MUNICH

Connext®
2019

May 21-22

Frederic Chucholowski

Vector

Sensor Simulation for Closed-Loop Testing of ADAS/AD Systems

Dr. Frederic Chucholowski, Manager Product Development
frederic.chucholowski@vector.com
+49 89 74 7377-29

Agenda

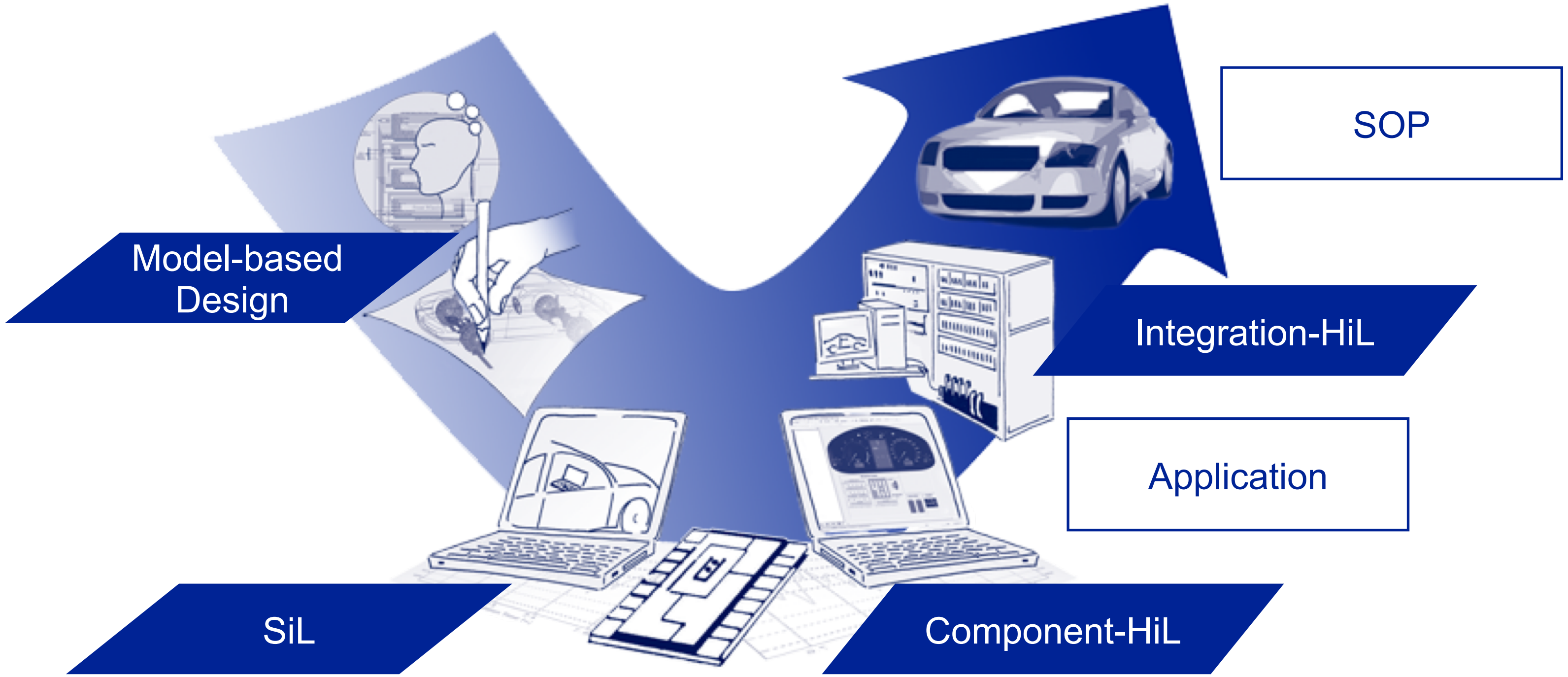
- Who we are
- Closed Loop Testing
- Sensor Simulation - Challenges
- Solution
- Connection to ROS
- Conclusion / Outlook

TESIS – Vector Informatik

VECTOR TESIS

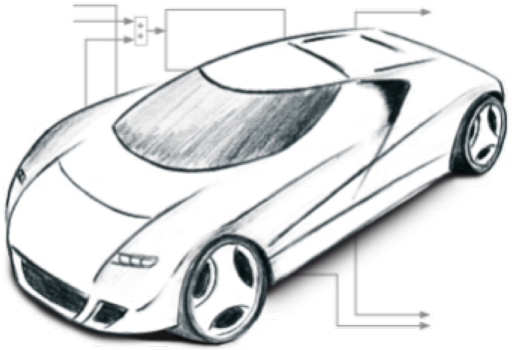
- TESIS (Technical Simulation and Software)
- Vehicle Simulation since 1985
  - Vehicle Dynamics
  - Thermodynamic Engine
  - Hybrid Drivetrain
  - Traffic
  - Real-Time (HiL) or faster
- Member of Vector Group since 2019

