

# Standard interfaces to improve testing

February 19, 2018 | Focus on Positioning interface

**Philippe COLLIOT**

*Navigation expert, software developer, PSA Groupe*

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

GENIVI is a registered trademark of the GENIVI Alliance in the USA and other countries.

Copyright © GENIVI Alliance 2018.



# Preamble

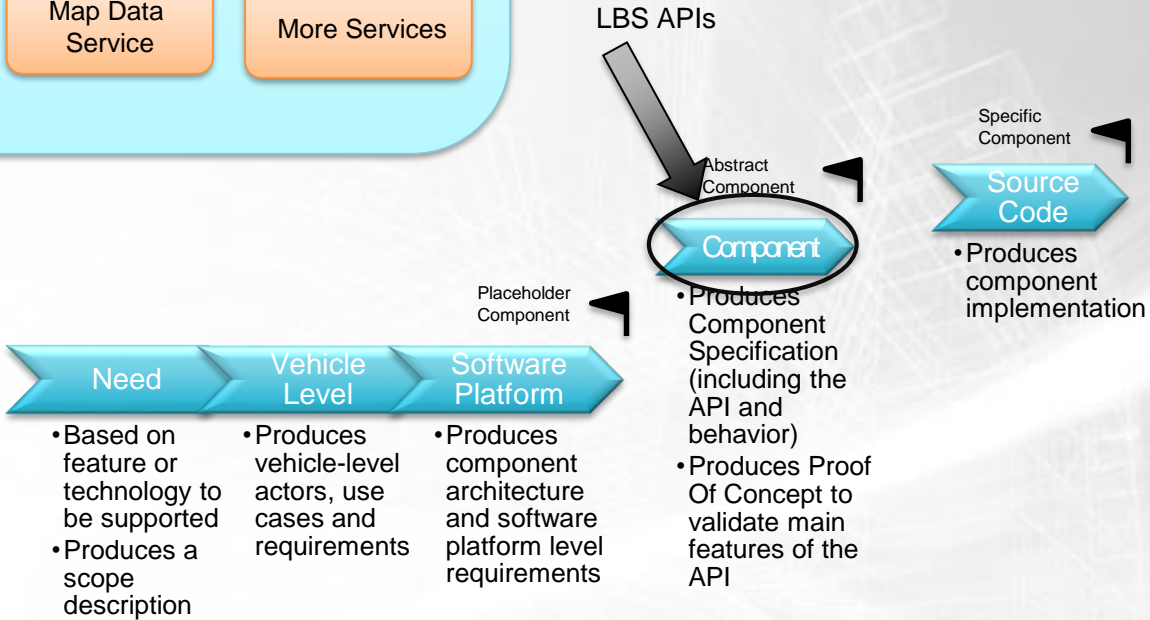
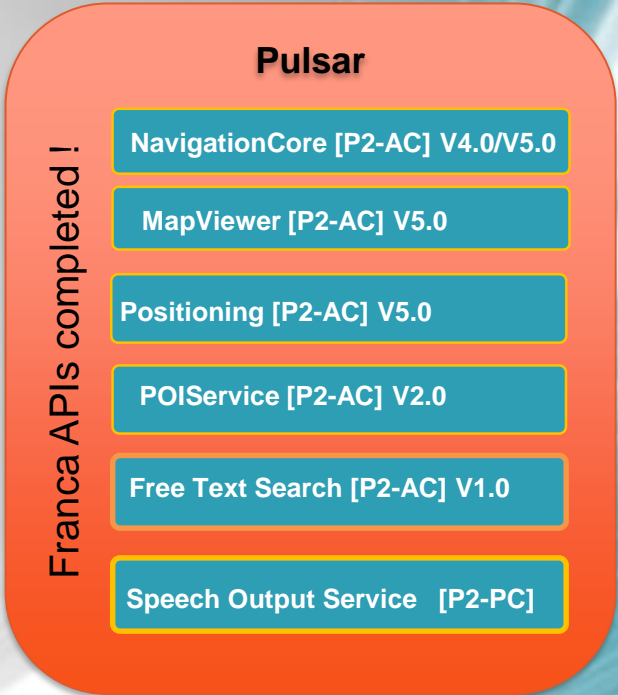
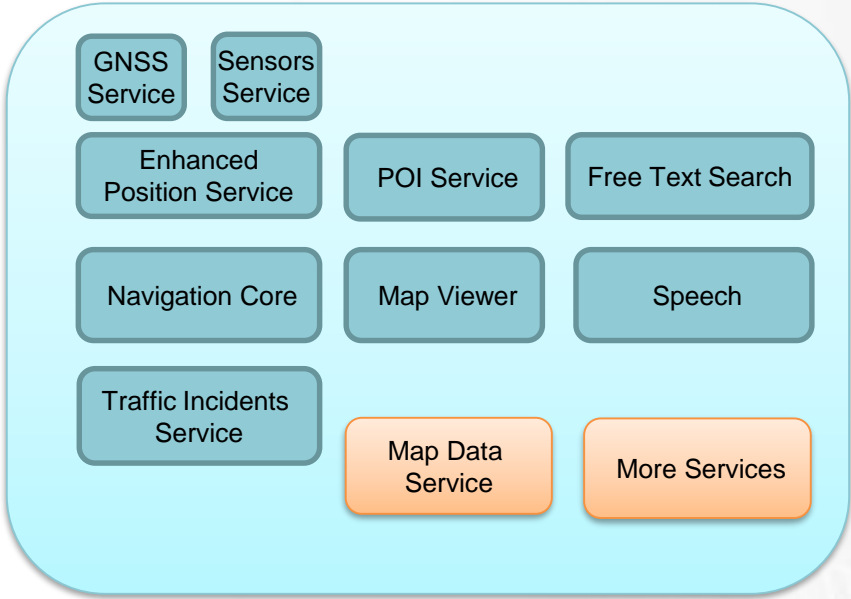
- Following slides are a snapshot of a work in progress project  
→so consider it 'as it is', some thought tracks
- PSA Groupe is a car manufacturer, 5 brands, millions of car sales but.. definitely not a software development company. Software is developed mainly by suppliers
- A car manufacturer has to care of integration, validation, continuous struggling for issues fixing of complex systems..  
→ that's why testing and use case context replaying is crucial

