

DRM Lease Manager

for Display Device Sharing

Outline

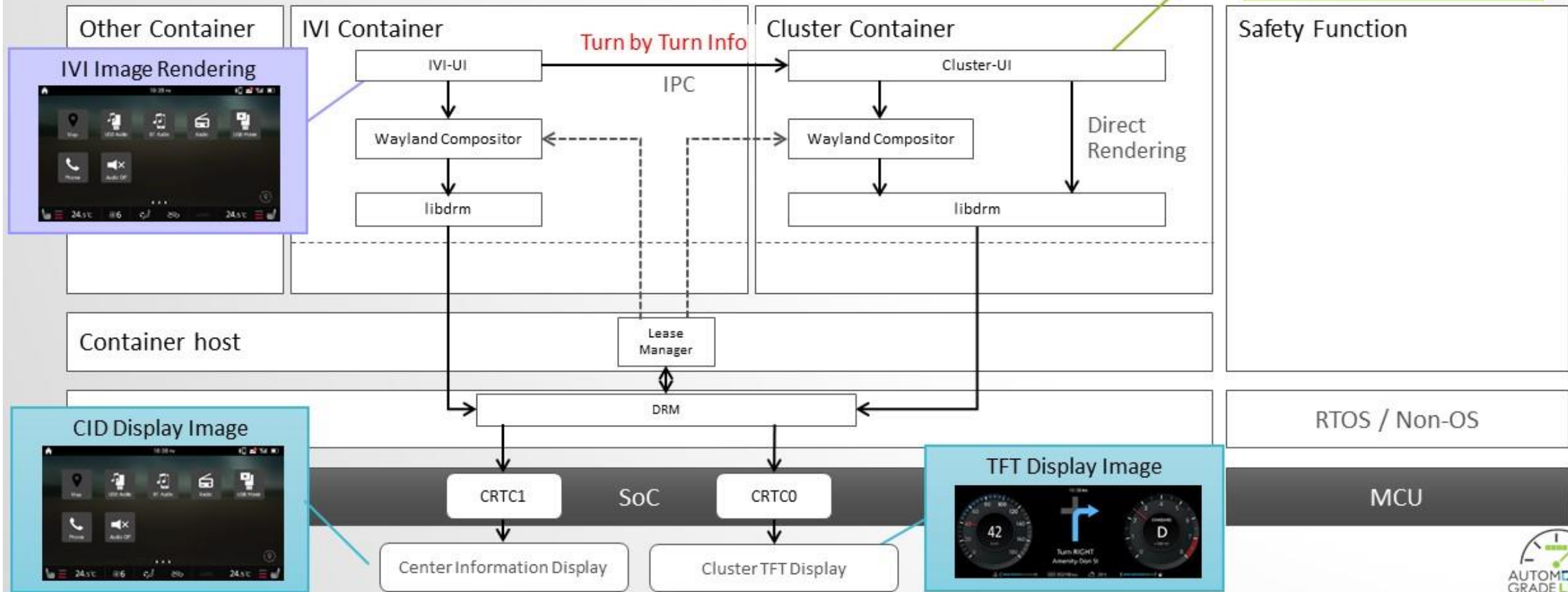
1. Use Case and Target Platform
2. DRM resource sharing
3. DRM Lease manager usage
4. Current Status and Next Steps

Use Case and Target Platform

DRM Lease use case

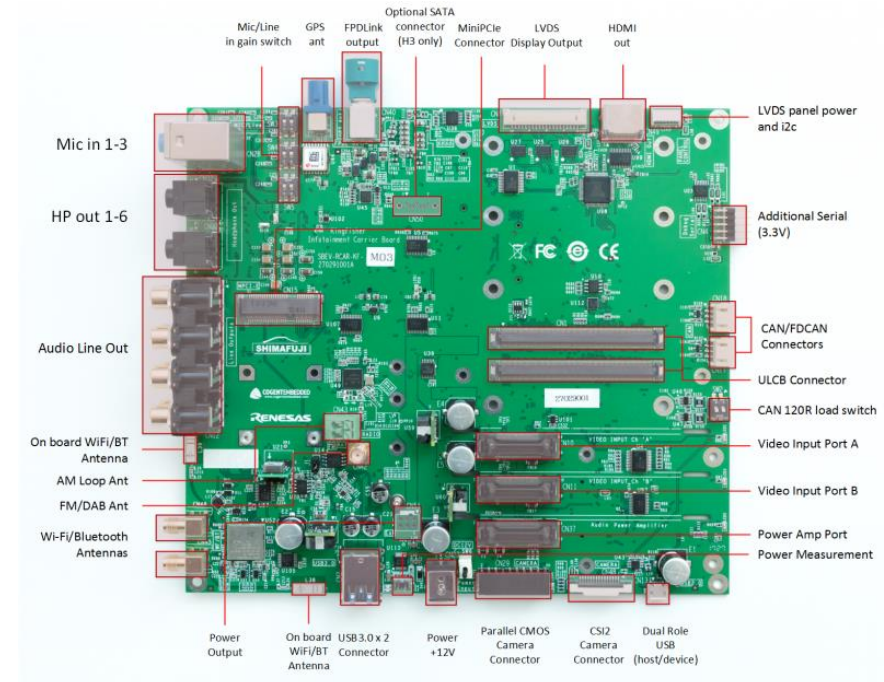
Multiple container DRM sharing shall be done by introducing DRM Lease Manager.

- GPU rendering/composition shall be done in application container, not container host
- It allows application container to render directory to the DRM device.
- It ensures other containers can still display their HMI via Weston.
- It allows both types of containers to render to the DRM device in parallel.



Target environment

- Software
 - AGL UCB (10.0 Jumping Jellyfish)
- Initial Hardware Target
 - Renesas R-Car SK + Kingfisher extension board
 - 1 HDMI output from R-Car SK
 - 1 HDMI output from Kingfisher



<https://elinux.org/R-Car/Boards/Kingfisher>

DRM Resource Sharing

DRM Resource Sharing Requirements

- Partition DRM device resources (e.g. CRTC, connectors, overlay planes) for use from multiple clients (containers)
 - DRM Lease kernel feature (introduced in 4.15)
 - Requires backporting to 4.14 for R-Car in UCB 10.0
- Manage partitioning of resources and distribute them to clients (containers)
 - **DRM Lease Manager**

DRM lease

Terminology

Lessor:

The traditional DRM Master (fd), created by opening `/dev/dri/cardX`, which can create and lease out resources to other DRM Masters

Lessee:

A DRM Master (fd) which controls resources leased from another DRM master. Each Lessee leases resources from a single Lessor.

Lease:

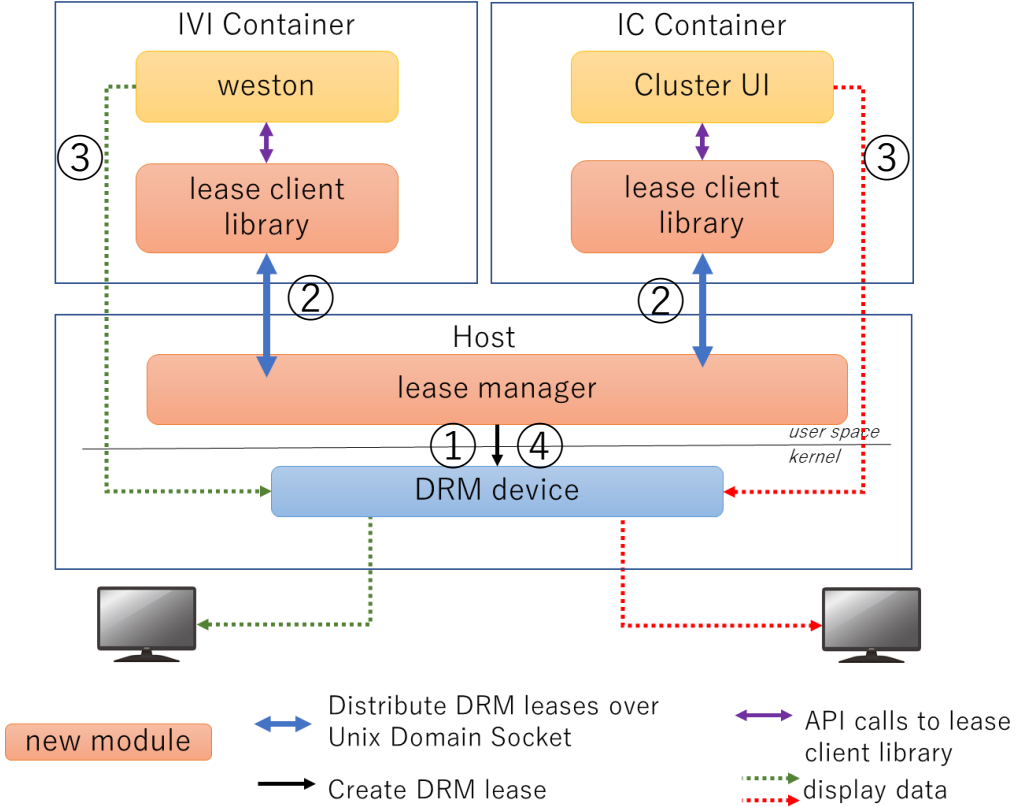
The contract between the Lessor and Lessee which identifies which resources which may be controlled by the Lessee.

The lessor creates a lessee that has DRM Master permissions over the resources defined by the lease.

ref: <https://keithp.com/blogs/DRM-lease/>

DRM Lease Manager Operation

1. DRM Lease Manager(daemon) opens DRM device and creates lessee DRM Masters for each client.
2. Sends lessee file descriptor to clients (via lease client library) when requested
3. Clients render directly to DRM device using lessee
4. DRM Lease Manager revokes lessee when the client is finished with it.



Seamless Transition to Next Client

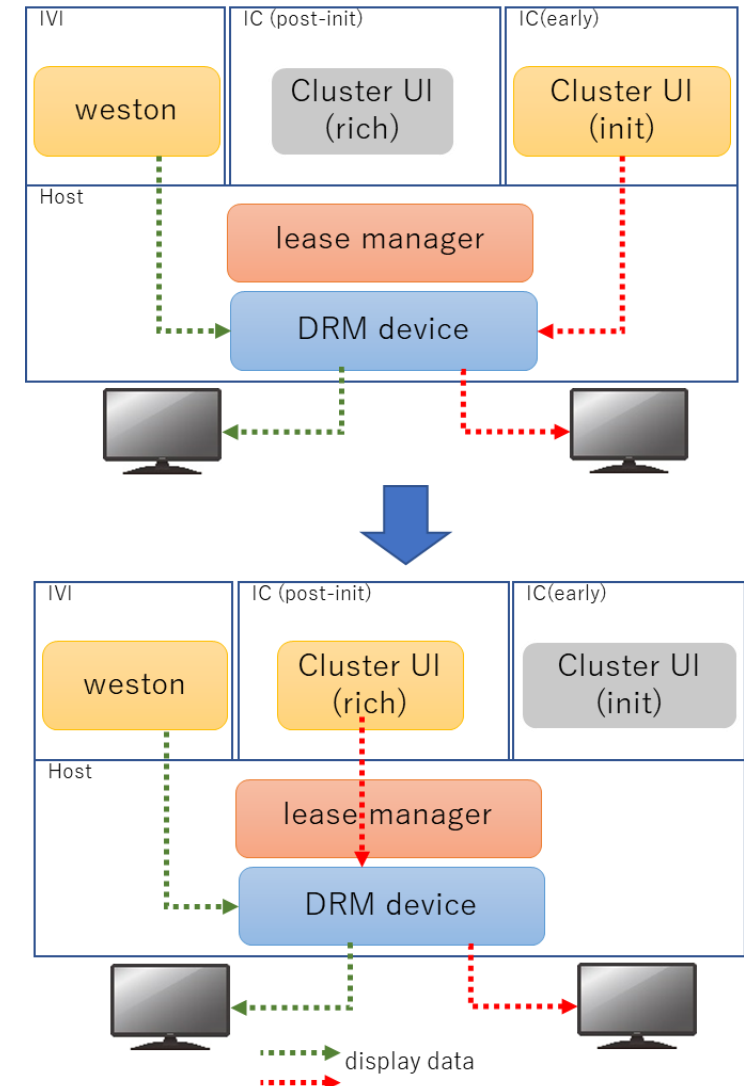
After client releases a lessee, resources are available for next client

May want to make a seamless (non screen blanking) transition to next client

Possible Use case:

Early Cluster UI rendering

- Display Cluster UI early in boot process
- Transition display to rich environment once system is fully loaded



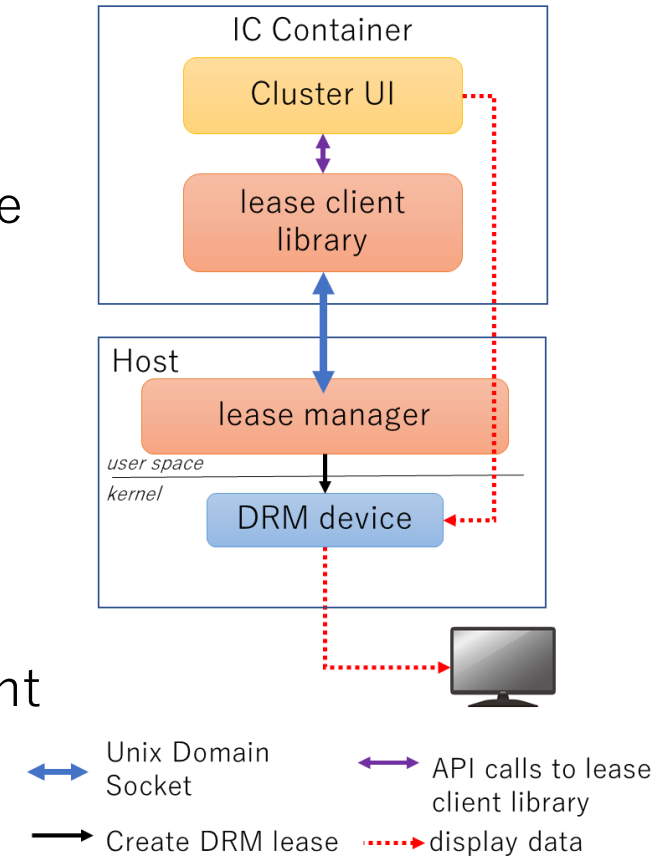
DRM Lease Manager Usage

DRM Resource Partitioning

- Partition of DRM Resources
 - 1 connector, CRTC, and all *CRTC exclusive* planes per lease
 - Planes shared between multiple CRTCs are not included
 - # of leases = MIN (# of connectors, # of CRTCs)
 - On Renesas R-Car M3/H3
 - No planes are shared between CRTCs
 - # of connectors = # of CRTCs
- Could use a more sophisticated partitioning mechanism in future

Distribution of lessee file descriptors

- Create a named Unix Domain Socket for each lease
- Client library connects to DRM Lease Manager and handles lease request and releases
- Client applications just replace calls to `drmOpen()` / `drmClose()` with lease client library APIs
 - Lease Manager Client provides DRM fd to client and keeps connection to Lease Manager open
 - Lease Manager will automatically revoke the lease if the client crashes.



Current Status and Next Steps

Current status

- DRM Lease kernel feature
 - Backported from 4.15 to 4.14
- DRM Lease Manager
 - DRM lease / UDS socket creation
 - Client implementation
 - Verified operation with modetest
 - Retrieve leased resources
 - Display color bars on screen

Next steps

- libweston integration (drm-backend)
 - Add drm-lease option to weson.ini / command line
 - TTY sharing
 - Add an option to bypass TTY init (and VT switch) to allow:
 - multiple Weston + drm-backend to run at the same time
 - Weston + drm-backend in container without host TTY.
- Verify GPU driver operation
- AGL integration
 - recipes, feature addition, upstreaming
- Seamless transition

Schedule

2021/2M

- DRM Lease Manager daemon
 - Lease Manager client library
 - libweston integration
 - GPU rendering
 - AGL integration
-
- 2021/2M~
 - Feedback
 - Seamless Transition