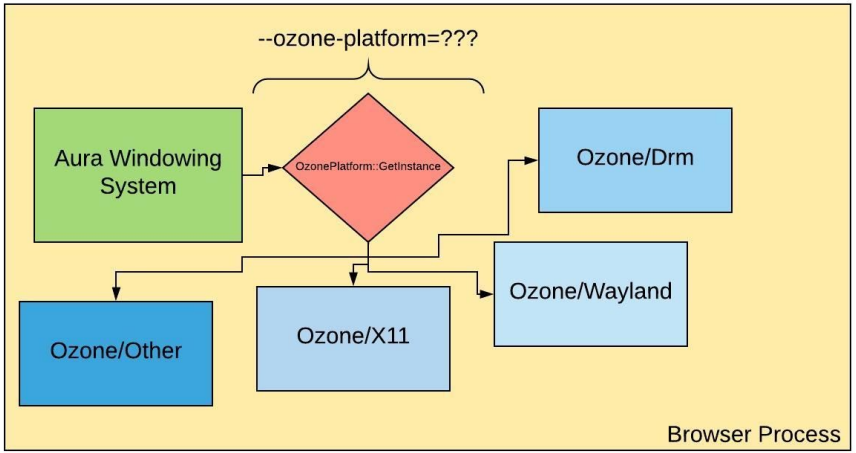
Essential parts to implement own Ozone backend

Web Engines Hackfest 2021
Maksim Sisov <msisov@iglaia.com>

What is Ozone (again)?

- Abstraction layer beneath the Aura window system.
  - wrapper for the platform
  - provides interfaces for everything
- Chosen at run time
- Many platforms in one binary



igalia

Example: Add Ozone/GTK

- Learn by implementing!
- Goal:
  - Add Ozone/Gtk backend that Chromium can draw to.
- Steps:
  - GN:
    - Entry points: add Ozone/Gtk to GN
  - Browser side. Implement
    - OzonePlatform,
    - PlatformWindow, PlatformScreen.
  - GPU side. Implement
    - SurfaceFactoryOzone,
    - GLOzone (HW accelerated path), SurfaceOzoneCanvas (SW path).
  - Stub gfx::ClientNativePixmapFactory.

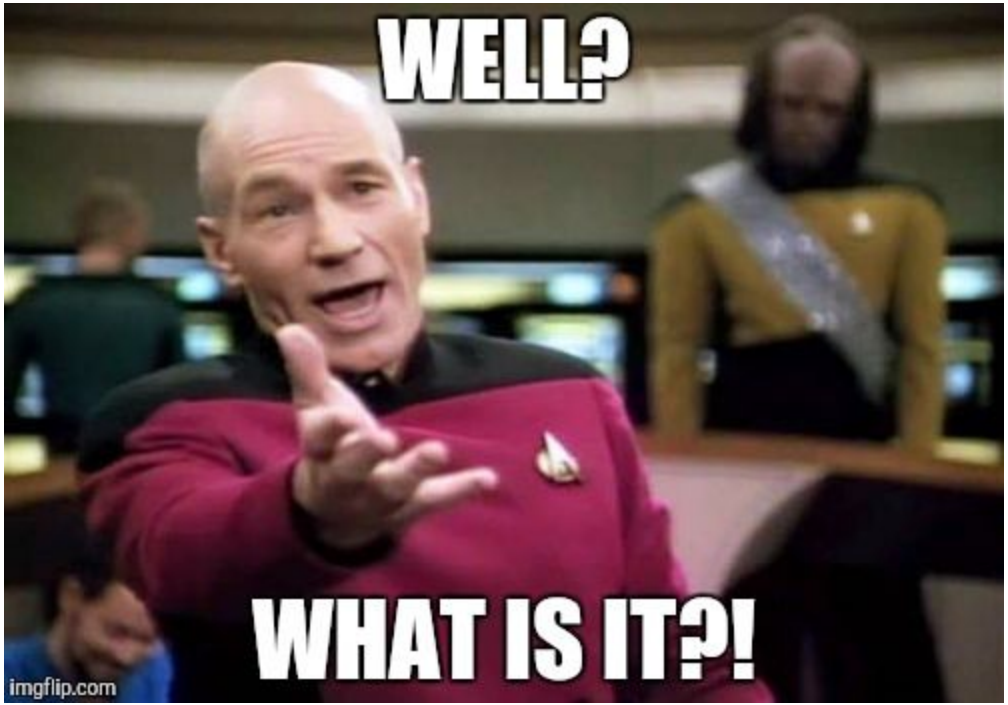
igalia

The Entry Points

Ozone and GN

- Python script that generates constructor list

```
10  namespace ui {
11
12  OzonePlatform* CreateOzonePlatformX11();
13  OzonePlatform* CreateOzonePlatformGtk();
14  OzonePlatform* CreateOzonePlatformHeadless();
15  OzonePlatform* CreateOzonePlatformWayland();
16
17  ClientNativePixmapFactory* CreateClientNativePixmapFactoryX11();
18  ClientNativePixmapFactory* CreateClientNativePixmapFactoryGtk();
19  ClientNativePixmapFactory* CreateClientNativePixmapFactoryHeadless();
20  ClientNativePixmapFactory* CreateClientNativePixmapFactoryWayland();
21
22  }  // namespace ui
```



