**User Guide**

**jWebSocket**

**Audio and Video transmission  
using the jWebSocket Framework**

**1.0**

**Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 24/04/2012 | 1.0 | Document Creation | Alexander Rojas Hernández |

**1. Characteristics of the solution**

The transmission of audio and video using jWebSocket framework guarantee a real time flow communication under a WebSocket Protocol.

The systems that currently uses streaming technology are based in a Client-server (C2S) structure using Real Time Protocol(RTP) as a protocol communication, which provides a good images and sound synchronization. However the use of RTP for this technology has disadvantages: It is not guaranteed that the packets send from the server are continuously received. Also, the disorder of the received packets cannot be avoided . In spite of the RTP is supported by UDP to make a fast send of the information, does not guarantee high levels of precision, availability and security.

The demonstrative application for audio and video transmission realized for Web is using the Java Media Framework (JMF) and a native application for the Smartphone is using PhoneGap, both applications use the jWebSocket Server that allows to receive encoded packets from the client and the broadcasting to all clients that are connected. With the native application it can access to the Smartphone video, audio and image controllers. The packets are processed in the client and send to the server for the broadcast; the other clients receive the packets and decode them for visualization. In the stationary application the process is similar. These applications guarantee a real time using WebSocket Protocol and a bidirectional work with high precision, beside that they guarantee the security and availability of the information.

The aspects mentioned before: real time and bidirectional communication supports a new Web Communication Paradigm. It is possible using the jWebSocket framework, a new technology oriented to the Web application development over the WebSocket communication protocol.

**2. Mean features**

The stationary application developed for the audio and video transmission using the jWebSocket framework presents the following functionalities:

* Capture of video from the webcam
* Visualize the content using a java applet
* Encode the capture data
* Send the data to the server
* Broadcast the data in the server
* Receive and decode the data at the clients

The native application developed for the audio and video transmission using the jWebSocket framework presents the following functionalities:

* Capture the image, audio and video controllers from the Smartphone.
* Encode the capture data
* Send the data to the server
* Broadcast the data in the server
* Received and decode the data
* Visualize the data

**3. Problems to be solved**

|  |  |
| --- | --- |
| **Problem** | **Contributions of the solution** |
| The audio and video transmission does not realize in real time using WebSocket protocol. Why is that a problem? | The application allows the audio and video transmission in real time using the WebSocket protocol, could be adapted to another clients.  Describe here what problem was fixed. |
| The availability of the information is affected in the transmission when used a RTP. True, but what are the negative consequences? E.g. “Parts of a video get lost a the client” | The application allows that doesn´t lose packets in the transmission, the server always controls the client connection.  And the benefits for the user? |

**4. Glossary of Terms**

**Streaming:** Technology that allows to transfer encoded packets over the network, processing in a regular and continue flow. Afterwards the packets are decoded and visualized at the client.

**Real Time in the Web:** Technologies and practices that allows to the users received information as soon that are published by the authors instead to checks the source continuously.

**Websocket:** The WebSocket protocol defines the procedures to update the HTTP connection to a full-bidirectional TCP WebSocket connection. The client sends a HTTP GET request to establish a WebSocket communication to the server. Once the connection is established the communication remains active until it is explicitely closed, allowing to exchange messages between the client and the server.

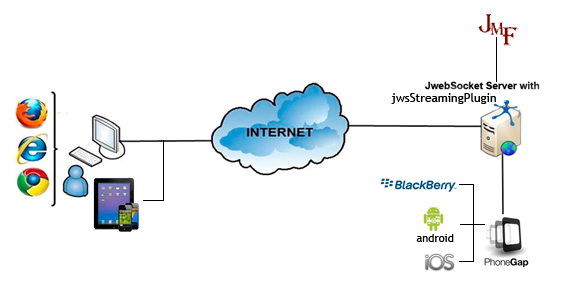
**jWebSocket:** New technology oriented to the application development based on the WebSocket protocol that gives high levels of speed, scalability, security and real time work, main element to the network.

**JMF:** Java Media Framework, this is a framework that allows the work with the audio and video functionalities from a PC (e.g. Webcams), by Java applications and applets.

**PhoneGap:** PhoneGap is an Open Source framework to create native applications for Smartphone’s, using HMTL, CSS and JavaScript, it is supported by different Operative Systems like iOS, Android, BlackBerry, Palm webOS, y Symbian WRT (Nokia).

