Simulation hochdynamischer omnidirektionaler mobiler Roboter in NVIDIA Isaac Sim

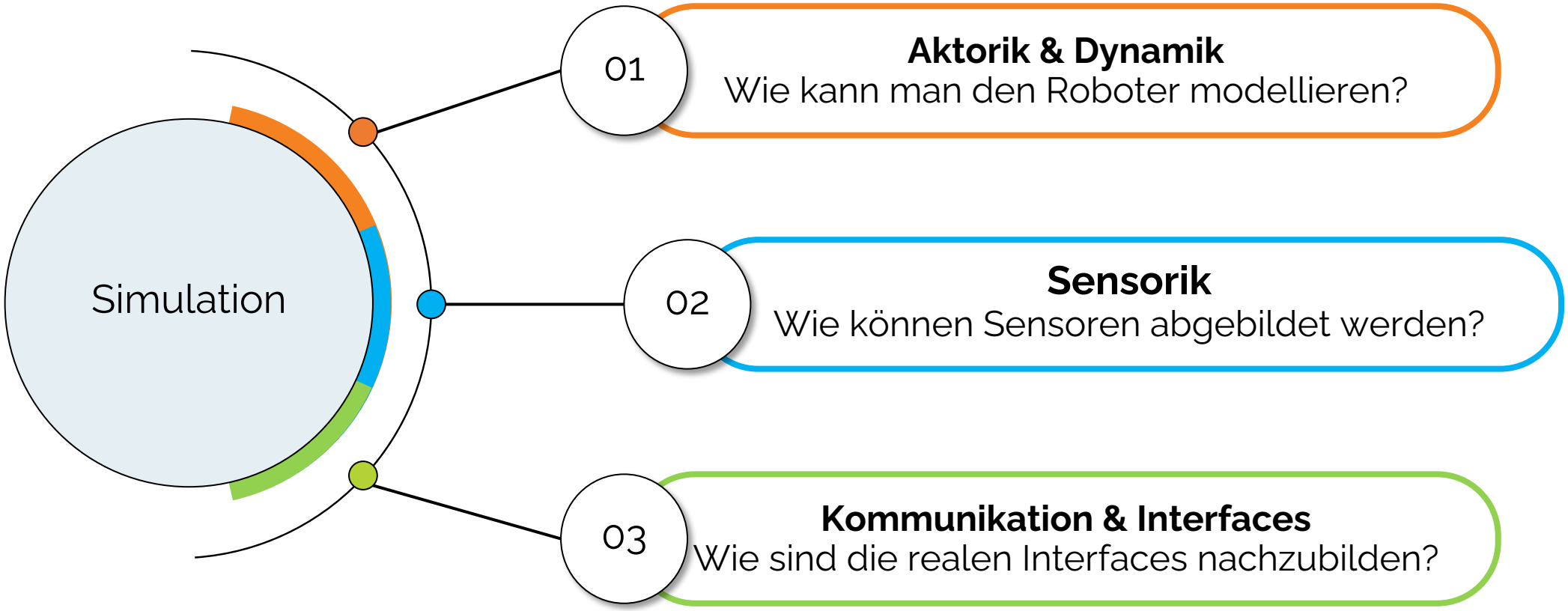
Marvin Wiedemann - Fraunhofer IML

Fraunhofer
IML

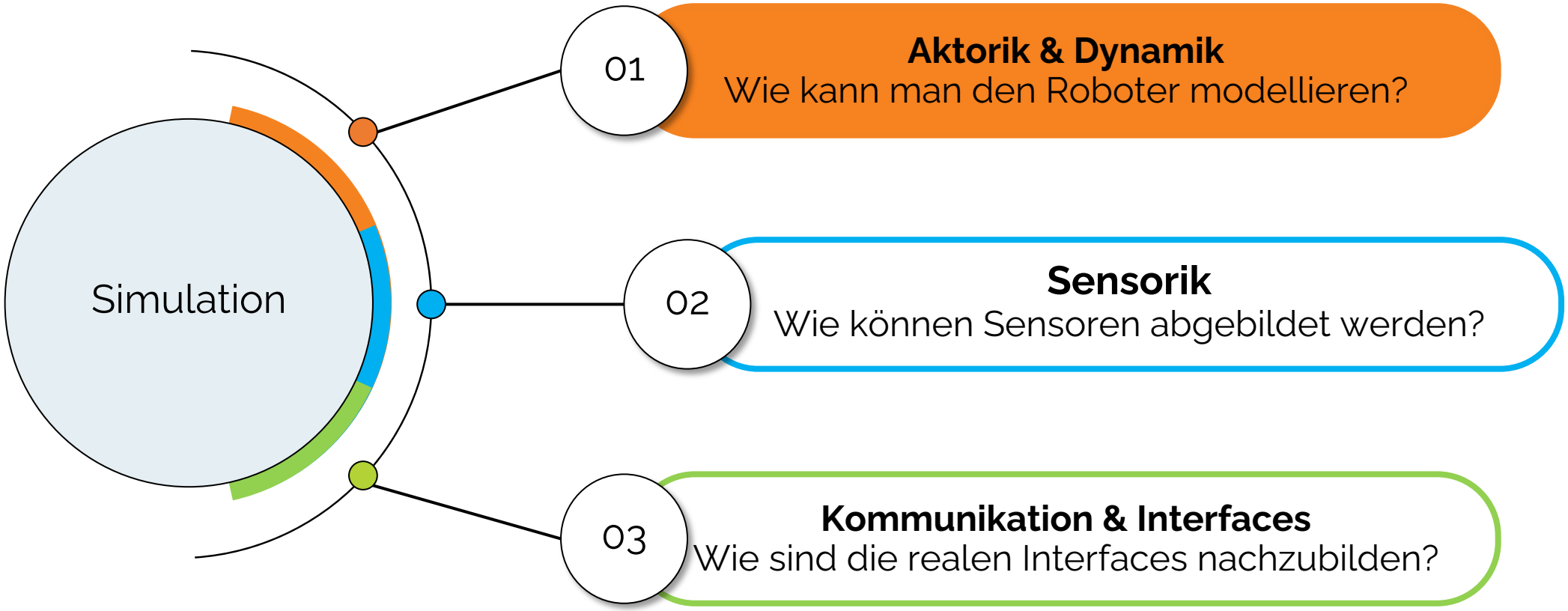
Eine Reise von hochdynamischen omnidirektionalen realen Robotern …

… zu hochdynamischen omnidirektionalen Simulationsmodellen!

