Architecture Independant Audio Management
Standalone SoC

- Main Display
- Cluster
- Camera
- Mic
- Amp
Proposed Targets

- **SoC / AGL as the Master**
  Central System Setup
  Central Resource Management
  Central Device Control
  Accessability of streams

- **Abstraction of ‘Networks and Boxes’**
  Application API independent from used networks
  No proprietary Box knowledge visible at application layer
  Transparent support for any Command protocol
  Based on standard Linux interfaces (ALSA, V4L2, TCP/IP, cdev)

- **Not limited to Audio!**
Desired System View
Modular System Design: Separation of Functionality
Audio/Video Management

- **SYSTEM SETUP Domain**
  - Setup Routing and provide bandwidth
  - Connects Sources and Sinks
  - Can be static or dynamic
  - Needs to be deterministic at ANY time

- **DEVICE CONTROL Domain**
  - Abstract Command-API
  - -> Stream start/stop/pause etc.
  - -> Volume up/down/ramp/xover etc.

- **STREAM ACCESS Domain**
  - Linux: ALSA, V4L2, TCP/IP
  - Embedded: I²S, TSI, SPI, MediaLB
Architectural Recommendations
Routing Manager

- Central entity of the SYSTEM SETUP domain

- Organizes connections/bandwidth abstractly
  Supports all datatypes (not only audio)
Routing Manager (cont.)

- Single XML for fixed connections, rules and priorities
  -> OEM specification (‘the Bible‘)
  -> System integrator creates XML and validates against
     Avoiding ‘Fulfillment by Collective‘ (multiple Tier1s)

- API for requesting dynamic connections
  -> Deterministic rules of the XML apply

- Ensures system behavior at ANY time:
  -> At initial release
  -> After updates
  -> Future additions
  -> Supplier change
Organizing bandwidth of a network system can be compared with managing memory.

The MMU (Memory Management Unit) is the central instance. Applications in need of memory request this by the MMU API.

Direct access is a bad thing ;-)
Audio Manager

- (One) entity of the DEVICE CONTROL domain

- Should manage the Audio, not the proprietary tech
  - Agnostic towards underlying networks
  - Agnostic towards underlying protocols
  - Agnostic towards underlying boxes

- Must not fiddle with the system routing directly
  - Using fixed connections build up at startup
  - Using Routing Manager API for dynamic needs
Video/Media Manager

- **Video Manager**
  (Another) entity of the DEVICE CONTROL domain

  Similar to the Audio Manager, but for video streams

- **Media Manager?**
  Combining Audio Manager + Video Manager

  Support of other standard Linux interfaces, not only ALSA
  Support of compressed datastreams (AC3, DTS, TS, etc.)