



A case for a media-job scheduler

Linux-media summit 2026

- Proposed by Dan Scally @ IoB
 - <https://www.spinics.net/lists/linux-media/msg279502.html>
 - June 2025 – old but not bitrotten
- “The framework is intended to facilitate communication between separate drivers which need to work together to fully operate a media pipeline”
- This presentation is about “the problem” and not on “the solution”
 - Patches are on the list :)

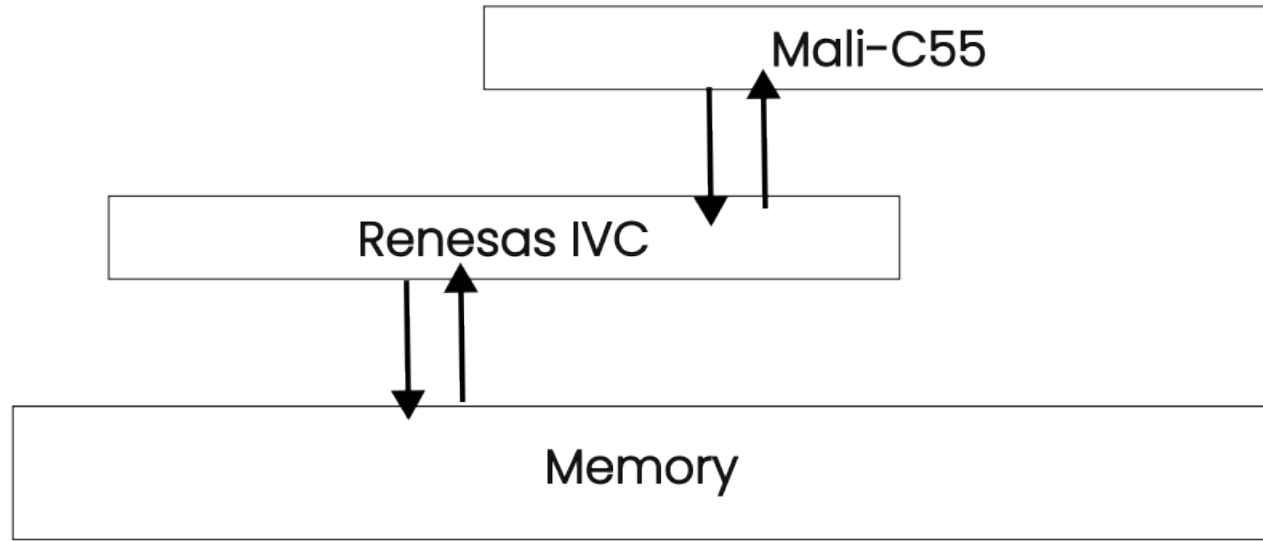
Media-job scheduler



- ARM Mali-C55
 - Support both in-line and m2m modes
 - The IP has no memory interfacing capabilities
- First integration in Renesas RZ/V2H(P) in m2m mode
- Paired with a companion chip that provides memory interfacing
 - Renesas IVC

Renesas RZ/V2H(P)