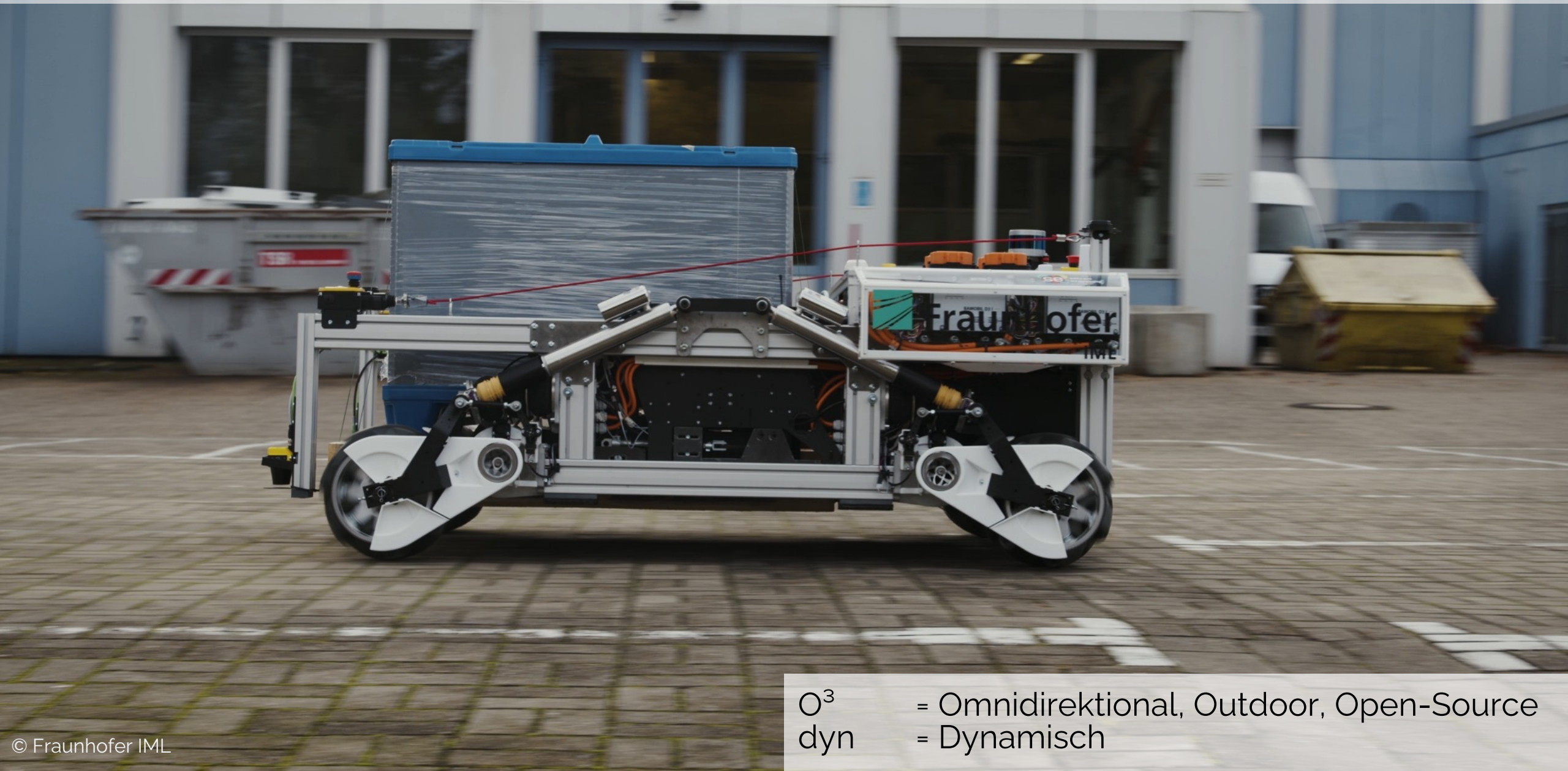
Agenda



Agenda



Der Demonstrator: O³dyn – die nächste Generation von Logistikrobotern



O³ = Omnidirektional, Outdoor, Open-Source
dyn = Dynamisch

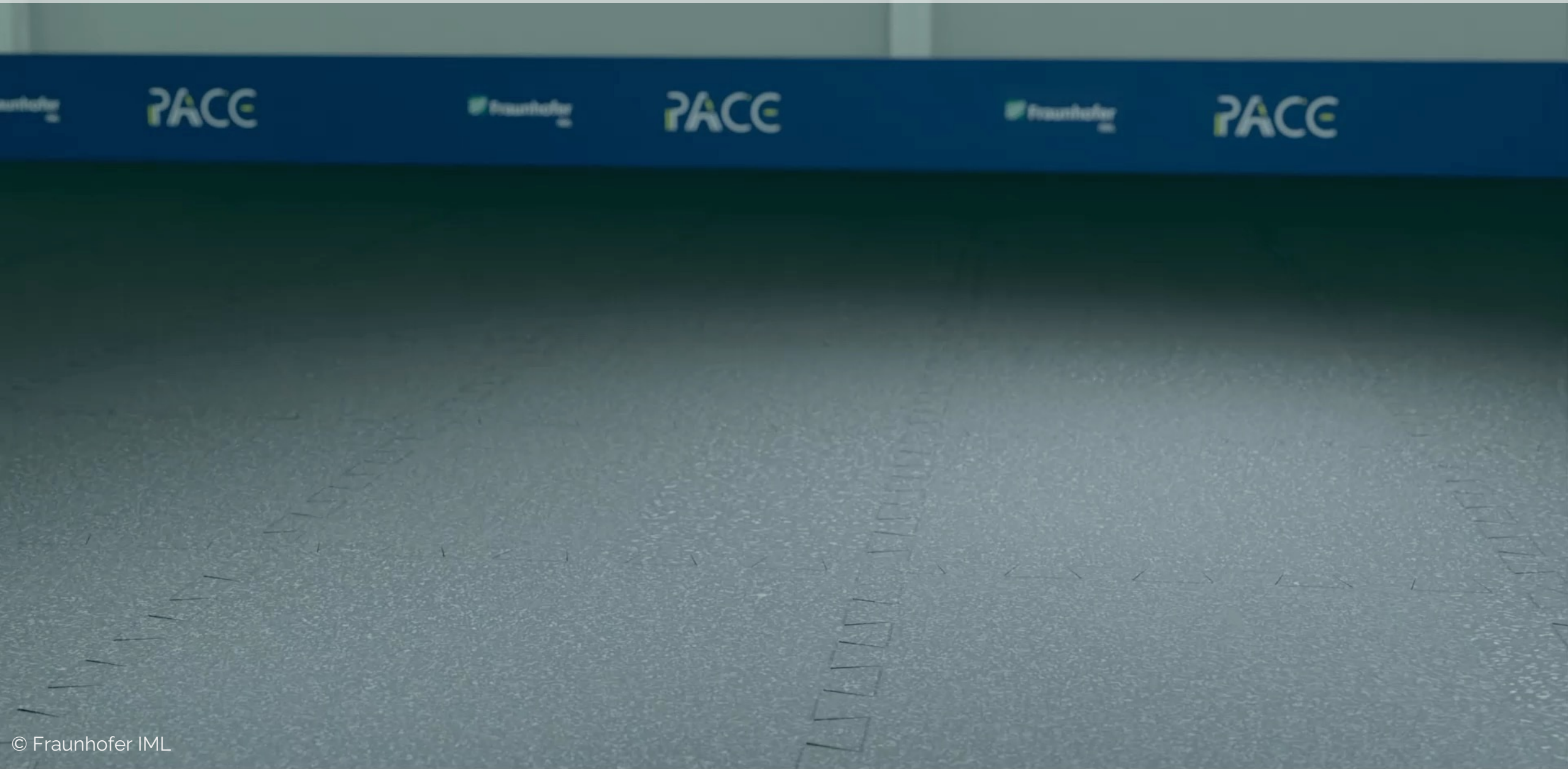
Die Reise beginnt: Import des Roboters

CAD-Import

URDF-Import

```
<!-- Dimension XSENS IMU -->
<xacro:property name="IMU_XSENS_dimension_x" value="0.0031" />
<xacro:property name="IMU_XSENS_dimension_y" value="0.0036" />
<xacro:property name="IMU_XSENS_dimension_z" value="0.0010" />


<!-- Dimensions of the Sick Tim 571 -->
<xacro:property name="sick_tim_dimension_x" value="0.01" />
<xacro:property name="sick_tim_dimension_y" value="0.01" />
<xacro:property name="sick_tim_dimension_z" value="0.01" />

<!-- Dimensions of the GPS Antenna -->
<xacro:property name="gps_antenna_dimension_x" value="0.01" />
<xacro:property name="gps_antenna_dimension_y" value="0.01" />
<xacro:property name="gps_antenna_dimension_z" value="0.01" />

<!-- Dimensions of the Realsense cameras -->
<xacro:property name="realsense_dimension_x" value="0.01" />
<xacro:property name="realsense_dimension_y" value="0.01" />
<xacro:property name="realsense_dimension_z" value="0.01" />


<!-- Include Robomaster Parts -->
<xacro:include filename="parts/base.xacro" />
<xacro:include filename="parts/laserscanners.xacro" />
<xacro:include filename="parts/imus.xacro" />
<xacro:include filename="parts/gps.xacro" />
<xacro:include filename="parts/wheels.xacro" />
<xacro:include filename="parts/gripper.xacro" />
<xacro:include filename="parts/cameras.xacro" />
```

