# OPENPASS
# ARCHITECTURE COMMITTEE MEETING

14.12.2018

Freude am Fahren

# AGENDA

1. Goals of openPASS
2. Example Case AEB
3. Redesign Input Files
4. openScenario in openPASS
5. Modules and components
6. Documentation
7. Next Steps and Timeline
8. License
9. Handling of different Use-Cases
10. General Points for further development (Coding Rules, Documentation, …) (ITK)
11. Hierarchical SystemEditor, Documentation GUI, ComponentGenerator (VW)

openPASS

# GOALS OF OPENPASS

Freude am Fahren

# GOALS OF OPENPASS

Standardized effectiveness assessment platform for driver assistance systems and autonomous driving

Usages of openPASS:

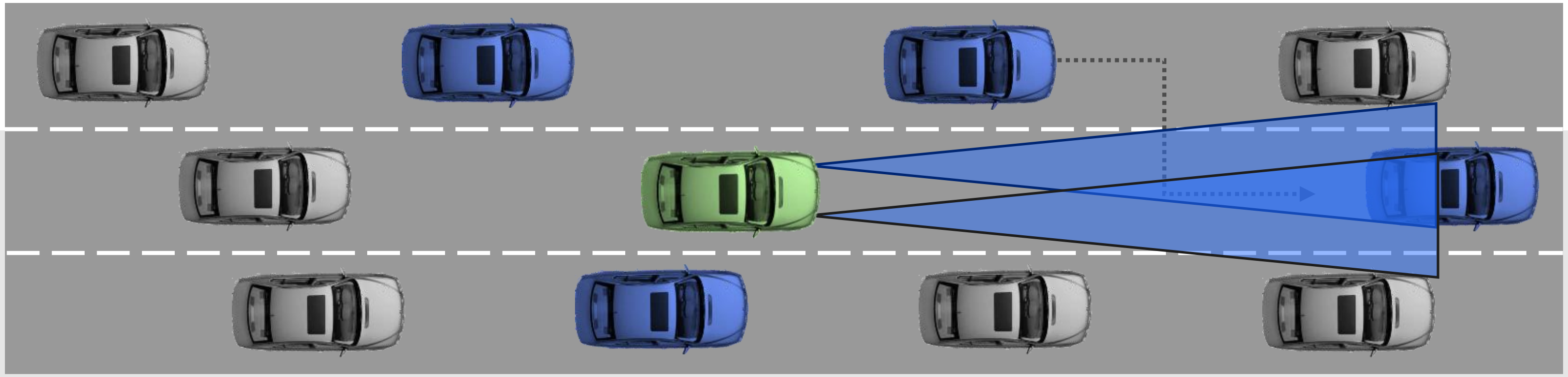
- Scenario-simulation with stochastic variation (e.g. NCAP)
- Closed loop traffic simulation (virtual FOT)
- Crash re-simulation (e.g. PCM)

openPASS

# EXAMPLE CASE AEB

Freude am Fahren

# EXAMPLE CASE AEB



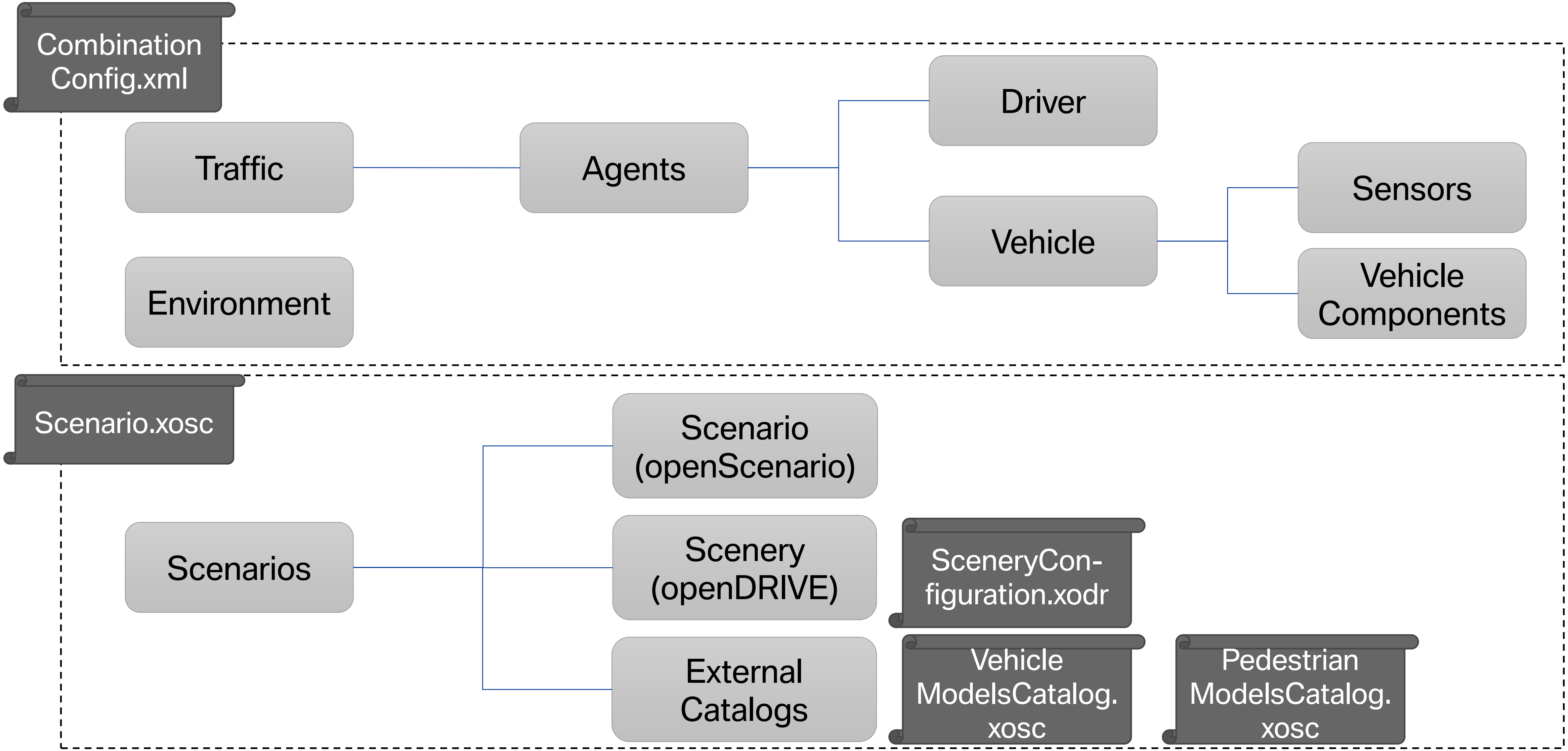
- ✓ Given is a road with 3 lanes
  Scope: OpenDRIVE import, OSI ground truth generation
- ✓ An ego car is placed at the center lane, surrounded by 4 scenario cars.
  The rest of the scenery is filled with common cars.
  Scope: OpenSCENARIO import, SpawnLogic, OSI world representation
- ✓ At a given time, one of the scenario cars performs a sudden lane change, following a predefined trajectory.
  Scope: TimeTriggerEventDetector, Manipulator, TrajectoryFollower
- ✓ Several cars are equipped with a simplified automatic emergency brake (AEB), which takes the input of two geometric sensors. TBD! Exemplarisch nur Ego
  Scope: Stochastic Distribution of Modules, AssistantSystems, GeometricSensors
- ✓ Each AEB automatically merges the information from its sensors and starts braking in case of emergency.
  Scope: AssistantSystem, SensorFusion, SignalPrioritizer
- ✓ The simulation is repeated several times with stochastic variation to see the variability of the AEB's intervention.
  Scope: Stochastic Placement of Agents, Output

