

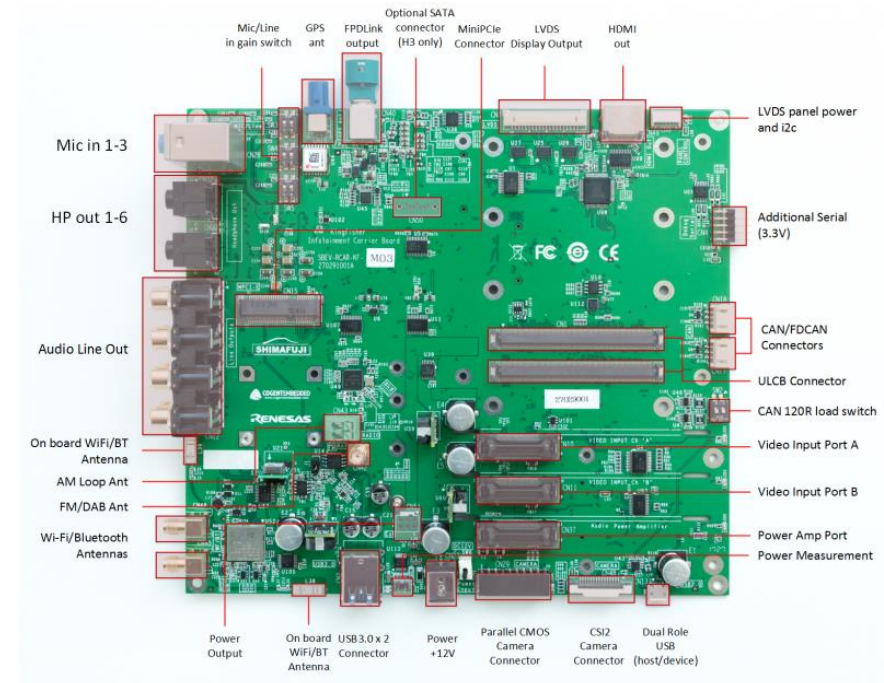
# DRM Lease Manager Status

March 19, 2021

# Target Platform and Use Case

# Target environment

- Software
  - AGL master branch
  - Linux 5.4
- Initial Hardware Target
  - Renesas R-Car SK + Kingfisher extension board
  - 1 HDMI output from R-Car SK
  - 1 HDMI output from Kingfisher

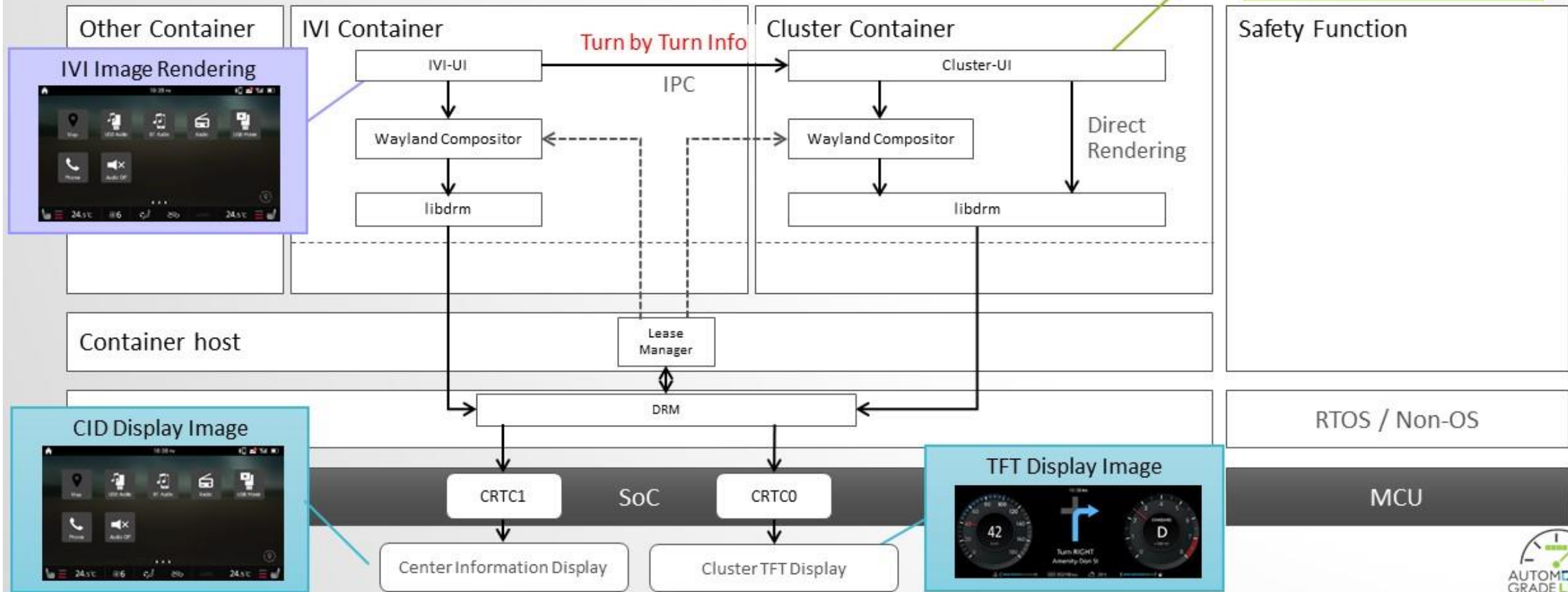


<https://elinux.org/R-Car/Boards/Kingfisher>

# DRM 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 directly 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.



