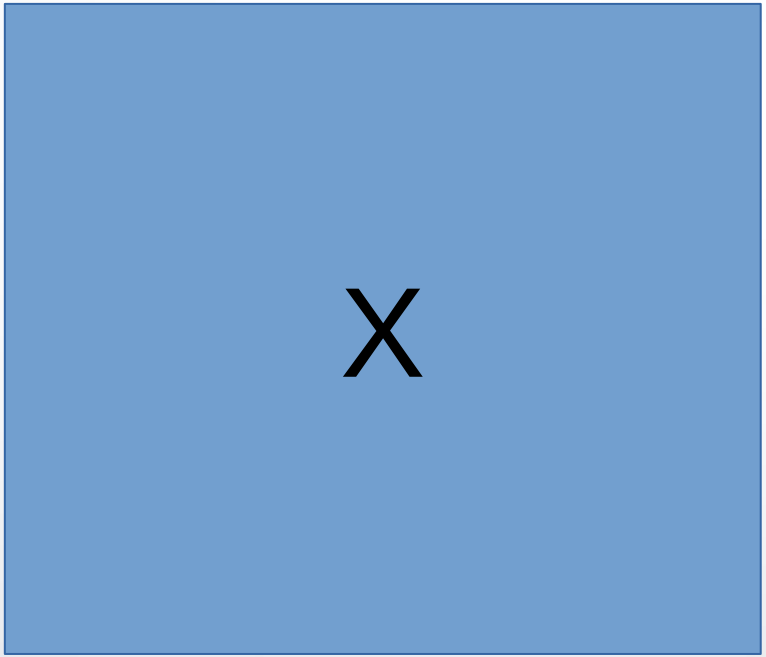
Utilizing Wayland to Improve Performance on Tizen
运用Wayland提升在Tizen平台的表现

Derek Foreman / Christopher Michael

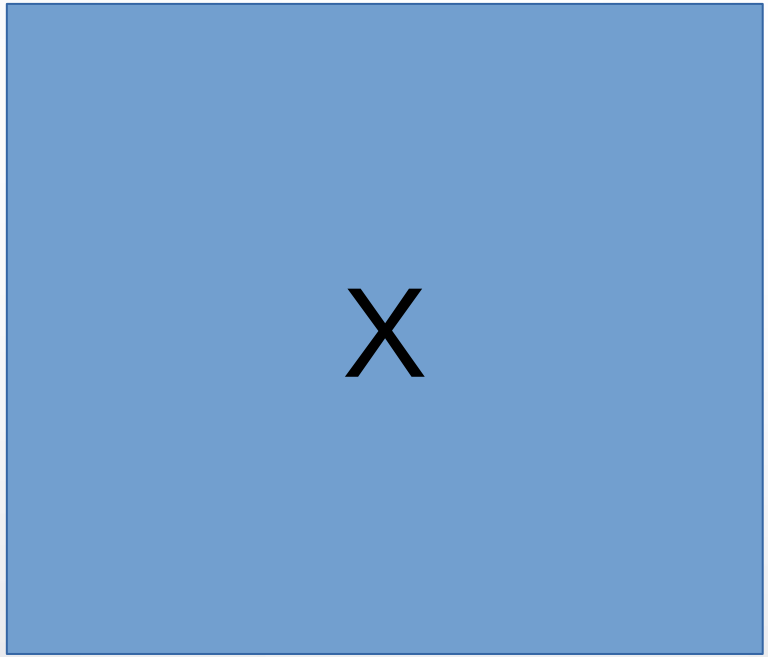
Samsung

Overview

X
Wayland (and Weston)
Under the hood differences
EFL on X
EFL on Wayland
Enlightenment: X vs Wayland



