openPASS Website

Content

1 Startpage

- add a download button (e.g. like in the picture above) with a link to https://git.eclipse.org/r/plugins/gitiles/simopenpass/simopenpass/
- Rename "GOALS AND VISION" to "TARGET OBJECTIVES"
- Rename "TECHNOLOGY" to "ARCHITECTURE"
- Add a tab named "Resources" between "Working Group" and "Contact"
- Add a dropdown menu for "Resources" with the points "Installation Guide", "Tutorial: How to configure a simulation with openPASS", "Documentations", "Glossary" and "Report a bug", "Project Page"

2 Target Objectives

- Wrong index: "Abouts" rename to "TARGET OBJECTIVES"
- Rename Headline "Abouts" to "TARGET OBJECTIVES"
- Current paragraphs regarding "Contact", "Governance" an "Scope" should be removed

Text in section "Target Objectives":

The increasing demand of virtual simulation to assess the safety effects of advanced driver assistance systems (ADAS) and automated driving (AD) functions in traffic led to the foundation of

the Eclipse Working Group openPASS. The aim is to support the development of Eclipse projects, which provide a software platform that enables the simulation of traffic situations to predict real-world effectiveness of ADAS and AD functions. In short, openPASS specifies an open source platform for a prospective safety assessment of those technologies. Originally, the term "openPASS" formed a backronym for "Open Platform for the Assessment of Safety Systems", but openPASS has been expanded beyond the limitations of safety systems towards any kind of ADAS and AD functions. Thus, one of the main agreed target objectives is to become a broadly accepted effectiveness assessment platform for ADAS and AD functions. In line with our target objective, as a first step, we aim to develop a trustworthy, reliable and transparent platform.

The following presentation gives you a short glance about the idea of openPASS: Link to the pdf "openPASS as a glance" (4th March 2020).

3   Architecture
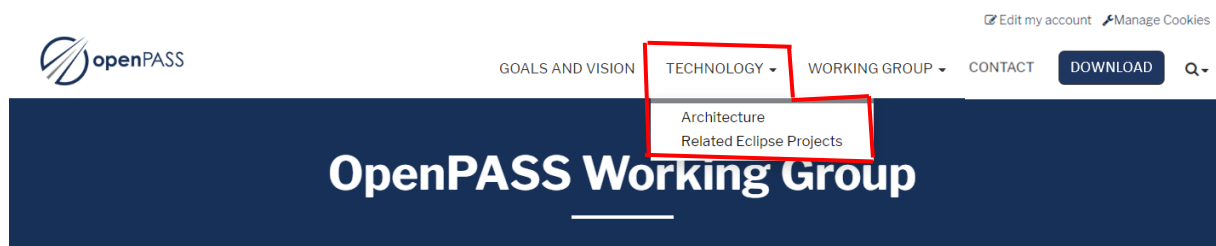
● Rename "Technology" into "Architecture"
● Delete Dropdown Menu

Text in this section:

Platform Concept

The platform concept is shown in the following picture.
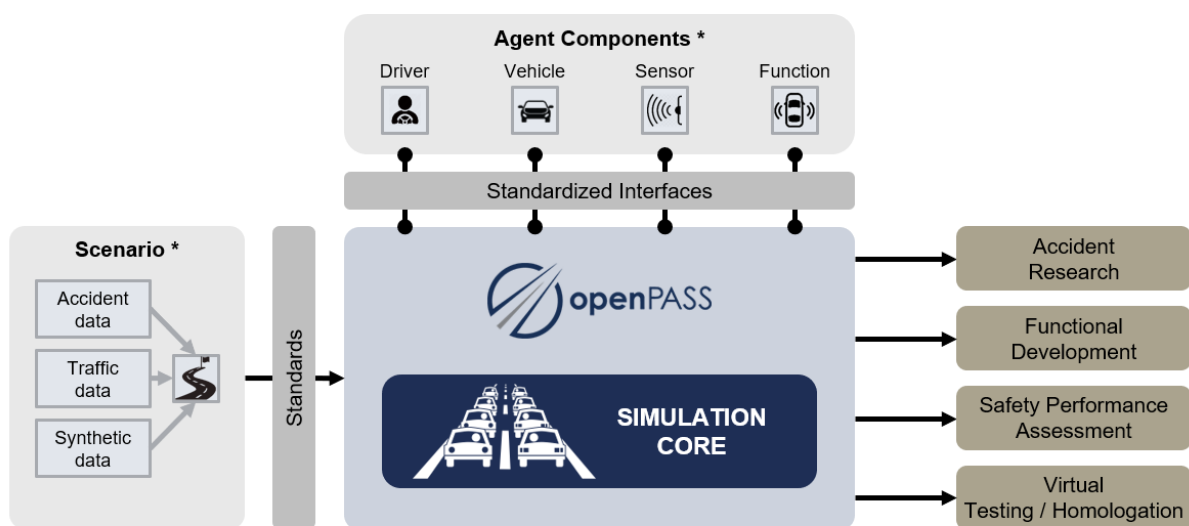
* Simple examples are provided

