

State of the Shell

Carlos Garnacho
Florian Müllner
Georges Stavrakas
Jonas Ådahl
Robert Mader



Welcome!



Mutter



Website



Mutter is a Wayland display server and X11 window manager and compositor library.

When used as a Wayland display server, it runs on top of KMS and libinput. It implements the compositor side of the Wayland core protocol as well as various protocol extensions. It also has functionality related to running X11 applications using Xwayland. When used on top of Xorg it acts as a X11 window manager and compositing manager. It contains functionality related to, among other things, window management, window compositing, focus tracking, workspace management, keybindings and monitor configuration.

Mutter is used by, for example, **GNOME Shell**, the GNOME core user interface, and by **Gala**, elementary OS's window manager. It can also be run standalone, using the command "mutter", but just running plain mutter is only intended for debugging purposes.

Contributing

Mutter is **Free Software** and is developed in the open.

To contribute, open merge requests at [https://gitlab.gnome.org/GNOME/mutter](#)



Pure Wayland Builds

- Many refactors
- Ifdefs for the last mile
- Intent: legacy-free builds



Explicit Sync

- Synchronization primitives for rendering pipelines
- Wayland clients passing buffers before they are finished (46)
- Compositor passing PipeWire buffers before they are finished (47)
- Implicitly handled in free drivers, but needed by NVIDIA
- Will make writing Vulkan drivers easier



DRM Lease

- Lease DRM devices to applications
- VR headsets!
- Monado, SteamVR

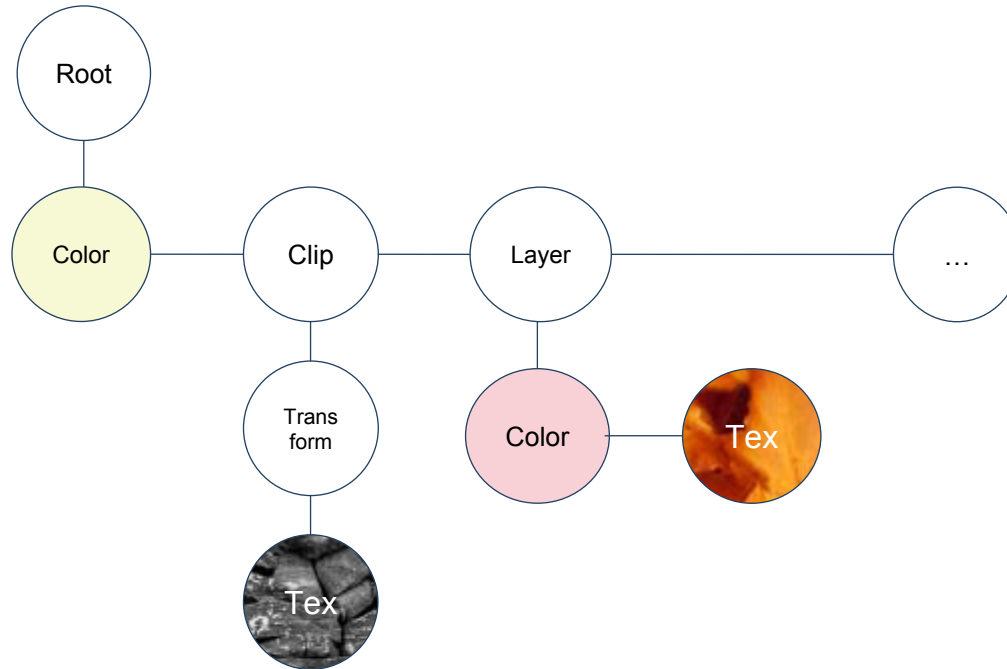


Snapshotting

- Rendering approach inspired (copied) from GTK4
- A new object - ClutterSnapshot - builds the rendering tree
- Completely retained render ops tree
- Still an experiment / research effort



Snapshotting



HDR

- More complex on a desktop compared to e.g. a TV and gaming console
- Mix SDR and HDR to avoid SDR being too bright (47)
- Allow Wayland clients to tag their surfaces (47? 48?)



HDR

- HDR can cause increased power usage so shouldn't be on all the time
- Still needs enabling via looking glass, but this will change



HDR

- Mutter uses 10 bits per component scanout format by default when supported by drivers

Should improve color accuracy / reduce banding (46)



Variable Refresh Rate

- Some things want to use the maximize refresh rate
 - Scrolling, animations
- Some things do not
 - Video, games
- Deciding when and mixing is difficult
- Kernel uAPI a bit awkward
- Thus, still behind an experimental feature flag



Video Offloading

- YUV/YCrCb “video pixel format” support (45)

YUV formats are usually require less space (“subsampling”).

Needs to get converted to RGB eventually.

“Offloading” means that this step happens in the compositor or even in the display hardware, both usually being more efficient than doing it in the app.

Toolkits like GTK4 and apps increasingly make use of this.

Related to HDR/color protocol work for optimal results.



Video Offloading

- Scaling and cropping using display hardware (46)

More efficient than doing an extra copy in GL/Vulkan

Useful for videos and games when the content dimensions don't match the screens.

Single-pixel-buffer protocol optimization allows black background when using a single hardware plane (“primary” plane), enabling zero-copy playback for most videos on most common hardware.

- Future (48?): Multiple hardware planes (“overlay”/“underlay”).



Screencasting

- Performance improvements (blitting) (46)
- Explicit modifiers (46)
- Explicit sync (47)



Input

- Last pieces of grab rework (46)
- Gesture framework (46)
- Gestures (47?)
- Global shortcuts (47?)



Input Configuration

- Tablets
 - Improved pressure curve (46)
 - Pressure range (47)
 - Stylus actions (47)
 - Tablet disambiguation (46)
- Trackball independent settings (46)
- XKB model (46)



Wayland Protocols

- DRM lease (46)
- linux-drm-syncobj-v1 (46)
- XDG dialog (47)
- Stable Tablet protocol (47)
- Session Management (47?)



Special Mentions



Special Mentions

- Cogl cleanups and GObjectification
- Removal of json-glib usage from clutter
- Reduction of deprecated APIs from split Cogl / Clutter times
- Moving away from Cairo

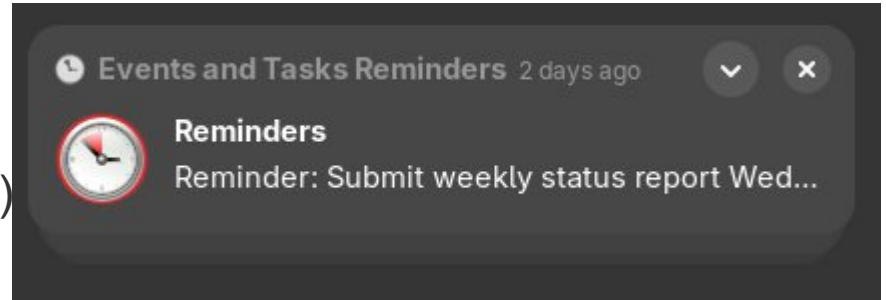


GNOME Shell

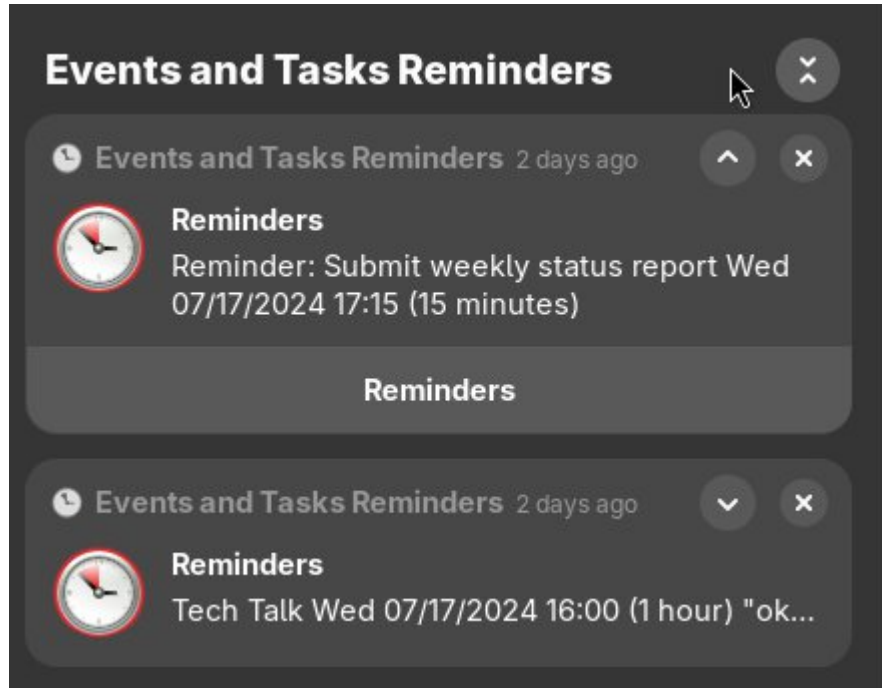


Notifications

- Headers (46)
- Expand in calendar / on touch (46/47)
- App grouping (47?)
- Support for protocol enhancements (v2)



