Integration of the Chromium Browser in the GENIVI Platform

Jacobo Aragunde Pérez

blogs.igalia.com/jaragunde

GENIVI®

igalia

igalia

- Open Source experts and consultants
- 15 years of experience
- Important contributions to:
  - Client-side web tecnologies: WebKit, Blink/Chromium, Servo
  - Graphics & Multimedia: Mesa, GStreamer
  - Compilers: V8, JavaScriptCore, SpiderMonkey, Guile
  - Software-defined networking: Snabb
  - ...

Introduction

Goals of the project

- Integrate full-featured Chromium browser in GDP
  - Use Intel’s Ozone-Wayland project, most complete implementation of Wayland so far
  - Get latest possible version of the browser working
- Analyze and fix multi-seat implementation, if required

Elements and versions

- Chromium: latest stable release was 54 at that point
- Ozone-Wayland: latest branch supports Chromium 53
- Meta-browser: supporting Chromium 48
- GENIVI BSPs

Rebase & integrate Chromium browser

Work on meta-browser

- Simplify configuration
  - Obsolete CHROMIUM_ENABLE_WAYLAND detection
- Build chromium+wayland version 53
  - Recipe was pointing to version 48
- Fix specific build scenarios
  - x86 32 bit, mixed x11+wayland libs
- Contributed to upstream meta-browser project

Integration in GDP

- Integration with HMI: .app and .service files
  - .desktop for GDP12/master
- Force window size to get the proper aspect ratio
- Fix compilation on different platforms: Raspberry Pi 3, R-Car Gen 3
- Backport upstream chromium patch to fix issues with kernel >=4.5 (GDP master)
- Under revision on PR#53 (GDP master)
  - GDP 11 patches available in branch

Current status

- GDP integration under revision on PR#53 (GDP master)
- Supported hardware:
  - Minnowboard is up and running
  - Raspberry Pi 3: work in progress
  - R-Car Gen. 3: work in progress
- HMI integration in GDP 12/master
  - Detected some issues in HMI side

Multi-seat implementation

Analysis

- Check status of multi-seat features in Ozone-Wayland
- Issues found:
  - Keyboard focus can be stolen
  - Only one browser window receives all input events on a multi-seat configuration

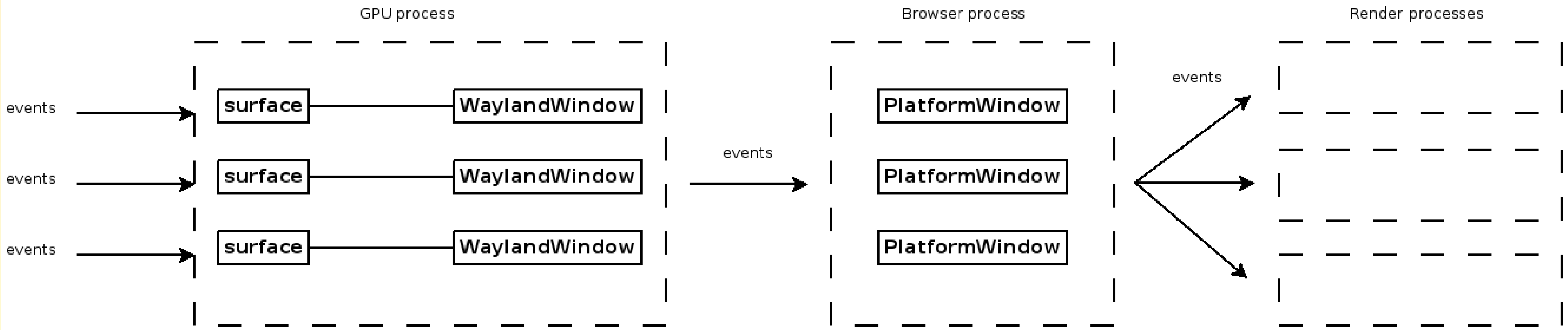
Keyboard focus can be stolen

- Underlying cause: no independent keyboard and pointer focus
  - Window with pointer focus also receives keyboard events
- Solution:
  - Split keyboard and pointer focus
  - Modify window focus logic
    - Enable window focus on keyboard events
    - Break assumption of only one focused window

Multi-seat configuration problems

- Underlying cause: assumption there is only one focused window
  - Focused window receives all events
  - Chromium browser process gathers all events and sends them to the focused window
  - No awareness of the different seats sending events

Flow of events in Chromium processes



Parts of the solution

- Input device abstractions must be aware of the seat they belong to
  - Add extra information to internal IPC messages related to events
- Browser process must be aware of the concept of “seat”
  - New IPC signal to sync seat objects between GPU and browser processes
- GPU process must check seat assignment
  - Sync with browser process using a new IPC signal

Current status

- WIP implementation available in branch wip/multi-seat
- Implementation basically complete
- Requires some additional work & clean-up
- Goal: contribute back to main Ozone-Wayland project
  - Probably required to remove IVI input bits

Next steps

Next steps

- CEF on Wayland
    - Will provide a stable API to build upon
    - Will provide API for embedders
- Browser API implementation
    - Start with a small subset of operations

Next steps

- Integration with GENIVI subsystems:
  - Audio Manager
  - Persistence API
- Longer term: replace Ozone-Wayland
  - Wait until upstream catches up