SET Level

Simulation-based Development and Testing of Automated Driving

Supported by:

Federal Ministry for Economic Affairs and Energy

on the basis of a decision by the German Bundestag

PROJECT of the PEGASUS FAMILY

Simulation Use Case 1 – A traffic simulation for criticality analysis

Arun Das, Günter Ehmen, Thomas Platzer

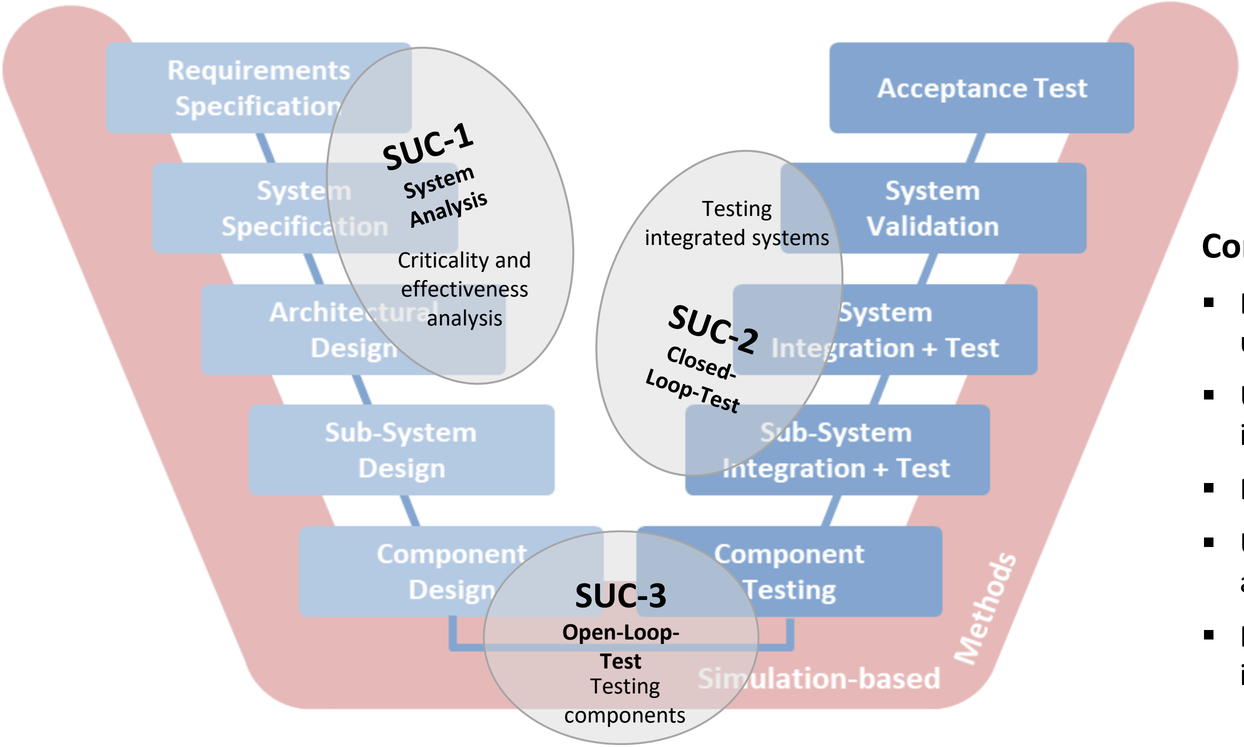
29.04.2021

Agenda

- Introduction of Simulation Use Case 1
- Specification
- Implementation
- Execution and Evaluation
- Summary and Outlook
- Q&A

Introduction of Simulation Use Case 1

Motivation



Simulation Use Cases (SUCs):

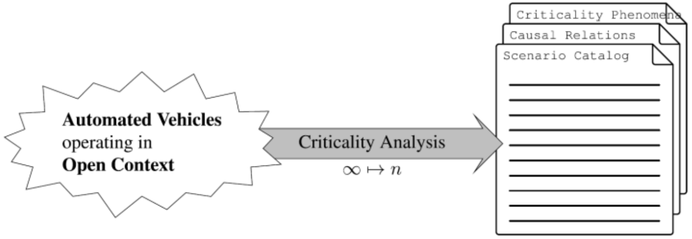
- Analysis example
- Test examples

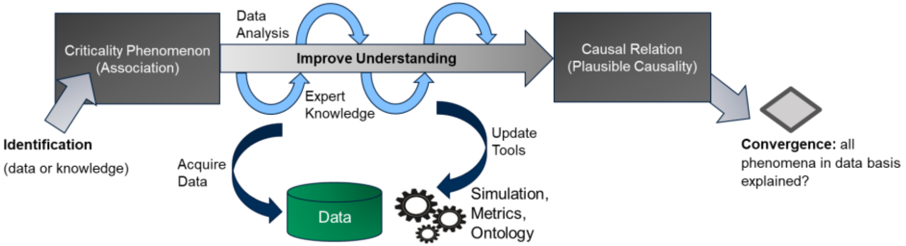
Common Demonstration goals:

- Demonstration of the applicability and usability of standards (OSI, FMI, SSP, …)
- Usage of appropriate architectures and interfaces
- Elaboration of KPIs
- Use of the credible simulation process and assurance of traceability
- Provide project internal feedback and identify needs for further work

Introduction of Simulation Use Case 1

Criticality Analysis Overview [1]





Criticality Analysis – Basic Concept (VV Methoden, TP2)

Map an infinite-dimensional domain onto a finite and manageable set of artifacts that capture and explain the emergence of critical situations for automated vehicles.

Core steps:

- Extract criticality phenomena
- Identify underlying causal relations
- Use abstraction and classification of causal relations for scenario space condensation

The approach is based on a combination of expert knowledge and data driven methods. Simulation can be applied to analyze criticality in various ways, e.g. to find critical scenarios or check plausibility of causal relations.

Key elements for simulation:

- Variation of traffic scenarios
- Evaluation metrics for criticality

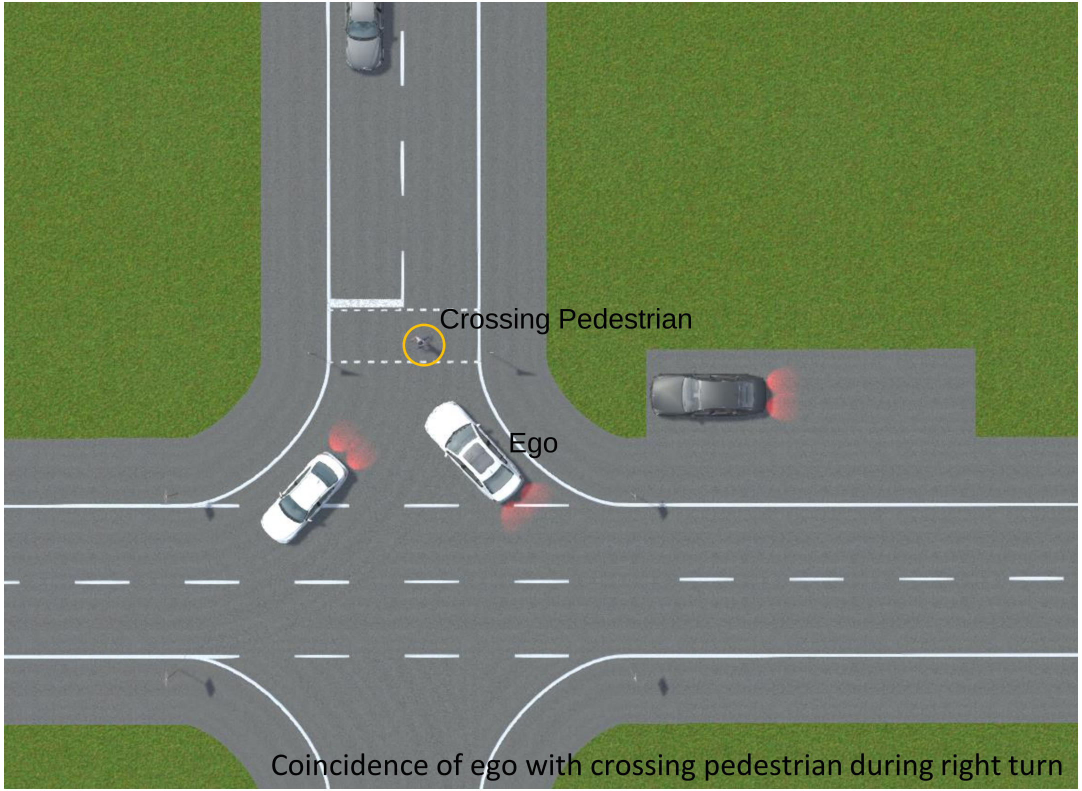
[1] Neurohr, Christian & Westhofen, Lukas & Butz, Martin & Bollmann, Martin & Eberle, Ulrich & Galbas, Roland. (2021). Criticality Analysis for the Verification and Validation of Automated Vehicles. IEEE Access. PP. 1-1. 10.1109/ACCESS.2021.3053159.

Introduction of Simulation Use Case 1

Simulation Goal

„Identify critical scenarios during a right turn on an urban crossing through simulation.“

Right turn is chosen as it may contain risks that can lead to a critical coincidence.



