Design Choices for Self-Organizing Railway Operations

Grant Agreement N°:	875022
Project Acronym:	**SORTEDMOBILITY**
Project Title:	Self-Organized Rail Traffic for the Evolution of Decentralized MOBILITY
Funding scheme:	Horizon 2020 ERA-NET Cofund
Project start:	1 June 2021
Project duration:	3 Years
Work package no.:	WP3
Deliverable no.:	D3.1
Status/date of document:	Final, 09/04/2022
Due date of document:	28/03/2022
Lead partner for this document:	ISTC-CNR
Project website:	www.sortedmobility.eu

Dissemination Level		
PU	Public	**X**
RE	Restricted to a group specified by the consortium and funding agencies	
CO	Confidential, only for members of the consortium and funding agencies	

Revision control / involved partners

Following table gives an overview on elaboration and processed changes of the document:

Revision	Date	Name / Company short name	Changes
1	01/02/2022	Vito Trianni, ISTC-CNR	Draft created
2	15/02/2022	Valentina Tibaldo, ISTC-CNR	Draft finalised
3	20/02/2022	Paola Pellegrini, univEiffel	Revision of figures
4	28/02/2022	Valentina Tibaldo, ISTC-CNR	New flow diagrams
5	15/03/2022	Vito Trianni, ISTC-CNR	Document finalised

Following project partners have been involved in the elaboration of this document:

Partner No.	Company short name	Involved experts
1	univEiffel	Paola Pellegrini, Nathalie Botticchio
2	SNCF	
3	DTU	
4	BDK	
5	ISTC-CNR	Valentina Tibaldo, Vito Trianni
6	RFI	
7	TUDelft	

Executive Summary

The objective of D3.1 is to is to describe the system architecture designed for self-organised traffic management. After a literature review, we have started from the output of the European project ONTIME, which provided basic concepts useful for both centralised and decentralised traffic management.

After some preliminary considerations, the detail of both a centralised and a decentralised system architecture are provided in Chapter 2. A few components are shared between the two architectures, namely all process that are performed centrally by a traffic control centre. However, in the decentralised architecture, trains themselves proceed to the definition of the so-called Real-Time Traffic Plan (RTTP), by first producing hypotheses computed on a train-centric neighbourhood, and then reaching consensus on the best hypothesis with neighbouring trains. Local RTTPs are then communicated to the traffic control centre, which will first aggregate them via a global merge process, and then will take action for the final management of the infrastructure and the involved trains.

Table of contents

Table of figures

1 INTRODUCTION

This deliverable reports on the analysis performed about the requirements for self-organising traffic management systems, where intelligent trains interact to determine their route and schedule in a completely decentralised way. On top of the requirements, a system architecture has been designed identifying the relevant components that need to be deployed for a realistic implementation, which takes into account the constraints and opportunity offered by the current system architecture.

2 DESIGN CHOICES

2.1 Preliminary considerations

Abstracting from the details of the rail traffic management system, the problem of coordinating the routes and schedules of autonomous vehicles can be thought of as the coordination of movements of agents on a graph. Here, the graph represents the rail network where nodes correspond to block sections (with the constraints that each block section can be occupied by only an agent at the time) and non-directional edges correspond to connections between different block sections. Agents must move on the graph from a starting node (i.e., the departing station) to a final node (i.e., the destination station). Movements on the network are constrained by the need to never overlap with other agents. For the sake of simplicity, we do not consider the additional complexity brought by the possible incompatibility of different block sections. The discussions that follow can indeed extend to the more general case.

While traversing the graph, every single decision step may involve one or more agents, who must decide their move while avoiding overlap. Such group decisions can be formalised as an *anti-coordination game*, whereby agents need not choose the same alternative. Imagine there are two trains, *A* and *B*, that want to occupy the same block section at the same time. In a decentralised system, the following alternatives may arise:

1. *A* decides to move occupying the block section, while *B* stays
2. *B* decides to move occupying the block section, while *A* stays
3. Both *A* and *B* decide to move
4. Both *A* and *B* decide to stay

Cases 1 and 2 above represent desired choices, which may or may not be optimal in terms of global delay to traffic (or other KPIs), but would both be practicable. Option 3 instead represents an inoperable choice, which is forbidden by safety systems. It must hence be managed specifically, for example by a central system that overrides the individual decisions, e.g., by giving priority to one of the trains according to (suboptimal) heuristics like *first-come-first-served*. Option 4 causes delays on both trains, which would propagate to the entire network. Hence, this should be avoided by letting trains determine not only their movements but also the movements of others.

