

# Mirror Cast

<sup>[1]</sup> V Dinesh Reddy <sup>[2]</sup> Dr. C Balarenga Durai, <sup>[3]</sup> Prasad B

<sup>[1]</sup>II/IV, <sup>[2]</sup><sup>[3]</sup>Associate Professor

<sup>[1]</sup><sup>[2]</sup><sup>[3]</sup> Department of CSE, Marri Laxman Reddy Institute of Technology and Management  
(MLRITM) Hyderabad

<sup>[1]</sup> dineshreddy1994@gmail.com <sup>[2]</sup> balarengadurai@mlritm.ac.in <sup>[3]</sup> bprasad@gmail.com

**Abstract:** Chrome cast is a small HDMI dongle device which is of 72mm long which provide users to stream online videos, music ,games to the television of high definition or house audio systems . Chromecast which is a digital media player developed by Google it directly streams audio and videos on HD TV via WI-FI or a local internet. Users can also cast personal computers screen as well as cast enabled android devices to HD TV. Chromecast streams content from the cloud, so you get the highest quality 1080p HD video and surround sound. With 2.4 / 5GHz Wi-Fi support built in, which keeps video resolution high definition and buffering at low. Chromecast comes with micro USB cable with source of energy as adapter . Also setting up of Chromecast is also simple and easy. Supported mobile apps for Chrome cast are available for iOS 7.1+, Android 4.0+ and web based apps enabled for Chromecast are available on computers enabled Google Chrome (on Windows xp,7+, Mac OS 10.0+, Chrome OS for Chrome books running Chrome 28+) with the installation of the Cast extension embedded in the Google chrome browser. The application configures Chromecast and connects it to user based Wi-Fi network. Once Chromecast setup and configured successfully, users can use any Chromecast compatible application (such as YouTube, Netflix, HBO Go, Google Play Movies & TV Motion Tennis Cast ), tap the Cast icon, and then the content will play on the TV and provides information about the "Backdrop" images shown on the television. Using Chromecast you can turn your TV screen into a game board, racetrack, dance floor or trivia quiz. With Chromecast, user can also play games using their personal phone as their controller, while watching the action live on the big screen.

**Keywords:** Cast, Casting, Cast-Enabled Apps, Mirror Cast, Cast Extension, Chrome Browser, HDMI Port, Cast Button, Mirror, Streaming, Streaming Audio, Streaming Video.

## I. INTRODUCTION

The fact that Connect offers a comprehensive platform support is one of the unique selling points for the product. The application developer could implement Connect is his set of applications and reach a scenario where the same multi-screen experience is achieved whether the end user has a smart TV and a iOS device; or a LG smart TV and an Android device. After the release of the Chromecast, some end users (of this application powered by Connect) would like to be able to use their Chromecast as the first screen platform. Consequently, the purpose of this is to evaluate the possibilities of adding support for Chromecast devices in the Connect ecosystem, and also to provide recommendations on how to achieve this. The aim is to develop a solution that does not limit any functionality of neither Connect nor Cast, yet keep a simple interface towards implementers of the solution.

### History:

At the time of Chromecast launch, four compatible apps were available: YouTube and Netflix were supported as Android, iOS, and Chrome web apps; Google Play Music and Google Play Movies & TV were also supported, but originally only as Android apps. Additional Chromecast-enabled apps would require access to the Google Cast software development

