



IVI LayerManagement

Oct 10th, 2013

Timo Lotterbach
LayerManagement ET
BMW

Nobuhiko Tanibata
LayerManagement ET
ADIT

Takanari Hayama
Guest @ GENIVI AMM
Igel Co., Ltd.

1. IVI LayerManagement
 - Introduction & Status (Lotterbach)

2. DirectFB & IVI LayerManagement
 - Status & Demo (Hayama)

3. Wayland IVI Extension
 - Introduction, Status & Demo (Tanibata)

Main Use Case

- End-users expect integration of a variety of applications into the HMI of their car
 - Navigation Software
 - TV & Video
 - Internet Browser
 - Vehicle Information
 - Assistant Information
 - Phone and Messaging
 - CE Device Integration
 - ...



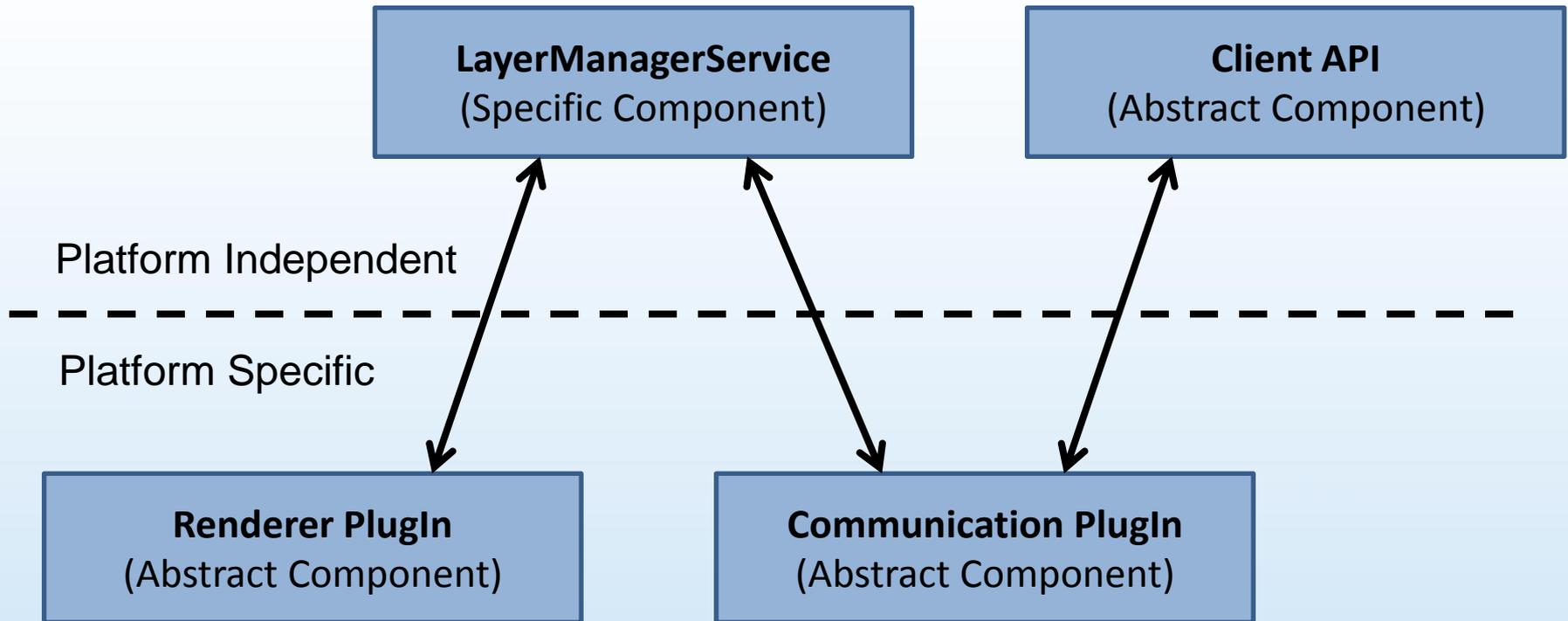
Pictures: <https://www.press.bmwgroup.com>



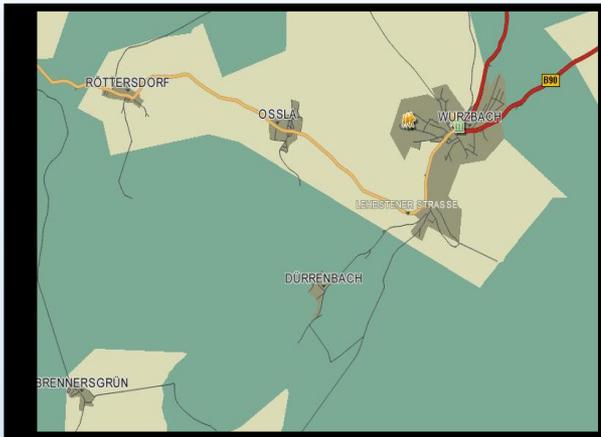
Compositing Requirements

- Standardized Compositing API & behavior
- Abstraction of underlying hardware
- Consistent usage of hardware acceleration
- Separation of HMI Software and Compositing
- Separation of Clients & Controllers
- Extendable during runtime

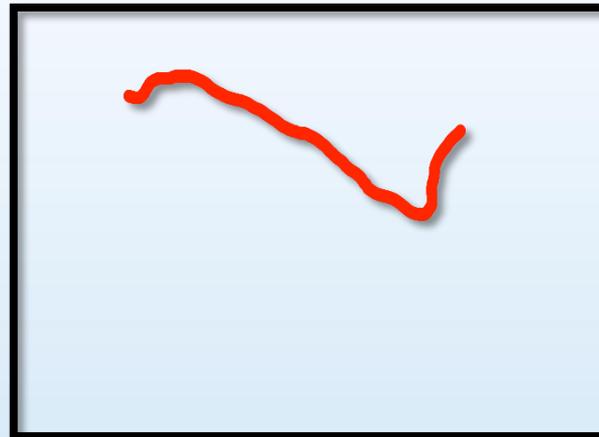
Architecture



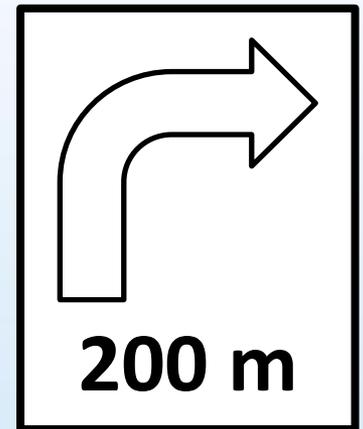
- Navigation Surfaces
 - Rendered by different Processes



MapViewer

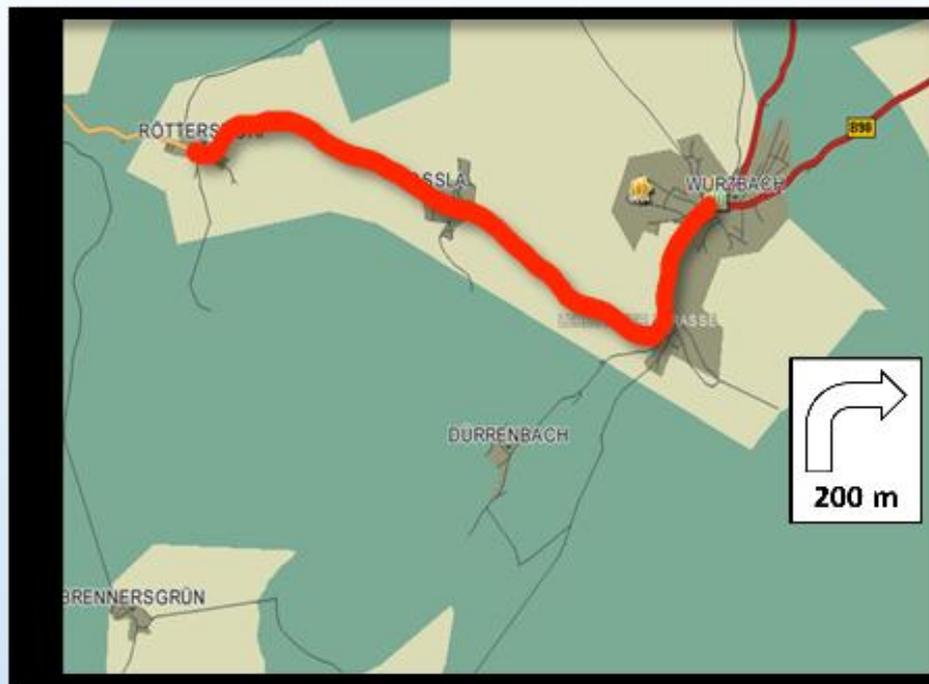


Overlay



Driver Assistance

- Navigation Layer
 - Contains Stack of Surfaces & Layout



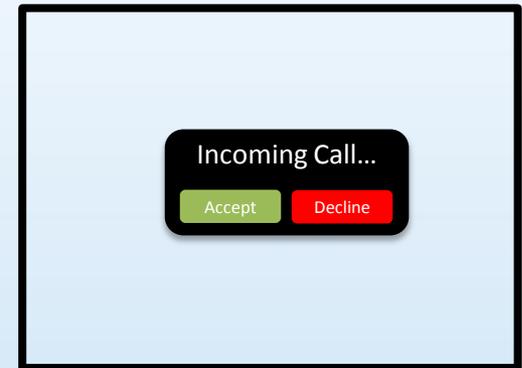
- Screen
 - Contains Stack of Layers & Layout
 - Combine visible Layers to final Picture



