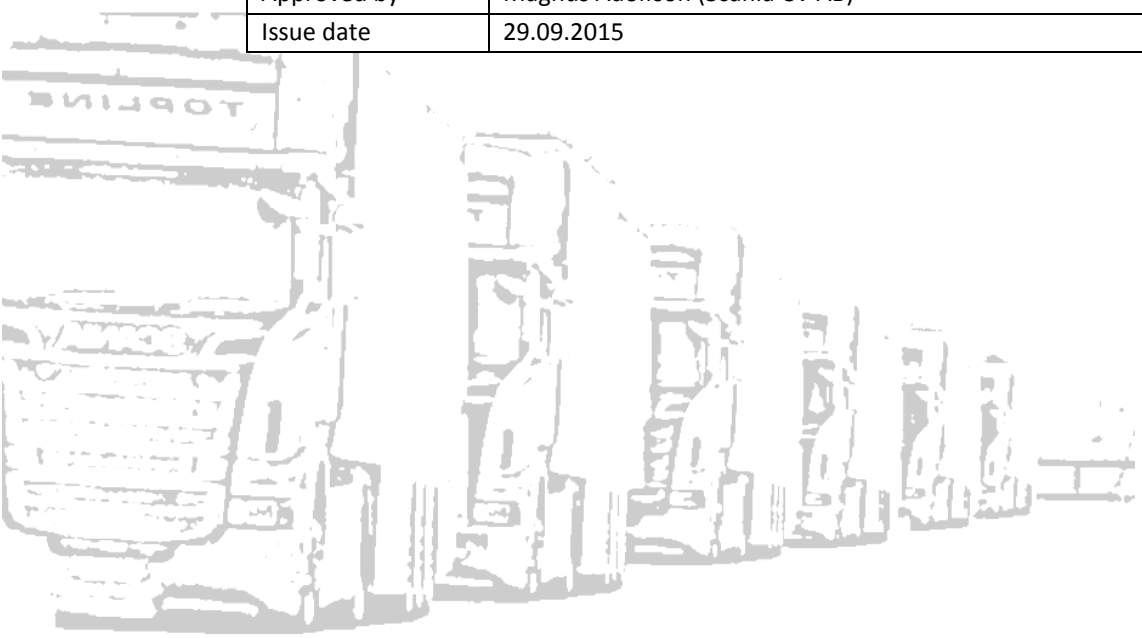
EUROPEAN COMMISSION
SEVENTH FRAMEWORK PROGRAMME
GA No. 610990

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

D3.1. Component Specifications for the Overall Architecture

Deliverable No.	COMPANION D3.1	
Deliverable Title	Component Specification for the Overall Architecture	
Dissemination level	Public	
Written By	Samuel Wickström (Scania CV AB)	18.10.2014
Checked by	Sönke Eilers (OFFIS)	25.09.2015
Approved by	Magnus Adolfson (Scania CV AB)	29.09.2015
Issue date	29.09.2015	



History log

Name	*Status*	*Version*	*Date*	*Summary of actions made*
Scania *Samuel Wickström*	Editor	1	18.10.2014	First draft. Scheme and descriptions
OFFIS *Sönke Eilers*	Contributor	1.1	09.03.2015	Overall Architecture, Main Flow of Information, History Log
Scania *Henrik Pettersson*	Contributor	1.2	24.04.2015	Edit Vicinity Control Layers and Road Segment Road Segment Governing Layer
OFFIS *Sönke Eilers*	Contributor	1.3	31.08.2015	Updated Overall Architecture Picture, References, Executive Summary
Scania *Henrik Pettersson*	Contributor	1.4	01.09.2015	Updated Backbone
OFFIS *Sönke Eilers*	Contributor	1.5	02.09.2015	Prepared Document for first Review
Volkswagen *Thilo Schaper*	Reviewer	1.5	04.09.2015	Clarifications and readability
OFFIS *Sönke Eilers*	Contributor	1.6	10.09.2015	Incorporated feedback and prepared for review by coordinator.
Scania *Samuel Wickström*	Contributor	1.7	16.09.2015	Changes of headers, text and graphics for a homogenous look and context logic.
Scania *Samuel Wickström*	Contributor	1.8	28.09.2015	Major overhaul. Nominal use cases moved to D3.2.

Executive summary

This deliverable *D3.1 Component specification of the overall architecture* describes the components of the overall architecture. The overall architecture consists of three main architectural layers: the Route Management layer, the Road Segment Governing layer and the Vicinity Control layer.

The Route Management layer comprises services for creating energy-optimized transports, including services for route planning, traffic prediction and platoon creation. These services create a strategic plan for the vehicle and are partitioned off-board.

The Road Segment Governing layer is concerned with fulfilling the plan in the best tactical way. This layer creates short term goals for the best possible fuel consumption along the current traveled segment and makes sure vehicles are able to meet their platooning partners on the road. In order to realize this the functionality is partitioned both off-board and on-board.

