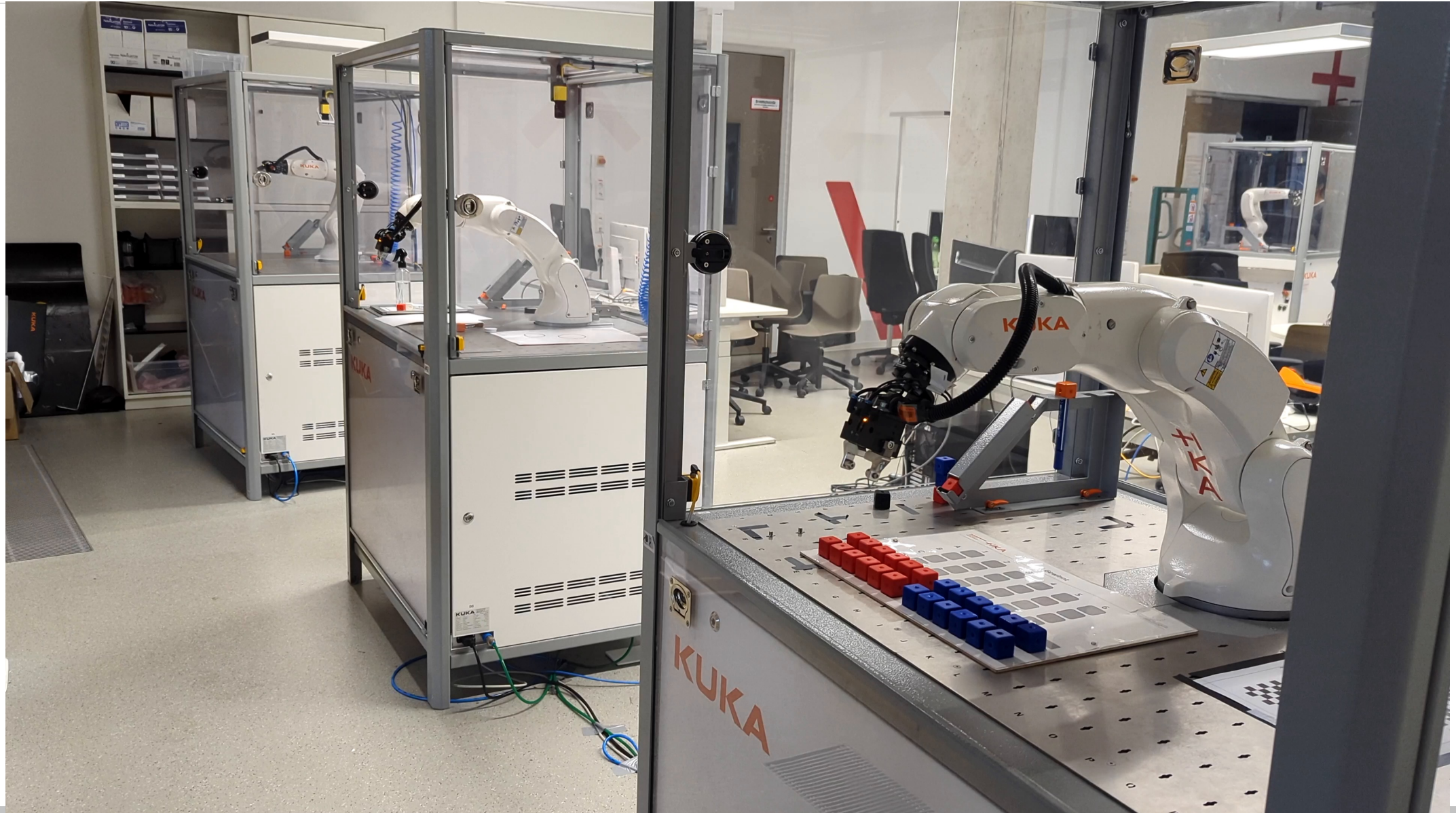

ros2_control ist aus meinem Rechner ausgebrochen

ROSCON 2023

Manuel Muth M.Sc.
Dr. Denis Štogl

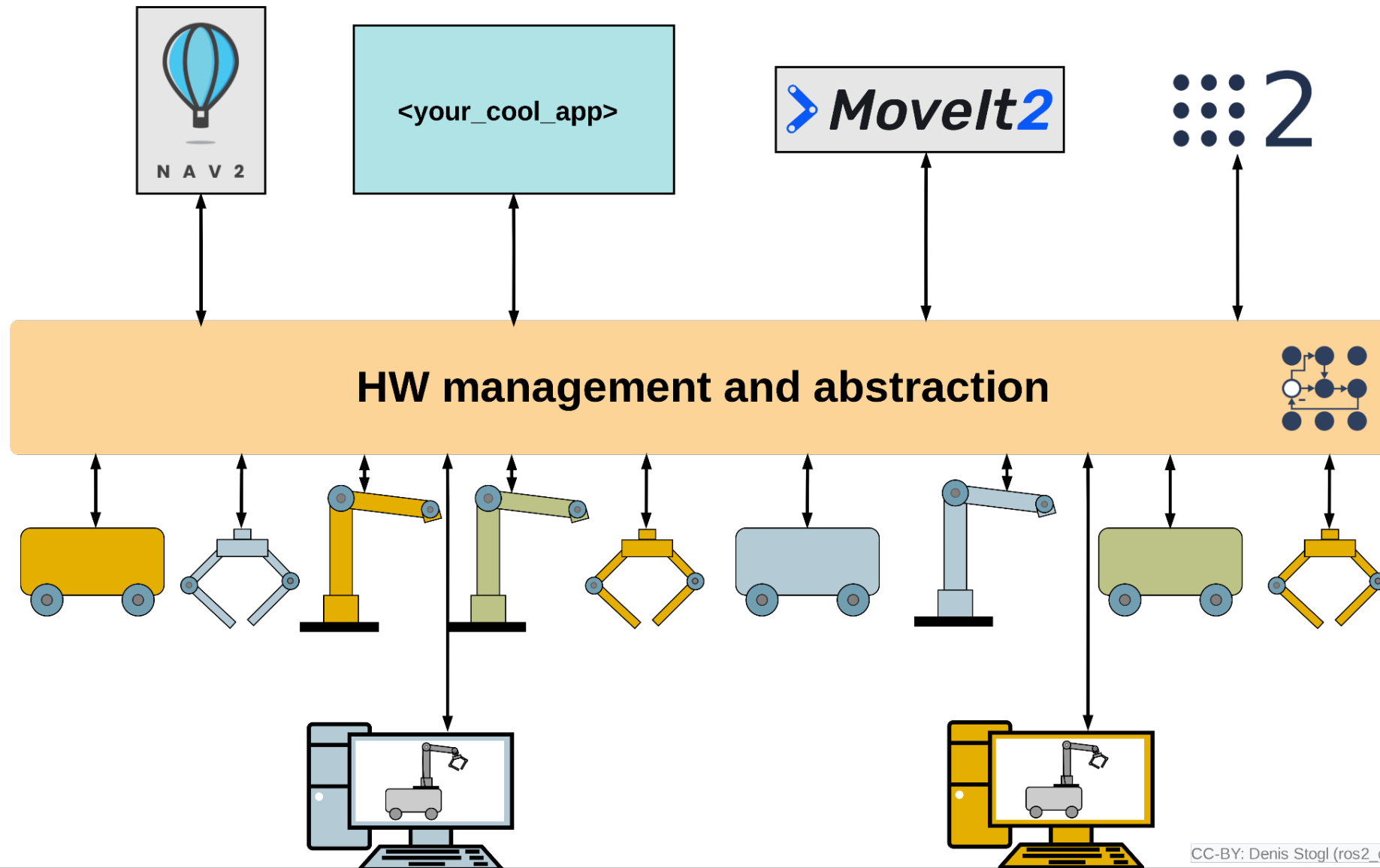
Evaluation



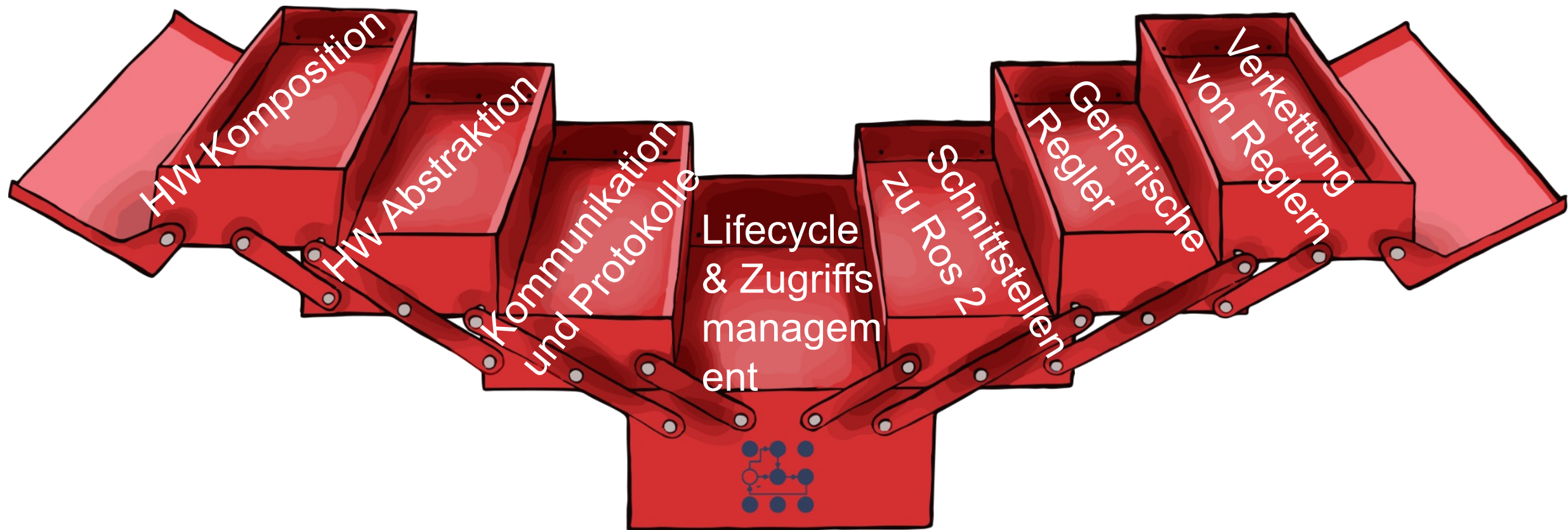
- Professionelle Integration und Entwicklung von ROS and ROS 2 Robotikanwendungen
 - Spezialisiert auf HW control und Integration von ROS
- Dr.-Ing. Denis Štogl
 - ros2_control maintainer, Robotik Experte
- Manuel Muth
Robotik Ingenieur
- 5 Vollzeit Ingenieure
- 2 Praktikanten



