

### **How Android Selects Hardware or Software Decoding**

1️⃣ **Application Requests Video Playback**

* The **Application Framework** (e.g., MediaCodec API) requests video decoding.

2️⃣ **Binder IPC Proxies Forward the Request**

* The request is sent to **MediaPlayerService** via **Binder IPC**.

3️⃣ **MediaPlayerService Determines the Decoder**

* MediaPlayerService.cpp (inside libmediaplayerservice) checks available **decoders**.

4️⃣ **Stagefright Engine Decides Software or Hardware Codec**

* **Stagefright (libstagefright) checks media\_codecs.xml**
* media\_codecs.xml lists available codecs:
  + **Hardware Codecs** → libstagefrighthw.so
  + **Software Codecs** → OMX software implementations (e.g., Google OMX)

5️⃣ **OMX Plugin Chooses the Codec Type**

* If a **hardware codec is available**, libstagefrighthw.so is used.
* Otherwise, it falls back to a **software decoder** (e.g., OMX.google.h264.decoder).

6️⃣ **OMX IL Component Handles Decoding**

* If hardware is selected, the **OMX IL Component** processes the video via the **VPU/GPU**.
* If software is selected, the **CPU handles decoding**.

7️⃣ **Final Output is Sent to the Renderer**

* Decoded frames are sent to **SurfaceFlinger** for display.

Default software decoder

<!-- Software Decoder Example -->

<MediaCodec name="c2.android.avc.decoder" type="video/avc"> <Alias name="OMX.google.h264.decoder" />

<Limit name="width" max="1920" />

<Limit name="height" max="1080" />

<Attribute name="software-codec"/> </MediaCodec>

FFmpeg software decoder

<MediaCodec name="ffmpeg.video.decoder.h264" type="video/avc">

<Alias name="OMX.ffmpeg.h264.decoder" />

<Limit name="width" max="1920" />

<Limit name="height" max="1080" />

<Limit name="bitrate" range="64000-20000000" />

<Attribute name="software-codec" />

</MediaCodec>

* name="ffmpeg.video.decoder.h264" → This is the **unique name** assigned to the decoder.
* type="video/avc" → This defines that the decoder supports **H.264 (AVC) video format**.

The **alias** provides an alternative name for the decoder.

OMX.ffmpeg.h264.decoder → This alias follows **OpenMAX (OMX) standards**.

This **restricts the maximum resolution** that the decoder supports.

**Max width:** 1920 pixels (Full HD 1080p).

**Max height:** 1080 pixels.

* Defines the **bitrate range** (in bits per second) that the decoder supports.
* **Minimum bitrate:** 64 kbps (64000 bps).
* **Maximum bitrate:** 20 Mbps (20000000 bps).
* Higher bitrates mean **better video quality** but require more processing power.

This tells **Android's media framework** that this is a **software decoder**.

It will use **CPU-based decoding**

### **Summary of the Flow**

1️⃣ Stagefright **asks OMX** for a hardware decoder.  
 2️⃣ OMX **searches for the codec** in /vendor/lib/omx/.  
 3️⃣ If a hardware codec is found, it is used.   
 4️⃣ If **no hardware codec is found**, OMX **returns an error** to Stagefright.   
 5️⃣ Stagefright **then searches for a software decoder** (SoftOMXPlugin).  
 6️⃣ If a software codec exists, Stagefright **loads it and decodes the video using the CPU.** 7️⃣ If no software decoder is found, **FFmpeg (if integrated) is used as the final fallback.**