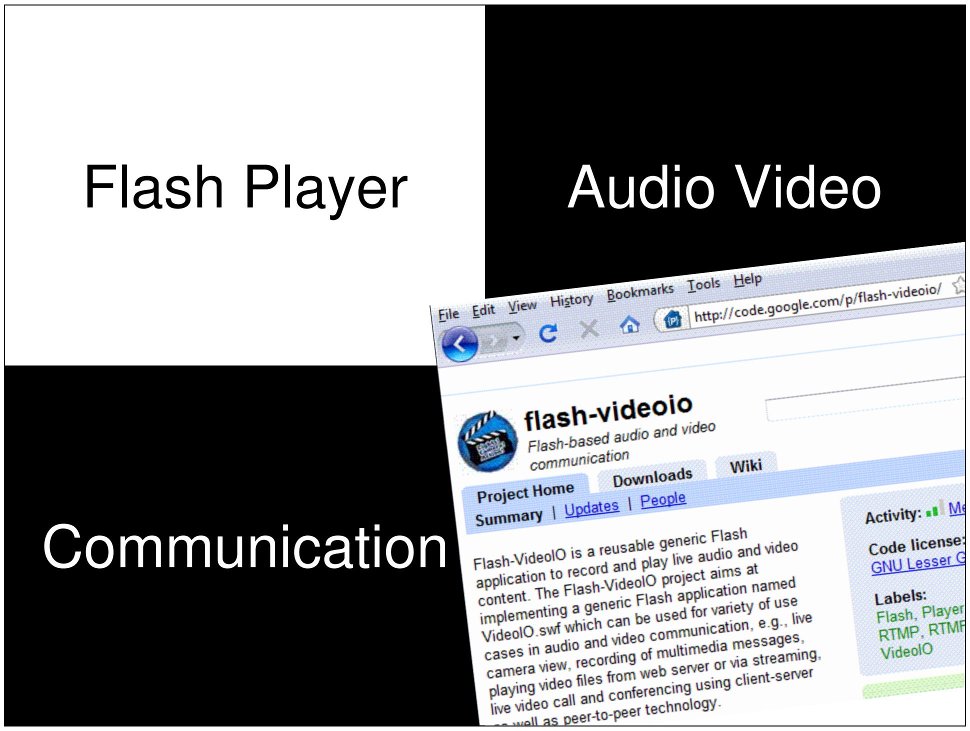


Flash Player

Audio Video

Communication

File Edit View History Bookmarks Tools Help
http://code.google.com/p/flash-videoio/
Project Home
Downloads
Wiki
Summary | Updates | People
Activity:
Code license:
GNU Lesser G
Labels:
Flash, Player
RTMP, RTMF
VideoIO

flash-videoio

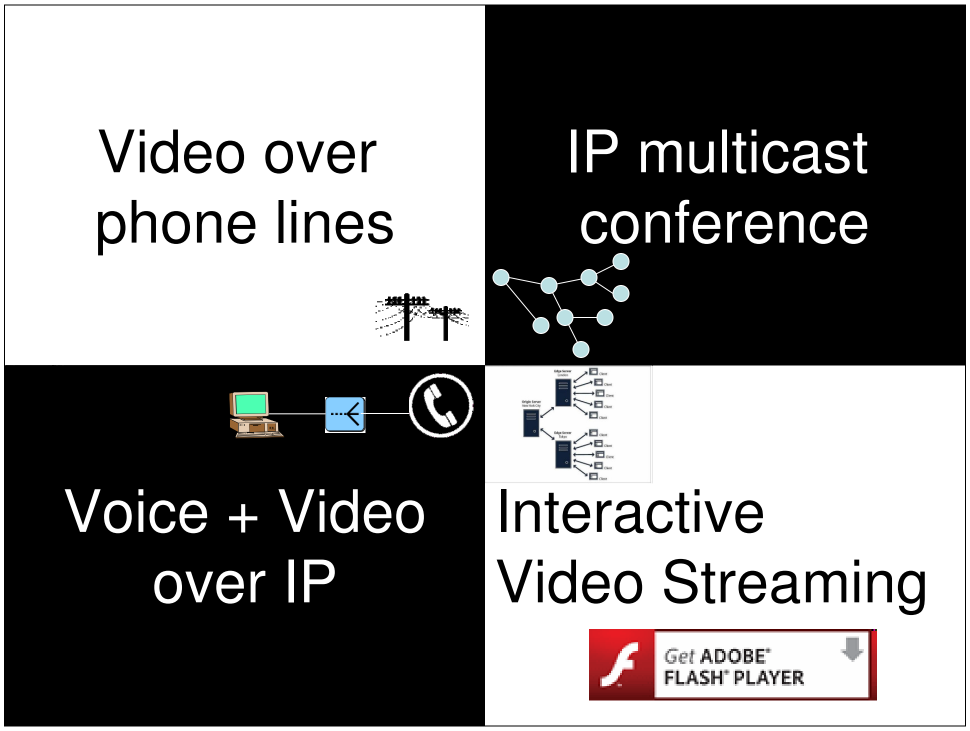
Flash-based audio and video communication

Flash-VideoIO is a reusable generic Flash application to record and play live audio and video content. The Flash-VideoIO project aims at implementing a generic Flash application named VideoIO.swf which can be used for variety of use cases in audio and video communication, e.g., live camera view, recording of multimedia messages, playing video files from web server or via streaming, live video call and conferencing using client-server as well as peer-to-peer technology.

Modern multimedia communication systems have roots in several different technologies: transporting video over phone lines, using multicast on Internet2's Mbone, adding video session to VoIP or instant messaging, or adding interactive mode in existing streaming systems. Adobe's Flash Player has emerged as the most popular web platform for video content and used by almost all web users. More recently, several companies have attempted Flash-based interactive video communication. The high level abstractions offered by Flash Player and Flex framework significantly eases the development of video applications.

This talk will present the differences in video communication technologies, how it is being deployed in practice using the Flash platform, shedding some light on many confusions about what works and what does not, and how can Flash video work with standards such as SIP and HTML 5. This developer focused presentation will walk you through what it takes to build a Flash-based video communication system using example code.

The talk is about Flash-based audio and video communication by Kundan Singh. The software used in this presentation is available from http://code.google.com/p/flash-videoio/



Video over phone lines

IP multicast conference

Voice + Video over IP

Interactive Video Streaming

Get ADOBE® FLASH® PLAYER

Audio video communication is not new. As early as 1960's we had video phones. And today we have dozens of brands of video phone equipments to choose from. Most of these early video phone systems worked on the philosophy of carrying encoded video bits over phone lines, or some connected circuit. For example, systems using H.320 room based video conferencing or H.324 over modems.

