

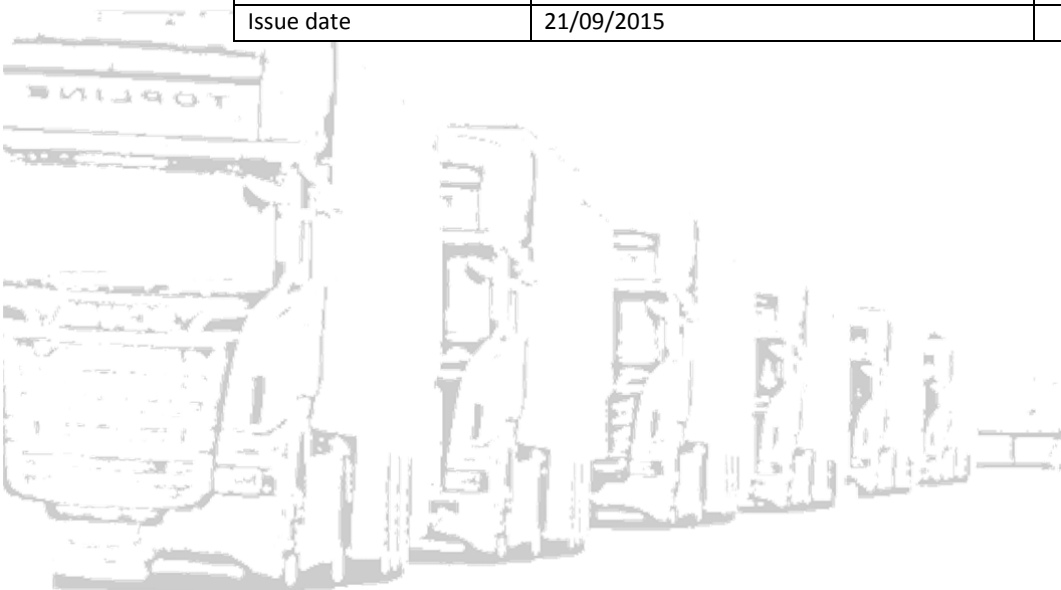
EUROPEAN COMMISSION
 SEVENTH FRAMEWORK PROGRAMME
 GA No. 610990

*Cooperative dynamic formation of platoons for safe and
 energy-optimized goods transportation*



D5.1. Fuel efficient route calculation

Deliverable No.	COMPANION D5.1	
Deliverable Title	Fuel efficient route calculation	
Dissemination level	Public	
Written By	Thilo Schaper (Volkswagen AG) Christian Bruns (Volkswagen AG)	28/08/2015
Checked by	Jonas Mårtensson (KTH)	10/09/2015
Approved by	Magnus Adolfson (Scania)	18/09/2015
Issue date	21/09/2015	



History log

<i>Name</i>	<i>Status</i>	<i>Version</i>	<i>Date</i>	<i>Summary of actions made</i>
VW Thilo Schaper Christian Bruns	Editor	0.1	22/07/2015	Written basic structure
VW Julia Kwasny	Editor	0.2	21/08/2015	Added content related to Additional dynamic data
VW Thilo Schaper	Editor	0.3	27/08/2015	Added section for Truck Routes and RCE and Traffic Monitor
VW Christian Bruns	Editor	1.0	28/08/2015	Revision and minor changes
VW Christian Bruns	Editor	1.12	01/09/2015	offboard -> off-board
VW Christian Bruns	Editor	1.13	03/09/2015	Integrated Feedback from OFFIS
VW Christian Bruns Thilo Schaper	Editor	1.14	03/09/2015	Integrated Feedback from KTH, Conclusion
VW Thilo Schaper	Editor	1.15	04/09/2015	Revised after feedback from OFFIS/KTH
VW Christian Bruns	Reviewer	1.15	10/09/2015	Reviewed
KTH Jonas Mårtensson	Reviewer	1.15	10/09/2015	Reviewed
OFFIS Sönke Eilers	Reviewer	1.15	10/09/2015	Reviewed
VW Christian Bruns	Editor	2.00	10/09/2015	Final review version
VW Christian Bruns	Editor	2.10	21/09/2015	Final version

Executive summary

This report constitutes the deliverable for the task T5.1: Fuel efficient route calculation. The overall objective of this task is to investigate and develop a software module for the calculation of eco routes for heavy duty vehicles [1]. Given the coordinates for a starting point of a route and a destination the most fuel efficient path for heavy vehicles based on the state of the infrastructure and environmental data (Task 4.1 and Task 4.2) will be identified.

In this task we have investigated and developed different variations for routing algorithms and developed a mechanism to supervise calculated routes for incoming traffic events. The resulting methodologies are implemented within a software component “Route Calculation Engine” (RCE) which is deployed as part of the off-board system for platoon coordination. The developed algorithms and software modules are applicable in the scope of COMPANION and are used as input for the scheduling of platoons (Task 5.2). Improvements and optimizations are in discussion and will be part of further implementations.

The report consists of two main parts. The first part describes methods for fuel efficient route calculation. It explains basic navigation fundamentals and describes the calculation of routes on a digital map. To improve the calculation results and adapt to the real life conditions, different additional dynamic data were evaluated and incorporated into the calculation. Since performance is a key factor in navigation systems, several algorithms are evaluated to match the needed computation speed. Furthermore the specific characteristics for heavy duty vehicles and fuel efficiency are described.

The second part is about the system components which have been developed in the context of route calculation for the COMPANION project off-board system. The COMPANION off-board architecture is designed with a stateless route calculation system in which the computations can be done on a scalable offboard-system. The offboard-system communicates with the COMPANION onboard components inside the trucks. The supervision of incoming traffic messages with relation to calculated truck routes is done by a monitoring component. This component supervises incoming incidents and computes the impact in the form of delayed estimated time of arrival and adapted recommended speed profile.

Contents

HISTORY LOG	2
EXECUTIVE SUMMARY	3
CONTENTS.....	4
1 INTRODUCTION	6
2 FUEL EFFICIENT ROUTE CALCULATION	7
2.1 Fundamentals.....	7
2.1.1 Basics of Navigation Systems	7
2.1.2 The Digital Map	8
2.1.3 Route Calculation	9
2.1.4 Travel-time and –speed estimation	13
2.2 Additional Dynamic Data	13
2.2.1 Real Time Traffic Information	14
2.2.2 Planned Special Events.....	16
2.2.3 Weather.....	18
2.2.4 Lane Level Traffic	19
2.3 Truck Routes	20
2.3.1 Problem Definition	20
2.3.2 Simplifications.....	20
2.3.3 NDS Eco Routing.....	21
2.3.4 Results.....	22
3 SYSTEM COMPONENTS.....	25
3.1 Route Calculation Engine.....	25
3.1.1 Internal Architecture	25
3.1.2 MS Azure integration.....	27
3.1.3 System Start-up / Initialization.....	32
3.1.4 Major Use cases	32
3.2 Traffic Monitor	35
3.2.1 Internal Architecture	35
3.2.2 MS Azure integration.....	35
3.2.3 System operation	36

3.2.4 Messages..... 37

4 CONCLUSION 39

REFERENCES..... 40

GLOSSARY 42

A APPENDIX..... 43

6.1 Restrictions and limitations of the demonstrator RCE..... 43

1 Introduction

This report constitutes the deliverable D5.1: Fuel efficient route calculation. It is the result of the work within the task T5.1: Fuel efficient route calculation, which belongs to work package 5: Off-Board System for Platoon Coordination.

The overall objective of task T5.1 is to develop and assess a route calculation engine able to identify the most fuel-efficient path from a given source position to a destination position. In addition to static map data, dynamic information shall be used to reflect the state of the infrastructure (compare tasks T4.1 and T4.2). A vehicle fuel consumption model shall be used, which is applicable for heavy vehicles (compare task T4.3).

The following chapters describe our investigation on different routing algorithms for fuel efficient truck routes. The report consists of two main parts. The first part describes methods for fuel efficient route calculation. It explains basic navigation fundamentals and describes the calculation of routes on a digital map. Additional dynamic data are used in the calculation to improve the calculation results and adapt environmental and traffic status to real life conditions. Special characteristics for truck routes are discussed as well as possible performance optimizations.

The second part is about the resulting system components and their integration and interfaces within the COMPANION off-board system, namely the MS Azure cloud. It describes the route calculation engine (RCE) and the handling of external events, e.g. incoming traffic messages. Furthermore the supervision of calculated routes is done by a new monitoring component Traffic Monitor, which is also responsible for the computation of traffic incident impact on time and speed variations for a calculated route.

We end the report with conclusion of the developed system.

2 Fuel efficient route calculation

This Chapter describes the evaluated methods for fuel efficient route calculation. It starts with an overview about navigation basics and the possibility to improve the calculation result by the addition of dynamic data. The COMPANION specific part to calculate truck routes is described at the end of this chapter.

2.1 Fundamentals

State of the art navigation systems consist of several parts which all work together either in a vehicle or in an offboard-system to provide an answer to the question of how to travel from a starting location A to a destination B. In the context of this evaluation the calculation of a fuel efficient route and the used data to get satisfying results is of major interest. Travel times and speed estimates are a secondary result for a calculated route and described at the end of this section.

2.1.1 Basics of Navigation Systems

From a functional view a navigation system consists of the following basic components, as described in [2]:

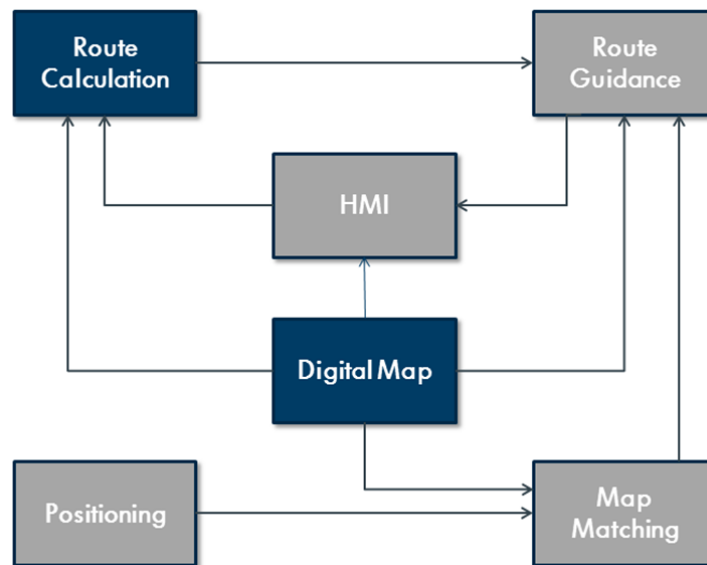


Figure 1: Functional Blocks of a navigation system.
The blue components are of major interest in the scope of COMPANION Task 5.1.

The fundament of navigation is the road network and the determination of the vehicles position within this network. The road network is generally located in a map database (Digital Map) and is a digitized machine-readable representation of the real existing streets with its information about geometrical structure, dimensions, limitations and further information like travel times, etc. This data is the source of information for the route calculation which calculates a route from a starting location A to a destination B. The module Positioning is used to determine the vehicle's current position. This is generally based on data from a GPS receiver. The position information is determined and described in WGS84 coordinates. The accuracy can be improved by incorporating additional sensors (e.g. odometer), dead reckoning and map matching techniques to compute more accurate information.

Map Matching projects the current “raw” position to the road network and makes it ready for further processing. The “map-matched” position contains a road identifier, logical travel direction and an offset along the road. The Route Guidance is responsible for providing navigation advices (e.g. driving instructions) to the driver, which are displayed on the vehicles HMI in a graphical or acoustical way.

2.1.2 The Digital Map

A digital map contains all information needed for a navigation system to calculate routes for a given request. The information can be regarded as a network of roads with corresponding attributes. The digital map is typically provided in a standardized format (GDF or NDS) by a map supplier. For map creation the map supplier uses several data sources and a manual or semi-automatic method for the vectorization of aerial images or drives the streets with special equipped vehicles for data collection. Digital maps can be characterized by content (what type of information and which attributes are contained) and coverage (where - in which area or country - is data available). A typical digital road network consists of the following objects:

Nodes	A node typically represents crossings or the end of a street. A node has a position, specified by its longitude and latitude coordinates (as WGS84 coordinates) and connected (adjacent) edges.
Edges	An edge is used to model a piece of road, connecting two nodes. Each edge has several basic attributes such as legal travel direction, geometry, road classification and average speed. Additional attributes are slope profile, historical speed (per hour/days of the week), etc.
Shapepoints	Shapepoints are used for modeling the geometry of edges. They can be used for visualization or referencing of attribute information.
Attributes	Edges and nodes can have attributes. They are used to model additional information, e.g. speed limits and driving limitations like turning restrictions. Attributes can be characterized as mandatory or optional, depending on the data model and map supplier.

