

# Scenario based virtual validation framework for autonomous driving functions

Requirements and further development for OpenSCENARIO

**Nager Markus,  
Leitner Andrea,  
Mungan Eren**

# Outline

- Problem & Motivation
- AVL's "Scenario based virtual validation framework for autonomous driving functions"
- Requirements for OpenSCENARIO

# Problem & Motivation

1. Uncountable potential situations (varying traffic and weather conditions, driver behavior, etc.)
2. Not feasible with real-world tests (country-specifics, etc.)

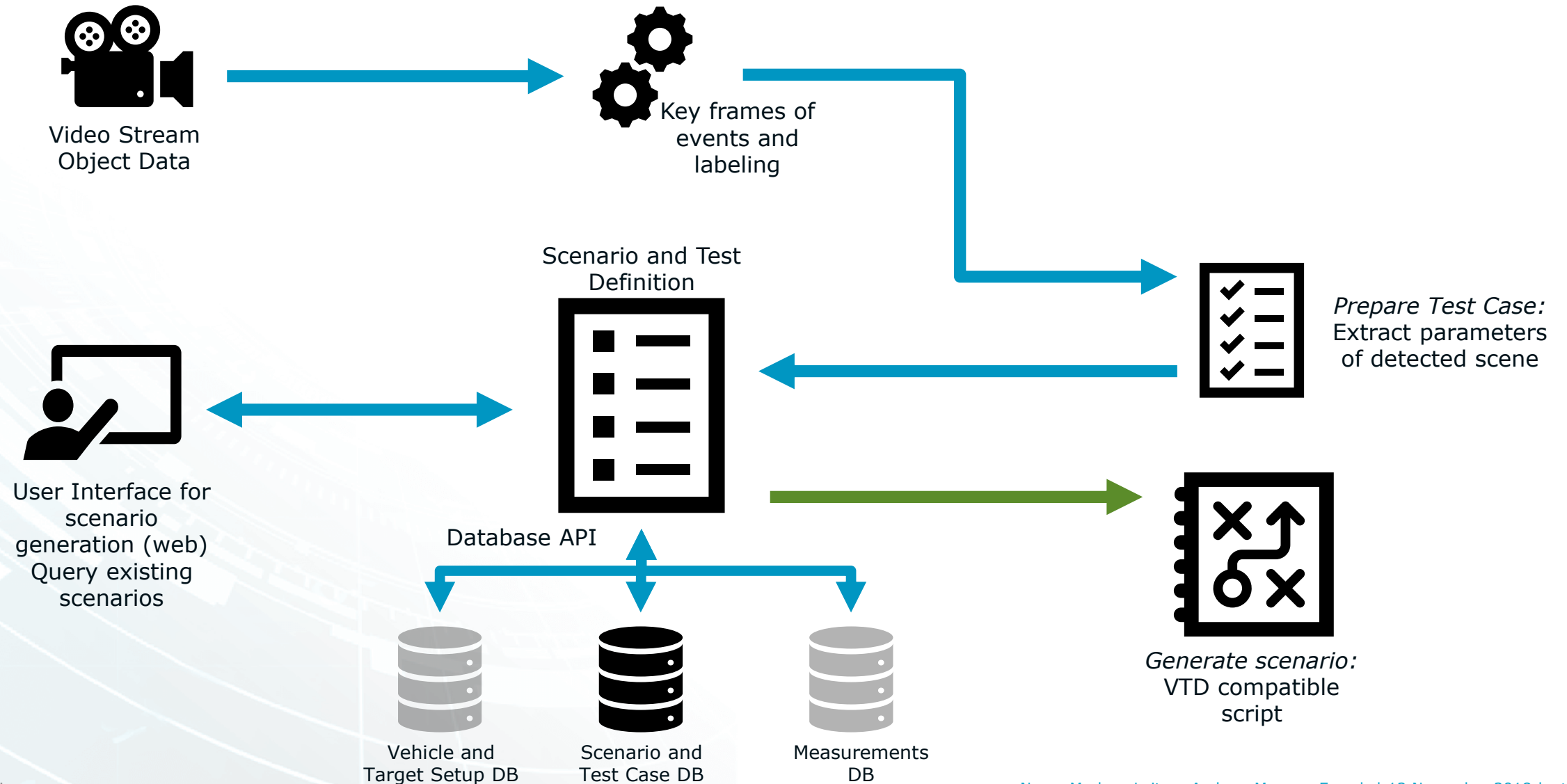
## => **Virtual Validation**

- Extraction & creation of Scenarios
- Storage of Scenarios
- Parametrization & execution of Scenarios

