

Remote desktop with full 3D support

- I'll try my best to go through how graphics on Linux work
 - I'm at most 90% sure of some of these things
 - X11 and Wayland
- Building all this with EasyBuild today
- Maybe a demo

Background

- Annoyed for the need for things like VirtualGL
 - Not flawless
 - LD_PRELOAD 🙄 (e.g. no good with flatpaks)
 - Needs to be installed into containers
 - EGL support (maybe?) 🤔
 - *vulkan actually seems to "just work" without extra steps*
 - No 3D acceleration of the desktop itself
- I can share my desktop screen with full 3D acceleration (game streaming, screen-sharing, screen recordings)
 - Why can't I just get that with headless remote desktop?!
 - I want GPU, headless, resizing, native and web client.
 - and great quality, low latency and wayland (as X11 is dying)

I'll try to assume you know very little about linux graphics

SILICATE CHEMISTRY IS SECOND NATURE TO US GEOCHEMISTS, SO IT'S EASY TO FORGET THAT THE AVERAGE PERSON PROBABLY ONLY KNOWS THE FORMULAS FOR OLIVINE AND ONE OR TWO FELDSPARS.

AND QUARTZ, OF COURSE.
OF COURSE.

The linux desktop stack

The image shows a Linux desktop environment with several windows open. The desktop background is a blue sky with a cityscape. The top panel shows the system tray with the time 18:44 and the user name Live System User. The left sidebar contains a list of applications, including Run Program..., Terminal Emulator, File Manager, Mail Reader, Web Browser, Settings, Administration, Accessories, Development, Documentation, Graphics, Internet, Multimedia, Office, System, and Log Out. The Settings window is open, showing the Personal section with options for Appearance, Desktop, File Manager, Notifications, Panel, Preferred Applications, Screensaver, Window Manager, Window Manager Tweaks, Workspaces, and Xfdashboard settings. The Hardware section shows Bluetooth, Display, and Keyboard. The File Manager window is open, showing the /home/liveuser/ directory with folders for Desktop, Documents, Downloads, Music, Pictures, Public, Templates, and Videos. The File Manager window also shows a list of devices and places. A red arrow points from the text 'Compositor specific plugin' to the File Manager window. Another red arrow points from the text 'pam-devel udev-devel' to the system tray area. A third red arrow points from the text 'CLIENTS' to the File Manager window. A fourth red arrow points from the text 'CLIENTS' to the Settings window. A fifth red arrow points from the text 'CLIENTS' to the File Manager window. A meme image of two characters from the movie 'The Incredibles' is overlaid on the bottom left of the screenshot.

Applications liveuser - File Manager Settings 18:44 Live System User

Settings
Customize your desktop

Personal

- Appearance
- Desktop
- File Manager
- Notifications
- Panel
- Preferred Applications
- Screensaver
- Window Manager
- Window Manager Tweaks
- Workspaces
- Xfdashboard settings