Table 1: Elements of a typical digital road network

The following figure shows a typical road network:

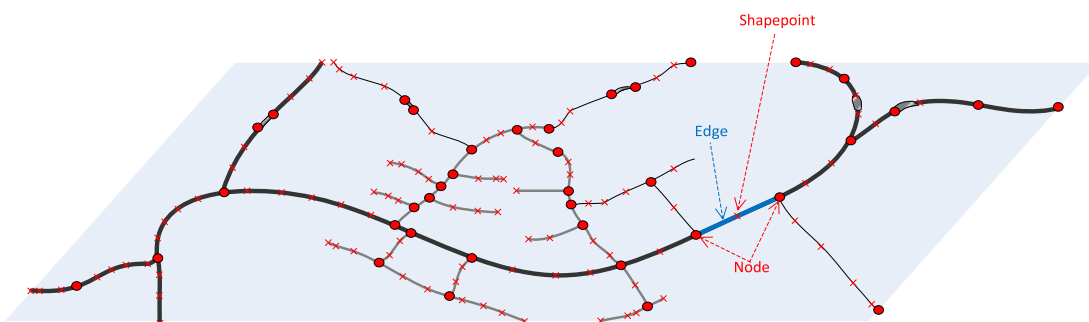


Figure 2: Example of a road network with nodes, edges and shapepoints. The different highlighting represents different street types (like highways or rural roads)

In the scope of the COMPANION project we use the Navigation Data Standard (NDS) as a source of information for road network. NDS is a standardized format for automotive navigation databases. The format is jointly developed by automobile manufacturers and suppliers. It aims at developing a

standardized binary database format that allows the exchange of navigation data between different systems. For more information see [3] and [4].

In this project we use the NDS specific “Routing Building Block” for road network information and a set of attributes for the route calculation. The following table gives a brief overview:

Name	Description	Effect/Usage
<i>Slope</i>	Contains information about the slope of an edge.	Used for eco routes, uphill driving (higher slope value) results in higher fuel consumption.
<i>Turning Restrictions</i>	Describes restrictions where it is not allowed to drive from one edge to another.	Used for general path finding. Required to generate only legal routes.
<i>Truck Restrictions</i>	Describes where trucks are not allowed to pass.	Used for truck routes. Ensure trucks do not exceed max. width and height.
<i>Authorization Restrictions</i>	Describes where vehicles are (not) allowed to pass.	Used for general path finding. Only travel on public roads.
<i>Geometry</i>	Describes the detailed geometry of a road element.	Used for visualization.
<i>Speed information (w.r.t. average speed, legal speed limit, historic speed)</i>	Describes speed information in different ways. Realistic, legal and time-dependent profile.	Used for general path finding and estimation of arrival time (ETA).
<i>Consumption Speed Variation</i>	Describes energy consumption due to breaking and accelerating at unusual sharp curves.	Used for eco path finding and estimation of fuel consumption.

Table 2: Overview of attributes used for route calculation

2.1.3 Route Calculation

This section gives an overview about route calculation in a road network. Given the basic definition of a digital map in section 2.1.2, we provide a brief overview of methods for route calculation in general. The different methods deliver as a result an ordered list of nodes and edges which a vehicle has to travel across to arrive at a destination.

2.1.3.1 The Basic Problem: “Find the shortest path”

In general the problem of “path finding” is to find the shortest path from a starting position A to a destination B. This problem has to be solved in an adequate amount of time (which is as fast as possible).

A road network can be mathematically represented as a graph

$$G = (N, E, l)$$

with N as the set of nodes, E as the set of Edges and l as a function mapping one edge to a value for (positive) “cost” of travelling along the edges

$$l : E \rightarrow \mathbb{R}_0^+$$

A route R_{AB} from A to B is a sequence of nodes n_1, n_2, \dots, n_p with $(n_i, n_{i+1}) \in E$, $n_1=A$, $n_p=B$. The Length of such a route is:

$$l(R) = \sum_{i=1}^{p-1} l(n_i, n_{i+1})$$

The length of the shortest path is defined as:

$$\delta(A, B) = \begin{cases} \min\{l(R_{AB})\}, & \text{if a path exists from A to B} \\ \infty, & \text{else} \end{cases}$$

A shortest path R_{AB} from A to B is a path of length $l(R_{AB}) = \delta(A, B)$

Note that “shortest” is used with respect to the edge weight. The “shortest path” is in fact the route with the minimum total weight (also known as abstract “cost”). So this algorithm can be applied to calculate routes for various use cases (most notably the “fastest route” minimizing the total travel time).

2.1.3.2 Dijkstra's Algorithm

One of the most common algorithms for solving shortest path problems (SPP) is Dijkstra's algorithm [7]. A short prose description of the algorithm is:

1. Initialize the source node with a distance of 0 and all other nodes with a distance ∞ .
2. As long as there are unvisited nodes select the unvisited node with the least distance to the source.
3. Mark the node as visited.
4. Calculate the distance to the source for all unvisited neighbor nodes.
5. Update the distance and store the current node as predecessor if the calculated distance is lower than the previously calculated value.
6. Stop if the destination node has been reached.

2.1.3.3 Basic A* Algorithm

The A* (A star) algorithm [8] is an extension of Dijkstra's algorithm. The A* algorithm uses a heuristic to estimate the minimum costs from each node to the destination. This results in a reduced search-tree that “grows” in the direction of the destination (see Figure 3 and Figure 4).

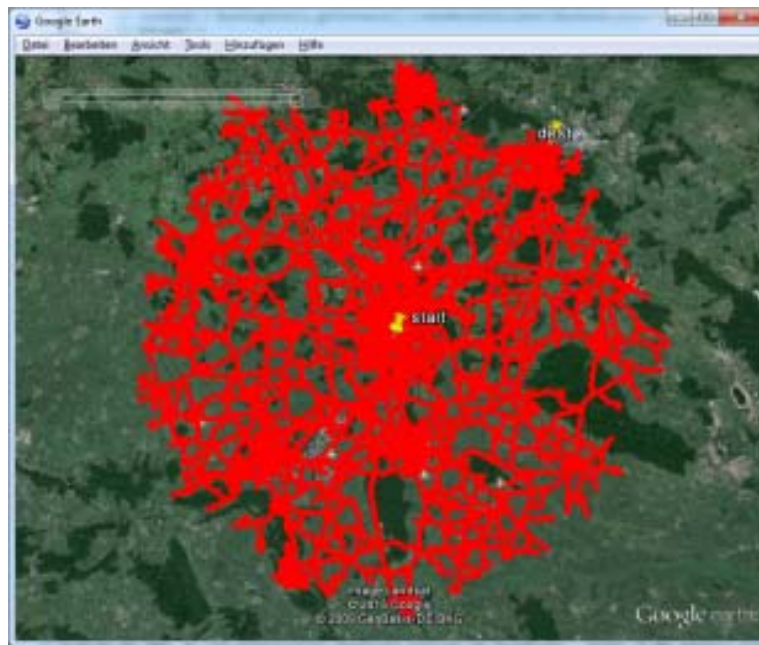


Figure 3: A* search-tree without heuristic (→ Dijkstra)

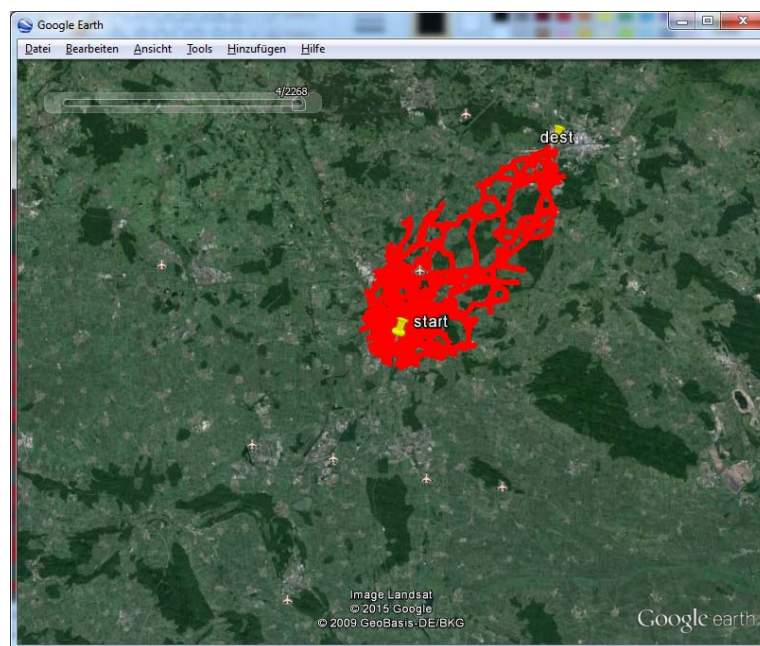


Figure 4: A* search-tree with heuristic

2.1.3.4 Optimizations of the basic A* algorithm

The optimization of the A* algorithm has been an active area of research. There have been developed several strategies to improve the performance of the A* algorithm. One possibility to increase the performance is to use a bidirectional search [9]. In this variant one search tree is initialized to search from start to destination and another in the opposite direction from the destination to the start.

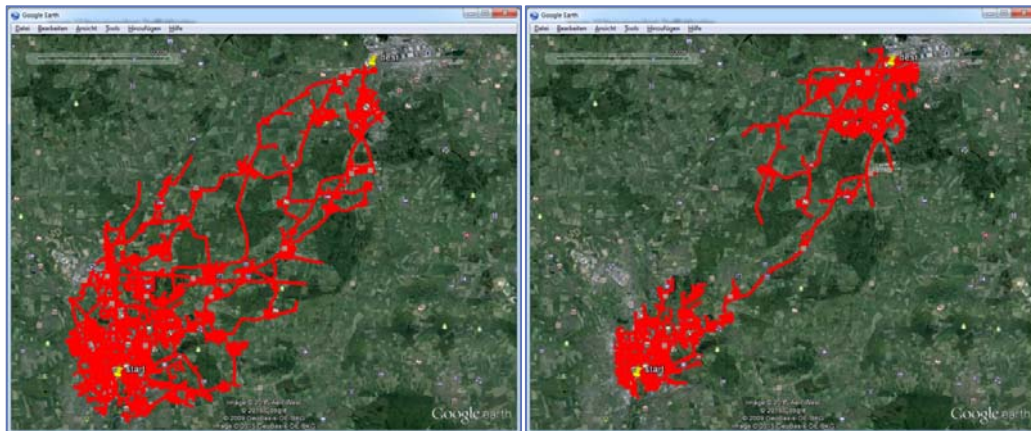


Figure 5: Search-tree comparison, left: uni-directional, right: bi-directional

An optimal route is found as soon as these two searches meet. With a valid heuristic, the cost sum of all remaining nodes is higher than the costs of the route found. The bidirectional A* algorithm can be applied to the road network without the need of preprocessing.

One drawback of the bidirectional approach is that, given a time of departure at the start of the route, the point in time for the elements of the backward search is unknown. The point in time is needed to determine the exact possible speed from speed-profiles, current traffic jams, road blockings, the current weather situation, etc. A way to use the bidirectional approach for time dependent route calculations is described by Delling [11].

In addition to bidirectional search, long distance routing can be improved by using “hierarchical search”. This variant requires several levels of road networks, also known as “highway hierarchies”. The NDS map contains precompiled multiple hierarchical layers of the road-network (compare [4] and [20]). For COMPANION we currently use a unidirectional A* algorithm which takes advantage of these multiple levels. For illustration, compare Figure 6.

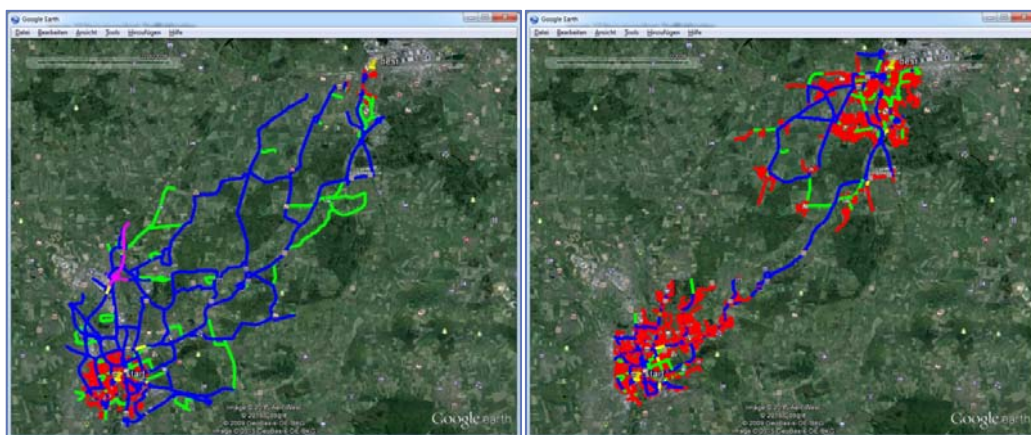


Figure 6: Search-graphs colored with multiple levels, left: uni-directional, right: bi-directional

2.1.3.5 High performance time dependent algorithms

The most common problem in the scope of route calculation is the reduction of computation time. There has been research done in the field of static road-networks which reduces query times to microseconds and different techniques to achieve acceptable calculation results exist. All high performance algorithms require a preprocessing of the road network. This preprocessing may take hours depending on the network size.

