

A TABLET SCREEN CAST RECEIVER FOR CLASSROOM WITH LOW END ANDROID DEVICES

Moh. Hasbi Assidiqi¹⁾, Aliv Faizal Muhammad²⁾, and Zarkasyi Kuncoro Aji³⁾

^{1,2,3)}Department of Multimedia Creative

Electronic Engineering Polytechnic Institute of Surabaya

Surabaya, Indonesia

e-mail: hasbi@pens.ac.id¹⁾, aliv@pens.ac.id²⁾, ajiekuncoro297@gmail.com³⁾

ABSTRAK

Salah satu aktivitas menggunakan tablet adalah presentasi. Saat ini, kebanyakan aktivitas presentasi dilakukan menggunakan adapter VGA untuk bisa tersambung dengan LCD Proyektor. Konfigurasi ini memungkinkan presentasi berbasis kabel. Dan ini adalah hal yang menyulitkan penggunaannya untuk perangkat tablet yang memiliki sifat mobilitas tinggi. Beruntung, sudah ada banyak vendor yang menyediakan sistem presentasi yang bersahabat dengan membuatnya menjadi nirkabel. Tapi sistem tersebut hanya mendukung perangkat tablet high end. Pada makalah ini, kami mengajukan sebuah penerima tablet screen cast untuk perangkat tablet android low end. Yang memiliki potensi untuk diimplementasikan di kelas. Dari eksperimen, kami memperoleh hasil 9 FPS dengan delay sebesar 2 detik.

Kata Kunci: duplikasi tampilan, eksperimen, jaringan nirkabel, low end android, penerima layar nirkabel.

ABSTRACT

One of the activities using tablet device is a presentation. Currently, most of the presentation activity is supported by the VGA adapter to connect to the LCD Projector. This setup supports a presentation in a wired way. And it is cumbersome to use it for tablet device which has properties of high mobility. Fortunately, there are many vendors that support friendly presentation system for a tablet by making it wireless. But they only support high-end tablet device. In this paper, we present a tablet screencast receiver for a low-end android device. It has the potential to be implemented in the classroom. From the experiment, we achieved a result of 9 FPS with a delay of 2 seconds.

Keywords: display mirroring, experiment, low end android, wireless display receiver, wireless network.

I. INTRODUCTION

BASED on the research done by Thomas et al[1], Tablets are effective for presenting slides, play audio/video, show web pages, and draw/sketch. Most of presentation activity is done by using a VGA adapter or HDMI adapter to connect to the LCD Projector. The most convenient way to send material to a projection system with a tablet is wirelessly. This type of transfer is supported by most modern television.

Currently, many vendors ([1], [2], [3]) provide a means of wireless projector. Most of them is used for entertainment purpose to be used in the living room. Most of them are very vendor specific. Currently, no standard agreed on a wireless projector system using tablet, even though one of them is working on becoming standard [2]. Looks like it's going to be not near in future on the agreement because each vendor uses the system as one of their main advantage over other competitors. The system mainly optimized for high end devices such as Google® Chrome Cast™ [4], which only support a number of tablet devices. Most of the supported device is a high end tablet. This kind of tablet has a low possibility to be implemented in the classroom because of its price.

Most of them also required several pre-requisites that not always available in the classroom, such as an active internet connection. For example, to get the Google Chrome Cast to work, we need an active internet connection. Currently it doesn't support proxy server, which makes it difficult to set up in the classroom with proxy based internet connectivity.

In this research, we present a wireless presentation system which could be used in a classroom with low end android and without internet connection. A proof of concept has been developed which allow user with a low end

Android device to cast their tablet screen to LCD projector using a PC as middleware for receiving screen cast and then project it to Projector Screen (Fig. 1).

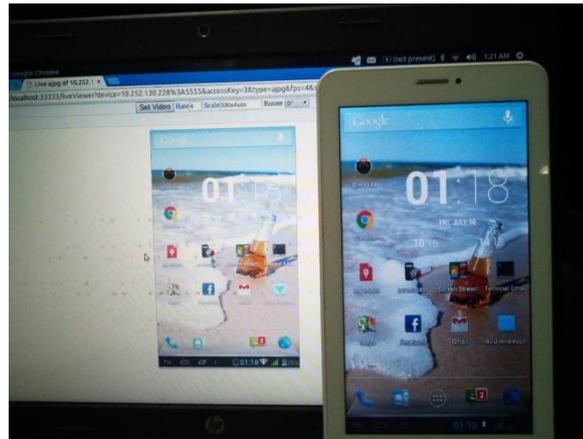


Fig. 1. The proposed system result

The main feature of the proposed system is the capability to be used for low end device with Android version lower than 4.4. The proposed system doesn't need internet connection so that it can be installed anywhere as long as there is Wi-Fi. Because it doesn't need internet connection, it is suitable to be used in classrooms where the internet connection is not always available. For example, in Indonesia, not every school has an internet connection.