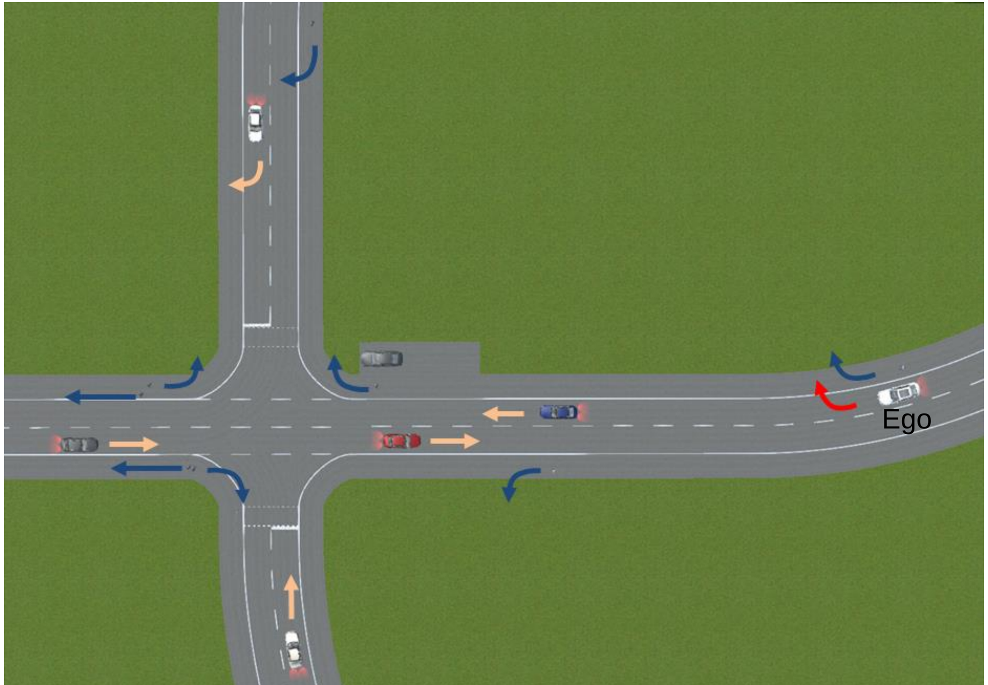
Introduction of Simulation Use Case 1

Simulation Goal

„Identify critical scenarios during a right turn on an urban crossing through simulation.“

Scenario Setup:

- Simple crossing with priority road
- Ego vehicle with automated driving function
- 6 surrounding vehicles with predefined destinations
- 8 pedestrians with predefined destinations
  2 pedestrians have to cross the street
- Evaluation of time to collision (TTC) and
  post encroachment time (PET)



Scenario setup with predefined routes

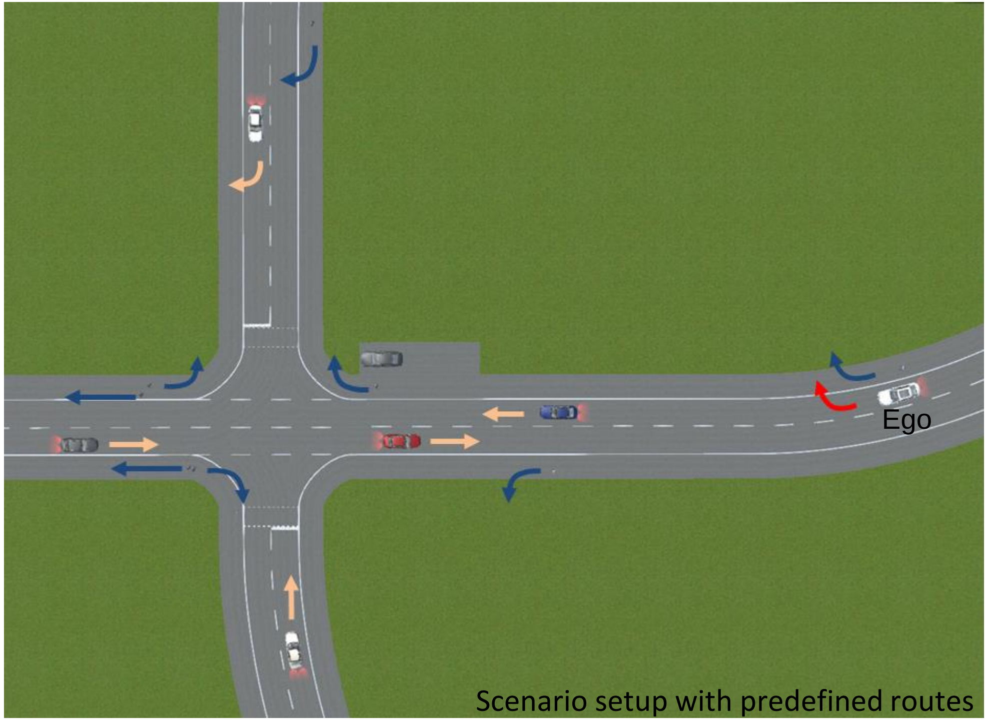
Introduction of Simulation Use Case 1

Simulation Goal

„Identify critical scenarios during a right turn on an urban crossing through simulation.“

Scenario Setup:

- Simple crossing with priority road
- Ego vehicle with automated driving function
- 6 surrounding vehicles with predefined destinations
- 8 pedestrians with predefined destinations
  2 pedestrians have to cross the street
- Evaluation of time to collision (TTC) and
  post encroachment time (PET)



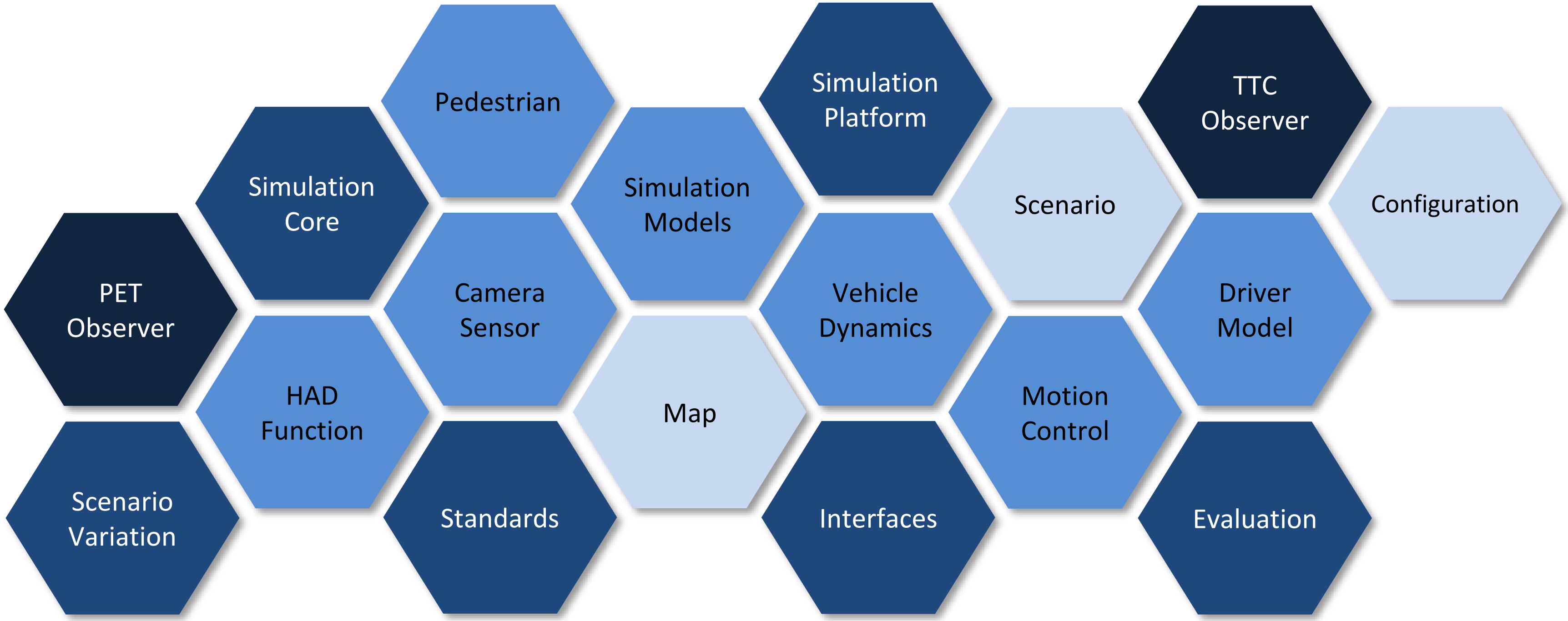
Scenario setup with predefined routes

Focus of the SUC-1:

- Specification and execution of analysis task
- Applicability of architecture and interfaces
- Utilization of standards
- Variation of the scenario
- Evaluation of criticality using corresponding metrics

Specification

Building Blocks for the Simulation



Specification

SET Level

Stochastic Variation on Scenario Level

Functional Scenario — *Abstract description of the scenario*

„Right turning ego agent on simple junction. Surrounding traffic …“

↓ Formalization

Logical Scenario — *Formal scenario description with specification of parameter distributions*

```xml
<parameter name="$Ego_Spawn_Position_X" unit="m" type="continuous">
    <distribution>
        <normalDistribution mean="60.0"
                            standardDev="7.0"
                            lowerLimit="35.0"
                            upperLimit="82.0"/>
    </distribution>
</parameter>
```

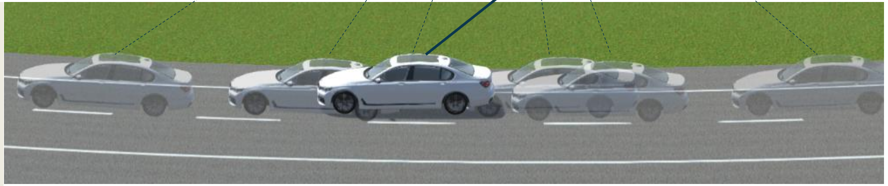
↓ Parameter sampling

Concrete Scenario — *Formal scenario description with fixed parameter values*

```xml
<ParameterDeclaration name="$Ego_Spawn_Position_X"
                      parameterType="double"
                      value="63.7"/>
```

