# Linux 显示介绍

图形显示-姚智情

# 一，已支持芯片图形显示情况

| 芯片 | 支持图形框架 |
| --- | --- |
| Rk3066(mali400)<br>Rk3188(mali400) | Xserver + xf86-video-mali(仅支持 3d 加速)<br>Wayland fbdev backend(仅支持 3d 加速) |
| IOTG(mali 450) | Xserver + xf86-video-armsoc(仅支持 3d 加速)<br>Wayland drm backend(支持 2d 和 3d 加速)<br>Gbm (支持 gpu 加速) |
| Rk3128(mali400)<br>(基于 kernel 3.10 drm 兼容层) | gbm(支持 gpu 加速) |
| 3036(mali400)<br>3066(mali400)<br>3328(mali450)<br>3288(malit76x)<br>3399(mali86x)<br>(使用 kernel 4.4 内核，mali utgard 和 midgard 架构的均支持) | Xserver + glamor + xf86-video-modesetting(支持 2d 和 3 d 加速)<br>Wayland drm backend(支持 2d 和 3d 加速)<br>Gbm (支持 gpu 加速) |
| Rk3368 | 由于 PVR 的 linux gpu 集成包需要授权，无集成包， 所以 rk3368 linux 图形暂无开发计划. |

# 二，相关资源下载:

## 1，相关文档资源

1) http://opensource.rock-chips.com
公司的 linux 开源网站，里面已经对 rockchip linux 开源开发做了很多描述, 可作为**重点阅读**研究.

2) http://opensource.rock-chips.com/wiki_Graphics
介绍 rockchip linux 图形 gpu 相关的知识.

3) https://github.com/rockchip-linux
对外发布开源的 git 工程，里面包含 mali 库，xserver，rootfs 的构建，视频 mpp 等等.

## 2, linux mali ddk

Kernel 部分：已内置于 29 kernel 工程.
User 部分:
    Midgard: git clone ssh://[username]@10.10.10.68:29418/graphics-midgard
    Utgard: git clone ssh://[username]@10.10.10.68:29418/graphics-utgard
编译方法参见 ddk 根目录的 readme.txt
集成方法:
EGL/GLES:
ln -s libmali.so libEGL.so
ln -s libmali.so libGLESv1_CM.so
ln -s libmali.so libGLESv2.so
Gbm 相关:
ln -s libmali.so libgbm.so
Opencl:
ln -s libmali.so libOpenCL.so
Wayland:
ln -s libmali.so libwayland-egl.so

将这些 gpu 库替换掉板子上 mesa 的实现即可.

# 3, Xserver 相关资源

源码：https://github.com/rockchip-linux/xserver
编译: 产品部门已经总结了较多的编译脚本，可由产品部门提供编译环境和脚本(docker, buildroot 或者是在板子上直接编译的脚本)

1) Xf86-video-mali:
**下载地址: https://developer.arm.com/products/software/mali-drivers/display-drivers**

2) Xf86-video-armsoc:
https://github.com/rockchip-linux/xf86-video-armsoc

3) xf86-video-modesetting
内置于 xserver 工程: hw/xfree86/drivers/modesetting

4) Glamor
内置于 xserver 工程: glamor 和 hw/xfree86/glamor_egl

# 4, wayland/weston

源码: https://cgit.freedesktop.org/wayland/wayland
https://cgit.freedesktop.org/wayland/weston/
编译：autogen.sh && configure && make && make install

# 5, mutter(基于 fedora 环境)

Mutter 和 weston 是类似的东西，和 weston 一样，都是 wayland server 是一个实现，mutter 是为 gnome  桌面支持 wayland 而开发的 wayland server.

Fedora 环境获限源码及编译都比较方便，所以这边简单介绍一下 fedora 包的编译
以 mutter 为例:

1) 源码: dnf download --source mutter
2) 安装相关依赖:
   dnf build-dep mutter
   rpm -i mutter-xxx.rpm
3) 安装编译
   cd ~/rpmbuild/
   rpmbuild -bc SPECS/mutter-xxx.spec
   cd ~/rpmbuild/BUILD/mutter-xxx/

后续就可以重复源码修改和 make install 编译步骤了.