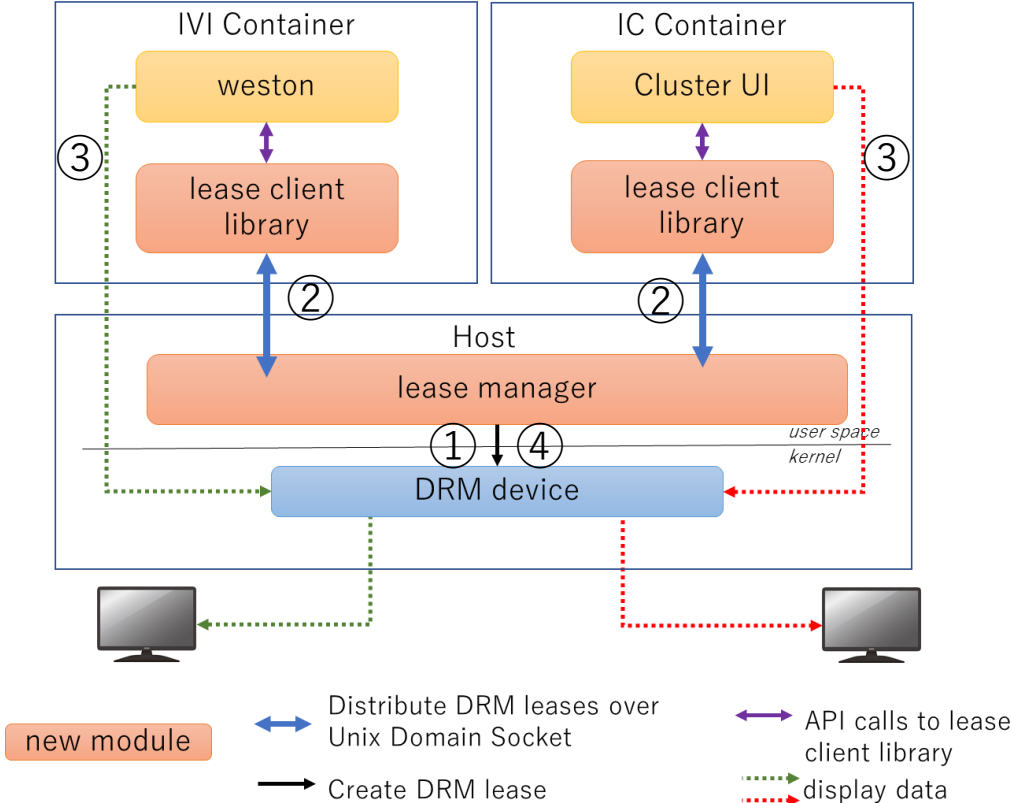
# DRM Resource Sharing

# 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)
- Manage partitioning of resources and distribute them to clients (containers)
  - **DRM Lease Manager**

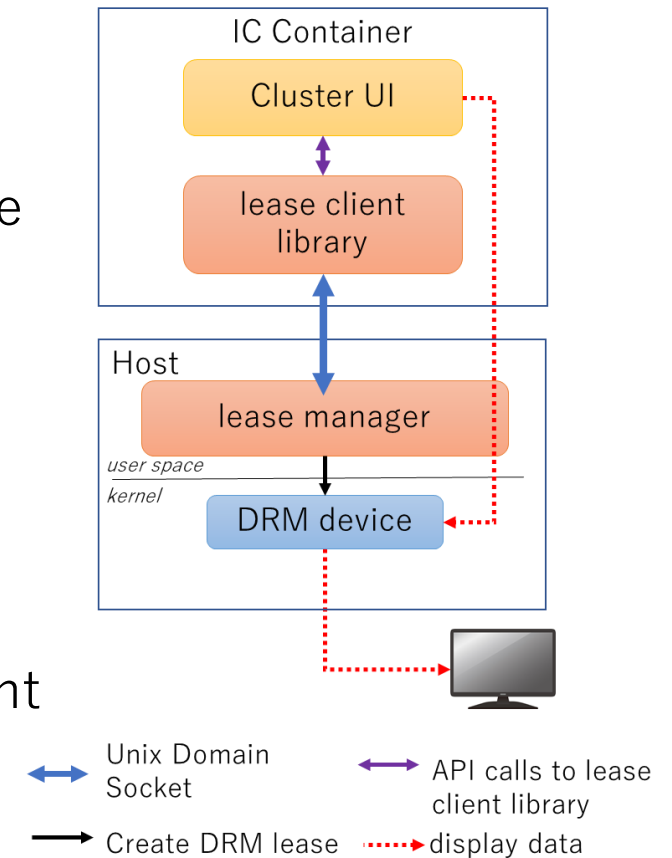
# DRM Lease Manager

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.



