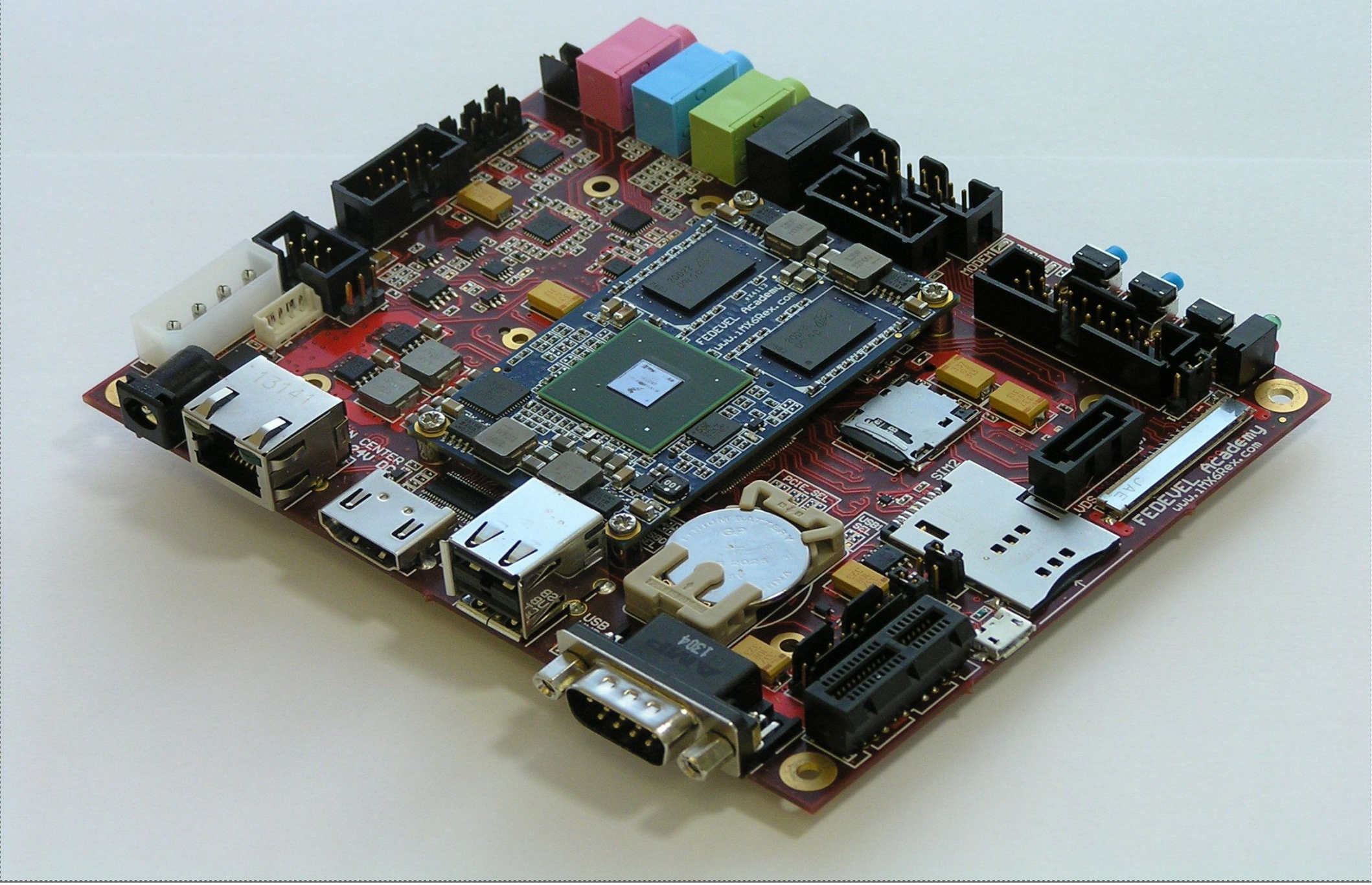
Creating Digital Products
Part 2 : Software

Cambridge Game Creators
London – 2015-05-21

Laurent Pinchart
laurent.pinchart@ideasonboard.com

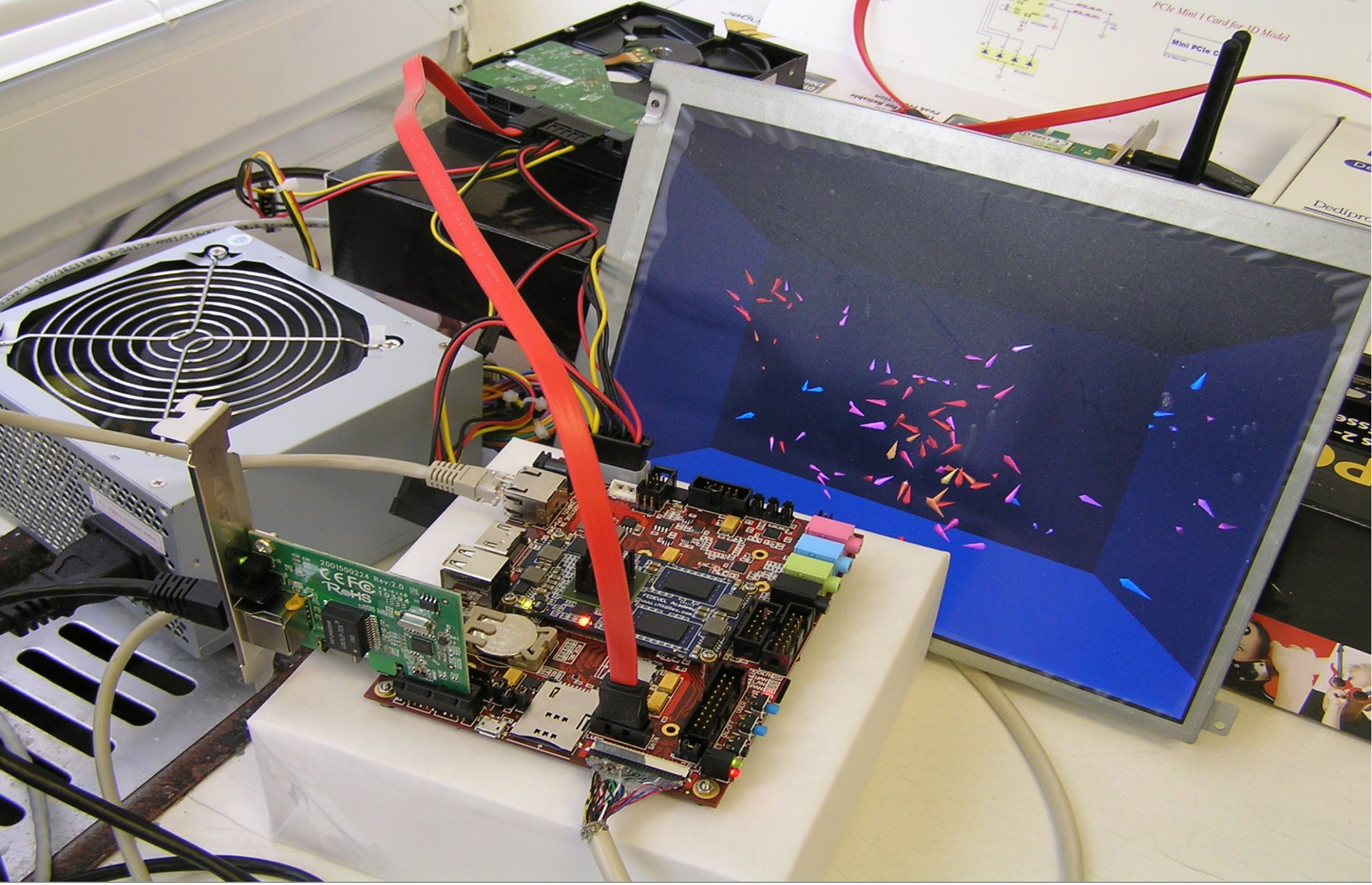


IDEAS ON BOARD

Problem Definition

CambridgeGameCreators

CAMPUS LONDON

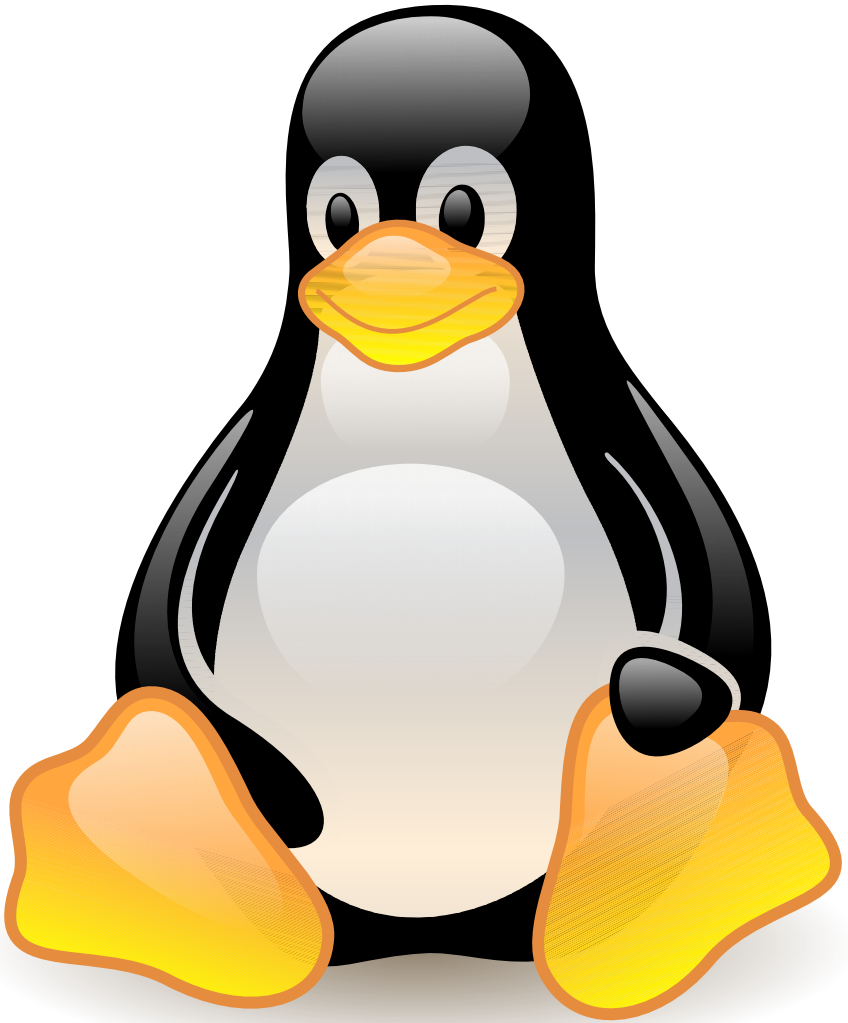


IDEAS ON BOARD

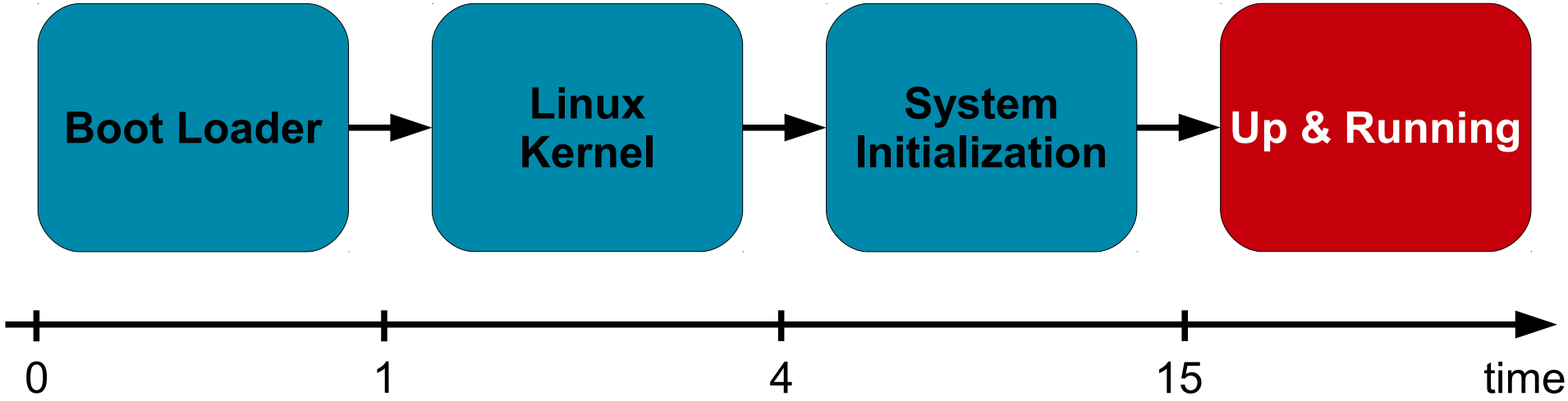
Problem Definition

CambridgeGameCreators

CAMPUS LONDON



Problem Definition


Boot Loader
Linux Kernel
System Initialization
Up & Running
0
1
4
15
time

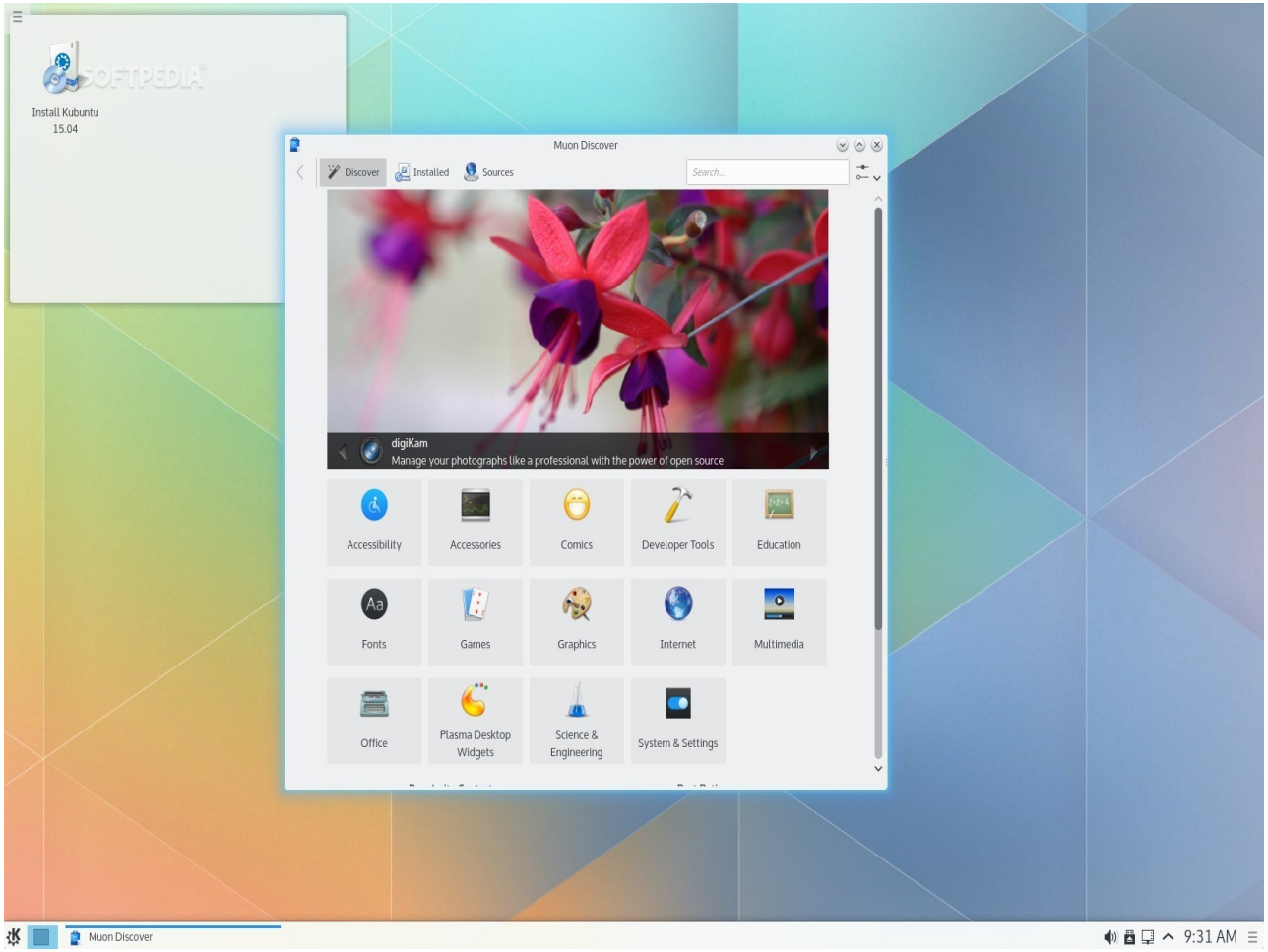

kubuntu


kubuntu

Fixed Logo

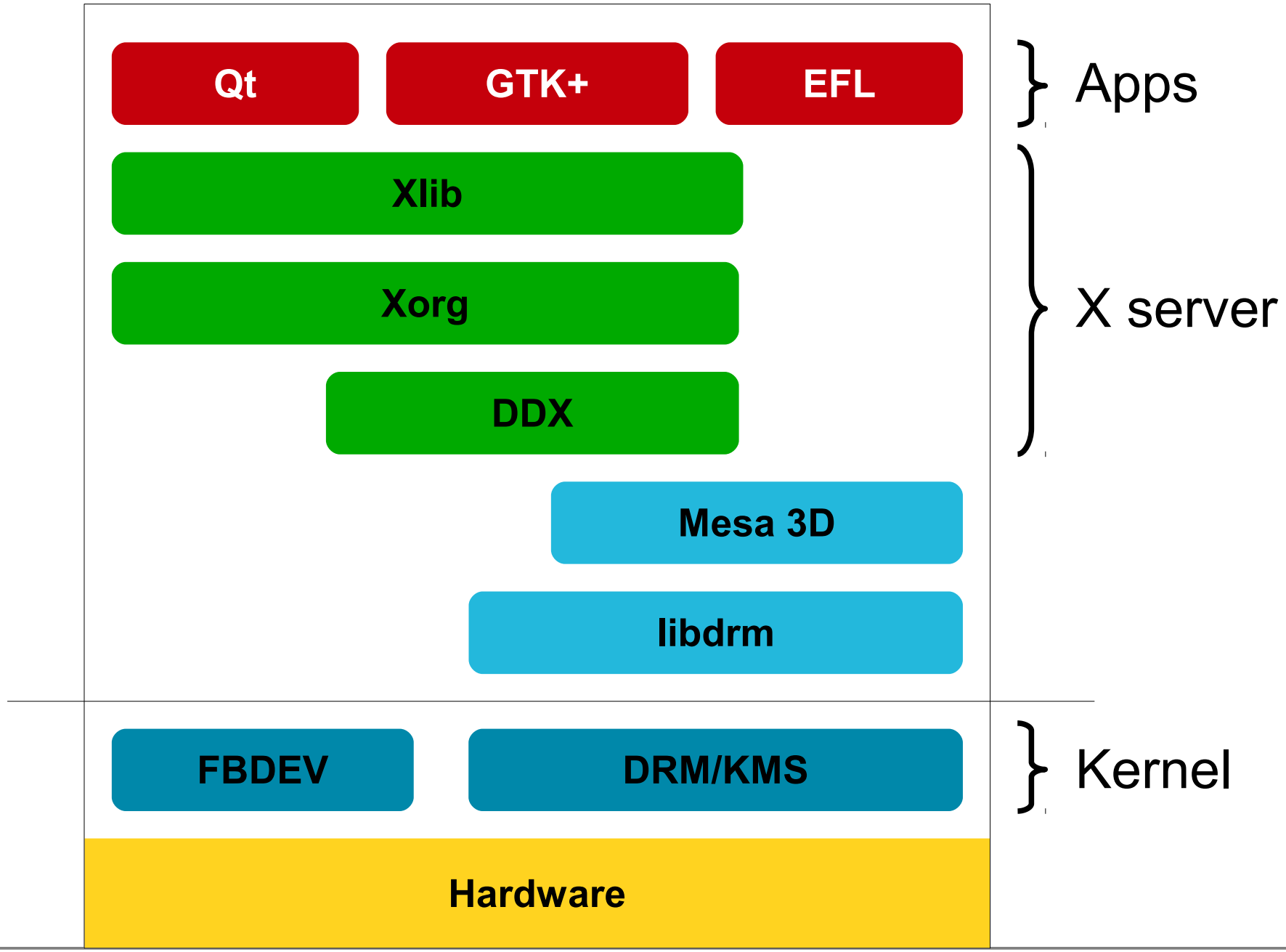
Fixed Logo

Animation, Feedback



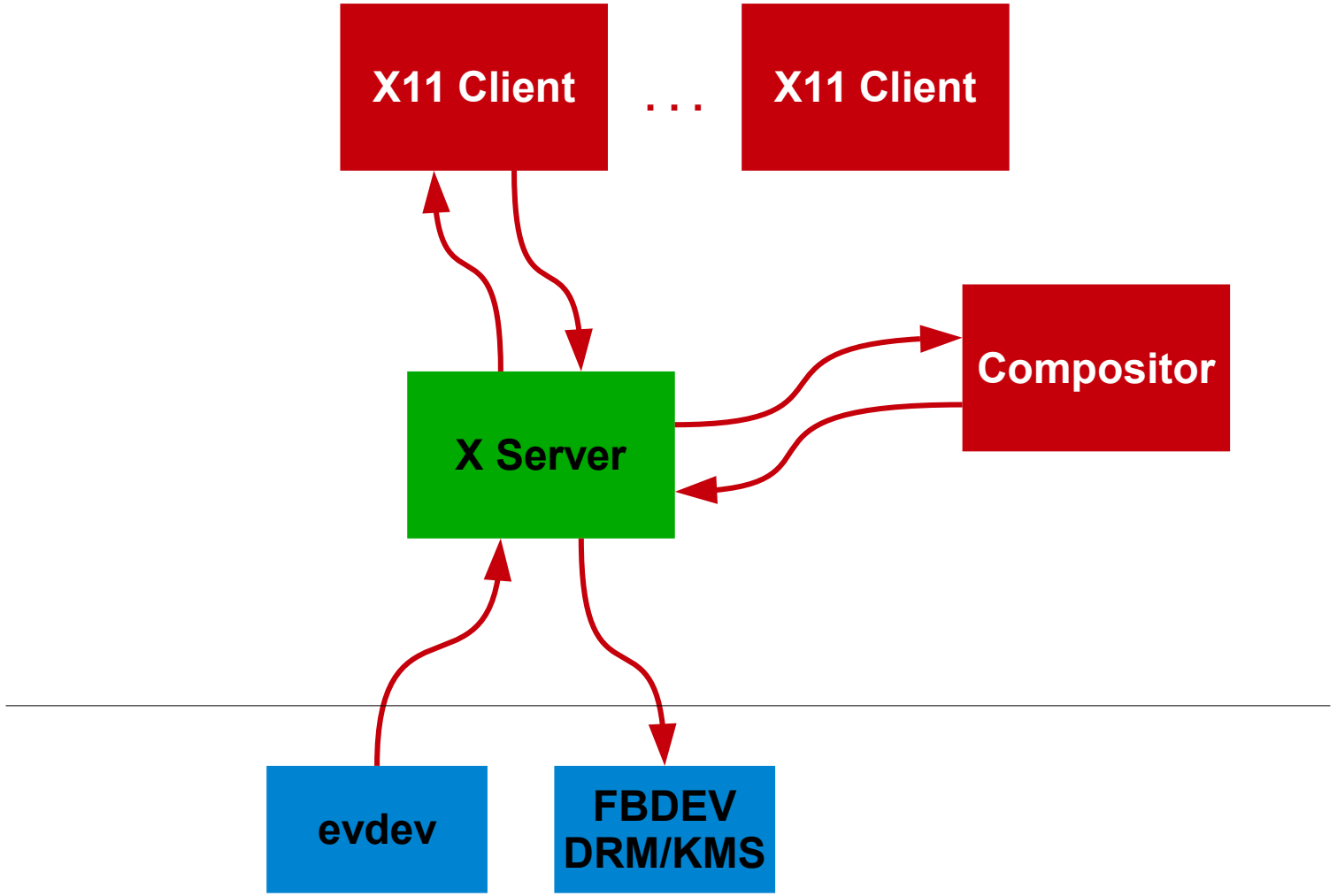
Full Graphics Stack

Traditional Linux Graphics Architecture

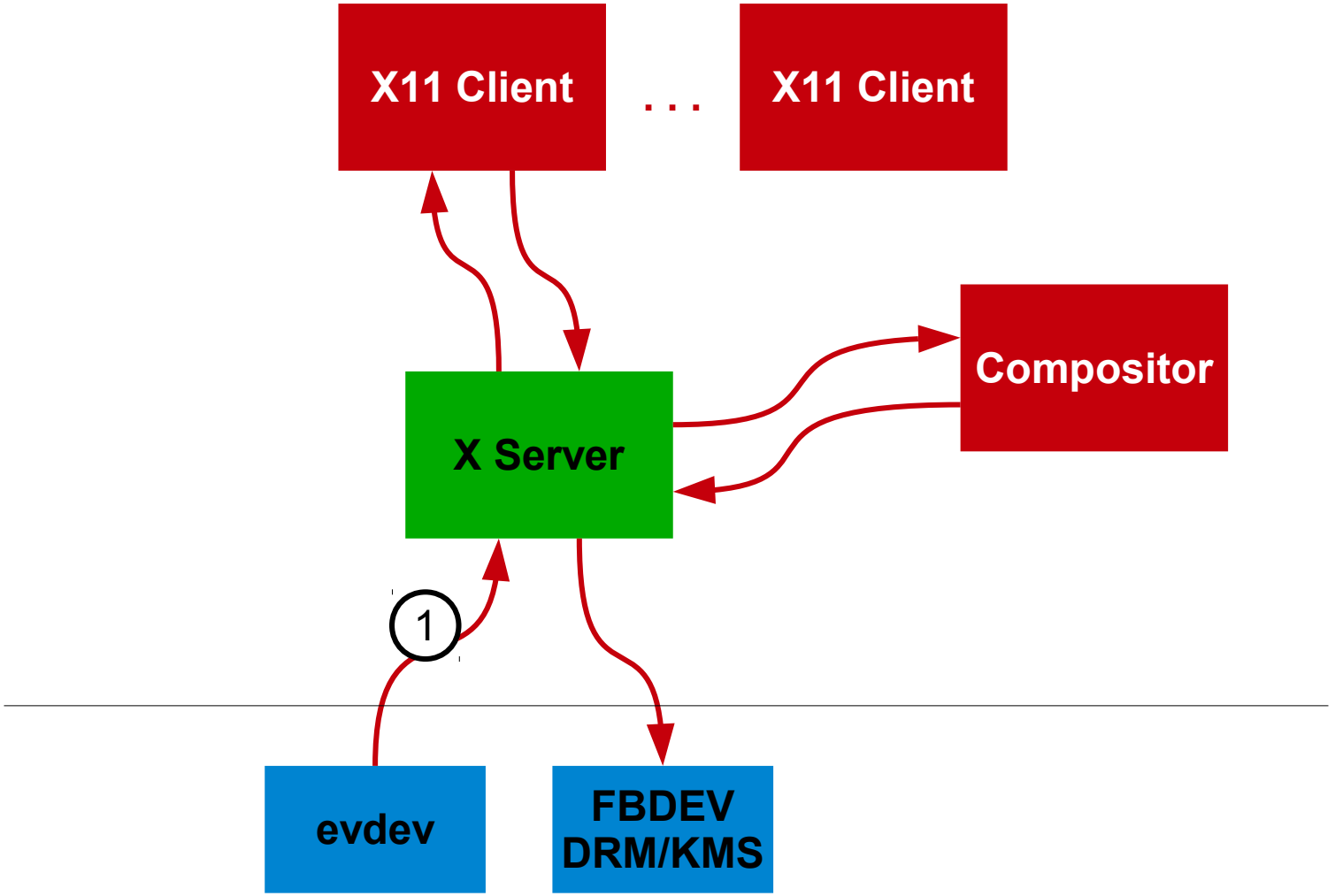


X11 Graphics Stack

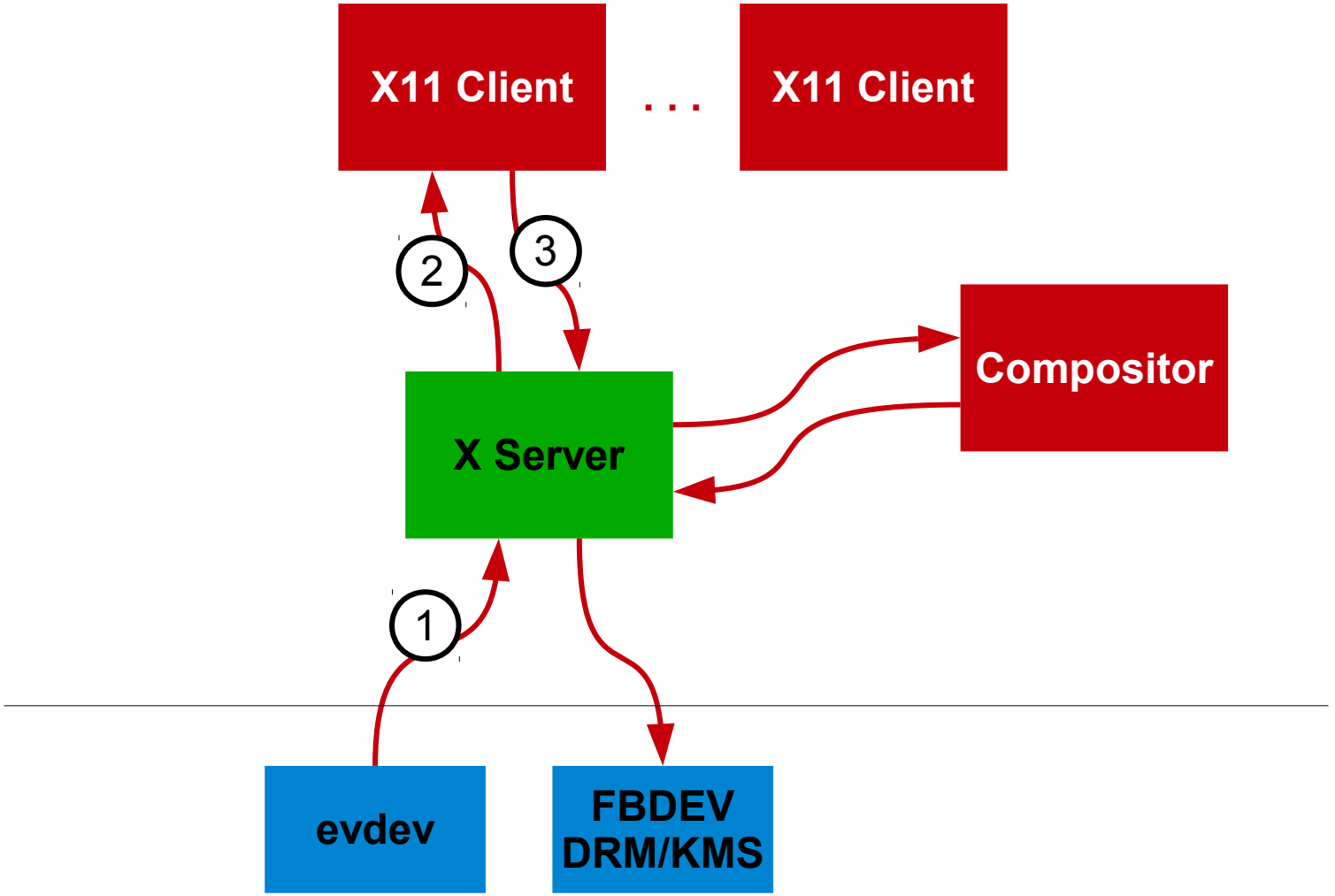
- **DDX** – Device Dependent X driver. Handles initialization, manages the display, performs 2D accelerated rendering.
- **DRM** – Direct Rendering Manager. Subsystem of the Linux kernel responsible for interfacing with GPUs.
- **EFL, GTK+, Qt** – Open-source graphical toolkits.
- **EGL** – Interface between rendering APIs and the underlying native platform windowing system.
- **FBDEV** – Frame Buffer Device. Subsystem of the Linux kernel responsible for accessing the display framebuffer.
- **GLES** – GL Embedded System. Embedded profile of the OpenGL API.
- **KMS** – Kernel Mode Setting. Subsystem of the Linux kernel responsible for interfacing with display controllers. Handles mode setting and hardware composition.
- **Mesa (3D)** – Open-source implementation of the EGL, GL, GLES, … APIs.
- **Xlib** – Reference implementation of the client side of the X11 protocol.
- **Xorg** – Reference implementation of the server side of the system.



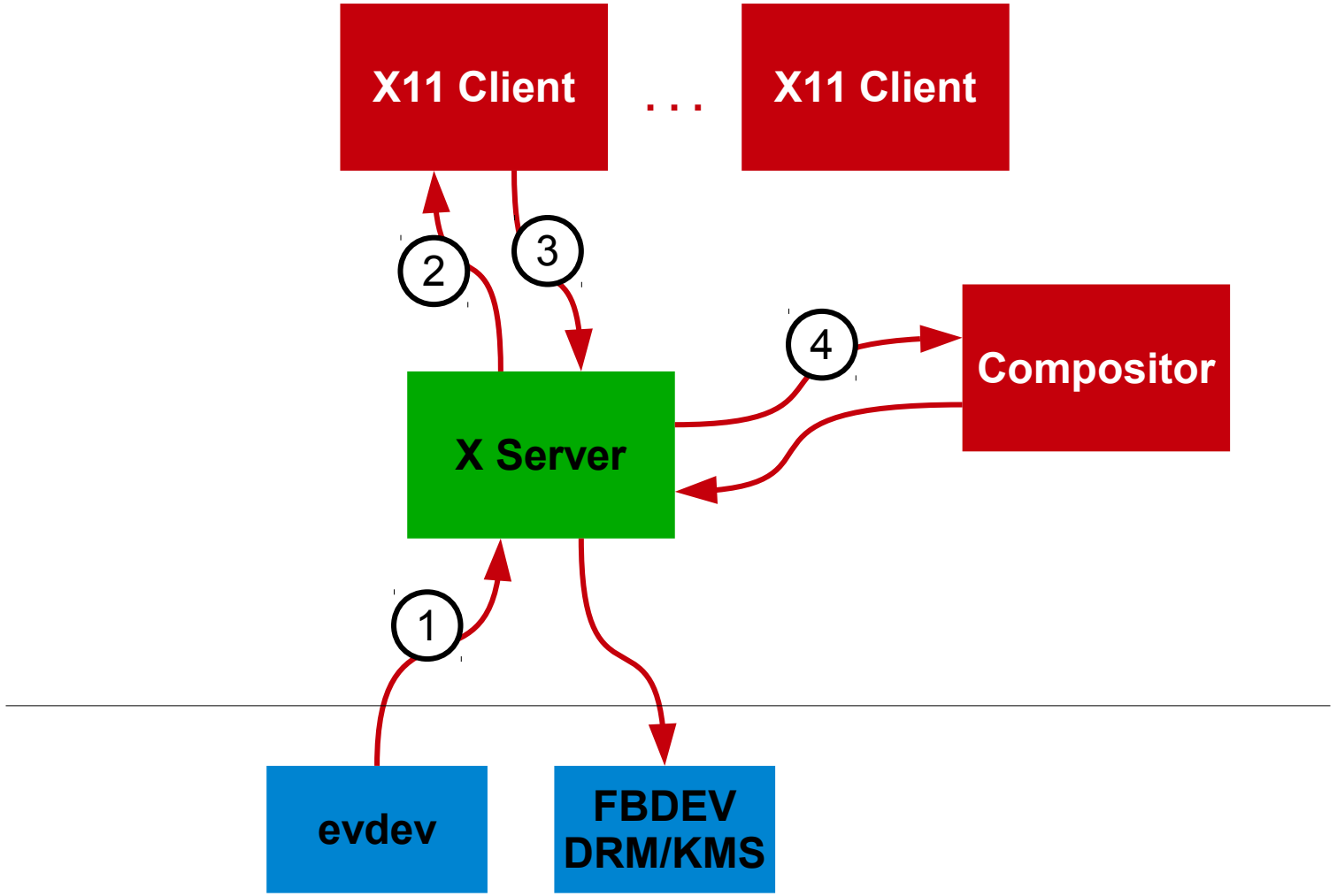
X11 Composition



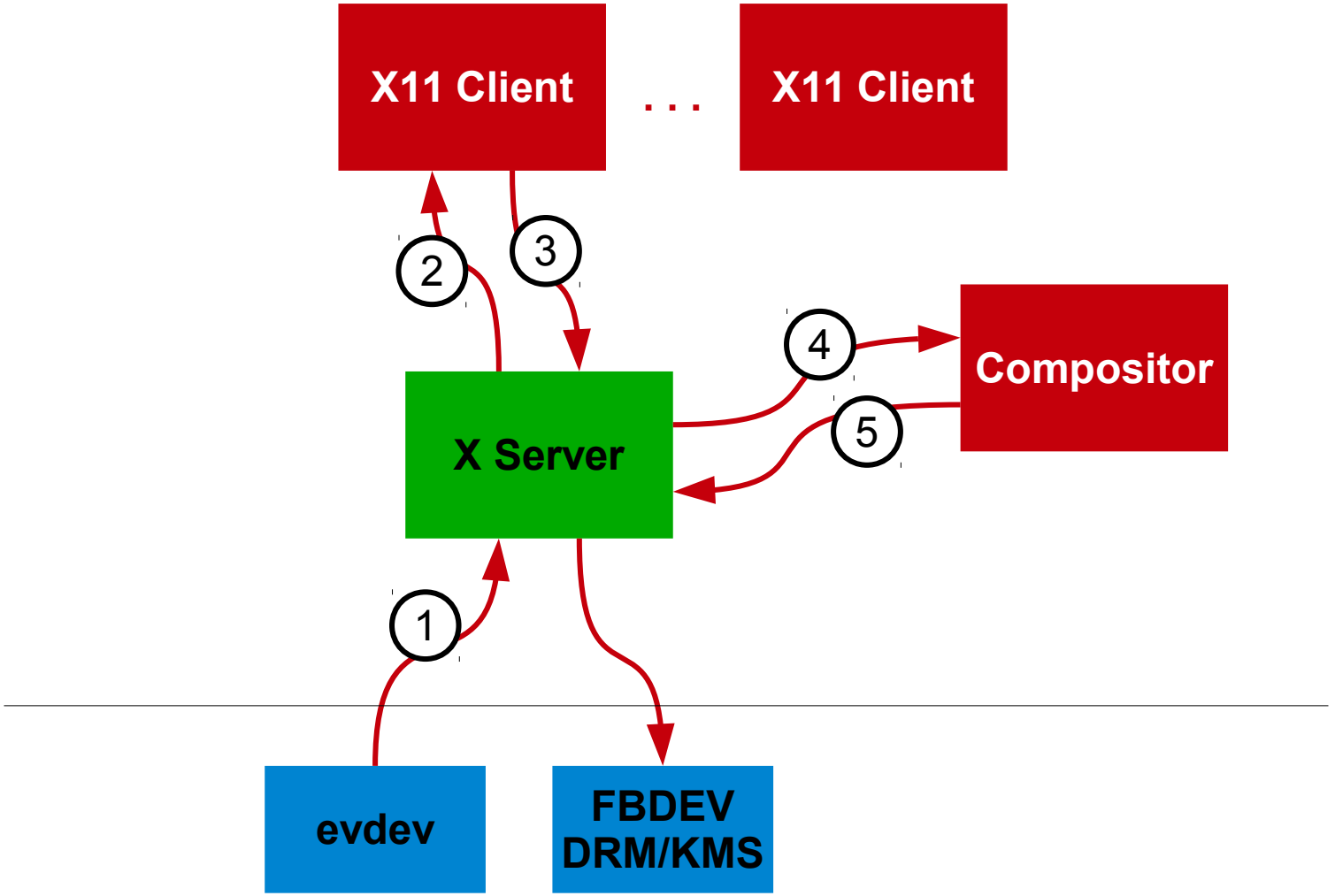
X11 Composition



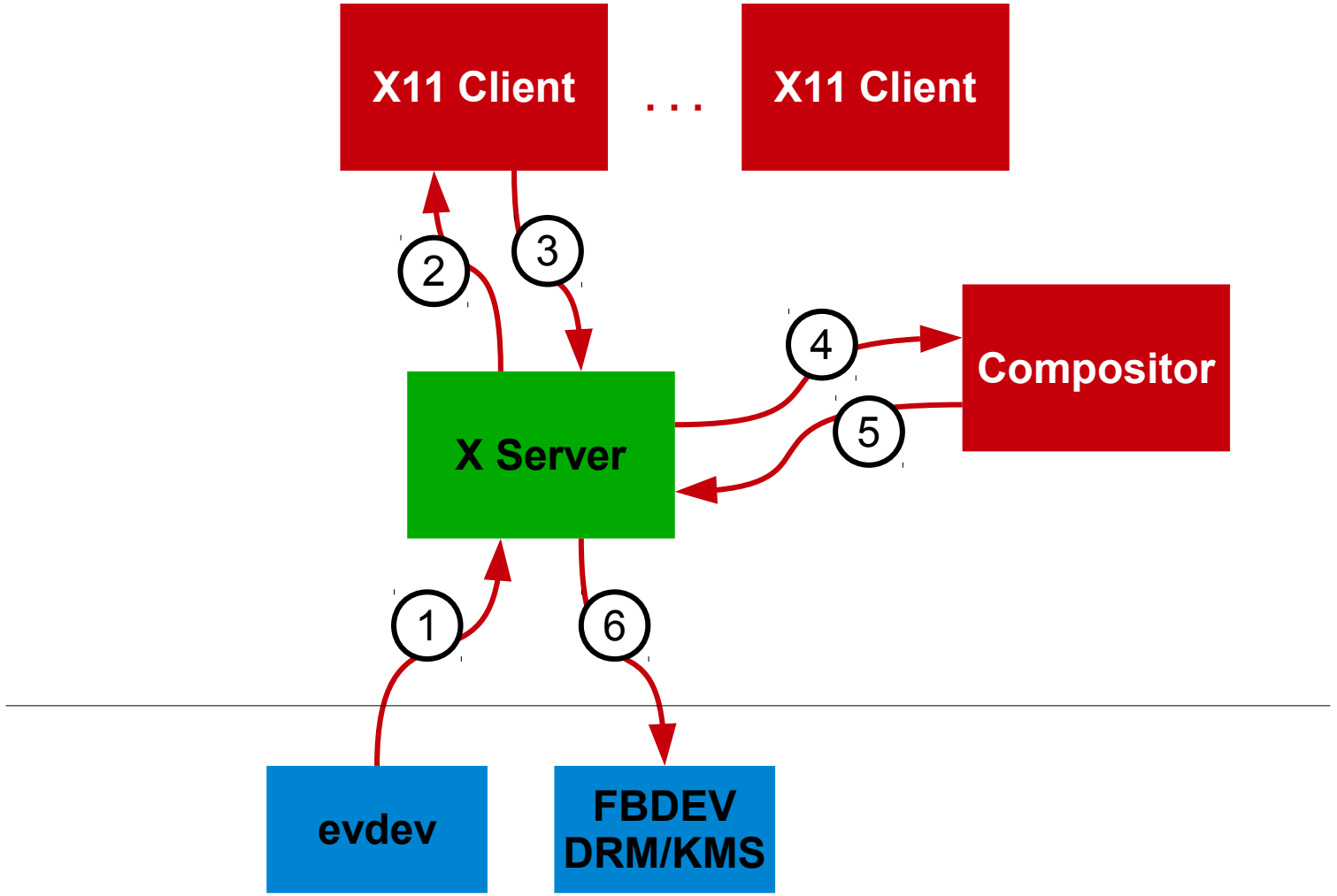
X11 Composition



X11 Composition



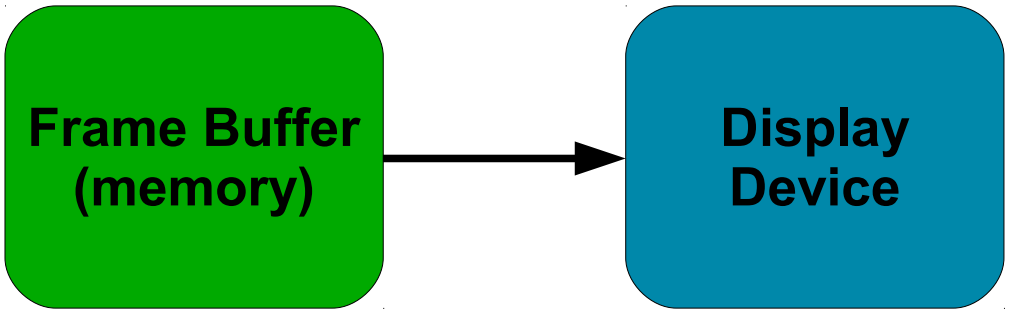
X11 Composition

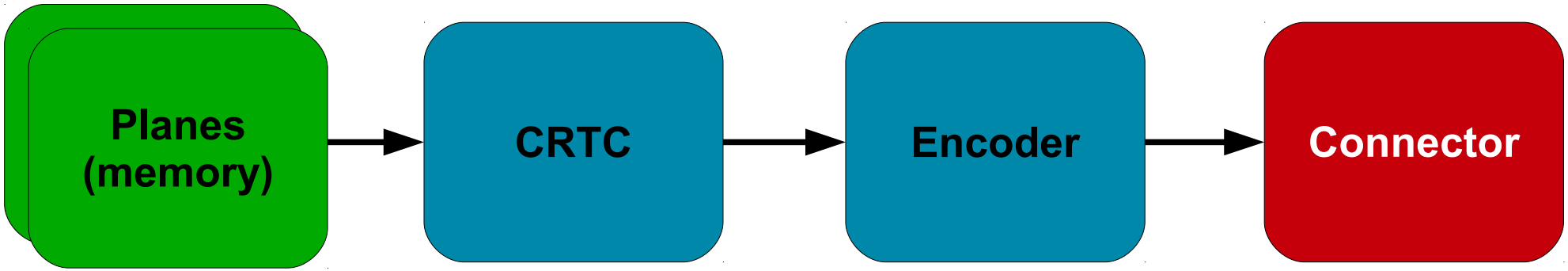


X11 Composition

Linux Kernel Display APIs

FBDEV vs. DRM


Frame Buffer
(memory)
Display
Device


Planes
(memory)
CRTC
Encoder
Connector

Source: http://xkcd.com/619/

IDEAS ON BOARD

Features

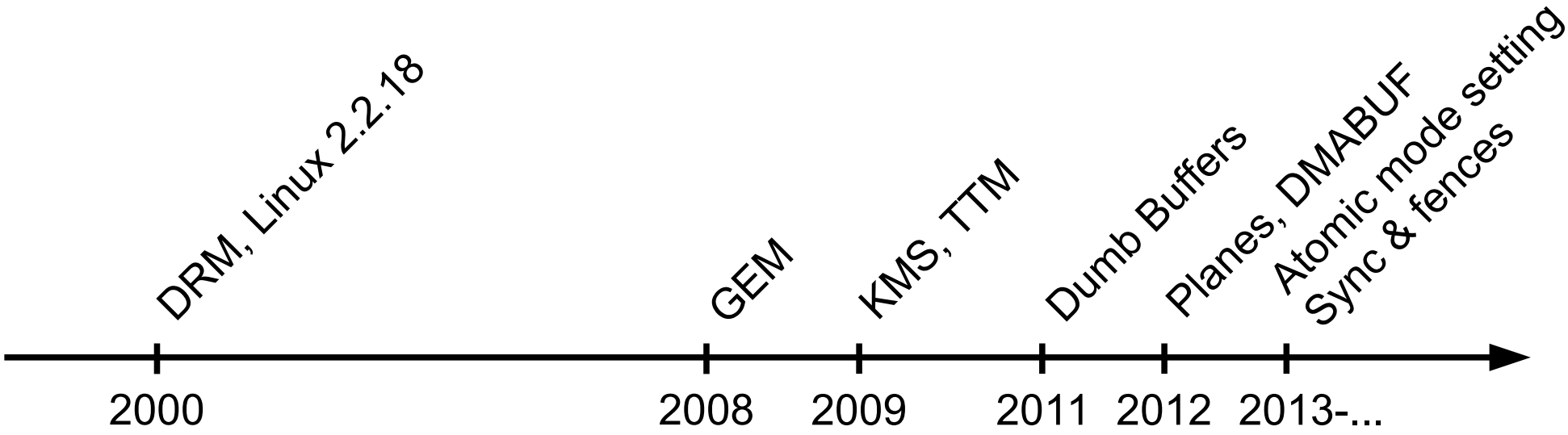
CambridgeGameCreators

CAMPUS LONDON

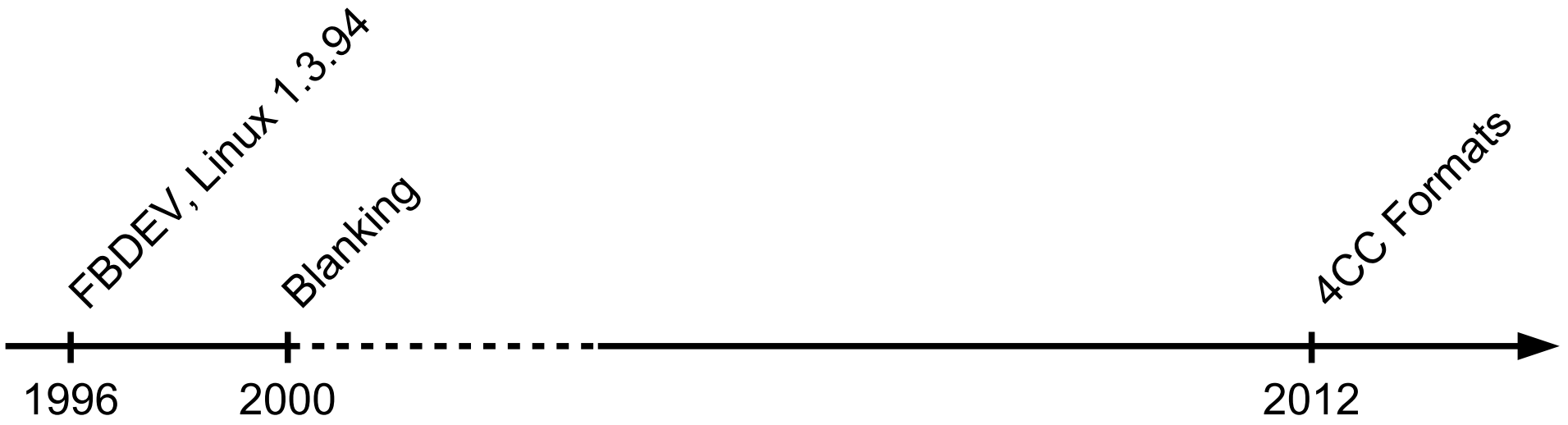
	DRM	FB
Dynamic Allocation	Yes	No
Multiple Buffers	Yes	panning
Import	dmabuf	No
Export	dmabuf mmap	mmap

	DRM	FB
Formats	4CC	RGB 4CC
Enumeration	Planes	No
Negotiation	No	No
Atomicity	Yes	No

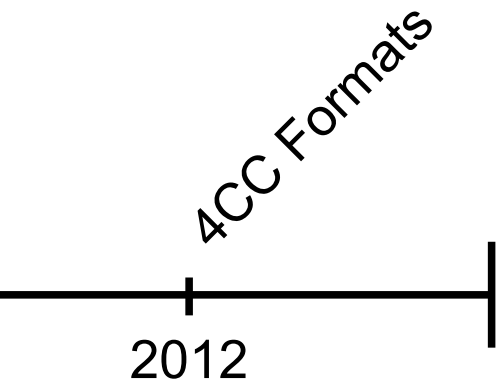
	DRM	FB
Overlays	Yes	No
Rotation	Yes	No
Scaling	Yes	No
Cropping/Panning	Yes	Yes



