PEACOX – Persuasive Advisor for CO2-reducing cross-modal trip planning

Project Reference: 288466

FP7-ICT 2011: 6.6 Low carbon multi-modal mobility and freight transport

Project Duration: 1 Oct 2011 – 30 Sept 2014

D6.2 Server and Client Implementation

[FLU]

Author(s)

Mirjana Artukovic (FLU)

Bilal Abunaim (FLU)

Lara Montini (ETHZ)

Efthimios Bothos (ICCS)

Tomáš Tvrtzký (TMX)

Aonghus McNabola (TCD)

Date: 26/04/2013

Dissemination level: (PU, PP, RE, CO): PU

Project Co-Funded by the European Commission within the 7th Framework Program

Table of Contents

1. Introduction 4

1.1 Background 4

1.1.1 Scope of this Deliverable 4

2. System Architecture 5

2.1 Overview 5

3. Clients Description 7

3.1 Journey Planner Application (FLU) 7

3.1.1 Description 7

3.1.2 Features 7

3.1.3 Architecture Components 7

3.2 Navigation Client (TMX) 8

3.2.1 Description 8

3.2.2 Features 8

3.2.3 Architecture Components 9

4. Server Component Description 10

4.1 Application Server (FLU) 10

4.1.1 Description 10

4.1.2 Features 10

4.1.3 Architecture Components 11

4.2 ICCS Client – Recommendation Service 12

4.2.1 Description 12

4.2.2 Features 12

4.2.3 Architecture Components 13

4.3 TCD Client – Emission and Exposure Model 14

4.3.1 Description 14

4.3.2 Features 14

4.3.3 Architecture Components 15

4.3.4 Emission and Exposure algorithms for TCD Application Server 17

4.4 ETHZ Client – Travel Mode and Trip Purpose Detection 21

4.4.1 Description 21

4.4.2 Features 21

4.4.3 Architecture Components 22

5. Processes 23

5.1 Route Calculation 23

5.2 GPS & Accelerometer Data Logging 24

5.3 Real-time Trip Mode Detection 24

6. Summary and Outlook 26

7. Figures 27

8. List of tables 27

1. Introduction

1.1 Background

The objective of this work package is to define, design and implement the PEACOX applications (journey planning application and navigation client). In the first step, the use cases for the PEACOX applications will be defined, representing the basis for the development phase of the PEACOX applications.

Furthermore, the overall system architecture will be finalised, as well as the definition of all system components. Another important aspect of this WP is the interface design of the PEACOX applications. Within this WP the interaction design as well as the visual design of the PEACOX applications will be finalised.

The last step of the workpackage will be the development of the PEACOX applications (journey planner application and navigation client) for the planned trials (Vienna, Dublin).

1.1.1 Scope of this Deliverable

In the first chapter the overall system architecture will be presented, helping to understand the separation of the system components and how they interact with each other.

Afterwards, the PEACOX applications, journey planner application and navigation client, will be explained in detail. The last chapter deals with the description of the server components and their requirements.

2. System Architecture

2.1 Overview

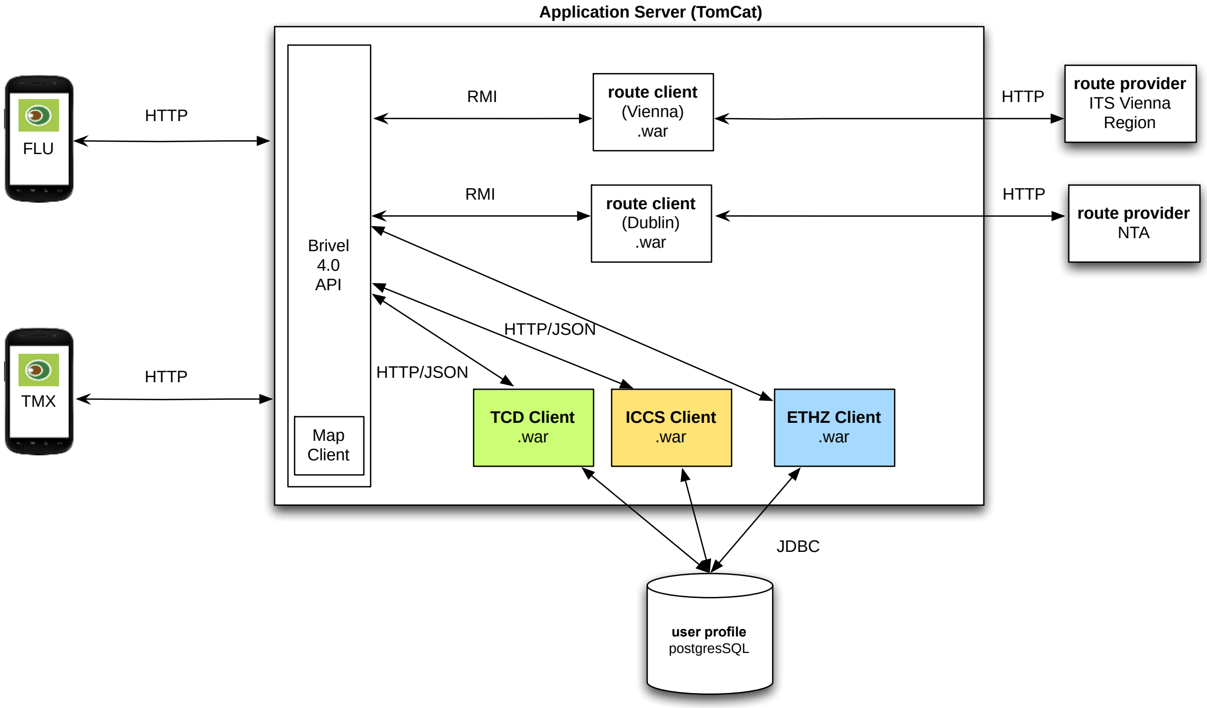


Figure 2.1: Overall system architecture

Figure 2.1 represent the overall system architecture of PEACOX.