Table 3 shows a brief overview on possible techniques and their corresponding preprocessing time, memory per node and query times. The example is adapted from “Daniel Delling, Dorothea Wagner. Time-Dependent Route Planning” [11].

Technique	Preprocessing time [h:mm]	Memory per node [Bytes]	Query time [ms]	Speed up factor
Dijkstra	0	0	1502.88	1
ALT	0:23	128	94.26	16
SHARC [12]	1:16	155	25.06	60
Contraction Hierarchies (CH)	0:25	1019	1.22	1231

Table 3: High performance time dependent routing algorithms (Germany, midweek)

In the scope of COMPANION we have to consider dynamic information in the form of traffic incidents and weather impact. This leads to the use of an algorithm usable in dynamic networks. The aforementioned techniques can be applied to further improve the implemented system to create a production-ready system.

2.1.4 Travel-time and -speed estimation

The route calculation produces, beside the list of road network edges, as a second result the estimated time of arrival (ETA) and a per-link speed profile estimation. We use several sources of information to improve the estimation of the travel-speed of the truck along the route.

The following static attributes from the NDS-map are evaluated:

- Time Dependent Speed Profiles
- Speed limits
- Average Speed
- Ferry travel times

The NDS map contains historic speed profiles, provided per day of the week and time of the day. If no speed profiles or Real Time Traffic Information is available, the average speed from the NDS map is used to estimate the truck speed. The speed limits serve as upper limit to the possible travel speed.

2.2 Additional Dynamic Data

The route calculation methods described in 2.1.3 are based on static information from the digital map. The road network is used to calculate routes with a given start time (or backward) which arrive at a destination at a given arrival time. To improve the calculation results (e.g. for the ETA) and adapt to the real life conditions, different additional dynamic data were evaluated and can be incorporated

into the calculation. While the deeper investigation is done as part of the Tasks 4.1 and 4.2 this section gives a brief overview on how these data can be used within a route calculation.

2.2.1 Real Time Traffic Information

Traffic serves as major influence factor for vehicle routing. Real Time Traffic Information (RTTI) delivers information about the traffic on the road at the current time or even at a future time (predicted traffic). Using RTTI within a route calculation is a key factor for today's navigation systems – as providing realistic/best routes for customer-acceptance.

There exist several (commercial) sources for traffic data:

INRIX	INRIX [13] provides services and mobile applications pertaining to road traffic and driver services. They collect data about roadway speeds from mobile phones, trucks, delivery vans, and other fleet vehicles equipped with GPS locator devices Data retrieved from.
TomTom	TomTom [14] delivers digital maps and other dynamic content for navigation and location-based services, including personal and in-car navigation systems, and provides data used in a wide range of mobile and Internet map applications.
Google	Google Traffic [15] works by crowdsourcing the GPS information from phone users. By calculating the speed of users along a stretch of road, Google is able to generate a live traffic map.

Table 4: Examples of commercial solutions for traffic data

In the scope of the COMPANION project we are using a Real Time Traffic Information (RTTI) live service provided by TomTom. The TomTom service has been licensed after qualitative verification of traffic data coverage and accuracy. The service provides traffic information for Europe. It contains traffic incidents which contain:

- traffic events
- operator actions
- advices
- impacts (delay)

From these datasets delays (due to abnormal traffic, e.g. congestion caused by an accident) and road blockings (due to road works) are provided to the route calculator. Blocked roads are excluded from the route calculation. The reported delays are used to correct the travel-times and recommended vehicle-speeds.

If any incident happens (like an accident or roadworks) which has an effect on traffic, it will be part of the traffic information. The current traffic situation can be checked at the TomTom Live Traffic website [14], see Figure 7.

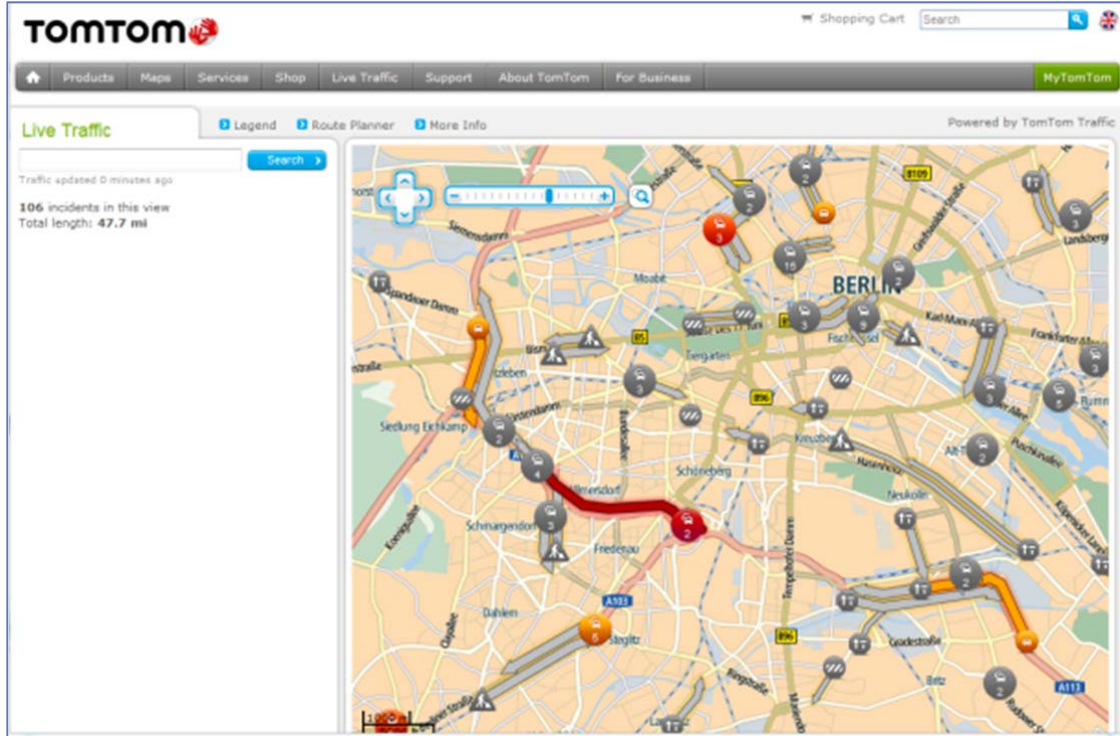


Figure 7: TomTom Live Traffic Map [14]

Figure 8 shows a data sample of the TomTom HD Traffic service. The service is implemented as REST-service via HTTP [21]. For each country a dataset in the DATEX 2 format [19] can be requested. The XML structured data contains “situation record” entities which contain the incident type, e.g. “AbnormalTraffic”, along with an identifier, version and observation time. The “validity” tag allows for modeling incidents with a time-based restriction, e.g. roads or lanes used in opposing travel directions during the day. The incident location is given in “groupOfLocations”, in this case a linear location reference encoded with “OpenLR” [20]. Finally, the impact on the road network is given by “delayTimeValue” and “averageSpeed”.

```
<?xml version="1.0"?>
- <d2LogicalModel modelBaseVersion="1.0" xmlns="http://datex2.eu/schema/1_0/1_0">
  - <exchange>
    - <supplierIdentification>
      <country>nl</country>
      <nationalIdentifier>TomTom HD Traffic Service</nationalIdentifier>
    </supplierIdentification>
  </exchange>
  - <payloadPublication lang="en" xsi:type="SituationPublication"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <publicationTime>2014-07-17T09:29:19Z</publicationTime>
    - <publicationCreator>
      <country>other</country>
      <nationalIdentifier>1b949889-7476-49ee-8a71-3188aab6a4fb</nationalIdentifier>
    </publicationCreator>
    - <situation id="TTI-1b949889-7476-49ee-8a71-3188aab6a4fb-TTL119701295462592">
      - <headerInformation>
        <confidentiality>internalUse</confidentiality>
        <informationStatus>real</informationStatus>
        <urgency>urgent</urgency>
      </headerInformation>
      - <situationRecord xsi:type="AbnormalTraffic" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" id="TTI-1b949889-7476-49ee-8a71-3188aab6a4fb-TTL119701295462592-1">
        <situationRecordCreationTime>2014-07-17T09:29:19Z</situationRecordCreationTime>
        <situationRecordVersion>1</situationRecordVersion>
        <situationRecordVersionTime>2014-07-17T09:29:19Z</situationRecordVersionTime>
        <situationRecordFirstSupplierVersionTime>2014-07-17T09:29:19Z</situationRecordFirstSupplierVersionTime>
        <probabilityOfOccurrence>probable</probabilityOfOccurrence>
      - <validity>
        <validityStatus>active</validityStatus>
        - <validityTimeSpecification>
          <overallStartTime>2014-07-17T09:29:19Z</overallStartTime>
        </validityTimeSpecification>
      </validity>
      - <impact>
        - <delays>
          <delayTimeValue>127.0</delayTimeValue>
        </delays>
      </impact>
      - <groupOfLocations>
        - <locationContainedInGroup xsi:type="Linear">
          - <locationExtension>
            - <openlr>
              <binary version="3">Cwd2hiUzkSOKCwM8/sojGw==</binary>
            </openlr>
          </locationExtension>
        </locationContainedInGroup>
      </groupOfLocations>
      - <situationRecordExtension>
        <alertCEventCode>108</alertCEventCode>
      </situationRecordExtension>
      <abnormalTrafficType>queueingTraffic</abnormalTrafficType>
      - <abnormalTrafficExtension>
        <averageSpeed>14.0</averageSpeed>
      </abnormalTrafficExtension>
    </situationRecord>
  </situation>
  - <situation id="TTI-1b949889-7476-49ee-8a71-3188aab6a4fb-TTL119258096601744">
    - <headerInformation>
```

Figure 8: TomTom traffic feed xml

2.2.2 Planned Special Events

Task 4.1 and 4.2 evaluates the influence factors on traffic congestion such as daily concerts or rush hour, traffic lights, construction zones, etc. As stated in [16] most of these factors can occur regularly (i.e. rush hour), exist only once and for a limited amount of time (i.e. construction zones) or their occurrence is completely unpredictable (i.e. accidents). Recurring situations are well handled by commercial solutions like TomTom's traffic service, for they can be derived from historic information (see [17]). In the context of Task 4.1 we engaged the influence of special planned events (PSE), which are non-periodic and, dependent on their probably large attendance, will have an effect on the traffic around the event area. PSEs can be considered as punctual incidents with a well-known start and end time.

Kwoczek et al. evaluated i.e. the effects around the LANXESS arena. Figure 9 displays the summed up delay time in a circular impact region with a radius of 500 m around the LANXESS Arena in Cologne (Germany) for all Tuesdays in a considered dataset, where no events happened.

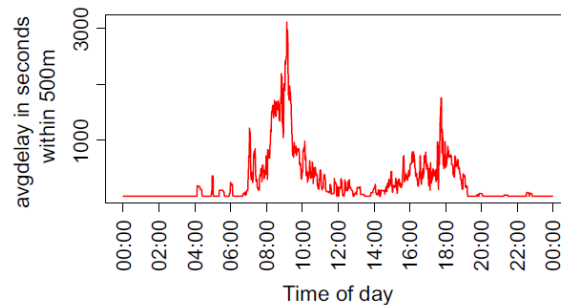


Figure 9: Average historic congestion around the LANXESS arena on Tuesdays.

He further stated that there exist peaks for inbound and outbound traffic, before and after the event happened and took the concert of Mark Knopfler on Tuesday the 2nd of July 2013 as an example (Figure 10).

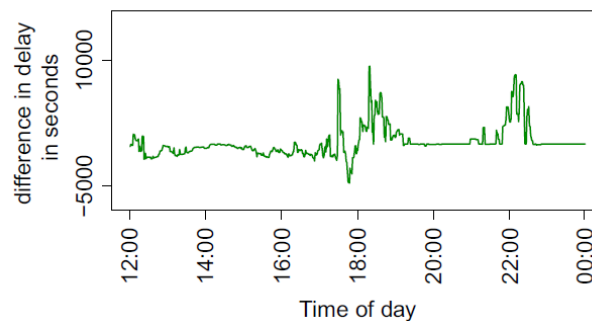


Figure 10: Additional delay due to the concert of Mark Knopfler on top of the historic trend line for Tuesdays

To improve the results of the route calculation and to avoid situations around special events and it's journey the RCE provides an interface that allows the import of traffic incidents predicted for current or future PSEs. The predicted delays are handled in the same way as the delays reported by the RTTI-service.

