Graphics Sharing & Distributed HMI

⭐ *What is this? Please refer to the projects overview page for a quick introduction, then for more details, the Kickoff slides and kick-off recording (see below).*

Next Meeting

We are currently on a weekly schedule, 30 minutes only.

⭐ Next meeting, Thursday, July 2, 2020 at 10:30 AM CE(S)T (30 minutes)

Zoom Meeting Join Link, (register first - you can do that before to prepare) ⭐ Password: gsha

- **Zoom meeting number: https://zoom.us/j/219826332**

Agenda and links

- Android/Linux graphics stack combination (ideas, architecture pictures) Analysis of the fundamental issues to solve: Implementation Needs
- Scoping/plan for Code project proposal (not yet active)
- Reusable meeting scratchpad: https://etherpad.net/p/gsha

- discussion on Android & Wayland
  - one of the major points to solve in the Wayland on Android use-case is the EGL implementation/configuration in the Wayland compositor because this one needs to have a Android backend and needs to have Android EGL! At the same time Wayland compositor needs to receive buffers from the wayland clients which are sending it via wayland protocols. So the Wayland compositor needs a way it import the wayland buffer or at least the dma_buf for rendering to the Android system. If the GPU vendor is providing the Android EGL which also supports the following EGL extensions: EGL_EXT_image_dma_buf_import the problem should be solved (we cannot say it for sure because most probably it is untested use-case for gpu stack)

Presentations & Minutes use link

Backlog

Topics to be (re)scheduled:

- Benchmarks and performance knowledge. (Pool together results from ADIT, Qt, Harman, BMW/Mentor/RAMSES, AllGo, and all...)
- Provide & discuss details on Weston-remote
- Android IVI use cases & Android's SurfaceFlinger and Wayland API's Mapping
  - HAL role and features?
- API Remoting of OpenCL (Artem/EPAM). (This is not for graphics, but it's GPU and it's API Remoting. Might be useful.)
- Coordinate "GPU sharing" topic with the Hypervisor Project
- Deep-dive? Web assembly as graphics transfer technology? (QtWebAssembly) (revisit in February)
- Study the graphics sharing part of SmartDeviceLink protocol and compare (need volunteer)

Requests:

- ⭐ Use-case / requirements workstream. Write down use cases, scenarios and requirements for graphics sharing & distributed HMI.
- More example/evidence of Android-based graphics sharing - (especially distributed network, since Hypervisor-based is common) **(DONE, but feel free to share more)**
- Example/evidence of QNX-based graphics sharing **(DONE, but feel free to share more)**
- Example/evidence of graphics distribution involving other graphics / HMI tools including various "commercial HMI tools" ❌

Previous Events

- **GENIVI 19th All-Member Meeting, May 14-16 2019 in Munich**
- **GENIVI Technical Summit, India** (October 10-11, 2018, see genivievents.com for latest events)
- **RAMSES F2F Workshop** – **Planning and content page**
- **All Member Meeting, Munich, April 2018**
    - Schedule and Presentations
    - Session planning for GSHA at AMM

Definitions

Overall (project scope)

- **Graphics Sharing**
  = 1. Graphics in the form of bitmap, scene graph or drawing commands generated on one ECU, transferred for display from another ECU (or between virtual machine instances)
  2. GPU sharing in a virtualized setup
- **Distributed HMI Compositing**
  = Methods and technologies to turn a multi-ECU system into what appears and acts as a single user-experience.

Results

The 5 Categories of Graphics Sharing technologies

(as developed by the group so far)

- **Display sharing**
  The physical display can be shared across multiple operating systems. HW compositor unit composites final display buffer from HW Layers of each OS. This requires virtualization of the display controller hardware.
    - See Display Sharing tech brief
- **Surface sharing**
  Operating systems exchange graphical (bitmap) content. Then, each OS has full flexibility to use this content. In some cases, the compositor API is made available remotely, e.g. Wayland->Waltham.
    - See Surface Sharing tech brief
    - Sub-category: "Virtual Display" – Full display transfer by encoding as a video stream. – Tech Brief is in development
- **API Remoting**
  Transfer API calls, corresponding to "drawing commands", or other abstract graphics representation from one ECU to another. Commands or scene representation to be executed on the GPU of the receiving ECU.
    - See API Remoting tech brief
- **Shared state, independent rendering**
    - Each system has independent graphics systems and bitmap information. The systems only synchronize their internal state and exchange abstract data. Based on this shared data, each system independently render graphics to make it appear like they are showing the same or related graphics.
        - Example: An appearance of synchronized map rendering could be achieved by drawing maps independently, and exchanging only the GPS position, scale of map, etc.
        - Comment: This differs slightly from API Remoting as follows: API Remoting implies active control from a primary to secondary system, i.e. a command exchange such as "Draw this", whereas Shared state is more akin to "here is the data you need - but you control how/what to draw" (still with the expectation that the appearance will match the "common" HMI).
    - See Shared State tech brief
- **GPU sharing**
  The GPU can be used from multiple operating systems, so it is shared. Concurrent access to the physical GPU has to be controlled by the hypervisor, hardware or other means which are implementation specific.
        - Investigation of this topic is primarily delegated to the Hypervisor Project group

Project Goals

- All project participants gain thorough understanding of available choices
- Produce technology demonstrators, newly created or (if exists already) found and highlighted.
- Publish hard data on learning: Performance, resource needs.
- Seek industry acceptance & alignment among Linux distributions, as well as across operating systems and domains
- Seek alignment on solutions and protocols among proponents of “closed” alternatives – commercial HMI-tools, etc.
- Promote open standards and implementations across industry
- Separately identify and describe Hypervisor-based opportunities, how they differ, characteristics, advantages and disadvantages.
- Summarize and create (implementation) documentation for recommended choices

Milestones

"Complete" Table-based overview
Work in progress. Decide on completion.

