Practical Experiences from Using Pulseaudio in Embedded Handheld Device

Jyri Sarha

18.9.2009

Background Information

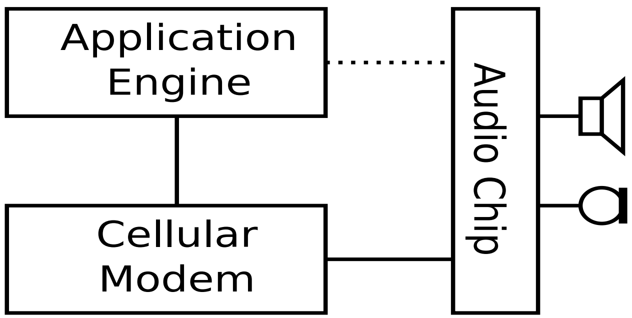
- Name: Jyri Sarha
- Education: Master of Computer Science, Helsinki University of Technology 2000
- Employer: Nokia Devices / Maemo
- Responsibility: Audio Subsystem Architecture and Development

Audio Features N900 Linux Phone

- Runs **Pulseaudio**
- Primary audio API is libpulse0
  - Used mostly through gstreamer and libcanberra
- Pulseuadio applies transducer specific audio processing and speech pre- and post-processing
- Implements integrated VoIP and Cellular call functionality
  - All audio is played through Pulseaudio
  - IOW the phone has *APE Centric Audio Architecture*

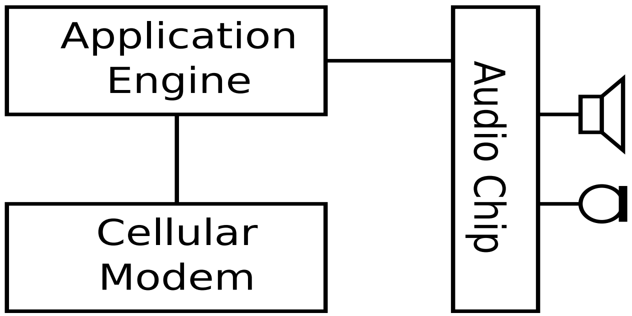
Typical Smart Phone Audio Architecture

- Cellular modem has direct connection to audio HW
  - Cellular call is power efficient and easy to implement
- *Application Engine* routes audio through cellular modem
  - Cellular modem wastes power in music playback use case
- Add alternative audio route from APE to audio HW
  - HW design is complex and expensive
  - Audio routing, mixing and processing becomes complicated



What is APE Centric Audio Architecture

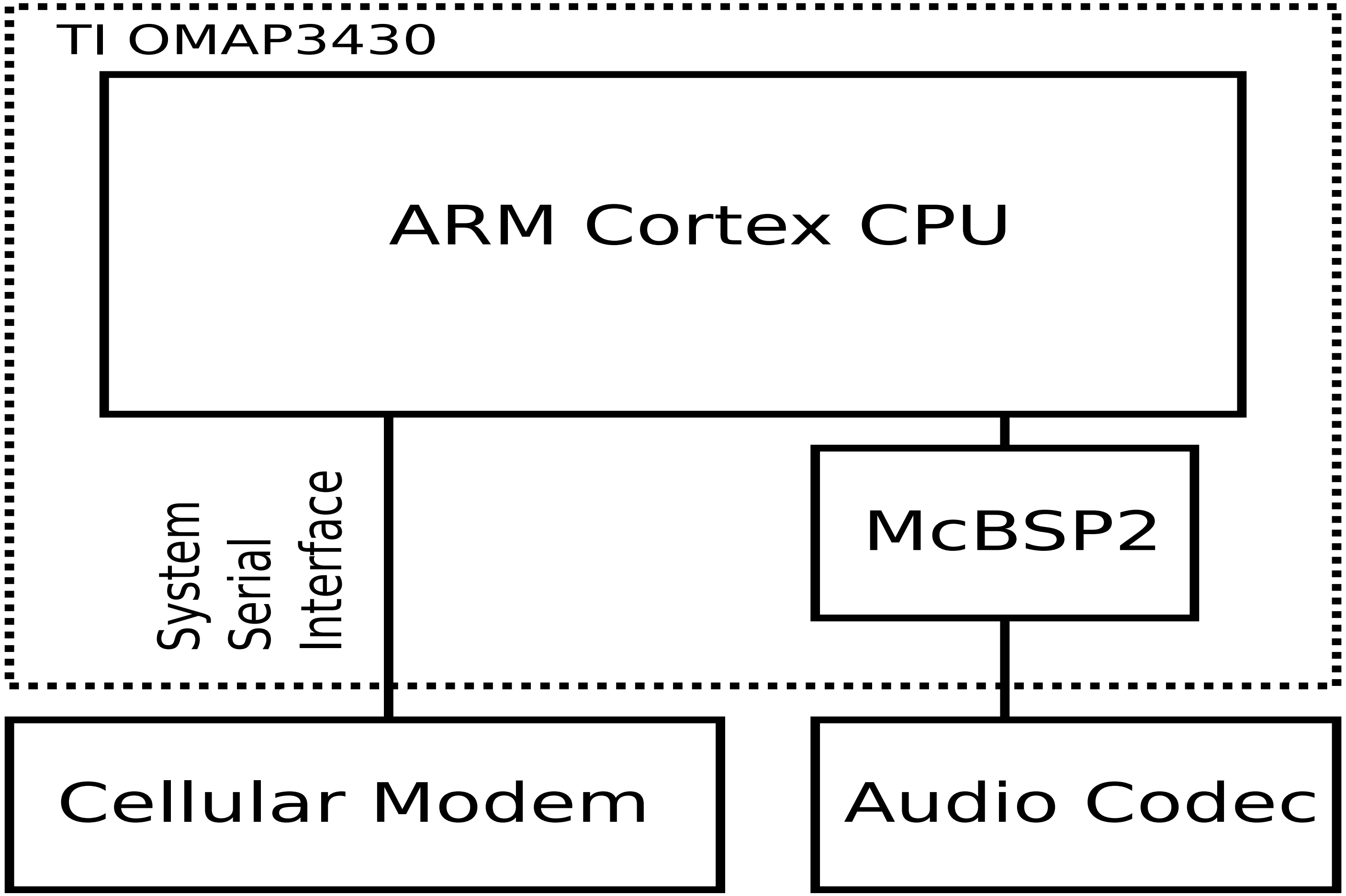
- Audio HW connected directly Application Engine
- Cellular call audio is routed via APE like any other application
- Cellular modem becomes more like data modem



Advantages & Challenges of APE Centric AA

- Advantages
  - All audio routing and mixing can be done in APE
  - Similar architecture to PC or laptop
  - VoIP and CS-call can share same audio processing pipeline
  - Conclusion: Simplifies SW
  - Fewer functional requirements for Cellular modem
  - Less wiring on HW layout
  - Conclusion: Simplifies HW
- Challenges
  - Cellular call latency
  - CS-call audio processing consumes APE CPU
  - Both Cellular Modem and APE cosume power during call

N900 Audio HW



Simplified picture of N900 audio HW

Implementation Challenges

- Minimize cellular call latency increase caused by the architecture
- Audio pipeline optimization to minimize power and CPU consumption
- Accurate feedback loop timing for Acoustic Echo Cancellation

Cellular Call Latency Challenge

- Cellular modem air interface alone causes a lot of latency
- GSM and 3G audio frame size is 20 ms
- Cellular frame timing is ruled by base station
- Timing of 20ms frames change in cellular hand over
- Simple buffering between cellular modem and audio codec adds at least 20 ms latency to each direction

Minimize Cellular Call Latency

- Run ALSA with two 5 ms fragments
  - This is doable even on ARM Linux with real-time priority
  - DMA buffering delay is 5ms each direction
- Synchronize up link audio buffering with Cellular Modem
  - Cellular Modem sends up-link timing adjustment messages
  - Align up-link buffering according to messages
  - Change UL timing with 5 ms granularity
- Synchronize down link audio buffering with Cellular Modem
  - Keep tight buffer management to minimize latency

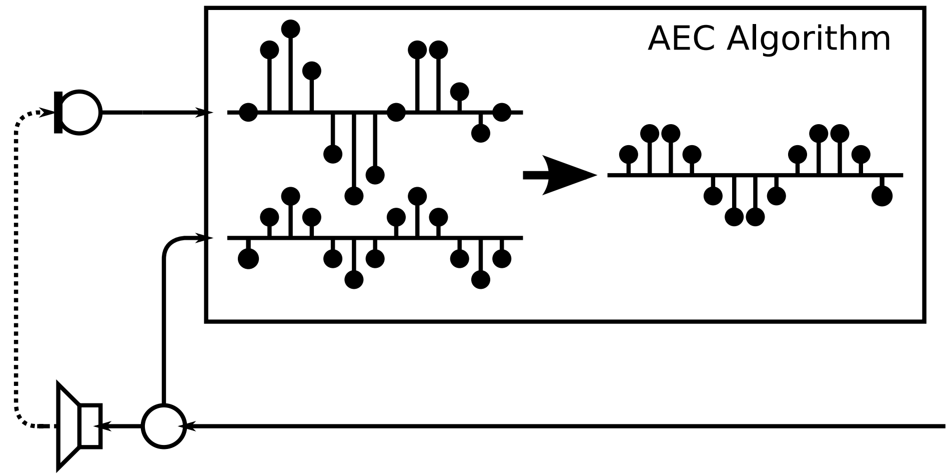
Use Time Optimization

- Do efficient power management
- Optimize CPU usage
  - Use NEON vectorization when applicable
  - Optimize all audio processing for APE CPU
  - Including: Speex SRC and Nokia proprietary algorithms