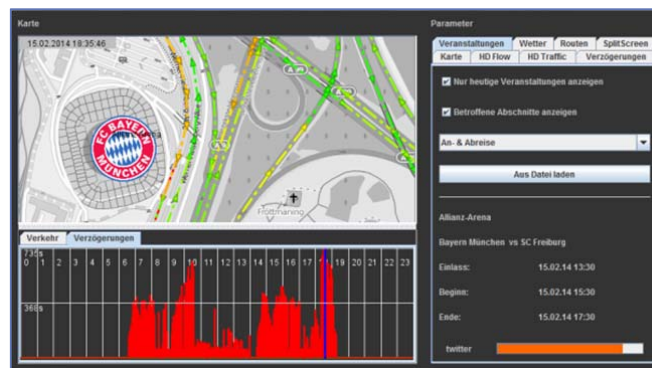


Figure 11: Analyzing the impact of Planned Special Events on the traffic flow

2.2.3 Weather

Factors that influence traffic like inclement weather conditions and road works are evaluated within Task 4.2. These factors lead to a speed reduction of the vehicles, increase their travel time and therefore influence fuel consumption. This is taken into account by the route calculation engine.

Next to road works, the following weather risk factors are relevant for travel times:

Precipitation like rain, snow, hail	Precipitation leads to a limited view of the driver through the windshield, slippery roads, aquaplaning and spray on the windshield caused by vehicles in front. This forces the driver to drive slower.
Slipperiness	Traffic slows down also after the occurrence of precipitation. Ice on the streets leads to the awareness of the increased risk of accidents and therefore drivers slow down.
Aquaplaning	Traffic slows down also after the occurrence of rain. Water on the streets leads to the awareness of the increased risk of accidents due to aquaplaning and therefore leads to a speed reduction.
Spray	During or after the occurrence of precipitation, the view of the driver through the windshield can be further affected by spray effects of previous vehicles. This effect depends on the road surface.
Fog	Fog leads to a limited view of the driver in front of the vehicle.
Wind	Wind especially coming from the front has an effect on the truck speeds.
Increased traffic	Due to bad weather conditions, more people will use their car. The consequence will be more traffic and a slow-down of traffic in certain situations.
More accidents	More accidents will happen due to bad weather conditions which lead to congestions and reduced speeds.

Table 5: Weather risk factors

Some of these weather factors also lead to an increased frictional resistance of the trucks. One example is especially windy conditions and precipitation. The frictional resistance additionally leads to an increased fuel consumption. These effects are taken into account within Task 4.3.

The goal is to include models for speed reduction caused by risk factors into the traffic feed that is used by the route calculation engine. Historic data can be used to create these models and by knowing the occurrence of the influencing factors in advance, these models can be used to predict the resulting traffic change. These predicted speed changes due to risk factors are given to the route calculation engine via the traffic feed. The traffic feed gives future speeds to the route calculation engine already taking into account the speed reduction due to influencing factors. By knowing the current and predicted speeds, the fuel calculation engine can calculate the optimal route according to travel time and fuel consumption.

An example for the output of the route calculation engine taking into account weather risk factors is shown in Figure 12. In general, weather can be regarded as a moving event with a spatial extent described by a polygon.

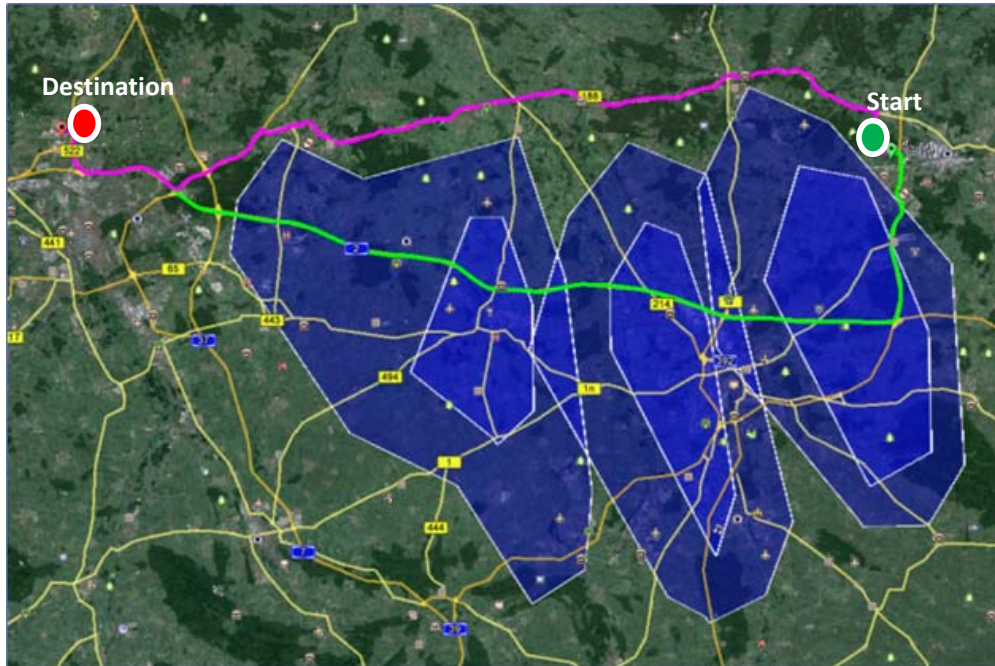


Figure 12: Weather-polygon with two routes from Start to Destination

The picture shows the weather situation between a start and a destination point for a truck route. The weather situation is displayed by a blue polygon with an inner and an outer area that represent different intensities of the weather effect. The weather polygons are moving from east to west while the polygon in the East part shows the weather situation when the truck starts. Not taking into account the bad weather conditions, the recommended route would have been the light green route. Taking into account the future weather situation and knowing the resulting reduced speeds on the green route due to the inclement weather situation in advance, the route calculation engine recommends the pink route. This alternative route is calculated by the route calculation engine solely based on the knowledge about future speeds on all routes.

2.2.4 Lane Level Traffic

Another point of interest is the improvement of speed information for certain lanes and vehicles. As part of Task 4.1 this can be achieved by the use of in car sensors. The goal is to derive information about the vehicles environment and determine e.g. speed profiles for trucks. A possible technique is discussed in [22]. Sasse et al. use a combination of sensor (frontal radar and camera, backward cameras) on vehicles to detect trucks and to compute the relative speed with respect to the ego-vehicle. By aggregating this information from a lot of sensors on a back-end, it becomes possible to learn the average speed of trucks.

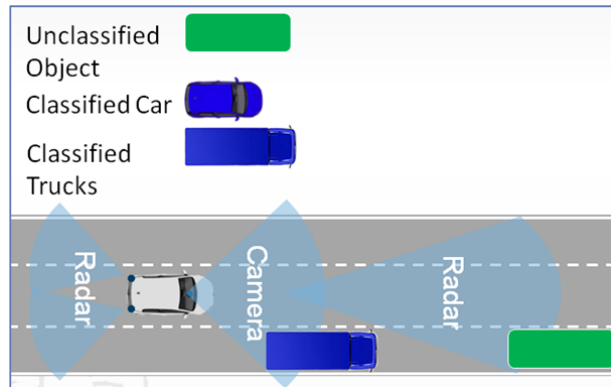


Figure 13: Classifying vehicles with XCD-vehicle sensors

This information can be used to enrich the digital map with truck specific and lane dependent speed information.

2.3 Truck Routes

Based on the stated route calculation fundamentals and dynamic data, this section describes the development of a fuel efficient route calculator for truck routes. We begin with the problem definition, summarizing the truck-specific requirements. Then we explain useful simplifications and evaluate a standardized model for eco routing. At last, we present results and discussion.

2.3.1 Problem Definition

The route calculator shall calculate a single route for a single truck. Each call to the route calculator shall produce a result which contains all information necessary for subsequent modules. The route calculator has to fulfill several technical requirements – for the full list, see [25]. The major functional requirements are:

- Calculate only legal routes for trucks. All routes shall be drivable according to European and country-specific law.
- Calculate a single route of length 1000 kilometers in less than 15 seconds.
- Provide optimization criteria „short“, „fast“ and „economic“.
- Support time (weekday, time of day) as input for starting/arrival time.
- Support a vehicle model for truck-specific constraints.
- Provide an option to consider traffic information.

Truck-specific restrictions have to be evaluated by the route calculation. These restrictions are modeled as attributes (e.g. Turning, vehicle type=Truck, Authorization).

2.3.2 Simplifications

In order to achieve all stated requirements, the following simplifications of the problem have been considered:

2.3.2.1 Truck Routes with Focus on Highways

The route calculator evaluates only roads which are part of the European highway road network. These roads are classified as “functional road class” FRC=0 or FRC=1. This reduces heavily the number of links that have to be evaluated.

2.3.2.2 Weather Integrated in RTTI

Weather data and risk factors, as described in T4.2 “Provision of weather and road maintenance risk factors”, are not used as separate modules/interfaces by the route calculator. Instead, the RTTI service (see section 2.2.1) provides all traffic-relevant incidents (including weather and risks) in an integrated and concise way. This allows for a uniform processing of all dynamic data.

By using the RTTI service as single data source, there exists no possible inconsistency of information from parallel data sources. Contention resolution or data fusion is not necessary. For example, the route calculator does not have to solve the task of deriving a resulting speed value for a traffic congestion (sent via RTTI) and a weather incident (e.g. black ice, thunderstorm or heavy rain).

2.3.2.3 Simplified Fuel Model

We use a simplified fuel calculation model within the route calculation. A detailed fuel model is developed in Task 4.3 and serves for further calculation steps in the Optimization Engine. Reasoning:

1. The fuel cost (absolute value in liters) for a road network link in the size of 100-2000 meters is heavily affected by vehicle speed when entering/leaving the link. The algorithm complexity would increase because longer link sequences would have to be evaluated for adequate results.
2. The route calculation does not require absolute values to compute the most fuel-efficient route. It is sufficient to calculate relative values (as long as these are correct with respect to “greater/smaller comparison” for ranking route alternatives). A bias in total cost does not necessarily change the resulting route.
3. Parallel activity: fuel model was under development, compare Gantt chart on page 14 [1].

Instead, we follow the eco routing approach proposed by NDS.

2.3.3 NDS Eco Routing

The following concept is based on the available Navigation Data Standard for eco routing and the approach described in [4]. The NDS consortium proposes an approach for fuel efficient routing. This approach is composed of a simplified vehicle model and relies on a small set of pre-compiled map attributes and several vehicle-specific factors. For an in-depth description with formulas and diagrams, see [4], chapters 6 “Eco Routing” and section 9.9 “Data Structures for Eco Routing”.

The goal of the model is to achieve good routing results (both in calculation time and eco-route quality). This is achieved by evaluating at runtime only simple formulas (no complex or iterative calculations) and a small set of attributes.

The vehicle model is shown below and simplified compared with the Scania fuel model. The required main parameters are shown in Table 6. Most notably these are vehicle mass and front area (for air drag coefficient).

<i>Parameter</i>	<i>Value</i>
<i>Name</i>	Scania R480 LA4x2
<i>Weight</i>	7500 kg (truck without trailer)
<i>EnginePower</i>	480 hp (358 kW)
<i>FrontArea</i>	≈ 9.0 m ²
<i>Length</i>	17.2 m (with trailer)
<i>Width</i>	2.5 m (wheel sides)
<i>VehicleLoad</i>	28,000 kg (trailer + cargo ≈ 8 + 20)

Table 6: Truck vehicle parameters

To calculate the link weight according to the NDS eco routing proposal, the information summarized in Table 7 gets evaluated.

<i>Map-dependent attributes</i>	<i>Vehicle-dependent parameters</i>
<i>Speed (average speed, historic speed profile, legal speed limits, RTTI)</i>	Consumption speed dependency curve
<i>Transitions (speed changes along road and at intersections)</i>	Vehicle mass and related consumption factors
<i>Curvature</i>	Curve speed limit function
<i>Slope</i>	Excess slope threshold

Table 7: Eco routing influence factors

For the comparison of the fuel consumption of two paths connecting the same pair of intersections, only slopes exceeding a certain threshold have to be considered. The impact of acceleration and deceleration due to curves, transitions, speed limits, traffic lights, uphill-climbs and downhill braking on the fuel consumption is reflected. The dependency of absolute vehicle speed (in the form of air drag, roll resistance, engine and drivetrain losses) on the fuel consumption is also evaluated.

