

## NBC Universal Hackathon - Miami 2018



This weekend I went and spent some time hacking some code at the "NBC Universal Hackathon" and trying it out new ideas, meeting new friends, and learning a ton on many technological aspects and innovating. The particular problem that we decided to solve was the irrelevance aspects of current TV and how more interactive could it be with current technologies. The way to solve it thru a collaborative experience where users can interact with their phones and cameras with the video shown on screen.

The team was composed by: Satya, Paul Valdez, Juan Gus, Myself, and Chris.

What we did what simple, we could create a website that could have a canvas that could be treated with effects, add the TV/Video feeds into it and that distribute the content using a platform like "Cloud to Cable TV" to cable operators or OTT/IPTV systems.



Cloud to Cable TV

The solution required a few items to be setup and configured:

- [RTMP Server](#) or WebRTC Setup to receive video feeds from Smartphones or your laptop,

- [FFMPEG](#) to encode, compress, and publish video/audio feeds
- [Mobile App with RTMP Client](#) or WebRTC Client or laptop. We tried several but this one worked out ok.
- A web application in Python to map each feed and position it on top of the TV Channel video source (assuming an M3U8 feed or a movie in MP4)

With this in place, it is a matter compiling CRTMP, FFMPEG, and we tried other components as Deep Learning such as the "[Deep Fakes](#)" project. The idea that we had was to replace one of the actors image, as well as superimposed our live feeds into the video.

Issues:

- The safari browser doesn't allow you to play content with autoplay features, meaning that the user MUST initiate a playback. If SAFARI sees or detects that onLoad the content autoplays this fails.
- There are issues with SAFARI and dynamically loading the content and **video.oncanplaythrough()** is required to be added to the javascript.

The live feeds had a delay of about 30-40seconds, as it had to:

- Convert and push from mobile phone to RTMP Server,
- Grab RTMP Stream and send it as an m3u8 encoded file to the website.

The standard CRTMP Screen would look like and connections from Gus and Pablo successfully took place:

```
+-----+
-----+ |
          Services| +---+-----+-----+-----+
-----+-----+-----+-----+ | c |      ip      | port|  protocol
stack name  |      application name  | +---+-----+-----+-----+
-----+-----+-----+-----+ |tcp|                0.0.0.0.
```

```
0| 1112| inboundJsonCli| admin| +---+---+
+-----+-----+-----+-----+-----+-----+
|tcp| 0.0.0.0| 1935| inboundRtmp| ap
pselector| +---+-----+-----+-----+-----+
+-----+ |tcp| 0.0.0.0| 8081| inboundRt
mps| appselector| +---+-----+-----+-----+
+-----+ |tcp| 0.0.0.0| 8080|
inboundRtmp| appselector| +---+-----+-----+
+-----+ |tcp|
0.0.0.0| 6666| inboundLiveFlv| flvplayback|
+---+-----+-----+-----+-----+
+-----+ |tcp| 0.0.0.0| 9999| inboundTcpTs|
flvplayback| +---+-----+-----+-----+
+-----+ |tcp| 0.0.0.0| 5544|
inboundRtsp| flvplayback| +---+-----+-----+
+-----+ |tcp| 0.0.0.
0| 6665| inboundLiveFlv| proxypublish| +---+---+
+-----+-----+-----+-----+-----+
|tcp| 0.0.0.0| 8989| httpEchoProtocol| samp
lefactory| +---+-----+-----+-----+
+-----+ |tcp| 0.0.0.0| 8988| echoProto
col| samplefactory| +---+-----+-----+
+-----+ |tcp| 0.0.0.0| 1111|
inboundHttpXmlVariant| vptests| +---+-----+
+-----+-----+-----+-----+-----+-----+
```

We were trying to use WebRTC but we had many issues with latency and delays.

The FFMPEG command required and that was used for the demo was:

```
ffmpeg -re -i rtmp://96.71.39.58/live/pablo -c:v libx264 -c:a aac -ac 1 -strict -2 -crf 18 -profile:v baseline -maxrate 200k -bufsize 1835k -pix_fmt yuv420p -flags -global_header -hls_time 3 -hls_list_size 4 -hls_wrap 10 -start_number 1 /var/www/html/live/pablo.m3u8 ffmpeg -re -i rtmp://96.71.39.58/live/gus -c:v libx264 -c:a aac -ac 1 -strict -2 -crf 18 -profile:v baseline -maxrate 200k -bufsize 1835k -pix_fmt yuv420p -flags -global_header -hls_time 3 -hls_list_size 4 -hls_wrap 10 -start_number 1 /var/www/html/live/gus.m3u8
```

The Mobile App published an RTMP Stream to the server under /live/pablo and /live/gus. The demo video on what it will look like:

<https://vimeo.com/299048743>Screen capture in Vimeo using Safari

For screen capturing in a Mac with FFMPEG with 3 screens, list your devices and capturing to avoid any MOOV issues and useless MOV/MP4 files.

```
ffmpeg -f avfoundation -list_devices true -i "" ffmpeg -f avfoundati  
on -i "3:0" -t 120 -pix_fmt yuv420p -c:v libx264 -c:a libmp3lame -r 25  
teleport.mov
```

The presentation we made to the judges at the "NBC Universal Hackathon" can be found here:

<https://docs.google.com/presentation/d/1sKAvc-Y-KHu2qclulH2Fp-8yWvTslq6bLaocyEgtfQ/edit?usp=sharing>

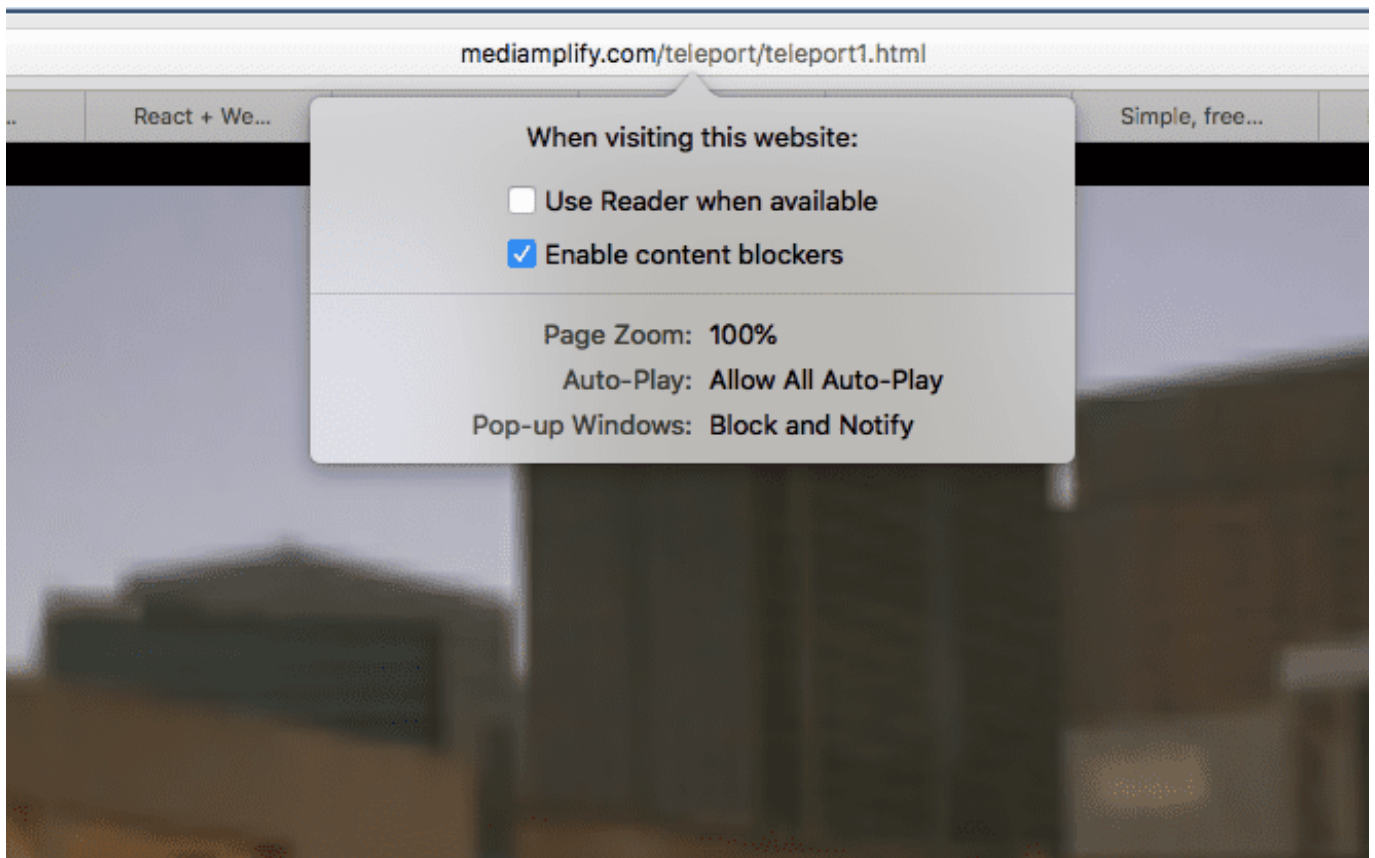
The source code consists on an HTML site using DOM objects, video source, and a canvas. As shown, the video is hidden it is native format in ways that you can use canvas drawing to copy the video from the "src" in m3u8, MOV, MP4 or whatever format your browser can handle to the canvas. The canvas is then the placeholder for all the overlays and divs. The idea with the canvas is that messages can then be typed and exchanged between users, as a WhatsApp application or any other chat application that uses the canvas.

```
var canvas = document.getElementById("c"); var context = canvas.getCon
```

```
text("2d"); window.onload = function() { // document.getElementById(
"fb-profile").style.display = "none"; var canvas = document.get
ElementById("c"); var context = canvas.getContext("2d"); // gr
ab the video element // var video = document.getElementById("v");
// drawVideo(context, video, canvas.width, canvas.height);
// calls drawVideo() function, passing all the objects } var splay
er = {}; function showIt(id, url, hideOrNot) { console.log(id+" "+
url+ " setting it to " +hideOrNot); splayer["v_"+id] = document.ge
tElementById("v_"+id); document.getElementById(id).style.display = h
ideOrNot; if (document.getElementById(id).style.display == "none" )
{ document.getElementById(id).style.display = "block"; var
vId = "vsrc_"+id; console.log("playing "+vId + " "+url); d
ocument.getElementById(vId).src = url; if (splayer["v_"+id].pause
d) { console.log("Video paused.... "); splayer["v_"+i
d].load(); splayer["v_"+id].oncanplaythrough = function() {
splayer["v_"+id].play(); }; } else { co
nsole.log("Video is playing already..."); } } else { cons
ole.log(" Stopping .... v_"+id); splayer["v_"+id].pause(); d
ocument.getElementById(id).style.display="none"; } } var player =
document.getElementById("v"); function ChangeHarry(){ console.lo