Define Graphics Sharing Categories

Publish Tech Brief - Shared State

- April 2018 – **DONE** (see page)

Publish Tech Brief - Surface Sharing

- July 2018 – **DONE** (see page)

Publish Tech Brief - GPU Sharing?

- We will first put it onto the agenda of the HV project to seek experts/volunteers among that group.

Publish Tech Brief - API Remoting (with case study: RAMSES)

- API Remoting working page for detailed definition and tech brief.
- Time plan: October Tech Summit

Publish Tech Brief - Display Sharing

- Time plan: Write tech brief for ~~mid-November~~. Ongoing, ready early February.
- 3 display canvas demo as example? GPU virtualization with HV, and display sharing (HW supported).
- Analysis on separate Wiki Page, tech brief draft work here
- **PUBLISHED** – see page

Publish Tech Brief - Surface Sharing: Virtual Display?

- Do we wish to publish a tech note on the Surface Sharing sub-category "Virtual Display"?
- Technologies known to group:
    - Surface remoting via video streaming.
    - Extending Weston to support virtual remote displays
    - AllGo demo (standard Android APIs + proprietary surface transfer/control protocol)
- **DONE,** should be announced / published

Publish Project Report and/or Whitepaper on all technologies

- Updated time plan: Q1 2019 delay Q2
- Compare technologies - make clear use-case and constraints that lead to decision.
    - Volunteers (writing / copy-editing):
        - Eugen, Harsha, Stephen, Gunnar.
          Qt Company provided short text for Remote Objects
          Others hopefully once document structure is up.

RAMSES Open Source Release

- **DONE** – see GitHub.

Use cases

Here we list some real-world functions (ideally from user perspective), to anchor the technical discussions to.

- Navigation - map display from IVI to cluster
- Entertainment (album art)
- Remote "application" display
- Telephone book (avatars/photos)
- Video content (e.g. in cluster, when not driving)
- Graphical information, triggered by context (people, location, ...)
- Mirroring / Projection mode style CE-device integration and SmartDeviceLink style
- Remote HMI, (car to CE-device)
    - "remote control" for media playback
    - controlling autonomous drive function
- In-car camera to CE-device (e.g. security from remote location)
- "Home screen" in a large combined HMI, the content could come from different sources.
- RSE - Media served from one ECU to simple devices in the back seat.
- Synchronized animation - e.g. content moving from one display to another (and consequently between ECUs)

Implementations and technologies

- Waltham (git repo, docs)
    - Please consider reading (and correcting) Waltham evaluation.

    - first waltham plugins based on Renesas and Gstreamer for buffer transfare should be available by 30 of March
- RAMSES
- QtWayland / Qt-specific ideas?
- AvB / TSN ? – providing bandwidth & timing guarantee (only relevant in combination with other tech)
- SmartDeviceLink (The graphics transfer feature, which is now (re)using a popular name: "Projection Mode")
- CastScreen is an open-source (Android device) screen-cast implementation
- Connection to proprietary solutions - HMI-Tooling like Rightware, Altera, GL Studio, Disti, ... (TODO)

Technology overview & current options organizes and categorizes the options for different scenarios.

Asking for:

- Any more/other CE-device technologies (that are popular, and future-proof)?
- ...more participation from commercial HMI tool vendors.

Evaluation Methods, Polling & Evaluation questions

- Reading documentation & presenting to the group
- Metrics from demonstration implementations (e.g. performance, resource usage)
- Contacting standards organizations behind each choice for presentation/discussion?
- Asking Questions – polling

Polling questions?

- Does your company use <topic> for in-car embedded systems?
  Yes / No / Don't know / (prefer not to answer?)
- Graphics sharing
    - Between an IVI and Cluster ECU?
    - Between virtualized systems running IVI and Cluster operating system instances?
    - Between car and CE-device?
    - Between car and IT (e.g. screen dumps for debugging?)
- Uses today vs has used?

Sub-page links

(all direct children to this page)

- [GSHA] Evaluating & choosing methods
- Analysis of draft tech brief Surface Sharing (temporary page)
- Android's SurfaceFlinger and Wayland API's Mapping
- Android and Wayland Graphical architectures
- API Remoting Tech Brief Work [GSHA]
- Graphics Sharing topics at Tech Summit Bangalore 2018
- GSHA: Display Sharing
- GSHA: Virtual Display
- GSHA at AMM Munich, April 2018
- GSHA Deliverable plan
- GSHA - Presentations & Minutes
- GSHA White Paper
- Planning for RAMSES workshop and SAT F2F
- Qt+RAMSES combination
- RAMSES
- Technology overview & current options
- Waltham evaluation