igalia

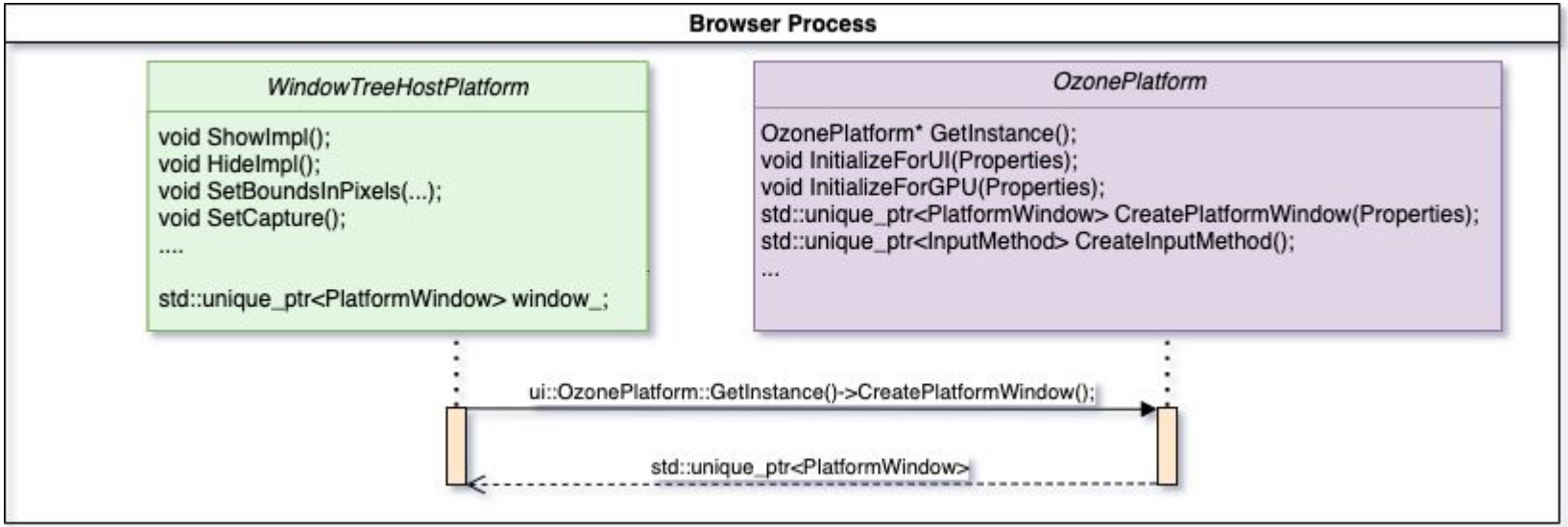
Ozone and GN

- Modify //build/config/ozone_extra.gni
- Create //ui/ozone/platform/gtk

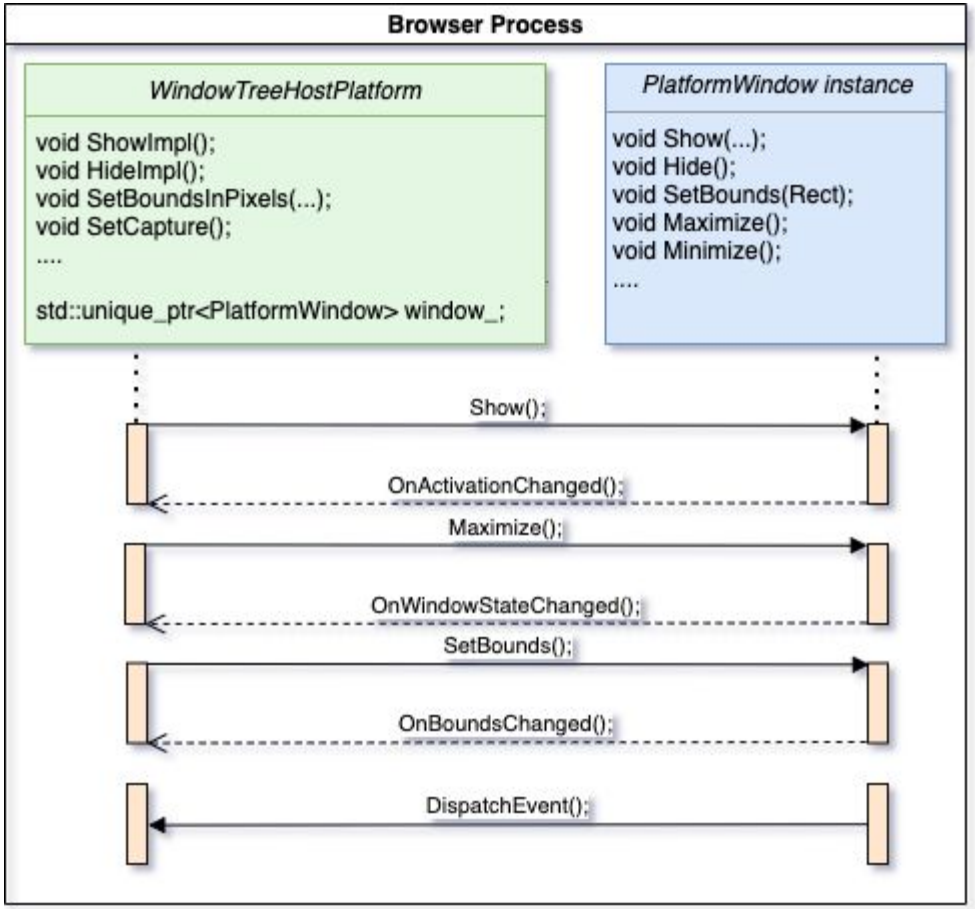
```
# Copyright 2016 The Chromium Authors. All rights reserved.
# Use of this source code is governed by a BSD-style license that can be
# found in the LICENSE file.

# This list contains the name of external platforms that are passed to the
# --ozone-platform command line argument or used for the ozone_platform build
# config. For example ozone_external_platforms = [ "foo1", "foo2", ... ]
ozone_external_platforms = [ "gtk" ]

# This list contains dependencies for external platforms. Typically, the Ozone
# implementations are placed into ui/ozone/platform/ and so this will look
# something like:
# ozone_external_platform_deps = [ "platform/foo1", "platform/foo_2", ... ]
ozone_external_platform_deps = [ "platform/gtk" ]

# If a platform has unit tests, the corresponding source_set can be listed here
# so that they get included into ozone_unittests.
# ozone_external_platform_test_deps = [ "platform/foo1:foo1_unitests", ... ]
ozone_external_platform_test_deps = []

# If a platform has test support files for ui, the corresponding source_set can
# be listed here so that they get included into ui_test_support.
# ozone_external_platform_ui_test_support_deps = [ "platform/foo1:ui_test_support", ... ]
ozone_external_platform_ui_test_support_deps = []
```