kit (SDK). The SDK was first released as a preview version on July 24, 2013. Google advised interested developers to use the SDK to create and test Chromecast-enabled apps, but not distribute them. While that admonition remained in force, Chromecast-enabled applications for Hulu Plus and Pandora Radio were released in October 2013, and HBO GO in November. Google invited developers to a two-day hackathon on December 7 at Googleplex, its Mountain View head quarters, offering the opportunity to test drive the SDK's "upcoming release". The session attracted 40 developers from 30 companies and was followed by 10 additional apps, including Plex, Avia, and Realplayer Cloud. Google opened the SDK to all developers on February 3, 2014. In its introductory documentation and video presentation, Google said the SDK worked with both Chromecast devices and other unnamed "cast receiver devices". Chromecast product manager Rish Chandra said that Google used the intervening time to improve the SDK's reliability and accommodate those developers who sought a quick and easy way to cast a photo to a television without a lot of coding. Google also made the SDK a part of the Google Play Services framework, thereby giving users access to new apps without having to update Android itself. Over time, many more applications have been updated to support Chromecast. At Google I/O 2014, the company

announced that 6,000 registered developers were working on 10,000 Google Cast-ready apps by the following year's conference, the number of compatible apps had doubled. Google's official list of compatible apps and platforms is available on the Chromecast website. The past few months have seen rapid growth for Chromecast, thanks in part to Google's open source philosophy. Google recently hosted a Chromecast hackathon for 40 developers from over 30 countries leading up to the release of its software development kit (SDK) to programmers at large. One of the results ? More apps!



**Fig 1:** Chrome cast connection example

The device doubled its stable of mainstream apps back in December of 2013 with the inclusion of some powerful new tools like Plex and RealPlayer Cloud. More recently, Chromecast has added even more familiar faces, and has also supplemented its current offerings with new tools and features.

## II. THE WORKING PRINCIPLE

To implement a Cast enabled application one need to consider both the sender and the receiver side of the Cast environment. This section will give an introduction to the ecosystem from an application developer viewpoint.

### Receiver:

The receiver application running on a Cast device is a web application that is developed using regular Hypertext MarkupLanguage (HTML), Cascading Style Sheets (CSS), and JavaScript. The Cast receiver API is a JavaScript library which needs to be included in the application.



**Fig 2:** Chrome cast device

The Chromecast runs a stripped-down version of Google's Chrome web browser. But even though it runs web applications, the hardware of the Chromecast is highly optimized for media playback, so implementing a application with a lot of heavy animations will not result in a good user experience even though it is totally doable from an implementation point of view. Clearly, the purpose for the Chromecast is not to provide full-fledged applications with heavy UIs (like the ones running on a smart TV), but to provide a simple way to consume content on the big screen. The receiver application, which is supposed to be hosted somewhere on the web, must be declared within Google's Cast SDK Developer Console. The developer console is a self-administration tool for application developers where they can declare receiver applications and development devices. When declaring a web application as a receiver application, one receives an application ID that is needed in the companion application when establishing a connection to a Cast device. When the Cast device receives a request with a specified ID, it looks up the application ID on Google's servers, and browse the corresponding website. Consequently, there are no applications that need to be downloaded to the Chromecast in beforehand. There are three types of receiver applications that can be implemented by the developer: the default media receiver, the styled media receiver, and the custom receiver.

### Default media receiver:

The default media receiver is a pre-built receiver application provided by Google. The application is a non-branded media player that supports the basic media playback commands such as play, pause, seek, volume changes etc. Any developer can use the default media receiver without any restrictions.

### Styled media receiver :

The second kind of receiver is the styled media receiver. This is basically the same application as the default media receiver but it allows the developer to override some CSS to brand the application with his own logos, images, colors, and so forth. To use the styled media receiver, the developer needs to register his application in the Cast Developer Console and provide a uniform resource locator (URL) to a CSS file.

**Custom receiver :**

The third and last receiver is a custom receiver, which is a regular web application that the developer can develop using HTML, CSS and JavaScript. The developer must implement the Cast JavaScript API and must handle all messages that are received by the application and develop corresponding functionality. This option is required if the developer wants to add more functionality than the default or styled media receiver can provide, for example support for HTTP Live Streaming (HLS) or level 1 digital right management (DRM) support such as Widevine or PlayReady. The receiver application can manage multiple connections to different senders simultaneously, enabling great multi-user experiences such as games or queuing of media. But as stated before, the hardware is limited and such implementations should be developed with a rigorous quality assurance process. Debugging of the receiver applications running on a Cast device can be performed with the Chrome Developer Tools. This is done by browsing the IP address of the Cast device on port number 9022, in a Chrome web browser.

