

# **GENIVI** Alliance

# **GENIVI Document CS00063**

## POIService

**Component Specification** 

# Accepted Version 2.0

25-01-2017

Sponsored by: GENIVI Alliance

Y Accepted

Y2013GENIVI Alliance. 2400 Camino Ramon, Suite 375, San Ramon, CA 94583, USA http://www.genivi.orgY

#### Abstract:

# **Keywords:** Y

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

1 Change History	.4
2 Introduction	.5
3 Terminology	.6
4 Architecture	.7
4.1 Interfaces	.7
4.2 Interaction with other Components	.8
5 API	.9
5.1 D-Bus	.9
5.2 Git Repository	.9
5.3 Naming Conventions	.9
5.4 Data Types Convention1	0
5.5 Errors	1
5.6 Use Cases1	.2
5.6.1 POI Search12	
5.6.1.1 The HMI displays the list of POI categories12	
5.6.1.2 The user searches for a restaurants around his/her position	
5.6.1.3 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 detected14	
5.6.2 POI Content Access15	
5.6.2.1 The user wants to get a list of car parks with free places around the current	
position sorted by price15	
5.6.2.2 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)16	
5.6.2.3 The user wants to navigate to his/her doctor (the one listed in his/her phone conta	ct
book) if the office is open17	
5.6.2.4 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 time18	
5.6.2.5 The user creates a proximity alert along its route to receive a sound in case of	
dangerous area19	
5.7 Interfaces	20

### **Table of contents**

# **1** Change History

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

## 2 Introduction

This document describes the API of the POIService component.

# 3 Terminology

Term	Description
POI	Point of Interest
CAM	Content Access Module

## 4 Architecture

### 4.1 Interfaces

- POISearch
- POIContentAccess
- POIContentAccessModule



Page 7

### 4.2 Interaction with other Components



Y2013 YThis work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.For GENIVI Members only.

Page 8

## 5 API

### 5.1 Git Repository

The POIService interfaces can be found in the GENIVI Git repository at: <u>https://github.com/GENIVI/navigation/tree/master/api/franca/navigation/poiservice</u>

### 5.2 Use Cases

### 5.2.1 POI Search

### 5.2.1.1 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.

POIS	ervice	Map Data Access & Content Access Mod
HMI starts to search for available categories to build a tree to browse all parent and children categories	CAM registers their	supported categories
getRootCategory()getRootCategory(ALL_CATEGORIES)		
[recursive until no more children categories]         check for children for all categories found         GetChildrenCategories(category=_id)         GetChildrenCategories(list_of_categories)	CheckCategories	(_id)
alt       [list_of_categories IS empty]         GetCategoriesDetails(id)	-	

### 5.2.1.2 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



Page 11

<sup>5.2.1.3</sup> 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.

-IMI application	anced Position	POI Service	Map Data Access & Content Access Modules
HMI starts to configure for fuel stations alert around vehicle			
Position get position Get_Position()			
Position(r=_, y=spee			
validate that category fuel sattionaurant is supported by POI service			
ValidateCategories(FUEL_STATION)	ValidateCategories(sta	tus=0K)	
select fuel station for proximity alert if category is available			
CreatePoiSearchHandle()	CreatePoiSearchHandle(LocationHa	andle=_)	
The HMI automatically configures the alert with relevant parameters			
SetCenter(LocationHandle, x, y) SetCategories(LocationHandle, (FUEL_STATION, 0))		>	
The HMI starts the alert and display a status message to the user StartPoProvimitAlert() ocationHandle FUEL_STATION = 15_SORT_RY_DISTANCE)			
······	StartPoiProximityAlert(Status=_,		
	PoiStatus(LocationHandle, SEAR	CHING)	
POI Alert	PoiSearchStatus(LocationHandle, SEAR	CHING)	
	ResultListChanged(LocationHar	ndle, 35)	e, x, y, desc, distance))
The list of results with a short description is displayed so the user can select one			
HequestResultList(0, 15)	RequestResultList(LocationHandle, FINISHED, 100, {(id, name,	x, y),)	
Once a fuel station is selected by the user, the HMI can use it to start a route calculation			

### **5.2.2 POI Content Access**

# 5.2.2.1 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 'connectiondown' 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.

Enhanced	<u>POI</u>	Map Data Access & Content Ac
[HMI starts to search for car park around the vehicle]		In ever CAM registers to propose car park with prices and places           RegisterContentAccessModule(hams=CarParkLive)           RegisterContentAccessModule(hams=CarParkLive)           Register/OntentAccessModule(hams=CarParkLive)           Register/OntentAccessModule(hams=CarParkLive)           Register/OntentAccessModule(hams=CarParkLive)           Register/OntentAccessModule(hams=CarParkLive)
get position Get_Position( Position(c=, y=) POISeptice		
	ValidateCategories(status=OK)	
CreatePoiSearchHandle(LocationHandle=_)=CreatePoiSearchHandle() The HMI automatically configures the search with relevant parameters SetCenter(LocationHandle, x, y)		»
SetCategones(LocationHandle, (CAR_/PARK, 100)) SetMithibudes(LocationHandle, (CAR_PARK, 100)) The HMI starts the search and display a status message to the user StartPolSearch(LocationHandle, '', 15, SORT_BY_PRICES)	Daldaank Oktoor	
•	PoiStatus(LocationHandle, SEARCHING)	search for car park with all available pol sources
<	PolSearchStatus(LocationHandle, SEARCHING)	PolSearchStarted(Starte=_Error_)  ResultLsRequested(camid_LocationHandle)  ResultLsRequested(camid_LocationHandle)  ResultLsRequested(camid_LocationHandle, FNISHED, 7, ((source_id, name, x, y),))
C	PolSearchStatus(LocationHandle, SEARCHING) PolSearchStatus(LocationHandle, FINISHED)	BuildList(if (places-0) add list(name, x, y, places, prices))
RequestResultList(0, 15) GelPolDetails(LocationHandle, Id)	ReguestResultListLocationHandle, FINISHED, 100, ((id, name, x, y),))	PolDetailsRequested(source_id)
Once a car park is selected by the user, the HMI can use it to start a route calculation	GelPolDetalls(name, x, y, places, prices))	PolDetailsRequested(name, x.y.places, prices))

5.2.2 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.

# 5.2.2.3 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.

### 5.2.2.4 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.

# 5.2.5 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.

Il application	ced Position	Map Data Access &	Content Access Modul
HMI starts to configure for fuel stations alert around vehicle Position		a new CAM registers to propose fuel stations information RegisterContentAccessModule(de-camid) AddCategories(camid, cateDANGEROUS_AREA, name=DangerAreaLive AddCategories(camid, cateDANGEROUS_AREA, name=Cangerous Area", other attributes; RegisterPolCategories(camid, cateDANGEROUS_AREA	<u>}</u> →
POIService Validate That category dangerous area is supported by POI service ValidateCategories(DANGEROUS_AREA)	ValidateCategories(status=OI)	1	
Select dangerous area for provimity alert if category is available CreatePolSearchHandle()	CreatePoiSearchHandle(LocationHandle=	1	
SetCateportes(LocationHandle, X, P) SetCateportes(LocationHandle, (DANGEROUS_AREA.0)) The HMI starts the alert and display a status message to the user StartPoiProximityAlert(LocationHandle,DANGEROUS_AREA.", 15, SORT_BY_DISTANCE)	StarPoProsimibAlert(Status=_Error=_	* → 1	
POI Alert J Search for dangerous area with all available poi sources taht registered this category when positon changes	PolStatus(LocationHandle, SEARCHING)	PolSearchStarted(LocationHandle, DANGEROUS_AREA."', 15, SORT_BY_DISTANCE)	1
K	PolSearchStatus(LocationHandle, SEARCHING)	ResultListRequested(camid_LocationHandle) ResultListRequested(camid_LocationHandle_FINISHED_3_((source_id_name_ix_y)) BuildList(list(name, x_y, desc, distance))	
The list of results with a short description is displayed so the user can select one RequestResultList(0, 15) GetPoDetails(LocationHandle, Id)	result.ist.nagee(LocationHande, .)		
4		PolDetailsRequested(name, x, y, dangerousness, description)	
Once a dangerous area is selected by the user, the HMI can avoid it from current guidance			

### 5.3 Interfaces

The following pages describe the interfaces of the POIService API.