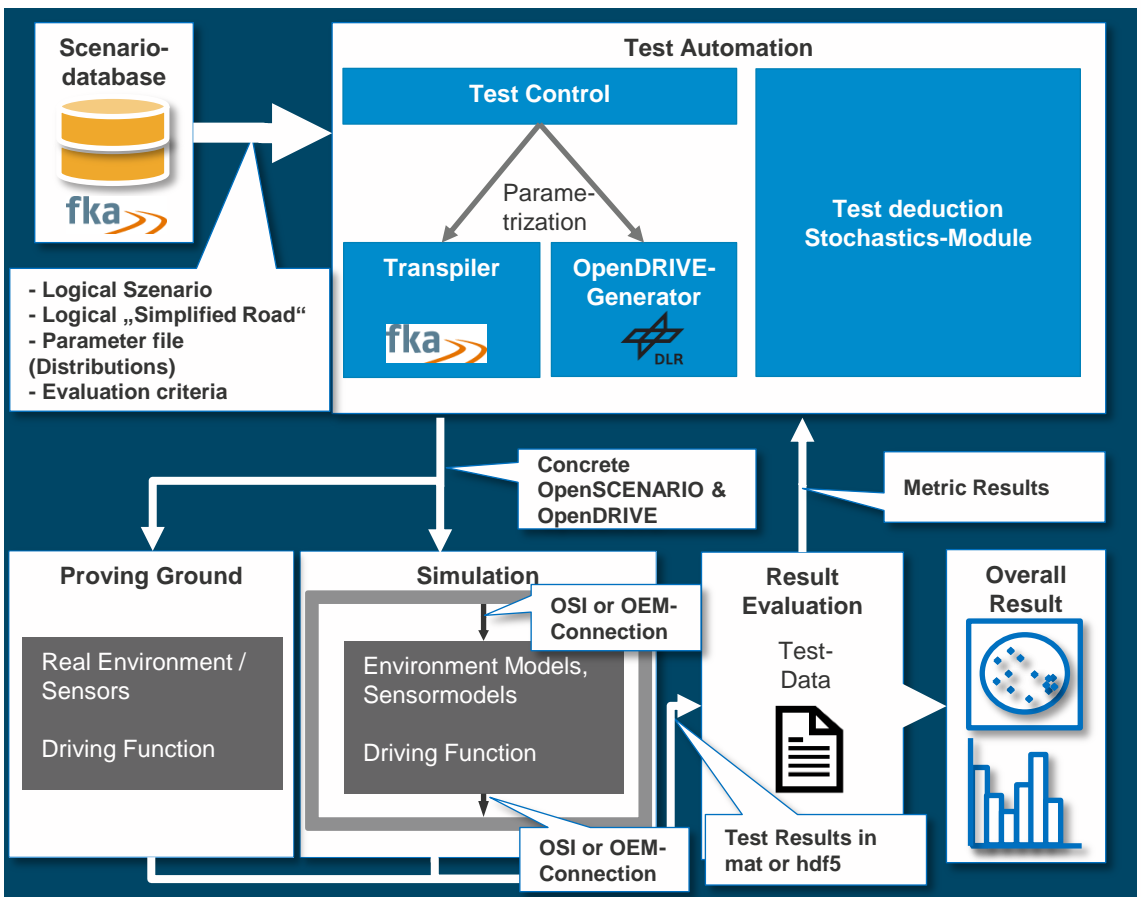


TEST EXECUTION INTERFACES



Interfaces - Overview

- ➔ Standards were defined in each execution interface (Scenario-database, Test Automation, Simulation) as well as in each execution modul
- ➔ Make use of standards to facilitate and simplify data transfer between tools, models, etc.
- ➔ Interchangeability of test cases between test instances, e.g. between simulation and proving ground



Supported by:



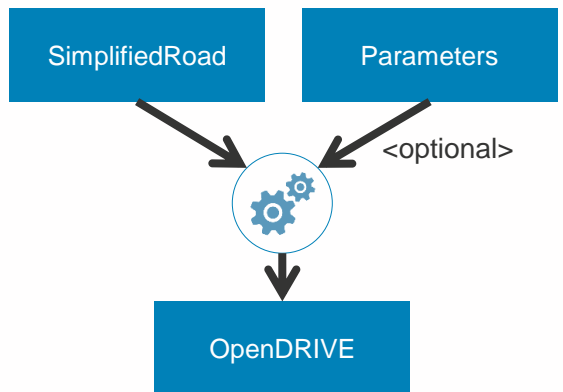
on the basis of a decision by the German Bundestag

TEST EXECUTION INTERFACES



OpenDRIVE Generator

- **OpenDRIVE Generation**
- **SimplifiedRoad** is a domain specific language for simplified modeling of road sections for test cases.
- **Parameters** can be applied to SimplifiedRoad models to allow variations.
- **OpenDRIVE** is the open, vendor-independent mature XML-based file format, which contains all features to model real road networks and is supported by commonly used simulation software.



- **SimplifiedRoad**
- Format based on XML with schema
- Based on naming of OpenDRIVE
- Variables for parametrization
- Several simplifications with focus on road sections, e.g.
 - Automated positioning of sections
 - Allow standardized cross sections from guidelines
 - Composition and positioning of common infrastructure: guide posts, guard rails, noise barriers

```

<section length="1000">
  <road crossSection="RQ36" mirror="true"/>
</section>

<section length="1000">
  <road crossSection="RQ36" mirror="true"/>
  <course>
    <arc radius="250"/>
  </course>
</section>
  
```

These lines result in 450 lines OpenDRIVE

Major Benefits

- Re-use of standardized elements
- Improved productivity (less and simpler model code)
- Improved maintainability of roads and their variations
- Simplified and batched prototyping and testing process, especially useful in combination with OpenSCENARIO



Supported by:



on the basis of a decision by the German Bundestag

TEST EXECUTION INTERFACES



Transpiler

Motivation

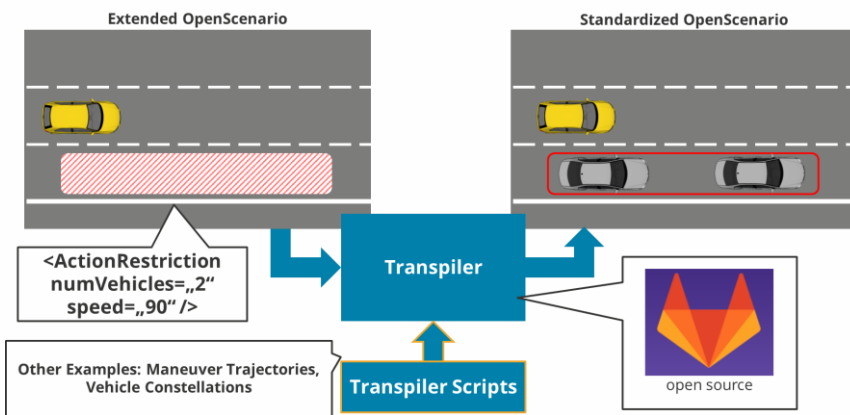
- Main output of the **PEGASUS Scenario-Database** are logical scenarios and parameter frequency distributions
- **Standard OpenSCENARIO** does not allow to define and parameterize all elements necessary for safety validation, e.g.
 - **Complex maneuver trajectories** like lane changes
 - **High-level scenario elements** like action restrictions (see below)



- **Logical Scenario**
- Logical „Simplified Road“
- **Parameter Distribution**
- Evaluation Criteria

Transpiler as a Solution

- OpenSCENARIO can be easily **extended by custom elements**
- Custom elements can describe complex scenario elements and be **parameterizable**, e.g.
`<ComplexLaneChange duration="2.0" aggressivity="0.8" />`
- **Compatibility** is ensured through translation into standardized OpenSCENARIO **by the transpiler**



Main Advantages

- **Agile** creation and usage of OpenSCENARIO extensions
- **Simple variation of complex elements** for testing
- Keeping **full compatibility** with tools using standardized OpenSCENARIO
- Extensions are easily **shareable via python scripts**
- Increased **readability and intuitiveness** of scenario files
- Transpiler platform is **open-source**



Supported by:



on the basis of a decision by the German Bundestag