Using Simulation to Support Development Process



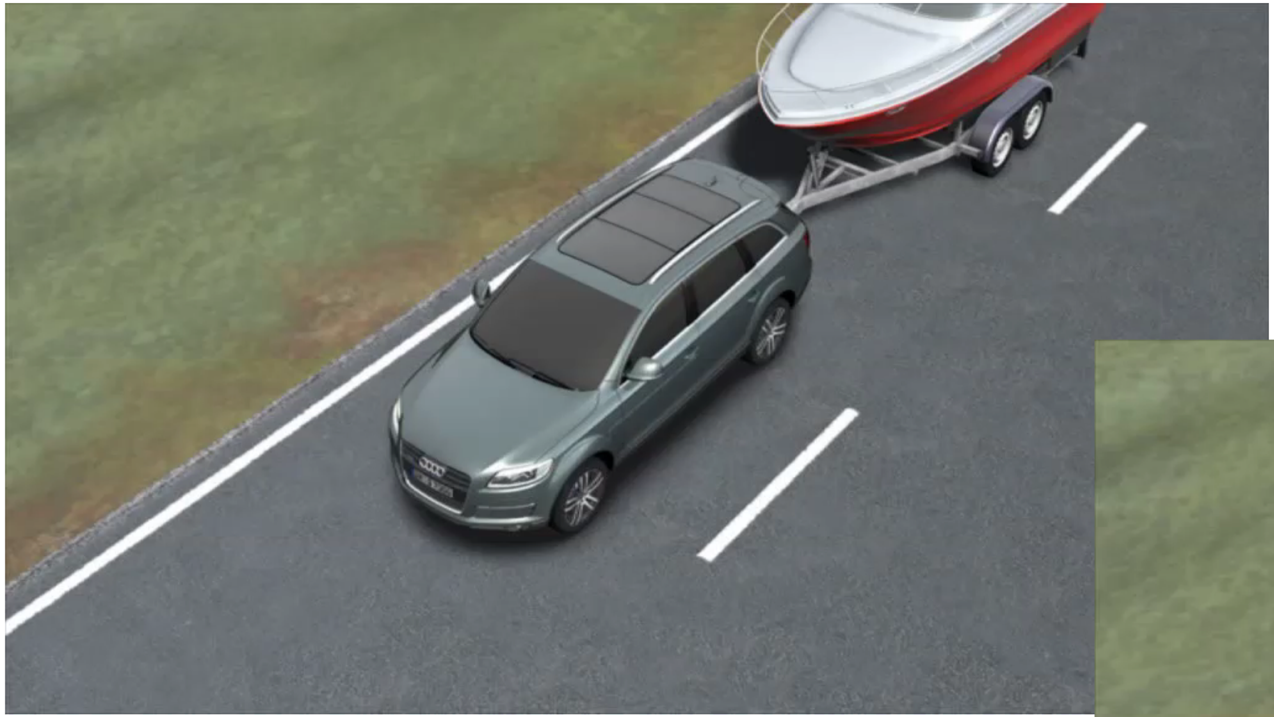
Virtual HiL → Real HiL, Vehicle

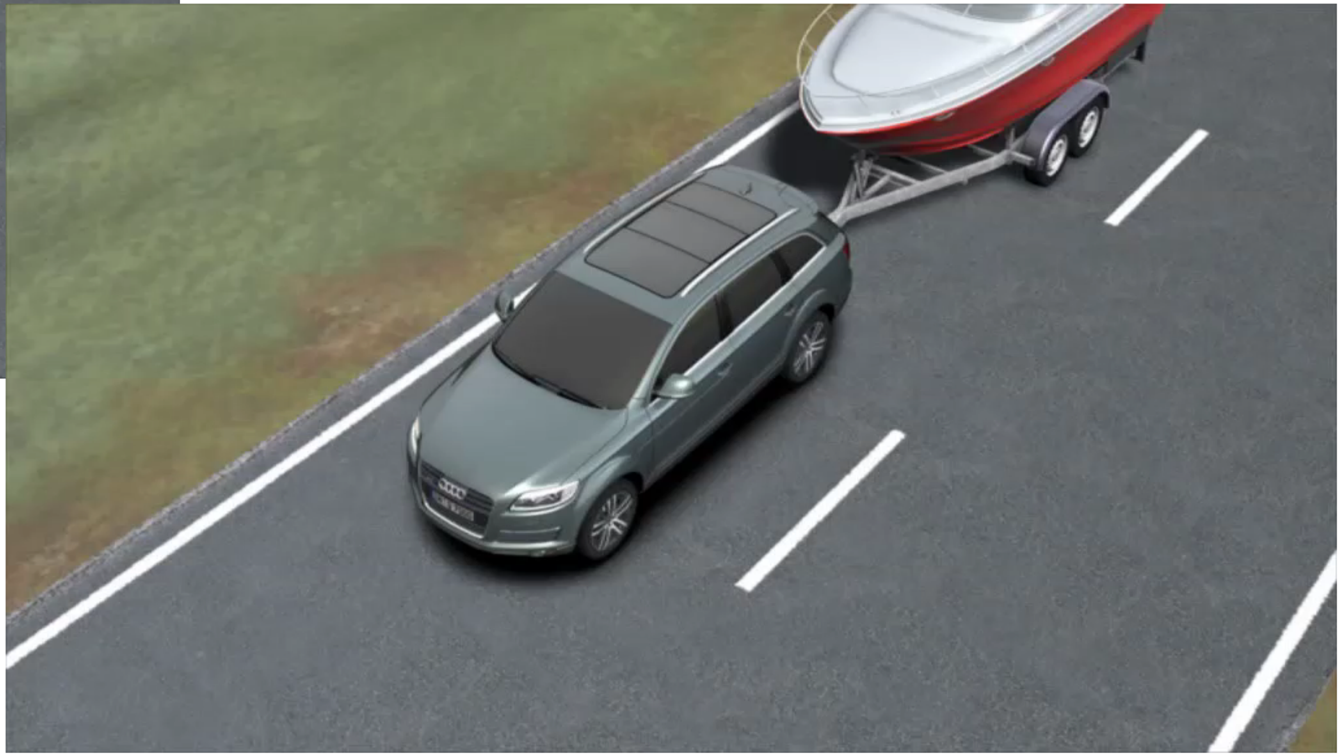
Virtual Vehicle



Real Vehicle

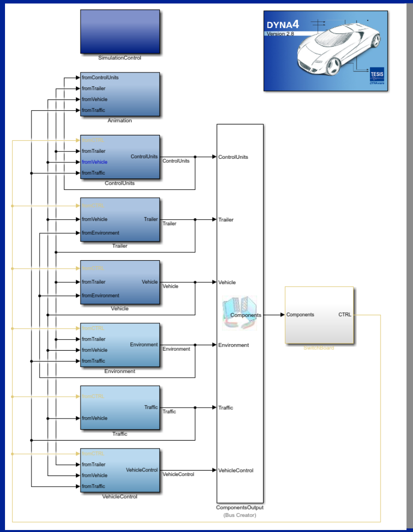
Example: HiL-Test of ESC-Function for Trailer Stabilization





System Setup for Vehicle Dynamics Simulation

Simulink (DYNA4)



Vehicle Dynamics Model

Shared Memory

TCP / UDP

TCP / UDP

DYNAanimation

Visualization

Compiled Executable

RT Target e.g. Vector CANoe

Problems:

- Portability - Configuration of target/port
- Late joining

Communication protocol details

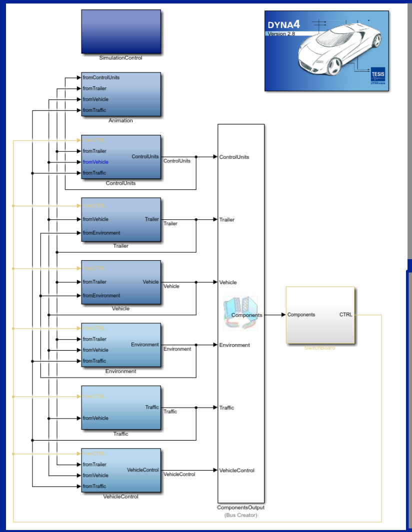
Object movement messages

- **Initialization** frame
  - Contains signal names
  - Sent once after TCP connection has been established
- **Data** frame
  - One float value per signal
  - Sent cyclically (e.g. every 1ms)
- **Termination** frame
  - Sent once after a simulation is finished

```
'vehicle.position.x'
'vehicle.position.y'
'vehicle.position.z'
'vehicle.rotation.x'
'vehicle.rotation.y'
'vehicle.rotation.z'
'vehicle.engineSpeed'
'vehicle.steeringWheelAngle'
'vehicle.speed'
'vehicle.gear'
'vehicle.brakeFlag'
'vehicle.frontAxle.left.wc.position.x'
'vehicle.frontAxle.left.wc.position.y'
'vehicle.frontAxle.left.wc.position.z'
'vehicle.frontAxle.right.wc.position.x'
'vehicle.frontAxle.right.wc.position.y'
'vehicle.frontAxle.right.wc.position.z'
'vehicle.rearAxle.left.wc.position.x'
'vehicle.rearAxle.left.wc.position.y'
'vehicle.rearAxle.left.wc.position.z'
```

