COLLABORA

WirePlumber 0.5

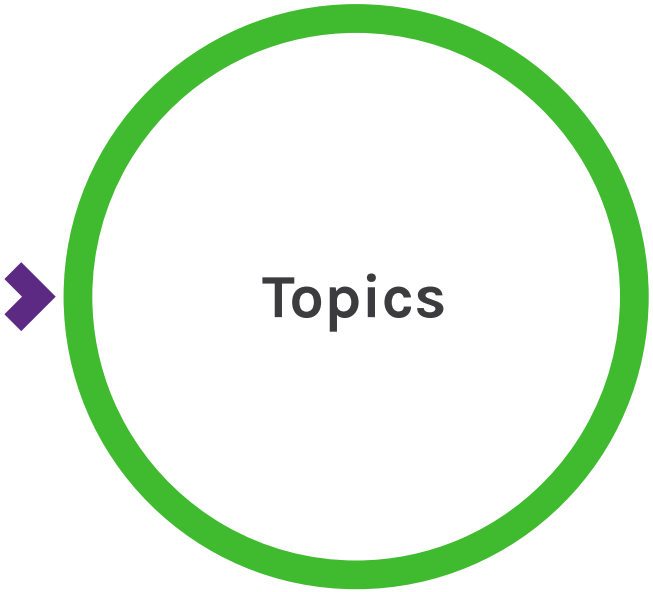
Propelling PipeWire for the Embedded

Ashok Sidipotu
Senior Software Engineer

Open First

Hello!! నమస్తే!! I am Ashok

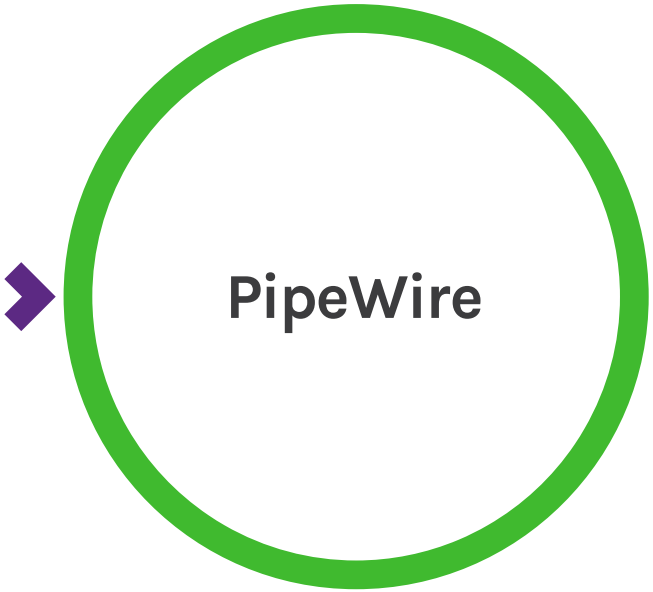
- WirePlumber and PipeWire since 2021
- Worked for over a decade with qualcomm, mostly writing audio software for their chipsets
- Dad of two small children – learning the art of parenting
- Avid Gardner, amateur farmer, birdwatcher etc– love working in tandem with mother nature
- Based out of Hyderabad, India



- Quick Introduction PipeWire
- Quick Introduction WirePlumber
- Wireplumber 0.5: New Configuration System
- Wireplumber 0.5: Event Dispatcher
- WirePlumber 0.5: System Scripts
- WirePlumber 0.5: Miscellaneous things
- WirePlumber 0.5: Status

