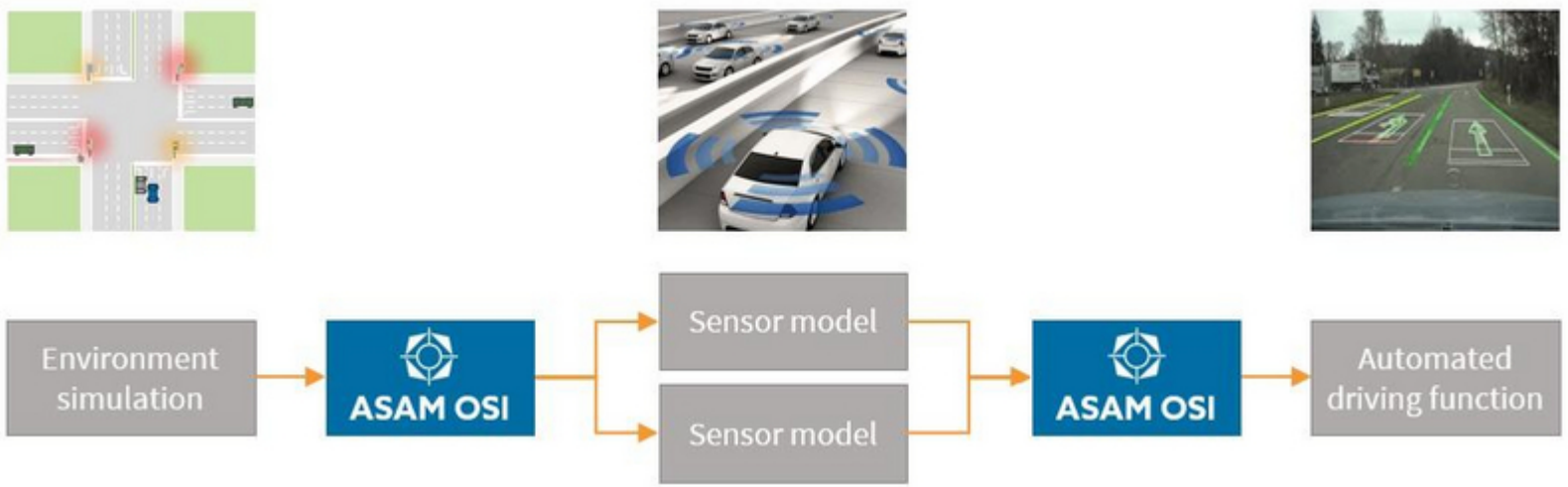
EMPOWERING OSI – BEYOND SENSOR MODELLING

AGENDA

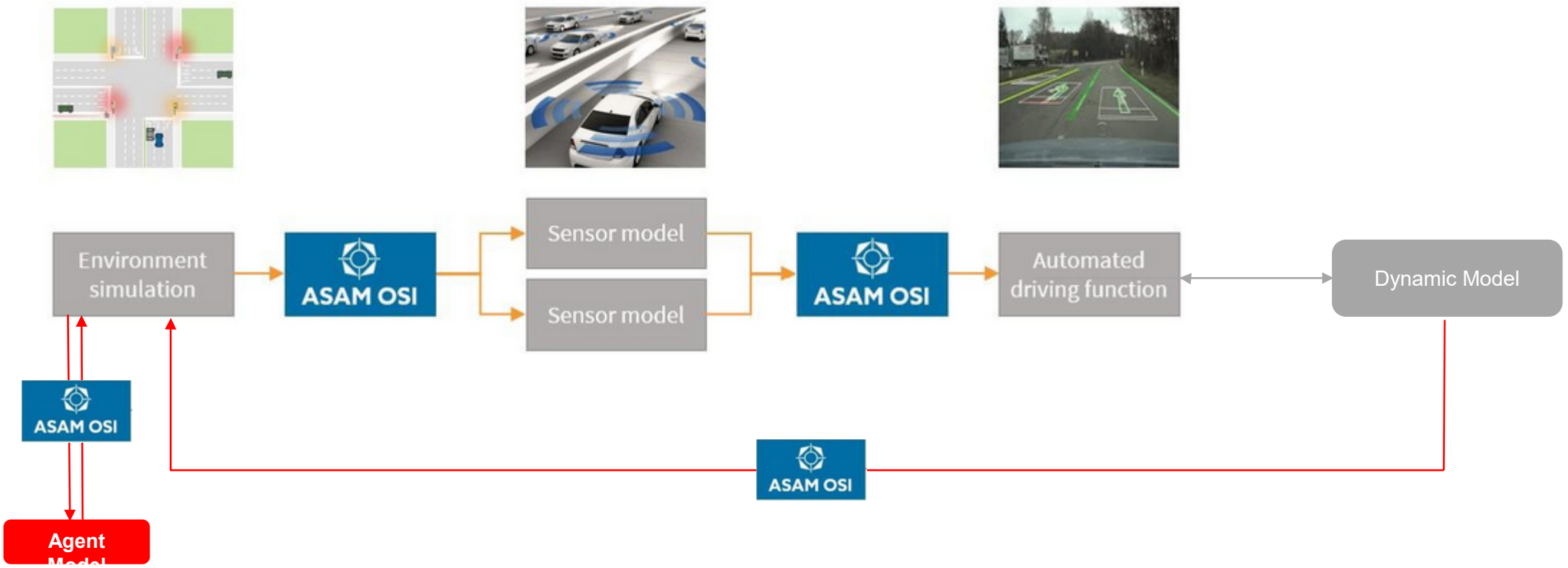
1. OSI today
2. Vision for the future
3. Example for a new proto-file: osi-vehicle
   - Usecase: Interface to Dynamic Model
   - Usecase: Visualization
   - Usecase: Measurement/Recording (AVL)
4. Customization
5. SETLevel 4to5