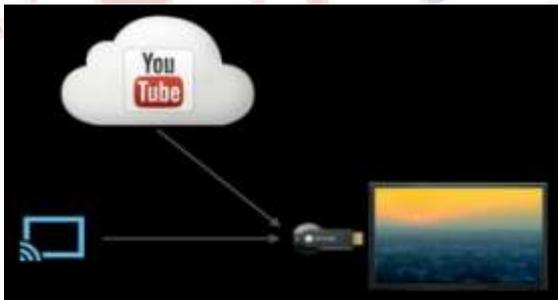


Fig 3: YouTube casting

**Sender:**

As stated before, the sender application is an application that runs on Android, iOS, or in a Chrome browser. In every case, the developer must implement a specific API for the platform. As stated earlier, this thesis will focus on the Android platform, but the other platforms behave in a similar way. The following text will introduce the Android Cast API for the reader, and does not have the purpose of giving an extensive walkthrough on how to develop a production ready application with Cast support. The general flow to establish a

connection with a receiver application from an Android application is as follows:

- Initiate a search for devices on the network
- The application receives a Cast device chosen by the end user
- Establishing a connection to the Google Play services
- Launch an application on the receiver
- Establish a communication channel to the receiver application

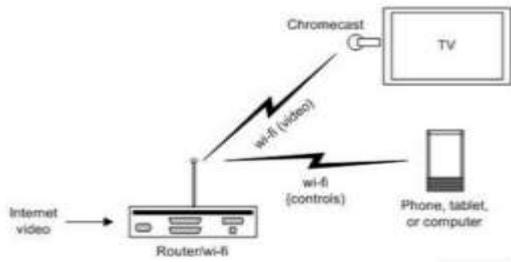
To initiate a search for devices, the developer can use the Android Media Router framework which is a support library that enables communication with external playback devices, such as screens or speakers. This can be seen in Listing 3.2. To start, we initialize the Media Router itself, the MediaRouteSelector which acts as a filter for devices, and the MediaRouterCallback object in which we receive the chosen device. As seen, the application ID (APP ID) received in the Cast Developer Console is entered in the initialization. The actual scan is then started when the MediaRouter.addCallback method is called, which is recommended to do in the onResume method of the Android Activity class.

**III. FUNCTIONS**

The primary use case for the Chromecast is to send content from a second screen to a big screen. The Chromecast streams media, over the Internet Protocol (IP), and the playback can be controlled from the companion application. Since the Chromecast is controlled only by companion applications, no other remote control is needed. The general flow of sending content from a sender application to a Chromecast is as follows:

- Launch the sender application, for example an Android application.
- Press the Cast button and select the device which you like to connect to.
- Choose some content on the sender application that will start playing on the Chromecast.

There are two fundamental ways of sending content to a Chromecast device: as remote playback, and as secondary output.



**Figure 4:** Diagram of the Chromecast solution

#### Remote Playback:

One of the Chrome cast's biggest strength over other streaming solutions can be seen. Streaming techniques that mirrors the screen of the companion device rely on the processing power of the companion device itself since it is the source of the stream. The Chromecast removes this dependency, since it can retrieve content directly from its source on the web, decode and play it by itself. This solution results in a lot of saved processing power on the companion device compared to a mirroring approach. Control of the playback is performed via messages that are sent to the Chromecast via a proprietary protocol over the local Wi-Fi network. So usually, the companion device browse a media library on the web and then sends a "play this specific media" message to the Chromecast, which in turn streams the media it by itself.

