





Overview of GENIVI projects October, 2015 | GENIVI AMM Korea

Jeremiah C. Foster GENIVI Community Manager Pelagicore AB



Open Source IVI git repo

user-managemenvuser-data-migration-service.git	Data migration service	GENIVI GIL USEI	NO COMMIS
common-api/c-poc.git	Common API C implementation PoC	GENIVI Git User	13 hours ago
lbs/positioning.git	Positioning services	GENIVI Git User	4 days ago
lbs/navigation-application.git	LBS Navigation application	GENIVI Git User	6 days ago
persistence/persistence-administrator.git	Administrator application	GENIVI Git User	9 days ago
lbs/navigation.git	LBS Navigation services	GENIVI Git User	10 days ago
dlt-viewer.git	Diagnostic Log and Trace clien	GENIVI Git User	11 days ago
dlt-daemon.git	Diagnostic Log and Trace daemon	GENIVI Git User	11 days ago
genivi-demo-platform.git	Demonstration Platform and	GENIVI Git User	2 weeks ago
meta-genivi-demo.git	Yocto based GENIVI Demo Platform	GENIVI Git User	2 weeks ago
wayland-ivi-extension.git	Extensions to the Wayland	GENIVI Git User	2 weeks ago
browser-poc.git	Browser proof of concept.	GENIVI Git User	3 weeks ago
gdp-test-suite.git	Test suite for demo platform	GENIVI Git User	5 weeks ago
persistence/persistence-health-monitor.git	Persistence Health Monitor	GENIVI Git User	7 weeks ago
persistence/persistence-client-library.git	Client library for persistent	GENIVI Git User	7 weeks ago
AudioManager.git	Low level audio manager /	GENIVI Git User	8 weeks ago
ivi-speech.git	API for Speech services	GENIVI Git User	2 months ago



Release Cadence





Project pages



GENIVI Open Source Projects

Overview | Why | What | How | Process | Projects | Wiki

Admin

Projects

- · AF BUS D-Bus Optimization
- Audio Manager
- · Browser Proof-of-Concept
- ▼ Diagnostic Log and Trace
 - DLT-Transport
- Franca Validator
- · GENIVI Demo Platform
- IPC CommonAPI C++
- IVI Layer Management
- IVI Navigation
- IVI Radio
- LXCBench
- ▼ Lifecycle Management
 - Node Health Monitor
 - Node Startup Controller
 - Node State Manager
- Media Manager
- Persistence Management

GENIVI Open Source Project

GENIVI was launched with a mission of driving the broad adoption of an In-Vehicle Infotainment (IVI) opensource development platform. (for more on the IVI platform, see What). The code behind this platform comes from three main sources:

- Upstream projects where the required functionality is already built (e.g., kernel.org)
- Upstream projects where a good starting point for required functionality exists upon which GENIVI can build additional automotive functionality (e.g., connman.net)
- GENIVI member projects where IVI code is developed because nothing is available in the open source community as a good starting point.

GENIVI has moved these member projects into the open source and invites all interested parties to participate in building the code (for more information, see Why). GENIVI hosted projects facilitate the development of required IVI functionality in an open and collaborative manner, consistent with the best practices of other FOSS projects (for more information, see How).



Common API C

Project Name	Common API C
Description:	Common API C is a C implementation of Common API. The purpose of the project is to enable programs written in C to work with interfaces defined in Franca IDL.
Source code:	GENIVI Git repository
Issue tracker:	JIRA Stats
Documentation:	



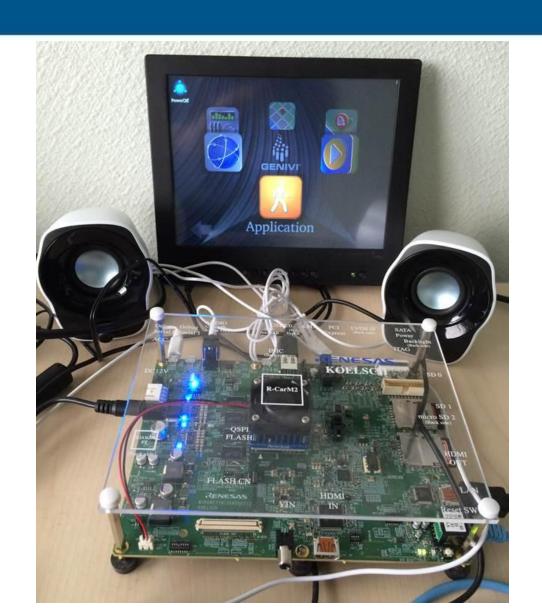
Common API C++

Project Name	Common API C++
Description:	IPC CommonAPI C++ is a C++ based abstraction API for communication stacks, which enables applications to use different communication middleware - so called language bindings - as backend without any changes to the application code.
Source code:	GENIVI Git repository [runtime]
Issue tracker:	Bugzilla
Documentation:	Doxygen code documentation



GENIVI Demo Platform

- → Holds nearly all of the GENIVI components including dependencies and subsystems
- → Evolving into a complete SDK or ADK
- → A good starting point for the latest code
- → Supported by multiple hardware vendors



Getting code

\$ git clone http://git.projects.genivi.org/dlt-daemon.git

• • •

\$ git commit -asm "Fixed magic code"

→ Create a patch



Contributing to Audio Manager

- == Working on the code & contribution
- .First get the code from the git: git clone
- .Get an overview of all branches: git branch
- .Switch to the branch you want to work on (see versioning scheme, the master is the feature branch) and verify that it has switched (* changed) git checkout <your branch> git branch
- .Best practice is to create a local branch based on the current branch: git branch working_branch

Start working, best practice is to commit smaller, compilable peaced during the work that makes it easier to handle later on.

