Surfaces for a Dock

Usecase, proof of concept, RFC

kaeser@gubbe.ch / furball@google.com

XDC 2025 / Lightning Talk

Who? Why?

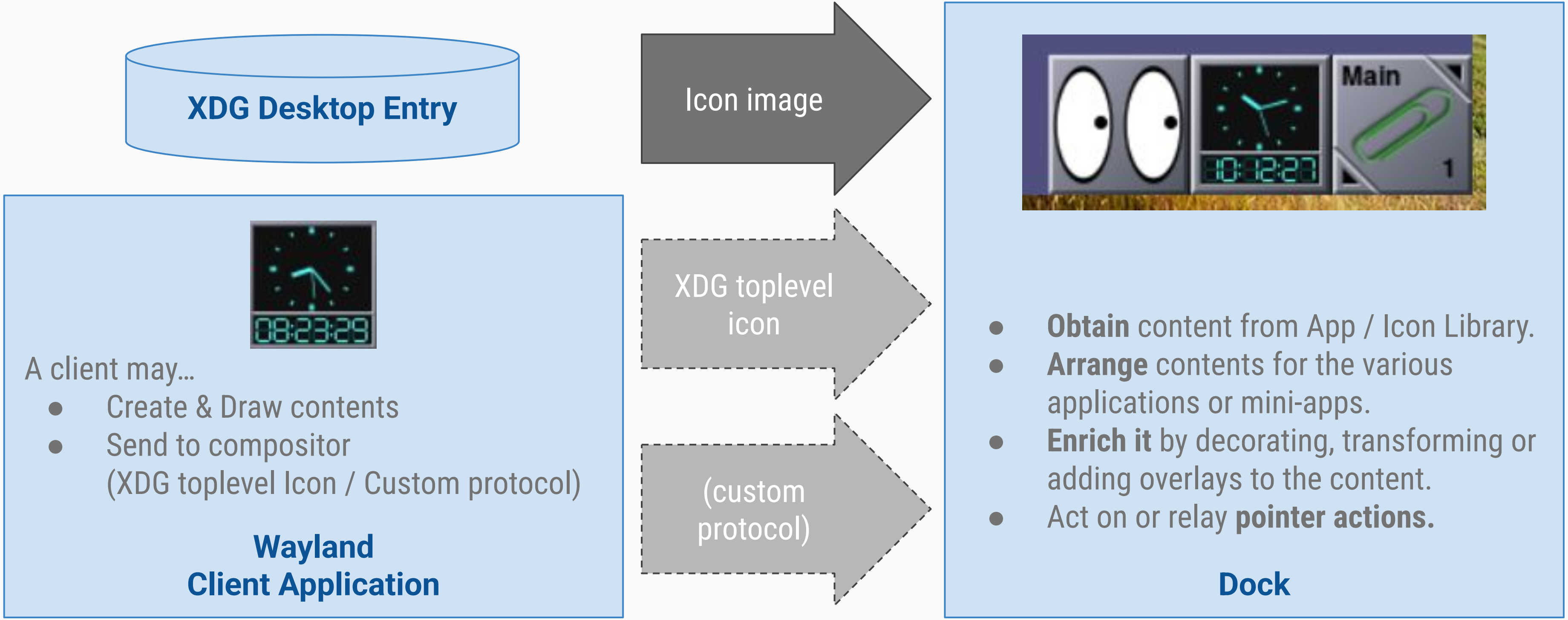
Who? Philipp Käser, Zurich, Switzerland

- Site Reliability Manager @ Google
- Day job: Run Google Cloud IAM

Side Hustle: Build a compositor to replace Window Maker

Surfaces for a Dock: **Use Case**

Window Maker's **Dock** holds icons and mini-apps ("dockapps") for launching programs and system monitoring.



