Editor: Jan Welvaarts, Lloyds
Baseliyos Jacob, DB Netz AG

Requirements on openETCS API

Puropose of the document:

This document should reflect the openETCS API requirements of the User and Manufacturer within the openETCS Consortium. The document will be complementary to the openETCS API requirements of WP 2.

1. Indroction

Safety assurance and certification are amongst the most expensive and time consuming tasks in development of safety-critical embedded systems. The increasing complexity and size of this kind of systems combined with the growing market demands requires the industry to implement coherent reuse strategy. A major problem arises as typically a safety-critical product and accompanying safety evidence is monolithic, based on the whole product, and evolutions to the product become costly and time consuming because they entail regeneration the entire evidence set. Another key difficulty appear when trying to reuse products from one application domain in another, because they are constrained by different standards and the full safety assurance certification process is applied as for a new product, thus reducing the return on investment of such reuse decision. Novel fields of applications within General Signaling Controll-Command systems have a safety-critical character and require the development of suitable electronic control systems. From a set of generic components, a fault-tolerant computing core network is realized in a way that--in combination with surrounding core-external redundant sensor and actuator structures--the requirements of a given application regarding safety and functionality are fulfilled. The Figure 1 presents the openETCS API and components with their interplay and explains the architecture of the management layer which organizes data flow within the core network. This management layer is strictly divided into a generic part, which is identical for all platform-based realizations, and into a system-specific part, which depends on the sensor and actuator data flow.

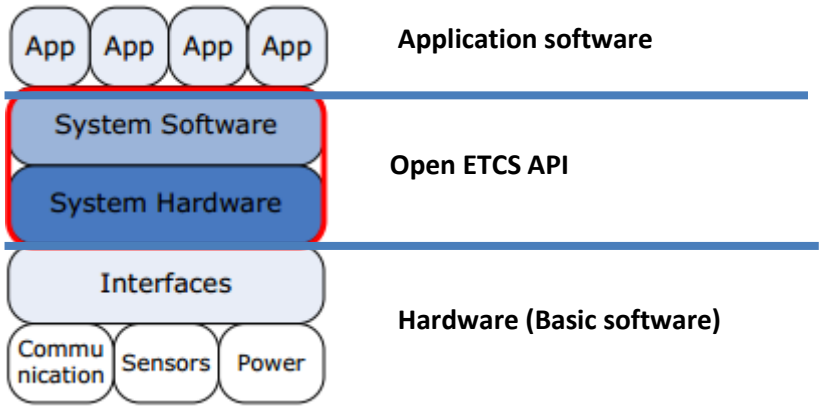
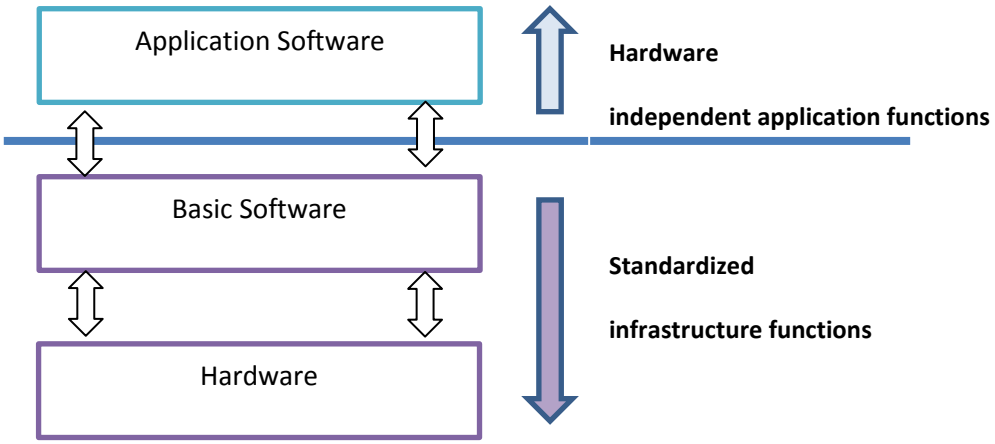