ros2_control - Kernel für ROS2

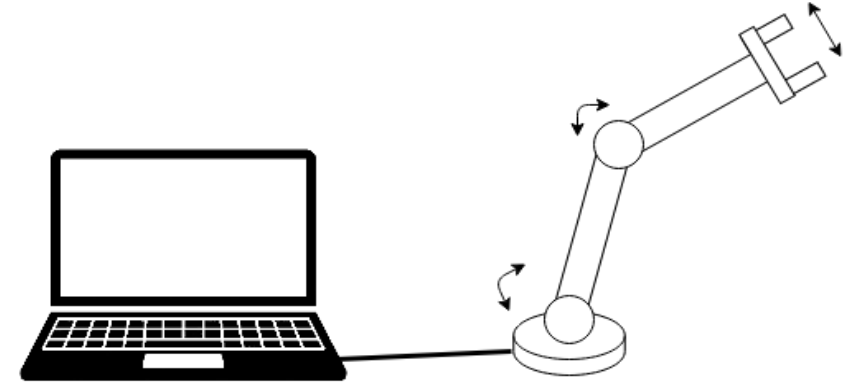


ros2_control - Kernel für ROS2



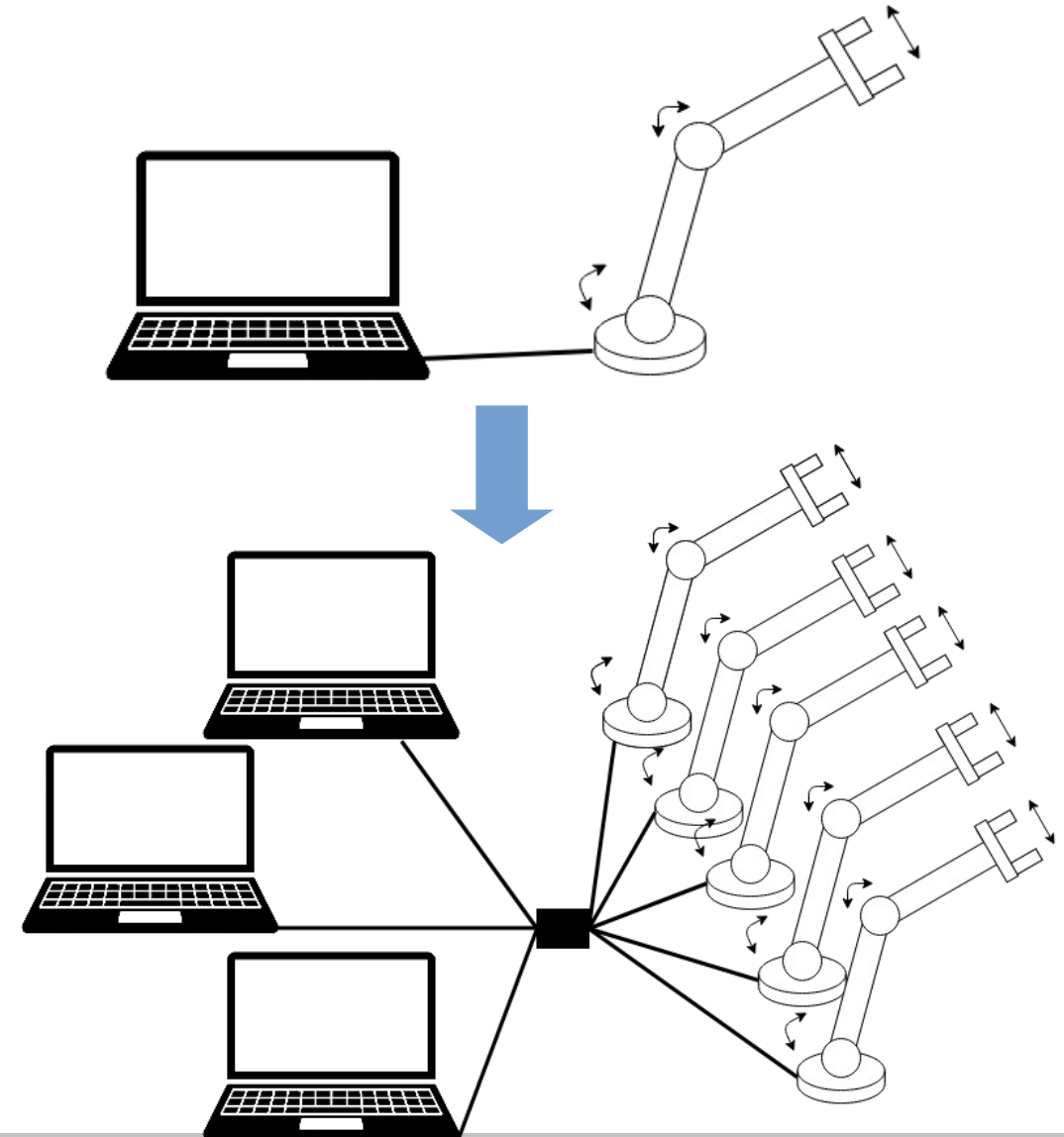
ros2_control - Architektur

- Architektur gedacht für Regelung:
 - Eines einzelnen Roboters
 - Einer kleinen Zelle



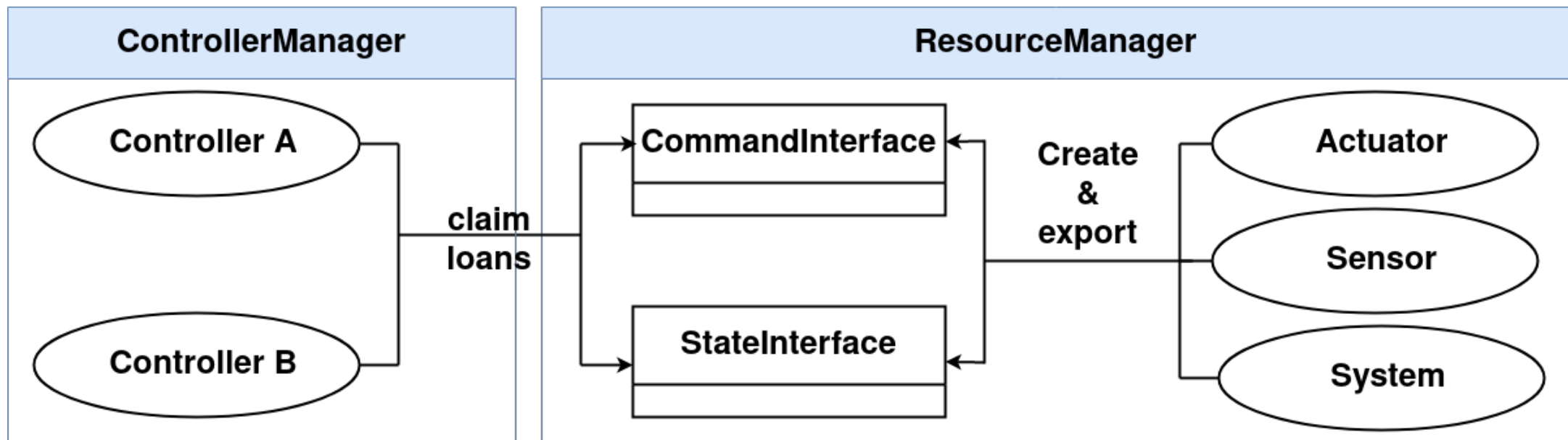
ros2_control - Architektur

- Architektur gedacht für Regelung:
 - Eines einzelnen Roboters
 - Einer kleinen Zelle
- ROS 2 baut auf DDS auf
 - ➔ ermöglicht verteilte Systeme
 - ➔ Regelung mehrerer Roboter
 - ➔ Potential für Verbesserung



ros2_control - Architektur

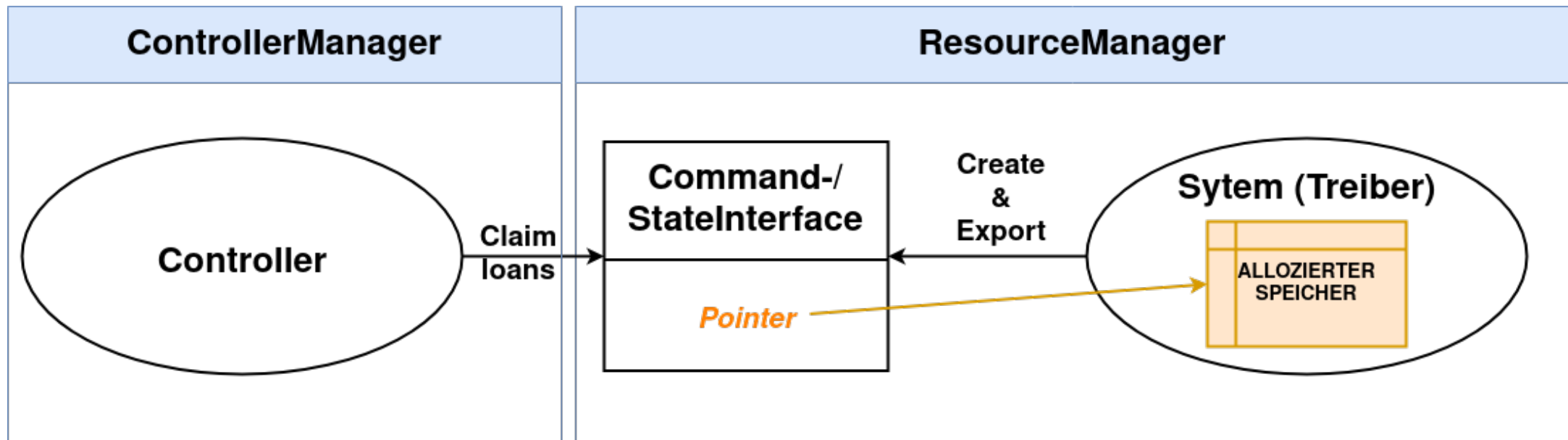
- ControllerManager → Management der Regler
- ResourceManager → Management der Hardware Komponenten



- Command-/StateInterfaces auch Handle genannt

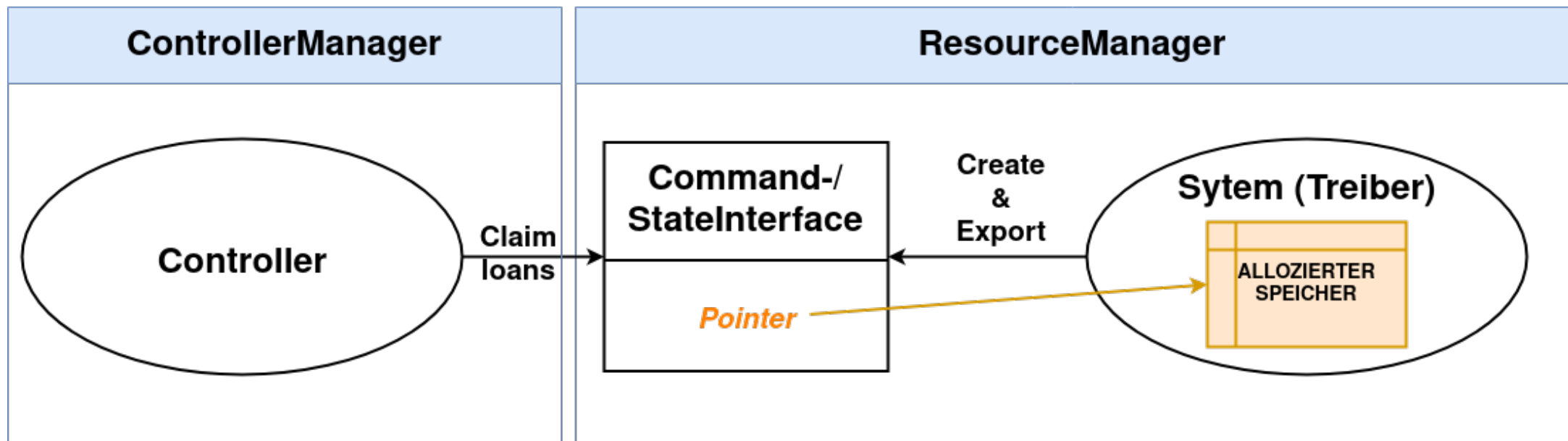
ros2_control - Architektur

- **Shared-Memory** Architektur
 - Hardware alloziert Speicher
 - Regler erhält über Handle Zugriff auf Speicher



ros2_control - Architektur

- Beispiel:
- Treiber schreibt einen Zustand

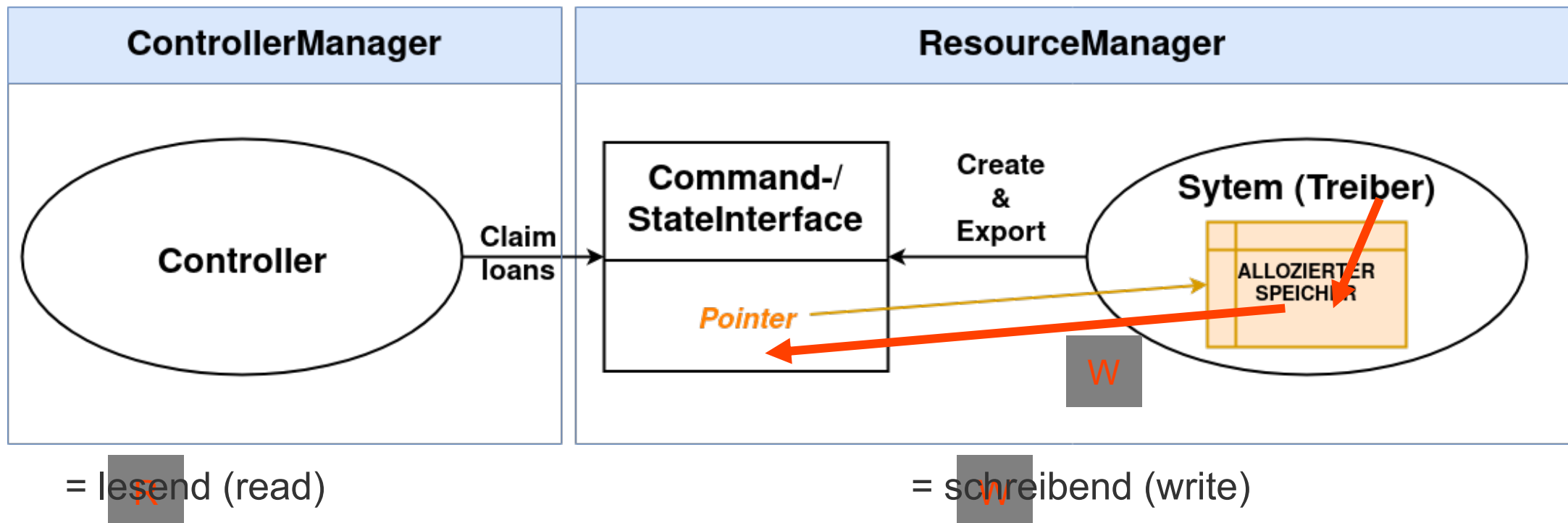


= lesend (read)