Implement OzonePlatformGtk and ...

Browser side



igalia


Browser Process
WindowTreeHostPlatform
void ShowImpl();
void HideImpl();
void SetBoundsInPixels(...);
void SetCapture();
....
std::unique_ptr<PlatformWindow> window_;
PlatformWindow instance
void Show(...);
void Hide();
void SetBounds(Rect);
void Maximize();
void Minimize();
....
Show();
OnActivationChanged();
Maximize();
OnWindowStateChanged();
SetBounds();
OnBoundsChanged();
DispatchEvent();

igalia

OzonePlatformGtk

- Functions to implement:
  - ui::SurfaceFactoryOzone* GetSurfaceFactoryOzone();
  - std::unique_ptr<PlatformWindow> CreatePlatformWindow(...);
  - std::unique_ptr<PlatformScreen> CreateScreen();
  - void InitializeUI(const InitParams& params);
  - void InitializeGPU(const InitParams& params);
  - const PlatformProperties& GetPlatformProperties();
- All the other virtual functions will return nullptr except:
  - CursorFactory* GetCursorFactory(); - Will return default cursor factory.
  - GpuPlatformSupportHost* GetGpuPlatformSupportHost(); - Will return stub.
  - std::unique_ptr<InputMethod> CreateInputMethod(...); - Wll return default input method.

GtkOzoneWindow

- Important points:
  - Create native window - *gtk_window_new(GTK_WINDOW_TOPLEVEL).*
  - Initialize window and return gfx::AcceleratedWidget for that window
    - gfx::AcceleratedWidget is a Chromium's internal representation of widget aka handle for a native window.
      - X11 => XWindow
      - Wayland => own counter
  - Return widget by calling *delegate_->OnAcceleratedWidgetAvailable(widget_);*
  - Also:
    - Implement Show(), SetBounds(), GetBounds()
    - And add event listeners to get events from server and send to delegate.
    - All the other pure virtual methods can be left unimplemented.

igalia

GtkScreen

- Provide basic implementation that
  - returns a display for a widget
  - returns widget at screen point.
- For demo, a primary display was hardcoded with the parameters checked from the display settings. The correct solution would be fetching the list of displays as it is done in Ozone/X11/Wayland.

igalia

Are we done?

igalia

GPU side

- Implement SurfaceFactoryOzone
  - returns supported GL implementations (HW acceleration or swiftshader)
    - They can be -

```
enum GLImplementation {
  kGLImplementationNone = 0,
  kGLImplementationDesktopGL = 1,
  kGLImplementationDesktopGLCoreProfile = 2,
  kGLImplementationSwiftShaderGL = 3,
  kGLImplementationAppleGL = 4,
  kGLImplementationEGLGLES2 = 5,  // Native EGL/GLES2
  kGLImplementationMockGL = 6,
  kGLImplementationStubGL = 7,
  kGLImplementationDisabled = 8,
  kGLImplementationEGLANGLE = 9,  // EGL/GL implemented using ANGLE
  kMaxValue = kGLImplementationEGLANGLE,
};
```

GPU side

- For demo, we will use SW path instead.

```
// Handles GL initialization and surface/context creation for GTK.
class GtkSurfaceFactory : public SurfaceFactoryOzone {
 public:
  GtkSurfaceFactory();
  ~GtkSurfaceFactory() override;

  // SurfaceFactoryOzone:
  std::vector<gl::GLImplementation> GetAllowedGLImplementations() override;
  GLOzone* GetGLOzone(const gl::GLImplementationParts& implementation) override;
  std::unique_ptr<SurfaceOzoneCanvas> CreateCanvasForWidget(
      gfx::AcceleratedWidget widget) override;

 private:
  std::unique_ptr<GLOzone> egl_implementation_;

  DISALLOW_COPY_AND_ASSIGN(GtkSurfaceFactory);
};
```