openPASS

# REDESIGN OF INPUT FILES

Freude am Fahren

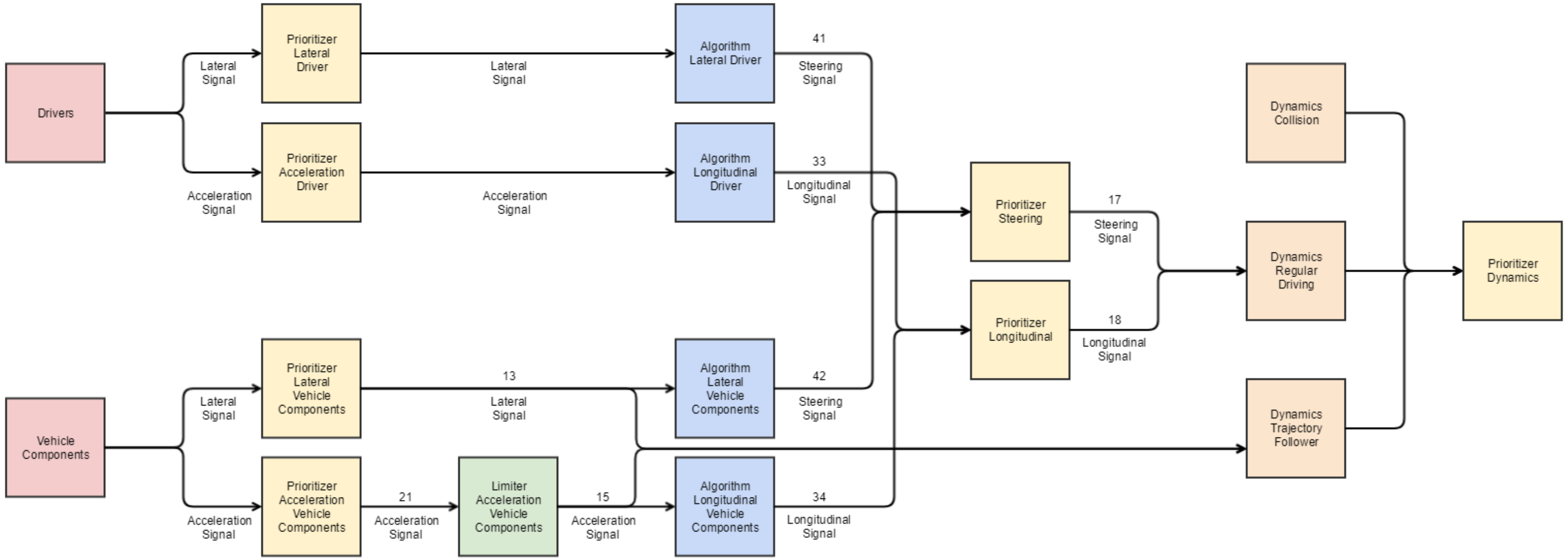
# EXPERIMENT SETUP
# COMBINATION AND SCENARIO CONFIGURATION



# SIMULATION CORE CONFIGURATION APPCONFIG.XML

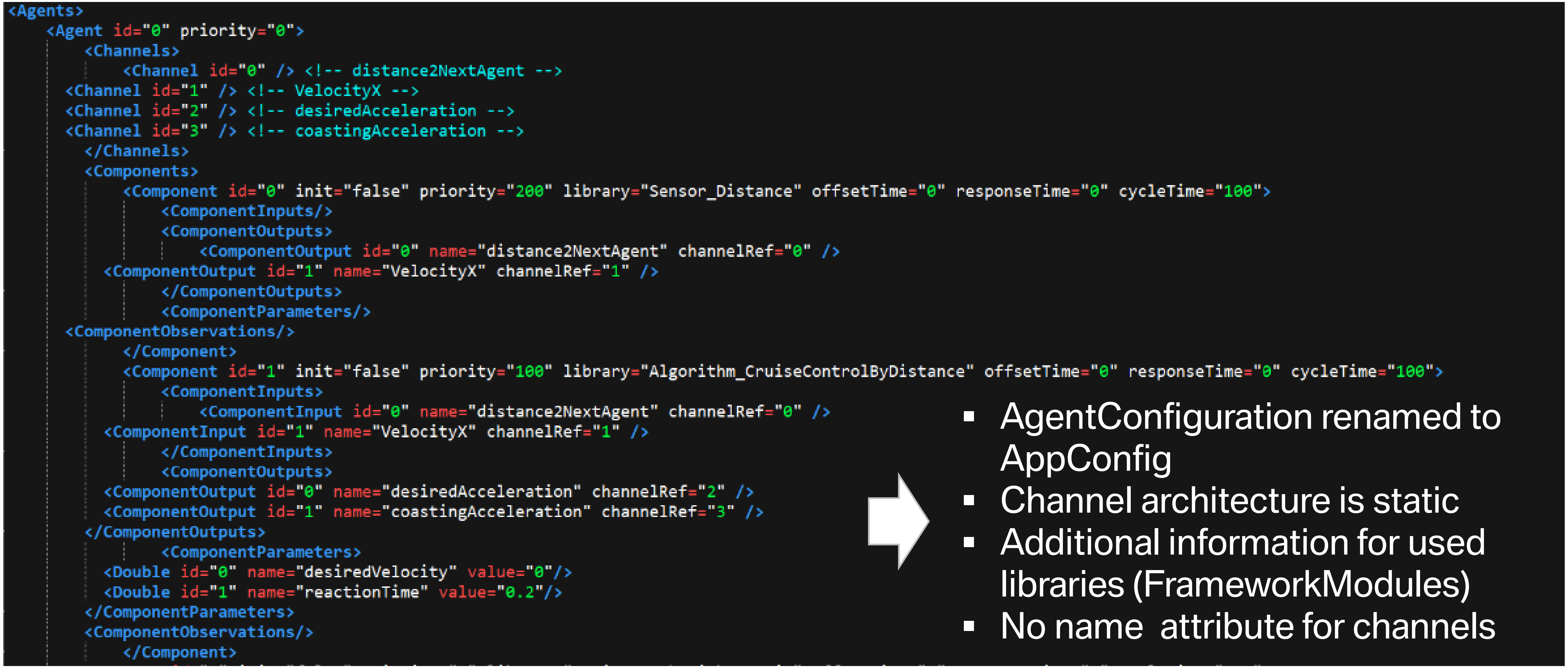
AppConfig.xml

static module and channel architecture for all agents



# REDESIGN AGENTCONFIGURATION

NewAppConfig.xml

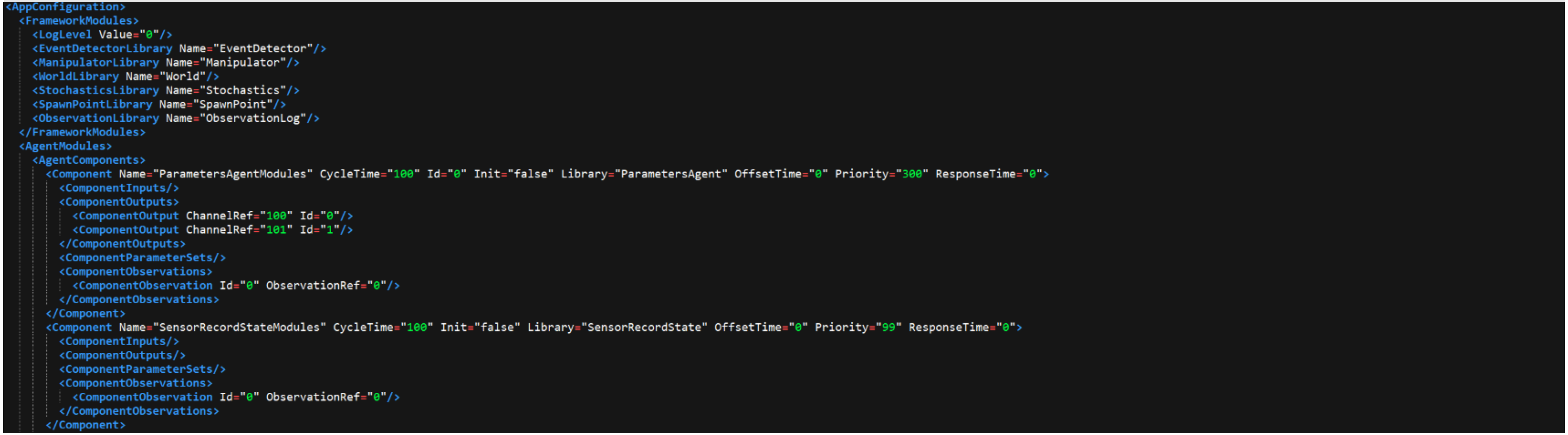


```
<Agents>
    <Agent id="0" priority="0">
        <Channels>
            <Channel id="0" /> <!-- distance2NextAgent -->
      <Channel id="1" /> <!-- VelocityX -->
      <Channel id="2" /> <!-- desiredAcceleration -->
      <Channel id="3" /> <!-- coastingAcceleration -->
        </Channels>
        <Components>
            <Component id="0" init="false" priority="200" library="Sensor_Distance" offsetTime="0" responseTime="0" cycleTime="100">
                <ComponentInputs/>
                <ComponentOutputs>
                    <ComponentOutput id="0" name="distance2NextAgent" channelRef="0" />
          <ComponentOutput id="1" name="VelocityX" channelRef="1" />
                </ComponentOutputs>
                <ComponentParameters/>
      <ComponentObservations/>
            </Component>
            <Component id="1" init="false" priority="100" library="Algorithm_CruiseControlByDistance" offsetTime="0" responseTime="0" cycleTime="100">
                <ComponentInputs>
                    <ComponentInput id="0" name="distance2NextAgent" channelRef="0" />
          <ComponentInput id="1" name="VelocityX" channelRef="1" />
                </ComponentInputs>
                <ComponentOutputs>
          <ComponentOutput id="0" name="desiredAcceleration" channelRef="2" />
          <ComponentOutput id="1" name="coastingAcceleration" channelRef="3" />
        </ComponentOutputs>
                <ComponentParameters>
          <Double id="0" name="desiredVelocity" value="0"/>
