# Weston multi-backend

screen recording, streaming, and remote control

Philipp Zabel – p.zabel@pengutronix.de

# About me

- Philipp Zabel
- Pengutronix e.K.
  - Kernel developer
  - Graphics Team
- p.zabel@pengutronix.de
- pH5 on OFTC, LiberaChat,
- and gitlab.freedesktop.org

# Agenda

Weston multi-backend support

- HMI GUI recording and remote control
- Embedded hardware vs desktop
- Wayland / Weston
  + PipeWire for recording,
  + VNC for remote control
- Demo, Weston MRs
- Improvements and alternatives

# HMI GUI recording and remote control

- Different hardware
  - agricultural, industrial, laboratory, or medical equipment
- Common properties: human machine interface
  - Touch screen, GUI – e.g. Qt or Web (Chromium, WPE)
  - Camera(s)
- Similar requirements
  - Camera + GUI recording for documentation, security, training
  - Remote control for user convenience or technical support

# Embedded hardware vs desktop

- Hardware selected for use case
  - Usually not much headroom
- Unified memory architecture
  - CPU, GPU, VPU all use system RAM
- Memory bandwidth limited
  - Raw video expensive, e.g. RGBx: 1080p60 ~0.5 GiB/s, 2160p60 ~2 GiB/s
  - Avoid frame buffer copies
  - Avoid rendering where possible
  - Use hardware features (hardware planes, GPU tiling, FBC)

# Introducing Wayland / Weston

- Wayland is a compositor IPC protocol and library
  - Replacement for X11, libX11/libxcb
  - Reduce roundtrips
  - Shared memory
  - “Every frame is perfect” – synchronized updates, security
  - Extensible via wayland-protocols
- Weston is a Wayland compositor
  - Reference implementation
  - Supports compositor bypass, hardware overlay planes

https://gitlab.freedesktop.org/wayland

# Wayland overview (1/2)

- Protocols (e.g. xdg-shell, linux-dmabuf-unstable-v1)
  - Interfaces
  - Requests and events
  - Global objects
- Wayland core protocol
  - Display, registry, surfaces, buffers, shared memory, input
- xdg-shell protocol
  - Window roles, movement and size

https://wayland.freedesktop.org/docs/html/

# Wayland overview (2/2)

- libwayland
  - Handles marshalling, local proxy objects
  - Listeners
    - Client registers function pointers for events on interfaces
- Surface
  - Attach buffer, commit
- Buffer
  - Shared memory pool, DMA-BUF

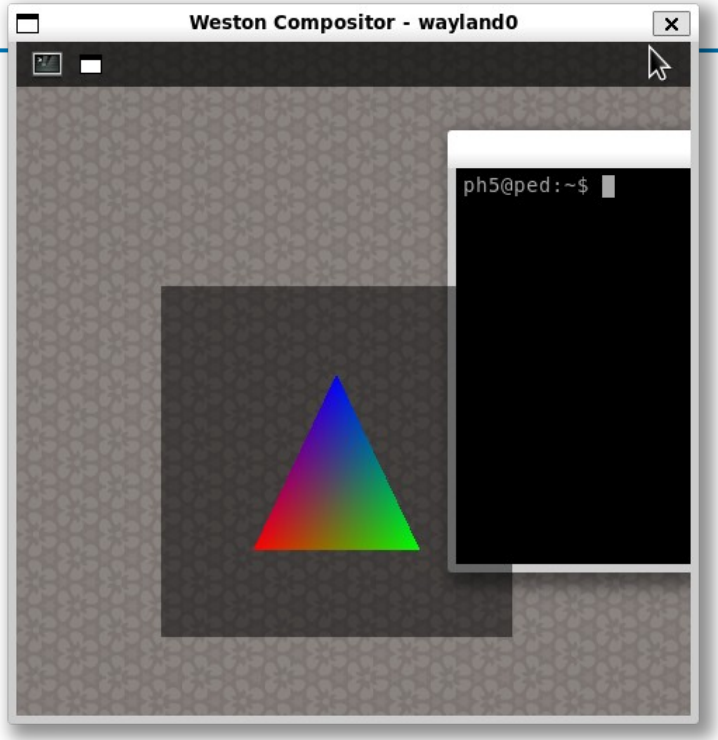
# Wayland client

To show something on the screen, a wayland client ...