Hardware

- Bluetooth
- Display
- Keyboard

liveuser - File Manager
Compositor specific plugin

/home/liveuser/

DEVICES

- File System
- Anaconda
- 1.1 GB Volume
- Anaconda

PLACES

- liveuser
- Desktop
- Trash

NETWORK

- Browse Network

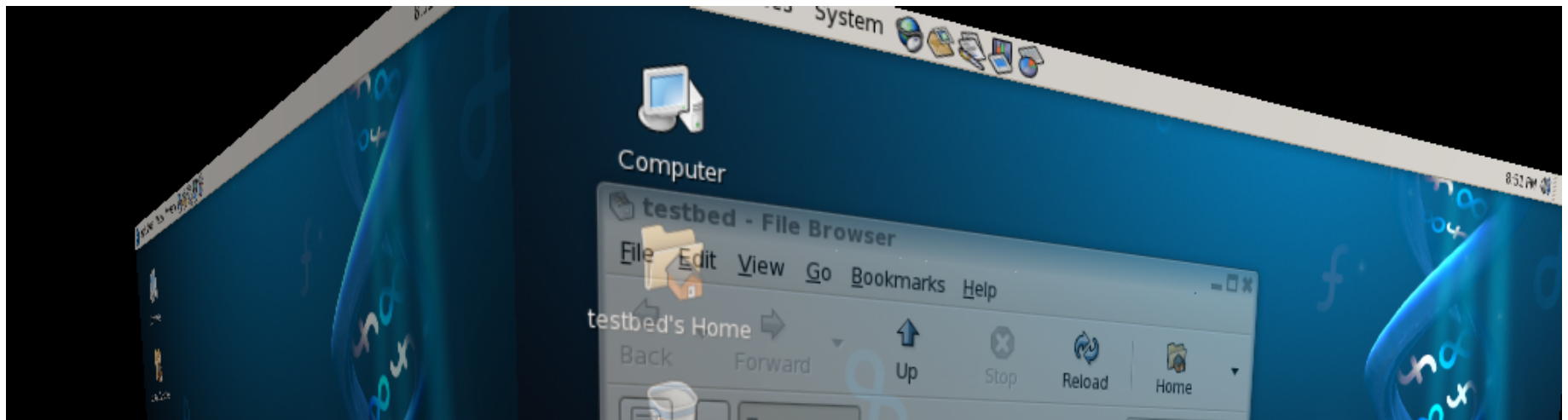
8 items, Free space: 1.7 GB

pam-devel
udev-devel

CLIENTS

Compositor

- composes application buffers into the final screen: Mutter, KWin, labwc
- Xorg itself is like a simple compositor, or runs the Composite extension to run e.g. picom/compiz
- Renders the final pixels, might do transparent windows, shadows, 3D, virtual desktops



Window manager

- controls placement, tiling/stacking, minimizing
 - openbox, i3
- On X11: window decorations, unless "client side decorations" with themes
- Light weight, called infrequently (e.g. only when resizing a window)

Clients

- The **server** is the one rendering and outputs to the screen, anything that then send draw commands to that server are the **clients**
- There may be X11/Wayland specific parts of a client
 - Wayland protocol extensions (e.g. `gtk-layer-shell`)
 - Implement X11 draw calls directly.
 - Or they are abstracted away via Qt, GTK, ...
- Essentially just normal GUI applications

Panels/toolbars/docks

- Just clients.
- Window hints instructs to treat it different from the rest
- Nothing special here: You can easily build a new "dock" application in EasyBuild just like you'd build ParaView and run it in your remote desktop.
 - Eww, Quickshell
- Virtual desktop switchers need to specifically be made for the underlying compositor

Desktop background

- Desktop background, right click, ~/Desktop/* icons , trash
- Also just a client (pinned to the bottom layer)
- Again you could build and run any you want very easily: you can run many desktop environments without it.

The 3D stack

- OpenGL, e.g. `glBegin(GL_TRIANGLES); glVertex3f(0, 0, 0);...`
 - GLX and EGL is how you initialize the context

```
Display *dpy = XOpenDisplay(NULL);  
...  
GLXContext ctx = glXCreateContext(dpy, vi, NULL, GL_TRUE);  
glXMakeCurrent(dpy, win, ctx);
```

- GLX is X11 specific, X11 can also do EGL (rare!)
 - Wayland does EGL.
- Vulkan, e.g. `vkCmdDraw(cmdBuffer, 3, 1, 0, 0);`
 - Different way of obtaining buffer with `VkSurfaceKHR` and platform extensions.
 - 300 LOC to get a single triangle on screen.
 - Vulkan even leaves picking the physical device up to the application.
 - Try out `Datoviz` demos

