RDK 3.0 and Secure Decryption on i.MX8MQ

Moorthy B S
moorthy.baskaravenkatraman-sambamoorthy@linaro.org
Agenda

- RDK 3.0 Features
- 64bit support on RDK Video profiles - media client & hybrid
- RDK Yocto build - Thud support
- RDK upstreaming
- Thunder Framework with WPE Webkit Browser
- Secure decryption using OPTEE on WPE with EME 3.0
Introduction

- **Reference Design Kit (RDK)**
  - open source software solution for connected homes that provides a common framework for powering customer-premises equipment (CPE) such as set-top boxes
  - standardizes the core functions of Video, Broadband, Camera and IoT devices
  - contains software components, tools and documentation that provides faster development of standard Linux based equipments

  Link - [https://wiki.rdkcentral.com/display/RDK/Overview](https://wiki.rdkcentral.com/display/RDK/Overview)

- **RDK Evolution**
  - RDK 1.0 - RDK base version
  - RDK 2.0 - TR69 Host Interface, HDMI-CEC, Storage Manager, QtWebKit, etc.
  - RDK 3.0 - Spark Framework, AAMP and WPE WebKit based RDK browser
RDK 3.0 Features

**Spark Framework** *(pxScene)*
- Open source, cross-platform windowing framework exposed to Javascript (JS) engine
- Core element for JS Engine *(pxCore)* contains Javascript Interface and Native windowing interface
- Framework application engine that contains JS bindings and handles 2D scene graph
- Uses ESSOS - simple library, companion of westeros compositor that can create application and runs as either native EGL or wayland clients

**Advanced Adaptive Media Player** *(aamp)*
- Media player based on gstreamer that plays media content over Internet Protocol (IP)
- Supports HTTP Live Streaming (HLS) and MPEG DASH streams

**WPE based RDK browser** *(rdkbrowser2)*
- Simple browser application that instantiates WPE WebKit
64 bit support on RDK - Overview

RDK Video Profiles

* Media Client - IP-only or QAM device profile mainly uses components such as Westeros (wayland) compositor, gstreamer media framework with basic open source components like Linux kernel, busybox, OpenSSL, UPnP and so on
* Hybrid - combination of both IP-only and QAM in a single device

Why RDK in 64bit?

- RDK currently available on 32-bit ARM platforms
- Freescale provides 64bit EGL library support for imx8 platforms

Why Thud Migration?

- RDK maintains yocto build system with daisy and morty support
- Freescale provides yocto meta (SoC) layer is based on thud
Efforts on 64-bit compatibility

### MSO Layer

**RDK 3.0**
- westeros
- rdkbrowser2
- app manager
- Aamp player
- Aamp gst plugin
- pxScene
- rtremote
- RDK logger
- gstreamer
- Playersinkbin (gst plugin)
- RMF streamer
- RDK media framework
- iarm-bus
- Service manager
- Device settings
- Storage manager
- iarm-hal
- ds-hal
- iarm-manager
- Host IF agent
- tr69
- systemd
- busybox
- core-utils
- RDK Base rootfs
- curl
- dropbear
- gupnp
- lighttpd

**SoC Layer**
- OpenGL / ES
- Linux Kernel
- Drivers
- decoders
- Gstreamer plugins

- Linaro Contribution on 64bit
- Open source / embedded components
- RDK open source components
- RDK components publicly open
## Challenges on 64-bit compatibility

<table>
<thead>
<tr>
<th>RDK Components</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>device settings</td>
<td>32-bit integer device handle changed to compatible pointer type (intptr_t)</td>
</tr>
<tr>
<td>pxCore, pxScene</td>
<td>Enabled ‘aarch64’ machine support</td>
</tr>
<tr>
<td>rtremote</td>
<td>‘aarch64’ support added for value reader and writer of IPC module <a href="https://github.com/pxscene/rtRemote/pull/4">https://github.com/pxscene/rtRemote/pull/4</a></td>
</tr>
<tr>
<td>rmfosal</td>
<td>32-bit ‘handle’ changed to compatible pointer type (intptr_t)</td>
</tr>
<tr>
<td>rmfgeneric</td>
<td>32-bit integer handles are changed to pointer type (uintptr_t) for compatibility. Handled ‘Port number’ type casting from integer pointer to integer.</td>
</tr>
<tr>
<td>rmfstreamer</td>
<td>Type definition for 64-bit integers in 32-bit architecture are defined for machines other than ‘aarch64’. Used predefined macros (<strong>aarch64</strong>)</td>
</tr>
<tr>
<td>iarmmgrs</td>
<td>Passing ‘unsigned int’ (32bit) type to ‘size_t’ pointers in function arguments are generalized</td>
</tr>
</tbody>
</table>
RDK yocto build system

- RDK uses **yocto project** build system and currently it supports on daisy, krogoth and morty versions
- RDK uses openembedded components such as busybox, systemd, core-utils, GUPnP, curl and so on from openembedded-core yocto layer
- RDK doesn’t allow GPLv3+ / LGPLv3+ licensing components; uses meta-gplv2 layer
- RDK maintains all its components in different meta-layers such as
  - *meta-rdk*- provides RDK distros and contains base components, package groups and image bb files
  - *meta-rdk-video* - components for video profiles
  - *meta-rdk-ext* - enhancements / RDK specific changes on open source components
- RDK maintains meta layers meta-cmf, meta-cmf-video and others in order to override the URI's to point to open source repositories from [code.rdkcentral.com](http://code.rdkcentral.com)
Thud Migration Approach

All RDK meta layers are generic and it made compatible to Thud based system.
Switched open embedded layers revisions to Thud (Yocto 2.6)
- openembedded-core - thud branch, meta-openembedded - thud branch
- bitbake - 1.40, Toolchain - gcc 8.2
- SoC Layer - meta-freescale - master branch
- Meta-wpe - master branch - westeros is picked from WPE layer as Compositor plugin depends on westeros maintained in meta-wpe

A new layer (meta-cmf-freescale) is introduced to have SoC specific RDK changes.

Open embedded components thud version  components alternatives

systemd    - 1.239  openssl    - 1.1.1b  1.0.2o
glibc       - 2.28   gnuTLS      - 3.6.4    3.3.x
busybox     - 1.29.3  gmp         - 6.1.2    4.2.1
dropbear    - 2018.76 nettle     - 3.4      2.7.1
dbus        - 1.12
gnutuuu     - 1.0.2
Challenges on Thud Migration

- To check the compatibility and dependency for open source components from oe-core layer’s thud branch
- Meta-rdk-ext layer applies RDK specific patches over few open source components. Repatching the changes on new versions
- gcc 8.2 causes lot of build errors where turns warnings to errors. Fixes made on source code / removes compilation flags
- RDK Media player (aamp) - hangs and reboots the system
  - A function turns to infinite looping due to gcc 8.2 Optimization flags
  - Removed align-loops and align-jumps (-fno-align-loops -fno-align-jumps)
- Components build and run time dependencies
  - Latest yocto system introduces a task prepare_recipe_sysroot() that generates sysroots on work directory whereas old version generates sysroots on a common directory
  - Need to add some more dependencies explicitly to RDK components
RDK Media Framework (RMF)

- Media Framework based on Gstreamer but abstraction of general Source, Sink and Filter elements
- Implements different sources and sinks for DVR, QAM and http (HN)

**playersinkbin**

- A gstreamer based plugin (GstBin) that instantiates SoC specific decoders & sinks and completes a media pipeline
- Developed playersinkbin for i.MX8MQ with SoC specific video decoder plugin
- General gstreamer pipeline using playersinkbin

```
$ gst-launch-1.0 httpsrc location=<url> ! playersinkbin
tsdemux
```

![Diagram of media pipeline with gstreamer elements](image-url)
RDK Upstreaming

- Upstreamed 64-bit and thud compatibility changes to Comcast gerrit’s feature branch as these changes are huge and affected on many of the components
- Build and regression testing validation on existing RDK devices
- Later to Build and Functional tests, all changes will be proposed to respective stable branch of Comcast gerrit
- Once the changes are merged to the stable branch, that will be lands on RDK CMF repositories