HMI



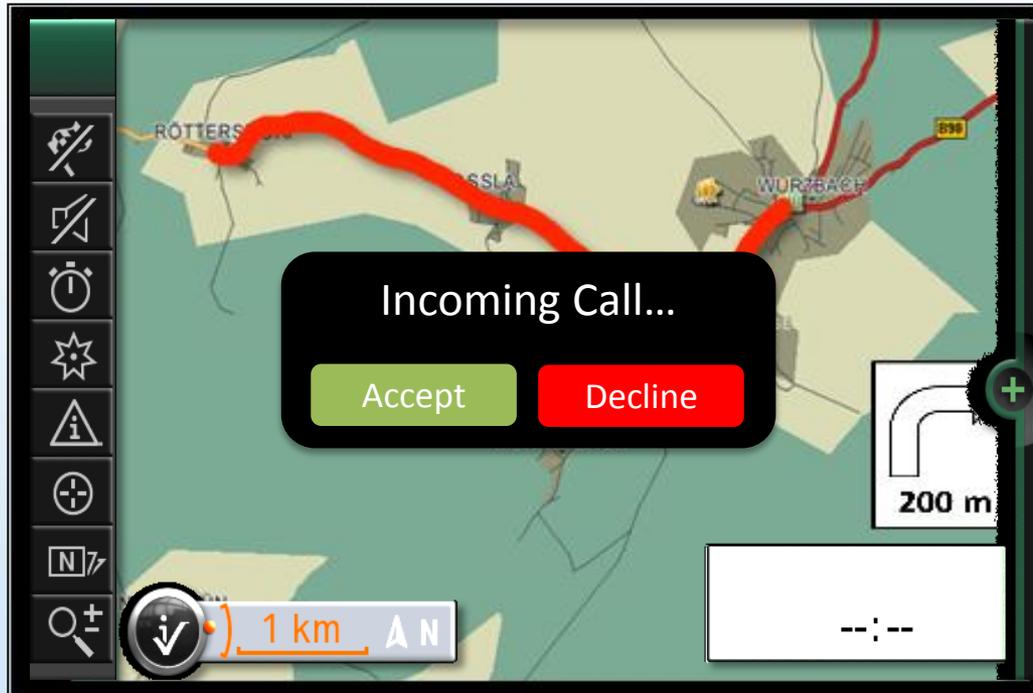
Navigation



Popup Manager

Scene Compositing Example 4/4

- Compositing Result (for one Screen)



Synchronized Surfaces

- Typical Use Case
 - Day / Night Switch
- Requested by HMI EG
- List of Surfaces can be marked synchronized
 - Compositing is stopped for synchronized surfaces, until each surface was updated

Updated Health Monitoring

- Thread deadlock detection was added
 - Requires support from Plugins
 - Reference Plugins have been updated
- Reference Health Monitor
 - Systemd based
 - Stops triggering system watchdog, if thread deadlock is detected



DirectFB Support

- (presented in Part 2)

- Too many to list here...
- We know, IVI LayerManagement is at least used on
 - ARM, X86 32/64bit
 - Linux, QNX, Cygwin
 - OpenGL ES 2, OpenGL, 2D Blitting Engines
 - X11, Wayland, DirectFB, OpenKODE

- IVI LayerManagement
 - Stabilization
 - OpenGL ES 3.0 Reference Renderer
 - New Features as required/requested
 - Update GENIVI compliance
 - Enable Other Compositors to GENIVI Compliant
- Wayland + IVI Extension
 - (presented in Part 3)

Agenda

1. IVI LayerManagement
 - Introduction & Status (Lotterbach)
2. DirectFB & IVI LayerManagement
 - Status & Demo (Hayama)
3. Wayland IVI Extension
 - Introduction, Status & Demo (Tanibata)

What is it all about?

- Genivi IVI Layer Management implemented on DirectFB, an open source graphics middleware.
 - <http://www.directfb.org/>
- DirectFB natively supports:
 - Hardware layers,
 - GPU acceleration of graphics rendering with software fallbacks,
 - Window system,
 - Input event handling,
 - And more.

Who is contributing?

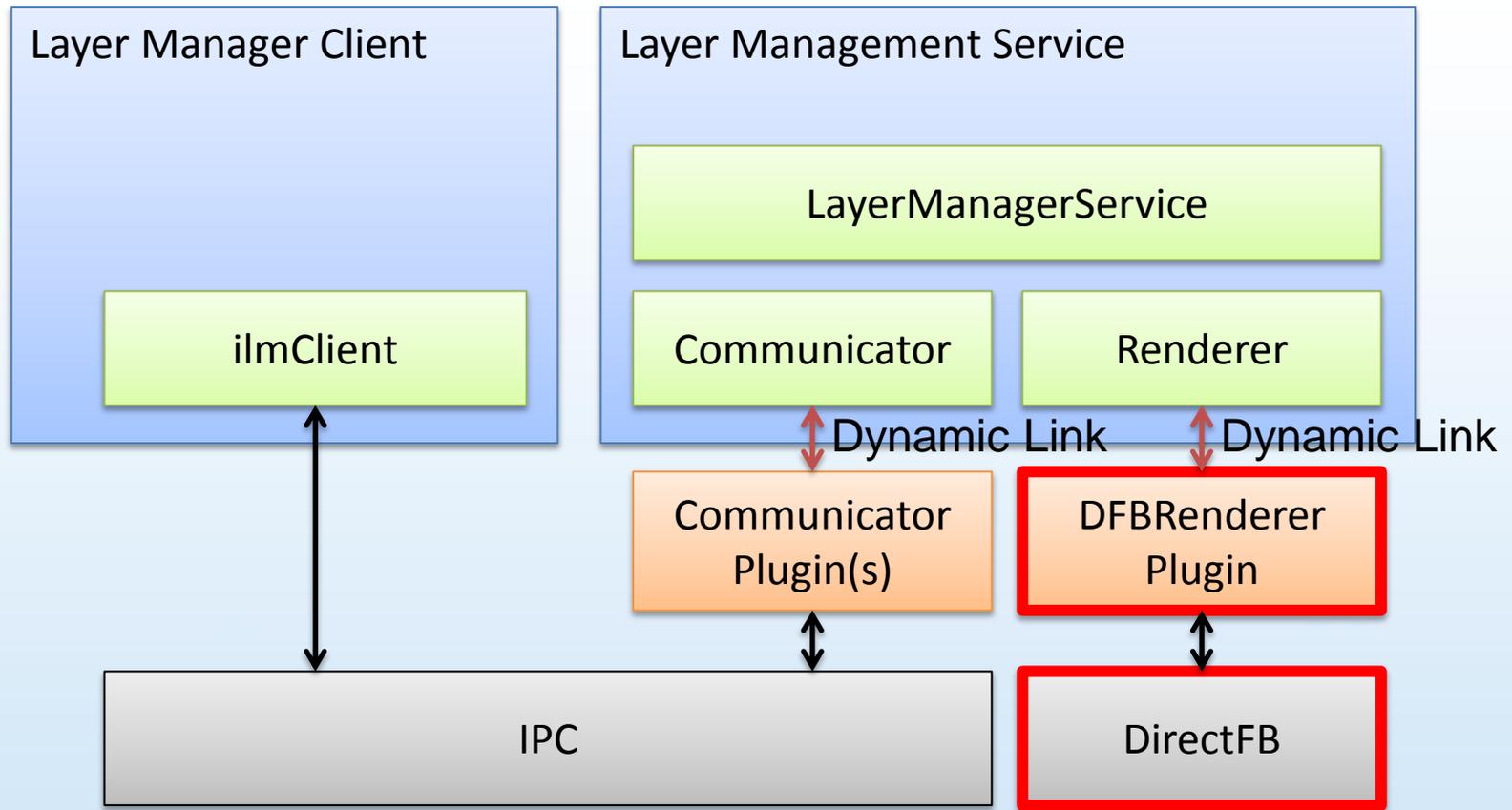
- Main Contributor
 - DirectFB integrated media GmbH
 - A joint venture found by Denis Oliver Kropp, a maintainer of DirectFB, and IGEL Co.,Ltd in 2012.
- This project is funded by Renesas



Status of DirectFB Genivi IVI LM

- The first contributions are made on October 1st, 2013 by Marek Pikarski from DirectFB integrated media GmbH.
 - 8 patches are sent to the Genivi IVI LM mailing list.
- All patches are accepted and got merged on October 2nd, 2013!
 - Thanks to Timo for reviewing the patches.

Where does DirectFB fit?



- Genivi IVI LM with DirectFB Renderer
 - GLESGraphicSystem
 - DFBGraphicSystem
- Running on Renesas' R-CarH1
 - Supports hardware layers via the DRM/KMS for R-Car.
 - The driver is available in the upstream Linux kernel.



Click to edit Master title style

DIRECTFB

What is DirectFB?

- Lightweight graphics middleware.
 - Small footprint (< 1.5MB core for ARM in version 1.7.0)
 - Licensed under LGPL.
- Supported features:
 - Graphics rendering with hardware acceleration.
 - All rendering primitives have software fallbacks.
 - Optimized software rendering, e.g. use of MMX.
 - OpenGL/ES2 support.
 - Multi-process, multi-layer, input handling, windowing etc.
 - Qt integration available.

