TECHNOLOGY BRIEF

APRIL 2018

Category: Vehicle Domain Interaction

Digital Cockpit HMI Distribution Using Shared State, Independent Rendering

Summary

A holistic digital cockpit HMI with seamless user experience across IVI and Instrument Cluster displays can be implemented using different approaches:

- Display Sharing
- GPU Sharing (virtualization)
- Surface Sharing
- API Remoting
- Shared State, Independent Rendering

Using shared state, independent rendering approach a multi-domain system performs cross-domain HMI/services data sharing and local HMI rendering within each domain.

Key Characteristics

Advantages to the Shared State, Independent Rendering approach are:

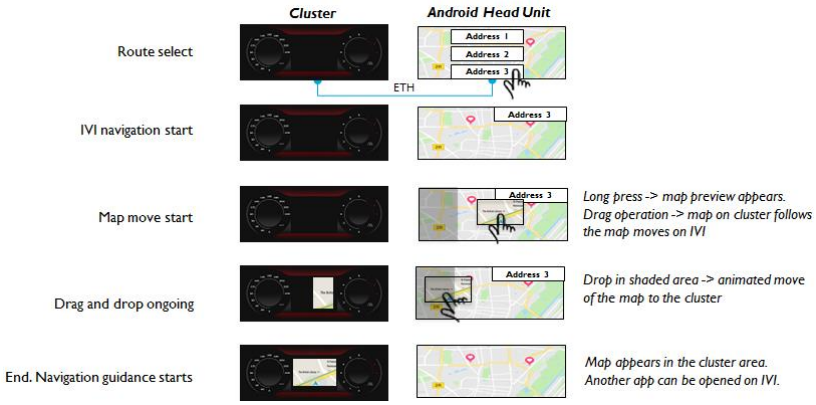
- Low inter-domain data channel bandwidth usage
- Applicability to mid/low performant SoC
- Operating System - agnostic approach

Description

The Automotive industry requires in-car HMI to appear and act as a single user-experience. This requirement must hold true for numerous combinations of the underlying architectures and technologies: single or distributed SoC-based platforms, real-time OS-based Instrument Clusters, GENIVI Linux-based or Android-based infotainment systems, etc.

Shared state independent rendering approach suggests self-contained graphics content rendering for each domain and a cross-domain data sharing service for HMI state and services data synchronization between domains.

The holistic perception of the HMI is achieved by ensuring that graphics data and rules for HMI look & feel are aligned beforehand, on the independent rendering devices. Synchronized software updates should be applied to ensure this relationship is intact if graphics are updated. This approach allows development of deeply-integrated functionally-rich UIs with complex screen animations and high degree of user interactivity.

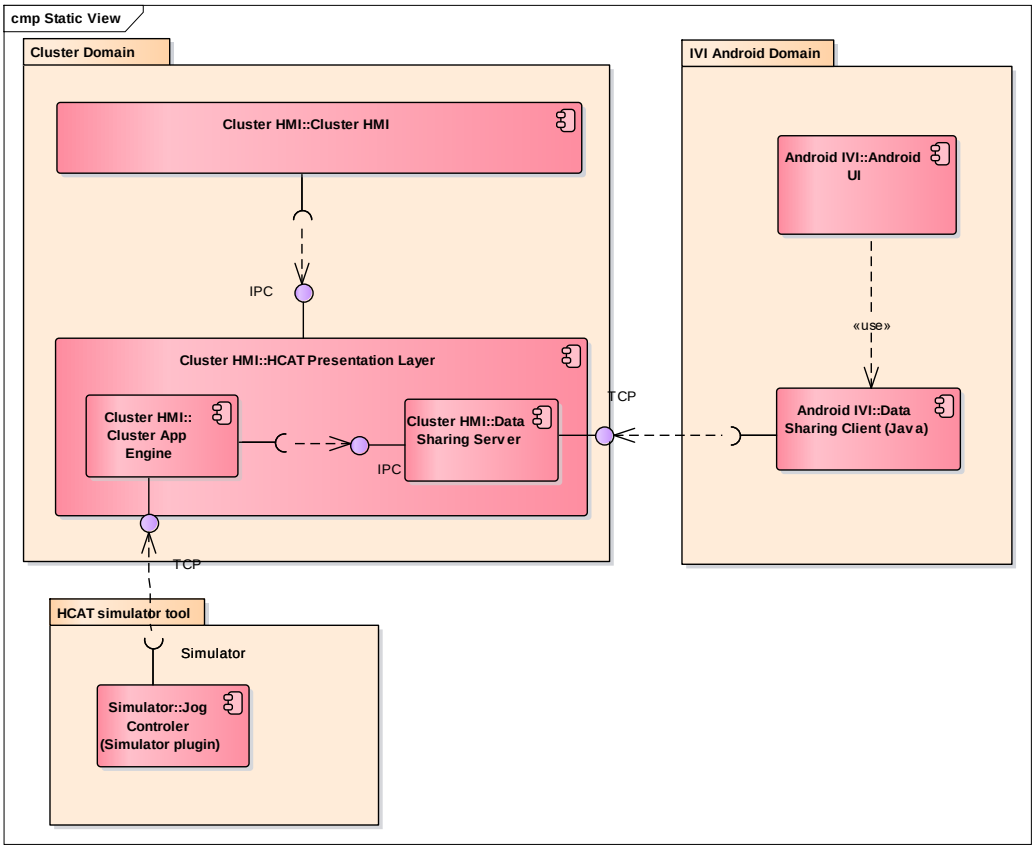


Shared State, Independent Rendering PoC

The approach has been implemented within the Proof of Concept project for the use-cases which show obvious customer significance. A clear case example is navigation application data sharing between IVI and Cluster.

The PoC setup has been built with GENIVI Linux for IC and Android for IVI. The HMI has been implemented using Android native HMI on IVI system, and with Qt/HCAT-based HMI on Linux Cluster.

To prove the shared state independent rendering approach the prototype of the data sharing service has been implemented. This service provides functionality for sharing variable data types/structures between domains by using mechanisms of data registration, subscription and notifications. It includes server part and client libraries for Linux and Android platforms.



Our results prove that with this approach, we are able to develop distributed HMIs and to provide synchronized UX for the following scenarios:

- IVI navigation content propagation to the Cluster display
- IVI and Cluster animation synchronization
- IVI content switching in accordance with Cluster application state

Alternatives & Related Technologies

- Alternatives: Surface sharing, API remoting
- Related: GPU sharing, Display sharing

References & Additional Reading

- GENIVI Domain Interaction Project on Graphics Sharing and Distributed HMI

Authors

Roman Leykin (Roman.Leykin@Harman.com), Harman Connected Services
Sergey Klevitskiy (Sergey.Klevitskiy@Harman.com), Harman Connected Services