↓ Input for simulation

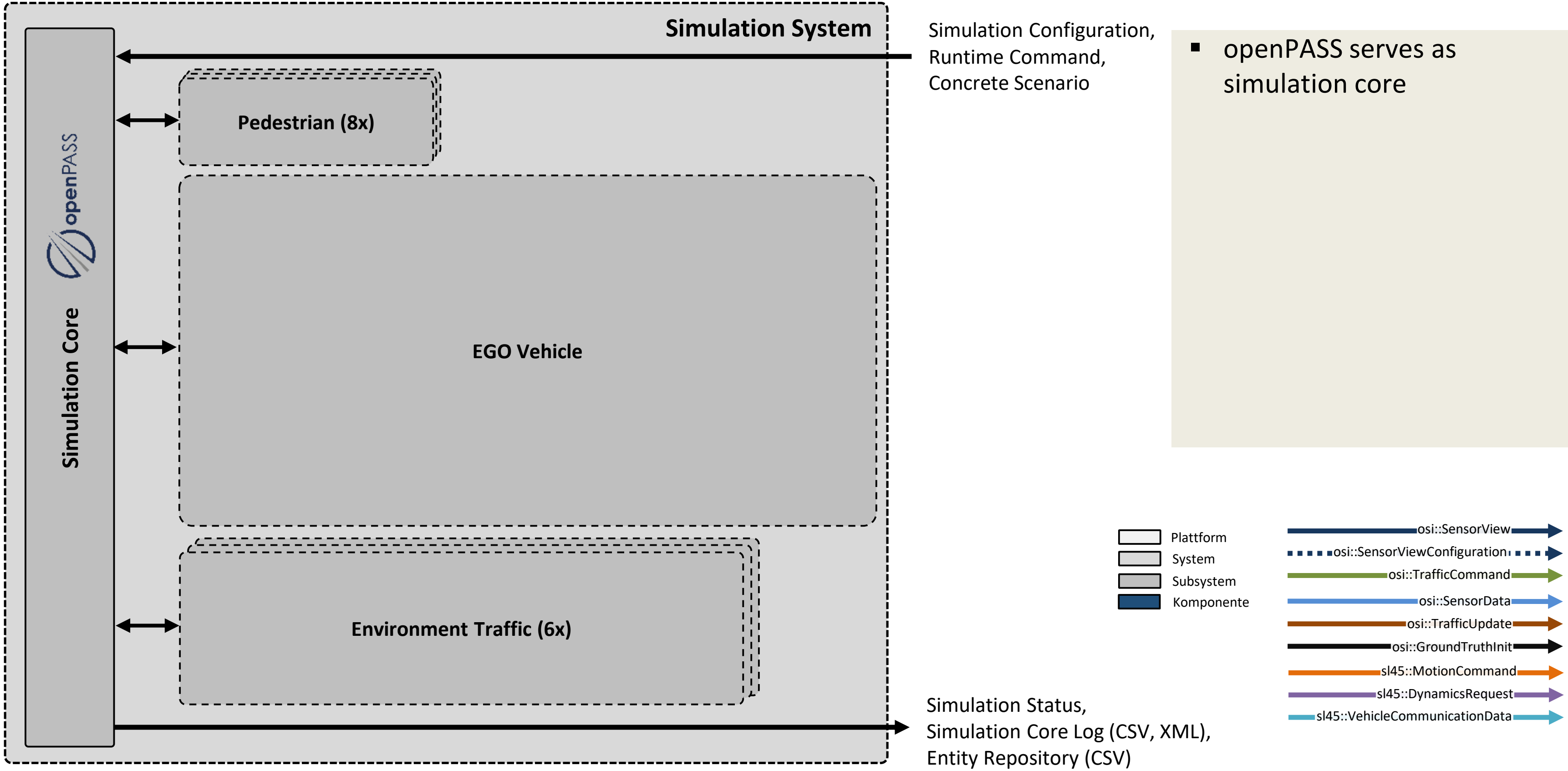
- Specification of stochastic and deterministic variations
- Definition of limits (truncated distributions)
- ➢ Input for OpenSCENARIO 1.1



Stochastic variation of initial position

Implementation

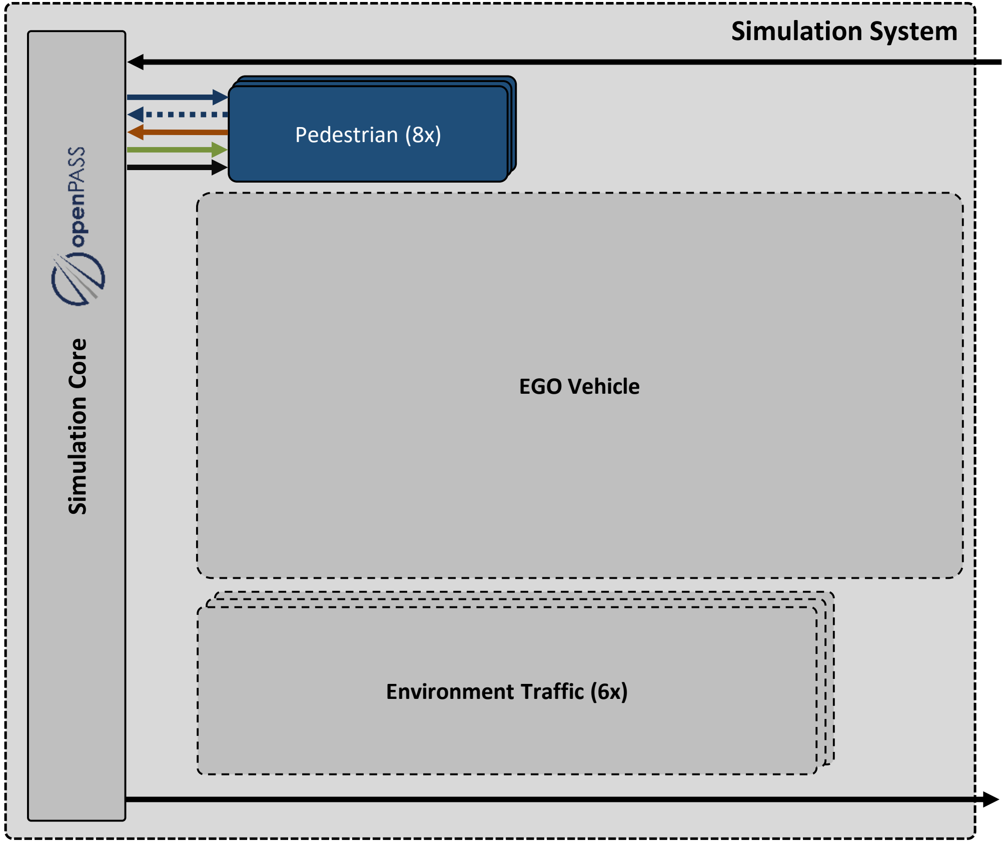
Simulation System Architecture



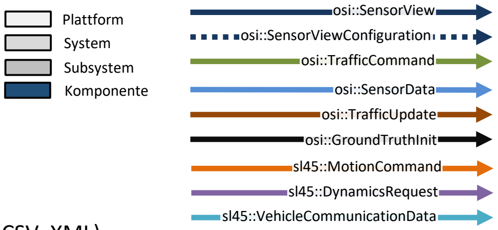
- openPASS serves as simulation core

Implementation

Simulation System Architecture



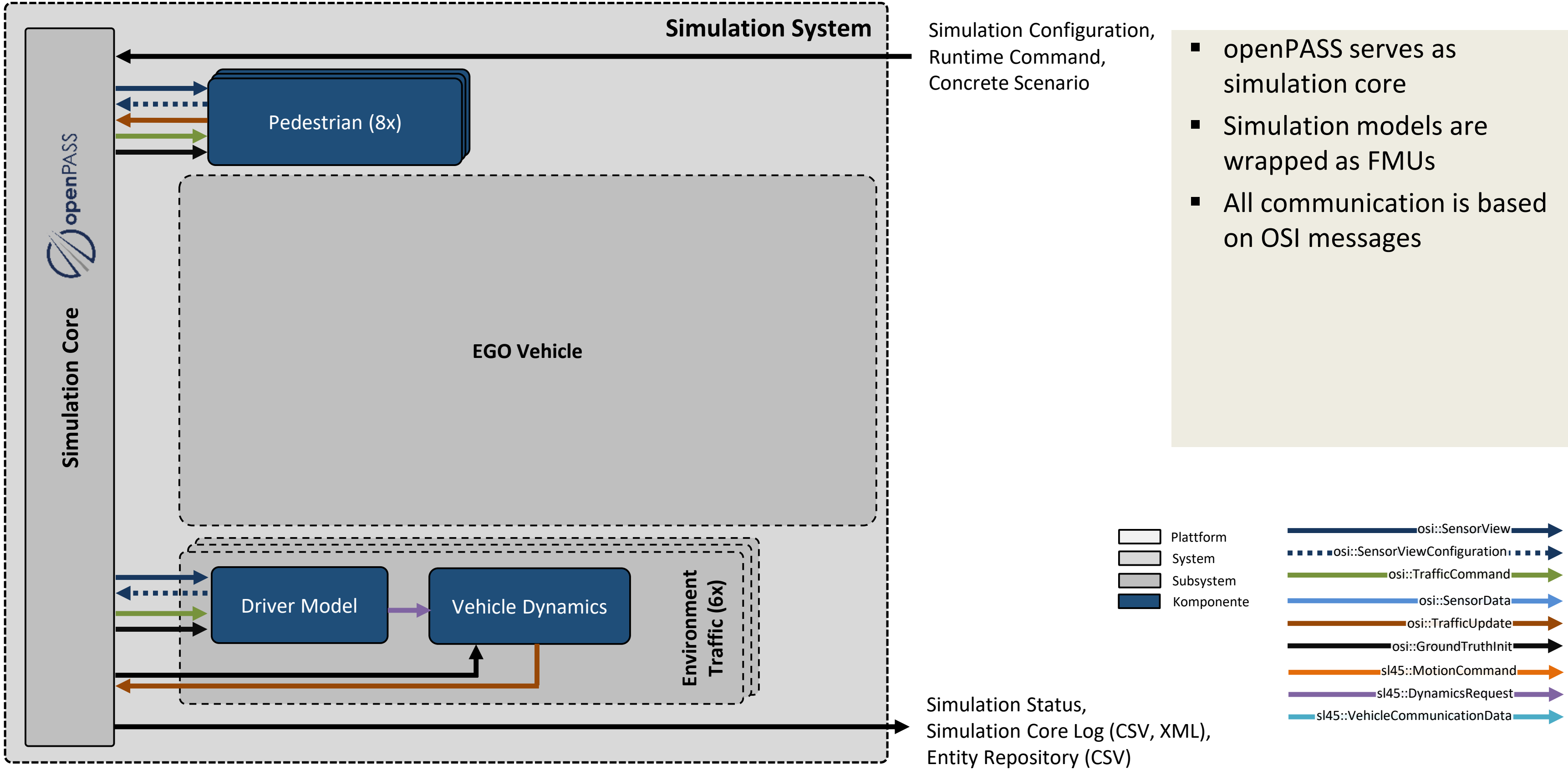
- openPASS serves as simulation core
- Simulation models are wrapped as FMUs
- All communication is based on OSI messages