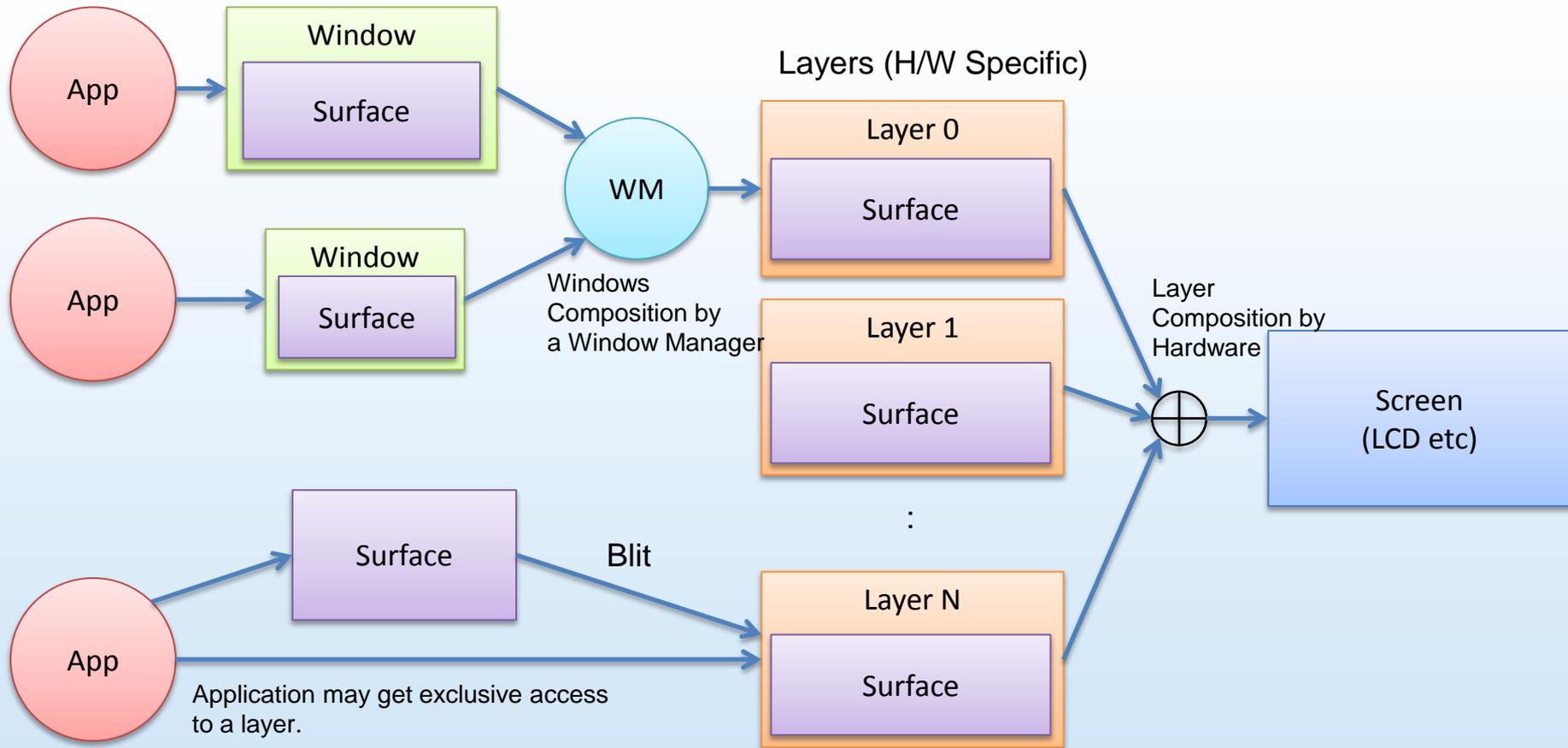


DirectFB

Key Components

- DirectFB Core
 - Graphic renderings
 - Input managements
 - Frame buffer managements
 - Graphics hardware managements
- Fusion, One, FusionDale
 - IPC mechanism
- SaWMan
 - Application Management Framework