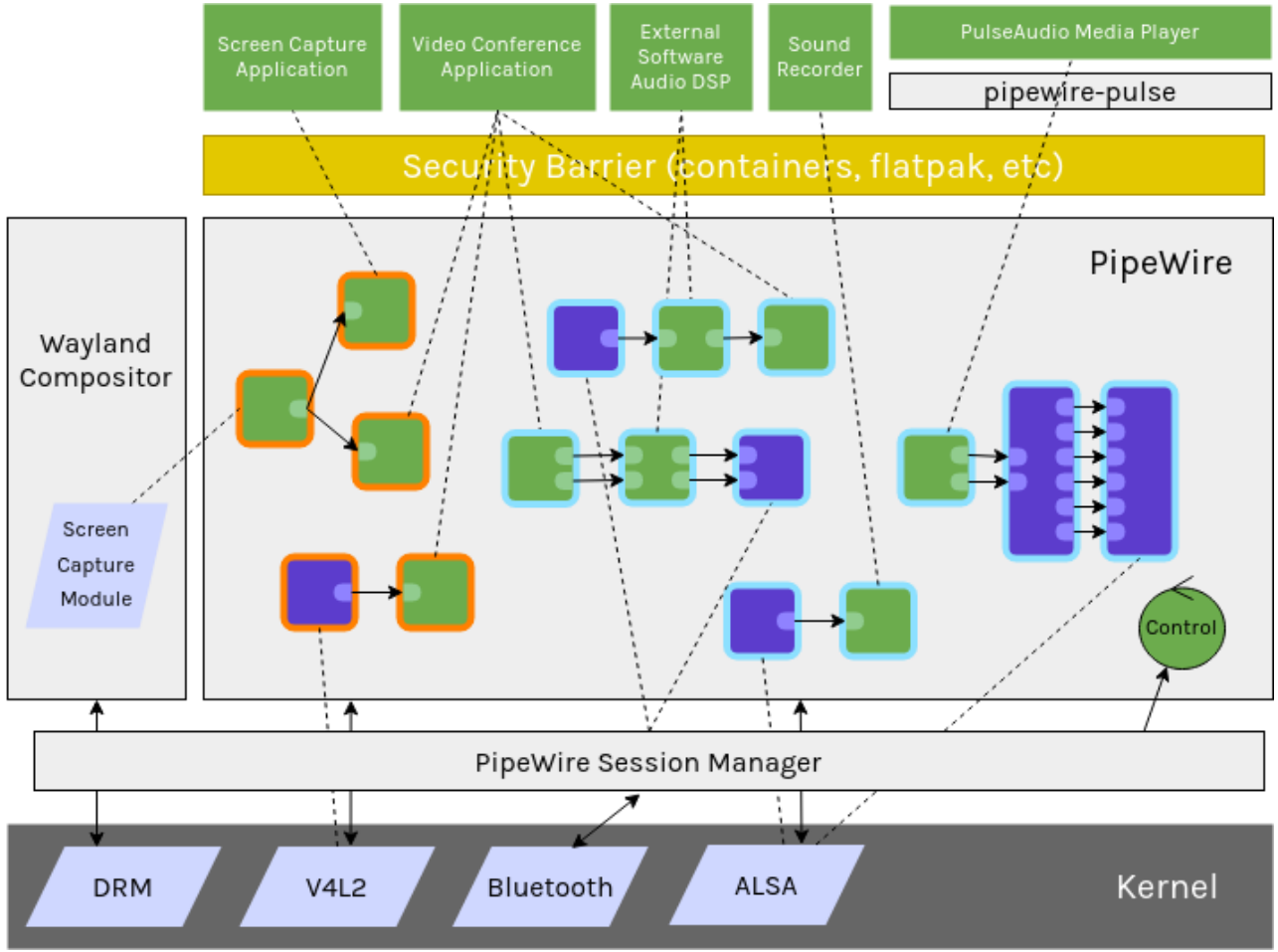
COLLABORA

PipeWire – a quick intro



- Next Generation Audio and Video Server
- Supports Pulse Audio and JACK functionality
- Very Low Latency capable
- Known for its exceptional performance
- Very Modular Design
- Graph based processing
- Built In Security
- Rich set of tools and vibrant community

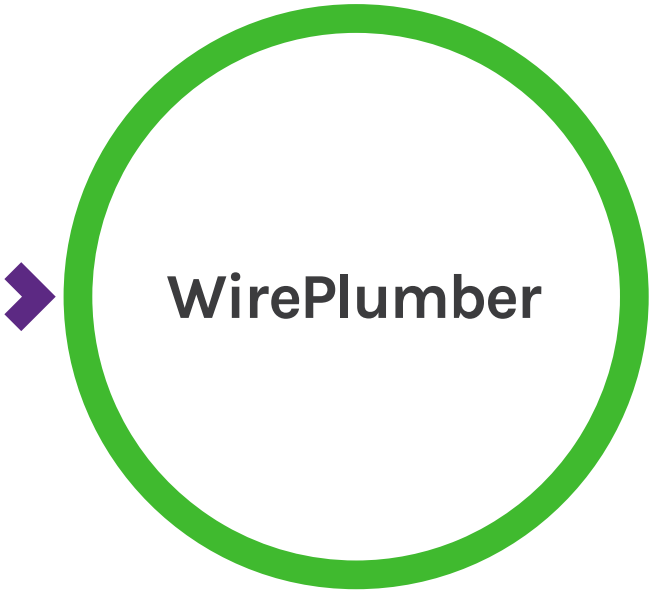
Block Diagram



COLLABORA

WirePlumber – a quick intro

- Manages the control path.
- Enumerates all the PipeWire objects
- Creates a few objects like Devices(Monitors), Metadata etc
- Session Management includes many things.
  - Managing Device profiles and Device hardware routes
  - Managing device preferences
  - Defining access permissions between objects
  - Remembering all the above user preferences across reboots
  - Link objects to form graph
- WirePlumber is the default session manager for Pipewire