Activity – DRM/KMS



Activity – FBDEV

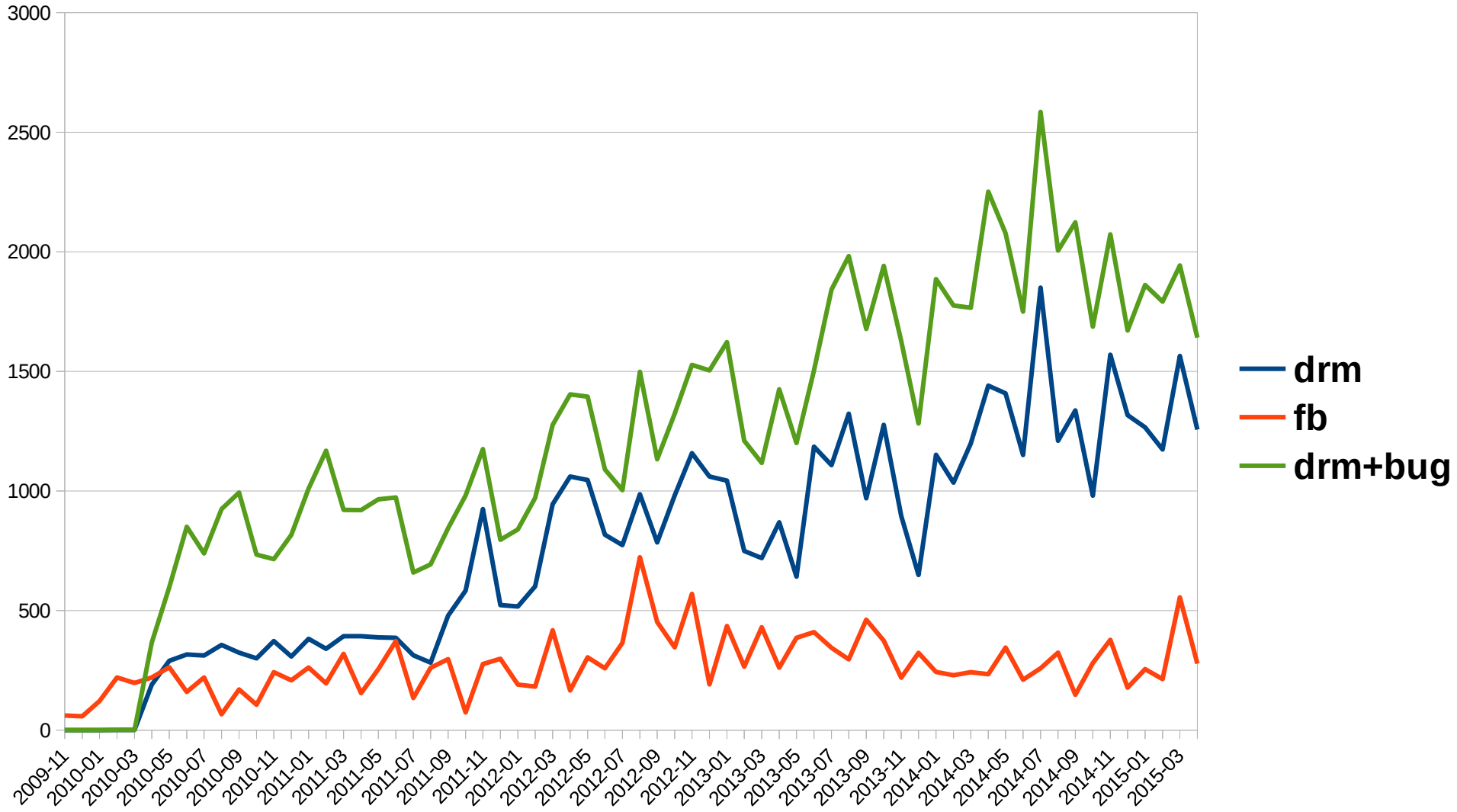


Source: http://valdodge.com/2010/07/

Activity – FBDEV

CambridgeGameCreators

CAMPUS LONDON

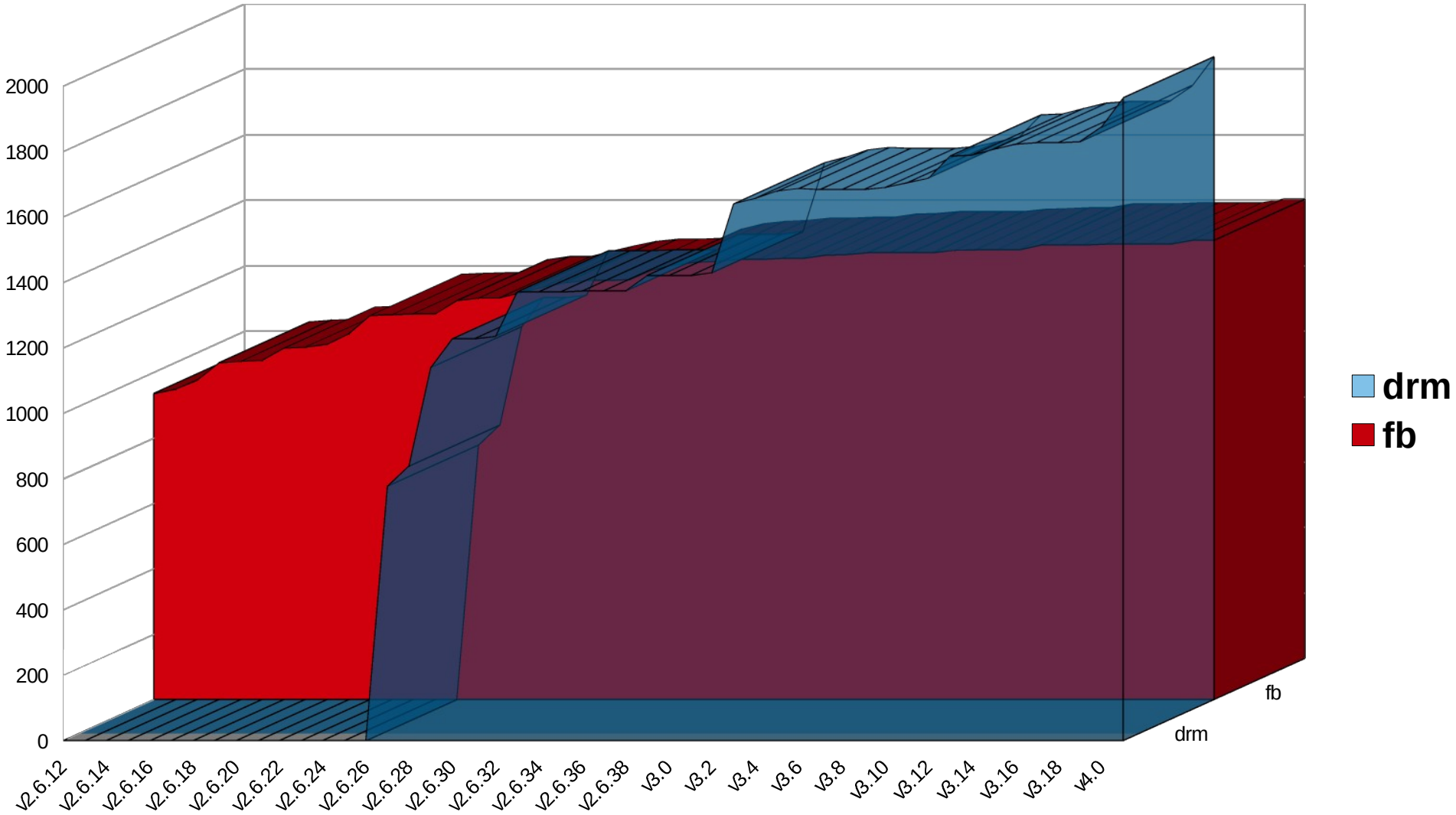


Mailing List Traffic

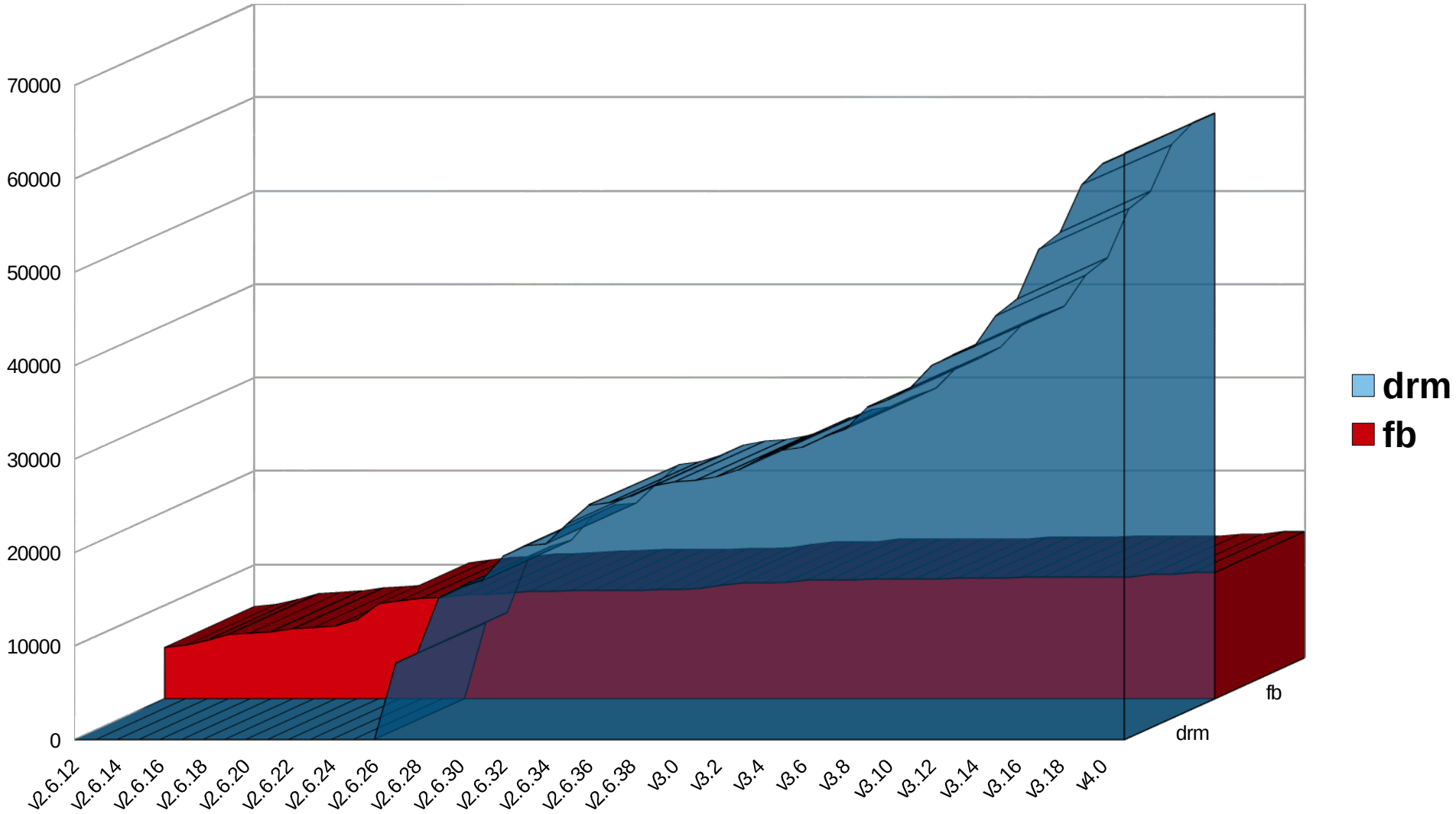
IDEAS ON BOARD

CambridgeGameCreators

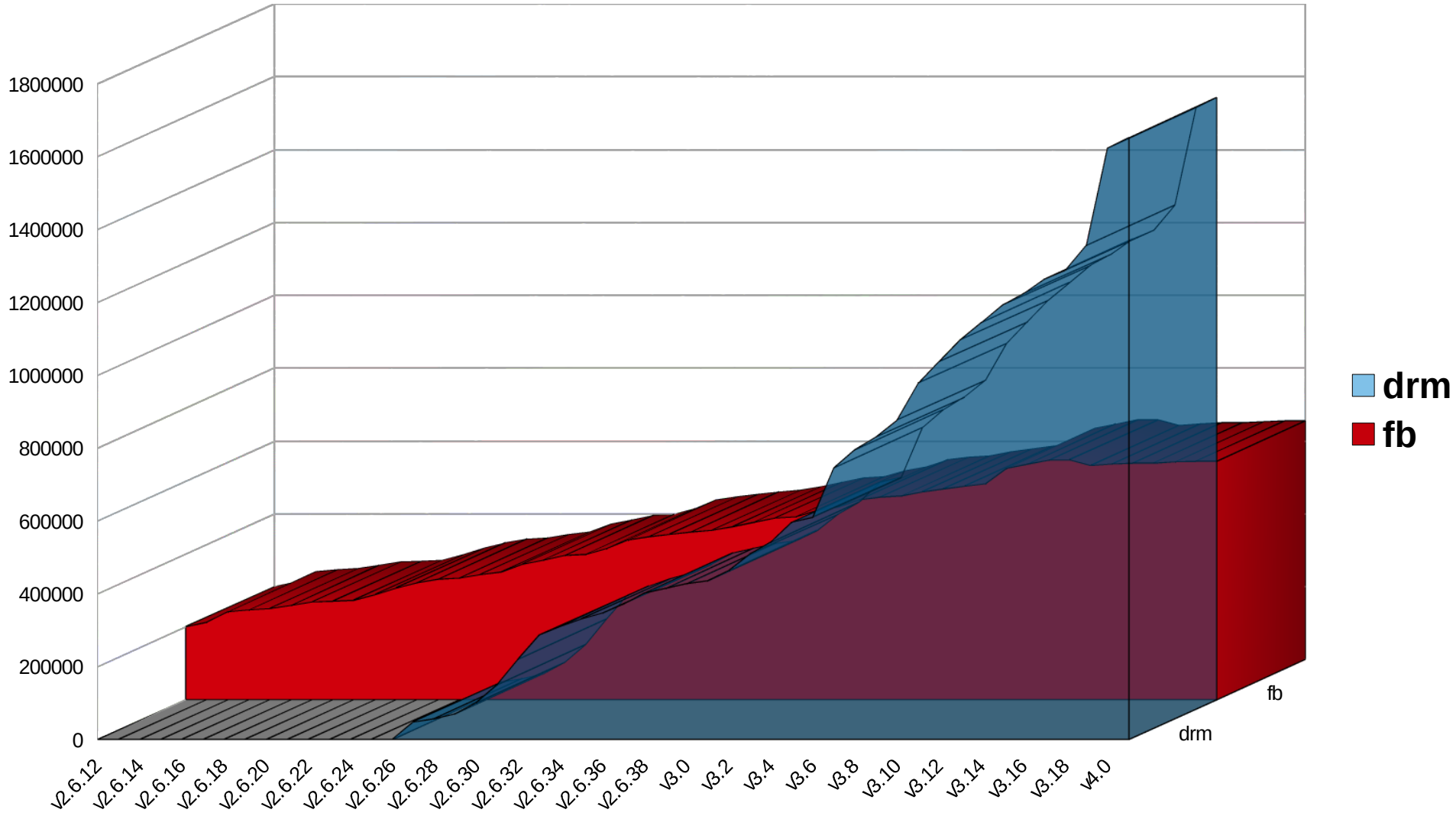
CAMPUS LONDON



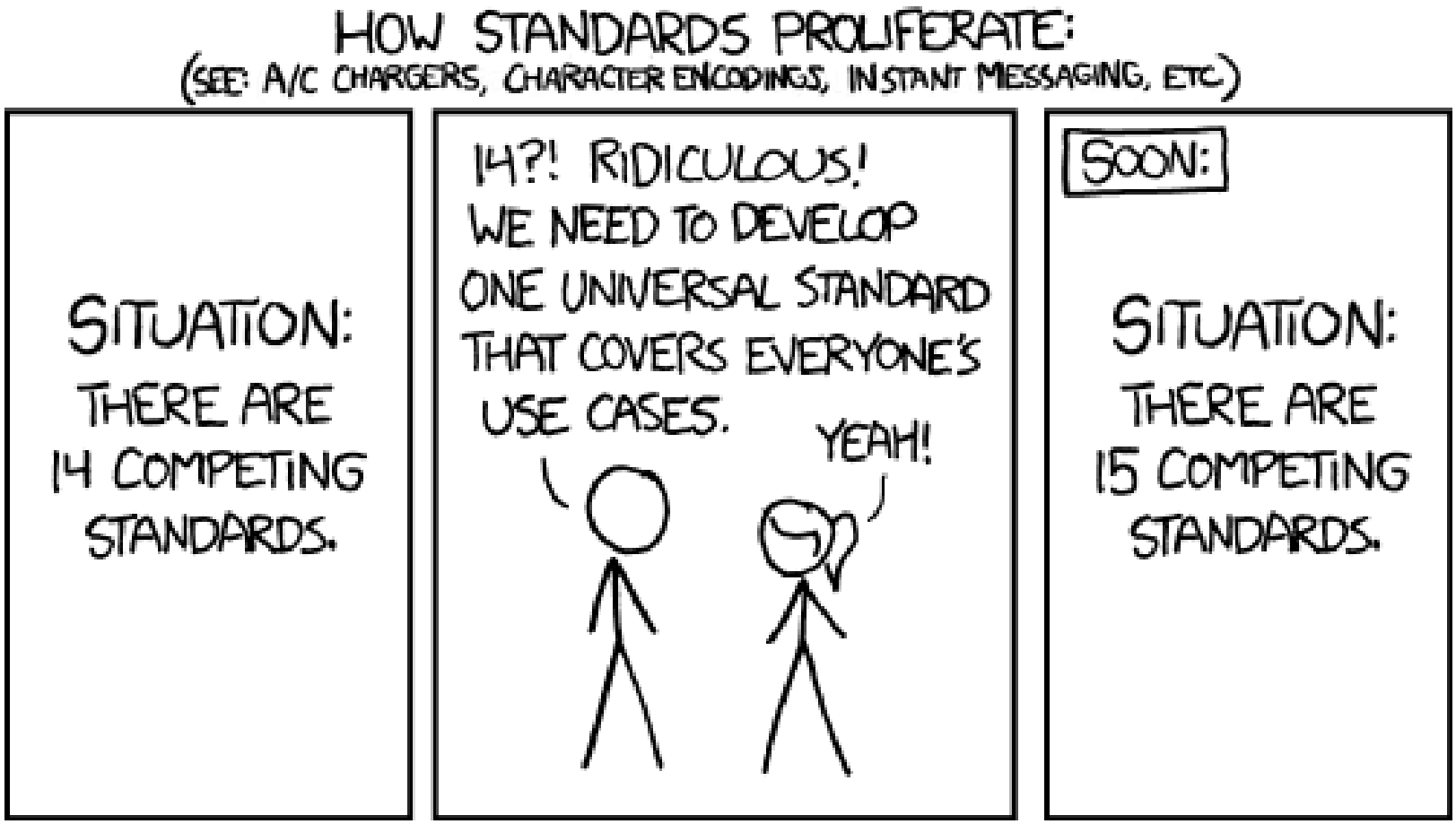
Cumulative Changes - API



Cumulative Changes - Core



Cumulative Changes - Drivers



Source: http://xkcd.com/927/

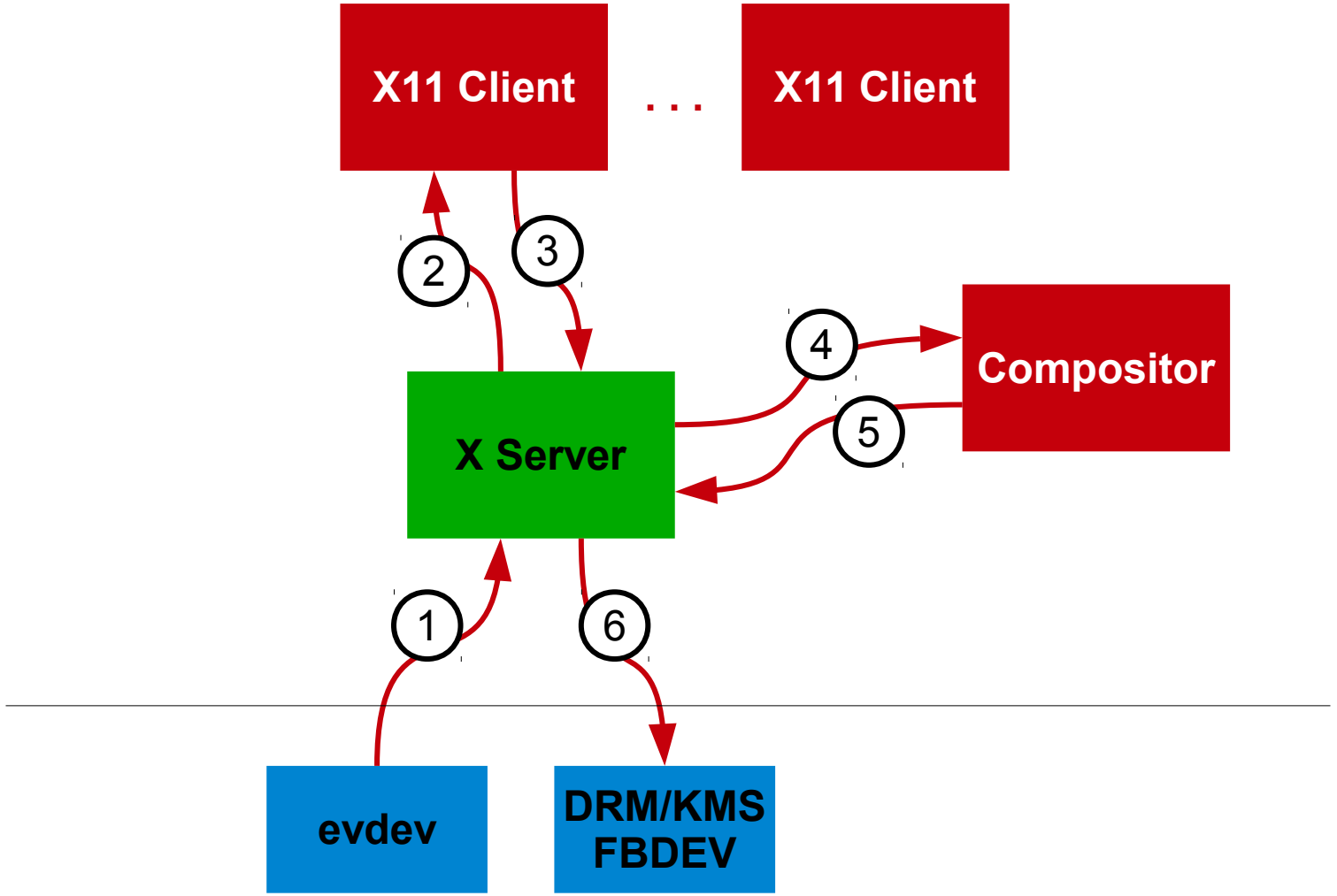
Use Cases

Use Cases - FBDEV

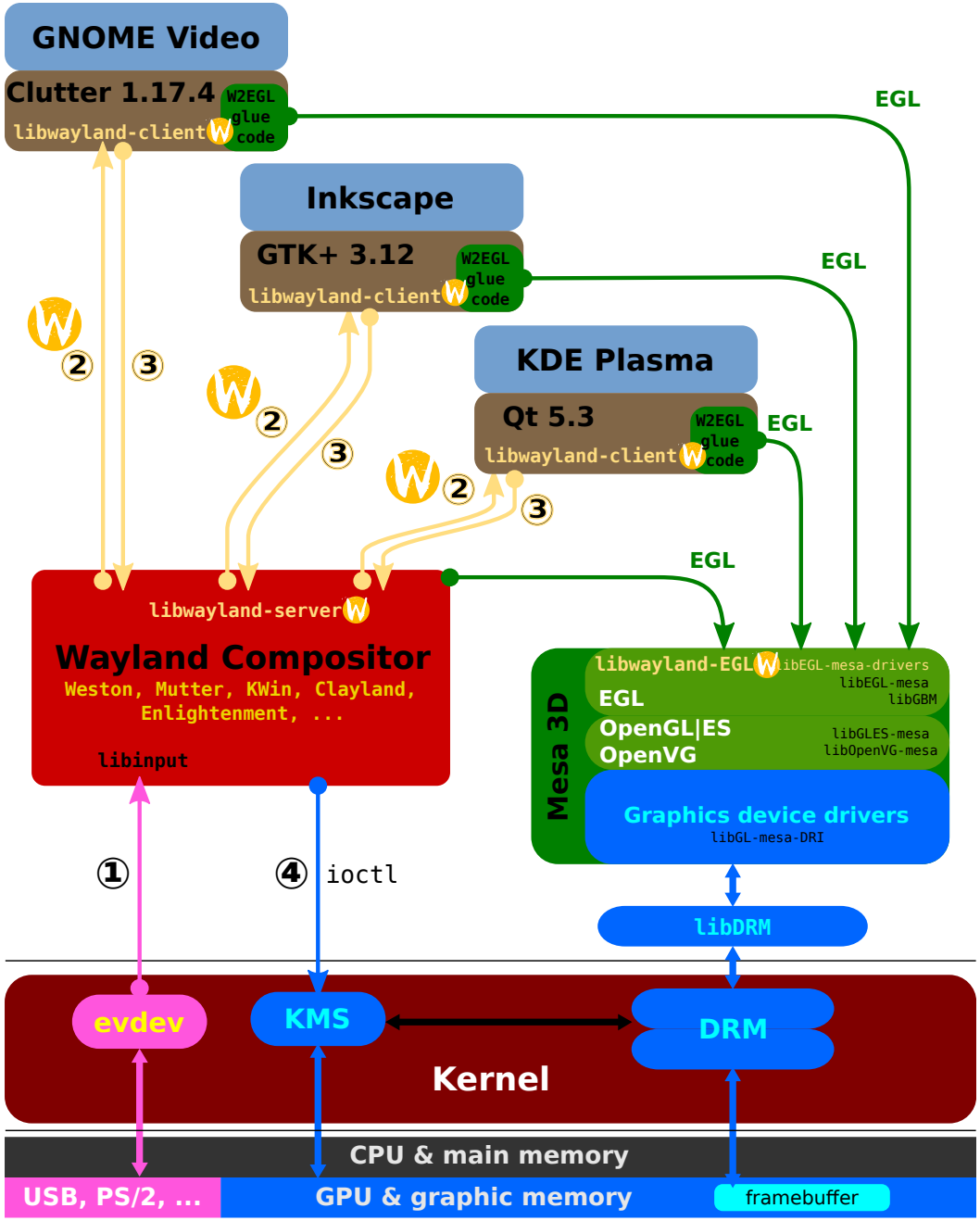
(that's it...)

Everything

Beyond X11



X11 Composition

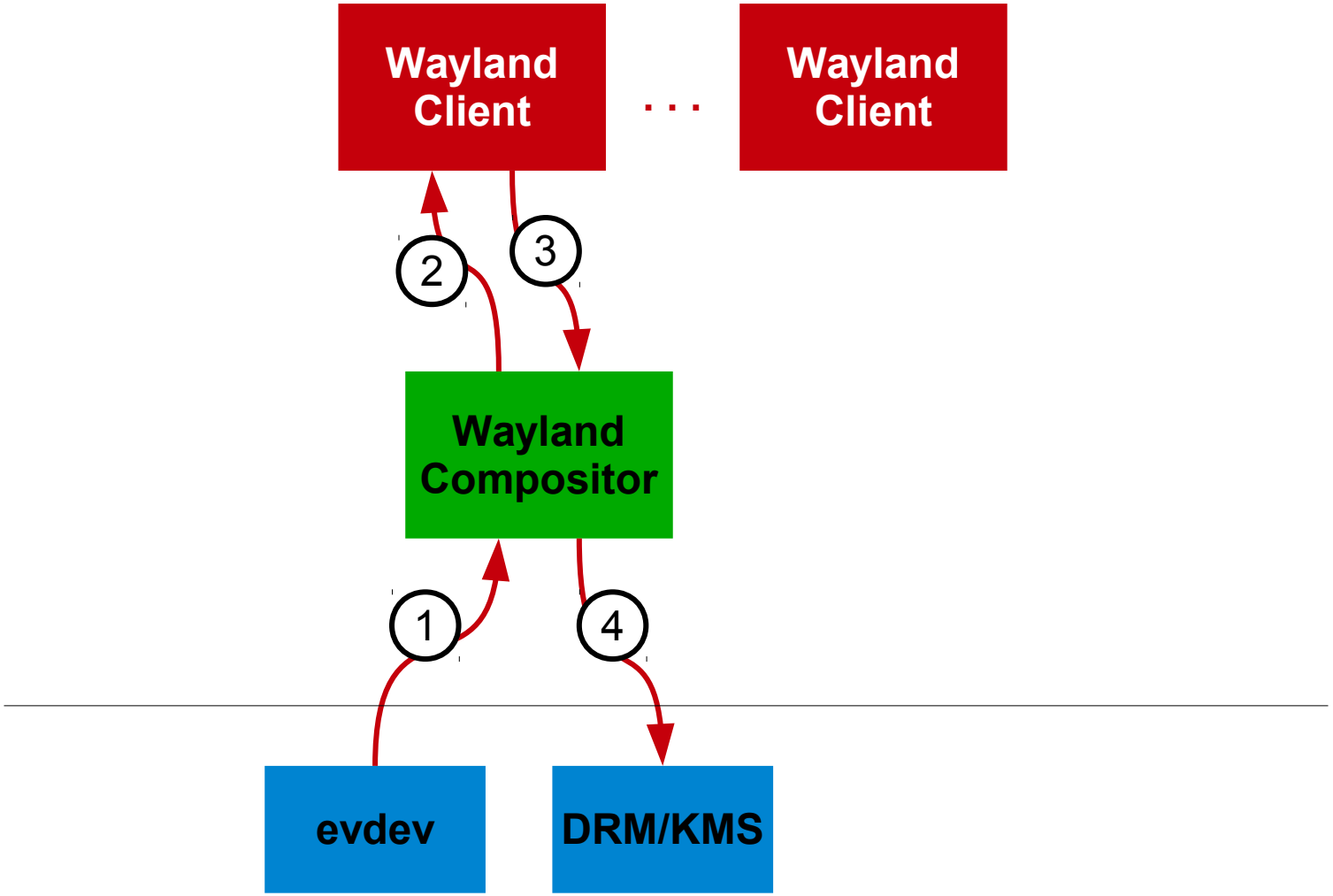


Wayland Graphics Stack

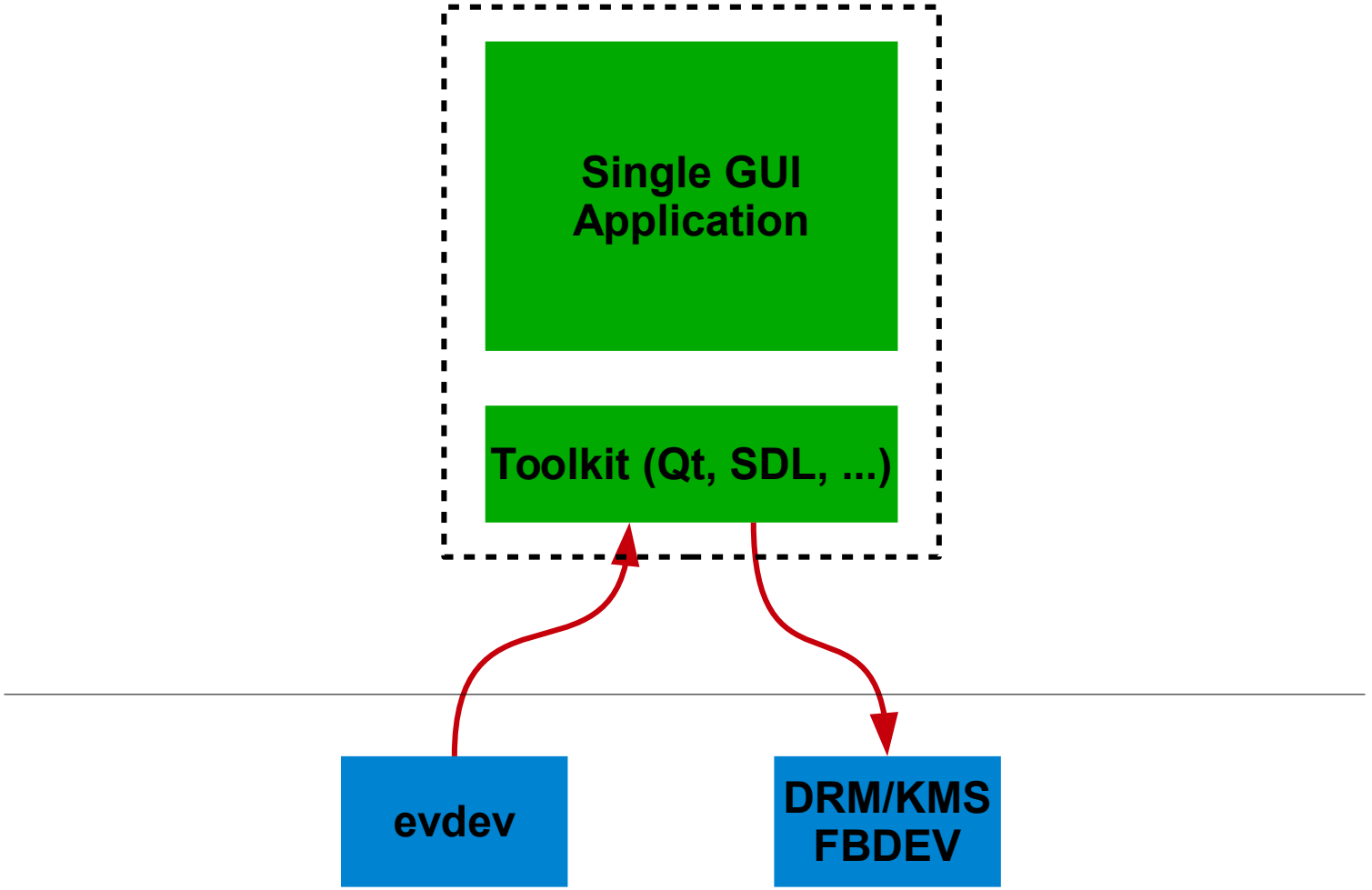
IDEAS ON BOARD

CambridgeGameCreators

CAMPUS LONDON

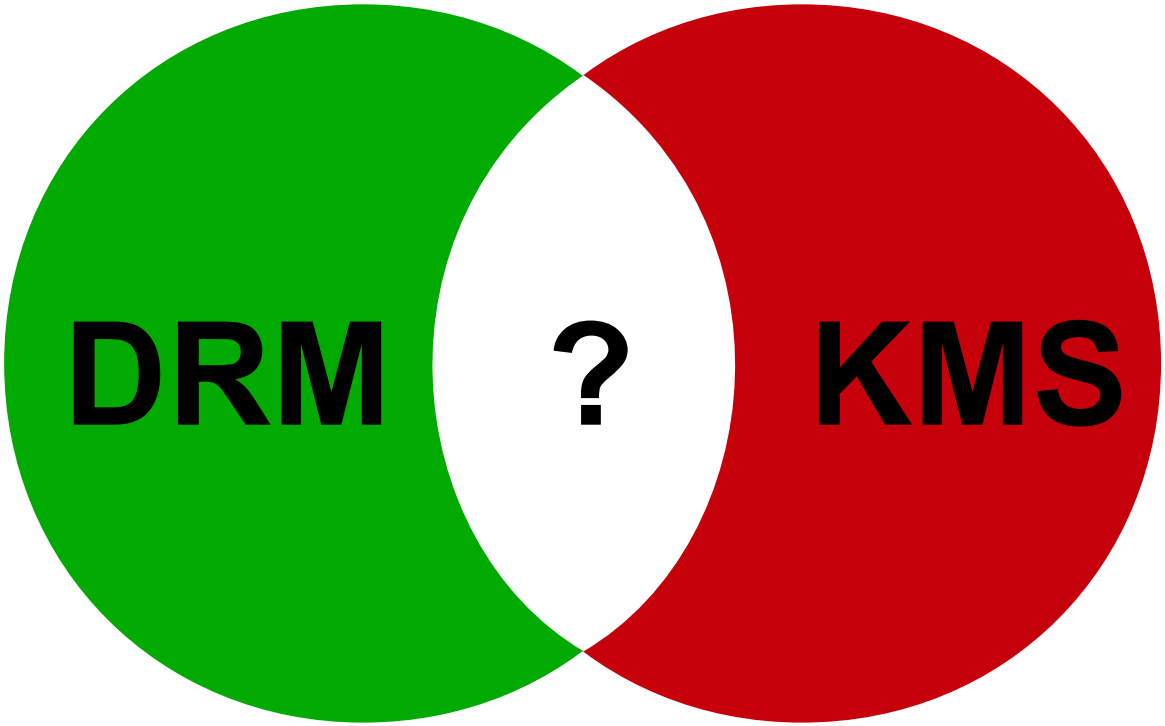


Wayland Composition



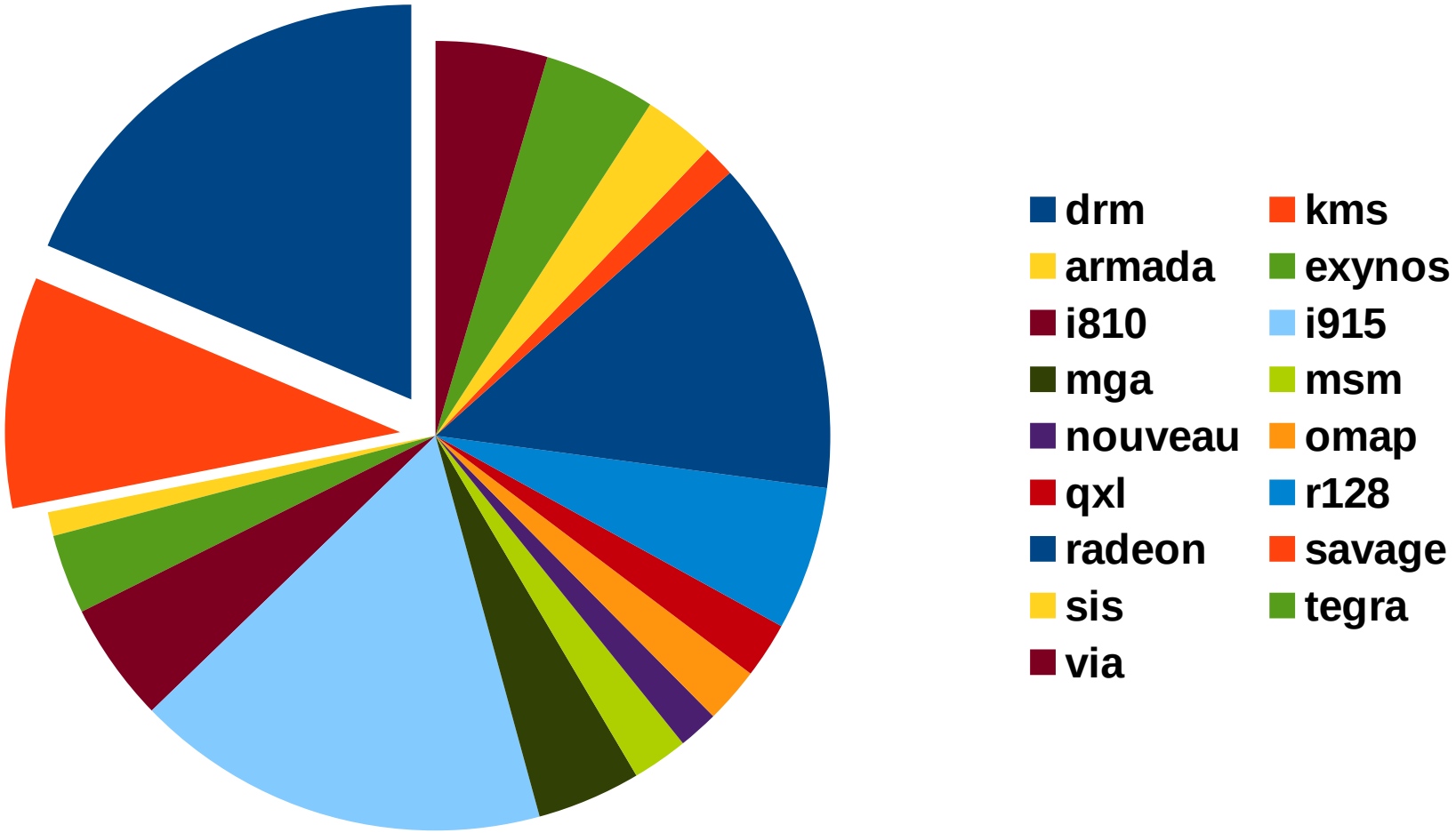
Custom Application

The DRM and KMS APIs

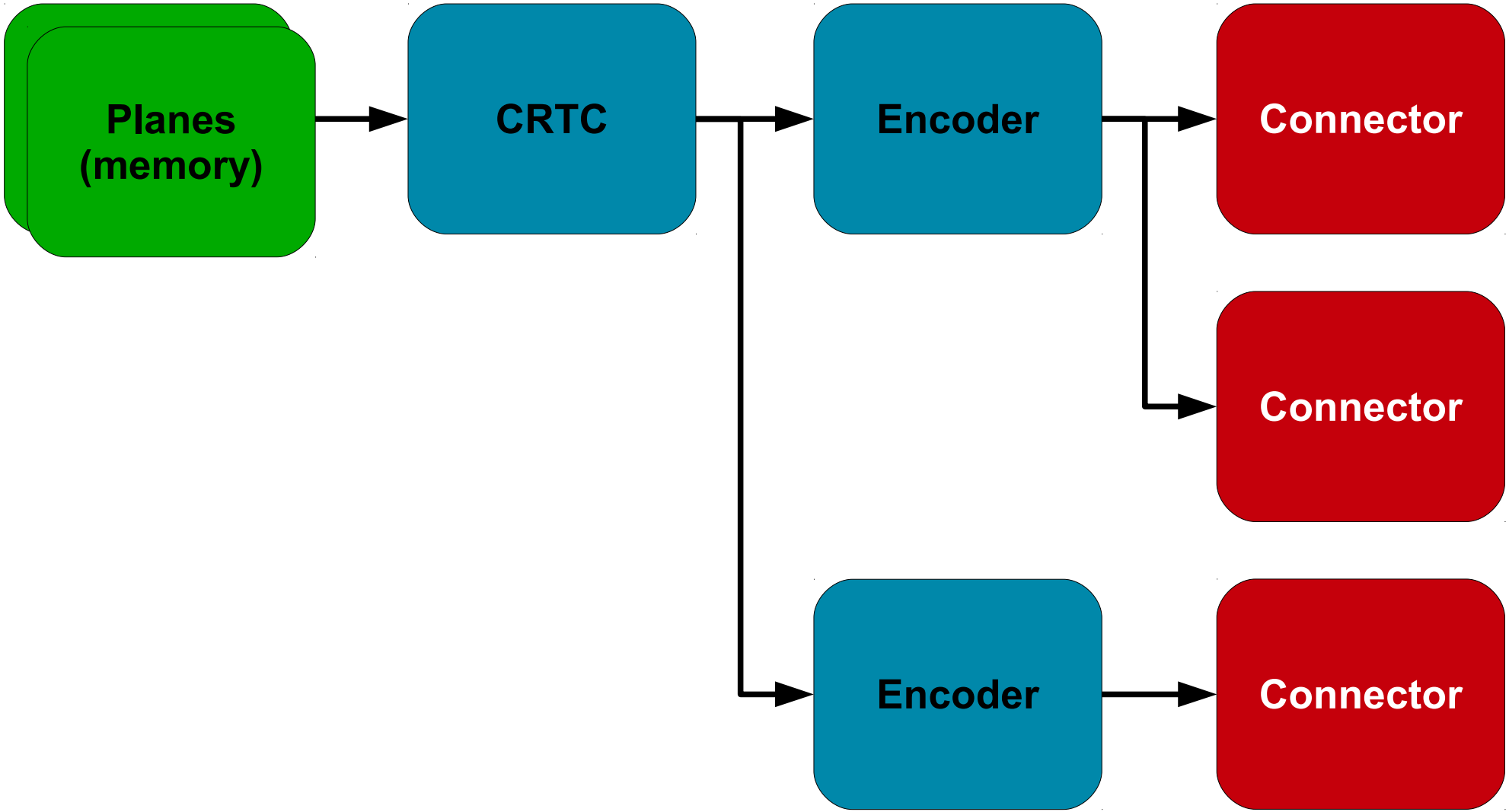

DRM
?
KMS

- Memory Management
- Command Stream
- Vertical Blanking
- Version, Authentication, Master, …

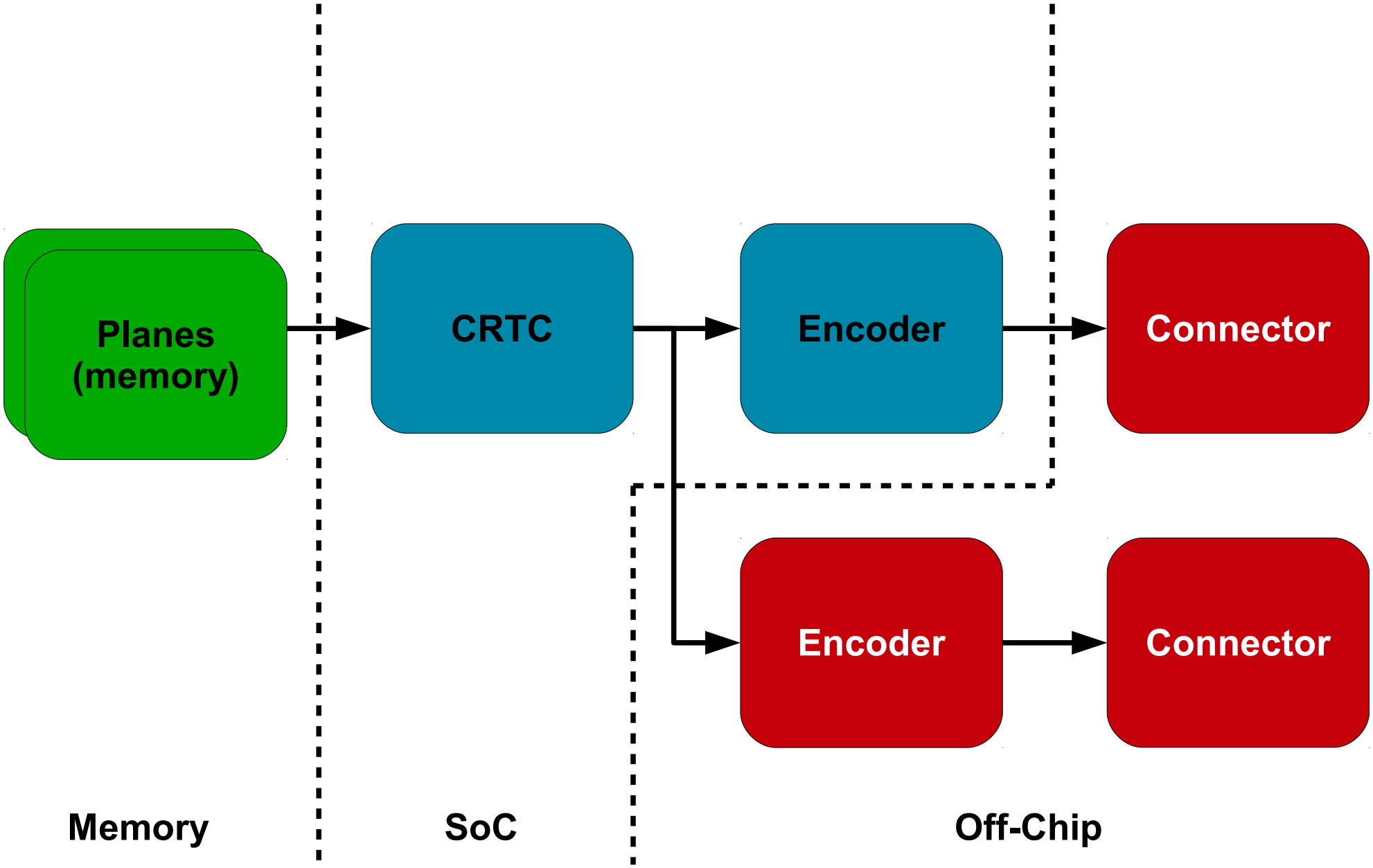
- Device Model
- Frame Buffer
- Modes
- Page Flip
- Planes
- Cursor, Gamma, ...



DRM/KMS API



Device Model



Device Model - SoC

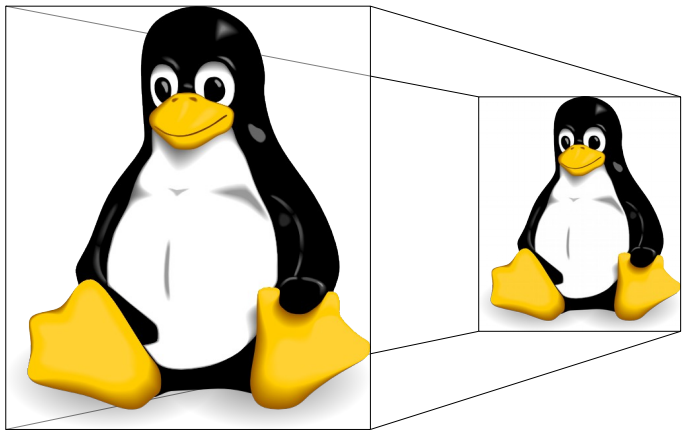
Frame Buffer

KMS – Scanout

Root Plane

Composition

Overlay Plane(s)

