

An Upstream-first Approach to Application Virtualization

Alyssa Ross



Concept

- Each (instance of an) application runs in its own VM
- Very little access to global resources by default
- Single desktop environment
- Single filesystem

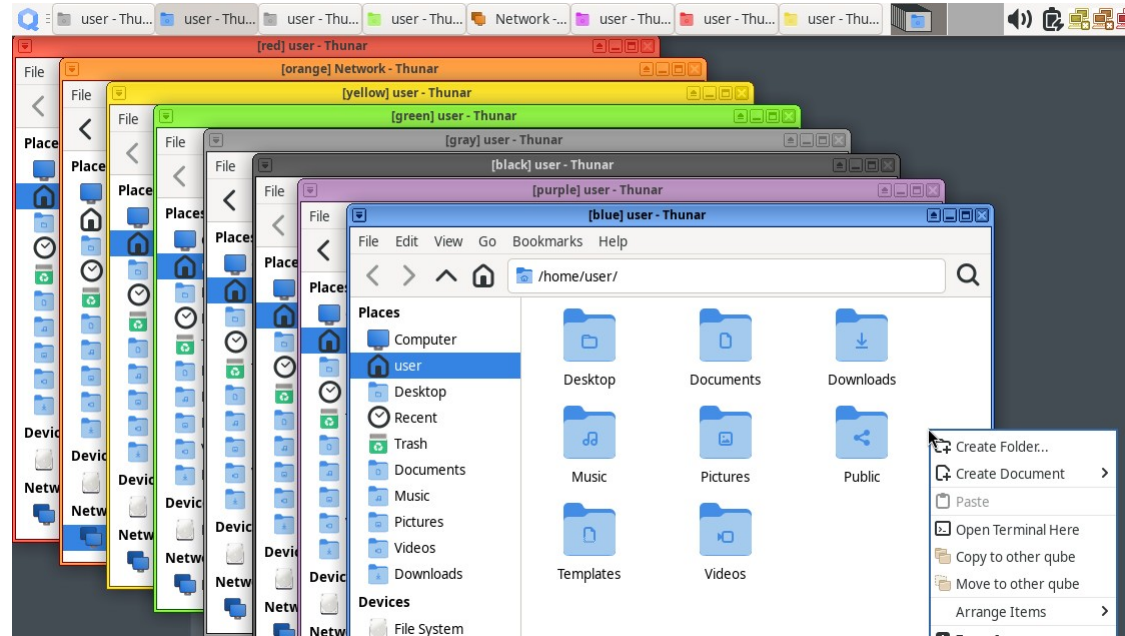


Background

- I was a dissatisfied Qubes OS user
- Learned about virtio-wl from Chrome OS
 - Later upstreamed as the Wayland virtio-gpu cross domain context type
- How hard can it be?



How do we do window decorations?



Window decorations in Qubes OS

So we just proxy Wayland, right?

- Proxying Wayland is difficult
- Introduces overhead
- Potential state synchronization problems

Identifying VMs to the compositor

- Also useful for limiting privileges of sandboxed clients
- There were two competing proposals
 - Flatpak-specific, using SOPEERPIDFD
 - Generic, using a trusted introducer
- Explained use case and provided implementation for generic solution
- Generic solution standardised as security-context-v1