The rest of this paper is organized as follows, section 2 describes the architecture and technical realization, in section 3 we compare the solution with similar solutions. Lastly, in section 4 we discuss about possible future work.

II. ARCHITECTURE AND TECHNICAL REALIZATION

To use the wireless presentation system, we assumed a classroom with particular setups as shown in Fig. 2. The setup consists of a classroom's PC connected to the LCD Projector. The tablet and the PC are on the same Wi-Fi network. The tablet acts as a data source. The proposed solution is designed as stand alone application to be installed on the PC.

In high level perspective, the big picture of the system is shown in the context diagram in Fig. 3. The system works as a desktop application. Then, using the application, the teacher can pick the particular tablet as source. And then periodically, the system will capture the screen of the selected tablet device. Then the system will allow the user to view the captured screen.

The proposed system consists of two containers as shown in container diagram in Fig. 4. The first container is a web application that serves as interface for user to the system. This container allows user to select registered tablet and view the captured tablet's screen. The second container is a background process or headless NodeJS application that periodically captures tablet's screen to file. The first container will instantiate a second container each time user activates registered tablet device.

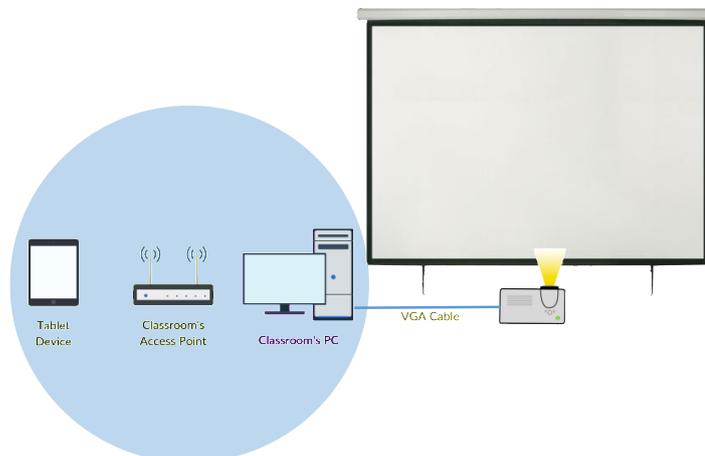


Fig. 2. Hardware Setup Assumption of Wireless Presentation System

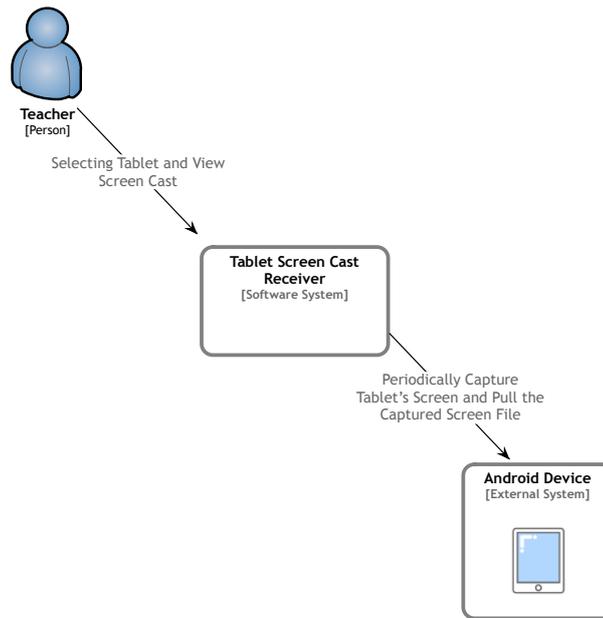


Fig. 3. Context diagram of the proposed tablet screen cast receiver

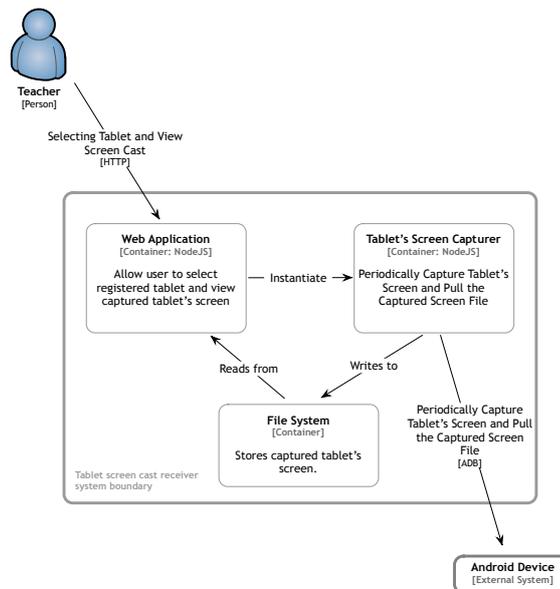


Fig. 4. Container diagram of the proposed tablet screen cast receiver

There are two main steps in the application, the first step is to register the tablet device to the screen caster application. And the second step is the screen casting step. The registration step is only done once to make the tablet device available to the application. And the screen casting step is done every time the teacher activates screen casting functionality in the application.

After the teacher select the registered device, the screen casting process begins. The screen casting is done using several steps that are done repeatedly as shown in Fig. 5. Those steps are capture step, collect step, and serve step. All steps are done repeatedly according to the specified rate setting or FPS (frames per second).

