SEVENTH FRAMEWORK PROGRAMME

TUDelft

ON-TIME National Workshop Sweden, 16 October 2014

Innovations in Timetable planning and Traffic control

ONTIME

[Optimal Networks for Train Integration Management across Europe]

Collaborative Project

7th Framework Programme

ON-TIME Real-Time Traffic Management of Minor Perturbations

Rob M.P. Goverde[1] & Thomas Albrecht

[1]Delft University of Technology, The Netherlands

r.m.p.goverde@tudelft.nl

What does WP4 do?

- Real-time perturbation management in case of small delays
- Control measures
  - Re-order trains
  - Re-route trains
  - Re-time trains
  - Cancel or add non-commercial stops (operational stops)
- *No interaction* with RU necessary
- Mainly *automatic* decisions

Questions from practice

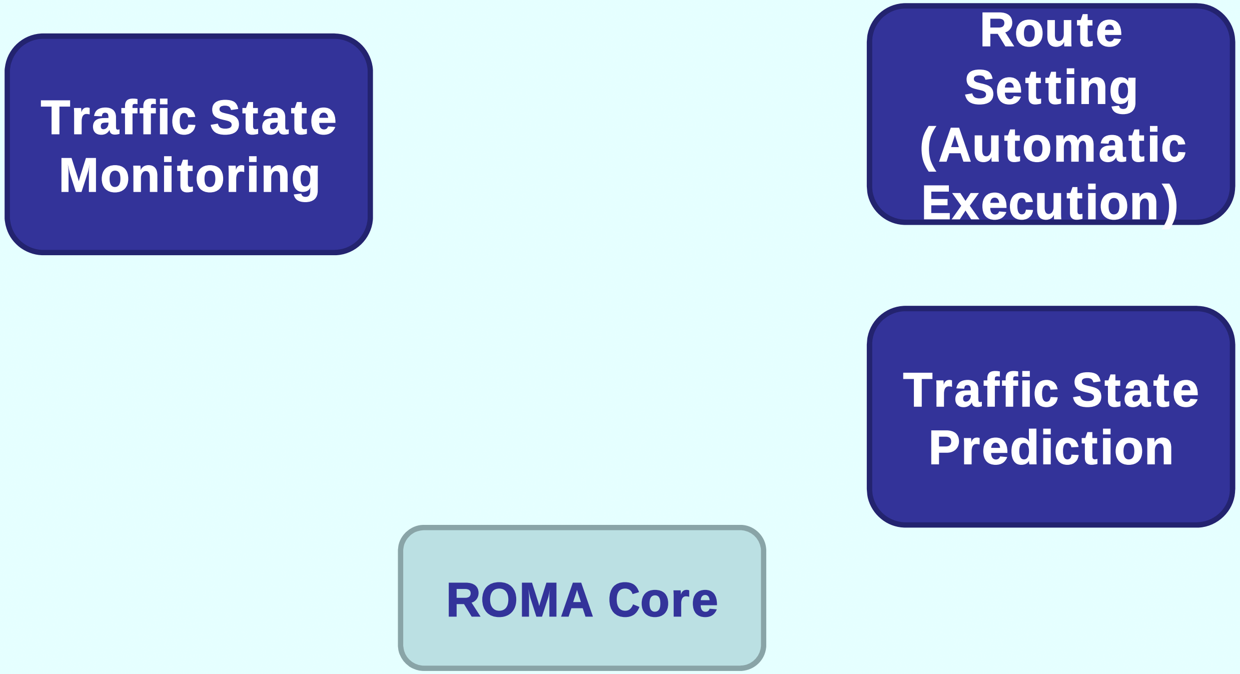
- These algorithms cover just particular aspects of optimisation.
- These algorithms can‘t work in real time.
- Are these algorithms really efficient in practice? (Are they able to provide additional capacity, less delays etc.?)
- These algorithms require data which is not available.
- What will our dispatchers say if they are confronted with these algorithms?

Questions from practice

- These algorithms cover just particular aspects of optimisation. → **Modular system design (1)**
- These algorithms can‘t work in real time.
  → **Real-time environment for data exchange (2)**
- Are these algorithms really efficient in practice? (Are they able to provide additional capacity, less delays etc.?)
  → **Evaluation in simulations (3)**
- These algorithms require data which is not available.
  → **Standardized data model (4)**
- What will our dispatchers say if they are confronted with these algorithms?
  → **Concept for human machine interaction (5)**

Modular design

- Different modules from different universities
- Different aspects of railway traffic control



Traffic State Monitoring

Route Setting (Automatic Execution)