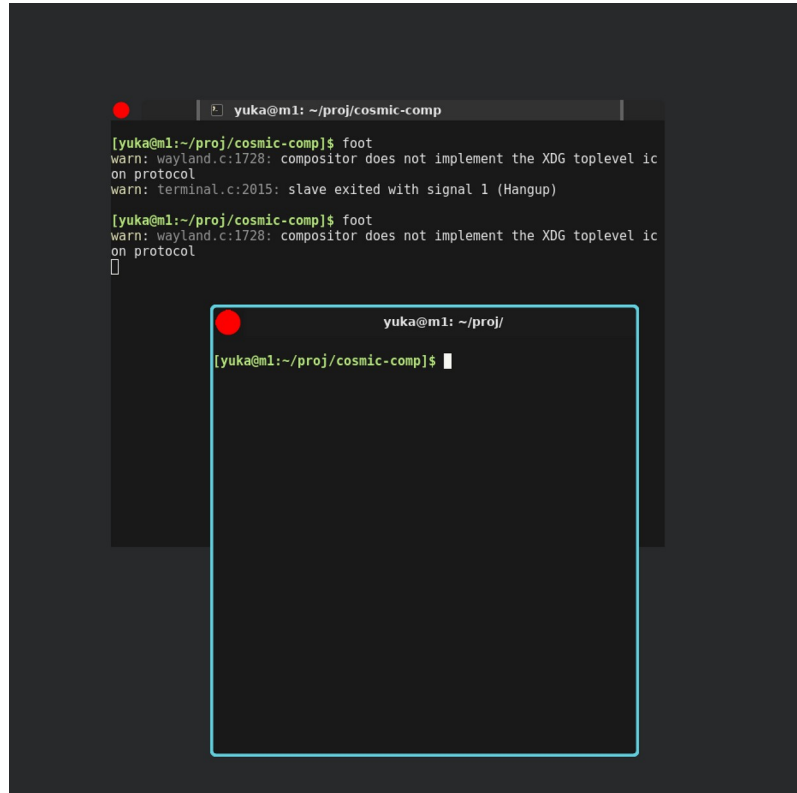


Drawing custom decorations

- We don't want our own special-purpose compositor
- Can we add customizability to an existing compositor?
 - Expose compositor main function in a library
 - Add hook interfaces where we need them



Customizable COSMIC!



The screenshot shows a terminal window titled 'yuka@m1: ~/proj/cosmic-comp'. The terminal output includes the command 'foot' and two warning messages: 'warn: wayland.c:1728: compositor does not implement the XDG toplevel ic on protocol' and 'warn: terminal.c:2015: slave exited with signal 1 (Hangup)'. A second terminal window is shown nested within the first, titled 'yuka@m1: ~/proj/' and containing the prompt '[yuka@m1:~/proj/cosmic-comp]\$'.

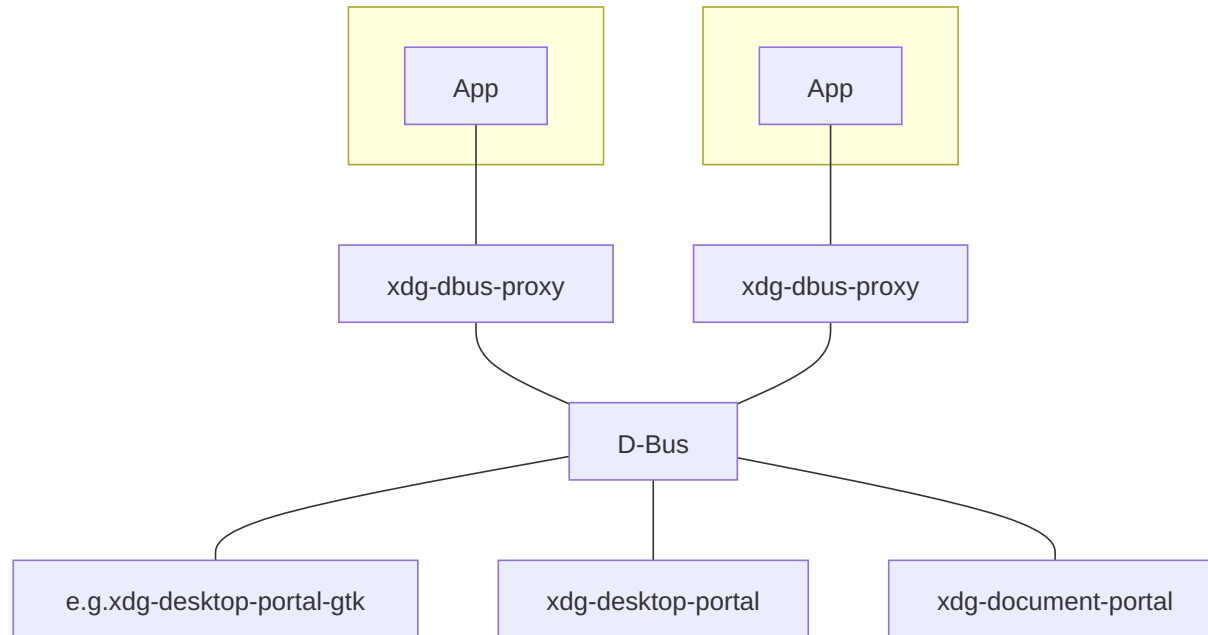
```
yuka@m1: ~/proj/cosmic-comp
[yuka@m1:~/proj/cosmic-comp]$ foot
warn: wayland.c:1728: compositor does not implement the XDG toplevel ic
on protocol
warn: terminal.c:2015: slave exited with signal 1 (Hangup)
[yuka@m1:~/proj/cosmic-comp]$ foot
warn: wayland.c:1728: compositor does not implement the XDG toplevel ic
on protocol
[yuka@m1:~/proj/cosmic-comp]$
```



