

(ITEA 3 \_ 15017)

open standard APplication Platform

for carS and TrAnsportation vehiCLEs

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Strategic Standardisation Plan

|  |  |
| --- | --- |
| Document Type: Plan | Classification: Public |
| Document Version: 1.0 | Contract Start Date: 01.10.2016 |
| Document Preparation Date: 03.08.2018 | Duration: 31.12.2019 |

Contents

[**Summary** 3](#_Toc535440775)

[**Introduction to Open Source Software (OSS)** 4](#_Toc535440776)

[**How to manage Common Pool Resources?** 4](#_Toc535440777)

[**What is Open Source Software?** 4](#_Toc535440778)

[**Why do we need Open Source?** 5](#_Toc535440779)

[**Why do we need Open Source Communities?** 6](#_Toc535440780)

[**The Eclipse Foundation** 7](#_Toc535440781)

[**The Eclipse Developer Community** 8](#_Toc535440782)

[**Eclipse Working Groups** 8](#_Toc535440783)

[**The Eclipse IoT Working Group?** 10](#_Toc535440784)

**Summary**

APPSTACLE will publish a significant part of its results, the APPSTACLE Open Platform as Open Source Software under the Eclipse Public License, in the IoT Working Group of the Eclipse Foundation dedicated to open source solutions for IoT. Such open source project follow a precise process that ensures the application of best practices including Intellectual Property analysis for the code and dependencies, configuration management, continuous integration and release management.

Note that these activities are not pure dissemination activities but rather technical activities that are necessary to ensure the success of the APPSTACLE Open Platform as an open source platform.

The consortium agreed to supports the publication of the mains results of the APPSTACLE  project as an open source platform hosted by Eclipse to support APPSTACLE approach. This task will be responsible to guarantee the awareness of the APPSTACLE project as an IoT

project, its acceptance and incubation as a project hosted by the Eclipse Foundation. This task actively coordinates and contributes to the creation and development of the open source community.

In June 2018, GitHub reported having more than 28 million users and more than 85 million repositories (https://github.com/about), making it the largest host of source code in the world.

Based on such huge numbers, how can a project that open source code on GitHub expect to (1) reach their targets, (2) get the desired attraction and (3) build a community that will contribute to the sustainability of the code. There is almost no chance unless you are named Google, Microsoft or Facebook.

For this reason, and also to collaborate with the Eclipse IoT Working Group, the APPSTACLE project consortium decided to work closely with the Eclipse Community to benefit from their developer network and from their experience in building sustainable communities.

This document explains the benefits in joining an open source community like the Eclipse Foundation. After a brief reminder of what is open source is and why an open source community is important to a project like APPSTACLE, we explain what are the best practices of open source communities (governance, IP management, collaborative infrastructure, and recommended development process) and why they contribute to the sustainability of the project. Finally, we describe how the APPSTACLE project decided to create the Eclipse KUKSA

Project to host the APPSTACLE Open Platform. We then describe the next steps necessary to create a sustainable open source project.

**Introduction to Open Source Software (OSS)**

Deciding to contribute to the open source, as a consumer or a producer of open source components, or both, is a volunteer act to manage **in common** some software code and all the resources attached to this software.

Because the initial investment is significant, it is important to feel guided.

**How to manage Common Pool Resources?**

Elinor Ostrom **[1][2]**, Nobel Prize of Economy in 2009, designed 8 principles to manage stable Common Pool Resources (CPR):

1. **Clearly defined boundaries** (clear definition of the contents of the common pool resource and effective exclusion of external un-entitled parties);
2. The appropriation and provision of common resources that are **adapted to local conditions**;
3. **Collective-choice** arrangements that allow most resource appropriators to participate in the decision-making process;
4. **Effective monitoring** by monitors who are part of or accountable to the appropriators;
5. A scale of **graduated sanctions** for resource appropriators who violate community rules;
6. Mechanisms of **conflict resolution** that are cheap and of easy access;
7. **Self-determination** of the community recognized by higher-level authorities; and
8. In the case of larger common-pool resources, organization in the form of **multiple layers of nested enterprises**, with small local CPRs at the base level.