## ***"Parametrized End-to-End Scenario Generation Architecture for Autonomous Vehicles"***

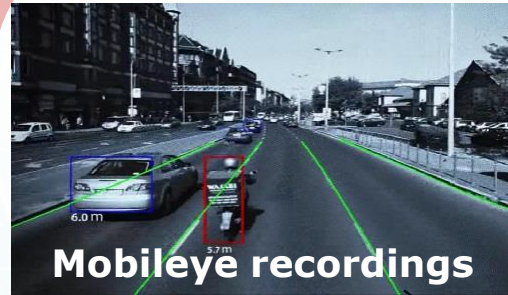
- Higher scenario coverage at lower costs and in shorter time

# High Level System Overview





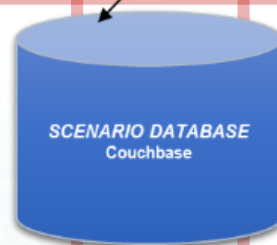
## End-to-End Process



Scenario generation

Query interface  
Couchbase – Test  
Sequence Generator

OSI compliant data



2

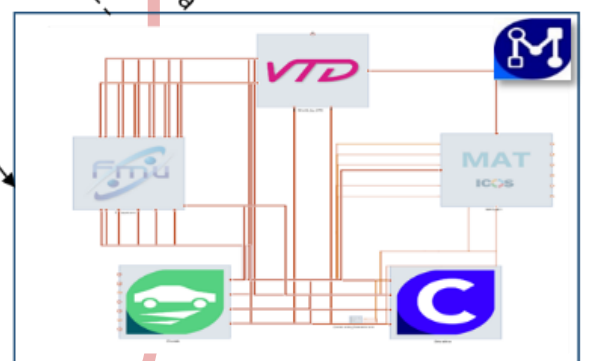


UI (especially query and filter capabilities) have to be defined as a joint effort.

Open Scenario

Parameter values

Simulation results will be defined as a next step



3

1

# Scenario Extraction

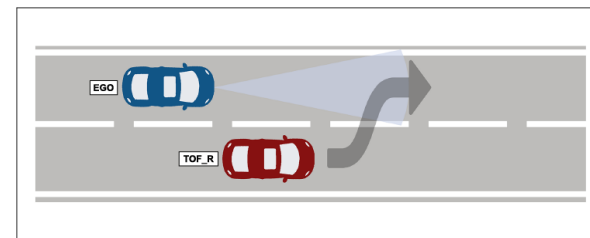
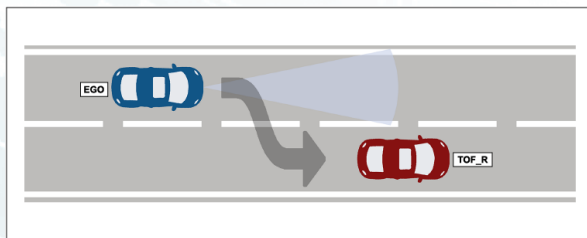
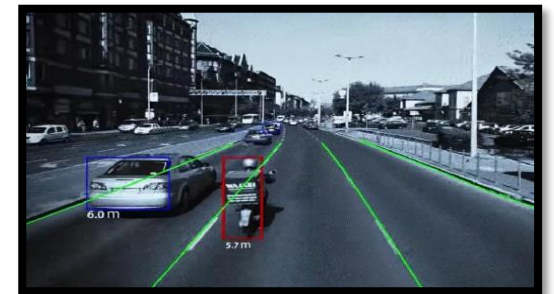
1

## Real-world sensory data:

- Sensor fusion (i.e. MobileEye camera, radar, lidar, etc.)
- Object level data to the standardized data format (Open Simulation Interface - OSI)

## Analyzing the video stream:

- Detect the timestamps (begin and end) of an event
- Match the event to a correct label (i.e. video segment between t1-t2 corresponds to TOF Cut-in maneuver”).
- Labeling with a rule-based approach (further enhanced by machine learning)



# Scenario Database

2

- Generic database (No-SQL)
- Standardized scenario definition
- Combination with external databases such as accident database, etc.