- libwireplumber Wraps PipeWire API and provides a higher level and more convenient API
- libwireplumber is GObject based.
- WirePlumber Daemon runs on top of the API and does session management
- Wireplumber Daemon runs Modules and Lua Scripts
- Lua Scripts make it very easy to manipulate the Graph as all the libwireplumber API is available in Lua
- WirePlumber 0.5 is next major release.

COLLABORA

New Configuration System

New Configuration System

- Not a scripting system (two entirely different things)
- Currently, both implemented with Lua
- Moving to SPA JSON as the new configuration system.

Lua drawbacks (for config)

- Configs are Static
- Difficult to override
- Different from PipeWire Config System.
- Clients can’t access the configs

Moving to SPA-JSON

- PipeWire uses SPA-JSON
- Much intuitive to write, thanks to the relaxed parser
- supports split-file configuration with override support

Features

- Configs can be Dynamic
- Easy Overrides
- Easy Extensions
- Configs can be persistent
- Clients can create and access configs at will

Dynamic configs

- JSON is parsed and configs are copied into “sm-settings” metadata
- pw-metadata API can be used to change config
- WP Settings API – fetches the configs from metadata
  - wp_settings_get()
  - wp_settings_subscribe()

```
wireplumber.settings = {
  ## Enables storing/restoring preferences to the file system at startup;
  ## when set to false, default nodes and routes are selected based on
  ## their priorities and any runtime changes do not persist after restart
  device.use-persistent-storage = true

  ## device default volume level
  device.default-volume = 0.064

  ## device default input volume
  device.default-input-volume = 1.0

  ## The persistent save interval in milliseconds when a change happens
  device.save-interval-ms = 1000

  ## Whether to auto-switch to echo cancel sink and source nodes or not
  device.auto-echo-cancel = true

  ## Sets the default echo-cancel-sink node name to automatically switch to
  device.echo-cancel-sink-name = "echo-cancel-sink"

  ## Sets the default echo-cancel-source node name to automatically switch to
  device.echo-cancel-source-name = "echo-cancel-source"
}
```

Static configs

- WP Conf – fetches the configs from JSON
- WP Conf API
  - wp_conf_get()
  - wp_conf_apply_rules()

```
monitor.alsa.properties = {
  ## The properties used when constructing the 'api.alsa.enum.udev' plugin
}

monitor.alsa.rules = [
  {
    matches = [
      {
        device.name = "~alsa_card.*"
      }
    ]
    update-props = {
      api.alsa.use-acp = true
      api.acp.auto-port = false
    }
  }
# This rule example allows changing properties on all ALSA devices.
  {
    matches = [
      {
        ## This matches all cards.
        device.name = "~alsa_card.*"
      }
    ]
    update-props = {
      ## Use ALSA-Card-Profile devices. They use UCM or the profile
      ## configuration to configure the device and mixer settings.
      api.alsa.use-acp = false

      ## Use UCM instead of profile when available. Can be
      ## disabled to skip trying to use the UCM profile.
      api.alsa.use-ucm = false
    }
  }
]
```

COLLABORA

Event Dispatcher

Event Dispatcher: Problem

- WirePlumber cannot
  - Prioritize between PipeWire signals
  - Prioritize between signal handlers
- Problems
  - Race conditions
  - Redundant Processing

