Wayland/X Compositor Architecture By Example: Enlightenment DR19

- **Blog:** http://e19releasemanager.wordpress.com
- **Contact:** zmike@enlightenment.org
- **E19 Release Date:** LOL

- **Source:** http://git.enlightenment.org/core/enlightenment.git/
  - “devs/discomfitor/e19” branch

Compositing 101

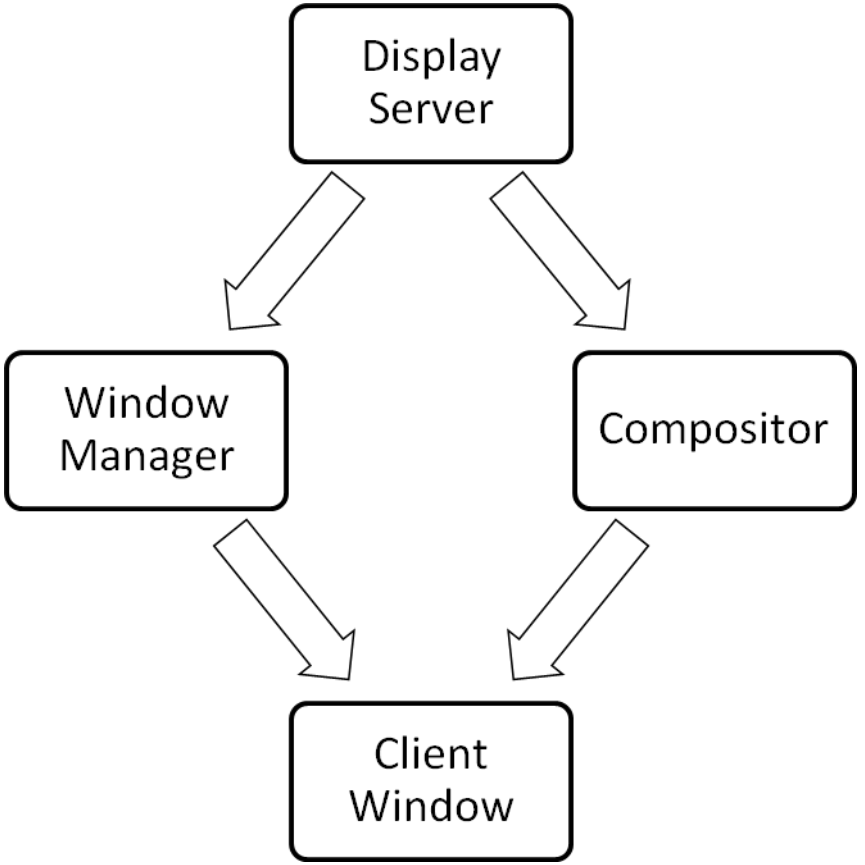
Compositing defined:

- “The combining of visual elements from separate sources into single images” (Wikipedia)

Compositing 102

Manual compositing

Where does the compositor fit in?



Key Terminology

Window Manager:

- Controls application window attributes:
  - Geometry (size, position)
  - Visibility
- Responds to application requests
  - Geometry requests
  - Drag-n-drop
- Can **ignore** application requests

Key Terminology (cont.)

Compositor:

- Only responsible for rendering
  - Receives/retrieves application window data
- Able to freely manipulate window pixel data
  - This is where “effects” come from
- Can **ignore** application visuals

- Example: xcompmgr

Key Terminology (cont.)