The Vicinity Control layer is a distributed control system in the trucks. This layer is all on-board and takes the previous layers output into operation. A key functionality is safe vehicle platooning. This feature maintains a safe distance between platooning partners and can be thought of as a cooperative automatic cruise control.

Contents

1. Introduction

The COMPANION project comprises functionality that ranges from strategic planning and optimization, weather and traffic predictions to real time vehicle control. It is a complex system partitioned in high performance server racks as well as in embedded vehicle computers. Hence, there is a need of an abstraction of the functionality in order to actually understand and design the system. The aim is to abstract the design into layers and functional components that can be developed independently, and isolated from the potential failure of each other.

Abstraction has been part of the scope of COMPANION since the creation of the project. It helped to formulate the challenges, and to find the right partners. Three levels, or layers, of abstraction were created from the start. In WP3 these three layers have been evolved into more tangible components and definitions. Nevertheless, a high level of abstraction has been kept throughout the design.

About this Document

This document represents the deliverable *D3.1 Component specification for the overall architecture*. The requirements have been derived from the user scenario and use cases defined in the WP2 deliverables D2.1 [1], D2.3 [2], D2.4 [3], D2.5 [4]. These deliverables are used to create an appropriate system design that will be packaged as deliverables D3.1 and D3.2. Deliverable D3.1 focuses on the actual components and how they relate to each other, while D3.2 [5] describes their interaction. These activities are essential for the subsequent design, implementation, testing, and validation of the actual COMPANION system. The flow is illustrated in Figure 1-1.

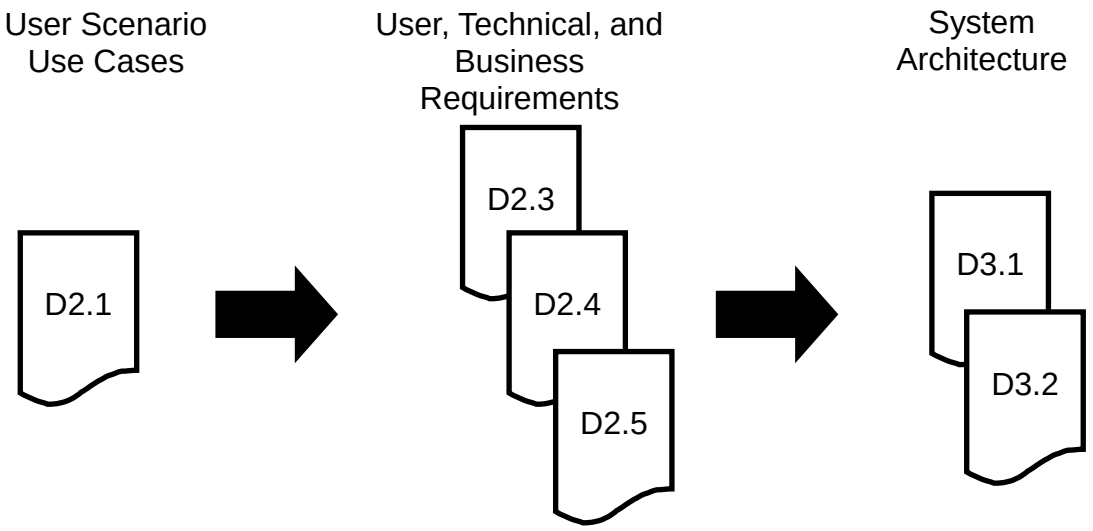


Figure 1-1 Overview of COMPANION deliverables.

This document is structured accordingly; the COMPANION layers, each layer and their specific components, and finally the supporting cross layer components.

2. COMPANION Layers

The complete partitioning of the COMPANION system into abstraction layers is depicted in Figure 2-1 below. It has three main layers with two sublevels for each layer, i.e. one could think of it as 6 layers. In the COMPANION system, the application starts at the top and finally finds its way down to the bottom layer where actuators are physically moving.