From these preliminary consideration, it emerges that (i) the proposed system architecture should strive to reach consensus on a feasible solution; (ii) solutions that are not maximising the infrastructure capacity should not be proposed; (iii) there should be a well defined procedure to resolve conflicts in case consensus is not achieved.

To define some general guidelines for the design of the system architecture for self-organising traffic management operations, we first analyse a possible deployment of a more classic centralised system. In particular, we focus on the proposal made in the ONTIME project (Optimal Networks for Train Integration Management across Europe (FP7-SCP0-GA-2011-265647). Then, a decentralised solution is proposed that can be benchmarked against the centralised one.

2.2 Centralised system architecture

A train infrastructure is an open system, in which external sources can interfere withtraffic. Examples are a dead tree on the railways, or a passenger blocking a closing train door at a station. These events can be categorised in either *perturbations* and *disruptions*, depending on the magnitude of the event and on the subjects involved in the process of understanding how to unravel traffic decisions. In perturbations, traffic plans do not need to be greatly modified, while in disruptions this is necessary. For example, passenger blocking a door may imply a few minutes of delay, and this is to be managed by dispatchers to avoid propagation. They will probably reorder some trains in some locations or change, for example, platform use at some stations. In a disruption, as a tree cutting a line, major decisions are required such as train cancelling and short-turning. These decisions need to be made after reaching an agreement with all actors involved, including at least the railway undertakings. In this project, we will focus on perturbations that can be managed autonomously by dispatchers.

To deal with perturbations, the Traffic Management System exploits a group of (software) modules deputed to detect and forecast traffic information and occurring perturbations, and manage traffic flows, for example by computing safe routing on the system infrastructure such that no two trains occupy the same block section at any time. These modules work in a pipeline in which the output of each module is the input of the next one (see Figure 1). The pipeline itself creates a loop for which the state of the network, resulting from the execution of the plan computed in the previous round, is fed as input to the next iteration of the process. This is typically referred to as a *closed-loop framework*.

In this framework, one can distinguish actors (yellow rectangles in Figure 1) and software modules (rounded rectangles) that are involved in the process.

The system is centralised because there is a unique well-identified actor deputed to make decisions of train retiming or reordering. Currently, this actor is the dispatcher operating the **Traffic Control Center**, that actuates the Traffic Plan through the Automatic Route Setting (ARS) that acts on the **Infrastructure**, and by informing the **Trains** on the updated modules.

The centralised software pipeline foresees the following steps:

- **Traffic state monitor (TSM)**: this module is responsible for the collection of the data on the infrastructures and trains such as train speed and positions and interlocking system state. This information is collected, processed and integrated with the daily timetable, to produce the *current traffic state.*
- **Traffic State Prediction (TSP)**: this module is in charge of producing a description of the *future traffic state* on the basis of the current available knowledge.
- **Demand prediction**: this module exploits knowledge of the current demand as well as historical data to make a *demand prediction* for different origin-destination pairs.

Conflict Detection and Conflict Resolution: these two modules detect possible conflicts on the network and solve them through appropriate rescheduling and rerouting actions on the traffic, that allow to avoid conflicts or to minimise their consequences on the network performances. The rescheduled timetable is then translated into the *Real-Time Traffic Plan (RTTP)*, which describes how the traffic is planned to develop in the future.

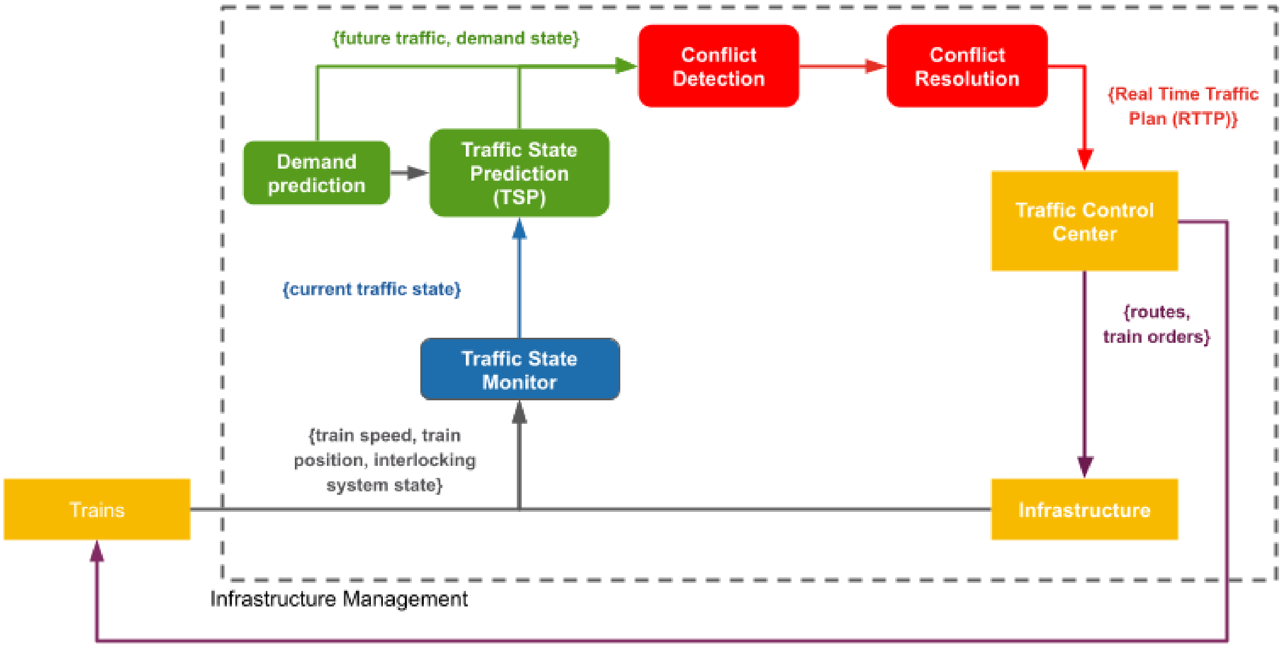

Figure 1: Pipeline of the centralised traffic management.

While the RTTP is sufficient for operating the infrastructure and the desired schedule from the Traffic Control Center, additional software modules can provide further details that may be useful for improving the traffic management operations. On the one hand, a train path envelope can be computed to determine the location-time boundaries under which the train can respect the RTTP indications. An additional traffic state prediction may be carried out to refine the RTTP providing additional precision to the traffic management operations, resulting in an enriched RTTP that is ultimately used to provide instruction to train and operate the infrastructure. This is particularly useful in implementations in which precise time information is necessary and is not delivered by a train path envelope computation. These last two steps are optional and are not represented here for the sake of simplicity.

2.3 Decentralised system architecture

In the decentralised system architecture, intelligent trains operate in a self-organised manner to guarantee high levels of service making the rail transport system more resilient and capable of self-adapting to an evolving environment with respect to the demand and in case of perturbations. Starting from the centralised system infrastructure, we designed the decentralised system ensuring that demand forecasts are integrated, and that the automatic decisions can be implemented—and possibly overridden—by the traffic control centre which ultimately operates the infrastructure and communicates with all trains.

The first choice for a self organising system is the definition of the element in which to distribute the computation. In a decentralised system, there is no single point where the decision is made: every agent in the system makes a decision for its own behaviour and the resulting system behaviour stands on the aggregation of the responses. For the proposed train traffic management architecture, the individual agents are the trains themselves and the behaviour of the whole transport system is obtained through a consensus process used for aggregating their responses. Hence, trains are active agents for the traffic management, as opposed to being passive in the centralised system.