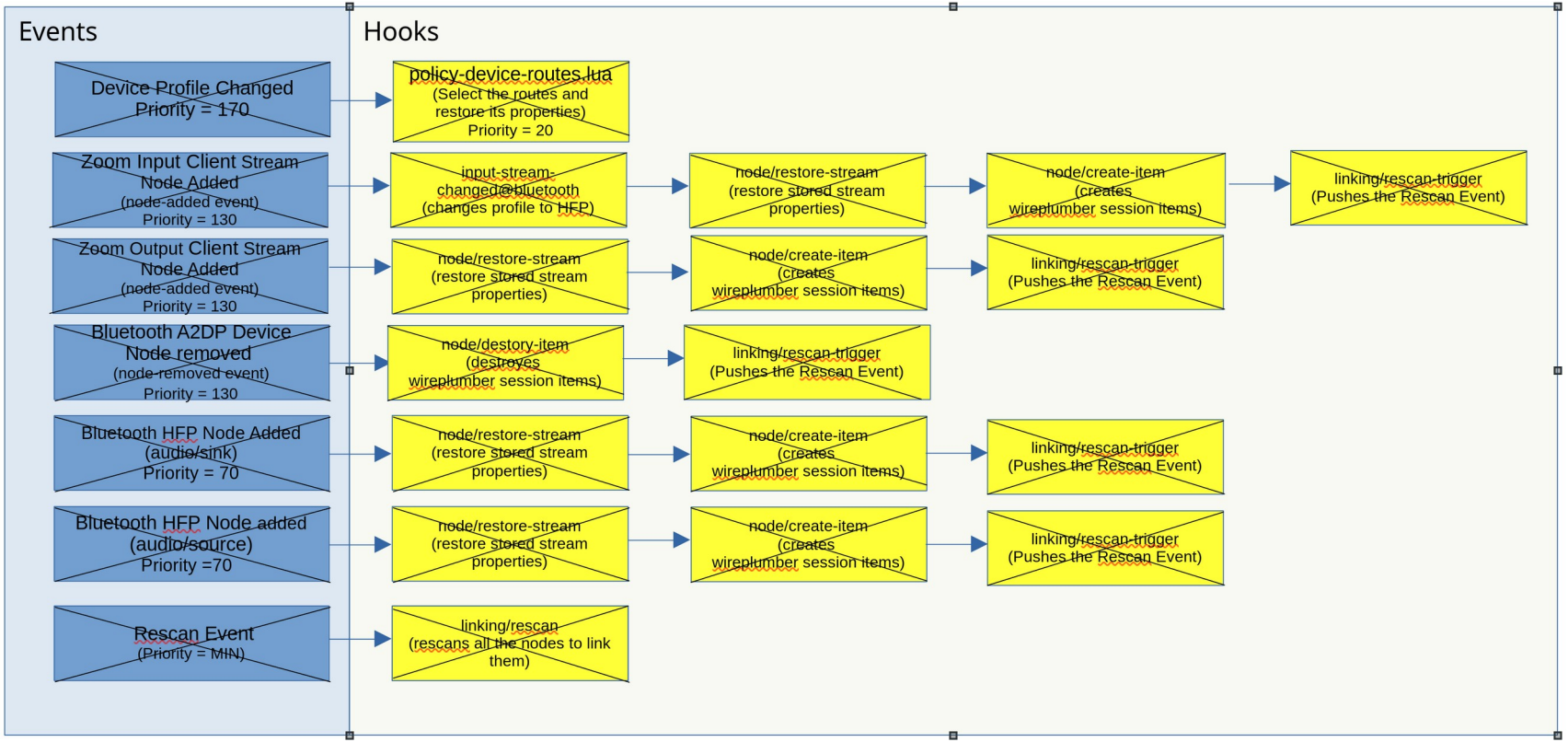
Event Dispatcher: what are Events?

- Convert PipeWire Signals into Events.
- Assign them a priority and push them to a stack.
- Pick and Execute highest priority event, one at a time

Event Dispatcher: what are Hooks?

- Hooks are the new handlers for Events
- Collected for every new Event
- Sorted by priority and executed in order
- Can be Async

Bluetooth auto switch example



COLLABORA

WirePlumber Scripts with Event Dispatcher

System Scripts

- Event Dispatcher transformed Lua Scripts.
- They look and feel completely different.
- They are much more
  - Modular
  - Extensible
  - No Redundant Processing
  - No Hacks

System Scripts: Example

policy-node.lua the main linking script is now a bunch of hooks each with a clearly defined task

Lua Source File	Event	Task
rescan.lua	rescan-for-linking	registers a hook for rescan event, which runs at lowest priority and raises a **select-target** event
find-defined-target.lua	select-target	find target from user preferences
find-default-target.lua	select-target	find target from system level defaults/settings
find-best-target.lua	select-target	find target from available targets
prepare-link.lua	select-target	prepares the selected source and target session items.
link-target.lua	select-target	links the session items.

COLLABORA

Miscellaneous

Miscellanous

- Log Enhancements
- Smart Filters
- Better component loading
- Test Enhancements
- Better libcamera Support (WIP)

WirePlumber 0.5: Status

- All the Features are up and running.
- WirePlumber next is the branch.
- Please try the branch and give us feedback.
- We are likely to release 0.5 some time this year

Thank you!

We are hiring

col.la/careers