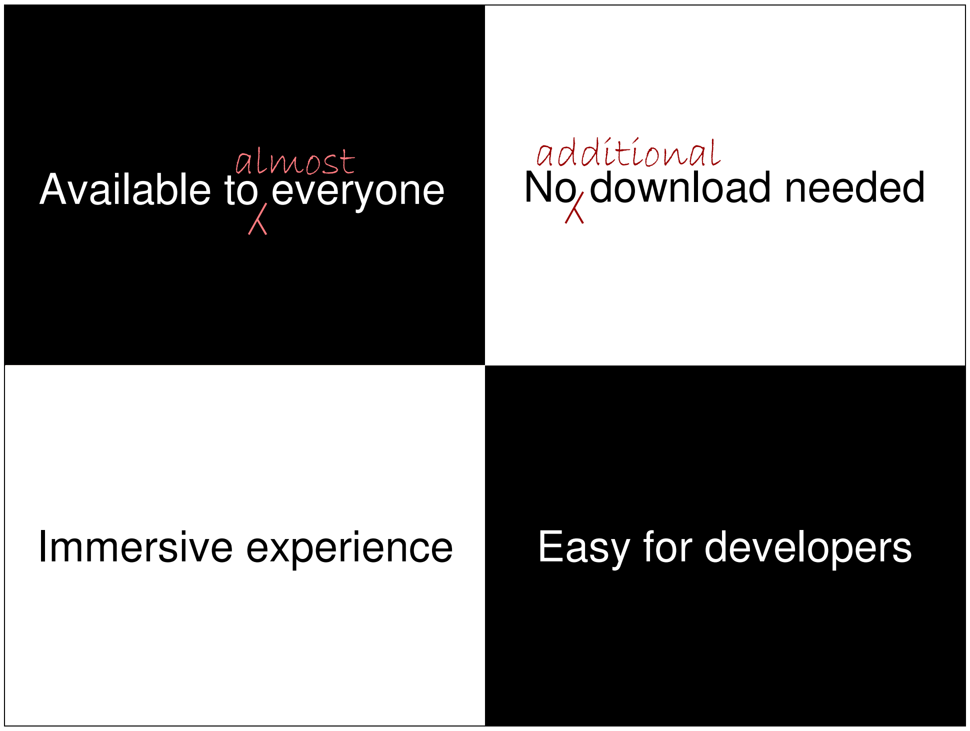
With the popularity of Internet Protocol and various extensions such as IP multicast, people started experimenting with desktop video conferences over multicast test bed, Mbone. For example, robust audio tool (rat) and video conferencing tool (vic), There were other related Internet applications that became popular, e.g., Voice over IP and video streaming. Video became a natural extension for VoIP protocols such as SIP and H.323. Video streaming systems were amended to allow interactivity for real-time video conferences.

A particular application sometimes inherits some advantages as well as problems based on its roots. For example, the H.32x applications following video-over-phone lines philosophy were too complex, and required huge telecom investment to work on them. IP multicast conference embraced the text-based protocols but could not be used on general Internet were multicast is not available. Video streaming applications opened up to web platform but inherited client-server nature of streaming which is not always suitable for interactive communication.

Without going in to further issues among the individual classes of systems, let us focus on the interactive video streaming systems on the web using Adobe' Flash Player.

Why is this approach promising?

Let us review what makes Flash Player a approach promising, and makes it different from other approaches.



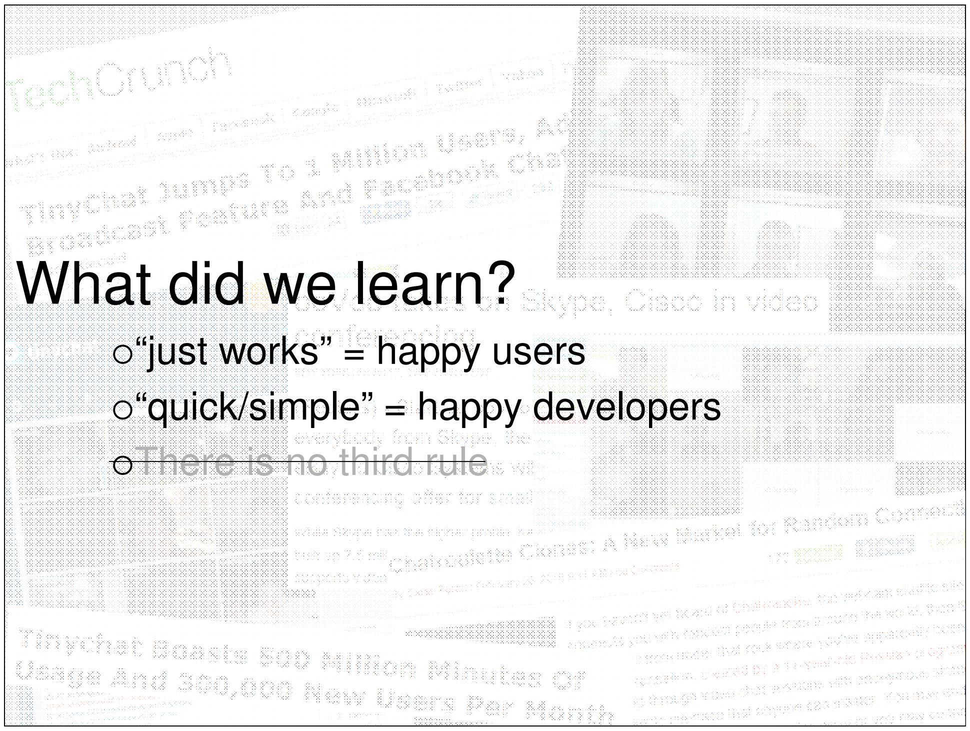
Available to almost everyone	No additional download needed
Immersive experience	Easy for developers

The most important reason is that Flash Player plugin is available to almost everyone with a PC and Internet. It is ubiquitous, more than a specific brand of browser. Anyone with an internet connection and a browser can generally use Flash-based applications.

The second reason is that developers find it very easy to work on Flash platform. Its cross browser platform support is amazing. Unlike a few hundred VoIP companies, there are millions of web developers. The programming language (ActionScript/mxml) used by Flash Player are similar to the web application languages such as JavaScript and HTML. The learning curve is quick, the development tools are awesome, and the community support is excellent!

Thirdly, the end user sees the whole rich internet application experience as embedded and immersive in to what he is doing. For example, when browsing Facebook, you can chat with other friends within the Facebook web page.

