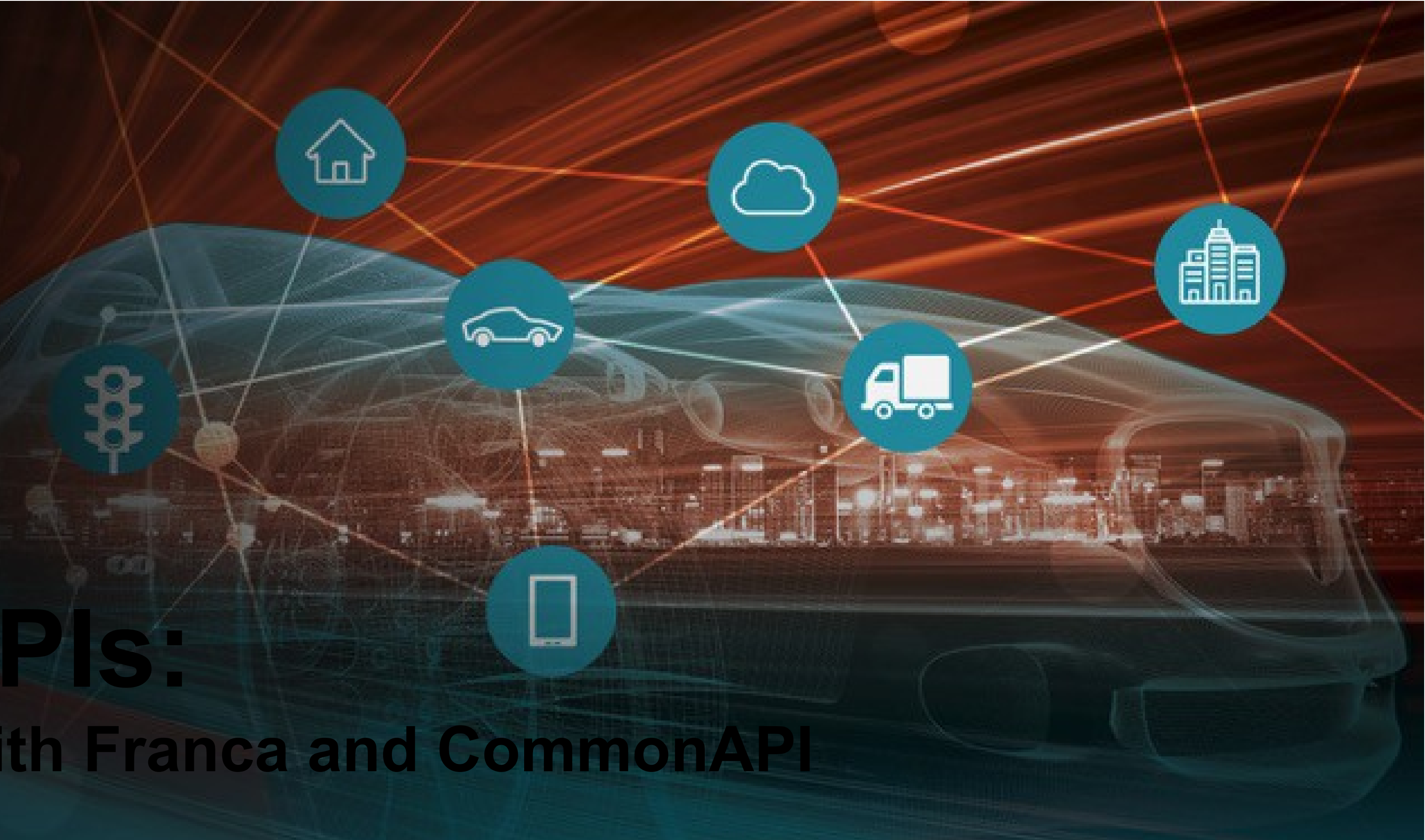


Navigation APIs:

From native to Web with Franca and CommonAPI

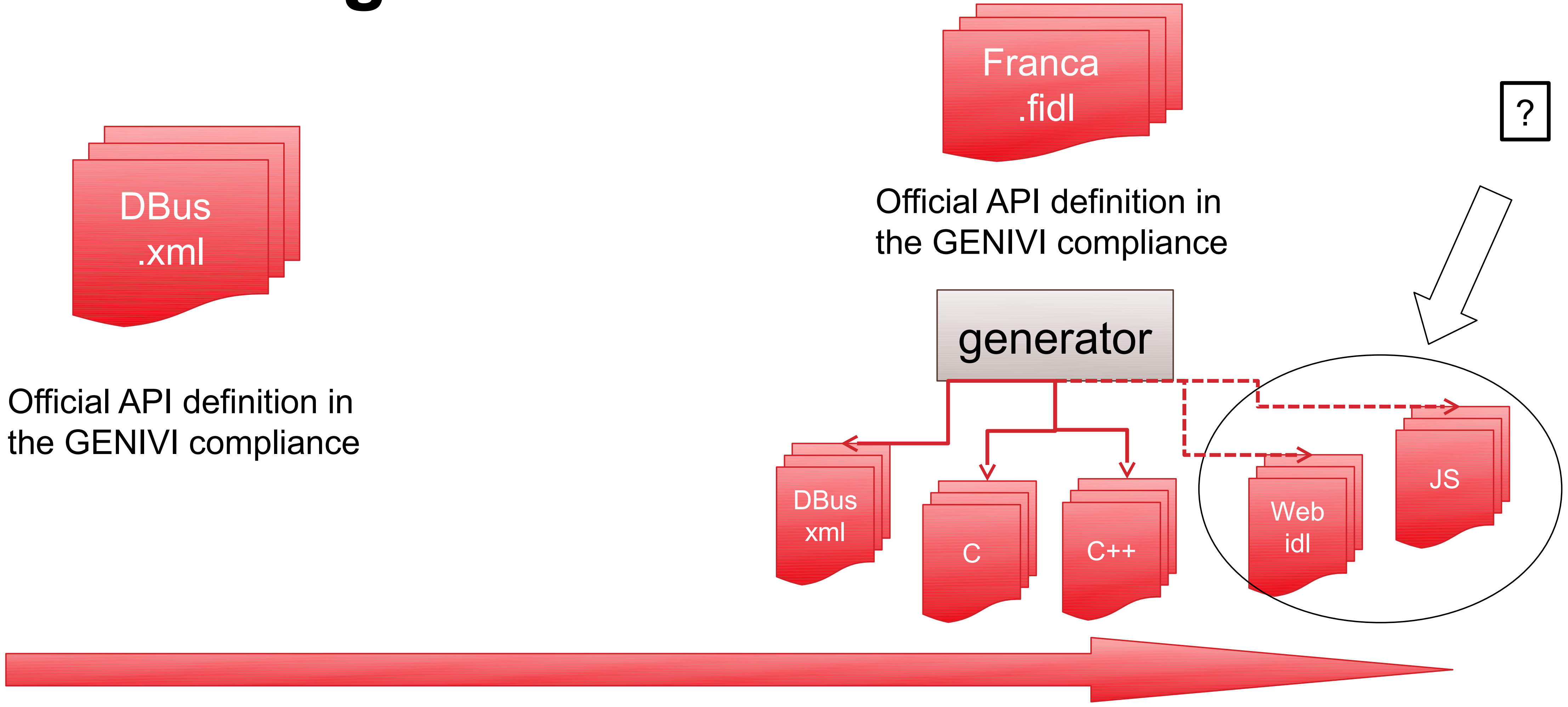
October 20, 2016 | **AMM Burlingame - All Members**

Philippe COLLIOT, LBS-EG Lead
PSA Group

Purpose of the presentation

- To report on the Franca and Common API migration of the LBS-EG interfaces
- To give a status of the Navigation Web APIs
- To discuss about how to achieve the goal and to concretely complete the Web API

Reminder: goals



Official API definition in the GENIVI compliance

Official API definition in the GENIVI compliance

Today

Tomorrow

Statement of work

05/16

Migration started (with draft Franca files provided by EB)