# 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 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

# Current status

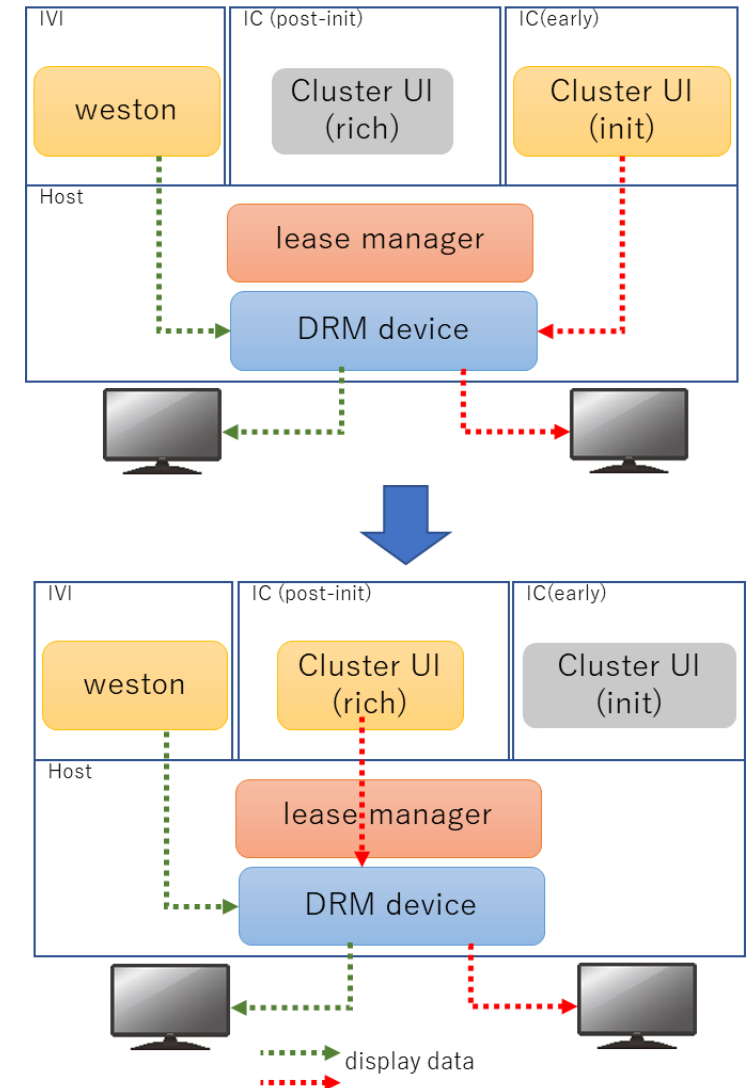
- DRM Lease Manager
  - DRM lease / UDS socket creation
  - Client library implementation
- Weston support
  - weston compositor
    - Retrieve DRM lease from lease manager
  - libweston
    - Get lease fd from compositor and use it in DRM backend
- Sample DRM application
  - DRM lease option added to kmscube test application

The above have been submitted to gerrit (SPEC-3729, SPEC-3730)

In progress

# Transition DRM Lease to New Client

- Seamlessly (non screen blanking) transition the display to a new client
- Use case:
  - **Early Cluster UI rendering**
    - Display Cluster UI early in boot process
    - Transition display to rich environment once system is fully loaded
  - New client will request lease manager to reallocate DRM lease resources to itself
    - Old client loses access to DRM device
- Should have patches ready for submission soon (target: end of March)



Next Steps

# Suggested enhancements

- DRM Lease Manager
  - Add a configuration file to control DRM lease creation
    - eg. user defined lease names
  - CI integration
    - Also testing on other platforms
- AGL Compositor integration
- Weston / libweston upstreaming
  - How to handle drm-lease-manager dependency upstream?
  - Generic multi-head support?

# DRM Authentication

- DRM Leases can't be used for DRM Authentication
  - Current kernel API only creates DRM lease as master
  - No way to get corresponding non-master file description
- Wayland clients therefore need EGL driver that doesn't use DRM\_AUTH
  - E.g. render nodes
- Latest R-Car driver uses render nodes for Wayland clients
  - R-Car display unit (rcar-du) kernel driver updates also needed.
- Render node requirements for other platforms?
  - Need investigation.