System Setup for ADAS/AD Simulation

Simulink (DYNA4)



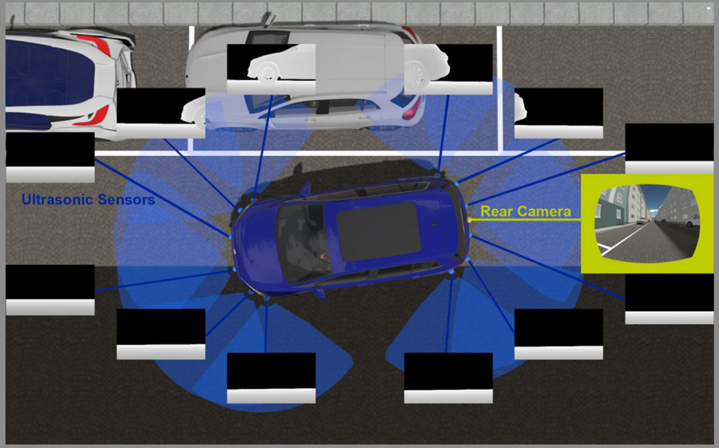
Vehicle Dynamics Model

Traffic Objects

DYNAanimation

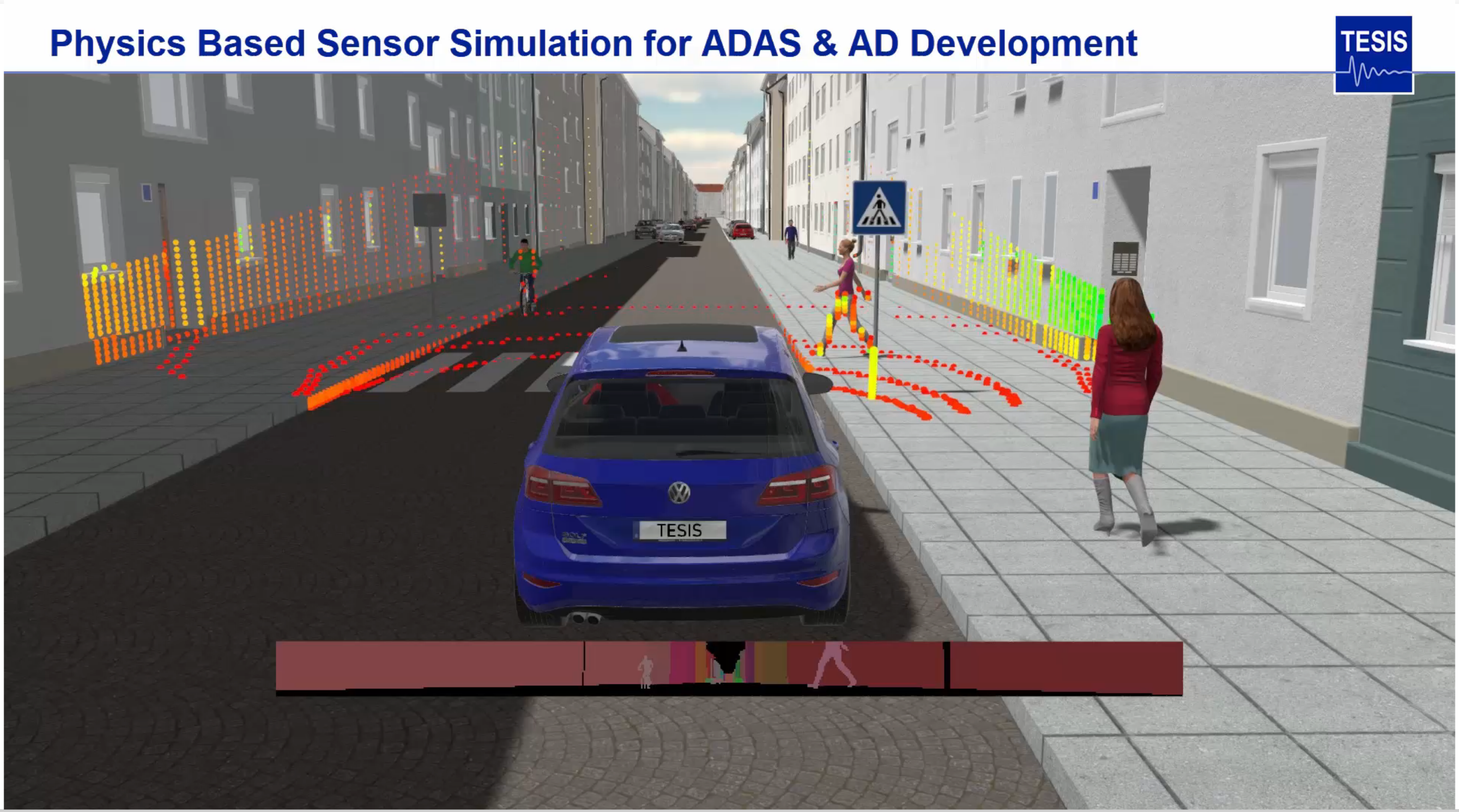
Visualization

GPU-based Sensors



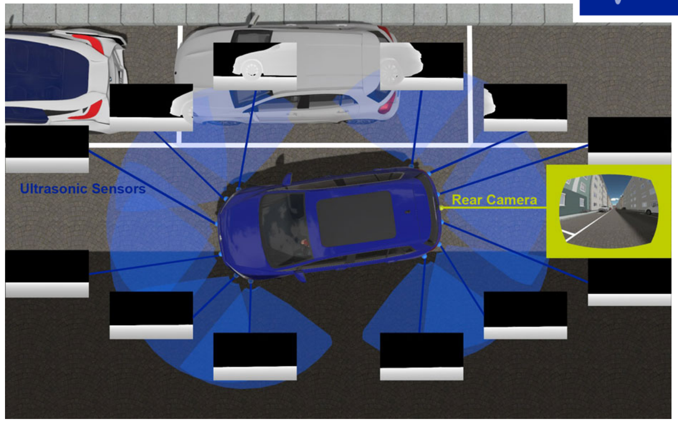
- Return sensor samples
- Different sample types
- Great amount of data