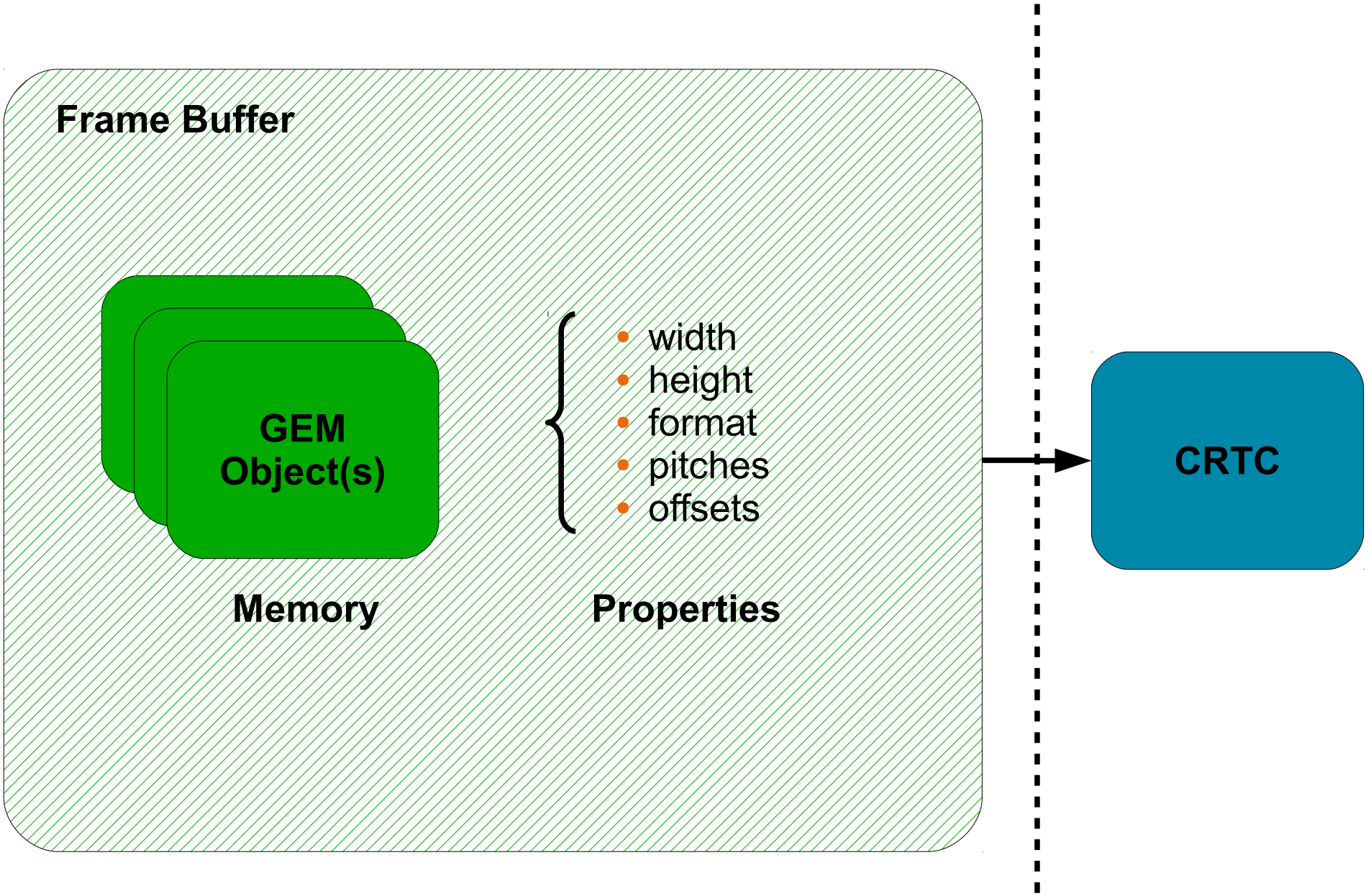


KMS – Composition

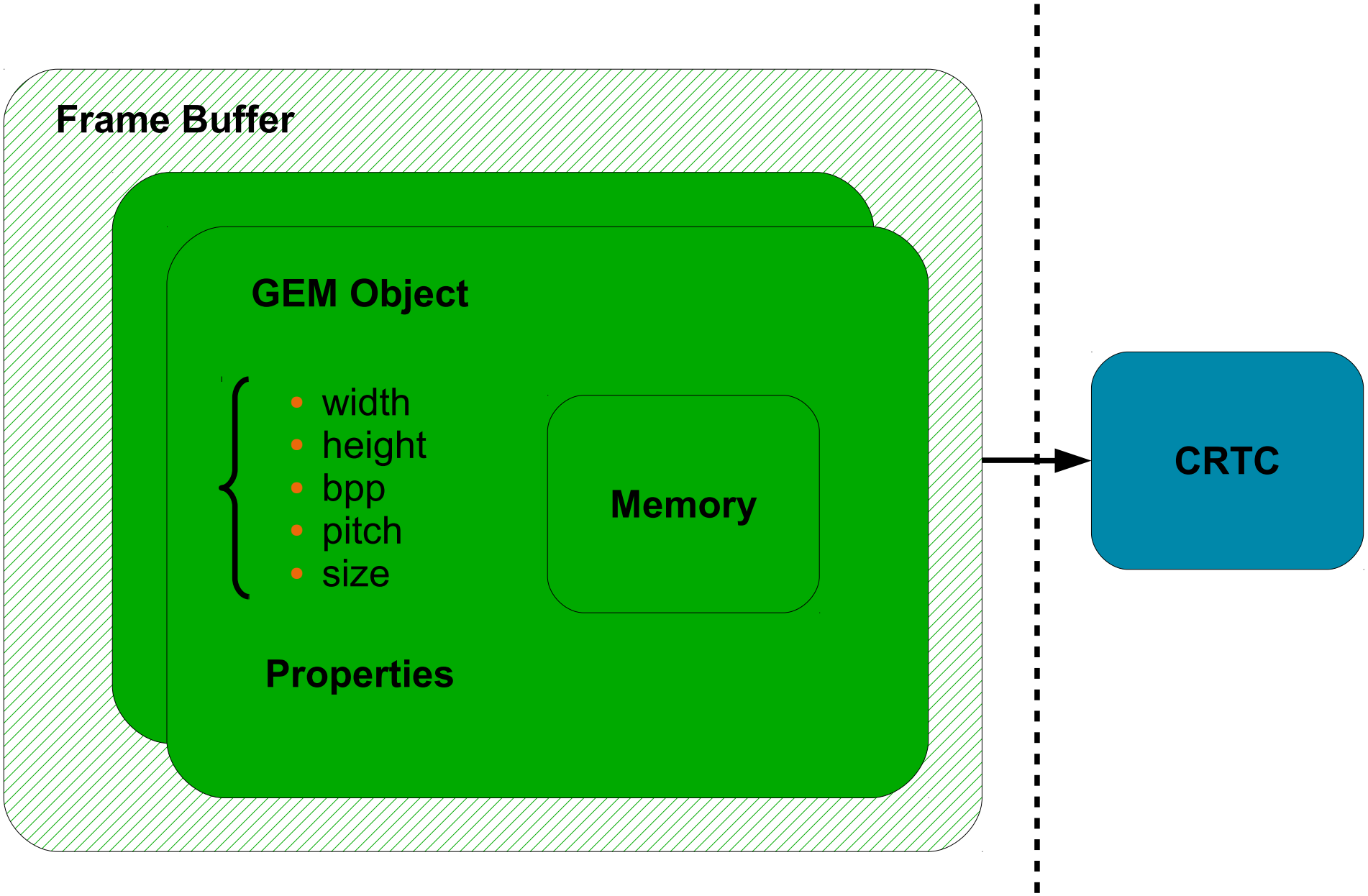
IDEAS ON BOARD

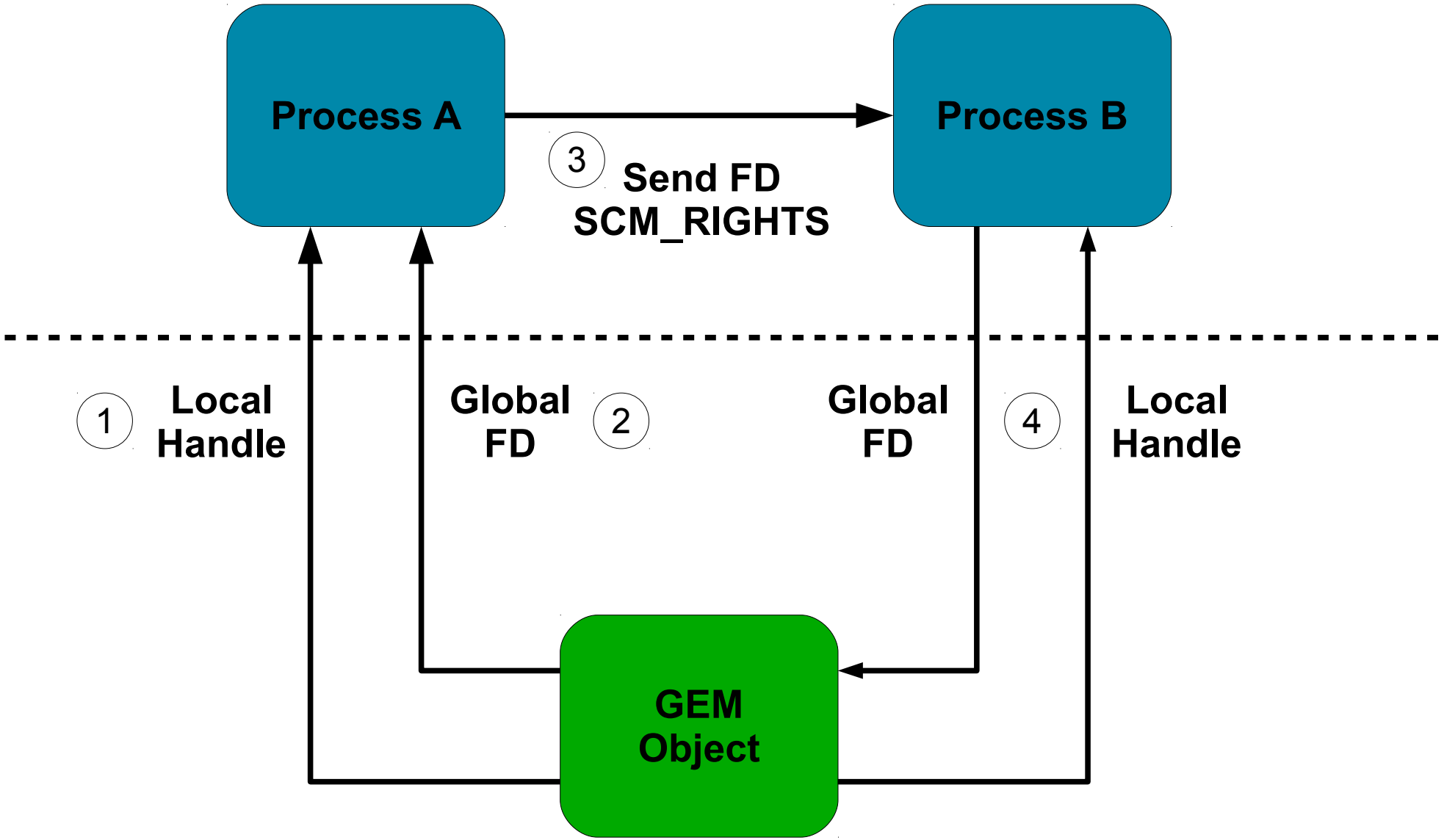
CambridgeGameCreators

CAMPUS LONDON

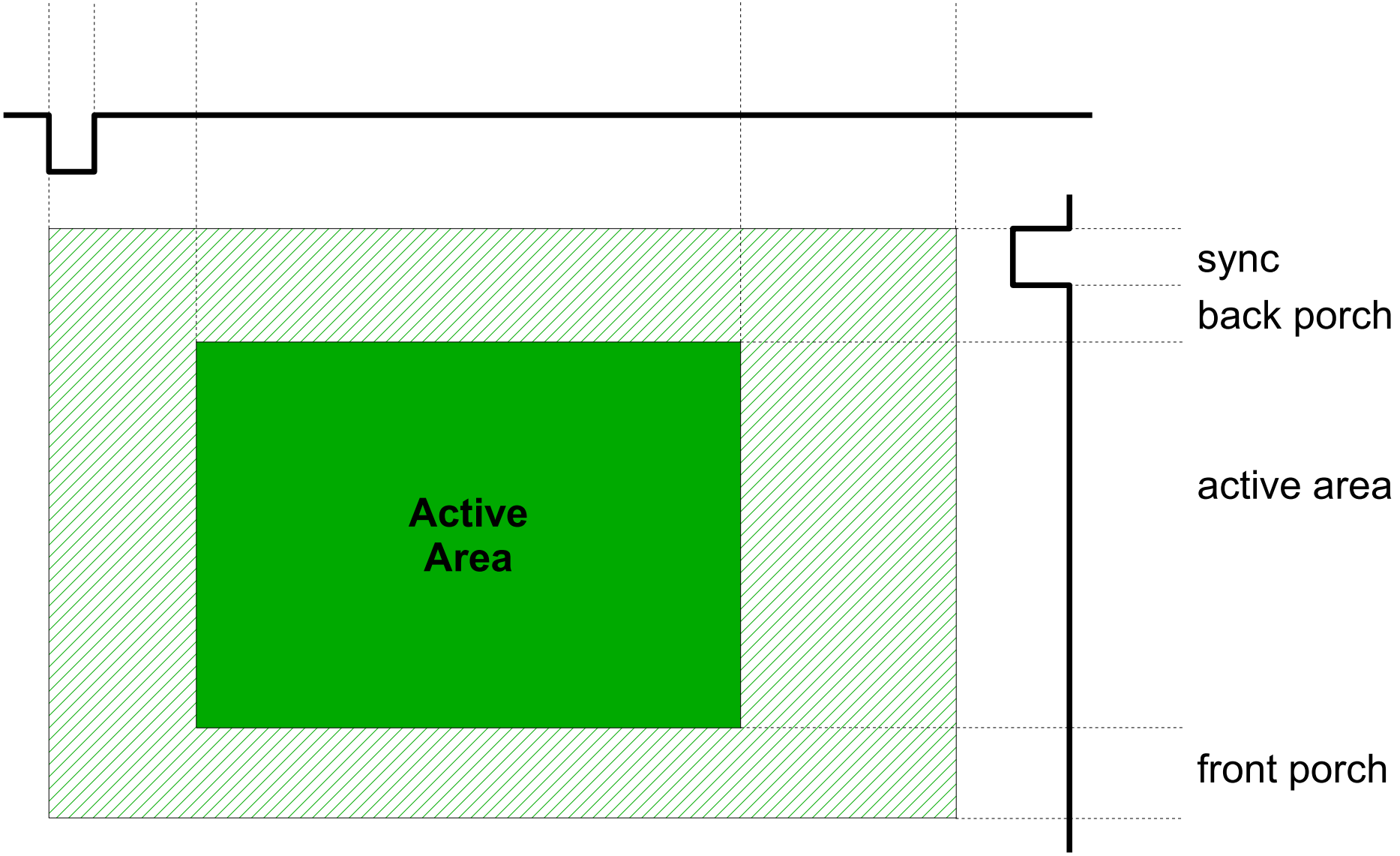


KMS – Frame Buffer

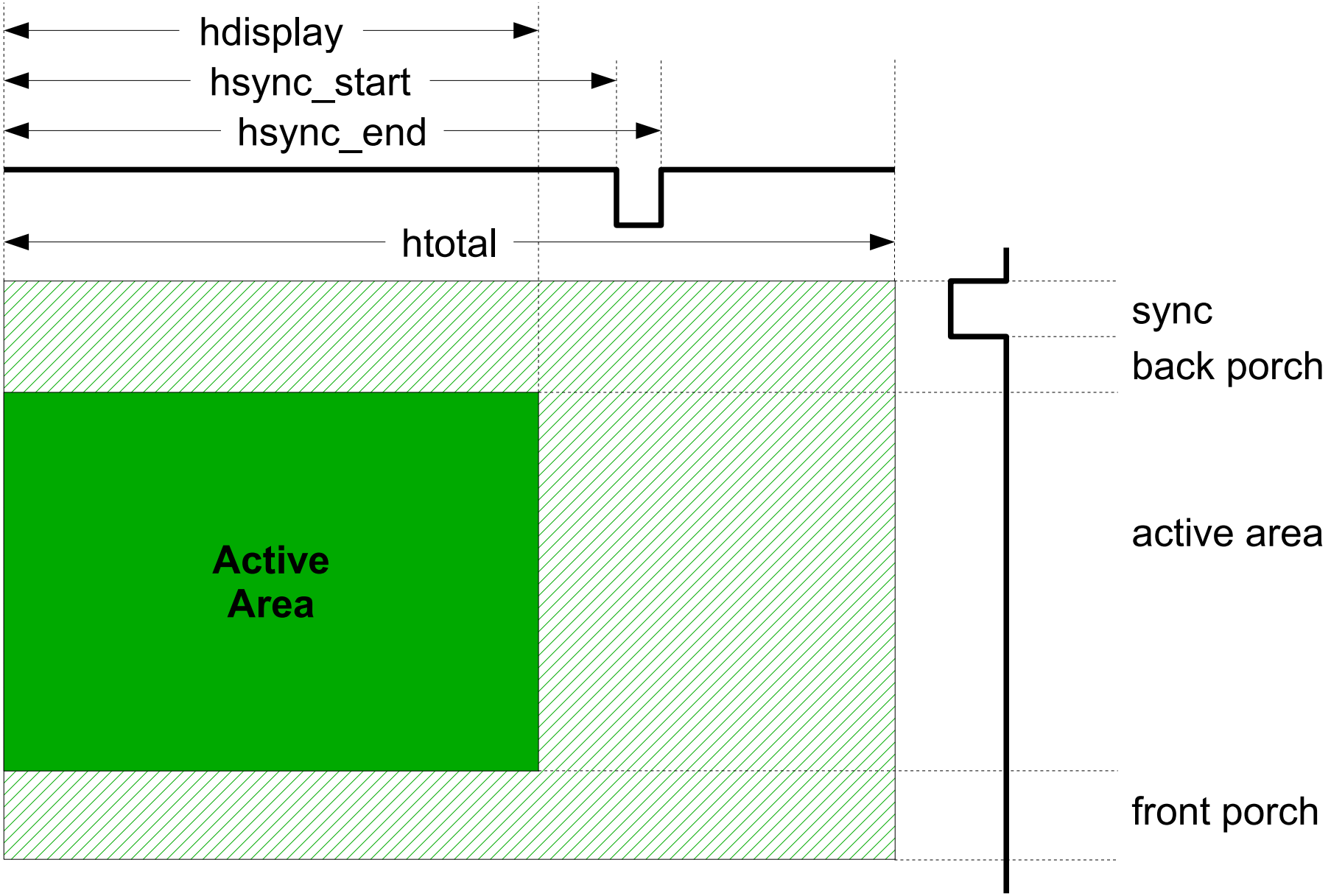

Frame Buffer
GEM Object
width
height
bpp
pitch
size
Memory
Properties
CRTC



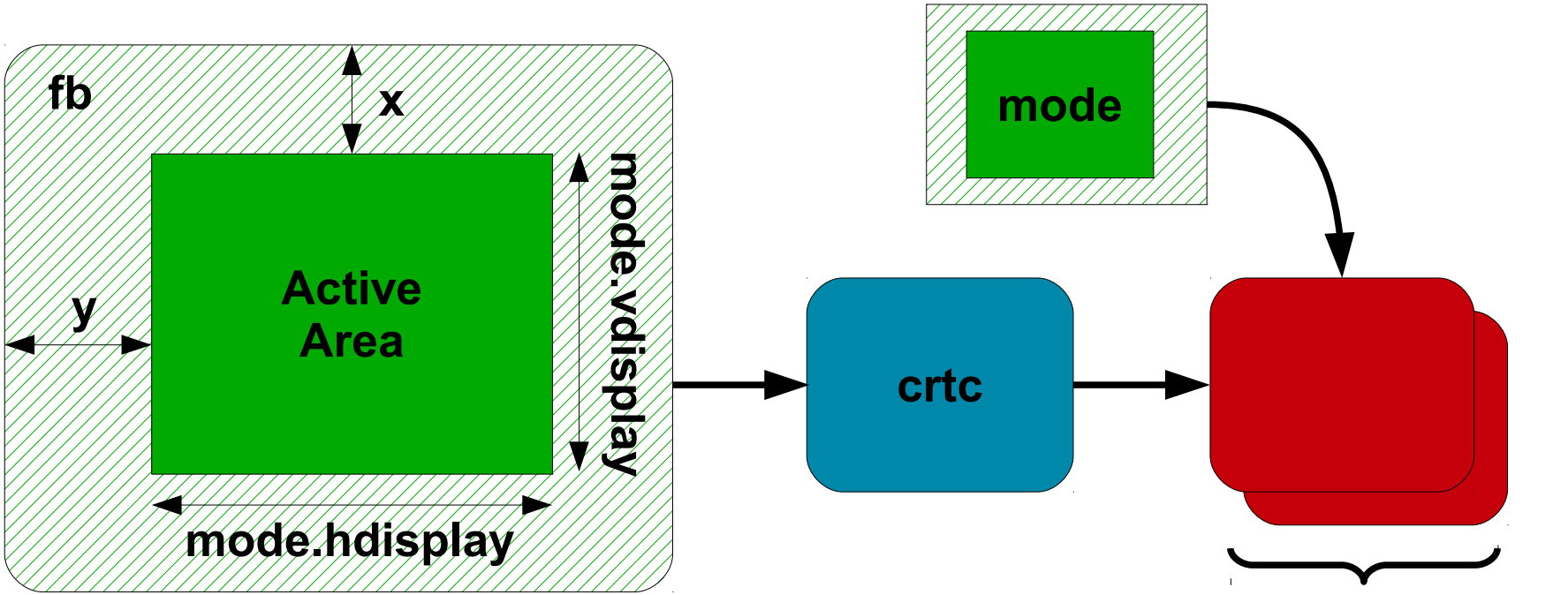
DRM – Handles



KMS – Modes (1/2)



