Load on Controller

The controller here acts as a back to back user agent (B2BUA). It receives the RTP flow from one client and transfers it to another client. It does the same in the opposite direction. That means all of the RTP traffic will go through the controller. So the load on controller will be heavier as with the number of concurrent users

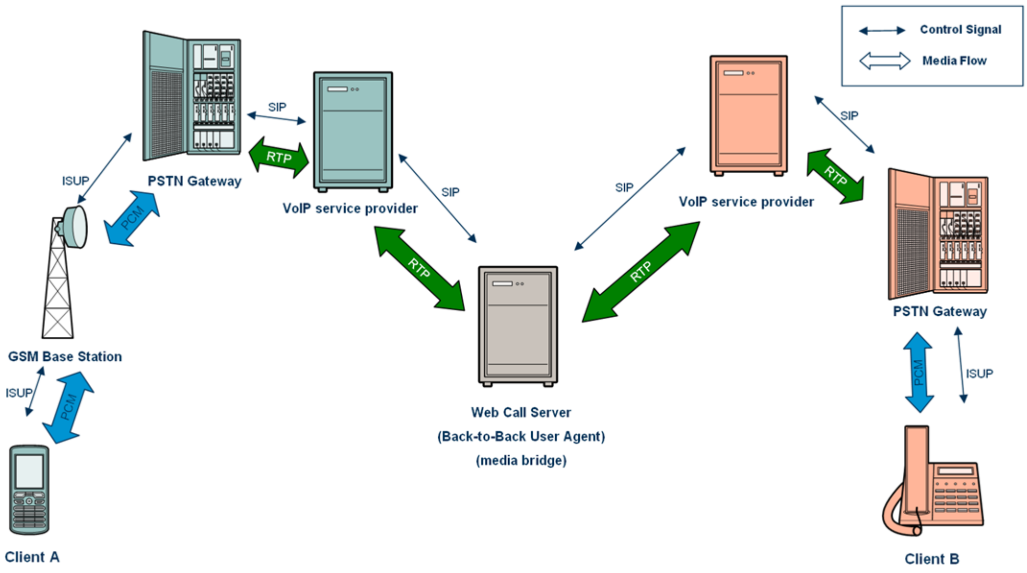

**Figure 4.1.** The signal and media flow of Relay Call

increases. A powerful host is needed to handle the RTP flows. However the design goal of Web Call SDK was simply a tool kit that can easily be integrated into a web site. Unfortunately, this current mechanism of session control will decrease the performance of whole web site.

### 4.2.2 The Latency of Audio

As can be seen from the session flow, the RTP goes from one client to the controller via a SIP provider and then the controller transfers the RTP to another client via the SIP provider again. This means that each direction of RTP will go through SIP provider twice. So the latency of the phone call via the SIP provider will be double that of normal calls. The latency of phone-to-phone call via Web Call SDK test turned to be more then 2 seconds. It is quite unacceptable.

### 4.2.3 Lack of Reliability

Since all of RTP flows go through the controller, thus if the controller crashes, all of calls will be immediately interrupted.

## 4.3 Third Party Call

In the traditional telephony context, third party call control allows one entity (which we call the controller) to set up and manage a communications relationship between two or more other parties. Third Party call control (referred as 3pcc) is often used for operator services (where an operator creates a call that connects two participants together) and for conferencing. The signal and media flow in third party call is show in Figure 4.2 The advantage of third party call in Web Call is that the controller only need to handle message transfer and leaves the media flow for the ISP.