Low level stuff

- Generic Buffer Manager (GBM) allocates GPU buffers for rendering.
 - NVidia fought this hard, wanted EGLStreams instead
- Direct Rendering Manager (DRM):
 - Abstract GPU devices for rendering, command submission, sync,
 - `/dev/dri/cardX`
 - `/dev/dri/renderD*` - access to GPU rendering and transcoding without display server permissions
- Kernel Mode Setting (KMS)
 - Exposed via Direct Rendering Manager (DRI) API
 - Handles outputs, not rendering
- 🤔 don't ask any hard questions here please

X11 and Wayland

- Lets compare X11 and Wayland

X11

- You run a X11 server and there are more than you probably think
 - Maybe you know of Xorg, Xvnc, Xvfb, XWayland?
 - XQuartz, Xephyr, Xnest, Xpra, KasmVNC, Xming, VcXsrv
 - Xgl, Xegl, Xsun, KDrive, X11rdp, XDarwin, Cygwing/X, X386, X410, X-Win32, Xsgi
- In a modern desktop you might run a compositor
 - picom, compton, kwin, xfwm4, kwin, ...

Some critical limitations

- Xvfb nor Xvnc does not support any direct rendering (DRI)
 - You rely on interposer like `VirtualGL` that renders it elsewhere.
- You *can* run a true Xorg for a user, but
 - Nvidia does not seem to like multiple Xorg sessions
 - You'd need to run something separate like the unmaintained `x11vnc` that captures any existing `DISPLAY`
 - Dynamic resizing is a pain (surprisingly broken in `Xdummy video`)
- Xorg xrdp wrapper video driver (called `xrdpvideo`) for multiseat does not support Nvidia drivers.

Wayland

- Merged the display server ("Xorg") + input handling + compositor.
- Wayland (and X11) is actually just a protocol; libwayland* is just protocol plumbing.
- No central Xorg binary equivalent:
 - you just start the compositor: `sway, labwc --session . . . , ...`
- Note: Compositors are (still) opinionated
 - they determine if/how virtual desktops work
 - window management
 - custom protocols used with "their own" clients (e.g. virtual desktop)
<https://absurdlysuspicious.github.io/wayland-protocols-table/>
 - all different CLI flags, environment variables and behavior
- You can nest compositors in wayland (try it on your laptop!) 🤔
- Wayland ❤️ XDG: Be ready to set `XDG_*_DIR`

Remote desktop (X11)

- The X server protocol was implemented by Xvnc, allowing any linux X11 desktop to support VNC automatically.
- xrdp for RDP.
- Quite easy to capture screen and handle inputs in any existing X11
 - x11vnc captures existing X server (typically Xorg)
 - Zoom, Steam, M\$ Teams

Remote desktop (Wayland)

- With wayland it's compositor specific, but:
 - compositors supporting `wlr` protocols can get VNC via `wayvnc`
 - `xrdp` development branches can work with `wayvnc`
 - `xdg-desktop-portal` allows for capturing screens (used by screenshot tools, Zoom, etc.)
 - Gnome and KDE doing their own things (not focused on headless)
- It's basically always "screen capturing" like with `x11vnc`

XWayland

- A full X server (like Xorg, Xvnc and the rest)
 - No LD_PRELOAD shenanigans
- You can technically still use VirtualGL under Wayland
- Gets its 3D buffers via the Waland compositor.
- Hopefully much more widely tested and used
- Does rely on a few wayland protcols that the underlying compositor need to support.
 - <https://absurdlysuspicious.github.io/wayland-protocols-table/>
 - Just google "wayland protocol support table"

Headless remote desktop

- Many remote desktop solutions expect you to be sharing an existing session.
- Lets quickly review some options and how they deal with
 - headless setup
 - authentication
 - client requesting desktop resize
- Note that it normally matters if your session is detected as having a **seat** or not:
`loginctl`.