There exist two client components, journey planner application and the navigation client. The journey planner application, provided by FLU, is connected to the ITS Vienna Region / NTA routing engine in order to receive routing results. The navigation client, provided by TMX, uses also the ITS Vienna Region / NTA routing engine in order to receive multimodal routing results. Therefore, FLU will provide TMX a route API (part of the brivel 4.0 API).

The application server represents a central component within the architecture, covering the main business logic that is necessary within the PEACOX system. The application server will be explained in detail in chapter 4.1.

Additionally, all other components will be integrated: ICCS Client (recommendation component), ETHZ client (travel mode detection and trip mode detection) and TCD client (emission model and exposure model) will be integrated within the application server.

In case of the TCD client and the ICCS client JSON (Java Script Object Nation) will be used as data-interchange format. The benefits of this format are: is easy to read for humans and it is easy for machines to pare and generate. Furthermore, JSON enables simple data structures because of the used lightweight text data interchange format.

All partner components will be provided as .war files and will be deployed within the application server. The deployment process will be defined together with all partners, depending on the development process of the individual partners.

Each partner will be connected to the central database, the user profile database. During the development phase the database will be hosted by ICCS. Afterwards, the dump of the final version will be also hosted n the application server.

Server components are free to JDBC to connect to the database and manage the database entities manually, or to use Hibernate framework to take the advantages provided by Hibernate framework.

3. Clients Description

3.1 Journey Planner Application (FLU)

3.1.1 Description

The journey planner represents one of the PEACOX applications. As already defined within the D2.3 Stakeholder and Technological Requirements, an Android application (platform 4.0+) will be developed.

3.1.2 Features

The functionalities that are a part of the PEACOX application are already defined within the D6.1 Use cases Report. The following use cases (UC) were defined:

- UC1 Route Request
- UC2 Context Aware and Personalised Routing Results (information before travelling)
- UC3 Live information (information during travelling)
- UC4 Logbook (information after travelling)
- UC5 Personal challenge and rewards
- UC6 Group challenge and rewards
- UC7 Triggering of the navigation client

3.1.3 Architecture Components

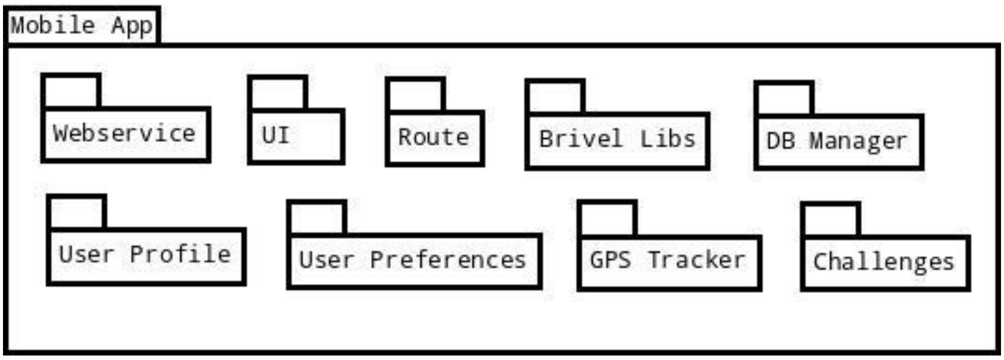


Figure 3.1: Client architecture – journey planner application

Description of architecture components

- Webservice: this package contains all classes required to send and receive data from/to the server.
- UI: this package contains all the elements related to the User interface.
- Route: The component is the route client. This client constructs the route request and displays the response from the server.
- Brivel Libs: the libraries and utilities provided by Brivel framework.
- DB Manager: This package is responsible for database management and data storage.
- User Profile: This package is used to manage the profile of the user.
- User preferences: This package is used to manage the preferences of the user.
- GPS Tracker: the component is used to collect the GPS points and accelerometer data and upload them to the server.
- Challenges: This component is used to inform the user about the challenges he/she has to accomplish, and facilitate sharing them in social networks.

3.2 Navigation Client (TMX)

3.2.1 Description

Dynavix PEACOX is a multimodal navigation application for Smartphones based on Android. It allows users to search for routes to their desired destination and select their transportation mode (walking, public transport, car)

3.2.2 Features

- Selecting destination – Users can choose their destination by address, favourite location, recent location, google search, POI or coordinates
- Transportation mode – When the destination is selected, users can choose their transportation mode. Three options are available – walk, public transport and car. The Car and walk routes are calculated inside Dynavix application. When public transport is selected, the application sends a query to a server providing a journey planner. This server delivers a list of public transport options as well as points of

travel mode changes (public transport stops). The server also provides the number of points which are used to plot routes for public transport.

3.2.3 Architecture Components

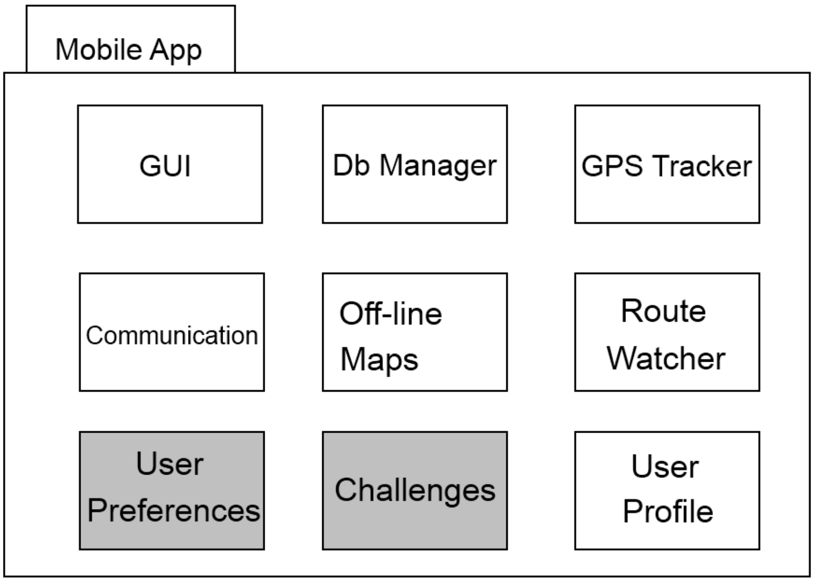


Figure 3.2: Client architecture – navigation client

Description of architecture components

- GUI: this package contains all the elements related to the graphical user interface.
- Db Manager handles all requirements for data reading and saving inside the mobile application
- GPS Tracker loggs the user's position and other relevant data
- The communication part is responsible for data exchange with the PEACOX system server
- Off-line maps are stored inside the mobile application for being used by car and walking navigation
- A route watcher handles rules for mapmatching and route recalculation

- User profiles store user's preferences for navigation
- User preferences and challenges are not fully developed yet.

