What

GENIVI Projects – What

Today, open source software runs in smartphones (Android and iPhones) to mission-critical stock trading systems to the devices that manage communication on the internet. The myth that open source software is created by amateurs has long been dispelled as paid developers from companies like IBM, Microsoft, and Samsung all participate in open source projects.

GENIVI was created to bring open source software into the car, starting with the most complicated car software system, the IVI unit. To support this effort, GENIVI works in already existing open source projects and hosts open source projects where IVI software does not already exist in the open community.

Below is a block diagram of the functional units of a GENIVI IVI software platform. Please click on any of the below units for a brief description of each and an indication of where GENIVI is working on the functionality.

Applications & Human-Machine Interface Frameworks

Scope:

As with smartphones and tablets, consumers are becoming entitled to have their favorite applications while they are mobile. Drivers want the same freedom to use their applications in the car. The IVI unit must have a framework to which a variety of applications can be written. An application framework is NOT intended to replace apps stores, but to provide a simple and consistent method for applications to interact with the IVI unit and its resources.

In addition to applications, a user interface framework must be created to for safe and convenient interaction with the driver. This framework does not prescribe buttons or knobs, but it does provide a consistent messaging channel for driver interactions with the IVI unit.

Projects (no code release identified until October 2013)

- Pop-up Management (Active)
- Translation and Internationalization (Discussing Approaches)
- Generic Speech or Voice Interface (Request Information)
- HTML5 Interface (Active, but Request Information)
- Registry (Discussing Approaches)
- Application Launcher and Life Cycle Manager (Active)
- Workload Manager (Discussing Approaches)

Project status: Active, Discussing Approaches, Request Information

Multimedia, Audio & Graphics

Scope:

The infotainment unit of a car must handle a vast amount of media content in various formats and for various uses. In addition to managing the content, the IVI unit must also display/play the content to the driver and passengers.

The Media & Audio & Graphics functional unit consists of three managers: media management, display management and audio management

Media management consists of:

- Management of external audio/video sources
- Web and radio tuners
- Picture manipulation
- Audio/video recording
- Media metadata tracking and indexing
- Search operations on media database
- Media server performance for streaming and metadata access.

The Display management consists of IVI layer management and interaction with audio/video management.

The Audio Management function handles a very unique set of requirements for prioritizing and managing various audio signals in a car. The IVI unit tends to be one of the primary channels of interaction between the car and the driver. So, audio signals related to various sensors like those associated with rear view alarms must be managed accurately for safety reasons.

Projects

- Layer Manager (Active) IVI Layer Management
- Audio Manager (Active) Audio Manager
- Web and Radio Tuner (Active)

CE Device Connectivity & Personal Information Management

Scope:

The CE Device Connectivity functional unit includes the various techniques of integrating smart devices (smartphones, tablets, etc.) into the in-dash IVI unit. Activities in this functional unit include managing connections with smart devices, retrieving/pushing content stored on the smart device, managing Bluetooth connections and managing near-field communications.

Projects
Bluetooth Audio Streaming (Active)
Bluetooth IOP (Discussing Approaches)
Device Management (Active)
Media Playback (Active)
Smart Device Connectivity (Active)
Personal Information Management (Active)
Near Field Communication (Active)

Navigation & Location Based Services
Scope:
This functional unit provides the typical navigation, traffic, points of interest, and other information to the driver. It interacts with a wide variety of in-car functionality including:

Contacts book
Speech Services
Connected Services
Traffic Data Provider
Sensor Services
Vehicle Data Provider
POI Provider
License Manager
Persistence Manager
Graphics: graphical rendering (for map display, for guidance)
Personalized Location Content
Map Database Provider
HMI (user interactions through buttons, knobs, speech recognition).

Projects
Positioning (Active)
Navigation APIs (Active)
Map Data Access (Active)
Traffic Information (Active)
POI Provider (Active)
LBS APIs Proof-Of-Concept (Active) pushed to open source soon (code)
POI APIs Proof-Of-Concept (Active) pushed to open source soon (code)

Networking & Internet
Scope:
The Networking and Internet functional unit is all about connecting vehicles to IT Infrastructure and to other vehicles by using wireless network connections over:

Mobile networks
WLAN access points.