**What is Open Source Software?**

The Open Source Initiative **[3]** provides a very good definition of Open Source Software (OSS) and defines it in *10 commandments* **[4]**:

1. **Free redistribution**: The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.
2. **Include source code**: The program must include source code, and must allow distribution in source code as well as in compiled form.
3. **Modifications and derived works**: The license must allow modifications and derived works, and must allow to distribute them under the same terms as the license of the original software.
4. **Integrity of author’s source code**: The license may restrict source-code from being distributed in modified form *only* if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.
5. **No discrimination against person and groups**: The license must not discriminate against any person or group of persons.
6. **No discrimination against fields of endeavor**: The license must not restrict anyone from making use of the program in a specific field of endeavour.
7. **Distribution of license**: The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.
8. **License not specific to a product**: The rights attached to the program must not depend on the program's being part of a particular software distribution.
9. **License not restricting other software**: The license must not place restrictions on other software that is distributed along with the licensed software.
10. **License technology neutral**: No provision of the license may be predicated on any individual technology or style of interface.

**Why do we need Open Source?**

From Small and Medium-sized Enterprises to large organizations, lots of companies have adopted and contributed for decades to one or more open source communities like the Apache Software Foundation, the Eclipse Foundation, the Linux Foundation or the OpenStack Foundation. This choice has nothing to do with altruism: it is a business strategy.

Actually, organizations adopt open source software for many reasons, including:

* **Maturity of the model:** Even if it is not systematic, there are numerous examples of projects and products based on Open Source Software (OSS) that are more reliable and sustainable than proprietary solutions. Here are a few of them:
	+ The Linux operating system, which is now widely adopted by major private or public organizations.
	+ Apache HTTP Server, certainly the most used HTTP server. It played a key role in the growth of the World Wide Web.
	+ The Mozilla web browser called Firefox, which in 2009 was the most popular web browser with 32% of the market. In 2016, between 9% and 16% of individuals use Firefox as their desktop browser, making it the second most popular web browser (the first one is Google Chrome).
	+ In 2001, IBM donated its Java IDE to the open source community, now known as the Eclipse IDE. After 15 years, this platform supports one of the largest development tool portfolios. Not only Rational, the IBM brand, has built all its desktop workbenches on top of Eclipse, but several SMEs have based their own development tool on top of the Eclipse Rich Client Platform, the minimal set of Eclipse plug-ins needed to build a rich client application.
* **Cost of acquisition:**  Adopters of OSS obtain a financial gain for each stage of a project. For example:
	+ **Free**: it is free to download and use.
	+ **Try before buy**: because it is free, companies can try different OSS solutions before making the decision to invest time or resources in a specific one.
	+ **Hiring is easier**: because OSS is free, many developers use it and become proficient with the software early on in their career or during their studies. This makes it easier and less expensive to find good developers that have experience with the open source technologies they have adopted for their project.
	+ **Training:** because the software is freely accessible and because there is a network of skilled experts able to give trainings, it is easier to train a team with the assets produced by OSS community of developers.
	+ **Customizability**: open source software can be tweaked to suit various needs. Since the code is open, it is simply a matter of modifying it to add the functionality needed by the project.
	+ **Time to Market is shorter**: products do not have to be built from scratch. Companies can rely on sustainable OSS and build their solution on top of it.
	+ **Lower total cost of ownership:** companies can rely on OSS community for maintenance and, by joining the community, they mutualize maintenance costs.
* **No Vendor Lock-In**: Organizations do not depend on the status of the subcontractor, who originally built the software. In OSS, if a contributor ceases working on a project for any particular reason, the source code stays accessible and someone else can take over the work.
* **Quality of the code:** OSS gets closer to what users want because those users can have a hand in improving it.
* **Auditability**: Some users consider OSS more secure and stable than proprietary software, mainly because they can check the source code, identify and fix bugs. The efforts are mutualized with the other community members, which results in better and stable source code.

**Why do we need Open Source Communities?**

An OSS community is the keystone for the sustainability of our project. If we are not able to attract and convince people that our code is worth spending time and resources on testing it, providing feedback, providing patches, and contributing in general, then all the intrinsic value of OSS is lost.

In other words, without Maturity, Quality, Cost of Acquisition and Control, the Sustainability of our code is nearly impossible. And vice-versa, Sustainability is a great indicator demonstrating the Maturity, the Quality and the Control of the code.

Therefore, an OSS community has four main roles:

* **Sustainability**: This ensures the sustainability of the code. If the initial committers leave the project, the code will continue to be maintained by the community.
* **Mutualization**: By building a community around your project, we are able to mutualize our effort and resources. The community, including few of our resources, will maintain the core of the project while the rest of our team will work on our product, our added value, to make the difference with our competitors.



