

# The Chromium/Wayland project

FOSDEM 2018  
(Feb/2018)

Maksim Sisov, msisov@  
Antonio Gomes, tonikitoo@



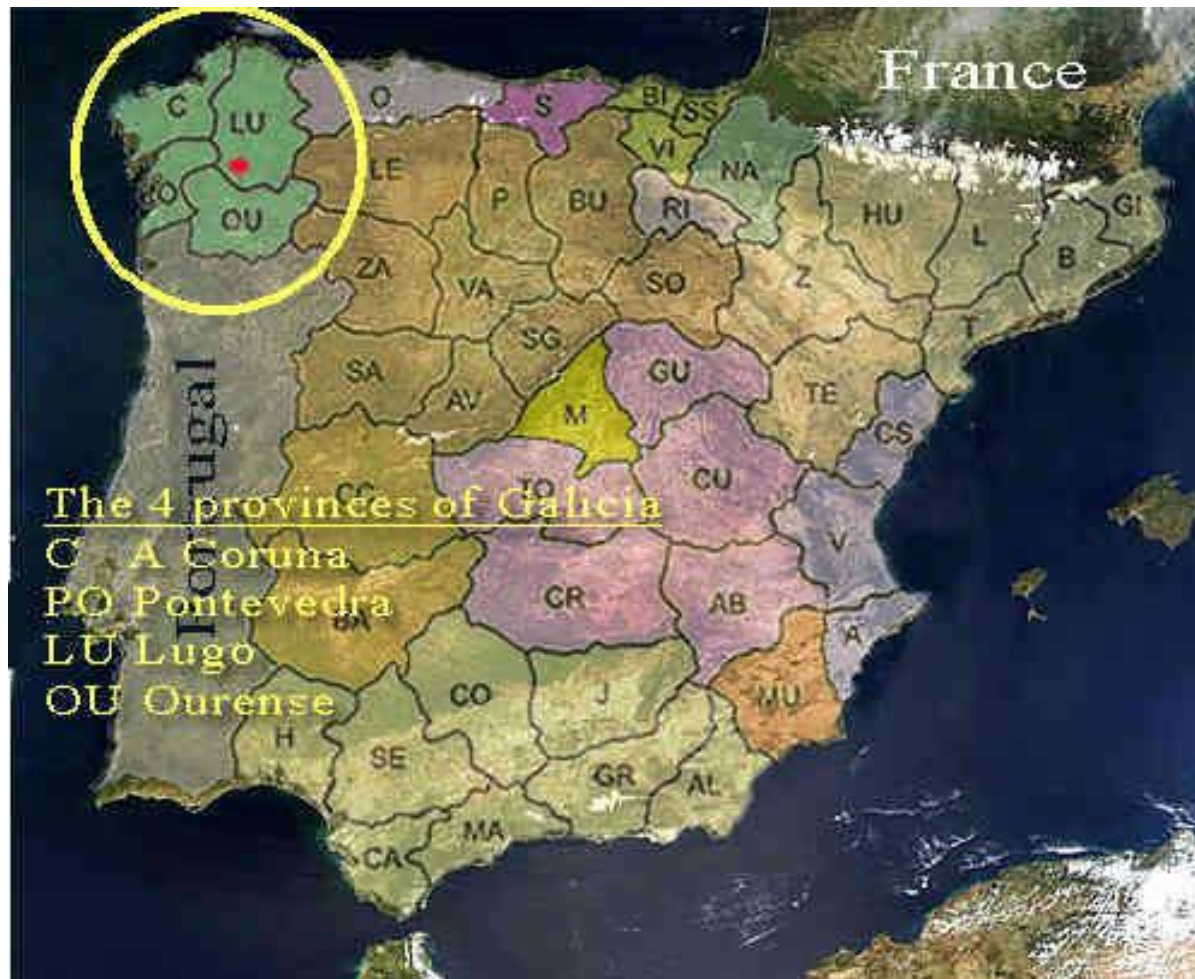
# Agenda

- About Igalia
- Goals & Motivation
- Background
- Developments
- Demonstration



# About Igalia

- Worker-owned, employee-run Open Source consultancy company, based in Galicia, Spain.

