Screencasting on Wayland

Doğukan Korkmaztürk

Addressing the shortcomings

Moving from X11 to Wayland fundamentally changed how screen recording works under the hood and brought new challenges.

Who this talk is for:

- Compositor developers.
- Client application developers that use screencasting.
- Goal is to promote discussion!

Use cases of screencasting

- Sharing desktop
  - Content creation
  - Streaming
  - Video conferencing
  - Custom solutions
- Desktop environments
  - Application and screen previews
- Software development
  - Test verification
  - Debugging (Remote or local)
  - Bug reports

Screencasting Methods on Wayland

- PipeWire based
  - XDG Desktop Portal
    - Screen cast portal
    - D-BUS interface
    - Uses PipeWire nodes
    - The most common way
    - Not Wayland specific
  - PipeWire backends
    - Backend for compositors
    - No need for XDG Desktop Portal
    - Some examples are:
      - Gamescope
      - Weston 15

- Wayland Protocols
  - ext-image-copy-capture-v1
  - wlr-screencopy-unstable-v1 (Superseded)
  - Limited support from compositors

- KMS
  - Requires elevated permissions
  - Some examples are:
    - ffmpeg - kmsgrab
    - Sunshine
    - gpu-screen-recorder

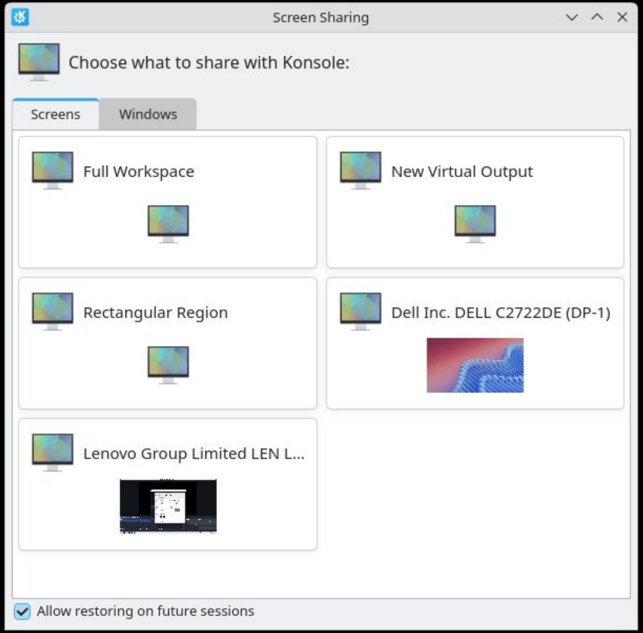
Explicit Synchronization in Screencasting

- Similar to Wayland explicit synchronization, accesses from clients and compositors to the shared buffers need to be synchronized.
    - XDC 2022 | Explicit Synchronization for Linux Display Servers | Erik Kurzinger
- For PipeWire based approaches:
    - PipeWire 1.2.0 - Added SPA_DATA_SyncObj
    - Mutter 47.0
    - KWin 6.3
    - OBS Studio 31.1 (and some fixes here)
    - NvFBC
- ext-image-copy-capture-v1 does not support explicit synchronization at the moment.
- Requires every client to implement it separately.
    - Unlike linux-drm-syncobj-v1, can't be hidden behind APIs like EGL, Vulkan WSI etc.

Permission Scheme

Problems with automation and remote recording

- Starting a screencasting session via XDG Desktop Portal requires GUI interaction.
  - Allow application to screencast.
  - Choose the output region for screencasting.
- Restricts automation and remote use cases.
  - Requires physical input from the user.
  - Client can’t request a specific screen for recording.
- Session can be restored without user input using tokens.
  - Creating the very first session still requires physical input.
  - Inconsistent behavior between compositors.
- Need for a common ahead-of-time permission scheme.



Screen Sharing

Choose what to share with Konsole:

✓ Allow restoring on future sessions

Call to action

- How can we solve these issues?
    - Explicit sync
        - Can we design an abstraction library?
    - The permission pop-up
        - Would the wider adoption of PipeWire backends for compositors help?
            - Implicit ahead-of-time permission.