= schreibend (write)

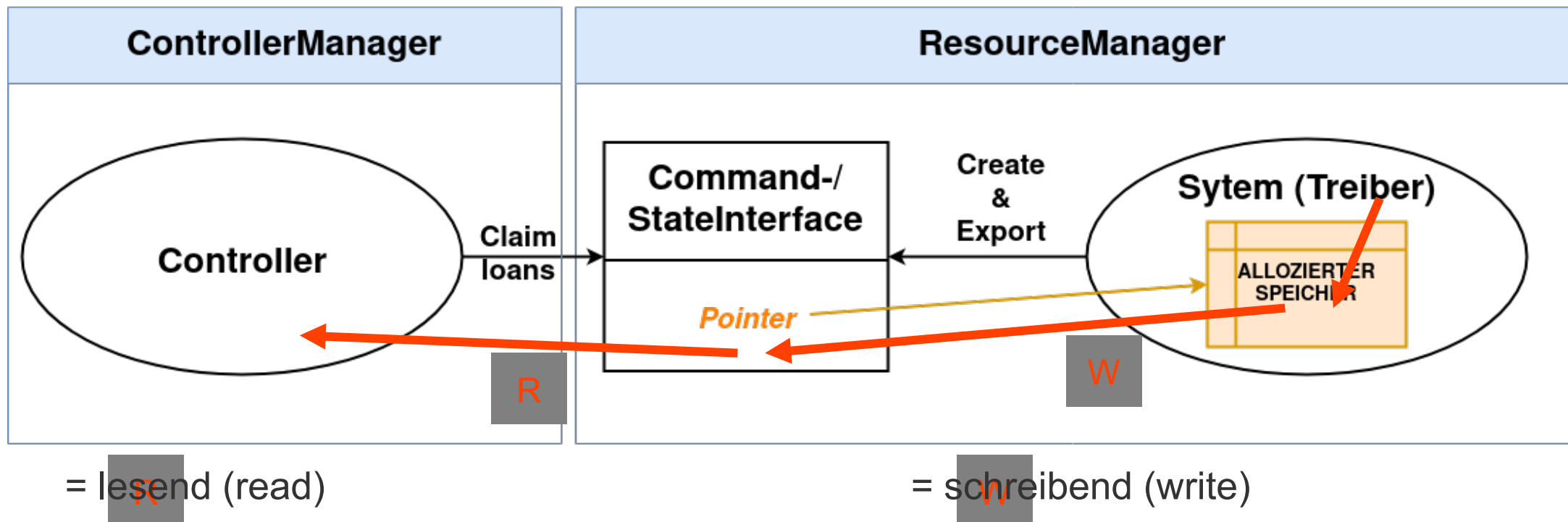
ros2_control - Architektur

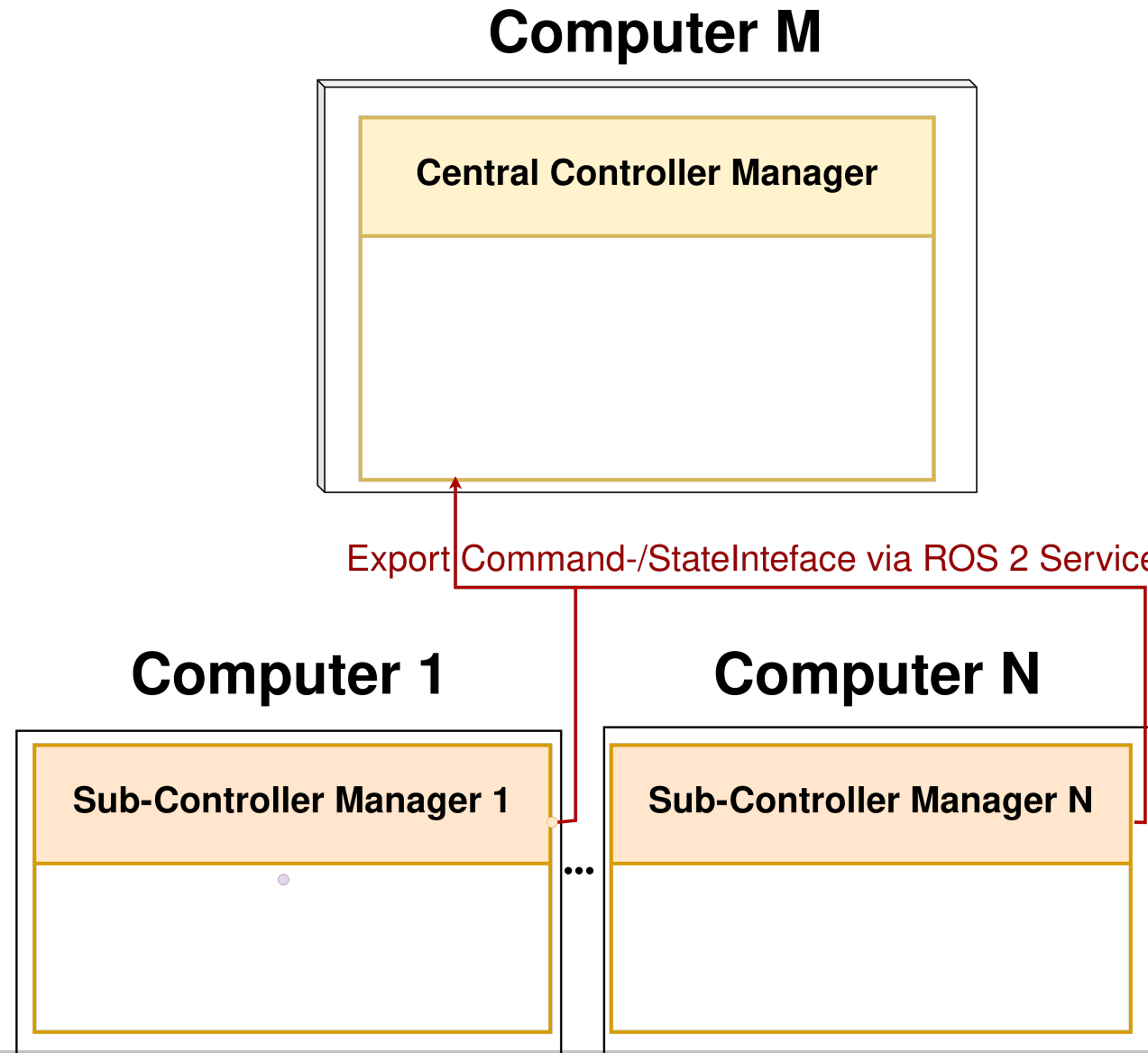
- Beispiel:
- Treiber schreibt einen Zustand