Finally, there is no additional download needed to install and configure, but it is just there. This is huge plus for some use cases, e.g., in cyber-café or secure machines, where you do not have the luxury to install and configure Skype, Adium, Yahoo, MSN, etc, you can still do Flash-based video conference.



What did we learn?

- “just works” = happy users
- “quick/simple” = happy developers
- ~~There is no third rule~~

These are the reasons for the popularity of Flash based video communication web sites in recent years. It is too easy and quick for you to get started --both from developers and users point of view.

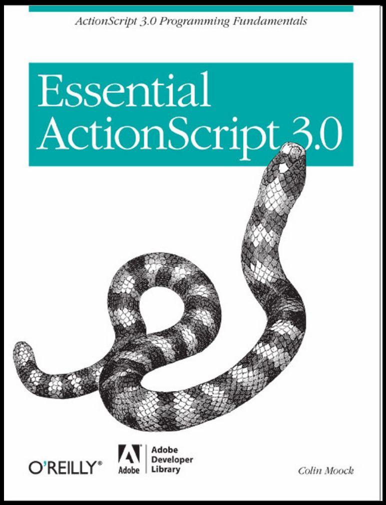
In technology, it is not always the big that wins, but the fast one does. And Flash Player gives the necessary tools to quickly add web video communication to your software product, service or platform.

What we learn from this are two important lessons: if the system “just works” without much installation, configuration, how-tos, then users are happy, which makes them use your system again and again, and helps your business. If the system is quick and simple to build, then developers are happy and motivated to add more functions, learn quickly, and innovate! This moves your business quickly from idea to production, instead of having to make big investments. There is no third rule to success in your video communication business!

How can you build upon this?

So the next question is how can you as a software professional take advantage of this?

Flash-based video communication is essentially built using the similar client-server architecture that you are familiar from the web application development. A client-server Flash application runs in a browser using the Flash Player plugin. It can connect to servers, including your web server, media server or third-party servers via standard transport of TCP.


ActionScript 3.0 Programming Fundamentals
Essential ActionScript 3.0
O'REILLY
Adobe Developer Library
Colin Moock

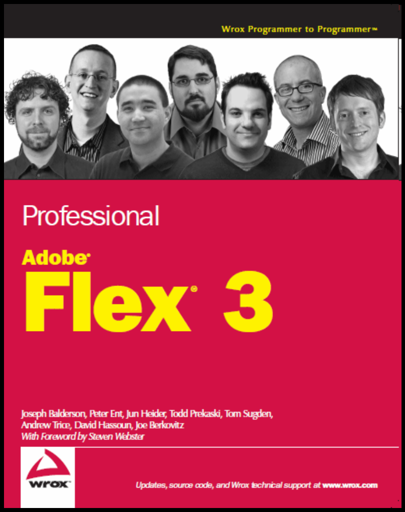
```
var cam:Camera = Camera.getCamera();
var video:Video;
if (cam != null)
{
    video = new Video(320, 240);
    video.attachCamera(cam);
    addChild(video);
}
```

Developers can get started with Actionscript and Flex.

Actionscript is the primary scripting language for developing client-side Flash applications. Actionscript is basically like Javascript but allows more strict type checking and maintainable source code. It is an event-driven object oriented programming language with clean syntax. The Flex compiler converts your scripts to a byte-coded SWF file, which you can embed in your HTML pages. The browser invokes the Flash Player plugin to run the SWF, and hence your flash application.

The ActionScript example in this slide shows how to get the default camera in your PC, and attach it to a Video display object to display the local camera view.

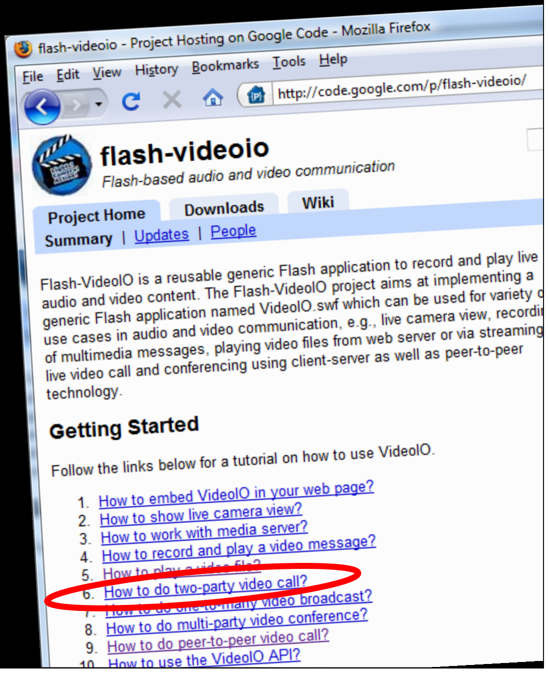
```
<mx:VideoDisplay id="vid"
    width="320" height="240"
    creationComplete="init()">
  <mx:Script>
    [Bindable]
    public var url:String;

    private function init():void
    {
       ...
    }
  </mx:Script>
</mx:VideoDisplay>
```


Wrox Programmer to Programmer™
Professional
Adobe Flex 3
Joseph Balderson, Peter Ent, Jun Heider, Todd Prekaski, Tom Sugden, Andrew Trice, David Hassoun, Joe Berkovitz
With Foreword by Steven Webster
wrox
Updates, source code, and Wrox technical support at www.wrox.com

Flex is an Adobe framework for building user interface applications using markup and script. This is similar to how a web application uses HTML markup and Javascript to render content and implement processing, respectively. In Flex, the MXML markup and ActionScript code work together to build the complete Flash application. The Flex SDK and compiler are freely available from Adobe.

The MXML example in this slide shows a new component derived from the built-in VideoDisplay component, with dimension of 320x240 pixels, and invokes the init() function on creation complete.