GPU side - SurfaceOzoneCanvas

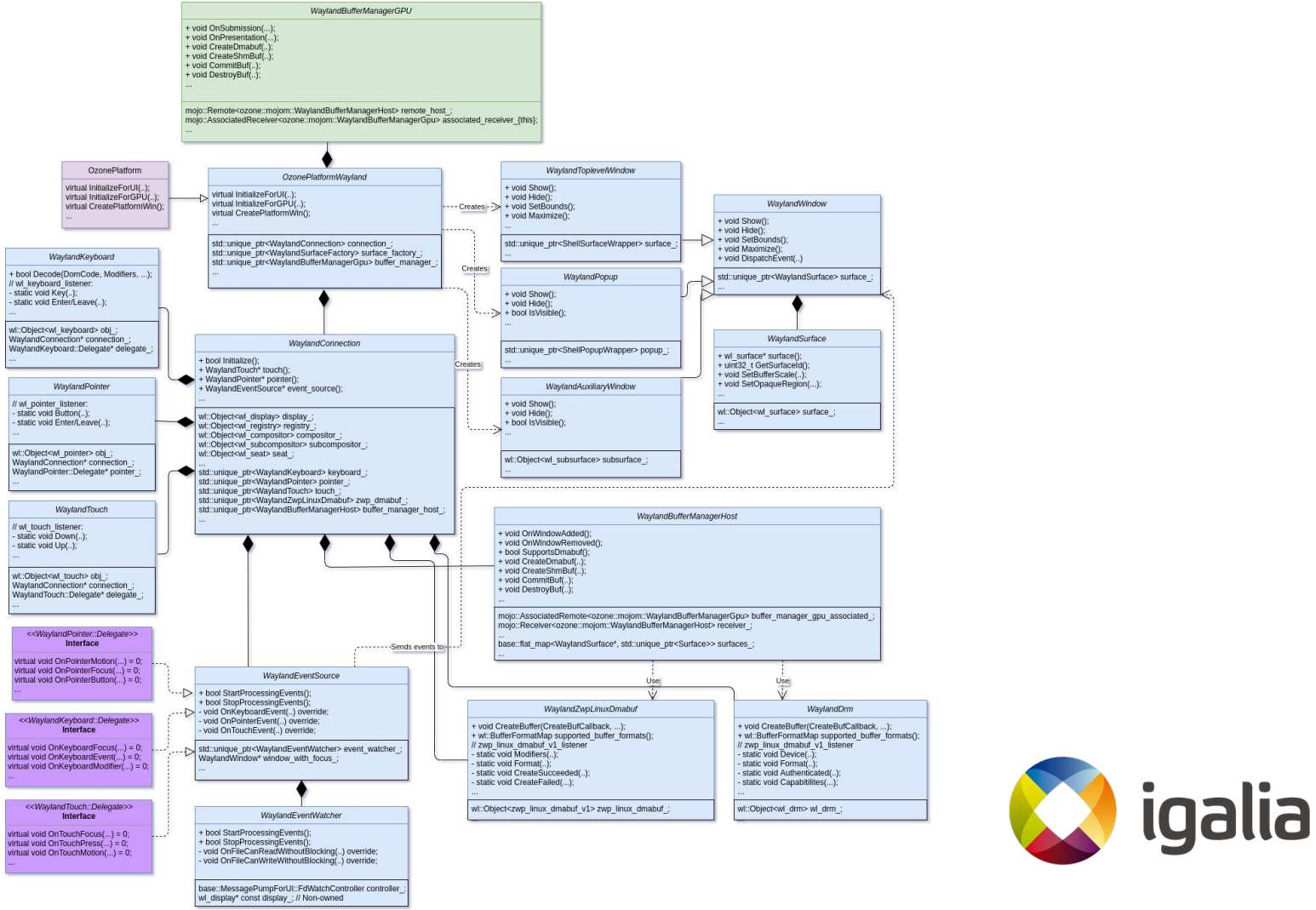
- Used when HW accelerated path is not available.
- Returns SkCanvas where Chromium draws to
  - When canvas has data written, PresentCanvas is called
  - When ResizeCanvas is called, SkCanvas becomes invalid.
- For GTK, gtk drawing area with cairo image surface was used -
  - Create gtk frame,
  - Create drawing area and add to frame
  - Create cairo image surface
  - Create SkSurface from cairo image surface
  - Return SkCanvas from SkSurface
- Once PresentCanvas is called
  - Create cairo region, get gdk drawing context
  - get cairo context, paint from the cairo image surface into the cairo context