The pipeline for decentralised traffic management is shown in Figure 2 (top panel). It shares several components with the centralised one, but differs in some crucial aspects. The pipeline is not so linear anymore, and information flows back and forth between trains and the other components of the traffic management systems (see also the sequence diagram in the bottom panel of Figure 2). Similarly to the centralised approach, the infrastructure management is operated centrally by the Traffic Control Centre. Through the sensing system spread on the infrastructure, information is gathered and elaborated from the TSM module to produce the current traffic state. The latter is fed to the TSP that produces the future traffic state, which is merged with the output of the Demand Prediction module. All these modules are identical as in the centralised case. In particular, the information of the future traffic state is computed centrally. In future implementations, this could be also decentralised and implemented on each single train, especially if trains can exploit information that is not available centrally. Indeed, given that trains are mostly interested in the local traffic, it could be reasonable to have predictions from simpler computations performed onboard the

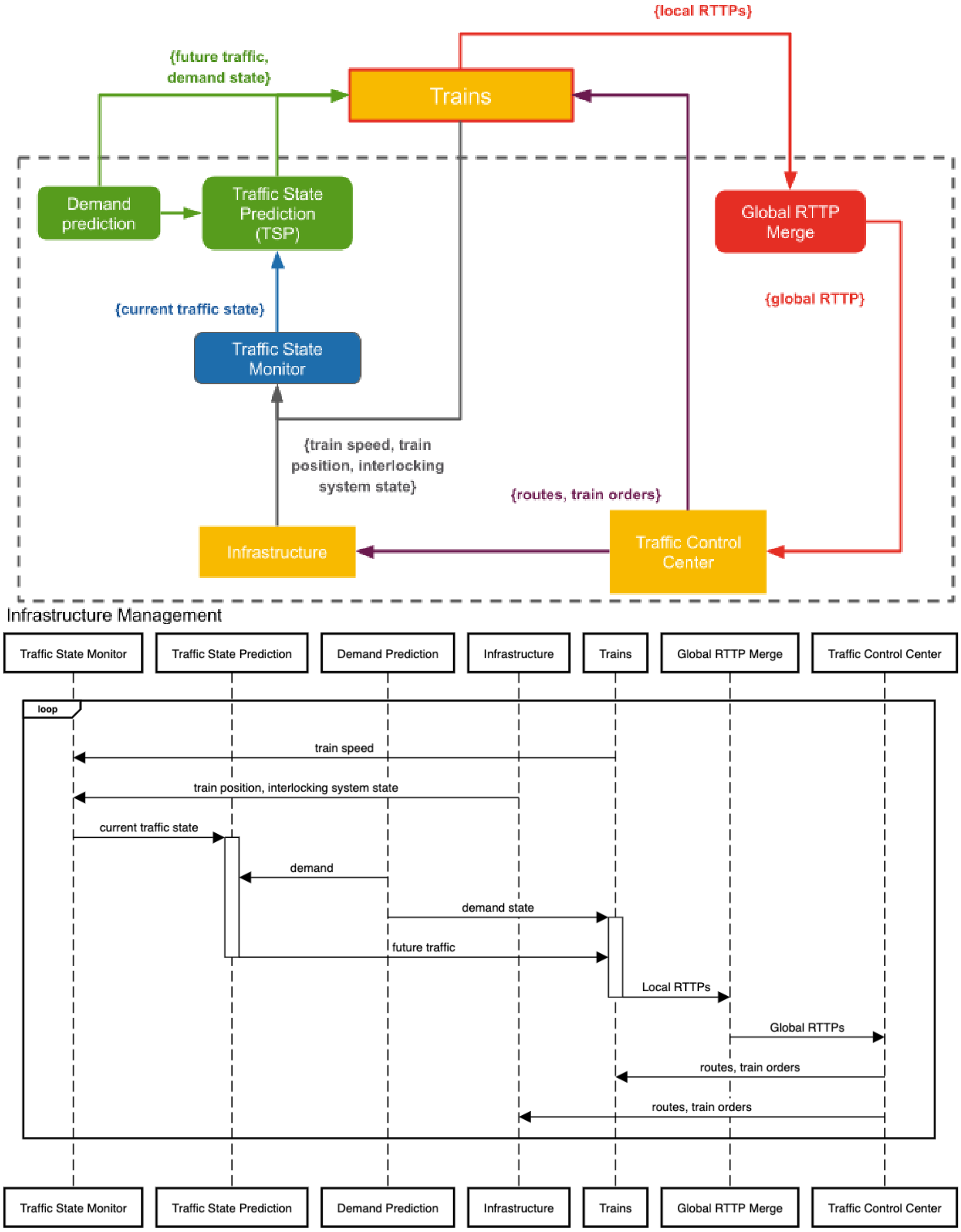

Figure 2: Top: Pipeline of decentralised traffic management. Bottom: sequence diagram that describe the data exchange among the different system modules

trains. However, as a first approximation, it makes sense to keep this component centralised.

In the proposed system, the centralised conflict detection and resolution modules are replaced by a decentralised decision-making approach: once the future traffic/demand state is delivered to each train, this is exploited to compute a train-centric plan to be executed in the future, in interaction with other trains on the network. Trains must be capable of trading on autonomous decisions, based on local data and priorities to produce a **local RTTP**, that is, a partial traffic plan that includes the train itself and some of its neighbours. The decentralised generation of local RTTPs results from a negotiation process in which several trains try to reach consensus on a feasible plan (see Figure 3).

This process involves the following modules:

- **Neighbourhood detection**: this module is responsible for the dynamic identification of a neighbourhood that is relevant for the current traffic conditions. The neighbourhood may include preceding and following trains selected on the basis of the influence that local decisions may have on close intersections, for example.
- **Hypotheses generation**: each train proceeds to the generation of hypotheses about possible plans (i.e., RTTPs), which are scored on individual and system-level metrics (e.g., the accumulated delay of the train and of a group of trains).
- **Consensus**: the consensus process is carried out among all the trains within a given neighbourhood, where the trains attempt to reach an agreement on a common RTTP. Note that the same train can take part to multiple neighbourhoods, so that a train must reach consensus within all neighbourhoods but on the same train-centric hypotheses. This entails a complex consensus project to be played on a heterogeneous network of interacting trains.

As a result of the consensus process (successful or not), each train outputs a local RTTP that must be communicated to the Traffic Control Centre to operate the infrastructure. However, before that, a consistency check is needed, and safety must be ensured on the occupation of the infrastructure, avoiding overlaps, namely the request for occupancy of the same track portion at the same time by two different trains. To this end, an additional, centralised module is necessary, responsible for the **Global RTTP Merge**: all the local, partial RTTPs produced by different trains are received and merged together to ensure *completeness* (all trains must be included in the planning) and *correctness* (there is no overlap). In case any correction is required (e.g., when no consensus is reached on local RTTPs), heuristics can be applied or human intervention can be called. The process ends with a new global RTTP that is exploited by the Traffic Control Center to operate the infrastructure and instruct all trains about the new schedule.

Note that the merge process—and specifically how many overlaps are found while merging different RTTPs from different trains—can give a measure on the quality of the consensus process: the more the consensus is effective, the less interventions are needed in order to achieve a global RTTPs. Ideally, a successful consensus will lead to a straightforward merge of local RTTPs into a global one.

3 CONCLUSIONS

In this document, we have outlined the system architecture for both a centralised and a decentralised system. These two alternatives are kept aligned as much as possible in several of their components, to allow straightforward implementation and reuse of software modules. Additionally, this approach supports systematic comparisons between the two approaches that can pinpoint to the benefits and costs related to choosing one approach or the other. Overall, the system development and evaluation is greatly simplified by this choice, providing useful answers to the main questions of the project: to what extent a self-organised approach can be beneficial for rail traffic management? The experimental activities that will follow this design phase will provide clear and quantitative responses.

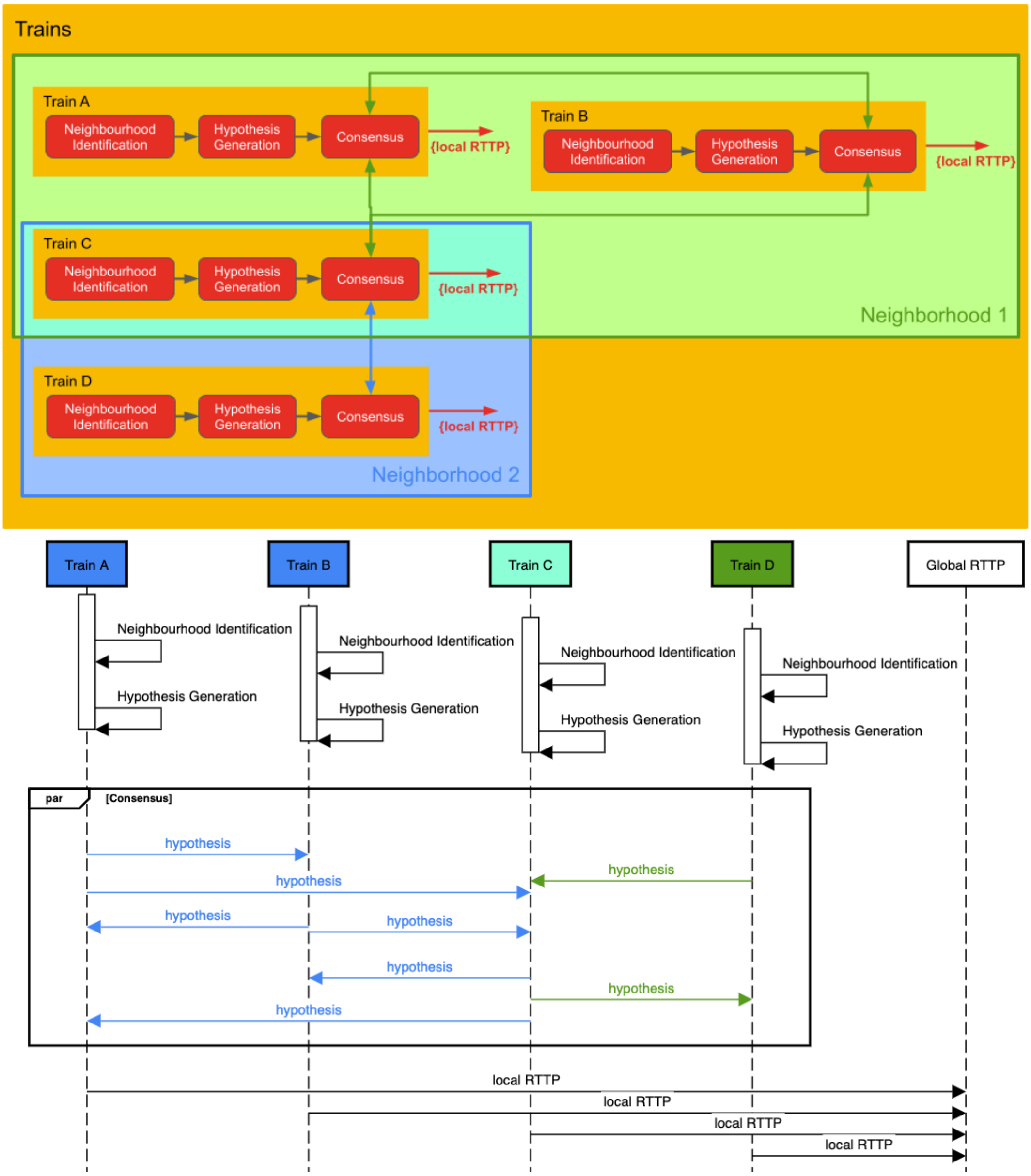

Figure 3: Detail of the consensus process. Top: Different colours define different neighbourhoods for different trains: blue for trains A,B,C and green for trains C and D. Note how train C is present in both of the neighbourhoods (depicted in cyan to identify the overlapping blue and green neighbourhoods). Bottom: sequence diagram for the consensus process in which the consensus part is carried out asynchronously and in parallel among trains in the same neighbourhood