XDG Desktop Entry

Icon image

XDG toplevel icon

(custom protocol)

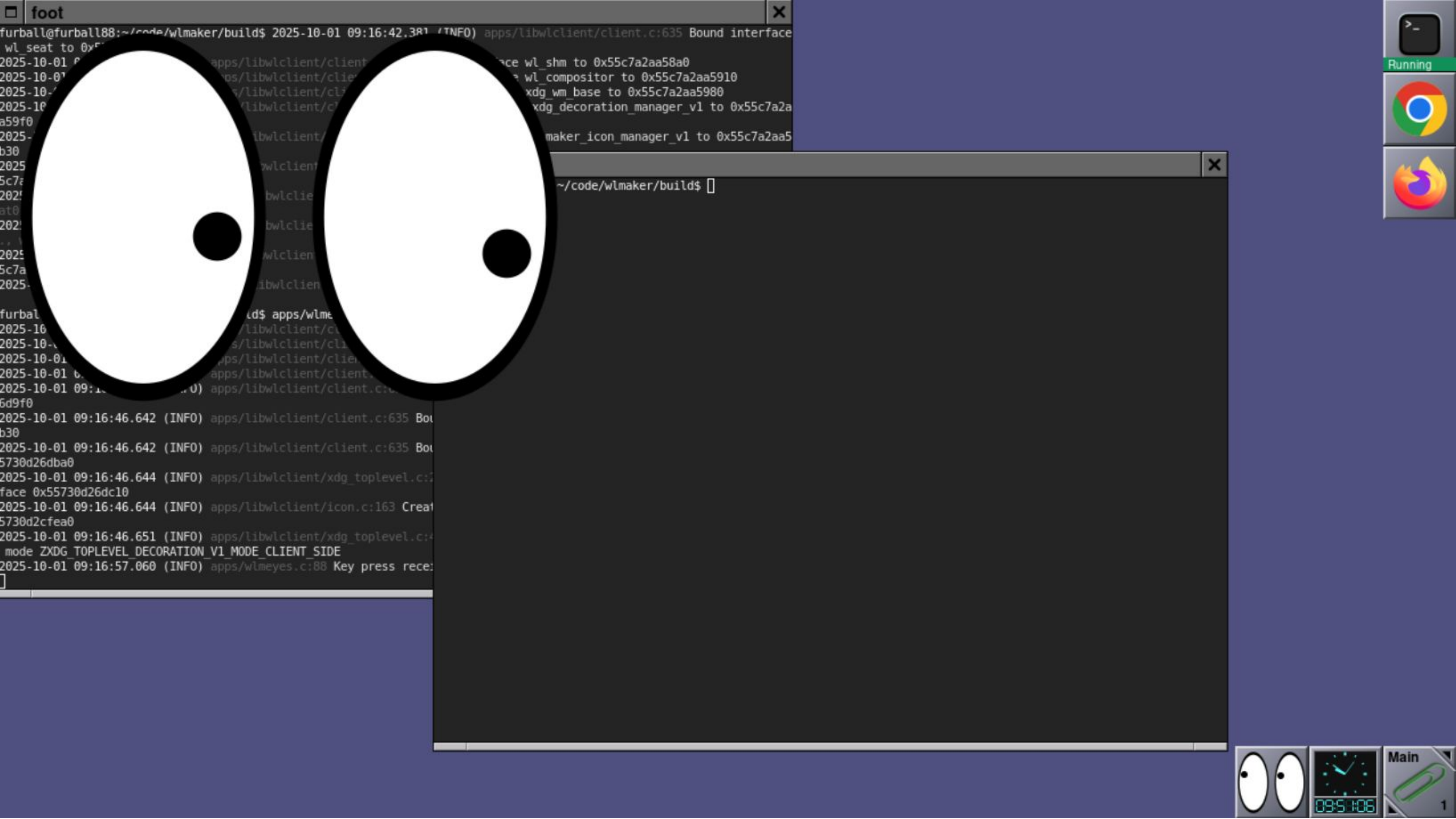
A client may…

- Create & Draw contents
- Send to compositor (XDG toplevel Icon / Custom protocol)

Wayland Client Application

- **Obtain** content from App / Icon Library.
- **Arrange** contents for the various applications or mini-apps.
- **Enrich it** by decorating, transforming or adding overlays to the content.
- Act on or relay **pointer actions.**