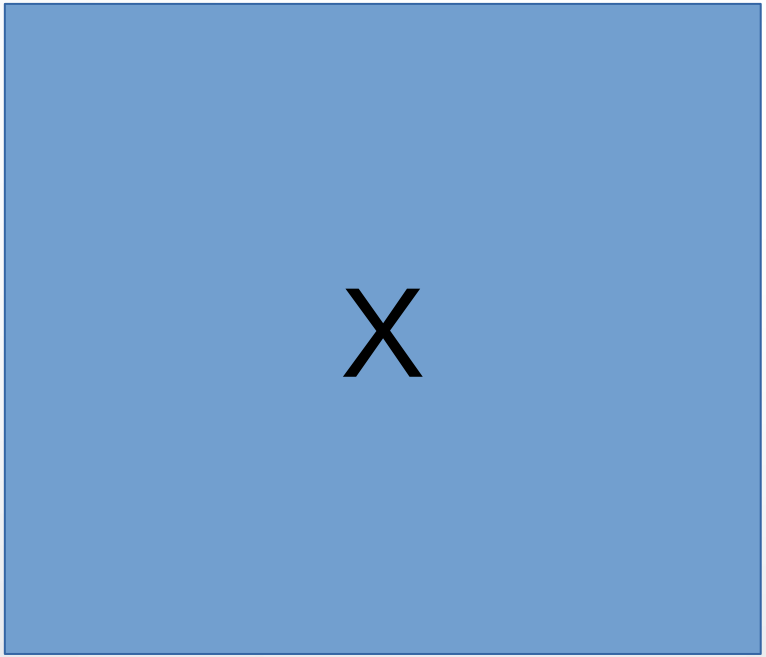
Missing features:



Missing features:

Compositing

X

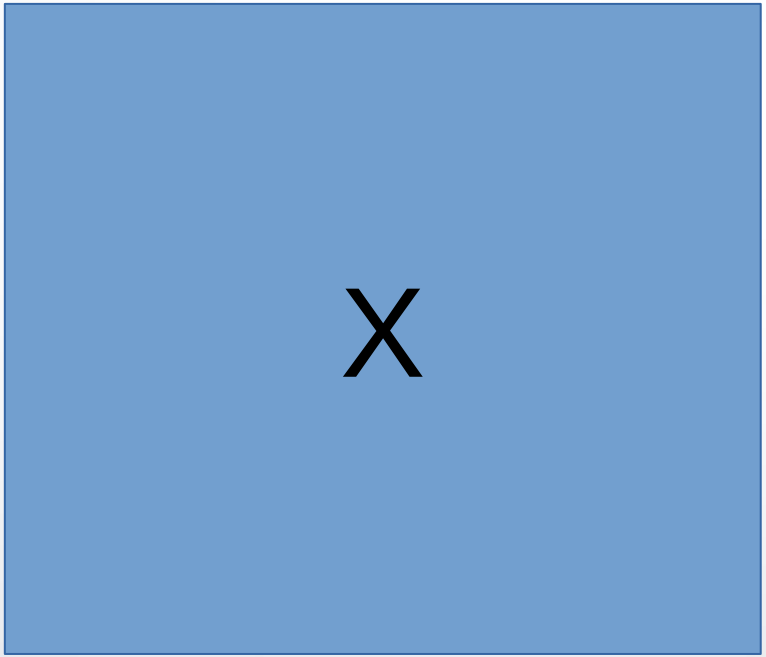


Missing features:

Compositing

3D Graphics

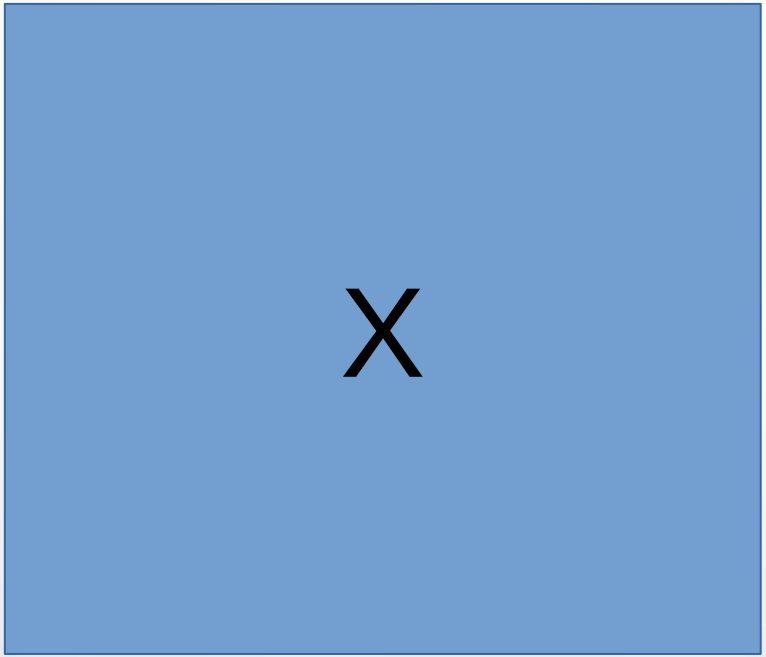
X



Missing features:

- Compositing
- 3D Graphics
- Video

X



Missing features:

- Compositing
- 3D Graphics
- Video
- Moving windows around

X – with Extensions

Missing features:

- Compositing
- 3D Graphics
- ~~Video~~
- Moving windows around

X – with Extensions and Support

Missing features:

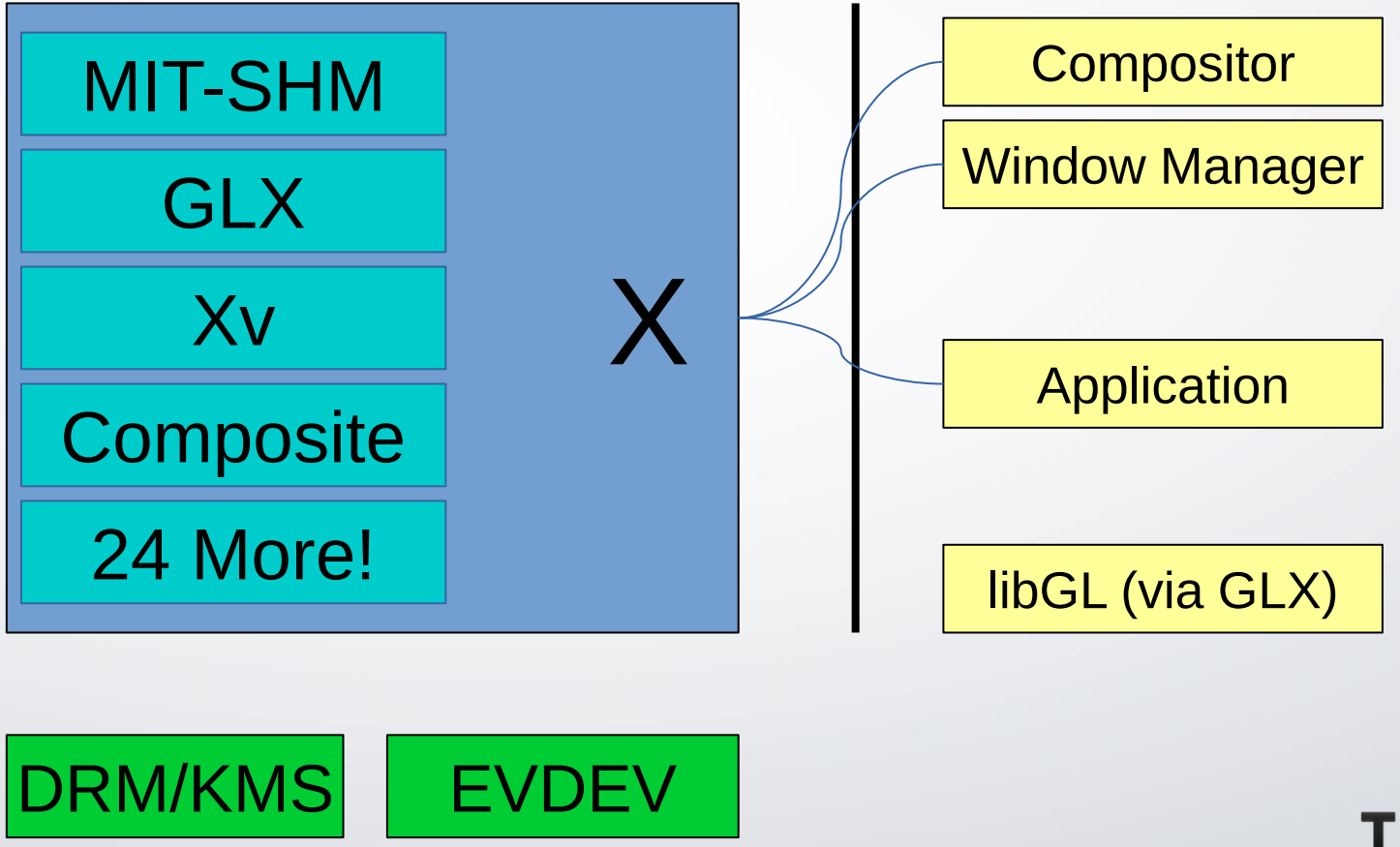
- Compositing
- ~~3D Graphics~~
- ~~Video~~
- Moving windows around

