

**GENIVI Alliance**

GENIVI Document CS00066

POIService

Component Specification

Accepted Version 2.0.0

**25-01-2017**

**Sponsored by:**

GENIVI Alliance

true

Accepted

**Abstract:**

This document provides the Component Specification for the POIService

**Keywords:**

POI, Search, Navigation

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.Copyright © 2017, trueBMW Car IT GmbH, Continental Automotive GmbH, PCA Peugeot Citroën, XS Embedded GmbH, TomTom International B.V., Alpine Electronics R&D Europe GmbH, AISIN AW CO.LTD.

All rights reserved.

The information within this document is the property of the true and its use and disclosure are restricted. Elements of GENIVI Alliance specifications may be subject to third party intellectual property rights, including without limitation, patent, copyright or trademark rights (and such third parties may or may not be members of GENIVI Alliance). GENIVI Alliance true not responsible and shall not be held responsible in any manner for identifying, failing to identify, or for securing proper access to or use of, any or all such third party intellectual property rights.true

“”“”

GENIVI and the GENIVI Logo are trademarks of GENIVI Alliance in the U.S. and/or other countries. Other company, brand and product names referred to in this document may be trademarks that are claimed as the property of their respective owners.

trueThis work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

http://creativecommons.org/licenses/by-sa/4.0

The above notice and this paragraph must be included on all copies of this document that are made.

GENIVI Alliance

2400 Camino Ramon, Suite 375

San Ramon, CA 94583, USA

1. Revision History

The following table shows the revision history for this document.

**Document Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Change** |
| 0.1 | 24 Oct 2012 | Marco Residori (XS Embedded) | Created empty template |
| 0.2 | 19 Dec 2012 | Marco Residori (XS Embedded) | Added architecture diagram.  Added sequence Diagrams.  Updated APIs,  Added Cedric Ranson’s use case descriptions. |
| 1.0 | 19 Dec 2012 | Marco Residori (XS Embedded) | Added architecture diagram.  Corrected typos. |
| 1.1 | 3 Feb 2017 | Philippe COLLIOT (PSA GROUPE) | Refine the document for Franca based API |

**Table of contents**

[1Introduction 5](#__RefHeading__44_1908500477)

[1.1System Overview 5](#__RefHeading___Toc1933_1592717581)

[1.2Component Overview 5](#__RefHeading___Toc1935_1592717581)

[2Terminology 6](#__RefHeading___Toc1937_1592717581)

[3Architecture 7](#__RefHeading__50_1908500477)

[3.1Interfaces 7](#__RefHeading__52_1908500477)

[3.2Interaction with other Components 8](#__RefHeading__54_1908500477)

[4API 9](#__RefHeading__58_1908500477)

[4.1Git Repository 9](#__RefHeading__60_1908500477)

[4.2Use Cases 9](#__RefHeading__64_1908500477)

[4.2.1POI Search 9](#__RefHeading___Toc1939_1592717581)

[4.2.1.1The HMI displays the list of POI categories 9](#__RefHeading___Toc1941_1592717581)

[4.2.1.2 The user searches for a restaurants around his/her position 11](#__RefHeading___Toc1943_1592717581)

[4.2.1.3The user is running out of fuel and therefore creates a proximity alert around his/her position to receive a sound alert in case a new fuel station is detected 13](#__RefHeading___Toc1945_1592717581)

[4.2.2POI Content Access 14](#__RefHeading___Toc1947_1592717581)

[4.2.2.1The user wants to get a list of car parks with free places around the current position sorted by price 14](#__RefHeading___Toc1949_1592717581)

[4.2.2.2The user wants to display on the map the location of all his/her Facebook friends around the vehicle position (the profile picture is used to mark a friend’s position) 16](#__RefHeading___Toc1951_1592717581)

[4.2.2.3The user wants to navigate to his/her doctor (the one listed in his/her phone contact book) if the office is open 17](#__RefHeading___Toc1953_1592717581)

[4.2.2.4The user wants to watch a movie. The user requests a list of movies projected in nearby cinemas with description, cinema name and price, all sorted by start time 18](#__RefHeading___Toc1955_1592717581)

[4.2.2.5The user creates a proximity alert along its route to receive a sound in case of dangerous area 19](#__RefHeading___Toc1957_1592717581)

[4.3Interfaces 21](#__RefHeading__114_1908500477)

# Introduction

This document describes the API of the POIService component.

## System Overview

The GENIVI Software Platform is a platform consisting of standardized middleware, application layer interfaces and frameworks defined or adopted by the GENIVI Alliance.

## Component Overview

The POIService is a software component of the above mentioned GENIVI Software Platform that offers POI search to client applications.