   make
   make install

# 6, Xwayland

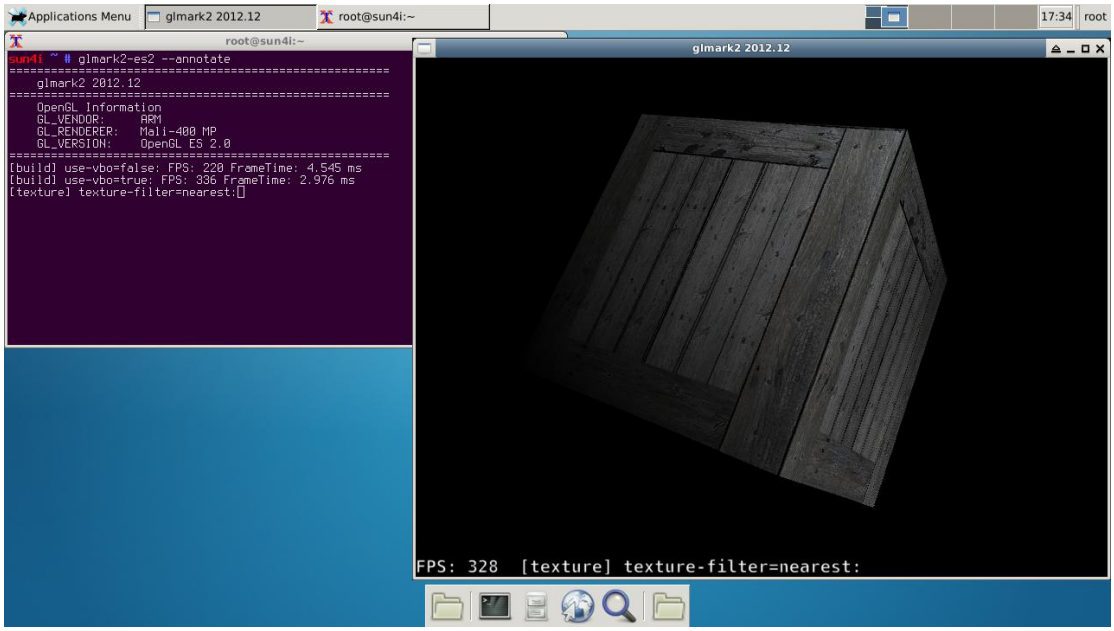
Xwayland 需要跑在 weston 或者 mutter 之上，做为 wayland 的 client, 可以支持 glamor.
Xwayland 内置于 xserver 工程：位于 hw/xwayland

通过 Xwayland, x11 的应用就可以跑在 wayland 环境底下.

# 三, Xserver, wayland, gbm 场景对比:

## 1, Xserver

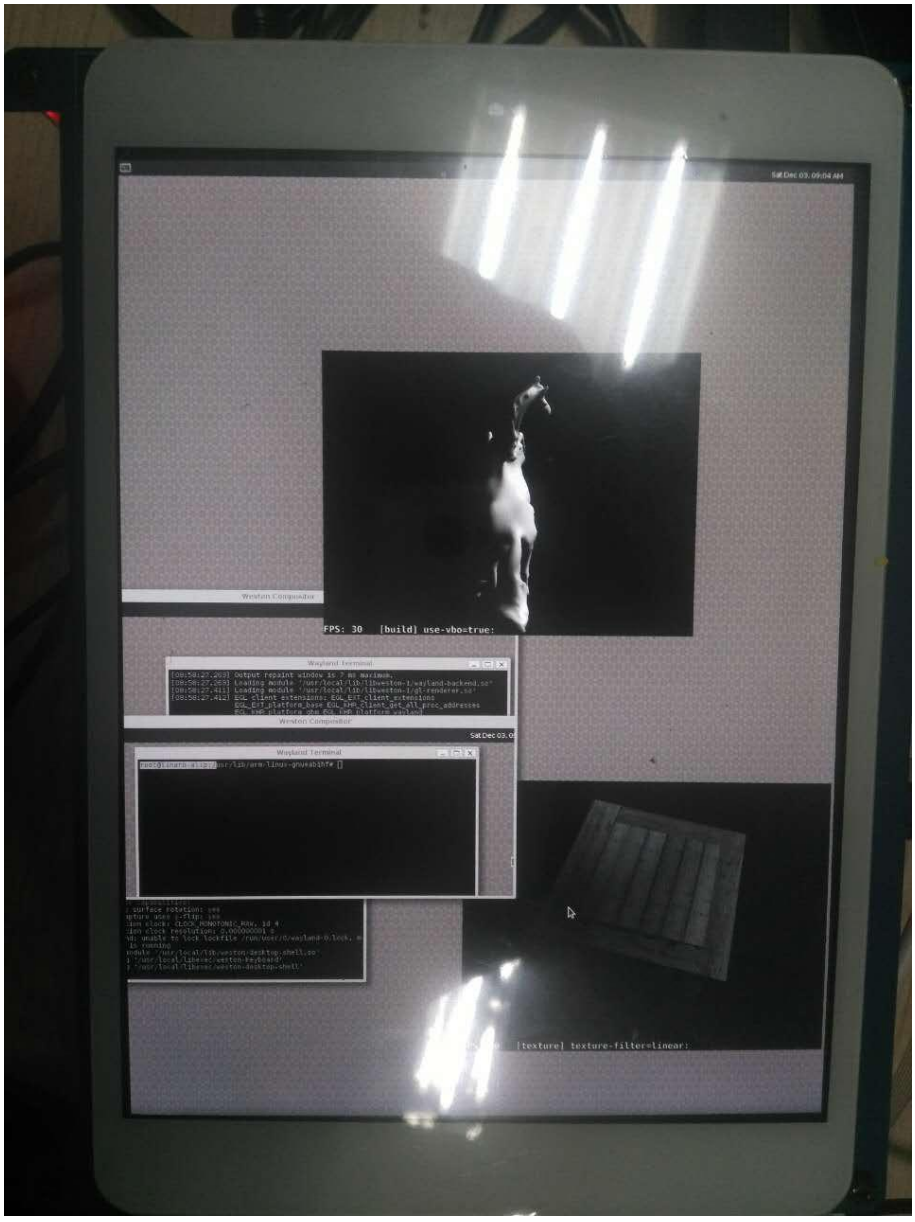
优势:  有成熟的 linux 发行版支持, 巨量的开源支持,  有较成熟的生态,  功能最为完善.
缺点:  显示框架比较老旧,  性能较差.
总结:  成熟度高,  但性能差
使用场景:  客户上手容易,  有较高自由度,  有大量开源生态支持.