libGL (via GLX)

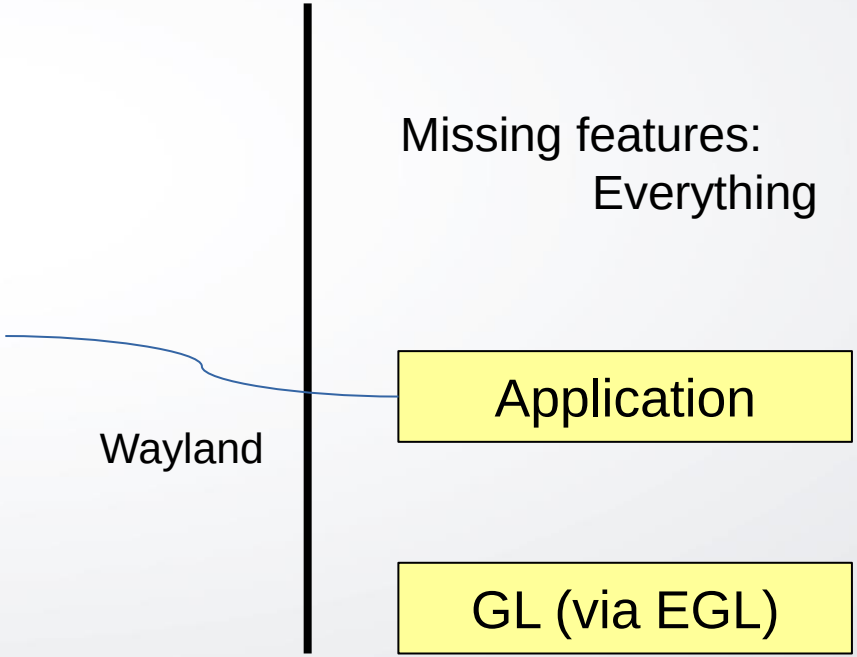
DRM/KMS

EVDEV

X – with Extensions and Support and Clients



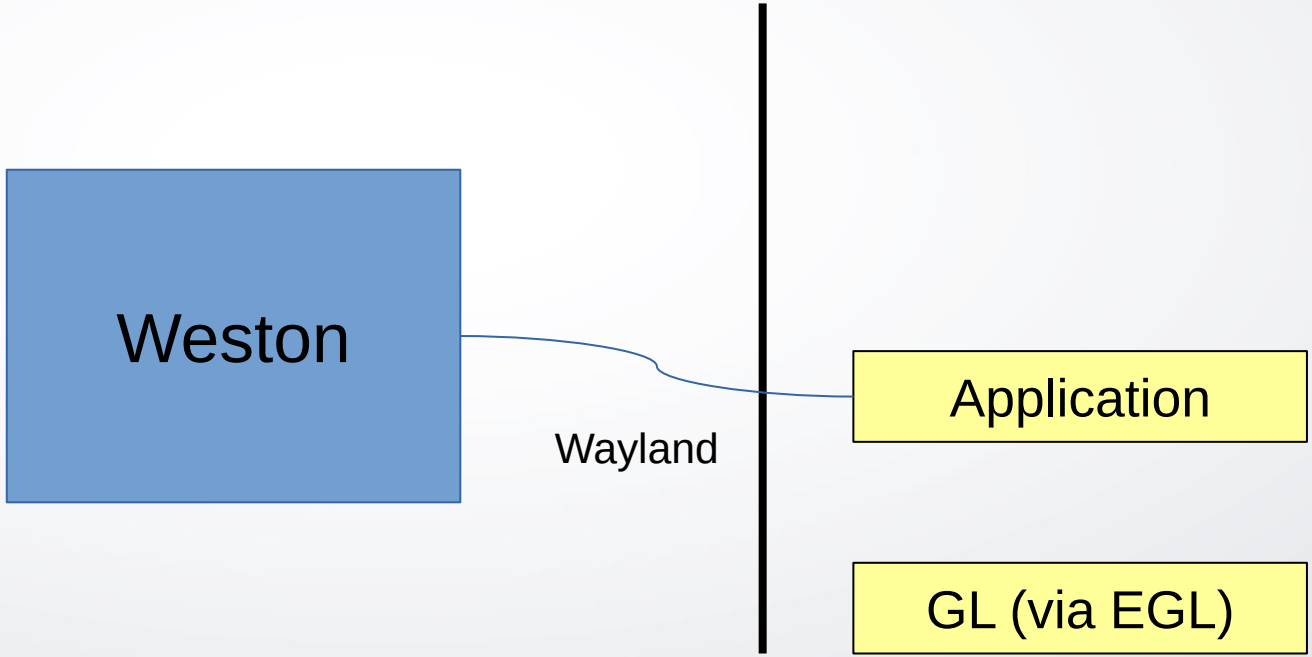
Wayland



DRM/KMS

EVDEV

Wayland and Weston



DRM/KMS

EVDEV

Under the Hood: Architecture

X

X Server

- Talks to hardware
- Does rendering
- Sends events

Under the Hood: Architecture

- X
- Window Manager
  - Moves windows
  - Handles some state
  - Does rendering
- X Server
  - Talks to hardware
  - Does rendering
  - Sends events

Under the Hood: Architecture

- X
- Compositor
  - Ensures tear free display
  - Wobbly Windows
  - Does rendering
- Window Manager
  - Moves windows
  - Handles some state
  - Does rendering
- X Server
  - Talks to hardware
  - Does rendering

Under the Hood: Architecture

X

Compositor *

- Ensures tear free display
- Wobbly Windows
- Does rendering

Window Manager *

- Moves windows
- Handles some state
- Does rendering

X Server

- Talks to hardware
- Does rendering
- Sends events

* May or may not be the same program

Under the Hood: Architecture

X

X Server

Window Manager

Compositor

Wayland

Compositor

Under the Hood: Architecture

X

X Server
Window Manager
Compositor

Wayland

Compositor
(Does all that stuff)

Under the Hood: Input

X

X Server
Window Manager
Compositor

Wayland

Compositor

Under X, compositor not involved with input

Benefits to Tizen:

Compositor puts windows anywhere
any orientation