Das Simulationsmodell

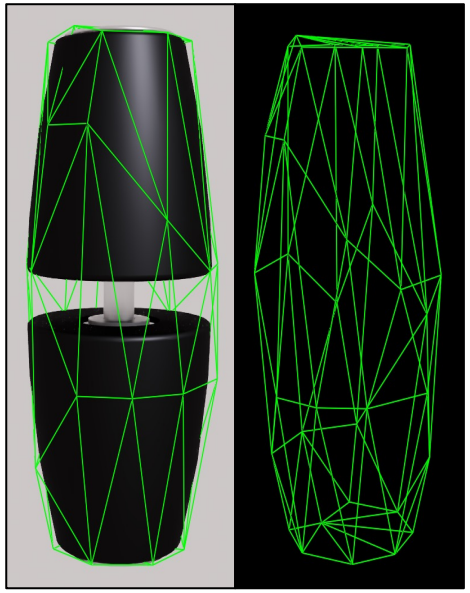


Die Bedeutung der Mecanum-Räder und Rollen

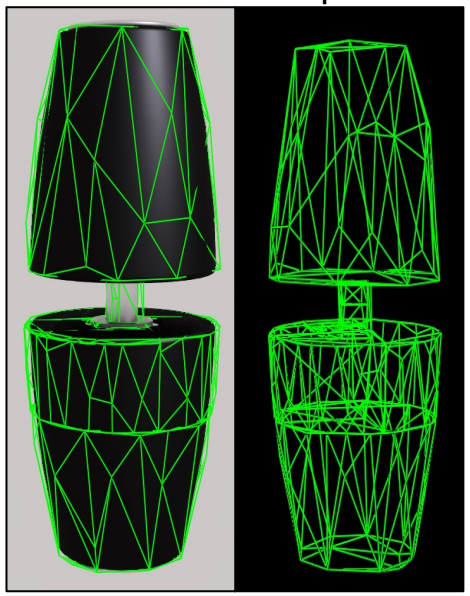
Modellierung der Rollen

Automatisch generierte Meshes

Convex Hull

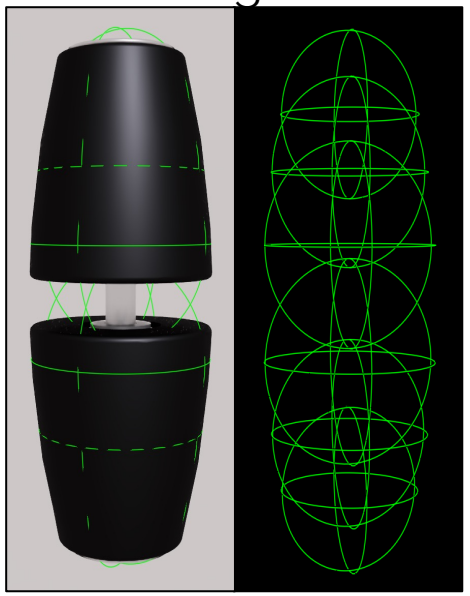


Convex Decomposition

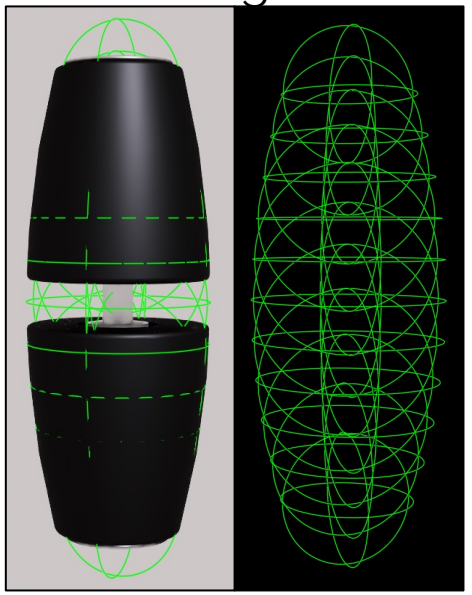


Eigene Collider Designs

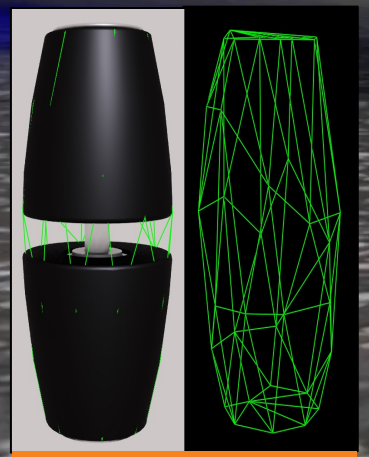
6 Kugeln



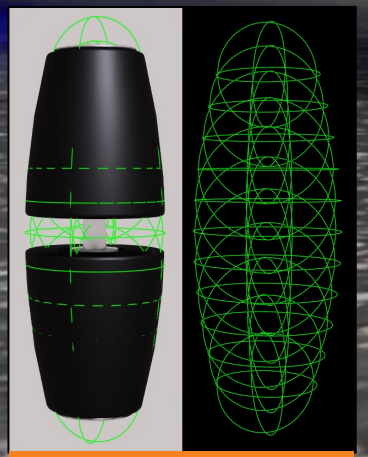
11 Kugeln



Modellierung der Rollen

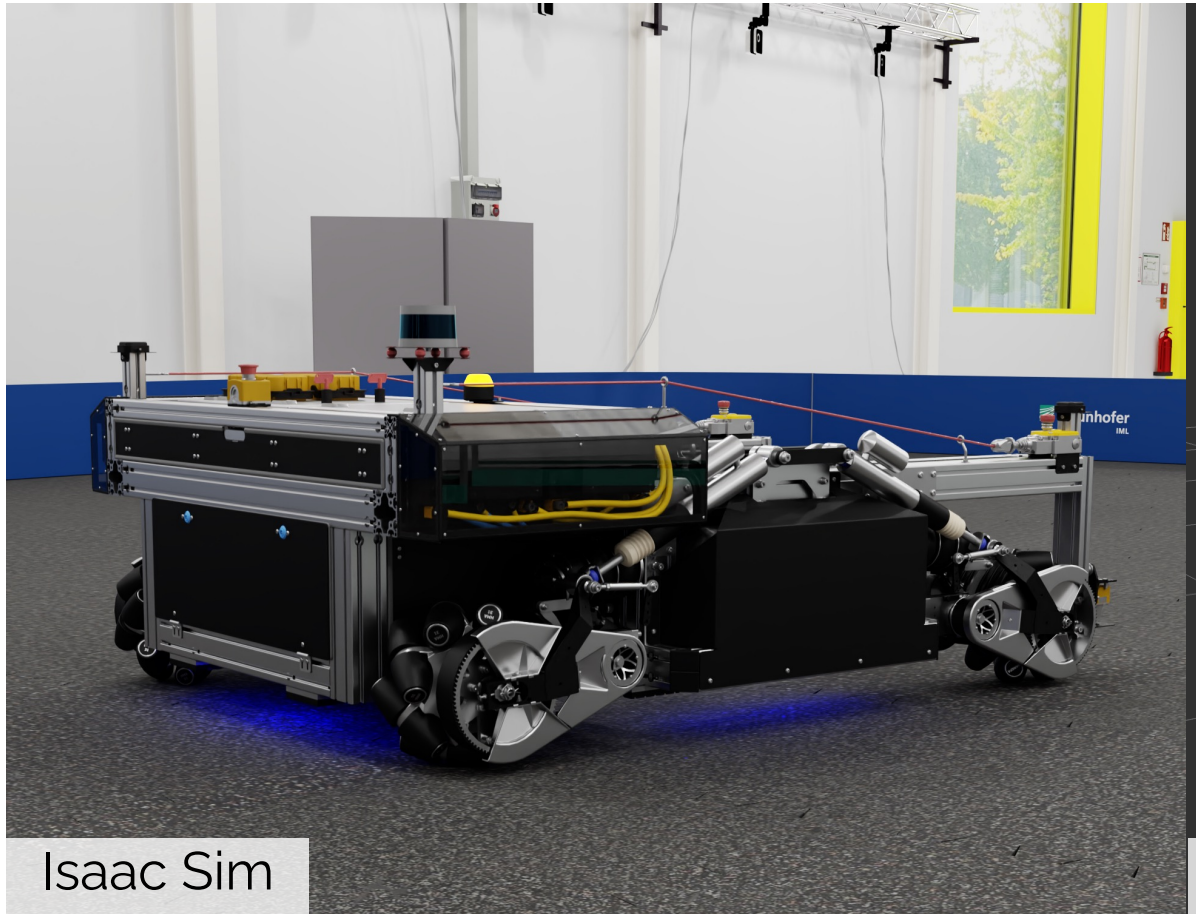


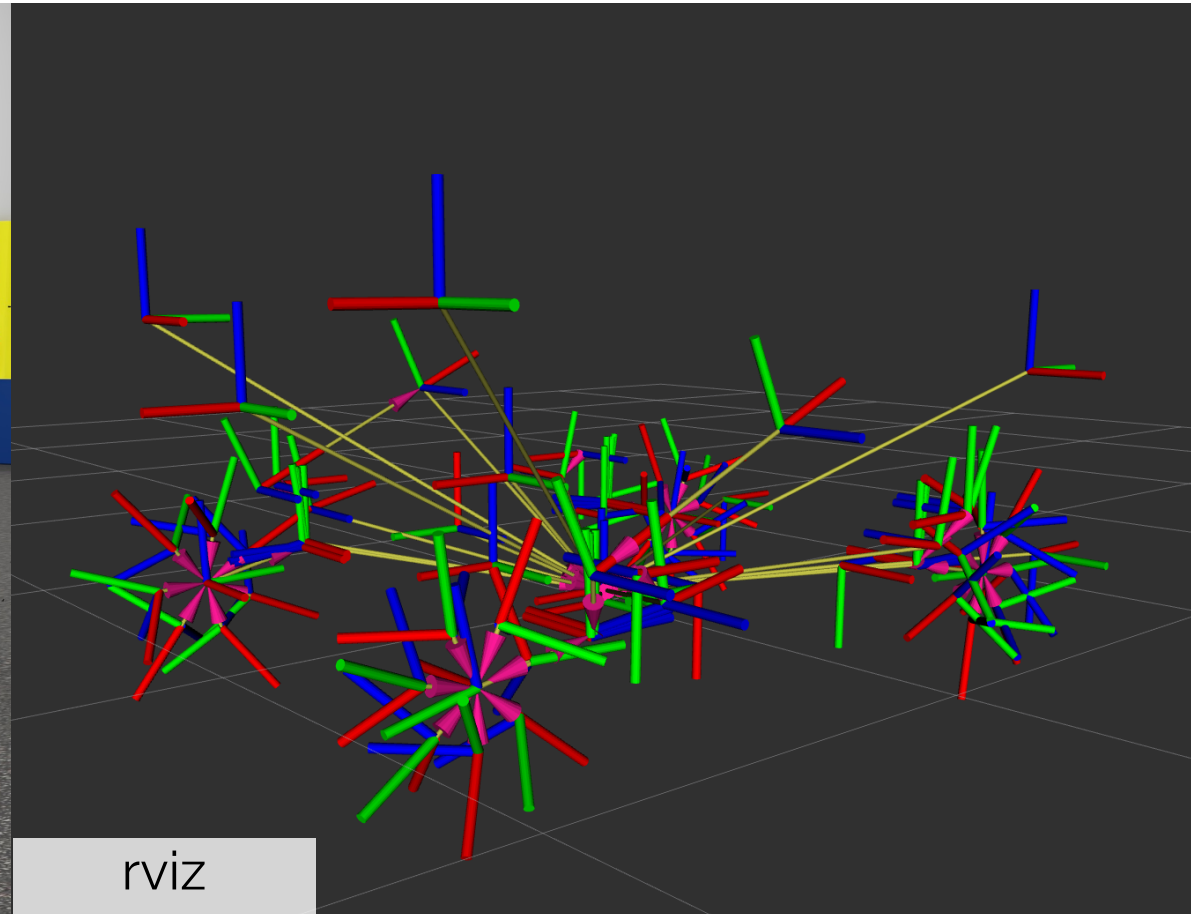
Convex Hull



11 Kugeln

Endlich! Ab in das ROS-Ökosystem

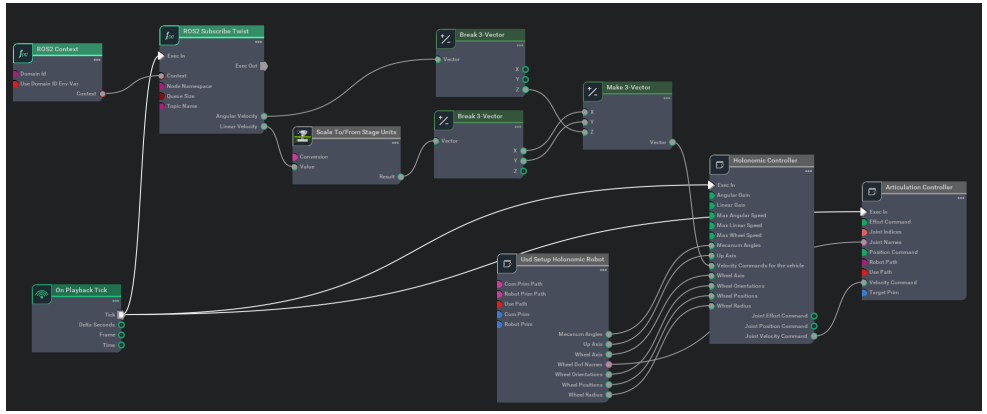




Kontrolle benötigt: Der Base Controller

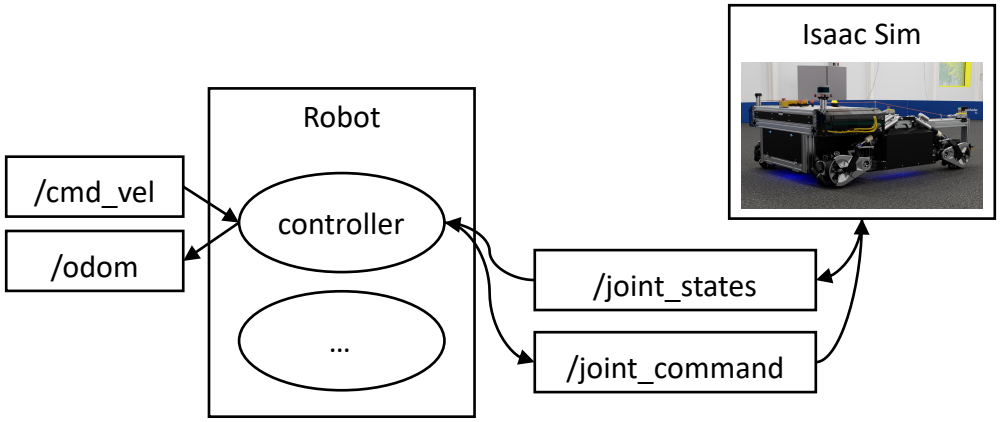
OmniGraph-Umsetzung

- Direkte Implementierung in Isaac Sim
- Visuelle Skriptsprache OmniGraph
- Beim Laden des Modells direkt aktiv

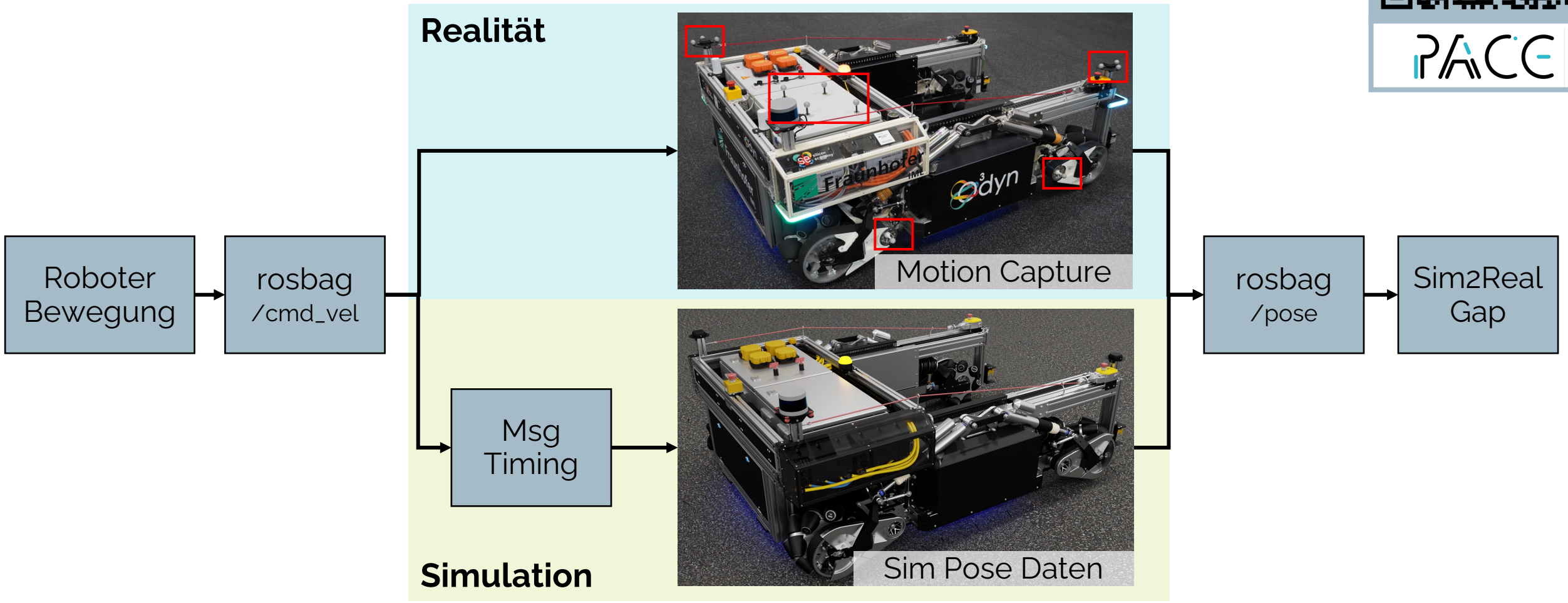


Eigener Controller

- Implementierung via ROS oder Python
- Eigenständiger Knoten
- Umsetzung spezifischer Systemverhalten

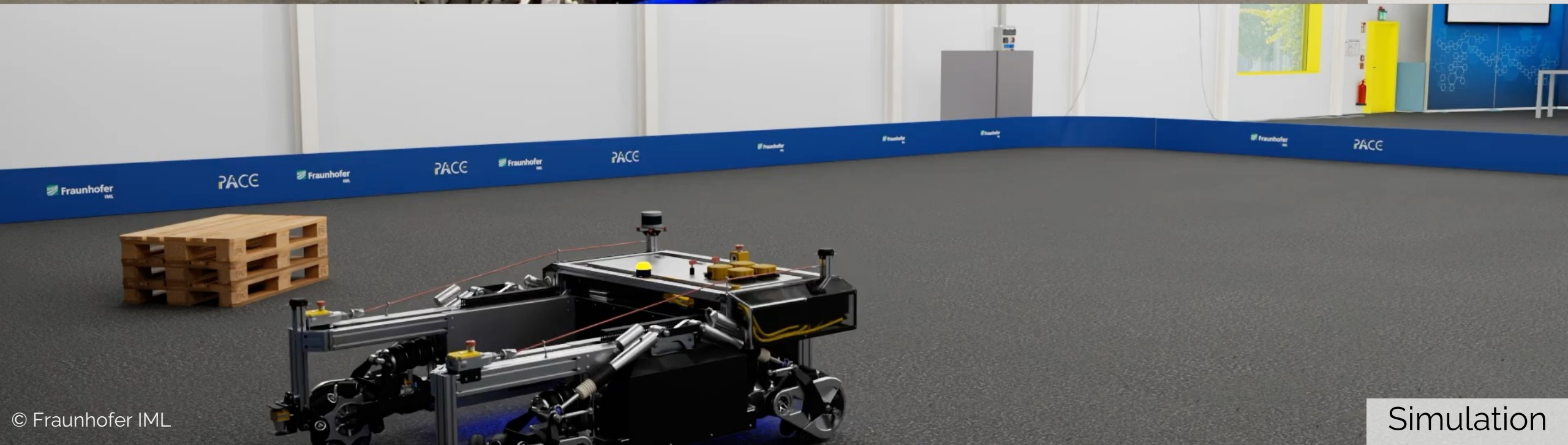


Sim2Real Vergleich – Verhalten die sich auch gleich?



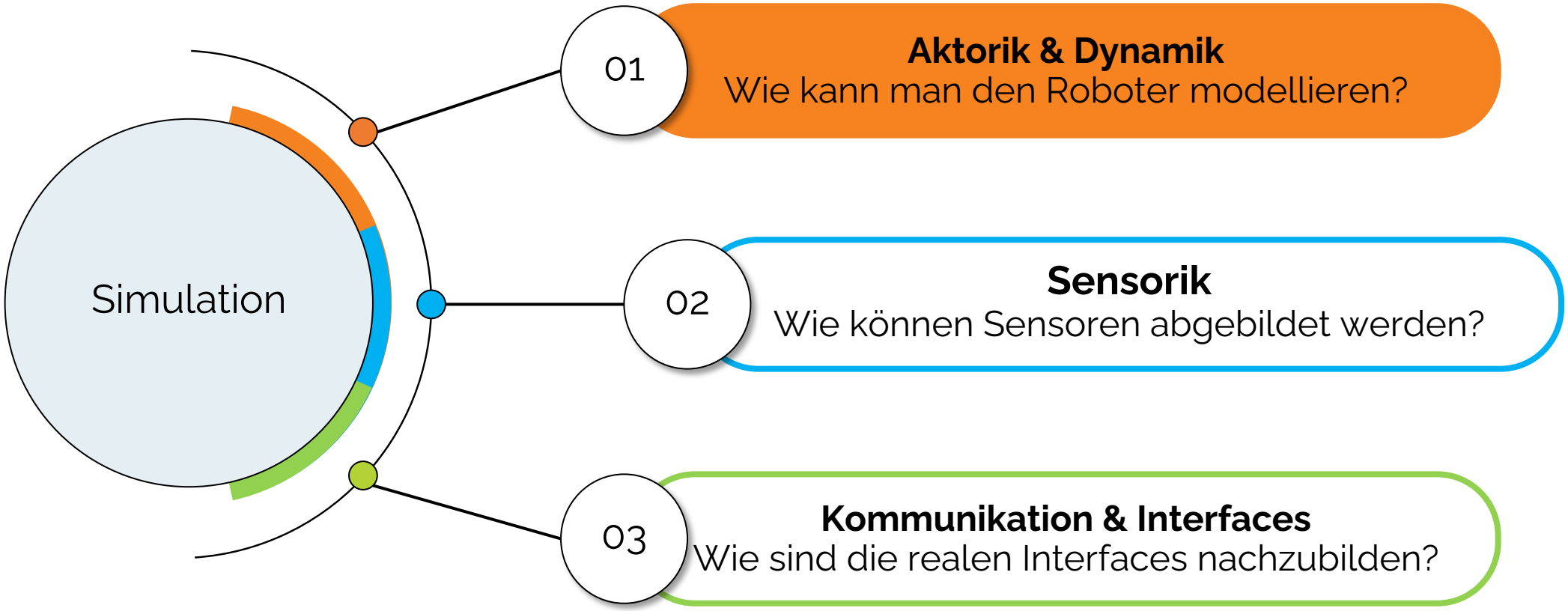
Sim2Real Vergleich – Verhalten die sich auch gleich?