          <Double id="1" name="reactionTime" value="0.2"/>
        </ComponentParameters>
        <ComponentObservations/>
            </Component>
```

- AgentConfiguration renamed to AppConfig
- Channel architecture is static
- Additional information for used libraries (FrameworkModules)
- No name  attribute for channels

# REDESIGN AGENTCONFIGURATION ADVANTAGES

- ✓ Better readability in case of complex channel structure
- ✓ Slave configuration with additional framework modules
- ✓ Static Components/channels for all agents (dynamically instantiated)

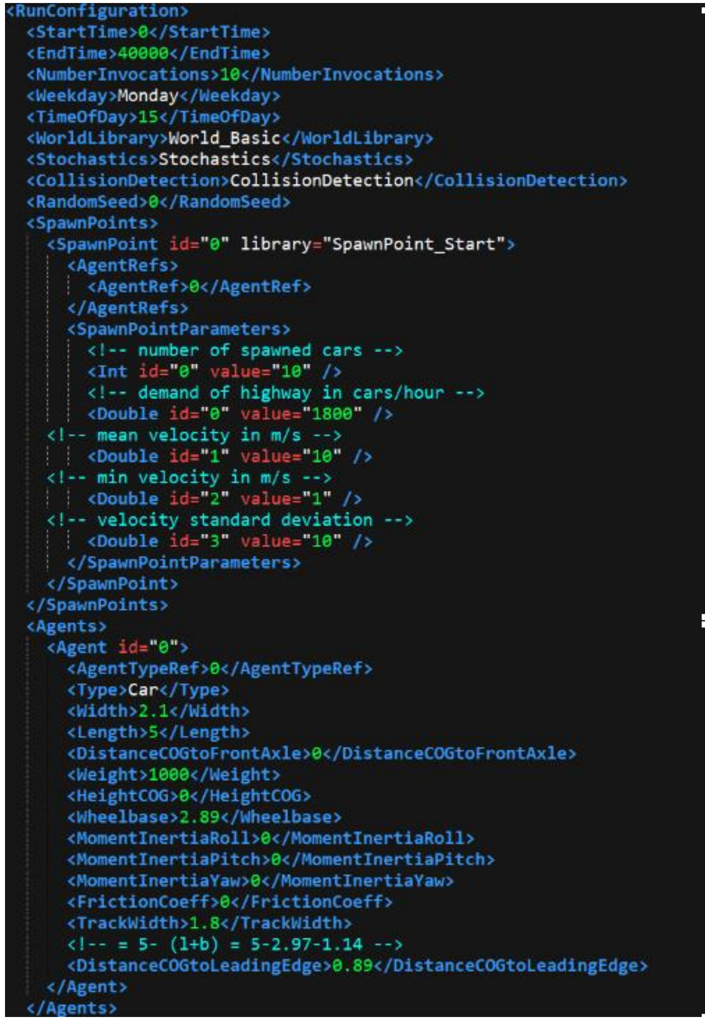


```
<AppConfiguration>
  <FrameworkModules>
    <LogLevel Value="0"/>
    <EventDetectorLibrary Name="EventDetector"/>
    <ManipulatorLibrary Name="Manipulator"/>
    <WorldLibrary Name="World"/>
    <StochasticsLibrary Name="Stochastics"/>
    <SpawnPointLibrary Name="SpawnPoint"/>
    <ObservationLibrary Name="ObservationLog"/>
  </FrameworkModules>
  <AgentModules>
    <AgentComponents>
      <Component Name="ParametersAgentModules" CycleTime="100" Id="0" Init="false" Library="ParametersAgent" OffsetTime="0" Priority="300" ResponseTime="0">
        <ComponentInputs/>
        <ComponentOutputs>
          <ComponentOutput ChannelRef="100" Id="0"/>
          <ComponentOutput ChannelRef="101" Id="1"/>
        </ComponentOutputs>
        <ComponentParameterSets/>
        <ComponentObservations>
          <ComponentObservation Id="0" ObservationRef="0"/>
        </ComponentObservations>
      </Component>
      <Component Name="SensorRecordStateModules" CycleTime="100" Init="false" Library="SensorRecordState" OffsetTime="0" Priority="99" ResponseTime="0">
        <ComponentInputs/>
        <ComponentOutputs/>
        <ComponentParameterSets/>
        <ComponentObservations>
          <ComponentObservation Id="0" ObservationRef="0"/>
        </ComponentObservations>
      </Component>
```