Implementation

Simulation System Architecture

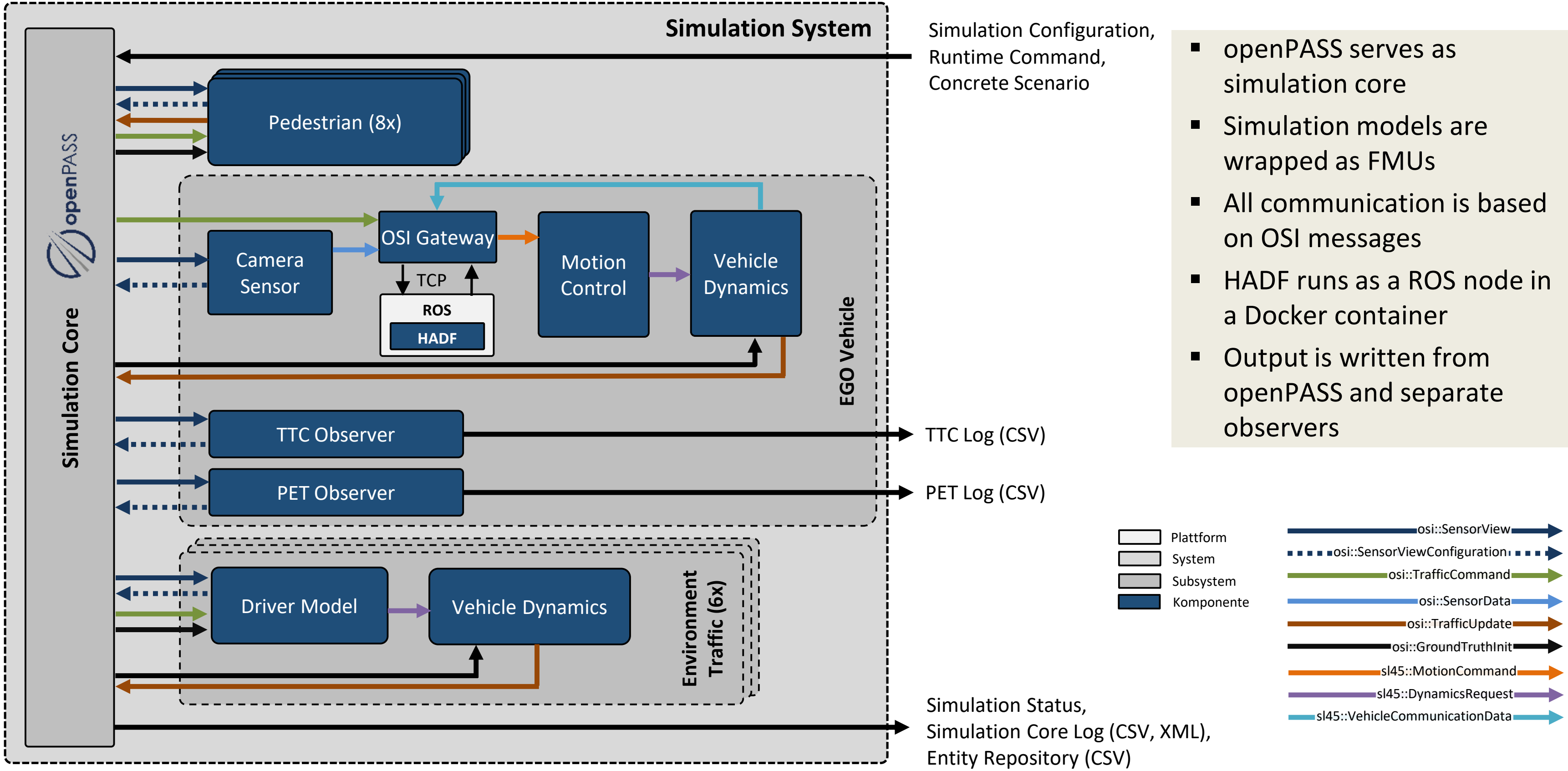
SET Level



- openPASS serves as simulation core
- Simulation models are wrapped as FMUs
- All communication is based on OSI messages

Implementation

Simulation System Architecture



- openPASS serves as simulation core
- Simulation models are wrapped as FMUs
- All communication is based on OSI messages
- HADF runs as a ROS node in a Docker container
- Output is written from openPASS and separate observers

Implementation

Simulation Models

SET Level

Camera Sensor

Technology: OSMP FMU (C++)
Inputs: osi3::SensorView
Outputs: osi3::SensorData
Features:

- Object based model
- Occlusion
- FOV – uncertainties at the edges
- Range - uncertainties in the distance
- State uncertainties (Position, velocity)

HAD Function

Technology: ROS, Docker, TCP
Inputs: osi3::SensorData
sl45::VehicleCommunicationData
osi3::TrafficCommand
Outputs: sl45::MotionCommand
Features:

- Plan and follow route
- Sensor fusion and object tracking
- Trajectory planning

Motion Control

Technology: Simulink, FMU
Inputs: sl45::MotionCommand
Outputs: sl45::DynamicsRequest
Features:

- Nonlinear state control for trajectory input
- Calculation of acceleration and curvature request

Vehicle Dynamics

Technology: Simulink, FMU
Inputs: sl45::DynamicsRequest
Outputs: osi3::TrafficUpdate
Features:

- 2D kinematic single track model
- Ideal actuators
- Vehicle dynamics limits based on physical effects (e.g. powertrain)

Driver Model

Technology: OSMP FMU (C++)
Inputs: osi3::SensorView
osi3::TrafficCommand
Outputs: sl4to5::DynamicsRequest
osi3::TrafficUpdate
Features:

- Plan and follow route
- Give right of way, consider speed limits
- Internal vehicle dynamics model

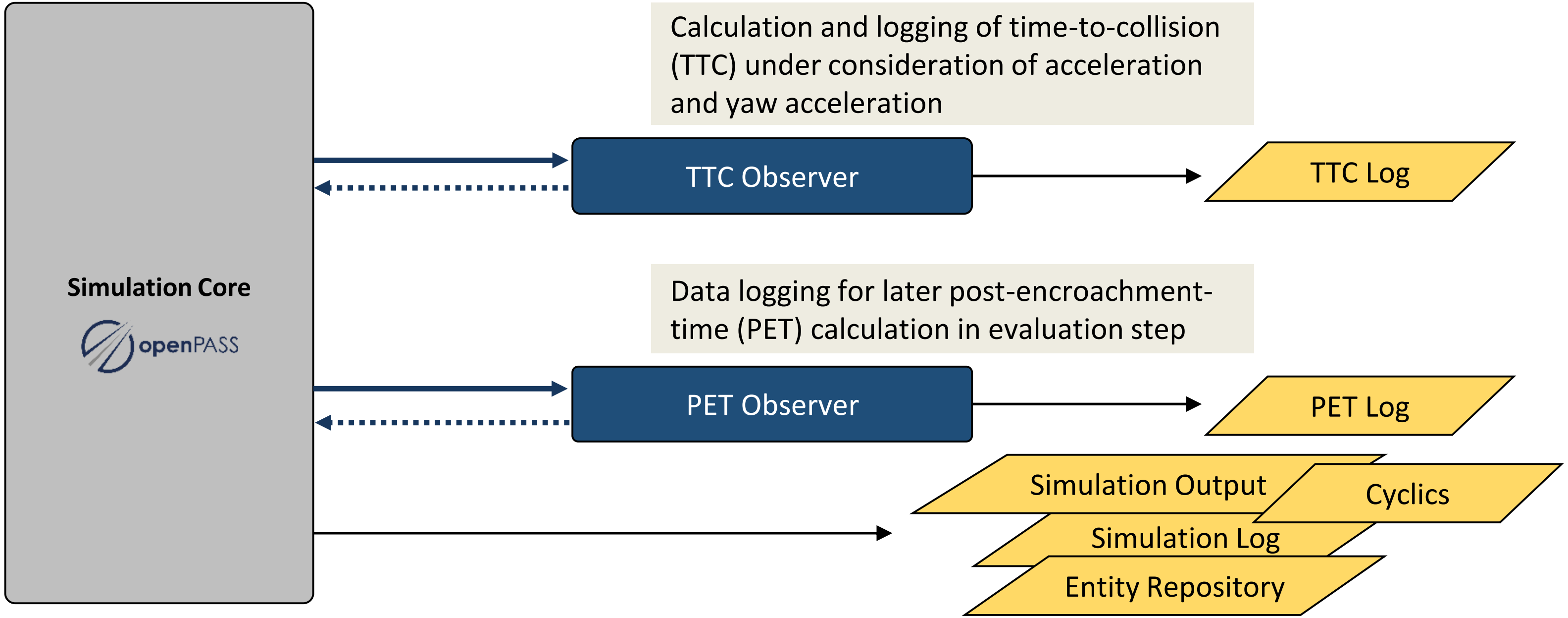
Pedestrian Model

Technology: OSMP FMU (C++)
Inputs: osi3::GroundTruth
osi3::SensorView
osi3::TrafficCommand
Outputs: osi3::TrafficUpdate
Features:

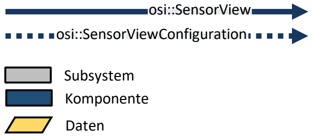
- Path-finding via Theta*
- Social force model for pedestrian collision avoidance

Implementation

Observer and Simulation Outputs

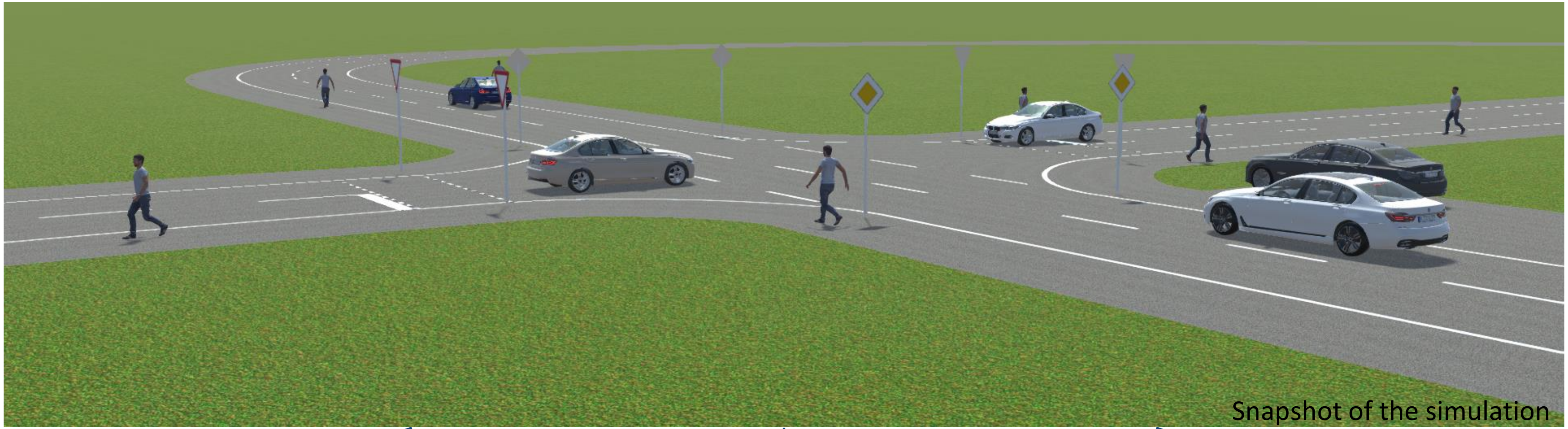


- Observers are independent FMUs communicating through osi::SensorView messages
- Observers may perform internal calculations or solely log data
- Outputs are generated from the simulation core directly and additionally from observers