Good old X11 VNC

- turbovnc -> Xvnc -> xsession script ->xfce-session etc.
 - Several other VNC implementations, e.g. KasmVNC
- Authentication via vncpasswd password is typical (or PAM)
- Typically supports client side desktop resize
 - Via RFB (the VNC protocol) commands (either or both)
 - SetDesktopSize
 - ExtendedDesktopSize (newer, async)
 - Client support for resize if worse than the server side


True Xorg

- Need permissions to `/dev/dri/cardX`, `/dev/dri/renderD128`
- Write a `xorg-headless.conf` with fake display. One per GPU if needed.
 - You might already be doing this for VirtualGL
- Launch Xorg, fill it with X11 clients `startxfce4`, and attach `x11vnc`:

```
1 #!/bin/sh
2 Xorg :1 -config /etc/X11/xorg-headless.conf &
3 export DISPLAY=:1
4 sleep 1
5 dbus-run-session startxfce4 &
6 exec x11vnc -display :1 -forever
```

- Need unmerged PR functionality for `SetDesktopSize` (no maintainer).

xrdp

- Auth goes via PAM, can't just launch it like turbovnc
 - it's like sshd, single xrdp service on port 3389
- xrdp-sesman manages auth and sessions
 - starts new sessions upon first login, reuses existing session
 - Fantastic for login nodes! (thinlinc )
 - Challenging for slurm jobs...
- Gets framebuffer and sends inputs via:
 1. Xorg using drivers xrdpkeyb, xrdpmouse, xrdpdev
 2. Xvnc shim
 3. wayvnc shim
 4. xdg-desktop-portal in the future...?

Xorg example (conf)

```
1 # /etc/xrdp/xrdp.ini
2 [Xorg]
3 name=Xorg
4 ...
```

```
1 # /etc/X11/xrdp/xorg.conf
2 Section "Device"
3     Identifier "Video Card (xrdpdev)"
4     Driver "xrdpdev"
5     Option "DRMDevice" "/dev/dri/renderD128"
6     Option "DRI3" "1"
7     Option "DRMAAllowList" "amdgpu i915 msm radeon"
8 EndSection
9 ...
```

```
1 # ~/.xsession
2 dbus-run-session xfce4-session
```

The Wayland situation

- wayvnc for compositors compatible with or based on wlroots (sway, river, wayfire, labwc).
 - Uses a set of wayland protocols for screen capture, input, resize.
- xdg-desktop-portal (remote desktop portal, screen cast portal) 🙄
- KDE and Gnome doing their own thing?
- Just like with Xorg, it (always?) comes down to:
 - Convincing the compositor to start headless (and still use the GPU)
 - Attach a remote desktop solution to it

Headless wlroots

```
1 module load labwc Xfce wayvnc
```

```
1 export WLR_LIBINPUT_NO_DEVICES=1
2 export WLR_BACKENDS=headless
3 has_gpu=$(nvidia-smi &> /dev/null && echo 1 || echo 0)
4 if [[ "$has_gpu" -eq 1 ]]; then
5     export WLR_RENDER_DRM_DEVICE=$(readlink -f /dev/dri/by-path/pci-$(nvidia-smi --query-gpu=gpu_bus_i
6     export EGL_PLATFORM=wayland
7     export WLR_RENDERER=vulkan
8     export VK_DRIVER_FILES=/usr/share/vulkan/icd.d/nvidia_icd.x86_64.json
9     export VK_IMPLICIT_LAYER_PATH=/usr/share/vulkan/implicit_layer.d/
10    export __EGL_VENDOR_LIBRARY_FILENAMES=/usr/share/glvnd/egl_vendor.d/10_nvidia.json
11    export __EGL_VENDOR_LIBRARY_DIRS=/usr/share/glvnd/egl_vendor.d/
12    export GBM_BACKENDS_PATH=/usr/lib64/gbm
13    export GBM_BACKEND=nvidia-drm
14 else
15     export WLR_RENDERER=pixman
16 fi
17 # export XDG_RUNTIME_DIR=...
18 dbus-run-session labwc -L -S ./xfce4-session_and_wayvnc.sh
```

```
1 #!/usr/bin/env bash
2 xfce4-session &
3 wayvnc -L info 0.0.0.0 # configure as needed e.g. --gpu
```

