The Wayland Compositor Architecture

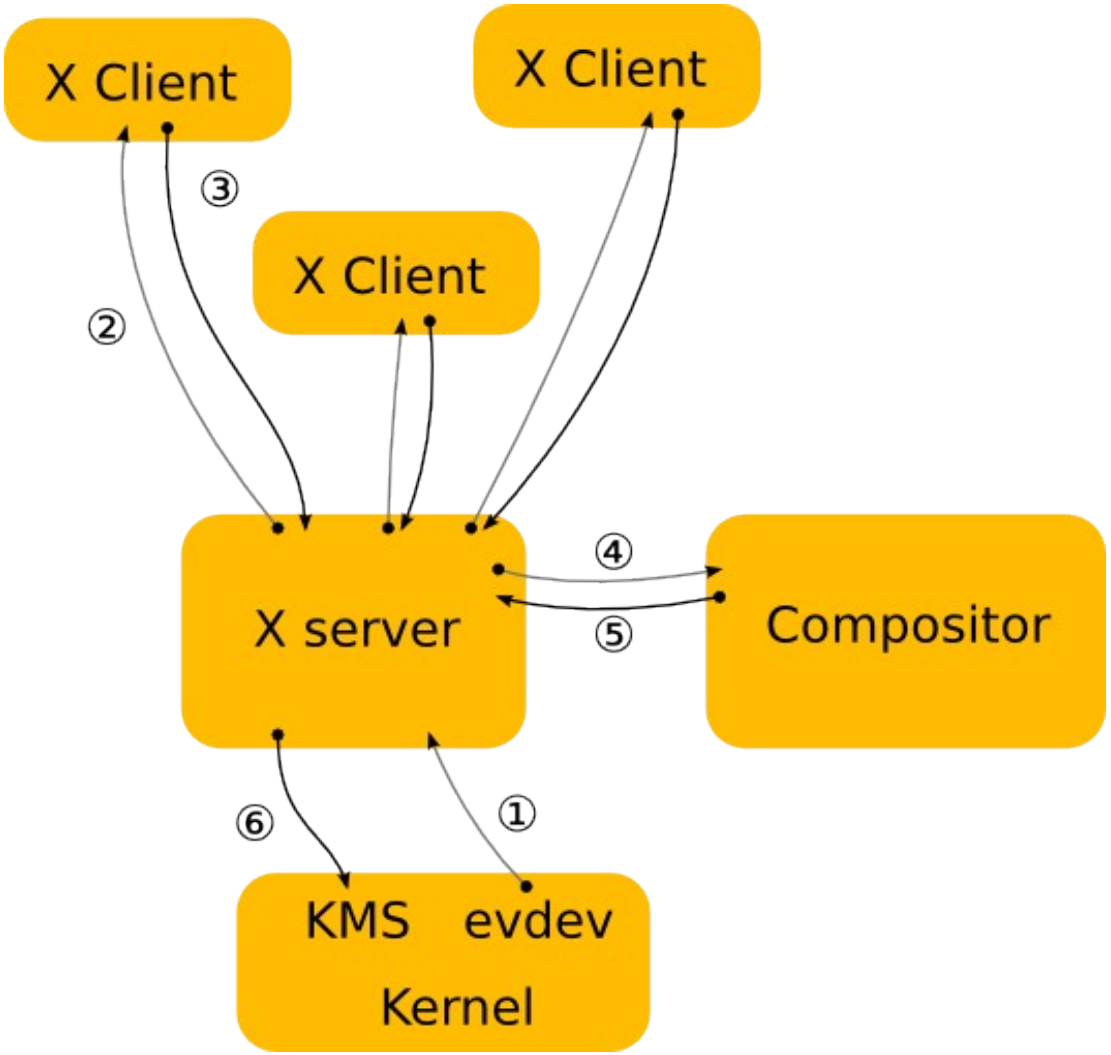
Kristian H. Kristensen
Intel

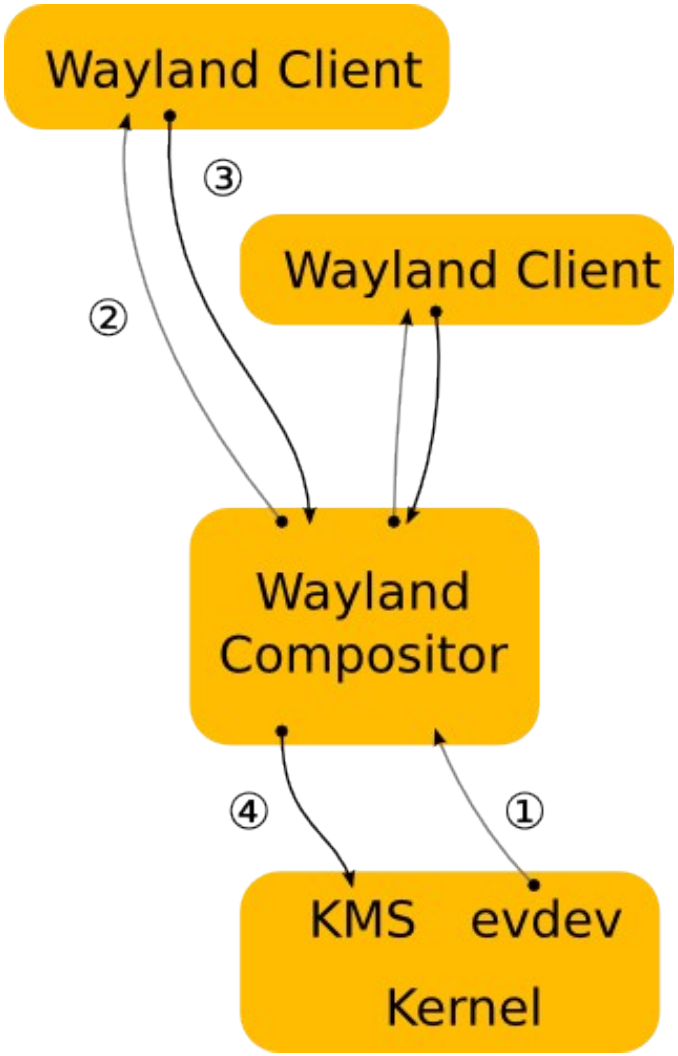
What is Wayland?

- A new protocol
- Client and server side libraries
- Sample compositor and clients

Background Story

- Functionality moving to the client
- Implementation moving into the kernel
- Final presentation moving to compositor





Wayland Rendering

- No rendering API
- Buffer sharing
- We can...
  - Attach a new buffer
  - Update an existing buffer
  - Tell the compositor what changed

Wayland Input

- Simpler than X, but not to be hand-waved
- Fewer grabs
- No subwindows
- No out-of-process WM/CM
- Reuse XKB

Inter-client, client/wm protocols

- ICCCM: selections, WM_NAME, WM_CLASS
- EWMH: window types, icons, struts, moving and resizing window
- XDND: drag and drop
- XSMP: Session Management

What else does X offer

- A platform for toolkits and UI middleware
- A platform for hardware vendors to enable
- 26 years of code churn
- 683792 lines of code

Toolkits on Wayland

- Qt
- GTK+
- Clutter
- SDL
- and X.org

Hardware Support

- Intel GenX (i915 and later)
- Intel PowerVR
- Open source nVidia and AMD drivers
- Wayland EGL platform for clients
- Server side driver API still WIP

Where is Wayland Going?

- MeeGo Tablet UI?
- Meego TV?
- GNOME Shell?
- Ubuntu Unity?

MeeGo Tablet UI

- Use Qt on Wayland in the clients
- Make mcompositor a Wayland server
  - Run mcompositor on EGL and KMS
  - Read input from evdev

Questions