In general, the application will connect to the tablet then capture the screen to a file. After that, it will collect the captured file to be served through a web page. This process, the overall process based on where the operation happens is shown in Fig. 6 while the process of capturing screen happens in the tablet.

A. Registration Step

The registration step is done once after installation of the system. This registration step is needed to allow PC to access the tablet through ADB protocol. The registration step is straightforward. First the PC needs to be installed with the Android SDK. Then, using the ADB command line tool, we scan the Android device. When the android device is detected by ADB then registration process is done.

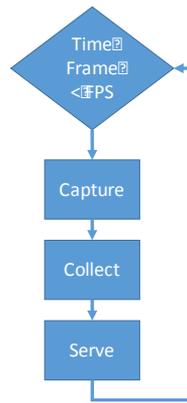


Fig. 5. Application main loop

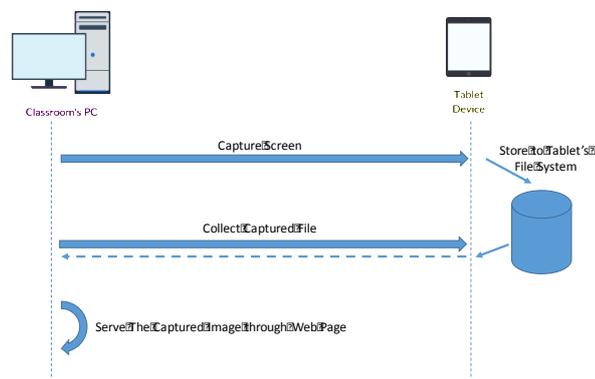


Fig. 6. Overall steps of tablet screen casting process

B. Screen Cast Step

The capture step is done remotely using ADB protocol [5]. Basically, ADB works the same as remote shell access. It is supported on every android device. The ADB is used to execute remote capture screen command. Then the captured screen will be stored in local file system of the tablet device.

ADB is chosen to realize the capturing functionality because it already supported in every android device. So the level of compatibility is high and reduced dependency application on the tablet device side. It helps our aim of creating a standalone PC application.

Next step, the captured screen file is collected from tablet device's file system into the PC's file system. To collect this file, we use ADB protocol again, which allow us to download file from tablet device to the PC.

The next step is to serve the captured screen file. Most vendors ([1], [2], [3]) use a separate application to act as video streaming client. Our system uses a different approach. We use Web Application Approach. We use HTTP Multipart [6] which could be used to emulate video streaming. And by using HTTP approach, our system doesn't need separate client application. Only browser needed. Where almost every computer has a browser installed nowadays. So it can reduce the number of applications that users have to install. Which means, it could be installed easily on classroom's average hardware specification PC.