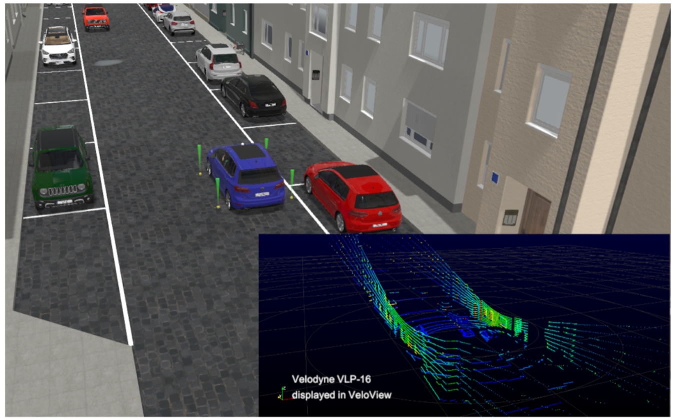
Video: Physics Based Sensor Simulation for ADAS & AD Development

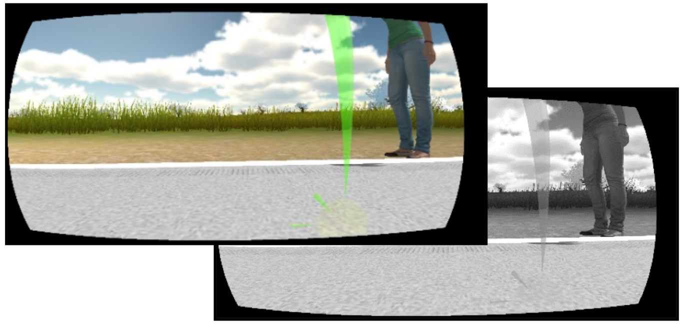


GPU-based Sensor Models: Ultrasonic and Lidar

- **Ultrasonic Sensor**
  - Propagation and atmospheric damping
  - Absorption and reflection based on object geometry and material properties
  - Output: intensity depth histogram
- **Lidar Sensor**
  - Reflection intensity based on angle between laser beam and object surface and its material properties
  - Rotating and non-rotating lidar sensors
  - Output: 3D point cloud as ROS Topic or in Velodyne format
- **Camera Sensor**
  - Adjustable distortion (OpenCV parameters) and color filters
  - Output: RGB image stream

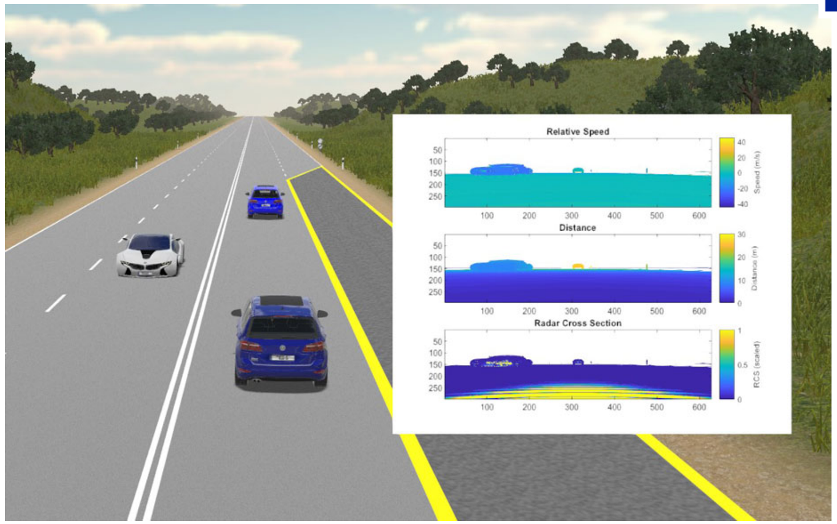


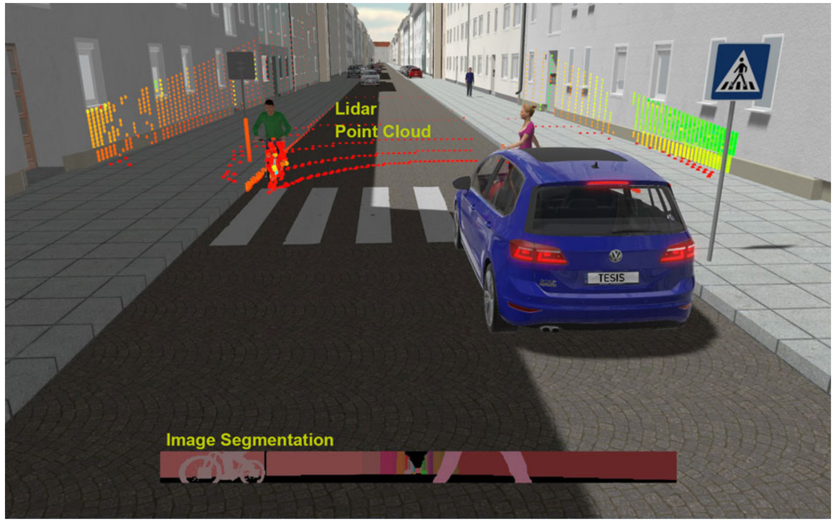




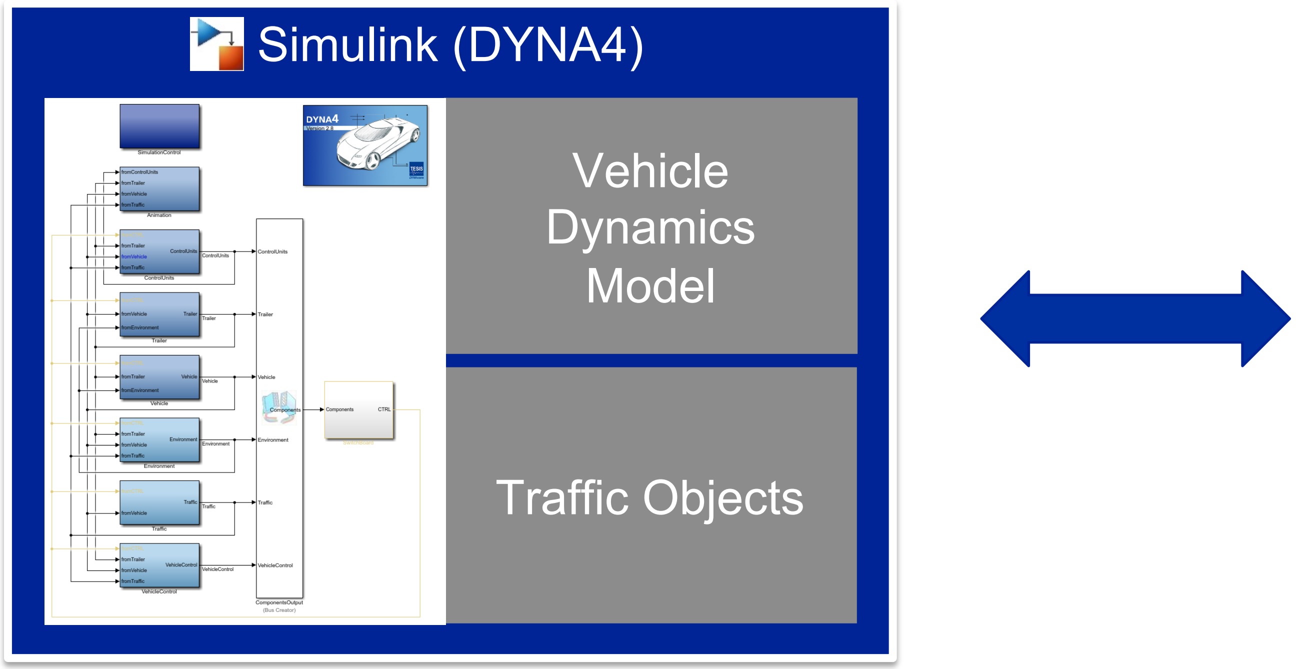
GPU-based Sensor Models: Radar and Camera