- Connects to the Wayland compositor socket
- Binds to (global) objects it wants to use
  - Creates local proxy object used to issue requests
  - Installs listeners (callback functions for events)
- Creates a surface
- Creates a buffer, shared with the compositor (memfd or DMA-BUF)
- Fills the buffer with content and attaches it to the surface
- Commits changes to the surface

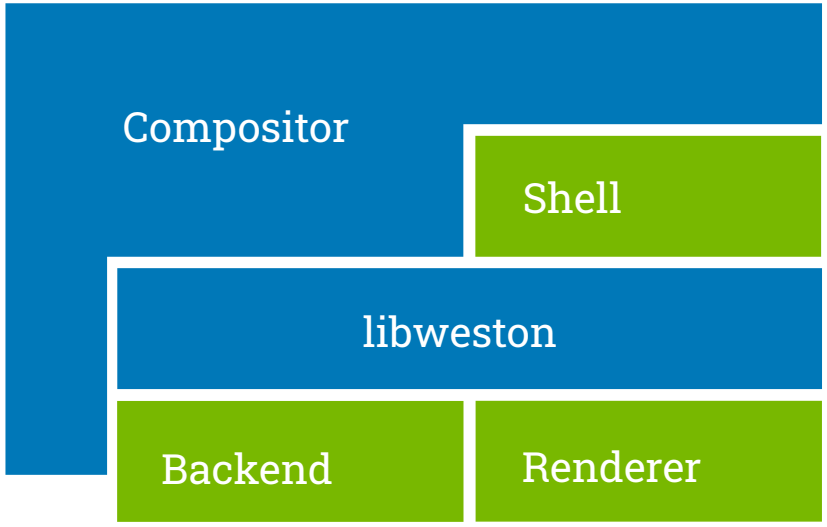
# Why Weston?

- Why Wayland compositor instead of single application?
  - Single point to implement shared functionality
  - Reusable for different use cases
  - Align with desktop development
- Why Weston instead of other Wayland compositors?
  - Hardware overlay planes



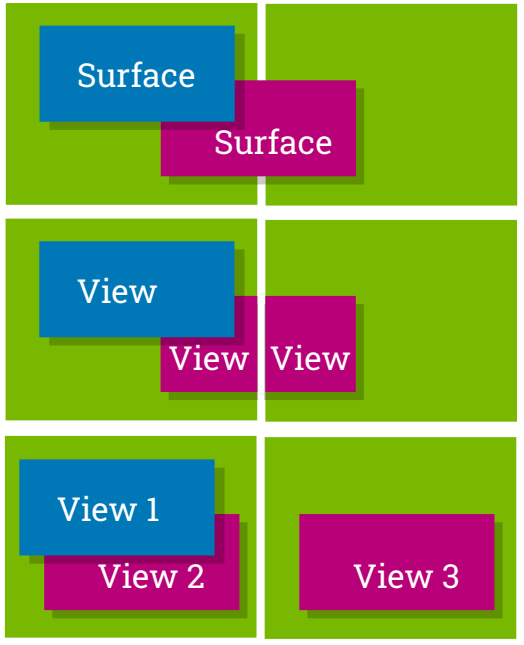
# Weston architecture (1/3)

- Compositor / libweston
- Shell plugin
  - e.g. desktop, kiosk, IVI
  - The window manager part
- Renderer plugin, GL or pixman
- Backend plugin



# Weston architecture (2/3)

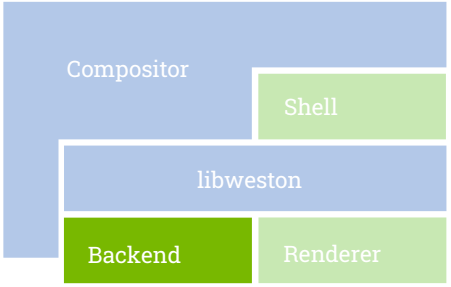
- Shell plugin
  - Decides surface positions on workspace
  - Creates views for surfaces on outputs they are visible on
    - e.g. two halves for a surface straddling two outputs
    - Or: two complete views of the same surface on separate outputs
  - IVI shell allows to write a small controller plugin to control where each surface goes