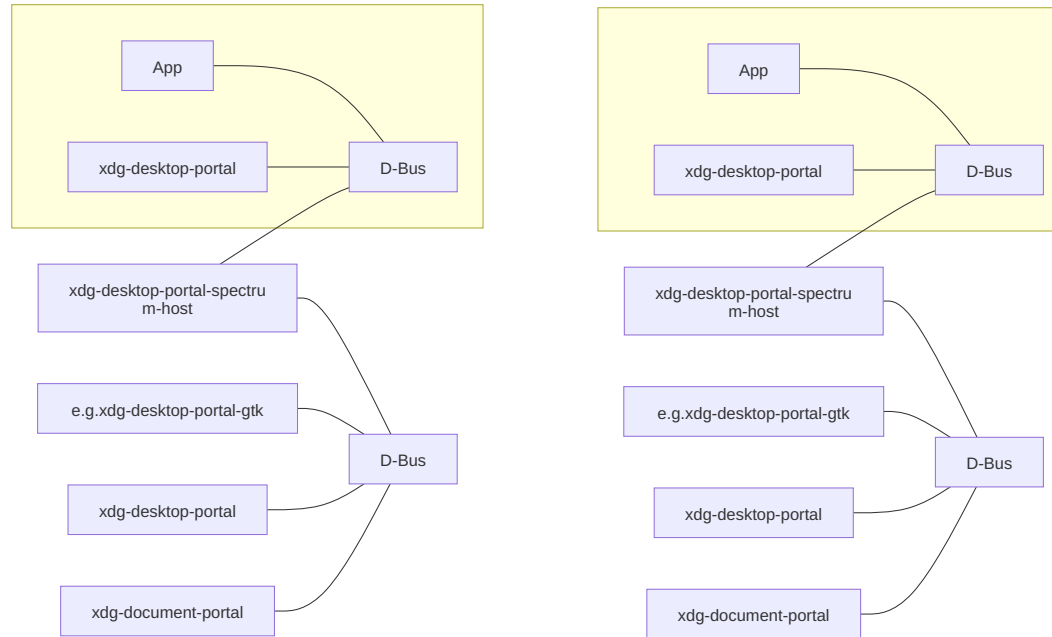
Window decorations: evaluation

- Upstream-first leading to a good result
- Little first-party code required
- Reusable upstream mechanisms
- Compromise is part of the process

File chooser portal: Flatpak



File chooser portal: Spectrum



Spectrum

Learnings for us & open questions

- Working upstream is slow
- Much better to be involved in the design phase, but...
 - How do we know when there's a proposal we need to be paying attention to?
 - What if we can't compromise to fit?
- Effort/momentum can make a big impact
- Focus on benefit to everybody, not just to us

Wishlist for future developments

- Consider virtualization/compartmentalization
- Wayland protocols are much easier to work with than D-Bus
- Think about privilege and complexity

Acknowledgements



@alyssais



qyliss



More about Spectrum (and me)

<https://spectrum-os.org/>

[#spectrum:fairydust.space](#)

Weekly “This Week in Spectrum”

More to come?

hi@alyssa.is

[@qyliss:fairydust.space](#)

[@qyliss@chaos.social](#)



Spectrum