Compositing Window Manager

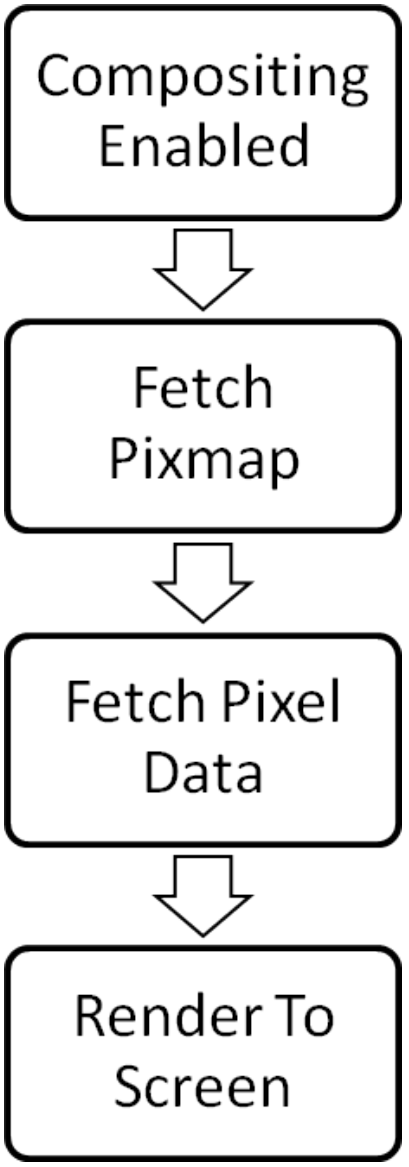
- Combines Window Manager and Compositor
- Controls all parts of application display
  - Geometry, requests, visuals
- Can ignore/change requests **AND** visuals

Display Systems: X(org)

- Windows can query other windows
  - Input can be grabbed
  - Visuals can be retrieved
- Borders drawn by Window Manager
- Compositing is *optional*
  - Compositor must manually fetch/query pixel data
  - Compositor only receives updates when regions are redrawn, **not** entire frames

Compositing: X

- Compositing is “active”
  - Compositor must fetch all data
    - Required when:
      - Resizing
      - Showing/Hiding
  - Compositor must “know” when to fetch
- Pixmap fetching
  - Requires protocol transmission
  - Introduces latency

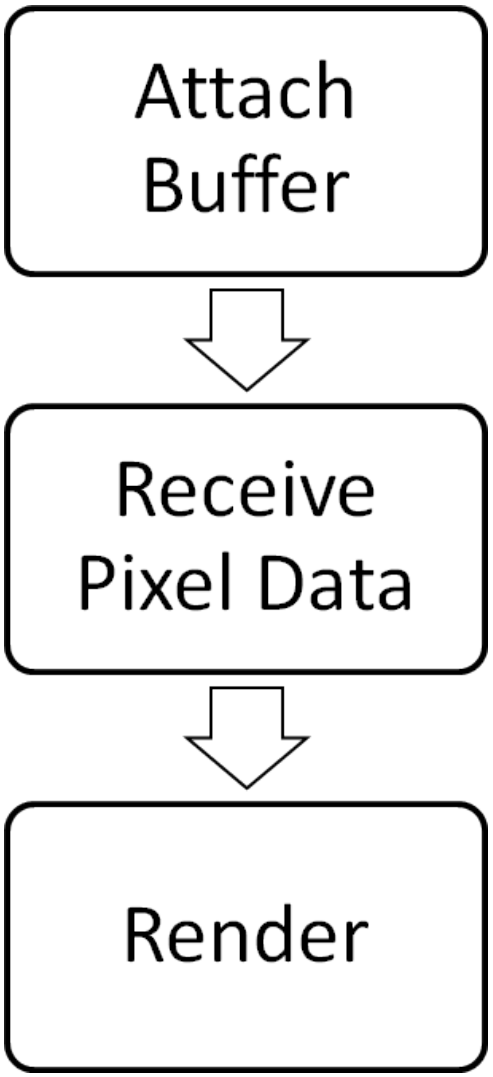


Display Systems: Wayland

- Compositing **required**
  - All Window Managers are Compositing Window Managers
- Windows cannot query other windows
  - Special mechanism for requesting screenshots
  - Input handled by WM only
- Borders “encouraged” to be drawn by clients
  - Not required, however
- Compositor = Display System

Compositing: Wayland

- Compositing is “passive”
  - Compositor *receives* pixel data
  - Compositor is notified of pixmap (buffer) invalidation and creation
- Compositor communicates directly with clients
  - No display server in between
- Pixmap receiving
  - Occurs for entire frames
  - No tearing!