2.3.4 Results

We selected the traditional A* route calculation algorithm described in section 2.1.3.3 as basis for the development. The basic A* is well-understood and widely used in the navigation industry.

2.3.4.1 Hierarchical search

The route-calculator uses the multi-level structures of NDS, which effectively reduces the number of search tree elements for long-distance routes (compared to a non-hierarchical search, e.g. only using NDS-level 13).

2.3.4.2 Unidirectional search

The algorithm runs a unidirectional search. It holds a single search tree root. From this root node, it expands adjacent links and calculates path weights according to section 2.1.3.2. Travel time gets propagated along the path. It is used for resolving matching speed information from historic traffic data (map attribute “speed profile”).

2.3.4.3 Forward vs. reverse search

In case the truck departure time is given, the algorithm is initialized as a forward search with the start coordinate as root of the search tree. Estimated time of arrival (ETA) is calculated for the destination coordinate by summarizing estimated link travel times.

In case the truck arrival time is given, the algorithm is initialized as a reverse search with the destination coordinate as root of the search tree. Estimated time of departure (ETD) is calculated for the start coordinate by summarizing estimated link travel times.

In both cases, estimated link travel times are calculated as link length divided by best-available speed information. Best-available speed means we are applying live traffic data on top of historic speed profile on top of legal and average speed information.

2.3.4.4 Optimization criteria, cost functions and heuristics

We have implemented the cost functions “short”, “fast” and “eco” with matching heuristics to optimize the calculated routes for either “minimum route length”, “minimum travel time” or “minimum fuel consumption”, compare Table 8.

Optimization criterion	Cost function	Heuristic (estimated remaining cost)
minimum route length	Short, length[m]	Airline distance [m]
minimum travel time	Fast, time[s] (length/speed)	Airline distance / v_{\max} [s]
minimum fuel consumption	Economic, fuel estimate	Airline distance / fc_{\max}

Table 8: Routing cost functions

Each heuristic provides an estimate for the remaining cost between current search node and search destination. It controls the beam-forming of the informed A* search tree, compare Figure 4.

In order to produce minimum-cost paths the heuristic must not overestimate the actual cost. It serves as lower bound for the total cost of the route, given the specific optimization criterion. Only when this rule is followed, A* will produce identical result paths as Dijkstra. It will terminate earlier and with a smaller search tree, resulting in faster computation time.

In all cases, the airline distance (“as the crow flies”) is used as base heuristic. It has to be transformed to a cost unit matching the respective domain. This is done by normalizing the value with the maximum vehicle speed v_{\max} (the vehicle cannot go any faster!) or a normalized upper bound for the fuel consumption fc_{\max} (in liters per kilometer).

For the “Economic” cost function, the fuel estimate is calculated according to the NDS approach explained in section 2.3.3.

2.3.4.5 Summary

We have developed a route calculator that complies with the problem definition given in section 2.3.1 under the simplifications explained in section 2.3.2.

The produced “route result” contains road geometry (including slope and curvature profile), speed information (map-based as well as applied RTTI), network identifiers (path link references) and aggregated data (total route length, estimated time of arrival).

We have assessed the calculated routes by means of both an automated test suite and a manual interactive selection of start/destination pairs. All routes are legally drivable, no suicide driving in one-way roads or illegal turns were observed. Total time for the route calculation is below the threshold of 15s/1000km.

All three optimization criteria were implemented. Depending on the selection of start/destination and the road network topology, they show significantly different resulting routes. Still, it may occur that the 3 calculated routes are identical.

3 System Components

In this chapter we describe the architectural details of the route calculation system. The resulting system consists of a digital map, a routing engine and a traffic monitor. The digital map contains the static road network data based on an NDS map and the routing engine contains several algorithms for the calculation of routes. Both parts are described in section 3.1. The last section 3.2 describes the monitoring of traffic information and evaluation of effects on already planned routes.

3.1 Route Calculation Engine

The route calculation engine (RCE) is integrated into the Azure platform and provides a routing interface. It's internal architecture, integration and major use-cases are described in the following.

3.1.1 Internal Architecture

The Route Calculation Engine (RCE) is implemented in java. The access to the NDS map is provided by the Digital Reality (DR) framework. Parallel to the route-calculator an RTTI-listener component retrieves and processes the RTTI from TomTom.

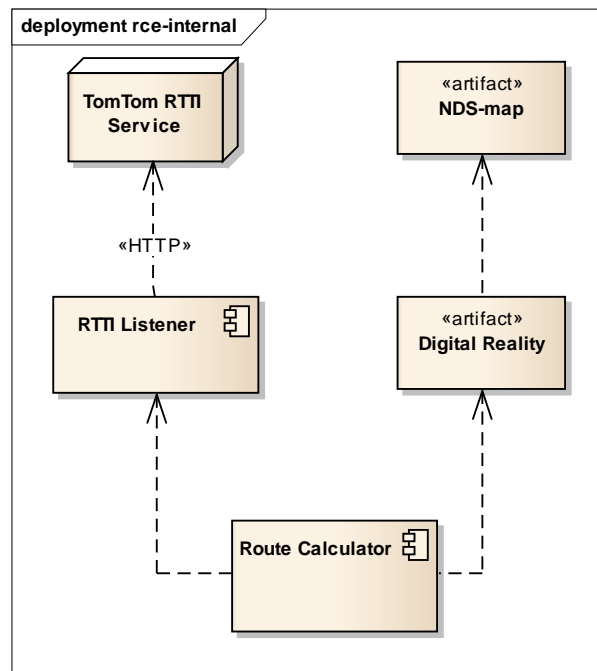


Figure 14: Internal architecture of a RCE instance

3.1.1.1 Map Access

The Route Calculation Engine (RCE) is based on the Digital Reality (DR) map framework. This framework is used to access the NDS map database (format spec, see [4]), which is handled by a separate access layer called “dynamic-map”. It takes care of decoding NDS structures (e.g. Update Regions per Country, levels and tiling scheme, routing tiles and attribute maps) to provide a network for routing. Network links get decoded with full attribution (NDS fixed and flexible attributes). Links and attributes can be used within the cost function of the routing algorithm.

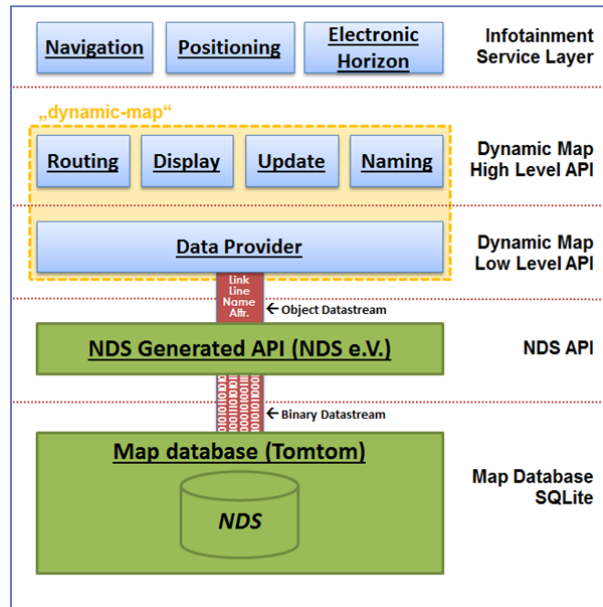


Figure 15: Digital Reality framework

3.1.1.2 Route calculation

For COMPANION a unidirectional A* route-calculator is used as described in Section 2. The route-calculator uses the multi-level structures of NDS.

3.1.1.3 RTTI integration

Next to the RCE a RTTI-listener component is deployed. This listener retrieves the RTTI-data from the TomTom server and processes it so that the route calculator can use it to map the incidents to the roads network. During the route-calculation the incidents on the route are evaluated to correct the driving time derived from the static map-data.

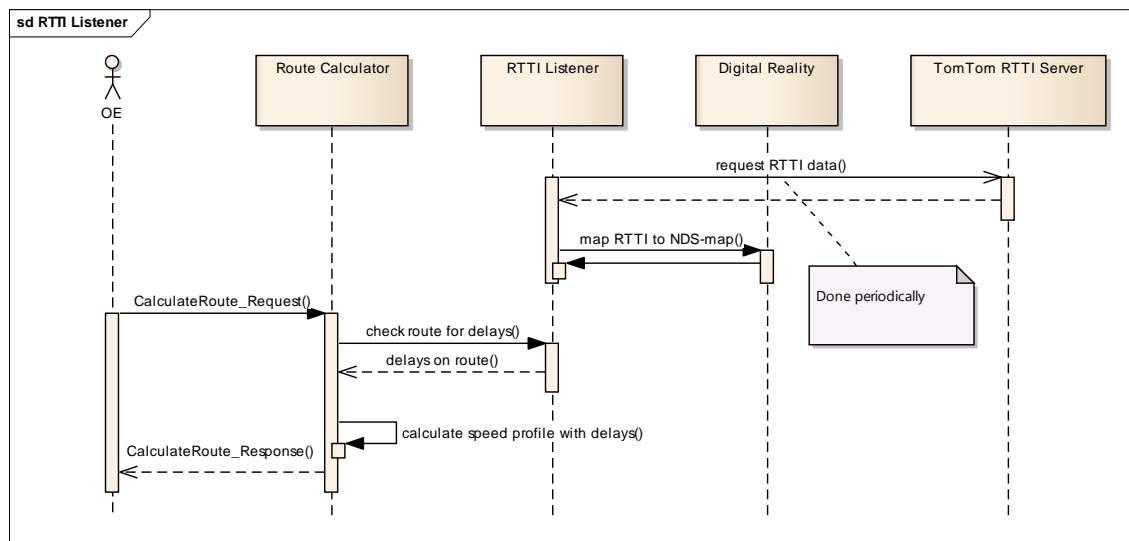


Figure 16: Evaluating RTTI during route calculation

3.1.2 MS Azure integration

As a result of WP 3, Microsoft Azure [18] has been declared as the platform for the COMPANION off-board system. Microsoft Azure is a cloud platform that provides a lot of flexible services like:

- Deploy Virtual machines (Linux/Windows)
- Communication (Enterprise Service Bus)
- Storage (in Tables, Blobs)
- SQL-Databases
- Management
- Load-Balancing

Within this platform all services for the off-board-system from the different project partners can be deployed. There are two ways to deploy software within the cloud:

- Infrastructure as a Service (IaaS)
In this setup Azure provides only the (virtualized) hardware. All software including the operating system has to be configured, managed and updated manually.
- Platform as a Service (PaaS)
In this setup the software is bundled with deployment-information, the needed data, etc. and will be deployed on a system managed by Azure.

An abstracted service like the Route Calculation Engine (RCE) is called Cloud Service. The service itself is provided by multiple Worker Role instances. A Worker Role is one instance of the service implementation. For each service Azure provides load-balancing, dynamic scaling, staging, etc.

3.1.2.1 Cloud Service Architecture

The RCE is provided as a Cloud Service. Internally the Cloud Service is realized by one or more Worker Roles. Requests are sent to the Cloud Service via the Enterprise Service Bus (ESB).

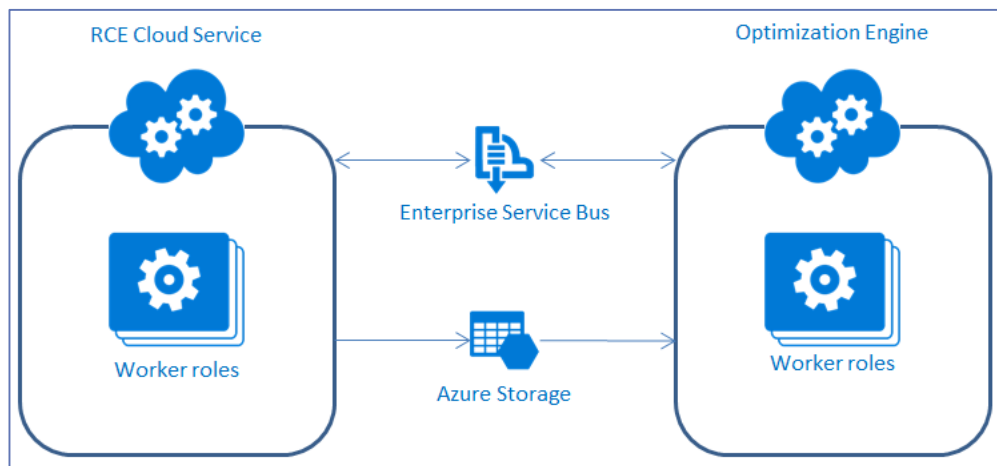


Figure 17: RCE as cloud service

The cloud service communicates with the other components like for example the Monitoring Engine via the Enterprise Service Bus (ESB). The ESB is an infrastructure within Azure for the queuing and

distribution of messages between services. The ESB offers message queues for sending messages to a single recipient and topics for sending messages to multiple recipients.