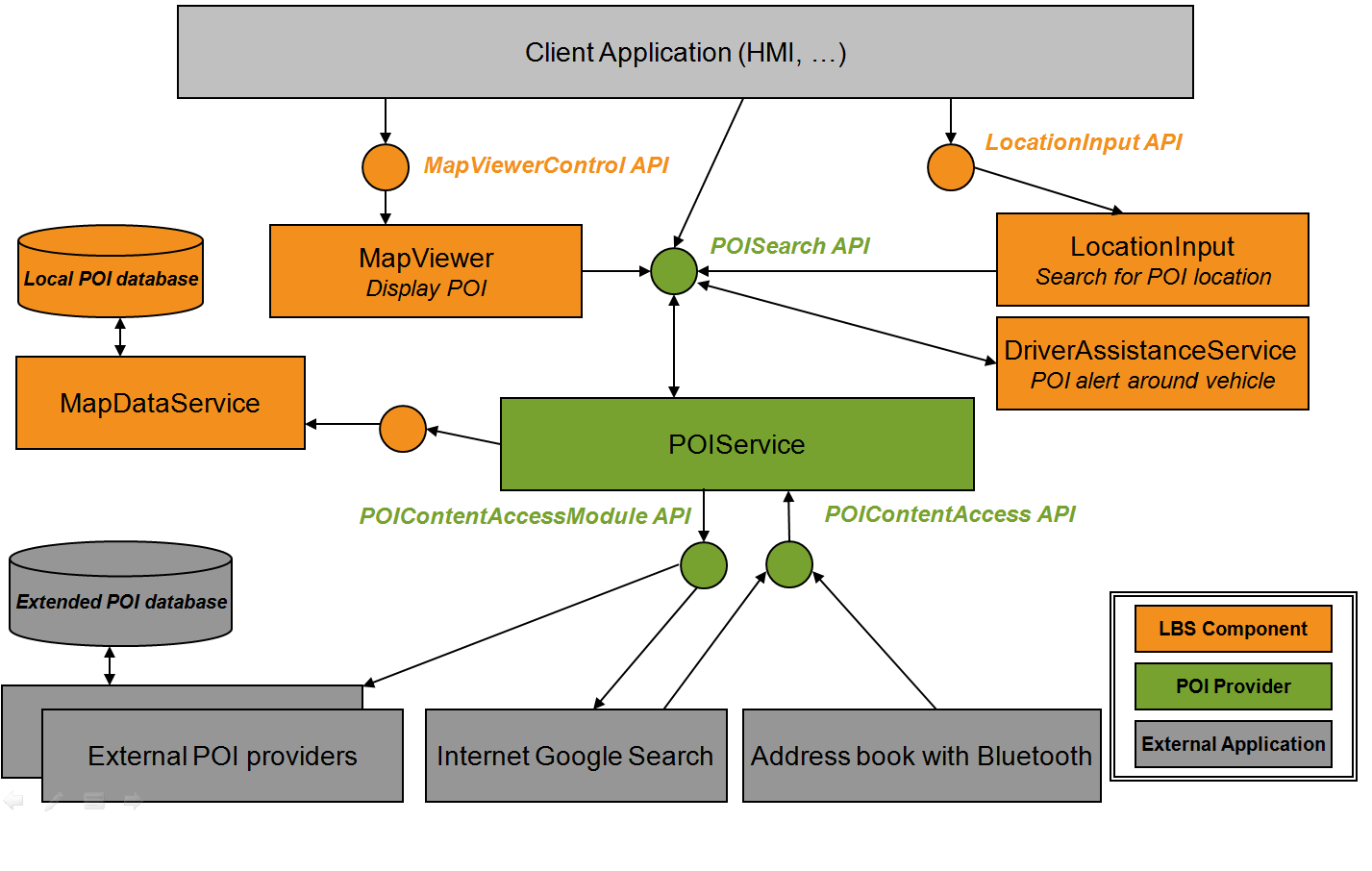
# Terminology

|  |  |
| --- | --- |
| ***Term*** | ***Description*** |
| POI | Point of Interest |
| CAM | Content Access Module |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Architecture