4. Server Component Description

4.1 Application Server (FLU)

4.1.1 Description

The PEACOX server is a central service that is responsible for the communication and data transfer between the PEACOX client and the different services that are implemented for PEACOX.

The server adopts a distributed computing architecture depicted in chapter 4.1.3.

4.1.2 Features

The business logic of the application server will cover the following points:

- Provide an interface between the mobile application and the different services required for PEACOX.
- Manage route requests.
- Manage user profiles.
- Manage user tracks.
- Manage traffic and real-time information.
- Manage the interaction between different system components.
- Manage recommendations.
- Manage challenges.
- Manage CO_2 emissions / exposure information.
- Integrate additional information with the route.
- Manage the database.
- Manage GPS logging and accelerometer data
- Manage connection to Facebook API

4.1.3 Architecture Components

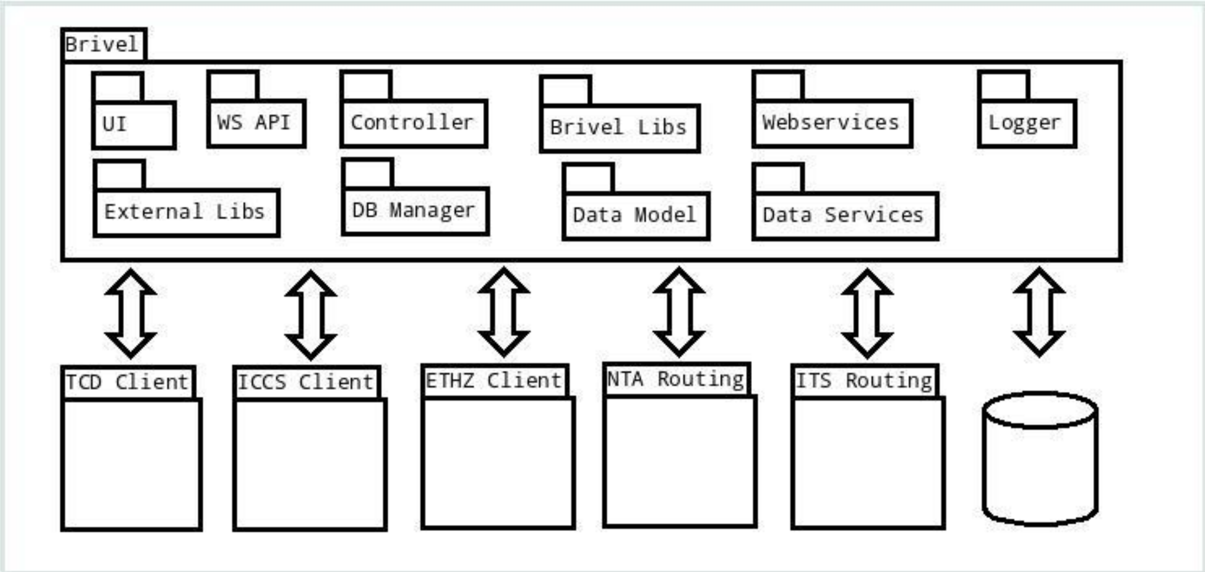


Figure 4.1: Server architecture – application server

Description of Architecture Components

- UI: The package contains the views that could be accessed using a normal web browser (service page).
- WS API: This package contains all the web services APIs that the client requires to connect with the server and exchange the data.
- Controller: This package contains the controller classes that control the data flow between the views and the data model.
- Brivel Libs: The libraries provided by framework of FLU "Brivel".
- Webservices: This package provides an interface to the web services required for PEACOX server.
- External Libs: The external libraries and utilities used in the application.
- DB Manager: The DB Manager is responsible for reading and writing the data to the Database.
- Data Model: This package contains all the data model entities.

- Data Service: this package provides classes that contain the business logic that is required to handle the data.
- Logger: a logging utility (GPS and accelerometer data).

4.2 ICCS Client – Recommendation Service

4.2.1 Description

The aim of the recommendation service is twofold. First the service personalizes routing queries in order to filter and select routing results that are within the limits of the users' preferences and match their individual characteristics (e.g. users who do not own a car will not receive results that involve the use of a car). Second, the recommendation service incorporates 'nudges' in the route selection process [for more details see PEACOX D2.3 Stakeholder and Technological Requirement], in order to highlight the personalized, yet environmentally friendliest, routes to users.

4.2.2 Features

The business logic of the recommendation service will cover the following points:

- Personalize routing queries according to user profile information stored in the PEACOX database and user preferences provided by the user before planning a trip.
- Contextualize routing queries according to weather information.
- Rank the available routes to be displayed to the user according to user preferences, user characteristics and route information (including emissions).
- Mine and identify user preferences based on GPS logs, usage of the PEACOX application and used modes of transport.