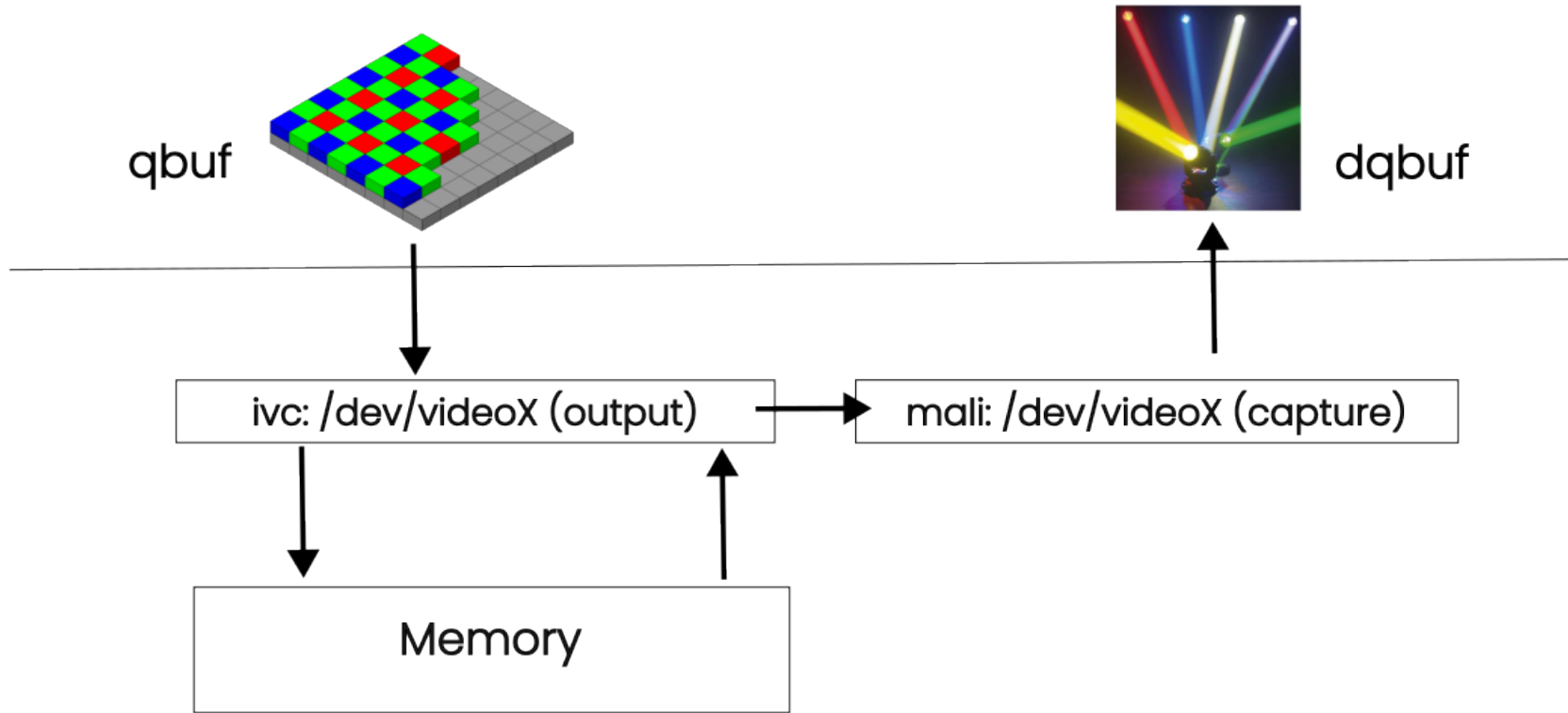




- `drivers/media/platform/renesas/rzv2h-ivc/`
- `drivers/media/platform/arm/mali-c55`

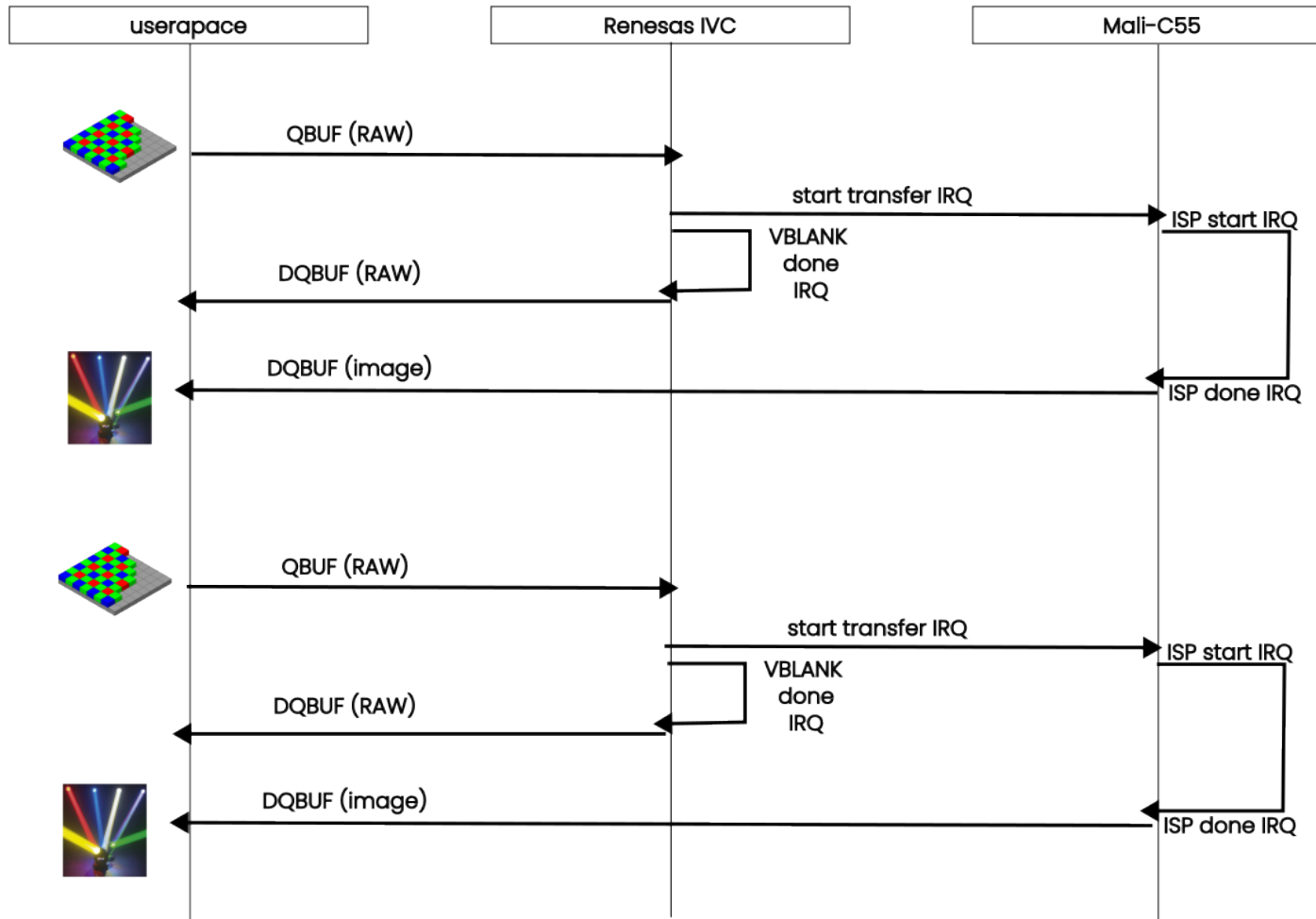
Renesas RZ/V2H(P)