JavaScript API


flash-videoio - Project Hosting on Google Code - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://code.google.com/p/flash-videoio/
Project Home
Downloads
Wiki
Summary | Updates | People

flash-videoio
Flash-based audio and video communication

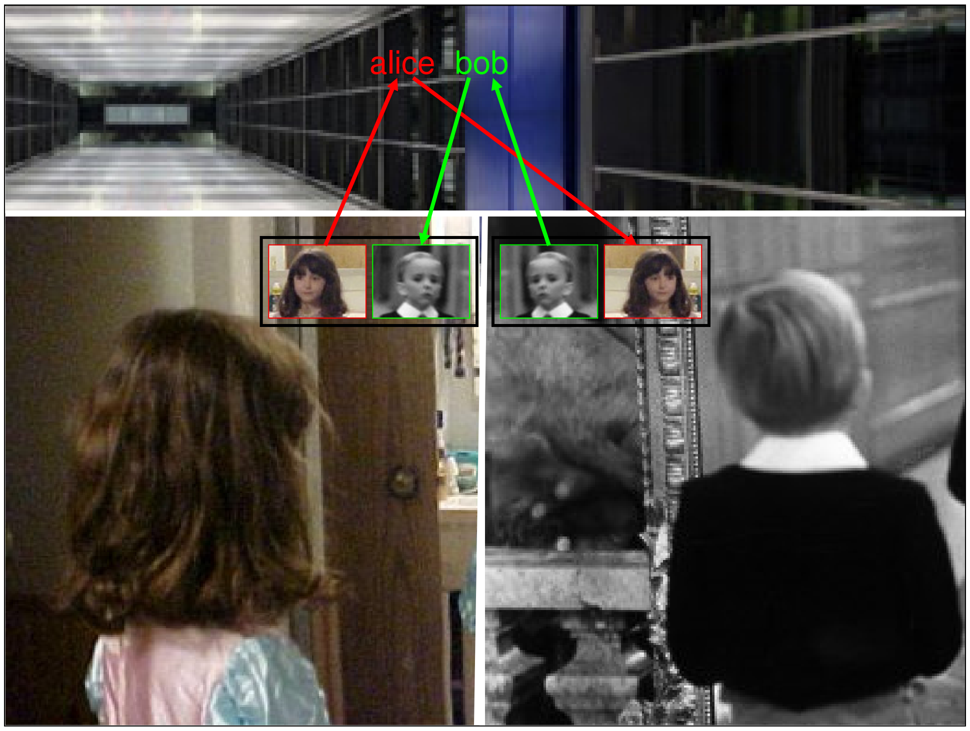
Flash-VideoIO is a reusable generic Flash application to record and play live audio and video content. The Flash-VideoIO project aims at implementing a generic Flash application named VideoIO.swf which can be used for variety o use cases in audio and video communication, e.g., live camera view, recordi of multimedia messages, playing video files from web server or via streaming live video call and conferencing using client-server as well as peer-to-peer technology.

Getting Started

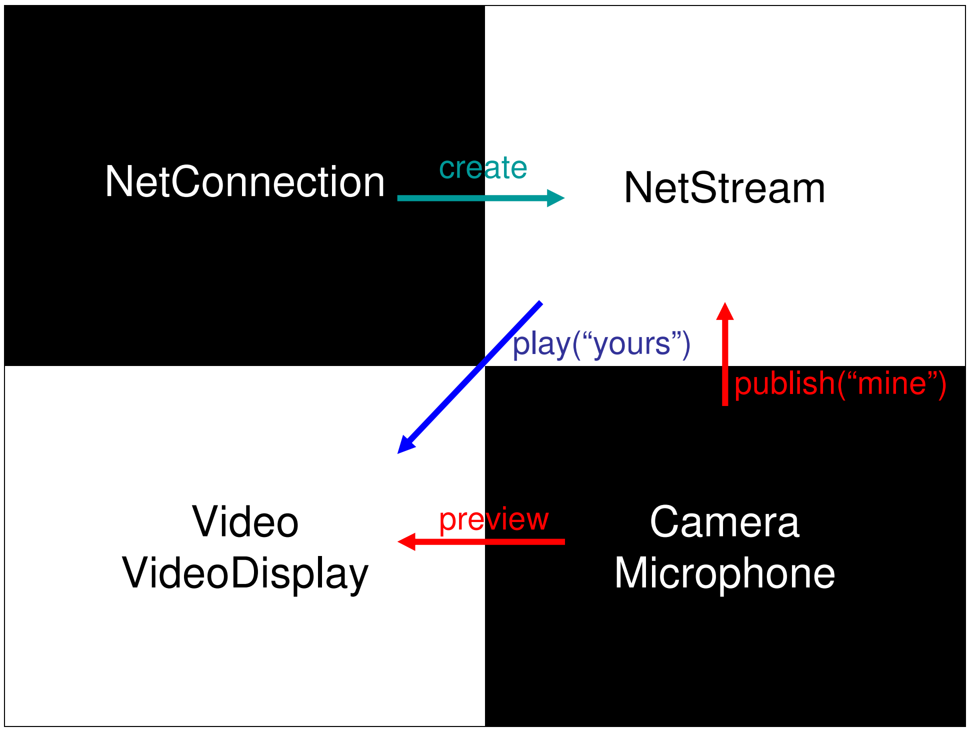
Follow the links below for a tutorial on how to use VideoIO.

1. How to embed VideoIO in your web page?
2. How to show live camera view?
3. How to work with media server?
4. How to record and play a video message?
5. How to play a video file?
6. How to do two-party video call?
7. How to do one-to-many video broadcast?
8. How to do multi-party video conference?
9. How to do peer-to-peer video call?
10. How to use the VideoIO API?

For those who know JavaScript and are reluctant to learn Flex/ActionScript, I have created a project with nice JavaScript API, that allows you to use Flash based audio video communication on your web site. There are examples of how to use this in various use cases. You can visit the project web site at http://code.google.com/p/flash-videoio/ for more information. Let us focus on the two party video call scenario for now.



Suppose Alice and Bob want to do a video call using the Flash media service. The media service provides abstract named streams which Alice can publish and Bob can play, and vice-versa. From the software point of view each side has two Video components, one configured to publish and other to play, to build such as video call application.



Flex ActionScript has four components of classes or abstractions related to video communication application.

The network connection, NetConnection, represents the association between the client Flash application and the media service, which is necessary to create a media stream.

The camera and microphone abstractions provide a platform independent device input. The Flash application does not deal with raw or encoded media stream, but just connects the camera and microphone to either video rendering object or the media stream.