08/16

POC update started *

10/16

Release Candidate
POC updated
FSA under debug

12/16

‘Hand written’
Navigation Routing
Web API

??/17

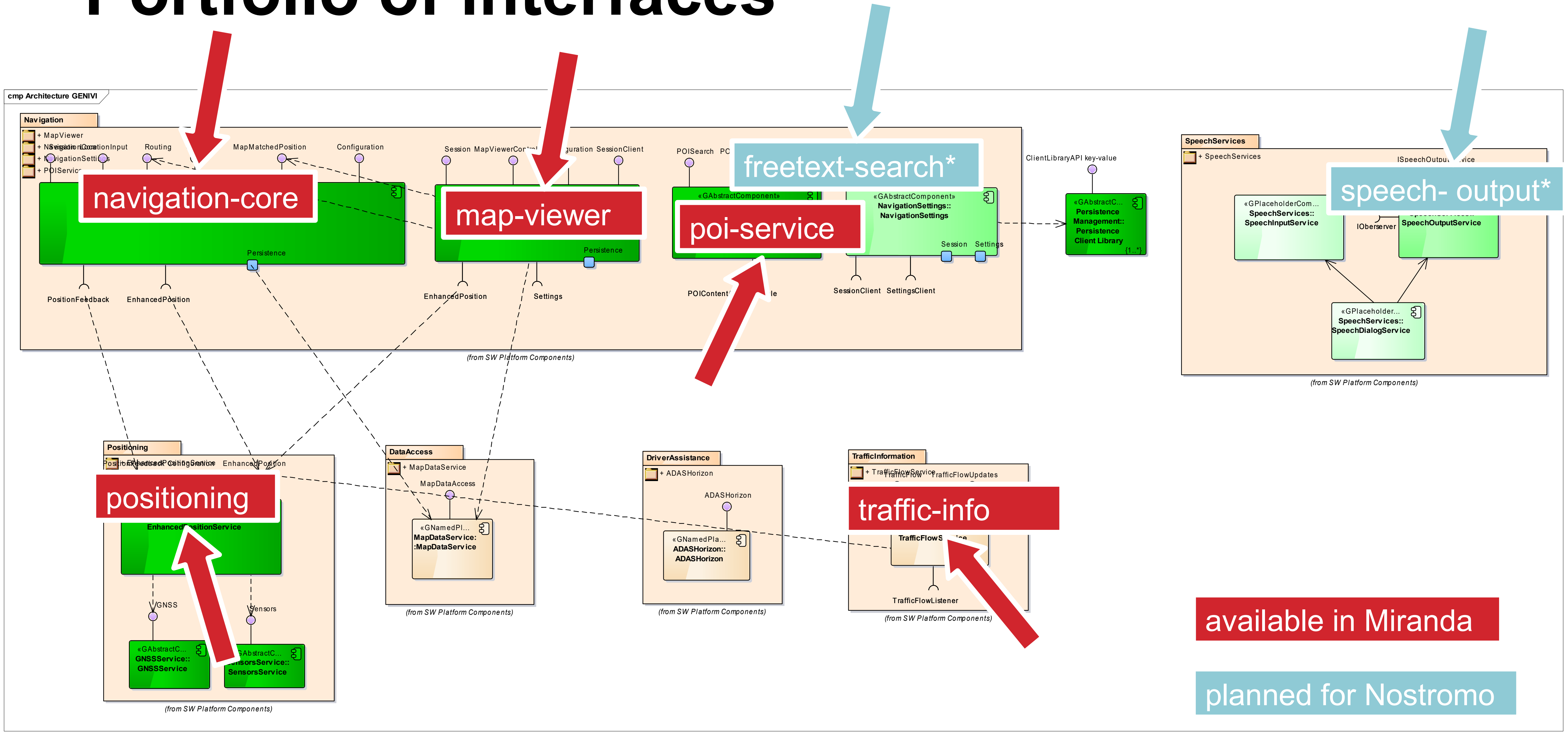
Franca generator

* As much as possible, the former DBus definition is aligned and the POC updated accordingly

Overview of the LBS APIs

Portfolio, available documentation…

Portfolio of interfaces

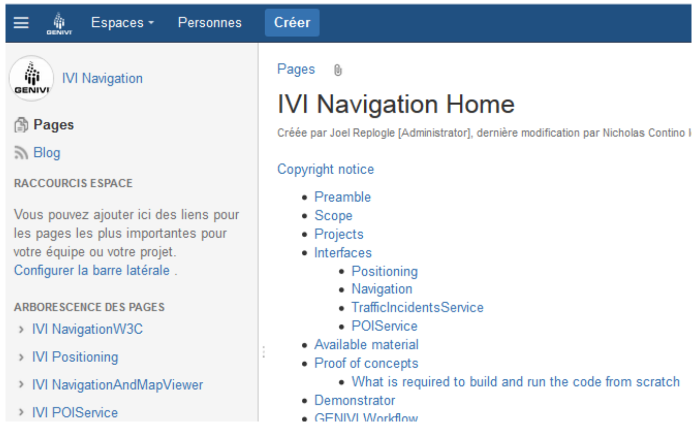


available in Miranda

planned for Nostromo

Available documentation

IVI Navigation Web portal
https://at.projects.genivi.org/wiki/display/NAV/IVI+Navigation+Home



IVI Navigation

Pages

Blog

RACCOURCIS ESPACE

Vous pouvez ajouter ici des liens pour les pages les plus importantes pour votre équipe ou votre projet.
Configurer la barre latérale .

ARBORESCENCE DES PAGES

- IVI NavigationW3C
- IVI Positioning
- IVI NavigationAndMapViewer
- IVI POIService

Pages

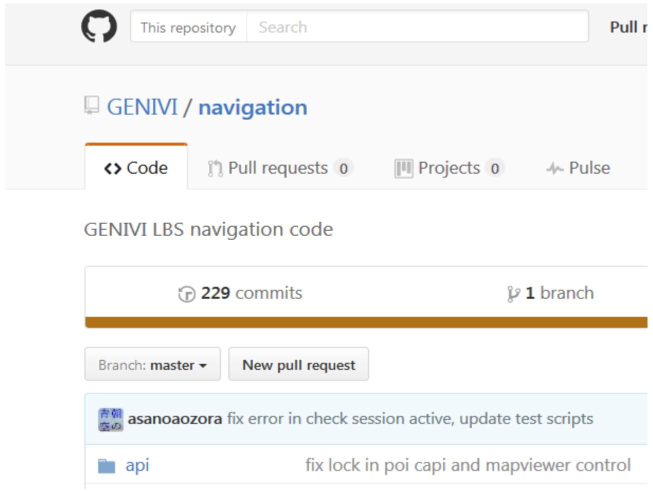
IVI Navigation Home

Créée par Joel Replogle [Administrator], dernière modification par Nicholas Contino l

Copyright notice

- Preamble
- Scope
- Projects
- Interfaces
  - Positioning
  - Navigation
  - TrafficIncidentsService
  - POIService
- Available material
- Proof of concepts
  - What is required to build and run the code from scratch
- Demonstrator
- GENIVI Workflow