Screen, Layer, Window, and Surface



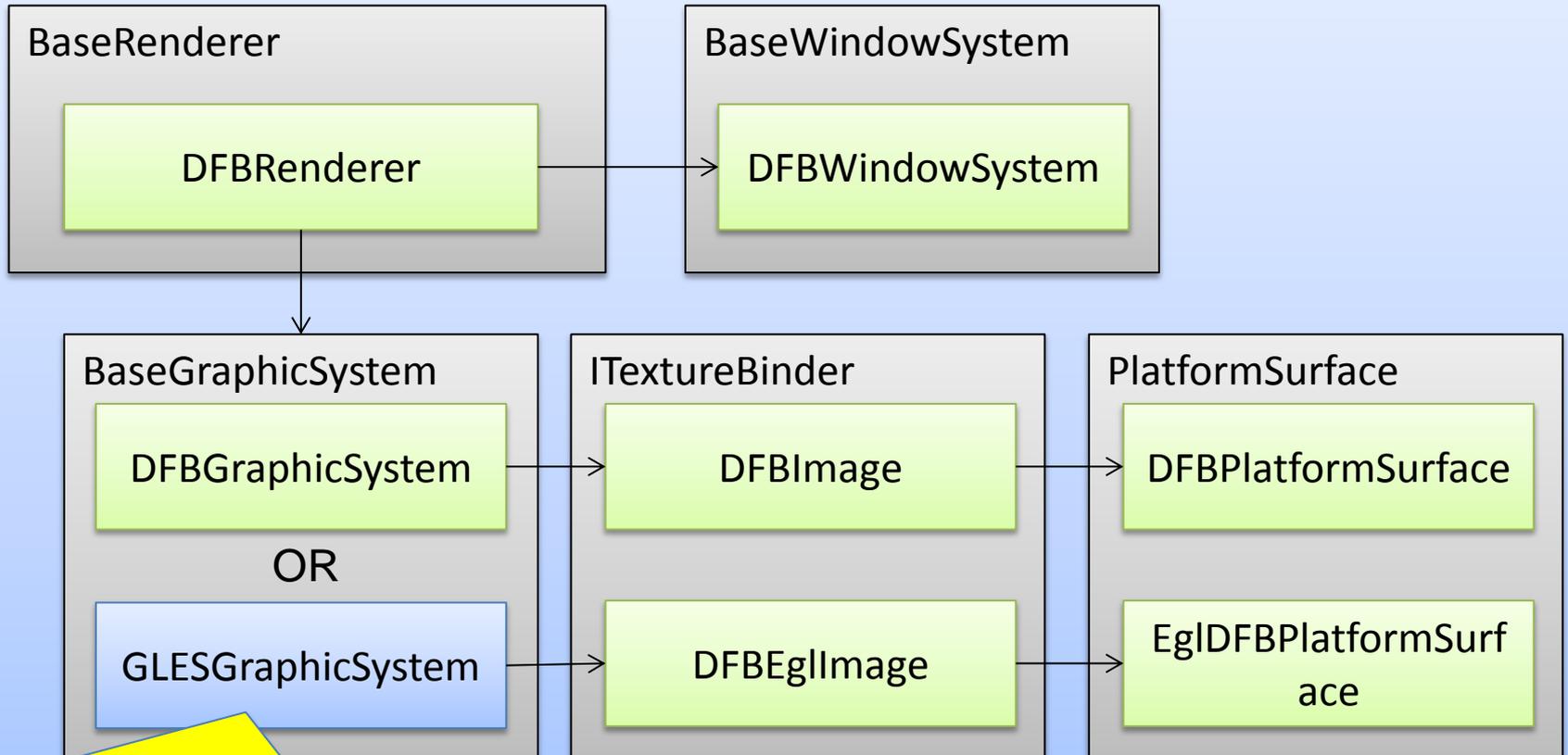


Click to edit Master title style

DIRECTFB RENDERER DETAILS

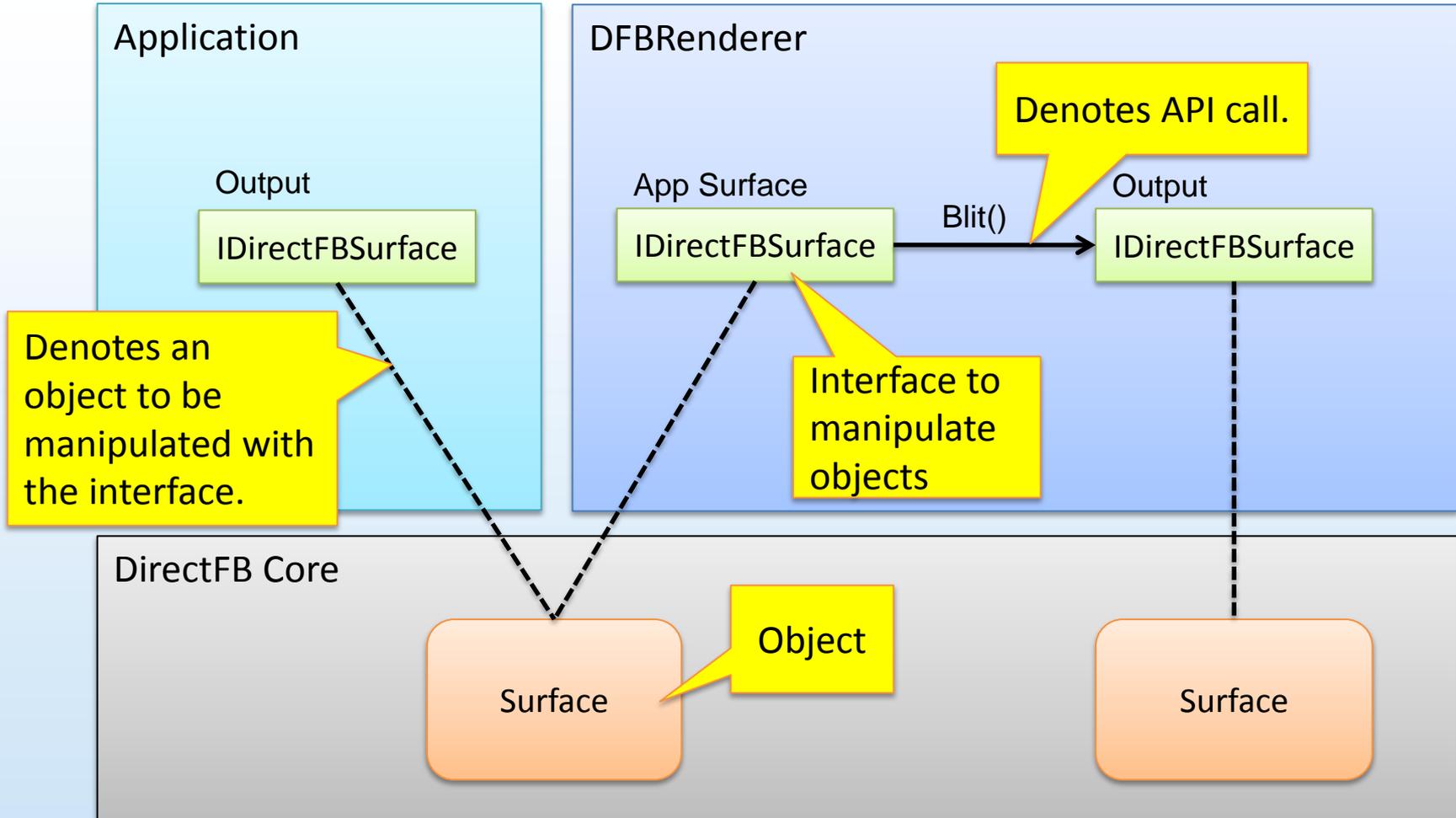
DFBRenderer Architecture

DFBRenderer Plugin



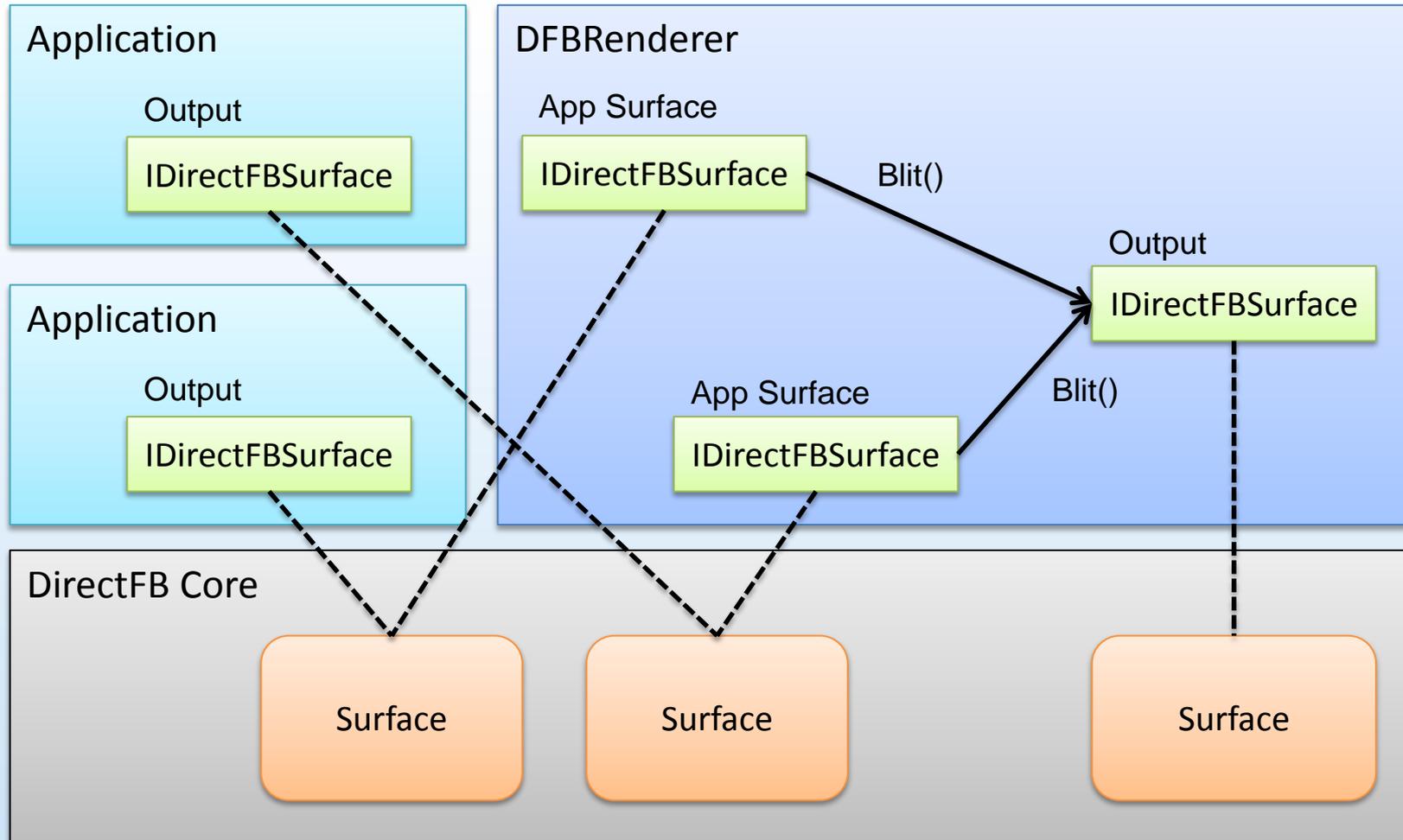
GL ESGraphicSystem is selected if an environment variable `IVI_DFBEGL` is set.

Before Going to Details...



DirectFB Graphic System

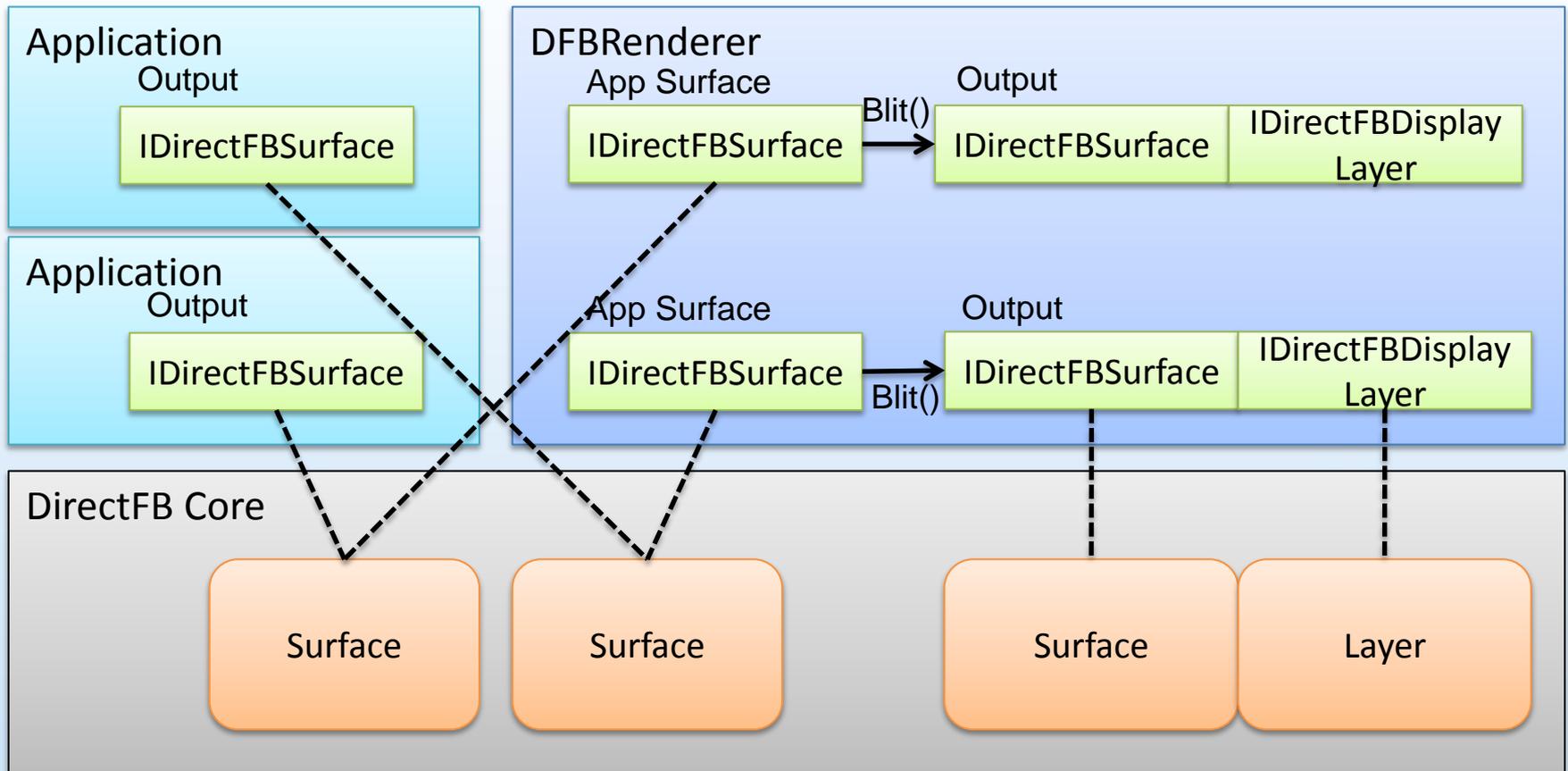
Accelerated 2D compositing including scaling, blending, chroma key, ...