A business friendly ecosystem based on extensible platforms

* **Standardization**: Open Source has proved to be a very good vector of standardization. Here is an outstanding example: In 2011, after 15 year of internal usage, IBM open sourced its Messaging Client for Embedded Devices: MQTT. In less than 5 years, this protocol was so widely adopted that in January 2016 it has become an ISO/IEC Standard.
* **Metrics**: Last, but not least, a good way to know if the OSS we want to use is sustainable and viable is to check the activity of its community: number of committers, number of commits, regularity of the release, and the quality and quantity of assets built around the OSS is a great indicator. In other words, the community is an excellent evaluation metric for a project.

This is the ‘snowball’ effect: a project attracts early adopters with the quality of code, the initial assets attached to the code like the Getting Started guide, documentation, scientific and technical papers, first releases, well defined code infrastructure (bug tracking system, continuous testing, continuous integration, etc.), and the interest of the adopters will do the rest. For this reason, it is very important to provide specific support to APPSTACLE early adopters and the project’s community as it grows.

**The Eclipse Foundation**

The Eclipse community gathers individuals and organizations that wish to collaborate on business-friendly open source software. Its projects are focused on building an open development platform comprised of extensible frameworks, tools and runtimes for building, deploying and managing software across the lifecycle. The Eclipse Foundation is a not-for-profit, member-supported corporation that hosts the Eclipse projects and helps cultivate both an open source community and an ecosystem of complementary products and services.

The Eclipse Project was originally created by IBM, in November 2001, and supported by a consortium of software vendors. Industry leaders like Borland, IBM, MERANT, QNX Software Systems, Red Hat, SuSE, TogetherSoft, and Webgain formed the initial eclipse.org Board of Stewards. By the end of 2003, this initial consortium had grown to over 80 members. Today there are over 250 Eclipse members.

Originally, a consortium that formed when IBM released the Eclipse Platform into open source, the Eclipse Project became an independent body that will drive the platform’s evolution to benefit the providers of software development offerings and end-users. All technology and source code provided to and developed by this fast-growing community is made available royalty-free via the Eclipse Public License (EPL).

The Eclipse Foundation, Inc. was created in January 2004 as an independent not-for-profit corporation to act as the steward of the Eclipse community. It was created to allow a vendor neutral, open and transparent community to be established around the Eclipse projects. At the end of 2016, the community consists of individuals and organizations from a cross section of the software industry.

**The Eclipse Developer Community**

Today, the Eclipse community consists of:

* Over 350 open source projects;
* 160 million lines of code;
* Over 1.500 individual committers;
* Over 5 million of active users;
* An average of 1,5 million downloads per month;
* An average of 2 million unique website visitors per month;
* A leading IDE on Java, C/C++, PHP, etc.;
* More than 270 members (including 12 strategic members);
* 75 events organized or co-organized each year, like EclipseCon events, Eclipse Days, Eclipse Summits or Eclipse DemoCamps, etc.

Joining this community brings the necessary visibility to assists the APPSTACLE consortium for the sustainability of its project and will help with the best practices of open source.

**Eclipse Working Groups**

Companies looking to drive innovation and efficiencies within their own organizations are increasingly looking for external sources to advance new ideas. Commonly referred to as “open innovation,” this paradigm encourages collaboration across organizational boundaries, and is often practiced in the open source community. In 2009, the Eclipse Foundation launched the notion of Working Groups to allow organizations to combine the best practices of open source development with a set of services required for open innovation, and in turn enabling organizations to foster industry collaborations.

Eclipse Working Groups provide a **vendor-neutral** governance structure that allows organizations to freely collaborate on new technology development. The Eclipse Foundation, through Eclipse Working Groups, provides five basic services to enable these types of collaborations:



Pillars of open collaborations

* **Governance**: Good governance that controls how decisions are made, policies established and disputes resolved is important for any successful collaboration.
* **Intellectual Property Management and Licensing**: Collaborations among different organizations require due diligences on the co-developed intellectual property. Eclipse Working Groups are established under the Intellectual Property (IP) policies of the Eclipse Foundation. These policies ensure that any open source software created in Eclipse projects is available for use by anyone, including developers of commercial software products.
* **Development Processes**: The Eclipse community has created a successful development process for large-scale distributed development that involves many different organizations.
* **IT Infrastructure**: The Eclipse Foundation manages the IT infrastructure for Eclipse Working Groups, including Git code repositories, Bugzilla databases, Hudson CI servers, development-oriented mailing lists and newsgroups, download sites, and websites.
* **Ecosystem Development**: An important way that the Eclipse Foundation supports the community is through active marketing and promotion of Eclipse Working Groups and the wider Eclipse ecosystem.

Any collaboration needs these services. Eclipse Working Groups make it easy to reuse the services provided by the Eclipse Foundation rather than creating them from scratch.