Realität

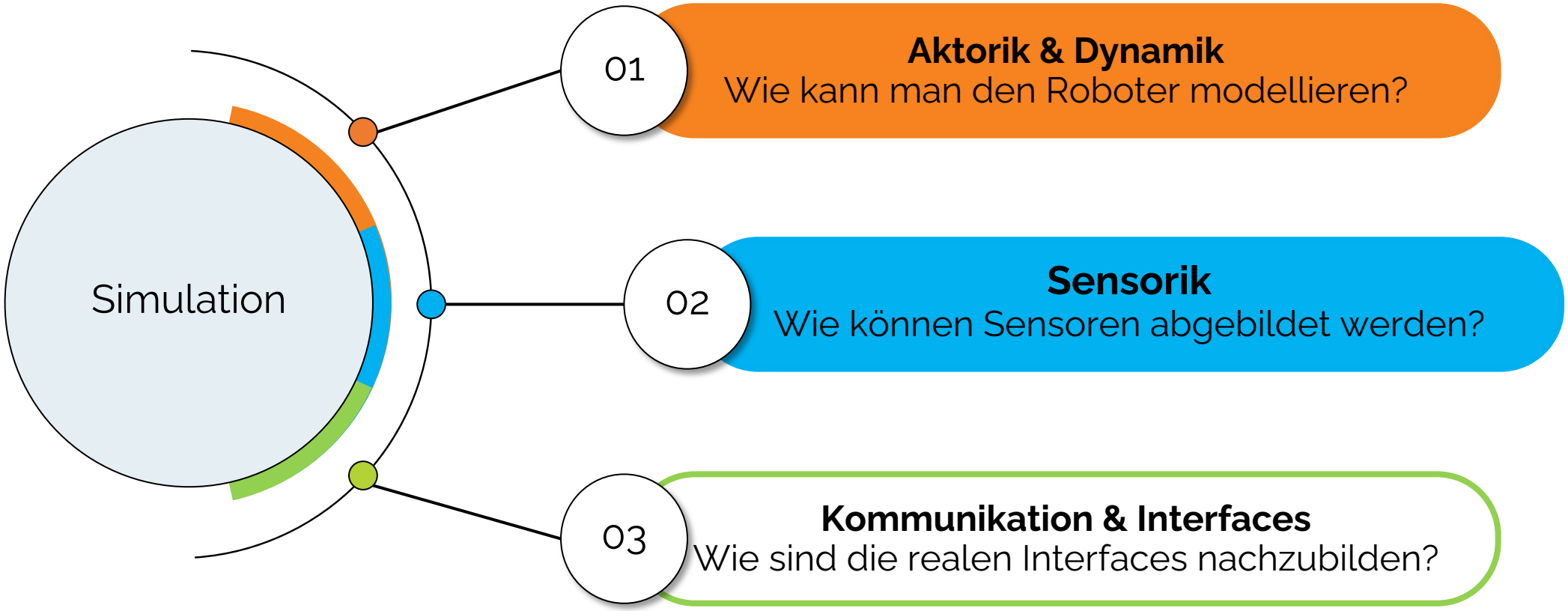


Simulation

Agenda



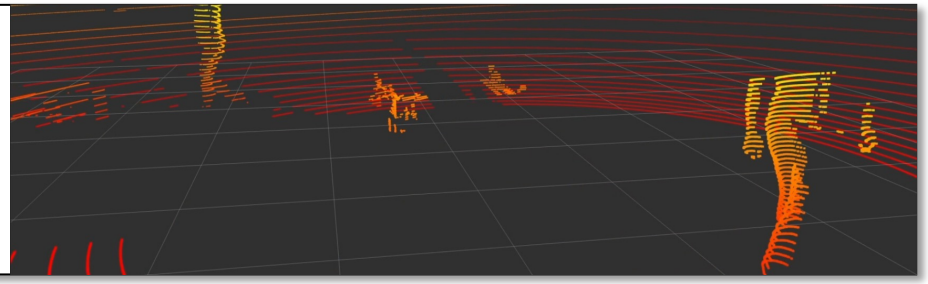
Agenda



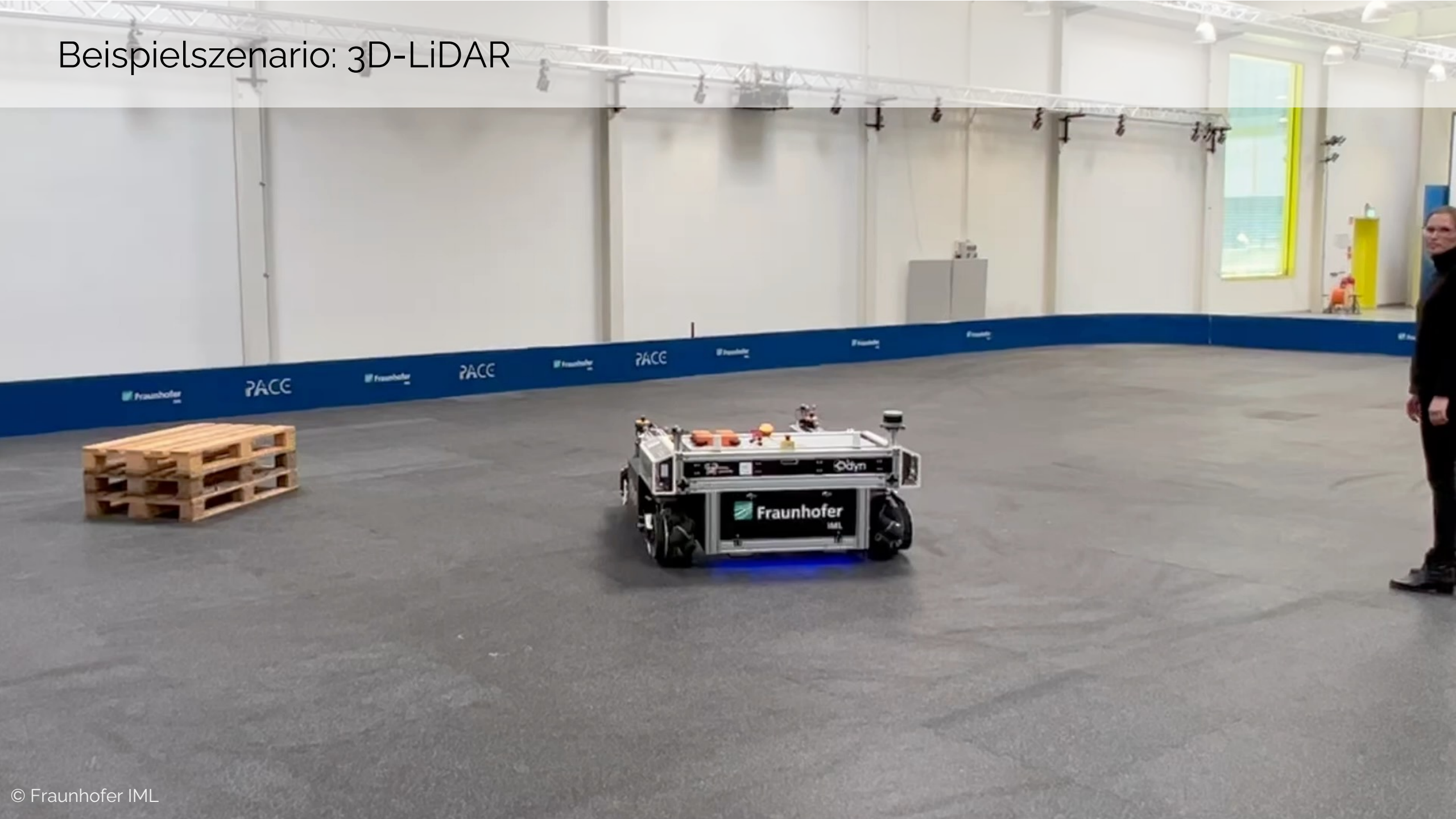
Sensorik

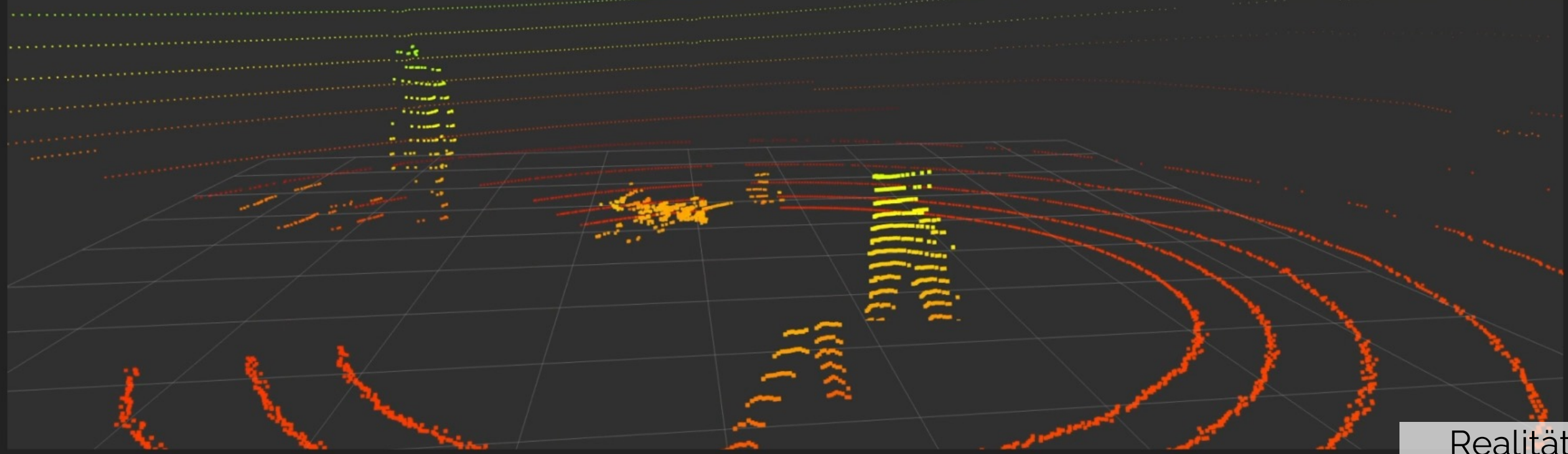
Jetzt geht's ans Wahrnehmen: Verfügbare Sensoren

Kamera	• RGB-Kamera • Tiefenkamera • Replicator zur Erzeugung synthetischer Daten
LiDAR	• PhysX-basierter LiDAR • RTX-basierter LiDAR
Sonstige	• PhysX-basierter Entfernungssensor (generisch, Ultraschall) • Kraft- und Kontaktsensoren • IMU • Näherungssensor