Figure 1: open ETCS API

Mastering complexity of Electric/Electronics (E/E) architectures is one of the challenges of the signaling industry. In order to manage growing system complexity and increasing number of dependencies while keeping the costs feasible, the basic software and the interfaces to applications and bus-systems have to be standardized in a future proof way. In the content of this should be realized within an open generic platform. The objective of this global cooperation was to establish an open industry standard for the CCS signaling architecture for suppliers and manufacturers according to the motto: “Cooperate on standards, compete on implementation”. Accordingly open generic platform standard should comprises a set of specifications describing software architecture components and defining their interfaces as well as the definition of a standardized development methodology. The open generic platform layered software architecture enables the application of independent software components. It aims to increase the reuse of software components, in particular between different vehicle platforms, and between subsystem supplier and Industry suppliers. It enables the scalability of embedded CCS software to different vehicle and platform variants, the transferability of functions throughout the vehicle network, and the integration of functional modules from multiple suppliers. Therefore, the open generic platform standard defines an architecture that separates application software from infrastructure related basic software.



Cycle time/service and calling sequences is a very crucial definition.
Process and methods of service and calling sequences can be different, but must avoid any time violation. **(e.g. see Alstom openETCS API Document)**

2. Tendering requirements

Including the standardized API in the requirements for on-board ETCS equipment, is a step on the migration from purchasing “black box” ETCS on-board systems to fully “open source”. The identified migration steps are:

1. “Black box”: an integrated proprietary on board ETCS system including subsystems like DMI, TIU, JRU,…..BTM, and radio.
2. “White box”: a proprietary on board ETCS system with standardized interfaces between the EVC (European vital computer executing the ETCS software) and the different subsystems.
3. Specifying the API for the EVC: Identical to the “White box” approach plus in addition an open (certified) interface between the platform (hardware plus basic software) and the ETCS

application software.
This step allows a later migration to new (possibly open source) software, without changing the hardware and the installation of the system.

4. Including the rights on the ETCS application software and the complete development dossier, including the tooling used for the software development.
In combination with a standardized API this will allow software maintenance and upgrades to be purchased in competition.
5. Requiring the use of standardized (open source) application software.
Standardization of the software will limit future upgrade costs as only one version has to be maintained.

In parallel with the last three steps, additional requirements concerning the IPR of the hardware and basic software can be included. This is however outside the scope of the ITEA2 openETCS project.

Figure 1. Migration from “black box” on board ETCS to fully open source “openETCS”.

Currently most tenders combine aspects of step 1 and step 2. Only DB also included aspects of step 4.

To create a level playing field for all ETCS on-board suppliers the standardized API shall meet the following requirements. The standardized API,

- Shall be fit for the use in tenders for on-board ETCS equipment
- Shall (as much as possible equally) enable all suppliers, who are contributing to the API development within the openETCS project, to bid with their current software and hardware.
- Shall be independent from the programming language.
- Shall not enforce supplier specific hardware solutions.
- Shall respect the independence between different subsystems (BTM, LTM, Radio, STM’s, DMI, TIU/BIU, JRU/DRU), i.e. no direct communication via the basic SW between subsystems.

- The API description shall be open under the EUPL.

3. Long term suitability

To enable the use of different generations of hardware with the same software, the API shall have a long life cycle compared to the hardware and application software life cycle. The standardized API shall therefore meet the following requirements. The standardized API,

- Shall not be touched by future hardware developments (parameters for the length of the calculation cycle, etc.)
- Shall be independent from interface specifications to subsystems, i.e. have a sufficient (70%?) spare capacity for future extensions of the communication.
- Shall support higher performance (response time, start-up time etc.) when required by the users and/or when enabled by better performing hardware.
- Shall not be limiting for functional changes.
- Shall be described in a formal way by using open source tools (e.g. SysML, UML)

4. General requirements

Apart from requirements to create a level playing field and to guarantee long term suitability general requirements are needed to provide safe and efficient platforms for the (open) ETCS application software.

The standardized API shall therefore meet the following requirements. The standardized API,

- Shall enforce a deterministic execution cycle.
- Shall comply with SIL4 requirements, i.e. be certified (assigning safety related functions including failure rates, clearly to BSW or ASW) to enable separate certification of hardware + basic software and application software.
- Shall be unambiguous, i.e. whenever application software and hardware platform (plus basic software) comply, they are guaranteed to fit and function correct.
- Shall support open interfaces to all subsystems.
- Shall enable the realisation of the required performance of ETCS and the STM's.

5. Manufacturer requirements

See Alstom openETCS API Document.

!!! Need Feedback from the industry partner!!