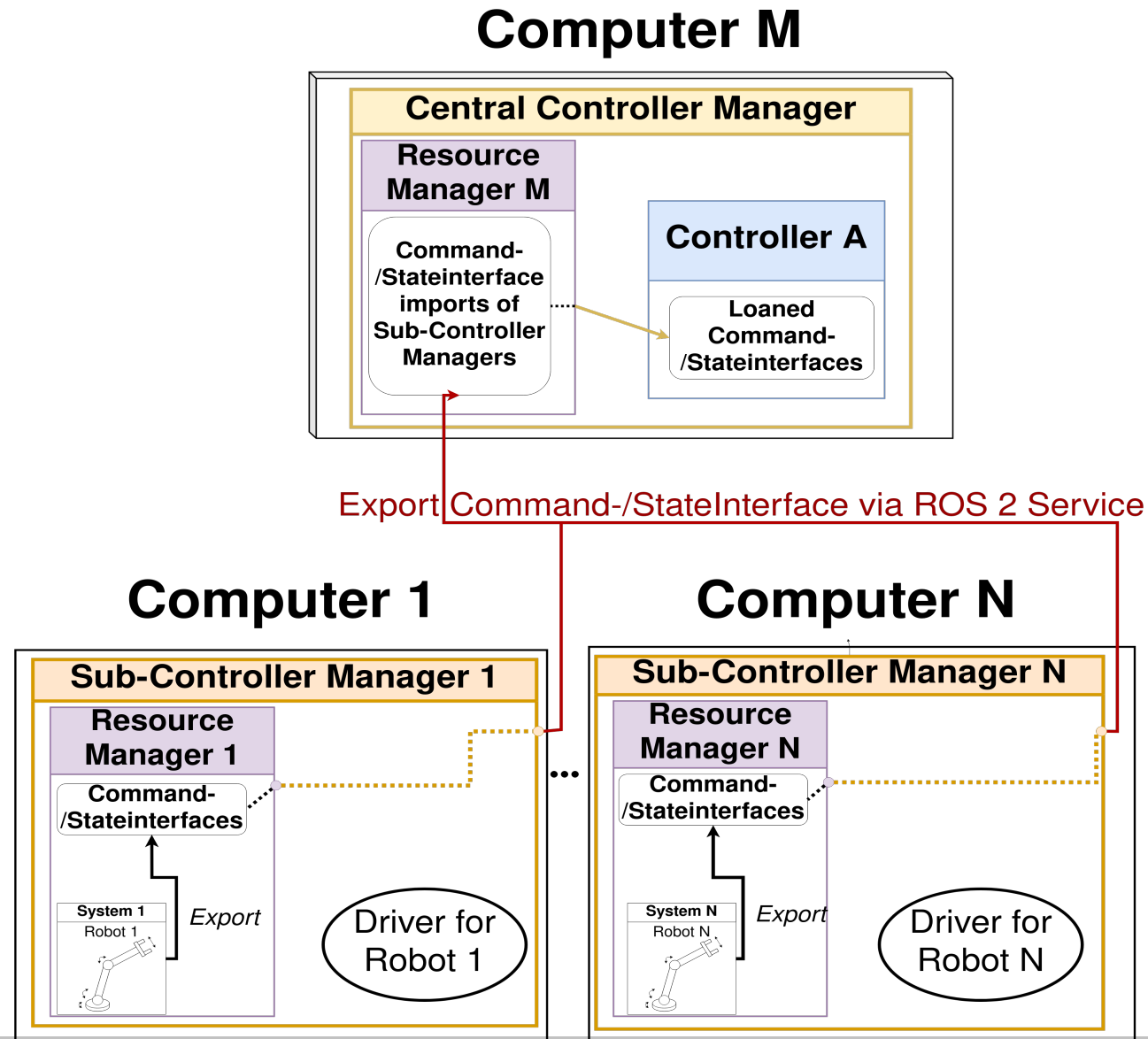
ros2_control - Architektur

- Beispiel:
- Treiber schreibt einen Zustand



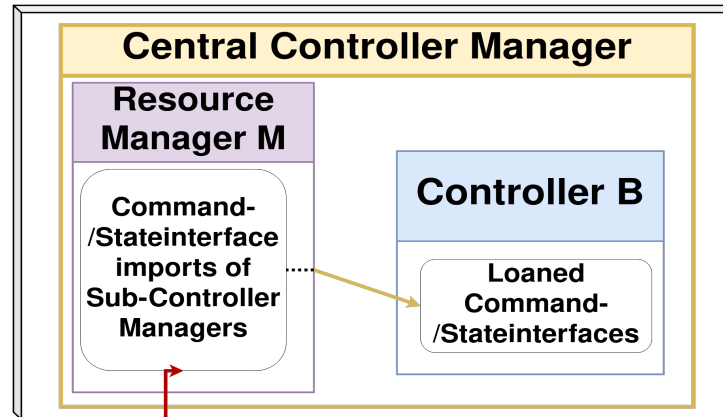


1.) Verteilte Treiber



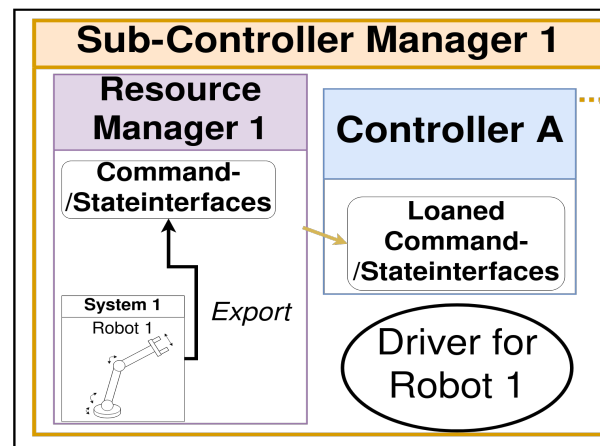
2.) Verteilte verkettet Regler

Computer M

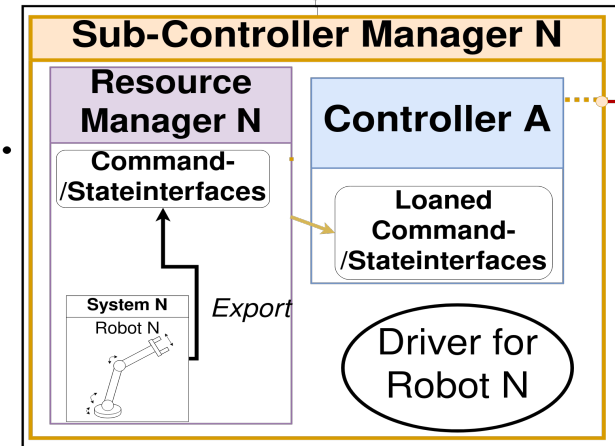


Export Command-/StateInterface via ROS 2 Service

Computer 1



Computer N



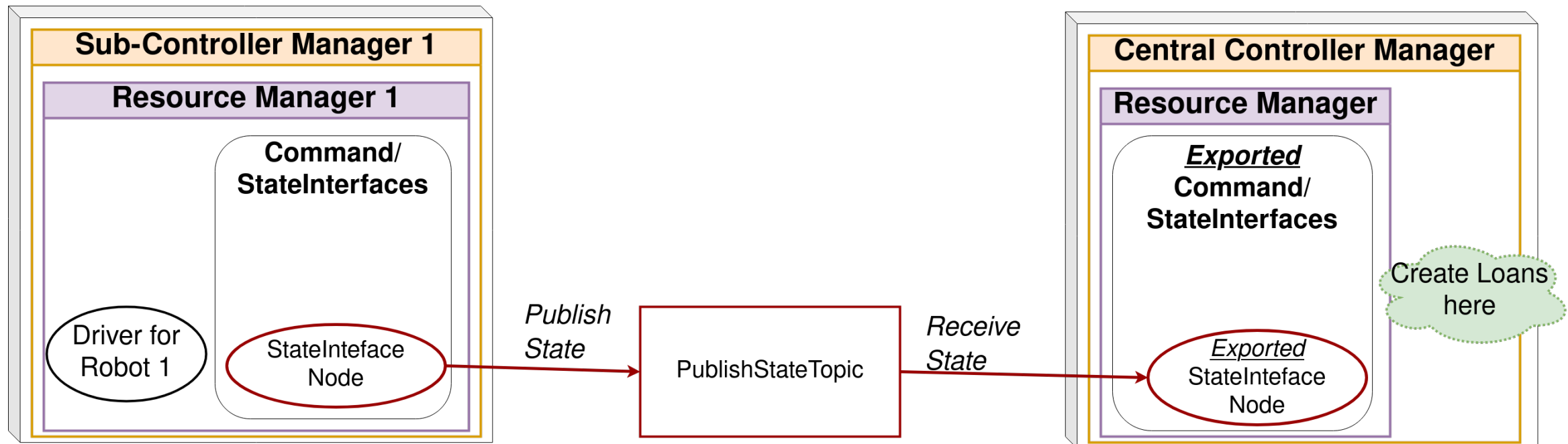
Konzept

- Adaption der Handles:
- Beschreibung wird exportiert
- Erzeugen von Publishers/Subscribers

Computer 1

ROS 2 Nodes and Topics

Computer 2



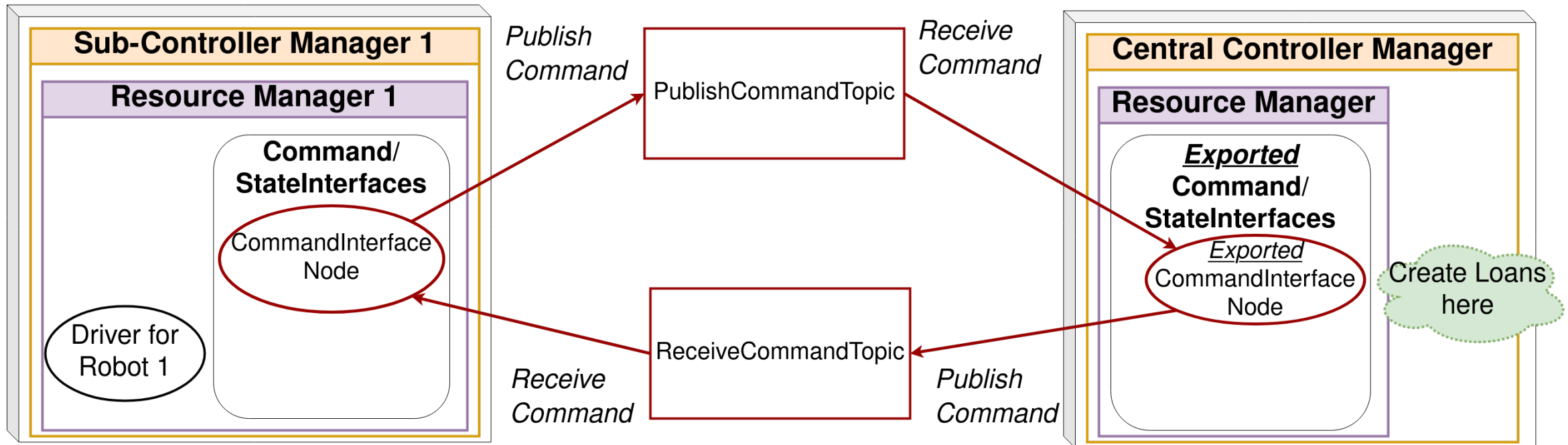
Konzept

- Adaption der Handles:
- Beschreibung wird exportiert
- Erzeugen von Publishers/Subscribers