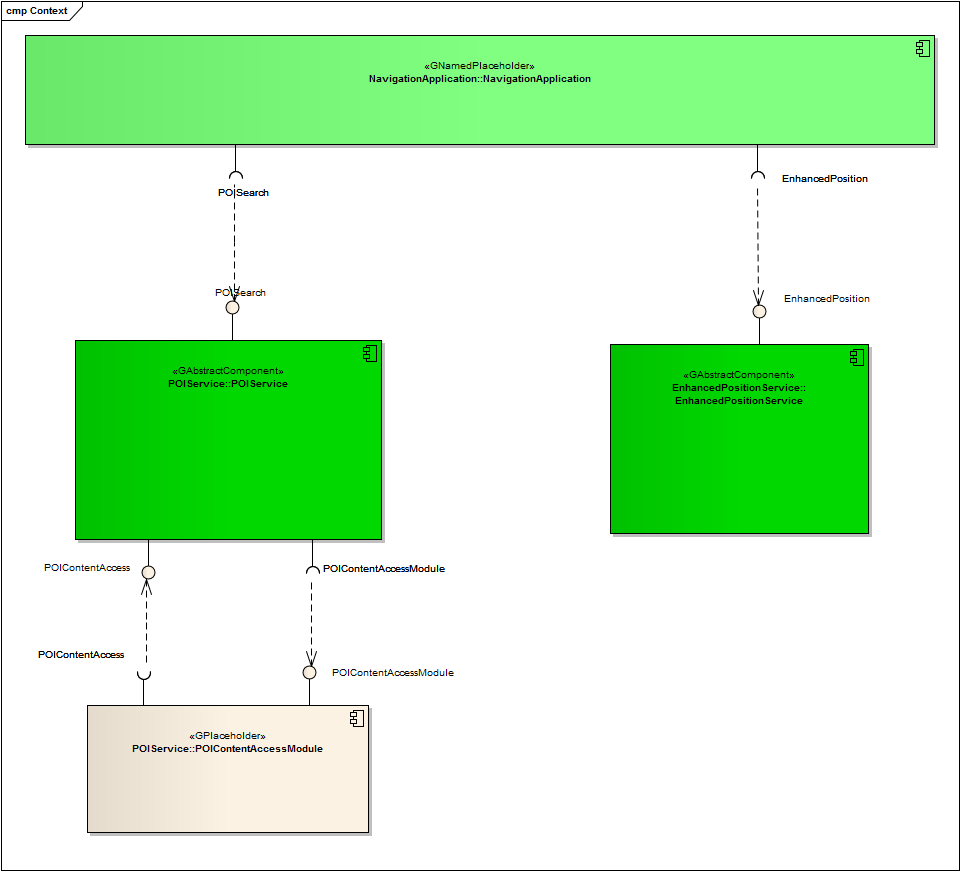
## Interfaces

* POISearch
* POIContentAccess
* POIContentAccessModule



## Interaction with other Components

## 



# API

## Git Repository

The POIService interfaces can be found in the GENIVI Git repository at:

<https://github.com/GENIVI/navigation/tree/master/api/franca/navigation/poiservice>

## Use Cases

### POI Search

#### The HMI displays the list of POI categories

**Description**

The HMI retrieves the root category from the component (should be ALL\_CATEGORIES).

The HMI retrieves the children categories of the root category.

For each child category, the HMI retrieves its children:

If there are no children, the HMI retrieves the category details from the component and displays its name as a searchable category.

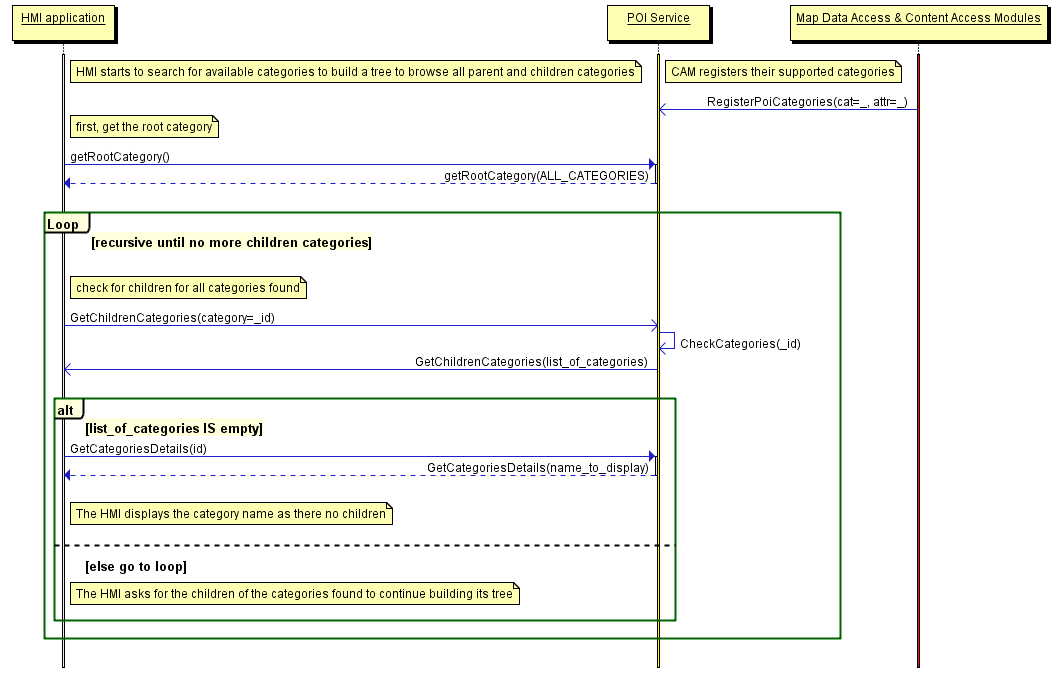
It can also display its visual icon.