Implementation

Configuration



Snapshot of the simulation

OpenDRIVE 1.6

- Road infrastructure
- Road markings
- Traffic signs

OpenSCENARIO 1.0

- Agent definitions
- Initial positions and speed
- Target positions and speed

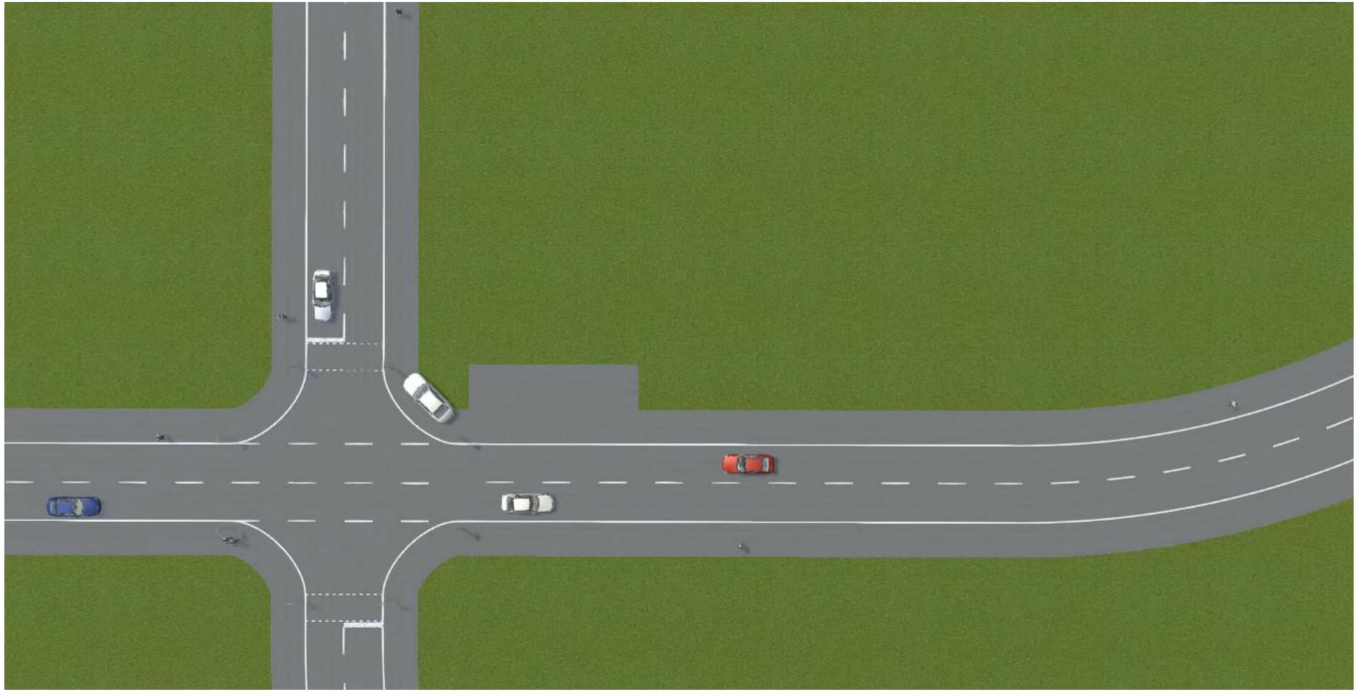
SSP 1.0 + openPASS specifics

- Agent setup
- Logging settings

Implementation

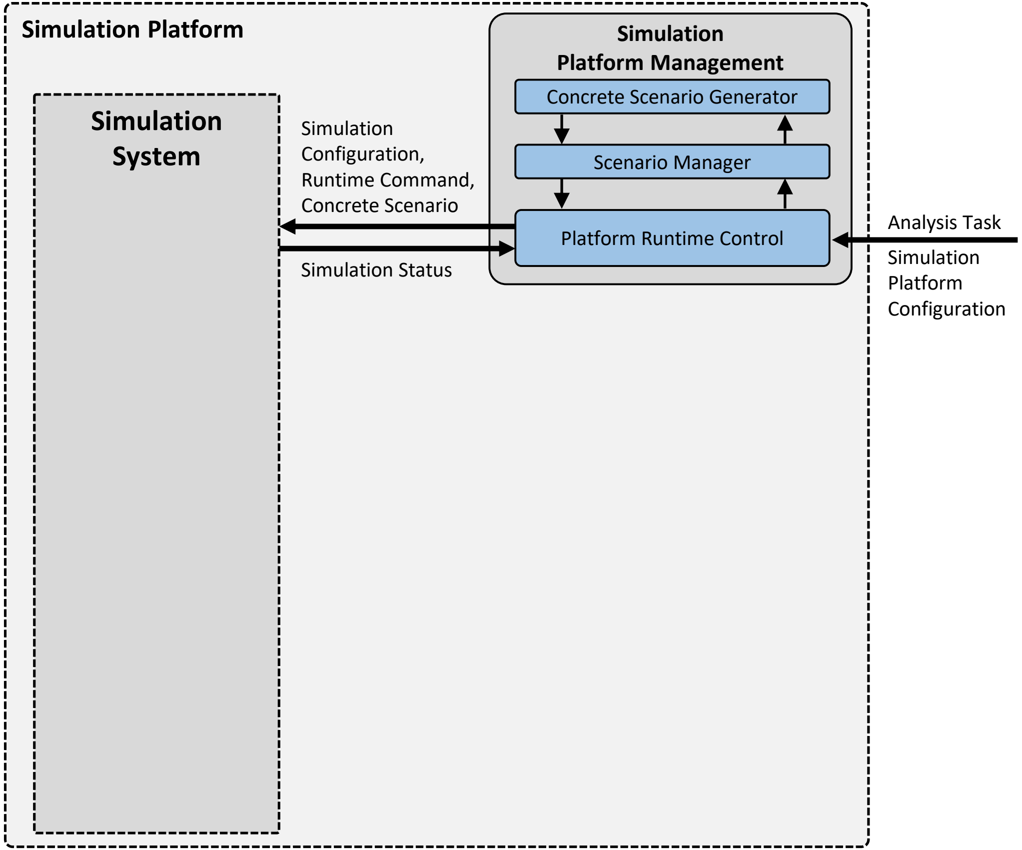
Demo

SET Level



Implementation

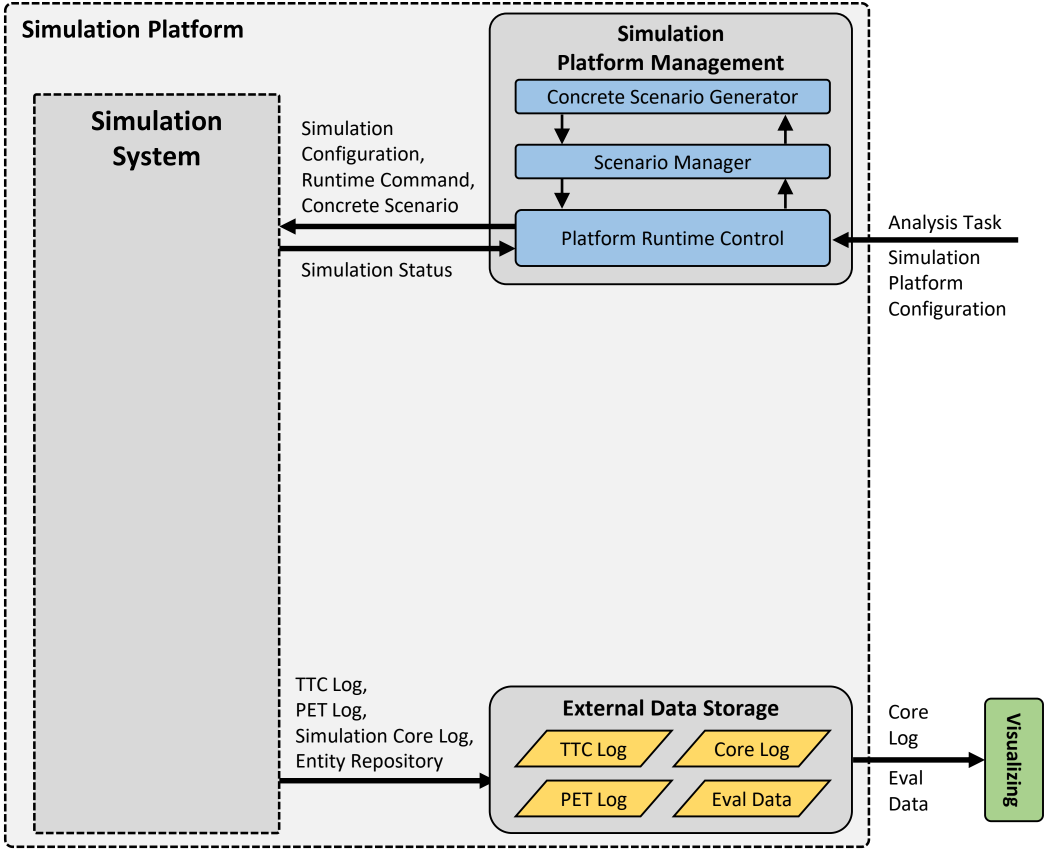
Simulation Platform Architecture



- The simulation platform is configured through analysis task and simulation platform configuration files
- The logical scenario is converted into multiple concrete scenarios based on parameter variations
- A simulation is triggered for each concrete scenario

Implementation

Simulation Platform Architecture

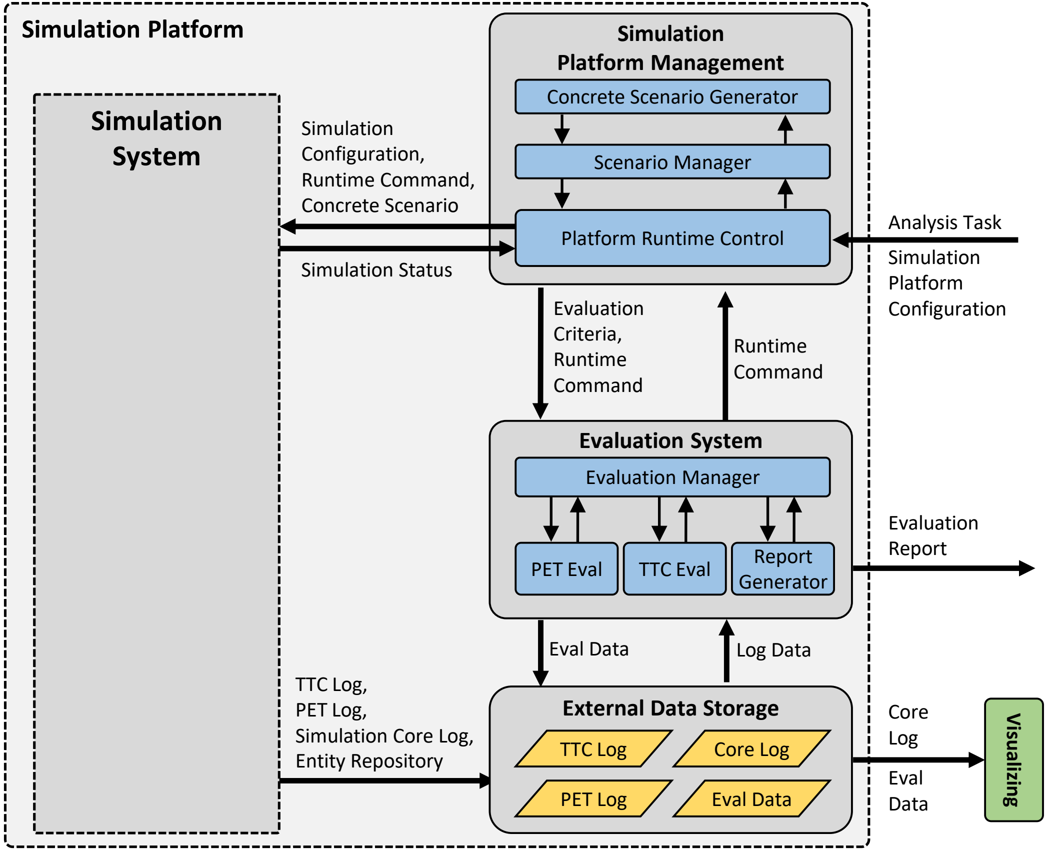


- The simulation platform is configured through analysis task and simulation platform configuration files
- The logical scenario is converted into multiple concrete scenarios based on parameter variations
- A simulation is triggered for each concrete scenario
- Output data is saved as files on the local filesystem

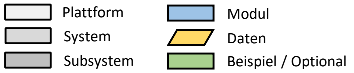
Plattform		Modul	
System		Daten	
Subsystem		Beispiel / Optional	

Implementation

Simulation Platform Architecture

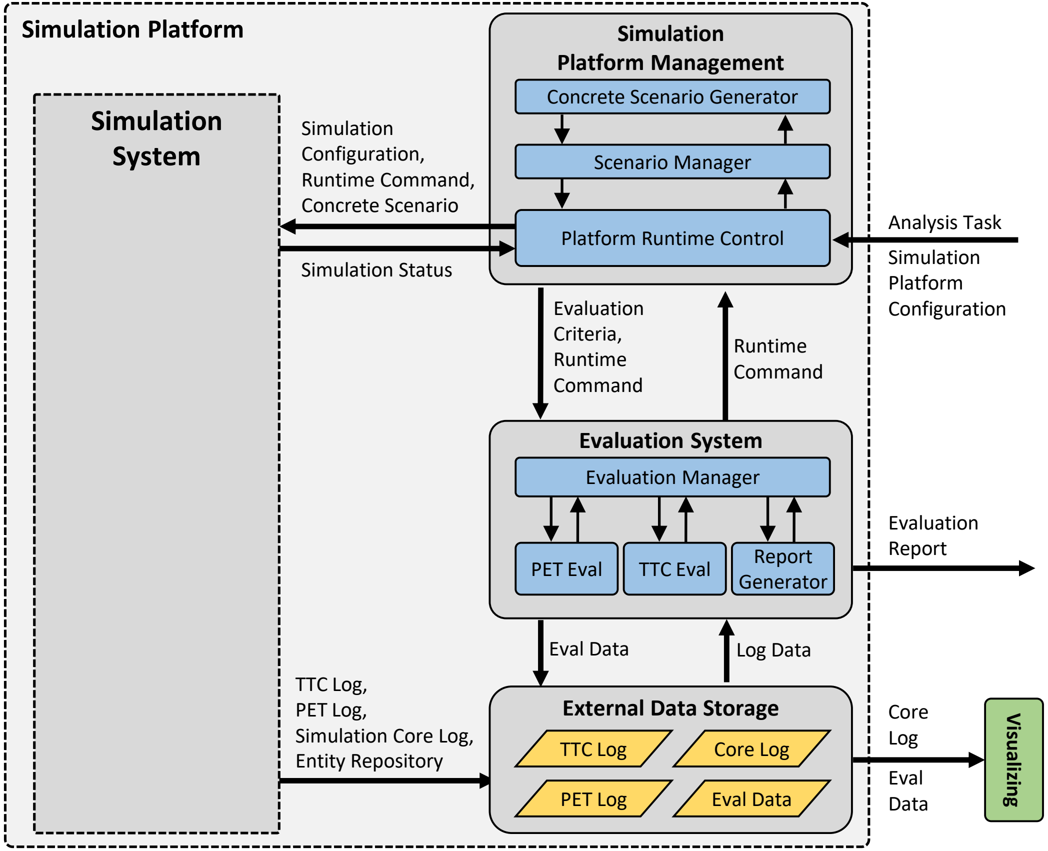


- The simulation platform is configured through analysis task and simulation platform configuration files
- The logical scenario is converted into multiple concrete scenarios based on parameter variations
- A simulation is triggered for each concrete scenario
- Output data is saved as files on the local filesystem
- Criticality is evaluated based on metrics (e.g. TTC, PET) and presented in a report (Markdown, CSV)