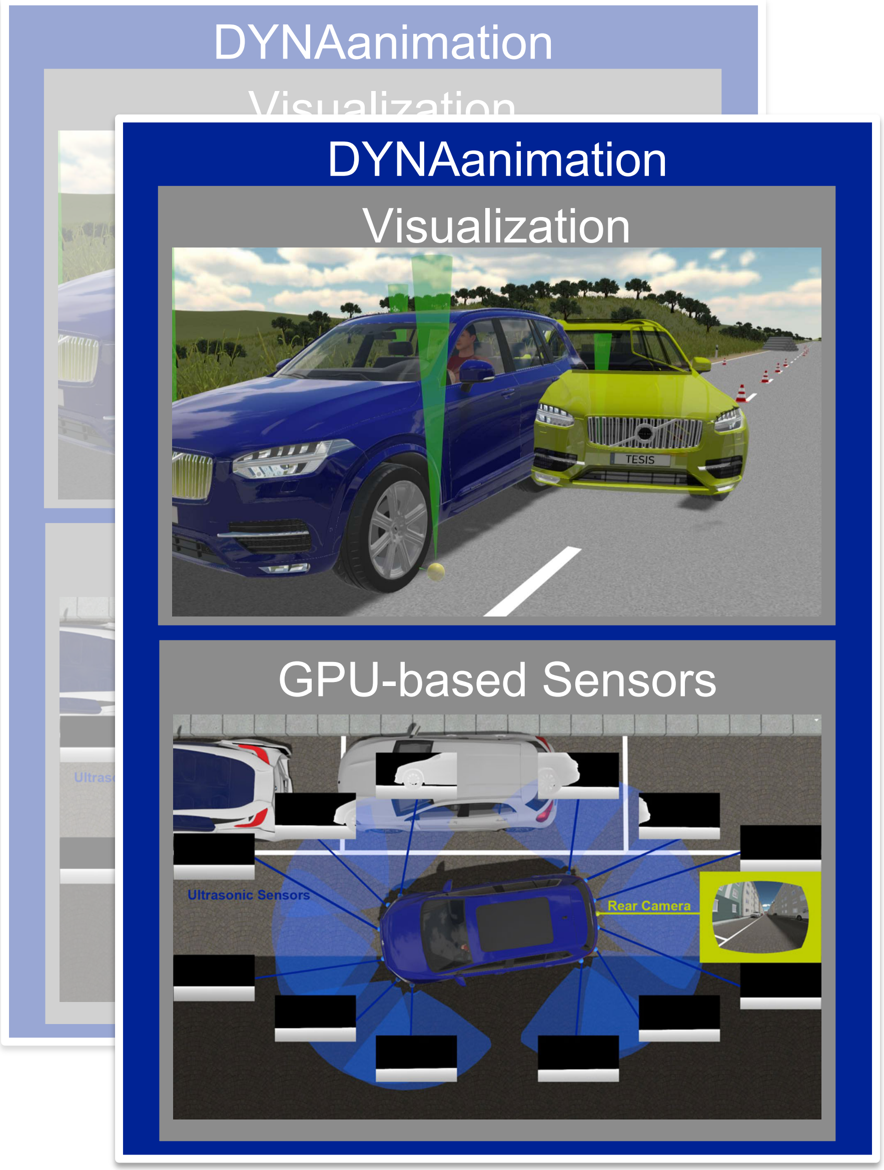
- **Radar Sensor**
  - Scattered radar waves based on object geometry and material properties
  - Consideration of different antenna characteristics (Short-, Mid-, Long-Range)
  - Output: relative velocity, distance to the object, intensity of the electric field
  - Optional: Range Doppler plots by GPU-based Fourier transformation
- **Semantic Image Segmentation (Object List)**
  - Automatic semantic image segmentation for depicting an ideal sensor fusion
  - Consideration and classification of all objects available in the object catalog with configurable object classes
  - Output: relative velocity, distance, object class





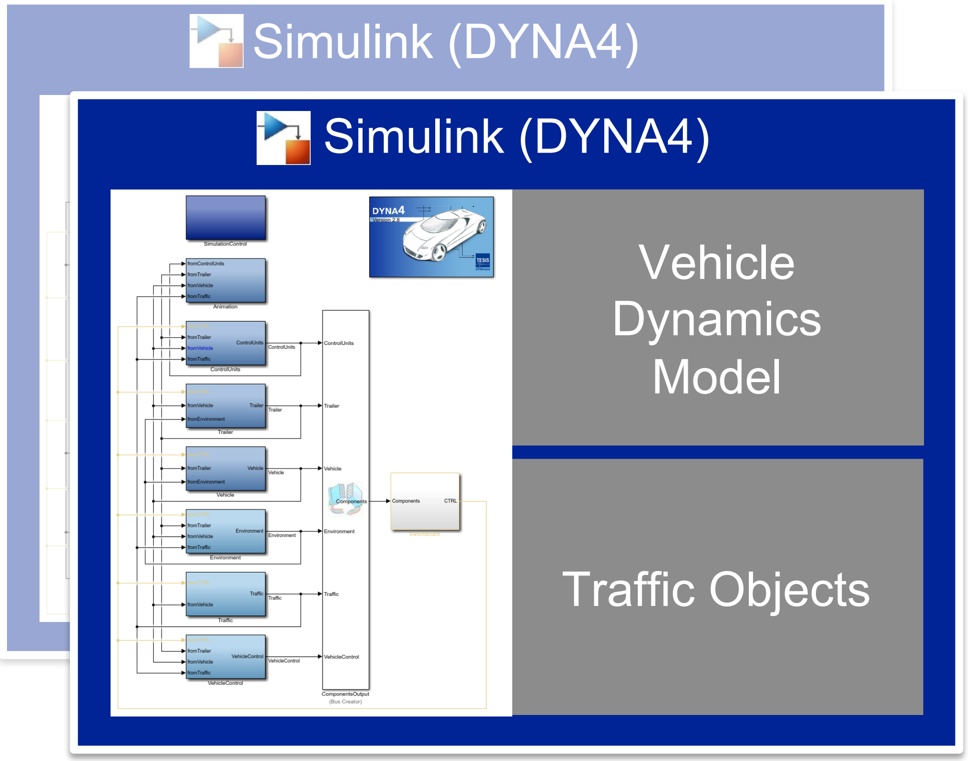
System Setup for ADAS/AD Simulation - Scalability