Today, the Eclipse Foundation has established five active working groups:

* **openMDM**: The openMDM Working Group (<http://www.openmdm.org/>) wants to foster and support an open and innovative eco-system providing tools and systems, qualification kits and adapters for standardized and vendor independent **management of measurement data** in accordance with the ASAM ODS standard.
* **LocationTech**: The LocationTech Working Group (<https://www.locationtech.org/>) is focusing on open source geospatial technologies.
* **Science**: The Science Working Group (<http://science.eclipse.org/>) is a collaboration of scientists developing software components used for basic **scientific research**.
* **PolarSys**: The PolarSys Working Group (<https://www.polarsys.org/>) was created by large industry players and by tool providers to collaborate on the creation and support of Open Source tools for the **development of embedded systems**.
* **Internet of Things**: The IoT Working Group ([http://iot.eclipse.org](http://iot.eclipse.org/)) is the place to learn about Eclipse technologies developed to make Internet of Things (IoT) development simpler.

Due to its focus, the APPSTACLE project collaborates in the **IoT working group**.

**The Eclipse IoT Working Group?**

Eclipse IoT Working Group ([http://iot.eclipse.org](http://iot.eclipse.org/)) is an ecosystem of companies and individuals that are working together to establish an Internet of Things based on open technologies. It provides open source implementations of the standards, services and frameworks that enable an Open Internet of Things.

Today the Eclipse IoT Working group has:

* 30 projects covering all the scope of the IoT from the device to the cloud, including gateway and edge devices;
* over 2.2 million Lines of Code;
* over 210 committers delivering the code;
* an average of 132K unique visitors per month.

The IoT Working Group is one of the most active working works at the Eclipse Foundation. Its most downloaded projects are:

* **Eclipse Paho** (16.5K downloads per month): Eclipse Paho (<https://www.eclipse.org/paho/>) provides open-source client implementations of MQTT and MQTT-SN messaging protocols aimed at new, existing, and emerging applications for the Internet of Things (IoT).
* **Eclipse Mosquitto** (12.8K downloads per month): Eclipse Mosquitto (<https://eclipse.org/mosquitto>) provides a lightweight server implementation of the MQTT protocol that is suitable for all situations from full power machines to embedded and low power machines. Sensors and actuators, which are often the sources and destinations of MQTT messages, can be very small and lacking in power.
* **Eclipse Californium** (4.4K downloads per month): Eclipse Californium (‘<https://www.eclipse.org/californium/>) is a powerful CoAP framework targeting back-end services and stronger Internet of Things devices. It provides a convenient API for RESTful Web services that support all of CoAP's features.
* **Eclipse SmartHome** (4K downloads per month): Eclipse SmartHome™ (<https://www.eclipse.org/smarthome/>) is a framework, not a ready-to-use solution. It offers a large set of features to choose from and leaves enough possibilities to design a smart home solution specific to your expectations. Its modular design brings millions of combinations and proves to be easily extensible by custom parts.
* **Eclipse Leshan** (2.5K downloads per month): Eclipse Leshan (<https://eclipse.org/leshan/>) is an OMA Lightweight M2M server and client in Java. It provides libraries that help people develop their own Lightweight M2M server and client.
* **Eclipse Kura** (600 downloads per month): Eclipse Kura (<https://www.eclipse.org/kura/>) is a Java/OSGi-based framework for IoT gateways. Kura APIs offer access to the underlying hardware (serial ports, GPS, watchdog, GPIOs, I2C, etc.), management of network configurations, communication with M2M/IoT Integration Platforms, and gateway management.
* **Eclipse Vorto** (N/A downloads per month): Eclipse Vorto (<https://www.eclipse.org/vorto>) is an open source tool that allows users to create and manage technology agnostic and abstract device descriptions, so called information models. Information models describe the attributes and the capabilities of real world devices and can be managed and shared within the Vorto Information Model Repository. Code Generators for Information Models then let you integrate devices into different platforms.

As noted above, APPSTACLE is planning to use several of these Eclipse projects: <TBC>.

It is clear that the Eclipse IoT Working Group is not only a great landing zone to promote the APPSTACLE technology and to gather early and involved adopters, but it is also a great open source platform to collaborate with.

[1] See Elinor Ostrom Wikipedia page:<https://en.wikipedia.org/wiki/Elinor_Ostrom>

[2] Governing the Commons: The Evolution of Institutions for Collective Action by Elinor Ostrom, 1991

[3] The Open Source Initiative WebSite is available at [https://opensource.org](https://opensource.org/))

[4] Annotated version of the definition of Open Source Software:<https://opensource.org/osd-annotated>