**Flashback on slides presented in Seoul fall 2017**

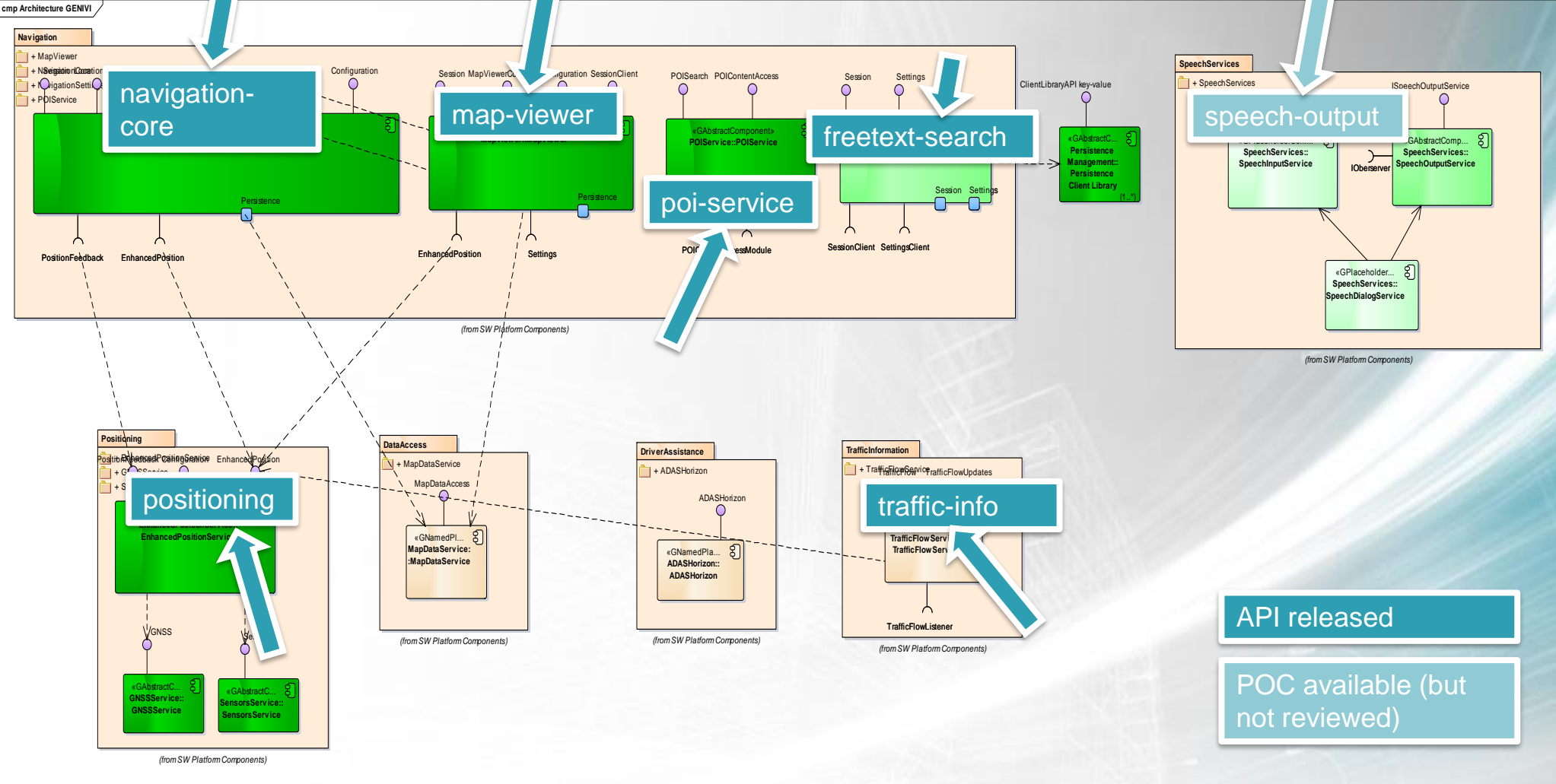




# Quick reminder, LBS-EG scope, status and components



# Portfolio of interfaces in details



# Available code and documentation

IVI Navigation Web portal

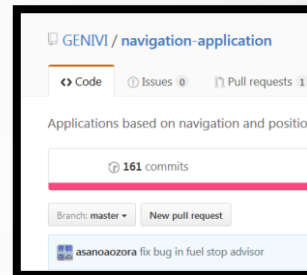
<https://at.projects.genivi.org/wiki/display/NAV/IVI+Navigation+Home>



Compliance documents

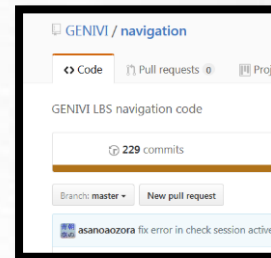
<https://collab.genivi.org/wiki/display/genivi/Compliance+Team>

APIs, documentation and code of proof of concepts in GitHub



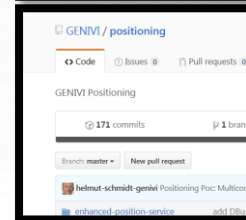
<https://github.com/GENIVI/navigation-application>

POC for navigation: client  
Fuel Stop Advisor application



<https://github.com/GENIVI/navigation>

POC for navigation: server & test script  
POC for POI search: server & client  
POC for Traffic Incident: server & client  
POC for FreeTextSearch (stubbed server)

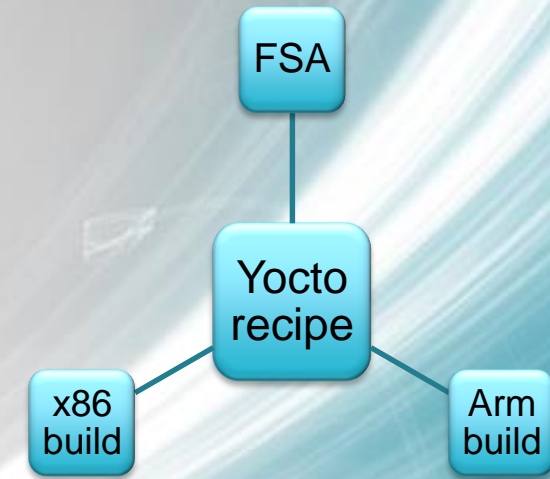
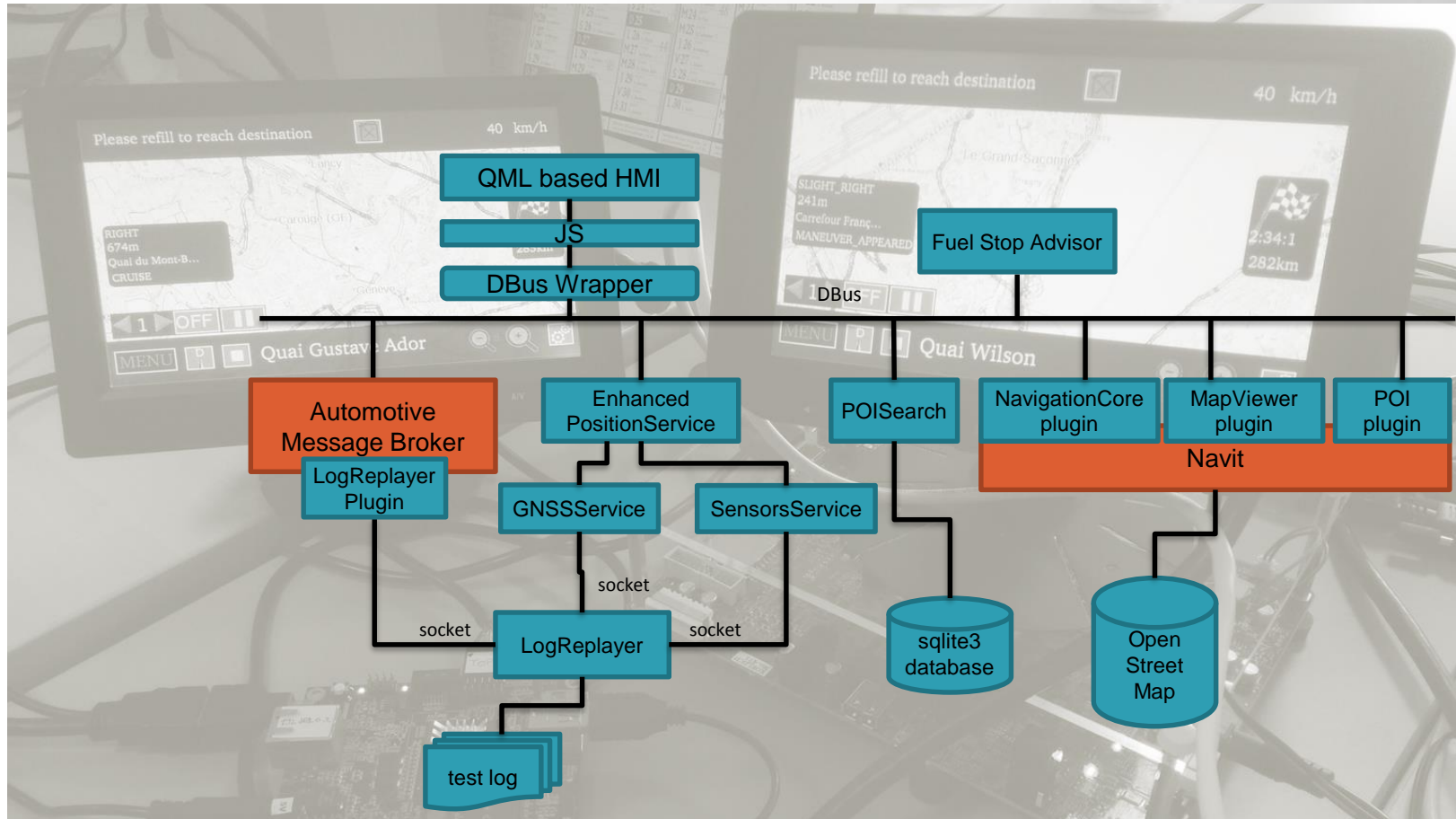