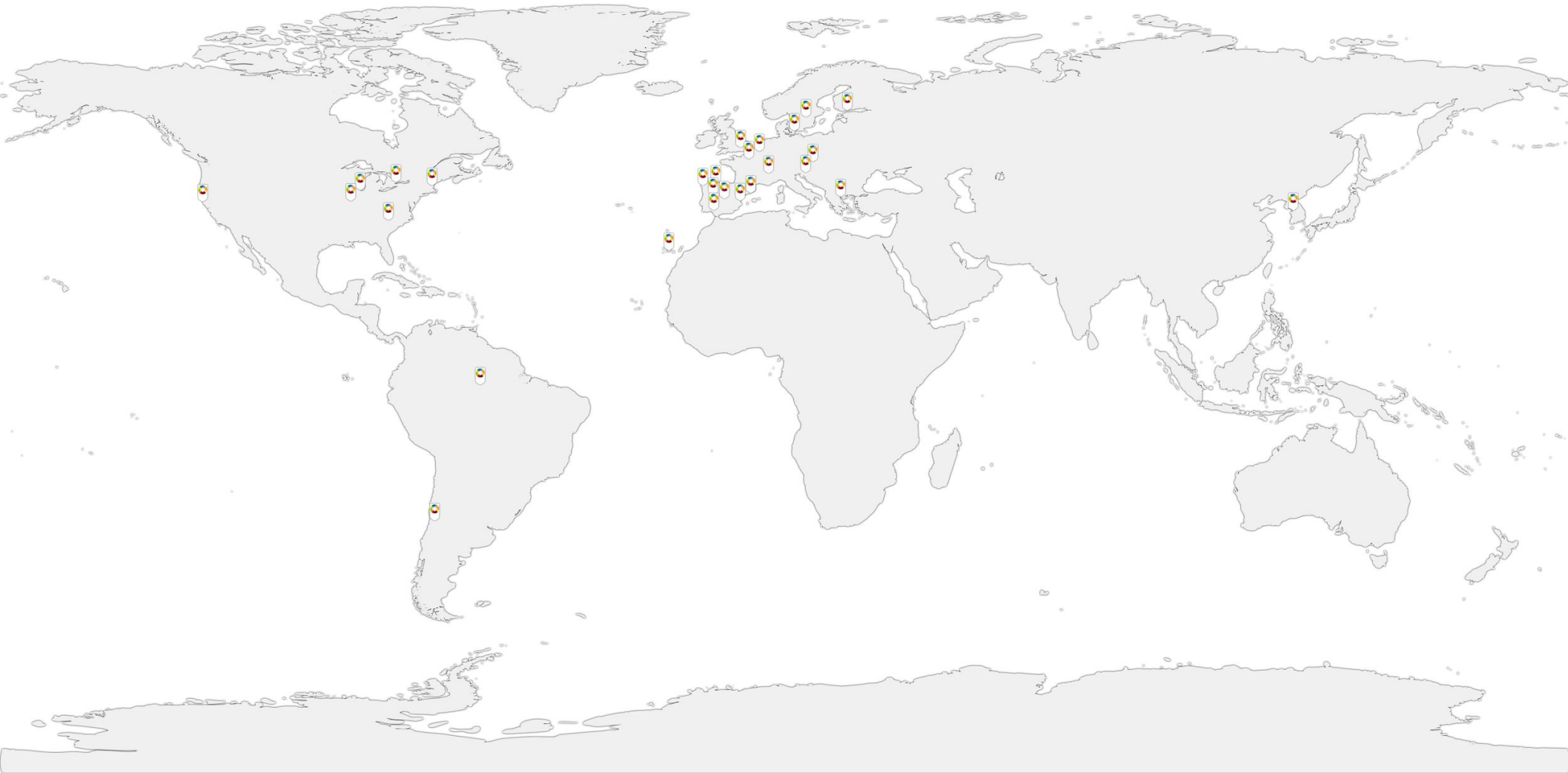


# About Igalia

- ~62 employees around the world.
- Areas
  - **Chromium/Blink**, WebKit and Servo; W3C member;
  - Compilers, JavaScript engines (V8, JSC);
  - Multimedia, Kernel, Networking;
  - Accessibility, Virtualization & Cloud.



# About Igalia



**igalia**

# Goals & Motivation



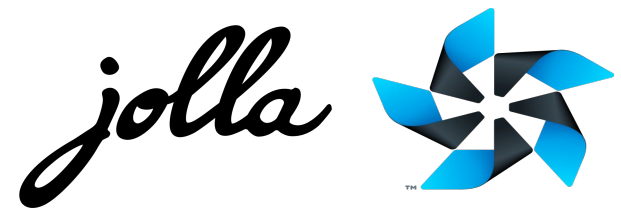
# Goal

- Be able to run Chromium natively on Wayland-based systems.