input works

Under the Hood: Wayland multi-seat

Collection of input devices:

- Keyboard
- Mouse
- Touch

Under the Hood: Wayland multi-seat

Collection of input devices:

- Keyboard
- Mouse
- Touch

Associated state:

- Cut and paste
- Focus
- Interactive moves and resizes

Benefits to Tizen:

Effective sharing of displays
New collaborative possibilities

Under the Hood: Drawing API

X

- Rectangles
- Lines
- Arcs
- Filled Polygons
- Filled Arcs
- Fonts
- Image buffers

Wayland

- Image buffers

Under the Hood: Drawing API

X

Junk

Image buffers

Wayland

Image buffers

Benefits to Tizen:

Less completely useless code
Saves storage space and memory

Under the Hood: Drawing API

X

Client draws stuff
Client sends buffer
X server does stuff
X server alerts compositor
Compositor renders

Wayland

Client draws stuff
Client commits buffer handle
Compositor renders
Compositor releases buffers

Under the Hood: Drawing API

X

Client draws stuff
Client sends buffer
X server does stuff
X server alerts compositor
Compositor renders

Wayland

Client draws stuff
Client commits buffer handle
Compositor renders
Compositor releases buffers

X server is single threaded

Under the Hood: Drawing API

X

Client draws stuff
Client sends buffer
X server does stuff
X server alerts compositor
Compositor renders

Wayland

Client draws stuff
Client commits buffer handle
Compositor renders
Compositor releases buffers

