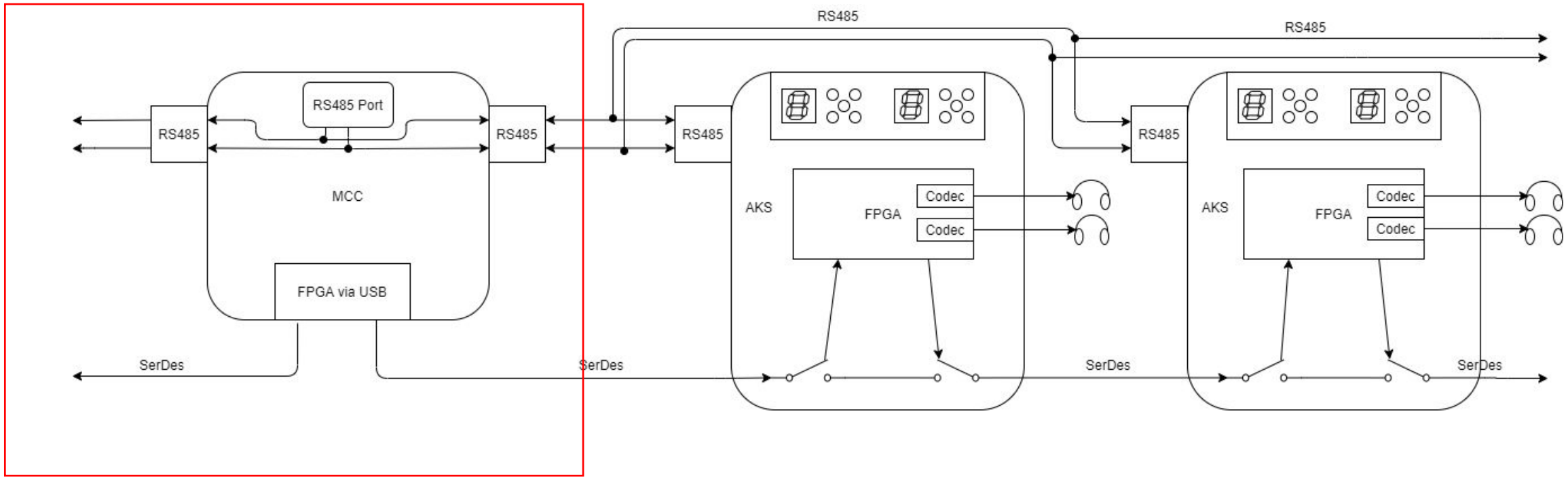
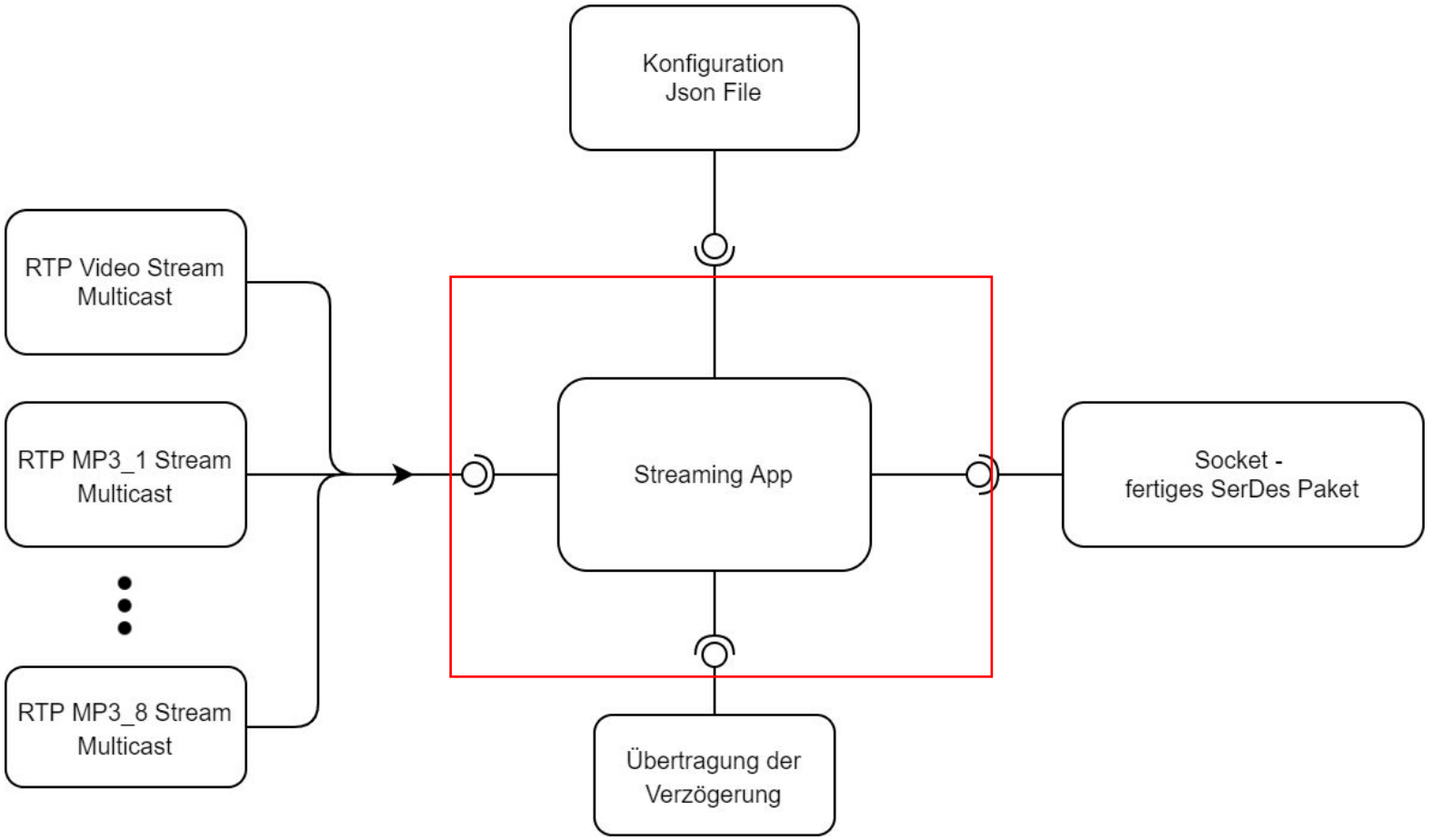
MCC Streaming App