#### Secondary Output:

The second alternative to send content to the Chromecast is actually to mirror parts of the companion device screen. The Google Cast extension in the Google Chrome web browser lets users mirror a specific tab in the browser. There are also experimental functions that give the opportunity to the entire computer screen on the Chromecast. This type of streaming requires a lot of processing power in the sender device, which can contribute to a bad user experience on low-end devices. Some settings can be adjusted to achieve a better performance due to lower quality but as for now, this functions is stated as beta and only available for the Google Chrome web browser. This option is not seen as the primary use case for the Chromecast. These use cases is possible due to the Cast ecosystem, which contains receiver applications, running on a Cast device, and sender applications, running on Android, iOS, or in a Chrome web browser. The sender applications can discover Cast devices on the local Wi-Fi network, launch the receiver application on it, establish a channel between them, and communicate over that channel.

#### IV. ADVANTAGES AND DISADVANTAGES OF THE TECHNOLOGY

##### Advantages:

- ❖ **It's the smallest device around:** Unlike other set top boxes, the Chromecast takes up virtually no space. If you have an open HDMI port and a place to plug in the power, you'll never even see the little device.
- ❖ **(Some) apps are group activities:** It's inevitable: when you have company over, one person wants to play a YouTube video, then everyone wants to join in. One of the neat things about how the YouTube app handles the Chromecast is that anyone can do so, and they can even add a video to a shared playlist to make things easy.
- ❖ **It's insanely cheap:** It should go without saying that the biggest advantage the Chromecast has is its price. Not only is it normally \$35, which is about the cheapest you can get for a device like this, but you can often find deals where you get a discount or even a free Chromecast with the purchase of something else.
- ❖ **Your phone is your remote:** While remote controls have been one of the most beloved technological advances of the 20th century, they're also very rudimentary and easy to lose. We tend to be much better at keeping up with our phones. While Apple TV users can download a remote app for the iPhone, the Chromecast remotes tend to be cross platform.
- ❖ **More importantly, your app is your remote:** With the Chromecast, your "remote control" is the native app of whatever you're watching rather than some direction pad and set of buttons. It's an app designed for mobile, and instead of clicking around a menu, you're picking your media from the app's normal menus. It's a much nicer experience. While not every company does this perfectly, the big names like Netflix, Hulu, and YouTube have done a good job. Note also that some devices like the Roku have smartphone-based remote control apps, but few are terribly well designed.
- ❖ **Tab casting is always a fallback:** While the feature can be a bit wonky, the Chromecast is able to stream video from any Chrome tab on your PC to your TV. It's not an ideal solution, but if you want to play videos from a less popular streaming site (say, a television network's own video player), tab casting is always an option. On an Apple TV or Roku, if a service isn't supported you're just out of luck.



**Fig 5:** Applications of ChromeCast

**Disadvantages:**

- ❖ **Many apps are still missing:** Want to watch Amazon Instant Video on your TV? Sorry. You're currently out of luck. Tab casting may be a decent solution if you want to pull out a laptop, but not all apps are so lucky. The advent of the Chromecast SDK means that we may see a flood of new options, but for now, there are still some areas where it's a waiting game.
- ❖ **You have no physical remote:** Yes, this very thing was listed as a feature in the previous section, but it cuts both ways. If you have a member of the house that doesn't own a Smartphone (for example, your children), it won't be easy for them to use the Chromecast.
- ❖ **It lacks a native interface:** One of the silent advantages to using a standalone streaming box is that everyone can browse together. A big wall of Netflix movies is visible to everyone in the room and each person can offer their input as you look for something to watch. With a Chromecast, everyone has to crowd around the phone or tablet to see what's available, which isn't exactly ideal.
- ❖ **You might already have a set top box:** Perhaps the biggest disadvantage of the Chromecast is that it came out in 2013, years after devices like the Apple TV and the Roku were already in living rooms. For those that haven't bought one already, many Blu-Ray players and TVs also have many of these same functions built in. \$35 may be a pittance to pay for what the Chromecast can do, but if you already have the ability to do so, the money isn't very well spent.
- ❖ Since there's no on-screen interface, you have to know whether a particular app works through the device. However, the new

Chromecast mobile app has been updated to make it easier to find those apps.