By using HTTP approach, we use web application as a screencast viewer. To serve the web application, we use NodeJS [7] which is lighter than traditional web server such as Apache HTTP Server([8], [9]). The use of it has possibilities to be packaged into runtime binaries by using packager such as Electron [10] and Node Webkit [11].

The details of the system are as follows. To build the system, there are several software stacks that are used as show in Fig. 7. The system consists of two main parts. The first part is for capturing tablet screen and the second part is for displaying the captured screen.

Both parts are realized using several software stacks as shown in Fig. 8. NodeJS is used as the glue between capture system and display system. ADB [5] is used to execute command remotely on tablet device. FFmpeg [12] is used to manipulate captured image.

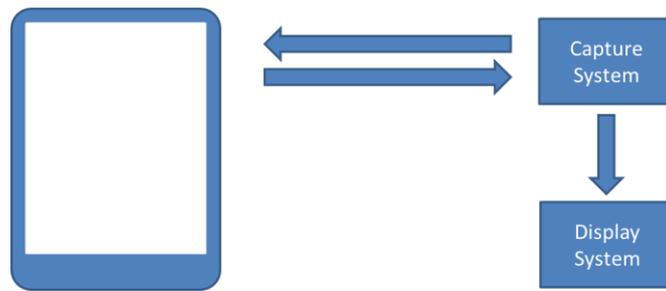


Fig. 7. System Diagram of Receiver

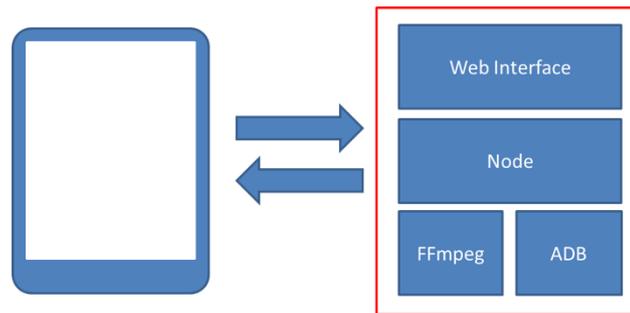


Fig. 8. Software stack (right) and communication diagram (arrows)

The system flow is as follows, through ADB, a command to capture tablet screen is executed remotely from a PC. Then the captured screen is converted to particular image format using FFmpeg. The converted result, then retrieved to a PC through ADB. In the PC, the converted image is delivered through HTTP Multipart[6]. The specific multipart used is multipart/x-mixed-replace which allows emulating streaming over HTTP.

III. REVIEW OF SIMILAR TECHNOLOGIES

In this section, we reviewed several similar technologies and compare it with our solution.

A. Comparison with Google Chromecast[4]

To be able to cast tablet screen to a projector, tablet device need running Android 4.4.2 or higher. But not only that, because of the technology that it used, the cast screen feature only works well on selected devices[4]. And to be able to use Google Chromecast, it has to be connected to the internet. Currently, Google Chromecast only support direct connection to the internet, it doesn't support proxy based internet[13].

The screen casting activity is initiated from the tablet. So, anyone at anytime could cast their tablet to the screencast receiver. This kind of approach may have an unwanted outcome when, for example, the teacher is currently using the screen cast to deliver a presentation, then accidentally, the student initiated screen casting in their device. This kind of possibility makes a wireless presentation as another problem in the classroom.

B. Comparison with Miracast[2]

To use Miracast, a tablet device needs to run Android 4.4 or higher. On the receiver side, it must use Miracast certified device. The hardware setup is similar to Google Chromecast, where we need a compatible Miracast dongle to be connected directly to the LCD projector. Then after connecting to the same Wi-Fi network, we can cast to this dongle using wireless display casting option in the Android 4.4+ device.

This solution requires a dongle to be connected to the LCD Projector. Assuming a classroom with already connected PC to LCD Projector, it will require disconnecting the VGA cable from the PC to the dongle. It's become an additional hassle to switch back and forth.

C. Comparison with Samsung AllShare[3]

Samsung selected tablet device must be used to use Samsung AllShare. And it only supports a subset of Samsung tablet devices. The supported device is lesser than Google Chromecast. Only Samsung flag ship product is supported. This system also uses dongle as a receiver. So the setup in the classroom should be connecting the dongle to LCD projector directly. As stated before, in a classroom with a PC already connected to LCD Projector, it will become an additional hassle to switch back and forth between dongle and a PC.