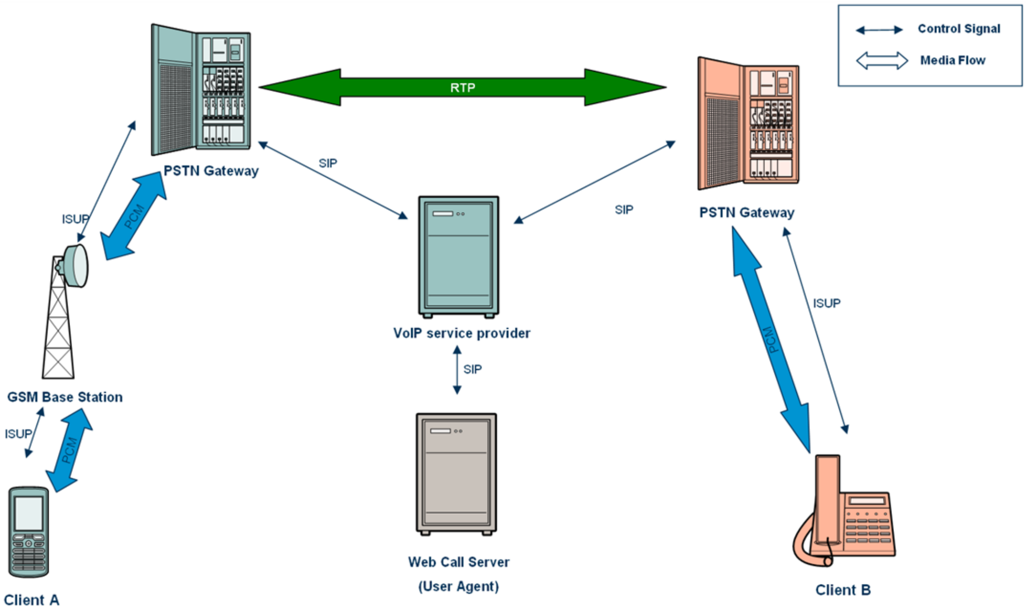

**Figure 4.2.** The signal and media flow of Third Party Call

### 4.3.1 Call Transfer

The call transfer implementation use a `REFER` method which defined in *The Session Initiation Protocol (SIP) Refer Method* (RFC 3515)[17]. “The `REFER` method indicates that the recipient (identified by the Request-URI) should contact a third party using the contact information provided in the request”[17].

The Call flow is shown in Figure 4.3. The controller first sends an `INVITE` to client A (1). This invite is just a normal invite. A’s phone rings and answers. This results in a `200 OK` (2). The controller then answer client A an `ACK` (3). Follow that, the controller send out a `REFER refer-to`: Client B (4), which means controller wish client A to make a phone call to client B. The client A understands that and returns a `200 OK` to controller (5). Then client A sends an `INVITE referred-By`: C to client B (6). Client A and Client B can establish a session according the `INVITE` from A to B. The `BYE` (8) and `200 OK` (9) means the controller cut the media stream between itself and client A.

To accomplish the whole call flow, client A must support RFC3515.

```
Controller               Client A                Client B
    |                        |                        |
    |   (1) INVITE           |                        |
    |----------------------->|                        |
    |                        |                        |
    |      (2) OK            |                        |
    |<-----------------------|                        |
    |                        |                        |
    |   (3) ACK              |                        |
    |----------------------->|                        |
    |                        |                        |
    |   (4) REFER TO B       |                        |
    |----------------------->|                        |
    |                        |     (5) INVITE         |
    |                        |----------------------->|
    |                        |                        |
    |                        |      (6) OK            |
    |                        |<-----------------------|
    |                        |                        |
    |                        |     (7) ACK            |
    |                        |----------------------->|
    |                        |                        |
    |                        |      (8) RTP           |
    |                        |........................|
    |                        |                        |
```

**Figure 4.3.** The signal and media flow of Call Transfer

### 4.3.2 SDP Swap

This implementation of third party call control swaps SDP from two clients. The prototype of SDP swap comes from *Best Current Practices for Third Party Call Control (3pcc) in the Session Initiation Protocol (SIP)* (RFC 3725). However, the call flow in RFC 3725 is quite complicated. The concept of **SDP Swep** is that controller calls to client and swaps two SDP which got from clients. These two clients seem to call controller but the media stream is connected directly between them.

The call flow of SDP swap implementation of third party call control is shown in Figure 4.4. Controller first call client A. This `INVITE` has no session description. Client A's phone rings, and client A answers. It results a `200 OK` (3) that contains an offer which contains client A's SDP [15]. Meanwhile, the controller does the same to client B (2) and gets client B's offer (4) which contains client B's SDP. So far, controller has both client A and client B's SDP. It just simply swaps these two SDPs, then `ACK` client B with client A's SDP (5) and `ACK` client A with client B's SDP (6). So client A gets client B's SDP by (6) and client B gets client A's SDP by (5). Therefore, media flows between client A and client B (7).

```
Client A                 Controller                Client B
    |                        |                        |
    |                        |                        |
    |(1) INVITE without SDP  | (2) INVITE without SDP |
    |<-----------------------|----------------------->|
    |                        |                        |
    |                        |                        |
    |(3) OK with SDP         |                        |
    |----------------------->|  (4) OK with SDP       |
    |                        |<-----------------------|
    |                        |                        |
    |                        |                        |
    |                        |(5) ACK with A's SDP    |
    |(6) ACK with B's SDP    |----------------------->|
    |<-----------------------|                        |
    |                        |                        |
    |                        |                        |
    |                  (7) RTP                        |
    |.................................................|
    |                                                 |
```

**Figure 4.4.** The signal and media flow of SDP Swap

### 4.3.3 Re-invite

A `Re-INVITE` is used to change the session parameters of an existing or pending call. It uses the same Call-ID, but the CSeq is incremented because it is a new request. The Re-invite implementation send two client's SDP to each other in the Re-invite process.