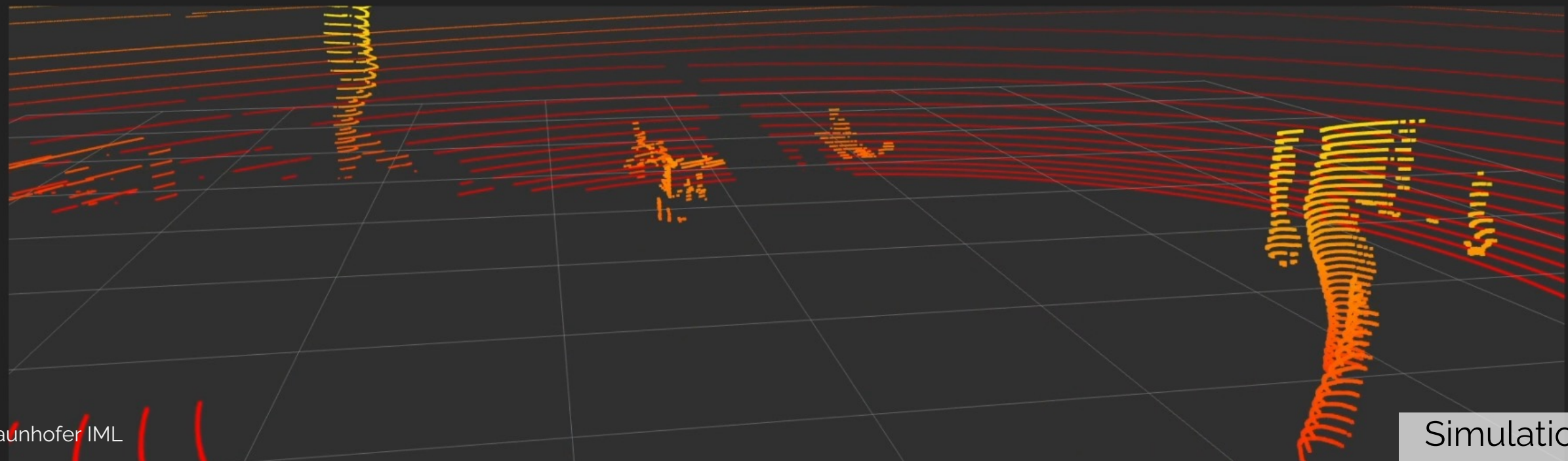


Beispielszenario: 3D-LiDAR



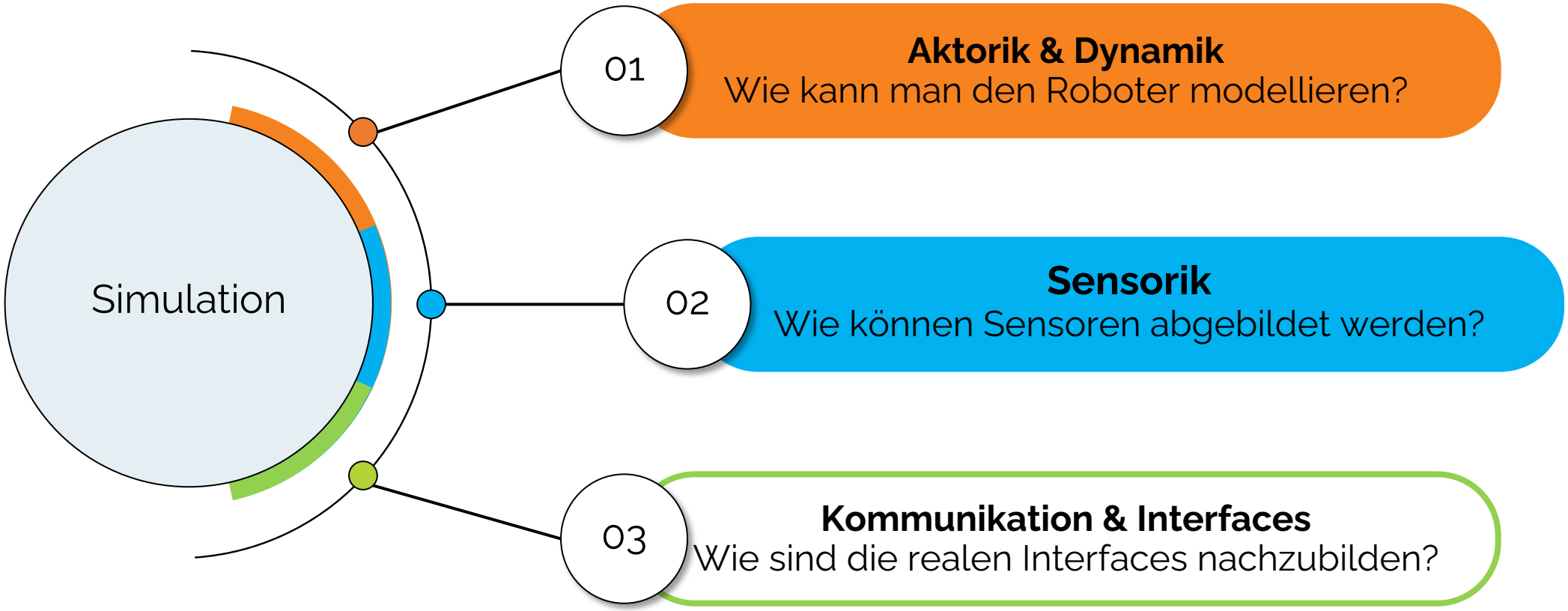


Realität

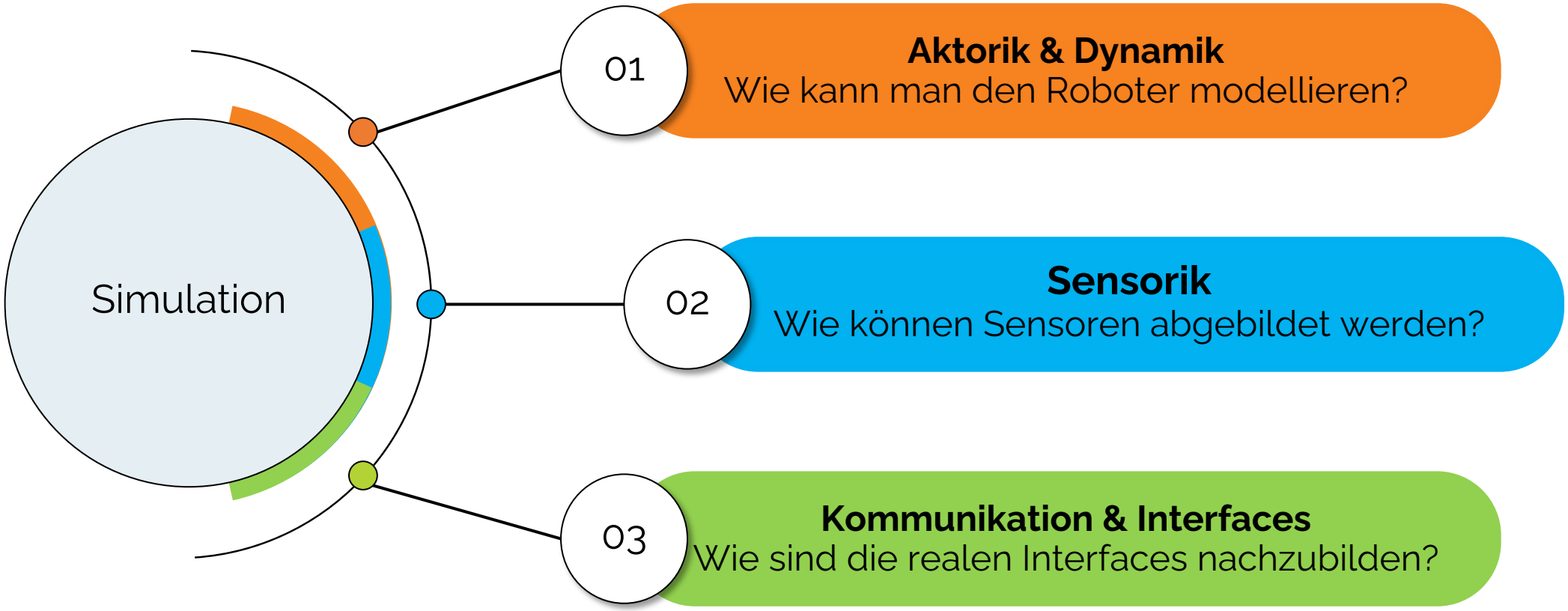


Simulation

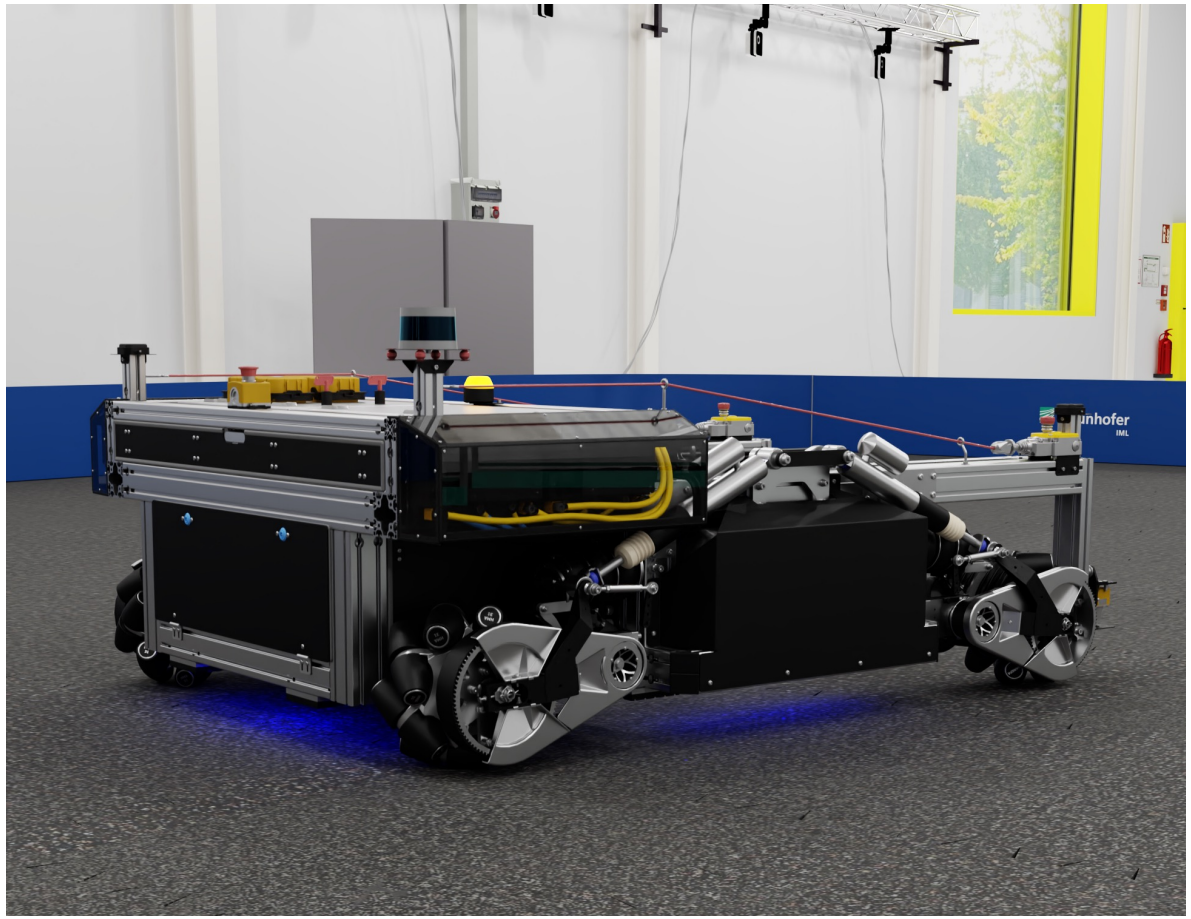
Agenda



Agenda

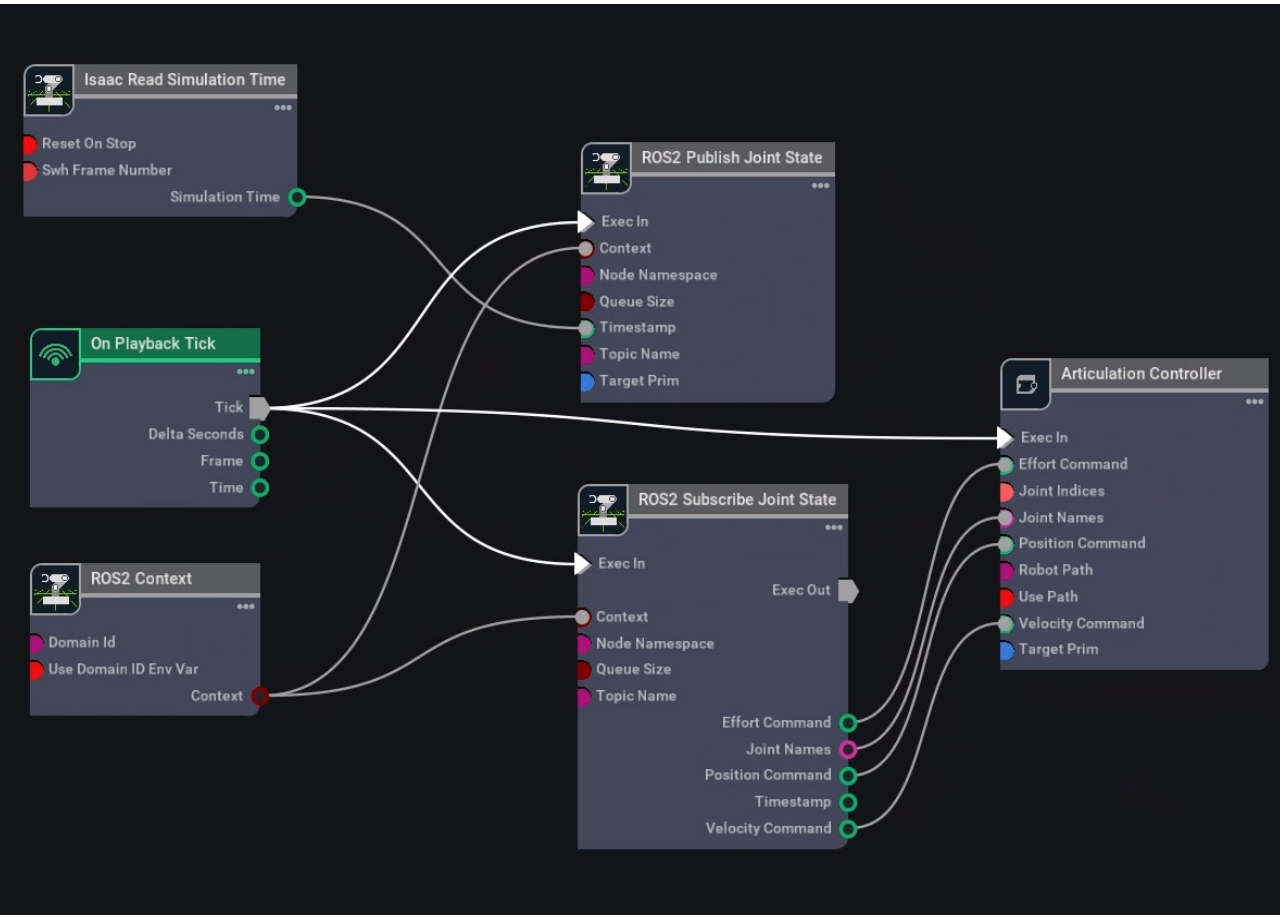


Kommunikation & Interfaces
Topics



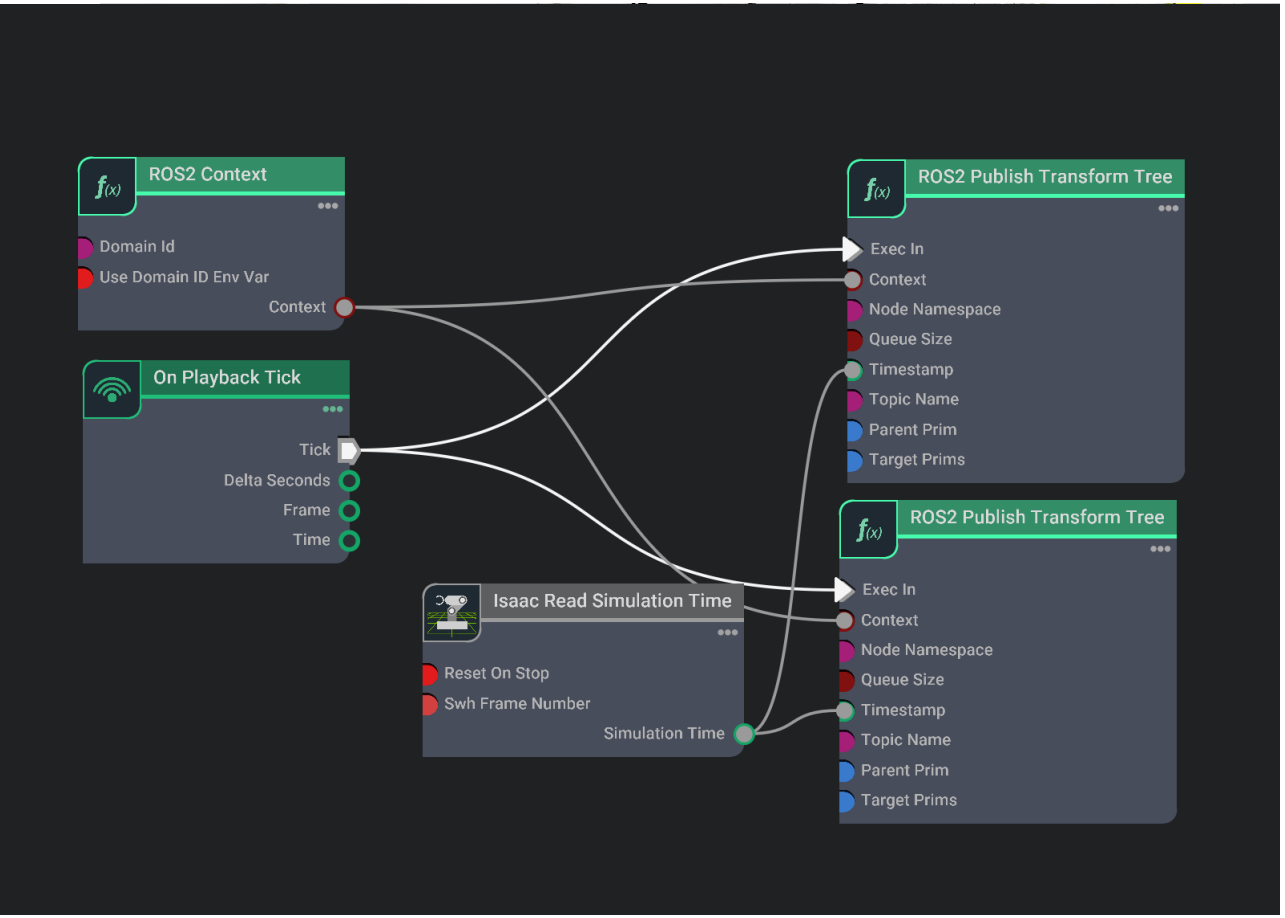
Base Controller	/cmd_vel, /odom
Joints	/joint_state, /joint_command
TF	/tf, /tf_static
Sensorik	/laser, /point_cloud