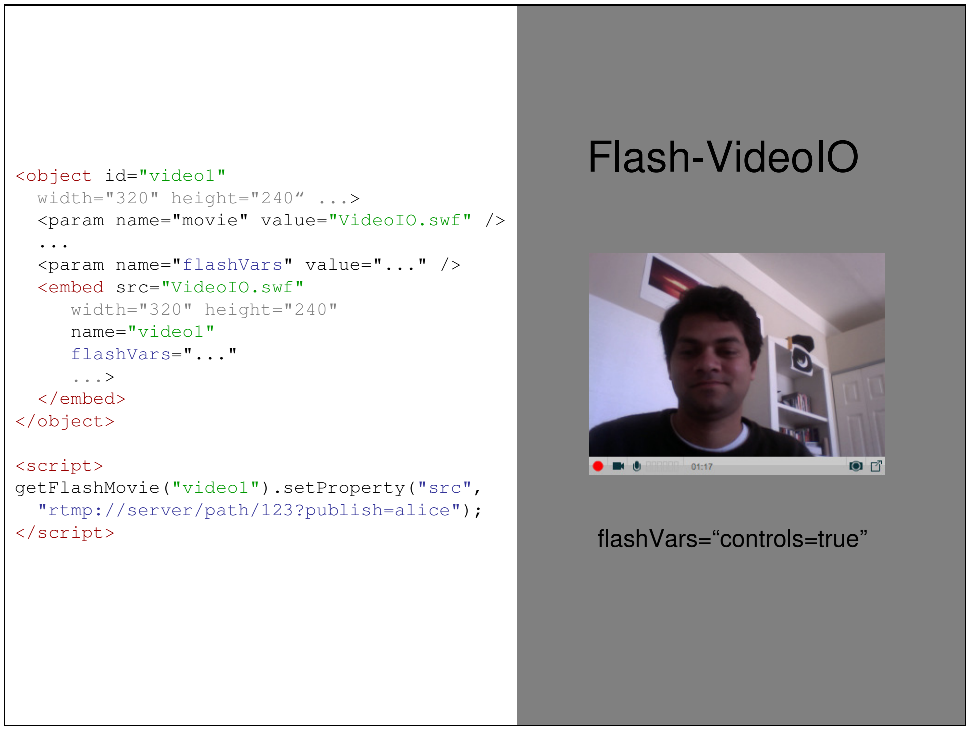
Typically your application will attach the camera to the local video display, and also to the published stream name. And play the stream name of the remote side to the video display.

NetConnection

NetStream

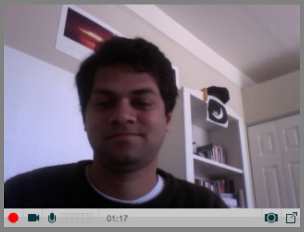
Video
VideoDisplay

Camera
Microphone



Flash-VideoIO

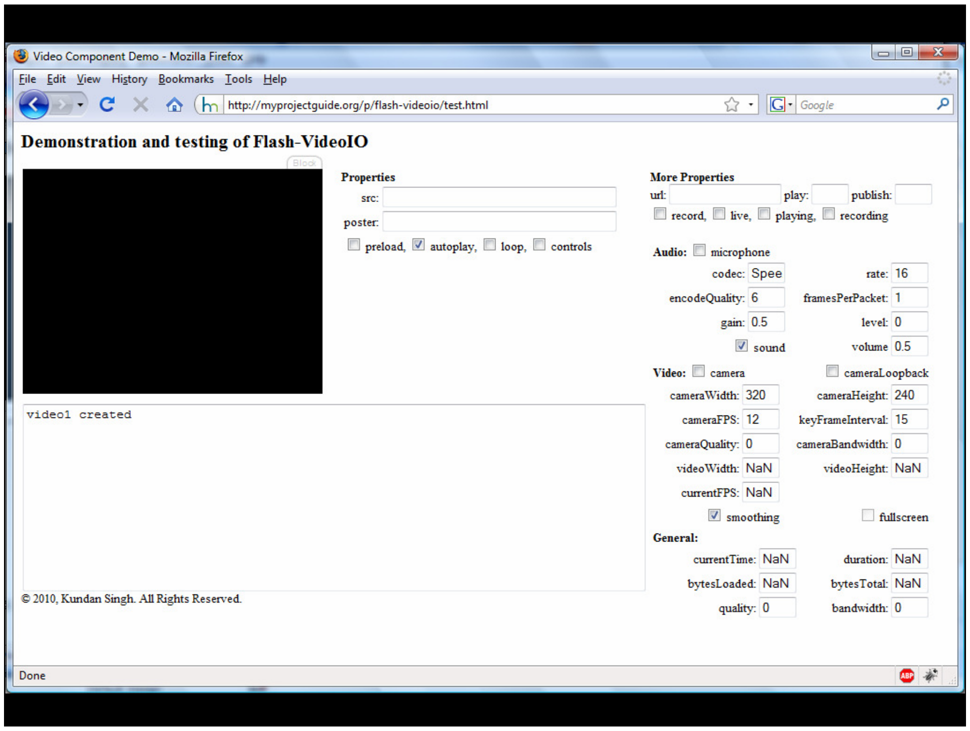
```
<object id="video1"
  width="320" height="240“ ...>
  <param name="movie" value="VideoIO.swf" />
  ...
  <param name="flashVars" value="..." />
  <embed src="VideoIO.swf"
     width="320" height="240"
     name="video1"
     flashVars="..."
     ...>
  </embed>
</object>

<script>
getFlashMovie("video1").setProperty("src",
  "rtmp://server/path/123?publish=alice");
</script>
```



flashVars="controls=true"

The VideoIO project combines all these abstractions into a single easy to use Flash application with extensive JavaScript API.

This example shows how the "src" property can set the VideoIO component to a video publish mode. As mentioned before with two such video components and some form of negotiation for media stream names, you can build a video call application.


Video Component Demo - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://myprojectguide.org/p/flash-videoio/test.html
Google
Done

Demonstration and testing of Flash-VideoIO

Properties
src:
poster:
[ ] preload, [✓] autoplay, [ ] loop, [ ] controls

```
video1 created
```

© 2010, Kundan Singh. All Rights Reserved.

More Properties
url: play: publish:
[ ] record, [ ] live, [ ] playing, [ ] recording

Audio: [ ] microphone

codec:	Spee	rate:	16
encodeQuality:	6	framesPerPacket:	1
gain:	0.5	level:	0
[✓] sound		volume	0.5

Video: [ ] camera [ ] cameraLoopback

cameraWidth:	320	cameraHeight:	240
cameraFPS:	12	keyFrameInterval:	15
cameraQuality:	0	cameraBandwidth:	0
videoWidth:	NaN	videoHeight:	NaN
currentFPS:	NaN		
[✓] smoothing		[ ] fullscreen	

General:

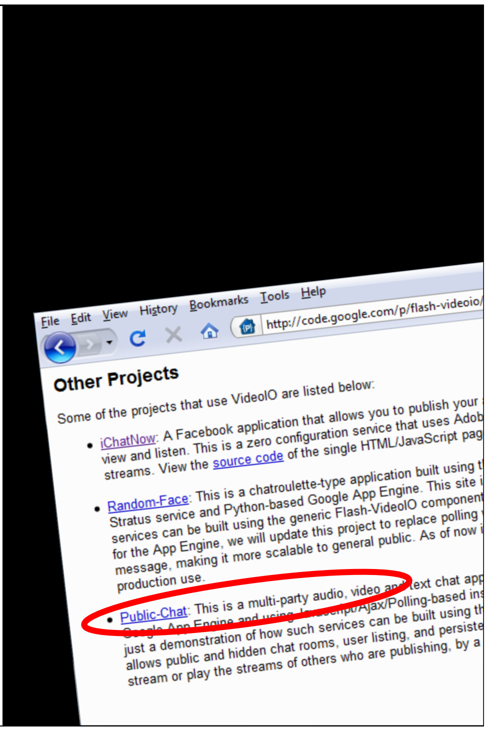
currentTime:	NaN	duration:	NaN
bytesLoaded:	NaN	bytesTotal:	NaN
quality:	0	bandwidth:	0

The VideoIO project also allows developers to explore various properties of the VideoIO component to see how they interact.

Multi-party audio/video conference

http://public-chat.appspot.com/

< 1000 lines of code


File Edit View History Bookmarks Tools Help
http://code.google.com/p/flash-videoio/

Other Projects

Some of the projects that use VideoIO are listed below:

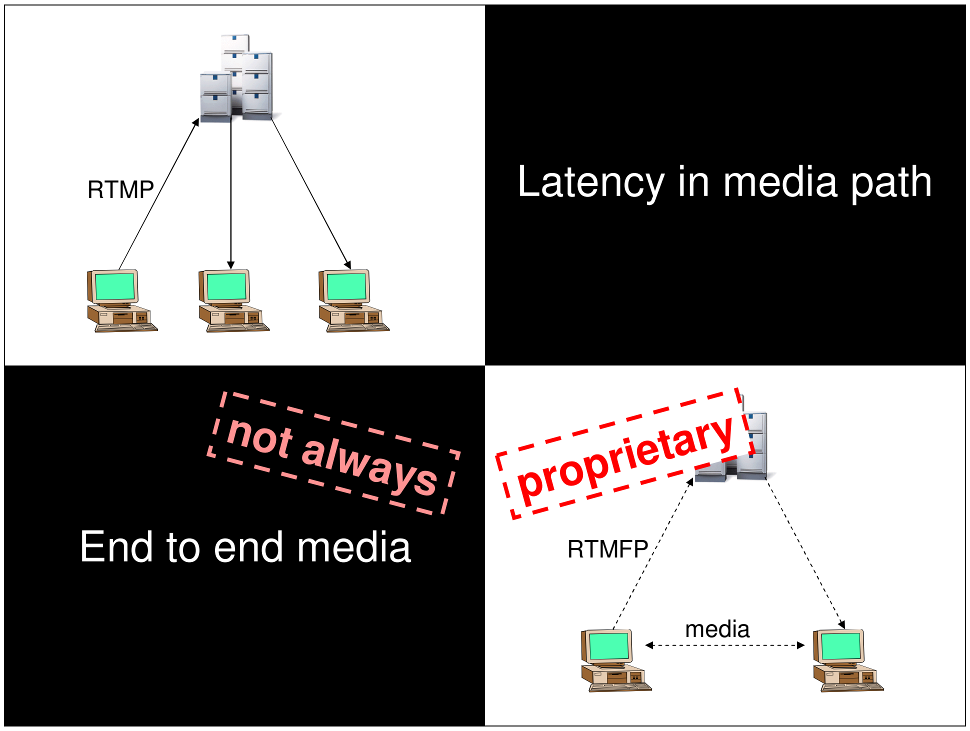
- iChatNow: A Facebook application that allows you to publish your view and listen. This is a zero configuration service that uses Adob streams. View the source code of the single HTML/JavaScript pag
- Random-Face: This is a chatroulette-type application built using t Stratus service and Python-based Google App Engine. This site i services can be built using the generic Flash-VideoIO component for the App Engine, we will update this project to replace polling message, making it more scalable to general public. As of now i production use.
- Public-Chat: This is a multi-party audio, video and text chat app Google App Engine and using Javascript/Ajax/Polling-based ins just a demonstration of how such services can be built using th allows public and hidden chat rooms, user listing, and persiste stream or play the streams of others who are publishing, by a

I have built several sample applications using VideoIO such as chat roulette type random chat system, as well as multi-party video conferencing. This is built on Google App Engine, using standard web application techniques such as AJAX and polling-based IM and presence.

Let me quickly show a demonstration of the public chat application. You can also visit the application at http://public-chat.appspot.com.

“All that glitters is not gold”

Now that we have seen the power and ease of Flash based audio video communication, let us look at some limitations.



This shows the high level client-server architecture for Flash video communication. RTMP or real-time messaging protocol, which runs over TCP, allows you to create streams at the media server, and publish or play media over it. This works well for one-to-many streaming application where provider can install multiple servers and do load balancing. It also works well for NAT and firewall traversal because of the client-server nature of the TCP connection, and all media and signaling path is same.

But the latency in media path can become huge, and is not desirable for interactive video calls. Luckily Adobe has worked on another protocol, RTMFP, which enables end-to-end media and significantly reduces the latency in the media path.

It works, but has two problems. First it is proprietary, and second, it cannot always do end-to-end media because of certain firewalls and NAT restrictions. Because it is proprietary, third-party cannot build scalable media relays, similar to super-node architecture of Skype. Thus, for a robust service, the provider needs to invest in service infrastructure or fall-back to client-server RTMP.