# Motivation

- Wayland is a mature solution.
- Demand from different industries.
  - **Automotive**
  - **Mobile**
  - **Desktop**





# Background



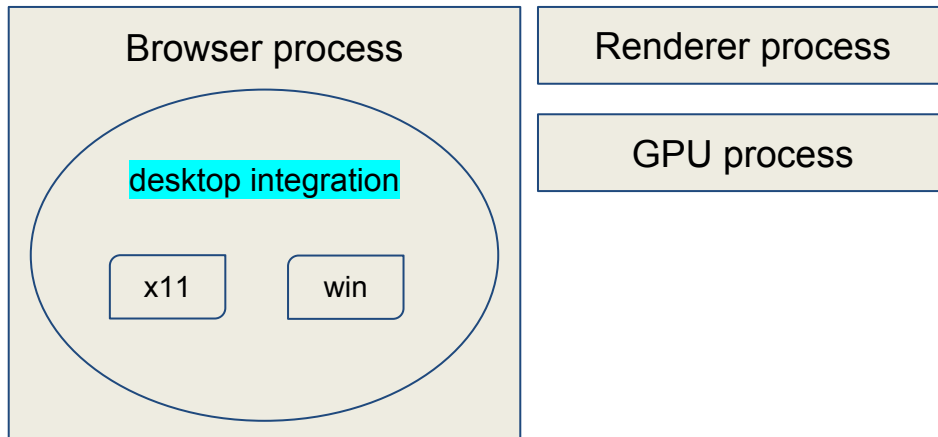
# Background - Ozone/Wayland

- By Intel / 01.org.
- [Ozone](#) project (original).
  - Abstraction layer for the construction of accelerated surfaces **underlying the Aura toolkit**, as well as input devices assignment and event handling.
  - Backends:
    - DRI -> DRM
      - GBM
      - ChromeOS
    - **Wayland (off trunk)**
      - **Linux**

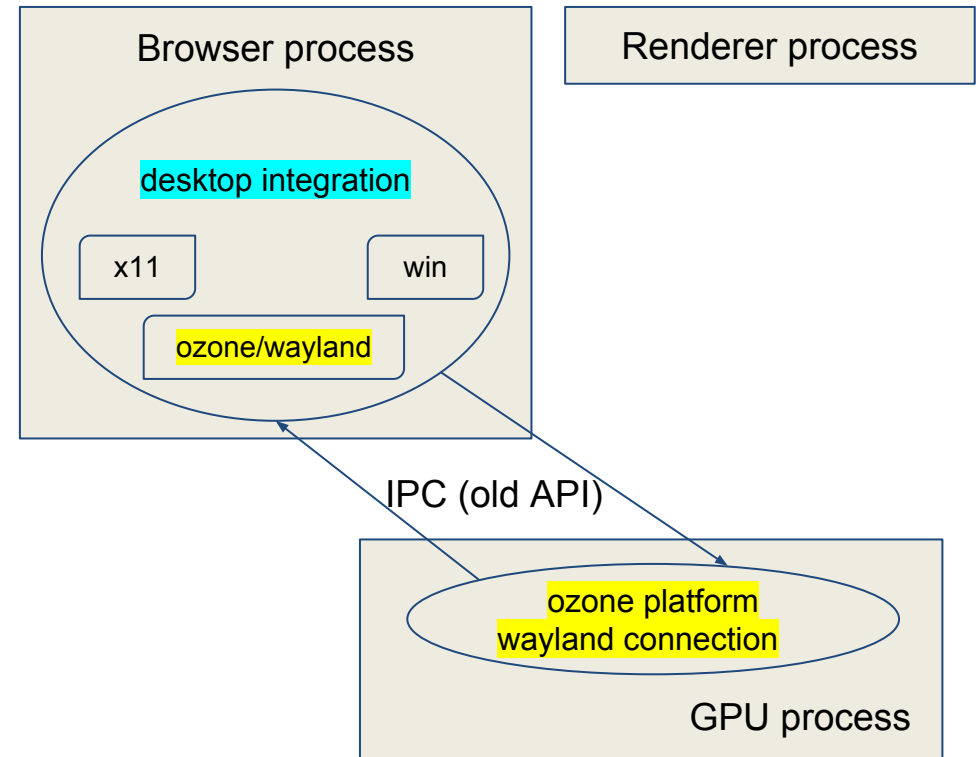


# Background - Ozone/Wayland

Desktop integration



Desktop integration (01.org)




# Background - Ozone/Wayland

- Good community adoption.
- **Project entered in “maintenance mode”.**
  - December/2015.
  - Chromium m49.
    - Today's ToT is **m66**.