- ❖ The original Chromecast had problems with Wi-Fi connectivity and they ended up having to release a separate adapter to try to solve the problem. I didn't have any problems with a Netflix movie I watched through the new model.
- ❖ While casting from a Chrome browser is a great feature in theory, it wasn't particularly well implemented in the original device. The audio was often out of sync with the video, especially if you started using a different application than Chrome. I really had high hopes that they would address this in Chromecast 2 but based on my initial usage, it's even worse. The audio/video lag is even more obvious.
- ❖ I'm not a gamer myself, so I may be wrong, but it looks to me like the list of games that Chromecast supports isn't as good as some of the other streaming media devices, which offer popular games like Crossy Road and Angry Birds. Plus, there's no game console or motion control remote, so you definitely are limited to playing the more casual types of games.
- ❖ You can't cast from a Kindle Fire tablet to a Chromecast and as mentioned earlier, there's no Amazon Video app for the Chromecast. You also can't cast from a PC other than through the Chrome browser.

**V. AREAS OF APPLICATIONS**

Some of the application of chrome cast are as follow

- ❖ Listen to Streaming Audio
- ❖ Cast a Chrome Browser Tab from Your Desktop
- ❖ Mirror Your Desktop
- ❖ Playing games
- ❖ Turn your home theater into a video junk box
- ❖ Mirror your Android phone or tablet
- ❖ Show off presentations from Google Drive
- ❖ Web browsing on TV
- ❖ Turn your TV on from your mobile device

**VI. CONCLUSION**

While there are plenty of alternatives out there to make an older television smart, Chromecast is one of the most affordable yet functional. What it lacks in apps now is made up for by its support for browser tab sharing. And the fact it works for both Android and iOS opens up its potential user base even wider. With a little time, after it is released internationally, it could be the new device developers can't ignore when making apps.

Its simplicity of use makes Chromecast a viable upgrade for anyone that enjoys streaming online video, music and photos.

#### REFERENCES

- [1] Watson, Todd (July 26, 2013). "Introducing Google Chromecast". Inside Investor. Retrieved July 26, 2013.
- [2] Goode, Lauren (September 29, 2015). "Google's new Chromecast has a new look, catches up to modern Wi-Fi standards". The Verge. Vox Media. Retrieved September 29, 2015.
- [3] Evangelho, Jason (July 24, 2013). "Google's Chromecast A Brilliant Play For The Living Room -- Especially With \$35 Price Tag". Forbes.com. Retrieved July 24, 2013.
- [4] DiPane, Jared (July 8, 2015). "Ethernet adapter for Chromecast makes its way to the Google Store for \$15". Android Central. Mobile Nations. Retrieved July 8, 2015.
- [5] Turner, Adam (July 14, 2014). "Hands on: Chromecast Android screen mirroring". The Sydney Morning Herald.
- [6] Maybury, Rick (April 5, 2014). "Should I buy Google Chromecast?". The Telegraph. Telegraph Media Group. Retrieved July 3, 2014.
- [7] Anthony, Sebastian (September 4, 2013). "HDMI 2.0 released: 18Gbps of bandwidth allowing for 4K @ 60 fps, 32 audio channels". ExtremeTech. Ziff Davis. Retrieved July 3, 2014.
- [8] Stein, Gabe (July 24, 2013). "Wow! Google's Chromecast Is Amazingly Hackable". Co.Labs. Fast Company. Retrieved January 9, 2014.
- [9] "GET STARTED WITH CHROMECAST: What's Under the Hood? – Chromecast Help". Google. Retrieved August 7, 2014.
- [10] Robertson, Adi (July 24, 2013). "Google reveals Chromecast: video streaming to your TV from any device for \$35". The Verge. Vox Media. Retrieved September 19, 2013.