Evolution of Compositing: E17

- Began in 2010
- Supported basic compositing
  - Effects: fading, shadows
  - Some general issues
    - Longstanding bug rendering windows showing/hiding repeatedly
    - Shaped windows :(
- Could be disabled
  - Unable to use compositor canvas for drawing
  - All objects required X window to be shown

E17 Compositing: Advantages

- Compositing
  - Fade in/out effects
- Generally smoother rendering

E17 Compositing: Disadvantages

- Gadgets required X windows
  - Serious slowdowns during startup due to buggy video driver texture resizing
- Hard to track/fix bugs
  - Compositing not universally used across user base
- Wayland support basically impossible

E17 Compositing: Lessons Learned

- Always composite
  - Makes troubleshooting + bug fixing easier
- Use compositor window/canvas for as much drawing as possible
  - Gadget resizing no longer an issue
- Integrate more effectively with Window Manager

Evolution of Compositing: E18

- Began in 2013
- Compositing no longer optional
  - Gadgets, borders, menus all drawn to compositor canvas
  - Greatly reduced compositing overhead
  - Improved accuracy
    - Shaped windows :(
- Effects improved
  - Desk flip animations now composited, smooth
  - Sparklebear!

E18 Compositing: Advantages

- Many X windows (compositor sources) removed
- Performance improvements
  - Shading windows no longer requires protocol/client resizes
  - Switching virtual desktops no longer requires moving/resizing clients
- Much easier to integrate Wayland client support

E18 Compositing: Demos

- Sparklebear!
- Teamwork!
- Entry-level window effects!

E18 Compositing: Disadvantages

- Clunky compositor architecture/API
  - Mostly the same as E17
    - Compositor “reacts” to window manager events instead of being integrated with window management
  - Difficult to use/manipulate composited source images
- Still not ideal for Wayland
  - All client management relies on having X window
  - X-less Wayland compositor required a separate binary to work around X dependencies

Evolution of Compositing: E19

- Began in late July
- Compositor and Window Manager rewrite
  - Window Manager manages compositor object
    - Rendering triggered as necessary based on display server
  - API stabilized (hopefully)
    - Uses regular canvas (evas) object API
  - Better abstractions for client pixmap management
- Full Wayland support from start
  - Standalone Wayland compositing since August

E19 Compositing: Advantages

- Code is more logical
  - Easier follow interactions
  - Small speed improvements / reduced latency in some cases
- Greatly improved API
  - Much easier to write cool effects
- Code de-duplication
  - Wayland and X rendering use same API

E19 Compositing: Demos

- Deskmirror API
  - Desksanity!
  - Pager16!

E19 Compositing: Disadvantages

- Most E infrastructure still built around previous compositing paradigm
  - Not using compositor canvas as effectively
  - Lots of workarounds for separate windows
    - Desktop gadgets :(
- Required a complete rewrite
  - Huge amounts of code changed
  - Lots of work still remaining

A Word On Failures

- Compbench

Showcase Demo

- Project Burrito
  - Enchilada

Architecture Tips

- Make window borders configurable
- Unify as much actual compositing code as possible
- Allow easy access to compositor sources
  - Also make using them easy + sensible
- Integrate compositor with window manager
  - Better effects

Wayland/X Compositor Architecture By Example: Enlightenment DR19

- **Blog:** http://e19releasemanager.wordpress.com
- **Contact:** zmike@enlightenment.org
- **E18 Release Date:** TBD

- **Source:** http://git.enlightenment.org/core/enlightenment.git/
  - “devs/discomfitor/e19” branch