APIs, documentation and code of proof of concepts in GitHub



GENIVI / navigation

GENIVI LBS navigation code

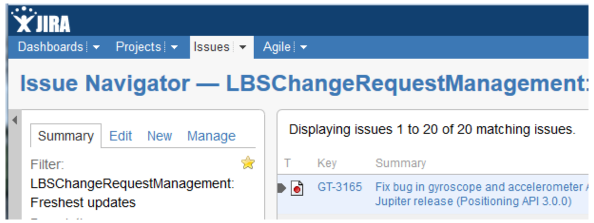
229 commits

1 branch

asanoaozora fix error in check session active, update test scripts

api fix lock in poi capi and mapviewer control

Change request management by Jira issues



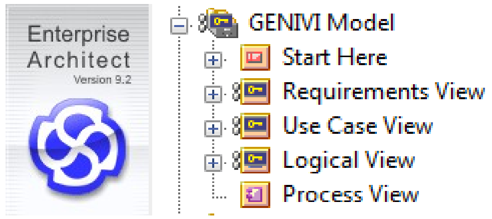
Issue Navigator — LBSChangeRequestManagement:

Filter:
LBSChangeRequestManagement:
Freshest updates

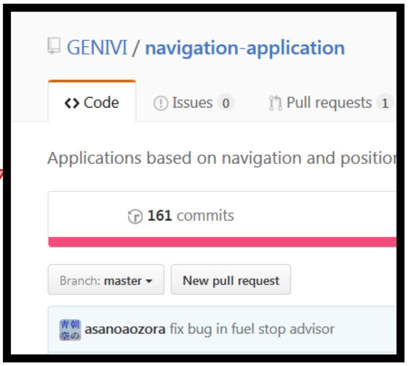
Displaying issues 1 to 20 of 20 matching issues.

T	Key	Summary
	GT-3165	Fix bug in gyroscope and accelerometer Jupiter release (Positioning API 3.0.0)

Use cases and requirements into an UML model



Interface release and proof of concept



GENIVI / navigation-application

Code | Issues 0 | Pull requests 1

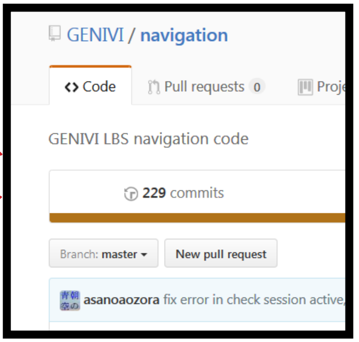
Applications based on navigation and positio

161 commits

Branch: master | New pull request

asanoaozora fix bug in fuel stop advisor

https://github.com/GENIVI/navigation-application
POC for navigation: client
Fuel Stop Advisor application



GENIVI / navigation

Code | Pull requests 0 | Proj

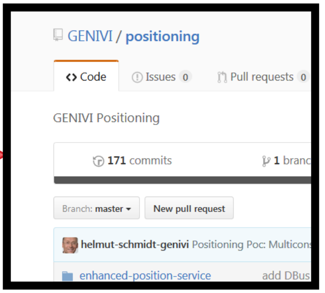
GENIVI LBS navigation code

229 commits

Branch: master | New pull request

asanoaozora fix error in check session active

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 Speech Output: server & client



GENIVI / positioning

Code | Issues 0 | Pull requests 0

GENIVI Positioning

171 commits | 1 branc

Branch: master | New pull request

helmut-schmidt-genivi Positioning Poc: Multicon

enhanced-position-service | add DBus

https://github.com/GENIVI/positioning
POC for positioning: server & test script

Migration to Franca and CommonAPI, a feedback

Naming constraints, reserved keywords …

GENIVI®

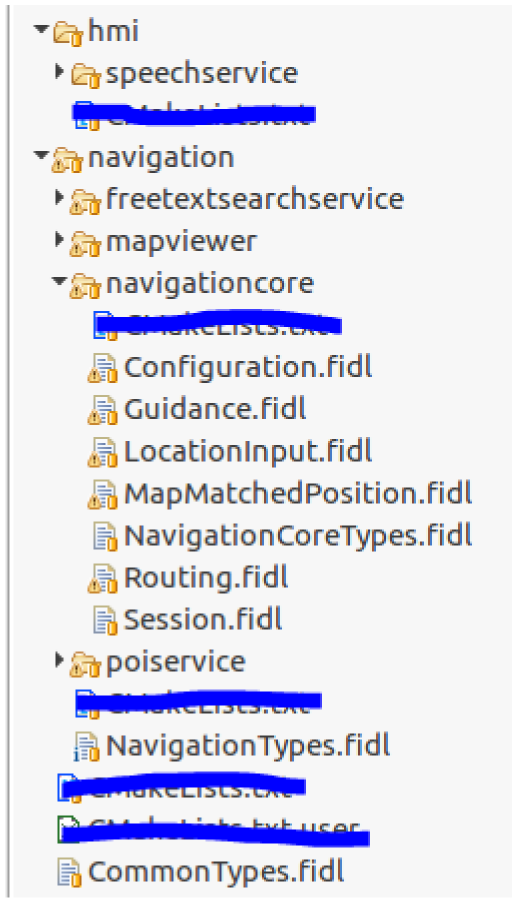
Versions

- Current versions used are:
  - Franca 0.9.1
  - Common API C++ and C++ DBus 3.1.5 v2
  - DBus patched 1.8.14
- Plan to move to the latest ones after the AMM