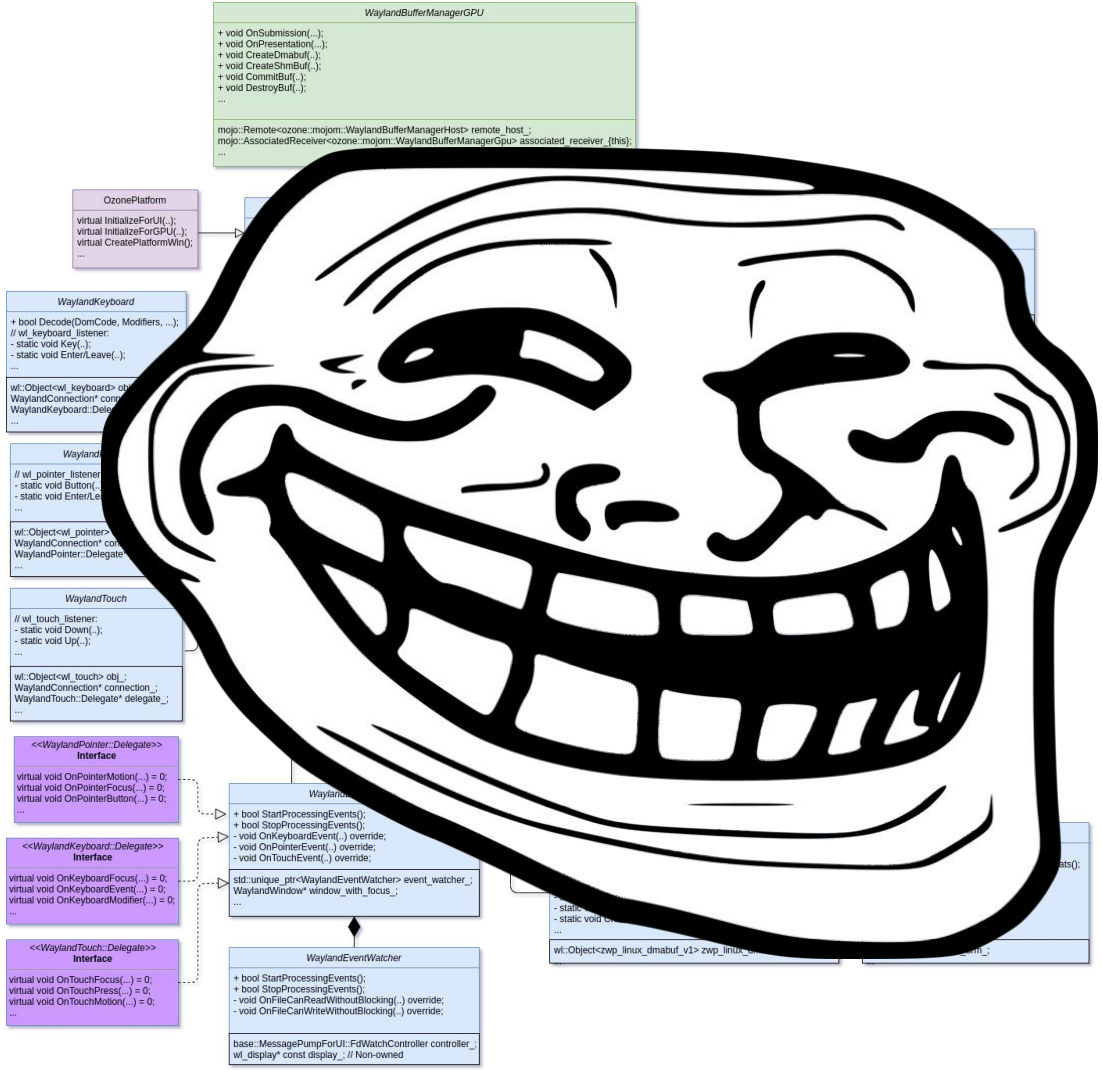
igalia

Are we done now?