D. Comparison with MirrorOp[15]

MirrorOp is software based solution for screen casting. It consists of sender and receiver applications. It has rather advanced setup in which the tablet device has to be rooted[20] first to be used as a sender. Rooting activity is hard and very risky. If someone makes mistakes when rooting a device, it may cause the device to be malfunctioned. This rooting requirement is one of main drawback of screencasting solution offered by MirrorOp.

E. Comparison with AirParrot[16] and Reflector 2[17]

AirParrot and Reflector 2 is a pair of sender and receiver applications from AirSquirrel. It can send the screen cast to other proprietary receivers such as Google Chrome Cast and AppleTV. So, basically AirParrot use the same protocol used by Google and Apple. AirSquirrel also has its own receiver called Reflector 2. The receiver software not only receive screen cast from AirParrot, but also can receive screencast from Google Chrome Cast and AirPlay Mirror. This solution is all software based.

F. Comparison with 2nd Screen Platform by Kovačič[19]

2nd screen platform doesn't work for screencast activity. It works as screen expansion for the tablet device. In the implementation, Kovačič uses NodeJS as main platform for implementation.

There is common limitation that interest us, in which they have support on select device and/or need internet connection. We consider both characteristic as a limitation that need to be addressed. Our solution solves this limitation as it doesn't need internet connection and can be used on a device running Android 4.3 and lower. This work also has similarity with Kovačič[19] which uses Nodejs. But differ in the usage scenario. Where in Kovačič, the TV works as a second screen, where in this work, the tablet is used as a screencast source.

We evaluated the system using a setup as follows. On the PC, we used Google Chrome Browser as an interface to the receiver system. Using the web interface, we can select android tablet to be streamed. We evaluated several rate setting or FPS. At each rate setting, the evaluated parameter is delay. The delay is measured by manual counting using stopwatch. This measurement technique is used to get closer to get the delay value that will be perceived by end users. Which is important to help decide whether this system have the potential to be used in the classroom or not.

The result of the evaluation is shown in Fig. 9. On average, the perceived delay is between 1.5 seconds to 2 seconds. Where on average usage of presentation activity, it could be acceptable if the presentation is using text and image slide.

The next evaluation is the variation of capture width size compared with the resulting delay. The result of the evaluation is shown in Fig. 10. In this evaluation, the system is evaluated for several screen capture width. Which is independent from device screen width. The screen capture width that is tested is 320, 640, and 1024. Logically, the wider the screen capture, the more bandwidth it use, and more delay introduced. From the result, we can see that the average delay of each screen capture width is between 1 and 2.5 seconds. It still has the possibility to be used to support presentation in the classroom.

In short, from the delay test, we can conclude that this wireless presentation system setup, have the potential to be used as a wireless presentation system in the classroom.

Next, we evaluate the setup step needed to install this system. Here, we assumed that the hardware is available and ready to use based on the specific hardware setup as shown in Fig. 2. Based on this assumption, the install step will cover about setting up the PC to be able to do a live record on android screen.



Fig. 9. Various Setting Delay Test Result

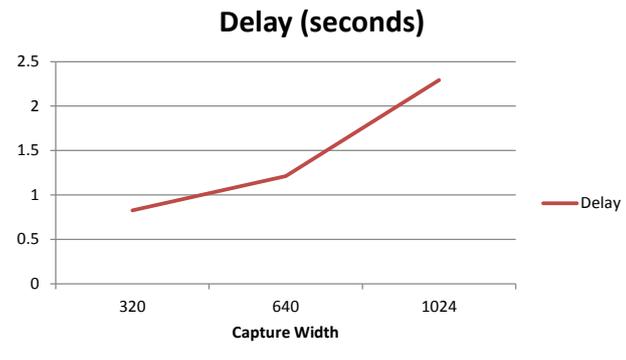


Fig. 10. Various Screen Capture Width Delay Test Result

There are two main steps of setting up the system:

1. Set up the PC as a live recorder
2. Register tablet device for wireless ADB

Here are the steps to set up the PC:

1. Install NodeJS
2. Download sji-android-screen-capture
3. Install ADB by install Android SDK
4. Install FFmpeg
5. Run NodeJS app

That is all the steps that is required to prepare the PC to start live recording of the tablet. The next step is to register the tablet device to be able to connect wirelessly using ADB protocol. There two ways of connecting ADB wirelessly. The first is by rooting the tablet device, then install an application to switch on wireless ADB support on the tablet device. The second way is to connect the tablet device by USB to the PC, then activate wireless ADB from the PC.