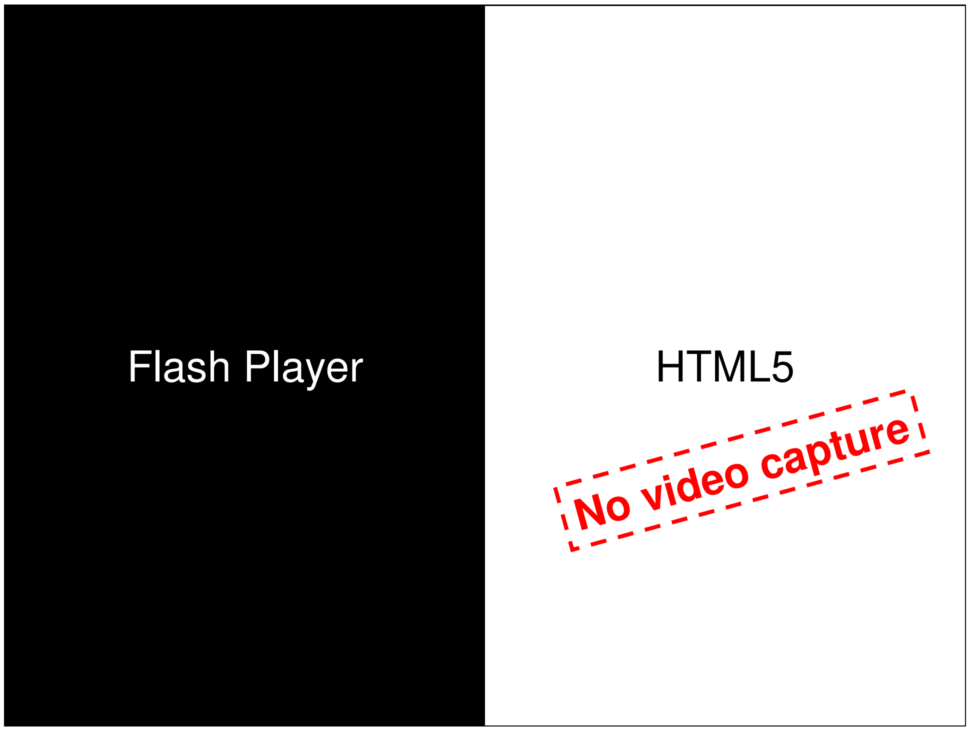
missing
Echo cancellation

High quality encoder
not integrated

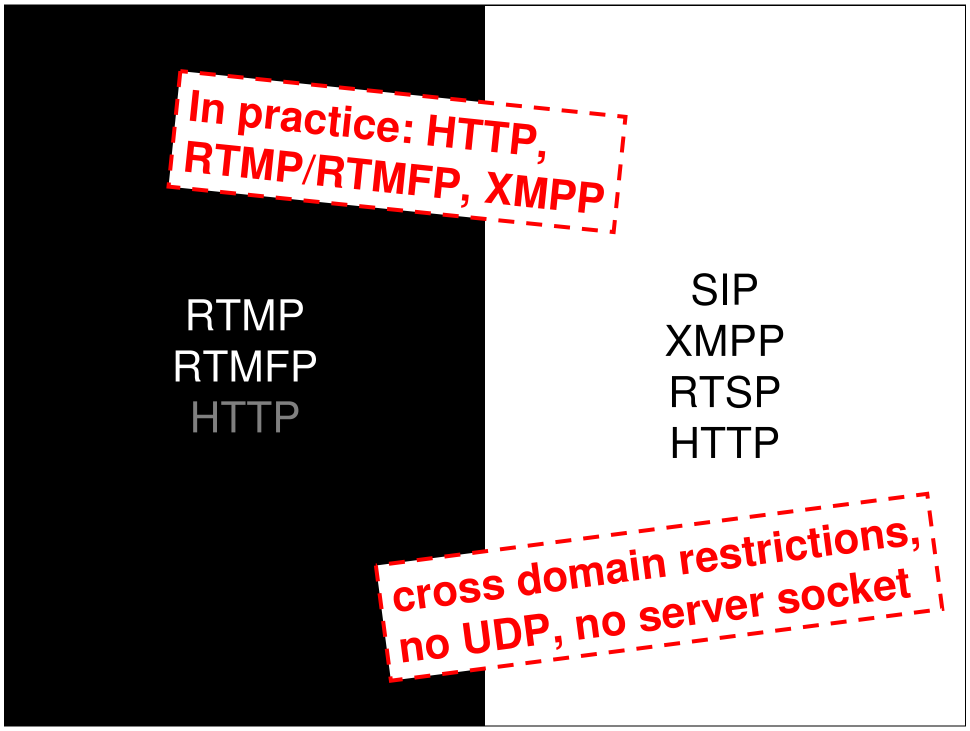
Another challenging issue with Flash Player is that it does not have good echo cancellation. In practice, web video sites recommend using a head-set or implement push-to-talk. Some have implemented their own plugin or download which performs audio processing to improve the quality.

In terms of video quality, Flash Player still uses an old and patented video encoder, Sorenson, for camera capture. While recent version of Flash Player incorporated high quality H.264 for playback, it is quite unlikely that they will add the H.264 encoder in Flash Player. Thus, based on the current technology, while you can do quick and easy web video conferencing, a high-quality video conferencing is not possible without additional kludges.

There is an alternative technology, Adobe Integrated Runtime or AIR, intended for desktop applications instead of within a browser, that has some advantages. For example, it allows creating UDP sockets and hence implementing UDP-based low latency application protocols such as SIP and RTP. Secondly, it allows access to raw microphone data in the application, and hence application can do any signal processing if needed. However, since it is not a browser-based platform, you miss out on certain advantages of Flash Player.



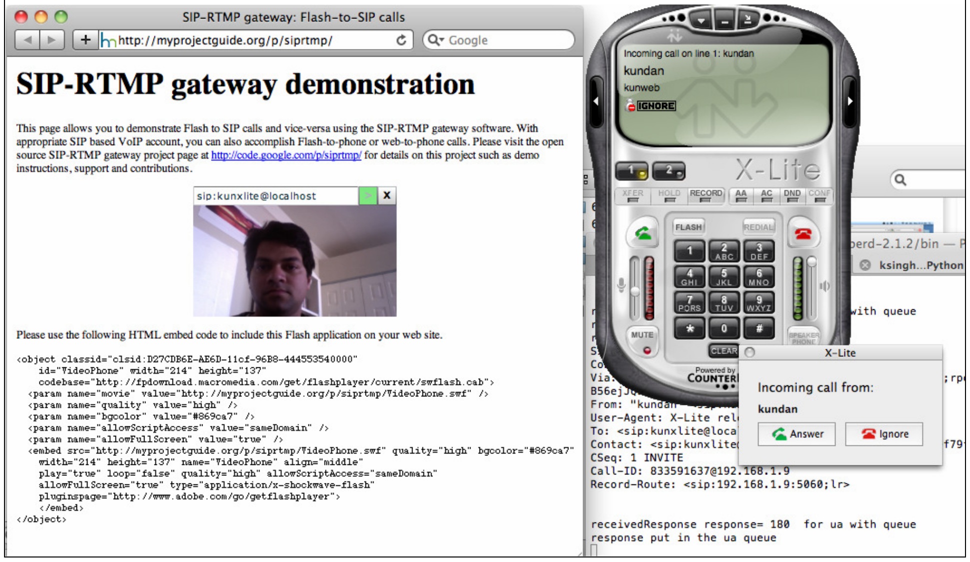
HTML5 has gained momentum recently and has introduced new audio and video tags. This is only for playback of videos and provides alternative encoding to reach out to wide variety of browsers. It does not have audio and video capturing tags, hence cannot yet be used for video communication. Due to this and other reasons such as digital rights management (DRM), Flash Player will continue dominate the web video communication technology for now.