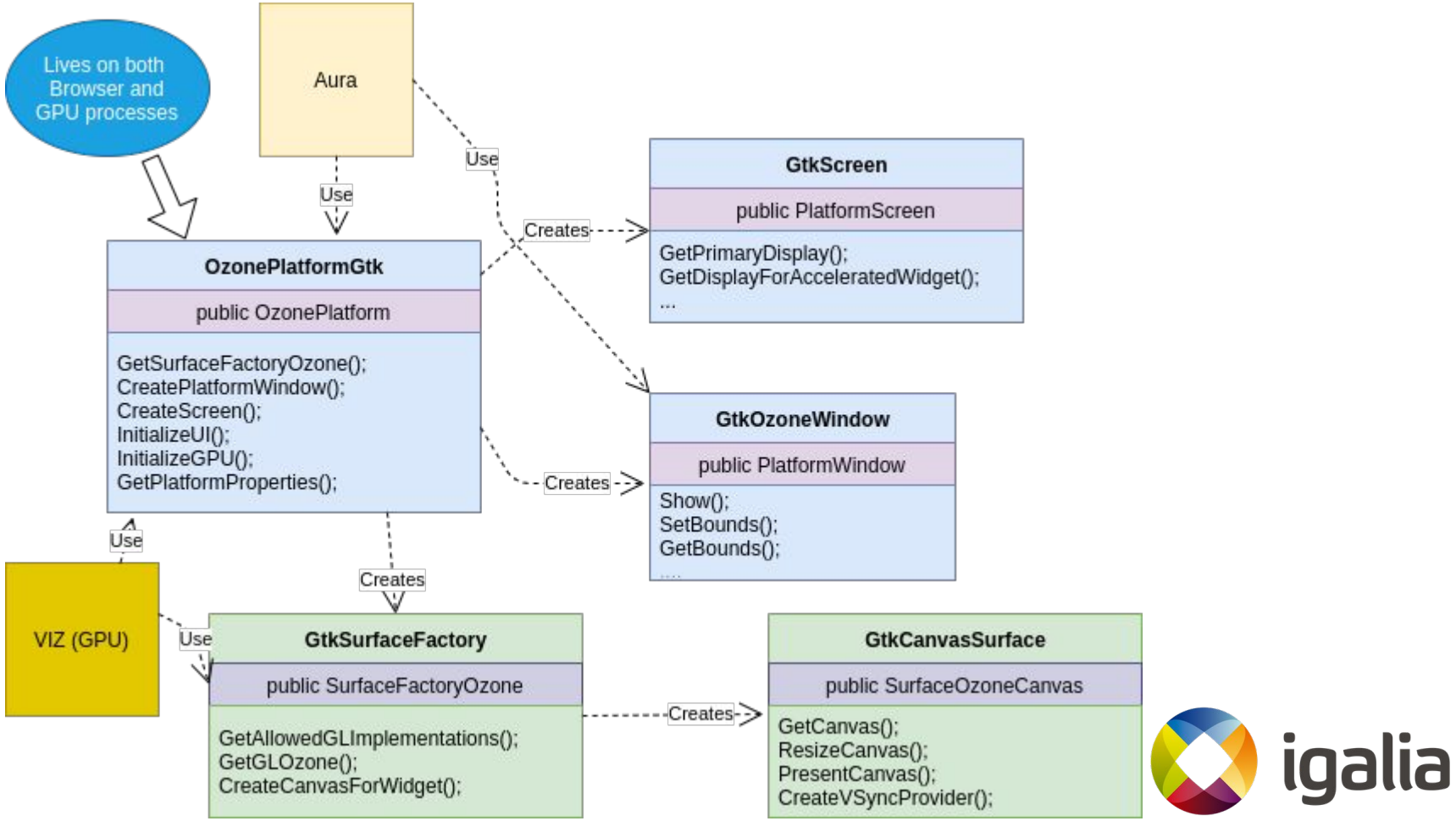
igalia

A big picture


WaylandBufferManagerGPU
+ void OnSubmission(...);
+ void OnPresentation(...);
+ void CreateDmabuf(..);
+ void CreateShmBuf(..);
+ void CommitBuf(..);
+ void DestroyBuf(..);
...
mojo::Remote<ozone::mojom::WaylandBufferManagerHost> remote_host_;
mojo::AssociatedReceiver<ozone::mojom::WaylandBufferManagerGpu> associated_receiver_{this};
...
OzonePlatform
virtual InitializeForUI(..);
virtual InitializeForGPU(..);
virtual CreatePlatformWin();
...
OzonePlatformWayland
virtual InitializeForUI(..);
virtual InitializeForGPU(..);
virtual CreatePlatformWin();
...
std::unique_ptr<WaylandConnection> connection_;
std::unique_ptr<WaylandSurfaceFactory> surface_factory_;
std::unique_ptr<WaylandBufferManagerGpu> buffer_manager_;
...
Creates
WaylandToplevelWindow
+ void Show();
+ void Hide();
+ void SetBounds();
+ void Maximize();
...
std::unique_ptr<ShellSurfaceWrapper> surface_;
...
WaylandWindow
+ void Show();
+ void Hide();
+ void SetBounds();
+ void Maximize();
+ void DispatchEvent(..)
std::unique_ptr<WaylandSurface> surface_;
...
WaylandPopup
+ void Show();
+ void Hide();
+ bool IsVisible();
...
std::unique_ptr<ShellPopupWrapper> popup_;
...
WaylandAuxiliaryWindow
+ void Show();
+ void Hide();
+ bool IsVisible();
...
wl::Object<wl_subsurface> subsurface_;
...
WaylandSurface
+ wl_surface* surface();
+ uint32_t GetSurfaceId();
+ void SetBufferScale(..);
+ void SetOpaqueRegion(...);
...
wl::Object<wl_surface> surface_;
...
WaylandKeyboard
+ bool Decode(DomCode, Modifiers, ...);
// wl_keyboard_listener:
- static void Key(..);
- static void Enter/Leave(..);
...
wl::Object<wl_keyboard> obj_;
WaylandConnection* connection_;
WaylandKeyboard::Delegate* delegate_;
...
WaylandConnection
+ bool Initialize();
+ WaylandTouch* touch();
+ WaylandPointer* pointer();
+ WaylandEventSource* event_source();
...
wl::Object<wl_display> display_;
wl::Object<wl_registry> registry_;
wl::Object<wl_compositor> compositor_;
wl::Object<wl_subcompositor> subcompositor_;
wl::Object<wl_seat> seat_;
...
std::unique_ptr<WaylandKeyboard> keyboard_;
std::unique_ptr<WaylandPointer> pointer_;
std::unique_ptr<WaylandTouch> touch_;
std::unique_ptr<WaylandZwpLinuxDmabuf> zwp_dmabuf_;
std::unique_ptr<WaylandBufferManagerHost> buffer_manager_host_;
...
WaylandPointer
// wl_pointer_listener:
- static void Button(..);
- static void Enter/Leave(..);
...
wl::Object<wl_pointer> obj_;
WaylandConnection* connection_;
WaylandPointer::Delegate* pointer_;
...
WaylandTouch
// wl_touch_listener:
- static void Down(..);
- static void Up(..);
...
wl::Object<wl_touch> obj_;
WaylandConnection* connection_;
WaylandTouch::Delegate* delegate_;
...
WaylandBufferManagerHost
+ void OnWindowAdded();
+ void OnWindowRemoved();
+ bool SupportsDmabuf();
+ void CreateDmabuf(..);
+ void CreateShmBuf(..);
+ void CommitBuf(..);
+ void DestroyBuf(..);
...
mojo::AssociatedRemote<ozone::mojom::WaylandBufferManagerGpu> buffer_manager_gpu_associated_;
mojo::Receiver<ozone::mojom::WaylandBufferManagerHost> receiver_;
...
base::flat_map<WaylandSurface*, std::unique_ptr<Surface>> surfaces_;
...
<<WaylandPointer::Delegate>>
Interface
virtual void OnPointerMotion(...) = 0;
virtual void OnPointerFocus(...) = 0;
virtual void OnPointerButton(...) = 0;
...
Sends events to
WaylandEventSource
+ bool StartProcessingEvents();
+ bool StopProcessingEvents();
- void OnKeyboardEvent(..) override;
- void OnPointerEvent(..) override;
- void OnTouchEvent(..) override;
std::unique_ptr<WaylandEventWatcher> event_watcher_;
WaylandWindow* window_with_focus_;
...
Use
WaylandZwpLinuxDmabuf
+ void CreateBuffer(CreateBufCallback, ...);
+ wl::BufferFormatMap supported_buffer_formats();
// zwp_linux_dmabuf_v1_listener
- static void Modifiers(..);
- static void Format(..);
- static void CreateSucceeded(..);
- static void CreateFailed(...);
...
wl::Object<zwp_linux_dmabuf_v1> zwp_linux_dmabuf_;
Use
WaylandDrm
+ void CreateBuffer(CreateBufCallback, ...);
+ wl::BufferFormatMap supported_buffer_formats();
// zwp_linux_dmabuf_v1_listener
- static void Device(..);
- static void Format(..);
- static void Authenticated(..);
- static void Capabilities(..);
...
wl::Object<wl_drm> wl_drm_;
<<WaylandKeyboard::Delegate>>
Interface
virtual void OnKeyboardFocus(...) = 0;
virtual void OnKeyboardEvent(...) = 0;
virtual void OnKeyboardModifier(...) = 0;
...
WaylandEventWatcher
+ bool StartProcessingEvents();
+ bool StopProcessingEvents();
- void OnFileCanReadWithoutBlocking(..) override;
- void OnFileCanWriteWithoutBlocking(..) override;
base::MessagePumpForUI::FdWatchController controller_;
wl_display* const display_; // Non-owned
<<WaylandTouch::Delegate>>
Interface
virtual void OnTouchFocus(...) = 0;
virtual void OnTouchPress(...) = 0;
virtual void OnTouchMotion(...) = 0;
...
igalia