DirectFB Graphic System

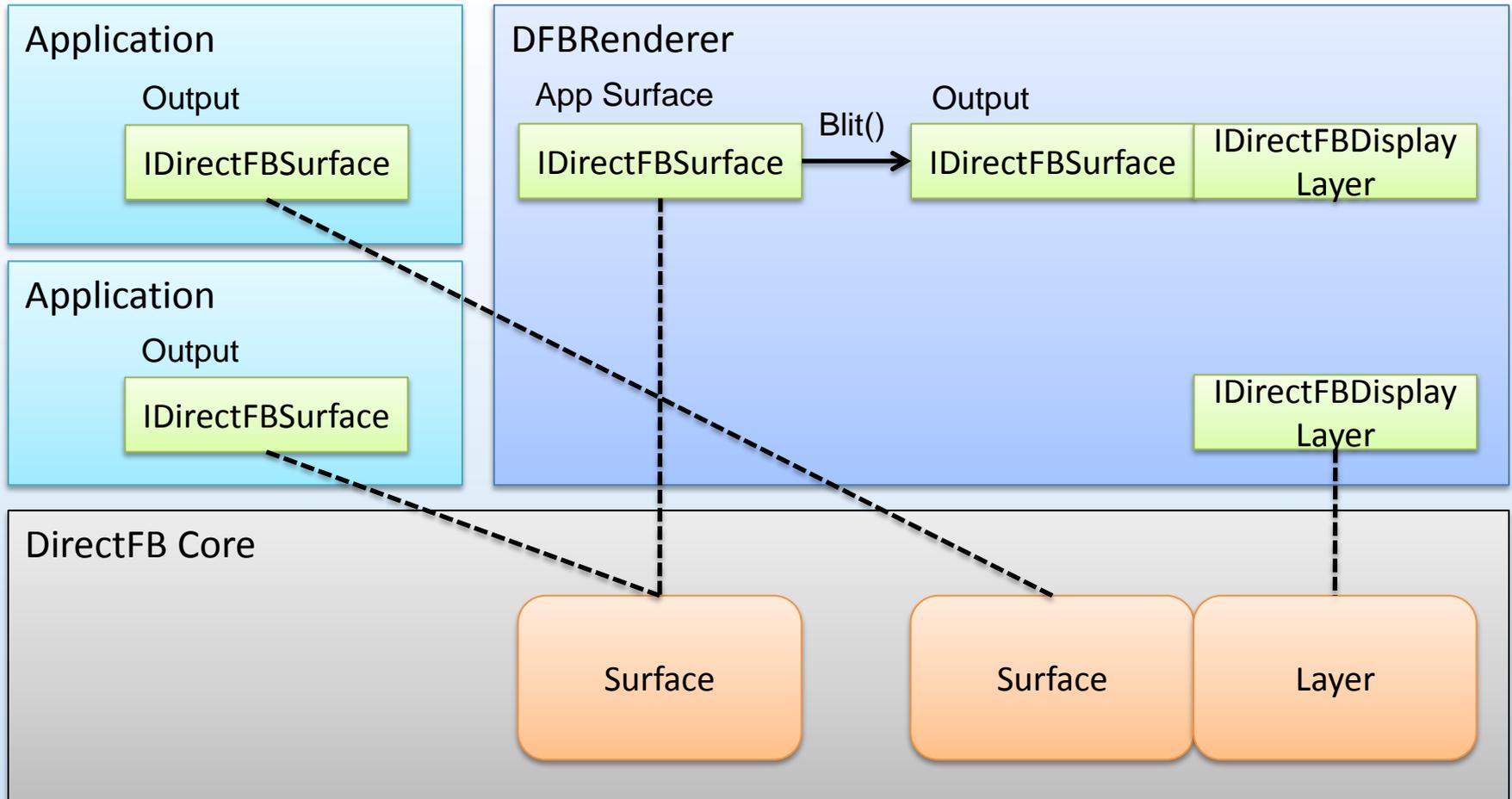
Real-time composition in hardware layers

- fixed set of planes with same or different capabilities
- pixel format, opacity, source, dest, blend, key, orientation



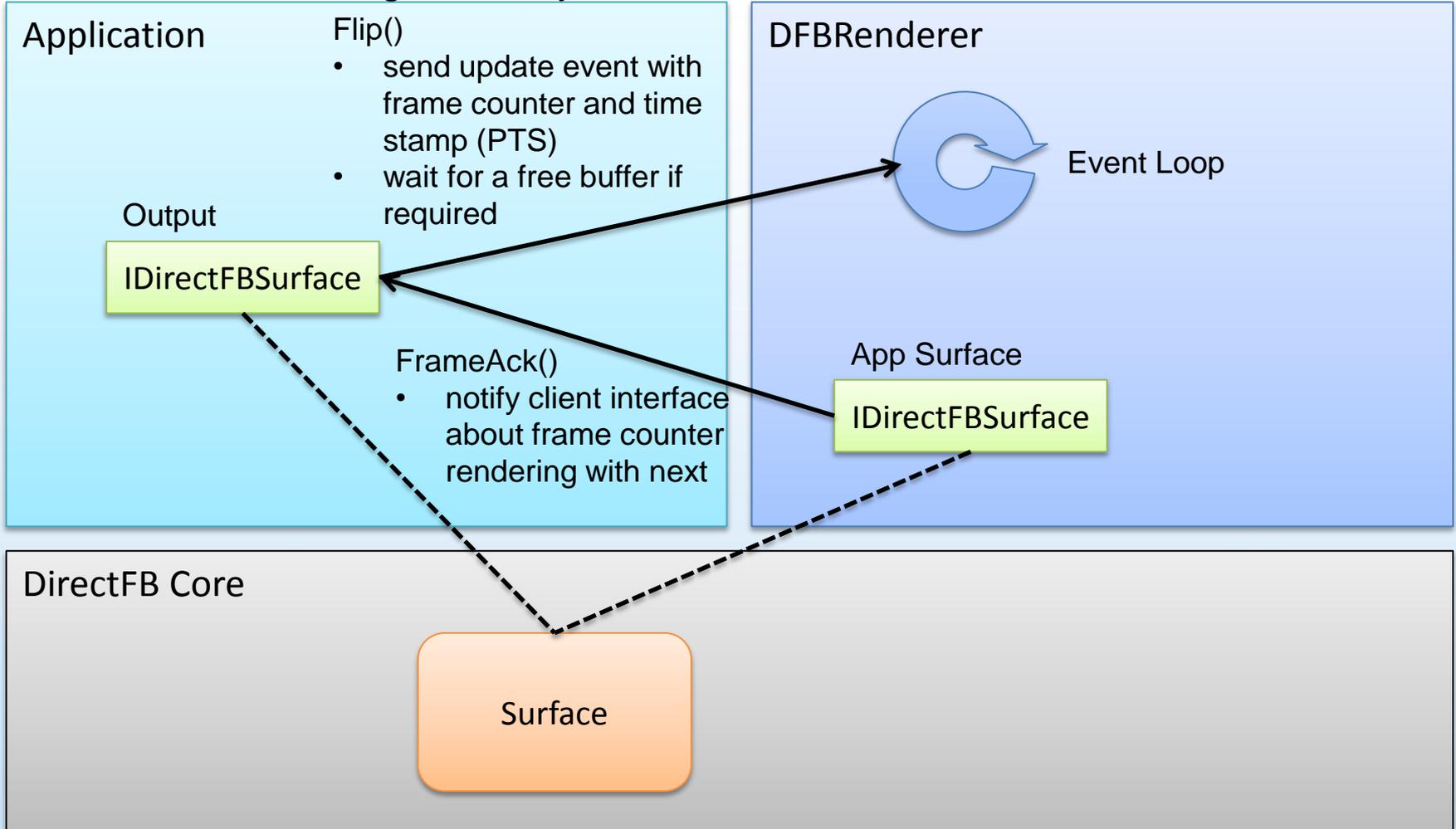
DirectFB Window System

Maximum performance clients rendering directly to the hardware layer, no need for composition by the Layer Manager itself.



DirectFB Window System

Internal frame management, synchronization with clients.

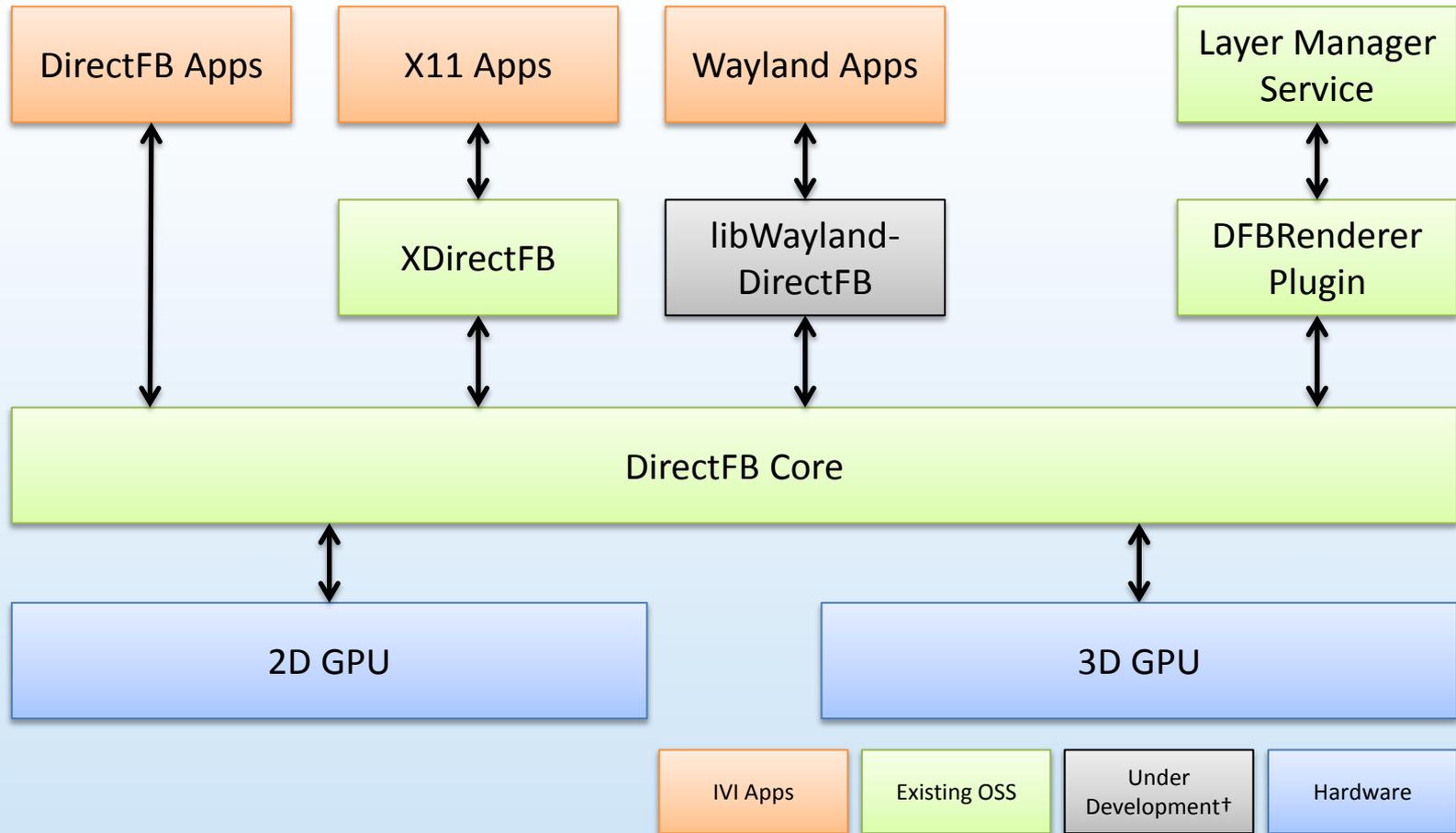




DirectFB Advantages for GENIVI LM

- Portable across platforms/APIs (no diversity needed in Layer Manager)
 - Abstract Hardware Display Layer API for Layer Manager's HW Layers
- Accelerated 2D Rendering on various hardware and software
 - DFBGraphicSystem does not require EGL or GLES
 - DirectFB 2D Clients also accelerated via GLES2 if available
- Crash safe Layer Manager by running as a Client (can restart/attach)
- Optimised performance for shared memory based Clients (non-EGL, e.g. SW)
 - DirectFB's Surface Core manages transfers between buffers, textures... with automatic updates on demand using `glTexSubImage2D`
- Flexible Layer Manager with Clients using different Window APIs
 - e.g. native DirectFB Clients, X11 Clients and Wayland Clients (planned)

DirectFB for Various Apps



† libWaylandDirectFB are planned to be released in H2 2013.

- DirectFB Compositor for Weston
 - The first code drop was made on October 6, 2013.
 - http://www.phoronix.com/scan.php?page=news_item&px=MTQ3ODQ
 - http://www.phoronix.com/scan.php?page=news_item&px=MTQ4MTI
- Directfb-compositor works with gl-compositor.

1. IVI LayerManagement
 - Introduction & Status (Lotterbach)

2. DirectFB & IVI LayerManagement
 - Status & Demo (Hayama)

- 3. Wayland IVI Extension**
 - Introduction, Status & Demo (Tanibata)**



Wayland IVI Extension Introduction, Status & Demo

Nobuhiko Tanibata



Trend of Graphic stacks in automotive segment

- Proprietary to Common
- Complexity to Light weight
- Wayland is one of candidates
 - Distill out functions from X server
- Trends
 - Genivi Layer management
 - Tizen IVI
 - Many companies may shift proprietary stacks to using Wayland.



Problem & Proposing solution for selecting Wayland

- **Problem:** Two candidates of Wayland implementation. Which is better for automotive system?
 - Weston: a Wayland compositor
 - Genivi Layer manager: Wayland plug-in, a Wayland compositor
- **Proposing Solution:** Consolidate two into one. Merge both advantages.

index : wayland/weston

The Weston Wayland Compositor

summary refs log tree commit diff

Branch	Commit message	Author	Age
0.85	configure: Fix build on debian by avoiding bashisms in the test command.	Eric Anholt	20 months
1.0	configure.ac: Bump version to 1.0.6	Kristian Hogsberg	6 months
1.1	xwayland: Use _exit() when exec() fails	Kristian Hogsberg	3 months
1.2	configure.ac: Bump version to 1.2.2	Kristian Hogsberg	4 weeks
master	evdev: Process touch up events of single-touch devices	Neil Roberts	26 hours
next	clients: Bring clients up to date	Kristian Hogsberg	2 years
terminal	Hold on to the scaled fonts we use	Kristian Hogsberg	3 years
wayland-demos-0.8	xserver: check whether pointer exists when exiting	Tiago Vignatti	2 years

Tag	Download	Author	Age
1.2.91	weston-1.2.91.zip weston-1.2.91.tar.gz	Kristian Hogsberg	3 days
1.2.2	weston-1.2.2.zip weston-1.2.2.tar.gz	Kristian Hogsberg	4 weeks
1.2.1	weston-1.2.1.zip weston-1.2.1.tar.gz	Kristian Hogsberg	5 weeks
1.2.0	weston-1.2.0.zip weston-1.2.0.tar.gz	Kristian Hogsberg	2 months

GENIVI Layer Management

About | Blog | Community | Documentation | Download | Contact Us | Projects

About the Project Scope & Goals

In the automotive domain, most HMI systems use their own window manager implementation. Many applications (e.g. navigation, reverse camera) are implemented standalone and therefore one service is used to composite all applications to final image on the screen Layer Manager. The goal of this work package is to define a common API and provide a proof-of-concept implementation for the IVI Layer Management Service. The service improves the existing vendor-specific layer management implementations which have the following features:

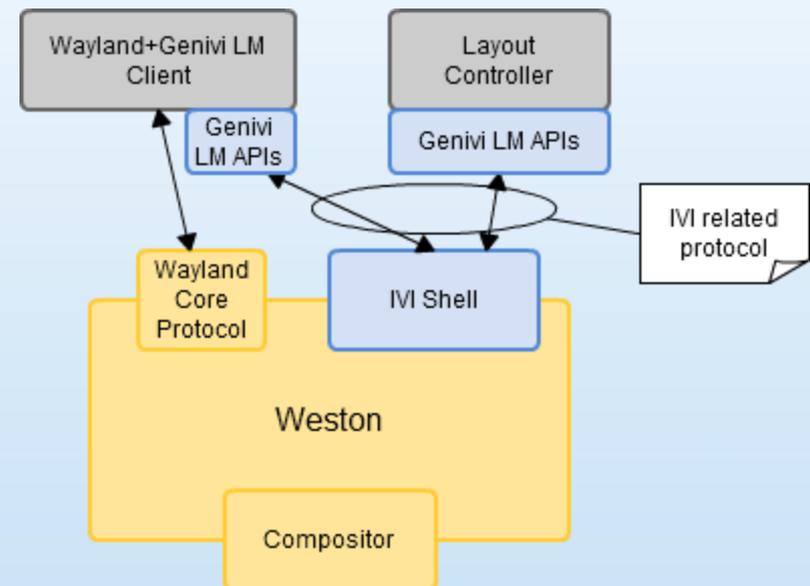
- IVI LayerManagement 1.1 available
timo.lotterbach – 23 Aug 2013 – 04:15
- IVI LayerManagement 1.1 release candidate available
timo.lotterbach – 13 Aug 2013 – 08:39
- IVI LayerManagement 1.0 available
timo.lotterbach – 27 Mar 2013 – 12:51
- IVI LayerManagement 1.0 release candidate 2 available
timo.lotterbach – 10 Mar 2013 – 10:10

Solution of consolidating two into one

- Define IVI related protocol and IVI shell to Wayland community.
- Genivi Layer manage APIs wrapping IVI related protocol to Layer manager community.