# 2，Wayland

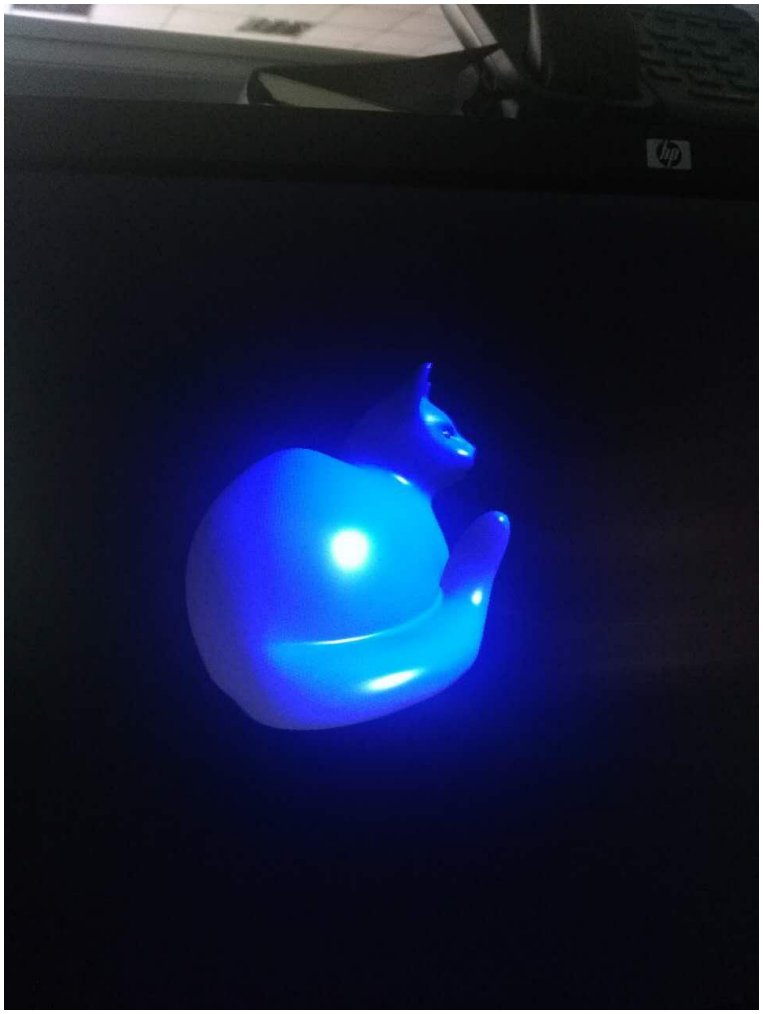
优势:  做为未来 linux 图形显示的趋势, wayland 融合了先进的图形思维,  能够达到较高的图形性能,  部分 linux 发行版已经支持 wayland，大量新的开发,  新的 feature 都是基于 wayland.
缺点:  基于 wayland 的桌面还很少,  缺少完整的生态.
使用场景: 对性能有需求,  社区支持一般

注：通过 Xwayland 支持，wayland 桌面上就可以跑 x11 应用，扩大了 wayland 的生态，但这方面的稳定性还是不如纯 xserver 桌面，但后续会越来越完善，fedora 已经 release 支持 wayland 的版本，ubuntu 据了解也快切成 wayland 的环境了.



