Allocation constraints workshop summary

James Jones & Simon Ser

Allocation constraints workshop summary

- No objections to high-level design
- No objections to common function to merge constraints
- One constraint set per modifier should work fine
- Vendored constraints ala modifiers: no need for now, but current design leaves the door open
- Constraints set only have a serialized form → can send them over Wayland/X11
- EGL API: `external_only` probably good enough of a proxy for texture/render usage

Allocation constraints workshop summary

- Allocators: GBM + Vulkan. Not a good fit for e.g. camera + encode. Design should work with new allocators.
- Possible tricky hw limitations?
  - BO planes each in a different bank, shadow buffer blits between layout transitions
  - Extra pixels with metadata → not fully solved by constraints
- Home for the merging library?
  - uAPI header: everything is public and inlined downstream
  - vDSO: need to justify inclusion in the kernel
  - Shared user-space library sync’ed from kernel (possibly part of libdrm)

Allocation constraints workshop summary

- Constraint for CPU access: “cached/uncached”
  - Can expand mmap’ing APIs (e.g. GBM) to take usage flags (RO, RW, sequential or random-access, etc)
- We need to refine our heap ID proposal
  - Work with dma-buf heap designers
  - Start with local unshareable & non-local shareable: find a way to expose and discover these heaps
- More concrete proposal to come
  - James to start an email chain
  - Post gitlab issues here: https://gitlab.freedesktop.org/emersion/drm-constraints