Dock



```
furball@furball88:~/code/wlmaker/build$ 2025-10-01 09:16:42.381 (INFO) apps/libwlclient/client.c:635 Bound interface
 wl_seat to 0x5
2025-10-01 0                  apps/libwlclient/client                   ace wl_shm to 0x55c7a2aa58a0
2025-10-01                    ps/libwlclient/clie                       e wl_compositor to 0x55c7a2aa5910
2025-10-                      s/libwlclient/cli                          xdg_wm_base to 0x55c7a2aa5980
2025-10                       /libwlclient/cl                            xdg_decoration_manager_v1 to 0x55c7a2a
a59f0
2025-                          ibwlclient/                                maker_icon_manager_v1 to 0x55c7a2aa5
b30
2025                            wlclient
5c7a
2025                            bwlclie
at0
2025                            bwlclie
..
2025                            wlclien
5c7a
2025-                          ibwlclien

furbal                         ld$ apps/wlme
2025-10                       /libwlclient/c
2025-10-                      s/libwlclient/cli
2025-10-01                   pps/libwlclient/clien
2025-10-01 0                 apps/libwlclient/client.
2025-10-01 09:1          O) apps/libwlclient/client.c:
6d9f0
2025-10-01 09:16:46.642 (INFO) apps/libwlclient/client.c:635 Bou
b30
2025-10-01 09:16:46.642 (INFO) apps/libwlclient/client.c:635 Bou
5730d26dba0
2025-10-01 09:16:46.644 (INFO) apps/libwlclient/xdg_toplevel.c:
face 0x55730d26dc10
2025-10-01 09:16:46.644 (INFO) apps/libwlclient/icon.c:163 Creat
5730d2cfea0
2025-10-01 09:16:46.651 (INFO) apps/libwlclient/xdg_toplevel.c:
 mode ZXDG_TOPLEVEL_DECORATION_V1_MODE_CLIENT_SIDE
2025-10-01 09:16:57.060 (INFO) apps/wlmeyes.c:88 Key press recei
```

```
~/code/wlmaker/build$
```

Surfaces for a Dock: **Design Choices - Alternatives**

Currently: Dock running in compositor process

✅ Got all buffers, icons, surfaces…

😞 Risk to compositor robustness/security

😞 Non-reusable

Alternative 1: Dock in a separate process, compositing in… compositor

Alternative 2: Dock in a separate process, render icons in Dock.

Alternative 3: (better ideas welcome)

Surfaces for a Dock: **Design Choices - Dock in separate process (1)**

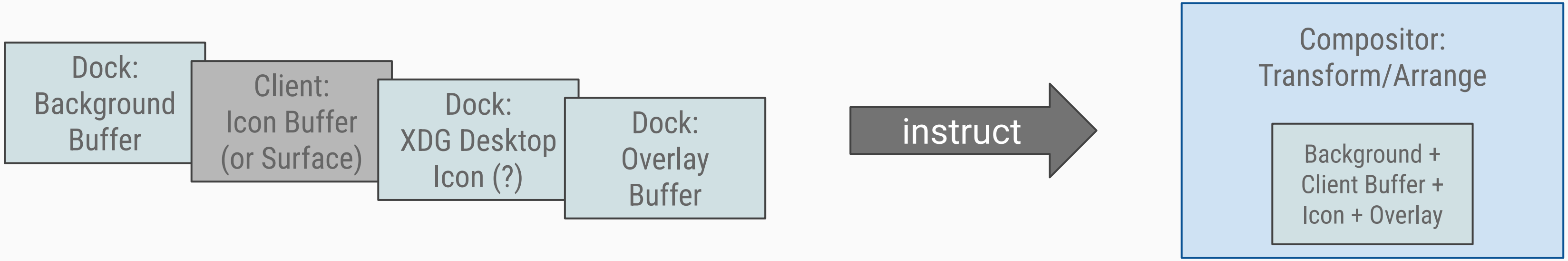
Currently: Dock running in compositor process

Alternative 1: Dock in a separate process, compositing in… compositor

✅ Functionality separation - reduced complexity/risk to compositor.

😞 Needs custom protocol to instruct compositor on transform/arrangement

😞 Reusable? Technically, but…


Dock: Background Buffer
Client: Icon Buffer (or Surface)
Dock: XDG Desktop Icon (?)
Dock: Overlay Buffer
instruct
Compositor: Transform/Arrange
Background + Client Buffer + Icon + Overlay

Alternative 2: Dock in a separate process, render icons in Dock.