# REDESIGN RUNCONFIGURATION
# REPLACED BY COMBINATION CONFIG AND SCENARIO

Scenario.xml

CombinationConfig.xml

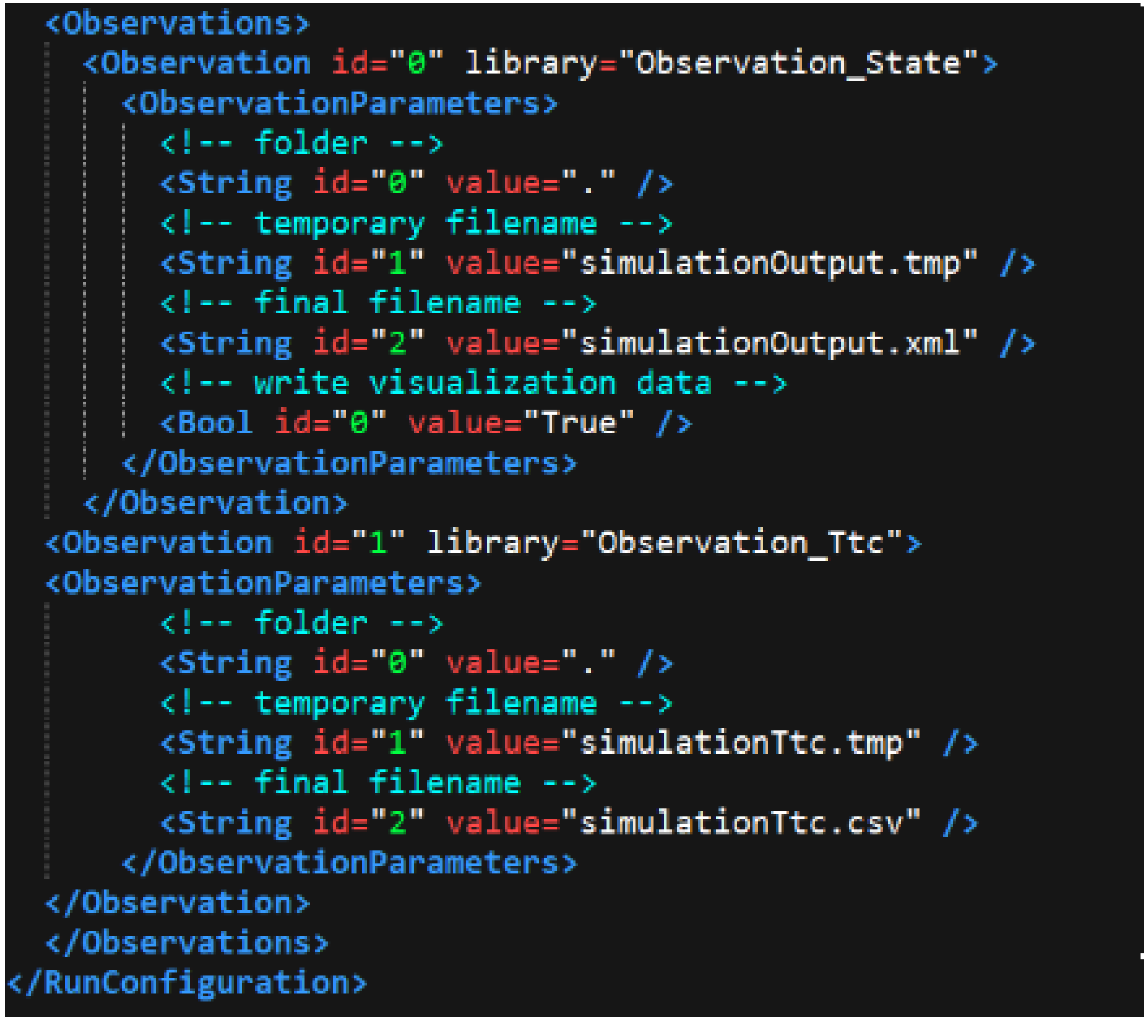


```xml
<RunConfiguration>
  <StartTime>0</StartTime>
  <EndTime>40000</EndTime>
  <NumberInvocations>10</NumberInvocations>
  <Weekday>Monday</Weekday>
  <TimeOfDay>15</TimeOfDay>
  <WorldLibrary>World_Basic</WorldLibrary>
  <Stochastics>Stochastics</Stochastics>
  <CollisionDetection>CollisionDetection</CollisionDetection>
  <RandomSeed>0</RandomSeed>
  <SpawnPoints>
    <SpawnPoint id="0" library="SpawnPoint_Start">
      <AgentRefs>
        <AgentRef>0</AgentRef>
      </AgentRefs>
      <SpawnPointParameters>
        <!-- number of spawned cars -->
        <Int id="0" value="10" />
        <!-- demand of highway in cars/hour -->
        <Double id="0" value="1800" />
    <!-- mean velocity in m/s -->
        <Double id="1" value="10" />
    <!-- min velocity in m/s -->
        <Double id="2" value="1" />
    <!-- velocity standard deviation -->
        <Double id="3" value="10" />
      </SpawnPointParameters>
    </SpawnPoint>
  </SpawnPoints>
  <Agents>
    <Agent id="0">
      <AgentTypeRef>0</AgentTypeRef>
      <Type>Car</Type>
      <Width>2.1</Width>
      <Length>5</Length>
      <DistanceCOGtoFrontAxle>0</DistanceCOGtoFrontAxle>
      <Weight>1000</Weight>
      <HeightCOG>0</HeightCOG>
      <Wheelbase>2.89</Wheelbase>
      <MomentInertiaRoll>0</MomentInertiaRoll>
      <MomentInertiaPitch>0</MomentInertiaPitch>
      <MomentInertiaYaw>0</MomentInertiaYaw>
      <FrictionCoeff>0</FrictionCoeff>
      <TrackWidth>1.8</TrackWidth>
      <!-- = 5- (l+b) = 5-2.97-1.14 -->
      <DistanceCOGtoLeadingEdge>0.89</DistanceCOGtoLeadingEdge>
    </Agent>
  </Agents>
```

Renamed and restructured in CombinationConfig.xml

Relocated to VehicleModelsCatalog.xosc

# REDESIGN RUNCONFIGURATION
# REPLACED BY COMBINATION CONFIG AND SCENARIO



```
  <Observations>
    <Observation id="0" library="Observation_State">
      <ObservationParameters>
        <!-- folder -->
        <String id="0" value="." />
        <!-- temporary filename -->
        <String id="1" value="simulationOutput.tmp" />
        <!-- final filename -->
        <String id="2" value="simulationOutput.xml" />
        <!-- write visualization data -->
        <Bool id="0" value="True" />
      </ObservationParameters>
    </Observation>
  <Observation id="1" library="Observation_Ttc">
  <ObservationParameters>
        <!-- folder -->
        <String id="0" value="." />
        <!-- temporary filename -->
        <String id="1" value="simulationTtc.tmp" />
        <!-- final filename -->
        <String id="2" value="simulationTtc.csv" />
      </ObservationParameters>
  </Observation>
  </Observations>
</RunConfiguration>
```

Reduced and relocated to CombinationConfig.xml

# REDESIGN RUNCONFIGURATION ADVANTAGES

- ✓ Usage of the openScenario standard
- ✓ Flexible experiment configuration using xml format
- ✓ Easy readability and editability of xml files
- ✓ Possibility to simply generate xml files automatically (e.g. GUI)

# REDESIGN RUNCONFIGURATION
# CONFIG-STRUKTUR

## old structure

Framework Configuration.xml

- Path and Logger Information

Run Configuration.xml

- Experiment Configuration
- Traffic Configuration (as SpawnPointParameters)
- Vehicle Model Parameters (as Agents)
- Observation Module Parametrization (OutputFile)

Scenery Configuration.xml

- Scenery in openDrive Format

Agent Configuration.xml

- Module Connections for each Agent
- Channels
- Modules with Channel References for Inputs and Outputs