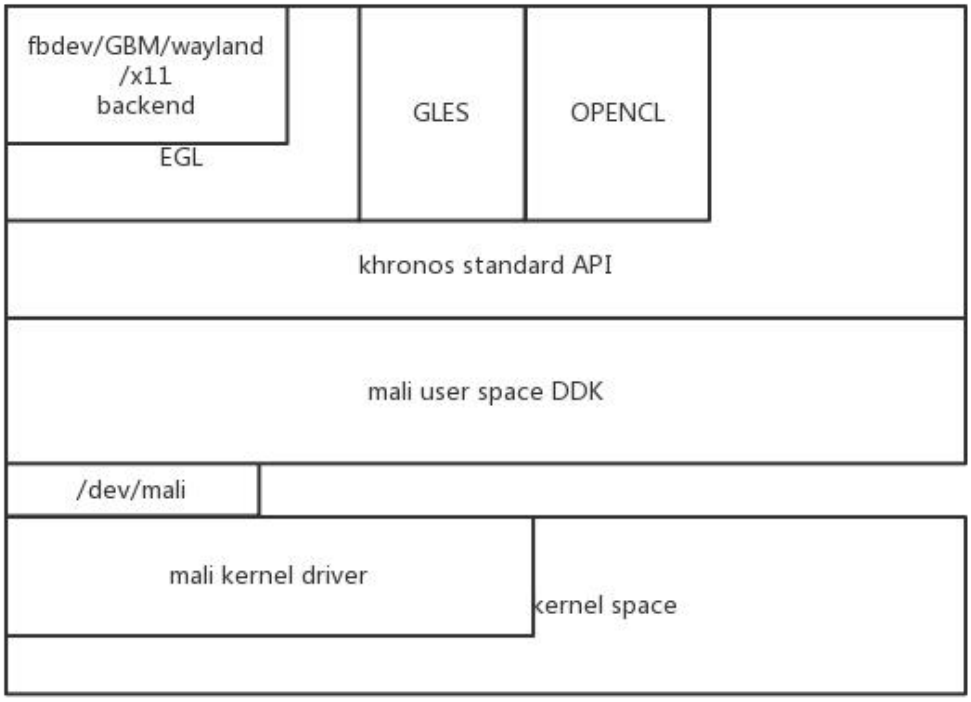
## 3，Gbm

优势:  用户支接使用 gpu api 进行画面绘制，  没有合成叠加，  性能最高.
缺点:  没有窗口管理，  只能跑单应用.
使用场景:  单应用，  对性能有需求，适合做单应用的瘦客户端



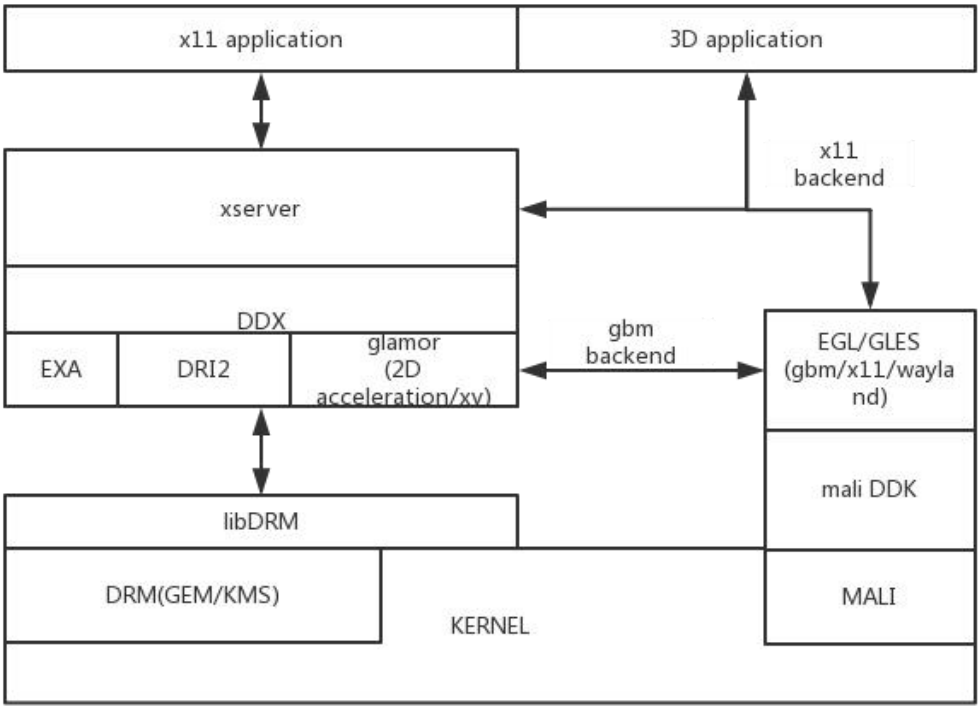
# 三，xserver 显示框架分析

## 1， Linux 上 mali 的框架

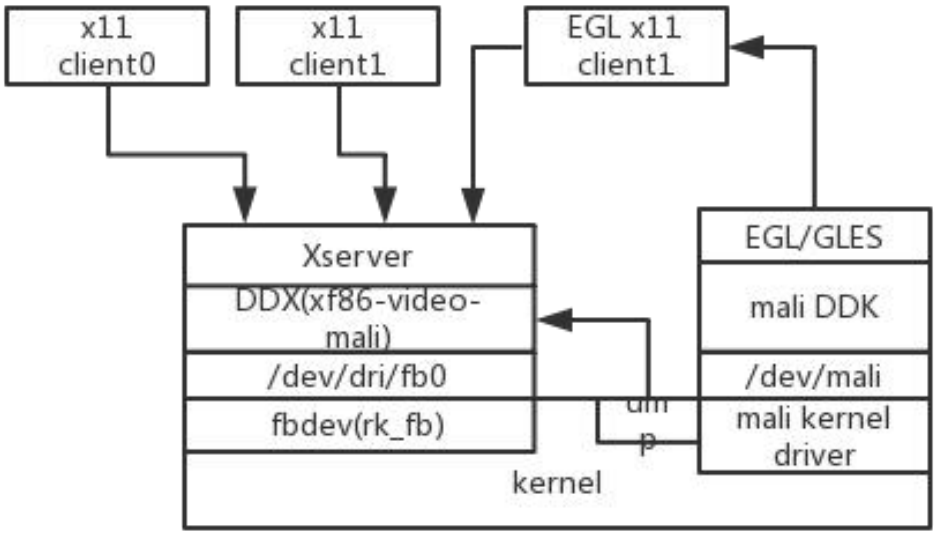



Mali 对外的 api 都是标准的 khronos 定义，所以可以复用开源社区中使用标准 khronos api 的 gpu 程序.

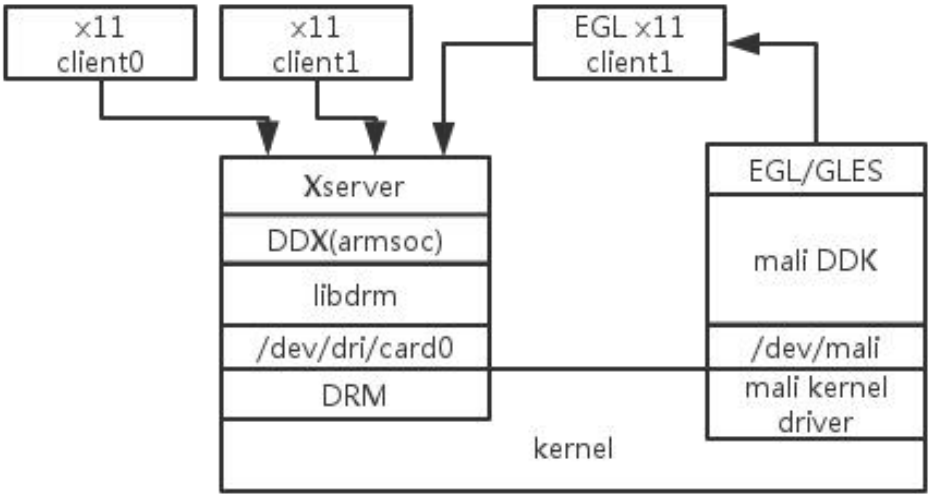
## 2 ,Xserver + glamor + xf86-video-modesetting




## 3 , Xserver + xf86-video-mali



## 4, Xf86-video-armsoc



# 四，概念介绍:

## 1 Xserver:

### 1) Xf86-video-mali

Mali 官方提供， xserver 上支持 fbdev 的 3d 加速 ddx

### 2) Xf86-video-armsoc

Mali 官方提供， xserver 上支持 drm 的 3d 加速 ddx

### 3) Xf86-video-Modesetting

Xserver 唯一内置的 ddx，随着 xserver 一起更新， 更新更频繁， 意味着 bug 更少，更稳定， 只要 kernel 支持 drm，就能使用 modesetting ddx，所以用的厂商会很多， 也是一个趋势， 而且带上 glamor 后,可以同时支持 2d 和 3d 的加速， 所以我们抛弃了 xf86-video-mali 和 xf86-video-mali，拥抱了 xf86-video-modesetting.

# 4) Glamor

Xserver 上使用 gpu 进行 2d 加速的框架，支持 opengl 和 opengles.

官方只支持基于 mesa 的 opengl/opengles。 使用 mali 来支持 glamor, 需要做很多工作.

使用 glamor 的优点:

1, 性能会比纯 cpu 的 xserver 高， 特别是窗口移动的场景， 比较明显.

2, 支持 mesa 开源图形框架的厂商主要是 pc 平台的, arm 平台的基本没有, mali 官方也不提交 glamor 支持， 我们做了很多工作才将 glamor 适配到 mali gpu 上， 其他竞争对手要上手 glamor 有一定难度， 所以 xserver gpu 加速方面我们有优势.

3, 有现成支持 glamor 的 modesetting ddx, 同时 modesetting ddx 可以支持 egl 3d, 所以基于 modesetting + glamor 的图形框架可以同时加速 2d 和 3d. Mali 官方提供的 xf86-video-armsoc 和 xf86-video-mali 只能加速 3d 应用,对于 2d 图形框架无能为力.

缺点:

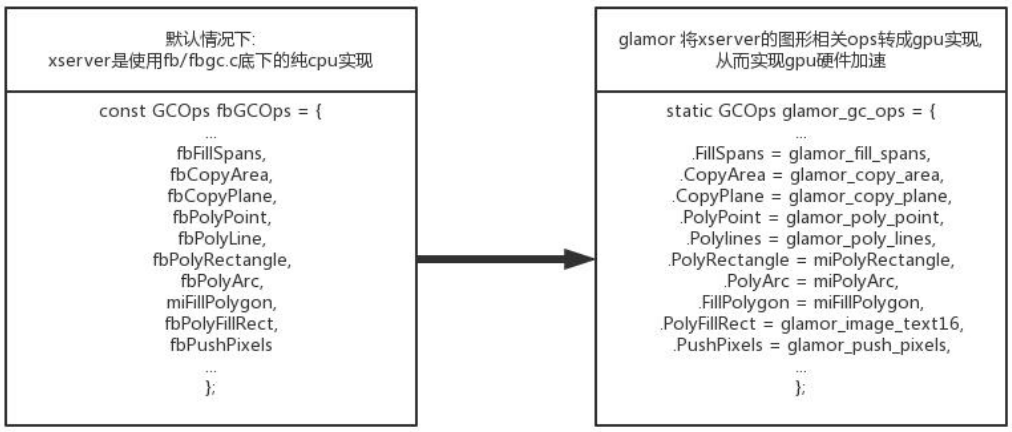
1, 由于使用 glamor 的厂商比较少， 基本上是 intel， redhat 这些老牌的 pc 平台厂商在搞 glamor, 而且主要是针对 opengl, 对于 opengles 开发验证比较少, bug 比较多.

2, mali 官方不对 glamor 支持， 所以有许多适配的工作

3, glamor 有些显示操作 gpu 无法支持， 就会返回到 cpu 端， 这些的图形操作就会比纯 cpu 平台的慢.

Glamor 运行机理:

Glamor 的实现原理就是将 xserver 的绘图 api 转成 gpu 实现，默认情况下，xserver 是使用纯软件实现的 fbGCops, 使用 glamor 后，就会被替换 gpu 实现的成 glamor_gc_ops.

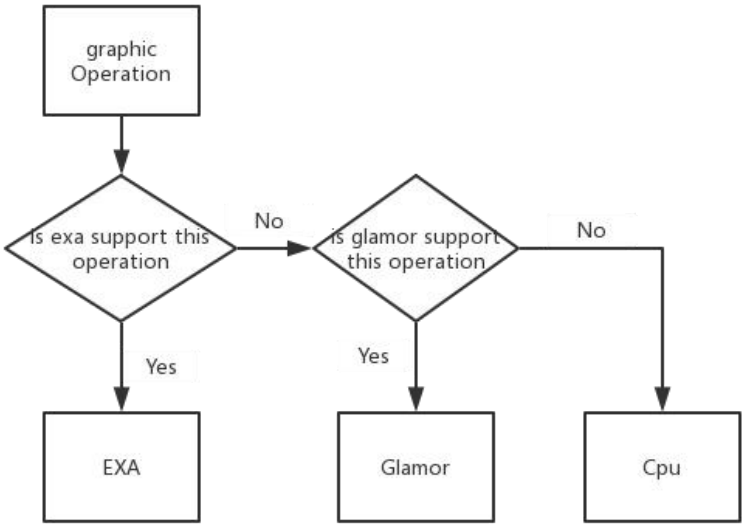


# 5) Exa

Xserver 另一个 2d 图形加速框架.
可以加速图形的 Copy, Solid, Composite 这三个操作， 这三个操作在 xserver 中对性能有很大的影响， 加速这三个操作会显著提升 xserver 的图形性能.

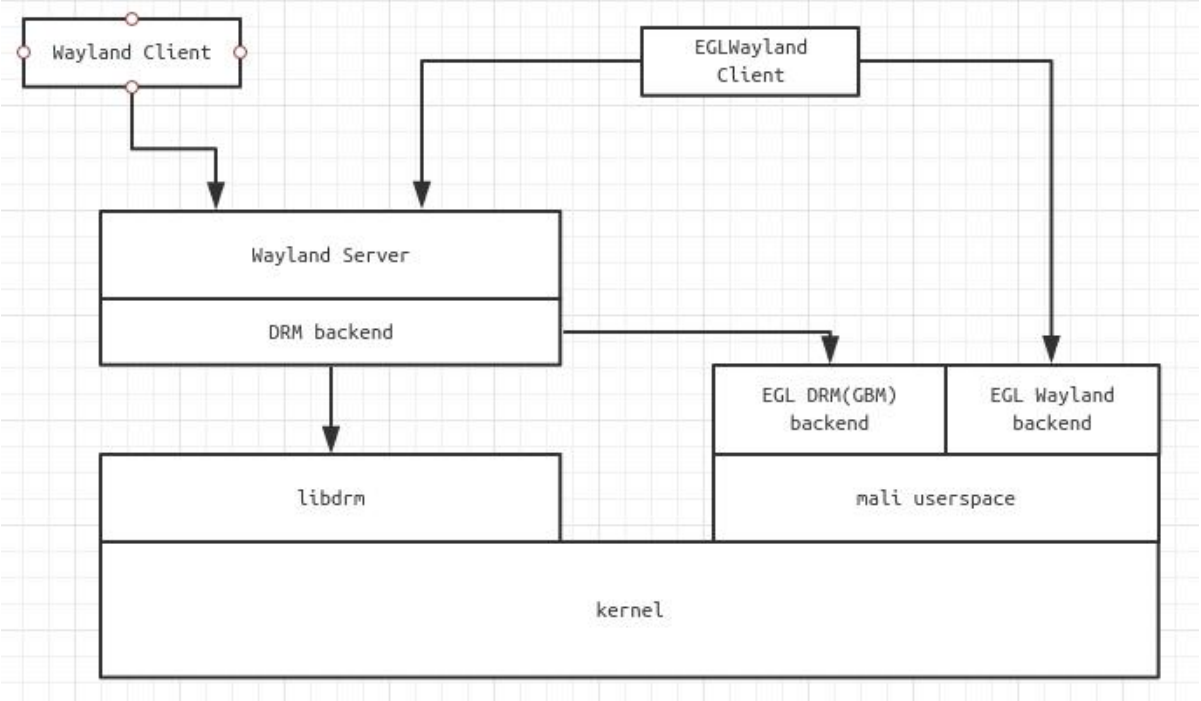
我们能使用 EXA 的硬件是 RGA, 产品三部那边已经有一个基于 RGA 的硬件加速， 但是由于某些原因， 性能还比较差， 还无法使用. 这是以后优化的一个方向.
Xserver 上图形的最终优化路线应该是这样:

EXA 和 glamor 是一个互补的图形框架， 不是替代关系， 当 EXA 支持的图形操作， 会走 EXA 加速， 不支持的， 会返回到 glamor 来进行加速， 如果 glamor 也不支持， 就使用 cpu 来绘制.