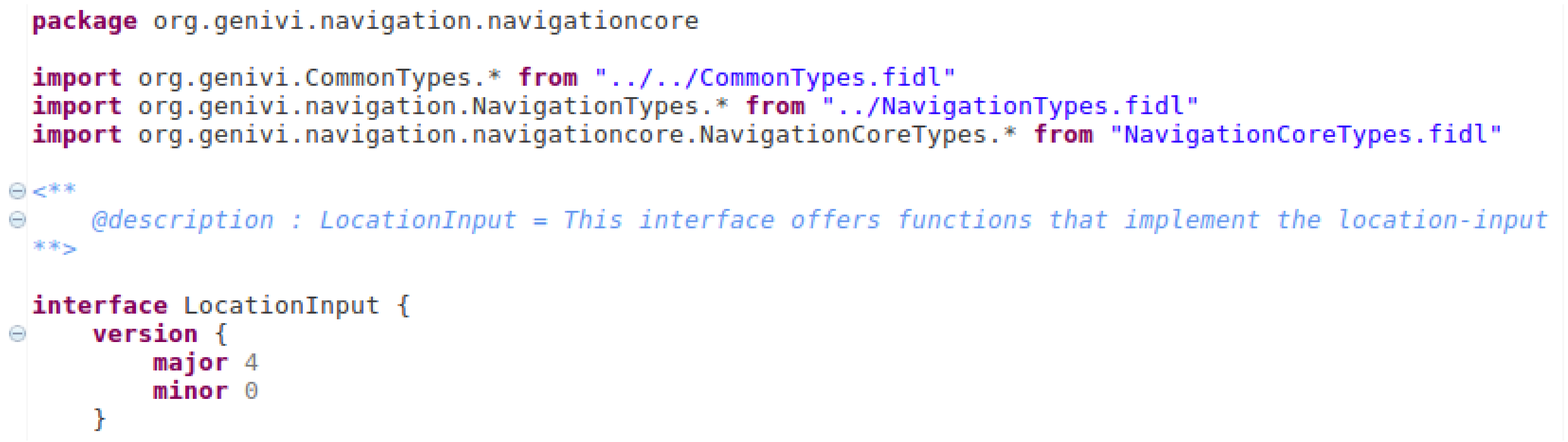
Overview of faced issues

- At API definition level:
  - Differences in naming conventions (camel case with capital letter first vs small letter first)
  - Propagation of comments
- At Common API implementation level (for DBus generation):
  - Refinement of the directory tree
  - One interface per file
  - Reserved names for Common API default methods

Directory tree



```
▾ hmi
  ▸ speechservice
    [redacted]
▾ navigation
  ▸ freetextsearchservice
  ▸ mapviewer
  ▾ navigationcore
      [redacted]
      Configuration.fidl
      Guidance.fidl
      LocationInput.fidl
      MapMatchedPosition.fidl
      NavigationCoreTypes.fidl
      Routing.fidl
      Session.fidl
  ▸ poiservice
    [redacted]
    NavigationTypes.fidl
  [redacted]
  [redacted]
  CommonTypes.fidl
```



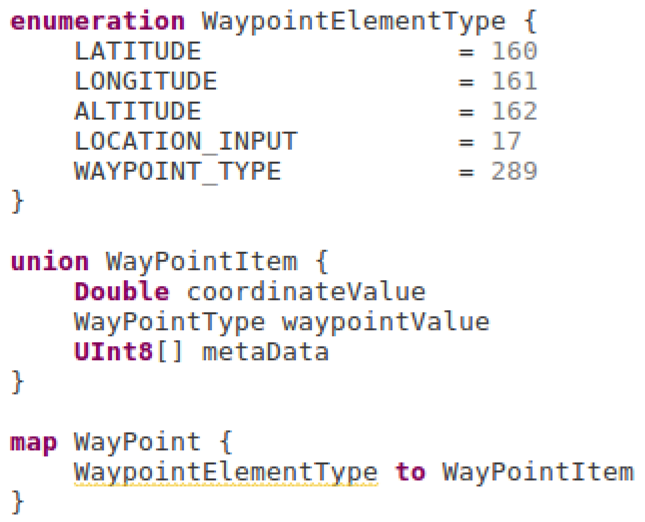
```
package org.genivi.navigation.navigationcore

import org.genivi.CommonTypes.* from "../../CommonTypes.fidl"
import org.genivi.navigation.NavigationTypes.* from "../NavigationTypes.fidl"
import org.genivi.navigation.navigationcore.NavigationCoreTypes.* from "NavigationCoreTypes.fidl"

<**
    @description : LocationInput = This interface offers functions that implement the location-input
**>

interface LocationInput {
    version {
        major 4
        minor 0
    }
```