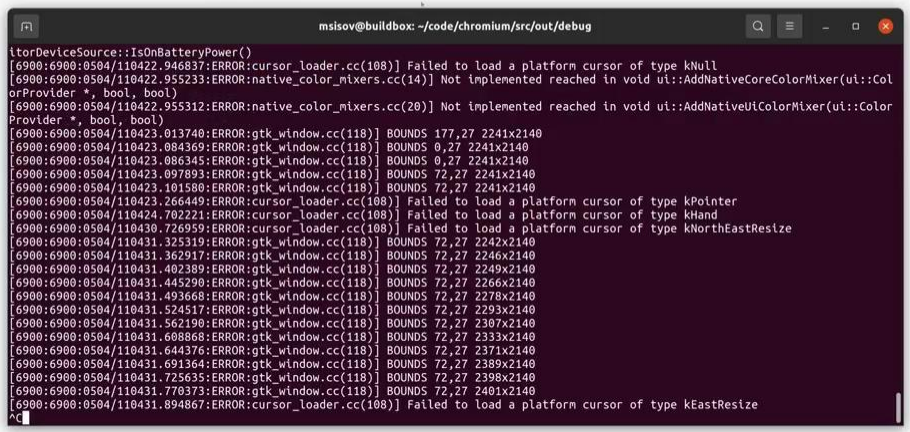

WaylandBufferManagerGPU
+ void OnSubmission(...);
+ void OnPresentation(...);
+ void CreateDmabuf(..);
+ void CreateShmBuf(..);
+ void CommitBuf(..);
+ void DestroyBuf(..);
...
mojo::Remote<ozone::mojom::WaylandBufferManagerHost> remote_host_;
mojo::AssociatedReceiver<ozone::mojom::WaylandBufferManagerGpu> associated_receiver_{this};
...
OzonePlatform
virtual InitializeForUI(..);
virtual InitializeForGPU(..);
virtual CreatePlatformWin();
...
WaylandKeyboard
+ bool Decode(DomCode, Modifiers, ...);
// wl_keyboard_listener:
- static void Key(..);
- static void Enter/Leave(..);
...
WaylandTouch
// wl_touch_listener:
- static void Down(..);
- static void Up(..);
...
wl::Object<wl_touch> obj_;
WaylandConnection* connection_;
WaylandTouch::Delegate* delegate_;
...
<<WaylandPointer::Delegate>>
Interface
virtual void OnPointerMotion(...) = 0;
virtual void OnPointerFocus(...) = 0;
virtual void OnPointerButton(...) = 0;
...
<<WaylandKeyboard::Delegate>>
Interface
virtual void OnKeyboardFocus(...) = 0;
virtual void OnKeyboardEvent(...) = 0;
virtual void OnKeyboardModifier(...) = 0;
...
<<WaylandTouch::Delegate>>
Interface
virtual void OnTouchFocus(...) = 0;
virtual void OnTouchPress(...) = 0;
virtual void OnTouchMotion(...) = 0;
...
+ bool StartProcessingEvents();
+ bool StopProcessingEvents();
- void OnKeyboardEvent(..) override;
- void OnPointerEvent(..) override;
- void OnTouchEvent(..) override;
std::unique_ptr<WaylandEventWatcher> event_watcher_;
WaylandWindow* window_with_focus_;
...
WaylandEventWatcher
+ bool StartProcessingEvents();
+ bool StopProcessingEvents();
- void OnFileCanReadWithoutBlocking(..) override;
- void OnFileCanWriteWithoutBlocking(..) override;
base::MessagePumpForUI::FdWatchController controller_;
wl_display* const display_; // Non-owned

igalia


Lives on both Browser and GPU processes
Aura
Use
Use
Creates
GtkScreen
public PlatformScreen
GetPrimaryDisplay();
GetDisplayForAcceleratedWidget();
...
OzonePlatformGtk
public OzonePlatform
GetSurfaceFactoryOzone();
CreatePlatformWindow();
CreateScreen();
InitializeUI();
InitializeGPU();
GetPlatformProperties();
Creates
GtkOzoneWindow
public PlatformWindow
Show();
SetBounds();
GetBounds();
....
Use
Creates
VIZ (GPU)
Use
GtkSurfaceFactory
public SurfaceFactoryOzone
GetAllowedGLImplementations();
GetGLOzone();
CreateCanvasForWidget();
Creates
GtkCanvasSurface
public SurfaceOzoneCanvas
GetCanvas();
ResizeCanvas();
PresentCanvas();
CreateVSyncProvider();