4.2.3 Architecture Components

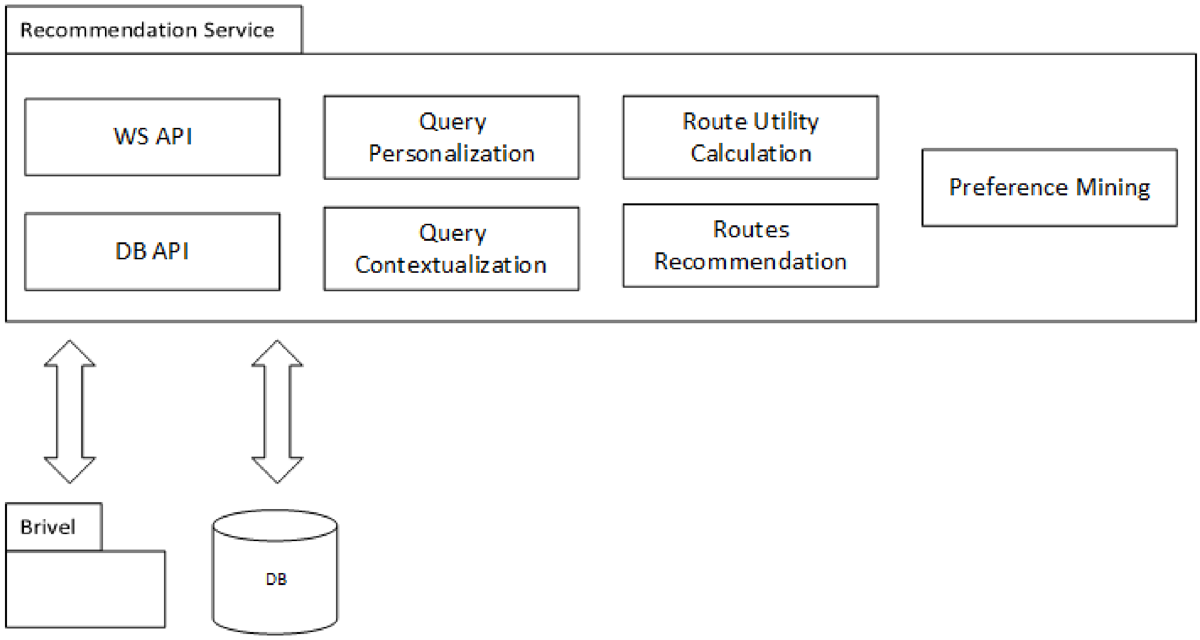


Figure 4.2: ICCS Client – Recommendation Service Architecture

Description of architecture components

- WS API: This package contains all the web services APIs that the ICCS application requires to connect with the server and exchange the data.
- DB API: This package contains all the libraries required to connect with PEACOX database and perform Create/Update/Delete operations.
- Query Personalization: This package combines information residing in the PEACOX database as well as options set through the UI in order to personalize the routing queries.
- Query Contextualization: This package adds contextual information or restrictions in the user query in order to filter not relevant results.
- Route Utility Calculation: This package provides functions that calculate a user perceived route utility per route.
- Routes Recommendation: This package provides functions that rank the available routes which are going to be presented to the user. The ranking function attempts to balance user preferences and emissions in order to present first routes that

reside within the users' comfort zone but do not cause excessive CO2 emissions to the environment.

- Preference Mining: This package provides the functionality to discover and mine user preferences based on past behaviour (i.e. trips and actions performed in the PEACOX application). The result is an updated user profile that leads to more personalized suggestions.

4.3 TCD Client – Emission and Exposure Model

This section provides an outline for developing the TCD application server to adopt the Emission and Exposure model provided by TCD in order to integrate with all components that will be developed within the PEACOX Project.

4.3.1 Description

This application provides an estimation of CO_2 emission and PM_{10} exposure rating associated with a single or multimodal trip. The CO_2 emission will be available to the users before the trip, as well as while the trip is underway. These information will be estimated based on real-time traffic information, vehicle trajectories and many situational variables for a more specific and realistic computation of CO_2 emission and exposure rating. Such computations are available for a number of modes of travel, namely car, bus, tram, metro, bicycle and walk.

Both of the door-to-door emission and exposure model contains libraries of factors and parameter values within the application in order to perform the computation as simply as possible and without calling for a greater number of outside data sets. However, few connections are made with other servers because of the minimum data requirements to run the models in real time. Real time traffic speed and vehicle speed are essential components of the models, and are needed to be included in the overall system architecture (Figure 2.1).

4.3.2 Features

The server should be designed to offer the following aspects:

- Accept information from other clients such as: route information, user information (e.g. vehicle type), real-time link speed and vehicle speed etc.
- Collect information from online sources, e.g. time, temperature etc.

- Accumulate stored information for a particular user.
- Manage all the above information to run the models.
- Collect estimations from the model and disseminate to the desired client.

4.3.3 Architecture Components

To develop the server architecture, it is necessary to understand the model architectures individually (Figure 4.3), because these provide direction to the input source and input pattern. A summary of the input has been graphically presented in the Figure 4.4. For details, please see the deliverables 3.1 and 3.2.

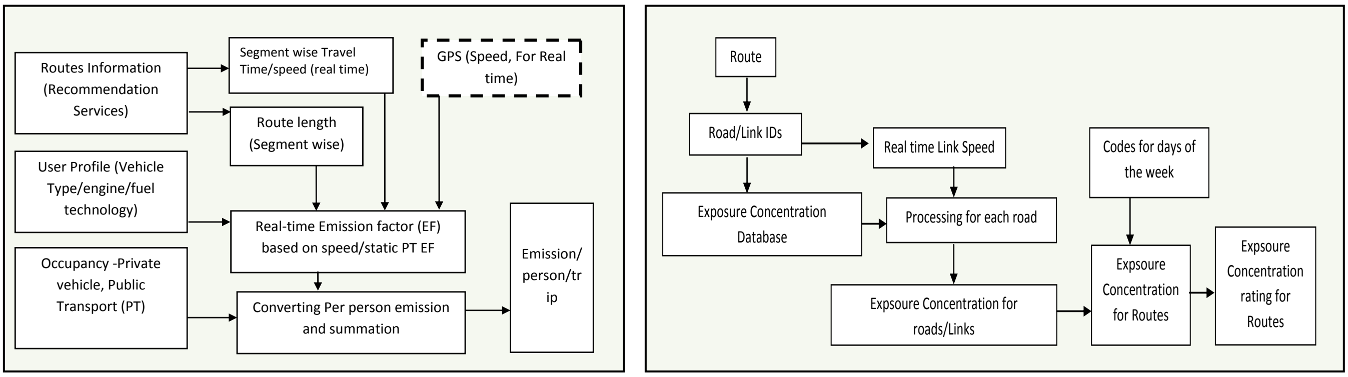
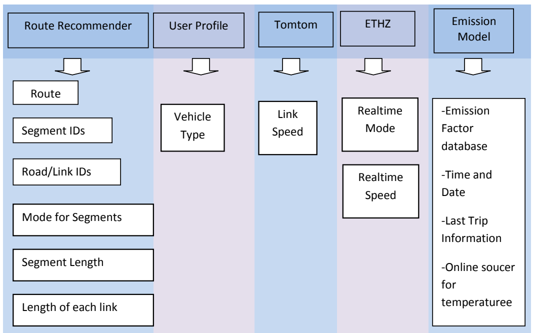


