Hol

**Estudi**

client: FUOC

projecte: Gravació amb Wowza

data: 15/01/2014

# CONFIDENCIALITAT

Drets d’Autor

© 2013, 3&Punt Solucions Informàtiques SL Tots els drets reservats.

# ÍNDEX

CONFIDENCIALITAT 2

ÍNDEX 3

INTRODUCCIÓ 4

Wowza lib 4

Flash JS 5

De Flash a JS 5

De JS a Flash 5

Wowza Eclipse plugin 5

# 

# INTRODUCCIÓ

Per tal de gravar el videochat a Wowza busquem informació de les apps actuals

/usr/local/WowzaStreamingEngine/conf/videochat/Application.xml

/usr/local/WowzaStreamingEngine/conf/webcamrecored/Application.xml

L’única diferència és que en

**Videochat**

<Streams><StreamType>live-*lowlatency*</StreamType>...</Streams>

**Webcamrecorder**

<Streams><StreamType>*default*</StreamType>...</Streams>

Aleshores buscant per internet trobem que un tòpic “[Record videochat](http://www.wowza.com/forums/showthread.php?12813-Record-videochat)”

El que recomanen és canviar el StreamType per *live-record* o *live-record-lowlatency*. On diuen que el lowlatency són per aplicacions de xats, haurien de ser més ràpides per l’usuari però comporta més feina per part del server (però que en gravacions no hauria d’afectar).

La idea per tal d’iniciar la gravació al mateix temps és enviar un senyal: **appInstance.broadcastmsg** els clients escolten amb **netconnection.call**

Aleshores per gravar hem de cridar

//canviem el tipus de stream a live

// només video

nsPublish.publish(publishName.text);

nc.call("setStreamType", null, "live-lowlatency");

Quan fan click en el gravar

//abertranb per gravar afegim el publish o append seria afegir

nsPublish.publish(publishName.text, "record");

nc.call("setStreamType", null, "live-record-lowlatency");

# Wowza lib

Fem una applicació per Wowza que el fa és escoltar de les nostres crides

Afegim un mètode:

**public** **void** startRecord(IClient client, RequestFunction function, AMFDataList params) {

//WMSProperties props = client.getProperties();

String room = params.getString(*PARAM1*);

*getLogger*().error("Videochat speakapps - startRecordClient "+room);

*\_appInstance*.broadcastMsg("startRecordClient", **new** AMFDataItem(room));

}

Això el que fa és que enviar un missatge broadcast a tots els clients (es pot passar els ids dels clients, però ho deixem per més endavant) indicant que la room es gravarà.

Per cridar-ho des del Flash fem:

nc.call("startRecord", null, "12345");

# Flash JS

## De Flash a JS

function setupCallbacks()

{

// Notify the container that the SWF is ready to be called.

ExternalInterface.call("setSWFIsReady");

}

## De JS a Flash

En Flash

ExternalInterface.addCallback("startRecordFromJS", doStartRecord);

En JS

function startRecordRequest(){

var flash = swfobject.getObjectById("videochat\_stream\_id");

flash.startRecordFromJS();

}

## De Flash a Wowza

En Flash

var nc:NetConnection = null;

…..

nc.call("startRecord", null, roomIDVar); //cridem al mètode startRecord de la llibreria de Wowza

## De Flash a Wowza

En Flash

\_appInstance.broadcastMsg("startRecordClient", new AMFDataItem(username), new AMFDataItem(room));

# Wowza Eclipse plugin

<http://www.wowza.com/streaming/developers/wowza-ide-software-update>

1. Install Eclipse IDE for Java Developers (version 3.5 SR1 [galileo] or greater)

2. Run Eclipse IDE for Java Developers

3. Select menu item Help: Install New Software...

4. In the Available Software dialog, click the Add... button

5. In the Add Site dialog, enter the following information and click the OK button:  Name: Wowza Location: wowza.com/wowzaide2

6. In the Available Software dialog, check the box next to the Wowza IDE item and click the Next> button

7. In the Install Details dialog, click the Next> button

8. In the Review Licenses dialog, select the Accept... radio option and click the Finish button - See more at: <http://www.wowza.com/streaming/developers/wowza-ide-software-update#sthash.mkFkgPLs.dpuf>

9. Genera un .jar que hay que copier en /usr/local/WowzaStreamingEngine/lib