OSI TODAY



Quelle: https://www.asam.net/news-media/press-releases/detail/news/bmw-transfers-open-simulation-interface-osi-to-asam/

VISION FOR THE FUTURE



Quelle: https://www.asam.net/news-media/press-releases/detail/news/bmw-transfers-open-simulation-interface-osi-to-asam/ mit Erweiterungen

OSI-VEHICLE – THE BASIC CONCEPT

osi-vehicle = „a deeper description of vehicles“

➔ Division into vehicle categories/components as submessages:

- vehicle_kinematics
- vehicle_powertrain
- vehicle_steering_wheel
- vehicle_wheels
- vehicle_automated_driving_function
- ...

OSI-VEHICLE – USECASES

Original idea:

1) Formulation of an interface to the dynamic model

The message format revealed to be well suited for prototyping purposes:

2) Visualization interface

3) Recording of real vehicle measurements

OSI-VEHICLE – USECASE DYNAMIC MODEL

Usecase 1 (Dynamic Model):

Discussion points for the ASAM project group:

- How to deal with a bidirectional format?
  - + Leading through values, better debugging
  - + Flexible format for prototyping
  - - Previously: unidirectional OSI philosophy
- Or is a separation into Dynamic In/Out more reasonable?

Experience shows that standardization of the dynamic model interface is difficult.

OSI-VEHICLE – USECASE VISUALIZATION

Usecase 2 (Visualization):

Goal: Linking graphic engines via a standardized interface

- Step 1: Extended format osi-vehicle as basis
- Step 2: Learn which signals describe visible properties (e.g. wheel positions, … → Refinement over time)
- Step 3: Transfer to OSI::MovingObject (part of OSI::GroundTruth)