**5. Solution Model**

Example:

****

Hmm, I don’t understand this diagram completely. Is it correct, that the SmartPhone is the Video provider here and is directly connected to the server, if so all is fine, just check please. In addition some text in the diagram would help to understand more easy. E.g. “Client”, “Videostream server” or similar.

**7. Requirements for use**

The application has the advantage to work on various Operating Systems (for the moment, only tested in Windows), it is needs the following elements:

* OpenJDK 1.7 or higher installed. What about Java JDK 1.6 and 1.7?
* OpenJRE 1.5 or higher installed in the client side (Stationary Application). What about Java JRE 1.5, 1.6 and 1.7?
* Specify the environment variables for the JMF library and the jWebSocket server (Stationary Application).
* jWebSocket Server needs to run.
* Broswer that support WebSocket (Stationary Application). Consider Flash and Comet fallbacks here!
* Webcam for the video capture process
* Smartphone (Android 2.3).

**8. Solution roles**

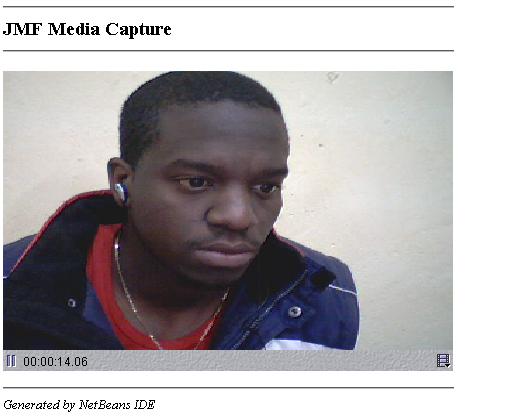
**System User:** Is able to start the capture process and the transmission to the server. Why does the user need to know about his role here, can he doe anything with that?

**9. System functionality**

**Main view for the stationary application**

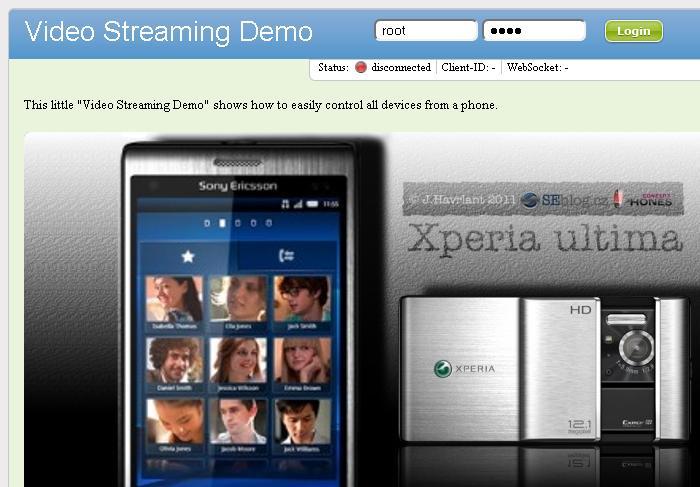
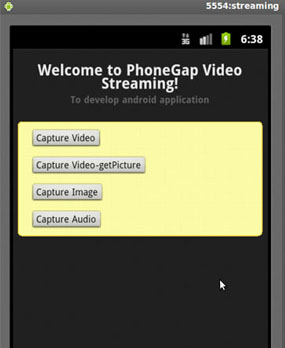
How do I start this view? Which executable/batch or what to run here?

**1**

****

1. Visualization in a java applet the capture using Java Media Framework.

1. **Main view for the native application**

****

**5**

**4**

**3**

**2**

**1**

1. Access to the video capture controller in the Smartphone.
2. Search a video stored in the SmartPhone.
3. Access to the image capture controller in the Smartphone.
4. Access to the audio capture controller in the Smartphone.
5. Player to show the content sent from the server.

**9. Solution Configurations**

Para poner en funcionamiento la aplicación estacionaria, se le debe de especificar en las variables de entorno la utilización de la librería jmf.jar. Please translate!

For the stationary application configure in the environment variables for JMF and jWebSocket server.

|  |
| --- |
| *$JMFHOME/C:/ARCHIV~1/JMF21~1.1E/lib/jmf.jar*  JWEBSOCKET\_HOME/C:/jWebSocket/ |

The configuration can be realized for the system users

**10.Solution Rules**

**R1:** If the PC on which the jWebSocket server is operated is not started correctly the capture process cannot be initialized. The clients has to be configured the JVM in the browers with supports for Websocket.

**R2**: The dispositive for the audio and video capture (Stationary application). Don’t understand that. What’s the problem and what is the solution?

**R3:** The native application must be present the WIFI connection. Solution? What if not?