



GENIVI[®]

Application Framework APIs in action - a case study

May 11, 2017 | GENIVI AMM, Birmingham, UK

**Ekaterina
Gerasimova**

QA lead, Collabora

**Andre
Magalhães**

Developer, Collabora

What is Apertis?

Apertis is a versatile open source infrastructure tailored to the automotive needs and fit for a wide variety of electronic devices. Security and modularity are two of its primary strengths. Apertis provides a feature-rich framework for add-on software and resilient upgrade capabilities. Beyond an operating system, it offers new APIs, tools and cloud services.



What does the Application Framework do?

- Application bundle structure
- Application installation (installation, removal, rollback, upgrade)
- Application and document launching
- Application lifecycle management
- Resource control
- Data management (per-user application data, private data, per-application data, *etc.*)
- Content and data sharing
- Application confidentiality, sandboxing and security
- Download management (system-wise)



GENIVI Application Framework efforts, what we did so far

- Created a concept/design document for GENIVI adoption
- Extracted and prioritized requirements
- Mapped requirements to functional areas (App Manager, Data Sharing, *etc.*)
- ...and to Apertis components
- Defined required components
- Investigated requirements covered by other GENIVI working groups to avoid duplication
- Converting Apertis APIs to Franca to align with GENIVI standards



GENIVI Application Framework efforts, what's next?

- Finalize Apertis APIs review for GENIVI reuse
- Finish updating APIs for Franca
- Implement/integrate required components



How the Application Framework is used in Apertis

- The Application Framework implementation is split in two main components in Apertis
- Helper APIs are provided for application development
- Also provides support for existing open-source libraries and components, such as BlueZ, GStreamer, *etc.*
- Integrates with the web runtime that can be used to develop apps using HTML/JS and allow access to all supported platform libraries (*via* GObject-Introspection)
- Integrates with the ade (Apertis Development Environment) commandline tool to configure, debug, install applications, *etc.*
- ade integration in the Eclipse IDE



Tools and processes in Apertis

- Feature development and bugs are managed through Phabricator
- Communication through mailing lists
- Code, design and documentation is under version control with reviews happening in Phabricator
- Automated image building and testing



Continuous integration and testing in Apertis

- Jenkins orchestrates building and testing of images
- All patches are taken from Phabricator and built automatically before they are reviewed
- OBS (Open Build Service) produces daily images for all supported hardware
- Images are passed to LAVA to run relevant tests



SDK for application development

Apertis comes with a complete SDK for app development:

- Can be run in a VM for quick setup
- Full desktop environment (XFCE)
- Offline documentation
- IDE (Eclipse) with convenience plugins
- Terminal available if preferred
- Simulator for testing apps in the Apertis HMI prior to testing on HW



SDK for Apertis development

- The SDK also comes with all the tools and documentation needed to develop Apertis itself!
- You can SSH into the VM or HW for development if you prefer to use your own setup
- Cross compilation tools are included in the SDK



Future plans for the SDK

Improvements to the SDK are a key priority for 2017.

Planned features include:

- Support for more languages
- ARM64 cross compiler
- Support for ARM64 in the IDE
- More documentation for building, debugging and cross compiling applications *via* the IDE and the ade commandline tool
- Deployment to target from IDE for testing
- More sample applications and modernization work on existing ones



Future plans

- Decouple frameworks from base platform
 - Fully enable Flatpak bundles
 - Minimal core
 - Containerized frameworks (Qt/Electron/GTK+/custom)
 - Update frameworks in the same way as app-bundles
- Multiple frameworks (*e.g.*, Qt)
- Optimise images for small footprint
- Electron-based (Chromium + NodeJS) web runtime



Thank you!

Visit Apertis at <https://apertis.org> and
<https://developer.apertis.org/>

This work is licensed under a Creative Commons Attribution-Share Alike 4.0 (CC BY-SA 4.0)