Custom Msgs	/battery_soc, …

Kommunikation & Interfaces
Topics



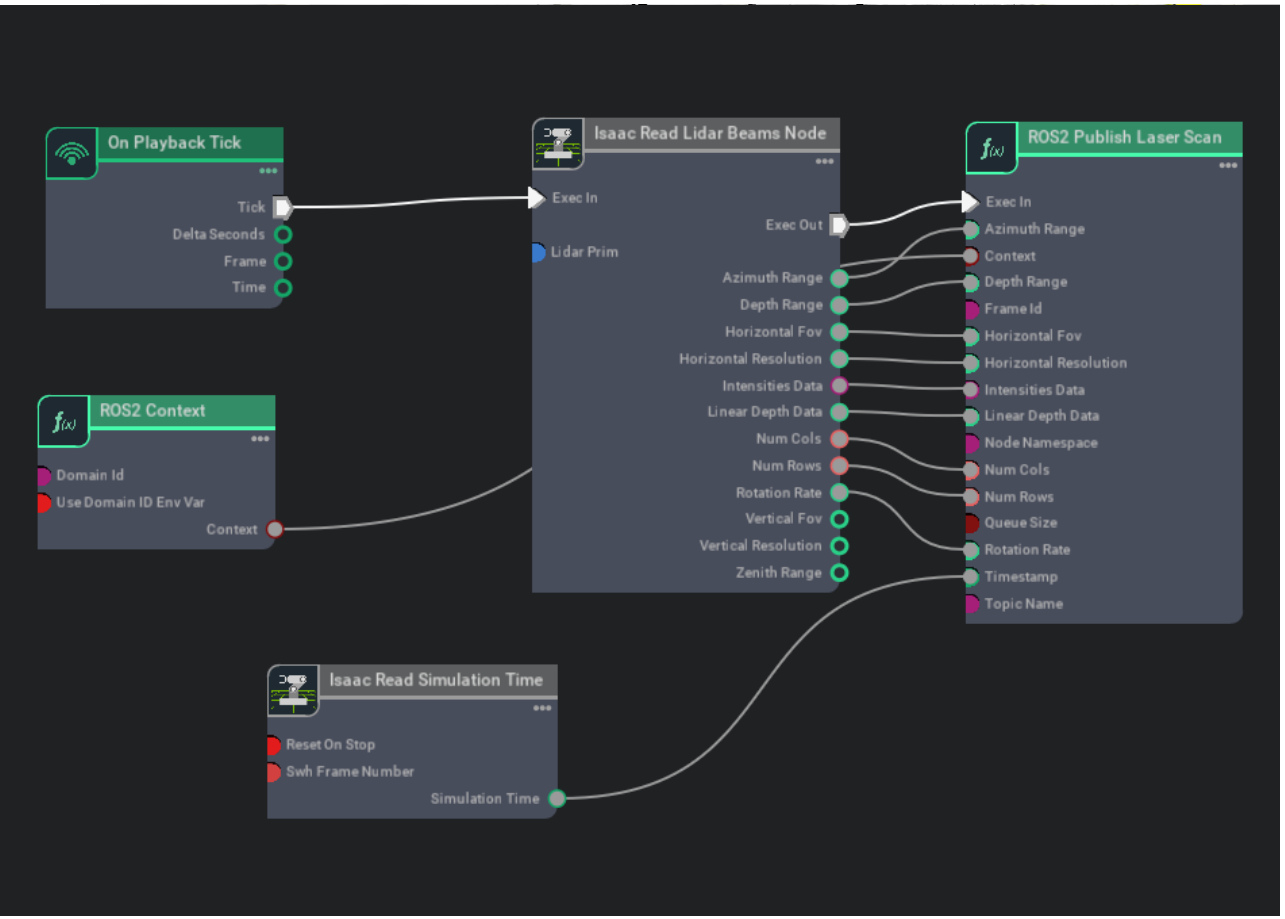
Base Controller	/cmd_vel, /odom
Joints	/joint_state, /joint_command
TF	/tf, /tf_static
Sensorik	/laser, /point_cloud
Custom Msgs	/battery_soc, …

Kommunikation & Interfaces
Topics



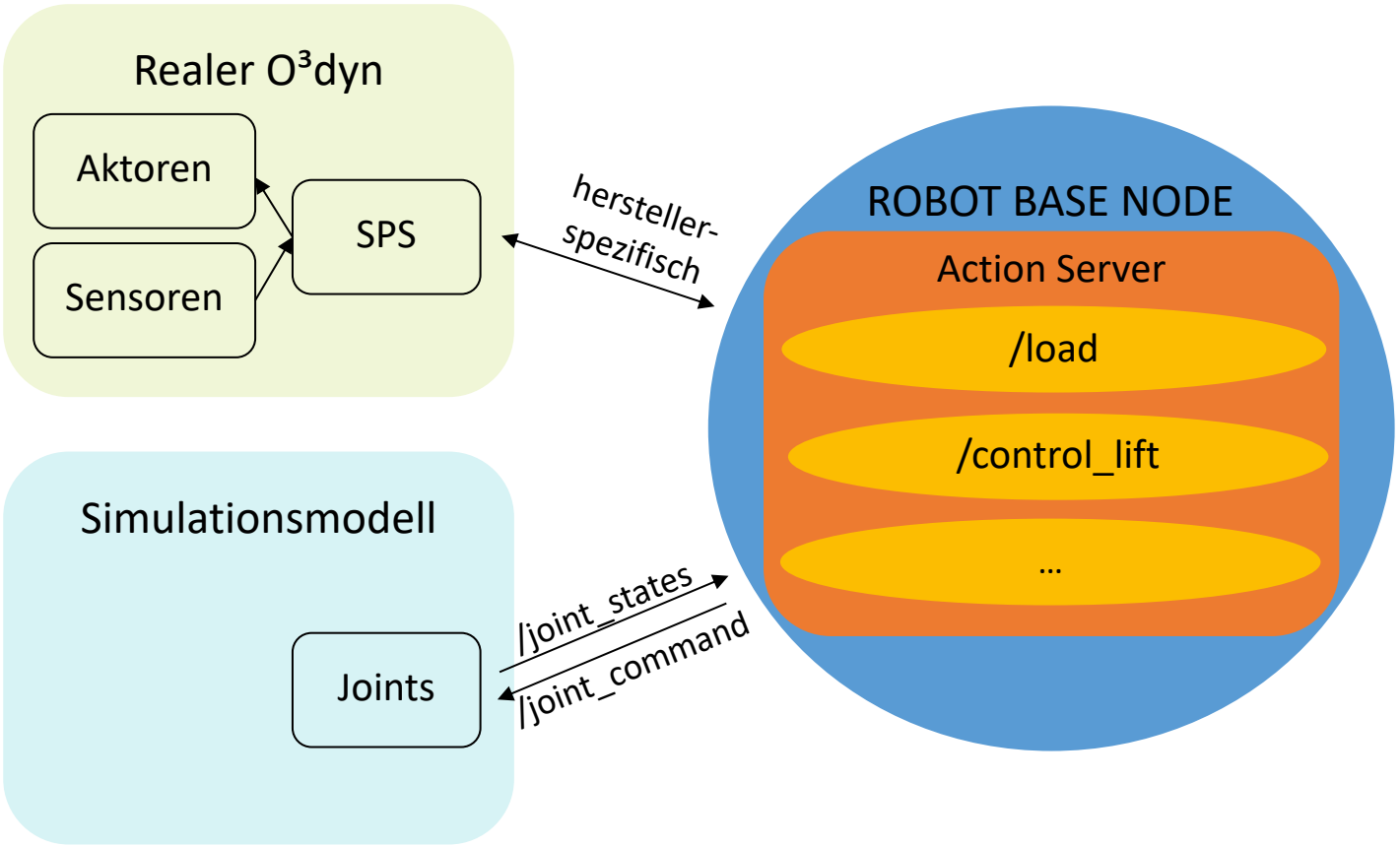
Base Controller	/cmd_vel, /odom
Joints	/joint_state, /joint_command
TF	/tf, /tf_static
Sensorik	/laser, /point_cloud
Custom Msgs	/battery_soc, …