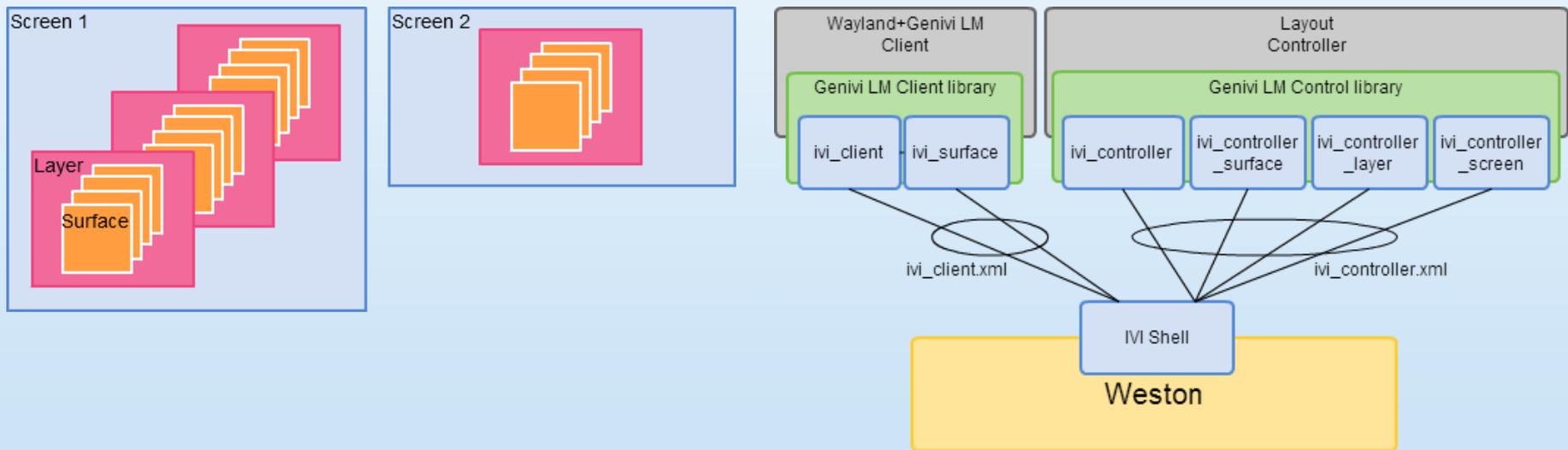
Comparison of IVI related features with current Wayland Protocol

GENIVI® Layer Management Client Interface Feature	Wayland Protocol
Surface Lifecycle Management	Yes
Surface Positioning	Yes, but through input device, not client supplied coordinate
Surface Visibility	No
Surface Opacity	No
Layer Lifecycle Management	No
Layer Positioning	No
Layer Visibility	No
Render order of surfaces in layer	No



IVI client/controller protocols

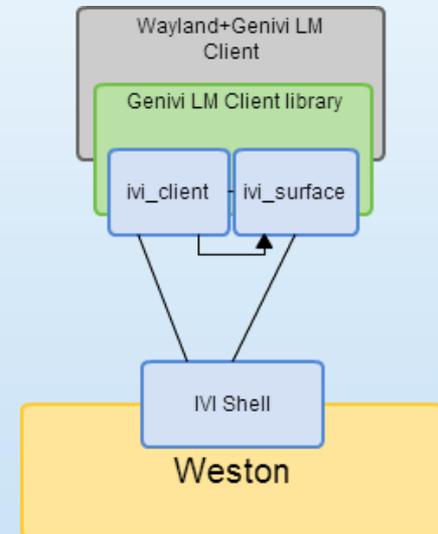
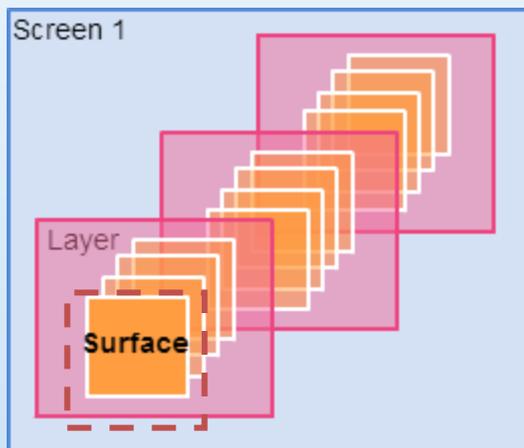
- Define IVI specific protocol to fit GENIVI layer management; managing surface->Layer->Screen.
- Client and Controller to clearly define a role of application and controller.



Use case: Wayland application set its native to ivi_surface

- ivi_client: the first protocols for creation of surface.
- ivi_surface: tie wl_surface to ivi_surface

Simple protocol to tie native and ivi_surface with global ID.
Global ID allow us to identify ivi_surface.



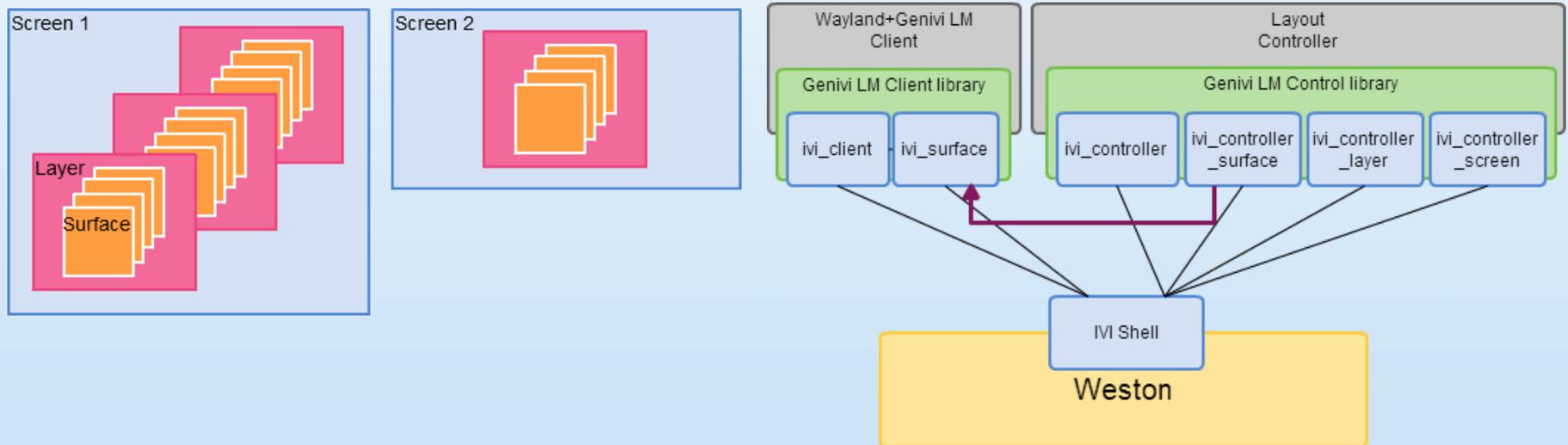


IVI controller protocol: ivi-controller.xml

under discussion in mailing list

Use case: Create layers, add surfaces to it and control them.

- `ivi_controller`: the first protocols for receiving events: creation of surface and create layer.
- `ivi_controller_surface`: set visibility e.g. in case of speed restriction.
- `ivi_controller_layer` : add/clear surfaces, set visibility, position,
- `ivi_controller_screen`: add layer to a screen