Documentation

Efforts on 64bit and Thud compatibility changes are documented in RDK Wiki - https://wiki.rdkcentral.com/display/RDK/RDK+porting+on+64bit+i.MX8MQ+EVK+using+Thud+Yocto+2.6

Procedure to build RDK on i.MX8MQ documentation - https://wiki.rdkcentral.com/display/RDK/Build+Procedure+for+64bit+RDK+Media+Client+using+Thud+Yocto+2.6
Other Contributions

Westeros
- DRM mode selection based on native window size - [commit](#)
- Adding EGL CFLAGS on westeros_gl, essos
  - [westeros commit1](#), [westeros commit2](#) and [westeros commit3](#)
- Enabling dynamic mode selection module on Direct Rendering Manager (DRM)
  - [westeros commit](#)
  - [https://github.com/WebPlatformForEmbedded/meta-wpe/pull/337](#)
- wrp-c : replacing strncpy with memcpy due to gcc 8.x - [https://github.com/xmidt-org/wrp-c/pull/80](#)

WPE: EGL CFLags inclusion
- [https://github.com/WebPlatformForEmbedded/WPEBackend-rdk/pull/34](#)
- [https://github.com/WebPlatformForEmbedded/WPEWebKit/pull/607](#)

Playready CDM(i) on WPE: Playready 3.3 support fixes -
[https://github.com/WebPlatformForEmbedded/OCDM-Playready/pull/15](#)
Thunder Framework

➢ Open source simple framework for embedded device browser experience
➢ Plugin based framework with inter-plugin dependencies
➢ Framework core
  ○ a core element with web API, RPC and controller for loading plugins
  ○ Contains Json parsers, WebSocket and Web API capabilities
➢ Framework plugins
  ○ Plugin for browser - instantiates WPE webkit and controlled using Web API
  ○ Plugin for Compositor - creates westeros compositor for wayland display
  ○ Plugin for network - configuring network for connection
➢ Framework UI
  ○ Web based User Interface (UI) to control framework and plugins remotely
  ○ Opens the webserver and interact with plugins (start / stop)
Secure Decryption on WPE with EME 3.0

Thunder Framework

Compositor

EME App.

Open CDM

EME 3.0

WPE WebKit

MSE

Gstreamer pipeline

RPC ->

payload

opaque

<- Callbacks

Shared buffer

OCDM Server

Playready.drm

Widevine.drm

Clearkey.drm

CDM(i)

loads

loads

loads

Playready

Widevine

OPTEE

Playready TA

Widevine TA

From Linaro

Linaro Enhancement on WPE

WPE & Open source
Playready Decryption Flow

- openCDM implements in gstreamer part of WebKit and Thunder framework loads OCDM server as a plugin
- Shared Buffer implements from the Framework and used for Protected content and Opaque (clear) content data exchange

OCDM Server
- Key Systems configuration from a JSON file
- CDM(i) implementation library for each key system under a specified directory
- Creates OpenCDM session based on the key requested from License Server
- Uses Producer-Consumer threads to synchronously exchange encrypted and decrypted data through OCDM session

Open Platform - TEE
- A Trusted Application (TA) will be loaded for each Key Systems in TrustZone (OPTEE) and TEE session will be initialized
- Decryption does in the Trusted Zone (OPTEE)
Current & Future work

Current Activities

● RDK Upstreaming
  - Validation Functional & Regression testing on existing RDK devices
  - Upstreaming to Comcast gerrits
  - Landing changes to RDK CMF community

● Widevine Integration
  - OEM crypto library integration
  - CDMi implementation with DRM APIs

Future Activities

● Secure Data Path (SDP) integration for fully Secure Video playback
● Contribution towards RDK Camera profile use cases
● RDK 4.0 bringup with Yocto 2.7 (warrior) support
Thank you

Join Linaro to accelerate deployment of your Arm-based solutions through collaboration

contactus@linaro.org