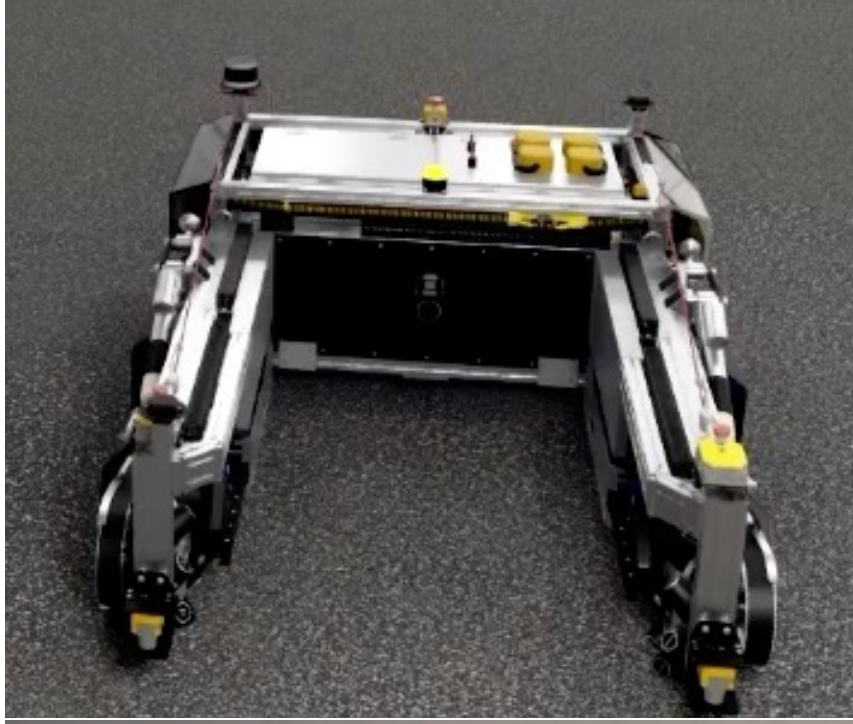
Kommunikation & Interfaces
Topics



Base Controller	/cmd_vel, /odom
Joints	/joint_state, /joint_command
TF	/tf, /tf_static
Sensorik	/laser, /point_cloud
Custom Msgs	/battery_soc, …

Kommunikation & Interfaces
Actions und Services





Zusammenfassung

Simulation hochdynamischer Roboter ist möglich. Aber Achtung bei der Rollen-Modellierung & nutze Realdaten.

Isaac Sim bietet viele Möglichkeiten für die Robotik mit Stärken im Rendering und der Sensor-Simulation.

ROS bildet die Brücke zwischen Simulation und Realität.

Einordnung NVIDIA Isaac Sim

Vorteile

GPU-Möglichkeiten und Renderer
Fortschrittliche Sensorsimulation inklusiv synthetischer Daten für Machine Learning.

Fortschrittliche Physiksimulation
Abbildung komplexer Rigid-Body Simulation, z.B. bei Mecanum-Rädern.

Omniverse Ökosystem
Kombination von Tools & Anwendungen: Robotik, SDG, RL, Menschensimulation, …

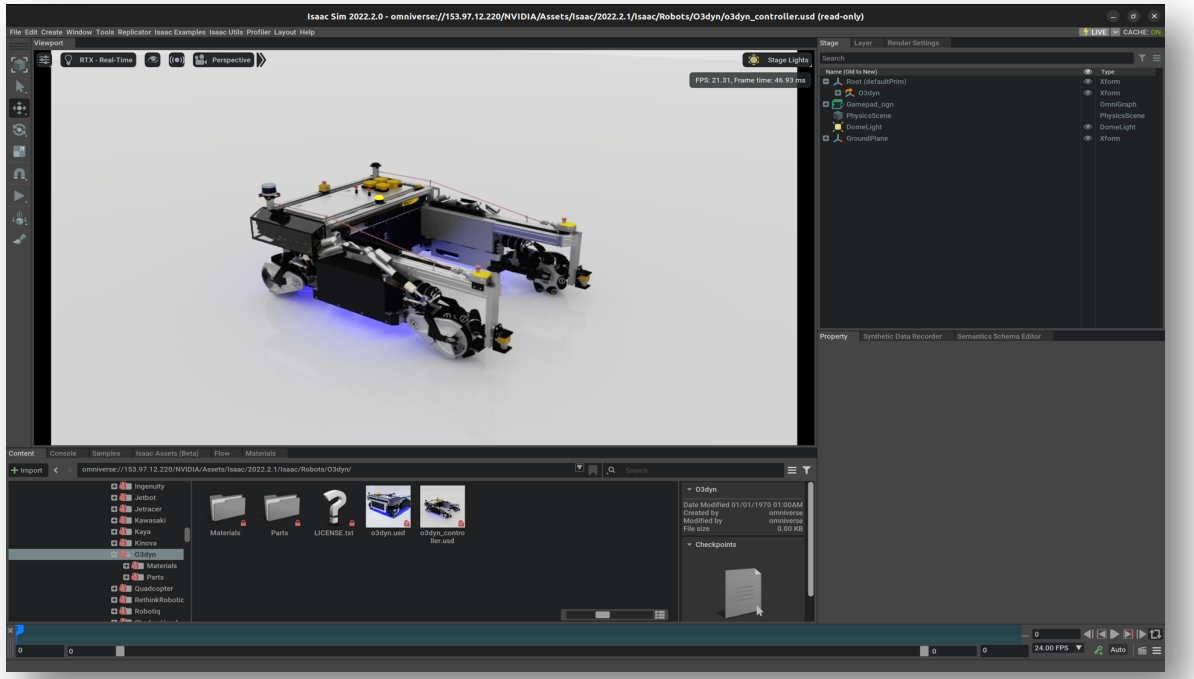
Nachteile

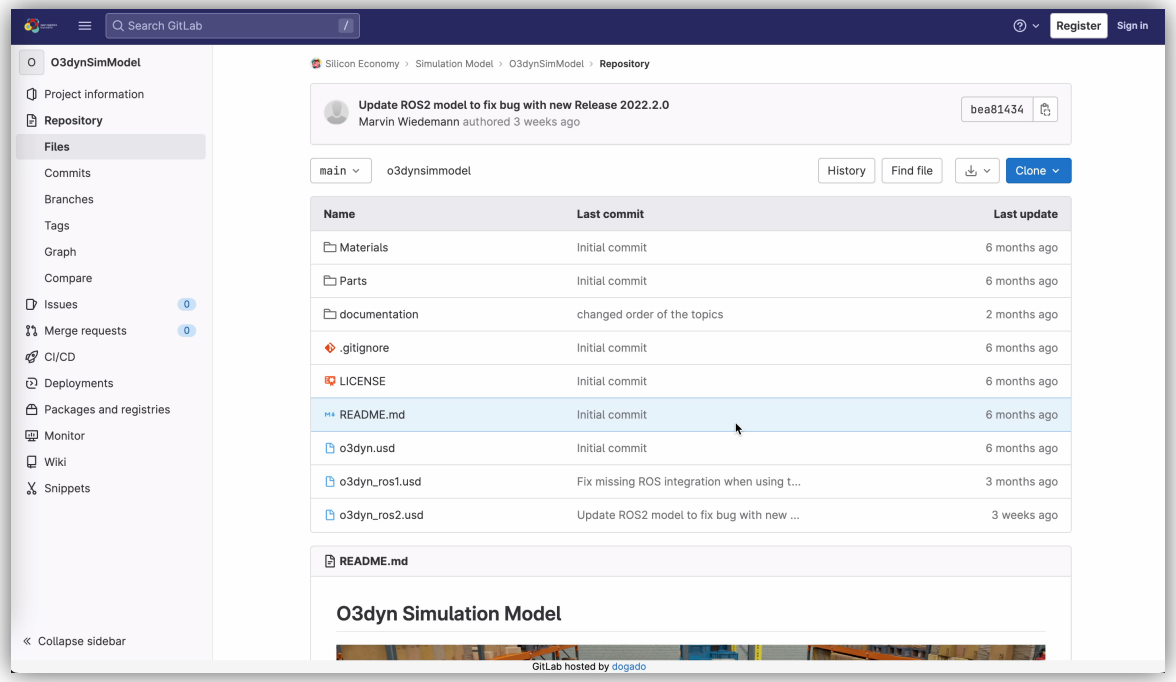
Hohe Systemanforderungen
NVIDIA RTX GPUs (min: RTX 2070)
Intel i7 oder Ryzen 5, 32 GB RAM.

Skalierbarkeit
Skalierung komplexer Robotermodelle aktuell nur eingeschränkt möglich.

Proprietär
Entwicklung von NVIDIA.
Open-Source Ansätze, aber nicht für Alles.

Interesse an O³dyn? Hol dir das Simulationsmodell nach Hause!





O³dyn auf deinem Nucleus

omniverse://localhost/NVIDIA/Assets/Isaac/2023.1.0/Isaac/Robots/O3dyn

Open Source @ Open Logistics Foundation

https://git.openlogisticsfoundation.org/silicon-economy/simulation-model/o3dynsimmodel/

Kontakt

Marvin Wiedemann
AI and Autonomous Systems
marvin.wiedemann@iml.fraunhofer.de

Danke an...
Ossama Ahmed[1], Renato Gasoto[2], Anna Vasileva[1], Elisabeth Drache[1], Steffen Daniel[1], Mrunal Hatwar[1], Jan Finke[1], and Sören Kerner[1]
[1] Fraunhofer IML; [2] Nvidia

Fraunhofer IML

Fraunhofer Institute for Material Flow and Logistics

Digitale Visitenkarte

Linkedin

O^3dyn Modell

Gefördert durch

silicon economy logistics ecosystem

LAMARR INSTITUTE FOR MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE