Integrating the driver experience

VIRTIO audio
Genivi AMM Munich 2019
Virtual Audio basics

- Allow virtual machines to access audio streams
  - playback stream for sound reproduction
  - capture stream for sound capturing

- Challenges
  - Don’t stop
  - Don’t interrupt
  - Low latency
  - Short stream startup time
  - Enable hardware offloading
Audio Buffer basics

Visualization of stereo (2-channels) audio playback
Audio Buffer basics

Playback:
- Period
- Hardware buffer
- SW
- Audio frames provided for playback
- HW

Capture:
- SW
- Audio frames captured by hardware
- HW

# of periods = 4

free space
Audio Buffer basics

Visualization of in memory layout for stereo (2-channels) audio frames

Interleaved mode:

```
L_1 R_1 L_2 R_2 L_3 R_3 L_4 R_4
```

Non-interleaved mode:

```
L_1 L_2 L_3 L_4 R_1 R_2 R_3 R_4
```
Virtualized device Architecture with VIRTIO

One queue for control (channel discovery, capabilities, etc)
One queue per channel, input and output are interleaved
- One channel is the set of coherent streams
- If multiple incoherent streams are needed, add more devices

VQ=virt-queue
SG=Scatter Gather
- Implements alsa driver model
  - Highly flexible
  - Dynamic probing to make hardware capabilities available in the guest
Outlook

- Spec currently on virtio-devel mailingslist
  - Discussions still ongoing
- PoC is running
  - With COQOS hypervisor running Linux on rcar
  - QEMU-ARM running Android
  - QEMU-KVM running Linux
  - QEMU-KVM running Windows 10
- Linux kernel driver RFC patchset will be posted shortly
- QEMU reference implementation to follow