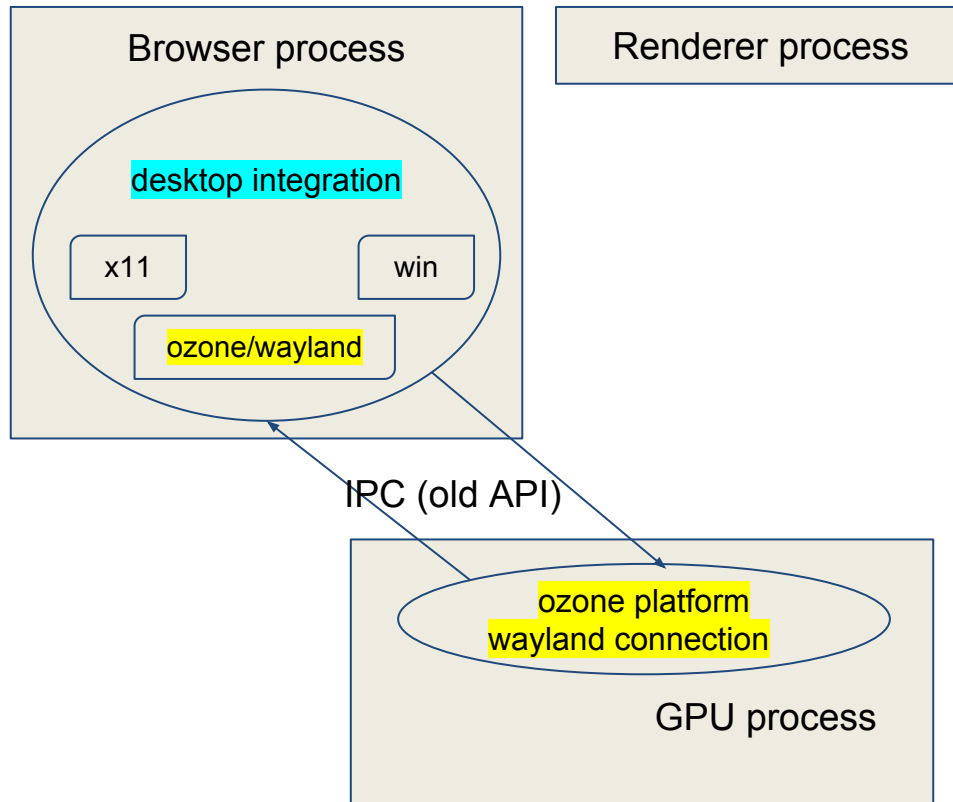
# Background - Cr Upstream (1/)

- In the meanwhile, Ozone layer in ToT received two new backends:
  - x11
  - **wayland**
- Is the problem solved? 
- The original “desktop integration” approach taken in **Ozone/Wayland** did not comply with the way future Linux desktop Chrome is foreseen.

