Android4TV – Development guide

iWedia

Filename	Android4TV-Development guide
Version	v1.0
Classification	PUBLIC
Status	Draft
Date	January 9, 2016
Author	Nikola Vranić;Vladimir Živkov;Jaroslav Hlavač;Ilija Đukić;Sreten Tanacković;Marko Živanović;Nikola Špirić

SUMMARY

PREFACE

Purpose of this paper is to document is to explain how to setup environment, run Android emulator with prebuild iWedia software and how to write application utilizing Android 4 TV library and API.

Idea is to provide our open source community way to development application on Android emulator with full Teatro 3.5. Because of that, we had to port out solution to x86 platform and made addition changes to fit emulator constraints.

DOCUMENT HISTORY

Ver.	Description of changes	Author(s)	Date
1.0	Initial version	Nikola Vranić	2015-09-01

RELATED DOCUMENTS

#	Title	Description	Version/Date
1.0	Software architecture	Teatro 3.5 solution software architecture	2015-09-01

DEFINITIONS/ACRONYMS/ABBREVIATIONS

Definition/Acronym/Abbreviation	Description
AOSP	Android Open Source Project
SoC	System on Chip
TIF	TV Input Framework
PVR	Personal Video Recording
EPG	Electronic Program Guide

1 IWEDIA EMULATOR OVERVIEW

Emulator architecture is little bit different comparing to architecture on real device [1]. Main new components are software demultiplexer and file tuner. Since emulator doesn't have real tuner, idea is to use file tuner which is doing tune operation on file which is located on SD card: /mnt/media_rw/sdcard/iwediaemu.

To be able to start tune operation, file tuner parse file_tuner.ini file (/mnt/media_rw/sdcard/iwediaemu/file_tuner.ini) from which gets location of transport stream(s), frequency and bitrate.

```
[stream_0]
file="/mnt/media_rw/sdcard/iwediaemu/dvb-t_818Mhz.ts"
freq=482000000
bitrate=20866663
```

Once tune is successfully completed, software demultiplexer parse transport stream. Video and audio packets are sending instead to SoC library, to local host HTTP port 4567. Other packet like teletext and subtitle are continuing voyage as normal though middleware core: TV Player Service. Customers can freely put new transport stream and change file_tuner.ini configure file.

When TV application request channel change, iWedia TV input service will bypass playback start from Android4TV API to Android Media Framework and iWedia Media Player.

All other operations, like install channels, audio track change, caption display/change, parental control, TV guide remains same functionality as on real device by utilizing Android4TV API.