<https://github.com/GENIVI/positioning>

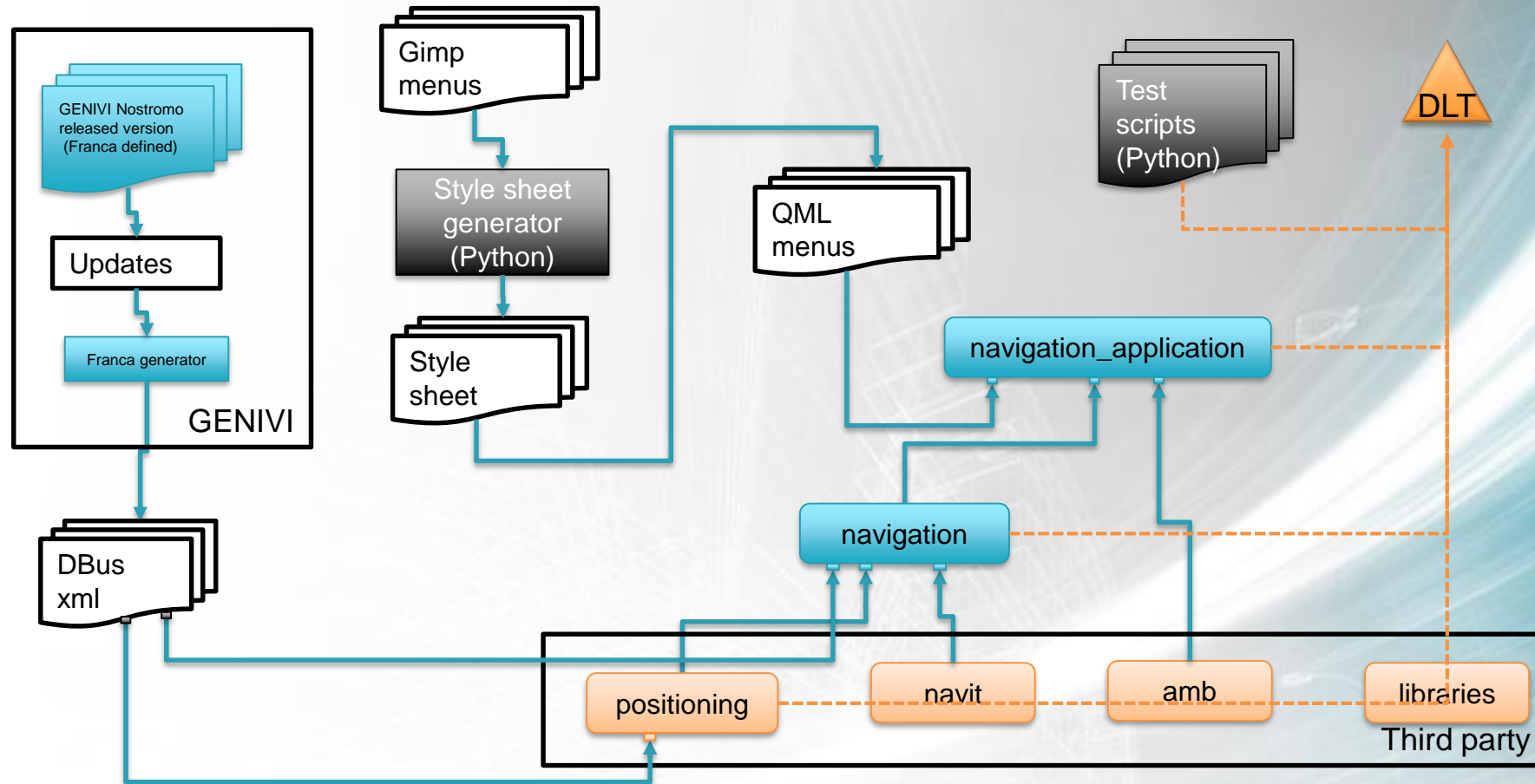
POC for positioning: server & test script