In practice: HTTP, RTMP/RTMFP, XMPP
RTMP
RTMFP
HTTP
SIP
XMPP
RTSP
HTTP
cross domain restrictions, no UDP, no server socket

In terms of protocols, Flash Player supports RTMP, RTMFP and HTTP for media streaming in playback mode but only RTMP and RTMFP in publish mode. On the other hand, rest of the world relies on standards such as SIP, XMPP and RTSP for signaling and control of real-time sessions. In practice, people have used a combination of HTTP, XMPP and RTMP for providing a comprehensive web based messaging, presence and video communication services.

Because of cross-domain restrictions, unavailable UDP socket, and missing server side socket in Flash Player, you cannot implement a full SIP stack in a Flash application. There are several gateway service implementations that translate between RTMP and SIP.

siprtmp.py in action



SIP-RTMP gateway: Flash-to-SIP calls

http://myprojectguide.org/p/siprtmp/

SIP-RTMP gateway demonstration

This page allows you to demonstrate Flash to SIP calls and vice-versa using the SIP-RTMP gateway software. With appropriate SIP based VoIP account, you can also accomplish Flash-to-phone or web-to-phone calls. Please visit the open source SIP-RTMP gateway project page at http://code.google.com/p/siprtmp/ for details on this project such as demo instructions, support and contributions.

sip:kunxlite@localhost

Please use the following HTML embed code to include this Flash application on your web site.

```
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
    id="VideoPhone" width="214" height="137"
    codebase="http://fpdownload.macromedia.com/get/flashplayer/current/swflash.cab">
  <param name="movie" value="http://myprojectguide.org/p/siprtmp/VideoPhone.swf" />
  <param name="quality" value="high" />
  <param name="bgcolor" value="#869ca7" />
  <param name="allowScriptAccess" value="sameDomain" />
  <param name="allowFullScreen" value="true" />
  <embed src="http://myprojectguide.org/p/siprtmp/VideoPhone.swf" quality="high" bgcolor="#869ca7"
    width="214" height="137" name="VideoPhone" align="middle"
    play="true" loop="false" quality="high" allowScriptAccess="sameDomain"
    allowFullScreen="true" type="application/x-shockwave-flash"
    pluginspage="http://www.adobe.com/go/getflashplayer">
    </embed>
</object>
```

Incoming call on line 1: kundan
kundan
kunweb

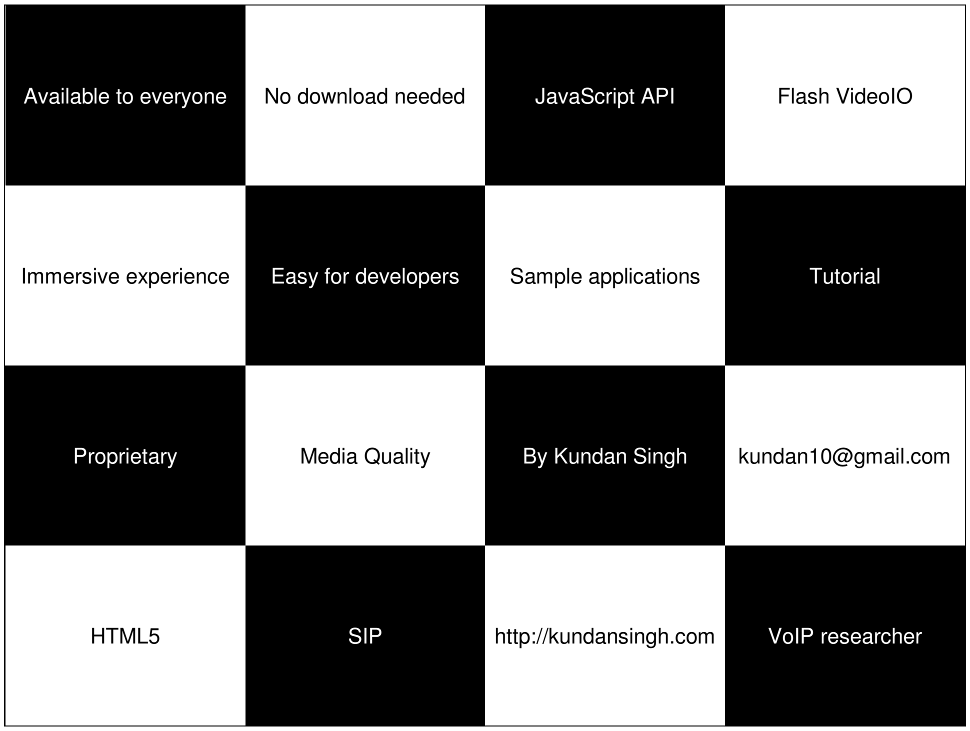
Incoming call from:
kundan

```
CSeq: 1 INVITE
Call-ID: 833591637@192.168.1.9
Record-Route: <sip:192.168.1.9:5060;lr>

receivedResponse response= 180  for ua with queue
response put in the ua queue
```

For example, siprtmp is one such project that I have been involved with. The gateway allows a Flash application embedded in a web page to register to SIP server and make or receive SIP calls. The translation mechanism and source code of the gateway is at
http://code.google.com/p/rtmplite/source/browse/trunk/siprtmp.py

While the gateway can allow you to translate signaling and voice, the video stream is more involved because the proprietary/patented Sorenson codec used by Flash Player. You can use the gateway to implement several web telephony use cases such as PC-to-phone dialing, allowing a phone user to dial into a web conference, or integrating your web application with existing VoIP infrastructure.

I have started a project on a downloadable application that will allow the gateway to run locally on your machine along with a P2P-SIP adaptor.



Available to everyone	No download needed	JavaScript API	Flash VideoIO
Immersive experience	Easy for developers	Sample applications	Tutorial
Proprietary	Media Quality	By Kundan Singh	kundan10@gmail.com
HTML5	SIP	http://kundansingh.com	VoIP researcher

In summary, we have seen the benefits of using Flash Player for audio and video communication on the web. I have described my Flash-VideoIO project that provides an easy JavaScript API and several sample applications. We discussed some of the problems associated with the Flash-based audio and video application, and how it can interoperate with SIP.

If you have questions or comments, you can reach me at my email.