Interoperability & Cooperation (1/2)

- Buffer sharing with dmabuf
  - Good on paper, broken in practice
  - Too little communication between the subsystems
  - Fix for V4L2 cache handling available, breaks other use cases

- Extensive documentation
  - Focus is on kAPI for DRM/KMS, uAPI for V4L2
  - Opportunities for cross-improvements

- Camera / Display pipeline configuration
  - Different formats and constraints, not reported by DRM/KMS, reconciliation is a hard problem
  - Solved in some userspace frameworks (e.g. gstreamer) but still painful in general
  - Difficult to influence the stride with dumb buffers in DRM/KMS
  - Drivers need to report more information, arbitration has to live in userspace

Interoperability & Cooperation (2/2)

- Enumeration & Configuration
  - DRM/KMS reports static information, and accepts or rejects a configuration – simple and easy but limited
  - V4L2 enumerates capabilities dynamically, and negotiates configurations – powerful but complex and hard
  - DRM/KMS could benefit from a more dynamic and negotiated approach, but care must be taken to not let drivers get it wrong
  - Failure hints reported by ATOMIC_TEST ?
  - Needs to be very carefully considered and solved for a camera API, regardless of where in the kernel it lives
  - A common mechanism covering graphics and cameras is desired even if the implementations are separate

A new camera API

- A camera atomic API is needed
- It took 5 years for display, we need it tomorrow for cameras
  - V4L2 has a tendency to be developed in bursts as long as a big corporate interest exists, and then die out
- How ?
  - Hard to implement on top of V4L2 (too many syscalls, legacy code hinders refactoring, …)
  - Can DRM/KMS come to the rescue ? Enables code sharing (bridges, writeback, …)
  - Something new, a.k.a. NIH syndrome ?