For each child found, the HMI retrieves the children again and then displays the category name if there are no children.

**Error Management**

Root category is not available.

**Sequence diagram**



#### The user searches for a restaurants around his/her position

**Description**

The HMI retrieves the list of POI categories from the **search POI** API.

The HMI retrieves the vehicle position from the **enhanced position** API.

The HMI selects restaurant as a category and searches for POI around the vehicle position. It does not provide a search radius.

The POIService component defines a default radius for the category *restaurant*.

The POIService component searches for restaurant around the vehicle position within the local database.

The POIService component provides the list of restaurants sorted by distance. It also provides basic details.

The HMI browses the list and ask for detailed information for one of the POI.

The POIService component gets detailed information from the local database.

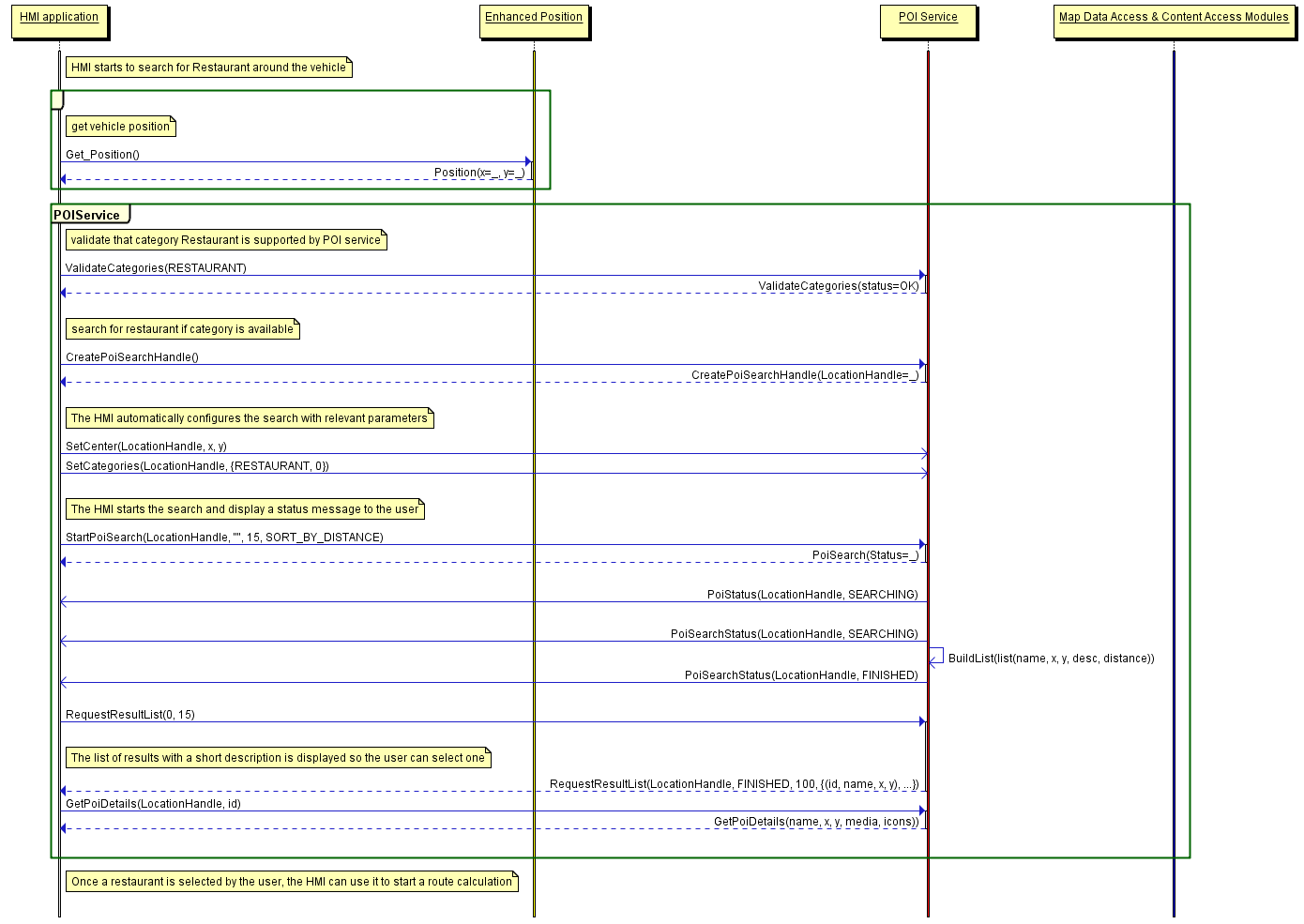
The HMI decides to drive to this restaurant.