Figure 4.3: a) Basic Emission and b) Exposure Modelling Methodology.

The following figures (4.4a, b) are a summary of the methodology, which guides the development of the server architecture.



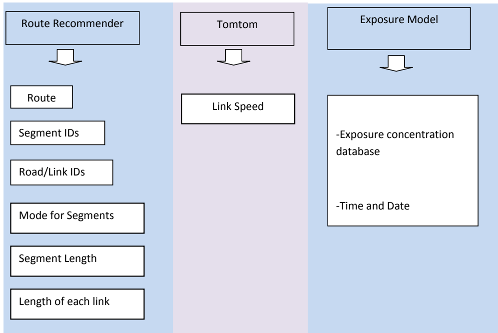


Figure 4.4: a) Desired server connections as data sources for a) Emission and b) Exposure Models. *Data type mentioned under Emission and Exposure Model category should be (as appropriate) either based on stored data (e.g. emission factor), or from any available online-source (e.g. time/date, city temperature).

The header columns (except the last one) for both Figures 4.4a and 4.4b show the connections with the other application servers needed to run the models. Besides, these figures also outline the type and number of inputs required for the models, and thus, provide a better understanding of the development of the system architecture for the server.

The server should be able to accept, deliver, store and update the stored data. Thus, database manager, data model, data service, etc. should be included. To store the emission factors and exposure concentrations, the server should contain libraries.

There should be calculating features separately for both models that provide functions of the models to estimate emission and exposure concentration, and route level outputs ready to be used in the other servers. To execute this function a query contextualisation package is necessary which will add contextual information e.g. city temperature, last trip time to the analysis.

To connect with the other servers, the TCD application server should be able to accept Web APIs. There may be an option to build user interfaces and associated necessary items to use the user interface. Other components could be added as per requirement of the PEACOX project while developing the server.

4.3.4 Emission and Exposure algorithms for TCD Application Server

The emission and exposure models described in the Deliverable 3.1 and Deliverable 3.2 have been converted from Matlab model and Arcgis format to Java codes for integration with the other components of the project. Here in this section, web service names, the input and output format, keys and units for corresponding keys for both the models, and a short description about changes made for such conversion/integration are mentioned.

4.3.4.1 Emission Model

There are two parts for the emission model. The first part will predict CO_2 emissions for a number of routes, and the second part will provide live emissions for only a single route selected by a user.

The following user scenario demonstrates the relationship between the two models. A set of options will be displayed on the graphical user interface where the emission prediction model (first part) will give possible emissions for different routes to an individual. One of the routes may be preferable to that individual, and they might choose such a route. Thus, the route, mode and corresponding information will be sent to the real-time model to start its calculation. At that time, the model (second part) will capture information from different sources along with the GPS component and estimate emissions for given inputs as long as the server sends information.

The inputs and outputs for each part of the emission model were given below. Table 4.1 and Table 4.2 show input and output for the web service of the predictive model as well as exposure model (web service is defined as “predict”), whereas Table 4.3 and Table 4.4 show the input and output for the web service named “realtime” (for real time model). Here, order/sequence of the models is important. The algorithm was developed in a way that, CO_2 emission and (particular matter) PM_{10} exposure rating for a given set of routes will be calculated first, and later, live CO_2 emission will be calculated for a selected route.

Table 4.1: List of inputs asked by the Emission Model (first part-Prediction part) & Exposure Model

Sl.	Input key in Java	Description	Unit of the input/Reference
1	userID	Unique for each user, connected with	---

		real-time model.	
2	routeID	Must be unique Id for each route.	---
3	city	vienna, dublin	----
4	userInfo.cat	Presence of catalyst or not	when user make a setting in the very first time, coded value for these parameters should be stored somewhere and send to the model for each time of use; please see Emission report for the codes (D3.1).
5	userInfo.emi	Emission standard of the vehicle	
6	userInfo.ew	Engine size and weight	
7	userInfo.fuel	Fuel type	
8	timezone	Etc/GMT+0 (Dublin), Etc/GMT+1 (Vienna)	----
9	routes.walkDis	Total distance of each mode for a multimodel trip. For instance, if there is two bus trip in a route, the input for bus distance will be sum of those two distance value.	kilometer
10	routes.carDis		
11	routes.luasDis		
12	routes.bikeDis		
13	routes.dartDis		
14	routes.busDis		
15	routes.trips.dis	Segment of distances within a trip (lowest parts of any trip. These should be roads/links)	kilometer
16	routes.trips.speed	Realtime speed for each segment.	kilometer /hour
17	routes.trips.x1	Latitude of starting point of a segment	Geographic Coordinate System: WGS 84, and Datum: World Geodetic System 1984
18	routes.trips.y1	Longitude of starting point of a segment	
19	routes.trips.x2	Latitude of ending point of a segment	
20	routes.trips.y2	Longitude of ending point of a segment	
21	routes.trips.mode	Walk was coded as 1,luas as 2, bus as 4, dart as 3,car as 5 and bike as 6	-----

Table 4.2: List of outputs given by the Emission Model (first part-Predition part) and exposure model

Sl.	Output key in Java	Description	Unit of the input
1	results.walkEmi	Emission from walk	Kilogram/km per person
2	results.bikeEmi	Emission from bike	
3	results.busEmi	Emission from bus	
4	results.carEmi	Emission from car	
5	results.luasEmi	Emission from luas	
6	results.dartEmi	Emission from dart	
7	results.totalEmi	Total Emission	
8	results.routeID	Route ID, same as table 4.3.1.	-----
9	Results.rating	Rating value for Exposure model	Comparative rating among the routes. 'A' will indicate excellent travel environment. Similarly, 'B' refers 'Good', 'C' indicates 'Average', 'D' as 'Poor', and 'E' refers 'Unhealthy' condition.