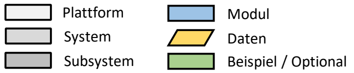


Implementation

Simulation Platform Architecture

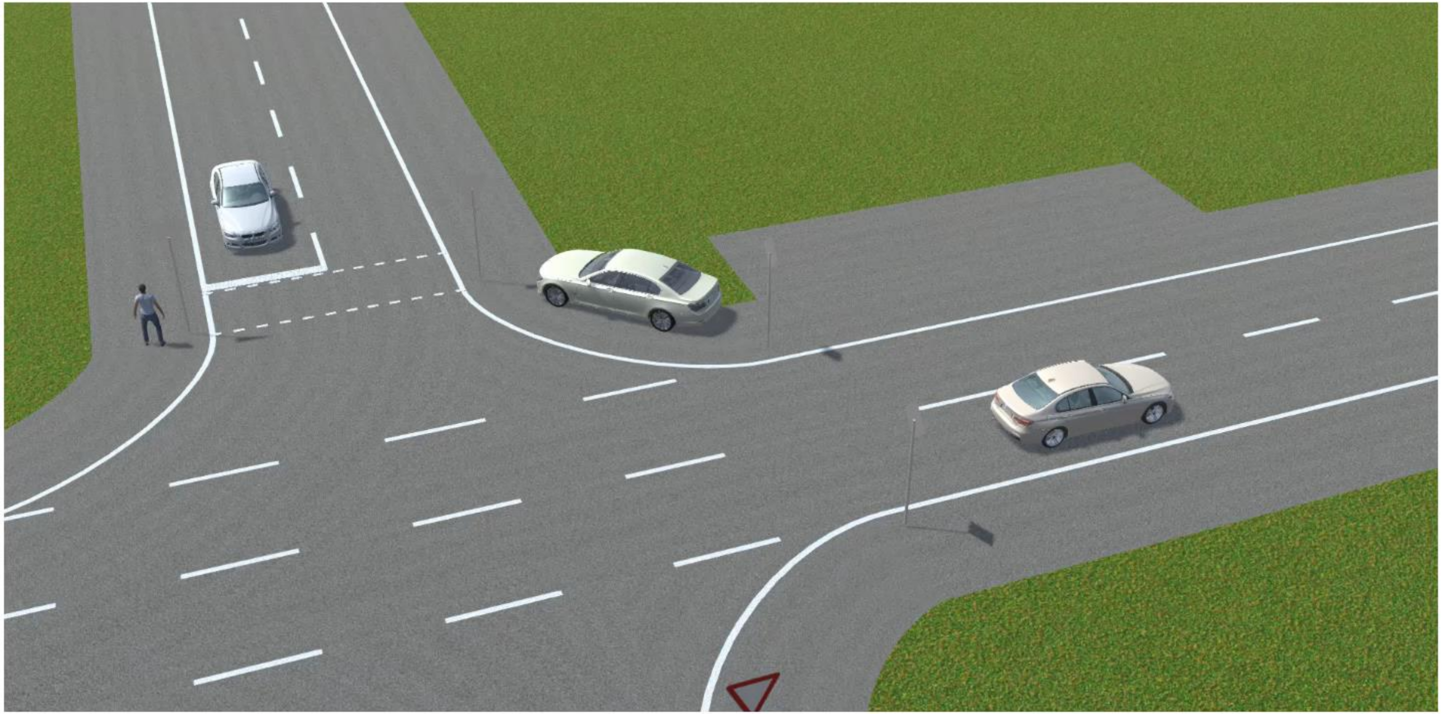


- The simulation platform is configured through analysis task and simulation platform configuration files
- The logical scenario is converted into multiple concrete scenarios based on parameter variations
- A simulation is triggered for each concrete scenario
- Output data is saved as files on the local filesystem
- Criticality is evaluated based on metrics (e.g. TTC, PET) and presented in a report (Markdown, CSV)
- Platform management and evaluation are implemented in Python
- The whole workflow from sampling concrete scenarios to the generation of the evaluation report is fully automated



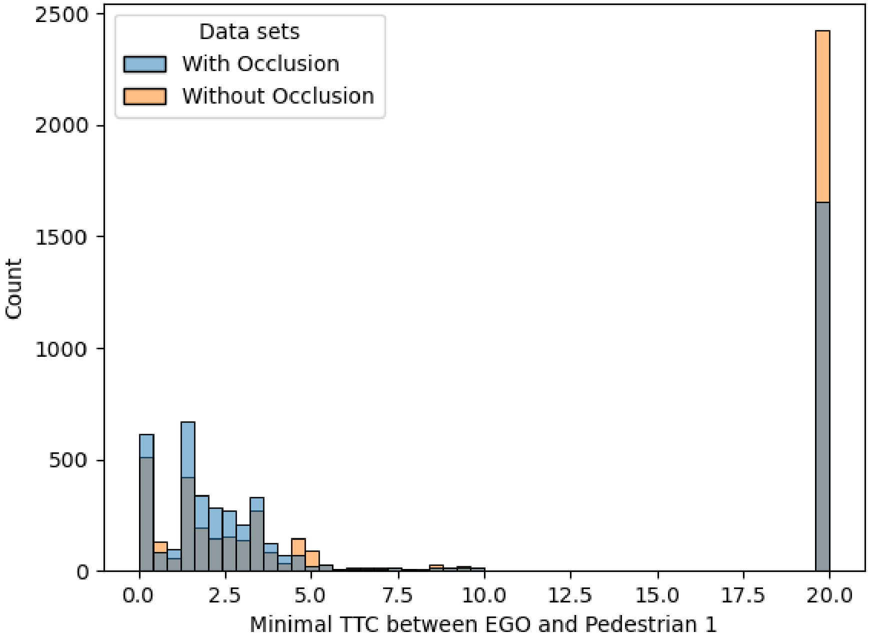
Execution and Evaluation

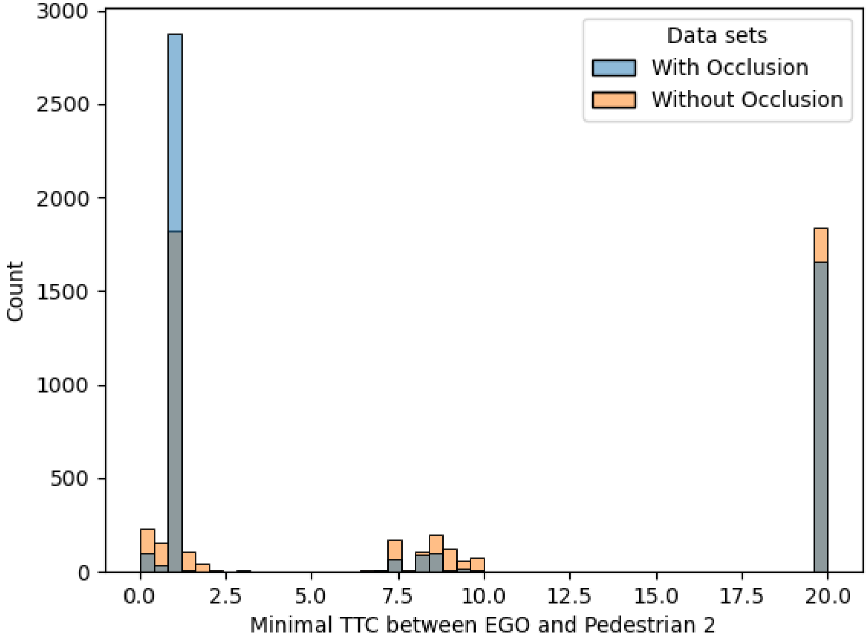
Comparison of Simulation Runs



Execution and Evaluation

Evaluation Report – Time-to-Collision (TTC)