IV. CONCLUSION

The proposed approach has the opportunity to be installed in the classroom scenario with setup as assumed as shown in Fig. 2. The main advantage is the support for low end Android device and independent of internet connection. It also uses a different approach in the initiating screen cast. Where the cast is initiated in PC. So that, the only person that could use the projector is the one who has access to a PC connected to the projector. This approach can help teacher to control who is currently using the projector. The teacher is the one who decide which tablet is active. This approach is in contrast with the currently used approach by vendors where the projector can be accessed by anyone at anytime. This kind of approach is not preferable in a classroom where students could accidentally access the projector while the teacher is currently using it.

REFERENCES

- [1] Thomas Murphy, Christopher Williams, Frank Katz, "A Comparison of Tablet Computers for Use as a Class Presentation Platform", ASEE Southeast Section Conference 2012
- [2] Wi-Fi Alliance, "Wi-Fi Certified Miracast", (<http://www.wi-fi.org/discover-wi-fi/wi-fi-certified-miracast>) accessed at October 6, 2015
- [3] Samsung, "AllShare Cast Dongle", (<http://www.samsung.com/global/galaxys3/allsharecastdongle.html>), accessed at August 4, 2015
- [4] Google, "Google Chromecast", (<http://www.google.co.id/chrome/devices/chromecast/>), accessed at August 4, 2015
- [5] Android Developer Team, "Android Debug Bridge", (<http://developer.android.com/tools/help/adb.html>) accessed at August 2, 2015
- [6] E. Levinson, "The MIME Multipart/Related Content-Type", (<https://tools.ietf.org/html/rfc2387>) accessed at October 6, 2015
- [7] Joyent, "Node.js", (<https://nodejs.org/en/>) accessed at October 6, 2015
- [8] A. Lyman Chapin: Communication Systems: The State of the Art : IFIP 17th World Computer Congress, TC6 Stream on Communication Systems, the State of the Art, August 25-30, 2002, Montréal, Québec, Canada
- [9] Andrew B Robinson, "Why are event-driven servers so great?", (<http://andrewbrobinson.com/2012/01/27/why-are-event-driven-servers-so-great/>)
- [10] Github Team, "Electron: Build Cross Platform Desktop Apps with Web Technologies", (<http://electron.atom.io/>) accessed at January 18, 2016
- [11] NWJS.io Team, "NW.js", (<http://nwjs.io/>) accessed at January 18, 2016
- [12] Ffmpeg Team, "A Complete, Cross-Platform Solution to Record, Convert, and Stream Audio and Video", (<https://www.ffmpeg.org/>) accessed at August 3, 2015
- [13] Google, "Optimized devices for Android Screen casting", (<https://support.google.com/chromecast/answer/6293757>) accessed at December 19, 2015
- [14] Chromecast Help Forum, "Problems trying to set up a chromecast in school that uses a proxy", (https://productforums.google.com/forum/#!topic/chromecast/hH7xz2Fk_rs), accessed at January 31, 2016
- [15] MirrorOp, "MirrorOp", (<http://www.mirrorop.com/>), accessed at February 1, 2016
- [16] AirSquirrel, "AirParrot", (<http://www.airsquirrels.com/airparrot/>), accessed at February 1, 2016

- [17] AirSquirrel, “Reflector”, (<http://www.airsquirrels.com/reflector/>), accessed at February 1, 2016
- [18] Google, “Project your Android device’s screen”, (<https://support.google.com/nexus/answer/2865484?hl=en>) accessed at September 1, 2015
- [19] Kovačič, A., and Jože Guna. "A Platform Enabling 2nd Screen Functionality for Mobile Applications." Proceedings of the 21st International Electrotechnical and Computer Science Conference, Portorož. 2012.
- [20] How-To-Geek, “What’s the Difference Between Jailbreaking, Rooting, and Unlocking”, (<http://www.howtogeek.com/135663/htg-explains-whats-the-difference-between-jailbreaking-rooting-and-unlocking/>), accessed at February 1, 2016
- [21] Hereshem, “Android Screen Capture”, (<https://github.com/hereshem/sji-android-screen-capture>) accessed at August 2, 2015