Alternative 3: (better ideas welcome)

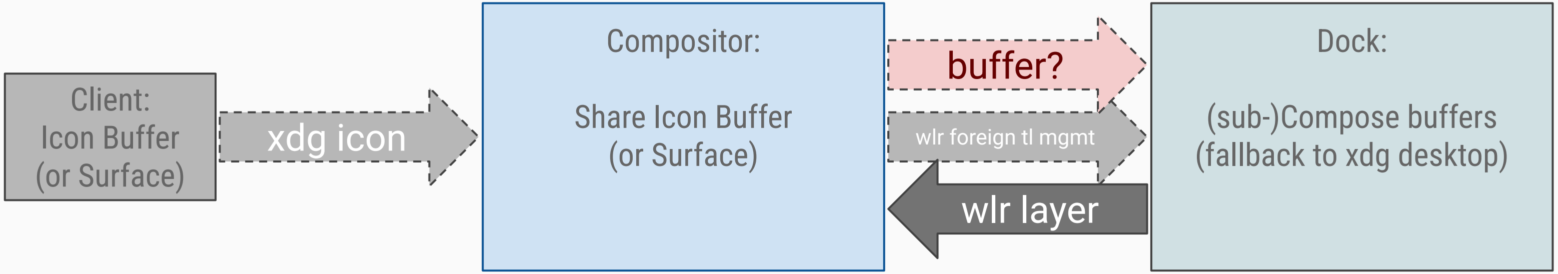
Surfaces for a Dock: **Design Choices - Dock in separate process (2)**

Currently: Dock running in compositor process

Alternative 1: Dock in a separate process, compositing in… compositor

Alternative 2: Dock in a separate process, render icons in Dock.

- ✅ Functionality separation - reduced complexity/risk to compositor.
- ✅ Reusable: Can work in absence of xdg icon / custom icon protocol.
- 😞 Need custom protocol to share icon buffer (surface?) from compositor to client
  Proof of concept: shmbuf ARGB32 to client.



Alternative 3: (better ideas welcome)

Surfaces for a Dock: **Design Choices - Request for Comments** 😀

Currently: Dock running in compositor process

Alternative 1: Dock in a separate process, compositing in… compositor

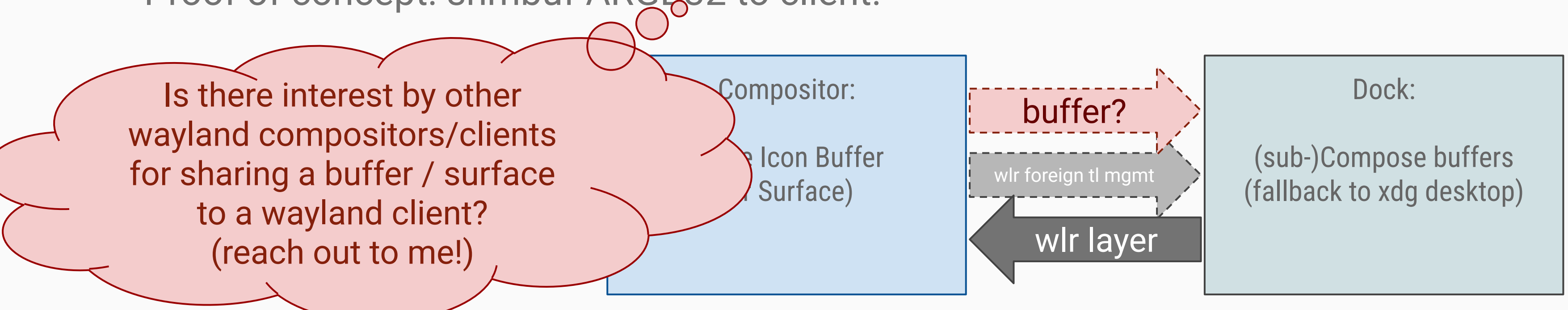
Alternative 2: Dock in a separate process, render icons in Dock.

✅ Risk isolated!

✅ Reusable: Can work in absence of xdg icon / custom icon protocol.

😞 Need custom protocol to share icon buffer (surface?) from compositor to client

Proof of concept: shmbuf ARGB32 to client.



Is there interest by other wayland compositors/clients for sharing a buffer / surface to a wayland client?
(reach out to me!)

Compositor:

e Icon Buffer
Surface)

buffer?

wlr foreign tl mgmt

wlr layer

Dock:

(sub-)Compose buffers
(fallback to xdg desktop)

Alternative 3: (better ideas welcome)

Surfaces for a Dock

Usecase, proof of concept, RFC

XDC 2025 Lightning Talk

kaeser@gubbe.ch

furball@google.com

https://github.com/phkaeser/wlmaker

PoC - Buffer exported to client

- https://github.com/phkaeser/wlmaker/tree/server-buffer

wlmeyes:

- Wayland-protocols !335 (kind of stalled… 😬)
- https://github.com/phkaeser/wlmaker (at HEAD)