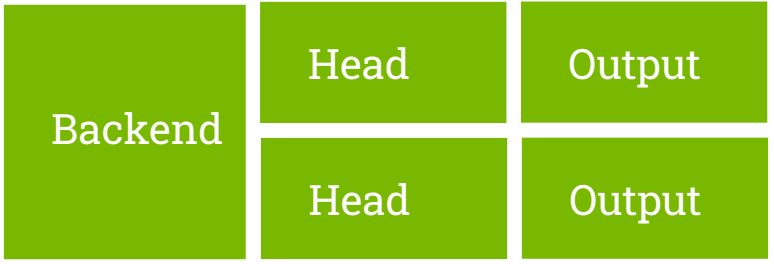


# Weston architecture (3/3)

Backend plugin

- e.g. DRM (standalone), Wayland (nested)
- Implements heads and outputs
- Heads represent monitors (or Wayland windows)
- Outputs
  - Control rendering
  - Control scanout
  - Have position in workspace



# Weston status quo

- Single renderer plugin
- Single backend plugin
- Example:
  - PipeWire plugin using virtual outputs of DRM backend

# Weston + PipeWire for recording*

- Status quo: PipeWire virtual output plugin for DRM backend
  - No DMA-BUF support
  - Maps PipeWire buffers and copies
- New: PipeWire backend (WIP)
  - Can import PipeWire DMA-BUFs into GL renderer FBOs
  - Per-output pixel format
  - Configurable via weston.ini

* and streaming

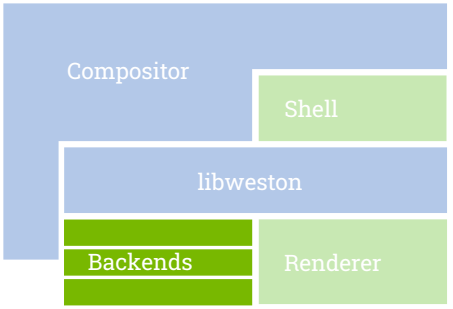
# Weston VNC for remote control

- Status quo: VNC backend (WIP)
  - Initial Weston VNC backend by Stefan Agner[1]
  - Uses Neat VNC by Andri Yngvason[2]
  - Software rendering only
  - No coexistence with DRM backend
- New: VNC backend changes (WIP)
  - Multi-backend support
  - GL render to FBO support with glReadPixels (not optimal)

[1] https://gitlab.freedesktop.org/wayland/weston/-/merge_requests/362
[2] https://github.com/any1/neatvnc

# New Weston backend architecture

- Single renderer plugin
- Multiple backend plugins
- PipeWire backend
  - DMA-BUF support
  - Render into them directly
  - PipeWire stream per output
- VNC backend
  - Support GL renderer