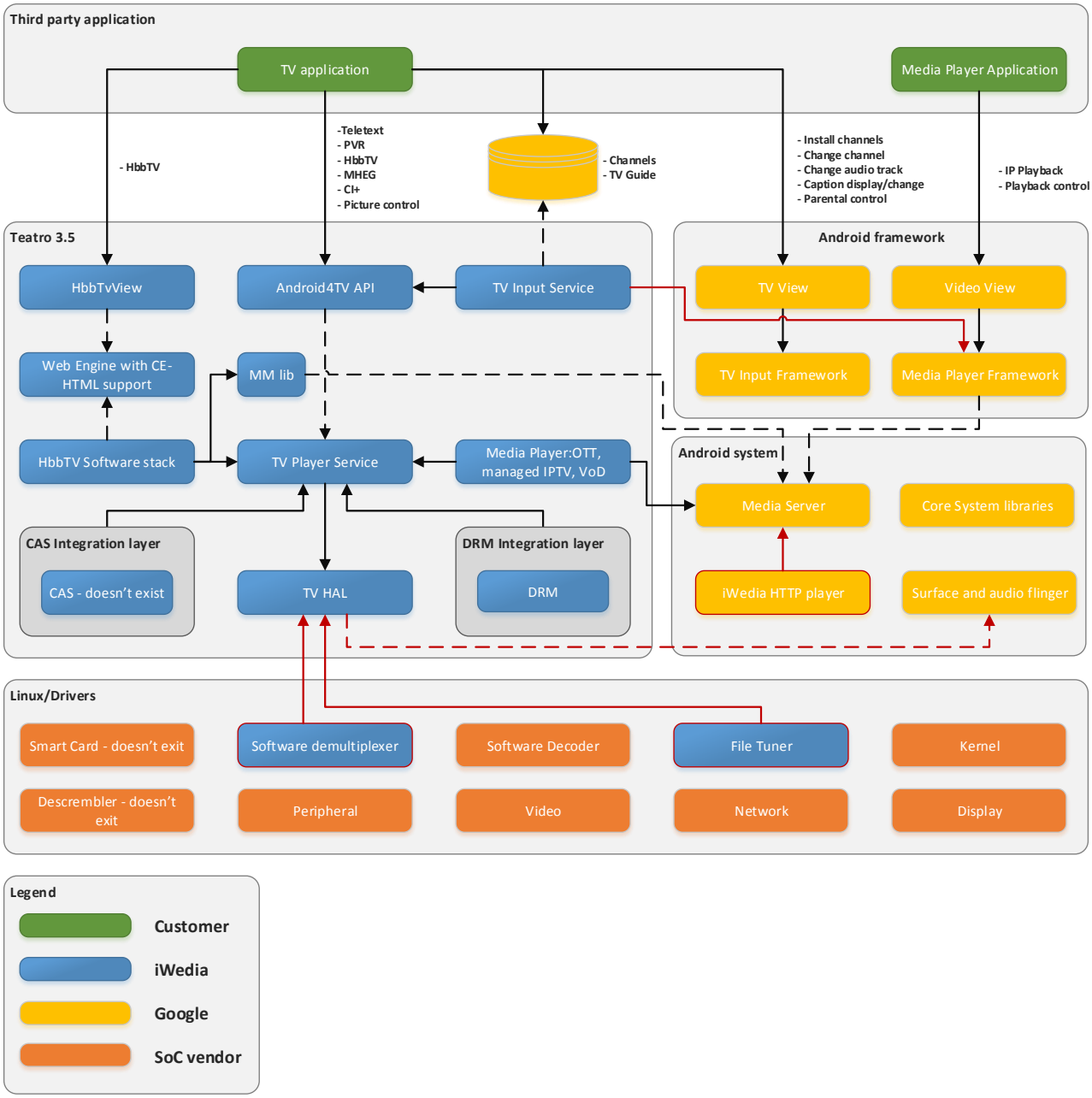

Figure 1 iWedia emulator solution overview

2 SETUP ENVIRONMENT

Since we still didn't figure out how to replace custom system image in some of Android TV emulator images inside Android SDK or how to extract emulator from Android Open Source Packet (AOSP) packet, we will explain how to download AOSP and run iWedia prebuild images from it. This is more time consumer way, but it could facilitate future development and testing to our customers and our self's.

Prerequisite: Ubuntu 14.04 x64 (It work's on other versions also, but we use official Ubuntu development version for AOSP)

1. Install Java

```
sudo apt-get update
sudo apt-get install openjdk-7-jdk
```

2. Configure java

```
sudo update-alternatives --config java
sudo update-alternatives --config javac
```

3. Install additional libraries and programs

```
sudo apt-get install bison g++-multilib git gperf libxml2-utils make python-
networkx zlib1g-dev:i386 zip
```

4. Download repo

```
mkdir ~/bin
PATH=~/bin:$PATH
curl https://storage.googleapis.com/git-repo-downloads/repo > ~/bin/repo
chmod a+x ~/bin/repo
```

5. Download AOSP
    - We use android-5.1.1_r9 version but it will probably work with other, newer AOSP versions.

```
mkdir working_directory
repo init -u https://android.googlesource.com/platform/manifest -b android-
5.1.1_r9
repo sync -j20
```

6. Setup AOSP environment for x86

```
source build/envsetup.sh
lunch aosp_x86-eng
```

7. Build AOSP

```
make -j20
```

3 RUN EMULATOR

1. Download latest prebuild image with iWedia software for Andorid4TV web page:

```
http://android4tv.iwedia.com/documents/iwediaemu-release-cw02-v1.0.0.tar.gz
```

2. Extract tar to AOSP root
3. Run emulator with iWedia prebuild images

```
croot
sh iwediaemu-release/current/run-emulator.sh
```

Figure 2 iWedia Android4TV emulator

4 BUILD APPLICATIONS

4.1 IWEDIA TV INPUT EXAMPLE

This example is based on Android4TV library and contains all TIF functionality from Android Lollipop:

- channel installation
- channel change
- caption display/change
- audio track change
- parental control
- TV guide (EPG)

Of course, Android4TV library covers much more functionality, like PVR, HbbTV, MHEG, reminders etc.

At the moment we are using GitHub to share this example and there are 2 important branches:

1. master: contains code for customer devices
2. emu: contains code for Android emulator

Code on master and emu branch is 99% same. Emu branch have some specifics needed to run iWedia TV input example on Android emulator.

Example can be downloaded from next location:

```
https://github.com/iWedia/iWediaSimpleTvInputService
```

Example is licenced under the Apache 2.0 license and current active version is 1.3.51.

4.1.1 Compiling and installing

At the moment, this example can be compiled with: gradle and in Android build system.

4.1.1.1 Gradle

Under Linux:

```
gradlew build
gradlew installDebug
```

Under Windows:

```
gradlew.bat build
gradlew.bat installDebug
```

Note that before app is installed, you must be connected to a device using ADB.

```
adb devices
adb connect IP_ADDRESS_OF_DEVICE
```

4.1.1.2 Android build system:

```
mmm external/iWediaSimpleTvInputService/
adb install -r
out/target/product/<ProductName>/system/app/iWediaSimpleTvInputService/iWediaSimpleTvInputService.apk
```

4.1.2 LiveTV application

To be able to test iWedia TV input example, you can use default TV application on Android TV: LiveTV. It becomes open source in Q3 2015. and can be downloaded with next instructions:

```
repo init -u https://android.googlesource.com/platform/manifest -b android-live-tv
repo sync -c –j20
. build/envsetup.sh
tapas LiveTv
make
```

4.2 CUSTOM APPLICATION WITH ANDROID4TV LIBRARY

To build custom TV application Android4TV library could be used.

Library is licenced under Apache 2.0 licence and current active version is 3.0.0. It can be manually downloaded from next link:

```
https://github.com/iWedia/repo
```

4.2.1 Compiling and installing

At the moment maven, gradle and Android build system are supported. Also library can be downloaded manually and included in Android Studio or Eclipse with ADT SDK project.

4.2.1.1 Maven:

1. In main pom.xml file add:

```
<repositories>
    <repository>
        <id>iWedia-Repo</id>
        <name>iWedia Public GitHub Maven Repository</name>
        <url>https://github.com/iWedia/repo/raw/master</url>
    </repository>
</repositories>
```

2. Add dependency where is needed:

```
<dependency>
    <groupId>com.iwedia.dtv</groupId>
    <artifactId>android4tv-framework</artifactId>
    <version>3.0.0</version>
    <scope>compile</scope>
</dependency>
```

4.2.1.2 Gradle:

1. In main build.gradle add:

```
allprojects {
    repositories {
        maven {
            url 'https://github.com/iWedia/repo/raw/master/';
        }
    }
}
```

2. Add dependency where is needed:

```
dependencies {
```

```
    compile group:'com.iwedia.dtv', name:'android4tv-framework', version:'3.0.0'
}
```

4.2.1.3 Android build system

1. Download and copy android4tv-framework-3.0.0.jar (or other version) library to lib folder
2. Include prebuild library in application Android.mk file:

```
LOCAL_STATIC_JAVA_LIBRARIES += \
    android4tv-framework \
```

3. At the end of application Android.mk file prebuild definition:

```
include $(CLEAR_VARS)

LOCAL_PREBUILT_STATIC_JAVA_LIBRARIES := \
   android4tv-framework:./libs/android4tv-framework-3.0.0.jar \

include $(BUILD_MULTI_PREBUILT)
```

4. Build Android application with flavour m* command: m, mm, mmm