## new structure

Combination Config.xml

Scenario.xosc

Vehicle ModelsCatalog.xosc

SceneryConfiguration.xodr

AppConfig.xml

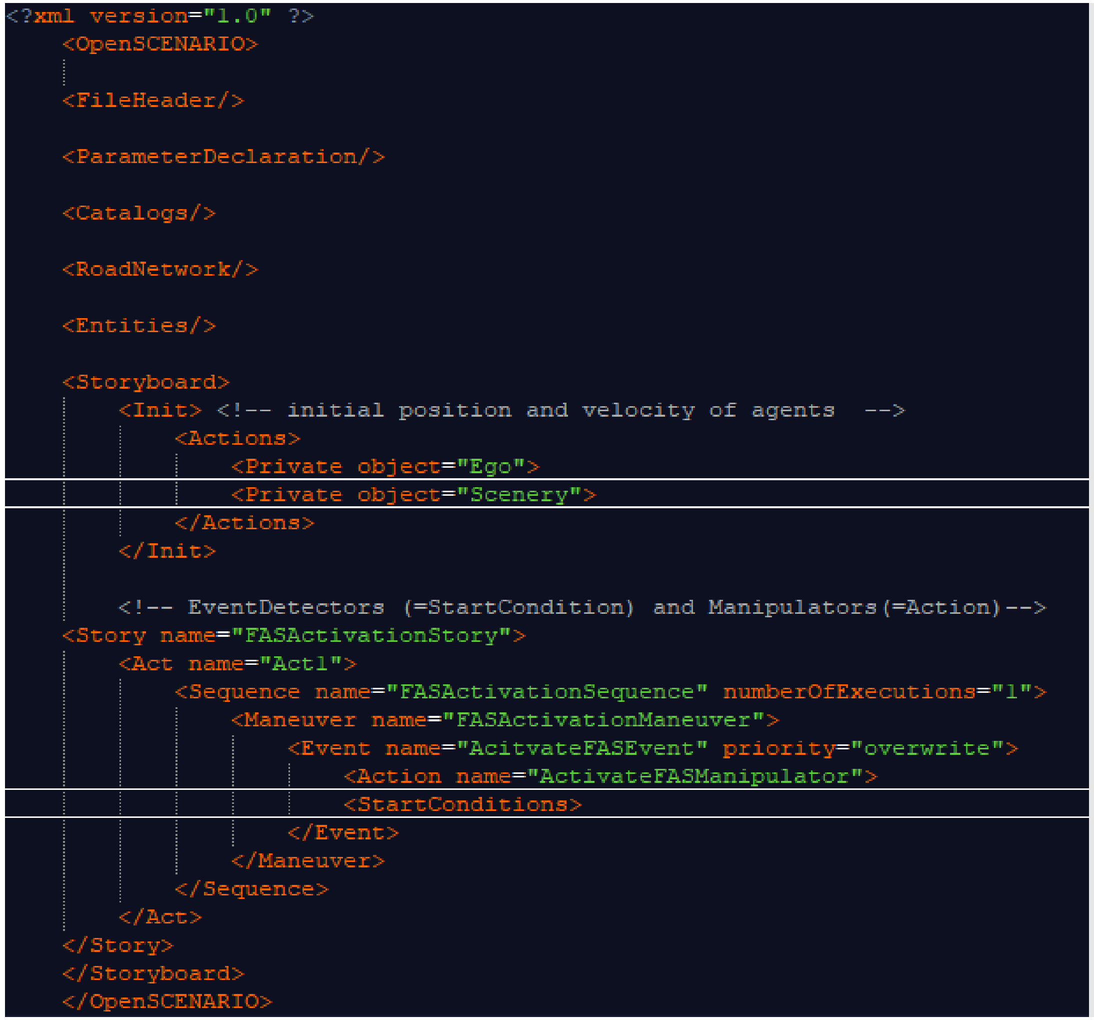
# REDESIGNED INPUT FILES SUMMARY

| Input Files | Changes |
|---|---|
| SceneryConfiguration.xodr | New file ending |
| AppConfig.xml | Combines information from frameworkConfiguration.xml and agentConfiguration.xml |
| CombinationConfig.xml | Replaces runConfiguration.xml<br>contains all information required for an experiment |
| Scenario.xocs | New file to specify a scenario (openScenario Standard) |
| VehicleModelsCatalog.xocs | New separate input file to specify vehicle parameters (from runConfiguration.xml) |
| Further Catalogs | Linked in Scenario.xosc (Pedestrians, etc.) |

# OPENSCENARIO IN OPENPASS

Freude am Fahren

# OPENSCENARIO IN OPENPASS
# OPENSCENARIO.XOSC



```
<?xml version="1.0" ?>
    <OpenSCENARIO>

    <FileHeader/>

    <ParameterDeclaration/>

    <Catalogs/>

    <RoadNetwork/>

    <Entities/>

    <Storyboard>
        <Init> <!-- initial position and velocity of agents  -->
            <Actions>
                <Private object="Ego">
                <Private object="Scenery">
            </Actions>
        </Init>

        <!-- EventDetectors (=StartCondition) and Manipulators(=Action)-->
    <Story name="FASActivationStory">
        <Act name="Act1">
            <Sequence name="FASActivationSequence" numberOfExecutions="1">
                <Maneuver name="FASActivationManeuver">
                    <Event name="AcitvateFASEvent" priority="overwrite">
                        <Action name="ActivateFASManipulator">
                        <StartConditions>
                    </Event>
                </Maneuver>
            </Sequence>
        </Act>
    </Story>
    </Storyboard>
    </OpenSCENARIO>
```

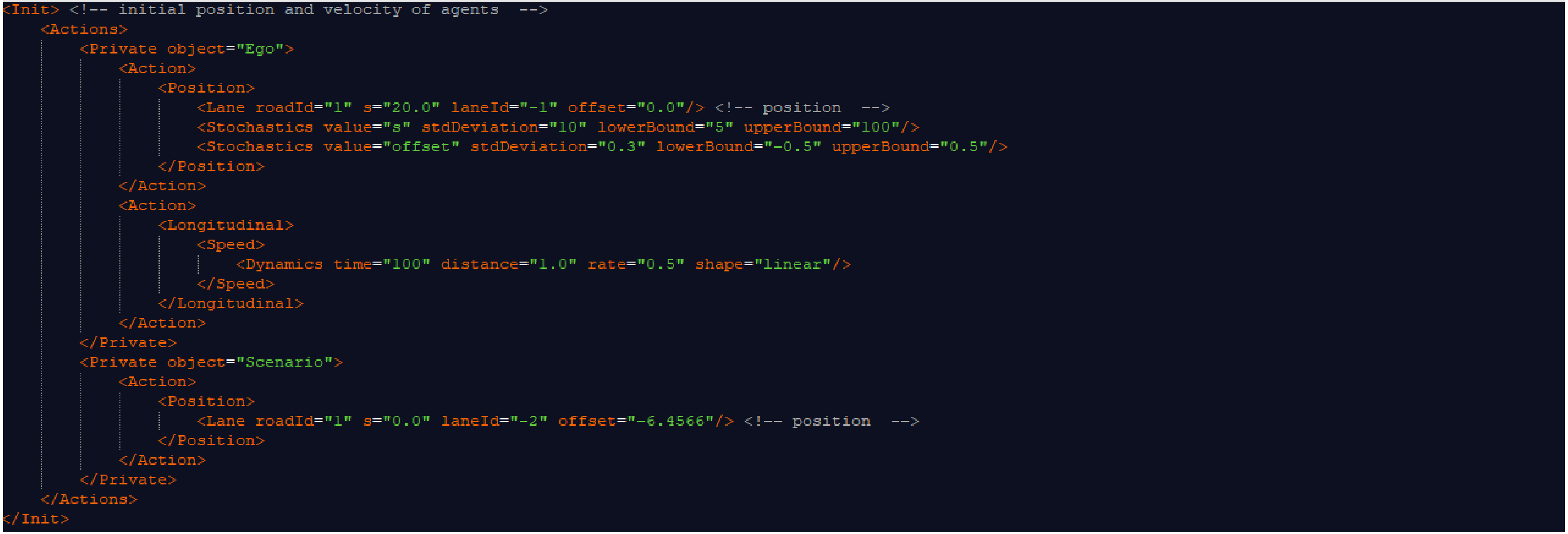
Link to Scenery.xodr and Catalogs (e.g. VehicleModelsCatalog.xosc)