The POIService component provides the location of the restaurant.

**Error Management**

If the local database access is down, the component returns an error.

**Sequence diagram**



#### The user is running out of fuel and therefore creates a proximity alert around his/her position to receive a sound alert in case a new fuel station is detected

**Preconditions**

The fuel station is a category that is supported by the local database.

**Description**

The HMI configures a new POI proximity alert for fuel station around the vehicle.

The HMI retrieves the vehicle position and the vehicle speed from the **enhanced position** API.

The POIService component defines the search radius of 500 meters.

The POIService component gets the list of fuel stations around the current position from the local database using **map data access** API.

A new list of POI is sent to the HMI with all the fuel station that match the alert configuration.

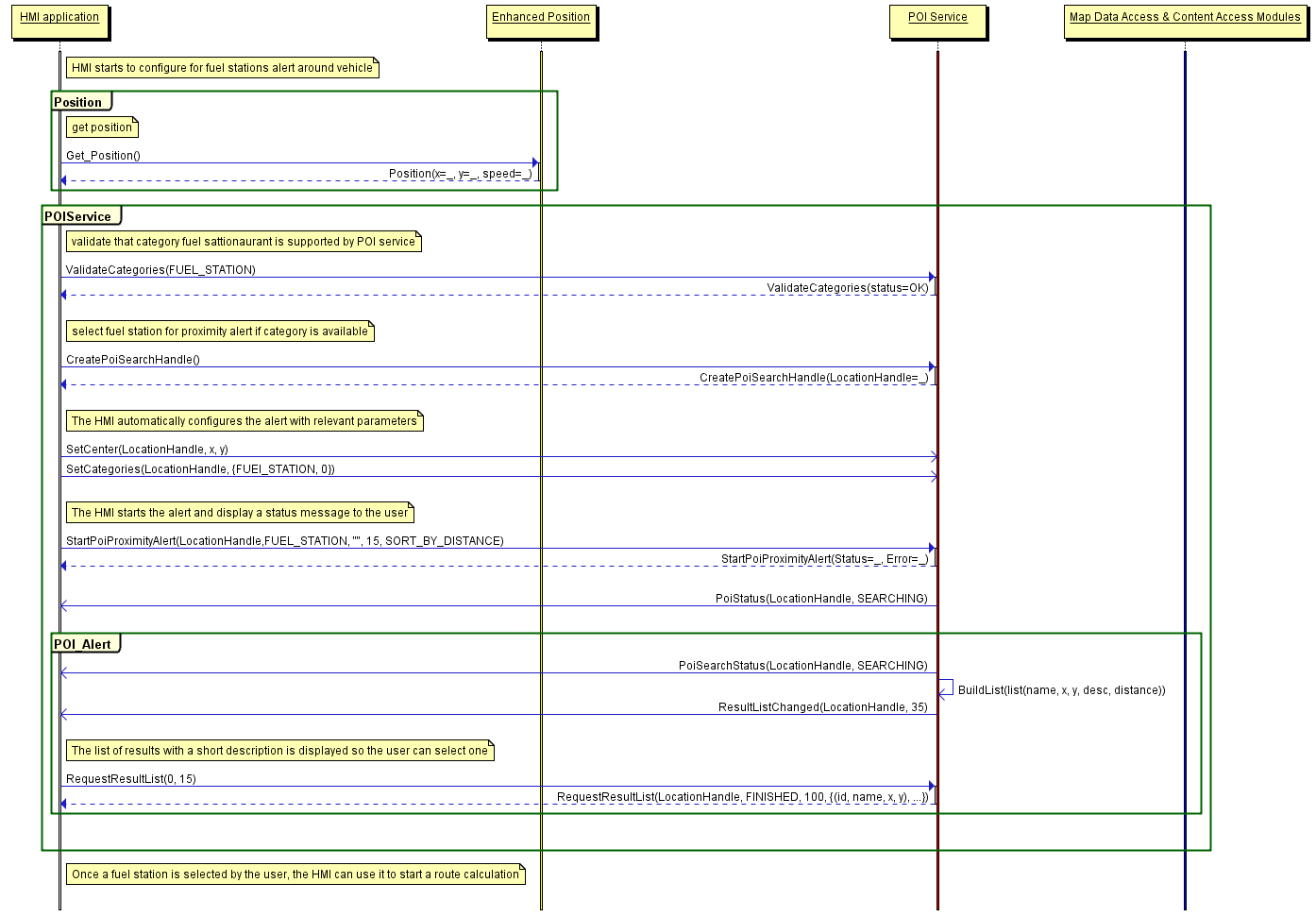
The HMI plays a sound when a new (non-empty) list of POI is received.

As soon as new fuel stations are available or current fuel stations are out the search radius, a new list is provided.

**Error Management**

If the local database access is down, the component returns an error.

**Sequence diagram**



### POI Content Access

#### The user wants to get a list of car parks with free places around the current position sorted by price

**Preconditions**

A plug-in was already installed in the POI Service component to access an identified online service that provides car park information (price and places left) from the name and coordinates of a car park.

**Description**

The HMI retrieves the vehicle position from the **enhanced position** API.

The HMI retrieves the list of car parks with free places sorted by price around the position from the **search POI** API.

The POIService component get the list of car parks around the current position from the local database using **map data access** API.

For each car park name and location found, the POI Service uses **POI content access** API to get free places available and the park price from the plug-in (described below).

The POIService component stores the result (all POI found with dynamic content) into the persistent storage to manage the ‘connection-down’ error. This content is deleted when it gets too old.

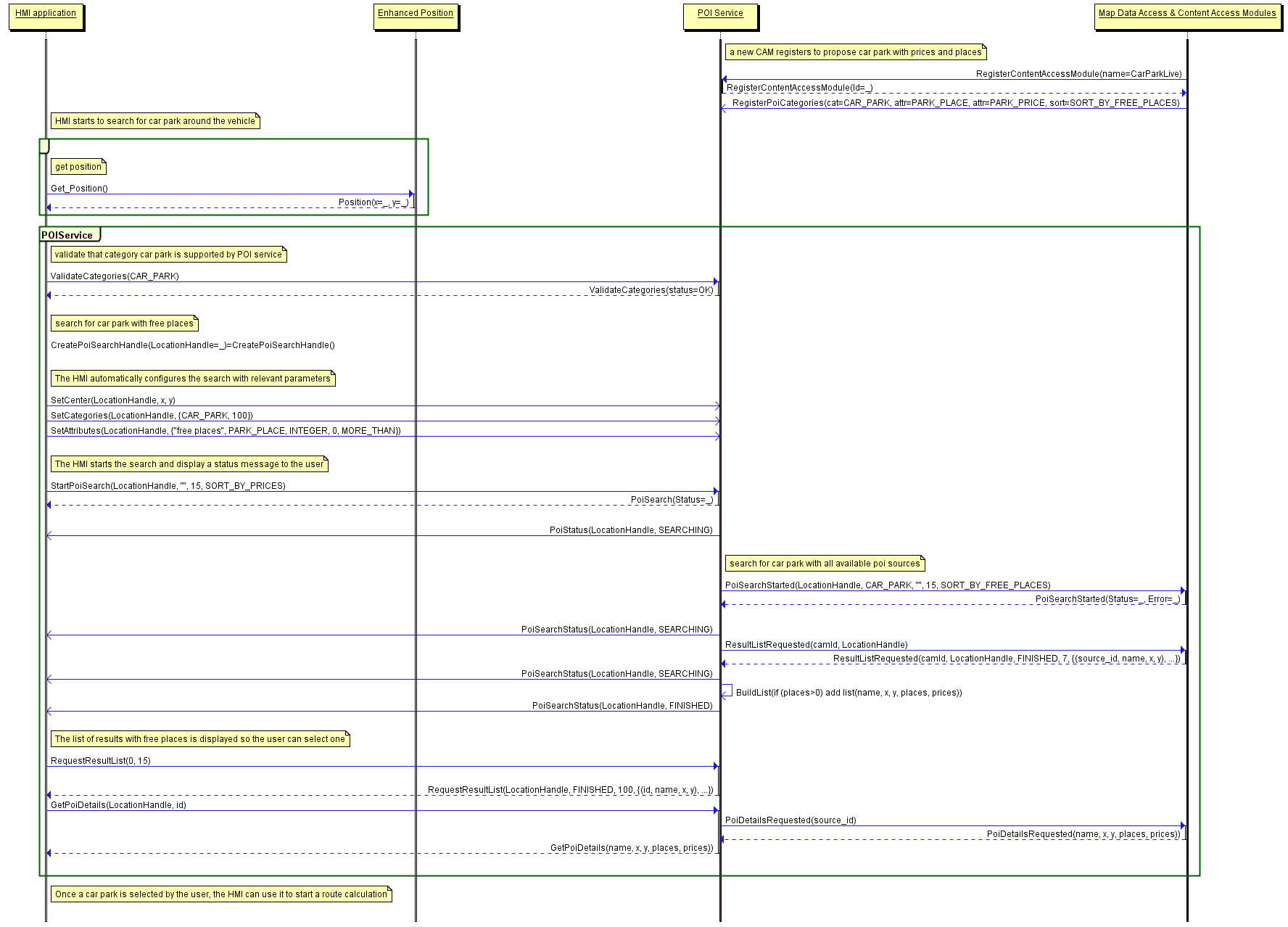
The POIService component updates each car park attribute with available places and sorts the list by price.