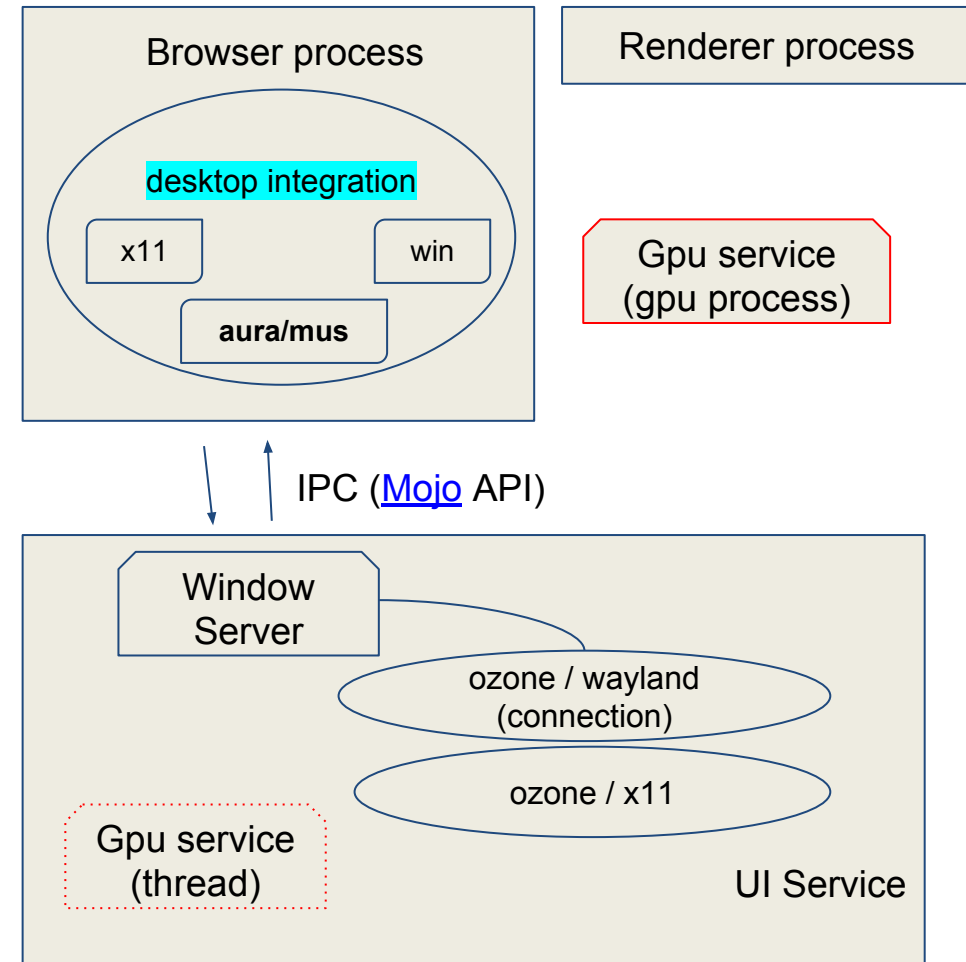


# Background - Desktop integration

## Linux desktop integration (O1.org)



## Mus Linux desktop integration



# Background - Cr Upstream (2/)

- [Ozone](#) project
  - Abstraction layer for the construction of accelerated surfaces **underlying the UI Service** (aka *Mus*), as well as input devices assignment and event handling.
  - Backends:
    - ChromeOS
      - DRM / GBM
      - x11
      - **Wayland**
    - Linux