If you want to commit you changes, send them to the audiomanager-dev list, you can create a patch like this: git format-patch working branch <your branch>

This creates a set of patches that are published via the mailing list. The patches will be discussed and then merged & uploaded on the git by the maintainer.

Patches can be accepted either under GENIVI Cla or MPL 2.0 (see section License). Please be sure that the signed-off-by is set correctly. For more, check out http://gerrit.googlecode.com/svn/documentation/2.0/user-signedoffby.html



Communication

- → Important to remember that GENIVI software developers are distributed throughout the world, North America, Europe, Asia
- Requires a distributed communication channel that is asynchronous, that channel is email quite often
- The mailing list serves as a database of searchable discussion that aggregates by subject matter
- Use the mailing lists!
- → It is in the open, so mailing lists are public
- → English is not everyone's first language and that is totally fine



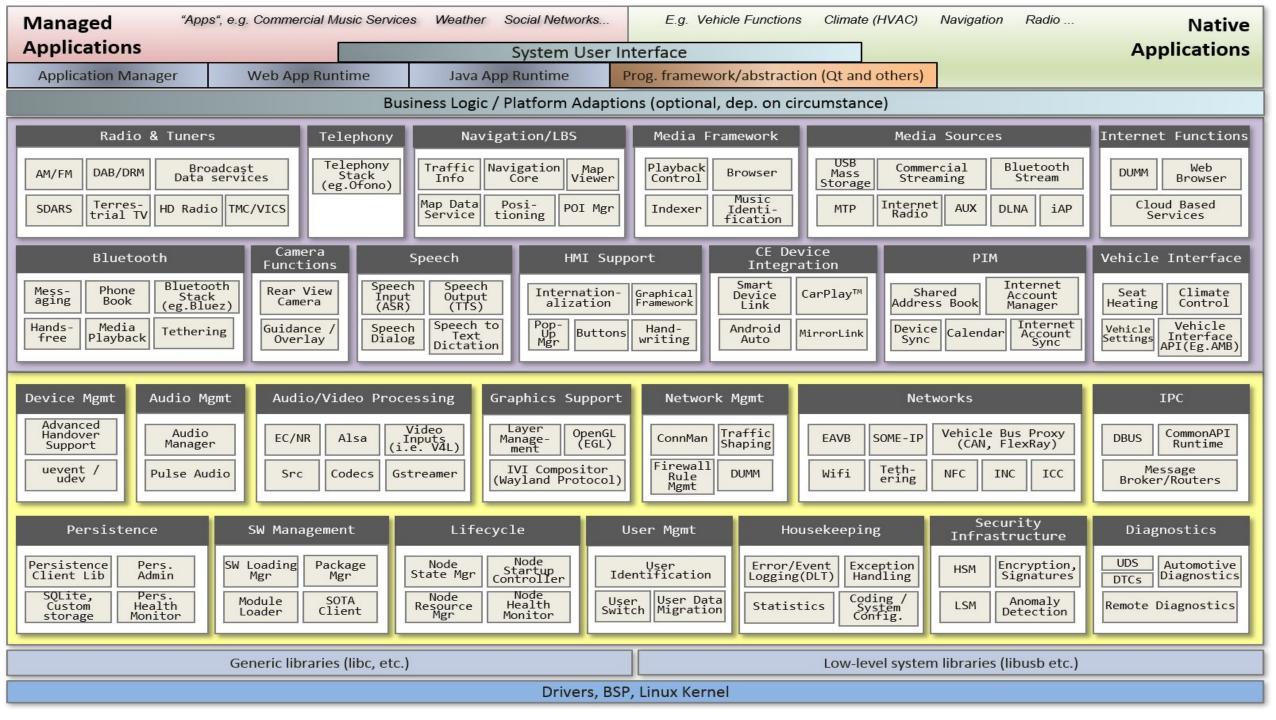
Tracking bugs

- → GENIVI has a public bug tracker which holds bugs for GENIVI components as well as for the GENIVI Demo platform
- → Easy to sign up to the bug tracker and file bugs, add patches, triage bugs
- → Sometimes it can be good to discuss the issue on the mailing list first to see if it is a known issue or a local issue



Toolbox

- → Public ssh key from the machine you develop on
- → Subscription to (at least) genivi-projects
- → git (there are numerous GUI git clients)
- → Login to bugzilla, wiki
- non-html mail client (nice to have)





Links

GENIVI git repository

http://git.projects.genivi.org

GENIVI wiki

http://wiki.projects.genivi.org

GENIVI project pages

http://wiki.projects.genivi.org

GENIVI bugzilla

http://bugs.genivi.org

Internal OSS launch dashboard

https://collab.genivi.org/wiki/display/genivi/Open+Source+Project+Launch+Dashboard

OpenHub

https://www.openhub.net/orgs/genivi

Checklist for pushing code into the open

https://collab.genivi.org/wiki/display/genivi/Checklist+for+pushing+code+into+the+open



Questions

Q&A