## Example of a lane change maneuver:

- The **states** of the vehicles are extracted **before** the lane change happened and another state **after** the lane change **maneuver** ended.
- State definitions are used to create a new test case, where parameters of the test case are filled according to the **subtraction** of those states.
  - If  $\text{SpeedDiff} = \text{TOF\_STATE\_SPEED\_TIME2} - \text{TOF\_STATE\_SPEED\_TIME1} \rightarrow \text{Speed change action}(\text{SpeedDiff})(\text{with additional parameters})$
  - If  $\text{LaneDiff} = \text{TOF\_STATE\_LANE\_TIME2} - \text{TOF\_STATE\_LANE\_TIME1} \rightarrow \text{Lane change action}(\text{LaneDiff})(\text{with additional parameters})$
- Moreover, a scenario that is created from the real-world data (e.g. ego-lane change) can be parametrized so that it is tested with varying conditions (i.e., the same extracted test-case **can be tested with different weather** conditions, **road** segments, different vehicle **speeds**, with/without trailer...)

# Scenario Parametrization

3

- **Standardized vendor-free scenario format** (OpenSCENARIO). Regardless of its origin (either from a real-world data or syntactic scenario definitions).
- **The output** of this module are **ready-to-execute** simulator files which can be run consecutively. Vehicle states are used to create vehicle actions.
- User can choose to **create test cases** that are relevant for an Adaptive Cruise Control. Our schema will query the relevant scenarios that are used in ACC tests and parametrize these scenarios to create test-cases (either **full-factorial** or with any methods of “Designs of Experiments”).
- In the end, the **generated data** will represent different scenarios with controlled variations which can be used as the **training** data for **machine learning** purposes.



# Scenario Parametrization

3

Configure ACC

Test Parameters

Test Name:  Id:

Number of Vehicles:  Number of Lanes:

Vehicle Actions:

	Order	ID	Lane	Position
▶	0	EGO	1	30
*				

EGO Target Speed:  -  km/h Acceleration:  -  Stepsize:

TOF Target Speed:  -  km/h Pitch:  -  Stepsize:

Trailer: ☐

Driver Parameters

VehicleActionsDialog

ID:  e.g. TOF1

Action Order:

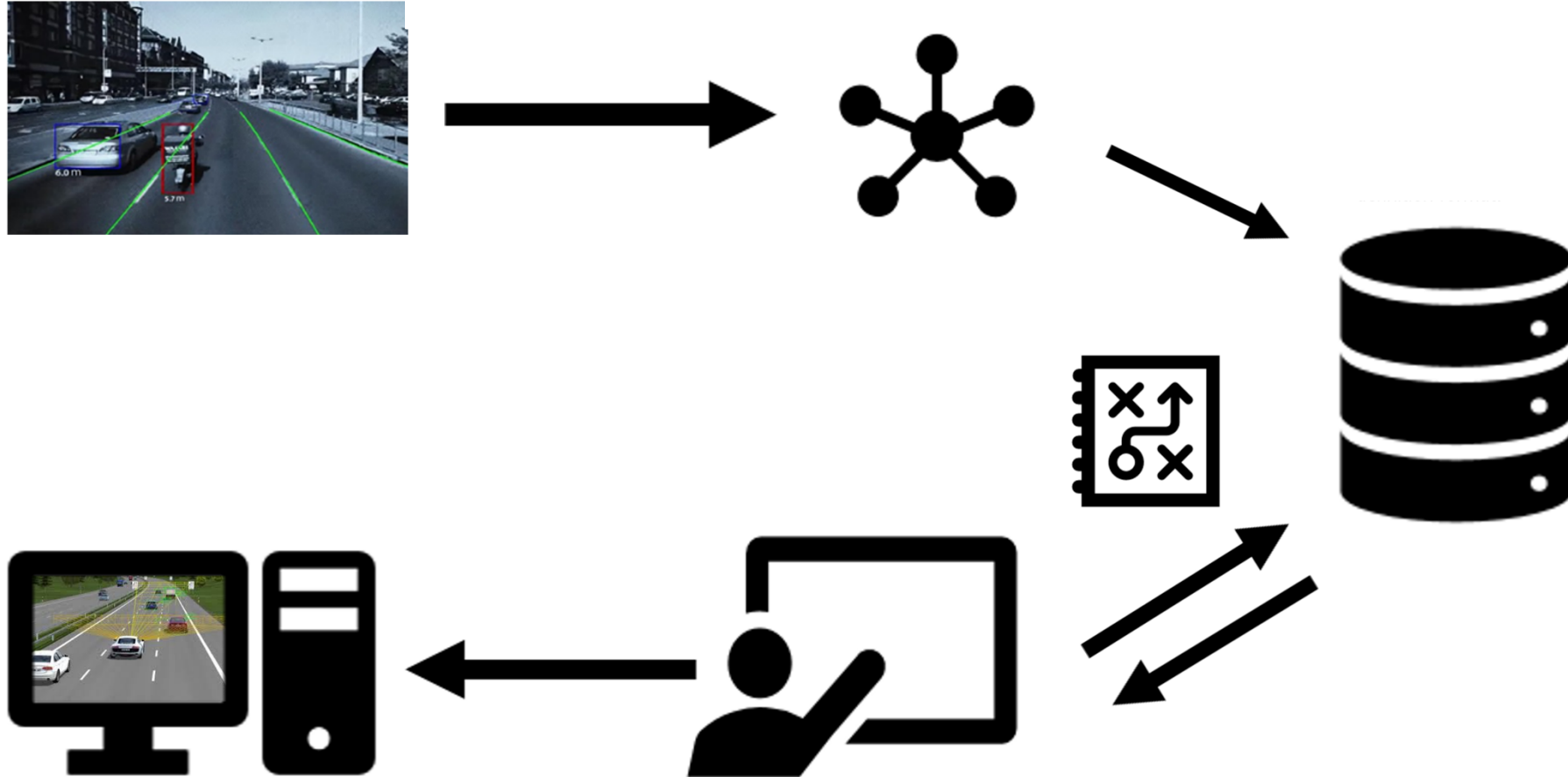
Lane Information:

Position Information:

## Example:

- Parametrization of scenarios to get **concrete** test cases that can then be executed in simulation.
- Definition of initial values and end values for **variable parameters** with respect to step size.
- **Configuration steps:**
  - Validation coverage
  - Autonomous function
  - Data Source
  - Test Environment
  - Environment conditions

# Work Flow



# Requirements for OpenSCENARIO

## Actions:

- Define lateral speed directly (only sinusoidal shaped lane change supported) same way as longitudinal
- Disable lateral control (vehicle tends to stay in the middle of the lane)

## Objects:

- Trailer should be able to be attached to vehicle

## SCP Messages:

- SCP messages should be embedded in the OpenSCENARIO file

## VTD:

- Lane coordinates for pedestrians (not only world coordinate system)
- Lane change shapes (also for linear, cubic, etc.)
- Pedestrian movements/character actions

### Private Action

#### Longitudinal

Distance

Speed

#### Lateral

Lane Change

Lane Offset

Distance

# Ideas for further development

## **Documentation:**

- Updated sources
- Examples

## **OpenSCENARIO file integrations:**

- Pedestrian movements
- Trailers (not attached in VTD)
- Camera angles
- Disabling driver properties (Lat. Control etc.)

## **Actor model:**

- More generic to support multiple simulation environments (reusability)

## **OpenDRIVE:**

- More intuitive integration to OpenSCENARIO file (not just linked)



# Thank You

**AVL**



markus.nager@avl.com  
andrea.leitner@avl.com

[www.avl.com](http://www.avl.com)