The platform is built up in a modular manner. The simulation core calculates different simulation runs. Scenarios and agent components can connect to the core over standardized interfaces like OpenSCENARIO, OpenDRIVE, OSI and FMI. The openPASS WG provides simple examples of scenarios and agent components. Consequently, the user can freely connect their own scenarios and agent components to the simulation core. Possible use cases

of using openPASS are accident research, functional development, safety performance assessment, virtual testing and virtual homologation.
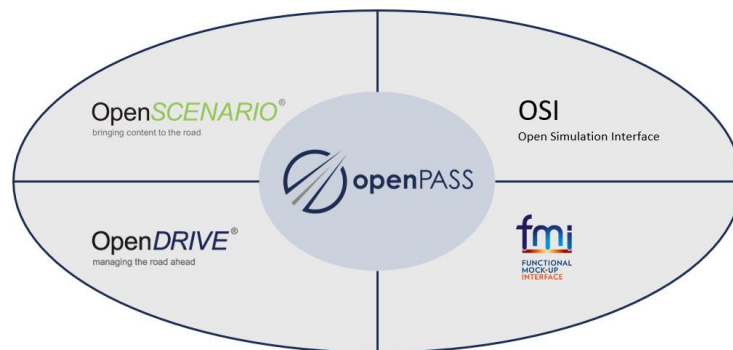
Top-level Architecture
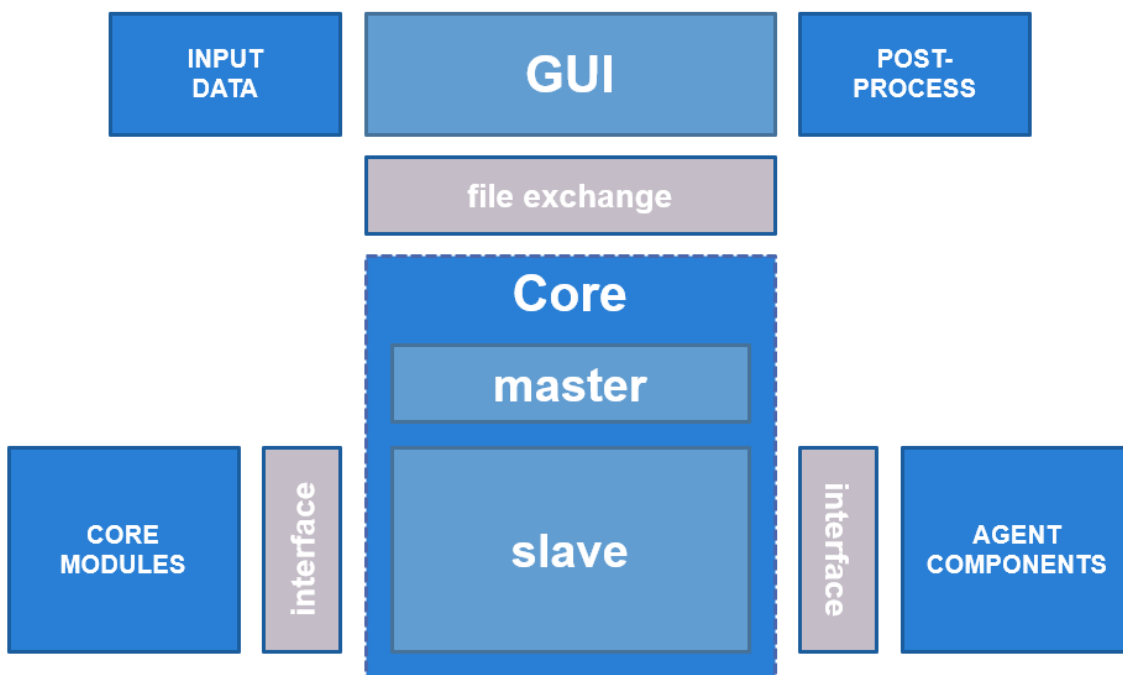
openPASS has two main parts: the graphical user interface (GUI) and the simulation framework.