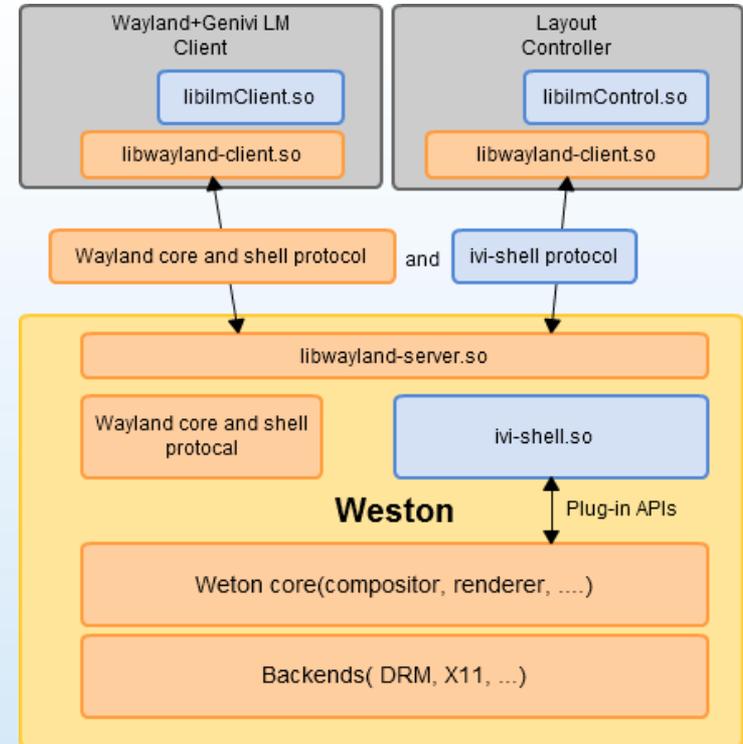




IVI client/controller Implementation

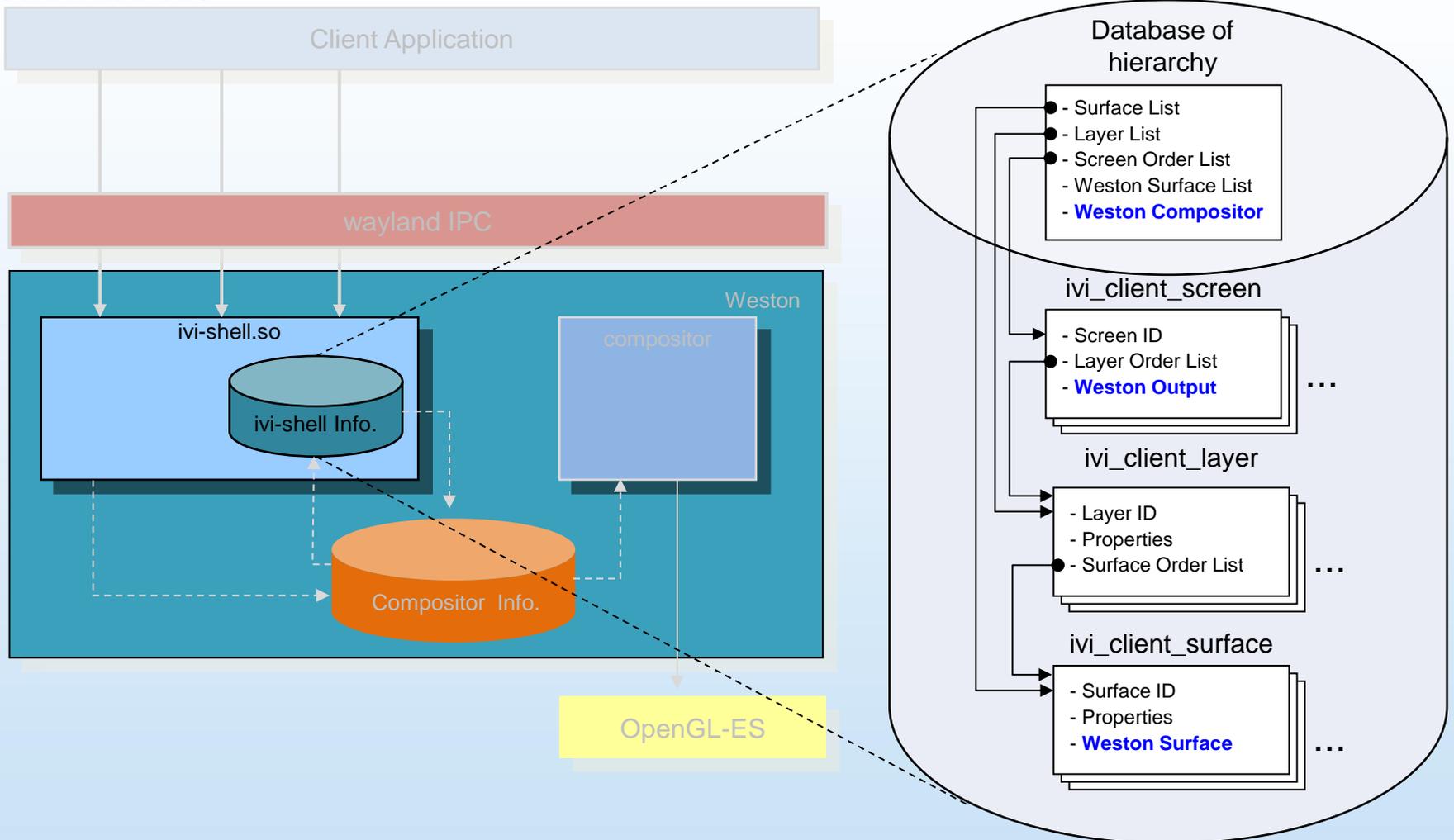
now a reference

- Ivi-shell.so:
 - Implement ivi-client/controller protocols
 - Communicate Weston by Plug-in APIs
 - Set name of plugin in weston.ini [shell] type=ivi-shell.so
- libilmClient.so
 - Wrap ivi-client protocol to be compatible with Genivi LM client APIs.
- libilmControl.so
 - Wrap ivi-controller protocol to be compatible with Genivi LM control APIs



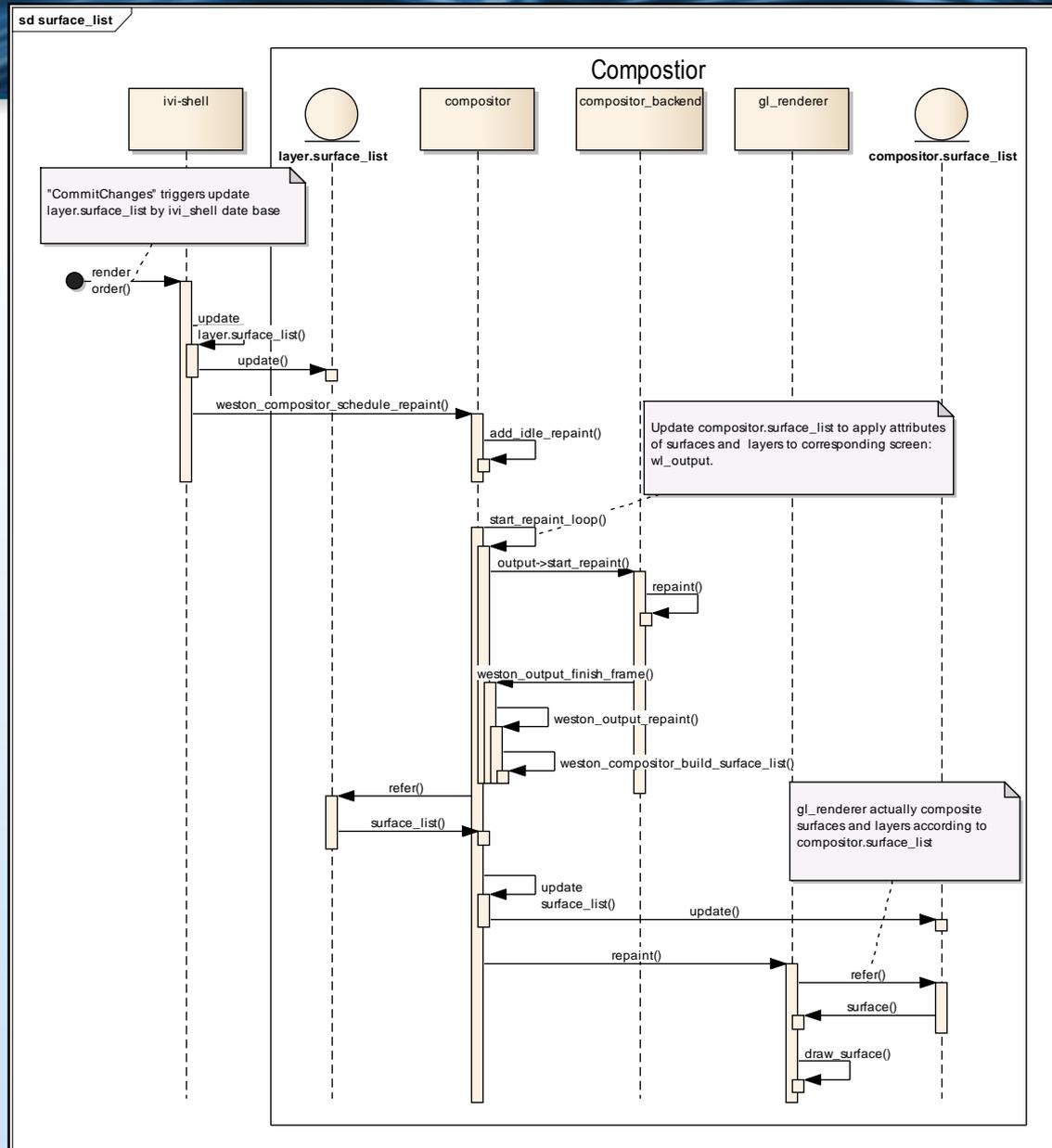
IVI-Shell property database

- ivi-shell.so contains a database to manage hierarchy and tie Weston resources.



IVI-Shell: how to composite surfaces

- IVI-shell just asks compositor to composite surface according to attributes.
- Update layer.surface_list by Database of attribute.





Compliant level under discussion

- Client APIs: Feature set, core
- Control APIs: Feature set, optional



Current status

- Genivi Layer management APIs almost supported
 - Except `getProperties`
 - Cache style vs Get style
- Sub-surfaces feature verified



Next steps

- Discussion how to merge Genivi LM apis on ivi-shell to ivi layer management.
 - Mailing list and telconf
- Discussion with other community, e.g. Wayland, AGL. I will present the same contents in Automotive Linux summit Fall.

Demonstration

- Fedora 19 32bit. Wayland/Weston 1.2.0.
- Qt, EFL, and Adobe AIR on Wayland and Layout management by Genivi Layer management APIs.
- Layout change and fade-in animation done by compositor.





Contribution

- Use case and requirements
 - Mailing list or telconf

- Wayland
 - <http://wayland.freedesktop.org/>
 - <http://cgit.freedesktop.org/wayland>
- GENIVI
 - <http://projects.genivi.org/ivi-layer-management/>