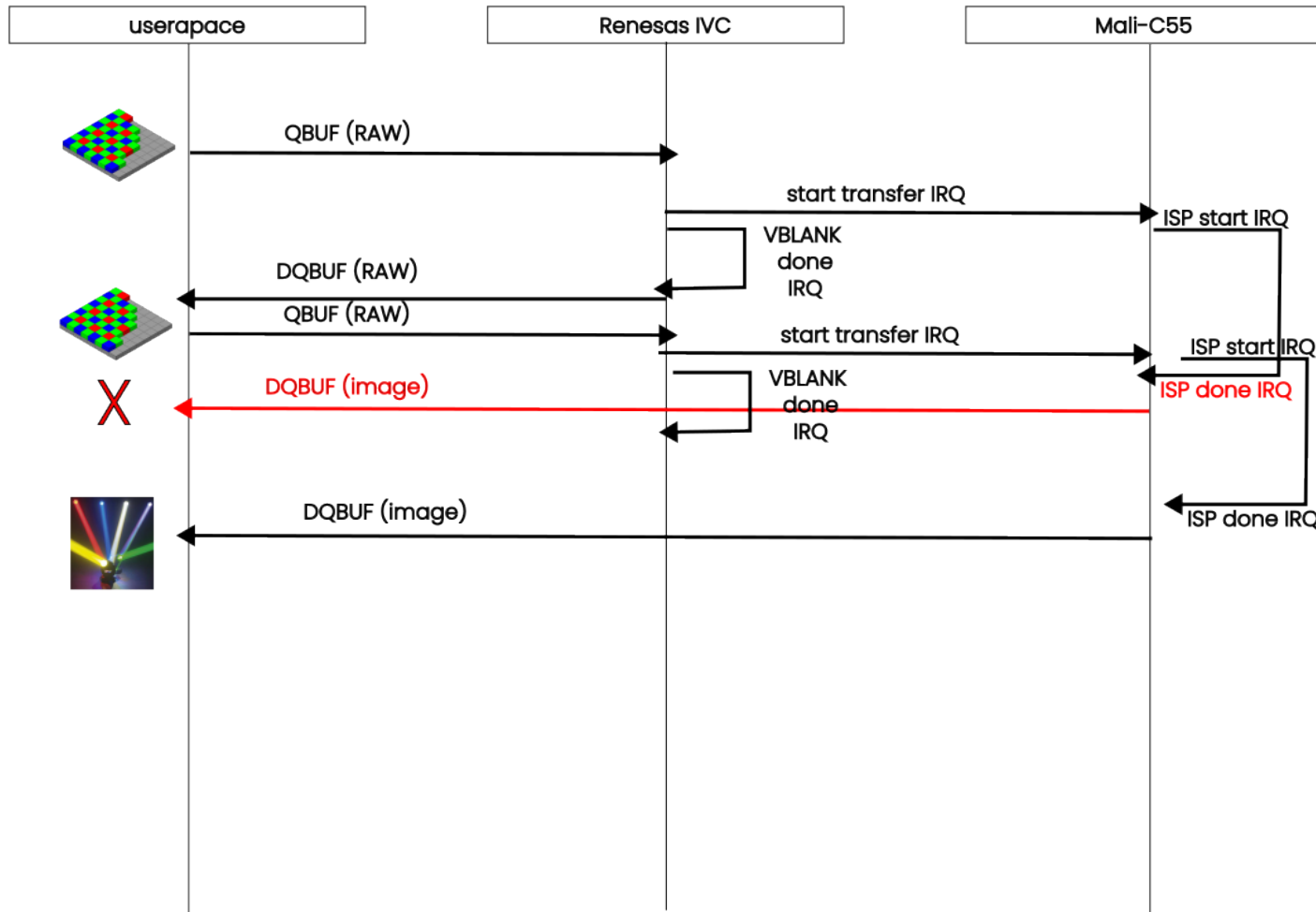
Renesas RZ/V2H(P)





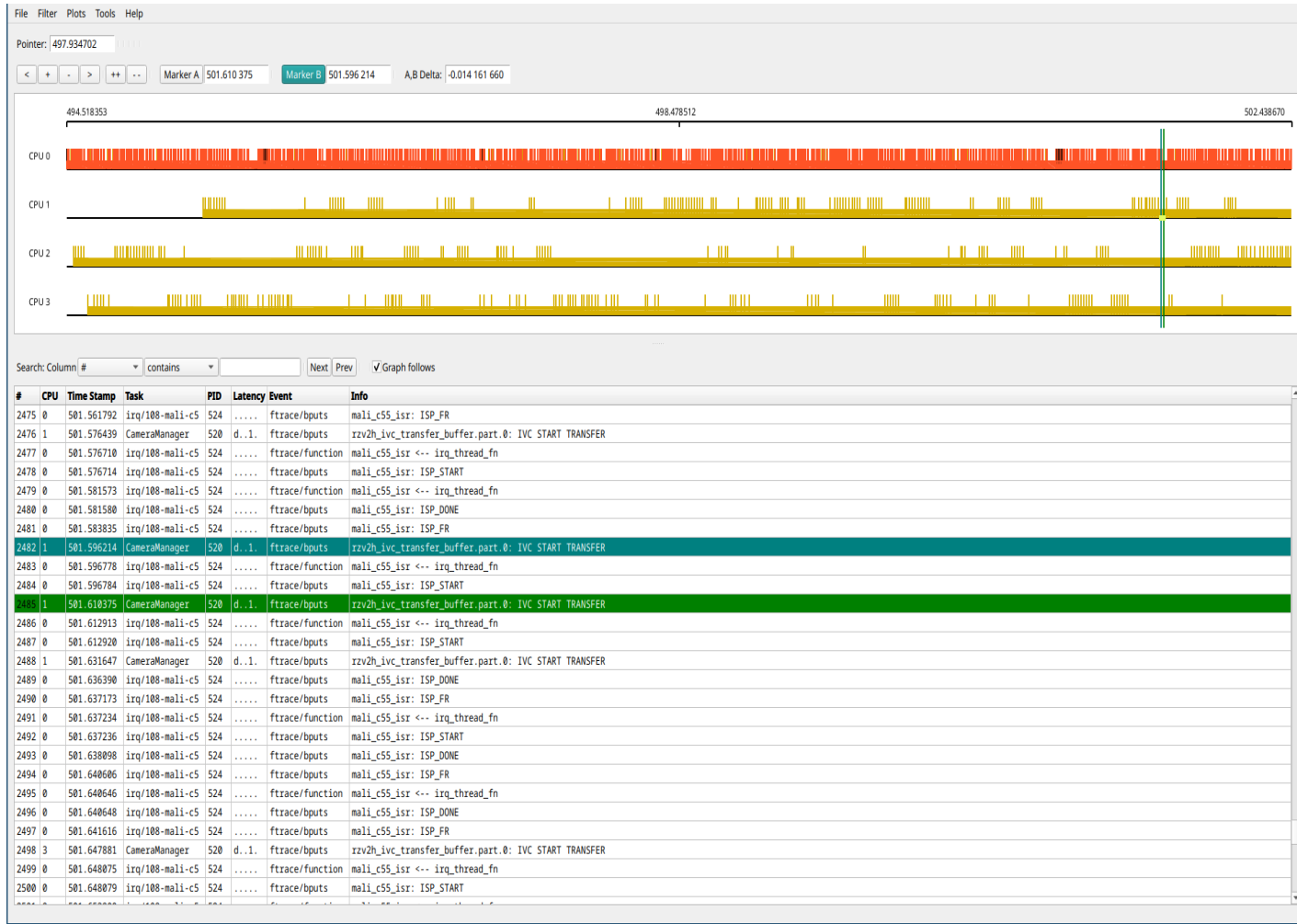
Operation sequence





(broken) operation sequence





(broken) operation sequence



- Userspace should not control the scheduling of job processing
- The ISP should “ask” for a new job to be started once it is done processing
 - Of course we can’t do cross-driver calls
 - The “pipeline” should be dynamically composed
 - Drivers can “subscribe” and register steps
 - The pipeline shall be walked to create a job and schedule it

Ideal operation model