The GUI is meant to give the user an easy access to configure experiments prior to simulation. In order to perform a traffic simulation an input must be available, which can be prepared and offered by one of the GUI plugins. In the GUI, the users will conveniently select traffic scenarios, traffic participants (including vehicles, drivers and systems to be tested) as well as post-processing of the simulation results. openPASS simulation accepts open formats such as openDRIVE, openSCENARIO and several openPASS-specific formats. Each file defines either simulation parameters, traffic situation, traffic composition, events or other key data. It is built as a modular software allowing attachment of independent or dependent plugins. A plugin can read input data (e.g. from a PCM database), process this data and prepare an experiment configuration, which can be understood by the simulation.

The simulation framework is a standalone executable (to be precise, two of them). It consists of:

- the simulation core (master and slave),
- the core modules

● and the agent components.

The core is a mostly generic assembly of data I/O routines, scheduler and a collection of interfaces. The master executable is meant to only collect and organize input data. The slave executable is the one performing the actual simulation. A master can, if configured so, start several experiments simultaneously by triggering several slaves.

The core modules and agent components belong to the slave application. Both communicate with the slave via interfaces and bound dynamically at runtime.

Core Modules
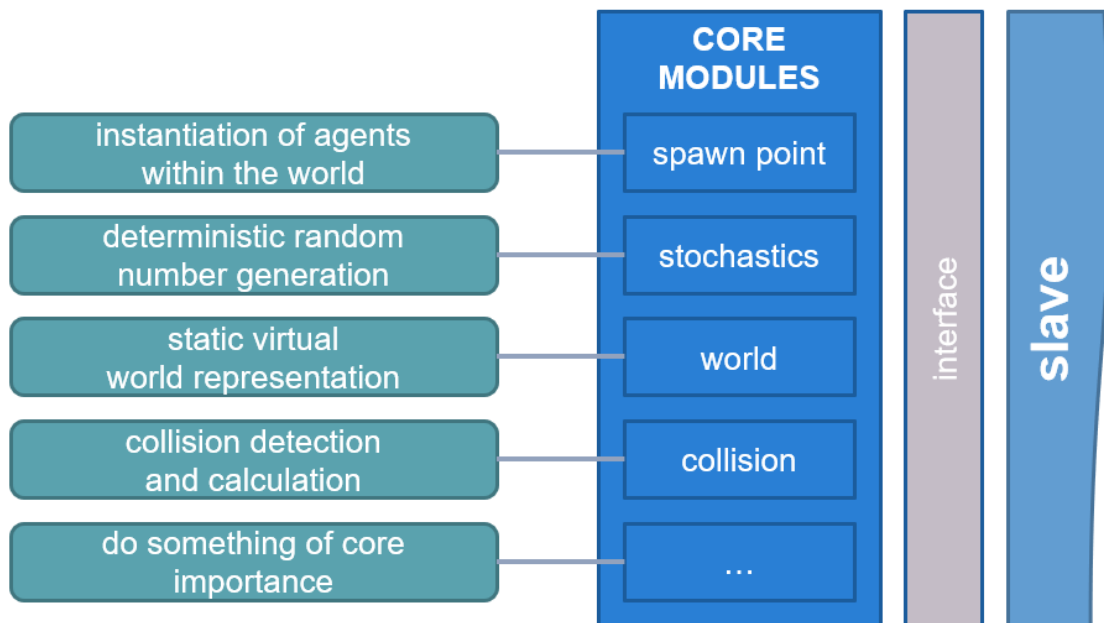
Core modules are all singletons and used by the slave and/or agent components. These are necessary for every simulation to run and cover basic needs like e.g.:
● contain world representation
● initiate agents within the world
● perform random number operations
● detect event
● log and output simulation results