# New developments

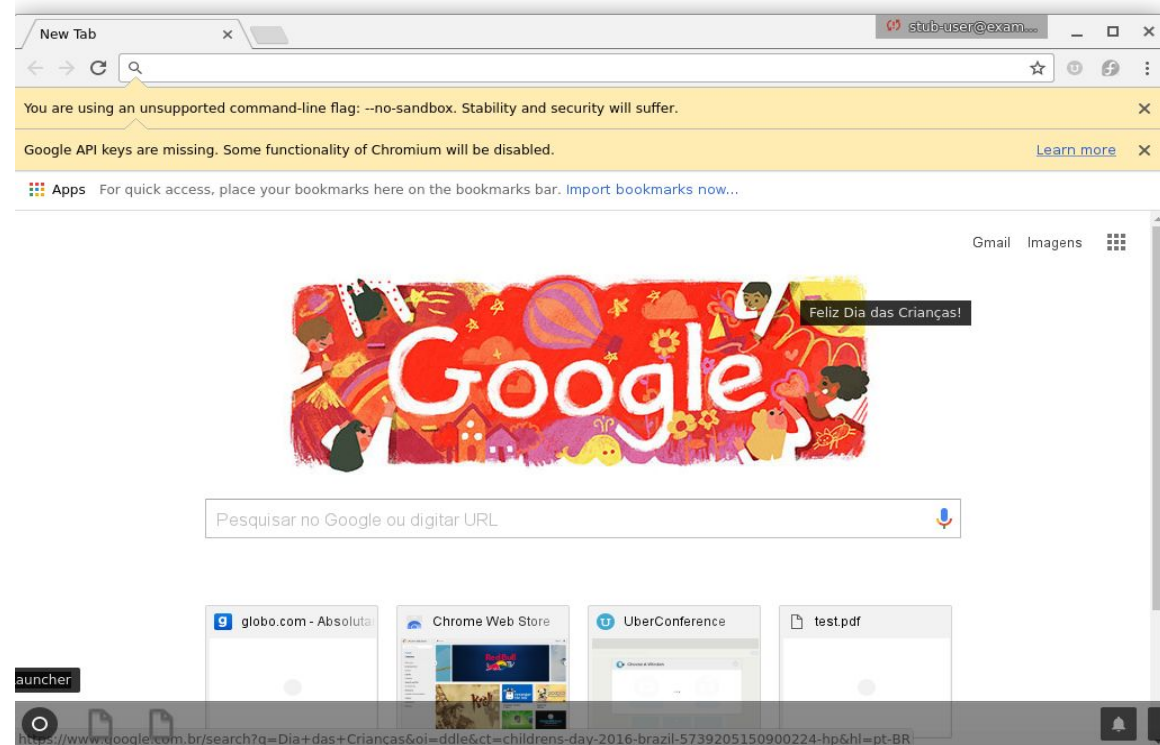
## Phase 1 - The bring up





# Phase 1 - The bring up

- Sept-Oct/16
  - Igalia brought up of Ozone's Wayland backend in ToT.
  - Experimented with “Ozone != ChromeOS”.
  - [Documentation](#)
  - [Buildbots](#)



# Phase 1 - CrOS

- **Internal-window mode**

- CrOS has a Window Manager (WM) and a ScreenManager (SM).
- Chrome and other app windows in the system
  - end up sharing a single display.
  - are embedded within a single top-level *acceleratedWidget*.



# Phase 1 - Desktop Chrome

- **External-window mode**

- Desktop Chrome has no WM.
  - One *acceleratedWidget* per Chrome window.
  - User manipulates *acceleratedWidgets* via the host OS window.
    - maximize, minimize, resizing, dragging, fullscreen.
- Desktop Chrome has no SM.



# Phase 1 - Demo



- Nov-Dec/16
  - CES demo: Linux/AGL/Wayland on R-Car M3.
  - [meta-browser](#)



# New developments

Phase 2 - Chrome / Mus



# Mus' External Window Mode (1/)

- Modify *IWM* so that it creates native *acceleratedWidget*'s for each top-level window.
  - **Extend Mus and Ozone** to support 'External Window' mode.
- No major functionality loss if compared to stock Chrome.



# Mus' External Window Mode (2/)

- Extend the *mus\_demo* to work in 'external window' mode.
- Rework internal window mode assumptions in the code
  - [1:1 relation of ws::Display and display::Display.](#)
- Extend Mus and Ozone to support 'external window mode'.
- Make the code that handles the existing **-mus** command line parameter non-ChromeOS specific.
  - Chrome today launches the same way it ought to, for Chrome/Mus.



# Mus' External Window Mode (3/)

- Added support to:
  - **XDG v6.**
  - **Keyboard events, auto repeat.**
  - Mouse cursors.
  - Touch events (thanks to Collabora!).
  - Multiple windows.
  - Built-in window decoration.
  - Window closing.
  - Menus, widgets, and tooltips.
  - Support to common windowing features:
    - maximize, minimize, restore, fullscreen, dragging and resizing.