Systemübersicht

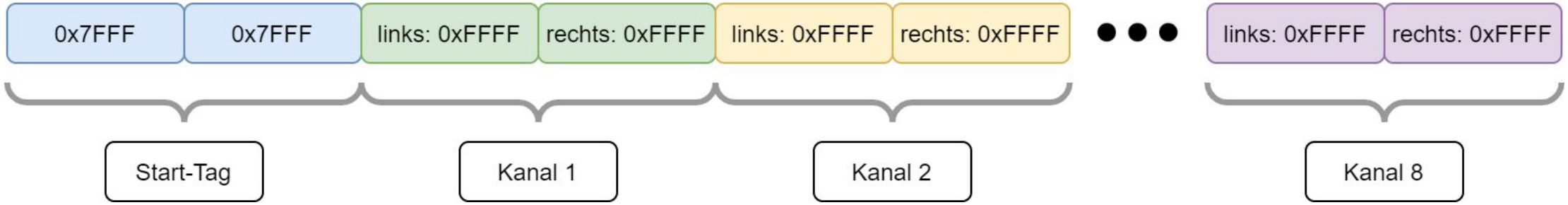


Übersicht Schnittstellen der App

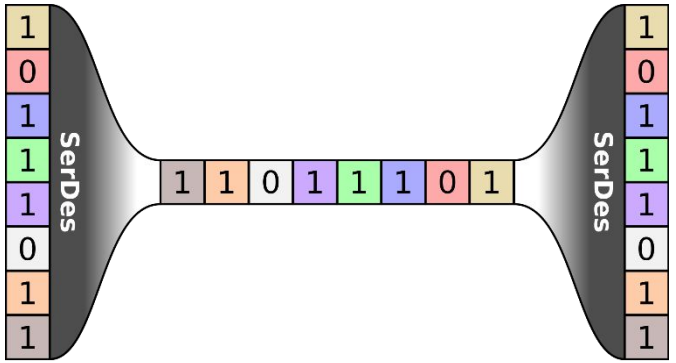


- Schnittstelle für die Streams
  - Es sollen bis zu 8 Streams parallel verarbeitet werden
  - dafür gibt es 8 Kanäle
  - 1 Video-Streams mit 2 Audiostreams
  - 6 Audio-Streams (MP3)
  - Übertragung/Empfang via RTP oder als lokale Datei
  - die Quelle für die Kanäle kann jederzeit geändert werden
- Ausgabe der bearbeiteten Daten (SerDes Paket)
  - die Daten sollen an eine definierte Senken ausgegeben werden
  - die Auswahl beschränkt sich auf einen Socket oder die Konsole
  - eventuell sollen die Daten auch direkt an den USB-Treiber übergeben werden, um Verzögerungen zu vermeiden, dies muss noch geklärt werden,

Framing und Format der Daten



- das SerDes Paket
  - jedes Paket hat eine Größe von 36 Byte (8 x 4Byte für die Kanäle, 4 Byte für die Startsequenz)
  - die Größe setzt sich zusammen aus den 8 Kanälen und der Startsequenz
  - die Startsequenz ist klar definiert: 0x7FFF 0x7FFF (4Byte)
  - der Wert 0x7FFF darf ansonsten in dem Paket nicht vorkommen
  - Sollte der Wert trotzdem berechnet werden so, ist er durch 0x7FFE zu ersetzen
  - Jeder Kanal hat 4 Byte zur Verfügung → 2 Byte für den linken Audiokanal und 2 Byte für den rechten Audiokanal
- Datenformat
  - **Sample Rate**: 44,1 kHz
  - **Sample Größe**: 16 Bit
  - **Sample Typ**: Signed Int
  - **Byte-Anordnung**: little Endian
  - **Codec**: raw audio
  - **Kanal Anzahl**: 2 (links & rechts)