Efficient Power Management

- Turn all possible power domains off as often as possible
- Take advantage of McBSP2 (Multi channel Buffered Serial Port)
  - Do block transfers to McBSP2 1280 word buffer
  - Power whole CPU down between blocks including DMA
  - Use bigger fragments when not in CS-call

Acoustic Echo Cancellation the Basic Idea

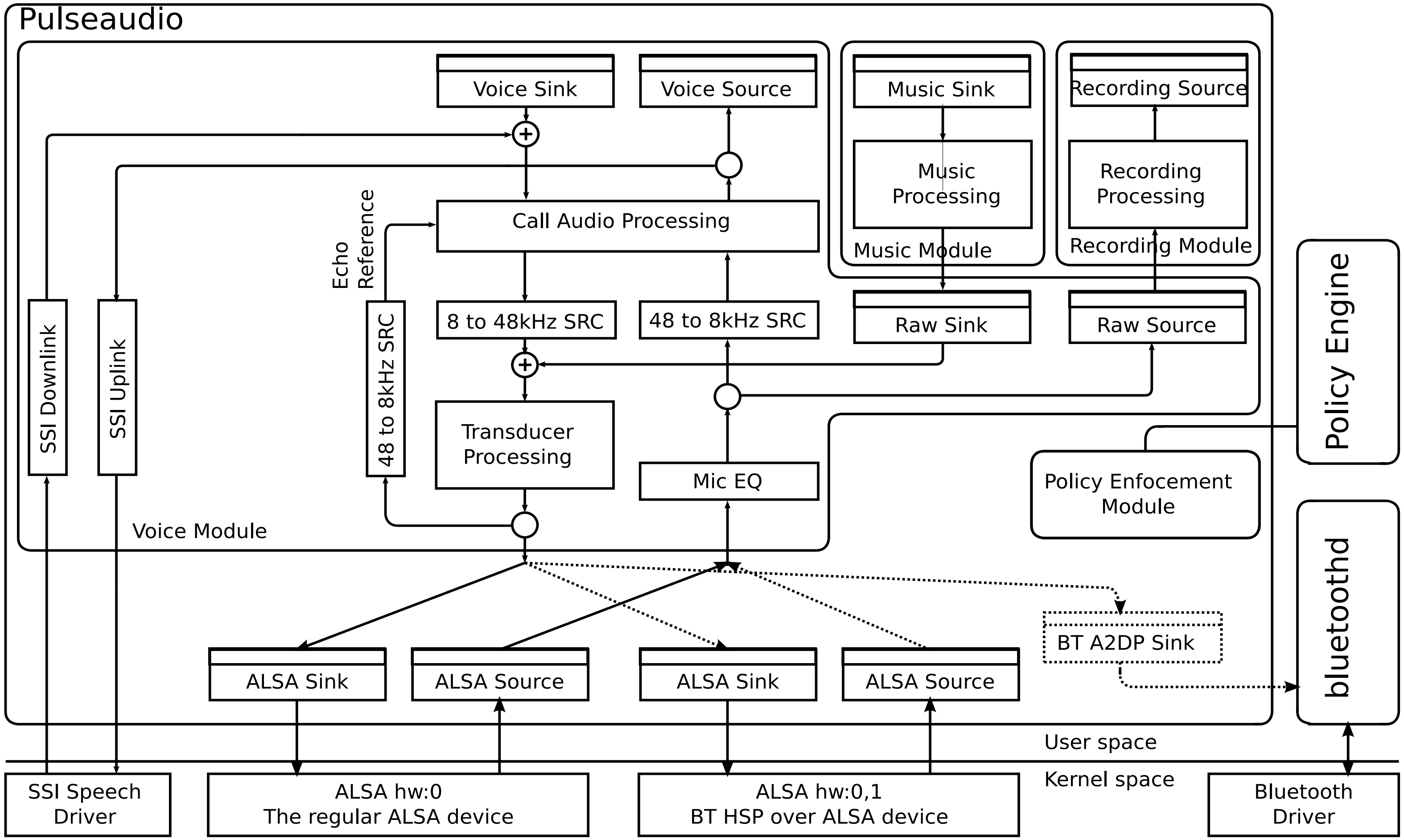


- Correlate echo reference from the mic input to find alignment
- Filter echo reference out from the mic input
- Good acoustic echo path modeling for transient signals is possible only with proper time alignment of the reference loop

Accurate feedback loop timing for AEC

- Accurate playback and capture latencies are needed for time alignment of echo reference and mic signal
- Latency functions of ALSA do not work well with DMA doing block transfers
- Solution: Implement ALSA sink latency functions based on snd_pcm_htimestamp()

N900 Pulseaudio Configuration



Thanks!

- to Lennart Poettering for the invitation
- to Kai Vehmanen, Kalervo Kontola and Seppo Ingalsuo for ideas and support
- to Marc-André Lureau for encouragement to come