

Waltham plugins upstreaming

Weston multiple backends support

last update: <Date of last Document Revision dd-mmm-yyyy>

Document classification: <public, internal, confidential, strictly confidential> Default classification is „internal“

Waltham plugins upstreaming ideas

- **Keep the Waltham plugins in AGL**
- **Upstream to own repository hosted by ... ADIT?**
- **Implement stand alone Waltham backend in Weston like any other backend today**
- **Rework the Weston to support multiple backends at the same time**

Keep the Waltham plugins in AGL

■ Pro

- ◆ No additional efforts 😊

■ Cons

- ◆ Visibility of the Waltham is only in AGL. No other users/ contributors

Upstream to own repository

■ Pro

- ◆ Potentially a bit better visibility
 - No need to use AGL to see the Waltham in action

■ Cons

- ◆ Additional repository and community to handle
 - Different community
 - Different release cycle with Weston
 - Compatibility with Weston has to be maintained

Implement stand alone Waltham backend in Weston

- **backends are working exclusively only, one at a time**
 - ◆ Presenting same content to e.g. drm-backend and Waltham backend is not possible
- **Currently not fitting to high level requirements of Waltham transmitter but a good starting point!**
- **The multiple backend implementation can be done in parallel (complex)**

Multiple backends support

- **Cleanup and rework the dependency between [Efforts ??]:**
 - ◆ Compositor, Renderer, Backend
 - ◆ Currently a lot of cross dependencies in compositor.c
- **Required sync points between backends [Efforts ??]**
 - ◆ handling of frame events
 - ◆ ...
- **Rework on existing backends [Efforts ??]**
 - ◆ Wayland
 - ◆ Pixman
 - ◆ Drm-backend
- **Add multi threading to weston? [Efforts ??]**
 - ◆ Avoid latency propagation between backends!

■ Additional points: