

# allocation

Brian Starkey – Arm – Display SW

# Part 1) Allocator module

- Allocation is a hard problem
  - Cross-device sharing, performance, efficiency
- Android has gralloc
  - Userspace module for handling all graphic buffer allocation
- Chrome has minigbm
  - Not so comprehensive, but works
- “Linux” has ???
  - gbm, sort-of
- Any long-term plan? Should we make one?

# [cubanismo/allocator](#) / UDMA / liballocator

- Interesting for solving some problems
  - Intersections of device capability and constraints
  - Conceptually similar to gralloc
- Any future?
- We spent some time on our use-cases – found some issues
- Doesn't actually cover physical allocation (part 2)

## Part 2) Actually allocating

- Once you've decided what to allocate, how do you allocate it?
- Centralised / decentralised?
- Is it possible to be generic?
- Can we suit Android, Chrome and Linux?

# ion

- Where does ion stand in Linux (not-Android) buffer allocation?
- string-based UAPI likely not stable enough
- Not likely to be adopted in desktop?

# DUMB

- Used/abused (at least by us)
- Limited, by design.
- Little interest in extending?
  - e.g. Modifiers
- “strange” UAPI – width, height, bpp
  - Can’t know pixel format

# New generic GEM ioctl?

- Should we define a new generic DRM ioctl for allocation?
- Can we unify any of the existing vendor-specific allocators?
- How would it look?
- Minimum:
  - We'd like to allocate buffers with fourcc + modifier.
- Should usages form part of it?