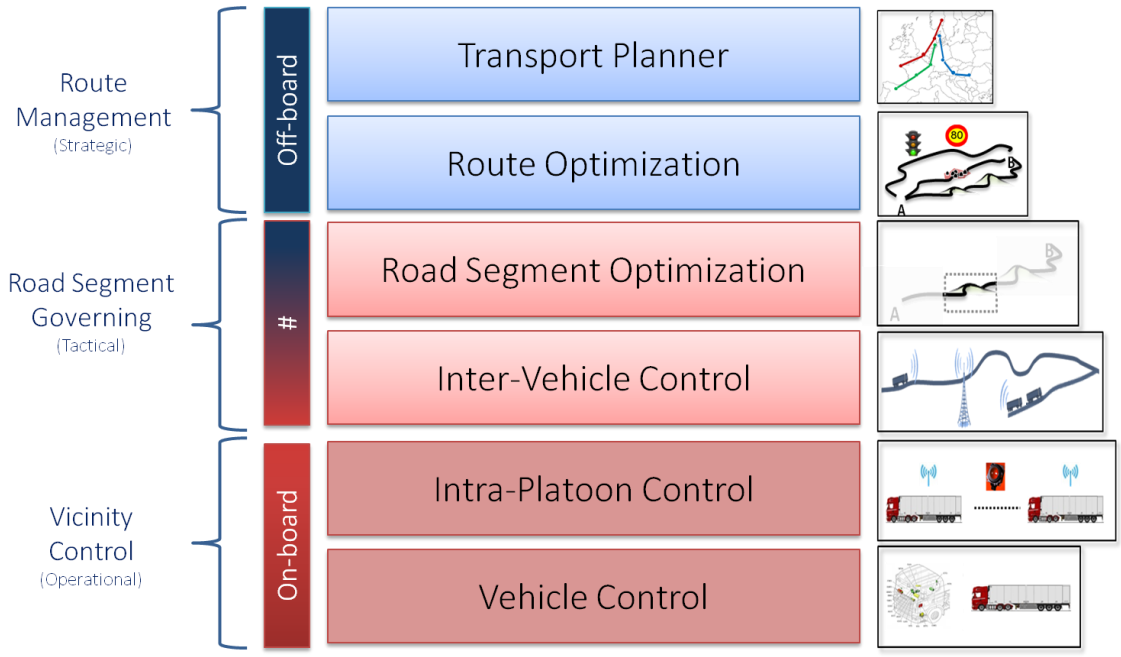


Figure 2-1 COMPANION Layers.

Three Layers

The three main layers are divided in such a way, that they easily can be labeled with different domain names and still have a valid level of abstraction. The common denominator is the temporal domain with tasks of the future at the top, and real-time actions at the bottom. Table 2-1 makes this clearer.

Table 2-1 The COMPANION layers with three different abstraction domains.

	Managerial Abstraction	***Executable Abstraction***	***Temporal Abstraction***
Top	Route Management	Strategic	Future
Middle	Road Segment Governing	Tactical	Soon
Bottom	Vicinity Control	Operational	Now

Three different designations are a product of how to communicate the level of abstraction to different people, partners and context. The managerial abstraction is the one commonly used since it is more specific in what each layer is doing functionally.

Future, Soon and Now

The COMPANION three-layer concept can also be depicted spatially. In Figure 2-2 a vehicle, depicted in the bottom left corner, is participating in a platoon, created by the COMPANION system.

The vehicle follows the route given by the route management layer. This top layer has a spatial horizon that stretches all the way to the vehicles final destination. It takes into account all events

affecting the vehicles path and speed along the way, e.g. traffic jams, joining and splitting of COMPANION platoons. All those tasks have to be decided long before the actual vehicle is there, hence this layer predicts as well as decides the future.

The vicinity control, the bottom layer, handles all decisions that involve the vehicles immediate surroundings. These are based on events sensed by the vehicles on-board systems and happens in real-time, i.e. now. Its spatial context is less than one kilometer a way, and within this horizon it performs the tasks given by the top layer, for example join a platoon.

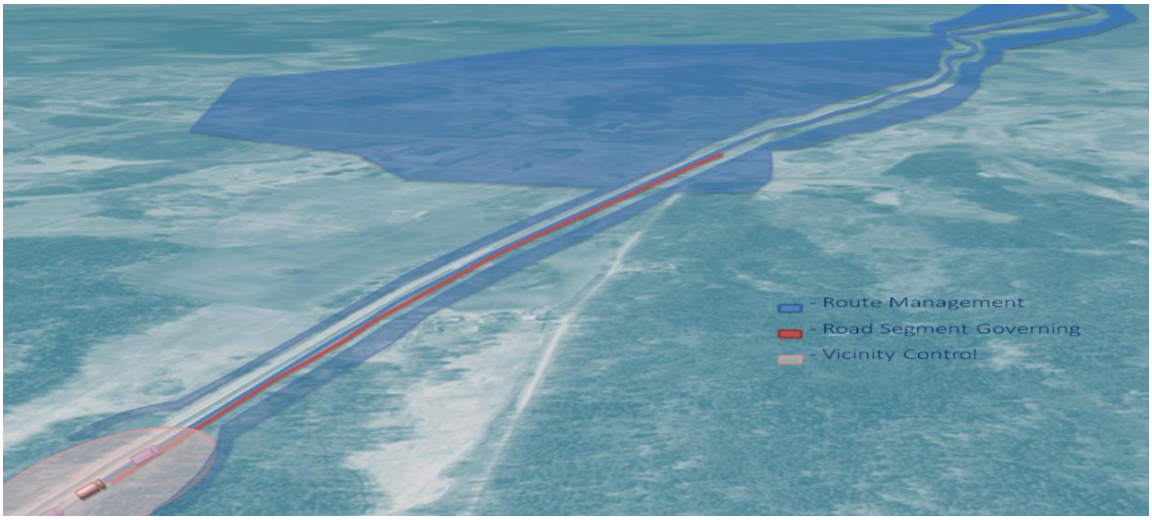


Figure 2-2 Spatial horizons: the Route Management is concerned with the blue line, the Road Segment Governor with the shorter red line and the Vicinity Control operates on the immediate surrounding of the truck.

These two layers are enough to fulfill the user requirements of COMPANION. A top layer optimizing the transport route for a COMPANION participant by taking traffic, speed and platooning possibilities into account, and a bottom layer executing the plan. However, once the vehicle starts its journey, the predicted future will be inaccurate and continuous recalculations are needed. To handle capacity issues, as well as keeping optimization problems comprehensive, the scope of the top layer is reduced into sub-problems, which translate into a new middle layer, the road segment-governing layer. This layer governs decisions within a spatial horizon of a few kilometers in front, therefore it assists the top layer by always optimizing the near future, i.e. finding sub-optimal solutions for events happening soon. In short, all three layers can be seen as sub-problems with different granularity.

Granularity

Every layer tries to find an efficient speed for a vehicle, or a platoon, for each layers spatial context. Essentially the output from each layer is a speed, and the granularity is the time between the speed changes. The top route management layer has a coarse granularity, while the bottom vicinity control has a fine granularity. The range differs from several minutes to tenth of seconds. Granularity and spatial horizons for the three layers are depicted in Figure 2-3.

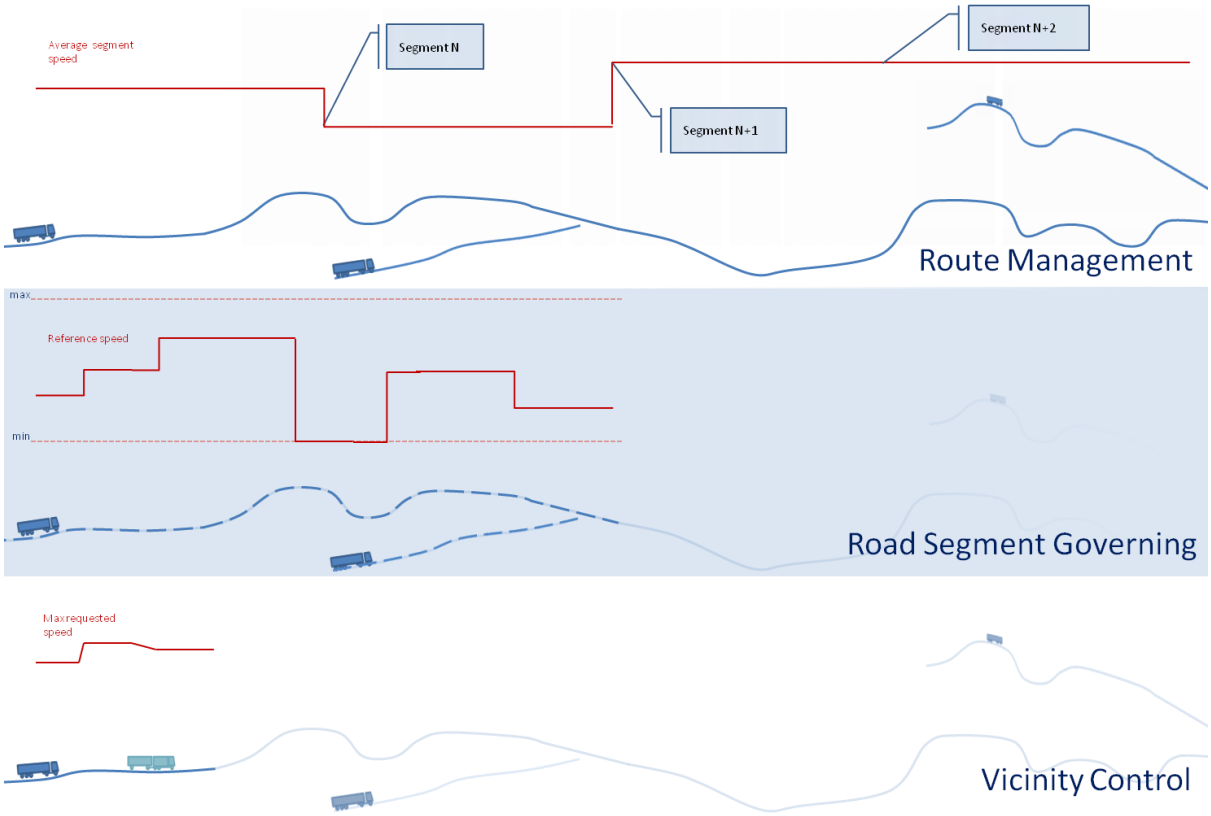


Figure 2-3 Granularity of speed.

Information Flow through the Layers

It's now well understood that each layer's desired speed travels downwards through the layers, from the top to the bottom. Naturally, information flows upwards as well.

Part of the concept of dividing the optimization into sub-problems is to avoid constant recalculation at a large scope. Therefore the most basic information travelling upwards is each layer's status, indicating if a vehicle does follow the desired speed or not. This is depicted in Figure 2-4.

Every layer monitors its progress and eventually flags for a new desired speed from the layer above, with the exception from the top layer. This is a pulling behavior.

From the top layer there is a push instead. This layer will monitor its forecasts and push a new desired plan whenever needed.

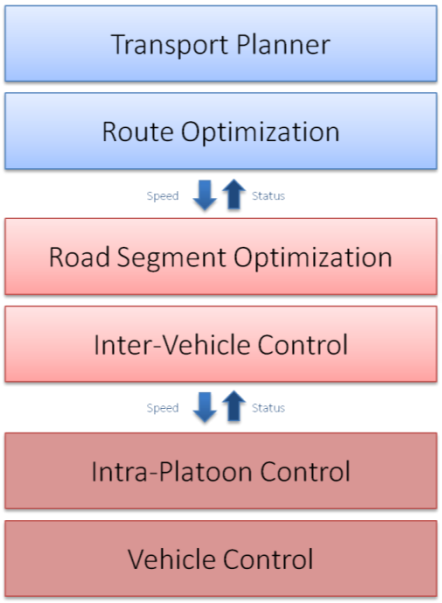


Figure 2-4 Information flow.

3. Route Management Layer

The Route Management layer comprises the strategic planning components and is concerned with the strategic task of route management in the COMPANION-System. This section describes the overall component architecture of the system.

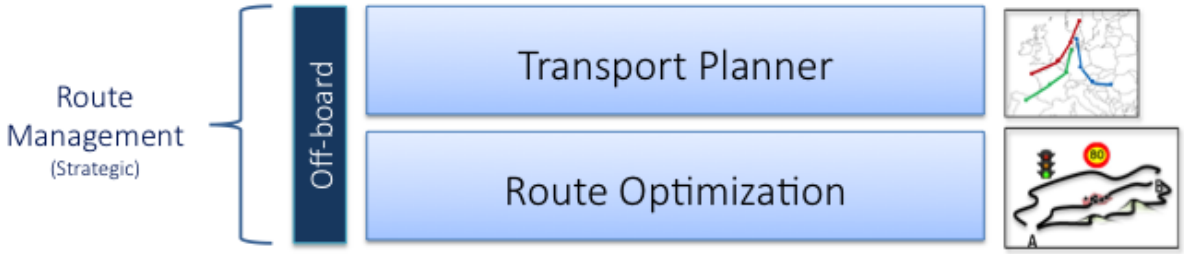


Figure 3-1 Route Management layer.

The Route Management layer, as depicted in Figure 3-1, is divided in to sublevels with different functional objectives. The first sublevel is the Transport Planner. Its creates a plan for what goes where, and with what and whom. For example: goods A goes to destination B with vehicle C and driver D etc. In COMPANION this input is given by the dispatcher. Hence, there will not be any resource optimization at this level in the COMPANION system.

The next sublevel is the Route Optimization. It is mainly concerned with the tasks of route calculation and energy optimization of these routes, meaning finding partners to platoon with and calculating an appropriate speed profile for each truck in order to merge with those partners.

Furthermore, the Route Management takes several risk factors in account like weather information and current and predicted traffic. If it is detected, that the current route of a vehicle or platoon is obstructed by such an event, like bad weather or a traffic jam, this event triggers a recalculation of the route, which is sent to the vehicle.

Components

Figure 3-2 shows the components of the route management layers. In this section the main components will be described briefly. For more a more detailed explanation of these components, the reader is referred to the according deliverables.

HMI Web Interface

The Off-board HMI is a human-machine interface which provides input and output functionality for the dispatcher. Dispatchers can create and manage transportation assignments. The calculated route, truck status, tracking data and the benefits of the platooning is displayed. The interface is accessible via a web-browser.

Deliverable D5.3 [6] describes the off-board HMI in detail.

Monitoring and Optimization Engine

The optimization engine calculates optimal and feasible velocity profiles and merging/splitting points for platoons, given a set of individual routes, velocity constraints, timing constraints and fuel consumption models.

Deliverable D5.2 [7] describes the Monitoring and Optimization Engine in detail.

Route Calculation Engine

The RCE calculates a single route for a single vehicle at a time. It takes into account the traffic situation (live and forecast), vehicle parameters (weight) and road attributes (slope, truck attributes).

The route result contains velocity windows (min, max) which the Optimization Engine uses to optimize the overall system platooning potential.

The RCE provides a Traffic Monitor component which observes changes in the traffic situation for all routes used by the Monitoring Engine.

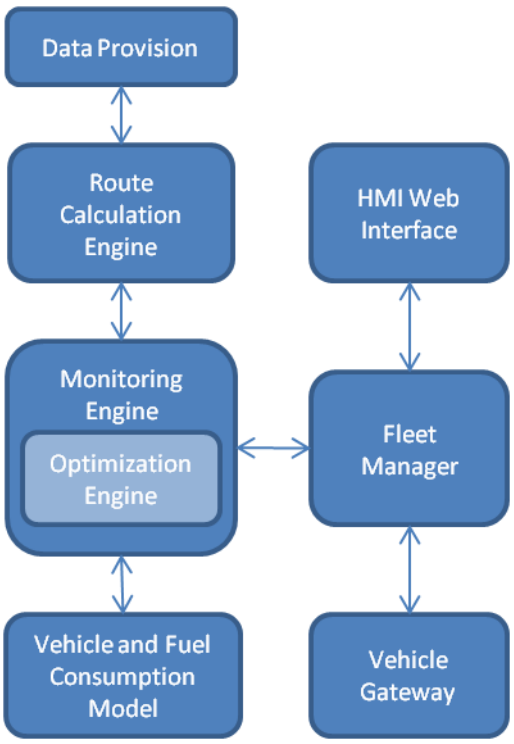


Figure 3-2 High level overall architecture of the Route Management layer.

Deliverable D5.1 [8] describes the architecture of the Route Calculation Engine in detail.

Data Provision

A machine learning system uses historical weather and traffic data to determine how weather affects travel times on road segments. Given a road segment and a numerical weather prediction the system can return an estimate for the delay as well as a confidence level for this estimate.

The system will be setup using lookup tables containing the delay and confidence level for each road segment given the weather. When no prediction is available for a road segment a zero delay will be returned to ensure the system will work

Deliverable D4.2 [9] describes the data provision of risk factors in detail.

Vehicle and Fuel Consumption Model

The Vehicle and Fuel Consumption Model service is used by the Monitoring Engine. It calculates possible speed sets for road segments depending on vehicle dynamic characteristics. Furthermore, the model calculates the fuel consumption for these possible speed sets.

Deliverable D4.3 [10] describes the Fuel Consumption Model in detail.

Fleet Manager

Having coarse-grained Transport Plans from Monitoring Engine and route, weather and traffic information from the Route Calculation Engine and Data Provision, the Fleet Manager generates Assignment Plans for each vehicle. Furthermore, the Fleet Manager gathers information about fleet

state and sends it to Monitoring Engine and HMI Web Interface.

Vehicle Gateway

The Vehicle Gateway handles the communication between the off-board components and the on-board components for each vehicle/driver connected to the COMPANION system.

This service makes sure all data is buffered (queued) if a vehicle currently is offline.

4. Road Segment Governing Layer

The Road Segment Governing layer acts in-between the top and bottom layer. It handles tactical short term goals for the vehicles in the COMPANION system. These goals are always tangible, happening within a near future, i.e. soon. The components in this layer are partitioned in both the off-board and the on-board systems. Basically, the output from this layer is a finer grained speed profile as mentioned earlier in chapter 2.

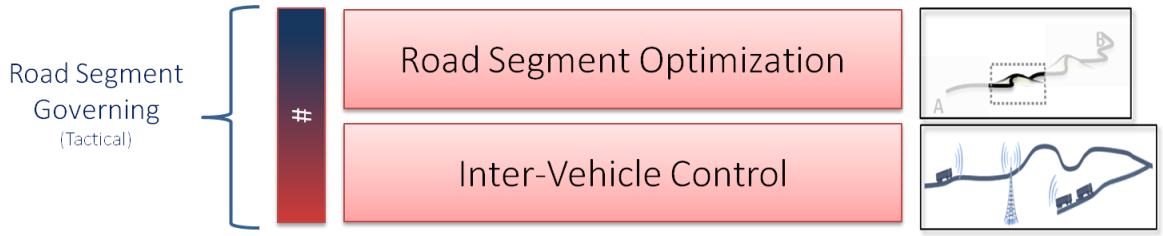


Figure 4-1 Road Segment Governing layer.

The Road Segment Governing layer consists of two sublevels as depicted in Figure 4-1. The Road Segment Optimization creates the speed profile for the segment based on the constraints given by the Assignment Plan, traffic, speed limits, topology and so on. The speed profile will be updated as the vehicle travels along the segment ensuring that the time constraint is met.

The bottom sublevel, Inter-Vehicle Control, synchronizes actions for the vehicle as it approaches a point where it has a cooperative task. For example a speed change just before a platoon formation in order to not miss each other.

Components

Here the main components of the Road Segment Governing layer are briefly described. More details can be found in deliverable D6.2 [11].

The Road Segment Governing layer shares a set of components with the bottom layer, Vicinity Control. These cross-layer components are called the backbone, as seen in Figure 4-2, and is further explained in chapter 0.