→ Future visualization via GroundTruth

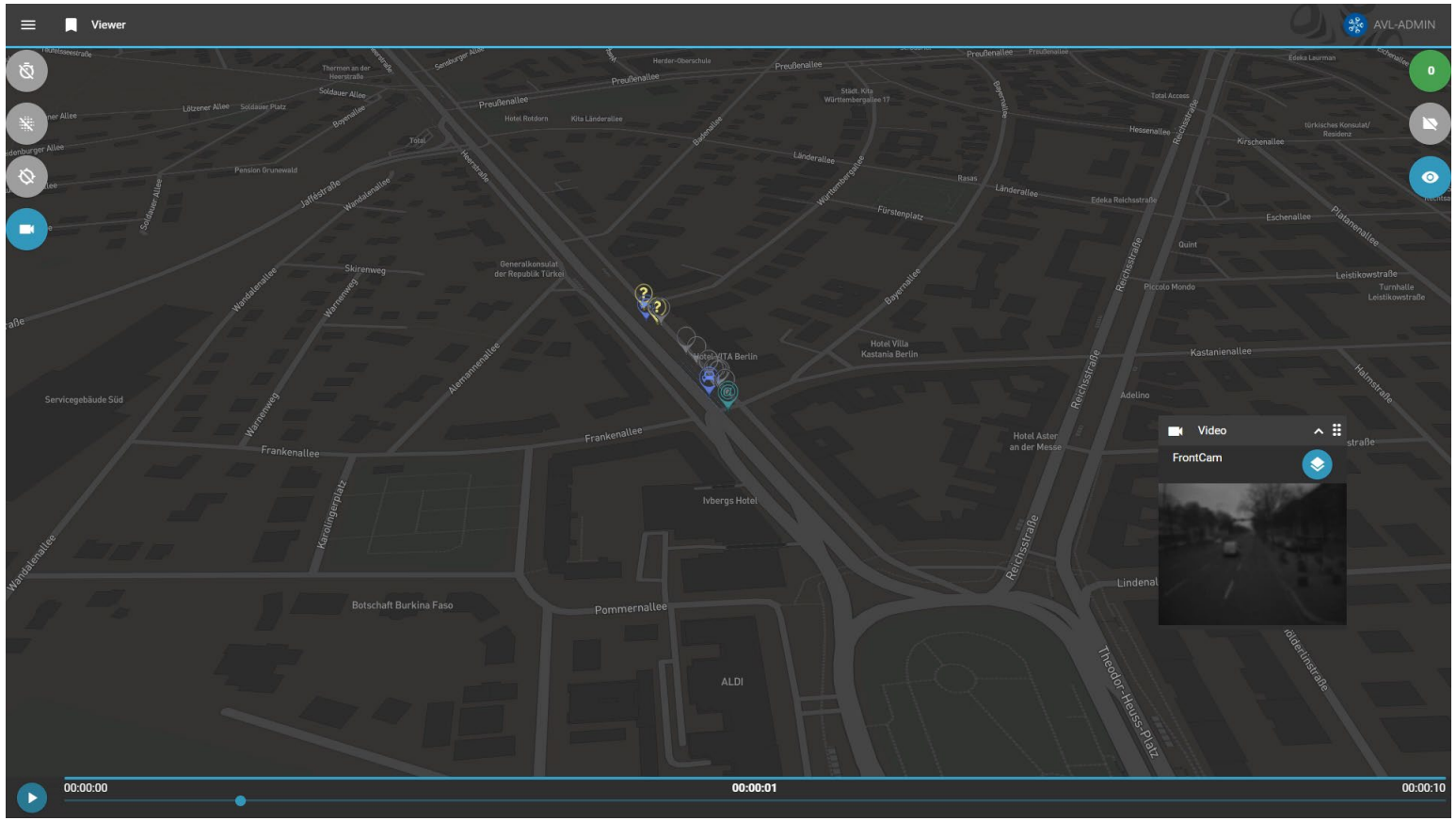
OSI-VEHICLE – USECASE MEASUREMENT (AVL)

Usecase 3 (Measurement):

- EGO is also participant of the very same scenario
- OSI is used as a measurement file within AVL‘s toolchain
  - Unifying of environment model measurement names
  - Unifying of EGO measurement names
- AVL cooperation with NorCom
  - Analyze and process OSI measurement files with DaSense (Big Data Analytics Platform) highly scalable in combination with time series data