X server is single threaded
Handles client storage

Under the Hood: Drawing API

X

Client draws stuff
Client sends buffer
X server does stuff
X server alerts compositor
Compositor renders

Wayland

Client draws stuff
Client commits buffer handle
Compositor renders
Compositor releases buffers

Using poorly synchronized client data

Under the Hood: Drawing API

X

Client draws stuff
Client sends buffer
X server does stuff
X server alerts compositor
Compositor renders

Wayland

Client draws stuff
Client commits buffer handle
Compositor renders
Compositor releases buffers

Using poorly synchronized client data
Tear-free display possible… sometimes

Benefits to Tizen:

"Every frame is perfect"
Easier benchmarking and profiling
Lower impact from a bad application

Under the Hood: Wayland buffer properties

- Buffer scale
  - Integer multiple
  - Compensates for display DPI difference

Under the Hood: Wayland buffer properties

- Buffer scale
  - Integer multiple
  - Compensates for display DPI difference
- Transform
  - 90 degree rotations
  - Application can render in display orientation

Benefits to Tizen:

Matching transforms may save copy
Scale allows for DPI awareness

Benefits to Tizen:

Matching transforms may save copy
Scale allows for DPI awareness

Less cpu usage, better images

Under the Hood: Frame callbacks

Wayland provides a frame callback to clients

- Means “commit now to hit vblank”
- Doesn’t happen for hidden apps
- Or sleeping displays

Benefits to Tizen:

Reliable timing mechanism
Simple, low latency

Benefits to Tizen:

Reliable timing mechanism
Simple, low latency

Smooth and responsive animations

Under the Hood: Video Handling

X

Xv
OpenGL
Image
Alpha Hole

Wayland

Buffers (fourcc)

Under the Hood: Video Handling

X

Xv
OpenGL
Image
Alpha Hole

Wayland

Buffers (fourcc)

“Descriptive, not prescriptive”

Benefits to Tizen:

Compositor can leverage hardware planes

Less cpu usage
Better quality video
Longer battery life

Under the Hood: Video Handling

X

- Xv
- OpenGL
- Image
- Alpha Hole

Wayland

- Buffers (fourcc)
- DMA-Buf (soon)

Under the Hood: Video Handling

X

Xv
OpenGL
Image
Alpha Hole

Wayland

Buffers (fourcc)
DMA-Buf (soon)

Zero copy!

Under the Hood: Nesting Structures

X	Wayland
Window reparenting	Subsurfaces

Under the Hood: Nesting Structures

X

Window reparenting

- Parent moves are atomic

Wayland

Subsurfaces

Under the Hood: Nesting Structures

X

- Window reparenting
  - Parent moves are atomic
  - Moving children isn't

Wayland

- Subsurfaces
  - Double buffered state
  - Atomic via parent commit

Under the Hood: Nesting Structures

X

Window reparenting

- Parent moves are atomic
- Moving children isn't

Wayland

Subsurfaces

- Double buffered state
- Atomic via parent commit

Scroll a browser window with lots of video elements...

Benefits to Tizen:

Coherent motion of complicated layouts
Again, “Every frame is perfect”

Applications have:

No control over own window position
No access to other client's buffers
No unfiltered access to input

Applications have:

- No control over own window position
  - No focus stealing
- No access to other client's buffers
  - No color picker
- No unfiltered access to input
  - No xeyes

Benefits to Tizen:

Better protection from malicious software

Under the Hood: Wayland protocol extensions

Easy to add private extensions

Under the Hood: Wayland protocol extensions

Easy to add private extensions

- Official protocol developed as Weston extensions

Under the Hood: Wayland protocol extensions

Easy to add private extensions

- Gaussian blur?
- Window orientation hints?

Benefits to Tizen:

Eye candy
Rapid innovation

Under the Hood: Portability

KMS works?
Weston works.

Under the Hood: Portability

KMS works?
Weston works.

Don’t have to write Xv driver!

Under the Hood: Portability

KMS works?
Weston works.*

*(GL has some additional requirements)

Benefits to Tizen:

Quick bring-up on new platforms
Easy adoption

See Mauro Chehab's talk "Bringing Tizen to a Raspberry PI 2 Near You!"