Computer 1

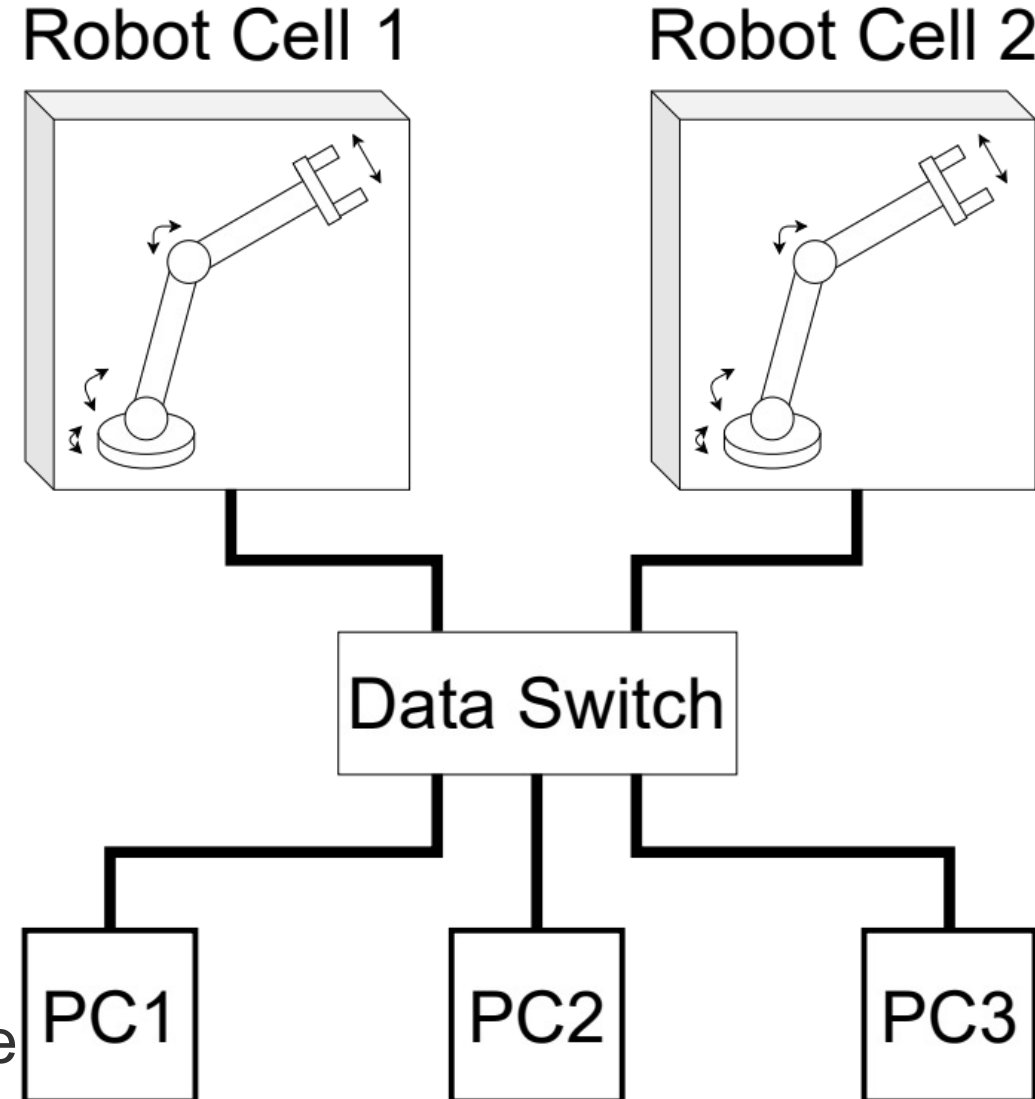
ROS 2 Nodes and Topics

Computer 2



Evaluation

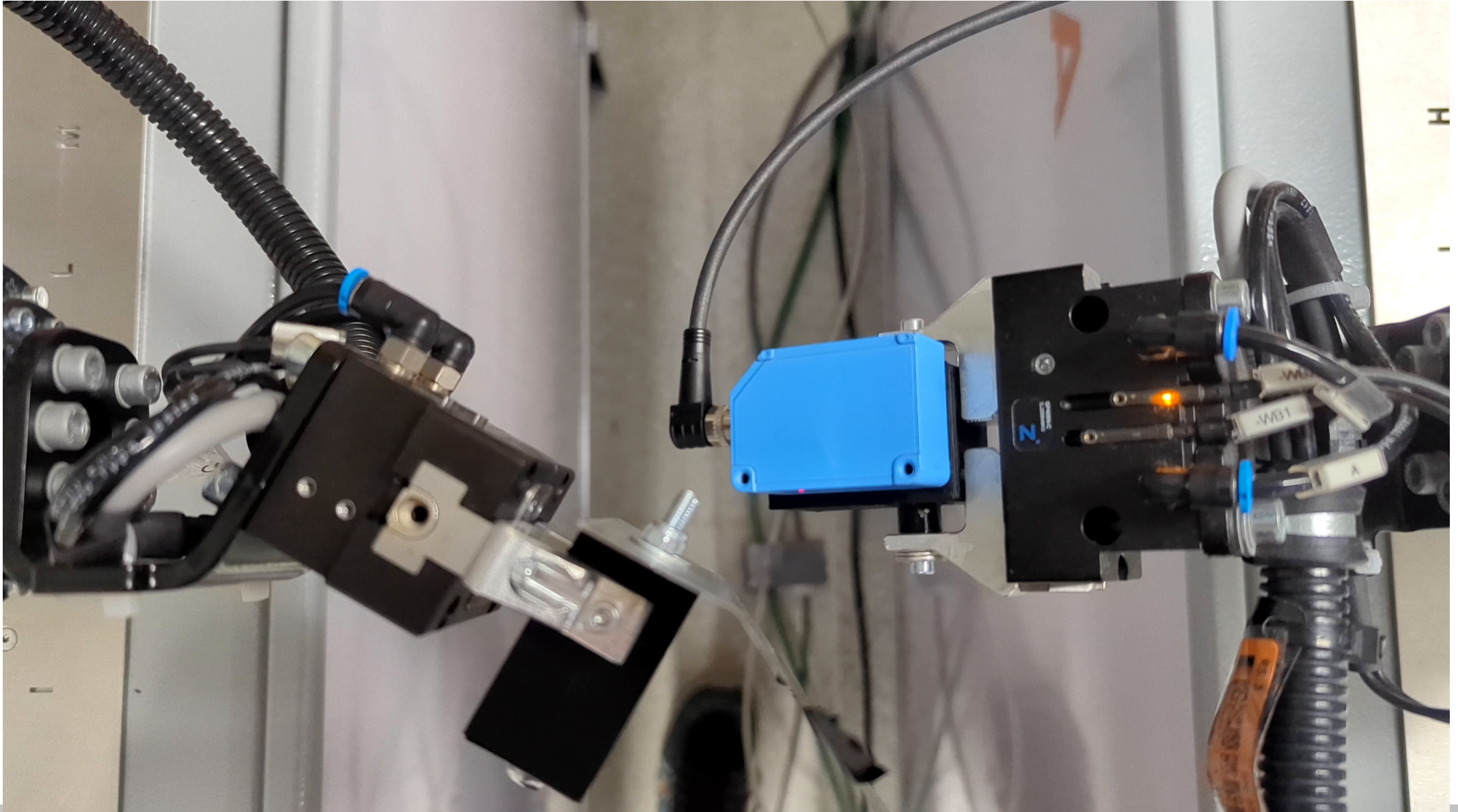
- 2 Roboterzellen
 - KUKA KR3 R 540
 - ready2_educate
- 1 Netzwerk Switch
 - Separates Netzwerk
 - Kein zusätzlicher Netzwerkverkehr
- 3 verschiedene Computer
 - RT-PREEMPT: 5.15.0-1048-realtime



Evaluation



Evaluation



Evaluation

- DDS Middlewares und QoS Profile:

QoS Profil	DDS Vendor	Fast DDS	Cyclone DDS	Connex DDS
Sensor Profil		X	X	X
Reliable Profil		X	X	X

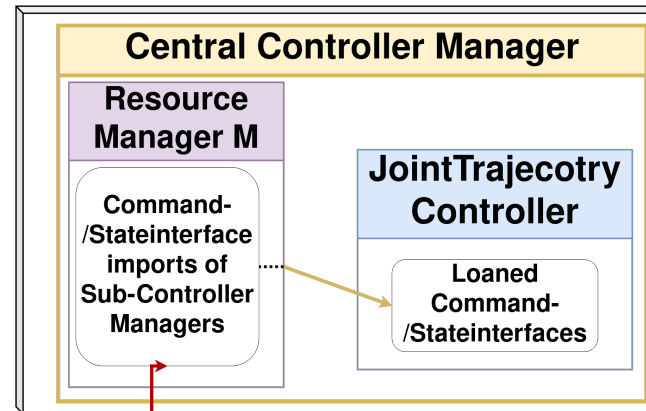
Evaluation

- DDS Middlewares und QoS Profile:

QoS Profil \ Policy	Reliability
Sensor Profil	BEST_EFFORT
Reliable Profil	RELIABLE