Example of LocationInput.fidl

Reserved names

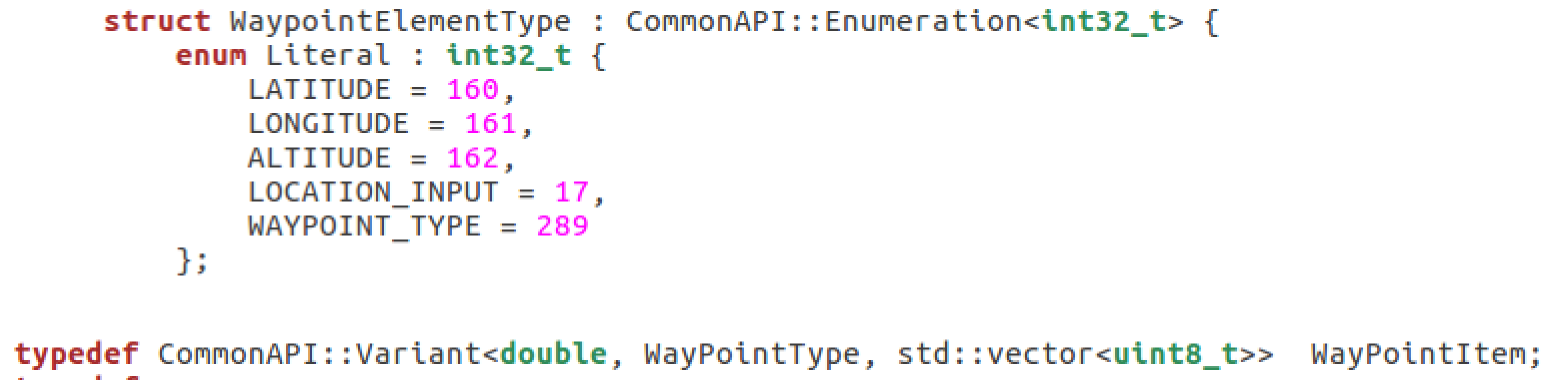
- In Common API:
  - ‘client’ is a reserved word, so it's not possible to use it for parameters, you got an error because of a conflicting declaration
  - For method names, the following ones are not allowed:
    - getAddress, isAvailable, isAvailableBlocking, getProxyStatusEvent, getInterfaceVersionAttribute
- Franca has also reserved names (see documentation)
- 'attributes' is a keyword in c++ --> result of Xtext check

Variant management in Common API DBus



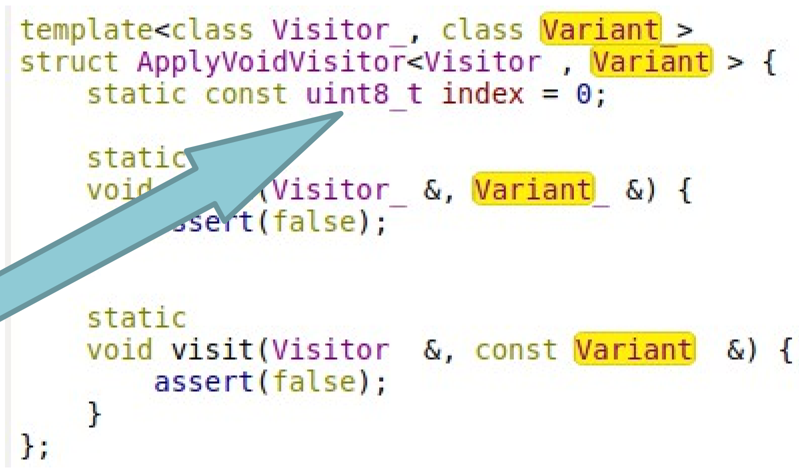
```
enumeration WaypointElementType {
    LATITUDE                = 160
    LONGITUDE               = 161
    ALTITUDE                = 162
    LOCATION_INPUT          = 17
    WAYPOINT_TYPE           = 289
}

union WayPointItem {
    Double coordinateValue
    WayPointType waypointValue
    UInt8[] metaData
}

map WayPoint {
    WaypointElementType to WayPointItem
}
```

Franca



```
struct WaypointElementType : CommonAPI::Enumeration<int32_t> {
    enum Literal : int32_t {
        LATITUDE = 160,
        LONGITUDE = 161,
        ALTITUDE = 162,
        LOCATION_INPUT = 17,
        WAYPOINT_TYPE = 289
    };

typedef CommonAPI::Variant<double, WayPointType, std::vector<uint8_t>>  WayPointItem;
```

Common API



```
template<class Visitor_, class Variant_>
struct ApplyVoidVisitor<Visitor_, Variant_> {
    static const uint8_t index = 0;

    static
    voi        (Visitor_ &, Variant_ &) {
             sert(false);

    static
    void visit(Visitor_ &, const Variant_ &) {
        assert(false);
    }
};
```