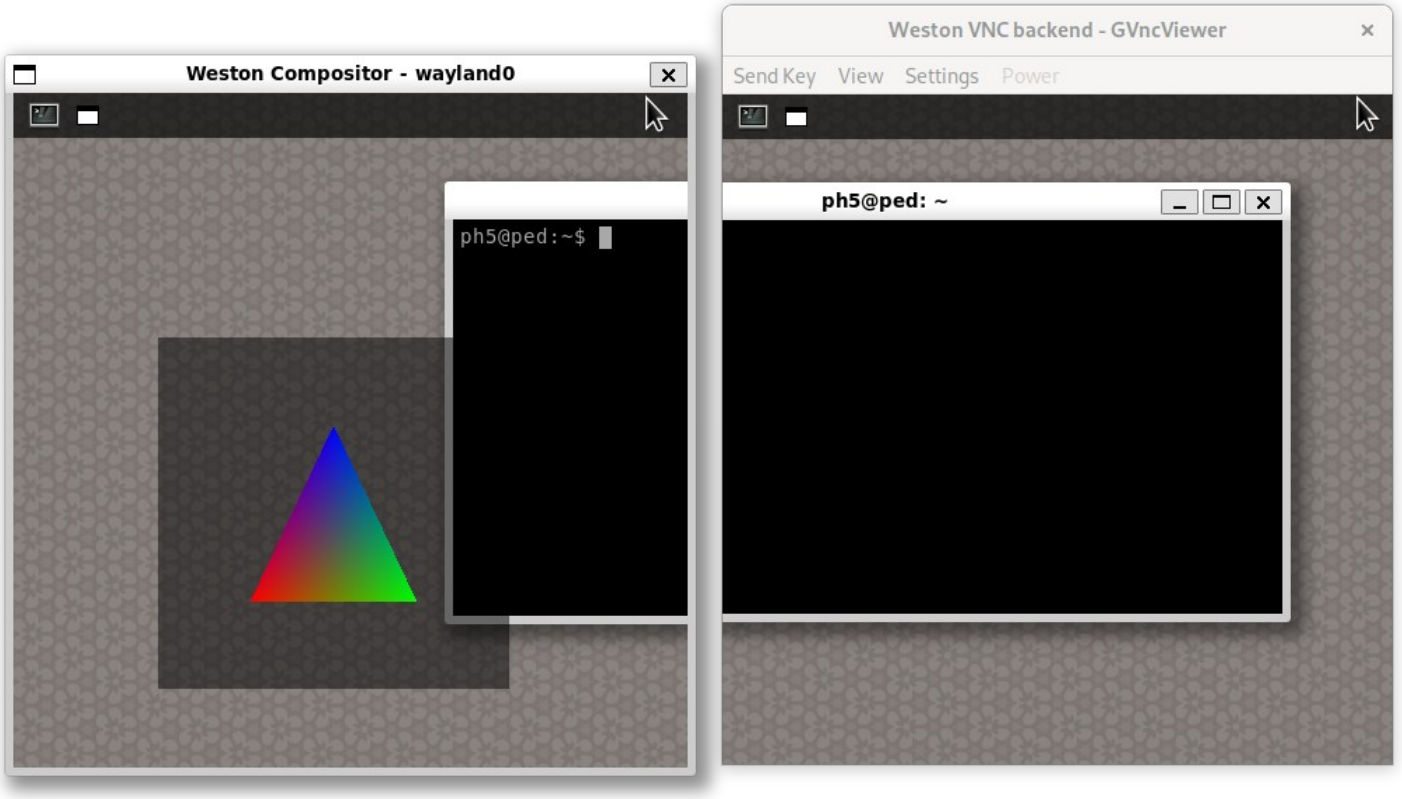



| | | |
|---|---|---|
| DRM | Head | Output |
| PipeWire | Head | Output |
| VNC | Head | Output |

# Weston multi-backend support

- Replace single backend on compositor with backend list
- Make heads point to their backend
  - So outputs are created and configured by the correct backend
- Move GL renderer interface from backends to compositor, to be shared by multiple backends
  - Renderer is still instantiated by the first backend
- Add command line options to load multiple backends
- Fix backends to support being loaded simultaneously

# Demonstration



# Weston merge requests

- Heterogeneous output support (!577)[1]
- Multi-backend support (!578)[2]
- GL renderer FBO support (!879)[3]
- PipeWire backend (TBD)
- VNC backend (!880, !881)

[1] https://gitlab.freedesktop.org/wayland/weston/-/merge_requests/577
[2] https://gitlab.freedesktop.org/wayland/weston/-/merge_requests/578
[3] https://gitlab.freedesktop.org/wayland/weston/-/merge_requests/879
https://gitlab.freedesktop.org/pH5/weston/-/tree/embedded-recipes-2022

# Outlook - Improvements

- Weston currently does not support overlapping outputs[1]
  - Damage tracking on the virtual desktop spanning all outputs
- Dynamic outputs
- Configurable frame rate
- VNC performance
  - Neat VNC GBM BO import
  - Latency reductions (preemptive render and encode)
  - Hardware accelerated encoding
- „Compositor bypass“ for PipeWire and VNC backends

[1] https://gitlab.freedesktop.org/wayland/weston/-/issues/494

# Outlook - Alternatives

- wlroots
  - Currently does not support hardware overlay planes
  - Work is ongoing to use libliftoff for this
  - Detailed in issue #3371 („Add a paint list struct“)[1]

- wayvnc
  - Works with wlroots, could possibly work with Weston headless via
  - ext-screencopy-v1 protocol[2]

[1] https://gitlab.freedesktop.org/wlroots/wlroots/-/issues/3371
[2] https://gitlab.freedesktop.org/wayland/wayland-protocols/-/merge_requests/124

# Thanks!

Questions?