Traffic State Prediction

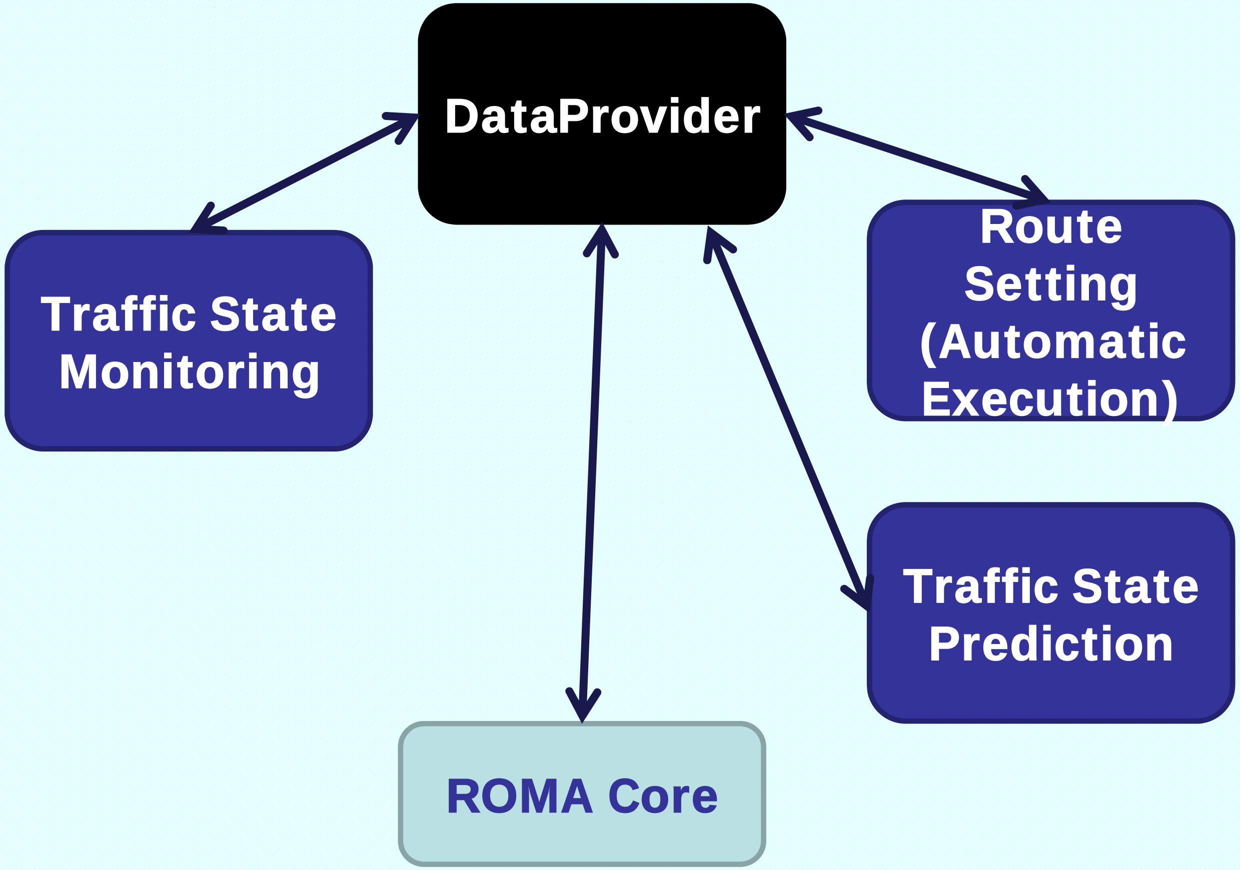
ROMA Core

Questions from practice

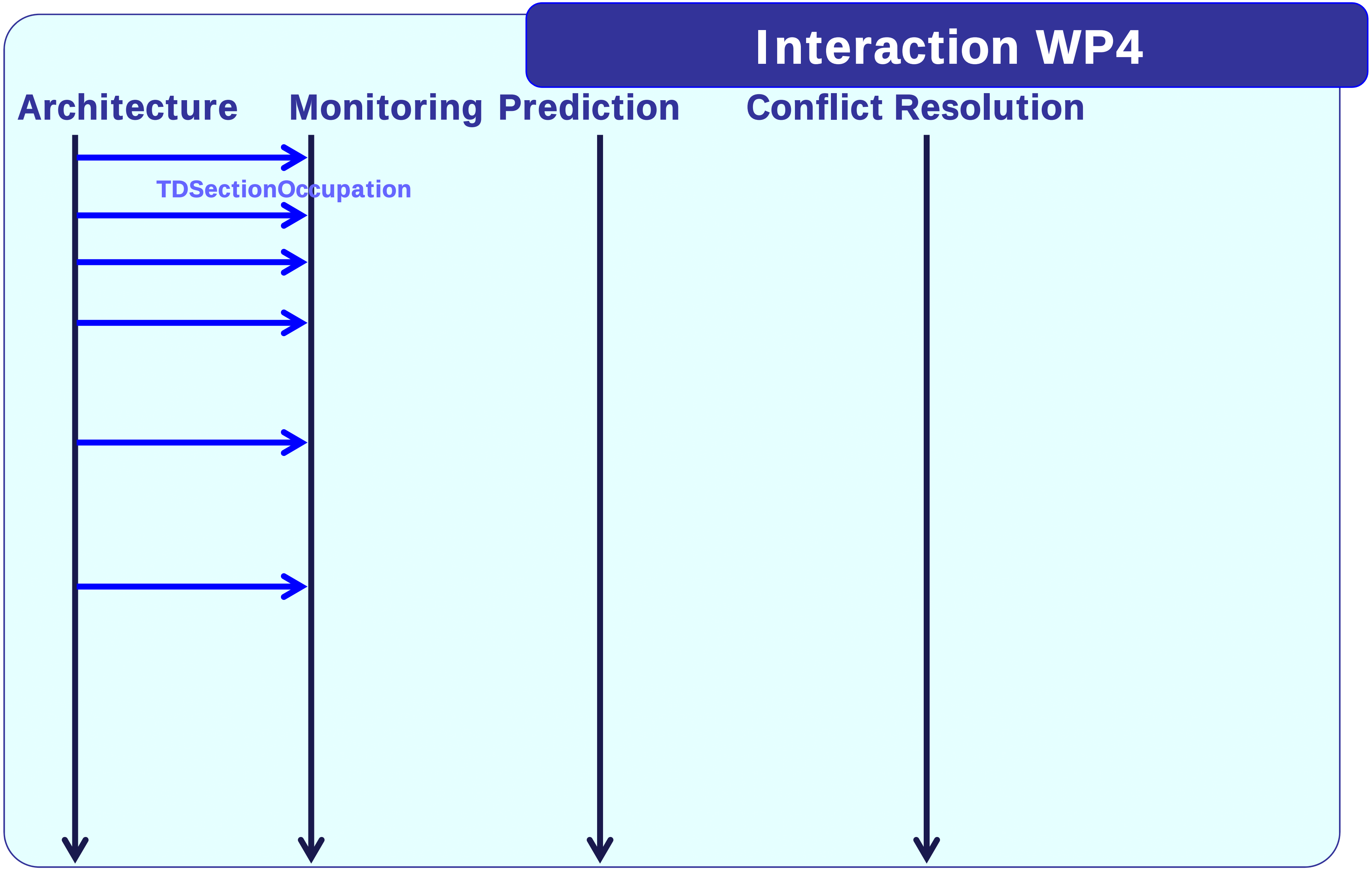
- These algorithms cover just particular aspects of optimisation. → **Modular system design (1)**
- These algorithms can't work in real time.
  → **Real-time environment for data exchange (2)**
- Are these algorithms really efficient in practice? (Are they able to provide additional capacity, less delays etc.?)
  → **Evaluation in simulations (3)**
- These algorithms require data which is not available.
  → **Standardized data model (4)**
- What will our dispatchers say if they are confronted with these algorithms?
  → **Concept for human machine interaction (5)**

Data exchange architecture

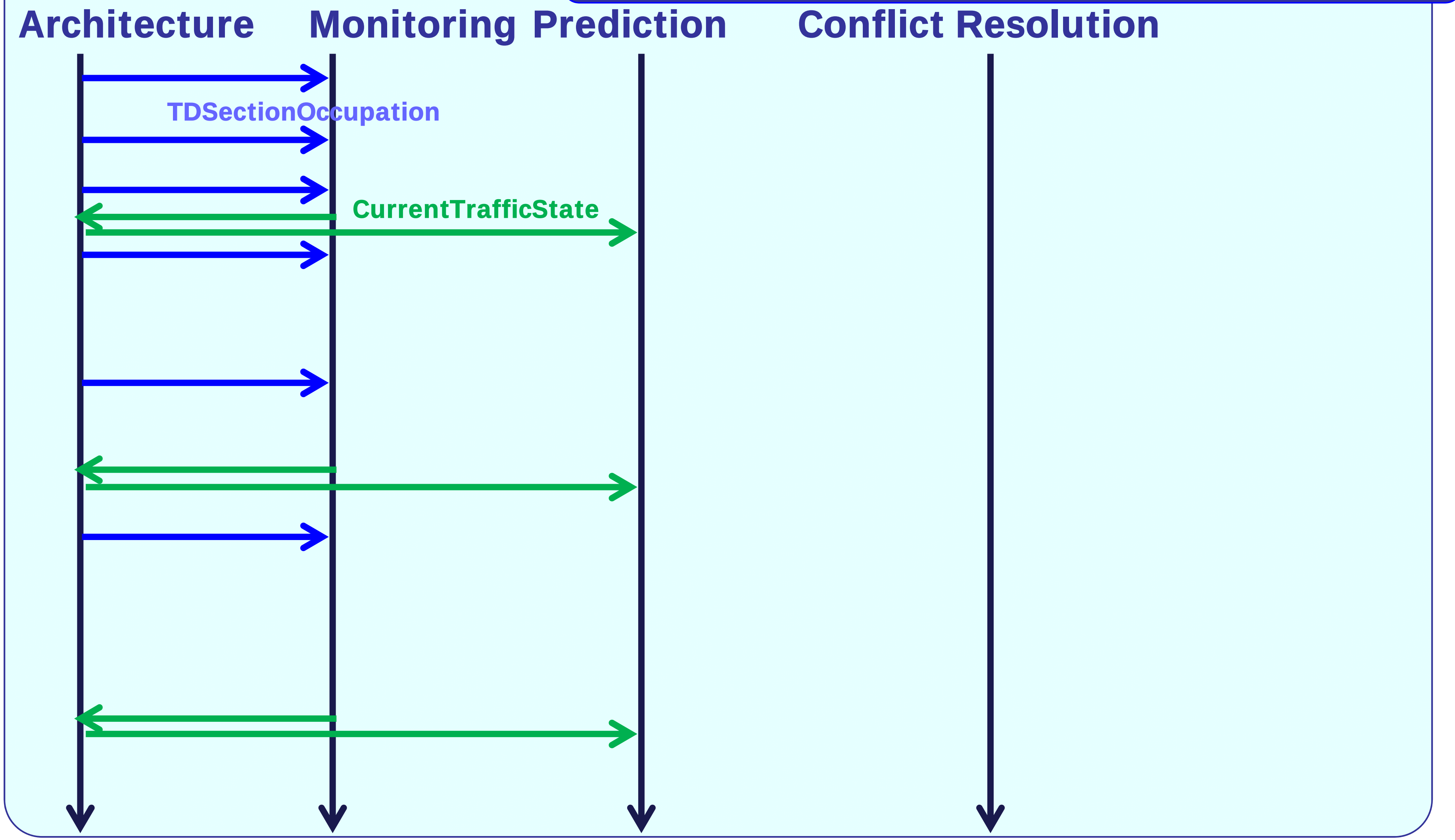
- Service-Oriented-Architecture
- Publish/ Subscribe Services