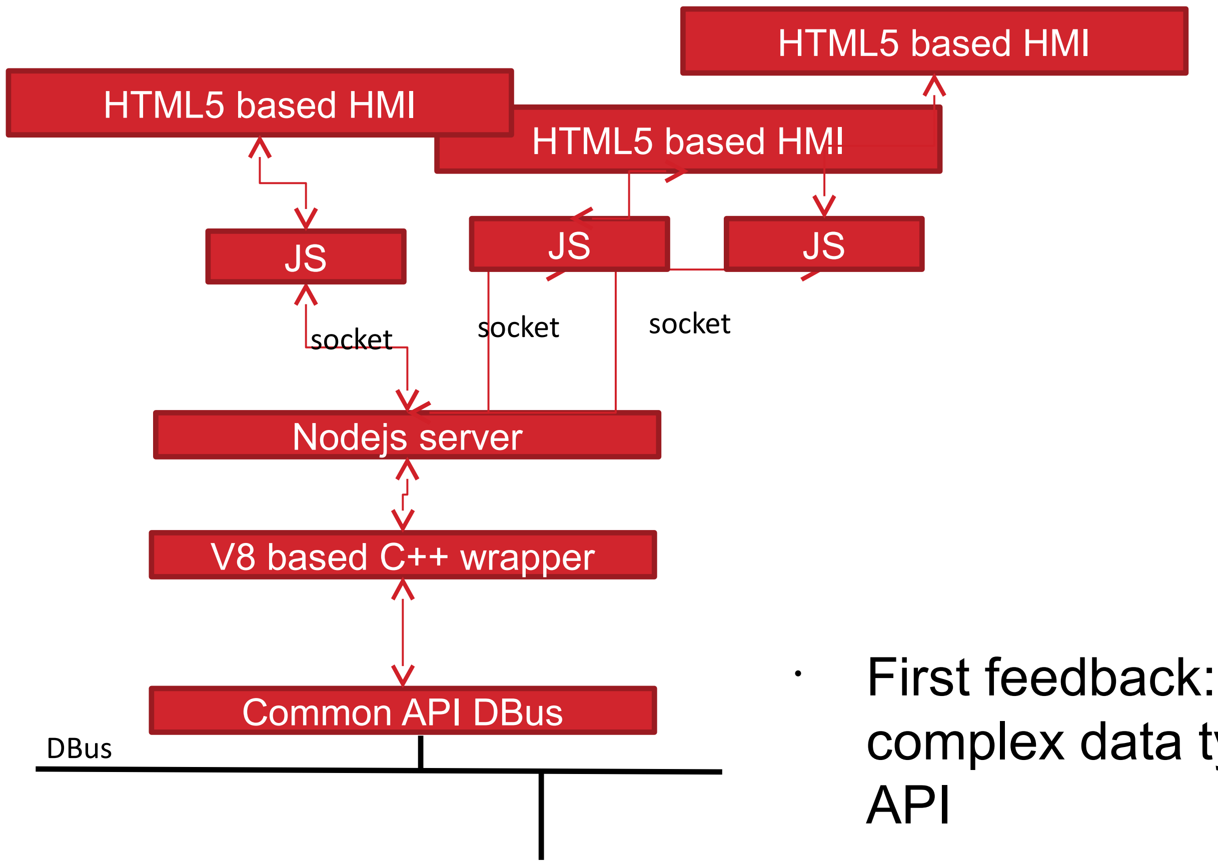
index

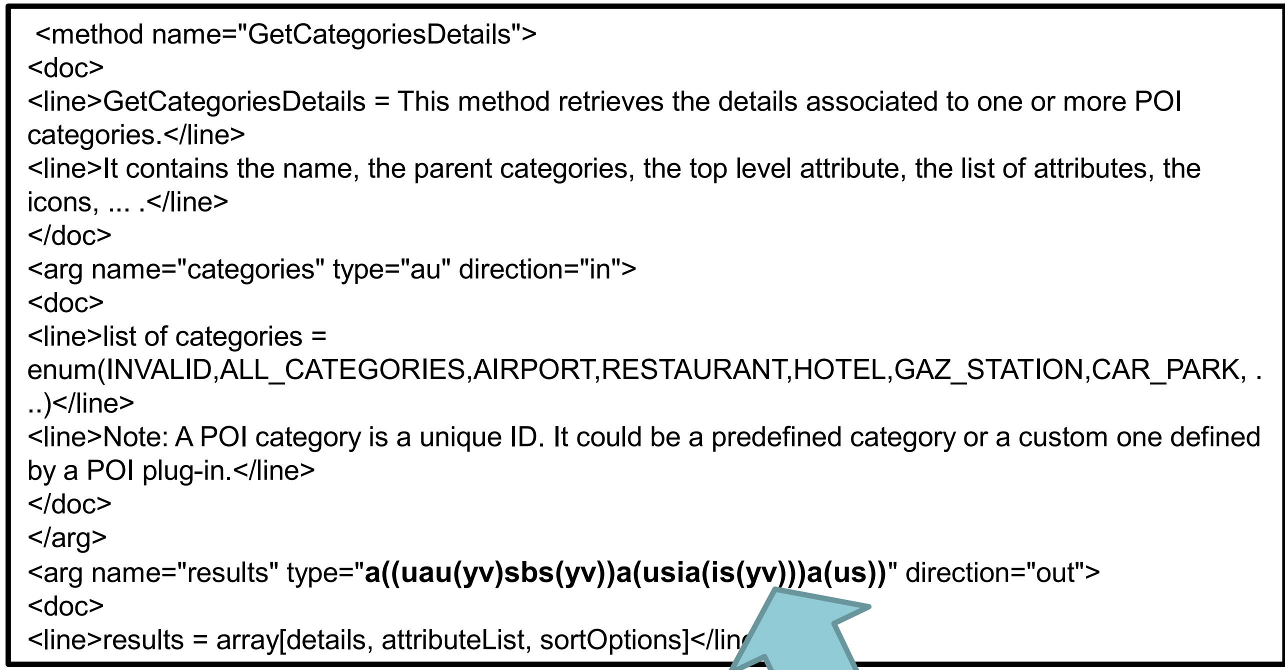
Web APIs, expected deliveries

Data types limitations, method call, Web IDL …

GENIVI®

First investigation: node.js based POC



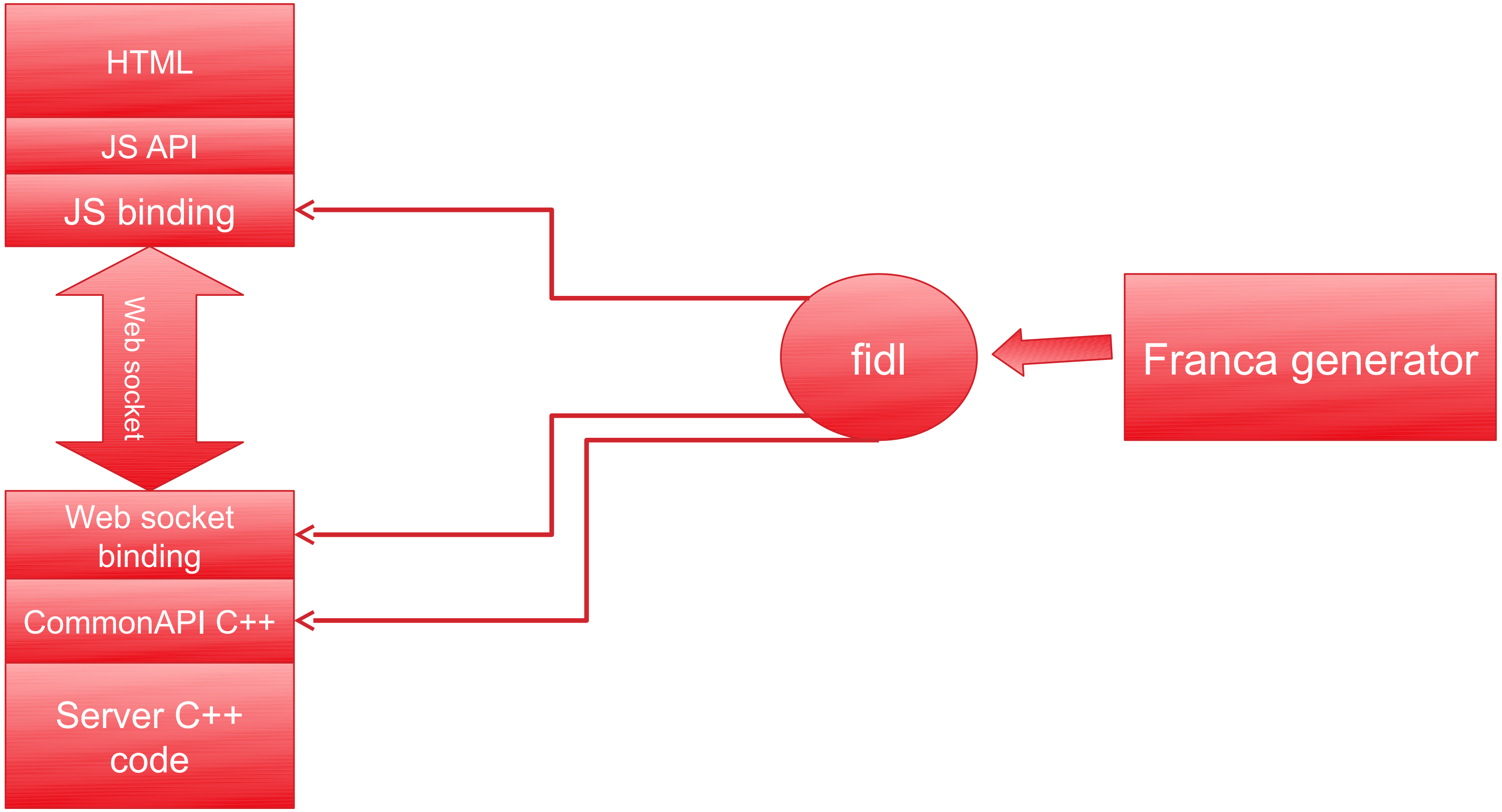


```
 <method name="GetCategoriesDetails">
<doc>
<line>GetCategoriesDetails = This method retrieves the details associated to one or more POI
categories.</line>
<line>It contains the name, the parent categories, the top level attribute, the list of attributes, the
icons, ... .</line>
</doc>
<arg name="categories" type="au" direction="in">
<doc>
<line>list of categories =
enum(INVALID,ALL_CATEGORIES,AIRPORT,RESTAURANT,HOTEL,GAZ_STATION,CAR_PARK, .
..)</line>
<line>Note: A POI category is a unique ID. It could be a predefined category or a custom one defined
by a POI plug-in.</line>
</doc>
</arg>
<arg name="results" type="a((uau(yv)sbs(yv))a(usia(is(yv)))a(us))" direction="out">
<doc>
<line>results = array[details, attributeList, sortOptions]</lin
```

Example of complex data type: Dbus signature extract

- First feedback: key point is to manage complex data types of the GENIVI API
- JSON encapsulation to serialize data could be a good solution

Franca to generate Web stuff


HTML
JS API
JS binding
Web socket
Web socket binding
CommonAPI C++
Server C++ code
fidl
Franca generator