On the other hand, agent components represent a composition of participants or agents. They define the behaviour and dynamics of each agent. A set of such components is then called system.
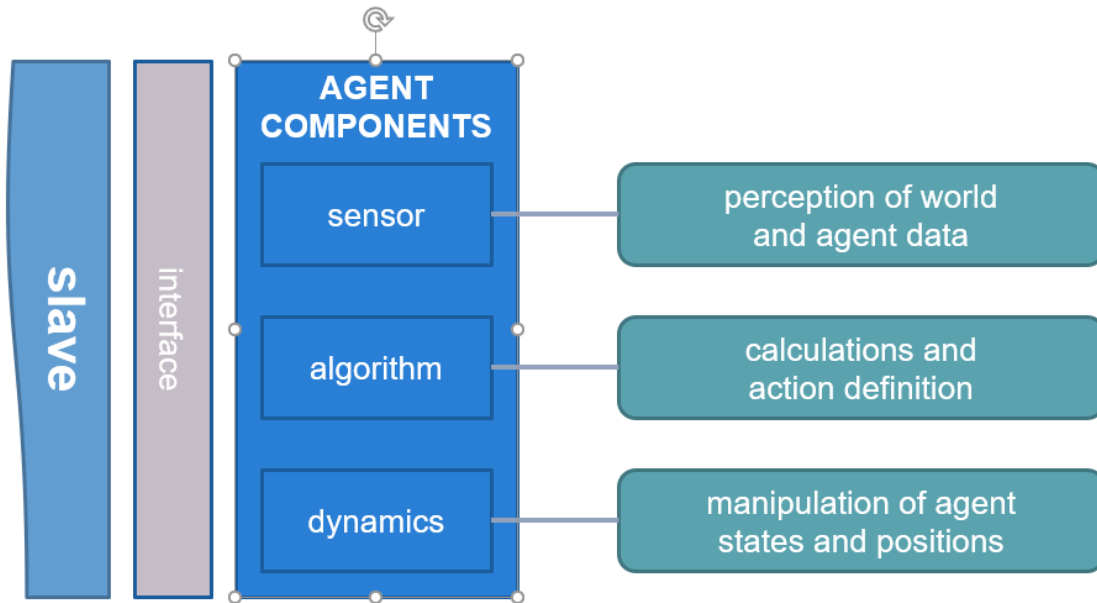
Unlike core modules, components can be freely selected and assembled by the user according to his scope of application. A typical system would consist of a sensor perceiving the environment, an algorithm performing analysis of this environment and making decisions and a dynamics algorithm receiving directives from the algorithm and calculating the actual physical simulation step. When assembling a system, the user shall of course care about connecting the chosen component via signals, which are understood by the sender and the receiver.

Completing the loop back to the graphical UI, a plugin can await the completion of one or several experiments and collect the output produced by the simulation in order to evaluate and/or visualize the results.