That's great – How do we use it?

Wayland and X share approximately 0% common API

That's great – How do we use it?

Wayland and X share approximately 0% common API

Rewrite all your applications?

That's great – How do we use it?

Wayland and X share approximately 0% common API

Rewrite all your applications?

Just kidding! Use EFL.

Enlightenment Foundation Libraries (EFL)

Data structures (lists, hash tables, etc)
Main loop event, I/O, timing core
Event queue and call handling
Canvas scene graph and rendering
Widgets/controls (buttons, sliders, etc)
Input method framework
Video & audio codec playback
IPC, D-Bus, network connectivity

EFL Applications on X

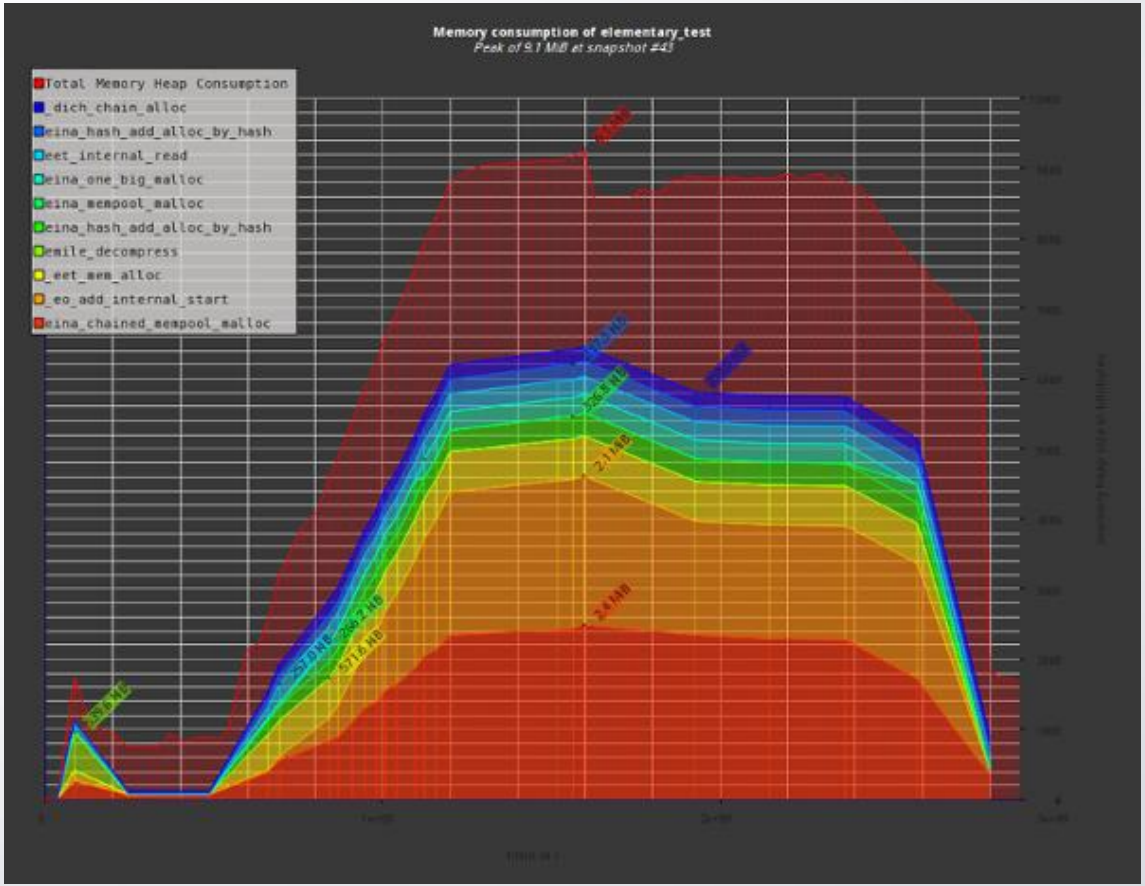
Startup time

- “time elementary_test”:
  - Real: 0m1.645s
  - User: 0m0.683s
  - System: 0m0.017s

Memory Usage: 29.8 Mb

EFL Applications on X (cont)

