Beyond complex cameras: Complex video graphs using PipeWire

George Kiagiadakis
Principal Software Engineer
Who am I?

- Principal multimedia engineer @ Collabora
- GStreamer, PipeWire, ...
- WirePlumber maintainer
Cameras
Cameras ...

- Viewfinder
- Data display
- Lens Elements
- Shutter release
- Electronic sensor
- Aperture
Cameras ...

Sensor → A-D converter → DSP → Compression → CPU
Complex cameras ...
Complex cameras ...

Diagram showing the process of processing images from sensors through A-D converters and DSPs to generate high-resolution still images, still image preview streams, and video streams.
Complex cameras ...

- Very different images captured from different sensors
- High computational requirements for combining
- Camera MCU, ISP & host CPU sharing the workload
  - Blurry boundary
  - Now also AI accelerators
- Images going back and forth between processing "blocks"
- Need software to manage the processing pipeline
libcamera

- Manages devices, sensors
- Manages processing pipeline
- Runs (proprietary) algorithms, sandboxed
- Device-agnostic / Device-specific components
- Abstract API for userspace
Multiple sensors → Multiple devices

• How to combine data from separate devices?
• Can we separate processing between userspace processes and secure them in separate containers?
• As complexity increases, we need
  – Separation of responsibility: “Divide and conquer”
  – Versatility
PipeWire

- Multimedia bus
- Multi-process MM graphs
- Resource sharing
- Low latency
- Low resource consumption
- External management component: WirePlumber
Applications

• Automotive (cameras, sensors, AI processing, ...)
• Cloud processing
• Mobile multimedia
• Many many more ...
Let’s not forget audio

- PipeWire: default audio daemon on Linux desktop
  - Replaces PulseAudio & JACK
  - Also on Steam Deck & other devices
- State of the art Bluetooth audio infrastructure
- Complex audio graphs made possible
Best described with a demo ...
Next steps: your call!
Thank you!
We are hiring
col.la/careers