3.1.2.2 Enterprise Service Bus interface

The communication of the RCE with the other parts of the off-board system is done via messages send over the Enterprise Service Bus.

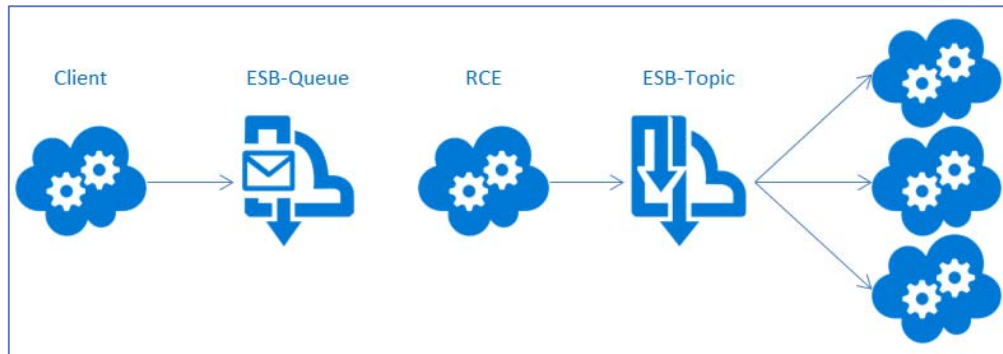


Figure 18: ESB communication scheme

Incoming requests to the RCE are sent to the ESB-queue (Table 9) and processed by an RCE-worker role. All responses by the RCE are sent to the common ESB-topic (Table 10) and can therefore be received by any service that registers for this topic.

Name	Value
ESB-namespace	CompanionCommon
ESB-queue-name	companion_rce_inputqueue

Table 9: ESB-input-queue of the RCE

Name	Value
ESB-namespace	CompanionCommon
ESB-topic	companion_common_topic

Table 10: Common ESB-topic

All messages within the COMPANION off-board system contain the generic fields shared by all ESB-messages (see Table 11).

Attribute	DataType	Description
Body	String	This field contains the payload specific to the message.
CallId	String	A unique id that will be used to correlate the corresponding answers.
MessageBodyLocation	String	Contains either "Message" if the payload is directly stored within the Body field or "URL" if the Body contains an URL to the data.
MessageType	String	Name of the message e.g. "CalculateRoute_Request".
Recipient	String	The recipient of this Message as defined in [23].

Table 11: The generic fields of the ESB-messages

The standard call-sequence of the RCE is shown in Figure 19.

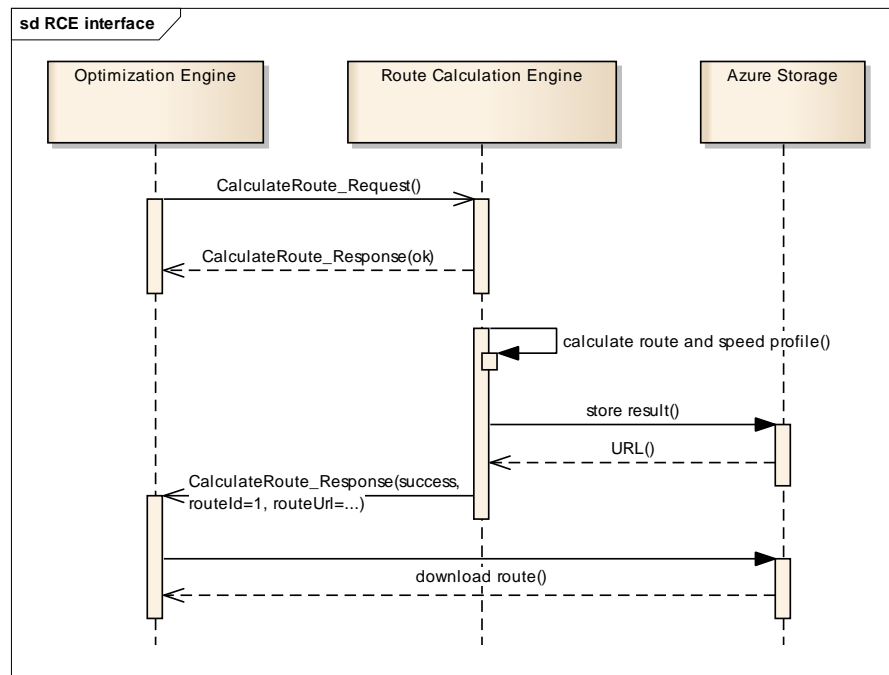


Figure 19: Standard RCE call-sequence

The RCE supports two messages. The first one is the CalculateRoute_Request sent to the RCE containing all information required to calculate the route e.g. start and destination. The CalculateRoute_Response message is broadcasted by the RCE to an ESB-topic and contains status-updates of the route calculations. After the route has been calculated the data containing the location and speed-profile of the route is stored in the Azure Blob-Storage.



Figure 20: Messages of the RCE

3.1.2.3 Requesting a route

A route calculation is initiated by an ESB-message of the type "CalculateRoute_Request". This message contains a RouteRequestData (Table 12) object encoded as JSON structure within the message body.

The RouteRequestData object contains the data needed to perform the route calculation, for example start, destination, departure and/or arrival time (see Table 12).

Attribute	DataType	Description
startLon	double	Decimal degrees of the wgs84-longitude of the start location.
startLat	double	Decimal degrees of the wgs84-latitude of the start location.
destLon	double	Decimal degrees of the wgs84-longitude of the destination.
destLat	double	Decimal degrees of the wgs84-latitude of the destination.
truckWeightInMetricTons	double	The recipient of this Message. Should be "RCE" as defined in [1].
departureTime	long	Unix-timestamp (UTC) of the departure time. If the arrivalTime is set and the departureTime is empty the departureTime will be calculated based on the arrivalTime.
arrivalTime	long	Unix-timestamp (UTC) of the planned arrival time.
useLiveTraffic	boolean	Live traffic data is used to determine the best route if this flag is activated.
routeType	String	"fastest", "shortest" or "eco".

Table 12: RouteRequestData

The initial request is answered with a response of result code "ok" and the assigned route-id if all input data is valid. After the route is calculated a message containing the result code "success" is sent in combination with URL to retrieve the route from the Azure storage.

Attribute	DataType	Description
resultCode	String	"ok", "error" or "success".
routeUrl	String	In case of "success" this field contains the link to the storage-blob where the route is stored.
drivingTimeSec	int	In case of "success" this field contains the estimated travel time along the route in seconds.
calculationTimeSec	int	In case of "success" this field contains the calculation time in seconds.
routeLengthMeters	double	In case of "success" this field contains the length of the route in meters.
eta	long	In case of "success" this field contains a Unix-timestamp (UTC) of the estimated arrival time.
errorMessage	String	In case of "error" this field contains an error message.
routeId	String	Unique identifier of the route.

Table 13: RouteResultData

The data of the calculated routes is stored within the Azure Blob Storage. This allows the storage and retrieval of binary data BLOBs paired with metadata. Table 14 lists all the metadata that is stored with the route.

Value	example
eta	2015-07-20T20:12:40.598Z
startLon	9.411978721618652
startLat	54.74736130731005
destLon	11.077136993408203
destLat	47.49046126751614

departure	2015-07-20T10:00:23.607Z
routeLength	1073564.1376245022
callId	fdcf8956-9764-41ca-8a37-12a22e79e2bf
routeType	FASTEST
useLiveTraffic	true
calculationDurationSec	14
truckWeightInMetricTons	1300

Table 14: Metadata stored with each RCE result

The route data holds information such as geometry, altitude and speed profile of the route. It consists of road sections which represent a link in the NDS map.

Note: typical RoadSection length: 100 .. 5000 m. See also Figure 22.

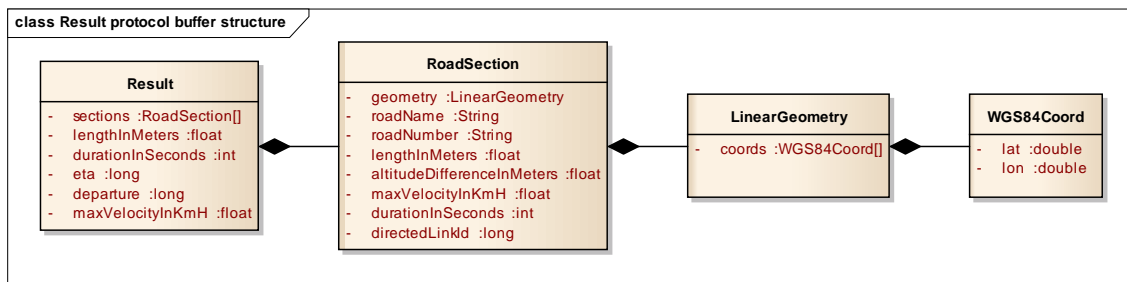


Figure 21: Structure of the stored calculation result

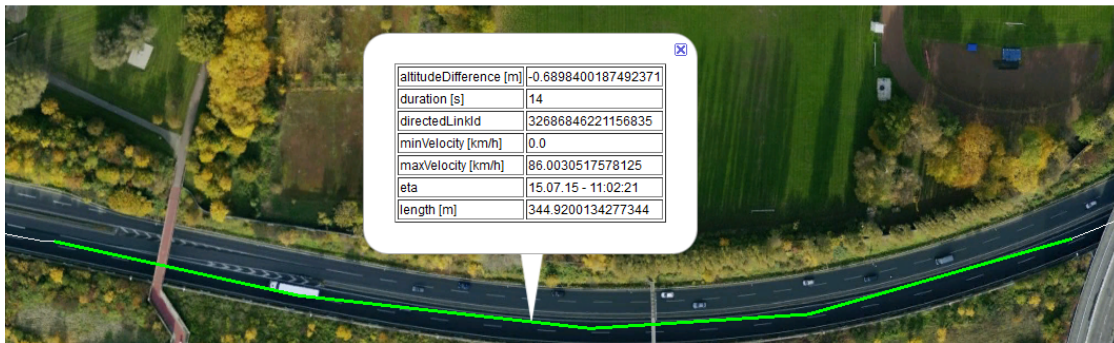


Figure 22: Example of a Road Section

The data structure shown in Figure 21 is serialized with Google Protocol Buffers [24]. This offers fast serialization and deserialization libraries for many programming languages.

The serialized Result is stored within the Azure storage.

3.1.2.4 Management, Reporting

Cloud services can be scaled automatically based on CPU-usage or number of messages in a queue. As soon as the peak-load within the COMPANION system is known we can provide as many worker-roles as needed. These can be activated on demand by the Azure platform. Currently no common way for logging and collecting diagnostic data is defined. There are multiple ways in Azure to collect these data centrally:

- Manual logging of data into a Azure storage container
- Windows Azure Diagnostics (WAD)

WAD is currently not supported for java so this solution is not usable within the RCE. The simple solution is to use a Custom Log directory for logging from the java-processes and let Azure persist this folder to a BLOB container.

3.1.3 System Start-up / Initialization

At the startup of a RCE-worker-role the current RTTI data is requested and processed. Additionally the map data is pre-cached to the configured degree. The initialization for a RCE-worker-role takes about 440 seconds and uses about 1.8 GB RAM.

#	Step	Duration [s]
1	Initial Parsing of RTTI-data	300
2	Precaching of binary map data	140

Table 15: Startup-times

3.1.4 Major Use cases

This section describes the three major use cases the route-calculation component of the RCE is used for by the Monitoring Engine.

3.1.4.1 Use case 1 Estimate travel time

This is a pre-planning that finds a nominal route and narrows down the departure time. The RCE is called with start and destination and with the latest time of arrival. Based on this information the route is calculated and an initial departure time is calculated.

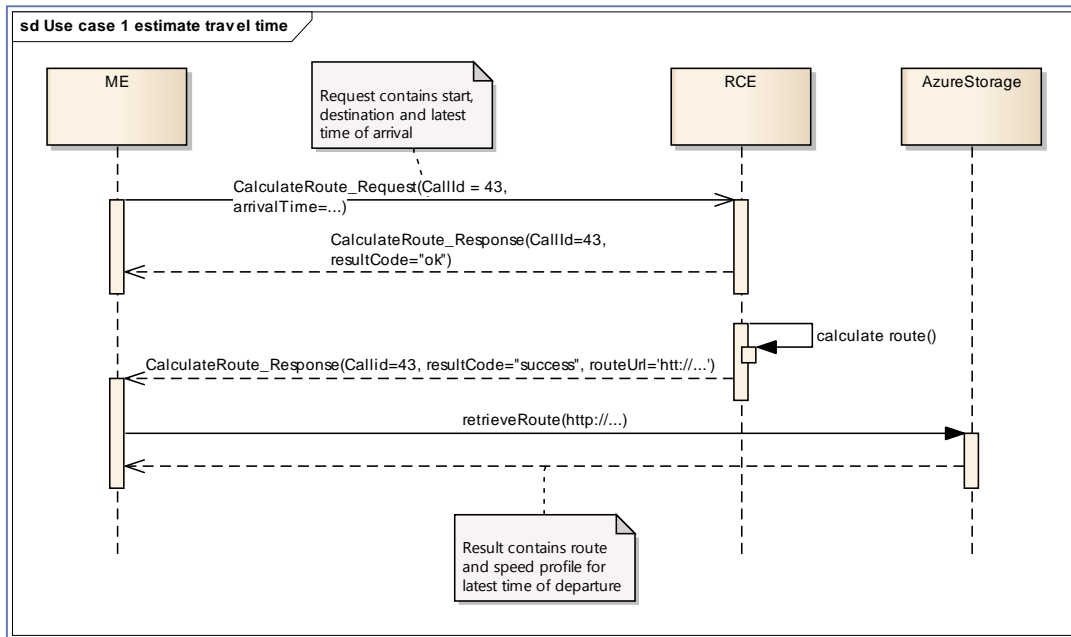


Figure 23: Sequence of Use case 1: Estimate travel time

The RCE determines the maximum possible speed for each link based on

- Speed limits
- Absolute maximum vehicle speed (e.g. 90 km/h for trucks)
- Historic speed profile
- Flow speed reported by RTTI

Based on this, the latest time of departure can be calculated. The Monitoring Engine can use this value and reduce the speed as a necessary factor to allow optimal platooning of multiple trucks.

3.1.4.2 Use case 2 update Speed profile

After the initial route is planned the speed profile for the optimal departure time is calculated. This represents the actual upper speed limit for the truck assignment. Based on the fuel consumption model and the planned platooning steps the truck has to drive at a lower speed (this is determined within monitoring engine).

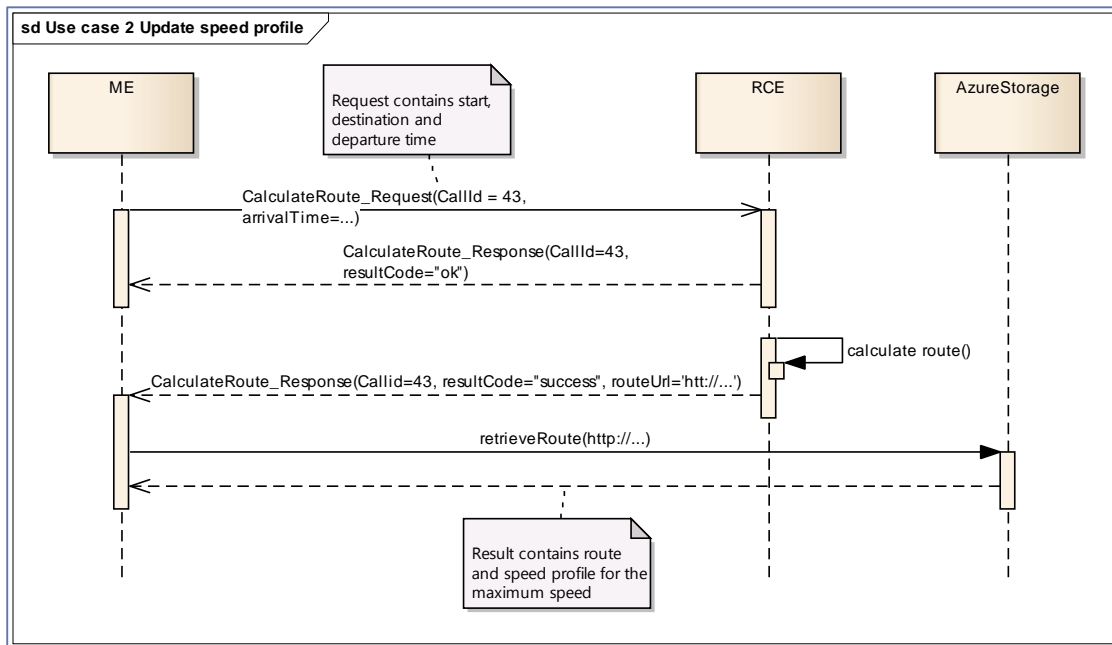


Figure 24: Sequence of Use case 2: update speed profile

3.1.4.3 Use case 3 Recalculation

This use case applies when a truck is not following the plan for some reason.

1. The truck is delayed due to some unpredicted events and this is reported from the Fleet Manager.
2. There is a predicted change in travel time caused by changes in traffic, which is reported by the Traffic Monitor.

The Monitoring Engine will choose to trigger a new route-calculation from the current position to the first stop and update the speed profiles.

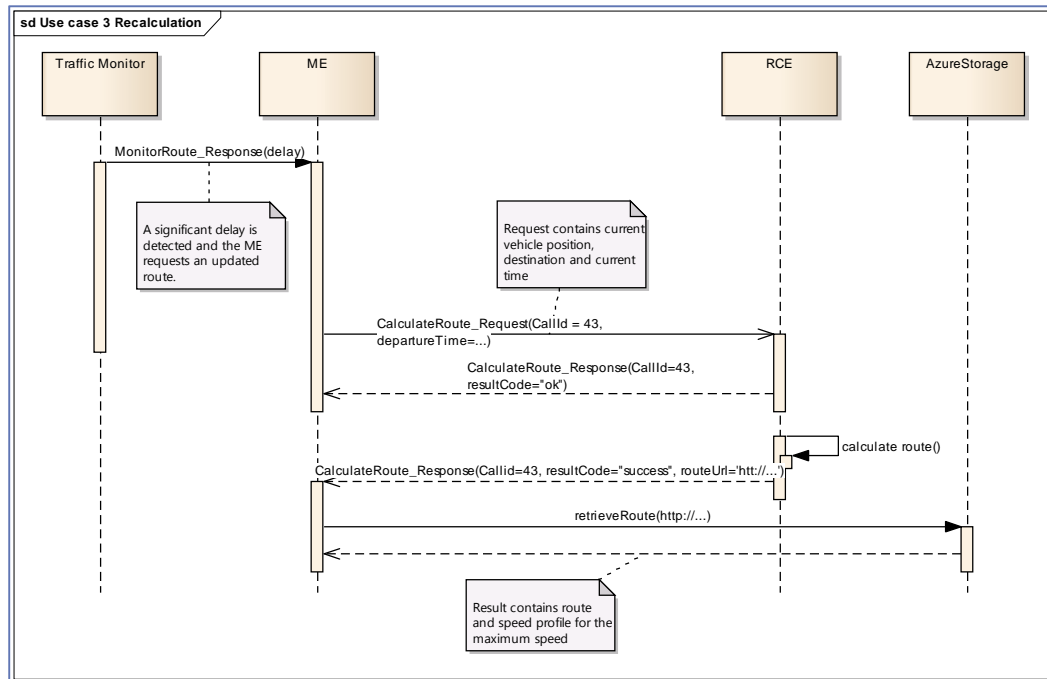


Figure 25: Sequence of Use case 3: Recalculation

3.2 Traffic Monitor

The Traffic Monitor is a module of the off-board system that monitors live traffic. Routes can be registered for monitoring. Any event en-route e.g. delays due to traffic events or blocked roads are reported. This section describes the internal structure of the Traffic Monitor, the integration into the Azure platform and the interfaces using the example of the main use cases.

3.2.1 Internal Architecture

The Traffic Monitor uses the same RTTI-listener component as the RCE, see section 3.1.1.3. The messages are cached and each registered route is checked for delays and blocked roads.

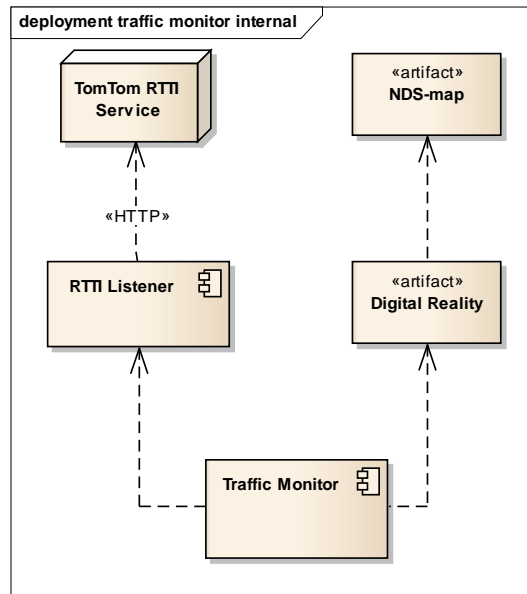


Figure 26: Internal architecture of a Traffic Monitor instance

3.2.2 MS Azure integration

The Traffic Monitor is implemented as cloud service. The communication is done via Enterprise Service Bus. Requests to the Traffic Monitor are sent to the ESB-queue described in Table 16 and processed by a Traffic-Monitor Worker Role.

Name	Value
ESB-namespace	CompanionCommon
ESB-queue-name	companion_traffic_monitor_inputqueue

Table 16: ESB-input-queue of the Traffic Monitor

Responses and detected delays are sent to the common ESB-topic described in Table 9: ESB-input-queue of the RCE

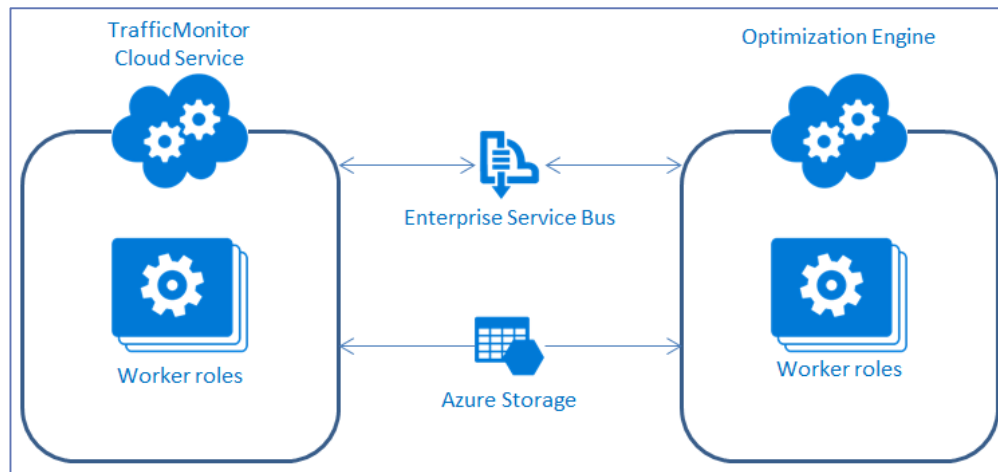


Figure 27: Traffic Monitor within the system

3.2.3 System operation

Each route calculated by the RCE can be assigned to the Traffic Monitor. Requests to the Traffic Monitor are sent to an ESB queue. Each Request contains the id of a route previously calculated by the RCE.

3.2.3.1 Register route for monitoring

After a route is calculated by the RCE, a unique route-id is returned. This can be used to register the route at the Traffic Monitor. The route is retrieved directly from the Azure Storage.

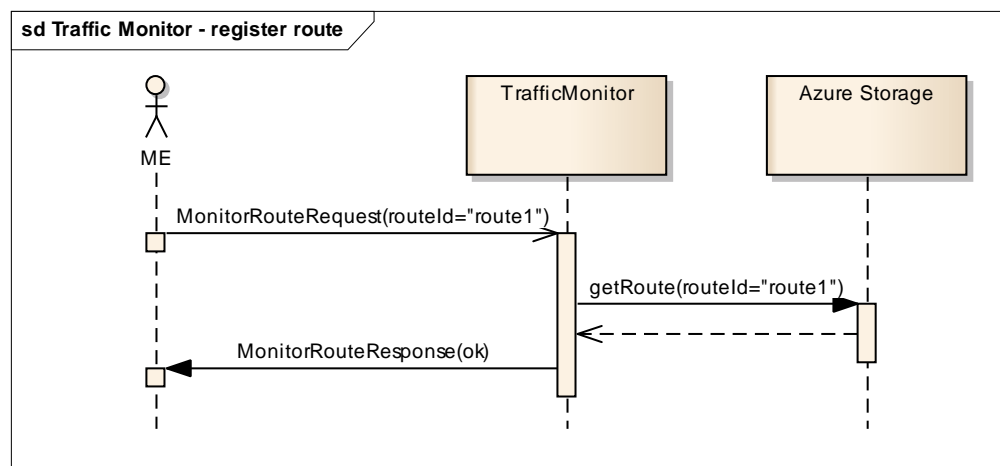


Figure 28: Register route at Traffic Monitor

3.2.3.2 Report detected delays or blocked roads

Delays or blocked road segments are reported by sending messages to the ESB –topic. All services interested on Traffic events can register for this topic and receive the messages for all routes.

