GSHA: Virtual Display

At a previous meeting we decided to split "Type 2" Display Sharing and eventually considered it to be a subcategory of Surface Sharing.

It describes a kind of "Virtual Display" setup that is frequently mentioned in some programming environments.

Virtual Display

- Full display transfer, often by encoding it as a video stream from one system to another. The transfer itself is basically a surface, but representing an entire screen such that it would normally be sent to a display (see here for comparison to Surface Sharing). The receiving system may however use content as part of a composition.

Virtual Display vs. Surface sharing:

This distinction is not 100% precise, and there are likely systems out there that could be considered somewhat hybrid in nature. Here is how we consider Surface & Display sharing different:

- **Surface Sharing** - **Bitmap transfer of a *partial* HMI** (for example the output of *one application*, or otherwise a part of the final screen content). It is assumed that the receiving side will decide how the surface is used and placed into the main HMI. In other words, there is a "compositor" combining the received surface with other content before it is rendered to the screen.
- **Virtual Display - Bitmap transfer representing the *entire screen area (for a display or function*)**. On the receiving side the content is treated as complete screen content, is not transformed significantly, and is not included as a combination of other graphics. On the sending side, application content is composited towards a display. All APIs are generally transparent, in other words applications need not change their behavior significantly because the display is "virtual". However, note that the virtual display may represent a *layer* in a final composition made up of layering complete screen areas "on top of" each other. For example the virtual display may contain the screen area for navigation turns or current audio playback track/album between the dials of a larger cluster physical display. The sender could composite the various parts of the navigation screen together at the correct resolution and send it. The receiver could receive the sent surface into a layer which is then composited into the cluster display.

It is clear that there may be some other variations when it becomes difficult to fully differentiate between for example Surface Sharing and Virtual Display. In such cases we would recommend calling it Surface Sharing. In the end the result matters more than the definition, of course.

Specific implementations

1. Weston compositor virtual-display and gst-recorder plugin

- Renesas Weston compositor virtual-display and gst-recorder plugin
  - virtual-display allows the creation of a composited display window the correct size which can then be remoted
  - gst-recorder GStreamer plugin in Weston Compositor encodes the virtual display using H.264 and transmits. Receiver uses GStreamer to receive and decode the stream, which can then be composited as required.
  - Pros:
    - No modification required to Weston on receiver side.
    - If layer control is required the decoded stream can be transformed as required on receiver side in gfx framework of your choosing.
    - Video encoding minimises network data and bandwidth
    - Virtual display means display size can be made only as large as needed.
    - Technology has multiple applications, e.g. has been used in Waltham implementation for gfx data transfer.
  - Cons:
    - Delay of about 2-3 vsync to encode, transmit, receive, decode and composite the stream.
    - Large display containing a lot of animated objects will impact network bandwidth
  - R-Car M3 PoC in AGL demonstrator. Open source: links TBA
    - Two M3 ECUs. Navigation and mapview app encodes mapview image using gst-recorder and transfers via ethernet (udp). Receiver mapview image, decodes the data and displays moving map image between dials of cluster display.
  - See the slides presented at the ALS 2018 talk "Remote Access and Output Sharing Between Multiple ECUs for Automotive"

2. Android ScreenCast over network to other OSs

- Renesas ported ScreenCast for virtual display remoting from Android
- A virtual display screen buffer is encoded, transmitted over the network, then decoded by the receiving OS. In this way it uses some similar technology from the Weston remote implementation above.
- Pros:
  - Widely supported transmission standard means wide variety of OSs and applications can be used on the receiving end.
  - If layer control is required the decoded stream can be transformed as required on receiver side in gfx framework of your choosing.
  - Video encoding minimises network data and bandwidth
  - Virtual display means display size can be made only as large as needed.
- Cons:
  - Large display containing a lot of animated objects will impact network bandwidth