Elementary Test
Peak memory: 9.1 MB



EFL Applications on Wayland

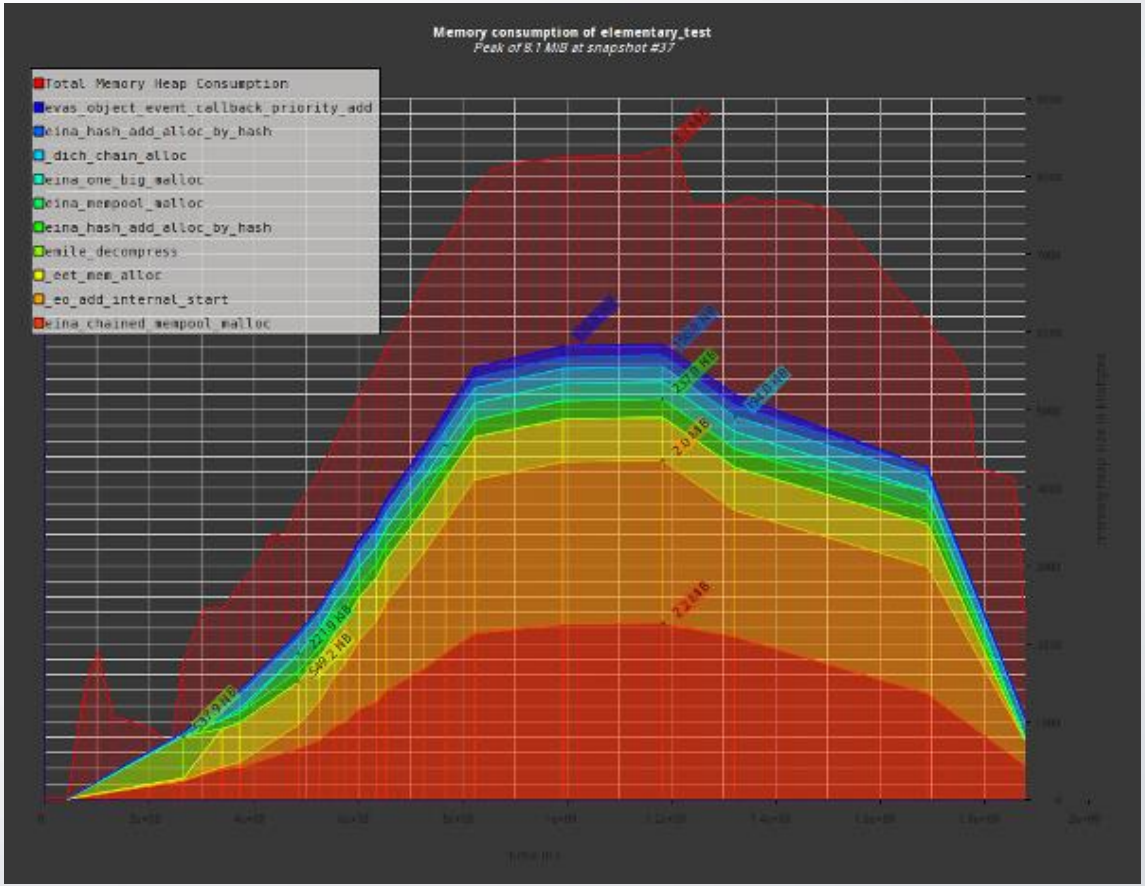
Startup time

- “time elementary_test”:
  - Real: 0m1.227s
  - User: 0m0.420s
  - System: 0m0.003s

Memory Usage: 18.2 Mb

EFL Applications on Wayland (cont)

Elementary Test
Peak memory: 8.1 MB



EFL Applications

- Utilizing Wayland:
  - Decreased Startup Time:
    - 0.418s average per application
  - Decreased Memory Usage:
    - 11.6 Mb average per application
  - Similar functionality

Enlightenment: X vs Wayland

- Startup time not measured
- Enlightenment Memory Usage:
  - In X11: 103.5 Mb
    - X Server: 62 Mb
    - Window Manager: 41.5 Mb
  - Wayland: 54.9 Mb
  - Average Memory Savings: 48.6 Mb
- Similar functionality