Description of initial setup

Screenplay for the simulation

# OPENSCENARIO IN OPENPASS
# DESCRIPTION OF INITIAL SETUP

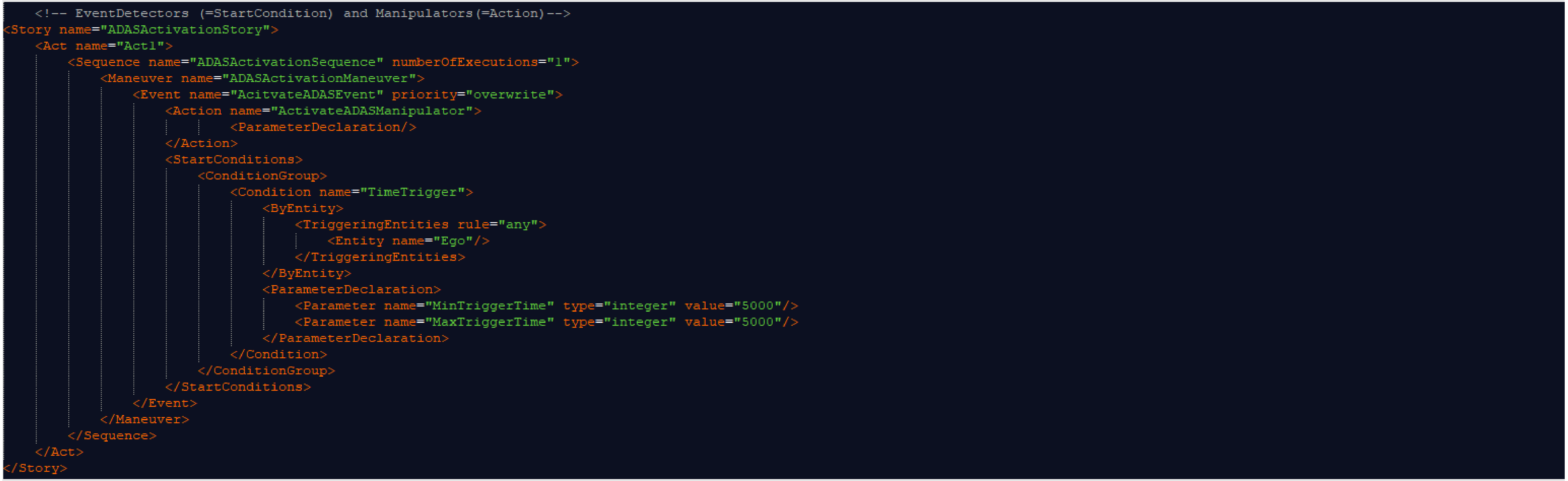
- Spawn-conditions for Ego and Scenario-Agents: Position, Velocity
- Stochastic Variation of Position and Velocity (custom implementation!)



```
<Init> <!-- initial position and velocity of agents  -->
    <Actions>
        <Private object="Ego">
            <Action>
                <Position>
                    <Lane roadId="1" s="20.0" laneId="-1" offset="0.0"/> <!-- position  -->
                    <Stochastics value="s" stdDeviation="10" lowerBound="5" upperBound="100"/>
                    <Stochastics value="offset" stdDeviation="0.3" lowerBound="-0.5" upperBound="0.5"/>
                </Position>
            </Action>
            <Action>
                <Longitudinal>
                    <Speed>
                        <Dynamics time="100" distance="1.0" rate="0.5" shape="linear"/>
                    </Speed>
                </Longitudinal>
            </Action>
        </Private>
        <Private object="Scenario">
            <Action>
                <Position>
                    <Lane roadId="1" s="0.0" laneId="-2" offset="-6.4566"/> <!-- position  -->
                </Position>
            </Action>
        </Private>
    </Actions>
</Init>
```

# OPENSCENARIO IN OPENPASS
# SCREENPLAY FOR THE SIMULATION

- Definition of conditions and actions for the simulation
- E.g. Activation of an ADAS at a certain time during the simulation
- Usage of Detectors and Manipulators



```
    <!-- EventDetectors (=StartCondition) and Manipulators(=Action)-->
<Story name="ADASActivationStory">
    <Act name="Act1">
        <Sequence name="ADASActivationSequence" numberOfExecutions="1">
            <Maneuver name="ADASActivationManeuver">
                <Event name="AcitvateADASEvent" priority="overwrite">
                    <Action name="ActivateADASManipulator">
                            <ParameterDeclaration/>
                    </Action>
                    <StartConditions>
                        <ConditionGroup>
                            <Condition name="TimeTrigger">
                                <ByEntity>
                                    <TriggeringEntities rule="any">
                                        <Entity name="Ego"/>
                                    </TriggeringEntities>
                                </ByEntity>
                                <ParameterDeclaration>
                                    <Parameter name="MinTriggerTime" type="integer" value="5000"/>
                                    <Parameter name="MaxTriggerTime" type="integer" value="5000"/>
                                </ParameterDeclaration>
                            </Condition>
                        </ConditionGroup>
                    </StartConditions>
                </Event>
            </Maneuver>
        </Sequence>
    </Act>
</Story>
```

# OPENSCENARIO IN OPENPASS
# ORGANISATION OF THE STANDARD

- Standard maintained by ASAM e.V.
- Kick-off workshop in November
- Roadmap: https://www.asam.net/conferences-events/detail/kick-off-workshop-asam-openscenario/

- Proposal workshop to define specific requirements and project plan (January 2019)
- Open points:
  - How do we handle stochastics?

# OPENSCENARIO IN OPENPASS FUTURE PLAN

Proposal by BMW:

- openScenario Implementation (with custom extensions) will be committed by BMW
- Configs for openPASS will be designed to fit to the openScenario Standard as far as possible
  - Add file header attributes
  - Add catalogs for vehicles and pedestrians (move from CombinationConfig.xml)
  - Add scenery reference (move from CombinationConfig.xml)
  - Adjust storyboard to comply with the standard (add some tags, etc.)
- Some differences to standard will exist further
  - Stochastics are not available
  - Environment and Traffic profile remain in CombinationConfig.xml
- Incorporate newer versions of openScenario as they are released