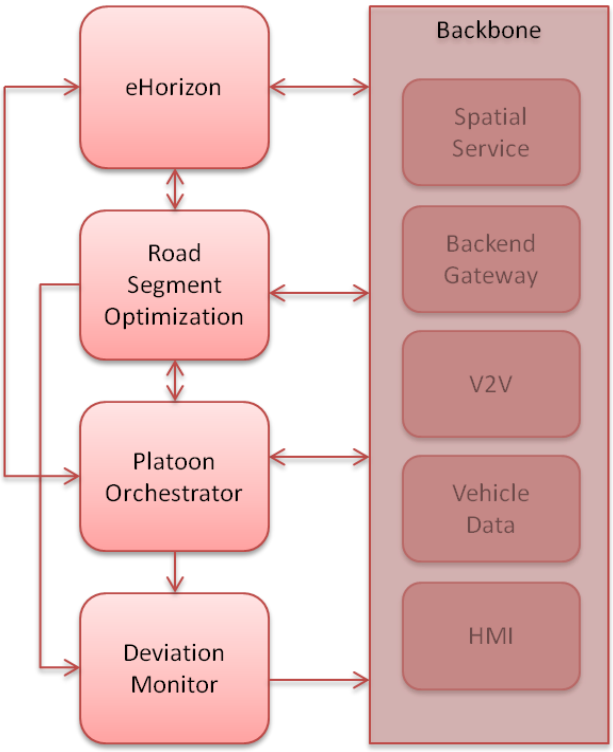


Figure 4-2 The abstract components of the Road Segment Governing layer

eHorizon

The eHorizon provides spatial data and tasks for the vehicles near future, e.g. road incline, speed limit, congestions, merge, split etc. The eHorizon collects data from the sources it needs to create the requested horizon.

Road Segment Optimizer

The Road Segment Optimizer will find the

most efficient speed profile for a single vehicle, or a vehicle part of a platoon, for the road segments ahead. It will try to meet the time constraint from the Assignment Plan. If it can't meet the constraint by changing the speed profile for the current and upcoming segment ahead it reports a deviation.

The output passes through the Platoon Orchestrator which has the final saying of the speed profile in this layer. The requested nominal speed will then go to the Speed Manager in the Vicinity Control layer.

Deviation Monitor

This component collects all deviation created by the concerning components of this layer.

Platoon Orchestrator

The Platoon Orchestrator monitors all COMPANION partners that are supposed to platoon during their route and makes sure that their speed is adjusted some kilometers before actual merging. It reports a deviation if it can't make it (for example, due to a sudden traffic jam)

When the vehicle is close enough for an actual merging the Scenario Orchestrator in the Vicinity Control layer takes over.

5. Vicinity Control Layer

The Vicinity Control layer handles the actual control of the vehicle. All components are located on-board the vehicle and their main task is to execute the speed profile given by the higher layers in real-time. This is done using the vehicle's engine, brakes and auxiliary brakes via the engine management system and the brake management system.

Figure 5-1 Vicinity Control layer.

A truck utilizing the COMPANION system can run in three main states; as a single vehicle, as a platoon leader or as a platoon member. The control strategies will vary depending on the state. In Figure 5-1 this is depicted with the two sublevels. Intra-Platoon Control handles the platooning states and the Vehicle Control the non-cooperative single vehicle state.

This is the bottom layer with real-time controlling. Regarding the communication, there is a major difference compared with the top layer. In the top layer, communication is mostly event-driven, while the bottom layer is periodic. In Figure 5-2 this can be seen as the components now work more independently communicating with something that seems to be a common bus, i.e. the backbone. However, the backbone is not a single bus. It is an abstraction of different communication technologies and information sources. The backbone is described in chapter 0.

Components

The main components of the Vicinity Control layer are described in this section. More technical details can be found in deliverable D6.2 [11].

Speed Manager

The Speed Manager controls the vehicle speed so that different requests are fulfilled according to the situation. For example the Speed Manager controls the distance to the vehicle in front in a platoon by adjusting the nominal speed request to the longitudinal actuator, i.e. the engine.

Emergency Braking

Emergency braking capability is crucial in short-gap platooning, but in other situations as well. The Emergency Braking component relies on sensor data from different sources, e.g. radar, camera and V2V. The quality and availability of this data depends on the vehicle specification and the current scenario.

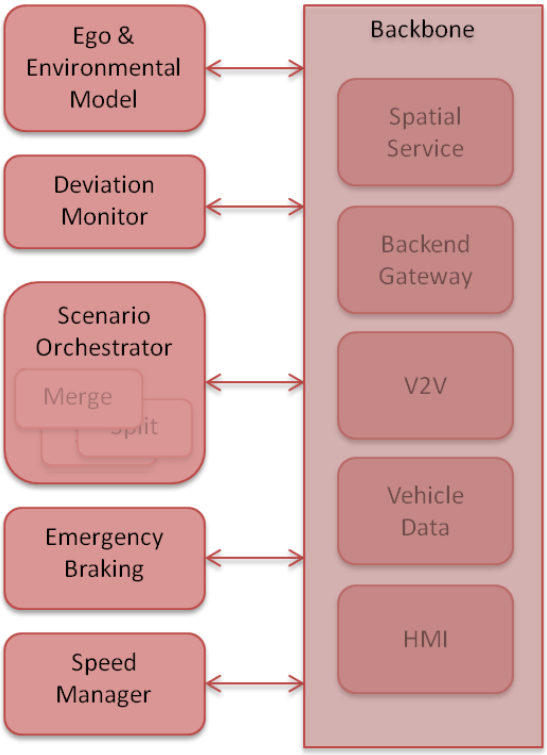


Figure 5-2 The abstract components of the Vicinity Control layer.