The call flow is shown in Figure 4.5. From the SIP message (1) to (6), the controller uses a three way handshake to establish connections with client A and client B. In (3) and (4) the controller gets client A and client B's SDP which are going to be used in the Re-invite phase in (7) and (8). The controller sends a `Re-INVITE` to A with B's SDP (7), which indicates the controller changes its media port to B's. At the moment, controller also send a `Re-INVITE` to client B with client A's SDP, which indicates the controller change its media port to A's. Both client A and client B gets each other's SDP, thus, a media stream could be established between A and B (13).

```
Client A              Controller              Client B
    |                      |                      |
    |(1) INVITE            | (2) INVITE            |
    |<---------------------|---------------------->|
    |                      |                      |
    |(3) OK with SDP       |                      |
    |--------------------->|  (4) OK with SDP     |
    |                      |<---------------------|
    |                      |                      |
    |                      | (5) ACK              |
    |            (6) ACK   |--------------------->|
    |<---------------------|                      |
    |                      |                      |
    |   (7) RE-INVITE with | (8) RE-INVITE with   |
    |        B's SDP       |     A's SDP          |
    |<---------------------|---------------------->|
    |                      |                      |
    |   (9) OK             |           (10) OK    |
    |--------------------->|<---------------------|
    |                      |                      |
    |            (11) ACK  |  (12) ACK            |
    |<---------------------|---------------------->|
    |                      |                      |
    |                                             |
    |                  (13) RTP                   |
    |.............................................|
    |                                             |
```

**Figure 4.5.** The signal and media flow of SDP Swap

### 4.3.4 Web Client

## 4.4 Conclusion

<table>
<tr><th rowspan="3"></th><th rowspan="3">PSTN</th><th colspan="5">VoIP</th></tr>
<tr><th rowspan="2">Relay Call</th><th colspan="4">Third Party Call</th></tr>
<tr><th>Call Transfer</th><th>SDP Swap</th><th>Re-Invite</th><th>Web Client</th></tr>
<tr><td rowspan="2">**service provider**</td><td rowspan="2">All PSTN switch</td><td rowspan="2">support by all VoIP service providers</td><td colspan="4">Not support by all VoIP service providers</td></tr>
<tr><td>The ones who support REFER method</td><td>The ones who do not filter SIP message which doesn’t carry SDP</td><td>The ones who support Re-Invite message</td><td>The ones who supply web site call</td></tr>
<tr><td rowspan="2">**Client**</td><td rowspan="2">Traditional telephone or mobile phone</td><td rowspan="2">All clients (traditional phone, mobile phone, software-client)</td><td colspan="3">Traditional phone, mobile phone, software-client, under particular requirements of software-client or PSTN gateway</td><td rowspan="2">All clients (traditional phone, mobile phone, software-client)</td></tr>
<tr><td>Client need support REFER method</td><td>Client need implement RFC 3725</td><td>Client need support Re-Invite</td></tr>
<tr><td>**Media Stream**</td><td>In PSTN network</td><td>Internet and/or PSTN, need to be handled on controller</td><td colspan="4">Internet and/or PSTN</td></tr>
<tr><td>**Latency**</td><td>Very Short / QoS guarantee</td><td>long</td><td colspan="4">Short, acceptable. But no QoS guarantee</td></tr>
<tr><td>**Cost**</td><td>Expensive (especially for international call )</td><td colspan="5">cheap</td></tr>
</table>

Chapter 5

# Web Call Architecture

Chapter 6

# SIP Call Component

Chapter 7

# Web Application

Chapter 8

# Web Service Interface

## Chapter 9

# Web Service Client

## Chapter 10

# Conclusion

# Bibliography

[1] Mary Campione and Kathy Walrath. The java programming language. http://java.sun.com/docs/books/tutorial/getStarted/intro/definition.html, February 2008. [cited at p. 4]