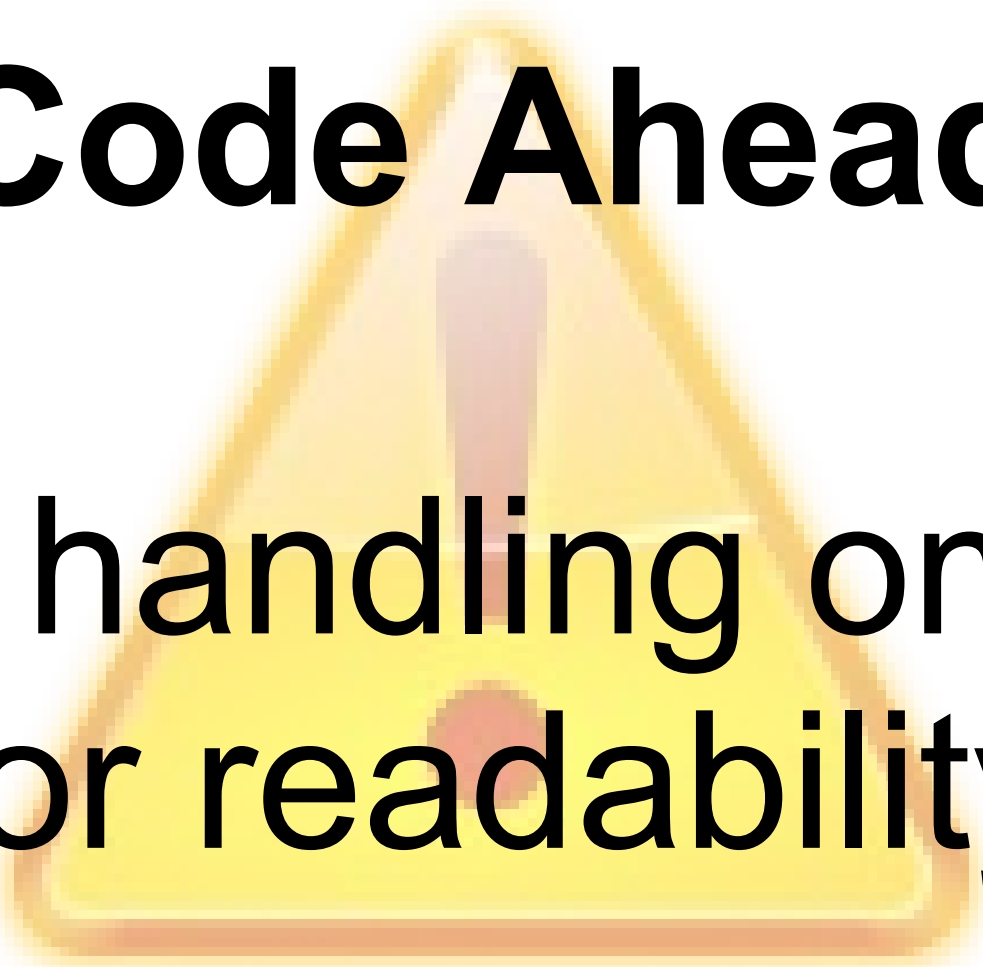
KMS – Modes (2/2)




```
struct drm_mode_set {
    struct drm_framebuffer *fb;
    struct drm_crtc *crtc;
    struct drm_display_mode *mode;
    uint32_t x;
    uint32_t y;
    struct drm_connector **connectors;
    size_t num_connectors;
};
```

KMS Programming Example

Code Ahead



Error handling omitted for readability

```c
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <string.h>

int main(int argc, char **argv)
{
        return 0;
}
```

Skeleton

```
/*
 * Open
 *
 * #include <xf86drm.h>
 *
 * int drmOpen(const char *name, const char *busid);
 */
int fd;

fd = drmOpen("imx-drm", NULL);
```

Open

```
/*
 * Get resources
 *
 * #include <xf86drmMode.h>
 *
 * drmModeResPtr drmModeGetResources(int fd);
 * void drmModeFreeResources(drmModeResPtr ptr);
 */
drmModeResPtr resources;
uint32_t crtc_id;
uint32_t connector_id;

resources = drmModeGetResources(fd);

crtc_id = resources->crtcs[0];
connector_id = resources->connectors[0];
```


Get Resources

```
/*
 * Get modes
 *
 * #include <xf86drmMode.h>
 *
 * drmModeConnectorPtr drmModeGetConnector(int fd,
 *             uint32_t connectorId);
 * void drmModeFreeConnector(drmModeConnectorPtr ptr);
 */
drmModeConnectorPtr connector;
drmModeModeInfo mode;
uint32_t width;
uint32_t height;

connector = drmModeGetConnector(fd, connector_id);

mode = connector->modes[0];
width = mode.hdisplay;
height = mode.vdisplay;
```

```
/*
 * Create buffer
 *
 * #include <drm.h>
 * #include <drm_mode.h>
 *
 * #define DRM_IOCTL_MODE_CREATE_DUMB ...
 *
 * int drmIoctl(int fd, unsigned long request, void *arg);
 */
struct drm_mode_create_dumb bo_create;

memset(&bo_create, 0, sizeof(bo_create));
bo_create.bpp = 32;
bo_create.width = width;
bo_create.height = height;

drmIoctl(fd, DRM_IOCTL_MODE_CREATE_DUMB, &bo_create);
```

```
/*
 * Fill buffer
 *
 * #include <drm.h>
 * #include <drm_mode.h>
 * #include <xf86drm.h>
 *
 * #define DRM_IOCTL_MODE_MAP_DUMB  ...
 *
 * int drmIoctl(int fd, unsigned long request, void *arg);
 */
struct drm_mode_map_dumb bo_map;

memset(&bo_map, 0, sizeof(bo_map));
bo_map.handle = bo_create.handle;

drmIoctl(fd, DRM_IOCTL_MODE_MAP_DUMB, &bo_map);
```


Fill Buffer (1/2)

```
/*
 * Fill buffer
 *
 * #include <sys/mman.h>
 *
 * void *mmap(void *addr, size_t length, int prot,
 *            int flags, int fd, off_t offset)
 * int munmap(void *addr, size_t length);
 */
void *mem;

mem = mmap(0, bo_create.size, PROT_READ | PROT_WRITE,
           MAP_SHARED, fd, bo_map.offset);
fill_pattern(DRM_FORMAT_ARGB8888, mem, width, height,
             bo_create.pitch);
munmap(mem, bo_create.size);
```

```
/*
 * Create frame buffer
 *
 * #include <drm_fourcc.h>
 * #include <xf86drmMode.h>
 *
 * int drmModeAddFB2(int fd, uint32_t width,
 *       uint32_t height, uint32_t pixel_format,
 *       uint32_t bo_handles[4], uint32_t pitches[4],
 *       uint32_t offsets[4], uint32_t *buf_id,
 *       uint32_t flags);
 */
uint32_t handles[4] = { bo_create.handle, };
uint32_t pitches[4] = { bo_create.pitch, };
uint32_t offsets[4] = { 0, };
uint32_t fb_id;

drmModeAddFB2(fd, width, height, DRM_FORMAT_ARGB8888,
              handles, pitches, offsets, &fb_id, 0);
```

```
/*
 * Set the mode
 *
 * #include <xf86drmMode.h>
 *
 * int drmModeSetCrtc(int fd, uint32_t crtcId,
 *                    uint32_t bufferId, uint32_t x, uint32_t y,
 *                    uint32_t *connectors, int count,
 *                    drmModeModeInfoPtr mode);
 */
drmModeSetCrtc(fd, crtc_id, fb_id, 0, 0,
               &connector_id, 1, &mode);
```

```
/*
 * Get plane resources
 *
 * #include <xf86drmMode.h>
 *
 * drmModePlaneResPtr drmModeGetPlaneResources(int fd);
 * void drmModeFreePlaneResources(drmModePlaneResPtr ptr);
 */
drmModePlaneResPtr planes;
uint32_t plane_id;

planes = drmModeGetPlaneResources(fd);
plane_id = planes->planes[0];
```

Get Plane Resources

```
/*
 * Set the plane
 *
 * #include <xf86drmMode.h>
 *
 * int drmModeSetPlane(int fd,
 *         uint32_t plane_id, uint32_t crtc_id,
 *         uint32_t fb_id, uint32_t flags,
 *         uint32_t crtc_x, uint32_t crtc_y,
 *         uint32_t crtc_w, uint32_t crtc_h,
 *         uint32_t src_x, uint32_t src_y,
 *         uint32_t src_w, uint32_t src_h);
 */
drmModeSetPlane(fd, plane_id, crtc_id, fb_id, 0,
          width / 4, height / 4, width / 2, height / 2,
          0, 0, (width / 2) << 16, (height / 2) << 16);
```

```
/*
 * List plane properties
 *
 * #include <drm_mode.h>
 * #include <xf86drmMode.h>
 *
 * drmModeObjectPropertiesPtr drmModeObjectGetProperties(
 *       int fd, uint32_t object_id, uint32_t object_type);
 * void drmModeFreeObjectProperties(
 *       drmModeObjectPropertiesPtr ptr);
 */
drmModeObjectPropertiesPtr properties;

properties = drmModeObjectGetProperties(fd, plane_id,
                                        DRM_MODE_OBJECT_PLANE);
```


List Plane Properties

```
/*
 * Find alpha property
 *
 * #include <xf86drmMode.h>
 *
 * drmModePropertyPtr drmModeGetProperty(int fd,
 *                                       uint32_t propertyId);
 * void drmModeFreeProperty(drmModePropertyPtr ptr);
 */
drmModePropertyPtr property;
unsigned int i;

for (i = 0; i < properties->count_props; ++i) {
        property = drmModeGetProperty(fd,
                                      properties->props[i]);
        if (!strcmp(property->name, "alpha"))
                break;
}
```

```
/*
 * Turn global transparency on
 *
 * #include <xf86drmMode.h>
 *
 * int drmModeObjectSetProperty(int fd, uint32_t object_id,
 *              uint32_t object_type, uint32_t property_id,
 *              uint64_t value);
 */
drmModeObjectSetProperty(fd, plane_id,
                         DRM_MODE_OBJECT_PLANE,
                         property->prop_id, 128);
```

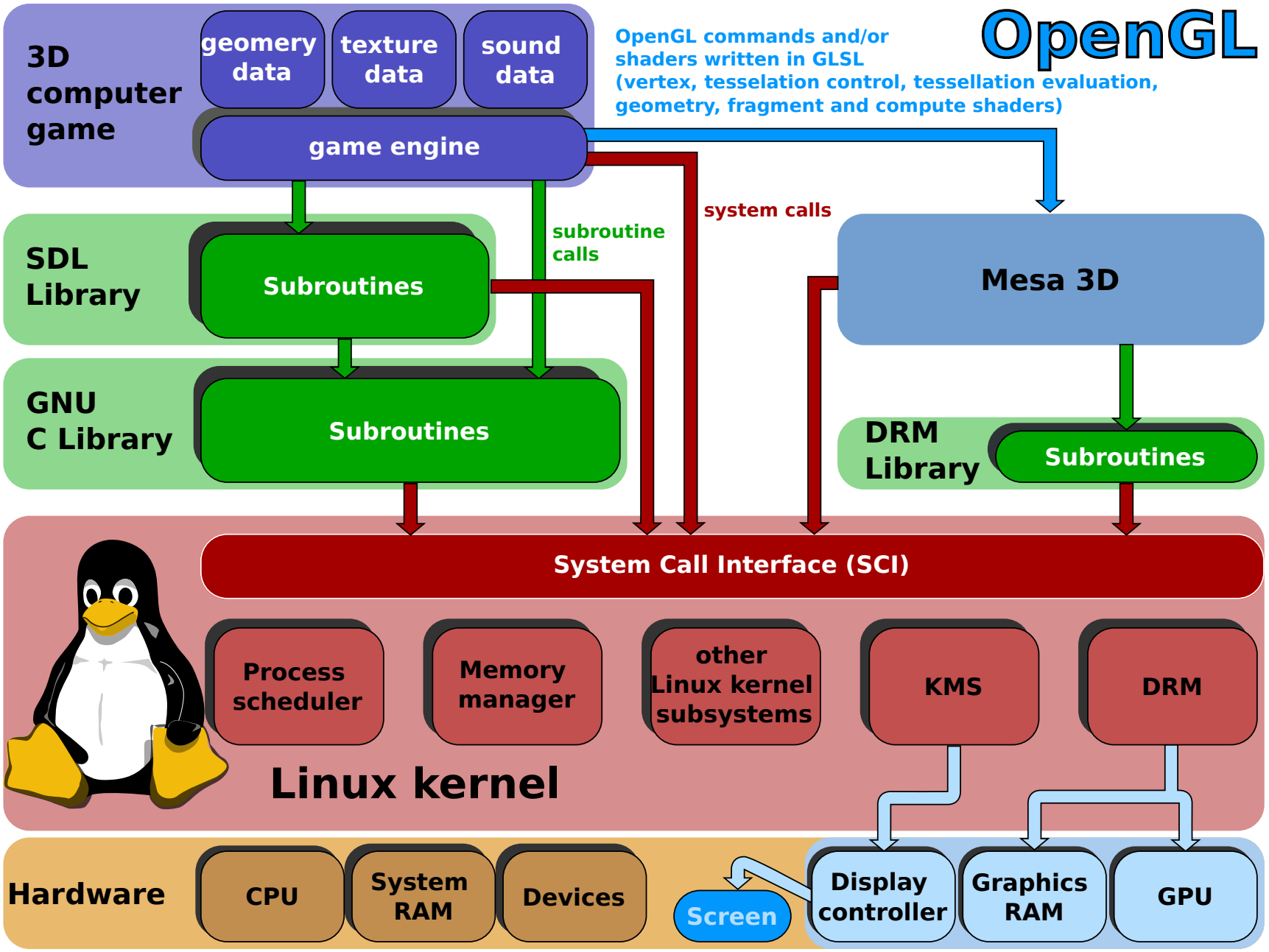
- Atomic Update
- Page Flip
- Cursor
- Events
- ...

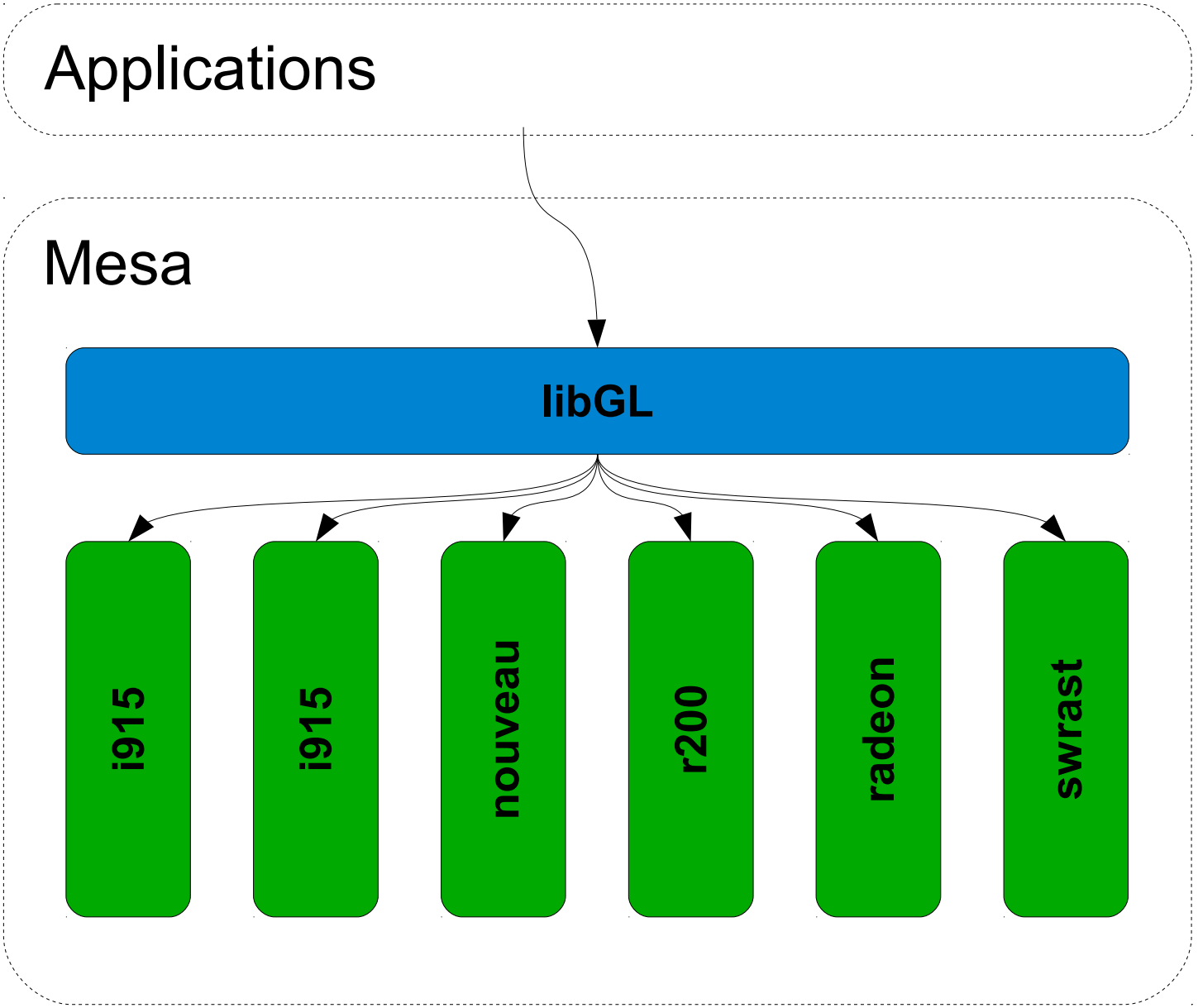
And Much More...