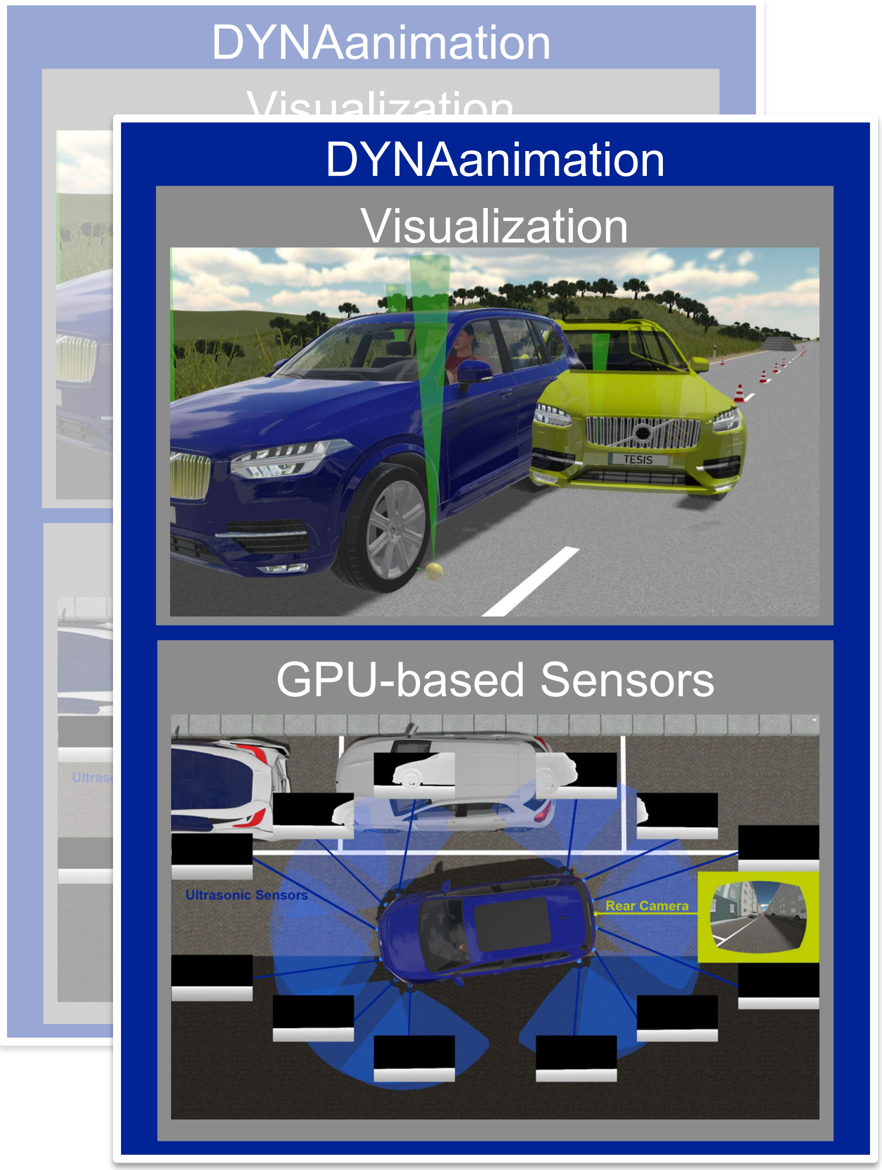




- Sensor calculation on multiple machines

System Setup for ADAS/AD Simulation - Scalability



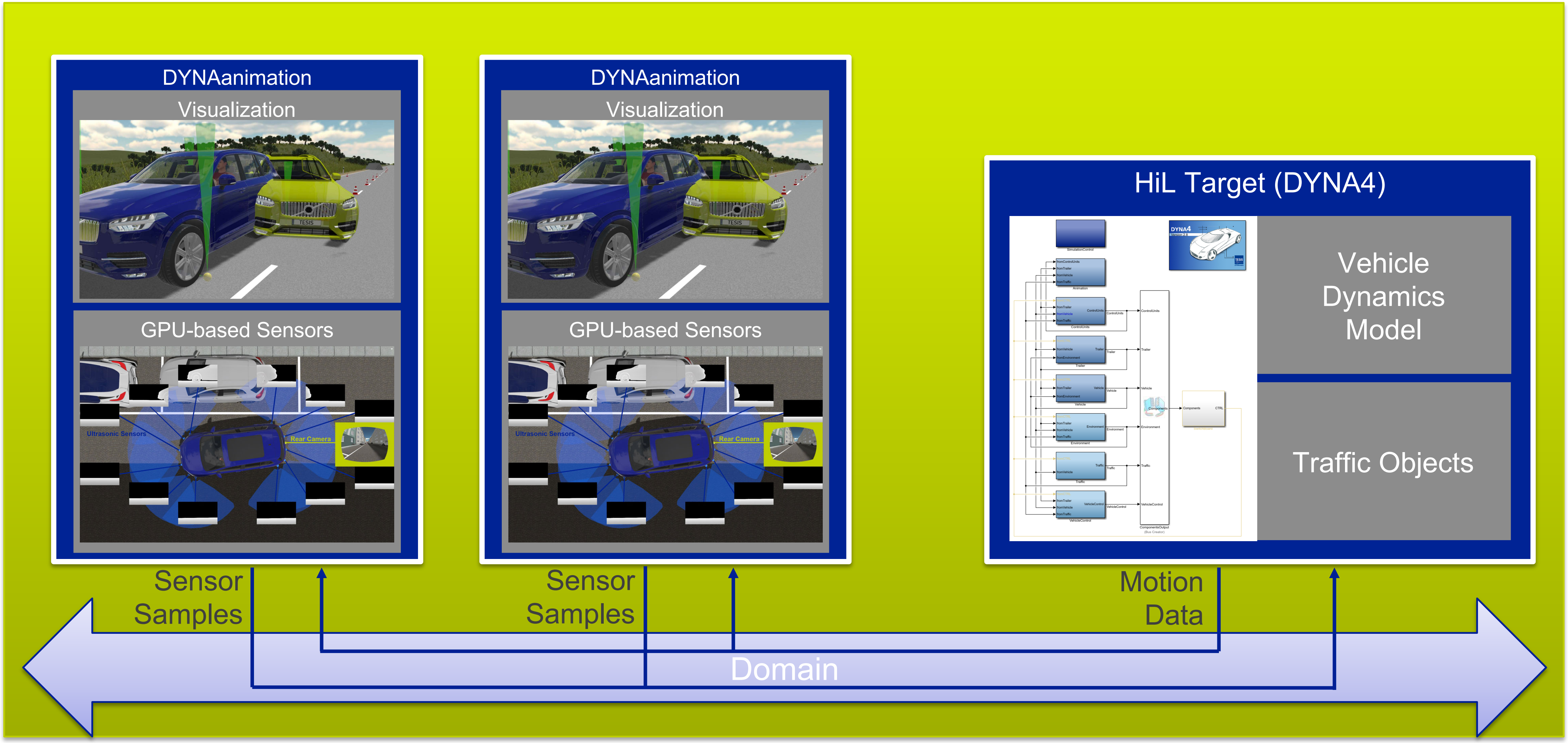


- Multiple VuT
- Multiple sources for traffic objects

Challenges

- Portability
  - Run simulations on local machine out of the box
  - Automatic discovery of participating nodes
- Easily set up multi-machine simulations
- Show data in animation even if started after the simulation
- Efficiently transport
  - Object movement signals
  - Sensor samples