Simulation Process from the User Perspective
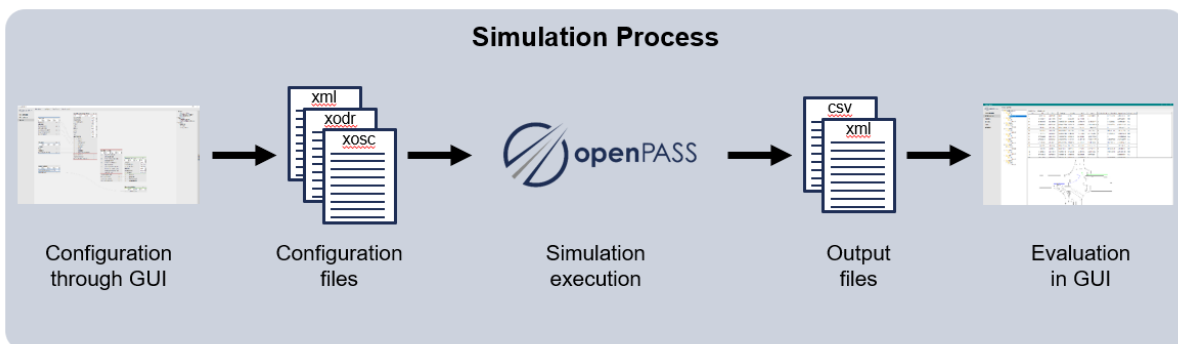
The illustration above shows the simulation process in openPASS. Over configurations in the GUI the user set up the desired simulation run. Basically, the GUI supports the user to set up the simulation in accessing and writing .xml, openSCENARIO and openDRIVE files. After the setup of the configuration files the simulation can be executed. As an output .csv and .xml files are generated. For the future, the evaluation over the GUI should be available which is yet still under development.

4 Working Group
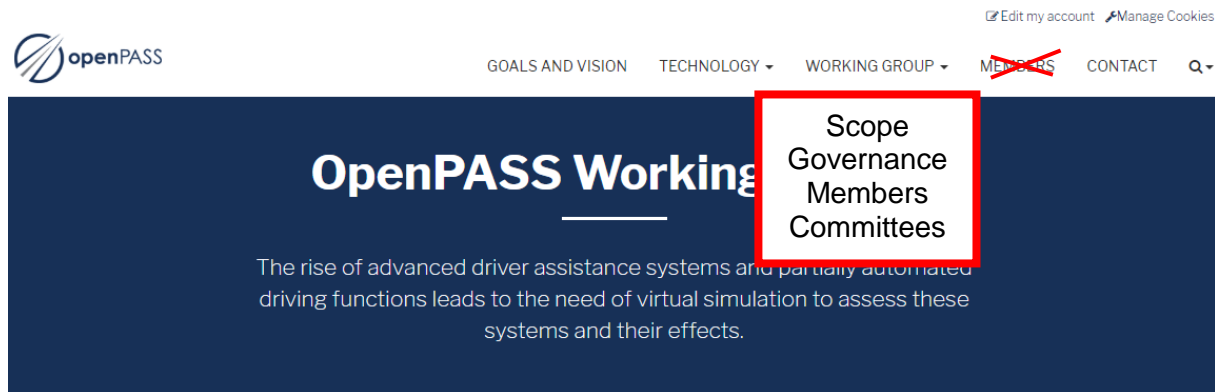
- Include the "Members" tab into the "Working Group" tab
- Delete the "Members" tab
- Dropdown menu in Working Group "Scope", "Timeline", "Governance", "Members", "Committees"

4.1 Scope

The Eclipse Working Group openPASS is managing the Eclipse project sim@openPASS.

Eclipse sim@openPASS provides a software platform that enables the simulation of traffic situations to predict the real-world effectiveness of advanced driver assistance systems or automated driving functions.

Safety relevant aspects of traffic, such as infrastructural elements or behavioural characteristics, are investigated with this platform. The platform can run entirely flexible multi-agent simulations which include stochastic variations as well as reproducibility. The flexibility is achieved by outsourcing to dynamic libraries (so called modules) all models defining the traffic scenario or behaviour of traffic participants. Additionally, to the platform, demonstrator modules are provided, which comprise simple traffic situations and behaviours of traffic participants as well as a collision model to answer basic research questions. Furthermore, demonstrator applications will be given to post process the simulation results. sim@openPASS users can incorporate their own developed modules and post processing applications; this allows them to answer a wide variety of their specific research questions. These questions are related to development, testing, and rating of vehicle functions.

Possible users can be automotive OEMs, suppliers, public transport authorities, consumer protection organizations, insurance companies, academia, and legislation.

