



# Weston/ wayland-ivi-extension

22 Oct 2015 | Genivi community

Eugen Friedrich  
Software Engineer  
Advanced Driver Information Technology

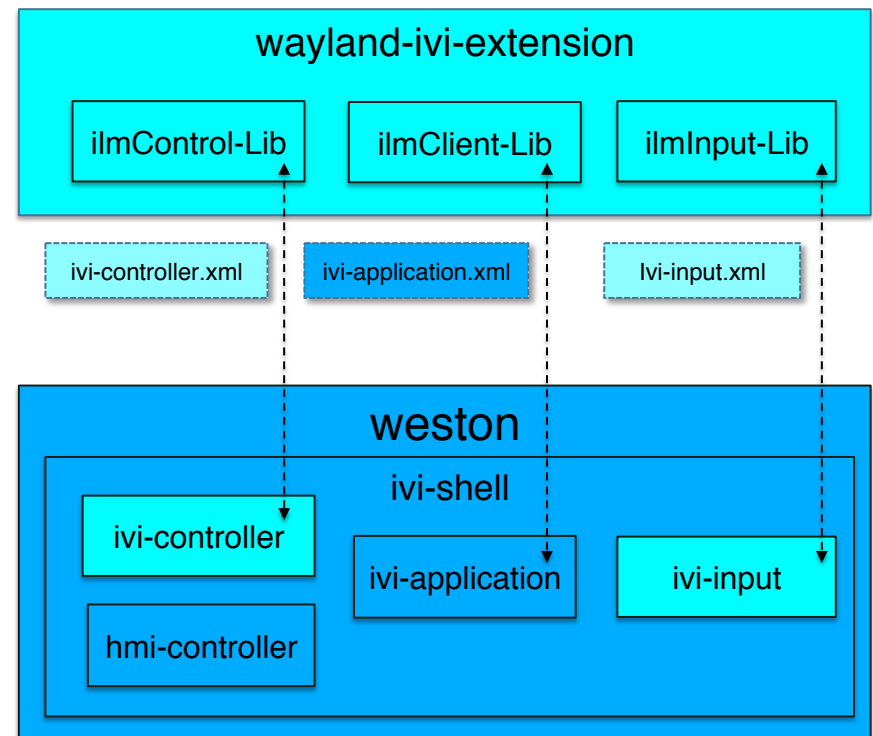
5-Oct-15

GENIVI is a registered trademark of the GENIVI Alliance in the USA and other countries  
This work is licensed under a Creative Commons Attribution-Share Alike 4.0 (CC BY-SA 4.0)

- Details of implementation
  - Git repositories, Genivi focus
- Motivation of changing the ILM interface
  - Strict separation between „normal“ client and hmi controller
  - Remove redundancy
  - Missing functionality
    - `Ilm_registerNotification`
  - Set of input handling api's
- Reference implementation versions

# ILM api implementation

- Wayland extension is Genivi repository
  - ILM api's are implemented here
  - main focus of genivi activities
- Weston is an independent open source project
  - ivi-shell impementation is here
- Dependency is very high
  - Improvements and bug fixes may affect both repositories





# ILM api implementation

- ivi-application.xml, ivi-controller.xml and ivi-input.xml
  - Wayland protocol definitions
  - protocols are designed to support the know api from LayerManager
  - ILM api in wayland-ivi-extension implements the protocols on the client site
  - Server site is implemented in ivi-shell
    - (ivi-controller,ivi-application,ivi-input)



## Motivation of changing the ILM interface

- Separation of “normal“ client and controller application
  - „normal“ clients only provide content to the compositor and should only include ilm\_client.h and link to libilmClient.so
  - Controller applications control the scene and use controller api and should include ilm\_control.h and link to libilmControl.so
  - From the begging this idea was in the api design but it was not implemented this way -> WE FIXED IT



## Motivation of changing the ILM interface

- Redundant functionality
  - Some of api had very similar functionality
  - Those are removed without loosing any functionality
  - Removed apis:
    - ilm\_surfaceSetPosition, ilm\_surfaceGetPosition
    - ilm\_layerSetPosition, ilm\_layerGetPosition
    - ilm\_layerSetDimension, ilm\_layerGetDimension
    - ilm\_surfaceSetDimension



# Motivation of changing the ILM interface

- Missing functionality
  - Notification of creation and destroying of layers and surfaces
  - New api is added `Ilm_registerNotification`

- Allows explicit input routing from concrete input device to ilm surfaces
- Input devices have to be grouped into seats
  - Default seat is always available
  - Final grouping and assigning to the seats has to be done in a concrete system according to the system requirements
- One input event can be distributed to several surfaces
  - Use case: multiple keyboard focus



- Setup the input routing by using input api
  - ilm\_setInputAcceptanceOn
  - to be done by HMI/Application controller
- Control the input focus :
  - ilm\_setInputFocus
  - to be done by HMI/Application controller

- ilm\_getInputAcceptanceOn
  - get the list of seats which the surface accepts
- ilm\_getInputFocus
  - get the list of focused surfaces with corresponding device types
- ilm\_getInputDevices
  - get the list of seats with given input device types
- ilm\_getInputDeviceCapabilities
  - get the list of input device types which the given seat has



and the big thanks goes to

- Emre Ucan
- Tanibata Nobuhiko
- Others LayerManagement community members



## Reference implementation details

- Weston repository tagged with 1.8  
<http://cgit.freedesktop.org/wayland/weston>
- Wayland-ivi-extension repository tagged with 1.5  
<http://git.projects.genivi.org/?p=wayland-ivi-extension.git>



Urgent questions?  
let get to practice ...



## Additional information

- [http://wiki.projects.genivi.org/index.php/Wayland\\_IVI\\_Extension\\_Design](http://wiki.projects.genivi.org/index.php/Wayland_IVI_Extension_Design)