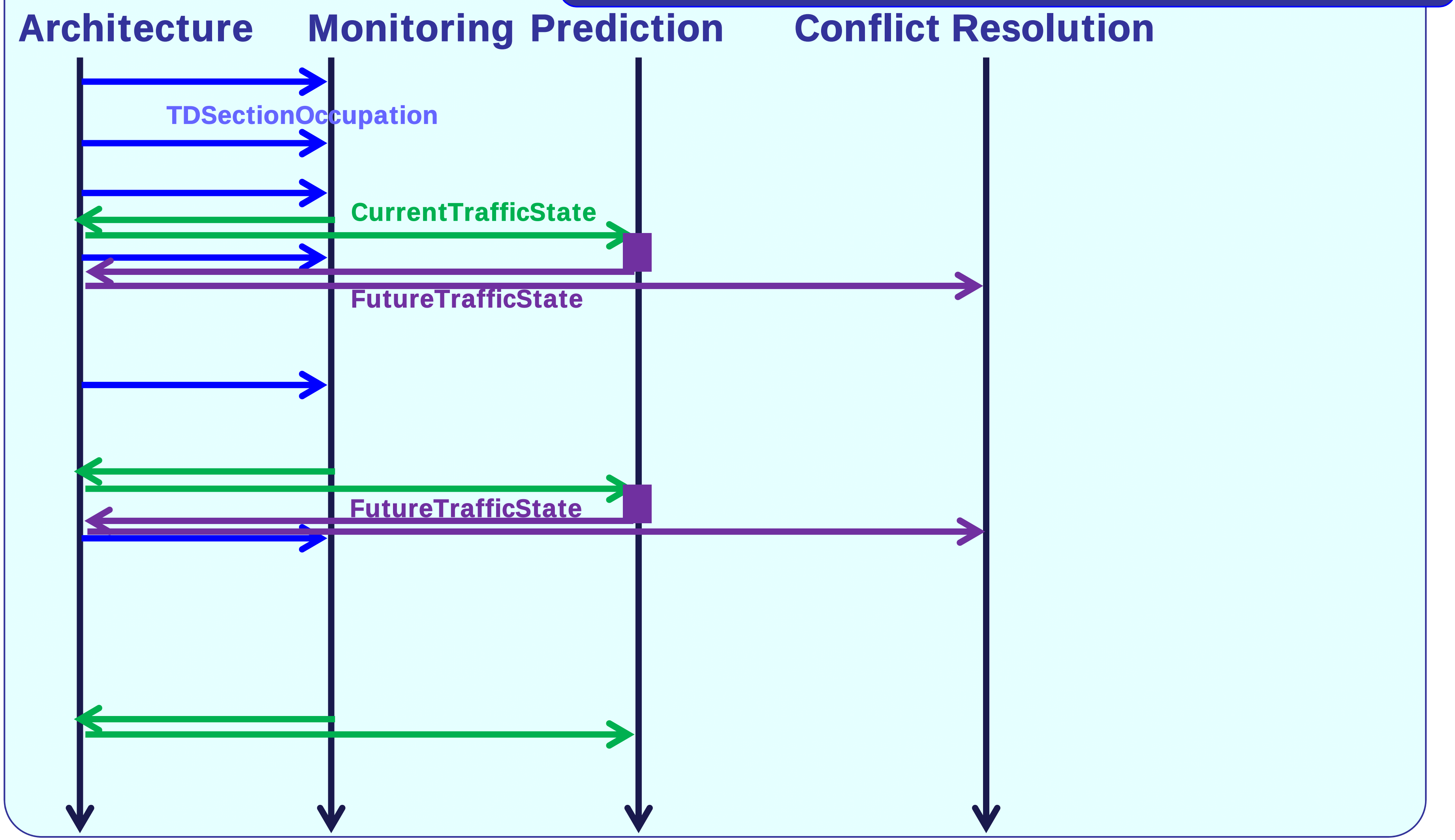
Interaction WP4



Interaction WP4



Interaction WP4



CDR

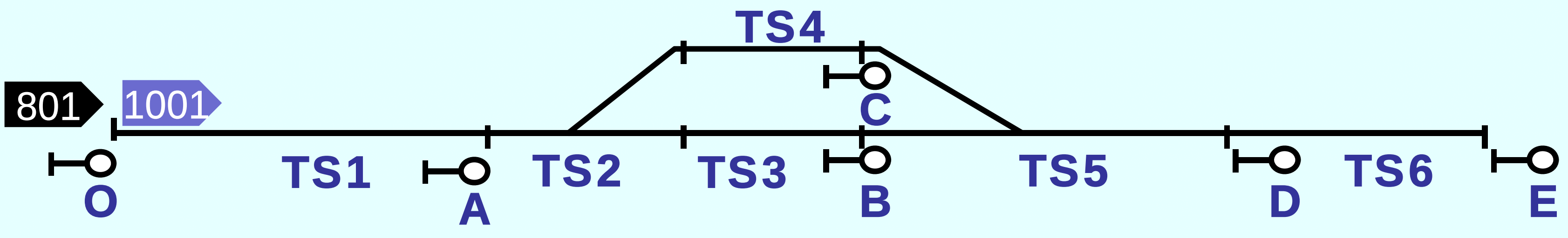
- Different models, classes of algorithms, and programming languages
- Implemented CDR approaches
  - **ROMA**: Alternative Graph Approach, B&B Solution (TU Delft)
  - **RECIFE**: MILP (IFSTTAR)
  - **DEJRM**: Evolutionary Algorithm (UoB)
- All algorithms extended and adapted to work with the control loop in real-time using the interfaces defined
- Result: **Real-Time Traffic Plan**

Real-time traffic plan

- Describes microscopically how the traffic shall be executed
- Routing, timing and stopping information

- Routing:
  - which routes will the trains take
  - in which order will trains pass over sections
- Timing:
  - when will a train occupy a certain section
- Stopping:
  - where and when will trains stop