Solution: RTI Connext DDS to handle communications



Advantages

- Discovery protocol
- Use hostname as default partition name to separate simulation instances
- Transient initialization frame to support late joining
- Automatically use shared memory if possible
- Easy extensible for new topics

Support for ROS

What is ROS?

- The Robot Operating System (ROS) is a set of software libraries and tools that help you build robot applications.
- From drivers to state-of-the-art algorithms, and with powerful developer tools, ROS has what you need for your next robotics project. And it’s all open source.
- ROS is a distributed framework of processes (aka *Nodes*) that enables executables to be individually designed and loosely coupled at runtime.

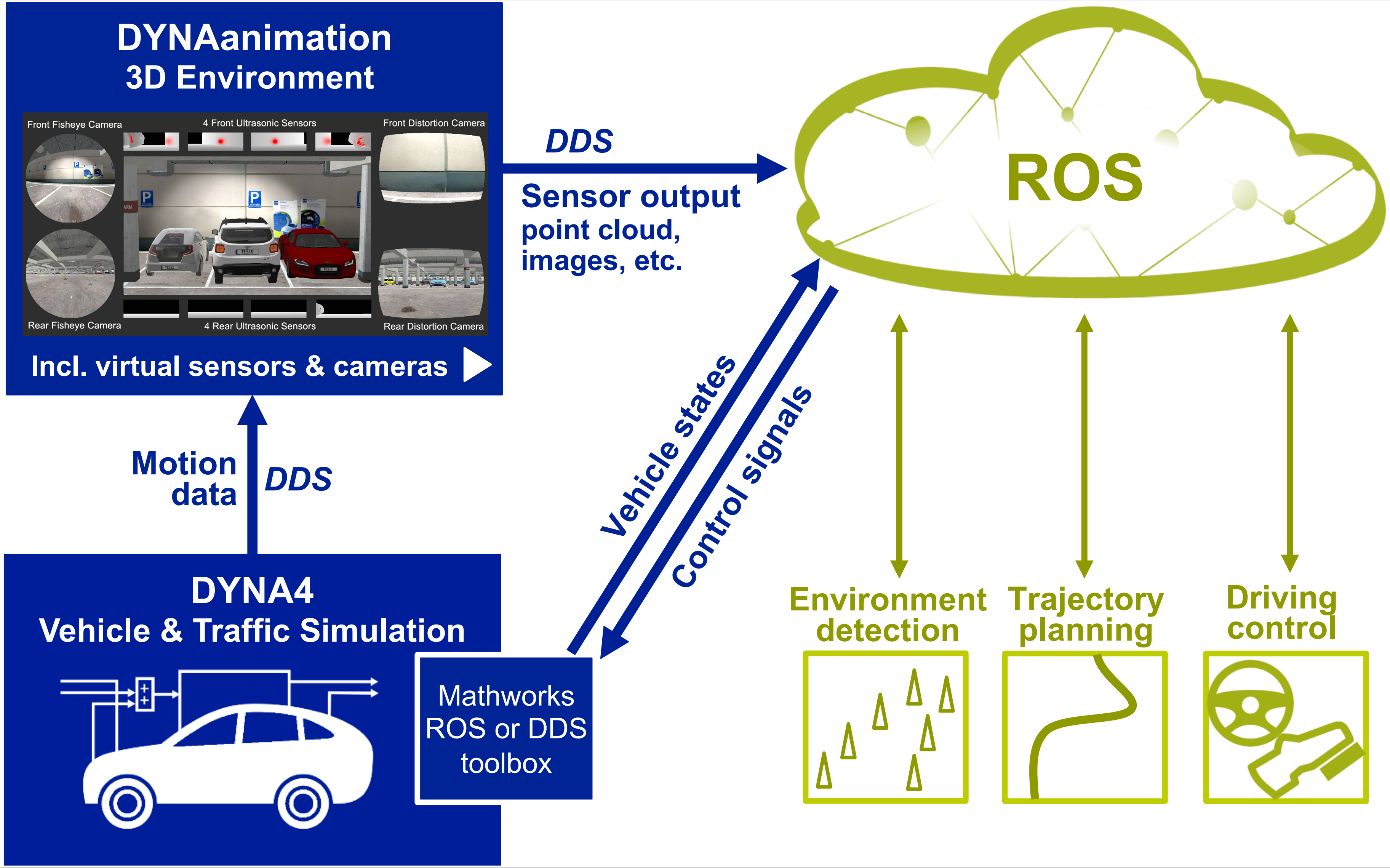
