Introduction to Wayland

Jesse Barnes
What is Wayland?

A compositing display management architecture and protocol
• Rolls window management, compositing, and the display server into a single process
  – Main open source implementation called Weston
• Does not include a rendering API
  – Clients use what they want and send buffer handles to the server
  – Current clients use Cairo, OpenGL, GLES
  – Software rendering fully supported as well through the SHM protocol
• Consolidates experience from the last decade of extending and enhancing X
Current status

Basics work today, full desktop support is close

• Qt was the first toolkit; port done by Intel and Nokia for MeeGo related projects
• EGL (with OpenGL or GLES) and Cairo have been supported since the beginning
• Improved Qt support coming in Qt 5
• GTK+ work coming along nicely
  – Many apps can work without any source changes (should even be possible to avoid a recompile in some cases)
  – Client side decorations in development
  – A few missing features like drag-n-drop

Track progress at http://wayland.freedesktop.org/
Wayland native applications

- Buffer handle passing, not copying
- Toolkits generally abstract windowing system
  - Pure Qt & GTK+ apps just needs a re-compile or loading with a different window system back end
  - Plain GL apps will need some work at the EGL level
Wayland with X11 compatibility

- X server can be started on-demand when X clients connect
- “Rootless” or full screen versions of X possible (just like Mac and Windows X servers)
FAQs

• “Who are these people, why don’t they just extend X?” or “These people must not understand X.”
  – Founder Kristian Høgsberg responsible for key X improvement of the past few years: DRI2, other core contributors are/were major X contributors as well
  – X not suited by design

• “What are the platform requirements for Wayland & Weston?”
  – Mainly buffer sharing so clients can render and pass a handle to the resulting buffer to the server
  – Short story: if you have a KMS and DRI driver you probably don’t need to do much
  – Weston back ends available for DRM, OpenWF, X, and Wayland, others definitely possible

• “How is input handled?” and “Does Wayland support touch/multitouch?”
  – Wayland protocol includes input handling, Weston supports input devices through evdev, and supports multitouch including libmtdev for devices requiring it

• “What about network transparency?” or “OMG they killed Kenny!”
  – Since there is no server side rendering, the problem is much simpler
  – RDP or similar protocols can be provided by a server like Weston, and likely be more efficient than X is today
Schedule

We expect that we can release Wayland 1.0 this year:
• 0.85, developer snapshot, protocol changes planned (already out)
• 0.90, starting beta, protocol frozen
• 0.9x, release candidates
• 1.0, first stable release
  – Marks beginning of stable protocol and API
  – Not world domination
  – Somewhere in first half of 2012
  – Will include 1.0 of Weston compositor as well