Wayland VNC

- You options are: wayvnc, maybe w0vncserver (very new, I have not tested)

```
1 sway -d # ... Running compositor on wayland display 'wayland-1'  
2  
3 export WAYLAND_DISPLAY=wayland-1  
4 wayvnc `hostname`
```

- or run wayvnc on compositor startup (autostart, add to session script, cf. previous slide)
- Desktop resize expects new asynchronous ExtendDesktopSize, does not seem to work in remmina(libvncserver), works in vncviewer (turbovnc).
- Auth via vncpasswd, PAM, ...?
- wayvnc --websocket ...

Wayland RDP

- xrdp can also grab from wayvnc
 - VNC-shim: connect to VNC via unix-socket, expose as RDP
 - if you use experimental branches...
 - and specifically use labwc and wayvnc...
 - Maybe adopts xdg-desktop-portal or wayland protocols in the future?
 - xrdp-session still needs to manage and launch sessions
 - I'm not the expert here (->Leonard 🧐)
- krdpserver, gnome-remote-desktop ? ? ?

What else

- xdg-desktop-portal planned/in progress for many:
 - Selkies - WebRTC based streaming (expect higher constant load)
 - RustDesk (i think).
- Sunshine + Moonlight. No Wayland yet i think.
- Xpra - Wants to tackle individual windows (complicated)
- other ideas like wpr's "xpra for wayland"
- waypipe
- Nesting wayland compositors!
 - Outer compositor does VNC, inner one can be any
 - May be some clipboard, input, XWayland gotcha's
 - I think cendio is going in this direction?

Building this stuff with EasyBuild

- We already built many clients; nothing special here.
 - `QT_QPA_PLATFORM` -> `xcb` or `wayland`
- (X11) Window managers are also trivial.
- Could we build Xorg?
 - Yes, it's almost the same as our Xvfb build.
 - But you'll need drivers (and `xorg-x11-nvidia`) which pulls in Xorg anyway.
 - If you have intel/AMD graphics, then maybe 🙄
- Wayland compositors and are surprisingly easy to build.
- Some stuff has (optional) deps on `pam-devel`, `udev` (`systemd-devel`)
- Building KDE/Gnome 😞
- My work is mostly exploratory, but we already have Xfce, sway, labwc, wayvnc (draft or merged PRs)

Your best bet today

- Following should be possible to launch with full 3D support:
- sway + wayvnc
- labwc + xfce (+ swaybg) is OK (Xfce-panel doesn't support ext-workspace-w1)
- Compose some custom with simple components:
 - labwc
 - pacmanfm-qt --desktop
 - sfwbar
 - vibecode a custom desktop in QML with quickshell (I'm actually serious)

Conclusion

- Still early days on the wayland side..
 - But it would likely solve all 3D acceleration natively
 - Also for containers
- Capturing screen in Wayland not as easy, but isn't all that bad...
 - requires `xdg-desktop-portal`, `wlr-screencopy`, ...
 - Main problem seems to be that the protocols never seem to get standardized
- Compositors also handle scanout of buffers to monitors
 - Still might come down to convincing them with a fake display, fake "seat", when using the GPU for rendering.
- Much effort only cares about desktop sharing on single user machines
 - Single sessions, no resize, ...
 - Gnome + systemd makes slurm jobs difficult
- See all discussions with dcommander on github:
 - <https://github.com/TigerVNC/tigervnc/issues/158>
 - <https://github.com/TurboVNC/turbovnc/issues/18>

Bonus slide:

- Using jupyterlab is an alternative for many users who want something desktop-like but doesn't need X11/Wayland applications specifically:
 - Terminal
 - File browser
 - Edit text files
 - Multiple tabs/windows
- While it can't run Wayland/X11 apps, there are many tools beyond just python notebooks:
 - xterm.js
 - jupyter-git
 - jupyterlab-h5web
 - jupyterview
 - jupyterlab-video-extension
 - NVdashboard