Routing part



Train view

Train 1001	
Route	Section
O-A	TS1
A-C	TS2
	TS4
C-D	TS5
D-E	TS6

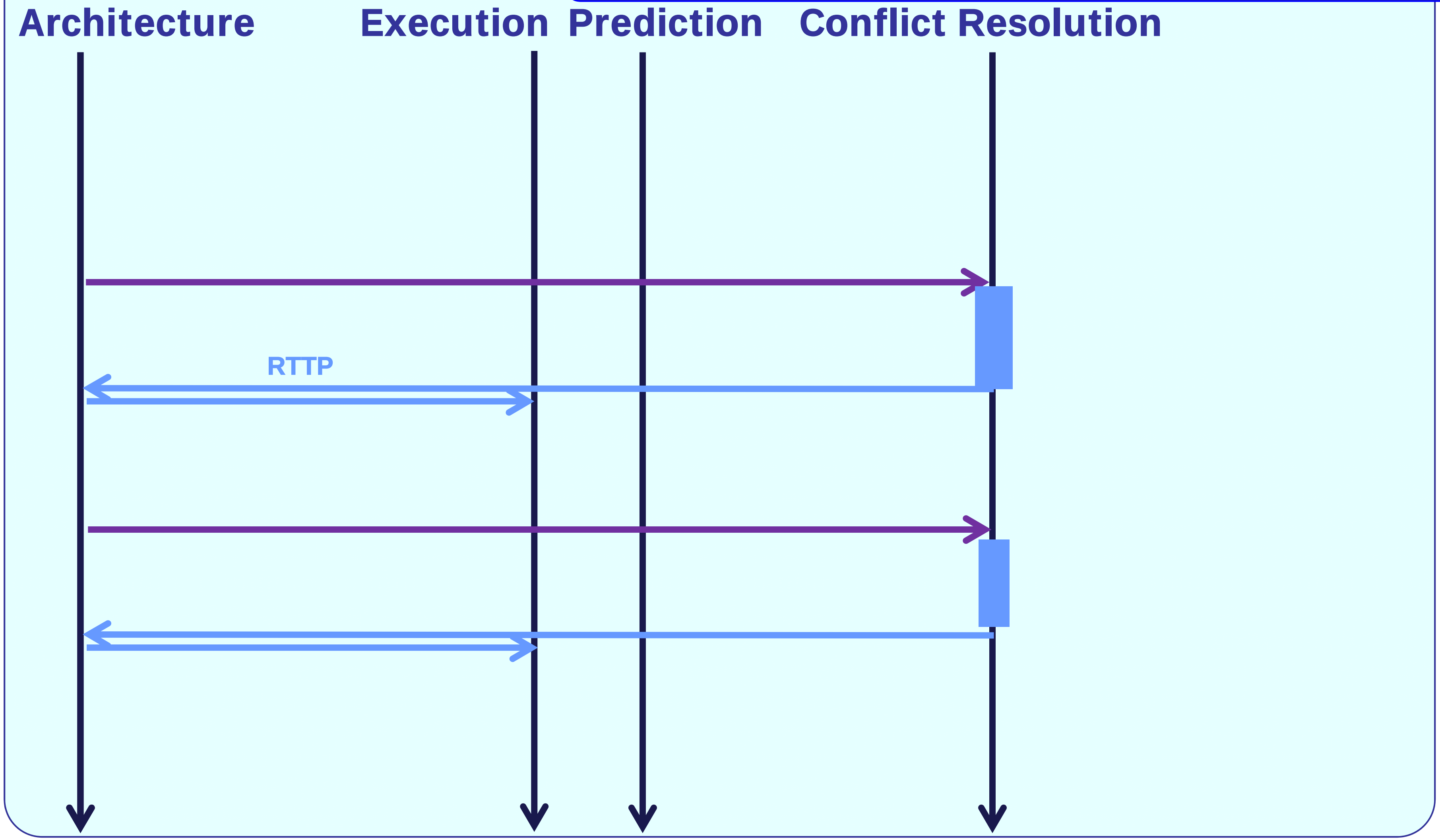
Train 801	
Route	Sections
O-A	TS1
A-B	TS2
	TS3
B-D	TS5
D-E	TS6

Infrastructure view

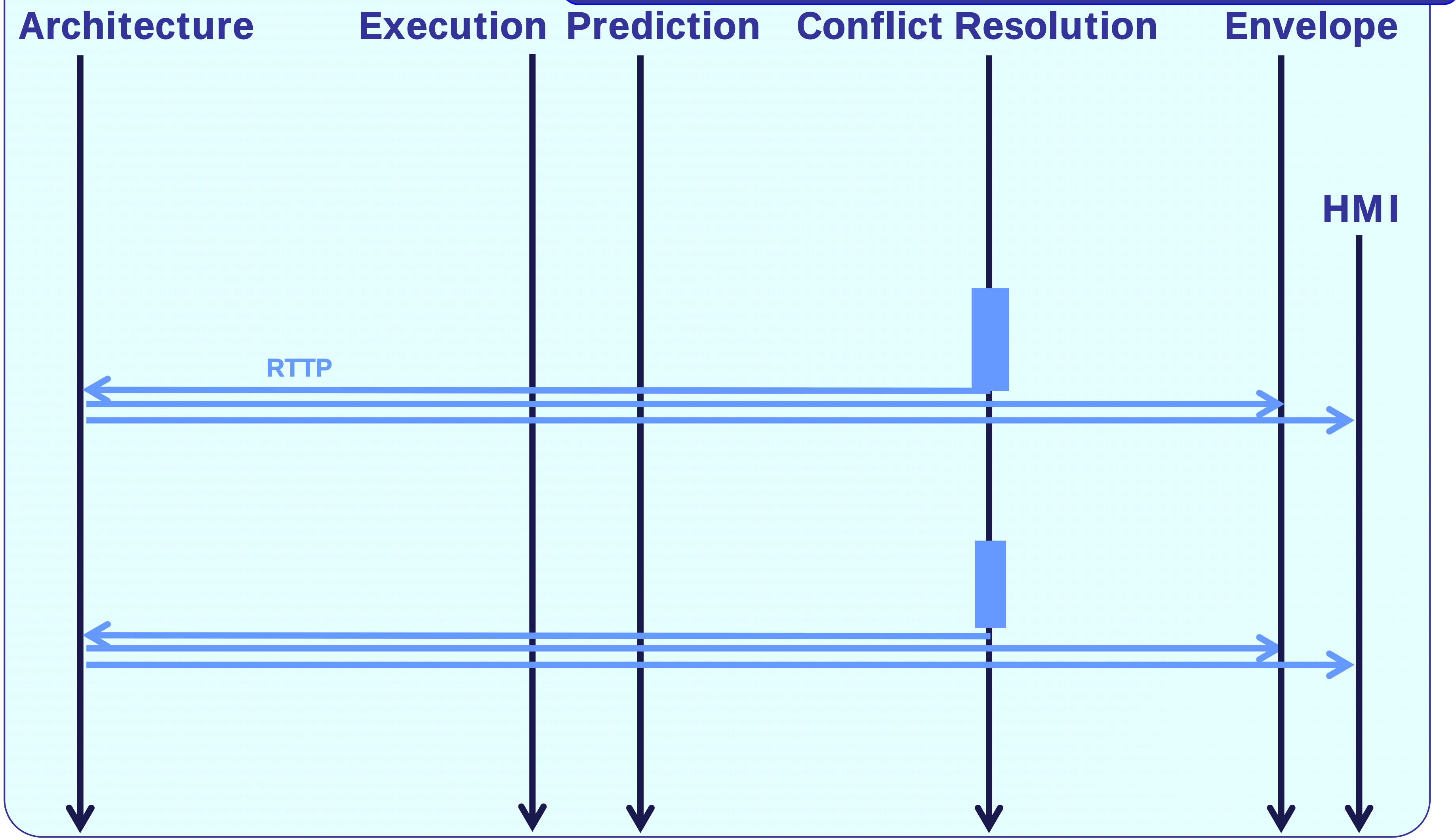
TS1	TS2	TS3	TS4	TS5	TS6
Order	Order	Order	Order	Order	Order
1001	1001	801	1001	801	801
801	801			1001	1001

RTTP = Result of conflict detection and resolution function

Interaction WP4



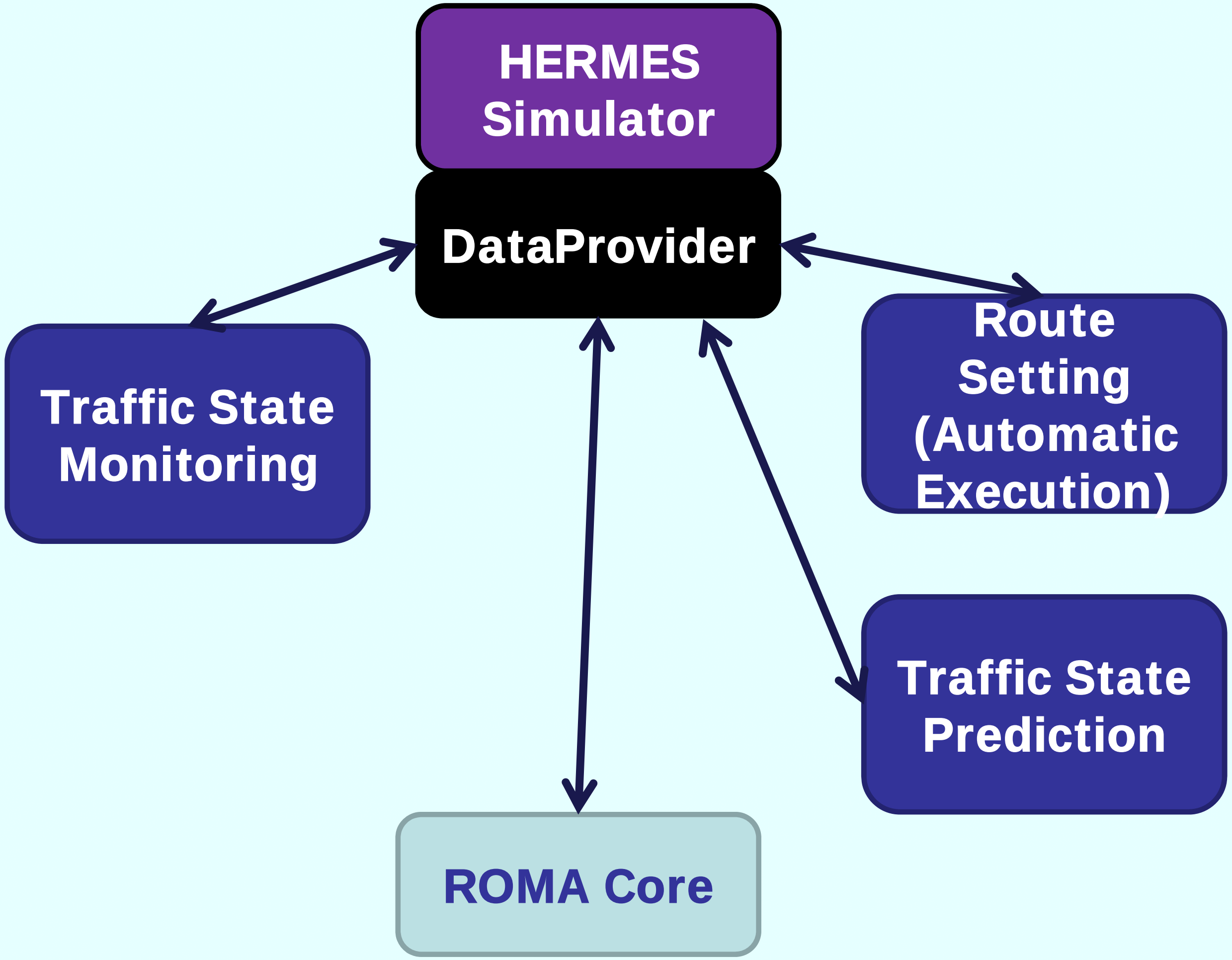
Interaction WP4



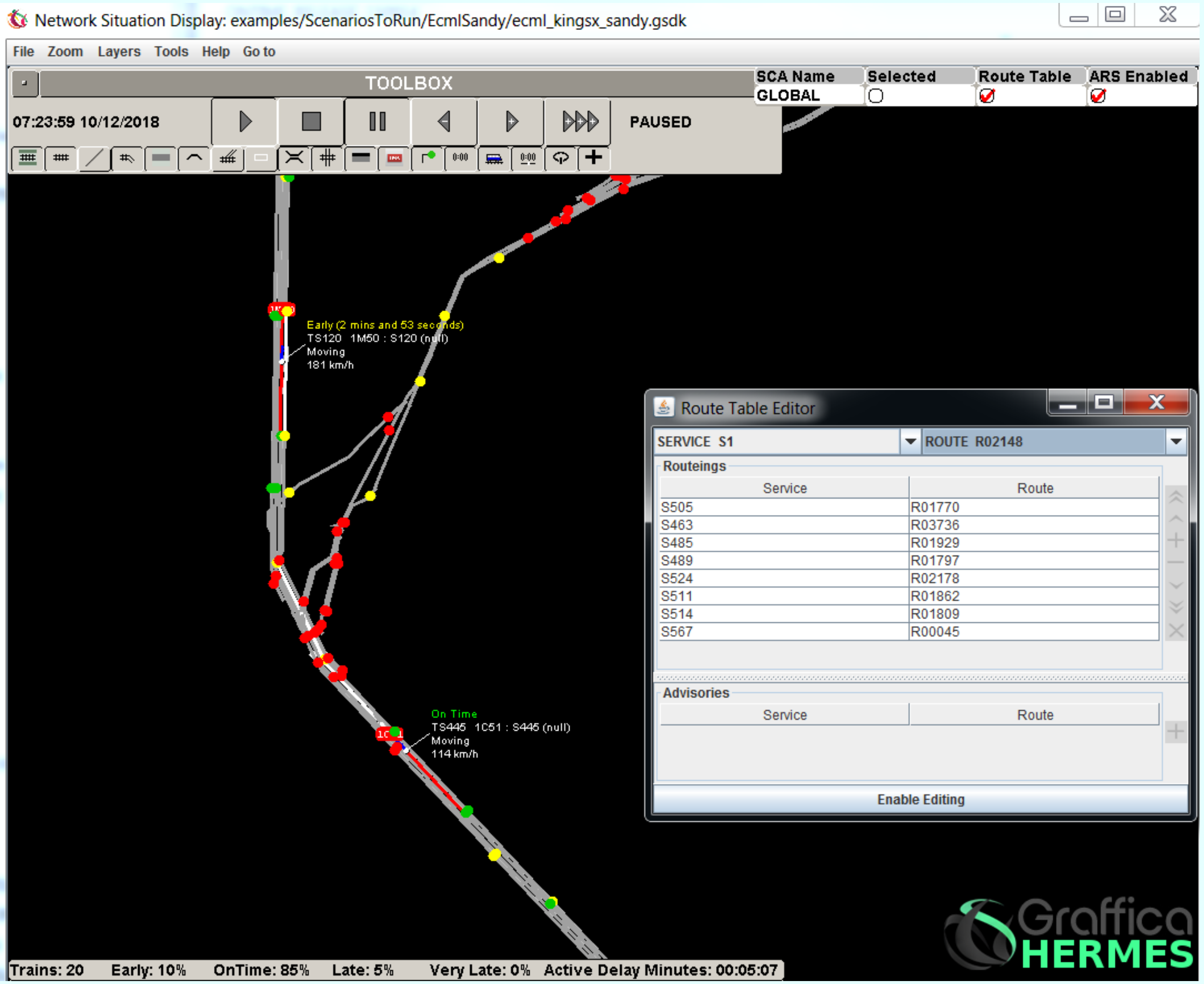
Questions from practice

- These algorithms cover just particular aspects of optimisation. → **Modular system design (1)**
- These algorithms can‘t work in real time.
  → **Real-time environment for data exchange (2)**
- Are these algorithms really efficient in practice? (Are they able to provide additional capacity, less delays etc.?)
  → **Evaluation in simulations (3)**
- These algorithms require data which is not available.
  → **Standardized data model (4)**
- What will our dispatchers say if they are confronted with these algorithms?
  → **Concept for human machine interaction (5)**