Also included in this functional unit is the management of voice calls (including telematics), a browser framework for automotive, Ethernet networking in the car, and connection management between various data sources and sinks in the car.

Projects
Connection Management (Active) (upstream: http://connman.net/)
Browser Framework (Active)
Service Frameworks (Request Information)
Download Upload Messaging Manager (Active)
Inter Node Communication - Transport Protocol (Active)
Inter Node Communication - Middleware (Active)
Ethernet-based Audio Video (Active)
Ethernet-based Diagnostics (Request Information)

Projects hosted by the Korea Regional Expert Group
Web Platform - Vehicle Web API (Active)
Car Data Logging (Request Information)

System Infrastructure
Scope:
This unit covers functionality of the automotive electronics control unit (ECU) system necessary to get the system into operational mode and manage the system resources. Much like your computer has “start-up” and “shut-down” activities it must perform, so the ECU is the car’s computer that must do similar activities.
As you can imagine, the scope of this unit is closely tied to the operating system. In this case, the OS is Linux, so the system infrastructure unit includes patches to the Linux kernel to make it better fitted for use in a car.

The functional components in the GENIVI SI unit include:

**Lifecycle**
- System start up and shut down
- Control of system states and modes
- System health monitoring
- Resource control mechanisms
- Persistence (storing settings and system parameters)
- Start up / Shut down specific aspects
- Persistence Manager
- User Management and Personalization

**Software Management**
- Installation and upgrade of the software running inside an IVI Linux system, i.e., firmware and applications.
- Basic Operating System (OS) support
- Interprocess Communication (IPC)
- Interface Description Language.

Within the System Infrastructure activities is a “sub-unit” of functionality called “Automotive”. Within this sub-unit is in-car functionality that is unique to the automobile. An example is the in-car networking functionality such as the CAN-bus. The CAN-bus is a in-car network over which car sensors pass status information to the ECU for non-IVI systems (e.g., safety) within a car. Another function is the diagnostic, log, and trace functionality of a car.

**Projects**
- Persistence Management – Persistence Client Library (Active) pushed to open source Persistence Administrator Service (code)
- Persistence Management – other components (Active) (code)
- User Management – Profile Manager (Active) pushed to open source soon
- Lifecycle Management – Node State Manager (Active) pushed to open source (code) Node State Manager
- Lifecycle Management - Node Startup Controller (Active) pushed to open source (code) Node Startup Controller
- Lifecycle Management – Resource Manager (Discussing Approaches)
- CommonIDL (Active) upstream, developed now at Eclipse Labs
- Common API (Active) pushed to open source soon (code)
- Software Management (Active)
- Automotive Diagnostic, Log & Trace (DLT) (active) pushed to open source Diagnostic Log and Trace (code)
- Virtualization (Request Information)

**Linux Kernel**

**Scope:**

The GENIVI IVI software platform runs in the Linux operating system. GENIVI may from time to time suggest enhancements to Linux functionality to meet certain automotive requirements. For example, in an automotive context, the operating environment must boot and make certain safety-critical functions (e.g. rear view camera display) available to the driver.

**Projects**
- AF-Bus dbus optimization (Active) pushed to open source AF_Bus D-Bus Optimization (code)

**Security, Tools, Utilis**

**Scope:**

GENIVI is embracing the very general concern about security in IT systems and is consequently building a security threat model for IVI systems and identifying countermeasures that will impact all functional units.

The development of an IVI software platform requires also a significant number of supporting tools and utilities. While not part of the software platform itself, these tools enable developers to more productively build and test their software.

Additionally, GENIVI considers all types of projects, some of which may not be tied directly to the IVI architecture. Thus, this functional unit is a “catch-all” for various tools, utilities, and miscellaneous projects which GENIVI deems to be useful to the automotive ecosystem.

**Projects**
- Security - threat model and deployment of countermeasures in all functional areas (Active)
- Linux Container Benchmarking for IVI (Active) – open source project LXC Bench (code)