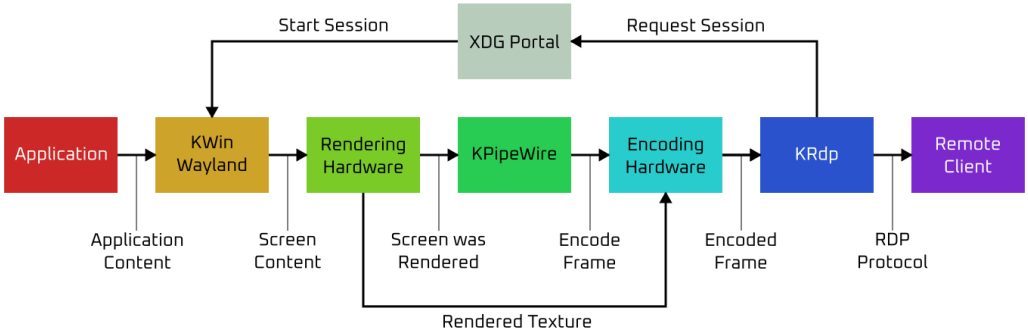
Spooky Action at a Distance

Remote Desktop for Plasma Wayland

Arjen Hiemstra

Akademy 2023

The Parts


Start Session
XDG Portal
Request Session
Application
KWin Wayland
Rendering Hardware
KPipeWire
Encoding Hardware
KRdp
Remote Client
Application Content
Screen Content
Screen was Rendered
Encode Frame
Encoded Frame
RDP Protocol
Rendered Texture

Next up...

1. In The Beginning
2. Setting the Stage
3. Moving Forward
4. The Future?

X11

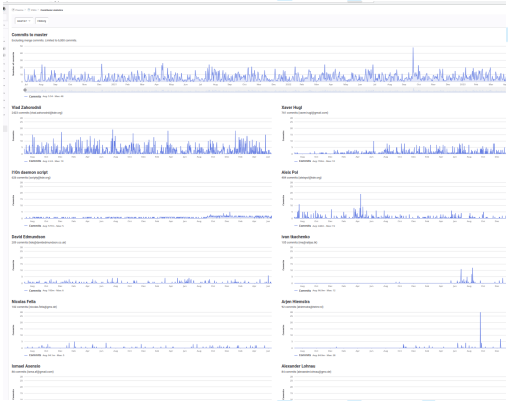
- X Protocol is Client-Server by design
- Forwarding over remote shell
- Various design limitations

Wayland

- New ground up design
- No longer client-server
- Aimed at maximum performance for local sessions

KWin

- Originally an X11 Window Manager
- These days also a Wayland compositor
- Lots of work going on at various levels



Next up...

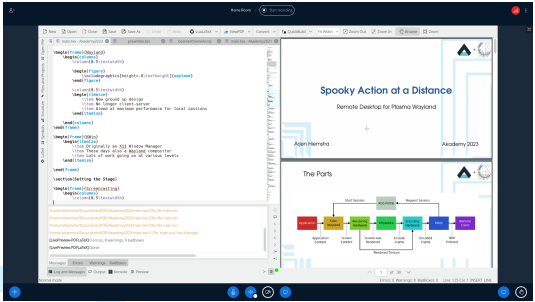
1 In The Beginning

2 Setting the Stage

3 Moving Forward

4 The Future?

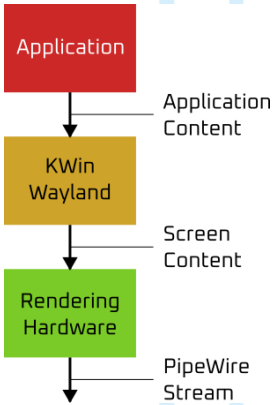
Screencasting



- Common feature in video conferencing software
- Share applicaton window or entire screen
- Easy but ugly on X11, quite a lot harder on Wayland

Screencasting Implementation

- Applications render contents to a texture
- KWin composites textures together into a screen image
- Expose screen image through PipeWire stream

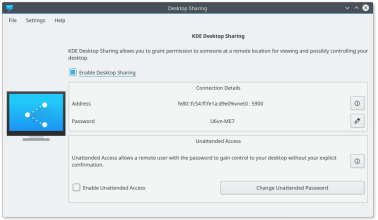


Remote Desktop

- Screencasting as base
- Remote input required
- XDG Remote Desktop Portal for permissions

KRFB and VNC

- Originally started to expose a KDE X11 session over VNC
- These days also supports Wayland
- VNC is rather limited



Evaluating

So we can already do Remote Desktop, but can we do better?

Requirements

- Low-latency (ideally zero-latency) between input and result
- Wide range of client support
- Extensible to support various additional features

RDP

- Created by Microsoft for Windows Remote Desktop
- Originally proprietary, now openly documented
- Many features through protocol extensions

FreeRDP

- Open-source implementation of the RDP protocol
- Both server and client implementations
- Many protocol extensions implemented

Next up...

1 In The Beginning

2 Setting the Stage

3 Moving Forward

4 The Future?

KPipeWire

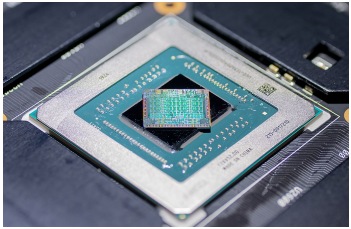
- Library to handle dealing with PipeWire video streams
- Initially for thumbnails
- Extended for video recording

Encoding

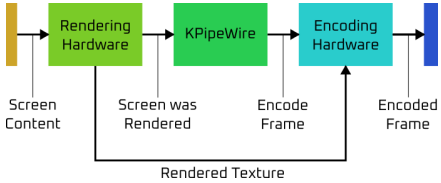
- Video streams need encoding using a video codec
- Uses FFmpeg underneath for actual encoding
- Support both writing to file or getting raw encoded frames

...Now Using Encoding Hardware

- Most modern hardware includes hardware to encode at least H.264
- Hardware encoding both faster and less resource intensive
- Currently using the VA-API implementation in FFmpeg



DRM (Direct Rendering Manager) Import



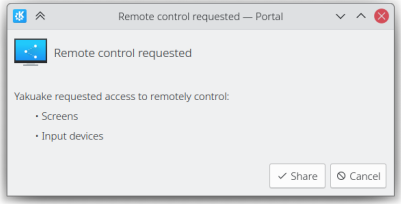
- Download then upload is expensive
- Also unnecessary if we can tell the encoder where the data is
- Low-level bits that can break in various ways

Permission Handling

- Arbitrary screen data sharing is dangerous
- Especially when connected to the internet
- Wayland by design disallows much of this

XDG Desktop Portal

- Originally started for permission handling in Flatpak
- Also useful for various Wayland related things outside Flatpak
- Common "front-end" API that delegates to desktop-specific implementation



Next up...

1 In The Beginning

2 Setting the Stage

3 Moving Forward

4 The Future?

KRdp

- Started at the beginning of this year
- Designed as library to implement an RDP server
- Also contains a command-line server

Current State

- Video streaming and remote input works
- Various situations and clients tested
- Alpha release any day now...

Future Plans

- Improve handling of the Remote Desktop portal
- Integrate into Plasma as system service
- Implement extensions such as clipboard and file sharing

Demo

Closing

Questions?

Links

KRdp	`https://invent.kde.org/plasma/krdp`
KPipeWire	`https://invent.kde.org/plasma/kpipewire`
FreeRDP	`https://github.com/freerdp/freerdp`