**Error Management**

If the local database access is down, the component returns an error.

If the network connection to the online provider is down, the content access module tries to retrieve latest content from its persistent storage (if any). If this content is too old (depends on CAM policy), it returns an error.

**Sequence diagram**



#### The user wants to display on the map the location of all his/her Facebook friends around the vehicle position (the profile picture is used to mark a friend’s position)

**Preconditions**

A plug-in was already installed in the POI Service component to access Facebook services and extract the profile picture and the current address from a list of friends.

This plug-in also registered a new POI category called **Facebook friends** and sub-category of predefined **User contact.**

The MapViewer API provides a way to create a custom location on the map with coordinates, name and icons.

**Description**

The HMI retrieves the vehicle position from the **enhanced position** API.

The HMI retrieves the list of Facebook friends from the **search POI** API around the vehicle position.

POI Service components gets the list of Facebook friends (with photo, name and coordinates) around the current position from the Facebook plug-in **POI content access** API.

The POI Service component stores the result into the persistent storage to manage the ‘connection-down’ error.

This content is deleted when it gets too old.

For each friend, the POI Service uses **MapViewer** API to update the map with icons.

**Error Management**

If the network connection to the online provider is down, the content access module tries to retrieve latest content from its persistent storage (if any). If this content is too old (1 hour), it returns an error.

#### The user wants to navigate to his/her doctor (the one listed in his/her phone contact book) if the office is open

**Preconditions**

A plug-in was already installed in the POI Service component to access a doctor's web site. The plug-in extracts the opening times and the doctor´s office address.

A plug-in was already installed in the POI Service component to access the user´s smartphone. The plug-in extracts a contact list from a group of names (friend, family, doctor, ...).

This plug-in created a new POI category called **My Phone Groups** with sub-categories like **friend** or **doctor.**

**Description**

The HMI retrieves the doctor´s office coordinates and opening times from the user's phone POI group doctor with the **search POI** API.

The POIService component gets the doctor´s office opening time and coordinates from the smartphone contacts plug-in using the **POI content access** API.

The HMI validates that the opening times match current time and start a route calculation.

The POIService component gets the doctor´s office coordinates from previously stored data.

The POIService uses **NavigationCore** API to start a route calculation from vehicle position to doctor's office.

**Error Management**

If the network connection to the online provider is down, the content access module returns an error.

#### The user wants to watch a movie. The user requests a list of movies projected in nearby cinemas with description, cinema name and price, all sorted by start time

**Preconditions**

A plug-in was already installed in the POI Service component to access an identified online service that provides cinema information (list of movies, opening times, prices and movies information like review and trailer link) from the name and coordinates of a cinema.

**Description**

The HMI retrieves the vehicle position from the **enhanced position** API.

The HMI retrieves the list of cinema with movies that starts in less than 1 hour around the current position from the **search POI** API. The list is sorted by start time.

The POIService component gets the list of cinemas around the current position from the local database using the **map data access** API.

For each cinema name and location found, the POI Service uses **POI content access** API to get list of movies available with start time, movie information and price from the plug-in (described below).

The HMI display the list of movies sorted by start time (the sooner the first) with a quick description, a review, the price and the cinema name.

The rear seat HMI downloads a video from movie trailer link and displays it.

Depending upon the selected film, the POIServicecomponent gets the cinema coordinates from previously stored data.

The POIService uses **NavigationCore** API to start a route calculation from vehicle position to the cinema.

**Error Management**

If the local database access is down, the component returns an error

If the network connection to the online provider is down, the content access module tries to retrieve latest content from its persistent storage (if any). If this content is too old (20 minutes), it returns an error.

#### The user creates a proximity alert along its route to receive a sound in case of dangerous area

**Preconditions**

A plug-in was already installed in the POI Service component to access an identified online service that provides dangerous area content (road conditions, weather, traffic jam, men at work) around a location.

The dangerous area is a category that is supported by the local database.

**Description**

The HMI configures a new POI proximity alert for dangerous events in front of the vehicle.

The HMI retrieves the vehicle position and speed from the **enhanced position** API.