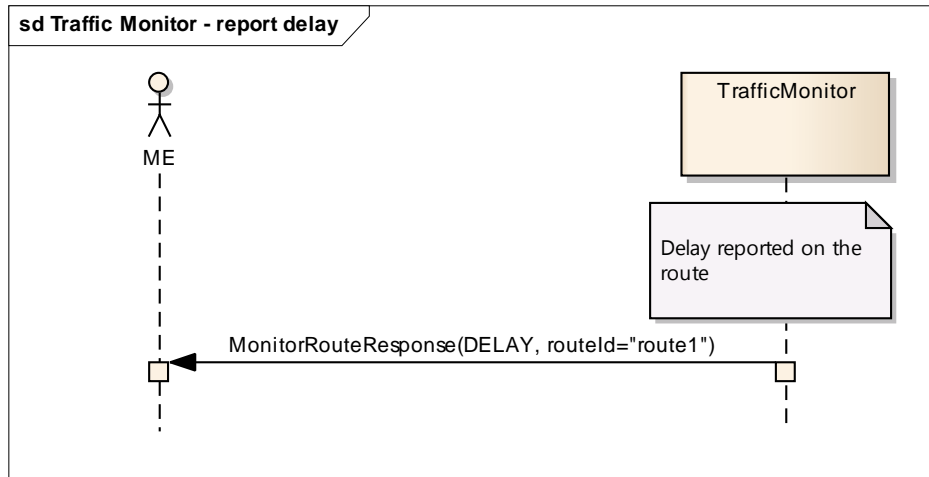


Figure 29: Reporting a delay

3.2.4 Messages

The communication of the Traffic Monitor with the other parts of the off-board system is also done via ESB messages.

The Traffic Monitor supports two messages; the first one is the MonitorRoute_Request sent to the Traffic Monitor containing the route that should be tracked. The MonitorRoute_Response message is broadcasted by the Traffic Monitor to an ESB-topic and contains details of the reported delays.

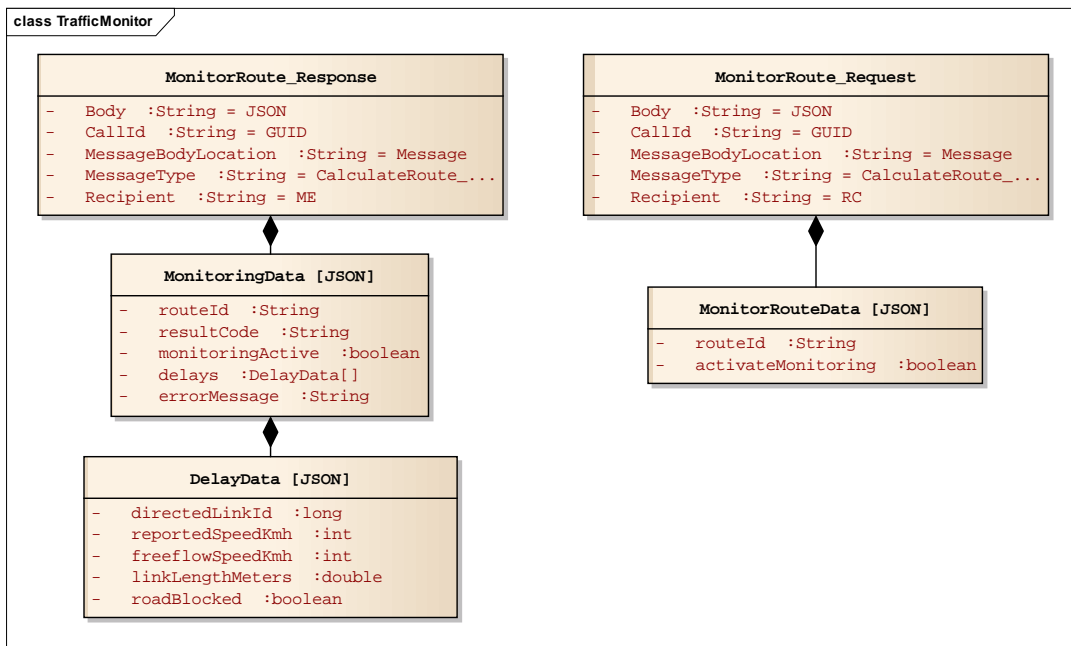


Figure 30: Messages of the Traffic Monitor

Attribute	DataType	Description
<i>routeId</i>	String	Unique identifier of the route that should be monitored.
<i>activateMonitoring</i>	boolean	Controls if the monitoring of the route should be started or stopped.

Table 17: MonitorRouteData (JSON payload of the message MonitorRoute_Request)

Attribute	DataType	Description
<i>routeId</i>	String	Unique identifier of the route.
<i>resultCode</i>	String	"ok", "delay", "roadBlocked", "error"
<i>monitoringActive</i>	Boolean	Current state of monitoring for the routeId.
<i>Delays</i>	DelayData[]	In case of "delay" this field contains the delay-details. In case of "roadBlocked" this field contains the list of blocked road sections.
<i>errorMessage</i>	String	In case of "error" this field contains an error message.

Table 18: MonitoringData (JSON payload of the message MonitorRoute_Response)

Attribute	DataType	Description
<i>directedLinkId</i>	long	The id of the link on that the delay has been reported.
<i>reportedSpeedKmh</i>	int	The reported speed on the link in km/h.
<i>freeflowSpeedKmh</i>	int	The freeflow speed on the link in km/h.
<i>linkLengthmeters</i>	double	The length of the link in meters.
<i>roadBlocked</i>	boolean	True if the link is blocked.

Table 19: IncidentData (JSON structure referenced in MonitoringData)

4 Conclusion

In this research we have investigated and confirmed the use of high quality NDS maps for the development of a route calculation engine for fuel-efficient routes for heavy duty vehicles. We examined different variations for routing algorithms. We have decided to use a unidirectional A* algorithm with a fuel model based on the NDS eco routing proposal.

Time-dependent road network attributes such as (truck-) restrictions and legal speed limits are taken into account to allow a realistic and reliable planning of platoons. We use additional dynamic data in the form of “Real Time Traffic Information” to adapt to current and predicted traffic situations. Useful simplifications have been made which enable a time efficient and fast calculation.

The resulting fuel efficient route calculator is successfully implemented within a software component “Route Calculation Engine” (RCE). The developed system can calculate routes from a starting location to a destination in two ways to support several use cases:

Forward planning: with a provided time of departure the arrival time gets estimated (ETA).

Reverse planning: with a provided time of arrival (delivery deadline) the departure time gets estimated.

The RCE is successfully deployed as part of the off-board system for platoon coordination and developed as a stateless routing service in AZURE, which allows to be easily scaled up for future demands. The result of the calculation is applicable as input for the scheduling of platoons which is developed in Task 5.2.

Furthermore a monitoring component shall supervise routes for incoming traffic events within the COMPANION off-board system. Routes can be registered for monitoring and any event en-route e.g. delays due to traffic events or blocked roads will be reported to be used in the off-board system for further decisions.

The main challenges of our approach are the performance of the “path finding” on the European scale, the processing of the live traffic information and the consideration of additional data for fuel efficiency.

As the project advances and the number of trucks and platoons increase we expect more feedback to the calculation results and their application. A further optimization phase will be part of our future work and will be executed before the final demonstration.

References

- [1] Annex 1 – “Description of work”, COMPANION project, EU Seventh framework programme, 2013.
- [2] Y. Zhao: Vehicle Location and Navigation Systems, Artech House, Boston 1997
- [3] The NDS-Standard (Homepage). [Online; URL: <http://www.nds-association.org/the-nds-standard/>]
- [4] Navigation Data Standard Format Specification NDS Version 2.3.2 (confidential)
- [5] Navigation Data Standard Compiler Interoperability Specification NDS Version 2.3.2 (confidential)
- [6] Martin Holzer, Frank Schulz, Dorothea Wagner. Engineering Multi-Level Overlay Graphs for Shortest-Path Queries, SIAM (2006)
- [7] Description of the Dijkstra algorithm: Edsger W. Dijkstra, A Note on Two Problems in Connexion with Graphs, Numerische Mathematik 1, 1959.
- [8] A* Algorithm description: P. E. Hart, N. J. Nilsson, B. Raphael: A Formal Basis for the Heuristic Determination of Minimum Cost Paths, IEEE Transactions on Systems Science and Cybernetics , 1968.
- [9] Giacomo Nannicini, Daniel Delling, Leo Liberti, Dominik Schultes. Bidirectional A* Search for Time-Dependent Fast Paths, 7th Workshop on Experimental Algorithms (2008)
- [10] Dominik Schultes. Route Planning in Road Networks, Ph.D. thesis TH Karlsruhe (2008)
- [11] Daniel Delling, Dorothea Wagner. Time-Dependent Route Planning , Robust and Online Large-Scale (2009)
- [12] Daniel Delling. Time-Dependent SHARC-Routing, Journal Algorithmica Volume 60 Issue 1, (2011)
- [13] INRIX Traffic Information. [Online; URL: Real-Time & Predictive Traffic, <http://inrix.com/xd-traffic/>]
- [14] TOMTOM Traffic Information. [Online; URL: <http://livetraffic.tomtom.com/>]
- [15] Google Traffic Information: “The bright side of sitting in traffic: Crowdsourcing road congestion data”. [Online; URL: <http://googleblog.blogspot.de/2009/08/bright-side-of-sitting-in-traffic.html>]
- [16] Simon Kwoczek, Sergio Di Martino, Wolfgang Nejdl. Predicting and visualizing traffic congestion in the presence of planned special events, Journal of Visual Languages & Computing (2014)
- [17] TomTom International Whitepaper – How TomTom's HD Traffic and IQ Routes Data Provides the Very Best Routing, Technical report, TomTom International, 2009.
- [18] Microsoft Azure cloud platform. [Online; URL: <https://azure.microsoft.com/>]
- [19] Datex2 standard. [Online; URL: <http://www.datex2.eu/>]
- [20] OpenLr location-referencing-method. [Online; URL: <http://www.openlr.org/>]
- [21] REST-architecture description. [Online; URL: https://de.wikipedia.org/wiki/Representational_State_Transfer]
- [22] Dr. Andreas Sasse, Dr. Markus Kerper, Dr. Sergio di Martino, Silviu Homoceanu. Fahrstreifengenaue und zustandsbasierte Verkehrsinformationen durch Auswertung von Fahrzeugsensordaten, AAET 2015

- [23] COMPANION-internal module names. [Online; URL:
https://vprojects.offis.de/redmine/projects/platoon/wiki/Subscriptions_and_Filters]
- [24] Google protocol-buffers project. [Online; URL: <https://developers.google.com/protocol-buffers/>]
- [25] D2.4: COMPANION D2.4 “Technical Requirements” (result of Task 2-4)

Glossary

This table contains all abbreviations used in this document in alphabetical order.

3G	Short form of third generation of mobile telecommunications technology
CH	Contraction Hierarchies, route calculation algorithm
DR	Digital Reality, NDS based mapping toolkit by Volkswagen
ESB	Enterprise Service Bus
GNSS	Global navigation satellite system
GPS	Global Positioning System, absolute positioning with latitude and longitude
GPRS	General packet radio service
LTE	Long-Term Evolution, commonly marketed as 4G
ME	Monitoring Engine: this is a component of the COMPANION off-board system that uses the RCE to maximize the platooning-benefit for multiple assignments
NDS	Navigation Data Standard, automotive standard map data format
PSE	Planned Special Events (see [6])
RCE	Route Calculation Engine, module of the off-board system
REST	Representational State Transfer, a web services architecture
RTTI	Real Time Traffic Information
SPP	Shortest Path Problem
VM	Virtual Machine
WAD	Windows Azure Diagnostics
XFCD	Extended Floating Car Data, using vehicles as sensors for gathering traffic information

A Appendix

6.1 Restrictions and limitations of the demonstrator RCE

- The current RCE is implemented as REST-service. The realization as Cloud Service is implemented as a prototype.
- Only abnormal traffic-events and road-closures are evaluated by the Traffic Monitor and the RCE.
- The impact of weather on the traffic is covered by the traffic-messages provided by TomTom.
- The impact of Predictable Special Events is not evaluated in the demonstrator.
- For the determination of the speed profiles of the trucks the speed profiles for normal passenger vehicles are used up to an maximum speed of 89km/h.
- RTTI is evaluated for UK, France, Germany, Spain, Sweden, Belgium, Netherlands and Denmark.
- The Traffic Monitor will be implemented in Q4/2015.
- Further performance optimizations will be implemented upon project advances.