igalia



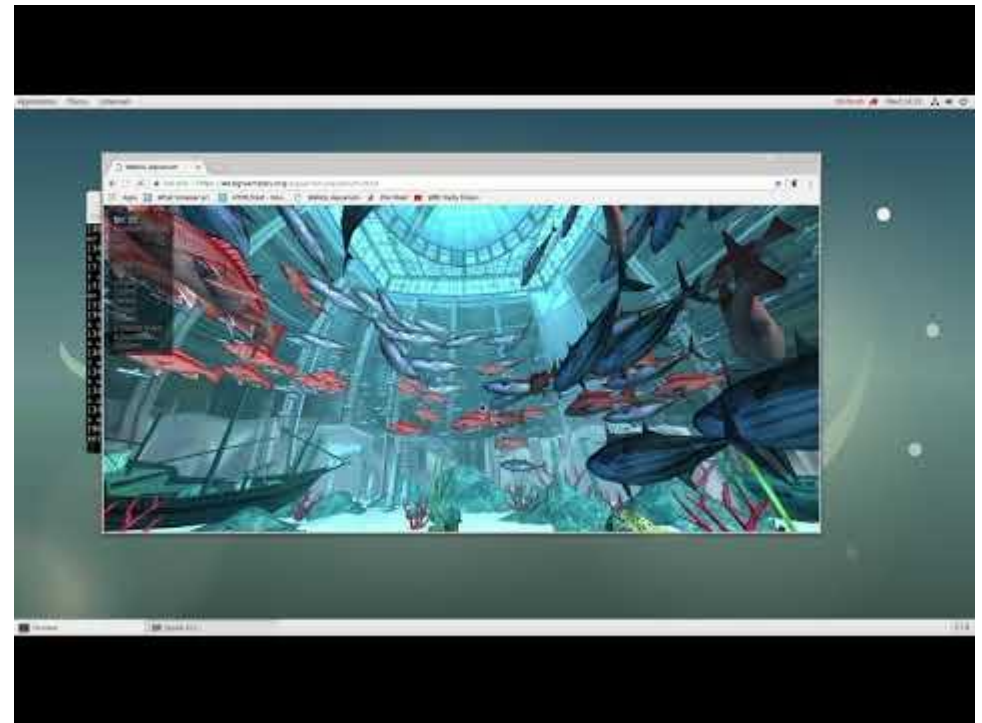
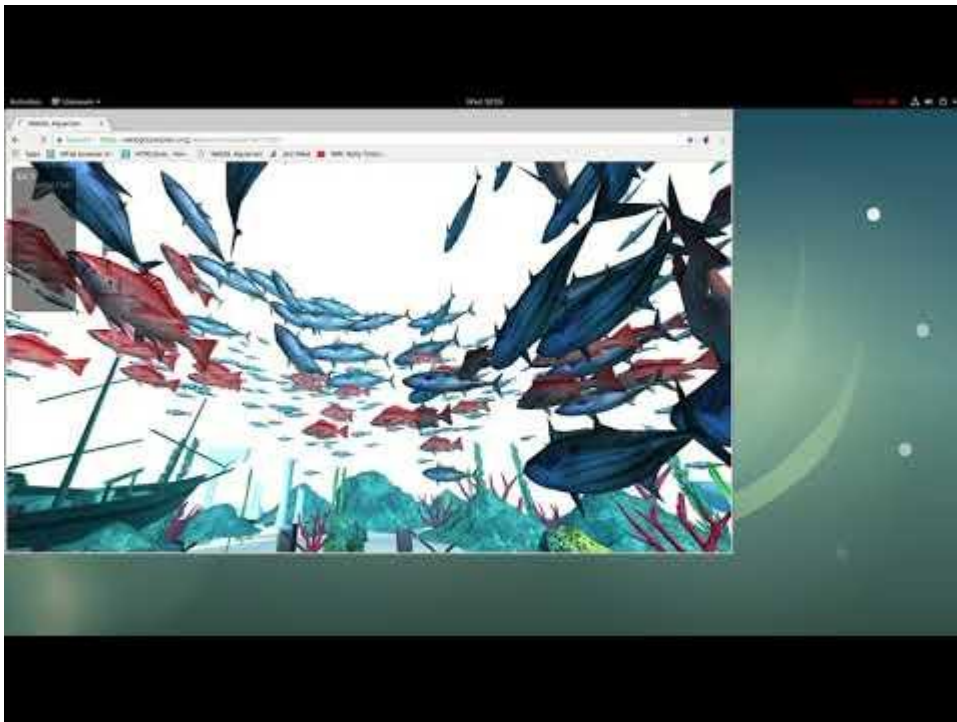
# Mus' External Window Mode (4/)

- Changed ownership model of some objects.
- Implemented keyboard/IME service integration.
- Implemented a slightly custom “window tree hierarchy”.
- Reworked our “access policy”.
- Followed mushrome’s process model.
- Worked extensively on stability and hardness of our impl.
- Made the content\_shell running with --mus on Linux.