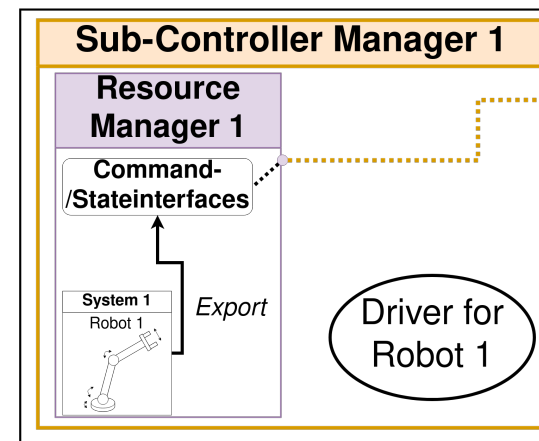
Evaluation: Verteilte Treiber

Computer 1

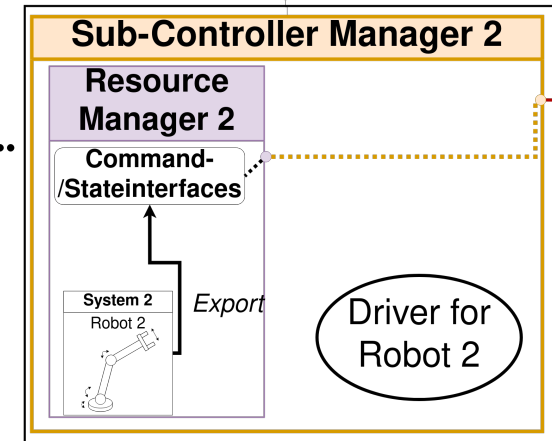


Export Command-/StateInterface via ROS 2 Service

Computer 2

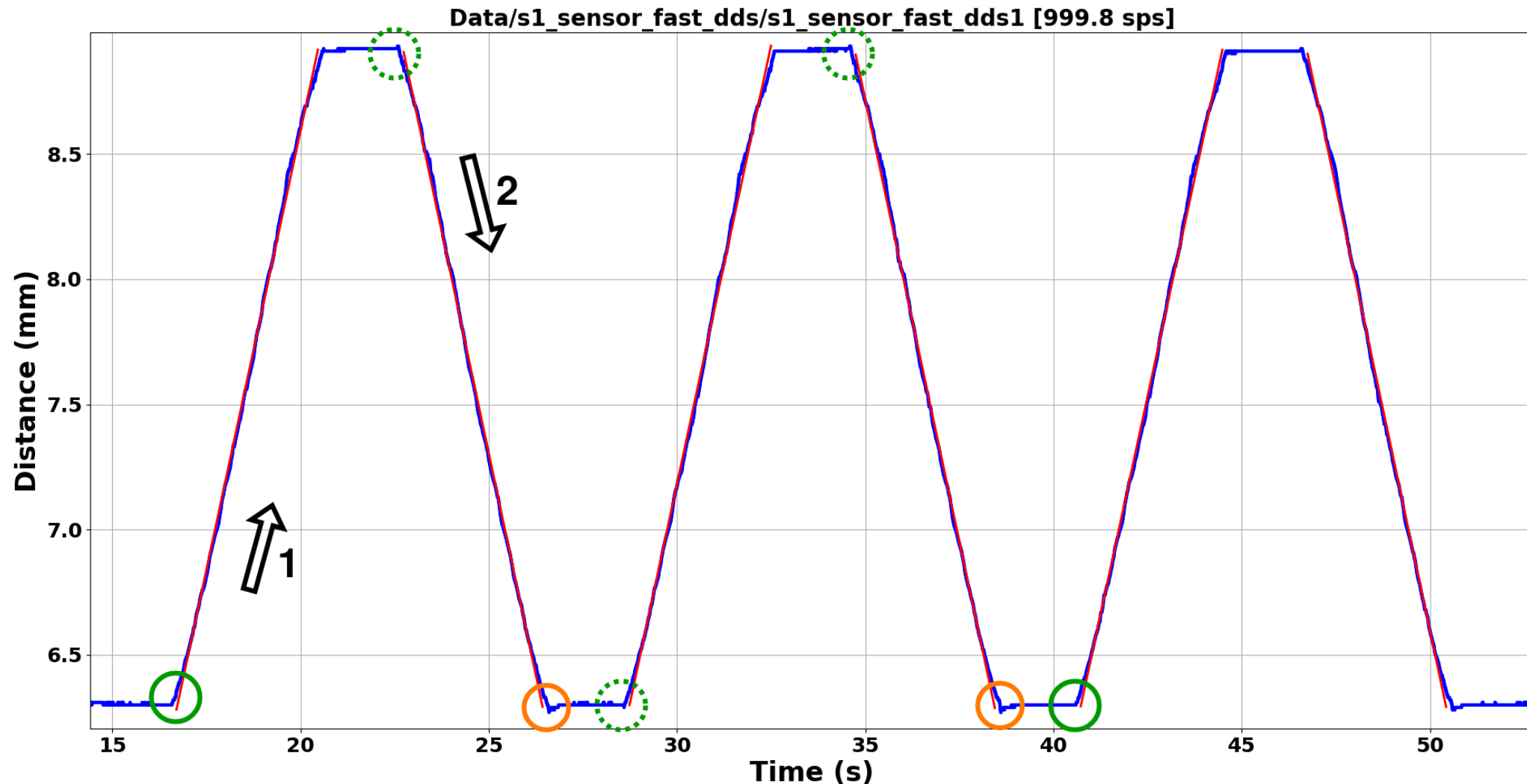


Computer 3



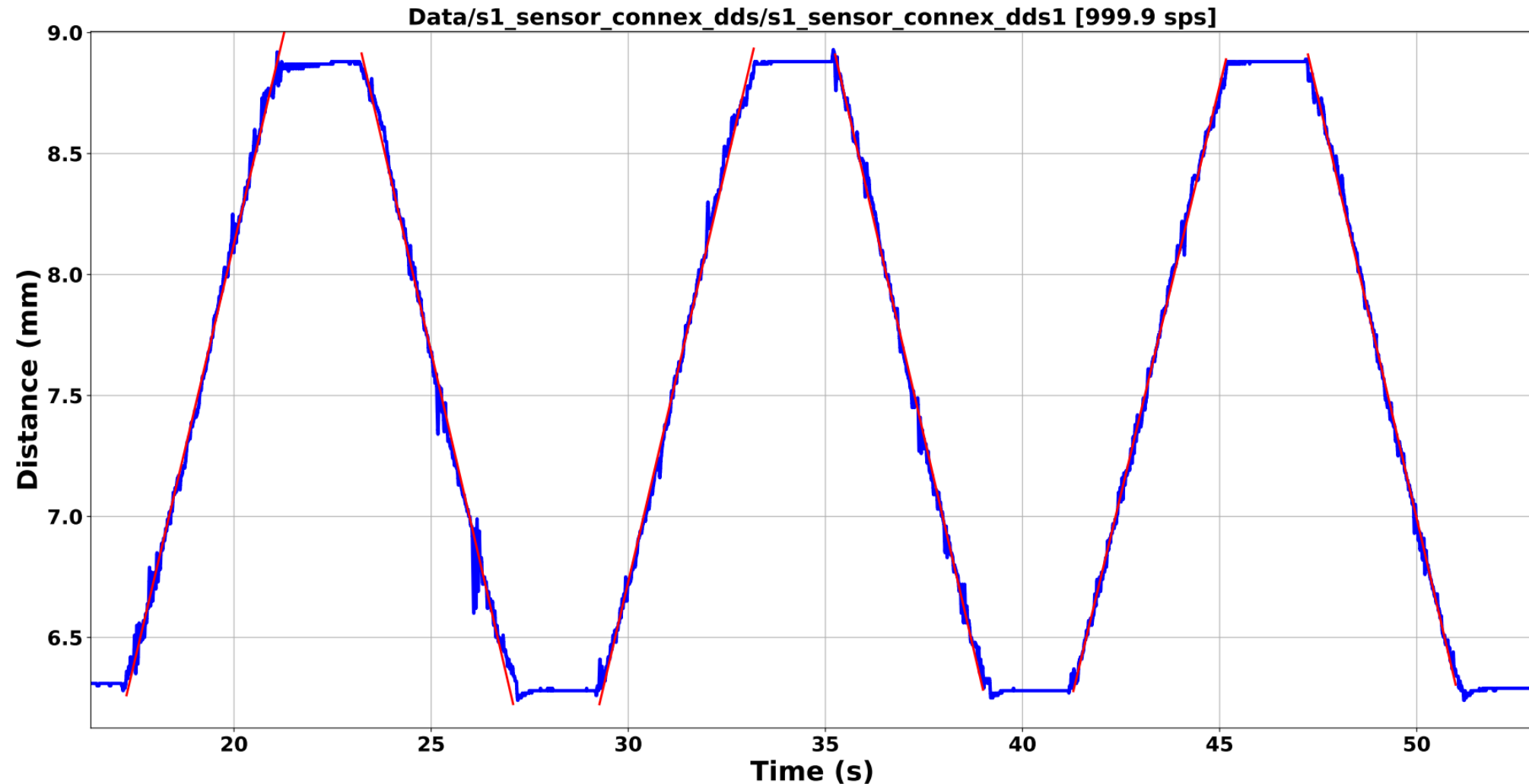
Evaluation: Verteilte Treiber

- Fast DDS & Sensor Profil



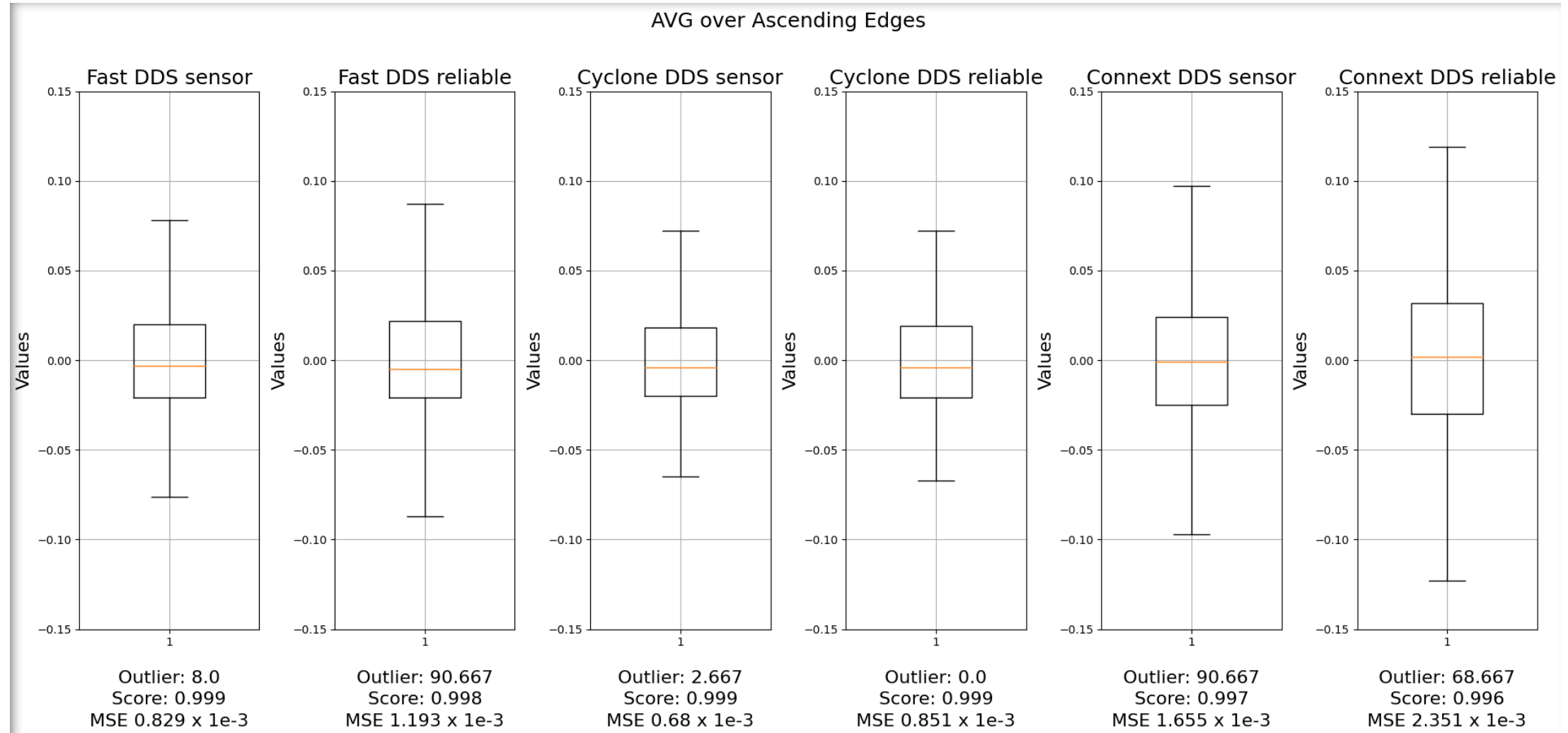
Evaluation: Verteilte Treiber

- Connex DDS & Sensor Profil



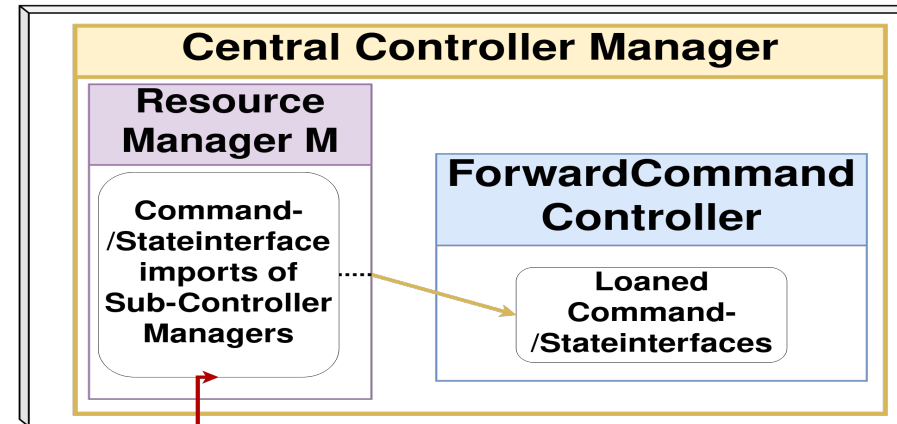
Evaluation: Verteilte Treiber