# FSA: an application built on top of LBS APIs



# Architecture of FSA





# How to make unitary tests running \*

- Build the navi with `./build.sh -cdlnp`
  - NB: `-l` to redirect log messages to DLT (in that case `dlt-daemon -d`)
- Launch the navi middleware with `./run -p`
- Test scripts batch `./test-all`
  - Detail of scripts
    - `test-location-input.py -l ../resource/locations.xml`
    - `test-address-input.py -l ../resource/location.xml`
    - `test-route-calculation.py -r ../resource/routes.xml`
    - `test-poi.py -l ../resource/location.xml -s "sChü"`
    - `test-poi-guidance.py -l ../resource/location-poi.xml -s "cOnT"`
    - `test-guidance.py -r ../resource/route.xml`
    - `test-map-viewer-control.py -l ../resource/location.xml`
- For testing speech, `./build.sh -cdlnps`, `./run -ps` and launch `test-speech.py`

```
psa ~/github/navigation/test/navigation/script $ ./test-all
Test location input PASSED
Test address input PASSED
Test route calculation PASSED
Test poi search PASSED
Test poi search/route/guidance PASSED
Test guidance/map viewer PASSED
Test map viewer PASSED
Batch test PASSED
```

Ctid	session	Type	Subtype	Mode	#Args
POIS	8057	log	info	ver_	1 POI content access server
POIS	8057	log	info	ver_	1 POI server
EPSP	8054	log	info	ver_	1 Starting EnhancedPositionService...
EPSP	8054	log	info	ver_	1 Starting EnhancedPosition dispatcher...
EPSP	8054	log	info	ver_	1 Starting PositionFeedback dispatcher...
EPSP	8054	log	info	ver_	1 Starting Configuration dispatcher...
MPVC	8058	log	info	ver_	1 map viewer control server
POIC	8072	log	info	ver_	1 POI content access module server
POIC	8072	log	info	ver_	1 POI content access client
POIS	8057	log	info	ver_	1 POI content access module client
POIC	8072	log	info	ver_	1 camid-1
TRPT	8093	log	info	ver_	2 Start test of location input
LOIS	8072	log	info	ver_	1 Creation location input
LOIS	8072	log	info	ver_	1 Set selection criterion
LOIS	8072	log	info	ver_	1 Search
LOIS	8072	log	info	ver_	1 Idle callback
LOIS	8072	log	info	ver_	1 Select entry
LOIS	8072	log	info	ver_	1 Set selection criterion
LOIS	8072	log	info	ver_	1 Idle callback

\* Under <https://github.com/GENIVI/navigation/tree/master/test/navigation/script>

# Test with Qemu

The screenshot displays a Linux desktop environment with a dark blue theme. In the top-left corner, there are several desktop icons: a folder named 'API\_PSA', a user icon for 'F.Culture' with ID '13364-' and date '06.11.2016-', a folder named 'common-api-xml-', a folder named 'Franca', and a file named 'Screenshots.png'. A terminal window is open in the center-left, showing the following commands and output:

```
psa ~ $ ls *sh
check-fsa-qemu.sh launch-fsa-qemu.sh update.sh
fsa-qemu.sh start-fsa-qemu.sh
psa ~ $ ./launch-fsa-qemu.sh
psa ~ $ ./launch-fsa-qemu.sh &
[1] 3471
psa ~ $ ./start-fsa-qemu.sh -h
Usage:
./start-fsa-qemu.sh [-hi]
-h: This help
-i: start the interface
psa ~ $ ./start-fsa-qemu.sh -i
psa ~ $ ./check-fsa-qemu.sh
```

In the top-right corner, a terminal window titled 'fsa-qemu.sh' shows the following output:

```
[sudo] password for psa:
WARNING: Image format was not specified for 'genivi-dev-pla'
and probing guessed raw.
Automatically detecting the format is dangerous fo
perations on block 0 will be restricted.
Specify the 'raw' format explicitly to remove the

Yocto GENIVI Baseline (Poky/meta-ivi) 13.0+snapshot qemuX86

qemuX86-64 login: root
Last login: Mon Oct 9 01:41:43 UTC 2017 on tty3
root@qemuX86-64:~#
```

In the bottom-right corner, a navigation application window titled 'ROOT: QEMU' is visible. It shows a map with a route highlighted in yellow. The map includes labels for 'Place de Neuve', 'Statue du Général Dufour', 'Rue du Général-Dufour', 'Rue Francis Diday', 'Musée Rath', 'Rue de la Constance', and 'Rue de la Terrasse'. The application displays a time of '01:45' on 'Monday, Oct 9', a distance of '172m', and a speed of '40 km/h'. A checkered flag icon indicates a destination, with a time of '1:57' and a distance of '7461m'. The bottom of the window shows a 'Rampe de la Treille' button and a 'BACK' button.

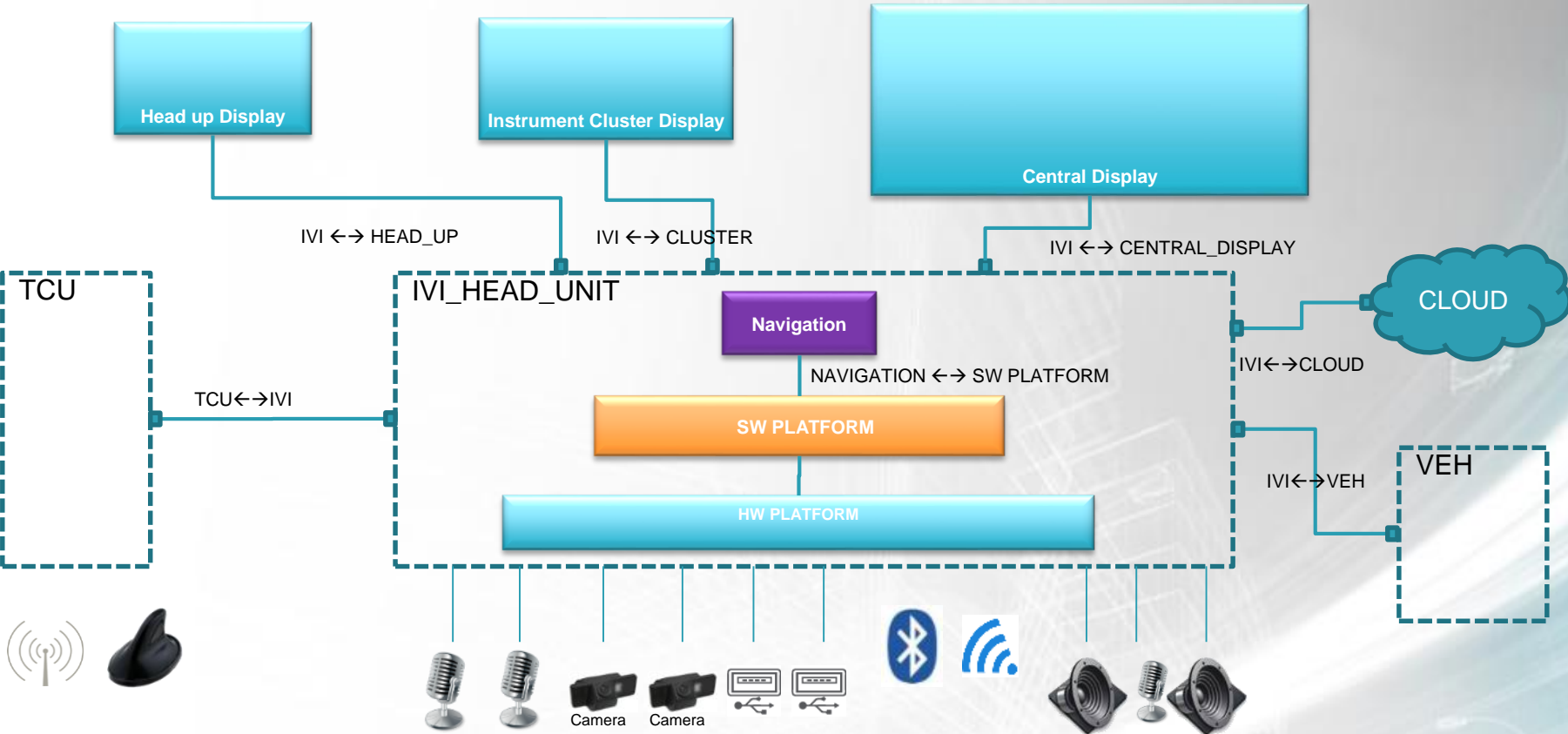
- So bunch of code is available that allows:
  - To experiment new tools
  - To get a feedback of code running on top of a 'real' navigation
- Prototyping is important for promotion towards top management



# Product issues feedback



# PSA architecture overview



# Overview of the source of issues

## HMI

- Wrong translation
- Size exceeded

## MAP VIEW

- Slow FPS

## NAV ENGINE

- Wrong route

## MAP DATABASE

- Wrong or missing road

## CLOUD

- Connection failure
- Map inconsistency

## POSITION

- Wrong vehicle position



# Typical issues faced when running a navigation

## Parallel road issue



Because of a not accurate dead reckoning, vehicle is located on the wrong road

## Roundabout issue



Because of a not accurate dead reckoning, map matching is erratic

## Map data issue



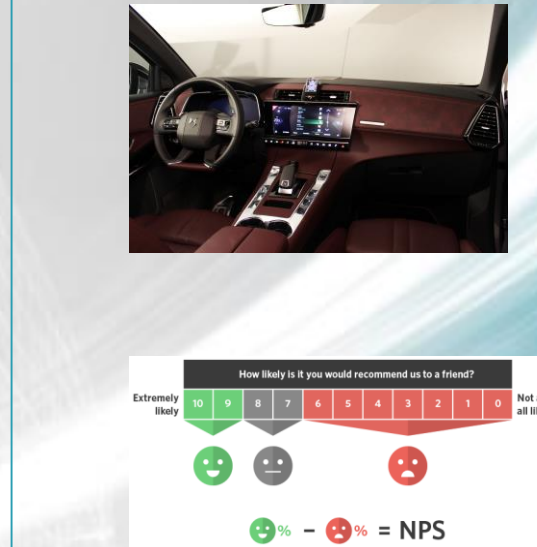
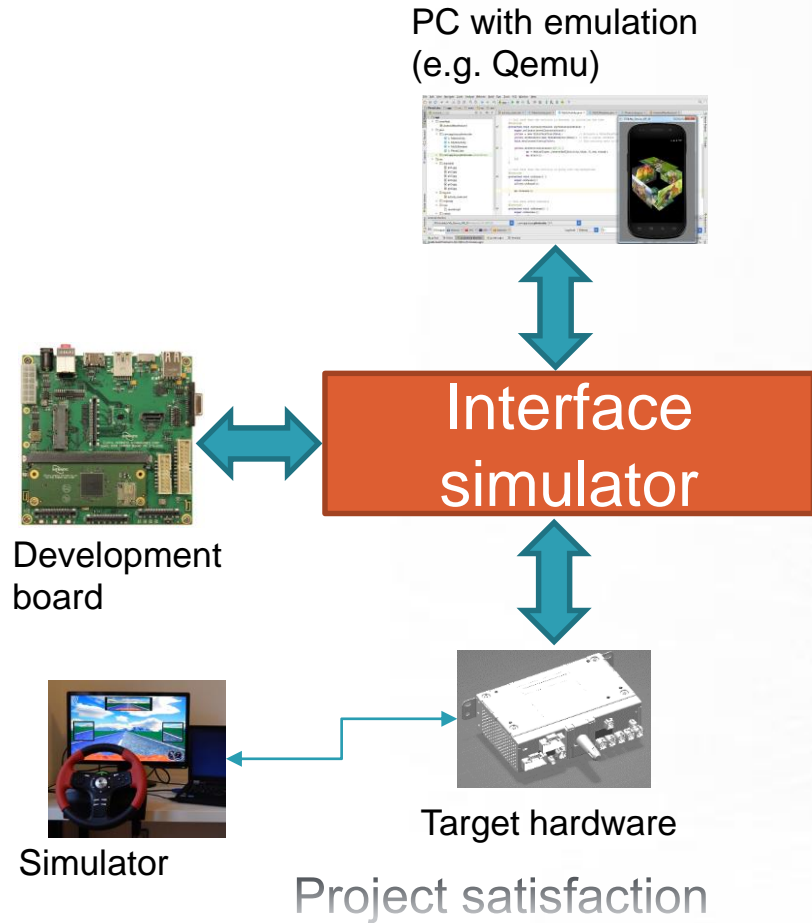
Because of map error, the road change is not detected in time

# Test configurations

Development phase

Vehicle passing tests

Vehicle on the market

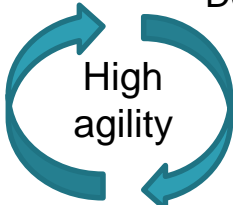


NPS: Net Promoter Score

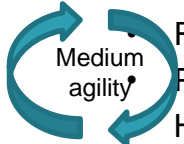


# Test scenari

- Three main phases



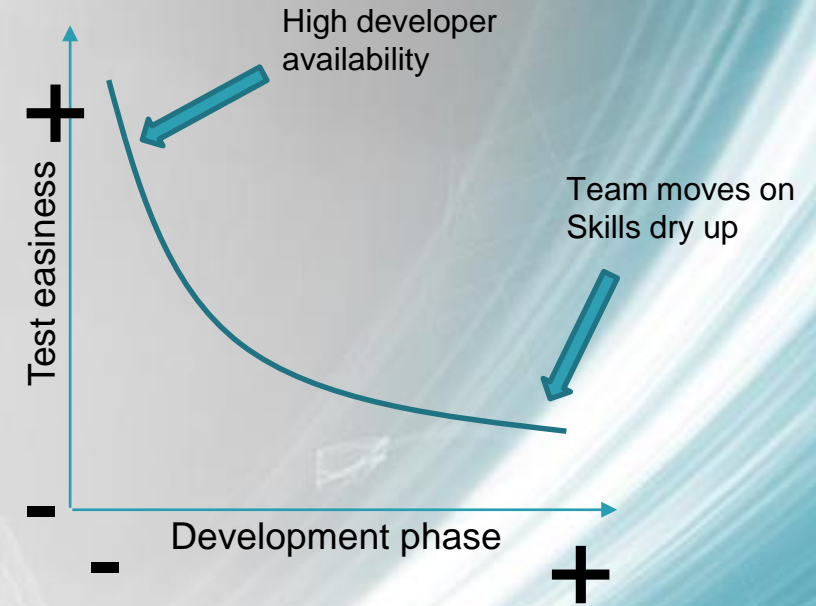
- Development phase (full test tool available)
  - Unitary, integration and validation tests
  - Build with debug, dedicated test software...
  - Code run on PC (emulated) or on target
  - Simulated interfaces



- Vehicle passing tests (more constraints)
  - Real vehicle interfaces, life cycle and use cases
  - Potentially new issues raised
  - How to duplicate on test bench ?



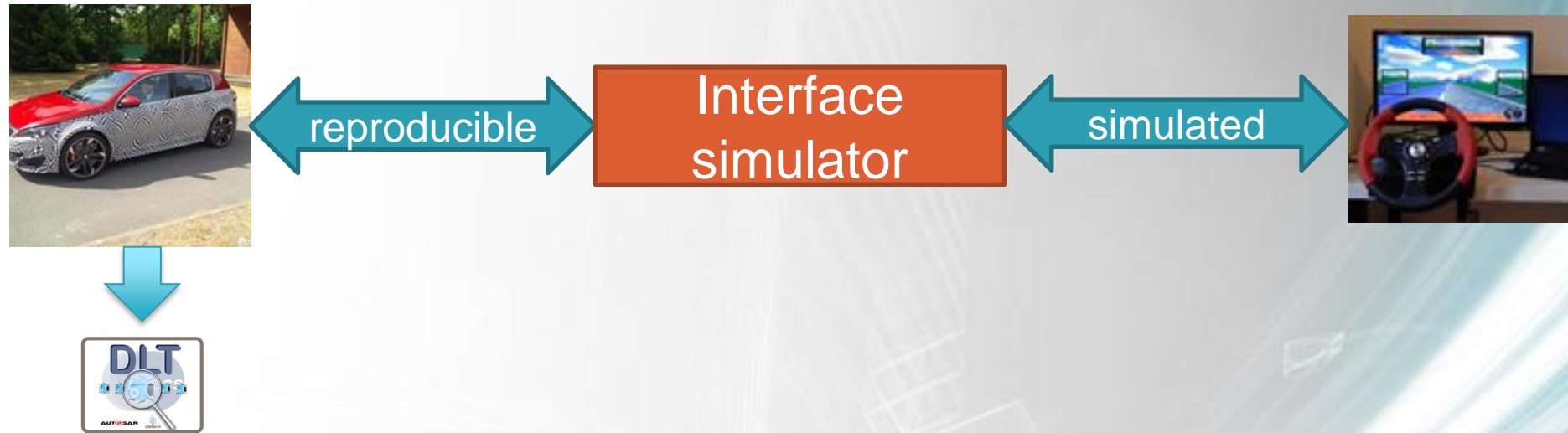
- Vehicle on the market (logs, limited spying possibilities)
  - Issue raised by the end user (verbatim), so customer dissatisfaction
  - How to deal with issue ?
    - Minor → maybe ignore or postpone to future release
    - Major → crisis !!



Issue duplication, simulation and automatic test are the key items !

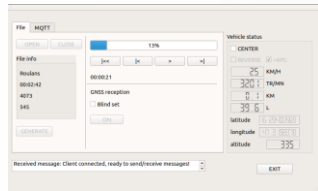


# Full test coverage, realistic simulation: the quest for the Holy Grail



- Prerequisites:
  - A realistic vehicle dynamic model (with target vehicle settings)
  - Geo-located terrains with road network
- Sensors of the system have to be simulated through external access (i.e. non intrusive testing)

# Testing overview



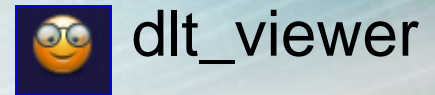
.rec

.kml

.gpslog

DLT

DLT



Format converter  
Additional data  
calculation

.log

Log replayer

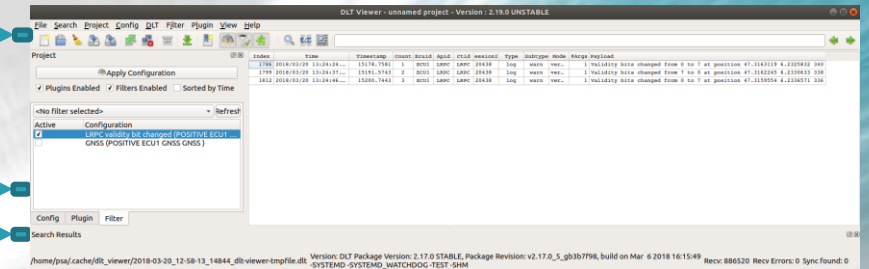
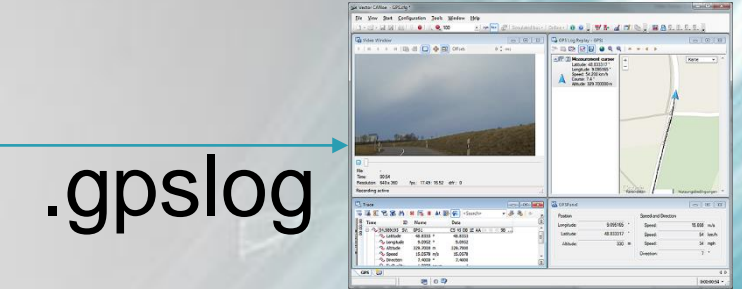
Sensors server

AMBD

GNSS server

DBus

Test client



# Next steps

- Old technology of the simulator with no support → need to move to more up-to-date one (a lightweight simulator)
  - GENIVI simulator, OpenDS ?
  - How to reuse vehicle dynamic model, terrain models, sensors simulation ?
- Define DLT format (based on LogReplayer) to replay testing situations
- Push the tool on Github
- How to face the huge amount of data, in a global testing perspective of an IVI system in an extended vehicle scope



# Thank you!

Visit GENIVI at <http://www.genivi.org> or <http://projects.genivi.org>

Contact us: [help@genivi.org](mailto:help@genivi.org)

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

GENIVI is a registered trademark of the GENIVI Alliance in the USA and other countries.

Copyright © GENIVI Alliance 2018.