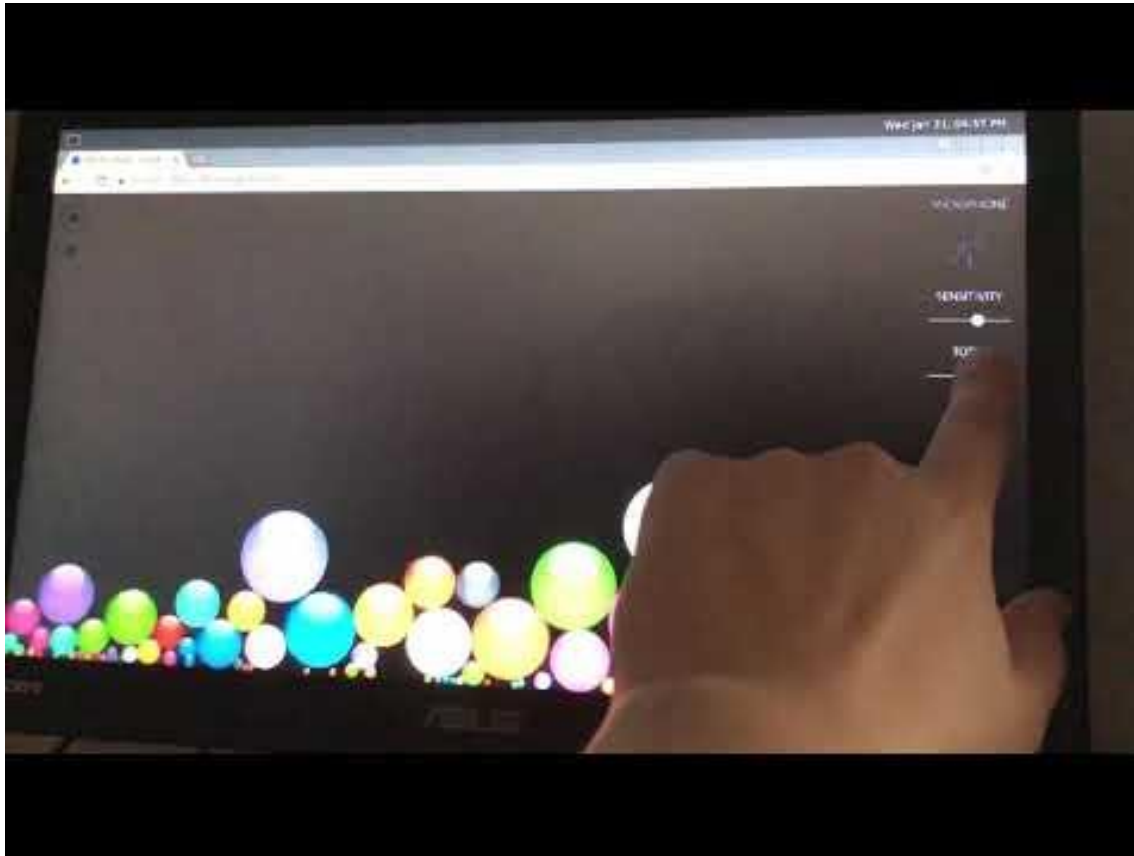
# Mus' External Window Mode (5/)

- What is the status today?  
**Ready for alpha testing.**



# Mus' External Window Mode (6/)

- Performance improvements



# About the project (1/)

- The project is being hosted on [GitHub](#).
- Well defined contribution policy:
  - Peer review.
  - Buildbot running existing tests:
    - *services\_unittests* and *ozone\_unittests*.
    - *browser\_tests*, *content\_browsertests* (~98% pass rate).
    - *content\_unittests*.



# About the project (2/)

- Rebase strategy:
  - Weekly based.
  - Continuous history clean up.
    - `git commit --fixup <SHA>`
    - Eliminate commit + revert “commit” pairs.
    - Use of [DoNotCarryForward] tag.
- Periodic sync up with Google.



# About the project (3/)

- Documentation available at  
[https://docs.google.com/document/d/1yzUWttsyqTh31vAyn4Xj4xblr3GOYIF44IBIFP\\_ixT0](https://docs.google.com/document/d/1yzUWttsyqTh31vAyn4Xj4xblr3GOYIF44IBIFP_ixT0)



# TODO

- Fix drag and drop.
- Fix clipboard between Chromium and host.
  - it works as in internal window mode now.
- Add multi screen support.
- Support non-english keyboard layouts.
- Ensure no feature losses or major performance penalties when compared to stock Chromium X11/Linux.



# TODO

- Integration with AGL. (Started and done by IoT.bzh)
- Release desktop installers (.deb .rpm).
- Continue upstreaming the project to ToT.
- Enable more tests in ToT buildbot.
- Decouple MUS from VIZ (visuals service).





# Questions?

[msisov@igalia.com](mailto:msisov@igalia.com) - Maksim Sisov

[tonikito@igalia.com](mailto:tonikito@igalia.com) - Antonio Gomes