g("Playing Harry Potter.... "); document.getElementById("vsrc_main
").src = "http://s3.us-east-2.amazonaws.com/teleportme/videos/HarryPot
terClip.mp4"; player.load(); player.play(); drawVideo(cont
ext, player, canvas.width, canvas.height); } function ChangeQueen(){
console.log("Playing Queen of the South ... "); player.pause();
document.getElementById("vsrc_main").src="http://96.71.39.58/queen0.mp
4"; player.load(); player.play(); // drawVideo(context, player,
canvas.width, canvas.height); } setTimeout(function() { showIt
("first", "https://mediamplify.com/teleport/iwantharry.mp4", "none");
setTimeout(ChangeHarry, 6000); }, 2000 ); setTimeout(fu
nction() { showIt ("first", "https://mediamplify.com/teleport/i
wantharry.mp4", "block"); }, 8000 ); setTimeout(showIt, 5000, "sec
ond", "http://96.71.39.58/live/pablo.m3u8", "none"); setTimeout(showIt
, 6000, "third", "http://96.71.39.58/live/gus.m3u8", "none"); console
.log("Starting changing stuff"); setTimeout(function() {
console.log("Preeping to switch to Queen of the South" );
showIt ("first", "https://mediamplify.com/teleport/iwantqueen.mp4",
"none"); }, 13000); setTimeout(showIt, 15000, "third",
"http://96.71.39.58/live/pablo.m3u8", "none"); setTimeout(showIt, 150
10, "second", "http://96.71.39.58/live/gus.m3u8" , "none"); // setT
imeout(showIt, 20000, "third", "http://96.71.39.58/live/gus.m3u8", "no
ne"); setTimeout(function() { console.log("Queen of the
South"); ChangeQueen();
showIt("first", "", "block"); }, 19000); function fbP
rofile() { var x = document.getElementById("fb-profile"); if (
x.style.display === "none") { x.style.display = "block"; }
```

```
else { x.style.display = "none"; } } function drawVideo(
context, video, width, height) { context.drawImage(video,
0, 0, width, height); // draws current video frame to canvas v
ar delay = 100; // milliseconds delay for slowing framerate setTime
out(drawVideo, delay, context, video, width, height); // recursively c
alls drawVideo() again after delay }
```

For a functional demo, 1st allow the site to play video in autoplay:



Update your settings in SAFARI

[Teleport ME - Demo on Mediampify.com](http://mediampify.com/teleport/teleport1.html)

We didn't win the "NBC Universal Hackathon" but had a ton of fun doing it!. We failed in the presentation, it was only 3 minutes and our presenter disappeared in the last minute, and Gus improvised and didn't use all the time provided by the judges. We knew we were done when no questions were asked. .... Anyways!!! You cannot always win.

---