ROS2

- Successor of ROS (1)
- Proprietary communication layer replaced by DDS

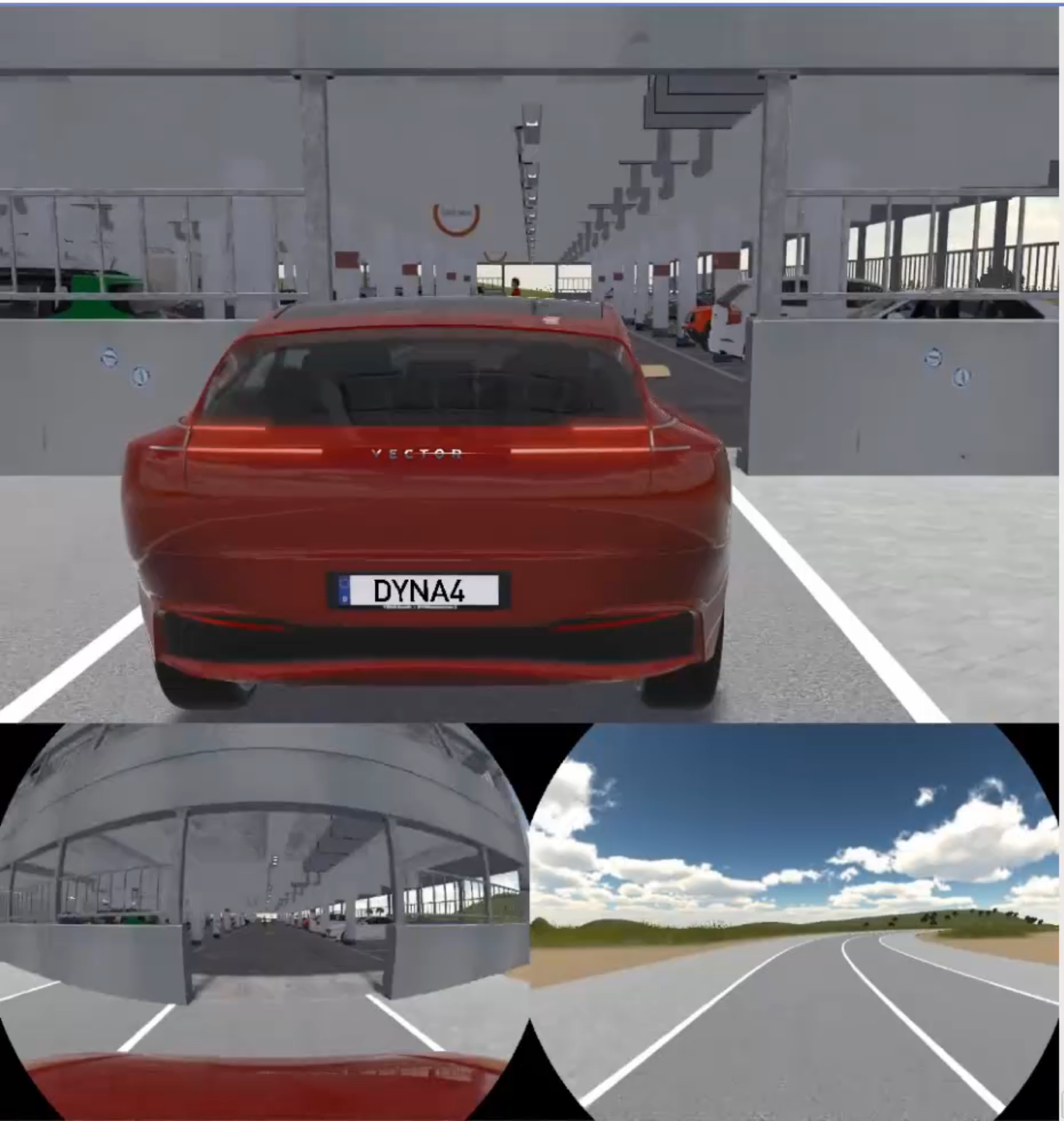
Access to ROS2

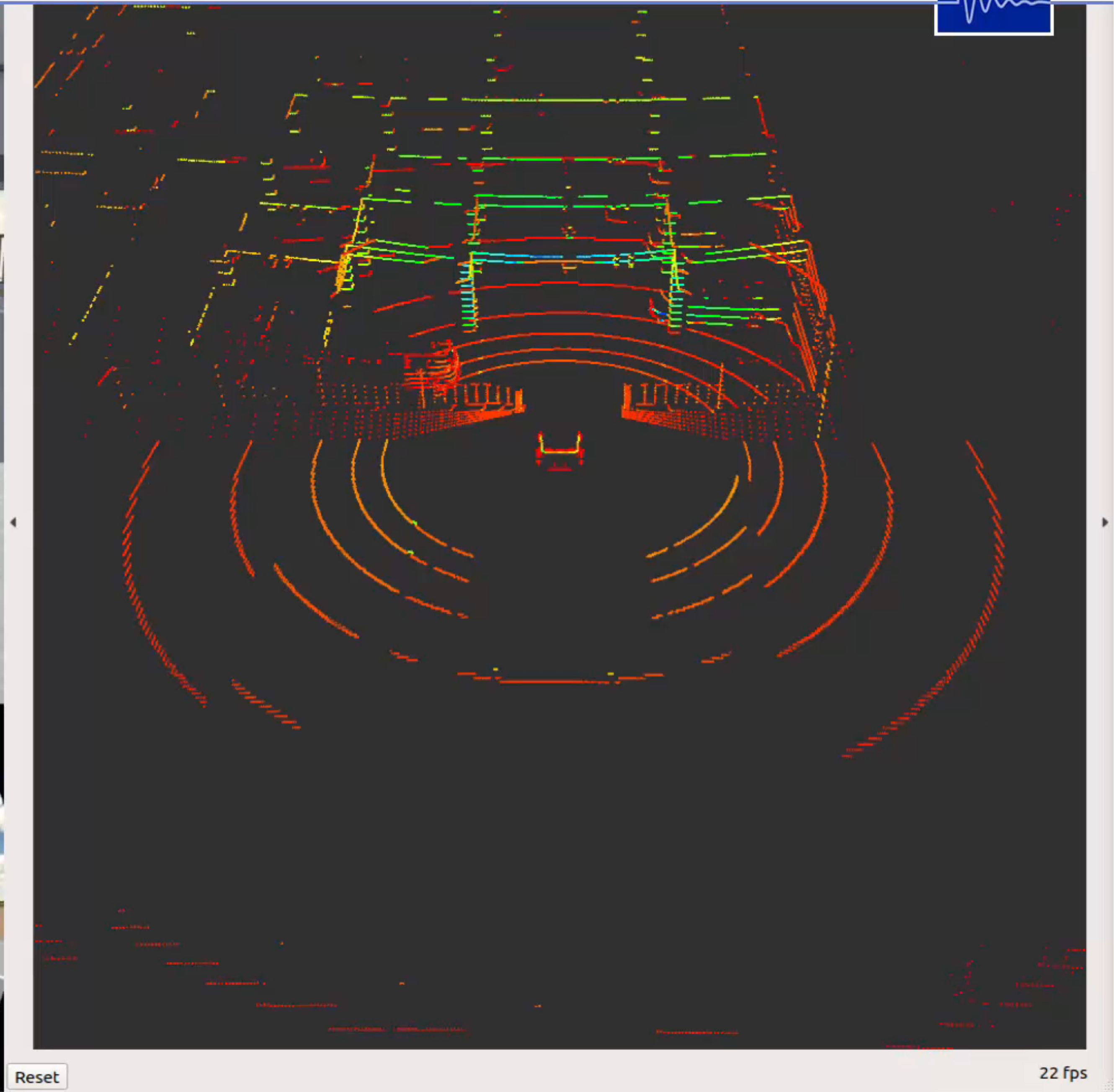
- Easy participation in ROS2 network
- No implementation had to be done!
- ROS1 access via ROS1 bridge
- Use ROS2 standard topics for sensor samples
  - Image
  - Point Cloud
  - Transform

DYNA4 to ROS Connection



ADAS Development with DYNA4 and ROS





Conclusion / Outlook

- DDS has proven to suit our requirements
- New sensor sample types can be added easily
- RTI Admin Console plays a vital role in the development process

Next steps:

- Make DDS communication available for all real time platforms supported by DYNA4
- Signal logging using DDS
- Simulation control using DDS