Table 4.3: List of inputs asked by the Emission Model (second part-Live emission estimation part):

Sl.	Input key in Java	Description	Unit of the input
1	routeID	route ID, same as table 4.3.1.	---
2	userID	userID, same as table 4.3.1.	---
3	city	same as table 4.3.1.	---
4	userInfo.cat	same as table 4.3.1.	---
5	userInfo.emi	same as table 4.3.1.	---
6	userInfo.ew	same as table 4.3.1.	---

7	userInfo.fuel	same as table 4.3.1.	---
8	timezone	same as table 4.3.1.	---
9	vSpeed	Vehicle speed. Real-time speed from each mode, should be come from GPS	kilometer /hour
10	mode	Real-time speed from each mode, should be come from GPS	Walk was coded as 1,luas as 2, bus as 4, dart as 3,car as 5 and bike as 6
11	dis	Real-time speed from each mode, should be come from GPS	kilometer

Table 4.4: List of outputs asked by the Emission Model (second part-Live emission estimation part)

Sl.	Output key in Java	Description	Unit of the input
1	curEmi	Live emission for each mode	Gram/kilometre per person
2	curMode	Walk,luas, bus, dart, car and bike	--
3	busEmiTotal	Total emission from bus	Kilogram/kilometre per person
4	carEmiTotal	Total emission from car	
5	bikeEmiTotal	Total emission from bike	
6	luasEmiTotal	Total emission from luas	
7	dartEmiTotal	Total emission from dart	
8	walkEmiTotal	Total emission from walk	
9	totalEmi	Total emission	

4.3.4.2 Exposure Model

The list of inputs and outputs of the exposure model has been given in the Table 4.1 and Table 4.2. Here, we have included a description of a modification that occured in the Exposure level estimation process in order to integrate the model with other components. The original exposure model (Landuse regression model) for estimating average PM_{10} values remains the same, however, the time factor and mode factors were added to calculate dose. The dose will indicate the level of exposure and it is the amount of pollutant that someone inhales during travel, and thus, it is a function of exposure concentration of a pollutant, travel time and inhalation rate. Here, the following equation was deployed to estimate dose:

$$D = \int_{t1}^{t2} C(t).\delta(t).IR(t,m).dt$$

Figure 4.5: Exposure model formula;D=dose (μg); $\delta(t)$= Time factor(unitless); $IR(t,m)$= Ingalitation rate (m^3/hr) based on mode; time in hour; and $C(t)$= μg/m^3

The time factor enhances the estimated average PM_{10} values, and mode factor provides scope to estimate dose for the travellers using different modes. The time factor has been calculated using hourly nitrogen oxide (NO_x) values of five monitoring stations in Dublin, as there is a limitation of availability for hourly PM_{10} data. The daily concentration of NO_x and PM_{10} has a r=.6 Pearson correlation, and that is comparatively better than with other pollutants (PM_{10} vs. NO was r=.51; PM_{10} vs. NO_2 was r=.58). The average hourly NO_x values for each monitoring station was calculated for 2009, and the two peaks were observed in the Figure 4.5 for all the monitoring stations (The hourly distribution over the day also showed the same tendency as average hourly values). These peaks are consistent with the peak traffic periods mentioned in the emission report (deliverable D3.1), that is, fluctuation of emission concentrations is consistent with traffic level. Then six time factors were estimated from the ratio of average concentration values in different periods, and daily average concentration for each monitor. Time factors (Table 4.5) for six time periods were calculated taking the average of the ratios in six different periods of all monitors.

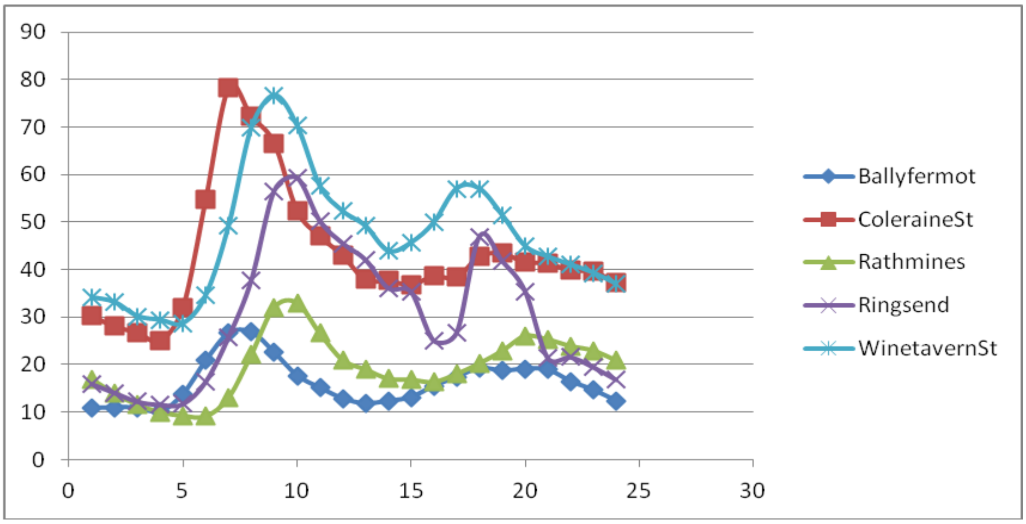


Figure 4.6: Average NO_x concentration in the monitoring stations

Table 4.5: List of Time Factors

Serial No.	Factor Name	Time Period	Time Factor
1	Kick off factor	5am -6.59am	0.74
2	Morning Peak Factor	7am-10.59am	1.35
3	Settling Factor: Noon	11am-13.59pm	0.96
4	Average Traffic Factor	14pm-15.59pm	0.96

5	Evening Peak	16pm-18.59pm	1.16
6	Settling Factor: Night	19pm-21.59pm	0.95
7	Night factor	22pm-4.59am	0.66

The mode factor, that is the inhalation factor (Table 4.6) for each mode has been taken from US EPA 2009 (Exposure Factors Handbook: 2009 Update. EPA/60 0/R-09/052A.)

Table 4.6: List of Inhalation factor

Serial No	Mode	Inhalation factor
1	Luas, Dart, Bus*	.228 m3/hr
2	Bike	1.62 m3/hr
3	Car	.57 m3/hr
4	Walk	.72 m3/hr

* Same as because the body movement is similar to the resting period while using these mode of travel.

All these factors were considered along with the average PM_{10} values to calculate dose for each type of mode users. The total value calculated from different modes provide the possible dose for each route. Although the outcome of the modelling will provide dose, the value will be expressed for the users as a band score. The level of concentration will be given in a scale rating where 'A' will indicate excellent travel environment. Similarly, 'B' refers 'Good', 'C' indicates 'Average', 'D' as 'Poor', and 'E' refers 'Unhealthy' condition. While there will be a number of alternative routes between an origin and destination, dose (µg) for each alternative will be calculated and lower dose will be rated as 'A' and so on.

4.4 ETHZ Client – Travel Mode and Trip Purpose Detection

4.4.1 Description

The travel mode and trip purpose client creates a user's travel diary using collected GPS and accelerometer data.

4.4.2 Features

Provided that the position data is logged the travel diary contains:

- Segmentation of the position data into single-mode stages and activities
- Travel mode for each stage
- Trips, that is aggregation of stages that are connected by a mode transfer
- Activity purpose for each activity

For real-time usage of the application, the travel mode and trip purpose client will also provide mode imputation during travel.

4.4.3 Architecture Components

The main components of the travel mode and trip purpose client are:

- the DB access module,
- the GPS and accelerometer post-processing module, and
- the real-time-information module.

The DB access module uses Hibernate to handle the access to the database. The main component however is the post-processing module, which implements four major steps of GPS processing:

- GPS filtering and smoothing,
- stage and activity detection,
- travel mode detection for stages, and
- activity purpose imputation.

The post-processing runs server-side and is triggered at certain times of a day and processes all available GPS points. The real-time-information module on the other hand provides on demand access and processes only the most recent measurements.

5. Processes

The purpose of this chapter is to describe the process and the involved steps for the main functionalities. Based on this process the responsibilities between server and client, as well as the different components, will be defined.

5.1 Route Calculation

Figure 5.1 describes the process for the route calculation.

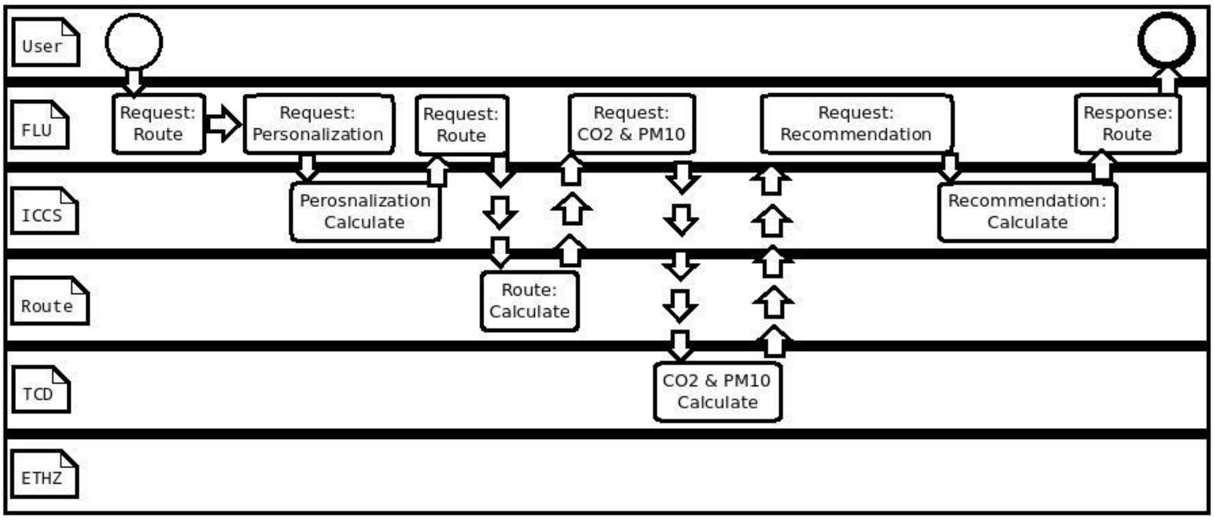


Figure 5.1: Process description for route calculation

Process description

1- User sends a route request.

2- FLU receives the request.

3- FLU sends the request for ICCS for personalization.

4- ICCS personalizes the request and return the result to FLU

5- FLU sends the personalized request to routing engine.

6- Route: calculates the route and return it to FLU.

7- FLU sends the route to TCD.

8- TCD calculates CO_2 & PM_{10} and return the result to FLU.

9- FLU receives the route from TCD and sends it for ICCS.

10- ICCS receives the route and calculates the recommendations.

11- FLU receives the final route and returns it to the client.

5.2 GPS & Accelerometer Data Logging

The process for the GPS and accelerometer is depicted in Figure 5.2.