Consequently, the Emergency Brake will have different performance depending on the circumstances. For platooning, this directly affects the shortest possible safe distance between the vehicles.

Ego & Environment Model

The Environment Model describes the sensor-fused representation of the surroundings, e.g. a car with speed S at position X, Y.

The Ego Model describes the position of the ego vehicle, i.e. the vehicles own position relative to the surroundings.

Scenario Orchestrator

The Scenario Orchestrator orchestrates all necessary states when a sequential scenario takes place, like merge and split of platoons.

Deviation Monitor

This component collects all deviation created by concerned components of this layer.

6. Backbone – Cross-Layer Support

The backbone is a set of services that are utilized by a number of components in the Road Segment Governing and Vicinity Control layers.

The services in the backbone are of two major types:

- Communicative: services enabling communication between different layers, component, actors and functions in the system, e.g. HMI, V2V, buses and gateways.
- Informative: services where desired information can be pulled from.

These services can be seen as cross-layer supporting components. The most important ones, i.e. those needed to make the COMPANION system work as a whole, are described here.

Components

Spatial Service: This service holds all relevant information in a spatial domain. For example; road slope, traffic jams and platooning merging points. The data can either be static or highly dynamic.

Backend Gateway: This is the communication link between the on- and off-board systems. A stable connection to the off-board system is of higher importance than latency and transmission speed. In COMPANION 3G/4G mobile communication is used.

Advance Message Queuing Protocol, AMQP, is used to securely communicate with Vehicle Gateway of the off-board system. Hypertext Transfer Protocol Secure, HTTPS, file transfer is also used to download larger data sets.

V2V: The vehicle to vehicle communication node is used for exchanging information about the state of the ego vehicle with the other vehicles in the platoon. Examples of information being exchanges are position, speed, intended speed of the vehicle, as well as information about if and how hard a vehicle is braking. This information is important for the vehicle control. Hence, low latency is of high importance.

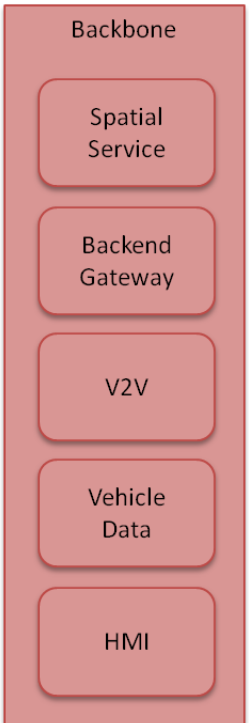


Figure 6-1 Backbone services.

ETSI ITS-G5 standard is used for the V2V communication. However, since the services of the ITS-G5 is not yet adapted for a platooning application the current predefined message services are not used, e.g. the Cooperative Awareness Message and the Decentralized Environmental Notification Message. Instead the Basic Transport Protocol of ITS G5 will be used, carrying proprietary data at a fixed transmission frequency of 20 Hz.

Further details can be found in deliverable D6.1 [12].

HMI: The Human Machine Interface has multi modal inputs and outputs, e.g. instrument cluster, screens, sounds and buttons. It enables interaction with the driver of a COMPANION vehicle.

A more detailed description can be found in deliverable D6.4 [13].

Vehicle Data: There is a lot of vehicle and driver related data needed by all layers, some of which are created by the components described in this document. But there is also data created by components not included in this abstraction to keep it simple and only focus on the things essential

for COMPANION. The Vehicle Data is the abstract service for all ego related vehicle data needed by the COMPANION components.

It has:

- static data like; vehicle ID, driver IDs, driver preferences and vehicle properties,
- low frequency dynamic data like; vehicle weight, current gear and driver settings,
- high frequency dynamic data like; vehicle position and longitudinal accelerations.

7. References

[1] COMPANION, *Deliverable D2.1 Potential scenarios for new platooning concepts,* 2014.

[2] COMPANION, *Deliverable D2.3 Final User Requirements,* 2014.

[3] COMPANION, *Deliverable D2.4 Technical Requirements,* 2015.

[4] COMPANION, *Deliverable D2.5 Business Requirements and Business Opportunities,* 2015.

[5] COMPANION, *Deliverable D3.2 Information Model for Platooning Services,* 2015.

[6] COMPANION, *Deliverable D5.3 Back Office HMI Prototype,* 2016.

[7] COMPANION, *Deliverable D5.2 Report on coordinated fault-tolerant Scheduling of Platoons,* 2016.

[8] COMPANION, *Deliverable D5.1 Report on fuel-efficient Route Calculation,* 2015.

[9] COMPANION, *Deliverable D4.2 Risk factors for route planning,* 2015.

[10] COMPANION, *Deliverable D4.3 Vehicle Models for Fuel Consumption,* 2015.

[11] COMPANION, *Deliverable D6.2 Semi-automated longitudinal vehicle,* 2016.

[12] COMPANION, *Deliverable D6.1 Integrated V2X communication,* 2016.

[13] COMPANION, *Deliverable D6.4 On-board HMI Prototype & Report on on-board HMI,* 2016.