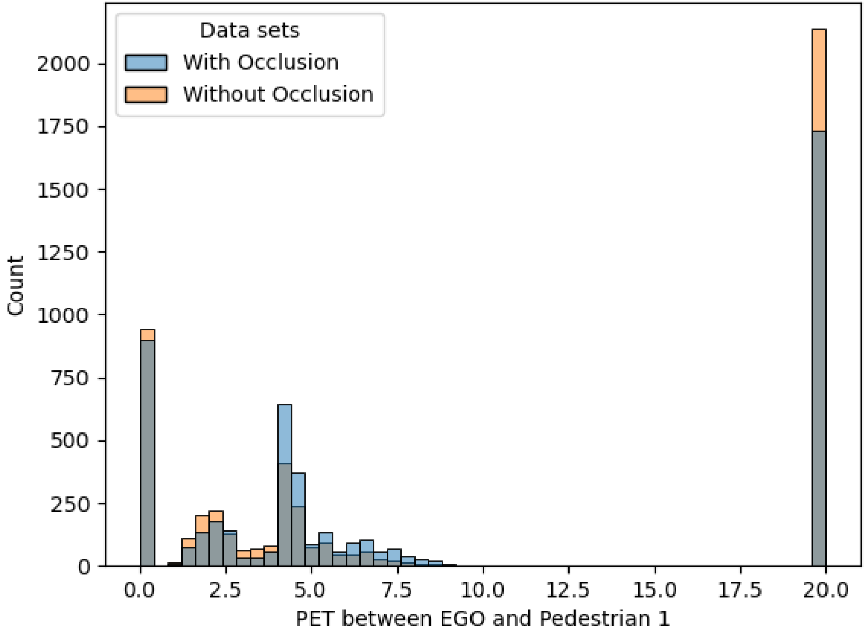


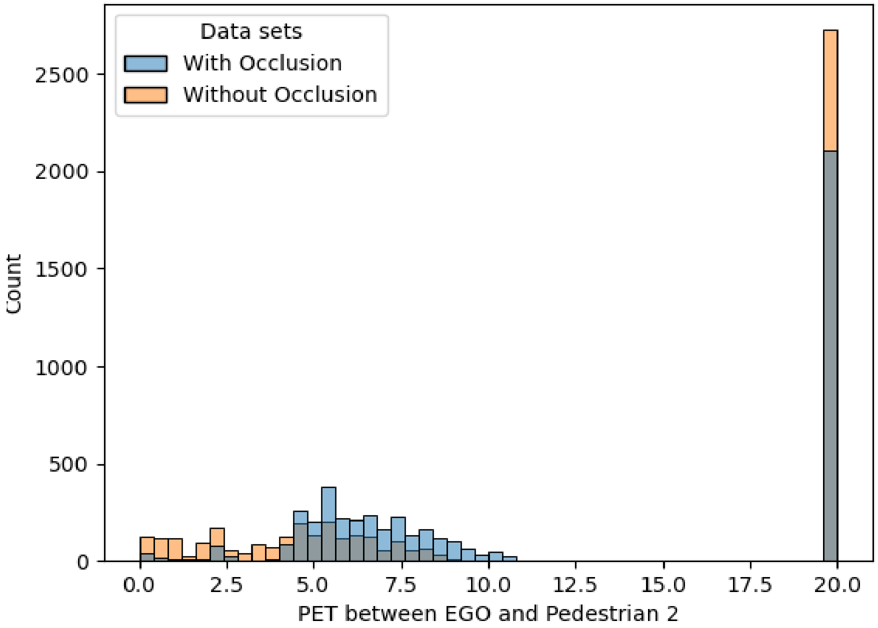


- TTC calculation based on current state and predicted further movement of HADF and pedestrians
- More runs with low TTCs in simulations with occlusion
- Unfiltered TTCs show also side effects from agent behavior

Execution and Evaluation

Evaluation Report – Post-Encroachment-Time (PET)

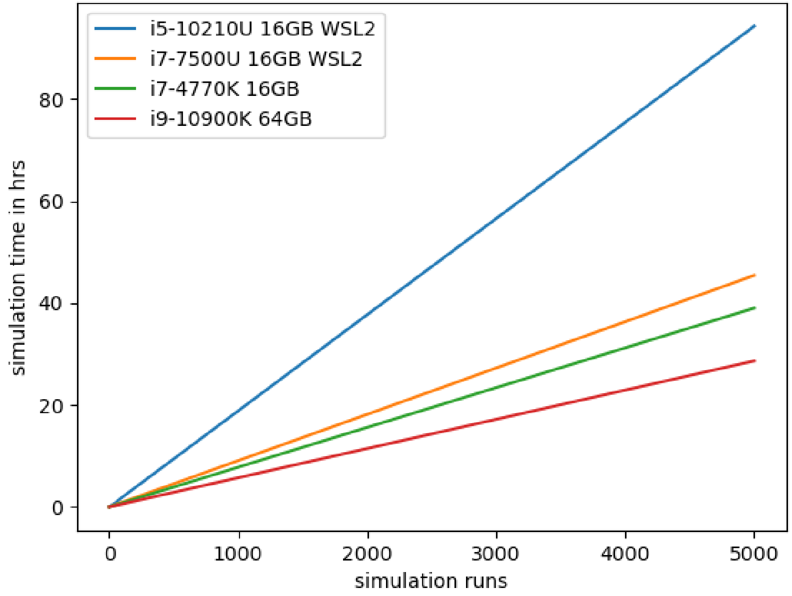


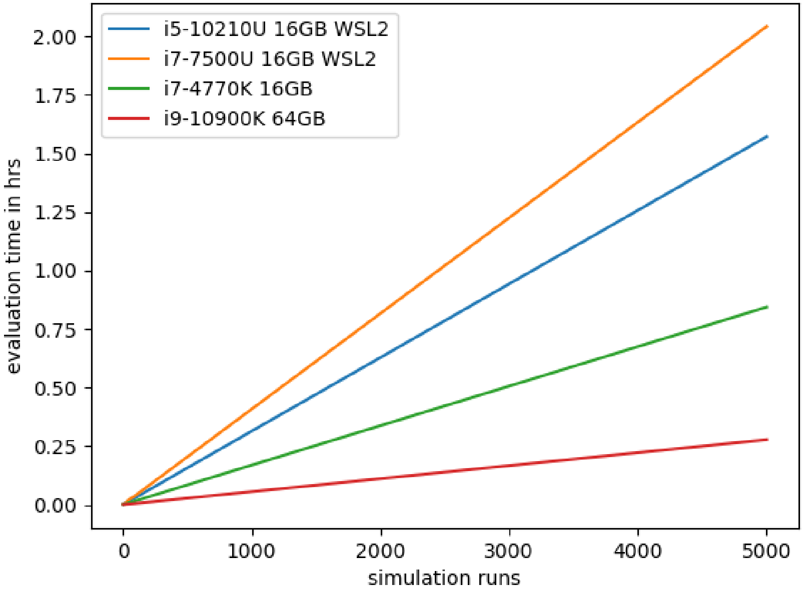


- Deadlock due to defensive driving of HADF in approx. 34% of simulations with occlusion and 42% of simulations without occlusion
- Some of the simulation runs with a PET of zero are also related to deadlocks due to the a posteriori collision detection
- Strong braking maneuvers by HADF in close situations lead to a longer time delay until the HADF accelerates again. Lower PET does not directly correspond to increased criticality.

Execution and Evaluation

Performance





- Scenario duration: 40 seconds
- Duration for generation of the concrete scenarios is negligible
- Debugging outputs of simulation models may have a negative impact on the performance (under investigation)
- Toolchain runs stable over large numbers of simulations also in conjunction with the Docker container

Summary and Outlook

Applied Standards and Extensions

Description of the scenario

OpenSCENARIO 1.0

Models are wrapped as FMUs

FMI 2.0

OpenDRIVE 1.6

Description of the map

OSI 3.2.0

All simulation models communicate through OSI messages

SSP 1.0

Specification of agent composition by multiple FMUs using SSP

Summary of standard extensions

- Logical scenario description through definition of parameter distributions
  - Based on results from Pegasus project
  - Stochastic and deterministic distributions
  - Definition of limits (truncated distributions)
  - ➢ Input for OpenSCENARIO Standardization
- New and extended OSI messages:
  - osi::TrafficCommand
  - sl45::MotionCommand
  - sl45::DynamicsRequest
  - sl45::VehicleCommunicationData
  - ➢ Input for OpenSimulationInterface Standardization
- Transfer of static environment through osi::GroundTruth during initialization phase

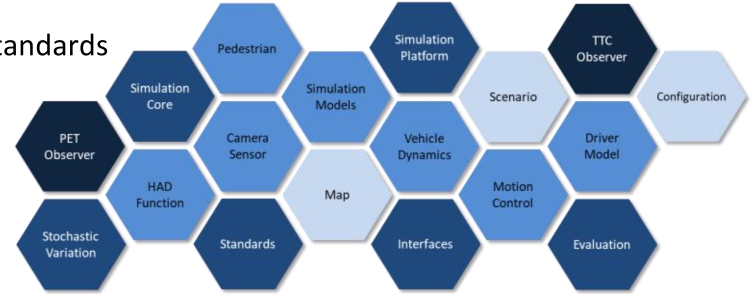
FMI = Functional Mockup Interface, OSI = Open Simulation Interface, SSP = System Structure and Parameterization

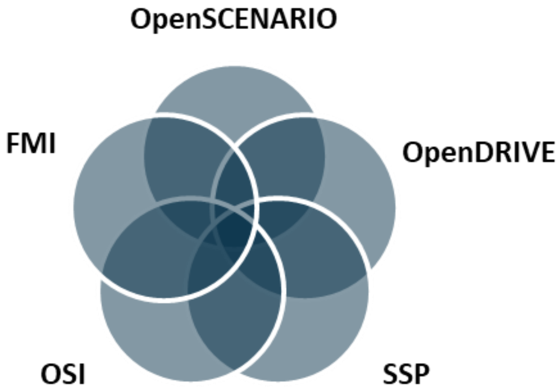
Summary and Outlook

SET Level

Summary

- Execution of an analysis task based on artifacts from the SET Level project
- Development of simulation models and simulation platform tooling
- Demonstration of modularity and integratability through utilization of standards
- Extensions of standards as input for standardization activities
- Implementation and demonstration of parameter variation mechanism
- Fully automated execution of the simulation for the analysis task





Outlook

- Further alignment and extension of technologies and standards (e.g. SSP)
- Improvement of agent model behavior and interaction
- Increasing complexity in maps and scenarios
- Extension of analysis task and corresponding evaluation (e.g. KPIs)

Thank you!