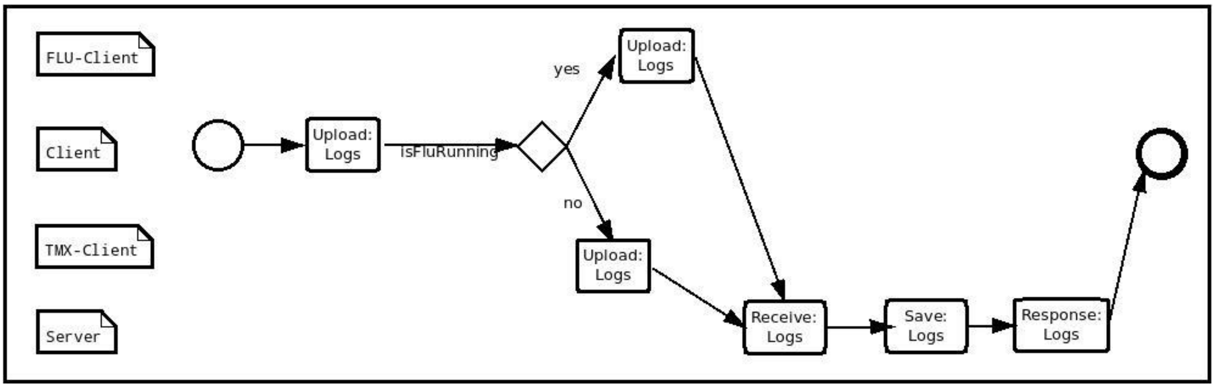


Figure 5.2: Process description for GPS and accelerometer data logging

Process description

The client in the process represents both clients (FLU & TMX).

The process starts when the client wants to upload the LOG (GPS and accelerometer logs) to the application server. If FLU client is running, the logs will be collected and send via FLU client. If FLU client is not running, TMX sends the logs to the server.

The server receives the logs and save them in the database.

5.3 Real-time Trip Mode Detection

Figure 5.3 describes the process for the real-time trip mode detection.

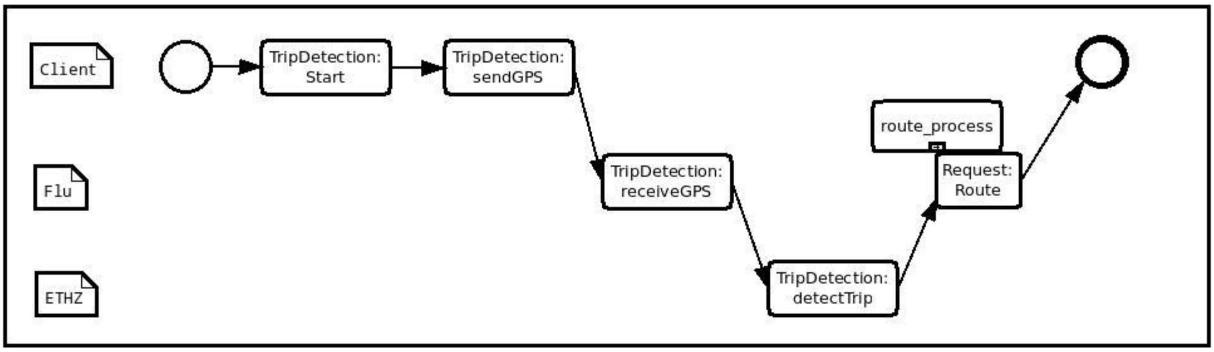


Figure 5.3: Process description for real-time trip mode detection

Process description

The client triggers the process for the real-time trip mode detection. Afterwards, the logs will be uploaded to the application server (FLU). In the next step, the GPS and accelerometer logs will be analysed by ETHZ. In the first step, ETHZ will send the detected mode back to the application server (FLU). Based on this answer, the application server (FLU) will decide if the answer should be forwarded directly to the client or if the route process should be started.

In the advanced version (second trial), the trip mode detection will be combined with the trip purpose detection.

6. Summary and Outlook

The deliverable documents the work performed in Task 6.2 "Recommendation Component", representing a part of the PEACOX WP6 "System Design and Implementation".

The main objective of the Task 6.2 was to build the recommendation component and investigate how the recommendation component will be connected to other components.

This deliverable described the implementation of server and client components and the selection of the specific methods will be justified. Hereby, the feature of the components will be described as well as the architecture components.

Following client components were described in detail:

- Journey Planner Application (FLU)
- Navigation Client (TMX)

The server components that are described in detail are:

- Application Server (FLU)
- ICCS Client – Recommendation components
- TCD Client – Emission and Exposure Model
- ETHZ Client – Travel Mode and Trip Purpose Detection

In the last chapter, the most important processes were defined and described in detail. Based on this description the responsibilities and chronological order were defined.

The deliverable D6.1 Use case Report, D6.2 Server and Client Implementation as well as the D6.3.1 System Design and Interface Definition represent together the results of the specification phase for the PEACOX clients.

7. Figures

Figure 2.1: Overall system architecture 5

Figure 3.1: Client architecture – journey planner application 7

Figure 3.2: Client architecture – navigation client 9

Figure 4.1: Server architecture – application server 11

Figure 4.2: ICCS Client – Recommendation Service Architecture. 13

Figure 4.3: a) Basic Emission and b) Exposure Modelling Methodology. 15

Figure 4.4: a) Desired server connections as data sources for a) Emission and b) Exposure Models. 16

Figure 4.5: Exposure model formula; 19

Figure 4.6: Average NO_x concentration in the monitoring stations........ 20

Figure 5.1: Process description for route calculation 23

Figure 5.2: Process description for GPS and accelerometer data logging 24

Figure 5.3: Process description for real-time trip mode detection 25

8. List of tables

Table 4.1: List of inputs asked by the Emission model (first part-Precition part) & Exposure Model 17

Table 4.2: List of outputs asked by the Emission model (first part-Precition part) and exposure model 18

Table 4.3: List of inputs asked by the Emission model (2nd part-Live emission estimation part): 18

Table 4.4: List of outputs asked by the Emission model (2nd part-Live emission estimation part): 19

Table 4.5: List of Time Factors 20

Table 4.6: List of Inhalation factor 21