4.2 Timeline
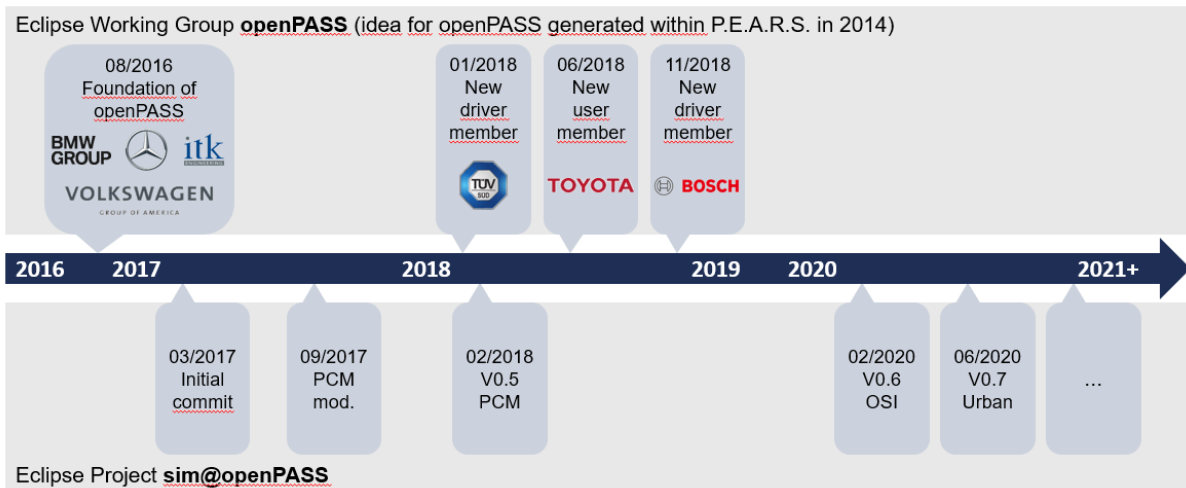
The Eclipse Working Group openPASS was founded in August 2016. The idea of openPASS was generated from the initiative P.E.A.R.S. The founding members were BMW Group, Mercedes Benz-AG, ITK Engineering GmbH and Volkswagen Group of America. In January 2018 TÜV SÜD Auto Service GmbH has joined the WG followed by Toyota Motor Europe NV/SA (TME) in June and Robert Bosch GmbH in November.

The WG is managing a single project namely sim@openPASS. The initial commit was in March 2017. Since then the open source platform has been developed further and several releases has been published.

4.3 Governance

The Governance is subject to the charter.

The following diagram shows the bodies that are distinguished in the Charter.

Privilege	Driver Member	User Member	Service Provider Member	Project Manager
Steering Committee	X	Elected	Elected	-
Architecture Committee	X	-	-	X
Quality Committee	X	Elected	Elected	X
General Assembly	X	X	X	-

4.4 Members

- display error in section Members (shown in picture above)

Further points in section Members:

- All logos should have the same size and orientation
- Bosch should be listed alphabetically under BMW (in category "Driver Members")

4.5 Committees

Product Manager: Tuan Duong Quang (TÜV Süd)

4.5.1 Steering Committee
Speaker of the Steering Committee: Jan Dobberstein (Daimler AG)
- BMW Group, Thomas Platzer
- Daimler AG, Jan Dobberstein
- Robert Bosch GmbH, Daniel Schmidt
- TÜV Süd, Emmeram Klotz
- Volkswagen Group of America, Inc., Jason Gainey
- ITK-Engineering GmbH, Dmitri Fix
- Toyota Motor Europe, Pablo Puente Guillen

To contact the openPASS Steering Committee, please write to openpass-sc@eclipse.org

4.5.2 Architecture Committee
Speaker of the Architecture Committee: Arun Das (BMW Group)
- BMW Group, Arun Das
- Daimler AG, Per Lewerenz
- Robert Bosch GmbH, Daniel Schmidt
- TÜV Süd, Emmeram Klotz
- Volkswagen Group of America, Inc., Jason Gainey
- ITK-Engineering GmbH, Dmitri Fix
- openPASS product manager, Tuan Duong Quang

Meetings minutes of both Committees can be found in our Eclipsepedia.