OSI-VEHICLE – USECASE MEASUREMENT (AVL) VISUALIZATION FRAMEWORK PROTOTYPE

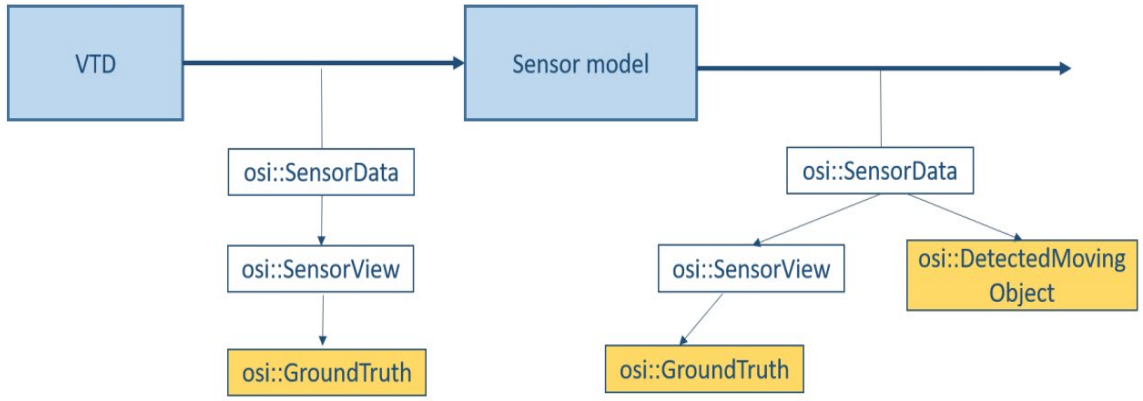
(USING UBER FRAMEWORK)



- OSI2XVIZ conversion
- AVL brand frontend
- Visualize and annotate scenarios for scenario extraction pipeline

OSI-VEHICLE – COMMENTS ON SENSOR MODELING (AVL)

Osi Sensor Model AVL



- Conversion from ground truth (osi::GroundTruth) to relative object list (osi::DetectedMovingObject)

- Objects filtered according to FOV of sensor, as a function of osi::EnvironmentalConditions:

  Visibility, Illumination, Precipitation

Possible Improvements for OSI ASAM

- Performance: Protobuf is optimized for bandwidth not speed
  → Serialization/Deserialization is slow!

- Definition of Environmental Conditions:
  discrete values not suitable for sensor model,
  parameters should be continuous

- Get rid of redundant messages like multiple ground truth
  → osi::GroundTruth in every osi::SensorView message

- How to handle raw data is not clearly defined

CUSTOMIZATION

Why customization?

- Special VR graphic requirements: In addition to pure traffic visualization also visualization of the OEM-specific display-operate concept (“Anzeige-Bedien-Konzept”)
- Other special requirements: Company/area-specific interface extensions

Central question:
How to account for company/area specific signals?

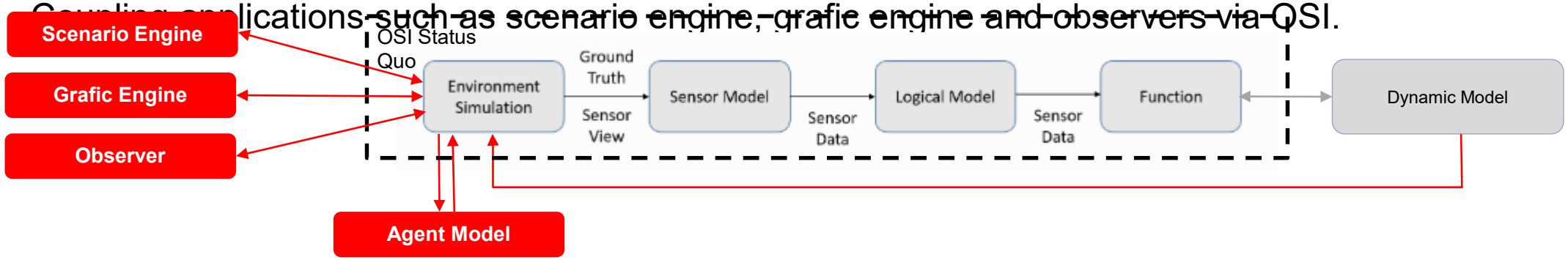
- So far: Appending additional fields in the same proto-file
  ➔ Unstable, adjustment with every OSI release necessary

Introduce Extension-Mechanism with google.proto.any (proto3) or extensions (proto2)
➔ Version-stable inclusion of proprietary .proto files

SETLEVEL 4TO5

Goals:

- Development of multi-agent simulations with externally coupled (e.g. fmi-packaged) Agent models.
- Empowering OSI for coupling Agent models in a standardized manner.
    - Using existing concepts (e.g. OSI::SensorView)
    - Development of OSI::TrafficCommand (in accordance to OpenSCENARIO 1.0) to allow external control of Agent models
    - Development of OSI::TrafficUpdate + extending OSI::MovingObject with visible properties of Agents (vehicle / pedestrian)
- Coupling applications such as scenario engine, grafic engine and observers via OSI.



THANK YOU FOR YOUR ATTENTION!