- Übersicht Kombinationen



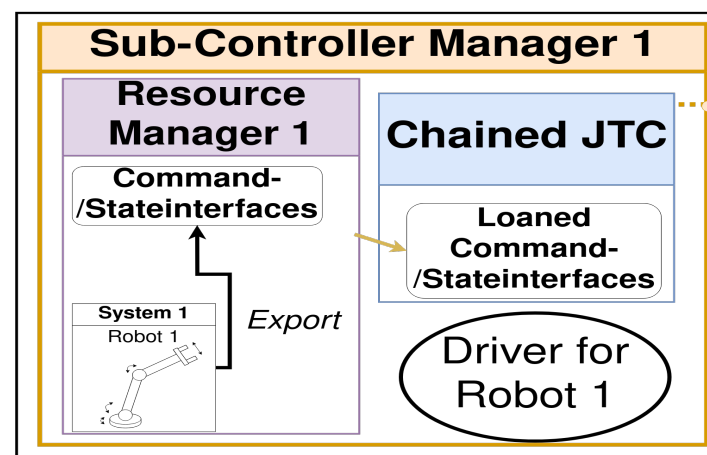
Eval.: Verteilte verkettete Regler

Computer 1

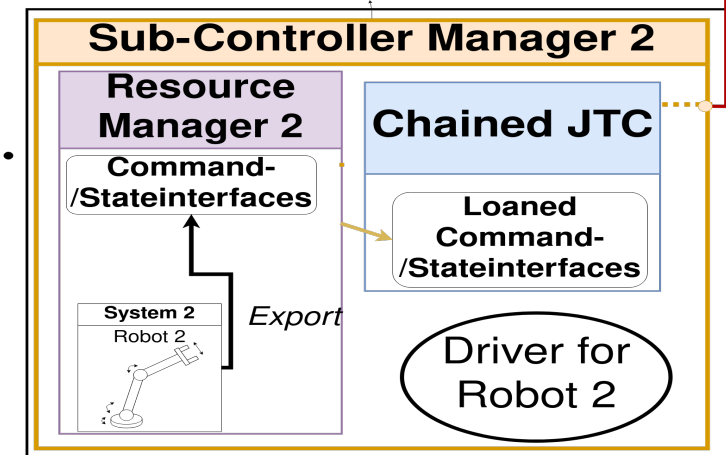


Export Command-/StateInterface via ROS 2 Service

Computer 1

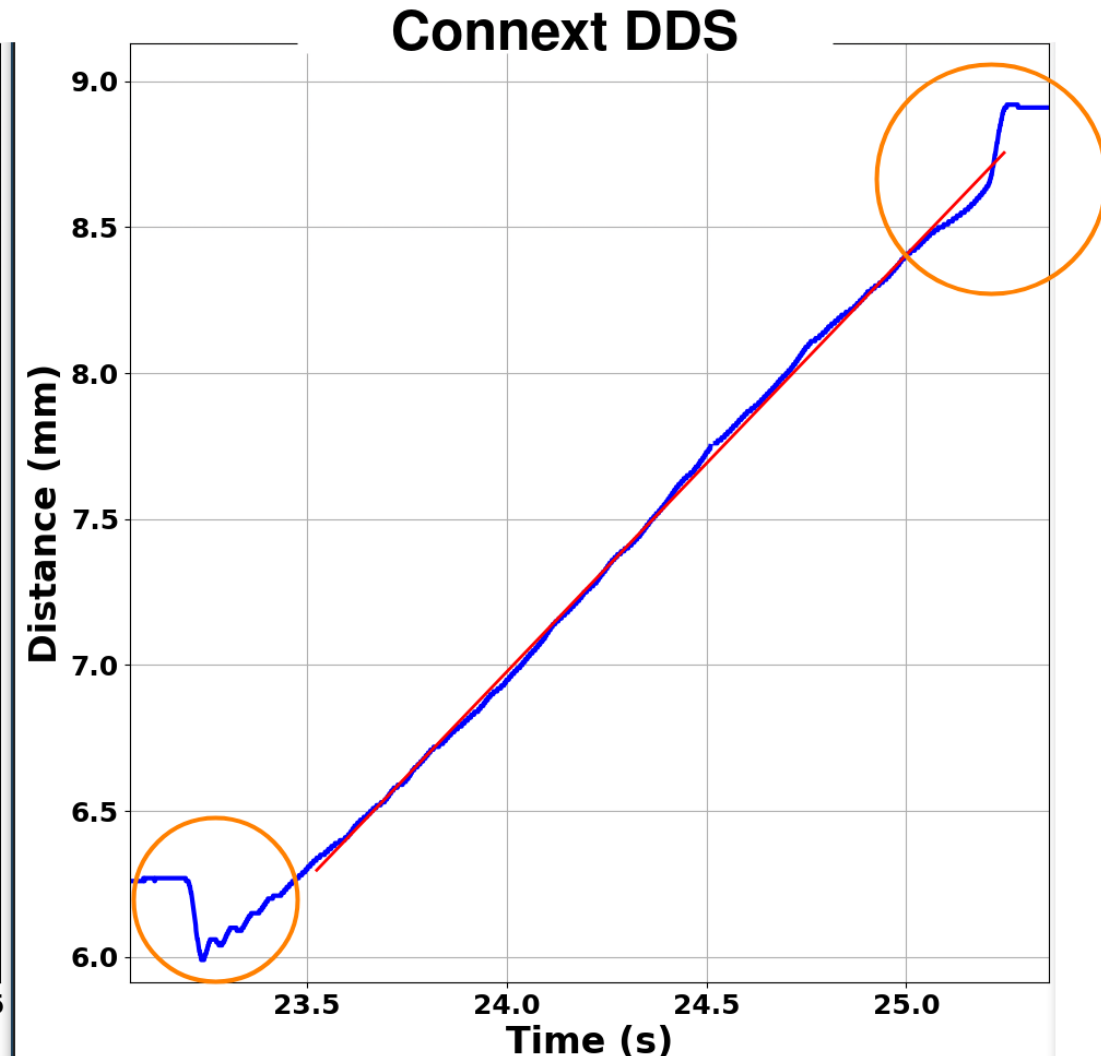
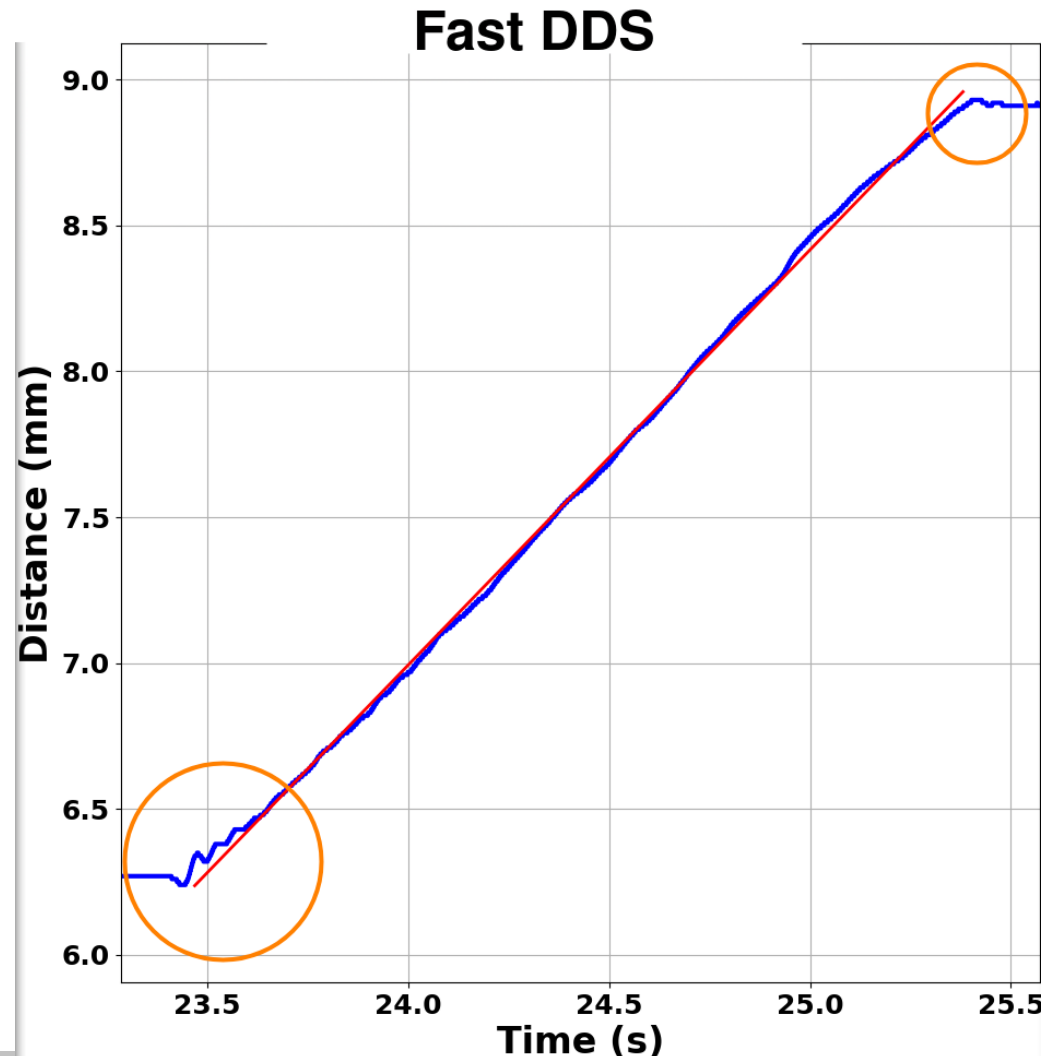


Computer 2



Eval.: Verteilte verkettete Regler

- Fast DDS vs Connect DDS mit Sensor Profil



Vielen Dank!

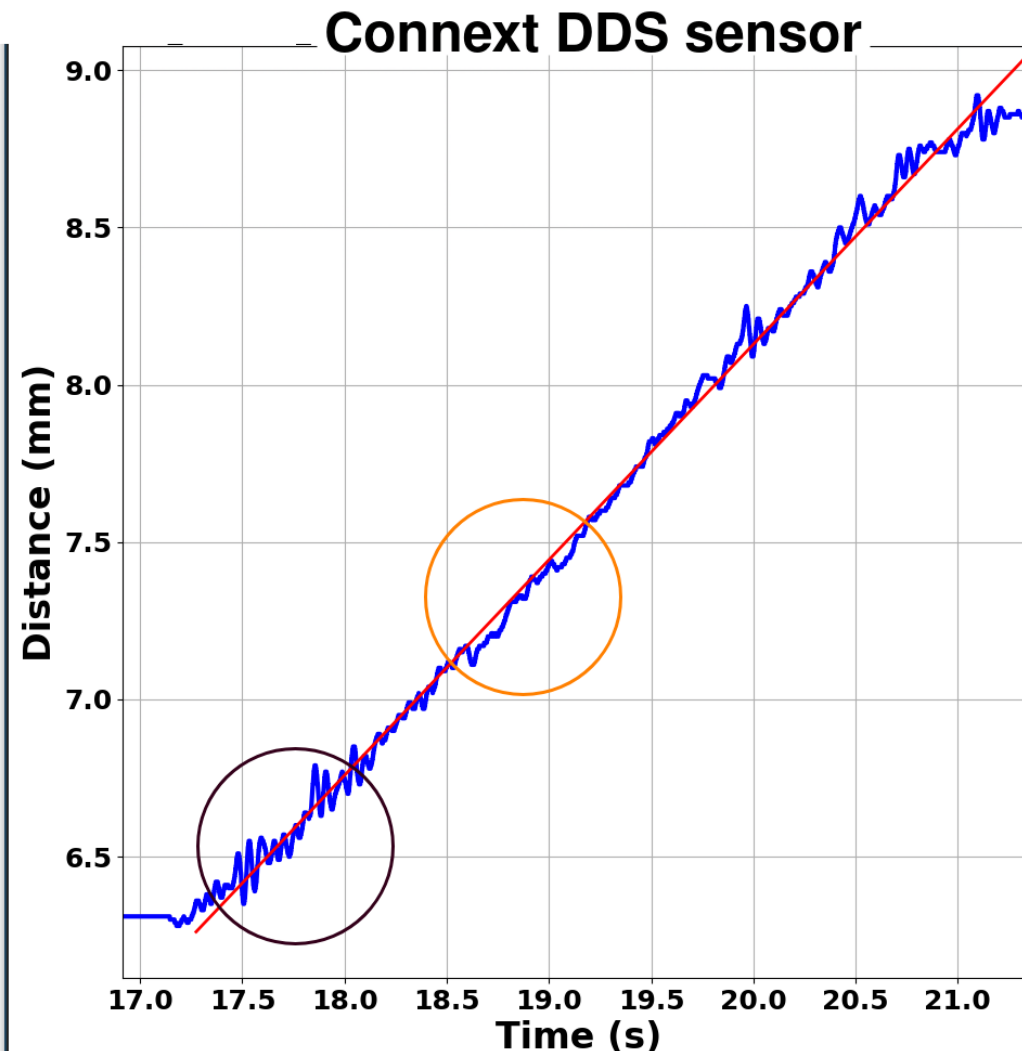
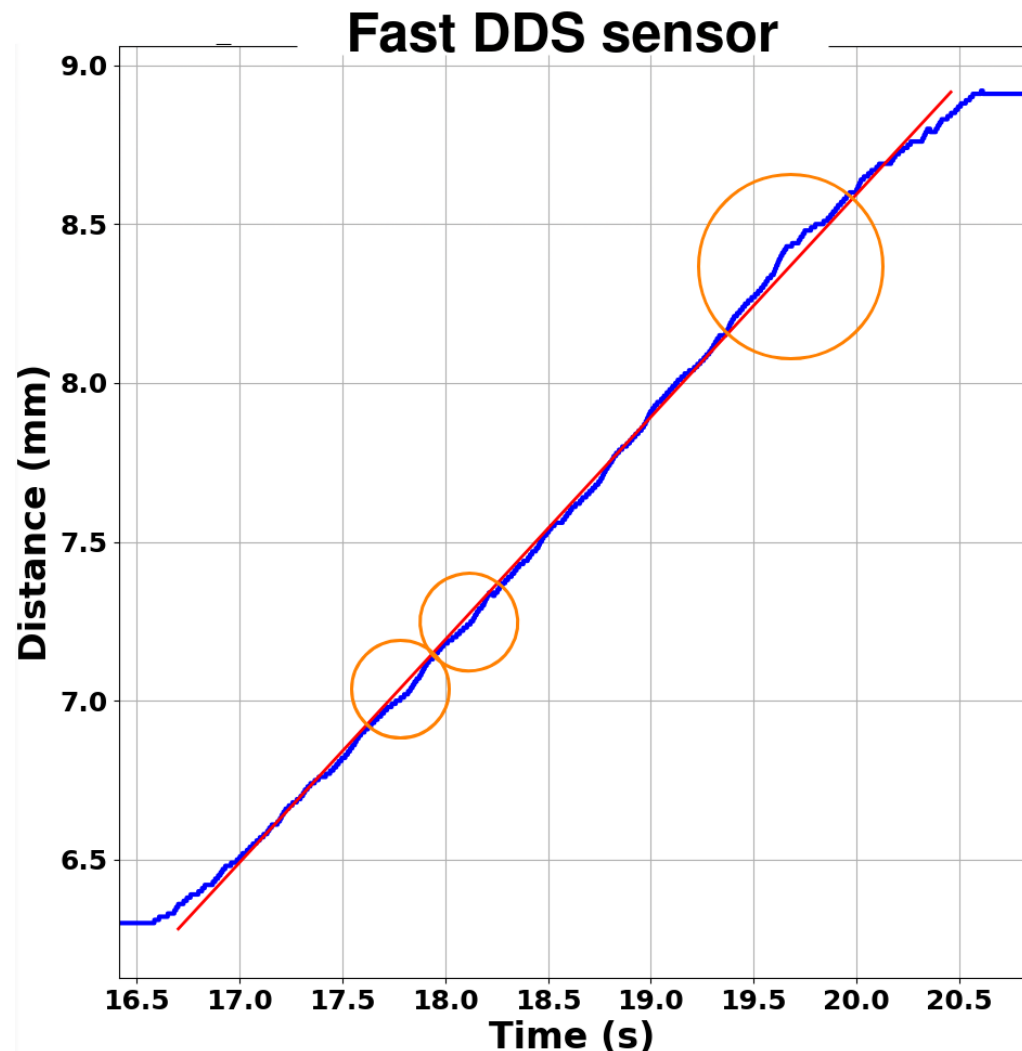
ROSCON 2023

Manuel Muth M.Sc.

Dr. Denis Štogl

Szenario 1

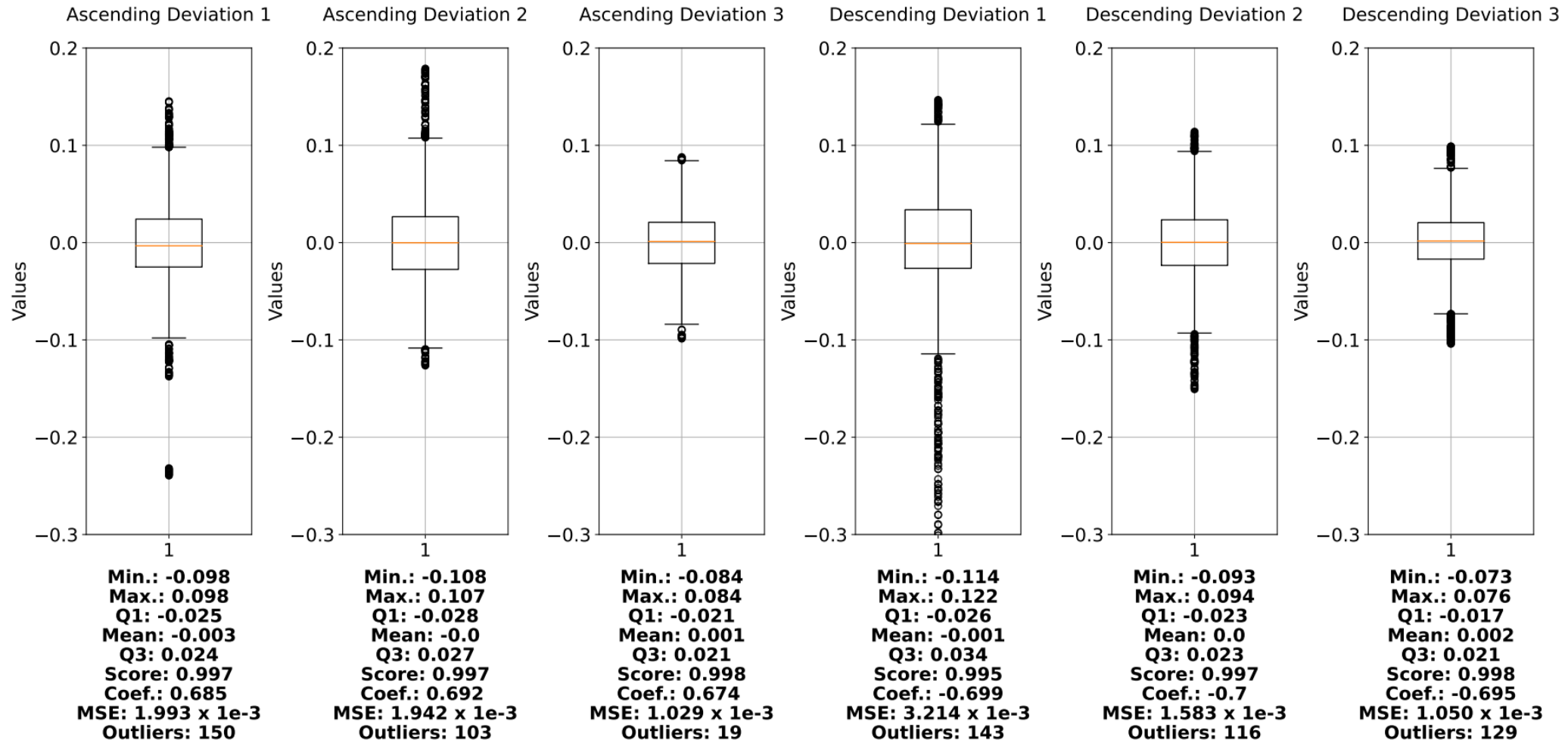
- Verteilte Treiber: Fast DDS sensor vs Connex DDS sensor



Szenario 1

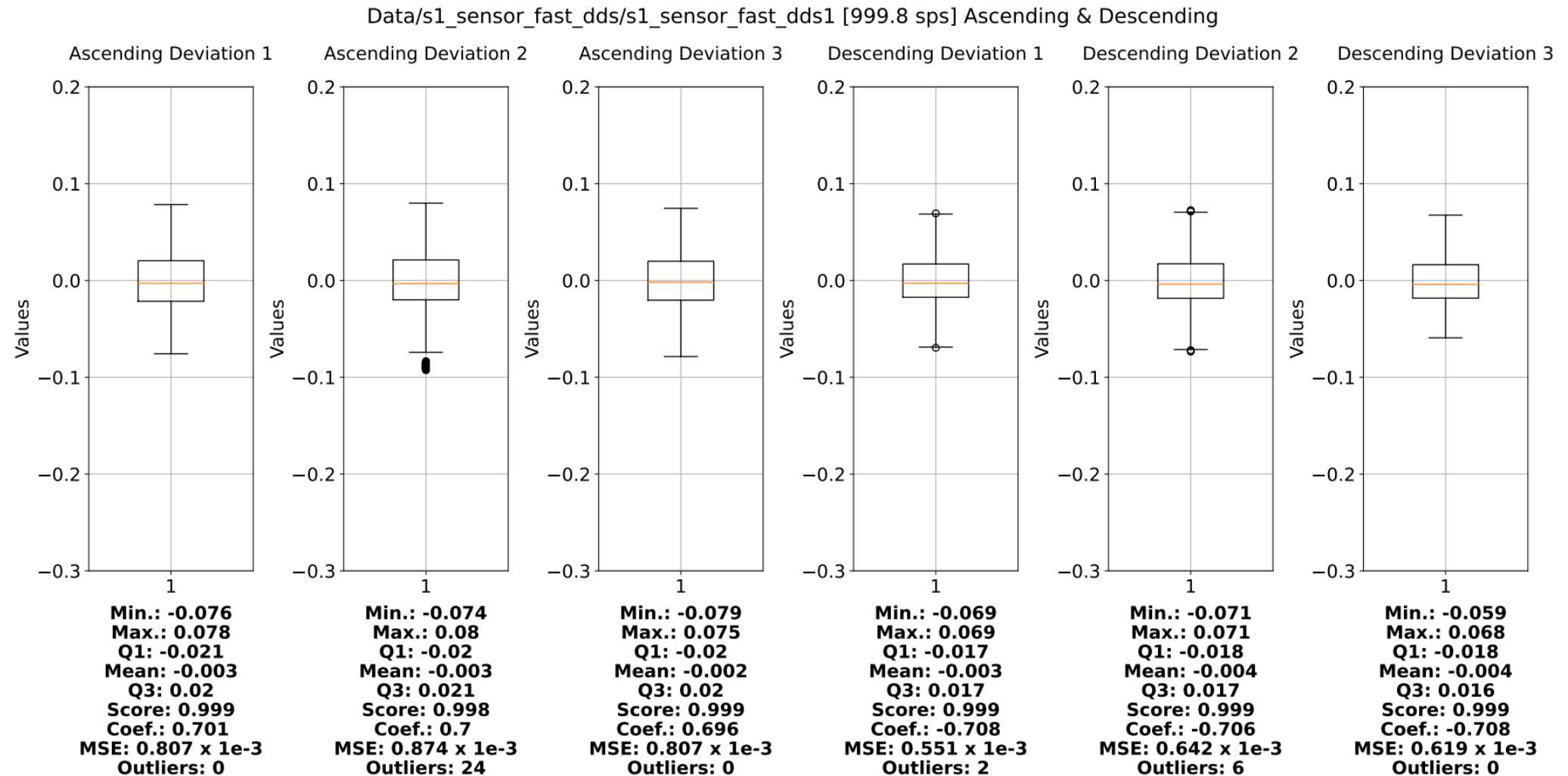
- Verteilte Treiber: Connex DDS Sensor Profil

Data/s1_sensor_connex_dds/s1_sensor_connex_dds1 [999.9 sps] Ascending & Descending



Szenario 1

- Verteilte Treiber: Fast DDS Sensor Profil



Zelle

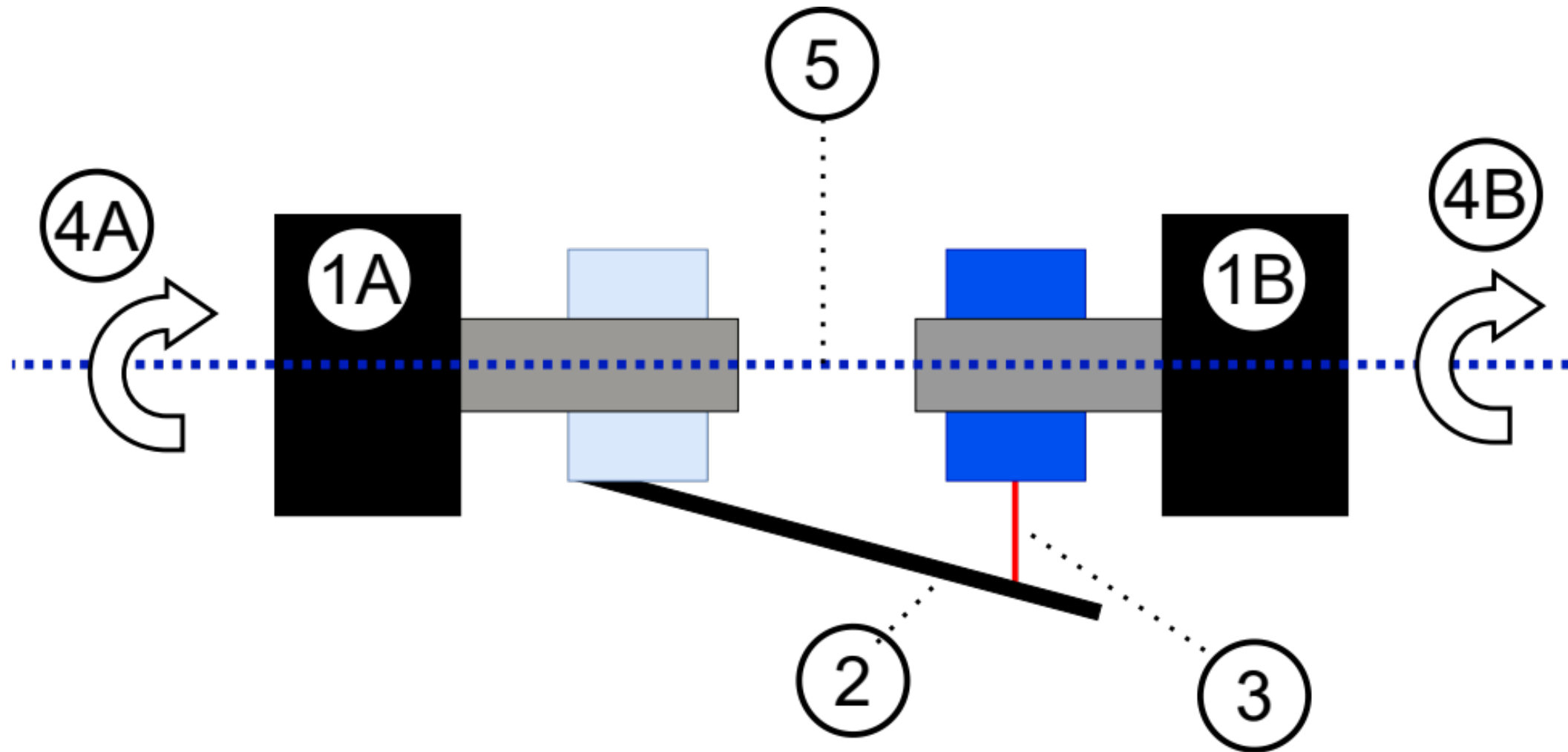
KUKA KR3 R 540

- ready2_educate



Evaluation

- Abstandsmessung mittels Distanzsensor:



Echtzeitfähigkeit ROS 2

- Seit der initialen Veröffentlichung 2018 mehrere Untersuchungen
 - Echtzeit Kernel (Preempt-RT)
 - Ping Pong Latenztest
 - Für initiale Version Ardent: weiche Echtzeitfähigkeit möglich [1]
- 2021 Eloquent
 - Mit Nutzung von PTP
 - Echtzeitfähiger Speicherverwaltung
 - Verbesserung der Latenzen $< 1\text{ms}$ $\rightarrow$ bis 1 kHz [2]
 - Weitere Untersuchungen $\rightarrow$ Harte Echtzeit möglich [3]

- [1] Carlos San Vicente Gutiérrez, Lander Usategui San Juan, Irati Zamalloa Ugarte, and Víctor Mayoral Vilches. Towards a distributed and real-time framework for robots: Evaluation of ROS 2.0 communications for real-time robotic applications, September 2018. URL <http://arxiv.org/abs/1809.02595>. ArXiv:1809.02595
- [2] L. Puck, P. Keller, T. Schnell, C. Plasberg, A. Tanev, G. Heppner, A. Roennau, and R. Dillmann. Distributed and Synchronized Setup towards Real-Time Robotic Control using ROS2 on Linux. In 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE), pages 1287–1293, August 2020. doi: 10.1109/CASE48305.2020.9217010. ISSN: 2161-8089.
- [3] L. Puck, P. Keller, T. Schnell, C. Plasberg, A. Tanev, G. Heppner, A. Roennau, and R. Dillmann. Performance Evaluation of Real-Time ROS2 Robotic Control in a Time-Synchronized Distributed Network. In 2021 IEEE 17th International Conference on Automation Science and Engineering (CASE), pages 1670–1676, August 2021. doi: 10.1109/CASE49439.2021.9551447. ISSN: 2161-8089.

Who are we?