[2] Yuening Chen. *Web Call SDK*. Uppsala Universitet, March 2003. [cited at p. 7, 15]

[3] Erik Christensen, Francisco Curbera, Greg Meredith, and Sanjiva Weerawarana. Web Services Description Language (WSDL) 1.1. http://www.w3.org/TR/wsdl, March 2001. [cited at p. 6]

[4] Elena Fersman and Peter Yeung. Using REST and Web Services to Mash Up Communications Capabilities. In *JavaOne 2009*, June 2009. [cited at p. 7]

[5] Martin Gudgin, Marc Hadley, Noah Mendelsohn, Jean-Jacques Moreau, Henrik Frystyk Nielsen, Anish Karmarkar, and Yves Lafon. Soap Version 1.2. http://www.w3.org/TR/soap12, April 2007. [cited at p. 5]

[6] M. Handley, V. Jacobson, and C. Perkins. SDP: Session Description Protocol , RFC 4566. http://www.ietf.org/rfc/rfc4566.txt, July 2006. [cited at p. 5]

[7] Hugo Hass and Allen Brown. Web services glossary. http://www.w3.org/TR/ws-gloss/, February 2004. [cited at p. 5]

[8] Paul D. Kretkowski. State of the voip market 2008. http://www.voip-news.com/feature/state-voip-market-2008-031008/, March 2008. [cited at p. 11]

[9] Richard Marejka. Learning Path: MIDlet Life Cycle. http://developers.sun.com/mobility/learn/midp/lifecycle/, February 2005. [cited at p. 4]

[10] Sun Microsystems. Java EE Technology. http://java.sun.com/javaee/technologies/, June 2009. [cited at p. 4]

[11] Sun Microsystems. Java ME Technology. http://java.sun.com/javame/technology/, June 2009. [cited at p. 4]

[12] Tim O'Reilly. What is web 2.0.
http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html, September 2005. [cited at p. 10]

[13] Benny Ritzén. JavaOne: Come to Ericsson's BOFs and technical session and be inspired .
http://www.ericsson.com/developer/sub/articles/other_articles/090514_javaone_bof, May 2009. [cited at p. 7]

[14] J. Rosenberg, J. Peterson, H. Schulzrinne, and G. Camarillo. Best Current Practices for Third Party Call Control (3pcc) in the Session Initiation Protocol (sip), RFC 3725.
http://www.ietf.org/rfc/rfc3725.txt, April 2004. [cited at p. 12]

[15] J. Rosenberg., H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol, RFC 3261.
http://www.ietf.org/rfc/rfc3261.txt, June 2002. [cited at p. 5, 10, 19]

[16] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications, RFC 3550.
http://www.ietf.org/rfc/rfc3550.txt, July 2003. [cited at p. 5]

[17] R. Sparks. The Session Initiation Protocol (SIP) Refer Method.
http://www.ietf.org/rfc/rfc3515.txt, April 2003. [cited at p. 17]

[18] Wikipedia. Transport layer security.
http://en.wikipedia.org/wiki/Secure_Sockets_Layer, June 2009. [cited at p. 4]

[19] Wikipedia. Web 2.0.
http://en.wikipedia.org/wiki/Web_2.0, June 2009. [cited at p. 10]

[20] Peter Yeung and Pär Johansson. Mobile Front Controller.
http://www.ericsson.com, March 2009. [cited at p. 5]

# Appendices

## Appendix A

# Appendix title

... some text ...

# List of Symbols and Abbreviations

| Abbreviation | Description | Definition |
|---|---|---|
| 3GPP | 3rd Generation Partnership Project | page 3 |
| 3pcc | Third Party Call Control | page 12 |
| CS | Circuit Switched | page 3 |
| GSM | Global System for Mobile communications | page **??** |
| IMS | IP Multimedia Subsystem | page 3 |
| IP | Internet Protocol | page **??** |
| ISDN | Integrated Services Digital Network | page **??** |
| ISUP | ISDN User Part | page **??** |
| JDK | Java Development Kit | page **??** |
| JRE | Java Runtime Environment | page **??** |
| JVM | Java Virtual Machine | page 4 |
| PSTN | Public Switched Telephone Network | page 11 |
| QOS | Quality Of Service | page **??** |
| SDP | Session Description Protocol | page 5 |
| SIP | Session Initiation Protocol | page 5 |
| UMTS | Universal Mobile Telecommunications System | page **??** |
| VoIP | Voice over Internet Protocol | page 3 |

# List of Figures

# List of Tables