RDK-C : RDK Media Streamer (RMS)

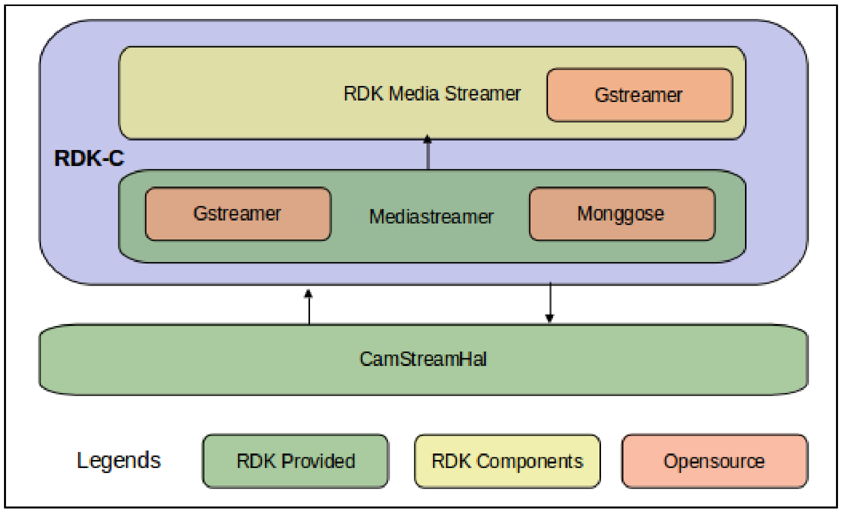
- Introduction
- Architecture
- Design Considerations
  - Gstreamer Soc Implementation for RPI-0 Camera
  - Gstreamer Implementation in RMS

Introduction

This page dedicated to understanding of High level design for RDK Media Streaming in R-Pi Zero.

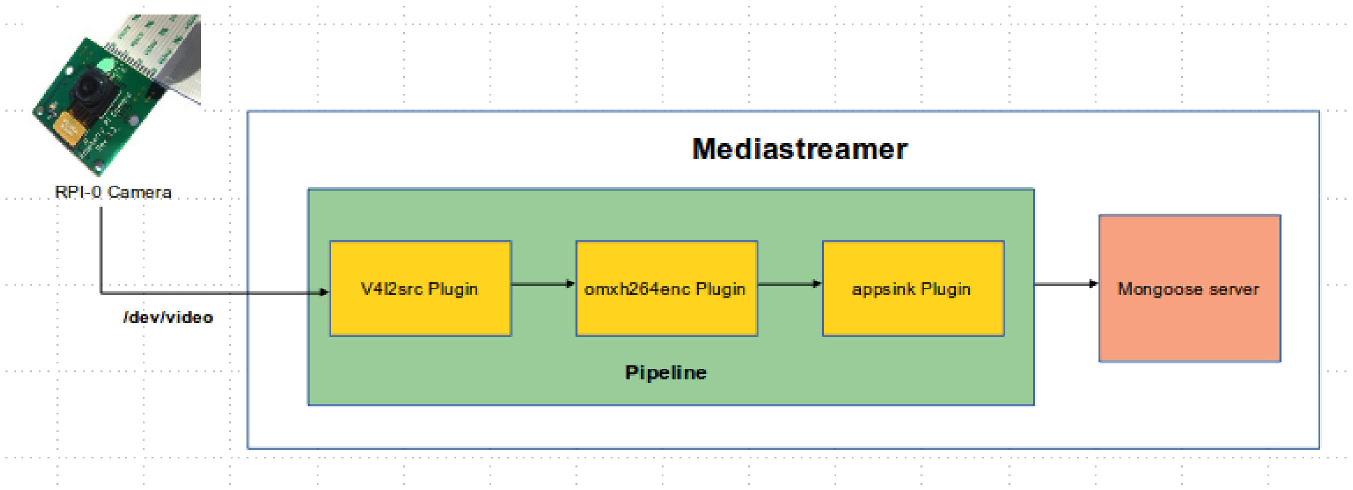
- Supported WiFi connection.
- v4l2 Driver is used to capture data from RPI-0 camera Device.
- /dev/video0 is the RPI-0 camera device to capture data.
- Supported Soc level Gstreamer plugins to capture data from camera device.
- Supported H264 encoding format.
- Supported SD,HD,FUll HD resolution.
- Supported automatic boot-up process for RMS functionality.

Architecture



Design Considerations

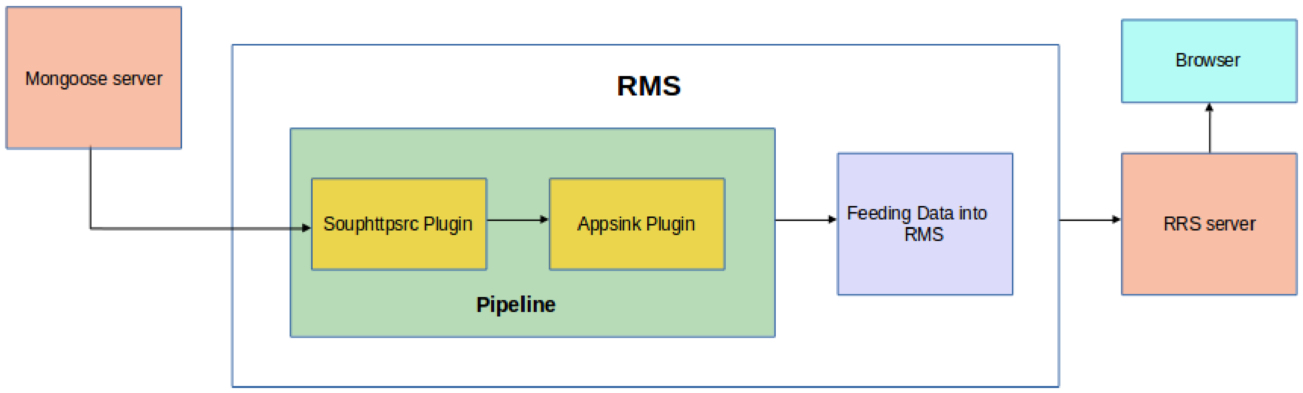
- **Gstreamer Soc Implementation for RPI-0 Camera**



- Enabled V4l2 driver in part of RPI-0 to capture data from /dev/video0 device.
- Implemented soc level gstreamer pipeline with v4l2src plugin,omxh264enc plugin and appsink plugin.

- V4l2src plugin is used to capture raw data from camera through v4l2 driver and transmitted captured raw data into omxh264enc plugin to encode raw data into h264 encoding format.After that transmitted encoded data into appsink plugin to wrote encoded data into 8080 port of mongoose server.
- Registered 8080 port in Mongoose server to listen data.
- Supported resolution SD(640*480 , 720*576) ,HD(1280*720) and Full HD(1920*1080)

- **Gstreamer Implementation in RMS**



- Implemented RMS side gstreamer pipeline with souphttpsrc plugin and appsink plugin.
- Souphttpsrc plugin is getting the encoded data from Mongoose server from 8080 port and transmitted encoded data into appsink plugin to feed the data into RMS.
- Supported ASCLI interface to configure pull stream property and to start webrtc to transmit data into RRS server through WiFi.
- Browser getting the data from RRS server and it display the received content.