5 Resources

5.1 Sim@openPASS

Sim@openPASS is the platform which is created, developed and maintained by the openPASS WG. The Eclipse sim@openPASS platform mainly consists of a GUI and a simulation core interacting with openPASS modules as well as external programs for post-processing. More information about the project can be found under https://projects.eclipse.org/projects/technology.simopenpass. The following link guides you to git repository where you can download the simulation platform

<div align="center">

Download sim@openPASS

[https://git.eclipse.org/r/plugins/gitiles/simopenpass/simopenpass/]

</div>

5.2   Installation

Here [Link to open "OSI World Setup Guide.pdf"] you can download the installation guide for sim@openPASS.

5.3   Tutorial: How to configure a simulation with openPASS

Here [Link to open "Tutorial_openPASS_GUI.pdf"] you can download a tutorial on how to configure a simulation with openPASS.

5.4   Documentations

Here [Link to open "openPASS_Function_Docu.chm"] you can download the documentation to openPASS.

Here [Link to open "Documentation.chm"] you can download the documentation to the openPASS GUI.

5.5   Glossary

5.5.1   General

Term	Definition
Active safety Systems	Active safety systems are, within the context of the tool, systems that have notable safety effect on the course of simulated situations.
ADAS	Advanced Driver Assistance System
PCM of GIDAS	PreCrash Matrix from the database of the German In-Depth Accident Study

5.5.2   openPASS
5.5.2.1   Framework-Architecture

Term	Definition
Core	Consists of a master-slave system and provides the main functionality of the framework, like creating agents, scheduling tasks and coordinating the flow of information.
Core Module	A module that provides a main function to the core, like a detection of collisions

Term	Definition
	or some stochastic equations. These modules are not part of the core, but part of the framework itself. They´re exchangeable, so that some functionality can be implemented differently.
Framework	A software environment which provides the functionality to simulate a set of different traffic scenarios. Consists of the Core and the Core Modules.
openPASS	Framework for the simulative evaluation of active safety systems in vehicles.
OpenPassMaster	Coordinator of the simulation process. It starts one slave for each scenario.
OpenPassSlave	Simulator. It runs one scenario with a defined number of similar situations.

5.5.2.2 Agents

Term	Definition
Agent	An arbitrary static or dynamic object in the simulation of a traffic situation. Therefore, it might (but does not have to) be a traffic participant. An agent consists of one Agent Equipment and its physical parameters.
Agent Equipment	A set of components determining an agent's behavior.
Component	A module that defines a specific part of an agent's behavior, like sensing the environment (Sensor) or calculating a desired acceleration (Algorithm).
Signal	A general term that subsumes inputs and outputs. Signals are transported by channels within an agent's equipment.

5.6 Report a Bug

The openPASS WG uses Tuleap, an open source agile project management tool. Among other things, we are using it to track bugs, create user stories, epics and plan the next releases. Under the tracker "Bug" you can report your bug. Furthermore, you can find a summary which concludes all the previous fixed, ongoing and the new bugs. Click here to create your first bug.

6 Contact

- contact details should be side by side (like the green box) without hierarchical order

Contact us about Membership

The openPASS Working Group is open for all new interested members. Contact us if your organization is interested in participating in the development of the target objectives of openPASS. By joining you can discuss with us the roadmap and collaborate actively with the working group.

Join Us

- Add a join button in the contact tab https://accounts.eclipse.org/contact/membership

In order to participate in the openPASS Working Group, an entity must be at least a Solutions Member of the Eclipse Foundation. The annual membership fee for Solutions Members is tiered based on revenue and are determined as described in the Eclipse Bylaws and are listed in the Eclipse Membership Agreement (link to https://www.eclipse.org/org/documents/eclipse_membership_agreement.pdf).

In the openPASS charter https://www.eclipse.org/org/workinggroups/openpasswg_charter.php you can find additional information about the membership fee structure and membership privileges.