The POI component defines the search radius in front of vehicle position depending on the category and the vehicle speed.

The POIService component gets the list of dangerous areas around the current position from the local database using the **map data access** API.

The POIService component gets the list of dangerous events around the position from content access module according to its defined radius and criteria.

The POIService component updates local events with dynamic contents if necessary.

A new list of POI is sent to the HMI with all the events (from local and from CAM) that match the alert configuration.

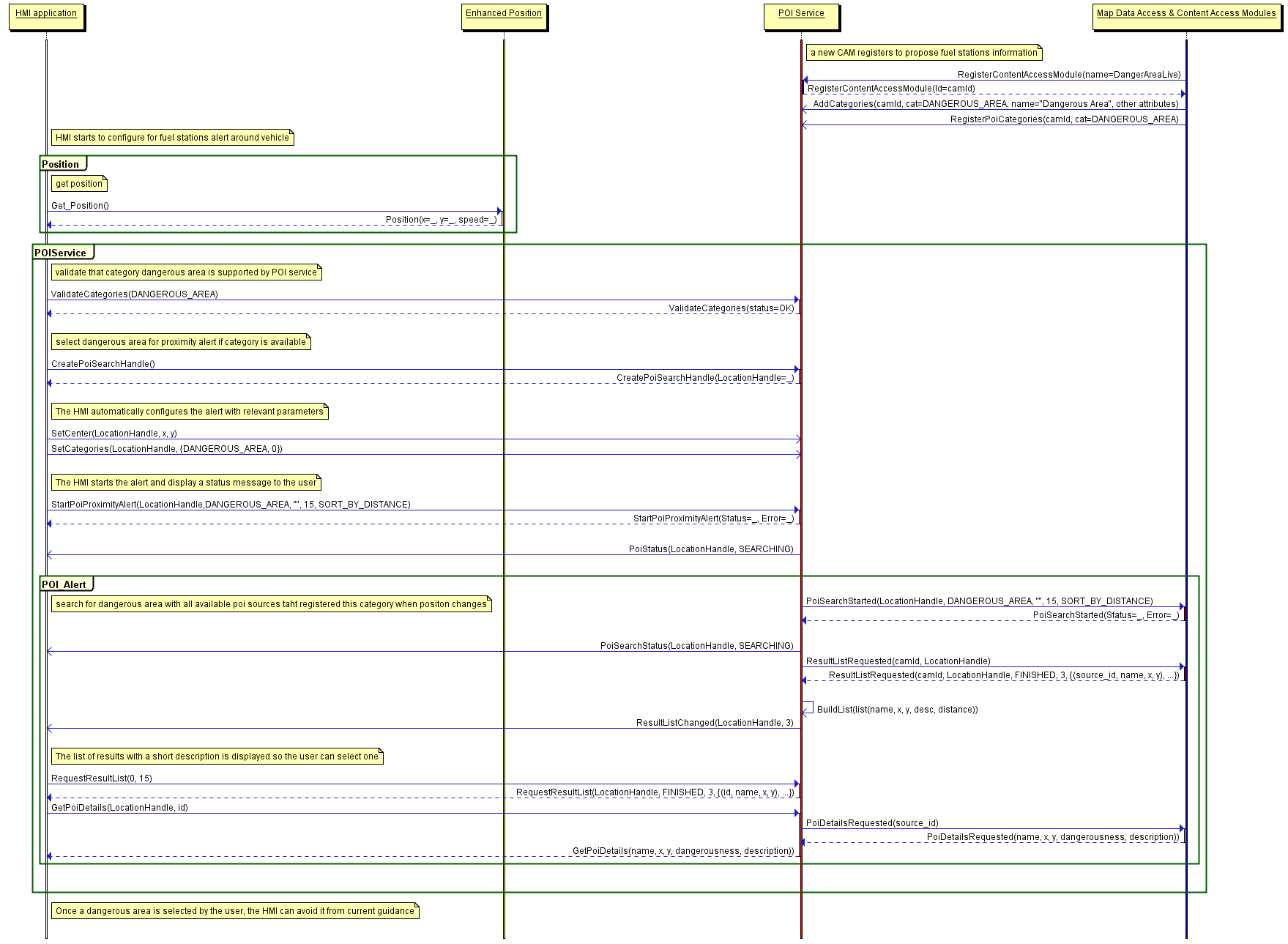
The HMI plays a sound when a new (non-empty) list of POI is received.

**Error Management**

If the local database access is down, the component returns an error.

If the network connection to the online provider is down, the content access module tries to retrieve latest content from its persistent storage (if any). If this content is too old (more than 2 minutes), it returns an error.

**Sequence diagram**



## Interfaces

The following pages describe the interfaces of the POIService API.