igalia

Demo


msisov@buildbox: ~/code/chromium/src/out/debug
itorDeviceSource::IsOnBatteryPower()
[6900:6900:0504/110422.946837:ERROR:cursor_loader.cc(108)] Failed to load a platform cursor of type kNull
[6900:6900:0504/110422.955233:ERROR:native_color_mixers.cc(14)] Not implemented reached in void ui::AddNativeCoreColorMixer(ui::ColorProvider *, bool, bool)
[6900:6900:0504/110422.955312:ERROR:native_color_mixers.cc(20)] Not implemented reached in void ui::AddNativeUiColorMixer(ui::ColorProvider *, bool, bool)
[6900:6900:0504/110423.013740:ERROR:gtk_window.cc(118)] BOUNDS 177,27 2241x2140
[6900:6900:0504/110423.084369:ERROR:gtk_window.cc(118)] BOUNDS 0,27 2241x2140
[6900:6900:0504/110423.086345:ERROR:gtk_window.cc(118)] BOUNDS 0,27 2241x2140
[6900:6900:0504/110423.097893:ERROR:gtk_window.cc(118)] BOUNDS 72,27 2241x2140
[6900:6900:0504/110423.101580:ERROR:gtk_window.cc(118)] BOUNDS 72,27 2241x2140
[6900:6900:0504/110423.266449:ERROR:cursor_loader.cc(108)] Failed to load a platform cursor of type kPointer
[6900:6900:0504/110424.702221:ERROR:cursor_loader.cc(108)] Failed to load a platform cursor of type kHand
[6900:6900:0504/110430.726959:ERROR:cursor_loader.cc(108)] Failed to load a platform cursor of type kNorthEastResize
[6900:6900:0504/110431.325319:ERROR:gtk_window.cc(118)] BOUNDS 72,27 2242x2140
[6900:6900:0504/110431.362917:ERROR:gtk_window.cc(118)] BOUNDS 72,27 2246x2140
[6900:6900:0504/110431.402389:ERROR:gtk_window.cc(118)] BOUNDS 72,27 2249x2140
[6900:6900:0504/110431.445290:ERROR:gtk_window.cc(118)] BOUNDS 72,27 2266x2140
[6900:6900:0504/110431.493668:ERROR:gtk_window.cc(118)] BOUNDS 72,27 2278x2140
[6900:6900:0504/110431.524517:ERROR:gtk_window.cc(118)] BOUNDS 72,27 2293x2140
[6900:6900:0504/110431.562190:ERROR:gtk_window.cc(118)] BOUNDS 72,27 2307x2140
[6900:6900:0504/110431.608868:ERROR:gtk_window.cc(118)] BOUNDS 72,27 2333x2140
[6900:6900:0504/110431.644376:ERROR:gtk_window.cc(118)] BOUNDS 72,27 2371x2140
[6900:6900:0504/110431.691364:ERROR:gtk_window.cc(118)] BOUNDS 72,27 2389x2140
[6900:6900:0504/110431.725635:ERROR:gtk_window.cc(118)] BOUNDS 72,27 2398x2140
[6900:6900:0504/110431.770373:ERROR:gtk_window.cc(118)] BOUNDS 72,27 2401x2140
[6900:6900:0504/110431.894867:ERROR:cursor_loader.cc(108)] Failed to load a platform cursor of type kEastResize
^C

Activities Demo_gtk --no-sandbox --disable-gpu --in-process-gpu --enable-crashpad youtube.com touko 4 11:06 en

Jäätävä moottoripyöräku youtube.com/watch?v=I5gOX5CaMTo

You are using an unsupported command-line flag: --no-sandbox. Stability and security will suffer.

Chromium isn't your default browser Set as default

YouTube FI Search SIGN IN

Jäätävä moottoripyöräkuume! - Vihdoin A-kortti!

23,974 views • May 3, 2021 1.6K 11 SHARE SAVE

Vinkare
159K subscribers
SUBSCRIBE

Vihdoin pääsin toteuttamaan haaveeni ja hankin A-kortin! :D Kerron miten suoriuduin moottoripyöräkurssista ja pähkäilen minkä prätkän hankkisin itselleni.

SHOW MORE

Mun piti vaan käydä kattomassa...
Ninaonilo
165 views • 36 minutes ago
New

KOEAJOSSA - TOYOTA CELSIOR (LEXUS LS) 400 F-PACKAGE -9...
meihemi
12K views • 18 hours ago
New

Heikelä vs. Maailma: Jaajo Linnonmaa
UnibetFinland
1.1K views • 17 hours ago
New

Mix - Vinkare
YouTube

MEIDÄN VIIMEINEN TAISTO
MattePlays
6.2K views • 1 day ago
New

Air Fryer ja parhaat kahvireseptit! - Keittiöspecial!
Vinkare
86K views • 10 months ago

PÖSÖN TERMOSTAATIN VAIHTO
Herra Hermeettinen
600 views • 3 hours ago
New

VLOG 032 - Bilsäsongen knackar på dörren
Jarudds Classics
15K views • 3 days ago
New

Nuljuumista
Itärajan idiootti
11K views • 1 day ago
New

Jättitelkka ja PS5 olohuone setuppiin!
Vinkare

Thank you