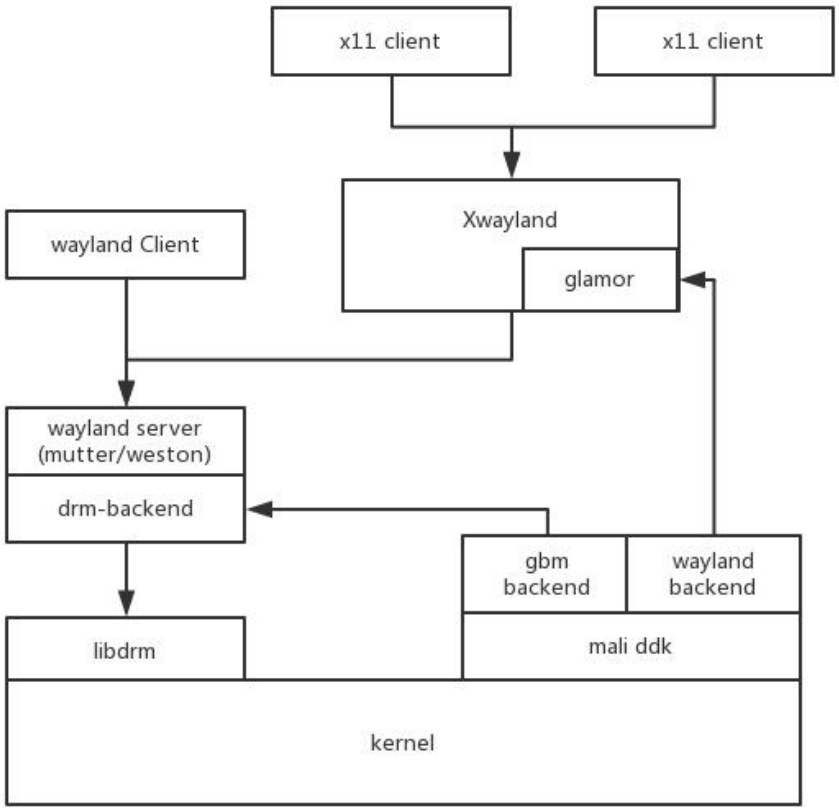


# 2, Wayland/weston

Wayland 在 IOTG 项目上产品化过， 这部分我们的改动比较少， 基本是基于 mali 给的集成包． 在 Utgard(mali400/450)和 MidGard(malit76x/86x)两个系列的 mali gpu 上均已集成过, 都能跑起 wayland 的图形支持.



# wayland 桌面: mutter/Xwayland (fedora)




以上框图是 **fedora** 使用 **gnome-session-wayland** 跑起 **linux** 桌面的图形栈，必竟 **x11** 的应用是主流，所以这边使用了 **Xwayland** 做为 **x11 app->wayland** 的兼容层，这样 **x11** 和 **wayland** 的应用都能在这套显示桌面上运行.

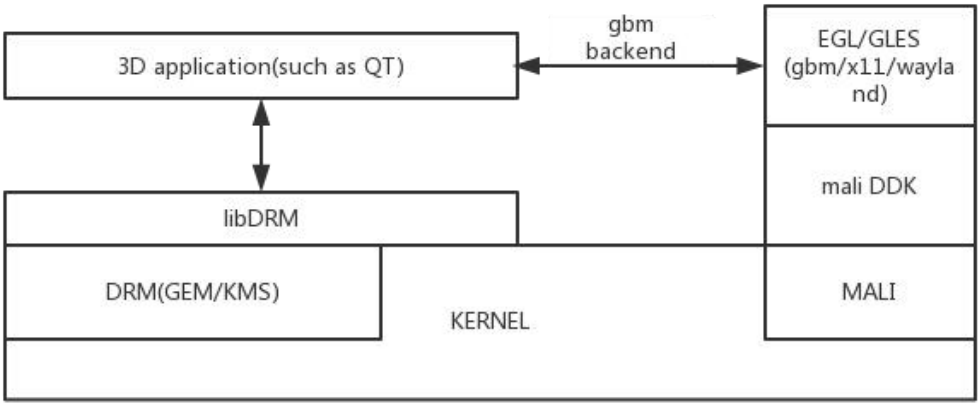
**Mutter/xwayland** 和第二点中的 **wayland/weston** 是同样的框架，只是分属不一样的实现.

# 3, Gbm

GBM (Generic Buffer Manager) basically provides a EGL native window type (just like Wayland and X11), so one could obtain a real EGL surface and create render target buffers. With that then, GL can be used to render into these buffers, which will be shown to the display by queuing a page flip via KMS/DRM API.

使用 gbm 可以为单应用提供 gpu 加速， mali 官方提供了这种方式的加速， chromeos 系统上目前所使用的 gpu 后端就是 gbm， 很多应用支持 gbm.

如果做单应用项目， 使用 gbm 是最为高效的， 如 QT 就支持 gbm， 或以做 QT+gbm 的瘦客户端.




以下是 gbm 的一个简单 demo, 通过该 demo 可以了解一下 gbm 和 gpu 的关系:
https://github.com/eyelash/tutorials/blob/master/drm-gbm.c

# 4, MESA vs Mali

Mesa 的详细介绍参看：
https://en.wikipedia.org/wiki/Mesa_(computer_graphics)

Mesa 支持 opengl 和 opengles, 提供标准的 egl/gles api, 当发现不支持硬件 gpu 时，会走软件实现方式.
Mali 仅支持 opengles, 不支持 opengl.

Mesa 是默认集成到各个 linux 发行版上的，经常可以看到客户集成 gpu 时，发现系统在找 rockchip_dri.so, 这个是由于客户的 linux gpu 集成不正确，应用使用的 gpu 库是 mesa 的实现，mesa gpu 框架会根据底层 drm 驱动名称进行库查询，在 rockchip 平台上就会尝试匹配查找 rockchip_dri.so, 但实际上是没有这个库文件的.

当发现有这个 rockchip_dri.so 时，就需要确认一下 mali gpu 库是否集成正确.