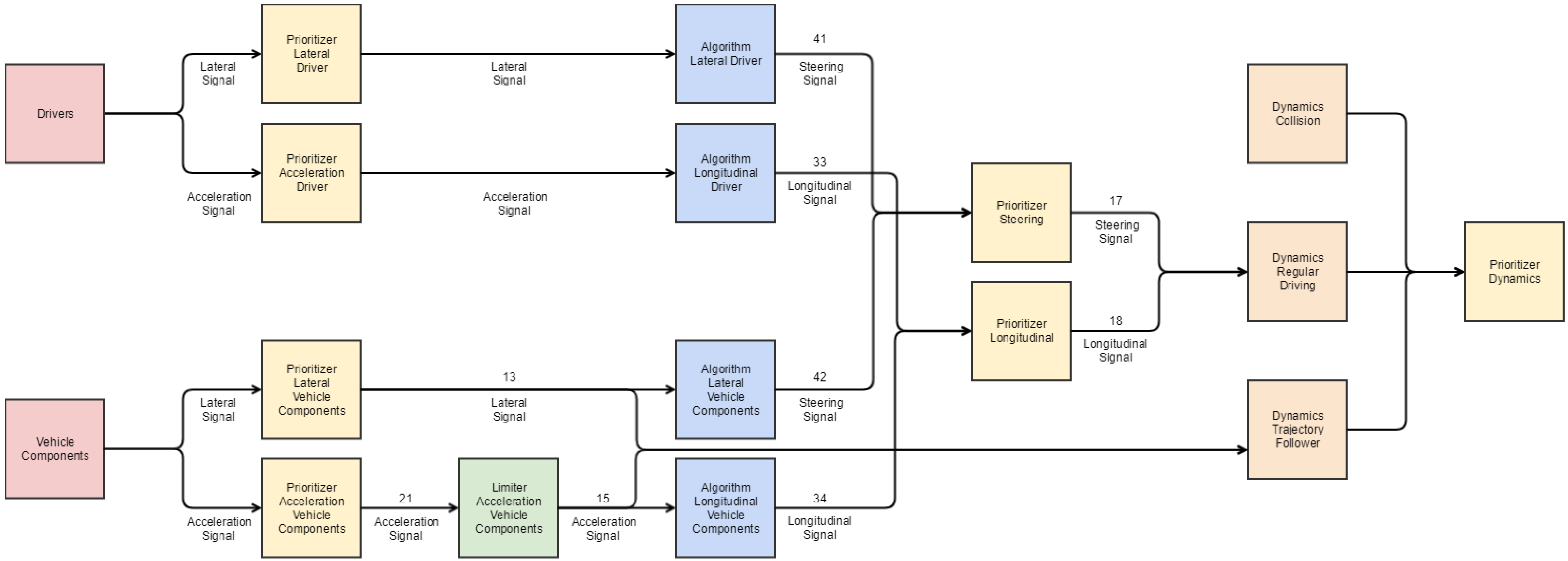
**Compliance with the Standard as target for the future**

openPASS

# MODULES AND COMPONENTS

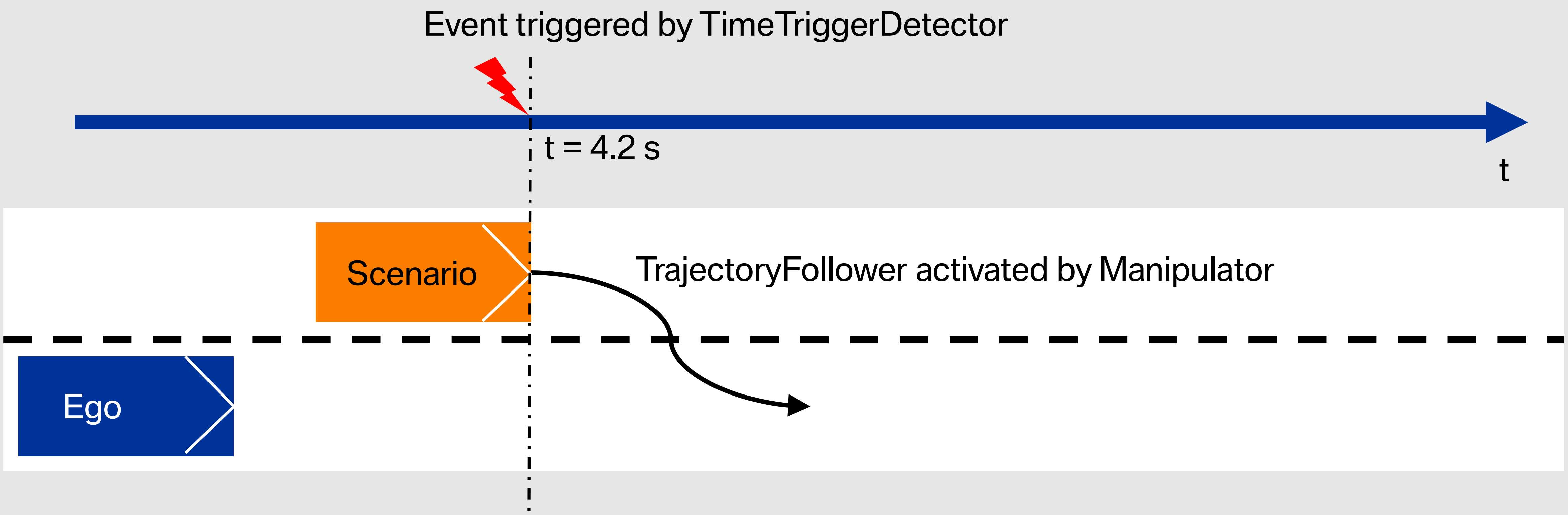
Freude am Fahren

# MODULES AND COMPONENTS
# AGENT STRUCTURE



# MODULES AND COMPONENTS
# SCREENPLAY WITH DETECTOR / MANIPULATOR

Screenplay according to example case with AEB:


Event triggered by TimeTriggerDetector
t = 4.2 s
t
Scenario
TrajectoryFollower activated by Manipulator
Ego

# MODULES AND COMPONENTS
# WHAT‘S NEW?

- **AlgorithmLongitudinalDriver / AlgorithmLongitudinalVehicle (Components)**

  Updates the agents pedal positions and gear.

- **AlgorithmLateralDriverTasks / AlgorithmLateralVehicle (Components)**

  Updates the agents steering wheel position.

- **ActionSecondaryDriverTasks (Components)**

  Updates the agents BrakingLight, Indicator, Horn and all Lightswitches (Headlight, HighBeam, Flasher).

- **DynamicsRegularDriving (Components)**

  Updates the agents position

- **DynamicsCollision (Components)**

  Calculates the collision partners

- **DynamicsTrajectoryFollower (Components)**

  This module forces agents to drive according to a specific trajectory, defined in a separate file.

# MODULES AND COMPONENTS WHAT‘S NEW?

- **SignalPrioritizer (Components)**
  - Collects signals of same kind and relays the one with highest priority
  - Example: Acceleration / Braking ADAS, Driver model
- **SensorRecordState (Components)**

  This module forwards the agent parameters to the Observer in each time step.
- **SensorFusion (Components)**

  The SensorFusion module allows unsorted aggregation of any data provided by sensors.
- **SensorObjectDetector (Components)**

  SensorObjectDetector is the common base of all sensors.
- **AlgorithmAgentFollowingDriverModel (Components)**

  Driver Model.

# MODULES AND COMPONENTS
# WHAT‘S NEW?

- **ParametersAgent (Components)**

  This component defines parameters for all other components of an agent. It includes constants and stochastic variables, which are divided into driver and vehicle parameters

- **VehicleControlUnit (todo)**

  Stores the states and acceleration of vehicle components and Driver

- **AgentUpdater (Components)**

  The AgentUpdater executes all Set-Methods of the agent dynamics after the DynamicsPrioritizer. This includes position, velocity, acceleration and rotation.