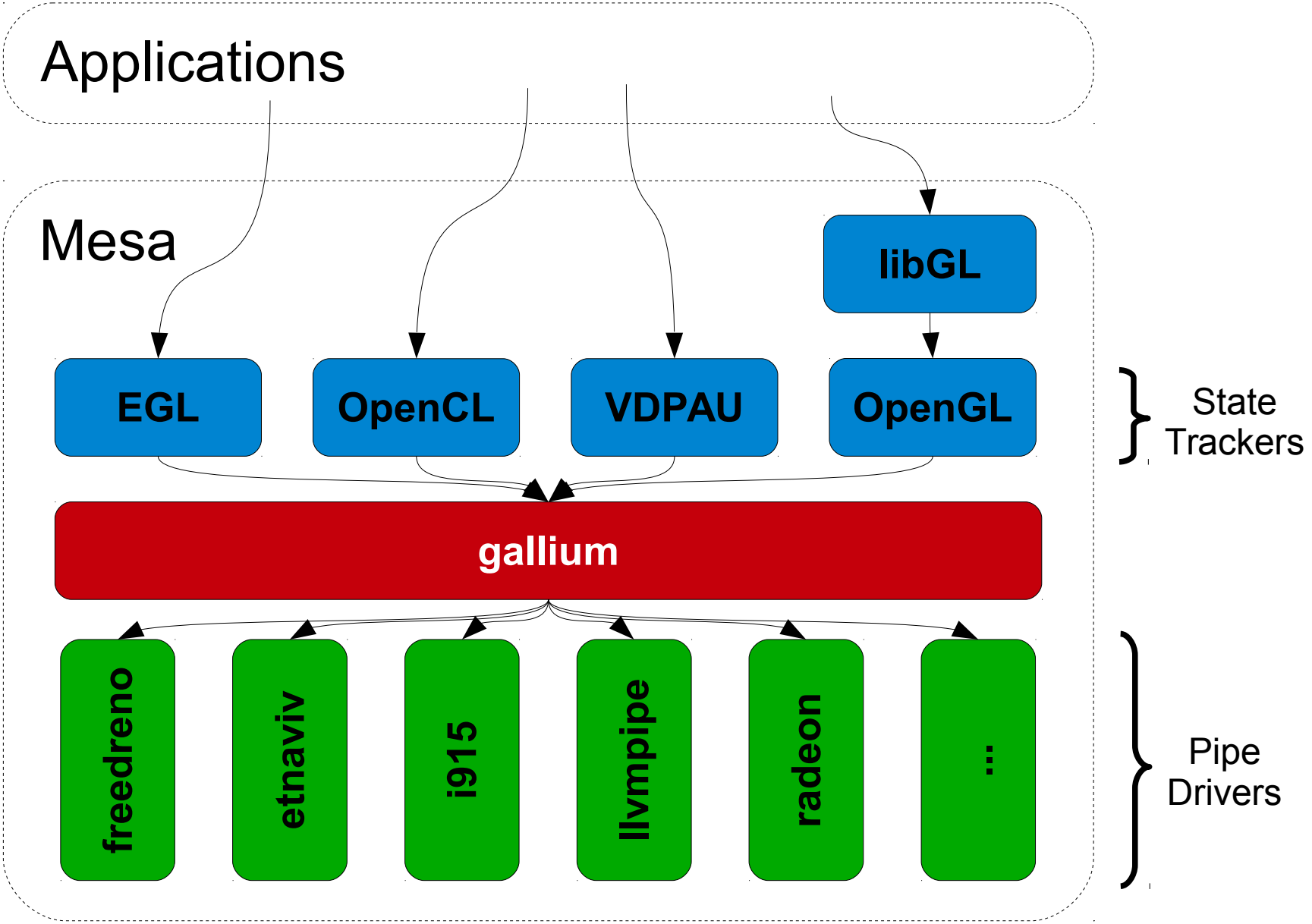
CambridgeGameCreators

CAMPUS
LONDON

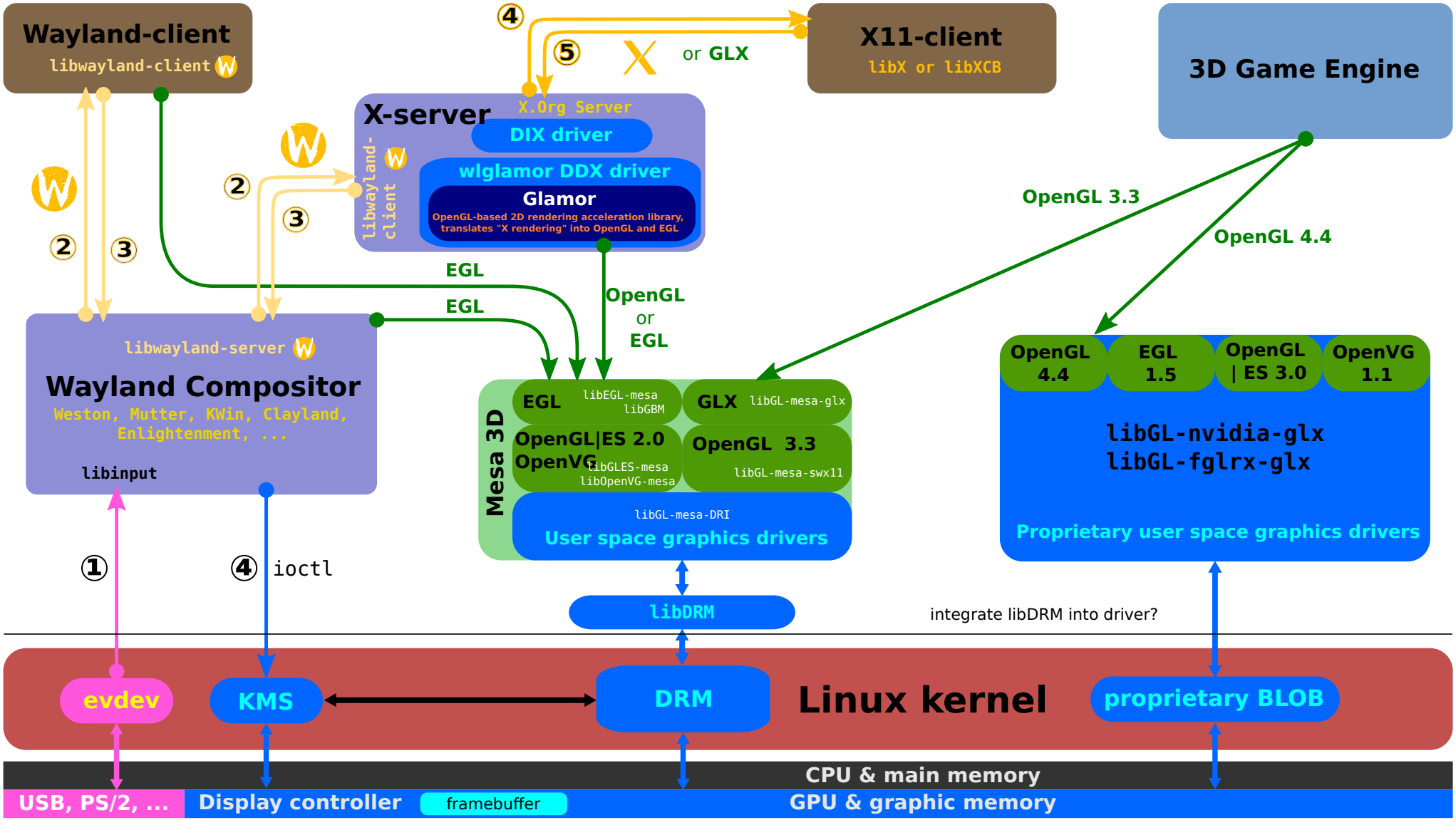
3D Acceleration


3D computer game
geomery data
texture data
sound data
game engine
OpenGL commands and/or shaders written in GLSL (vertex, tesselation control, tessellation evaluation, geometry, fragment and compute shaders)
OpenGL
system calls
subroutine calls
SDL Library
Subroutines
Mesa 3D
GNU C Library
Subroutines
DRM Library
Subroutines
System Call Interface (SCI)
Process scheduler
Memory manager
other Linux kernel subsystems
KMS
DRM
Linux kernel
Hardware
CPU
System RAM
Devices
Screen
Display controller
Graphics RAM
GPU


Applications
Mesa
libGL
i915
i915
nouveau
r200
radeon
swrast



Gallium 3D



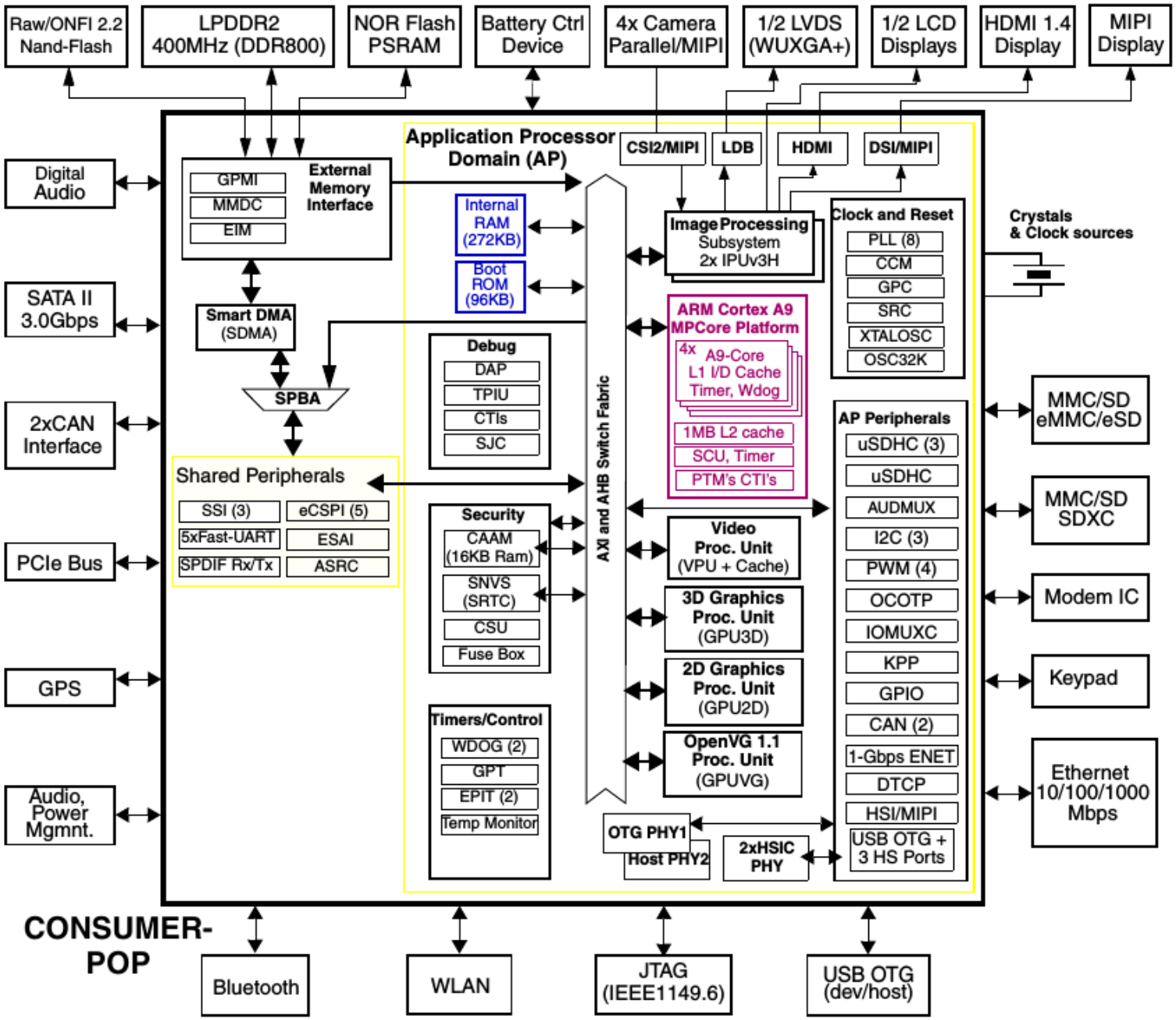
Free vs. Proprietary

IDEAS ON BOARD

CambridgeGameCreators

CAMPUS LONDON

	Free	Proprietary
Performance	+(+)	++(+)
Features	+(+)	++
Security	+	-
Kernel	open	open/ closed
Evolutions	yes	vendor lock-in

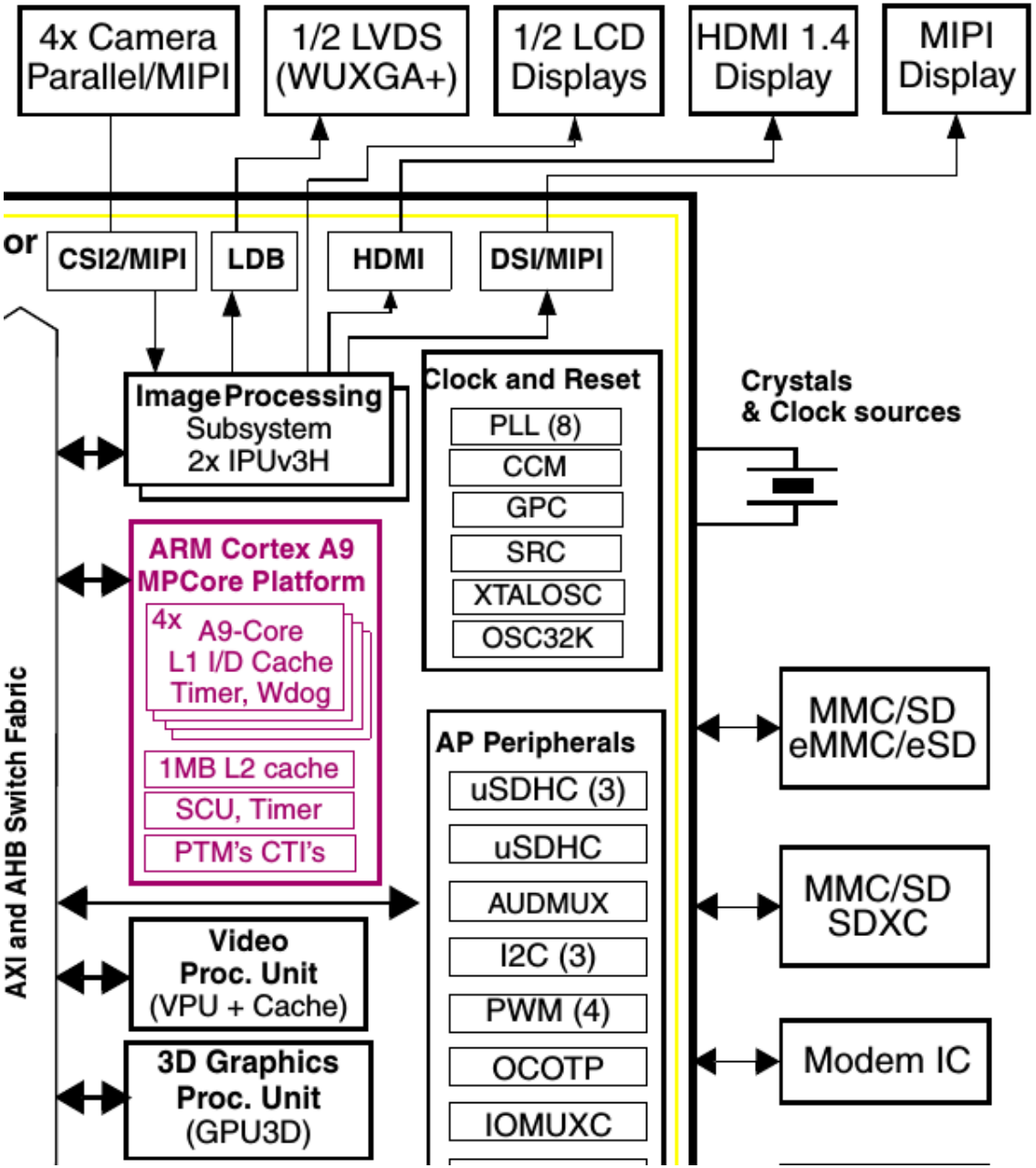


IDEAS ON BOARD

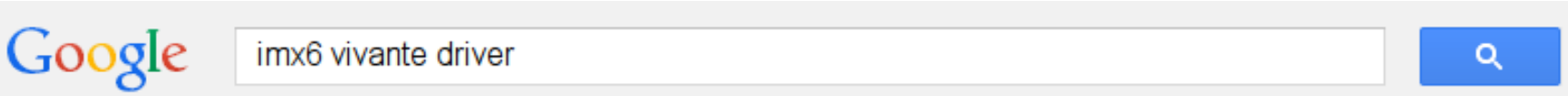
i.MX6

CambridgeGameCreators

CAMPUS LONDON


4x Camera Parallel/MIPI
1/2 LVDS (WUXGA+)
1/2 LCD Displays
HDMI 1.4 Display
MIPI Display
or
CSI2/MIPI
LDB
HDMI
DSI/MIPI
Image Processing Subsystem 2x IPUv3H
Clock and Reset
PLL (8)
CCM
GPC
SRC
XTALOSC
OSC32K
Crystals & Clock sources
ARM Cortex A9 MPCore Platform
4x A9-Core L1 I/D Cache Timer, Wdog
1MB L2 cache
SCU, Timer
PTM's CTI's
AP Peripherals
uSDHC (3)
uSDHC
AUDMUX
I2C (3)
PWM (4)
OCOTP
IOMUXC
MMC/SD eMMC/eSD
MMC/SD SDXC
Modem IC
AXI and AHB Switch Fabric
Video Proc. Unit (VPU + Cache)
3D Graphics Proc. Unit (GPU3D)

- http://nouveau.freedesktop.org/wiki/
- http://limadriver.org/
- https://github.com/freedreno
- https://github.com/kusma/tegra-re
- https://github.com/etnaviv/etna_viv
- https://github.com/hermanhermitage/videocoreiv/



Web Images Videos News Shopping Maps Books

About 4,190 results

Any country
Country: the UK

Any time
Past hour
Past 24 hours
Past week
Past month
Past year

All results
Verbatim

Vivante GPU (Freescale i.MX6) - SDKs - StreamComputing
streamcomputing.eu/knowledge/sdks/**vivante**-gpu/
Vivante has created the GPU-**drivers**, but you have to contact the chip-maker to obtain ...
i.mx6 ubuntu opengl install; opencl **imx6**; alternativ **driver vivante** g2000 ...

laanwj/etna_viv · GitHub
https://github.com/laanwj/etna_viv
14 Oct 2014 **...** Project Etnaviv is an open source user-space **driver** for the **Vivante** GCxxx ...
MX6 , GCABI **imx6** or imx6_v4_0_0 depending on BSP); GC1000.

OpenCL on **imx6** | Freescale Community
https://community.freescale.com/thread/306852
30 Apr 2013 **...** Kernel **driver** for **vivante**. **vivante** vdk. open cl headers. I'm not really sure where to find the **vivante** components here. I've downloaded the SDK ...

Vivante and opengl broken on **imx6** board	22 Nov 2013
Want to try hard float **vivante drivers** for MX6?	7 May 2013
The computing power of the **Vivante's** GC2000 GPU in i.mx6q **...**	21 Mar 2013
OpenGL ES on i.MX6 with X11 - GPU SDK	14 Feb 2013

i.MX6Q: i.MX 6Quad Processors - Quad Core, High Performance **...**
www.freescale.com/webapp/sps/site/prod_summary.jsp?code=i...
Order, Part Number, Suggested **Drivers**, Package ..... Freescale **iMX6**: Soluciones multicore escalables para los nuevos retos de experiencia para el usuario:La ...

IDEAS ON BOARD

Reverse Engineering

CambridgeGameCreators

CAMPUS LONDON

IDEAS ON BOARD

Reverse Engineering

CambridgeGameCreators

CAMPUS LONDON

```
commit b13a0a9e2922cbfbf7b400f4f0fd0acd19f941df
Author: Alexandre Courbot <acourbot@nvidia.com>
Date:   Sat Jul 26 18:41:41 2014 +0900

    drm/nouveau/gk20a: reclocking support

    Add support for reclocking on GK20A, using a
    statically-defined pstates table. The algorithms for
    calculating the coefficients and setting the
    clocks are directly taken from the ChromeOS kernel.

    Signed-off-by: Alexandre Courbot <acourbot@nvidia.com>
    Signed-off-by: Ben Skeggs <bskeggs@redhat.com>
```

Assembling the Pieces

ubuntu

debian

fedora™

IDEAS ON BOARD

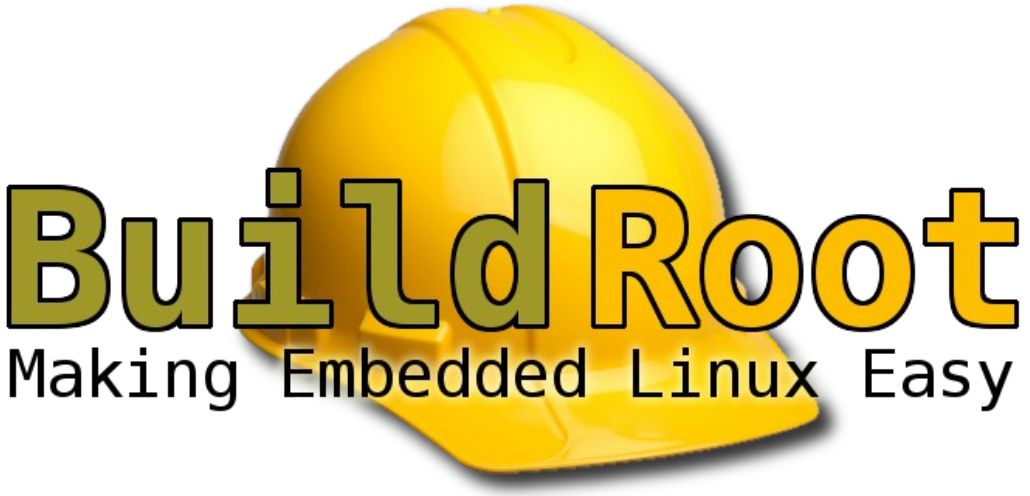
Distributions

CambridgeGameCreators

CAMPUS LONDON

Distributions

yocto PROJECT

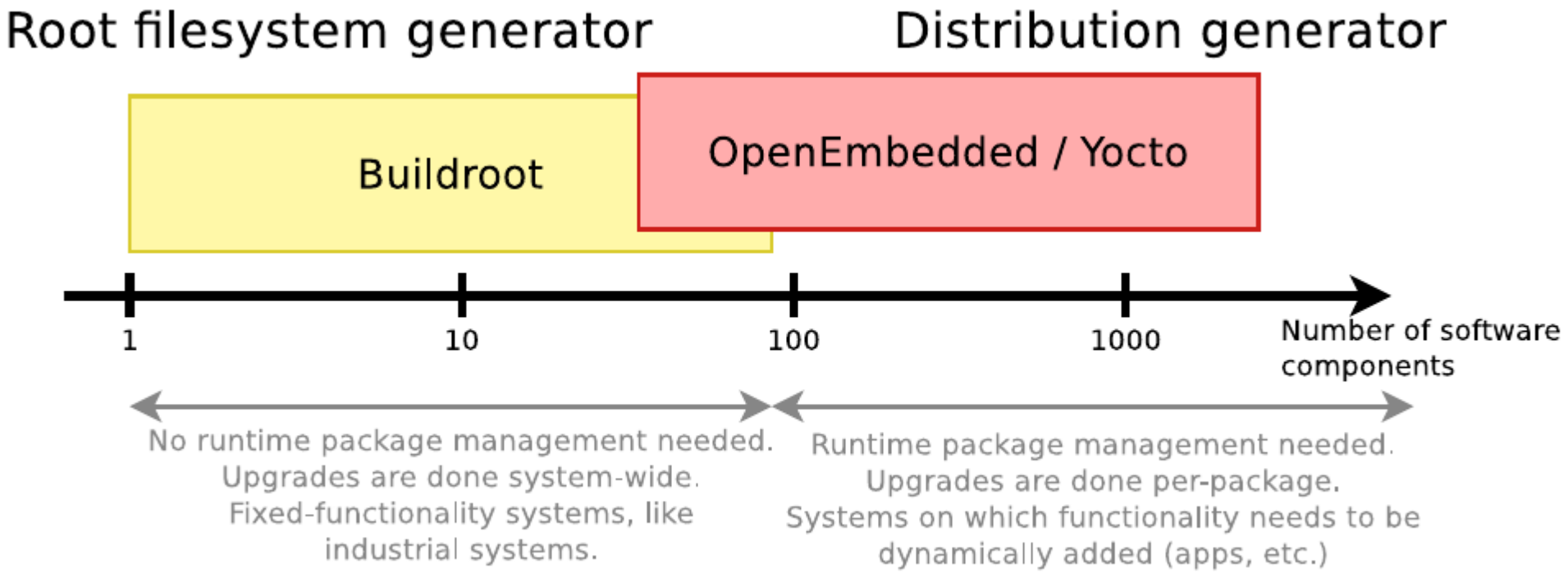


IDEAS ON BOARD

Meta-Distributions

CambridgeGameCreators

CAMPUS LONDON



Credit: http://free-electrons.com/pub/conferences/2012/lsm/buildroot/

- FBDEV, DRM/KMS
- Bare, X11, Wayland
- Mainline, BSP
- Distribution

Summary

Resources

- http://free-electrons.com/docs/
- http://phd.mupuf.org/
- dri-devel@listsfreedesktop.org

Resources

laurent.pinchart@ideasonboard.com

Contact

IDEAS ON BOARD

CambridgeGameCreators

CAMPUS LONDON

Thank you.