Simulator as railway system



HERMES Simulation



Network Situation Display: examples/ScenariosToRun/EcmlSandy/ecml_kingsx_sandy.gsdk

07:23:59 10/12/2018 PAUSED

SCA Name	Selected	Route Table	ARS Enabled
GLOBAL	○	☑	☑

Early (2 mins and 53 seconds)
TS120 1M50 : S120 (null)
Moving
181 km/h

On Time
TS445 1C51 : S445 (null)
Moving
114 km/h

Route Table Editor

SERVICE S1 — ROUTE R02148

Routeings

Service	Route
S505	R01770
S463	R03736
S485	R01929
S489	R01797
S524	R02178
S511	R01862
S514	R01809
S567	R00045

Advisories

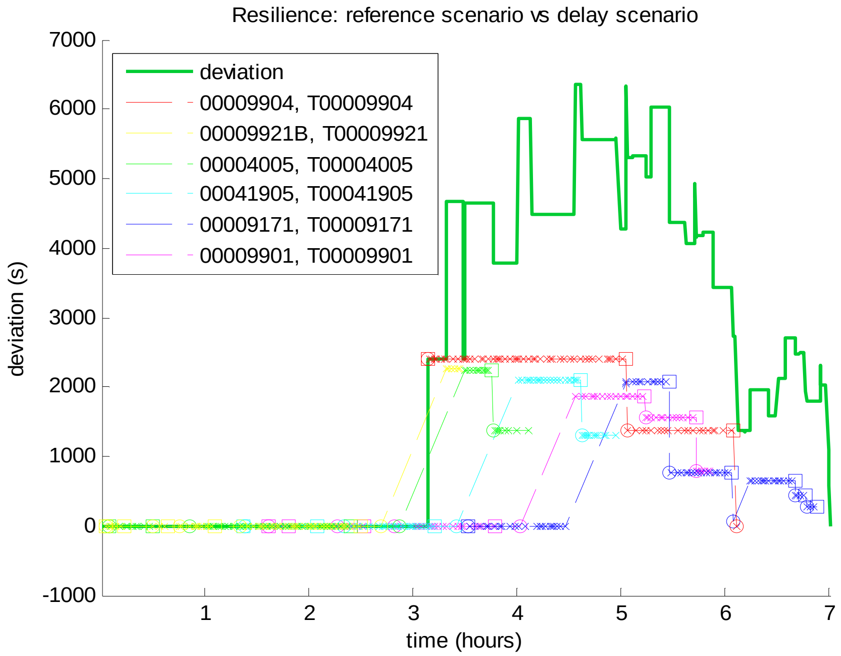
Service	Route

Trains: 20 Early: 10% OnTime: 85% Late: 5% Very Late: 0% Active Delay Minutes: 00:05:07

Results Iron-Ore Line

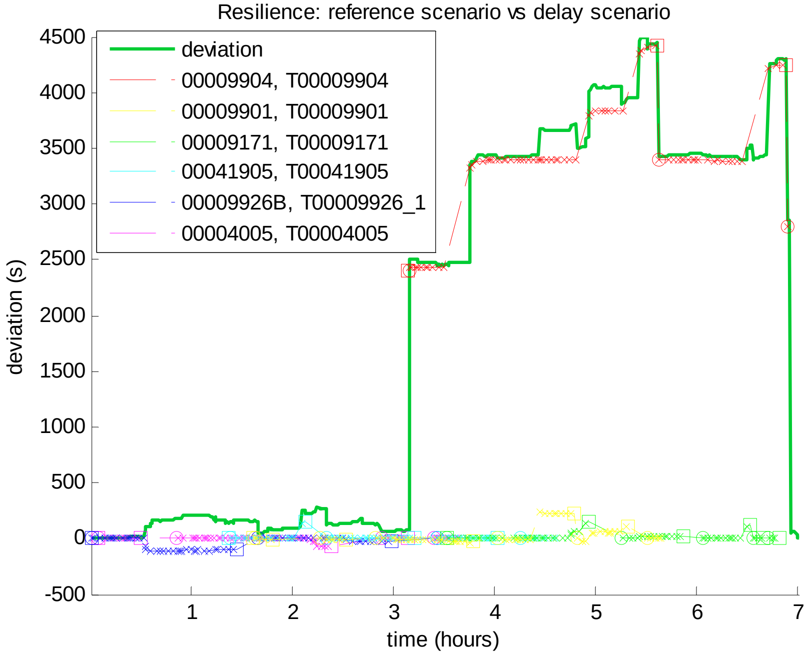
- Scenario 1: Entrance delay train 9904 of 40 min

Without ON-TIME



Deviation area [h^2]	4:15
Maximum delay [h:mm]	1:45
Settling time [h:mm]	3:50

With ON-TIME (ROMA)

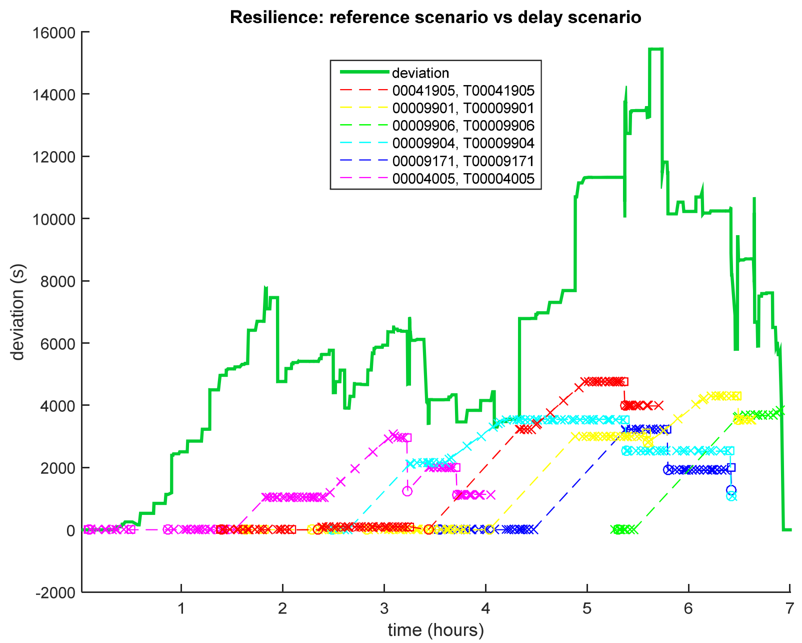


Deviation area [h^2]	3:45
Maximum delay [h:mm]	1:15
Settling time [h:mm]	3:45

Results Iron-Ore Line

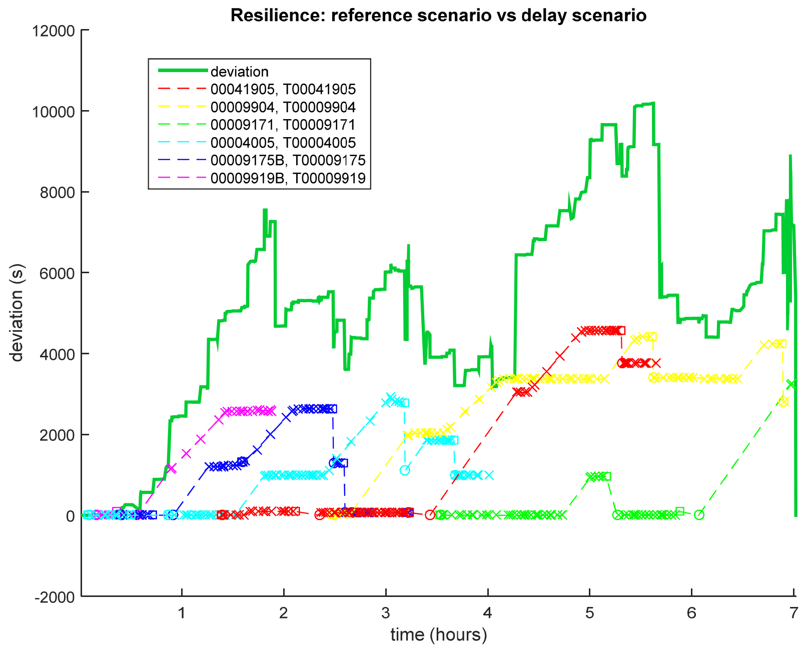
- Scenario 2: Speed restriction 20 km/h Rensjön-Bergfors

Without ON-TIME



Deviation area [h^2]	11:43
Maximum delay [h:mm]	4:17
Settling time [h:mm]	6:19

With ON-TIME (ROMA)



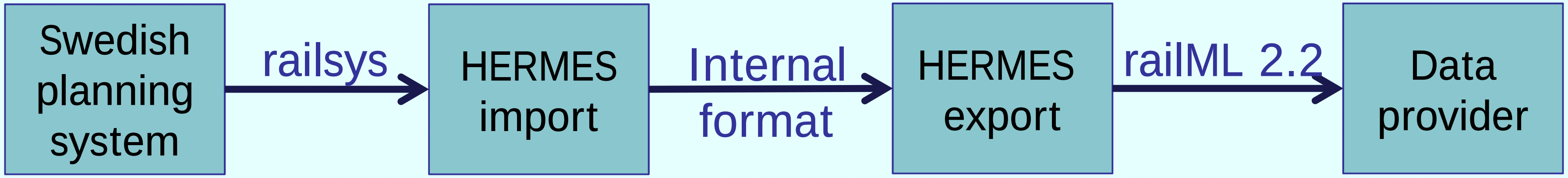
Deviation area [h^2]	9:42
Maximum delay [h:mm]	2:50
Settling time [h:mm]	6:25

Questions from practice

- These algorithms cover just particular aspects of optimisation. → **Modular system design (1)**
- These algorithms can‘t work in real time.
  → **Real-time environment for data exchange (2)**
- Are these algorithms really efficient in practice? (Are they able to provide additional capacity, less delays etc.?)
  → **Evaluation in simulations (3)**
- These algorithms require data which is not available.
  → **Standardized data model (4)**
- What will our dispatchers say if they are confronted with these algorithms?
  → **Concept for human machine interaction (5)**

Standardized data

- railML 2.2
  - Timetable
  - Infrastructure
  - Rolling stock
  - Interlocking (**NEW!**)
- Microscopic data
- Example for Sweden

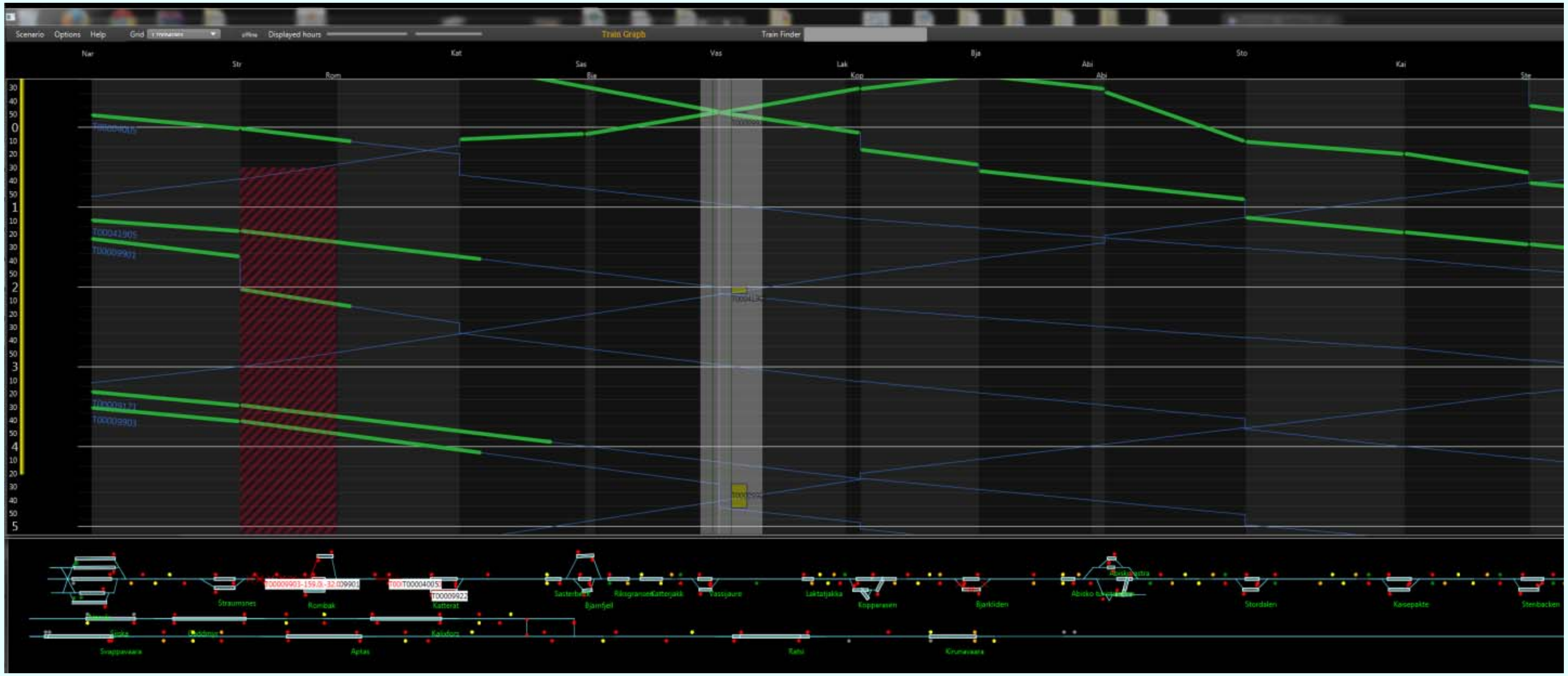


Questions from practice

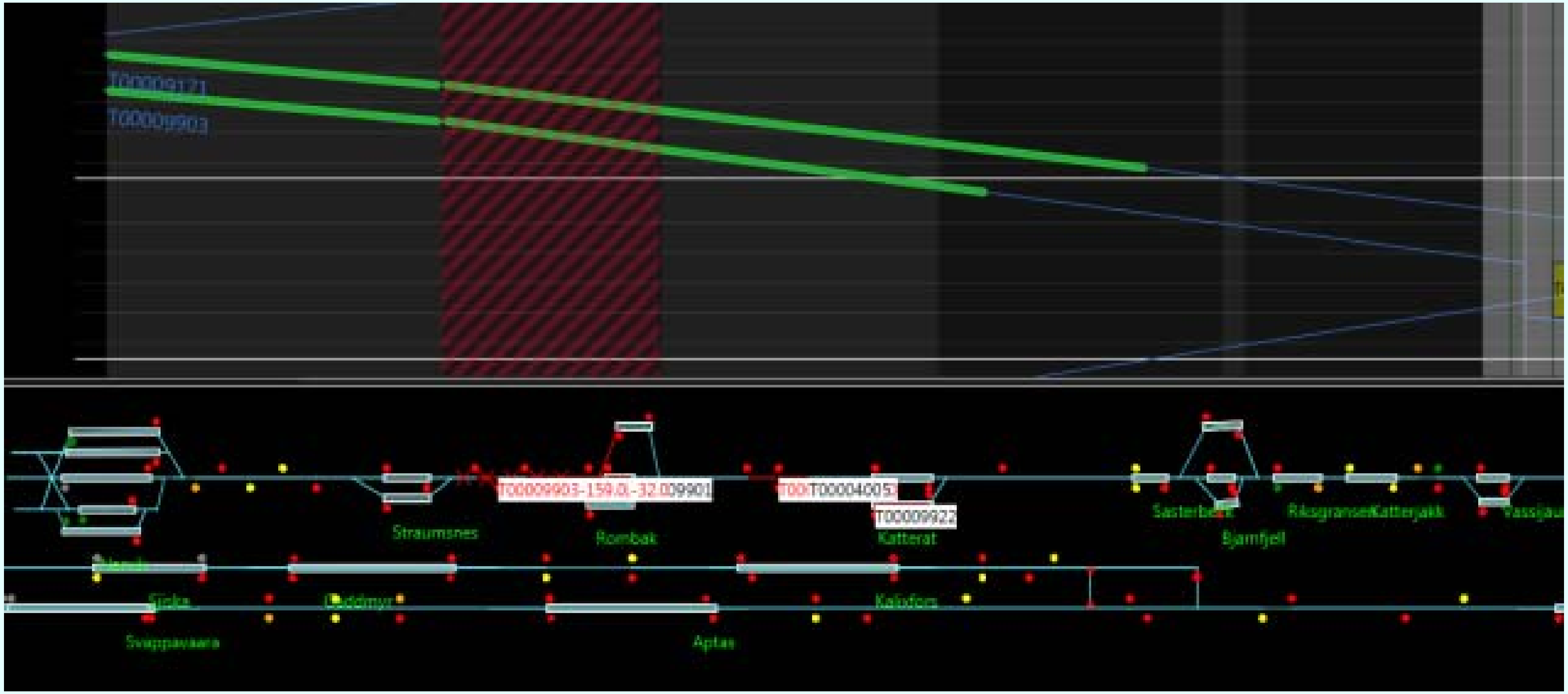
- These algorithms cover just particular aspects of optimisation. → **Modular system design (1)**
- These algorithms can't work in real time.
  → **Real-time environment for data exchange (2)**
- Are these algorithms really efficient in practice? (Are they able to provide additional capacity, less delays etc.?)
  → **Evaluation in simulations (3)**
- These algorithms require data which is not available.
  → **Standardized data model (4)**
- What will our dispatchers say if they are confronted with these algorithms?
  → **Concept for human machine interaction (5)**

Human Machine Integration

- Ansaldo HMI

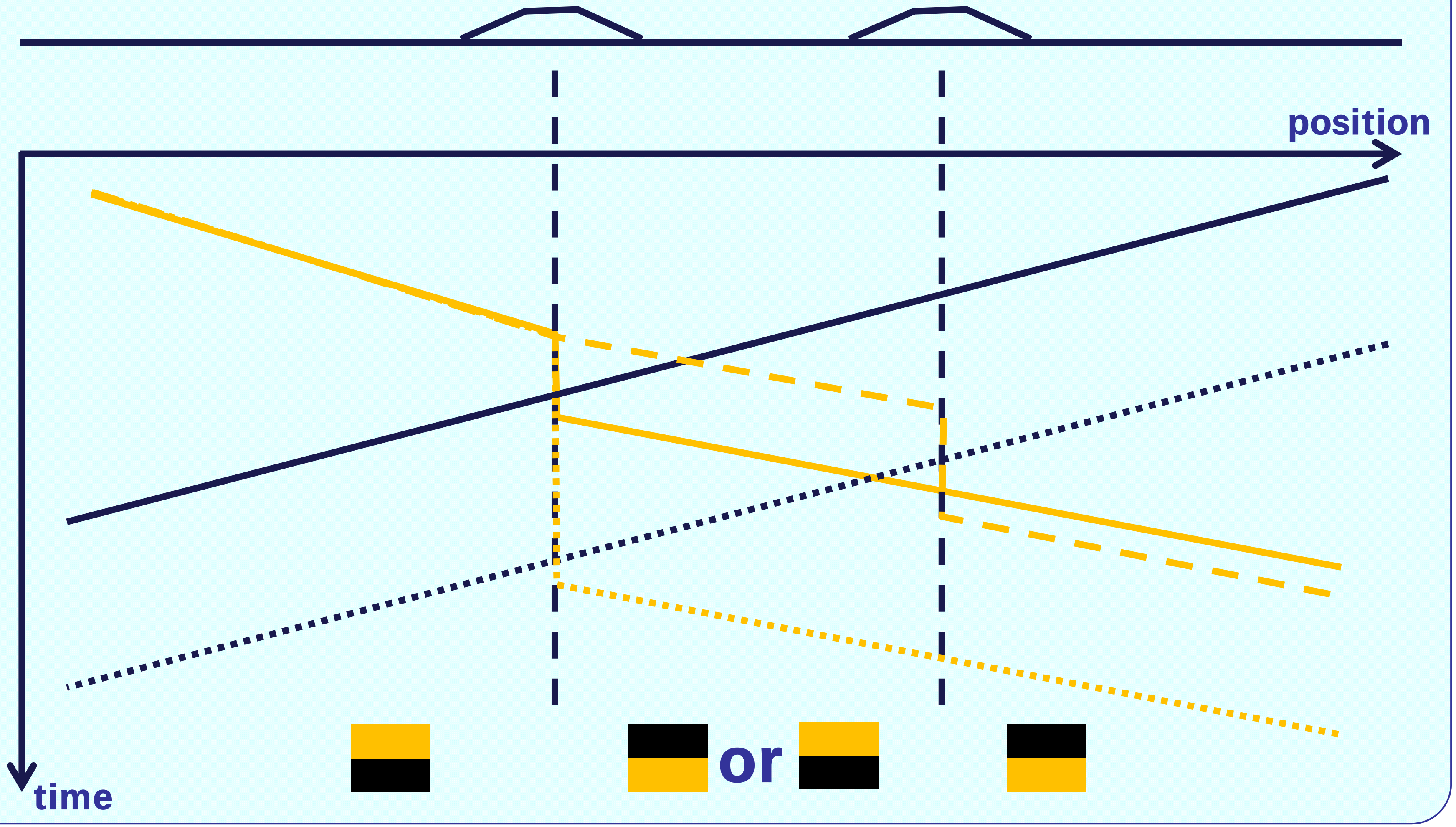


Train describer + Train graph



Example: shift overtaking location

Human constraints to optimization



Open issues

- Including interaction with human
- Including interaction with driving optimization
- More stochasticity in the simulation
- Comparisons with reality

- More detailed data models
  (to see how much is still to gain)

Conclusions / Lessons learnt

- A modular automatic real-time traffic management of small perturbations is feasible
- Modules, tools and experience are available for next steps of test and integration with real rail system

- Testing requires automation, but automation requires „100% solutions“
- Handling with simulators can be as difficult as with real-world railway systems
- Data modelling/consistency and system integration tests were the most underestimated and time-consuming aspects of the project