Notifications

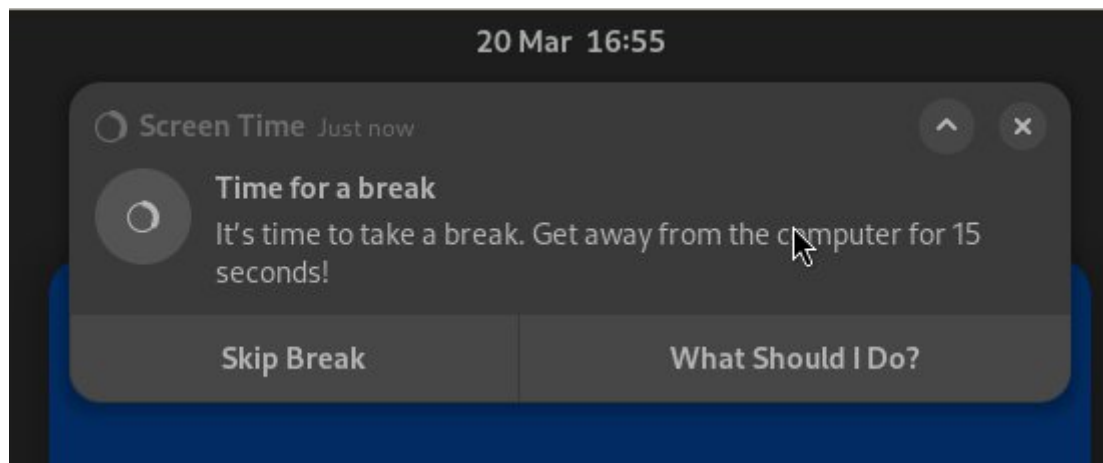


Break Timers

- Settings for user well-being

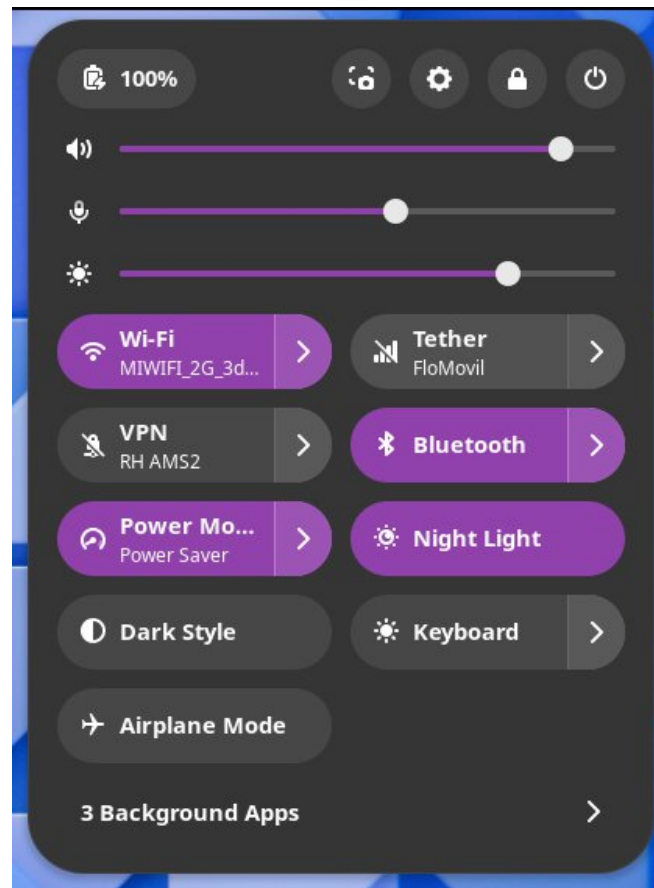
Typing breaks

Eye rest breaks



Accent Colors

- New appearance setting
- Standardized via desktop portal
- Support in libadwaita and the Shell



A11y Improvements

- Reworked high-contrast style (46)
- Follow “switch-shapes” setting (46)
- Shortcuts portal (47?)



Screencasting

- Using DMABuf sharing with Pipewire, doing format conversion and download in GL (46)
- VA-API HW encoding (47)



Special Mentions



Special Mentions

- New on-screen keyboard features (46)
- Better extension states (requested vs. real) (46)
- Improved captive portal handling (47)



Q&A



GNOMETM



Thank You!



GNOMETM



State of the Shell

Carlos Garnacho
Florian Müllner
Georges Stavrakas
Jonas Ådahl
Robert Mader

