

Testing a browser on a real Wayland compositor

Orko Garai

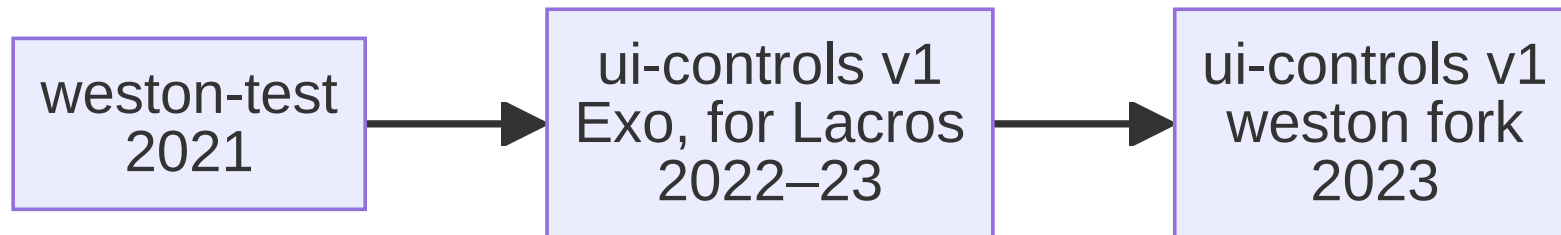
The problem

Chromium has thousands of **interactive UI tests** that need:

- **Synthesized input:** pointer, keys, touch
- **Surface-relative pointer:** move within a given window
- **Synchronization:** know when a request has taken effect
- A **headless** display with a fixed size

Wayland clients **can't inject input**, by design (security). No portable test API.

Before: weston



Chrome's `ui_controls` test API as requests: keys, mouse, touch, each acked by `request_processed`

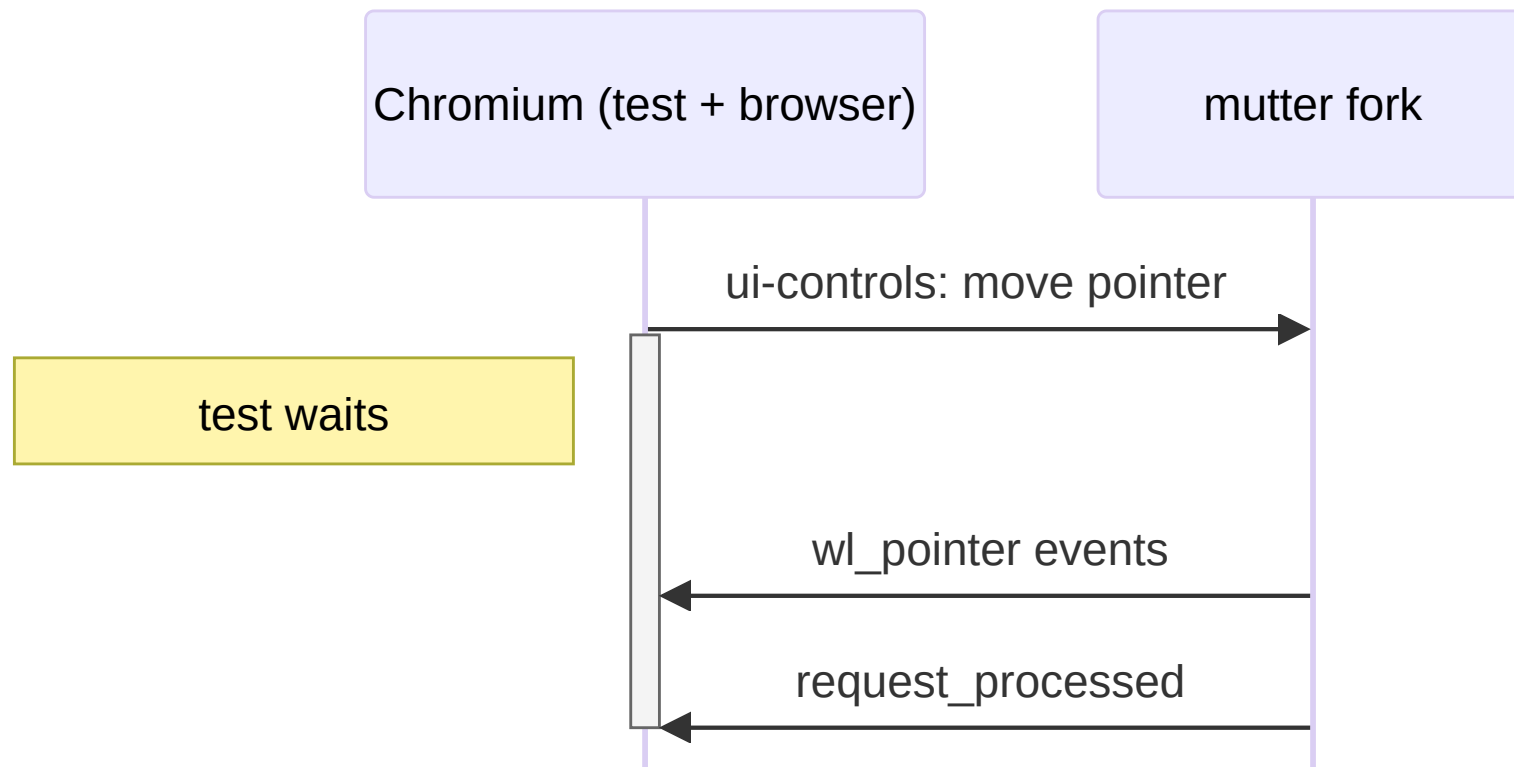
- Weston lacks **toplevel-drag**
- Weston-only bugs → flaky tests

What exists? Not enough.

Option	Missing for us
wlroots virtual-pointer / virtual-keyboard	event sync, surface-relative pointer
KDE fake-input	event sync, KDE-only
GNOME RemoteDesktop D-Bus	needs gnome-shell + unsafe mode
libei	event sync, surface-relative pointer

Decision: add our protocol to **mutter**, which real users run.

What we built



- **ui-controls-v2**: `wl_surface`, so bubble subsurfaces work

Into Chromium CI (2025)

When	Milestone
Mar	mutter in tree, <code>xvfb.py --use-mutter</code> , CI tester
Apr-May	<code>content_browsertests</code> + <code>browser_tests</code> on mutter
May	<code>linux-wayland-mutter-rel-tests</code> joins gardening rotation
Jun	mutter try bot replaces weston on CQ

It works! 🎉

- **Weston-disabled tests run:** menus, drag and drop, tab drag, focus
- **More stable** than weston, key with thousands of tests
- Found **real bugs** 🐛 on both sides
 - Chromium: bookmark menu popup, xdg-foreign
 - mutter: no-Xwayland crash, **fixed upstream (!4351)**

The catch 🥹

We carry **forks of mutter and weston**, just to run tests.

- Rebased onto every upstream release
- Tests only cover the compositors we forked
- Chromium, KDE, wlroots: each has its own input protocol

The ask: test apps on any compositor

One compositor-agnostic interface for:

1. **Input emulation:** pointer relative to a surface, keys, touch
2. **Synchronization:** "this request has fully taken effect"
3. **Headless outputs** with a set size and scale
4. **Room to grow** for what other apps and toolkits need

Win-win: real desktops for apps, real apps for compositors

👉 Join **wayland-protocols #230**, or find me in the hallway!

Thank you!

- [Design doc](#)
- Discussion: [wayland-protocols #230](#)
- Code in chromium/src:
 - `testing/xvfb.py`
 - `third_party/mutter`
 - `ui-controls-unstable-v2.xml`

Join us!



Questions?

orko@igalia.com

<https://www.igalia.com/jobs>