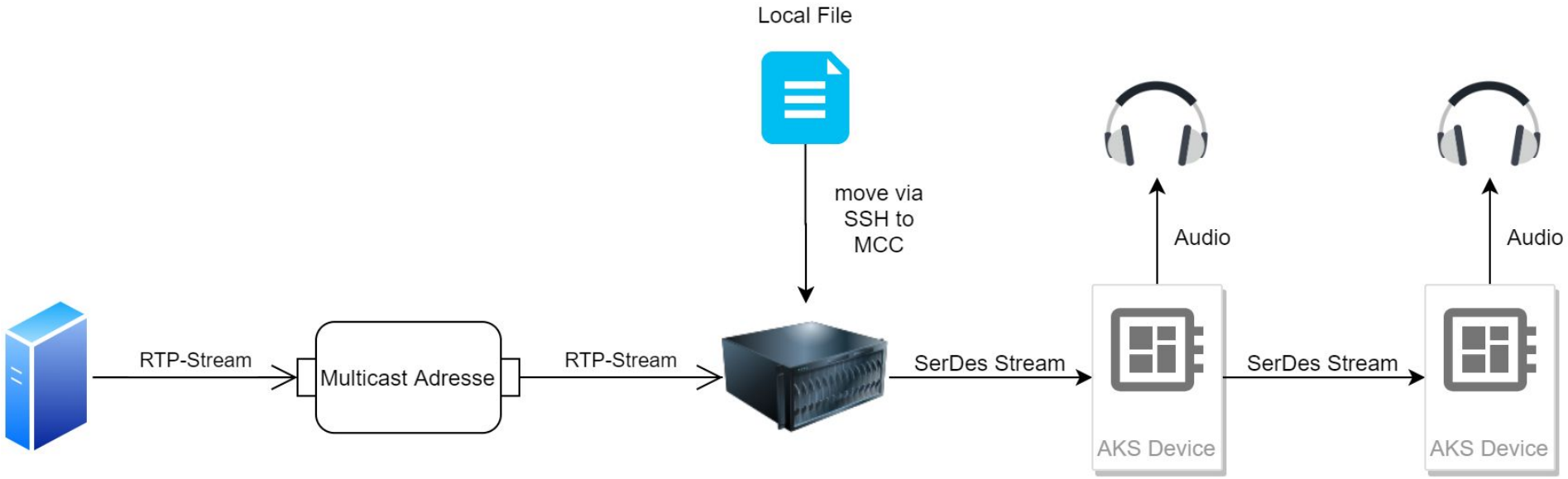


Aufgabe

- Entwicklung nur der Streaming App für Linux (Schnittstelle)
  - Keine Diagnose, kein Update der AKS-Module
- Die Ausgabe über die AKS ist schon entwickelt.

- Vorraussetzungen
- Entwicklungsumgebung / Sprache
  - c++ / gcc (Debian default)
  - Qt 5.11.3 (Debian default)
  - Linux 64 Bit
  - Standardpakete aus Debian 10 - 64 Bit
  - MCC + 3 x AKS + 3 x Kopfhörer + Kabel

Testszenario



- Wie kann getestet werden?
- Man erstellt einen RTP Server und streamt eine Datei
  - z.B. eine Raspberry Pi mit VLC Player → über den VLC Player kann dann ein RTP-Strteam erstellt werden
- Es wird eine Datei lokal auf das MCC kopiert via ssh
  - Die Datei wird dann geöffnet, gelesen und per SerDes übertragen
  - der Ablageort kann über die Konfiguration eingestellt werden und ist daher beliebig
  - Anmeldedaten für ssh auf dem Gerät werden noch definiert
- Schritte 1 & 2 können jeweils mit einer oder mehreren Audiodateien und einer Videodatei ausgeführt werden
- Die Audiodaten werden auf dem AKS auf dem entsprechenden Kanal ausgegeben

- Wie kann getestet werden?
- Man erstellt einen RTP Server und streamt eine Datei
  - z.B. eine Raspberry Pi mit VLC Player → über den VLC Player kann dann ein RTP-Stream erstellt werden
- Es wird eine Datei lokal auf das MCC kopiert via ssh
  - Die Datei wird dann geöffnet, gelesen und per SerDes übertragen
  - der Ablageort kann über die Konfiguration eingestellt werden und ist daher beliebig
  - Anmeldedaten für ssh auf dem Gerät werden noch definiert
- Schritte 1 & 2 können jeweils mit einer oder mehreren Audiodateien und einer Videodatei ausgeführt werden
- Die Audiodaten werden auf dem AKS auf dem entsprechenden Kanal ausgegeben