- **EventDetector, Manipulator (Core Modules)**
- **AEB (Components)**

**All modules are documented with inline doxygen documentation**

openPASS

# DOCUMENTATION

Freude am Fahren

openPASS

# NEXT STEPS AND TIMELINE

Freude am Fahren

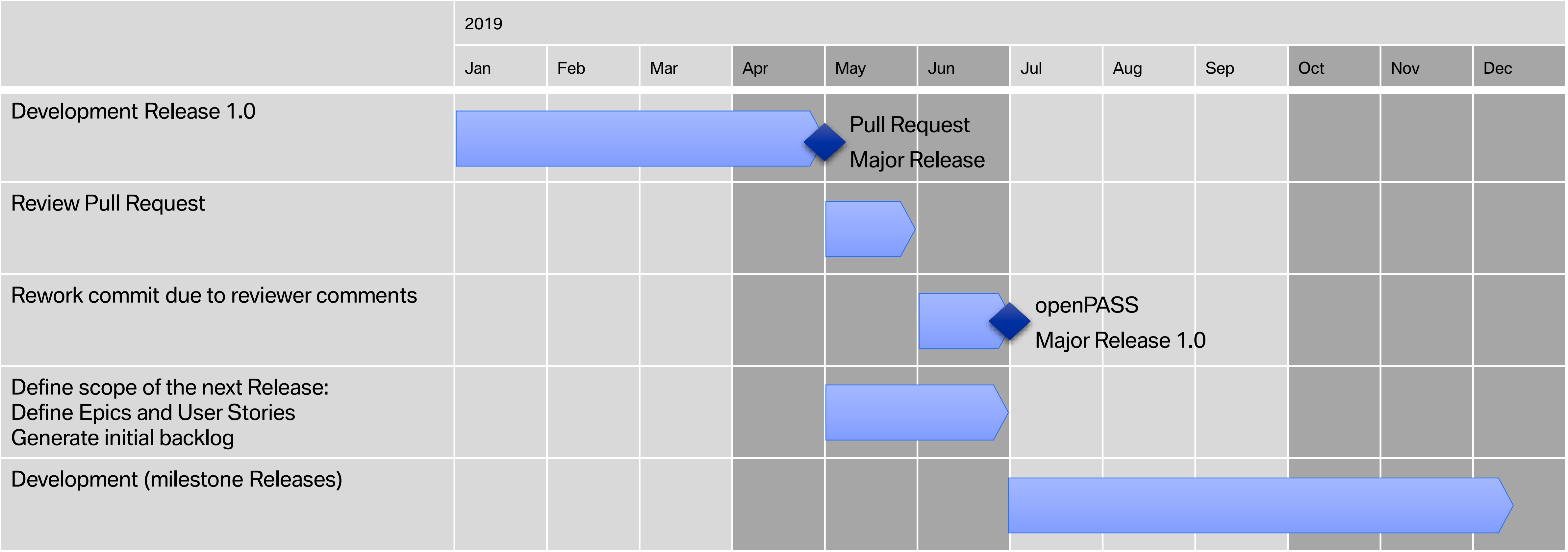
# NEXT STEPS

The future development should follow the process flow defined in 2018.

- ➢ Define Scope of the Release
- ➢ Create Epics and  User Stories in Backlog
- ➢ Define Milestone / Minor Releases as nessecary

**BMW will make a proposal in the next SC meeting**

# MILESTONES



2019

Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

Development Release 1.0

Pull Request

Major Release

Review Pull Request

Rework commit due to reviewer comments

openPASS

Major Release 1.0

Define scope of the next Release:
Define Epics and User Stories
Generate initial backlog

Development (milestone Releases)

openPASS

# LICENSE

# LICENSE

- OSI:

  Mozilla Public License

  - modified and copied source code must remain under the MPL
  - OSI is also planning to move to ASAM e.V.

- openScenario / openDrive:

  Association for Standardization of Automation and Measuring Systems - ASAM e.V.

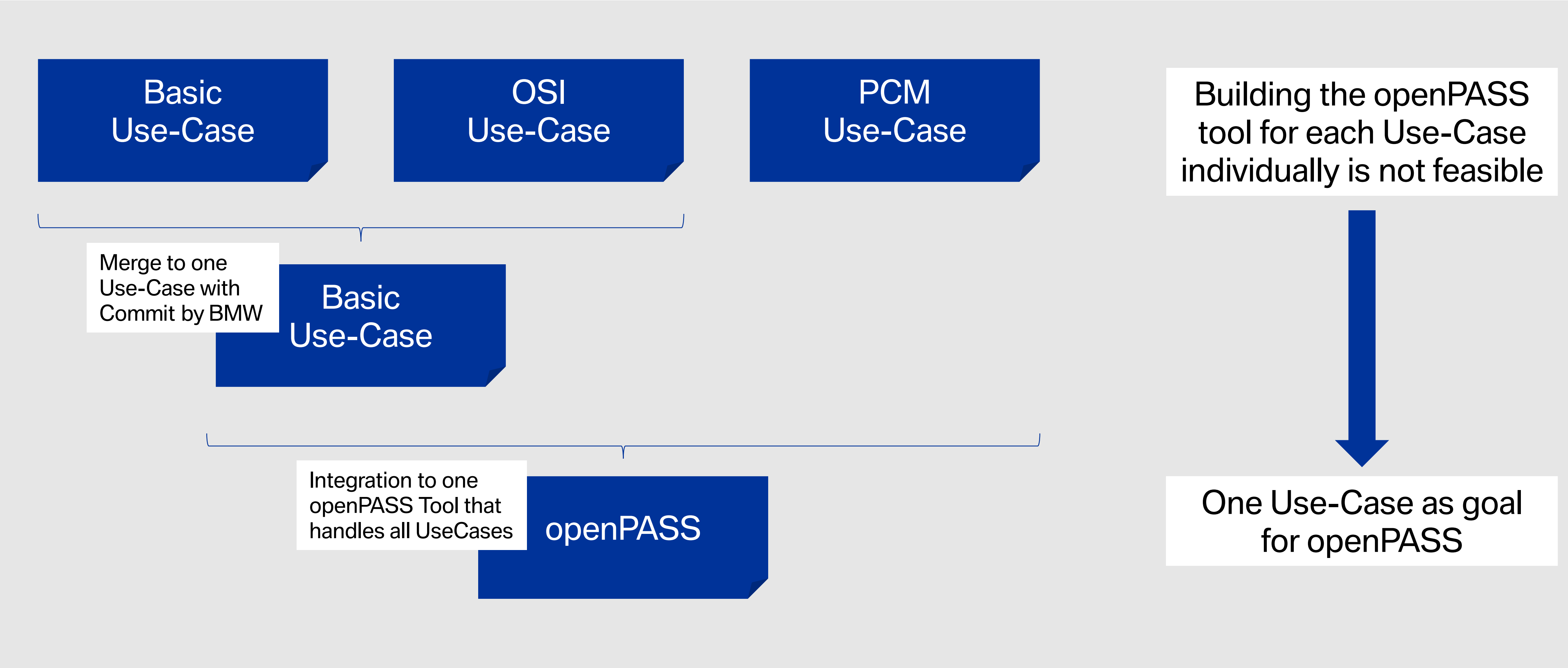
  License Terms: https://www.asam.net/license/

openPASS

# FURTHER TOPICS

Freude am Fahren

# HANDLING OF DIFFERENT USE-CASES


Basic Use-Case
OSI Use-Case
PCM Use-Case
Merge to one Use-Case with Commit by BMW
Basic Use-Case
Integration to one openPASS Tool that handles all UseCases
openPASS
Building the openPASS tool for each Use-Case individually is not feasible
One Use-Case as goal for openPASS