PipeWire: The New Multimedia Service, Now Ready for Automotive

Julian Bouzas
Software Engineer

Open First
Hi, I’m Julian

• Multimedia Team at Collabora since 2019

• PipeWire and Wireplumber Developer

• julian.bouzas@collabora.com
What is PipeWire?
Fresh multimedia service for Linux

• Originally meant for video only: PulseAudio for Video (PulseVideo)
• Now generic multimedia service for both Video and Audio
• Video Capture:
  - Cameras
  - Graphic Sources (Wayland, Vulkan, OpenGL...)
• Audio Playback and Capture:
  - Microphone and Speakers
  - Bluetooth devices
Why do we need PipeWire?

- Unifies both PulseAudio and Jack audio servers, greatly simplifying the Linux Multimedia Stack
- Permissions: Supports containers like Flatpak and does not rely on *video* and *audio* user groups
- Low latency: can handle very small buffers sizes (e.g. 32 samples)
- Flexible: external Session Manager adaptable to any use cases
Compatibility APIs on top of PipeWire

- ALSA applications
  - PipeWire PCM plugin

- PulseAudio applications
  - Replacement for libpulse.so and libpulse-mainloop-glib.so

- JACK applications
  - Replacement for libjack.so
Architecture and Design

- Modular with Plugins
- Graph based like GStreamer: Nodes, Ports and Links
- Multi-Process:
  - Daemon processes most of the data (nodes can also run in the clients to avoid stalling)
  - External session manager configures and links the nodes
- Fully based on its internal and Simple Plugin API library (SPA)
  - Extremely simple and lightweight generic purpose multimedia library
  - Mostly header-only C library with no other dependencies (glib, GStreamer, etc...)
Performance and Efficiency

- Static code design approach
  - Almost no mallocs

- Uses modern Linux APIs
  - memfd & dmabuf to zero-copy device buffers
  - eventfd & timerfd for scheduling

- Low CPU Usage and low-latency real-time capable
CPU Usage against PulseAudio

- Hardware: Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz
- Source: PipeWire
- PipeWire handles buffer sizes of 32 samples, PulseAudio underruns
CPU Usage against JACK

- Hardware: Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz
- Source: PipeWire
- PipeWire handles buffer sizes of 32 samples, JACK underruns
Security

- External Session Manager grants permissions to applications

- Nodes can be only visible for some applications

- Type of Permissions:
  - (R) Read: Visible, Capture data
  - (W) Write: Play data
  - (X) Execute: allow executing methods on objects (e.g. Setup format on nodes)
External Session Manager

- Not included in the PipeWire project

- Creates and Configures Devices to emit new Nodes

- Sets up Nodes (Format, Ports, etc...)

- Creates links based on its policy logic when a client connects

- Grants security and access control to clients (applications)

- Launched by the PipeWire daemon at startup
Current Status

- Version 0.3.5 released in May 2020 and distributed in Fedora 32
  - Plenty of JACK applications already working with PipeWire
  - Many PulseAudio apps work as well
  - Bluetooth starting to be fully supported, needs more testing
  - Musical Instrument Digital Interface (MIDI) works
  - Plans to replace PulseAudio soon
  - Video capture from V4L2 devices works well
  - Wayland screencasting from weston, gnome-shell, and wlroots supported

- Adopted by AGL (Automotive Grade Linux) as the core audio framework
Who started this

• Author: Wim Taymans
  - Well-known old GStreamer developer & ex-maintainer
  - Sponsored by: Red Hat

• Embraced by PulseAudio developers
  - Seen as the next generation of PulseAudio

• Welcomed by ALSA and JACK developers

• License: MIT
PipeWire DOT tool

- Generates DOT graphs
- Usage:
  - $ pw-dot pw.dot [options]
- Options:
  - --all (shows all object types: nodes, devices, ports, clients, etc...)
  - --smart (only shows linked objects)
  - --details (show all object properties)
PipeWire in the Automotive Industry
Why PipeWire suits Automotive

- **Problem:** Device handling in connected cars is complex:
  - Plenty of speakers and cameras
  - Multiple audio streams: Radio, Emergency, Navigation, Communication...
  - Phone calls over Bluetooth

- **Solution:** PipeWire with a flexible external Session Manager
  - Custom policy logic
  - Custom hardware pipelines
  - Hardware control abstraction
  - Security
WirePlumber

- First external session manager implementation for PipeWire
- Originally planned for Embedded only (Automotive)
- Now generic and fully featured Session Manager for both Embedded and Desktop
- Based on GObject to support writing bindings in other languages: Rust, Python, LUA...
Introduction of new Objects

- **Endpoint**: object that handles PipeWire nodes
  - Audio Software DSP Endpoint
  - Simple Node Endpoint

- **Stream**: connection points of an Endpoint

- **Session**: set of Endpoints
  - Video Session
  - Audio Session
Software DSP example

PipeWire
Software DSP example
Hardware DSP Example

PipeWire
Hardware DSP example
WirePlumber Design

- Provides an libwireplumber API that makes easy writing WirePlumber modules and even other session managers
- Modular design
WirePlumber Modules

- **Monitor**
  - monitors devices and creates nodes when enabled

- **ClientPermissions**
  - grants permissions to clients when connected

- **ConfigEndpoint**
  - creates different endpoints per nodes based on configuration files

- **ConfigPolicy**
  - links endpoints based on configuration files
Bindings example usages

- Python/Rust session manager to avoid use of low level PipeWire API and objects
- WirePlumber Module that interprets LUA files for quick and easy custom policy logic
WirePlumber Versions

- v0.1.0 (Jul 2019): used in AGL Happy Halibut 8.0.0
- v0.1.2 (Oct 2019): used in AGL Happy Halibut 8.0.2
- v0.2.0 (Dec 2019): used in both AGL Happy Halibut 8.0.5 and AGL Itchy Icefish 9.0.0
- v0.3.0 (June 2020): First version with Desktop Support
Future Release

- Support for Bindings in other languages
- Clean the API and make it stable (almost there)
- Improve Documentation
- More unit tests and examples
Who started this

• Author: George Kiagiadakis
  – Sponsored by: Collabora

• Welcomed by PipeWire developers

• Git repository:
  – https://gitlab.freedesktop.org/PipeWire/wireplumber

• Documentation:
  – https://PipeWire.pages.freedesktop.org/wireplumber

• License: MIT
Showtime
Thank you for watching

• Join us on IRC at #pipewire on Freenode

• https://gitlab.freedesktop.org/pipewire

• julian.bouzas@collabora.com